

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**APLICATIVO PARA OFERTA E OBTENÇÃO DE CARONAS**  
**NA PLATAFORMA ANDROID**

**JOÃO PAULO CONCEIÇÃO**

**BLUMENAU**  
**2015**

**2015/1-11**

**JOÃO PAULO CONCEIÇÃO**

**APLICATIVO PARA OFERTA E OBTENÇÃO DE CARONAS  
NA PLATAFORMA ANDROID**

Trabalho de Conclusão de Curso submetido à  
Universidade Regional de Blumenau para a  
obtenção dos créditos na disciplina Trabalho  
de Conclusão de Curso II do curso de Sistemas  
de Informação— Bacharelado.

Prof. Jacques Robert Heckmann, Mestre - Orientador

**BLUMENAU  
2015**

**2015/1-11**

# **APLICATIVO PARA OFERTA E OBTENÇÃO DE CARONAS NA PLATAFORMA ANDROID**

Por

**JOÃO PAULO CONCEIÇÃO**

Trabalho aprovado para obtenção dos créditos  
na disciplina de Trabalho de Conclusão de  
Curso II, pela banca examinadora formada  
por:

Presidente:	<hr/> Prof. Jacques Robert Heckmann, Mestre - Orientador, FURB
Membro:	<hr/> Prof. Cláudio Ratke, Mestre - FURB
Membro:	<hr/> Prof. Francisco Adell Péricas, Mestre - FURB

Blumenau, 07 de julho de 2015.

Dedico este trabalho a todos os amigos e familiares, especialmente aqueles que me ajudaram diretamente na realização deste.

Todas as vitórias ocultam uma abdicação.

Simone de Beauvoir

## **AGRADECIMENTOS**

A Deus, pelo seu imenso amor e graça.

À minha família, que me apoiou em todos os momentos e forneceu a estrutura necessária para chegar a este momento.

Aos meus amigos, pelos empurrões e cobranças.

Ao meu orientador, professor Jacques Robert Heckmann, por ter contribuído com suas sugestões, incentivado a conclusão deste trabalho e acreditado nele em todos os momentos.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

## RESUMO

Os custos para aquisição e manutenção de um veículo automotor, somados à situação de caos urbano encontrada no trânsito de alguns grandes centros originam a busca por meios alternativos de transporte. A utilização de caronas é uma alternativa para redução de custos com deslocamento e melhoria de mobilidade urbana. Este trabalho apresenta um aplicativo para a plataforma Android que automatiza a oferta de caronas e a localização de usuários que percorram trajetos semelhantes. Para o desenvolvimento do sistema foi utilizada a linguagem Java, juntamente com a API Google *Cloud Endpoints*, utilizada para persistência de dados na nuvem. A API Google *Directions* foi utilizada para obtenção da rota percorrida pelo usuário com base na origem e destino inseridos. Os resultados obtidos demonstram a adequação das tecnologias utilizadas para o propósito determinado.

Palavras chaves: Caronas. Google *Cloud Endpoints*. Google *Directions*.

## **ABSTRACT**

The costs of a motor vehicle acquisition and maintenance , added to the urban caos situation founded on the traffic of some big centers originate the search for alternative transport methods. The carpooling use is an alternative to reduce the displacement costs and urban mobility improvement . This paper presents an Android platform app which automates the offer of carpooling and the location of users who course similar paths. Java language was used to the system development , along with Google Cloud Endpoints API, used for cloud data persistence. The Google Directions API was used to obtain the route covered by the user based on the inserted origin and destination. Obtained results demonstrate the adequacy of the used technologies to the particular purpose.

**Key-Words:** Carpooling. Google Cloud Endpoints. Google Directions.



## LISTA DE FIGURAS

Figura 1 - Diagrama de representação da nuvem .....	19
Figura 2 - Esquema de cluster com mais de um rack .....	21
Figura 3 - Arquitetura básica do Google <i>Cloud Endpoints</i> .....	22
Figura 4 - URL de requisição à API e retorno em texto .....	23
Figura 5 - Tela do Aplicativo Uber para Android .....	24
Figura 6 - Diagrama de Casos de Uso.....	27
Figura 7 - Modelo Entidade-Relacionamento .....	28
Figura 8 - Classes e pacotes .....	29
Figura 9 - Tela Inicial da IDE Android Studio .....	31
Figura 10 - Android SDK Manager .....	31
Figura 11 - Estrutura de Classes Java do App.....	32
Figura 12 - Estrutura de Classes de <i>Layout XML</i> .....	32
Figura 13 - Estrutura de Classes Java do <i>Backend</i> .....	33
Figura 14 - Tela de <i>Login</i> do Aplicativo .....	34
Figura 15 - Tela de Cadastro de Usuário .....	34
Figura 16 - Entidade usuário visualizada pelo administrador .....	35
Figura 17 - Tela de Cadastro de Carona.....	35
Figura 18 - Pontos Rota de um trajeto .....	36
Figura 19 - Classe <i>ConsultaDirections</i> .....	36
Figura 20 - Trajeto entre as ruas Amazonas e Nereu Ramos .....	37
Figura 21 - Busca de Carona I.....	37
Figura 22 - Busca de Carona II.....	38
Figura 23 - Método <i>consultarCaronas()</i> .....	39
Figura 24 - Fluxograma do Método <i>consultarCaronas()</i> .....	40

## LISTA DE QUADROS

Quadro 1 - Requisitos funcionais .....	26
Quadro 2 - Requisitos não funcionais .....	26
Quadro 3 - Comparação entre sistemas correlatos .....	41
Quadro 4 - Descrição dos Casos de Uso .....	48
Quadro 5 - Tabela Carona .....	50
Quadro 6 - Tabela Usuario .....	50
Quadro 7 - Tabela PontoRota .....	51

## LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas  
ANP – Agência Nacional de Petróleo  
API – *Application Programming Interface*  
BCC – Curso de Sistemas de Informação – Bacharelado  
CNT – Confederação Nacional de Transporte  
Denatran – Departamento Nacional de Trânsito  
DSC – Departamento de Sistemas e Computação  
IDE – *Integrated Development Environment*  
IEA – *International Agency of Energy*  
GEE – Gases de Efeito Estufa  
HOV – *High-Occupancy Vehicle*  
HTTP - *Hypertext Transfer Protocol*  
MCT – Ministério de Ciência e Tecnologia  
MER – Modelo Entidade-Relacionamento  
UC – *Use Case*  
UML – *Unified Modeling Language*  
URL – *Uniform Resource Location*  
XML – *Extensible Markup Language*

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>12</b>
1.1 OBJETIVOS DO TRABALHO.....	13
1.2 ESTRUTURA DO TRABALHO .....	13
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
2.1 CARONA .....	15
2.2 A QUESTÃO AMBIENTAL .....	16
2.3 LEGISLAÇÃO BRASILEIRA.....	17
2.4 PLATAFORMA ANDROID.....	18
2.5 COMPUTAÇÃO NA NUVEM .....	18
2.6 BANCO DE DADOS NÃO RELACIONAL.....	20
2.6.1 Google <i>cloud endpoints</i> .....	21
2.7 GOOGLE <i>DIRECTIONS</i> API.....	22
2.8 TRABALHOS CORRELATOS .....	23
<b>3. DESENVOLVIMENTO DO APLICATIVO E DO <i>BACKEND</i> .....</b>	<b>25</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	25
3.2 ESPECIFICAÇÃO.....	25
3.2.1 Requisitos Funcionais e Não Funcionais .....	25
3.2.2 Casos de Uso .....	26
3.2.3 Modelo Entidade-Relacionamento .....	27
3.3 IMPLEMENTAÇÃO.....	30
3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS .....	30
3.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO .....	33
3.4 RESULTADOS E DISCUSSÃO.....	41
<b>4. CONCLUSÕES .....</b>	<b>42</b>
4.1 EXTENSÕES .....	42
<b>REFERÊNCIAS.....</b>	<b>44</b>
<b>APÊNDICE A – DETALHAMENTO DOS CASOS DE USO .....</b>	<b>48</b>
<b>APÊNDICE B – DICIONÁRIO DE DADOS.....</b>	<b>50</b>

## 1. INTRODUÇÃO

Atualmente existem vários custos para utilização regular de um automóvel no Brasil. A condição de deterioração de grande parte das estradas (fator que gera necessidade de maiores gastos com manutenção) fica explícita em estudo realizado pela Confederação Nacional do Transporte (CNT) (CONFEDERAÇÃO NACIONAL DO TRANSPORTE, 2013), no qual foi identificado que 63,8% do total de rodovias pesquisadas apresentavam problemas no pavimento, na sinalização ou na geometria da via. Também existem os impostos obrigatórios e o crescente valor dos combustíveis que estão listados entre as causas de despesas dos motoristas.

Apenas para exemplificar este último item, segundo a Agência Nacional de Petróleo (ANP) (AGÊNCIA NACIONAL DE PETRÓLEO, 2014a), o preço médio da gasolina, principal combustível consumido no Brasil, apresentou um aumento anual médio de 3,97% entre 2011 e 2014. Já o etanol, outro combustível consumido no país, apresenta uma média de aumento ainda maior, alcançando 8,62% anuais no mesmo período.

No mesmo contexto, existem também os desafios da falta de mobilidade urbana, realidade de grande parte dos polos urbanos no Brasil, e da degradação do meio ambiente.

O consumo de energia de fontes fósseis das sociedades modernas, fruto de um modo de vida pouco sustentável, vem provocando significativos impactos ambientais. Além da degradação ambiental, provocada por acidentes ambientais relacionados ao petróleo, o consumo dessas fontes provoca alteração na concentração atmosférica de GEE (Gases de Efeito Estufa). Tais alterações de origem antrópica têm provocado as chamadas mudanças climáticas. (MCT, 2010 apud AZEVEDO, 2013, p.2).

Essa conjuntura de fatores econômicos, sociais e ambientais estimula as pessoas a buscarem novas alternativas de locomoção. A carona solidária é uma alternativa que apresenta benefícios a todos os envolvidos e, mesmo que indiretamente, à sociedade. Um benefício incentivador aos praticantes da solução de Carona Solidária é a redução de custos, principalmente de combustíveis, se divididos entre os participantes.

É possível também aplicar um sistema de revezamento de veículos entre grupos, o que reduz a deterioração dos automóveis e, por consequência, contribui com a diminuição de posteriores gastos com a manutenção de cada veículo. Existem grupos de usuários em redes sociais como o Caronas Blumenau-Floripa (FACEBOOK, 2014b) e Caronas Blumenau

(FACEBOOK, 2014a), os quais somados ultrapassam os 4.400 usuários. Recentemente foi criado também na rede social o grupo Caronas FURB, visando auxiliar o deslocamento à instituição, o qual já ultrapassou os 800 membros (FACEBOOK, 2014c). O Facebook, porém, não possui implementadas em seus grupos algumas praticidades que facilitariam a viabilidade de um sistema de caronas funcional, como a busca por georeferência. Visando auxiliar as pessoas interessadas em participar desse sistema de caronas, propõe-se neste trabalho o desenvolvimento de um aplicativo móvel que facilite a disponibilização e obtenção de caronas.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo geral do trabalho é apresentar o desenvolvimento de um aplicativo móvel que facilite a oferta e obtenção de caronas.

Os objetivos específicos do trabalho são:

- a) permitir aos usuários o acompanhamento de caronas em tempo real;
- b) utilizar as interfaces de programação fornecidas pela plataforma Android para obtenção das coordenadas do aparelho e processamento de georeferências.

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em quatro capítulos que estão descritos a seguir.

No primeiro capítulo tem-se a introdução ao tema principal deste trabalho com a apresentação da justificativa e dos objetivos.

No segundo capítulo apresenta-se a fundamentação teórica pesquisada sobre o conceito de carona, a questão ambiental, a legislação brasileira relacionada a caronas, a Plataforma Android, Computação na Nuvem, Banco de dados não relacional, Google *Cloud Endpoints* e Google *Directions* API, além dos trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento da aplicação, iniciando-se com o levantamento de informações, tendo na sequência a especificação, os requisitos, casos de uso, modelo entidade-relacionamento, implementação, técnicas e ferramentas utilizadas, operacionalidade da implementação, além de resultados e discussão.

No quarto capítulo tem-se as conclusões deste trabalho bem como apresentam-se sugestões para trabalhos futuros.

## 2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos a serem apresentados nas seções a seguir sobre o conceito de carona, a questão ambiental, a legislação brasileira relacionada a caronas, a plataforma android, computação na nuvem, banco de dados não relacional, Google *Cloud Endpoints* e Google *Directions API*, além de trabalhos correlatos.

### 2.1 CARONA

O dicionário Michaelis (2008, p. 166) define carona como “condução ou transporte gratuito em qualquer veículo”. Recentemente também surgiu o termo Carona Solidária, que segundo a Gazeta do Povo (2012) “[...] consiste em fazer com que duas ou mais pessoas que percorrem trajetos iguais ou parecidos usem o mesmo veículo, dividindo os custos”.

Segundo Grava (2002), para que um sistema de caronas funcione bem, é necessário que alguns fatores sejam atendidos. Dentre eles destacam-se:

- a) destinos agrupados preferencialmente em um único local. As moradias também devem ser próximas;
- b) horários de trabalho regulares, com horas extras e deslocamentos para locais diferentes ocorrendo regularmente;
- c) tratamento preferencial no sistema viário para as caronas, como o uso de faixas exclusivas para veículos com mais de um ocupante (High-Occupancy Vehicle - HOV) e vagas especiais;
- d) caronas devem ser aplicadas em deslocamento para o trabalho ou outras grandes instituições com frequentadores regulares (como universidades e centros médicos).

Alguns países com alta taxa de ocupação das rodovias em horários de pico já adotaram o sistema de vias exclusivas, para veículos de alta ocupação (normalmente mais de 2 passageiros). Segundo o Texas Department of Transportation (2014), somente neste estado americano existem 7 rodovias com faixas exclusivas para veículos de alta ocupação, um benefício que estimula a prática de caronas. Outro ponto citado por Grava (2002), a carona dentro de instituições é uma prática adotada inclusive no Brasil. O site Caronetas (2014a) cita que empresas como o Banco do Brasil, AON e Abril já aderiram à modalidade.



## 2.2 A QUESTÃO AMBIENTAL

Um das maiores preocupações da sociedade atualmente é em relação à sustentabilidade ecológica. Para Almeida (2007, p. 15), “[...] as atividades humanas estão utilizando os serviços ambientais num ritmo tal que já não é mais garantida a capacidade dos ecossistemas de atenderem às necessidades de futuras gerações.” A preocupação das nações com essas questões fica evidente ao resultar em diversas conferências e convenções mundiais voltadas à preservação e manutenção do meio ambiente, como a conferência de Estocolmo, em 1972, a ECO-92, realizada no Rio de Janeiro, e o protocolo de Kyoto, iniciado em 1997. A seguir é citado um trecho da Carta da Terra, um dos documentos resultantes da ECO-92:

A humanidade é parte de um vasto universo em evolução. A Terra, nosso lar, está viva com uma comunidade de vida única. As forças da natureza fazem da existência uma aventura exigente e incerta, mas a Terra providenciou as condições essenciais para a evolução da vida. A capacidade de recuperação da comunidade da vida e o bem-estar da humanidade dependem da preservação de uma biosfera saudável com todos seus sistemas ecológicos, uma rica variedade de plantas e animais, solos férteis, águas puras e ar limpo. O meio ambiente global com seus recursos finitos é uma preocupação comum de todas as pessoas. A proteção da vitalidade, diversidade e beleza da Terra é um dever sagrado. (MINISTÉRIO DO MEIO AMBIENTE, 2014, p. 1).

Marcovitch (2006, p.36) lembra que o consumo em grande escala de materiais da natureza, o qual remete à Revolução Industrial, compromete o equilíbrio do ciclo natural do carbono nos oceanos, plantas, terra e atmosfera. Segundo o autor, “as emissões antrópicas de CO<sub>2</sub> desequilibram este ciclo natural e aquecem o planeta numa escala indesejável”. Essa emissão de gases na atmosfera é citada por Almeida (2007, p.22) como um dos causadores do efeito estufa e, por consequência, do aquecimento global.

### 2.2.1 RELAÇÃO DA INFRAESTRUTURA GLOBAL DE TRANSPORTES COM O MEIO AMBIENTE

A utilização de equipamentos para transportes representa um grande conforto na vida humana. Porém, para Almeida (2007, p. 33), “Seja qual for a hipótese analisada [...], a conclusão é sempre a mesma: o setor de transporte não é sustentável.” Segundo o estudo realizados pela Internacional Agency of Energy (IEA) (INTERNACIONAL AGENCY OF ENERGY, 2012, p.15), o setor de transportes é responsável por aproximadamente 23% das emissões de monóxido de carbono mundiais.

A frota nacional brasileira, conforme relatório do Departamento Nacional de Trânsito (Denatran), totaliza mais de 84 milhões de veículos (DEPARTAMENTO NACIONAL DE TRÂNSITO, 2014). Em 2004 esse número não atingia os 40 milhões. Segundo estudo da ANP (AGÊNCIA NACIONAL DE PETRÓLEO, 2014b), o Óleo Diesel e a Gasolina Tipo C são os combustíveis mais vendidos no Brasil, ultrapassando os 214 milhões e 131 milhões de galões vendidos apenas em 2014, respectivamente. O Gás Natural Veicular vem atrás com o equivalente a mais de 34 milhões de galões vendidos em 2014, e o Etanol possui vendas que superam os 25 milhões de galões neste ano. A queima de combustíveis fósseis libera dióxido de carbono na atmosfera, o qual Almeida (2007, p.22) cita como o principal gás responsável pelo aquecimento global.

Existem diversas alternativas para a diminuição da emissão de gases na atmosfera no âmbito dos transportes. Soluções tecnológicas como veículos híbridos e de alta eficiência energética, além de soluções como aumento do uso de transporte ferroviário e incentivo de caronas. Tais soluções, porém, por vezes esbarram em interesses políticos e econômicos. Almeida (2007, p.38) cita o caso dos Estados Unidos, onde “[...] O congresso bloqueia qualquer nova proposta de controle dos gases de efeito estufa e o Executivo não se empenha em adotar medidas de desestímulo ao consumo de combustíveis fósseis.” Em novembro de 2014, os Estados Unidos da América e a China assinaram um acordo que pretende diminuir a emissão de gases até 2030 (G1, 2014).

## 2.3 LEGISLAÇÃO BRASILEIRA

De acordo com a Lei Nº 12.468, de 26 de agosto de 2011 (BRASIL, 2011), fica restrito aos taxistas o transporte remunerado de passageiros, considerando que “[...] É atividade privativa dos profissionais taxistas a utilização de veículo automotor, próprio ou de terceiros, para o transporte público individual remunerado de passageiros [...]” Apesar disso, o Artigo 736 do Código Civil, Lei 10406/02, de 10 de Janeiro de 2002 (BRASIL, 2002), considera que “Não se subordina às normas do contrato de transporte o feito gratuitamente, por amizade ou cortesia”. O Artigo 730 da mesma Lei 10406/02 define contrato de transporte como a situação em que “[...] alguém se obriga, mediante retribuição, a transportar, de um lugar para outro, pessoas ou coisas”.

Segundo o portal R7 (2014), a disponibilização de caronas pagas, que tem se propagado principalmente pela utilização de aplicativos móveis, vem causando uma “onda de descontentamento [...] em taxistas de todo o mundo, que acusam [...] de concorrência desleal e

de transportar passageiros ilegalmente”. Sendo assim, a carona sem visar lucros não apresenta complicações legais aos participantes: “Oferecer carona ao colega de trabalho ou a alguém por cortesia e sem visar o lucro é uma prática legal, prevista na lei [...]” (ÉPOCA NEGÓCIOS, 2014).

## 2.4 PLATAFORMA ANDROID

O Brasil é o quarto país no mundo em número de *smartphones* – cerca de 70 milhões – atrás apenas de China, Estados Unidos e Japão (EXAME, 2013). Dos *smartphones* no país, 85,1% utilizam o sistema operacional Android (OLHAR DIGITAL, 2014). Estudos apontam que, em todo o mundo, mais de 3 bilhões de pessoas – aproximadamente metade da população mundial – possuem um aparelho celular (LECHETA, 2010, p. 19).

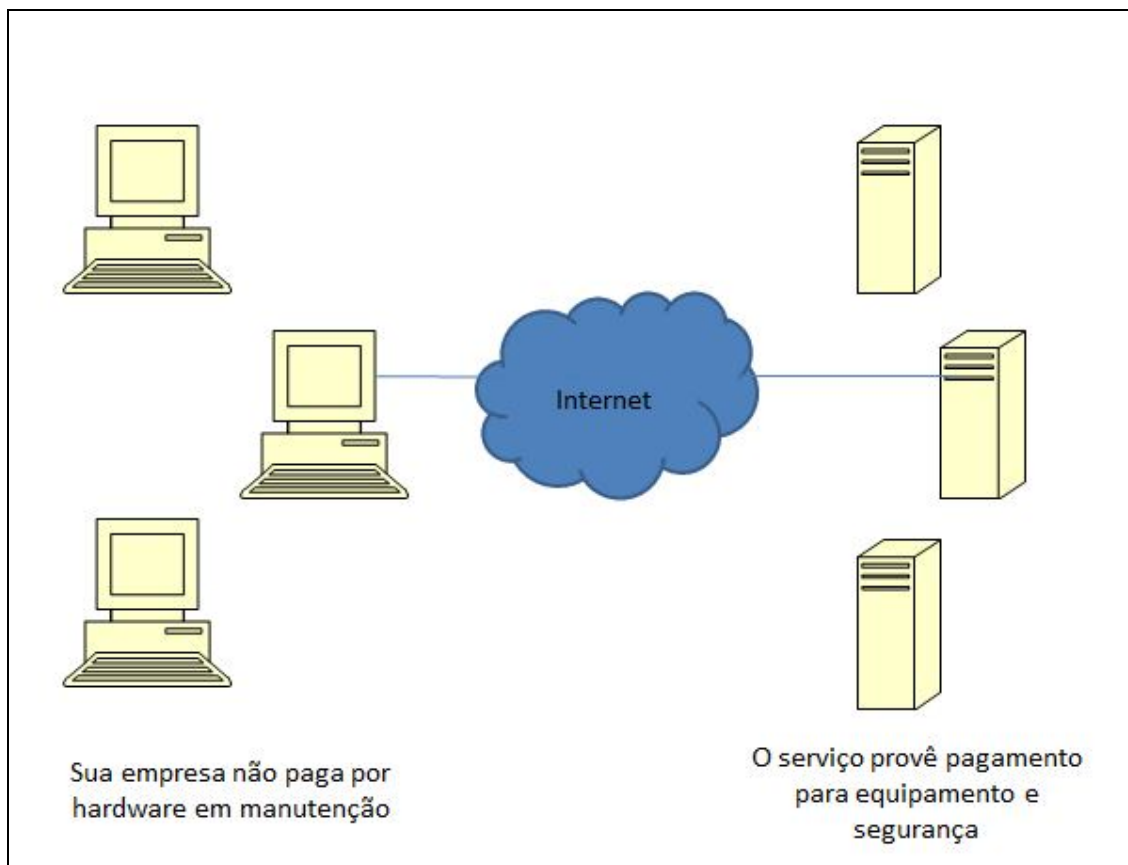
O Android é um sistema operacional com foco principal em dispositivos *mobile*. Segundo Lecheta (2010, p.20), “Consiste em uma [...] plataforma de desenvolvimento para aplicativos móveis, baseada no sistema operacional Linux, com [...] um ambiente de desenvolvimento bastante poderoso, ousado e flexível”. Diariamente mais de 1 milhão de novos dispositivos com o sistema operacional Android são ativados (ANDROID DEVELOPERS, 2014).

Aplicativos desenvolvidos para a plataforma Android são utilizados para os mais diversos fins. Lecheta (2012, p.16) cita que “Estamos na década da mobilidade, [...] e o mercado busca incessantemente [...] aplicativos comerciais e corporativos para os mais diversos setores, como varejo, saúde, economia, jogos e muito mais”. Conforme Android Developers (2014), mais de 1,5 bilhão de aplicativos e jogos são baixados mensalmente pelos usuários do Android.

## 2.5 COMPUTAÇÃO NA NUVEM

Segundo Sosinsky (2011, p.3), a computação na nuvem consiste em “[...] aplicações e serviços que rodam em redes distribuídas usando recursos virtualizados e acessados por protocolos comuns de internet e padrões de rede”. De acordo com Rittinghouse e Ransome (2010, p.26) “O termo nuvem é historicamente utilizado como uma metáfora para internet. Essa utilização deriva de sua comum representação em diagramas da internet como o esboço de uma nuvem”. Na Figura 1 é exibida a representação de um diagrama utilizando computação na nuvem.

Figura 1 - Diagrama de representação da nuvem



Fonte: Velte (2011).

A computação na nuvem, segundo Rittinghouse e Ransome (2010, p.13), apresenta benefícios como:

- a) custos de implementação e manutenção reduzidos;
- b) mobilidade aumentada para uma força de trabalho global;
- c) infraestruturas flexíveis e escaláveis;
- d) disponibilidade aumentada de aplicações de alta performance para pequenas e médias empresas.

Existem vários fornecedores que oferecem serviços de nuvem. Dentre eles destacam-se alguns, como a Amazon, com serviços como o Elastic Compute Cloud (EC2), Serviço Simples de Armazenamento (S3), a Google, com o GoogleApp Engine e a Microsoft, com serviços como o Windows Azure, Live Services e Microsoft SharePoint Services (VELTE, 2011, p.21).

## 2.6 BANCO DE DADOS NÃO RELACIONAL

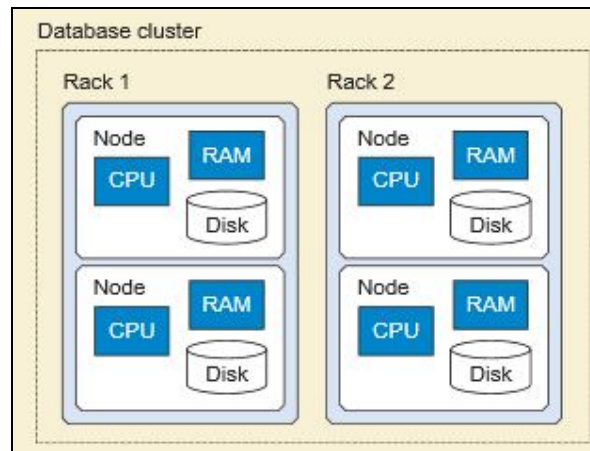
Segundo Sadalage (2013, p.36), os bancos de dados não relacionais “[...] atuam sem um esquema, permitindo que sejam adicionados, livremente, campos aos registros do banco de dados, sem ter de definir primeiro quaisquer mudanças na estrutura”. Também conhecidos por NoSQL, os bancos de dados não relacionais, segundo Hurwitz (2013, p.112) “[...] não dependem do modelo de tabela/chave endêmico aos bancos de dados relacionais”.

Sadalage (2013, p.39) relacionam as principais características dos bancos de dados não relacionais:

- a) não utilizam o modelo relacional;
- b) tem uma boa execução em *clusters*;
- c) seu código é aberto (*open source*);
- d) são criados para propriedades na *web* do século XXI;
- e) não têm esquema.

Algumas das vantagens na utilização de bancos de dados NoSQL são a facilidade nas alterações de estruturas mesmo antes do fim da modelagem dos dados, a facilidade de aumento de performance, adição de nós ao *cluster*, e a possibilidade de armazenamento de dados de grande variabilidade. A indústria do NoSQL geralmente utiliza o conceito de *clusters* como nodos de processamento (MCCREARY; KELLY 2013, p.20), conforme exibido na Figura 2.

Figura 2 - Esquema de cluster com mais de um rack



Fonte: McCreary e Kelly (2014, p.20).

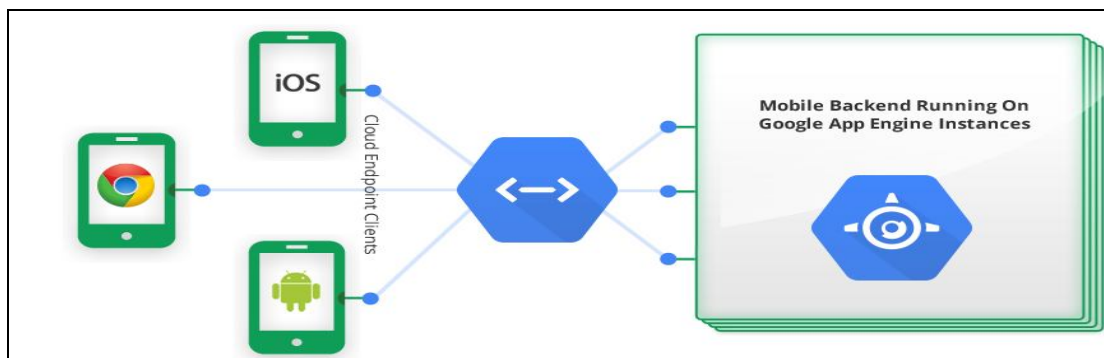
### 2.6.1 Google cloud endpoints

O Google *Cloud Endpoints* é uma ferramenta disponibilizada pela Google que permite o armazenamento de dados de uma aplicação na nuvem. Consiste em ferramentas, bibliotecas e recursos que permitem a geração de APIs e bibliotecas de cliente de uma aplicação App Engine, referidos como um *backend* API, para simplificar o acesso do cliente aos dados de outros aplicativos. *Endpoints* visa facilitar a criação de um *backend web* para os clientes da *web* e clientes móveis, como o Android ou o iOS, da Apple (GOOGLE CLOUD, 2015).

O processo de desenvolvimento utilizando o Google *Cloud Endpoints* envolve (GOOGLE CLOUD, 2015):

- a) criar o *backend* da API do projeto;
- b) anotar o código do *backend* da API, para que as classes e bibliotecas do cliente possam ser criadas a partir dele;
- c) gerar a biblioteca do cliente;
- d) escrever o código do aplicativo cliente, usando a biblioteca do cliente quando este faz chamadas ao *backend* da API.

O Google *Cloud Endpoints* é compatível com as plataformas Android e IOS, além de JavaScript. Na Figura 3 é mostrada a arquitetura básica do Google *Cloud Endpoints*, exibindo a possibilidade de comunicação com o *backend* a partir de diferentes plataformas.

Figura 3 - Arquitetura básica do Google *Cloud Endpoints*

Fonte: Google (2015).

## 2.7 GOOGLE DIRECTIONS API

O Google *Directions* API é um serviço que calcula as direções entre locais usando uma solicitação HTTP. É possível localizar instruções para vários modos de transporte, trânsito, dirigindo, a pé ou de bicicleta. O *Directions* pode receber dados como origem, destinos e pontos de passagem, tanto como sequências de texto ou como coordenadas de latitude/longitude (GOOGLE DEVELOPERS, 2015).

Usuários da plataforma grátis possuem algumas restrições como um máximo de 2500 requisições diárias, máximo de 8 pontos de passagem e 2 requisições por segundo. O *Directions* também aceita URLs com um comprimento máximo de 2000 caracteres (GOOGLE DEVELOPERS, 2015). Na Figura 4 é exibido o modelo de uma URL aceita pela API e um trecho do retorno em texto.

Figura 4 - URL de requisição à API e retorno em texto

```

https://maps.googleapis.com/maps/api/directions/json?origin=Blumenau&destination=Brusque

{
  "routes" : [
    {
      "bounds" : {
        "northeast" : {
          "lat" : -26.8947246,
          "lng" : -48.8801448
        },
        "southwest" : {
          "lat" : -27.099274,
          "lng" : -49.0717429
        }
      },
      "copyrights" : "Dados cartográficos ©2015 Google",
      "legs" : [
        {
          "distance" : {
            "text" : "42,1 km",
            "value" : 42101
          },
          "duration" : {
            "text" : "58 minutos",
            "value" : 3489
          },
          "end_address" : "Brusque - SC, Brasil",
          "end_location" : {
            "lat" : -27.098241,
            "lng" : -48.9111096
          },
          "start_address" : "Blumenau - SC, Brasil",
          "start_location" : {
            "lat" : -26.9166279,
            "lng" : -49.0717429
          },
          "steps" : [
            {
              "distance" : {
                "text" : "1,9 km",
                "value" : 1899
              },
              "duration" : {
                "text" : "4 minutos",

```

Fonte: Google (2015).

## 2.8 TRABALHOS CORRELATOS

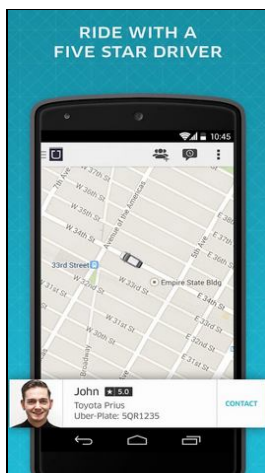
Alguns projetos de caronas se transformaram em iniciativas bem sucedidas e se tornaram realidade em outros países e também no Brasil. A seguir estão descritas algumas ferramentas de caronas que possuem números significativos de usuários.

O Caronetás é um site brasileiro que integra colaboradores de empresas e centros empresariais. Ele utiliza uma moeda virtual chamada caronetás para efetuar o pagamento das caronas. O site também possui funcionalidades para compartilhamento de táxi e bicicleta (CARONETAS, 2014b).



Existem também algumas ferramentas comerciais para auxílio na busca de caronas. O Uber é uma das principais existentes. Conforme o site do Uber (UBER, 2014), para que um usuário possa ofertar caronas utilizando o aplicativo, é necessário que ele realize um cadastro prévio com a empresa, que irá checar antecedentes criminais, informações da carteira de motorista e do veículo. Além disso, também segundo o site, existe uma taxa de pagamento pré determinada (da qual o Uber retira sua porcentagem), calculada com base na distância e tempo do trajeto, somados a uma tarifa de base. No Brasil, o serviço está apenas disponível nas cidades do Rio de Janeiro, Belo Horizonte, São Paulo e Brasília. Na Figura 5 pode ser vista uma tela do aplicativo.

Figura 5 - Tela do Aplicativo Uber para Android



Fonte: Google Play Store (2015).

### 3. DESENVOLVIMENTO DO APLICATIVO E DO *BACKEND*

Neste capítulo estão descritas particularidades técnicas do aplicativo e do *backend* desenvolvidos, tais como o levantamento de informações, a especificação com a apresentação dos requisitos funcionais e não funcionais, os diagramas de atividades, o diagrama de casos de uso e suas descrições. Na sequência tem-se a operacionalidade do aplicativo e do sistema, bem como são apresentados os resultados e discussões.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

Com base nos estudos realizados considerando os trabalhos correlatos já existentes e as necessidades apresentadas, foi desenvolvido um aplicativo para a plataforma Android que permite a localização de usuários que percorram trajetos semelhantes.

O aplicativo permite o cadastro de um caminho a ser percorrido, determinando o trajeto com base na origem e destino inseridos. Os pontos do trajeto são inseridos no banco de dados, no formato latitude/longitude.

O sistema também possibilita a busca por caronas já cadastradas e que percorram um trajeto igual ou que contenham o trajeto desejado. Para essa busca são percorridos os pontos das caronas já cadastradas e verificada a compatibilidade das latitudes e longitudes para determinar a proximidade.

#### 3.2 ESPECIFICAÇÃO

Nesta seção é apresentada a especificação do aplicativo e também do *backend*, acessado para persistência de dados. Para a modelagem utilizou-se da ferramenta Enterprise Architect. Utiliza-se ainda a notação *Unified Modeling Language* (UML) para a representação do diagrama de casos de uso.

##### 3.2.1 Requisitos Funcionais e Não Funcionais

O Quadro 1 apresenta os requisitos funcionais previstos para o aplicativo e sua rastreabilidade, ou seja, vinculação com o caso de uso associado.

Quadro 1 - Requisitos funcionais

<b>Requisitos Funcionais</b>	<b>Caso de Uso</b>
RF01: O sistema deverá permitir cadastro de usuários.	UC01
RF02: O sistema deverá permitir o cadastro de viagem.	UC02
RF03: O sistema deverá identificar e notificar usuários com trajetos rotineiros semelhantes.	UC03
RF04: O sistema deve permitir a busca de viagens já cadastradas.	UC04
RF05: O sistema deverá permitir ao usuário efetuar <i>login</i> .	UC05
RF06: O sistema deverá permitir que o administrador visualize estatísticas por meio de uma interface <i>web</i> .	UC06

O Quadro 2 lista os requisitos não funcionais previstos para o sistema.

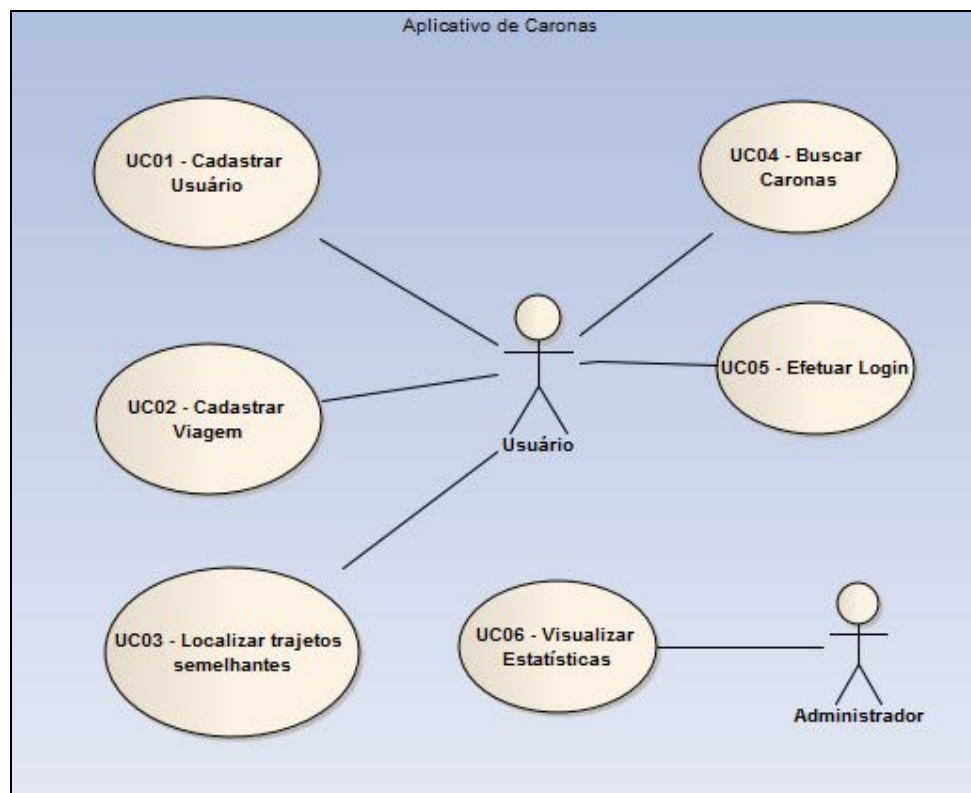
Quadro 2 - Requisitos não funcionais

<b>Requisitos Não Funcionais</b>
RNF01: O aplicativo mobile deverá ser desenvolvido na plataforma Android sendo compatível a versão 4.0 ou superior
RNF02: O sistema deverá utilizar o Google <i>Cloud Endpoints</i> para persistência de dados.
RNF03: O sistema deverá utilizar o Google <i>Directions API</i> para obtenção dos trajetos.
RNF04: O aplicativo mobile deverá ser desenvolvido na linguagem Java, SDK versão 8.0.25.18 para a plataforma Android.

### 3.2.2 Casos de Uso

Esta sub-seção apresenta na Figura 6 o Diagrama de Casos de Uso do sistema desenvolvido. Para desenvolver o diagrama foi utilizada a ferramenta Enterprise Architect (EA). A descrição expandida dos principais casos de uso é apresentada no Apêndice A.

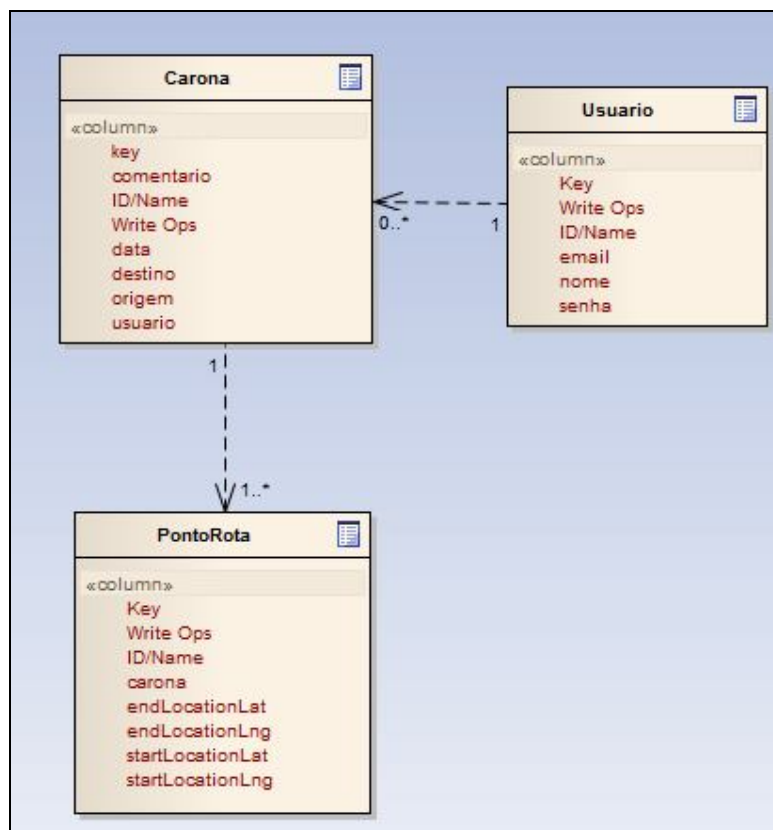
Figura 6 - Diagrama de Casos de Uso



### 3.2.3 Modelo Entidade-Relacionamento

A Figura 7 mostra o Modelo Entidade-Relacionamento (MER) do sistema. O Dicionário de Dados está apresentado no Apêndice B.

Figura 7 - Modelo Entidade-Relacionamento



O sistema armazena os dados por meio da API Google *Cloud Endpoints*.

A tabela *Usuario* armazena informações dos usuários que fazem uso do sistema. Essa tabela é referenciada para identificação do usuário que cadastrou uma carona e também para *login* no sistema.

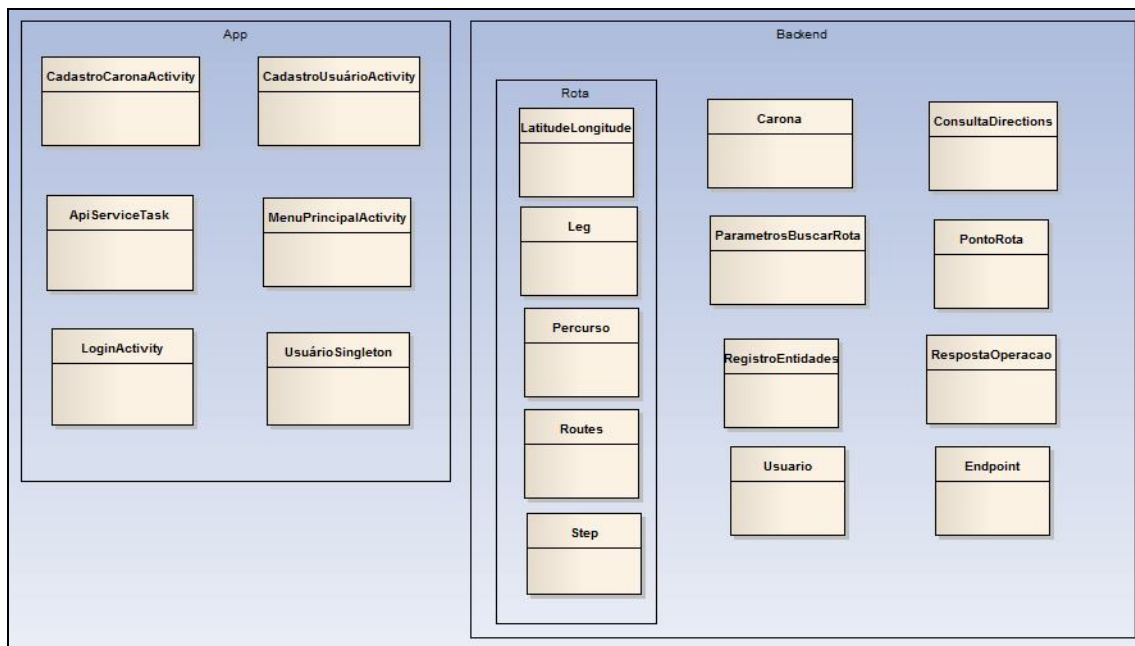
A tabela *Carona* armazena informações das caronas cadastradas no sistema. Estas informações serão utilizadas para exibir a outros usuários informações de caronas disponíveis.

A tabela *PontoRota* armazena latitude e longitude dos pontos contidos no trajeto de cada carona. Estas informações são utilizadas para determinar se o trajeto percorrido por outro usuário está contido (ou aproximadamente contido) em alguns dos trajetos já cadastrados.

### 3.2.4 Classes

A Figura 8 exibe as classes existentes no sistema desenvolvido e seus respectivos pacotes.

Figura 8 - Classes e pacotes



As classes `CadastroCaronaActivity` e `CadastroUsuarioActivity` apresentam formulários com campos para cadastros. Além disso, a classe de cadastro de carona também possui um botão que permite buscar as caronas já existentes.

A classe `ApiServiceTask` é responsável pela conexão com o banco de dados.

A classe `LoginActivity` apresenta um formulário para inserção dos dados do usuário, que serão validados.

A classe `MenuPrincipalActivity` exibe os botões que permitem acessar as telas de cadastro de caronas e exibir caronas.

A classe `UsuarioSingleton` é utilizada para manter o usuário da sessão, que será utilizado ao cadastrar uma nova carona.

As classes `Leg`, `Routes`, `Step` e `Percurso` são utilizadas para leitura dos dados fornecidos pela Google *Directions* API. A classe `LatitudeLongitude` possibilita a criação de um objeto com essas duas informações.

As classes *Usuario*, *Carona* e *PontoRota* são utilizadas para criação dos objetos que serão salvos no banco de dados.

A classe *ConsultaDirections* é utilizada para buscar as informações da API *Google Directions*.

A classe *ParametrosBuscarRota* é utilizada para criação de um objeto com os parâmetros para a busca de rotas na API.

A classe *RegistraEntidades* registra as entidades no banco de dados.

A classe *RespostaOperação* retorna o status da operação após a comunicação com o banco de dados.

A classe *Endpoint* é responsável pela criação e validação de usuário, além de criação e busca de carona.

### 3.3 IMPLEMENTAÇÃO

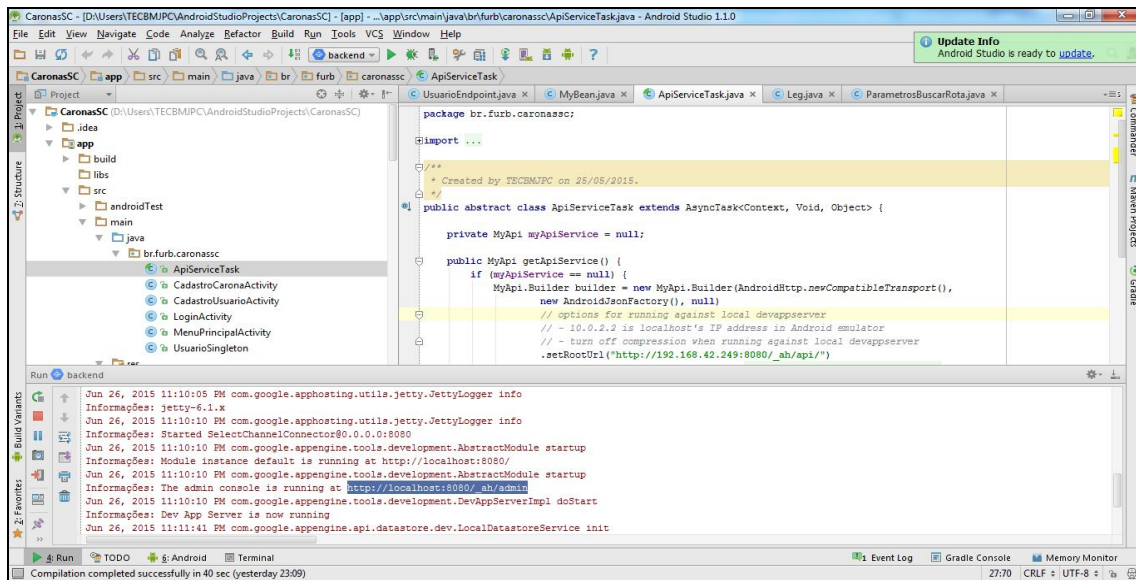
A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

#### 3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Para o desenvolvimento do aplicativo foi utilizada a *Integrated Development Environment* (IDE) *Android Studio*, versão 1.1.0. Esta ferramenta proporciona funcionalidades, ferramentas e recursos voltados especialmente ao desenvolvimento de aplicativos para a plataforma *Android*. A linguagem utilizada para o desenvolvimento do aplicativo foi o *Java*.

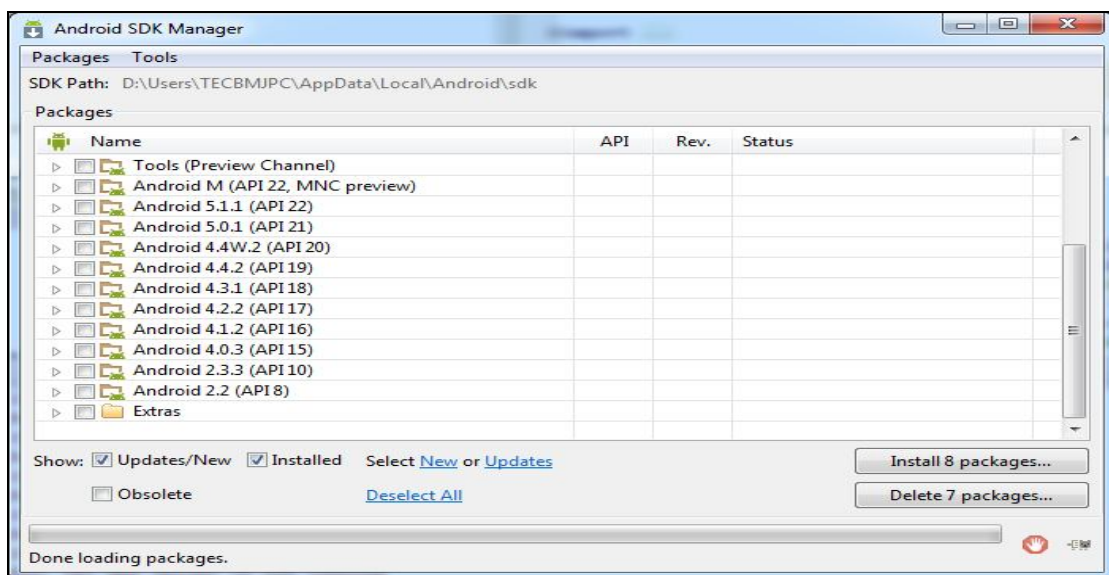
Na Figura 9 é exibida a tela inicial do *Android Studio*.

Figura 9 - Tela Inicial da IDE Android Studio



Para o desenvolvimento do aplicativo foi realizado o *download* das *Applications Programming Interface* (APIs) de números 15,16,17,18, 19, 20, 21 e 22 através da ferramenta Android SDK Manager, disponível no Android Studio. Isto permite a compatibilidade com a versão 4.0.3 do Android ou superior, conforme exibido na Figura 10.

Figura 10 - Android SDK Manager





A execução e depuração da aplicação Android foi realizada utilizando o *smartphone* Motorola Moto G 2, possuindo o sistema operacional Android na versão 5.0.2 e também no emulador virtual do Android, existente no Android Studio.

A aplicação foi estruturada em pacotes, separando as classes do *layout*, da construção do App e a do *Backend*. A Figura 11 exibe a estrutura das classes Java do aplicativo. Na Figura 12 estão os arquivos *eXtensible Markup Language* (XML) contendo as informações do *layout* das telas. Finalmente, na Figura 13 estão as classes java do *backend*.

Figura 11 - Estrutura de Classes Java do App

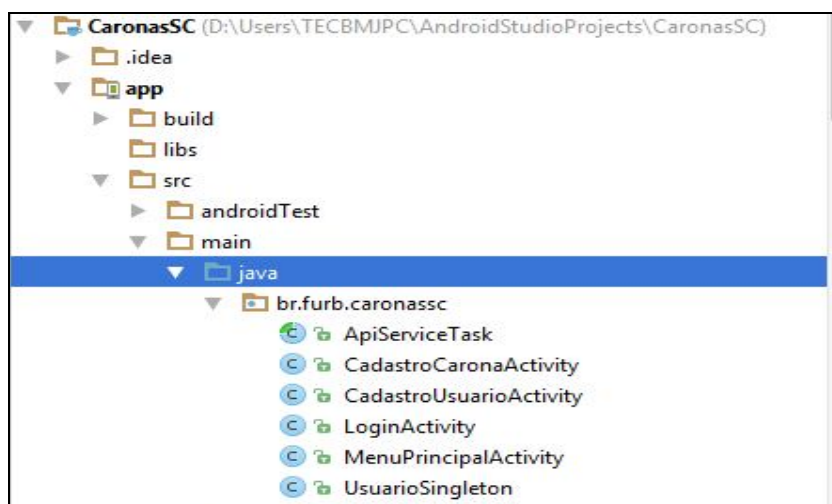


Figura 12 - Estrutura de Classes de *Layout* XML

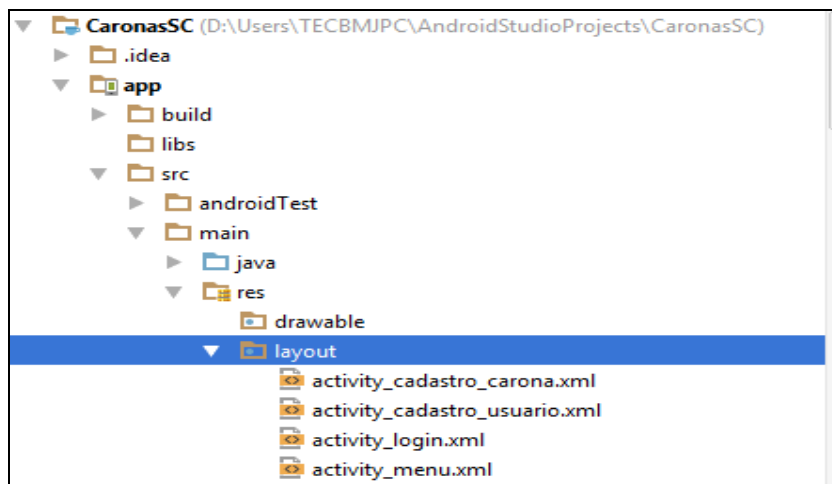
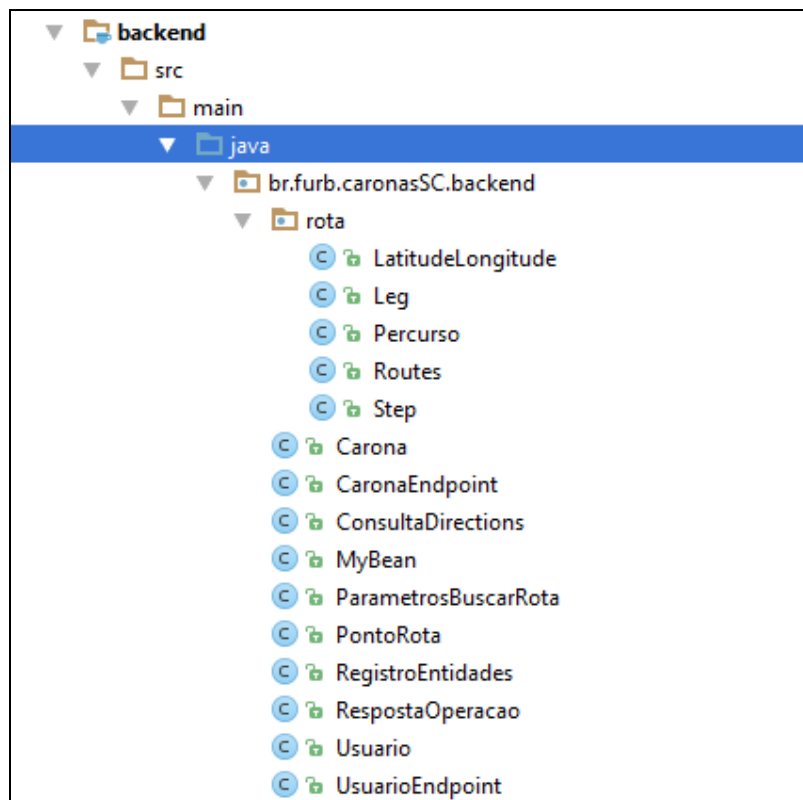


Figura 13 - Estrutura de Classes Java do *Backend*

### 3.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Nesta subseção serão exibidas as principais telas e funcionalidades do sistema desenvolvido.

Na Figura 14 é exibida a tela inicial do aplicativo, que permite ao usuário efetuar o *login* no sistema ou criar uma nova conta.

Figura 14 - Tela de *Login* do Aplicativo

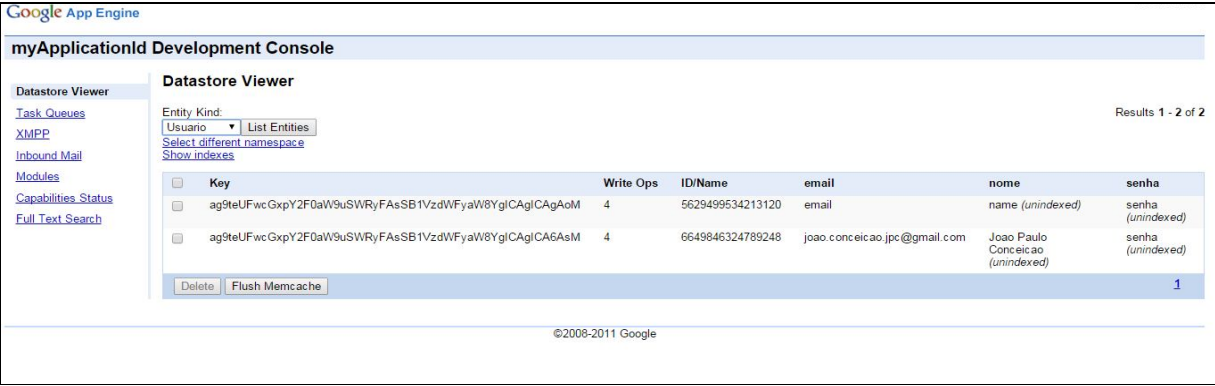
Ao selecionar a opção Criar Cadastro, o usuário visualiza uma tela para preenchimento de seus dados básicos, conforme mostra a Figura 15.

Figura 15 - Tela de Cadastro de Usuário



Após realizar o cadastro, as informações são salvas no banco de dados e o cadastro do usuário já pode ser visualizado no *Development Console*, que permite a visualização das informações de todas as entidades contidas no banco de dados, conforme exibido na Figura 16.

Figura 16 - Entidade usuário visualizada pelo administrador



Google App Engine  
myApplicationId Development Console

Datastore Viewer

Entity Kind: **Usuario** [List Entities](#)  
[Select different namespace](#)  
[Show indexes](#)

Results 1 - 2 of 2

<input type="checkbox"/>	Key	Write Ops	ID/Name	email	nome	senha
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFA5SB1VzdWFyaW8YglCAglCAgAoM	4	5629499534213120	email	name (unindexed)	senha (unindexed)
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFA5SB1VzdWFyaW8YglCAglCA6AsM	4	6649846324789248	joao.conceicao.jpc@gmail.com	Joao Paulo Conceicao (unindexed)	senha (unindexed)

[Delete](#) [Flush Memcache](#) 1

©2008-2011 Google

Para cadastrar uma carona, o usuário irá informar Origem e Destino, além da data e hora, obrigatoriamente, conforme exibido na Figura 17. Também é possível inserir algum comentário.

Figura 17 - Tela de Cadastro de Carona



Cadastrar Carona

Origem

Destino

Data

Hora

Comentario

BUSCAR CARONAS

CADASTRAR

No momento em que o usuário cadastra uma nova carona, o sistema irá localizar os pontos (latitude/longitude) do trajeto, por meio da *Google Directions API* e salvá-los no banco de dados. Na Figura 18, como um exemplo, estão listados os pontos da rota entre a Rua Amazonas, Blumenau, SC, e a Rua Nereu Ramos, também em Blumenau, SC.

Figura 18 - Pontos Rota de um trajeto

**Datastore Viewer**

Entity Kind: **PontoRota** [List Entities](#) [Select different namespace](#) [Show indexes](#)

Results 1 - 7 of 7

<input type="checkbox"/>	Key	Write Ops	ID/Name	carona	endLocationLat	endLocationLng	startLocationLat	startLocationLng
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFgsSCVBvbnRvUm90YRIAgICAgICAw	12	4521191813414912	5875790138834944	-26.9264624	-49.0631095	-26.9289639	-49.0641180
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFgsSCVBvbnRvUm90YRIAgICAgICAw	12	5031365208702976	5875790138834944	-26.9354833	-49.0673769	-26.9375576	-49.0654040
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFgsSCVBvbnRvUm90YRIAgICAgICAw	12	5312840185413632	5875790138834944	-26.9441766	-49.0718559	-26.9441812	-49.0717266
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFgsSCVBvbnRvUm90YRIAgICAgICAw	12	5594315162124288	5875790138834944	-26.9293094	-49.0626478	-26.9310896	-49.0632463
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFgsSCVBvbnRvUm90YRIAgICAgICAw	12	6157265115545600	5875790138834944	-26.9310896	-49.0632463	-26.9354833	-49.0673769
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFgsSCVBvbnRvUm90YRIAgICAgICAw	12	6438740092256256	5875790138834944	-26.9375576	-49.065404099999999	-26.9441766	-49.0718559
<input type="checkbox"/>	ag9teUFwcGxpY2F0aW9uSWRyFgsSCVBvbnRvUm90YRIAgICAgICAw	12	6720215068966912	5875790138834944	-26.9289639	-49.064118099999999	-26.9293094	-49.0626478

[Delete](#) [Flush Memcache](#) 1

©2008-2011 Google

Na Figura 19 está exibido o código da classe *ConsultaDirections*, que é responsável por enviar a requisição por uma URL à API e retornar o resultado da API em formato *String*.

Figura 19 - Classe *ConsultaDirections*

```
public class ConsultaDirections {
    public String getDirections(String origem, String destino) throws IOException {
        String retorno = "";
        StringBuilder resultado = new StringBuilder();

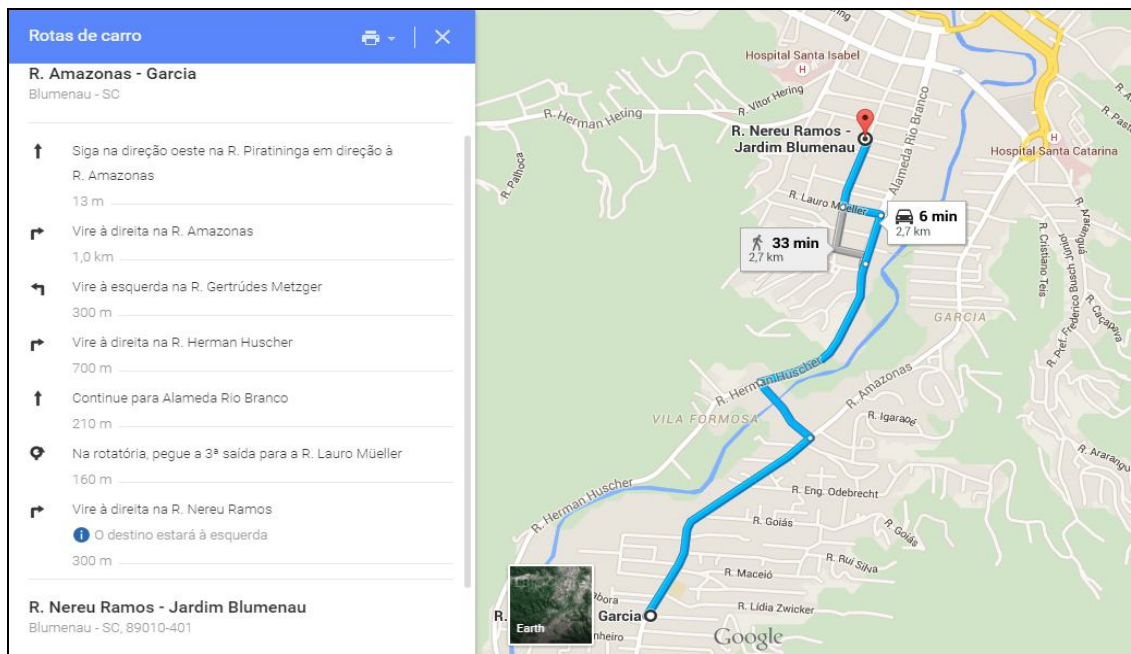
        URL url = new URL("https://maps.googleapis.com/maps/api/directions/json?origin=" + origem.replace(" ", "%20") + "&destination=" + destino.replace(" ", "%20"));
        BufferedReader reader = new BufferedReader(new InputStreamReader(url.openStream()));
        String line;

        while ((line = reader.readLine()) != null) {
            resultado.append(line);
        }
        reader.close();

        return resultado.toString();
    }
}
```

O usuário também tem a opção de buscar por caronas já cadastradas, que contenham o trajeto desejado. Na Figura 20 fica explícito visualmente o resultado retornado pelo Google Maps para o trajeto.

Figura 20 - Trajeto entre as ruas Amazonas e Nereu Ramos



Fonte: Google Maps (2015).

O caminho sugerido na Figura 19 contém as Ruas Gertrudes Metzger e Alameda Rio Branco. Neste cenário, um usuário que buscar por um trajeto contido neste percurso, visualizará a carona previamente cadastrada. Na Figura 21 é exibido o retorno para uma busca entre as ruas Amazonas e Alameda Rio Branco.

Figura 21 - Busca de Carona I

Cadastrar Carona

Rua Amazonas Blumenau

Alameda Rio Branco Blumenau

27/06/2015 18:00:13

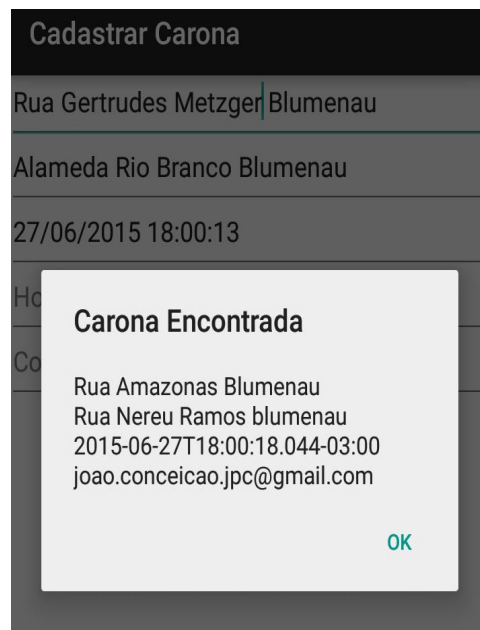
Carona Encontrada

Rua Amazonas Blumenau  
Rua Nereu Ramos Blumenau  
2015-06-27T18:00:18.044-03:00  
joao.conceicao.jpc@gmail.com

OK

Na Figura 22 é exibido o retorno para uma busca entre as ruas Gertrudes Metzger e Alameda Rio Branco, ambas contidas no trajeto cadastrado previamente.

Figura 22 - Busca de Carona II



A busca pelas caronas com trajeto semelhante ocorre na classe Endpoint, por meio do método `consultarCaronas()`. O código do método `consultarCaronas()` é exibido na Figura 23.

Figura 23 - Método consultarCaronas()

```

@ApiMethod(name = "consultarCaronas")
public Carona consultarCaronas(ParametrosBuscarRota parametrosBuscarRota) {
    try {

        ConsultaDirections consulta = new ConsultaDirections();
        String rota = consulta.getDirections(parametrosBuscarRota.getOrigem(), parametrosBuscarRota.getDestino());

        Percurso percurso = new Gson().fromJson(rota, Percurso.class);
        Leg leg = percurso.getRoutes().get(0).getLegs().get(0);
        LatitudeLongitude startLocation = leg.getStart_location();
        LatitudeLongitude endLocation = leg.getEnd_location();

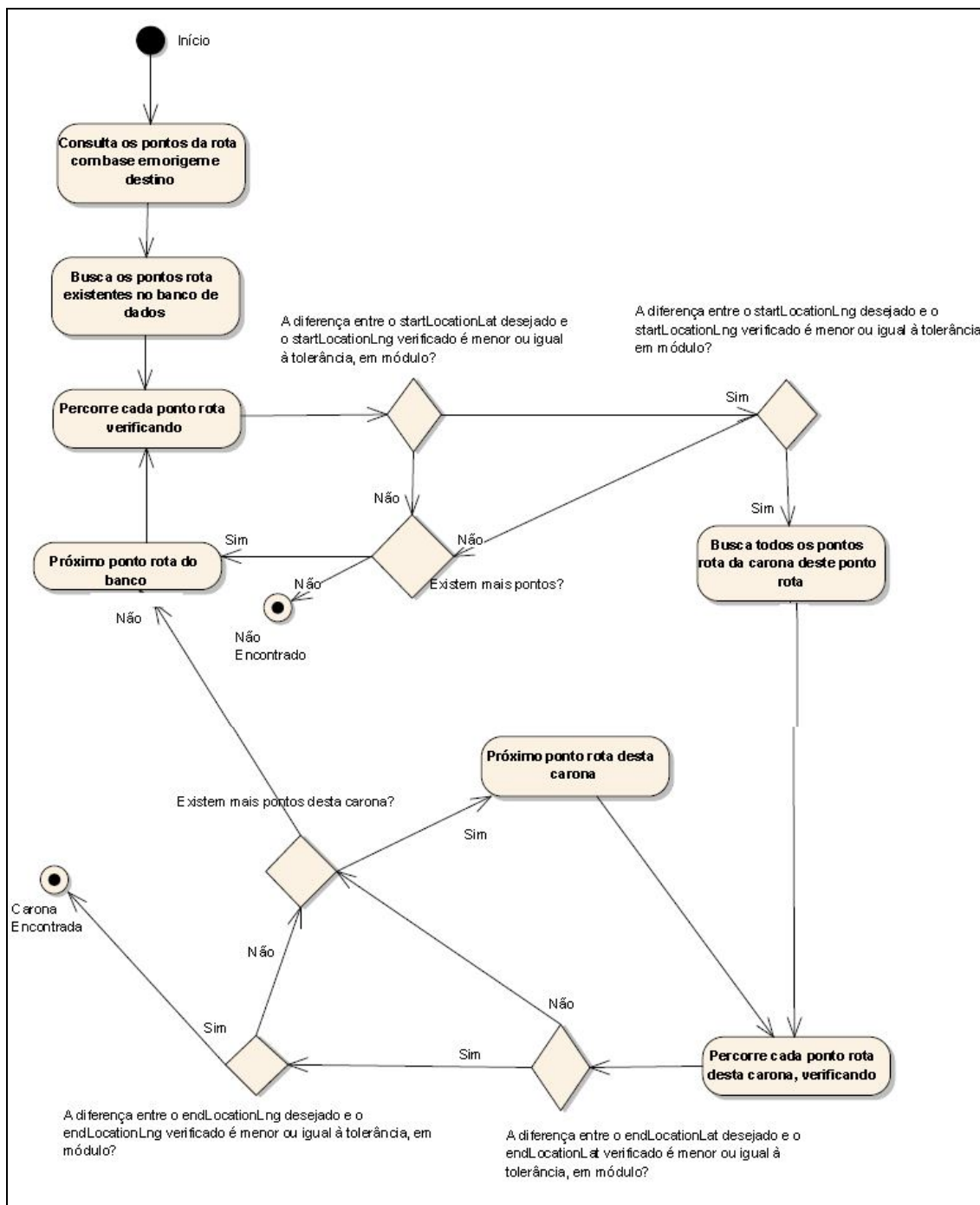
        List<PontoRota> pontosRotaBanco
            = ofy().load().type(PontoRota.class).list();
        for (PontoRota pontoRota : pontosRotaBanco) {
            if (Math.abs(pontoRota.getStartLocationLat() - startLocation.getLat()) <= tolerancia) {
                if (Math.abs(pontoRota.getStartLocationLng() - startLocation.getLng()) <= tolerancia) {
                    // List<PontoRota> pontosRotaOk
                    // = ofy().load().type(PontoRota.class).filter("Carona", pontoRota.getCarona()).list();
                    List<PontoRota> pontosNova = new ArrayList<>();
                    for (PontoRota pontoRota2 : pontosRotaBanco) {
                        if (pontoRota2.getCarona() == pontoRota.getCarona()) {
                            pontosNova.add(pontoRota2);
                        }
                    }
                    for (PontoRota pontoRotaOk : pontosNova) {
                        if (Math.abs(pontoRotaOk.getEndLocationLat() - endLocation.getLat()) <= tolerancia) {
                            if (Math.abs(pontoRotaOk.getEndLocationLng() - endLocation.getLng()) <= tolerancia) {
                                if (Math.abs(pontoRotaOk.getEndLocationLng() - endLocation.getLng()) <= tolerancia) {
                                    Carona carona = new Carona();
                                    List<Carona> caronas = ofy().load().type(Carona.class).list();
                                    for (Carona caronal : caronas) {
                                        if (caronal.getId() == pontoRotaOk.getCarona()) {
                                            carona = caronal;
                                            return carona;
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
        return null;
    }
    return null;
}

```

A Figura 24 exibe um fluxograma que define a lógica utilizada pelo método que consulta as caronas existentes com trajetos semelhantes.



Figura 24 - Fluxograma do Método consultarCaronas()



O método exibido na Figura 24 inicia buscando os pontos da rota com base na origem e no destino inseridos pelo usuário. Após isso, busca todos os pontos rota existentes no banco de dados.

Ele verifica então se a latitude e a longitude do ponto inicial da rota buscada pelo usuário estão contidos em algum ponto rota já cadastrado, considerando uma tolerância de 0,001 graus. Caso não exista nenhum ponto rota com estas informações, assume que não existe nenhuma carona cadastrada com o trajeto desejado.

Ao encontrar algum ponto rota que esteja dentro destas condições, são levantados todos os pontos rota desta mesma carona. É feita então uma comparação em cada ponto rota desta carona para verificar se a latitude e a longitude do ponto final do trajeto desejado pelo usuário estão contidas em algum deles, considerando a mesma tolerância de 0,001 graus. Caso não exista nenhum ponto rota nestas condições, retorna à verificação anterior do ponto inicial, enquanto houverem pontos rota.

Caso seja encontrada alguma carona que atenda a todas estas condições, é considerada a carona adequada para o trajeto inserido pelo usuário, pois contém ou é idêntica ao trajeto desejado.

### 3.4 RESULTADOS E DISCUSSÃO

O objetivo de deste trabalho foi desenvolver uma solução para facilitar a localização e obtenção de caronas em trajetos semelhantes. Este objetivo foi alcançado permitindo, por meio de APIs externas, localizar os pontos de trajetos e compará-los aos de outros trajetos.

Em comparação com os trabalhos correlatos, apresenta algumas vantagens como a de ser um aplicativo *mobile* e possuir disponibilidade para utilização em qualquer localidade com acesso à *internet*. Uma comparação entre este trabalho e os trabalhos correlatos é exibida no Quadro 3. O aplicativo desenvolvido não apresenta complicações legais por estimular apenas a carona solidária, prevista em lei, sem remuneração.

Quadro 3 - Comparação entre sistemas correlatos

	Aplicativo <i>Mobile</i>	Localidade no Brasil	Complicações Legais
Este trabalho	Sim	Qualquer	Não
Uber	Sim	Rio de Janeiro, São Paulo, Belo Horizonte e Brasília	Sim
Caronetas	Não	Qualquer	Não

## 4. CONCLUSÕES

Ao término do desenvolvimento do sistema de caronas, é realizada uma avaliação dos objetivos estabelecidos no início. Foi alcançada a premissa de permitir o acompanhamento em tempo real de caronas. A obtenção da geolocalização é realizada por meio de uma API e o acompanhamento pela interface *web* pode ser realizado por meio do *Google Developer Console*.

A principal limitação encontrada foi com relação aos pontos (latitude e longitude) retornados pela API do Google. O espaçamento entre esses pontos dificulta a comparação entre os caminhos e é necessário utilizar uma margem de tolerância para corrigir este problema parcialmente.

O aplicativo foi desenvolvido para o sistema operacional Android utilizando a linguagem de programação Java, juntamente com as bibliotecas para auxiliar o desenvolvimento, como a biblioteca de *API Servlet*, por exemplo, além de APIs externas como o *Google Directions*. Para a persistência dos dados foi utilizada a ferramenta *NoSQL Google Cloud Endpoints*.

Diante dos resultados apresentados, os benefícios apresentados aos usuários são avaliados positivamente, sendo o principal deles possibilitar o encontro de trajetos semelhantes, facilitando assim a viabilização de um sistema de caronas eficiente.

### 4.1 EXTENSÕES

Para trabalhos futuros, existe a possibilidade de aperfeiçoamento do aplicativo e do método de buscas. Permitir-se que o usuário selecione o trajeto em um mapa, obtendo as coordenadas a partir desse.

Também seria interessante aprimorar o cadastro de usuários, adicionando funções como avaliações após as caronas, cadastro de veículos e, num âmbito mais complexo, consulta de informações do motorista ou do veículo junto ao departamento de trânsito. Outra função que pode ser adicionada é a de cálculo dos gastos com combustível.

Ainda como extensão, pode ser avaliado o impacto da implantação de um sistema de caronas funcional, calculando as diminuições nas emissões de gases e as melhorias de mobilidade urbana.

## REFERÊNCIAS

AGÊNCIA NACIONAL DE PETRÓLEO. **Resumo Mensal**. [S.l.], 2014a. Disponível em: <[http://www.anp.gov.br/preco/prc/resumo\\_mensal\\_index.asp](http://www.anp.gov.br/preco/prc/resumo_mensal_index.asp)>. Acesso em: 20 ago. 2014.

AGÊNCIA NACIONAL DE PETRÓLEO. **Vendas, pelas Distribuidoras, dos Derivados Combustíveis de Petróleo (barris equivalentes de petróleo)**. [S.l.], 2014b. Disponível em: <[www.anp.gov.br/?dw=11049](http://www.anp.gov.br/?dw=11049)>. Acesso em: 06 set. 2014.

ALMEIDA, Fernando. **Os desafios da sustentabilidade**. Rio de Janeiro: Elsevier, 2007.

ANDROID DEVELOPERS. **About**. [S.l.], 2014. Disponível em: <<http://developer.android.com/about/index.html>>. Acesso em: 20 ago. 2014.

AZEVEDO, Valdir Braz. **Sistema de Carona Legal: SISCARLEG**. [S.l.], 2013. Disponível em: <<http://portaldavila.com/Anuncios%20Clientes/Carona%20Legal/materias/PROJETO%20DF%202013%20-%20SISTEMA%20DE%20CARONA%20LEGAL.pdf>>. Acesso em: 22 ago. 2014.

BRASIL. **Lei 10406/02, de 10 de Janeiro de 2002**. Brasília, 2002. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/leis/2002/110406.htm](http://www.planalto.gov.br/ccivil_03/leis/2002/110406.htm)> Acesso em: 06 set. 2014.

BRASIL. **Lei Nº 12.468, de 26 de agosto de 2011**. Brasília, 2011. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_Ato2011-2014/2011/Lei/L12468.htm](http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2011/Lei/L12468.htm)> Acesso em: 06 set. 2014.

CARONETAS. **Participantes**. São Paulo, 2014<sup>a</sup>. Disponível em: <<http://www.caronetass.com.br/apoio/participantes>>. Acesso em: 01 set. 2014.

CARONETAS. **Quem somos**. São Paulo, 2014b. Disponível em: <<http://www.caronetass.com.br/quem-somos>>. Acesso em: 07 set. 2014.

CONFEDERAÇÃO NACIONAL DE TRANSPORTE. **Pesquisa CNT de Rodovias 2013**. Brasília, 2013. Disponível em: <[http://pesquisarodovias.cnt.org.br/Relatorio%20Geral/Pesquisa\\_Rodovias\\_2013\\_LOW\\_31.10.13.pdf](http://pesquisarodovias.cnt.org.br/Relatorio%20Geral/Pesquisa_Rodovias_2013_LOW_31.10.13.pdf)>. Acesso em: 01 set. 2014.

DEPARTAMENTO NACIONAL DE TRÂNSITO. **Frota nacional**. Brasília, 2014. Disponível em: <<http://www.denatran.gov.br/download/frota/FrotaRegioesTipoUFJUN.2014.zip>>. Acesso em: 06 set. 2014.

ÉPOCA NEGÓCIOS. **Aplicativos de carona geram polêmica sobre legalidade do serviço.** Rio de Janeiro, 2014. Disponível em: <  
<http://epocanegocios.globo.com/Inspiracao/Empresa/noticia/2014/03/aplicativos-de-carona-geram-polemica-sobre-legalidade-do-servico.html>>. Acesso em: 07 set. 2014.

EXAME. **Brasil é o quarto país do mundo em número de smartphones.** São Paulo, 2013. Disponível em: <  
<http://exame.abril.com.br/tecnologia/noticias/brasil-e-o-quarto-pais-do-mundo-em-numero-de-smartphones>>. Acesso em: 07 set. 2014.

EXAME. **App de caronas Uber pode ser suspenso pela Prefeitura de SP.** São Paulo, 2014. Disponível em: <  
<http://exame.abril.com.br/tecnologia/noticias/app-de-caronas-uber-pode-ser-suspenso-pela-prefeitura-de-sp>>. Acesso em: 15 set. 2014.

FACEBOOK. **Grupo caronas BNU.** [S.l.], 2014a. Disponível em: <  
<https://www.facebook.com/groups/557270207663845/?fref=ts>>. Acesso em: 07 set. 2014.

FACEBOOK. **Grupo caronas Blumenau - Floripa.** [S.l.], 2014b. Disponível em: <  
<https://www.facebook.com/groups/caronafloripablumenau/?fref=ts>>. Acesso em: 07 set. 2014.

FACEBOOK. **Grupo caronas FURB.** [S.l.], 2014c. Disponível em: <  
<https://www.facebook.com/groups/1471903513096105/?fref=ts>>. Acesso em: 07 set. 2014.

G1. **EUA e China anunciam acordo para reduzir emissão de gases poluentes.** Rio de Janeiro, 2014. Disponível em: <  
<http://g1.globo.com/natureza/noticia/2014/11/eua-e-china-anunciam-acordo-para-reduzir-emissao-de-gases-poluentes.html>>. Acesso em: 25 jun. 2015.

GAZETA DO POVO. **Carona solidária, boa para o bolso e para o trânsito.** Curitiba, 2012. Disponível em: <  
<http://www.gazetadopovo.com.br/vidaecidadania/conteudo.phtml?id=1270090&tit=Carona-solidaria-boa-para-o-bolso-e-para-o-transito>>. Acesso em: 25 ago. 2014.

GRAVA, Sigurd . **Urban Transportation Systems: Choices for communities.** 01 ed. Mcgraw-hill, 2002.

GOOGLE CLOUD. **The Google Cloud Platform** [S.l.], 2015. Disponível em: <  
<https://cloud.google.com/appengine/docs/java/endpoints/>> Acesso em: 01 jul. 2015.

GOOGLE DEVELOPERS. **The Google Directions API** [S.l.], 2015. Disponível em: <  
<https://developers.google.com/maps/documentation/directions/>> Acesso em: 26 jun. 2015.

GOOGLE MAPS. **Google Maps** [S.l.], 2015. Disponível em: <  
<https://www.google.com.br/maps/dir/R.+Amazonas++Garcia,+Blumenau++SC/R.+Nereu+Ramos++Jardim+Blumenau,+Blumenau++SC>> Acesso em: 26 jun. 2015

GOOGLE PLAY STORE. **Uber** [S.l.], 2014. Disponível em: <<https://play.google.com/store/apps/details?id=com.ubercab>> Acesso em: 07 set. 2014.

HURWITZ, Judith. **Big Data for dummies**. Hoboken: Willey, 2013.

INTERNACIONAL AGENCY OF ENERGY. **Railway Handbook 2012** [S.l.], 2012. Disponível em: <[http://www.uic.org/com/IMG/pdf/iea-uic\\_2012final-lr.pdf](http://www.uic.org/com/IMG/pdf/iea-uic_2012final-lr.pdf)> Acesso em: 06 set. 2014.

LECHETA, Ricardo R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 02 ed. São Paulo: Novatec, 2010.

LECHETA, Ricardo R. **Google Android para tablets: aprenda a desenvolver aplicações para o Android: de smartphones a tablets**. São Paulo: Novatec, 2012.

MARCOVITCH, Jacques. **Para mudar o futuro: mudanças climáticas, políticas públicas e estratégias empresariais**. São Paulo: Saraiva, 2006.

MCCREARY, Dan; KELLY, Ann. **Making Sense of NoSQL: A guide for managers and the rest of us**. Greenwich: Manning, 2013.

MINISTÉRIO DO MEIO AMBIENTE. **A Carta da Terra**. Brasília, 2014. Disponível em: <[http://www.mma.gov.br/estruturas/agenda21/\\_arquivos/carta\\_terra.pdf](http://www.mma.gov.br/estruturas/agenda21/_arquivos/carta_terra.pdf)>. Acesso em: 02 set. 2014.

MICHAELIS. **Dicionário prático da língua portuguesa**. São Paulo: Melhoramentos, 2008.

OLHAR DIGITAL. **Samsung detém mais da metade do mercado de smartphones no Brasil**. [S.l.], 2014. Disponível em: <<http://olhardigital.uol.com.br/pro/noticia/samsung-detem-mais-da-metade-do-mercado-de-smartphones-no-brasil/40282>>. Acesso em: 07 set. 2014.

RITTINGHOUSE, John W.; Ransome, James F. **Cloud Computing: implementation, management and security**. Boca Raton: CRC Press, 2010.

R7. **Polêmico entre taxistas, aplicativo para "carona paga" Uber estreia no Brasil**. São Paulo, 2014. Disponível em: <<http://noticias.r7.com/tecnologia-e-ciencia/polemico-entre-taxistas-aplicativo-para-carona-paga-uber-estreia-no-brasil-28062014>> Acesso em: 06 set. 2014.

SADALAGE, Pramond J. **NoSQL Essencial: Um guia conciso para o mundo emergente de consistência poliglota**. São Paulo: Novatec, 2013.

SOSINSKY, Barrie. **Cloud Computing Bible**. Indianapolis: Willey, 2011.

TEXAS DEPARTMENT OF TRANSPORTATION. **North Texas HOV Lanes**. [S.l.], 2014. Disponível em: <<http://www.txdot.gov/inside-txdot/projects/studies/dallas/high-occupancy-vehicle-lanes.html>>. Acesso em: 01 set. 2014.

UBER. **Rio de Janeiro** [S.l.], 2014. Disponível em: <<https://www.uber.com/cities/rio-de-janeiro>>. Acesso em: 07 set. 2014.

VELTE, Anthony T. **Cloud Computing: computação em nuvem: uma abordagem prática**. 2 ed. Rio de Janeiro: Alta Books, 2011.



## APÊNDICE A – DETALHAMENTO DOS CASOS DE USO

Este Apêndice apresenta a descrição dos casos de uso conforme previstos no diagrama apresentado na seção 3.2.2, conforme o Quadro 4.

Quadro 4 - Descrição dos Casos de Uso

### **UC01 Cadastrar Usuário**

Permite ao usuário informar os dados de um novo usuário do sistema.

### **UC02 Cadastrar Viagem**

Permite ao usuário inserir os dados de uma viagem que irá realizar no sistema..

### **UC03 Identificar trajetos rotineiros semelhantes**

Permite ao usuário visualizar quais outros usuários percorrem caminhos próximos ao seu.

#### **Constraints**

*Pré-condição* . O usuário deve estar cadastrado no sistema.

*Pós-condição* . Usuário visualiza outros usuários com trajetos semelhantes.

#### **Cenários**

##### **Localiza outros usuários {Principal}.**

1. Usuário informa local de partida e destino
2. Usuário o horário em que percorre o trajeto
3. São listados os demais usuários que realizam trajetos semelhantes.
4. Usuário encerra o caso de uso.

##### **Não localiza outros usuários {Alternativo}.**

No passo 3, o sistema verifica que não existe nenhum outro usuário com trajeto e horário semelhantes

- 2.1. Sistema exibe a mensagem de “Carona não encontrada!”

### **UC04 Buscar Caronas**

Permite ao usuário buscar por caronas já cadastradas no sistema.

### **UC05 Efetuar Login**

Permite ao usuário informar seu *login* e senha para acessar o sistema.

#### **Constraints**

*Pré-condição* . O usuário deve estar cadastrado no sistema.

*Pós-condição* . Usuário acessa o sistema.

#### **Cenários**

##### **Acesso ao sistema {Principal}.**

1. Usuário informa usuário e senha de acesso
2. Sistema valida as informações
3. Usuário acessa a tela principal do sistema.

##### **Senha Incorreta {Alternativo}.**

No passo 2, o sistema verifica que a senha não corresponde à cadastrada para o usuário

2.1. Sistema exibe a mensagem de “Senha incorreta”

2.2. Usuário é redirecionado para a tela de *login*.

**Usuário Inexistente {Alternativo}.**

No passo 2, o sistema verifica que o usuário informado não existe na base de dados

2.1. Sistema exibe a mensagem de “Usuário inexistente”

2.2. Usuário é redirecionado para a tela de *login*.

**UC06 Visualizar estatísticas**

Permite ao administrador visualizar por meio de uma interface *web* informações estatísticas como quantidade de caronas cadastradas, destinos mais procurados e usuários com maior quantidade de caronas.

## APÊNDICE B – DICIONÁRIO DE DADOS

Este Apêndice apresenta a descrição das tabelas do banco de dados apresentadas na subseção 3.2.4 deste trabalho. Nos Quadros de 5 a 7 estão o dicionário de dados das tabelas do sistema.

Os tipos de dados utilizados nos atributos são:

- a) *integer*: armazena números inteiros;
- b) *String*: armazena conjuntos de caracteres;
- c) *date*: armazena data;
- d) *double*: armazena números de precisão e escala fixos.

Quadro 5 - Tabela Carona

Carona		
Campo	Tipo	Descrição
key	<i>int</i>	Chave gerada pelo <i>Endpoints</i>
comentario	<i>string</i>	Comentário inserido pelo usuário na carona
ID/Name	<i>int</i>	ID Gerado pelo objectify.
Write Ops	<i>int</i>	Gerado pelo objectify.
data	<i>date</i>	Data da Carona
destino	<i>string</i>	Destino da Carona
origem	<i>string</i>	Origem da Carona
usuario	<i>int</i>	Usuário da Carona

Quadro 6 - Tabela Usuario

Usuario		
Campo	Tipo	Descrição
key	<i>int</i>	Chave gerada pelo <i>Endpoints</i>
ID/Name	<i>int</i>	ID Gerado pelo objectify.
Write Ops	<i>int</i>	Gerado pelo objectify.
email	<i>string</i>	Email do usuário
nome	<i>string</i>	Nome do usuário
senha	<i>string</i>	Senha do usuário.

Quadro 7 - Tabela PontoRota

Usuario		
Campo	Tipo	Descrição
key	<i>int</i>	Chave gerada pelo <i>Endpoints</i>
ID/Name	<i>int</i>	ID Gerado pelo objectify.
Write Ops	<i>int</i>	Gerado pelo objectify.
carona	<i>int</i>	Código da Carona à qual o ponto pertence.
endLocationLat	<i>double</i>	Latitude da localidade final do ponto.
endLocationLng	<i>double</i>	Longitude da localidade final do ponto.
startLocationLat	<i>double</i>	Latitude da localidade inicial do ponto.
startLocationLng	<i>double</i>	Longitude da localidade inicial do ponto.