

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

INSTITUTO DE INFORMÁTICA

CURSO DE CIÊNCIA DA COMPUTAÇÃO

HENRIQUE VIEIRA HALLMANN

**Comparação entre *Softwares* Simuladores de  
Trânsito**

Trabalho de Graduação.

Profa. Dra. Ana Lúcia C. Bazzan  
Orientadora

Porto Alegre, 2011.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

INSTITUTO DE INFORMÁTICA

CURSO DE CIÊNCIA DA COMPUTAÇÃO

HENRIQUE VIEIRA HALLMANN

**Comparação entre *Softwares* Simuladores de  
Trânsito**

Porto Alegre, 2011

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profª. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## Agradecimentos

Agradeço a família pelo apoio incondicional. Aos amigos que se preocuparam e ajudaram no que era possível. A minha mãe Paulina, com incentivos até a conclusão. A minha namorada Érica pelos conselhos e por todos que ajudaram de alguma maneira a realizar este trabalho, significa muito para mim.

# SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO .....</b>	<b>13</b>
<b>2</b>	<b>CONCEITOS BÁSICOS .....</b>	<b>16</b>
<b>2.1</b>	<b>Nível de detalhe dos modelos de simulação de trânsito .....</b>	<b>20</b>
2.1.1	Modelo macroscópico .....	21
<b>2.1.2</b>	<b>Modelo Microscópico .....</b>	<b>21</b>
2.1.3	Modelo Mesoscópico .....	23
<b>3</b>	<b>APRESENTAÇÃO DOS SIMULADORES .....</b>	<b>24</b>
<b>3.1</b>	<b>SUMO .....</b>	<b>24</b>
3.1.1	Características de movimento .....	24
3.1.2	Malha viária .....	26
3.1.3	Controladores de tráfego .....	27
3.1.4	Atribuição de demanda .....	28
3.1.5	Relatórios .....	29
<b>3.2</b>	<b>ITSUMO .....</b>	<b>30</b>
3.2.1	Características de movimento .....	30
3.2.2	Malha viária .....	31
3.2.3	Controladores de tráfego .....	33
3.2.4	Atribuição de demanda .....	34
3.2.5	Relatórios .....	34
<b>3.3</b>	<b>AIMS UN .....</b>	<b>35</b>
3.3.1	Características de movimento .....	36
3.3.2	Malha viária .....	37
3.3.3	Controladores de tráfego .....	39
3.3.4	Atribuição de demanda .....	41
3.3.5	Relatórios .....	42

3.4	<b>VISSIM.....</b>	<b>43</b>
3.4.1	Características de movimento .....	44
3.4.2	Malha viária .....	45
3.4.3	Controladores de tráfego .....	47
3.4.4	Atribuição de demanda .....	49
3.4.5	Relatórios .....	51
<b>4</b>	<b>ESTUDO DE CASO .....</b>	<b>53</b>
4.1	<b>Grade 6x6.....</b>	<b>53</b>
4.1.1	SUMO .....	56
4.1.2	ITSUMO .....	61
4.1.3	AIMSUN.....	64
4.1.4	VISSIM .....	68
4.1.5	Análise dos Resultados.....	71
4.2	<b>Porto Alegre, vias principais.....</b>	<b>78</b>
4.2.1	Malha viária.....	79
4.2.2	Atribuição de tráfego .....	80
4.2.3	Controladores de tráfego .....	80
4.2.4	Resultados obtidos.....	<b>Erro! Indicador não definido.</b>
<b>5</b>	<b>COMPARAÇÃO ENTRE OS SOFTWARES.....</b>	<b>85</b>
5.1	<b>Instalação dos <i>softwares</i> .....</b>	<b>85</b>
5.1.1	Sistema Operacional .....	86
5.1.2	Licença .....	86
5.1.3	Requisitos mínimos .....	87
5.1.4	Documentação.....	87
5.1.5	Banco de dados .....	87
5.1.6	Idiomas oferecidos .....	88
5.2	<b>Malha viária .....</b>	<b>89</b>
5.2.1	Importar malhas viárias .....	89
5.2.2	Interface.....	89
5.2.3	Ferramentas para interação .....	90
5.2.4	Intersecções não sinalizadas .....	91
5.2.5	Intersecções sinalizadas .....	91
5.3	<b>Atribuição de demanda .....</b>	<b>92</b>
5.3.1	Tipos de veículos .....	92

5.3.2	Suporte a matriz OD .....	93
5.3.3	Regras de trânsito regionais .....	93
5.3.4	Interface.....	93
5.3.5	Tráfego assinalado dinamicamente .....	93
5.4	<b>Simulação .....</b>	<b>94</b>
5.4.1	Interface.....	94
5.4.2	Ferramentas .....	95
5.4.3	Simular acidentes.....	96
5.5	<b>Aspectos gerais .....</b>	<b>96</b>
5.5.1	Extensibilidade.....	96
5.5.2	Ruído e poluição .....	97
5.5.3	Validação .....	97
5.5.4	Calibração .....	<b>Erro! Indicador não definido.</b>
5.5.5	Vídeos.....	98
5.5.6	Resultados.....	98
5.6	<b>Tabela final.....</b>	<b>99</b>
	<b>CONCLUSÃO.....</b>	<b>100</b>
	<b>REFERÊNCIAS.....</b>	<b>102</b>

## LISTA DE FIGURAS

<i>Figura 2.1. Representação de demanda em uma matriz origem – destino. Os círculos representam as regiões e as linhas, as vias.</i>	17
<i>Figura 2.2 Representação de uma mapa cartográfico para um modelo digital</i>	19
<i>Figura 2.3: Representação gráfica de uma intersecção com opções de conversão</i>	20
<i>Figura 3.1: Informações de nodos a esquerda e arestas a direita, transformado em malha viária pelo NETCONVERT</i>	27
<i>Figura 3.2: Exemplo de malha viária criada no software SUMO</i>	27
<i>Figura 3.3: Configuração de um nodo com semáforo simples</i>	27
<i>Figura 3.4: Interface para editar topologias no ITSUMO</i>	32
<i>Figura 3.5: Malha viária obtida no ITSUMO</i>	33
<i>Figura 3.6: Configuração de semáforo no ITSUMO</i>	33
<i>Figura 3.7: Configuração de probabilidades marcado em vermelho, para um “laneset” escolhido</i>	34
<i>Figura 3.8: Tutorial do AIMSUN</i>	38
<i>Figura 3.9: Nodo sendo criado</i>	39
<i>Figura 3.10: Visualizações 2D e 3D disponíveis no AIMSUN</i>	39
<i>Figura 3.11: Controladores de tráfego em uma intersecção no AIMSUN</i>	40
<i>Figura 3.12: Configuração de plano semafórico</i>	40
<i>Figura 3.13: Centróide no AIMSUN</i>	41
<i>Figura 3.14: Edição de uma Matriz OD no AIMSUN</i>	41
<i>Figura 3.15: Séries plotadas no AIMSUN a partir das informações personalizada em Séries Temporais</i>	42
<i>Figura 3.16: Vias coloridas de acordo com fluxo</i>	43
<i>Figura 3.17: Intersecções não sinalizadas. A figura da esquerda representa uma área de conflito padrão, e da direita, a linha pare (vermelho) associada a marcas de conflito.</i>	45
<i>Figura 3.18: Representação de um nodo em segmentos de via, à esquerda, e malha viária com diversos nodos à direita</i>	47
<i>Figura 3.19: Exemplo de controladores configurados no VISSIM. As linhas vermelhas representam estes controladores</i>	48
<i>Figura 3.20: Semáforos com intervalos de tempo a esquerda e interface gráfica VisVAP à direita</i>	48
<i>Figura 3.21: Estacionamentos inseridos nas bordas da malha viária</i>	50
<i>Figura 3.22: Sequência de ações para inserir matriz OD no VISSIM (Traffic→DynamicAssignment→Matrices→New→Matrix file)</i>	51



<i>Figura 3.23: Informação de veículos a esquerda e semáforos a direita</i> .....	52
<i>Figura 4.1: Exemplo de uma Grade 6x6, desenvolvida no software ITSUMO</i> .....	54
<i>Figura 4.2: Intersecções selecionadas para inserir semáforos</i> .....	55
<i>Figura 4.3: Matriz OD configurada para os softwares VISSIM e SUMO.</i> .....	56
<i>Figura 4.4: Arquivo 6x6.nod.xml que contém o código para configurar nodos no SUMO. Coordenadas do plano cartesiano (x e y) são configuradas nesta etapa, indicando a localização de cada nodo</i> .....	57
<i>Figura 4.5: Arquivo 6x6.edg.xml que contém o código para configurar arestas no SUMO</i> .....	57
<i>Figura 4.6: Código para o software NETCONVERT construir a malha viária, onde: netconvert é a ferramanta, 6x6.nod.xml é o arquivo com as posições dos nodos, 6x6.edg.xml é o arquivo das arestas e 6x6.net.xml é a malha viária resultante</i> .....	57
<i>Figura 4.7: Grade 6x6 construída no SUMO</i> .....	58
<i>Figura 4.8: Regiões formadas no SUMO e suas conexões através do arquivo 6x6.taz.xml</i> .....	59
<i>Figura 4.9: Comando para gerar trips no SUMO, onde: od2trips executa a ferramenta, 6x6.taz.xml é o arquivo que contém as regiões configuradas, 6x6.odm.txt contém a matriz e 6x6.trp.xml das viagens</i> ....	59
<i>Figura 4.10: Trips/Viagens criadas automaticamente com o software OD2TRIPS</i> .....	59
<i>Figura 4.11: Arquivo adicional para incluir resultados na simulação</i> .....	60
<i>Figura 4.12: Relatório de resultados da simulação adquirido com o “o DetectNetTraffic”</i> .....	60
<i>Figura 4.13: Comando para gerar as rotas com origem e destino</i> .....	60
<i>Figura 4.14: Arquivo 6x6.rou.xml - Rota completa de cada veículo, calculada a partir do software DUAROUTER</i> .....	61
<i>Figura 4.15: Arquivo principal do SUMO, responsável pelo início da simulação</i> .....	61
<i>Figura 4.16: Definição de nodos no ITSUMO com as coordenadas em x e y do plano cartesiano</i> .....	62
<i>Figura 4.17: Inserindo vias no ITSUMO através do recurso “Create street”</i> .....	62
<i>Figura 4.18: Arquivo de configuração do ITSUMO</i> .....	63
<i>Figura 4.19: Arquivo com definições de origem e destino para cada veículo</i> .....	63
<i>Figura 4.20: Exemplo de resultados obtidos com o lane_sensor</i> .....	64
<i>Figura 4.21: Malha viária criada no AIMSUN a partir de uma imagem importada do ITSUMO.</i> .....	65
<i>Figura 4.22: Plano de controle utilizado no AIMSUN</i> .....	66
<i>Figura 4.23: Correção de um problema com centróides. Inserção de uma via para entrada e saída de veículos da simulação</i> .....	67
<i>Figura 4.24: Tabela com valores coletado</i> .....	68
<i>Figura 4.25: Intersecção sendo criada no VISSIM</i> .....	69
<i>Figura 4.26: Malha viária construída no VISSIM</i> .....	69
<i>Figura 4.27 Definição de nodos e estacionamentos na grade 6x6. O nodo está representado pelo quadrado maior e o estacionamento, pelo retângulo sobre a pista</i> .....	70
<i>Figura 4.28: Configuração de tempo de viagem. A ferramenta “Travel Time” está em destaque</i> .....	71
<i>Figura 4.29: Vias selecionadas para análise do tempo médio de viagem</i> .....	73
<i>Figura 4.30: Tempos de viagem: resultados obtidos nos softwares SUMO, ITSUMO, AIMSUN e VISSIM.</i> .....	74
<i>Figura 4.31 Veículos sobrepostos no VISSIM</i> .....	75

<i>Figura 4.32: Resultados dos simuladores com semáforos .....</i>	<i>76</i>
<i>Figura 4.33: Veículos que não cumpriram a simulação e vias que mais congestionaram. A via 5 está congestionada no AIMSUN e a 10 aparece congestionada no SUMO.....</i>	<i>77</i>
<i>Figura 4.34: Vias congestionadas no ITSUMO.....</i>	<b><i>Erro! Indicador não definido.</i></b>
<i>Figura 4.35: Congestionamento na via 5 a esquerda. Congestionamento nas vias 9 e 10 a direita.....</i>	<i>77</i>
<i>Figura 4.36: Topologia desenvolvida para a cidade Porto Alegre.....</i>	<i>78</i>
<i>Figura 4.37: Escala utilizada no VISSIM.....</i>	<i>79</i>
<i>Figura 4.38: Resultados obtidos no VISSIM e no SUMO, sem semáforos ....</i>	<b><i>Erro! Indicador não definido.</i></b>
<i>Figura 4.39: Via Carlos Gomes lotada no VISSIM .....</i>	<i>82</i>
<i>Figura 4.40: Resultados do AIMSUN e ITSUMO, sem semáforos.....</i>	<b><i>Erro! Indicador não definido.</i></b>
<i>Figura 4.41: Vias coloridas no AIMSUN conforme densidade .....</i>	<i>83</i>
<i>Figura 4.42: Resultados da simulação para o software AIMSUN, SUMO, ITSUMO com semáforos .....</i>	<i>83</i>
<i>Figura 4.43: Alta densidade de veículos na metade da via Carlos Gomes no SUMO .....</i>	<i>84</i>

## LISTA DE TABELAS

<i>Tabela 4.1: Vias seleccionadas para análise do tempo médio de viagem.....</i>	<i>72</i>
<i>Tabela 5.1: Classificação dos softwares quanto aos itens de instalação.....</i>	<i>88</i>
<i>Tabela 5.2: Classificação dos softwares quanto aos itens da malha viária.....</i>	<i>92</i>
<i>Tabela 5.3: Classificação dos softwares quanto aos itens de atribuição de demanda.....</i>	<i>94</i>
<i>Tabela 5.4: Classificação dos softwares quanto aos itens de simulação.....</i>	<i>96</i>
<i>Tabela 5.5: Classificação dos softwares quanto aos aspectos gerais.....</i>	<i>98</i>

## RESUMO

Os Sistemas Inteligentes de Transporte (ITS) se valem do uso de tecnologia de informação para realizar gerenciamento e controle de sistemas de transporte. Entre eles, os *softwares* simuladores de trânsito são ferramentas úteis no controle do tráfego urbano e rodoviário. Devido à imensa quantidade existente destes programas, faz-se necessário um estudo que avalie e compare estes programas. O objetivo deste trabalho foi analisar quatro *softwares* produzidos por universidades, centros de pesquisa e empresas privadas, são eles: AIMSUN, VISSIM, SUMO e ITSUMO.

Entre os conduzidos por universidades o ITSUMO, desenvolvido na Universidade Federal do Rio Grande do Sul, e o SUMO, projetado no Instituto de Sistemas de Transporte, representam os *softwares* simuladores de trânsito de código aberto. Entre as empresas privadas, os *softwares* VISSIM e AIMSUN, desenvolvido pela empresa PTV e TSS, respectivamente, representam este ramo.

Foram avaliados aspectos para a comparação entre eles, como a instalação do programa, a elaboração da malha viária, a atribuição de tráfego, a simulação visual e aspectos gerais.

Em praticamente todos os aspectos estudados, os *softwares* que pertencem a empresas privadas apresentaram vantagens em relação aos disponibilizados sobre a licença GPL.

**Palavras-Chave:** *software* simulador de trânsito, Sistemas Inteligentes de Transporte, ITS, ITSUMO, SUMO, VISSIM, AISUM.

# Comparison Between Traffic Simulator Softwares

## ABSTRACT

The Intelligent Transportation Systems (ITS) rely on the use of information technology to perform management and control of transportation systems. Among them, the traffic software simulators are useful in control of traffic systems. Due to the huge amount of existing programs, it is necessary to a study to evaluate and compare these programs. The objective of this study was to analyze four softwares produced by universities, research centers and private companies, they are AIMSUN, VISSIM, SUMO e ITSUMO.

Among the conducted by universities the ITSUMO, developed at the University Federal of Rio Grande do Sul, and SUMO, designed at the Institute of Transportation Systems, represent the traffic simulation software open source. Among private companies, VISSIM and AIMSUN, developed by PTV and TSS, respectively, represent this branch.

Criteria were evaluated to compare them, as the installation of the program, developing the road network, the traffic assignment, the visual simulation and general aspects.

In virtually all the criteria studied, the software that belong to private companies had advantages over those made available under the GPL license.

**Keywords:** Software simulation traffic, Intelligent Transportation Systems, ITS ITSUMO, SUMO, VISSIM, AIMSUN

## 1. INTRODUÇÃO

O desenvolvimento socioeconômico aliado ao crescimento desorganizado das cidades acarretou em impactos alarmantes no trânsito. A saturação do número de veículos circulantes, aliado às dificuldades em aumentar a capacidade das rodovias, leva a longos congestionamentos nas cidades, situações que podem ser evitadas com um planejamento urbano que inclua a dinâmica do trânsito. *Softwares* que auxiliem no controle e previsão deste trânsito tornam-se ferramentas essenciais para minimizar e até mesmo corrigir possíveis problemas.

A utilização de *softwares* para observar o comportamento do trânsito é a melhor opção para lidar com a complexidade dos sistemas de transporte. Atualmente no mercado, existe uma ampla variedade de softwares simuladores de trânsito (SST), uma das principais ferramentas relacionada a Sistemas Inteligentes de Transporte. Apesar das diversas pesquisas e *workshops* relacionados ao assunto, ainda existem lacunas sobre o assunto, pois as maiores fontes relacionadas ao assunto são os manuais fornecidos pelos softwares. Neste documento serão analisados alguns destes *softwares*.

O *Advanced Interactive Microscopic Simulator for Urban and Non-urban Networks* (AIMSUN), originalmente desenvolvido como programa de pesquisa na Universidade da Catalonia (UPC), está agora na sétima versão comercial sob comando da empresa *Transport Simulation Systems* (TSS). Contem em uma única versão três tipos de modelo de transporte, incluindo ferramentas de assinalamento de trânsito dinâmicas e um simulador mesoscópico. (AIMSUM, 2011).

Entre os objetivos principais do programa destaca-se a engenharia de trânsito e, mais recentemente, o suporte à decisão de controle em tempo real do trânsito. A empresa visa fornecer soluções para curto e médio prazo de problemas operacionais e de planejamento, presente em mais de 2000 licenças, entre usuários de governos, consultores e universidades em aproximadamente 60 países. Este *software* concorre diretamente com o *Verkehr In Städten – SIMulations modell* (VISSIM) (BARCELÓ, 2010).

O *software* VISSIM, atualmente desenvolvido pela empresa *Traffic Mobility Logistics* (PTV) está em sua quinta versão comercial, incluindo um alto nível de detalhamento e uma modelagem realística do fluxo de trânsito, com suporte à simulações de pedestres e ciclistas (PTV, 2011). O VISSIM Combina engenharia de trânsito com opções de apresentações, inclusive animações em 3D. Este SST possui em torno de 7000 licenças distribuídas mundialmente nos últimos 15 anos, com um terço das vendas entre consultores na indústria, um terço entre agências públicas e a última parte é aplicada em institutos acadêmicos para ensino e pesquisa. (PTVAG, 2011). Desenvolvido para ser um *software* de simulação microscópica discreta, tanto em ambiente urbano como interurbano, o VISSIM está baseado em diversos modelos matemáticos. (BARCELÓ, 2010).

Paralelo ao desenvolvimento dos *softwares* comerciais, a comunidade acadêmica criava *softwares* de simulação para utilizar em suas pesquisas. Alguns centros acadêmicos disponibilizaram seus *softwares* para a comunidade. O *Simulation of Urban Mobility* (SUMO), desenvolvido em colaboração entre o *Center for Applied Informatics Cologne* (ZAIK), e o *Institute of Transportation Systems* (ITS), localizado no *German Aerospace Center* (DLR), na Alemanha, e o *Intelligent Transportation System for Urban Mobility* (ITSUMO) desenvolvido pelo Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS) no Brasil, são exemplos destes *softwares*.

O SUMO é um simulador microscópico de trânsito. O *software* utiliza um modelo de passos discretos de tempo de 1s, permitindo simular em tempo real de 100.000 a 200.000 veículos em um computador pessoal de mesa, incluindo a simulação dos semáforos, regras de trânsito e movimentos de mudança de faixa pelos veículos. O *software* é útil para simular grandes áreas em tempo mais rápido que o real.

Seu desenvolvimento foi influenciado por duas premissas: rapidez e portabilidade. Assim, o SUMO deve rodar em qualquer ambiente comum e a simulação deve ser aberta e de fácil compreensão. O SUMO foi desenvolvido no sistema operacional Solaris, uma derivação do núcleo Unix, e apresenta licença *software* livre (GPL).

O simulador ITSUMO é um modelo microscópico de movimentação física, baseado no autômato celular de Nagel e Schreckenberg (1992), pois este tipo de modelo leva em

consideração os veículos individualmente (BAZZAN, KLÜGL, 2007). Este programa prevê a integração de diversas funcionalidades, tais como o controle de tráfego e informações em tempo real (SILVA, et al 2006).

Um dos objetivos com o desenvolvimento do ITSUMO é criar um sistema de informação para capacitar a mobilidade urbana em integrar diferentes funcionalidades, tais como simples controladores de tráfego em tempo real, fornecidos via internet e/ou telefone móvel. O sistema tem sido utilizado também para investigar cenários de decisão de rotas e é distribuído sobre a licença de software livre. O projeto e os autores são financiados pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). O cenário de tráfego urbano é uma área rica para experimentos realizados com técnicas baseadas em agentes, considerando a compreensão do comportamento dos motoristas e semáforos (SILVA,et al 2006).

Neste contexto, o objetivo do presente estudo foi comparar estes quatro *softwares* simuladores de trânsito: SUMO, ITSUMO, AIMSUN e VISSIM. Para isso, foram realizado dois estudos: uma análise da documentação fornecida pelos desenvolvedores destes *softwares*, contendo exemplos da utilização do *software* e também a análise de livros e artigos que contemplam o assunto. Por último foi avaliado as características dos simuladores, realizando dois estudos de casos em dois cenários selecionados. Como a literatura atual carece de trabalhos comparando estes *softwares*, o objetivo específico desta dissertação foi fornecer uma fonte de pesquisa sobre estes SSTs.



## 2 CONCEITOS BÁSICOS

Devido ao avanço na tecnologia e na engenharia de *software*, a simulação de sistemas de transporte se tornou a melhor forma de abordar uma análise sobre o trânsito. A habilidade de emular a variação do tempo sobre a enorme quantidade de variáveis que surgem ao redor do trânsito, fez dos SSTs uma ferramenta única capaz de lidar com esta complexidade.

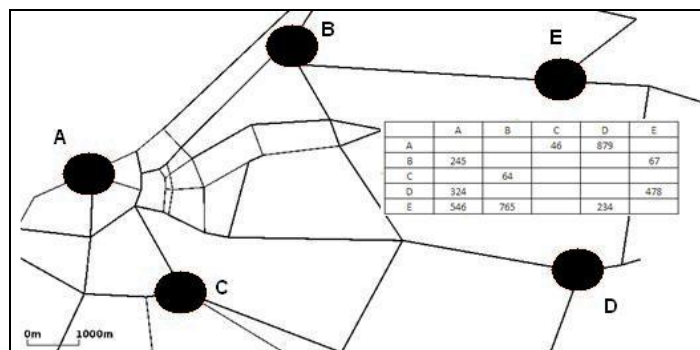
Esta ferramenta é utilizada para prever o resultado de um sistema real sobre diversas situações, com uma imensidão de dados, sem necessariamente utilizar o sistema real para saber como este se comporta. Construir um sistema assim exige compreender como o trânsito funciona, se comporta e evolui. É necessário ressaltar que não existe uma única forma correta de modelar um sistema (BARCELÓ, 2010).

Modelos formais são utilizados para descrever as propriedades e os relacionamentos entre os componentes do sistema. Modelos utilizando o formalismo da matemática para expressar os relacionamentos do sistema em termos quantitativos, por serem fáceis de manipular e de se obter respostas com um pequeno esforço computacional, foram utilizados para desenvolver os sistemas de trânsito analisados (BARCELÓ, 2010).

A alocação de demanda é o principal componente de qualquer simulador de trânsito, por diversas razões: serve de base para a simulação; prediz os efeitos das mudanças de longo prazo na malha viária (a construção de um centro comercial, por exemplo); prediz a dinâmica do tráfego durante eventos de curta duração (eventos culturais, shows, entre outros); ajuda a compreender as relações entre a dinâmica do tráfego e as decisões individuais dos usuários da rede; produz dados (através de um modelo matemático) a fim de cobrir a lacuna advinda da inexistência de dados reais (BAZZAN, KLÜGL, 2007). A abordagem mais utilizada para moldar esta demanda é através de uma matriz origem – destino (FIGURA 2.1).

A malha viária é dividida em regiões, cada uma gerando e atraindo viagens para um determinado propósito, durante um período de tempo. O percurso das viagens é

realizado sobre os caminhos disponíveis na cidade (as vias). As linhas e colunas da matriz representam as origens e os destinos respectivamente, e os valores da matriz são os números de viagens de origem até o destino.



**Figura 2.1. Representação de demanda em uma matriz origem – destino. Os círculos representam as regiões e as linhas, as vias.**

O modelo tradicional de demanda de tráfego é denominado de processo de quatro etapas (*4-step-process*) (Ortúzar and Willumsen, 2001). As viagens são utilizadas para encontrar uma solução de equilíbrio para o fluxo de veículos presentes nas vias de trânsito. O processo é dividido da seguinte forma:

- Geração da viagem: para cada origem, o número de viagens de saída é determinado. Da mesma forma que, para cada destino, o número de viagens de entrada é determinado;
- Distribuição da viagem: esta etapa liga as origens e destinos, ou seja, para cada origem é determinado qual será o destino. O resultado da distribuição de viagens é chamado de matriz origem-destino (OD), que especifica o número de viagens que vão de cada origem para cada destino.
- Modalidade de transporte: as viagens podem ser feitas por diferentes meios de transporte (a pé, bicicletas, motos, automóveis, ônibus, etc);
- Definição da rota: para cada viagem é atribuído um caminho que o veículo pode percorrer na via de trânsito. Esses caminhos são sensíveis a congestionamentos. Deve ser atribuído também um tempo de viagem para o percurso OD.

Verifica-se, assim, que o processo de “quatro etapas” é bastante simples e não contempla os problemas atuais de transporte. Trata-se de um processo que utiliza apenas fluxos de tráfego sem discernir os viajantes individualmente e não há dependência de tempo (BALMER, 2007).

A abordagem baseada em agentes para a geração de demanda de viagens, também conhecida como geração de demanda baseada em atividade ou *activity-based demand generation* (ABGD), pode resolver o problema de não haver uma modelagem consistente do processo de decisão dos motoristas, como ocorre no processo de quatro passos. Na abordagem ABDG, todos os motoristas de uma região são modelados individualmente no seu processo de tomada de decisão relativa ao tráfego, porém, torna-se difícil modelar esta tomada de decisão de forma consistente quando a simulação de tráfego é independente de tempo (BAZZAN, KLÜGL, 2007).

Para uma modelagem consistente, a alocação de tráfego necessita ser feita dinamicamente, esta alocação dinâmica de tráfego pode ser feita via *dynamic traffic assignment* (DTA), que separa efetivamente a simulação de escolha de rota e a simulação de fluxo de tráfego do processo de alocação em dois módulos diferentes. (BAZZAN, KLÜGL, 2007).

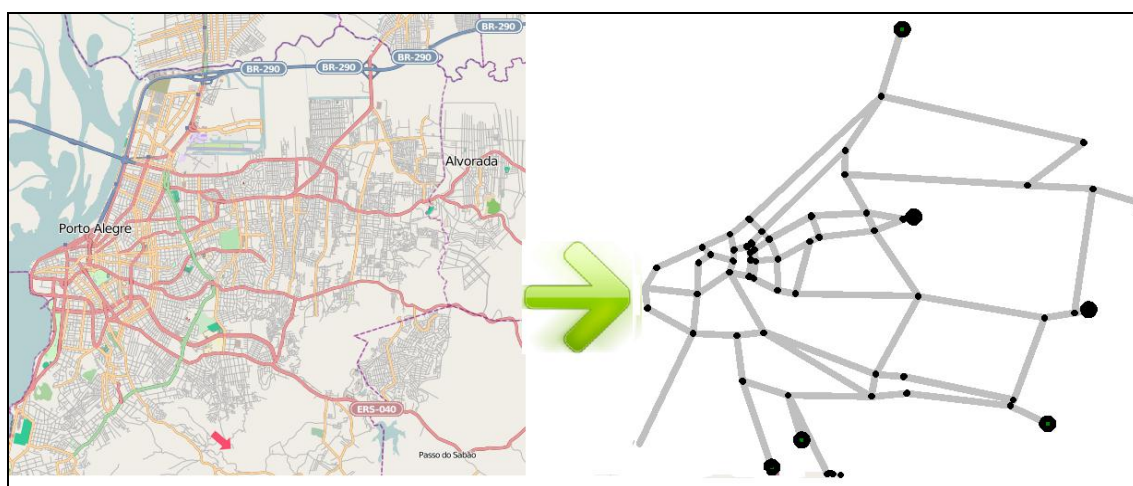
Trabalhos recentes acoplaram ABDG e DTA em matrizes OD, tornando estes métodos compatíveis com o processo de quatro passos. A desvantagem é que o conhecimento do agente a partir do ABDG não é mais conhecido no nível de DTA. Devido a estas dificuldades, a abordagem típica utilizada pelos SSTs é assumir modelos simples de comportamento humano, onde os motoristas se distribuem através de rotas com probabilidades baseadas nas previsões de tempo de percurso. (BAZZAN, KLÜGL, 2007).

Tráfego é o número de unidades (veículos, passageiros, carga, entre outros), que utilizam uma aresta de uma malha viária em um dado período de tempo. O controle do fluxo de tráfego tem como principais objetivos: prover o movimento ordenado do tráfego, aumentar a capacidade na intersecção (cruzamento de duas ou mais vias), interromper o tráfego principal a fim de permitir o fluxo do tráfego secundário. O controlador de tráfego mais antigo é o semáforo, datando de 1868. Este permite alternar o direito de passagem na zona de conflito de uma intersecção. Os SSTs permitem a

simulação deste componente essencial e a maioria dos softwares disponibiliza para o usuário maneiras de programar a lógica associada ao semáforo. (BAZZAN, KLÜGL, 2007).

Com o objetivo de minimizar os atrasos e tempos de viagem, as autoridades de transporte possuem a própria avaliação sobre como o sistema está executando, através de equipamentos ao longo das vias que determinam as medidas das variáveis de trânsito (volume, densidade, velocidade, entre outros). Diversos tipos de tecnologia realizam esta função, tais como câmeras de TV, processamento de imagens, etc. Por essa razão, os softwares simuladores de trânsito também devem retornar esta informação e, com isso, ser possível estudar um conjunto de políticas de controle de trânsito em busca de coordenar o trânsito de maneira eficiente, inclusive fazer previsões acerca deste. (BARCELÓ, 2010).

Grande parte dos SSTs disponíveis apresenta editores gráficos para transformar a informação geográfica contida nos mapas em malha viária. Dependendo do tipo de análise, mais precisa ou não, a informação desejada é adicionada à geometria. A FIGURA 2.2 retrata um exemplo. O lado esquerdo representa um mapa cartográfico e, a direita, a sua tradução em uma representação de malha viária em forma de grafo. Durante a tradução as intersecções são consideradas nodos, e as vias são transformadas em arestas.



**Figura 2.2 Representação de uma mapa cartográfico para um modelo digital**

É utilizada geralmente a abordagem objeto – entidade que caracteriza o sistema em termos de atributos. Em relação a simuladores de trânsito, a entidade é uma via

representando uma aresta e os atributos do objeto incluem a largura da via, limites de velocidade, quantidade de faixas, etc.

Alguns modelos necessitam detalhar ainda mais a topologia, configurando o conjunto de movimentos permitidos para cada intersecção. Este tipo de configuração é utilizado em situações que é necessário modelar funções de volume de tráfego associado a cada direção após a curva. (FIGURA 2.3.)

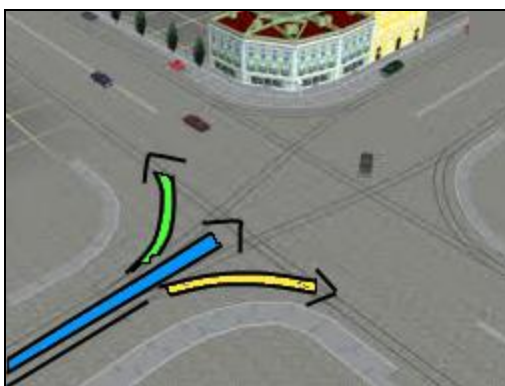


Figura 2.3: Representação gráfica de uma intersecção com opções de conversão

## 2.1 Nível de detalhe dos modelos de simulação de trânsito

O processo de simulação pode ser visto como uma técnica que imita no computador, a operação de um sistema do mundo real evoluindo ao longo do tempo. O aspecto básico em um processo de simulação está relacionado com o tempo. Existem duas abordagens metodológicas para lidar com o tempo em simulações: assíncronas e síncronas. Sincronizar o tempo na simulação corresponde em orientar o modelo para avançar de acordo com uma unidade de tempo ( $\Delta t$ ), o passo da simulação. O assíncrono, também chamado de simulações baseadas em eventos, são aquelas nas quais o tempo avança em uma quantidade variável, que corresponde ao instante no tempo que ocorreram eventos que alteraram o estado do modelo. As principais abordagens de simulação de trânsito estão baseadas no avanço síncrono do tempo, em pré-definidos passos de tempo (BARCELÓ, 2010).

Existem diversos meios de modelar o fluxo do tráfego para simular sua propagação temporal ao longo da malha viária. Baseado na teoria de fluxo de tráfego, os modelos de

simulação de trânsito podem ser divididos em macroscópicos, microscópicos e mesoscópico.

### 2.1.1 Modelo macroscópico

O modelo macroscópico tem como objetivo descrever a evolução no tempo e espaço das variáveis, caracterizando o fluxo macroscópico (densidade, volume e velocidade). As variáveis volume  $q(x,t)$ , velocidade  $u(x,t)$  e densidade  $k(x,t)$ , são definidas a cada instante no tempo  $t$  e cada ponto no espaço  $x$ . A equação principal representando formalmente esta teoria é a equação de conservação:

$$\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = 0$$

Similar à equação de continuidade em hidrodinâmica, esta representa formalmente o pressuposto de que, entre duas estações de contagem em uma seção de auto-estrada, sem entradas e saídas, o número de veículos é conservado.

Cada seção de vias é discretizada no tempo e no espaço, portanto, a simulação macroscópica pertence ao modelo de abordagem síncrona do tempo. Cálculos que são utilizados em dinâmica computacional de fluídos podem ser utilizados para solucionar as equações deste modelo (BARCELÓ, 2010).

Este modelo aborda o problema de fluxo de tráfego em um nível baixo de detalhes, não existindo interesse por cada unidade individual, e sim no processo com um todo.

### 2.1.2 Modelo Microscópico

Este modelo é apropriado para estudos que exigem um elevado nível de detalhamento, pois é baseado na descrição do movimento de cada veículo individualmente. Isto implica modelar ações como aceleração, desaceleração e troca de faixa (*lane-change*) de cada motorista.

A modelagem mais evidente para esta abordagem é chamado de *car-following*, baseado no fato de que um “veículo persegue o outro veículo”. De acordo com May (1990), teorias sobre como um veículo segue outro veículo foram desenvolvidas na década de 50 e 60,. Pesquisas no grupo *General Motors* levaram ao desenvolvimento de

modelos na forma de estímulo-resposta, no qual, a resposta é a reação de um motorista ao movimento do veículo imediatamente anterior a ele, no fluxo de tráfego. A resposta é sempre acelerar ou desacelerar em proporção à magnitude do estímulo no tempo  $t$ , e começa depois de um intervalo de tempo  $T$ , o tempo de reação do seguidor.

O caso mais simples corresponde à resposta representada pela aceleração ou desaceleração do motorista perseguidor. O estímulo é representado pela variação na velocidade relativa. Se  $X_n(t)$  e  $X_{n+1}(t)$  são posições do líder e do perseguidor, respectivamente, no tempo  $t$ , então o modelo básico é:

$$\ddot{x}_{n+1}(t+T) = \lambda [\dot{x}_n(t) - \dot{x}_{n+1}(t)]$$

A resposta depende do sinal do estímulo. Positivo se a velocidade relativa é positiva, isto é  $X_n(t) > X_{n+1}(t)$  e negativo caso contrário. Este modelo é conhecido como modelo linear *car-following* por causa da resposta diretamente proporcional ao estímulo. A integração desta equação retorna a velocidade do veículo  $n+1$ , a qual pode ser interpretada como a velocidade do fluxo de tráfego.

Na prática, a simulação microscópica é implementada em um simulador síncrono, ou seja, cada passo explora todas as entidades no modelo e atualiza os estados das entidades.

Geralmente os *softwares* realizam o processo descrito a seguir:

- a) Inicialização: define um tempo de ciclo para atualizar os caminhos na malha viária. Calcula o menor caminho para cada par da matriz OD.
- b) Repetir a inserção de veículos até carregar toda a demanda de tráfego na malha viária. A cada repetição, os seguintes passos ocorrem:
  - Atualização da posição de cada veículo no modelo:
    1. Determina o próximo movimento no passo atual
    2. Aplica o correspondente modelo: *lane-change*, *car-following*, etc.
    3. Calcula a nova posição ao final do passo
  - Coleta da estatística de acordo com o que foi definido

- Atualiza caminhos mais curtos

A diferença entre os modelos utilizados nos simuladores está na forma que implementam este processo genérico e combinam o núcleo dos modelos de *car-following* e *lane-change* (BARCELÓ, 2010).

### 2.1.3 Modelo Mesoscópico

Representa uma alternativa intermediária entre as duas dinâmicas anteriores. Consiste de uma simplificação que captura a essência da dinâmica de fluídos, consumindo menos processamento. Veículos individuais não são considerados, mas sim encapsulados em grupos, tornando a simulação mais eficiente que o modelo microscópico. As abordagens clássicas utilizam o tempo síncrono, orientado a passos, enquanto outras abordagens utilizam o tempo assíncrono, avançando o tempo em quantidades variadas dependentes da ocorrência de um evento (BARCELÓ, 2010).

As vias são modeladas dividindo-as em duas zonas: Zona livre para circular e zona enfileirada. A zona livre corresponde à parte onde os veículos não sofreram atraso por causa de um congestionamento em um semáforo à frente, por exemplo. Portanto a zona congestionada se refere aos veículos que não conseguiram sair da via por causa de um semáforo ou regras de trânsito, como preferenciais e placas de pare.

As intersecções são modeladas de acordo com a interação entre o fluxo de trânsito, podem ser nodos de transferência ou como servidor de filas, a fim de abordar explicitamente o comportamento dos semáforos e os atrasos que estes causam.

O movimento dos veículos é aproximado por simplificação do modelo *car-following*, que é compatível com o relacionamento macroscópico velocidade-densidade na via. Esta velocidade é utilizada para estimar o momento que o veículo pode sair da via, a menos que seja afetado por um congestionamento ao atingir a borda entre a zona livre e a zona congestionada. Na zona congestionada, a dinâmica do veículo é regrada pelo processo de desenfileiramento.



## 3 APRESENTAÇÃO DOS SIMULADORES

O primeiro item a ser configurado em um *software* simulador de trânsito deve ser a malha viária, essencial para a visualização da topologia, seguida da atribuição de veículos que percorrerão as vias. Nas simulações de trânsito, os motoristas assumem conceitos pré-programados, que orientam o movimento dos veículos tanto na longitudinal como na lateral. Neste capítulo, serão apresentadas as características de cada simulador.

### 3.1 SUMO

Projetado para as necessidades das pesquisas do Instituto de Sistemas de Transporte, este *software* segue três objetivos básicos: pouco consumo de memória, facilidade em estender a aplicação e velocidade nas simulações.

A utilização do SUMO se dá através de linhas de comando. Instalar o programa, simular um veículo percorrer uma via, configurar uma malha viária, entre outros, dependem destas linhas de comando. As entradas e saídas do programa são realizadas diretamente em arquivos XML.

Para manter a velocidade da simulação, não apresenta uma interface gráfica para construção topológica. Deve ser mencionado que a opção de utilizar linhas de comando foi uma decisão do projeto, visando manter a qualidade da simulação. Somado a isso, o *software* foi dividido em diversas partes, cada uma com um propósito que deve ser executado separadamente. Assim o usuário passa por diversas etapas até chegar à simulação final. Por apresentar aplicações menores, deve executar melhor que um grande programa monolítico com diversas funções.

#### 3.1.1 Características de movimento

No simulador SUMO foi utilizada uma versão modificada de tempo discreto e espaço contínuo do modelo *car-following*. Este modelo modificado, proposto por Kraub, primeiramente executa o movimento longitudinal e esta baseado na distância

que o carro perseguidor, o EGO, precisa parar atrás do veículo seguinte, o LÍDER, sem colidir com este. As máximas desacelerações e o tempo de reação do EGO são considerados, garantindo um comportamento livre de colisões (BARCELO, 2010).

No SUMO, como na maioria das simulações microscópicas, o movimento longitudinal é executado separadamente do movimento lateral e estes interagem muito pouco entre si. Uma inovação deste modelo é assumir que um motorista não é perfeito em realizar a velocidade desejada, pelo contrário, a velocidade realmente determinada é menor. Esta pequena diferença gera congestionamentos espontâneos e o chamado *slow-to-start* (início lento), características dos motoristas reais. O modelo é rápido durante a execução, principalmente devido ao pequeno número de computações, em comparações com outros modelos.

O modelo de movimento longitudinal é somente uma parte da modelagem no comportamento do motorista. Quando testado em um ambiente mais complexo, com vias de tamanhos diferentes, regras de trânsito complexas e diferentes rotas de veículos, alguns novos métodos foram implementados para a simulação funcionar. Primeiramente o veículo deve observar a infra-estrutura a sua frente e possíveis alterações na velocidade permitida, para assegurar um sistema livre de colisões.

Este modelo, apesar de ser rápido e já ter sido validado para algumas situações, apresenta alguns problemas: o modelo de imperfeição do motorista, por ser muito simples, não foi validado contra trajetórias reais e a simplificação para utilizar um tempo de reação de um segundo é uma problemática conhecida em cenários densos (BARCELÓ, 2010).

O modelo troca de faixa, conhecido por *lane-changing* vem evoluindo desde que o SUMO foi lançado. No modelo original somente foi abordada a decisão tática, baseada na concepção que o motorista deseja dirigir o mais rápido possível, desconsiderando a parte de troca entre faixas para continuar na rota.

A implementação atual do comportamento *lane-change* está baseado no modelo de Krajzewicz(2009), o qual computa um caminho válido através da malha viária. A faixa onde o veículo se encontra e as de sua proximidade são examinadas, assim, quando for necessário realizar uma troca de faixa, a distância da posição na qual a rota não pode ser

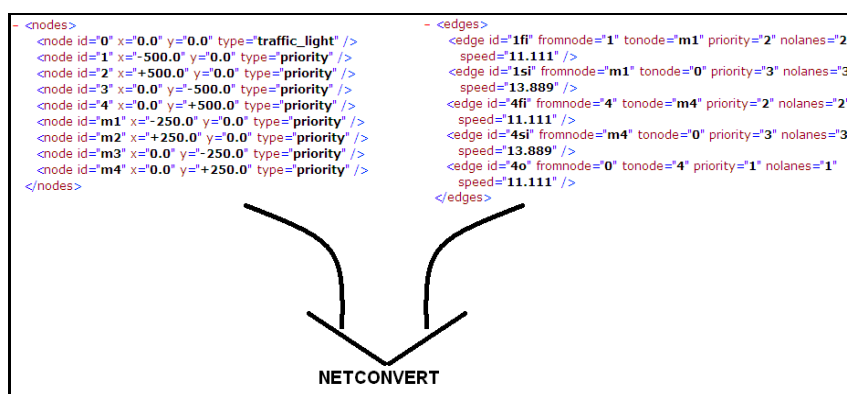
continuada, deve ser menor que uma distância assumida para a troca de faixa (BARCELÓ, 2010).

Para modelar o desejo de se mover mais rápido, o comportamento do motorista armazena a cada intervalo de tempo os benefícios de trocar de faixa durante o trajeto. Este benefício é a diferença entre a velocidade segura na faixa vizinha e na faixa corrente, dividida pela velocidade máxima que o veículo pode utilizar em condições de fluxo livre. A troca de faixa somente é iniciada se o valor absoluto da memória armazenada é maior que um determinado limite e se existe espaço na faixa desejada. Este modelo se comporta bem em cenários urbanos e rodovias rápidas, assumindo que os veículos escolhem antecipadamente as faixas, ocupando todas as disponíveis, porém, não considera o comportamento dos demais veículos e em algumas situações pode causar problemas (BARCELÓ, 2010).

### 3.1.2 Malha viária

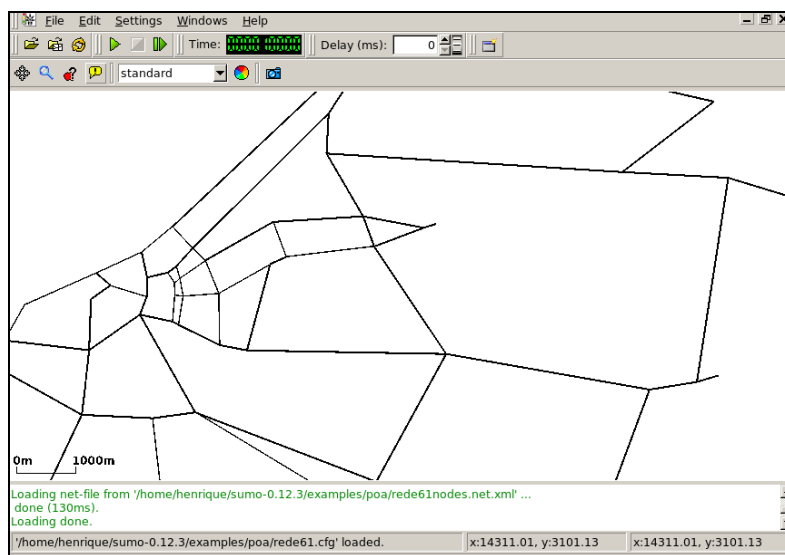
A principal forma de construir a malha viária é através da importação de modelos prontos de outros simuladores. A ferramenta *NETCONVERT*, que acompanha o *software*, realiza a importação de diversos formatos para os mapas do SUMO. Entre os possíveis formatos, estão o *VISUM* (2011), *TIGER* (2011), *VISSIM*, *Robocup Rescue League* (2011), *OpenStreetMap* (2011), e um formato nativo de XML de representação da malha viária (SUMO, 2011).

Para obter a malha viária completa, utiliza-se dois arquivos: um com as informações de cada intersecção (chamadas de nodos) da via e outro com as vias interligadas (as arestas). Após a configuração dos arquivos, o programa *NETCONVERT* gera um arquivo na linguagem do SUMO (FIGURA 3.1).



**Figura 3.1: Informações de nodos a esquerda e arestas a direita, transformado em malha viária pelo *NETCONVERT***

Outra função da ferramenta *NETCONVERT* é auxiliar o usuário, validando as entradas nos arquivos XML. Problemas com a malha viária final devem ser ajustados nos arquivos XML iniciais. A FIGURA 3.2 ilustra uma malha viária criada a partir do *software* SUMO.



**Figura 3.2: Exemplo de malha viária criada no *software* SUMO**

### 3.1.3 Controladores de tráfego

Um semáforo de comportamento simples pode ser inserido no SUMO durante as definições de um nodo, inserindo a palavra *traffic\_light* na configuração do tipo de nodo. A FIGURA 3.3 demonstra este processo:

```
<node>

    <node id="0" x="0" y="0" type="traffic_light"/>

</node>
```

**Figura 3.3: Configuração de um nodo com semáforo simples**

Este semáforo padrão não representa toda a complexidade de comportamento de um semáforo real. Para isso foi desenvolvido a ferramenta *TraCI* (*Traffic Control*

*Interface*), que permite a conexão de um programa externo, programado em *Python*, controlar o semáforo com a lógica desejada.

### 3.1.4 Atribuição de demanda

Escolher o método de atribuir o tráfego depende da pesquisa que está sendo conduzida. Como o SUMO foi projetado para simular uma população hipotética que consiste de veículos com distintas rotas e explícitos tempos de saída, o modelo de demanda nativo do SUMO é uma lista de veículos, com tempos de saída e dois pontos no mapa para origem e destino, chamado de *trip* (viagem). Utilizando uma das ferramentas que acompanham o SUMO, o *DUAROUTER*, é possível com um simples algoritmo de caminho mais curto, obter uma lista com a rota completa de todos os veículos. Esta rota completa é chamada de *route* (rota) e é armazenada em arquivos XML para carregar o programa.

Além desta maneira, o usuário pode atribuir tráfego com este simulador de diversas maneiras:

- Utilizando definições de fluxo que permitem agregar mais veículos em uma viagem;
- Assinalamento dinâmico de demanda (DTA): para obter um conjunto realístico de rotas através da malha viária;
- Utilizando a ferramenta *JTRROUTER*, tem-se a possibilidade de utilizar probabilidades em cada intersecção, o qual consiste em modelar o trânsito estatisticamente, definindo a possibilidade de um veículo optar por continuar o caminho ou contornar para outro local. Por ser um método pouco preciso, é utilizado para simulações pequenas, pois nada impede que os veículos andem em círculos;
- A ferramenta *OD2TRIPS* converte uma matriz OD (origem-destino) em viagens. Possui opções para definir o intervalo de tempo que esta matriz produz rotas para os veículos ou para distribuir estes valores uniformemente entre o intervalo dado. É possível dividir grandes matrizes em intervalos pré-determinados (entre as 13h e 15h da tarde, por exemplo) ou em um dia inteiro.

- Descrever a população na área analisada, também conhecido como atribuição de demanda baseado na atividade, suporta as atividades de trabalho, escola, caminhadas, bicicletas, carros e ônibus.
- Através da ferramenta *DFROUTER*, que utiliza injetores (*sources*) e coletores de veículos (*sinks*), os veículos entram na malha viária e a percorrem até encontrar um coletor. São utilizados para simular rodovias, pois em cenários urbanos podem ficar para sempre dentro da simulação.

### 3.1.5 Relatórios

Permite gerar uma quantidade considerável de resultados. Estes valores são escritos em arquivos texto ou através de *sockets* para o usuário. Os relatórios mais comuns, como as posições dos veículos, informações de viagens, rotas de veículos e estatísticas, estão disponíveis através de recursos na linha de comando. Os demais relatórios devem ser definidos através de arquivos adicionais, listados abaixo.

- a) Informação baseada no veículo:
  - Posições dos veículos ao longo do tempo: contém posições e velocidades de todos os veículos em todos os passos;
  - Posições dos veículos ao longo do tempo para um determinado tipo de veículo.
- b) Detectores simulados:
  - Detectores de indução simulada;
  - Câmeras de movimento;
  - Entrada e saída de veículos simulados.
- c) Valores para arestas ou faixas:
  - Via/faixa medidas de desempenho na rede;
  - Via/faixa emissão de poluição por veículos baseado em HBEFA (Manual de Fatores de Emissão para o Transporte Rodoviário);
  - Via/faixa emissão de ruído baseado em Harmonoise (escala de ruídos ambientais).
- d) Informação agregada ao veículo:
  - Informações de viagem agregada de cada veículo;

- Rotas de veículos.
- e) Estatísticas sobre o estado corrente da simulação.
- f) Relatórios de Semáforos:
  - Informações sobre os estados dos semáforos;
  - Informações sobre os ciclos de um semáforo específico, responsável por uma determinada intersecção;
  - Câmeras acopladas aos semáforos para captar veículos.

## 3.2 ITSUMO

Primeiramente chamado de SISCOT, o *Intelligent Transportation System for Urban Mobility*, foi proposto como uma alternativa aos *softwares* comerciais. Este simulador considera os veículos individualmente, resultando em um modelo mais complexo que o macroscópico (BAZZAN, KLÜGL, 2007).

O núcleo do simulador foi implementado em C++, uma linguagem amplamente conhecida e eficiente. Os principais módulos do simulador são o banco de dados, o núcleo da simulação, o controle de tráfego, a demanda e os resultados do programa como visualizações e estatísticas.

### 3.2.1 Características de movimento

Este simulador é baseado em um modelo microscópico de movimentação física, baseado no autômato celular de Nagel-Schreckenberg, onde cada via é dividida em células de tamanho fixo (normalmente 5m em cenários urbanos), assim, cada veículo ocupa posições discretas dentro destas vias. Este trecho é modelado como um vetor unidimensional, onde cada veículo tem uma velocidade, dada por um inteiro positivo  $v$  (dado em unidades de elementos ocupados em uma unidade de tempo, em geral um segundo). A posição do veículo ao longo da via é dado por um valor  $x$  e o número de elementos vazios na frente do veículo (posição relativa entre eles) é dado pela variável *lacuna(gap)*, em número de células. (ITSUMO, 2011)

No movimento dos veículos na longitudinal foi utilizada uma forma especial de *car-following*. Este calcula a aceleração e a velocidade do veículo a partir da posição

relativa ao veículo da frente, conforme a magnitude do afastamento (BAZZAN, et al 2010). Os motoristas obedecem aos limites de aceleração mínimos e máximos informados e tem como objetivo atingir a velocidade desejada atribuída na geração do veículo.

Diferentemente dos outros SST que se preocupam com o movimento dos veículos fundamentado em equações matemáticas que modelam somente o veículo líder, o ITSUMO foi desenvolvido sobre o paradigma baseado em agentes, onde cada motorista pode ser descrito o mais detalhado possível. O caminho a ser percorrido pelo motorista, assim como suas decisões em alterar a rota, pode ser abordado neste modelo de orientação a agentes. (BAZZAN, et al, 2011)

Assim como os motoristas, os semáforos também podem ser configurados nesta abordagem baseada em agentes, desde o modelo clássico baseado em fases de períodos pré-determinados até os avanços mais recentes, como os modelos de negociação, livre comunicação, teoria dos jogos, aprendizado por reforço, inteligência baseada em enxames entre outros. Diversos algoritmos podem ser desenvolvidos, conforme a intenção do usuário. Acompanha a distribuição do software alguns exemplos de implementações.

A simulação é composta de passos discretos de tempo, com uma série de atualizações nas posições dos veículos na malha viária. Assim, cada série de atualizações pode alterar o comportamento em tempo de execução do motorista ou de um semáforo (BAZZAN, A. et al, 2010).

### **3.2.2 Malha viária**

Toda a informação da topologia da rede de trânsito está armazenada na forma de arquivos XML. Entre os atributos principais deste arquivo estão as coordenadas cartesianas de cada intersecção, as vias com suas características e os semáforos com as definições de movimentos permitidos. Se a simulação contiver pontos que inserem e removem veículos, será armazenado também neste arquivo XML.

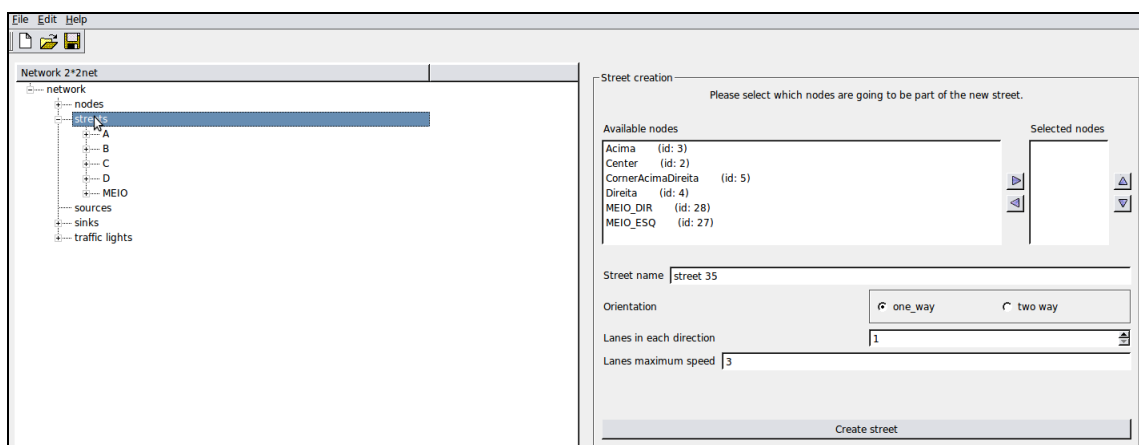
O sistema de vias foi dividido em quatro partes, nomeados em inglês como *street* (conjunto de vias), *section*(seção) que define as prioridades, *laneset*(via) delimitado



pelas intersecções, assim possuindo tamanho e velocidade máxima e *lane*(faixa) que possui como propriedade a largura.

Mapas podem ser criados manualmente com os arquivos XML ou utilizando o GUI (*Graphical User Interface*) fornecido pelo *software*. Pode ainda ser importado diretamente do *Open Street Map* (OSM, [www.openstreetmap.org](http://www.openstreetmap.org)), através do aplicativo “OSM2ITSUMO” que acompanha o programa.

A interface para o usuário preencher os valores das coordenadas de cada nodo, assim como definir as arestas de uma topologia está ilustrado na FIGURA 3.4. Através dos pontos cartesianos inseridos na topologia é possível construir vias apenas selecionando estes pontos, ilustrado na parte direita do *software*. Nesta mesma tela é definido o nome da via, a orientação, a quantidade de faixas e a velocidade máxima permitida.



**Figura 3.4: Interface para editar topologias no ITSUMO**

Os nodos e vias são criados na parte esquerda da tela e as propriedades de cada item são preenchidas no lado direito. Nesta mesma tela podem-se definir os sinais de controle das intersecções e os pontos injetores e coletores de veículos para elaborar o fluxo de veículos na malha viária. Após sair desta tela, uma pré-visualização do modelo é fornecida. A FIGURA 3.5 ilustra uma malha viária criada por este *software*.

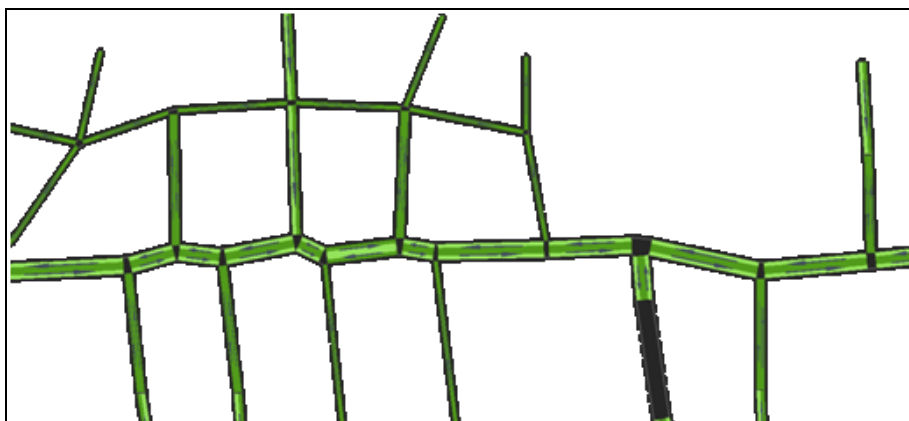


Figura 3.5: Malha viária obtida no IISUMO

### 3.2.3 Controladores de tráfego

Para inserir semáforos, existe uma aba no programa que trata somente disto. O usuário pode criar, manualmente, todas as definições dos planos semaforicos das intersecções, incluindo as direções de cada via e o tempo de ciclo. O programa informa quais as vias estão sendo visualizado no momento e o usuário define os movimentos permitidos e o tempo de cada fase. A FIGURA 3.6, demonstra a interface associada aos semáforos.

O *software* apresenta diversos exemplos em relação à utilização de agentes no controle dos semáforos, que podem ser utilizados pelo usuário como base em sua programação.

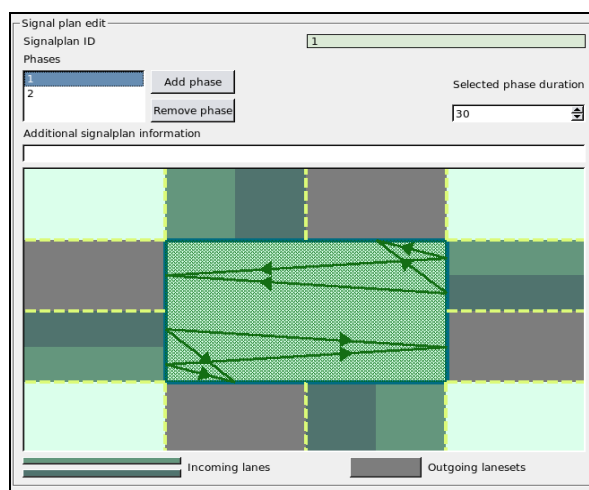


Figura 3.6: Configuração de semáforo no IISUMO

### 3.2.4 Atribuição de demanda

Este *software* não apresenta interface para lidar com matrizes OD, mas apresenta um exemplo de matriz pronta, seguida dos passos de como configurá-la.

A atribuição de tráfego pode ser realizada ainda por métodos manuais. Neste caso, o usuário, através da interface, seleciona nodos de intersecções e define-os em nodos de injetores e coletores de veículos. Em seguida deve definir probabilidades de inserir veículos na rede para um ou mais nodos escolhidos. O mesmo nodo pode originar 40% das viagens e coletar 50%, por exemplo. Pode-se ainda, inserir um fluxo constante de veículos a cada passo da simulação.

Os veículos não possuem um destino pré-definido enquanto trafegam, então, para cada intersecção é definido uma probabilidade para realizar a conversão do veículo na direção desejada. A FIGURA 3.7 representa um exemplo de configuração padrão, onde os veículos possuem probabilidades iguais de conversão (50%).

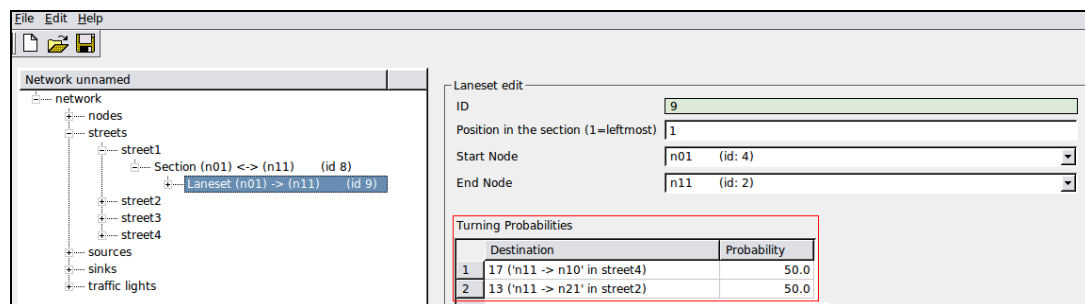


Figura 3.7: Configuração de probabilidades marcado em vermelho, para um “laneset” escolhido

### 3.2.5 Relatórios

Na tela inicial configuram-se quais resultados serão obtidos após a simulação. Diversos modelos de relatórios podem ser obtidos, entre eles:

- Lane sensor*: disponibiliza um mapa de células com a ocupação de veículos, densidade e velocidade média em cada faixa;
- Laneset sensor*: oferece a densidade das vias que estão conectados a um nodo controlado por um semáforo;
- Laneset sensor without border nodes*: Oferece a densidade das vias que estão conectados a um nodo controlado por um semáforo, porém, não estão conectados diretamente com um injetor de veículo;

- d) *Stoppedcars in laneset*: Retorna o total de veículos que ficaram parados nas vias durante cada passo da simulação;
- e) *Controlled lanes summary*: imprime informações resumidas, tais como, o grau de saturação (indica a porcentagem de determinada via atingir o máximo da capacidade), o número de veículos parados em filas, velocidade média, distância total percorrida;
- f) *Travel times*: imprime os tempos de viagem para cada rota planejada;
- g) *Average travel times*: acumula os valores verificados no item anteriormente e executa a média.

Estas opções são armazenadas em arquivos texto na pasta destinada pelo usuário. A própria simulação visual do ITSUMO é considerada um resultado do programa, pois é gravada em arquivo para o usuário visualizar em outro momento. Existe o resultado clássico, que consiste na simulação microscópica propriamente dita, e dois outros formatos: visualizar as vias com cores atribuídas de acordo com a densidade de veículos ou de acordo com a média de veículos parados.

### 3.3 AIMSUN

O AIMSUN foi formulado sobre os seguintes princípios básicos: integração, modularidade, extensibilidade e desempenho. Na integração, quanto mais o *software* é desenvolvido, novos modelos de conhecimento são aprimorados, exigindo uma programação robusta que acompanhe esta complexidade dinâmica, não exigindo assim, uma total reimplementação do *software*.

Todos os modelos compartilham a malha viária, as demandas OD e os caminhos registrados. Com esta integração, resultados de custos para percorrer uma malha viária, por exemplo, são completamente integrados entre o modelo mesoscópico e microscópico. A modularidade consiste em reutilizar os processos e as tarefas, estando de acordo com esta filosofia de integração.

O terceiro princípio lida com a extensibilidade do software, ou seja, a habilidade para os usuários programar extensões customizadas relativamente fáceis.

Enquanto os três primeiros princípios lidam com a arquitetura do *software*, o quarto princípio trata do desempenho, pois a velocidade de processamento dos chips aumenta a cada ano e novas técnicas de desenvolvimento também surgem como a programação paralela. O que era antes impossível, como modelar dinamicamente toda a cidade de

Nova York agora é possível. O desafio é manter um ritmo acelerado de acordo com as mudanças intensivas nos paradigmas de computação. (BARCELÓ, 2010).

Pode-se ainda, inferir sobre um quinto princípio, o da interoperabilidade, que é a habilidade do programa em trocar informações com outras aplicações em diversos formatos, como dados em formato GIS e construções em 3D.

### 3.3.1 Características de movimento

O AIMSUN foi implementado com o modelo *car-following* proposto por Gipps. Este é influenciado por parâmetros locais dependendo do tipo de motorista, as características da via e a influência dos veículos nas faixas adjacentes. O modelo consiste de dois componentes: aceleração e desaceleração. Se o veículo percorre livremente a via, ele tentará a máxima velocidade, de acordo com o veículo e a velocidade máxima da via. A desaceleração corresponde à limitação de seguir o carro líder (BARCELÓ, 2010).

O AIMSUN oferece simulações tanto no modelo microscópico quanto no mesoscópico. A diferença entre eles está no nível de abstração e no processo empregado para atualizar cada estado do veículo. O processo de simulação pode ser considerado como diversas fatias de tempo com adicionais eventos agendados. A cada intervalo de tempo, a simulação atualiza itens que não dependem de outras atividades, como os semáforos. Em seguida, um conjunto de repetições inicia para atualizar entidades como vias, intersecções e veículos e, por último, são inseridos novos veículos e os dados de relatórios. Cada veículo é tratado individualmente e contém atributos comportamentais, dados a eles quando entram no sistema, permanecendo constantes durante todo o traslado.

Os veículos quando executam o movimento longitudinal obedecem ao modelo *car-following*. A cada passo da simulação, a posição e a velocidade de cada veículo é atualizada, respeitando o seguinte critério: se é necessário trocar de faixa então o modelo de troca é efetuada, caso contrário, aplica-se o modelo *car-following*. Complementando este modelo, existe a influência das faixas adjacentes, isto é, quando um veículo está sozinho percorrendo a seção, com veículos lentos na faixa de menor velocidade ao lado, este ganha um aumento na sua velocidade.

O modelo de troca de faixa também foi proposto por Gipps. Este modelo é uma aproximação do comportamento do motorista, através da decisão: é necessário ou desejável trocar de faixa? Cada vez que um veículo é atualizado, a resposta depende da distância até a próxima curva e das condições de trânsito. Também é analisado se há benefício nesta troca, como aumento de velocidade na outra faixa ou uma fila menor. Por último é analisado se existe espaço para esta troca e se é possível efetuar-la com segurança (BARCELÓ, 2010).

### **3.3.2 Malha viária**

Ao iniciar um novo projeto, existe a opção de utilizar modelos pré-definidos que isentam definições básicas, como a largura da via, existência de acostamento, permitir ultrapassagens, entre outras. Estas podem ser elaboradas uma única vez, sem sofrer alterações contínuas. Modelos de veículos também estão pré-configurados, como a cor do veículo padrão, o modelo para a visualização 3D, aceleração máxima, tempo para frear, etc. Há ainda, a possibilidade de iniciar um projeto completamente novo.

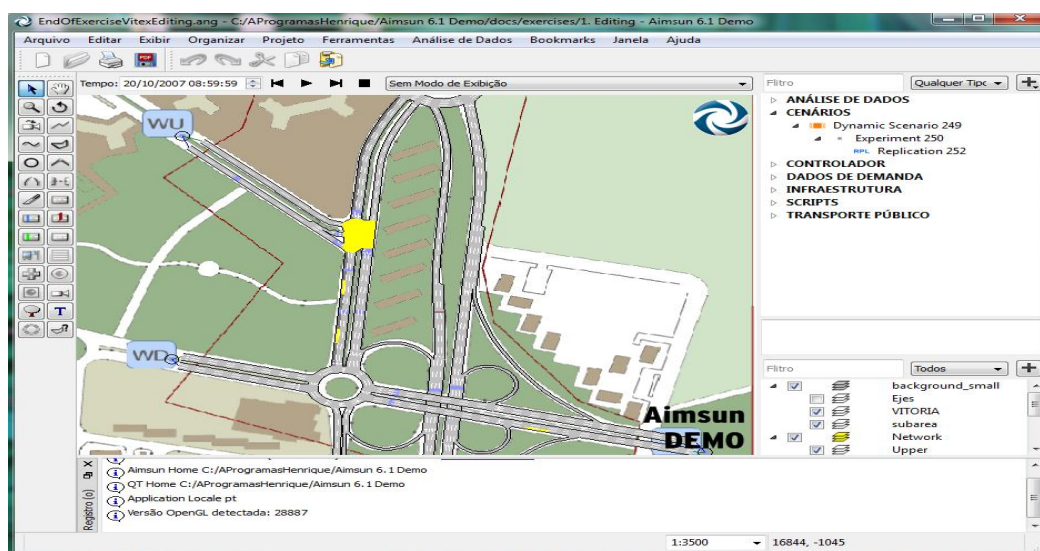
Para construir um modelo de simulação de transportes no AIMSUN é necessário adquirir informações da infraestrutura, como os formatos das vias, o número de faixas, a existência da faixa reservada, as permissões de conversão em cada intersecção e inclusive os cruzamentos de pedestres se for necessário esta análise. O programa também oferece um amplo conjunto de opções para o usuário definir o transporte público.

O AIMSUN suporta os seguintes formatos de arquivos para ser importado:

- Imagens aéreas, como PNG, JPG, BMP, GIF, SVG, SID e TIFF;
- Modelos 3D, como 3dsmax 3DS;
- CAD, como AutoCad DWG ou DXF;
- GIS, como OpenGIS GML, GPX e Google KML;
- Mapas digitais de GPS como os arquivos de mapa Navteq;
- Formatos de outras aplicações como VISUM, VISSIM, PARAMICS, TRANSYT e SYNCHRO.

Os formatos de arquivos listados acima estão em ordem de ascendência com a quantidade de informação útil que acompanha cada arquivo, ou seja, o mais proveitoso é listado por último. Os arquivos de imagens e CAD apenas podem ser utilizados como pano de fundo do desenvolvimento da topologia, enquanto os arquivos provenientes de outros simuladores podem criar uma malha viária completa. Ainda assim, é necessário fazer alguns ajustes nos parâmetros que somente o AIMSUN possui.

A interface gráfica permite definir todos os itens necessários para configurar a malha viária escolhida, como vias, intersecções, pontos de ônibus, etc. A tela inicial do software apresenta opções que auxiliam o usuário mais inexperiente. O programa oferece um tutorial (FIGURA 3.8) e um documento em anexo, onde constam os passos envolvidos no processo de gerar uma simulação.



**Figura 3.8: Tutorial do AIMSUN.**

Após criar as vias com a ferramenta *Section tool*, as intersecções são criadas com a ferramenta *Node Tool*. A FIGURA 3.9 ilustra a criação de um nodo. Neste caso, como existem quatro opções de vias, quatro conexões devem ser criadas. O sistema ilustra o caminho que o veículo vai percorrer através de flechas. Para evitar colisões é necessário definir alguma via como preferencial ou inserir um semáforo.

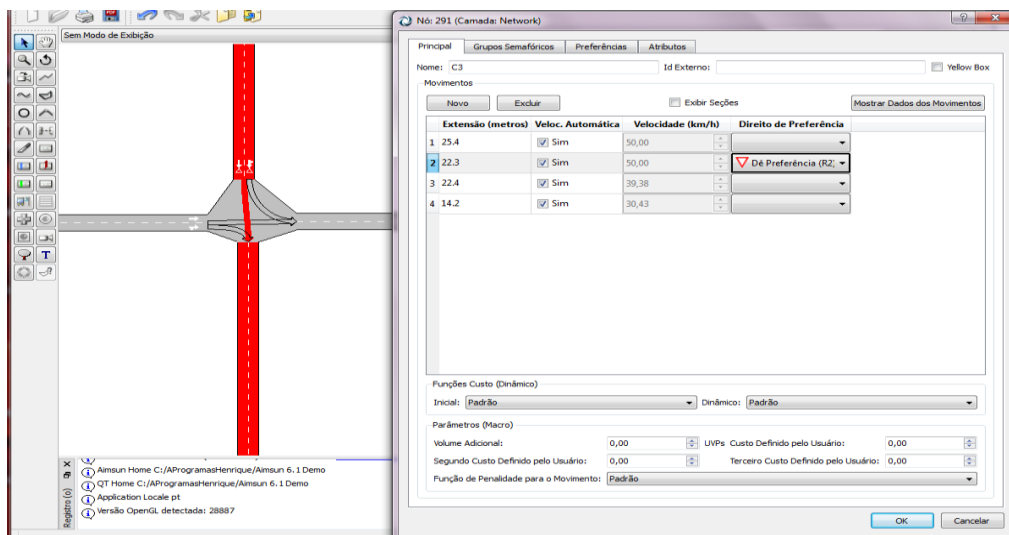


Figura 3.9: Nodo sendo criado

A interface de construção de malhas viárias auxilia nas diversas tarefas da modelagem. O software possui outros atributos, como criar camadas de rodovias com a possibilidade de sobrepô-las, os comandos copiar e colar e acesso aos comandos pelo teclado. A FIGURA 3.10 ilustra as visualizações 2D e 3D de uma simulação.

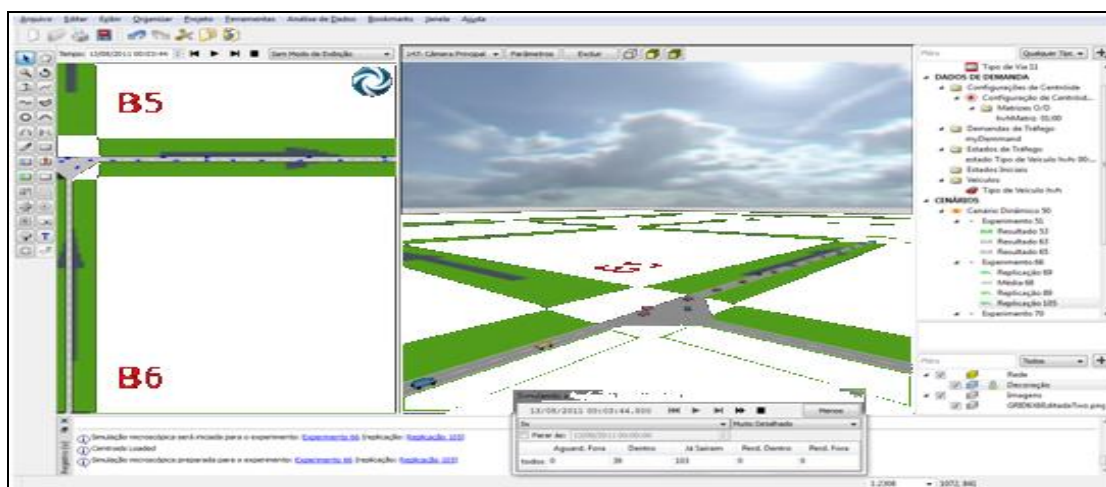


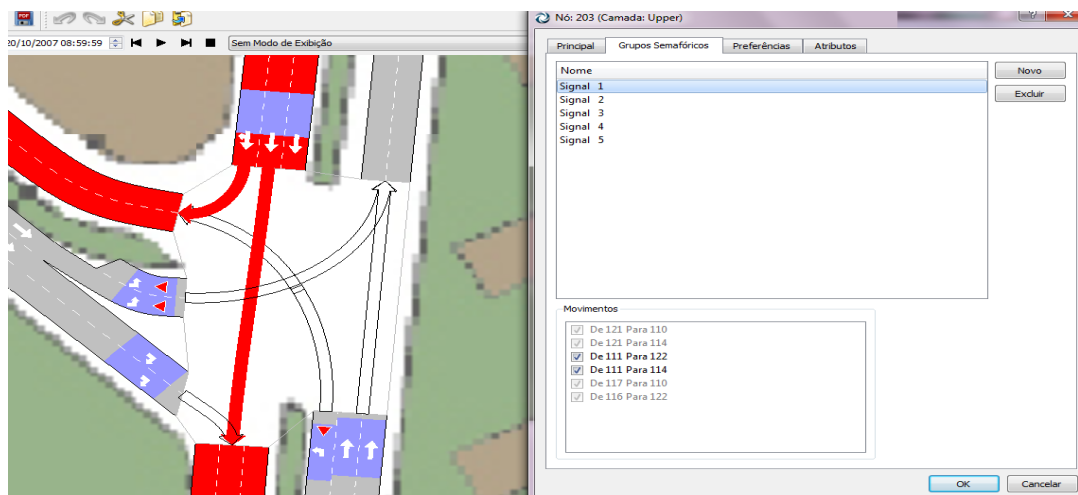
Figura 3.10: Visualizações 2D e 3D disponíveis no AIMSUN

### 3.3.3 Controladores de tráfego

Cada nó no AIMSUN permite a inserção de semáforos. A aba grupo semafóricos, nas configurações do nó, habilita a inclusão de movimentos permitidos em grupos. A FIGURA 3.11 ilustra a seleção de um grupo que permite somente dois movimentos.

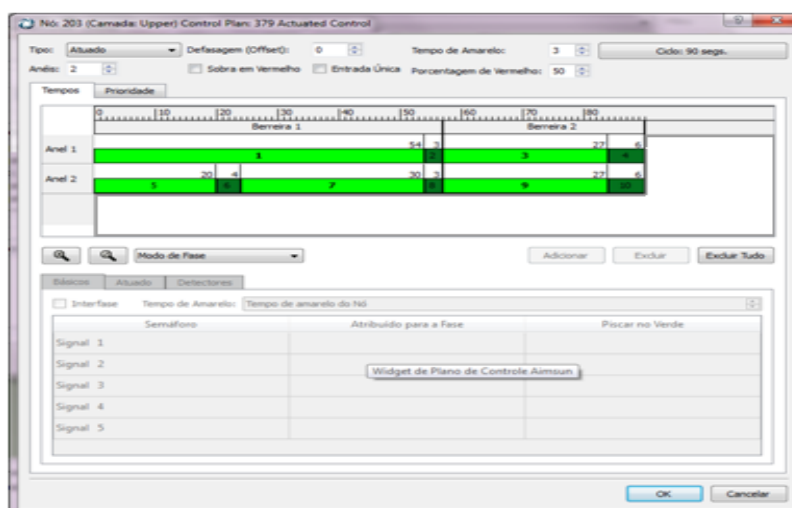


Mais adiante, quando for definido este plano semafórico, este grupo pode ser associado a uma fase do semáforo.



**Figura 3.11: Controladores de tráfego em uma intersecção no AIMSUN**

Os semáforos podem ser fixos ou atuados dinamicamente, dependendo das condições de trânsito. Cada semáforo deve ser configurado com o tempo de início e duração de cada plano, o tamanho do ciclo, a duração da fase amarela e os grupos associados em cada nó. Este plano de controle é configurado para um ou mais nodos seleccionados. A FIGURA 3.12 ilustra um exemplo de plano semafórico.



**Figura 3.12: Configuração de plano semafórico**

Os controles dinâmicos podem ser modelados utilizando detectores virtuais, que simulam os detectores de laço indutivo do mundo real. A interfase auxilia o usuário na programação de uma lógica de controle baseada nos padrões NEMA, SCATS,

UTOPIA, VS-PLUS, SICE e SCOOT. Caso a intenção é programar um controle customizado, a API (Interface de Programação da Aplicação) do AIMSUN fornece a comunicação entre a lógica e o simulador.

### 3.3.4 Atribuição de demanda

O AIMSUN suporta como opções de demanda de tráfego, matrizes OD ou estados de trânsito. Se a opção for matriz OD, é necessário possuir o zoneamento das áreas modeladas para correta inserção dos chamados centróides. Estes centróides são como pontos de entrada e saída de veículos da simulação. O usuário deve separar as matrizes em diferentes tipos de veículos e utilizar pequenas porções de tempo, algo em torno de 15 minutos. A FIGURA 3.13 ilustra um centróide em uma intersecção.

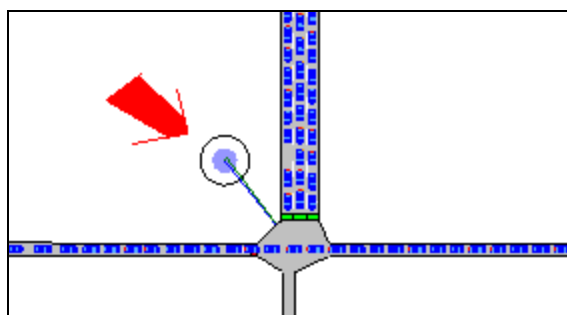


Figura 3.13: Centróide no AIMSUN

A interface para definir matrizes OD permite cópias entre programas de planilhas eletrônicas ou diretamente de um banco de dados via conexão ODBC. A FIGURA 3.14 ilustra a interface mencionada.

Viagens	284: NW	285: NE	286: WU	287: WD	288: E	289: S	Total
284: NW		190	95	285	475	1140	2185
285: NE	190		95	95	95	285	760
286: WU	95	95		95	190	190	665
287: WD	380	190	95		570	380	1615
288: E	380	190	190	285		285	1330
289: S	1045	665	95	285	380		2470
Total	2090	1330	570	1045	1710	2280	9025

Figura 3.14: Edição de uma Matriz OD no AIMSUN

Existem também os chamados estados de trânsito, que são como os injetores e coletores de veículos dos demais *softwares*. Este tipo de simulação não é aconselhado para cenários complexos.

Ferramentas de verificação automática conferem se os modelos implementados estão corretos. Verificam, por exemplo, se as vias estão conectadas ou se um par OD possui um caminho que não pode ser atingido.

### 3.3.5 Relatórios

Os resultados podem ser visualizados dentro do próprio *software* em tempo real, ou seja, os valores se alteram ao longo da simulação. Assim, pode-se acompanhar um veículo e visualizar todas as variáveis sendo processadas. Estes dados podem ser integrados ao banco de dados.

Ao final da simulação, o *software* oferece um sumário com todos os valores processados, que podem ser acessados a qualquer momento através da aba Séries Temporais, presente em qualquer entidade da malha viária após a simulação. Assim, após a simulação é possível escolher uma determinada via e saber todas as informações a respeito dela. Possui ainda, opções para fazer médias e desvios.

Esta ferramenta inclui gráficos, tabelas e a possibilidade de configurar quais critérios se quer definir para a pesquisa. A FIGURA 3.15, apresenta um gráfico gerado a partir da consulta realizada na Séries Temporais.

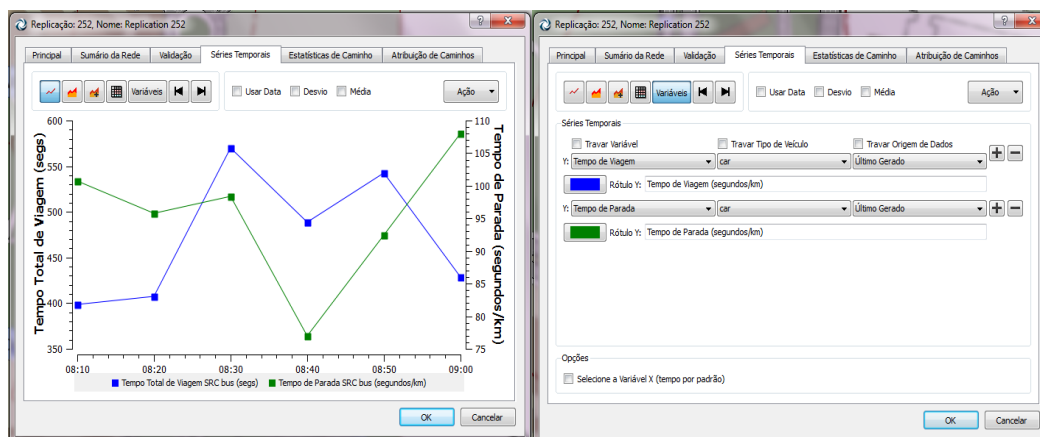


Figura 3.15: Séries plotadas no AIMSUN a partir das informações personalizada em Séries Temporais

Além da possibilidade de definir relatórios, podem-se configurar novas maneiras de visualizar a simulação. As vias e veículos podem ficar coloridos ou mostra legendas de acordo com as escolhas feitas. A FIGURA 3.16 ilustra uma visualização padrão, no caso, as vias coloridas de acordo com o fluxo.

Ao iniciar um novo projeto, o software insere automaticamente os seguintes modos: Veículos perdidos, coloridos por tipo (como veículos e transporte público), de acordo com a definição de rotas; vias coloridas de acordo com fluxo, qualidade de serviço e densidade.

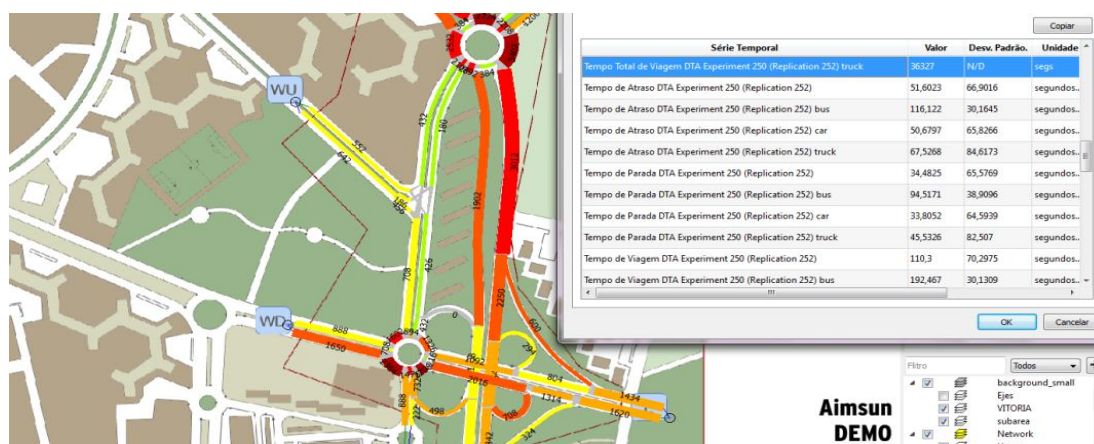


Figura 3.16: Vias coloridas de acordo com fluxo

### 3.4 VISSIM

VISSIM é um simulador de tráfego microscópico, multi-modal, que oferece aplicações urbanas e interurbanas, integrando o transporte público e privado. Condições complexas de trânsito são visualizadas de acordo com modelos de tráfego realísticos (BARCELO, 2010).

Este software possui uma longa história de desenvolvimento, onde basicamente o seu primeiro estudo iniciou em 1974 com uma tese de PhD de Wiedemann, descrevendo o movimento de cada veículo na longitudinal, conhecido também como modelo *car-following*. Somente em 1994 após diversas pesquisas e implementações, foi lançada a primeira versão comercial do VISSIM na Alemanha para análise de capacidade em intersecções sinalizadas com controle atuado.

O *software* foi desenvolvido em C++, considerando a programação orientada a objetos. Três modelos matemáticos compõem o núcleo do *software*: o primeiro representa os aspectos técnicos e físicos envolvendo os transportes, o segundo a demanda para gerar as pessoas e veículos percorrendo o sistema e o último, um modelo que controla o trânsito com semáforos, regras de preferências, etc. Este três modelos interagem entre si.

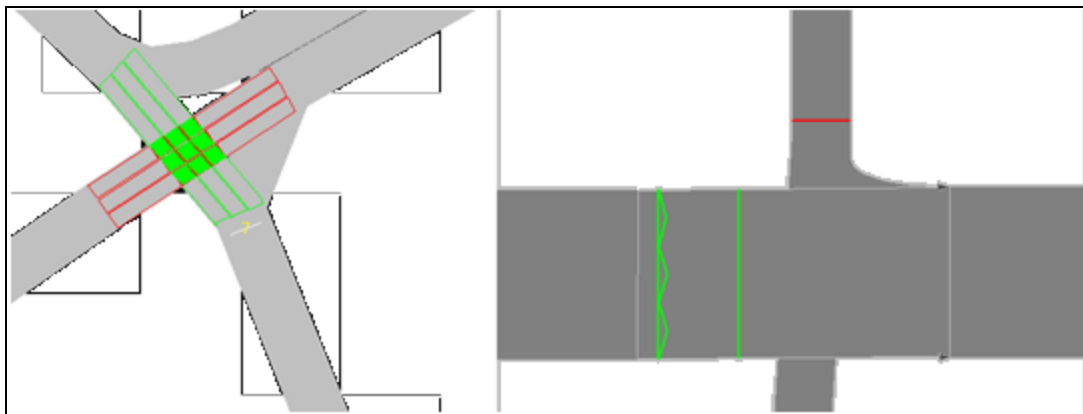
### 3.4.1 Características de movimento

O modelo de movimentação dos veículos no VISSIM baseia-se no modelo de Michaelis e, assim como os anteriores, utiliza a abordagem *car-following* para orientar os demais veículos. O motorista reconhece alterações no aparente tamanho de um veículo líder à medida que se aproxima deste. Diferenças de velocidade são percebidas através de alterações no ângulo de visão.

Os movimentos laterais são compostos de três comportamentos diferentes: seleção da faixa, troca de faixa e o movimento lateral contínuo. Para a seleção da faixa, o motorista escolhe a faixa com a melhor interação disponível, contanto que ele esteja distante da próxima intersecção relevante. Cada intersecção mantém um controle da distância e esta deve ser obedecida para ser efetuada a conversão (BARCELO, 2010).

A conversão de faixa só será efetuada se o veículo conseguir entrar no espaço desejado. Caso isso não ocorra, o motorista toma decisões que envolvem fazer o veículo ao lado desacelerar, podendo até mesmo, desacelerar seu próprio veículo. Quanto mais próximo da conversão, mais agressivo o veículo fica, podendo inclusive parar.

Existem duas formas de modelar as regras de preferências em cruzamentos não sinalizados. A FIGURA 3.17 representa as opções disponíveis. À esquerda, foi configurada uma área de conflito. O usuário deve apenas selecionar qual via possui a permissão de passagem. Com isso, o motorista decide o momento de atravessar, através de um plano para cruzar a área de conflito pela observação dos veículos que se aproximam da via principal. Caso ele tenha que parar ou diminuir a velocidade, devido a outros veículos, ele calcula mais tempo para cruzar a área de conflito (PTV, 2011).



**Figura 3.17: Intersecções não sinalizadas.** A figura da esquerda representa uma área de conflito padrão, e da direita, a linha pare (vermelha) associada a marcas de conflito.

A figura da direita representa a segunda maneira de configurar a regra de permissão de passagem, utilizada somente se a área de conflito não possui os resultados esperados. A regra consiste de uma linha de pare associada a uma ou mais marcas de conflito. Quando um veículo deseja entrar na via e percebe outro veículo se aproximando da linha verde, a uma velocidade  $x$ , este calcula se é possível efetuar a transição sem causar acidentes.

O movimento lateral não pode se tornar uma manobra instantânea, por isso existe o comportamento lateral contínuo. A escolha desta posição lateral é decidida no local onde o motorista possui o máximo intervalo até uma possível colisão. Para encontrar esta posição, as vias são divididas em linhas virtuais do lado esquerdo e direito do veículo, incluindo uma distância segura. Dependendo do veículo, o motorista aplica uma velocidade lateral sobre a faixa escolhida, de forma a manter esta distância segura dos demais veículos.

O VISSIM também oferece uma modelagem da movimentação dos pedestres, recurso que os demais simuladores aqui analisados, não oferecem.

### 3.4.2 Malha viária

No desenvolvimento de uma simulação de trânsito complexa é necessário replicar o modelo da infraestrutura em escala. O visual padrão das vias é em cor cinza com setas indicando a direção, porém, isto pode ser configurado para outras opções de visualização. O botão principal do mouse está configurado para ser o direito.

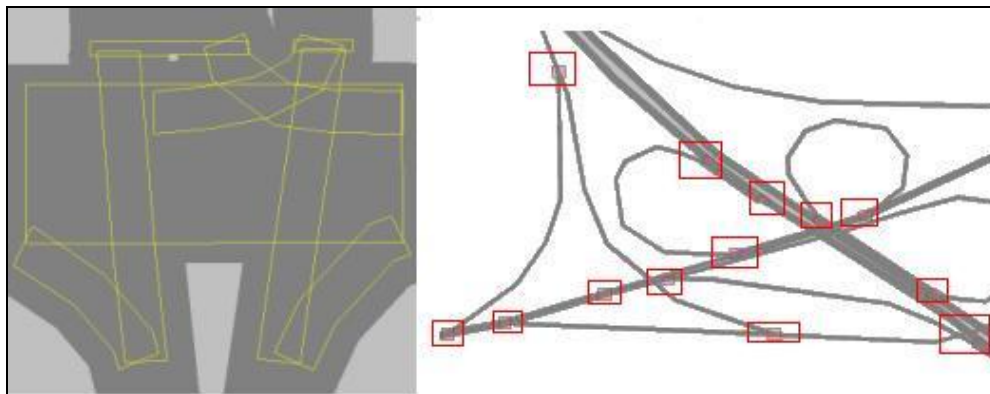
Cada módulo é utilizado separadamente, ou seja, ao se escolher uma das ferramentas, o usuário tem acesso somente aos recursos alterar, inserir e apagar, relacionados a este módulo.

Topologias com escala são importadas de *softwares* como o GIS, fotografias aéreas ou desenhos CAD, que se comportam como papel de parede do *software*. Podem-se utilizar métodos para definir a escala dessas figuras depois de inseridas no programa. As vias são construídas como descrito a seguir:

Selecionando o botão arestas e nodos, o *software* habilita o desenho das vias e dos conectores. Pressionado e arrastando o botão direito do mouse sobre uma área vazia, será construído uma via. Caso a intenção seja conectar duas vias, basta selecionar a origem e conectar com o destino, ligando-as. Cada item criado abre uma janela para se especificar as propriedades.

A malha viária é representada como um grafo, onde os nodos são intersecções e as arestas são segmentos de vias. O VISSIM não exige uma definição explícita dos nodos, pois para modelos com rotas aleatórias, este nível de detalhamento não é necessário. Geralmente nos softwares as intersecções são modeladas como nodos no grafo da malha viária. No VISSIM existem as interconexões que conectam as vias, transformando a intersecção em um conjunto de pequenas vias sobrepostas. Foi utilizado este método para reduzir o tempo de computação. Porém, é necessário utilizar os nodos nas áreas da topologia onde os caminhos podem divergir. Visando a atribuição de tráfego dinâmica, estes caminhos devem estar definidos para não ocorrer incoerências.

Cada via possui propriedades como unicidade, coordenadas planares, número de faixas, tamanhos e veículos habilitados a percorrê-la. Estes itens fazem parte da classe principal. A FIGURA 3.18 da esquerda ilustra a representação de um nodo e da direita, um exemplo de configuração de nodos para uma malha viária.



**Figura 3.18: Representação de um nodoem segmentos de via, à esquerda, e malha viária com diversos nodos à direita.**

Os próximos itens fazem parte da classe opcional. Itens como placas de orientação e regras informativas são colocados em cada faixa para determinar sua função completa. Assim, um veículo ao passar por um sinal de limite de velocidade, ajusta sua velocidade conforme a orientação. Dentro desta classe existe mais uma subdivisão, que determina lugares especiais ao longo da via, como estacionamentos, paradas de ônibus ou os detectores que registram os dados da via. Os estacionamentos têm uma importância especial e será abordado na próxima seção, pois determinam a origem e destino dos veículos quando as rotas são definidas dinamicamente.

### **3.4.3 Controladores de tráfego**

A FIGURA 3.19 ilustra como é possível modelar um controlador de tráfego complexo através do VISSIM. Este cruzamento foi configurado utilizando as fórmulas pré-programadas fornecidas com o software. Funções próprias de lógica de programação no semáforo podem ser definidas através de programação em linguagem C ou Pascal.





Figura 3.19: Exemplo de controladores configurados no VISSIM. As linhas vermelhas representam estes controladores.

O software *Vehicle-Actuated Signal Control* (VisVAP) acompanha o programa, possui uma interface gráfica para desenvolver lógicas através de diagramas. A FIGURA 3.20 ilustra a configuração dos intervalos de tempo e a interface gráfica.

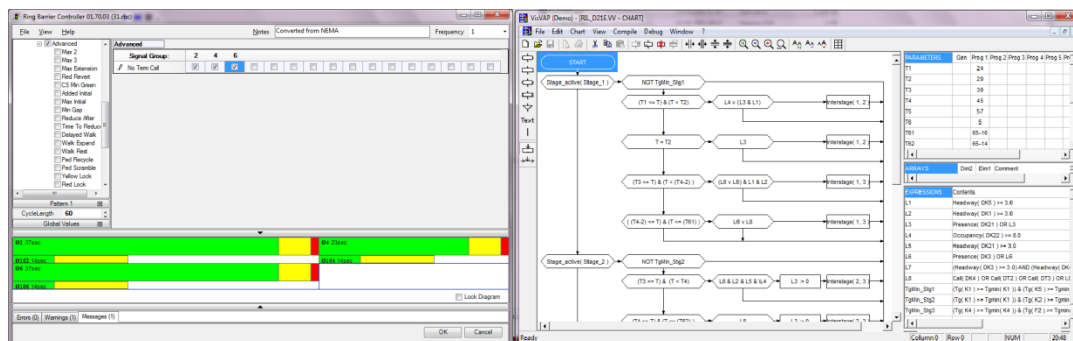


Figura 3.20: Semáforos com intervalos de tempo a esquerda e interface gráfica VisVAP à direita

As intersecções com semáforos, no modelo mais simples, possuem tabelas onde o usuário pode definir o tempo de ciclo de cada fase para determinado semáforo. Desenvolvido para ser um controlador simples, esta tarefa é feita pelo software VISSIM, que inclui diversas opções, incluindo otimizações para dar preferência à determinada via.

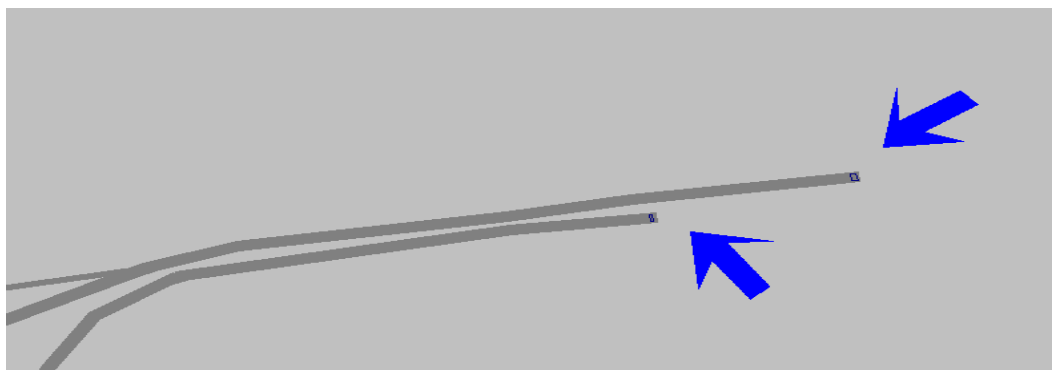
### 3.4.4 Atribuição de demanda

Existem tipos de veículos que devem ser determinados de acordo com algumas propriedades, como o tamanho do veículo, peso, a velocidade e aceleração que atingem. Para os modelos 3D, devem ser definidos as cores e os formatos dos veículos, além de uma série de outros atributos opcionais.

Os veículos são gerados randomicamente em vértices de entradas ou estacionamentos. Estes fluxos são definidos individualmente para múltiplos períodos de tempo. Como o número de saídas é dado em um intervalo  $[0,t]$ , seguindo a distribuição de Poisson, que significa  $\lambda t$ , então o intervalo de tempo  $x$  entre dois veículos sucessivos vai seguir a distribuição exponencial que significa  $1/\lambda t$ .  $\lambda$  é a medida em veículos por hora. Se o volume de trânsito exceder a capacidade da via, os veículos são empilhados fora da simulação até obter espaço (BARCELO, 2010).

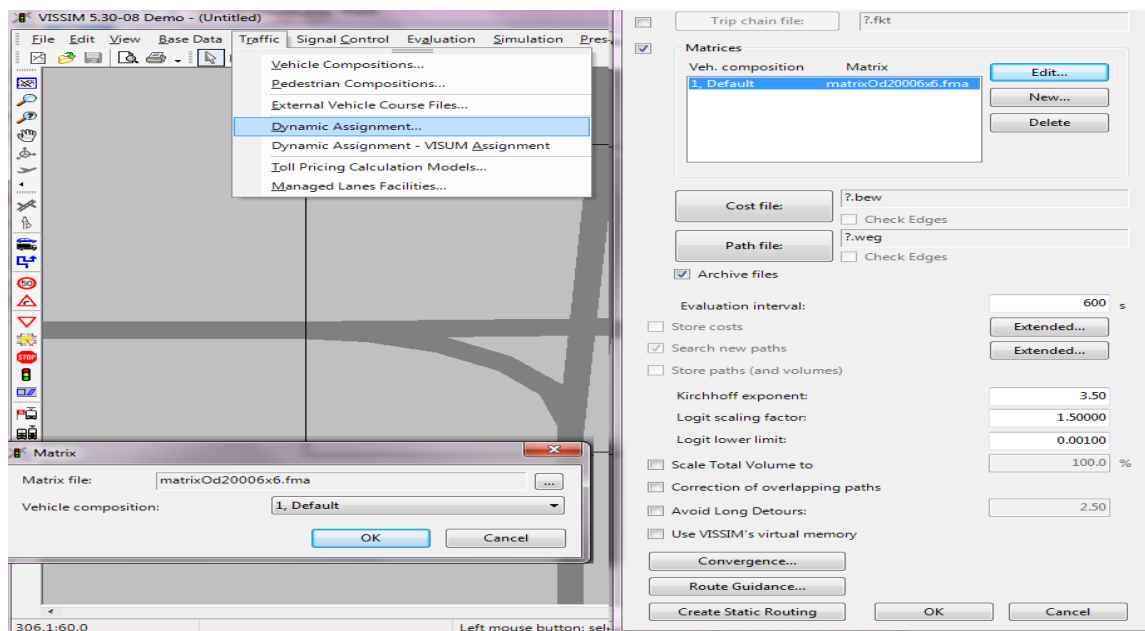
Definir a matriz OD no VISSIM não envolve apenas o local onde o motorista deve chegar, mas como ele irá processar a informação da topologia para escolher o melhor caminho até atingir o objetivo. O motorista armazena as informações de simulações anteriores e, cada vez que percorrer novamente a malha viária, utilizará estes resultados. Assim, a simulação não é executada apenas uma vez, mas diversas vezes, permitindo aos motoristas escolherem as rotas baseado em um custo atribuído a cada viagem.

O conceito escolher a melhor rota está baseado em grafos, onde os nodos constituem os pontos dos grafos. No VISSIM, estes nodos são demarcados por formas geométricas na intersecção escolhida pelo usuário (conforme FIGURA 3.18). Através destes nodos é possível definir a direção que o motorista pode seguir até chegar ao próximo nodo. Concluída esta etapa, devem ser inseridos os chamados estacionamentos, que podem ser abstratos ou se comportar como um simulador de estacionamento (FIGURA 3.20).



**Figura 3.21: Estacionamentos inseridos nas bordas da malha viária**

Cada estacionamento abstrato deve se conectar a um nodo previamente inserido e, através das propriedades do estacionamento, definido como uma zona. Na FIGURA 3.21 os estacionamentos estão inseridos antes dos nodos. Os veículos acessam este estacionamento para entrar e sair da simulação. Após estas definições é possível inserir uma matriz OD. O simulador não possui uma interface para editar as matrizes, ficando a cargo do usuário, utilizar um editor de texto para inserir as informações necessárias no formato exigido. Este arquivo deve conter as zonas envolvidas nas rotas, o intervalo de tempo selecionado e o fluxo de veículos para este intervalo de tempo, entre as zonas. Este arquivo deve ser inserido conforme indica a sequência de imagens da FIGURA 3.22. Concluídas estas etapas, o *software* está pronto para iniciar a simulação.



**Figura 3.22:** Sequência de ações para inserir matriz OD no VISSIM (*Traffic*→*DynamicAssignment*→*Matrices*→*New*→*Matrix file*)

Para o transporte público, são definidas as entradas e paradas ao longo da via de acordo com uma tabela. Possui ainda uma opção para o usuário informar se o trânsito é baseado na direita ou na esquerda.

### 3.4.5 Relatórios

O programa oferece diversos formatos para visualizar os resultados da simulação, inclusive em tempo real. Existe a possibilidade de analisar relatórios em arquivos, armazenar a informação em banco de dados ou em modo janela, observando os dados no momento da coleta em uma janela separada. É possível configurar o formato destas informações, como a linguagem e as unidades.

Para visualizar os resultados em tempo real, o *software* oferece informações tanto para os veículos como para os pedestres, pressionando-se o botão do mouse sobre estes. Devem ser determinados os semáforos que serão analisados, assim como as vias e os tempos de viagem. A FIGURA 3.23 a seguir ilustra alguns exemplos.

Veh. No. 4	
Vehicle Type	1
Vehicle Type	Car
Link	2
Lane	1
v	[km/h] 18.2
vDesired	[km/h] 45.6
a	[m/s <sup>2</sup> ] -2.0

Signal changes				
SimSec	CycleSec	SC	SG	Aspect
123.0	33.0	5	3	red
123.0	33.0	5	2	red
124.0	34.0	3	13	red/amber
125.0	35.0	10	99	red
125.0	35.0	5	23	green
125.0	35.0	3	13	green
126.0	36.0	3	3	amber
127.0	37.0	5	24	green
128.0	38.0	5	21	green
128.0	38.0	3	24	green
128.0	38.0	3	7	red/amber
129.0	39.0	3	7	green
129.0	39.0	3	3	red

**Figura 3.23: Informação de veículos a esquerda e semáforos a direita**

O módulo de resultados tem acesso a todas as entidades da simulação e diversas informações podem ser retiradas. Entre as opções estão: informações dos veículos, dados dos nodos, desempenho da topologia, paradas de ônibus, tempos de viagens, troca de faixa, tamanho das filas, tempo de atraso, entrada de veículos e caminhos assinalados dinamicamente.

## 4 ESTUDO DE CASO

Para comparar a eficiência e praticidade dos *softwares* apresentados, estudos de casos foram elaborados a partir de um objetivo comum entre eles. Foram analisadas as principais propriedades de cada um até atingir este objetivo.

A configuração de hardware da máquina onde foram executadas as simulações possui processador Inter Core 2 Duo E8400 3.00GHz, memória de 4 GB, Sistema Operacional Windows 7 de 64 Bits e Ubuntu 10.04 LTS LucydLynx.

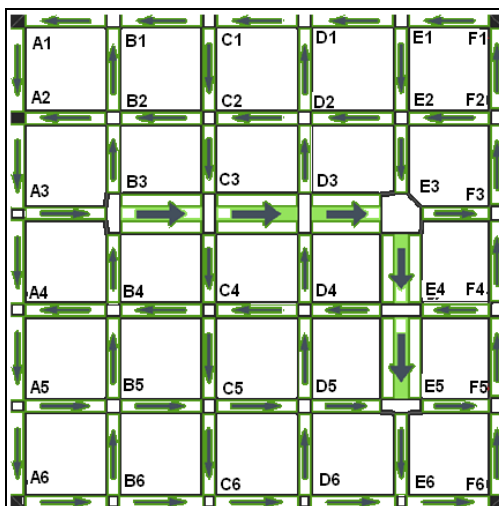
Os softwares analisados estão na seguinte versão:

- SUMO – Versão 0.12.3
- ITSUMO – Sem versão
- AIMSUN – Versão 6.1 Demo
- VISSIM – Versão 5.30 32 Bit Demo

### 4.1 Grade 6x6

O primeiro item avaliado nestes *softwares* foi a elaboração de uma malha viária simples, que consistia de 36 pontos distribuídos uniformemente em um quadrado externo de 1500m<sup>2</sup>, subdividido internamente em quadrados de 300m<sup>2</sup>, semelhante ao ilustrado na FIGURA 4.1.

Os nodos foram representados por letras e números (A a F na horizontal e 1 a 6 na vertical). A primeira coluna representa a via de letra A, dividida entre os nodos A1 até A6, com 300m de distância entre eles. A via A1 para A2 recebe o nome “A1toA2” e assim por diante até a letra F, concluindo a malha viária.

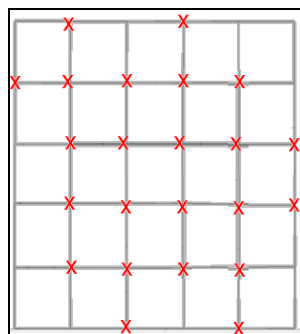


**Figura 4.1:** Exemplo de uma Grade 6x6, desenvolvida no *software* ITSUMO

Os controladores de tráfego de cada simulador também foram analisados. Foram avaliados a dificuldade para inserir semáforos e a possibilidade de construir uma lógica própria de controle.

Como existem intersecções sinalizadas e não sinalizadas, analisaram-se os dois modelos. Primeiramente, avaliaram-se as intersecções sem sinalização e logo em seguida, a mesma intersecção com a inserção de sinalização. As não sinalizadas foram configuradas através de marcas nas vias, assim, os motoristas, de acordo com estas marcas, decidem se podem ou não fazer a conversão para a via desejada, apenas com o campo de visão que possuem no momento. Durante as simulações sem semáforos, foi determinado que as vias horizontais possuíam a preferência.

As intersecções sinalizadas são representadas por semáforos inseridos ao longo da malha viária. Geralmente os simuladores disponibilizam um modelo básico, deixando a critério do usuário, escrever uma lógica mais complexa. No estudo de caso, todos os cruzamentos marcados com um X, como na FIGURA 4.2, foram selecionados para incluírem semáforos. Foram definidos ciclos de 100 segundos, divididos igualmente em 47 segundos para cada fase, com a inserção da fase de transição com sinal amarelo de 3 segundos, antes da troca do sinal verde para o vermelho. Devido às características da malha viária, os cruzamentos exigiram somente duas fases para controlar o trânsito.



**Figura 4.2: Intersecções selecionadas para inserir semáforos**

O fluxo de veículos (atribuição de tráfego) foi atribuído por matrizes de origem e destino (OD), pois, além de geralmente serem suportadas pela maioria dos softwares, aproximam-se da realidade. Os nodos das intersecções entre as linhas e colunas da malha viária foram simulados como uma região. Deste modo, estes nodos comportam-se como injetores e coletores de veículos. O nodo F3 foi definido como ponto principal de destino, simulando um centro comercial, por exemplo, reunindo uma grande quantidade de pessoas.

O conteúdo da matriz OD foi o mesmo utilizado para os softwares, porém, cada software possui uma maneira específica de processar esta informação. No estudo de caso, as quantidades totais de veículos utilizadas foram de 2000, 3000 e 4000, incrementando os valores desta matriz proporcionalmente.

Os programas VISSIM e SUMO, seguiram a convenção de matriz OD da FIGURA 4.3. Onde o primeiro parâmetro corresponde ao formato utilizado na matriz OD e, em seguida, é definido o tipo de veículo que obedece a esta matriz. Foi necessário configurar como o tempo é distribuído ao longo da matriz e também um fator de distribuição na existência de mais de uma matriz. Os parâmetros seguintes configuram o número de regiões e os respectivos nomes. No final do arquivo foram distribuídos os valores para cada par origem/destino.

No AIMSUN esta configuração é realizada na própria interface e no ITSUMO não existem estas configurações, sendo permitido, somente, alocar a quantidade de veículos informados na matriz.





```

<nodes>
  <node id="A1" x="2000" Y ="3500"/>
  <node id="A2" x="2000" Y ="3200"/>
  <node id="A3" x="2000" Y ="2900"/>
  <node id="A4" x="2000" Y ="2600"/>
  <node id="A5" x="2000" Y ="2300"/>
  <node id="A6" x="2000" Y ="2000"/>
  <node id="B1" x="2300" Y ="3500"/>
  ...

```

**Figura 4.4:** Arquivo `6x6.nod.xml` que contém o código para configurar nodos no SUMO. Coordenadas do plano cartesiano (x e y) são configuradas nesta etapa, indicando a localização de cada nodo

```

<edges>
  <edgefromnode="A1" id="A1toA2" tonode="A2"/>
  <edgefromnode="A2" id="A2toA3" tonode="A3"/>
  ...
  <edgefromnode="E3" id="E3toA4" tonode="E4" nolanes="3"/>
  ...

```

**Figura 4.5:** Arquivo `6x6.edg.xml` que contém o código para configurar arestas no SUMO

Utilizando a ferramenta *NETCONVERT*, que acompanha o *software*, criou-se uma malha viária através da linha de comando no diretório ‘src’(FIGURA 4.6).

```

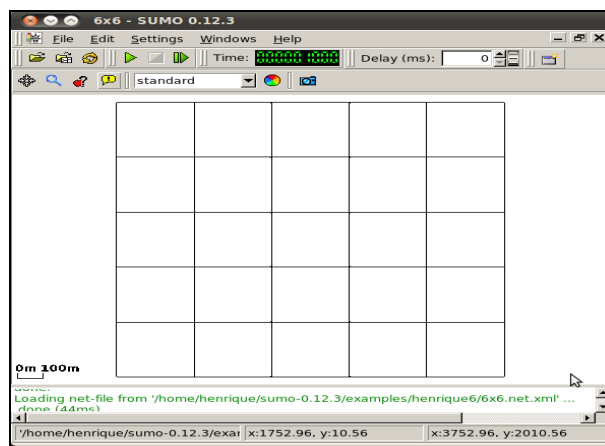
./netconvert -n 6x6.nod.xml -e 6x6.edg.xml -o 6x6.net.xml

```

**Figura 4.6:** Código para o *software NETCONVERT* construir a malha viária, onde: *netconvert* é a ferramenta, `6x6.nod.xml` é o arquivo com as posições dos nodos, `6x6.edg.xml` é o arquivo das arestas e `6x6.net.xml` é a malha viária resultante

A opção `-o` representa o nome do arquivo que o usuário deseja obter ao final da execução (no caso do estudo, “`6x6.net.xml`”). As convenções dos nomes de arquivos, como `.nod`, `.edg` e `.net` foram definidas pela equipe de desenvolvimento do SUMO, sendo boas práticas de programação.

A FIGURA 4.7 ilustra o final do processo.



**Figura 4.7: Grade 6x6 construída no SUMO**

#### 4.1.1.2 Controladores de Tráfego

Nos testes propostos, o simulador não apresentou falhas com a inserção de semáforos.

#### 4.1.1.3 Atribuição de demanda

O *software* não apresenta uma interface para definir matrizes OD, porém, possui uma ferramenta executada por linha de código que processa o trabalho repetitivo de definir origens e destinos para os veículos. Existe uma convenção em torno dos arquivos XML que define como nomear as regiões sobre a malha viária.

As regiões foram definidas através de um arquivo “6x6.taz.xml”. Para cada região foi definido um nome, uma origem e destino. Como o SUMO considera somente as vias na definição de uma região, para transformar um nodo da grade 6x6 em uma região, foram atribuídas às vias que chegam e saem deste nodo, formalizando assim a região. Foi definido, por exemplo, que o nodo A1 possui conexão com o nodo B1 e A2, ou seja, somente é permitido ir de B1 para A1 e de A1 para A2 (FIGURA 4.8).

Pode-se ainda, definir a influência na tomada de decisão dos motoristas sobre qual caminho escolher, através da propriedade peso (*weight*), definida para cada aresta. Este parâmetro é utilizado quando se define a probabilidade de utilizar uma região. No caso proposto, foi mantido um número igual para todos (1).

```

<tazs>
  <Taz id="A1">
    <tazSource id="AltoA2" weight="1.0"/>
    <tazSink id="B1toA1" weight="1.0"/>
  </taz>
  ...

```

**Figura 4.8: Regiões formadas no SUMO e suas conexões através do arquivo 6x6.taz.xml**

O programa oferece somente a alternativa para importar formatos de matriz OD de outros softwares, Assim, foi utilizado o padrão do VISSIM.

Através da ferramenta OD2TRIPS e dos arquivos XML parametrizados corretamente, foi obtido um arquivo de saída com as definições de viagens de cada veículo.

Utilizando a matriz do VISSIM, definida na FIGURA 4.3, foi criado o arquivo 6x6.odm.txt, onde se definiu a atribuição de tráfego no SUMO, ou seja, as viagens (*trips*). O comando ilustrado na FIGURA 4.9 foi utilizado no sistema operacional:

```
./od2trips -n 6x6.taz.xml -d 6x6.odm.txt -o 6x6.trp.xml
```

**Figura 4.9: Comando para gerar *trips* no SUMO, onde: od2trips executa a ferramenta, 6x6.taz.xml é o arquivo que contém as regiões configuradas, 6x6.odm.txt contém a matriz e 6x6.trp.xml das viagens**

A FIGURA 4.10 ilustra o arquivo obtido através da ferramenta *OD2TRIPS*. Este inclui informações de origem, destino, zonas, velocidade, faixa e tipo de veículo.

```

<tripdefs>
<tripdef id="6" depart="1.00" from="A2toA3" to="E6toF6" type="Normal" fromtaz="2" totaz="36" departlane="free" departspeed="max"/>
<tripdef id="12" depart="7.00" from="B1toA1" to="E6toF6" type="Normal" fromtaz="7" totaz="36" departlane="free" departspeed="max"/>
<tripdef id="2" depart="11.00" from="A1toA2" to="E6toF6" type="Normal" fromtaz="1" totaz="36" departlane="free" departspeed="max"/>
<tripdef id="0" depart="17.00" from="A1toA2" to="B2toA2" type="Normal" fromtaz="1" totaz="2" departlane="free" departspeed="max"/>
<tripdef id="4" depart="19.00" from="A2toA3" to="B1toA1" type="Normal" fromtaz="2" totaz="1" departlane="free" departspeed="max"/>
<tripdef id="10" depart="27.00" from="B1toA1" to="A1toA2" type="Normal" fromtaz="7" totaz="2" departlane="free" departspeed="max"/>
<tripdef id="7" depart="37.00" from="A2toA3" to="C1toB1" type="Normal" fromtaz="2" totaz="7" departlane="free" departspeed="max"/>
<tripdef id="1" depart="42.00" from="A1toA2" to="E5toF5" type="Normal" fromtaz="1" totaz="35" departlane="free" departspeed="max"/>
<tripdef id="8" depart="43.00" from="F6toF5" to="F6toF5" type="Normal" fromtaz="36" totaz="35" departlane="free" departspeed="max"/>
<tripdef id="9" depart="48.00" from="B1toA1" to="B1toA1" type="Normal" fromtaz="7" totaz="1" departlane="free" departspeed="max"/>
<tripdef id="11" depart="49.00" from="B1toA1" to="F6toF5" type="Normal" fromtaz="7" totaz="35" departlane="free" departspeed="max"/>
<tripdef id="5" depart="52.00" from="A2toA3" to="F6toF5" type="Normal" fromtaz="2" totaz="35" departlane="free" departspeed="max"/>
<tripdef id="3" depart="57.00" from="A1toA2" to="B2toB1" type="Normal" fromtaz="1" totaz="7" departlane="free" departspeed="max"/>

```

**Figura 4.10: Trips/Viagens criadas automaticamente com o software *OD2TRIPS***

#### 4.1.1.4 Relatórios

Durante a atribuição de demanda, foi apresentado um arquivo de configuração que incluía um parâmetro de coleta de informações da simulação. Para o estudo de caso, o arquivo “netStateOutputConfig.add.xml” foi configurado para incluir um relatório de

resultados ao final da simulação, de nome “oDetectNetTraffic”. Dentro deste arquivo foi inserido o seguinte comando (FIGURA 4.11):

```
<meandata-edge id="oDetectNetTraffic" file="outputLogTraffic"/>
```

**Figura 4.11: Arquivo adicional para incluir resultados na simulação**

A informação é armazenada no arquivo XML configurado no parâmetro *file* da FIGURA 4.7. Este arquivo contém diversas informações sobre as vias da malha viária. Entre elas, o *traveltime*, que é o valor aproximado que os veículos levaram para atravessar cada via durante a simulação. O relatório obtido no estudo de caso está ilustrado na FIGURA 4.12.

```
<netstats>
  <interval begin="0.00" end="10000.00" id="oDetectNetTraffic">
    <edge id="A1toA2" sampledSeconds="4720.03" traveltime="30.25" density="1.60" occupancy="1.20" waitingTime="2.00" speed="9.76" departed="1" />
    <edge id="A2toA3" sampledSeconds="15127.30" traveltime="51.98" density="5.09" occupancy="3.82" waitingTime="458.00" speed="5.72" departed="1" />
    <edge id="A3toA4" sampledSeconds="1915.49" traveltime="51.36" density="0.66" occupancy="0.49" waitingTime="1107.00" speed="5.66" departed="1" />
    <edge id="A3toB3" sampledSeconds="20791.36" traveltime="69.64" density="7.00" occupancy="5.25" waitingTime="2320.00" speed="4.27" departed="1" />
    <edge id="A4toA5" sampledSeconds="119096.33" traveltime="708.13" density="40.09" occupancy="30.06" waitingTime="96208.00" speed="0.42" departed="1" />
  </interval>
</netstats>
```

**Figura 4.12: Relatório de resultados da simulação adquirido com o “oDetectNetTraffic”**

Este arquivo de viagens “6x6.trp.xml” passou, ainda, por uma segunda ferramenta, o *DUAROUTER*, que calculou a rota mais curta entre a origem e o destino, gerando o arquivo final das rotas, “6x6.rou.xml” (FIGURA 4.14), através do comando da FIGURA 4.13:

```
./duarouter -n 6x6.net.xml -d 6x6.trp.xml -o 6x6.rou.xml
```

**Figura 4.13: Comando para gerar as rotas com origem e destino**

O arquivo representado na Figura 4.144 contém todas as arestas que o veículo deve percorrer para chegar até o destino. Contém outras propriedades definidas automaticamente, como o momento de saída, a velocidade e um código identificador.

```

<routes>
  <vtype accel="2.6" decel="4.5" id="Normal" maxSpeed="80.0" sigma="0.0" color="1,1,0"/>
  <vehicle id="6" type="Teste" depart="1.00" departlane="free" departspeed="max" fromtaz="2" totaz="36">
    <route edges="A2toA3 A3toB3 B3toC3 C3toD3 D3toE3 E3toE4 E4toE5 E5toE6 E6toF6"/>
  </vehicle>
  <vehicle id="12" type="Teste" depart="7.00" departlane="free" departspeed="max" fromtaz="7" totaz="36">
    <route edges="B1toA1 A1toA2 A2toA3 A3toB3 B3toC3 C3toD3 D3toE3 E3toE4 E4toE5 E5toE6 E6toF6"/>
  </vehicle>
  <vehicle id="2" type="Teste" depart="11.00" departlane="free" departspeed="max" fromtaz="1" totaz="36">
    <route edges="A1toA2 A2toA3 A3toB3 B3toC3 C3toC4 C4toC5 C5toD5 D5toE5 E5toE6 E6toF6"/>
  </vehicle>
  <vehicle id="0" type="Teste" depart="17.00" departlane="free" departspeed="max" fromtaz="1" totaz="2">
    <route edges="A1toA2 A2toA3 A3toB3 B3toB2 B2toA2"/>
  </vehicle>
  <vehicle id="4" type="Teste" depart="19.00" departlane="free" departspeed="max" fromtaz="2" totaz="1">
    <route edges="A2toA3 A3toB3 B3toB2 B2toB1 B1toA1"/>
  </vehicle>
  <vehicle id="10" type="Teste" depart="27.00" departlane="free" departspeed="max" fromtaz="7" totaz="2">
    <route edges="B1toA1 A1toA2"/>
  </vehicle>
</routes>

```

Figura 4.14: Arquivo 6x6.rou.xml - Rota completa de cada veículo, calculada a partir do software DUAROUTER

Ao se adicionar o arquivo de rotas (“6x6.rou.xml”) e as definições da malha viária (“6x6.net.xml”) ao arquivo de configuração principal, o software iniciou a simulação. Este arquivo de configuração inclui as informações criadas para a simulação, como o tempo de simulação e arquivos adicionais que incluem novas funcionalidades (FIGURA 4.15).

```

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/6x6Configuration.xsd">
  <input>
    <net-file value="6x6.net.xml"/>
    <route-files value="routes4000.rou.xml"/>
    <additional-files value="netStateOutputConfig.add.xml"/>
  </input>
  <time>
    <begin value="0"/>
    <end value="1800"/>
  </time>
</configuration>

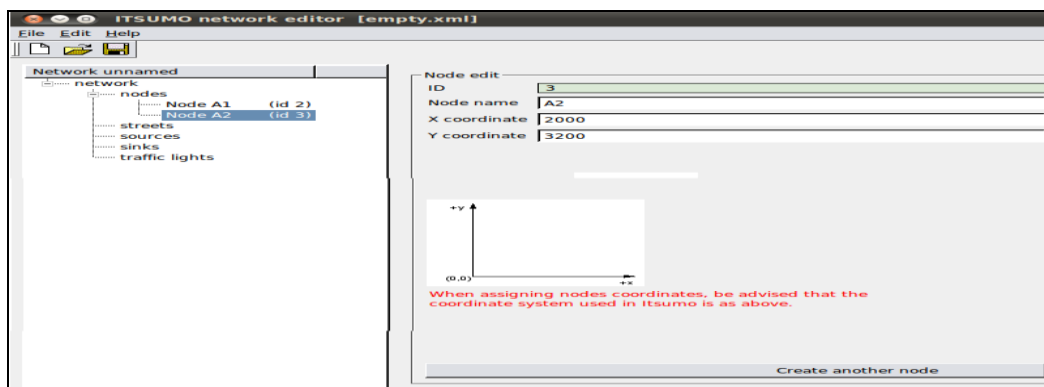
```

Figura 4.15: Arquivo principal do SUMO, responsável pelo início da simulação

## 4.1.2 ITSUMO

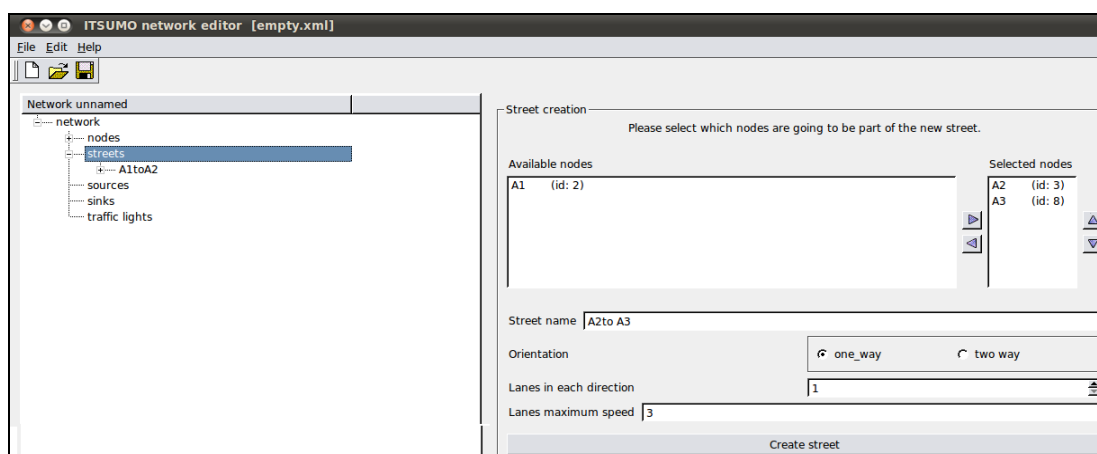
### 4.1.2.1 Malha viária

Na tela inicial do software existe o botão *Edit/Create Net* que disponibiliza a interface para desenvolver a malha viária. O software atribuiu um número identificador único para cada nodo criado, coube ao usuário, inserir um nome e uma coordenada em X e Y para este nodo (FIGURA 4.16).



**Figura 4.16: Definição de nodos no ITSUMO com as coordenadas em x e y do plano cartesiano**

Após a seleção dos nodos que definiriam a via (chamada *street*), e da configuração de propriedades como nome, sentido e quantidade de faixas, o recurso *Create Street* do *software* finalizou a via. A FIGURA 4.17 ilustra como as vias foram criadas.



**Figura 4.17: Inserindo vias no ITSUMO através do recurso *Create Street***

A FIGURA 4.1, que ilustrou uma Grade 6X6, foi obtida por estes passos descritos, sendo, portanto, o resultado da malha viária utilizando o *software* ITSUMO.

#### 4.1.2.2 Controladores de tráfego

Nos testes executados neste trabalho, o *software* comportou-se normalmente, sem ocorrência de falhas durante a simulação com os semáforos. Um problema encontrado na interface ocorreu quando se trocou o tempo de ciclo do semáforo, pois, em alguns momentos, esta alteração não foi processada.

#### 4.1.2.3 Atribuição de demanda

Não existe interface para configurar a demanda no ITSUMO. Os parâmetros são informados juntamente com o arquivo de configuração principal do ITSUMO, ilustrado na FIGURA 4.18.

```

<config>
<file>input/TCCgrid6x6_net.xml</file>
<steps>500</steps>
<odmatrix>input/TCCgrid6x6_ODM2000.xml</odmatrix>
<odsize>2000</odsize>
<default_deceleration>0</default_deceleration>
<sensor_interval>5</sensor_interval>
<agent_interval>600</agent_interval>
<car_max_speed>3</car_max_speed>
<cell_size>5</cell_size>
<iteration_length>1</iteration_length>
<drivers></drivers>
<sensors>
<sensor>
<name>visualization_microscopic_level</name>
<file>../../output/TCCvisualization_microscopic_level.txt</file>
<state>ON</state>
<time_tag>NO</time_tag>
</sensor>
<sensor>
<name>laneset_sensor</name>
<file>/home/henrique/itsumo/output/TCClaneset_sensor.txt</file>
<state>ON</state>
<time_tag>NO</time_tag>
</sensor>
</sensors>

```

Figura 4.18: Arquivo de configuração do ITSUMO

Este arquivo contém as informações inseridas na tela principal do ITSUMO, como o arquivo de malha viária a ser utilizado, os passos da simulação, a desaceleração padrão, a velocidade do veículo, o tamanho das células, configurações de sensores para resultados e visualização da simulação. Dentro do parâmetro “odmatrix” (FIGURA 4.19), foi inserido o arquivo que continha as informações sobre a demanda de tráfego (origem e destino de cada veículo). O parâmetro “odsize” definiu a quantidade de veículos que percorreram a malha viária.

```

- <od_drivers>
- <od_driver>
  <od_driver_id> 1 </od_driver_id>
  <source> 57 </source>
  <sink> 65 </sink>
</od_driver>
- <od_driver>
  <od_driver_id> 2 </od_driver_id>
  <source> 50 </source>
  <sink> 66 </sink>
</od_driver>
- <od_driver>
  <od_driver_id> 3 </od_driver_id>
  <source> 25 </source>
  <sink> 66 </sink>
</od_driver>

```

Figura 4.19: Arquivo com definições de origem e destino para cada veículo

Após este processo, o *software* iniciou a tentativa de injetar o máximo número de veículos na simulação. O cálculo de menor caminho foi realizado ao iniciar a simulação. Não houve possibilidade de definir tipos de veículos ou intervalos de tempo para esta demanda, tornando o *software* limitado neste quesito.



#### 4.1.2.4 Relatórios

O ITSUMO possui um relatório chamado *lane\_sensor*. Através deste foi possível obter a densidade e a velocidade das faixas observadas durante a simulação. O resultado foi informado para cada passo do experimento, como ilustrado na Figura 4.20. Para processar esta quantidade enorme de informação que foi fornecida, foi necessário utilizar um banco de dados ou uma planilha eletrônica para agrupar estes valores.

# Step	Lane	Density	Average speed
0	644	0,02	3
0	645	0,02	2
0	648	0,01	1
0	647	0,1	1,2
5	644	0,02	3
5	645	0.05	3.00
5	648	0.30	1.62
5	647	0.40	2.44

Figura 4.20: Exemplo de resultados obtidos com o *lane\_sensor*

Agrupado estes valores para as vias desejadas e obtidas as médias, o valor da velocidade foi alterado para m/s, para posterior comparação.

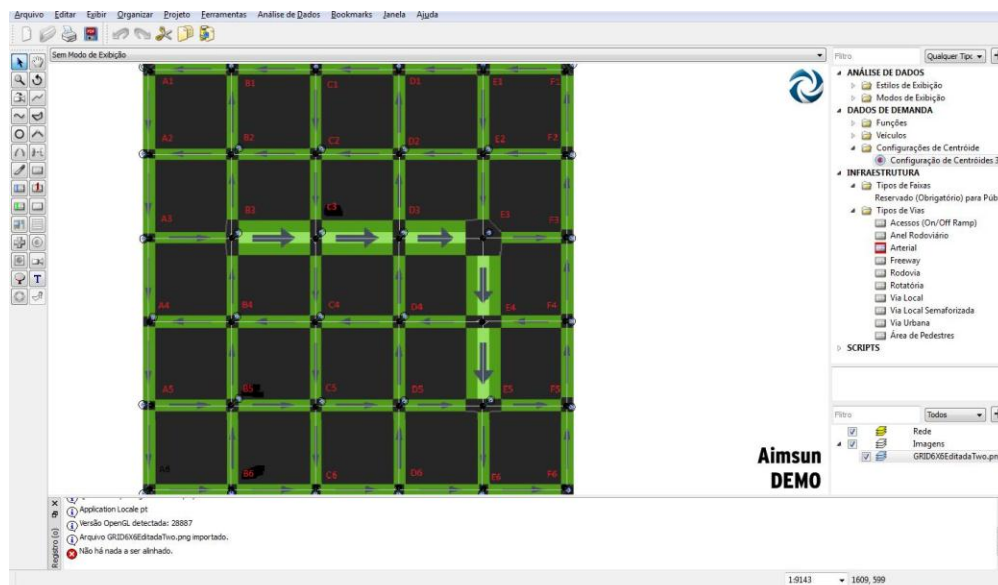
Como este modelo está baseado no autômato celular de Nagel-Schreckenberg, as vias foram divididas em células de tamanho fixo de 5 unidades e velocidade máxima de 3 unidades. Assim, se um veículo anda na velocidade máxima de três células a cada passo da simulação (considerando que cada passo equivale a 1 segundo), a velocidade final(v) corresponde a 54 km/h. Com esta lógica foi obtido os demais valores em km/h. Conhecendo o tamanho(d) de cada via (300m), foi possível obter o tempo(t) gasto para o veículo atravessar a via com a fórmula  $t=v/d$ .

### 4.1.3 AIMSUN

#### 4.1.3.1 Malha viária

A malha viária obtida no *software* ITSUMO serviu como pano de fundo da aplicação. A escala utilizada na imagem reflete 1500m<sup>2</sup> da área total. O resultado é mostrado na FIGURA 4.21

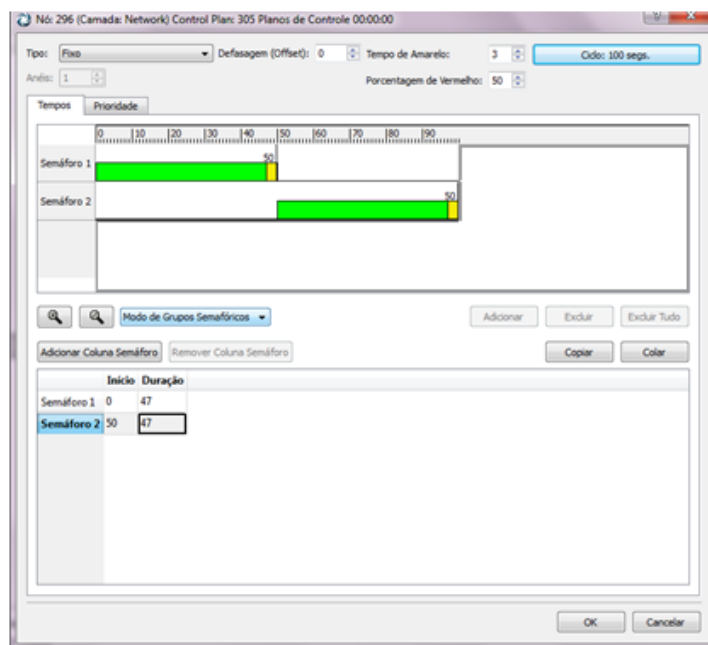
Com a ferramenta *Section tool* as vias foram criadas sobre a figura importada do ITSUMO, seguindo o sentido marcado nas flechas. Por último, a ferramenta *Node Tool* criou as intersecções.



**Figura 4.21: Malha viária criada no AIMSUN a partir de uma imagem importada do ITSUMO.**

#### 4.1.3.2 Controladores de tráfego

No estudo de caso, foram definidos os semáforos para as intersecções. Todos os nodos com semáforos foram atribuídos a um plano de controle, configurado de acordo com a FIGURA 4.22



**Figura 4.22: Plano de controle utilizado no AIMSUN**

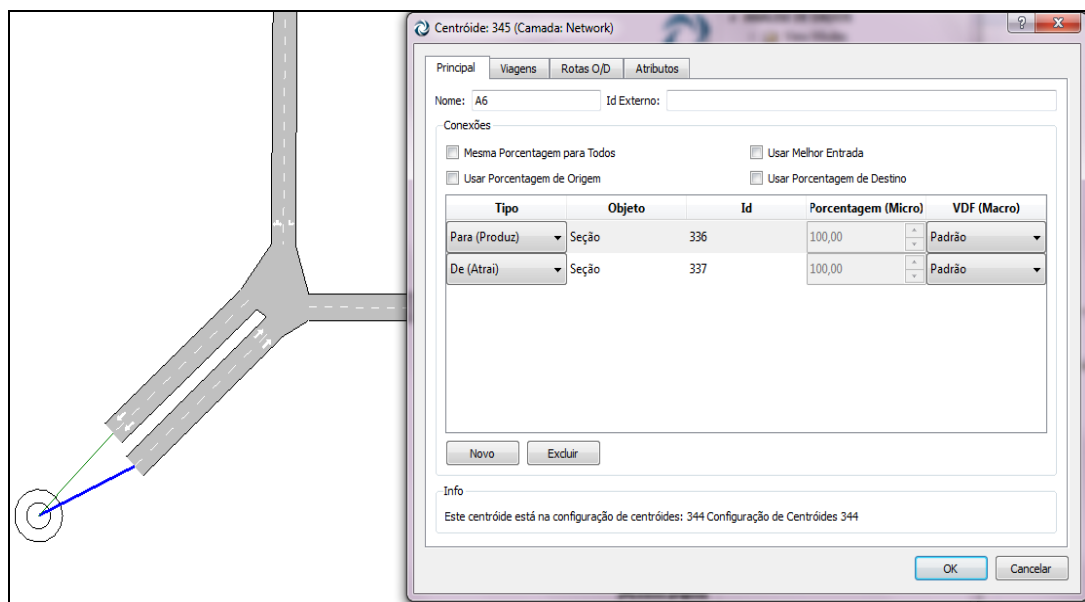
#### 4.1.3.3 Atribuição de demanda

O AIMSUN possui uma interface gráfica para esta tarefa. Nesta interface, foram definidos os pontos de entrada e saída dos veículos, através dos chamados centróides, que apontam para uma via ou intersecção da rede.

Podem existir um ou mais tipos de configurações para os centróides, pois o sistema suporta diversas matrizes OD. No estudo de caso, todos os centróides inseriam e removiam veículos nas intersecções.

A matriz OD ficou pronta depois de preenchidos os campos na interface. O *software* acrescentou linhas e colunas agrupadas, que facilitaram o cálculo da quantidade de veículos que foram simulados. Faltou a opção para organizar a ordem dos centróides, não sendo possível colocá-los em ordem crescente.

Em alguns momentos ocorreram falhas durante a execução da simulação com matriz OD, nestes momentos, foram necessários incluir uma via para servir de entrada e saída de veículos, ao invés de utilizar as intersecções (centróides). A FIGURA 4.23 ilustra como foi corrigido o problema.



**Figura 4.23: Correção de um problema com centróides. Inserção de uma via para entrada e saída de veículos da simulação**

#### 4.1.3.4 Relatórios

Após a conclusão da simulação, foi encontrada a via desejada na malha viária, verificado suas propriedades e acessado a aba de Séries Temporais. Dentro desta aba encontram-se todos os cálculos efetuados com as propriedades das vias (FIGURA 4.24)

	Densidade SRC car (veic/km)	Tempo de Viagem SRC car (segundos/km)
08:10:00	13,05 (0,00)	96,14 (60,80)
08:20:00	15,38 (0,00)	104,92 (64,72)
08:30:00	17,08 (0,00)	110,02 (58,58)
08:40:00	16,92 (0,00)	114,55 (75,02)
08:50:00	15,72 (0,00)	109,44 (61,72)
09:00:00	17,16 (0,00)	110,98 (65,06)
Valor global	15,88	107,97
Média	15,88	107,67

**Figura 4.24: Tabela com valores coletado**

#### 4.1.4 VISSIM

##### 4.1.4.1 Malha Viária

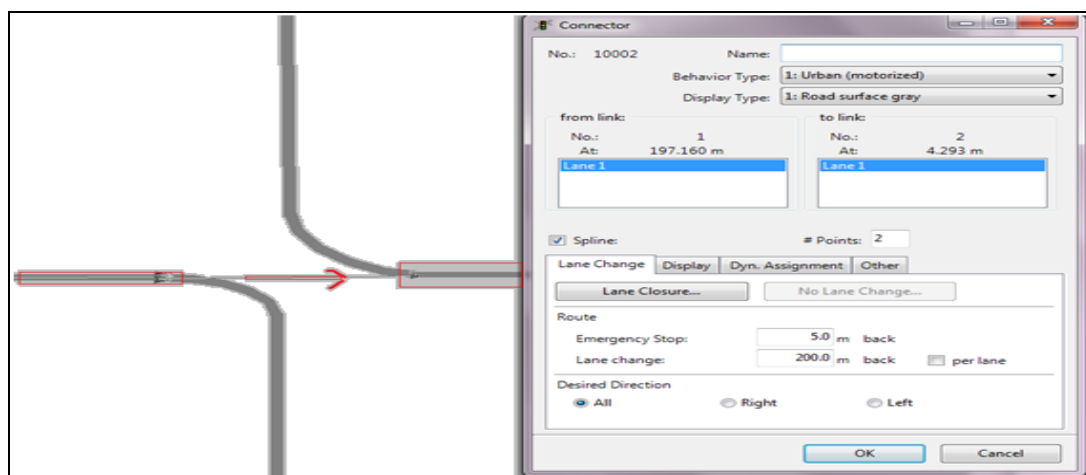
Após importar a figura do software ITSUMO e alterar a escala, as vias foram posicionadas sobre a imagem como descrito a seguir:

Selecionou-se o botão arestas e nodos e, assim, o *software* habilitou o desenho das vias e dos conectores. Pressionado e arrastando o botão direito do mouse sobre uma área vazia, foi construído uma via. Para conectar duas vias, selecionou-se a origem e o destino, com isto, o *software* as uniu. A cada item criado, uma janela se abriu permitindo especificar as propriedades deste item. As vias foram nomeadas de acordo com o código de letras proposto anteriormente.

No VISSIM, os nodos não unem as vias, como nos demais *softwares*, estas conexões devem ser criadas utilizando conectores (neste *software*, os nodos auxiliam na definição das rotas). Cada intersecção precisa do número exato de conectores para os movimentos habilitados no cruzamento. A FIGURA 4.25 ilustra um conector sendo criado.

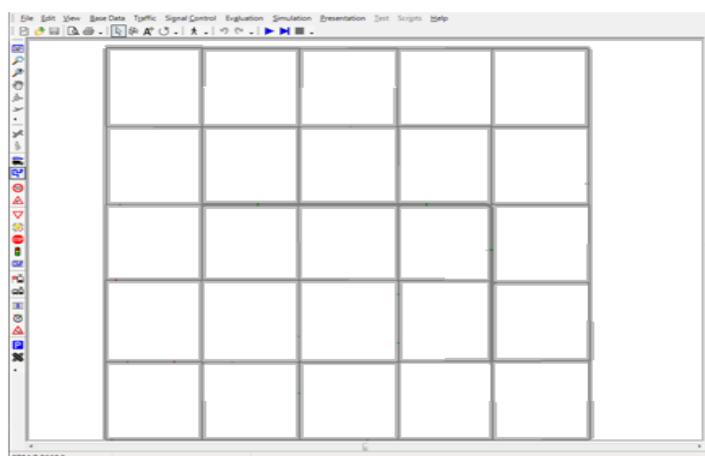
As curvas deste cruzamento foram elaboradas utilizando o parâmetro *Spline*, que permitiu definir pontos no desenho deste cruzamento, habilitando o movimento

curvilíneo. O *software* calculou automaticamente este processo. Neste caso foram utilizados seis pontos para definir uma curva.



**Figura 4.25: Intersecção sendo criada no VISSIM**

O resultado final da configuração da malha viária pode ser observado na FIGURA 4.26 a seguir.



**Figura 4.26: Malha viária construída no VISSIM**

#### 4.1.4.2 Controladores de Tráfego

O controle dos semáforos somente é possível através da adição de *softwares* de terceiros, alguns destes, disponibilizados junto com a distribuição. Entre eles, o VISSIM é um controlador com tempo de ciclo fixo.

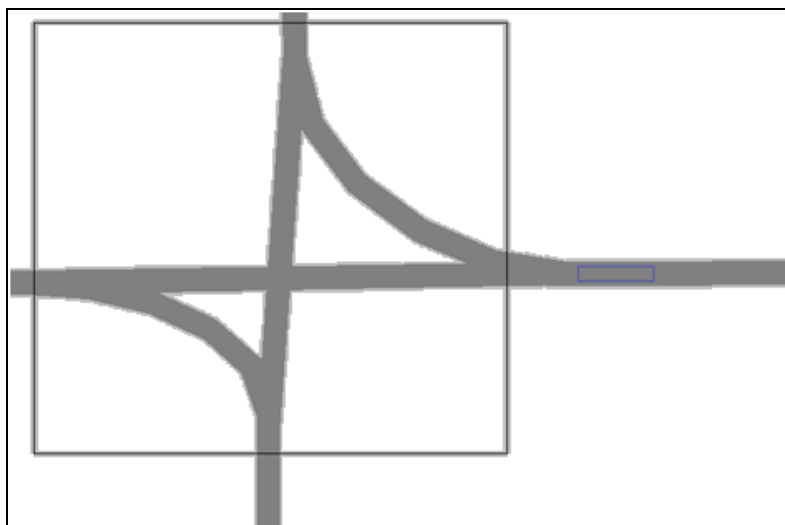
A distribuição do *software* usada neste estudo, não permite salvar o projeto e para inserir um controlador de tráfego é necessário salvar o projeto antes de executar esta ação. Portanto não foi possível executar este passo com este simulador.

#### 4.1.4.3 Atribuição de demanda

Para definir rotas manualmente usou-se a ferramenta *routes*. Porém, para definir a matriz OD, requisito do estudo de caso, uma série de passos foram seguidos.

Primeiro foram definidos os nodos da malha viária de cada intersecção, conforme a FIGURA 4.27. Nesta mesma figura é possível visualizar os estacionamentos inseridos ao lado destes nodos, representando a zona em questão. Estes estacionamentos receberam o mesmo nome da zona que é mencionado no arquivo de matriz OD. O cruzamento da figura foi ampliado para facilitar a visualização.

Nas bordas da topologia, os cruzamentos não possuem a mesma configuração, porém basta criar um nodo semelhante à figura e adicionar um estacionamento na via vertical.

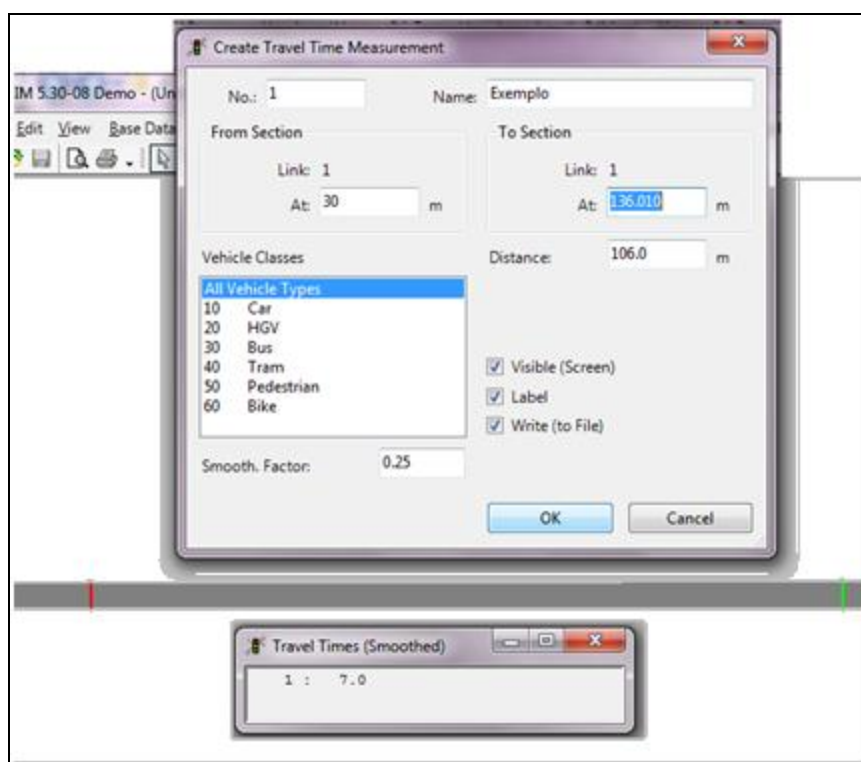


**Figura 4.27** Definição de nodos e estacionamentos na grade 6x6. O nodo está representado pelo quadrado maior e o estacionamento, pelo retângulo sobre a pista

#### 4.1.4.4 Relatórios

A versão de demonstração do VISSIM não liberou o acesso à seção de relatórios do *software*, porém, as janelas que permitem obter informações em tempo real estavam

disponíveis e foram utilizadas. O programa possui uma ferramenta para calcular os tempos de viagem em qualquer via da malha viária. Para inseri-lo, foi selecionado a ferramenta *Travel Time* e selecionado os pontos de início (linha vermelha) e fim (linha verde) da via observada, como na Figura 4.288.



**Figura 4.28:** Configuração de tempo de viagem. A ferramenta *Travel Time* está em destaque

O tamanho da via analisada foi de 300 metros. O parâmetro *smoothed*, definido em 0.25, corresponde ao peso que um novo dado de coleta afeta a média.

#### 4.1.5 Análise dos resultados

Para realizar a simulação na Grade 6x6, os nodos da malha viária foram considerados como pontos injetores e coletores de veículos. Os testes foram realizados com dois, três e quatro mil veículos em 1800 segundos de tempo de simulação, em duplicata, avaliando a performance dos softwares na presença ou ausência de semáforos. O *software* VISSIM foi avaliado somente na ausência de semáforos, devido à limitação da licença.

Cada *software* fornece um conjunto complexo de resultados, assim, encontrar um resultado semelhante, que possa ser usado para a comparação entre os *softwares*, é uma



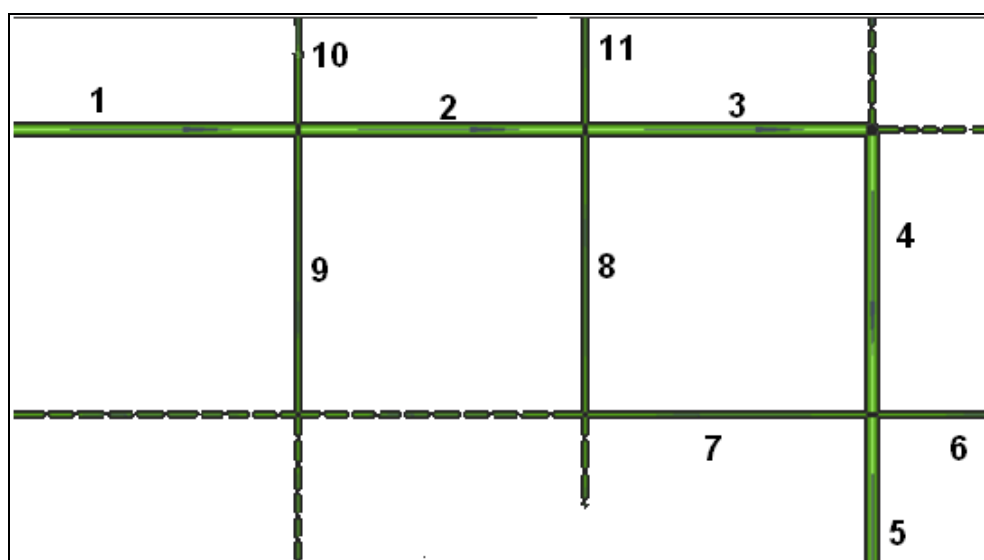
tarefa complexa. Não há padronização entre os programas e, principalmente, as unidades de medidas adotadas diferem entre si.

A medida para avaliar a performance dos *softwares* foi o tempo médio de viagem, dados em segundos, para determinadas vias. Com este dado, foi possível obter conclusões sobre como os congestionamentos ocorreram, o desempenho dos semáforos e a velocidade média obtida pelos veículos.

Analisar todas as vias da malha viária exigiu um esforço manual, pois, em alguns softwares, esta informação não foi fornecida. Foram escolhidas onze vias centrais da malha viária. As cinco primeiras avaliadas eram formadas pela via arterial, composta de três faixas, e as demais, eram vias que cruzavam a via arterial, como indicado na TABELA 4.1 e a FIGURA 4.29.

**Tabela 4.1: Vias selecionadas para análise do tempo médio de viagem**

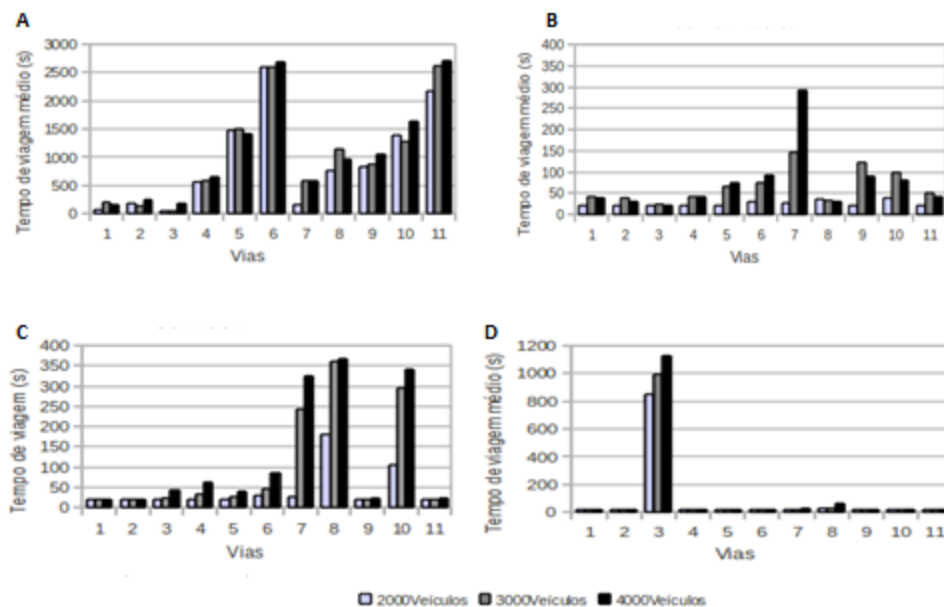
Identificador	Nodo Origem	Nodo Destino
1	B3	C3
2	C3	D3
3	D3	E3
4	E3	E4
5	E4	E5
6	F4	E4
7	E4	D4
8	D4	D3
9	C3	C4
10	C2	C3
11	D3	D2



**Figura 4.29: Vias selecionadas para análise do tempo médio de viagem**

Os resultados obtidos na simulação da grade 6X6, na ausência de semáforos, estão representados nos gráficos da FIGURA 4.30. O eixo y dos gráficos representa o tempo gasto em cada simulação (em segundos), de acordo com o número de veículos circulantes e no eixo x estão as vias que foram avaliadas.

De acordo com os gráficos, o tempo médio de viagem, em geral, não foi alterado com o aumento na quantidade de veículos circulantes, resultado que pode ser explicado pela capacidade da malha viária. Aumentos acima da capacidade máxima da via, provavelmente aumentariam este tempo médio de viagem.



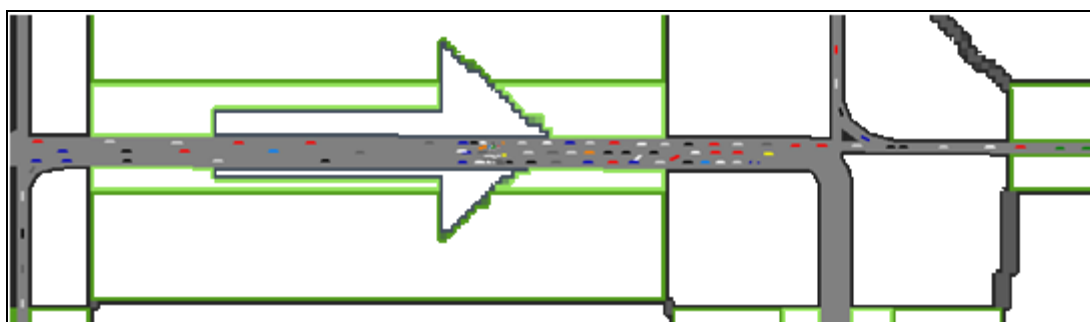
**Figura 4.30: Tempos de viagem: resultados obtidos nos softwares na ausência de semáforos. A. SUMO, B. ITSUMO, C. AIMSUN e D. VISSIM.**

Pela análise dos resultados obtidos no SUMO, observa-se que as 3 primeiras vias seguem um padrão de fluxo constante, independente do aumento no número de veículos circulantes. Da via 4 até a via 6, há um aumento progressivo no tempo de viagem médio, chegando o tempo a ser 6 vezes maior do que nas faixas iniciais. O fluxo torna-se novamente constante na via 7, porém, nas vias de número 8 até 11, que dão acesso direto as vias iniciais, o tempo médio de viagem volta a aumentar. Estes aumentos estão relacionados, provavelmente, aos congestionamentos que se formaram nas intersecções destas vias, pois, o *software* não possui um método eficiente de definir a preferencial em intersecções. Somado a isso, nos congestionamentos que impossibilitaram a saída do veículo, o *software* transportou o veículo para próximo do destino, porém, esta operação foi demorada, atrasando a conclusão da simulação. No ITSUMO, os tempos de viagem médios, em cada via, em geral, mantiveram-se constantes. Um pico no aumento do tempo é observado somente na via 7, provavelmente devido a preferência dos veículos por esta via para chegar ao destino, aumentando o tempo em 6 vezes para o maior número de veículos e 3 vezes para 3000 veículos. Este aumento, porém, está muito abaixo do tempo de velocidade média obtido no SUMO. Podemos inferir assim, que no quesito avaliado, os veículos levam menos tempo para atravessar as intersecções no ITSUMO em relação ao SUMO. Porém, devido às vias do ITSUMO serem divididas em

células, os veículos, ao realizar a conversão, podem ter sido favorecidos em relação ao *software* SUMO.

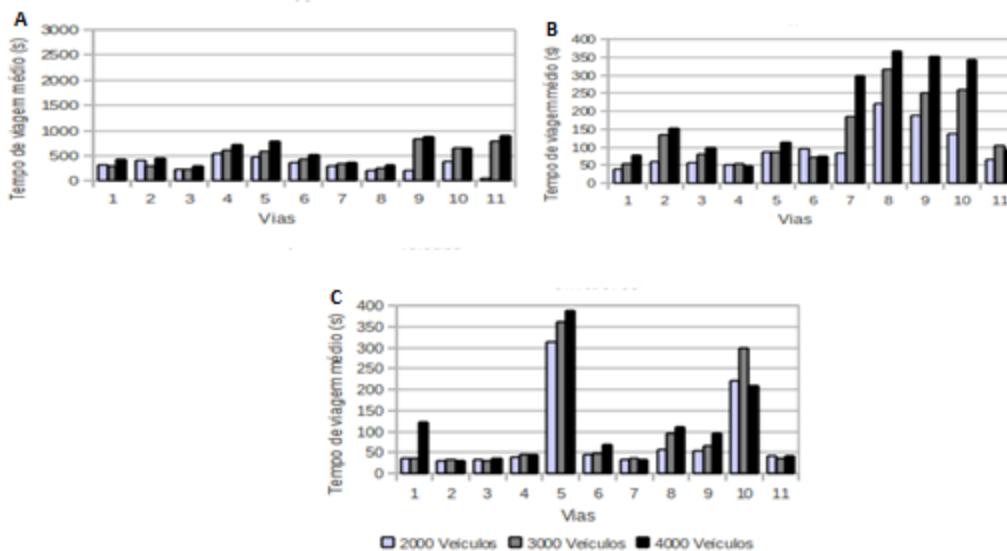
No AIMSUN, os principais congestionamentos ficaram localizados nas vias de acesso à via principal. O aumento de tempo nestas vias foi semelhante aos aumentos observados no *software* ITSUMO. O AIMSUN chega, até mesmo, a apresentar maior número de vias congestionadas em relação ao ITSUMO.

O VISSIM apresentou somente congestionamento na via 3, uma via principal. Nas demais vias apresentou comportamento semelhante e rápido. Na via 3, no decorrer da execução, alguns veículos ficaram sobre os outros na via, aumentando consideravelmente o tempo médio de viagem (FIGURA 4.31). Não foi encontrada uma maneira de impedir este comportamento, pois a configuração preconizava, para o motorista, manter uma distância de no mínimo dois metros do veículo à frente. Os demais intervalos de tempo variaram em torno de 20 segundos.



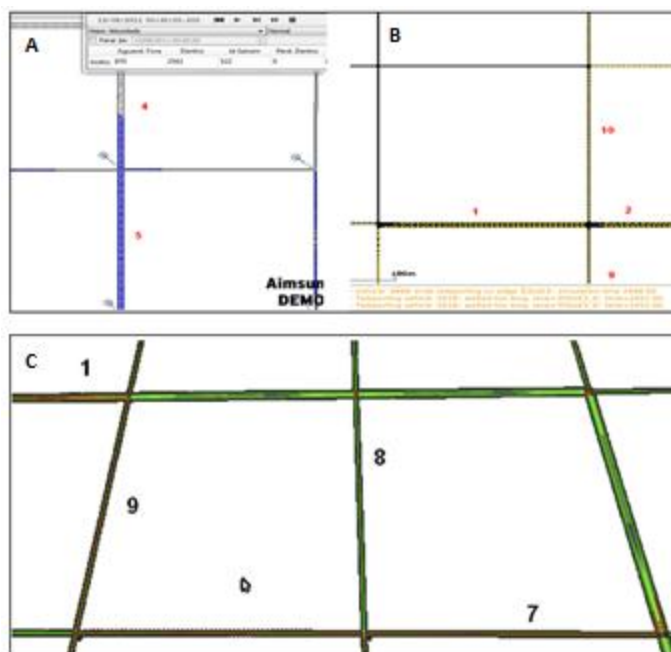
**Figura 4.31 Veículos sobrepostos no VISSIM**

Após este experimento, a mesma simulação foi novamente realizada, porém agora, existem semáforos nas intersecções. Estes semáforos possuem intervalos de fases igualmente distribuídos. Os resultados obtidos estão ilustrados na FIGURA 4.32.



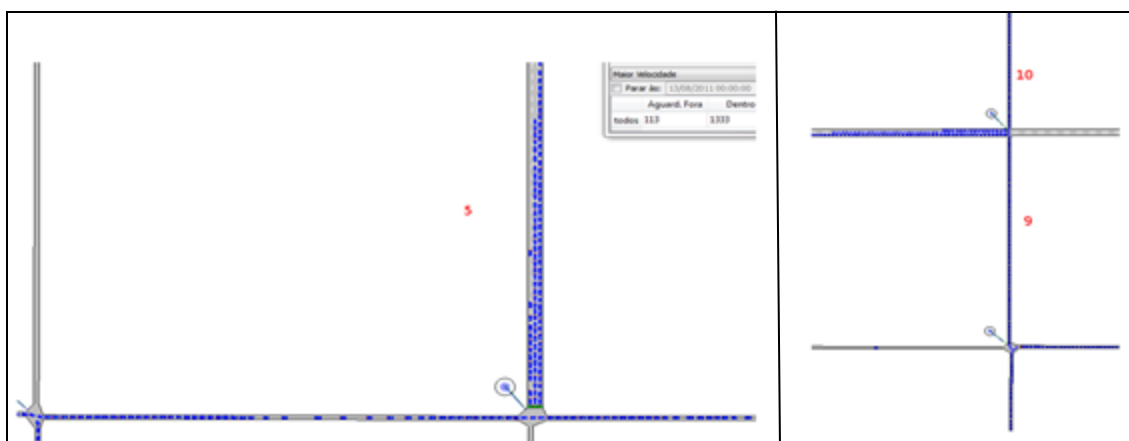
**Figura 4.32: Resultados dos simuladores com semáforos. A. SUMO, B. ITSUMO e C. AIMSUN**

Com a inserção de semáforos, o tempo médio de viagem no *software* SUMO diminuiu consideravelmente (pico de tempo de 3000s para 1000s). No ITSUMO, os semáforos alteraram os tempos de diversas vias, principalmente aumentando este tempo (FIGURA 4.32). Para o *software* AIMSUN, a inserção dos semáforos favoreceu algumas vias (7, 8 e 10), porém, aumentou o tempo pra outras (5). Com 4000 veículos, nenhum dos simuladores finalizou a simulação com todos os veículos chegando ao destino (FIGURA 4.33: informações do simulador AIMSUN, SUMO e ITSUMO).



**Figura 4.33: Veículos que não cumpriram a simulação e vias que mais congestionaram. A. via 5 congestionada no AIMSUN, B. via 10 congestionada no SUMO e em C. Vias congestionadas no ITSUMO**

O tempo de viagem aumentou entre a quarta e o final da quinta via, para todos os *softwares*, isto ocorreu devido ao congestionamento que se formou quando os veículos que percorrem três faixas devem trocar apenas para uma. Na nona e décima vias, o semáforo não consegue disponibilizar tempo suficiente para ambos os lados. A FIGURA 4.34, ilustra estas duas situações respectivamente.

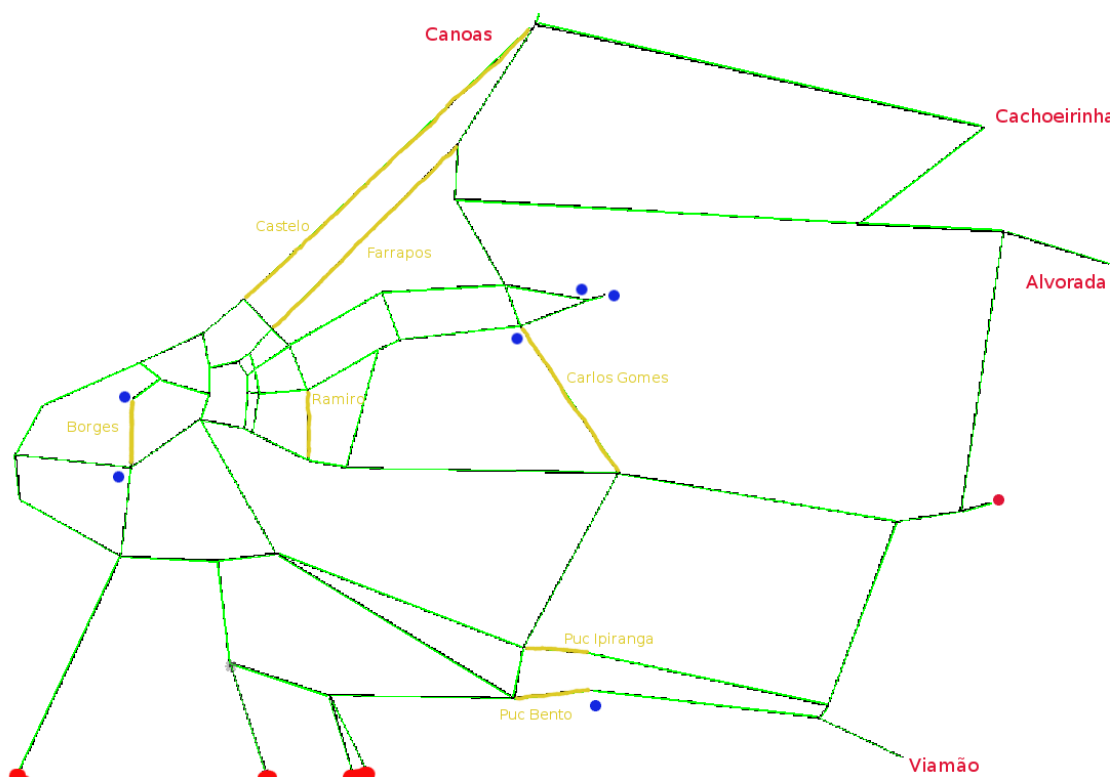


**Figura 4.34: Congestionamento na via 5 a esquerda. Congestionamento nas vias 9 e 10 a direita**

## 4.2 Porto Alegre, vias principais

O segundo estudo de caso a ser analisado constitui uma generalização de malha viária verdadeira. A cidade de Porto Alegre está situada nas coordenadas  $30^{\circ} 05'S$ ,  $51^{\circ} 10'W$ , pertencente ao estado do Rio Grande do Sul, Brasil. Possui aproximadamente  $500 \text{ km}^2$  de área e diversas ruas e avenidas.

Suas cidades limítrofes são Canoas, Cachoeirinha, Viamão e Alvorada. Neste estudo de caso foram construídas somente as vias principais da cidade de Porto Alegre e as vias que fazem conexão entre estes municípios. A FIGURA 4.35 ilustra o formato final da topologia a ser analisada.



**Figura 4.35: Topologia desenvolvida para a cidade Porto Alegre através do software ITSUMO**

As principais características para desenvolver uma simulação foram abordadas na análise da topologia em formato  $6 \times 6$ . Portanto nesta seção somente foi apresentado os parâmetros que envolvem a obtenção dos resultados.

### 4.2.1 Malha viária

Nos simuladores de licença livre foi necessário adaptar a topologia importando o modelo do site *OpenStreetMaps* (OSM), para a cidade de Porto Alegre, deixando apenas as vias principais. Através das coordenadas fornecida no arquivo OSM foi possível exportar estes valores para o ITSUMO. Devido a falta de detalhe para modelar curvas na malha viária, o resultado se afasta da realidade. Com os pontos cartesianos da malha viária do ITSUMO é possível construir um arquivo XML semelhante para o SUMO (FIGURA 4.35).

Nos *softwares* pagos, foi utilizada a ferramenta para colocar os mapas conforme a escala de uma figura. A figura utilizada foi retirada do SUMO, pois contém a indicação da escala, FIGURA 4.36. Após inserir a escala corretamente, devem ser inseridas as vias por cima da figura, formando a malha viária.

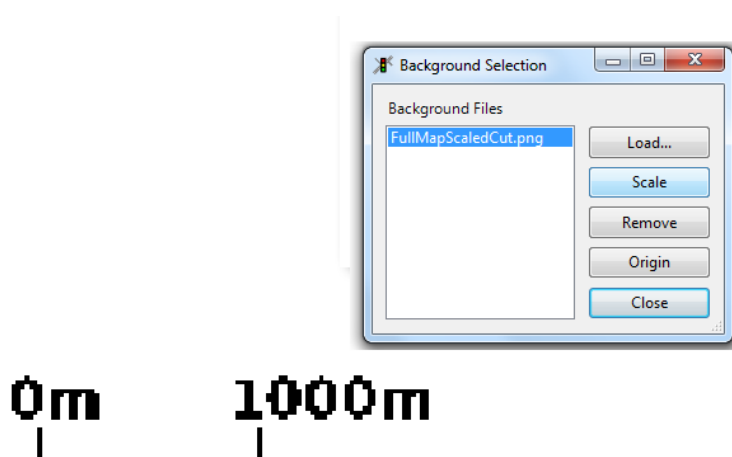


Figura 4.36: Escala utilizada no VISSIM

Vias aleatórias foram escolhidas para fazer parte deste estudo de caso. As vias em amarelo na FIGURA 4.35 representam os nomes e as vias que serão analisadas nesta seção. Todas possuem três faixas, exceto a via Ramiro, que possui duas e tem sentido único.

Após a escolha das vias principais a ser analisada na simulação, algumas dificuldades foram encontradas para modelar cruzamentos complexos existentes nas cidades. Viadutos com cinco metros de altura construídos para facilitar o trânsito nas cidades, não podem ser modelados nos programas com código aberto. Para não



prejudicar os resultados da simulação, os softwares privados também foram modelados sem a construção de viadutos.

#### **4.2.2 Atribuição de tráfego**

Para definir a demanda de tráfego nas simulações, também foram utilizadas matrizes OD. Os pontos escolhidos onde devem ser injetados e removidos os veículos, estão principalmente na periferia da malha viária. São pontos na região sul do mapa e as cidades fronteiriças em vermelho. Os pontos em azul no mapa representam os demais pontos de origem e destino. Estes pontos são responsáveis por 6,7% das viagens que se originam em cada nodo. Foram utilizadas matrizes com a seguinte quantidade de veículos: 20 mil, 40 mil e 60 mil. A malha viária é composta de 151 vias de aproximadamente 76 mil metros com 3 faixas em cada direção. Supondo o tamanho médio de um veículo popular, foi estimado que a malha viária suporta até 100 mil veículos. Portanto os valores utilizados são para avaliar uma situação de trânsito congestionado. (BAZZAN, A. et al, 2011)

#### **4.2.3 Controladores de tráfego**

Os controladores de tráfego são essenciais em cidades grandes, pois sem eles o trânsito fica congestionado de tal maneira que os veículos não podem se mover. Encontrada esta situação, cada software lida com ela de uma maneira. O SUMO transporta os veículos para a região mais próxima do destino dos veículos onde não existe trânsito. O ITSUMO finaliza a simulação porque foi encontrada uma situação estática. O AIMSUN e o VISSIM mantêm as filas como estão e deixa a cargo do usuário visualizar este tipo de erro. Este comportamento no VISSIM deve ser bem modelado, senão os veículos se atravessam e não ocorre colisão.

Foram escolhidos intersecções para inserir os semáforos na malha viária, para aqueles cruzamentos que possuem mais de duas vias. Cada semáforo possui 60 segundos de tempo de ciclo para distribuir entre as vias. Não foi possível definir semáforos para o VISSIM, devido à licença disponibilizada para o estudo.

#### 4.2.4 Análise dos resultados

As simulações executadas na topologia de Porto Alegre resultaram em situações semelhantes às encontradas nos testes com a malha viária Grade 6x6. Em suma, quando não há semáforo, ocorrem congestionamentos que impedem o fluxo totalmente. Quando os semáforos são inseridos, o trânsito entre as vias é efetuado normalmente nos primeiros momentos da execução, porém com o passar do tempo, os semáforos não conseguem lidar com a elevada carga de veículos e os congestionamentos aumentam.

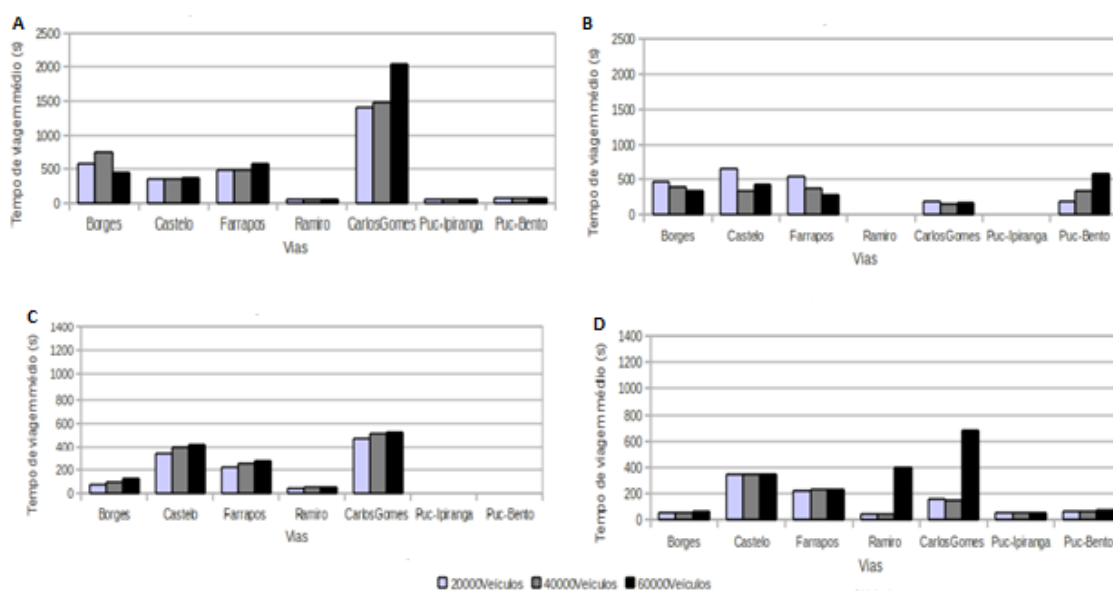


Figura 4.347: Tempos de viagem: resultados obtidos nos *softwares* na ausência de semáforos nas principais vias de Porto Alegre. A. SUMO, B. ITSUMO, C. AIMSUN e D. VISSIM.

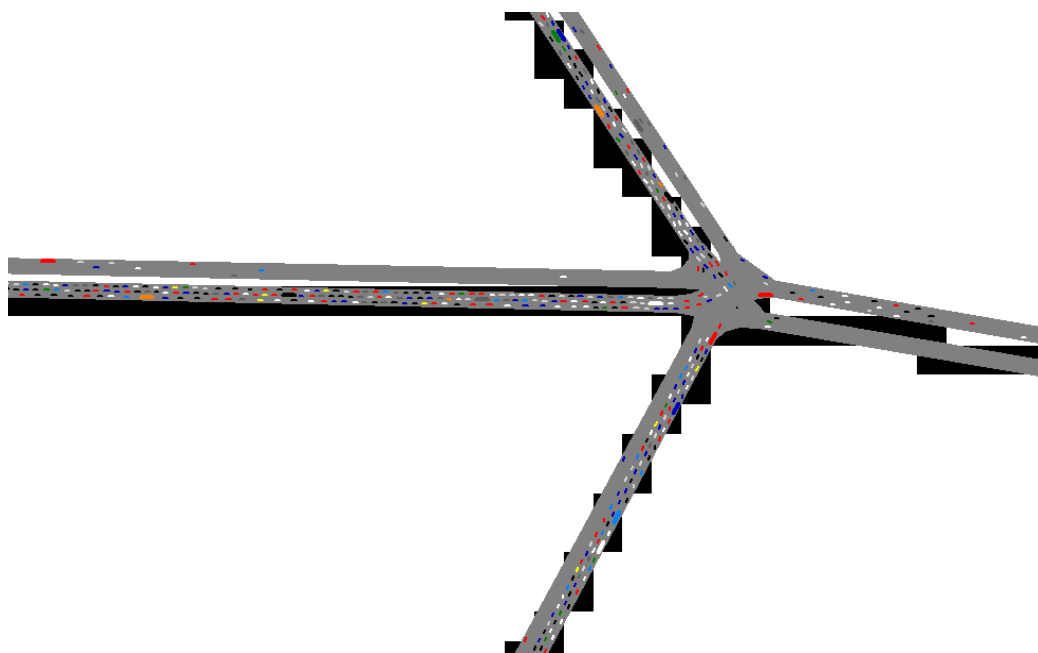
Analisando as quatro primeiras vias dos *softwares* é possível verificar que os resultados estão próximos. São vias que não apresentaram comportamentos anômalos, pois não ocorreram congestionamentos nas proximidades e os fluxos de veículos percorreram as vias sem atingir a capacidade. No ITSUMO e no SUMO, a Borges apresentou um aumento no tempo de viagem devido à proximidade com um ponto de origem e destino.

O VISSIM apresentou em média resultados muito semelhantes, mesmo com a inserção de mais veículos na demanda. Na última simulação com 60000 veículos a intersecção da Carlos Gomes com a Puc-Ipiranga causou grandes congestionamentos. Neste caso, os veículos não se sobrepuseram uns sobre os outros, como na grade 6X6, sendo assim possível verificar o comportamento dos motoristas.

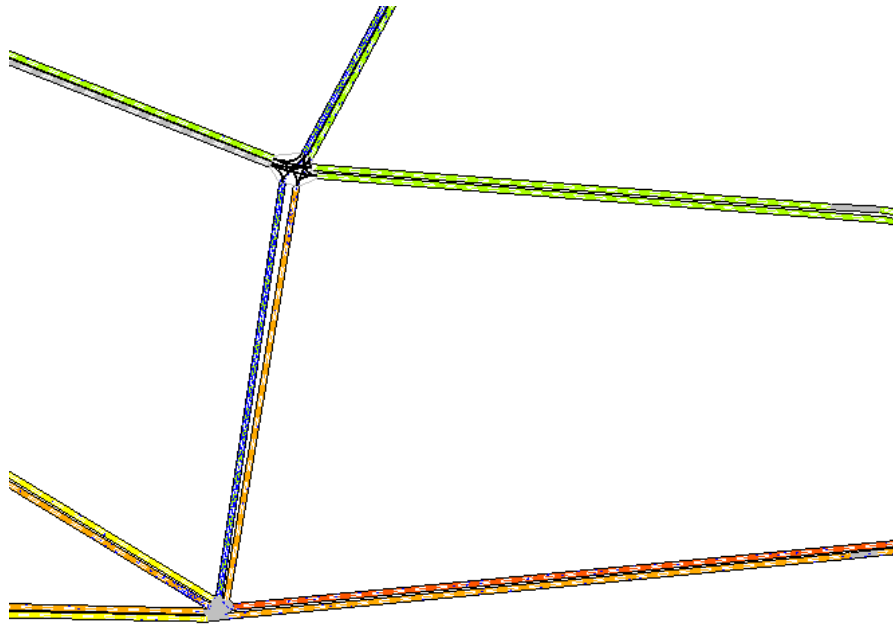
O SUMO se comportou semelhante ao VISSIM quando não havia congestionamento, como na via Ramiro e Puc-Ipiranga. Nos outros casos, os congestionamentos foram elevados, principalmente a CarlosGomes o que acarretou um acréscimo no tempo de viagem.

O AIMSUN ignorou a via Puc-Ipiranga nesta primeira etapa dos testes. Os tempos de viagem foram semelhantes comparados ao VISSIM, exceto em casos de congestionamentos como na via Carlos Gomes. A FIGURA 4.38 apresenta uma situação de congestionamento no software VISSIM.. Devido ao imenso congestionamento formado na via Puc-Bento o *software* AIMSUN não computou os valores de tempo de viagem médio (FIGURA 4.39).

O ITSUMO apresentou um comportamento diferente do esperado, pois os tempos de viagem diminuíram com a entrada de mais veículos na matriz OD.

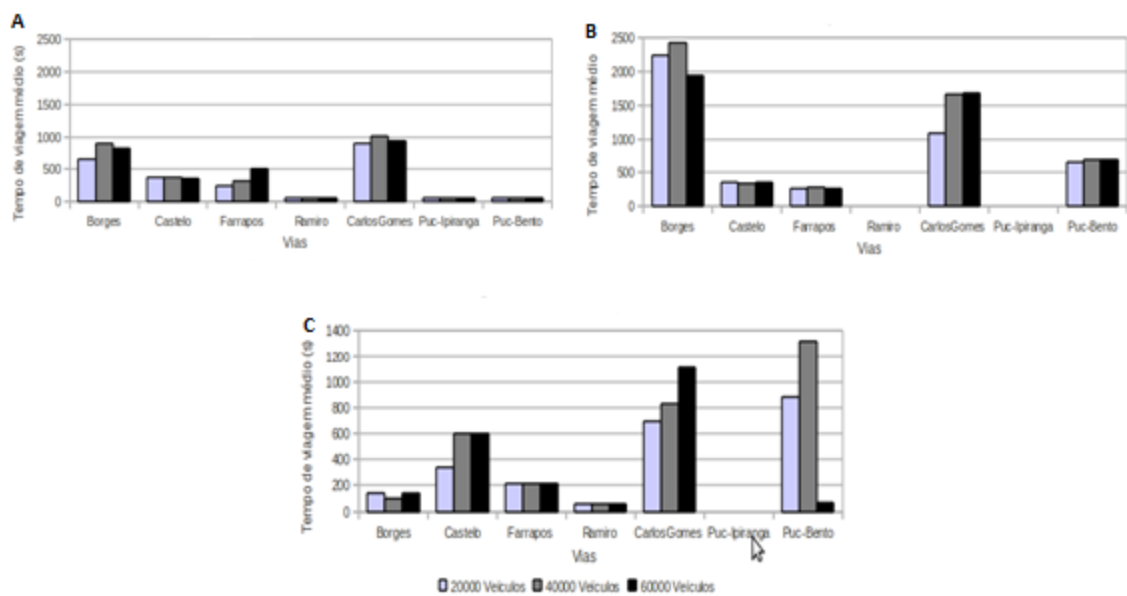


**Figura 4.38: Via Carlos Gomes lotada no VISSIM**



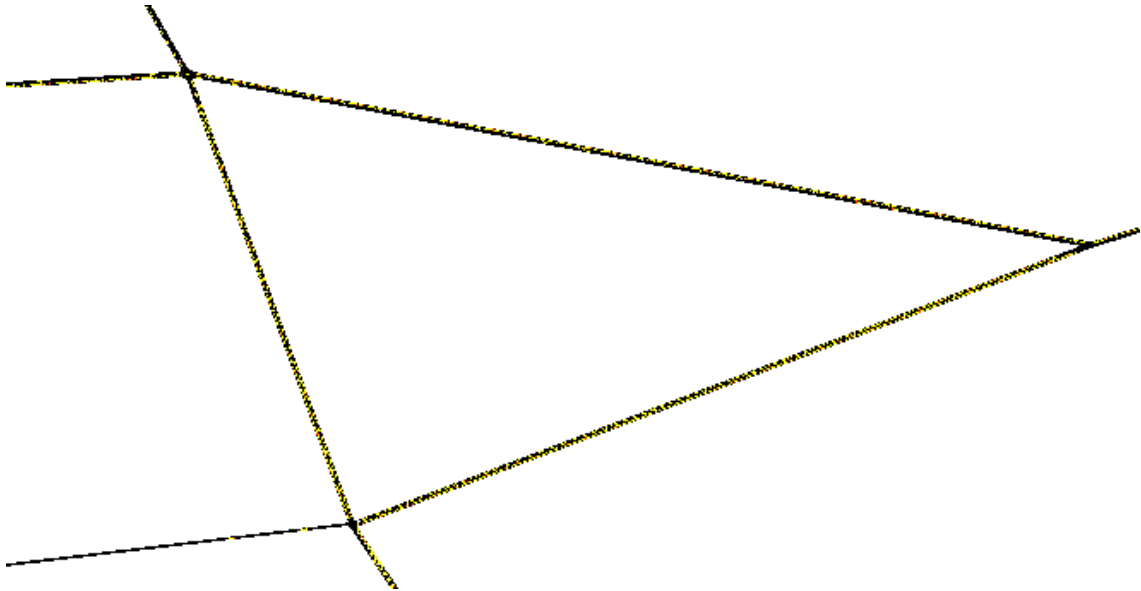
**Figura 4.39: Vias coloridas no AIMSUN conforme densidade**

A FIGURA 4.40 representa os gráficos gerados com os tempos de viagem das vias previamente mencionadas, porém agora com semáforos. Estes gráficos apresentaram semelhanças entre si para todos os simuladores analisados, como o tempo de viagem na via Castelo em torno de 500s e na via Farrapos, em torno de 300s para percorrer-la.



**Figura 4.350: Resultados da simulação na malha viária de Porto Alegre na presença de semáforos. A. SUMO, B. ITSUMO e C. AIMSUN.**

Todos os softwares apresentaram grandes congestionamentos na Carlos Gomes, devido aos pontos de origem e destino nas proximidades, FIGURA 4.41. Nesta via o ITSUMO apresentou o menor tempo de viagem encontrado, o motivo pode estar relacionado à maneira como os veículos aceleram após o semáforo liberar o fluxo, conseguindo liberar mais veículos que os outros simuladores.



**Figura 4.361: Alta densidade de veículos na metade da via Carlos Gomes no SUMO**

## 5 COMPARAÇÃO ENTRE OS *SOFTWARES*

Cinco itens foram analisados para a comparação entre os *softwares*:

- a) Procedimentos de instalação do programa;
- b) Elaboração da malha viária;
- c) Atribuição de tráfego;
- d) Características da simulação
- e) Aspectos gerais

Estes cinco itens, cada qual com subitens, formaram a base para a comparação entre os *softwares*. Os dados adquiridos ao longo do estudo de caso, juntamente com dados da literatura, auxiliaram este processo. Ao final da análise de cada classe, os resultados foram compilados em uma tabela e classificados em ótimo, bom, regular ou ruim, de acordo com o desempenho cada software.

### 5.1 Instalação dos softwares

Foram analisados os passos que envolvem o processo de instalação dos softwares, incluindo o sistema operacional, as licenças disponibilizadas por cada empresa responsável, os requisitos mínimos para executar simulações, a documentação fornecida e a integração com bancos de dados. Foram analisadas as dificuldades e facilidades para cumprir cada tarefa.

Os SSTs com licença livre estão vinculados aos pacotes fornecidos pela comunidade. Encontrar estes pacotes de instalação, com a versão correta, pode vir a ser um problema, por isto foi avaliado. Os SST de licença livre analisados, SUMO e ITSUMO, apresentaram algum tipo de problema relacionado a este quesito. A

existência do comando *make* na pasta *src* informa ao usuário se a instalação procedeu corretamente, facilitando este processo. Ambos os *softwares* de licença livre possuem este comando.

Os softwares AIMSUN e VISSIM foram instalados no sistema operacional Windows sem problemas. Seguindo o programa de instalação, passo a passo, a tarefa foi concluída.

### 5.1.1 Sistema Operacional

O SUMO possui disponibilidade em sistemas baseados em Linux, como OpenSuse, Fedora, e Debian. Sistemas baseados em Mac, Solaris e Cygwin, não apresentam interface gráfica, porém compila e executa. Existe também compatibilidade com o sistema operacional Windows, contanto que o *software* seja compilado utilizando o software proprietário, Visual Studio.

Atualmente o ITSUMO roda em Linux, Ubuntu, Fedora e OpenSuse 11.

O AIMSUN possui suporte ao sistema operacional Windows, nas versões XP, Vista e 7. Pode ser executado em versões de 32 e 64 bits. Pode ser utilizado também nos sistemas Mac OS 10.6.6 64-bit (SnowLeopard), e Linux versão Ubuntu 10.04.

O VISSIM está plenamente adaptado ao sistema operacional Windows, nas versões XP, Vista e 7. Pode ser executado em versões de 32 e 64 bits.

### 5.1.2 Licença

O ITSUMO e o SUMO possuem licença de *software* livre sobre os termos de GPL (*General Public License*). O AIMSUN fornece quatro tipos de licença: *Small*, *Standard*, *Professional* e *Advanced*, as vantagens entre estas incluem, ferramentas adicionais de suporte a múltiplas CPUs e acesso a ferramentas de programação. Aos interessados em pesquisar o programa é oferecida uma versão de demonstração. Esta licença disponibiliza acesso a todas as funcionalidades do sistema, mas restringe o usuário de salvar o projeto, armazenar informações no banco de dados e escrever scripts. Após trinta dias, esta licença expira.

O VISSIM possui licença paga e de demonstração. Na versão de demonstração, não é permitido salvar o projeto e a simulação roda apenas 1800 segundos. Os relatórios de

avaliação não estão disponíveis e a interface COM para comunicação com o simulador está desabilitada, restringindo o acesso aos semáforos na simulação, pois é necessária esta comunicação.

### **5.1.3 Requisitos mínimos**

As performances destes simuladores estão associadas aos seguintes fatores: número de veículos contido na malha viária, número e tipo de semáforos controlando as intersecções e o número de núcleos de processadores utilizados. Utilizando configurações idênticas, o computador mais rápido processará uma simulação mais rápida.

O VISSIM recomenda 2GB de memória RAM para aplicações grandes. No manual, uma informação é dada como regra: um veículo ocupa cerca de 2 kB de memória RAM, portanto, se a simulação contém 50000 veículos, estes ocuparão 100 MB de RAM.

O AIMSUN, SUMO e ITSUMO não fornecem informações sobre os requisitos mínimos, apenas informam o sistema operacional necessário.

### **5.1.4 Documentação**

O ITSUMO, apesar de apresentar na internet informações sobre o software, em alguns momentos, esta informação é superficial, caso encontrado na apresentação das matrizes OD ou quando menciona os algoritmos que foram implementados no software. O manual do SUMO também está disponibilizado na internet e, inclusive, a aparência foi reformulada durante a elaboração deste documento. A página do SUMO, apesar de possuir mais conteúdo que o ITSUMO, ainda é limitada. Tanto o AIMSUN quanto o VISSIM apresentam manuais de instalação completos, com diversas informações e exemplos.

### **5.1.5 Banco de dados**

Entre os bancos de dados mais populares está o SLQ Server e Oracle, os quais implementam o modelo de banco de dados relacional, também conhecido como Sistema Gerenciador de Banco de Dados Relacional (SGBDR). A integração com estes bancos de dados é uma característica importante em um simulador de trânsito, pois possibilita realizar consultas complexas sobre as simulações executadas a qualquer momento.



Os *softwares* SUMO e ITSUMO não referenciam a integração de seus softwares com o banco de dados, já os softwares VISSIM e AIMSUN possuem esta integração com banco de dados.

### 5.1.6 Idiomas oferecidos

O SUMO possui a documentação e interface em inglês. Há uma tradução para o idioma persa, mas somente os passos da instalação. O ITSUMO possui a documentação e interface em inglês e uma versão antiga em português. O VISSIM possui setup de instalação em inglês, francês e alemão. Manual em inglês e alemão e interface em inglês. O AIMSUN possui suporte completo ao idioma inglês, português, espanhol e francês, incluindo a adaptação da aplicação ao idioma selecionado.

A tabela a seguir, resume os itens analisados nesta seção:

**Tabela 5.1: Classificação dos *softwares* quanto aos itens de instalação**

Critér. \ Simul.	SUMO	ITSUMO	AIMSUM	VISSIM
Sist. Oper.	Bom	Regular	Ótimo	Regular
Licença	Ótimo	Ótimo	Regular	Regular
Req. Mín.	Ótimo	Ótimo	Ótimo	Ótimo
Docum.	Bom	Regular	Ótimo	Ótimo
Banco de Dados	Ruim	Ruim	Bom	Bom
Idiomas	Regular	Regular	Ótimo	Bom

## 5.2 Malha viária

Este item é fundamental para iniciar um projeto em um *software* simulador de trânsito. São avaliadas as ferramentas disponibilizadas ao usuário para construir o cenário de testes adequados.

### 5.2.1 Importar malhas viárias

Iniciar a construção de uma topologia é um processo lento e complicado. A opção para importar topologias de outros softwares surge como alternativa ao usuário.

No software ITSUMO, somente há a opção de importar malhas viárias desenvolvidas de acordo com o padrão *Open Street Maps*, com algumas restrições. O VISSIM apresenta somente alguns formatos privados, enquanto o AIMSUN apresenta muitas opções de formatos, com a grande maioria na área privada.

### 5.2.2 Interface

A interface é importante para o usuário se familiarizar e se adaptar facilmente com o software. Os *softwares* livres analisados não investiram para obter interfaces elaboradas. O software SUMO exige conhecimento prévio em comandos por linha de código e manipulação em arquivos XML.

O ITSUMO apresenta uma interface na qual o usuário pode inserir erros básicos, pois somente está habilitado o cadastro das informações. Isto exige uma rede pré-definida com a definição de todos os detalhes, pois a malha viária no ITSUMO é construída através de pontos e arestas em um sistema cartesiano de coordenadas, portanto, quanto maior a complexidade, pior o suporte do programa.

O AIMSUN apresenta uma interface completa e moderna. As ferramentas disponíveis ficam no lado esquerdo, a barra de menus na parte superior e informações adicionais são posicionadas no lado direito. Comandos simples pelo mouse possibilitam a construção de topologias rapidamente.

O VISSIM possui interface adequada, mas faltam detalhamentos na apresentação e não possui barras com informações do que foi construído até o momento. Não há distinção entre as cores de uma intersecção e uma via normal. A topologia final se

assemelha mais a um grafo que uma simulação de trânsito. Fica a critério do usuário, definir marcas nas vias para visualização. Existe dificuldade em alinhar as vias desenhadas na topologia e o software não apresenta recursos que facilitem esta tarefa. Por fim, demora um pouco para acostumar-se com a utilização do botão direito do mouse na interação com o software.

### 5.2.3 Ferramentas para interação

As ferramentas disponibilizadas definem a interação do *software* com o usuário, evitando problemas ao desenvolver algum projeto.

O SUMO apresenta seus recursos através de linha de comando, assim, qualquer modificação realizada na topologia deve ser consultada no manual para verificar a sintaxe a ser aplicada e, se for necessário, os arquivos de topologia devem ser alterados manualmente. Portanto, este software está mais para cientistas da computação do que para engenheiros de tráfego. O *software* não possui determinadas funções, como a definição de faixa de pedestres, preferencial de intersecções sem semáforo, suporte às vias com definição de altura, entre outras. São detalhes menores que podem alterar o resultado de uma simulação.

O ITSUMO apresenta um componente para construir uma malha viária que está em um estágio inicial de desenvolvimento, faltando ferramentas para definir curvas, alturas diferentes para os mapas, definição de como se comporta determinada intersecção, inserir detectores para gerar informação e mais uma série de detalhes.

O AIMSUN possui todas as ferramentas básicas disponíveis, porém, encontrar ferramentas complexas exige consulta a documentação, pois o software, por ser muito extenso, possui muitas funcionalidades. Inclusive, o mesmo comando pode estar em mais de um lugar para auxiliar. Há possibilidade de adicionar transporte público, semáforos, detectores, faixa de pedestres e intersecções através da interface facilmente.

No VISSIM, para cada alteração deve ser escolhida a ferramenta adequada, pois o mouse seleciona somente o módulo previamente escolhido. Por exemplo, não é possível alterar os semáforos enquanto a ferramenta de construção de vias está escolhida. Há uma ferramenta que auxilia no momento de escolher itens empilhados um sobre o outro. Em geral, as ferramentas são completas e possuem diversas customizações.

#### 5.2.4 Intersecções não sinalizadas

Para definir como os veículos se comportam em intersecções não sinalizadas, o *software* deve fornecer algum suporte para esta configuração.

No SUMO é possível inserir estas definições através de um arquivo de intersecções com a informação sobre qual via tem a preferência. No ITSUMO existe a opção de marcar qual via tem a preferência, porém não é possível visualizar o movimento dos veículos nas intersecções.

No AIMSUN existe a possibilidade de definir para cada via da intersecção se deve ser cedido a preferência para outra via ou se é obrigatório parar. O VISSIM possui dois modelos para definir esta questão, como definido na seção da malha viária sobre o simulador no capítulo três.

Tanto o modelo do AIMSUN como o do VISSIM quando configurados corretamente apresentaram bom desempenho.

#### 5.2.5 Intersecções sinalizadas

Todos os *softwares* permitem inserir e manipular semáforos na simulação.

Nos testes com o formato 6x6 o simulador SUMO se comportou normalmente com a inserção do comando para inserir semáforos. Na tentativa de explicar como a ferramenta controladora *Traci* funciona, a documentação omitiu algumas informações, pois não há um passo a passo de como ocorre esta interação entre a ferramenta e o simulador.

Nos testes propostos neste trabalho, o *software* ITSUMO comportou-se normalmente e não ocorreram falhas durante a simulação com os semáforos. Um problema encontrado na interface ocorre quando troca-se o tempo de ciclo e esta alteração nem sempre é efetuada. Também parece existir um limite para o tempo de ciclo, mas este não é informado.

O simulador AIMSUN realizou esta tarefa adequadamente, porém não possui um bom tratamento de erros, pois em diversos momentos o *software* travou simulando com semáforos, sendo necessário reiniciá-lo.

Não foi possível efetuar esta tarefa no estudo de caso do VISSIM, porém o problema está relacionado com a licença, apenas de demonstração. A versão inclui demonstrações

de semáforos em funcionamento, na qual foi possível verificar simulações complexas em torno de intersecções.

A tabela a seguir, resume os itens analisados nesta seção:

**Tabela 5.2: Classificação dos *softwares* quanto aos itens da malha viária**

Crítér. \ Simul.	SUMO	ITSUMO	AIMSUN	VISSIM
Importar	Bom	Ruim	Bom	Regular
Interface	Ruim	Regular	Ótimo	Ótimo
Ferramentas	Regular	Regular	Ótimo	Ótimo
Intersecções não sinalizadas	Bom	Regular	Ótimo	Ótimo
Intersecções sinalizadas	Bom	Bom	Bom	Bom

### 5.3 Atribuição de demanda

Definir a demanda de tráfego é fundamental em um *software* simulador de trânsito e precisa ser analisada precisamente pelo engenheiro de trânsito ou responsável antes de iniciar a simulação, evitando falhas na análise sobre o comportamento do trânsito.

#### 5.3.1 Tipos de veículos

Os tipos de veículos que o *software* suporta são essenciais para se aproximar de um comportamento real do trânsito. O SUMO possui este tipo de categorização e permite estender inclusive o modelo que o motorista se comporta durante a simulação. Existe uma grande variedade de classes, encontradas em softwares semelhantes. O ITSUMO não possui referência a este tipo de conceito. Os *softwares* VISSIM e AIMSUN possuem esta categorização e incluem modelos 3D para serem visualizados.

### 5.3.2 Suporte a matriz OD

O SUMO oferece suporte a matriz OD através de uma ferramenta para importar matrizes OD no formato do VISSIM. Permite ao usuário escrever sua própria matriz OD de acordo com o mesmo padrão. O ITSUMO oferece suporte a matriz OD, porém não traz software adicional para lidar com as matrizes.

O AIMSUN foi o melhor *software* entre os estudados para executar este tipo de operação e possui total integração da interface com o simulador.

O VISSIM permite criar a matriz OD através de arquivos de texto conforme o padrão descrito no manual. Depois estes arquivos são importados para o software.

### 5.3.3 Regras de trânsito regionais

Um *software*, para ser completo, deve se adaptar a todas as condições encontradas em diversos países. Facilidade de definir novas regras de trânsito, ou apenas alterar alguns conceitos básicos é importante. O SUMO e o ITSUMO não se adaptam a este tipo de regionalização, enquanto o VISSIM e o AIMSUN possuem este suporte.

### 5.3.4 Interface

Como mencionado anteriormente, somente o AIMSUN possui interface para lidar com matrizes OD, além desta, possui interface para outros formatos de definir demandas. O software ITSUMO possui interface para inserir injetores e coletores de veículos, e a opção para definir caminhos específicos para um motorista.

O VISSIM não apresenta interface para definir a matriz OD, mas possui uma caixa de diálogo, onde é possível definir diversos parâmetros nesta tarefa. O SUMO não possui interface, como previsto no projeto.

### 5.3.5 Tráfego assinalado dinamicamente

Consiste de uma característica importante nos softwares de simulação de trânsito, porém, apenas será avaliado se o software possui ou não esta funcionalidade. No caso do SUMO, é fornecida a ferramenta *DUAITERATE* para computar rotas. No ITSUMO este tipo de demanda deve ser realizada manualmente realizando iterações sobre os arquivos de matriz OD.

No AIMSUN existe a possibilidade de escolher entre microscópica ou macroscópica para definir a demanda, incluindo uma série de opções e manuais em relação a este assunto. O VISSIM oferece esta funcionalidade através dos estacionamentos que definem zonas de origem e destino e abre ao usuário a possibilidade de definir um programa próprio para computar rotas dinâmicas.

A tabela a seguir, resume os itens analisados nesta seção:

**Tabela 5.3: Classificação dos *softwares* quanto aos itens de atribuição de demanda**

Critér. \ Simul.	SUMO	ITSUMO	AIMSUN	VISSIM
Tipos de Veic.	Bom	Ruim	Ótimo	Ótimo
Matriz OD	Bom	Ruim	Ótimo	Bom
Regras Reg.	Ruim	Ruim	Ótimo	Ótimo
Interface	Regular	Ruim	Ótimo	Regular
Tráf. Ass. Din.	Bom	Bom	Bom	Bom

## 5.4 Simulação

A simulação analisada corresponde às possibilidades que o usuário tem para interagir com o *software* em tempo de execução. Algumas características da simulação também foram avaliadas.

### 5.4.1 Interface

A interface desta seção se refere à visualização dos veículos em movimento na via e aos filtros e opções disponíveis ao usuário. A simulação do SUMO mostra para o usuário o que está ocorrendo com determinada simulação, assim como a direção das vias e dos veículos. Apresenta interface de fácil compreensão seguindo o padrão de

janelas do Windows. Possui barra de menus para o usuário definir sua própria visualização, remover, adicionar botões e incluir diversas janelas na mesma aplicação.

A interface do ITSUMO não foi muito elaborada, existem linhas em verde para as vias e flechas vermelhas para os veículos. Porém, a simulação atinge o objetivo em ilustrar ao usuário o que ocorreu na simulação, assim como a direção das vias e dos veículos.

O AIMSUN possui interface com todos os itens analisados. Oferece para o usuário variedade nas visualizações disponíveis e possibilidade de customização. Paralelo ao AIMSUN, o VISSIM possui a mesma capacidade, porém não fornece opções pré programadas. Quando o programa inicia as vias são de cor cinza e os veículos, pequenas linhas coloridas. O usuário deve definir seus próprios modelos de veículos, vias, céu, terreno, etc. Em suma, deve modelar todo o ambiente para, através da interface em 2D ou 3D, ficar completo este critério. É permitido salvar para posteriormente utilizar as mesmas configurações.

#### **5.4.2 Ferramentas**

As ferramentas são opcionais que conferem ao usuário maior poder de interação com o *software*. O SUMO apresenta ferramentas que podem ser utilizadas durante a simulação para o usuário alterar funcionalidades que está visualizando, como acelerar o tempo da simulação, pausar, reiniciar, definir múltiplas janelas para visualizar, visualizar passo a passo, etc. É possível procurar vias e semáforos através de uma ferramenta para localizar.

O ITSUMO não apresenta muitas ferramentas que podem ser utilizadas durante a simulação, apenas consegue pausar reiniciar e passo a passo, além de movimentações da câmera do observador sobre o ambiente.

O AIMSUN inclui a possibilidade de aumentar em até cinco vezes a velocidade da simulação, escolher a quantidade de detalhes apresentado na simulação, verificar os atributos das vias e veículos em tempo real etc.

O VISSIM permite alterar a velocidade de simulação e verificar informações como os atributos dos veículos, as fases dos semáforos e os tempos de viagem nas vias.



### 5.4.3 Simular acidentes

Não foi definido explicitamente o modelo de simulação de acidentes nos *softwares*, porém a documentação, no caso do SUMO, explica ao usuário a oportunidade de deixar um veículo parado na via e configurar para reduzir a velocidade desta. Os demais *softwares* afirmam esta possibilidade através das extensões que podem ser adicionadas ao *software*.

A tabela a seguir, resume os itens analisados nesta seção:

**Tabela 5.4: Classificação dos *softwares* quanto aos itens de simulação**

Crítér. \ Simul.	SUMO	ITSUMO	AIMSUN	VISSIM
Interface	Bom	Regular	Ótimo	Ótimo
Ferramentas	Regular	Ruim	Ótimo	Ótimo
Acidentes	Regular	Regular	Regular	Regular

## 5.5 Aspectos gerais

Esta última seção aborda os aspectos que não se enquadraram em nenhuma outra classe, porém são considerados relevantes para um *software* simulador de trânsito.

### 5.5.1 Extensibilidade

Todos os *softwares* apresentaram algum tipo de extensão que pode ser criada pelo usuário sem a necessidade de reprogramação. No SUMO, existe esta preocupação em manter as extensões feitas por outros usuários ou equipes. Existem ferramentas que acessam direto o núcleo do simulador e controlam o processamento executando diversas tarefas. Enquanto outras extensões apenas tentam adicionar maior facilidade durante o uso, como um projeto tentando criar uma interface para criar topologias.

O ITSUMO possui o único modelo de extensibilidade visto entre os softwares analisados, o qual corresponde à utilização de agentes na simulação. Através destes é

possível programar o comportamento dos motoristas ou dos semáforos de forma totalmente independente ou interagindo entre si.

No AIMSUN existe um SDK (*Software Development Kit*) para executar este tipo de tarefa e é possível adicionar elementos a base de dados, criar novos editores, novas funções e filtros para utilizar em qualquer tipo de dado proveniente da simulação. O usuário utiliza um editor de scripts que acompanha a distribuição.

O VISSIM disponibiliza diferentes API (*Application Programming Interface*) como forma do usuário estender seus próprios modelos e também uma interface COM, que consiste em exportar os modelos e objetos do VISSIM para o software do usuário.

### **5.5.2 Ruído e poluição**

Este critério analisa algo que os *softwares* estão cada vez mais preocupados, a sustentabilidade do planeta, através de simulações do impacto ambiental na utilização das rodovias. O SUMO apresenta um relatório de ruído (*noise*) produzido pelos veículos durante os testes. Este modelo funciona de acordo com Harmonoise (escala de ruídos ambientais). Também foi encontrada informação com relação à poluição criada pelos veículos durante os testes, segundo a base de dados HBEFA (Manual de Fatores de Emissão para o Transporte Rodoviário).

Não foi encontrado nada no ITSUMO e no VISSIM em relação ao ruído (*noise*) e poluição produzidos pelos veículos durante os testes. No AIMSUN, durante a definição do modelo de veículo é possível configurar o quanto de cada substância o veículo está liberando para acelerar e desacelerar. Para o ruído nada foi encontrado.

### **5.5.3 Validação**

A validação consiste em determinar se a simulação realizada está próxima o suficiente do ambiente real de estudo, constatando uma representação exata do sistema. Consiste de uma característica importante nos softwares de simulação de trânsito, porém, apenas será avaliado se o software possui ou não esta funcionalidade.

No SUMO esta validação ocorre em pequenas unidades de testes para comparar se as funções estão retornando os valores esperados, também é executado um teste de

aceitação da aplicação contra o que foi declarado correto, este teste é executado a cada noite, utilizando mais de 2000 testes.

O AIMSUN e o VISSIM emulam o processo de detecção de trânsito e produzem uma série de observações simuladas que, ao serem comparadas com as medidas reais do caso de estudo, determinam a exatidão do comportamento do sistema. Existem ferramentas que auxiliam nesta conclusão.

#### 5.5.4 Vídeos

Disponibilidade de exportar um vídeo ao final de uma simulação é importante em alguns casos, como realizar apresentações formais. Somente o VISSIM apresentou esta opção. Possui diversos parâmetros para serem configurados e ao final da simulação gera um arquivo com extensão AVI.

#### 5.5.5 Resultados

Os *softwares* analisados possuem uma quantidade adequada de relatórios disponíveis, o ITSUMO que apresenta a menor quantidade e nenhuma opção de customização apresentou o pior desempenho deste critério. Os arquivos são escritos em simples documentos de texto e faltam informações básicas. O SUMO apresenta os resultados em arquivos no formato XML, que dificulta exportações para planilhas ou bancos de dados.

O AIMSUN possui um módulo somente de relatórios com diversas opções, inclui opções para o usuário configurar visualizações em tempo real gráficos dos resultados. No VISSIM não foi possível validar esta informação, porém as informações do manual indicam uma ferramenta adequada.

A tabela a seguir, resume os itens analisados nesta seção:

**Tabela 5.5: Classificação dos softwares quanto aos aspectos gerais**

Critér. \ Simul.	SUMO	ITSUMO	AIMSUN	VISSIM
Extensibilidade	Bom	Bom	Ótimo	Ótimo

Ruído e Poluiç.	Bom	Ruim	Bom	Ruim
Validação	Bom	Ruim	Bom	Bom
Vídeos	Ruim	Ruim	Ruim	Ótimo
Resultados	Ruim	Regular	Ótimo	Bom

## 5.6 Resultado final

No decorrer do capítulo, foram apresentadas tabelas que resumem as características de cada *software* e atribuído uma avaliação de acordo com a comparação com os outros softwares. A avaliação varia entre ruim, regular, bom e ótimo que correspondem respectivamente aos valores numéricos um, dois, três e quatro. Agregando as classes através de médias destes valores é possível obter um resultado final da avaliação dos softwares.

O resultado confirma o que foi percebido durante o desenvolvimento do trabalho. Desde o começo o AIMSUN apresentou superioridade em relação aos demais, visual moderno, facilidades de interação, manuais e tutoriais completos levaram a esta conclusão. Enquanto o VISSIM está próximo deste patamar, deve ser levado em conta que este não é o produto principal da empresa. A qual possui opções para diversas áreas relacionadas ao assunto, como planejamento estratégico, operação de transporte e logística.

Levando em consideração a gratuidade do software SUMO, este é considerado adequado para diversos tipos de pesquisa relacionados ao assunto. Existem algumas dificuldades na interface e faltam opções para certas configurações, porém estes são problemas clássicos encontrados em *softwares* livre. Seguindo a mesma linha de raciocínio, o ITSUMO apresenta potencial de crescimento se continuar sendo desenvolvido, pois no momento é considerado um software imaturo e precisa de casos de sucesso para se firmar no mercado.

## CONCLUSÃO

Os *softwares* simuladores de trânsito são ferramentas úteis no controle do tráfego urbano e rodoviário. Neste estudo, a análise realizada em quatro destes *softwares*, aponta vantagens e desvantagens entre eles.

Em praticamente todos os aspectos estudados, os *softwares* que pertencem a empresas privadas, no caso, AIMSUM e VISSIM apresentaram vantagens em relação aos disponibilizados sobre a licença GPL, neste estudo representado pelo SUMO e ITSUMO.

Os *softwares* privados, além de estarem a mais tempo no mercado, conseguem manter-se atualizados e com constantes inovações. Os *softwares* de distribuição gratuita analisados, entretanto, principalmente devido aos baixos investimentos da comunidade, ainda estão em fase inicial de desenvolvimento.

As técnicas desenvolvidas no trabalho envolveram os princípios básicos destes *softwares* ao iniciar uma simulação de trânsito efetiva, porém distante da realidade. Para se aproximar de situações verdadeiras, estas simulações precisam ser realizadas utilizando os conceitos de validação e verificação para fornecer dados confiáveis ao usuário. Durante a aquisição de dados para a simulação, a atribuição de demanda pode ser realizada utilizando-se de dados coletados na malha viária em questão. Visando aprofundar as conclusões do trabalho, com estes dados surge a opção de desenvolver um estudo abrangente sobre *dynamic traffic assignment* em cada simulador para aperfeiçoar o comportamento dos veículos em escolha de rotas.

O *software* livre permite ao usuário a liberdade de estudar o funcionamento do programa, adaptá-lo às suas necessidades, modificar o programa e liberar estas modificações, de modo que toda a comunidade se beneficie. Princípios opostos aos encontrados nos *softwares* comerciais, que acabam restringindo o usuário no acesso aos níveis mais fundamentais do funcionamento do *software*. Estas características tornam os

*softwares* livres uma opção para a comunidade, que deseja estudar e ampliar as funcionalidades deste tipo de programa.

Para cada situação de tráfego a ser analisada existe um *software* capaz de realizar a tarefa adequadamente. O estudo apresentou quatro simuladores de características distintas e oferecidas para públicos diferentes, os *softwares* livre e comerciais. Para os usuários que possuem maior afinidade na engenharia do tráfego, os *softwares* comerciais são mais indicados, pois possuem um suporte mais adequado além de ser um *software* de uso já consolidado. Entre os dois analisados o AIMSUN obteve pontuação superior ao VISSIM nos itens avaliados neste estudo. Os *softwares* livres exigem conhecimentos de informática e maior envolvimento nos conceitos básicos de simulação de trânsito. O código fonte é liberado para possíveis extensões, porém somente desenvolvedores e programadores possuem o conhecimento necessário para implementar estas tarefas. No caso analisado o SUMO obteve maior pontuação que o ITSUMO.

## REFERÊNCIAS

AIMSUN, **Página principal**. Disponível em: <<http://www.aimsun.com/>>. Acesso em dia 14 set. 2011.

BALMER, M. Travel Demand Modeling for Multi-Agent Transport Simulations: Algorithms and Systems, 2007, Tese de PhD ETH, Zurich

BARCELÓ, J. **Fundamentals of Traffic Simulation, International Series in Operations Research & Management Science**. [S.l.]:Springer Science Business Media. 2010.

BAZZAN, A. et al **ITSUMO: an Agent-Based Simulator for ITS Applications**. 4th Workshop on Artificial Transportation Systems and Simulation. 2010. [S.l.:s.n]

BAZZAN, A.; KLÜGL, F. Sistemas Inteligentes de Transporte e Tráfego: uma Abordagem de Tecnologia da Informação. In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA, **Anais**. [S.l.:s.n], 2007.

BAZZAN, A et al. Extending traffic simulation based on cellular automata: from particles to autonomous agents. In: Agent Based Simulation. **Proceedings**, [s.l.] ABS/ECMS. 2011

ITSUMO. **ITSUMO Main Page**, Porto Alegre, Disponível em: <<http://www.inf.ufrgs.br/maslab/traffic/itsumo/>>. Acesso em: dia 24 Mar. 2011

KRAJZEWICZ, D. **Kombination von taktischen und strategischen Einflüssen in einer mikroskopischen Verkehrsflusssimulation**, Th. Jürgensohn, H. Kolrep (Hrsg.). 2009.

May, A. **Traffic flow fundamentals**. New Jersey, Prentice Hall. 1990

NAGEL, K.; SCHRECKENBERG M. **A cellular automaton model for freeway traffic**. Journal de Physique [S.l.] v. I. 1992.

OSM- UCL VR CENTRE, IMPERIAL COLLEGE DE LONDRES EBYTEMARK.. **Open Street Maps**, Londres Disponível em: <<http://www.openstreetmap.org>>. Acesso em: dia 18 set. 2011.

ORTÚZAR, J. WILLUMSEN L.G Modelling Transport. [s.l.]: John Wiley & Sons, 3<sup>rd</sup> ed, 2001

PTV. **VISSIM 530 User's Manual Basics**. Berlim PTV. 2011

PTVAG. **VISSIM Main Page**, Berlim, Disponível em:  
<<http://www.ptvag.com/index.php?id=1801>>. Acesso em: dia 19 set. 2011.

SILVA, B. et al. ITSUMO: an Intelligent Transportation System for Urban Mobility. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 5., 2006, **Proceedings**. Hakodate, AAMAS. 2006.

SUMO . SUMO Main Page, [s.l.], Disponível em: <<http://sumo.sourceforge.net/>>. Acesso em: dia 25 Nov.2011

TIGER. TIGER Main Page, [s.l.], Disponível em  
<<http://www.census.gov/geo/www/tiger/tgrshp2011/tgrshp2011.html>>. Acesso em: dia 30 Nov. 2011

ROBOCUP. Robocup Rescue Main Page, [s.l.], Disponível em  
<<http://www.robocuprescue.org/>>. Acesso em: dia 30 Nov. 2011

VISUM. VISUM Main Page, [s.l.], Disponível em  
<<http://www.ptvag.com/software/transportation-planning-traffic-engineering/software-system-solutions/visum/>>. Acesso em: dia 30 Nov. 2011