



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE CIÊNCIAS EXATAS E DA TERRA  
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA  
APLICADA  
BACHARELADO EM ENGENHARIA DE SOFTWARE



# Vemcar: aplicativo de caronas solidárias para a comunidade acadêmica

Thiago de Lima Andrade

Natal-RN  
Junho de 2017

Thiago de Lima Andrade

Vemcar: aplicativo de caronas solidárias para a  
comunidade acadêmica

Proposta de Monografia de Graduação  
apresentada ao Departamento de Informática  
e Matemática Aplicada do Centro de Ciências  
Exatas e da Terra da Universidade Federal do  
Rio Grande do Norte como requisito parcial  
para a obtenção do grau de bacharel em  
Engenharia de Software.

Orientador

Dr. Gibeon Soares de Aquino Junior

Universidade Federal do Rio Grande do Norte – UFRN  
Departamento de Informática e Matemática Aplicada – DIMAp

Natal-RN

Junho de 2017

Catálogo da Publicação na Fonte. UFRN / SISBI / Biblioteca Setorial  
Centro de Ciências Exatas e da Terra – CCET.

Andrade, Thiago de Lima.

Vemcar, o aplicativo de caronas solidárias para a comunidade acadêmica /  
Thiago de Lima Andrade. - Natal, 2017.  
88f.: il.

Orientador: Prof. Dr. Gibeon Soares de Aquino Junior.

Monografia (Graduação) – Universidade Federal do Rio Grande do Norte.  
Centro de Ciências Exatas e da Terra. Departamento de Informática e Matemática  
Aplicada. Bacharelado em Engenharia de Software.

1. Engenharia de software – Monografia. 2. Mobilidade urbana – Monografia. 3.  
Aplicativo – Monografia. 4. Carona – Monografia. 5. Offdroid – Monografia. 6.  
MQTT – Monografia. I. Aquino Junior, Gibeon Soares de. II. Título.

RN/UF/BSE-CCET

CDU: 004.41

Monografia de Graduação sob o título “*Vemcar, o aplicativo de caronas solidárias para a comunidade acadêmica*” apresentada por Thiago de Lima Andrade e aceita pelo Departamento de Informática e Matemática Aplicada do Centro de Ciências Exatas e da Terra da Universidade Federal do Rio Grande do Norte, sendo aprovada por todos os membros da banca examinadora abaixo especificada:

---

Dr. Gibeon Soares de Aquino Junior

Orientador

Departamento de Informática e Matemática Aplicada  
Universidade Federal do Rio Grande do Norte

---

Me. Jean Guerethes Fernandes Guedes

Superintendência de Informática

Universidade Federal do Rio Grande do Norte

---

Me. Mário Andrade Vieira de Melo Neto

Departamento de Informática e Matemática Aplicada  
Universidade Federal do Rio Grande do Norte

Natal-RN, 12 de junho de 2017.

Ao meu pai e minha mãe, que me ensinam a ser quem sou. Amo vocês.

# Agradecimentos

Este trabalho não seria possível sem o apoio dos meus pais Maria do Carmo e José Edilson, que, durante toda a graduação esteve ao meu lado apoiando minhas escolhas, inclusive em relação ao intercambio. Sou profundamente grato também aos amigos que conheci na Itália, durante o CsF. A estes amigos, Lenilson e Renato, agradeço as conversas cotidianas no intercambio sobre problemas políticos e urbanos do Brasil. Foram nessas conversas que percebi os problemas de mobilidade urbana que enfrentamos.

Ao meu amigo Marcel, agradeço a paciência durante períodos de bolsa onde precisávamos desenvolver aplicativos e naquela época não tínhamos tanto conhecimento técnico. Mas sempre estudamos e conseguimos cumprir as obrigações da bolsa e aprender os princípios do desenvolvimento em Android. A Shay e Vinicius, agradeço toda ajuda, dedicação e tempo que foi compartilhado comigo em cada momento vivido.

Graças a 4Soft e aos meus amigos de empresa júnior, tive a oportunidade de desbravar novos caminhos no curso. Agradeço a todos esses meus amigos que me chamam de Peter, por suas influências positivas em cada “sabadev” que pude participar.

Em especial, agradeço aos professores Gibeon e Itamir, pelas oportunidades que surgiram na disciplina de desenvolvimento móvel. Poder participar da SINFO como resultado de um projeto de disciplina foi fundamental e indispensável para que esse trabalho chegasse onde chegou. Por isso, sou inteiramente grato a equipe PADAWANS e de design que me ajudaram diariamente a me tornar um melhor programador, além de contribuírem de forma direta com código para o Vemcar.

Obrigado!

*“A respostas certa, não importa nada: o essencial é que as perguntas estejam certas.”*

Mário Quintana.

# Vemcar: aplicativo de caronas solidárias para a comunidade acadêmica

Autor: Thiago de Lima Andrade

Orientador: Dr. Gibeon Soares de Aquino Junior

## RESUMO

Mobilidade urbana é um tema que requer atenção no planejamento urbano dentro das grandes cidades, ou mesmo em espaços reduzidos como campi de universidades. Em áreas urbanas a população comumente enfrenta problemas de mobilidade devido à grande quantidade de carros em circulação. Além disso, em instituições de ensino como a Universidade Federal do Rio Grande do Norte (UFRN) a quantidade de ônibus coletivo de uso público não atende o fluxo de pessoas em determinados horários do dia, prejudicando a mobilidade. Nesse contexto analisa-se caronas como uma maneira de contribuir para que algumas pessoas tenham outra opção de mobilidade, enquanto outros podem ceder uma vaga no seu veículo de forma solidária. Este trabalho introduz todo o processo de desenvolvimento de software e validação da ideia do Vemcar, um aplicativo de caronas para a comunidade acadêmica da UFRN.

*Palavras-chave:* Mobilidade urbana, aplicativo, carona, REST, MQTT.

# Vemcar: a carpool app for the academic community

Author: Thiago de Lima Andrade

Advisor: D.r Gibeon Soares de Aquino Junior

## ABSTRACT

Urban mobility is a topic that requires attention in urban planning of big cities or even in reduced fields such as university campuses. In urban areas, the population commonly faces mobility problems due to the large number of cars. Furthermore, in educational institutions such as Federal University of Rio Grande do Norte (UFRN), the number of school buses does not meet the flow of people at certain times. In this scenario, it is analyzed carpooling as a way of contributing to what some people have other mobility option. This work introduces the entire process of software development and validation of Vemcar idea, a carpool app for the academic community of UFRN.

*Keywords:* Urban mobility, car shared, mobile app, REST, MQTT.

# Lista de figuras

Figura 1. Tempo médio (em minutos) no deslocamento casa-trabalho – regiões metropolitanas selecionadas no Brasil e no mundo. Fonte: (SCHWANEN et al, 2013).....	20
Figura 2. Número de carros no RN entre 2012 e 2016.....	21
Figura 3. Número de habitantes no RN entre 2012 e 2016. ....	21
Figura 4. Taxa de crescimento (%) do número de automóveis e habitantes no RN.....	21
Figura 5. Postagem realizada para o questionário no perfil da SINFO no Facebook.....	24
Figura 6. Divisão dos tipos de vínculos nas respostas do questionário.....	27
Figura 7. Divisão dos gêneros no questionário. ....	27
Figura 8. Frequência de deslocamento para o campus nos turnos de aula. ....	28
Figura 9. Divisão dos entrevistados que já receberam e/ou ofereceram caronas ou não.....	29
Figura 10. Escala de importância de alguns aspectos para um serviço de caronas. ....	31
Figura 11. Tipos de caronas que as pessoas mais se interessam. ....	32
Figura 12. Ciclo de vida de uma Activity (DEVELOPER, 2016).....	37
Figura 13. Navigation Drawer e FAB (DEVELOPER, 2016).....	38
Figura 14. Componentes do paradigma Pub/Sub.....	42
Figura 15. Value Proposition Canvas (OSTERWALDER, 2014).....	45
Figura 16. Imagem promocional retirada do site do Caronaphone. ....	47
Figura 17. Imagem promocional retirada do site do Caronaê. ....	48

Figura 18. Quadro de post-its da equipe mobile da SINFO com legenda de cores dos projetos.....	55
Figura 19. Exemplo de nota adesiva no quadro. ....	55
Figura 20. Foto do quadro de post-its da SINFO.....	56
Figura 21. Diagrama de atividades de uma tarefa no quadro.....	57
Figura 22. VPC construído com base na Tabela 5. (GRILO A.; NETO, 2016). .....	60
Figura 23. Rede semântica da palavra segurança. (GRILO A.; NETO, 2016)	61
Figura 24. Rede semântica das palavras coletividade e integração. (GRILO A.; NETO, 2016).....	61
Figura 25. Resultado da votação dos nomes.....	62
Figura 26. Evolução da paleta de cores do Vemcar.....	69
Figura 27. Storyboard I: instituições, login, termos, instruções e veículo. ...	70
Figura 28. Storyboard II: todas as caronas, minhas caronas, menu lateral, configurações e sobre. ....	71
Figura 29. Storyboard III: solicitar e oferecer carona, TimePicker e caixa de diálogo de confirmação. ....	72
Figura 30. Storyboard IV: detalhes, mapa de uma carona e caixa de diálogo de confirmação de saída. ....	73
Figura 31. Relação entre Fragment, Activity e Adapter no Vemcar. ....	77
Figura 32. Pacotes do Vemcar e suas classes. ....	78
Figura 33. Autenticação e autorização na API.sistemas (APISISTEMAS, 2017). .....	80
Figura 34. Diagrama de classes do pacote model que utilizam a classe PersisteBD do OffDroid.....	82
Figura 35. Diagrama de sequência de uma solicitação de lista de caronas..	83
Figura 36. Diagrama de classes do pacote MQTT. ....	84

# Lista de tabelas

Tabela 1. Pacote de controle do MQTT.....	44
Tabela 2. Comparativo entre aplicativos de serviço de mobilidade. ....	49
Tabela 3. Autores do Vemcar com registro de patente no INPI. ....	54
Tabela 4. Projetos desenvolvidos na SINFO. ....	54
Tabela 5. Seleção de opiniões retiradas da pesquisa da seção 2.5.....	59
Tabela 6. Requisitos funcionais do Vemcar. ....	64
Tabela 7. Especificação dos requisitos funcionais do Vemcar. ....	68
Tabela 8. Relação entre requisitos funcionais e telas do Vemcar.....	75
Tabela 9. Anotações do OffDroid nas classes de domínio do Vemcar.....	82

# Lista de abreviaturas e siglas

UFRN – Universidade Federal do Rio Grande do Norte

DIMAp – Departamento de Informática e Matemática Aplicada

IPEA – Instituto de Pesquisa Econômica Aplicada

UFPE – Universidade Federal de Pernambuco

DENATRAN – Departamento Nacional de Trânsito

IBGE – Instituto Brasileiro de Geografia e Estatística

TCU – Tribunal de Contas da União

Ceftru – Centro Interdisciplinar de Estudos em Transporte

UnB – Universidade de Brasília

UFRJ – Universidade Federal do Rio de Janeiro

UX – User Experience (Experiência de Usuário)

SINFO – Superintendência de Informática da UFRN

SIGAA – Sistema Integrado de Gestão de Atividades Acadêmicas

MQTT - Message Queuing Telemetry Transport

REST – Representational State Transfer

VPC – Value Proposition Canvas

# Sumário

<b>1. Introdução.....</b>	<b>14</b>
1.1. Problemática.....	15
1.2. Proposta .....	15
1.3. Metodologia .....	16
1.4. Organização do trabalho .....	16
<b>2. Motivação.....</b>	<b>18</b>
2.1. Trafego de automóveis e qualidade da mobilidade urbana .....	19
2.2. Frota de carros no Rio Grande do Norte .....	20
2.3. Caronas e o futuro da mobilidade .....	22
2.4. Pesquisa sobre aplicativo de caronas na UFRN.....	23
2.4.1. Perguntas do questionário.....	25
2.4.2. Análise das respostas.....	27
<b>3. Fundamentação teórica .....</b>	<b>36</b>
3.1. Componentes e linha de design do Android .....	36
3.1.1. Activity e Fragment .....	36
3.1.2. Design Support Library .....	38
3.1.3. Material Design.....	38
3.2. Serviços REST .....	39
3.3. OffDroid.....	41
3.4. Padrão Publish/Subscribe .....	41
3.5. Protocolo MQTT .....	43
3.6. Value Proposition Canvas (VPC) .....	44

<b>4. Estado da prática .....</b>	<b>46</b>
4.1. Caronaphone .....	46
4.2. Caronaê.....	47
4.3. Serviços de carona remunerada .....	48
4.4. Comparativo .....	49
<b>5. Proposta de solução .....</b>	<b>51</b>
5.1. Como surgiu .....	51
5.1.1. Sala de aula .....	51
5.1.2. SINFO.....	53
5.2. Scrum adaptado na SINFO .....	54
5.3. Nome do aplicativo.....	58
5.3.1. Definição da proposta de valor do aplicativo .....	59
5.3.2. Redes semânticas e resultado.....	60
5.4. Requisitos funcionais.....	62
5.4.1. Especificação dos requisitos funcionais .....	65
5.5. Interface de usuário .....	68
5.5.1. Paleta de cores .....	68
5.5.2. Storyboards .....	69
<b>6. Implementação da solução.....</b>	<b>76</b>
6.1. Modelagem estrutural .....	76
6.2. Acesso e autorização utilizando API.sistemas .....	79
6.3. Persistência de dados com OffDroid .....	80
6.4. Uso do MQTT.....	83
<b>7. Considerações finais .....</b>	<b>85</b>
<b>Referências.....</b>	<b>87</b>

# 1. Introdução

Transporte urbano é um tema relevante por impactar diariamente na qualidade de vida de muitas pessoas. Nas últimas décadas percebeu-se um aumento significativo na frota de automóveis nas grandes cidades do Brasil. Estudos (SCHWANEN *et al.*, 2013) comprovam que a qualidade da mobilidade urbana em cidades é influenciada pela quantidade de veículos em circulação. Porém é possível analisar questões de mobilidade em escopos menores, como em uma universidade.

Algumas universidades oferecem para a comunidade a opção de transporte coletivo gratuito. A Universidade Federal do Rio Grande do Norte (UFRN) é um exemplo de instituição com este serviço, onde tais transportes coletivos são chamados simplesmente de “circular”. Porém, em alguns horários do dia, quando aumenta o fluxo de pessoas para UFRN, é comum ocorrer uma superlotação nos transportes circulares. Este fato impacta diretamente na qualidade do serviço oferecido.

Assim como em uma grande cidade, dentro da universidade é fácil observar que muitas pessoas utilizam carros como meio de transporte. Nesse sentido, surgem ofertas e pedidos de caronas de maneira informal ou por postagens em grupos específicos de caronas nas redes sociais. Então foi desenvolvido o Vemcar, um aplicativo para dispositivos móveis que facilita a comunicação entre membros da comunidade acadêmica que desejam solicitar e/ou oferecer caronas.

Este trabalho introduz sobre o Vemcar todas as etapas do processo de desenvolvimento do software, desde: a concepção e validação da ideia, o levantamento e análise de requisitos, o projeto de interface de usuário, o desenvolvimento e manutenção do software, os testes, a implantação, a entrega para o usuário final e o recolhimento de feedbacks.

## 1.1. Problemática

Especialistas estudam cada vez mais os impactos do crescimento da frota de veículos nas grandes cidades e como esse crescimento pode estar relacionado com os frequentes congestionamentos no trânsito. Pesquisas elaboradas no decorrer do trabalho e apresentadas na seção 2.2 mostram que nos últimos anos o estado do Rio Grande do Norte registrou taxas de crescimento da frota de carros maiores que as taxa de crescimento do número de habitantes.

O individualismo do uso do carro afeta então a qualidade da mobilidade urbana. No meio acadêmico isso não é diferente. Muitas pessoas utilizam veículos particulares para se locomoverem no trajeto ida e volta para a universidade. Desse modo, pessoas que moram umas próximas das outras utilizam o mesmo percurso para irem e voltarem das universidades, porém não o mesmo veículo.

Recentemente surgiram alguns aplicativos que oferecem serviços pagos de compartilhamento de carros, como Uber, 99Taxi, dentre outros. Porém, nesses aplicativos não é possível garantir que um determinado usuário é de fato membro de uma universidade, o que pode levar a uma situação de risco para o passageiro da carona.

Alguns desses aplicativos oferecem serviços de mobilidade que são pagos. Nesse sentido, um aplicativo de caronas solidárias pode auxiliar estudantes de baixa renda que não têm condições financeiras para utilizar tais serviços pagos.

## 1.2. Proposta

Este trabalho propõe a análise e desenvolvimento de um sistema informatizado de caronas de uso exclusivo da comunidade acadêmica da Universidade Federal do Rio Grande do Norte (UFRN). Nomeado de Vemcar, tal sistema tem o objetivo de promover ofertas e solicitações de caronas em um ambiente seguro para seus usuários.

Além de fornecer uma nova opção de mobilidade urbana no trajeto casa-universidade, o Vemcar busca promover a integração de pessoas que compõe a comunidade acadêmica dos campi da universidade.

### 1.3. Metodologia

O aplicativo começou seu desenvolvimento dentro de uma disciplina oferecida no curso de Bacharelado em Engenharia de Software, em seguida, o projeto que até então era um protótipo foi introduzido como projeto da Superintendência de Informática da UFRN.

Lá foi feita uma pesquisa inicial para validação da ideia foi elaborada, publicada e respondida exclusivamente por membro da comunidade acadêmica da UFRN. A partir do resultado obtido, surgiu os primeiros requisitos do sistema e então regras de um processo de software são aplicadas ao projeto.

Técnicas de nomificação e criação de marca foram conduzidas com o apoio da equipe de design. Continuamente foram realizados testes de software, tendo uma versão beta lançada previamente para alguns usuários. O aplicativo foi lançado no dia 31 de março de 2017 e desde essa data vem recolhendo feedbacks e avaliações dos usuários.

### 1.4. Organização do trabalho

Nesta seção é apresentado como está organizado o trabalho, sendo descrito, portanto, do que trata cada capítulo. Conforme a seguir:

- *Capítulo 2 – Motivação:* descreve o problema da crescente frota de carros em comparação com a qualidade da mobilidade urbana. Contém um estudo específico desse crescimento no estado do Rio Grande do Norte. Expõe a prática de caronas como uma solução que pode auxiliar alguns problemas de mobilidade. Por fim, é feito um estudo de viabilidade de ideia para um aplicativo de caronas solidárias;
- *Capítulo 3 – Fundamentação teórica:* traz alguns conceitos que podem auxiliar a leitura dos próximos capítulos. São abordados conceitos de alguns componentes da plataforma Android. É explicado também os

conceitos de uma aplicação web *RESTfull*, introduzindo o *Offdroid*. Além do paradigma *Publish/Subscribe* que é utilizado no *MQTT*. O capítulo é encerrado com a teoria por traz de um *Value Proposition Canvas*;

- *Capítulo 4 – Estado da prática*: são apresentados alguns aplicativos de caronas, similares ao Vemcar, de outras universidades brasileiras. Seguido por serviços pagos de mobilidade e ao final do capítulo é feito um pequeno estudo comparativo entre as soluções apresentadas;
- *Capítulo 5 – Proposta de solução*: sobre o Vemcar, aqui são descritos: como o aplicativo surgiu, o processo de software aplicado na SINFO, as técnicas utilizadas para nomeá-lo, seus principais requisitos funcionais e a interface e interação de usuário;
- *Capítulo 6 – Implementação da solução*: como o próprio título sugere, este capítulo apresenta aspectos técnicos da implementação do Vemcar e sua interação com outros sistemas para garantir que os requisitos sejam cumpridos;
- *Capítulo 7 – Considerações finais*: demonstra os primeiros resultados pós-lançamento do aplicativo na UFRN e as perspectivas de evolução do mesmo.

## 2. Motivação

Impulsionados pela elevada urbanização no país e no mundo, especialistas estudam cada vez mais os impactos do crescimento da frota de veículos nas grandes cidades e como esse crescimento estar relacionado com os frequentes congestionamentos no trânsito.

Estudos elaborados e apresentados na seção 2.2 mostram que no Rio Grande do Norte a frota de carros cresce em uma taxa mais elevada que o crescimento populacional. Em contrapartida, poucos projetos são postos em prática para atender o fluxo crescente de veículos.

Nesse sentido, caronas podem ser vistas como uma forma de mobilidade urbana que ajuda pessoas a se locomoverem, além de otimizar a quantidade de passageiros dentro dos veículos particulares. Com isso, espera-se uma redução na frota de veículos em circulação, além de oferecer uma nova opção de mobilidade para pessoas que dependem de transporte público por vezes superlotados.

Esse capítulo detalha o problema do aumento do tráfego de automóveis na qualidade da mobilidade urbana na seção 2.1. A seção 2.2 retrata especificamente o crescimento da frota de carros dentro do estado do Rio Grande do Norte. Caronas podem ser vistas como uma opção de mobilidade urbana que otimiza a quantidade de carros nas ruas e a seção 2.3 trata desse assunto com mais detalhes. Mediante a isso, serviços de caronas surgem e são descritos na seção 2.4. Já a seção 2.5 trata do estado da arte de sistemas informatizados de caronas apresentando outros sistemas com essa finalidade e realizando comparativos com o Vemcar.

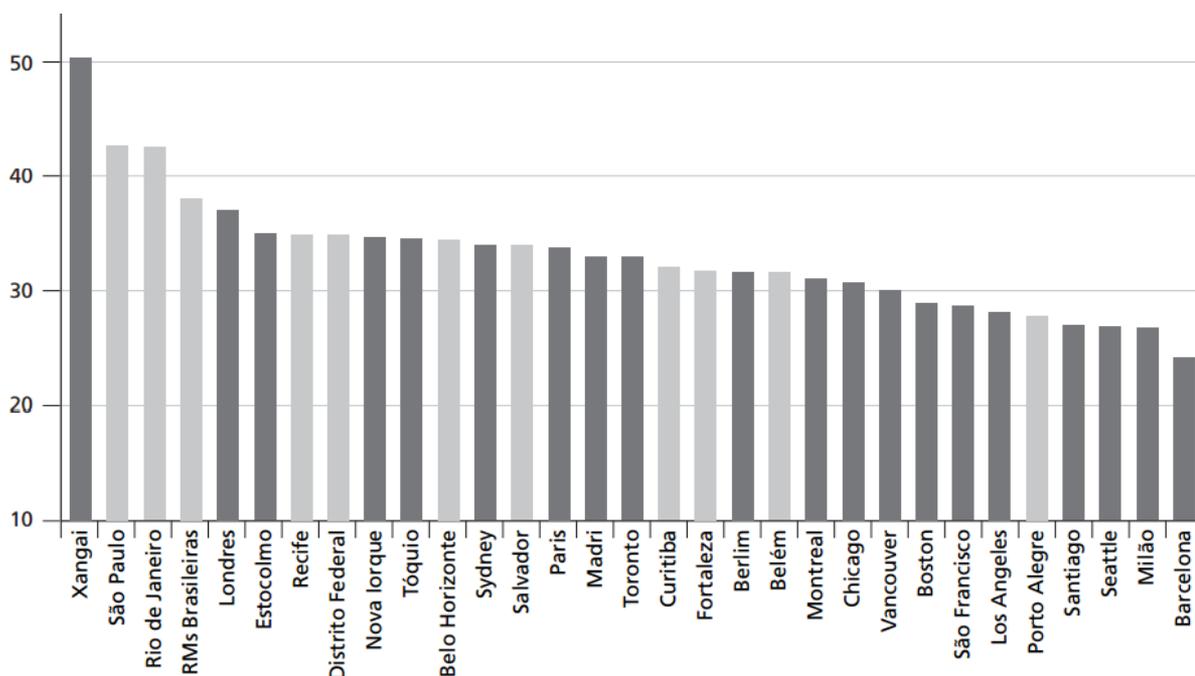
## 2.1. Tráfego de automóveis e qualidade da mobilidade urbana

Jan Gehl, urbanista dinamarquês de renome mundial, cita em seu livro intitulado *Cidade para pessoas* (GEHL, 2013) que a pouco mais de cinquenta anos, em 1961, a jornalista e escritora americana Jane Jacobs, teve seu livro de planejamento urbano *Morte e Vida das Grandes Cidades* publicado pela primeira vez. Nele, Jacobs descrevia como dramático o aumento do tráfego de automóveis nas grandes cidades dos Estados Unidos. Meio século se passou desde a primeira publicação do livro da Jane Jacobs e poucas cidades ao redor do mundo conseguiram executar um planejamento urbano que dá menor prioridade ao tráfego de automóveis em detrimento a outras formas de locomoção para oferecer melhores espaços urbanos aos seus habitantes.

No Brasil, apesar de existirem poucos dados que permitam análises empíricas sobre questões de transporte urbano a nível, o Instituto de Pesquisa Econômica Aplicada (IPEA) disponibilizou em fevereiro de 2013 um estudo comparativo que mostra o tempo gasto no deslocamento casa-trabalho (SCHWANEN *et al*, 2013) em diversas regiões metropolitanas do país.

Segundo esse estudo leva em consideração os seguintes aspectos das regiões metropolitanas observadas: população, área total, densidade demográfica, PIB *per capita*, taxa de motorização e o tempo médio de deslocamento casa-trabalho. No gráfico da Figura 1, adaptado do estudo disponibilizado pelo IPEA, é possível perceber que nas regiões metropolitanas brasileiras (RM's Brasileiras) se gasta mais tempo nesse deslocamento que em várias outras regiões do mundo.

Atrelado ao crescimento vertiginoso do tráfego de automóveis nas cidades, fica evidente o individualismo de tais meios de transporte. O professor de filosofia Érico Andrade da Universidade Federal de Pernambuco (UFPE), em seu artigo *O individualismo motorizado* (ANDRADE, 2012) diz que o carro “expressa a irracionalidade do individualismo: um transporte de duas toneladas para uma pessoa de 80 quilos”. Esse crescimento pode ser visto em diversos estados brasileiros, e não diferente disso, o Rio Grande do Norte mostra um crescimento da frota de veículos, como descrito na seção a seguir.



Fonte: PNAD (IBGE, vários anos); Santiago (Chile), dados disponíveis em: <<http://www.sectra.gob.cl>>; demais países – Toronto Board of Trade (2012).

Notas: <sup>1</sup> Tóquio (2005); Santiago e Europa (2006); Brasil (2009); Austrália, Canadá, Xangai e Estados Unidos (2010).

<sup>2</sup> Os critérios para delimitação das fronteiras das regiões metropolitanas europeias podem variar entre os países. Os dados para os Estados Unidos se baseiam nas regiões metropolitanas americanas (*metropolitan statistical areas*).

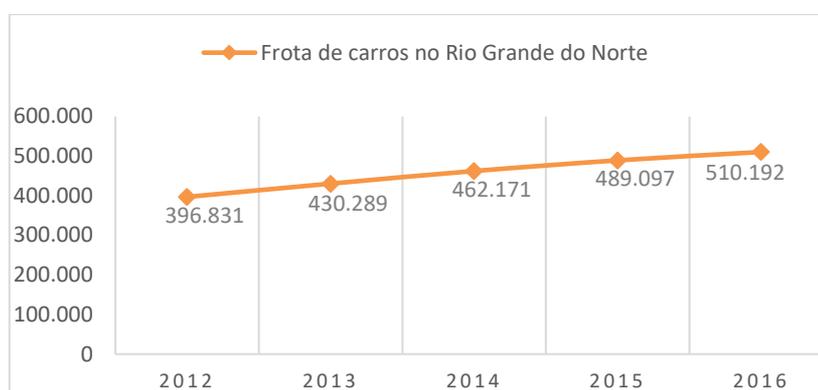
**Figura 1.** Tempo médio (em minutos) no deslocamento casa-trabalho – regiões metropolitanas selecionadas no Brasil e no mundo. Fonte: (SCHWANEN *et al*, 2013).

## 2.2. Frota de carros no Rio Grande do Norte

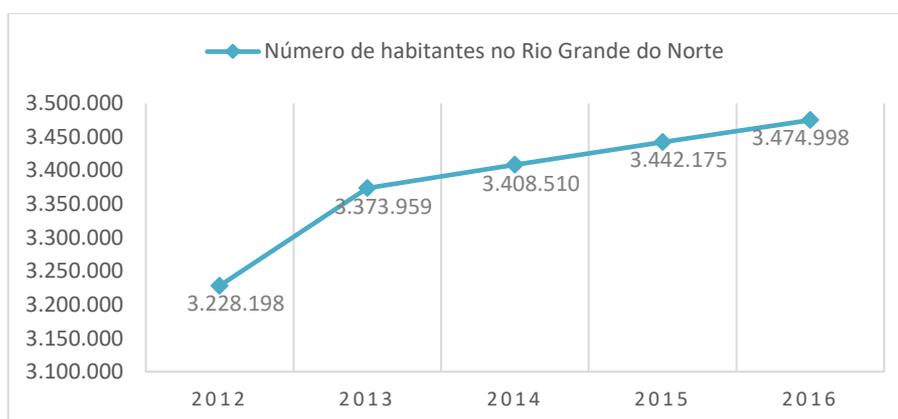
O Departamento Nacional de Trânsito (DENATRAN), pertencente ao Ministério das Cidades, disponibiliza anualmente um relatório sobre a frota de veículos por regiões, estados e municípios do Brasil. O Instituto Brasileiro de Geografia e Estatística (IBGE) faz uma estimativa do número de habitantes de cada município do país com base nos números enviados pelas prefeituras dos municípios para o Tribunal de Contas da União (TCU).

Para demonstrar o crescimento da quantidade de carros no estado do Rio Grande do Norte, foi recolhido dados referentes a frota de automóveis, especificamente carros, do mês de dezembro nos anos de 2012 a 2016 (DENATRAN, 2017) no estado. A partir destes dados o gráfico da Figura 2 foi elaborado. No mesmo período houve um crescimento populacional no estado. Estes dados foram verificados com base nas estimativas apresentadas pelo IBGE (IBGE, 2017) para cada ano. A Figura 3 apresenta um gráfico para o crescimento populacional para o mesmo período do gráfico da Figura 2.

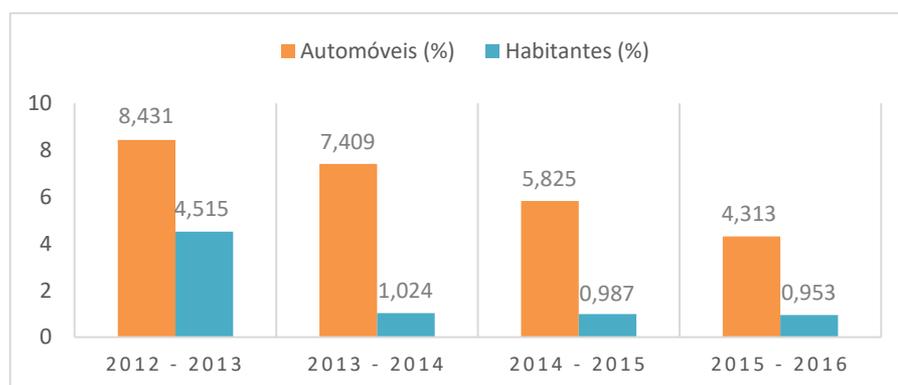
Pelos gráficos da Figura 2 e Figura 3, é possível perceber um aumento tanto na frota de carros quanto no número de habitantes no estado do Rio Grande do Norte. Apesar de ambos fatores aumentarem de um ano para outro, a diferença na taxa de aumento entre automóveis e habitantes é um valor considerável.



**Figura 2.** Número de carros no RN entre 2012 e 2016.



**Figura 3.** Número de habitantes no RN entre 2012 e 2016.



**Figura 4.** Taxa de crescimento (%) do número de automóveis e habitantes no RN.

A Figura 4 mostra as diferenças nas taxas de crescimento do número de carros e habitantes. Percebe-se que em todos os cinco anos para os quais o levantamento foi feito, a taxa de crescimento da frota de automóveis é maior que a taxa de crescimento da população. No último registro, de 2015 para 2016, o Rio Grande do Norte teve um **crescimento populacional de 0,95%** enquanto a **frota de carros aumentou em 4,31%**. Isso representa um acréscimo de 21.095 carros à frota do estado no intervalo de um ano.

### 2.3. Caronas e o futuro da mobilidade

A partir do aumento na frota de veículos em diversas regiões do mundo e a recente demanda por serviços de mobilidade, a empresa de auditoria Deloitte disponibilizou um estudo, intitulado “*The future of mobility: How transportation technology and social trends are creating a new business ecosystem*”, que mostra como tecnologias para transportes e tendências sociais estão mudando o mercado automobilístico e criando um novo ecossistema de negócios.

O estudo divide o futuro da mobilidade em quatro estados. Porém, apenas os dois primeiros estados serão analisadas nesse trabalho, uma vez que o terceiro e quarto estado dizem respeito a veículos autônomos particulares e compartilhados, respectivamente, sendo então uma tendência futura. Já os dois primeiros estados, que tratam de veículos dirigidos por um motorista, com ou sem assistência, são chamadas de: 1º) *Incremental change*; 2º) *A world of carsharing*.

No primeiro estado, a propriedade particular permanece predominante, com consumidores optando por opções particulares de flexibilidade, segurança, e conveniência ao possuírem seus veículos particulares. Neste estado são incorporadas tecnologias de assistência na direção como atrativos tecnológicos em carros e geralmente os consumidores assumem que carros totalmente autônomos não estarão disponíveis nem tão cedo.

O segundo estado antecipa o crescimento contínuo de acesso compartilhado à veículos. Nesse estado, a expansão do serviço de carros compartilhados cresce para novos territórios geográficos e segmentos

especializados de clientes. Aqui, os passageiros agregam mais valor a conveniência do transporte de um ponto a outro sem a preocupação de terem que dirigir e procurarem difíceis vagas de estacionamento em um trânsito caótico. Esse movimento auxilia no surgimento de serviços de caronas e compartilhamento de carros.

Segundo a pesquisa aplicada pela Deloitte, 43% dos participantes de sua da pesquisa que pertencem a geração Y e Z utilizam serviços de caronas compartilhadas, como o Uber e Cabify. E desses, 55% questionam a necessidade de possuírem um carro. Esse estudo mostra que caronas e compartilhamento de carros estão ganhando cada vez mais adeptos. A seção seguinte trata de uma pesquisa sobre caronas aplicada na UFRN, confirmando as tendências também no estado do Rio Grande do Norte.

## 2.4. Pesquisa sobre aplicativo de caronas na UFRN

Em meados do semestre 2016.1 foi elaborado e aplicado uma pesquisa em parceria com a Equipe de Design, Usabilidade e UX (experiência de usuário) da Superintendência de Informática da UFRN (SINFO) com a comunidade acadêmica da universidade. O objetivo dessa pesquisa era colher informações sobre o perfil dos usuários disposto a usar um aplicativo de caronas, bem como suas experiências em relação a caronas. Outro fator importante que motivou a pesquisa foi identificar o que mais importava para os usuários de um aplicativo com essa finalidade.

A pesquisa foi montada na forma de questionário e desenvolvida com a ferramenta *Formulários Google*. Essa ferramenta faz parte da suíte de aplicativos web do *Google Docs*. Com ela é possível: coletar respostas rapidamente, disponibilizando formulários online que são acessados por links públicos; criar perguntas personalizadas de múltipla escolha, grade de múltipla escolha, caixas de seleção, além de perguntas subjetivas curtas e longas. As respostas coletadas são armazenadas automaticamente e para elas são geradas planilhas e gráficos.

O questionário era composto de perguntas objetivas e subjetivas. Com as perguntas objetivas foi possível fazer uma análise quantitativa importante

para algumas tomadas de decisão no desenvolvimento do aplicativo. As questões subjetivas foram importantes para colher opiniões e sugestões acerca da ideia. As questões subjetivas também foram importantes para o processo de elaboração do nome do aplicativo, como demonstrado na seção 3.2.

Este questionário foi disponibilizado durante 24 horas, entre os dias 28 e 29 de fevereiro de 2016, no perfil oficial da SINFO no Facebook. Alguns usuários da rede social ainda expressaram suas opiniões na própria postagem na rede, como pode ser vista na Figura 5. Durante este período **735 pessoas** da comunidade acadêmica da UFRN responderam o questionário. O expressivo número de respostas para um questionário aberto por apenas um dia reforça a importância do tema pelo número de pessoas engajadas que se propuseram a responder.



**Figura 5.** Postagem realizada para o questionário no perfil da SINFO no Facebook.

A seção 2.4.1 apresenta as perguntas do questionário e o porquê delas, enquanto a seção 2.4.2 contém uma análise das respostas coletadas nesse período.

### 2.4.1. Perguntas do questionário

O questionário foi composto de nove questões sendo duas de múltipla escolha, duas de grade de múltipla escolha, duas de caixa de seleção e três subjetivas. Estas questões foram as seguintes:

1. Vínculo com a universidade:
  - a. Estudante de graduação;
  - b. Estudante de pós-graduação;
  - c. Servidor técnico-administrativo;
  - d. Docente.
2. Gênero:
  - a. Masculino;
  - b. Feminino;
  - c. Outro.
3. Turnos que mais se desloca para o campus;
4. Você já recebeu ou ofereceu carona para a UFRN?
  - a. Sim, recebi;
  - b. Sim, ofereci;
  - c. Não.
5. Se sua resposta para a questão anterior foi “sim”, por favor descreva sua experiência. Se “não”, explique quais foram os impeditivos;
6. Indique a importância dos aspectos abaixo em um serviço de oferecer e receber caronas na UFRN;
7. Em que tipo de carona você estaria interessado?
  - a. Dentro do campus (ex.: entre departamentos, setores de aulas, centros, etc.);
  - b. Do campus para suas imediações ou vice-versa (ex.: Shopping Via Direta);
  - c. Do campus para pontos de referência, ou vice-versa (ex.: shoppings, comércios, etc.);
  - d. Do campus para bairros específicos ou vice-versa.
8. Espaço para sugestões;

9. Você gostaria de ser informado sobre novidades e informações desse projeto? Se sim, informe um e-mail para contato.

A questão 1 tem como objetivo uma análise estatística de quais membros da comunidade acadêmica estão interessados no aplicativo. A questão 2, busca analisar o gênero dos interessados. Tanto a questão 1 quanto a questão 2 foram de múltipla escolha e só permitem uma resposta. As questões três e seis foram do tipo grade de múltipla escolha. Nesse tipo de pergunta são apresentados tópicos e cada um deles possuem múltiplas escolhas, resultando numa resposta em forma matricial de linhas e colunas.

Na questão 3 foi perguntado em que turno a pessoa mais se desloca para o campus e para cada um dos tópicos manhã, tarde e noite, deveria ser escolhido uma das opções: nunca, raramente, regularmente ou frequentemente. Já na questão 6 existia para cada tópico uma série de 1 a 5, sendo um menos importante e cinco mais importante. Os tópicos da questão 6 que as pessoas deveriam avaliar sua importância eram: segurança; praticidade; conforto; otimização de tempo e interação com novas pessoas.

Caixas de seleção foram usadas para as questões quatro e sete. Nesse tipo de questão uma ou mais opções podem ser escolhidas. Desse modo, a questão 4 perguntava se o entrevistado já tinha recebido ou oferecido carona para a universidade. A questão 7 perguntava o sobre o tipo de carona que mais interessava o entrevistado. Dentre os tipos era possível escolher caronas: dentro do campus; do campus para suas imediações; do campus para pontos de referência e do campus para bairros específicos.

O entrevistado tinha na questão 5 a oportunidade de explicar como foi a experiência de oferecer ou receber caronas para a universidade, ou de elencar impeditivos que fizeram com que nunca oferecesse ou recebesse caronas. Esta questão era subjetiva.

Outra questão subjetiva importante era a oitava, nela tinha apenas um espaço para opiniões e sugestões sobre o projeto. Nela os entrevistados eram livres para escrever suas opiniões e esta questão serviu para analisar e escolher palavras chaves que foram utilizadas no processo de nomeação do

aplicativo, conforme descrito na seção 3.2.

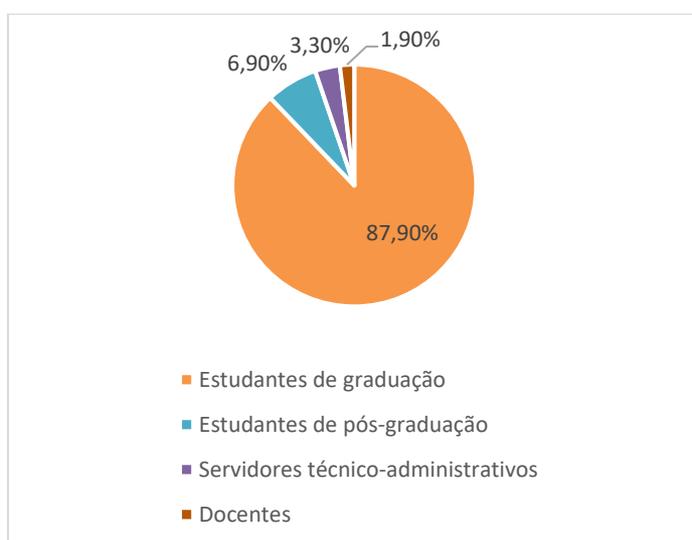
A oitava e nona questões foram subjetivas e facultativas, ou seja, os entrevistados não foram obrigados a responde-las. Na nona questão, pedia um endereço de e-mail para que as pessoas fossem atualizadas sobre o projeto, se assim desejassem. Os resultados obtidos com a aplicação desse questionário serão explanados na sessão a seguir.

#### 2.4.2. Análise das respostas

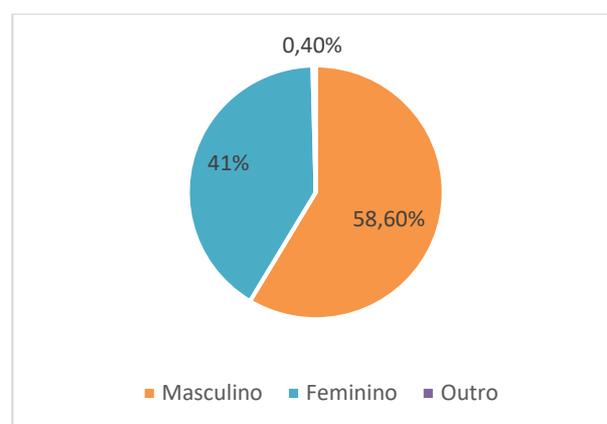
Todos os gráficos dessa sessão foram produzidos a partir dos resultados do questionário em análise das 735 respostas obtidas.

**Questão 1:** sobre o vínculo com a universidade. Exatamente 646 (87,9%) respostas foram de estudantes de graduação. Ainda responderam 51 (6,9%) estudantes de pós-graduação responderam, seguidos de 24 servidores técnico-administrativos e 14 docentes (3,3% e 1,9%, respectivamente). Esse resumo pode ser visto na Figura 6.

**Questão 2:** sobre o gênero do entrevistado. A Figura 7 contém o resumo do resultado da pesquisa, apontando que 431 (58,6%) entrevistados são do sexo masculino, em seguida, 301 (48%) entrevistadas do sexo feminino e outras 3 (0,4%) optaram por outro gênero.



**Figura 6.** Divisão dos tipos de vínculos nas respostas do questionário.

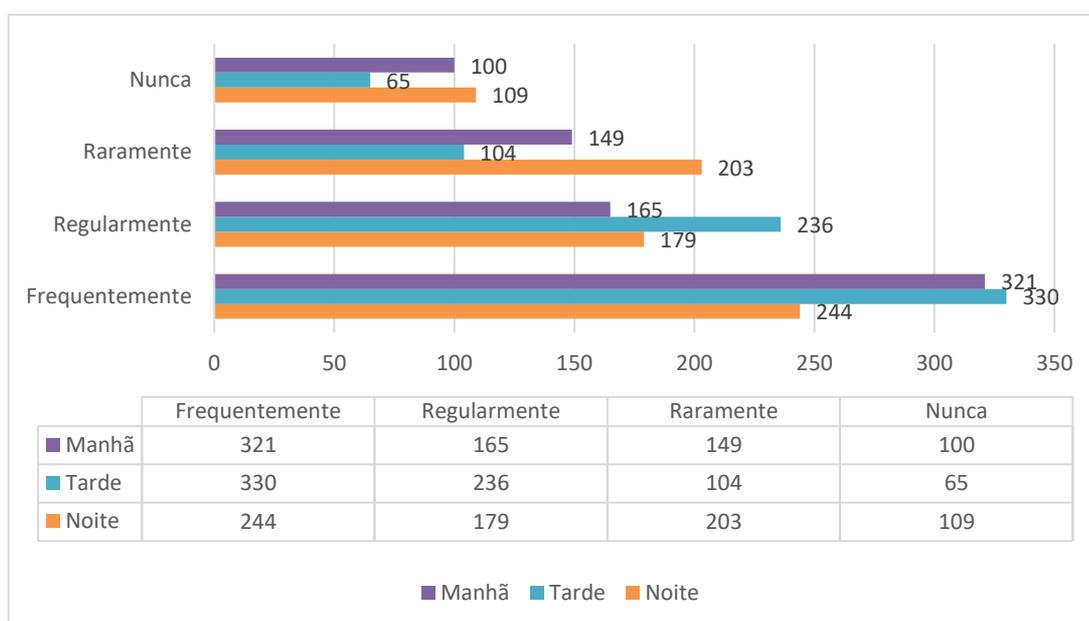


**Figura 7.** Divisão dos gêneros no questionário.

**Questão 3:** dos turnos que mais se deslocam para o campus central.

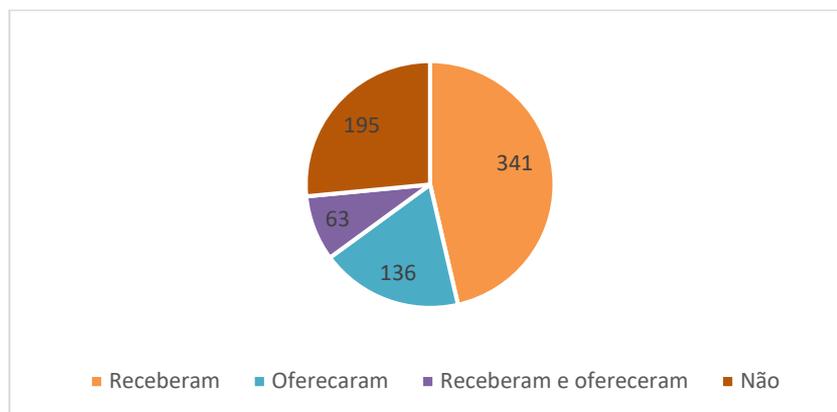
Essa questão foi importante para analisar estatisticamente os turnos que as pessoas mais se deslocam para UFRN. É possível visualizar, a partir do gráfico da Figura 8, que frequentemente vão mais pessoas para a universidade no turno da tarde, sem seguida no turno da manhã.

De acordo com as respostas, o fato de uma quantidade maior de pessoas ter selecionado “raramente” para o turno noturno, mostra que os maiores fluxos de pessoas na universidade são no turno matutino e vespertino.



**Figura 8.** Frequência de deslocamento para o campus nos turnos de aula.

**Questão 4:** sobre já ter oferecido ou recebido carona. Com base no resultado apresentado na Figura 9, pode-se somar a quantidade de pessoas que já receberam e/ou solicitaram caronas. Assim, 540 dos entrevistados já estiveram engajados em uma carona. Essa soma representa 73,4% dos entrevistados. Apenas aproximadamente 25% dos entrevistados não participaram de caronas.



**Figura 9.** Divisão dos entrevistados que já receberam e/ou ofereceram caronas ou não.

**Questão 5:** sobre a experiência de receber ou oferecer caronas. Essa questão solicitava que os entrevistados escrevessem como foi a experiência de receber ou oferecer carona. Para aqueles que não tinham recebido ou oferecido ainda, pedia-se que escrevessem os motivos que os impediram. Alguns dos relatos foram os seguintes:

- *“Sim, mas de amigos bem próximos!”*
- *“Carona com amigos”*
- *“Experiência de solidariedade”*
- *“Não sei se aceitariam”*
- *“Tenho medo”*
- *“Questões de segurança”*
- *“Sim, recebi carona algumas vezes, o que facilita e agiliza bastante o deslocamento. Principalmente em relação à espera do circular do campus”*
- *“Uma grande praticidade, não precisei pegar o circular”*
- *“Professores já me ofereceram carona ao passarem pela parada. Ocorreu de uma carona de uma pessoa que passava me salvar de perder uma prova, pois era fim de semana e estava sem circular. No entanto, quando são pessoas que não conhecemos, ficamos com um pé atrás.”*
- *“Já recebi e ofereci carona. O maior impedimento é o fato da maioria das vezes não conhecermos a pessoa, o que gera receio de aceitar/oferecer carona.”*

- *“Normal e prático, a carona foi oferecida por um servidor. Muito educado, por sinal.”*
- *“Muitos têm medo por não conhecer a pessoa que está oferecendo carona. Mas quando é colega de sala, aí a carona é tranquila”*
- *“Foi bastante cômodo, pois estudo a noite e voltar pra casa é muito perigoso. Teve um que me deixou na porta de casa.”*
- *“Eu acho super valido dar caronas dentro do campus. Mas nem sempre vejo uma forma boa de oferecer caronas”*

Além desses relatos, muitos outros foram apresentados mostrando pontos positivos e negativos de oferecer ou receber caronas. Muitos citam a praticidade de não ter que usar o transporte público circular da universidade, que julgam demorado ou superlotado. Alguns alunos que estudam à noite apontaram a comodidade de receber carona na volta para casa, pois julgam perigoso o retorno.

Para aqueles que nunca receberam ou ofereceram caronas, os impeditivos mais recorrentes se relacionam com questões de segurança. Por medo, receio ou desconfiança em pessoas desconhecidas, deixam de utilizar caronas. Alguns relatam que não viria problemas se a carona fosse com colegas do curso ou conhecidos.

A questão cinco trouxe à tona um fato relevante. Devido a cultura nacional e problemas de segurança, caronas só são uma opção de meio de transporte confiável para muitos indivíduos se esse conhecer as outras pessoas envolvidas na carona, ou se tiver dessas pessoas o mínimo de informações necessárias para se sentir seguro em oferecer ou receber carona com terceiros.

Ainda a respeito das respostas da questão cinco, uma pessoa respondeu que nem sempre ver uma boa forma de oferecer carona, ou seja, existe o interesse em oferecer caronas, porém não sabe como fazê-lo.

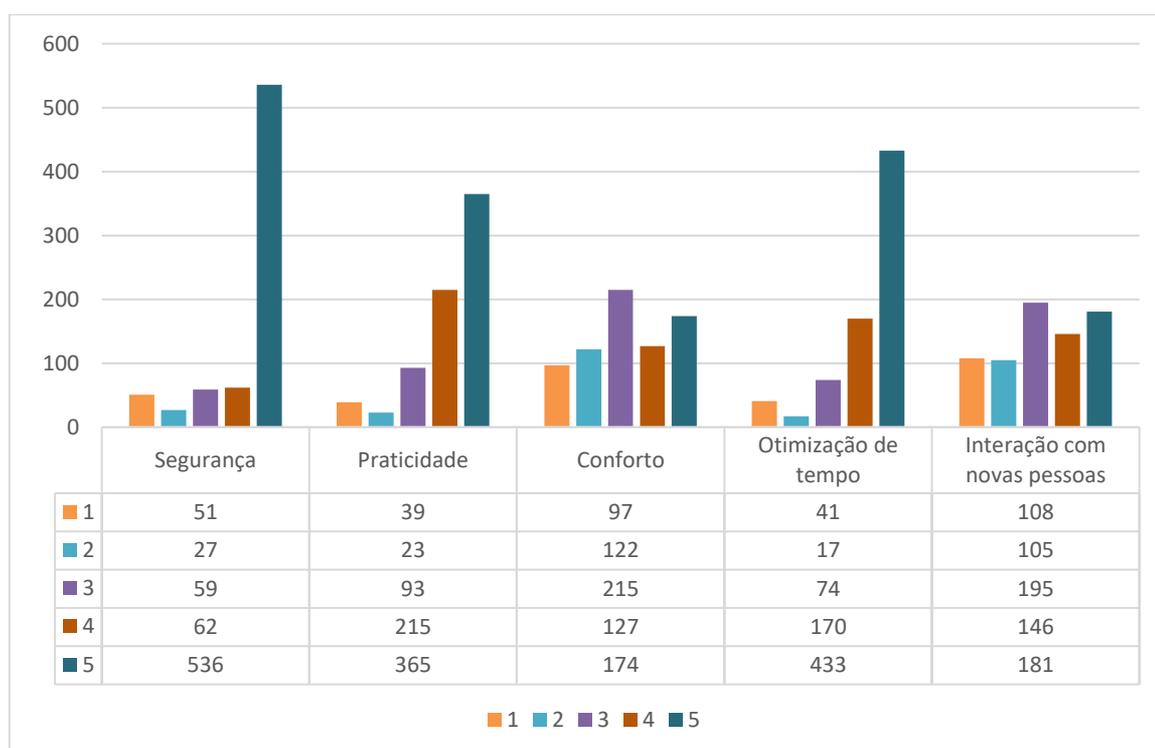
**Questão 6:** sobre a importância de alguns aspectos para um serviço de ofertar e receber caronas na UFRN. A Figura 10 mostra o gráfico com os resultados das respostas para essa questão. Foi pedido que os entrevistados

julgassem os pontos nele apresentados de um a cinco, sendo o valor cinco o mais importante.

Bem como na questão cinco, o aspecto “segurança” teve destaque. De modo que 72,9% dos entrevistados, ou seja, 536 pessoas deram nota máxima de cinco no quesito segurança. Nenhum dos outros aspectos analisados tiveram uma margem tão expressiva em um único valor.

Esse aspecto é primordial tanto para as pessoas que desejam oferecer caronas ou para aquelas que desejam solicitar caronas. Reforçando a ideia de que um aplicativo de caronas deve ser confiável e seguro ao usuário. Isso pode definir o sucesso ou fracasso do aplicativo.

A praticidade é outra qualidade que beneficia tanto os usuários que desejam receber caronas quanto aos que desejam oferecer. Um total de 365 pessoas julgaram esse fator importante com valor cinco. Assim, notavelmente as pessoas desejam que o serviço de caronas seja prático em seu uso.



**Figura 10.** Escala de importância de alguns aspectos para um serviço de caronas.

O fator conforto foi o mais bem distribuído. A maioria, ou seja, 215 pessoas deram valor três para ele, em seguida o valor cinco foi elegido para

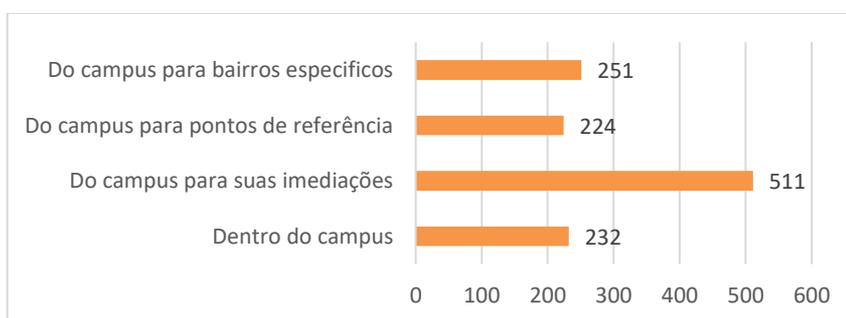
174 pessoas. A diferença foi de apenas cinco pessoas entre os valores dois e quatro para esse aspecto.

Conforto é um aspecto mais importante para as pessoas que desejam solicitar caronas, por vezes pela comodidade de não utilizar transportes públicos, como relataram algumas pessoas na questão cinco. Em linhas gerais, pela distribuição de importância apresentada, conforto é desejável, porém não é indispensável como outros aspectos.

A otimização de tempo, assim como segurança, foi um dos aspectos que obtiveram maior concentração no valor cinco. Foram 433 pessoas que julgaram otimização de tempo um fator de grande importância. Vale analisar que a otimização de tempo é mais importante para aqueles usuários que desejam solicitar caronas, visto que quem possui veículos já contam com estes meios de transporte a disposição.

O último aspecto analisado nessa questão pelos entrevistados foi a interação com novas pessoas. Assim como o conforto, o valor mais votado nesse aspecto foi o três, com 195 pessoas. Em seguida, por uma diferença mínima, o valor cinco aparece com 181 pessoas escolhendo ele. Isso demonstra que, de forma bem distribuída, ou as pessoas sentem indiferença com o fato de interação em caronas ou acham este um fator importante.

**Questão 7:** sobre o tipo de caronas que as pessoas estariam interessadas. O gráfico da Figura 11 expressa as respostas obtidas. Vale lembrar que essa questão permitiu que as pessoas escolhessem uma ou mais opções, portanto o somatório dos valores do gráfico ultrapassa o número de 735 respostas.



**Figura 11.** Tipos de caronas que as pessoas mais se interessam.

Ainda assim, o tipo de carona que as pessoas mais se interessaram foi para as proximidades do campus, ou seja, pontos como o Via Direta ou outros destinos do transporte circular. Desse modo, pode-se inferir que a maioria das pessoas que poderiam vir a utilizar um aplicativo de carona, buscam uma alternativa ao transporte público oferecido.

**Questão 8:** espaço para sugestões. Essa questão teve sua importância em vários momentos. Com ela alguns requisitos foram levantados e serão descritos na sessão 4.2. Também foi utilizada no processo de nomeação do aplicativo, conforme a seção 3.1.

A seguir são apresentadas algumas sugestões que merecem destaque pela importância que trouxeram ao Vemcar, são elas:

- *“Seria interessante no aplicativo que dê pra ver o perfil da pessoa que está oferecendo ou pedindo carona.”*
- *“Poderia ser feito um selo que colocasse nos carros para mostrar que o motorista é aluno também. Esse selo seria distribuído pelos centros acadêmicos depois da identificação da pessoa como aluno.”*
- *“Seria interessante um sistema de avaliação e comentários, tanto pra quem oferece carona quanto pra quem pega, onde se avaliaria as condições do carro, se o motorista desrespeitou as regras de trânsito, se houve algum tipo de assédio, se a pessoa que pegou carona furtou algo do veículo, etc.”*
- *“Se realmente vir a se concretizar, seria bom um mecanismo que assegurasse que os que vão dar carona são realmente vinculados à UFRN”*
- *“Esperamos que tenha segurança”*
- *“Essencial que se preze pela segurança de quem oferece e recebe carona. O aplicativo poderia restringir os usuários a estudantes e servidores da UFRN, além de exigir uma foto no perfil de cada usuário do aplicativo.”*
- *“Acho de grande importância a criação do aplicativo no qual as pessoas que dão carona possam se inscrever a partir da matrícula com a UFRN.”*
- *“Seria interessante ter a opção de cancelar uma carona caso aconteça um imprevisto.”*

- “Seria bom também um chat onde as pessoas interessadas em dar e pegar carona pudessem marcar onde se encontrar, etc.”
- “Grupo de chat temporário das pessoas do veículo”
- “(...) seria interessante principalmente para as mulheres terem cuidado com quem oferece a carona.”
- “Acredito que seria importante uma opção onde as mulheres filtrassem os resultados de caronas somente entre mulheres.”
- “Seria legal um tipo de carona de mulheres para mulheres, para garantir mais segurança para as meninas que usarem esse serviço, principalmente a noite”
- “O mais importante é se preocupar com a segurança, principalmente de mulheres.”
- “Isso deveria entrar em vigor o mais rápido possível, principalmente no campus da Unidade Acadêmica Especializada em Ciências Agrárias na EAJ. Lá, depender de caronas é quase que um pré-requisito para se estudar, infelizmente.”

Os interessados no aplicativo deixaram suas opiniões nesta questão e muitas dessas opiniões foram levadas em conta. Na lista de opiniões apresentada anteriormente alguns pontos foram sublinhados destacando sua importância. Um dos pedidos mais recorrentes é para que o aplicativo forneça um perfil com foto do usuário e mais do que apenas isso, é muito desejável que o aplicativo garanta que os usuários tenham vínculo ativo com a universidade.

Quanto as restrições de usuários para apenas alunos e servidores, sugeriram que o cadastro dos usuários fosse validado por meio da matrícula do SIGAA (Sistema Integrado de Gestão de Atividades Acadêmicas). Uma funcionalidade também desejada é que o aplicativo permita que o usuário possa cancelar uma carona oferecida ou solicitada caso assim deseje. Também foi pedido que houvesse um chat temporário para as pessoas que estivessem envolvidas em uma carona.

Devido aos perigos e vulnerabilidades para os quais as mulheres estão constantemente ameaçadas na nossa sociedade, houve por volta de sete

pedidos para que tivesse no aplicativo um filtro ou alguma forma de garantir que mulheres pedissem ou oferecessem caronas apenas para outras mulheres. Isso é um importante mecanismo de segurança que ajuda a prevenir crimes de assédios. Este é um mal ainda presente na sociedade brasileira.

**Questão 9:** sobre manter informado a respeito do projeto. Nessa questão os entrevistados poderiam deixar um e-mail de contato para se manter informado acerca do aplicativo. Aqui, 199 pessoas deixaram seus e-mails para contato posterior.

Os e-mails recolhidos aqui foram utilizados outro momento quando o aplicativo estava prestes a ser lançado. Estes e-mails foram convidados para serem usuários em primeira mão da versão beta do aplicativo.

## 3. Fundamentação teórica

Este capítulo tem a finalidade de apresentar alguns conceitos que o leitor possa se interessar durante a leitura do restante do trabalho. Serão abordados alguns componentes do Android, serviços web REST, projeto OffDroid, dentre outros assuntos relevantes neste trabalho.

### 3.1. Componentes e linha de design do Android

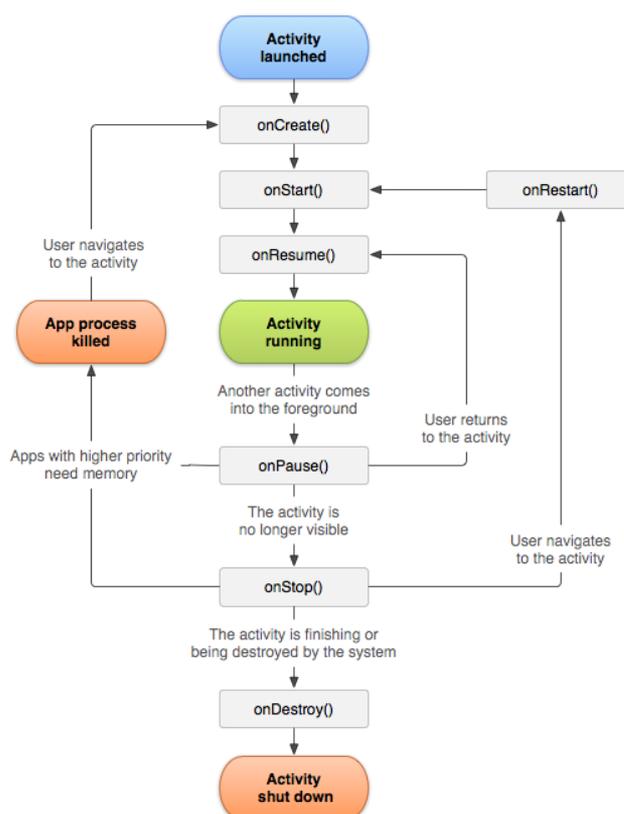
Por este trabalho tratar de um projeto desenvolvido para a plataforma Android, é interessante que o leitor conheça ao menos alguns componentes desse sistema. Por isso, a seguir serão descritos os componentes: *Activity*, *Fragment* e *Design Support Library*.

#### 3.1.1. Activity e Fragment

*Activity* (ou Atividade, em português) é, segundo o site oficial do Google para referências em Android (DEVELOPER, 2016), um componente de aplicativo que fornece uma tela com a qual os usuários podem interagir para fazer algo, como discar um número de telefone, tirar uma foto, enviar um e-mail ou ver um mapa. Cada *activity* recebe uma janela que exibe a interface de usuário. Geralmente a janela preenche a tela, mas pode ser menor que a tela e flutuar sobre outras janelas.

Aplicativos geralmente têm várias *activitys* pouco vinculadas entre si. Cada atividade pode, então, iniciar outra atividade para executar diferentes ações. Ao iniciar uma nova atividade, a atividade anterior é interrompida, mas o sistema conserva a atividade em uma pilha (a “pilha de retorno”). A pilha de retorno segue o mecanismo básico de uma pilha LIFO (*last in, first out*), ou seja, último a entrar, primeiro a sair. Assim, quando o usuário termina a atividade atual e aperta o botão *voltar*, ela sairá da pilha (é destruída) e a

atividade anterior será retomada. A Figura 12 a seguir representa o ciclo de vida de uma atividade.



**Figura 12.** Ciclo de vida de uma Activity (DEVELOPER, 2016).

Segundo (DEVELOPER, 2016), um *Fragment* (Fragmento, em português) é uma parte da interface de usuário de uma aplicação ou comportamento que pode ser introduzido em uma *activity*. A interação com um fragmento é feita através da classe *FragmentManager*, que pode ser obtida pelo método “*.getFragmentManager()*” de uma atividade.

A classe *Fragment* pode ser utilizada de diversas formas para se atingir uma ampla variedade de resultados. Apesar do fragmento apresentar um ciclo de vida próprio, ele depende do ciclo de vida de uma atividade. Desse modo, se uma atividade parar, nenhum fragmento dentro dela pode ser inicializado e quando uma atividade é destruída, todos os fragmentos dentro dela também serão destruídos. Nesse sentido, fragmentos podem ser utilizados para fornecer comportamentos diferenciados dentro de uma mesma atividade.

### 3.1.2. Design Support Library

Uma variedade de componentes do *Material Design* está disponível como parte do framework Android. A biblioteca Design Support provê uma API para fornecer importantes componentes adicionais do material design. Com essa biblioteca é possível, por exemplo, criar *Floating Action Button* e *Navigation Drawer*, como demonstrados na Figura 13.



**Figura 13.** Navigation Drawer e FAB (DEVELOPER, 2016).

Para utilizar a biblioteca Design Support, deve-se adicionar a seguinte dependência do *Gradle* como modulo da aplicação:

```
compile 'com.android.support:design:25.3.1'
```

### 3.1.3. Material Design

Durante todo o processo de prototipação e implementação da interface de usuário do Vemcar, foram levadas em consideração a linguagem visual introduzidas em 2014 pelo Google durante sua conferência anual de desenvolvimento, o Google I/O. Essa linguagem, chamada de *Material Design*, incorpora princípios de design para interfaces de usuário de múltiplos dispositivos, sejam os vestíveis, smartphones, tablets, desktops e TVs. Com ela, a empresa norte americana pretende prover movimentos fluidos em superfícies sensíveis ao toque, sem deixar de dar suporte a dispositivos baseados em teclado e mouse.

Estabelecendo princípios para animações, estilos e layouts, o Material Design expressa recomendações para um grande número de componentes visuais, tais como botões, cartões (cards), grades (grids), caixas de diálogo (dialogs), etc. Também relaciona ações e gestos, além de incluir um guia de acessibilidade.

Segundo o próprio Google, eles “desafiaram eles mesmo para criar uma linguagem visual para seus usuários que sintetize os princípios clássicos de um bom design com a inovação e a possibilidade de tecnologia e ciência (...)” (GOOGLE, 2017).

## 3.2. Serviços REST

O termo REST representa a sigla *Representational State Transfer*, que trata de um estilo arquitetural construído para aplicações web e para se desenvolver um sistema seguindo esse estilo, algumas restrições devem ser atendidas. De acordo com (PLANSKY, 2014), um sistema REST deve atender as cinco restrições a seguir.

A primeira delas é uma restrição básica do REST. A arquitetura deve ser dividida no paradigma cliente-servidor. Assim, o cliente, considerado um consumidor do serviço, não precisa se preocupar com tarefas que envolvem comunicação com banco de dados, gerenciamento de cache, etc. E o servidor, por sua vez, é um provedor de serviço e não implementa e nem se preocupa com interface e experiência de usuário, por exemplo. Isso permite uma evolução independente das duas arquiteturas.

O mesmo cliente pode enviar várias requisições simultâneas ao servidor, porém, pela propriedade *stateless*, cada uma delas devem ser independentes. Desse modo, toda requisição feita por um cliente deve conter todas as informações necessárias para que o servidor consiga entendê-la e processá-la.

Pelo fato de que muitos clientes podem solicitar o mesmo recurso ao servidor, as respostas do servidor devem serem mantidas em cache, evitando processamento desnecessário. A essa qualidade dá-se o nome de *cacheable*. Isso significa que um primeiro cliente solicita um determinado recurso para o

servidor, então o servidor processa essa requisição e armazena temporariamente isso em cache. Quando os demais clientes solicitam o mesmo recurso, o servidor responde com o que está salvo em cache, sem ter que processar a informação novamente.

Um quarto requisito para um sistema REST é chamado de *uniform interface*. Basicamente, trata-se de um contrato para comunicação entre cliente e servidor. São pequenas regras que visam deixar um componente o mais genérico possível, tornando-o mais fácil de ser refatorado e melhorado.

Por fim, a aplicação deve ser composta por camadas, sendo fáceis de alterar, adicionar e remover camadas. Essa restrição sobre camadas se chama *layred system* e um dos princípios é que o cliente nunca deve chamar diretamente o servidor da aplicação sem antes passar por um intermediador que faça a interface com o servidor. Isso garante que o cliente se preocupe apenas com a comunicação com o intermediador, ficando este segundo responsável por distribuir as requisições no servidor.

Um sistema é dito *RESTful* se atender as cinco restrições descritas anteriormente. Sendo elas:

- Cliente-servidor;
- *Stateless*;
- *Cacheable*;
- *Uniform interface*;
- *Layred system*.

Por vezes, quando se usa diferentes URIs, verbos HTTP e diferentes formatos de retorno, costumam dizer que o sistema em questão é *RESTful*. Apesar desses pontos serem importantes, eles sozinhos não garantem que um sistema seja considerado *RESTful*. É necessário obrigatoriamente que ele atenda os cinco princípios descritos anteriormente para ser considerado um sistema assim.

### 3.3. OffDroid

O framework *OffDroid* foi proposto e implementado por Jean Guerethes (GUERETHES, 2015) para fornecer um conjunto de funções de persistência de dados, replicação e de sincronização, possibilitando o funcionamento de um sistema no modo *off-line*.

Três estratégias foram definidas para a manipulação das classes de domínio, na primeira estratégia a anotação *@OnlyOnline* deve ser utilizada. Essa estratégia deve ser aplicada na classe de domínio quando não se julgar necessário realizar persistência da classe para o banco de dados do dispositivo móvel. Neste caso, a classe não apresentará representatividade no banco de dados do dispositivo móvel e esse objeto só irá ser preenchido quando o dispositivo estiver conectado.

Uma segunda estratégia, contrária a anterior, faz com que a classe de domínio apresente representatividade somente no banco de dados local do dispositivo móvel. Essa classe não deverá ser sincronizada e nem enviada para o servidor da aplicação. Nesse caso, deve-se ser utilizada a anotação *@OnlyLocalStorage*.

A terceira estratégia é aplicada quando uma classe requer que seus dados sejam armazenados tanto no banco de dados local, quanto persistidas no servidor. Por se tratar de uma condição padrão adotada por muitos desenvolvedores em suas classes de domínio, não se faz necessário utilizar nenhuma anotação para atender a essa estratégia.

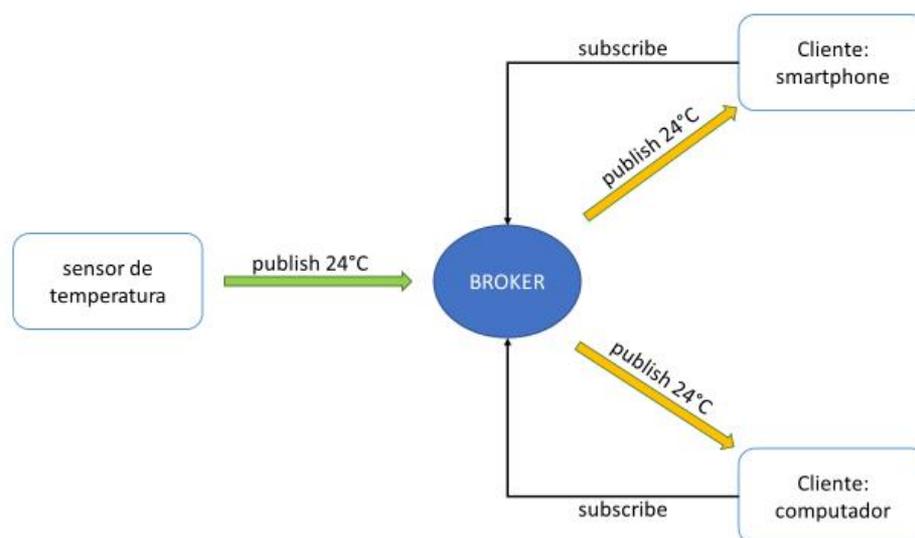
### 3.4. Padrão Publish/Subscribe

O padrão Publish/Subscribe (também conhecido como Pub/Sub) é uma alternativa para o tradicional modelo cliente-servidor, onde um cliente se comunica diretamente com um servidor. No paradigma Pub/Sub, temos os clientes desacoplados em dois tipos, primeiramente aqueles que estão enviando uma mensagem particular (chamados de *publishers*), os outros clientes são aqueles que estão recebendo mensagens (chamados de *subscribers*). De inglês, em uma tradução direta, teríamos os publicadores e

subscritores, ou seja, aqueles interessados em enviar mensagens e os interessados em receber as mensagens, respectivamente.

Nesse padrão, os *publishers* e *subscribers* não sabem da existência uns dos outros. Então existe um terceiro componente, chamado de *Broker*, que conhece tanto os que publicam quanto os subscritores. O papel do *Broker* é filtrar todas as mensagens recebidas e distribuir elas adequadamente.

A Figura 14 a seguir foi adaptada do site HiveMQ (HIVEMQ, 2017) e descreve este padrão. Nela, o *Publisher* “sensor de temperatura” pode publicar a temperatura de 24°C no tópico “temperatura”. Os dois clientes na figura podem ser *subscribers* no tópico temperatura, então o *Broker* encaminhará para eles a mensagem com a temperatura registrada pelo sensor.



**Figura 14.** Componentes do paradigma Pub/Sub.

Através da filtragem das mensagens é possível que apenas determinados clientes recebem determinadas mensagens. Como mencionado, o principal aspecto do Pub/Sub é o desacoplamento dos publicadores e subscritores, que podem ser diferenciados em algumas características.

- Desacoplamento espacial: publicadores e subscritores não precisam conhecer uns aos outros (por endereço IP e porta, por exemplo);

- Desacoplamento de tempo: publicadores e subscritores não precisam serem executados ao mesmo tempo;
- Desacoplamento de sincronização: as operações em ambos os componentes não são interrompidas durante a publicação ou recebimento.

Ainda segundo (HIVEMQ, 2017), se comparado com a abordagem tradicional de modelo cliente-servidor, ao adotar o Pub/Sub irá promover um ganho em escalabilidade. Isso se dá pelo fato de que as operações no *Broker* podem ser altamente paralelizadas e processadas em um aspecto dirigido a eventos. Também, muitas vezes o cache de mensagens e o encaminhamento inteligente das mensagens são decisivos para melhorar a escalabilidade.

### 3.5. Protocolo MQTT

O acrônimo MQTT deriva de Message Queuing Telemetry Transport. Segundo (MQTT, 2016), adota um paradigma publish/subscribe e é um protocolo de mensagens extremamente simples e leve. Foi projetado para dispositivos restritos a baixa largura de banda de rede, alta latência e redes não confiáveis.

O MQTT define um pacote de controle (Control Packets) para indicar a ação desejada a ser executada pelo recurso desejado. Os mesmos são os seguintes:

<b>Controle</b>	<b>Ação</b>
CONNECT	Cliente solicita uma ligação com um servidor
CONNACK	Reconhece solicitação de conexão
PUBLISH	Publicar mensagem
PUBACK	Reconhecimento de publicação
PUBREC	Publicação recebida. (QoS 2 Publicação recebida., part 1)
PUBREL	Publicação publicada. (QoS 2 Publicação recebida., part 2)
PUBCOMP	Publicação completada. (QoS 2 Publicação recebida., part 3)
SUBSCRIBE	Inscrever-se em um tópico
SUBACK	Reconhecimento de inscrição
UNSUBSCRIBE	Cancelamento de inscrição em um tópico
UNSUBACK	Reconhecimento de cancelamento de inscrição
PINGREQ	PING request

PINGRESP	PING response
DISCONNECT	Notificação de desconexão.

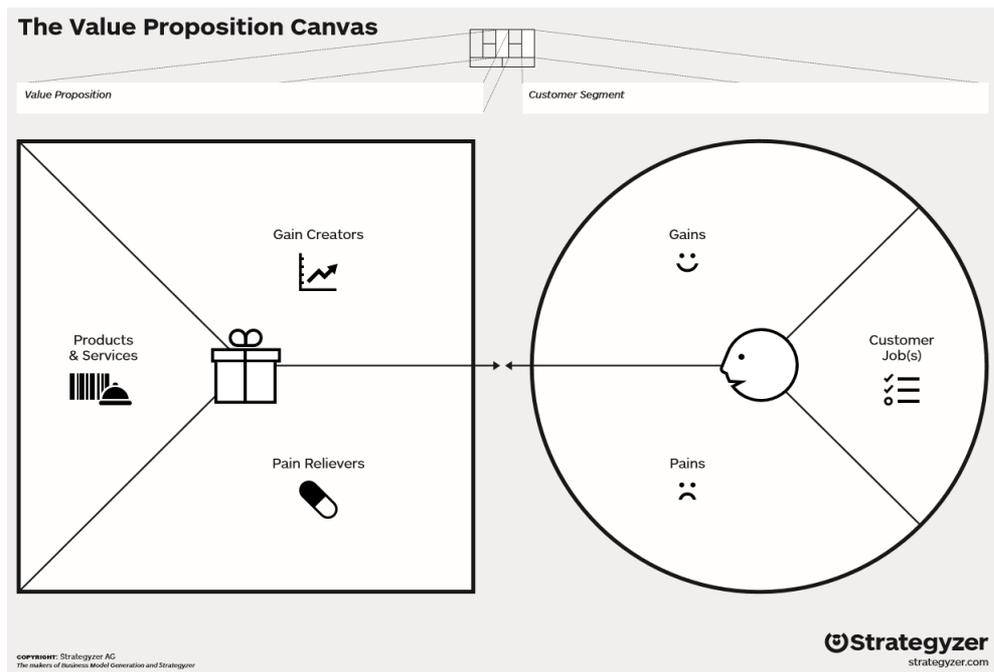
**Tabela 1.** Pacote de controle do MQTT.

### 3.6. Value Proposition Canvas (VPC)

Em julho de 2014, o Dr. Alexander Osterwalder, famoso cientista de negócios, publicou junto com seu sócio Alan Smith o livro *“Value Proposition Design: How to Create Services and Products Customers Want”* (OSTERWALDER, 2014). Neste livro introduziram o *Value Proposition Canvas* (VPC) como uma ferramenta útil para entender as necessidades dos usuários, desenvolvendo produtos e serviços do interesse deles.

Focado no segmento de clientes e na proposta de valor do produto, o VPC é dividido em duas dimensões, produto e cliente. A Figura 15 contém um modelo não preenchido, onde o lado esquerdo representa a dimensão do produto e o lado direito a dimensão do cliente. As duas dimensões são postas em encontro de modo que uma equipe encontre correspondências entre os atributos de ambas dimensões.

Do lado direito, existe as demandas que precisam serem atendidas para os clientes. As demandas são inseridas na seção “customer job(s)” do modelo. A partir dessas demandas, os clientes pretendem atingir alguns benefícios ou ganhos, que são representados na seção “gains” (ganhos). Porém, alguns impeditivos podem ocorrer, trazendo dificuldades ou frustrações para os clientes. Essas dificuldades e frustrações devem ser inseridas na seção “pains” do canvas.



**Figura 15.** Value Proposition Canvas (OSTERWALDER, 2014).

Já o lado esquerdo da Figura 15 apresenta a dimensão do produto no VPC, sendo dividida em três seções. Na seção “products & services”, são inseridas as características do produto que atendem as demandas dos clientes. Os geradores de benefício do produto, ou “gain creators”, são inseridos para atingir os ganhos dos clientes. Por fim, a seção “pain relievers” faz uma analogia aos analgésicos que tiram as dores (pains) do cliente, pois deve conter os aspectos do produto que confrontam as dificuldades e frustrações dos clientes.

## 4. Estado da prática

Serviços informatizados para auxiliar a mobilidade urbana surgem para oferecer a população alternativas criativas de transporte. Alguns desses serviços buscam, assim como o Vemcar, minimizar os problemas de mobilidade apresentados anteriormente. Grande parte destes serviços estão também disponíveis no Brasil e alguns aplicativos similares ao Vemcar foram encontrados na loja de aplicativos oficial do Android, o Google Play Store. Dois aplicativos de carona para universitários merecem destaque, são eles o *Caronaphone* e *Caronaê*.

Serão apresentados a seguir estes dois aplicativos, o Caronaphone e o Caronaê nas subseções 4.1 e 4.2, respectivamente. Será discutido os pontos positivos e negativos de cada um, além das semelhanças e diferenças com o Vemcar. Em seguida, a subseção 4.3 traz informações acerca de outros aplicativos populares que oferecem serviços de mobilidade. Por fim, um comparativo entre todos os aplicativos apresentados nesse capítulo é feito na seção 4.4.

### 4.1. Caronaphone

O Caronaphone conta com o apoio do Centro Interdisciplinar de Estudos em Transportes (CEFTRU) da Universidade de Brasília (UnB). Com o lema “*Carros cheios e ruas vazias*”, propõe uma opção sustentável de transporte por meio de caronas solidárias. A Figura 16 foi retirada do site oficial (CARONAPHONE, 2017) do Caronaphone e exibe a tela de detalhes de uma carona neste aplicativo.

Este aplicativo Android funciona conectado com o Facebook para a realização do login, compartilhando informações públicas (nome, fotos e

amigos em comum) dos usuários. Portanto, não é um aplicativo de uso restrito para a comunidade acadêmica da UnB.



**Figura 16.** Imagem promocional retirada do site do Caronaphone.

## 4.2. Caronaê

O Caronaê é um aplicativo feito exclusivamente para a comunidade acadêmica da Universidade Federal do Rio de Janeiro (UFRJ). Seu site oficial (CARONAÊ, 2017) diz que “a mudança de cultura é a chave para um mundo melhor”. Para isso, são destacadas em seu site as seguintes vantagens do Caronaê:

- *Segurança:* uso exclusivo para a comunidade acadêmica da UFRJ, evitando presença de usuários desconhecidos;
- *Tempo:* as caronas reduzem significativamente o tempo com a locomoção até a Cidade Universitária, sobrando tempo para o que realmente importa;
- *Centralização:* unifica as ofertas de carona em uma plataforma única, aumentando a oportunidade de viagens compartilhadas;
- *Meio ambiente:* aumentando a taxa média de ocupação de veículos, reduzimos o número de carros nas ruas e a emissão de poluentes;
- *Integração:* em caronas já surgiram grandes amizades, ideias mirabolantes e acordos milionários;
- *Pontos de caronas:* bem sinalizados, entre os prédios e os estacionamentos, eles facilitam o encontro na Cidade Universitária da UFRJ.

A Figura 17 apresenta uma imagem promocional do Caronaê e foi retirada do seu site oficial (CARONAÊ, 2017). Nela é possível perceber que o aplicativo está disponível nas duas principais lojas de aplicativos, a Google Play Store e a App Store, para usuários de smartphones Android e iOS, respectivamente.



**Figura 17.** Imagem promocional retirada do site do Caronaê.

### 4.3. Serviços de carona remunerada

Nesta seção foram separados outros serviços de mobilidade que utilizam o conceito de caronas e carros compartilhados. Aqui serão apresentados rapidamente o Uber e BlaBlaCar. Estes serviços foram postos em conjunto aqui, pelo fato de diferentemente dos já apresentados até então, não serem gratuitos ao usuário, sendo considerados serviços de “carona remunerada”.

Utilizando o aplicativo do Uber, usuários que precisam de transporte conseguem facilmente encontrar motoristas parceiros que oferecem esse serviço. Dentre os aplicativos avaliados, este é o único que oferece serviços diferenciados de conforto e segurança para o usuário que aceita pagar a mais por isso na modalidade UberBLACK, serviço que oferece carro sedã com bancos em couro, ar condicionado sempre ligado e motoristas “bem vestidos”.

Destes, o Uber é o aplicativo mais popular, por estar disponível em mais de 500 cidades ao redor do mundo. É também o aplicativo mais polêmico por causar conflitos aqui no Brasil com taxistas.

Já o BlaBlaCar tem como principal proposta que o motorista ofereça caronas de uma cidade para outra, encontre passageiros e assim reduza os

custos da viagem. As caronas são então remuneradas, porém com um valor mais acessível que o Uber, por exemplo. Para ter acesso ao BlaBlaCar o viajante deve ter a documentação confirmada.

#### 4.4. Comparativo

Os sistemas aqui apresentados têm muito em comum, pois todos promovem caronas, solidárias ou remuneradas, porém alguns possuem público alvo específico e outros não. A seguir, na Tabela 2, foi comparado os aplicativos Caronaphone, Caronaê, BlaBlaCar, Uber e Vemcar.

	<b><i>Caronaphone</i></b>	<b><i>Caronaê</i></b>	<b><i>BlaBlaCar</i></b>	<b><i>Uber</i></b>	<b><i>Vemcar</i></b>
<b><i>Oferece/solicita caronas</i></b>	Sim	Sim	Sim	Não	Sim
<b><i>Identificação de usuários</i></b>	Facebook	Facebook	CNH	CNH	SIGAA, SIGRH, SIPAC
<b><i>Restrição de público</i></b>	Livre	UFRJ	Livre	Livre	UFRN
<b><i>Localização no mapa</i></b>	Não	Não	Não	Sim	Sim
<b><i>Rotas entre usuários</i></b>	Não	Não	Não	Sim	Sim
<b><i>Chat entre usuários</i></b>	Sim	Não	Sim	Não	Sim
<b><i>Custo de viagem</i></b>	Gratuito	Gratuito	Pago	Pago	Gratuito
<b><i>Plataforma</i></b>	Android	Android, iOS	Android, iOS	And., iOS, WP	Android

**Tabela 2.** Comparativo entre aplicativos de serviço de mobilidade.

Foi levado em consideração aspectos como: a capacidade do aplicativo permitir que usuários ofereçam e solicitem caronas; mecanismo de identificação do usuário; restrição para álbum público alvo específico ou livre para qualquer pessoa; localização de usuários com mapa; rotas entre usuários como GPS; sistema próprio de comunicação entre usuários; custo de cada viagem, separados por gratuitos e pagos; plataforma disponível, sendo Android, iOS e Windows Phone.

Com base nas motivações que levaram ao projeto e desenvolvimento do Vemcar, ele se destaca dos demais aplicativos avaliados por ter um

mecanismo de identificação mais confiável se comparado a restrição de público alvo. Além de ser gratuito e oferecer integração com o mapa (Google Maps, no caso do Vemcar).

## 5. Proposta de solução

O Vemcar foi desenvolvido com o intuito de facilitar as ofertas e solicitações de caronas realizadas por membros da comunidade acadêmica dentro e fora da UFRN.

Este capítulo trata do surgimento da ideia do aplicativo, sua prototipação em sala de aula e aperfeiçoamento como produto de software com o auxílio da Superintendência de Informática da UFRN (SINFO). Além disso, serão apresentadas também as questões técnicas referentes as atividades do processo de software adotado, levantamento de requisitos, modelagem do sistema, definição da interface de usuário, arquitetura de software e as tecnologias utilizadas para o desenvolvimento do aplicativo.

### 5.1. Como surgiu

A partir de disciplinas de graduação, o Vemcar foi incorporado a suíte de aplicativos da SINFO. Esse processo de incorporação foi importante para o aplicativo que pode contar com uma equipe multidisciplinar em várias etapas do seu desenvolvimento.

As seções a seguir tratam do desenvolvimento de um protótipo em sala de aula, no decorrer de duas disciplinas e também o processo de desenvolvimento do aplicativo na SINFO.

#### 5.1.1. Sala de aula

No semestre 2015.2 foi ofertada a disciplina de Desenvolvimento de Sistemas para Dispositivos Móveis (código DIM0524), que é oferecida na UFRN pelo DIMAp para os cursos de Bacharelado em Engenharia de Software, Bacharelado em Ciência da Computação e Bacharelado em Tecnologia da Informação. Nesse semestre a disciplina foi ministrada pelos professores

Gibeon Soares de Aquino Junior e Itamir de Moraes Barroca Filho, foi proposto aos alunos dessa disciplina o desafio de desenvolver um protótipo funcional de um aplicativo para dispositivos Android.

Além disso, este desafio contava com outro requisito. Deveria ser desenvolvido um aplicativo com a temática “**vida no campus**”, ou seja, o produto desenvolvido deveria fornecer um serviço que ajudasse a comunidade acadêmica da UFRN de alguma forma. Então, a partir de uma pesquisa sobre problemas de mobilidade urbana e de experiências de alunos com o transporte circular da universidade, surgiu a ideia de um aplicativo de caronas.

O protótipo funcional que foi desenvolvido era um aplicativo que deveria conter os conceitos aprendidos em sala de aula. Os conceitos apresentados em DIM0524, conforme sua ementa, são: arquiteturas de software para sistemas de dispositivos móveis; padrões de projetos para estes sistemas; plataforma, linguagens de programação e ambientes de desenvolvimento. Os projetos desenvolvidos contavam como parte da nota final na disciplina e poderiam ser feitos em grupos de até três pessoas, porém o protótipo funcional do Vemcar foi desenvolvido individualmente.

O desenvolvimento do aplicativo na disciplina servia também como suporte prático para as teorias apresentadas. Aulas práticas eram oferecidas com frequência, auxiliando o aprendizado e desenvolvimento do protótipo com acompanhamento dos professores em cada projeto desenvolvido pelos alunos.

Outra particularidade desse semestre para a disciplina DIM0524 era a possibilidade de integração com a **API.sistemas** (APISISTEMAS, 2017), uma interface de programação de aplicações que disponibiliza dados dos sistemas da UFRN. Mais detalhes sobre essa tecnologia, seu funcionamento e benefícios serão descritos na seção 6.2.

Ainda no semestre 2015.2, o protótipo do Vemcar foi utilizado em outra disciplina que agregou valor ao seu desenvolvimento. Nesse caso, tratava-se da Tópicos Especiais em Engenharia de Software IV (código DIM0534). Esta disciplina foi ofertada pelo DIMAp como componente optativo para o curso de Bacharelado em Engenharia de Software. Ministrada pela professora Marcia

Jacynta Nunes Rodrigues Lucena, se tratava de uma disciplina sobre bancos de dados NoSQL.

DIM0534 tem o objetivo levar o aluno a construir e manipular bancos de dados não relacionais, além de entender conceitos relacionados com o tema e as diferenças entre bancos de dados relacionais e os NoSQL. Ao final da disciplina, cada dupla de alunos deveria apresentar um projeto de banco de dados. Então, com o auxílio do então aluno Jonathan Martins Barros Costa, foi projetado e implementado um banco de dados NoSQL para um servidor do protótipo do Vemcar. Várias tecnologias que suportam bancos de dados não relacionais foram vistas, porém no projeto se fez uso do MongoDB e Flask que não serão descritas nesse trabalho.

Além dessas ferramentas já citadas, vale destacar que durante esse semestre era utilizado o GitHub para repositório de código e controle de versão.

No período de férias entre os semestres 2015.2 e 2016.1, a convite do professor Gibeon, iniciou-se a fase de desenvolvimento do aplicativo de caronas na SINFO. Esse foi o início de uma etapa importante para o Vemcar e será descrita a seguir.

### 5.1.2. SINFO

Em meados de janeiro de 2016 se iniciou o desenvolvimento do aplicativo na SINFO. Na Superintendência o projeto não recomeçou do zero, pois a versão desenvolvida até então para as disciplinas foi mantida como base. Inicialmente, foram feitos levantamentos de requisitos em brainstorms com a equipe de desenvolvimento para dispositivos móveis (ou simplesmente equipe mobile). Assim foi possível identificar as necessidades do aplicativo e preparar uma arquitetura cliente-servidor.

Algumas outras equipes da SINFO deram suporte ao desenvolvimento do Vemcar. Estas equipes são formadas por colaboradores contratados e estagiários de diferentes cursos da universidade.

Um pedido de registro de patente foi feito no Instituto Nacional da Propriedade Industrial (INPI) sob o número “BR 51 2016 000943 8”, com os seguintes autores membros da SINFO:

<b>Autor</b>	<b>Função na SINFO</b>
André Grilo de Souza	Designer
Geraldo Lins de Oliveira Neto	Designer
Gibeon Soares de Aquino Junior	Diretor de sistemas
Jean Guerethes Fernandes Guedes	Analista de sistemas
Johnnylee Bryan Marques da Rocha	Desenvolvedor Android
Thiago de Lima Andrade	Desenvolvedor Android
Victor Jerônimo Galdino Lopes de Oliveira	Desenvolvedor Android

**Tabela 3.** Autores do Vemcar com registro de patente no INPI.

## 5.2. Scrum adaptado na SINFO

A equipe de desenvolvimento para dispositivos móveis da SINFO, liderada pelo analista de sistemas Jean Guerethes, trabalha com alterações do Scrum padrão. Essas alterações são válidas para todos os projetos da equipe, que é subdividida em equipe Android e iOS. Cada subdivisão da equipe conta com um colaborador contratado da SINFO que atua como líder da subequipe. Com essa divisão os projetos são divididos de acordo com a plataforma na qual será implementado.

Até o momento de produção desse trabalho, a equipe de desenvolvimento da SINFO desenvolvia os projetos mostrados na Tabela 4 conforme as plataformas.

<b>Nome do projeto</b>	<b>Android</b>	<b>iOS</b>
SIGAA Mobile	Sim	Sim
SigEventos	Sim	Sim
Olar	Sim	Sim
Vemcar	Sim	Não

**Tabela 4.** Projetos desenvolvidos na SINFO.

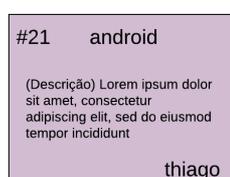
Para auxiliar no gerenciamento de requisitos dos *product backlogs* de cada projeto, existe um quadro de tarefas baseado em cartões onde os membros da equipe de desenvolvimento e testes inserem, removem e alteram os *post-its* (notas adesivas) para melhor visualizar em que estado a tarefa a se

encontra. Para isso, o quadro é dividido em estados conforme mostrado na ilustração da Figura 18.



**Figura 18.** Quadro de post-its da equipe mobile da SINFO com legenda de cores dos projetos

Uma nota adesiva no quadro representa uma tarefa. Um exemplo da estrutura de informações contidas em uma dessas notas adesivas é apresentado na Figura 19. Essas tarefas geralmente são *issues* (problemas) ou novas funcionalidades que serão desenvolvidas. O número de identificação da tarefa é o mesmo número gerado pelo *GitLab* para uma determinada *issue* cadastrada. Em seguida, se tem a plataforma referente a tarefa em questão, podendo ser Android ou iOS. No corpo da nota adesiva se encontra uma breve descrição da atividade e no canto inferior direito é escrito o nome do membro da equipe responsável pela atividade.



**Figura 19.** Exemplo de nota adesiva no quadro.

Uma tarefa de um determinado projeto inicia seu ciclo em uma *sprint* na seção “a fazer” do quadro. Quando algum membro da equipe inicia seu

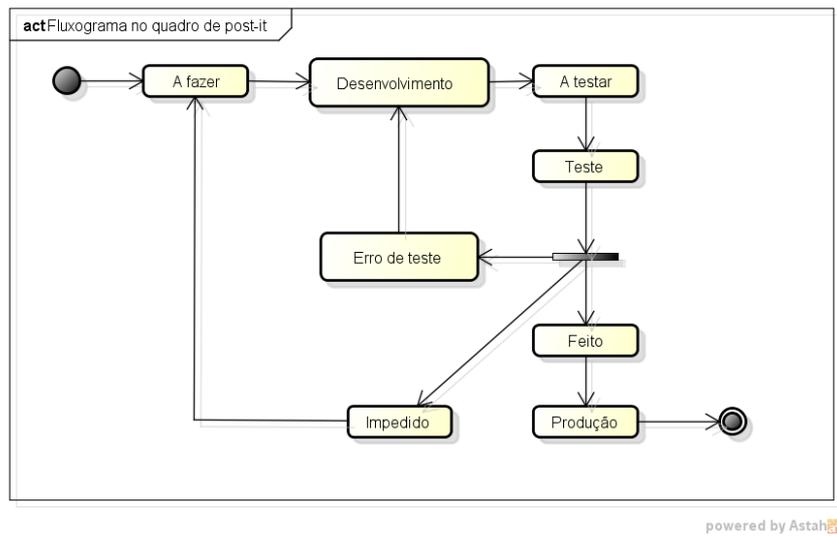
desenvolvimento, a nota deve seguir para a seção referente a isso no quadro. Quando uma tarefa estar no campo “a testar” significa que ela está na fila de testes do testador da equipe, seguindo para o campo “teste” do quadro quando essa atividade iniciar. Depois de testada, a atividade desenvolvida pode seguir para “erro de teste”, “feito” ou “impedido”. Veja na Figura 20 uma foto do quadro em um dado momento.



**Figura 20.** Foto do quadro de post-its da SINFO.

Um erro de teste faz com que a atividade volte para o status de desenvolvimento e o responsável por essa atividade é notificado. Quando uma atividade contém um impedimento, significa que ela depende de outra para ser concluída. Se validada, ela passa para o status “feito” e em seguida é posta em “produção”, onde já está disponível para o usuário. A Figura 21

representa um diagrama de atividades para uma atividade (nota adesiva) no quadro e representando também seu status.



**Figura 21.** Diagrama de atividades de uma tarefa no quadro.

### 5.3. Nome do aplicativo

Em sua fase de prototipação na disciplina de Desenvolvimento de Sistemas para Dispositivos Móveis, o aplicativo se chamava *Thumbsapp*. Este nome foi escolhido por ser uma palavra parônima, ou seja, sonoramente parecida, a “thumbs up”, que significa um gesto afirmativo com o polegar. Esse mesmo gesto está associado ao pedido de carona, que é amplamente difundido em todo mundo.

Todavia, não tinha sido feita nenhuma pesquisa ampla para a definição do nome *Thumbsapp*. Durante o período de desenvolvimento na SINFO, estudos mostraram que já existiam várias referências a este nome na internet e em outros aplicativos. Além disso, jugou-se que o nome atual não era sonoramente agradável de se pronunciar por falantes da língua portuguesa. Nesse momento, resolveu-se projetar um novo nome e desenho de marca para o aplicativo.

Nome e desenho de uma marca atuam de forma conjunta para entregar os valores propostos pela marca. O nome, no entanto, exerce ainda uma função institucional e jurídica, uma vez que ajuda a distinguir um serviço de outro similar e concorrentes. Assim, o nome de uma marca define substancialmente sua procedência e também a que público se destina (GRILO, 2016). *Naming* é um termo adotado por designers para a aplicação das técnicas de criação de nomes.

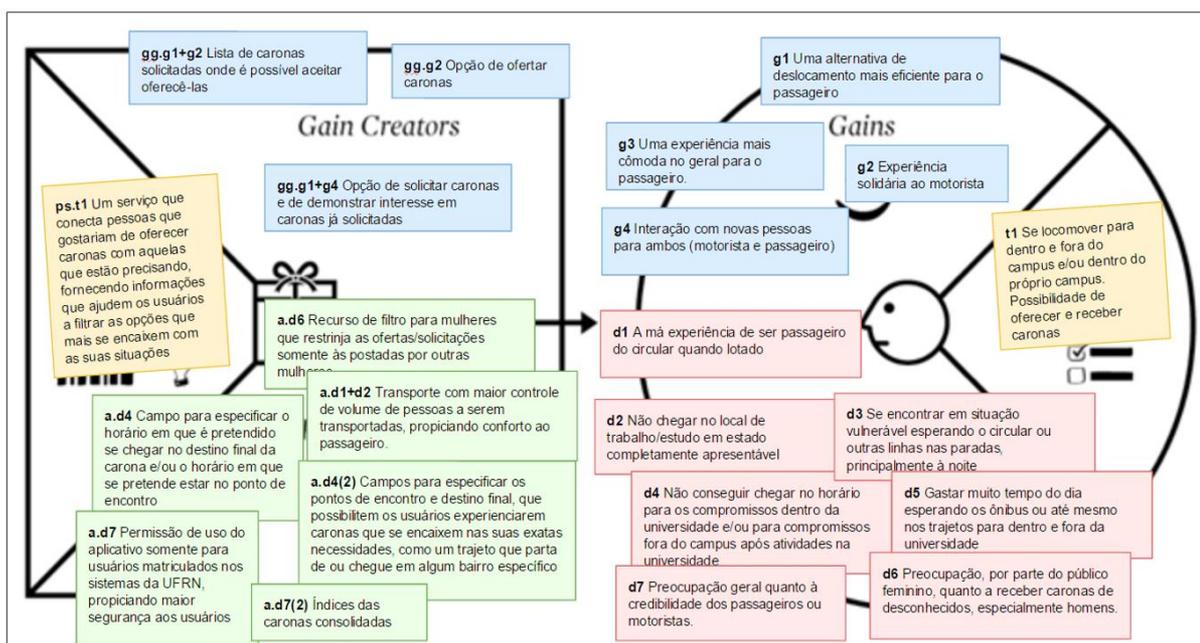
No ano de 2016 a equipe de design da SINFO aplicou técnicas de *naming* na construção da marca do aplicativo para consolidar o produto. Nesse processo o usuário foi o principal direcionador do nome do produto. Segundo (GRILO, 2016), decorre desse entendimento que o produto não deve oferecer apenas uma contribuição tecnológica e econômica para os indivíduos, mas também deve contemplar questões culturais e sociais, aspectos indissociáveis na formação humana.

### 5.3.1. Definição da proposta de valor do aplicativo

Um VPC (Figura 22) foi construído para o aplicativo com base nos dados recolhidos pela pesquisa que foi apresentada na seção 2.5. Na Tabela 5 a seguir, foram selecionadas algumas opiniões utilizadas no canvas.

<b>Relatos de experiências</b>	<b>Sugestões para o aplicativo</b>
<p>“Excelente, pude chegar mais rápido e com conforto ao destino. E ainda vivenciei uma prática de gentileza que alegrou o dia de qualquer um, em troca ficou minha gratidão.”</p> <p>“Estava na parada e um cara que não conheço ofereceu carona para mim e umas amigas minhas. Eu aceitei, mas as meninas, por questão de segurança, não aceitaram. Mas a carona foi tranquila e o cara me deixou na parada do circular no Via Direta.”</p> <p>“Dei carona para conhecidos da minha turma que moram perto de mim, foi uma experiência tranquila e boa pois conheci melhor as pessoas.”</p> <p>“Geralmente quando algum amigo que tem carro me vê na parada do circular, oferece carona. Fora que no meu bairro muitos amigos também tem carro e eles me oferecem também.”</p>	<p>“Um serviço de carona com identificação, na qual só poderá se cadastrar no aplicativo quem tiver matrícula ativa.”</p> <p>“Para oferecer tais caronas, os usuários do aplicativo têm que ter um cadastro para uso e ser realmente verificado esse cadastro (matricula, foto, etc. Em primeiro lugar vem a segurança de quem pegará a carona.”</p> <p>“Acredito que seria importante uma opção onde as mulheres filtrassem os resultados de carona somente entre mulheres.”</p> <p>“Todas as minhas experiências com carona foram com amigos, por isso ressalto a importância de assegurar de alguma forma a segurança.”</p>

**Tabela 5.** Seleção de opiniões retiradas da pesquisa da seção 2.5.



**Figura 22.** VPC construído com base na Tabela 5. (GRILO A.; NETO, 2016).

Como é possível observar na Figura 22, foi construída uma taxinomia (GRILO A.; NETO, 2016) para identificar atributos de cada uma das seções do VPC, de modo que – ganhos (g); dores/frustações (d); tarefas (t); geradores de ganhos (gg); analgésicos (a); produtos e serviços (ps) – são identificados por numeração e diferenciados por cores. Portanto, para se referir a um gerador de ganho para um determinado ganho esperado, escreve-se:  $gg.g(n)$ , onde “n” é o número identificador do ganho. A sintaxe adotada quando mais de um atributo estão envolvidos é, por exemplo,  $gg.g(n')+g(n)$ .

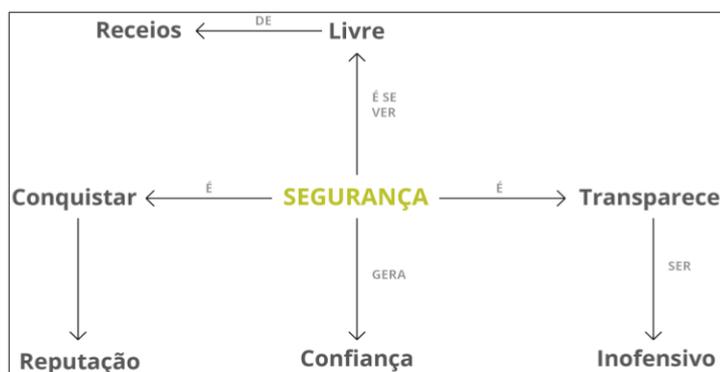
Após analisar os atributos relacionados ao produto e clientes, foram construídas redes semânticas para auxiliar o processo de desenvolvimento do nome do aplicativo.

### 5.3.2. Redes semânticas e resultado

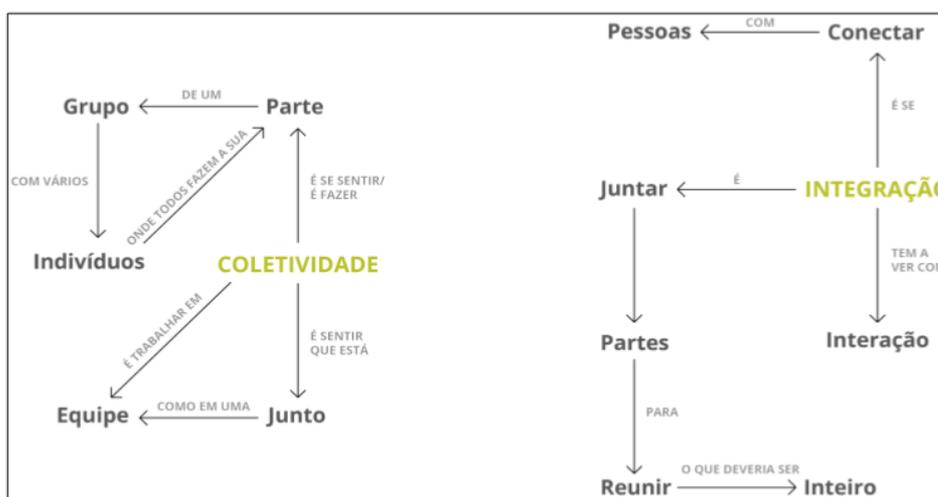
Baseado nas opiniões do questionário da seção 2.5, foi selecionado termos chave e eixos conceituais para o processo de nomeação. Com isso, foram identificadas as qualidades: segurança; empatia; solicitude; coletividade; integração.

Em seguida foram criadas *redes semânticas* para algumas das qualidades mais relevantes. Uma rede semântica representa uma forma de

representação do conhecimento definida como um grafo direcionado no qual os vértices são os conceitos, e as arestas são as relações semânticas entre os conceitos (WIKIPEDIA, 2017). As Figura 23 e 24 apresentam as redes semânticas de segurança, conectividade e integração.



**Figura 23.** Rede semântica da palavra segurança. (GRILO A.; NETO, 2016)

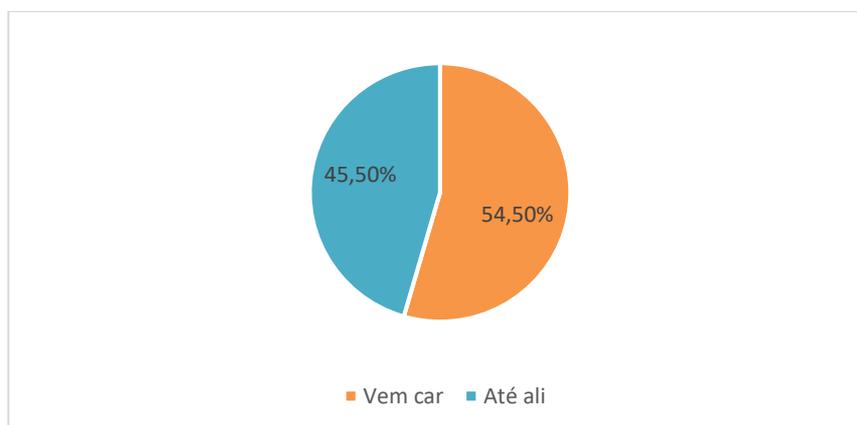


**Figura 24.** Rede semântica das palavras coletividade e integração. (GRILO A.; NETO, 2016)

As redes semânticas são importantes para criar associações entre palavras por meio dos seus vértices e arestas. E a partir dessas redes, foram criados nomes para o aplicativo através de *brainstorms* realizados com as pessoas envolvidas no projeto. Então, foram selecionadas duas opções dentre os vários nomes que surgiram:

- “Até ali”: que sugere um gesto prestativo, confiável e ajudador, utilizando uma expressão gentil e despretensiosa;
- “Vem car”: que sugere uma forma amigável e com linguagem associada a conteúdo da internet, a ideia de coletividade e aproximação.

Depois de pesquisa de disponibilidade nas lojas de aplicativos móveis do Android e iOS, esses dois nomes foram postos para votação junto com o público-alvo. Participaram do processo de votação para o nome do aplicativo as pessoas que deixaram seus e-mails de contato conforme detalhado na questão 9 da seção 2.5.2.



**Figura 25.** Resultado da votação dos nomes.

Dentre as pessoas que deixaram e-mail para contato sobre informações do projeto, um total de 78 participaram do processo de votação para o nome do aplicativo, respondendo à votação que lhes foram enviadas por e-mail. A Figura 25 mostra o gráfico de resultado da votação, onde 54,5% das pessoas optaram por “Vem car” enquanto os 45,5% restantes escolheram “Até ali”. Ficando “Vem car” a opção escolhida pela comunidade acadêmica.

Por fim, a equipe envolvida no projeto optou por concatenar o nome do aplicativo para parecer mais adequado a marca que estava sendo desenvolvida. Tornando oficialmente Vemcar como nome do aplicativo.

## 5.4. Requisitos funcionais

Após um estudo inicial de viabilidade, conforme apresentado na seção 2.5, uma próxima atividade de desenvolvimento foi a elicitação e análise de requisitos. A partir disso, foram elegidos os requisitos funcionais e suas especificações.

A seguir são apresentados requisitos funcionais do Vemcar em diferentes níveis de detalhamento. Verificar Tabela 6.

<b>RF</b>	<b>Nome</b>	<b>Prioridade</b>	<b>Pré-condições</b>	<b>Pós-condições</b>
<b>RF001</b>	Escolher IES associada	Desejável		
<b>RF002</b>	Realizar login	Essencial		
<b>RF003</b>	Realizar logout	Importante	RF002	RF029
<b>RF004</b>	Oferecer carona	Essencial	RF002, RF036, RF038, RF039, RF040, RF041	RF012, RF030
<b>RF005</b>	Solicitar carona	Essencial	RF002, RF038, RF039, RF040, RF041	RF012, RF030
<b>RF006</b>	Ver lista de caronas oferecidas	Importante	RF002	
<b>RF007</b>	Ver lista de caronas solicitadas	Importante	RF002	
<b>RF008</b>	Ver detalhes de uma carona oferecida	Importante	RF002, RF014	
<b>RF009</b>	Ver detalhes de uma carona solicitada	Importante	RF002, RF014	
<b>RF010</b>	Oferecer carona já existente	Desejável	RF002, RF009	RF013, RF032
<b>RF011</b>	Solicitar carona já existente	Desejável	RF002, RF008	RF013, RF032
<b>RF012</b>	Receber uma carona do servidor	Essencial		
<b>RF013</b>	Ver no mapa os componentes de uma carona	Desejável	RF002, RF008 ou RF009	
<b>RF014</b>	Ver lista de caronas oferecidas pelo usuário	Desejável	RF002, RF021	
<b>RF015</b>	Ver lista de caronas solicitadas pelo usuário	Desejável	RF002, RF021	
<b>RF016</b>	Sair de uma carona solicitada	Essencial	RF002, RF005 ou RF011	RF013, RF022, RF032
<b>RF017</b>	Sair de uma carona oferecida	Essencial	RF002, RF004 ou RF010	RF013, RF022, RF032
<b>RF018</b>	Cancelar uma carona	Essencial	RF002, RF008 ou RF009	RF013, RF022, RF032
<b>RF019</b>	Mudar status de uma carona	Importante		RF013, RF032
<b>RF020</b>	Mudar status de um passageiro	Importante		RF013, RF032

<b>RF021</b>	Notificar usuário quando uma carona for solicitada	Desejável	RF002, RF012	
<b>RF022</b>	Notificar usuário quando uma carona for ofertada	Desejável	RF002, RF012	
<b>RF023</b>	Deixar de notificar usuário para novas caronas solicitadas	Desejável		
<b>RF024</b>	Deixar de notificar usuário para novas caronas ofertadas	Desejável		
<b>RF025</b>	Atualizar lista de caronas	Importante	RF002, RF021	
<b>RF026</b>	Limpar dados armazenados pelo aplicativo	Importante	RF003	
<b>RF027</b>	Adicionar uma carona ao banco de dados local	Importante	RF004 ou RF005 ou RF014	
<b>RF028</b>	Remover uma carona do banco de dados local	Importante	RF029	
<b>RF029</b>	Atualizar uma carona no banco de dados local	Importante		
<b>RF030</b>	Mostrar alerta de erro	Desejável		
<b>RF031</b>	Concordar com termos de uso	Importante	RF002	
<b>RF032</b>	Cadastrar veículo	Desejável	RF002	
<b>RF033</b>	Selecionar um veículo cadastrado	Desejável	RF002	
<b>RF034</b>	Informar local de origem da carona	Importante	RF002	
<b>RF035</b>	Informar local de destino da carona	Importante	RF002	
<b>RF036</b>	Informar horário da carona	Importante	RF002	
<b>RF037</b>	Informar a quantidade de vagas para uma carona ofertada	Importante	RF002	
<b>RF038</b>	Checar se todos os dados da carona estão preenchidos	Desejável		

**Tabela 6.** Requisitos funcionais do Vemcar.

### 5.4.1. Especificação dos requisitos funcionais

Nesta seção são apresentadas as especificações dos requisitos funcionais descritos anteriormente. Verificar na Tabela 7.

<b>RF</b>	<b>Nome</b>	<b>Descrição</b>
<b>RF001</b>	Escolher IES associada	O usuário pode escolher uma IES associada ao Vemcar para realizar login.
<b>RF002</b>	Realizar login	Caso o usuário tenha um vínculo ativo com a IES selecionada, ao informar seu usuário e senha de um dos sistemas integrados, terá acesso ao Vemcar.
<b>RF003</b>	Realizar logout	O usuário faz logout no aplicativo. Excluindo dados locais armazenados e voltando para a tela de login.
<b>RF004</b>	Oferecer carona	O usuário informa os dados referentes a uma carona que deseja oferecer e então a carona é adicionada à lista de caronas oferecidas.
<b>RF005</b>	Solicitar carona	O usuário informa os dados referentes a uma carona que deseja solicitar e então a carona é adicionada à lista de caronas solicitadas.
<b>RF006</b>	Ver lista de caronas oferecidas	Na tela principal, o usuário ver uma lista com as caronas que foram oferecidas.
<b>RF007</b>	Ver lista de caronas solicitadas	Na tela principal, o usuário ver uma lista com as caronas que foram solicitadas.
<b>RF008</b>	Ver detalhes de uma carona oferecida	Na tela de detalhes de uma carona, o sistema apresenta algumas informações referente a mesma. São as informações de local de origem e destino de uma carona oferecida, data e hora de partida, foto e nome do motorista (se houver) e os passageiros daquela carona (também se houver).
<b>RF009</b>	Ver detalhes de uma carona solicitada	Na tela de detalhes de uma carona, o sistema apresenta algumas informações referente a mesma. São as informações de local de origem e destino de uma carona solicitada, data e hora de partida, lista com um ou mais pessoas que solicitaram a carona.
<b>RF010</b>	Oferecer carona já existente	Um usuário interessado em se tornar o motorista de uma carona que foi solicitada, pode informar quantidade de vagas disponíveis, local de origem e escolher um ou mais passageiros para a carona.
<b>RF011</b>	Solicitar carona já existente	1) O usuário pode solicitar uma carona que foi solicitada e entrar na lista de pessoas

		<p>que demonstraram interesses por aquela carona.</p> <p>2) O usuário pode solicitar uma carona que foi oferecida e entrar na lista de pessoas que demonstraram interesses por aquela carona.</p>
<b>RF012</b>	Receber uma carona do servidor	O aplicativo deve receber uma carona do servidor quando realizar uma solicitação de serviço web.
<b>RF013</b>	Ver no mapa os componentes de uma carona	É apresentado ao usuário um mapa com componentes de uma carona marcados como pontos interativos.
<b>RF014</b>	Ver lista de caronas oferecidas pelo usuário	O sistema deve apresentar uma lista com todas as caronas ativas que foram oferecidas pelo usuário logado.
<b>RF015</b>	Ver lista de caronas solicitadas pelo usuário	O sistema deve apresentar uma lista com todas as caronas ativas que foram solicitadas pelo usuário logado.
<b>RF016</b>	Sair de uma carona solicitada	Um usuário pode sair da carona solicitada, deixando de ter interesse por ela. Se ele for o único usuário que solicitou esta carona, a mesma deve ser cancelada.
<b>RF017</b>	Sair de uma carona oferecida	<p>1) Um usuário candidato a passageiro pode sair de uma carona oferecida.</p> <p>2) Um motorista pode sair de uma carona oferecida. Se existir outros usuários candidatos a passageiro, a carona passa a ser do tipo solicitada. Se não existir outros usuários interessados na carona, ela será cancelada.</p>
<b>RF018</b>	Cancelar uma carona	Uma carona deve alterar seu status para cancelada.
<b>RF019</b>	Mudar status de uma carona	O servidor deve atualizar o status de uma carona.
<b>RF020</b>	Mudar status de um passageiro	O servidor deve atualizar o status de um passageiro.
<b>RF021</b>	Notificar usuário quando uma carona for solicitada	O aplicativo deve notificar o usuário quando uma nova carona for solicitada se existir uma preferência para isso.
<b>RF022</b>	Notificar usuário quando uma carona for ofertada	O aplicativo deve notificar o usuário quando uma nova carona for ofertada se existir uma preferência para isso.
<b>RF023</b>	Deixar de notificar usuário	O aplicativo não irá notificar novas caronas solicitadas.

	para novas caronas solicitadas	
<b>RF024</b>	Deixar de notificar usuário para novas caronas ofertadas	O aplicativo não irá notificar novas caronas ofertadas.
<b>RF025</b>	Atualizar lista de caronas	A lista de caronas em questão é atualizada com os dados mais recentes do servidor.
<b>RF026</b>	Limpar dados armazenados pelo aplicativo	Ao realizar o logout todos os dados e preferências salvos são apagados
<b>RF027</b>	Adicionar uma carona ao banco de dados local	Uma carona é adicionada sem persistência no servidor.
<b>RF028</b>	Remover uma carona do banco de dados local	Uma carona adicionada ao banco de dados local é removida.
<b>RF029</b>	Atualizar uma carona no banco de dados local	Atualiza uma carona local sem persistência no servidor.
<b>RF030</b>	Mostrar alerta de erro	Uma caixa de diálogo é apresentada ao usuário com uma mensagem de erro.
<b>RF031</b>	Concordar com termos de uso do aplicativo	O usuário deve concordar com os termos de uso do aplicativo para poder utiliza-lo.
<b>RF032</b>	Cadastrar veículo	O usuário pode informar a marca, modelo e placa do seu veículo para poder oferecer carona.
<b>RF033</b>	Selecionar um veículo cadastrado	Para oferecer uma carona o usuário deve antes selecionar um veículo cadastrado.
<b>RF034</b>	Informar local de origem da carona	O usuário deve informar um local onde estará originalmente para oferecer ou solicitar uma carona.
<b>RF035</b>	Informar local de destino da carona	O usuário deve informar um local de destino da carona que deseja ofertar ou solicitar.
<b>RF036</b>	Informar horário da carona.	O usuário deve informar um horário onde estará no local de origem da carona.
<b>RF037</b>	Informar quantidade de vagas para a carona ofertada	Ao ofertar uma carona o usuário deve informar a quantidade de vagas disponível.
<b>RF038</b>	Checar se todos os dados da	O sistema deve checar se todos os campos de uma nova carona solicitada ou ofertada foram

carona estão preenchidos	preenchidos pelo usuário para poder confirmar a oferta ou solicitação de carona.
--------------------------	--

**Tabela 7.** Especificação dos requisitos funcionais do Vemcar.

## 5.5. Interface de usuário

Intrinsecamente relacionado com as disciplinas de Interação Humano-Computador (IHC), Alan Dix publica em seu recente artigo (DIX, 2016) os fundamentos e novos paradigmas da IHC. Ele explica que atualmente os computadores se tornaram ubíqua por ser raro encontrar alguma tecnologia que não envolva computação em algum nível, seja no artefato produzido em si, no seu processo de design, processo de entrega, ou até mesmo na solicitação e oferta de caronas. Fazendo com que novos paradigmas surjam e tragam para a IHC novos desafios.

Um desses paradigmas remete-se a simplicidade. Foi pensada então uma interface de usuário simples para o Vemcar, no sentido de se utilizar componentes já familiarizados pelos usuários na sua composição. Seguindo linhas de design bem-conceituadas para dispositivos móveis, em particular aqueles com sistema operacional Android, a interface gráfica do Vemcar começou a ser desenvolvida ainda durante a disciplina de Desenvolvimento de Sistemas para Dispositivos Móveis. Sua evolução pode ser vista na seção 5.1.

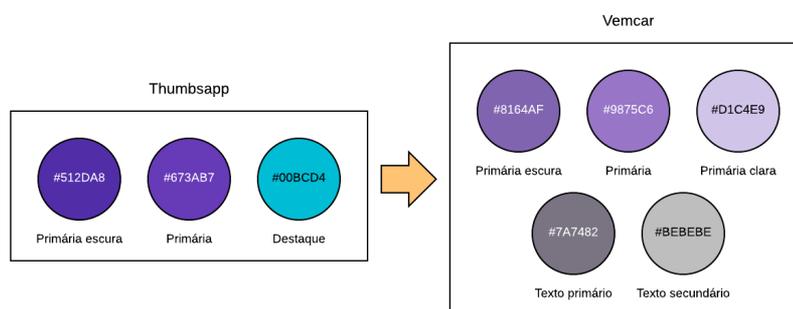
### 5.5.1. Paleta de cores

Manter uma paleta de cores é importante para a uniformidade de uma interface de usuário e ajuda a atingir duas das dez heurísticas de usabilidade de Nielsen (NIELSEN, 1993), são elas: consistência e padrões; projeto minimalista e estético.

Seguindo os princípios apresentados por pelo Material Design (GOOGLE, 2017) para paleta de cores, foram então escolhidas algumas cores para o aplicativo, conforme mostrado na Figura 26 para a versão inicial Thumbsapp e as cores atuais do Vemcar.

Durante o período de desenvolvimento em sala de aula, o aplicativo até então chamado de Thumbsapp possuía três cores retiradas das paletas de

cores oficiais do Material Design. A cor primária é utilizada na barra de títulos, e botões simples. Botão flutuante para solicitar carona e outras informações de destaque, usavam uma tonalidade de azul. O lado direito da Figura 26 apresenta algumas das cores utilizadas na paleta do Vemcar em sua versão de lançamento.



**Figura 26.** Evolução da paleta de cores do Vemcar.

### 5.5.2. Storyboards

Na prototipação e documentação de uma interface de usuário, uma *storyboard* é uma representação visual de como o usuário irá interagir com a interface. Também é utilizado para demonstrar o processo que o usuário deve realizar para atingir certo objetivo. Para auxiliar na prototipação de interfaces, elas podem ser elaboradas a partir de histórias de usuários ou com base nos requisitos levantados.

Nelas, são comuns a presença de setas que indicam o fluxo da interação do usuário, uma descrição para cada quadro ou tela, além de uma numeração para auxiliar a referência entre telas. A seguir, serão apresentadas storyboards do Vemcar com a interface de usuário real e atual do sistema.



**Figura 27.** Storyboard I: instituições, login, termos, instruções e veículo.

A Figura 27 mostra o “*storyboard I*” para o fluxo de atividades de: selecionar uma IES vinculada ao Vemcar; realizar login com sucesso informando as credenciais válidas de um usuário ativo em um dos sistemas integrados ao aplicativo; ler e aceitar os termos e condições de uso do aplicativo; ver ou pular as instruções de algumas funcionalidades; e pular ou cadastrar informações sobre o carro do usuário, informando marca, modelo e placa.



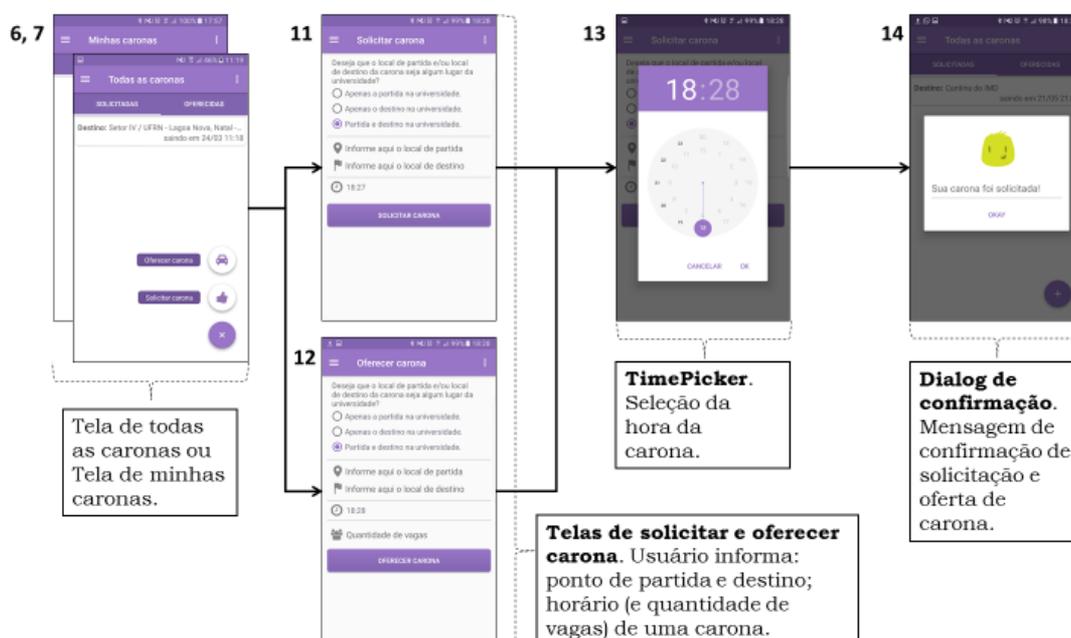
**Figura 28.** Storyboard II: todas as caronas, minhas caronas, menu lateral, configurações e sobre.

O “*storyboard II*” apresentado na Figura 28 mostra o acesso a algumas telas do aplicativo a partir do fragmento principal e menu lateral. Arrastando o dedo da lateral esquerda para a direita nas telas 6, 7, 9 e 10, é possível acessar o menu lateral descrito na tela 8. Nesse menu se tem as seguintes opções:

- Início: mostra o fragmento principal (6) do aplicativo com duas *tabs* para “todas as caronas solicitadas” e “todas as caronas ofertadas”;
- Minhas caronas: mostra o fragmento principal (7), porém desta vez apenas as caronas solicitadas e ofertadas pelo usuário serão exibidas nas *tabs* de “minhas caronas solicitadas” e “minhas caronas ofertadas”, respectivamente;
- Configurações: uma tela de preferencias (9) é apresentada. Nela o usuário pode optar por quais tipos de notificações deseja receber, podendo ser notificado para novas caronas solicitadas, novas caronas ofertadas, ou apenas as caronas que o usuário faz parte;
- Sobre: é apresentada uma tela (10) com o logotipo do aplicativo, a versão instalada no dispositivo, além de links para as principais redes sociais

da SINFO. Por fim também contém um botão que exibe uma caixa de diálogo com os termos de uso do aplicativo;

- Sair: irá limpar, com o auxílio do *Offdroid*, todos os dados locais do Vemcar que estão armazenados no aparelho. Em seguida, o usuário será redirecionado novamente para a tela de login (2).



**Figura 29.** Storyboard III: solicitar e oferecer carona, TimePicker e caixa de diálogo de confirmação.

Outras telas do Vemcar podem ser vistas no “*storyboard III*” presente na Figura 29. Nessa sequência são registradas as tarefas de oferecer e solicitar caronas no aplicativo.

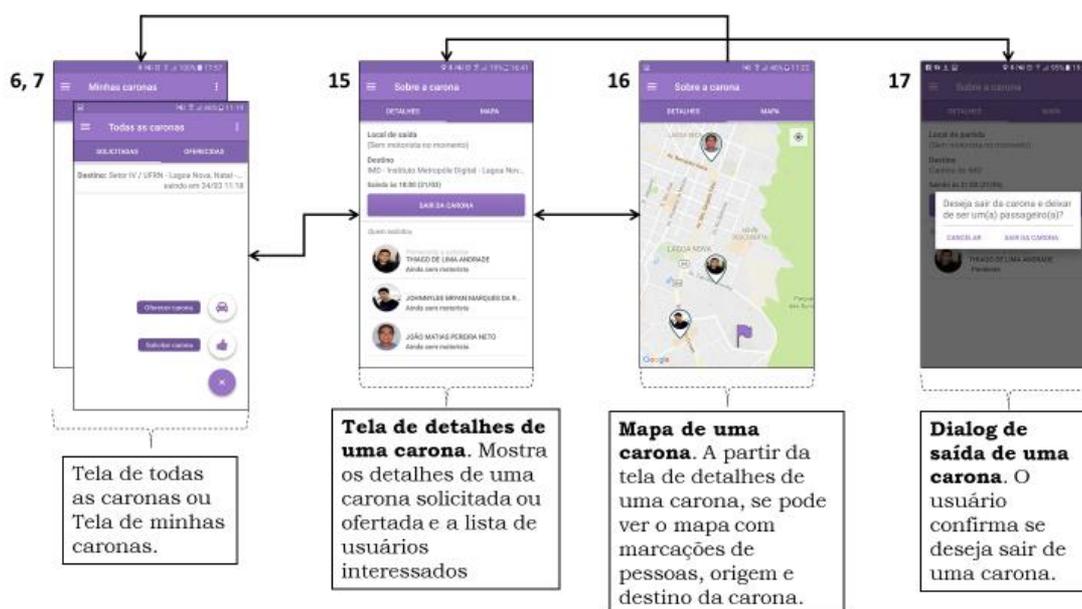
Nessas telas (11 e 12), o que diferencia uma carona oferecida de uma solicitada é que a primeira requer que o usuário informe o veículo utilizado e a quantidade de vagas para a carona em questão. Já uma carona solicitada requer apenas os mesmos pontos requeridos também por uma oferecida, que são:

- Ponto de origem e destino: nesse campo de texto o usuário irá informar os pontos de origem e destino da sua carona oferecida ou solicitada. Ao digitar nesses dois campos, um recurso de auto completar o texto é ativado a partir do terceiro caractere inserido.

Desse modo o usuário pode selecionar um dos pontos que aparecem em uma caixa “dropdown” (Spinner no Android);

- TimePicker: essa caixa de diálogo com um TimePicker (13) é comum no Android quando se deseja que um usuário informe um horário. Esse horário no caso define a hora de saída de uma carona solicitada ou oferecida. A caixa de diálogo com TimePicker é idêntica tanto numa carona solicitada quanto em uma oferecida;

Com todos os campos requeridos preenchidos é possível oferecer ou solicitar uma carona. Se estiver todos os campos em conformidade com o que é requisitado pelo aplicativo, então uma carona é criada, persistida no banco de dados local, no servidor remoto e a caixa de diálogo da tela 14 é apresentado com a mensagem de confirmação de que a carona foi solicitada ou oferecida.



**Figura 30.** Storyboard IV: detalhes, mapa de uma carona e caixa de diálogo de confirmação de saída.

Na Figura 30, o “*storyboard IV*” que começa no fragmento principal e se houver uma carona que o usuário selecione, apresentará uma tela com detalhes daquela carona, exemplificado pela tela 15. A activity de detalhes de uma carona contém um *tabview* com duas *tabs*. Na primeira, mostram os detalhes em campos de texto para o local de origem, local de destino da

carona, horário e lista de interessados nessa carona. A segunda *tab* contém um mapa do Google Maps com marcadores para cada interessado, origem e destino da carona. O usuário pode interagir com os marcadores os selecionando para oferecer uma carona.

Pode-se perceber nesses quatro *storyboards* que algumas telas atendem diretamente requisitos funcionais descritos na seção 4.4.1. Graças a essa ligação entre atividades distintas no processo de produção de um produto de software, foi possível criar a Tabela 8 a seguir.

<b>Requisito Funcional</b>	<b>Tela nos storyboards I, II, III e IV</b>
RF001	<b>1:</b> é possível selecionar uma das IES participantes do Vemcar a partir da lista de instituições presentes nesta tela.
RF002	<b>2:</b> o usuário deve informar um nome de usuário e senha com vínculo ativo em um dos sistemas associados ao Vemcar para poder utilizar o aplicativo. Não é possível se cadastrar de outra forma.
RF034	<b>3:</b> para utilizar o aplicativo o usuário deve selecionar o checkbox dessa tela, declarando que leu os termos de uso e aceita-los.
RF035	<b>5:</b> para oferecer caronas, o usuário deve cadastrar um veículo selecionando uma das marcas e modelos existentes no aplicativo, ou informando manualmente a marca e modelo. Em seguida deve-se informar a placa do veículo.
RF006, RF007 e RF021	<b>6:</b> mostra todas as caronas ativas que foram solicitadas ou ofertadas.
RF016, RF017 e RF021	<b>7:</b> mostra todas as caronas ativas que foram solicitadas ou ofertadas pelo usuário logado.
RF024, RF025, RF026 e RF027	<b>9:</b> permite o usuário escolher quais notificações deseja receber.
RF005, RF012, RF030, RF037, RF038, RF039 e RF041	<b>11:</b> solicitar uma carona informando o local de origem, destino e horário.
RF006, RF012, RF030, (RF035 ou RF036), RF037, RF038, RF039 e RF041	<b>12:</b> ofertar uma carona informando o local de origem, destino, horário, veículo e quantidade de vagas
RF008 ou RF009	<b>15:</b> ver detalhes de uma carona solicitada ou ofertada.
RF015	<b>16:</b> ver mapa de uma carona solicitada ou ofertada.
RF013, ((RF018 e RF020) ou RF018),	<b>17:</b> sair de uma carona solicitada ou ofertada.

((RF019 e RF020) ou RF019)	
-------------------------------	--

**Tabela 8.** Relação entre requisitos funcionais e telas do Vemcar.

## 6. Implementação da solução

Este capítulo trata como os componentes internos do cliente do Vemcar se relacionam para que o aplicativo funcione. Além disso, serão descritas as interações dos componentes do aplicativo com suas dependências externas.

### 6.1. Modelagem estrutural

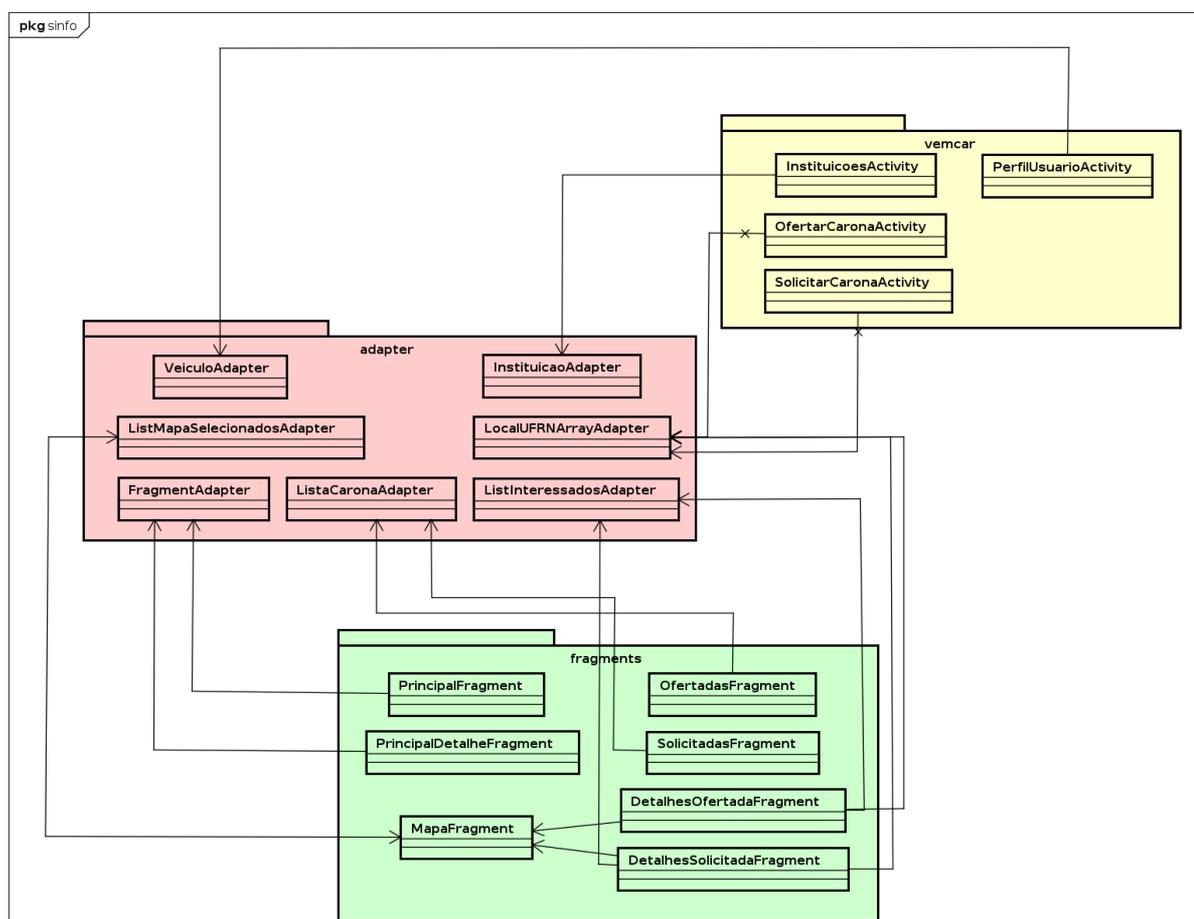
O aplicativo foi projetado seguindo o modelo arquitetural Cliente-Servidor. O cliente foi desenvolvido utilizando aproximadamente 60 classes escritas em Java, conforme a Figura 32 a seguir, que apresenta os pacotes do Vemcar com suas respectivas classes. Além de classes Java, diversos layouts e recursos de desenho na tela foram escritos usando XML, assim como definido no Android.

Os pacotes do Vemcar podem ser definidos conforme suas responsabilidades. Assim segue:

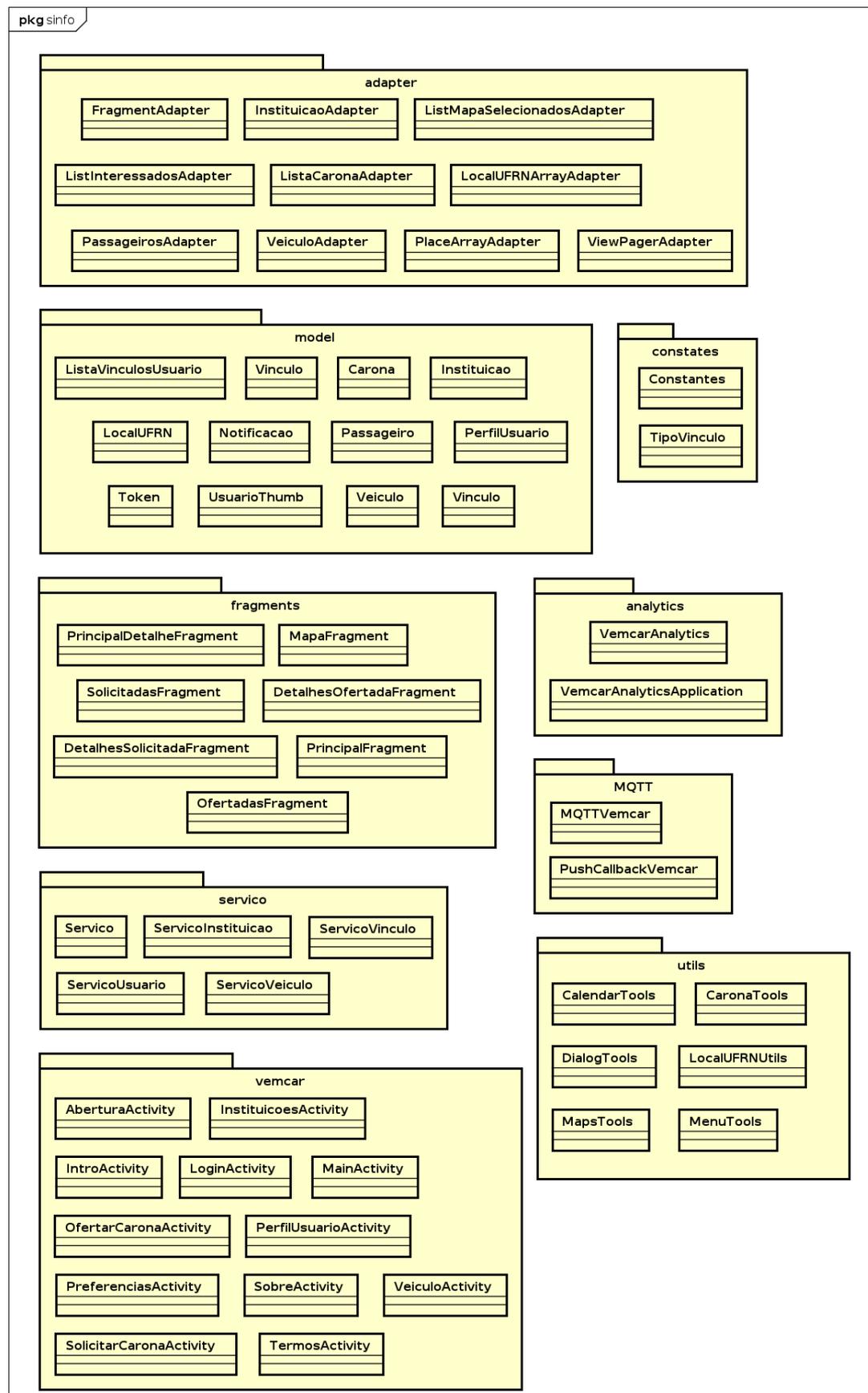
- Adapter: ligam os dados oriundos de requisições ou inseridos pelo usuário com os elementos de layout por meio das Activitys e Fragments que compõem o aplicativo;
- Model: agrupa as classes de modelo, ou domínio, da aplicação;
- Constantes: guardam classes que contêm enumerações ou tipos primitivos estáticos que são utilizados com frequência no aplicativo;
- Fragments: agrupam classes que estendem o Fragment do Android;
- Analytics: classes que se comunicam com o Google Analytics para geração de dados estatísticos do aplicativo;
- MQTT: implementa alguns dos recursos do Subscriber do MQTT no Vemcar;

- Servico: são as classes que se comunicam com a API.sistema e o servidor do Vemcar para todas as funcionalidades do aplicativo que requer internet;
- Utils: são classes que auxiliam o reuso de software, provendo manipulação de calendário, caixas de dialogo para usuário, funções comuns a todos os tipos de carona, etc.;
- Vemcar: armazena todas as Activitys do aplicativo.

A Figura 31 mostra outro diagrama de classes do Vemcar. Nele estão representadas as classes que atendem a manipulação de dados e recursos internos do aplicativo com sua interface. São visões e modelos do aplicativo, em analogia ao MVC (Model-View-Controller). As ligações entre Fragment e Adapter ou Activity e Adapter, indicam quais Fragments e Activitys incluem quais Adapters.



**Figura 31.** Relação entre Fragment, Activity e Adapter no Vemcar.



**Figura 32.** Pacotes do Vemcar e suas classes.

## 6.2. Acesso e autorização utilizando API.sistemas

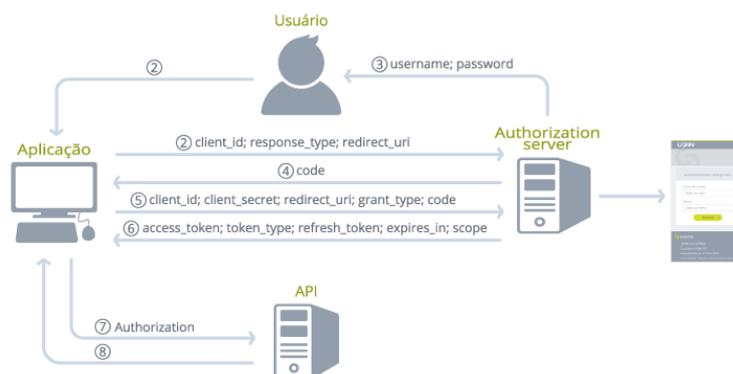
A *API.sistemas* é um serviço *RESTful* da UFRN e tem como principal função disponibilizar os dados dos sistemas da universidade e, assim, possibilitar que a comunidade acadêmica, crie aplicações em qualquer plataforma desejada, que necessitam desses dados para funcionar.

Para que o Vemcar possa oferecer os requisitos de realizar login utilizando login e senha que os usuários normalmente utilizam nos sistemas da UFRN e também para acessar informações privadas das contas dos usuários, um recebimento de credências, autorização e autenticação na *API.sistemas* deve ser feito. Assim, os seguintes passos abaixo precisam ser seguidos por uma aplicação que deseja realizar autenticação nos serviços da UFRN:

1. O usuário inicia a interação com a aplicação, como na tela de login do Vemcar, por exemplo;
2. O aplicativo faz uma requisição GET ao *authorization server* através da URL `http://apitestes.info.ufrn.br/authz-server/oauth/authorize`, passando os parâmetros *cliente\_id*, *response\_type* e *redirect\_uri* como QueryParam;
3. O usuário é redirecionado para o *authorization server*. Na página de autenticação exibida, deve informar suas credenciais (*username*, *password*) e, em seguida, informar se autoriza que a aplicação utilize seus dados. Para garantir a segurança das informações dos usuários, normalmente a página de autenticação exibida é a do servidor de autenticação da SINFO. Vale ressaltar que o Vemcar funciona de modo diferente nesse quesito, por se tratar de um aplicativo oficial da SINFO. No caso do Vemcar, a página de autenticação exibida é própria do aplicativo e controlada pela classe *LoginActivity*.
4. O *authorization server* retorna o código de autorização para a aplicação;
5. Em posse desse código, a aplicação pode usá-lo para obter um *access\_token* para o usuário. Desse modo, ela realiza uma nova requisição, que neste caso é do tipo POST, ao *authorization server* através da URL `http://apitestes.info.ufrn.br/authz-server/oauth/token`,

passando os parâmetros *cliente\_id*, *cliente\_secret*, *redirect\_uri*, *grant\_type* e *code* como QueryParam;

6. O *authorization server* retorna à aplicação um JSON contendo o *access\_token*, *cliente\_token*, *token\_type*, *refresh\_token*, *expires\_in* e *scope*.  
Ex.: { "access\_token": "111", "token\_type": "bearer", "refresh\_token": "abcd", "expires\_in": 7431095, "scope": "read" }
7. Em posse dessas informações, a aplicação já pode acessar os dados disponibilizados pela API passando o *token* através do parâmetro *Authorization* no *header* da requisição desejada;
8. Os dados da API são retornados para aplicação no formato JSON.



**Figura 33.** Autenticação e autorização na API.sistemas (APISISTEMAS, 2017).

A ilustração desses passos descritos anteriormente pode ser vista na Figura 33.

No Vemcar a classe que controla o mecanismo de login é a *LoginActivity*, fazendo uso de métodos com essa finalidade contidos na classe *ServicoUsuario*.

### 6.3. Persistência de dados com OffDroid

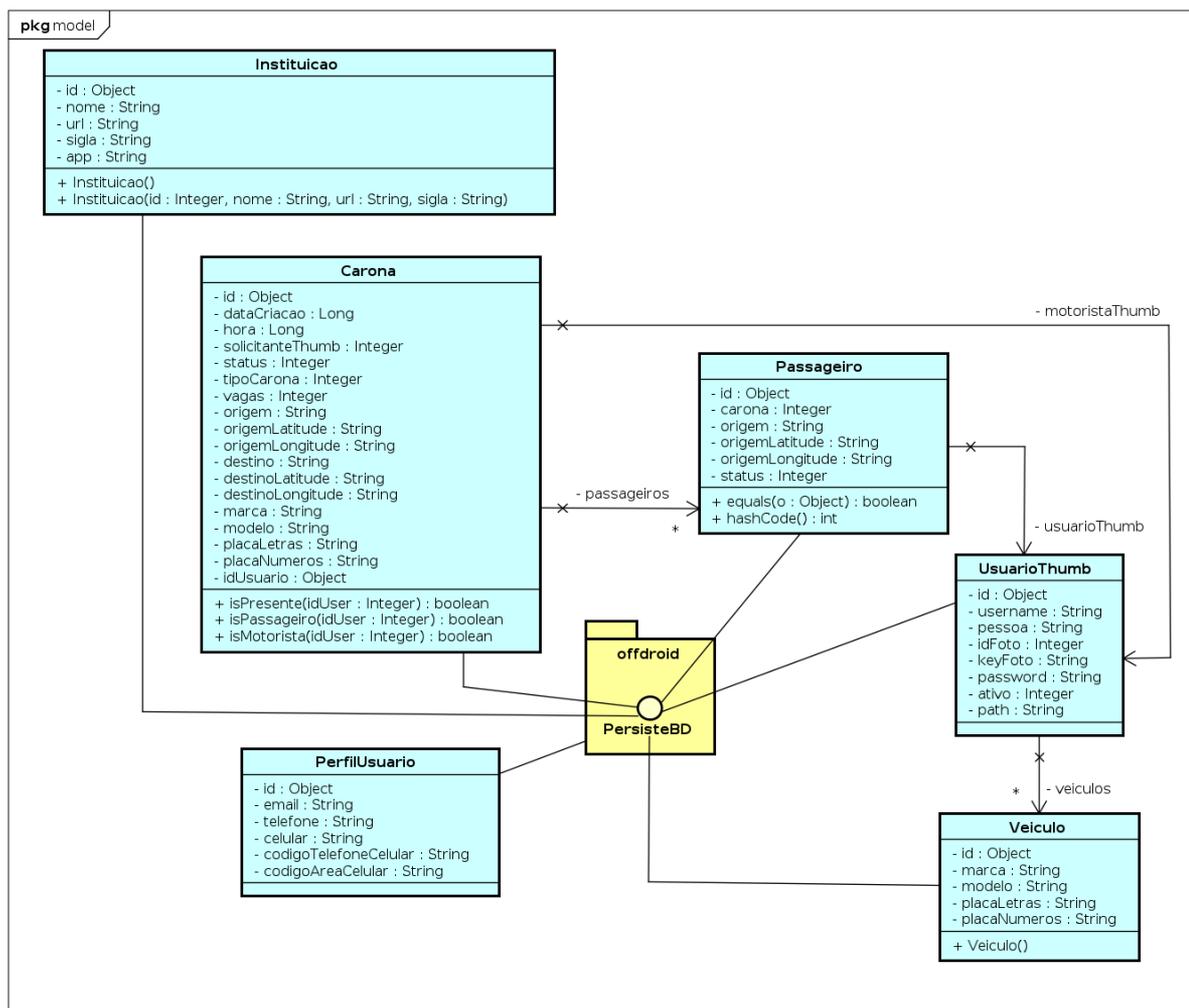
Algumas informações do lado cliente do aplicativo Vemcar devem ser armazenadas no banco de dados local do dispositivo, outras precisam que sejam salvas tanto localmente quanto persistidas no servidor para registro de uso e consultas futuras.

O armazenamento de informações tanto offline e local quanto online no servidor se dá por meio do framework OffDroid. O framework também auxilia

em consultas e atualizações REST. Uma recente atualização no framework faz adição de novas anotações que foram adotadas para classes de domínio. As novas anotações utilizadas no Vemcar são:

- *@Headers("Authorization")*: informa um atributo no cabeçalho da requisição. “Authorization” neste caso, diz respeito ao requisito necessário imposto pela API.sistema;
- *@Path("carona")*: essa anotação diz o caminho que a requisição deve ser feita. Pode-se apenas concatenar uma string a uma URL base armazenada no cliente, como foi feito com “carona”, ou pode-se também passar por parâmetro da anotação uma URL completa.
- *@Encoding("UTF-8")*: diz respeito ao tipo de codificação do corpo da requisição.
- *@GET("url\_api")*: informa por parâmetro uma URL para requisição GET.

A Figura 34 acima mostra um diagrama de classes do pacote “model” do aplicativo. Esse pacote contém classes de domínio que implementam a interface PersisteBD do OffDroid. O sufixo “Thumb” das classes de usuário e motorista se refere a antiga nomenclatura do aplicativo na fase descrita na seção 5.1.1.



**Figura 34.** Diagrama de classes do pacote *model* que utilizam a classe *PersisteBD* do OffDroid.

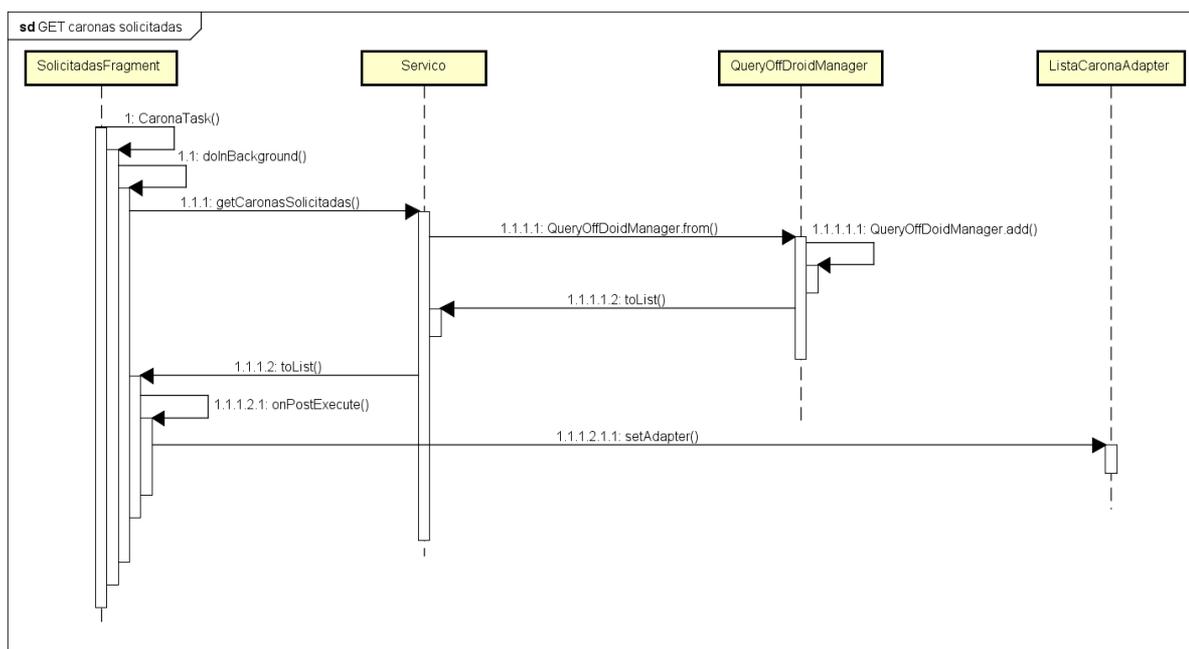
O diagrama também mostra a relação entre as classes de domínio. Uma instancia *Carona* pode ter zero ou um motorista (0..1) e zero ou muitos passageiros (0..\*). Um passageiro e um motorista possuem um *UsuarioThumb*. E uma instancia desse usuário pode ter zero ou muitos veículos (0..\*).

	@Headers	@Path	REST	@OnlyOnline
<i>Instituição</i>	Sim	Sim	@GET	Sim
<i>Carona</i>	Sim	Sim		
<i>Passageiro</i>	Sim	Sim		
<i>PerfilUsuario</i>	Sim	Sim	@GET	Sim
<i>UsuarioThumb</i>	Sim	Sim	@GET	
<i>Token</i>		Sim	@POST	Sim

**Tabela 9.** Anotações do OffDroid nas classes de domínio do Vemcar.

A Tabela 9 faz uma relação das anotações utilizadas nas classes do diagrama anterior.

A seguir é apresentado um exemplo de requisição usando o OffDroid para solicitar através da classe *QueryOffdroidManager* uma lista de caronas solicitadas. O diagrama de sequência da Figura mostra o fragmento de caronas solicitadas (*SolicitadasFragment*) executando uma atividade assíncrona que chama o método *getCaronasSolicitadas()* da classe *Servico* e em seguida, após execução, atualiza o Adapter que mostrará tais caronas ao usuário na forma de lista.



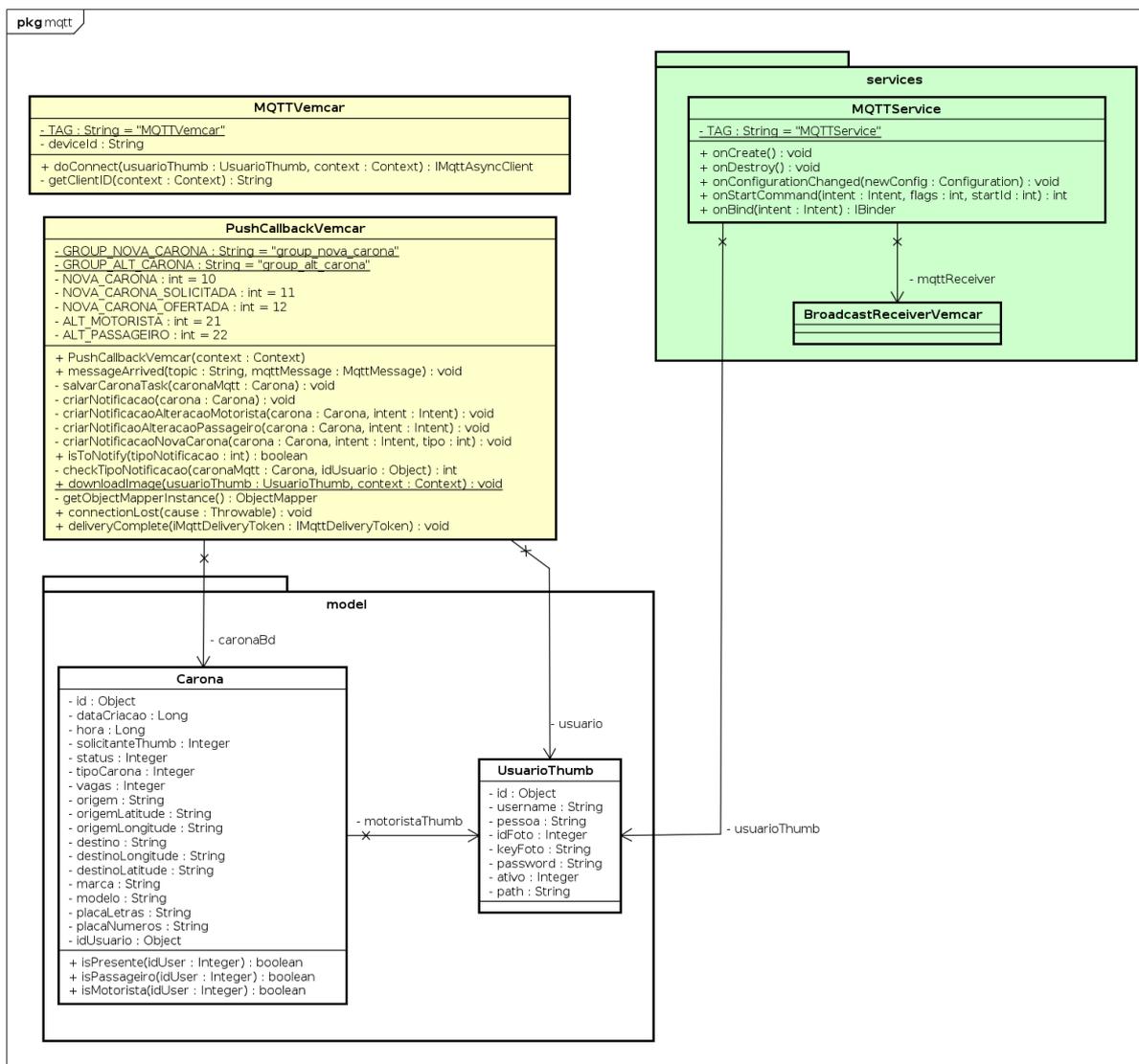
**Figura 35.** Diagrama de sequência de uma solicitação de lista de caronas.

Portanto, conforme apresentado, o OffDroid executa papel importante no aplicativo. Sendo ele responsável por intermediar a comunicação entre Cliente e Servidor.

## 6.4. Uso do MQTT

O diagrama de classes a seguir, apresenta o pacote MQTT do cliente Vemcar. A classe *MQTTVemcar* é responsável por estabelecer uma conexão com o MQTT no Servidor. No pacote Services, a classe *MQTTService* em conjunto com *BroadCastReceiverVemcar*, implementar o Subsciber do MQTT. No Servidor existe uma classe *MQTTUtils* que trabalha como Broker.

Já o `PushCallbackVemcar` é acionado quando uma mensagem chega no tópico “carona”. Nesse tópico estão inscritos todos os clientes do aplicativo.



**Figura 36.** Diagrama de classes do pacote MQTT.

## 7. Considerações finais

Neste trabalho foi descrito o projeto e desenvolvimento do Vemcar, um aplicativo de caronas voltado para uso exclusivo da comunidade acadêmica da Universidade Federal do Rio Grande do Norte. Foram adotados aspectos de tendências sociais para oferta e solicitação de caronas e conhecimentos técnicos na área de desenvolvimento de software para a produção de uma solução entregue ao usuário final.

O desenvolvimento do Vemcar foi um grande exercício de Engenharia de Software, para seu projeto e implementação foram utilizados conhecimentos obtidos ao longo do curso. Diversas atividades da Engenharia de Software foram utilizadas na prática para a confecção do projeto, dentre eles, as disciplinas de Análise e Desenvolvimento de Programação Orientada a Objetos, Engenharia de Requisitos, Qualidade de Software, Manutenção de Software, Boas Práticas de Programação, Arquitetura de Software, Processos de Software, Gerenciamento e Planejamento de Projetos, dentre outras.

Durante o período de um ano e meio o aplicativo sofreu várias mudanças de interface e atualizações de requisitos. Todo esse percurso trabalhado em uma equipe multidisciplinar, composta por desenvolvedores mobile, analista de sistema e designer, tornaram a experiência de desenvolver um aplicativo ainda mais construtiva. Foi graças a equipe da SINFO que o protótipo de aplicativo que surgiu em uma disciplina, pode enfim se tornar real e entregue para a comunidade acadêmica.

Colher feedbacks de usuários com suas críticas construtivas e outras nem tão construtivas assim também fez parte do processo de amadurecimento do aplicativo, que ainda tem muito o que crescer.

E já vem crescendo. Durante as duas semanas seguintes ao lançamento do Vemcar, foram dadas entrevistas a telejornais locais e rádio, além de programa de estúdio. Foram entrevistas com a TV Câmara, Band, TV Tropical (Rede Record), Inter TV Cabugi (Rede Globo) e Rádio Universitária. Poder expor

o Vemcar em veículos de mídia foi uma grande oportunidade para o aplicativo e experiência pessoal, uma vez que o contato com câmeras não é tão presente na vida de muitas pessoas.

Tudo isso trouxe como resultado, cerca de 1500 downloads na loja do Google Play durante os 14 primeiros dias depois de lançado. Esse expressivo número de download para um aplicativo até então inexistente, refletiu também na quantidade de caronas solicitadas e ofertadas. Foram uma média de 50 requisições diárias armazenadas nos servidores da SINFO.

Isto trouxe outro ensinamento, a correção rápida de erros. Pois uma vez lançado, surgiram erros que impediam o uso do aplicativo em alguns smartphones da marca ASUS. Isso fez com que algumas avaliações negativas comesçassem a surgir em meio as tantas positivas. Trabalhar na correção rápida de erros devido a isso foi uma experiência que não tinha tido até então na universidade. Afinal, não se tratava mais de um projeto de disciplina que uma vez entregue, o professor retorna para os alunos alguns feedbacks e uma nota. Mas nesse momento não, nesse momento se trata de um aplicativo entregue a usuários reais, com problemas reais. Ou seja, uma ótima atividade de software.

Como trabalhos futuros, espera-se o desenvolvimento de uma versão do Vemcar na plataforma iOS, pois estes dispositivos representam em média um quarto dos usuários de aplicativos oficiais da instituição. Além da versão iOS, o aprimoramento e desenvolvimento de novas funcionalidades na versão Android é de fundamental importância. Dentre as novidades, a integração com o aplicativo Olar, um aplicativo de chat do SIGAA, tornará o Vemcar mais dinâmico, permitindo conversas entre os usuários.

Outras novidades para trabalhos futuros seria as mudanças de layout que irão ocorrer nas próximas versões do Vemcar. Futuramente o aplicativo irá deixar de contar apenas com duas listas na tela inicial para conter com um mapa que terá integrado todos os pontos referentes a caronas solicitadas e oferecidas. Tornando seu layout similar ao aplicativo Uber, líder de downloads e engajamento na categoria. Finalizo encorajando os graduandos de Engenharia de Software a completarem um projeto de disciplina do início até a entrega para um usuário final. Não há arrependimentos!

## Referências

ANDRADE, É. O individualismo motorizado. *UFPE – Universidade Federal de Pernambuco*, p. b9, 2012.

APISISTEMAS: Autorização e Autenticação. Disponível em: <<https://api.ufrn.br>> Acesso em Abril 25, 2017.

CARONAE: Aplicativo de caronas da UFRJ. Disponível em: <<https://caronae.ufrj.br>>. Acesso em Março 13, 2017.

CARONAPHONE: Carona Phone, uma solução inovadora de carona solidária. Disponível em: <<http://caronaphone.com>>. Acesso em Março 13, 2017.

DENATRAN: Frota de veículos por ano. Disponível em: <<http://www.denatran.gov.br/index.php/estatistica/237-frota-veiculos>>. Acesso em Abril 21, 2017.

DEVELOPER: Desenvolvedores Android | Android Developers. Disponível em: <<https://developer.android.com/index.html>>. Acesso em Setembro 10, 2016.

DIX, A. Human-computer interaction, foundations and new paradigms. *Journal of Visual Languages and Computing*, Julho 2016.

GEHL, J. Cidades para pessoas. 2nd. ed. [S.l.]: *Perspectiva*, 2013.

GOOGLE. *Introducion – Material Design*. Outubro, 2014. Disponível em: <<https://material.io/guidelines/>>. Acesso em Abril de 2017.

GRILO, A; NETO, G. F. L. Value proposition canvas aplicado ao processo de design: Desenvolvimento de nome e identidade visual para aplicativo de mobilidade em campus universitário. [S.l.]: *Anais do 12º Congresso Brasileiro de Pesquisa e Desenvolvimento em Design*, Belo Horizonte - MG, 2016.

GUERETHES, J. Uma proposta de solução para funcionamento offline em aplicações Android. [S.l.] *Programa de Pós-Graduação em Sistemas e Computação, UFRN*. Natal – RN, 2015.

HIVEMQ: MQTT Essentials Part 2: Publish & Subscribe. Disponível em:

<<http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe>>. Acesso em Maio 19, 2017.

IBGE: Estimativa da população 2016. Disponível em:

<[http://www.ibge.gov.br/home/estatistica/populacao/estimativa2016/serie\\_2001\\_2016\\_tcu.shtm](http://www.ibge.gov.br/home/estatistica/populacao/estimativa2016/serie_2001_2016_tcu.shtm)>. Acesso em Abril 21, 2017.

NIELSEN, J. *Usability Engineering*. Boston: Academic Press, Cambridge – MA, 1993.

OSTERWALDER, A.; SMITH, A.; PIGNEUR, Y; BERNARDA, G. *Value Proposition Design: How to Create Products and Services Customers Want*. John Wiley & Sons, 2014.

PLANSKY, R. Definição, restrições e benefícios do modelo de arquitetura REST. Disponível em: <<https://imasters.com.br/desenvolvimento/definicao-restricoes-e-beneficios-modelo-de-arquitetura-rest/?trace=1519021197&source=single>>. Publicado em Outubro 02, 2014.

SCHWANEN, T.; PEREIRA, R. Tempo de deslocamento casa-trabalho no Brasil (1992-2009): Diferença entre regiões metropolitanas, níveis de renda e sexo. [S.l.]: IPEA, 2013.

SOMMERVILLE, I. *Software Engineering*. 9th. ed. [S.l.]: Pearson, 2011.

WIKIPEDIA: Redes semânticas. Disponível em: <[https://pt.wikipedia.org/wiki/Rede\\_sem%C3%A2ntica](https://pt.wikipedia.org/wiki/Rede_sem%C3%A2ntica)>. Acesso em Maio 1, 2017.