



**Universidade de Brasília**

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# Simulação Baseada em Agentes para Transporte Colaborativo Utilizando MATSim

Bruno Tiago da Silva Santana

Monografia apresentada como requisito parcial  
para conclusão do Curso de Computação — Licenciatura

Orientadora  
Prof.<sup>a</sup> Dr.<sup>a</sup> Celia Ghedini Ralha

Brasília  
2015

Universidade de Brasília — UnB  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Curso de Computação — Licenciatura

Coordenador: Prof. Dr. Wilson Henrique Veneziano

Banca examinadora composta por:

Prof.<sup>a</sup> Dr.<sup>a</sup> Celia Ghedini Ralha (Orientadora) — CIC/UnB  
Prof. Dr. Pastor Willy Gonzáles Taco — ENC/FT/UnB  
Ronny Marcelo Aliaga Medrano — ENC/FT/UnB

### **CIP — Catalogação Internacional na Publicação**

Santana, Bruno Tiago da Silva.

Simulação Baseada em Agentes para Transporte Colaborativo Utilizando MATSim / Bruno Tiago da Silva Santana. Brasília : UnB, 2015.

72 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. simulação, 2. carona solidária

CDU 004.4

Endereço: Universidade de Brasília  
Campus Universitário Darcy Ribeiro — Asa Norte  
CEP 70910-900  
Brasília-DF — Brasil



# Dedicatória

Dedico este trabalho a todos que me motivaram a seguir em frente quando eu mesmo já não tinha mais ânimo ou interesse. Em especial, minha namorada, por me compreender e me acompanhar sempre, por me mostrar que devo seguir em frente independente das adversidades. Aos meus pais que me deram o suporte e incentivo desde pequeno a trilhar a vida acadêmica. A minha avó Iolanda, que mesmo nos seus últimos dias de vida, nunca deixou de se preocupar com meu bem estar.

# Agradecimentos

Agradeço a Deus em primeiro lugar, pela minha saúde e minha fé que se mantiveram firmes perante os desafios, e sua digníssima Esposa Maria que sempre intercedeu por mim desde que minha mãe me consagrou; à minha orientadora Célia por me dar apoio e um norte nos momentos de medo e indecisão. Por fim, aos meus familiares e colegas que suportaram meus piores momentos, sempre me dando suporte e palavras de ânimo.

# Resumo

Existe uma grande aceitação por parte da comunidade acadêmica, sobretudo entre os graduandos, ao projeto *Carona Solidária UnB*. Desta forma, a hipótese deste trabalho é que um estudo sobre ferramentas de modelagem e simulação baseado em agentes, com foco na ferramenta MATSim, para desenvolver um modelo de simulação deve auxiliar no desenvolvimento de um módulo inteligente para o sistema automatizado do projeto *Carona Solidária UnB*.

Neste trabalho foi desenvolvido um modelo de simulação utilizando o MATSim e dados reais de agentes coletados por entrevistas em 2013 no Campus Darcy Ribeiro. As simulações foram analisadas com o objetivo de adquirir conhecimento para auxiliar no desenvolvimento de um módulo inteligente a ser integrado ao sistema *Carona Solidária UnB*. Este trabalho tem como objetivo principal auxiliar o Plano Diretor de Mobilidade Urbana Sustentável da UnB a ser implantado em todos os Campi da UnB.

**Palavras-chave:** simulação, carona solidária

# Abstract

The *Carona Solidária UnB* project is widely accepted in the academic community, especially among the undergraduate students. Thus, this work states as hypothesis that a study of agent-based simulation tools, specially focusing the MATSim tool, can be useful to develop a simulation model to help to develop an intelligent module to the *Carona Solidária UnB* system.

In this work, a simulation model using MATSim and real data from agents, collected from interviews during 2013 in the Campus Darcy Ribeiro of UnB, was conducted. The simulations were analyzed to help gather knowledge to develop the intelligent module to be integrated to the *Carona Solidária UnB* system. This work has the main objective to assist in the Plano Diretor de Mobilidade Urbana Sustentável da UnB which is intended to be deployed in all UnB campi.

**Keywords:** simulation,carpooling

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problema . . . . .	2
1.2	Objetivos . . . . .	2
1.3	Hipótese da Pesquisa . . . . .	2
1.4	Organização do Documento . . . . .	2
<b>2</b>	<b>Fundamentação Teórica</b>	<b>4</b>
2.1	Mobilidade Urbana Sustentável . . . . .	4
2.2	Sistema de Informação Geográfica . . . . .	5
2.2.1	Histórico . . . . .	5
2.2.2	Definição e Características Principais . . . . .	6
2.2.3	Relação entre o SIG e o Geoprocessamento . . . . .	6
2.3	Tratamento de Dados . . . . .	6
2.3.1	Automato Celular . . . . .	7
2.3.2	Grafos . . . . .	7
2.4	Simulação Baseado em Agentes . . . . .	9
2.4.1	Motivação e Abordagens . . . . .	10
2.4.2	Características . . . . .	10
2.5	Ferramentas de Simulação Baseado em Agentes . . . . .	11
2.5.1	MATSim . . . . .	11
2.5.2	AnyLogic . . . . .	20
2.5.3	NetLogo . . . . .	23
2.5.4	RePAST . . . . .	24
2.6	Ferramentas de Simulação em Transportes . . . . .	25
2.6.1	Macrossimulação - Visum . . . . .	25
2.6.2	Microssimulação - Vissim . . . . .	26
<b>3</b>	<b>Apresentação da Proposta</b>	<b>28</b>
3.1	Metodologia . . . . .	29
3.2	Trabalhos Correlatos . . . . .	29
<b>4</b>	<b>Experimentos</b>	<b>31</b>
4.1	Simulação de Brasília e Entorno . . . . .	31
4.2	Análise dos Resultados . . . . .	33
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>35</b>



<b>Referências</b>	<b>37</b>
<b>A Histogramas da Simulação de Brasília</b>	<b>40</b>
<b>B Características do Vostro 5470</b>	<b>60</b>

# Lista de Figuras

2.1	Exemplo de grafos direcionados [1]. . . . .	8
2.2	Exemplo de subgrafo [1]. . . . .	8
2.3	Exemplo de grafo completo [1]. . . . .	9
2.4	Exemplo de grafo pleno ou cíclico [1]. . . . .	9
2.5	Exemplo de grafo bipartido [1]. . . . .	10
2.6	Dados de entrada do MATSim [26] . . . . .	14
2.7	Captura de tela do arquivo config.xml aberto no sublime text 2 [26] . . . .	14
2.8	Captura de tela do arquivo facilities.xml aberto no sublime text 2 [26] . . .	15
2.9	Captura de tela do arquivo plans.xml aberto no sublime text 2 [26] . . . .	16
2.10	Captura de tela do arquivo network.xml aberto no sublime text 2, mos- trando o cabeçalho e as arestas do grafo [26] . . . . .	16
2.11	Captura de tela do arquivo network.xml aberto no sublime text 2, mos- trando os vértices do grafo [26] . . . . .	17
2.12	Dados de saída do MATSim [26] . . . . .	17
2.13	Módulo de controladores da simulação do MATSim [26] . . . . .	18
2.14	Arquivo de saída do MATSim <i>run0.scorestats.png</i> . . . . .	19
2.15	Ciclo de simulação e otimização do MATSim [26] . . . . .	20
2.16	Ciclo de processamento do MATSim . . . . .	21
2.17	Sistemas dinâmicos com AnyLogic [16] . . . . .	21
2.18	Esquema de modelagem de um sistema dinâmico no AnyLogic . . . . .	22
2.19	Esquema de interação entre um modelo SD e agentes em um evento discreto	23
4.1	Fluxograma do desenvolvimento da simulação e a análise dos dados . . . .	31
4.2	Captura de tela da representação gráfica da simulação no VIA-SENOZON	33
A.1	Histograma da iteração 0 . . . . .	40
A.2	Histograma da iteração 1 . . . . .	41
A.3	Histograma da iteração 2 . . . . .	42
A.4	Histograma da iteração 3 . . . . .	43
A.5	Histograma da iteração 4 . . . . .	44
A.6	Histograma da iteração 5 . . . . .	45
A.7	Histograma da iteração 6 . . . . .	46
A.8	Histograma da iteração 7 . . . . .	47
A.9	Histograma da iteração 8 . . . . .	48
A.10	Histograma da iteração 9 . . . . .	49
A.11	Histograma da iteração 10 . . . . .	50
A.12	Histograma da iteração 11 . . . . .	51

A.13 Histograma da iteração 12 . . . . .	52
A.14 Histograma da iteração 13 . . . . .	53
A.15 Histograma da iteração 14 . . . . .	54
A.16 Histograma da iteração 15 . . . . .	55
A.17 Histograma da iteração 16 . . . . .	56
A.18 Histograma da iteração 17 . . . . .	57
A.19 Histograma da iteração 18 . . . . .	58
A.20 Histograma da iteração 19 . . . . .	59
B.1 Vostro 5470 . . . . .	60

# Capítulo 1

## Introdução

Há décadas, um problema cotidianamente enfrentado em todo o mundo, especialmente por quem mora em grandes metrópoles, é o aumento do fluxo de automóveis graças ao crescimento natural da população e do poder aquisitivo da mesma [33]. Na capital do Brasil, geralmente no início e no fim do horário comercial, as vias de acesso da cidade acabam congestionadas [13]. Essa realidade atinge diretamente a comunidade acadêmica da Universidade de Brasília (UnB) e por essa razão surgiram ideias diversas para solucionar tal problema. Dentre as soluções propostas, encontra-se o projeto de mobilidade sustentável chamado *Carona Solidária UnB*, o qual está em desenvolvimento desde 2012. Uma das iniciativas dentro deste projeto conta com a participação ativa de mais de 300 membros cadastrados no grupo do Facebook, chamado UnB Carona Solidária, criado em 2013 pela aluna de Comunicação Social, Thaís Ellen Rodrigues [31].

A tradição de caronas é antiga na UnB. Durante o período letivo é costumeiro observar alunos pedindo e dando carona em diversos horários e locais do campus, uma prática de solidariedade e socialização, bem vista pela maioria dos membros da comunidade acadêmica. Alguns cobram, outros praticam gratuitamente e combinam durante um ou mais semestres. As questões imediatas do projeto tem relação com as possibilidades das pessoas de participarem ativamente de uma rotina coletiva, frequente e assídua. O projeto *Carona Solidária UnB*, que está sendo desenvolvido por pesquisadores da UnB, tem como foco a implantação do Plano Diretor de Mobilidade Urbana Sustentável da UnB [31]. Coordenado pelo professor pesquisador da Pós-Graduação em Transportes da UnB Dr. Pastor Willy Gonzales Taco, o projeto visa alcançar todos os campi da UnB - Darcy Ribeiro, Planaltina, Gama e Ceilândia, e com o foco na implantação de ações voltadas para [12]:

- uso compartilhado solidário de veículos particulares;
- campanhas educativas de trânsito;
- utilização de modos de transporte sustentáveis;
- transporte público;
- transporte de cargas; e
- espaços de circulação e tráfego.

## 1.1 Problema

Com o crescimento acentuado dos membros da comunidade acadêmica da UnB e as restrições existentes no transporte público, torna-se muito importante um projeto de mobilidade sustentável, mas na atual conjuntura existem pontos que necessitam de solução. A sintonia no processo da carona depende de fatores de espaço e tempo. Informações sobre compromissos em locais e horários auxiliam na prática da carona, se e somente se, tais informações das agendas forem de fácil e rápido acesso. Por isso, um sistema que faça esse controle aumenta as possibilidades de um crescimento notável nas caronas, viabilizando a implantação do Plano Diretor de Mobilidade Urbana Sustentável da UnB.

## 1.2 Objetivos

O objetivo geral deste trabalho é o estudo e a definição de um modelo de simulação baseado em agentes para o transporte colaborativo na UnB. As simulações realizadas devem auxiliar na análise das rotas colaborativas dos membros da comunidade fazendo o melhor uso das variáveis tempo e espaço do modelo real.

Como objetivos acessórios podemos citar:

1. estudo de ferramentas de simulação baseada em agentes úteis no domínio de transporte;
2. estudo detalhado e uso do framework Multi-Agent Transport Simulation (MATSim) para viabilizar a realização de simulações de transporte colaborativo na UnB;
3. utilização de dados reais dos membros da comunidade da UnB que residem no Distrito Federal e entorno; e
4. analisar os dados de simulação para auxílio na definição de um módulo inteligente no projeto *Carona Solidária da UnB*.

## 1.3 Hipótese da Pesquisa

A hipótese deste trabalho é que o uso de um simulador baseado em agentes pode aumentar o entendimento do problema e auxiliar no desenvolvimento de um módulo inteligente capaz de suprir as necessidades do sistema proposto pelo projeto *Carona Solidária UnB*. Em outras palavras, os resultados da simulação podem servir para auxiliar o levantamento de requisitos para o desenvolvimento do módulo inteligente capaz de processar os dados das agendas da comunidade acadêmica dos campi da UnB e retornar os resultados para o sistema já existente ou novos.

## 1.4 Organização do Documento

No Capítulo 2 será descrito brevemente aspectos relacionados a teoria de mobilidade urbana sustentável, a teoria do sistema de informação geográfica, geoprocessamento, conceitos e definições de grafos como instrumento de representação dos mapas de rota dos agentes, modelos e ferramentas de simulação baseado em agentes, e o simulador MATSim

com maiores detalhes. No Capítulo 3 será apresentado a proposta de solução, junto com trabalhos correlatos que serviram de inspiração para este trabalho. No Capítulo 4 serão descritos os experimentos com os resultados alcançados. No Capítulo 5 são descritas as conclusões e sugestões de trabalhos futuros.

# Capítulo 2

## Fundamentação Teórica

Neste capítulo será apresentado os fundamentos teóricos relacionados a este trabalho, incluindo uma breve exposição sobre mobilidade urbana sustentável, juntamente com a legislação vigente no Distrito Federal. Também será apresentado conceitos de sistema multiagente e algumas ferramentas de simulação baseado em agentes. Serão expostos também outros conceitos utilizados neste trabalho, envolvendo sistema de informação geográfica, geoprocessamento e representação em grafos.

### 2.1 Mobilidade Urbana Sustentável

A humanidade depende de meios de transporte a séculos, como um agente do desenvolvimento, tornando possível o comércio global, o acesso aos serviços de saúde, educação e lazer, além do crescimento das áreas urbanas [24]. Os transportes têm influenciado positivamente o crescimento de teorias que relacionam acessibilidade e mobilidade ao progresso econômico e social. Levando em consideração que os transportes são essenciais para a manutenção de diversos setores da sociedade e ao mesmo tempo, são responsáveis por efeitos prejudiciais ao meio ambiente e a qualidade de vida, como a poluição do ar e o estresse dos congestionamentos [10].

Das diversas ações para minimizar o impacto negativo do excesso de automóveis, a prática do *carpooling* (carona solidária), força a diminuição de carros na rua e consequentemente os danos ao meio ambiente, melhorando a qualidade de vida, além de promover um espaço de sociabilização entre os participantes [11]. Tal prática é embasada na teoria da mobilidade urbana sustentável, que é defendida por muitos pesquisadores na área de engenharia de transportes. Segundo [5], de acordo com as dimensões do desenvolvimento sustentável, pode-se considerar que a mobilidade dentro da visão da sustentabilidade possa ser alcançada de duas formas: (i) relacionado com a adequação da oferta de transporte ao contexto socio-econômico; e (ii) relacionado com a qualidade ambiental. O Plano Nacional de Mobilidade Urbana é previsto na lei ordinária 12.587/2012 estando consoante com a visão da European Environmental Agency (EEA), a qual define cinco princípios urbanos de sustentabilidade [5], a saber:

- da capacidade ambiental, cidades devem ser projetadas e gerenciadas dentro dos limites impostos pelo seu ambiente natural;

- da reversibilidade, intervenções no ambiente urbano devem ser planejadas de maneira a ser reversíveis, tanto quanto possível de forma a não por em risco a capacidade da cidade de se adaptar a novas demandas por mudanças nas atividades econômicas e da população, sem prejudicar a capacidade ambiental;
- da resiliência, quanto mais resistente for a cidade, mais pressão externa será suportada;
- da eficiência ambiental e social, a visão de máximo benefício econômico por cada unidade de recurso ambiental utilizado e máximo aproveitamento de cada atividade econômica desempenhada por um ou mais habitantes da cidade; e
- da igualdade, a promoção do acesso a atividades e serviços para todos os habitantes.

## 2.2 Sistema de Informação Geográfica

A finalidade do Sistema de Informação Geográfica (SIG) é a virtualização de dados reais para o desenvolvimento de pesquisas que levem em conta o espaço, assim como as simulações de transportes, semelhante a que será desenvolvida neste trabalho. O objetivo da simulação de transportes é uma aproximação da realidade do deslocamento de pessoas no cotidiano, bem como, virtualizar o espaço e assim permitir o uso desses dados para prever eventos é um dos pontos fundamentais para o desenvolvimento do geoprocessamento.

### 2.2.1 Histórico

Desde as civilizações antigas, mapas vem sendo usados para registrar informações sobre a superfície da terra pois mapas armazenam informações de forma compacta e acessível. No século XX, a criação de grandes volumes de dados geográficos, apresentados no formato de mapas proporcionou análises sofisticadas para a época [4]. No entanto esse processo tinha suas limitações, os dados usados na geração do mapa deveriam ser generalizados para facilitar a interpretação, o que acometia na perda de detalhes. A atualização dos mapas é normalmente um procedimento caro e por isso era não era interessante para possíveis investidores e conseqüentemente desmotivava grupos ou pesquisadores [4].

Graças ao advento tecnológico e computacional, a obtenção, o armazenamento e a manipulação eficiente de grandes volumes de dados a custos relativamente baixos [4] possibilitou a utilização de imagens obtidas por sensores orbitais como ferramentas indispensáveis na análise e monitoramento de fenômenos espaço-temporais, devido suas características de repetitividade e periodicidade. SIG é a manipulação da informação conectada a um banco de dados geográficos, que possui dados espaciais e atributos referentes aos mesmos.

O desenvolvimento dos SIG começou nos EUA, na década de 60, por iniciativa dos pesquisadores Fisher, Tomlinson e Dangermond [7]. Fisher iniciou os trabalhos de mapeamento computadorizado, fez parte da fundação do SYMAP (Laboratório de Computação Gráfica da Universidade de Havard), ambiente no qual, foi desenvolvido o primeiro pacote de mapeamento computadorizado largamente distribuído para a manipulação de dados geográficos. Tomlinson participou da pesquisa florestal no leste da África junto a Air Service, ajudou no desenvolvimento do CGIS (Canadian Geographic Information System) que



foi utilizado como inspiração para o SIG, além de ajudar a promover as duas primeiras conferências internacionais sobre o SIG em Ottawa. E Dangermond foi um dos fundadores da Environment Systems Research Institute (ESRI), a primeira empresa norte americana que entrou para a área de SIG sem derivação das áreas de Desenvolvimento assistido por computador e Manufatura assistida por computador [4]. Mesmo assim, apenas em 1987, SIG recebeu seu reconhecimento científico, pelo estabelecimento do National Center for Geographic Information and Analysis (NCGIA) [7], sendo fonte de pesquisas frutíferas até a atualidade.

### 2.2.2 Definição e Características Principais

O SIG foi projetado para coleta, armazenamento e análise de objetos e fenômenos, sustentados pela localização geográfica [4]. Pela visão de Smith, "*SIG é um banco de dados indexados espacialmente, sobre o qual opera um conjunto de procedimentos para responder a consultas sobre entidades espaciais*" [23]. Geralmente os SIG utilizam dados geograficamente referenciados e dados não espaciais, incluindo operações de análise e suporte a decisões. O campo de aplicações e análises realizadas por um SIG é tão vasto quanto a disponibilidade de conjuntos de dados geográficos para a região em estudo. Um SIG possui cinco elementos básicos, que o caracterizam:

- hardware, plataforma computacional, periféricos de entrada e periféricos de saída;
- software, constituído de módulos computacionais que executam diversas funções, desde de a manutenção dos dados a aplicação de modelos matemáticos;
- dados, que é considerado o elemento fundamental para o SIG, geralmente é a parte mais cara da implantação do sistema;
- profissionais, pessoas responsáveis pelo projeto, implementação e o uso do SIG, normalmente equipes multidisciplinares são formadas; e
- métodos, técnicas, critérios e experiências, que irão nortear o uso do SIG na solução dos problemas propostos.

### 2.2.3 Relação entre o SIG e o Geoprocessamento

O geoprocessamento é definido como a integração das tecnologias de informática, mapeamento e sensoriamento remoto [4]. Existem diversos tipos de geoprocessamento, o que foi aplicado nesse trabalho é o processamento digital de imagens através do SIG *Open Streer Map* que utiliza as imagens disponibilizadas pelo *Google Maps*, essas imagens são transformadas em bancos de dados geográficos, no formato XML, com as coordenadas geográficas reais e outros atributos relacionados com as estradas, como faixas, velocidade limite e afins [26].

## 2.3 Tratamento de Dados

A interpretação e o tratamento dos dados disponibilizados para o desenvolvimento da simulação, se baseiam na visão de autômato celular e na definição de grafo. Autômato

celular é uma vertente da teoria da computação e auxilia a virtualização das ações dos alunos durante o deslocamento na simulação. E a definição de grafo, que parte de princípios matemáticos sobre as relações entre conjuntos, oferece uma interpretação para um tratamento computacional das rotas de deslocamento da simulação.

### 2.3.1 Automato Celular

Cada aluno será representado por um agente na simulação, assim como cada agente é considerado um autômato celular (AC). Um AC é uma idealização matemática de sistemas físicos cujo espaço e tempo são discretos, e as quantidades físicas assumem valores finitos e discretos. O AC consiste em uma estrutura uniforme regular, usualmente com uma extensão finita, com variáveis discretas ocupando diversos locais [28]. O estado do AC é completamente especificado pelos valores e variáveis de cada local [27]. As variáveis dos locais são atualizadas simultaneamente, baseadas nos valores das variáveis de seus vizinhos que as precedem no tempo, e de acordo com o conjunto definido de "regra local" [35].

Segundo [27], utilizando uma abordagem de autômato celular, a modelagem de fluxo do tráfego possui dois modelos: o microscópico e o macroscópico. O modelo microscópico descreve um o comportamento do tráfego com o início em entidades discretas interagindo entre si, sendo o alcance de simples modelos analíticos. O modelo macroscópico é a descrição de comportamentos agregados ao tráfego pela caracterização de relacionamentos fundamentais entre velocidade do veículo, fluxo e densidade. Por fim, o modelo mesoscópico é a aplicação híbrida dos dois, no qual começamos com uma modelagem microscópica que vai ganhando características de uma modelagem macroscópica. Nas simulações desenvolvidas neste trabalho, será abordada a modelagem microscópica, pois cada agente fará o papel de um aluno, a unidade mínima da simulação.

Apesar da construção de um autômato celular ser feita de forma simples, ele é capaz de representar comportamentos de alta complexidade. Para a maioria dos modelos de autômatos celulares, o único método para determinar a média dinâmica dos dados qualitativos do sistema é a execução de simulações em um computador com configurações globais, como é o caso do MATSim [26].

### 2.3.2 Grafos

Uma forma de representar mapas computacionalmente é através da utilização de uma definição clássica da matemática, o grafo. O grafo  $G = (V, E)$  é um conjunto não vazio e finito de vértices  $V(v_i)$  e um conjunto finito de arestas  $E(e_m)$  que se relacionam, e dependendo dessa relação se caracterizam de maneiras distintas [1]. Atualmente, no campo da Ciências da Informação e da Ciências da Computação, a teoria de grafos é uma ferramenta poderosa, podendo não só expressar diversas situações como também ajudar na resolução de problemas apresentados nas mesmas [3]. Os grafos podem ser classificados de diversas maneiras:

Conforme apresentado na Figura 2.1, existem grafos direcionados, também chamados de digrafos, que possuem um conjunto de arestas que ligam um conjunto de vértices que possuem direção e são denominadas como arcos;

Grafos parciais são a representação de um grafo  $G = (V, E)$ ,  $E'(e_j) \subseteq E(e_i), i < j$  com apenas algumas arestas ou arcos mais sempre todos os vértices, assim como o grafo a direita na Figura 2.1;

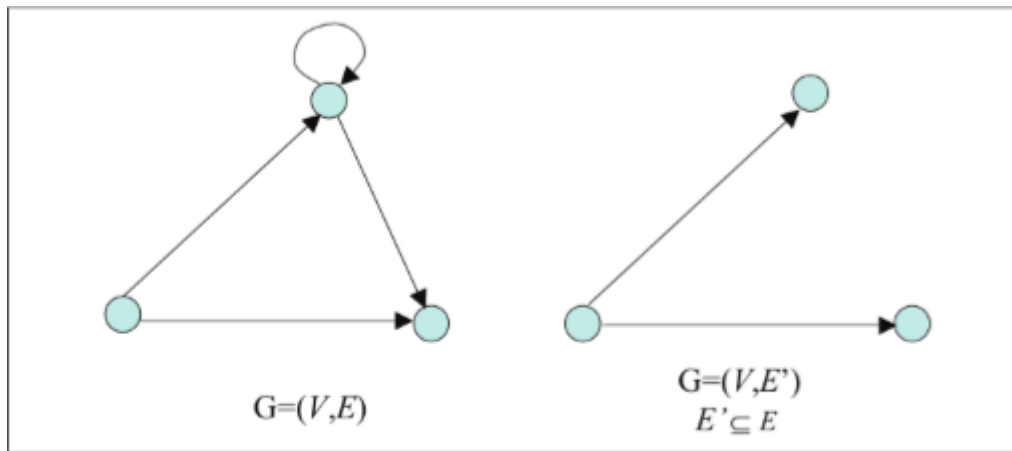


Figura 2.1: Exemplo de grafos direcionados [1].

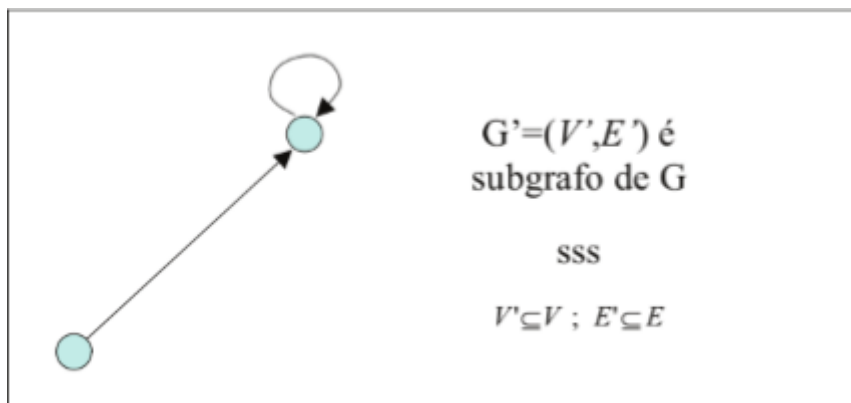


Figura 2.2: Exemplo de subgrafo [1].

O subgrafo  $G' = (V', E')$  ilustrado na Figura 2.2 é uma parte do grafo, separada para análise local, assim como será feito neste trabalho, separando o  $G' = (V', E')$  que corresponde ao Distrito Federal e uma parte do Entorno.

Conforme apresentado na Figura 2.3, os grafos completos, são aqueles que possuem um conjunto de arestas na situação em que nenhuma aresta tem intersecção com outra.

Conforme apresentado na Figura 2.4 existem grafos plenos ou cíclicos, os quais são bons para representar vias de deslocamento, dado que independente do vértice escolhido, se pode chegar a qualquer outro vértice.

Conforme pode ser visto na Figura 2.5, os grafos bipartidos apresentam os vértices que podem ser separados em dois grupos distintos, utilizados geralmente para representação de classificação.

Existem muitas outras classificações de grafos, mas as que foram citadas são as mais expressivas para o problema que será abordado nesse trabalho, tendo em vista um estudo prévio das possíveis representações, os grafos direcionados, bipartidos e subgrafos cíclicos

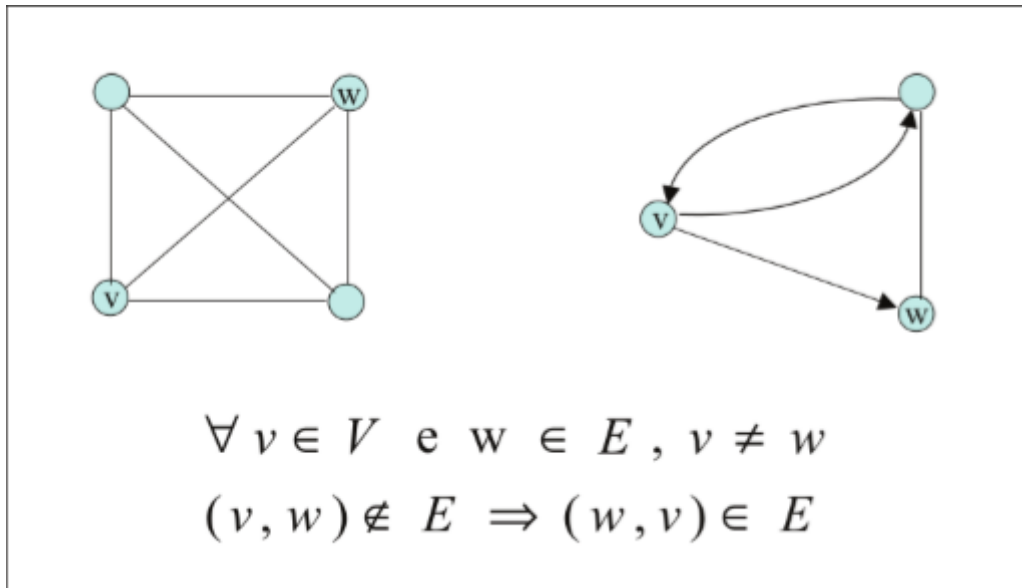


Figura 2.3: Exemplo de grafo completo [1].

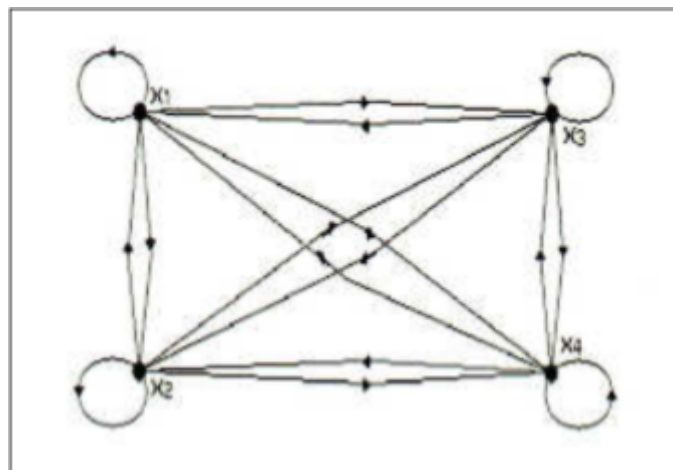


Figura 2.4: Exemplo de grafo pleno ou cíclico [1].

serão demasiadamente utilizados nas representações dos objetos que serão manipulados pelas ferramentas de simulação que serão citadas ainda neste capítulo.

## 2.4 Simulação Baseado em Agentes

Para prever situações futuras, analisar possibilidades com risco zero ou até mesmo acelerar o tempo, se o problema pode ser representado virtualmente, a simulação passa a ser atrativa para quem enfrenta tais casos ou semelhantes. Como apresentado anteriormente, agentes serão representações virtuais de pessoas e dentro dessa realidade virtual, desempenham atividades próximas as reais, com características atribuídas pelos desenvolvedores da simulação, dependendo apenas da possibilidade de representar computacionalmente cada ato. Ainda nessa perspectiva, a simulação baseada em agentes pode ser

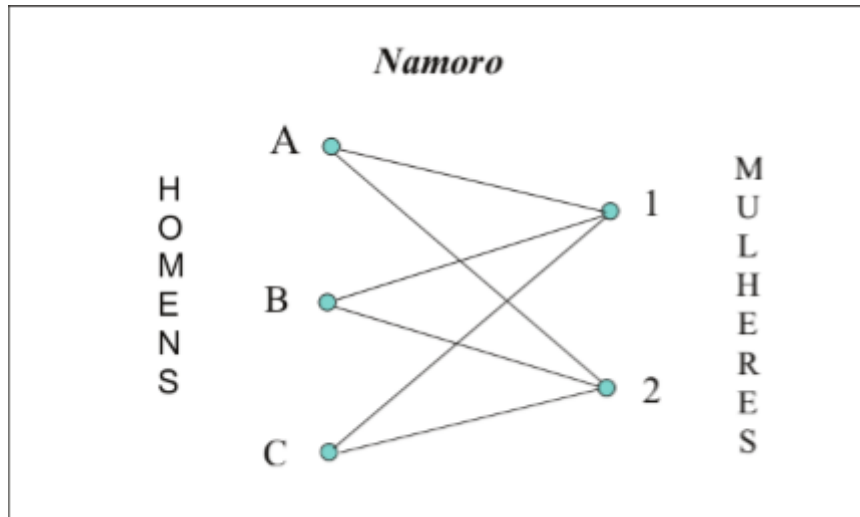


Figura 2.5: Exemplo de grafo bipartido [1].

vista como uma cópia simplificada de um sistema natural como será feito com o trânsito de Brasília neste trabalho.

### 2.4.1 Motivação e Abordagens

O surgimento da simulação baseada em agentes se deu através da observação de alguns sistemas naturais, nos quais é perceptivo o surgimento de um comportamento inteligente, por meio da interação de elementos do mesmo. Entre as características do Sistema Multiagentes (SMA), a interação entre os AC, o ambiente e a organização são as fundamentais para definição do SMA [19]. A área de SMA estuda o comportamento de um grupo organizado de agentes autônomos que cooperam na resolução de tarefas que apenas um agente não conseguiria executar. Para fazer essa análise é fundamental observar duas propriedades, aparentemente contraditórias, a autonomia de cada agente do sistema e a organização do sistema como um todo [15].

As abordagens no processo de desenvolvimento de SMA que costumam conciliar autonomia e controle são *top-down*, que inicia com aspectos coletivos que são refinados até a definição de agente e *bottom-up*, que parte da definição de aspectos individuais, relacionados aos agentes e com a combinação destes aspectos a definição de aspectos coletivos. As definições de interação e organização tem como base o ponto de vista dos agentes.

### 2.4.2 Características

O ponto de vista que é utilizado no desenvolvimento da simulação deste trabalho é o *bottom-up* [26], e possui as seguintes características [19]:

- os agentes são concebidos independentemente de um problema particular;
- a comunicação entre os agentes não é estritamente planejada, os protocolos de comunicação devem ser adaptivos a situações genéricas;
- a divisão de tarefas pode ser feita pelos próprios agentes; e

- não existe um controle centralizado para a solução de tarefas.

Um SMA com tais características é adaptivo e evolutivo, ou seja, tem a capacidade de adaptação a diversas situações, mesmo com a exclusão ou inclusão de agentes ou com a mudança da organização do sistema. É considerado uma *metáfora natural* para sistemas complexos e distribuídos [20], onde o conhecimento, o controle os recursos são distribuídos. Além da modelagem do sistema, a divisão de um problema e atribuição dos sub-problemas a agentes permite abstração e independência entre as partes do sistema [15].

A perspectiva *bottom-up*, é aplicável em ambientes heterogêneos e distribuídos, pois agentes com diferentes arquiteturas, que trabalham em plataformas distintas, distribuídas em na mesma rede computacional, podem cooperar na resolução de problemas. De maneira que, o uso das potencialidades particulares e pela distribuição de cada arquitetura, melhora o desempenho do sistema [20]. Além de permitir a concepção de sistemas abertos, onde os agentes podem entrar e sair de sociedades, mesmo que possuam objetivos distintos ou seja desenvolvidos por projetistas diversos, o que possibilita uma adaptabilidade ao sistema [19].

## 2.5 Ferramentas de Simulação Baseado em Agentes

Existem uma infinidade de frameworks para o desenvolvimento de simulações de diversos sistemas. Os apresentados nesse trabalho, foram selecionados pela possibilidade de representar a simulação desejada, pelo apoio didático oferecido pelos desenvolvedores e ou colaboradores e por fim, pelas referências mais recentes sobre trabalhos semelhantes ao proposto.

### 2.5.1 MATSim

O MATSim é recente [26] e tem um poder de expressividade interessante para a proposta do projeto *Carona Solidária UnB*. Utilizando uma metodologia quantitativa, o MATSim produz resultados sobre dados de deslocamentos que podem servir para a oferta de caronas. Além de oferecer também dados para análise da entrada, como média, gráficos e informações de comportamento da simulação.

MATSIM é uma ferramenta de simulação baseada em agentes, desenvolvida em JAVA 1.6 pelos grupos *Transport Systems Planning and Transport Telematics* e *Transport Planning*, que são liderados pelos professores *Kai Nagel* e *Dr. Kay W. Axhausen* em conjunto a companhia SENOZON [26]. A estrutura do MATSim é baseada em três módulos: a entrada, o ciclo de processamento da entrada e uma saída. A entrada é composta por arquivos na extensão XML (eXtensible Markup Language), contendo informações sobre as vias de deslocamento (*network*), horários de compromissos e maneiras de deslocamento (*plans*) e configurações para a otimização da simulação (*config*).

O ciclo de processamento possui três etapas e um módulo de controle que mantém a ordem e gera dados para a saída. A primeira etapa é a execução da simulação, onde os arquivos *network.xml*, *plan.xml.gz* e *config.xml* são carregados na memória e executados, sendo que a execução será avaliada por uma função de utilidade (Equação 2.1), a qual recebe parâmetros dos arquivos *config.xml*, *plan.xml.gz* e *network.xml*.

$$U = V + \epsilon \quad (2.1)$$

$$V = \beta * t_x * \ln(x/t_0) \quad (2.2)$$

A simulação é otimizada e uma nova simulação é gerada pelo módulo de controle, juntamente com um relatório de avaliação sobre as atividades. Nas Equações 2.1 e 2.2,  $V$  é a função de *Charypar-Nagel* e o  $\epsilon$  é a variação encontrada nas simulações utilizando os parâmetros de entrada e os resultados das interações anteriores, com exceção da primeira iteração que usa um valor tirado da média dos valores iniciais como base. onde  $\beta$  é tido como a duração típica da atividade,  $x$  é o deslocamento no espaço,  $t_x$  é o tempo de duração da atividade e  $t_0$  é o tempo inicial da atividade. Em resumo é uma função que nos retorna uma escala logarítmica em função do tempo  $\ln(x/t_0)$ .

Resultados, são portanto dados como, múltiplas interações associadas a um nível de confiança, enquanto que uma simples interação não é mais do que uma possibilidade dentro de uma gama de situações possíveis.

### Evaluación da performace dos planos com a função *Charypar-Nagel*

A função de avaliação do MATSim é descrita no [6]. A utilidade dos planos executados é computada como:

$$V_p = \sum_{i=1}^n (V_{perf,i} + V_{late,i} + V_{tr,i}) \quad (2.3)$$

com  $V_p$ , na Equação 2.3, como a utilidade acumulada de um dado plano  $p$  com  $n$  atividades,  $V_{perf,i}$  é a utilidade da performace da atividade  $i$ ,  $V_{late,i}$  é a desutilidade, i.e. a utilidade negativa, de uma atividade  $i$  que começou atrasada, e  $V_{tr,i}$  é a desutilidade do deslocamento de uma atividade  $i$  para uma atividade  $i + 1$ . Os planos são assumidos para envolver em torno de um dia de 24 horas, por esse motivo, a última atividade é assumida como sendo a mesma que a primeira, e em consequência, há tantas viagens como existem atividades. A utilidade positiva para a realização de uma atividade tem uma forma logarítmica, e é definida como:

$$V_{perf,i}(t_{perf,i}) = \beta_{perf} \cdot t_{*,i} \cdot \ln\left(\frac{t_{perf,i}}{t_{0,i}}\right) \quad (2.4)$$

Na Equação 2.4,  $t_{perf}$  é a duração efetiva da atividade  $i$  realizada,  $t_{*,i}$  é a duração média da atividade  $i$ ,  $\beta_{perf}$  dá a utilidade marginal de qualquer atividade em sua duração média.  $t_{0,i}$  está relacionada com a duração mínima e com a importância de uma atividade, mas não tem efeito enquanto a atividade de fim não faz parte da simulação. A desutilidade por estar atrasado é definido como:

$$V_{late,i}(t_{late,i}) = \beta_{late} \cdot t_{late,i} \quad (2.5)$$

Na Equação 2.5,  $\beta_{late}$  é a utilidade marginal que pode ser tardia (*utilidade/hora*), enquanto  $t_{late}$ ,  $i$  dão a quantidade de tempo (em horas) de uma atividade tardia  $i$ . Originalmente, a desutilidade de viajar em [6] é dada como:

$$V_{tr,i} = \beta_{tr} \cdot t_i \quad (2.6)$$

Na Equação 2.6  $\beta_{tr}$  é a utilidade marginal de viajar, dada em (*utilidade/hora*), convertendo os tempos de viagem em *utils* e  $t_i$  é o tempo de viagem (em horas) para viajar até a atividade  $i$ . A versão típica da  $V_{tr,i}$ , que não inclui alterações no veículo, mas se os preços de pedágio rodoviário é utilizado:

$$V_{tr,i} = \beta_{tr} \cdot t_i + \beta_{td} \cdot td_i + \beta_m \cdot m_i \quad (2.7)$$

com  $\beta_m$ , na Equação 2.7, como a utilidade marginal do dinheiro (*unidadesutils/monetrias*) e  $m_i$  como os custos de portagem monetários.

## Detalhamento da Instalação e Configurações

A instalação do framework MATSim é simples para programadores experientes em JAVA, embora exista um tutorial bem detalhado no site do mesmo, em inglês. Existem diversos trabalhos desenvolvidos com o MATSim, em sua maioria, os dados utilizados para a geração das simulações são oriundos de fontes públicas, logo a única dificuldade real está na implementação das configurações, pois os parâmetros afetarão de maneira crítica as avaliações de utilidade.

O MATSim possui um controle de reescrita, que impede que o simulador reescreva as saídas das simulações. Localizado dentro do algoritmo básico de controlador, conforme apresentado no Algoritmo 2.

---

### Algorithm 1: controlador [26]

---

```
package tutorial

import org.matsim.core.config.Config
import org.matsim.core.config.ConfigUtils
import org.matsim.core.controller.Controller

public class MyFirstController

public static void main(String[] args)
Config config = ConfigUtils.loadConfig("input/config.xml");
Controller controller = new Controller(config);
//controller.setOverwriteFiles(true); // comentar para não ocorrer sobre escrever
controller.run();
```

---

A Classe *Controller* é a classe inicial que chama todas as demais e assim, executando a simulação. Os *import.org.matsim.core* são utilizados para fazer a conexão com as classes do núcleo da API, que por sua vez passam os parâmetros para as demais classes e após a execução da simulação, geram os resultados exibidos em gráficos, logs e arquivos compactados. As classes que apoiam diretamente a classe *Controller* são as classes *Config* e *ConfigUtils* que abrem o arquivo *config.xml* e carregam em memória todos os parâmetros para a simulação.

## Entrada do MATSim

A entrada do MATSim é composta por um grupo de arquivos em XML, extraídos de diversas fontes, onde pessoas cedem dados reais para que sejam representados virtualmente por agentes. Esses dados podem ser obtidos por meio de pesquisa de campo, oriundos de repositórios ou de alguma outra fonte confiável. O MATSim possui uma



classe (*ConvertOSM*) que converte arquivos OSM, extraídos do *Open Street Maps*, que corresponde ao mapa de estradas. Existem classes que geram os agentes através de parâmetros de horário e coordenadas georeferenciadas (*POnePersonPopulationGenerator*). Todo o tratamento das entradas é feito através de arquivos DTD que são nativos do próprio MATSim, transformando os dados em XML no formato certo para a simulação.

Assim como mostrado na Figura 2.6, os arquivos principais da entrada são *network.xml*, *plans.xml.gz* e *config.xml* (parameters), os demais não são usados na simulação. Os parâmetros são carregados para a memória e então é feita a simulação, os mesmos DTD's que são utilizados para gerar os arquivos de entrada são utilizados para interpretar as entradas.

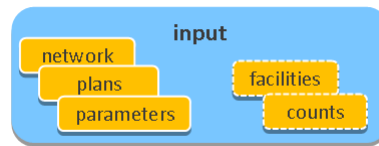


Figura 2.6: Dados de entrada do MATSim [26]

- *config.xml* - Conforme é mostrado na Figura 2.7, o arquivo *config.xml* contém as configurações básicas para a simulação, locais dos demais arquivos para a simulação, quantidade de iterações, valores de variáveis para o cálculo de utilidade de cada rota processada em cada iteração de cada agente, valores das atividades com seus respectivos tipos.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v1.dtd"
3 [
4     <!ENTITY OUTBASE "./output">
5     <!ENTITY INBASE  "./input">
6 ]
7 >
8
9 <!-- Definição de arquivos de entrada e saída -->
10
11 <config> <!-- Início do arquivo de configurações -->
12
13     <!-- ===== -->
14
15     <!-- Sistema de coordenadas -->
16     <module name="global">
17         <param name="randomSeed" value="5399998864266990000" />
18         <param name="coordinateSystem" value="CH1903_LV03" />
19         <param name="numberOfThreads" value="4" />
20     </module>
21
22     <!-- ===== -->
23
24     <!-- Entrada do arquivo network.xml (onde os agentes irão percorrer) -->
25     <module name="network">
26         <param name="inputNetworkFile" value="&INBASE;/network.xml" />
27     </module>
28

```

Figura 2.7: Captura de tela do arquivo *config.xml* aberto no sublime text 2 [26]

- facilities.xml.gz - Conforme a Figura 2.8 o arquivo contém as informações de cada local frequentado pelos agentes, horário de abertura e fechamento, tipo do local e id do local.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE facilities SYSTEM "http://www.matsim.org/files/dtd/facilities_v1.dtd">
3 <facilities>
4
5 <!-- ===== -->
6
7   <facility id="0" x="-8174682.89325" y="-4885678.56027">
8     <activity type="home">
9       <opentime day="wk" start_time="00:00:00" end_time="24:00:00" />
10    </activity>
11  </facility>
12
13 <!-- ===== -->
14
15   <facility id="1" x="-8175928.0456" y="-4883153.62738">
16     <activity type="home">
17       <opentime day="wk" start_time="00:00:00" end_time="24:00:00" />
18    </activity>
19  </facility>
20
21 <!-- ===== -->
22
23   <facility id="2" x="-8175818.06546" y="-4882860.34694">
24     <activity type="home">
25       <opentime day="wk" start_time="00:00:00" end_time="24:00:00" />
26    </activity>
27  </facility>

```

Figura 2.8: Captura de tela do arquivo facilities.xml aberto no sublime text 2 [26]

- plans.xml.gz - Assim como na Figura 2.9 o arquivo armazena os dados das agendas dos alunos, a forma de deslocamento em cada deslocamento, coordenadas com id's dos locais, horários de saída dos locais e tipos de atividade desenvolvida em cada local.
- network.xml - As Figuras 2.10 e 2.11 o arquivo contém um grafo onde cada nó do grafo é a intercessão entre duas ou mais estradas, possuindo um id e coordenadas globais fornecidas pelo OSM (Open Street Map). Cada aresta representa uma estrada e possui um id de identificação, sentido da via (dado pelo nó inicial e nó final), comprimento da via, velocidade limite, capacidade de veiculos, faixas, *legs* permitidas (formas de deslocamento) e id original dado também pelo OSM.

## Saída do MATSim

O programa Visualizer da VIA/SENOZON foi feito para visualizar graficamente os deslocamentos, fazendo análises pontuais ou coletivas, dependendo do tipo de agentes que estiverem descritos no arquivo plans.xml.gz. ou dos parâmetros de visualização escolhidos no Visualizer. Outro fator externo que pode alterar o a saída é o tipo de licença, pois uma licença gratuita não permite a inserção de mais de 500 agentes, por isso foram utilizados poucos agentes durante a geração da simulação de Brasília. Após a simulação bem sucedida, o MATSim gera arquivos de análise quantitativa e qualitativa. Eventos de transporte público não são interessantes para a proposta deste artigo, mas convém mencionar que o MATSim é capaz de diferenciar a maneira com a qual cada agente desloca pelas vias permitidas.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE plans SYSTEM "http://www.matsim.org/files/dtd/plans_v4.dtd">
3
4 <plans>
5
6 <!-- ===== -->
7
8   <person id="0" age="18" employed="yes">
9     <plan selected="yes">
10       <act type="h1" facility="0" x="-8174682.89325" y="-4885678.56027" end_time="07:10:00" />
11       <leg mode="car" dep_time="07:10:00">
12         </leg>
13       <act type="e14" facility="100" x="-8172407.77068" y="-4885482.42901" end_time="07:55:00" />
14       <leg mode="car" dep_time="07:55:00">
15         </leg>
16       <act type="s2" facility="117" x="-8205208.98278" y="-4874469.63204" end_time="09:55:00" />
17       <leg mode="car" dep_time="09:55:00">
18         </leg>
19       <act type="e14" facility="101" x="-8172372.02108" y="-4884400.0689" end_time="13:55:00" />
20       <leg mode="car" dep_time="13:55:00">
21         </leg>
22       <act type="s2" facility="117" x="-8205208.98278" y="-4874469.63204" end_time="18:55:00" />
23       <leg mode="car" dep_time="18:55:00">
24         </leg>
25       <act type="h1" facility="0" x="-8174682.89325" y="-4885678.56027" />
26     </plan>
27   </person>
28
29 <!-- ===== -->
30
31   <person id="1" age="22" employed="yes">
32     <plan selected="yes">
33       <act type="h1" facility="1" x="-8175928.0456" y="-4883153.62738" end_time="07:10:00" />
34       <leg mode="car" dep_time="07:10:00">

```

Figura 2.9: Captura de tela do arquivo plans.xml aberto no sublime text 2 [26]

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE network SYSTEM "http://www.matsim.org/files/dtd/network_v1.dtd">
3 <network>
4
5 <!-- ===== -->
6
7   <nodes>
8     <node id="1017335024" x="-8211733.0792511245" y="-4861090.321612632" />
9     <node id="1017335029" x="-8211707.4461615635" y="-4861097.336467277" />
10    <node id="1017335681" x="-8211728.4251719825" y="-4861041.017828668" />
11    <node id="1017335683" x="-8211702.900438252" y="-4861058.06078097" />
12    <node id="1017339267" x="-8211645.086168529" y="-4860565.479611186" />
13    <node id="1017339271" x="-8211667.618438526" y="-4860522.144843103" />
14    <node id="1017339286" x="-8211672.165817671" y="-4860561.186090009" />
15    <node id="1017339293" x="-8211639.539915924" y="-4860530.4304040475" />
16    <node id="1017359346" x="-8210980.195611352" y="-4861642.106760521" />
17    <node id="1017359351" x="-8211033.110363942" y="-4861628.676494338" />
18    <node id="1017394343" x="-8211727.298445748" y="-4861303.1033666255" />
19    <node id="1017399971" x="-8211662.795524453" y="-4865581.996543572" />
20    <node id="1017411102" x="-8212447.191417123" y="-4861134.5249154465" />
21    <node id="1017411122" x="-8212477.129653184" y="-4861090.827401794" />
22    <node id="1017411124" x="-8212444.370838196" y="-4861100.216330739" />
23    <node id="1028268451" x="-8184286.318466594" y="-4894111.912553297" />
24    <node id="1028268453" x="-8184336.630386023" y="-4894452.233385542" />

```

Figura 2.10: Captura de tela do arquivo network.xml aberto no sublime text 2, mostrando o cabeçalho e as arestas do grafo [26]

Criando a possibilidade de gerar uma interação entre o transporte privado e o público. Durante a criação da simulação, a quantidade de memória reservada para a simulação foi de 2000 MB, bem mais que no tutorial que utiliza 700 MB, isso foi proposto não pela simulação em si, mas pela geração do arquivo *network.xml* que necessitou de mais memória para ser gerado. O Visualizer foi feito para indicar as coordenadas georreferenciadas, implicando numa acurácia impressionante e também em possibilidades de interações en-



```

82917 </node id="987213781" x="-8199887.6831333133" y="-4879338.838819247" />
82918 <node id="997954876" x="-8211731.471973727" y="-4857988.973194638" />
82919 <node id="997960634" x="-8211961.521874647" y="-4857886.614497681" />
82920 </nodes>
82921
82922 <!-- ===== -->
82923
82924 <links capperiod="01:00:00" effectivecellsize="7.5" effectivecellwidth="3.75">
82925 <link id="1" from="1481533557" to="1406402856" length="71.92693292524224"
freespeed="12.5" capacity="600.0" permlanes="1.0" oneway="1" modes="car" origid
="134810557" />
82926 <link id="10" from="1406402859" to="1406402863" length="103.06896713085735"
freespeed="12.5" capacity="600.0" permlanes="1.0" oneway="1" modes="car" origid
="134810557" />
82927 <link id="100" from="1371120370" to="2187670532" length="79.45156611729256"
freespeed="16.666666666666668" capacity="2000.0" permlanes="2.0" oneway="1"
modes="car" origid="26414666" />
82928 <link id="1000" from="1557155533" to="1557155531" length="159.75658376118392"
freespeed="8.333333333333334" capacity="600.0" permlanes="1.0" oneway="1" modes
="car" origid="142267456" />
82929 <link id="10000" from="2479609635" to="2479609546" length="251.68039351235393"
freespeed="8.333333333333334" capacity="600.0" permlanes="1.0" oneway="1" modes
="car" origid="240209925" />
82930 <link id="100000" from="2396390722" to="3213374848" length="22.21386900043661"

```

Figura 2.11: Captura de tela do arquivo network.xml aberto no sublime text 2, mostrando os vértices do grafo [26]

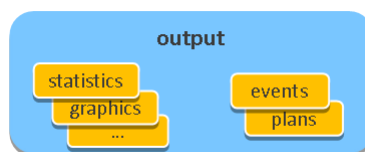


Figura 2.12: Dados de saída do MATSim [26]

tre estados e até países. Cada iteração terá gráficos respectivos e otimizados dos casos anteriores.

De maneira geral podemos dizer que o MATSim otimiza os casos através das iterações, quanto mais iterações melhor será a economia de percurso por cada agente, mas essa quantidade tem um limite que depende da quantidade de agentes circulando num espaço. A proporção não é claramente citada no tutorial, mas é indicado a utilização de pelo menos 10 iterações em 500 agentes. A Classe geradora do arquivo *network.xml* também é utilizada para gerar o arquivo *count.xml* que por sua vez é opcional para a geração da simulação, pois os DTD's são carregados na classe e o arquivo *count.xml* depende diretamente do arquivo *network.xml* para ser gerado. Outra possibilidade interessante é utilizar um software QGIS que através dos shapefiles permite a geração do arquivo *counts.xml*. Graças a visualização gráfica dada pelo QGIS, tendo uma noção geoespacial do ambiente, é prático aplicar as contagens em cada ponto do mapa e gerando assim o *counts.xml* que é responsável pela informação de quantidade de pessoas que se encontram em um agente ou em um *facility*.

Depois que uma simulação acaba o MATSim gera uma saída composta por uma pasta com os planos, eventos e histogramas de cada interação. Dois arquivos de log e quatro arquivos de avaliação qualitativa sobre o comportamento dos agentes durante toda a

simulação. O arquivo de saída mais interessante para este trabalho é o *events.xml.gz* pois é nele que se encontram as rotinas de deslocamento de cada agente.

## Controlador do MATSim

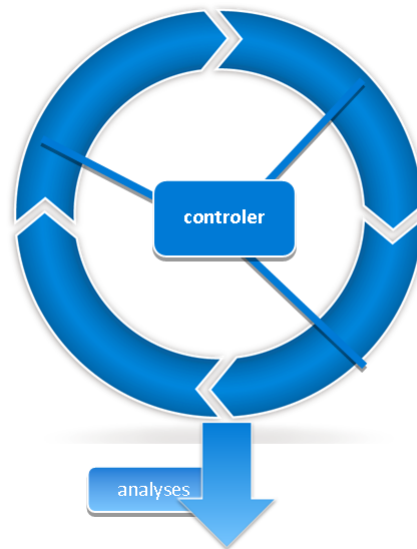


Figura 2.13: Módulo de controladores da simulação do MATSim [26]

Assim como está representado na Figura 2.13, As classes de controle são o centro da simulação, é através desse *core* que todas as decisões são tomadas. Os dados carregados na memória são entregues as classes de controle que repassam para as de cálculo de utilidade. O algoritmo base utilizado para o cálculo de caminho para cada agente é o famoso algoritmo Dijkstra, Algorithm 2, que escolhe o menor caminho sempre. A classe *CharyparNagelOpenTimesScoringFunction*, calcula a utilidade para os percursos dos agentes e assim os *scores* são feitos. O percurso de melhor *score* é selecionado e é posto no arquivo de saída *events.xml.gz*, as médias de todas as interações das simulações são feitas e postas no arquivo de log *run0.scorestats.txt*, como demonstrado na Figura 2.14.

Conforme é representado na Figura 2.15, toda simulação é baseada em um ciclo: primeiro os percursos são definidos e avaliados (*scoring*), depois são otimizados através do replanejamento (*replanning*). Tendo em vista, que a responsabilidade de gerenciar todas essas atividades é da classe *Controler* que faz parte do pacote *org.matsim.api.core* (O núcleo da API).

Vale ressaltar que é possível redefinir a função de utilidade *Charypar-Nagel* (Equação 2.2) na classe *CharyparNagelOpenTimesScoringFunction*, desde que ela utilize os mesmos parâmetros. Assim como é possível adicionar classes que costumizem eventos gerados, bem como toda a saída produzida pela simulação. Além de existir a possibilidade de editar a classe de impressão *printCounts* e da classe de leitura *ReadEvents*.

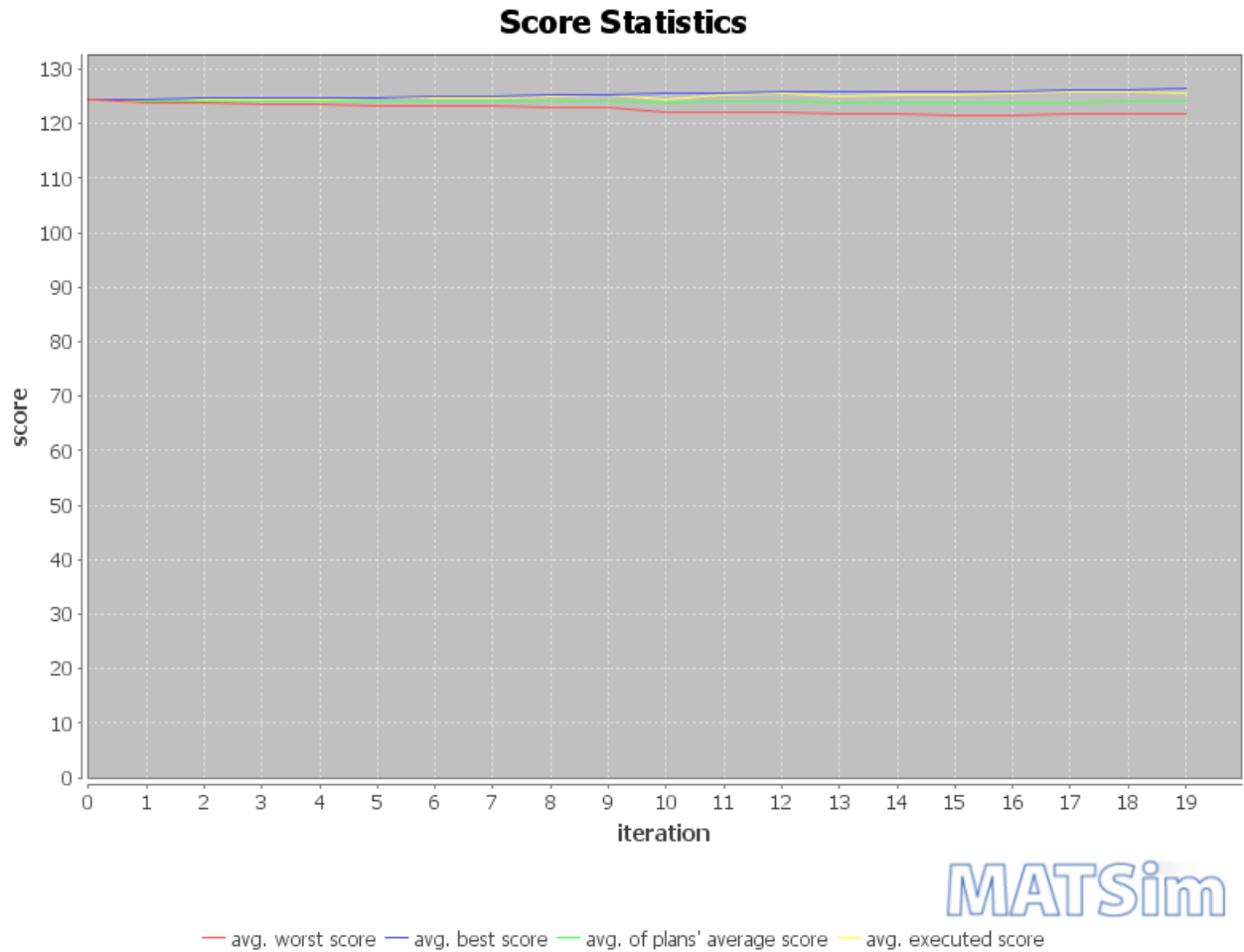


Figura 2.14: Arquivo de saída do MATSim *run0.scorestats.png*

---

#### Algorithm 2: Dijkstra [14]

---

```

Function Dijkstra (graph, source)
  Initialize the cost of each node
  Initialize the cost of the source to 0
  while there are unknown nodes left in the graph do
    Select the unknown node with the lowest cost: n
    Mark n as known
    foreach node a which is adjacent to n do
      | a's cost = min(a's old cost, n's cost + cost of (n,a))
    end
  end
end

```

---

#### Aspectos sobre o projeto

A Figura 2.16, demonstra a integração da entrada, do ciclo de processamento e a saída das simulações feitas pelo MATSim, no tutorial apresentado no site do framework MATSim, a lição 8 na verdade é a proposta de fazer uma simulação de uma outra região (no tutorial é feita a região de Zurique), para a complementação do projeto que é de cunho *open source* [26]. São citados exemplos no site e dentro do framework também são encontrados.

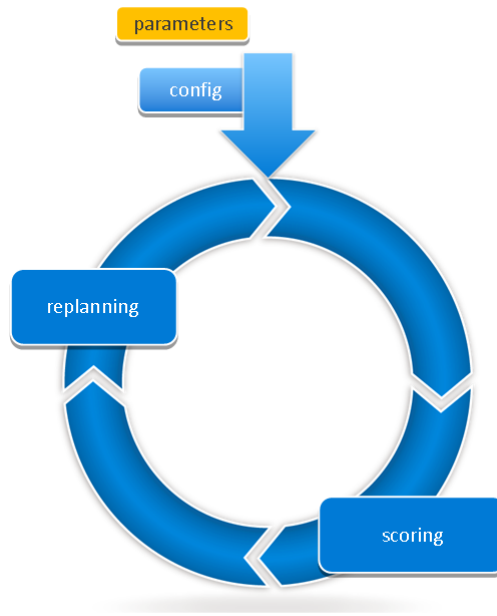


Figura 2.15: Ciclo de simulação e otimização do MATSim [26]

## 2.5.2 AnyLogic

O framework AnyLogic é baseado na perspectiva de um Sistema Dinâmico (SD), que por sua vez, é um conjunto de ferramentas conceituais que nos permitem compreender a estrutura e dinâmica de sistemas complexos. SD é também um método de modelagem rigorosa que nos permite construir simulações de computador formais de sistemas complexos e usá-los para conceber as políticas e as organizações mais eficazes. Juntas, estas ferramentas permitem-nos para criar gestões de simuladores de voo em micro mundos onde o espaço e o tempo podem ser compactados e diminuir para que possamos experimentar os efeitos colaterais a longo prazo das decisões, a aprendizagem de velocidade, desenvolver o nosso entendimento de sistemas complexos, e estruturas e estratégias de design para o maior sucesso [16].

A metodologia de SD é tipicamente utilizada em longo prazo, em modelos estratégicos e assume um elevado grau de agregação dos objetos a ser modelado. Pessoas, produtos, eventos e outros itens discretos são representados em modelos SD por suas quantidades, de maneira que eles perdem quaisquer propriedades individuais, histórias ou dinâmica. Este nível de abstração é apropriado para o problema proposto nessa monografia. Contudo, se detalhes individuais são importantes, você sempre pode voltar a conceituar todo ou parte do seu modelo usando *Agent Based* ou *Discrete Event* sem nunca sair do ambiente *Anylogic*.

### Modelagem de Sistema Dinâmico no AnyLogic

Anylogic apoia o projeto e simulação de estruturas de feedback (diagramas de estoques e de fluxos e regras de decisão, incluindo variáveis de matriz ou índices) de forma a maioria dos modeladores do SD são usadas para:

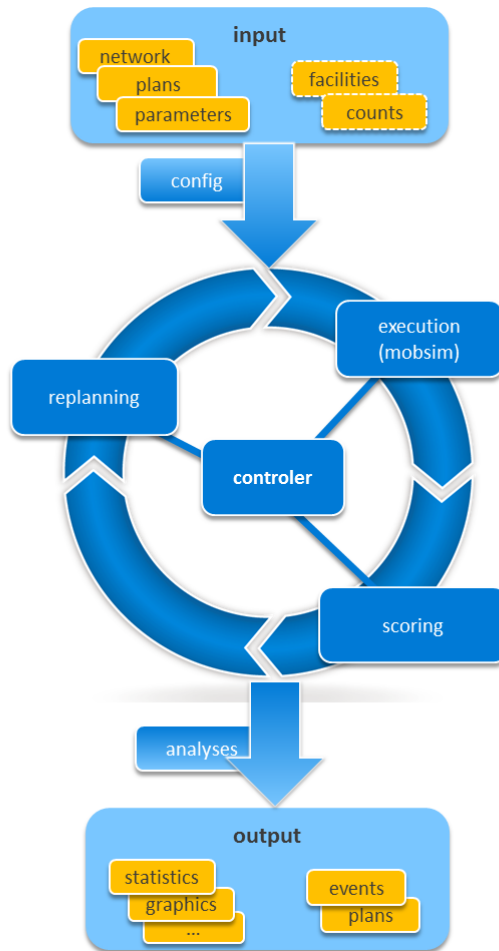


Figura 2.16: Ciclo de processamento do MATSim

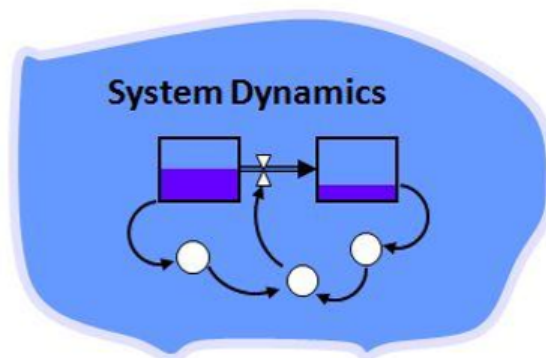


Figura 2.17: Sistemas dinâmicos com AnyLogic [16]

- definir estoques e fluxos variáveis uma a uma ou usando da opção *flow tool*;
- utilizar a conclusão automática de códigos em fórmulas pela opção *code completion*;
- definir variáveis *shadow* para melhor legibilidade do seu modelo;
- use as funções de tabela com passo, linear, ou interpolação de spline;





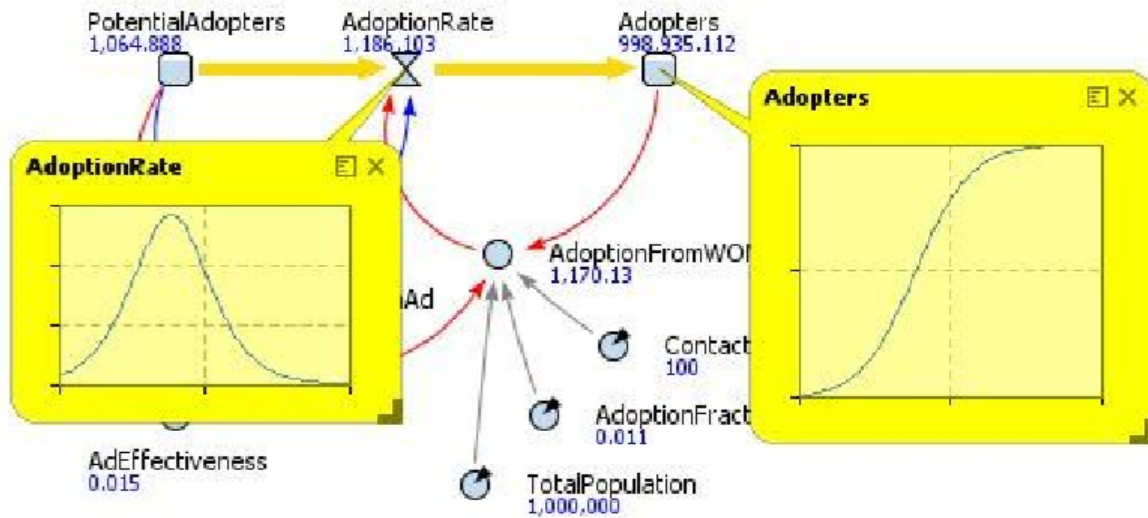


Figura 2.19: Esquema de interação entre um modelo SD e agentes em um evento discreto

### AnyLogic X MATSim

O AnyLogic é capaz de realizar a mesma tarefa que o MATSim. A diferença está apenas nas ferramentas de apoio, VIA/SENOZON e o QGis que permitem uma visualização e uma manipulação e análise gráfica dos dados não suporta a saída do AnyLogic. Embora as entradas possam ser adaptadas, utilizando ou até mesmo remodelando o DTD disponibilizado dentro da lib do MATSim. Outra coisa interessante do AnyLogic é que ele não possui limitação na quantidade de agentes para simulação, mas como não existe a necessidade para utilização de uma grande quantidade de agentes, essa vantagem não é um fator determinante para a troca de framework.

### 2.5.3 NetLogo

NetLogo é um ambiente de modelagem de simulações, multi-agente [2], desenvolvido na linguagem Java e suportado por diversas plataformas. Ele é capaz de desenvolver simulações fenômenos complexos de cunho natural ou social. Foi escrito por Uri Wilensky em 1999 [34]. É um sistema *open source*, que possui portabilidade entre os sistemas operacionais mais populares (Linux, Windows e MAC), além de um suporte ao conjunto internacional de caracteres.

No quesito programação, ele possui uma sintaxe acessível. Possui um grande vocabulário de primitivas de linguagem embutidos, o que facilita a leitura do código. Suas funções de calculos foram desenvolvidas com precisão dupla em ponto flutuante e por fim permite a concorrência em multi-plataforma [34].

Sobre ambiente, o NetLogo possui um centro de comando para interação *on-the-fly*, não somente na máquina onde está ocorrendo a simulação mas a qualquer máquina dentro da rede da hospedeira da simulação e que possua o cliente HubNet. Possui uma interface gráfica interativa e intuitiva, áreas de anotação sobre cada evento da simulação, funções de exportação e importação de dados da simulação, além da gravação de filme da simulação. A capacidade de modelagem em 2D e 3D e a visualização também em 2D e 3D.

## NetLogo X MATSim

Uma das características mais fortes do NetLogo é a possibilidade de desenvolver análises sobre simulações microscópicas, macroscópicas e também mesoscópicas. Consequentemente demonstrando assim a flexibilidade com a qual foi projetado e desenvolvido. Assim como o MATSim ele possui diversos exemplos de simulações pré-escritos, tais exemplos abordam as áreas ciências naturais e sociais, incluindo biologia, medicina, física e química, matemática, ciência da computação economia e psicologia social.

Outro fato em comum com o MATSim é a existência de um software que auxilia na usabilidade dos dados, no caso do NetLogo, o HubNet, permite o controle de um agente, durante a simulação, além do fato que é de leve processamento, ou seja, pode ser executado de calculadoras gráficas.

A razão pela qual o MATSim foi escolhido é a praticidade de gerar um mapa onde os agentes podem deslocar, o NetLogo não disponibiliza tal suporte. Mesmo assim é possível o desenvolvimento, embora mais trabalhoso.

### 2.5.4 RePAST

O *RePAST* (Recursive Porous Agent Simulation Toolkit), é um framework de modelagem de simulação por agentes e é uma derivação do framework *Swarm* [8]. *RePAST* foi inicialmente concebida como uma biblioteca de classes Java que iria trabalhar simplificar geração de simulações da *Swarm*. Essa concepção inicial foi o resultado de preocupações dos pesquisadores da *University of Chicago* com a complexidade do framework *Swarm* que inicialmente foi feita em *Objective-C* e também pelo respeito a maturidade e elegância da mesma.

Esta noção do framework *RePAST* servir como uma extensão para *Swarm* foi abandonada por uma série de razões, dentre elas a mais taxativa era a parcial redundância com o lançamento do framework *Swarm* escrita em Java. Antes da liberação da *Swarm* escrita em Java, os pesquisadores da *University of Chicago* desenvolveram um *framework* completamente independente em Java, mas emprestando as principais abstrações presentes no framework *Swarm*. Convencidos da viabilidade e utilidade para os pesquisadores da *University of Chicago*, a exploração inicial se transformou na atual versão da *RePAST* [8].

#### Características

As metas do desenvolvimento da *RePAST* foram definidos a partir da preocupação de atender as necessidades dos pesquisadores da *University of Chicago* que almejavam uma ferramenta interativa e adaptiva, bem como responder as preocupações sobre a extensibilidade e robustez. Tais objetivos são respondidos pelo desenvolvimento do projeto do *RePAST*, através da abstração da infra-estrutura de simulação, extensibilidade, e desempenho [8].

*RePAST* abstrai a maioria dos elementos-chave e representa-os como classes que cooperam para a criação de simulações baseadas em agentes. A versão atual do framework fornece uma classe exemplo que serve como modelo para as abstrações de infra-estruturas comuns de uma simulação baseada em agentes (como programação, coleta de dados, e assim por diante) e uma variedade de componentes genéricos para a construção de elementos de virtualização de ambientes reais. Esses componentes genéricos incluem espaços onde os

agentes executaram ações e alguns tipos de agentes genéricos. *RePAST* é particularmente forte em seu apoio à rede (social e afins) simulações. Este apoio é tanto de infra-estrutura (layouts de gráfico, a geração de rede, poupança e carregamento) como representacional (nó padrão e as classes de borda) [22].

A extensibilidade do framework cresce em grande parte do sucesso do projeto e implementação de abstrações-chave, de modo que o pesquisador pode usar essas abstrações como base para os seus próprios modelos. Ao fazer uso de algumas das abstrações do *Swarm*, *RePAST* herdou um design testado pelo tempo que contribui para a sua extensibilidade. Ao implementar algumas dessas abstrações, utilizando padrões de projeto a extensibilidade é novamente reforçada. Por exemplo, o mecanismo de escalonamento (o objeto Programar e as várias classes de ação) é implementado de acordo com o padrão de design composto, que permite que o código do cliente para tratar ações individuais e as composições dessas ações uniformemente. Isto proporciona clareza ao mecanismo de programação [22]. Extensibilidade é também fornecida pela utilização de Java como uma linguagem de implementação. Linguagens orientadas a objeto permitem a criação de quadros extensíveis através da utilização de herança e composição [19].

O desempenho refere-se a um nível de desempenho que é aceitável quando comparados com os outros benefícios da API [22]. Enquanto otimizações de desempenho não faziam parte do projeto inicial do *RePAST*, o cuidado foi tomado para minimizar a criação do objeto e atingir a velocidade de exibição aceitável [22]. Isso não quer dizer que as preocupações de desempenho são ignoradas; há melhorias de desempenho incrementais com cada lançamento. *RePAST* oferece um desempenho comparável a simulações semelhantes e só vai ficar mais rápido com a melhoria contínua de máquinas virtuais Java. Como resultado dessas metas, *RePAST* é robusto, extensível, embora o pesquisador necessite de conhecimentos sobre a linguagem Java [22].

## RePAST X MATSim

O framework *RePAST* é capaz de desenvolver a simulação deste trabalho, por causa da capacidade adaptativa, extensibilidade. Mas a complexibilidade dos métodos utilizados para desenvolver a simulação levaria mais tempo, uma vez que, o MATSim já possui um apoio completo para o desenvolvimento e análise de simulações de transporte. Logo optou-se pelo MATSim, mesmo sem avaliar a diferença de desempenho entre os frameworks.

## 2.6 Ferramentas de Simulação em Transportes

Nesta seção serão apresentados dois simuladores para uso em transporte, sendo uma ferramenta com macrossimulação - Visum e outra com microssimulação - Vissim, ambas desenvolvidas pela PTV Group [17, 18].

### 2.6.1 Macrossimulação - Visum

O Visum é um software desenvolvido pela PTV para análise de tráfego, planejamento e gestão baseada em dados de SIG. Ele gera macrossimulações através de modelos de forma consistente para diversos tipos de veículos que circulam em estradas e suas interações. Especialistas em transportes utilizam o Visum para modelar redes de transporte e demanda

de viagens, analisar as mudanças de tráfego esperado, planejar serviços de transporte público, desenvolver estratégias avançadas e soluções de transporte. As previsões das simulações do Visum são baseadas em modelos de oferta e demanda. Isto permite uma modelagem realista e a análise de impacto do tráfego. Através da criação de bancos de dados o Visum possibilita a análise de viabilidade econômica e de impacto ambiental. Outra característica é o modelo de exportação de rede que procura modelos de transporte e os volumes correspondentes para o Vissim [18].

O Visum cria análises, estatísticas e relatórios. Obtendo a representação detalhada das seguintes análises [18]:

- Comparação de cenários;
- Matrizes de análise em histograma;
- Cálculo do fluxo do feixe;
- Busca interativa do caminho mais curto;
- Isócrono;
- Análise do impacto ambiental; e
- Análise da incidência de acidentes.

O Visum interage com as seguintes interfaces [18]:

- Interface COM para aplicações baseadas em script;
- Interfaces com sistemas de programação, p. ex. Google Transit alimentação, Hafas, VDV 452, DIVA, railML;
- Interfaces de exportação, por exemplo, forma, arquivos SVG; e
- Interfaces de importação, p. ex. Forma, OpenStreetMap, Saturno, TransCAD, Emme, arquivos de cubo.

### 2.6.2 Microsimulação - Vissim

O Vissim é um software feito pela PTV para modelar microsimulações, permitindo a análise do tráfego local, a comparação com diferentes tipos de cruzamentos de dados e a análise da implementação de medidas de prioridade ao transporte público e do impacto de um plano de tráfego leve. O Vissim permite o desenvolvimento de modelos que podem representar todos os usuários de um via pública e estudar suas interações: carros, de carga ou qualquer tipo de transporte público, além de ferroviários ou rodovias [17].

O software oferece flexibilidade em muitos aspectos: o conceito de arcos e conectores permitem aos usuários modelar a geometria de qualquer tipo. As características dos condutores e veículos permitem parâmetros individuais dentro de uma única simulação. Além de existir uma variedade de interfaces que podem ser integradas a outros sistemas de controle como: semáforo, gestão de tráfego ou modelos de emissão.

As opções de análises do Vissim são abrangentes e permitem avaliar e planejar tanto infra-estrutura viária urbana, bem como uma ferramenta inter-urbana. Este software pode

obter resultados numéricos detalhados em animações 3D que descrevem vários cenários. É uma infraestrutura ideal para as propostas com agentes responsáveis pela tomada de decisões [17].

# Capítulo 3

## Apresentação da Proposta

A proposta deste trabalho é utilizar o *framework* MATSim e os dados coletados pela equipe do projeto *Carona Solidária UnB* durante o semestre 2013/1, para desenvolver uma simulação de abordagem microscópica que nos leve a uma conclusão sobre a possibilidade do uso dessa ferramenta para o desenvolvimento de um módulo inteligente que retorne ofertas automaticamente para os usuários do projeto *Carona Solidária UnB*.

O MATSim é disponibilizado gratuitamente no site oficial [26], tendo em vista que, outros programas foram utilizados neste trabalho para a análise dos resultados da simulação, contendo licenças gratuitas ou licença para fins educacionais. ArcGis, para visualizar os grafos em formato OSM e SHP, que representam os mapas de Brasília e entorno. Google Earth, para localizar os *facilities* que cada estudante indicou em sua agenda. Sublime Text, para abertura dos arquivos em XML e OSM em formato texto, além dos arquivos *run0.logfile.log*, *run0.logfileWarningsErrors.log*, *run0.scorestats.txt*, *run0.stopwatch.txt* e *run0.traveldistancestats.txt* que possuem resultados quantitativos da simulação e são de suma importância para a análise dos resultados.

Com relação a entrada, foram entregues em tabelas no formato XLS (extensão padrão do Microsoft Excel), contendo os dados das agendas semanais de 122 alunos, locais que costumam frequentar assiduamente. Foram selecionados e utilizados dados de 100 alunos de faixa etária variada, entre 18 e 50 anos, com diversas ocupações e tarefas cotidianas. Não existe uma identificação real, apenas dados reais sobre a rotina semanal. Foram marcados ao todo 171 instalações sendo 100 moradias (casas ou apartamentos, sem situação definida), 17 instalações da UnB (departamentos, blocos de sala de aula e laboratórios) e 54 instalações diversificadas entre academias, lojas, shoppings, restaurantes, empresas de pequeno e médio porte, ministérios, supermercados, entre outros lugares com a finalidade de representar o trabalho remunerado ou voluntário.

Cada agente gerado terá o comportamento de um autômato celular e a regra local será definida pelos parâmetros da simulação, assim como explicado na fundamentação teórica. De forma análoga, como os alunos serão representados por agentes com comportamento AC, as vias de deslocamento por onde os alunos passam cotidianamente, serão representados por um grafo. A simulação terá uma abordagem microscópica, pois se baseia em agentes individuais e a rotina de cada um deles. Os resultados gerados pela simulação servirão para concluir se existe a possibilidade de utilização dos mesmos para o desenvolvimento de uma análise de requisitos para a concepção de um módulo de inteligência para o software do projeto *Carona Solidária UnB*.

## 3.1 Metodologia

Através do estudo de metodologias de simulação por agentes, será gerado um modelo de simulação para o transporte coletivo proposto pelo projeto *Carona Solidária UnB*, que leve em consideração as localidades, horários e revezamento entre caroneiros e caronistas. Utilizando o framework MATSim que apresenta funcionalidades necessárias para o desenvolvimento desse modelo, tais como geração de agentes de diversas identidades e comportamentos, geração de mapas que correspondem as rodovias reais com características como faixas, velocidade limite da via e capacidade da via. Será levando em consideração as abordagens de modelos microscópica, macroscópica e mesoscópica de simulação de transportes.

A base teórica desta monografia é uma aplicação inovadora na abordagem de pesquisas de transporte colaborativo, que utiliza a visão *bottom up*, ao contrário da maioria das pesquisas dessa mesma estirpe que utiliza a visão *top down*. Segundo Nicholson [29], as pesquisas possuem erros de aproximação quanto a aplicação de simulações como no método de Monte Carlo. O simulador MATSim, forma populações de agentes através da similaridade entre eles que representam agendas de indivíduos. Neste sentido, os agentes que possuem deslocamentos em horários e destinos paralelos são transformados em agentes populacionais [26]. Além de apontar as melhores rotas para cada um dos agentes utilizando para isso o algoritmo *Dijkstra* [14] em conjunto com a função de avaliação de utilidade de *Nager-Charypar* [6].

## 3.2 Trabalhos Correlatos

O trabalho de Morais [25] serviu como um norte para iniciar uma base teórica sobre simulações. É esperado chegar a mesma conclusão relatada em [25]: "É necessária a evolução do framework para que ocorra a interação entre os transportes já existentes". Utilizando as saídas, pode-se gerar um relatório de similaridade entre as rotas de agentes distintos e desse modo, sugerir possíveis caronas, sem a perda das rotinas que se aproximam da realidade, uma vez que no cotidiano, as pessoas tendem a ser assíduas em seus compromissos.

O foco de Morais é a possível virtualização do comportamento do transporte sobre trilhos, o que diverge do objetivo deste trabalho. Não são citados impactos no meio ambiente ou na sociedade, apenas o desenvolvimento de uma ferramenta capaz de gerar simulações de agentes com comportamento inteligente de autômatos celulares. A dificuldade de representar situações aleatórias é citada como um trabalho futuro [25], Morais foca em quadros possíveis e não em exceções. A interação entre outros tipos de transporte público não é retratado também.

Silveira possui uma visão consistente do problema da mobilidade urbana sustentável. Procura por meio de levantamentos bibliográficos de comportamento em transportes e programas de carona em conjunto com pesquisas de opiniões com formulários aplicados ao público alvo, a possibilidade da implantação do projeto *Carona Solidária UnB* [32]. A previsão de possíveis dificuldades na implantação do projeto foram respondidas através de questionários em redes sociais. A hipótese deste trabalho é uma das recomendações do trabalho de Silveira que possibilitou o começo do projeto *Carona Solidária UnB*, além



de servir como base, norteou muitas outras pesquisas que têm como objetivo o desenvolvimento a mobilidade urbana sustentável nos campus da UnB.

A divergência com o [32] é o foco, pois agora sabemos da aprovação da comunidade acadêmica e das dificuldades para a implantação do projeto. Todavia um aplicativo já foi desenvolvido por um grupo estudantil da UnB, liderado pelo ex-aluno Márcio Batista [30]. Este trabalho, caso a hipótese seja comprovada, auxiliará na evolução do aplicativo.

Dentro da proposta de Junior, foi feito um trabalho similar ao de Silveira em escala menor, mas com o mesmo foco, práticas que auxiliam a mobilidade urbana sustentável, como a aceitação da prática do *carpooling*, a criação e a utilização consciente de ciclovias. Junior defende a ideia de constantes investimentos para que as iniciativas dêem frutos [21]. A visão de Junior foi aproveitada neste trabalho, reforçando a de Silveira em [32], juntamente com a parte de mobilidade urbana sustentável na fundamentação teórica. A diferença está apenas no foco, Junior tentar provar sua hipótese através de exemplos de práticas sustentáveis que já foram desenvolvidas ou estão em implantação.

A hipótese de Moura sobre o *carpooling* foi aproveitado neste trabalho, pois durante todo o desenvolvimento da pesquisa desenvolvida, Moura questiona a aceitação da carona solidária [11]. Além da questão de mobilidade urbana sustentável de Botucatu (São Paulo), que é largamente tratada. Diversos integrantes do corpo acadêmico foram entrevistados e através do teste de Goodman, foram feitas análises estatísticas descritivas. Os resultados se mostraram favoráveis a implantação de um programa de carona na instituição alvo. Ao contrário da abordagem microscópica da simulação desenvolvida neste trabalho, Moura tange sua hipótese baseada numa pequena amostragem e dessa forma adquirindo uma margem de erro nas suas expectativas.

# Capítulo 4

## Experimentos

Neste capítulo será exposto o experimento feito através dos dados cedidos pelo projeto *Carona Solidária UnB*, aplicando os tratamentos das classes do MATSim que formatam a entrada dos dados para que a simulação possa ser executada. Em seguida, uma breve análise dos resultados emitidos pelo MATSim, em conjunto com uma descrição do ambiente no qual foi realizado a simulação.

### 4.1 Simulação de Brasília e Entorno

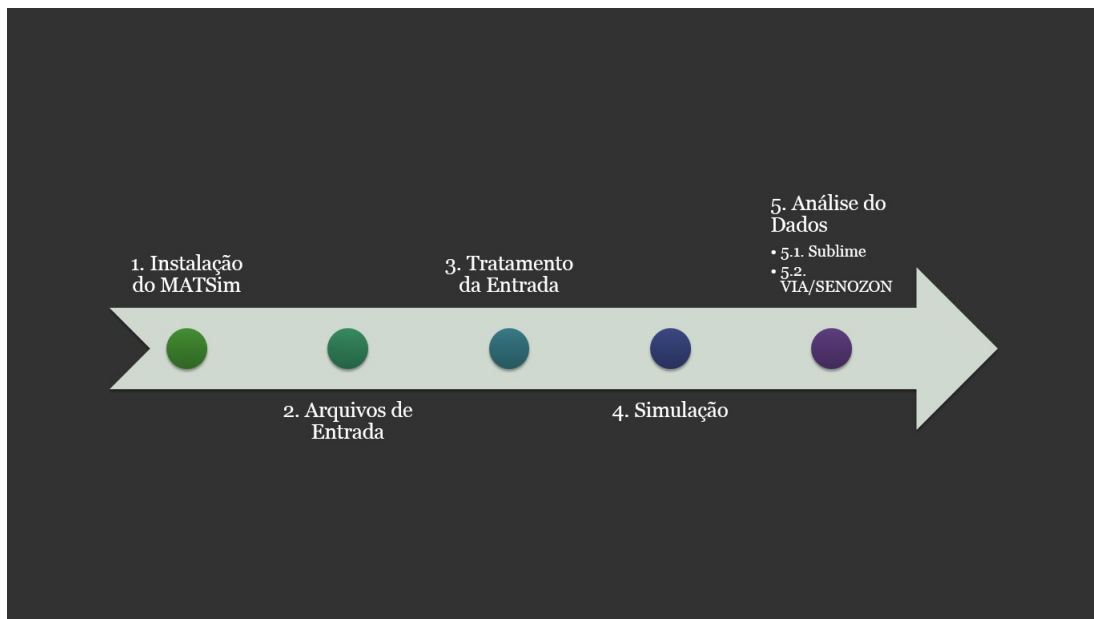


Figura 4.1: Fluxograma do desenvolvimento da simulação e a análise dos dados

A simulação foi realizada seguindo os do fluxograma que está na Figura 4.1, na etapa "1 - Instalação do MATSim" do fluxograma foi seguindo dos seguintes passos:

1. download da API no site [26];
2. criação de um projeto no programa Eclipse Versão 4.3.2 com a inclusão da API conforme o tutorial disponibilizado no site [26]; e

3. criação das classes de controle, criação de agentes e conversão do mapa, conforme o tutorial citado no passo anterior.

Na etapa "2 - Arquivos de Entrada" do fluxograma foram seguidos os seguintes passos:

1. captura do mapa do Distrito Federal e Entorno pelo site do Open Street Map no formato OSM;
2. conversão do arquivo OSM para o arquivo *network.xml* pela classe *ConvertOSM*; e
3. inserção manual dos dados na classe *POnePersonPopulationGenerator*, responsável pela geração do arquivo *plans.xml.gz*.

Na etapa "3 - Tratamento de Entrada" do fluxograma foram seguidos os seguintes passos:

1. verificação manual do arquivo de configuração *config.xml*, mantendo os parâmetros similares ao exemplo proposto pelo tutorial na simulação de Zurique [26]; e
2. verificação dos dados presentes nos arquivos *network.xml*, *plans.xml.gz* e *facilities.xml.gz* através da abertura no arquivo *network* no programa VIA/SENOZON.

Na etapa "4 - Simulação" do fluxograma foram seguidos os seguintes passos:

1. foram realizadas inúmeras simulações com os dados, por essa razão foram utilizadas cópias dos dados originais e a configuração de reescrita na classe de controlador conforme citado na Seção 2.5.1.
2. após testes, pelo gráfico descrito na Figura 2.14, foi definido a utilização de 20 iterações para realizar a análise dos dados de saída, pois os valores de *score* convergem para a melhor curva a partir da nona iteração e converge para o melhor resultado esperado a partir da décima quarta iteração, se matendo sempre com pouca diferença; e
3. A última simulação foi a única a ser considerada satisfatória para a análise, as demais foram tratadas como testes para a aprendizagem do uso da ferramenta e desconsideradas como resultado final.

E por fim, na etapa "5 - Análise dos Dados" do fluxograma foram seguidos os seguintes passos:

1. verificação do gráfico descrito na Figura 2.14, que mede a qualidade dos *scores* das trajetórias dos agentes;
2. verificação dos arquivos de log e registros de atividades em cada iteração (histogramas, *run0.0.events.xml.gz* e demais relatórios);
3. verificação da integridade dos arquivos de saída através da abertura com o VIA/SENOZON;
4. a simulação pode ser observada graficamente através do VIA/SENOZON como é representado na Figura 4.2;

5. o tempo da simulação está dentro do arquivo *run0.logfile.log*, tempo de início 2015-05-25 21:02:29 e o tempo de término 2015-05-25 21:06:10; e
6. a simulação foi realizada em um Vostro 5470, a descrição detalhada da máquina está no Anexo B.

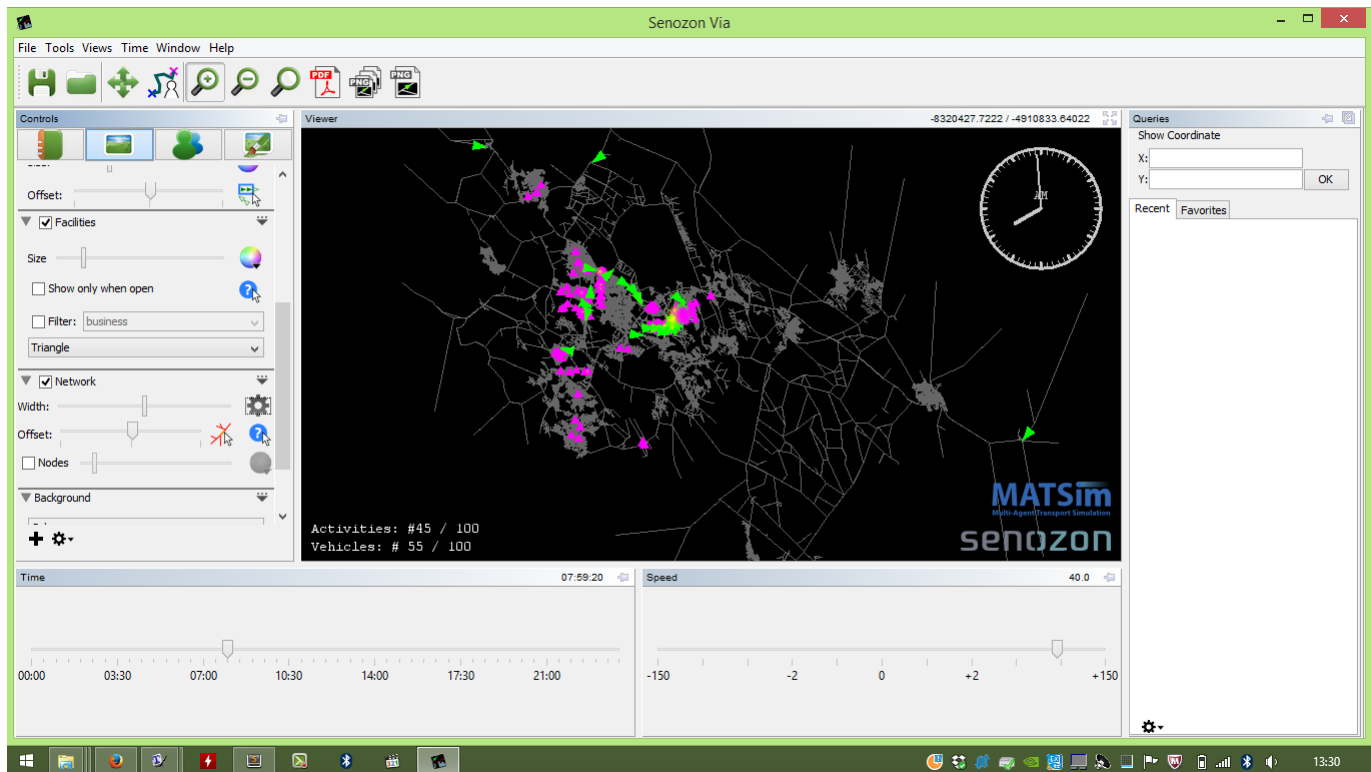


Figura 4.2: Captura de tela da representação gráfica da simulação no VIA-SENOZON

## 4.2 Análise dos Resultados

Conforme os histogramas do Anexo A, é perceptível a utilização otimizada das vias, pois o MATSim melhora altera o fluxo de agentes nas vias a cada iteração, aprendendo sobre os horários de pouco tráfego através do módulo de controle descrito na Seção 2.5.1. O que comprova essa afirmação é a observação do crescimento de agentes que assumem o estado *en route* e *arrival*, juntamente com um decrescimento da quantidade de agentes que assumem o estado *departure*.

Com relação a quantidade de agentes que assumem o estado *arrival*, é pequena pois os alunos estão possuem horários distintos em suas agendas, e é um bom indicador para as questões de caronas, levando em consideração que dependendo da carona algumas horas serão desperdiçadas chegando mais cedo no compromisso agendado. Vale ressaltar que na Figura A.20, a quantidade de agentes no estado *departure* tem um decrescimento em comparação com os demais histogramas do Anexo B, apenas por causa da fuga de horários de pico, os agentes assumem outros horários de maneira que o máximo possível consiga alcançar a velocidade máxima da via e dessa forma otimizar o fluxo do tráfego. Por não

ter muitas restrições nas vias, como pardais, pedestres, sinais de trânsito e o fluxo de outros automóveis, a simulação se tornou utópica. Para que o comportamento se aproxime do quadro real é necessário a inserção de outros agentes.

Sobre a quantidade de iterações da simulação, quanto maior for a quantidade, maiores serão as variações desses três tipos de estado que cada agente pode assumir durante a simulação. A quantidade de iterações realizadas em uma simulação é influenciada pela quantidade de agentes que fazem parte da mesma [26]. Vale ressaltar que todos os agentes foram caracterizados como usuários de automóveis particulares. Outro ponto interessante é a análise dos histogramas, com relação ao impacto do tráfego, pois toda vez que um agente entra no estado *en route*, significa que ele está em uma via, logo podemos prever picos de alunos dirigindo.

Por fim, é importante citar que a variação mostrada nos gráficos não afeta a parte interessante para o trabalho. Assim como no cotidiano temos diversas situações durante o deslocamento de um lugar a outro, a simulação retrata os agentes em seus deslocamentos, tentando otimizar seus caminhos e tempos, e mantém os pontos iniciais e finais, com seus tempos o mais próximo possível do que é proposto dentro dos planos em *plans.xml.gz*. Com o arquivo *events.xml.gz* conseguiremos os pontos iniciais e finais com horários de cada agente, e assim, fazer as ofertas de caronas, além de propor melhores rotas para os alunos.

## Capítulo 5

# Conclusões e Trabalhos Futuros

Com a análise dos histogramas e dos arquivos de saída do MATSim, concluiu-se que o framework pode ser utilizado no desenvolvimento do módulo de análise dos dados de entrada do sistema do projeto *Carona Solidária UnB*, uma vez que tenha demonstrado através dos resultados que é capaz de simular os deslocamentos dos alunos, mantendo as origens e destinos, além de horário das agendas disponibilizadas pela equipe do projeto *Carona Solidária UnB*. Com o arquivo *events.xml.gz*, que possui os deslocamentos de cada aluno, um sistema pode trabalhar as ofertas de caronas automaticamente por meio de padrões pré-estabelecidos pelos usuários do sistemas.

A adaptação do arquivo *events.xml.gz* para outros sistemas não é uma problemática por causa do arquivo de interpretação dos dados (na extensão DTD) do MATSim, tornando possível a utilização desse arquivo no sistema já vigente. Com os resultados da simulação do MATSim, podemos criar módulos de suporte que façam o controle de ofertas de caronas para os escritos no projeto *Carona Solidária UnB*, com equidade entre eles, através de programas de pontos, dando vantagens para quem deu mais caronas, por exemplo.

Uma possibilidade para a simulação é prever o impacto que o projeto *Carona Solidária UnB* terá ao ser implantado, prevendo diminuição no fluxo de carros nos campus e consequentemente no Distrito Federal e entorno. E utilizar esse relatório para conscientizar toda a comunidade acadêmica para a importância da implantação do Plano Diretor de Mobilidade Urbana Sustentável da Universidade de Brasília. Além de tentar uma possível parceria com redes de postos de combustível de todo o Distrito Federal, visando desse modo, vantagens para os participantes do programa como desconto no abastecimento de combustível. Outra aplicação para o relatório da simulação do MATSim, é a descoberta dos requisitos que o sistema de caronas necessita, para uma atualização ou reformulação do sistema.

O uso de poder de processamento que o MATSim necessita é algo a ser considerado, tendo em vista que o projeto deve funcionar em uma plataforma que permita a comunicação entre caroneiros e caronistas em tempo real. Os tempos de resposta para oferta de carona não podem demorar muito, ao ponto das pessoas venham a perder tempo esperando resultados de uma simulação. Como o tratamento não pode ser feito antes do cadastramento de todo o corpo acadêmico e durante todo semestre poderá ter alterações nas agendas, não existe outra saída além de manter um padrão de velocidade de processamento sempre alto. Outra possibilidade é otimizar os tempos de resposta tanto da

simulação como da interação entre o framework do MATSim e a do sistema do projeto *Carona Solidária UnB*. Através de otimizações no código fonte, por meio da adaptação para uma nova linguagem ou novos métodos de programação, pois existe uma infinidade de métodos de programação de sistemas e a cada dia novos sistemas são desenvolvidos [15], uma adaptação pode ser custosa e mesmo assim viável.

# Referências

- [1] Dario José Aloise and JSD Cruz. Teoria dos grafos e aplicações. <http://xa.yimg.com/kq/groups/23212136/1217905042/name/tg-aloise.pdf>. Acessado 2014-09-10. [x](#), [7](#), [8](#), [9](#), [10](#)
- [2] Paulo Blikstein, Dor Abrahamson, and Uri Wilensky. Netlogo: Where we are, where we're going. In *Proceedings of the annual meeting of Interaction Design and Children, Press*, 2005. [23](#)
- [3] Tony Buzan. *Mapas mentais e sua elaboração*. Editora Cultrix, São Paulo, Brasil, 1st edition, 2005. [7](#)
- [4] Lúcia Calijuri and S ROHM. Sistema de informação geográfica. *Viçosa: UFV*, 1993. [5](#), [6](#)
- [5] Vânia Barcellos Gouvêa Campos. Uma visão da mobilidade urbana sustentável. *Revista dos Transportes Públicos*, 2:99–106, 2006. [4](#)
- [6] David Charypar and Kai Nagel. Q-learning for flexible learning of daily activity plans. *Transportation Research Record: Journal of the Transportation Research Board*, 1935(1):163–169, 2005. [12](#), [29](#)
- [7] Nicholas R Chrisman. *Charting the unknown : how computer mapping at Harvard became GIS*. ESRI Press, Independent Publishers Group, Redlands, Calif., 1st edition, 2006. [5](#), [6](#)
- [8] Nick Collier. Repast: An extensible framework for agent simulation. *The University of Chicago's Social Science Research*, 36:2003, 2003. [24](#)
- [9] Dell COPYRIGHT. Dell. <http://http://www.dell.com/>. Acessado 2015-05-10. [60](#)
- [10] Marcela da Silva Costa. Mobilidade urbana sustentável: um estudo comparativo e as bases de um sistema de gestão para brasil e portugal. Master's thesis, Dissertação (Mestrado)- Universidade de São Paulo, São Paulo, Brasil, 2003. [4](#)
- [11] Keila Rodrigues de Moura and SERGIO AUGUSTO RODRIGUES. Carpooling como uma alternativa para a melhoria do trânsito: Aceitabilidade e características dos possíveis usuários. *Tekhne e Logos*, 4(3):154–168, 2013. [4](#), [30](#)
- [12] Programa de Pós Graduação em Transportes. Programa de pós graduação em transportes - unb. <http://www.transportes.unb.br/>. Acessado 2014-10-18. [1](#)



- [13] DETRAN-DF. Departamento de trânsito do distrito federal. <http://www.detran.df.gov.br>. Acessado 2014-09-30. 1
- [14] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959. 19, 29
- [15] Lee D Erman, Frederick Hayes-Roth, Victor R Lesser, and D Raj Reddy. The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys (CSUR)*, 12(2):213–253, 1980. 10, 11, 36
- [16] Ilya Grigoryev. *AnyLogic 6 in three days: a quick course in simulation modeling*, volume 1 of 1. AnyLogic North America, Chicago, 1 edition, 10 2012. x, 20, 21, 22
- [17] PTV Group. Ptv vissim. <http://vision-traffic.ptvgroup.com/es/productos/ptv-vissim/>. Acessado 2015-07-24. 25, 26, 27
- [18] PTV Group. Ptv visum. <http://vision-traffic.ptvgroup.com/es/productos/ptv-visum/>. Acessado 2015-07-24. 25, 26
- [19] Jomi Fred Hübner and Jaime Simão Sichman. Organização de sistemas multiagentes. *III Jornada de Mini-Cursos de Inteligência Artificial (JAIA 03)*, 8:247–296, 2003. 10, 11, 25
- [20] M. J Jennings, N. R. e Wooldridge. Agent technology: foundations, applications, and markets. *Springer Verlag*, 1998. 11
- [21] Ricardo Mendes Júnior and Rafael Fusco. Projeto carona solidária na ufpr. *Revista Latino-Americana de Inovação e Engenharia de Produção*, 1(1), 2013. 30
- [22] Argonne National Laboratory. The repast suite. <http://repast.sourceforge.net>. Acessado 2015-06-18. 25
- [23] David J Maguire. An overview and definition of gis. *Geographical information systems: Principles and applications*, 1:9–20, 1991. 6
- [24] Erick de Santana Mello. Mobilidade urbana sustentável em projetos estruturantes: análise urbanística e ambiental do corredor de transporte da avenida bernardo vieira natal/rn. Master’s thesis, Dissertação (Mestrado)- Universidade Federal do Rio Grande do Norte, Rio Grande do Norte, Brasil, 2008. 4
- [25] Daniel MG Morais and Luciano A Digiampietri. Simulação de transportes sobre trilhos usando autômatos celulares para os trens e agentes inteligentes para os usuários. Simpósio Brasileiro de Sistemas de Informação, 5 2014. Londrina. X Simpósio Brasileiro de Sistemas de Informação. 29
- [26] Kai Nagel. Matsim - multi-agent transport simulation. <http://www.matsim.org/index.php>. Acessado 2014-09-30. x, 6, 7, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 28, 29, 31, 32, 34
- [27] Kai Nagel. *Cellular Automata Models for Transportation Applications*, volume 2493, pages 20–31. Springer Berlin Heidelberg, 2002. 2015-01-01. 7

- [28] Tijs Neutens, Tim Schwanen, and Harvey J Miller. Dealing with timing and synchronization in opportunities for joint activity participation. *Geographical analysis*, 42(3):245–266, 2010. 7
- [29] Alan Nicholson. Transport network reliability measurement and analysis. *Transportes*, 11(2), 2003. 29
- [30] reportagem de Hugo Costa em 20 de Março de 2015 no UnB Notícias. Da Secretaria de Comunicação da UnB. Carona para sustentabilidade. <http://www.unb.br/noticias/unbagencia/unbagencia.php?id=9317>. 2015-06-19. 30
- [31] reportagem de Pedro Alves e Thaís Carneiro em 23 Setembro 2013 no Campus online. Jornal-laboratório da FAC/UnB. Projeto carona solidária unb. <http://campus.fac.unb.br/arquivo/campus12014/universidade/item/2980-carona-solidaria>. 2015-06-19. 1
- [32] A.F. SILVEIRA. Rumo á mobilidade sustentável no campus da unb: Carona solidária. Monografia de Projeto Final, 2013. Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF. 29, 30
- [33] John Michael Thomson. *Great cities and their traffic*. Littlehampton Book Services Ltd, Faraday Close, Worthing, West Sussex BN13 3RB, United Kingdom, 1st edition, 1978. 1
- [34] Uri Wilensky. Netlogo - user manual. <http://ccl.northwestern.edu/netlogo/docs/>. Acessado 2014-10-21. 23
- [35] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):1–35, 1984. 7

# Apêndice A

## Histogramas da Simulação de Brasília

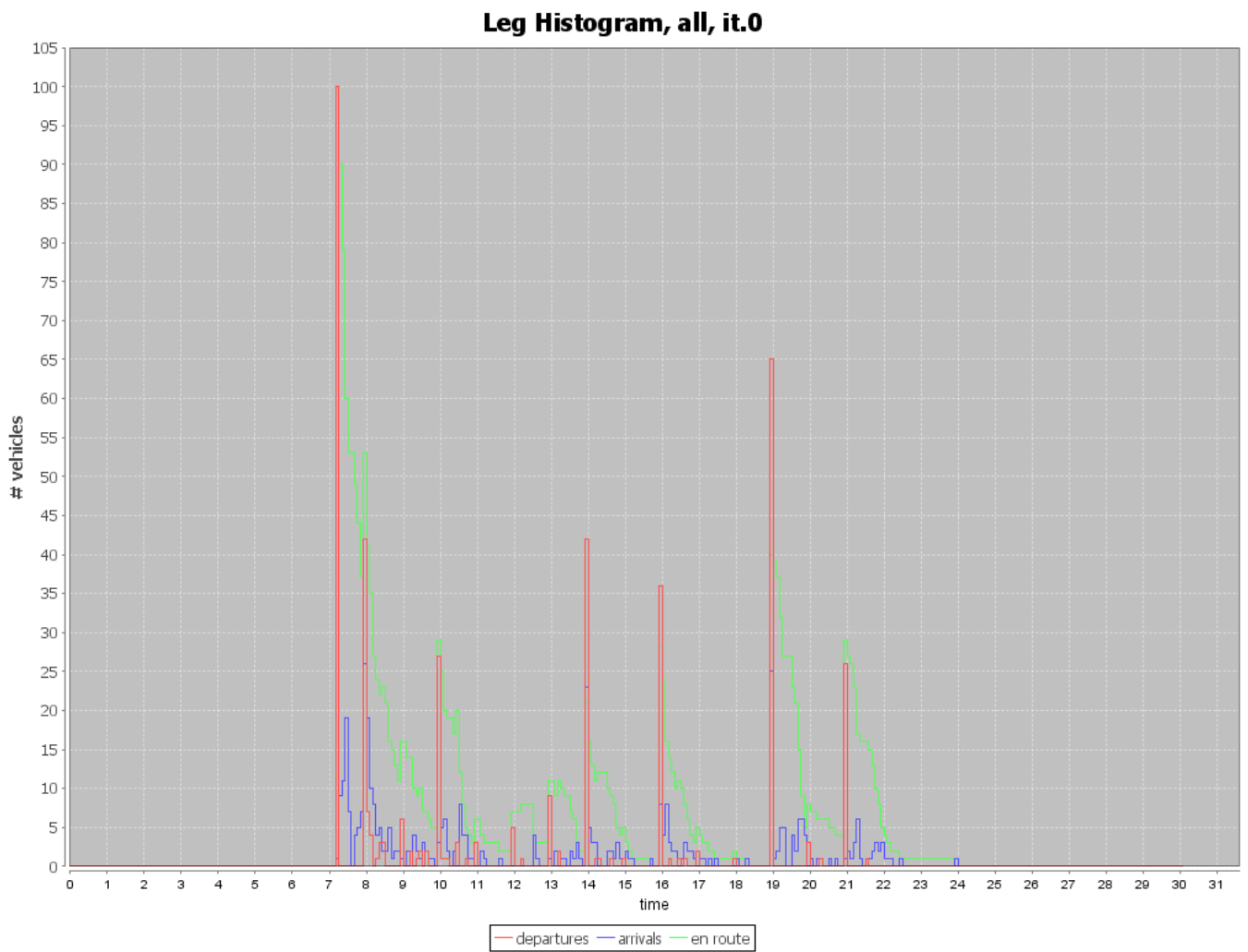


Figura A.1: Histograma da iteração 0

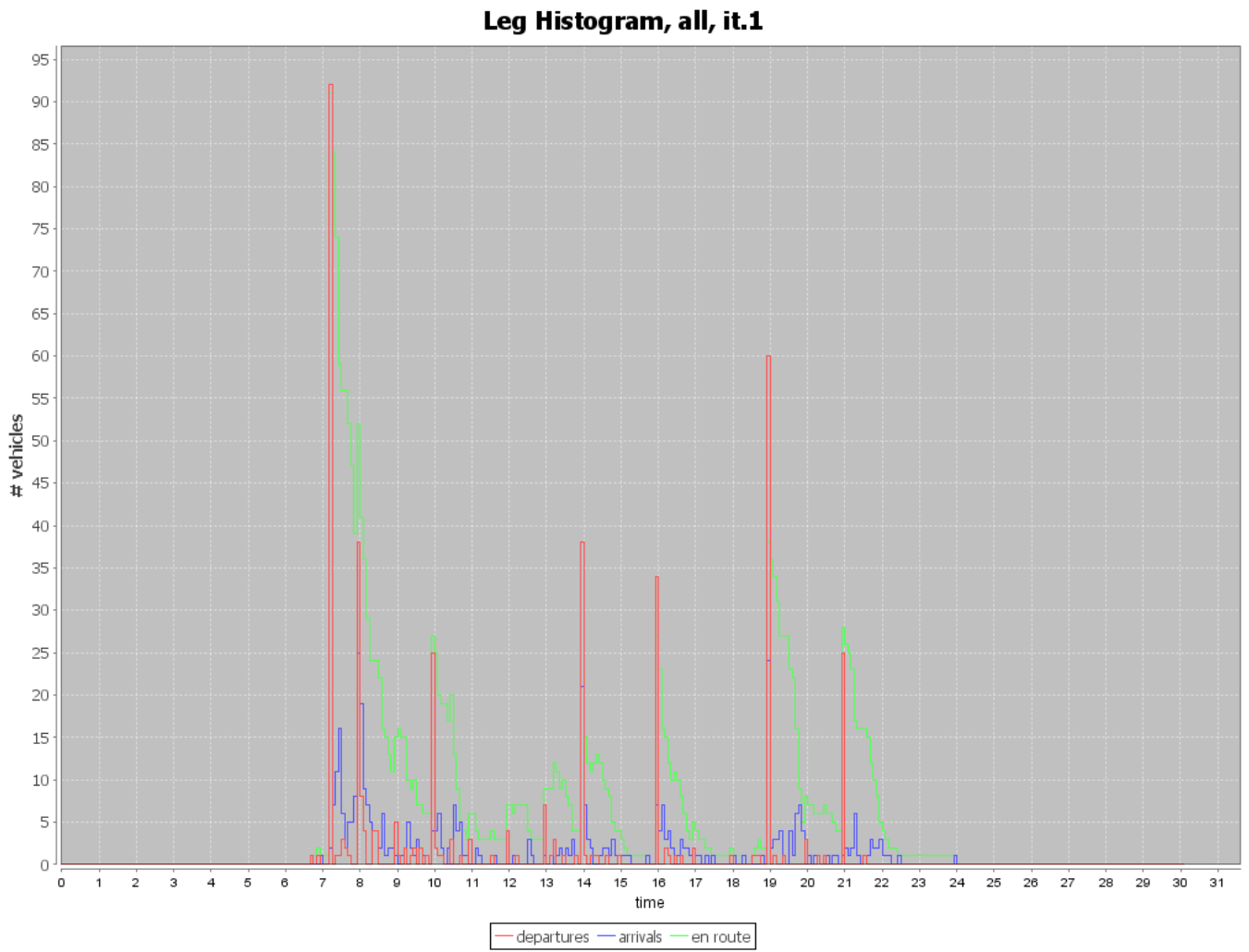


Figura A.2: Histograma da iteração 1

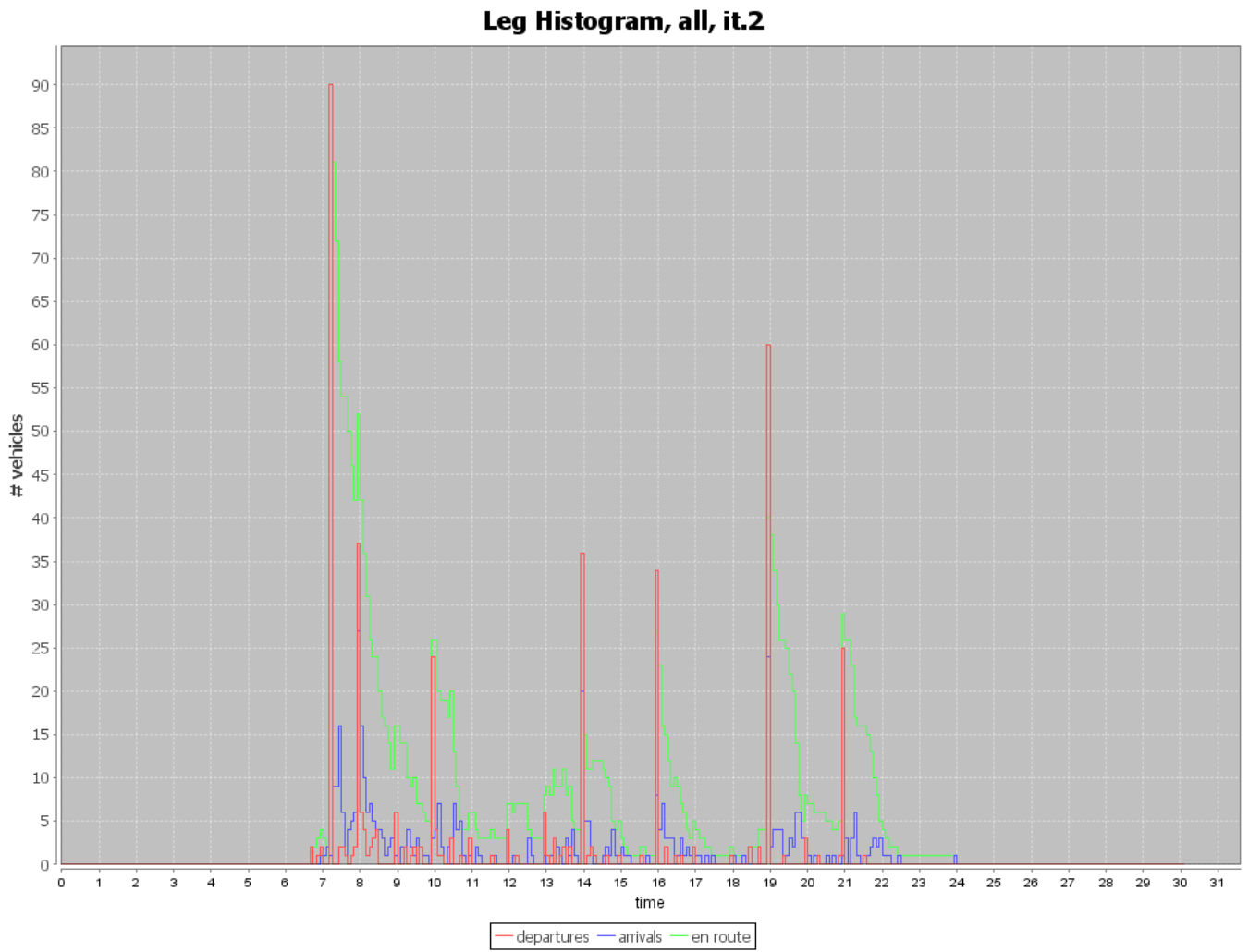


Figura A.3: Histograma da iteração 2

**Leg Histogram, all, it.3**

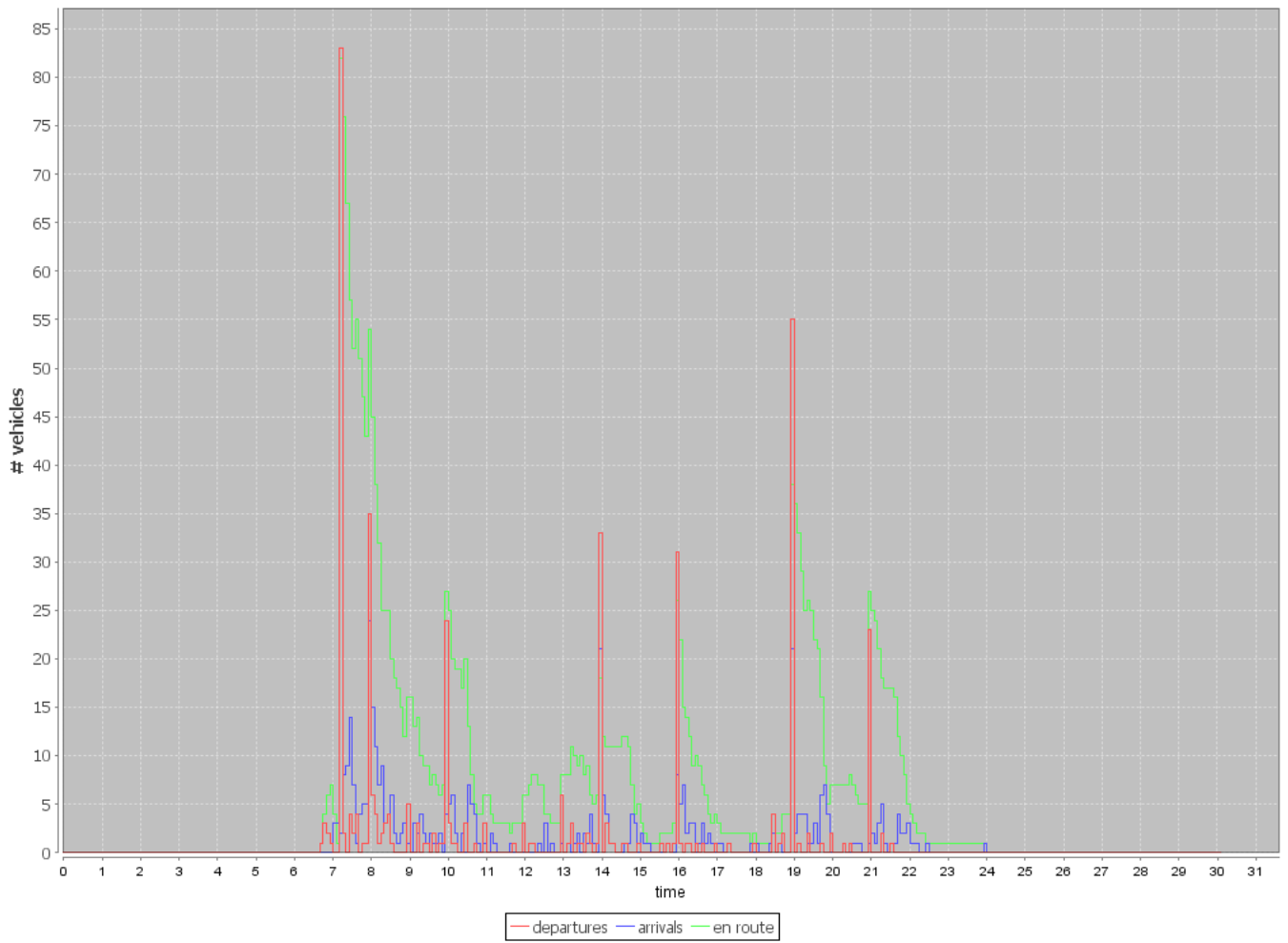


Figura A.4: Histograma da iteração 3

**Leg Histogram, all, it.4**

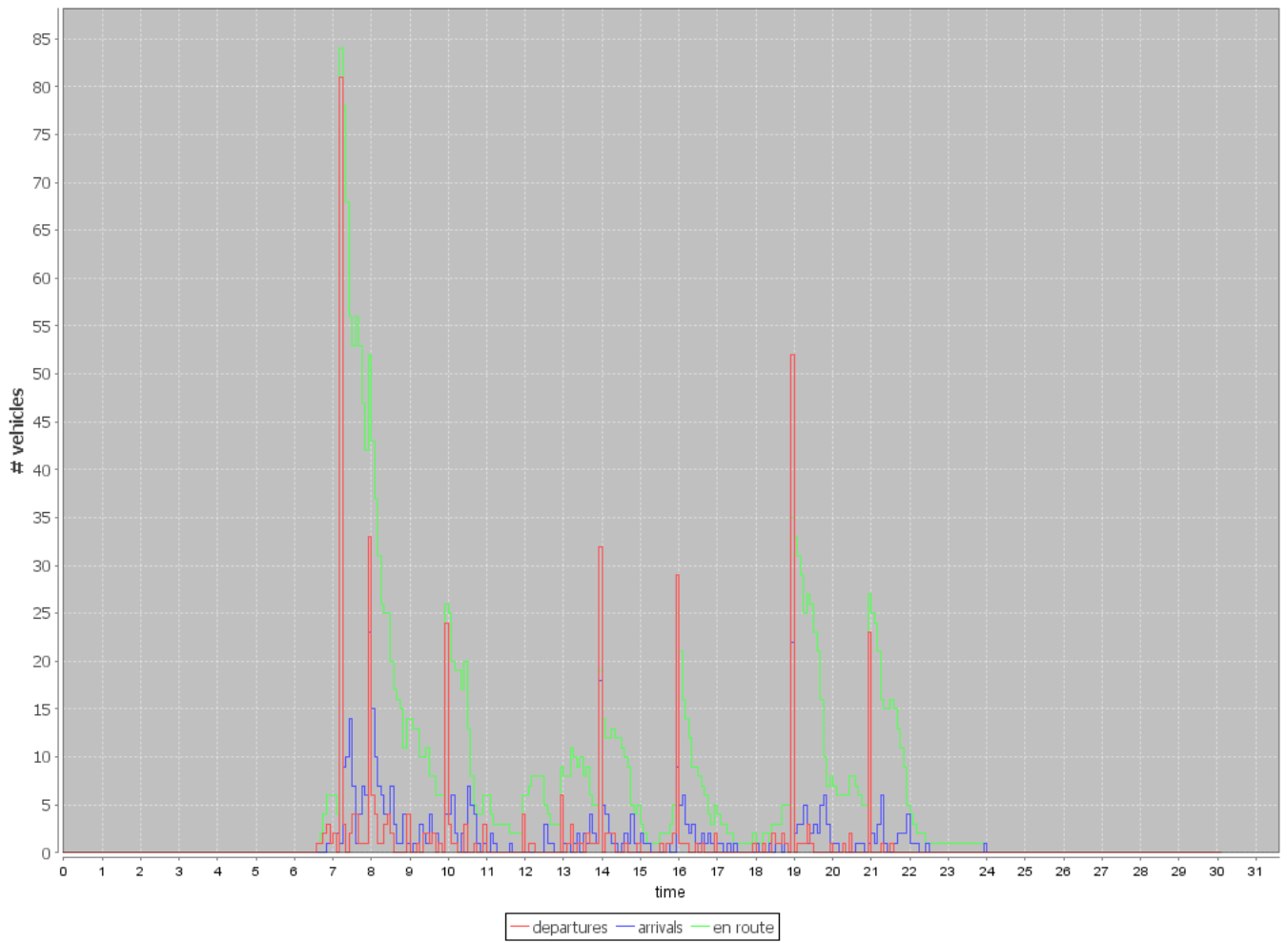


Figura A.5: Histograma da iteração 4

**Leg Histogram, all, it.5**

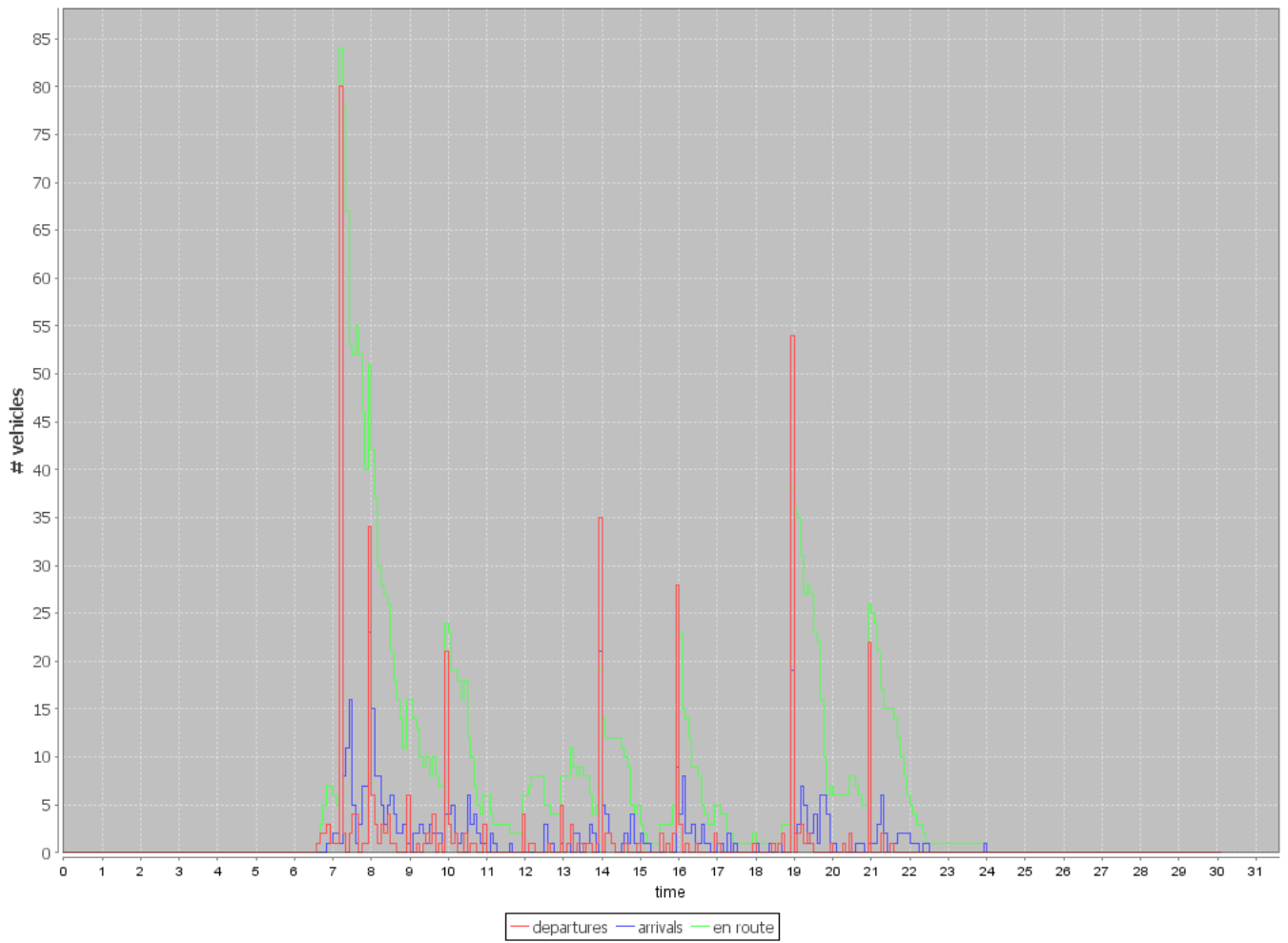


Figura A.6: Histograma da iteração 5



**Leg Histogram, all, it.6**

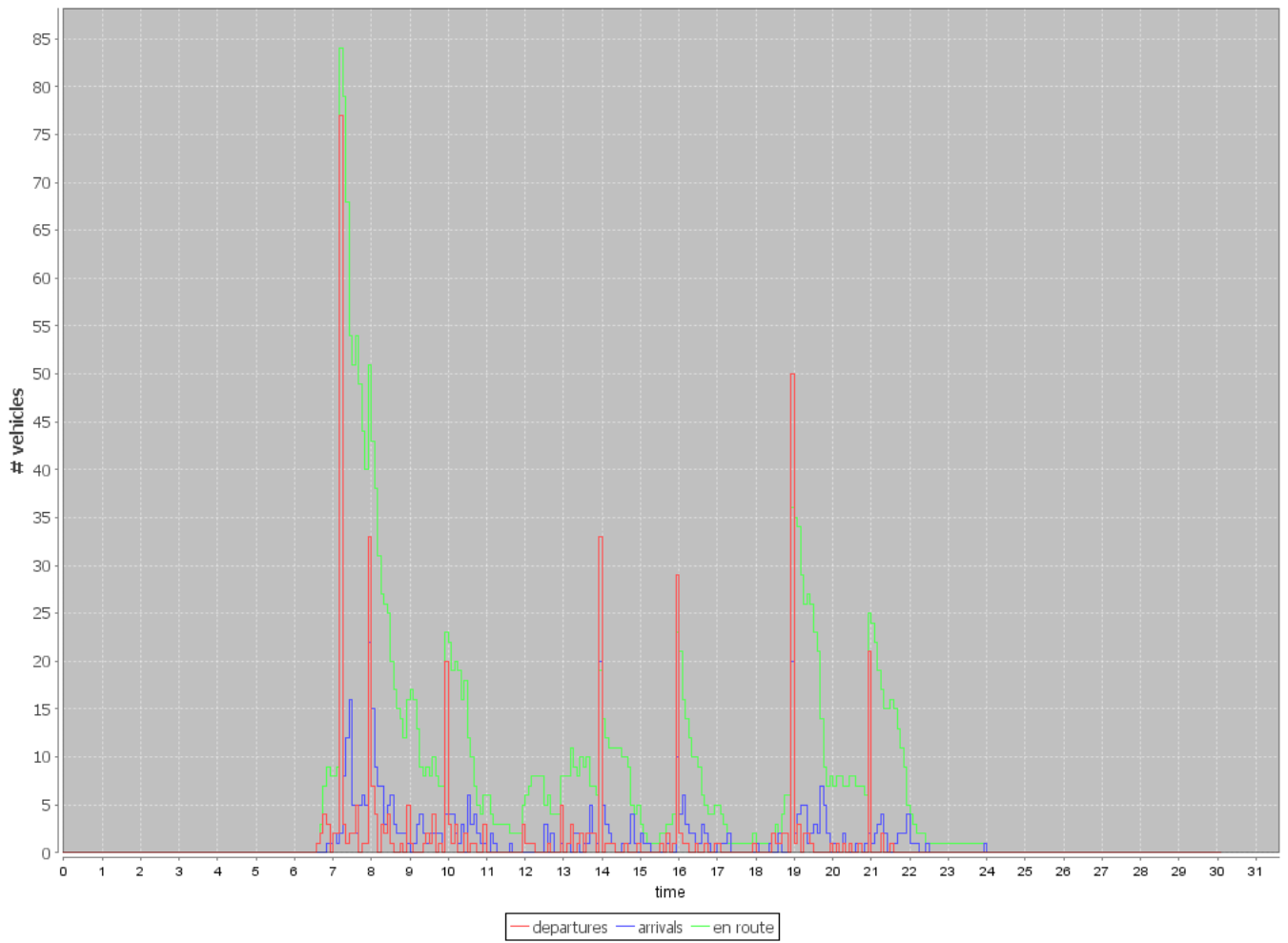


Figura A.7: Histograma da iteração 6

**Leg Histogram, all, it.7**

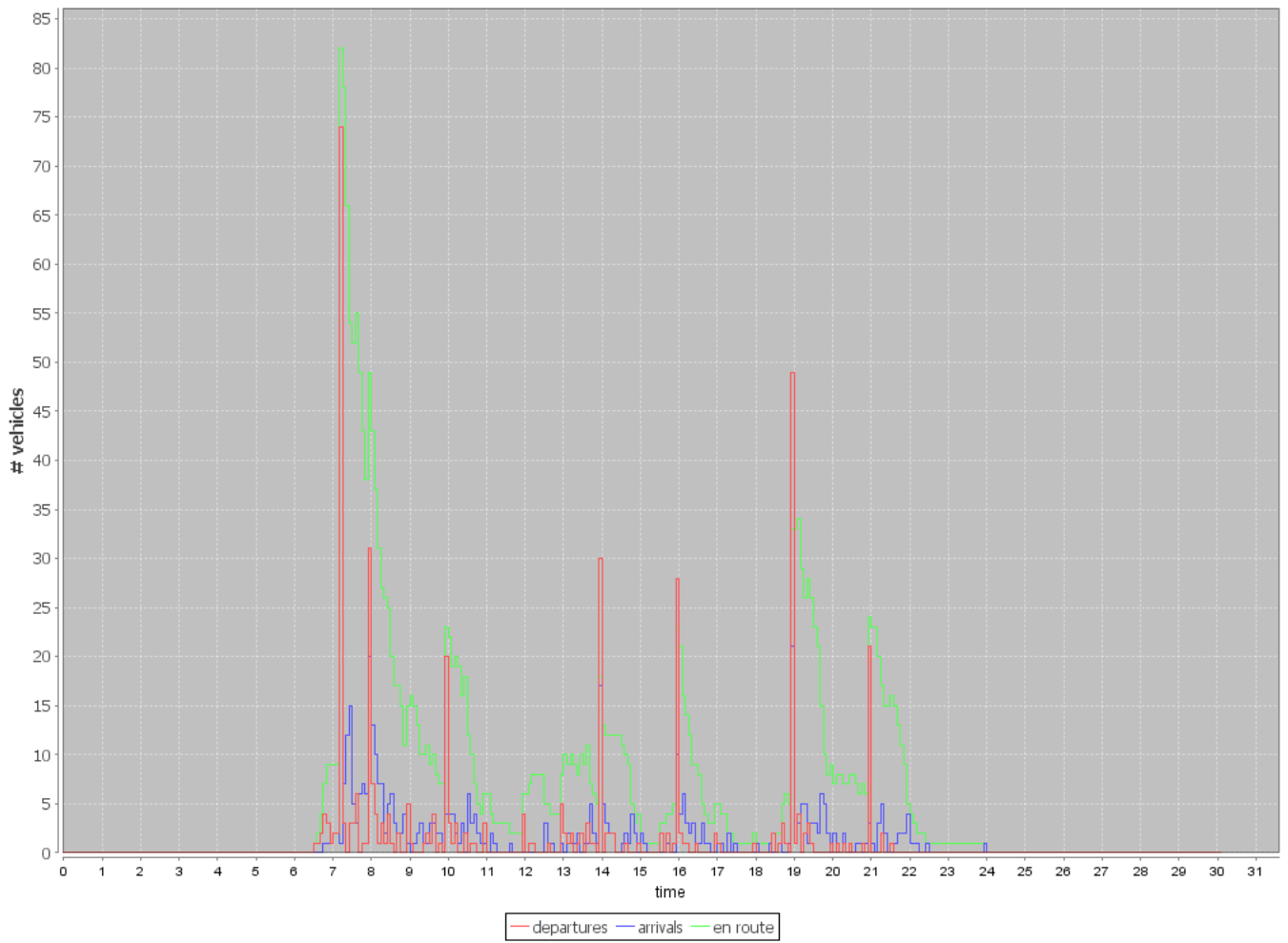


Figura A.8: Histograma da iteração 7

**Leg Histogram, all, it.8**

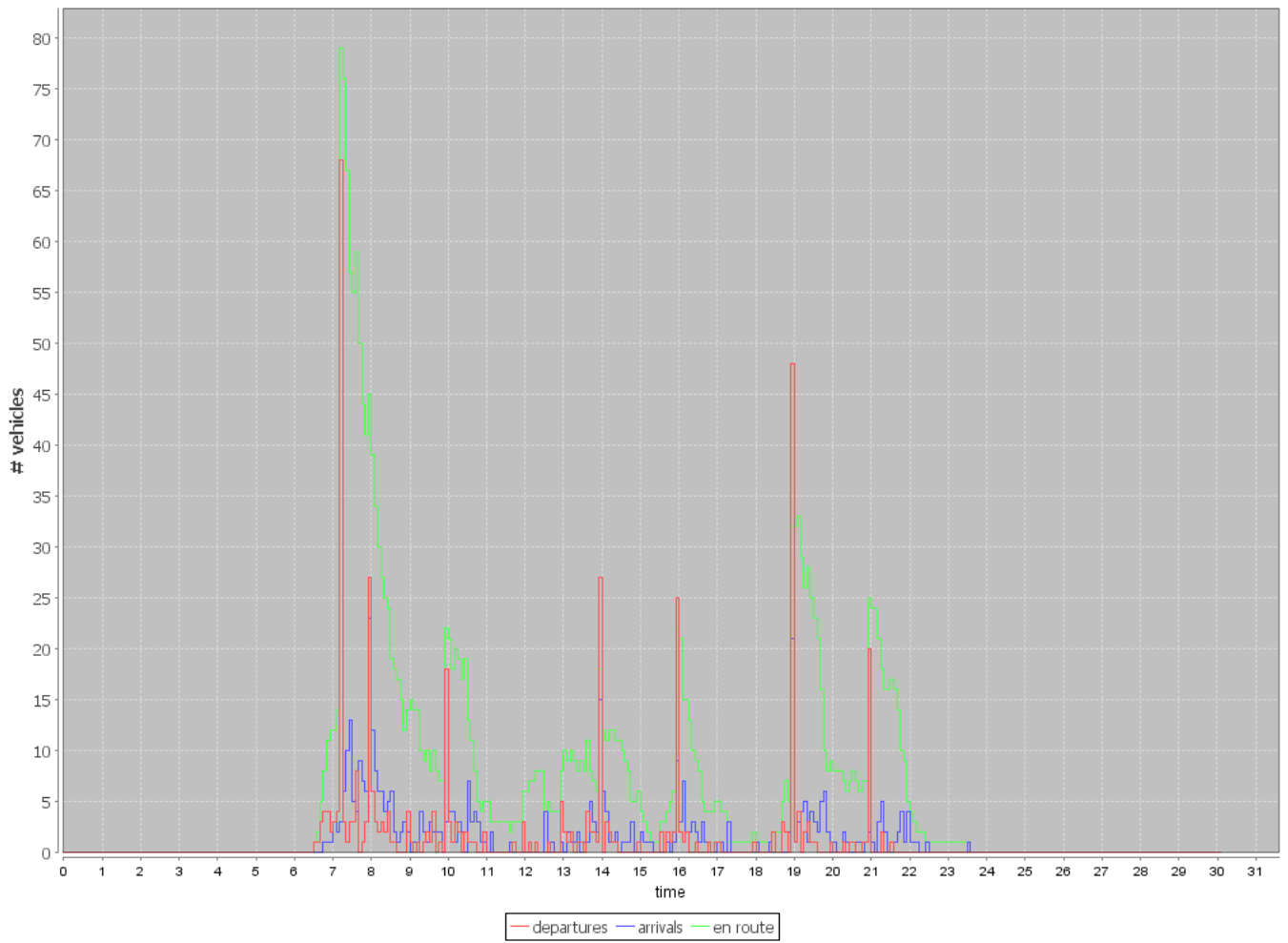


Figura A.9: Histograma da iteração 8

**Leg Histogram, all, it.9**

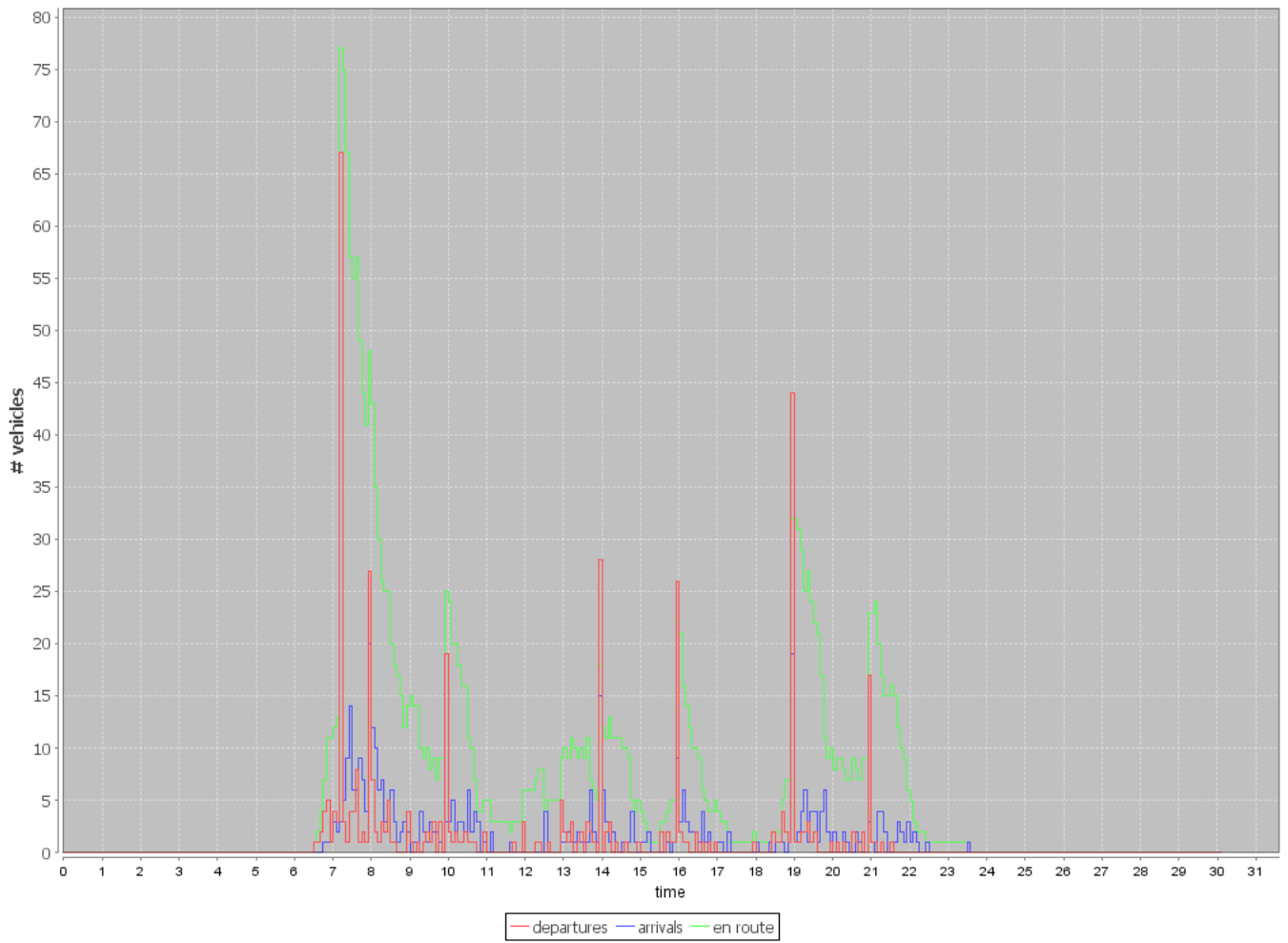


Figura A.10: Histograma da iteração 9

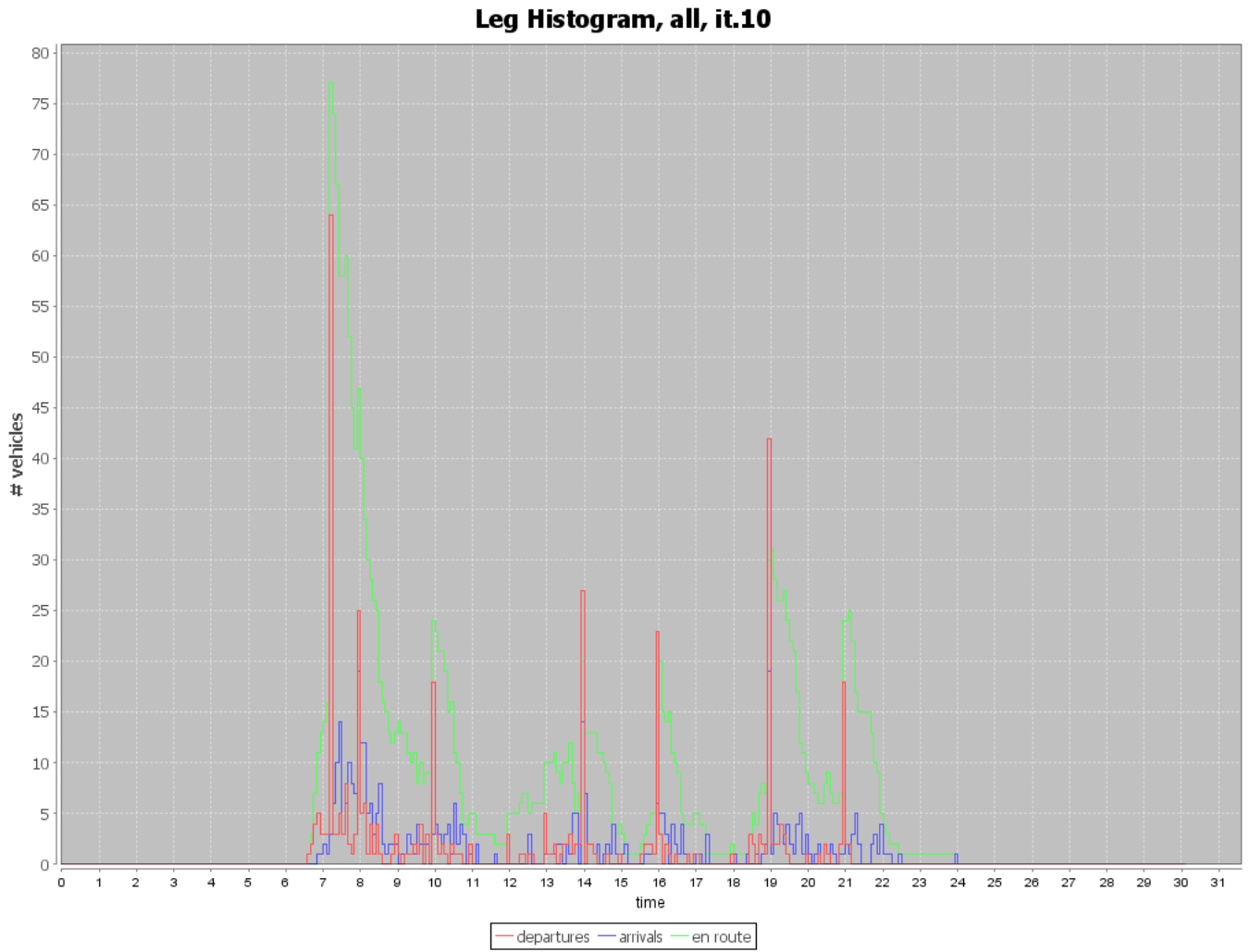


Figura A.11: Histograma da iteração 10

**Leg Histogram, all, it.11**

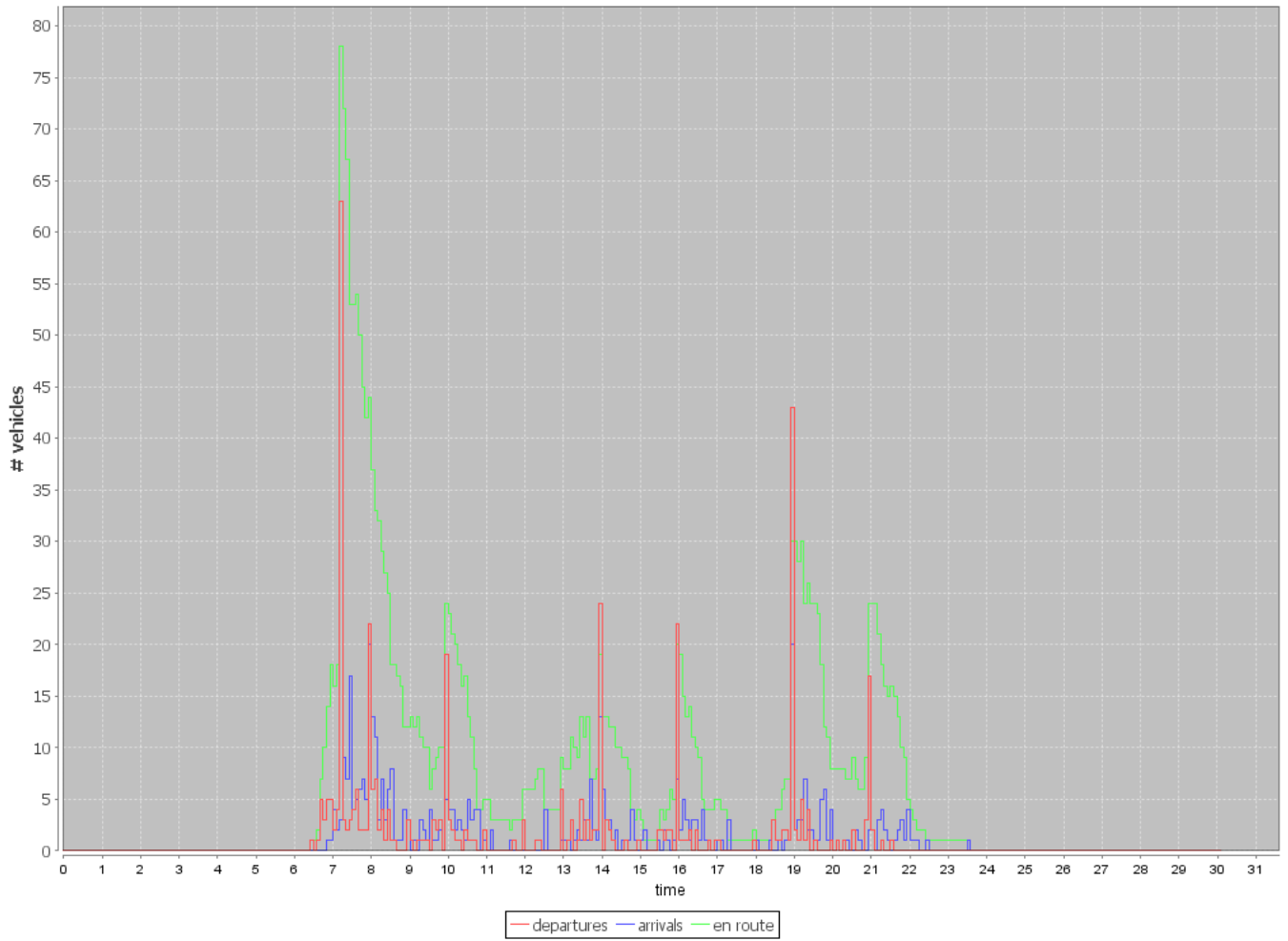


Figura A.12: Histograma da iteração 11

**Leg Histogram, all, it.12**

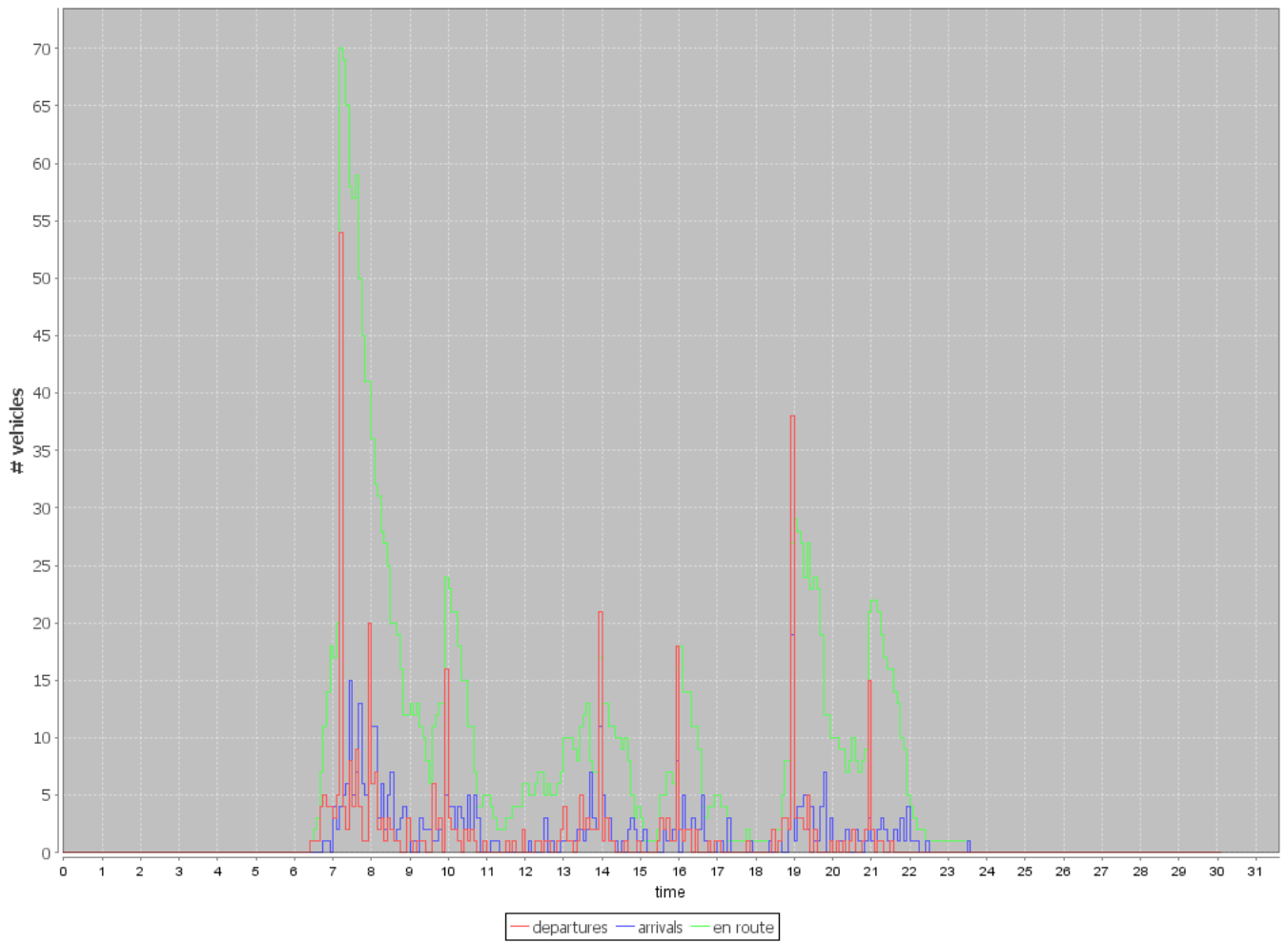


Figura A.13: Histograma da iteração 12

**Leg Histogram, all, it.13**

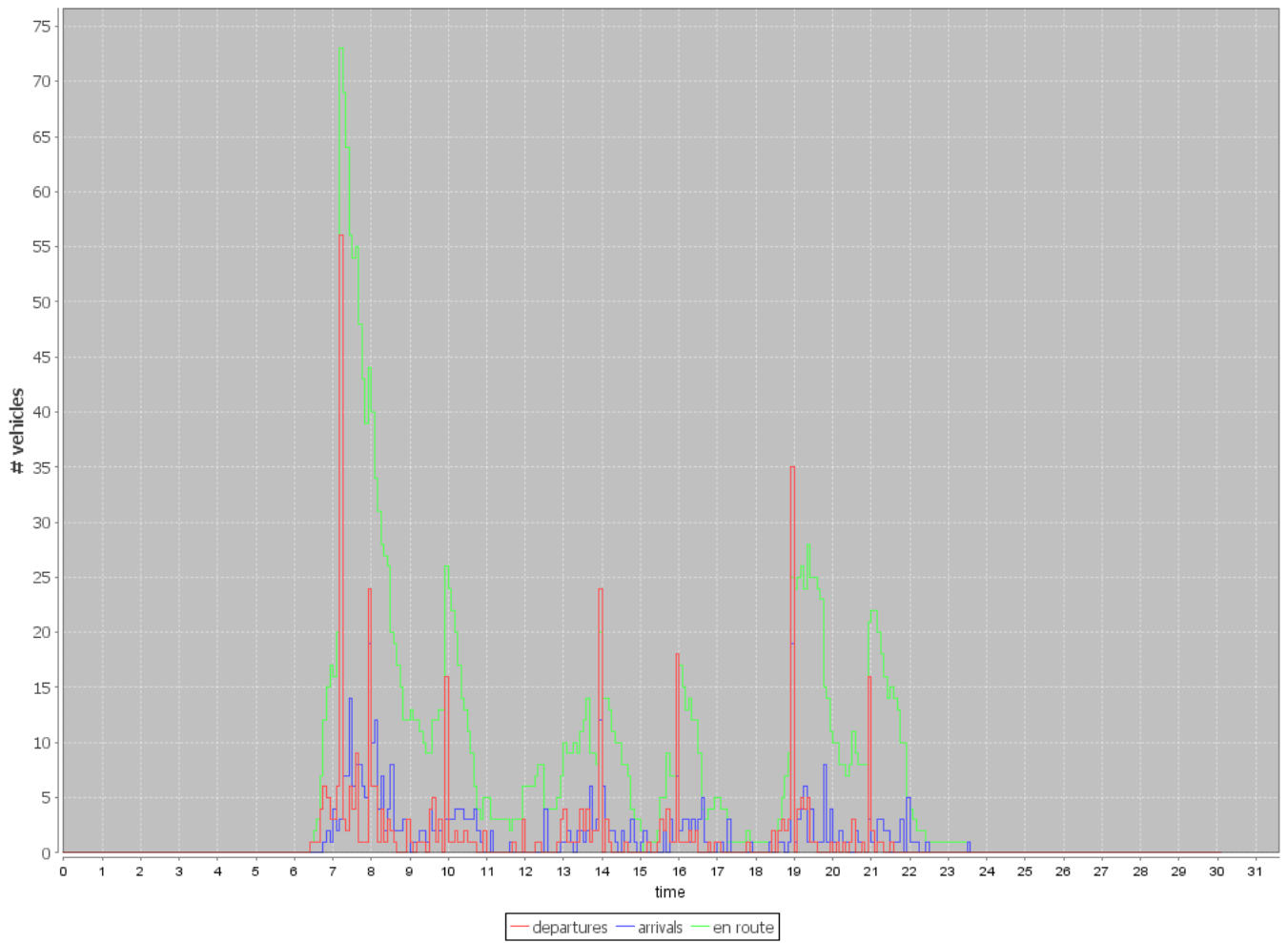


Figura A.14: Histograma da iteração 13



**Leg Histogram, all, it.14**

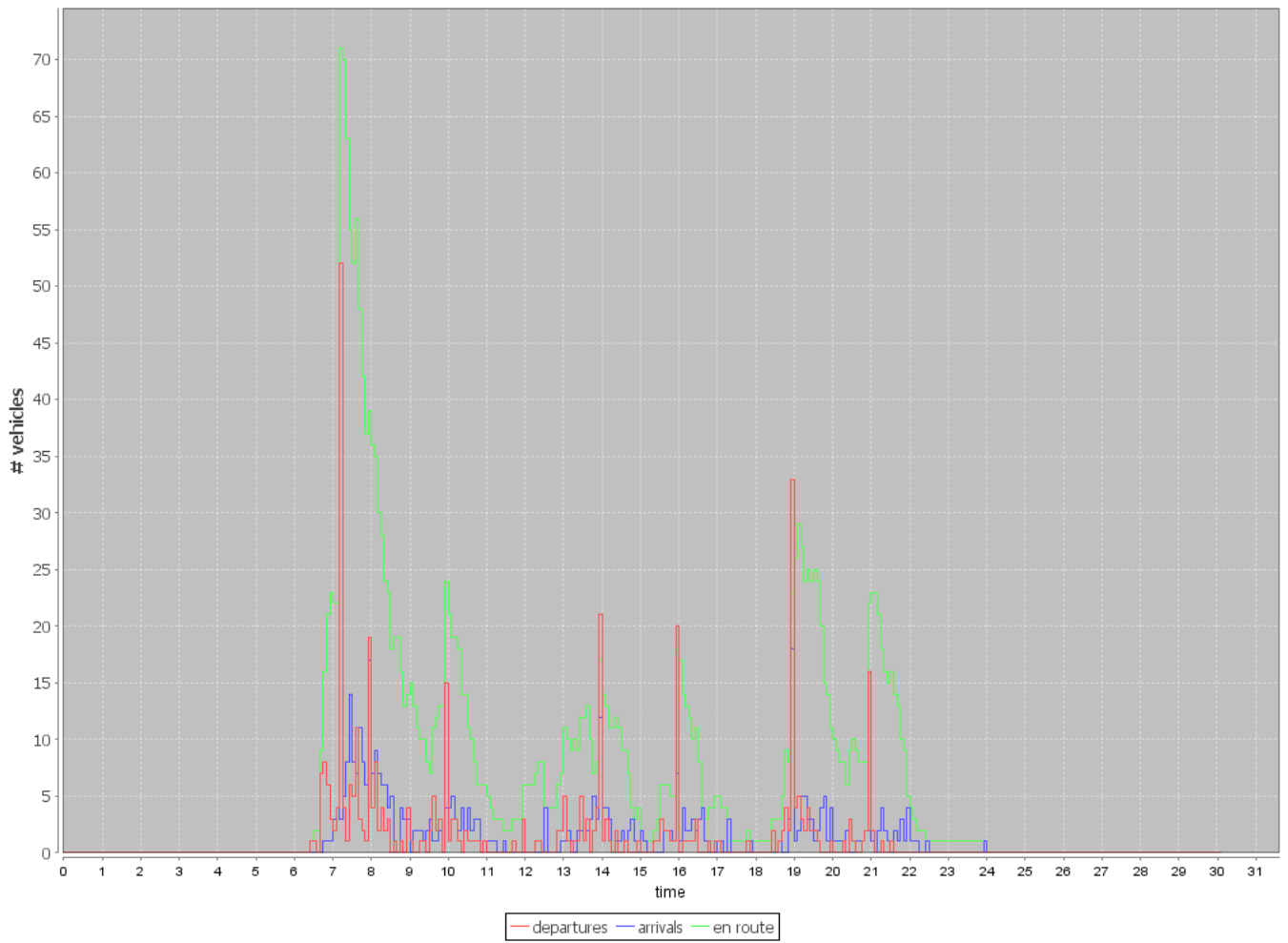


Figura A.15: Histograma da iteração 14

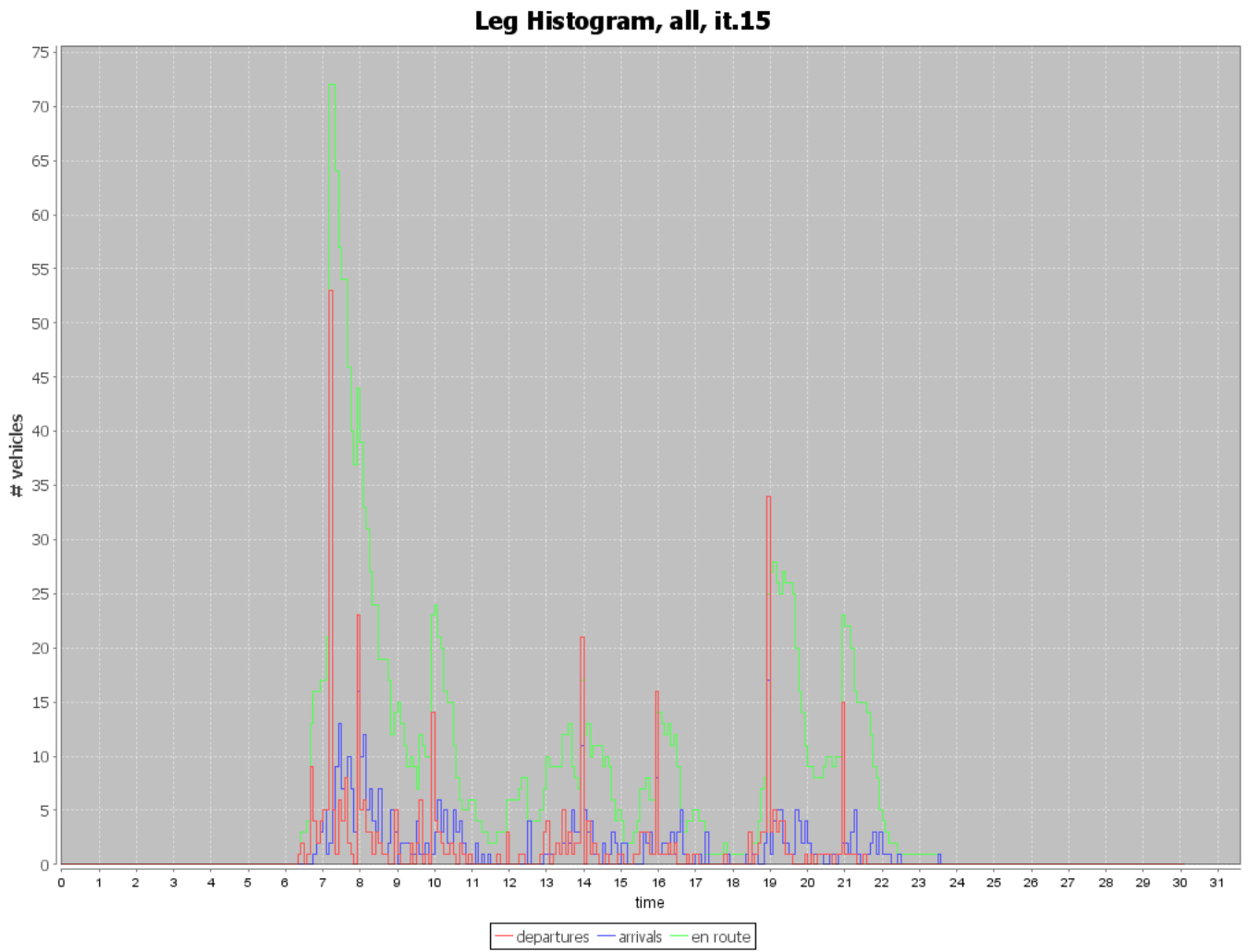


Figura A.16: Histograma da iteração 15

**Leg Histogram, all, it.16**

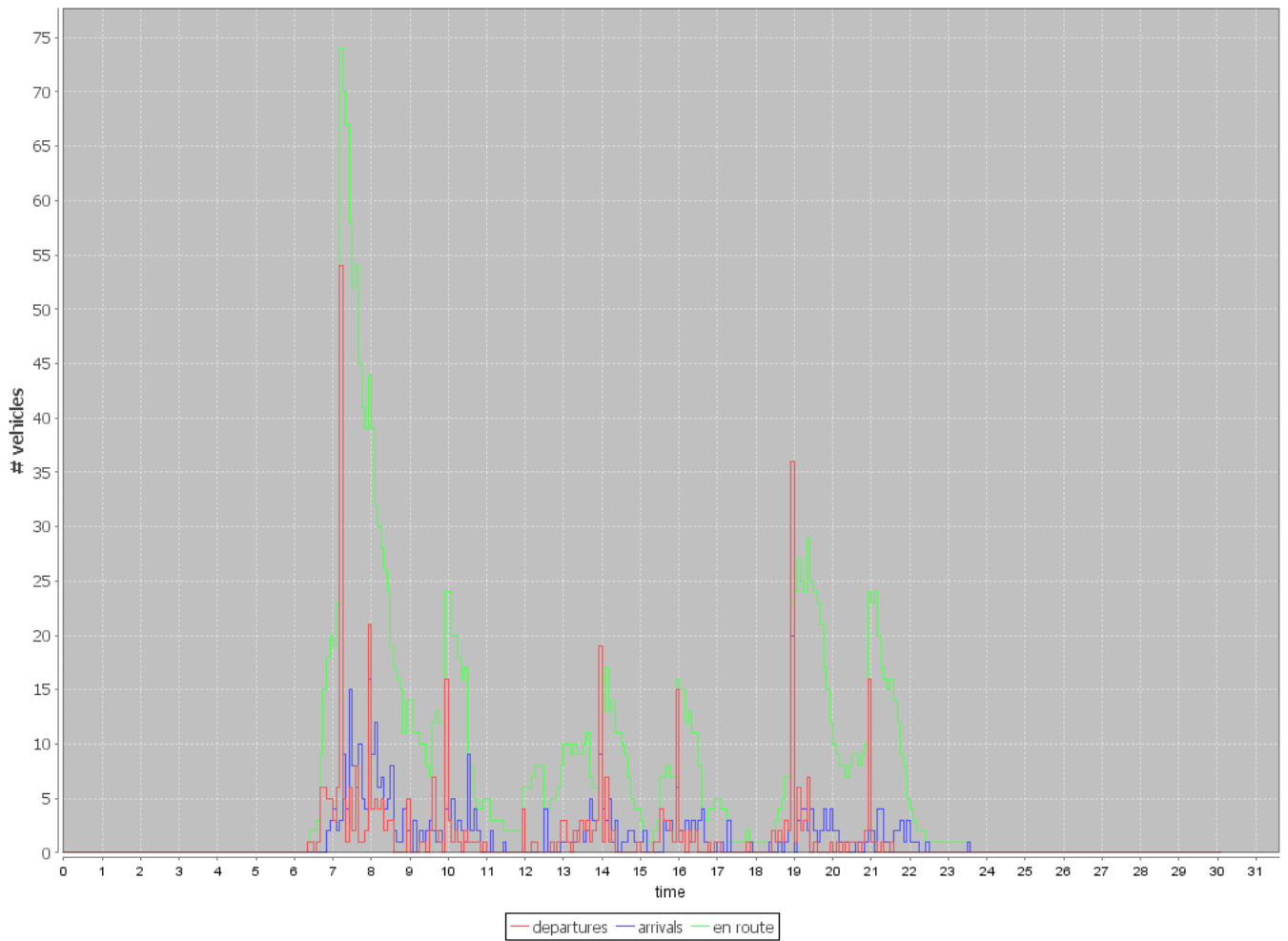


Figura A.17: Histograma da iteração 16

**Leg Histogram, all, it.17**

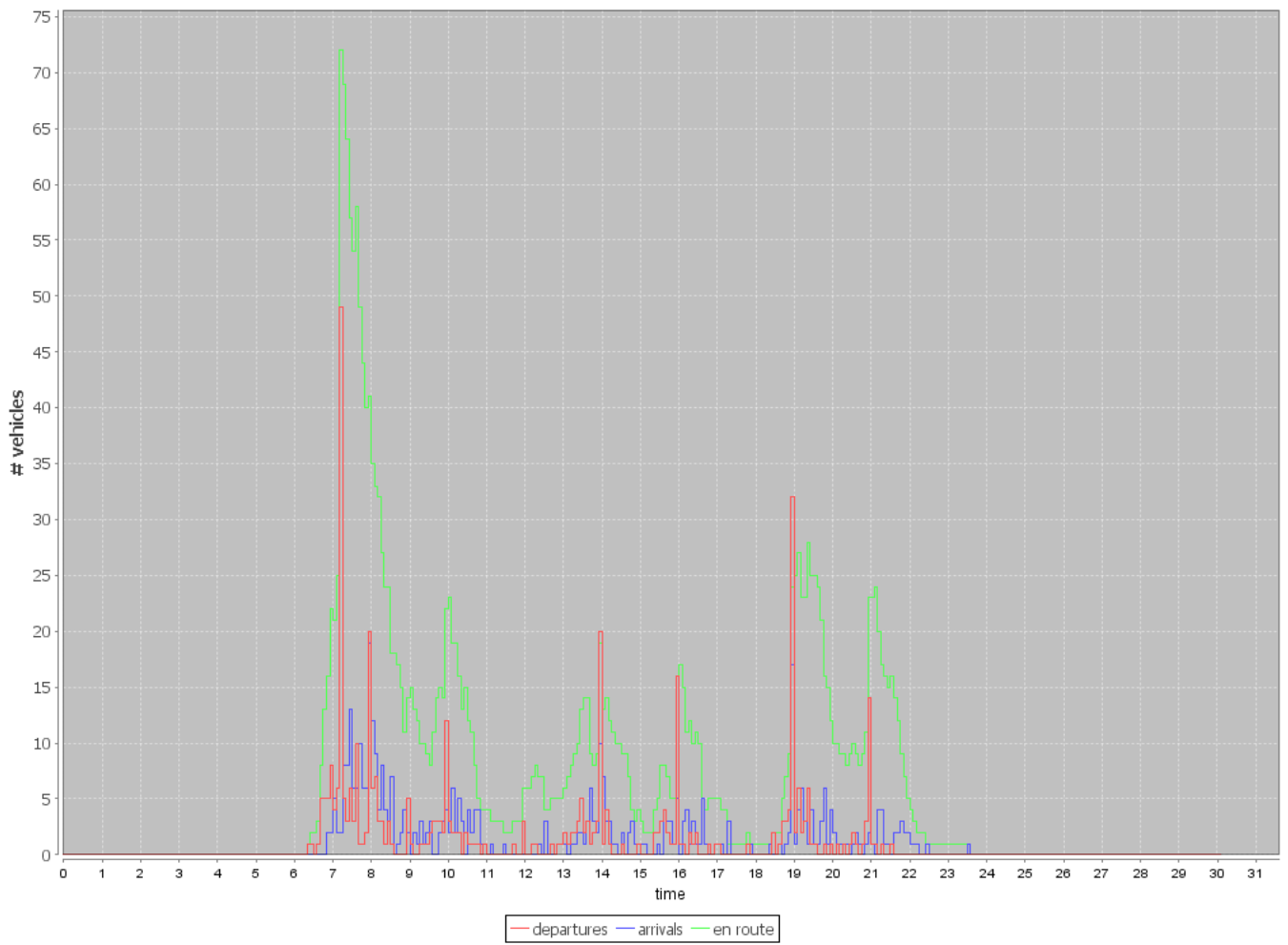


Figura A.18: Histograma da iteração 17

**Leg Histogram, all, it.18**

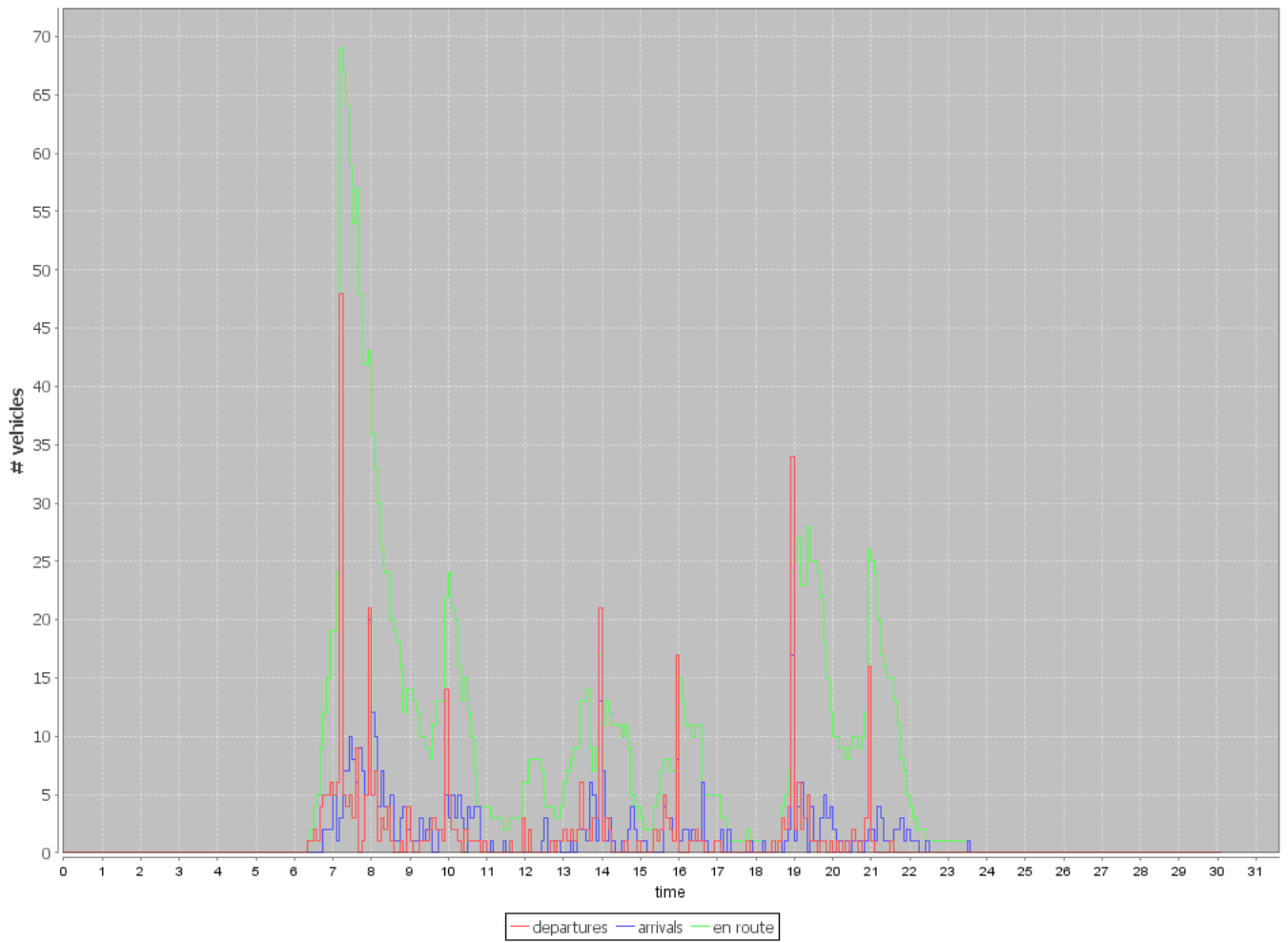


Figura A.19: Histograma da iteração 18

**Leg Histogram, all, it.19**

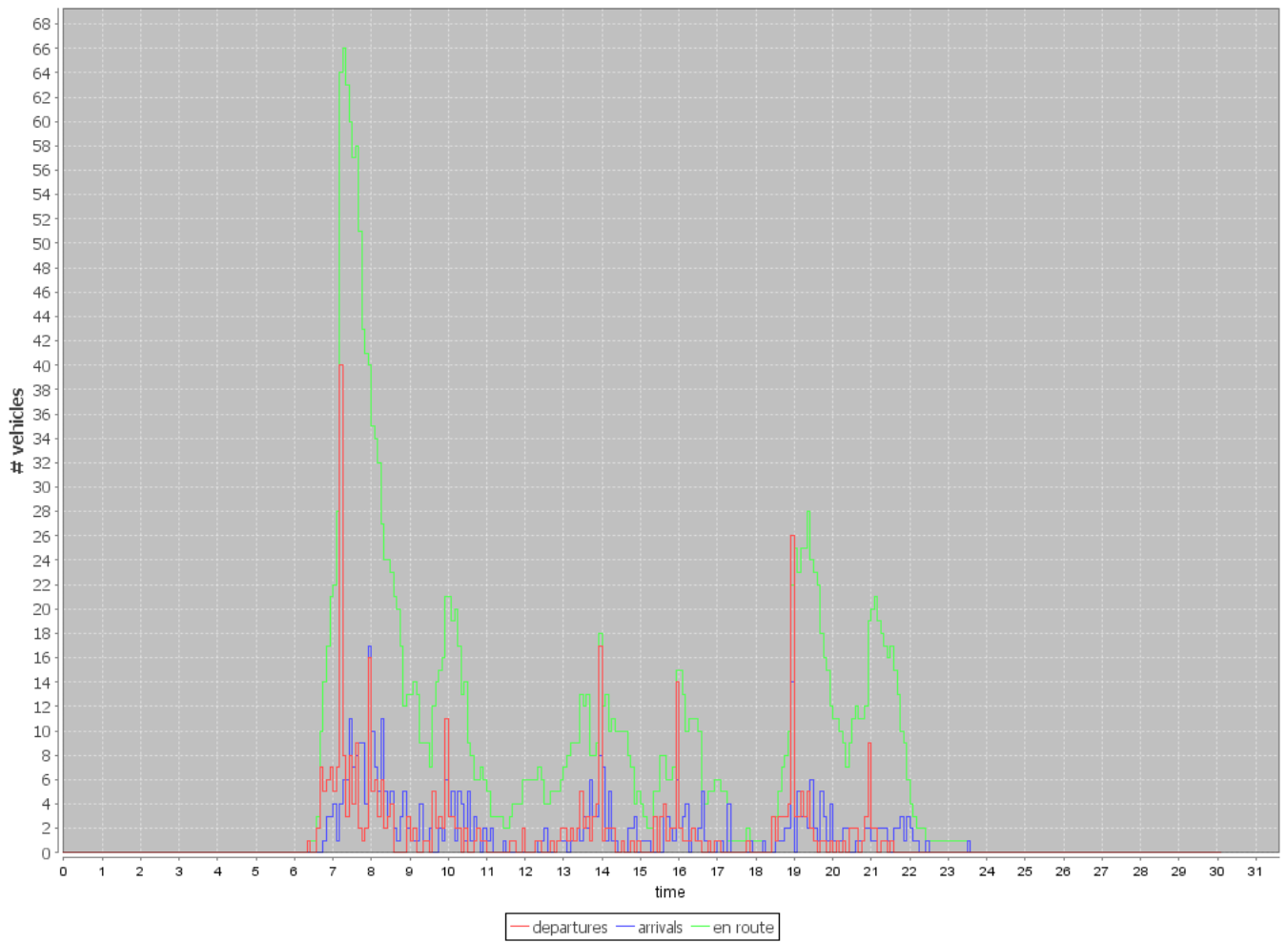


Figura A.20: Histograma da iteração 19

# Apêndice B

## Características do Vostro 5470



Figura B.1: Vostro 5470

Descrição da máquina [9]:

- Processador: i5-4200U 1.6 GHz (até 2.6 GHz com Turbo Boost)
- Memória: 4GB DDR3 1600MHz
- Vídeo: GeForce GT 740M, 64 bits, 2GB DDR3
- HD: 500GB, 5400 rpm
- M-sata: possui uma half m-sata

- Tela: 14", 1366 x 768
- Teclado: Padrão ABNT 2
- Leitor Biométrico
- Wireless a/b/g/n, bluetooth 4.0, rede 10/100/1000, web cam
- Conexões: HDMI, rede, 3 usb, leitor de cartão, 1 Porta combo
- Bateria: Bateria de 3 células e 51 WHr
- Dimensões: 33,7 cm (L) x 23,3 cm (P) x 19,3 (A)
- Espessura: 1,83 cm
- Peso: 1,62kg