



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA – DAINF
CURSO DE TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS DISTRIBUÍDOS

Marcos Vinicio Moreno Munhoz Filho

Sistema de Informação para Compartilhamento de Caronas entre Alunos dos
Campi da UTFPR-CT

TRABALHO DE CONCLUSÃO DE CURSO

Curitiba

2013

Marcos Vinicio Moreno Munhoz Filho

Sistema para Compartilhamento de Caronas entre Alunos dos Campi da
UTFPR-CT

Trabalho de Conclusão de Curso
apresentado à UTFPR como requisito
parcial para obtenção do grau de Tecnólogo
em Desenvolvimento de Sistemas
Distribuídos

Orientador: Professor Alexandre
Reis Graeml

Curitiba

2013

AUTOPSILOGRAFIA

“Trace the tracelessness of the ant,
Every ant has reached this perfection.
As he comes, so he goes,
Flowing as water flows,
Essential but secret like a rose”

José Garcia Villa

SUMÁRIO

1	INTRODUÇÃO.....	9
1.1	FORMULAÇÃO DO PROBLEMA.....	10
1.2	OBJETIVOS.....	10
1.3	JUSTIFICATIVA.....	12
2	LEVANTAMENTO BIBLIOGRÁFICO.....	14
2.1	ANÁLISES E PROPOSTAS SOBRE CARPOOLING.....	15
2.1.1	Políticas de encorajamento à prática de <i>carpooling</i>	16
2.1.2	Abordagens sistêmicas.....	19
2.1.2.1	<i>Carpooling</i> convencional (<i>pre-arrangement</i>).....	19
2.1.2.2	<i>Carpooling</i> casual (" <i>slug lines</i> ").....	24
2.1.2.3	<i>Carpooling</i> dinâmico.....	25
2.2	TECNOLOGIA GIS.....	31
2.2.1	Conceito de GIS.....	31
2.2.2	Enquadramento estrutural.....	32
2.2.3	Tipos de GIS.....	33
2.2.4	GIS aplicado ao <i>carpooling</i>	35
2.3	SÍNTESE DO LEVANTAMENTO BIBLIOGRÁFICO.....	36
3	METODOLOGIA.....	37
3.1	ANÁLISE DOS TIPOS DE <i>CARPOOLING</i>	37
3.2	TECNOLOGIAS EMPREGADAS.....	39
3.2.1	Tecnologias empregadas diretamente.....	39
3.2.2	Tecnologias empregadas indiretamente.....	40
3.2.3	Tecnologias empregadas direta e indiretamente.....	41
3.2.4	Motivos para o emprego indireto de algumas tecnologias.....	42
3.2.5	Escolha de uma aplicação de <i>Web GIS</i>	43
3.2.5.1	Qualidade e confiabilidade.....	43
3.2.6	Utilização de APIs do <i>Google Maps</i>	45
3.2.6.1	Geocode.....	45
3.2.6.2	Directions.....	47
3.2.6.3	Staticmaps.....	49
3.3	SÍNTESE DA METODOLOGIA.....	51
4	RESULTADOS.....	52
4.1	ANÁLISE DO SISTEMA.....	52
4.1.1	Levantamento de requisitos.....	52
4.1.1.1	Requisitos funcionais.....	52
4.1.1.2	Requisitos não-funcionais.....	53
4.1.2	Diagrama de estados do sistema de informação.....	53
4.1.3	Casos de uso.....	55

4.1.3.1	Cadastro.....	55
4.1.3.2	<i>Login</i>	55
4.1.3.3	<i>Logout</i>	56
4.1.3.4	Gerenciamento de dados	56
4.1.3.5	Busca	56
4.1.3.6	Envio de mensagem	57
4.1.3.7	Envio de Convite	57
4.1.3.8	Visão geral dos casos de uso	58
4.1.4	Diagrama de máquina de estados de convites	58
4.1.5	Modelo entidade-relacional	60
4.2	DESENVOLVIMENTO DO SISTEMA	62
4.2.1	Paradigma e metodologia de programação	63
4.2.2	Estrutura.....	64
4.2.3	Mecanismo de busca e ordenação de pares.....	68
4.3	SÍNTESE DOS RESULTADOS.....	73
5	CONCLUSÃO	75
6	REFERÊNCIAS	78

LISTA DE ABREVIATURAS E SIGLAS

- API:** Application Programming Interface
- BD:** Banco de Dados
- DAO:** Data Access Object.
- DTO:** Data Transfer Object.
- DCPP:** Daily Car Pooling Problem.
- GIS:** Geographic Information System.
- GISci:** Geographic Informaition Science.
- HTML:** Hypertext Markup Language.
- HTTP:** Hypertext Transfer Protocol.
- HTTPS:** Hypertext Transfer Protocol Secure.
- IPEA:** Instituto de Pesquisa e Estatística Aplicada.
- JSP:** Java Server Pages.
- LCPP:** Long-Term Car Pooling Problem.
- MAS:** Multi-Agent System.
- MER:** Modelo Entidade-Relacional.
- MVC:** Model-View-Controller.
- SDI:** Spatial Data Infrastructure.
- SI:** Sistema de Informação.
- SOV:** Single Occupant Vehicle.
- TPB:** Teory of the Planed Behavior.
- XML:** Extended Markup Language.

Resumo

Este estudo propõe um sistema de informação com a finalidade de promover o compartilhamento de caronas para o deslocamento de alunos da UTFPR dos *campi* de Curitiba. Ao longo dessa pesquisa, analisam-se os exemplos existentes de Sistemas de Informação (SIs) com esse objetivo, separadamente, de acordo com suas respectivas categorias ou abordagens sistêmicas, também levando-se em conta fatores não necessariamente funcionais, tais como: privacidade e comportamento de deslocamento. O estudo aborda o conceito da tecnologia GIS (*Geographic Information System*) e procura contextualizar o seu papel em aplicações que utilizam dados de geoprocessamento. Já a etapa de metodologia serviu para definir a abordagem mais adequada para implementar no SI. Tendo o projeto – batizado de Caronas-UTF – sido concluído, o resultado se mostrou suficientemente satisfatório no que diz respeito à utilização de informações geográficas para determinar grupos de usuários para o compartilhamento de veículos no deslocamento entre os *campi* da UTFPR-CT, facilitando a distribuição de informações entre os interessados e contribuindo para um uso mais eficiente de veículos, a redução do número de carros nas ruas e um ambiente mais saudável.

Palavras-chave: Sistema de Informação Geográfica. Compartilhamento de Caronas. Compartilhamento de Veículos. GIS. WebGIS. Java. JSP.

Abstract

This study proposes an information system for carpooling to be used by UTFPR students of the two campi in Curitiba. Throughout this research, examples of existing ISs (Information Systems) with this goal are analyzed, separately, according to their respective categories or systemic approaches. Factors that are not necessarily functional, such as privacy and travel behavior, are also taken into account. The study addresses the concept of GIS technology (Geographic Information System) and contextualizes its role in applications that deal with geoprocessing data. In the methodology section the alternatives to evaluate which systemic approach would be more appropriate to implement the IS are assessed. Having completed project – named UTFPR-Carpooling – the result showed to be satisfactory with respect to the possibility of using geographic information to determine appropriate groups for sharing vehicles to travel from one university campus to the other, making it easier for those interested to share information and contributing for a more efficient use of vehicles, reduction of the number of cars in the streets and a healthier environment.

Keywords: Geographic Information System. Rideshare. Carpool. GIS. WebGIS. Java. JSP.

1 INTRODUÇÃO

Veç por outra surgem iniciativas e incentivos com respeito a compartilhar veículos para trajetos do dia-a-dia. Às vezes tais iniciativas são reações às más condições no trânsito de áreas urbanas, outras vezes têm cunho financeiro e ainda outras refletem preocupações ambientais. Um dos exemplos mais expressivos, relacionado ao aspecto financeiro, foi a crise do alto preço do petróleo em meados dos anos 1970, que fez com que muitos cidadãos dos Estados Unidos da América recorressem à prática do *car pooling* (SOLTYS, 2009).

Quando um grupo de pessoas que se conhecem resolve compartilhar um veículo, isso não acarreta em grandes problemas de organização. No entanto, enquanto aumentam consideravelmente os problemas relacionados ao fluxo de veículos nas vias urbanas, bem como o esgotamento de vagas de estacionamento em pontos movimentados, começa-se a pensar sobre a possibilidade de compartilhar veículos em larga escala, visto que muitas pessoas fluem para pontos relativamente próximos para trabalhar, estudar etc. Naturalmente, quando se aumenta a escala de usuários, a tarefa de arranjar um grupo de pessoas em cada veículo disponível fica mais complexa e a possibilidade de reunir em um veículo pessoas desconhecidas traz à tona questões de segurança e integridade de informações. É difícil conceber este modelo de maior escala sem o uso de recursos computacionais.

Neste trabalho foram analisadas abordagens computacionais (algorítmicas) referentes ao compartilhamento de veículos, bem como às tecnologias de geoprocessamento – úteis tanto para mapeamento e apresentação de endereços físicos ou geográficos (baseados em coordenadas) como para definir regras para formação de grupos de compartilhamento.

O produto final deste esforço é um SI voltado ao problema do compartilhamento de caronas, desenvolvido em linguagem Java com interface *web*, que põe alunos cujos itinerários são semelhantes em contato (sem deixar de levar em conta questões de privacidade neste processo) e determina rotas otimizadas para seus trajetos. Este SI foi batizado de Caronas-UTF.

1.1 Formulação do problema

O problema do compartilhamento de veículos é abordado do ponto de vista sistêmico, sendo que três modelos foram encontrados e analisados. Antes disso, contudo, foi feita uma análise com respeito às possibilidades de aceitação de uma implementação informatizada nesse campo. Durante o levantamento bibliográfico percebe-se que a prática de compartilhar veículos é válida, pelos aspectos que a tornam justificável – financeiro, social, ambiental ou de tempo. Além disso, a formulação abrange uma breve análise sobre tecnologias aplicáveis ao problema.

Com as opções de abordagens sistêmicas tendo sido cobertas, durante a etapa de metodologia faz-se reflexões sobre qual o modelo mais adequado para o sistema de informação a ser desenvolvido. Apresentam-se os resultados referentes ao produto almejado com este trabalho – o SI para compartilhamento de caronas – e, em uma última seção, discute-se o êxito e limitações deste trabalho, incluindo sugestões quanto à sua continuidade.

1.2 Objetivos

Este trabalho propõe um protótipo de solução computacional para atender o compartilhamento de caronas para deslocamento de universitários aos seus respectivos *campi*, tendo como público-alvo os alunos da UTFPR dos *campi* de Curitiba (UTFPR-CT).

O SI deve ser capaz de identificar alunos cujos endereços favoreçam o compartilhamento de um veículo, sem alterar significativamente a rota de deslocamento dos usuários. A solução não necessariamente satisfará todas as demandas de uma aplicação comercial, ressaltando que se trata de um protótipo.

O objetivo principal em termos de algoritmo envolve a formação de *matches* de indivíduos com potencial para compartilhamento de veículo. Em relação a termos não-funcionais, a análise traz à atenção principalmente questões de confiabilidade de informações e privacidade. Para atingir esta meta, os esforços foram divididos em três campos:

- determinação sistêmica do SI: determinar a abordagem sistêmica do SI influi não somente em questões algorítmicas, como também em toda a estrutura do projeto em si, de modelagem a *hardware*;
- análise das tecnologias de geoprocessamento: é essencial visto que o sistema necessita identificar locais e definir rotas. Também é essencial identificar o papel que estas tecnologias desempenharão na solução – meramente um recurso (fonte de dados), ou parte do conjunto de entidades do sistema? – sendo que a escolha do papel desta pode acarretar em drásticas diferenças estruturais de uma para outra opção;
- análise de sistema: necessária para determinar a melhor representação possível das entidades do mundo real de acordo com os requisitos inerentes ao sistema.

A atenção do estudo é voltada a estes pontos até a etapa de construção do sistema, ou seja, da programação em si, registrada a partir do item 4.2.

1.3 Justificativa

A iniciativa por trás deste trabalho decorre da constatação da necessidade de alternativas para o deslocamento em áreas urbanas, buscando minimizar o impacto causado por estruturas de transporte deficientes, melhorando também a qualidade de vida das pessoas nessas áreas. Visto que abrange um número limitado do público de Curitiba e região – alunos da UTFPR-CT – a utilização de um SI baseado no protótipo resultante do trabalho não causará um impacto significativo na redução de veículos nas vias da cidade. De qualquer forma, tal esforço de pesquisa, bem como seu produto final, servem como base para expansão futura da iniciativa, já que o modelo pode ser adaptado a outras instituições de ensino e organizações de vários segmentos. O protótipo também poderá ser usado como base para estudo quanto à aceitação do público – em especial do público acadêmico – em relação ao compartilhamento de caronas.

A pesquisa também se justifica pelos benefícios de ordem prática que o compartilhamento de caronas poderia trazer, tais como:

- redução de custos pessoais e sociais: pessoas que compartilham um mesmo veículo podem determinar uma forma de dividir os custos relacionados a combustível. De acordo com dados do *site* precodoscombustiveis.com.br (PREÇO DOS COMGUSTÍVEIS, s.d.), o custo médio do litro da gasolina vendida nos postos capital paranaense (verificado em junho de 2013) é de R\$2,84. A prática também pode ser vantajosa em relação ao deslocamento por meio de transporte público tendo em vista o alto custo da passagem de ônibus, atualmente em R\$2,70. Adicionalmente, existe a possibilidade de contribuir para uma redução de custos em contexto mais amplo, já que engarrafamentos contribuem para um prejuízo considerável para as cidades. Estima-se que a cidade de São Paulo, por exemplo, “perde em produção R\$ 26,8 bilhões por

ano, valor adicional de riqueza que poderia ser gerada, se o tempo perdido no trânsito fosse gasto no trabalho” (IPEA, 2011);

- Redução do número de veículos nas vias públicas: a grande quantidade de veículos trafegando nas vias públicas, especialmente em horários de pico, causam congestionamentos. Devido a esse problema, a velocidade média dos automóveis em Curitiba é de 27 km/h e a dos ônibus de 21 km/h no horário de pico da manhã – das 7h às 9h. (IPEA, s.d.)

Tendo o projeto sido explanado e detalhado quanto às suas intenções e motivações, segue a etapa de levantamento bibliográfico, onde se busca um embasamento teórico para atingir o objetivo almejado.

2 LEVANTAMENTO BIBLIOGRÁFICO

A prática de compartilhamento de carona é uma entre várias medidas possíveis de diminuição de efeitos nocivos dos problemas de trânsito e obtenção de um nível sustentável de mobilidade, especialmente em áreas urbanas. Schmitt (2006) lista cinco estratégias de “gerenciamento da demanda de viagens”, dividindo-as em categorias, como segue:

- uso mais eficiente dos automóveis e restrições ao seu uso;
- incentivo ao uso de modos não motorizados;
- incentivo ao transporte coletivo;
- programas de redução de viagens de trabalhadores;
- gerenciamento de estacionamentos.

Neste contexto, o termo *carpooling* é mencionado, juntamente a *vanpooling* e *carsharing* como categorias de *ridesharing*, ou “carona programada” como é chamada a prática no Brasil¹. *Ridesharing*, por sua vez, aparece como uma subcategoria da primeira categoria listada acima e destacada em sublinhado, por ser a única relacionada diretamente ao objeto deste estudo.

Esta seção aborda especificamente estudos e projetos realizados com o intuito de estimular ou implantar a prática de *carpooling*, que é a prática de arranjar pessoas em um grupo de veículos disponíveis. Para isso, as abordagens determinam, dentre outras coisas, um público-alvo (pessoas potencialmente dispostas a aderir à ideia), condições ideais etc. Note-se que algumas análises feitas neste capítulo não são necessariamente válidas como argumento para implantação do projeto Caronas-UTF, visto que lidam

¹ É preciso atenção para não confundir os termos *carpooling* e *carsharing*. Tanto Schmitt (2006) como Fukuda *et al.* (2005) descrevem o último como uma prática de locação de veículos em curto prazo (por duas pessoas ou mais, com o fim de compartilhá-lo) idealizada como uma alternativa à propriedade dos veículos.

com dados de cidades com diferentes variáveis sociais assim como diferentes modelos de sistemas de transporte. No entanto, observar as propostas deste capítulo ajuda a definir um quadro que seja mais adequado para a implantação do Caronas-UTF.

2.1 Análises e Propostas Sobre Carpooling

Existem formas já implementadas de *carpooling* em algumas cidades do mundo. Em geral, há um estudo precedendo cada projeto e as características do sistema de compartilhamento de veículos são concebidas de acordo com o perfil do trânsito da cidade em questão. Além desses projetos já implantados, existe um vasto número de teses e artigos acadêmicos abordando o assunto – algumas vezes, baseando seus argumentos em observações de resultados dos sistemas que estão em funcionamento.

No entanto, quando se estuda a prática de *carpooling* em suas variadas formas, a estrutura do sistema de caronas não é o único aspecto a ser abordado. Além de como os passageiros serão distribuídos ou como devem ser estabelecidos os pontos de encontro, também se discutem formas de educação e conscientização do público e de entidades públicas e privadas, a fim de que haja um apoio representativo à prática. Fica clara a importância que tem o aspecto da conscientização social, pois quanto maior a aderência ao *carpooling*, mais evidentes ficarão os seus impactos benéficos no trânsito.

Tendo posse do conhecimento sobre os fatores que afetam – ou podem afetar – o comportamento de deslocamento ou viagem das pessoas, um bom planejamento pode enfatizar fatores positivos e diminuir fatores negativos em relação ao *carpooling*. Soltys (2009) destaca três grupos principais de fatores que afetam a prática do *carpooling*:

- fatores individuais (custo, restrições de mobilidade, aspectos demográficos e escolha do modo de transporte);
- fatores espaciais (acessibilidade e padrões de deslocamento);
- fatores temporais (tempo de deslocamento e agendamento).

Com estas informações em mente seguem os subtópicos desta seção.

2.1.1 Políticas de encorajamento à prática de *carpooling*

Como dito anteriormente, para que a adoção do *carpooling* por um determinado público seja bem sucedida, é necessário o apoio deste público. Dewan e Ahmad (2007) mencionam as campanhas e a publicidade sobre benefícios do *carpooling* como sendo de “suprema importância”. Em seu artigo os autores propõem que a iniciativa de promover o *carpooling* na cidade de Delhi, Índia, venha de autoridades governamentais, sendo o governo local o coordenador da implantação de *carpooling* tanto nas agências governamentais e ministérios, como nas instituições privadas. Também caberia ao governo fornecer um serviço de “*match-making*”² para usuários de *carpooling*. As medidas de encorajamento à prática, sugeridas por Dewan e Ahmad (2007), incluem pistas dedicadas a veículos com alta taxa de ocupação, reserva de espaço em estacionamentos sem cobrança aos que praticam o *carpooling* e o fim desse privilégio (caso exista) para os que não o praticam. Às empresas que usam estacionamento próprio, é sugerida uma taxa para os empregados que não aderem ao sistema de *carpooling*.

Outras abordagens são viáveis no que se refere a encorajar o público, além dos benefícios econômicos citados. Por exemplo, alguns

² *Match-making*: um serviço para relacionar características semelhantes ou coincidentes entre objetos de uma dada população. Em se tratando de *ridesharing* pode servir para determinar potenciais parceiros de viagem para compartilhar o mesmo veículo baseando-se em proximidade geográfica, janelas de horário coincidentes ou próximas e comportamento de viagem idênticos ou semelhantes (DEWAN e AHMAD, 2007)

pesquisadores utilizaram-se de dados resultantes de estudos que preveem o impacto do *carpooling* no trânsito, meio ambiente e na sociedade em geral. Minnett e Pearce (2011) contam que propuseram ao ACR (*Auckland Regional Council*), a realização de um estudo de estimativa de custos relacionados ao congestionamento das vias públicas de Auckland, Nova Zelândia, e de uma estimativa de impacto que o uso de *carpooling* teria sobre este custo. O ACR criou um modelo assumindo que um grupo de 2500 pessoas das que eles chamaram de motoristas de “veículo de um único ocupante” (SOV, do inglês: *single occupant vehicle*) adotariam o sistema de *carpooling* para três ocupantes (incluindo o motorista), o que resultaria na participação de mais 5000 passageiros (2 passageiros para 1 motorista em cada *pool*). Dentre outros resultados, foi estimada uma economia de 9,5 milhões de litros de petróleo fóssil por ano na cidade de Auckland, sendo que, desta soma, 2,5 milhões de litros seriam poupados exclusivamente pelos usuários do sistema de *carpooling*. Outro benefício previsto relaciona-se ao consumo de energia em função da velocidade média dos veículos nas vias públicas. A velocidade média em horários de pico em Auckland é de 37,81 km/h. A velocidade média estimada com o uso de *carpooling* (nas condições que o estudo estabeleceu) subiria para 40,44 km/h. Baseados em um estudo realizado nos EUA, os pesquisadores determinaram a redução de energia em *Joules* (que implica em redução de emissões de CO₂) causada pelo uso de *carpooling* naquelas condições, conforme mostrado na Figura 1.

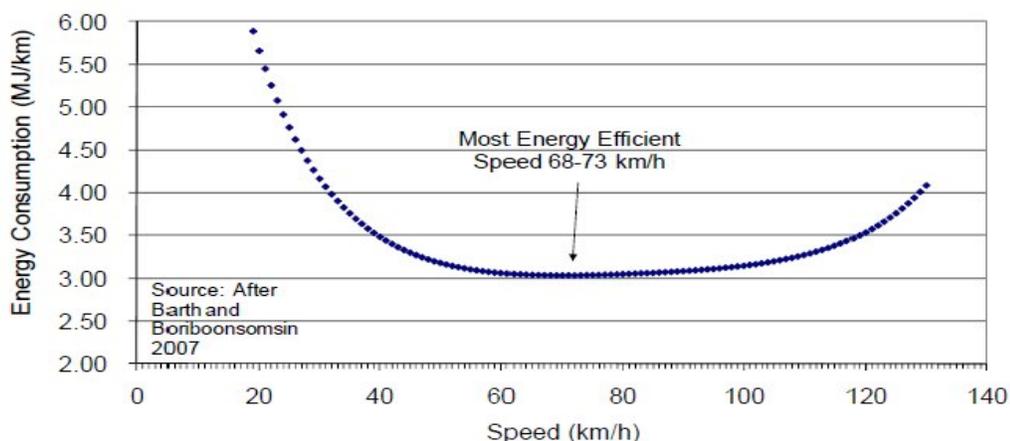


Figura 1. Energy Consumption vs. Speed

Fonte: Minett e Pearce (2011)

Baseando-se no aumento da velocidade média que foi estimada, nota-se uma redução de 3,7 MJ/km para cerca de 3,4 MJ/km. É uma redução considerável se for pensada em longo prazo, gerando, além da economia de combustível, uma significativa diminuição de poluentes.

Medidas como as descritas nesta seção podem ajudar a motivar a sociedade a aderir à prática de *carpooling*. Quando aprovada, no entanto, não é absolutamente certo que a prática obterá sucesso. Segundo Soltys (2009), é necessário também planejar o funcionamento de um sistema de *carpooling* de acordo com as características de comportamento e de tomadas de decisões dos seus potenciais usuários. Para isso, a autora sugere o uso da Teoria do Comportamento Planejado (TPB, sigla do termo original em inglês) para lidar com estas questões.

Na seção seguinte abordar-se-ão formas de idealizar o compartilhamento de caronas do ponto de vista sistêmico.

2.1.2 Abordagens sistêmicas

Como dito no início deste capítulo, existem variáveis a serem consideradas antes da implantação de um sistema de compartilhamento de caronas em uma dada localidade. Além das características já abordadas neste capítulo, a forma de se projetar o compartilhamento das caronas também é uma variável. E esta variável pode ser, em geral, definida em função das variáveis já mencionadas por Soltys (2009) – os três grupos de fatores. As variações conhecidas de implementação de *carpooling* são descritas nos próximos subtópicos.

2.1.2.1 *Carpooling* convencional (*pre-arrangement*)

Boa parte da literatura sobre *carpooling* trata do assunto do ponto de vista do modelo pré-arranjado, ou seja, a prática que envolve definir grupos compatíveis para lotação de cada veículo. A compatibilidade de tais usuários pode ser dada por localização geográfica, preferências demográficas (tais como sexo, idade, raça, grau de escolaridade etc.), tempo de partida/regresso de viagem e preferências relacionadas ao comportamento de viagem (hábitos de tabagismo, costume de ouvir rádio, maneira como se conduz o veículo etc.), conforme ressaltado por Soltys (2009).

Calvo *et al.* (2004) descrevem este modelo considerando duas submodalidades: *daily car pooling problem* (DCPP) e *long-term car pooling problem* (LCPP). Na primeira, cada dia um usuário motorista declara sua disponibilidade para dar carona informando sua rota para aquele dia em específico. O problema do DCPP, segundo os autores, é – a cada dia – atribuir passageiros aos motoristas e identificar rotas a serem percorridas com o fim de minimizar os custos da viagem, podendo os passageiros ser prejudicados devido à capacidade do veículo e às janelas de horário dos usuários. O LCPP consiste em formar (em caráter mais permanente) grupos

de motoristas que revezam a responsabilidade de transportar os demais integrantes do grupo aos seus locais de destino. O objetivo de ambos os modelos é maximizar o número de usuários por *pool* e minimizar a distância total viajada pelos veículos adotados.

Os sistemas informatizados para realização de *carpoolings* pré-arranjados, em geral, estão baseados na premissa de que os grupos são formados para compartilhar veículos em longo prazo – no modelo LCPP. De fato, o site da *Smart Commute Association* (SCA), que provê serviços de *carpooling* no Canadá, deixa explícito em seu termo contratual que se for detectado o uso do *site* para viagens ocasionais, os administradores revogarão a filiação do usuário com tal comportamento (SOLTYS, 2009).

Além da frequência de atividade nos *pools*, Calvo *et al.* (2004) classificam os sistemas de *carpooling* quanto a sua acessibilidade – aberto ou restrito. O primeiro pode ser utilizado por qualquer usuário por meio de uma interface web na *Internet*. Este modelo tem como grande problema a confiabilidade das informações coletadas. Já o segundo modelo, segundo os autores, parece mais interessante para casos em que os usuários têm um endereço de chegada em comum – por exemplo, funcionários de uma mesma organização. Entretanto, Calvo *et al.* (2004) não explicitam quais meios de restrição devem ser aplicados, seja documento pessoal, código institucional interno etc. De qualquer forma, neste modelo é mais fácil confirmar as informações coletadas e o fato de se ter um ponto de chegada em comum reduz drasticamente a complexidade da abordagem algorítmica do sistema.

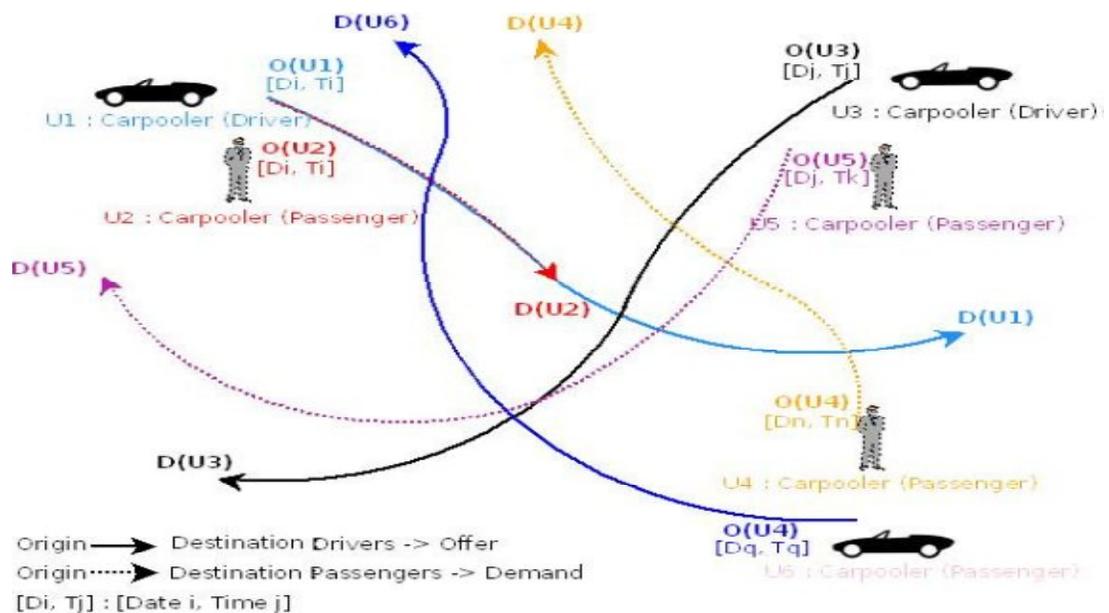


Figura 2. Carpooling: ofertas e procuras

Fonte: Sghaier *et al.* (2011)

Na Figura 2 é apresentado um modelo ilustrativo típico de *carpooling* pré-arranjado com n pontos de partida e n destinos, bem como as informações referentes às janelas de horários. Modelos com um único ponto de partida e/ou chegada são semelhantes.

Hoje em dia, há vários *sites pelo mundo* oferecendo serviços de *carpooling* pré-arranjado. Seguem alguns exemplos do que existe hoje no Brasil e no mundo.

www.smartcommute.ca/en/home: um *site* canadense que tem objetivo de ajudar empregadores e empregados, nas regiões de Toronto e Hamilton, a fazer uso de *carpooling* como uma forma alternativa de deslocamento. O serviço consiste em cadastrar empresas e funcionários (estes últimos só podem se cadastrar no sistema se a empresa em que trabalham já estiver cadastrada). Quando um usuário fornece seus dados, o sistema do *smartcommute* busca por parceiros de viagem (não necessariamente da

mesma empresa) e exibe os resultados para o usuário que fez a busca. Se forem encontrados parceiros adequados, o usuário pode escolher se entra, ou não, em contato com os potenciais parceiros por meio do *site*.

www.craigslist.org: um portal de anúncios para comunidades *online* vinculadas ao *site*, disponível para várias cidades do mundo, mas principalmente difundido em cidades dos EUA. Os usuários cadastrados podem receber anúncios que variam de ofertas de emprego a serviços de relacionamentos (encontros) em sua localidade. Dentre as categorias de anúncios se encontra uma chamada “*rideshare*”. Dentro deste tópico há uma lista de *links* para postagens de usuários que anunciam sua intenção de dar ou pegar carona. Cabe ao usuário que posta o anúncio informar a frequência com que estará disponível para fazer o trajeto (caso não seja um deslocamento eventual), bem como outras particularidades. Usuários interessados respondem à postagem no próprio fórum do *site* ou solicitam o *e-mail* do anunciante para combinar detalhes do compartilhamento de carona.

www.carpooling.fr: *site* que promove gratuitamente a prática de *carpooling* pela França, lá também conhecida como *covoiturage*. Em sua página inicial, além de explicar superficialmente o conceito de *carpooling*, o *site* – que afirma ser o primeiro da Europa nesse tipo de serviço – disponibiliza um formulário de trajeto (que inclui a frequência com que este é feito) para usuários preencherem e submeterem ao *site*. Este, por sua vez, relaciona trajetos similares de outros usuários (motoristas ou passageiros).

www.carpoolglobal.com: *site* concebido para promover a interação de pessoas com roteiros similares para compartilhamento de carona, em nível mundial. Para utilizar o serviço um usuário deve fazer um cadastro com algumas informações referentes à sua localidade, frequência e horário do trajeto.

www.caronasolidaria.com: *site* que oferece um serviço gratuito de contato entre usuários que tencionam pegar ou oferecer caronas. Membros deste portal podem acessar e postar dados de trajetos a fim de compartilhar um veículo.

www.caronabrasil.com.br: *site* brasileiro que cadastra gratuitamente usuários, para fins de compartilhamento de carona, por todo o Brasil. Os próprios usuários criam seus roteiros e informam a frequência com que farão a viagem. Eles também indicam se estão oferecendo ou pegando carona, ou ainda se estão disponíveis para ambos os casos. Este *site* também oferece um serviço pago destinado a instituições universitárias chamado “Carona Brasil Campus”. Em ambos os serviços, as combinações de carona baseadas no perfil de cada usuário são definidas pelo que o *site* chama de “ferramentas GeoWeb”.

www.caroneiros.com: também brasileiro, consiste em uma rede social para caronas voltada especialmente a empresas. Tal rede é organizada em comunidades virtuais, sendo cada empresa uma comunidade. Os usuários devem filiar-se a uma comunidade adequada, ou seja, naquela da empresa em que trabalham, ou ainda algum lugar ao qual o usuário se desloque com frequência (uma academia, por exemplo).

Estes exemplos demonstram a flexibilidade deste tipo de *carpooling*. Foram vistos alguns serviços que atendem ao público de maneira mais informal, apenas colocando usuários interessados em contato uns com os outros – seja para viagens eventuais ou frequentes. E também foram examinados serviços que empregam TI de modo mais intenso, utilizando desde cruzamento de informações para detectar usuários possivelmente “compatíveis”, até tecnologias GIS para uma melhor estratégia de formação de grupos de viagem e para melhorar o aspecto intuitivo dos sistemas por meio de mapas.

2.1.2.2 *Carpooling* casual (“*slug lines*”)

O *carpooling* casual mantém certas características de compartilhamento de carona, mas é muito diferente do *carpooling* convencional, na forma como é concebido. Segundo Minett e Pierce (2011), este tipo de sistema opera somente em três cidades dos EUA: São Francisco CA, Washington DC e Houston TX, sendo que nas duas últimas o sistema também é conhecido por “*slug lines*”³. O *carpooling* casual é baseado em “pontos de coleta matinal”. De acordo com os autores mencionados, pela manhã, passageiros enfileiram-se nos pontos de encontro “como se estivessem em uma parada de taxi”. Os motoristas apanham um número adequado de passageiros – geralmente três ou mais, com a intenção de trafegar em faixas reservadas para veículos com alta ocupação – e seguem até um destino pré-determinado. Ainda segundo Minett e Pierce (2011), a maioria dos pontos de encontro entre motoristas e passageiros assim como os destinos de viagem são de “conhecimento interno”, ou as informações podem ser localizadas nos *websites* desenvolvidos para cada sistema. Quando um veículo tem múltiplos destinos, o motorista deve informá-los à primeira pessoa da fila, a qual deve repetir em alta voz para que todos escutem.

É digno de nota que o termo “casual” da expressão “*carpooling* casual” está relacionado primariamente à casualidade com que os *pools* são formados, e não propriamente à ideia de eventualidade. Embora alguns adeptos de *carpooling* casual possam utilizar este sistema apenas em ocasiões específicas ou apenas quando lhes convêm, o conceito geral é o de que o sistema casual deve ser praticado regularmente. Reforçando a ideia da casualidade na formação dos grupos, o *site ridenow.org* fala sobre o

³ O *site ridenow.org*, que gerencia um sistema de *carpooling* casual para a cidade de São Francisco CA, também usa o termo “*ad hoc car pools*” como um sinônimo para “*casual car pools*”. Já o termo “*slug line*”, que literalmente significa “trilha de lesma”, também se refere a uma linha de texto abreviado (contendo a localização e a hora do dia) que precede cada cena de um roteiro. (http://en.wiktionary.org/wiki/slug_line).

método chamado “*first-come; first-served*”, que usa como regra a ordem de chegada dos passageiros para a lotação dos veículos. Mas o *site* observa que pode haver “furadas de fila” baseadas na preferência dos usuários, por exemplo: mulheres motoristas talvez tenham preferência por dar carona a passageiras, assim como mulheres passageiras talvez prefiram pegar carona com motoristas do mesmo sexo. Tal atitude é prevista por Soltys (2009), conforme mencionado no início da seção 2.1 como um “fator individual”, de cunho demográfico, inerente ao *carpooling*.

Quanto ao custo, motoristas em geral não cobram para dar carona. Mas, especificamente em São Francisco, espera-se que os passageiros contribuam com um valor entre US\$1,00 e US\$3,00 (dependendo do número de pessoas no veículo) para dividir o custo do “pedágio de ponte” (taxa cobrada para atravessar a ponte da baía de São Francisco) (RIDENOW.ORG, s.d.)

Assim como acontece no início, o destino final do trajeto de volta é um ponto pré-estabelecido.

2.1.2.3 *Carpooling* dinâmico

A ideia de *carpooling* dinâmico consiste em um sistema capaz de criar grupos para compartilhamento de carona em tempo real. Neste modelo não há um pré-arranjo de grupos, pois estes consistem na localização dos usuários do sistema, que inicialmente é desconhecida. O sistema deve ser capaz de rastrear usuários para determinar sua localização, para então estar apto a formar grupos. Portanto, é praticamente indispensável o uso de tecnologias de geolocalização, como GPS, para que seja possível localizar usuários em tempo real, de forma eficaz (SGHAIER *et al.*, 2011). Um sistema com estes moldes, naturalmente se restringe à abordagem de viagens únicas (CALVO *et al.*, 2004), pois admite que um motorista ou passageiro pode fazer uma requisição de compartilhamento de carona a partir de qualquer local e seus trajetos são inicialmente desconhecidos. Além

destas características, o *site* carpoolglobal.com lista algumas outras ideias comuns à concepção deste tipo de sistema, tais como uso de telefones celulares para localizar requisições e ofertas de compartilhamento de carona, serviços instantâneos de compatibilidade para caronas via serviço de dados, um serviço integrado para compensar o gasto do motorista durante a carona e, até mesmo, o conceito de “*multi-hop*”, que envolve uma troca de veículos que um passageiro faria para completar uma viagem (CARPOOLGLOBAL.COM, s.d.).

Apesar de a aplicabilidade técnica do *carpooling* dinâmico ser comprovada, segundo globalcarpool.com, o serviço é oferecido em baixa escala. Sghaier *et al.* (2011), por sua vez, afirmam que, dentre os poucos serviços envolvendo esta abordagem, grande parte não existia mais na época da publicação de seu artigo. Ademais, segundo estes autores, os serviços que ainda existiam, não obtiveram sucesso por não levar em conta, adequadamente, “questões de segurança”. Sghaier *et al.* (2011) apresentam um sistema chamado DOMARTiC, do inglês: “*Distributed Optimized approach based on the Multi-Agent concept for the implementation of a Real Time Carpooling service*”. Este se foca particularmente no aspecto dinâmico da representação geográfica dos elementos envolvidos ao passo que leva em conta questões como ferramentas automáticas para posicionamento geográfico e algoritmos de otimização no tratamento de requisições. Segundo os autores, ferramentas de localização geográfica, como o GPS, por si só, garantem a segurança dos usuários dada sua característica de rastreabilidade. A otimização de um sistema de *carpooling* dinâmico se faz necessária, segundo Sghaier *et al.* (2011), no tratamento das requisições minimizando o tempo das respostas. Para este fim, esses autores criaram um parâmetro chamado $\Delta\varepsilon$ que representa um intervalo de tempo que o sistema observa antes de processar uma requisição. Durante este tempo é possível que ocorram mais requisições paralelamente, as quais serão tratadas em uma só *query* para a formação de *carpools* logo que o intervalo

estipulado termine. Naturalmente, a variação de tempo utilizada é tal que usuários obtenham respostas as suas requisições o mais rapidamente possível para que o sistema mantenha sua característica de funcionamento em “tempo real”. Segue o modelo idealizado por Sghaier *et al.* (2011) para atender ao fluxo de requisições.

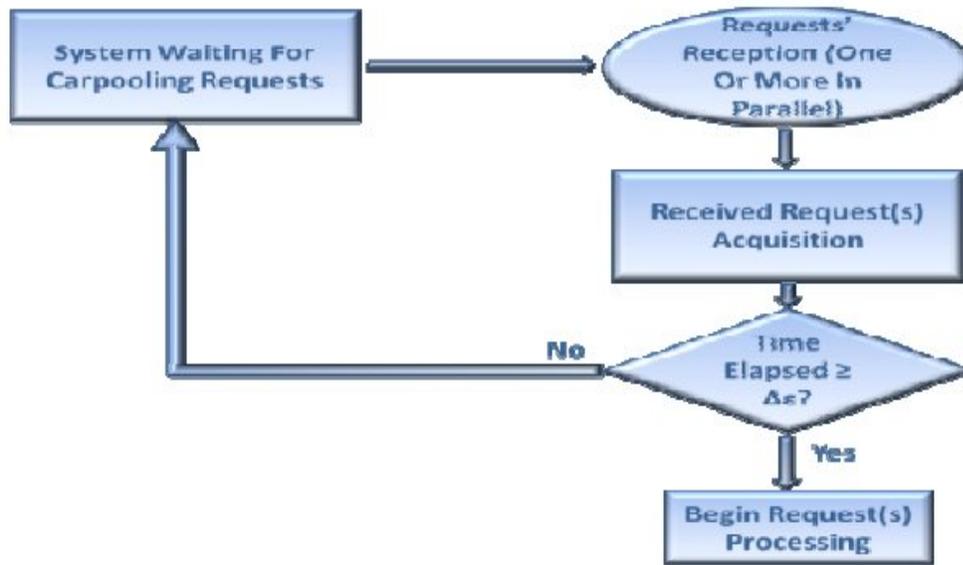


Figura 3. Processamento de solicitações em paralelo dentro de um sistema de *carpooling* em tempo real otimizado

Fonte: Sghaier *et al.* (2011)

A resposta à requisição consiste na representação de uma zona geográfica limitada, na qual o usuário se encontra, contendo os possíveis parceiros para *carpooling* (se houver algum) e seus trajetos no tempo corrente. Visto que os carros e passageiros estão se movendo, Sghaier *et al.* (2011) modelam o problema da representação em função do tempo (t), como segue.

$T(t) = (G(t), D(t), O(t))$ onde:

– $G(t) = (N(t), A(t))$, é um grafo direcionado:

$N(t)$ é o conjunto de nós baseado nas coordenadas geográficas correspondentes às origens, destinos e “destinos intermediários” de cada pedido ou oferta de viagem em um dado tempo t ,

$A(t)$ é um conjunto de arcos (ou arestas) direcionados relacionando os caminhos de origens até seus respectivos destinos. Os arcos referenciam o itinerário restante de cada veículo e requisições de rotas em função de destinos intermediários existentes.

- $D(t)$ é o conjunto de requisições (*demands*) de caronas feitas em um dado tempo t .
- $O(t)$ é o conjunto de ofertas (*offers*) de caronas feitas em um dado tempo t .

Os autores seguem formalizando matematicamente outras subdivisões do problema. Entretanto, visto que esta seção visa a apenas esclarecer o conceito geral do modelo sistêmico em questão – *carpooling* dinâmico – não cabe analisar todos os pormenores de um projeto específico, ainda porque pode haver variações na representação de um projeto para outro.

Ainda assim, é possível perceber um ponto em comum em todas as abordagens a respeito de *carpooling* dinâmico que foi levantado na fase de pesquisa – o uso do conceito de agentes, ou multi-agentes⁴, na análise e arquitetura dos sistemas.

O *design* dos agentes varia de acordo com a análise que acompanha a concepção de cada projeto. A seguir, apresentam-se as descrições de dois exemplos de *design* de agentes em sistemas de *carpooling* dinâmico.

⁴ Agentes e multi-agentes são termos encontrados dentro do escopo do paradigma de programação conhecido como AOP – do inglês, *agent-oriented programming*.

O primeiro é o sistema *Genghis* proposto por Kothari (2004). Este assume que todos os agentes dentro do MAS (*Multi-Agent System*) estão eternamente ativos e para cada usuário há um “*UserAgent*” correspondente. O sistema de Kothari (2004) é composto pelos seguintes agentes:

- *UserAgent*: representa um usuário humano e sua interação permitida com o sistema *Genghis*;
- *ProxyAgentN*: este agente atenderá a requisições HTTP e atuará como *middleware* entre o *container* JADE⁵ e a aplicação *web*. Podem haver *N* diferentes agentes caracterizados por esse papel;
- *JourneyRoundupAgent*: sinaliza jornadas que passaram do seu tempo final para fins de *feedback* humano;
- *JorneyNotifyAgent*: monitora as jornadas desejadas e ativas e sinaliza *UserAgents* caso surja algum *match*. A idéia aqui é que o humano seja notificado do resultado novo de uma pontuação de *matchings* e possa enxergar a opção específica de *match* correspondente. Pode-se optar por classificar este *match* dentro de um limite e dar-lhe um “sinal de trânsito”, acionando [este *match*] somente se a luz for verde. O ato de reservar um *match* que foi requerido previamente é revertido para um processo de reserva padrão.

O segundo modelo, proposto por Sghaier *et al.* (2011), mais conceitual e detalhado (com mais entidades), é descrito de acordo com as principais características e responsabilidades de cada entidade:

- *User Interface Agent* (UIA): criado para cada usuário conectado ao sistema e responsável por receber suas demandas e transmiti-las

⁵ JADE: *Java Agent DEvelopment Framework* é um *software*, inteiramente desenvolvido em linguagem Java, que tem por objetivo simplificar a implementação de sistemas multi-agentes (<http://jade.tilab.com/doc/html/intro.htm>).

ao agente de informação (IA). Depois que o processo otimizado é executado, ele provê as respostas geradas ao usuário;

- *Vehicle Interface Agent* (VIA): assegura o intercâmbio do motorista com os agentes envolvidos (*i.e.* IA respectivamente ao MA) transmitindo a oferta que propõem ou recebendo notificações de carona e endereço enviados;
- *Information Agent* (IA): deve prover o agente de decomposição com a informação necessária sobre as requisições de usuários e itinerários (*i.e.* ofertas de motoristas) que podem ser adaptáveis a tais requisições;
- *Decomposing Agent* (DA): tendo recebido a informação requerida (*i.e.* especificações de requisições e ofertas de carona) do IA, o DA é responsável por executar o princípio de subdivisão e, de acordo com este, determinar proximidade entre os nós e criar uma zona para cada grupo de vizinhos e um agente de otimização para cada zona estabelecida;
- *Optimizing Agent* (OA): “algoritmos de otimização executados localmente para buscar alocação conveniente e otimizada de veículos. A geração de respostas otimizadas é executada em paralelo, de forma descentralizada, de acordo com a arquitetura física distribuída definida por meio do processo de subdivisão [de área]. De fato, um *optimizing agent* é criado para cada área estabelecida pelo processo de decomposição e pelas requisições de usuário processadas localmente”.

Curiosamente, Sghaier *et al.* (2011) não fazem uma descrição categórica de MA, citada na descrição de VIA, assim como fazem para os demais agentes. No entanto, em um algoritmo de pseudo-código presente em seu trabalho, usa-se MA para referir-se ao termo “*Merging*”, portanto supõe-se que MA remonte a *Merging Agent*. Para entender o porquê da existência de MAs, é preciso descrever as zonas geográficas criadas

dinamicamente, já citadas anteriormente nesta seção. As zonas são circulares, sempre possuindo o mesmo perímetro, e podem estar interseccionadas entre si, dependendo da proximidade das localizações dos usuários do sistema. O MA possivelmente trata desta intersecção. O que reforça essa ideia é um diagrama de agentes no qual Sghaier *et al.* (2011) ilustram uma troca de informações entre MAs e OAs em ambos os sentidos.

Estes exemplos dão uma ideia da complexidade de implementação de um sistema de *carpooling* dinâmico, sem falar no custo em termos de tempo dedicado ao estudo para dominar o, relativamente novo, paradigma de orientação a agentes – que (dentre as obras levantadas) foi uma escolha unânime como abordagem para este tipo de problema.

2.2 Tecnologia GIS

De acordo com as intenções do estudo, foi feita uma breve análise sobre sistemas de informações geográficas a fim de averiguar a possibilidade de estes serem usados, de alguma forma, na implantação do Caronas-UTF. Nesta seção, a tecnologia GIS é abordada do ponto de vista conceitual e estrutural, além de serem classificadas categorias dentro de seu escopo. Finalmente, são analisados alguns exemplos de utilização deste recurso aplicado ao *carpooling*.

2.2.1 Conceito de GIS

GIS, um acrônimo para *Geographic (ou Geographical) Information System*, é um conjunto de dados e *softwares* utilizado para capturar, armazenar, analisar, gerenciar e apresentar dados geográficos, bem como analisar relações espaciais entre elementos (ou informações) e modelar processos espaciais (WIKI.GIS.COM, s.d.)

Na mesma área do conhecimento que abrange GIS existe uma ambiguidade para este acrônimo – GIS, remontando a *Geographic*

Information Science. No entanto, há trabalhos em que este último é referenciado por GISci, o que facilita a distinção entre a ciência e o sistema em si.

GISci é considerada “a ciência de processamento de dados espaciais” ou “a ciência por trás dos sistemas [GIS]”. Enquanto GIS trata de questões de *hardware*, *software* e uma combinação de gerenciamento de banco de dados e recursos gráficos, GISci se preocupa com a teoria subjacente ao desenvolvimento e aplicação dos GIS abrangendo também teoria de banco de dados, métodos de análise e técnicas de visualização (BRIEN, 2009).

Apesar de os dois campos estarem muito inter-relacionados, é preciso distingui-los, pois este estudo foca, sobretudo, nas possibilidades de implementação de um SI para compartilhamento de carona e nas possibilidades do uso de informações geográficas com o fim de criar grupos adequados a tal compartilhamento. Portanto, as seções seguintes deste capítulo tratarão de GIS apenas do ponto de vista sistêmico.

2.2.2 Enquadramento estrutural

Hoje há vasta quantidade disponível de dados geográficos, sejam resultantes de esforços nas áreas de cartografia e topografia, de estudos de impacto ambiental, da captura de imagens por satélites para variados fins, entre outros. Uma vez que se possa identificar, contextualizar e catalogar tais informações, cria-se um conceito de estrutura de dados. Há implementações destas estruturas, conhecidas como SDIs (*Spatial Data Infrastructure*, em português “Infraestrutura de Dados Espaciais”). (NEBERT, 2004).

O termo “*Spatial Data Infrastructures*” é por vezes usado para denotar uma coleção básica de tecnologias, políticas e arranjos institucionais para facilitar a disponibilidade e o acesso a dados espaciais. Uma SDI “provê uma base para descobertas e avaliações de dados espaciais e aplicações para

usuários e provedores [de conteúdo] para todos os níveis de governo, do setor comercial, de ONGs, da área acadêmica e de cidadãos em geral” (NEBERT, 2004, p. 8).

Ainda Segundo Nebert (2004), SDI é mais do que um conjunto ou uma base de dados, pois além das informações geográficas, a estrutura deve hospedar os atributos destas, assim como documentação suficiente sobre as informações (ou seja, metadados) e prover “um meio de descoberta, visualização e validação de dados (catálogos e *web mappings*) e um meio de prover acesso aos dados geográficos”. Nesse contexto aparece o GIS como um *software* para dar suporte à manipulação de tais dados.

Dentre os serviços de GIS estão projeção de mapa, definição de polígonos, definição de camadas de mapa, referenciamento de pontos em relação a uma superfície (*Datum* Geodésico), geoprocessamento etc. (WIKI.GIS.COM, s.d.).

Para realizar tais serviços, um GIS deve contar com um conjunto de protocolos, catálogos, metadados e uma base de dados espaciais. Tais itens conferem ao GIS um caráter sistêmico, diferenciando-o de uma simples aplicação. No entanto, não há uma convenção para se definir catálogos e protocolos, ficando esta tarefa a cargo de cada desenvolvedor, o que dificulta muito a integração entre SDIs (NEBERT, 2004).

2.2.3 Tipos de GIS

Eldrandaly (2007) classifica GIS em cinco principais categorias:

- *Desktop GIS software*: deve sua origem aos computadores pessoais (PCs) e ao sistema operacional *Microsoft Windows*. Provê ferramentas de produtividade pessoal para uma ampla variedade de usuários além das fronteiras do ramo insustrial. Esta categoria

abrange soluções desde simples visualizadores até mapeamento em *desktop*;

- *Server GIS*: roda em um computador servidor capaz de manejar requisições de processamento [de informações geográficas] concorrentes de um grupo de clientes conectados em rede. Inicialmente focado em exibir e buscar informações geográficas, atualmente também oferece serviços de mapeamento, roteamento, publicação de dados e adequação de mapas;
- *Developer GIS*: conjunto de ferramentas com funções GIS (componentes), que um programador com razoável conhecimento [na área de GIS] pode utilizar para criar aplicações de propósitos específicos. Tais componentes podem prover um alto nível de customização e otimização às aplicações, podendo ser executadas em *desktop* ou embutidas em outro *software*;
- *Hand-held GIS*: sistema leve, projetado para uso móvel e uso “em campo”. Recentemente se desenvolveu muito devido à disponibilidade dos chamados “*smart phones*”, capazes de portar um *high-end* (interface com o usuário final) para aplicações GIS, estes aparelhos podem lidar com uma quantidade relativamente grande de dados e aplicações sofisticadas. Os *hand-helds* geralmente operam com uma abordagem de alternância entre os estados “conectado” e “desconectado” para obter os dados geográficos;
- *Other types of GIS software*: outros *softwares* comerciais e não-comerciais de GIS com disposições de domínio-público, *open source* e *software* livre.

De acordo com dados de 2003, duas empresas norte-americanas lideravam a produção destes *softwares*: ESRI (*Environmental Systems Research Institute*) e *Intergraph Corporation* – que juntas cobriam quase metade de todo o mercado (47,1%) (ELDRANDALY, 2007).

Recentemente surgiu um serviço que vem se tornando popular, chamado de *Web GIS*, que converge algumas características das categorias citadas por Eldrandaly (2007). *Web GIS* basicamente é uma aplicação *web* (portanto, não um GIS em si) por meio da qual um usuário pode efetuar consultas geográficas de vários tipos e visualizar o resultado plotado sobre um mapa (DE OLIVEIRA, 2011). Alguns exemplos de serviços de *Web GIS* bem conhecidos estão disponíveis na Internet, tais como: *Google Maps*, *Yahoo Maps*, *MapQuest*, *Bing Maps* entre outros.

2.2.4 GIS aplicado ao *carpooling*

A tecnologia GIS tem um amplo potencial de utilização, podendo ser empregada em problemas cujo *status* pode variar de relativamente simples a crítico. O *site* www.appliedgis.net fornece uma grande quantidade de publicações a respeito da aplicação da tecnologia em várias situações. Neste *site*, pode-se encontrar artigos e teses que abordam a utilização de GIS como solução de problemas que variam desde a otimização de uso de vagas em estacionamentos até a previsão de impacto hidrológico causado pela agricultura em áreas florestais.

No caso de aplicações voltadas ao *carpooling*, encontram-se também exemplos, dentre os quais alguns foram descritos ou citados neste trabalho, na seção 2.1.2.

Um exemplo de aplicação é o portal *Smart Commute*, que utiliza o *Google Maps* para definir as rotas dos integrantes de *carpooling* cadastrados em seu *site*, bem como para buscar integrantes adequados para formar um *carpool*. Já o *site* Carona Brasil afirma utilizar uma “plataforma GIS” sem especificar um nome de aplicação ou sistema. Calvo *et al.* (2004), por sua

vez, afirmam que seu sistema utiliza uma base de dados geográfica⁶ para coletar mapas e dados referentes à rede rodoviária da região para a qual seu sistema foi projetado. O sistema então usa esses dados para gerar uma “saída GIS” em uma interface *web*.

2.3 Síntese do levantamento bibliográfico

O levantamento de informações envolvendo a prática de *carpooling* foi útil para se obter, primeiramente, uma noção de seu impacto, tanto sobre o trânsito como sobre a possível rotina dos adeptos da prática. Foi observada a necessidade de certo empenho individual e uma mudança de pensamento em relação ao deslocamento urbano, mas também se pôde notar as vantagens obtidas na adoção do *carpooling* como modo de transporte.

Foram detectadas três abordagens sistêmicas de *carpooling*, que podem ajudar a definir a lógica para o sistema almejado.

A tecnologia GIS, em diferentes contextos, provou-se útil na implementação de sistemas de *ridesharing*. No entanto, um ponto negativo constatado foi a dificuldade em obter detalhes a respeito do uso da tecnologia em alguns sistemas mencionados nesta seção, principalmente, e compreensivelmente, nos de cunho comercial. Por exemplo, em alguns casos não é possível saber se são usadas informações de localidade, distância ou rota para determinar o encaixe dos usuários em um *pool* (no caso do *carpooling* pré-arranjado). De qualquer forma, os dados levantados foram suficientes para assegurar que é possível a implementação do sistema Caronas-UTF com o uso de tecnologia GIS, bem como são suficientes para a definição da metodologia empregada ao referido sistema. O capítulo seguinte discorre sobre esta definição.

⁶ Base de Dados Geográficos ou *Geographic Database*: um banco de dados especializado para armazenar, consultar e manipular dados geoespaciais e seus atributos (espaciais ou não) (WIKI.GIS.COM, s.d.).

3 METODOLOGIA

Este capítulo se dedica à concepção do Caronas-UTF em termos de abordagens algorítmicas e de estrutura, pensando no papel que as tecnologias de geoprocessamento assumem no escopo do sistema. Tal concepção também visa a atender, da melhor forma possível, problemas de cunho não funcional – privacidade, confiabilidade de informações etc.

3.1 Análise dos tipos de *Carpooling*

A análise das características de cada um dos tipos de *carpooling* levantados na seção 2.1.2 ajuda a lançar luz na tomada de decisão sobre qual destes adotar. A abordagem sistêmica deve se enquadrar às características do padrão de deslocamento do público-alvo – alunos da UTFPR-CT.

Logo de início, pode-se notar que *carpooling* dinâmico não fornece uma estrutura de grupos (*pools*) adequada ao público-alvo. Neste caso, os grupos são formados em tempo real, à medida que o sistema localiza os membros cadastrados (onde quer que estejam) no intuito de atender a uma requisição de usuário. No entanto, os usuários-alvo têm um ponto de partida e de chegada pré-definidos, sendo o primeiro fornecido por meio de cadastro – provavelmente sua casa – e o último sendo um dos *campi* da UTFPR-CT (na volta, o destino pode ser o endereço inicial de origem ou um segundo endereço, também previamente cadastrado). Desta maneira, desenvolver um sistema para atender requisições e montar *pools* em tempo real seria acrescentar funcionalidades desnecessárias e complexas à solução do problema.

As outras duas modalidades de *carpool* parecem mais adequadas ao problema proposto, pois estabelecem o preceito de pontos de partida e

chegada definidos. No entanto, analisando-as, é possível enxergar ao menos um problema relevante em cada uma delas.

No caso do *carpooling* convencional (pré-arranjado), nota-se uma relação de dependência acentuada entre os membros de um *pool* que pode contribuir para o desencorajamento da adoção do sistema pelo público. Por exemplo, a possibilidade de ter que lidar com atrasos de parceiros de viagem – sejam motoristas ou passageiros – ou não ter seu veículo para um caso de emergência (no caso de alguém que abriu mão de utilizar o próprio veículo para compartilhar outro). Sem mencionar ainda o desvio de rota pelo motorista para coletar os demais passageiros do veículo compartilhado.

Já a implementação de um sistema de *carpooling* casual se depara com um sério problema de aspecto logístico. Nas cidades onde esta modalidade é praticada, os grupos se reúnem em locais públicos, geralmente estacionamentos, o que colabora para a diminuição da interdependência de membros do sistema. No entanto, em uma metrópole brasileira como Curitiba não seria fácil estabelecer um local para usar como ponto de encontro. Na verdade, a escassez de vagas em estacionamentos nesta cidade é uma das motivações deste estudo. Ademais não parece haver vias públicas que comportem um grande número de carros estacionados (para que seus donos esperem seus parceiros de viagem), sem comprometer o fluxo de veículos por elas. Também parece impraticável haver apenas um ponto de encontro sendo que os usuários do sistema podem morar em zonas totalmente divergentes em relação ao *campus* em que estudam. Talvez, uma solução seria diversificar os pontos de encontro baseando-se na distribuição geográfica dos usuários do sistema, reduzindo tanto o esforço dos usuários em chegar ao local como o número de veículos estacionados em cada ponto de encontro. Ainda assim, o processo para estabelecer locais de encontro apropriados continuaria sendo uma tarefa relativamente árdua e que parece caber mais às áreas de arquitetura e urbanismo do que à Ciência da Computação propriamente.

Sendo assim, apesar de algumas desvantagens, a modalidade de *carpooling* convencional (pré-arranjada) foi escolhida como abordagem sistêmica para este projeto.

3.2 Tecnologias Empregadas

Nesta seção serão listadas as tecnologias a serem utilizadas no sistema de informação. Estas estão classificadas quanto à forma como são empregadas – direta ou indiretamente. Esta classificação é importante para estruturação dos requisitos de sistema e para direcionar a implementação a tarefas objetivas. Segue, então, a descrição das tecnologias, separadas de acordo com a empregabilidade.

3.2.1 Tecnologias empregadas diretamente

São consideradas tecnologias de emprego direto aquelas que constituem parte funcional do trabalho e foram utilizadas diretamente pelo autor, seja na análise ou no desenvolvimento em si. Seguem:

- Linguagem de programação Java: usada para definir classes de acesso a dados (de um banco de dados) e nas classes de persistência de tais dados. Também utilizada para validar ações na camada de interface gráfica de usuário (GUI, do inglês) visto que código Java pode ser “embutido” no código HTML de páginas JSP, por meio de *scriptlets* e *expression language* (não foram usados *frameworks*). O controle de navegação e a lógica do negócio ficam a cargo de *servlets*, classes Java especializadas em tratar requisições de clientes *web*. O ambiente de desenvolvimento foi o NetBeans, em sua versão 7.0.1;
- Java Server Pages (JSP): utilizadas na camada GUI para fazer a interface com o usuário e tratar os atributos de requisição ou sessão criados nos *servlets*. Nas páginas JSP utilizam-se *tags* de *scriptlets* e *expression language* (E.L.);

- Linguagem SQL: utilizada para o acesso e a persistência de dados da base relacional dos dados que representam as entidades do mundo real – alunos, endereços, veículos etc. A ferramenta gerenciadora da base de dados propriamente dita foi o *MySQL Server*;
- Servidor de aplicações: o Apache Tomcat foi o servidor de aplicações Java, sendo o *container* dos *servlets* do projeto. O protótipo roda no *localhost*, portanto não foi montado um servidor *web*;
- Unified Modeling Language (UML): linguagem de modelagem utilizada na etapa de análise do sistema. A ferramenta CASE (*Computer-Aided Software Engineering*) usada na modelagem foi *Visual Paradigm*.

3.2.2 Tecnologias empregadas indiretamente

São consideradas tecnologias de emprego indireto aquelas que não foram utilizadas diretamente pelo autor. Ainda que tenham um papel funcional significativo, suas execuções não se localizam no sistema, mas são usadas como recursos deste. Seguem:

- GIS: o uso desta tecnologia é indireto, pois as funções de geoprocessamento ficaram a cargo de um *Web GIS*, especificamente o *Google Maps*, ou *gMaps*. De acordo com o *site* wiki.gis.com, a empresa fornecedora da tecnologia GIS utilizada no *Google Maps* é a *Tele Atlas*, uma multi-nacional cuja matriz está embasada nos Países Baixos (WIKI.GIS.COM, s.d.);
- SDI: esta tecnologia é utilizada de forma ainda mais indireta para o projeto, pois fornece dados à aplicação GIS, que já é utilizada de forma indireta. Visto que não foram encontradas especificações, ou mesmo citações sobre uma infraestrutura espacial específica da qual o *Google Maps* faz uso, o SDI é um módulo desconhecido ao

sistema proposto, embora se saiba que sua existência seja vital para o projeto;

- APIs de GIS: a Google fornece diversas APIs para que aplicações *web* e móveis possam utilizar recursos do gMaps. Dentre alguns destes recursos estão funções de acesso a imagens estáticas e dinâmicas de mapas, distância entre duas localizações geográficas, tempo de deslocamento de um trajeto (de acordo com o transporte utilizado), criação de rotas etc.

3.2.3 Tecnologias empregadas direta e indiretamente

Estas são tecnologias que em determinados momentos se aplicam aos quesitos do item 3.2.1 e em outros momentos se aplicam aos quesitos do item 3.2.2. Seguem.

- Linguagem XML (*Extensible Markup Language*): arquivos com dados de deslocamento são criados por um servidor da Google em resposta às requisições de localização ou trajeto feitas por um cliente *web*. Tais arquivos, em formato XML, são manipulados pelo sistema Caronas-UTF, que apenas extrai as informações das quais necessita para indicar pares adequados à formação de um *carpool* mediante à requisição de um usuário do sistema. Já o emprego direto desta tecnologia ocorre na estrutura do projeto do NetBeans, que utiliza dois arquivos XML (*web.xml* e *context.xml*), ambos relacionados ao servidor TomCat, com finalidades de configuração. O primeiro define o mapeamento dos *servlets* em relação às classes *servlets* criadas, o padrão da URL de acesso a este *servlet* no contexto do servidor, *timeout* de uma sessão, dentre outras funcionalidades. Já o segundo, define caminhos dentro do contexto de Catalina – diretório raiz do TomCat – para o acesso a recursos do projeto. No caso do Caronas-UTF, este arquivo é usado para definir um *pool* de conexões à base de dados.

3.2.4 Motivos para o emprego indireto de algumas tecnologias

A justificativa de deixar a implementação de GIS, bem como a construção de infraestrutura de dados espaciais, a cargo de aplicações externas ao sistema é basicamente o custo envolvido, em termos de dinheiro e tempo.

Apenas para dar uma ideia, menciona-se a variação de preço praticado pelas empresas fornecedoras de GIS em duas categorias da seção 2.2.3. Na categoria *Desktop GIS*, o preço de uma solução varia entre US\$1000,00 e US\$20000,00 por licença; um *Server GIS* pode custar de US\$5000,00 a US\$25000,00 (ELDRANDALY, 2007).

Já o custo de um SDI “é relativamente alto comparado ao custo de hardware e software requeridos para um GIS” (NEBERT, 2004). Além disso, não há necessidade de começar um SDI “do zero” para utilizar em uma aplicação como Caronas-UTF que não precisa de informações geográficas complexas (apenas endereços e eventuais cálculos de distância entre eles).

A figura a seguir mostra o encapsulamento das tecnologias envolvidas indiretamente no sistema.

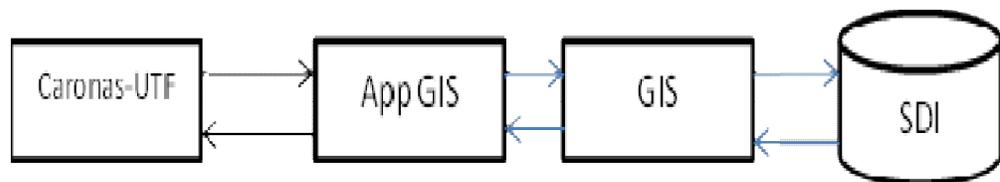


Figura 4. Fluxo de dados do SI

Fonte: autoria própria

As flechas de cor preta representam requisição e resposta de informações geográficas encapsuladas no protocolo HTTP (o *Google Maps*, que faz o papel da aplicação GIS na figura, também oferece o HTTP seguro, o HTTPS, como alternativa ao HTTP). Já a forma com que as informações

são trocadas entre App GIS e GIS, bem como GIS e SDI (setas azuis) são assuntos alheios ao conhecimento do Caronas-UTF.

3.2.5 Escolha de uma aplicação de *Web GIS*

A opção por uma aplicação *Web GIS* pode ser feita com base nos meios pelos quais tal aplicação fornece um dado serviço. Outras bases para uma decisão seriam o “encaixe”, ou conveniência, desta aplicação ao sistema que a utilizará, bem como a qualidade e confiabilidade do serviço da aplicação *Web GIS*.

Durante esta etapa foram analisados alguns aspectos de quatro aplicações populares deste tipo de ferramenta: *Bing Maps*, *Google Maps*, *MapQuest* e *Yahoo Maps*.

3.2.5.1 Qualidade e confiabilidade

Com respeito à qualidade dos serviços, apenas o *MapQuest* se mostrou falho na tarefa de localizar endereços quando não eram fornecidas informações completas. Por exemplo, ao tentar localizar o *campus* central da UTFPR, sem mencionar o nome – ou sigla – do estado (UF), era retornado um resultado na cidade do Rio de Janeiro, ainda que “Curitiba” fosse um dos parâmetros da busca. Os outros retornam o resultado correto ainda que informações como nome do bairro ou cidade estejam faltando. De fato, quando utilizando os serviços do *Google Maps* ou *Yahoo Maps*, por meio de um navegador, pode-se notar o recurso de “auto-completar” desses *sites*, sugerindo uma estrutura de relação semântica muito bem fundamentada.

Já a confiabilidade deve estar associada à precisão de localização dos endereços uma vez que a medição das distâncias entre os endereços cadastrados é importante para que o mecanismo de busca de pares do sistema funcione de forma eficiente. De acordo com Bakshi, Knoblock e Thakkar (2004), geralmente um GIS usa um processo chamado de

geocodificação para localizar um endereço. Tal processo consiste basicamente em converter um endereço real em coordenadas geográficas. O método mais tradicional de conversão usa uma fonte de dados baseada em vetores de ruas para obter uma coleção de endereços e as coordenadas do segmento de rua no qual determinado endereço está contido. Em seguida, uma técnica de aproximação é usada para estimar a localização do endereço desejado usando uma escala de coordenadas da rua que o contém. Tal método é ineficiente, pois pode coletar, por exemplo, escalas de endereço com numeração de 600 a 699 em um segmento que possui bem menos de cem endereços. Neste caso hipotético, para determinar a localização do endereço 625, as tecnologias disponíveis seriam capazes de identificar o lado ímpar do segmento, dividi-lo em 50 partes (os outros 50 seriam os pares) e atribuir a área fracionada adequada ao endereço 625. Esta é a técnica de aproximação mencionada. No entanto, sabe-se que propriedades em geral têm áreas diferentes umas das outras. Assim, tal limitação em determinar o endereço a ser geodificado acarreta em um erro médio de 36,85 metros (BASHKI, KNOBLOCK e THAKKAR, 2004).

Visto que todos *web GIS* utilizam fontes de dados com as mesmas técnicas e tecnologia (BASHKI, KNOBLOCK e THAKKAR, 2004), não há um diferencial em relação à qualidade do serviço por eles oferecido. Mas, neste aspecto, o *Google Maps* chama a atenção por possuir em sua API *Geocode* a classe *GeocoderLocationType*, responsável por determinar níveis de precisão de uma dada coordenada. Os níveis são classificados em (DEVELOPERS.GOOGLE.COM, s.d.):

- *APPROXIMATE*: indica que o valor retornado é aproximado;
- *GEOMETRIC_CENTER*: indica que o valor retornado é o centro de uma linha (exemplo: rua) ou polígono (região);
- *RANGE_INTERPOLATED*: indica que o valor encontra-se em uma intersecção de dois pontos precisos;

- *ROOFTOP*: indica que o valor retornado reflete uma coordenada “precisa”.

Sem deixar de reconhecer a boa funcionalidade de outras aplicações *web GIS*, adotou-se para este projeto o uso das APIs da *Google Maps*.

3.2.6 Utilização de APIs do *Google Maps*

Foram utilizadas três APIs neste trabalho: *Geocode*, *Directions* e *Staticmap*.

A forma de acesso aos serviços das APIs é feita via requisições HTTP ou HTTPS. A aplicação faz chamadas no seguinte formato: [https://maps.googleapis.com/maps/api/\[nome_api\]/\[output\]?parameters](https://maps.googleapis.com/maps/api/[nome_api]/[output]?parameters).

O endereço de acesso é <https://maps.googleapis.com/maps/api/>, seguido do nome da API da qual se requer o serviço. O *output* refere-se ao tipo de arquivo de saída enviado no HTTP/HTTPS *response* (DEVELOPERS.GOOGLE.COM, s.d.). O gMaps suporta arquivos do tipo JSON e XML em seu *response*. O Caronas-UTF utiliza a segunda opção. E, por último, os parâmetros de entrada.

As três próximas seções descrevem as APIs utilizadas pelo sistema implementado, bem como seus parâmetros de entrada e seu arquivo de saída.

3.2.6.1 Geocode

Esta API fornece serviços referentes à geolocalização de endereços, provendo coordenadas geográficas (latitude e longitude) para um endereço físico bem como o processo contrário (*reverse geocoding*).

A API recebe os seguintes parâmetros obrigatórios:

- *address* ou *latlng* ou *components*: o endereço a ser geocodificado, ou geodecodificado, se o parâmetro for *latlng*. Neste segundo caso passa-se a latitude e longitude separadas por vírgula. Já *components* são filtros que restringem a busca filtrando por uma especificação de um ou mais componentes de endereço. Os seguintes componentes fazem parte da estrutura do arquivo de resposta:
 - *route*: nome de uma rota;
 - *locality* e *sublocality*: geralmente cidade/província e bairro
 - *administrative_area*: pode representar unidades federativas, estados, comunidades autônomas etc. Em geral tem uma ocorrência e é apresentado, na resposta, como tipo *administrative_area_level_1*, mas se a hierarquia administrativa do lugar for mais complexa, pode haver ocorrências de componentes *administrative_area_level_2* e assim por diante;
 - *postal_code* e *postal_code_prefix*: contêm valor do CEP em uma busca feita no Brasil;
 - *country*: especifica o país por meio de uma sigla de duas letras, exemplos: Brasil: BR, Espanha: ES, Portugal: PT etc.

Vale mencionar que o *components* pode ser utilizado como parâmetro opcional se na requisição estiver especificado o endereço (*address*);

- *sensor*: indica se a requisição vem a partir de um dispositivo móvel (assume valor: *true*) ou não (assume valor: *false*).

Parâmetros opcionais:

- *bounds*: especifica uma caixa delimitadora na imagem com o fim de destacar a área do endereço encontrado.
- *language*: especifica o idioma no qual os resultados aparecerão. Para este parâmetro ser válido, o idioma deve estar em uma lista

de idiomas suportados encontrada na documentação do site developers.google.com;

- *region*: o código ccTLD (*country code top-level domain*), além de funcionar como filtro, obtém o resultado de acordo com as unidades medidas usadas na região especificada. Note-se porém que a API usa *region* como uma “preferência”, ou seja, se forem encontrados resultados mais relevantes fora da região estes serão incluídos;
- *Components*: já descritos. São especificados utilizando-se de um “|”(*pipe*) como separador.

Os dados de saída apresentam informações de componentes de endereço (*address_components*), localização geográfica (*lat,lng*) incluindo o tipo de precisão (*location_type*), propriedades de visualização do mapa (*viewport*) e limites geográficos do mapa (*bounds*).

No Caronas-UTF, a API *Geocode* é utilizada para converter os endereços físicos cadastrados pelos alunos em coordenadas geográficas, tanto por ocasião do cadastro em si, como em casos de alteração de endereço ou inclusão de um novo endereço.

3.2.6.2 Directions

A API *Directions* do *Google Maps* fornece dados de origem, destino e pontos intermediários (*waypoints*) de um trajeto.

Parâmetros obrigatórios:

- *origin*: o endereço de origem, físico ou o valor de latitude e longitude, sem espaços em branco;
- *destination*: o endereço de destino, físico ou o valor de latitude e longitude, sem espaços em branco;

- *sensor*: indica se a requisição vem a partir de um dispositivo móvel (assume valor: *true*) ou não (assume valor: *false*).

Parâmetros opcionais:

- *mode*: especifica o tipo de transporte ao calcular uma rota, por *default* assume o valor *driving* (dirigindo um veículo). Também aceita os valores: *walking*, *bicycling* ou *transit* (para transporte público, quando disponível);
- *waypoints*: especifica um conjunto de pontos intermediários para o trajeto. Cada ponto é referenciado como um endereço, da mesma forma que a origem e o destino, e os pontos são separados pelo caractere “|” (*pipe*). Este parâmetro não é suportado quando o *mode* tem valor *transit*;
- *alternatives*: quando assume o valor *true* a API *Directions* pode retornar mais de uma rota;
- *avoid*: utilizado para indicar que a rota deve evitar pedágios e/ou alto-estradas;
- *language*: especifica o idioma no qual os resultados aparecerão. Para este parâmetro ser válido, a linguagem deve estar em uma lista de linguagens suportadas encontrada na documentação do *site* developers.google.com;
- *units*: especifica o sistema de unidades a ser exibido na resposta (métrico ou imperial);
- *region*: o código ccTLD (*country code top-level domain*) da região;
- *departure_time* e *arrival_time*: podem ser usados quando o parâmetro *mode* tem o valor *transit* ou *driving*. No primeiro caso deve-se especificar um dos dois (em segundos a partir da data 1º de janeiro, 1970 UTC) para que a API calcule o outro. Por exemplo, quando passada a hora de partida a resposta virá com a hora de chegada. No segundo caso se dá o mesmo, mas na versão comercial (*Maps for Business*) é possível receber dados com as

condições do trânsito no período determinados por *departure_time* ou *arrival_time*.

O retorno da API traz as distâncias e tempos de duração de forma ponto-a-ponto dentro de *tags* chamadas *legs*. Cada *leg* contém um conjunto de *tags* “*step*” que detalha o percurso, por exemplo: “virar a direita na rua [nome_da_rua]”. Além dos *steps* cada *leg* contém informações de tempo (*duration*) e distância (*distance*).

Para cada análise de *match* o CaronasUTF chama *Directions* duas vezes: uma vez utilizando o endereço inicial e final do itinerário do aluno⁷ como origem e destino, respectivamente; e outra vez com os mesmos dados somados aos endereços intermediários, fazendo a soma dos valores de distância de cada *leg* para obter a distância total do trajeto, para então utilizá-la no cálculo de percentual de acréscimo de rota, em relação à rota direta que o motorista faria sem buscar nenhum passageiro.

3.2.6.3 Staticmaps

Esta API fornece mapas estáticos, ou seja, mapas em formato de imagem estática (*non scrollable*). É utilizada no projeto na ocasião em que um aluno acha um par com o qual ainda não compartilha nenhum trajeto. Esse modelo é apropriado por ser menos detalhado no que diz respeito a informações de endereço do par retornado na pesquisa, já que nesta etapa o aluno requisitante da busca tem a escolha de não convidá-lo para um compartilhamento de carona.

Parâmetros Obrigatórios:

⁷ O sistema proposto é flexível de modo a permitir ao aluno criar um itinerário de ida (de um endereço particular ao *campus*) bem como de volta (de um *campus* para um endereço particular), além de um itinerário envolvendo um deslocamento *intercampi*.

- *size*: define o tamanho das dimensões retangulares da imagem do mapa;
- *sensor*: indica se a requisição vem a partir de um dispositivo móvel (assume valor: *true*) ou não (assume valor: *false*).

Parâmetros Opcionais:

- *scale*: afeta a quantidade de *pixels* retornada;
- *format*: o formato da imagem resultante. Por *default* assume o formato PNG, mas suporta outros formatos dentre eles GIF e JPEG;
- *maptypes*: define o tipo de mapa a ser apresentado: *roadmap*, com informações de ruas; *satellite*, exibe imagem de satélite; *hybrid*, inclui as características dos dois anteriores no mapa; e *terrain*, apresenta representações de características físicas do terreno;
- *language*: especifica o idioma no qual os resultados aparecerão. Para este parâmetro ser válido, o idioma deve estar em uma lista de idiomas suportados encontrada na documentação do site developers.google.com;
- *region*: o código ccTLD (*country code top-level domain*) da região;
- *markers*: define marcadores para plotar no mapa em um endereço, físico ou geográfico, especificado;
- *path*: define um único caminho entre dois ou mais pontos de um mapa;
- *visible*: usado para garantir que um ou mais lugares esteja visível no mapa, ainda que nenhum marcador tenha sido especificado;
- *style*: define um conjunto de regras para customizar o estilo de um mapa;
- *center*: define o centro do mapa baseado em um endereço físico ou em coordenadas geográficas. Se nenhum *marker* foi definido, este parâmetro passa a ser obrigatório;

- *zoom*: determina o nível de ampliação do mapa retornado. Se nenhum *marker* tiver sido definido, este parâmetro passa a ser obrigatório.

O CaronasUTF utiliza especialmente os parâmetros *zoom* e *maptype*, com o valor “*terrain*”, para fins de privacidade, buscando não dar claros indícios do endereço de alunos na etapa que precede o envio de convite ou troca de mensagens.

O retorno HTTP dessa API é uma imagem contendo o mapa especificado. Para utilizar a imagem basta referenciar a URL de chamada da API como fonte (“src”) em uma *tag* HTML de imagem, como a seguir:

```

```

3.3 Síntese da metodologia

A escolha da categoria convencional (ou pré-arranjada) de *carpool* e o processamento de informações geográficas deixado a cargo de um WebGIS foram fatores determinantes para a modelagem final do sistema. O fluxo de dados e a representação das informações físicas e geográficas foi mostrado superficialmente na seção 3.2.4. No capítulo seguinte veremos detalhes da troca de informações entre os módulos da Figura 4 na seção mencionada, bem como as regras adotadas para formação de *carpools* com base nestas informações. Além disso, é feito uma análise detalhada do sistema Caronas-UTF em si.

4 RESULTADOS

Este capítulo destina-se a relatar o resultado final do trabalho, fruto de toda a fundamentação teórica construída até então. O produto final pode ser dividido em duas partes. A primeira é a análise formal do sistema Caronas-UTF, já que, do ponto de vista da Engenharia de *Software*, esta prática está agregada ao produto. A segunda envolve o programa em si e diz respeito à estrutura do código-fonte e aos padrões e metodologias utilizados na sua produção.

4.1 Análise do Sistema

Esta seção abrange os requisitos funcionais e não-funcionais para o sistema. Também fornece uma visão global do sistema e da sua interação com o usuário. Representações das entidades do mundo real e de procedimentos específicos são exibidos na forma de tabela, figura ou um diagrama apropriado.

4.1.1 Levantamento de requisitos

Seguem os requisitos do sistema, separados em duas listas de acordo com o impacto, direto ou indireto, que estes têm na funcionalidade do programa.

4.1.1.1 Requisitos funcionais

Segue a lista com requisitos que impactam no sistema em si, em termos de código e estrutura lógica.

- Mecanismo de acesso de usuários por meio de *login*, utilizando controle de sessão;
- Mecanismo de registro de usuários no sistema (cadastro);

- Mecanismo de gerenciamento de dados pessoais do aluno, incluindo endereço, veículos (quando houver algum), itinerários e *carpools* vinculados ao aluno;
- Mecanismo de controle de informações referentes aos *campi* de acesso restrito a um administrador do sistema;
- Mecanismos de validação e controle de informações fornecidas por usuários;
- Controle de navegação (das páginas *web*) de acordo com requisições do usuário;
- Mecanismo de busca por parceiros de *carpool*, mediante solicitação de usuário;
- Mecanismo para troca de mensagens entre usuários para promover o estabelecimento de um *carpool*;
- Mecanismo de notificações para criação, alteração ou deleção de informações relacionadas a um *carpool*.

4.1.1.2 Requisitos não-funcionais

Segue a lista com requisitos de caráter mais subjetivo ao sistema.

- Confiabilidade dos dados fornecidos;
- Privacidade;
- Segurança de dados pessoais (endereço).

4.1.2 Diagrama de estados do sistema de informação

Com o fim de proporcionar uma visão geral do fluxo do programa de acordo com suas funcionalidades, segue a estrutura de grafos representando estados do sistema.

4.1.3 Casos de uso

Nesta seção são detalhados alguns casos de uso como um complemento à rede de grafos do sistema. Os casos de uso referentes às funções do administrador em relação aos dados dos *campi* foram omitidos por não serem essenciais à compreensão do sistema. Os casos relativos ao gerenciamento de informações do aluno não são detalhados em nível de operações de banco de dados (criação, alteração, deleção e leitura), mas como casos agregadores do caso de uso abstrato “Gerencia Dados UC”.

4.1.3.1 Cadastro

Tabela 1. Caso de uso CadastroUC

Nome	CadastroUC
Ator	Aluno
Descrição	Aluno faz cadastro no sistema por meio de interface <i>web</i>
Pré-condições	Nenhuma
Pós-condições	Dados validados
Fluxo básico	Confirmação por <i>e-mail</i>
Fluxo alternativo	Mensagem de erro em um ou mais dos dados cadastrados

4.1.3.2 Login

Tabela 2. Caso de uso LoginUC

Nome	LoginUC
Ator	Aluno
Descrição	Aluno faz <i>login</i> no sistema por meio de interface <i>web</i>
Pré-condições	Aluno cadastrado
Pós-condições	Login validado
Fluxo básico	Criação de sessão para usuário
Fluxo alternativo	Mensagem de <i>login invalid</i>

4.1.3.3 Logout

Tabela 3. Caso de uso LogoutUC

Nome	LogoutUC
Ator	Aluno
Descrição	Aluno faz <i>logout</i> no sistema por meio de interface <i>web</i>
Pré-condições	Aluno “logado”
Pós-condições	Nenhuma
Fluxo básico	Retirar a sessão do usuário
Fluxo alternativo	Mensagem de erro (comunicação com o servidor de banco de dados, por exemplo)

4.1.3.4 Gerenciamento de dados

Tabela 4. Caso de uso GerenciamentoDeDadosUC

Nome	GerenciamentoDeDadosUC
Ator	Aluno, BD
Descrição	Aluno altera (cria, apaga, atualiza) ou lista informações pessoais, de <i>carpoolings</i> , de itinerários ou de mensagens e convites
Pré-condições	Aluno “logado”
Pós-condições	Alterações validadas
Fluxo básico	Alterar dados
Fluxo alternativo	Mensagem de erro – ou no BD, ou nas informações

4.1.3.5 Busca

Tabela 5. Caso de uso BuscaUC

Nome	BuscaUC
Ator	Aluno, BD e gMaps
Descrição	Aluno busca pares apropriados com base em seus endereços e

	preferências de deslocamento, para um determinado itinerário
Pré-condições	Aluno “logado”, servidor do Google Maps operando sem problemas
Pós-condições	Nenhuma
Fluxo básico	Listar pares apropriados
Fluxo alternativo	Mensagem de que nenhum par apropriado foi encontrado

4.1.3.6 Envio de mensagem

Tabela 6. Caso de uso MensagemUC

Nome	MensagemUC
Ator	Aluno, BD
Descrição	Aluno envia uma mensagem a um par encontrado por meio da busca do sistema
Pré-condições	Aluno “logado”, a busca por pares ter encontrado ao menos um par
Pós-condições	Mensagem salva em BD
Fluxo básico	Enviar mensagem e enviar notificação desta por e-mail ao usuário destinatário
Fluxo alternativo	Exibir mensagem de erro

4.1.3.7 Envio de Convite

Tabela 7. Caso de uso ConviteUC

Nome	ConviteUC
Ator	Aluno, BD
Descrição	Aluno envia um convite a um par encontrado por meio da busca do sistema
Pré-condições	Aluno “logado”, a busca por pares ter encontrado ao menos um par
Pós-condições	Convite salvo em BD
Fluxo básico	Enviar mensagem e enviar notificação desta por e-mail ao usuário destinatário
Fluxo alternativo	Exibir mensagem de erro

4.1.3.8 Visão geral dos casos de uso

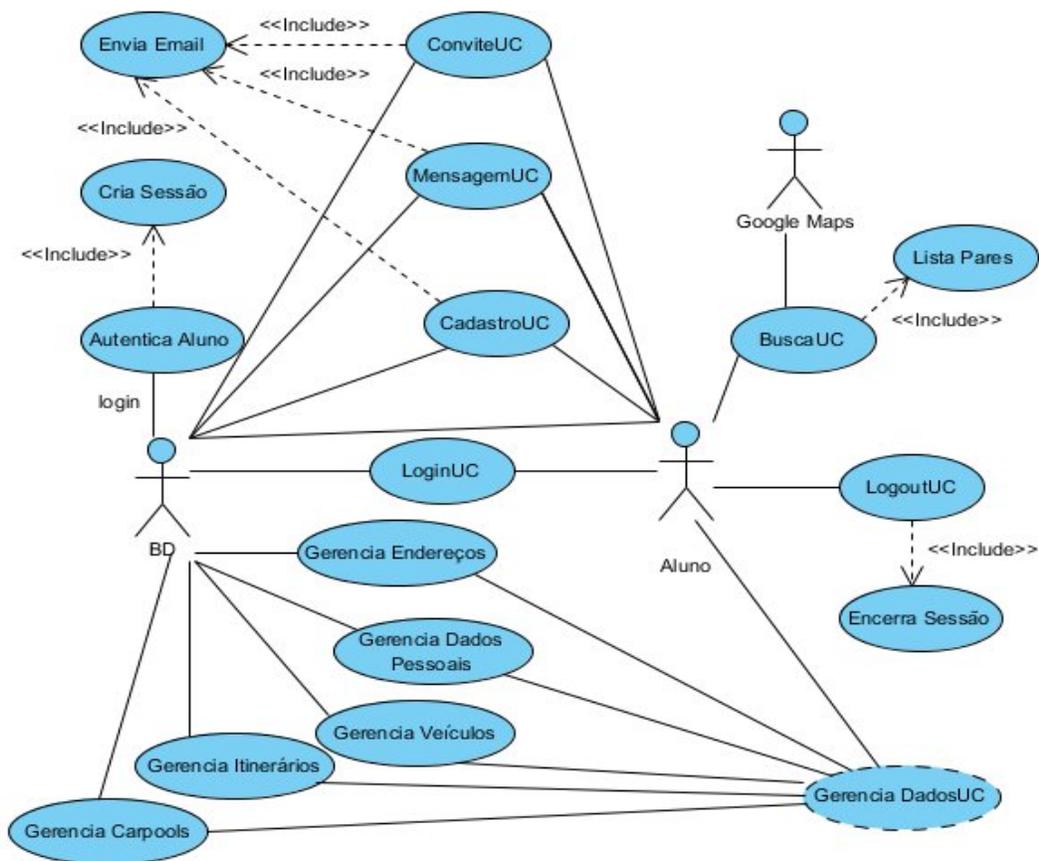


Figura 6. Visão Geral dos Casos de Uso

Fonte: autoria própria

4.1.4 Diagrama de máquina de estados de convites

Como será visto na representação do modelo entidade-relacional no subtópico seguinte, há entidades auxiliares que servem para relacionar outras entidades entre si. Uma dessas entidades auxiliares é *convitedealuno*, que armazena os *status* dos convites criados mediante solicitações feitas por usuários para criação de um *carpool*. Para o sistema, um convite é válido ou “utilizável” quando se encontra no estado “pendente”. E, ainda que a entidade *convite* esteja relacionada com *aluno* (em uma

relação 1 para 2, representado por *convitedealuno*), para o sistema, um convite é único no que se refere ao seu estado.

Para melhor visualização, segue o diagrama da máquina de estado de um convite, de acordo com as ações dos usuários.

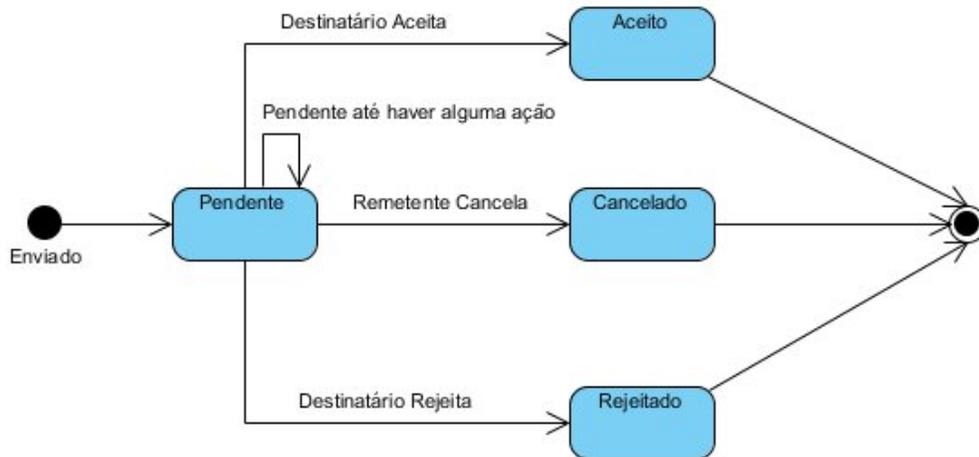


Figura 7. Máquina de Estado de Convites

Fonte: autoria própria

Para facilitar a compreensão do que o usuário enxerga, segue uma tabela que diz respeito à interação do sistema em cada situação, e uma explicação posterior.

Tabela 7. Interação do SI segundo o estado de um convite

Ação	Status Atual do Convite (R)	Status Atual do Convite Convite (D)	Interação do Sistema	Visibilidade do Convite ao Final da Ação (visível para)
Convidar (R)	Pendente	Pendente	Envio de E-mails	Ambos
Aceitar (D)	Aceito	Aceito	Envio de E-mails	Nenhum
Aceitar (D)	Cancelado	Rejeitado	Mensagem de Erro	Nenhum
Cancelar (R)	Cancelado	Pendente	Mensagem de Confirmação	Destinatário
Cancelar (R)	Cancelado	Rejeitado	Mensagem de Confirmação	Nenhum
Rejeitar (D)	Pendente	Rejeitado	Mensagem de Confirmação	Remetente
Rejeitar (D)	Cancelado	Rejeitado	Mensagem de Confirmação	Nenhum

Os caracteres “R” e “D” representam, respectivamente, o remetente e o destinatário do “convite único” (entidade *convite*). Células contendo texto

em azul indicam que aquele estado (*status*) específico foi alterado sem uma intervenção do usuário. Por exemplo, na segunda linha, segunda coluna, o sistema altera o *status* do convite do remetente de “pendente” para “aceito”, baseado em uma ação do destinatário (aceitar o convite).

Na tabela estão omitidas as intervenções do sistema em casos de dois convites relativos ao mesmo itinerário e como o usuário age em função do tempo. Este tipo de situação pode ser explanada da seguinte forma: dados os usuários U_1 , U_2 onde o primeiro é o remetente e o outro destinatário de um convite C , U_1 pode compartilhar um veículo para o itinerário vinculado ao convite C (seja mediante envio ou aceitação de outro convite) antes de U_2 aceitar o convite C . Neste caso, o convite C assume um *status* de “cancelado”, de modo que, se U_2 aceitar o convite, o sistema retorna uma mensagem de erro (de acordo com a tabela acima). De forma semelhante, quando U_2 engaja-se a um outro *carpool* vinculado ao itinerário referente ao convite C , o *status* de C torna-se “rejeitado”.

4.1.5 Modelo entidade-relacional

O modelo da base de dados do sistema é de natureza relacional e utiliza a linguagem SQL. Segue a representação do MER.

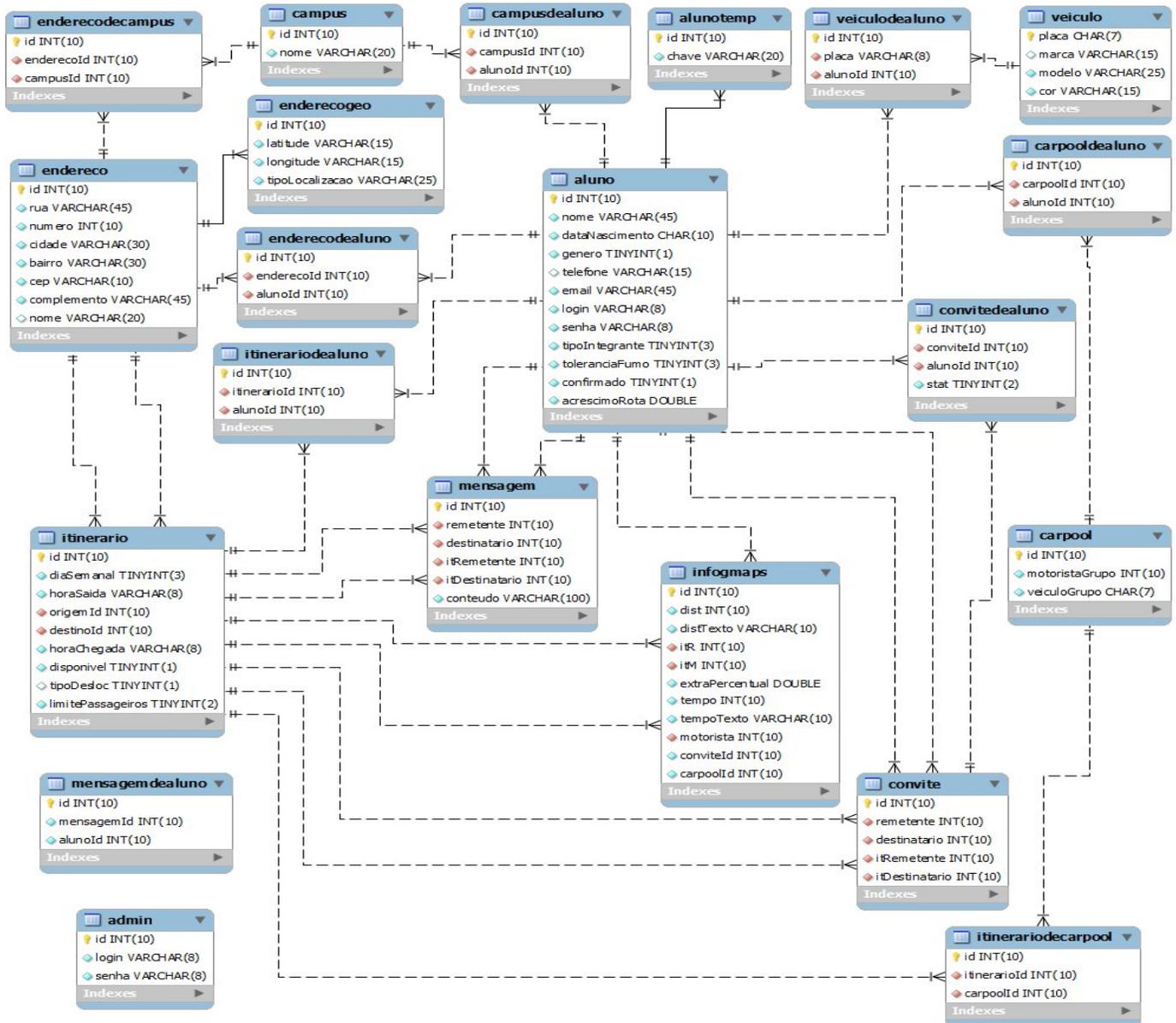


Figura 8. Modelo entidade-relacional

Fonte: autoria própria

O Caronas-UTF tem como entidades centrais *aluno*, *campus*, *endereco* e *itinerario*. Em torno destas quatro entidades, há tabelas responsáveis por armazenar informações complementares a elas. Por exemplo, *mensagem* e *convite* têm o objetivo de representar relações de contato entre usuários do sistema. Além destas, há a tabela *carpool*, que

apesar de não estar entre as entidades ditas centrais, tem o importante papel de convergir informações sobre grupos que se formam por meio do sistema, a fim de compartilhar um veículo.

Outras entidades são *admin* (já que o sistema prevê um administrador para cadastrar os *campi*), *infogmaps*, que armazena informações (retornadas do Google *Maps*) sobre itinerários coincidentes, como distância e motorista indicado para o trajeto.

As demais entidades são de natureza auxiliar, servindo de relação entre duas outras entidades. São exemplos: *convitedealuno*, *enderecodealuno*, *enderecodecampus*, *itinerariodealuno* etc. Ainda que *mensagemdealuno* não seja vista relacionada com *mensagem* por meio de chave estrangeira neste modelo, ela leva a informação do campo *id* de *mensagem* (chave primária desta) para relacionar um aluno a uma mensagem. Optou-se por definir esta relação específica sem uso de chave estrangeira por questões de facilitação de visibilidade da mensagem pelo usuário. Por exemplo, se um usuário destinatário exclui uma mensagem recebida, esta ainda é mantida na caixa de mensagens enviadas do usuário remetente. Também vale destacar a entidade *endecogeo*, concebida como um complemento para *endereco*. Ela guarda informações de geolocalização de um endereço (latitude e longitude) que ajudam o sistema a determinar se um par tem um posicionamento ideal para formar um *carpool*.

4.2 Desenvolvimento do Sistema

Nesta seção são apresentados aspectos relacionados aos padrões, paradigma e metodologia do sistema implementado. Depois os componentes do Caronas-UTF são mostrados em uma forma ainda mais estrutural, no sentido de como estão dispostos no contexto de um servidor de aplicações – neste caso, o TomCat. O capítulo termina com uma explicação sobre a lógica de busca utilizada na formação dos possíveis pares. Deve-se dizer “possíveis” porque, mesmo após o sistema relacionar um ou mais pares

adequados para o aluno que efetuou a busca, o *carpooling* ainda não está efetivado, pois necessita do envio de convite deste aluno e posterior confirmação do destinatário do convite.

Todas as classes mencionadas neste trabalho são classes Java. Por questão de conveniência a terminação “.java” foi suprimida nas menções a elas, grafando-se apenas seus nomes iniciados com letra maiúscula e em itálico para destacá-las do resto do texto.

4.2.1 Paradigma e metodologia de programação

Como dito na seção 3.2, o programa foi escrito em linguagem Java. Portanto, o paradigma de programação não poderia ser outro que não o da Programação Orientada a Objetos (OOP, da sigla em inglês). O Caronas-UTF é baseado em classes, estando estas divididas de acordo com sua funcionalidade dentro do padrão MVC – *Model View Controller*.

Internamente ao padrão MVC foi implementada uma outra metodologia, DAO (*Data Access Object*), para gerenciar aspectos relacionados com o acesso à base de dados. As classes DAO são responsáveis por funções que abrangem desde a obtenção da conexão com a base de dados até a sua persistência e atualização. Classes DAO também podem ser usadas para mapear objetos Java para tipos de dados SQL, mas essa função não foi aplicada neste trabalho em particular. Do ponto de vista da metodologia DAO as classes de dados são conhecidas como DTO – *Data Transfer Object*, classes básicas que guardam os atributos das representações das entidades do sistema contidas na base de dados. As classes DAO – geralmente providas de métodos de projeção (seleção), inserção, alteração e deleção – usam as classes DTO para alterar ou retornar dados da base de dados. Portanto, as classes DTO atuam como parâmetros de entrada ou saída (dependendo da situação) dos métodos das classes DAO.

A Figura 9 dá uma visão geral da disposição dos elementos envolvidos no sistema.

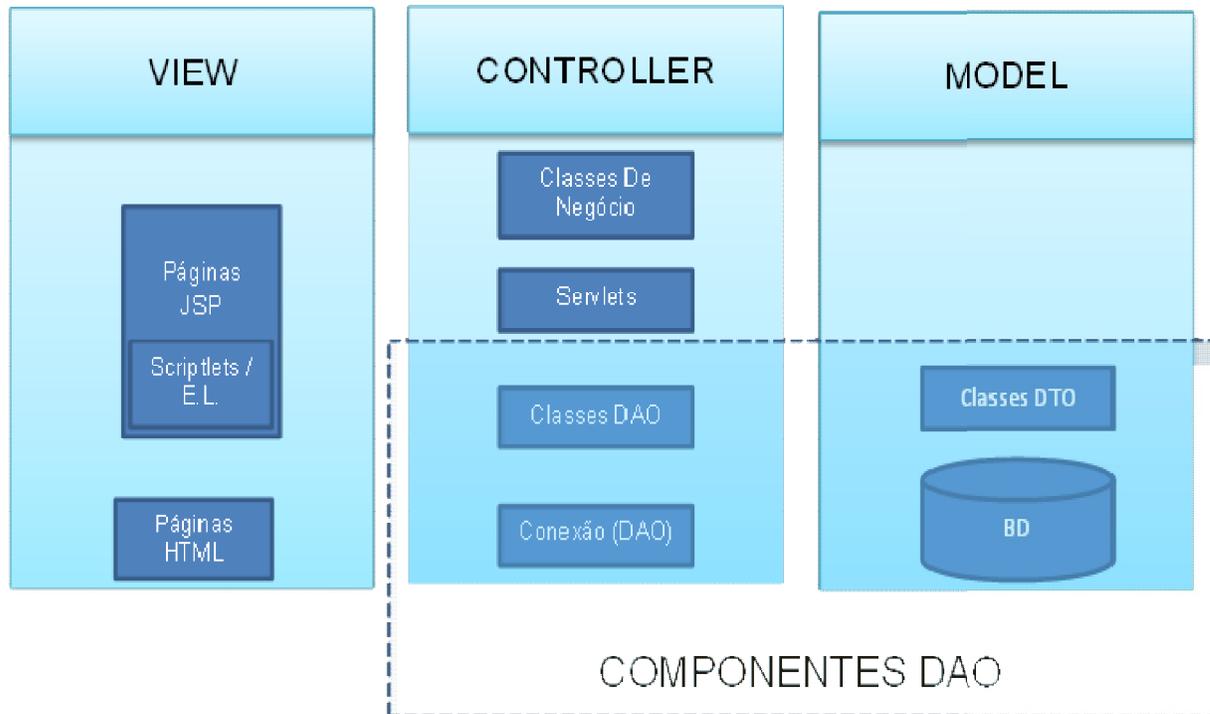


Figura 9. Componentes do Sistema de Acordo com a Metodologia

Fonte: autoria própria

4.2.2 Estrutura

As classes do modelo são representações das entidades contidas no banco de dados. As classes limitam-se a conter campos equivalentes aos atributos da entidade correspondente e métodos para manipular e acessar os valores desses campos – *getters* e *setters*. Adicionalmente, há três classes de dados. Uma para persistir informações referentes a um trajeto específico de alunos de um *pool* chamada *InfoGMaps*. As outras duas são representações de mensagens, uma carregando as mensagens padrão que devem ser enviadas para um ou mais usuários por *e-mail*, de acordo com a ação dos usuários (por exemplo, ter um convite para *carpool* ou ter recebido uma mensagem referente a um *carpool*). Basicamente, esta classe,

chamada *MensagemDeEmail*, funciona como auxiliar da classe *Notificador* (descrita no próximo subtópico). A outra classe representa mensagens de sistema, ou seja, é responsável por montar respostas de *feedback* às ações do usuário, quando necessário, informando se houve sucesso ou fracasso para determinadas requisições. Os objetos desta classe escrevem um *output em HTML* após determinadas ações do usuário contendo a resposta. Em caso de erro, podem também aconselhar algum tipo de procedimento e um *link* para um lugar apropriado, de acordo com a ação e seu resultado.

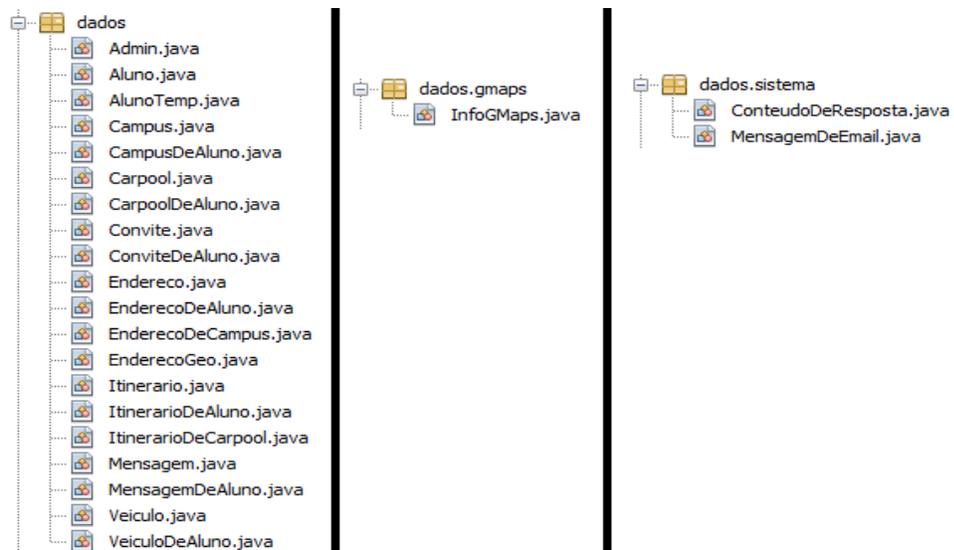


Figura 10. Pacote e Subpacotes das Classes de Dados (*Model*)

Fonte: autoria própria

O módulo de controle é constituído de classes de acesso a dados (DAO), descritas anteriormente nesta seção – incluindo a classe *Conexao*, vista como uma classe utilitária no modelo DAO – classes *servlets*, que tratam as requisições de um cliente *web* e seis classes específicas: *AuxiliarDeItinerario*, *Busca*, *CalculoIdade*, *Geocodificador*, *XMLUpdater* e *Notificador*.

Cinco destas são meramente utilitárias:

- *AuxiliarDeItinerario*: fornece um método para formatar a “assinatura” de um itinerário (chamado na sobrecarga de *toString()* da classe *Itinerario*) e outro método para determinar o tipo do deslocamento envolvido neste itinerário: “campus_campus”, “campus_endereço” e “endereço_campus”. Basicamente, determina se o trajeto é de: ida, volta ou *intercampi*. Leia-se “endereço” como um endereço cadastrado pelo aluno (em geral, sua casa). Note-se que o sistema não permite a possibilidade de um trajeto hipotético “endereço1_endereço2”. O aluno deve incluir ao menos um *campus* no itinerário, a fim de que o propósito do sistema não se perca;
- *CalculoIdade*: recebe a data de nascimento cadastrada pelo aluno e retorna um valor do tipo “inteiro” de sua idade atual.
- *Geocodificador*: utiliza o endereço físico cadastrado pelo aluno para convertê-lo em coordenadas geográficas. Utiliza a API *Geocode* fornecida pelo *Google Maps*;
- *Notificador*: utiliza recursos de SMTP (*Simple Mail Transfer Protocol*) disponíveis em uma API de Java para enviar *e-mails* com diversos fins de notificação aos usuários;
- *XMLUpdater*: atualiza dados de um *carpool* quando este sofre alguma alteração que não implique em seu cancelamento. Por exemplo, quando um novo membro entra em um *carpool* já existente. A classe acessa a API *Directions* do *Google Maps* com os novos dados para obter os novos valores da distância e tempo total do trajeto e o novo acréscimo percentual que o trajeto do motorista sofre em relação ao trajeto direto.

Já a classe *Busca* desempenha um papel mais complexo, acessando a API *Directions* do *Google Maps*, usando dados relativos a itinerários de alunos como parâmetro (e utilizando também regras do sistema), para

determinar possíveis *matches*. A seção 4.2.3 detalha de forma mais minuciosa o funcionamento desta classe.

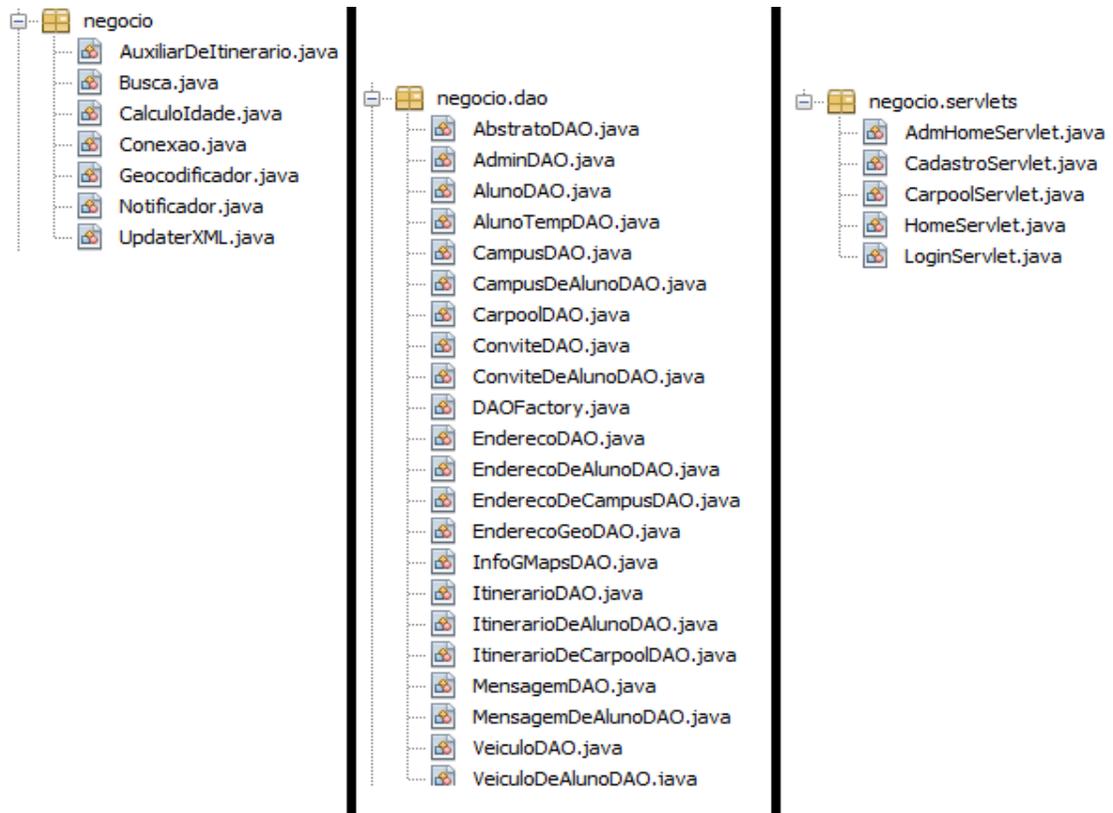


Figura 11. Pacote e Subpacotes das Classes de Negócio (*Controller*)

Fonte: autoria própria

A parte referente a *view* é composta por páginas JSP e HTML, sendo as primeiras para manipular entradas do usuário e exibir seus dados e as últimas para informar o *status* de algumas ações que um usuário executa em uma página JSP. Internamente à *view*, utiliza-se a tecnologia de *scriptlets* e *E.L.*, com a finalidade de “injetar” código Java na interface de usuário. A partir dos *scriptlets* e de *E.L.s* são instanciados objetos DAO e DTO para acessar e exibir dados das entidades do sistema. Como pode ser observado na Figura 12, as páginas estão divididas em diretórios dentro do contexto de execução do servidor, com exceção de três páginas que se encontram diretamente na raiz, duas referentes ao cadastro e a outra à página inicial

para *login*. Os diretórios estão distribuídos de acordo com a finalidade das páginas, ou seja, com o tipo de informação que as páginas exibem ou fornecem meios para se acessar.

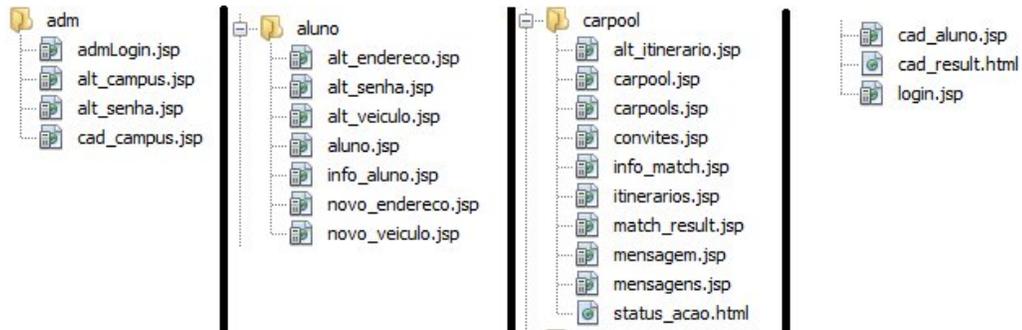


Figura 12. Páginas Web Organizadas em Diretórios (*View*)

Fonte: autoria própria

Além das páginas, o módulo da *view* utiliza dois recursos: uma imagem que aparece na página de *login* e um arquivo CSS para caracterização estética do *site*.

4.2.3 Mecanismo de busca e ordenação de pares

Para determinação de pares para compartilhamento de carona o programa utiliza uma “abordagem de cascata” para testar condições que possibilitem a formação de um *pool*. A análise de tais condições tem, em um primeiro momento, o objetivo de filtrar alunos com base em data (dia semanal) e hora (aproximada) do deslocamento e pontos de partida e chegada do itinerário em questão. Por meio da filtragem, as informações dos pares potencialmente adequados são armazenadas em uma *array* (faz-se uso de objetos da classe InfoGMaps, que carrega informações que relacionam percursos de dois pares além de uma referência para estes). Então os pares filtrados são ordenados de acordo com o percentual extra do percurso total (motorista buscando o passageiro) em relação ao percurso direto (que o motorista faria sem buscar o passageiro).

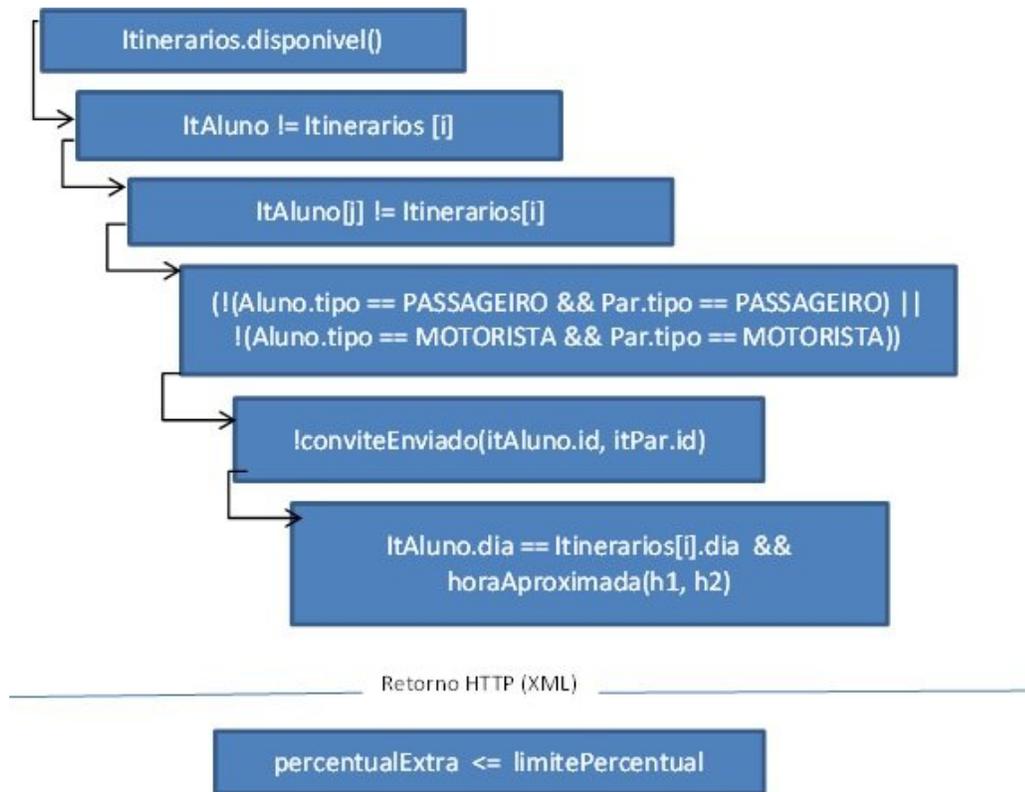


Figura 13. Processo de Busca

Fonte: autoria própria

Esclarecendo os índices da figura acima: i representa o índice de um elemento do conjunto do total de itinerários – $Itinerarios[]$, variando de 1 a n , onde n é o índice do último elemento no conjunto. Já j representa a variação de todos os itinerários relacionados ao aluno requisitante ($itAluno[]$) da busca a fim de garantir que não se relacionem dois itinerários de um mesmo aluno para um *carpool* (terceira condição da figura).

Podemos resumir as condições representadas na Figura 13 por extenso também. Colocando a expressão “então se” para testar uma condição mais interna dentro da cascata após a condição imediatamente externa ter sido validada, o resultado é:

- Para cada itinerário, se este estiver disponível para *carpool*;
- Então se o itinerário não é o mesmo do aluno requisitante;

- Então se o itinerário não pertence ao conjunto de itinerários do aluno;
- Então se requisitante e par não são do mesmo tipo⁸ comportamental;
- Então se os itinerários analisados não estão vinculados a um convite (o que indica que ainda não foi feito contato entre o usuários referente a um *pool* para os itinerários envolvidos);
- Os horários de saída ou chegada (depende do tipo de deslocamento) são próximos (o sistema considera próximo um intervalo de até 30 minutos);
- Então se o percentual de deslocamento do motorista em relação ao percurso direto é menor ou igual ao valor máximo que este cadastrou como aceitável.

Após este procedimento, o conjunto de dados está pronto para ser ordenado com base em hábitos relacionados ao fumo. A seguir quatro telas do sistema mostram etapas que abrangem desde a busca até o *carpool* formado.

Figura 14. Página de gerenciamento de Itinerários

Fonte: autoria própria

⁸ Podem ser do mesmo tipo quando este for do tipo “AMBOS”, ou seja, quando o aluno está disposto tanto a oferecer como a pegar caronas.

Em um primeiro momento o aluno deve criar um ou mais itinerários com os dados referentes ao seus endereços cadastrados, que serão carregados automaticamente. O aluno deve definir o dia semanal e a hora de seus deslocamentos.



Figura 15. Página de gerenciamento de Itinerários

Fonte: autoria própria

Esta página exibe informações dos itinerários do aluno, dando a ele a possibilidade de excluí-los ou buscar parceiros para compartilhar carona (botão “Carpool”).

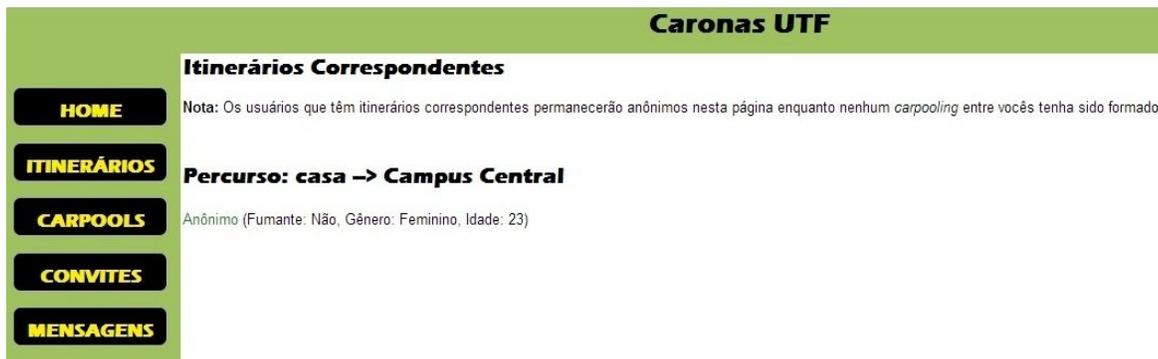


Figura 16. Lista de *matches* retornados em uma busca

Fonte: autoria própria

Para cada itinerário para o qual foram encontrados um ou mais *matches*, os resultados aparecem abaixo. Se os usuários (o aluno que fez a busca e o *match*) ainda não compartilham veículo para nenhum outro itinerário então o *match* aparece como “Anônimo” e algumas informações básicas sobre ele aparecem entre parêntesis. Do contrário, o nome de usuário é exibido.

The screenshot shows the 'Caronas UTF' interface. On the left is a navigation menu with buttons for HOME, ITINERÁRIOS, CARPOOLS, CONVITES, and MENSAGENS. The main content area displays match information for a route from Curitiba to Pinhais. It includes a legend (D = Destino; V = Você; P = Par; #s = Outros Membros), the destination 'Campus Central', a total distance of 6.4 km, and an estimated time of 14 minutes. A note states that if there is a carpool, the driver's route will increase by 34.70% compared to a direct route. A map shows the route with markers for 'V' (You) in Curitiba and 'P' (Partner) in Pinhais. Below the map are buttons for 'Enviar' (Send message) and 'Convidar' (Invite), along with a 'Retornar' (Return) button.

Figura 17. Informações do *match*

Fonte: autoria própria

Esta página exibe superficialmente os endereços dos alunos envolvidos no trajeto em potencial, e dá informações sobre este, como tempo estimado, distância e acréscimo na rota do motorista. O aluno decide se deve ou não convidar o *match* para partilhar um veículo para aquele trajeto, ou mandar uma mensagem antes para conhecer melhor seu potencial par.

Caronas UTF

HOME

ITINERÁRIOS

CARPOOLS

CONVITES

MENSAGENS

Parceiros de carpool:
fulano
fulana (M)

Veículo: ASA1030
[Gerar Mapa](#) [Abandonar](#)

Acrescimento de rota para o motorista: 34,70%

Bem vindo(a) fulano Logout

A Avenida Afonso Penna, 297 - Tarumá, Curitiba - PR, 82530-280, República Federativa do Brasil

5,5 km - cerca de 9 minutos

1. Siga na direção norte na Av. Afonso Penna em direção à R. Monte Castelo 0,3 km
2. Curva suave à direita na R. Raphael Papa 0,1 km
3. Vire à direita para pegar a Rod. Régis Bittencourt 2,5 km
4. Pegue a saída em direção a Av. Prof. Lothário Meissner 0,1 km
5. Mantenha-se à direita na bifurcação e pegue a Av. Prof. Lothário Meissner 1,2 km
6. Continue para Av. Prof. Omar Sabbag 0,8 km
7. Faça um retorno na R. Eng. Rebouças O destino estará à direita 0,5 km

B Avenida Prefeito Omar Sabbag, 894 - Jardim Botânico, Curitiba - PR, 80210-000, República Federativa do Brasil

3,3 km - cerca de 10 minutos

1. Siga na direção nordeste para a Av. Prof. Omar Sabbag 14 m
2. Vire à esquerda na Av. Prof. Omar Sabbag 0,6 km
3. Continue para Vd. do Cananema 0,6 km

Figura 18. Informações do *carpool*

Fonte: autoria própria

Após um convite ser aceito, é possível visualizar o trajeto com detalhes além de possibilitar a visualização dos dados do par (clicando no *link* do nome de usuário) omitidos antes de o *carpool* ser efetivado pelo sistema.

4.3 Síntese dos resultados

A análise do sistema foi fundamental para conceber uma estrutura que proporcionou a flexibilidade que o sistema possui. Por exemplo, por dar um caráter atômico à entidade *Itinerário*, que contém os atributos de origem e destino, é possível engajar um aluno em mais de um grupo em um mesmo dia. Esta etapa também ajudou a compreender a dimensão de um projeto voltado ao problema do arranjo de caronas, tanto em termos de código como

em termos de relações entre as entidades do mundo real para sua representação mais fidedigna possível na base de dados.

Na etapa de desenvolvimento foram implementados paradigmas de programação já há tempos consagrados, o que ajudou para agilizar esse processo. Os problemas mais complexos em termos de algoritmo limitaram-se à manipulação de dados de fontes externas, o cuidado na representação da interação entre alunos (por meio de convites e mensagens) e, principalmente, ao método de busca e ordenação de pares. O esforço de programação (incluindo o estudo de APIs externas) para criar o sistema demandou um tempo estimado entre 25 e 30 horas.

Com os resultados em mãos, encerra-se o trabalho com algumas considerações no capítulo seguinte.

5 CONCLUSÃO

Ao término do protótipo proposto neste trabalho, fica claro o potencial da tecnologia de geoprocessamento existente para redução de problemas urbanos como o abordado na pesquisa. O compartilhamento de caronas pode ser bem mais interessante e abrangente quando um sistema informatizado pode garantir os melhores arranjos possíveis e certa confidencialidade durante o contato entre as partes interessadas. O Caronas-UTF, se implantado, pode ser o início de uma cultura de transporte viável para os alunos da instituição UTFPR. O conceito adotado, prezando pela flexibilidade, faz com que seja uma alternativa para os alunos interessados, mesmo que estes não o utilizem com muita frequência.

Este trabalho deixa como legado um sistema para compartilhamento de caronas específico a um público bem restrito – alunos da UTFPR-CT. O sistema se mostrou razoavelmente complexo em termos de abstração e implementação mas, plenamente viável. Nesta seção se discorre sobre sugestões para trabalhos futuros que possam se utilizar deste como base. Antes, porém, vale mencionar duas pendências: um servidor *web* e um recurso de criptografia para senhas de aluno. Obviamente, para estar disponível ao público, um servidor *web* é essencial, e, por questões de segurança, senhas de acesso devem ser criptografadas antes de armazenadas em BD. Trabalhos futuros devem implementar estes recursos caso precisem estar disponíveis ao público.

Seguem as sugestões:

- Unificação de *login* com o sistema acadêmico: a Universidade Tecnológica possui um sistema informatizado pelo qual alunos tratam de questões relacionadas às suas matrículas. Cada aluno possui um código único e uma senha para acessar tal sistema. Na hipótese de o Caronas-UTF utilizar esses dados, seus usuários não

precisariam de nenhuma outra informação adicional para comprovar que são alunos da UTFPR. Tal medida reforçaria a segurança e privacidade do sistema e não é tão complexa, visto que a única coisa que o Caronas-UTF teria que fazer é importar uma lista com os alunos matriculados nos *campi* da cidade de Curitiba uma vez em cada semestre (excluindo os egressos e incluindo os novos alunos ingressos). Naturalmente, essa decisão foge do domínio do autor desse trabalho, sendo que necessita da autorização da instituição;

- Disponibilidade para dispositivos móveis: tendo em vista o notável aumento do uso de dispositivos móveis para acessar a *Internet*, é interessante que a estrutura forneça uma plataforma de acesso para usuários que utilizam esta tecnologia com frequência. Em adição à estrutura já implementada, seria necessária uma nova interface de usuário apropriada a este tipo de dispositivo. Além disso seriam necessárias alterações na camada de negócio (*controller*) do sistema, já que classes *servlets* são específicas para aplicações *web*.
- Mecanismo de cálculo de custo de um trajeto: se for possível determinar o consumo de combustível (litros por quilômetro) do veículo de um *carpool*, calcular o custo de um deslocamento (em reais, por sua vez), e disponibilizar aos integrantes do grupo, serviria de auxílio na divisão de custos entre estes.
- Agregação de outro estilo de *Carpool*: ainda que o *carpool* casual não seja tão indicado para o compartilhamento de caronas na cidade de Curitiba, poderia ser usado ao menos em trajetos inter-*campi* já que os veículos já estão em vias públicas ou estacionamentos, cabendo apenas um sistema de cadastramento para que os usuários possam se identificar inicialmente.
- Inclusão de outras categorias de usuários: para fins de simplificação, o protótipo levou em conta apenas alunos como

usuários-alvo. No entanto, é possível que o sistema abarque outras categorias de usuário, como professores e servidores.

- Otimização de rotas: atualmente as rotas são calculadas estabelecendo a origem e destino do motorista como origem e destino do trajeto propriamente, enquanto os endereços dos passageiros são vistos como pontos de parada. Especificamente o deslocamento do tipo *casa-campus* possibilita que os passageiros possam deslocar-se a fim de parar em um ponto mais adequado à rota do motorista, evitando acréscimos ao seu trajeto. Para implementar tal otimização, uma ideia seria permitir ao usuário passageiro definir o limite de deslocamento a pé – em quadras ou em metros. A vantagem de definir este limite baseado em quadras é a maior facilidade de compreensão por parte do usuário, mas, de qualquer forma, este valor teria que ser convertido em metros por meio de alguma regra de conversão produzida pelo programador, já que a API do Google *Maps* pode lidar com valores nesta unidade de medida. De posse deste dado, o algoritmo deve ser capaz de estabelecer a direção e o ponto do trajeto ao qual o passageiro precisaria se deslocar.

Ao longo deste estudo foi possível constatar tanto vantagens como desvantagens no que se refere à prática de *carpooling*. Um sistema de informação voltado a esse problema deve, além de se utilizar de recursos que salientem as vantagens, amenizar as desvantagens. Por exemplo, a divisão do custo racionalizado ao deslocamento pode compensar a comodidade da utilização de veículos com um único ocupante. Sistemas restritos a instituições podem amenizar o receio de dividir um veículo com um desconhecido.

As tecnologias de geoprocessamento se mostraram muito úteis na solução do problema do compartilhamento de carona, cabendo às intuições fazer uso destas da forma mais adequada a cada contexto.

6 REFERÊNCIAS

BAKSHI, Rahul; KNOBLOCK, Craig A.; THAKKAR, Snehal. Exploiting online sources to accurately geocode addresses. In: GIS '04 Annual ACM International Workshop on Geographic Information Systems, 12., New York, *Proceedings...* p. 194-203. New York: ACM, 2004.

BRIEN, Lynn F. ***Privacy and geospatial technologies***. Submitted to the Graduate Faculty of the University of New Orleans in partial fulfillment of the requirements for the degree of Master of Arts In Geography. Estados Unidos, 2009.

CALVO, Roberto W.; LUIGI, Fabio de; HAASTRUPC, Palle; MANIEZZO, Vittorio. A distributed geographic information system for the daily car pooling problem. ***Computers & Operations Research***, v. 31, n. 13, p. 2263–2278, November, 2004.

CARPOOLGLOBAL.COM. Fighting global warming since 2006. s.d. Disponível em: <<http://www.carpoolglobal.com/>>. Último acesso em junho de 2013.

DE OLIVEIRA, Robert A. Nogueira. **Web GIS OL4JSF**. Revista FOSSGIS Brasil, página 33. Edição nº 1, março de 2011. Brasil, 2011.

DEVELOPERS.GOOGLE.COM. The Google Geocoding API. s.d. Disponível em: <<https://developers.google.com/maps/documentation/geocoding/>>. Último acesso em maio de 2013.

DEWAN, Kum K; AHMAD, Israr. Carpooling: a step to reduce congestion (a case of study of Delhi). ***Engineering Letters***, v. 14, n. 1, 2007.

ELDRANDALY, K. GIS software selection: a multi-criteria decision making approach. ***Applied GIS***, v. 3, 2007.

FUKUDA , Tuenjai; KASHIMA, Shigeru; FUKUDA, Atsushi; NARUPITI, Sorawit. Analysis of car sharing application on consumer orientation and their modal selection in Bangkok. *Journal of the Eastern Asia Society for Transportation Studies*, v. 6, 2005.

IPEA - Instituto de Pesquisa Econômica Aplicada. **O custo do caos**, 2011.

IPEA - Instituto de Pesquisa Econômica Aplicada. **Proposta de diretrizes de política de transporte urbano**, ????.

KOTHARI, Amit B. Genghis - a multiagent carpooling system. B.Sc. Dissertation work, submitted to the University of Bath. May 11, 2004.

MINETT, Paul; PIERCE, Jonh. Estimating the Energy Consumption Impact of Casual Carpooling. *Energies*, v. 4, n. 1, p. 126-139, 2011. doi:10.3390/en4010126.

NEBERT, Douglas D. **Global spatial data infrastructure**. The SDI Cookbook. GSDI. 2004.

PREÇO DOS COMBUSTÍVEIS. Preço dos combustíveis. s.d. Disponível em: <<http://precodoscombustiveis.com.br>>. Último acesso em maio de 2013.

RIDENOW.ORG. Ride now! s.d. Disponível em: <<http://www.ridenow.org>> Último acesso em abril de 2013.

SCHIMITT, Rafael da Silva. **Impactos da implantação de medidas de gerenciamento da mobilidade em uma área urbana com múltiplos pólos atratores de viagem**. Dissertação de mestrado. Programa de Pós-graduação em Engenharia de Produção. Universidade Federal do Rio Grande do Sul, 2006.

SGHAIER, Manel; ZGAYA, Hayfa; HAMMADI, Slim; TAHON, Christian. A distributed optimized approach based on the multi agent concept for the

implementation of a real time carpooling service with an optimization aspect on siblings. ***International Journal of Engineering***, v. 5, n. 2, p. 217, 2011.

SOLTYS, Kalina A. **Toward an understanding of carpool formation and use.** *A thesis submitted in conformity with the requirements for the degree of Master of Arts, Graduate Department of Geography and Planning, University of Toronto.* Canadá, 2009.

WIKI.GIS.COM. The GIS encyclopedia. s.d. Disponível em: <<http://wiki.gis.com>>. Último acesso em março de 2013.