

**UNIVERSIDADE DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL**

**METODOLOGIA DE TRATAMENTO DE DADOS PARA**  
**SIMULAÇÃO DE MODELO BASEADO EM ATIVIDADES**  
**USANDO O *SOFTWARE* MATSIM**

**DANIELE FIRME MIRANDA**

**ORIENTADORA: Fabiana Serra de Arruda**

**MONOGRAFIA DE PROJETO FINAL EM TRANSPORTES**

**BRASÍLIA / DF: JUNHO-2017**

**UNIVERSIDADE DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL**

**METODOLOGIA DE TRATAMENTO DE DADOS PARA  
SIMULAÇÃO DE MODELO BASEADO EM ATIVIDADES  
USANDO O *SOFTWARE* MATSIM**

**DANIELE FIRME MIRANDA**

MONOGRAFIA DE PROJETO FINAL SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM ENGENHARIA CIVIL.

**APROVADA POR:**

---

**Prof<sup>a</sup>. FABIANA SERRA DE ARRUDA, Doutora (ENC-UnB)**  
**(Orientadora)**

---

**Prof<sup>a</sup>. MICHELLE ANDRADE, Doutora (ENC-UnB)**  
**(Examinadora Interna)**

---

**Prof. PASTOR WILLY GONZALES TACO, Doutor (ENC-UnB)**  
**(Examinador Interno)**

**BRASÍLIA/DF, 23 DE JUNHO DE 2017.**

## FICHA CATALOGRÁFICA

MIRANDA, DANIELE FIRME

Metodologia de tratamento de dados para simulação de modelo baseado em atividades usando o *software* MATSim. [Distrito Federal] 2017.

xii, 135p., 210x297mm (ENC/FT/UnB, Bacharel, Engenharia Civil, 2017)

Monografia de Projeto Final – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Civil e Ambiental

1. Planejamento de transportes

2. Modelagem de transportes

3. Tratamento de dados

4. Modelos de atividades

I. ENC/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

MIRANDA, D. F. (2017). Metodologia de tratamento de dados para simulação de modelo baseado em atividades usando o *software* MATSim. Monografia de Projeto Final, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 135p.

## CESSÃO DE DIREITOS

NOME DA AUTORA: Daniele Firme Miranda

TÍTULO DA MONOGRAFIA DE PROJETO FINAL: Metodologia de tratamento de dados para simulação de modelo baseado em atividades usando o *software* MATSim.

GRAU / ANO: Bacharel em Engenharia Civil / 2017

É concedida à Universidade de Brasília a permissão para reproduzir cópias desta monografia de Projeto Final e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. A autora reserva outros direitos de publicação e nenhuma parte desta monografia de Projeto Final pode ser reproduzida sem a autorização por escrito da autora.

---

Daniele Firme Miranda  
danielefirme@gmail.com

## **AGRADECIMENTOS**

À professora e orientadora Fabiana, por ter me incentivado a escolher a Engenharia de Transportes como área de pesquisa e por ter me proporcionado um grande crescimento como pesquisadora nesse ano de trabalho.

Aos professores Pastor e Michelle, pelas contribuições e recomendações feitas durante a primeira banca de projeto final, que muito me auxiliaram no desenvolvimento do trabalho a partir dali.

Aos demais professores do PPGT, incluindo a prof<sup>a</sup>. Yaeko e o prof. Paulo César, com os quais tive contato durante a graduação, e cujo conhecimento transmitido teve fundamental importância na minha escolha da área de transportes como campo de pesquisa.

Ao querido amigo Samuel, não só pelos conceitos de programação que me ensinou, mas também por ser meu companheiro de todas horas, e pelo apoio inclusive nos momentos mais difíceis da graduação.

Aos meus pais, Élvis e Veneza, e ao meu irmãozinho Júnior, por tudo.



## RESUMO

Modelos são representações simplificadas do mundo real, muitas vezes obtidos a partir de relações matemáticas entre variáveis selecionadas, e são ferramentas importantes para o planejamento de transportes. Existem modelos de base teórica simples, agregados, como o Modelo Quatro Etapas, e também mais complexas, como a modelagem desagregada, baseada no indivíduo. Modelos desagregados, apesar de produzirem resultados mais precisos, geralmente demandam maiores custos financeiros tanto na aquisição de *softwares* e recursos computacionais quanto na obtenção de dados, o que representa um gargalo para o planejamento em países em desenvolvimento, como o Brasil.

Nesse contexto, surge o questionamento: como tornar possível o uso da modelagem desagregada em situações em que há limitações técnicas e financeiras ao planejamento? O presente trabalho objetiva, portanto, desenvolver uma metodologia de tratamento de dados para simulação de um modelo baseado em atividades (desagregado) usando o *software* MATSim, que é gratuito e com código livre. Para aplicação da metodologia desenvolvida, foi feito um estudo de caso da região do Distrito Federal, e os dados tratados incluem a rede viária e uma população sintética.

Primeiramente, os dados geográficos da área de estudo foram obtidos da base de dados gratuita *OpenStreet Map*. As informações incluíam a rede viária do Distrito Federal e os pontos de realização de atividades. Esses dados foram tratados utilizando rotinas em linguagem Python e posteriormente convertidos para o formato de entrada no MATSim. Em seguida, a população sintética foi gerada com base em padrões de atividades estabelecidos de acordo com dados censitários da região, utilizando uma rotina em Java.

O modelo foi verificado fazendo-se uma comparação com contagens volumétricas em alguns trechos viários da área de estudo, em que se verificou um subdimensionamento do cenário gerado, mas os padrões de fluxo encontrados foram razoavelmente coerentes.

## ABSTRACT

Models are simplified representations of the real world, often obtained from mathematical relations among selected variables, and they are important tools to transport planning. There are aggregate models, based on simple theories, for instance, the Four Step Model, and there are also more complex models, such as the disaggregate theories, that focus on each individual. Disaggregate models, despite of producing more accurate results, often demand higher financial costs, because of their expensive softwares and data collection. This is an issue for transport planning in Brazil.

Given the background, there is the question: how can one make disaggregate transport modelling feasible in situations where is lack of finance and technical resources? Thus, the aim of this research is to develop a methodology for data treatment, concerning input information on a disaggregated model, using the free and open-source software MATSim. To test the methodology developed, a study case was conducted within Distrito Federal region, and the treated data included its road network and a synthetic population.

First, geographic data of the studied area was obtained from the open-source database OpenStreet Map. Information included Distrito Federal's road network and points of interest where activities were done. These data were treated using Python scripts and then they were converted into MATSim input format. Finally, the synthetic population was generated based on activity patterns created according Census data, using a Java script.

The model was verified by comparing it with volumetric counts performed in some roads within the study area. The scenario generated was undersized, but the flow patterns observed were reasonably consistent.

## SUMÁRIO

AGRADECIMENTOS .....	iv
RESUMO .....	v
ABSTRACT .....	vi
SUMÁRIO .....	vii
LISTA DE TABELAS .....	x
LISTA DE FIGURAS .....	xi
1. INTRODUÇÃO .....	1
1.1 Apresentação do Tema .....	1
1.2 Pergunta de pesquisa .....	3
1.3 Objetivos do Trabalho .....	3
1.4 Justificativa .....	3
1.5 Estrutura do Trabalho .....	4
2. REVISÃO DA LITERATURA .....	5
2.1 Introdução .....	5
2.2 Metodologia Sistemática de Revisão .....	5
2.3 Modelagem no Planejamento de Transportes .....	5
2.3.1 Generalidades .....	5
2.3.2 Evolução histórica dos modelos de transporte .....	6
2.3.3 Características e limitações da modelagem para o planejamento .....	8
2.4 Modelagem de sistemas multiagente .....	10
2.4.1 Generalidades .....	10
2.4.2 Exemplos de aplicação da abordagem multiagente .....	11
2.5 Modelagem baseada em atividades .....	12
2.5.1 Generalidades .....	12
2.5.2 Entrada e saída de um modelo baseado em atividades .....	14
2.6 Tratamento de dados para modelagem .....	15
2.7 MATSim: Multi-Agent Transport Simulation .....	16
2.7.1 Histórico do MATSim .....	16
2.7.2 Características básicas do MATSim .....	17
2.7.3 Estudos anteriores usando MATSim .....	18

2.7.4	Ciclo iterativo do MATSim .....	19
2.7.5	Atribuição de Scores .....	21
2.7.6	Modelagem do fluxo de tráfego .....	23
2.7.7	Arquivos de entrada no MATSim .....	24
3.	MÉTODO .....	30
3.1	Introdução .....	30
3.2	Geração da rede viária ( <i>network.xml</i> ) .....	30
3.2.1	<i>Softwares</i> e arquivos necessários .....	30
3.2.2	Método para geração da rede viária .....	31
3.3	População sintética ( <i>population.xml</i> ) .....	41
3.3.1	<i>Softwares</i> e arquivos necessários .....	41
3.3.2	Método para geração de população sintética .....	42
3.4	Arquivo de configuração ( <i>config.xml</i> ) .....	52
3.4.1	<i>Softwares</i> e arquivos necessários .....	52
3.4.2	Método desenvolvido para criação do arquivo de configuração .....	54
4.	RESULTADOS .....	56
4.1	Introdução .....	56
4.2	Utilizando o MATSim 0.8.1 .....	56
4.3	Produtos da simulação .....	57
4.4	Visualização dos eventos .....	59
4.5	Análise dos resultados .....	60
4.5.1	Ponto 1: Interseção da DF-085 (EPTG) e DF-001 (EPCT) com a Avenida Central de Taguatinga .....	61
4.5.2	Ponto 2: Interseção da DF-075 (EPNB) com a DF-003 (EPIA) .....	62
4.5.3	Ponto 3: Interseção da DF-095 (EPCL) com a DF-003 (EPIA), Viaduto Ayrton Senna 64	
4.5.4	Ponto 4: BR-010/020/030 no Balão do Colorado/Via de acesso a Sobradinho II 65	
5.	CONCLUSÃO E DISCUSSÕES .....	67
5.1	Conclusão do estudo .....	67
5.2	Limitações encontradas na pesquisa .....	68
5.3	Contribuições do estudo .....	68
5.4	Sugestões para trabalhos futuros .....	69
	BIBLIOGRAFIA .....	70

APÊNDICE A - Revisão Sistemática de Literatura.....	74
APÊNDICE B - Estudos Anteriores com MATSim.....	76
APÊNDICE C - Detalhamento das variáveis envolvidas no cálculo de <i>scores</i> do MATSim	79
APÊNDICE D - Especificações da máquina utilizada para executar os programas.....	82
APÊNDICE E - Rotina para organização do banco de dados geográfico.....	83
APÊNDICE F - Rotina para conversão de arquivo Shapefile das vias em dado de entrada no MATSim	87
APÊNDICE G - Categorias de pontos do <i>OpenStreet Map</i> .....	91
APÊNDICE H - Rotina para organização do banco de dados de pontos.....	104
APÊNDICE I - Rotina para conversão do arquivo Shapefile dos pontos em dado de entrada no MATSim	108
APÊNDICE J - Rotina para geração do arquivo <i>population.xml</i> .....	110
APÊNDICE K - Arquivo <i>config.xml</i> utilizado na simulação .....	124
ANEXO A - Classificação das rodovias pela base de dados <i>OpenStreet Map</i> .....	135

## LISTA DE TABELAS

Tabela 2.1: Exemplos de aplicação de simulação baseada em agentes .....	12
Tabela 3.1: Correspondência entre classificação de vias na base <i>OpenStreet Map</i> e no Código Brasileiro de Trânsito (Brasil, 1997) .....	36
Tabela 3.2: Cálculo da capacidade de acordo com a classificação das vias .....	41
Tabela 3.3: Grupos de indivíduos e padrões de atividades .....	47
Tabelas 3.4 e 3.5: Distribuição dos padrões de atividades .....	49
Tabela 3.6: Total de agentes simulados no estudo de caso .....	53
Tabela 4.1: Contagens e erros relativos no Ponto 1 .....	61
Tabela 4.2: Contagens e erros relativos no Ponto 2 .....	63
Tabela 4.3: Contagens e erros relativos no Ponto 3 .....	64
Tabela 4.4: Contagens e erros relativos no Ponto 4 .....	66

## LISTA DE FIGURAS

Figura 2.1: Evolução histórica dos modelos de transporte. ....	8
Figura 2.2: Motivos pelos quais os planejadores em transporte e uso do solo deixam de usar a modelagem em suas análises (Traduzido de Brömmelstroet & Bertolini, 2011). ....	10
Figura 2.3: À esquerda, arquitetura do sistema de agente único (centralizado); À direita, arquitetura do sistema multiagente (Traduzido de Stone, 1997). ....	11
Figura 2.4: Realismo comportamental vs. Complexidade computacional de modelos de transporte.....	13
Figura 2.5: Modelagem de dados de entrada para simulação (Chwif & Medina, 2014). ....	15
Figura 2.6: Arquitetura do TRANSIMS (Traduzida de Smith <i>et al.</i> , 1995). ....	17
Figura 2.7: Processo de simulação do MATSim. ....	19
Figura 2.8: Teoria das filas (Horni et al., 2016).....	23
<b>Figura 2.9:</b> Estrutura de um arquivo de texto XML. ....	24
Figura 2.10: Árvore de dados do arquivo <i>config.xml</i> .....	25
Figura 2.11: Exemplo de arquivo de texto <i>config.xml</i> .....	26
Figura 2.12: Árvore de dados do arquivo <i>network.xml</i> .....	27
Figura 2.13: Exemplo de arquivo de texto <i>network.xml</i> .....	27
Figura 2.14: Árvore de dados do arquivo <i>plans.xml</i> .....	29
Figura 2.15: Exemplo de arquivo de texto <i>network.xml</i> .....	29
Figura 3.1: Esquema da metodologia proposta para geração e tratamento de dados geográficos.....	31
Figura 3.2: Arquivo importado pelo <i>software</i> Quantum GIS da base <i>Open Street Map</i> . ....	33
Figura 3.3: Atributos dos elementos viários na camada “ <i>lines</i> ”.....	34
Figura 3.4: Elemento do tipo “ <i>primary_link</i> ” formado por 4 segmentos de linha. ....	38
Figura 3.5: Segmentos de linha individualizados após procedimento de conversão em grafos. ....	38
Figura 3.6: Modelo de Greenberg para relação entre fluxo de tráfego ( $q$ ) e concentração de veículos ( $k$ ) (Silva, 2007).....	40
Figura 3.7: Esquema da metodologia proposta para geração e tratamento de dados de população sintética.....	42
Figura 3.8: Atributos dos elementos pontuais na camada “ <i>lines</i> ”.....	43
Figura 3.9: Hierarquia das categorias de atividades para cada ponto. ....	45
Figura 3.10: Banco de dados tratado com os elementos pontuais. ....	46

Figura 3.11: Regiões do Entorno do DF incluídas na área de estudo .....	50
Figura 4.1: Interface do MATSim 0.8.1; em destaque, campo para especificação do endereço do arquivo de configuração.....	56
Figura 4.2: Tempo de execução dos processos em cada iteração .....	57
Figura 4.3: Evolução dos <i>scores</i> dos planos ao longo das iterações.....	58
Figura 4.4: Trecho do arquivo <i>events.xml</i> gerado como produto da simulação do <i>MATSim</i> ..	59
Figura 4.5: Visualização dos eventos de simulação no pico da manhã (07h30).....	59
Figura 4.6: Visualização dos eventos de simulação no pico da noite (18h30) .....	60
Figura 4.7: Pontos de contagem volumétrica avaliados.....	60
Figura 4.8: Movimentos analisados no Ponto 1 .....	61
Figura 4.9: Comparação entre o volume de tráfego simulado para cada hora e o volume das contagens de tráfego, para os movimentos A (esquerda) e B (direita), no Ponto 1 .....	62
Figura 4.10: Movimentos analisados no Ponto 2.....	62
Figura 4.11: Comparação entre o volume de tráfego simulado para cada hora e o volume das contagens de tráfego, para os movimentos A (esquerda) e B (direita), no Ponto 2.....	63
Figura 4.12: Movimentos analisados no Ponto 3 .....	64
Figura 4.13: Comparação entre o volume de tráfego simulado para cada hora e o volume das contagens de tráfego, para os movimentos A (esquerda) e B (direita), no Ponto 3.....	65
Figura 4.14: Movimentos analisados no Ponto 4.....	65
Figura 4.15: Comparação entre o volume de tráfego simulado para cada hora e o volume das contagens de tráfego, para os movimentos A (esquerda) e B (direita), no Ponto 4.....	66



# 1. INTRODUÇÃO

## 1.1 Apresentação do Tema

Há mais de uma década, Garber & Hoel (2001) definiram planejamento de transportes como sendo o levantamento racional e não-enviesado dos efeitos de determinado projeto de transportes em uma comunidade. Em uma concepção mais ampla e atual, Campos (2013) conceitua o planejamento de transportes como sendo a definição de estratégias para compatibilizar a oferta de transporte com a demanda presente ou a prevista para o futuro.

O planejamento é essencial porque é no presente que escolhas importantes são tomadas para se garantir futuros sustentáveis (Brömmelstroet & Bertolini, 2011). Nos últimos anos, correlações complexas entre problemas e fatos puderam ser melhor compreendidas, e a sinergia entre grandes áreas, como comportamento da população e políticas de transporte, por exemplo, pôde ser verificada. Sem o devido planejamento, a análise *a posteriori* das decisões tomadas pode se mostrar insuficiente para permitir a eficácia dos projetos.

Em consonância com necessidade de valorização do planejamento de transportes para o desenvolvimento adequado das cidades, foi instituída no Brasil a Política Nacional de Mobilidade Urbana – PNMU, na figura da lei 12.587/2012 (Ministério das Cidades, 2015). A lei reconhece a importância da mobilidade como fator de influência direta na qualidade de vida de uma comunidade. Em um de seus artigos, a lei determina que os municípios com mais de 20 mil habitantes, obrigatoriamente, devem elaborar seus Planos de Mobilidade Urbana, sob pena de perderem acesso aos recursos federais para investimentos no setor.

A elaboração de um Plano de Mobilidade Urbana requer a análise dos meios de deslocamento que têm impacto na circulação dentro de um município. Entre as sugestões dadas pelo Caderno de Referência para Elaboração do Plano de Mobilidade Urbana (Ministério das Cidades, 2015), está a elaboração de modelos para análise de projeções e alternativas de investimentos.

Modelos, que são representações simplificadas do mundo real (Ortúzar & Willumsen, 2004), permitem a visualização virtual dos efeitos que mudanças causam em cenários reais. Quanto mais forte o embasamento teórico de um modelo, e mais confiáveis e detalhados seus dados de entrada, maior a verossimilhança da simulação obtida.

A teoria clássica que embasa modelos de transportes há décadas é o chamado Modelo Sequencial, ou Modelo de Quatro Etapas, de baixa complexidade matemática. Com o passar dos anos, o desenvolvimento de ferramentas tecnológicas viabilizou que novas teorias dessem origem a modelos mais complexos e representativos com relação aos fatos reais. Uma dessas teorias é baseada na abordagem multiagente, que considera as interações dos elementos estudados (motoristas, pedestres ou até mesmo semáforos) entre si e com o ambiente ao seu redor (Wooldridge, 1992).

Um tipo de abordagem multiagente que tem apresentado muitas vantagens na modelagem de transportes é a modelagem baseada em atividades (Macal & North, 2010). Esse tipo de modelo tem como princípio o fato de que o transporte não é uma demanda em si, e sim uma demanda derivada da necessidade de se realizar uma atividade. Os agentes, portanto, são os indivíduos que precisam do transporte para realizar suas rotinas diárias, e as principais variáveis de entrada desse tipo de modelo são as atividades realizadas pelas pessoas durante um dia, os horários dessas atividades e os modos de transporte escolhidos para realizar viagens entre atividades (Ettema, 1996).

Apesar da gama de vantagens oferecida pela modelagem de transportes ao planejamento, são diversas as limitações encontradas pelos próprios planejadores na sua utilização. Brömmelstroet & Bertolini (2011), por exemplo, citam que entre os motivos que desestimulam os tomadores de decisão no uso das ferramentas de modelagem estão a dificuldade no uso do *software* e o custo desses programas. No Brasil, mesmo com a exigência por lei dos Planos de Mobilidade para municípios com mais de 20 mil habitantes, as dificuldades encontradas pelos planejadores também foram a falta de profissionais capacitados para elaboração dos planos e a falta de recursos financeiros (Associação Nacional das Empresas de Transportes Urbanos, 2015). Assim, levanta-se a discussão sobre a restrição dos municípios brasileiros em utilizar modelagem e simulação como ferramentas do planejamento de transporte.

## 1.2 Pergunta de pesquisa

Como tornar possível o uso da modelagem desagregada (indivíduo) em situações em que há limitações técnicas e financeiras ao planejamento?

## 1.3 Objetivos do Trabalho

O objetivo principal deste trabalho é desenvolver uma metodologia de tratamento de dados para simulação de um modelo baseado em atividades (desagregado) usando o *software* MATSim.

Como objetivos específicos da pesquisa, é possível citar:

- Elaboração de rotina para criação de bancos de dados geográficos e vetores que auxiliem a modelagem multiagente;
- Elaboração de método simplificado para geração de população sintética e planos de atividades, baseado em dados censitários da região de estudo;
- Estudo de caso com a simulação de um modelo de demanda por transporte, baseado em atividades, para a região do Distrito Federal.

## 1.4 Justificativa

Apesar da Lei 12.587/2012 ter exigido que, em até 3 anos, todos os municípios com mais de 20 mil habitantes tivessem elaborado seus Planos de Mobilidade Urbana, o cenário observado em 2015 foi muito diferente do esperado. À época, segundo consulta feita pelo Ministério das Cidades, 95% dos municípios com mais de 50 mil habitantes não tinham concluído seus Planos de Mobilidade, sendo que, destes, apenas 29% estavam em processo de elaboração (Associação Nacional das Empresas de Transportes Urbanos, 2015). Os principais problemas identificados pelos gestores municipais eram a falta de recursos financeiros e de pessoal qualificado para a elaboração dos planos.

Diante desse cenário, percebe-se a necessidade de elaboração de uma metodologia que ofereça, de forma acessível e simplificada, ferramentas que auxiliem o planejamento de transportes nos municípios. É nesse contexto que se destaca o *software* de simulação MATSim: um programa gratuito de código livre (*open-source*); versátil, pois simula modos

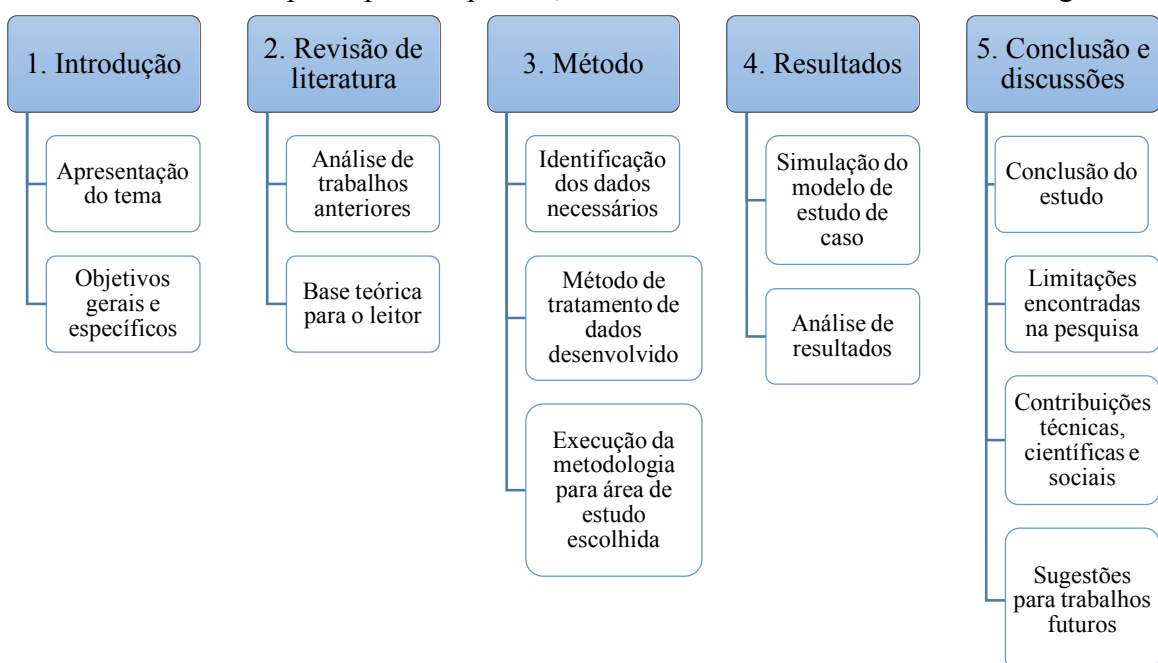
rodoviário, ciclovário ou de transporte público e; bem avaliado no meio acadêmico em comparação aos *softwares* mais populares de simulação de demanda (Farinha, 2013; Bicudo, 2015; Medrano, 2016).

Além disso, o MATSim utiliza a abordagem baseada em atividades, que nas últimas décadas foi reconhecida como uma das abordagens mais completas para o planejamento de transporte (Ettema, 1996; Spear, 1996; Shiftan & Ben-Akiva, 2011). Diferentemente da modelagem clássica, que apenas reproduz condições existentes no momento da pesquisa, os modelos baseados em atividades têm caráter comportamental e são capazes de fornecer informações mais fidedignas sobre potenciais efeitos de políticas de transporte no comportamento de viagem (Arruda, 2005).

A modelagem baseada em atividades exige coleta e preparação de dados mais cuidadosas que a metodologia clássica (Bhat & Koppelman, 2003). Sua validação costuma se dar pela aplicação de questionários de diários de atividades, que são mais longos que os questionários de viagens, o que pode dificultar o uso do modelo. Assim, surge a necessidade da simplificação da etapa de obtenção e tratamento de dados para modelagem, de forma a incentivar o uso dessa ferramenta para o planejamento.

## 1.5 Estrutura do Trabalho

Este trabalho será composto por 5 capítulos, abordando os elementos descritos a seguir:



## **2. REVISÃO DA LITERATURA**

### **2.1 Introdução**

Este capítulo tem o propósito de fornecer um entendimento geral dos assuntos abordados por esse projeto, que são: modelagem no planejamento de transportes, modelagem multiagente, abordagem baseada em atividades, metodologias de tratamento de dados e o MATSim (Multi-Agent Transport Simulation). Cada tema é esquematicamente discutido nos tópicos desse capítulo. Além disso, é feita a descrição da metodologia de pesquisa utilizada, que foi a revisão sistemática de literatura.

### **2.2 Metodologia Sistemática de Revisão**

Para a execução da revisão de literatura, foi utilizada a metodologia de análise sistemática, cujo princípio é desenvolvido por Treinta et al. (2014). Primeiramente, foram listadas palavras-chave a partir dos objetivos da pesquisa. Esses termos foram, então, inseridos em diferentes bases de dados, conforme detalhado no Apêndice A, e um catálogo de obras relevantes (com base em número de citações) e foi levantado. Na etapa seguinte, foram lidos os resumos dos textos a fim de se verificar a importância do documento para a pesquisa. Os documentos mais relevantes foram lidos na íntegra e fizeram parte desta revisão de literatura.

### **2.3 Modelagem no Planejamento de Transportes**

#### **2.3.1 Generalidades**

Um modelo é definido por Ortúzar & Willumsen (2004) como sendo uma representação simplificada de parte do mundo real, que concentra elementos considerados importantes para sua análise a partir de um certo ponto de vista. A filosofia básica por trás da operação de um modelo é a adoção de hipóteses que ofereçam a explicação do fenômeno investigado, e a demonstração de que essas hipóteses permitem a reprodução de uma base de dados satisfatória (Timmermans & Arentze, 2011). Isso significa que um modelo, para gerar bons resultados, deve ter um embasamento teórico forte, com dados de entrada confiáveis e uma calibração correta.

É importante diferenciar corretamente os conceitos de modelo e simulação. Horni et al. (2016) explicitam essa diferença ao descrever modelo como a caracterização dos aspectos de um sistema, ao passo que a simulação é a aplicação de um modelo. Por exemplo, um modelo de transporte de escolha modal pode dizer que uma rota “A” tem 25% de probabilidade de ser escolhida, enquanto a rota “B” tem 75%. Uma simulação desse modelo pode permitir a visualização prática dessa distribuição, e como o ambiente como um todo é caracterizado por essas escolhas.

### **2.3.2 Evolução histórica dos modelos de transporte**

A necessidade por modelos de demanda começou a ser reconhecida pela maioria dos planejadores e pesquisadores por volta de década de 1950, com a modelagem macroeconômica de fluxos de pessoas e bens (Sivakumar, 2007). Antes disso, o planejamento de transportes era meramente baseado em contagens de tráfego. Em 1955, com o estudo de caso na cidade de Chicago, nos Estados Unidos, foi criada a primeira estrutura conceitual da modelagem de transporte, que é o Modelo Sequencial, ou de Quatro Etapas (Hensher & Button, 2008).

O Modelo Clássico de Quatro Etapas é um modelo de previsão de demanda assim denominado por ser composto de 4 partes: geração de viagens, distribuição de viagens, repartição modal e alocação de viagens (Maerivoet & De Moor, 2008). Esse é um modelo que recebe dados agregados em zonas de tráfego, dentro das quais os indivíduos possuem características semelhantes. As teorias que embasam esse tipo de modelo são predominantemente físicas, como por exemplo a teoria gravitacional. Durante muitas décadas, o modelo clássico foi aprimorado, principalmente com os avanços computacionais que ocorreram. Entretanto, ao mesmo tempo, novas abordagens mais complexas e com maior verossimilhança foram desenvolvidas (Ortúzar & Willumsen, 2004).

Na década de 1960 começaram os estudos com abordagem desagregada comportamental, ou seja, com foco no indivíduo como unidade tomadora de decisão, em vez de conjuntos de moradores em uma zona de tráfego (Sivakumar, 2007). Dentre os vários modelos que foram desenvolvidos nessa época, o que diferenciava cada um era a técnica estatística adotada, podendo ser, por exemplo, análise discriminante ou análise *probit*. Também nessa década

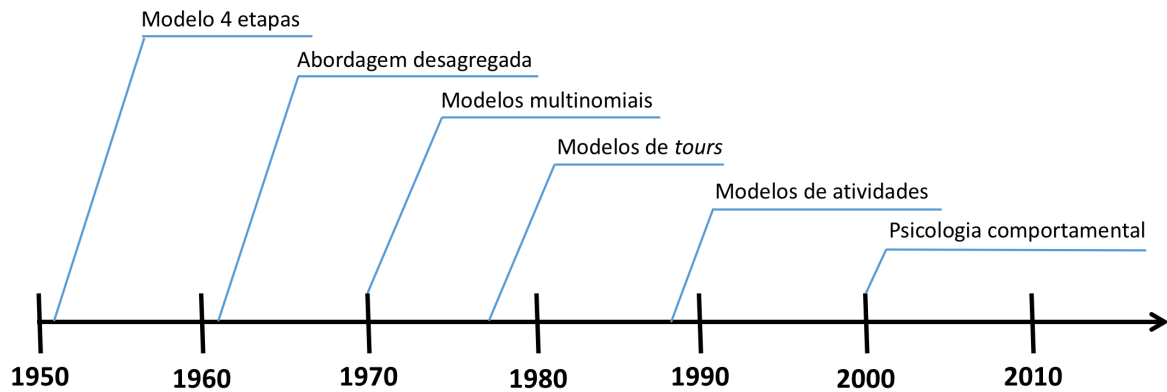
foram desenvolvidos os primeiros modelos desagregados baseados na teoria de maximização da função utilidade (Boyce & Williams, 2015).

A década seguinte, dos anos 1970, foi marcada pela introdução de modelos multinomiais de escolha multimodal, pelas inovações na coleta de dados para modelagem (pesquisas de preferência revelada), e pelo início dos modelos de demanda baseados em *tours* (Sivakumar, 2007). Os *tours* são encadeamentos de viagens que começam e terminam na residência do indivíduo. Por exemplo, as viagens “casa-trabalho” e “trabalho-casa” formam um *tour* com motivo “trabalho”.

O conceito de *tours* é essencial para o entendimento da geração de modelos que surgiu no fim da década de 1970 e início da década de 1980: os modelos baseados em atividades. Essa estrutura de modelagem decompõe os *tours* baseados na residência em viagens independentes, com foco nas atividades desenvolvidas pelo indivíduo (Maerivoet & De Moor, 2008). A viagem passa a ser entendida como uma demanda derivada da necessidade de se praticar atividades (Timmermans & Arentze, 2011). Essa abordagem será detalhada no tópico 2.5.

A modelagem baseada em atividades foi extensivamente analisada ao longo da década de 1990 e no início dos anos 2000. Ainda hoje, essa abordagem é considerada uma das mais representativas do realismo comportamental do indivíduo. Apenas as teorias comportamentais complexas, desenvolvidas no século XXI, baseadas nas teorias de psicologia ambiental, se aproximam mais do objeto modelado do que a abordagem baseada em atividades (Boyce & Williams, 2015).

Com base na revisão de literatura, foi elaborada a linha do tempo representativa da evolução dos modelos de transporte, apresentada na Figura 2.1.



**Figura 2.1:** Evolução histórica dos modelos de transporte.

Gudmundsson (2011) expõe que o planejamento de transportes é fortemente assistido por modelos de simulação, que juntamente com os indicadores e os métodos de avaliação formam a importante área de estruturação de problemas baseada na tecnologia de informação. Um exemplo da importância da modelagem de transportes para tomada de decisão em prática na atualidade é apresentada no estudo de Jonsson et al. (2011), que descreve que os modelos de transporte são ferramentas centrais no planejamento de infraestrutura na Suécia.

Outros países como Estados Unidos e Holanda (Badoe & Miller, 2000; Timmermans & Arentze, 2011), também já colocam em prática a utilização de modelos de transporte nacionais (desenvolvidos para o país, especificamente) como auxílio das tomadas de decisões governamentais.

### 2.3.3 Características e limitações da modelagem para o planejamento

Apesar da nítida utilidade da modelagem para o planejamento, ainda são feitas muitas críticas a esse processo por não produzir resultados 100% corretos. Timmermans & Arentze (2011) argumentam que a questão não é se os resultados obtidos pelos modelos são exatos, mas se, na média, eles produzem análises melhores que a intuição dos tomadores de decisão.

Os mesmos autores mencionam que a experiência pessoal dos administradores, intuição e senso comum estão sujeitos a erros, ao passo que a modelagem avançada, que o estado da arte já proporciona atualmente, possui suportes muito mais sólidos.

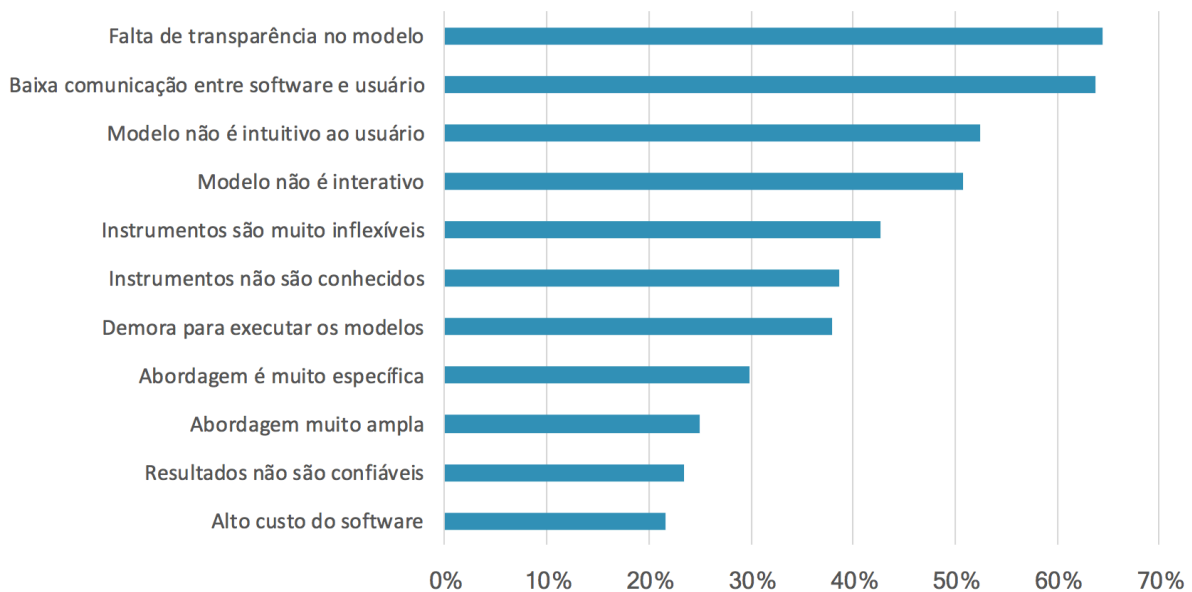


O estudo de Jonsson et al. (2011), ao descrever o conhecimento obtido com a experiência sueca de anos de aplicação da modelagem, também comenta sobre a precisão dos modelos. Segundo os autores, o modelo ideal equilibra tempo de execução e acurácia dos resultados. Esse equilíbrio é necessário porque modelos mais precisos geralmente demandam mais tempo para levantamento de dados e processamento, o que pode atrasar o processo do planejamento. Ao mesmo tempo, modelos mais simplificados, terem uma precisão muito alta, têm resultados liberados com mais rapidez, o que agiliza a solução dos problemas.

Ortúzar & Willumsen (2004) por sua vez, defendem que não apenas o resultado dos modelos é importante para o processo de planejamento. Para os autores, durante a formulação de um modelo, sua calibração e uso, o planejador pode aprender, por experimentação, sobre o comportamento e funcionamento interno do sistema em questão. Dessa forma, a intuição e experiência do planejador também são enriquecidos, permitindo um gerenciamento mais inteligente do sistema de transporte. Brömmelstroet & Bertolini (2011) também seguem essa linha de raciocínio, de que muito do que se tem conhecimento sobre elaboração de estratégias para um sistema é obtido durante o processo de criação do modelo.

Há mais de uma década, pesquisadores identificaram que as maiores limitações da modelagem de transporte eram humanas (falta de capacitação) e técnicas (baixo desenvolvimento computacional) (Ortúzar & Willumsen, 2004). O amplo acesso a aparelhos computacionais de baixo custo e alta capacidade praticamente eliminou o elemento tecnológico como empecilho ao desenvolvimento da modelagem. O gargalo que existia nessa área, portanto, era a falta de profissionais hábeis com o uso dos *softwares* de modelagem e a limitação teórica dos modelos.

Mais recentemente, Brömmelstroet & Bertolini (2011) mencionam uma pesquisa realizada na Holanda que teve por objetivo levantar os motivos pelos quais os tomadores de decisão das áreas de planejamento de transportes e uso do solo não utilizavam modelagem em seu gerenciamento. A Figura 2.2 apresenta os resultados da pesquisa. Nota-se que os motivos mais citados são relacionados aos aspectos de usabilidade dos modelos. Poucos entrevistados questionaram a utilidade da modelagem no processo de planejamento, o que enfatiza o fato de que os planejadores reconhecem as potenciais contribuições dos modelos para se desenvolver estratégias de gerenciamento mais efetivas.



**Figura 2.2:** Motivos pelos quais os planejadores em transporte e uso do solo deixam de usar a modelagem em suas análises (Traduzido de Brömmelstroet & Bertolini, 2011).

## 2.4 Modelagem de sistemas multiagente

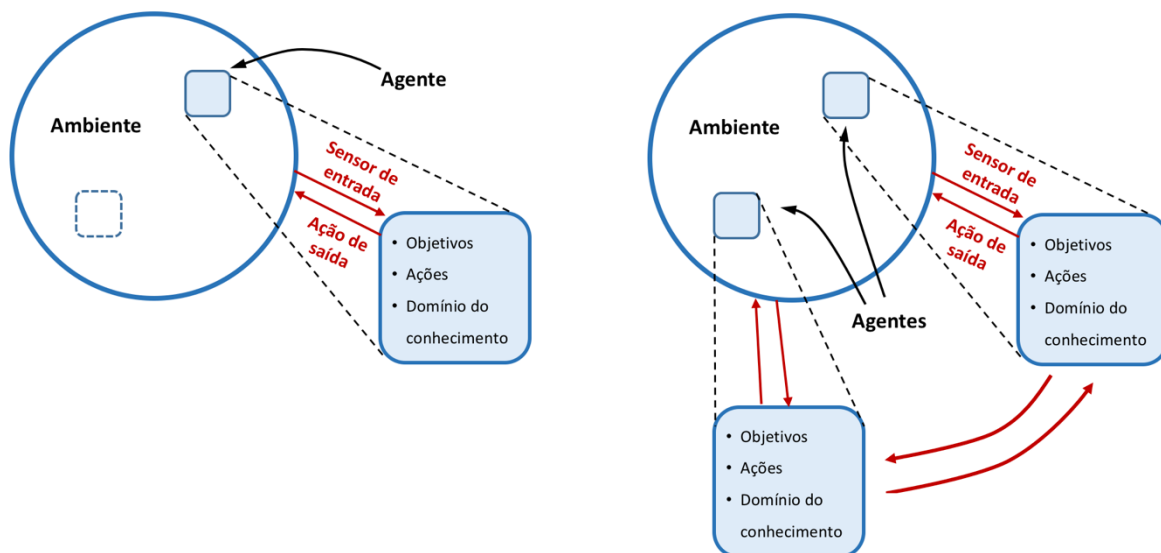
### 2.4.1 Generalidades

Um dos primeiros estudos com foco na modelagem baseada em agentes foi apresentado por Wooldridge (1992). No documento, o autor define agente como sendo um elemento ativo do sistema, que origina ações que afetam o ambiente ao seu redor, em vez de passivamente deixar o ambiente o afetar. As principais características de um agente seriam, portanto, autonomia e racionalidade. Na época, o termo se tornava cada vez mais comum no âmbito da ciência da computação, em trabalhos relacionados a inteligência artificial.

Segundo descrito por Siebers & Aickelin (2008), em simulações baseadas em agentes (Agent-Based Simulations – ABS), o pesquisador descreve explicitamente os processos de decisão dos atores microssimulados. Esses modelos são, portanto, abordagens de baixo para cima (*bottom-up*), que podem ser usadas em situações onde a variabilidade individual dos agentes não pode ser ignorada. A metodologia baseada em agentes permite a modelagem de uma população heterogênea onde cada indivíduo tem motivações e incentivos pessoais, representando a nível macro os grupos e suas interações.

A diferenciação entre sistemas de agente único (centralizado) e sistemas multiagentes é apresentada por Stone (1997). O autor descreve que sistemas centralizados possuem um único agente que toma todas as decisões, enquanto os outros agem como seus dependentes. Esse tipo de sistema ainda pode ter um certo número de indivíduos, mas se esses indivíduos enviam suas percepções para uma central de comando e recebe dela as ações que devem ser tomadas, há apenas um agente: a própria central de comando.

Como apresentado na Figura 2.3, no sistema centralizado, o agente altera, sozinho, o ambiente e suas interações. Se outros indivíduos existirem, eles são considerados parte do ambiente. Já no sistema multiagente, os indivíduos interagem entre si alterando seus objetivos e ações.



**Figura 2.3:** À esquerda, arquitetura do sistema de agente único (centralizado); À direita, arquitetura do sistema multiagente (Traduzido de Stone, 1997).

#### 2.4.2 Exemplos de aplicação da abordagem multiagente

Alguns exemplos de aplicação da simulação baseada em agentes são expostos na Tabela 2.1, conforme apresentado por Siebers & Aickelin (2008).

**Tabela 2.1:** Exemplos de aplicação de simulação baseada em agentes  
(Siebers & Aickelin, 2008)

Campo	Exemplos de aplicação
Ciências Sociais	Sociedades de insetos, dinâmica de grupo em voos, crescimento e queda de sociedades antigas, aprendizagem de grupo, propagação de epidemias, desobediência civil.
Economia	Mercado de ações, mercados autorregulados, redes de comércio, comportamento do consumidor, mercados de energia elétrica desregulados.
Ecologia	Dinâmica de populações de salmão e truta, dinâmica do uso do solo, comportamento de grupos de peixes e aves, crescimento de florestas tropicais.
Ciência Política	Direito de consumo de água em países em desenvolvimento, competição de partidos políticos, origens e padrões da violência política, divisão de poder em Estados multiculturais.

Já nos sistemas de transporte, muitos atores se encaixam nas definições de agentes autônomos, como por exemplo os motoristas, os pedestres e até mesmo as interseções semaforizadas (Bazzan, 2009).

## 2.5 Modelagem baseada em atividades

### 2.5.1 Generalidades

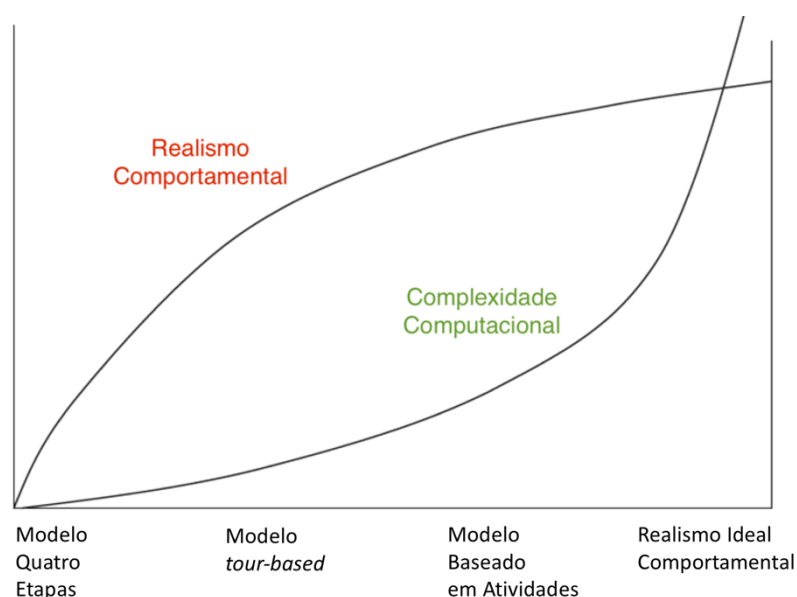
Os primeiros estudos a mencionar a modelagem baseada em atividades datam das décadas de 1970 e 1980, mas seu desenvolvimento foi maior ao longo da década de 1990. Ettema (1996) detalhou o que seria a modelagem baseada em atividades, ideia básica que perpetua até os dias atuais. Essa abordagem parte do pressuposto que o transporte não é uma demanda em si, e sim uma demanda derivada da necessidade de se realizar uma atividade. A modelagem, portanto, dá ênfase na programação de atividades diárias de cada indivíduo. O autor também destaca a importância do desenvolvimento da inteligência artificial para os modelos baseados em atividades. Essa técnica científica seria essencial para descrever o planejamento diário de atividades de forma coerente com o raciocínio humano. Entretanto, à época, essa tecnologia ainda era difícil de ser calibrada porque envolvia variáveis complexas de se estimar.

Spear (1996) apresentou algumas das considerações que devem ser feitas em modelos baseados em atividades:

- Influência do uso do solo, no sentido em que se definem as características dos estabelecimentos onde as atividades são feitas (localização, horário de funcionamento);

- Influência do sistema de transporte, como velocidades de vias, por exemplo;
- Características socioeconômicas dos domicílios da área de estudo;
- Tipos de atividades a serem realizadas, sendo elas obrigatórias ou não, com ou sem horário fixo.

Ao longo dos anos, o número de estudos relacionados aos modelos baseados em atividades aumentou, mas os conceitos básicos definidos na década de 1990 permaneceram. Além disso, muitas comparações foram feitas de forma a evidenciar as vantagens desse modelo sobre o modelo baseado em viagens (quatro etapas). Shiftan & Ben-Akiva (2011), por exemplo, identificam que a abordagem baseada em atividades é essencial para se entender a complexidade das mudanças no comportamento de viagem em resposta a novas políticas de gerenciamento de tráfego. Os autores apresentam um gráfico que compara o modelo de quatro etapas, modelos *tour-based* e o modelo baseado em atividades. Esse gráfico, apresentado na Figura 2.4, deixa claro que os modelos baseados em atividades, com um pequeno incremento computacional, promove grande aumento no realismo comportamental de sua modelagem, refletindo de uma forma razoavelmente boa o comportamento dos indivíduos.



**Figura 2.4:** Realismo comportamental vs. Complexidade computacional de modelos de transporte (Traduzida de Shiftan & Ben-Akiva, 2011).

No Brasil, o primeiro estudo com referência à modelagem baseada em atividades foi apresentado por Arruda (2005), que aplicou um modelo com essa abordagem para analisar a relação uso do solo e transportes. Desde então, poucos foram os estudos no país que mencionaram essa metodologia. Um destaque é o estudo de Lopes (2015), que propõe a construção de um modelo baseado em atividades que preencha as lacunas deixadas pelos modelos que consideram apenas o uso do solo.

### **2.5.2 Entrada e saída de um modelo baseado em atividades**

Conforme exposto por Arruda (2005), alguns dos dados de entrada dos modelos baseados em atividades são os seguintes:

- Informações geográficas;
- Informações sociodemográficas individuais (idade, gênero, situação do emprego, ocupação);
- Informações sobre os recursos disponíveis ao domicílio (propriedade de automóvel, características do domicílio, renda);
- Dados sobre os locais de realização das atividades (localização, horário de funcionamento dos estabelecimentos, nível de acessibilidade);
- Dados referentes à malha viária.

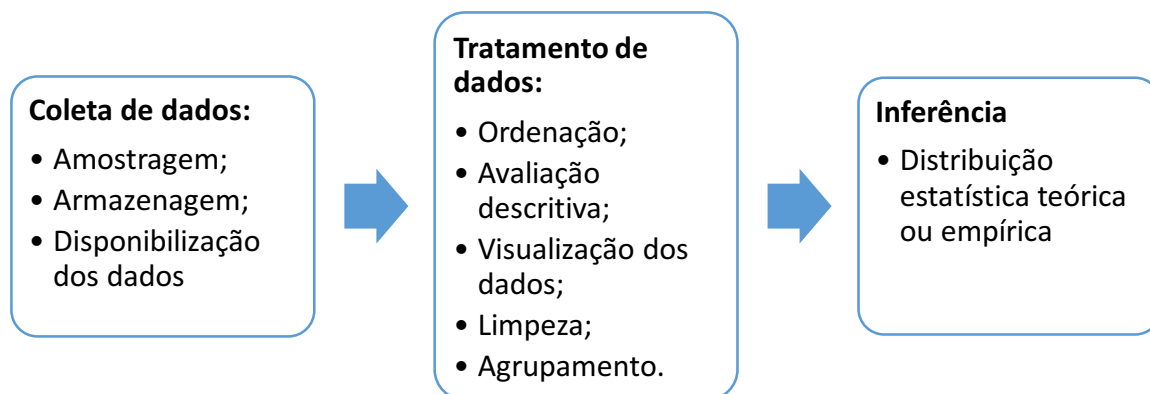
Ettema (1996) sugere que o uso de diários de atividades seja o meio mais eficiente de coleta de dados para calibração desse tipo de modelo. Uma pesquisa utilizando diários de atividades fornece informações sobre a sequência de atividades e viagens realizadas pelas pessoas no decorrer de um dia ou em múltiplos dias, além de informações socioeconômicas detalhadas sobre o domicílio e os moradores, caracterizando a desagregação do modelo (Bhat et al., 2000).

É muito importante que o instrumento de pesquisa seja bem estruturado, uma vez que a qualidade dos dados adquiridos está intimamente relacionada ao resultado do modelo. A dificuldade em montar e aplicar a pesquisa de diário de atividades, devido à sua extensão, é uma limitação da modelagem baseada em atividades (Bhat & Koppelman, 2003).

## 2.6 Tratamento de dados para modelagem

Hensher & Button (2008) afirmam que a disponibilidade de dados é o fator mais importante na validação de teorias e modelos de transporte. Limitações econômicas e temporais podem influenciar a capacidade do pesquisador de obter dados adequados para inserção no modelo, o que tem efeito direto na qualidade do resultado obtido.

Chwif & Medina (2014) também defendem que processos de simulação, não apenas de transporte, mas de uma forma geral, requerem que seus dados de entrada sejam modelados adequadamente para uma melhor qualidade do produto de simulação. Essa modelagem é incluída nas etapas mostradas na Figura 2.5.



**Figura 2.5:** Modelagem de dados de entrada para simulação (Chwif & Medina, 2014).

Segundo o processo de modelagem proposto pelos autores, a coleta de dados deve ser formalizada para permitir a identificação de erros no processo e auxiliar a consolidação dos dados obtidos. Depois, com o tratamento dos dados, deve ser gerado um banco de dados adequado para o modelo utilizado. Por fim, na etapa de inferência, deve-se preencher as lacunas do banco de dados utilizando uma distribuição estatística teórica, se houver aderência, ou empírica, se não.

Alguns pesquisadores já se dedicaram ao estudo da modelagem de dados para simulações em diversas áreas. Pereira et al. (2004), por exemplo enfatizam que a inadequação de dados disponíveis para modelagens na em meteorologia é um problema encontrado pelos

pesquisadores da área. Em seu estudo, os autores propõem, então, uma metodologia de tratamento de dados climáticos para inserção adequada em *softwares* específicos de modelagem.

Similarmente, Marin & Tomi (2010) propuseram uma metodologia de modelagem de dados para simulação de lavra (extração de minérios) de minas. Entre suas conclusões, estava a constatação de que a modelagem de dados tem papel essencial nas simulações de minas, apesar do processo não ter destaque na maioria das pesquisas sobre o tema.

Apesar de não tratarem especificamente sobre metodologia de modelagem de dados de entrada para simulações, Oliveira & Favaretto (2013) também analisaram a fase de coleta de dados para concepção de uma simulação de operação de restaurante. Os autores deram ênfase à necessidade de se desenvolverem métodos simples e rápidos para aquisição e organização de informações, uma vez que a dificuldade na obtenção de dados por vezes inibe a utilização de um modelo pelo planejador.

Não foram encontrados na literatura modelos de dados de entrada para simulação na área de engenharia de transportes, especificamente. Entretanto, muitos autores que tratam da modelagem em transporte reconhecem a relevância da correta organização das informações de entrada para obtenção de um resultado de qualidade (Ortúzar & Willumsen, 2004; Hensher & Button, 2008; Möller, 2014).

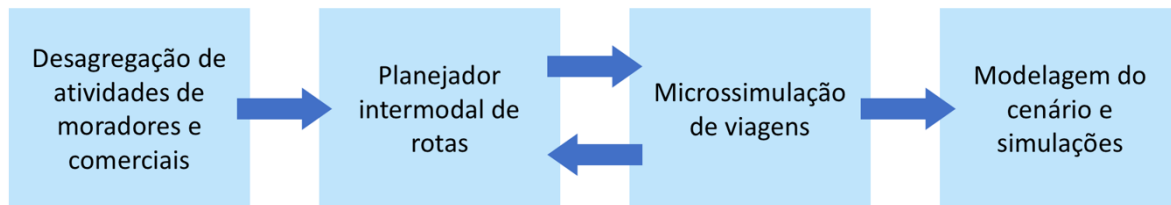
## **2.7 MATSim: Multi-Agent Transport Simulation**

### **2.7.1 Histórico do MATSim**

O desenvolvimento do MATSim começou em 1995 com a criação de um *software* anterior, chamado TRANSIMS (TRansportation ANalysis and SIMulation System) (Horni et al., 2016). Como descrito por Smith et al. (1995), o projeto TRANSIMS era parte de um programa de desenvolvimento de modelos de viagem patrocinado pelo governo dos Estados Unidos. O objetivo do projeto era desenvolver um conjunto de simulações, modelos e bases de dados mutuamente colaborativos que empregassem técnicas computacionais e analíticas de forma a criar um ambiente de análise integrado dos sistemas de transporte (Smith et al., 1995). Os resultados das simulações serviriam para auxiliar planejadores de transporte,



engenheiros e outros tomadores de decisão que lidem com gerência do uso do solo, segurança viária e economia de transportes, por exemplo. A arquitetura básica do TRANSIMS é apresentada na Figura 2.6.



**Figura 2.6:** Arquitetura do TRANSIMS (Traduzida de Smith *et al.*, 1995).

Ao longo dos anos seguintes, o físico alemão Kai Nagel deu continuidade ao trabalho de modelagem de transportes partindo da arquitetura desenvolvida no TRANSIMS. Seu objetivo final era produzir um programa de código aberto. Para isso, trabalhou em colaboração com o engenheiro civil alemão Kay Axhausen por mais de 10 anos desenvolvendo os sistemas do *software* MATSim, que hoje é uma ferramenta flexível, de código aberto, para análise de comportamento de viagens e condições da via (Horni et al., 2016).

### 2.7.2 Características básicas do MATSim

O MATSim é uma ferramenta *open-source* de simulação de multiagentes, baseada em atividades e implementada na linguagem Java. O *software* foi desenvolvido para a modelagem de um cenário de um dia, que é a unidade padrão de modelos baseados em atividades. Sua metodologia é baseada no princípio coevolutivo. Cada agente otimiza repetidamente seu programa de atividades diárias, baseado no conflito com todos os outros agentes participantes da infraestrutura de transporte. Esse processo é similar ao ciclo iterativo de alocação de viagens do modelo clássico de quatro etapas. Entretanto, o modelo do MATSim vai além, no sentido em que pode incorporar outras variáveis no ciclo iterativo, tais como escolha do tempo de saída, escolha modal e escolha de destino (Horni et al., 2016).

### 2.7.3 Estudos anteriores usando MATSim

Horni et al. (2016) apresentam estudos conduzidos em todo o mundo, que utilizaram o MATSim como ferramenta. Uma tabela que resume as principais características desses trabalhos consta no Apêndice B.

Boa parte dos modelos, como os feitos para as cidades de Cingapura, Londres e Nova Iorque, por exemplo, tiveram como objetivo principal desenvolver uma ferramenta que auxiliasse a tomada de decisão no planejamento de transportes da região. Entretanto, é possível perceber que o MATSim pode ser usado com outros focos de pesquisa. Modelos gerados para cidades na Turquia, Bélgica e Indonésia, por exemplo, tinham como intuito a previsão de demanda por transportes durante a evacuação de áreas de risco de enchentes. Já em Munique, na Alemanha, e Tampa, nos Estados Unidos, pesquisadores utilizaram o MATSim para verificar a contribuição dos modos de transporte com a poluição atmosférica. Outros exemplos são os estudos realizados em Bruxelas e em Seattle, com o objetivo de combinar o MATSim a modelos baseados no uso do solo, como o UrbanSim.

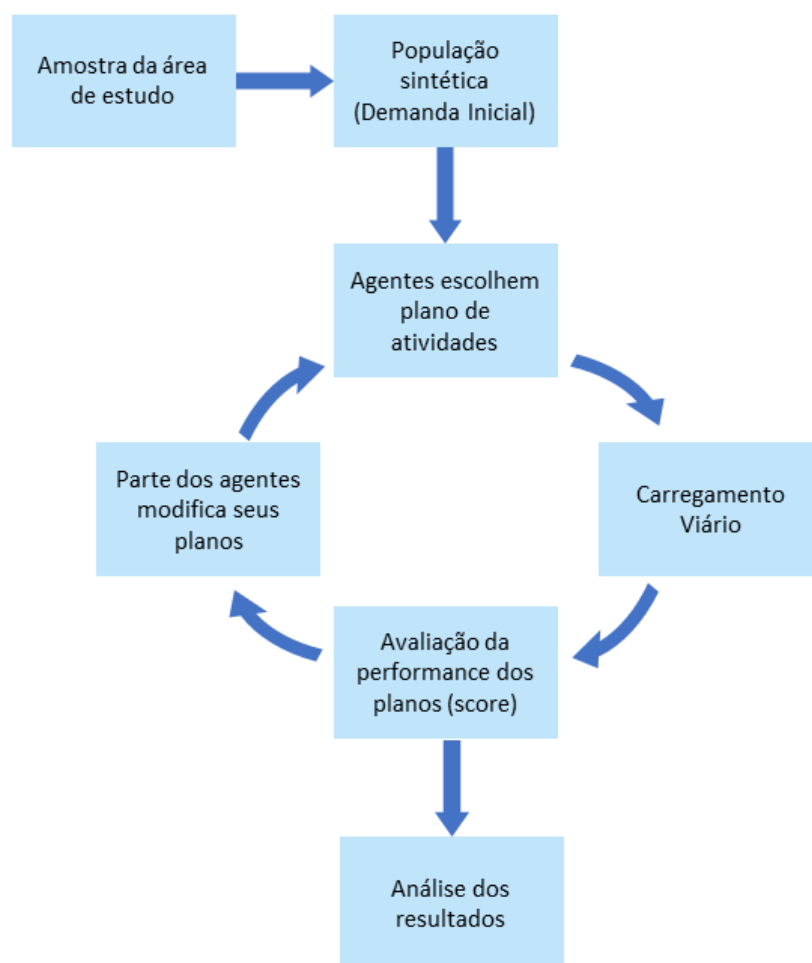
No Brasil, o primeiro estudo a utilizar o MATSim como ferramenta foi feito por Bicudo (2015), com o objetivo de construir um modelo de tráfego baseado em agentes para a cidade de Joinville, em Santa Catarina. Esse modelo foi o primeiro passo para um projeto de modelagem completa dos transportes da região nordeste do estado, que visa englobar os modos terrestre, aéreo e marítimo (Horni et al., 2016). Como conclusão do trabalho, o autor avaliou como boa a comparação entre os resultados obtidos pelo modelo e as contagens realizadas nas principais vias da área.

Outro estudo brasileiro que fez uso do MATSim foi conduzido por Medrano (2016). Em sua tese, o autor formula a estrutura teórica do comportamento de transporte, subsidiada pelo modelo ontológico de Bunge. O modelo criado é, então, denominado modelo intencional. Por fim, o autor compara a abordagem baseada em viagens do modelo quadro etapas, do *software* VISUM, e a abordagem baseada em atividades do MATSim, a partir da perspectiva teórica do modelo intencional. Com as simulações comparativas entre os modelos MATSim e VISUM, conclui-se que o MATSim fica mais próximo do comportamento humano.

Não apenas os estudos realizados no Brasil, mas também a vasta maioria das pesquisas conduzidas em outros países consideram o MATSim um grande avanço nos modelos de demanda ao produzir uma gama de tipos de informação em seus resultados.

#### 2.7.4 Ciclo iterativo do MATSim

Um ciclo iterativo do MATSim é composto pelas etapas apresentadas na Figura 2.7. As primeiras etapas incluem a entrada da demanda inicial e a geração de população sintética e planos de atividades.



**Figura 2.7:** Processo de simulação do MATSim.

A primeira etapa do uso do MATSim é a geração da demanda inicial. A partir de dados empíricos de uma amostra da área de estudo, o MATSim gera uma população sintética, de número de habitantes equivalente ao real, refletindo as características coletadas na amostra. Essa população sintética é composta por agentes. Cada agente dentro do MATSim simula o comportamento de um usuário individual dentro do sistema de transporte (Horni et al., 2016).

Baseado nos dados empíricos da amostra de estudo, são gerados planos de atividades que incluem o conjunto de atividades que o indivíduo tem que realizar (Medrano, 2016). Além disso, os planos de atividades incluem a escolha modal que o indivíduo faz para realizar suas atividades. O *software*, no entanto, não está preparado para modelar completamente a demanda por viagens. A sequência e duração média de atividades, por exemplo, devem ser fornecidas como dados de entrada no MATSim.

Para um mesmo agente podem ser geradas várias alternativas de planos de atividades, cuja atratividade é medida pelos *scores*, que é o valor que avalia o desempenho do plano. Os critérios para definição se o plano de atividades teve bom ou mau desempenho serão detalhados no tópico 2.7.5.

Planos de atividades diárias com seus respectivos *scores*, são armazenados na memória dos agentes. Na primeira etapa do ciclo, cada agente seleciona um plano de atividades da sua memória, baseado no *score* que é computado ao fim de cada iteração. Em seguida, ocorre a etapa de carregamento viário, que é quando as viagens dos agentes são simuladas. O ciclo termina com a avaliação da performance de cada plano escolhido pelos agentes e com a atribuição de *scores* para esses planos: quanto melhor o plano, maior o *score*, e vice-versa.

Quem decide os critérios para a atribuição de *scores* são os agentes de cada região. Alguns podem preferir uma viagem congestionada, mas no conforto do carro, outros podem escolher uma viagem mais barata de transporte público e outros podem preferir usar bicicleta. A metodologia para atribuição de *scores* é apresentada no tópico 2.7.5.

Antes do início de um novo ciclo iterativo, uma parcela dos agentes (geralmente 10%, mas esse número é configurável), pode fazer alguma modificação no plano de atividades escolhido anteriormente. Essa modificação pode ser no tempo de partida, rota, modo ou destino e pode ser feita de forma aleatória ou por estimativa de melhor resposta, procurando sempre a

escolha ótima. A escolha de rota, por exemplo, costuma ser alterada pela estimativa de melhor resposta, enquanto a alteração do tempo de partida e modo de transporte geralmente são mutações aleatórias.

### 2.7.5 Atribuição de Scores

A metodologia de atribuição de *scores* do MATSim foi desenvolvida por Charypar & Nagel (2005) com foco em simulações de modelos multiagente, tais como o próprio MATSim e o precursor TRANSIMS. Os autores optaram por usar a metodologia de algoritmos genéticos para que o *software* procurasse selecionar bons planos de atividades diárias para os agentes. Considerando a impossibilidade de alocar o melhor plano possível para cada pessoa do modelo, é suficiente para o problema em questão encontrar planos que sejam apenas “bons”.

A implementação do algoritmo genético se dá por uma função de utilidade (*utility function*), que é uma função de ajuste (*fitness function*) que define o parâmetro de “bom plano de atividades” para um agente. Uma vantagem do algoritmo genético é que a função de ajuste pode ser adaptada facilmente de acordo com as preferências do usuário (Charypar & Nagel, 2005). Conforme desenvolvido pelos autores, a função básica de utilidade, que resulta no score de um plano de atividades  $S_{plan}$ , é dada pela Eq. 2.1 abaixo. Em seguida, seus parâmetros são definidos.

$$S_{plan} = \sum_{q=0}^{N-1} S_{act,q} + \sum_{q=0}^{N-1} S_{trav,mode(q)} \quad \text{Eq. 2.1}$$

Sendo:

$S_{act,q}$  = Vantagens de todas as atividades;

$S_{trav,mode(q)}$  = Utilidades (negativas) de todas as viagens;

$N$  = Número de atividades;

$q$  = Viagem subsequente à atividade  $q$ ;

Para o cálculo do *score*, a última atividade é combinada com a primeira a fim de se ter um número igual de viagens e atividades. A função que define a utilidade de uma atividade  $q$  é definida pela Eq. 2.2.

$$S_{act,q} = S_{dur,q} + S_{wait,q} + S_{late.ar,q} + S_{early.dp,q} + S_{short.dur,q} \quad \text{Eq. 2.2}$$

Sendo:

$S_{dur,q}$  a utilidade de se realizar a atividade  $q$ , levando em conta o horário de abertura dos locais onde as atividades acontecem;

$S_{wait,q}$  o tempo de espera, por exemplo, em frente ao local onde será feita a atividade  $q$ , antes de sua abertura;

$S_{late.ar,q}$  a utilidade negativa por se chegar atrasado no local da atividade  $q$ .

$S_{early.dp,q}$  a penalidade por não se permanecer tempo suficiente na atividade  $q$

$S_{short.dur,q}$  é a utilidade negativa da realização de uma atividade  $q$  muito curta

As subcontribuições da **Eq. 2.2** são detalhadas no Apêndice C.

Já o termo  $S_{trav,mode(q)}$  da Eq. 2.3, que incorpora as desvantagens de uma viagem  $q$ , é dado pela Eq. 3-7 abaixo, sendo suas subcontribuições detalhadas em seguida.

$$S_{trav,q} = C_{mode(q)} + (\beta_{trav,mode(q)})(t_{trav,q}) + (\beta_m)(\Delta m_q) + [\beta_{d,mode(q)} + (\beta_{d,mode(q)})(\gamma_{d,mode(q)})](d_{trav,q}) + (\beta_{transfer})(x_{transfer,q}) \quad \text{Eq. 2.3}$$

Sendo:

$C_{mode(q)}$  = Uma constante específica para o modo;

$\beta_{trav,mode(q)}$  = Utilidade marginal direta do tempo gasto utilizando determinado modo;

$t_{trav,q}$  = Tempo de viagem entre os locais das atividades  $q$  e  $q+1$ ;

$\beta_m$  = Utilidade marginal do custo da viagem (normalmente positiva);

$\Delta m_q$  = Variação do custo causado por taxas ou pedágios no caminho tomado (normalmente negativa ou nula);

$\beta_{d,mode(q)}$  = Utilidade marginal da distância de viagem (normalmente negativa ou nula);

$\gamma_{d,mode(q)}$  = Taxa de custo em função da distância viajada usando determinado modo (normalmente negativo ou zero);

$d_{trav,q}$  = Distância de viagem entre os locais das atividades  $q$  e  $q+1$ ;

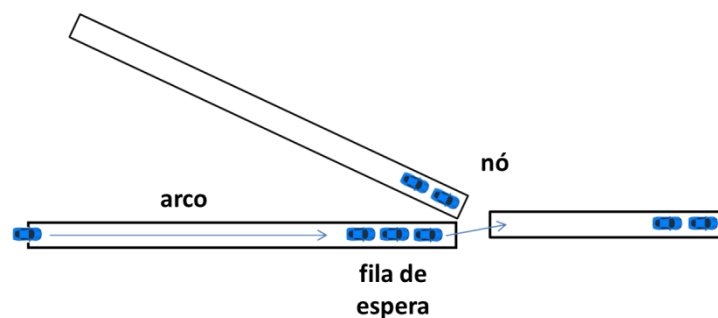
$\beta_{transfer}$  = Desvantagens por transferência entre modos de transporte público;

$x_{transfer,q}$  = Variável binária 0 ou 1 sinalizando se ocorre transferência entre a viagem anterior e a atual.

### 2.7.6 Modelagem do fluxo de tráfego

Cada processo dentro do algoritmo do MATSim é executado por uma parcela de código chamada módulo. Um exemplo que já foi descrito anteriormente é o módulo responsável pela geração de população sintética. Já o fluxo de tráfego é modelado em um módulo chamado de *mobsim* dentro do *software*. O MATSim já é programado com duas opções de módulo modelador de fluxo de tráfego: o QSim e o JDEQSim, mas por ser um *software* de código aberto, é possível configurar módulos personalizados (Horni et al., 2016).

Ambas as configurações-padrão de modelagem de fluxo de tráfego do MATSim seguem a abordagem microscópica baseada na teoria das filas (Figura 2.8). De acordo com essa metodologia, para acessar um trecho viário por meio de uma interseção, o veículo é adicionado a uma fila de espera. Ele permanece lá até que o veículo seja o primeiro da fila, o tempo para se percorrer o trecho em fluxo livre tenha passado e a interseção permita o acesso. Essa abordagem é muito eficiente, mas tem como desvantagem uma resolução reduzida por não capturar os efeitos de *car-following* (Horni et al., 2016).



**Figura 2.8:** Teoria das filas (Horni et al., 2016).

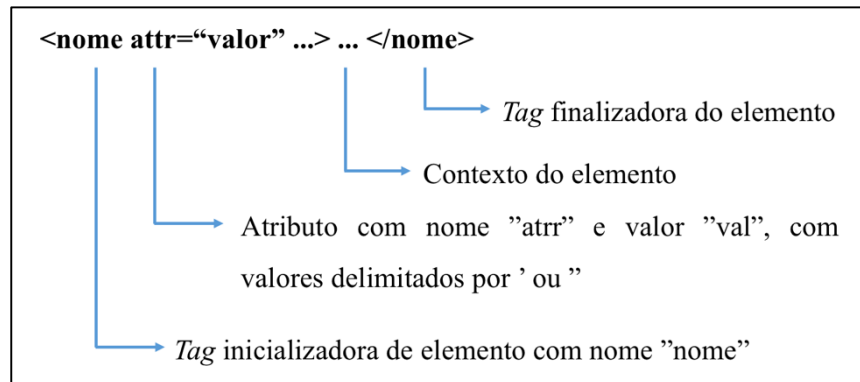
### 2.7.7 Arquivos de entrada no MATSim

O MATSim recebe como entrada, no mínimo, os seguintes arquivos:

- *config.xml*, contendo as opções de configuração gerais do *software*, definidas pelo usuário;
- *network.xml*, com a descrição da rede viária;
- *plans.xml*, com as informações das demandas de viagem.

Percebe-se que todos os arquivos de entrada no MATSim possuem a extensão “.xml”. A sigla que forma a extensão “XML” significa *Extensible Markup Language* (Linguagem de Marcação Extensiva, em tradução livre). Essa linguagem permite a descrição de dados em formato estruturado, facilitando a declaração do conteúdo (Furtado Junior, 2008). De forma simples, um documento XML é um arquivo de texto com etiquetas (*tags*) de marcação em estrutura de árvore. As etiquetas de marcação denotam a estrutura apresentada na Figura 2.9.

Os arquivos de entrada do MATSim são descritos nos tópicos seguintes, juntamente com a apresentação da estrutura em árvore das informações que os compõem.



**Figura 2.9:** Estrutura de um arquivo de texto XML.

#### a) Arquivo config.xml

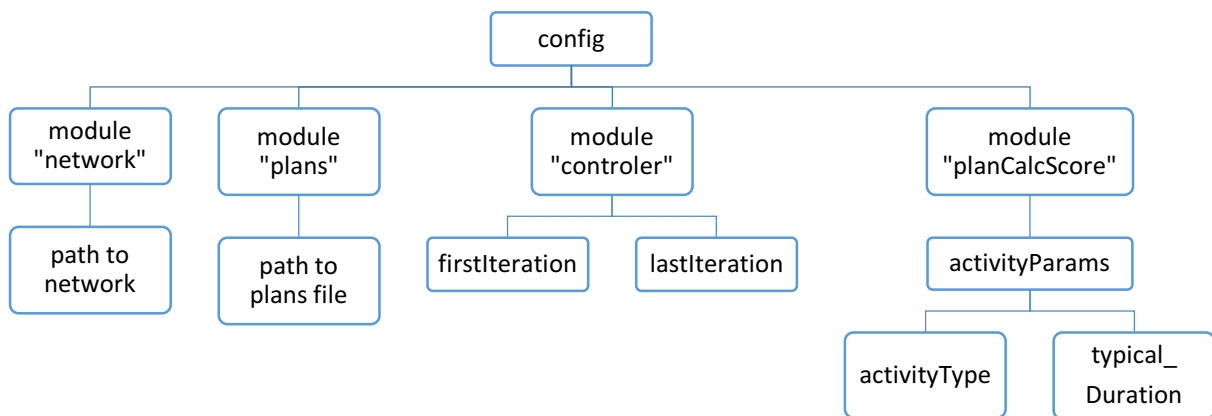
O arquivo *config.xml*, ou arquivo de configuração do MATSim, é a ponte entre o usuário e o *software*, que especifica como a simulação deve funcionar. Ele é um arquivo de texto dividido em módulos, que podem se referir a parâmetros relacionados à rede viária, às atividades e ao próprio algoritmo de modelagem do MATSim. Um arquivo de configuração deve ter, no mínimo, os seguintes parâmetros:



- Local onde está armazenado o arquivo *network.xml*;
- Local onde está armazenado o arquivo *plans.xml*;
- Número de iterações que devem ser executadas no modelo;
- Características das atividades
  - Nome (exemplo: trabalho);
  - Duração típica (exemplo: 12:00:00).

Além das informações mínimas necessárias, outras configurações-padrão podem ser alteradas no arquivo *config.xml*. Alguns exemplos são: parâmetros de escolha de rota; reprogramação de alguns planos entre iterações; parâmetros de cálculo dos scores, entre outros.

A estrutura em árvore do arquivo *config.xml* é bastante simples, na qual os elementos e atributos são basicamente caminhos até o local dos arquivos ou as definições de entrada do usuário que já foram mencionadas acima (iterações a serem executadas, características das atividades). Um exemplo dessa estrutura está apresentado na Figura 2.10, seguido por um exemplo de um arquivo em texto com os parâmetros mínimos necessários para o MATSim na Figura 2.11.



**Figura 2.10:** Árvore de dados do arquivo *config.xml*.

```

1  <!-- Cabeçalho do arquivo .xml, com a versão da linguagem utilizada e a configuração
2  do formato de texto para o MATSim -->
3  <?xml version="1.0" encoding="UTF-8"?>
4  <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v1.dtd">
5  <config>
6
7  <!-- Módulo para indicar o caminho para o arquivo network.xml -->
8  <module name="network">
9      <param name = "inputNetworkFile" value="path-to-network-file"
10     />
11 </module>
12
13 <!-- Módulo para indicar o caminho para o arquivo plans.xml -->
14 <module name="plans">
15     <param name="inputPlansFile" value="path-to-plans-file"
16     />
17
18 <!-- Módulo para indicar o número de iterações a serem feitas -->
19 <module name="controler">
20     <param name="firstIteration" value="0" />
21     <param name="lastIteration" value="0" />
22 </module>
23
24 <!-- Módulo para caracterizar as atividades, com seu nome e duração típica. Nesse caso,
25 respectivamente as atividades "h" (home) e "w" (work) -->
26 <module name="planCalcScore">
27     <parameterset type= "activityParams">
28         <param name="activityType" value="h"/>
29         <param name="typicalDuration" value="12:00:00"/>
30     </parameterset>
31
32     <parameterset type= "activityParams">
33         <param name="activityType" value="w"/>
34         <param name="typicalDuration" value="08:00:00"/>
35     </parameterset>
36 </module>

```

**Figura 2.11:** Exemplo de arquivo de texto *config.xml*  
(Adaptado de Horni et al., 2016, com comentários da autora).

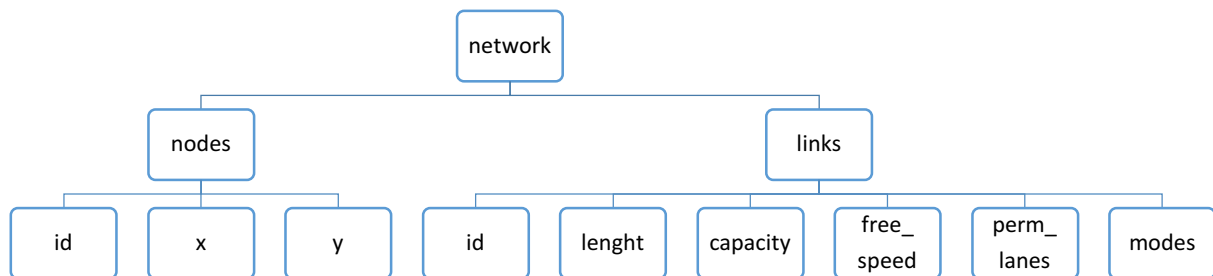
#### b) Arquivo network.xml

O arquivo *network.xml* inclui as características da infraestrutura por onde os agentes podem se movimentar. A rede viária é composta de nós (*nodes*) e arcos (*links*), que correspondem, na teoria gráfica, a vértices e arestas. Além da identificação de nós e arcos em posições com coordenadas x e y, as seguintes características devem constar no arquivo de rede viária:

- Comprimento do arco;
- Capacidade do arco;
- Velocidade máxima permitida no arco;
- Número de faixas disponíveis na direção especificada do arco;
- Quais modos são permitidos a transitar nesse arco.

É importante ressaltar que todos os arcos são unidirecionais. Para vias de mão dupla, dois arcos devem ser definidos com direções diferentes.

A estrutura em árvore do arquivo *network.xml* é a representação da rede viária em grafos, com elementos do tipo *nodes* (nós) e *links* (arcos). Cada tipo de elemento possui atributos específicos: os nós possuem apenas suas coordenadas no sistema de referência geográfica escolhido, enquanto os arcos podem apresentar capacidade de fluxo de tráfego, número de faixas, velocidade máxima permitida e, obviamente, os dois nós que compõem suas extremidades. Um exemplo dessa estrutura está apresentado na Figura 2.12, seguido por um exemplo de um arquivo em texto com os parâmetros mínimos necessários para o MATSim na Figura 2.13.



**Figura 2.12:** Árvore de dados do arquivo *network.xml*.

```

1  <!-- Cabeçalho do arquivo .xml, com a versão da linguagem utilizada e a configuração
2  do formato de texto para o MATSim -->
3  <?xml version="1.0" encoding="UTF-8"?>
4  <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v1.dtd">
5  <config>
6
7  <!-- Inicia uma nova rede viária e a nomeia -->
8  <network name="example network">
9
10     <!-- Caracteriza os nós da via -->
11     <nodes>
12         <node id="1" x="0.0" y="0.0" />
13         <node id="2" x="1000.0" y="0.0" />
14         <node id="3" x="1000.0" y="1000.0" />
15     </nodes>
16
17     <!-- Caracteriza os arcos da via -->
18     <links>
19         <link id = "1" from="1" to="2" length="3000.00" capacity="3600"
20             freespeed="27.78" permlanes="2" modes="car"/>
21         <link id = "2" from="2" to="3" length="4000.00" capacity="1800"
22             freespeed="27.78" permlanes="1" modes="car"/>
23         <link id = "3" from="3" to="2" length="4000.00" capacity="1800"
24             freespeed="27.78" permlanes="1" modes="car"/>
25         <link id = "4" from="3" to="1" length="6000.00" capacity="3600"
26             freespeed="27.78" permlanes="2" modes="car"/>
27     </links>
28 </network>
  
```

**Figura 2.13:** Exemplo de arquivo de texto *network.xml* (Adaptado de Horni et al., 2016, com comentários da autora).

c) Arquivo plans.xml

O arquivo *plans.xml* incorpora a demanda descrita pelos planos de atividades dos agentes. Esse documento é a estrutura de dados central do MATSim. O arquivo é composto por uma lista de pessoas, cada uma com seus respectivos planos, contendo suas respectivas atividades e deslocamentos. Para cada pessoa, exatamente um plano é marcado como “selecionado”, que o faz ser simulado durante a etapa de carregamento viário. Um arquivo de demanda deve apresentar, no mínimo, um plano de atividades para cada pessoa devidamente identificada, contendo:

- Tipo da atividade (exemplo: casa, trabalho...);
- Localização da atividade (no final de qual *link*?);
- Tempo de encerramento da atividade;
- Modo escolhido para o deslocamento até a atividade seguinte (exceto para a última atividade do dia).

É possível definir rotas específicas para cada deslocamento, mas isso não é necessário. Quando a simulação começa, o MATSim automaticamente calcula uma rota adequada baseado na proximidade dos arcos com as atividades sucessivas. Entretanto, para que esse processo ocorra, é preciso ativá-lo no módulo *Controler* do arquivo *config.xml*.

A estrutura em árvore do arquivo *plans.xml* é a representação dos planos de cada agente. A raiz da estrutura é o indivíduo, e seus atributos imediatos são os planos de atividades. Cada plano é composto pelo tipo de atividade da origem, vias utilizadas (*links*) horário de saída, modo de transporte escolhido, e horário de chegada. Um exemplo dessa estrutura está apresentado na Figura 2.14, seguido por um exemplo de um arquivo em texto com os parâmetros mínimos necessários para o MATSim na Figura 2.15.

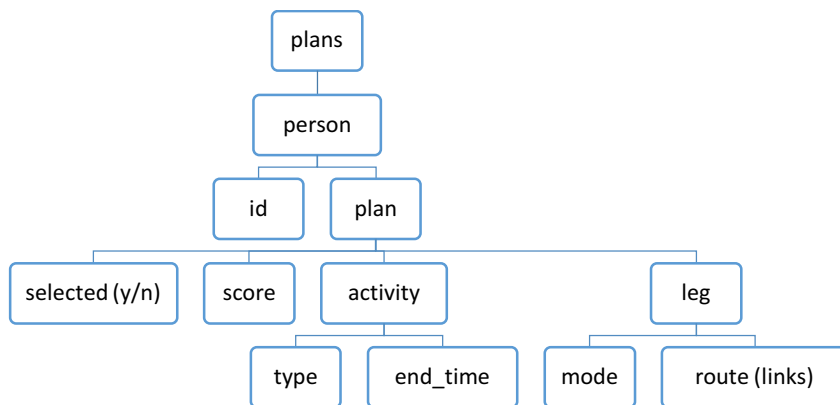


Figura 2.14: Árvore de dados do arquivo *plans.xml*.

```

1  <!-- Cabeçalho do arquivo .xml, com a versão da linguagem utilizada e a configuração
2  do formato de texto para o MATSim -->
3  <?xml version="1.0" encoding="UTF-8"?>
4  <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v1.dtd">
5  <config>
6
7  <!-- Inicia um novo arquivo de planos -->
8  <plans>
9
10     <!-- Insere uma nova pessoa -->
11     <person id="1">
12         <plan selected="yes" score="93.2987721">
13             <act type="home" link="1" end_time="07:16:23"/>
14             <leg mode="car">
15                 <route type="links">1 2 3</route>
16             </leg>
17             <act type="work" link="3" end_time="17:38:34"/>
18             <leg mode="car">
19                 <route type="links">3 1</route>
20             </leg>
21             <act type="home" link="1"/>
22         </plan>
23     </person>
24
25     <!-- Insere uma nova pessoa -->
26     <person id="2">
27         <plan selected="yes" score="144.39002">
28             ... <!-- repetir até completar todas as pessoas da população -->
29         </plan>
30     </person>
31 </plans>
  
```

Figura 2.15: Exemplo de arquivo de texto *network.xml*.  
(Adaptado de Horni et al., 2016, com comentários da autora)

## 3. MÉTODO

### 3.1 Introdução

Nesse capítulo, será apresentado o método desenvolvido para obtenção e tratamento dos dados necessários à simulação de um modelo baseado em atividades com o *software* MATSim. A metodologia de tratamento de dados desenvolvida para obter cada arquivo é apresentada em tópicos, juntamente com os programas utilizados para executar cada método. Simultaneamente, para ilustrar a aplicação da metodologia desenvolvida, é feita sua aplicação na área de estudo do Distrito Federal e Entorno.

### 3.2 Geração da rede viária (*network.xml*)

#### 3.2.1 *Softwares* e arquivos necessários

Para a execução da metodologia desenvolvida para gerar a rede viária como dado de entrada no MATSim não é necessário nenhum arquivo prévio, são necessários apenas os seguintes *softwares* gratuitos:

- Quantum GIS com *plugin Network*, ou outro *software* de GIS;
- Sublime Text, ou outro compilador.

O Quantum GIS (QGIS) é um *software* de geoprocessamento com licença livre (*open source*) fruto de um projeto voluntário com contribuição de programadores de todo o mundo (QGIS Foundation, 2017). Sendo uma ferramenta de SIG (Sistema de Informações Geográficas), o programa permite captura, armazenamento, manipulação e análise de dados referenciados espacialmente (Cavalcante, 2015).

O QGIS, diferentemente de *softwares* de SIG privados, como o ArcGIS, não possui nenhum formato de arquivo exclusivo, mas é compatível com os formatos mais comuns de arquivos georreferenciados, como, por exemplo, o Shapefile.

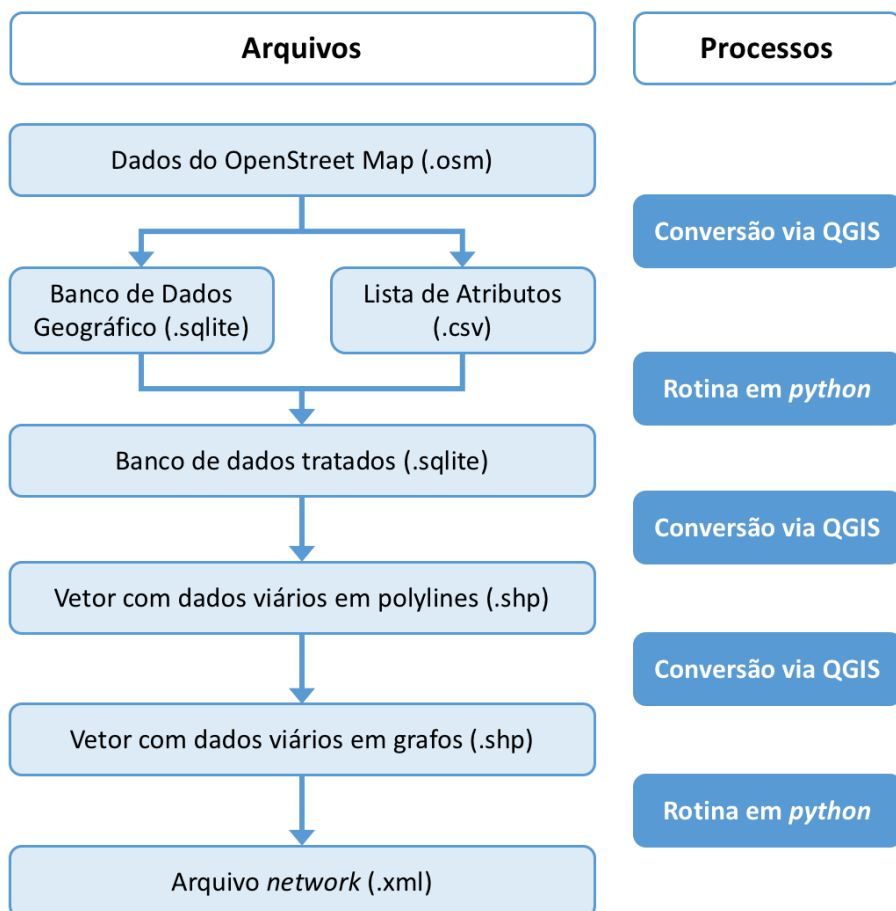
Shapefile é o formato de arquivos nativo dos *softwares* de geoprocessamento da empresa ESRI, incluindo o ArcGIS e ArcMap. Esse tipo de arquivo armazena atributos de vetores

geográficos, tais como posição e forma (ESRI ArcGIS, 2017). Entretanto, aplicações *open source* como o Quantum GIS também têm compatibilidade com arquivos do tipo Shapefile.

Sublime Text é um editor de texto com suporte a códigos-fonte e compilação de várias linguagens de programação, incluindo Python e C++. Sugere-se o uso deste *software* em especial por ser gratuito e compatível com diversos sistemas operacionais (*Windows, Mac OS, Linux*). Entretanto, qualquer outro *software* com suporte a Python pode ser utilizado para manipular os scripts desenvolvidos nessa metodologia.

### 3.2.2 Método para geração da rede viária

A estrutura metodológica desenvolvida para gerar o arquivo de rede viária (*network.xml*) é ilustrada na Figura 3.1 e descrita em detalhe no texto em seguida.



**Figura 3.1:** Esquema da metodologia proposta para geração e tratamento de dados geográficos.

Existem alguns métodos já desenvolvidos para se obter a rede viária no formato exigido para dado de entrada no MATSim (*xml*). O que difere em cada um desses métodos é o dado em estado bruto disponível para o estudo em questão. Bicudo (2015), por exemplo, partiu de um arquivo em formato Shapefile com dados sobre a rede viária de Joinville, cidade onde executou seu estudo. Utilizando *scripts* em linguagem de programação Python e criando hipóteses a respeito dos dados faltantes, o autor tratou as informações e as converteu ao formato de entrada no MATSim.

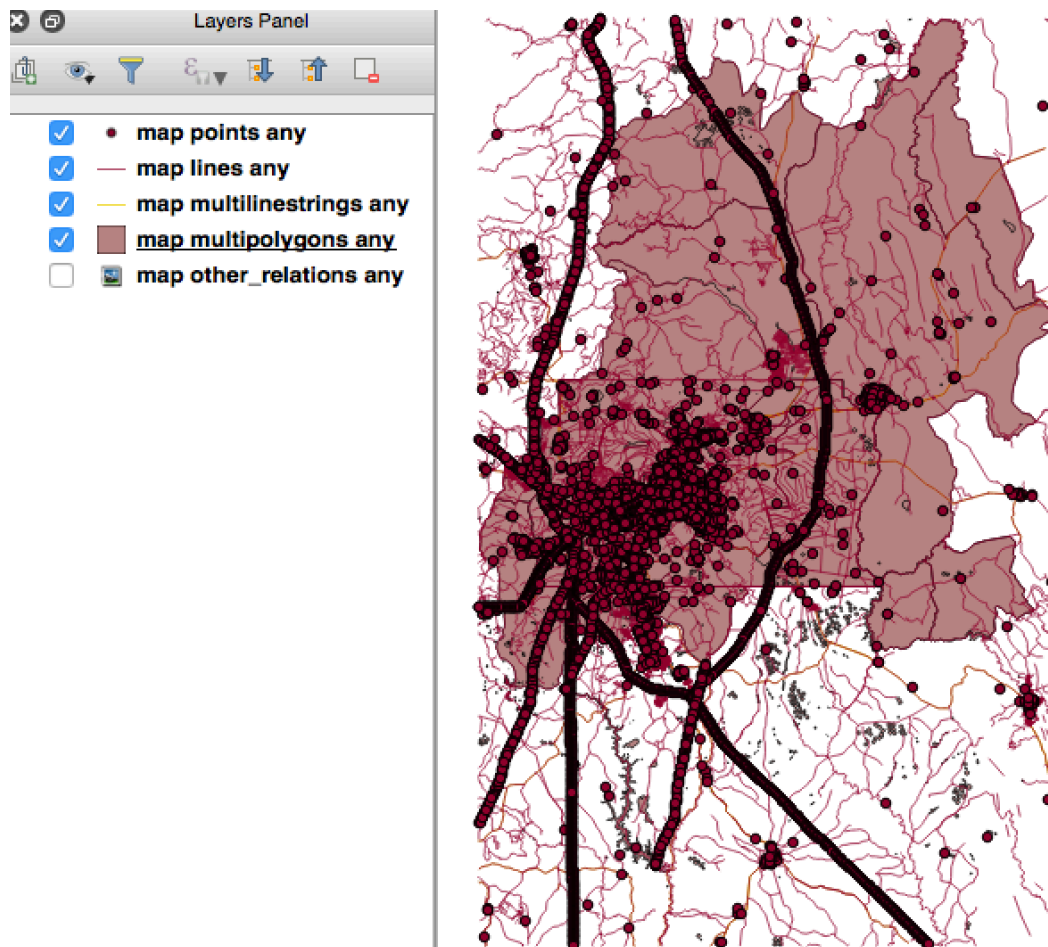
No presente estudo, apesar de existirem alguns arquivos governamentais disponíveis a respeito da rede viária do Distrito Federal, optou-se por simular a falta completa de dados para desenvolver uma metodologia com potencial aplicação generalizada aos municípios brasileiros. O processo de obtenção de informações viárias se deu, então, nas etapas descritas a seguir.

a) Importação da rede do *Open Street Map*.

O banco de dados *Open Street Map*, disponível no domínio <http://www.openstreetmap.org/>, é um banco de dados geográficos colaborativo e aberto, ou seja, suas informações são livres para uso com qualquer fim. Pesquisadores de todo o mundo mantêm o banco atualizado constantemente, incluindo dados sobre redes de transporte (vias para automóveis, ciclovias, trilhos), uso do solo e pontos de interesse (OpenStreet Map Wiki, 2017).

Para importar as informações do banco de dados, utilizou-se o *plugin* do *Open Street Map* no *software* Quantum GIS. É necessária a entrada com os dados de latitudes e longitudes da área a ser importada. No caso do Distrito Federal e Entorno, foi selecionada a mesma área de referência dos dados da pesquisa origem-destino de 2008, pois as contagens utilizadas para validar o modelo seriam retiradas dessa pesquisa. Os dados importados cobriam a área retangular entre os pontos (-14,535074; -48,422712) e (-17,003822; -46,835524), latitude e longitude, respectivamente. A visualização do arquivo importado é representada na Figura 3.2.





**Figura 3.2:** Arquivo importado pelo *software* Quantum GIS da base *Open Street Map*.

No visualizador GIS, é possível identificar que o vetor importado do Open Street Maps possui: uma camada de polígonos (“*multipolygons any*”), com algumas características sobre o uso do solo das regiões; uma camada de pontos de interesse, incluindo elementos de tráfego (semáforos, travessias de pedestres, controladores de velocidade) e também instalações como escolas, restaurantes, comércio. A camada de pontos será tratada em detalhe no item **Error! Reference source not found.** Na etapa de tratamento de dados da rede viária, a camada utilizada é a de “*lines*”, que são geometrias formadas por múltiplas linhas. No banco de dados do *Open Street Map*, cada geometria é um segmento viário com seus próprios atributos. Um exemplo de tabela de atributos da camada “*lines*” é apresentado na Figura 3.3.

osm_id	name	highway	waterway	aerialway	barrier	man_made	other_tags
13475771	Via W5 Sul	secondary	NULL	NULL	NULL	NULL	"maxspeed"=>"60","surface"=>"paved"
13475739	Via HCE Um	secondary	NULL	NULL	NULL	NULL	"foot"=>"yes","lanes"=>"2","maxspeed"=>"60","oneway"=>"yes","side...
13475743	Via HCE Um	primary	NULL	NULL	NULL	NULL	"bicycle"=>"yes","cycleway"=>"opposite","foot"=>"yes","lanes"=>"2",...
479865106	Sces Trecho 3	trunk	NULL	NULL	NULL	NULL	"lanes"=>"2","maxspeed"=>"80","oneway"=>"yes","surface"=>"paved"
5081572	S1	trunk	NULL	NULL	NULL	NULL	"lanes"=>"3","maxspeed"=>"60","oneway"=>"yes","surface"=>"paved"
8159668	S1	trunk	NULL	NULL	NULL	NULL	"lanes"=>"6","maxspeed"=>"60","oneway"=>"yes","surface"=>"paved"
479865097	S1	trunk	NULL	NULL	NULL	NULL	"lanes"=>"6","maxspeed"=>"60","oneway"=>"yes","surface"=>"paved"

**Figura 3.3:** Atributos dos elementos viários na camada “lines”

Na primeira coluna da tabela de atributos, está a identificação por código da base de dados *Open Street Map*, seguida, na segunda coluna, do nome da via. Da terceira até a quinta colunas há a caracterização do elemento linear representado, de acordo com seu tipo: *highway*, se rodovia; *waterway* se hidrovia; *aerialway*, se aerovia; *barrier*, se barreira física linear (cerca, corrimão, mureta); ou *man\_made*, se for um elemento linear feito pelo homem (gasoduto, oleoduto, barragem). Cada elemento só possuirá atributo referente a uma dessas colunas, aquela na qual ele se enquadrar. Na etapa b) serão apresentados os atributos específicos para cada tipo de elemento linear.

A sexta coluna da tabela, chamada de *other\_tags*, apresenta todos os outros atributos que não se encaixem nas anteriores. Percebe-se que para as rodovias, existem nessa coluna atributos como: número de faixas, velocidade máxima permitida e tipo de superfície. Entretanto, essas informações ficam agrupadas na mesma coluna, o que dificulta a análise dos dados. Além disso, a camada inclui elementos que não devem compor o arquivo de vias a ser inserido no MATSim, como ciclovias e estradas. Por esses motivos, a etapa a seguir consiste na organização do banco de dados.

#### b) Organização do banco de dados

Para organizar o banco de dados obtido do *Open Street Map*, foi feita a conversão da camada “lines” para o formato *Spatialite* e formato CSV (*Comma Separated Values*). *Spatialite* é uma biblioteca de código livre (*open source*) projetada para o banco de dados *SQLite* com o objetivo de lhe promover suporte a informações espaciais. É um tipo de banco de dados simples, que permite ao usuário visualizar atributos de dados geográficos em tabelas, sem a necessidade de instalação ou configuração (Furieri, 2017).

É importante frisar que sempre que for feita alguma exportação dos dados se utilize a conversão para coordenadas geográficas cartesianas, uma vez que o arquivo importado da base *OpenStreet Map* possui coordenadas esféricas (WGS84), o que dificulta a manipulação feita pelo *software* MATSim. Nesse estudo, optou-se por utilizar o sistema de coordenadas em zonas UTM, sendo a zona EPSG:32723 UTM 23S a adequada para se utilizar na área de estudo do DF e Entorno.

Com os dois arquivos gerados, utilizando o *software* Sublime Text, executou-se a rotina em linguagem *Python* apresentada no Apêndice E. O primeiro objetivo desse procedimento é separar os dados contidos na coluna *other\_tags* e organizá-los em colunas específicas, de uma forma que as informações possam ser acessadas mais facilmente no banco de dados. No código apresentado, essa etapa ocorre entre as linhas 58 e 68. Depois, filtra-se os elementos que são, de fato, vias rodoviárias, eliminando os demais. A porção do código responsável por isso, das linhas 123 a 127 do código apresentado, é uma *query* que exclui todos os elementos que não possuam atributo na coluna “*highway*”, pois isso significa que eles não são elementos rodoviários.

Em seguida, deve-se eliminar as vias de menor importância para não sobrecarregar desnecessariamente a base de dados. Isso foi feito baseado na classificação das rodovias fornecida pela base *OpenStreet Map*, com informações retiradas de sua própria documentação, que estão apresentadas no Anexo A. No presente estudo, optou-se por remover do banco todos os tipos de vias especiais (“*special road types*”, na classificação da base), todas as trilhas (“*paths*”), cicloviárias (“*cycleway*”) e vias propostas e em construção (“*proposed*” e “*construction*”, respectivamente). Dessa forma, foram mantidas apenas as vias com classificações principais do banco de dados, sendo elas “*motorway*”, “*trunk*”, “*primary*”, “*secondary*”, “*tertiary*”, “*unclassified*”, “*residential*” e “*service*”. No código apresentado no Apêndice E, essa rotina é programada entre as linhas 129 e 187.

O último passo na etapa de organização do banco de dados é o preenchimento dos dados faltantes por meio de hipóteses baseadas nos conceitos de engenharia de tráfego. A atribuição de velocidades máximas permitidas é feita de acordo com a classificação da via na base *OpenStreet Map* e sua classificação correspondente no Código Brasileiro de Trânsito, conforme informações apresentadas no Anexo A. Essa correspondência é apresentada na Tabela 3.1.

**Tabela 3.1:** Correspondência entre classificação de vias na base *OpenStreet Map* e no Código Brasileiro de Trânsito (Brasil, 1997)

Classificação na base OpenStreet Map	Classificação do Código Brasileiro de Trânsito	Velocidade máxima permitida
motorway e motorway_link trunk e trunk_link	Via de Trânsito Rápido	80 km/h
primary e primary_link secondary e secondary_link	Via Arterial	60 km/h
tertiary e tertiary_link	Via Coletora	40 km/h
unclassified residential service	Via Local	30 km/h

A rotina que executa a atribuição de velocidades às vias do banco que não a possuíam está entre as linhas 189 e 220 do código apresentado no Apêndice E.

Outro dado faltante que deve ser preenchido é a direção de cada via do banco de dados. Sabe-se que para as categorias “motorway” e “trunk”, quando a via é de sentido duplo, a própria base de dados *OpenStreet Map* cria duas instâncias separadas, cada uma com um sentido e, portanto, ambas com atributo “*oneway = yes*”. Por isso, quando a via não possui atributo de “*oneway*” especificado, é razoável assumir que se trata de uma via de menor importância, de mão dupla.

A rotina que executa a atribuição de sentido às vias do banco que não a possuíam está entre as linhas 222 e 225 do código apresentado no Apêndice E.

Por fim, o último dado relevante que deve ser atribuído a todas as instâncias do banco de dados é o número de faixas de cada via. A hipótese feita é a de que, para os elementos que não possuem atributo de número de faixas, se deva atribuir 1 faixa para cada sentido de fluxo da via. Portanto, as vias com atributo “*oneway*” = “*yes*” recebem o número “1” na coluna “*lanes*” e as com atributo “*oneway*” = “*no*” recebem o número “2” (uma faixa para cada sentido).

A rotina que executa a atribuição de número de faixas às vias do banco que não o possuíam está entre as linhas 227 e 236 do código apresentado no Apêndice E. Conclui-se, assim, a etapa de organização do banco de dados.

c) Criação de arquivo Shapefile

A partir do banco de dados organizado resultante da etapa anterior, utilizando o *software* Quantum GIS, gera-se o arquivo Shapefile da área de estudo. Mais uma vez, é importante frisar a necessidade da escolha da projeção geográfica correta durante exportação do arquivo no *software*.

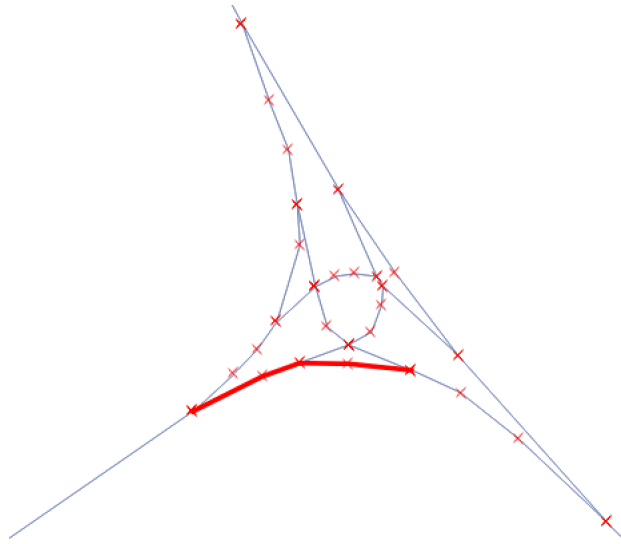
O arquivo Shapefile gerado agora pode servir de banco de dados colaborativo para todos os interessados no planejamento da área de estudo, sendo constantemente atualizado e conferido pelos usuários. Para ser convertido ao formato de leitura do *software* MATSim, no entanto, os dados devem passar pelo tratamento descrito na etapa a seguir.

d) Conversão do arquivo Shapefile no dado de entrada *network.xml* para MATSim

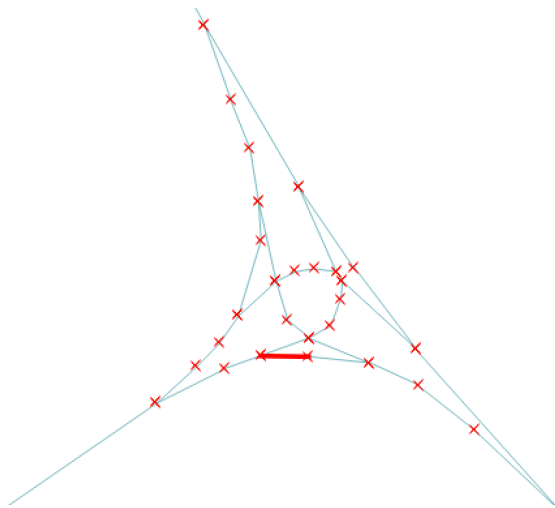
Foi desenvolvida uma rotina em linguagem de programação Python para converter os dados geográficos em formato Shapefile, obtidos na etapa anterior, em arquivo XML compatível com o *software* MATSim. No entanto, as bibliotecas utilizadas na programação manipulam elementos lineares segmentados, com apenas dois nós e uma ligação orientada entre eles (grafos). O Shapefile em questão possui elementos de multissegmentos, ou seja, cada via que constitui o banco de dados é formada por um número de segmentos. Isso pode ser verificado no exemplo da Figura 3.4. Quando se seleciona um dos arcos da interseção representada, um elemento do tipo “*primary\_link*” do banco de dados, verifica-se que tal elemento é formado por um conjunto de 4 segmentos de linha.

Para corrigir esse problema, devem-se transformar os elementos que constituem o Shapefile em grafos, utilizando, para tanto, o *plugin Networks*, para *software* Quantum GIS. Ao realizar esse procedimento, o número de elementos viários no banco de dados passa de 64.433 para 271.151.

Na Figura 3.5 é possível notar que após o procedimento de conversão em grafos, cada segmento que antes formava uma única via agora pode ser selecionado individualmente.



**Figura 3.4:** Elemento do tipo “*primary\_link*” formado por 4 segmentos de linha.



**Figura 3.5:** Segmentos de linha individualizados após procedimento de conversão em grafos.

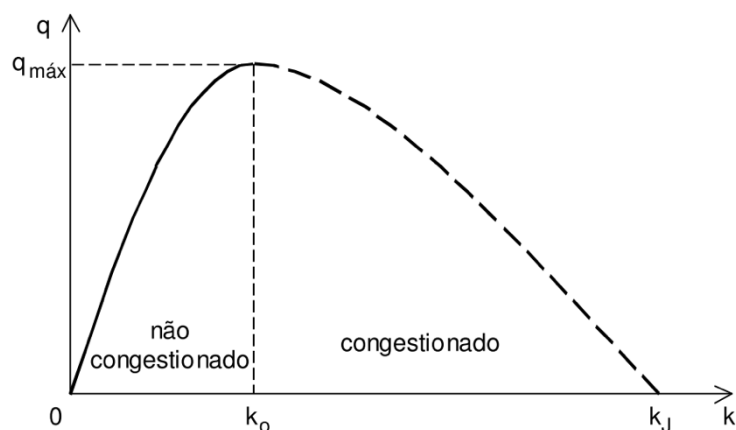
Com o Shapefile formado apenas por grafos, é possível executar a rotina apresentada no Apêndice F para converter o arquivo em formato legível pelo MATSim. Essa rotina utiliza a biblioteca *NetworkX* para Python, que permite analisar grafos, como os que formam o Shapefile das vias. As principais definições contidas nesse código são: a forma como se gerenciam as vias de mão dupla; o método de cálculo do comprimento das vias e; o método de cálculo da capacidade das vias.

Como os grafos são unidirecionais, as vias que possuem atributo “*oneway*” = “*no*” no Shapefile, ou seja, as que são de mão dupla, devem ser processadas duas vezes pelo código. Isso é feito criando dois elementos com mesmos nós em suas extremidades, mas orientados em sentidos contrários. Isso é mostrado no código do Apêndice F no condicional a partir da linha 125. Graficamente, os trechos viários de mão dupla seriam apresentados como duas linhas, uma sobre a outra.

O comprimento de cada “*link*”, ou trecho viário, é calculado pela posição dos nós de suas extremidades. Como o sistema de coordenadas utilizado foi o UTM, as coordenadas dos nós são dadas em metros, o que permite que o comprimento da linha que os une seja calculada por uma equação simples de Pitágoras, pela raiz da diferença entre as coordenadas do eixo X ao quadrado, somada à diferença das coordenadas do eixo Y ao quadrado. Esse cálculo é apresentado na linha 70 do código do Apêndice F.

Por fim, é necessário atribuir a cada via do arquivo de entrada no MATSim uma capacidade de fluxo veicular. Capacidade é a máxima taxa horária de fluxo de tráfego que pode ser esperada numa seção da via, por sentido (ou nos 2 sentidos para o caso de vias de sentidos opostos), durante um dado período de tempo (normalmente 1 hora) (Goldner, 2014). O método clássico de cálculo da capacidade viária é dado pelo *Highway Capacity Manual* (HCM), mas demanda um grande número de variáveis sobre a via, como greide e taxa de veículos pesados que nela trafegam.

Nesse trabalho, portanto, foi desenvolvido um método simplificado para cálculo da capacidade das vias em função de sua classificação (expressa, arterial, coletora ou local) e do número de faixas que compõem a via. Para tanto, partiu-se do modelo de Greenberg (Silva, 2007), que relaciona fluxo de tráfego ( $q$ ) e concentração de veículos ( $k$ ) em um trecho viário, conforme diagrama da Figura 3.6.



**Figura 3.6:** Modelo de Greenberg para relação entre fluxo de tráfego ( $q$ ) e concentração de veículos ( $k$ ) (Silva, 2007).

Desse modelo, o máximo fluxo de tráfego  $q_{máx}$ , que define a capacidade da via, é calculado pela expressão da Eq. 3.1:

$$q_{máx} = v_0 \frac{k_j}{e},$$

**Eq. 3.1**

em que:

- $v_0$  é a velocidade ótima, correspondente ao ponto que se alcança  $q_{máx}$ , e dada pela razão entre a velocidade de fluxo livre da via e a constante  $e$ ;
- $k_j$  é a concentração máxima, correspondente à situação de completo congestionamento;
- $e$  é a constante de Euler, com valor aproximado de 2,7183.

Considerou-se que na situação de completo congestionamento, os veículos permanecem parados a uma distância de 60cm entre cada um. Considerando um comprimento de projeto de 580cm para cada veículo e somando à distância assumida entre eles, obtém-se uma concentração  $k_j = 156$  veículos por quilômetro de via, por faixa.

A velocidade de fluxo livre para classificação viária, utilizada para o cálculo da velocidade ótima  $v_0$ , foi obtida da tabela de velocidades de fluxo livre padrões dada pelo HCM (Transportation Research Board, 2000). A Tabela 3.2 mostra o procedimento de cálculo das capacidades de cada tipo de via de acordo com os parâmetros mencionados.



**Tabela 3.2:** Cálculo da capacidade de acordo com a classificação das vias

Classificação da Via	Classificação Equivalente no HCM (TRB, 2000)	Velocidade padrão de fluxo livre: $v_f$ (TRB, 2000)	Velocidade ótima $v_0 = v_f/e$	$q_{máx} = (v_0)(k_j)/e$
Trânsito rápido	I	90 km/h	33 km/h	1893 veíc/h
Arterial	II	70 km/h	26 km/h	1492 veíc/h
Coletora	III	55 km/h	20 km/h	1147 veíc/h
Local	IV	55 km/h	20 km/h	1147 veíc/h

No código, os valores adotados para capacidade de vias de trânsito rápido, arterial e coletora ou local, foram, respectivamente, 1900, 1500 e 1200 veículos por hora, por faixa de rolamento.

Assim, executando-se a rotina apresentada no Apêndice F, tem-se o arquivo *network.xml* pronto para ser usado como dado de entrada no MATSim.

### 3.3 População sintética (*population.xml*)

#### 3.3.1 Softwares e arquivos necessários

Para a execução da metodologia desenvolvida para gerar a população sintética como dado de entrada no MATSim é necessário ter previamente os arquivos:

- Arquivo com categorias de pontos do *OpenStreet Map* (Apêndice G);
- Shapefile com limites das regiões estudadas (disponível no *website* do IBGE: [ftp://geoftp.ibge.gov.br/organizacao\\_do\\_territorio/malhas\\_territoriais/malhas\\_municipais/municipio\\_2015/UFs/](ftp://geoftp.ibge.gov.br/organizacao_do_territorio/malhas_territoriais/malhas_municipais/municipio_2015/UFs/))

Além dos arquivos mencionados, os seguintes *softwares open-source* são necessários para se gerar a população sintética:

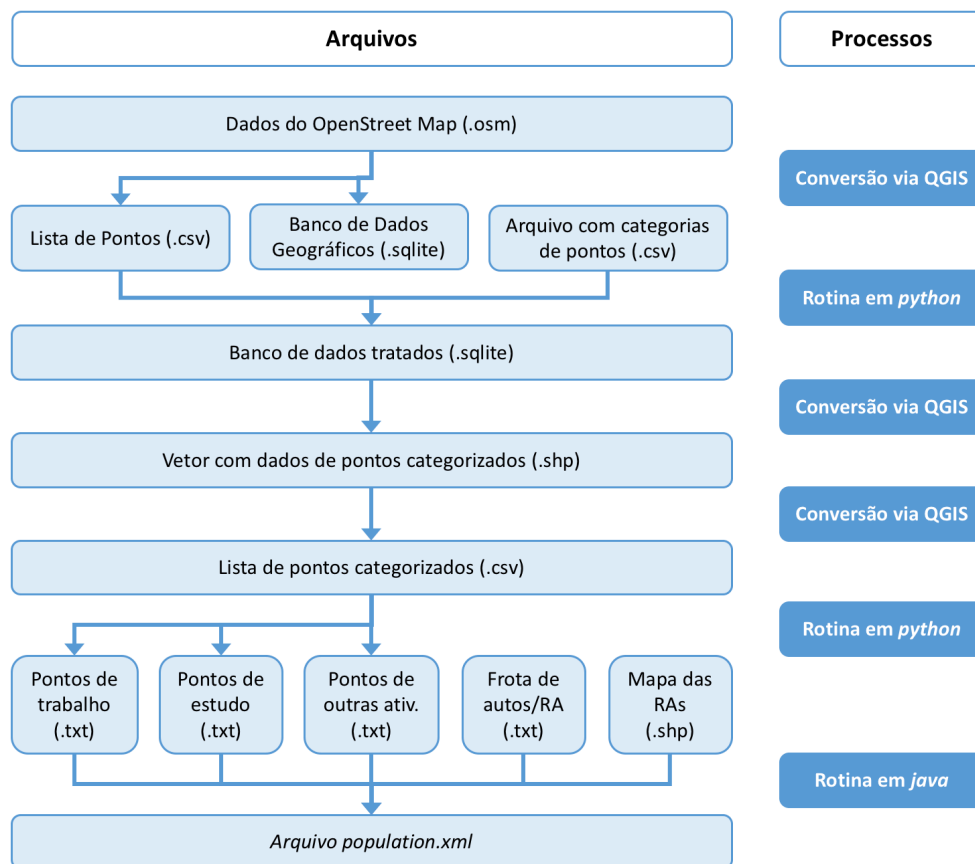
- Quantum GIS com *plugin Network*;
- *Sublime Text*;
- Plataforma *Java Eclipse*.

### 3.3.2 Método para geração de população sintética

Para dar início ao ciclo iterativo da simulação, o MATSim precisa de um arquivo de população sintética representativa da área de estudo. Por ser um *software* de simulação de modelo baseado em atividades, também é necessário que se estabeleça quais são os possíveis planos de atividades desenvolvidas pelos indivíduos modelados. Esses parâmetros são definidos nessa etapa do tratamento de dados.

Em sua simulação, Bicudo (2015) optou por criar a população sintética a partir de dados censitários referentes à idade dos moradores de cada subregião estudada. De acordo com as faixas etárias, era definido se o indivíduo trabalhava ou estudava, e então lhe era atribuída uma subregião de trabalho ou estudo. Os planos de atividades eram gerados de forma aleatória, podendo incluir atividades diferentes de trabalho ou estudo (categoria “outro”).

Neste trabalho, desenvolveu-se uma estrutura metodológica alternativa para geração do arquivo de população sintética (*population.xml*), que é ilustrada na Figura 3.7 e detalhada no texto em seguida.



**Figura 3.7:** Esquema da metodologia proposta para geração e tratamento de dados de população sintética

a) Importação dos pontos do *OpenStreet Map*

Como mencionado na etapa de tratamento de dados viários, importando o vetor geográfico da base *OpenStreet Map* para o *software* QGIS, obtém-se um arquivo de três camadas: polígonos, linhas, multilinhas e pontos. Para o tratamento dos dados viários foi utilizada a camada de linhas. Nessa nova etapa do método, utilizou-se a camada de pontos para identificar locais onde os indivíduos da população sintética realizariam determinadas atividades.

Na base de dados do *OpenStreet Map*, um ponto de interesse é um elemento do mapa que ocupa um ponto de coordenadas particulares, diferentemente de elementos lineares, como as vias, ou polígonos, como os tipos de ocupação do solo (OpenStreet Map Wiki, 2016). Um ponto, no entanto, não necessariamente é um local de realização de atividades. Na base de dados estão incluídas, por exemplo, caixas de correio, bebedouros, torres de energia, entre outros elementos de mobiliário urbano. Uma representação dos atributos associados aos elementos do tipo “ponto” ao serem visualizados no *software* QGIS é apresentada na Figura 3.8.

osm_id	name	barrier	highway	ref	address	is_in	place	man_made	other_tags
4726189...	Veterinári...	NULL	NULL	NULL	NULL	NULL	NULL	NULL	"amenity"=>"veterinary"
4726189...	Veterinario	NULL	NULL	NULL	NULL	NULL	NULL	NULL	"amenity"=>"veterinary"
265840384	Valparaís...	NULL	NULL	NULL	NULL	NULL	city	NULL	"is_in:continent"=>"South America","is_i...
4726189...	Tesoura D...	NULL	NULL	NULL	NULL	NULL	NULL	NULL	"shop"=>"fashion"
4726189...	SuperBom	NULL	NULL	NULL	NULL	NULL	NULL	NULL	"shop"=>"supermarket"
251810547	Shopping	NULL	NULL	NULL	NULL	NULL	NULL	NULL	"public_transport"=>"stop_position","rail...
4726189...	Santander	NULL	NULL	NULL	NULL	NULL	NULL	NULL	"amenity"=>"bank","atm"=>"yes"
4726189...	Restauran...	NULL	NULL	NULL	NULL	NULL	NULL	NULL	"amenity"=>"restaurant"

**Figura 3.8:** Atributos dos elementos pontuais na camada “*lines*”

As duas primeiras colunas da tabela de atributos representam a identificação do ponto, pelo código da base de dados (coluna 1) e pelo seu nome identificador (coluna 2). As 7 colunas seguintes (3 a 9) apresentam as características principais de cada ponto. Cada elemento pode ter uma ou mais das características seguintes:

- *barrier* (barreira): são elementos caracterizados como obstáculos ao deslocamento. Exemplos: cerca (curta, considerada pontual), portão;
- *highway* (vias): são elementos pontuais associados ao tráfego. Exemplos: semáforo, barreira eletrônica;

- *ref* (referência): é o código que caracteriza o elemento. Exemplo: para um ponto que represente um aeroporto, o atributo “Ref” pode apresentar seu código internacional;
- *address* (endereço): é a localidade postal de um ponto. Exemplos: quadra, rua.
- *is\_in* (está contido em): é um meio de inserir hierarquias de limites territoriais. Exemplo: se o ponto representa o centroide de uma cidade, é possível inserir em sua coluna *is\_in* o estado do qual ela faz parte;
- *place* (local): essa coluna é preenchida quando o ponto representa o centroide de um limite territorial político. Exemplos: cidades, estados, distritos, países.
- *man\_made* (construído pelo homem): são atributos de pontos que representam estruturas artificiais adicionadas à paisagem. Exemplos: torre de comunicação, silo, moinho.

A décima e última coluna da tabela de atributos, intitulada “*other\_tags*” traz características secundárias para pontos que já foram categorizados nas colunas de 3 a 7, ou pode trazer características diversas para pontos que não possuem nenhum dos atributos descritos. Após análise da documentação do *OpenStreet Map*, constatou-se que os locais típicos de realização de atividades (escolas, escritórios, restaurantes) têm seus atributos principais na coluna “*other\_tags*”. Portanto, foi necessário criar uma rotina para filtrar e organizar o banco de dados, deixando apenas os pontos e atributos interessantes à aplicação de um modelo de atividades. Essa rotina é detalhada no próximo tópico.

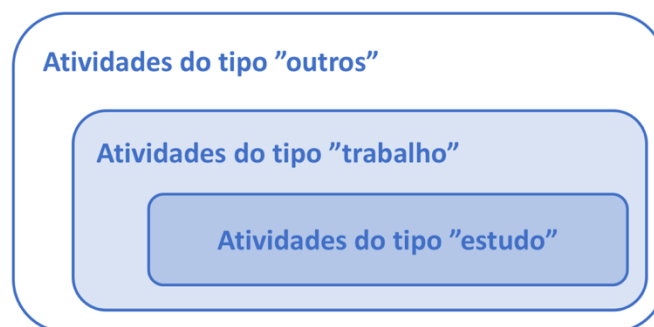
#### b) Organização da base de dados

Para organizar o banco de dados obtido do *OpenStreet Map*, foi feita a conversão da camada “*points*” para os formatos *Spatialite* e CSV (*Comma Separated Values*) utilizando o *software* QGIS, procedimento similar ao que foi realizado no tratamento de dados viários. É importante frisar que se utilizem as coordenadas geográficas corretas sempre que for feita alguma conversão de dados: no caso da área de estudo, que é o DF, foi utilizada sempre a zona EPSG: 32723 UTM 23S.

Com a análise da documentação da base *OpenStreet Map*, constatou-se que os atributos apresentados na coluna “*other\_tags*” podem ser divididos em 27 categorias, dentre elas, “*amenity*” (instalações com funções específicas), “*shop*” (lojas), “*leisure*” (lazer). As 27

categorias de atributos são ainda divididas em 473 subcategorias. A categoria “*leisure*”, por exemplo, inclui as subcategorias “*fishing*” (pescaria), “*golf\_course*” (campo de golfe), “*park*” (parque), entre outros.

A etapa de tratamento dos dados do banco de dados consistiu, então, na identificação dos pontos que seriam locais de realização de atividades de trabalho, estudo e “outros”. As atividades de trabalho incluem todo tipo de comércio, escritórios e alguns pontos que também são de lazer e estudo. As atividades de estudo foram consideradas apenas para o público adulto, já que a simulação a ser feita incluiria apenas usuários de automóvel (motoristas). Atividades do tipo “outros” incluem os locais de compras, lazer, refeição, negócios e saúde, por exemplo. Dessa forma, é fácil perceber que alguns pontos recebem mais de uma classificação. Uma universidade, por exemplo, pode ser um ponto de estudo para um indivíduo, mas também pode ser o local de trabalho para outro, da mesma forma que pode ser uma atividade secundária, um local do tipo “outros” para o motorista que apenas passa na faculdade para levar ou deixar alguém no caminho de sua atividade principal. A Figura 3.9 representa a forma como as categorias de atividades se sobrepõem e o Apêndice G lista todas as subcategorias dos atributos dos pontos e suas respectivas classificações por tipo de atividade realizada.



**Figura 3.9:** Hierarquia das categorias de atividades para cada ponto.

Após a identificação das categorias que representavam locais de atividade do tipo “trabalho”, cada ponto recebeu um código representativo do número de postos de trabalho que ele integra. Esses códigos podiam ser: “P” (pequeno, aproximadamente 5 postos de trabalho; MP (médio-pequeno, aproximadamente 10 postos de trabalho); M (médio, aproximadamente 25 postos de trabalho); MG (médio-grande, aproximadamente 50 postos de trabalho) ou; G (grande, aproximadamente 100 postos de trabalho). Um ponto da subcategoria “*carwash*” (lava a jato), por exemplo, é um estabelecimento de pequeno porte, e, por isso, foi classificado

com código P. Já um aeroporto foi classificado como G devido ao grande número de postos de trabalho associados a esse tipo de estabelecimento. A lista completa com classificações por categoria e subcategoria é apresentada no Apêndice G.

De posse dos dois arquivos extraídos do vetor de pontos do *OpenStreet Map* e com o arquivo contendo a tabela do Apêndice G, executou-se a rotina em Python apresentada no Apêndice H a fim de organizar o banco de dados com todos os critérios já mencionados e, adicionalmente, traduzindo as categorias originais da base de dados para a língua portuguesa. A Figura 3.10 apresenta o banco de dados resultante do processo de organização.

osm_id	name	descricao	estudo	outros	trabalho	tamtrabalho
Filter	Filter	Filter	Filter	Filter	Filter	Filter
506285265	Pontão do Lago Sul	Lazer	N	S	S	P
506288365	Carrefour Bairro	Comercio	N	S	S	MP
506288372	Carrefour Bairro	Comercio	N	S	S	MP
560224132	Clube de Tênis Resort	Hotel	N	N	S	M
599435122	Supermercado Veneza	Comercio	N	S	S	MP
599435489	Supermercado Serve Mais	Comercio	N	S	S	MP

**Figura 3.10:** Banco de dados tratado com os elementos pontuais.

c) Criação de arquivo Shapefile

Com o banco de dados tratados, é possível converter o arquivo *Spatialite* para Shapefile utilizando o *software* QGIS. Essa etapa é importante porque, no contexto do planejamento governamental, o vetor Shapefile pode ser constantemente atualizado e expandido. As etapas a seguir, portanto, descrevem o procedimento de utilização do banco de dados de pontos obtido para gerar o arquivo *population.xml* como dado de entrada do MATSim.

d) Exportação de pontos por categoria

Para gerar o arquivo *population.xml* como dado de entrada do MATSim, é necessário ter listas de pontos para cada tipo de atividade a ser realizada. Partindo do Shapefile gerado na etapa anterior, é possível obter um arquivo CSV, similar ao procedimento realizado anteriormente com os dados do *OpenStreet Map*. Utilizando a rotina apresentada no Apêndice

I, obtiveram-se do arquivo CSV três arquivos em formato de texto (.txt): um para os pontos de realização de atividade “trabalho”, um para “estudo” e outro para “outros” tipos de atividade. Cada arquivo é composto pelas colunas que incluem coordenadas X e Y do ponto no sistema de coordenadas definido anteriormente (EPSG: 32723 UTM 23S) e pela coluna que contém o código de identificação do ponto. Estes arquivos são utilizados nas próximas etapas como acessórios à rotina em Java para criação do arquivo *population.xml*.

e) Elaboração dos planos de atividades

Como mencionado no tópico 2.7.4, o modelo integrado ao *software* MATSim não é capaz de gerar padrões de atividades, essa é uma informação que deve ser inserida previamente no *software*. Portanto, com base em discussões e resultados observados em estudos anteriores (Taco, 2003; Arruda, 2005; Pitombo, 2007; Takano, 2010; Medrano, 2012), foram definidas três categorias de indivíduo de acordo com as atividades que realizam: pessoas que só trabalham, que estudam e trabalham, que apenas estudam e que não têm atividade. Os possíveis padrões de atividades realizados por cada grupo de pessoas são apresentados na Tabela 3.3.

**Tabela 3.3:** Grupos de indivíduos e padrões de atividades

Grupos de Indivíduos	Padrões de Atividades		
Só trabalham	a	H-W-H	casa – trabalho – casa
	b	H-O-W-H	casa – outro – trabalho – casa
	c	H-W-O-H	casa – trabalho – outro – casa
	d	H-W-O-W-H	casa – trabalho – outro – trabalho – casa
Só estudam	e	H-S-H	casa – estudo – casa
	f	H-O-S-H	casa – outro – estudo – casa
	g	H-S-O-H	casa – estudo – outro – casa
	h	H-S-O-S-H	casa – estudo – outro – estudo – casa
Trabalham e estudam	i	H-S-W-H	casa – estudo – trabalho – casa
	j	H-W-S-H	casa – trabalho – estudo – casa
	k	H-S-O-W-H	casa – estudo – outro – trabalho – casa
	l	H-W-O-S-H	casa – trabalho – outro – estudo – casa
Sem atividade	m	H-O-H	casa – outro – casa
	n	H	casa
H = “house” (casa); W = “work” (trabalho); O = “other” (outro).			

Para definir quantos indivíduos da população sintética pertenceriam a cada grupo, foram utilizados os microdados da Pesquisa Distrital por Amostra de Domicílios do Distrito Federal (CODEPLAN, 2012), mas para os municípios brasileiros esse dado está presente nos relatórios do Censo do IBGE. Os resultados observados para o DF foram discretizados por região administrativa.

Dentro de cada grupo, o indivíduo pode optar por quaisquer dos padrões de atividade disponíveis. No momento da simulação, o MATSim sorteia um plano para cada indivíduo e a cada iteração o algoritmo permite que alguns dos indivíduos do modelo mudem de plano de atividade, até que se obtenha o padrão mais provável. Portanto, para a inserção no modelo, seria aceitável, nos termos da execução do modelo, apenas distribuir igualmente dentro de cada grupo de indivíduos os padrões de atividades realizados (Exemplo: dentro do grupo de indivíduos que só trabalham, determinar que 25% deles tenham o padrão “a”, 25% o padrão “b”, e assim sucessivamente). Entretanto, para convergir à situação mais provável, o *software* dependeria de um número de iterações muito maior do que na situação que se conhecem as distribuições de probabilidade.

Portanto, para aumentar a velocidade de convergência da simulação ao cenário ótimo, optou-se por estimar distribuições mais prováveis para os padrões de atividades. Isso foi feito utilizando os dados da pesquisa origem-destino realizada para a área de estudo no ano de 2009 (Secretaria de Transportes, 2011). De posse dos dados sobre a proporção de viagens diárias para cada motivo de viagem, utilizando o modo motorizado individual, foi feita uma compatibilização desse número de viagens realizadas e o número de pessoas em cada categoria. Utilizando o *plugin Solver* para *Microsoft Excel*, o resultado obtido está representado nas Tabelas 3.4 e 3.5.

Com a definição das proporções dos padrões de atividade verificados e com os arquivos de texto com as coordenadas de cada ponto de realização de atividades, é possível executar a rotina para gerar a população sintética, que será descrita no próximo tópico.



**Tabelas 3.4 e 3.5:** Distribuição dos padrões de atividades

GRUPO	% REAL	% MODELADO	DIFERENÇA	PADRÕES DE VIAGEM					
SÓ TRABALHO	52,00%	52,00%	0,00%	H-W-H	35,45%	home-work	work-home		
				H-O-W-H	5,30%	home-other	other-work	work-home	
				H-W-O-H	5,30%	home-work	work-other	other-home	
				H-W-O-W-H	5,96%	home-work	work-other	other-work	work-home
SÓ ESTUDO	6,10%	7,81%	1,71%	H-S-H	2,85%	home-study	study-home		
				H-O-S-H	1,00%	home-other	other-study	study-home	
				H-S-O-H	1,00%	home-study	study-other	other-home	
				H-S-O-S-H	2,97%	home-study	study-other	other-study	study-home
TRABALHO E ESTUDO	6,80%	11,73%	4,93%	H-S-W-H	3,86%	home-study	study-work	work-home	
				H-W-S-H	3,86%	home-work	work-study	study-home	
				H-S-O-W-H	2,01%	home-study	study-other	other-work	work-home
				H-W-O-S-H	2,01%	home-work	work-other	other-study	study-home
SEM ATIVIDADE	35,00%	28,46%	3,36%	H-O-H	12,44%	home-other	other-home		
				H	16,02%	home-home			
<b>SOMA DOS ERROS</b>			<b>10,00%</b>						

NÚMERO DE VIAGENS				
TIPO	MODELADO	NORMALIZADO	REAL	DIFERENÇA
Trabalho	69,69%	53,53%	53,41%	0,12%
Estudo	22,51%	17,30%	25,03%	7,73%
Outro	37,97%	29,17%	21,56%	7,61%
<b>SOMA</b>	<b>130,17%</b>	<b>100,00%</b>	<b>100,00%</b>	<b>15,47%</b>

f) Geração do arquivo *population.xml*

A geração do arquivo *population.xml* demanda o uso de algumas funções nativas do MATSim, portanto, o script utilizado nessa etapa é feito em linguagem Java. O código desse script, na íntegra, é apresentado no Apêndice J, e seu funcionamento é descrito a seguir.

Na primeira etapa do código (linhas 39 a 48) são informados ao programa os caminhos aos diretórios dos arquivos necessários à sua execução. Esses arquivos foram obtidos nas etapas anteriores dessa metodologia, e incluem: o arquivo XML de vias da região de estudo, *network.xml*; o arquivo Shapefile com os limites das zonas de estudo, nesse caso, as regiões administrativas do DF; os três arquivos de texto com as coordenadas dos pontos de acordo com cada tipo de atividade (estudo, trabalho e outros).

O único arquivo de entrada necessário para a rotina que ainda não foi descrito nessa metodologia é um arquivo de texto contendo a frota de cada região de estudo. Na pesquisa atual, utilizaram-se os microdados da Pesquisa Distrital por Amostra de Domicílios do Distrito Federal (CODEPLAN, 2012) para obter tal informação, mas para os municípios brasileiros esse dado está presente nos relatórios do Censo do IBGE. O arquivo de texto é

simples, contendo apenas 2 colunas: uma com o código da subregião de análise e a outra com a frota veicular dessa área.

As duas últimas informações fornecidas ao programa são o diretório de saída, onde deve ser salvo o arquivo *population.xml* gerado, e o fator de escala a ser usado na geração da população sintética. O fator de escala é importante porque, ao utilizar o MATSim, geralmente não é simulada a totalidade da população da área de estudo, por conta das limitações de tempo e também da capacidade de processamento de *hardware* disponível. Assim, é sugerido no próprio manual do MATSim que seja utilizado um fator de escala, por exemplo, de 10%, que afeta não só a população sintética mas, similarmente, a capacidade das vias adotada durante a simulação (Horni et al., 2016).

A função principal (*main*) é iniciada na linha 66 do *script*. Primeiramente, o programa lê o arquivo Shapefile com os limites geográficos das regiões analisadas. No presente estudo de caso, o Shapefile utilizado era formado por polígonos das regiões administrativas do DF, cujo atributo principal era o número oficial da RA, e mais 8 polígonos com regiões do Entorno do DF, numeradas sequencialmente a partir do último código de RA. A Figura 3.11 apresenta as regiões fora do DF que foram incluídas na área de estudo.



**Figura 3.11:** Regiões do Entorno do DF incluídas na área de estudo

Em seguida, o programa lê os arquivos de texto contendo as informações sobre os pontos de realização das atividades e a frota de cada RA, utilizando funções definidas no próprio *script* para transformar os arquivos de texto em matrizes dentro do programa.

O programa então, faz uma iteração ao longo das subregiões de estudo, realizando um *loop* a partir da área identificada pelo número 1, até a 39, incluindo as 30 regiões administrativas do DF existentes em 2008 e as 8 áreas do Entorno selecionadas. A cada iteração, é lida da matriz de frota a quantidade de automóveis da região. Para definir quantos veículos o simulador deve incluir, a frota de cada região é multiplicada primeiramente pelo fator de escala adotado e em seguida pela proporção dos carros que saem diariamente às ruas.

Para definir a proporção de carros que saem das garagens diariamente, foram utilizados os microdados da Pesquisa Distrital por Amostra de Domicílios do Distrito Federal (CODEPLAN, 2012), cruzando os dados dos domicílios que possuíam pelo menos 1 veículo e os domicílios em que pelo menos 1 pessoa utilizava o automóvel para se deslocar até o trabalho. Verificou-se que em 45% dos domicílios que possuíam pelo menos 1 veículo, nenhum dos moradores utilizava esse modo para se deslocar até o trabalho. Portanto, a proporção da frota que sai às ruas diariamente foi adotada como 55%.

Definido o número de veículos a ser simulado para cada região analisada, o programa, então, adota as proporções calculadas no item e) para gerar indivíduos com planos de atividades. No presente estudo de caso, para cada região foram modelados 52% dos indivíduos com atividade principal “trabalho”, 7,81% “estudo”, 11,73% “trabalho e estudo” e 28,46% “sem atividade”. Para cada categoria de indivíduo, foram alocados todos os planos possíveis na memória do agente, mas para definir o plano ativo foram utilizadas as proporções também calculadas no item e). Dos indivíduos que só trabalhavam, por exemplo, 35,45% tinham como plano ativo H-W-H, 5,30% H-O-W-H, 5,30% H-W-O-H, 5,96% H-W-O-W-H. A Tabela 3.6 apresenta a quantidade de indivíduos simulados em cada RA no estudo de caso, utilizando um fator de escala de 10% e as demais proporções citadas.

Então, para cada indivíduo simulado, é sorteado um ponto com coordenadas X e Y dentro do polígono da sua região de origem. Em seguida, das matrizes com pontos de realização de atividades, são sorteados um ponto de trabalho (caso o indivíduo tenha essa atividade), um ponto de estudo (caso o indivíduo tenha essa atividade) e um ponto do tipo “outros”.

Os planos são montados de forma que o trabalho tenha início e fim fixos compatíveis com o horário de funcionamento do estabelecimento sorteado. Caso o indivíduo trabalhe em um lugar que funciona por mais de 10 horas, por exemplo, ele será alocado em um turno de trabalho com no máximo 8 horas de duração, para ser compatível com a realidade. Cada pessoa do modelo também recebe o parâmetro de sair de casa 1 hora antes do início de sua primeira atividade, e de realizar as atividades subsequentes com 1 hora de intervalo entre elas, à exceção de trabalho e estudo que têm horários fixos. Essa atribuição serve apenas como parâmetro inicial dos planos adotados por cada agente, pois, caso o indivíduo chegue atrasado ou muito adiantado para sua atividade, o plano receberá um *score* baixo e o MATSim fará a adaptação adequada de horários de saída para otimizar o plano, o que será descrito nas próximas seções.

O arquivo com os dados dos indivíduos gerados para todas as regiões administrativas, e seus respectivos planos, selecionados ou não, é gravado em formato XML e está pronto para ser utilizado como dado de entrada no MATSim.

### **3.4 Arquivo de configuração (*config.xml*)**

#### **3.4.1 Softwares e arquivos necessários**

Para gerar o arquivo de configuração do MATSim, é necessário adaptar o arquivo de configuração padrão da versão do *software* utilizada. Para obter tal arquivo, deve-se executar o seguinte código no terminal de comando do computador: “*java -cp matsim.jar org.matsim.run.CreateFullConfig fullConfig.xml*”. O endereço do aplicativo *matsim.jar* deve ser adaptado de acordo com o local onde o a pasta do MATSim tenha sido salva.

Além do arquivo mencionado, é necessário um *software* editor de texto. Sugere-se utilizar o programa *Sublime Text*, já utilizado nas etapas anteriores do método.

**Tabela 3.6:** Total de agentes simulados no estudo de caso

Área de estudo		Frota Total	Frota na rua (55%)	Fator de Escala (10%)	TOTAIS SIMULADOS			
					Só trabalham (52%)	Só estudam (7,81%)	Estudam e trabalham (11,73%)	Sem atividade (26,46%)
1	Brasília	138894	76392	7639	3972	597	896	2021
2	Gama	32329	17781	1778	925	139	209	470
3	Taguatinga	66736	36705	3670	1909	287	431	971
4	Brazlândia	5535	3044	304	158	24	36	81
5	Sobradinho	23310	12821	1282	667	100	150	339
6	Planaltina	21907	12049	1205	627	94	141	319
7	Paranoá	3485	1917	192	100	15	22	51
8	Núcleo Bandeirante	7798	4289	429	223	33	50	113
9	Ceilândia	42888	23588	2359	1227	184	277	624
10	Guará	56199	30909	3091	1607	241	363	818
11	Cruzeiro	12213	6717	672	349	52	79	178
12	Samambaia	17719	9745	975	507	76	114	258
13	Santa Maria	20771	11424	1142	594	89	134	302
14	São Sebastião	16658	9162	916	476	72	107	242
15	Recanto das Emas	10010	5506	551	286	43	65	146
16	Lago Sul	24170	13294	1329	691	104	156	352
17	Riacho fundo	7085	3897	390	203	30	46	103
18	Lago Norte	15539	8546	855	444	67	100	226
19	Candangolândia	3979	2188	219	114	17	26	58
20	Águas Claras	36613	20137	2014	1047	157	236	533
21	Riacho Fundo II	6399	3519	352	183	27	41	93
22	Sudoeste/Octogonal	27059	14882	1488	774	116	175	394
23	Varjão	477	262	26	14	2	3	7
24	Park Way	7162	3939	394	205	31	46	104
25	SCIA	1817	999	100	52	8	12	26
26	Sobradinho II	19182	10550	1055	549	82	124	279
27	Jardim Botânico	10736	5905	590	307	46	69	156
28	Itapoã	11467	6307	631	328	49	74	167
29	SIA	388	213	21	11	2	3	6
30	Vicente Pires	13714	7543	754	392	59	88	200
31	Águas Lindas	13365	7351	735	382	57	86	195
32	Cidade Ocidental	5864	3225	323	168	25	38	85
33	Formosa	18224	10023	1002	521	78	118	265
34	Luziânia	25123	13818	1382	719	108	162	366
35	Novo Gama	9901	5446	545	283	43	64	144
36	Planaltina	8816	4849	485	252	38	57	128
37	Santo Antônio do Descoberto	6572	3615	361	188	28	42	96
38	Valparaíso	22785	12532	1253	652	98	147	332
<b>TOTAIS SIMULADOS</b>					22105	3320	4986	11248
					<b>41659</b>			

### 3.4.2 Método desenvolvido para criação do arquivo de configuração

A adaptação das configurações originais consiste na simples alteração dos parâmetros de cada módulo do arquivo de texto XML. Nesse estudo de caso, optou-se por modificar apenas 7 dos 24 módulos que compõem o arquivo de configuração completa do MATSim 0.8.1. No arquivo *config.xml* utilizado como dado de entrada no *software*, só é preciso registrar os módulos que sofreram alteração. O MATSim assume os parâmetros padrão para os módulos que não são registrados no arquivo *config.xml*. Os tópicos a seguir detalham as configurações adotadas para a simulação deste estudo de caso, e o arquivo completo *config.xml* adotado é apresentado no Apêndice K.

#### a) Módulo “Controler”

O principal parâmetro definido nesse módulo é o número de iterações feitas na simulação, nesse caso adotado como 50, pois testes preliminares com o MATSim revelaram que, para a população de estudo, após 50 iterações o *score* total se estabilizava.

Um parâmetro secundário definido nesse módulo é: a frequência com que os arquivos de saída são gerados (definido como 10 para este estudo).

#### b) Módulo “Counts”

Nesse módulo, define-se o diretório do arquivo com as contagens de tráfego em alguns dos nós da rede viária de estudo. Também é importante definir a escala de população utilizada (10% nesse caso), para que a comparação seja feita corretamente.

#### c) Módulo “Network”

Define o diretório do arquivo *network.xml*.

#### d) Módulo “Plans”

Define o diretório do arquivo *population.xml*.

#### e) Módulo “plansCalcScore”

Nesse módulo, definem-se os parâmetros da função de utilidade usada para calcular os *scores* de cada plano e atividade. Nesse estudo, foram mantidas as configurações gerais de pontuação do MATSim, e introduzidas apenas as características de cada tipo de atividade considerada. Para atividades do tipo “*work*”, por exemplo, foi necessário especificar categorias de horários

de funcionamento dos estabelecimentos de trabalho. Essa especificação impede que o MATSim altere os horários de trabalho dos indivíduos no sentido de melhorar o *score* de um plano.

f) Módulo “*qSim*”

Nesse módulo é definido o número máximo de horas do dia contidas em cada simulação. Isso é importante para evitar que o programa faça iterações infinitas caso haja algum erro de configuração. Estabeleceu-se como 30 horas o limite de cada iteração (o programa simula desde a meia-noite de um dia até as 6 horas da manhã do dia seguinte).

g) Módulo “*Strategy*”

Esse módulo estabelece o que muda para os agentes entre cada iteração simulada. Três parâmetros estratégicos podem ser definidos: porcentagem dos agentes que troca de plano entre os planos contidos em sua memória (definido como 80%); porcentagem dos agentes que tenta uma nova rota (definido como 40%); porcentagem dos agentes que muda o horário de alguma de suas atividades móveis (definido como 20%).

Quando são criadas novas rotas ou horários de atividades, o MATSim insere um novo plano na memória do agente. Por isso, é importante definir também o número máximo de planos armazenados para cada agente e o que deve ser feito quando esse número é atingido. Definiu-se, nesse módulo, que cada agente teria no máximo 8 planos em sua memória e que, à medida que mais planos fossem criados, aqueles com menor *score* fossem eliminados da memória do agente.

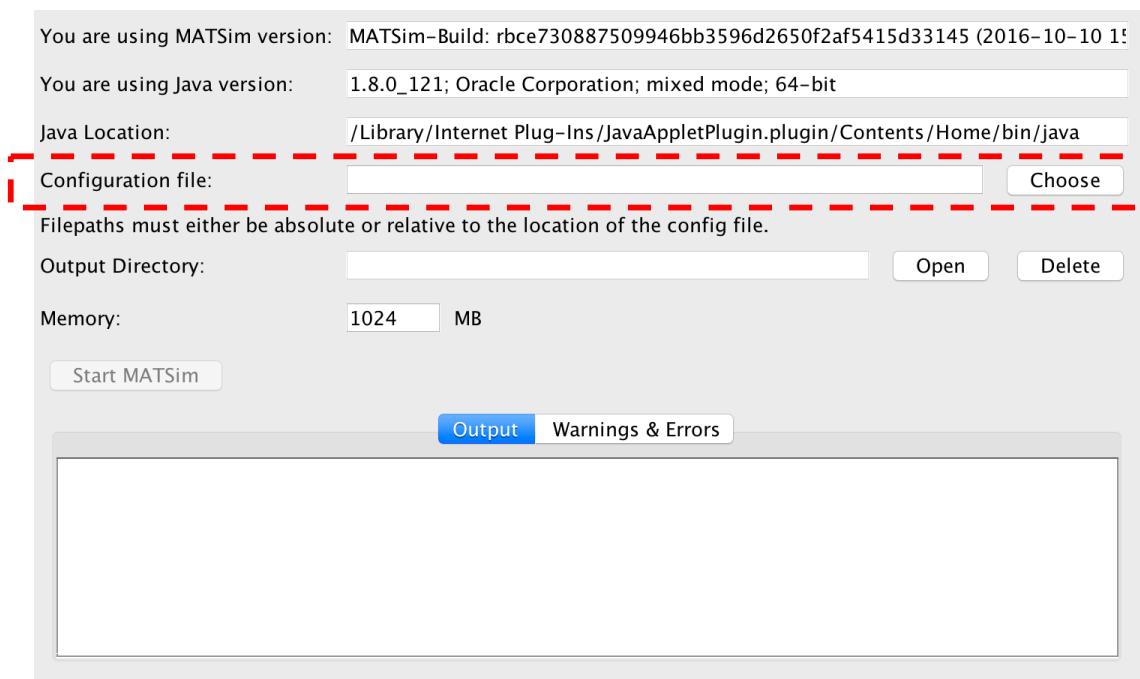
## 4. RESULTADOS

### 4.1 Introdução

Neste capítulo, são apresentados os resultados da simulação feita no *software* MATSim utilizando os arquivos gerados pela metodologia descrita no capítulo anterior. Dentre os produtos apresentados, estão o gráfico com os tempos de execução de cada processo da simulação, o gráfico de evolução dos *scores* dos planos ao longo das iterações e a visualização dos eventos ao longo de um dia.

### 4.2 Utilizando o MATSim 0.8.1

Diferente das primeiras versões lançadas do *software*, que necessitavam configuração via *Prompt* de comando, o MATSim 0.8.1 apresenta uma interface simplificada para o usuário. Nela, há o campo destinado a especificar a localização do arquivo de configuração, conforme apresentado na Figura 4.1.



**Figura 4.1:** Interface do MATSim 0.8.1; em destaque, campo para especificação do endereço do arquivo de configuração

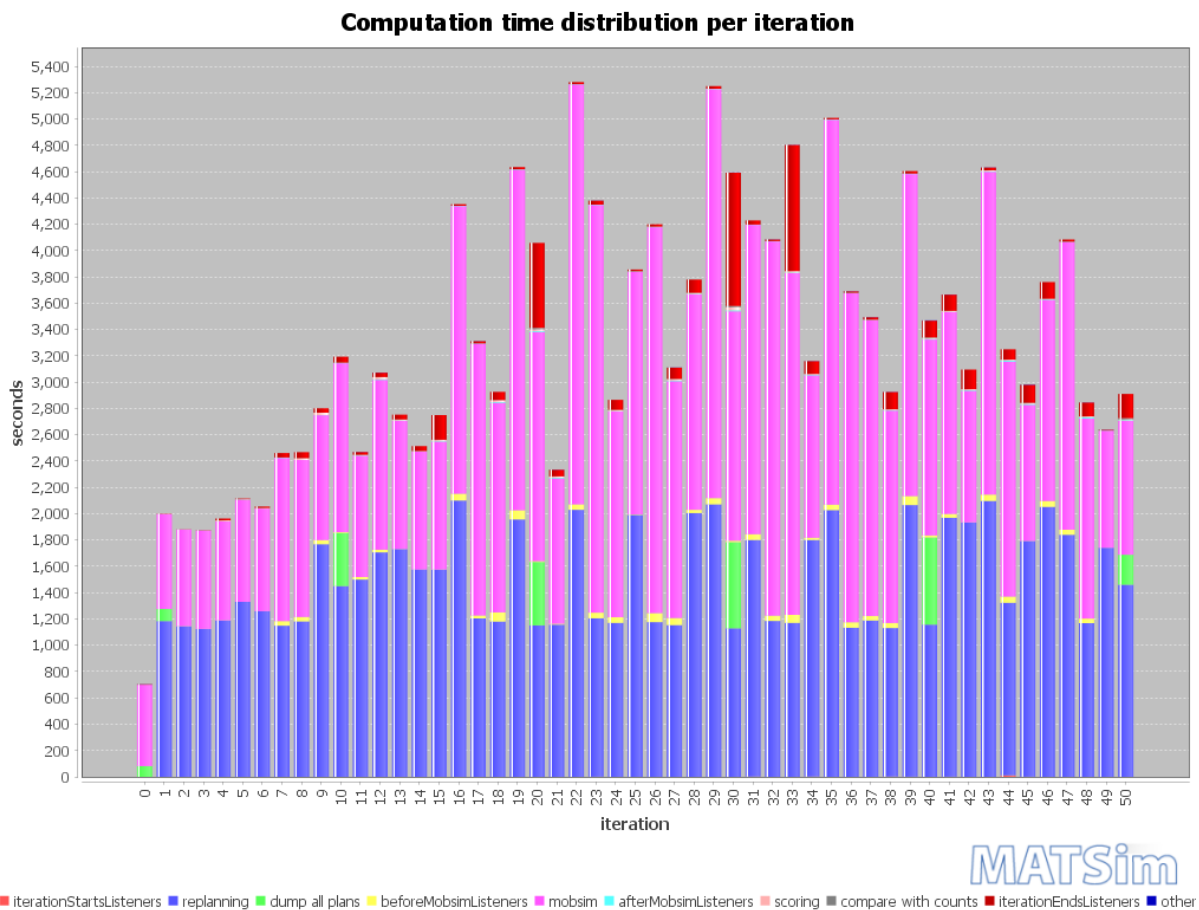
O campo “*memory*” da *interface* do MATSim 0.8.1 recebe como entrada a quantidade, em *Megabytes*, de memória RAM do computador que deve ser alocada para a aplicação Java.



Nesse estudo de caso, para a magnitude da simulação executada, em número de indivíduos e extensão da rede viária, provou-se necessário, no mínimo, 3072MB (ou 3GB) de memória RAM alocada para evitar o erro de falta de memória durante a execução do programa.

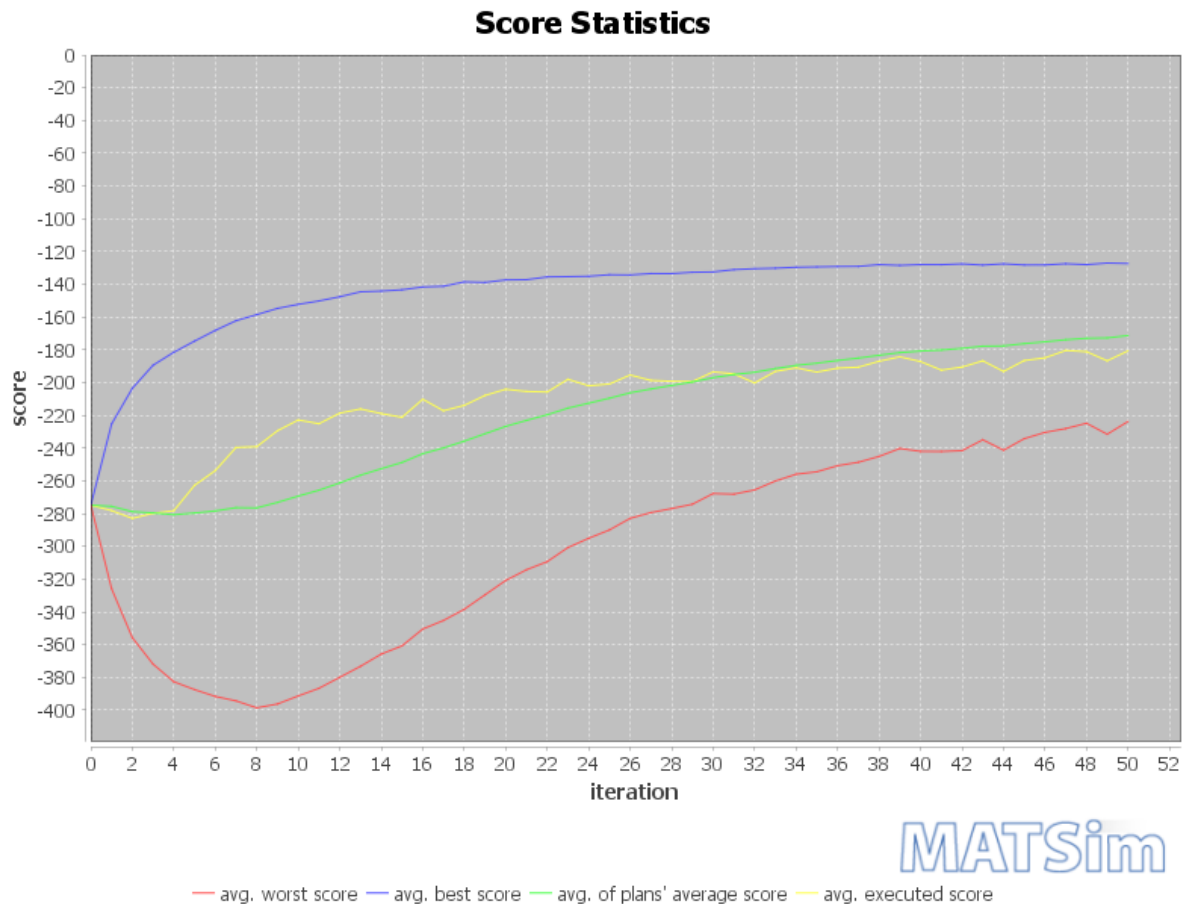
### 4.3 Produtos da simulação

No computador de configurações descritas no Apêndice D, a simulação no MATSim 0.8.1 com 50 iterações e com escala de 10% da população do DF, levou 51 horas, 11 minutos e 22 segundos para ser realizada. A Figura 4.2, gerada pelo próprio *software*, detalha o tempo gasto, em segundos, em cada processo de cada iteração. Percebe-se que os processos mais demorados são a simulação das rotas e o replanejamento entre cada iteração.



**Figura 4.2:** Tempo de execução dos processos em cada iteração (gerada pelo *software* MATSim como produto da simulação)

O MATSim também apresenta graficamente a evolução dos *scores* dos planos ao longo das iterações, conforme apresentado na Figura 4.3.



**Figura 4.3:** Evolução dos *scores* dos planos ao longo das iterações (gerada pelo *software* MATSim como produto da simulação)

Além das informações gráficas apresentadas, o MATSim também gera como produto um arquivo de texto com os planos dos agentes conforme executado na última simulação e um arquivo com os *eventos* em ordem cronológica durante o dia simulado. É considerado um *evento* todo momento em que um indivíduo começa ou termina uma atividade, entra ou sai de um veículo e entra ou sai de um trecho viário (*link*). Esse arquivo geralmente ocupa grande quantidade de memória em disco: nesse estudo de caso, o arquivo *events.xml* gerado ocupava 4,93 GB de espaço de memória. Um trecho do arquivo de eventos gerado é apresentado na Figura 4.4.

```

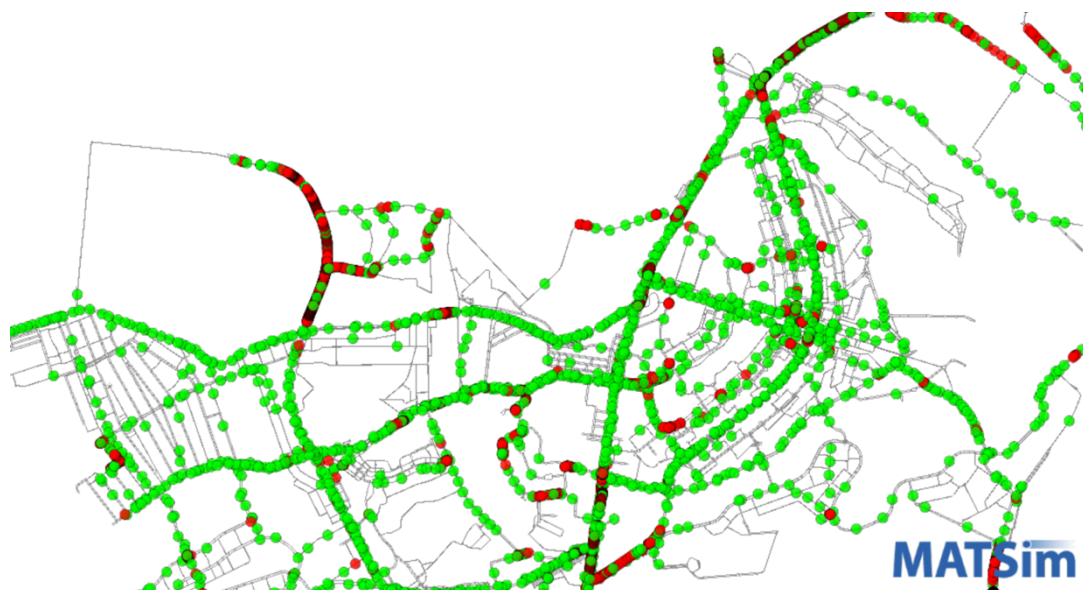
1 <?xml version="1.0" encoding="utf-8"?>
2 <events version="1.0">
3   <event time="2833.0" type="actend" person="1work1215" link="1015307.0" actType="home" />
4   <event time="2833.0" type="departure" person="1work1215" link="1015307.0" legMode="car" />
5   <event time="2833.0" type="PersonEntersVehicle" person="1work1215" vehicle="1work1215" />
6   <event time="2833.0" type="vehicle enters traffic" person="1work1215" link="1015307.0"
7   vehicle="1work1215" networkMode="car" relativePosition="1.0" />
8   <event time="2834.0" type="left link" vehicle="1work1215" link="1015307.0" />
9   <event time="2834.0" type="entered link" vehicle="1work1215" link="1082125.0" />
10  <event time="2847.0" type="left link" vehicle="1work1215" link="1082125.0" />
11  <event time="2847.0" type="entered link" vehicle="1work1215" link="1023978.0" />
12  <event time="2853.0" type="left link" vehicle="1work1215" link="1023978.0" />
13  <event time="2853.0" type="entered link" vehicle="1work1215" link="1026316.0" />
14  <event time="2863.0" type="left link" vehicle="1work1215" link="1026316.0" />
15  <event time="2863.0" type="entered link" vehicle="1work1215" link="1012907.0" />
16  <event time="2894.0" type="left link" vehicle="1work1215" link="1012907.0" />
17  <event time="2894.0" type="entered link" vehicle="1work1215" link="1047992.0" />
18  <event time="2903.0" type="left link" vehicle="1work1215" link="1047992.0" />
19  <event time="2903.0" type="entered link" vehicle="1work1215" link="1033133.0" />
20  <event time="2910.0" type="left link" vehicle="1work1215" link="1033133.0" />

```

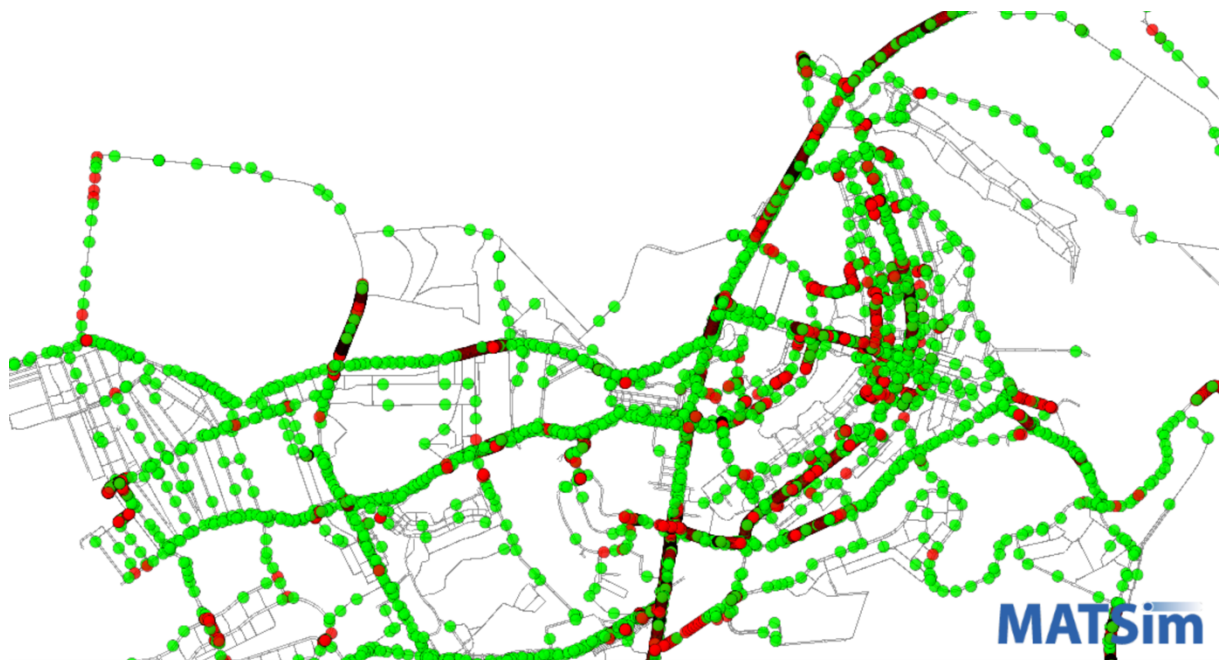
**Figura 4.4:** Trecho do arquivo *events.xml* gerado como produto da simulação do *MATSim*

#### 4.4 Visualização dos eventos

Para visualizar os eventos graficamente na rede viária, o *MATSim* disponibiliza uma extensão, o *OTFVis*, que pode ser acessado via janela de comando do sistema, de acordo com as instruções constantes na sua documentação. Nesse visualizador, os veículos que trafegam na velocidade de fluxo livre da via são apresentados em cor verde, enquanto os que têm velocidade reduzida são apresentados em vermelho, mais escuro quanto menor a velocidade. Nas Figuras 4-5 e 4-6 são representadas as visualizações para a hora de pico da manhã (7h30) e da noite (18h30).



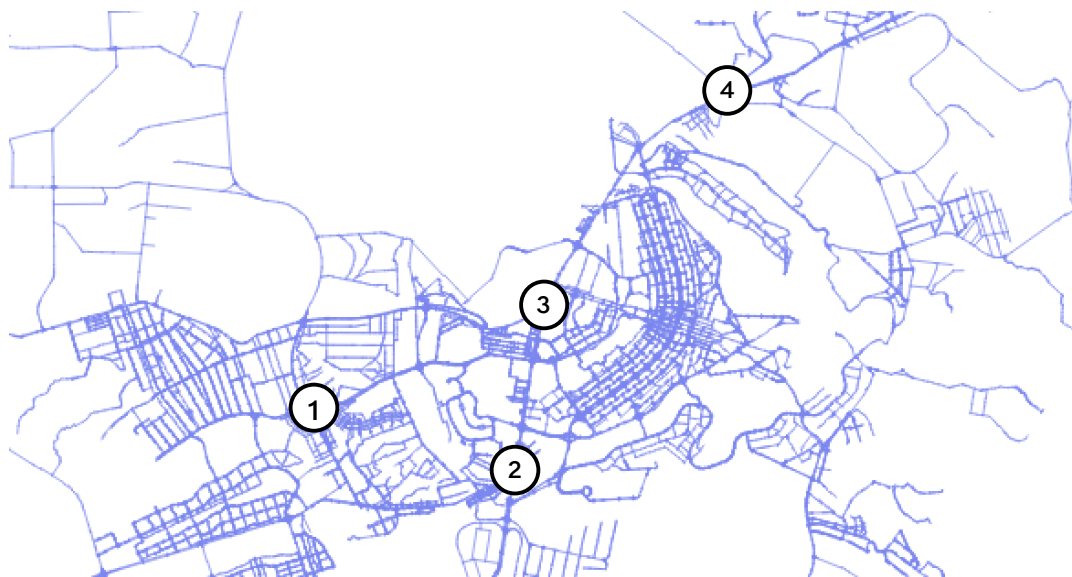
**Figura 4.5:** Visualização dos eventos de simulação no pico da manhã (07h30)



**Figura 4.6:** Visualização dos eventos de simulação no pico da noite (18h30)

#### 4.5 Análise dos resultados

Uma análise da simulação foi feita comparando-se o volume de veículos que percorriam determinados trechos viários a cada hora na simulação e as contagens reais desses volumes apresentadas nos relatórios do Plano Diretor de Transporte Urbano do DF, de 2009. Os pontos escolhidos para a comparação volumétrica são apresentados na Figura 4.7.



**Figura 4.7:** Pontos de contagem volumétrica avaliados

#### 4.5.1 Ponto 1: Interseção da DF-085 (EPTG) e DF-001 (EPCT) com a Avenida Central de Taguatinga

Os movimentos analisados nesse ponto são apresentados na Figura 4.8. A comparação entre volumes simulados e reais é apresentada na Tabela 4.1 e na Figura 4.9.

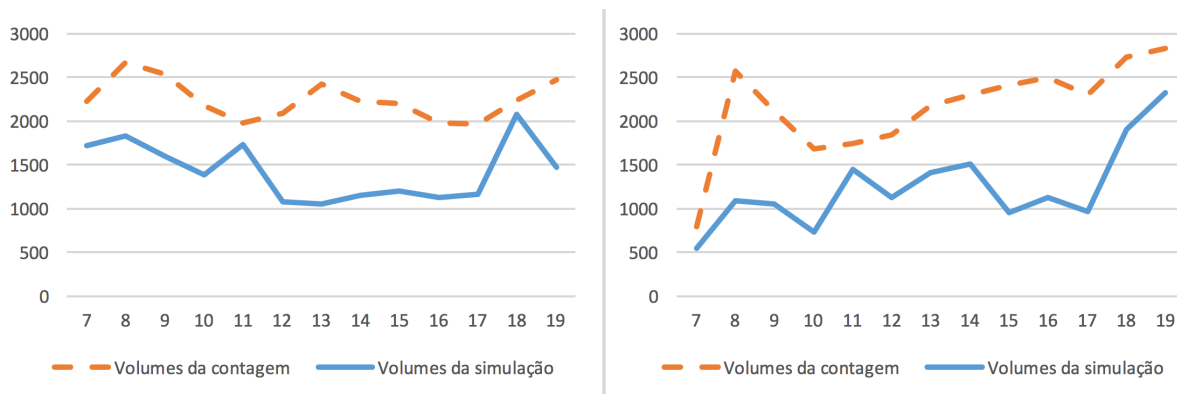


**Figura 4.8:** Movimentos analisados no Ponto 1

**Tabela 4.1:** Contagens e erros relativos no Ponto 1

HORÁRIO	CONTAGEM MOV. A	SIMULAÇÃO MOV. A	ERRO RELATIVO	CONTAGEM MOV. B	SIMULAÇÃO MOV. B	ERRO RELATIVO
7	2225	1720	23%	794	550	31%
8	2673	1830	32%	2572	1090	58%
9	2530	1600	37%	2109	1050	50%
10	2175	1380	37%	1684	730	57%
11	1974	1730	12%	1738	1450	17%
12	2086	1070	49%	1838	1130	39%
13	2427	1050	57%	2178	1410	35%
14	2228	1150	48%	2303	1510	34%
15	2198	1200	45%	2413	950	61%
16	1972	1130	43%	2500	1120	55%
17	1967	1160	41%	2295	960	58%
18	2235	2070	7%	2728	1900	30%
19	2472	1470	41%	2830	2320	18%
ERRO MÉDIO			36%	ERRO MÉDIO		42%
ERRO MÍNIMO			7%	ERRO MÍNIMO		17%
ERRO MÁXIMO			57%	ERRO MÁXIMO		61%





**Figura 4.9:** Comparação entre o volume de tráfego simulado para cada hora e o volume das contagens de tráfego, para os movimentos A (esquerda) e B (direita), no Ponto 1

Nesse ponto de comparação, observa-se que a simulação foi subdimensionada em aproximadamente 40% em ambos os sentidos. Não foi possível identificar o pico da manhã na simulação do Movimento A, que seria esperado para esse sentido pois é o fluxo em direção ao centro de Brasília.

#### 4.5.2 Ponto 2: Interseção da DF-075 (EPNB) com a DF-003 (EPIA)

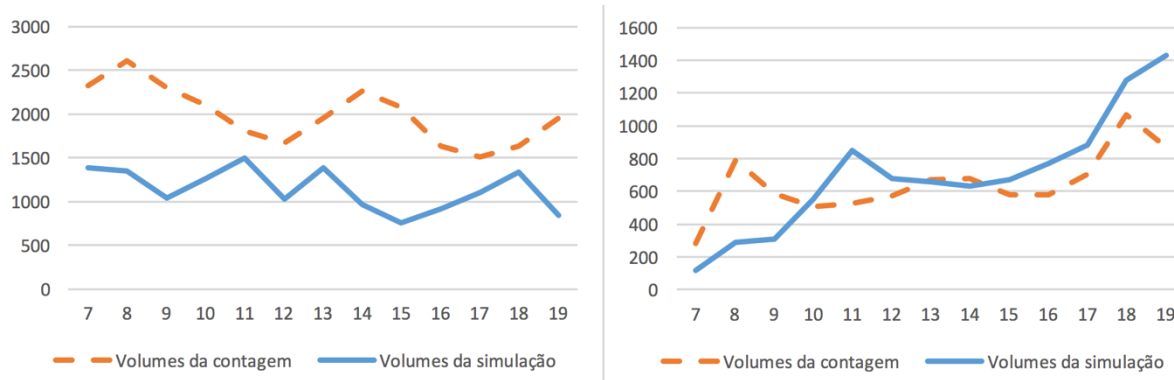
Os movimentos analisados nesse ponto são apresentados na Figura 4.10. A comparação entre volumes simulados e reais é apresentada na Tabela 4.2 e na Figura 4.11.



**Figura 4.10:** Movimentos analisados no Ponto 2

**Tabela 4.2:** Contagens e erros relativos no Ponto 2

HORÁRIO	CONTAGEM MOV. A	SIMULAÇÃO MOV. A	ERRO RELATIVO	CONTAGEM MOV. B	SIMULAÇÃO MOV. B	ERRO RELATIVO
7	2321	1390	40%	285	120	58%
8	2603	1350	48%	793	290	63%
9	2303	1040	55%	588	310	47%
10	2101	1260	40%	504	550	9%
11	1799	1490	17%	524	850	62%
12	1663	1020	39%	570	680	19%
13	1947	1380	29%	669	660	1%
14	2259	970	57%	679	630	7%
15	2075	760	63%	579	670	16%
16	1626	920	43%	581	770	33%
17	1512	1100	27%	702	880	25%
18	1632	1340	18%	1070	1280	20%
19	1958	840	57%	868	1430	65%
ERRO MÉDIO			41%	ERRO MÉDIO		33%
ERRO MÍNIMO			17%	ERRO MÍNIMO		1%
ERRO MÁXIMO			63%	ERRO MÁXIMO		65%



**Figura 4.11:** Comparação entre o volume de tráfego simulado para cada hora e o volume das contagens de tráfego, para os movimentos A (esquerda) e B (direita), no Ponto 2

Nesse ponto de comparação, percebe-se que o volume do movimento A foi subdimensionado em aproximadamente 40%. Nele, não é possível identificar o ponto de pico da manhã, que seria esperado por ser o movimento em direção ao centro de Brasília. Já o movimento B teve caracterização coerente com o fluxo acentuado no pico da noite, que é na direção de saída do centro de Brasília. Porém, o pico da manhã no movimento B foi atrasado em cerca de 3 horas.

### 4.5.3 Ponto 3: Interseção da DF-095 (EPCL) com a DF-003 (EPIA), Viaduto Ayrton Senna

Os movimentos analisados nesse ponto são apresentados na Figura 4.12. A comparação entre volumes simulados e reais é apresentada na Tabela 4.3 e na Figura 4.13.

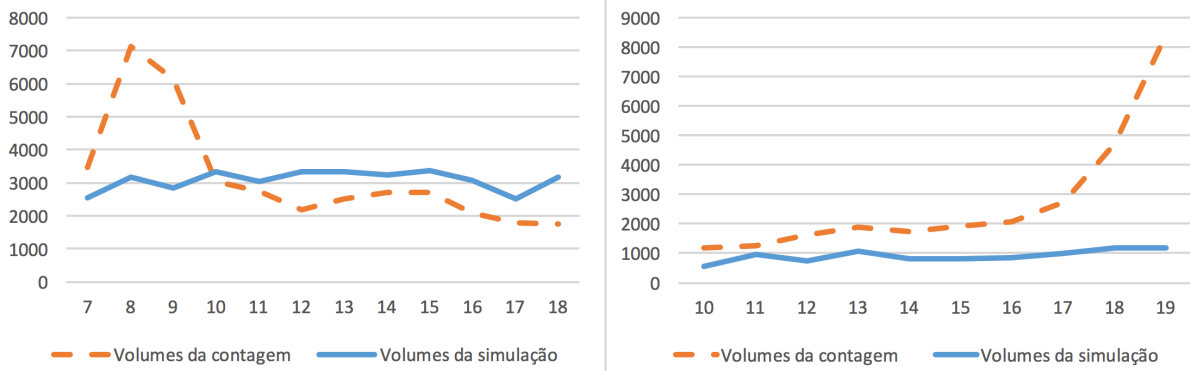


**Figura 4.12:** Movimentos analisados no Ponto 3

**Tabela 4.3:** Contagens e erros relativos no Ponto 3

HORÁRIO	CONTAGEM MOV. A	SIMULAÇÃO MOV. A	ERRO RELATIVO	CONTAGEM MOV. B	SIMULAÇÃO MOV. B	ERRO RELATIVO
7	3457	2550	26%	-	-	-
8	7137	3190	55%	-	-	-
9	6150	2840	54%	-	-	-
10	3035	3350	10%	1168	550	53%
11	2734	3030	11%	1253	960	23%
12	2176	3350	54%	1609	730	55%
13	2517	3330	32%	1874	1090	42%
14	2716	3250	20%	1748	810	54%
15	2724	3360	23%	1918	810	58%
16	2088	3090	48%	2082	860	59%
17	1785	2510	41%	2750	1010	63%
18	1749	3160	81%	4721	1180	75%
19	-	-	-	8433	1200	86%
ERRO MÉDIO			38%	ERRO MÉDIO		57%
ERRO MÍNIMO			10%	ERRO MÍNIMO		23%
ERRO MÁXIMO			81%	ERRO MÁXIMO		86%





**Figura 4.13:** Comparação entre o volume de tráfego simulado para cada hora e o volume das contagens de tráfego, para os movimentos A (esquerda) e B (direita), no Ponto 3

O ponto de comparação 3 destoou dos demais pois uma das vias analisadas (DF-095) sofre reversão durante os picos da manhã e da noite (entre 6h00 e 8h45 e entre 17h30 e 19h00). Essa reversão não foi incluída no modelo de simulação, então é possível perceber uma grande discrepância entre os volumes de tráfego das contagens e os simulados nos períodos de pico.

#### 4.5.4 Ponto 4: BR-010/020/030 no Balão do Colorado/Via de acesso a Sobradinho II

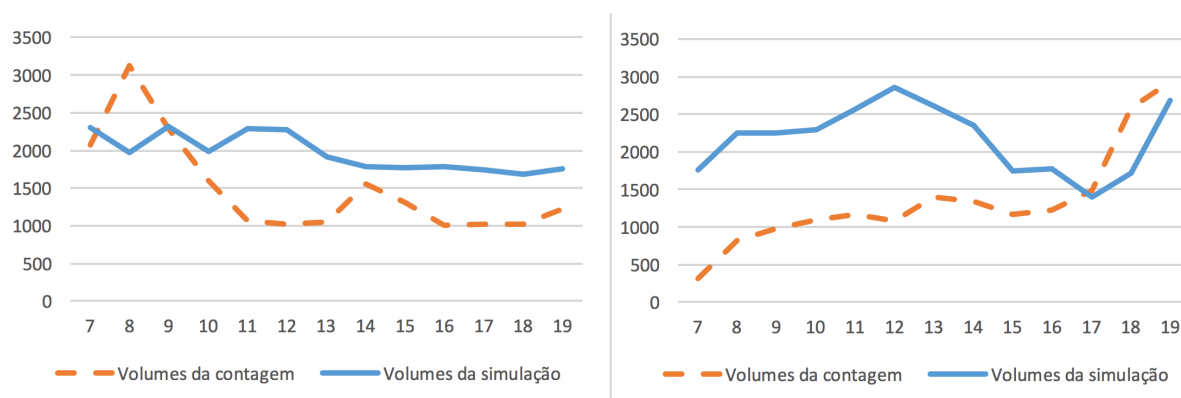
Os movimentos analisados nesse ponto são apresentados na Figura 4.14. A comparação entre volumes simulados e reais é apresentada na Tabela 4.4 e na Figura 4.15.



**Figura 4.14:** Movimentos analisados no Ponto 4

**Tabela 4.4:** Contagens e erros relativos no Ponto 4

HORÁRIO	CONTAGEM MOV. A	SIMULAÇÃO MOV. A	ERRO RELATIVO	CONTAGEM MOV. B	SIMULAÇÃO MOV. B	ERRO RELATIVO	
7	2078	2300	11%	321	1760	448%	
8	3125	1970	37%	816	2250	176%	
9	2288	2320	1%	983	2250	129%	
10	1590	1990	25%	1102	2300	109%	
11	1062	2290	116%	1165	2570	121%	
12	1023	2270	122%	1087	2850	162%	
13	1052	1910	82%	1395	2610	87%	
14	1549	1780	15%	1345	2350	75%	
15	1304	1770	36%	1171	1750	49%	
16	1012	1790	77%	1232	1770	44%	
17	1016	1740	71%	1491	1400	6%	
18	1024	1680	64%	2583	1720	33%	
19	1215	1760	45%	2931	2680	9%	
ERRO MÉDIO			54%	ERRO MÉDIO			111%
ERRO MÍNIMO			1%	ERRO MÍNIMO			6%
ERRO MÁXIMO			122%	ERRO MÁXIMO			448%



**Figura 4.15:** Comparação entre o volume de tráfego simulado para cada hora e o volume das contagens de tráfego, para os movimentos A (esquerda) e B (direita), no Ponto 4

Diferente dos demais pontos, os volumes de tráfego passando pelo ponto de contagem 4 foram superdimensionados em relação às contagens reais. Além disso, os picos da manhã e da noite foram pouco definidos nos dois movimentos.

O estudo de Bicudo (2015), que usou metodologia diferente na obtenção dos dados para simulação no MATSim, também obteve, de uma forma geral, resultados subdimensionados em relação às contagens de tráfego, e seu erro relativo mínimo obtido foi de cerca de 30%.

## 5. CONCLUSÃO E DISCUSSÕES

### 5.1 Conclusão do estudo

Nesse estudo, foi desenvolvida uma metodologia de tratamento de dados para simulação de modelo multiagente utilizando o *software* MATSim. As informações foram obtidas a partir de bases de dados *open-source* e relatórios censitários, e o tratamento e análise foram feitos utilizando *softwares* gratuitos e pouco complexos. Com isso, mostrou-se possível a modelagem desagregada de transporte em situações em que o planejamento sofre limitações técnicas e financeiras.

Os vetores geográficos criados, contendo informações sobre a rede viária da região do Distrito Federal e os pontos de realização de atividades, se mostraram ferramentas úteis para o gerenciamento de informações da área de estudo. As rotinas em linguagem Python necessárias para a criação e organização dos bancos de dados tiveram velocidade de processamento relativamente rápida em um computador de médio porte, o que viabiliza a criação de vetores para outras cidades e regiões.

A elaboração do arquivo de população sintética demandou um grande número de informações censitárias a respeito da área de estudo, além de exigir um código mais longo e complexo do que o utilizado na geração do banco de dados geográficos. Para simplificar o processo, foi necessário agrupar algumas informações (por exemplo, assumir que todas as subregiões de estudo apresentavam a mesma taxa de carros que saíam da garagem diariamente), o que certamente influenciou a verossimilhança do produto gerado.

Os dados tratados pela metodologia desenvolvida foram utilizados para alimentar um modelo do *software* MATSim em um estudo de caso da região do Distrito Federal e Entorno. Como produto principal, obteve-se um arquivo com informações detalhadas sobre o comportamento de transporte de cada indivíduo e suas interações mútuas ao longo de um dia. O modelo foi verificado fazendo-se uma comparação com contagens volumétricas em alguns trechos viários da área de estudo. Os resultados mostraram que o modelo, de uma forma geral, foi subdimensionado, o que pode ser consequência do modelo só incluir automóveis, desconsiderando ônibus e outros veículos que compõem o sistema de transporte da região.

## 5.2 Limitações encontradas na pesquisa

Como mencionado anteriormente, o modelo gerado pelo MATSim incluiu apenas o modo de transporte “automóvel”. Apesar do *software* oferecer extensões que possibilitam a inclusão de transporte público no modelo, o tempo de pesquisa foi insuficiente para obter e tratar os dados referentes ao transporte público da região.

Além disso, devido à falta de dados recentes a respeito das contagens de tráfego na região de estudo, foi necessário comparar os resultados do modelo com dados de contagens feitos em 2009, o que era incompatível com outros dados utilizados na modelagem. A rede viária, por exemplo, obtida a partir da base *OpenStreet Map*, foi feita com base em informações de 2016, e alguns dados censitários obtidos da Pesquisa Distrital por Amostra de Domicílios faziam referência ao ano de 2011.

## 5.3 Contribuições do estudo

Para o meio acadêmico, a contribuição principal desta pesquisa foi viabilização da criação de vetores geográficos que facilitam o gerenciamento de informações a longo prazo. Com isso, permite-se que os pesquisadores mantenham bases de dados completas e atualizadas de forma colaborativa, compatibilizando as pesquisas realizadas na Universidade, por exemplo. Além disso, conforme verificado durante a etapa de revisão de literatura, este estudo foi o primeiro a tratar de tratamento de dados para a modelagem na área de transportes.

A contribuição técnica mais evidente do estudo está representada pelos códigos e rotinas desenvolvidos no método de pesquisa. Esse projeto final apresenta novas formas de abordar o planejamento de transportes, utilizando técnicas de programação que podem servir como base para projetos futuros.

Por fim, a contribuição social da pesquisa é a viabilização da modelagem de transporte como ferramenta para um planejamento mais moderno no Brasil, principalmente no âmbito dos municípios. A metodologia, que faz uso de *softwares* livres e informações acessíveis, pode contribuir para a universalização da modelagem para o planejamento de transportes no país.

## 5.4 Sugestões para trabalhos futuros

Para trabalhos futuros realizados na mesma temática, sugere-se:

- Verificar a otimização dos códigos e rotinas gerados, de forma a reduzir o tempo de processamento dos programas;
- Aprimorar a rotina de geração de população sintética, utilizando diários de atividades para validar e calibrar as hipóteses feitas na elaboração do código;
- Realizar estudos de caso em regiões com menores dimensões, mas efetuando mais iterações no *software* MATSim, ou com maiores fatores de escala na simulação;
- Testar novas formas de configuração das estratégias de otimização dos planos (proporção de agentes afetados a cada iteração, por exemplo).

## BIBLIOGRAFIA

ARRUDA, F. S. (2005) *Aplicação de um modelo baseado em atividades para análise da relação uso do uso e transportes no contexto brasileiro*. Tese de doutorado, Universidade de São Paulo - USP, 199 p.

ASSOCIAÇÃO NACIONAL DAS EMPRESAS DE TRANSPORTES URBANOS. (2015) Menos de 30% das principais cidades brasileiras possuem Plano de Mobilidade Urbana. *NTUurbano*, v. 3, n. 13, p. 14–19.

BADDOE, D. A.; MILLER, E. J. (2000) Transportation-land-use interaction: empirical findings in North America, and their implications for modeling. *Transport Research Part D*, v. 5, p. 235–263.

BAZZAN, A. L. C. (2009) Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, v. 18, n. 3, p. 342–375.

BHAT, C.; GUO, J.; SRINIVASAN, S.; SIVAKUMAR, A. (2000) *Guidebook on Activity-Based Travel Demand Modeling for Planners*. University of Texas at Austin, Austin, TX, 30 p.

BHAT, C. R.; KOPPELMAN, F. S. (2003) *Activity-Based Modeling of Travel Demand*. Handbook of Transport Science, HALL, R. (ed.), Springer Science & Business Media, p. 39–65.

BICUDO, D. G. (2015) *Aplicação do Simulador de Tráfego MATSim à Cidade de Joinville/SC*. Trabalho de Conclusão de Curso, Universidade Federal de Santa Catarina - UFSC, 129 p.

BOYCE, D.; WILLIAMS, H. (2015) *Forecasting Urban Travel: Past, Present and Future*. Edward Elgar Publishing, 672 p.

BRASIL. Lei nº 9.503, de 23 de setembro de 1997, Institui o Código Brasileiro de Trânsito.

BRÖMMELSTROET, M. TE; BERTOLINI, L. (2011) The Role of Transport-Related Models in Urban Planning Practice. *Transport Reviews*, v. 31, n. 2, p. 139–143.

CAMPOS, V. B. G. (2013) *Planejamento de Transportes: Conceitos e Modelos*. Interciência, Rio de Janeiro, 188 p.

CAVALCANTE, R. (2015) *Apostila de Introdução ao SIG*. Universidade Federal de Minas Gerais, Belo Horizonte, 38 p.

CHARYPAR, D.; NAGEL, K. (2005) Generating complete all-day activity plans with genetic algorithms. *Transportation*, v. 32, n. 4, p. 369–397.

CHWIF, L.; MEDINA, A. (2014) *Modelagem e simulação de eventos discretos: teoria e aplicações*. 4a. ed. Elsevier Brasil, São Paulo, 320 p.

CODEPLAN. (2012) *Pesquisa distrital por amostra de domicílios - Distrito Federal - PDAD/DF 2011*. Secretaria do Estado de Planejamento e Orçamento do Distrito Federal, Brasília, 228 p.

ESRI ARCGIS. (2017) Shapefiles. Disponível em: <<http://doc.arcgis.com/pt-br/arcgis-online/reference/shapefiles.ht>>. Acesso em: 11 maio 2017.

ETTEMA, D. (1996) *Activity-based travel demand modeling*. Tese de Doutorado, Technische Universiteit Eindhoven, 317 p.

FARINHA, P. M. L. (2013) *Modelos de Simulação em MATSim aplicados à análise de Sistemas de Transportes*. Dissertação de Mestrado, Instituto Superior de Engenharia de Lisboa.

FURIERI, A. F. (2017) SpatiaLite. Disponível em: <<https://www.gaia-gis.it/fossil/libspatialite/index>>. Acesso em: 11 maio 2017.

FURTADO JUNIOR, M. B. (2008) XML - Extensible Markup Language. Disponível em: <[https://www.gta.ufrj.br/grad/00\\_1/miguel/index.html](https://www.gta.ufrj.br/grad/00_1/miguel/index.html)>. Acesso em: 12 mar. 2017.

GARBER, N. J.; HOEL, L. A. (2001) *Traffic and highway engineering*. 3a. ed. Brooks/Cole, Pacific Grove, Estados Unidos, 1249 p.

GOLDNER, L. G. (2014) *Análise de Capacidade de Vias com base no HCM 2010*. Universidade Federal de Santa Catarina, Florianópolis, 61 p.

GUDMUNDSSON, H. (2011) Analysing Models as a Knowledge Technology in Transport Planning. *Transport Reviews*, v. 31, n. 2, p. 145–159.

HENSHER, D.; BUTTON, K. (2008) *Handbook of Transport Modelling*. Emerald, Bingley, Reino Unido, 546 p.

HORNI, A.; NAGEL, K.; AXHAUSEN, K. W. (2016) *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, London, 662 p.

JONSSON, D.; BERGLUND, S.; ALMSTRÖM, P.; ALGERS, S. (2011) The Usefulness of Transport Models in Swedish Planning Practice. *Transport Reviews*, v. 31, n. 2, p. 251–265.

LEYTON-BROWN, K.; SHOHAM, Y. (2008) *Essentials of game theory*. Morgan & Claypool, 104 p.

LOPES, A. S. (2015) *Transportes, Uso do Solo e Atividades - Modelagem Conceitual para o Planejamento da Acessibilidade Urbana*. Tese de Doutorado, Universidade Federal do Ceará (UFC).

MACAL, C.; NORTH, M. (2010) Tutorial on agent-based modelling and simulation. *Journal of Simulation*, v. 4, n. 3, p. 151–162.

MAERIVOET, S.; DE MOOR, B. (2008) *Transportation Planning and Traffic Flow Models*. Katholieke Universiteit Leuven, Leuven, Bélgica, 51 p.

- MARIN, T.; TOMI, G. F. C. DE. (2010) Modelagem de dados de entrada para simulação estocástica de lavra. *Rem: Revista Escola de Minas*, v. 63, n. 3, p. 559–562.
- MEDRANO, R. M. A. (2012) *Modelagem de padrões de viagens e expansão urbana*. Dissertação de Mestrado, Universidade de Brasília, 157 p.
- MEDRANO, R. M. A. (2016) *O Modelo Intencional de Transporte: Contribuições da Ontologia de Bunge para formalização da Teoria de Comportamento em Transporte*. Universidade de Brasília, 195 p.
- MINISTÉRIO DAS CIDADES. (2015) *PlanMob: Caderno de referência para elaboração do plano de mobilidade urbana*. Secretaria Nacional de Transporte e da Mobilidade Urbana, Brasília, DF, 238 p.
- MÖLLER, D. P. F. (2014) *Introduction to Transportation Analysis, Modeling and Simulation*. Springer-Verlag, London, 343 p.
- OLIVEIRA, N. M. DE C.; FAVARETTO, F. (2013) Coleta De Dados Para Construção De Modelo De Simulação a Eventos Discretos De Um Restaurante. *XLV SBPO - Simpósio Brasileiro de Pesquisa Operacional*, p. 1332–1343.
- OPENSTREET MAP WIKI. (2015) Highway: International Equivalence. Disponível em: <[http://wiki.openstreetmap.org/w/index.php?title=Highway:International\\_equivalence&oldid=1225661](http://wiki.openstreetmap.org/w/index.php?title=Highway:International_equivalence&oldid=1225661)>. Acesso em: 31 maio 2017.
- OPENSTREET MAP WIKI. (2016) Points of Interest. Disponível em: <[http://wiki.openstreetmap.org/w/index.php?title=Points\\_of\\_interest&oldid=1338929](http://wiki.openstreetmap.org/w/index.php?title=Points_of_interest&oldid=1338929)>. Acesso em: 12 abr. 2017.
- OPENSTREET MAP WIKI. (2017) About OpenStreetMap. Disponível em: <<http://wiki.openstreetmap.org/wiki/About>>. Acesso em: 11 maio 2017.
- ORTÚZAR, J. DE D.; WILLUMSEN, L. G. (2004) *Modelling Transport*. 3a. ed. John Wiley & Sons, 508 p.
- PARSONS, S.; WOOLDRIDGE, M. (2002) Game Theory and Decision Theory in Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, v. 5, n. 3, p. 243–254.
- PEREIRA, I.; ALVES, T.; PINHEIRO, R.; ASSIS, E. (2004) Metodologia de tratamento de dados climáticos para inserção em softwares de simulação energética de edifícios. *Conferência Latino-Americana De Construção Sustentável, 1., Encontro Nacional De Tecnologia Do Ambiente Construído, 10., 2004, São Paulo*,
- PITOMBO, C. S. (2007) *Estudos de relações entre variáveis socioeconômicas, de uso do solo, participação em atividades e padrões de viagens encadeadas urbanas*. Tese de Doutorado, Universidade de São Paulo, 285 p.
- QGIS FOUNDATION. (2017) Discover. Disponível em: <<http://qgis.org/en/site/about/index.html>>. Acesso em: 11 maio 2017.



- SECRETARIA DE TRANSPORTES. (2011) *Plano diretor de transporte urbano e mobilidade do Distrito Federal e entorno - Relatório Técnico nº 3*. Brasília, 78 p.
- SHIFTAN, Y.; BEN-AKIVA, M. (2011) A practical policy-sensitive, activity-based, travel-demand model. *Annals of Regional Science*, v. 47, n. 3, p. 517–541.
- SIEBERS, P.-O.; AICKELIN, U. (2008) *Introduction to multi-agent simulation*. University of Nottingham, 25 p.
- SILVA, P. C. M. DA. (2007) *Teoria do Fluxo de Tráfego*. Universidade de Brasília, 28 p.
- SIVAKUMAR, A. (2007) *Modelling transport: a synthesis of transport modelling methodologies*. Imperial College of London Working Paper, 32 p.
- SMITH, L.; BECKMAN, R.; BAGGERLY, K. (1995) *TRANSIMS: Transportation analysis and simulation system*. Los Alamos National Laboratory, Los Alamos, EUA.
- SPEAR, B. D. (1996) New approaches to transportation forecasting models: A synthesis of four research proposals. *Transportation*, n. 23, p. 215–240.
- STONE, P. (1997) Single-Agent vs . Multiagent Systems. Disponível em: <<http://www.cs.cmu.edu/afs/cs/usr/pstone/public/papers/97MAS-survey/node4.html>>. Acesso em: 20 mar. 2017.
- TACO, P. W. G. (2003) *Redes Neurais Artificiais Aplicadas na Modelagem Individual de Padrões de Viagens Encadeadas a Pé*. Tese de Doutorado, Universidade de São Paulo, 225 p.
- TAKANO, M. S. M. (2010) *Análise da influência da forma urbana no comportamento de viagens encadeadas com base em padrões de atividades*. Dissertação de Mestrado, Universidade de Brasília, 233 p.
- TIMMERMANS, H.; ARENTZE, T. A. (2011) Transport Models and Urban Planning Practice: Experiences with Albatross. *Transport Reviews*, v. 31, n. 2, p. 199–207.
- TRANSPORTATION RESEARCH BOARD. (2000) *Highway Capacity Manual*. TRB, Washington DC, USA., 1207 p.
- TREINTA, F. T.; FARIAS FILHO, J. R.; SANT’ANNA, A. P.; RABELO, L. M. (2014) Metodologia de pesquisa bibliográfica com a utilização de método multicritério de apoio à decisão. *Production*, v. 24, n. 3, p. 508–520.
- WOOLDRIDGE, M. (2002) *An introduction to multi-agent statistics*. John Wiley & Sons, London, 365 p.
- WOOLDRIDGE, M. J. (1992) *The Logical Modelling of Computational Multi-Agent Systems*. University of Manchester, 163 p.

## APÊNDICE A - Revisão Sistemática de Literatura

Abaixo, são apresentadas as bases de dados consultadas, os termos utilizados em cada busca e os respectivos resultados obtidos.

Data da Pesquisa:	23/08/2016
Base de Dados:	Banco de Teses da CAPES
Termos Pesquisados:	Modelagem AND Transportes AND Atividades
Resultados:	46
Documentos selecionados de acordo com relevância: MEDRANO, R. M. A. Modelagem de padrões de viagens e expansão urbana. Brasília: UnB, 2012. LOPES, A, S. Transportes, uso do solo e atividades: Modelagem conceitual para o planejamento da acessibilidade urbana. Fortaleza: UFC, 2015.	

Data da Pesquisa:	23/08/2016
Base de Dados:	Banco de Teses da CAPES
Termos Pesquisados:	Modelagem AND Transportes AND Agentes
Resultados:	14
Documentos selecionados de acordo com relevância: Nenhum.	

Data da Pesquisa:	06/09/2016
Base de Dados:	Portal de Periódicos da CAPES
Termos Pesquisados:	Planejamento AND Transportes in: "Título"
Resultados:	11
Documentos selecionados de acordo com relevância: VASCONCELLOS, E. A. A crise do planejamento de transportes nos países em desenvolvimento: reavaliando pressupostos e alternativas. Revista Transportes, v.3, n.2. 1995. CORREIA, D. E. R.; YAMASHITA, Y. Metodologia para identificação da qualidade da informação para planejamento de transportes. Revista Transportes, v.12, n.1. 2004.	

Data da Pesquisa:	11/09/2016
Base de Dados:	ProQuest
Termos Pesquisados:	Planejamento AND Transportes
Resultados:	5
Documentos selecionados de acordo com relevância: ANDRADE, A. R.; BALASSIANO, R.; SANTOS, M. P. S. Planejamento de Transportes: Informação e participação como fundamentos para o seu desenvolvimento. Revista de Gestão USP, v.13, n.3, p.13-22, jul/2006.	

Data da Pesquisa:	11/11/2016
Base de Dados:	Portal de Periódicos da CAPES
Termos Pesquisados:	Activity AND Based AND Travel AND Demand AND Model in: "Transportation"(Categoria)
Resultados:	130
Documentos selecionados de acordo com relevância: SPEAR, B. New approaches to transportation forecasting models. Transportation, n.23, p.215-240, 1996.	

BEN-AKIVA, M.; BOWMAN, J.; GOPINATH, D. Travel demand model system for the information era. *Transportation*, n. 23, p. 241-266, 1996  
 SHIFTAN, Y.; BEN-AKIVA, M. A practical policy-sensitive, activity-based, travel demand model. *Ann Reg Sci*, n. 47, p.517-541, 2011.  
 TRAVEL FORECASTING RESOURCE. Tour-based models. Disponível em: [http://tfresource.org/Tour-based\\_models](http://tfresource.org/Tour-based_models); Acesso em: 11 nov 2016.

Data da Pesquisa:	13/11/2016
Base de Dados:	Portal de Periódicos da CAPES
Termos Pesquisados:	(modelagem OR modelo) AND atividades AND transporte
Resultados:	24
Documentos selecionados de acordo com relevância: Nenhum.	

A seguir, são apresentadas as referências secundárias, que são documentos selecionados a partir das referências bibliográficas dos documentos primários (obtidos das pesquisas em bases de dados).

Referências secundárias
BAZAN, A. L. C. et al. ITSUMO: an Agent-Based Simulator for ITS Applications. Workshop on Artificial Transportation Systems and Simulation. 2010.
BHAT. Guidebook on Activity-Based Travel Demand Modeling for Planners. 2003.
MACAL, C. M.; NORTH, M. J. Tutorial on agent-based modelling and simulation. <i>Journal of Simulation</i> , n.4, p51-162. 2010.
PITOMBO, C. S. Estudos de relações entre variáveis socioeconômicas, de uso do solo, participação em atividades e padrões de viagens encadeadas urbanas. São Carlos: EESC/USP, 2007.
SMITH, W. S.; Mass Transport for High Rise High Density Living. <i>Journal of Transportation Engineering</i> , v.110, n.6, p.521-535, jul/1984
ANTIPOVA, A.; WANG, F.; WILMOT, C. Urban land uses, socio demographic attributes and commuting: A multilevel modeling approach. <i>Applied Geography</i> , v.31, n.3, p.1010-1018, jul/2011.
RODE, P. et al. Accessibility in cities: transport and urban form. <i>New Climate Economy Cities</i> , Paper 03. LSE Cities, London School of Economics and Political Science, London, 2014.
ZHU, Y.; FERREIRA JR, J. Synthetic population generation at disaggregated spatial scales for land use and transportation microsimulation. <i>Journal of the Transportation Research Board</i> . n.2429, p168-177, dez. 2014.
BECKOR, S., DOBLER, C., and AXHAUSEN, K. W. (2011). "Integration of activity-based and agent-based models." <i>Transportation Research Record</i> 2255, Transportation Research Board, Washington, DC, 38-47.

## APÊNDICE B - Estudos Anteriores com MATSim

Local do Estudo	Trabalho	Características
Berlim, Alemanha	Neumann, A., M. Balmer and M. Rieser (2014) Converting a static trip-based model into a dynamic activity-based model to analyze public transport demand in Berlin, in: M. Roorda and E. Miller (eds.) Travel Behaviour Research: Current Foundations, Future Prospects, 151–176, International Association for Travel Behaviour Research (IATBR).	Análise de demanda por transporte público na cidade. Usou módulos complementares referentes a transporte público e modos ativos.
Berlim, Alemanha	Ziemke, D., K. Nagel and C. Bhat (2015) Integrating CEMDAP and MATSim to increase the transferability of transport demand models, Transportation Research Record, 2493, 117–125.	Buscou a modelagem usando apenas dados simples de serem obtidos, disponíveis nas bases do governo, para gerar a demanda inicial.
Suíça	Balmer, M., A. Horni, K. Meister, F. Ciari, D. Charypar and K. W. Axhausen (2009a) Wirkungen der Westumfahrung Zürich: Eine Analyse mit einer Agenten-basierten Mikrosimulation, Final Report, Baudirektion Kanton Zurich, IVT, ETH Zurich, Zurich, February 2009.	Foi o cenário utilizado pelos criadores do MATSim como base para testar os algoritmos desenvolvidos.
Zurique, Suíça	Horni, A., D. Charypar and K. W. Axhausen (2011a) Variability in transport microsimulations investigated with the multi-agent transport simulation MATSim, Working Paper, 692, IVT, ETH Zurich, Zurich.	Assim como o cenário da Suíça, foi um cenário utilizado para testar algoritmos desenvolvidos para o MATSim, mas de forma mais detalhada.
Cingapura, Cingapura	Erath, A., A. Chakirov, P. J. Fourie, S. A. Ordóñez Medina, M. Shah, M. A. B. van Eggermond and K. W. Axhausen (2012) A largescale agent-based transport travel demand model for Singapore: The implementation of MATSim, Working Paper, Future Cities Laboratory, Singapore-ETH Centre (SEC), Singapore.	Modelo gerado para a cidade de Cingapura e usado para auxiliar o planejamento de transportes.
Munique, Alemanha	Kickhöfer, B., F. H. Hülsmann, R. Gerike and K. Nagel (2013) Rising car user costs: Comparing aggregated and geo-spatial impacts on travel demand and air pollutant emissions, in: T. Vanoutrive and A. Verhetsel (eds.) Smart Transport Networks: Decision Making, Sustainability and Market structure, NECTAR Series on Transportation and Communications Networks Research, 180–207, Edward Elgar Publishing Ltd, Cheltenham.	Modelo usado para simular a poluição atmosférica e as emissões de gases de efeito estufa, e como esses fatores são afetados com novas medidas de gerenciamento.
Aliaga, Turquia	Onelcin, P., Mutlu, M. M., Alver, Y. (2013) Evacuation plan of an industrial zone: Case study of a chemical accident in Aliaga, Turkey, and the comparison of two different simulation softwares, Safety Science, v.60, p.123-130.	Estudo que simula o tempo de evacuação e a demanda por transporte em uma área de risco após um acidente hipotético nas indústrias químicas da região.
Barcelona, Espanha	Trabalho em progresso, sendo realizado por Picornell, M. e Lenormand, M.	Modelo criado para avaliar o impacto de diferentes sistemas de bicicletas públicas na cidade.
Bélgica	Saadi, I., H. Eftekhar, A. Mustafa, J. Teller and M. Cools (2014) An agent-based micro-simulation framework to assess the impact of river floods on transportation systems: implementation trajectory for an assessment in the brussels metropolitan area, in: International Conference for Traffic and Transport Engineering, Belgrade, November 2014.	Simulação do impacto de enchentes nas demandas por transporte durante a evacuação de áreas.

Local do Estudo	Trabalho	Características
Bruxelas, Bélgica	Röder, D., I. Cabrita and K. Nagel (2013) Simulation-based sketch planning, part III: Calibration of a MATSim-model for the greater Brussels area and investigation of a cordon pricing for the highway ring, VSP working paper, 13-16, TU Berlin, Berlin.	Estudo que une a modelagem do MATSim a uma modelagem baseada no uso do solo (UrbanSim) de forma a avaliar o impacto do transporte no uso do solo e vice-versa.
Cottbus, Alemanha	Grether, D. S. (2014) Extension of a multi-agent transport simulation for traffic signal control and air transport systems, Ph.D. Thesis, TU Berlin, Berlin.	Modelo de simulação de sinalização semafórica.
Dublin, Irlanda	McArdle, G., E. Furey, A. Lawlor and A. Pozdnoukhov (2012) City-scale traffic simulation from digital footprints, in: International Workshop on Urban Computing, Beijing, August 2012.	Uso da microssimulação de tráfego com MATSim para demonstrar um novo modelo de escolha espacial.
Filipinas	Produzido por Yaneza, E., Doroja, G., Abuzo, A., sem publicação.	Estudo com objetivo de adaptar a simulação de tráfego baseada em agentes para auxiliar as rotas de veículos de resposta a desastres durante emergências.
Gauteng, África do Sul	Nagel, K., B. Kickhöfer and J. W. Joubert (2014) Heterogeneous tolls and values of time in multi-agent transport simulation, Procedia Computer Science, 32, 762–768.	Estudo dos padrões de desvio de diferentes grupos de usuários de vias após a implementação de pedágios.
Hamburgo, Alemanha	Lämmel, G. and H. Klüpfel (2012) Slower is faster: the influence of departure time distribution on the overall evacuation performance, in: International Conference on Evacuation Modeling and Management, Northwestern University, Evanston, Illinois, USA.	Estudo de evacuação de áreas de risco de enchentes.
Joinville, Brasil	Bicudo, D. G. Aplicação do Simulador de Tráfego MATSim à Cidade de Joinville/SC. Joinville: UFSC, 2015.	Foi o primeiro passo de um projeto de simulação que englobará todo o nordeste do estado de Santa Catarina, incluindo tráfego aéreo, marítimo e por automóvel.
Londres, Reino Unido	Serras, J., Bosredon, M., Zachariadis, V., Batty, M. (2014) Building a travel demand model using an activity-centred approach, in the ECCS, September, 22-25, Italy.	Modelo criado para auxiliar o planejamento de transportes.
Nova Iorque, EUA	Parsons Brinckerhoff (2009) New York Best Practice Model (NYBPM) For Regional Travel Demand Forecasting - NYBPM User Documentation BPM 2005 Update, Final Report, New York Metropolitan Transportation Council (NYMTC), Parsons Brinckerhoff Quade & Douglas, Inc., New York.	Modelo criado para auxiliar o planejamento de transportes.
Padang, Indonésia	Lämmel, G. (2011) Escaping the tsunami: Evacuation strategies for large urban areas. concepts and implementation of a multi-agent based approach, Ph.D. Thesis, TU Berlin, Berlin.	Estudo de evacuação de áreas de risco de enchentes.
Poznan, Polónia	Maciejewski, M., B. Piatkowski and W. Walerjanczyk (2014) Od makroskopowego modelu popytu na podroze do calodobowej mikroskopowej symulacji przeplywu ruchu, Przegląd Komunikacyjny, 2, 27–31.	Modelo criado para auxiliar o planejamento de transportes.
Quito, Equador	Armas, R., H. Aguirre and K. Tanaka (2014) Effects of mutation and crossover operators in the optimization of traffic signal parameters, in: The Tenth International Conference on Simulated Evolution And Learning (SEAL 2014), Otago, December 2014.	Estudo com objetivo de avaliar o sistema de transporte da região e propor soluções sustentáveis para os problemas de poluição e consumo energético.

Local do Estudo	Trabalho	Características
Rotterdam, Holanda	Lovric, M., T. Li and P. Vervest (2013) Sustainable revenue management: A smart card enabled agent-based modeling approach, <i>Decision Support Systems</i> , 54 (2 - Special Issue: Rapid Modeling for Sustainability) 1587–1601.	Simulação de cenários para estudo de retorno financeiro do transporte público utilizando gestão estratégica.
Samara, Rússia	Saprykina, O. and O. Saprykin (2014) The detection algorithm of concentration points of traffic flows on the road network in GIS, vol. 1 of <i>IT&amp;Transport</i> , Inteltrans, Samara.	Modelo criado para auxiliar o planejamento de transportes.
São Francisco, EUA	Pozdnoukhov, A., Campbell, A., Feygin, S., Yin, M., Mohanty, S. Sem publicação.	Baseado no MATSim, o estudo desenvolveu uma plataforma capaz de gerar modelos de demanda a partir de megadados.
Santiago, Chile	Kickhöfer, B., D. Hosse, K. Turner and A. Tirachini (2016) Creating an open MATSim scenario from open data: The case of Santiago de Chile, VSP Working Paper, 16-02, TU Berlin, <i>Transport Systems Planning and Transport Telematics</i> .	Modelo criado para auxiliar o planejamento de transportes.
Seattle, EUA	Nagel, K. Sem publicação.	Estudo que iniciou a ideia de junção do MATSim com um modelo de simulação baseado no uso do solo (UrbanSim).
Seul, Coreia do Sul	Kim, J. Y., Y. S. Yu, S. J. Lee, H. Hu and J. G. Sung (2012) Application of multi-agent transport simulation for urban road network operation in incident case, <i>International Journal of Highway Engineering</i> , 14 (4) 163–163.	Modelo criado para auxiliar o planejamento de transportes.
Shanghai, China	Zhang, L., W. Yang, J. Wang and Q. Rao (2014) Large-scale agent-based transport simulation in Shanghai, China, <i>Transportation Research Record</i> , 2399, 34–43.	Modelo criado para auxiliar o planejamento de transportes.
Sochi, Russia	Projeto criado pela empresa Senozon em parceria com a Masterconcept Consulting GmbH.	Modelo criado para auxiliar o planejamento de transportes, especificamente para simular a demanda durante as Olimpíadas de Inverno de 2014.
Stockholm, Suécia	Bischoff, J. (2013) Agentenbasierte Simulation elektrifizierter Taxiflotten, Master's thesis, TU Berlin, Institute for Land and Sea Transport Systems, Berlin, 09 2013.	Modelo criado para auxiliar o planejamento de transportes.
Tampa, EUA	Gurram, S., A. L. Stuart and A. R. Pinjari (2015) Impacts of travel activity and urbanicity on exposures to ambient oxides of nitrogen and on exposure disparities, <i>Air Quality, Atmosphere &amp; Health</i> , 8 (1) 97–114.	Estudo sobre a contribuição dos modos de transporte com a poluição atmosférica.
Tel Aviv, Israel	Dobler, C., A. Horni and K. W. Axhausen (2014) Integration of activity-based and agent-based models: Recent developments for Tel Aviv, Israel, Working Paper, 1027, IVT, ETH Zurich, Zurich.	Modelo criado para auxiliar o planejamento de transportes.
Tokyo, Japan	Ito, M., D. Fukuda and J. Ma (2015) Development of hyperpath-based risk-averse route guidance system and its verification, in: 51st Conference of Infrastructure Planning and Management, Japan Society for Civil Engineers (in Japanese) (in Japanese).	Desenvolvimento de um sistema de navegação para motoristas alternativo ao modelo de caminho mais curto.

**APÊNDICE C - Detalhamento das variáveis envolvidas no cálculo de *scores* do  
MATSim**

Partindo da equação básica de utilidade desenvolvida por Charypar & Nagel (2005), tem-se:

$$S_{plan} = \sum_{q=0}^{N-1} S_{act,q} + \sum_{q=0}^{N-1} S_{trav,mode(q)}$$

**Eq. C.1**

Sendo:

$S_{act,q}$  = Vantagens de todas as atividades;

$S_{trav,mode(q)}$  = Utilidades (negativas) de todas as viagens;

$N$  = Número de atividades;

$q$  = Viagem subsequente à atividade  $q$ ;

A função que define a utilidade de uma atividade  $q$  é definida pela Eq. C.2.

$$S_{act,q} = S_{dur,q} + S_{wait,q} + S_{late.ar,q} + S_{early.dp,q} + S_{short.dur,q}$$

**Eq. C.2**

As subcontribuições da Eq. C.2 são detalhadas a seguir.

- $S_{dur,q}$  é a utilidade de se realizar a atividade  $q$ , levando em conta o horário de abertura dos locais onde as atividades acontecem.

$$S_{dur,q} = (\beta_{dur,q})(t_{typ,q}) \left[ \ln \left( \frac{t_{dur,q}}{t_{0,q}} \right) \right]$$

**Eq. C.3**

Sendo:

$\beta_{dur,q}$  = Parâmetro relacionado à utilidade marginal da duração da atividade;

$t_{dur,q}$  = Duração da atividade realizada;

$t_{0,q}$  = Duração para a qual a utilidade passa a ser positiva.

- $S_{wait,q}$  é o tempo de espera, por exemplo, em frente ao local onde será feita a atividade  $q$ , antes de sua abertura.

$$S_{dur,q} = (\beta_{wait,q})(t_{wait,q})$$

**Eq. C.4**

Sendo:

$\beta_{wait,q}$  = Utilidade marginal direta do tempo gasto esperando (recomenda-se adotar zero para esse termo, segundo Horni, Nagel e Axhausen, 2016);

$t_{wait,q}$  = Duração do tempo de espera.

- $S_{late.ar,q}$  caracteriza a utilidade negativa por se chegar atrasado no local da atividade  $q$ .

$$S_{late.ar,q} = (\beta_{late.ar,q})(t_{start,q} - t_{latest.ar,q}), \text{ se } t_{start,q} > t_{latest.ar,q};$$

**Eq. C.5**

Ou:

$$S_{late.ar,q} = 0, \text{ nos demais casos.}$$

Sendo:

$t_{start,q}$  = O horário de início da atividade  $q$ ;

$t_{latest.ar,q}$  = O horário máximo para se chegar na atividade  $q$  sem utilidade negativa.

- $S_{early.dp,q}$  define a penalidade por não se permanecer tempo suficiente na atividade  $q$ .

$$S_{early.dp,q} = (\beta_{early.dp,q})(t_{end,q} - t_{earliest.dp,q}), \text{ se } t_{end,q} > t_{earliest.dp,q};$$

**Eq. C.6**

Ou:

$$S_{early.dp,q} = 0, \text{ nos demais casos.}$$

Sendo:

$t_{end,q}$  = O horário de término da atividade  $q$ ;

$t_{earliest.dp,q}$  = O horário mínimo para se sair da atividade  $q$  sem utilidade negativa.

Horni et al. (2016) recomendam adotar  $\beta_{early.dp,q} = 0$ , a não ser que se tenham dados realmente confiáveis sobre esse parâmetro.



- $S_{short.dur,q}$  é a utilidade negativa da realização de uma atividade  $q$  muito curta.

$$S_{short.dur,q} = (\beta_{short.dur,q})(t_{short.dur,q} - t_{dur,q}), \text{ se } t_{short.dur,q} > t_{dur,q};$$

**Eq. C.7**

Ou:

$$S_{short.dur,q} = 0, \text{ nos demais casos.}$$

Sendo:

$t_{short.dur,q}$  = O tempo de duração mais curto possível para a atividade  $q$ .

Horni et al. (2016) recomendam adotar  $\beta_{short.dur,q} = 0$ , a não ser que se tenham dados realmente confiáveis sobre esse parâmetro.

## **APÊNDICE D - Especificações da máquina utilizada para executar os programas**

Para a execução das rotinas e códigos apresentados nesse trabalho, foi utilizado computador com as seguintes especificações:

- Fabricante: Dell;
- Modelo: Inspiron 4400;
- Sistema operacional: Windows 7 Ultimate 64-bit;
- Processador: Intel Pentium T4200 2,00 GHz;
- Memória RAM Instalada: 4,00 GB

Esses parâmetros podem ser utilizados para interpretar o tempo requerido para execução de cada rotina apresentada neste documento.

## APÊNDICE E - Rotina para organização do banco de dados geográfico

No computador de especificações apresentadas no Apêndice D, a rotina a seguir, em Python, levou 86,1 segundos para ser executada, organizando o banco de dados geográficos do Distrito Federal e entorno, com 72.200 elementos viários.

```
1 # Bibliotecas para Python que permitem a leitura de arquivos CSV (Comma Separated Values)
2 # e bases de dados SQLite
3 import csv
4 import sqlite3
5
6 # Definindo os caminhos ate os arquivos que devem ser lidos
7 path_csv = 'map.csv';
8 path_sqlite = 'map.sqlite';
9
10 # Ordem das colunas do arquivo CSV
11 osm_idCol=0
12 nameCol=1
13 highwayCol=2
14 waterwayCol=3
15 aerialwayCol=4
16 barrierCol=5
17 man_madeCol=6
18 other_tagsCol=7
19
20 # Criando a classe "Vias" com todos os seus atributos
21 class Vias:
22     osm_id=''
23     name=''
24     highway=''
25     waterway=''
26     aerialway=''
27     barrier=''
28     man_made=''
29     lanes=''
30     maxspeed=''
31     surface=''
32     oneway=''
33
34 # Criando de um vetor dentro do programa com as informacoes do arquivo CSV
35 vias = []
36
37 # Conectando com o arquivo CSV
38 csvfile = open(path_csv, 'rt')
39
40 next(csvfile) # Pulando a primeira linha do arquivo
41 csvreader = csv.reader(csvfile, delimiter=',', quotechar='"')
42 register = 0;
43 for row in csvreader:
44     newEntry = Vias();
45     newEntry.osm_id = row[osm_idCol]
46     newEntry.name = row[nameCol]
47     newEntry.highway = row[highwayCol]
48     newEntry.waterway = row[waterwayCol]
49     newEntry.aerialway = row[aerialwayCol]
50     newEntry.barrier = row[barrierCol]
51     newEntry.man_made = row[man_madeCol]
52
53     other_tags = row[other_tagsCol].replace("=>", ",")
54     other_tags = other_tags.replace('"', '')
55     other_tags = other_tags.split(',')
56
```



```

128 command = "DELETE FROM result WHERE highway = 'living_street'"
129 cursor.execute(command)
130 db.commit()
131
132 command = "DELETE FROM result WHERE highway = 'pedestrian'"
133 cursor.execute(command)
134 db.commit()
135
136 command = "DELETE FROM result WHERE highway = 'track'"
137 cursor.execute(command)
138 db.commit()
139
140 command = "DELETE FROM result WHERE highway = 'bus_guideway'"
141 cursor.execute(command)
142 db.commit()
143
144 command = "DELETE FROM result WHERE highway = 'escape'"
145 cursor.execute(command)
146 db.commit()
147
148 command = "DELETE FROM result WHERE highway = 'raceway'"
149 cursor.execute(command)
150 db.commit()
151
152 command = "DELETE FROM result WHERE highway = 'road'"
153 cursor.execute(command)
154 db.commit()
155
156 command = "DELETE FROM result WHERE highway = 'footway'"
157 cursor.execute(command)
158 db.commit()
159
160 command = "DELETE FROM result WHERE highway = 'bridleway'"
161 cursor.execute(command)
162 db.commit()
163
164 command = "DELETE FROM result WHERE highway = 'steps'"
165 cursor.execute(command)
166 db.commit()
167

```

```

168 command = "DELETE FROM result WHERE highway = 'path'"
169 cursor.execute(command)
170 db.commit()
171
172 command = "DELETE FROM result WHERE highway = 'cycleway'"
173 cursor.execute(command)
174 db.commit()
175
176 command = "DELETE FROM result WHERE highway = 'proposed'"
177 cursor.execute(command)
178 db.commit()
179
180 command = "DELETE FROM result WHERE highway = 'construction'"
181 cursor.execute(command)
182 db.commit()
183
184 command = "DELETE FROM result WHERE highway = 'corridor'"
185 cursor.execute(command)
186 db.commit()
187
188 command = "DELETE FROM result WHERE highway = 'residential'"
189 cursor.execute(command)
190 db.commit()
191
192 command = "DELETE FROM result WHERE highway = 'service'"
193 cursor.execute(command)
194 db.commit()
195

```





```

57 # No segundo set de iterações, os atributos dos links são obtidos dos grafos; seus
58 # comprimentos são calculados usando simples relação de Pitágoras com as coordenadas dos
59 # nós, que são dadas em metros
60 print("starting iteration set 2");
61
62 for i in range(len(G.edges())):
63     print("iteration #" + str(i));
64     startnode = G.edges()[i][0]
65     endnode = G.edges()[i][1]
66     startx = G.node[startnode]['coord'][0]
67     starty = G.node[startnode]['coord'][1]
68     endx = G.node[endnode]['coord'][0]
69     endy = G.node[endnode]['coord'][1]
70     length = math.sqrt((endx-startx)*(endx-startx)+(endy-starty)*(endy-starty))
71     maxspeed = float(maxspeed_col[(startnode, endnode)])
72
73     # Criação de SubElementos de links com atributos retirados das listas acima
74     # Capacidade viária é calculada em função do número de faixas de rolamento
75     if (oneway_col[(startnode, endnode)]) == 'yes':
76         if (maxspeed >= 80):
77             ET.SubElement(links, "link", attrib={
78                 'id': str(i),
79                 'from': str(startnode),
80                 'to': str(endnode),
81                 'length': str(length),
82                 'freespeed': "25",
83                 'capacity': str(1900*int(lanes_col[(startnode, endnode)])),
84                 'permlanes': str(lanes_col[(startnode, endnode)]),
85                 'oneway': "1",
86                 'modes': "car"
87             })
88         elif (maxspeed < 80 and maxspeed >= 60):
89             ET.SubElement(links, "link", attrib={
90                 'id': str(i),
91                 'from': str(startnode),
92                 'to': str(endnode),
93                 'length': str(length),
94                 'freespeed': "19.44",
95                 'capacity': str(1500*int(lanes_col[(startnode, endnode)])),
96                 'permlanes': str(lanes_col[(startnode, endnode)]),
97                 'oneway': "1",
98                 'modes': "car"
99             })
100         elif (maxspeed < 60 and maxspeed >= 40):
101             ET.SubElement(links, "link", attrib={
102                 'id': str(i),
103                 'from': str(startnode),
104                 'to': str(endnode),
105                 'length': str(length),
106                 'freespeed': "15.28",
107                 'capacity': str(1200*int(lanes_col[(startnode, endnode)])),
108                 'permlanes': str(lanes_col[(startnode, endnode)]),
109                 'oneway': "1",
110                 'modes': "car"
111             })
112         elif (maxspeed < 40):
113             ET.SubElement(links, "link", attrib={
114                 'id': str(i),
115                 'from': str(startnode),
116                 'to': str(endnode),
117                 'length': str(length),
118                 'freespeed': "15.28",
119                 'capacity': str(1200*int(lanes_col[(startnode, endnode)])),
120                 'permlanes': str(lanes_col[(startnode, endnode)]),
121                 'oneway': "1",
122                 'modes': "car"
123             })

```



```

124
125     else:
126         if (maxspeed>=80):
127             ET.SubElement(links, "link", attrib={
128                 'id': str(i),
129                 'from': str(startnode),
130                 'to': str(endnode),
131                 'length': str(length),
132                 'freespeed': "25",
133                 'capacity': str(950*int(lanes_col[(startnode, endnode)])),
134                 'permlanes':str(lanes_col[(startnode, endnode)]),
135                 'oneway': "1",
136                 'modes': "car"
137             })
138             ET.SubElement(links, "link", attrib={
139                 'id': str(i+1e6),
140                 'from': str(startnode),
141                 'to': str(endnode),
142                 'length': str(length),
143                 'freespeed': "25",
144                 'capacity': str(950*int(lanes_col[(startnode, endnode)])),
145                 'permlanes':str(lanes_col[(startnode, endnode)]),
146                 'oneway': "1",
147                 'modes': "car"
148             })
149         elif (maxspeed<80 and maxspeed>=60):
150             ET.SubElement(links, "link", attrib={
151                 'id': str(i),
152                 'from': str(startnode),
153                 'to': str(endnode),
154                 'length': str(length),
155                 'freespeed': "19.44",
156                 'capacity': str(750*int(lanes_col[(startnode, endnode)])),
157                 'permlanes':str(lanes_col[(startnode, endnode)]),
158                 'oneway': "1",
159                 'modes': "car"
160             })

```

```

161             ET.SubElement(links, "link", attrib={
162                 'id': str(i+1e6),
163                 'from': str(startnode),
164                 'to': str(endnode),
165                 'length': str(length),
166                 'freespeed': "19.44",
167                 'capacity': str(750*int(lanes_col[(startnode, endnode)])),
168                 'permlanes':str(lanes_col[(startnode, endnode)]),
169                 'oneway': "1",
170                 'modes': "car"
171             })
172         elif (maxspeed<60 and maxspeed>=40):
173             ET.SubElement(links, "link", attrib={
174                 'id': str(i),
175                 'from': str(startnode),
176                 'to': str(endnode),
177                 'length': str(length),
178                 'freespeed': "15.27",
179                 'capacity': str(600*int(lanes_col[(startnode, endnode)])),
180                 'permlanes':str(lanes_col[(startnode, endnode)]),
181                 'oneway': "1",
182                 'modes': "car"
183             })
184             ET.SubElement(links, "link", attrib={
185                 'id': str(i+1e6),
186                 'from': str(startnode),
187                 'to': str(endnode),
188                 'length': str(length),
189                 'freespeed': "15.27",
190                 'capacity': str(600*int(lanes_col[(startnode, endnode)])),
191                 'permlanes':str(lanes_col[(startnode, endnode)]),
192                 'oneway': "1",
193                 'modes': "car"
194             })

```

```

195     elif (maxspeed<40):
196         ET.SubElement(links, "link", attrib={
197             'id': str(i),
198             'from': str(startnode),
199             'to': str(endnode),
200             'length': str(length),
201             'freespeed': "15.27",
202             'capacity': str(600*int(lanes_col[(startnode, endnode)])),
203             'permlanes':str(lanes_col[(startnode, endnode)]),
204             'oneway': "1",
205             'modes': "car"
206         })
207         ET.SubElement(links, "link", attrib={
208             'id': str(i+1e6),
209             'from': str(startnode),
210             'to': str(endnode),
211             'length': str(length),
212             'freespeed': "15.27",
213             'capacity': str(600*int(lanes_col[(startnode, endnode)])),
214             'permlanes':str(lanes_col[(startnode, endnode)]),
215             'oneway': "1",
216             'modes': "car"
217         })
218
219     print("finished iteration set 2");
220
221     # Toda a árvore de elementos é salva no formato de um arquivo xml com o cabeçalho
222     # padrão do arquivo network.xml de entrada no MATSim
223     tree = ET.ElementTree(network)
224
225     print("writing file");
226
227     with open('network.xml', 'w') as f:
228         f.write("""<?xml version="1.0" encoding="UTF-8" ?>
229         <!DOCTYPE network SYSTEM "http://www.matsim.org/files/dtd/network_v1.dtd">
230         """)
231         tree.write(f, pretty_print = True)

```

## APÊNDICE G - Categorias de pontos do *OpenStreet Map*

A tabela abaixo foi montada a partir dos dados constantes em OpenStreet Map Wiki (2017b), que se referem às categorias de pontos cadastradas no banco de dados do OpenStreet Map. Para cada categoria da base de dados, foi feita a classificação em ponto de estudo, trabalho ou “outros”. Pontos classificados como locais de trabalho também tiveram um tamanho associado, relacionado à quantidade de postos de trabalho gerados nesse ponto. Por fim, foi atribuído um horário genérico de funcionamento aos locais.

Categoria Original	Subcategoria Original	Ponto de Estudo?	Ponto de "Outros"?	Ponto de Trabalho?	Tamanho do ponto de trabalho	Descrição	Abre	Fecha
aeroway	aerodrome	N	S	S	G	Aeroporto	00:00	23:59
aeroway	terminal	N	S	S	G	Aeroporto	04:00	23:59
aeroway	apron	N	N	N	-	-	-	-
aeroway	gate	N	N	N	-	-	-	-
aeroway	hangar	N	N	N	-	-	-	-
aeroway	helipad	N	N	N	-	-	-	-
aeroway	heliport	N	N	N	-	-	-	-
aeroway	navigationaid	N	N	N	-	-	-	-
aeroway	runway	N	N	N	-	-	-	-
aeroway	taxilane	N	N	N	-	-	-	-
aeroway	taxiway	N	N	N	-	-	-	-
aeroway	windsock	N	N	N	-	-	-	-
amenity	college	S	S	S	MG	Faculdade	08:00	22:30
amenity	university	S	S	S	G	Universidade	08:00	22:30
amenity	arts_centre	N	S	S	P	Geral	09:00	18:00
amenity	bank	N	S	S	P	Banco	11:00	17:00
amenity	bar	N	S	S	P	Restaurante	17:00	23:59
amenity	bbq	N	S	S	P	Restaurante	11:00	23:59
amenity	biergarten	N	S	S	P	Restaurante	17:00	23:59
amenity	blood_donation	N	S	S	P	Geral	08:00	18:00
amenity	bureau_de_change	N	S	S	P	Cambio	08:00	18:00
amenity	cafe	N	S	S	P	Restaurante	09:00	18:00
amenity	car_wash	N	S	S	P	Lava Jato	09:00	18:00
amenity	cinema	N	S	S	P	Cinema	10:00	22:00
amenity	clinic	N	S	S	MP	Saude	08:00	22:00
amenity	community_centre	N	S	S	P	Geral	09:00	18:00
amenity	courthouse	N	S	S	MG	Orgao Publico	08:00	18:00
amenity	coworking_space	N	S	S	MG	Escritorio	08:00	20:00
amenity	dentist	N	S	S	MP	Saude	08:00	18:00
amenity	doctors	N	S	S	MP	Saude	08:00	18:00
amenity	fast_food	N	S	S	P	Restaurante	11:00	23:00
amenity	food_court	N	S	S	MP	Restaurante	11:00	23:00
amenity	fuel	N	S	S	P	Posto de Gasolina	04:00	23:59
amenity	hospital	N	S	S	G	Saude	04:00	23:59
amenity	ice_cream	N	S	S	P	Restaurante	10:00	22:00
amenity	internet_cafe	N	S	S	P	Restaurante	09:00	18:00

amenity	kindergarten	N	S	S	M	Pre-escola	07:30	18:30
amenity	language school	N	S	S	MP	Escola	08:00	20:00
amenity	marketplace	N	S	S	MP	Feira	07:00	17:00
amenity	nursing home	N	S	S	MG	Saude	08:00	18:00
amenity	pharmacy	N	S	S	P	Farmacia	04:00	23:59
amenity	planetarium	N	S	S	P	Lazer	09:00	18:00
amenity	post office	N	S	S	MP	Correios	09:00	18:00
amenity	pub	N	S	S	P	Restaurante	17:00	23:59
amenity	restaurant	N	S	S	MP	Restaurante	11:00	23:00
amenity	school	N	S	S	M	Escola	07:00	18:30
amenity	social centre	N	S	S	P	Geral	09:00	18:00
amenity	social facility	N	S	S	P	Geral	09:00	18:00
amenity	studio	N	S	S	P	Estudio	09:00	18:00
amenity	theatre	N	S	S	P	Teatro	15:00	22:00
amenity	veterinary	N	S	S	P	Veterinaria	09:00	18:00
amenity	biyccle rental	N	S	S	P	Transporte	09:00	18:00
amenity	boat sharing	N	S	S	P	Transporte	09:00	18:00
amenity	car rental	N	S	S	P	Transporte	09:00	18:00
amenity	crematorium	N	S	S	P	Crematorio	08:00	18:00
amenity	driving school	N	S	S	P	Autoescola	09:00	18:00
amenity	embassy	N	S	S	M	Embaixada	09:00	18:00
amenity	ferry terminal	N	S	S	MP	Transporte	04:00	23:59
amenity	fire station	N	S	S	MG	Bombeiros	04:00	23:59
amenity	library	N	S	S	P	Biblioteca	08:00	20:00
amenity	music school	N	S	S	P	Escola	08:00	20:00
amenity	police	N	S	S	MG	Policia	04:00	23:59
amenity	prision	N	S	S	MG	Prisao	04:00	23:59
amenity	townhall	N	S	S	MG	Orgao Publico	08:00	18:00
amenity	water transfer station	N	S	S	P	Saneamento	08:00	18:00
amenity	dive centre	N	S	N	-	Esportes	08:00	20:00
amenity	place of worship	N	S	N	-	Templo religioso	07:00	22:00
amenity	drinking water	N	N	N	-	-	-	-
amenity	animal boarding	N	N	N	-	-	-	-
amenity	animal shelter	N	N	N	-	-	-	-
amenity	atm	N	N	N	-	-	-	-
amenity	baby hatch	N	N	N	-	-	-	-
amenity	baking oven	N	N	N	-	-	-	-
amenity	bench	N	N	N	-	-	-	-
amenity	brothel	N	N	N	-	-	-	-
amenity	bus station	N	N	N	-	-	-	-
amenity	bycicle parking	N	N	N	-	-	-	-
amenity	bycicle repair station	N	N	N	-	-	-	-
amenity	car sharing	N	N	N	-	-	-	-
amenity	casino	N	N	N	-	-	-	-
amenity	charging station	N	N	N	-	-	-	-
amenity	clock	N	N	N	-	-	-	-
amenity	crypt	N	N	N	-	-	-	-
amenity	dojo	N	N	N	-	-	-	-
amenity	fountain	N	N	N	-	-	-	-
amenity	gambling	N	N	N	-	-	-	-
amenity	game feeding	N	N	N	-	-	-	-
amenity	grave yard	N	N	N	-	-	-	-
amenity	grit bin	N	N	N	-	-	-	-
amenity	hunting stand	N	N	N	-	-	-	-
amenity	kneipp water cure	N	N	N	-	-	-	-
amenity	motorcycle parking	N	N	N	-	-	-	-
amenity	nightclub	N	N	N	-	-	-	-
amenity	parking	N	N	N	-	-	-	-
amenity	parking entrance	N	N	N	-	-	-	-
amenity	photo booth	N	N	N	-	-	-	-
amenity	post box	N	N	N	-	-	-	-
amenity	public bookcase	N	N	N	-	-	-	-
amenity	ranger station	N	N	N	-	-	-	-
amenity	recycling	N	N	N	-	-	-	-

amenity	rescue_station	N	N	N	-	-	-	-
amenity	shelter	N	N	N	-	-	-	-
amenity	shower	N	N	N	-	-	-	-
amenity	stripclub	N	N	N	-	-	-	-
amenity	swingerclub	N	N	N	-	-	-	-
amenity	table	N	N	N	-	-	-	-
amenity	taxi	N	N	N	-	-	-	-
amenity	telephone	N	N	N	-	-	-	-
amenity	toilets	N	N	N	-	-	-	-
amenity	vending_machine	N	N	N	-	-	-	-
amenity	waste_basket	N	N	N	-	-	-	-
amenity	waste_disposal	N	N	N	-	-	-	-
amenity	water_point	N	N	N	-	-	-	-
amenity	watering_place	N	N	N	-	-	-	-
barrier	block	N	N	N	-	-	-	-
barrier	bollard	N	N	N	-	-	-	-
barrier	border_control	N	N	N	-	-	-	-
barrier	bump_gate	N	N	N	-	-	-	-
barrier	bus_trap	N	N	N	-	-	-	-
barrier	cattle_grid	N	N	N	-	-	-	-
barrier	chain	N	N	N	-	-	-	-
barrier	cycle_barrier	N	N	N	-	-	-	-
barrier	debris	N	N	N	-	-	-	-
barrier	entrance	N	N	N	-	-	-	-
barrier	full_height_turnstile	N	N	N	-	-	-	-
barrier	gate	N	N	N	-	-	-	-
barrier	hampshire_gate	N	N	N	-	-	-	-
barrier	height_restrictor	N	N	N	-	-	-	-
barrier	horse_stile	N	N	N	-	-	-	-
barrier	jersey_barrier	N	N	N	-	-	-	-
barrier	kent_carriage_gap	N	N	N	-	-	-	-
barrier	kerb	N	N	N	-	-	-	-
barrier	kissing_gate	N	N	N	-	-	-	-
barrier	lift_gate	N	N	N	-	-	-	-
barrier	log	N	N	N	-	-	-	-
barrier	motorcycle_barrier	N	N	N	-	-	-	-
barrier	rope	N	N	N	-	-	-	-
barrier	sally_port	N	N	N	-	-	-	-
barrier	spikes	N	N	N	-	-	-	-
barrier	stile	N	N	N	-	-	-	-
barrier	sump_buster	N	N	N	-	-	-	-
barrier	swing_gate	N	N	N	-	-	-	-
barrier	tank_trap	N	N	N	-	-	-	-
barrier	toll_booth	N	N	N	-	-	-	-
barrier	turnstile	N	N	N	-	-	-	-
building	bakehouse	N	S	S	P	Comercio	09:00	18:00
building	commercial	N	S	S	G	Comercio	09:00	18:00
building	hospital	N	S	S	G	Saude	04:00	23:59
building	industrial	N	S	S	MG	Industria	08:00	18:00
building	kiosk	N	S	S	P	Comercio	09:00	18:00
building	public	N	S	S	MG	Orgao Publico	08:00	18:00
building	retail	N	S	S	G	Comercio	10:00	22:00
building	service	N	S	S	M	Geral	09:00	18:00
building	university	N	S	S	G	Universidade	08:00	22:30
building	warehouse	N	S	S	MP	Comercio	09:00	18:00
building	train_station	N	S	S	M	Transporte	04:00	23:59
building	transportation	N	S	S	M	Transporte	04:00	23:59
building	cathedral	N	S	N	-	Templo religioso	07:00	22:00
building	chapel	N	S	N	-	Templo religioso	07:00	22:00
building	church	N	S	N	-	Templo religioso	07:00	22:00
building	civic	N	S	N	-	Geral	09:00	18:00
building	school	N	S	N	-	Escola	07:30	18:30
building	shrine	N	S	N	-	Templo religioso	07:00	22:00
building	stadium	N	S	N	-	Estadio	18:00	23:00

building	synagogue	N	S	N	-	Templo religioso	07:00	22:00
building	temple	N	S	N	-	Templo religioso	07:00	22:00
building	barn	N	N	N	-	-	-	-
building	bridge	N	N	N	-	-	-	-
building	bunker	N	N	N	-	-	-	-
building	cabin	N	N	N	-	-	-	-
building	carport	N	N	N	-	-	-	-
building	construction	N	N	N	-	-	-	-
building	cowshed	N	N	N	-	-	-	-
building	digester	N	N	N	-	-	-	-
building	farm_auxiliary	N	N	N	-	-	-	-
building	garage	N	N	N	-	-	-	-
building	garages	N	N	N	-	-	-	-
building	greenhouse	N	N	N	-	-	-	-
building	hangar	N	N	N	-	-	-	-
building	hut	N	N	N	-	-	-	-
building	mosque	N	N	N	-	-	-	-
building	roof	N	N	N	-	-	-	-
building	ruins	N	N	N	-	-	-	-
building	shed	N	N	N	-	-	-	-
building	stable	N	N	N	-	-	-	-
building	sty	N	N	N	-	-	-	-
building	transformer_tower	N	N	N	-	-	-	-
craft	agricultural_engines	N	S	S	P	Comercio	09:00	18:00
craft	bakery	N	S	S	P	Comercio	09:00	18:00
craft	basket_market	N	S	S	P	Comercio	09:00	18:00
craft	beekeeper	N	S	S	P	Comercio	09:00	18:00
craft	blacksmith	N	S	S	P	Comercio	09:00	18:00
craft	boat_builder	N	S	S	P	Comercio	09:00	18:00
craft	bookbinder	N	S	S	P	Comercio	09:00	18:00
craft	brewery	N	S	S	P	Comercio	09:00	18:00
craft	builder	N	S	S	P	Comercio	09:00	18:00
craft	carpenter	N	S	S	P	Comercio	09:00	18:00
craft	carpet_layer	N	S	S	P	Comercio	09:00	18:00
craft	caterer	N	S	S	P	Comercio	09:00	18:00
craft	chimney_sweeper	N	S	S	P	Comercio	09:00	18:00
craft	clockmaker	N	S	S	P	Comercio	09:00	18:00
craft	confectionery	N	S	S	P	Comercio	09:00	18:00
craft	distillery	N	S	S	P	Comercio	09:00	18:00
craft	dressmaker	N	S	S	P	Comercio	09:00	18:00
craft	electrician	N	S	S	P	Comercio	09:00	18:00
craft	floorer	N	S	S	P	Comercio	09:00	18:00
craft	gardener	N	S	S	P	Comercio	09:00	18:00
craft	glazery	N	S	S	P	Comercio	09:00	18:00
craft	handicraft	N	S	S	P	Comercio	09:00	18:00
craft	hvac	N	S	S	P	Comercio	09:00	18:00
craft	insulation	N	S	S	P	Comercio	09:00	18:00
craft	jeweller	N	S	S	P	Comercio	09:00	18:00
craft	key_cutter	N	S	S	P	Comercio	09:00	18:00
craft	locksmith	N	S	S	P	Comercio	09:00	18:00
craft	metal_construction	N	S	S	P	Comercio	09:00	18:00
craft	optical	N	S	S	P	Comercio	09:00	18:00
craft	painter	N	S	S	P	Comercio	09:00	18:00
craft	parquet_layer	N	S	S	P	Comercio	09:00	18:00
craft	photographer	N	S	S	P	Comercio	09:00	18:00
craft	photographic_laboratory	N	S	S	P	Comercio	09:00	18:00
craft	piano_tuner	N	S	S	P	Comercio	09:00	18:00
craft	plasterer	N	S	S	P	Comercio	09:00	18:00
craft	plumber	N	S	S	P	Comercio	09:00	18:00
craft	pottery	N	S	S	P	Comercio	09:00	18:00
craft	rigger	N	S	S	P	Comercio	09:00	18:00
craft	roofer	N	S	S	P	Comercio	09:00	18:00
craft	saddler	N	S	S	P	Comercio	09:00	18:00
craft	sailmaker	N	S	S	P	Comercio	09:00	18:00

craft	sawmill	N	S	S	P	Comercio	09:00	18:00
craft	scaffolder	N	S	S	P	Comercio	09:00	18:00
craft	sculptor	N	S	S	P	Comercio	09:00	18:00
craft	shoemaker	N	S	S	P	Comercio	09:00	18:00
craft	stand builder	N	S	S	P	Comercio	09:00	18:00
craft	stonemason	N	S	S	P	Comercio	09:00	18:00
craft	sun protection	N	S	S	P	Comercio	09:00	18:00
craft	tailor	N	S	S	P	Comercio	09:00	18:00
craft	tiler	N	S	S	P	Comercio	09:00	18:00
craft	tinsmith	N	S	S	P	Comercio	09:00	18:00
craft	tuner	N	S	S	P	Comercio	09:00	18:00
craft	upholsterer	N	S	S	P	Comercio	09:00	18:00
craft	watchmaker	N	S	S	P	Comercio	09:00	18:00
craft	window construction	N	S	S	P	Comercio	09:00	18:00
craft	winery	N	S	S	P	Comercio	09:00	18:00
emergency	ambulance station	N	N	N	-	-	-	-
emergency	assembly point	N	N	N	-	-	-	-
emergency	defribillator	N	N	N	-	-	-	-
emergency	fire extinguisher	N	N	N	-	-	-	-
emergency	fire flapper	N	N	N	-	-	-	-
emergency	fire hose	N	N	N	-	-	-	-
emergency	fire hydrant	N	N	N	-	-	-	-
emergency	life ring	N	N	N	-	-	-	-
emergency	lifeguard place	N	N	N	-	-	-	-
emergency	lifeguard plataform	N	N	N	-	-	-	-
emergency	lifeguard tower	N	N	N	-	-	-	-
emergency	ses station	N	N	N	-	-	-	-
emergency	siren	N	N	N	-	-	-	-
emergency	water tank	N	N	N	-	-	-	-
geological	moraine	N	N	N	-	-	-	-
geological	outcrop	N	N	N	-	-	-	-
geological	paleontological site	N	N	N	-	-	-	-
highway	bus stop	N	N	N	-	-	-	-
highway	crossing	N	N	N	-	-	-	-
highway	elevator	N	N	N	-	-	-	-
highway	emergency access point	N	N	N	-	-	-	-
highway	give way	N	N	N	-	-	-	-
highway	mini roundabout	N	N	N	-	-	-	-
highway	motorway junction	N	N	N	-	-	-	-
highway	passing place	N	N	N	-	-	-	-
highway	phone	N	N	N	-	-	-	-
highway	rest area	N	N	N	-	-	-	-
highway	services	N	N	N	-	-	-	-
highway	speed camera	N	N	N	-	-	-	-
highway	stop	N	N	N	-	-	-	-
highway	street lamp	N	N	N	-	-	-	-
highway	traffic signals	N	N	N	-	-	-	-
highway	turning circle	N	N	N	-	-	-	-
historic	aircraft	N	N	N	-	-	-	-
historic	archaeological site	N	N	N	-	-	-	-
historic	battlefield	N	N	N	-	-	-	-
historic	boundary stone	N	N	N	-	-	-	-
historic	building	N	N	N	-	-	-	-
historic	cannon	N	N	N	-	-	-	-
historic	castle	N	N	N	-	-	-	-
historic	city gate	N	N	N	-	-	-	-
historic	citywalls	N	N	N	-	-	-	-
historic	farm auxiliary	N	N	N	-	-	-	-
historic	fort	N	N	N	-	-	-	-
historic	gallows	N	N	N	-	-	-	-
historic	highwater mark	N	N	N	-	-	-	-
historic	locomotive	N	N	N	-	-	-	-
historic	manor	N	N	N	-	-	-	-
historic	memorial	N	N	N	-	-	-	-

historic	milestone	N	N	N	-	-	-	-
historic	monastery	N	N	N	-	-	-	-
historic	monument	N	N	N	-	-	-	-
historic	optical telegraph	N	N	N	-	-	-	-
historic	pillory	N	N	N	-	-	-	-
historic	ruins	N	N	N	-	-	-	-
historic	rune stone	N	N	N	-	-	-	-
historic	ship	N	N	N	-	-	-	-
historic	tomb	N	N	N	-	-	-	-
historic	wayside cross	N	N	N	-	-	-	-
historic	wayside shrine	N	N	N	-	-	-	-
historic	wreck	N	N	N	-	-	-	-
leisure	adult gaming centre	N	S	S	P	Lazer	17:00	22:00
leisure	amusement arcade	N	S	S	P	Lazer	17:00	22:00
leisure	bandstand	N	S	S	P	Lazer	17:00	22:00
leisure	beach resort	N	S	S	MG	Lazer	04:00	23:59
leisure	common	N	S	S	P	Lazer	09:00	18:00
leisure	dance	N	S	S	P	Lazer	17:00	23:59
leisure	fitness centre	N	S	S	P	Lazer	06:00	23:00
leisure	marina	N	S	S	P	Lazer	08:00	18:00
leisure	miniature golf	N	S	S	P	Lazer	17:00	22:00
leisure	water park	N	S	S	MG	Lazer	10:00	17:00
leisure	fishing	N	S	N	-	Pescaria	10:00	17:00
leisure	garden	N	S	N	-	Jardim	09:00	21:00
leisure	golf course	N	S	N	-	Campo de golfe	09:00	18:00
leisure	horse riding	N	S	N	-	Joquei	10:00	22:00
leisure	ice rink	N	S	N	-	Pista de patinacao	10:00	22:00
leisure	nature reserve	N	S	N	-	Reserva natural	04:00	23:59
leisure	park	N	S	N	-	Parque	06:00	22:00
leisure	picnic table	N	S	N	-	Mesa de Piquenique	06:00	22:00
leisure	pitch	N	S	N	-	Campo	06:00	22:00
leisure	playground	N	S	N	-	Playground	06:00	22:00
leisure	slipway	N	S	N	-	Cais	06:00	22:00
leisure	sports centre	N	S	N	-	Centro esportivo	06:00	22:00
leisure	stadium	N	S	N	-	Estadio	18:00	23:00
leisure	swimming pool	N	S	N	-	Piscina	06:00	22:00
leisure	bird hide	N	N	N	-	-	-	-
leisure	dog park	N	N	N	-	-	-	-
leisure	firepit	N	N	N	-	-	-	-
leisure	hackerspace	N	N	N	-	-	-	-
leisure	summer camp	N	N	N	-	-	-	-
leisure	track	N	N	N	-	-	-	-
leisure	wildlife hide	N	N	N	-	-	-	-
man_made	adit	N	N	N	-	-	-	-
man_made	beacon	N	N	N	-	-	-	-
man_made	bunker silo	N	N	N	-	-	-	-
man_made	campanile	N	N	N	-	-	-	-
man_made	chimney	N	N	N	-	-	-	-
man_made	communications tower	N	N	N	-	-	-	-
man_made	crane	N	N	N	-	-	-	-
man_made	cross	N	N	N	-	-	-	-
man_made	dovecote	N	N	N	-	-	-	-
man_made	flagpole	N	N	N	-	-	-	-
man_made	gasometer	N	N	N	-	-	-	-
man_made	hot water tank	N	N	N	-	-	-	-
man_made	kiln	N	N	N	-	-	-	-
man_made	lighthouse	N	N	N	-	-	-	-
man_made	mast	N	N	N	-	-	-	-
man_made	mineshaft	N	N	N	-	-	-	-
man_made	monitoring station	N	N	N	-	-	-	-
man_made	obelisk	N	N	N	-	-	-	-
man_made	observatory	N	N	N	-	-	-	-
man_made	offshore platform	N	N	N	-	-	-	-
man_made	petroleum well	N	N	N	-	-	-	-



man_made	pumping_station	N	N	N	-	-	-	-
man_made	reservoir_covered	N	N	N	-	-	-	-
man_made	silo	N	N	N	-	-	-	-
man_made	snow_fence	N	N	N	-	-	-	-
man_made	snow_net	N	N	N	-	-	-	-
man_made	storage_tank	N	N	N	-	-	-	-
man_made	street_cabinet	N	N	N	-	-	-	-
man_made	surveillance	N	N	N	-	-	-	-
man_made	survey_point	N	N	N	-	-	-	-
man_made	telescope	N	N	N	-	-	-	-
man_made	tower	N	N	N	-	-	-	-
man_made	wastewater_point	N	N	N	-	-	-	-
man_made	water_tap	N	N	N	-	-	-	-
man_made	water_tower	N	N	N	-	-	-	-
man_made	water_well	N	N	N	-	-	-	-
man_made	water_works	N	N	N	-	-	-	-
man_made	watermill	N	N	N	-	-	-	-
man_made	windmill	N	N	N	-	-	-	-
man_made	works	N	N	N	-	-	-	-
military	naval_base	N	S	S	M	Base Militar	08:00	20:00
military	office	N	S	S	M	Escritorio Militar	08:00	18:00
military	airfield	N	N	N	-	-	-	-
military	ammunition	N	N	N	-	-	-	-
military	barracks	N	N	N	-	-	-	-
military	bunker	N	N	N	-	-	-	-
military	checkpoint	N	N	N	-	-	-	-
military	danger_area	N	N	N	-	-	-	-
military	nuclear_explosion_site	N	N	N	-	-	-	-
military	range	N	N	N	-	-	-	-
military	trench	N	N	N	-	-	-	-
natural	bay	N	N	N	-	-	-	-
natural	beach	N	N	N	-	-	-	-
natural	cave_entrance	N	N	N	-	-	-	-
natural	cliff	N	N	N	-	-	-	-
natural	fell	N	N	N	-	-	-	-
natural	geyser	N	N	N	-	-	-	-
natural	moor	N	N	N	-	-	-	-
natural	peak	N	N	N	-	-	-	-
natural	rock	N	N	N	-	-	-	-
natural	saddle	N	N	N	-	-	-	-
natural	sinkhole	N	N	N	-	-	-	-
natural	spring	N	N	N	-	-	-	-
natural	stone	N	N	N	-	-	-	-
natural	tree	N	N	N	-	-	-	-
natural	valley	N	N	N	-	-	-	-
natural	volcano	N	N	N	-	-	-	-
natural	wood	N	N	N	-	-	-	-
office	accountant	N	S	S	M	Escritorio	08:00	18:00
office	adoption_agency	N	S	S	M	Escritorio	08:00	18:00
office	advertising_agency	N	S	S	M	Escritorio	08:00	18:00
office	architect	N	S	S	M	Escritorio	08:00	18:00
office	association	N	S	S	M	Escritorio	08:00	18:00
office	charity	N	S	S	M	Escritorio	08:00	18:00
office	company	N	S	S	M	Escritorio	08:00	18:00
office	educational_institution	N	S	S	MG	Escritorio	08:00	18:00
office	employment_agency	N	S	S	M	Escritorio	08:00	18:00
office	energy_supplier	N	S	S	MG	Escritorio	08:00	18:00
office	estate_agency	N	S	S	MG	Escritorio	08:00	18:00
office	forestry	N	S	S	M	Escritorio	08:00	18:00
office	foundation	N	S	S	MG	Escritorio	08:00	18:00
office	government	N	S	S	MG	Orgao Publico	08:00	18:00
office	guide	N	S	S	M	Escritorio	08:00	18:00
office	insurance	N	S	S	M	Escritorio	08:00	18:00
office	it	N	S	S	M	Escritorio	08:00	18:00

office	lawyer	N	S	S	M	Escritorio	08:00	18:00
office	logistics	N	S	S	M	Escritorio	08:00	18:00
office	moving_company	N	S	S	M	Escritorio	08:00	18:00
office	newspaper	N	S	S	M	Escritorio	08:00	18:00
office	ngo	N	S	S	M	Escritorio	08:00	18:00
office	notary	N	S	S	M	Escritorio	08:00	18:00
office	politycal_party	N	S	S	M	Escritorio	08:00	18:00
office	private_investigator	N	S	S	M	Escritorio	08:00	18:00
office	quango	N	S	S	M	Escritorio	08:00	18:00
office	religion	N	S	S	M	Escritorio	08:00	18:00
office	research	N	S	S	M	Escritorio	08:00	18:00
office	tax	N	S	S	M	Escritorio	08:00	18:00
office	tax_advisor	N	S	S	M	Escritorio	08:00	18:00
office	telecommunication	N	S	S	M	Escritorio	08:00	18:00
office	therapist	N	S	S	M	Escritorio	08:00	18:00
office	travel_agency	N	S	S	M	Escritorio	08:00	18:00
office	water_utility	N	S	S	M	Escritorio	08:00	18:00
place	allotments	N	N	N	-	-	-	-
place	borough	N	N	N	-	-	-	-
place	city	N	N	N	-	-	-	-
place	city_block	N	N	N	-	-	-	-
place	continent	N	N	N	-	-	-	-
place	country	N	N	N	-	-	-	-
place	county	N	N	N	-	-	-	-
place	district	N	N	N	-	-	-	-
place	farm	N	N	N	-	-	-	-
place	hamlet	N	N	N	-	-	-	-
place	islet	N	N	N	-	-	-	-
place	isolated_dwelling	N	N	N	-	-	-	-
place	locality	N	N	N	-	-	-	-
place	municipality	N	N	N	-	-	-	-
place	neighbourhood	N	N	N	-	-	-	-
place	plot	N	N	N	-	-	-	-
place	province	N	N	N	-	-	-	-
place	quarter	N	N	N	-	-	-	-
place	region	N	N	N	-	-	-	-
place	square	N	N	N	-	-	-	-
place	state	N	N	N	-	-	-	-
place	suburb	N	N	N	-	-	-	-
place	town	N	N	N	-	-	-	-
place	village	N	N	N	-	-	-	-
power	catenary_mast	N	N	N	-	-	-	-
power	compensator	N	N	N	-	-	-	-
power	converter	N	N	N	-	-	-	-
power	generator	N	N	N	-	-	-	-
power	heliostat	N	N	N	-	-	-	-
power	insulator	N	N	N	-	-	-	-
power	pole	N	N	N	-	-	-	-
power	substation	N	N	N	-	-	-	-
power	switch	N	N	N	-	-	-	-
power	terminal	N	N	N	-	-	-	-
power	tower	N	N	N	-	-	-	-
power	transformer	N	N	N	-	-	-	-
public_transport	platform	N	N	N	-	-	-	-
public_transport	station	N	N	N	-	-	-	-
public_transport	stop_area	N	N	N	-	-	-	-
public_transport	stop_position	N	N	N	-	-	-	-
railway	buffer_stop	N	N	N	-	-	-	-
railway	crossing	N	N	N	-	-	-	-
railway	derail	N	N	N	-	-	-	-
railway	halt	N	N	N	-	-	-	-
railway	level_crossing	N	N	N	-	-	-	-
railway	platform	N	N	N	-	-	-	-
railway	railway_crossing	N	N	N	-	-	-	-

railway	signal	N	N	N	-	-	-	-
railway	stop position	N	N	N	-	-	-	-
railway	subway entrance	N	N	N	-	-	-	-
railway	switch	N	N	N	-	-	-	-
railway	tram stop	N	N	N	-	-	-	-
railway	traverser	N	N	N	-	-	-	-
railway	turntable	N	N	N	-	-	-	-
shop	agrarian	N	S	S	MP	Comercio	09:00	18:00
shop	alcohol	N	S	S	MP	Comercio	09:00	18:00
shop	anime	N	S	S	MP	Comercio	09:00	18:00
shop	antiques	N	S	S	MP	Comercio	09:00	18:00
shop	art	N	S	S	MP	Comercio	09:00	18:00
shop	baby goods	N	S	S	MP	Comercio	09:00	18:00
shop	bag	N	S	S	MP	Comercio	09:00	18:00
shop	bakery	N	S	S	MP	Comercio	09:00	18:00
shop	bathroom furnishing	N	S	S	MP	Comercio	09:00	18:00
shop	beauty	N	S	S	MP	Comercio	09:00	18:00
shop	bed	N	S	S	MP	Comercio	09:00	18:00
shop	beverages	N	S	S	MP	Comercio	09:00	18:00
shop	bicycle	N	S	S	MP	Comercio	09:00	18:00
shop	bookmaker	N	S	S	MP	Comercio	09:00	18:00
shop	books	N	S	S	MP	Comercio	09:00	18:00
shop	boutique	N	S	S	MP	Comercio	09:00	18:00
shop	brewing supplies	N	S	S	MP	Comercio	09:00	18:00
shop	butcher	N	S	S	MP	Comercio	09:00	18:00
shop	camera	N	S	S	MP	Comercio	09:00	18:00
shop	candles	N	S	S	MP	Comercio	09:00	18:00
shop	car parts	N	S	S	MP	Comercio	09:00	18:00
shop	car repair	N	S	S	MP	Comercio	09:00	18:00
shop	carpet	N	S	S	MP	Comercio	09:00	18:00
shop	carpet	N	S	S	MP	Comercio	09:00	18:00
shop	charity	N	S	S	MP	Comercio	09:00	18:00
shop	cheese	N	S	S	MP	Comercio	09:00	18:00
shop	chemist	N	S	S	MP	Comercio	09:00	18:00
shop	chocolate	N	S	S	MP	Comercio	09:00	18:00
shop	clothes	N	S	S	MP	Comercio	09:00	18:00
shop	coffee	N	S	S	MP	Comercio	09:00	18:00
shop	collector	N	S	S	MP	Comercio	09:00	18:00
shop	computer	N	S	S	MP	Comercio	09:00	18:00
shop	confectionery	N	S	S	MP	Comercio	09:00	18:00
shop	convenience	N	S	S	MP	Comercio	09:00	18:00
shop	copyshop	N	S	S	MP	Comercio	09:00	18:00
shop	cosmetics	N	S	S	MP	Comercio	09:00	18:00
shop	craft	N	S	S	MP	Comercio	09:00	18:00
shop	curtain	N	S	S	MP	Comercio	09:00	18:00
shop	dairy	N	S	S	MP	Comercio	09:00	18:00
shop	deli	N	S	S	MP	Comercio	09:00	18:00
shop	department store	N	S	S	MP	Comercio	09:00	18:00
shop	doityourself	N	S	S	MP	Comercio	09:00	18:00
shop	dry cleaning	N	S	S	MP	Comercio	09:00	18:00
shop	e-cigarette	N	S	S	MP	Comercio	09:00	18:00
shop	electrical	N	S	S	MP	Comercio	09:00	18:00
shop	electronics	N	S	S	MP	Comercio	09:00	18:00
shop	energy	N	S	S	MP	Comercio	09:00	18:00
shop	erotic	N	S	S	MP	Comercio	09:00	18:00
shop	fabric	N	S	S	MP	Comercio	09:00	18:00
shop	farm	N	S	S	MP	Comercio	09:00	18:00
shop	fashion	N	S	S	MP	Comercio	09:00	18:00
shop	fireplace	N	S	S	MP	Comercio	09:00	18:00
shop	fishing	N	S	S	MP	Comercio	09:00	18:00
shop	florist	N	S	S	MP	Comercio	09:00	18:00
shop	frame	N	S	S	MP	Comercio	09:00	18:00
shop	free flying	N	S	S	MP	Comercio	09:00	18:00
shop	fuel	N	S	S	MP	Comercio	09:00	18:00

shop	funeral_directors	N	S	S	MP	Comercio	09:00	18:00
shop	furniture	N	S	S	MP	Comercio	09:00	18:00
shop	games	N	S	S	MP	Comercio	09:00	18:00
shop	garden_centre	N	S	S	MP	Comercio	09:00	18:00
shop	garden_furniture	N	S	S	MP	Comercio	09:00	18:00
shop	gas	N	S	S	MP	Comercio	09:00	18:00
shop	general	N	S	S	MP	Comercio	09:00	18:00
shop	gift	N	S	S	MP	Comercio	09:00	18:00
shop	glazery	N	S	S	MP	Comercio	09:00	18:00
shop	greengrocer	N	S	S	MP	Comercio	09:00	18:00
shop	hairdresser	N	S	S	MP	Comercio	09:00	18:00
shop	hairdresser_supply	N	S	S	MP	Comercio	09:00	18:00
shop	hardware	N	S	S	MP	Comercio	09:00	18:00
shop	hearing_aids	N	S	S	MP	Comercio	09:00	18:00
shop	herbalist	N	S	S	MP	Comercio	09:00	18:00
shop	hifi	N	S	S	MP	Comercio	09:00	18:00
shop	houseware	N	S	S	MP	Comercio	09:00	18:00
shop	hunting	N	S	S	MP	Comercio	09:00	18:00
shop	ice_cream	N	S	S	MP	Comercio	09:00	18:00
shop	interior_decoration	N	S	S	MP	Comercio	09:00	18:00
shop	jewelry	N	S	S	MP	Comercio	09:00	18:00
shop	kiosk	N	S	S	MP	Comercio	09:00	18:00
shop	kitchen	N	S	S	MP	Comercio	09:00	18:00
shop	lamps	N	S	S	MP	Comercio	09:00	18:00
shop	laundry	N	S	S	MP	Comercio	09:00	18:00
shop	leather	N	S	S	MP	Comercio	09:00	18:00
shop	locksmith	N	S	S	MP	Comercio	09:00	18:00
shop	lottery	N	S	S	MP	Comercio	09:00	18:00
shop	mall	N	S	S	MP	Comercio	09:00	18:00
shop	massage	N	S	S	MP	Comercio	09:00	18:00
shop	medical_supply	N	S	S	MP	Comercio	09:00	18:00
shop	mobile_phone	N	S	S	MP	Comercio	09:00	18:00
shop	model	N	S	S	MP	Comercio	09:00	18:00
shop	money_lender	N	S	S	MP	Comercio	09:00	18:00
shop	motorcycle	N	S	S	MP	Comercio	09:00	18:00
shop	music	N	S	S	MP	Comercio	09:00	18:00
shop	musical_instrument	N	S	S	MP	Comercio	09:00	18:00
shop	newsagent	N	S	S	MP	Comercio	09:00	18:00
shop	nutrition_supplements	N	S	S	MP	Comercio	09:00	18:00
shop	optician	N	S	S	MP	Comercio	09:00	18:00
shop	outdoor	N	S	S	MP	Comercio	09:00	18:00
shop	paint	N	S	S	MP	Comercio	09:00	18:00
shop	pasta	N	S	S	MP	Comercio	09:00	18:00
shop	pastry	N	S	S	MP	Comercio	09:00	18:00
shop	pawnbroker	N	S	S	MP	Comercio	09:00	18:00
shop	perfumery	N	S	S	MP	Comercio	09:00	18:00
shop	pet	N	S	S	MP	Comercio	09:00	18:00
shop	photo	N	S	S	MP	Comercio	09:00	18:00
shop	pyrotechnics	N	S	S	MP	Comercio	09:00	18:00
shop	radiotechnics	N	S	S	MP	Comercio	09:00	18:00
shop	religion	N	S	S	MP	Comercio	09:00	18:00
shop	scuba_diving	N	S	S	MP	Comercio	09:00	18:00
shop	seafood	N	S	S	MP	Comercio	09:00	18:00
shop	second_hand	N	S	S	MP	Comercio	09:00	18:00
shop	security	N	S	S	MP	Comercio	09:00	18:00
shop	shoes	N	S	S	MP	Comercio	09:00	18:00
shop	spices	N	S	S	MP	Comercio	09:00	18:00
shop	sports	N	S	S	MP	Comercio	09:00	18:00
shop	stationary	N	S	S	MP	Comercio	09:00	18:00
shop	supermarket	N	S	S	MP	Comercio	09:00	18:00
shop	swimming_pool	N	S	S	MP	Comercio	09:00	18:00
shop	tailor	N	S	S	MP	Comercio	09:00	18:00
shop	tattoo	N	S	S	MP	Comercio	09:00	18:00
shop	tea	N	S	S	MP	Comercio	09:00	18:00

shop	ticket	N	S	S	MP	Comercio	09:00	18:00
shop	tiles	N	S	S	MP	Comercio	09:00	18:00
shop	tobacco	N	S	S	MP	Comercio	09:00	18:00
shop	toys	N	S	S	MP	Comercio	09:00	18:00
shop	trade	N	S	S	MP	Comercio	09:00	18:00
shop	travel_agency	N	S	S	MP	Comercio	09:00	18:00
shop	trophy	N	S	S	MP	Comercio	09:00	18:00
shop	tyres	N	S	S	MP	Comercio	09:00	18:00
shop	vacant	N	S	S	MP	Comercio	09:00	18:00
shop	vacuum_cleaner	N	S	S	MP	Comercio	09:00	18:00
shop	variety_store	N	S	S	MP	Comercio	09:00	18:00
shop	video	N	S	S	MP	Comercio	09:00	18:00
shop	video_games	N	S	S	MP	Comercio	09:00	18:00
shop	watches	N	S	S	MP	Comercio	09:00	18:00
shop	weapons	N	S	S	MP	Comercio	09:00	18:00
shop	window_blind	N	S	S	MP	Comercio	09:00	18:00
sports	10pin	N	S	S	P	Esportes	08:00	20:00
sports	9pin	N	S	S	P	Esportes	08:00	20:00
sports	aikido	N	S	S	P	Esportes	08:00	20:00
sports	american_football	N	S	S	P	Esportes	08:00	20:00
sports	archery	N	S	S	P	Esportes	08:00	20:00
sports	athletics	N	S	S	P	Esportes	08:00	20:00
sports	australian_football	N	S	S	P	Esportes	08:00	20:00
sports	badminton	N	S	S	P	Esportes	08:00	20:00
sports	bandy	N	S	S	P	Esportes	08:00	20:00
sports	base	N	S	S	P	Esportes	08:00	20:00
sports	baseball	N	S	S	P	Esportes	08:00	20:00
sports	basketball	N	S	S	P	Esportes	08:00	20:00
sports	beachvolleyball	N	S	S	P	Esportes	08:00	20:00
sports	billiards	N	S	S	P	Esportes	08:00	20:00
sports	bmw	N	S	S	P	Esportes	08:00	20:00
sports	bobsleigh	N	S	S	P	Esportes	08:00	20:00
sports	boules	N	S	S	P	Esportes	08:00	20:00
sports	bowls	N	S	S	P	Esportes	08:00	20:00
sports	boxing	N	S	S	P	Esportes	08:00	20:00
sports	canadian_football	N	S	S	P	Esportes	08:00	20:00
sports	canoe	N	S	S	P	Esportes	08:00	20:00
sports	chess	N	S	S	P	Esportes	08:00	20:00
sports	cliff_diving	N	S	S	P	Esportes	08:00	20:00
sports	climbing	N	S	S	P	Esportes	08:00	20:00
sports	climbing_adventure	N	S	S	P	Esportes	08:00	20:00
sports	cockfighting	N	S	S	P	Esportes	08:00	20:00
sports	cricket	N	S	S	P	Esportes	08:00	20:00
sports	croquet	N	S	S	P	Esportes	08:00	20:00
sports	curling	N	S	S	P	Esportes	08:00	20:00
sports	cycling	N	S	S	P	Esportes	08:00	20:00
sports	darts	N	S	S	P	Esportes	08:00	20:00
sports	dog_racing	N	S	S	P	Esportes	08:00	20:00
sports	equestrian	N	S	S	P	Esportes	08:00	20:00
sports	fencing	N	S	S	P	Esportes	08:00	20:00
sports	field_hockey	N	S	S	P	Esportes	08:00	20:00
sports	free_flying	N	S	S	P	Esportes	08:00	20:00
sports	gaelic_games	N	S	S	P	Esportes	08:00	20:00
sports	golf	N	S	S	P	Esportes	08:00	20:00
sports	gymnastics	N	S	S	P	Esportes	08:00	20:00
sports	handball	N	S	S	P	Esportes	08:00	20:00
sports	hapkido	N	S	S	P	Esportes	08:00	20:00
sports	horse_racing	N	S	S	P	Esportes	08:00	20:00
sports	horseshoes	N	S	S	P	Esportes	08:00	20:00
sports	ice_hockey	N	S	S	P	Esportes	08:00	20:00
sports	ice_skating	N	S	S	P	Esportes	08:00	20:00
sports	ice_stock	N	S	S	P	Esportes	08:00	20:00
sports	judo	N	S	S	P	Esportes	08:00	20:00
sports	karting	N	S	S	P	Esportes	08:00	20:00

sports	kitesurfing	N	S	S	P	Esportes	08:00	20:00
sports	korfball	N	S	S	P	Esportes	08:00	20:00
sports	model_aerodrome	N	S	S	P	Esportes	08:00	20:00
sports	motocross	N	S	S	P	Esportes	08:00	20:00
sports	motor	N	S	S	P	Esportes	08:00	20:00
sports	multi	N	S	S	P	Esportes	08:00	20:00
sports	netball	N	S	S	P	Esportes	08:00	20:00
sports	obstacle_course	N	S	S	P	Esportes	08:00	20:00
sports	orienteering	N	S	S	P	Esportes	08:00	20:00
sports	paddle_tennis	N	S	S	P	Esportes	08:00	20:00
sports	parachuting	N	S	S	P	Esportes	08:00	20:00
sports	paragliding	N	S	S	P	Esportes	08:00	20:00
sports	pelota	N	S	S	P	Esportes	08:00	20:00
sports	racquet	N	S	S	P	Esportes	08:00	20:00
sports	rc_car	N	S	S	P	Esportes	08:00	20:00
sports	roller_skating	N	S	S	P	Esportes	08:00	20:00
sports	rowing	N	S	S	P	Esportes	08:00	20:00
sports	rugby_league	N	S	S	P	Esportes	08:00	20:00
sports	rugby_union	N	S	S	P	Esportes	08:00	20:00
sports	running	N	S	S	P	Esportes	08:00	20:00
sports	sailing	N	S	S	P	Esportes	08:00	20:00
sports	scuba_diving	N	S	S	P	Esportes	08:00	20:00
sports	shooting	N	S	S	P	Esportes	08:00	20:00
sports	skateboard	N	S	S	P	Esportes	08:00	20:00
sports	soccer	N	S	S	P	Esportes	08:00	20:00
sports	sumo	N	S	S	P	Esportes	08:00	20:00
sports	surfing	N	S	S	P	Esportes	08:00	20:00
sports	swimming	N	S	S	P	Esportes	08:00	20:00
sports	table_soccer	N	S	S	P	Esportes	08:00	20:00
sports	table_tennis	N	S	S	P	Esportes	08:00	20:00
sports	tennis	N	S	S	P	Esportes	08:00	20:00
sports	toboggan	N	S	S	P	Esportes	08:00	20:00
sports	volleyball	N	S	S	P	Esportes	08:00	20:00
sports	water_polo	N	S	S	P	Esportes	08:00	20:00
sports	water_ski	N	S	S	P	Esportes	08:00	20:00
sports	weightlifting	N	S	S	P	Esportes	08:00	20:00
sports	wrestling	N	S	S	P	Esportes	08:00	20:00
sports	yoga	N	S	S	P	Esportes	08:00	20:00
tourism	aquarium	N	S	S	MP	Aquario	10:00	22:00
tourism	gallery	N	S	S	MP	Galeria	10:00	22:00
tourism	museum	N	S	S	MP	Museu	10:00	22:00
tourism	theme_park	N	S	S	M	Parque	10:00	22:00
tourism	zoo	N	S	S	MG	Zoologico	09:00	17:00
tourism	attraction	N	S	S	M	Atracao Turistica	09:00	18:00
tourism	hostel	N	S	S	MP	Hostel	04:00	23:59
tourism	hotel	N	S	S	M	Hotel	04:00	23:59
tourism	information	N	S	S	MP	Informacoes	09:00	21:00
tourism	motel	N	S	S	MP	Motel	04:00	23:59
tourism	picnic_site	N	S	N	-	Area de piquenique	06:00	22:00
tourism	alpine_hut	N	N	N	-	-	-	-
tourism	apartment	N	N	N	-	-	-	-
tourism	artwork	N	N	N	-	-	-	-
tourism	camp_site	N	N	N	-	-	-	-
tourism	caravan_site	N	N	N	-	-	-	-
tourism	chalet	N	N	N	-	-	-	-
tourism	guest_house	N	N	N	-	-	-	-
tourism	viewpoint	N	N	N	-	-	-	-
tourism	wilderness_hut	N	N	N	-	-	-	-
waterway	boatyard	N	N	N	-	-	-	-
waterway	culvert	N	N	N	-	-	-	-
waterway	dock	N	N	N	-	-	-	-
waterway	fuel	N	N	N	-	-	-	-
waterway	lock_gate	N	N	N	-	-	-	-
waterway	turning_point	N	N	N	-	-	-	-

waterway	water_point	N	N	N	-	-	-	-
waterway	waterfall	N	N	N	-	-	-	-
waterway	weir	N	N	N	-	-	-	-

## APÊNDICE H - Rotina para organização do banco de dados de pontos

No computador de especificações apresentadas no Apêndice D, a rotina a seguir, em Python, levou 1,9 segundos para ser executada, organizando o banco de dados dos pontos de realização de atividades, originalmente com 5.217 instâncias.

```
1 # Bibliotecas para Python que permitem a leitura de arquivos CSV (Comma Separated Values)
2 # e bases de dados SQLite
3 import csv
4 import sqlite3
5
6 print("starting program...")
7
8 # Definindo os caminhos ate os arquivos que devem ser lidos
9 path_csv = 'points.csv';
10 path_sqlite = 'points.sqlite';
11
12 # Ordem das colunas do arquivo CSV
13 osm_idCol=2
14 other_tagsCol=11
15
16 # Criando a classe "Pontos" com todos os seus atributos
17 class PontosClass:
18     osm_id=''
19     categ=''
20     subcateg=''
21
22 # Criando de um vetor dentro do programa com as informacoes do arquivo CSV
23 vetorpontos = []
24
25 print("connecting to csv...")
26
27 # Conectando com o arquivo CSV
28 csvfile = open(path_csv, 'rt')
29 print("csv connected!")
30
31 print("reading csv file...")
32 next(csvfile) # Pulando a primeira linha do arquivo
33 csvreader = csv.reader(csvfile, delimiter=',', quotechar='"')
34
35 register = 0;
36 for row in csvreader:
37     print("running csv row " + str(register))
38     register = register+1
39
40     newEntry = PontosClass()
41     newEntry.osm_id = row[osm_idCol]
42
43     other_tags = row[other_tagsCol].replace("=>", ",")
44     other_tags = other_tags.replace('"', '')
45     other_tags = other_tags.split(',')
46
```



```

47     i=0
48     while(i<len(other_tags)): # Separando a coluna other_tags
49         if other_tags[i]=='aeroway':
50             newEntry.categ = other_tags[i]
51             newEntry.subcateg = other_tags[i+1]
52         elif other_tags[i]=='amenity':
53             newEntry.categ = other_tags[i]
54             newEntry.subcateg = other_tags[i+1]
55         elif other_tags[i]=='boundary':
56             newEntry.categ = other_tags[i]
57             newEntry.subcateg = other_tags[i+1]
58         elif other_tags[i]=='building':
59             newEntry.categ = other_tags[i]
60             newEntry.subcateg = other_tags[i+1]
61         elif other_tags[i]=='craft':
62             newEntry.categ = other_tags[i]
63             newEntry.subcateg = other_tags[i+1]
64         elif other_tags[i]=='leisure':
65             newEntry.categ = other_tags[i]
66             newEntry.subcateg = other_tags[i+1]
67         elif other_tags[i]=='military':
68             newEntry.categ = other_tags[i]
69             newEntry.subcateg = other_tags[i+1]
70         elif other_tags[i]=='office':
71             newEntry.categ = other_tags[i]
72             newEntry.subcateg = other_tags[i+1]
73         elif other_tags[i]=='shop':
74             newEntry.categ = other_tags[i]
75             newEntry.subcateg = other_tags[i+1]
76         elif other_tags[i]=='sport':
77             newEntry.categ = other_tags[i]
78             newEntry.subcateg = other_tags[i+1]
79         elif other_tags[i]=='tourism':
80             newEntry.categ = other_tags[i]
81             newEntry.subcateg = other_tags[i+1]
82         i=i+2
83     vetorpontos.append(newEntry)
84
85     print("end reading csv")
86

```

```

87 # Criando a classe "PontosMod" com todos os seus atributos
88 class PontosModClass:
89     osm_id=''
90     descricao=''
91     estudo=''
92     outros=''
93     trabalho=''
94     tamtrabalho=''
95     abre=''
96     fecha=''
97     tipo=''
98
99 # Criando de um vetor dentro do programa com as informacoes do arquivo CSV
100 vetorpontosmod = []
101
102 # Inserindo os valores do vetor "vetorpontos", criado anteriormente, nesse novo vetor
103 # pontosmod
104 register=0
105 for row in vetorpontos:
106     print("running    vetorpontos row" + str(register))
107     register = register+1
108
109     if row.categ!='':
110         osm_id=row.osm_id
111         categoria=row.categ
112         subcategoria=row.subcateg
113

```

```

114     # print('opening categorias.csv...')
115     csvfile = open('categorias.csv', 'rt')
116     # print('categorias.csv is opened!')
117     next(csvfile) # Pulando a primeira linha do arquivo
118     csvreader = csv.reader(csvfile, delimiter=';')
119
120     for row in csvreader:
121         if row[0]==categoria:
122             if row[1]==subcategoria:
123                 newEntry = PontosModClass()
124                 newEntry.osm_id=osm_id
125                 newEntry.descricao=row[6]
126                 newEntry.estudo=row[2]
127                 newEntry.outros=row[3]
128                 newEntry.trabalho=row[4]
129                 newEntry.tamtrabalho=row[5]
130                 abrestring = row[7]
131                 if abrestring != '':
132                     abrestring = abrestring.split(':')
133                     abrehora = abrestring[0]
134                     abremin = abrestring[1]
135                     newEntry.abre = str((float(abrehora)*60+float(abremin))*60)
136                     fechastring = row[8]
137                     fechastring = fechastring.split(':')
138                     fechahora = fechastring[0]
139                     fechamin = fechastring[1]
140                     newEntry.fecha = str((float(fechahora)*60+float(fechamin))*60)
141                 else:
142                     newEntry.abre = ''
143                     newEntry.fecha = ''
144                 newEntry.tipo =row[9]
145                 vetorpontosmod.append(newEntry)
146                 #print(newEntry)
147             else:
148                 print('sem categoria')
149
150     print("end creating pontosmod")
151

```

```

152 # Conectando com a base de dados SQLite
153 db = sqlite3.connect(path_sqlite)
154 cursor = db.cursor();
155
156 # Criando uma nova tabela "pontos" no banco de dados
157 createcommand = """CREATE TABLE IF NOT EXISTS pontos(osm_id INTEGER NOT NULL PRIMARY KEY
158                 AUTOINCREMENT, descricao VARCHAR(255), estudo VARCHAR(255),
159                 outros VARCHAR(255), trabalho VARCHAR(255), tamtrabalho VARCHAR(255),
160                 abre VARCHAR(255), fecha VARCHAR(255), tipo VARCHAR);"""
161
162 cursor.execute(createcommand)
163 db.commit()
164
165 # Inserindo os valores do vetor "pontosmod", criado anteriormente, nessa tabela do
166 # banco de dados
167 i=0
168 for row in vetorpontosmod:
169     if row.descricao!='':
170         insertcommand = """INSERT INTO pontos(osm_id,descricao,estudo,outros,trabalho,
171         tamtrabalho,abre,fecha,tipo) VALUES (" +row.osm_id +",\"\" + row.descricao + "\",
172         \"\" + row.estudo + "\",\"\" + row.outros + "\",\"\" + row.trabalho + "\",\"\" +
173         row.tamtrabalho + "\",\"\" + row.abre + "\",\"\" + row.fecha + "\",\"\" + row.tipo+
174         "\",\"\";"""
175         cursor.execute(insertcommand)
176         db.commit()
177         print ("running      " + str(i)) # Exibindo na tela qual o elemento iterado
178         i=i+1
179

```

```

180 print("end_insert")
181
182 # Mesclando a tabela "pontosmod" com a tabela original do banco de dados (renomeada para
183 # "temp"), em uma nova tabela de nome "result"
184 command = "ALTER TABLE pontos RENAME TO temp;"
185 cursor.execute(command)
186 db.commit()
187
188 command = """CREATE TABLE result AS SELECT temp.OGC_FID,
189         temp.GEOMETRY,temp.osm_id,temp.name,pontos.descricao,pontos.estudo,
190         pontos.outros,pontos.trabalho,pontos.tamtrabalho, pontos.abre, pontos.fecha,
191         pontos.tipo
192         FROM temp INNER JOIN pontos ON temp.osm_id = pontos.osm_id;"""
193 cursor.execute(command)
194 db.commit()
195
196 # Apagando a tabela "temp"
197 command = "DROP TABLE temp;"
198 cursor.execute(command)
199 db.commit()
200
201 # Apagando a tabela "pontos"
202 command = "DROP TABLE pontos;"
203 cursor.execute(command)
204 db.commit()
205
206 command = "ALTER TABLE result RENAME TO points;"
207 cursor.execute(command)
208 db.commit()
209
210 # Mensagem de fim do programa
211 print("end")

```

## APÊNDICE I - Rotina para conversão do arquivo Shapefile dos pontos em dado de entrada no MATSim

No computador de especificações apresentadas no Apêndice D, a rotina a seguir, em Python, levou 0,6 segundos para ser executada.

```
1 # Biblioteca para Python que permite a leitura de arquivos CSV (Comma Separated Values)
2 import csv
3
4 # Definindo os caminhos ate o arquivo que deve ser lido
5 path_csv = 'points.csv';
6
7 # Ordem das colunas do arquivo CSV
8 X_Col = 0
9 Y_Col = 1
10 osm_idCol=3
11 nameCol=4
12 descricaoCol=5
13 estudoCol=6
14 outrosCol=7
15 trabalhoCol=8
16 tamtrabalhoCol=9
17 abreCol=10
18 fechaCol=11
19 tipoCol=12
20
21 # Criando a classe "Pontos" com todos os seus atributos
22 class Pontos:
23     osm_id=''
24     x=''
25     y=''
26     abre=''
27     fecha=''
28     tipo=''
29
30 # Conectando com o arquivo CSV
31 origcsv = open(path_csv, 'rt')
32
33 next(origcsv) # Pulando a primeira linha do arquivo
34
35 csvreader = csv.reader(origcsv, delimiter=',', quotechar='')
36
37 # Criando o txt de destino para os pontos de estudo
38 estudotxt = open('estudo.txt', 'wt')
39
40 estudotxt.writelines('osm_id\tx\ty\tabre\tfecha\ttipo\n')
41
42 for row in csvreader:
43     print(row[estudoCol])
44     if row[estudoCol]!='S':
45         newEstudo = Pontos()
46         newEstudo.osm_id=row[osm_idCol]
47         newEstudo.x=row[X_Col]
48         newEstudo.y=row[Y_Col]
49         newEstudo.abre=row[abreCol]
50         newEstudo.fecha = row[fechaCol]
51         newEstudo.tipo = row[tipoCol]
52         estudotxt.writelines(newEstudo.osm_id + '\t' + newEstudo.x + '\t' + newEstudo.y +
53         '\t' + newEstudo.abre + '\t' + newEstudo.fecha + '\t' + newEstudo.tipo + '\n');
```

```

55 # Conectando com o arquivo CSV
56 origcsv = open(path_csv, 'rt')
57
58 next(origcsv) # Pulando a primeira linha do arquivo
59 csvreader = csv.reader(origcsv, delimiter=',')
60
61 # Criando o txt de destino para os pontos de estudo
62 outrostxt = open('outros.txt', 'wt')
63
64 outrostxt.writelines('osm_id\tx\ty\tabre\tfecha\ttipo\n')
65
66 for row in csvreader:
67     if row[outrosCol]=='S':
68         newOutros = Pontos()
69         newOutros.osm_id=row[osm_idCol]
70         newOutros.x=row[X_Col]
71         newOutros.y=row[Y_Col]
72         newOutros.abre=row[abreCol]
73         newOutros.fecha = row[fechaCol]
74         newOutros.tipo = row[tipoCol]
75         outrostxt.writelines(newOutros.osm_id + '\t' + newOutros.x + '\t' + newOutros.y +
76                               '\t' + newOutros.abre + '\t' + newOutros.fecha + '\t' + newOutros.tipo + '\n');
77
78 # Conectando com o arquivo CSV
79 origcsv = open(path_csv, 'rt')
80
81 next(origcsv) # Pulando a primeira linha do arquivo
82 csvreader = csv.reader(origcsv, delimiter=',')
83
84 # Criando o txt de destino para os pontos de estudo
85 trabalhotxt = open('trabalho.txt', 'wt')
86
87 trabalhotxt.writelines('osm_id\tx\ty\tabre\tfecha\ttipo\n')
88
89 for row in csvreader:
90     if row[trabalhoCol]=='S':
91
92         contador = 0
93
94         # Estabelecendo proporções para o número de postos de trabalho em cada ponto
95         if row[tamtrabalhoCol]=='P':
96             prob=1
97         elif row[tamtrabalhoCol]=='MP':
98             prob=2
99         elif row[tamtrabalhoCol]=='M':
100            prob=5
101         elif row[tamtrabalhoCol]=='MG':
102            prob=10
103         elif row[tamtrabalhoCol]=='G':
104            prob=20
105
106         newTrab = Pontos()
107         newTrab.osm_id=row[osm_idCol]
108         newTrab.x=row[X_Col]
109         newTrab.y=row[Y_Col]
110         newTrab.abre=row[abreCol]
111         newTrab.fecha = row[fechaCol]
112         newTrab.tipo = row[tipoCol]
113
114         # Repetindo as linhas com informações de pontos de trabalho de acordo com as
115         # proporções estabelecidas para cada tamanho
116         while contador<prob:
117             trabalhotxt.writelines(newTrab.osm_id + '\t' + newTrab.x + '\t' + newTrab.y +
118                                   '\t' + newTrab.abre + '\t' + newTrab.fecha + '\t' + newTrab.tipo + '\n');
119             contador=contador+1
120
121 print("end")

```

## APÊNDICE J - Rotina para geração do arquivo *population.xml*

No computador de especificações apresentadas no Apêndice D, a rotina a seguir, em *Java*, levou 149 segundos para ser executada.

```
1 package mycodes;
2
3 // Importando bibliotecas a serem usadas
4 import java.io.FileNotFoundException;
5 import java.io.FileReader;
6 import java.util.HashMap;
7 import java.util.Map;
8 import java.util.Random;
9 import java.util.Scanner;
10
11 // Importando bibliotecas do MATSim
12 import org.matsim.api.core.v01.Coord;
13 import org.matsim.api.core.v01.Id;
14 import org.matsim.api.core.v01.Scenario;
15 import org.matsim.api.core.v01.population.Activity;
16 import org.matsim.api.core.v01.population.Leg;
17 import org.matsim.api.core.v01.population.Person;
18 import org.matsim.api.core.v01.population.Plan;
19 import org.matsim.api.core.v01.population.PopulationWriter;
20 import org.matsim.core.config.ConfigUtils;
21 import org.matsim.core.gbl.MatsimRandom;
22 import org.matsim.core.network.MatsimNetworkReader;
23 import org.matsim.core.scenario.ScenarioUtils;
24 import org.matsim.core.utils.geometry.geotools.MGC;
25 import org.matsim.core.utils.gis.ShapeFileReader;
26 import org.opengis.feature.simple.SimpleFeature;
27
28 import com.vividsolutions.jts.geom.Geometry;
29 import com.vividsolutions.jts.geom.GeometryFactory;
30 import com.vividsolutions.jts.geom.Point;
31 import com.vividsolutions.jts.io.ParseException;
32 import com.vividsolutions.jts.io.WKTReader;
33
34 // Iniciando a classe
35 public class CreateDemand {
36
37     // Caminhos para os diretórios dos arquivos de entrada
38     private static final String NETWORKFILE = "input/network.xml";
39     private static final String REGION = "input/ZONA2008.shp";
40     private static final String WORK = "input/trabalho.txt";
41     private static final int WORKQUANT = 13863;
42     private static final String STUDY = "input/estudo.txt";
43     private static final int STUDYQUANT = 53;
44     private static final String OTHER = "input/outros.txt";
45     private static final int OTHERQUANT = 4866;
46     private static final String CENSUS = "input/frota.txt";
47
48     // Diretorio de saida
49     private static final String PLANSFILEOUTPUT = "output/plans.xml";
50
```



```

51     private static double SCALEFACTOR = 0.1;
52
53     // Criando cenário
54     private Scenario scenario;
55     private Map<String,Geometry> shapeMap;
56
57     // Criando a população
58     CreateDemand (){
59         this.scenario = ScenarioUtils.createScenario(ConfigUtils.createConfig());
60         new MatsimNetworkReader(scenario.getNetwork()).readFile(NETWORKFILE);
61     }
62
63     // Função "Main"
64     private void run() throws Exception {
65         // Assign shapefiles and txts to maps
66         this.shapeMap = readShapeFile(REGION, "RA");
67         String[][] workpoints = readfacilities(WORK,WORKQUANT);
68         String[][] studypoints = readfacilities(STUDY, STUDYQUANT);
69         String[][] otherpoints = readfacilities(OTHER,OTHERQUANT);
70
71         int regions = 39;
72         String frota[][] = readcensus(CENSUS,regions);
73
74         // Iterando em cada RA
75         for(int x=1;x<regions;x++){
76             double frotaRa = Double.parseDouble(frota[x][1]);
77             double commuters = 0.55*frotaRa*SCALEFACTOR;
78             double workonly = commuters*0.52;
79             double studyonly = commuters*0.078;
80             double studyandwork = commuters*0.117;
81             double noactivity = commuters*0.28;
82             Geometry home = this.shapeMap.get(Integer.toString(x));
83
84             // Criando pessoas que apenas trabalham
85             for (int a = 1; a<=workonly;a++){
86                 String mode = "car";
87                 // Sorteando ponto de moradia dentro do mapa da RA
88                 Coord homec = drawRandomPointFromGeometry(home);
89                 Random rndline = new Random();
90                 // Sorteando linha no arquivo de texto de pontos de trabalho
91                 int workline = rndline.nextInt(WORKQUANT-1)+1;
92                 // Registrando as coordenadas do ponto de trabalho
93                 double xCoord = Double.parseDouble(workpoints[workline][1]);
94                 double yCoord = Double.parseDouble(workpoints[workline][2]);
95                 // Registrando os horários de funcionamento do ponto de trabalho
96                 double wopentime = Double.parseDouble(workpoints[workline][3]);
97                 double wclosetime = Double.parseDouble(workpoints[workline][4]);
98                 String wtype = workpoints[workline][5];
99                 Coord workc = new Coord(xCoord,yCoord);
100                // Sorteando ponto de atividade do tipo "outros"
101                int otherline = rndline.nextInt(OTHERQUANT-1)+1;
102                // Registrando coordenadas da atividade do tipo "outros"
103                xCoord = Double.parseDouble(otherpoints[otherline][1]);
104                yCoord = Double.parseDouble(otherpoints[otherline][2]);
105                String otype = otherpoints[otherline][5];
106                Coord otherc = new Coord(xCoord, yCoord);
107
108                // Criando pessoa com as características definidas
109                createWorkOnly(x, a, homec, workc, wopentime, wclosetime, wtype,
110                    otherc, otype, mode, "work");
111            }
112        }
113

```

```

114 // Criando pessoas que apenas estudam; procedimento semelhante
115 for (int a = 1; a<=studyonly;a++){
116     String mode = "car";
117     // Sorteando ponto de moradia dentro do mapa da RA
118     Coord homec = drawRandomPointFromGeometry(home);
119     Random rndline = new Random();
120     // Sorteando linha no arquivo de texto de pontos de estudo
121     int studyline = rndline.nextInt(STUDYQUANT-1)+1;
122     // Registrando coordenadas do ponto de estudo
123     Double xCoord = Double.parseDouble(studypoints[studyline][1]);
124     Double yCoord = Double.parseDouble(studypoints[studyline][2]);
125     Coord studyc = new Coord(xCoord,yCoord);
126     // Sorteando ponto de atividade do tipo "outros"
127     int otherline = rndline.nextInt(OTHERQUANT-1)+1;
128     // Registrando coordenadas da atividade do tipo "outros"
129     xCoord = Double.parseDouble(otherpoints[otherline][1]);
130     yCoord = Double.parseDouble(otherpoints[otherline][2]);
131     String otype = otherpoints[otherline][5];
132     Coord otherc = new Coord(xCoord, yCoord);
133
134     // Criando pessoa com as características definidas
135     createStudyOnly(x, a, homec, studyc, otherc, otype, mode, "study");
136
137 }
138
139 // Criando pessoas que estudam e trabalham
140 for (int a = 1; a<=studyandwork;a++){
141     String mode = "car";
142     // Sorteando ponto de moradia dentro do mapa da RA
143     Coord homec = drawRandomPointFromGeometry(home);
144     Random rndline = new Random();
145     // Sorteando linha no arquivo de texto de pontos de estudo
146     int studyline = rndline.nextInt(STUDYQUANT-1)+1;
147     // Registrando coordenadas do ponto de estudo
148     Double xCoord = Double.parseDouble(studypoints[studyline][1]);
149     Double yCoord = Double.parseDouble(studypoints[studyline][2]);
150     Coord studyc = new Coord(xCoord,yCoord);
151     // Sorteando linha no arquivo de texto de pontos de trabalho
152     int workline = rndline.nextInt(WORKQUANT-1)+1;
153     // Registrando coordenadas do ponto de trabalho
154     xCoord = Double.parseDouble(workpoints[workline][1]);
155     yCoord = Double.parseDouble(workpoints[workline][2]);
156     // Registrando horários de funcionamento do ponto de trabalho
157     Double wopentime = Double.parseDouble(workpoints[workline][3]);
158     Double wclosetime = Double.parseDouble(workpoints[workline][4]);
159     String wtype = workpoints[workline][5];
160     Coord workc = new Coord(xCoord,yCoord);
161     // Sorteando linha no arquivo de texto de pontos do tipo "outros"
162     int otherline = rndline.nextInt(OTHERQUANT-1)+1;
163     // Registrando coordenadas do ponto do tipo "outros"
164     xCoord = Double.parseDouble(otherpoints[otherline][1]);
165     yCoord = Double.parseDouble(otherpoints[otherline][2]);
166     String otype = otherpoints[otherline][5];
167     Coord otherc = new Coord(xCoord, yCoord);
168
169     // Criando pessoa com as características definidas
170     createStudyAndWork(x, a, homec, workc, wopentime, wclosetime, wtype,
171         studyc, otherc, otype, mode, "studyandwork");
172 }
173

```



```

174 // Criando pessoas sem atividade (nem trabalha nem estuda)
175 for (int a = 1; a<=noactivity;a++){
176     String mode = "car";
177     // Sorteando ponto de moradia dentro do mapa da RA
178     Coord homec = drawRandomPointFromGeometry(home);
179     Random rndline = new Random();
180     // Sorteando linha no arquivo de texto de pontos do tipo "outros"
181     int otherline = rndline.nextInt(OTHERQUANT-1)+1;
182     // Registrando coordenadas do ponto do tipo "outros"
183     Double xCoord = Double.parseDouble(otherpoints[otherline][1]);
184     Double yCoord = Double.parseDouble(otherpoints[otherline][2]);
185     String otype = otherpoints[otherline][5];
186     Coord otherc = new Coord(xCoord, yCoord);
187
188     // Criando pessoa com as características definidas
189     createNoActivity(x, a, homec, otherc, otype, mode, "noactiv");
190 }
191
192 // Adicionando pessoa criada ao cenário
193 PopulationWriter pw = new PopulationWriter(scenario.getPopulation(),
194     scenario.getNetwork());
195 pw.write(PLANSFILEOUTPUT);
196 }
197 }
198
199 // Método para ler a tabela com dados censitários e transformar em matrizes
200 private String[][] readcensus(String filePath,int n){
201     String matrix[][] = new String[n][2];
202     try {
203         Scanner input = new Scanner(new FileReader(filePath));
204
205         for(int i=0;i<n;i++){
206             for(int j=0;j<2;j++){
207                 matrix[i][j]=input.next();
208             }
209         }
210     }
211
212     catch (FileNotFoundException e) {
213         // TODO Auto-generated catch block
214         e.printStackTrace();
215     }
216     return matrix;
217 }
218
219 // Método para ler os arquivos de texto com dados dos pontos de realização
220 // de atividades e transformar em matrizes
221 private String[][] readfacilities(String filePath,int n){
222     String matrix[][] = new String[n][6];
223     try {
224         Scanner input = new Scanner(new FileReader(filePath));
225
226         for(int i=0;i<n;i++){
227             for(int j=0;j<6;j++){
228                 matrix[i][j]=input.next();
229             }
230         }
231     }
232     catch (FileNotFoundException e) {
233         // TODO Auto-generated catch block
234         e.printStackTrace();
235     }
236     return matrix;
237 }

```

```

238 // Método para criar pessoas que só trabalham
239 private void createWorkOnly(int regadm, int i, Coord coordHome, Coord coordWork,
240                             Double wopentime, Double wclosetime, String wtype,
241                             Coord coordOther, String otype, String mode, String
242                             type)
243 {
244     // A identificação da pessoa é a sua RA de moradia, o seu tipo de atividade e
245     // um número inteiro
246     Id<Person> personId = Id.createPersonId(regadm+type+i);
247     Person person = scenario.getPopulation().getFactory().createPerson(personId);
248
249     Random rnd = new Random();
250     Double wopenperiod = wclosetime - wopentime;
251
252     //Criando atividades comuns a todos os planos
253     Activity work = scenario.getPopulation().getFactory()
254                     .createActivityFromCoord("w"+wtype, coordWork);
255     if (wopenperiod <=10*60*60){
256         work.setStartTime(wopentime);
257         work.setEndTime(wclosetime);
258     }else if (wopenperiod > 10*60*60 && wopenperiod<=16*60*60){
259         int shift = rnd.nextInt(1);
260         work.setStartTime(wopentime+(shift*wopenperiod/2));
261         work.setEndTime(wopentime+(shift+1)*wopenperiod/2);
262     }else{
263         int shift = rnd.nextInt(2);
264         work.setStartTime(wopentime+(shift*wopenperiod/3));
265         work.setEndTime(wopentime+(shift+1)*wopenperiod/3);
266     }
267
268     // Criando a primeira atividade do dia: Home
269     Activity home = scenario.getPopulation().getFactory()
270                     .createActivityFromCoord("home", coordHome);
271
272     //Criando plan1 = H-W-H
273     Plan plan1 = scenario.getPopulation().getFactory().createPlan();
274
275     Activity home1 = scenario.getPopulation().getFactory()
276                     .createActivityFromCoord("home", coordHome);
277     // Atribuindo horário de saída de casa como 40 min antes da próxima atividade
278     home1.setEndTime(work.getStartTime()-40*60);
279
280     // Adicionando atividades em sequência dentro desse plano
281     plan1.addActivity(home1);
282     Leg hw = scenario.getPopulation().getFactory().createLeg(mode);
283     plan1.addLeg(hw);
284     plan1.addActivity(work);
285     Leg wh = scenario.getPopulation().getFactory().createLeg(mode);
286     plan1.addLeg(wh);
287     plan1.addActivity(home);
288
289     // Adicionando o plano à memória deste agente
290     person.addPlan(plan1);
291
292     //Criando plan2 = H-0-W-H
293     Plan plan2 = scenario.getPopulation().getFactory().createPlan();
294
295     Activity other1 = scenario.getPopulation().getFactory()
296                     .createActivityFromCoord("o"+otype, coordOther);
297     // A atividade "outro" acaba 40min antes do horário de trabalho e
298     // dura 60min
299     other1.setEndTime(work.getStartTime()-40*60);
300     other1.setStartTime(other1.getEndTime()-40*60);
301

```

```

302 Activity home2 = scenario.getPopulation().getFactory()
303             .createActivityFromCoord("home", coordHome);
304 home2.setEndTime(other1.getStartTime()-60*60);
305
306 // Adicionando as atividades em sequência dentro desse plano
307 plan2.addActivity(home2);
308 Leg ho = scenario.getPopulation().getFactory().createLeg(mode);
309 plan2.addLeg(ho);
310 plan2.addActivity(other1);
311 Leg ow = scenario.getPopulation().getFactory().createLeg(mode);
312 plan2.addLeg(ow);
313 plan2.addActivity(work);
314 plan2.addLeg(wh);
315 plan2.addActivity(home);
316
317 // Adicionando o plano à memória desse agente
318 person.addPlan(plan2);
319
320 //Criando plan3 = H-W-0-H
321 Plan plan3 = scenario.getPopulation().getFactory().createPlan();
322
323 Activity home3 = scenario.getPopulation().getFactory()
324             .createActivityFromCoord("home", coordHome);
325 // Atribuindo horário de saída de casa como 40 min antes da próxima atividade
326 home3.setEndTime(work.getStartTime()-40*60);
327 // Atividade do tipo "outro" não tem horário de início mas acaba 2h depois
328 // do fim da atividade de trabalho
329 Activity other2 = scenario.getPopulation().getFactory()
330             .createActivityFromCoord("o"+otype, coordOther);
331 other2.setEndTime(work.getEndTime()+(2*60*60));
332
333 // Adicionando as atividades em sequência dentro desse plano
334 plan3.addActivity(home3);
335 plan3.addLeg(hw);
336 plan3.addActivity(work);
337 Leg wo = scenario.getPopulation().getFactory().createLeg(mode);
338 plan3.addLeg(wo);
339 plan3.addActivity(other2);
340 Leg oh = scenario.getPopulation().getFactory().createLeg(mode);
341 plan3.addLeg(oh);
342 plan3.addActivity(home);
343
344 // Adicionando o plano à memória desse agente
345 person.addPlan(plan3);
346
347 //Criando plan4 = H-W-0-W-H
348 Plan plan4 = scenario.getPopulation().getFactory().createPlan();
349 // A atividade de trabalho é dividida em duas partes
350 Activity work1 = scenario.getPopulation().getFactory()
351             .createActivityFromCoord("w"+wtype, coordWork);
352 Activity work2 = scenario.getPopulation().getFactory()
353             .createActivityFromCoord("w"+wtype, coordWork);

```

```

354     Double lunchtime;
355     // O horário de almoço depende da quantidade de horas trabalhadas,
356     // que é função do horário total de funcionamento do estabelecimento
357     // já que se funcionar por muitas horas, são criados turnos de trabalho
358     if (wopenperiod <=10*60*60){
359         work1.setStartTime(wopentime);
360         lunchtime = wopentime+(wopenperiod-2)/2;
361         work1.setEndTime(lunchtime);
362         work2.setStartTime(lunchtime+(2*60*60));
363         work2.setEndTime((2*lunchtime)+(2*60*60)-wopentime);
364     }else if (wopenperiod > 10*60*60 && wopenperiod<=16*60*60){
365         int shift = rnd.nextInt(1);
366         work1.setStartTime(wopentime+(shift*(wopenperiod-1)/2));
367         lunchtime = wopentime+(shift+1)*(wopenperiod-1)/2;
368         work1.setEndTime(lunchtime);
369         work2.setStartTime(lunchtime+(2*60*60));
370         work2.setEndTime((2*lunchtime)+(2*60*60)-wopentime);
371     }else{
372         int shift = rnd.nextInt(2);
373         work1.setStartTime(wopentime+(shift*(wopenperiod-1)/3));
374         lunchtime = wopentime+(shift+1)*(wopenperiod-1)/3;
375         work1.setEndTime(lunchtime);
376         work2.setStartTime(lunchtime+(2*60*60));
377         work2.setEndTime((2*lunchtime)+(2*60*60)-wopentime);
378     }
379     // Atribuindo atividade de "outros" entre as atividades de trabalho
380     Activity other3 = scenario.getPopulation().getFactory()
381         .createActivityFromCoord("o"+otype, coordOther);
382     other3.setStartTime(lunchtime+(30*60));
383     other3.setEndTime(lunchtime+(30*60)+60*60);
384
385     // Adicionando as atividades em sequência dentro desse plano
386     plan4.addActivity(home1);
387     plan4.addLeg(hw);
388     plan4.addActivity(work1);
389     plan4.addLeg(wo);
390     plan4.addActivity(other3);
391     plan4.addLeg(ow);
392     plan4.addActivity(work2);
393     plan4.addLeg(wh);
394     plan4.addActivity(home);
395
396     // Adicionando plano à memória desse agente
397     person.addPlan(plan4);
398
399     // Atribuindo probabilidades na eleição de cada plano como ativo para
400     // cada agente
401     Double prob = rnd.nextDouble()*100;
402
403     if (prob<=68.17){
404         person.setSelectedPlan(plan1);
405     } else if (prob>68.17 && prob <=78.36) {
406         person.setSelectedPlan(plan2);
407     } else if (prob>78.36 && prob<=88.55) {
408         person.setSelectedPlan(plan3);
409     } else {
410         person.setSelectedPlan(plan4);
411     }
412
413     // Adicionando o agente criado ao cenário
414     scenario.getPopulation().addPerson(person);
415 }
416

```

```

417 // Método para criar pessoas que apenas estudam
418 private void createStudyOnly(int regadm, int i, Coord coordHome,
419                             Coord coordStudy, Coord coordOther,
420                             String otype, String mode, String type)
421 {
422     Id<Person> personId = Id.createPersonId(regadm+type+i);
423     Person person = scenario.getPopulation().getFactory().createPerson(personId);
424     Random rnd = new Random();
425
426     //Criando atividades comuns a todos os planos
427     Activity study = scenario.getPopulation().getFactory()
428         .createActivityFromCoord("study", coordStudy);
429     Double shift = rnd.nextDouble();
430     // Sorteando, com probabilidades iguais, o turno do estudo
431     // (manhã, tarde ou noite)
432     if (shift<=0.33){
433         study.setStartTime(8*60*60);
434         study.setEndTime(12*60*60);
435     } else if (shift>0.33 && shift <= 0.66) {
436         study.setStartTime(14*60*60);
437         study.setEndTime(18*60*60);
438     } else {
439         study.setStartTime(19*60*60);
440         study.setEndTime(23*60*60);
441     }
442
443     Activity home = scenario.getPopulation().getFactory()
444         .createActivityFromCoord("home", coordHome);
445
446     //Criando plan1 = H-S-H
447     Plan plan1 = scenario.getPopulation().getFactory().createPlan();
448
449     Activity home1 = scenario.getPopulation().getFactory()
450         .createActivityFromCoord("home", coordHome);
451     // A pessoa sai de casa 40min antes do início da próxima atividade
452     home1.setEndTime(study.getStartTime()-40*60);
453
454
455     // Adicionando as atividades em sequência dentro desse plano
456     plan1.addActivity(home1);
457     Leg hs = scenario.getPopulation().getFactory().createLeg(mode);
458     plan1.addLeg(hs);
459     plan1.addActivity(study);
460     Leg sh = scenario.getPopulation().getFactory().createLeg(mode);
461     plan1.addLeg(sh);
462     plan1.addActivity(home);
463
464     // Adicionando plano à memória desse agente
465     person.addPlan(plan1);
466
467     //Criando plan2 = H-0-S-H
468     Plan plan2 = scenario.getPopulation().getFactory().createPlan();
469
470     Activity other1 = scenario.getPopulation().getFactory()
471         .createActivityFromCoord("o"+otype, coordOther);
472     // A atividade do tipo "outros" termina 40min antes do início da atividade
473     // estudo e tem duração de 1 hora
474     other1.setEndTime(study.getStartTime()-40*60);
475     other1.setStartTime(other1.getEndTime()-60*60);
476
477     Activity home2 = scenario.getPopulation().getFactory()
478         .createActivityFromCoord("home", coordHome);
479     // A pessoa sai de casa 40min antes da atividade seguinte começar
480     home2.setEndTime(other1.getStartTime()-40*60);

```



```

481 // Adicionando as atividades em sequência dentro desse plano
482 plan2.addActivity(home2);
483 Leg ho = scenario.getPopulation().getFactory().createLeg(mode);
484 plan2.addLeg(ho);
485 plan2.addActivity(other1);
486 Leg os = scenario.getPopulation().getFactory().createLeg(mode);
487 plan2.addLeg(os);
488 plan2.addActivity(study);
489 plan2.addLeg(sh);
490 plan2.addActivity(home);
491
492 // Adicionando plano à memória desse agente
493 person.addPlan(plan2);
494
495 //Criando plan3 = H-S-0-H
496 Plan plan3 = scenario.getPopulation().getFactory().createPlan();
497
498 Activity home3 = scenario.getPopulation().getFactory()
499     .createActivityFromCoord("home", coordHome);
500 // A pessoa sai de casa 40min antes do início da próxima atividade
501 home3.setEndTime(study.getStartTime()-40*60);
502
503 Activity other2 = scenario.getPopulation().getFactory()
504     .createActivityFromCoord("o"+otype, coordOther);
505 // A atividade do tipo "outros" termina 2h depois do término da atividade
506 // de estudo
507 other2.setEndTime(study.getEndTime()+(2*60*60));
508
509 // Adicionando as atividades em sequência dentro desse plano
510 plan3.addActivity(home3);
511 plan3.addLeg(hs);
512 plan3.addActivity(study);
513 Leg so = scenario.getPopulation().getFactory().createLeg(mode);
514 plan3.addLeg(so);
515 plan3.addActivity(other2);
516 Leg oh = scenario.getPopulation().getFactory().createLeg(mode);
517 plan3.addLeg(oh);
518 plan3.addActivity(home);
519
520 // Adicionando plano à memória desse agente
521 person.addPlan(plan3);
522
523 //Criando plan4 = H-S-0-S-H
524 Plan plan4 = scenario.getPopulation().getFactory().createPlan();
525
526 // Para a criação desse plano, assume-se que a pessoa estuda em ambos os
527 // turnos, com intervalo para almoço
528 Activity home4 = scenario.getPopulation().getFactory()
529     .createActivityFromCoord("home", coordHome);
530 home4.setEndTime(8*60*60-40*60);
531
532 Activity study1 = scenario.getPopulation().getFactory()
533     .createActivityFromCoord("study", coordStudy);
534 study1.setStartTime(8*60*60);
535 study1.setEndTime(12*60*60);
536
537 Activity study2 = scenario.getPopulation().getFactory()
538     .createActivityFromCoord("study", coordStudy);
539 study2.setStartTime(14*60*60);
540 study2.setEndTime(18*60*60);
541

```

```

542     Activity other3 = scenario.getPopulation().getFactory()
543         .createActivityFromCoord("o"+otype, coordOther);
544     other3.setStartTime(12*60*60+15*60);
545     other3.setEndTime(13*60*60+45*60);
546
547     // Adicionando as atividades em sequência dentro desse plano
548     plan4.addActivity(home4);
549     plan4.addLeg(hs);
550     plan4.addActivity(study1);
551     plan4.addLeg(so);
552     plan4.addActivity(other3);
553     plan4.addLeg(os);
554     plan4.addActivity(study2);
555     plan4.addLeg(sh);
556     plan4.addActivity(home);
557
558     // Adicionando plano à memória desse agente
559     person.addPlan(plan4);
560
561     // Atribuindo probabilidades de eleição de cada plano como sendo o ativo
562     // para cada indivíduo
563     Double prob = rnd.nextDouble()*100;
564
565     if (prob<=36.44){
566         person.setSelectedPlan(plan1);
567     } else if (prob>36.44 && prob <=49.18) {
568         person.setSelectedPlan(plan2);
569     } else if (prob>49.18 && prob<=61.93) {
570         person.setSelectedPlan(plan3);
571     } else {
572         person.setSelectedPlan(plan4);
573     }
574
575     scenario.getPopulation().addPerson(person);
576 }
577
578 // Método para criar pessoas que estudam e trabalham
579 private void createStudyAndWork(int regadm, int i, Coord coordHome, Coord coordWork,
580     Double wopentime, Double wcloseTime, String wtype,
581     Coord coordStudy, Coord coordOther, String otype,
582     String mode, String type)
583 {
584     Id<Person> personId = Id.createPersonId(regadm+type+i);
585     Person person = scenario.getPopulation().getFactory().createPerson(personId);
586     Random rnd = new Random();
587     Double wopenperiod = wcloseTime - wopentime;
588
589     //Criando atividades comuns a todos os planos: a atividade de trabalho tem
590     // características fixas
591     Activity work = scenario.getPopulation().getFactory()
592         .createActivityFromCoord("w"+wtype, coordWork);
593     if (wopenperiod <=10*60*60){
594         work.setStartTime(wopentime);
595         work.setEndTime(wcloseTime);
596     }else if (wopenperiod > 10*60*60 && wopenperiod<=16*60*60){
597         int shift = rnd.nextInt(1);
598         work.setStartTime(wopentime+(shift*wopenperiod/2));
599         work.setEndTime(wopentime+(shift+1)*wopenperiod/2);
600     }else{
601         int shift = rnd.nextInt(2);
602         work.setStartTime(wopentime+(shift*wopenperiod/3));
603         work.setEndTime(wopentime+(shift+1)*wopenperiod/3);
604     }
605
606     Activity home = scenario.getPopulation().getFactory()
607         .createActivityFromCoord("home", coordHome);
608

```

```

609 //Criando plan1 = H-S-W-H
610 Plan plan1 = scenario.getPopulation().getFactory().createPlan();
611
612 Activity study1 = scenario.getPopulation().getFactory()
613     .createActivityFromCoord("study", coordStudy);
614 // Nesse plano, o estudo é no turno da manhã, e o trabalho à tarde
615 study1.setStartTime(8*60*60);
616 study1.setEndTime(12*60*60);
617
618 Activity home1 = scenario.getPopulation().getFactory()
619     .createActivityFromCoord("home", coordHome);
620 // A pessoa sai de casa 40min antes do início da atividade seguinte
621 home1.setEndTime(study1.getStartTime()-40*60);
622
623 // Adicionando as atividades em sequência dentro desse plano
624 plan1.addActivity(home1);
625 Leg hs = scenario.getPopulation().getFactory().createLeg(mode);
626 plan1.addLeg(hs);
627 plan1.addActivity(study1);
628 Leg sw = scenario.getPopulation().getFactory().createLeg(mode);
629 plan1.addLeg(sw);
630 plan1.addActivity(work);
631 Leg wh = scenario.getPopulation().getFactory().createLeg(mode);
632 plan1.addLeg(wh);
633 plan1.addActivity(home);
634
635 // Adicionando plano à memória desse agente
636 person.addPlan(plan1);
637
638 //Criando plan2 = H-W-S-H
639 Plan plan2 = scenario.getPopulation().getFactory().createPlan();
640
641 // Nesse plano, o estudo é no turno da noite
642 Activity study2 = scenario.getPopulation().getFactory()
643     .createActivityFromCoord("study", coordOther);
644 study2.setStartTime(19*60*60);
645 study2.setEndTime(23*60*60);
646
647 Activity home2 = scenario.getPopulation().getFactory()
648     .createActivityFromCoord("home", coordHome);
649 home2.setEndTime(work.getStartTime()-40*60);
650
651 // Adicionando as atividades em sequência dentro desse plano
652 plan2.addActivity(home2);
653 Leg hw = scenario.getPopulation().getFactory().createLeg(mode);
654 plan2.addLeg(hw);
655 plan2.addActivity(work);
656 Leg ws = scenario.getPopulation().getFactory().createLeg(mode);
657 plan2.addLeg(ws);
658 plan2.addActivity(study2);
659 Leg sh = scenario.getPopulation().getFactory().createLeg(mode);
660 plan2.addLeg(sh);
661 plan2.addActivity(home);
662
663 // Adicionando plano à memória desse agente
664 person.addPlan(plan2);
665
666 //Criando plan3 = H-S-0-W-H
667 Plan plan3 = scenario.getPopulation().getFactory().createPlan();
668
669 Activity home3 = scenario.getPopulation().getFactory()
670     .createActivityFromCoord("home", coordHome);
671 home3.setEndTime(study1.getStartTime()-40*60);
672
673 Activity other1 = scenario.getPopulation().getFactory()
674     .createActivityFromCoord("o"+otype, coordOther);
675 other1.setEndTime(work.getStartTime()-(60*60));
676

```



```

677 // Adicionando as atividades em sequência dentro desse plano
678 plan3.addActivity(home3);
679 plan3.addLeg(hs);
680 plan3.addActivity(study1);
681 Leg so = scenario.getPopulation().getFactory().createLeg(mode);
682 plan3.addLeg(so);
683 plan3.addActivity(other1);
684 Leg ow = scenario.getPopulation().getFactory().createLeg(mode);
685 plan3.addLeg(ow);
686 plan3.addActivity(work);
687 plan3.addLeg(wh);
688 plan3.addActivity(home);
689
690 // Adicionando plano à memória desse agente
691 person.addPlan(plan3);
692
693 //Criando plan4 = H-W-0-S-H
694 Plan plan4 = scenario.getPopulation().getFactory().createPlan();
695
696 Activity home4 = scenario.getPopulation().getFactory()
697     .createActivityFromCoord("home", coordHome);
698 home4.setEndTime(work.getStartTime()-40*60);
699
700 Activity other2 = scenario.getPopulation().getFactory()
701     .createActivityFromCoord("o"+otype, coordOther);
702 other2.setEndTime(study2.getStartTime()-40*60);
703
704 // Adicionando as atividades em sequência dentro desse plano
705 plan4.addActivity(home4);
706 plan4.addLeg(hw);
707 plan4.addActivity(work);
708 Leg wo = scenario.getPopulation().getFactory().createLeg(mode);
709 plan4.addLeg(wo);
710 plan4.addActivity(other2);
711 Leg os = scenario.getPopulation().getFactory().createLeg(mode);
712 plan4.addLeg(os);
713 plan4.addActivity(study2);
714 plan4.addLeg(sh);
715 plan4.addActivity(home);
716
717 // Adicionando plano à memória desse agente
718 person.addPlan(plan4);
719
720 // Definindo probabilidades para eleição de cada plano como sendo o ativo
721 // para cada agente
722 Double prob = rnd.nextDouble()*100;
723
724 if (prob<=32.89){
725     person.setSelectedPlan(plan1);
726 } else if (prob>32.89 && prob <=65.77) {
727     person.setSelectedPlan(plan2);
728 } else if (prob>65.77 && prob<=82.89) {
729     person.setSelectedPlan(plan3);
730 } else {
731     person.setSelectedPlan(plan4);
732 }
733
734 // Adicionando o agente criado ao cenário
735 scenario.getPopulation().addPerson(person);
736 }
737

```

```

738 // Criando pessoa sem atividade principal (apenas atividade do tipo "outros")
739 private void createNoActivity(int regadm, int i, Coord coordHome, Coord coordOther,
740                               String otype, String mode, String type)
741 {
742     Id<Person> personId = Id.createPersonId(regadm+type+i);
743     Person person = scenario.getPopulation().getFactory().createPerson(personId);
744     Random rnd = new Random();
745
746     //Criando unico plano = H-0-H
747     Plan plan = scenario.getPopulation().getFactory().createPlan();
748
749     Activity home1 = scenario.getPopulation().getFactory()
750                       .createActivityFromCoord("home", coordHome);
751     home1.setEndTime((rnd.nextInt(10)+8)*60*60);
752
753     Activity other = scenario.getPopulation().getFactory()
754                      .createActivityFromCoord("o"+otype, coordOther);
755     other.setEndTime(home1.getEndTime()+4*60*60);
756
757     Activity home = scenario.getPopulation().getFactory()
758                     .createActivityFromCoord("home", coordHome);
759
760     // Adicionando as atividades em sequência dentro desse plano
761     plan.addActivity(home1);
762     Leg ho = scenario.getPopulation().getFactory().createLeg(mode);
763     plan.addLeg(ho);
764     plan.addActivity(other);
765     Leg oh = scenario.getPopulation().getFactory().createLeg(mode);
766     plan.addLeg(oh);
767     plan.addActivity(home);
768
769     // Adicionando plano à memória do agente e definindo como "ativo"
770     person.addPlan(plan);
771     person.setSelectedPlan(plan);
772
773
774     // Adicionando agente criado ao cenário
775     scenario.getPopulation().addPerson(person);
776 }
777
778 // Método para sortear ponto aleatório dentro de uma geometria
779 private Coord drawRandomPointFromGeometry(Geometry g) {
780     Random rnd = MatsimRandom.getLocalInstance();
781     Point p;
782     double x, y;
783     do {
784         x = g.getEnvelopeInternal().getMinX() + rnd.nextDouble()
785             *(g.getEnvelopeInternal().getMaxX() - g.getEnvelopeInternal()
786               .getMinX());
787         y = g.getEnvelopeInternal().getMinY() + rnd.nextDouble()
788             *(g.getEnvelopeInternal().getMaxY() - g.getEnvelopeInternal()
789               .getMinY());
790         p = MGC.xy2Point(x, y);
791     } while (!g.contains(p));
792     Coord coord = new Coord(p.getX(), p.getY());
793     return coord;
794 }
795
796 // Método Main
797 public static void main(String[] args) throws Exception {
798
799     CreateDemand cd = new CreateDemand();
800     cd.run();
801 }
802

```

```

803 // Método para leitura de shapefile
804 public Map<String,Geometry> readShapeFile(String filename, String attrString){
805
806     Map<String,Geometry> shapeMap = new HashMap<String, Geometry>();
807
808     for (SimpleFeature ft : ShapeFileReader.getAllFeatures(filename)) {
809
810         GeometryFactory geometryFactory= new GeometryFactory();
811         WKTRReader wktReader = new WKTRReader(geometryFactory);
812         Geometry geometry;
813
814         try {
815             geometry = wktReader.read((ft.getAttribute("the_geom")).toString());
816             shapeMap.put(ft.getAttribute(attrString).toString(),geometry);
817
818         } catch (ParseException e) {
819             // TODO Auto-generated catch block
820             e.printStackTrace();
821         }
822
823     }
824     return shapeMap;
825 }
826
827 }

```

## APÊNDICE K - Arquivo *config.xml* utilizado na simulação

O arquivo de configuração utilizado na simulação feita nesse trabalho é apresentado abaixo. As configurações que não são contempladas no arquivo foram tomadas pelo *software* como as configurações padrão.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd">
3 <config>
4
5 <!-- ===== -->
6
7 <module name="controler" >
8
9 <!-- Default=0; -->
10 <param name="firstIteration" value="0" />
11
12 <!-- Default=1000; -->
13 <param name="lastIteration" value="50" />
14
15 <param name="overwriteFiles" value="deleteDirectoryIfExists" />
16
17 <!-- iterationNumber % writeEventsInterval == 0 defines in which iterations events
18 are written to a file. `0' disables events writing completely. -->
19 <param name="writeEventsInterval" value="10" />
20
21 <!-- iterationNumber % writePlansInterval == 0 defines (hopefully) in which
22 iterations plans are written to a file. `0' disables plans writing completely.
23 Some plans in early iterations are always written -->
24 <param name="writePlansInterval" value="10" />
25
26 <!-- iterationNumber % writeSnapshotsInterval == 0 defines in which iterations
27 snapshots are written to a file. `0' disables snapshots writing completely -->
28 <param name="writeSnapshotsInterval" value="0" />
29
30 </module>
31
32 <!-- ===== -->
33
34 <module name="counts" >
35
36 <!-- Transport modes that will be respected for the counts comparison. 'car' is
37 default, which includes also bussed from the pt simulation module. Use this
38 parameter in combination with 'filterModes' = true! -->
39 <param name="analyzedModes" value="car" />
40
41 <!-- Specifies over how many iterations the link volumes should be averaged that
42 are used for the counts comparison. Use 1 or 0 to only use the link volumes of a
43 single iteration. This values cannot be larger than the value specified for
44 writeCountsInterval -->
45 <param name="averageCountsOverIterations" value="1" />
46
47 <!-- factor by which to re-scale the simulated values. necessary when simulation
48 runs with something different from 100%. needs to be adapted manually -->
49 <param name="countsScaleFactor" value="0.10" />
50
51 <!-- input file name to counts package -->
52 <param name="inputCountsFile" value="C:\Users\Junior\Desktop\counts.xml" />
53
```

```

44     <!-- possible values: `html`, `kml`, `txt`, `all` -->
45     <param name="outputformat" value="txt" />
46
47     <!-- Specifies how often the counts comparison should be calculated and written.
48     -->
49     <param name="writeCountsInterval" value="5" />
50 </module>
51 <!-- ===== -->
52
53 <module name="network">
54
55     <param name="inputNetworkFile" value="C:\Users\Junior\Desktop\network.xml"/>
56
57 </module>
58
59 <!-- ===== -->
60
61 <module name="plans">
62
63     <param name="inputPlansFile" value="C:\Users\Junior\Desktop\plans.xml"/>
64
65 </module>
66
67 <!-- ===== -->
68

```

```

69 <module name="planCalcScore" >
70
71     <parameterset type="scoringParameters" >
72         <param name="earlyDeparture" value="-0.0" />
73         <param name="lateArrival" value="-18.0" />
74         <param name="marginalUtilityOfMoney" value="1.0" />
75         <param name="performing" value="6.0" />
76         <param name="subpopulation" value="null" />
77         <param name="utilityOfLineSwitch" value="-1.0" />
78         <param name="waiting" value="-0.0" />
79         <param name="waitingPt" value="-6.0" />
80
81     <parameterset type="activityParams" >
82         <param name="activityType" value="w10to17" />
83         <param name="closingTime" value="17:00:00" />
84         <param name="earliestEndTime" value="undefined" />
85         <param name="latestStartTime" value="undefined" />
86         <param name="minimalDuration" value="04:00:00" />
87         <param name="openingTime" value="10:00:00" />
88         <param name="priority" value="1.0" />
89         <param name="scoringThisActivityAtAll" value="yes" />
90         <param name="typicalDuration" value="undefined" />
91
92         <!-- method to compute score at typical duration. Use uniform for
93         backwards compatibility (all activities same score; higher proba to drop
94         long acts). -->
95         <param name="typicalDurationScoreComputation" value="uniform" />
96     </parameterset>

```

```

96 <parameterset type="activityParams" >
97   <param name="activityType" value="w10to22" />
98   <param name="closingTime" value="22:00:00" />
99   <param name="earliestEndTime" value="undefined" />
100  <param name="latestStartTime" value="undefined" />
101  <param name="minimalDuration" value="04:00:00" />
102  <param name="openingTime" value="10:00:00" />
103  <param name="priority" value="1.0" />
104  <param name="scoringThisActivityAtAll" value="yes" />
105  <param name="typicalDuration" value="undefined" />
106
107  <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
108  <param name="typicalDurationScoreComputation" value="uniform" />
109 </parameterset>
110
111 <parameterset type="activityParams" >
112   <param name="activityType" value="w11to17" />
113   <param name="closingTime" value="17:00:00" />
114   <param name="earliestEndTime" value="undefined" />
115   <param name="latestStartTime" value="undefined" />
116   <param name="minimalDuration" value="04:00:00" />
117   <param name="openingTime" value="11:00:00" />
118   <param name="priority" value="1.0" />
119   <param name="scoringThisActivityAtAll" value="yes" />
120   <param name="typicalDuration" value="undefined" />
121
122  <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
123  <param name="typicalDurationScoreComputation" value="uniform" />
124 </parameterset>
125
126 <parameterset type="activityParams" >
127   <param name="activityType" value="w15to22" />
128   <param name="closingTime" value="22:00:00" />
129   <param name="earliestEndTime" value="undefined" />
130   <param name="latestStartTime" value="undefined" />
131   <param name="minimalDuration" value="04:00:00" />
132   <param name="openingTime" value="15:00:00" />
133   <param name="priority" value="1.0" />
134   <param name="scoringThisActivityAtAll" value="yes" />
135   <param name="typicalDuration" value="undefined" />
136
137  <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
138  <param name="typicalDurationScoreComputation" value="uniform" />
139 </parameterset>
140
141 <parameterset type="activityParams" >
142   <param name="activityType" value="w17to22" />
143   <param name="closingTime" value="22:00:00" />
144   <param name="earliestEndTime" value="undefined" />
145   <param name="latestStartTime" value="undefined" />
146   <param name="minimalDuration" value="04:00:00" />
147   <param name="openingTime" value="17:00:00" />
148   <param name="priority" value="1.0" />
149   <param name="scoringThisActivityAtAll" value="yes" />
150   <param name="typicalDuration" value="undefined" />
151
152  <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
153  <param name="typicalDurationScoreComputation" value="uniform" />
154 </parameterset>
155

```



```

156 <parameterset type="activityParams" >
157   <param name="activityType" value="w24h" />
158   <param name="closingTime" value="23:59:59" />
159   <param name="earliestEndTime" value="undefined" />
160   <param name="latestStartTime" value="undefined" />
161   <param name="minimalDuration" value="04:00:00" />
162   <param name="openingTime" value="04:00:00" />
163   <param name="priority" value="1.0" />
164   <param name="scoringThisActivityAtAll" value="yes" />
165   <param name="typicalDuration" value="undefined" />
166
167   <!-- method to compute score at typical duration. Use uniform for
        backwards compatibility (all activities same score; higher proba to drop
        long acts). -->
168   <param name="typicalDurationScoreComputation" value="uniform" />
169 </parameterset>
170
171 <parameterset type="activityParams" >
172   <param name="activityType" value="w6to23" />
173   <param name="closingTime" value="23:00:00" />
174   <param name="earliestEndTime" value="undefined" />
175   <param name="latestStartTime" value="undefined" />
176   <param name="minimalDuration" value="04:00:00" />
177   <param name="openingTime" value="06:00:00" />
178   <param name="priority" value="1.0" />
179   <param name="scoringThisActivityAtAll" value="yes" />
180   <param name="typicalDuration" value="undefined" />
181
182   <!-- method to compute score at typical duration. Use uniform for
        backwards compatibility (all activities same score; higher proba to drop
        long acts). -->
183   <param name="typicalDurationScoreComputation" value="uniform" />
184 </parameterset>
185
186 <parameterset type="activityParams" >
187   <param name="activityType" value="w730to2030" />
188   <param name="closingTime" value="20:30:00" />
189   <param name="earliestEndTime" value="undefined" />
190   <param name="latestStartTime" value="undefined" />
191   <param name="minimalDuration" value="04:00:00" />
192   <param name="openingTime" value="07:30:00" />
193   <param name="priority" value="1.0" />
194   <param name="scoringThisActivityAtAll" value="yes" />
195   <param name="typicalDuration" value="undefined" />
196
197   <!-- method to compute score at typical duration. Use uniform for
        backwards compatibility (all activities same score; higher proba to drop
        long acts). -->
198   <param name="typicalDurationScoreComputation" value="uniform" />
199 </parameterset>
200
201 <parameterset type="activityParams" >
202   <param name="activityType" value="w7to1830" />
203   <param name="closingTime" value="18:30:00" />
204   <param name="earliestEndTime" value="undefined" />
205   <param name="latestStartTime" value="undefined" />
206   <param name="minimalDuration" value="04:00:00" />
207   <param name="openingTime" value="07:00:00" />
208   <param name="priority" value="1.0" />
209   <param name="scoringThisActivityAtAll" value="yes" />
210   <param name="typicalDuration" value="undefined" />
211
212   <!-- method to compute score at typical duration. Use uniform for
        backwards compatibility (all activities same score; higher proba to drop
        long acts). -->
213   <param name="typicalDurationScoreComputation" value="uniform" />
214 </parameterset>
215

```

```

216 <parameterset type="activityParams" >
217   <param name="activityType" value="w7to22" />
218   <param name="closingTime" value="22:00:00" />
219   <param name="earliestEndTime" value="undefined" />
220   <param name="latestStartTime" value="undefined" />
221   <param name="minimalDuration" value="04:00:00" />
222   <param name="openingTime" value="07:00:00" />
223   <param name="priority" value="1.0" />
224   <param name="scoringThisActivityAtAll" value="yes" />
225   <param name="typicalDuration" value="undefined" />
226
227   <!-- method to compute score at typical duration. Use uniform for
228   backwards compatibility (all activities same score; higher proba to drop
229   long acts). -->
230   <param name="typicalDurationScoreComputation" value="uniform" />
231 </parameterset>
232
233 <parameterset type="activityParams" >
234   <param name="activityType" value="w8to18" />
235   <param name="closingTime" value="18:00:00" />
236   <param name="earliestEndTime" value="undefined" />
237   <param name="latestStartTime" value="undefined" />
238   <param name="minimalDuration" value="04:00:00" />
239   <param name="openingTime" value="08:00:00" />
240   <param name="priority" value="1.0" />
241   <param name="scoringThisActivityAtAll" value="yes" />
242   <param name="typicalDuration" value="undefined" />
243
244   <!-- method to compute score at typical duration. Use uniform for
245   backwards compatibility (all activities same score; higher proba to drop
246   long acts). -->
247   <param name="typicalDurationScoreComputation" value="uniform" />
248 </parameterset>
249
250 <parameterset type="activityParams" >
251   <param name="activityType" value="w8to20" />
252   <param name="closingTime" value="20:00:00" />
253   <param name="earliestEndTime" value="undefined" />
254   <param name="latestStartTime" value="undefined" />
255   <param name="minimalDuration" value="04:00:00" />
256   <param name="openingTime" value="08:00:00" />
257   <param name="priority" value="1.0" />
258   <param name="scoringThisActivityAtAll" value="yes" />
259   <param name="typicalDuration" value="undefined" />
260
261   <!-- method to compute score at typical duration. Use uniform for
262   backwards compatibility (all activities same score; higher proba to drop
263   long acts). -->
264   <param name="typicalDurationScoreComputation" value="uniform" />
265 </parameterset>
266
267 <parameterset type="activityParams" >
268   <param name="activityType" value="w8to23" />
269   <param name="closingTime" value="23:00:00" />
270   <param name="earliestEndTime" value="undefined" />
271   <param name="latestStartTime" value="undefined" />
272   <param name="minimalDuration" value="04:00:00" />
273   <param name="openingTime" value="08:00:00" />
274   <param name="priority" value="1.0" />
275   <param name="scoringThisActivityAtAll" value="yes" />
276   <param name="typicalDuration" value="undefined" />
277
278   <!-- method to compute score at typical duration. Use uniform for
279   backwards compatibility (all activities same score; higher proba to drop
280   long acts). -->
281   <param name="typicalDurationScoreComputation" value="uniform" />
282 </parameterset>

```



```

276 <parameterset type="activityParams" >
277   <param name="activityType" value="w9to18" />
278   <param name="closingTime" value="18:00:00" />
279   <param name="earliestEndTime" value="undefined" />
280   <param name="latestStartTime" value="undefined" />
281   <param name="minimalDuration" value="04:00:00" />
282   <param name="openingTime" value="09:00:00" />
283   <param name="priority" value="1.0" />
284   <param name="scoringThisActivityAtAll" value="yes" />
285   <param name="typicalDuration" value="undefined" />
286
287   <!-- method to compute score at typical duration. Use uniform for
        backwards compatibility (all activities same score; higher proba to drop
        long acts). -->
288   <param name="typicalDurationScoreComputation" value="uniform" />
289 </parameterset>
290
291 <parameterset type="activityParams" >
292   <param name="activityType" value="wfrom11" />
293   <param name="closingTime" value="23:59:59" />
294   <param name="earliestEndTime" value="undefined" />
295   <param name="latestStartTime" value="undefined" />
296   <param name="minimalDuration" value="04:00:00" />
297   <param name="openingTime" value="11:00:00" />
298   <param name="priority" value="1.0" />
299   <param name="scoringThisActivityAtAll" value="yes" />
300   <param name="typicalDuration" value="undefined" />
301
302   <!-- method to compute score at typical duration. Use uniform for
        backwards compatibility (all activities same score; higher proba to drop
        long acts). -->
303   <param name="typicalDurationScoreComputation" value="uniform" />
304 </parameterset>
305
306 <parameterset type="activityParams" >
307   <param name="activityType" value="wfrom17" />
308   <param name="closingTime" value="23:59:59" />
309   <param name="earliestEndTime" value="undefined" />
310   <param name="latestStartTime" value="undefined" />
311   <param name="minimalDuration" value="04:00:00" />
312   <param name="openingTime" value="17:00:00" />
313   <param name="priority" value="1.0" />
314   <param name="scoringThisActivityAtAll" value="yes" />
315   <param name="typicalDuration" value="undefined" />
316
317   <!-- method to compute score at typical duration. Use uniform for
        backwards compatibility (all activities same score; higher proba to drop
        long acts). -->
318   <param name="typicalDurationScoreComputation" value="uniform" />
319 </parameterset>
320
321 <parameterset type="activityParams" >
322   <param name="activityType" value="o10to17" />
323   <param name="closingTime" value="17:00:00" />
324   <param name="earliestEndTime" value="undefined" />
325   <param name="latestStartTime" value="undefined" />
326   <param name="minimalDuration" value="undefined" />
327   <param name="openingTime" value="10:00:00" />
328   <param name="priority" value="0.5" />
329   <param name="scoringThisActivityAtAll" value="yes" />
330   <param name="typicalDuration" value="undefined" />
331
332   <!-- method to compute score at typical duration. Use uniform for
        backwards compatibility (all activities same score; higher proba to drop
        long acts). -->
333   <param name="typicalDurationScoreComputation" value="uniform" />
334 </parameterset>
335

```

```

336 <parameterset type="activityParams" >
337   <param name="activityType" value="o10to22" />
338   <param name="closingTime" value="22:00:00" />
339   <param name="earliestEndTime" value="undefined" />
340   <param name="latestStartTime" value="undefined" />
341   <param name="minimalDuration" value="undefined" />
342   <param name="openingTime" value="10:00:00" />
343   <param name="priority" value="0.5" />
344   <param name="scoringThisActivityAtAll" value="yes" />
345   <param name="typicalDuration" value="undefined" />
346
347   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
348   <param name="typicalDurationScoreComputation" value="uniform" />
349 </parameterset>
350
351 <parameterset type="activityParams" >
352   <param name="activityType" value="o11to17" />
353   <param name="closingTime" value="17:00:00" />
354   <param name="earliestEndTime" value="undefined" />
355   <param name="latestStartTime" value="undefined" />
356   <param name="minimalDuration" value="undefined" />
357   <param name="openingTime" value="11:00:00" />
358   <param name="priority" value="0.5" />
359   <param name="scoringThisActivityAtAll" value="yes" />
360   <param name="typicalDuration" value="undefined" />
361
362   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
363   <param name="typicalDurationScoreComputation" value="uniform" />
364 </parameterset>
365
366 <parameterset type="activityParams" >
367   <param name="activityType" value="o15to22" />
368   <param name="closingTime" value="22:00:00" />
369   <param name="earliestEndTime" value="undefined" />
370   <param name="latestStartTime" value="undefined" />
371   <param name="minimalDuration" value="undefined" />
372   <param name="openingTime" value="15:00:00" />
373   <param name="priority" value="0.5" />
374   <param name="scoringThisActivityAtAll" value="yes" />
375   <param name="typicalDuration" value="undefined" />
376
377   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
378   <param name="typicalDurationScoreComputation" value="uniform" />
379 </parameterset>
380
381 <parameterset type="activityParams" >
382   <param name="activityType" value="o17to22" />
383   <param name="closingTime" value="22:00:00" />
384   <param name="earliestEndTime" value="undefined" />
385   <param name="latestStartTime" value="undefined" />
386   <param name="minimalDuration" value="undefined" />
387   <param name="openingTime" value="17:00:00" />
388   <param name="priority" value="0.5" />
389   <param name="scoringThisActivityAtAll" value="yes" />
390   <param name="typicalDuration" value="undefined" />
391
392   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
393   <param name="typicalDurationScoreComputation" value="uniform" />
394 </parameterset>
395

```

```

396 <parameterset type="activityParams" >
397   <param name="activityType" value="o24h" />
398   <param name="closingTime" value="23:59:59" />
399   <param name="earliestEndTime" value="undefined" />
400   <param name="latestStartTime" value="undefined" />
401   <param name="minimalDuration" value="undefined" />
402   <param name="openingTime" value="04:00:00" />
403   <param name="priority" value="0.5" />
404   <param name="scoringThisActivityAtAll" value="yes" />
405   <param name="typicalDuration" value="undefined" />
406
407   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
408   <param name="typicalDurationScoreComputation" value="uniform" />
409 </parameterset>
410
411 <parameterset type="activityParams" >
412   <param name="activityType" value="o6to23" />
413   <param name="closingTime" value="23:00:00" />
414   <param name="earliestEndTime" value="undefined" />
415   <param name="latestStartTime" value="undefined" />
416   <param name="minimalDuration" value="undefined" />
417   <param name="openingTime" value="06:00:00" />
418   <param name="priority" value="0.5" />
419   <param name="scoringThisActivityAtAll" value="yes" />
420   <param name="typicalDuration" value="undefined" />
421
422   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
423   <param name="typicalDurationScoreComputation" value="uniform" />
424 </parameterset>
425
426 <parameterset type="activityParams" >
427   <param name="activityType" value="o730to2030" />
428   <param name="closingTime" value="20:30:00" />
429   <param name="earliestEndTime" value="undefined" />
430   <param name="latestStartTime" value="undefined" />
431   <param name="minimalDuration" value="undefined" />
432   <param name="openingTime" value="07:30:00" />
433   <param name="priority" value="0.5" />
434   <param name="scoringThisActivityAtAll" value="yes" />
435   <param name="typicalDuration" value="undefined" />
436
437   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
438   <param name="typicalDurationScoreComputation" value="uniform" />
439 </parameterset>
440
441 <parameterset type="activityParams" >
442   <param name="activityType" value="o7to1830" />
443   <param name="closingTime" value="18:30:00" />
444   <param name="earliestEndTime" value="undefined" />
445   <param name="latestStartTime" value="undefined" />
446   <param name="minimalDuration" value="undefined" />
447   <param name="openingTime" value="07:00:00" />
448   <param name="priority" value="0.5" />
449   <param name="scoringThisActivityAtAll" value="yes" />
450   <param name="typicalDuration" value="undefined" />
451
452   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
453   <param name="typicalDurationScoreComputation" value="uniform" />
454 </parameterset>
455

```

```

456 <parameterset type="activityParams" >
457   <param name="activityType" value="o7to22" />
458   <param name="closingTime" value="22:00:00" />
459   <param name="earliestEndTime" value="undefined" />
460   <param name="latestStartTime" value="undefined" />
461   <param name="minimalDuration" value="undefined" />
462   <param name="openingTime" value="07:00:00" />
463   <param name="priority" value="0.5" />
464   <param name="scoringThisActivityAtAll" value="yes" />
465   <param name="typicalDuration" value="undefined" />
466
467   <!-- method to compute score at typical duration. Use uniform for
         backwards compatibility (all activities same score; higher proba to drop
         long acts). -->
468   <param name="typicalDurationScoreComputation" value="uniform" />
469 </parameterset>
470
471 <parameterset type="activityParams" >
472   <param name="activityType" value="o8to18" />
473   <param name="closingTime" value="18:00:00" />
474   <param name="earliestEndTime" value="undefined" />
475   <param name="latestStartTime" value="undefined" />
476   <param name="minimalDuration" value="undefined" />
477   <param name="openingTime" value="08:00:00" />
478   <param name="priority" value="0.5" />
479   <param name="scoringThisActivityAtAll" value="yes" />
480   <param name="typicalDuration" value="undefined" />
481
482   <!-- method to compute score at typical duration. Use uniform for
         backwards compatibility (all activities same score; higher proba to drop
         long acts). -->
483   <param name="typicalDurationScoreComputation" value="uniform" />
484 </parameterset>
485
486 <parameterset type="activityParams" >
487   <param name="activityType" value="o8to20" />
488   <param name="closingTime" value="20:00:00" />
489   <param name="earliestEndTime" value="undefined" />
490   <param name="latestStartTime" value="undefined" />
491   <param name="minimalDuration" value="undefined" />
492   <param name="openingTime" value="08:00:00" />
493   <param name="priority" value="0.5" />
494   <param name="scoringThisActivityAtAll" value="yes" />
495   <param name="typicalDuration" value="undefined" />
496   <!-- method to compute score at typical duration. Use uniform for
         backwards compatibility (all activities same score; higher proba to drop
         long acts). -->
497   <param name="typicalDurationScoreComputation" value="uniform" />
498 </parameterset>
499
500 <parameterset type="activityParams" >
501   <param name="activityType" value="o8to23" />
502   <param name="closingTime" value="23:00:00" />
503   <param name="earliestEndTime" value="undefined" />
504   <param name="latestStartTime" value="undefined" />
505   <param name="minimalDuration" value="undefined" />
506   <param name="openingTime" value="08:00:00" />
507   <param name="priority" value="0.5" />
508   <param name="scoringThisActivityAtAll" value="yes" />
509   <param name="typicalDuration" value="undefined" />
510
511   <!-- method to compute score at typical duration. Use uniform for
         backwards compatibility (all activities same score; higher proba to drop
         long acts). -->
512   <param name="typicalDurationScoreComputation" value="uniform" />
513 </parameterset>
514

```



```

515 <parameterset type="activityParams" >
516   <param name="activityType" value="o9to18" />
517   <param name="closingTime" value="18:00:00" />
518   <param name="earliestEndTime" value="undefined" />
519   <param name="latestStartTime" value="undefined" />
520   <param name="minimalDuration" value="undefined" />
521   <param name="openingTime" value="09:00:00" />
522   <param name="priority" value="0.5" />
523   <param name="scoringThisActivityAtAll" value="yes" />
524   <param name="typicalDuration" value="undefined" />
525
526   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
527   <param name="typicalDurationScoreComputation" value="uniform" />
528 </parameterset>
529
530 <parameterset type="activityParams" >
531   <param name="activityType" value="ofrom11" />
532   <param name="closingTime" value="23:59:59" />
533   <param name="earliestEndTime" value="undefined" />
534   <param name="latestStartTime" value="undefined" />
535   <param name="minimalDuration" value="undefined" />
536   <param name="openingTime" value="11:00:00" />
537   <param name="priority" value="0.5" />
538   <param name="scoringThisActivityAtAll" value="yes" />
539   <param name="typicalDuration" value="undefined" />
540
541   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
542   <param name="typicalDurationScoreComputation" value="uniform" />
543 </parameterset>
544
545 <parameterset type="activityParams" >
546   <param name="activityType" value="ofrom17" />
547   <param name="closingTime" value="23:59:59" />
548   <param name="earliestEndTime" value="undefined" />
549   <param name="latestStartTime" value="undefined" />
550   <param name="minimalDuration" value="undefined" />
551   <param name="openingTime" value="17:00:00" />
552   <param name="priority" value="0.5" />
553   <param name="scoringThisActivityAtAll" value="yes" />
554   <param name="typicalDuration" value="undefined" />
555   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
556   <param name="typicalDurationScoreComputation" value="uniform" />
557 </parameterset>
558
559
560 <parameterset type="activityParams" >
561   <param name="activityType" value="home" />
562   <param name="closingTime" value="undefined" />
563   <param name="earliestEndTime" value="undefined" />
564   <param name="latestStartTime" value="undefined" />
565   <param name="minimalDuration" value="undefined" />
566   <param name="openingTime" value="undefined" />
567   <param name="priority" value="0.1" />
568   <param name="scoringThisActivityAtAll" value="yes" />
569   <param name="typicalDuration" value="undefined" />
570
571   <!-- method to compute score at typical duration. Use uniform for
backwards compatibility (all activities same score; higher proba to drop
long acts). -->
572   <param name="typicalDurationScoreComputation" value="uniform" />
573 </parameterset>
574

```

```

575     <parameterset type="activityParams" >
576         <param name="activityType" value="study" />
577         <param name="closingTime" value="23:00:00" />
578         <param name="earliestEndTime" value="undefined" />
579         <param name="latestStartTime" value="undefined" />
580         <param name="minimalDuration" value="undefined" />
581         <param name="openingTime" value="8:00:00" />
582         <param name="priority" value="1.0" />
583         <param name="scoringThisActivityAtAll" value="yes" />
584         <param name="typicalDuration" value="undefined" />
585     </parameterset>
586 </module>
587
588 </module>
589
590 <!-- ===== -->
591
592 <module name="qsim" >
593     <param name="endTime" value="30:00:00" />
594     <param name="flowCapacityFactor" value="0.10" />
595     <param name="storageCapacityFactor" value="0.10" />
596 </module>
597
598 <!-- ===== -->
599
600 <module name="strategy">
601     <param name="maxAgentPlanMemorySize" value="8" /> <!-- 0 means unlimited -->
602
603     <param name="ModuleProbability_1" value="0.8" />
604     <param name="Module_1" value="ChangeExpBeta" />
605
606     <param name="ModuleProbability_2" value="0.4" />
607     <param name="Module_2" value="ReRoute" />
608
609     <param name="ModuleProbability_3" value="0.2" />
610     <param name="Module_3" value="TimeAllocationMutator" />
611
612 </module>
613
614 </config>

```

## ANEXO A - Classificação das rodovias pela base de dados *OpenStreet Map*

Para traçar equivalência entre as classificações de rodovias obtidas pelo OpenStreet Map e as definidas pelo Código de Trânsito Brasileiro, foi necessário considerar as características da classificação da base de dados, que são apresentadas abaixo. Essas informações foram traduzidas de OpenStreet Map Wiki (2017).

Classificação no OpenStreet Map	Características
<i>Motorway</i>	Via expressa (velocidade $\geq 80$ km/h), pavimentada (asfalto ou concreto), pista dupla (com divisão física entre os sentidos opostos) ou pista simples, com duas ou mais faixas de rolamento, SEM interseções ou obstruções (semáforos, lombadas, etc), acessível por entradas e saídas tangentes à via. Sempre mapeada como vias paralelas separadas para cada sentido.
<i>Trunk</i>	Via rápida (velocidade $\geq 80$ km/h), pavimentada (asfalto ou concreto), de pista dupla (com divisão física entre os sentidos opostos), ou de pista simples, com duas ou mais faixas de rolamento, COM interseções ou obstruções (semáforos, lombadas, etc), acessível diretamente pelo tráfego local por vias perpendiculares, que podem cruzar diretamente a via rápida. Sempre mapeada como vias paralelas separadas para cada sentido.
<i>Primary</i>	<b>Rural:</b> Via de trânsito rápido (velocidade $\geq 80$ km/h, ou 60km/h em terreno montanhoso), pavimentada (asfalto ou concreto), com espaço lateral suficiente para estacionar um veículo (2m ou mais) ou com mais de uma faixa de rolamento por sentido. <b>Urbana:</b> Via arterial, rápida (velocidade $\geq 60$ km/h), pavimentada (asfalto ou concreto), pista simples ou dupla, conectando muitos bairros, formando a rede de circulação principal em grandes cidades ou rotas de entrada e saída principais em cidades pequenas e médias.
<i>Secondary</i>	<b>Rural:</b> Via pavimentada (asfalto, concreto, pedra ou outro pavimento firme) que não tenha os parâmetros atribuídos às vias do tipo “ <i>Primary</i> ”. <b>Urbana:</b> Via coletora, velocidade média ( $\geq 40$ km/h), pavimentada (asfalto, concreto, pedra ou outro pavimento firme), geralmente de pista simples, formando uma rede de circulação secundária entre bairros de grandes cidades ou a rede de circulação principal de pequenas cidades.
<i>Tertiary</i>	<b>Rural:</b> Estrada não pavimentada (terra, brita, areia, etc) larga o suficiente para a passagem de dois veículos ( $\geq 6$ m), geralmente de solo compactado. <b>Urbana:</b> Via coletora, geralmente pavimentada e de pista simples, formando a rede de circulação interna de bairros em grandes cidades ou a rede de circulação secundária de pequenas cidades.
<i>Unclassified</i>	<b>Rural:</b> Estrada não pavimentada (terra, brita, areia, etc) larga o suficiente para a passagem de dois veículos ( $\geq 4$ m), com tráfego geralmente fluindo pelo centro da pista, geralmente feita de solo não-compactado. <b>Urbana:</b> Via local, com fluxo insignificante de veículos, fora de áreas residenciais. Para áreas residenciais, usa-se a classificação “ <i>residential</i> ” e para vias onde os pedestres têm prioridade sobre o automóvel, usa-se “ <i>living-street</i> ”.