

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA E  
INFORMÁTICA

WILLIAM HITOSHI TSUNODA MEIRA

**PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM  
ENTREGAS E COLETAS MISTAS E JANELAS DE TEMPO:  
APLICAÇÃO EM UMA EMPRESA DA REGIÃO  
METROPOLITANA DE CURITIBA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2013

WILLIAM HITOSHI TSUNODA MEIRA

**PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM  
ENTREGAS E COLETAS MISTAS E JANELAS DE TEMPO:  
APLICAÇÃO EM UMA EMPRESA DA REGIÃO  
METROPOLITANA DE CURITIBA**

Monografia apresentada à disciplina de Trabalho de Conclusão de Curso 2 do curso de Engenharia de Computação apresentado ao Departamento Acadêmico de Eletrônica e Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Engenheiro em Computação”.

Orientador: Prof. Dr. Leandro Magatão

**CURITIBA**

**2013**

À memória do meu avô, Antônio F. de Meira, que sempre me perguntava quando eu iria me formar.

## RESUMO

TSUNODA MEIRA, William Hitoshi. PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM ENTREGAS E COLETAS MISTAS E JANELAS DE TEMPO: APLICAÇÃO EM UMA EMPRESA DA REGIÃO METROPOLITANA DE CURITIBA. 96 p. Trabalho de Conclusão de Curso – Departamento Acadêmico de Eletrônica e Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

A Pesquisa Operacional (PO) é um ramo da matemática aplicada que busca modelar problemas reais em diversas áreas do conhecimento. Uma das aplicações com destaque em PO é o problema de roteamento de veículos (*Vehicle Routing Problem* – VRP). O objetivo do trabalho é desenvolver uma abordagem de solução para o problema de roteamento de veículos de uma transportadora na região metropolitana de Curitiba. Na metodologia, leva-se em consideração restrições e características de cenários reais para desenvolvimento e validação do modelo matemático. Em particular, o problema em estudo caracteriza-se como sendo uma variante do VRP, o VRPMPDTW (*Vehicle Routing Problem with Mixed Pickup and Delivery and Time Windows*). Para a modelagem e resolução de cenários da empresa, utiliza-se uma técnica oriunda de PO, a Programação Linear Inteira Mista (PLIM), associada com algoritmos de pré-processamento e pós-processamento. Adicionalmente, desenvolveu-se uma interface humano-computador em Java para manipulação de cenários e análise dos resultados. O solucionador utilizado para a modelagem matemática foi o *IBM ILOG CPLEX Optimization Studio 12.5*. Foram obtidos resultados válidos em tempo computacional não proibitivo (segundos a poucos minutos) para cenários típicos da empresa. Contribuições futuras podem visar à diminuição do tempo computacional para cenários similares ou com um número maior de veículos e serviços, pois o tempo de resposta tende a crescer exponencialmente conforme essas variáveis aumentam.

**Palavras-chave:** Pesquisa Operacional (PO). Problema de Roteamento de Veículos (VRP). VRPMPDTW. PLIM. Logística de transporte

## ABSTRACT

TSUNODA MEIRA, William Hitoshi. VEHICLE ROUTING PROBLEM WITH MIXED PICKUP AND DELIVERY AND TIME WINDOWS. 96 p. Trabalho de Conclusão de Curso – Departamento Acadêmico de Eletrônica e Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

The Operational Research (OR) is a branch of applied mathematics that has the purpose of modeling real problems in various areas of knowledge. One of the applications with emphasis in OR is the Vehicle Routing Problem (VRP). The objective of the work is to develop a solution approach for the vehicle routing problem of a haulier from the metropolitan region of Curitiba. The methodology takes into account constraints and characteristics of real scenarios for development and validation of the mathematical model. In particular, the studied problem is characterized as a variant of the VRP, the VRPMPDTW (Vehicle Routing Problem with Mixed Pickup and Delivery and Time Windows). In order to model and solve scenarios of the company, it is used a technique originated from OR, the Mixed Integer Linear Programming (MILP), associated with pre-processing and post-processing algorithms. Additionally, it was developed a human-computer interface in Java to manage scenarios and analyse results. The solver used for mathematical modeling was the IBM ILOG CPLEX Optimization Studio 12.5. Valid results were obtained in non-prohibitive computational time (seconds to few minutes) for typical scenarios of the company. Future contributions may focus on reducing the computational time for scenarios with similar or greater numbers of vehicles and services, since the response time tends to grow exponentially as these variables increase.

**Keywords:** Operation Research (OR). Vehicle Routing Problem (VRP). VRPMPDTW. MILP. Transport Logistic

## LISTA DE FIGURAS

FIGURA 1	– Exemplo CVRP. ....	21
FIGURA 2	– Classificação de VRPs com foco em DPP. ....	23
FIGURA 3	– Comparação de Solução entre VRPB e Mixed VRPB. ....	25
FIGURA 4	– Comparação entre SOAP e REST. ....	30
FIGURA 5	– Diagrama de Casos de Uso. ....	36
FIGURA 6	– Fluxo Model-View-Controller. ....	41
FIGURA 7	– Fluxo de Execução do Solucionador. ....	43
FIGURA 8	– Entidade Scenario. ....	44
FIGURA 9	– Entidades Solution. ....	46
FIGURA 10	– Tela de Gerenciamento de Cenários. ....	47
FIGURA 11	– Tela de Gerenciamento de Cenários Executando Solucionador. ....	48
FIGURA 12	– Tela de Criação de Novo Cenário. ....	49
FIGURA 13	– Tela de Edição de Cenário. ....	50
FIGURA 14	– Tela de Criar Veículo. ....	50
FIGURA 15	– Tela de Criar Serviço. ....	51
FIGURA 16	– Tela de Solução do Cenário. ....	52
FIGURA 17	– Tela de Solução do Cenário com Erro. ....	53
FIGURA 18	– Solução do Estudo de Caso 1. ....	66
FIGURA 19	– Solução do Estudo de Caso 1: Alteração (i). ....	67
FIGURA 20	– Solução do Estudo de Caso 1: Alteração (ii). ....	68
FIGURA 21	– Solução do Estudo de Caso 2 - Caminhão 3/4(1). ....	72
FIGURA 22	– Solução do Estudo de Caso 2 - Caminhão 3/4(2). ....	73
FIGURA 23	– Solução do Estudo de Caso 2 - Caminhão 3/4(3). ....	74
FIGURA 24	– Cronograma Estimado. ....	77
FIGURA 25	– Cronograma Real. ....	77

## LISTA DE TABELAS

TABELA 1	–	Informações sobre a transportadora em estudo. ....	16
TABELA 2	–	Estudo de Caso 1: matriz de tempo de viagem. ....	65
TABELA 3	–	Dados do Estudo de Caso 1. ....	65
TABELA 4	–	Estudo de Caso 2: matriz de tempo de viagem. ....	71
TABELA 5	–	Dados do Estudo de Caso 2. ....	71
TABELA 6	–	Alocação de horas das atividades do projeto. ....	79

## LISTA DE QUADROS

QUADRO 1 –	Requisitos funcionais. ....	34
QUADRO 2 –	Requisitos não-funcionais. ....	35



## LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
AWT	<i>Abstract Window Toolkit</i>
CLP	<i>Constraint Logic Programming</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CPM	<i>Critical Path Method</i>
CVRP	<i>Capacitated Vehicle Routing Problem</i>
DPP	<i>Delivery and Pickup Problems</i>
EF	<i>Environmental Factor</i>
HVRPMBTW	<i>Heterogeneous Fleet Vehicle Routing Problem with Mixed Backhauls and Time Windows</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IBM	<i>International Business Machines</i>
IDE	<i>Integrated Development Environment</i>
mTSP	<i>multi Traveling Salesman Problem</i>
MVC	<i>Model-View-Controller</i>
MVRPB	<i>Mixed Vehicle Routing Problem with Backhaul</i>
NP-hard	<i>Non-deterministic Polynomial-time hard</i>
OPL	<i>Optimization Programming Language</i>
PDP	<i>Pickup and Delivery Problem</i>
PERT	<i>Program Evaluation and Review Technique</i>
PL	<i>Programação Linear</i>
PLIM	<i>Programação Linear Inteira Mista</i>
PO	<i>Pesquisa Operacional</i>
PSO	<i>Particle Swarm Optimization</i>
REST	<i>Representational State Transfer</i>
RPC	<i>Remote Procedure Call</i>
SDPP	<i>Simultaneous Delivery and Pickup Problem</i>
TCC	<i>Trabalho de Conclusão de Curso</i>
TCF	<i>Technical Complexity Factor</i>
TSP	<i>Traveling Salesman Problem</i>
TW	<i>Time Windows</i>
UAW	<i>Unadjusted Actor Weight</i>
UCP	<i>Use Case Points</i>
URL	<i>Uniform Resource Locator</i>
UTFPR	<i>Universidade Tecnológica Federal do Paraná</i>
UUCP	<i>Unadjusted Use Case Point</i>
UUCW	<i>Unadjusted Use Case Weight</i>
VRP	<i>Vehicle Routing Problem</i>
VRPB	<i>Vehicle Routing Problem with Backhaul</i>
VRPDP	<i>Vehicle Routing Problem with Delivery and Pickup</i>
VRPMB	<i>Vehicle Routing Problem with Mixed Backhaul</i>

VRPMPD	<i>Vehicle Routing Problem with Mixed Pickup and Delivery</i>
VRPMPDTW	<i>Vehicle Routing Problem with Mixed Pickup and Delivery and Time Windows</i>
VRPSPD	<i>Vehicle Routing Problem with Simultaneous Pickup and Delivery</i>
XML	<i>Extensible Markup Language</i>
XSD	<i>XML Schema Definition</i>
WSDL	<i>Web Services Description Language</i>

## LISTA DE SÍMBOLOS

kg	Quilograma
m <sup>3</sup>	Metros Cúbicos
km	Quilômetro
km/h	Quilômetro por hora

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>13</b>
1.1 CARACTERIZAÇÃO DO PROBLEMA	14
1.2 JUSTIFICATIVA	15
1.3 OBJETIVOS	16
1.3.1 Objetivo Geral	16
1.3.2 Objetivos Específicos	17
1.4 METODOLOGIA	17
1.5 ESTRUTURA DO DOCUMENTO	18
<b>2 ESTUDOS</b>	<b>19</b>
2.1 PROBLEMA DE ROTEAMENTO DE VEÍCULOS	19
2.1.1 Problema do Caixeiro Viajante	19
2.1.2 Problema de Roteamento de Veículos Capacitado	20
2.1.3 Métodos de Solução	20
2.1.4 Variantes do Problema de Roteamento de Veículos	22
2.1.5 Problemas Estáticos e Dinâmicos	22
2.1.6 Problema de Entregas e Coletas	23
2.1.7 Problema de Roteamento de Veículos com Entregas e Coletas Mistas e Janelas de Tempo	25
2.2 PROGRAMAÇÃO MATEMÁTICA	26
2.2.1 Programação Linear Inteira Mista	27
2.2.2 Formulação <i>Big-M</i>	28
2.2.3 Solucionador	28
2.3 <i>WEB SERVICES</i>	29
2.3.1 REST	29
2.3.2 Serviços REST Bing <i>Maps</i>	30
2.4 CONSIDERAÇÕES	31
<b>3 DESENVOLVIMENTO</b>	<b>33</b>
3.1 LEVANTAMENTO DE REQUISITOS	33
3.2 DIAGRAMA DE CASOS DE USO	34
3.2.1 Manter Cenário	35
3.2.2 Executar Solucionador	38
3.2.3 Mostrar Solução	39
3.3 ARQUITETURA DO <i>SOFTWARE</i>	40
3.4 FLUXO EXECUTAR SOLUCIONADOR	42
3.5 CAMADA DE MODELO	43
3.6 CAMADA DE VISÃO E CONTROLE	46
3.6.1 Tela de Gerenciamento de Cenários	47
3.6.2 Tela de Criação ou Edição de Cenários	48
3.6.3 Tela de Solução do Cenário	51
3.7 FORMULAÇÃO DO MODELO MATEMÁTICO	52
3.7.1 Notação de Conjuntos e Índices	53

3.7.2	Notação de Parâmetros .....	54
3.7.3	Notação de Variáveis .....	56
3.7.4	Função Objetivo .....	57
3.7.5	Restrições .....	58
3.8	CONSIDERAÇÕES .....	61
<b>4</b>	<b>RESULTADOS .....</b>	<b>63</b>
4.1	ESTUDO DE CASO 1 .....	63
4.1.1	Parâmetros de Entrada .....	63
4.1.2	Solução .....	65
4.1.3	Análise .....	65
4.2	ESTUDO DE CASO 2 .....	67
4.2.1	Parâmetros de Entrada .....	67
4.2.2	Solução .....	70
4.2.3	Análise .....	72
4.3	CONSIDERAÇÕES .....	74
<b>5</b>	<b>GESTÃO DE PROJETO .....</b>	<b>76</b>
5.1	TEMPO .....	76
5.1.1	Cronograma .....	76
5.1.2	Atividades .....	78
5.2	CUSTOS .....	78
5.3	RISCOS .....	79
5.4	CONSIDERAÇÕES .....	80
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>82</b>
	<b>REFERÊNCIAS .....</b>	<b>84</b>
	Apêndice A – XML CENÁRIO .....	87
	Apêndice B – CÓDIGO MODELO DE DADOS .....	91
	Apêndice C – CÓDIGO MODELO MATEMÁTICO .....	92

## 1 INTRODUÇÃO

A Pesquisa Operacional (PO) é um ramo da matemática aplicada, que procura modelar matematicamente problemas reais enfrentados em diversas áreas do conhecimento (WILLIAMS, 1999). Uma das aplicações com destaque em PO é o problema de roteamento de veículos, referenciado por VRP (*Vehicle Routing Problem*).

O VRP foi proposto inicialmente por Dantzig e Ramser (1959), podendo ser considerado como uma generalização para o problema do caixeiro viajante. Hoje possui diversas variantes que procuram solucionar diferentes características que surgem em sistemas de transporte de mercadorias ou de pessoas.

Em Laporte e Osman (1995) realiza-se uma revisão das principais características de problemas de roteamento. Estes problemas surgem, por exemplo, em transportes de cargas terrestres. O roteamento está associado ao ato de movimentar alguém ou um bem de consumo ao longo de uma rota. Problemas reais de roteamento de veículos podem necessitar a consideração de várias restrições, como faixa de horário que um cliente pode ser atendido, capacidade de transporte dos veículos, existência de frota heterogênea, quantidade de depósitos envolvidos, possibilidade de realização de coletas e entregas ao longo da rota de um veículo, dentre outras características (LIN, 2008).

Nesse trabalho aborda-se o caso real de roteamento de veículos em uma transportadora da região metropolitana de Curitiba. Consideram-se restrições e características do cenário em estudo com o objetivo de desenvolver uma abordagem de solução para o problema. Em particular, o problema em estudo, conforme apresentado na seção 1.1, caracteriza-se como sendo uma variante do VRP, o VRPMPDTW (*Vehicle Routing Problem with Mixed Pickup and Delivery and Time Windows*).

O VRPMPDTW, conforme Belmecheri et al. (2012) e Wang e Chen (2013), provém do subproblema VRPDP (*Vehicle Routing Problem with Delivery and Pickup*), onde veículos não são requeridos apenas para entregas, mas também para coleta de mercadorias.

Com relação à ordem na qual as tarefas de coleta e entrega são atendidas, o pro-

blema se divide em categorias como o VRPMPD (*Vehicle Routing Problem with Mixed Pickup and Delivery*), onde as coletas e entregas podem ser executadas em qualquer ordem, permitindo que o veículo esteja carregado com mercadorias para entrega e durante o trajeto de entregas também realize coletas. Este fato difere o problema de outra variante conhecida como VRPB (*Vehicle Routing Problem with Backhaul*), onde todas as tarefas de entrega são primeiramente realizadas, esvaziando o veículo, para depois começar a sequência de coletas. A sigla TW (*Time Windows*) indica que o problema leva em consideração restrições de janelas de disponibilidade (janelas de tempo) para o atendimento de um cliente.

Conforme salientado por Laporte e Osman (1995), problemas de roteamento de veículos (e suas variantes) são problemas combinatoriais de difícil resolução. O presente trabalho aborda um caso real de roteamento de veículos. Assim, do ponto de vista prático, soluções alternativas à prática operacional, podem vir a serem propostas. Do ponto de vista teórico, almeja-se criar uma abordagem de solução para cenários reais deste problema combinatorial de otimização, contribuindo para o estado da arte do tema.

## 1.1 CARACTERIZAÇÃO DO PROBLEMA

A transportadora em questão atende a região metropolitana de Curitiba realizando entregas e coletas de mercadorias para diversos clientes. A empresa possui uma quantidade limitada de veículos, sendo que nem todos têm a mesma capacidade de carga.

Um único depósito abriga todos os produtos para as entregas ainda não realizadas na região e que devem ser alocadas no início do dia para os veículos. Salienta-se que a região considerada envolve deslocamentos de poucas dezenas de quilômetros e os veículos devem retornar ao depósito até o final do dia. Além de entregas são também alocadas coletas a serem realizadas. Portanto, um veículo sai do depósito com um roteiro de coletas e entregas e, ao terminar as tarefas, retorna ao depósito.

A região atendida pela empresa é setORIZADA em sub-regiões para a posterior distribuição de tarefas. Existem mais depósitos localizados em regiões distantes, porém não será levado em consideração entregas/coletas a partir destes depósitos.

Veículos possuem características como capacidade de carregamento, custos envolvendo combustível, velocidade média de movimento, além do turno de trabalho associado ao motorista. Há restrições associadas ao cliente como, por exemplo: tempo de atendimento desde a coleta até a entrega e janelas de tempo do cliente para receber as

mercadorias (TW da caracterização como VRPMPD**TW**). Estes fatores influenciam na qualidade de serviço prestado. Outra restrição no cenário de estudo é a impossibilidade de haver trocas de mercadorias entre veículos.

Adicionalmente, existe a possibilidade de surgir uma coleta durante o roteiro de algum veículo. Esta situação prática é denominada de coleta dinâmica. É estabelecida a comunicação com o motorista e avaliada a possibilidade de se encaixar no roteiro o novo serviço, sem influenciar nas tarefas já designadas. A coleta dinâmica não será tratada no desenvolvimento desse trabalho.

Se houver uma entrega e uma coleta designada para um mesmo cliente estas devem ser alocadas para um mesmo veículo e realizadas simultaneamente (MPD - *Mixed Pickup and Delivery* da caracterização VRPMPDTW). Evita-se, assim, que visitas em horários diferentes sejam realizadas ao mesmo cliente, contribuindo para a qualidade de serviço e diminuição de custos do processo.

Na prática operacional, a partir de um cenário de entrada, geram-se itinerários de entregas e coletas para cada um dos veículos, respeitando as restrições citadas. Ressalta-se que no presente trabalho o problema de como deve ser realizado o carregamento nos veículos não será detalhado. Contudo, serão respeitadas as capacidades de carga e volume de cada um dos veículos na geração dos roteiros.

A tabela 1 apresenta as características principais dos veículos da empresa estudada. Segundo dados históricos fornecidos pela empresa, a soma de coletas e entregas realizadas diariamente é, na média, igual a 10, podendo abranger toda região metropolitana de Curitiba.

## 1.2 JUSTIFICATIVA

Uma das principais motivações para o desenvolvimento do projeto é o fato de trabalhar com uma aplicação de Pesquisa Operacional real, em que a transportadora em questão fornece dados e informações do seu processo de trabalho. Principalmente, no que diz respeito a distribuição de veículos para realização de entregas e coletas.

A decisão de gerar os itinerários para atender todos os serviços fica a cargo de um funcionário experiente, que tenta organizá-los por regiões. Porém, o processo é ineficiente o que acarreta, muitas vezes, em atrasos ou uso desnecessário de veículos. Portanto, justifica-se a implantação de um *software* de apoio à tomada dessas decisões, pois além de facilitar o trabalho do operador acarretaria em uma diminuição dos custos para a empresa



**Tabela 1: Informações sobre a transportadora em estudo.**

Transportadora			
Média diária de Entregas + Coletas a partir do depósito central			10
Veículos			
Quantidade	Tipo	Capacidade	
		Carga ( kg)	Volumétrica ( m <sup>3</sup> )
3	3/4 <sup>1</sup>	3000	23
1	Truck <sup>2</sup>	12000	70

**Fonte: Autoria própria.**

e aumentaria a satisfação dos clientes com relação ao atendimento.

Por fim, o trabalho também se justifica como um incentivo à inserção de tecnologias de Pesquisa Operacional em apoio à tomada de decisões, não apenas para empresas de transporte terrestres, mas para todo o setor de logística com o objetivo de aumentar sua eficiência.

### 1.3 OBJETIVOS

Os objetivos do presente trabalho são divididos em objetivo geral e específicos.

#### 1.3.1 OBJETIVO GERAL

O objetivo principal é desenvolver uma abordagem de solução para o problema de roteamento de veículos de uma transportadora na região metropolitana de Curitiba.

Conforme descrito na seção 1.1, o problema em estudo caracteriza-se como sendo uma variante do VRP, o VRPMPDTW. Métodos exatos com núcleo em um modelo de otimização em Programação Linear Inteira Mista associado com algoritmos de pré-processamento são utilizados a fim de obter resultados em tempo computacional não proibitivo para os cenários estudados.

<sup>1</sup> Caminhões 3/4 possuem um único eixo traseiro e carroceria curta com comprimento máximo de 6,3 metros e largura máxima de 2,2 metros. Adequado para áreas urbanas.

<sup>2</sup> Caminhões *Truck* possuem eixo duplo na carroceria, ou seja, dois eixos juntos. Podem carregar maiores cargas que os caminhões 3/4. Seu comprimento máximo é de 14 metros.

### 1.3.2 OBJETIVOS ESPECÍFICOS

- Estudar o problema da empresa e procurar problemas semelhantes na literatura.
- Projetar e desenvolver um *software* para interface ao apoio à tomada de decisões do problema da empresa.
- Implementar/adaptar uma formulação matemática que modele o problema em análise.

### 1.4 METODOLOGIA

A metodologia empregada pode ser dividida em quatro etapas:

**Etapa 1 - Estudos:** essa etapa envolveu o conhecimento do problema. Realizou-se o levantamento bibliográfico através de pesquisas em artigos e dissertações publicados na área de logística de transporte de carga terrestre. Além disso, abrangeu o estudo das tecnologias envolvidas no desenvolvimento do projeto, entre elas programação matemática e o método de solução PLIM, aprendido da linguagem de modelagem utilizada pelo solucionador IBM ILOG CPLEX, *Web Services* e o protocolo de comunicação REST (*Representational State Transfer*) e conceitos relacionados com a linguagem de programação Java e o paradigma Orientado a Objetos.

**Etapa 2 - Desenvolvimento do *Software*:** durante essa etapa foi elaborado o projeto de *software*, modelando a estrutura a ser utilizada. Levou-se em consideração a comunicação com o solucionador e com os serviços e bibliotecas necessárias. Conforme a implementação, o projeto foi sendo refatorado para condizer melhor com aspectos antes não analisados que só ficaram claros depois do amadurecimento do trabalho.

**Etapa 3 - Desenvolvimento do Modelo Matemático:** baseado em modelos matemáticos encontrados na literatura, foi desenvolvido um modelo levando em consideração as restrições características da empresa em estudo.

**Etapa 4 - Análise dos Resultados e Testes:** após a integração do *software* e do modelo matemático foram realizados testes utilizando diferentes cenários com tamanhos variados a fim de analisar as soluções, averiguando se estavam condizentes com os resultados esperados. Em caso de erros, as etapas de desenvolvimento eram revisadas e alteradas, conforme necessário.

## 1.5 ESTRUTURA DO DOCUMENTO

Nesse capítulo foi feita uma introdução ao problema proposto, caracterizando o sistema utilizado pela empresa em estudo, e apresentando-se a justificativa, o objetivo geral e os objetivos específicos do projeto.

No capítulo 2 é feito o estudo da literatura necessária para compreensão do projeto. Envolve o estudo de VRP e algumas de suas variantes, métodos de solução e o estudo das tecnologias e padrões utilizados no desenvolvimento.

No capítulo 3 é descrito o desenvolvimento da ferramenta. Inclui a metodologia, o projeto de *software* e o desenvolvimento do modelo matemático.

No capítulo 4 são apresentados os resultados do sistema desenvolvido, avaliando alguns cenários típicos da transportadora.

No capítulo 5 são detalhados aspectos relacionados com a gestão do projeto e os recursos nele utilizados. Analisa-se os prazos, custos e riscos.

No capítulo 6 são feitas as considerações finais do projeto.

## 2 ESTUDOS

Neste capítulo são apresentados os estudos realizados para o desenvolvimento do projeto. Serão abordados os seguintes assuntos: o problema de roteamento de veículos e suas variantes; Programação Matemática e Programação Linear Inteira Mista e tecnologias adotadas para desenvolvimento do *software*.

### 2.1 PROBLEMA DE ROTEAMENTO DE VEÍCULOS

Desde a sua formulação na década de 1950 por Dantzig e Ramser (1959), o problema de roteamento de veículos vem sendo muito estudado devido, principalmente, à sua grande importância no setor logístico. O problema possui complexidade elevada conforme o número de pontos de atendimento aumenta. A sua classificação está entre os problemas NP-hard (*Non-deterministic Polynomial-time hard*).

A presente seção contextualiza o problema de roteamento de veículos desde o problema do caixeiro viajante, que serve como base ao VRP, até suas diversas variantes, que incluem restrições como janelas de tempo, e ao conjunto de problemas denominados de coleta e entrega (*Delivery and Pickup Problems* - DPP). Apresentam-se também os principais métodos utilizados para a solução de VRPs, como abordagens exatas, algoritmos aproximados e soluções heurísticas. O estudo prossegue até a caracterização de problemas semelhantes como o da transportadora em estudo, explorando-se as soluções empregadas.

#### 2.1.1 PROBLEMA DO CAIXEIRO VIAJANTE

A base dos problemas de roteamento é o Problema do Caixeiro Viajante (*Traveling Salesman Problem* - TSP). O TSP pode ser definido a partir de um grafo  $G = (V, A)$ , onde  $V$  é um conjunto de  $n$  cidades e  $A$  um conjunto de arcos com custos associados à distância, partindo-se do vértice  $i$  para o vértice  $j$  (LAPORTE, 1991).

O objetivo do problema é encontrar um circuito Hamiltoniano com a menor distância percorrida pelo vendedor. Entende-se por circuito Hamiltoniano o circuito que

passa por todos os  $n$  vértices uma única vez. Se o custo associado ao arco do vértice  $i$  até o vértice  $j$  for igual ao caminho inverso diz-se que o TSP é simétrico, caso contrário assimétrico. Quanto ao número de vendedores, se existir mais de um vendedor deve-se montar uma rota para cada um sem repetir vértices. Esta condição de múltiplos vendedores dá origem ao mTSP (*multi Traveling Salesman Problem*) (DAVENDRA, 2010). Uma pesquisa mais aprofundada sobre TSPs pode ser encontrada em Laporte (1991) e Davendra (2010).

### 2.1.2 PROBLEMA DE ROTEAMENTO DE VEÍCULOS CAPACITADO

O problema de roteamento de veículos é uma generalização do TSP, ou melhor, uma generalização do mTSP considerando-se condições adicionais (DANTZIG; RAMSER, 1959). Um conjunto de veículos deve realizar entregas para uma quantidade de clientes distribuídos em uma região. A rota de atendimento deve iniciar e terminar no mesmo depósito e nenhum cliente pode ser visitado duas vezes. O objetivo do problema é minimizar a distância total percorrida. Em sua forma clássica, o problema está sujeito à restrição de capacidade que cada veículo pode carregar. Caracteriza-se esta forma como sendo o CVRP (*Capacitated Vehicle Routing Problem*).

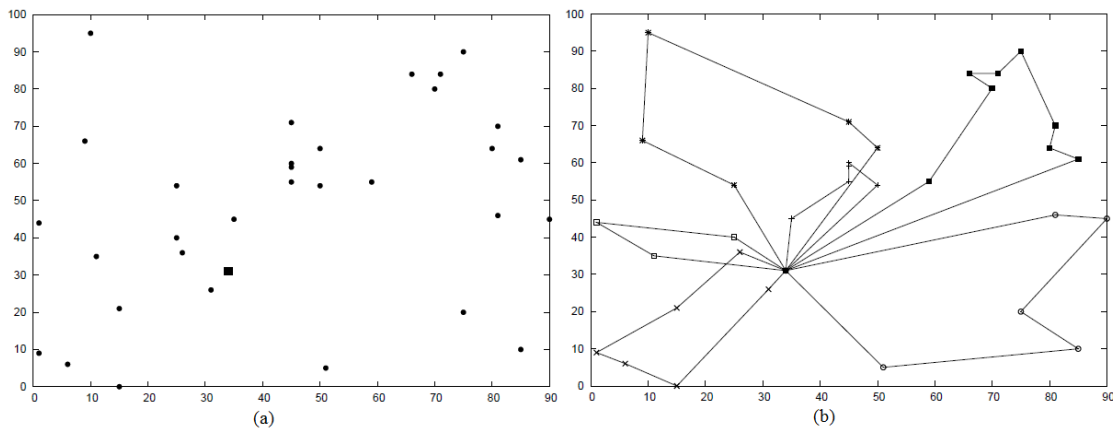
Na figura 1 tem-se um exemplo de CVRP com 32 clientes (pontos) espalhados em uma região, um depósito central (quadrado) e uma frota com 6 veículos. Nota-se que, durante a rota, veículos podem passar próximo a um cliente, sendo a tarefa mesmo assim delegada para outro veículo. Este fato pode ser resultado, por exemplo, da restrição de capacidade de cada veículo.

A viabilidade de solução do CVRP para um número de consumidores maiores é estudada na literatura e formas diversas de solucionar o problema vêm sendo desenvolvidas no decorrer dos últimos anos, como apresentado a seguir no item 2.1.3. O problema também se ramifica em diversos outros mais específicos e uma revisão das principais variantes é apresentada no item 2.1.4.

### 2.1.3 MÉTODOS DE SOLUÇÃO

Inúmeros métodos de solução para CVRP foram elaborados no decorrer dos últimos 50 anos, desde a sua proposição. Eles podem ser classificados em dois grandes grupos: métodos heurísticos e métodos determinísticos.

Abordagens heurísticas e metaheurísticas se utilizam de métodos que tendem a en-



**Figura 1: Exemplo CVRP: (a) 32 clientes (pontos) distribuídos em uma região e um depósito central (quadrado), (b) Solução ótima para o problema utilizando 6 veículos.**

**Fonte: Ropke (2005).**

contrar soluções satisfatórias, contudo na maioria dos casos, a prova de otimalidade destas soluções obtidas não é realizada. A tendência é que estas abordagens sejam empregadas para resolução de problemas reais e instâncias de grande porte, onde a carga computacional de abordagens exatas se mostrou proibitiva. Algumas das abordagens utilizadas são algoritmos genéticos, colônias de formiga, redes neurais, heurísticas de melhoria como Or-opt, 2-opt, busca Tabu, dentre outras. Informações aprofundadas sobre soluções heurísticas e metaheurísticas em VRP podem ser encontrados em sínteses como Cordeau et al. (2005) e Vidal et al. (2012). Adicionalmente, Ropke (2005) e Subramanian (2012) citam diversos trabalhos nessa área.

Em uma descrição simplificada, algoritmos exatos são métodos que objetivam obter a solução ótima para um modelo matemático do problema abordado. Uma revisão relativamente antiga de soluções exatas para o VRP é apresentada em Laporte (1987), com uma separação em três categorias de algoritmos: busca direta em árvore, Programação Dinâmica e Programação Linear Inteira. A última categoria, segundo os autores, pode, ainda, ser dividida em formulações de particionamento em grupos, de fluxo de veículos ou de fluxo de mercadoria. Complementarmente, Toth e Vigo (2002) e Baldacci et al. (2010) fazem um levantamento de métodos e tendências de abordagens exatas.

Em Subramanian (2012) são citados diversos trabalhos separados por variantes de VRP e apresenta-se uma linha do tempo das publicações e seus respectivos métodos de solução.

#### 2.1.4 VARIANTES DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS

Nos sistemas de transporte reais, há mais restrições em relação ao VRP base que deve ser levada em conta de modo a se modelar o problema. Há necessidade de considerar-se, por exemplo, o atendimento ao cliente no seu tempo disponível. Para delimitar as diferenças entre modelos, uma nova nomenclatura para cada variante do VRP é utilizada. Essas podem decorrer de vários aspectos como formação da frota (homogênea ou heterogênea), número de depósitos (único ou múltiplo), dinamicidade do atendimento (estático e determinístico, dinâmico ou estocástico), forma de atuação dos veículos (independente ou colaborativa), forma de serviço, função objetivo do modelo, entre outras possibilidades.

Usualmente os clientes possuem uma janela de tempo (intervalo temporal) no qual estão disponíveis para receber uma entrega. Esta característica restringe de forma significativa à rota que o veículo deverá tomar. Outro detalhe é a possibilidade ou não do veículo poder esperar para entrar na janela de tempo de um cliente. A sigla TW caracteriza VRPs que consideram essa restrição.

O problema pode envolver apenas entregas ou englobar coletas e entregas, assim caracterizando um grupo de problemas chamado DPP (*Delivery and Pickup Problems*) ou PDP (*Pickup and Delivery Problems*). A diferença entre DPP e PDP se refere a como o sistema é operado. No DPP considera-se apenas um pedido de serviço do cliente por vez; já em PDP um serviço é composto de mais de um pedido, uma coleta seguida de uma entrega. A notação é usada para diferenciação por alguns autores como Wang e Chen (2012) e Parragh et al. (2008a, 2008b). PDP não será abordado no presente trabalho. Para mais informações remete-se o leitor aos trabalhos de Parragh et al. (2008b) e Berbeglia et al. (2007). DPP será abordado por conta de semelhanças com o sistema da transportadora em estudo.

#### 2.1.5 PROBLEMAS ESTÁTICOS E DINÂMICOS

Nos problemas de roteamento estáticos todas as informações necessárias para a formulação das rotas são conhecidas antecipadamente, sem alterações no decorrer das rotas. Em problemas de roteamento dinâmico podem ocorrer solicitações de serviço depois dos veículos terem saído do depósito, portanto cada nova solicitação exige um replanejamento das rotas de forma a atender sua respectiva demanda (VECCHINI, 2011).

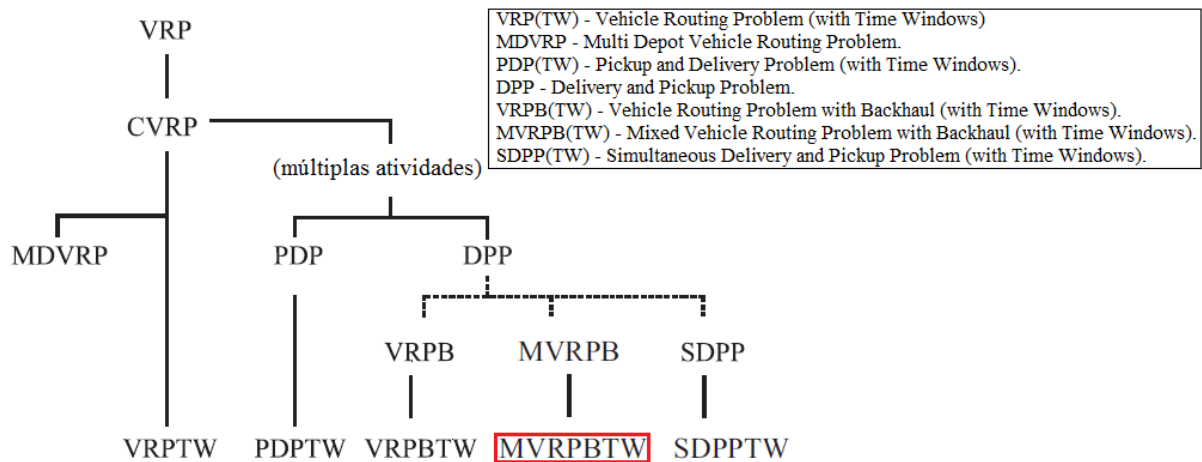
Na empresa em estudo existe a possibilidade de surgir uma coleta durante o roteiro

de algum veículo. No entanto, essa coleta dinâmica não será tratada no desenvolvimento do projeto, caracterizando o problema como estático. O problema de roteirização dinâmica de veículos com janelas de tempo é estudado com maior profundidade na dissertação de mestrado de Vecchini (2011) que também leva em consideração uma empresa real de transportes de carga.

### 2.1.6 PROBLEMA DE ENTREGAS E COLETAS

DPPs abordam problemas com multi-atividades, no caso, entregas e coletas em que um serviço é composto apenas por uma dessas atividades. Uma rota pode conter, ao mesmo tempo, entregas e coletas que são feitas de ou para o depósito central, nunca de um cliente para outro. Berbeglia et al. (2010) categoriza problemas de VRP com esse tipo de modo de operação como *one-to-many-to-one* (um para vários-vários para um).

DPPs podem ser classificados conforme a figura 2, adaptada de Wang e Chen (2012). Nesta figura, DPP é dividido em três subcategorias, baseado na estratégia utilizada: VRPB (*Vehicle Routing Problem with Backhaul*), MVRPB (*Mixed Vehicle Routing Problem with Backhaul*) e SDPP (*Simultaneous Delivery and Pickup Problem*). Na sequência faz-se uma descrição das três principais subcategorias de DPP apresentadas na figura 2.



**Figura 2: Classificação de VRPs com foco em DPP.**

**Fonte: Adaptado de Wang e Chen (2012).**

Nos VRPB(TW)s realizam-se todas as entregas primeiro e depois todas as coletas (*delivery-first, pickup-second*). Assim, o veículo nunca vai possuir simultaneamente na sua carga materiais de coletas e entregas. A estratégia é simplista com relação à im-

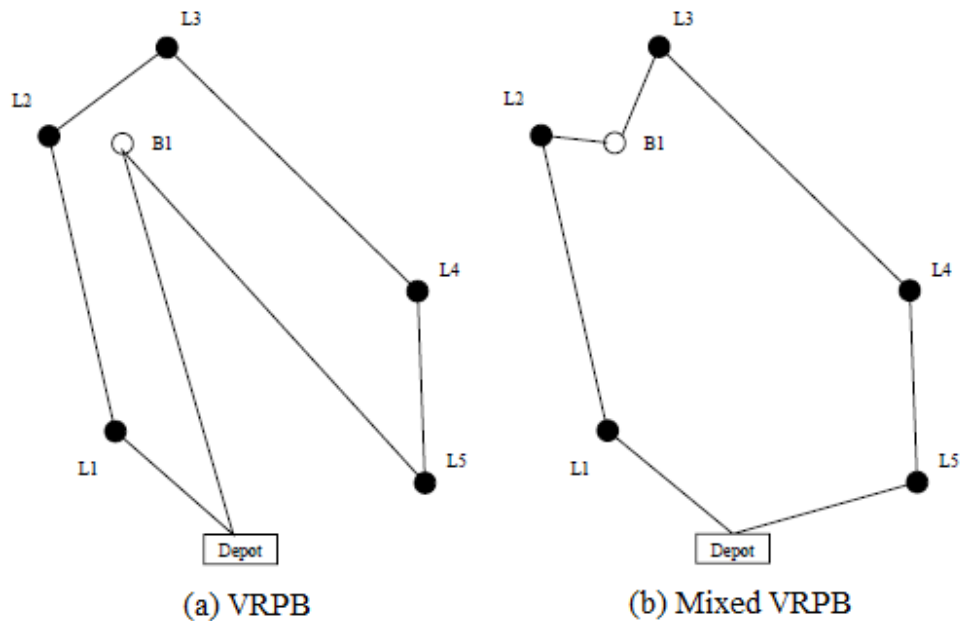


plementação, pois evita problemas com violação de capacidade que podem ocorrer ao se realizar uma coleta enquanto há entregas a serem realizadas. Considera-se que a soma das cargas de entrega e a soma das cargas de coleta não ultrapassem o valor da capacidade. Chama-se de *linehaul* os clientes que solicitam entregas e de *backhaul* os que solicitam coletas. Um veículo pode ter que visitar o cliente duas vezes, caso ele tenha dois serviços diferentes, ou realizar trajetos maiores para satisfazer a estratégia, o que é uma desvantagem. Goetschalckx e Jacobs-Blecha (1989) e Toth e Vigo (1997) estudam VRPB e métodos para solucioná-lo em seus respectivos trabalhos.

O SDPP aparece na literatura também com o nome VRPSPD (*Vehicle Routing Problem with Simultaneous Pickup and Delivery*). Nesse problema clientes requerem um serviço de entrega e coleta em seu estabelecimento e, como estratégia, os serviços são realizados de maneira simultânea. Assim, apenas uma parada em cada cliente é necessária, contribuindo para a diminuição do tempo de serviço. Permite-se que coletas e entregas estejam presentes ao mesmo tempo no veículo. Essa estratégia é muito utilizada em sistemas de transporte de uma cadeia de suprimentos cíclica, com logística inversa, por exemplo, em processos como o de reciclagem. Outro exemplo, as empresas de bebida que coletam engarrafados de garrafas vazias. Wang e Chen (2012) citam diversos trabalhos que estudam o problema e o próprio artigo apresenta uma solução com algoritmos genéticos para o SDPPTW.

O MVRPB(TW) é um caso particular de SDPP(TW) onde os clientes podem ter uma entrega ou uma coleta. Cada serviço é tratado de forma separada. Caso um cliente deseje uma entrega e uma coleta serão gerados dois serviços separados. Uma desvantagem é que pode ocorrer de um cliente ser visitado duas vezes, porém otimiza-se a estratégia em relação ao VRPB. Também é conhecido na literatura como VRPMPD (*Vehicle Routing Problem with Mixed Pickup and Delivery*) e VRPMB (*Vehicle Routing Problem with Mixed Backhaul*), citado em Subramanian (2012). O autor lista também diversos artigos com um histórico das soluções para MVRPB. Wade e Salhi (2002) descrevem o problema e utilizam uma heurística construtiva para solucioná-lo. Uma modelagem PLIM para o VRPMPD pode ser encontrada em Nagy et al. (2013).

Uma comparação das estratégias entre VRPB e uma MVRPB pode ser observada na figura 3, onde LX são *linehauls* e BX são *backhauls*. Observa-se que a solução (b) percorre uma distância menor do que (a).



**Figura 3:** Comparação de Solução entre (a) VRPB e (b) Mixed VRPB.

Fonte: Adaptado de Wade e Salhi (2002).

### 2.1.7 PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM ENTREGAS E COLETAS MISTAS E JANELAS DE TEMPO

Na figura 2 destacou-se com um retângulo o MVRPBTW, também conhecido como problema de roteamento de veículos com entregas e coletas mistas e janelas de tempo (VRPMPDTW). Isto se deve à semelhança deste problema com o caso prático da transportadora em estudo, que também possui restrições de janelas de tempo. Na transportadora, veículos carregados com mercadoria para entrega realizam uma rota cada, prestando serviços de entrega e coleta em uma região. Os veículos devem obedecer às restrições de faixas de tempo dos clientes e podem transportar, ao mesmo tempo, mercadorias para coleta e entrega, respeitando-se a restrição de capacidade dos veículos.

A descrição de problema apresentada por Belmecheri et al. (2012) é ainda mais específica, chamado de HVRPMBTW (*Heterogeneous Fleet Vehicle Routing Problem with Mixed Backhauls and Time Windows*). Belmecheri et al. (2012) também modelam matematicamente o problema e propõe uma solução heurística com PSO (*Particle Swarm Optimization*).

Vale lembrar que no caso real cada cliente poderá ter múltiplas janelas de tempo para atendimento. Um cliente pode, por exemplo, estar livre no período da manhã, ausentar-se para almoço e mais tarde estar livre novamente. Esse caso não é frequente-

mente tratado e é uma consideração extra que também será levada em conta no presente trabalho. Em Vieira (2012) apresenta-se uma abordagem baseada em PLIM para uma condição similar ao problema em estudo. Contudo, restrições de capacidade dos veículos ao longo das rotas e rigoroso respeito às janelas de tempo não são, por exemplo, observadas.

A literatura na variante específica do VRP identificada é relativamente escassa, só aparecendo pela primeira vez na década de 1980 e explorada mais profundamente a partir do ano 2000 com, na maioria dos casos, abordagens heurísticas, segundo lista Subramanian (2012). A pequena escala da transportadora em estudo permite, em tese, a utilização de uma abordagem exata ao problema. A abordagem exata vem sendo pouco explorada nessa variante e, desta forma, estudos adicionais com este foco são uma demanda. Desta forma, o presente trabalho pode vir a contribuir para o estado da arte neste tema específico.

## 2.2 PROGRAMAÇÃO MATEMÁTICA

A Programação Matemática é uma subárea da Pesquisa Operacional (GOLDBARG; LUNA, 2005). Por sua vez, a Pesquisa Operacional, conforme anteriormente mencionado, é um ramo da matemática aplicada voltada para a resolução de problemas reais enfrentadas em diversas áreas de conhecimento (WILLIAMS, 1999). Os modelos de Programação Matemática são estruturas representadas matematicamente através de equações, inequações e dependências lógicas, que utilizam variáveis quantitativas com o objetivo de otimizar o funcionamento do sistema.

A Programação Matemática pode ser utilizada em sistemas de apoio à tomada de decisões no gerenciamento de sistemas de grande porte, desde que possível a modelagem de forma matemática. A técnica permite a observações de inter-relações entre diversas variáveis que seriam difíceis de descobrir intuitivamente (GOLDBARG; LUNA, 2005).

O campo de Programação Matemática abrange uma vasta quantidade de problemas e possui diversas aplicações que levaram a especificações e particularizações dos métodos de solução (GOLDBARG; LUNA, 2005). Dentre eles destacam-se: Programação Linear, Programação Não-linear, Programação Inteira, Programação Linear Inteira Mista.

O projeto utiliza o método de Programação Linear Inteira Mista (PLIM) para modelagem do problema.

### 2.2.1 PROGRAMAÇÃO LINEAR INTEIRA MISTA

Programação Linear Inteira Mista envolve modelos que podem apresentar variáveis contínuas, inteiras, ou até mesmo binárias, e que estão relacionadas por meio de restrições lineares. Variáveis binárias geralmente representam variáveis de decisão que podem significar sim ou não, ligado ou desligado, verdadeiro ou falso e permitem ao programador delinear condições lógicas ao modelo. Variáveis inteiras restringem grandezas consideradas indivisíveis, como, quantidade de veículos ou quantidade de pessoas.

O sistema de equações e inequações modelado possui um espaço de soluções válidas, ou espaço de busca. Por meio da avaliação da função objetivo do modelo, procura-se encontrar uma solução dentro do espaço de busca que maximize ou minimize grandezas de interesse. Por exemplo: minimizar custos, tempo de produção, uso de recursos; maximizar lucros, produtividade, capacidade de armazenamento (MAGATÃO, 2005). Matematicamente a função de maximização pode ser escrita como de minimização simplesmente multiplicando o objetivo por (-1) e minimizando-o, conforme a equivalência apresentada na expressão 2.1. Nesta equivalência,  $c_j$  são coeficientes (parâmetros) e  $x_j$  representam variáveis decisórias.

$$\max z = \sum_{j=1}^n c_j x_j \equiv \min (-z) = \sum_{j=1}^n (-c_j) x_j \quad (2.1)$$

PLIM oferece uma estrutura capaz de modelar uma grande variedade de problemas como: problema de alocação de pessoal, problema da mochila, problemas de fluxo em rede, problemas de roteamento, *scheduling*, problemas de cobertura, etc. Os problemas modelados, no entanto, devem ser resolvidos por um procedimento (algoritmo) de busca. Algoritmos de busca exploram soluções que satisfaçam as restrições delimitadas pelo modelo, incluindo-se as inerentes questões de integralidade de variáveis. Em PLIM, procedimentos de busca em árvore (*branch-and-bound*) são geralmente empregados. Estes procedimentos realizam um sucessivo particionamento do espaço de soluções válidas no intuito de explorar regiões promissoras para soluções do modelo (WILLIAMS, 1999).

Os modelos elaborados podem, no entanto, representar problemas que estejam em classes computacionais conhecidas como NP-Completo ou, ainda, NP-*Hard*. Na prática, isto significa que estes modelos podem se tornar de difícil resolução, ou mesmo serem computacionalmente intratáveis, na medida que os cenários de entrada avaliados aumentarem de dimensão (GOLDBARG; LUNA, 2000).

Adicionalmente, um conjunto de dados de entrada pode representar um cenário em que não existe solução válida para o modelo PLIM. Neste caso, o procedimento de busca reporta a infactibilidade do modelo, mas, via de regra, não indica a causa geradora desta infactibilidade. Na prática, contudo, pode-se inserir variáveis de folga para realizar o “afrouxamento” (relaxação) de certas restrições. O uso de variáveis de relaxação pode propiciar o entendimento de eventuais problemas de modelagem, ou mesmo evidenciar dados de entrada que estejam ocasionando a infactibilidade do modelo.

No presente trabalho, conforme evidenciado na seção 3.7, variáveis de folga (ou relaxação) são utilizadas na modelagem PLIM proposta.

### 2.2.2 FORMULAÇÃO *BIG-M*

A formulação *Big-M* é usada em PLIM quando precisa-se estabelecer uma conexão lógica entre variáveis binárias e as restrições com variáveis contínuas do sistema, sem perder-se a característica linear do equacionamento (MAGATÃO, 2005). Permite criar estruturas lógicas como operadores condicionais (*if-then*), além de restrições em forma de implicação ou equivalência.

O procedimento consiste na inserção de um parâmetro  $M$ , cujo valor supera em algumas vezes o valor máximo esperado para a parte contínua. Recomenda-se que o valor de  $M$  não seja muito elevado, pois tornaria o processo de diminuir espaço de busca dos algoritmos em árvore menos eficazes, além de poder causar instabilidades numéricas no processo de busca. Tomando-se um exemplo simples, em que  $x$  e  $y$  são variáveis contínuas e  $\delta$  uma variável binária e deseja-se formular linearmente a condição de que se  $\delta = 1$  então  $x - y \leq 0$ . Assim, pode-se formular utilizando a estrutura *Big-M* da inequação 2.2.

$$x - y \leq M(1 - \delta) \tag{2.2}$$

Se  $\delta = 1$ , satisfaz-se a condição desejada e, caso contrário, se  $\delta = 0$  a inequação pode assumir qualquer valor menor que  $M$ , sendo este próximo ao máximo possível para a diferença de  $x$  e  $y$ .

### 2.2.3 SOLUCIONADOR

Um solucionador (do inglês, *solver*) é um *software* que possui algoritmos destinados a resolver modelos implementados em PLIM. A modelagem do sistema é construída

utilizando uma linguagem declarativa que varia de acordo com o formato estabelecido pelo solucionador escolhido.

A linguagem de alguns solucionadores permite construir estruturas como conjuntos e tuplas, além de diversos operadores, somatórios e laços. Permite a criação do modelo completo com variáveis, restrições e função objetivo.

Neste trabalho foi utilizado o solucionador IBM ILOG CPLEX *Optimization Studio* 12.5. Esse solucionador é um *software* comercial e de propriedade da IBM, porém para este projeto, que possui escopo de formação educacional, foi utilizada a versão acadêmica.

### 2.3 WEB SERVICES

Para montar os cenários de entrada do modelo matemático, é necessário obter informações de rotas e tempo de viagem entre os nós, baseado nos endereços de cada cliente. Informações desse tipo são complicadas de obter sem o auxílio de alguma plataforma de mapas para geocodificação de endereços ou obter rotas. Por esse motivo, *Web Services* especializados foram utilizados no trabalho.

*Web Services* trazem consideráveis vantagens para o desenvolvimento de aplicações proporcionando a agilidade necessária para a criação de *softwares*. Devido a adoção de protocolos e padrões muito utilizados como XML e HTTP garante a operação dos serviços em conjunto com diversas linguagens de programação (CRUZ, 2005).

W3C (2004) define *Web Services* como um *software* designado para suportar a interoperabilidade entre máquinas na rede. Possui uma interface descrita em um formato processável por máquina, em especial o WSDL (*Web Services Description Language*) e SOAP (*Simple Object Access Protocol*) que utilizam o formato XML.

#### 2.3.1 REST

REST (*Representational State Transfer*) é um protocolo de comunicação cliente-servidor que utiliza o protocolo HTTP. Não mantém estado, ou seja, cada pedido do cliente para o servidor deve conter todas as informações necessárias sem levar em conta nenhuma consulta ou algum outro valor prévio como referência pelo servidor. Por outro lado, o protocolo permite que do lado do cliente os dados possam ser armazenados em *cache* para evitar requisições desnecessárias ao servidor (FIELDING, 2000).

REST é uma alternativa leve e simples para *Web Services* ou mecanismos como

CORBA (*Common Object Request Broker Architecture*), RPC (*Remote Procedure Call*). Assim como um *Web Service* é independente de plataforma, independente de linguagem, baseado em padrões (no caso, HTTP) e também não oferece recursos de segurança embutidos, mas que podem ser inseridos em cima do protocolo HTTP (ELKSTEIN, 2008).

O exemplo da figura 4, retirado de Elkstein (2008), mostra a simplicidade das solicitações REST (b) em comparação com o Web Service SOAP (a). Nele é descrito uma possível solicitação de detalhes da conta do usuário em uma aplicação de agenda telefônica. A solicitação REST é apenas uma URL em que todas as informações necessárias são nela inseridas. Por meio de um pedido GET, o servidor responde por HTTP enviando os dados crus, podendo ser utilizado diretamente sem necessitar nenhum processamento.

<pre>&lt;?xml version="1.0"?&gt; &lt;soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding"&gt;   &lt;soap:body pb="http://www.acme.com/phonebook"&gt;     &lt;pb:GetUserDetails&gt;       &lt;pb:UserID&gt;12345&lt;/pb:UserID&gt;     &lt;/pb:GetUserDetails&gt;   &lt;/soap:Body&gt; &lt;/soap:Envelope&gt;</pre>	(a)
<pre>http://www.acme.com/phonebook/UserDetails/12345</pre>	(b)

**Figura 4: Comparação entre (a) SOAP e (b) REST.**

**Fonte: Adaptado de Elkstein (2008).**

Serviços REST são fornecidos por diversas empresas, entre elas grandes nomes como Google, Amazon e Yahoo fornecem APIs (*Application Programming Interface*), que auxiliam no desenvolvimento de *softwares* antes impossibilitados pela incapacidade de obter de maneira fácil e barata essas informações.

### 2.3.2 SERVIÇOS REST BING MAPS

A Microsoft®Bing™*Maps Platform* fornece uma API de serviços REST que possibilita operações como geocodificação de endereços, criação de mapas estáticos, criação de rotas e obtenção de metadados de imagens com todo o suporte da plataforma de mapas Bing (BING, 2013).

A API oferece três tipos de chaves para acesso aos serviços: *trial*, *basic* e *enterprise*. A chave *trial* é livre e fornece acesso por 90 dias sem exceder 10.000 transações

de nenhum tipo de serviço em um período de 30 dias. A chave *basic* também é livre e pode ser utilizada por qualquer aplicação que não exceda 50.000 transações de nenhum tipo em um período de 24 horas, sendo adequada para este projeto. A chave *enterprise* é paga sob condições de acordo e serve para aplicações empresariais que exigem maior número de acessos. Todas as chaves devem ser utilizadas de acordo com os termos de serviço disponíveis em Bing (2013).

Neste trabalho foram utilizados os serviços de geocodificação de endereços, obtenção de rotas e de imagens estáticas com rotas traçadas. A geocodificação transforma um endereço em termos de latitude e longitude. A obtenção de rotas entre dois endereços retorna dados de distância, tempo de transporte e itinerário de viagem. O serviço de obtenção de imagens estáticas permite definir uma sequência de endereços, o formato da imagem de saída, o tamanho desta imagem, com a possibilidade de traçar rotas entre os pontos sobre a imagem retornada.

## 2.4 CONSIDERAÇÕES

A área de pesquisa logística aumentou consideravelmente em importância na atualidade, devido à crescente presença da tecnologia em seus processos. Muitos artigos foram publicados nos últimos anos tratando diversos casos específicos em diferentes regiões do planeta, sendo a particularidade de cada caso um incentivo a pesquisa e ao desenvolvimento de novos métodos de solução. O caso em estudo neste trabalho leva em consideração diversas características reais do problema, tornando-o específico e próximo a uma variante de VRP pouco explorada na literatura.

O fato do número de operações diárias da empresa ser relativamente pequeno possibilitou a escolha de uma abordagem exata utilizando PLIM, porém, conforme a entrada aumenta, o tempo de resolução cresce de forma proibitiva. Assim, torna-se inviável a utilização dessa abordagem para casos maiores, em que seria necessário a implementação de outros algoritmos, heurísticas ou, ainda, abordagens híbridas com a finalidade de diminuir o tempo de resolução computacional.

Além dos estudos citados, vale considerar também conhecimentos relacionados a programação como a linguagem Java que utiliza o paradigma orientado a objetos e a arquitetura de desenvolvimento de *software* MVC (*Model-View-Controller*). Esta arquitetura divide o programa em três camadas independentes, em que o *Model* é a camada dos objetos da aplicação, a *View* é a interface de acesso e o *Controller* reage através da interface às entradas do usuário (GAMMA et al., 2000).



No Capítulo 3, a seguir apresentado, os conceitos estudados no presente capítulo serão utilizados para o desenvolvimento da abordagem de solução ao problema em estudo.

### 3 DESENVOLVIMENTO

Este capítulo aborda o desenvolvimento do *software* e do modelo matemático. Descreve como o trabalho foi desenvolvido desde sua concepção em projeto até sua completude.

#### 3.1 LEVANTAMENTO DE REQUISITOS

Os requisitos definem o que o sistema deve fazer, quais os serviços, restrições e condições de desenvolvimento definidas pelo cliente (SOMMERVILLE, 2007). Neste trabalho, o sistema objetiva auxiliar o programador - usuário principal - nas atividades de manipulação de cenários e análise dos resultados provenientes do modelo matemático. No contexto, cenários envolvem um conjunto de dados de entrada que contém, por exemplo, quantidade de peso e volume de cargas a serem entregues e coletadas, número de veículos disponíveis e características de capacidade dos mesmos, janelas de tempo nas quais as entregas e coletas devem ser realizadas em cada cliente, tempos de viagem entre os pontos de entrega/coleta.

Os requisitos podem ser classificados em requisitos funcionais e requisitos não-funcionais. Requisitos funcionais são as funções ou serviços que o sistema deve fornecer, como o sistema deverá reagir a determinadas entradas e também podem estabelecer o que o sistema não deve fazer. Requisitos não-funcionais estabelecem restrições aos serviços, relacionam-se com o processo de desenvolvimento, definindo qualidades desejadas ao sistema como um todo (SOMMERVILLE, 2007).

Nos quadros 1 e 2, estão descritos, respectivamente, os requisitos funcionais e não-funcionais para o sistema deste trabalho.

<b>Requisitos Funcionais do Sistema</b>		
<b>Referência</b>	<b>Nome</b>	<b>Descrição</b>
<b>RF1</b>	Cadastrar Cenários	O sistema deverá permitir o cadastro de novos cenários.
<b>RF2</b>	Gerar Relação de Cenários	O sistema deverá exibir a relação dos cenários cadastrados.
<b>RF3</b>	Editar Cenários	O sistema deverá permitir a edição de cenários.
<b>RF4</b>	Excluir Cenários	O sistema deverá permitir a exclusão de cenários.
<b>RF5</b>	Solucionar Cenários	O sistema deverá solucionar o cenário selecionado.
<b>RF6</b>	Visualizar Solução	O sistema deverá permitir a visualização da solução do cenário, mostrando as rotas de cada veículo.
<b>RF7</b>	Visualizar Informações da Solução	O sistema deverá permitir a visualização das informações da solução do cenário.

**Quadro 1: Requisitos funcionais.**

**Fonte: Autoria própria.**

### 3.2 DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso define o comportamento pretendido para o sistema, sem especificar como será implementado. Um caso de uso é definido como uma descrição de uma sequência de eventos, incluindo variantes, que o sistema executa para produzir um resultado de valor observável para um usuário ou agente externo (ator) (BOOCH et al., 2005). Cada ator representa um papel desempenhado pelos usuários quando interagindo com os casos de uso. O ator pode ser humano, máquina ou outro sistema que atue sobre o sistema principal.

A figura 5 apresenta o diagrama de casos de uso deste projeto. O sistema possui três atores: o usuário, o serviço externo REST da plataforma Bing *Maps* e o solucionador IBM ILOG CPLEX. O usuário é atendido por todos os casos de uso, sendo o coordenador das tarefas. O solucionador, o IBM ILOG CPLEX, é um programa externo ao sistema principal e atua em conjunto com o usuário na tarefa de solucionar o cenário. O serviço

Requisitos Não-Funcionais do Sistema		
Referência	Nome	Descrição
RNF1	Solucionar Cenário em Tempo Não Proibitivo	O sistema deverá encontrar uma solução válida para cenários com até 15 serviços (coletas mais entregas) em menos de 20 minutos. Levando-se em consideração a velocidade de processamento de máquina utilizada no desenvolvimento do projeto, que possui um processador Intel Core i5 2.4 GHz e 4 GB de memória RAM.
RNF2	Tempo de Desenvolvimento de 6 Meses	O sistema deverá ser desenvolvido em, no máximo, 6 meses.

**Quadro 2: Requisitos não-funcionais.**

**Fonte: Autoria própria.**

externo fornece dados para criação dos gráficos do solucionador e informações de rotas necessárias para determinar as distâncias e tempo de viagem entre os clientes e o depósito. Os casos de uso são descritos nas subseções 3.2.1, 3.2.2 e 3.2.3.

### 3.2.1 MANTER CENÁRIO

**Referências:** RF1, RF2, RF3 e RF4.

**Atores:** Usuário e Serviços REST.

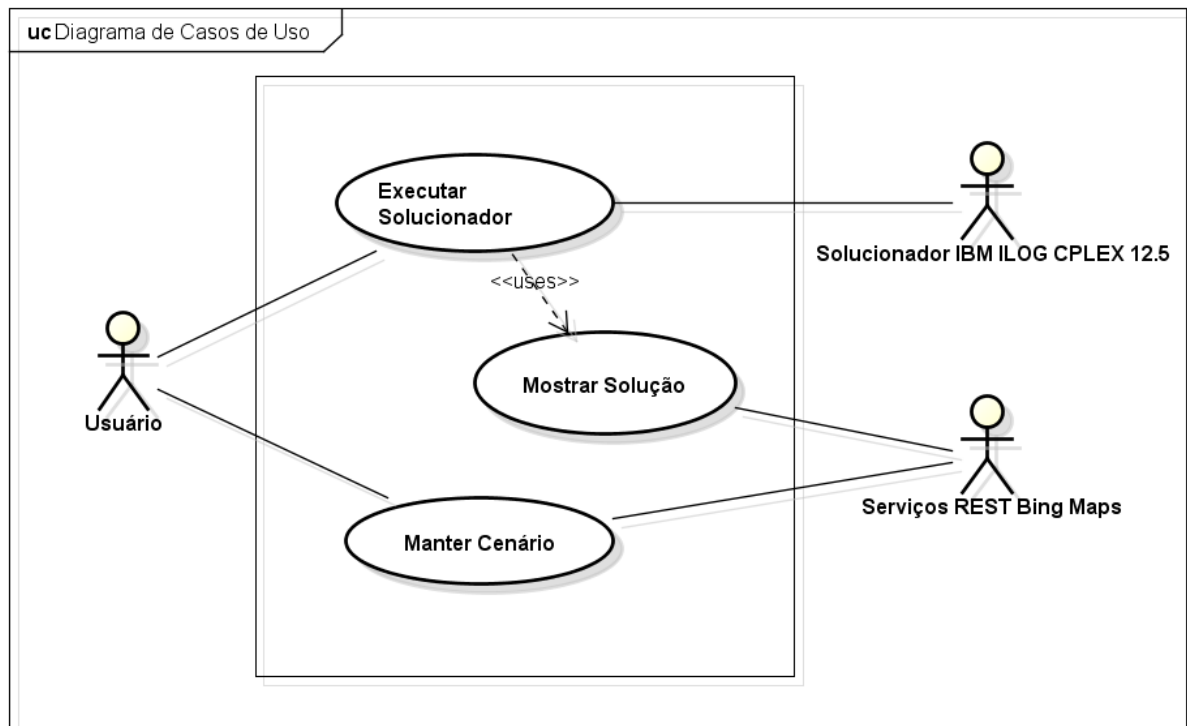
**Pré-Condições:** Nenhuma.

**Pós-Condições:** Cenários podem ser referenciados por outros casos de uso, fluxos alternativos 1 e 2.

**Fluxo Básico:**

1. O caso de uso inicia quando o ator entra na tela de gerenciamento de cenários (usuário).
2. A sistema exibe a lista de cenários cadastrados.
3. O fluxo segue para um dos fluxos alternativos.

**Fluxos Alternativos:**



**Figura 5: Diagrama de Casos de Uso.**

**Fonte: Autoria própria.**

1. O ator clica em cadastrar novo cenário. (programador)

(\*) A qualquer momento o ator clica em cancelar (usuário).

O sistema desconsidera as informações digitadas.

O sistema retorna para a tela de gerenciamento de cenários.

(a) O sistema mostra o formulário para criar o novo cenário (classe Scenario).

**Entregas** (classe Delivery)

**Coletas** (classe Pickup)

**Depósito Central** (classe Deposit)

**Veículos** (classe Vehicle)

(b) O ator insere todos os dados nos devidos campos e submete o cadastro (usuário).

(c) O sistema valida os dados de acordo com as regras de negócio 1, 2 e 3.

Se o cadastro for válido o sistema emite uma mensagem de sucesso.

Se não, o sistema emite uma mensagem de erro e retorna ao passo (b).

- (d) O serviço REST do Bing *Maps* calcula as rotas e o itinerário entre todos os nós (serviços e depósito).
  - (e) O sistema cadastra o novo cenário.
  - (f) O sistema retorna para a tela de gerenciamento de cenário.
2. O ator seleciona um cenário da lista e clica em editar cenário (usuário).
- (\*) A qualquer momento o ator clica em cancelar (usuário).
    - O sistema desconsidera as informações digitadas.
    - O sistema retorna para a tela de gerenciamento de cenários.
  - (a) O sistema mostra o formulário preenchido com os dados do cenário cadastrado (classe Scenario).
    - Entregas** (classe Delivery)
    - Coletas** (classe Pickup)
    - Depósito Central** (classe Deposit)
    - Veículos** (classe Vehicle)
  - (b) O ator edita os dados do cenário e solicita a atualização (usuário).
  - (c) O sistema valida os dados de acordo com as regras de negócio 2 e 3.
    - Se o cadastro for válido o sistema emite uma mensagem de sucesso.
    - Se não, o sistema emite uma mensagem de erro e retorna ao passo (b).
  - (d) O serviço REST do Bing *Maps* calcula as rotas e o itinerário entre todos os nós (serviços e depósito).
  - (e) O sistema cadastra o novo cenário.
  - (f) O sistema retorna para a tela de gerenciamento de cenário.
3. O ator seleciona um cenário da lista e clica em excluir cenário (usuário).
- (a) O sistema emite uma mensagem de confirmação.
    - Se o ator confirmar o fluxo vai para (b).
    - Se o ator cancelar o fluxo vai para (d).
  - (b) O sistema exclui permanentemente o cenário.
  - (c) O sistema emite uma mensagem de sucesso.
  - (d) O sistema retorna para a tela de gerenciamento de cenário.

**Regras de Negócio:**

1. O cenário não pode ser cadastrado com o mesmo nome de arquivo que outro cenário previamente existente.
2. O cenário deve possuir no mínimo 1 veículo.
3. Todos os campos obrigatórios devem ser preenchidos corretamente.

**3.2.2 EXECUTAR SOLUCIONADOR**

**Referências:** RF5 e RNF1.

**Atores:** Usuário e Solucionador.

**Pré-Condições:** Algum cenário válido cadastrado.

**Pós-Condições:** Informações da solução disponíveis para manipulação.

**Fluxo Básico:**

1. O caso de uso inicia quando o ator deseja executar o solucionador para o cenário selecionado (usuário).
2. O sistema executa o modelo matemático utilizando o solucionador IBM ILOG CPLEX *Optimization Studio* 12.5 (solucionador).
3. O ator espera a solução do cenário. O tempo de espera deve satisfazer a regra de negócio 1.
4. A solução é armazenada em uma estrutura de dados (classe Solution).
5. O fluxo segue para o fluxo básico do caso de uso Mostrar Solução.
6. O ator analisa os resultados (programador).

**Fluxos Alternativos:**

1. Após analisar os dados o ator pode clicar em fechar a janela.
  - (a) O sistema retorna para a tela de gerenciamento de cenários.
2. Caso (c) resulte em uma solução infactível.

- (a) O sistema apaga todas as informações obtidas.
  - (b) O sistema emite uma mensagem de erro.
  - (c) O sistema retorna para a tela de gerenciamento de cenários.
3. Caso a regra de negócio 1 não seja atendida.
- (a) O sistema para a execução.
  - (b) Caso nenhuma solução tenha sido encontrada:
    - O sistema inicia o fluxo alternativo 2 do caso de uso EXECUTAR SOLUCIONADOR.
  - (c) Caso alguma solução próxima da ótima tenha sido encontrada:
    - O sistema emite uma mensagem de estouro de tempo.
    - O sistema informa que solução encontrada não é a ótima.
    - O sistema segue para o fluxo básico 4 do caso de uso EXECUTAR SOLUCIONADOR.

### **Regras de Negócio:**

1. O tempo de espera para um cenário com até 20 serviços não pode ultrapassar 20 minutos.

### 3.2.3 MOSTRAR SOLUÇÃO

**Referências:** RF6 e RF7.

**Atores:** Usuário e Serviço REST.

**Pré-Condições:** Estrutura de dados obtida do solucionador (classe Solution).

**Pós-Condições:** Nenhuma.

### **Fluxo Básico:**

1. O caso de uso inicia quando o sistema recebe a estrutura de dados da solução do cenário (usuário).
2. O sistema através do uso do serviço REST do Bing *Maps* gera um gráfico mostrando a solução com as rotas de cada veículo (serviço REST).



3. O sistema mostra o itinerário de viagem entre cada ponto da rota encontrada pela solução.
4. O sistema mostra uma tabela com informações em cada ponto da rota encontrada pela solução.

### Fluxos Alternativos:

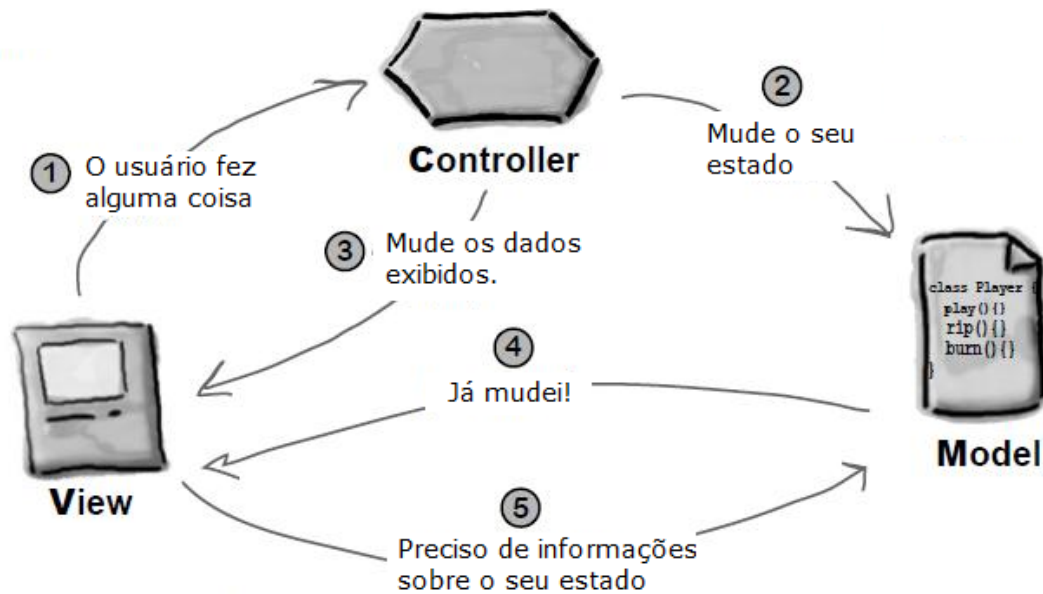
1. Após analisar os dados o ator pode clicar em fechar a janela.
  - (a) O sistema retorna para a tela de gerenciamento de cenários.

### 3.3 ARQUITETURA DO *SOFTWARE*

O *software* utiliza como estrutura base o padrão de arquitetura de desenvolvimento MVC em conjunto com as bibliotecas gráficas *Swing* e AWT (*Abstract Window Toolkit*) da linguagem Java. O padrão MVC divide o *software* em três camadas (FREEMAN, 2004):

- **Model:** contém todos os dados, os estados e toda lógica da aplicação. Providencia uma interface para manipulação e consulta do seu estado e envia notificações de alteração de estado para seus observadores. Representa as entidades.
- **View:** é a interface com o usuário e representa o estado atual do modelo. Consulta os dados que necessita diretamente do modelo. São as telas desenvolvidas utilizando as bibliotecas gráficas do Java.
- **Controller:** captura os dados de entrada, interpreta a ação requisitada e transmite para o modelo. O usuário requisita algo através de ações como cliques, em que os ouvintes (*listeners*) dos objetos gráficos (botões, caixas de textos, menus suspensos...), que são os *controllers*, interpretam o comando e fazem as preparações necessárias para o modelo executar a alteração correta.

A figura 6, adaptada do livro Freeman (2004), mostra como ocorre o fluxo de operações entre as camadas. Por exemplo, o usuário clica em um botão da tela para salvar as alterações realizadas, enviando, assim, as novas informações para a camada de controle que organiza os dados para serem entendidas pela camada de modelo. A camada de modelo altera os dados necessários, enquanto a visão espera a camada de controle e



**Figura 6: Fluxo Model-View-Controller.**

**Fonte: Adaptado de Freeman (2004).**

o modelo avisarem que terminaram as operações. A visão requisita as novas informações do modelo e altera as informações mostradas ao usuário.

Existe uma quarta camada, onde ficam armazenados os dados dos cenários. Para cada cenário existe um arquivo no formato XML, que possui uma estrutura definida, que é interpretada pelo *software*. O Apêndice A apresenta um exemplo de cenário XML estruturado. Utiliza-se a biblioteca Java *open-source* JDOM para manipulação do XML. Todos os cenários ficam armazenados em um diretório comum conhecido pelo programa. Quando o programa é iniciado, os cenários são lidos, validados e então transformados em entidades que irão compor a camada *model*. A validação é feita através de um arquivo XSD (*XML Schema Definition*) que verifica se a estrutura XML é condizente com a definida.

O arquivo de configuração **path.properties** está localizado no caminho base do programa e define os diretórios em que estão localizados os arquivos de cenários, os arquivos do modelo matemático, os ícones e também qual é o arquivo de validação de cenários. Por padrão não é necessário alterar seus valores, eles são relativos ao diretório em que está localizado o programa.

Outros dois componentes complementam a estrutura do *software*, sendo os dois relacionados aos atores externos: solucionador e serviços REST. O componente do solucionador é responsável pela comunicação com o IBM ILOG CPLEX, manipulando os

arquivos de dados e executando o modelo matemático, assim fornecendo os resultados para o programa. O componente de serviços REST faz a comunicação com a plataforma Bing *Maps*, obtendo as informações requisitadas.

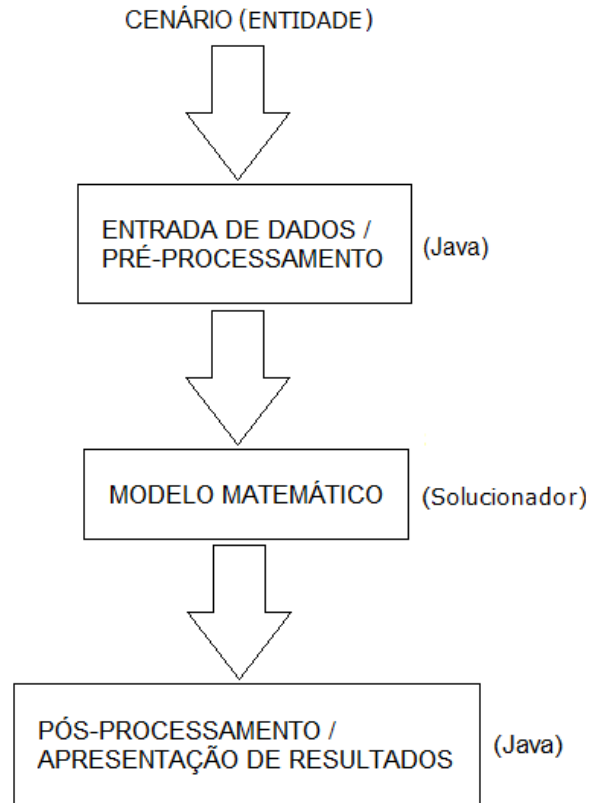
### 3.4 FLUXO EXECUTAR SOLUCIONADOR

O fluxo de execução do solucionador é o principal do *software*. A figura 7 ilustra a sequência de eventos. Após configurar o cenário (*model*) na tela de gerenciamento, a próxima etapa é solucioná-lo. O cenário escolhido passa por uma etapa de pré-processamento, em que primeiramente é feita uma avaliação inicial apontando através de avisos ao usuário problemas simples de serem encontrados, caso existam. Os problemas simples dizem respeito a dados do cenário que o tornam infactível. As seguintes avaliações são feitas:

- Se algum serviço está em uma distância maior que 320 km do depósito, que o tornaria fora da região metropolitana de Curitiba, e impossibilitaria algum veículo de executar o serviço e voltar para o depósito em menos de 8 horas de viagem, assumindo uma velocidade média de 80 km/h.
- Se alguma coleta ou entrega possui peso (volume) superior a dos veículos do cenário. Nenhum veículo poderia transportar o produto sem estourar sua capacidade máxima de peso (volume).
- Se a soma de todo o peso (volume) de todas as entregas é superior à soma de todas as capacidades máximas de peso (volume) dos veículos. Não é possível carregar todas as entregas nos veículos sem estourar suas capacidades máximas. O mesmo vale para as coletas.
- Se as janelas de disponibilidade dos clientes estão mal formadas. As janelas de tempo são realocadas para não ficarem sobrepostas e estarem dentro do tempo de operação do depósito.

Em seguida, ainda nessa etapa, gera-se o arquivo de modelo de dados (.dat) que será utilizado como entrada para o solucionador. Esse arquivo descreve para o modelo matemático como o cenário está configurado através de uma linguagem compreendida pelo IBM ILOG CPLEX.

A próxima etapa é a execução do modelo. O arquivo VRPMPDTW.mod, contido por padrão dentro da pasta **models** do diretório base, contém a descrição completa do



**Figura 7: Fluxo de Execução do Solucionador.**

**Fonte: Autoria própria.**

modelo matemático (variáveis, restrições e função objetivo) utilizando a linguagem do solucionador IBM ILOG CPLEX. Dependendo do número de serviços, número de veículos e das janelas de tempo, esta etapa pode demorar alguns minutos para o solucionador encontrar alguma solução factível, se existente.

Caso exista uma solução factível, o resultado é pós-processado retirando as informações consideradas importantes obtidas pelo solucionador e, então, montando a entidade de solução com todas as rotas de todos os veículos presentes no cenário. Utilizando o serviço REST do Bing *Maps* são gerados mapas com o caminho percorrido por cada veículo. As informações e os mapas, por fim, são apresentados em uma tela ao usuário.

### 3.5 CAMADA DE MODELO

A camada de modelo (*model*) contém as entidades do sistema. Entidades são representações de objetos concretos do mundo real ou conceitos abstratos que são modelados no paradigma orientado a objetos como **classes**. Uma entidade pode ser composta

de outras entidades a fim de abstrair seu conceito (FREEMAN, 2004). Como, por exemplo, uma entidade time de futebol é formado por uma associação de várias entidades jogadores.

Duas entidades fundamentais ao sistema desenvolvido são *Scenario* e *Solution*. Ambas possuem uma complexa associação de entidades que as compõem e, por esse motivo, serão descritas separadamente. A nomenclatura utilizada para todas as entidades está em inglês e cada palavra é iniciada com letra maiúscula.

A figura 8 ilustra as associações e a multiplicidade existente entre as entidades que compõe a entidade *Scenario*. Para entender melhor cada entidade é preciso entender o que representa cada uma delas:

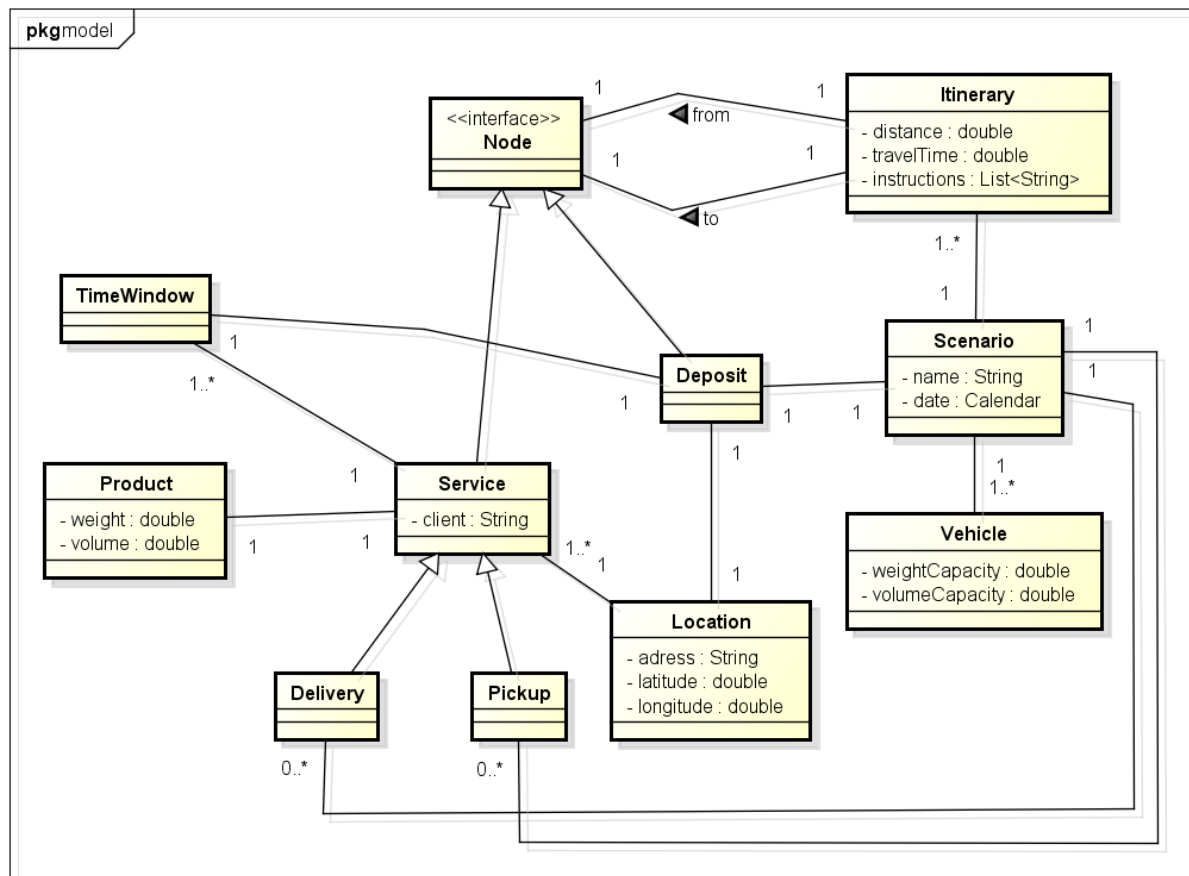


Figura 8: Associações que compõe a entidade Scenario.

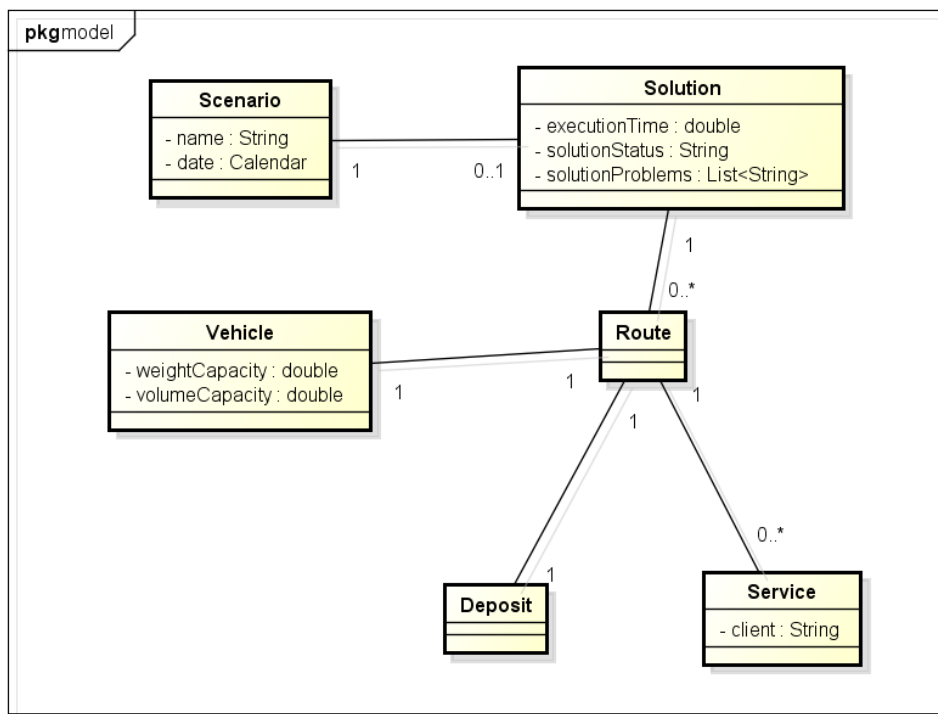
Fonte: Autoria Própria.

- **Vehicle:** representa um veículo. Possui como propriedades valores de capacidade máxima de peso (*weightCapacity*) e de volume (*volumeCapacity*).

- **TimeWindow:** conceito abstrato de janela de tempo. Cada janela possui um horário inicial e um final.
- **Product:** representa um produto. As propriedades são peso e volume.
- **Location:** é uma localização, uma posição geográfica. Tem como valores o endereço, a latitude e a longitude do lugar.
- **Deposit:** representa o depósito do cenário. Um depósito possui um endereço (*Location*) e opera durante um período do dia sem pausas (*TimeWindow*).
- **Service:** representa um serviço que possui um cliente (*client*) como propriedade. O cliente está associado a um endereço (*Location*), um produto (*Product*) e a uma ou mais janelas de disponibilidade (*TimeWindow*) durante o dia.
- **Delivery/Pickup:** simboliza uma entrega/coleta. A entrega/coleta é uma especialização da entidade serviço, portanto possui as mesmas associações e propriedades dela.
- **Node:** representa um nó, possui uma posição geográfica. O nó é uma generalização das entidades *Deposit* e *Service* que possuem uma localização (*Location*) associada.
- **Itinerary:** diz respeito ao itinerário entre dois nós. Descreve como ir do nó A (*from Node*) ao nó B (*to Node*).
- **Scenario:** representa a entidade cenário. As propriedades são data (*date*) e nome (*name*). Um cenário é composto por um ou mais veículos (*Vehicle*), nenhuma ou muitas coletas (*Pickup*), nenhuma ou muitas entregas (*Delivery*), um depósito central (*Deposit*) e para cada dois nós (coletas, entregas e depósito) está associado um itinerário (*Itinerary*).

A figura 9 apresenta as entidades que compõe a entidade *Solution*. Algumas já foram descritas anteriormente, as novas são a seguir elencadas:

- **Route:** representa uma rota. Uma rota está associada a um veículo (*Vehicle*), a um depósito (*Deposit*) e a nenhum ou muitos serviços (*Service*) do cenário.
- **Solution:** é a entidade que representa a solução do cenário. Está associada a um cenário (*Scenario*) e a uma ou mais rotas (*Route*), sendo o número de rotas igual ao número de veículos do cenário. Veículos não movimentados possuem uma rota do depósito para o depósito. Uma solução contém dados de execução do solucionador



**Figura 9: Entidades Solution.**

**Fonte: Autoria Própria.**

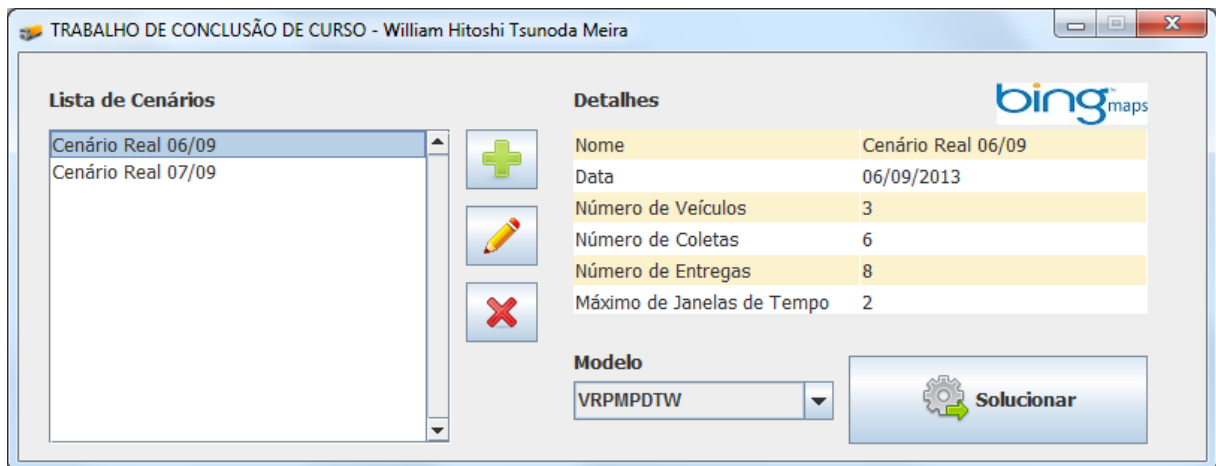
como tempo gasto (*executionTime*) e o estado da solução (*solutionStatus*) que pode ser ótima, com tempo limite excedido ou infactível. A solução também pode ser relaxada, ou seja, não foi possível encontrar uma solução ótima sem quebrar alguma restrição de capacidade ou de tempo. Neste caso, a lista de problemas (*solutionProblems*) é preenchida verificando quais variáveis de relaxação foram violadas.

### 3.6 CAMADA DE VISÃO E CONTROLE

A camada de visão (*view*) são as telas do sistema e representam o estado das entidades da camada modelo. Foram desenvolvidas utilizando as bibliotecas gráficas *Swing* e *AWT* da linguagem Java. A camada de controle está relacionada às ações do usuário e são chamadas através de interações com os objetos presentes na interface como botões, menus suspensos ou caixas de texto. Ao comando, o controle “ouve” (*listeners*) a requisição e organiza os dados necessários para o modelo executar a lógica e as alterações necessárias no modelo.

### 3.6.1 TELA DE GERENCIAMENTO DE CENÁRIOS

Quando o programa inicia a tela inicial é a de gerenciamento de cenários (figura 10). Do lado esquerdo, encontra-se uma lista com todos os cenários cadastrados. Alterando-se a seleção, um controle atualiza a tabela do lado direito com os detalhes do novo cenário selecionado. Os detalhes permitem avaliar de forma quantitativa o tamanho do cenário.



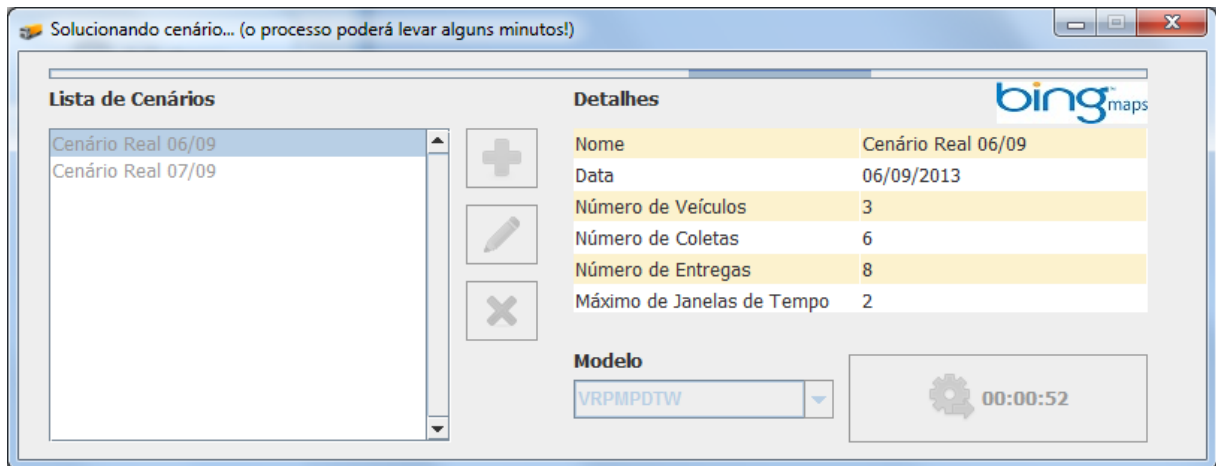
**Figura 10: Tela de Gerenciamento de Cenários.**

**Fonte: Autoria Própria.**

A tela possui quatro botões, cada um deles possui um controlador associado. O botão com o ícone de “mais” direciona o usuário para a tela de criação de um novo cenário, a tela é descrita na seção 3.6.2. O botão com ícone de “lápis” redireciona, caso exista algum cenário e um deles esteja selecionado, para a tela de edição de cenário, também descrito na seção 3.6.2. O botão com ícone de “x” exclui o cenário selecionado. O botão maior, no canto inferior direito, executa o solucionador para o cenário selecionado.

Enquanto o cenário está sendo solucionado a tela de gerenciamento assume o estado da figura 11, desabilitando todos os botões. Um indicador gráfico é apresentado para mostrar ao usuário que o processo de solução está em andamento. Internamente, um contador é iniciado para verificar o tempo computacional gasto com a solução do cenário. Após obter a solução, caso nenhum erro tenha ocorrido, o usuário é redirecionado para a tela de solução, descrita na seção 3.6.3.





**Figura 11: Tela de Gerenciamento de Cenários executando Solucionador.**

**Fonte: Autoria Própria.**

### 3.6.2 TELA DE CRIAÇÃO OU EDIÇÃO DE CENÁRIOS

A tela para criar um novo cenário é ilustrada na figura 12. Nesta tela todas as configurações do novo cenário devem ser definidas. Para o caso de editar um cenário os campos são preenchidos com os dados cadastrados. A figura 13 apresenta um exemplo de edição de cenário. A tela para os dois casos possui igual estrutura. Na parte superior define-se o nome e a data do novo cenário e logo abaixo a tela é dividida em quatro partes que são os componentes que formam o cenário: depósito, veículos, entregas e coletas.

No painel do depósito central define-se o endereço e seus horários de abertura e fechamento. Um botão com ícone de “lupa” ativa o controle para verificar o endereço inserido, geocodificando o endereço em latitude e longitude utilizando o serviço REST do Bing Maps. Caso exista algum problema com o endereço, um aviso é dado ao usuário.

O painel de veículos lista em uma tabela todos os veículos existentes no cenário. Três botões estão localizados na lateral deste painel. De cima para baixo, o primeiro edita o veículo selecionado, o segundo adiciona um novo veículo e o terceiro exclui o veículo selecionado. O primeiro e o segundo botão redirecionam o programa para a tela da figura 14, onde os dados que definem um veículo podem ser manipulados.

Os painéis inferiores listam os serviços de entrega e coleta do cenário em uma tabela. Também possuem três botões nas laterais da cada painel, porém para editar ou adicionar um serviço o programa é redirecionado para a janela da figura 15. Nela são configurados o nome do cliente, o endereço do serviço, o peso e o volume do produto a ser

**Figura 12: Tela de Criação de Novo Cenário.**

**Fonte: Autoria Própria.**

entregue ou coletado. Possibilita-se ao usuário inserir até quatro janelas de disponibilidade do cliente. Em linhas gerais, contudo, os clientes costumam apresentar até duas janelas de disponibilidade. Por exemplo, uma janela única das 8h00 às 17h00 ou duas janelas: 8h00 às 12h00; 13h00 às 17h00. Em termos práticos, não é operacionalmente adequado um cliente restringir os horários de entrega/coleta a mais de quatro intervalos durante o dia útil.

Ao inserir um novo serviço o endereço é primeiramente geocodificado e depois se calcula todas as rotas entre o novo serviço e todos os outros nós (depósito mais serviços) já cadastrados e vice-versa. Salienta-se que entre dois nós da rede, a rota de ida pode não ser a mesma da rota no sentido inverso. Assim, forma-se uma matriz de itinerários entre todos os pontos. Cada itinerário está associado a dois nós e possui a distância, o tempo de viagem e a sequência de instruções para ir de um ponto para outro. As rotas entre dois pontos são calculadas utilizando o serviço REST do Bing *Maps*.

**Nome do Cenário**  
Cenário Real 06/09

**Data do Cenário**  
06/09/2013

**Depósito Central**

Endereço:  
Rua Luiz Valenza, Curitiba - PR, 81350-080, Brasil

Latitude: -25.4925251 Longitude: -49.3326454

Horário de Abertura: 08:00

Horário de Fechamento: 17:00

**Veículos**

Nome	Peso (kg)	Volume (m³)
Caminhão 3/4(1)	3000,0	23,0
Caminhão 3/4(2)	3000,0	23,0
Caminhão 3/4(3)	3000,0	23,0

**Entregas**

Cliente	Endereço	Peso (kg)	Volume (m³)	Disponibilidade
Narciso	Rua João Eugênio Baptista, Curitiba - PR, 8...	374,0	3,0	12:00-17:00
João Pinhais	Rua Sudão, Pinhais - PR, 83320-030, Brasil	352,0	2,0	08:00-17:00
Lorenço	Rua Ponta Grossa, São José Dos Pinhais - ...	132,0	2,0	13:00-17:00
Leandro Augus...	Rua Arapongas, 583, Pinhais - PR, 83325-3...	163,0	3,0	08:00-11:00 15:00-17:00

**Coletas**

Cliente	Endereço	Peso (kg)	Volume (m³)	Disponibilidade
Luiz Boqueirão	Rua Bom Jesus de Iguape, Curitiba - PR, 8...	360,0	2,0	09:30-12:00 13:00-17:00
Lourenço	Rua Mateus Leme, 3980, Curitiba - PR, 822...	480,0	4,0	08:00-12:00
Guilherme	Rua Luiz Tramontin, 300, Curitiba - PR, 81...	108,0	2,0	08:00-17:00
Ferreira	Rua Edmundo Ferreira, 300, Araucária - P...	1422,0	10,0	10:00-15:00

Salvar Cancelar

Figura 13: Tela de Edição de Cenário.

Fonte: Autoria Própria.

**Adicionar Veículo**

Nome:

Capacidade de Peso (kg):

Capacidade de Volume (m³):

Salvar Cancelar

Figura 14: Tela de Criar Veículo.

Fonte: Autoria Própria.

O botão “salvar” no canto inferior da tela da figura 13 cria toda a estrutura XML com os dados definidos. Este arquivo XML é armazenado na pasta de cenários, que estará

Adicionar Entrega

Cliente:

Endereço:

Peso(kg):  Volume(m<sup>3</sup>):

Intervalos de Tempo Disponíveis para Atendimento

Intervalo 1:

Intervalo 2:

Intervalo 3:

Intervalo 4:

**Figura 15: Tela de Criar Serviço.**

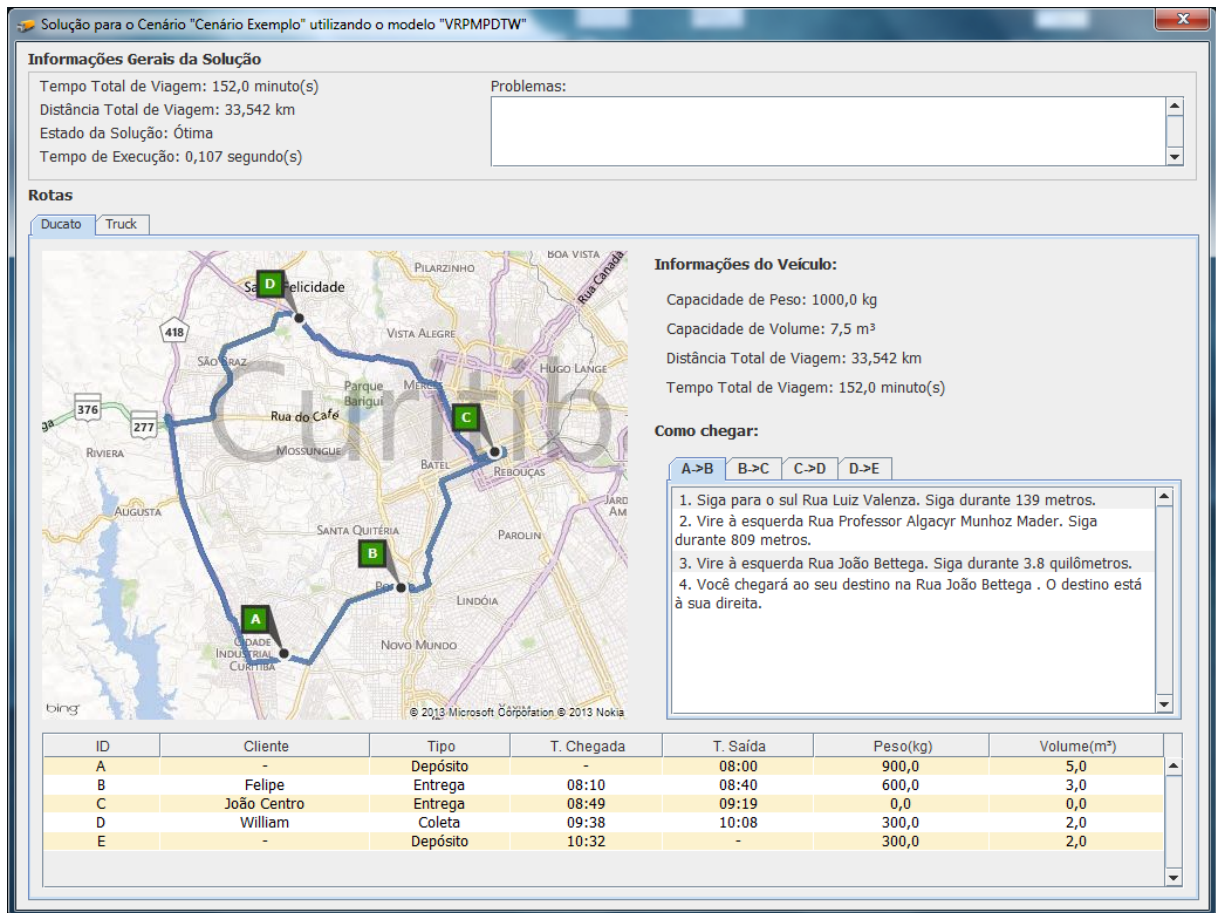
**Fonte: Autoria Própria.**

disponível ao retornar para a tela de gerenciamento de cenários. Caso o cenário seja editado, o programa fornece a opção de salvar o cenário como novo ou apenas editar as alterações feitas. A qualquer momento é possível cancelar todas as alterações clicando no botão “cancelar”, também localizado no canto inferior da tela da figura 13.

### 3.6.3 TELA DE SOLUÇÃO DO CENÁRIO

A figura 16 é a tela de solução do cenário que representa o modelo *Solution*. No painel superior do lado esquerdo estão dispostas as informações gerais da solução: tempo total de viagem do cenário, distância total de viagem do cenário, estado da solução e o tempo de execução do modelo matemático. Caso o estado da solução seja relaxada, ou seja, alguma restrição teve que ser quebrada para encontrar uma solução factível, estarão destacados, do lado direito, os problemas que ocorreram. A figura 17 mostra uma solução de um cenário que o veículo “Ducato” teria que ultrapassar em 100 kg sua capacidade máxima recomendada (1000 kg) durante seu percurso.

As informações das rotas estão dispostas no painel abaixo das Informações Gerais da Solução, separadas por abas, sendo cada aba um veículo do cenário. Os mapas com as rotas traçadas foram obtidas utilizando-se o serviço REST de geração imagens estáticas do Bing *Maps*. Informações do veículo, distância total percorrida, tempo total de viagem



**Figura 16: Tela de Solução do Cenário.**

**Fonte: Autoria Própria.**

estão dispostas ao lado do mapa, assim como as instruções de viagem desde o ponto inicial até o ponto final do percurso. A área inferior do painel contém uma tabela com informações de cada ponto da rota, como horário de chegada e de saída de cada cliente e valores de peso e volume ocupados no veículo em cada momento. Valores que quebram restrições de capacidade ou janela de tempo são destacadas em vermelho, como pode ser observado na primeira linha da tabela da figura 17.

### 3.7 FORMULAÇÃO DO MODELO MATEMÁTICO

O modelo matemático formulado procura representar o problema da empresa caracterizada na seção 1.1. Utilizou-se o ambiente de desenvolvimento do ILOG CPLEX *Optimization Studio* 12.5, cuja modelagem é feita em uma linguagem própria denominada OPL (*Optimization Programming Language*). Mais informações sobre a ferramenta da IBM podem ser encontradas em ILOG (2013). O modelo foi elaborado em PLIM e tem

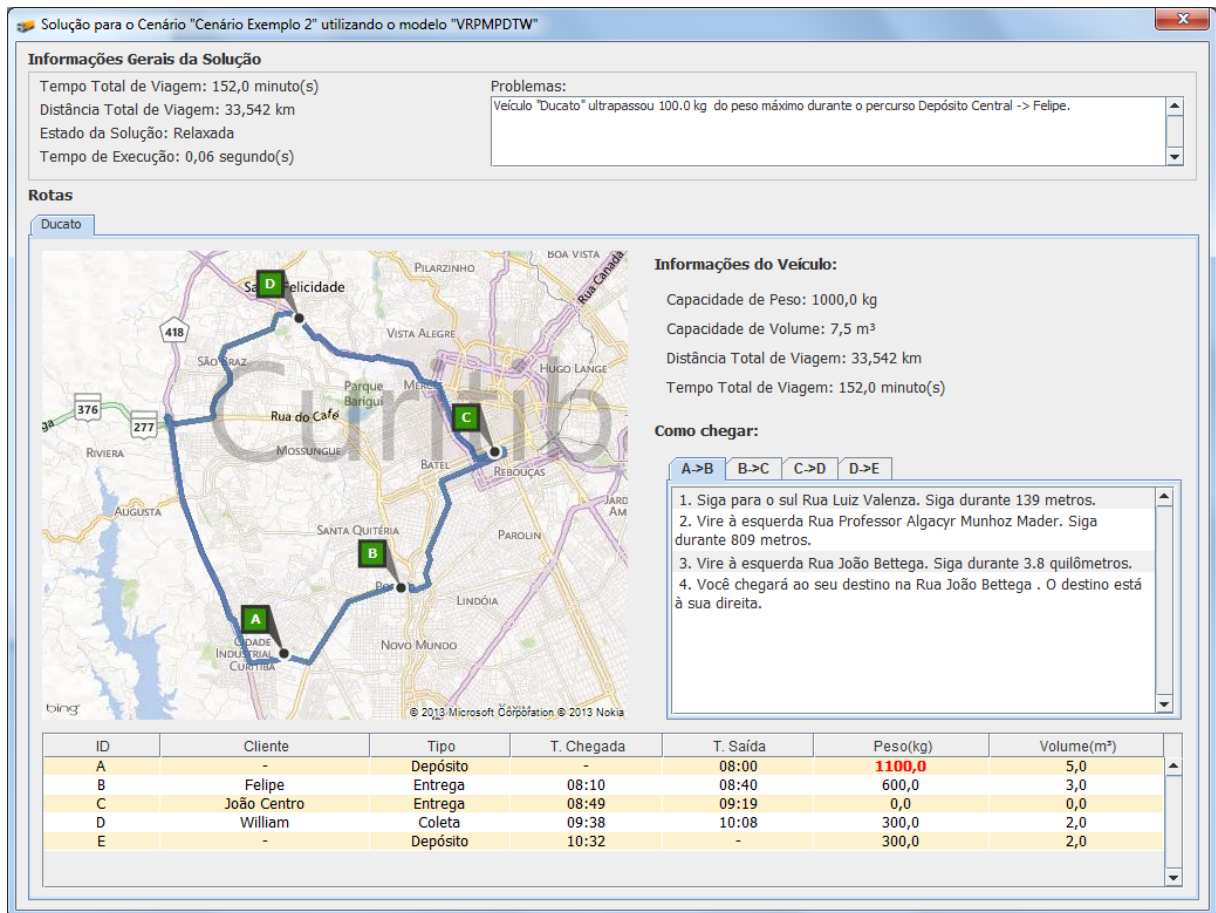


Figura 17: Tela de Solução do Cenário com Erro.

Fonte: Autoria Própria.

como objetivo principal minimizar o tempo de viagem gasto pelos veículos. Caso alguma violação de capacidade de peso, capacidade de volume ou desrespeito a janelas de disponibilidade de algum cliente ocorra, pode-se identificar o problema verificando as variáveis de relaxação inseridas no modelo.

A seção 3.7.1 contém a notação utilizada para conjuntos e índices do modelo. A notação para os parâmetros e as variáveis é definida nas seções 3.7.2 e 3.7.3, respectivamente. A seção 3.7.4 descreve a função objetivo e, por fim, na seção 3.7.5 as restrições do modelo são apresentadas. O arquivo do modelo matemático e do modelo de dados utilizado pelo IBM ILOG CPLEX pode ser encontrado nos apêndices C e B.

### 3.7.1 NOTAÇÃO DE CONJUNTOS E ÍNDICES

Os seguintes conjuntos fazem parte do modelo:

<i>Nos</i>	$\{1..NumeroNos\}$	Nós do cenário. É um conjunto que contém números inteiros de 1 até número total de nós do cenário ( <i>NumeroNos</i> ). O nó 1 é o depósito do sistema. Salienta-se que nó pode ser um depósito, um ponto de coleta/entrega.
<i>NumJT</i>	$\{1..MaximoJanelasTempo\}$	Janela de Tempo de cada nó. É um conjunto que contém números inteiros de 1 até o número máximo de janelas de tempo do cenário ( <i>MaximoJanelasTempo</i> ). A cardinalidade do conjunto é igual à do nó com maior número de janelas de tempo.
<i>Veiculos</i>	$\{1..NumeroVeiculos\}$	Veículos do cenário. É um conjunto que contém números inteiros de 1 até o número total de veículos ( <i>NumeroVeiculos</i> ) que podem ser utilizados no cenário.

E os seguintes índices foram utilizados:

$i, j, k$	$\in Nos$	Nó. Onde, $i, j$ ou $k=1$ é o nó depósito.
$jt$	$\in NumJT$	Janela de Tempo.
$v$	$\in Veiculos$	Veículo.

### 3.7.2 NOTAÇÃO DE PARÂMETROS

Este item detalha a notação dos parâmetros utilizados para formular o modelo matemático. Todos os parâmetros começam com letra maiúscula e seguem o estilo *camel case*.

$CapPesoMax_v$	Capacidade de peso (kg) de cada veículo $v$ .
$CapVolMax_v$	Capacidade volumétrica ( $m^3$ ) de cada veículo $v$ .
$InicioJT_{i,jn}$	Início de cada janela de tempo $jn$ para um determinado nó $i$ em minutos. Todos os valores têm como base o início de operação do depósito, ou seja, eles são contados a partir desse valor. Se o nó possui menos janelas de tempo que o tamanho de $NumJT$ , os valores serão os mesmos do último início de janela inserido. Assim, um início de janela em 0 minuto significa que o cliente pode começar a receber, em tese, a partir do horário de abertura do depósito.
$FimJT_{i,jn}$	Fim de cada janela de tempo $jn$ para um determinado nó $i$ em minutos. Todos os valores têm como base o início de operação do depósito, ou seja, eles são contados a partir desse valor. Se o nó possui menos janelas de tempo que o tamanho de $NumJT$ , os valores serão os mesmos do último fim de janela inserido. Assim, um fim de janela especificado em 540 minutos, por exemplo, significa que o cliente pode receber até o tempo de 540 minutos (9 horas) após a abertura do depósito.
$QPesoColeta_i$	Peso (kg) do produto para coleta em cada nó $i$ . Caso o nó $i$ seja um depósito ou uma entrega o valor é 0.
$QPesoEntrega_i$	Peso (kg) do produto para entrega em cada nó $i$ . Caso o nó $i$ seja um depósito ou uma coleta o valor é 0.
$QVolColeta_i$	Volume ( $m^3$ ) do produto para coleta em cada nó $i$ . Caso o nó $i$ seja um depósito ou uma entrega o valor é 0.
$QVolEntrega_i$	Volume ( $m^3$ ) do produto para entrega em cada nó $i$ . Caso o nó $i$ seja um depósito ou uma coleta o valor é 0.
$TempoServico_i$	Tempo gasto, em minutos, para executar o serviço do nó $i$ . Se $i=1$ (depósito), o tempo de serviço é 0. Para o problema em análise, se $i>1$ , o tempo de serviço é de 30 minutos. Este valor é o tempo médio, segundo informações da empresa, que é gasto para carregar ou descarregar um serviço de coleta ou entrega, respectivamente.
$TempoViagem_{i,j}$	Tempo de viagem, em minutos, de um nó $i$ para um nó $j$ .



Outros três parâmetros foram utilizados para formulações *Big-M*, um para cada situação, conforme equações 3.1 a 3.3 a seguir especificadas. Como explanado na seção 2.2.2 o valor de  $M$  não deve, em tese, ser muito superior ao esperado por questões de eficiência computacional. Devido a este fato, um cálculo determina o valor máximo possível para a variável contínua de cada uma das três situações, e para garantir a validade do parâmetro obtido, uma constante de multiplicação (3) é aplicada.

A equação 3.1 define o parâmetro  $MQ$ . Utilizada para formulações *Big-M* que envolvam quantidade de peso dos serviços.

$$MQ = 3 \times \sum_{i \in Nos} (QPesoColeta_i + QPesoEntrega_i) \quad (3.1)$$

A equação 3.2 define o parâmetro  $MV$ . Utilizado para formulações *Big-M* que envolvam quantidade de volume dos serviços.

$$MV = 3 \times \sum_{i \in Nos} (QVolColeta_i + QVolEntrega_i) \quad (3.2)$$

A equação 3.3 define o parâmetro  $MJ$ . Utilizada para as formulações *Big-M* que envolvam tempo.

$$MJ = 3 \times \max_{i \in Nos, jt \in NumJT} (FimJT_{i,jt}) \quad (3.3)$$

### 3.7.3 NOTAÇÃO DE VARIÁVEIS

Esta seção detalha a notação das variáveis utilizadas para formular o modelo matemático. Todas as variáveis começam com letra minúscula e seguem o estilo *camel case*.

$emJanela_{i,jt}$	Variável binária que indica em qual janela de tempo $jt$ iniciou-se o atendimento do nó $i$ .
$quantPeso_{i,j,v}$	Variável contínua que indica a carga em peso (kg) do veículo $v$ durante o percurso do nó $i$ para o nó $j$ .
$quantVolume_{i,j,v}$	Variável contínua que indica a carga em volume ( $m^3$ ) do veículo $v$ durante o percurso do nó $i$ para o nó $j$ .
$relaxJanela_{i,v}$	Variável contínua de relaxação que indica uma violação, em minutos, da janela de tempo no nó $i$ pelo veículo $v$ .
$relaxPeso_{i,j,v}$	Variável contínua de relaxação que indica o excesso de carga em peso (kg) presente no veículo $v$ quando sai do nó $i$ para o nó $j$ .
$relaxVolume_{i,j,v}$	Variável contínua de relaxação que indica o excesso de carga em volume ( $m^3$ ) presente no veículo $v$ quando sai do nó $i$ para o nó $j$ .
$rota_{i,j,v}$	Variável binária que indica, caso assuma o valor 1, a existência da rota entre $i$ e $j$ percorrida pelo veículo $v$ .
$tempoChegada_{i,v}$	Variável inteira que indica, em minutos, o instante de tempo de chegada do veículo $v$ no nó $i$ . Este tempo é contado a partir do horário de abertura do depósito.

### 3.7.4 FUNÇÃO OBJETIVO

Em linhas gerais, o objetivo principal do modelo é minimizar o tempo total de viagem de todos os veículos. No estudo realizado, o tempo de viagem foi considerado diretamente proporcional à distância percorrida. Portanto, dentro do escopo avaliado no trabalho, pode-se dizer que a distância total percorrida é também minimizada. Além disso, objetiva-se minimizar eventuais violações de peso, volume e janelas temporais.

A equação 3.4 é a função objetivo do modelo. Esta função é composta por quatro termos principais. O primeiro prioriza a minimização dos tempos de chegada dos veículos ao depósito central. O segundo, terceiro e quarto termos penalizam violações de peso, volume e janelas de tempo. A cada violação ocorrida, que está associada às variáveis de relaxação, é adicionado um peso multiplicado por  $MJ$ . O parâmetro  $MJ$  foi escolhido apenas por representar um valor relativamente elevado em relação ao primeiro termo e,

com isso, priorizar uma solução sem nenhuma violação. Repare que quebrar em uma unidade a capacidade máxima de peso ou volume de algum veículo é algumas vezes pior que ocorrer um atraso ou adiantamento de uma unidade de tempo de algum serviço. O primeiro e o último termo estão quantificados em minutos, porém o segundo e o terceiro estão em quilos e metros cúbicos, respectivamente, por esse motivo existe uma diferença nos pesos aplicados. Assim, o modelo, caso seja necessário, dará prioridade a uma violação da restrição de janela de tempo em relação a violações de capacidade. De fato, em função dos fatores de ponderação adotados, violações tendem a ser aceitas somente em casos de inconsistência encontradas na solução.

$$\begin{aligned}
\text{Minimizar } z = & \sum_{v \in \text{Veiculos}} \text{tempoChegada}_{1,v} \\
& + 10 \times MJ \times \sum_{\substack{i,j \in \text{Nos} \\ v \in \text{Veiculos}}} \text{relaxPeso}_{i,j,v} \\
& + 100 \times MJ \times \sum_{\substack{i,j \in \text{Nos} \\ v \in \text{Veiculos}}} \text{relaxVolume}_{i,j,v} \\
& + MJ \times \sum_{\substack{i \in \text{Nos} \\ v \in \text{Veiculos}}} \text{relaxJanela}_{i,v}
\end{aligned} \tag{3.4}$$

### 3.7.5 RESTRIÇÕES

O modelo matemático está sujeito às restrições 3.5 a 3.21 a seguir especificadas:

1. Todos os veículos devem iniciar a rota a partir do depósito, ou seja, a partir do nó 1, conforme a restrição 3.5.

$$\sum_{j \in \text{Nos}} \text{rota}_{1,j,v} = 1, \quad \forall v \in \text{Veiculos} \tag{3.5}$$

2. Todos os veículos terminam a rota no depósito, ou seja, no nó 1, conforme a restrição 3.6.

$$\sum_{i \in \text{Nos}} \text{rota}_{i,1,v} = 1, \quad \forall v \in \text{Veiculos} \tag{3.6}$$

3. Todos os clientes devem ser atendidos. Os veículos chegam uma única vez em um

cliente  $j$ , conforme a restrição 3.7.

$$\sum_{\substack{i \in Nos, i \neq j \\ v \in Veiculos}} rota_{i,j,v} = 1, \quad \forall j \in Nos \mid j > 1 \quad (3.7)$$

4. A restrição 3.8 estabelece que se um veículo chega a um cliente deve sair desse cliente para outro nó  $k$ . Garante que cada cliente  $j$  é visitado somente uma vez por um veículo  $v$  proveniente da localização  $i$ , podendo  $i$  representar um outro cliente ou o depósito.

$$\sum_{i \in Nos, i \neq j} rota_{i,j,v} - \sum_{k \in Nos, k \neq j} rota_{j,k,v} = 0, \quad \forall j \in Nos, \forall v \in Veiculos \quad (3.8)$$

5. Cálculo de tempo de chegada a cada nó  $j$ . Um serviço num nó só pode ser realizado depois do nó  $i$  (visitado imediatamente antes) ter sido servido e o veículo ter se deslocado até o nó  $j$ . A formulação *Big-M* estabelece a validade da restrição apenas se a rota de  $i$  para  $j$  existir. Se  $i=1$ , utiliza-se a restrição 3.9, caso contrário a restrição 3.10. Particularmente, a 3.9 estabelece o tempo de chegada a um nó  $j$  quando o veículo sai do depósito central.

$$tempoChegada_{j,v} \geq TempoServico_i + TempoViagem_{i,j} - MJ \times (1 - rota_{i,j,v}), \quad (3.9) \\ \forall v \in Veiculos, \forall i \in Nos, \forall j \in Nos \mid i \neq j, i = 1$$

$$tempoChegada_{j,v} \geq tempoChegada_{i,v} + TempoServico_i + TempoViagem_{i,j} \quad (3.10) \\ - MJ \times (1 - rota_{i,j,v}), \quad \forall v \in Veiculos, \forall i, j \in Nos \mid i \neq j, i > 1$$

6. A restrição 3.11 estabelece que o tempo de chegada só assume valor diferente de zero se existir uma rota ativa entre  $i$  e  $j$  de um veículo  $v$ .

$$tempoChegada_{j,v} \leq MJ \times \sum_{i \in Nos, i \neq j} rota_{i,j,v}, \quad \forall v \in Veiculos, \forall j \in Nos \quad (3.11)$$

7. O veículo  $v$  deve chegar a um cliente  $j$  depois do início da janela de tempo  $jt$ , se existir uma rota de  $i$  para  $j$  e se estiver dentro de uma janela  $jt$  válida, conforme a restrição 3.12.

$$tempoChegada_{j,v} \geq InicioJT_{j,jt} - (2 - emJanela_{j,jt} - \sum_{i \in Nos, i \neq j} rota_{i,j,v}) \times MJ \quad (3.12) \\ - relaxJanela_{j,v}, \quad \forall v \in Veiculos, \forall j \in Nos, \forall jt \in NumJT$$

8. O veículo  $v$  deve chegar a um cliente  $j$  antes do fim da janela de tempo  $jt$  válida, conforme a restrição 3.13.

$$\begin{aligned} \text{tempoChegada}_{j,v} &\leq \text{FimJT}_{j,jt} + \text{relaxJanela}_{j,v} + (1 - \text{emJanela}_{j,jt}) \times MJ, & (3.13) \\ &\forall v \in \text{Veiculos}, \forall j \in \text{Nos}, \forall jt \in \text{NumJT} \end{aligned}$$

9. Um cliente  $j$  é atendido dentro de uma de janela de tempo  $jt$ , conforme a restrição 3.14.

$$\sum_{jt \in \text{NumJT}} \text{emJanela}_{j,jt} \geq 1, \quad \forall j \in \text{Nos} \quad (3.14)$$

10. O veículo  $v$  sai do depósito com a carga para entregar aos clientes que constituem sua rota. Neste caso, considera-se a quantidade de carga em peso no nó  $i=1$ , conforme a restrição 3.15.

$$\begin{aligned} \sum_{j \in \text{Nos}, j > 1} \text{quantPeso}_{1,j,v} &= \sum_{i,j \in \text{Nos}, i \neq j} (\text{rota}_{i,j,v} \times Q\text{PesoEntrega}_j), & (3.15) \\ &\forall v \in \text{Veiculos} \end{aligned}$$

11. A quantidade de peso durante a rota de um veículo  $v$  não deve exceder sua capacidade máxima de peso, , conforme a restrição 3.16.

$$\begin{aligned} \text{quantPeso}_{i,j,v} &\leq \text{rota}_{i,j,v} \times \text{CapPesoMax}_v + \text{relaxPeso}_{i,j,v}, & (3.16) \\ &\forall i \in \text{Nos}, \forall j \in \text{Nos}, \forall v \in \text{Veiculos} \end{aligned}$$

12. A restrição 3.17 estabelece que o veículo  $v$  retorna ao depósito com a quantidade de peso igual à carga coletada nos clientes visitados.

$$\begin{aligned} \sum_{i \in \text{Nos}, i > 1} \text{quantPeso}_{i,1,v} &= \sum_{i,j \in \text{Nos}, i \neq j} (\text{rota}_{i,j,v} \times Q\text{PesoColeta}_j) & (3.17) \\ + \sum_{j \in \text{Nos}, j > 1} \text{quantPeso}_{1,j,v} &- \sum_{i,j \in \text{Nos}, i \neq j} (\text{rota}_{i,j,v} \times Q\text{PesoEntrega}_j), \quad \forall v \in \text{Veiculos} \end{aligned}$$

13. Balanço de massa dos nós deve ser respeitado. A quantidade de peso do veículo  $v$  ao sair do cliente  $j$  para  $k$  deve ser igual ao peso de saída do nó  $i$  mais o coletado ou menos o entregue no nó  $j$ , conforme a restrição 3.18.

$$\begin{aligned} \text{quantPeso}_{j,k,v} &\leq \text{quantPeso}_{i,j,v} - Q\text{PesoEntrega}_j + Q\text{PesoColeta}_j & (3.18) \\ + (1 - \text{rota}_{i,j,v} \times MQ), &\quad \forall i, j, k \in \text{Nos}, \forall v \in \text{Veiculos} \mid i \neq j, j \neq k, j > 1 \end{aligned}$$

14. O veículo  $v$  sai do depósito com a carga para entregar aos clientes que constituem sua rota. Neste caso, considera-se a quantidade de carga em volume no nó  $i=1$ , conforme a restrição 3.19.

$$\sum_{j \in Nos, j > 1} quantVolume_{1,j,v} = \sum_{i,j \in Nos, i \neq j} (rota_{i,j,v} \times QVolEntrega_j), \quad \forall v \in Veiculos \quad (3.19)$$

15. A quantidade de volume durante a rota de um veículo  $v$  não deve exceder sua capacidade máxima de volume, conforme a restrição 3.20.

$$\sum_{j \in Nos, j > 1} quantVolume_{1,j,v} = \sum_{i,j \in Nos, i \neq j} (rota_{i,j,v} \times QVolEntrega_j), \quad \forall v \in Veiculos \quad (3.20)$$

16. A restrição 3.21 estabelece que o veículo  $v$  retorna ao depósito com a quantidade de volume igual a carga coletada nos clientes visitados.

$$\begin{aligned} \sum_{i \in Nos, i > 1} quantVolume_{i,1,v} &= \sum_{i,j \in Nos, i \neq j} (rota_{i,j,v} \times QVolColeta_j) \quad (3.21) \\ + \sum_{j \in Nos, j > 1} quantVolume_{1,j,v} &- \sum_{i,j \in Nos, i \neq j} (rota_{i,j,v} \times QVolEntrega_j), \quad \forall v \in Veiculos \end{aligned}$$

17. Balanço de massa dos nós deve ser respeitado. A quantidade de volume do veículo  $v$  ao sair do cliente  $j$  para  $k$  deve ser igual ao volume de saída do nó  $i$  mais o coletado ou menos o entregue no nó  $j$ , conforme a restrição 3.22.

$$\begin{aligned} quantVolume_{j,k,v} &\leq quantVolume_{i,j,v} - QVolEntrega_j + QVolColeta_j \quad (3.22) \\ &+ (1 - rota_{i,j,v} \times MV), \quad \forall i, j, k \in Nos, \forall v \in Veiculos \mid i \neq j, j \neq k, j > 1 \end{aligned}$$

### 3.8 CONSIDERAÇÕES

Desenvolveu-se um *software* capaz de resolver o problema de roteamento de veículos semelhante ao da transportadora em estudo, caracterizado na seção 1.1. O programa permitiu a manipulação de cenários que, em conjunto com o solucionador IBM ILOG CPLEX, possibilitou resolvê-los buscando otimizar o tempo de viagem gasto para atender os serviços cadastrados.

O desenvolvimento atingiu os níveis desejados de acordo com os requisitos funcionais e não-funcionais estabelecidos (vide seção 3.1). A arquitetura e a organização do *software* se mostrou eficaz respeitando o fluxo de dados proposto. Por fim, a formulação

matemática para todos os cenários testados apresentou um tempo de resolução não proibitivo, dentro dos limites estabelecidos. Adicionalmente, as soluções obtidas, de acordo com avaliação de especialista da empresa, mostraram-se coerentes à operação do sistema em estudo.

## 4 RESULTADOS

A seção de apresentação dos resultados estuda dois cenários distintos. O primeiro possui poucos serviços, o que permite uma análise rápida do resultado esperado e o obtido. O segundo foi baseado em um cenário real de um dia de trabalho da transportadora, em que o número de serviços foi acima da média. Assim, o cenário apresentado é grande o suficiente para ilustrar que a análise prática do problema de roteamento de veículos em estudo é uma tarefa complicada para o especialista do sistema. Evidencia-se, desta forma, que o *software* proposto pode contribuir para a melhoria da prática operacional.

Nos estudos de caso realizados, adotaram-se dois procedimentos no intuito de preservar o sigilo dos dados da empresa: (i) os endereços de clientes e do depósito foram alterados; e, (ii) os valores de entregas/coletas a serem realizadas foram multiplicados por um fator de ponderação. Adicionalmente, a plataforma computacional utilizada nos experimentos é um Intel Core i5-2430M 2.4GHz com 4 GB de memória RAM DDR3 e sistema operacional *Windows 7*.

### 4.1 ESTUDO DE CASO 1

No cenário utilizado no estudo de caso 1, poucos serviços são demandados, o que permitiu uma comparação entre o resultado esperado e o obtido. Na sequência, algumas modificações foram realizadas nos dados de entrada a fim de impor condições onde violações de carga eram inerentes. O objetivo foi observar a atuação das variáveis de relaxação consideradas na modelagem matemática diante destas condições errôneas de entrada de dados.

#### 4.1.1 PARÂMETROS DE ENTRADA

O cenário é composto por 1 veículo, 2 entregas e 2 coletas, sendo que as janelas de tempo de todos os serviços são únicas (parâmetro *MaximoJanelasTempo=1*) e coincidentes com o tempo de operação do depósito central.



### Depósito Central

Endereço: Rua Luiz Valenza 30, Cidade Industrial, Curitiba - PR.

Horário de Operação: 08:00 - 17:00.

### Veículos

#### 1. Ducato

Capacidade de peso: 1000,0 kg.

Capacidade de volume: 7,5 m<sup>3</sup>.

### Entregas

#### 1. João Centro

Endereço: Avenida Manoel Ribas, 5920, Santa Felicidade, Curitiba - PR.

Peso do Produto: 300,0 kg.

Volume do Produto: 2,0 m<sup>3</sup>.

Disponibilidade: 08:00 - 17:00.

#### 2. Guilherme

Endereço: Rua Carmem Miranda, 9, São José dos Pinhais - PR.

Peso do produto: 265,0 kg.

Volume do produto: 3,0 m<sup>3</sup>.

Disponibilidade: 08:00 - 17:00.

### Coletas

#### 1. William

Endereço: Avenida Sete de Setembro, 3300, Centro, Curitiba - PR.

Peso do Produto: 450 kg.

Volume do Produto: 3,0 m<sup>3</sup>.

Disponibilidade: 08:00 - 17:00.

#### 2. Felipe

Endereço: Rua Olívio Domingos Leonardi, 41, Curitiba - PR.

Peso do Produto: 365,0 kg.

Volume do Produto: 4,0 m<sup>3</sup>.

Disponibilidade: 08:00 - 17:00.

A tabela 2 apresenta os valores calculados de tempo de viagem, em minutos, entre todos os nós do cenário. Este cálculo é realizado a partir do serviço de rotas da plataforma Bing *Maps*.

O número de restrições e variáveis presentes no cenário é apresentado na tabela 3.

**Tabela 2: Estudo de Caso 1: matriz de tempo de viagem em minutos.**

	Depósito	Entrega 1	Entrega 2	Coleta 1	Coleta 2
Depósito	0	22	33	20	20
Entrega 1	22	0	31	14	33
Entrega 2	30	31	0	20	23
Coleta 1	17	12	21	0	24
Coleta 2	15	31	26	26	0

Fonte: Autoria própria.

**Tabela 3: Dados do Estudo de Caso 1.**

<b>Restrições</b>	<b>238</b>
<b>Variáveis</b>	<b>142</b>
Binárias	30
Inteiras	5
Contínuas	107
<b>Coefficientes Diferentes de Zero</b>	<b>814</b>

Fonte: Autoria própria.

#### 4.1.2 SOLUÇÃO

Encontrou-se a solução ótima para o cenário em apenas 0,103 segundo com valor objetivo de 215 minutos e 70,132 quilômetros percorridos. A tela de solução da figura 18 detalha o resultado obtido.

Um único veículo do cenário carregou todas as entregas da sua rota. Os limites de peso e volume não foram ultrapassados em nenhuma parte do percurso, conforme pode ser observado na figura 18, na porção inferior desta figura. Devido às janelas de disponibilidade relativamente extensas, as quais duravam das 8h00 às 17h00, e percursos curtos, o veículo “Ducato” fez a rota mais curta para todos os pontos: depósito, João Centro, William, Guilherme, Felipe, depósito novamente.

#### 4.1.3 ANÁLISE

O cenário é simples e apresenta poucas limitações para o veículo. Devido a isso, era previsto que a solução ótima fosse realizar coletas e entregas na sequência de proximidade. A partir do caso base apresentado, foram realizados dois experimentos adicionais:

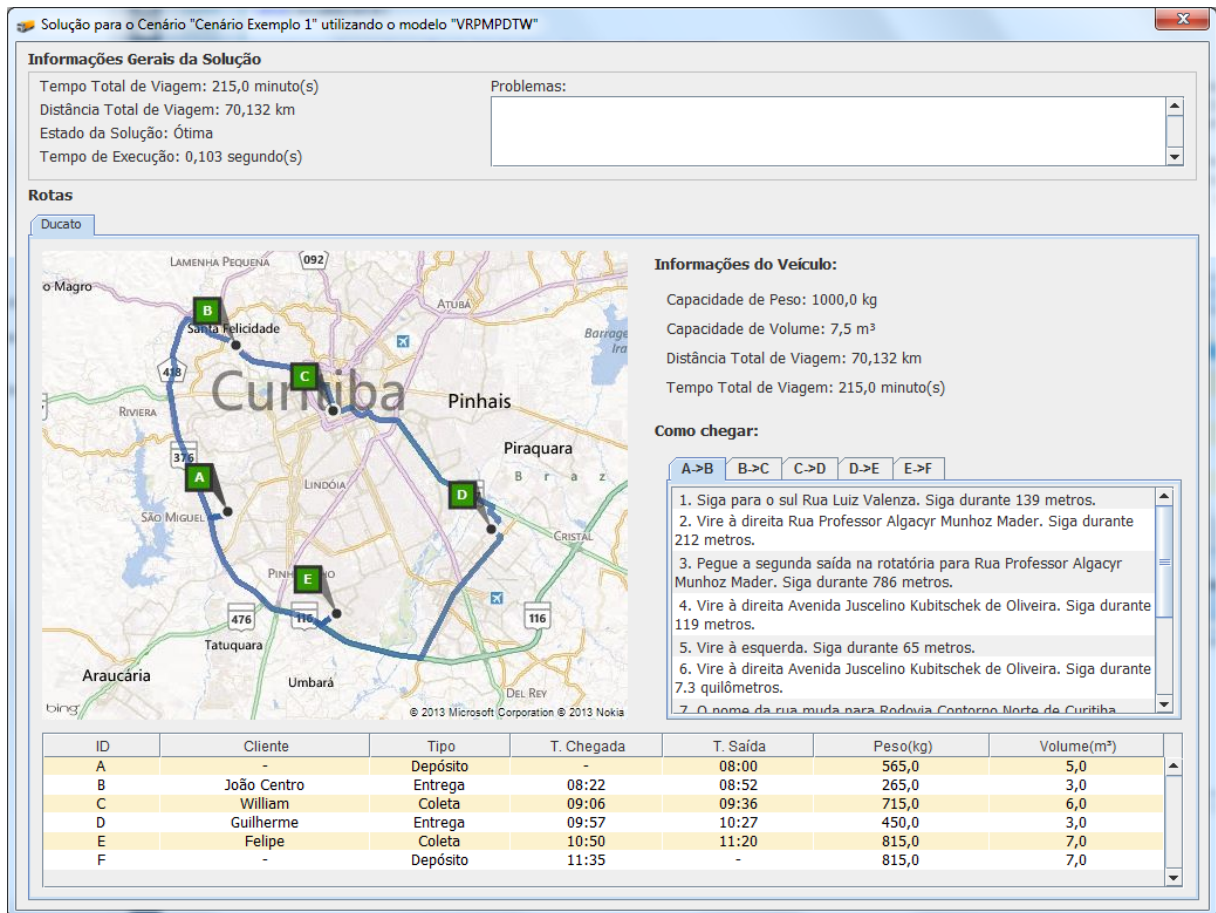


Figura 18: Solução do Estudo de Caso 1.

Fonte: Autoria Própria.

(i) alterou-se a janela de tempo do cliente William para entre 12h00 e 17h00; (ii) alterou-se o peso do produto do cliente “Guilherme” para 765 kg.

Na modificação (i), a solução proposta pelo modelo realiza a entrega para todos os demais clientes, deixando o cliente “William” como sendo o último a ser atendido, conforme ilustrado na figura 19.

Na modificação (ii), o cenário cairá num problema avaliado pelo pré-processamento, em que o peso das entregas se torna superior a soma das capacidades de peso de todos os veículos do cenário. No caso, o veículo “Ducato” seria sobrecarregado em 65 kg no nó depósito para poder tornar a solução relaxada. A figura 20 ilustra o resultado para essa alteração, destacando o erro da solução na caixa de texto “Problemas” e o nó onde o erro ocorreu é destacada na tabela inferior.

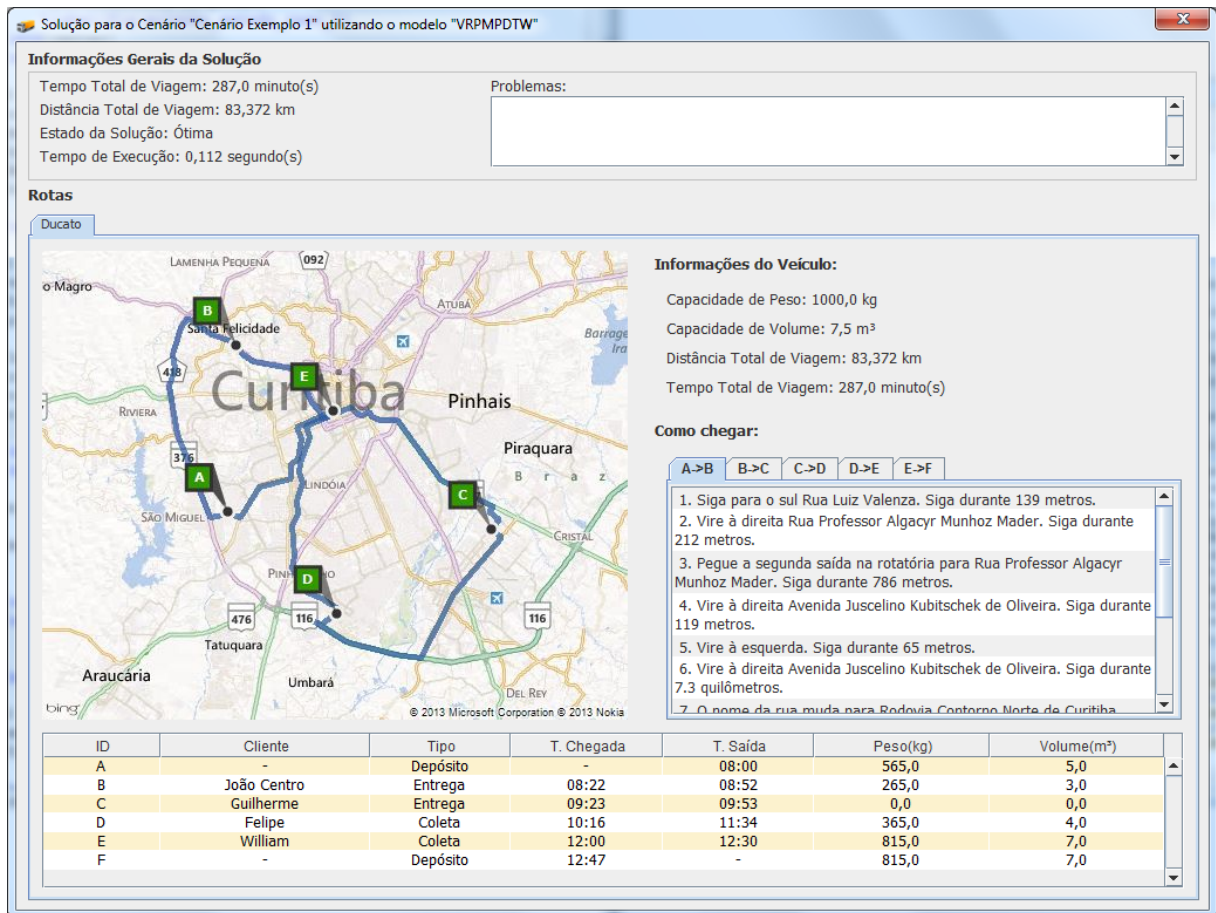


Figura 19: Solução do Estudo de Caso 1: Alteração (i).

Fonte: Autoria Própria.

## 4.2 ESTUDO DE CASO 2

Este cenário foi baseado em um dia real de trabalho da transportadora, com os mesmos veículos disponíveis na ocasião prática e o mesmo número de serviços. Conforme anteriormente mencionado no início do capítulo, modificações foram feitas para manter o sigilo dos dados da empresa. Será feita uma análise do resultado procurando entender os motivos das escolhas tomadas pelo solucionador.

### 4.2.1 PARÂMETROS DE ENTRADA

O cenário é composto por 3 veículos, 8 entregas e 6 coletas. No máximo duas janelas de tempo distintas são especificadas nos clientes. Por exemplo, das 8h00 às 12h00 e das 13h00 às 17h00.

#### Depósito Central

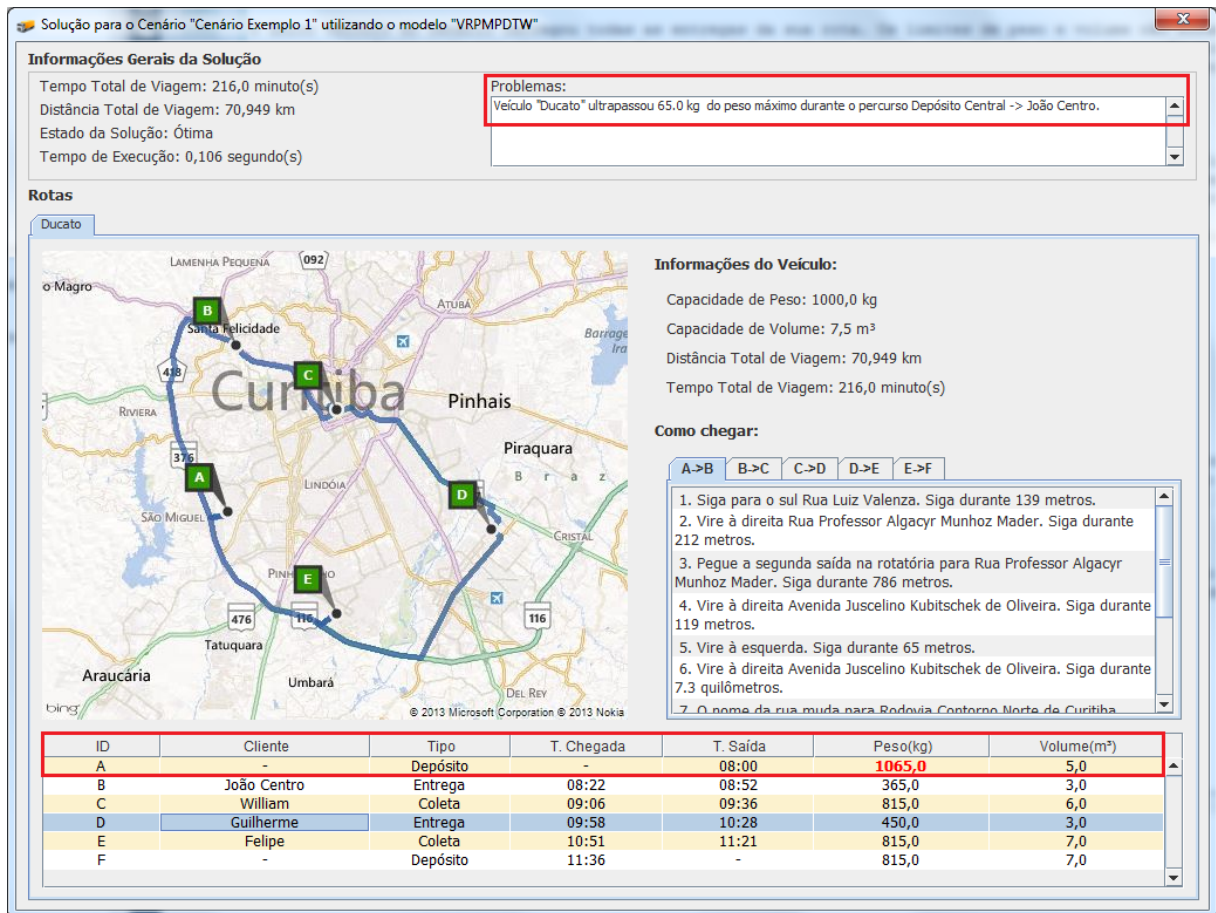


Figura 20: Solução do Estudo de Caso 1: Alteração (ii).

Fonte: Autoria Própria.

Endereço: Rua Luiz Valenza 30, Cidade Industrial, Curitiba - PR.

Horário de Operação: 08:00 - 17:00.

### Veículos

1. Caminhão 3/4(1)  
 Capacidade de peso: 3000,0 kg.  
 Capacidade de volume: 23,0 m<sup>3</sup>.
2. Caminhão 3/4(2)  
 Capacidade de peso: 3000,0 kg.  
 Capacidade de volume: 23,0 m<sup>3</sup>.
3. Caminhão 3/4(3)  
 Capacidade de peso: 3000,0 kg.  
 Capacidade de volume: 23,0 m<sup>3</sup>.

### Entregas

1. Narciso

Endereço: Rua João Eugênio Baptista, 30, Curitiba - PR.

Peso do Produto: 374,0 kg.

Volume do Produto: 3,0 m<sup>3</sup>.

Disponibilidade: 12:00 - 17:00.

2. João Pinhais

Endereço: Rua Sudão, 23, Pinhais - PR.

Peso do produto: 352,0 kg.

Volume do produto: 2,0 m<sup>3</sup>.

Disponibilidade: 08:00 - 17:00.

3. Loreço

Endereço: Rua Ponta Grossa, 42, São José dos Pinhais - PR.

Peso do produto: 132,0 kg.

Volume do produto: 2,0 m<sup>3</sup>.

Disponibilidade: 13:00 - 17:00.

4. Leandro Augusto

Endereço: Rua Araongas, 583, Pinhais - PR.

Peso do produto: 163,0 kg.

Volume do produto: 3,0 m<sup>3</sup>.

Disponibilidade: 08:00 - 11:00 15:00 - 17:00.

5. Gilberto

Endereço: Rua Alcides Darcanchy, 1, Curitiba - PR.

Peso do produto: 678,0 kg.

Volume do produto: 5,0 m<sup>3</sup>.

Disponibilidade: 08:00 - 17:00.

6. André C.

Endereço: Rua Rui Barbosa, 143, Campo Largo - PR.

Peso do produto: 835,0 kg.

Volume do produto: 5,0 m<sup>3</sup>.

Disponibilidade: 10:00 - 13:00 15:00 - 17:00.

7. Miguel Centro

Endereço: Avenida Sete de Setembro, 2500, Curitiba - PR.

Peso do produto: 230,0 kg.

Volume do produto: 2,0 m<sup>3</sup>.

Disponibilidade: 08:00 - 17:00.

8. Lenna S.

Endereço: Rua Pedro Zagonel, 473, Curitiba - PR.

Peso do produto: 850,0 kg.

Volume do produto: 6,0 m<sup>3</sup>.

Disponibilidade: 09:00 - 12:00 13:30 - 17:00.

## Coletas

1. Luiz Boqueirão  
Endereço: Rua Bom Jesus de Iguape, 12, Curitiba - PR.  
Peso do Produto: 360 kg.  
Volume do Produto: 2,0 m<sup>3</sup>.  
Disponibilidade: 09:30 - 12:00 13:00 - 17:00.
2. Lourenço  
Endereço: Rua Matheus Leme, 3980, Curitiba - PR.  
Peso do Produto: 480,0 kg.  
Volume do Produto: 4,0 m<sup>3</sup>.  
Disponibilidade: 08:00 - 12:00.
3. Guilherme  
Endereço: Rua Luiz Tramontim, 300, Curitiba - PR.  
Peso do Produto: 108,0 kg.  
Volume do Produto: 2,0 m<sup>3</sup>.  
Disponibilidade: 08:00 - 17:00.
4. Ferreira  
Endereço: Rua Edmundo Ferreira, 300, Araucária - PR.  
Peso do Produto: 1422,0 kg.  
Volume do Produto: 10,0 m<sup>3</sup>.  
Disponibilidade: 10:00 - 15:00.
5. Carlos  
Endereço: Avenida Áustria, 100, Fazenda Rio Grande - PR.  
Peso do Produto: 620,0 kg.  
Volume do Produto: 4,0 m<sup>3</sup>.  
Disponibilidade: 08:00 - 17:00.
6. José Pereira  
Endereço: Rua Islândia, 50, Fazenda Rio Grande - PR.  
Peso do Produto: 480,0 kg.  
Volume do Produto: 3,0 m<sup>3</sup>.  
Disponibilidade: 09:00 - 12:00.

A tabela 4 apresenta os valores calculados de tempo de viagem, em minutos, entre todos os nós do cenário. Este cálculo é realizado a partir do serviço de rotas da plataforma *Bing Maps*.

O número de restrições e variáveis presentes no cenário é apresentado na tabela 5.

#### 4.2.2 SOLUÇÃO

O tempo limite de execução de 20 minutos foi atingido, valor de tempo limite estabelecido conforme requisito não-funcional contido na seção 3.1. Uma solução válida

**Tabela 4: Estudo de Caso 2: matriz de tempo de viagem em minutos. D = Depósito, E = Entrega, C = Coleta.**

	D	E1	E2	E3	E4	E5	E6	E7	E8	C1	C2	C3	C4	C5	C6
D	0	20	37	28	40	21	32	24	10	33	35	17	26	33	34
E1	19	0	26	26	29	34	46	19	12	15	30	26	31	30	31
E2	38	28	0	31	10	37	55	20	31	33	27	36	49	49	49
E3	26	26	33	0	37	40	51	25	30	18	36	37	36	36	37
E4	40	30	10	38	0	40	57	22	34	38	29	38	51	51	52
E5	22	36	40	40	41	0	32	24	29	47	17	23	39	44	45
E6	31	45	58	49	61	31	0	41	40	58	46	32	36	53	54
E7	22	21	22	28	25	24	40	0	16	27	15	18	42	42	42
E8	9	12	30	26	33	29	41	18	0	26	30	17	29	30	31
C1	32	16	31	15	34	45	56	23	26	0	35	38	41	42	43
C2	32	28	27	35	30	17	42	12	26	36	0	25	50	49	50
C3	16	27	35	37	39	21	33	18	17	38	26	0	36	41	42
C4	26	29	48	33	51	40	36	41	32	42	52	37	0	37	38
C5	28	27	45	31	48	41	52	38	30	39	50	38	36	0	3
C6	30	29	47	33	51	43	55	40	32	41	52	40	38	3	0

**Fonte: Autoria própria.**

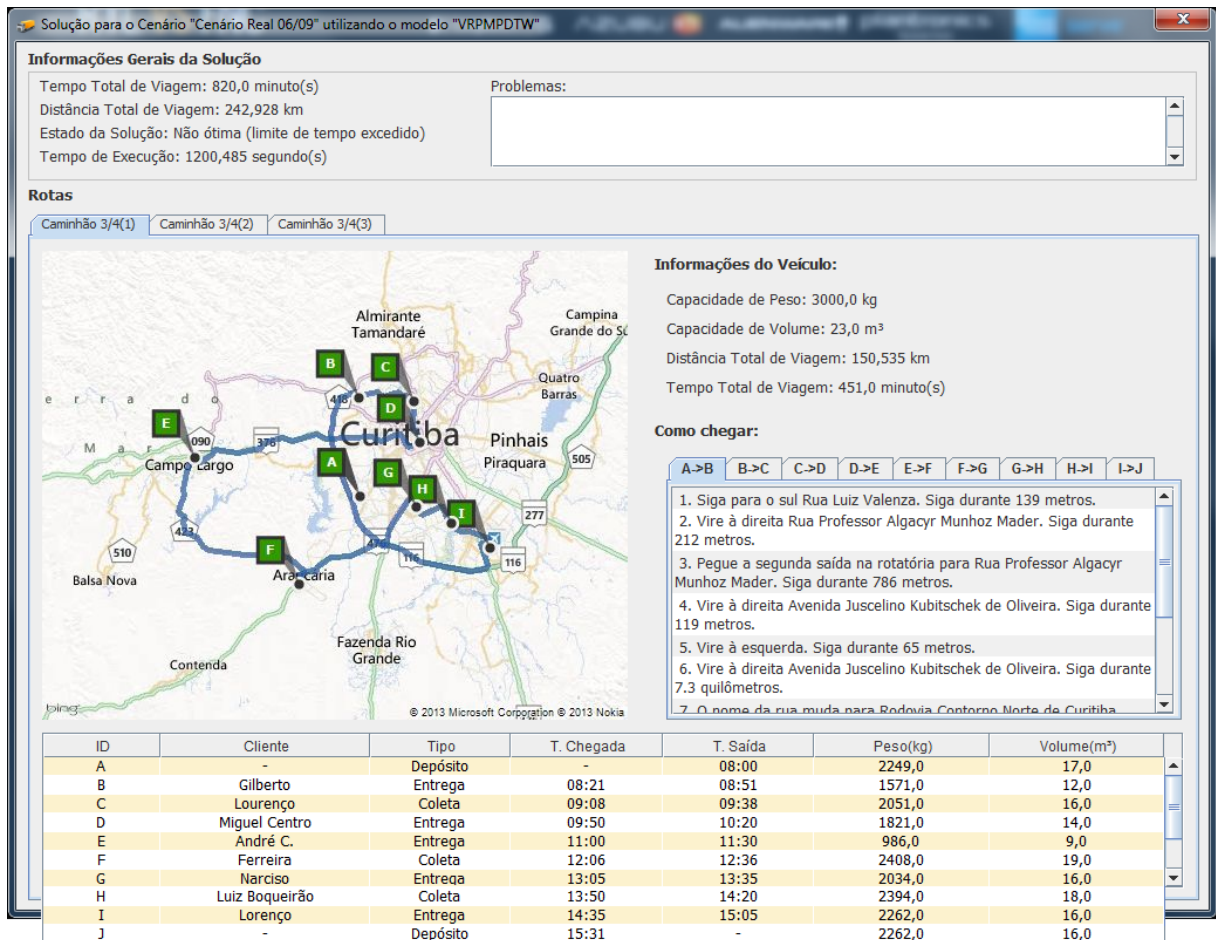
**Tabela 5: Dados do Estudo de Caso 2.**

<b>Restrições</b>	<b>18804</b>
<b>Variáveis</b>	<b>3497</b>
Binárias	705
Inteiras	45
Contínuas	2747
<b>Coeficientes Diferentes de Zero</b>	<b>61879</b>

**Fonte: Autoria própria.**

não-ótima (*gap* de integralidade de 92,63%) foi encontrada, sem nenhuma violação de restrição. Salienta-se, contudo, que a primeira solução factível foi encontrada pelo solucionador em menos de 3 segundos e após 7 minutos não ocorreu evolução no valor da solução encontrada.





**Figura 21: Solução do Estudo de Caso 2 - Caminhão 3/4(1).**

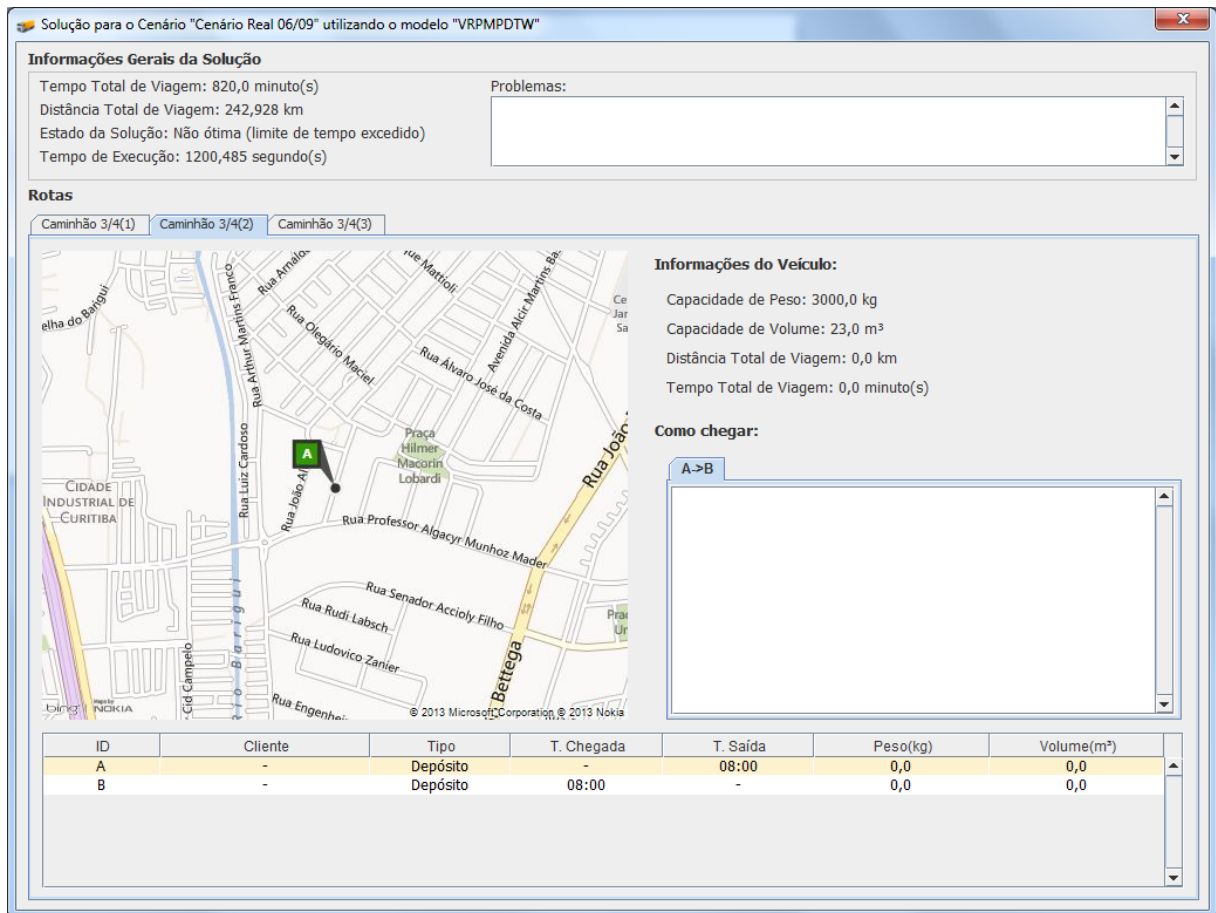
**Fonte: Autoria Própria.**

O valor da função objetivo retornado pelo modelo após o limite de tempo de execução foi de 820 minutos com um total de 242,928 quilômetros percorridos. As telas de solução com as rotas de cada um dos três veículos podem ser observadas nas figuras 21, 22 e 23.

O resultado para o veículo “Caminhão 3/4(2)” significa que não foi necessário seu uso na solução do cenário. Já os outros dois caminhões, como pode ser observado nas figuras 21 e 23, realizaram 8 e 7 serviços, respectivamente.

#### 4.2.3 ANÁLISE

A transportadora, normalmente, separa seus caminhões por setores de atendimento (Ex: Fazenda Rio Grande, Campo Largo, Pinhais, Curitiba Centro...) e define quais clientes cada um deles deveriam atender baseado nisso. Consideram-se também verificações como capacidade e avaliações superficiais de disponibilidade do cliente, base-



**Figura 22: Solução do Estudo de Caso 2 - Caminhão 3/4(2).**

**Fonte: Autoria Própria.**

ado no conhecimento de tempo de viagem do responsável por programar as viagens. Por ser um método manual acontecem imprevistos e erros como atraso ou impossibilidade de atendimento, que conseqüentemente prejudicam a relação com o cliente. Portanto, uma solução tão boa quanto a encontrada para um número relativamente grande de serviços seria difícil de ser obtida sem o auxílio do *software* de apoio a decisões desenvolvido.

As rotas dos veículos 1 e 3 estão completamente espalhadas em Curitiba e região metropolitana, sendo que mesmo sem utilizar o veículo 2, nenhuma restrição de capacidade e de tempo de disponibilidade é violada. Os caminhos percorridos são longos, com mais de 90 km percorridos por cada um dos veículos utilizados. Porém, percebe-se que as rotas estabelecidas dão preferência a atender clientes distantes ao invés de chegarem antes do início da janela de tempo de um mais próximo. Como é o caso do cliente "Narciso" atendido pelo "Caminhão 3/4(1)". O cliente está próximo ao depósito, mas como não estava dentro da janela para ser atendido, outros clientes foram atendidos primeiro e no meio da viagem, antes de ir para Araucária, aproveitou-se o fato de estar em um horário

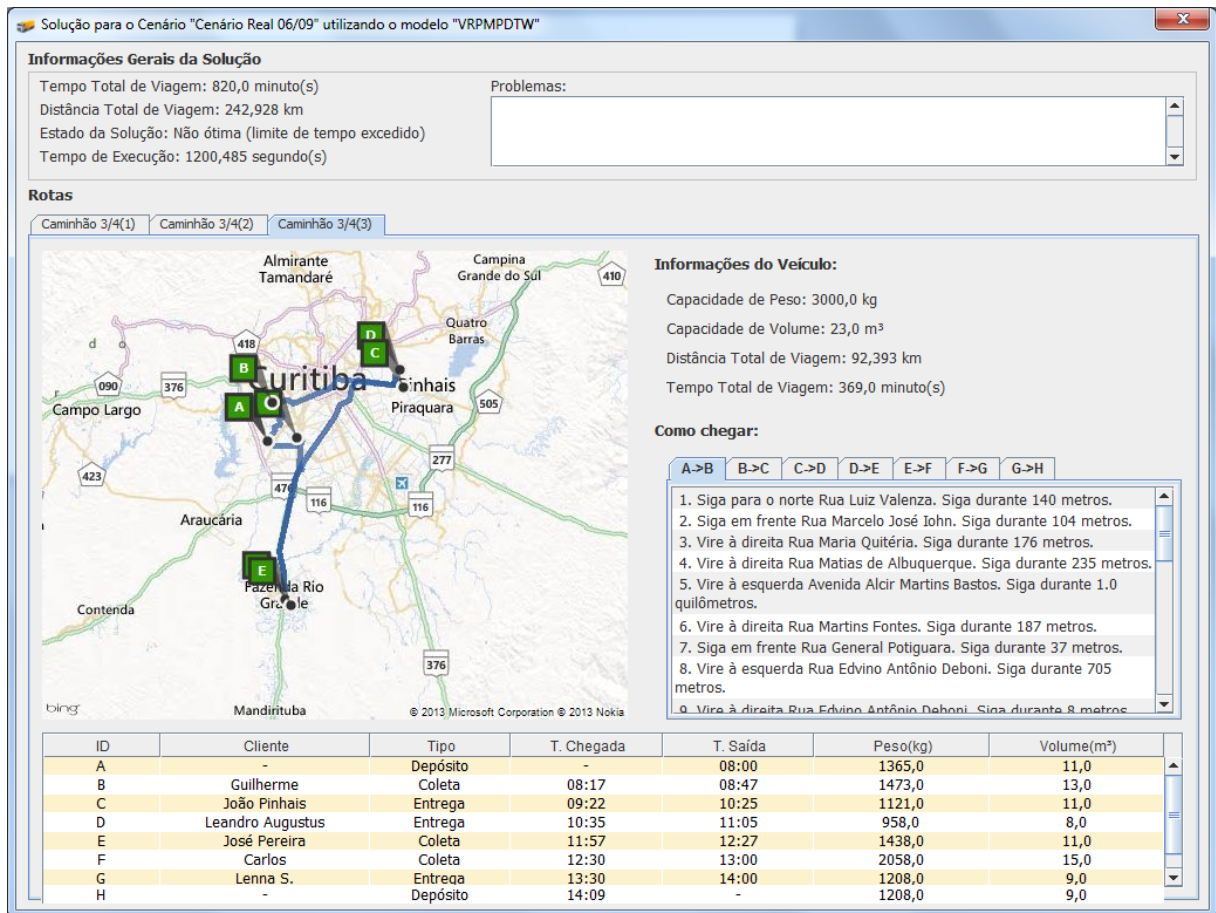


Figura 23: Solução do Estudo de Caso 2 - Caminhão 3/4(3).

Fonte: Autoria Própria.

válido para realizar o serviço.

Pequenas alterações no cenário podem resultar em um rearranjo dos veículos muito diferente do atual. Alterações de janelas de tempo, carga ou adição de um ou dois serviços poderiam ser acomodados no cenário devido à folga de frota existente. Contudo uma solução válida diferente da anteriormente proposta tenderia a ser obtida.

### 4.3 CONSIDERAÇÕES

O aumento do número de serviços e veículos aumenta muito a complexidade de encontrar uma solução ótima para o problema, o que pode ser observado pelo crescimento de forma proibitiva no tempo de resposta comparando os dois casos de estudo. No segundo caso encontrou-se uma solução válida, mas o tempo de execução limite foi atingido. De fato, observou-se que o sistema apresenta rápida convergência para uma solução válida, caso exista, permitindo que mesmo executando com um tempo

de execução em um limite inferior, ainda assim, uma solução válida poderia ser obtida.

O uso otimizado da frota permite que mais serviços sejam realizados com o mesmo número de veículos. Desta forma, contribui-se economicamente à empresa, como por exemplo: gastos com combustível, manutenção de veículos, investimentos desnecessários em frota, aumento da qualidade de serviço, assim mantendo e atraindo novos clientes.

## 5 GESTÃO DE PROJETO

Neste capítulo serão abordados tópicos relativos com a gestão do projeto: tempo, custos e riscos, detalhados nas seções 5.1, 5.2 e 5.3, respectivamente.

### 5.1 TEMPO

A gestão de tempo do projeto será discutida nesta seção. Será feito, na seção 5.1.1, um paralelo entre o cronograma preliminar, feito ao iniciar o projeto, e como realmente foi executado. A seção 5.1.2 também faz uma comparação do tempo estimado e real gasto para completar as atividades planejadas.

#### 5.1.1 CRONOGRAMA

O tempo disponível para o desenvolvimento do trabalho de conclusão de curso foi de 6 meses e 10 dias, com início no mês de abril de 2013 e finalização no começo de outubro do mesmo ano. A data de entrega coincide com o final do semestre letivo da Universidade. Com isso, foi estipulado um cronograma preliminar para desenvolvimento completo do projeto, apresentado na figura 24. O projeto foi dividido em seis grupos de atividades descritas a seguir:

**Estudos:** envolve o estudo do problema de roteamento de veículos em artigos, dissertações, teses; e o das tecnologias utilizadas neste trabalho: programação matemática, PLIM, da ferramenta IBM ILOG CPLEX, serviços REST, Java.

**Projeto de *Software*:** o projeto do *software* consiste no levantamento dos requisitos, casos de uso, documentos e planejamento de sua arquitetura, ou seja, como são organizadas as entidades, os pacotes e classes do programa.

**Desenvolvimento do *Software*:** envolve atividades de implementação do *software* projetado. Programação da interface para leitura dos cenários, pré-processamentos, interface de entrada, dos serviços e de captura de saída do ILOG CPLEX, pós-processamento e interface gráfica.

**Formulação do Modelo Matemático:** atividades de modelagem do problema,

	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro
Estudos	■	■					
Projeto de Software		■					
Desenvolvimento do Software			■	■	■		
Formulação do Modelo Matemático				■	■		
Análise dos Resultados						■	
Redação do TCC	■	■	■	■	■	■	■

Figura 24: Cronograma Estimado.

Fonte: Autoria Própria.

	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro
Estudos		■	■				
Projeto de Software		■	■				
Desenvolvimento do Software			■	■	■		
Formulação do Modelo Matemático				■	■		
Análise dos Resultados						■	
Redação do TCC	■	■	■	■	■	■	■

Figura 25: Cronograma Real.

Fonte: Autoria Própria.

obtenção de resultados preliminares da modelagem, reformulações do modelo e validação do modelo utilizando o ambiente de modelagem IBM ILOG CPLEX *Optimization Studio*.

**Análise dos Resultados:** atividade de análise dos resultados quanto a sua validade operacional e tempo de execução. Diferentes cenários de entrada, baseados no cenário real, são testados com o intuito de verificar os resultados obtidos sob diferentes condições de operação. O processo pode resultar em alterações em algoritmos a fim de otimizá-los.

**Redação do Trabalho de Conclusão de Curso (TCC):** esta atividade com-

preende a redação do TCC (este documento), conforme normas vigentes na UTFPR.

Devido a atrasos para iniciar as atividades e problemas de planejamento o cronograma teve algumas alterações no decorrer do projeto. A figura 25 mostra o cronograma realmente seguido. As atividades de projeto do *software* demoraram um tempo maior que o estimado para serem finalizadas, devido a problemas anteriormente não pensados, o que resultou em um atraso da tarefa que dependia diretamente dela, o desenvolvimento do *software*. Por esse motivo, foi realizado um esforço para completar a tarefa de desenvolvimento em um tempo menor. Felizmente, foi possível compensar o tempo excessivo gasto na tarefa anterior. Os três últimos grupos de atividades foram completados conforme planejadas, sem atrasos.

### 5.1.2 ATIVIDADES

Cada atividade possui uma carga horária que deve ser alocadas entre os membros participantes do projeto. Como o projeto é composto apenas por um integrante, é responsabilidade dele executar todas as atividades.

A tabela 6 mostra uma comparação entre o número de horas estipulado para cada atividade e o quanto realmente foi trabalhado para completá-la. Os dados foram contabilizados o mais próximo possível das horas reais trabalhadas. Pode-se concluir que o planejamento ficou condizente, apesar de diferenças em algumas atividades. No todo, a quantidade de horas estimadas ficou próxima do real.

As diferenças mais graves foram nas seguintes atividades: projeto da arquitetura do *software*, desenvolvimento das camadas MVC e de serviços e a redação final. No primeiro, foi subestimado o tempo para pensar na organização do programa e várias reformulações tiveram que ser feitas. No segundo, a implementação das camadas MVC também foram subestimadas, principalmente, as classes da camada de visão, que geraram diversos contra-tempos. Por outro lado, os serviços REST foram muito mais simples de implementar do que o realmente esperado. Na redação final, pode-se aproveitar muito do material já escrito no relatório de plano de projeto, o que reduziu em muito o tempo de escrita.

## 5.2 CUSTOS

O custo do projeto está relacionado apenas ao número de horas trabalhadas pela equipe. Não houve nenhum gasto com *hardware* ou com aquisição de licenças de *software*. Foi utilizado a versão acadêmica do solucionador IBM ILOG CPLEX 12.5 e o serviço Bing *Maps* oferece uma chave básica, sem custos, para *softwares* com uma média baixa de acessos à ferramenta (até 50.000 acessos por dia). Outro custo que não foi contabilizado

**Tabela 6: Alocação de horas das atividades do projeto.**

Tarefa	H. Estimadas	H. Trabalhadas
<b>Estudos</b>	<b>100</b>	<b>95</b>
Estudar o problema de roteamento de veículos	35	35
Estudar programação matemática	40	40
Estudar o ambiente IBM ILOG CPLEX 12.5	15	10
Estudar serviços REST	10	10
<b>Projeto de <i>Software</i></b>	<b>100</b>	<b>120</b>
Levantamento de Requisitos	30	30
Casos de Uso	30	30
Projeto da Arquitetura do <i>Software</i>	40	60
<b>Desenvolvimento do <i>Software</i></b>	<b>340</b>	<b>360</b>
Camadas MVC	200	240
Camada de comunicação com o solucionador	70	70
Camada de comunicação com os serviços REST	70	50
<b>Formulação do Modelo Matemático</b>	<b>200</b>	<b>200</b>
<b>Análise dos Resultados</b>	<b>60</b>	<b>50</b>
<b>Redação do TCC</b>	<b>180</b>	<b>150</b>
<b>Total de Horas</b>	<b>980</b>	<b>970</b>

**Fonte: Autoria própria.**

é o de depreciação do computador, pois foi considerado não significativo.

O trabalho custou um total de 970 horas de trabalho (tabela 5.1) por uma equipe de um programador. Admitindo que um programador ganhe 35,00 reais por hora. O *software* teria um custo total de R\$ 33.950,00 diluídos no decorrer de 6 meses de desenvolvimento.

### 5.3 RISCOS

Riscos de projeto são situações adversas que podem ocorrer durante seu desenvolvimento. Sommerville (2007) define riscos como problemas adversos não previstos durante a fase de planejamento que podem ameaçar de alguma forma o projeto, o *software* ou a equipe, influenciando na sua qualidade ou impactando o cronograma.



O fato de apenas um programador desenvolver o projeto, aumenta o risco de diversos problemas de cronograma. Qualquer problema pessoal, de saúde ou familiar impacta em atrasos, pois não existe ninguém para substituir ou mitigar o tempo perdido. A ocorrência deste risco foi constatada no início do cronograma observado nas figuras 24 e 25, causado por problemas de saúde, que resultaram em atrasos das atividades.

Outro risco é a subestimativa do tamanho do projeto e seu nível de dificuldade para desenvolvimento. A inexperiência com planejamento e estimativas de projetos de *software* e de algumas tecnologias envolvidas resultaram em um mau dimensionamento do tempo. Algumas atividades demandaram mais horas do que o estimado, resultando, conseqüentemente, em alterações no cronograma.

Riscos de atrasos ou falhas de comunicação entre a empresa e a equipe também eram passíveis de ocorrerem, o que prejudicaria o desenvolvimento do projeto, principalmente do modelo matemático. No fim, não houve problemas entre as duas entidades.

Existiam também riscos de tecnologia, como problemas na integração do solucionador com a linguagem Java, que poderia impactar na escolha de outra tecnologia para substituí-lo. A probabilidade do risco era baixa desde o início devido às experiências anteriores da equipe com essa integração.

No mais, não ocorreu nenhum risco com impacto alto para o desenvolvimento do projeto. Todos os problemas ocorridos foram contornados reorganizando o calendário de atividades e aumentando a carga de trabalho.

#### 5.4 CONSIDERAÇÕES

Apesar de erros de planejamento, nenhum problema acarretou em prejuízos que impossibilitaram o desenvolvimento de alguma etapa do projeto. Todos os riscos decorridos puderam ser mitigados com replanejamentos do cronograma e aumento da carga de trabalho, possibilitado pela folga nos horários disponíveis pelo programador. As diferenças de horas de trabalho estimadas para o realmente trabalhado se contra balancearam entre as atividades, resultando em uma pequena diferença de 10 horas a menos do estipulado, em um total de 970 horas.

O projeto não acarretou em nenhum gasto monetário para a equipe, apenas horas de trabalho, que convertidas em dinheiro valeriam R\$ 33.950,00, se cada hora valesse R\$ 35,00 para a empresa.

A gerência do projeto não foi complicada, pois todos os dados estavam concentrados em apenas um programador, o que facilitou o trabalho de alterações e tomada de decisões. O sucesso do planejamento também se deve ao conhecimento do ritmo de

trabalho que poderia ser empregado, baseado em experiências profissionais vivenciadas no decorrer do curso de graduação.

## 6 CONSIDERAÇÕES FINAIS

O sistema desenvolvido atingiu o objetivo proposto, resolvendo o problema de roteamento de veículos da empresa estudada, sendo o problema semelhante a uma variante do VRP, o VRPMPDTW (*Vehicle Routing Problem with Mixed Pickup and Delivery and Time Windows*). Levou-se em consideração restrições e características de cenários reais para implementação e validação do modelo matemático. Uma interface gráfica, em Java, foi desenvolvida para criação/edição de cenários e visualização da solução, responsável também por fazer a comunicação com o solucionador IBM ILOG CPLEX, ferramenta de modelagem matemática.

Utilizou-se uma técnica de Pesquisa Operacional, a Programação Linear Inteira Mista (PLIM), como abordagem para solucionar o problema da transportadora. A formulação matemática estabeleceu 17 grupos de restrições que modelam aspectos da transportadora, como janelas de disponibilidade dos clientes e limites de capacidade de peso e volume para os veículos. A função objetivo procurou minimizar o tempo de viagem gasto pelos veículos com o menor número de violações de restrições possíveis.

Resultados válidos em tempo computacional não proibitivo (até 20 minutos) foram obtidos para cenários típicos da empresa, com uma média de 10 serviços (entregas/coletas) por dia. Porém, percebe-se a complexidade do problema observando o quanto o tempo de resolução cresce ao aumentar-se o número de serviços do cenário, como visto nos estudos de caso do capítulo 4. Casos maiores exigiriam a utilização de algoritmos alternativos, como CLP (*Constraint Logic Programming*) ou a união de PLIM com CLP (MAGATÃO, 2005; MAGATÃO et al., 2011). Abordagens heurísticas também poderiam ser utilizadas a fim de melhorar o desempenho computacional.

O projeto foi desenvolvido em um total 970 horas ao longo de 6 meses e meio de desenvolvimento. Não houve custos associados com *hardware* e licenças de *software* ao projeto, apenas relacionados ao custo de trabalho. Se o valor da hora de trabalho do desenvolvedor for de R\$ 35,00, o valor total estimado do projeto seria de R\$ 33.950,00.

Trabalhos futuros podem incluir algumas características do sistema de transporte que não foram levadas em consideração no presente trabalho. Por exemplo, a possibilidade de surgir coletas no decorrer das rotas, que inseriria dinamicidade ao problema. Outra

característica seria a possibilidade dos veículos realizarem mais de uma rota, ou seja, poderem voltar ao depósito para descarregar e carregar com novas mercadorias.

Outras possíveis contribuições ou aprimoramentos que podem ser realizados na sequência do presente trabalho estão elencadas a seguir:

- Alterar a função objetivo do modelo a fim também de minimizar os gastos com combustível. Desta forma, levaria-se em consideração a autonomia de cada veículo (quantidade de quilômetros por litro), optando, sempre que possível, por uma solução econômica também neste aspecto.
- Os serviços REST da plataforma Bing *Maps* permitem calcular rotas levando em consideração o tráfego de veículos, dependendo da hora do dia, o que impacta diretamente no tempo de viagem de um ponto para outro. Essa característica poderia ser incorporada ao modelo, aproximando-o mais do ambiente real.
- Inserir mais de um nó depósito no cenário. Veículos estariam associados a cada depósito no início da jornada. Podendo existir, ou não, a possibilidade de transporte de mercadorias entre depósitos.
- Tratar o problema da mochila, que é complementar ao problema de roteamento de veículos. Esse problema diz respeito à ordem e como as mercadorias deveriam ser carregadas no veículo. Pode ocorrer da capacidade de peso e volume ser respeitada, mas o formato da mercadoria não permitir o melhor aproveitamento do espaço e, assim, impossibilitar o carregamento planejado.

Problemas de roteamento de veículos vêm sendo muito estudados no decorrer das últimas décadas, devido à importância econômica do setor logístico e à crescente inserção da informática nas empresas. Abordagens customizadas são, via de regra, necessárias para o mapeamento da realidade de cada transportadora. Conforme mais características são incorporadas, o problema se torna cada vez mais complexo. Geram-se novas restrições, novas variáveis e modificações na forma de modelagem.

O presente trabalho contribui para a solução do problema de roteamento de veículos da transportadora em estudo, apresentando uma abordagem que alia ferramentas de Pesquisa Operacional com técnicas computacionais de estado da arte. O resultado, o *software*, constitui uma ferramenta de auxílio ao processo de tomada de decisões operacionais do problema abordado.

## REFERÊNCIAS

- BALDACCI, R.; TOTH, P.; VIGO, D. Exact algorithms for routing problems under vehicle capacity constraints. **Annals of Operations Research**, Springer US, v. 175, p. 213–245, 2010. ISSN 0254-5330. Disponível em: <<http://dx.doi.org/10.1007/s10479-009-0650-0>>.
- BELMECHERI, F. et al. Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. **Journal of Intelligent Manufacturing**, Springer US, p. 1–15, 2012. ISSN 0956-5515. Disponível em: <<http://dx.doi.org/10.1007/s10845-012-0627-8>>.
- BERBEGLIA, G. et al. Static pickup and delivery problems: a classification scheme and survey. **TOP: An Official Journal of the Spanish Society of Statistics and Operations Research**, v. 15, n. 1, p. 1–31, July 2007.
- BERBEGLIA, G.; CORDEAU, J.-F.; LAPORTE, G. Dynamic pickup and delivery problems. **European Journal of Operational Research**, v. 202, n. 1, p. 8 – 15, 2010.
- BING. **Bing Maps REST Services**. 2013. Disponível em: <<http://msdn.microsoft.com/en-us/library/ff701713.aspx>>. Acesso em: 15 de setembro de 2013.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML - Guia do Usuário**. 2. ed. Rio de Janeiro, RJ: Elsevier, 2005.
- CORDEAU, J.-F. et al. **New heuristics for the vehicle routing problem**. [S.l.]: Public Health Service, 2005. 279–297 p.
- CRUZ, S. M. S. da. **Serviços Web ? Uma breve introdução**. 2005. Disponível em: <<http://www.nce.ufrj.br/conceito/artigos/2005/01p1-1.htm>>. Acesso em: 14 de setembro de 2013.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science, INFORMS**, v. 6, p. 80–91, 1959.
- DAVENDRA, D. **Traveling Salesman Problem, Theory and Applications**. Rijeka, Croácia: InTech, 2010.
- ELKSTEIN, D. M. **Learn REST: A Tutorial**. 2008. Disponível em: <<http://rest.elkstein.org/>>. Acesso em: 14 de setembro de 2013.
- FIELDING, R. T. **Representational State Transfer**. 2000.
- FREEMAN, E. F. E. **Head First Design Patterns**. [S.l.]: O’Reilly, 2004.
- GAMMA, E. et al. **Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos**. [S.l.]: Bookman, 2000.

GOETSCHALCKX, M.; JACOBS-BLECHA, C. The vehicle routing problem with backhauls. **European Journal of Operational Research**, v. 42, n. 1, p. 39 – 51, 1989. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/037722178990057X>>.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização Combinatória e Programação Linear - Modelos e Algoritmos**. Rio de Janeiro, RJ: Campus, 2000.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização Combinatória e Programação Linear - Modelos e Algoritmos**. 2. ed. Rio de Janeiro, RJ: Campus, 2005.

ILOG. **IBM ILOG CPLEX Optimization Studio**. USA, 2013. Disponível em: <<http://www-03.ibm.com/software/products/us/en/ibmilogcpleoptistud/>>. Acesso em: 17 de setembro de 2013.

LAPORTE, G. The traveling salesman problem: An overview of exact and approximate algorithms. **European Journal of Operational Research**, Montreal, Canadá, v. 59, p. 231–247, 1991.

LAPORTE, G.; OSMAN, I. H. Routing problems: A bibliography. **Annals of Operations Research**, Baltzer Science Publishers, Baarn/Kluwer Academic Publishers, v. 61, p. 227–262, 1995. ISSN 0254-5330.

LAPORTE, Y. N. G. Exact algorithms for the vehicle routing problem. **Annals of Discrete Mathematics**, v. 31, p. 147–184, 1987.

LIN, C. A cooperative strategy for a vehicle routing problem with pickup and delivery time windows. **Computers & Industrial Engineering**, v. 55, n. 4, p. 766 – 782, 2008. ISSN 0360-8352.

MAGATÃO, L. **Constraint Logic Programming and Mixed Integer Linear Programming: Towards a Unified Modeling Framework**. Tese de Doutorado — Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil, 2005.

MAGATÃO, L.; ARRUDA, L.; NEVES-JR, F. A combined clp-milp approach for scheduling commodities in a pipeline. **Journal of Scheduling**, v. 14, n. 1, p. 57–87, 2011.

NAGY, G.; WASSAN, N.; SALHI, S. The vehicle routing problem with restricted mixing of deliveries and pickups. **Journal of Scheduling**, Springer US, v. 16, p. 199–213, 2013. ISSN 1094-6136. Disponível em: <<http://dx.doi.org/10.1007/s10951-012-0296-7>>.

PARRAGH, S.; DOERNER, K.; HARTL, R. A survey on pickup and delivery problems. part i: Transportation between customers and depot. **Journal für Betriebswirtschaft**, Springer-Verlag, v. 58, p. 21–51, 2008. ISSN 0344-9327.

PARRAGH, S.; DOERNER, K.; HARTL, R. A survey on pickup and delivery problems. part ii: Transportation between pickup and delivery locations. **Journal für Betriebswirtschaft**, Springer-Verlag, v. 58, p. 81–117, 2008. ISSN 0344-9327.

ROPKE, S. **Heuristic and exact algorithms for vehicle routing problems**. Tese de Doutorado — Universidade de Copenhagen, Copenhagen, Dinamarca, 2005.

SOMMERVILLE, I. **Engenharia de Software**. 8. ed. [S.l.]: Pearson Addison-Wesley, 2007.

SUBRAMANIAN, A. **Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problem**. Tese de Doutorado — Universidade Federal Fluminense, Niterói, Rio de Janeiro, 2012.

TOTH, P.; VIGO, D. An exact algorithm for the vehicle routing problem with backhauls. **Transportation Science**, Itália, v. 31, p. 372–385, 1997.

TOTH, P.; VIGO, D. Models, relaxations and exact approaches for the capacitated vehicle routing problem. **Discrete Appl. Math.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 123, n. 1-3, p. 487–512, nov. 2002. ISSN 0166-218X. Disponível em: <[http://dx.doi.org/10.1016/S0166-218X\(01\)00351-1](http://dx.doi.org/10.1016/S0166-218X(01)00351-1)>.

VECCHINI, D. C. **Uma abordagem híbrida ao problema de roteização dinâmica de veículos com janela de tempo**. Dissertação de Mestrado — Universidade Estadual de Campinas, Campinas, Brasil, 2011.

VIDAL, T. et al. **Heuristics for Multi-Attribute Vehicle Routing Problems: A Survey and Synthesis**. 2012.

VIEIRA, A. F. P. **Planeamento e Otimização de Rotas para Veículos**. Dissertação de Mestrado — Universidade de Aveiro, Aveiro, Portugal, 2012.

W3C. **Web Services Glossary**. 2004. Disponível em: <<http://www.w3.org/TR/ws-gloss/>>. Acesso em: 14 de setembro de 2013.

WADE, A.; SALHI, S. An investigation into a new class of vehicle routing problem with backhauls. **Omega**, v. 30, n. 6, p. 479 – 487, 2002. ISSN 0305-0483. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305048302000567>>.

WANG, H.-F.; CHEN, Y.-Y. A genetic algorithm for the simultaneous delivery and pickup problems with time window. **Computers & Industrial Engineering**, v. 62, n. 1, p. 84 – 95, 2012. ISSN 0360-8352. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360835211002476>>.

WANG, H.-F.; CHEN, Y.-Y. A coevolutionary algorithm for the flexible delivery and pickup problem with time windows. **International Journal of Production Economics**, v. 141, n. 1, p. 4 – 13, 2013. ISSN 0925-5273. <ce:title>Metaheuristics for manufacturing scheduling and logistics problems</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925527312001697>>.

WILLIAMS, H. P. **Model Building in Mathematical Programming, 4th Edition**. 4. ed. [S.l.]: Wiley, 1999. Paperback. ISBN 0471997889.

## APÊNDICE A – XML CENÁRIO

Cenário com 1 veículo, 1 coleta e 1 entrega.

```
<?xml version="1.0" encoding="UTF-8"?>
<scenario name="Cenário Exemplo 2" date="17/09/2013">
  <deposit name="Depósito Central">
    <location address="Rua Luiz Valenza, Curitiba - PR, 81350-080, Brasil" latitude="-25.4925251
      " longitude="-49.3326454" />
    <availability from="08:00" to="17:00" />
  </deposit>
  <vehicles>
    <vehicle name="Ducato" weight_capacity="1100.0" volume_capacity="7.5" />
    <vehicle name="Truck" weight_capacity="7000.0" volume_capacity="30.0" />
  </vehicles>
  <deliveries>
    <delivery name="João Centro">
      <location address="Avenida Sete de Setembro, 3200, Curitiba - PR, 80230-010, Brasil"
        latitude="-25.4386215" longitude="-49.2699242" />
      <product weight="600.0" volume="3.0" />
      <availabilities>
        <availability from="08:00" to="17:00" />
      </availabilities>
    </delivery>
  </deliveries>
  <pickups>
    <pickup name="William">
      <location address="Avenida Manoel Ribas, 5920, Curitiba - PR, 82020-000, Brasil" latitude="
        -25.4041767" longitude="-49.3280792" />
      <product weight="300.0" volume="2.0" />
      <availabilities>
        <availability from="08:00" to="17:00" />
      </availabilities>
    </pickup>
  </pickups>
  <itineraryMatrix>
    <node>
      <itinerary distance="0.0" travelTime="0.0" />
      <itinerary distance="11.428" travelTime="19.53">
        <itineraryItem type="DepartStart" instruction="Siga para o sul Rua Luiz Valenza. Siga
          durante 139 metros." />
      </itinerary>
    </node>
  </itineraryMatrix>
</scenario>
```



```

<itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Professor Algacyr Munhoz
  Mader. Siga durante 809 metros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua João Bettega. Siga
  durante 4.1 quilômetros." />
<itineraryItem type="TurnRight" instruction="Vire à direita Rua Doutor João Tobias Pinto
  Rebelo. Siga durante 119 metros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Luiz Parigot de Souza.
  Siga durante 287 metros." />
<itineraryItem type="Continue" instruction="Siga em frente Rua Augusto de Mari. Siga
  durante 331 metros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Eduardo Carlos Pereira.
  Siga durante 475 metros." />
<itineraryItem type="Continue" instruction="Siga em frente Rua Guilherme Pugsley. Siga
  durante 2.1 quilômetros." />
<itineraryItem type="Continue" instruction="Siga em frente Largo Doutor Acir Mulinari.
  Siga durante 80 metros." />
<itineraryItem type="RoadNameChange" instruction="O nome da rua muda para Rua Guilherme
  Pugsley. Siga durante 21 metros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Saint'Hilaire. Siga
  durante 623 metros." />
<itineraryItem type="TurnRight" instruction="Vire à direita Avenida Silva Jardim. Siga
  durante 2.1 quilômetros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Avenida Marechal Floriano
  Peixoto. Siga durante 184 metros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Avenida Sete de Setembro.
  Siga durante 159 metros." />
<itineraryItem type="ArriveFinish" instruction="Você chegará ao seu destino na Avenida
  Sete de Setembro . O destino está à sua direita." />
</itinerary>
<itinerary distance="16.923" travelTime="22.01">
  <itineraryItem type="DepartStart" instruction="Siga para o sul Rua Luiz Valenza. Siga
    durante 139 metros." />
  <itineraryItem type="TurnRight" instruction="Vire à direita Rua Professor Algacyr Munhoz
    Mader. Siga durante 212 metros." />
  <itineraryItem type="ExitRoundabout" instruction="Pegue a segunda saída na rotatória para
    Rua Professor Algacyr Munhoz Mader. Siga durante 786 metros." />
  <itineraryItem type="TurnRight" instruction="Vire à direita Avenida Juscelino Kubitschek
    de Oliveira. Siga durante 119 metros." />
  <itineraryItem type="TurnLeft" instruction="Vire à esquerda. Siga durante 65 metros." />
  <itineraryItem type="TurnRight" instruction="Vire à direita Avenida Juscelino Kubitschek
    de Oliveira. Siga durante 7.3 quilômetros." />
  <itineraryItem type="RoadNameChange" instruction="O nome da rua muda para Rodovia
    Contorno Norte de Curitiba. Siga durante 4.9 quilômetros." />
  <itineraryItem type="KeepRight" instruction="Mantenha-se à direita na direção Sta
    Felicidade/Campo Magro. Siga durante 230 metros." />
  <itineraryItem type="ExitRoundabout" instruction="Pegue a primeira saída na rotatória na
    direção Sta Felicidade/Campo Magro. Siga durante 3.2 quilômetros." />
  <itineraryItem type="ArriveFinish" instruction="Você chegará ao seu destino na Avenida
    Manoel Ribas . O destino está à sua esquerda." />
</itinerary>
</node>

```

```

<node>
  <itinerary distance="10.732" travelTime="17.2">
    <itineraryItem type="DepartStart" instruction="Siga para o sudoeste Avenida Sete de
      Setembro. Siga durante 2.4 quilômetros." />
    <itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Castro Alves. Siga
      durante 711 metros." />
    <itineraryItem type="Continue" instruction="Siga em frente Rua Guararapes. Siga durante
      1.3 quilômetros." />
    <itineraryItem type="Continue" instruction="Siga em frente Rua Francisco Frischmann. Siga
      durante 653 metros." />
    <itineraryItem type="KeepLeft" instruction="Mantenha-se à esquerda Rua Francisco
      Frischmann. Siga durante 634 metros." />
    <itineraryItem type="TurnRight" instruction="Vire à direita Rua João Bettega. Siga
      durante 4.1 quilômetros." />
    <itineraryItem type="TurnRight" instruction="Vire à direita Rua Professor Algacyr Munhoz
      Mader. Siga durante 809 metros." />
    <itineraryItem type="TurnRight" instruction="Vire à direita Rua Luiz Valenza. Siga
      durante 139 metros." />
    <itineraryItem type="ArriveFinish" instruction="Você chegará ao seu destino na Rua Luiz
      Valenza . O destino está à sua esquerda." />
  </itinerary>
  <itinerary distance="0.0" travelTime="0.0" />
  <itinerary distance="8.165" travelTime="12.18">
    <itineraryItem type="DepartStart" instruction="Siga para o sudoeste Avenida Sete de
      Setembro. Siga durante 723 metros." />
    <itineraryItem type="TurnRight" instruction="Vire à direita Rua Brigadeiro Franco. Siga
      durante 2.5 quilômetros." />
    <itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Alcides Munhoz. Siga
      durante 540 metros." />
    <itineraryItem type="TurnRight" instruction="Vire à direita Avenida Manoel Ribas. Siga
      durante 3.5 quilômetros." />
    <itineraryItem type="ExitRoundabout" instruction="Pegue a terceira saída na rotatória
      para Avenida Manoel Ribas. Siga durante 155 metros." />
    <itineraryItem type="TurnRight" instruction="Vire à direita Avenida Manoel Ribas. Siga
      durante 724 metros." />
    <itineraryItem type="ArriveFinish" instruction="Você chegará ao seu destino na Avenida
      Manoel Ribas . O destino está à sua direita." />
  </itinerary>
</node>
<node>
  <itinerary distance="15.751" travelTime="22.05">
    <itineraryItem type="DepartStart" instruction="Siga para o noroeste Avenida Manoel Ribas.
      Siga durante 329 metros." />
    <itineraryItem type="TurnLeft" instruction="Vire à esquerda Avenida Vereador Toaldo Túlio
      . Siga durante 4.8 quilômetros." />
    <itineraryItem type="TurnRight" instruction="Vire à direita Rodovia do Café. Siga durante
      1.3 quilômetros." />
    <itineraryItem type="KeepRight" instruction="Mantenha-se à direita na direção Retorno/
      Plast. Paraná. Siga durante 395 metros." />
    <itineraryItem type="TurnRight" instruction="Vire à direita na direção Retorno/Plast.
      Paraná. Siga durante 238 metros." />
  </itinerary>
</node>

```

```

<itineraryItem type="RoadNameChange" instruction="O nome da rua muda para Avenida
  Juscelino Kubitschek de Oliveira. Siga durante 491 metros." />
<itineraryItem type="KeepLeft" instruction="Mantenha-se à esquerda Avenida Juscelino
  Kubitschek de Oliveira. Siga durante 5.3 quilômetros." />
<itineraryItem type="KeepRight" instruction="Mantenha-se à direita Avenida Juscelino
  Kubitschek de Oliveira. Siga durante 245 metros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Raul Pompéia. Siga
  durante 104 metros." />
<itineraryItem type="TurnRight" instruction="Vire à direita Avenida Juscelino Kubitschek
  de Oliveira. Siga durante 1.4 quilômetros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Professor Algacyr Munhoz
  Mader. Siga durante 743 metros." />
<itineraryItem type="ExitRoundabout" instruction="Pegue a segunda saída na rotatória para
  Rua Professor Algacyr Munhoz Mader. Siga durante 256 metros." />
<itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Luiz Valenza. Siga
  durante 139 metros." />
<itineraryItem type="ArriveFinish" instruction="Você chegará ao seu destino na Rua Luiz
  Valenza . O destino está à sua esquerda." />
</itinerary>
<itinerary distance="8.958" travelTime="14.55">
  <itineraryItem type="DepartStart" instruction="Siga para o sudeste Avenida Manoel Ribas.
    Siga durante 724 metros." />
  <itineraryItem type="TurnLeft" instruction="Vire à esquerda Avenida Manoel Ribas. Siga
    durante 129 metros." />
  <itineraryItem type="ExitRoundabout" instruction="Pegue a primeira saída na rotatória
    para Avenida Manoel Ribas. Siga durante 3.7 quilômetros." />
  <itineraryItem type="BearRight" instruction="Vire levemente à direita Alameda Presidente
    Taunay. Siga durante 302 metros." />
  <itineraryItem type="TurnLeft" instruction="Vire à esquerda Rua Júlia Wanderley. Siga
    durante 231 metros." />
  <itineraryItem type="TurnRight" instruction="Vire à direita Rua Desembargador Motta. Siga
    durante 2.5 quilômetros." />
  <itineraryItem type="TurnLeft" instruction="Vire à esquerda Avenida Silva Jardim. Siga
    durante 1.0 quilômetros." />
  <itineraryItem type="TurnLeft" instruction="Vire à esquerda Avenida Marechal Floriano
    Peixoto. Siga durante 184 metros." />
  <itineraryItem type="TurnLeft" instruction="Vire à esquerda Avenida Sete de Setembro.
    Siga durante 159 metros." />
  <itineraryItem type="ArriveFinish" instruction="Você chegará ao seu destino na Avenida
    Sete de Setembro . O destino está à sua direita." />
</itinerary>
<itinerary distance="0.0" travelTime="0.0" />
</node>
</itineraryMatrix>
</scenario>

```

## APÊNDICE B – CÓDIGO MODELO DE DADOS

Modelo de dados gerado a partir do cenário do apêndice A.

```
1 LimiteTempo = 1200; //20 minutos
2 NumeroNos = 3;
3 NumeroVeiculos = 2;
4 CapPesoMax = [1100 7000];
5 CapVolMax = [7.5 30];
6 QPesoEntrega = [0 600 0];
7 QPesoColeta = [0 0 300];
8 QVolEntrega = [0 3 0];
9 QVolColeta = [0 0 2];
10 TempoServico = [0 30 30];
11 MaximoJanelasTempo = 1;
12 InicioJT = [[0
13             [0
14             [0]];
15 FimJT = [[540
16           [540
17           [540]];
18 TempoViagem = [[0 20 22]
19                [17 0 12]
20                [22 15 0]];
```

## APÊNDICE C – CÓDIGO MODELO MATEMÁTICO

```

1  /*****
2  * Universidade Tecnológica Federal do Paraná
3  * Trabalho de Conclusão de Curso
4  * Engenharia de Computação
5  *
6  * Aluno: William Hitoshi Tsunoda Meira
7  * Orientador: Prof. Dr. Leandro Magatão
8  * Data de Criação: 20/08/2013
9  *****/
10
11 int LimiteTempo = ...;    //Limite de tempo para executar o modelo
12 float TempoExecucao = 0; //Tempo de execução do modelo em segundos
13 int status = 0;         //Estado da solução do modelo
14
15 int NumeroNos = ...;    //número de Nós [#]
16 range Nos = 1..NumeroNos; //conjunto contendo todos os nós da rede
17 int NumeroVeiculos = ...; //número de veículos (rotas distintas) [#]
18 range Veiculos = 1..NumeroVeiculos; //conjunto contendo todos os veículos da rede
19 int TempoViagem[Nos][Nos] = ...; //tempo de viagem entre cada um dos nós [min]
20 int TempoServico[Nos] = ...; //tempo de serviço [min]
21
22 float CapPesoMax[Veiculos]=...; //capacidade máxima do veículo - peso [kg]
23 float QPesoColeta[Nos]=...; //quantidade de mercadoria a coletar [kg]
24 float QPesoEntrega[Nos]=...; //quantidade de mercadoria a entregar - demanda [kg]
25
26 float CapVolMax[Veiculos]=...; //capacidade máxima do veículo - volume [m3]
27 float QVolColeta[Nos]=...; //quantidade de mercadoria a coletar [m3]
28 float QVolEntrega[Nos]=...; //quantidade de mercadoria a entregar - demanda [m3]
29
30 int MaximoJanelasTempo = ...; //número máximo de janelas [#]
31 range NumJT = 1..MaximoJanelasTempo; //conjunto com todas as janelas possíveis dos clientes
32 float InicioJT[Nos][NumJT] = ...; //tempos de início de cada janela em cada nó [min]
33 float FimJT[Nos][NumJT] = ...; //tempos de final de cada janela em cada nó [min]
34
35 // Cálculo dos valores de 'M' para formulações Big-M [#]
36 float MQ = 3*(sum (i in Nos) (QPesoColeta[i] + QPesoEntrega[i]));
37 float MV = 3*(sum (i in Nos) (QVolColeta[i] + QVolEntrega[i]));
38 float MJ = 3*(max (i in Nos, jt in NumJT) FimJT[i][jt]);
39
40 // Variáveis de Decisão

```

```

41 dvar boolean rota[Nos][Nos][Veiculos]; //binária que indica se a rota entre n e n' é
    percorrida pelo veículo.
42 dvar int+ tempoChegada[Nos][Veiculos]; //instante de tempo de chegada do veículo ao nó i.
43
44 dvar float+ quantPeso[Nos][Nos][Veiculos]; //carga presente no veículo v quando sai do nó n
    para o nó n2 [kg].
45 dvar float+ quantVolume[Nos][Nos][Veiculos]; //volume presente no veículo v quando sai do nó n
    para o nó n2 [m3].
46
47 dvar boolean emJanela[Nos][NumJT]; //binária para detectar em qual janela está a chegada no
    nó.
48
49 dvar float+ relaxJanela[Nos][Veiculos]; //total de estouro de janela de tempo no decorrer das
    rotas.
50 dvar float+ relaxPeso [Nos][Nos][Veiculos]; //excesso de carga presente no veículo v quando
    sai do nó n para o nó n2 [kg].
51 dvar float+ relaxVolume[Nos][Nos][Veiculos]; //excesso de volume presente no veículo v quando
    sai do nó n para o nó n2 [m3].
52
53 dvar float+ objetivo;
54
55 //-----
56 // Pré-processamento
57 //-----
58 float TempoInicial;
59 execute load_ExecTime{
60     var Tempo = new Date();
61     TempoInicial=Tempo;
62
63     cplex.tilim = LimiteTempo; //Definindo o limite de tempo para executar o modelo.
64     cplex.MIPEmphasis = 1; //Definindo a ênfase do processo de busca (1: factibilidade).
65 };
66
67 // Função Objetivo
68 // Minimizar o tempo total que os veículos levam para realizar as suas rotas.
69 minimize
70     objetivo +
71     10 * MJ * (sum(i,j in Nos, v in Veiculos) (relaxPeso[i][j][v]+relaxVolume[i][j][v])) +
72     100 * MJ * (sum(i,j in Nos, v in Veiculos) (relaxVolume[i][j][v])) +
73     MJ * (sum(i in Nos, v in Veiculos) relaxJanela[i][v]);
74
75 subject to {
76
77 objetivo == sum(v in Veiculos) tempoChegada[1][v];
78
79 // (1) O veículo inicia viagem no nó 1, depósito.
80 forall(v in Veiculos)
81     sum(j in Nos) rota[1][j][v] == 1;
82
83 // (2) O veículo termina a rota no nó 1, depósito.
84 forall(v in Veiculos)

```

```

85     sum(i in Nos) rota[i][1][v] == 1;
86
87 // (3) O veículo chega só uma vez a um cliente.
88 // Todos os clientes devem ser atendidos.
89 forall(j in Nos: j>1)
90     sum(i in Nos: i!=j, v in Veiculos) rota[i][j][v] == 1;
91
92 // (4) Se um veículo chega a um cliente tem de sair desse cliente para outro nó
93 // Garante que cada cliente j é visitado somente uma vez por um veículo v
94 // proveniente da localização i , podendo i representar um outro cliente ou o depósito.
95 forall(j in Nos, v in Veiculos)
96     (sum(i in Nos:i!=j)rota[i][j][v] -
97     sum(k in Nos:k!=j)rota[j][k][v] ) == 0;
98
99 // (5) Cálculo do tempo de chegada a cada nó
100 // Um serviço num nó só pode ser realizado depois do nó visitado imediatamente
101 // antes ter sido servido e o veículo se ter deslocado até ao presente nó.
102 forall (v in Veiculos, i in Nos, j in Nos : j!=i && i==1)
103     tempoChegada[j][v] >= (TempoServico[i] + TempoViagem[i][j]) - MJ*(1- rota[i][j][v]);
104
105 forall (v in Veiculos, i in Nos, j in Nos : j!=i && i>1)
106     tempoChegada[j][v] >= (tempoChegada[i][v] + TempoServico[i] +
107     TempoViagem[i][j]) - MJ*(1-rota[i][j][v]);
108
109
110 // (6) O tempo de chegada só deve assumir valor diferente de zero
111 // se existir rota ativa entre i e j de um veículo v.
112 forall (v in Veiculos, j in Nos)
113     tempoChegada[j][v] <= MJ * sum(i in Nos : j!=i) rota[i][j][v];
114
115 // (7), (8) e (9)
116 // O veículo deve chegar a um cliente dentro de uma janela temporal válida, se existir a rota.
117 forall(v in Veiculos, j in Nos, jt in NumJT)
118     tempoChegada[j][v] >= InicioJT[j][jt] -
119     (2 - emJanela[j][jt] - sum(i in Nos:j!=i)rota[i][j][v])*MJ - relaxJanela[j][v];
120
121 forall(v in Veiculos, j in Nos, jt in NumJT)
122     tempoChegada[j][v] <= FimJT[j][jt] + relaxJanela[j][v] + (1-emJanela[j][jt])*MJ;
123
124 forall(j in Nos)
125     sum (jt in NumJT) emJanela[j][jt] >=1;
126
127 // (10) e (11) No momento que o veículo v sai do depósito para realizar as entregas,
128 // transporta a carga a entregar a N fornecedores que constituem a rota associada
129 // ao referido veículo. Esta carga não ultrapassa a capacidade do veículo.
130 forall(v in Veiculos)
131     sum(j in Nos:j>1) quantPeso[1][j][v] == sum (i,j in Nos: i!=j) rota[i][j][v]*QPesoEntrega[j];
132
133 forall(i,j in Nos, v in Veiculos)
134     quantPeso[i][j][v] <= rota[i][j][v]*CapPesoMax[v] + relaxPeso[i][j][v];
135

```

```

136 // (12) Quando o veículo v retorna ao depósito, transporta a carga carregada nas localizações
137 // do conjunto de fornecedores visitados. Esta carga não excede a capacidade do veículo em
    questão.
138 forall(v in Veiculos)
139     sum(i in Nos:i>1) quantPeso[i][1][v] == sum(i,j in Nos:j!=i) rota[i][j][v]*QPesoColeta[j]
140         + (sum(j in Nos:j>1) quantPeso[1][j][v] - sum(i,j in Nos: j!=i) rota[i][j][v]*
            QPesoEntrega[j]);
141
142 //(13) Balanço de massa dos nós deve ser respeitado.
143 forall(i,j,k in Nos, v in Veiculos: (i!=j)&&(j!=k)&&(j>1) )//&&(k>1)
144     quantPeso[j][k][v] <= quantPeso[i][j][v] - QPesoEntrega[j] + QPesoColeta[j] + (1 - rota[i][j][
    v])*MQ;
145
146
147 // (14) e (15) No momento que o veículo v sai do depósito para realizar as entregas,
148 // transporta a carga a entregar a N fornecedores que constituem a rota associada
149 // ao referido veículo. Esta carga não ultrapassa o volume máximo do veículo.
150 forall(v in Veiculos)
151     sum(j in Nos:j>1) quantVolume[1][j][v] == sum(i,j in Nos:j!=i) rota[i][j][v]*QVolEntrega[j];
152
153 forall(i,j in Nos, v in Veiculos)
154     quantVolume[i][j][v] <= rota[i][j][v]*CapVolMax[v] + relaxVolume[i][j][v];
155
156 // (16) Quando o veículo v retorna ao depósito, transporta a carga carregada nas localizações
157 // do conjunto de fornecedores visitados. Esta carga não excede o volume máximo do veículo em
    questão.
158 forall(v in Veiculos)
159     sum(i in Nos:i>1) quantVolume[i][1][v] == sum(i,j in Nos:j!=i) rota[i][j][v]*QVolColeta[j] +
160         (sum(j in Nos:j>1) quantVolume[1][j][v] - sum(i,j in Nos: j!=i) rota[i][j][v]*
            QVolEntrega[j]);
161
162 // (17) Balanço de volume dos nós deve ser respeitado.
163 forall(i,j,k in Nos, v in Veiculos: (i!=j)&&(j!=k)&&(j>1))
164     quantVolume[j][k][v] <= quantVolume[i][j][v] - QVolEntrega[j] + QVolColeta[j] + (1-rota[i][j][
    v])*MV;
165
166
167 //-----
168 // (xx) Inequação auxiliar para zerar os valores das variáveis de rotas que estabelecem ciclos
    ao mesmo nó,
169 // a exceção do nó de origem, onde veículos não utilizados podem fazer rotas 'dummy'.
170 forall(i in Nos, v in Veiculos: i>1)
171     rota[i][i][v]==0;
172
173 } // Fim do Modelo
174
175
176 /*-----*/
177 //Pós-processamento
178 /*-----*/
179 execute{

```



```
180     status = cplex.getCplexStatus();
181     var TempoFinal = new Date();
182         TempoExecucao = (TempoFinal - TempoInicial)/1000;
183     };
```