

INSTITUTO MILITAR DE ENGENHARIA

NEI DE ASSIS

**MODELO DE SISTEMA DE INFORMAÇÃO EM AMBIENTE LIVRE
PARA O TRANSPORTE MULTIMODAL DE CARGA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia de Transportes do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Engenharia de Transportes.

Orientadores: Prof. Luiz Antônio Silveira Lopes – D.Sc. e Prof. Altair dos Santos Ferreira Filho - D.Sc.

Rio de Janeiro

2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

c2007

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 – Praia Vermelha
Rio de Janeiro – RJ CEP: 222290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

Assis, Nei de

Modelo de Sistema de Informação em Ambiente Livre para o Transporte Multimodal de Carga / Nei de Assis – Rio de Janeiro: Instituto Militar de Engenharia, 2007.
168p.: il., tab.

Dissertação (mestrado) – Instituto Militar de Engenharia – Rio de Janeiro, 2007.

1. Transporte Multimodal. 2. Sistema de Informação em Ambiente Livre. 3. Transporte Multimodal de Carga. I. Título. II. Instituto Militar de Engenharia.

CDD

INSTITUTO MILITAR DE ENGENHARIA

NEI DE ASSIS

**MODELO DE SISTEMA DE INFORMAÇÃO EM AMBIENTE LIVRE
PARA O TRANSPORTE MULTIMODAL DE CARGA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia de Transportes do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Engenharia de Transportes.

Orientadores: Prof. Luiz Antônio Silveira Lopes – D.Sc. e Prof. Altair dos Santos Ferreira Filho – D.Sc.

Aprovada em 26 de fevereiro de 2007 pela seguinte Banca Examinadora:

Prof. Luiz Antônio Silveira Lopes – D.Sc. do IME - Presidente

Prof. Altair dos Santos Ferreira Filho – D.Sc. do IME

Prof. Carlos David Nassi – DR. Ing. da COPPE/UFRJ

Rio de Janeiro

2007

À minha mãe, Guilhermina, e ao meu pai, Assis (*In memorium*), exemplos de dedicação, amor e perseverança, e à minha companheira de vida, Vanda, que me apoiou e me encorajou no cumprimento desta jornada.

AGRADECIMENTOS

A DEUS, pela força e capacidade para alcançar os meus objetivos nesta vida.

Ao Instituto Militar de Engenharia, que me deu a honra e o privilégio de cursar o Mestrado em Engenharia de Transportes.

À CAPES e ao Banco Santander, pelo fomento à pesquisa.

Aos professores Silveira Lopes e Ferreira Filho, pela orientação neste trabalho, e pelo estímulo que me deram para desenvolvê-lo.

A todos os professores da PG Transportes do IME, que ao transmitirem os seus conhecimentos, tornaram possível esta dissertação.

A todos os colegas de mestrado de 2005 e 2006, em especial ao José Carlos, Macedo, Natália, Giovanni, Olívio, Denise, Danilo, Erblai, Giuseppe e Siquara, pela cooperação e companhia agradável que sempre me proporcionaram.

Ao SgtOazem e André Medeiros, pela ajuda e apoio inestimáveis prestados na secretaria da PG Transportes.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	09	
LISTA DE TABELAS	10	
1	INTRODUÇÃO	14
1.1	TEMA	14
1.2	DELIMITAÇÃO DO TEMA	14
1.3	PROBLEMA	15
1.4	HIPÓTESE	15
1.5	JUSTIFICATIVA	16
1.6	OBJETIVO	16
1.7	ORGANIZAÇÃO DO TRABALHO	17
2	TRANSPORTE MULTIMODAL DE CARGA	18
2.1	INTRODUÇÃO	18
2.2	CARACTERIZAÇÃO DE MODAIS	19
2.2.1	MODAL FERROVIÁRIO	22
2.2.1.1	HISTÓRICO	22
2.2.1.2	ASPECTOS GERAIS	23
2.2.2	MODAL RODOVIÁRIO	25
2.2.2.1	HISTÓRICO	25
2.2.2.2	ASPECTOS GERAIS	26
2.2.3	MODAL AQUAVIÁRIO	29
2.2.3.1	TRANSPORTE MARÍTIMO	30
2.2.3.2	TRANSPORTE FLUVIAL	33
2.2.3.3	TRANSPORTE LACUSTRE	34
2.2.4	MODAL AEROVIÁRIO	35
2.2.5	MODAL DUTOVIÁRIO	36
2.2.6	TERMINAIS	38
2.3	SISTEMA DE INFORMAÇÃO PARA O TRANSPORTE	40

3	MODELAGEM DE SISTEMA DE INFORMAÇÃO EM AMBIENTE LIVRE	45
3.1	INTRODUÇÃO	45
3.2	AMBIENTE LIVRE	47
3.2.1	LEGISLAÇÃO DE PROGRAMAS ABERTOS	52
3.3	SISTEMA DE BANCO DE DADOS	54
3.3.1	SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS (SGBD)	56
3.3.2	POSTGRESQL	57
3.3.3	POSTGIS	58
3.3.4	MAPSERVER	59
3.4	MODELAGEM DE SISTEMA DE BANCO DE DADOS	60
3.4.1	MODELO CONCEITUAL DE BANCO DE DADOS	60
3.4.2	MODELO LÓGICO DE BANCO DE DADOS	64
3.4.3	MODELO FÍSICO DE BANCO DE DADOS	66
3.5	LINGUAGEM DE MODELAGEM UNIFICADA – UML	66
3.5.1	FERRAMENTA PARA DIAGRAMAÇÃO UML	67
4	MODELAGEM PROPOSTA E PROTOTIPAÇÃO	78
4.1	INTRODUÇÃO	78
4.2	DIAGRAMA DE CASOS DE USO COM DESCRIÇÃO	78
4.3	DIAGRAMA DE ATIVIDADES	80
4.4	DIAGRAMA DE CLASSE	81
4.5	PROJETO CONCEITUAL	83
4.6	PROTOTIPAÇÃO	85
4.6.1	METODOLOGIA	91
4.6.2	IMPLEMENTAÇÃO DO PROTÓTIPO	99
5	CONCLUSÕES	101
6	REFERÊNCIAS BIBLIOGRÁFICAS	103
7	APÊNDICES	105

7.1	APÊNDICE 1: DESCRIÇÃO DE CASOS DE USO.....	106
7.2	APÊNDICE 2: REFERÊNCIAS DE SQL.....	136
7.3	APÊNDICE 3: TELAS DO PROTÓTIPO.....	140
7.4	APÊNDICE 4: JSP/JAVA.....	146
7.5	APÊNDICE 5: CÓDIGOS DE OPERAÇÕES COM O MAPSERVER.....	161

LISTA DE ILUSTRAÇÕES

FIG.2.1	Sistema Ferroviário Nacional	25
FIG.2.2	Estrutura de Modelo Conceitual para o PLANMOB	42
FIG.3.1	Componentes Básicos de um Sistema de Informação Gerencial	46
FIG.3.2	Sistema de Informação – Esquema de Operações	46
FIG.3.3	Evolução Dos SBD	55
FIG.3.4	Exemplo De Modelo Conceitual	61
FIG.3.5	Representação Gráfica De Relacionamento	62
FIG.3.6	Auto-relacionamento	62
FIG.3.7	Projeto de Sistemas Com Foco Nos Dados	64
FIG.3.8	Janela Principal do Modelador Umbrello UML	68
FIG.3.9	Janela do Diagrama de Caso de Uso – UML	70
FIG.3.10	Janela do Diagrama de Classe – UML	72
FIG.3.11	Janela do Diagrama de Atividade – UML	76
FIG.4.1	Diagrama de Casos de Uso	79
FIG.4.2	Diagrama de Atividade	80
FIG.4.3	Diagrama de Classe	82
FIG.4.4	Diagrama E-R	84
FIG.4.5	Esquema de Estrutura Funcional	95
FIG.4.6	Janela Principal do Protótipo	96
FIG.4.7	Solicitação de Cadastro	97
FIG.4.8	Janela do Gerente do Sistema	98
FIG.4.9	Janela do Pesquisador do Sistema	98

FIG.4.10	Janela de Localização das 10 Maiores Áreas da Terra	99
FIG.7.1	Janela de Localização de Terminais	140
FIG.7.2	Janela Cadastro de Terminais	140
FIG.7.3	Janela de Seleção de Veículos	141
FIG.7.4	Janela de Localização de Veículos Aquaviários	141
FIG.7.5	Cadastramento de Veículos Aquaviários	142
FIG.7.6	Janela de Requisição de Mudanças	142
FIG.7.7	Janela de Localização de Usuários	143
FIG.7.8	Janela de Cadastramento de Usuários	143
FIG.7.9	Janela de Requisição de Mudança pelo Usuário	144
FIG.7.10	Janela de Localização de um País	144
FIG.7.11	Janela de Municípios entre Ceará e Piauí	145
FIG.7.12	Janela de Ferrovias do Nordeste	145

LISTA DE TABELAS

TAB.1.1	Dados Estatísticos sobre o Setor de Transporte de Cargas	15
TAB.2.1	Quantidade de carga transportada em Ton-Km, por modo de transporte 1996-2000	20
TAB.2.2	Composição percentual da carga transportada, em Ton-Km, por modo de transporte	20
TAB.2.3	Matriz do Transporte de Cargas	22
TAB.2.4	Principais Mercadorias Transportadas	24
TAB.2.5	Principais Mercadorias Transportadas	27
TAB.2.6	Cargas Transportadas pelo Modal Dutoviário	37
TAB.3.1	Equivalência de Softwares Proprietários	51

RESUMO

O propósito deste trabalho é pesquisar sobre a Tecnologia da Informação existente aplicada à área de transporte, apontando-a como uma das causas da deficiência da infra-estrutura de apoio ao transporte de cargas no Brasil e apresentar um modelo de sistema de informação que atualize e amplie os estudos existentes, tornando-os compatíveis com as novas tecnologias da informação. Pretende-se, com isso, colaborar na minimização da situação crítica da infra-estrutura de apoio.

Para se atingir o propósito desta pesquisa, inicialmente, são apresentados fundamentos da caracterização dos modais e do sistema de informação para o transporte multimodal de carga no Brasil. Em seguida, é apresentado um estudo sobre o estado-da-arte da tecnologia da informação em ambiente livre que inclui tópicos sobre o desenvolvimento e o gerenciamento de sistemas de banco de dados com softwares de código aberto e a modelagem UML dos dados aplicados a projetos orientados a objeto.

Na seqüência é proposto um modelo de sistema de informação para o transporte multimodal de carga, cuja implementação é demonstrada através de um protótipo criado numa plataforma de desenvolvimento Java J2EE para aplicações Web.

A conclusão do trabalho mostra que esta pesquisa amplia os horizontes de estudos na área de Sistema de Informação de Transporte, tornando-os compatíveis com o que há de mais atualizado na Tecnologia da Informação.

ABSTRACT

The purpose of this work is to research on the existing Information Technology applied to Transportation, pointing it out as one of the causes for the deficiency of the supportive infra-structure to the load transportation in Brazil, and to present an information system model that updates and extends the existing studies, turning them out compatibles with the new information technologies. We intend to cooperate in minimizing the supportive infra-structure critical situation.

For this purpose, it is presented, at first, the fundamentals for the modal characterization and for the load multimodal transportation information system in Brazil. Next, it is presented a study on the “state-of-the-art” of the information technology in free environments including topics about database systems development and management with open source softwares and data UML modelling applied to Object-Oriented Projects.

Continuing, it is proposed a load multimodal transportation information system model with its implementation demonstrated in a prototype created on a Java J2EE development platform for Web applications.

The conclusion shows that this research expands the horizon studies on Transportation Information System, turning them compatibles with the most updates Information Technologies.

1 INTRODUÇÃO

1.1 TEMA

Este trabalho propõe pesquisar sobre a Tecnologia da Informação existente, aplicada à área de transporte, apontando-a como uma das causas da deficiência da infra-estrutura de apoio ao transporte de cargas no Brasil.

1.2 DELIMITAÇÃO DO TEMA

Segundo os estudos realizados pela Confederação Nacional de Transportes (CNT) em conjunto com o Centro de Estudos em Logística (Coppead), o transporte de cargas no Brasil apresenta sintomas que apontam para graves problemas que comprometem o desenvolvimento econômico e social do Brasil. Um desses sintomas é a deficiência da infra-estrutura de apoio, que inclui, entre outras causas, um banco de dados do setor de transportes deficiente e uma tecnologia de informação desatualizada, com ferramentas de alto custo de aquisição e, muitas vezes, pouco utilizadas.

Por outro lado, o Setor de Transporte de Cargas Brasileiro sofre com a ausência de uma política abrangente de coleta e análise contínua dos seus principais indicadores de desempenho. A TAB.1.1 mostra que, no Brasil, há uma grande deficiência de dados estatísticos sobre o setor, causada, principalmente, pela ausência de uma entidade que seja responsável pela coleta de estatísticas, análises abrangentes e contínuas sobre o transporte de cargas e seus efeitos no desenvolvimento do país. Na realidade, os dados estatísticos de transporte estão disponíveis, mas não possuem uma uniformização das informações, tornando-os de difícil interpretação e análise.

Outro fato agravante neste quadro é que, no Brasil, apenas 46,8% das empresas de transporte rodoviário de carga informatizaram o controle de suas frotas.

TAB.1.1: Dados Estatísticos sobre o Setor de Transporte de Cargas

ESTATÍSTICAS DE	EUA	BRASIL
Alocação de Recursos Governamentais e Privados	●	◐
Consumo de Energia do Setor de Transporte	●	◐
Infra-estrutura por modal	●	◐
Quilometragem percorrida	●	◐
Toneladas Transportadas	●	◐
Produção de Transporte (TKU)	●	◐
Contribuição do Transporte (% do PIB)	●	◐
Faturamento do setor de transporte	●	◐
Empregos do setor de transporte	●	◐
Nível de estoque na indústria	●	◐
Vendas e importações de equipamentos de transp.	●	●
Registro de Equipamentos de Transporte	●	○
Roubo de Carga	◐	◐
Acidentes e Mortes no Trânsito	●	◐
Emissões de poluentes no transporte	●	○
Desastres ambientais	●	○

● Dado Disponível ◐ Dado Parcialmente Disponível ○ Dado Indisponível

(Fonte: COPPEAD – www.coppead.ufrj.br, 2006)

1.3 PROBLEMA

Segundo FERREIRA FILHO (1998), para que o setor de transporte possa planejar e operar seus diversificados sistemas é necessário coletar e processar um grande número de dados referentes a vias, veículos, capacidades, disponibilidades, equipamentos, pessoal, tipos de carga, manutenções diversas e outros. O Sistema de Informação é um dos mecanismos existentes que auxilia no processamento desses dados. Entretanto, para que as empresas envolvidas com transporte possam planejar mais adequadamente os seus negócios, otimizando o tempo de tomada de decisão gerencial com uma baixa margem de erro, é necessário que o Sistema de Informação possa processar os dados de uma forma rápida, clara e precisa.

Esta pesquisa pretende modelar um Sistema de Informação que atenda estes requisitos, viabilizando os ganhos de produtividade e minorando uma das causas da deficiência da infra-estrutura de apoio.

1.4 HIPÓTESE

Espera-se que esta modelagem forneça subsídios para demonstrar a funcionalidade do modelo visualizado no PROBLEMA e que será verificado pela pesquisa.

1.5 JUSTIFICATIVA

Segundo DIOGO (1994), a primeira manifestação no sentido de implantar um Sistema de Informação para os Transportes no Brasil é encontrada em 1973, no Plano Básico de Desenvolvimento Científico Tecnológico (PBDCT) que atribuiu ao Grupo de Estudos para Integração de Política de Transportes (GEIPOT) a definição e a implantação de um Sistema de Informação de Transportes. Neste ano foi estabelecida a organização de um Centro Nacional de Informações e Documentação de Transportes (CIDOT). Em 1975, é proposto o Sistema de Informações de Transportes (SIT). Em 1977 é proposta uma estrutura para o SIT com classes de informações gerenciais, técnico-científicas e documentais. Em 1989, a Secretaria Especial de Informática (SEI), em trabalho com o Ministério dos Transportes, produziu uma Proposta de Plano Setorial de Informática nos Transportes e propôs o chamado Sistema Integrado de Informações em Transportes (SIIT), organizado em três níveis de execução e um Conselho Diretor:

- Centro de Informação (CI);
- Centros de Informação Modais ou Multimodais (CIM), para integrar os CI;
- Centro de Informação em Transportes (CIT), para integrar os CIM;
- Conselho Diretor da Política de Informações em Transportes (CDPIT), que gerencia o SIIT.

Portanto, desde a criação do PBDCT, em 1973, aliada a necessidade de se criar um sistema de informação que minimizasse as causas apontadas no item 1.2 deste trabalho, mais a existência de uma legislação que dispõe sobre a utilização de programas abertos pelos entes de direito público e de direito privado - PROJETO DE LEI N.º 2.269 DE 1999, foram as razões que levaram ao desenvolvimento desta dissertação..

1.6 OBJETIVO

O fim teórico e prático que se propõe alcançar com esta pesquisa é apresentar um modelo de sistema de informação para o transporte multimodal de carga, com o seu desenvolvimento e implantação em ambiente livre, atualizando e ampliando os estudos existentes de sistema de informação para os transportes, tornando-os compatíveis com as novas Tecnologias de Informação. Os passos para se atingir esse objetivo são:

- Pesquisar as características gerais dos modais e dos terminais de carga no Brasil, a legislação pertinente e os sistemas de informação aplicados ao transporte de carga;
- Apresentar o estado-da-arte da tecnologia da informação em ambiente livre;
- Propor um modelo de sistema de informação que tenha aplicabilidade a um projeto orientado a objeto;
- Avaliar a operacionalidade do modelo criado através de um protótipo que seja utilizado na WEB.

Essa dissertação pretende colaborar com a minimização do problema apresentado, revertendo a situação crítica que passa o setor e sugerindo um método de desenvolvimento de sistema de informação que seja viável, num ambiente robusto, flexível, aberto e de baixo custo como o ambiente livre.

1.7 ORGANIZAÇÃO DO TRABALHO

A presente dissertação está organizada sob a forma de capítulos, estruturada na seguinte ordem:

- Capítulo 1 – Introdução. Contextualiza o assunto, tecendo as considerações gerais, o objetivo, a justificativa e a forma como o trabalho está organizado.
- Capítulo 2 – Transporte Multimodal de Carga. Apresenta a caracterização dos modais e dos terminais e comenta sobre o sistema de informação para o transporte de carga.
- Capítulo 3 – Modelagem de Sistema de Informação em Ambiente Livre. Apresenta o estado-da-arte da tecnologia da informação em ambiente livre, fundamentando o sistema de banco de dados e a modelagem de dados.
- Capítulo 4 – Modelagem Proposta. Propõe o desenvolvimento de um modelo de sistema de informação para transporte multimodal de carga que tenha aplicabilidade em um projeto orientado a objeto.
- Capítulo 5 – Prototipação. Implementa o modelo criado através de um protótipo que demonstra e avalia a operacionalidade do mesmo na WEB.
- Capítulo 6 – Conclusões. Demonstra os resultados alcançados e faz sugestões para trabalhos futuros.

2 TRANSPORTE MULTIMODAL DE CARGA

2.1 INTRODUÇÃO

O Brasil criou a Lei Nº 9.611/98 que tem o objetivo de reger o Transporte Multimodal de Cargas.

Segundo DEMARIA (2004), devido à aprovação e regulamentação desta Lei, o país começou a viver uma situação diferente em relação à forma de entrega de uma mercadoria. Esta Lei foi promulgada em 1998 após cerca de dez anos de estudos, mas só foi regulamentada dois anos e dois meses depois pelo Decreto Nº 3.411 de 13/04/2000, apesar de estar prevista na lei que a regulamentação deveria acontecer em 180 dias. Os especialistas da área apontavam a questão tributária, os interesses de setores da economia e a falta de conhecimento dos empresários como os maiores entraves para a regulamentação.

A regulamentação foi uma fase complicada, principalmente por essa questão envolver vários ministérios, além dos estados. O Imposto sobre Circulação de Mercadorias e Serviços (ICMS) é um ponto que ainda representa um entrave, e continuará sendo até a sua resolução na esfera jurídica. Outro ponto em questão é o seguro multimodal obrigatório, ainda inexistente devido à falta de experiência do mercado segurador, que por isso não consegue reunir parâmetros para estabelecer cláusulas e taxas.

O real objetivo desse instrumento legal foi inserir o Brasil na modernidade, já que a multimodalidade é valorizada praticamente em todo o mundo, pois fornece aos importadores e exportadores a oportunidade de gerenciar suas mercadorias em uma única e consolidada fonte de informações sobre o andamento dos processos e a localização de sua carga.

O Conhecimento de Transporte Multimodal de carga evidencia um contrato de transporte multimodal que rege toda a operação, desde o recebimento da carga até a sua entrega no destino.

O Art. 2º da Lei 9.611-1998 define o Transporte Multimodal de Cargas como aquele que, regido por um único contrato, utiliza duas ou mais modalidades de transporte, desde a origem até o destino, e é executado sob a responsabilidade única de um Operador de Transporte Multimodal (OTM), que pode realizar o trajeto por meios próprios ou subcontratando transportadores que o façam.

O transporte multimodal pode incluir os serviços de coleta, unitização, desunitização,

movimentação, armazenagem, consolidação e desconsolidação de cargas.

Já o transporte intermodal utiliza dois ou mais modais no transporte de cargas desde a origem ao destino final (door-to-door) mediante a celebração de um contrato de transporte para cada etapa do itinerário. O transportador de cada etapa/modal é responsável por danos, avarias, atrasos ou extravio da carga durante a sua respectiva prestação de serviços (Guia De Logística – www.guiadelogistica.com.br, 2006).

GUIMARÃES (1999) afirma que tanto o termo Transporte Intermodal como Transporte Multimodal são usados para definir transportes que utilizam mais de um modal. A diferença está em como os autores interpretam as operações realizadas pelos mesmos. Para uns, a diferença está na documentação e responsabilidade do transportador enquanto, que para outros, nos equipamentos utilizados para transferência de carga de um modal para outro.

2.2 CARACTERIZAÇÃO DE MODAIS

Segundo BASTOS (2003), o que deu maior estímulo ao desenvolvimento em muitas nações no século XIX foram as ferrovias. As ferrovias eram o único meio de transporte utilizado para atingir locais, velocidade e segurança que não eram possíveis para o transporte hidroviário, então predominante.

No Brasil, o transporte ferroviário predominou por longo tempo, de 1854 até 1954. Após 100 anos, o transporte ferroviário começou a sofrer competição com a rodovia por ser um sistema bem mais ágil, e por esse motivo, em menos de 20 anos grande parte da rede ferroviária foi desativada, dando lugar ao transporte rodoviário (GÔMARA – 2000). Vários fatores econômicos contribuíram para a ocorrência desse evento, principalmente pela perda da competitividade das ferrovias em relação ao transporte rodoviário. Normalmente, as ferrovias possuíam grande extensão, sofriam forte regulação estatal e os preços das tarifas eram baseados no valor das mercadorias transportadas (CAIXETA-FILHO E MARTINS – 2001).

Para CAIXETA-FILHO E MARTINS (2001): “Esses aspectos, comparados com o transporte de baixa escala, com pouca intervenção governamental e com fretes baseados nos custos, centralizados pelo transporte rodoviário, foram decisivos para a perda de importância das ferrovias num ambiente de competição entre os modais”.

BASTOS (2003) afirma que, hoje, o transporte rodoviário atinge praticamente todo o território nacional, tendo uma função importante na economia. A concorrência nessa área é

bastante acirrada, apesar das transportadoras sofrerem algumas dificuldades como: condição inadequada das estradas, falta de um melhor planejamento para o setor por parte dos órgãos governamentais e alguns aspectos operacionais.

A TAB.2.1 mostra um quadro comparativo de como o rodoviário é o muito mais utilizado do que os outros modais.

TAB.2.1: Quantidade de carga transportada em toneladas-quilômetro, por modo de transporte 1996-2000. (x 10⁶)

Modo de Transporte	1996	1997	1998	1999	2000
Aéreo	2036	1709	2173	2244	2432
Aquaviário	71310	77402	90444	94770	103390
Dutoviário	23528	30431	31609	33131	33246
Ferrovário	128976	138724	142446	140817	155590
Rodoviário	396060	421131	445795	447353	451370
TOTAL	621910	669401	712467	718315	746028

(Fonte: Geipot – www.Geipot.Gov.br, 2006)

Para permitir uma melhor análise do objetivo da TAB.2.1, foi desenvolvida a TAB.2.2 com a apresentação de uma composição percentual da carga transportada.

TAB.2.2: Composição percentual da carga transportada, em toneladas-quilômetro, por modo de transporte 1996-2000.

Modo de Transporte	1996	1997	1998	1999	2000
Aéreo	0,33	0,26	0,31	0,31	0,33
Aquaviário	11,47	11,56	12,69	13,19	13,86
Dutoviário	3,78	4,55	4,44	4,61	4,46
Ferrovário	20,74	20,72	19,99	19,6	20,86
Rodoviário	63,68	62,91	62,57	62,29	60,49
TOTAL	100	100	100	100	100

(Fonte: Geipot – www.geipot.gov.br, 2006)

LAMBERT *et al.* (1998) citam que: “Geralmente o transporte rodoviário compete com o aéreo em pequenas cargas e com o ferroviário em grandes cargas”. Mas o transporte rodoviário é predominante. A principal característica desse modal é a flexibilidade que é proporcionada, pois pode trafegar por qualquer estrada e realizar transportes “porta a porta”, diferentemente dos outros modais.

PASSARI (2002) afirma que, a predominância do transporte rodoviário é uma característica do Brasil, já que em países desenvolvidos, como os Estados Unidos da América, Japão e outros países da Europa, apresentam matrizes de transportes com a utilização mais equilibrada dos modais - geralmente a massa transportada pela distância percorrida são parecidas em todos os modais, com exceção do aéreo, que sempre transporta em menor quantidade em razão do seu custo elevado.

Segundo DEMARIA (2004), principalmente no comércio exterior, que em geral contempla maiores distâncias a serem percorridas e a participação de detalhes adicionais ao transporte interno, a escolha do modo de transporte a ser utilizado não deve ser baseada exclusivamente na comparação entre tarifas de frete, mas sim em análise mais ampla, que considere as variáveis que estão ligadas a cada modalidade.

A escolha do modal a ser utilizado na distribuição internacional das mercadorias é um ponto essencial. Por isso, é de suma importância conhecer cada um deles detalhadamente para criação e desenvolvimento de uma logística adequada.

Segundo MENDONÇA E KEEDI (1997), atualmente, os tipos de modais utilizados são:

1. Complexo Terrestre:
 - Rodoviário
 - Ferroviário
2. Complexo Aquaviário:
 - Marítimo
 - Fluvial
 - Lacustre
3. Aéreo
4. Dutoviário

SILVA (2001) observa que, de acordo com o perfil de desempenho das modalidades com relação às características básicas do transporte, cada um dos sistemas analisados possui determinadas áreas onde operam de forma mais conveniente.

Na TAB.2.3, a Agência Nacional de Transportes Terrestres (ANTT) apresenta a seguinte Matriz de Transporte:

TAB.2.3: Matriz do Transporte de Cargas

Modal	Milhões TKU 2000	PARTICIPAÇÃO (%)	Milhões TKU 2001	PARTICIPAÇÃO (%)
Rodoviário	470888	61,50%	485625	61,10%
Ferrovário	155950	20,40%	164809	20,70%
Aquaviário	103390	13,50%	108000	13,60%
Dutoviário	33246	4,30%	33300	4,20%
Aéreo	2432	0,30%	3169	0,40%
TOTAL	765906	100,00%	794903	100,00%

(Fonte: ANTT – www.antt.gov.br/destaques/ANTTEmNumeros20051226.pdf, 2006)

2.2.1 MODAL FERROVIÁRIO

2.2.1.1 HISTÓRICO

Em 1922, existia no país um sistema ferroviário com 29.000 quilômetros de extensão. Daquele ano até 1974, destacaram-se o seguinte importantes acontecimentos na história das ferrovias:

1930 - Introdução da tração elétrica para substituir à tração a vapor em determinados trechos;

1939 - Substituição da tração a vapor pela tração diesel-elétrica;

1940 - Reorganização e saneamento das estradas de ferro existentes, com a encampação de empresas estrangeiras e nacionais, inclusive estaduais, criando a Inspeção Federal de Estradas - IFE, órgão do Ministério de Viação e Obras Públicas, encarregado de gerir as ferrovias e rodovias federais;

1941 - Instituição do Departamento Nacional de Estradas de Ferro - DNEF e do Departamento Nacional de Estradas de Rodagem - DNER, pelo Decreto Lei n.º 3.155, de 28/03/1941;

1942 - Criação da Companhia Vale do Rio Doce - CVRD, que absorveu a Estrada de Ferro Vitória a Minas (construída a partir de 1903);

1957 - Criação da Rede Ferroviária Federal S.A. - RFFSA pela Lei n.º 3.115, de 16/03/1957, unificando administrativamente as 18 estradas de ferro pertencentes à União, que totalizavam 37.000 quilômetros de linhas distribuídas pelo país;

1971 - Unificação das estradas de ferro do Estado de São Paulo, com a criação da FEPASA - Ferrovia Paulista S.A., pela Lei n.º 10.410/SP;

1974 - Extinção, em dezembro, do DNEF e transferência de suas funções para a Secretaria-Geral do Ministério dos Transportes, bem como para a RFFSA.

1992 - Início do processo de desestatização do setor ferroviário, pelo Decreto n.º 473/92, a partir da inclusão da Rede Ferroviária Federal S.A. - RFFSA no Programa Nacional de Desestatização (ANTT – www.antt.gov.br/concessaofer/historico.asp, 2006).

2.2.1.2 ASPECTOS GERAIS

Com o objetivo de otimizar a oferta e melhorar os serviços, o governo federal decidiu colocar em prática, ações voltadas para a privatização, concessão e delegação de serviços públicos de transporte a Estados, Municípios e iniciativa privada. Para tanto, criou e sancionou a Lei n.º 8.031/90, de 12/04/90, e suas alterações posteriores, instituindo o Programa Nacional de Desestatização – PND. O processo de desestatização do setor ferroviário foi iniciado em 10/03/92, a partir da inclusão da Rede Ferroviária Federal S.A. - RFFSA no PND, pelo Decreto n.º 473/92. Com o leilão da Malha Oeste para a Ferrovia Novoeste, em 05/03/96, iniciou-se o processo de concessão das malhas da RFFSA.

O Plano Nacional de Desestatização, relativamente à modalidade ferroviária, tem como principais objetivos:

- Desonerar o Estado;
- Melhorar a alocação de recursos;
- Aumentar a eficiência operacional;
- Fomentar o desenvolvimento do mercado de transportes, e;
- Melhorar a qualidade dos serviços.

(Fonte: ANTT – www.antt.gov.br/concessaofer/aspectos.asp, 2006)

Na TAB.2.4, a Agência Nacional de Transportes Terrestres (ANTT) apresenta as principais mercadorias transportadas em 2002, 2003 e 2004 pelo modo ferroviário:

TAB.2.4: Principais Mercadorias Transportadas

Produto Agregado	2002	2003	2004	Crescimento(%)
Minério de Ferro	192851635	202315600	237695700	17,5
Indústria Siderúrgica	28385328	29460600	31898100	8,3
Cimento	5236490	4801400	4812100	0,2
Indústria Cimenteira	1734794	951000	567200	-40,4
Carvão/Coque	11525714	12631600	13292700	5,2
Granéis Minerais	6336764	8349300	10200700	22,2
Soja e Farelo de Soja	28083227	31504100	29090200	-7,7
Produção Agrícola	7487715	8751400	10209000	16,7
Adubos e Fertilizantes	4375862	5368000	6282300	17
Extração Vegetal e Celulose	2044507	2159300	2304200	6,7
Combustíveis, Derivados do Petróleo e do Álcool	8433036	7687100	7806600	1,6
Contêiner	88133	328200	533400	62,5
Carga Geral-não containerizada	153284	189700	653800	244,6
Outras Mercadorias	24254491	30613700	224300	-26,7
TOTAL	320992000	345111000	377776000	9,5

(Fonte: ANTT – www.antt.gov.br/destaques/ANTTemNumeros20051226, 2006)

De acordo com DEMARIA (2004), o transporte ferroviário liga normalmente países limítrofes. Não tem flexibilidade de percurso por ficar restrito a um único caminho e não é tão ágil quanto o transporte rodoviário. A participação do transporte ferroviário do Brasil com os países latino-americanos é ainda pequena, sendo a diferença de bitola um dos maiores problemas.

O Acordo sobre Transporte Internacional Terrestre (ATIT) define, para os países do Cone Sul (Brasil, Argentina, Paraguai, Uruguai, Bolívia, Chile e Peru), o formulário único de transporte e trânsito aduaneiro para o modal ferroviário, denominado “Conhecimento-Carta de Porte Internacional”.

O frete ferroviário tem base em dois fatores: quilometragem percorrida e peso da

mercadoria. Assim, pode ser calculado pela multiplicação da tarifa por tonelada ou metro cúbico, sendo utilizado aquele que gerar maior receita. Pode também ser aplicada uma taxa de estadia do vagão, cobrada por dia. Não é comum incidirem taxas de armazenagem, manuseio ou qualquer outra, sendo admitida a cobrança de taxa administrativa pelo transbordo.

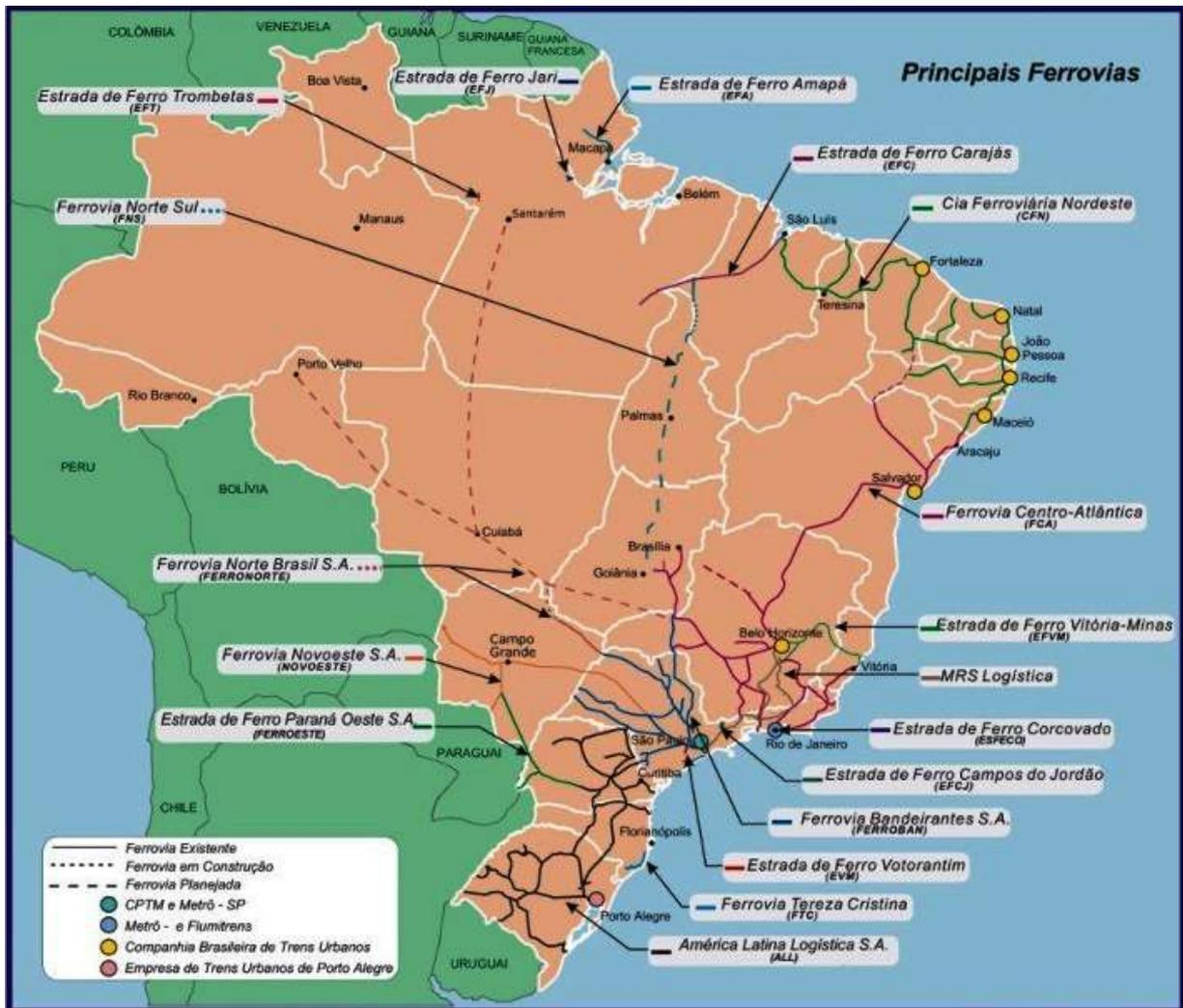


FIG.2.1: Sistema Ferroviário Nacional. (Fonte: ANTT – <http://www.antt.gov.br/concessaofer/mapa.asp>, 2006)

2.2.2 MODAL RODOVIÁRIO

2.2.2.1 HISTÓRICO

O Ministério dos Transportes implantou o Programa de Concessões de Rodovias

Federais com a licitação dos cinco trechos que haviam sido pedagiados (extensão de 854,5 km), ao mesmo tempo em que realizava outros estudos para identificar outros segmentos considerados técnica e economicamente viáveis para inclusão no Programa de Concessões. Na fase inicial, foram analisados 17.247 km de rodovias, dos quais 10.379 foram considerados viáveis para concessão e 6.868 km viáveis somente para a concessão dos serviços de manutenção.

Em 1997/98, foram incluídos no Programa os trechos BR-381/MG/SP, entre Belo Horizonte e São Paulo, e todo o Corredor da BR-116/SP/PR, BR-376/PR e BR-101/SC, entre São Paulo, Curitiba e Florianópolis.

A Lei n.º 9.277, de maio de 1996 (Lei das Delegações), possibilitou que os Estados, os Municípios e o Distrito Federal incluíssem trechos de rodovias federais em seus Programas de Concessão de Rodovias. A Portaria n.º 368/96, do Ministério dos Transportes, definiu os procedimentos para a delegação de rodovias federais aos estados.

Foram assinados Convênios de Delegação com os Estados do Rio Grande do Sul, Paraná, Santa Catarina e Minas Gerais entre 1996 e 1998. Em 1997, o Estado de Mato Grosso do Sul assinou Convênio de Delegação visando a construção de uma Ponte sobre o Rio Paraguai na BR-262/MS. Esta ponte foi financiada com recursos do FONPLATA - Fundo Financeiro dos Países da Bacia do Prata e está sendo explorada mediante a cobrança de pedágio para amortização do financiamento.

Também foram incluídos no Programa, trechos de rodovias federais dos estados de Goiás e do Pará para serem delegados e concedidos. Entretanto, estudos demonstraram que os trechos não tinham viabilidade para concessão e, portanto, foram retirados do Programa (ANTT – www.antt.gov.br/concessaorod/historico.asp, 2006).

2.2.2.2 ASPECTOS GERAIS

Com base na Lei n.º 9.277/96, e pela Agência Nacional de Transportes Terrestres (ANTT), o Programa de Concessão de Rodovias Federais abrange 10.000 quilômetros de rodovias, desdobrado em concessões promovidas pelo Ministério dos Transportes, pelos governos estaduais.

As concessões administradas hoje, diretamente, pela ANTT são constituídas de seis trechos de rodovias já concedidos à iniciativa privada, numa extensão de 1.474,4 km, das

quais cinco foram contratadas pelo Ministério dos Transportes, entre 1994 e 1997, e uma pelo Governo do Estado do Rio Grande do Sul, em 1998, a qual teve o Convênio de Delegação das Rodovias denunciado e o contrato sub-rogado à União em 2000 (ANTT – www.antt.gov.br/concessaorod/aspectos.asp, 2006).

Na TAB.2.5 é apresentada a seguinte Malha Rodoviária:

TAB.2.5: Malha Rodoviária (Plano Nacional de Viação)

RODOVIAS	PAVIMENTADAS	NÃO PAVIMENTADAS	TOTAL
Federais	57933,1	35131,9	92065,0
Estaduais	115426,0	161349,9	276755,0
Municipais	22734,8	1359286,3	1382031,1
TOTAL	196093,9	1555768,1	1751862,0

(Fonte: ANTT – www.antt.gov.br/destaques/ANTTemNumeros20051226, 2006)

Segundo DEMARIA (2004), o transporte rodoviário internacional é aquele efetuado por caminhões ou carretas e normalmente faz ligação entre países limítrofes.

Entre todos os modais, o rodoviário é o mais utilizado no transporte de mercadorias em viagens curtas e médias.

As cargas são transportadas em espaços reservados diretamente com os transportadores, o que pode ser feito juntamente com outras cargas ou isoladamente, quando a carga for suficiente para o espaço total do veículo.

No transporte rodoviário de cargas, podem ser utilizados caminhões (veículos fixos), carretas (veículos articulados), trailers, plataformas (para transporte de contêineres), entre outros. Quase todas as empresas que operam no transporte rodoviário internacional de cargas são associadas à Associação Brasileira de Transportadores Internacionais (ABTI).

O Decreto Nº 99.704, de 20/11/1990, dispõe sobre a execução no Brasil do Acordo sobre Transporte Internacional Terrestre entre Brasil, Argentina, Bolívia, Chile, Paraguai, Peru e Uruguai, que propicia regulamentação conjunta do transporte internacional terrestre no Cone Sul da América, permitindo a garantia de regularidade de atendimento, bem como definições pertinentes a direitos e obrigações de usuários e transportadores.

O Conhecimento Internacional de Transporte Rodoviário (Anexo B) foi aprovado na

XVII Reunião de Ministros de Obras Públicas e Transportes dos países do Cone Sul em setembro de 1990 em Assunção, no Paraguai. É documento de emissão obrigatória, em três vias originais (uma para o remetente, negociável, uma para acompanhar a mercadoria e a última para o transportador).

O CRT funciona como contrato de transporte rodoviário, recibo de entrega e como título de crédito, devendo conter obrigatoriamente as seguintes informações: embarcador, consignatário, locais de origem e destino da mercadoria, ponto de fronteira de liberação da mercadoria e veículo, data de entrega da mercadoria ao transportador, descrição da mercadoria e sua embalagem, com pesos e quantidades, marcas especiais e valor do frete.

É interessante a utilização do Manifesto Internacional de Carga Rodoviária/ Declaração de Trânsito Aduaneiro (MIC/ DTA), formulário único combinado, o qual permite que o desembaraço aduaneiro de importação ocorra no destino final, desde que a carga embarcada ocupe todo o veículo, reduzindo o *transit-time*, sem a necessidade de vistoria de carga em fronteira, mas com a conferência do lacre com o qual deve o veículo efetuar todo o percurso previsto.

Quando o exportador pretende utilizar a via rodoviária como modal de transporte, a empresa deve verificar se a transportadora está realmente autorizada para esse fim, bem como se o seguro foi contratado, já que é obrigatório nesse tipo de transporte.

As tarifas de frete são organizadas individualmente por cada empresa de transporte. A estrutura tarifária básica leva em conta a natureza da mercadoria, o custo do transporte, o peso ou volume e a distância. O frete pode ser calculado por peso, volume ou por lotação de veículo, podendo ser adicionados uma taxa *ad valorem* para mercadorias de alto valor e o custo do seguro rodoviário obrigatório. A simplicidade de funcionamento do transporte rodoviário é o seu ponto forte, pois não apresenta qualquer dificuldade e está sempre disponível para embarques urgentes. Esse modal confere maior flexibilidade, oferecendo algumas vantagens citadas a seguir:

- As vendas podem ser feitas na condição de entrega porta a porta com mais segurança;
- Menos manuseio de carga, já que o caminhão é lacrado no local de carregamento e aberto no local de entrega;
- Rapidez na entrega da carga em curta distância;

- O transporte vai até a carga em vez de obrigar o comerciante a ir até o transporte;
- Possibilidade de utilização de embalagens mais simples e de menor custo.

2.2.3 MODAL AQUAVIÁRIO

Segundo o Ministério dos Transportes, hidrovias, aquavia, via navegável, caminho marítimo ou caminho fluvial são designações sinonímicas, enquanto que hidrovias interiores ou via navegável interior são denominações comuns a rios, lagos ou lagoas navegáveis. Entretanto, devido à falta de expressões como hidrovias interiores artificiais, para denominar aquelas que não eram navegáveis e que adquiriram essa condição em função de obras de engenharia e hidrovias interiores melhoradas para caracterizar as que tiveram suas condições de navegação ampliadas, usa-se genericamente a expressão hidrovias interiores.

As cartas de navegação são mapas delimitadores das rotas de navegação.

Assim como existem estradas de rodagem carroçáveis, não pavimentadas, pavimentadas e rodovias, existem rios fluviáveis, de navegação rudimentar, francamente navegáveis e hidrovias.

Podemos aproveitar as rodovias para, fazendo um paralelismo, caracterizar alguns conceitos hidroviários:

- Balizamento das hidrovias corresponde às faixas que são longitudinalmente pintadas nos pavimentos das rodovias;
- A sinalização de margem das hidrovias pode ser associada às placas que são colocadas às margens das rodovias e que são conhecidas como sinais de trânsito;
- Como os canais de navegação não são materializáveis e as pistas de rolamento das rodovias sim, as hidrovias requerem cartas de navegação para proporcionar segurança às embarcações;
- As rodovias são projetadas para um determinado veículo rodoviário, isto é, para um veículo tipo. As pontes são projetadas considerando que esse veículo tipo tenha no máximo "x" toneladas; os vãos sob os viadutos e passarelas ou os túneis, que esse veículo tenha no máximo "y" metros de altura; e assim por diante. Nas hidrovias, o mesmo se sucede com as embarcações tipo. É necessário ressaltar que embarcação tipo é um artifício de projeto e não condicionante à confecção de embarcações, pelo menos no conjunto de suas características. A embarcação tipo é uma abstração que

reúne as características para as quais a hidrovia é projetada, ou seja, ela é projetada para um comprimento "x" de embarcação, para uma boca "y" e para um calado máximo "z", sendo este para a situação de águas mínimas, que concomitantemente definem uma embarcação hipotética chamada tipo. No que se refere às dimensões, pode-se dizer que as embarcações que serão lançadas e que navegarão na hidrovia em voga devem ter no máximo o comprimento e a largura (Boca) da embarcação tipo e que, no que alude a calados, em condições de águas mínimas, deve navegar com calado no máximo igual ao da embarcação tipo.

Finalizando, cabe informar que existem hidrovias interiores de tráfego apenas diurno e as franqueadas à navegação noturna, em função do tipo de balizamento e de sinalização adotados (Ministério dos Transportes – www.transportes.gov.br, 2006).

DEMARIA (2004) define o transporte aquaviário como a denominação moderna do setor em que estão inseridos os transportes marítimos, fluvial e lacustre.

No Brasil, o Departamento de Marinha Mercante do Ministério dos Transportes é o órgão governamental responsável pelo acompanhamento dessa modalidade, referente à distribuição de linhas e oferta de espaço, aos valores de frete praticados e ao funcionamento das empresas de navegação, editando a regulamentação necessária.

Conforme as normas brasileiras, a navegação pode ser enquadrada numa das seguintes formas:

- Cabotagem: navegação realizada entre portos ou pontos do território brasileiro, utilizando a via marítima ou as vias navegáveis interiores.
- Navegação interior: realizada em hidrovias interiores, em percurso nacional ou internacional.
- Navegação de longo curso: realizada entre portos brasileiros e estrangeiros.

2.2.3.1 TRANSPORTE MARÍTIMO

MENDONÇA E KEEDI (1997) definiram que o transporte marítimo é aquele realizado por navios a motor, de grande porte, nos mares e oceanos. O meio de transporte mais utilizado no momento para movimentação no comércio internacional é o marítimo. Os navios cargueiros apresentam-se em várias formas: os convencionais, de carga geral, os de carga frigorífica, graneleiros, navios-tanque, roll-on roll-off, porta-containers, entre outros,

para adaptação dos mais variados tipos de carga.

Internacionalmente, o transporte marítimo é controlado pela *International Maritime Organization* (IMO), Organização Marítima Internacional, entidade ligada à Organização das Nações Unidas (ONU), cuja função é promover a segurança no mar, a eficiência da navegação e tomar medidas preventivas para evitar a poluição que pode ser causada pelos navios.

Não há como discorrer sobre esse modal sem ressaltar a figura do armador, que é pessoa jurídica estabelecida e registrada com a finalidade de realizar transporte marítimo local ou internacional através de operação de navios em determinadas rotas e que se oferece para transportar cargas de todos os tipos de um porto a outro.

O armador não precisa necessariamente ser o proprietário de todos os navios que está operando, pois pode utilizar navios afretados de terceiros para compor sua frota. Responsável pela carga que está transportando, responde juridicamente por todos os problemas sobre ela a partir do momento que a recebe para embarque, devendo fornecer ao embarcador um Conhecimento de Embarque - Bill of Lading - B/L (Anexo A), que é o contrato de transporte, normalmente emitido e assinado pelo agente marítimo em nome e por conta do armador.

Agência Marítima é a empresa que representa o armador em determinado país, estado ou porto, fazendo ligação entre o armador e o comerciante/exportador/importador. Não é comum o contato do comerciante com o armador diretamente, visto que essa função pertence ao agente marítimo.

Em resumo, o transporte marítimo de linha regular pode ser contratado através de agentes marítimos (representantes das companhias de navegação, como dito anteriormente); de *freight forwarders* (transitários ou agentes internacionais de carga que não possuem fidelidade ao armador); ou de *Non Vessel Owner Common Carriers* (NVOCC) (companhias de navegação que não possuem navios próprios).

O serviço contratado se encarrega da emissão do conhecimento de transporte, que é o documento que comprova o embarque e possui os dados relativos à carga transportada. Cada companhia de navegação pode ter seu modelo de conhecimento de embarque, a ser preenchido com os dados necessários, tais como: nome do exportador, nome e endereço da companhia de navegação, nome do importador, porto de embarque, porto de destino, nome de quem vai ser notificado quando da chegada da mercadoria, total de volumes, nome da mercadoria, peso

bruto e volume, forma de pagamento do frete (*prepaid* ou *collect*), valor do frete (em algarismos e por extenso), nome do agente da companhia transportadora no porto de embarque, com o carimbo e a assinatura do responsável e o carimbo do local de estiva da mercadoria (*shipped on board*).

A estiva das mercadorias geralmente acontece na empresa exportadora ou no terminal de carga marítimo, que é o local especializado no armazenamento, unitização e movimentação de cargas para embarques e desembarques e localizado fora das áreas portuárias, sendo também utilizado pelos armadores para armazenamento de containeres vazios a serem entregues aos embarcadores.

Dependendo do local de estiva, no modal marítimo são reconhecidas as seguintes contratações para transporte: *House to House* (A mercadoria é colocada no container nas instalações do exportador e desunitizada no pátio do consignatário), *Pier to Pier* (Apenas entre dois terminais marítimos) e *Pier to House* ou *House to Pier*.

Pontos importantes a serem vistos são os custos do transporte marítimo, que são influenciados por características, peso e volume cúbico da carga, fragilidade, embalagem, valor, distância entre os portos de embarque e desembarque e localização dos portos. Em geral, a tarifa de frete é denominada frete básico (valor cobrado segundo o peso ou cubagem da mercadoria, prevalecendo sempre o que gerar maior receita ao armador). Entretanto, outras cobranças costumam ser aplicadas. Dispõe-se abaixo as mais ocorridas:

- *Ad Valorem* (percentual que incide sobre o valor no local de embarque da mercadoria): aplicado normalmente quando esse valor corresponder a mais de USD 1,000.00 por tonelada.
- Sobretaxa de combustível (*Bunker Surcharge*): percentual aplicado sobre o frete básico destinado a cobrir custos com combustível.
- Taxa para volumes pesados (*Heavy Lift Charge*): atribuída às cargas cujos volumes individuais são excessivamente pesados e necessitam de condições especiais na estivagem.
- Taxa para volumes de grandes dimensões (*Extra Length Charge*): aplicada geralmente às mercadorias com comprimento superior a 12 metros.
- Adicional de Porto: taxa cobrada quando a mercadoria tem como origem ou destino algum porto secundário ou fora da rota.

- Sobretaxa de congestionamento (*Port Congestion Surcharge*): incide sobre o frete básico para portos onde existe demora a atracação dos navios.

Deve-se citar a ocorrência do Adicional de Frete para Renovação da Marinha Mercante (AFRMM), que consiste na aplicação de um percentual de 25% sobre o frete para a navegação de longo curso, cobrado do consignatário da carga pela empresa de navegação, que o recolhe posteriormente. Passa a ser devido no porto brasileiro de descarga e na data da operação (início efetivo da operação de descarregamento), ou seja, não é cobrado na exportação, somente na importação. O valor do frete como base de cálculo do AFRMM equivale à remuneração do transporte mercante porto a porto, incluídas as despesas portuárias e outras despesas constantes no conhecimento de embarque.

Denomina-se prescrição de carga a obrigatoriedade de utilização de veículos de bandeira brasileira nas operações de comércio exterior, aplicável no transporte marítimo nas seguintes situações:

- No comércio com os países com os quais existem acordos ou convênios internacionais de divisão de cargas bilaterais (obrigatória a utilização da bandeira de um dos dois signatários), ressaltando-se a existência, quase sempre, de produtos excetuados, normalmente minérios a granel e petróleo;
- Cargas importadas por órgãos ou empresas governamentais da administração direta ou indireta ou por empresas que venham a ser beneficiadas com reduções ou isenções de tributos (ex-tarifário), com algumas exceções.

A liberação da carga (concessão de *waiver*) pode ser solicitada por ocasião do embarque, caso a obrigatoriedade prevista não possa ser cumprida por falta de navio ou espaço em navio de bandeira específica, ao delegado do Departamento de Marinha Mercante (DMM).

2.2.3.2 TRANSPORTE FLUVIAL

Segundo DEMARIA (2004), o transporte realizado em rios tem utilização muito pequena no Brasil, se considerado o potencial de suas bacias hidrográficas. Este é um assunto que tem sido bastante visado, pois há possibilidades de significativas reduções de custo de

transporte em relação aos modais rodoviário e ferroviário, graças ao interesse que vem despertando no transporte de produtos agrícolas, especialmente no que diz respeito à região Centro-Oeste.

A maior parte das mercadorias transportadas por esse meio são os *commodities*; entretanto na Bacia Amazônica também ocorre o transporte de mercadorias manufaturadas juntamente com a madeira; aí o transporte se realiza de forma internacional, ligando diversos portos brasileiros no Amazonas, Pará, Amapá, Roraima a portos no Peru e Colômbia.

Deverá ter grande importância, num futuro próximo, o transporte de contêineres via fluvial, principalmente na rota do Mercosul. Nos EUA, por exemplo, o transporte em barças é realizado por via fluvial para entrega no interior do país, principalmente no sul pelo rio Mississippi, sendo parte da produção de soja transportada por esse meio.

Segundo MENDONÇA E KEEDI (1997), nesse tipo de transporte os equipamentos utilizados são as balsas, chatas e pequenos barcos, bem como navios de médio porte.

Nesse modal, a principal sistemática do cálculo de frete é baseada na tonelada/quilômetro, ou seja, a tonelage transportada em relação à distância da viagem, podendo também o frete ser cobrado por unidade, no caso de transporte de containeres.

Os custos dos fretes são inferiores aos demais modais, comparando o mesmo tipo de mercadoria e o percurso, o que faz do transporte fluvial uma opção interessante, principalmente no mercado interno.

2.2.3.3 TRANSPORTE LACUSTRE

De acordo com MENDONÇA E KEEDI (1997, p.91), este modal tem as mesmas características do fluvial, porém consiste no transporte em lagos, podendo ser considerado incipiente, não tendo importância relativa no transporte de cargas no comércio internacional. As suas rotas são determinadas por vias adequadas, providas pela própria natureza e estabelecidas pelo homem.

Em princípio todos os lagos são navegáveis, porém a navegação comercial vai depender de suas características – tamanho, profundidade, localização e viabilidade econômica. Normalmente os lagos são utilizados para o transporte de mercadorias nas regiões circunvizinhas.

Alguns lagos navegáveis, citados por MENDONÇA E KEEDI (1997, p.91), em que

há transporte de carga são:

- Os Grandes Lagos, na fronteira entre os Estados Unidos e o Canadá;
- No Brasil, a Lagoa Mirim, que liga o Brasil ao Uruguai, e a Lagoa dos Patos, ligando Rio Grande a Porto Alegre;
- Na Bolívia, o Lago Titicaca, que liga a Bolívia ao Peru.

As embarcações são as mesmas utilizadas na navegação fluvial, e o sistema de frete também, ou seja, tem como base a tonelada/quilômetro da viagem, podendo ser por unidade, no caso de transporte de containeres.

2.2.4 MODAL AEROVIÁRIO

DEMARIA (2004) diz que o transporte aéreo é uma atividade que envolve com facilidade vários países pela velocidade do meio utilizado. O princípio seguido é o mesmo, tanto para cargas nacionais (transporte doméstico ou cabotagem), quanto para as cargas internacionais (operações de comércio exterior), baseado em normas da *International Air Transport Association* (IATA) e em acordos e convenções internacionais. O órgão que acompanha os serviços prestados pelos transportadores aéreos e seus respectivos agentes é o Departamento de Aviação Civil (DAC), do Comando da Aeronáutica. Denominado *Air Waybill* - AWB (Anexo 'C'), o conhecimento de embarque aéreo é composto de três vias originais, não negociáveis. O conhecimento de embarque deve conter: nome e endereço do embarcador; nome e endereço da empresa aérea e do agente de carga; nome e endereço do consignatário; pontos de partida e destino, com indicações de escala se houver; código e data do voo; informações sobre embalagem; total de volumes; identificação da mercadoria; peso bruto e dimensões; forma de pagamento do frete (*prepaid* ou *collect*); valor do frete. O frete aéreo é obtido pela multiplicação do peso transportado pela tarifa. Porém, para determinação do peso de uma mercadoria embalada, deve ser levado em conta o “fator estiva”, o qual define se a cobrança do frete ocorre sobre o peso ou sobre o volume, prevalecendo o maior número apurado. As tarifas que se baseiam em rotas, tráfegos e custos inerentes são estabelecidas no âmbito da IATA pelas empresas aéreas, para serem cobradas uniformemente, conforme as classificações a seguir:

- Tarifa Geral de Carga (*General Cargo Rate*):

Normal: aplicada aos transportes de até 45 Kg.

Quantidade: para pesos superiores a 45 Kg.

- Tarifa Classificada (*Class Rate*): percentual adicional ou deduzido da tarifa geral, conforme o caso, quando do transporte de mercadorias específicas (produtos perigosos, restos mortais e urnas, animais vivos, jornais e periódicos, entre outros).
- Tarifas Específicas de Carga (*Specific Commodity Rates*): tarifas reduzidas, aplicáveis a determinadas mercadorias entre dois pontos determinados (Transporte regular), para um peso mínimo por embarque (*Break point*) definido por mercadoria.
- Tarifas ULD (*Unit Load Device*): transporte de unidade domicílio a domicílio, aplicável a cargas unitizadas, em que o carregamento e o descarregamento das unidades ficam por conta de remetente e destinatário (prevista a cobrança de *demurrage* ou taxa de sobre estadia por dia ou fração até que a preparação ou liberação da unidade de carga esteja concluída).

2.2.5 MODAL DUTOVIÁRIO

Nas palavras de DEMARIA (2004), o transporte dutoviário é aquele que utiliza dutos para o transporte das mercadorias. Pelas próprias condições físicas do meio, os principais produtos que se utilizam desse modal são os derivados de petróleo, refinados de grãos (óleo), gases e álcool, em sua maioria.

As principais atribuições da Agência Nacional de Transportes Terrestres (ANTT) com relação ao modal dutoviário são:

- Promover levantamentos de dados relativos a dutovias brasileiras;
- Organizar o cadastro do sistema de dutovias do Brasil e das empresas proprietárias de equipamentos e instalações de transporte dutoviário.

A competência da Superintendência de Logística e Transporte Multimodal - SULOLOG, em seu artigo 46 - inciso XIII é elaborar o cadastro do sistema de dutovias e das empresas proprietárias de equipamentos e instalações de transporte dutoviário. Esta competência foi estabelecida pelo Regimento Interno e pela Estrutura Organizacional da Agência, aprovado pela Resolução n.º 104, de 17 de outubro de 2002.

O transporte Dutoviário pode ser dividido em:

- Oleodutos, cujos produtos transportados são, em sua grande maioria: petróleo, óleo combustível, gasolina, diesel, álcool, GLP, querosene e nafta, e outros;
- Minerodutos, cujos produtos transportados são: Sal-gema, Minério de ferro e Concentrado Fosfático;
- Gasodutos, cujo produto transportado é o gás natural. O Gasoduto Brasil-Bolívia (3150 km de extensão) é um dos maiores do mundo.

Comparado com os modais rodoviário e ferroviário, o modal dutoviário se revela como uma das formas mais econômicas de transporte para grandes volumes (ANTT – www.antt.gov.br/carga/dutoviario/apresentação.asp, 2006).

A Agência Nacional de Transportes Terrestres apresenta na TAB.2.6 os tipos de cargas transportadas pelo modal dutoviário.

A movimentação por dutos é bastante lenta. Em compensação, o transporte opera 24 horas por dia e sete dias por semana. Com relação ao tempo de trânsito, o transporte dutoviário é o mais confiável de todos, pois existem poucas interrupções para causar variabilidade no tempo de entrega.

TAB.2.6: Cargas Transportadas pelo Modal Dutoviário

OLEODUTOS	MINERODUTOS	GASODUTOS
Alcool	Sal-gema	Gás Natural
Butadieno	Minério de Ferro	Gás Combustível de Xisto
Diluyente	Concentrado Fosfático	Gás Residual
Gasolina	Caulim	CO ₂
GLP – Gás Liquefeito de Petróleo		
GOCRAC - Gasóleo para craqueamento		
GOPECOQ – Gasóleo Pesado de Coque		
LCI – Óleo leve		
MTBE – Metil terciil butil eter		
Nafta		
NAFCOQTR – Nafta de coque tratada		
ODESASF – Óleo desasfaltado		
Óleo Combustível		
Óleo Diesel		
OLEVETUR – Óleo leve para turbina elétrica		
Petróleo		
Propeno		
Querosene		
Resíduo		
RESATCR – Resíduo ATM p/ craqueamento		

(Fonte: ANTT – www.antt.gov.br/destaques/ANTTemNumeros20051226.pdf, 2006)

Danos e perdas de produtos em dutos são baixos, pois líquidos e gases não estão

sujeitos a danos no mesmo grau que produtos manufaturados, e a quantidade de perigos que podem ocorrer na operação dutoviária é limitada. Há responsabilidade legal por danos ou perdas, uma vez que dutos têm o mesmo status que transportadores regulares, mesmo que muitos deles sejam de operação própria.

No Brasil, a Lei N. 7.029, de 13/09/1982, é a única que dispõe sobre esse modal e trata do transporte dutoviário de álcool.

2.2.6 TERMINAIS

O conhecimento dos conceitos que abrangem os terminais de transporte é essencial para o bom desempenho da logística de transportes.

DEMARIA (2004) diz que o termo Terminal corresponde etimologicamente a uma realidade funcional, desde que cada tipo de instalação permita especificamente o término de uma seqüência ou o início de outra.

Para HAY (1977) os terminais podem ser considerados a soma total de facilidades e sua localização onde o transporte pode ser originado, terminado e/ou intercambiado antes, durante ou depois do deslocamento, incluindo os serviços de facilidades para veículos e equipamentos utilizados no transporte.

Em outras palavras, o terminal é a origem ou destino do transporte, mas em alguns casos, o terminal é o ponto onde a carga necessita de serviços de consolidação e desconsolidação para a distribuição da carga. Como exemplo, pode-se citar uma exportação marítima em container LCL (*Less than Container Load*): a carga é enviada inicialmente a um terminal designado pelo transportador marítimo, onde será consolidada e posteriormente enviada ao porto para o despacho ao exterior.

THIRIET-LONGS (1982) afirma que os terminais têm uma importância capital na adequação dos transportes. De sua localização, de seu dimensionamento e de seus componentes físicos e organizacionais dependem a eficiência e a rentabilidade dos sistemas em nível nacional. Ao mesmo tempo, será garantida ou não a equiparação operacional e a competitividade da nação perante o mercado internacional. De acordo com as finalidades fundamentais, os terminais justificam uma classificação em três categorias de instalações:

- Destinada ao transbordo de interface;
- Para consolidação/desconsolidação (estocagem transitória);

- Para racionalizar o preparo-prolongamento dos fluxos intermodais (otimização de modais).

Essas categorias correspondem a determinadas funções operacionais que irão influir diretamente sobre os acessos modais.

No transbordo de interface, a função básica se resume na transferência da unidade de carga de uma modalidade para outra.

No terminal de consolidação/desconsolidação, as funções essenciais se norteiam em operações transitórias de estocagem, de armazenagem ou ensilagem (carga granel sólida). Trata-se de cargas indo ou vindo da rede de transportes e então consolidadas (ovadas em unidades, como containeres, trailers, etc.) ou desconsolidadas (distribuição). Esse tipo de terminal não é considerado como um depósito, pois nele estão apenas mercadorias em trânsito.

O objetivo dos terminais de preparo ou prolongamento dos fluxos dos intermodais é o atendimento das seguintes alternativas:

- Descongestionamento dos tráfegos urbanos;
- Agrupamento de lotes de cargas locais por transportadoras, usuários ou indústrias locais, sendo esse agrupamento complementado por funções específicas tais como entrepostos.

O objetivo principal dos sistemas intermodal e multimodal de transporte é a dinamização das operações da origem até o destino da carga. No processo de integração dos modais, procura-se conferir uma determinada dinâmica operacional para cada tipo de terminal:

- Dinâmica do escoamento intermodal para o terminal de transbordo ou interface, com o objetivo de transferir rapidamente a carga já unitizada sem que seja significativamente aumentado o tempo de escoamento das unidades de carga nas interfaces rodovia, ferrovia e hidrovia;
- Dinâmica do acondicionamento que trata apenas da estocagem transitória das mercadorias pela qual a carga será unitizada corretamente com o intuito de permitir um escoamento por rotas que utilizem diferentes modais, através de equipamentos

especializados e de alta produtividade. No processo inverso, esse terminal permitirá a desconsolidação das cargas recebidas para então despachá-las ao proprietário;

- Dinâmica do prosseguimento para o terminal de preparo ou prolongamento de fluxos intermodais, no qual se procura otimizar o tráfego local, evitando os desperdícios causados pelos congestionamentos. Para alcançar essa otimização é preciso racionalizar a demanda e a oferta de transporte. Operacionalmente, as instalações serão subdivididas entre várias empresas privadas, dentro de galpões comuns ou separados, interessando principalmente aos pólos de consumo-produção.

Segundo a Associação Brasileira de Transportadores Intermodais (ABTI), para o desenvolvimento das estruturas de terminais de transporte intermodais e/ou multimodais, os principais critérios técnicos operacionais a serem determinados na programação nacional de implantação dos terminais são os seguintes:

- Localização dos terminais;
- Tipos de terminais a serem adotados em função da localização;
- Compatibilização da capacidade operacional entre os terminais (dimensionamento, equipamentos, processo, organização, documentação padrão, etc.);
- Agilização da regulamentação nas redes modais, criação de serviços alfandegários e de fiscalização adaptados às necessidades dinâmicas dos terminais.

2.3 SISTEMA DE INFORMAÇÃO PARA O TRANSPORTE DE CARGA

Segundo FERREIRA FILHO (1998), para que o Setor de Transporte possa planejar e operar seus diversificados sistemas é necessário coletar e processar um grande número de dados referentes a vias, veículos, capacidades, disponibilidades, equipamentos, pessoal, tipos de carga, manutenções diversas, e outros. O Sistema de Informação é um dos mecanismos existentes que auxilia no processamento desses dados. Entretanto, para que as empresas envolvidas com transporte possam planejar mais adequadamente os seus negócios, diminuindo o tempo de tomada de decisões gerenciais, é necessário que o Sistema de Informação possa processar os dados de uma forma rápida, clara e precisa.

O Sistema de Informação para os Transportes – SIT visa satisfazer necessidades legais e transacionais de Transportes, fornecendo informações gerenciais que apoiem nas atividades

de planejamento, acompanhamento e tomada de decisão. Para se atingir tal objetivo, utiliza-se basicamente as potencialidades de Banco de Dados, que atuam nos processos como Sistemas de Recuperação de Informações armazenadas em repositórios específicos, associadas a aplicativos voltados para os Transportes, como análise de redes viárias e estudo de tarifas.

De acordo com DIOGO (1994), para se desenvolver um Sistema de Informação que atenda ao setor de transporte deve-se seguir os seguintes passos:

- Visualizar o Sistema sob o enfoque das grandes funções que realiza:
Transporte de carga, transporte de passageiros de longa distância e transporte urbano de passageiros;
- Enfocar o Sistema segundo os modos que propiciam aquelas funções:
Ferroviário, rodoviário, aquaviário, aeroviário e os terminais, que permitem a interface entre os modais;
- Observar as informações necessárias ao ciclo de vida do Sistema de Transportes em cada modo, constituído pelas atividades de planejamento, pesquisa, projeto, construção, operação e manutenção;
- Usar uma metodologia apropriada para o desenvolvimento do Sistema de Informação que pode recair sobre duas características básicas: técnicas estruturadas e não estruturadas. A metodologia estruturada incorpora o aspecto científico e rigoroso de análise e especificação dos processos, dados e estrutura organizacional, enquanto que a metodologia não estruturada (conhecida como Prototipação ou Prototipagem) prefere a ação criativa e a intuição do ser humano. A Prototipação é um conjunto de técnicas e ferramentas de software para o desenvolvimento de modelos de sistemas. Seu principal objetivo é antecipar ao usuário final uma versão do sistema para que ele possa avaliar sua funcionalidade, identificando os erros e as omissões.

TANAKA (1994) diz que as principais vantagens da Prototipação, quanto à filosofia de *Design*, são: Redução do fator tempo, análise de características do sistema que as especificações escritas não mostram, entre outras. A principal desvantagem é que a prototipação exige uma alta capacitação gerencial da equipe de projeto.

Quanto ao desenvolvimento de um Sistema de Informação de Transporte, é aconselhável que seja realizado em duas fases:

- Na primeira, direcionada a gerenciar os dados, possibilitando uma visão da base de dados;
- Na segunda, voltado a atender ao processamento de informações para as atividades de planejamento e operação.

Com relação ao levantamento de dados, DIOGO (1994) mostra que para se iniciar o processo de identificação de entidades de dados de um Sistema de Transporte deve-se responder as seguintes perguntas:

- Quais as Entidades de Dados que constituem a base para a construção do sistema?
- Quais as respostas que o sistema deve dar em termos de dados?
- Que dados estão vinculados a essas respostas e que devem servir de entrada para o sistema?

Tendo a relação das entidades de dados, volta-se para a construção do Modelo Conceitual de Dados (ER) de Peter Chen.

SILVA (2001) apresenta um modelo que apóia na decisão sobre a racionalização da utilização dos meios de transportes disponíveis numa situação de mobilização. Para atingir este objetivo, SILVA (2001) revisa os conceitos sobre a mobilização, logística e transportes, além de levantar as etapas que integram o processo de planejamento dos transportes militares e analisar os principais fatores que influem na tomada de decisão, na hipótese de existir mais de uma alternativa de transporte em situação de mobilização.

Com base nesses conceitos, foi desenvolvido um modelo chamado PLANMOB, que é utilizado no apoio ao planejamento da mobilização militar dos transportes.

A estrutura do Modelo Conceitual pela qual o PLANMOB se baseia está mostrado na FIG.2.2:

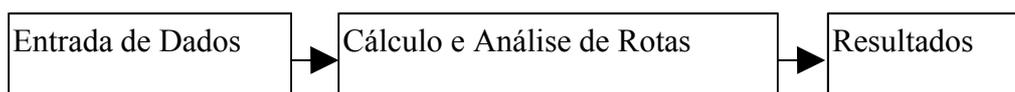


FIG.2.2: Estrutura de Modelo Conceitual para o PLANMOB

1. Entrada de Dados – se refere a três grupos de dados:

- Cargas: descreve natureza, peso, área, volume; pontos de origem e destino; e prazos em que deve chegar ao destino. Os dados são levantados pelas Diretorias;
- Recursos de Transporte: descreve disponibilidade, capacidade, velocidade dos meios de transporte; infra-estrutura (vias, terminais e operação), restrições (gabarito, peso); descrições para os diferentes modais de transporte. Dados obtidos do banco de dados do SIMMTRA;
- Cenários: descreve as áreas onde há possibilidades de ação do inimigo e de interferência do clima. Estas informações fazem parte do planejamento global da operação militar.

2. Cálculo e Análise de Rotas – se referem ao cálculo do número de veículos a mobilizar e ao procedimento para análise de rotas.

O cálculo da frota de veículos de transportes é a mais importante função deste modelo.

Este cálculo é baseado no modo de transporte, nas características da carga, e na disponibilidade de caminhões, vagões ou embarcações.

Para se obter a formulação matemática deste problema, foi proposta a seguinte seqüência:

- Seleção dos tipos de veículos por grupo;
- Cálculo da capacidade útil dos veículos por subgrupo;
- Formulação do problema;
- Solução do problema.

Em seguida, SILVA (2001) faz uma avaliação da rede viária, calculando os fatores de impedância associada aos segmentos da rede viária. Neste caso os principais fatores de impedância a serem analisados estavam ligados ao tempo de viagem e ao risco militar. A impedância, em análise de redes, pode ser definida como distância, tempo, custo ou outro valor específico.

Finalmente, SILVA (2001) faz uma análise de rotas viáveis e adequadas para o deslocamento de cargas, numa situação de mobilização militar dos transportes. Para este efeito usa a teoria do Problema do Caminho Mínimo.

GUIMARÃES (1999) desenvolve um procedimento para a geração de uma base multimodal de transporte a partir de bases unimodais, usado como ferramenta para o planejamento de transporte em ambiente SIG.

GUIMARÃES (1999) estuda conceitos sobre o processamento do Transporte Multimodal e Sistema de Informações Geográfica com aplicabilidade nos transporte de carga.

Com base nesses conceitos, GUIMARÃES (1999) faz uma Compatibilização das Estruturas⁽¹⁾ das Bases existentes e a União das Estruturas⁽²⁾ com outros procedimentos que levam à aplicação de um roteamento usando a base multimodal. A partir desse ponto, é desenvolvido um Modelo Conceitual (ER) para os transportes.

⁽¹⁾ A Compatibilização das Estruturas é feita usando-se o aplicativo Transcad. Durante o processo de Compatibilização, apenas alguns dos atributos são mantidos na base multimodal, para se evitar redundâncias e inconsistências.

⁽²⁾A União das Estruturas é realizada por meio de uma rotina do Gisdk (Geographic Information System Developer's Kit). Este procedimento é proposto com a finalidade de tornar possível a união de arquivos geográficos modais computacionais distintos para a análise comparativa de rotas de transporte. Embora o procedimento tenha sido proposto para a união de bases relacionadas com qualquer modal de transporte, só são utilizadas as bases nos modos: ferroviário, aquaviário e rodoviário. Para demonstrar a aplicabilidade e viabilidade do sistema proposto, analisou-se opção alternativa de transporte no eixo Rio-São Paulo. Para isso, usou-se uma aplicação envolvendo a união das bases representativas das redes rodo, ferro e aquaviária brasileiras, com o objetivo de verificar a possibilidade de roteamento multimodal.

3 MODELAGEM DE SISTEMA DE INFORMAÇÃO EM AMBIENTE LIVRE

3.1 INTRODUÇÃO

Segundo FIGUEIREDO NETO (2005), o conceito de informação envolve um processo de redução de incerteza. Ele definiu a informação como sendo o conhecimento disponível que permite orientar a ação, reduzindo a margem de incerteza que cerca as decisões cotidianas. Os sistemas de informação são sistemas específicos de busca, coleta, armazenamento, classificação e tratamento das informações importantes e relevantes criados pelas organizações para facilitar seu funcionamento e melhorar o seu processo de decisão.

De acordo com MURTA (1999), o *Global Logistics Research Team* da *Michigan State University* (1995) observou que a tecnologia de informação é uma das sete capacidades que se combinam para a integração dos processos logísticos e performance de classe mundial. Ao fazer esta observação, esse grupo de pesquisas definiu a capacidade de informação tecnológica como “a aplicação do *hardware*, *software* e redes para incrementar o fluxo de informações e facilitar as decisões”.

FIGUEIREDO NETO (2005) mostra, na FIG.3.1, que tanto as fontes de informação internas como externas à organização influenciam na tomada de decisão, sendo que estas decisões causam efeitos no ambiente externo e na organização. Estes efeitos pressupõem a necessidade do atendimento de uma demanda e devem ser comparados com as metas previamente estabelecidas. Caso os efeitos obtidos difiram dos padrões preestabelecidos, são geradas novas informações que realimentam a tomada de decisão. Normalmente, só se consegue atingir os resultados desejados após sucessivas aplicações do processo descrito. Espera-se que, quanto mais precisas forem as informações, menores serão os números de interações necessárias à obtenção dos resultados esperados. Por outro lado, caso se disponha de informações com baixo nível de confiabilidade, mesmo que em grande quantidade, deve-se esperar dificuldade na obtenção dos objetivos.

MELLENDEZ FILHO (1990) define o Sistema de Informação como um conjunto sistemático e formal de componentes, que realiza o processamento de dados visando satisfazer as necessidades legais e transacionais de uma organização e fornecer informações à gerência para apoio às atividades de planejamento, acompanhamento e tomada de decisão.

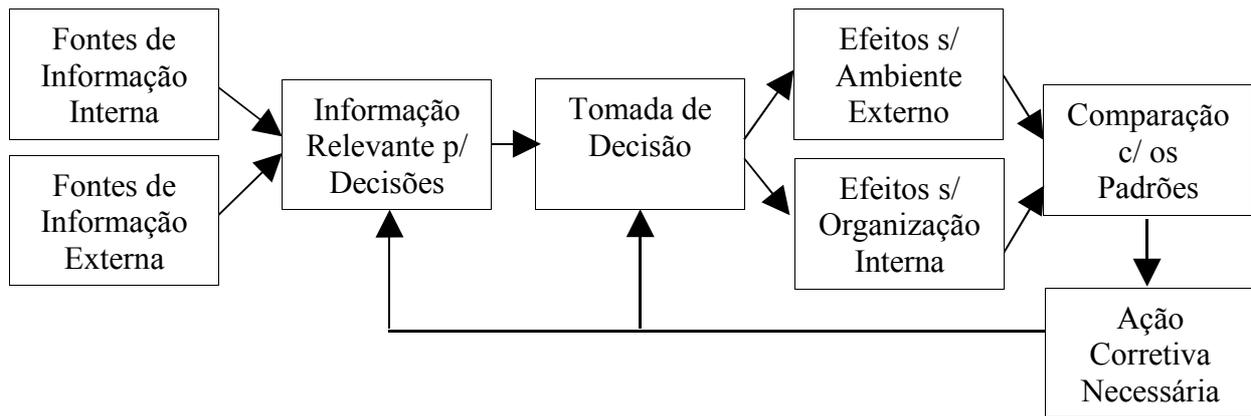


FIG.3.1: Componentes Básicos de um Sistema de Informação Gerencial (Fonte: Herbert G. Hicks E C.Ray Gullet - The Management of Organization, New York, McGraw-Hill Book Co., 1976)

Para GUIMARÃES (1999), a função de um Sistema de Informação é a de aperfeiçoar a habilidade na tomada de decisões. Ela também cita CALKINS e TOMLINSON (1997), que definem um Sistema de Informação como a cadeia de operações que leva do planejamento da observação e coleta de dados, passando pela estocagem e análise destes dados e chegando ao uso da informação derivada, a um processo de tomada de decisão.

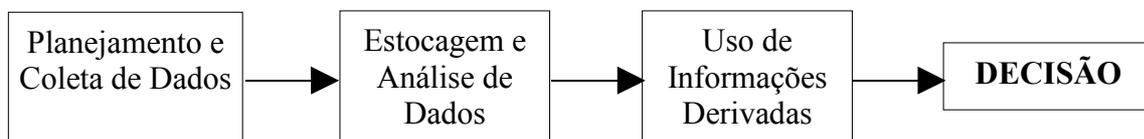


FIG.3.2: Sistema de Informação – Esquema de Operações

Para FERREIRA FILHO (1998), o desenvolvimento tecnológico tem evoluído de uma forma tão acelerada, que impõe a adoção de mecanismos para processar os dados de uma forma rápida, clara e precisa, para permitir que sejam tomados decisões empresariais rápidas, e que num futuro bem próximo, tal facilidade será um instrumento básico de apoio à chefia.

Convém destacar que embora este trabalho procure mostrar a importância de um modelo de Sistema de Informação de Transportes (SIT) em ambiente livre, praticamente, não há SIT desenvolvido em Linux no Brasil.

3.2 AMBIENTE LIVRE

Segundo UCHÔA (2004), existem dois movimentos que muitas vezes se confundem: Software Livre (*Free Software*) e Software de Código Aberto (*Open Source Software*). Abaixo, é apresentada uma descrição da diferença entre esses dois elementos e as licenças pelas quais eles são regidos, assim como um esclarecimento da diferença entre Softwares Gratuito, Semilivre, Proprietário e Comercial.

“SOFTWARE LIVRE (SL)

SL é um programa de computador como qualquer outro programa proprietário. Ele tem a mesma finalidade, ou seja, é direcionado para atender uma determinada demanda como, por exemplo: planilhas de cálculos, editores de textos, editores de imagens. O que realmente o faz *tanto diferente*? A resposta está no tipo de licença associada ao software livre. Esta licença deve garantir ao SL, segundo a Fundação Software Livre (*Free Software Foundation*), quatro liberdades:

- A liberdade de executar o programa, para qualquer propósito;
- A liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades.
- A liberdade de redistribuir cópias, permitindo a ajuda ao próximo;
- A liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie.

Acesso ao código fonte é um pré-requisito para esta liberdade.

Para precisar o conceito de SL, deve ser observado que o mesmo está associado a um grande movimento social, onde a idéia de liberdade do uso do software é pregada como solução do problema gerado pela limitação do conhecimento tecnológico imposta pelos sistemas proprietários. Você pode procurar mais informações sobre o movimento do SL no link www.gnu.org/philosophy/freesoftwareforfreedom.html.

SOFTWARE DE CÓDIGO ABERTO

Nesta categoria de softwares o usuário tem acesso ao código fonte, podendo alterá-lo para atender as suas necessidades. Muitas vezes, as idéias de SL e código aberto se confundem no nosso cotidiano. A melhor forma de compreender a diferença entre eles é observar que, normalmente, o software de código aberto deixa de atender alguma(s) das quatro liberdades do SL (vide item anterior). Outra forma de analisar esta diferença é pensar que “o código aberto faz alusão a uma metodologia de desenvolvimento, enquanto o software livre está relacionado a um movimento social”.

SOFTWARE GRATUITO

Estes sistemas são disponibilizados de forma gratuita, porém,

normalmente, não podem ser modificados e não se tem acesso ao código fonte.

É possível também que a licença impeça a redistribuição do mesmo. Também são conhecidos como *Freeware*. Cuidado para não confundir com os *Shareware*, pois estes últimos apesar de também serem gratuitos, possuem alguma limitação funcional em relação ao software original.

SOFTWARE SEMILIVRE

É o software que não é livre, mas que permite: a utilização, a cópia, a modificação e a distribuição (incluindo a distribuição de versões modificadas) para fins não lucrativos. **PGP**-Pretty Good Privacy (Ótima Privacidade) é um exemplo de programa semilivre.

Os sistemas desta categoria não podem ser incluídos em sistemas operacionais livres, ou seja, não podem acompanhar uma distribuição GNU/Linux.

SOFTWARE PROPRIETÁRIO

Estes sistemas normalmente são protegidos por algum tipo de patente. Seu uso, redistribuição ou modificação é proibido, ou requer que você peça permissão ou é restrito de tal forma que você não possa efetivamente fazê-lo livremente. Um software proprietário pode ser feito com finalidades comerciais ou não. Um exemplo de um software proprietário que não possui finalidade comercial é o **SPRING**, do Instituto Nacional de Pesquisas Espaciais (INPE).

SOFTWARE COMERCIAL

É o software desenvolvido por uma empresa visando obter alguma forma de lucro. Apesar de softwares *comerciais* e *proprietários* estarem muitas vezes associados, eles pertencem a categorias diferentes. Existem softwares livres que são comerciais, assim como existe software de código aberto que também é comercial. Como exemplo deste último, podemos citar o **JUMP**, foi desenvolvido pela empresa canadense Vivid Solutions.

Para se aprofundar no assunto, uma boa orientação é o site: www.gnu.org/philosophy/categories.html

LICENÇAS DE CÓDIGO ABERTO

As licenças **BSD** e **Apache** são duas das mais antigas licenças de código aberto. Elas ilustram bem alguns princípios básicos das licenças de código aberto. Estas licenças, juntamente com a **MIT** (também conhecida por **X**), são licenças de código aberto clássicas para o licenciamento de softwares e são utilizadas em muitos projetos de código aberto. Por isso, a abordagem será concentrada nelas. Alguns exemplos bem conhecidos de sistemas baseados nestas licenças são: o servidor **Apache HTTP** e os sistemas operacionais **BSDNet** e **FreeBSD**.

Para começar a entender estas licenças, deve-se observar o que ocorre quando elas são aplicadas a um determinado código fonte.

O primeiro fato é que este código poderá ser utilizado para compor um

sistema proprietário, sendo que não é exigido que versões de código aberto deste sistema sejam distribuídas. Isto significa que os sistemas abertos criados sob estas licenças podem se transformar em sistemas fechados, gerando uma perda para a comunidade de código aberto. Porém, justamente por este fato, estas licenças são bastante flexíveis e compatíveis com quase todo tipo de licença de código aberto. No próximo tópico, serão abordadas as licenças livres que apresentam, entre outras características, a garantia do código sempre ser aberto.

Licença MIT-Massachusetts Institute of Technology (ou X): é uma licença de código aberto relativamente simples. Ela dá total liberdade (sem restrições) de uso, cópia, modificação, publicação, distribuição e também permite a venda de cópias do programa. Assim como é comum nas licenças abertas, existe, no texto da licença, a informação de que não há qualquer tipo de garantia pelo uso do software ou por qualquer tipo de dano que o mesmo possa causar, deixando o autor livre de tal responsabilidade. Não se assuste com esse fato, pois esse tipo de isenção de garantia está presente até nas licenças proprietárias, que não assumem qualquer tipo de prejuízo no trabalho do usuário.

Licença BSD-Berkeley Software Distribution: esta licença é um pouco mais restritiva do que a **MIT**. Existem inúmeras formas similares a ela, como, por exemplo, a **UCB-University of California/LBL-Lawrence Berkeley Laboratory**. Até 1999, existia uma cláusula exigindo que, em qualquer tipo de material divulgando as funcionalidades ou o uso do programa, fosse mencionado os direitos autorais.

Imagine que você não poderia sequer escrever um pequeno comentário sobre o programa sem mencionar algo do tipo: “este produto inclui um software desenvolvido pelo Fulano de Tal”. Esta cláusula saiu da licença em 1999, principalmente por estar em desacordo com a idéia da filosofia do código aberto que defende a exploração comercial dos sistemas por todos os interessados. Alguns detalhes que devem ser destacados nesta licença são as condições de uso dos binários, códigos-fontes e modificações. Os direitos autorais e as condições da licença devem ser incluídos na distribuição do código-fonte.

No caso da distribuição de binários, deve-se incluir também os termos da licença em toda a documentação e em outros produtos inclusos nesta distribuição. Outro cuidado a se tomar é que não se pode utilizar o nome da instituição e/ou o nome dos autores para promover algum sistema derivado sem uma autorização por escrito dos mesmos. Essa proibição é a diferença mais substancial entre esta licença e a licença **MIT**.

Licença Apache: esta licença é bastante similar às duas citadas anteriormente. A versão 1.1 da licença **Apache** segue as mesmas premissas da **BSD** em relação a distribuição e modificação, apresentando um texto relativamente sem restrições. A licença **Apache v1.1**, apesar de possuir um texto um pouco mais longo que as anteriores, segue

basicamente as mesmas idéias como: copiar, distribuir, modificar, respeitar a autoria, etc. A licença **Apache v2.0** é uma revisão da versão anterior e possui diferenças substanciais com relação aos direitos de patentes e às licenças de sistemas derivados desta versão (2.0). Assim como em outras licenças abertas, na Apache não existe a obrigação do usuário disponibilizar algum sistema derivado de um software desenvolvido sob esta licença, seja através de uma licença de código aberto ou de software livre.

LICENÇAS DE SOFTWARES LIVRES

Nos tópicos anteriores, foram abordadas as 4 liberdades do SL e agora é a vez das principais licenças que garantem estas liberdades. Para iniciar a abordagem, tem-se que entender a mais importante licença livre: a GNU-General Public License GPL-General Public License (no decorrer do texto, será utilizada a denominação GPL). Além das 4 liberdades do SL, essa licença possui o que podemos chamar de efeito contaminante (nos textos em inglês é possível encontrar o termo “reciprocal”, ou também “viral”, sendo essa última denominação mais ligada aos defensores dos softwares proprietários). Isto significa que um sistema derivado de algum software sob a licença GPL, vai ter que ser licenciado sob a mesma licença. Quando não for possível “herdar” a licença GPL por algum motivo qualquer (royalties, patentes, decisão judicial, etc), o sistema **não** poderá utilizar aquele software GPL. Caso o sistema já tenha sido desenvolvido e venha a ter alguma limitação contrária a licença GPL, o mesmo **não** poderá ser utilizado, comercializado, distribuído, etc. A FIG.3.1 representa a idéia do efeito contaminante da GPL.” (Fonte: Geoprocessamento com Software Livre (v 1.0) – Helton Nogueira Uchoa, 2006).

Tendo estas informações em mãos, pode-se definir o Ambiente Livre como um ambiente operacional de um computador que contém Software Livre com o Código Aberto.

O software livre vem sendo a cada dia mais utilizado na tecnologia da informação em toda a parte do mundo, mas, pelo fato de se desconhecer os produtos que se enquadram neste conceito, muitos usuários desconhecem as vantagens que podem ser obtidas nesta área. Além das vantagens já mencionadas, pode-se ainda acrescentar duas outras que agregam valor de importância ao uso do do software livre:

- Existe grátis na Internet, uma grande quantidade de apostilas, tutoriais e páginas de ajuda;
- Existe uma comunidade ativa de usuários na Internet que se auxiliam sem custar nada.

Dentre os milhares de softwares livres que podem ser copiados e instalados em ambiente livre, o mais importante é o sistema operacional Linux. Sua instalação é gratuita e é absolutamente legal. Além disso, é possível ter apenas software livre no computador.

Na TAB.3.1 são apresentados alguns exemplos de softwares "pagos" (softwares "proprietários") mais comuns e alguns dos seus equivalentes livres.

TAB.3.1: Equivalência de Softwares Proprietários

Tipo de software	Softwares proprietários	Softwares livres equivalentes
Sistema operacional	MS-Windows	Linux
Suíte de escritório	MS-Office	Open Office, Koffice
Edição de imagens	Photoshop	Gimp, Gphoto, KolourPaint, Ksnapshot
Navegador de Internet	MS-Internet Explorer	Mozilla, Firefox, Konqueror
E-mail	Outlook	KMail, Evolution, Thunderbird
Multimídia	Winamp, MS-Media Player, Real Player	XMMS, Noatum, KDE Media Player
Banco de dados	Access, SQL Server	MySQL, PostgreSQL

Existe uma grande variedade de opções de escolha de softwares livres de qualidade igual ou superior aos seus equivalentes proprietários. Empresas como IBM, HP, Intel, Sun Microsystems, Dell, Oracle, Borland e várias outras têm investido bilhões de dólares em pesquisa, desenvolvimento e apoio ao software livre. É daí que vem o incentivo que mantém os softwares gratuitos. No Brasil, a USP e a UNICAMP são dois dos muitos exemplos de universidades ativas neste setor.

Além da disposição dos desenvolvedores em permitir a livre cópia, modificação e distribuição de um software, para ele ser de fato "livre" é necessário que existam garantias de que esta liberdade é real e não será revogada, restringida ou cobrada posteriormente. Para dar esta garantia foi criada a "Free Software Foundation" (Fundação do Software Livre) que mantém licenças especiais para os softwares e documentações livres (Ortensi – www.ortensi.com/info/sl_convite.php, 2006).

A razão de o Linux ser o mais importante dos softwares livres é que ele é o sistema mais utilizado em grandes servidores de rede, especialmente para a Internet. O crescente uso do Linux se deve, principalmente, aos seguintes fatores:

- Ele é mais estável, ou seja, é menos sujeito a travamentos e erros;
- É mais seguro, pois é quase impossível que o Linux pegue vírus e, caso pegue, suas proteções próprias impedem que o sistema seja destruído;
- Quando bem configurado, minimiza os riscos de invasões por crackers, roubo de dados, etc;
- Ele é livre para baixar pela Internet, copiar e até modificar.

Quanto ao suporte técnico, além de haver centenas de manuais, apostilas e tutoriais gratuitos na Internet, existem as "listas de discussão" na Internet, onde há milhares de usuários que já enfrentaram e resolveram os mesmos problemas e dão apoio gratuito na solução de problemas e respondem dúvidas (Ortensi – www.ortensi.com/info/linux_5.php, 2006).

3.2.1 LEGISLAÇÃO DE PROGRAMAS ABERTOS

“PROJETO DE LEI N.º 2.269 DE 1999

Dispõe sobre a utilização de programas abertos pelos entes de direito público e de direito privado sob controle acionário da administração pública.

Artigo 1º- A administração pública, em todos os níveis, os Poderes da República, as empresas estatais e de economia mista, as empresas públicas, e todos os demais organismos públicos ou privados sob controle da sociedade brasileira, ficam obrigadas a utilizarem preferencialmente, em seus sistemas e equipamento de informática, programas abertos, livres de restrição proprietária quanto a sua cessão, alteração e distribuição.

Artigo 2º - Entende-se por programa aberto aquele cuja licença de propriedade industrial ou intelectual não restrinja sob nenhum aspecto a sua cessão, distribuição, utilização ou alteração de suas características originais.

Artigo 3º - O programa aberto deve assegurar ao usuário acesso irrestrito ao seu código fonte, sem qualquer custo, com vista a modificar o programa, integralmente, se necessário, para o seu aperfeiçoamento.

Parágrafo Único. O código fonte deve ser o recurso preferencial utilizado pelo programador para modificar o programa, não sendo permitido ofuscar a sua acessibilidade, nem tampouco introduzir qualquer forma intermediária como saída de um pré-processador ou tradutor.

Artigo 4º - A licença de utilização dos programas abertos deve permitir modificações e trabalhos derivados e sua livre distribuição sob os mesmos termos da licença do programa original.

§ 1º - A licença somente poderá restringir a distribuição do código fonte em forma modificada caso permita a distribuição de programas alterados conjuntamente com o código fonte original, objetivando a alteração do programa durante o processo de compilação.

§ 2º - Deve permitir também explicitamente a distribuição de programa compilado a partir do código fonte modificado, podendo para tanto exigir que os programas derivados tenham diferentes nomes ou números de versão, que os diferenciem do original.

Artigo 5º - Não poderá haver cláusula na licença que implique em qualquer forma de discriminação a pessoas ou grupos.

Artigo 6º - Nenhuma licença poderá ser específica para determinado produto, possibilitando que os programas extraídos da distribuição original tenham a mesma garantia de livre alteração, distribuição ou utilização, que o programa original.

Artigo 7º - As licenças de programas abertos ou restritos, não restringirão outros programas distribuídos conjuntamente.

Artigo 8º - Os certames licitatórios que objetivem transacionar programas de computador com os entes especificados no artigo 1º desta lei, deverão obrigatoriamente ser regidos pelos princípios estabelecidos nesta legislação.

Artigo 9º - Apenas será permitida a utilização pelos entes do artigo 1º, de

programas de computador cujas licenças não estejam em acordo com esta lei, na ausência de programas abertos que não contemplem a contento as soluções objeto da licitação pública.”

A) JUSTIFICAÇÃO

Há mais de quinze anos discute-se, em todo o mundo, a livre manipulação dos programas de computador ou "open/free software".

Em 1984, era impossível usar um computador moderno sem a instalação de um sistema operacional proprietário, fornecido mediante licenças restritivas de amplo espectro. Ninguém tinha permissão para compartilhar programas (software) livremente com outros usuários de computador, e dificilmente alguém poderia mudar os programas para satisfazer as suas necessidades operacionais específicas.

O projeto GNU, que data do início do Movimento do Software livre, foi fundado para mudar isso. Seu primeiro objetivo foi desenvolver um sistema operacional portátil compatível com o UNIX que seria 100% livre para alteração e distribuição, proporcionando aos usuários que contribuíssem com o seu desenvolvimento e alteração de qualquer parte de sua constituição original.

Tecnicamente GNU é como UNIX, mas difere do UNIX pela liberdade que se proporciona aos seus usuários. Para a confecção deste programa aberto, foram necessários muitos anos de trabalho, por centenas de programadores, para desenvolver este sistema operacional. Em 1991, o último componente mais importante de um sistema similar ao UNIX foi desenvolvido: o LINUX.

Hoje a combinação do GNU e do Linux é usada por milhões de pessoas, de forma livre, em todo o mundo. Este programa é apenas um exemplo de como a liberdade na alteração, distribuição e utilização de programas de computador poder transformar ainda mais rapidamente, e de maneira mais democrática, o perfil do desenvolvimento social e tecnológico no mundo (Deputado Walter Pinheiro, 1999).

3.3 SISTEMA DE BANCO DE DADOS

De acordo com ULLMAN E WIDOM (1997), o termo Banco de Dados se refere à coleção de dados que são gerenciados por um Sistema Gerenciador de Banco de Dados.

Segundo TANAKA (1997), Banco de Dados é uma coleção de dados inter-relacionados, logicamente coerentes e com algum significado inerente. É projetado, construído e povoado com dados para um propósito específico. Possui aplicações pré-concebidas e visa atender certo grupo de usuários.

FERREIRA FILHO (1998) mostra esquematicamente na FIG.3.3 a evolução dos Sistemas de Banco de Dados em função do Tempo, descrevendo os Dados nos primórdios da computação, cujos armazenamentos eram feitos dentro dos programas, depois sendo armazenados separadamente dos processos em Arquivos de Dados, logicamente relacionados, e finalmente desvinculados totalmente dos processos, sendo gerenciados por um Sistema de Gerenciamento de Banco de Dados (SGBD).

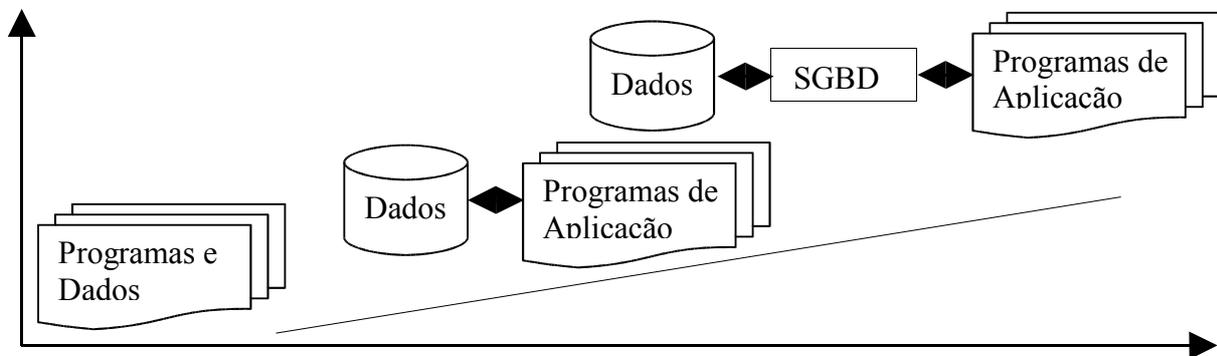


FIG.3.3: Evolução Dos SBD

Segundo JEPSON *et al.* (2000), os sistemas de bancos de dados usam tabelas, que são repositórios de dados compostos de uma grade de linhas e colunas, com o propósito de armazenar informações sobre os objetos do mundo real ou conceitos (entidades).

Cada linha representa uma determinada ocorrência de uma entidade, que pode ser uma pessoa, um local ou qualquer outra coisa, enquanto que a coluna é um atributo ou uma propriedade da entidade.

Para fins de ilustração, abaixo, são apresentas as colunas CÓDIGO, NOME, ENDEREÇO e TELEFONE, que são os atributos da entidade pessoal. Cada linha é uma ocorrência dessa entidade que representa uma pessoa.

CÓDIGO	NOME	ENDEREÇO	TELEFONE
0001	Nei de Assis	Rua das Laranjeiras, 55	2222-1000
0002	Gertrudes	Av. das Américas, 199	1234-2000

3.3.1 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS (SGBD)

Para GUIMARÃES (1999), a criação dos Sistemas de Gerenciamentos de Banco de Dados (SGBD) possibilitou desvincular os dados de um Banco de Dados dos processos de programação, pois os SGBDs atuam como camadas intermediárias lógicas, localizando e manipulando os dados de interesse do sistema, isto é, os SGBDs fazem a interface entre o Banco de Dados e o usuário, permitindo operações de consulta, inserção, remoção e modificação de dados, e produzindo informações úteis a seus programas de aplicação.

Segundo GUIMARÃES (1999), as características de um SGBD que melhor demonstram as vantagens de uma estrutura de dados relacional podem ser vistas a seguir:

- Evita Redundância de dados, pois os mesmos são armazenados apenas uma vez.
- Evita Inconsistência de dados, pois quando o mesmo for atualizado não haverá conflito de informações. Se houver dados duplicados, o SGBD atualiza todos os campos onde o dado existe.
- Dados podem ser compartilhados, podendo ser acessados por várias aplicações diferentes.
- Maior Segurança, pois se pode definir qual usuário terá acesso aos dados.
- Maior Integridade, evitando entradas de dados como semanas com 200 horas, por exemplo.
- Garantia quanto à perda de dados no caso de qualquer falha no sistema.

HEUSER (1999) define um Sistema de Gerenciamento de Banco de Dados como um software que incorpora as funções de definição, recuperação e alteração de dados em um Banco de Dados. Ele afirma que essa modularização de programas tem várias vantagens. A manutenção de programas torna-se mais simples, pois uma separação clara de funções torna programas mais facilmente compreensíveis. A produtividade de programadores também aumenta, já que os programas ficam menores, pois usam funções já construídas.

Para JEPSON *et al.* (2000), um Sistema de Gerenciamento de Banco de Dados Relacional é um sistema de banco de dados onde os dados são organizados em tabelas que podem ser manipuladas com comandos tais como SELECT, UPDATE, INSERT e DELETE, possibilitando as tabelas serem vinculadas umas às outras com o uso da cláusula WHERE. Esses comandos são parte da Linguagem de Consulta Estruturada (SQL – Structured Query

Language) que é usada para manipular os dados de um banco de dados relacional..

3.3.2 PostgreSQL

Embora existam diversos tipos de Sistema de Gerenciamento de Banco de Dados Relacionais, nesse trabalho será instalado o PostgreSQL pelo fato de ser ele um Sistema de Gerenciamento amplamente usado em aplicações para a Internet, além de ser um dos mais populares entre os banco de dados com código-fonte aberto e possuir uma tecnologia complexa de banco de dados com um baixo custo operacional.

O PostgreSQL utiliza a linguagem padrão SQL (Structured Query Language), e compete com vários SGBD, entre eles o MySQL, que também é um poderoso e excelente SGBDR. A diferença está no tipo de licença e na diferença de desenvolvimento de aplicações. Enquanto que o PostgreSQL está disponível sob a licença BSD, o MySQL possui a Licença Pública GNU (GPL) além de uma licença convencional opcional. Quanto a diferença no desenvolvimento, o PostgreSQL é mais robusto e possui muito mais recursos que o tornam mais qualificado do que o MySQL.

Em alguns aspectos, o PostgreSQL é mais eficiente do que o MySQL pelo motivo de possuir um sofisticado mecanismo de bloqueio (MVCC-Multiversion Concurrency Control), suportar tamanhos ilimitados de linhas, bancos de dados e tabelas até 16 TB, aceitar vários tipos de subconsultas e contar com um bom mecanismo de *failsafe* (segurança contra falhas). Entretanto, quando se trata de um banco de dados muito grande, o MySQL tem a vantagem de acessar os dados mais rapidamente que o PostgreSQL (Juliano Niederauer - Integrando PHP5 com MySQL, São Paulo, Novatec Editora Ltda., 2005).

Segundo JEPSON *et al.* (2000), o PostgreSQL está baseado no POSTGRES, um banco de dados de pesquisa desenvolvido na Universidade da Califórnia. Quando se torna necessário desenvolver um banco de dados de topo de linha, deve-se considerar seriamente o PostgreSQL pelo fato de que ele pode ser utilizado em muitas variações do Unix, incluindo recursos encontrados em sistemas como Oracle e Sybase.

Segundo JEPSON *et al.* (2000), para se instalar o PostgreSQL em Ambiente Livre, é necessário seguir os seguintes procedimentos:

- Antes de instalar:

Normalmente, as distribuições do Linux já incluem o PostgreSQL, que geralmente

está no diretório `/usr/doc` com o nome `postgresql-x.y.z`, onde `x.y.z` é o número da versão do SGBD. Mas, se for decidido fazer uma instalação a partir do código-fonte, ele inclui os arquivos `README` e `INSTALL`, que ensinam como instalar o pacote. Entretanto, é aconselhável que se instale a versão que acompanha a distribuição do Linux ou que se procure as versões binárias existentes na *home-page* do PostgreSQL (`www.postgresql.org`) ou que se procure um *site* espelhado local, se possível, já que isto ajuda para evitar o congestionamento da rede no servidor principal.

- Depois de instalar:

Se a distribuição do Linux já inclui o PostgreSQL, pode-se configurar o banco de dados para iniciar ou parar no desligamento e configurar o ambiente para o seu uso. Caso contrário, procura-se o subdiretório `contrib/linux` da distribuição do fonte pelo *script* de inicialização que possa iniciar e parar o PostgreSQL, e para modificar a configuração das variáveis de ambiente necessárias basta seguir as instruções contidas no arquivo `INSTALL`.

O PostgreSQL envia comandos do SQL (*Structured Query Language*) para o servidor através de uma ferramenta interativa chamada **psql** que é executada a partir da linha de comando do Linux (*Shell* do Linux). Se for criado um usuário e um banco de dados, o *logon* é feito e o usuário é conectado. Uma vez conectado, basta digitar instruções em SQL seguida por “\g” ou “;” e pressionar *Enter*. Para sair, digitar “\q” e pressionar *Enter*.

3.3.3 POSTGIS

“PostGIS é uma extensão ao sistema de banco de dados object-relational PostgreSQL, que permite o uso de objetos GIS (Sistemas de Informação Geográfica) ser armazenado em banco de dados. PostGIS inclui suporte para índices *GiST-based* índices espaciais *R-Tree*, e funções para análise básica e processamento de objetos GIS.” (Fonte: <http://webgis.com.br/postgis>, 2006).

PostGIS também é um recurso computacional que deve ser instalado no computador para desenvolver o protótipo do modelo proposto no Capítulo 4 - Modelagem Proposta.

A documentação desta ferramenta é muito vasta e pode ser encontrada em língua portuguesa no site <http://webgis.com.br/postgis>.

3.3.4 MAPSERVER

MapServer é outro recurso computacional que, juntamente com o PostGIS será implementado no computador para auxiliar na construção de um aplicativo espacial na Internet.

Apesar de não ser um Sistema de Informação Geográfica completo, ele apresenta dados espaciais (mapas, imagens e dados vetoriais) na web.

A documentação completa do MapServer pode ser encontrada no site <http://mapserver.gis.umn.edu/>.

UCHÔA (2004), explica de uma forma sucinta a utilidade deste sistema e como esta ferramenta se tornou um estado-da-arte no desenvolvimento de soluções de SIG para a WEB:

“Este sistema permite o desenvolvimento de aplicações popularmente conhecidas como “servidor de mapas”. O MapServer é o carro-chefe das aplicações de código aberto para área de Geotecnologias com inúmeros casos de uso ao redor do mundo.

Para se ter uma idéia inicial do quanto esta ferramenta é completa, basta olhar a TAB.5.1 com os formatos suportados por este sistema.

Para quem não tem experiência em programação, o MapServer fornece um CGI com inúmeras funcionalidades para desenvolvimento de aplicações mais simples de SIG em ambiente WEB. Dentre estas funcionalidades, podemos citar as principais:

- Suporta aos formatos de vetores: ESRI® Shapefiles, PostGIS, ESRI® ArcSDE (versão alfa), etc;
- Suporte ao formato matricial (Apenas 8 bit): TIFF/GeoTIFF, GIF, PNG, ERDAS, JPEG e EPPL7;
- Indexação espacial *quadtree* para shapefiles;
- Customizável através de templates;
- Seleção de características por ítem/valor, ponto, área ou outra característica;
- Suporte à fonte TrueType;
- Suporte para dados matriciais e vetoriais;
- Geração automática de legenda e barra de escala;
- Geração de mapas temáticos usando expressões lógicas ou regulares baseadas em classes;
- Característica de rotulação (*labels*) incluindo mediação de colisão de rótulos;
- Configuração dinâmica através de URLs;

- Projeção dinâmica.

Para programadores mais experientes, o MapServer fornece um completo API que pode ser acessado através de Python, Perl, PHP, Java e C (linguagem nativa). O site oficial oferece uma excelente (e vasta) documentação para quem desejar trabalhar com esta aplicação. Também é possível obter ajuda nas comunidades de usuários dentro e fora do país, sendo o Brasil um dos países colaboradores no desenvolvimento do sistema. A única ressalva com relação às linguagens que permitem acesso ao API é que o suporte à Java é muito deficiente.

Com relação às especificações do OGC, este sistema trabalha com o SFS (Através do PostGIS), WFS (somente leitura), WMS e GML. Já está em desenvolvimento, segundo o site oficial, o padrão WCS.” (Fonte: Geoprocessamento com Software Livre (v 1.0) – Helton Nogueira Uchoa / Paulo Roberto Ferreira).

3.4 MODELAGEM DE SISTEMA DE BANCO DE DADOS

Segundo HEUSER (1999), um modelo de (Banco de) dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados. O modelo de dados não informa quais os produtos que estão armazenados, mas apenas que o banco de dados contém informações sobre produtos. Em outras palavras, HEUSER define o modelo de dados como uma descrição formal da estrutura de um banco de dados. Segundo ele, para se construir um modelo de dados, usa-se uma linguagem de modelagem de dados, que pode ser classificada de acordo com a forma de apresentar modelos, textos ou gráficos. O modelo pode ser chamado de esquema de banco de dados. Ele explica que um banco de dados pode ser descrito a vários níveis de abstração, de acordo com a intenção do modelador, quer seja para explicar a organização do banco de dados a um usuário leigo em informática, quer seja para informar detalhes de como as informações estão organizadas para um técnico de informática. Num projeto de banco de dados, normalmente há três tipos de modelos de dados: o modelo conceitual, o modelo lógico e o modelo físico.

3.4.1 MODELO CONCEITUAL DE BANCO DE DADOS

HEUSER (1999) define o Modelo Conceitual como um modelo de dados abstrato, que descreve a estrutura de um banco de dados de forma independente de um Sistema de Gerenciamento de Banco de Dados (SGBD) particular. O modelo conceitual registra dados no

banco de dados, mas não como eles estão armazenados ao nível de Sistema de Gerenciamento de Banco de Dados.

O Modelo Conceitual é normalmente representado através de um diagrama chamado entidade-relacionamento (DER). A FIG.3.4 apresenta um DER parcial para o problema de uma fábrica.

O exemplo de modelo conceitual da FIG.3.4 contém informações sobre os produtos e os respectivos tipos de produto que estão relacionados entre si. Para a entidade Produto, o banco armazena o Código do produto, a Descrição do produto, o Preço e o Código do Tipo de Produto. Para a entidade Tipo de Produto, o banco armazena o Código do tipo de produto e a Descrição do tipo de produto.

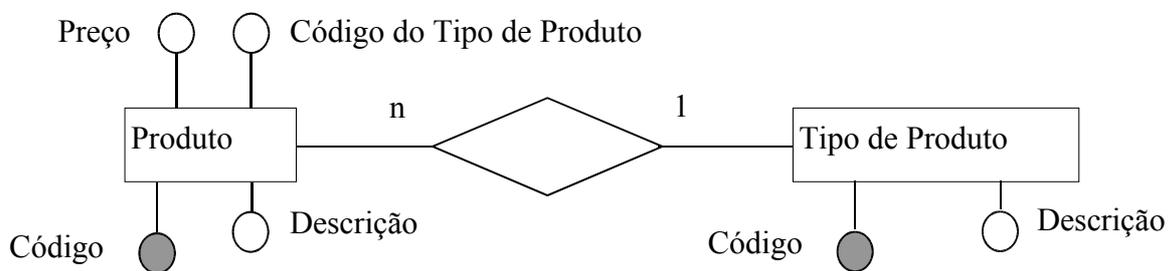


FIG.3.4: Exemplo De Modelo Conceitual. (Fonte: Projeto de Banco de Dados – Carlos Alberto Heuser, 1999)

A abordagem entidade-relacionamento (ER) foi criada em 1976 por Peter Chen, que é considerada como padrão para a modelagem conceitual. Mesmo a modelagem orientada a objetos se baseia nos conceitos da abordagem ER.

Segundo HEUSER (1999), os conceitos centrais da abordagem ER são os seguintes:

- Entidade: é o conjunto de objetos da realidade modelada sobre os quais deseja-se manter informações no banco de dados. Aqui interessa somente os objetos sobre os quais deseja-se manter informações. No caso da FIG.3.4, as entidades são os produtos e os tipos de produtos. As entidades podem representar tanto objetos concretos da realidade quanto abstratos. A entidade é representada através de um retângulo com o nome da entidade.
- Relacionamento: é o conjunto de associações entre entidades. O relacionamento é

representado através de um losango, ligado por linhas aos retângulos representativos das entidades que participam do relacionamento. A FIG.3.5 apresenta um DER contendo as entidades EMPREGADO e DEPARTAMENTO com um relacionamento LOTAÇÃO.



FIG.3.5: Representação Gráfica de Relacionamento

O exemplo acima informa que o banco de dados mantém informações sobre um conjunto de objetos classificados como empregados, e outro como departamentos, e outro conjunto de associações, cada uma ligando um departamento a um empregado.

Nem sempre um relacionamento associa entidades diferentes. Um DER pode conter um auto-relacionamento, ou seja, um relacionamento entre ocorrências de uma mesma entidade. Neste caso, é necessário um conceito adicional, o de *papel* da entidade no relacionamento. O papel de uma entidade define que função uma instância da entidade cumpre dentro de uma instância do relacionamento. Como exemplo na FIG.3.6, supondo-se uma identidade PESSOA e um relacionamento CASAMENTO, uma ocorrência de pessoa exerce o papel de marido e a outra ocorrência de pessoa exerce o papel de esposa.

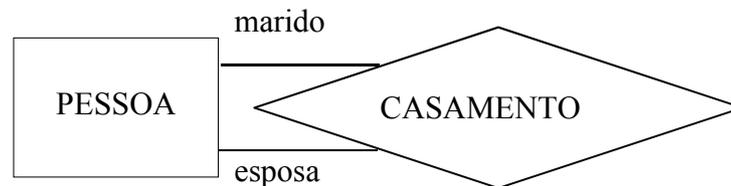


FIG.3.6: Auto-relacionamento

- A Cardinalidade de Entidade em Relacionamento: é o número (mínimo, máximo) de ocorrências de entidade associadas a uma ocorrência da entidade em questão através do relacionamento, isto é, é a quantidade de ocorrências que uma entidade pode estar associada a uma determinada ocorrência através do relacionamento. No exemplo da FIG.3.5, a entidade EMPREGADO tem cardinalidade máxima 1 no relacionamento LOTAÇÃO, isto é, uma ocorrência de EMPREGADO pode estar

associada a no máximo 1 ocorrência de DEPARTAMENTO, enquanto que a entidade DEPARTAMENTO tem cardinalidade máxima n no relacionamento LOTAÇÃO, isto é, uma ocorrência de DEPARTAMENTO pode estar associada a no máximo n ocorrências de EMPREGADO. Na prática, usam-se apenas duas cardinalidades: 1 e “muitos”, referida pela letra n.

- **Atributo:** é o dado que é associado a cada ocorrência de uma entidade ou de um relacionamento, isto é, o atributo serve para associar informações a ocorrências de entidades ou de relacionamentos. Na prática os atributos não são representados graficamente para não sobrecarregar os diagramas.
- **Identificador:** é um conjunto de atributos e relacionamentos cujos valores distinguem uma ocorrência da entidade das demais.
- **Generalização/especialização:** além de relacionamentos e atributos, as propriedades podem ser atribuídas a entidades através do conceito de Generalização/Especialização. É possível atribuir propriedades particulares a um subconjunto das ocorrências (especializadas) de uma entidade genérica. No diagrama entidade-relacionamento (DER), o símbolo é um triângulo isóscele. Associada ao conceito de generalização/especialização está a idéia de herança de propriedades, que significa que cada ocorrência da entidade especializada possui, além de suas próprias propriedades, também as propriedades da ocorrência da entidade genérica correspondente.
- **Entidade Associativa:** é a redefinição de um relacionamento, que passa a ser tratado como se fosse também uma entidade.

HEUSER (1999) explica que o esquema Entidade-Relacionamento pode estar na forma textual e que nesta forma tem uma notação mais rica que a notação gráfica. Na prática, pode-se combinar as duas formas de representação esquemática. A seguir é apresentado um exemplo da sintaxe referente a um esquema ER de um sistema intermodal com acesso intermodal na forma textual:

Esquema: SISTEMA_INTERMODAL

Entidade: ACESSO INTERMODAL

Atributo:

CÓDIGO DO ACESSO: CHAR(10)

TIPO DE ACESSO

CAPACIDADE DO ACESSO

Identificador: CÓDIGO DO ACESSO

Entidade: ACESSO FERROVIÁRIO

Atributo:

TIPO DE LOCOMOTIVA

POTÊNCIA DA LOCOMOTIVA

BITOLA DO ACESSO

Identificador: ACESSO INTERMODAL via ACESSO_ESPECIAL

Entidade: ACESSO RODOVIÁRIO

Atributo:

LARGURA DA FAIXA

CLASSE DE PAVIMENTAÇÃO

QUANTIDADE DE FAIXAS

LARGURA DO ACOSTAMENTO

Identificador: ACESSO INTERMODAL via ACESSO_ESPECIAL

Relacionamento: LIGA

3.4.2 MODELO LÓGICO DE BANCO DE DADOS

FERREIRA FILHO (1998) descreve os diferentes níveis de representação dos Modelos de Dados durante o desenvolvimento de um Sistema de Informação e apresenta a Abordagem de Projeto de Sistemas com Foco nos Dados na FIG.3.7:

- 1.Modelos Conceituais: descrevem a estrutura do Banco de Dados em alto nível, especificando entidades, atributos, restrições, etc.
- 2.Modelos Lógicos: descrevem a estrutura do Banco de Dados em nível intermediário, indicando como os dados são organizados no computador.
- 3.Modelos Físicos: descrevem a organização dos arquivos e os métodos de acesso.

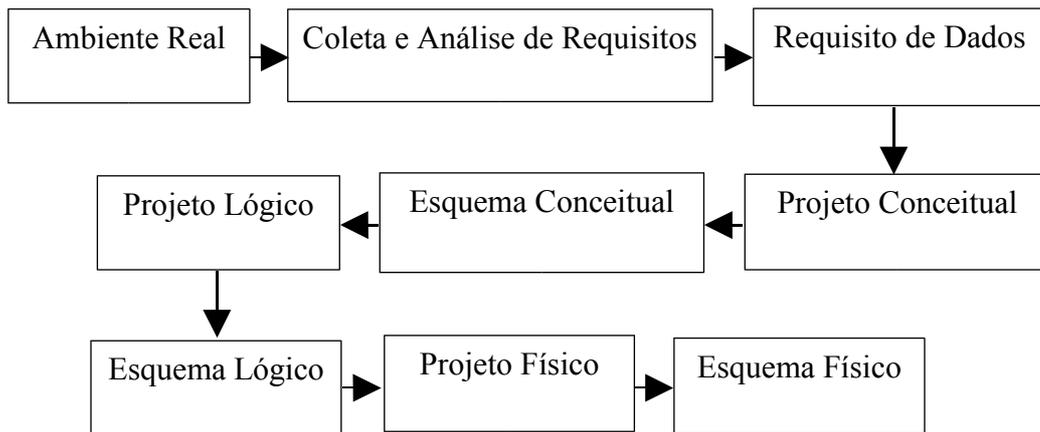
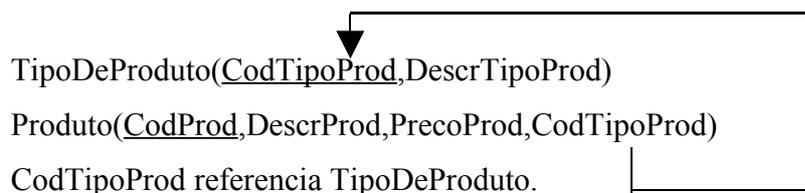


FIG.3.7: Projeto de Sistemas Com Foco Nos Dados

HEUSER (1999) define o modelo lógico como um modelo de dados que representa a estrutura de dados de um banco de dados conforme vista pelo usuário do Sistema de Gerenciamento de Banco de Dados (SGBD). Em outras palavras, é uma descrição de um banco de dados ao nível de abstração visto pelo usuário do SGBD. Por este motivo, o modelo lógico é dependente do tipo particular de SGBD que está sendo usado.

HEUSER explica que um modelo lógico de um banco de dados relacional deve definir quais as tabelas que o banco contém, com os respectivos nomes de colunas. Como exemplo temos o modelo lógico da FIG.3.4 apresentada anteriormente:



O banco armazena para o Produto, o código (CodProd), a descrição (DescrProd), o preço (PrecoProd) e o código do tipo de produto (CodTipoProd), que é a chave estrangeira de Produto, isto é, é o campo que referencia Tipo de Produto através da sua chave primária CodTipoProd.

Por sua vez, o banco armazena para o Tipo de Produto, o código do tipo de produto (CodTipoProd) e a descrição do tipo de produto (DescrTipoProd).

3.4.3 MODELO FÍSICO DE BANCO DE DADOS

Segundo HEUSER (1999), os detalhes de armazenamento interno de informação que não influencia as aplicações no SGBD, mas sim a estrutura de arquivos usada no acesso às informações, é representada no Modelo Físico. As notações e linguagens do modelo físico variam de produto a produto e a tendência moderna é esconder o modelo físico do usuário e transferir a tarefa de otimização ao próprio SGBD.

FERREIRA FILHO (1998), explica que o Modelo Físico descreve a organização dos arquivos e os métodos de acesso, isto é, como os dados se encontram distribuídos no computador. Isto é obtido através de informações tais como: tipos de arquivos, formatos diversos, ordenação de registros, métodos de acesso, índices, etc. Para gerar Esquemas Físicos de Dados, usam-se ferramentas CASE, que mapeiam um Esquema Lógico existente.

3.5 LINGUAGEM DE MODELAGEM UNIFICADA - UML

A Linguagem de Modelagem Unificada (*Unified Modelling Language* – UML) é uma linguagem de diagramação ou notação para especificar, visualizar e documentar modelos de sistemas de software Orientados a Objeto. Pode-se representar várias idéias em UML usando-se diferentes tipos de diagramas.

Neste trabalho, é feita uma rápida introdução sobre a UML, pois não faz parte do escopo dessa dissertação aprofundar no estudo dessa ferramenta e sim como usá-la para o desenvolvimento de um modelo voltado para o sistema de transportes. Existem muitos tutoriais detalhados na Internet acerca do assunto.

A UML não é um método de desenvolvimento, o que significa que ela não diz o que fazer primeiro e em seguida como desenhar o sistema. Ela auxilia a visualizar o desenho do modelo e como comunicar os objetos entre si. A UML é voltada para o desenho de software Orientado a Objeto e tem um uso limitado para outros paradigmas de programação.

A UML é composta por muitos elementos de modelo que representam as diferentes partes de um sistema de software. Os elementos UML são usados para criar diagramas que representam uma determinada parte ou um ponto de vista do sistema.

JEPSON *et al.* (2000) diz que a UML é uma notação para representar os problemas e especificar o projeto de sistemas que lidam com os problemas. Esta linguagem evoluiu a partir

de raízes profundas no processo e notação orientados a objetos e é reconhecida como um padrão para a modelagem dentro da Análise e Projeto Orientados a Objetos (A&POO). Por ser padrão, o UML não limita o usuário a desenvolver um modelo de certo modo ou de certa ordem. Pode-se desenvolver um modelo genérico e escolher apenas algumas partes desse modelo.

JEPSON *et al.* (2000) afirma que a fase da análise do processo está preocupada com a expansão da sua compreensão do domínio do problema (A área que o problema afeta). Por isso, a análise procura modelar os problemas do ponto de vista de um especialista do domínio (Alguém que sabe tudo sobre o problema).

A seguir, dentro da fase de análise do processo, é apresentada uma ferramenta para diagramação UML que serve de base para o desenvolvimento do modelo do sistema de transporte proposto nessa dissertação.

3.5.1 FERRAMENTA PARA DIAGRAMAÇÃO UML

Existem diversas ferramentas em ambiente livre que podem ser usadas para diagramação e modelagem: DIA – para desenhar qualquer tipo de diagrama; Argo/UML – produz um sistema de suporte cognitivo para projetos dirigidos a objetos; DOME – para o desenvolvimento e uso de modelos de software CASE (*Computer Aided Software Engineering* – Engenharia de Software Auxiliada por Computador); TGIF – ferramenta de desenho vetorial interativo, usada também para produzir diversos diagramas; e o Modelador UML Umbrello – uma ferramenta de diagramação que ajuda no processo de desenvolvimento de um *software* durante a fase de análise e projeto.

O Umbrello foi escolhido como a ferramenta ideal para desenvolver o modelo dessa dissertação por dois motivos: primeiro pelo fato de que o UML Umbrello tem uma versão que acompanha o Linux Conectiva 10, usado na elaboração desse estudo, e segundo porque possui um ambiente bastante amigável, documenta eficazmente o projeto do sistema e gera os códigos em diversas linguagens de programação como a do SQL, Java, C++ e outras. O Umbrello pode ser baixado da home page <http://uml.sourceforge.net/>.

A) JANELA DE INTERFACE COM O USUÁRIO DO MODELADOR UMBRELLO UML

A janela principal do Modelador Umbrello UML na FIG.3.8 é dividida em três áreas: Visão de Árvore, Janela de Documentação e Área de Trabalho.

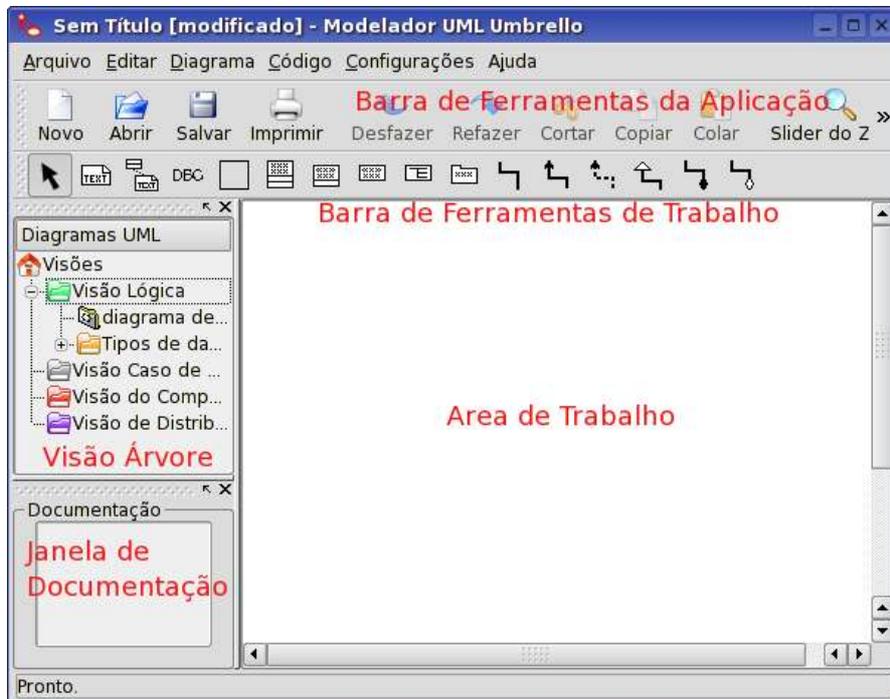


FIG.3.8: Janela Principal do Modelador Umbrello UML

Se alguém está trabalhando em um modelo com mais do que algumas poucas classes e diagramas, a Visão de Árvore pode auxiliar a manter no topo as coisas, organizando seus elementos do modelo em pastas. Pode-se criar pastas selecionando a opção apropriada no menu de contexto (clique botão direito do mouse em uma das pastas na visão de árvore) e pode-se organizar seus elementos movendo-os para a pasta apropriada (Arrastar e soltar)

A-1) JANELA DE DOCUMENTAÇÃO

A Janela de Documentação é a pequena janela localizada na parte inferior esquerda do Modelador Umbrello UML e fornece uma rápida previsão da documentação para o item atualmente selecionado. A Janela de Documentação é um pouco pequena porque ela pretende permitir apenas uma rápida observação dos elementos da documentação ocupando o menor espaço possível. Se alguém precisar ver a documentação com mais detalhe basta abrir as propriedades do item.

A-2) ÁREA DE TRABALHO

A Área de Trabalho é a janela principal do Modelador Umbrello UML, e é onde as ações reais ocorrem. A Área de Trabalho é usada para editar e ver os diagramas em seu modelo. Atualmente, somente um diagrama pode ser mostrado na Área de Trabalho de cada vez (http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/working-with-umbrello.html, 2006).

O Modelador Umbrello UML 1.2 suporta os seguintes tipos de Diagrama com os seus respectivos componentes:

- Caso de Uso: atores (pessoas ou outros usuários do sistema), casos de uso (os cenários onde eles usam o sistema) e seus relacionamentos.
- Classe: classes e os seus relacionamentos.
- Seqüência: objetos e uma seqüência das chamadas do método feitas para outros objetos.
- Colaboração: objetos e seus relacionamentos, colocando ênfase nos objetos que participam na troca de mensagens.
- Estado: estados, mudanças de estado e eventos num objeto ou uma parte do sistema.
- Atividade: atividades e as mudanças de uma atividade para outra com os eventos ocorridos em alguma parte do sistema.
- Componente: componentes de programação de alto nível (como KParts ou Java Beans).
- Distribuição: instâncias dos componentes e seus relacionamentos.

Os Diagramas de Caso de Uso, de Classe e de Atividade são os diagramas que foram selecionados para mostrar o desenvolvimento do modelo do sistema de informação de transporte multimodal proposto nesta dissertação.

B) DIAGRAMA DE CASO DE USO

Diagramas de Caso de Uso (FIG.3.9) descrevem relacionamentos e dependências entre um grupo de *Caso de Uso* e os Atores participantes no processo.

Diagramas de Caso de Uso não são adequados para representar o desenho e não podem descrever os mecanismos internos de um sistema. Diagramas de Caso de Uso são feitos para facilitar a comunicação com os futuros usuários do sistema, com o cliente e são especialmente úteis para determinar os recursos necessários que o sistema deve ter. Diagramas de Caso de

Uso dizem o quê o sistema deve fazer, mas não faz especifica como isto será conseguido.

B-1) CASO DE USO

Um Caso de Uso descreve — do ponto de vista dos atores — um grupo de atividades num sistema que produz um resultado concreto e tangível.

Casos de Uso são descrições de interações típicas entre os usuários de um sistema e o sistema propriamente dito. Eles representam a interface externa do sistema e especificam um conjunto de exigências do que o sistema deve fazer .

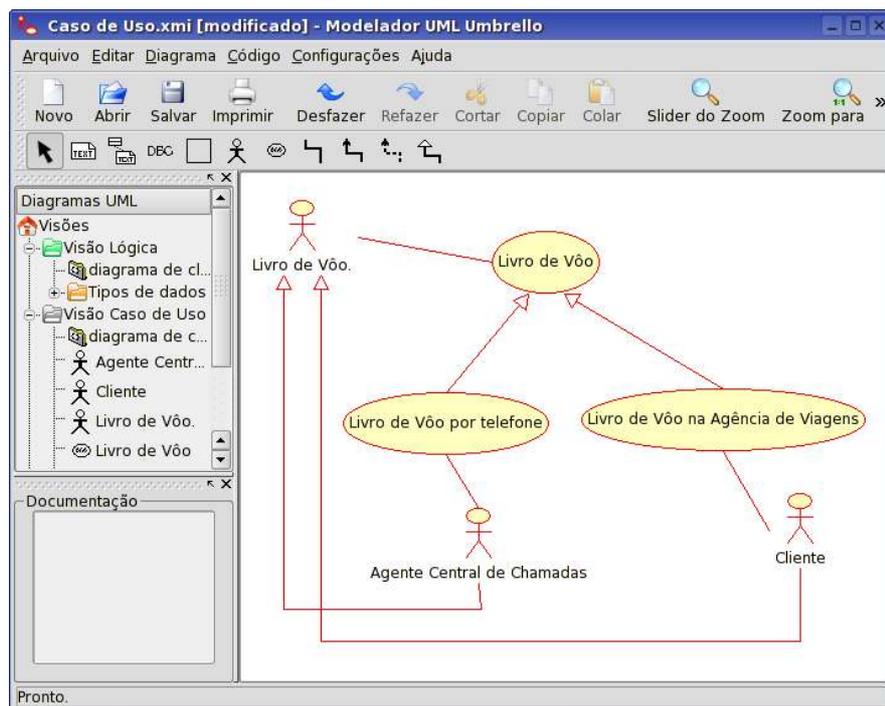


FIG.3.9: Janela do Diagrama de Caso de Uso - UML

Quando trabalhar com Casos de Uso, é importante lembrar-se de algumas regras simples:

- Cada Caso de Uso está relacionado com, no mínimo, um ator;
- Cada Caso de Uso possui um iniciador (isto é um ator);
- Cada Caso de Uso liga-se a um resultado relevante (um resultado com “valor de negócio”).

Casos de Uso também podem ter relacionamentos com outros Casos de Uso. Os três tipos mais comuns de relacionamento entre Casos de Uso são:

- *inclui-se* que especifica que um Caso de Uso toma lugar *dentro* de outro Caso de Uso;
- *estende* que especifica que em determinadas situações, ou em algum ponto (chamado um ponto de extensão) um Caso de Uso será estendido por outro;
- *Generalização* especifica que um Caso de Uso herda as características do “Super” Caso de Uso, e pode sobrepor algumas delas ou adicionar novas de maneira semelhante a herança entre classes.

B-2) ATOR

Um ator é uma entidade externa (fora do sistema) que interage com o sistema participando (e freqüentemente iniciando) um Caso de Uso. Atores podem ser pessoas reais (por exemplo, usuários do sistema), outro sistema de computador ou eventos externos.

Atores não representam as pessoas físicas ou sistemas, mas sua regra. Isto significa que quando uma pessoa interage com o sistema de diferentes maneiras (Assumindo diferentes regras) ela será representada por diversos atores. Por exemplo, uma pessoa que fornece suporte ao cliente por telefone e recebe ordens do cliente para o sistema pode ser representado por um ator da “Equipe de Suporte” e um ator “Representante de Vendas”.

B-3) DESCRIÇÃO DO CASO DE USO

Descrição do Caso de Uso é narrativa de texto do Caso de Uso. Ela usualmente toma a forma de uma nota ou um documento que é de alguma maneira ligado ao Caso de Uso, e explana o processo ou atividades que tomam lugar no Caso de Uso (http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-elements.html, 2006).

C) DIAGRAMA DE CLASSE

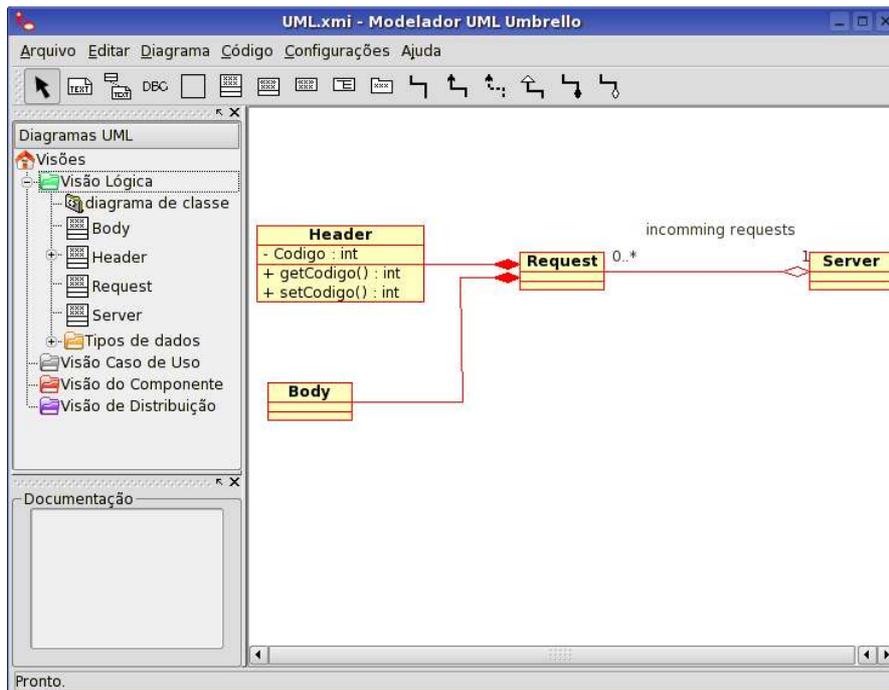


FIG.3.10: Janela do Diagrama de Classe - UML

Diagramas de Classe (FIG.3.10) mostram as diferentes classes que fazem um sistema e como elas se relacionam. Os Diagramas de Classe são chamados diagramas “estáticos” porque mostram as classes, com seus métodos e atributos bem como os relacionamentos estáticos entre elas: quais classes “conhecem” quais classes ou quais classes “são partes” de outras classes, mas não mostram a troca de mensagens entre elas.

C-1) CLASSE

Uma Classe define os atributos e os métodos de um conjunto de objetos. Todos os objetos desta classe (instâncias desta classe) compartilham o mesmo comportamento e possuem o mesmo conjunto de atributos (cada objeto possui seu próprio conjunto). O termo “Tipo” é algumas vezes usado ao invés de Classe, mas é importante mencionar que estes dois termos não são a mesma coisa, e Tipo é um termo mais genérico.

Em UML, Classes são representadas por retângulos, com o nome da classe e podem também mostrar os atributos e operações da classe nos dois outros compartimentos dentro do retângulo.

C-2) ATRIBUTOS

Na UML, atributos são mostrados com, pelo menos, seu nome e podem também mostrar seu tipo, valor inicial e outras propriedades. Atributos podem também ser exibidos com sua visibilidade:

- + indica atributos *públicos*;
- # indica atributos *protegidos*;
- - indica atributos *privados*.

C-3) OPERAÇÕES

Operações (métodos) também são exibidas com pelo menos seu nome, e podem também mostrar seus parâmetros e valores de retorno. Operações podem, como os Atributos, mostrar sua visibilidade:

- + indica operações *públicas*;
- # indica operações *protegidas*;
- - indica operações *privadas*.

C-4) MODELOS

Classes podem ter modelos, um valor que é usado para uma classe ou tipo não especificado. O tipo de modelo é especificado quando uma classe é iniciada (isto é um objeto é criado). Modelos existem no C++ moderno e foram introduzidos no Java 1.5 onde eles são chamados de Genéricos.

C-5) ASSOCIAÇÕES DE CLASSE

As Classes podem relacionar-se (serem associadas) com outras de diferentes maneiras:

Generalização:

Herança é um dos conceitos fundamentais da programação Orientada a Objeto, na qual uma classe “herda” todos os atributos e operações da classe da qual deriva e pode sobrescrever/modificar alguns deles, bem como adicionar mais atributos e operações próprios.

Em UML, uma Generalização entre duas classes as coloca numa hierarquia representando o conceito de herança de uma classe derivada de uma classe base. Generalizações são representadas por uma linha conectando duas classes, com uma seta no

lado da classe base.

Associações:

Uma associação representa um relacionamento entre classes e fornece a semântica comum e a estrutura para muitos tipos de “conexões” entre objetos.

Associações são os mecanismos que permitem os objetos comunicarem-se entre si. Elas descrevem a conexão entre diferentes classes (A conexão entre os objetos atuais é chamada conexão do objeto ou *link*).

Associações podem ter uma regra que especifica o propósito da associação e pode ser uni ou bi-direcional (indicando se os dois objetos participantes do relacionamento podem mandar mensagens para o outro, ou se apenas um deles sabe sobre o outro). Cada ponta da associação também possui um valor de multiplicidade, que dita como muitos objetos num lado da associação podem relacionar-se com o outro lado.

Em UML, associações são representadas como linhas conectando as classes participantes do relacionamento e podem também mostrar a regra e a multiplicidade de cada um dos participantes. A multiplicidade é exibida como um intervalo [min...máx] de valores não negativos, com uma estrela (*) no lado máximo representando infinito.

Agregação:

Agregações são tipos especiais de associação na qual as duas classes participantes não possuem níveis iguais, mas fazem um relacionamento “todo-parte”.

Uma Agregação descreve como a classe que possui a regra do todo é composta (tem) de outras classes, que possuem a regra das partes. Para Agregações, a classe que age como o todo sempre tem uma multiplicidade de um.

Em UML, Agregações são representadas por uma associação que mostra um rombóide no lado do todo.

Composição:

Composições são associações que representam agregações muito fortes. Isto significa que Composições formam relacionamentos todo-parte também, mas o relacionamento é tão forte que as partes não podem existir independentes. Elas existem somente dentro do todo, e se o todo é destruído as partes morrem também.

Em UML, Composições são representadas por um rombóide sólido no lado do todo.

C-6) OUTROS ITENS DO DIAGRAMA DE CLASSE

Diagramas de Classe podem conter diversos outros itens além das classes.

Interfaces:

Interfaces são classes abstratas que significam instâncias que não podem ser diretamente criadas delas. Elas podem conter operações mas não podem conter atributos. Classes podem derivar de interfaces (Através da realização de uma associação) e instâncias podem então ser feitas destes diagramas.

Tipos de dados:

Tipos de dados são primitivos, uma vez que são tipicamente construídos numa linguagem de programação. Exemplos comuns são inteiros e lógicos. Eles não podem ser relacionados às classes, mas as classes podem se relacionar a eles.

Enumerações:

Enumerações são uma lista simples de valores. Um exemplo típico é uma enumeração para dias da semana. As opções de uma enumeração são chamadas Literais de Enumeração. Como tipos de dados, elas não podem ter relacionamentos às classes mas as classes podem se relacionar a elas.

Pacotes:

Pacotes representam um espaço de nomes numa linguagem de programação. Num diagrama, eles são usados para representar partes de um sistema que contém mais de uma classe (http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-elements.html, 2006).

D) DIAGRAMA DE ATIVIDADE

O Diagrama de Atividade descreve a seqüência de atividades num sistema com a ajuda das atividades. O Diagrama de Atividade é uma forma especial do Diagrama de Estado, que somente (ou principalmente) contém Atividades.

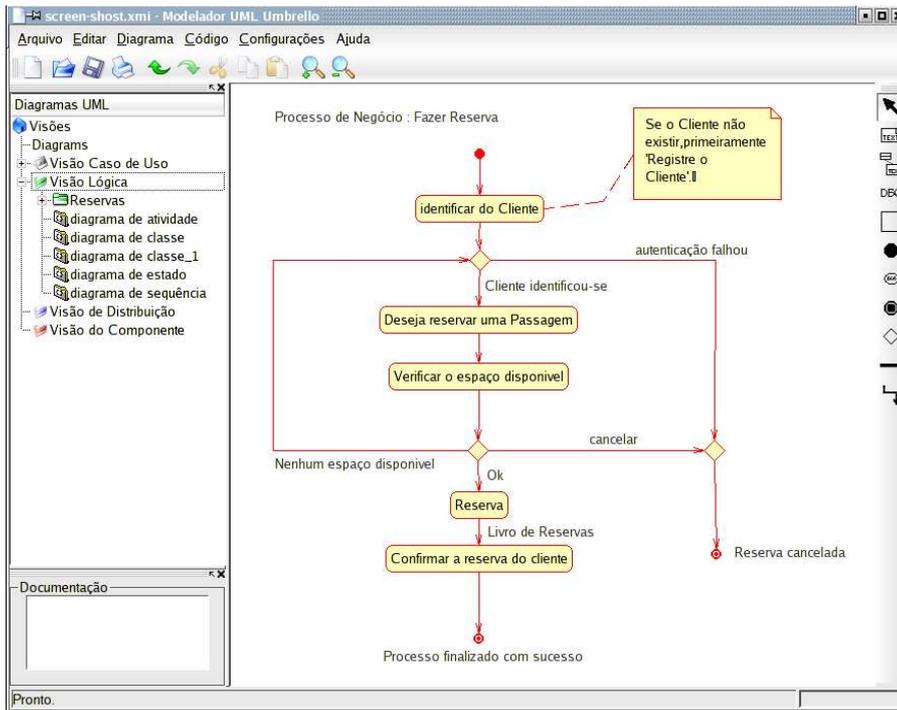


FIG.3.11: Janela do Diagrama de Atividade - UML

Diagramas de Atividade também são similares aos Diagramas de Fluxo de procedimentos, com a diferença de que todas as Atividades são claramente anexas aos Objetos e são sempre associadas a uma *Classe*, uma *Operação* ou um *Caso de Uso*.

Os Diagramas de Atividade suportam Atividades seqüenciais bem como paralelas. A execução paralela é representada pelos ícones Forquilha/Esperar. Para as Atividades executadas em paralelo não é importante a ordem na qual elas são executadas (podem ser executadas ao mesmo tempo ou uma após a outra).

D-1) ATIVIDADE

Uma Atividade é um passo simples num processo. É um estado no sistema com atividade interna e, pelo menos, uma transição de saída. Atividades podem também ter mais de uma transição de saída, se possuem condições diferentes.

Atividades podem formar hierarquias. Isto significa que uma Atividade pode ser composta por diversas Atividades em “detalhe”, na qual as transições de entrada e saída devem corresponder às transições de entrada e saída do diagrama de detalhe.

D-2) ELEMENTOS AUXILIARES

Existem três elementos em UML que não possuem nenhum valor real semântico para o modelo, mas auxiliam a elucidar partes do diagrama. Estes elementos são:

- Linhas de texto:

Linhas são úteis para adicionar informações curtas de texto ao diagrama. São textos livres e não possuem nenhum significado para o Modelo propriamente dito.

- Notas de Texto e Âncoras:

Notas são úteis para adicionar informações mais detalhadas sobre um objeto ou situação específica. Elas possuem a grande vantagem de poderem ser ancoradas a Elementos UML para mostrar que a nota pertence a um objeto específico ou situação.

- Caixas:

Caixas são retângulos de forma livre que podem ser usados para agrupar itens tornando os diagramas mais legíveis. Eles não possuem nenhum significado lógico no modelo.

http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-elements.html

4 MODELAGEM PROPOSTA E PROTOTIPAÇÃO

4.1 INTRODUÇÃO

O Sistema de Informação para Transporte Multimodal de Carga (SITMMC) que está sendo criado nesta dissertação é composto de três camadas distintas: Terminais, Veículos e Informações Geográficas. Cada camada possui uma coleção de dados inter-relacionados que visam atender solicitações de usuários referentes às informações contidas nos mesmos.

Para que o usuário tenha acesso aos dados, ele precisa estar cadastrado no SITMMC. Se não estiver cadastrado, o usuário solicita através do sistema a sua inclusão. Uma vez incluído, ele pode localizar dados de terminais, veículos ou informações geográficas para consultas e pode requisitar inclusão, exclusão ou alteração de dados. Ao completar a operação solicitada, o usuário recebe uma confirmação do sistema sobre a inclusão ou modificação de dados ou, se houver algum erro, recebe uma mensagem que aponte para este erro.

A base de dados usada para o sistema de informação geográfica é uma base de dados fornecida pelo Instituto Militar de Engenharia.

A modelagem proposta é composta das seguintes fases de desenvolvimento:

- Criação do diagrama de casos de uso com descrição
- Criação do diagrama de atividades
- Criação do diagrama de classe
- Criação do projeto conceitual

4.2 DIAGRAMA DE CASOS DE USO COM DESCRIÇÃO

Neste estudo, os atores e os processos foram identificados de acordo com as instruções do item 3.5.1-B (Ferramenta Para Diagramação UML – Diagrama de Caso de Usos).

O método usado para identificar os casos de uso foi o de LARMAN (1998): Identificar todos os participantes envolvidos com os requisitos (Terminais, veículos e informações geográficas) e, depois, identificar os processos ou eventos que estão envolvidos (com base no ator).

Neste estudo, os atores do sistema são o Pesquisador e o Gerente do sistema (FIG.4.1).

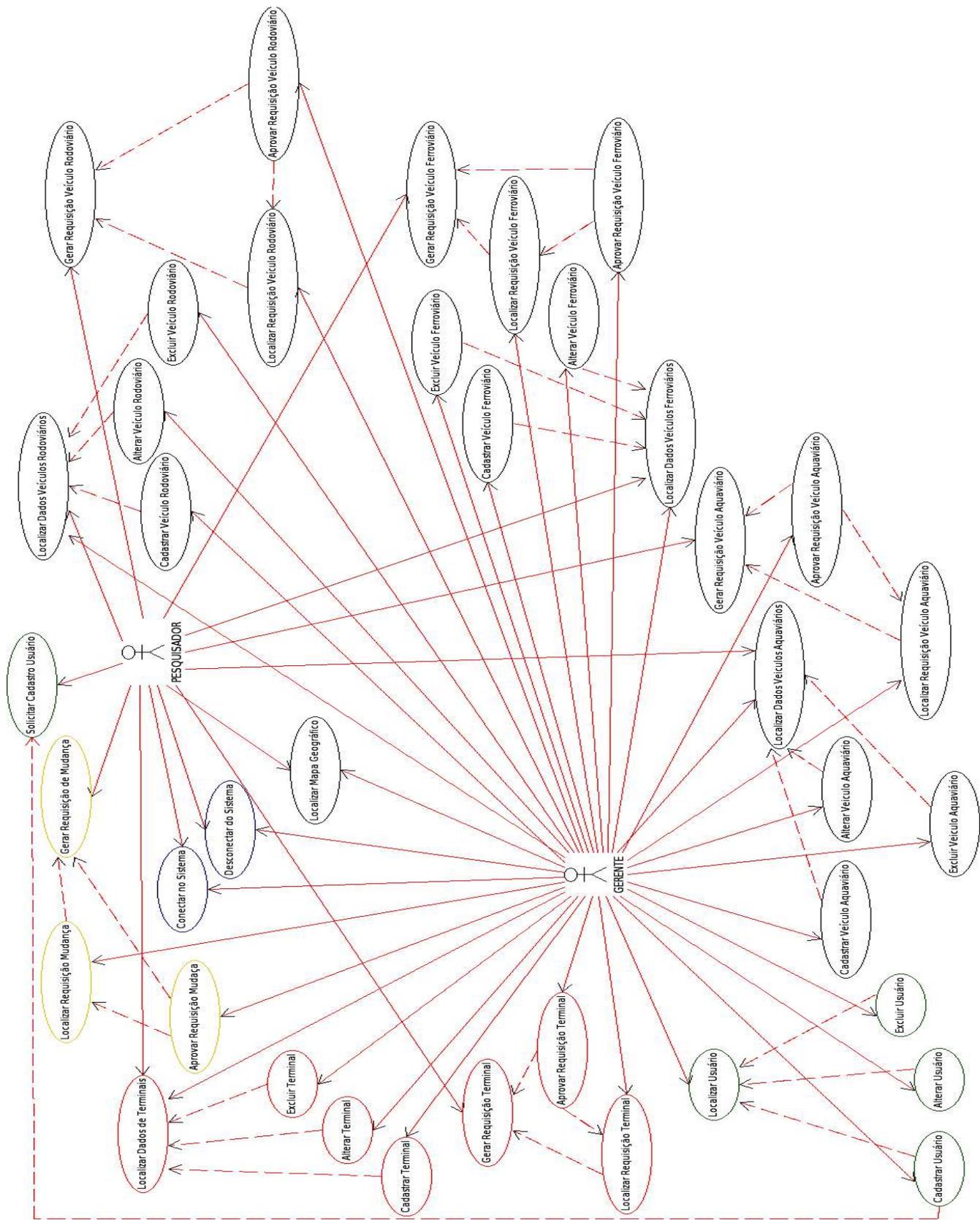


FIG.4.1: Diagrama de Casos de Uso

No APÊNDICE 1 é apresentada uma descrição detalhada de cada Casos de Uso mostrando as pré-condições, a descrição geral e a descrição detalhada dos fluxos operacionais do sistema, possibilitando uma compreensão expandida de todos os requisitos.

4.3 DIAGRAMA DE ATIVIDADES

A partir do diagrama de casos de uso, foi criado o Diagrama de Atividades (FIG.4.2), que descreve a seqüência de atividades desde o início até o término do processo.

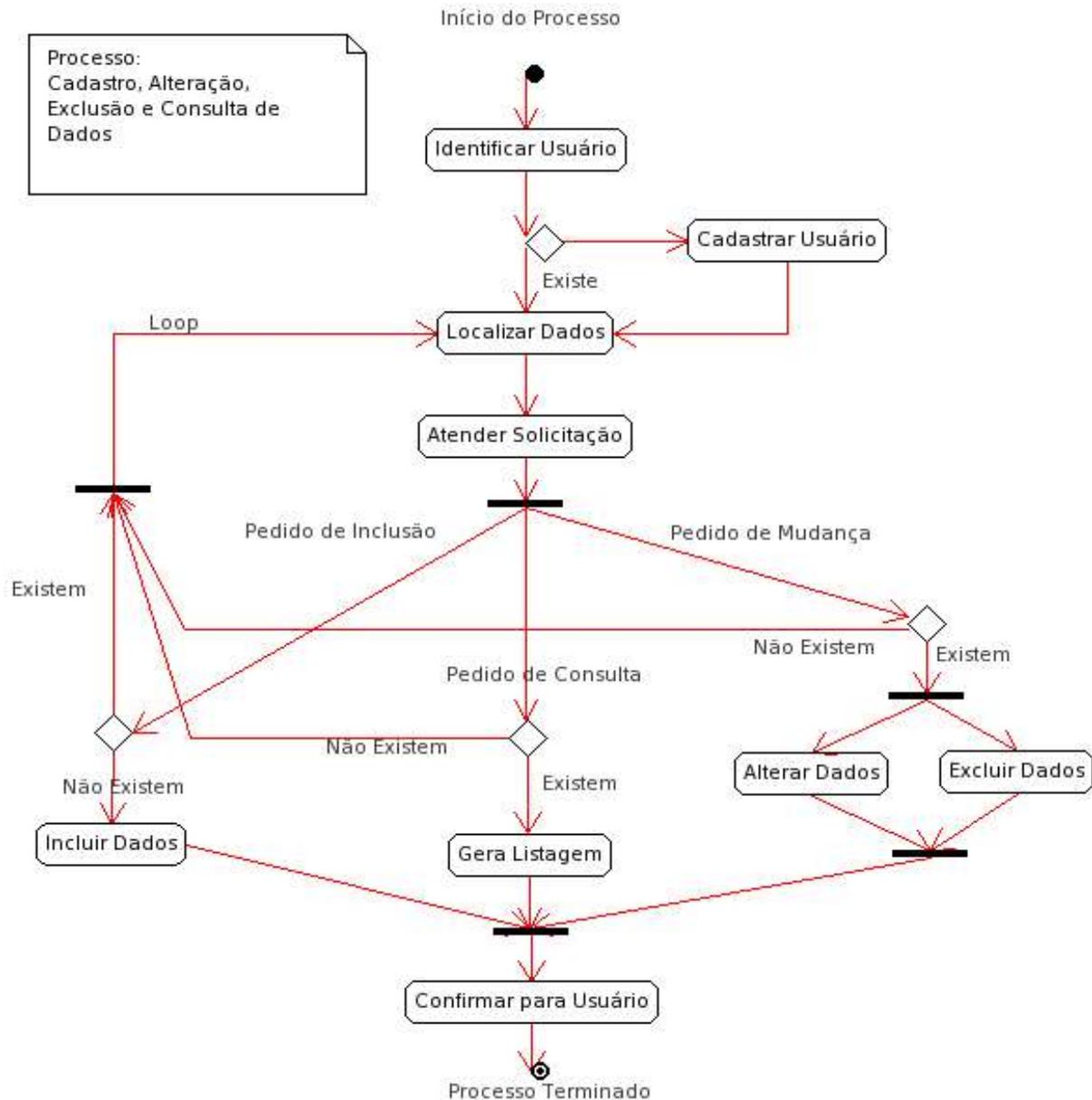


FIG.4.2: Diagrama de Atividade

A primeira atividade neste diagrama é identificar o usuário que pode ser o pesquisador ou o gerente do sistema, que precisam estar cadastrados no sistema de banco de dados. Uma vez identificados pelo sistema, o pesquisador pode solicitar consultas, inclusões, alterações e exclusões de dados sobre terminais e veículos, que serão atendidas pelo gerente do sistema caso atendam os requisitos do mesmo.

Uma vez atendidas as solicitações, o sistema confirma a inclusão ou a mudança de dados, e fornece uma listagem de consultas, terminando o processo.

4.4 DIAGRAMA DE CLASSE

A próxima etapa no desenvolvimento do modelo foi a criação do Diagrama de Classe. Para atingir esse propósito, primeiramente identificaram-se os Conceitos, isto é, as ideias ou coisas, e com esses conceitos criaram-se os Diagramas de Classe contendo os Atributos, Métodos e Associações.

A) IDENTIFICAÇÃO DOS CONCEITOS (IDÉIAS OU COISAS)

Neste trabalho, identificam-se os conceitos usando-se a técnica de JEPSON *et al.* (2000): Identificar o substantivo da frase, eliminando todos os conceitos que são atributos de outros conceitos.

A lista de conceitos gerados a partir do diagrama de caso de usos está definida da seguinte forma:

1. Ator 1: Pesquisador > Solicitação de Cadastro, Geração de Requisição de Mudança, Conexão e Desconexão com o Sistema, Localização de Dados de Terminais e de Veículos e Localização de Sistema de Informação Geográfica
 2. Ator 2: Gerente > Conexão e Desconexão com o Sistema, Localização de Requisição de Mudança e Aprovação, Localização de Dados de Terminais e de Veículos e respectivos cadastramento, alteração e exclusão de dados
 3. Camada de Terminais
 4. Camada de Veículos > Veículo Aquaviário, Ferroviário e Rodoviário
 5. Camada de Informações Geográficas > Aquavias, Rodovias e Ferrovias
- É importante observar que não existe um modelo conceitual que seja considerado

como o modelo certo. A idéia é refinar o modelo conceitual enquanto se trabalha na análise e projeto, chegando bem perto dos requisitos do pesquisador.

B) CRIAÇÃO DO DIAGRAMA DE CLASSE CONTENDO ATRIBUTOS, MÉTODOS E ASSOCIAÇÕES

Com a lista de conceitos, usa-se o Modelador Umbrello UML para desenhar o Diagrama de Classe do SITMMC (FIG.4.3), onde as associações são mostradas como linhas entre os conceitos.

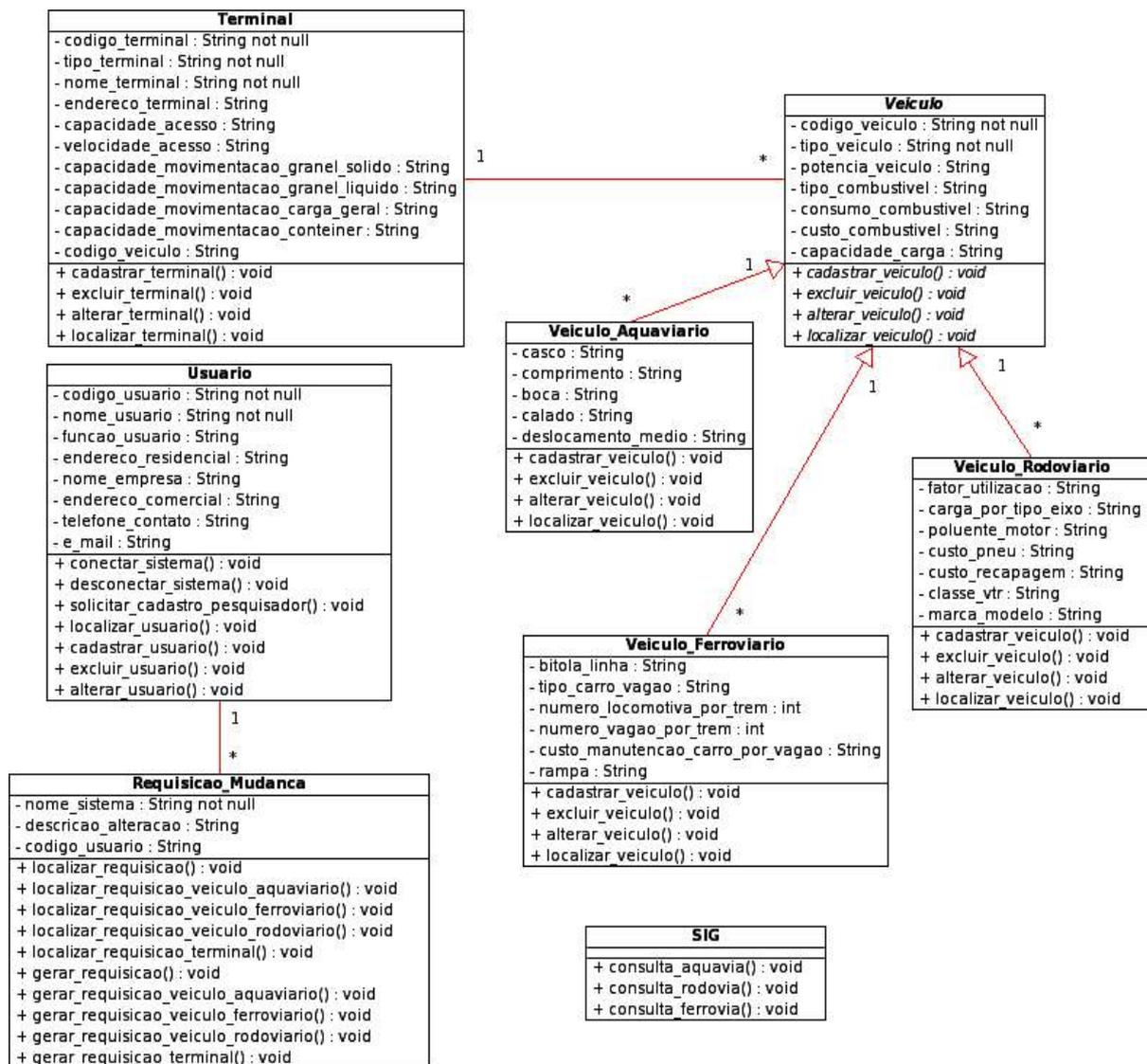


FIG.4.3: Diagrama de Classe

Neste diagrama, de acordo com o item 3.5.1-C-5 (Ferramenta Para Diagramação UML – Associações de Classe), as classes derivadas Veículo_Aquaviário, Veículo_Ferrovário e Veículo_Rodoviário herdam todos os atributos e operações da classe base (ou classe abstrata) Veículo, podendo sobrescrever ou modificar alguns deles, bem como adicionar mais atributos e operações próprias. As linhas com setas no lado da classe base Veículo representam uma associação de generalização.

Quanto a multiplicidade (ou cardinalidade), explicada em 3.4.1 (Modelo Conceitual de Banco de Dados – A Cardinalidade de Entidade em Relacionamento), as classes Veículo_Aquaviário, Veículo_Ferrovário e Veículo_Rodoviário têm multiplicidade máxima 1 no relacionamento com a classe Veículo, enquanto que a classe Terminal tem multiplicidade * com a classe Veículo, isto é, uma ocorrência de cada modal aquaviário, ferroviário ou rodoviário pode estar associada a no máximo 1 ocorrência de Veículo, e uma ocorrência de terminal está associada a no máximo * ocorrência de Veículo, enquanto que a entidade Veículo tem cardinalidade máxima * no relacionamento com os três modais e 1 com terminal, isto é, uma ocorrência de Veículo pode estar associada à no máximo * ocorrências de cada modal e uma ocorrência de terminal.

A classe Requisição_Mudança tem multiplicidade máxima 1 no relacionamento com a classe Usuário, isto é, uma ocorrência de Requisição_Mudança pode estar associada à no máximo 1 ocorrência de Usuário, enquanto que uma ocorrência de Usuário pode estar associada à no máximo * ocorrências de Requisição_Mudança.

Dentro dos retângulos que representam a classe, foram inseridos os nomes das classes, os atributos e as operações das classes (ou métodos). Os atributos são os dados pertencentes ao conceito, que podem ser pensados como Tipos de Dados (número, data, string, caractere, etc.), mas não diz como as classes vão ser implementadas.

A classe SIG – Sistema de Informação Geográfica não contém atributos, somente operações e na implementação ela desaparece.

4.5 PROJETO CONCEITUAL

Partindo dos conceitos apresentados no item 3.4.1 desta dissertação, criou-se o Modelo Conceitual, que é representado por um Diagrama Entidade-Relacionamento - DER (FIG.4.4).

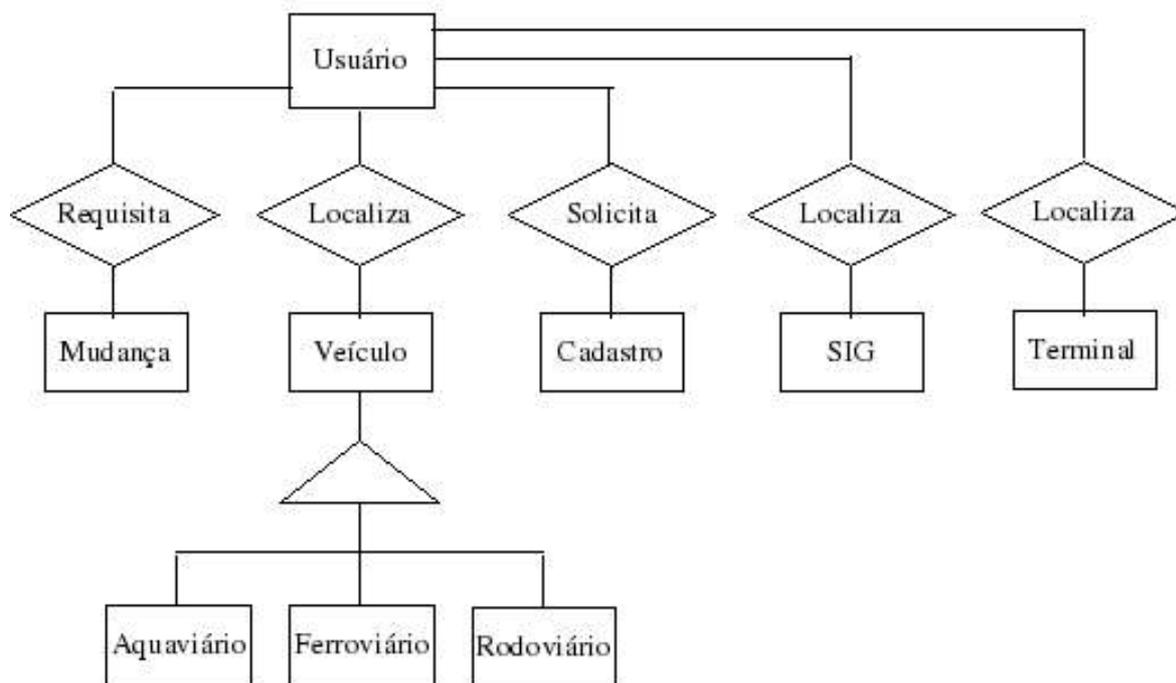


FIG.4.4: Diagrama E-R

Trata-se de um modelo de dados abstrato que descreve a estrutura do SITMMC de forma independente do PostgreSQL, o SGBD usado neste trabalho.

O modelo conceitual apresentado nesse trabalho contém informações sobre a entidade Usuário (Pesquisador ou Gerente), e as entidades terminais, veículos, informações geográficas e os respectivos cadastros e mudanças relacionados entre si. Essas entidades são objetos sobre os quais se desejam manter informações no sistema. Os relacionamentos entre as entidades são representados através de losangos ligados por linhas aos retângulos representativos das entidades.

Para a entidade Usuário, o banco armazena o código do usuário, nome do usuário, função do usuário, etc. Para a entidade Terminal, o banco armazena o código do terminal, o tipo de terminal, o nome do terminal, etc. Para se conhecer com detalhe o que o banco armazena para cada entidade basta examinar o Esquema Físico contido no item 4.6.1-A desta dissertação. Normalmente não se representa no DER os dados armazenados pelo banco, para não ficar confusa a visualização do mesmo. Por isso os dados não são mostrados no DER do modelo desenvolvido.

4.6 PROTOTIPAÇÃO

Com a modelagem do sistema de banco de dados terminada, inicia-se a fase de implementação e avaliação do mesmo através de um protótipo, indicando-se as alterações caso sejam necessárias.

Os recursos computacionais necessários para o desenvolvimento do protótipo devem ser instalados no computador a partir de uma ferramenta do Linux chamada Synaptic, que gerencia os pacotes do sistema operacional. Caso não estejam incluídos devem ser baixados da Internet. O Linux utilizado para desenvolver este protótipo é o Conectiva 10 e os recursos computacionais instalados no computador são os listados abaixo com os seus respectivos procedimentos de instalação:.

- Compilador Java jdk-1_5_0_09:

Site Principal: http://mirror.dcc.fc.up.pt/Java/jdk-1_5_0_09-linux-i586-rpm.bin/

```
[root@tech ]# cd /usr/local/jdk
[root@tech jdk]# chmod +x jdk-1_5_0_09-linux-i586-rpm.bin
[root@tech jdk]# ./jdk-1_5_0_09-linux-i586-rpm.bin
[root@tech jdk]# rpm -ihv jdk-1_5_0_09-linux-i586-rpm
[root@tech jdk]# updatedb
[root@tech jdk]# PATH=$PATH:/usr/java/jdk1_5_0_09/bin
```

- PostgreSQL 8.1.4

Site Principal: <http://mirrors.isc.org/pub/postgresql/v8.1.4/>

```
[root@tech ]# cd /usr/local/postgresql
[root@tech postgresql]# tar xvjf postgresql-8.1.4.tar.bz2
[root@tech postgresql]# cd postgresql-8.1.4
[root@tech postgresql-8.1.4]# ./configure
[root@tech postgresql-8.1.4]# gmake
[root@tech postgresql-8.1.4]# gmake install
[root@tech postgresql-8.1.4]# adduser postgres
[root@tech postgresql-8.1.4]# mkdir /usr/local/pgsql/data
[root@tech postgresql-8.1.4]# chown postgres /usr/local/pgsql/data/
[root@tech postgresql-8.1.4]# su postgres
```

```
[postgres@tech postgresql-8.1.4]$ /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data/
(cria o servidor de banco de dados)
```

```
[postgres@tech postgresql-8.1.4]$ cd (entra no diretório /home/postgres)
```

-O postmaster inicializa o servidor de banco de dados. A opção -D especifica o diretório **data** como raiz da árvore de diretórios, redirecionando para um arquivo de nome **logfile**, sendo inicializado na port padrão 5432. Para verificar o processo, digite o comando:

```
$ ps -e (verifica se o processo do postmaster está em execução)
```

```
[postgres@tech ~]$ /usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data > logfile
2>&1 &
```

```
[postgres@tech ~]$ /usr/local/pgsql/bin/createdb teste
```

```
CREATE DATABASE
```

```
[postgres@tech ~]$
```

```
[postgres@tech ~]$ /usr/local/pgsql/bin/psql teste
```

```
Welcome to psql 8.0.3, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
```

```
\h for help with SQL commands
```

```
\? for help with psql commands
```

```
\g or terminate with semicolon to execute query
```

```
\q to quit
```

```
teste=#
```

- Driver JDBC postgresql-8.2

Site Principal: <http://jdbc.postgresql.org/>

```
[root@tech ]# cd /usr/local/jdbc
```

Copia o arquivo (.jar) para o endereço do jdk:

```
[root@tech jdbc]# cp postgresql-8.2dev-503.jdbc3.jar /usr/java/jdk1.5.0_09/jre/lib/
ext/
```

- Postgis 1.2.0

Site Principal: <http://www.postgis.org/download/> para /usr/local/postgresql-8.1.4/contrib/

```
[root@tech contrib]# tar xvzf postgis-1.2.0.tar.gz
```

```
[root@tech contrib]# cd postgis-1.2.0
```

```
[root@tech postgis-1.2.0]$ PATH=$PATH:/usr/local/pgsql/bin
```

```
[root@tech postgis-1.2.0]$ ./configure
```

```
[root@tech postgis-1.2.0]$ make
```

```
[root@tech postgis-1.2.0]# make install
```

```
[root@tech postgis-1.2.0]# su postgres
```

```
[postgres@tech postgis-1.2.0]$ /usr/local/pgsql/bin/createdb gis
```

```
CREATE DATABASE
```

```
[postgres@tech postgis-1.2.0]$ /usr/local/pgsql/bin/createlang plpgsql gis
```

-lwpostgis.sql cria funções topológicas que serão usadas com o banco de dados gis

```
[postgres@tech postgis-1.2.0]$ /usr/local/pgsql/bin/psql -d gis -f lwpostgis.sql
```

-spatial_ref_sys.sql cria funções espaciais que serão usadas com o banco de dados gis

```
[postgres@tech postgis-1.0.1]$ /usr/local/pgsql/bin/psql -d gis -f spatial_ref_sys.sql
```

- MapServer 4.8.4, e outros aplicativos dependentes

Para instalar o MapServer é necessário verificar se os arquivos libjpeg, curl, gd e libtiff estão instalados no Linux usando o comando **\$ rpm -q <arquivo>**. Senão, abrir o gerenciador de pacotes Synaptic para instala-los, ou, baixar da Internet.

```
[root@tech mapserver]# rpm -q libjpeg lib
```

```
resposta: jpeg-6b-33
```

```
[root@tech mapserver]# rpm -q curl
```

```
resposta: curl-7.12.3-2
```

```
[root@tech mapserver]# rpm -q gd
```

```
resposta: gd-2.0.28-1.30.1
```

```
[root@tech mapserver]# rpm -q libtiff
```

```
resposta: libtiff-3.6.1-9.fc3
```

Em seguida, instala-se os arquivos proj4, libgeotiff, geos e gdal:

Proj4 é uma biblioteca muito utilizada nos sistemas livres (e de código aberto) para tratamento de projeções. Ela é a responsável, por exemplo, pelo recurso de mudança de projeção em tempo real presente no MapServer. Com capacidade de transformações entre diferentes elipsóides e datums, esta biblioteca é muito poderosa, tendo implementado complexos algoritmos matemáticos.

Site Principal: <http://www.remotesensing.org/proj>

```
[root@tech proj4]# unzip proj-4.4.9.zip
```

```
[root@tech proj-4.4.9]# ./configure
```

```
[root@tech proj-4.4.9]# make
```

```
[root@tech proj-4.4.9]# make install
```

Site Principal: <http://www.remotesensing.org/geotiff>

```
[root@tech geotiff]# tar xvzf libgeotiff-1.2.2.tar.gz
```

```
[root@tech geotiff]# cd libgeotiff-1.2.2
```

```
[root@tech libgeotiff-1.2.2]# ./configure --with-libtiff --with-proj=/usr/local
```

```
[root@tech libgeotiff-1.2.2]# make
```

```
[root@tech libgeotiff-1.2.2]# make install
```

GEOS (Geometry Engine, Open Source).

Esta biblioteca é uma “tradução” da JTS de Java para C++. Este projeto de tradução surgiu para atender uma demanda existente no código do PostGIS, pois este não contempla a especificação SFS em 100%. A “criação” da GEOS tornou possível a total compatibilidade do PostGIS com a SFS, pois agora é possível compilar o PostGIS incluindo o código da GEOS. Este é um exemplo de interação entre empresas dentro da filosofia do SL.

Site Principal: <http://geos.refrations.net>

```
[root@tech geos]# tar xvjf geos-2.1.2.tar.bz2
```

```
[root@tech geos]# cd geos-2.1.2
```

```
[root@tech geos-2.1.2]# ./configure
```

```
[root@tech geos-2.1.2]# make
```

```
[root@tech geos-2.1.2]# make install
```

GDAL (Geospatial Data Abstraction Library) / OGR Simple Feature Library.

Esta é a biblioteca de código aberto mais poderosa no quesito visualização / conversão de formatos matriciais e vetoriais. Ela é amplamente utilizada não somente nos projetos livres, mas também nos sistemas proprietários. A biblioteca GDAL trata especificamente dos formatos matriciais (raster), mas internamente ela possui uma biblioteca denominada OGR que manipula os formatos vetoriais (vector).

Site Principal: <http://www.remotesensing.org/gdal/>

```
[root@tech mapserver]# tar xvzf gdal-1.2.5.tar.gz
```

```
[root@tech mapserver]# cd gdal-1.2.5
```

```
[root@tech gdal-1.2.5]# ./configure --with-libtiff=/usr/local --with-pg=/usr/local/pgsql/bin/pg_config --with-geotiff=/usr/local --with-geos=yes --with-static-proj4=/usr/local
```

```
[root@tech gdal-1.2.5]# make
```

```
[root@tech gdal-1.2.5]# make install
```

MapServer:

Site Principal: <http://mapserver.gis.umn.edu/>

```
[root@tech mapserver]# tar xvzf mapserver-4.6.0.tar.gz
```

```
[root@tech mapserver]# cd mapserver-4.6.0
```

```
[root@tech mapserver-4.6.0]# ./configure --disable-ignore-missing-data --includedir=/usr/local/include --with-curl-config=/usr/local/bin --with-proj --with-gdal=/usr/local/bin/gdal-config --with-jpeg --with-gd --with-freetype --with-zlib --with-png --with-tiff --with-ogr --with-postgis --with-wfs --with-wmsclient --with-wfsclient--with-curl-config --with-gif --with-threads --with-postgis=/usr/local/pgsql/bin/pg_config --with-mygis
```

```
root@tech mapserver-4.6.0]# make
```

```
root@tech mapserver-4.6.0]# make install
```

Acrescentar a linha /usr/local/lib no arquivo /etc/ld.so.conf:

```
[root@tech mapserver-4.6.0]# vi /etc/ld.so.conf
```

```
[root@tech mapserver-4.6.0]# ldconfig
```

```
[root@tech mapserver-4.6.0]# ./mapserv -v
resposta: MapServer version 4.6.0 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG
OUTPUT=WBMP OUTPUT=SVG SUPPORTS=PROJ SUPPORTS=FREETYPE
SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT
SUPPORTS=WFS_SERVER SUPPORTS=WFS_CLIENT SUPPORTS=THREADS
INPUT=TIFF INPUT=EPPL7 INPUT=POSTGIS INPUT=OGR INPUT=GDAL
INPUT=SHAPEFILE
```

```
[root@tech mapserver-4.6.0]# cd mapscript
[root@tech mapscript]# cd python
[root@tech python]# python setup.py build
[root@tech python]# swig -python -shadow -modern -o mapscript_wrap.c ../
mapscript.i
[root@tech python]# cd tests/cases
[root@tech cases]# python runalltests.py -v
```

- Netbeans 5_5-beta2

Download do arquivo netbeans-5_5-beta2-linux.bin a partir do site:

<http://www.netbeans.org> para /usr/local/netbeans

```
[root@tech ]# cd /usr/local/netbeans
[root@tech netbeans]# ./netbeans-5_5-beta2-linux.bin
```

- Jump 1.1.2

Download do arquivo jump-1.1.2.zip a partir do site:

<http://www.jump-project.org/project.php?PID=JUMP&SID=DOWN> para /usr/local/jump

```
[root@tech ]# cd /usr/local/jump
[root@tech jump]# unzip jump-1.1.2.zip
[root@tech jump]# cd bin
[root@tech bin]# chmod +x JUMP*
[root@tech bin]# ./JUMPWokbench-unix-2.sh
```

- Apache 2.0.49

Todos os pacotes, com exceção do Netbeans e do Apache, foram instalados nos seus respectivos diretórios a partir do **/usr/local**.

O Netbeans foi instalado no **/opt**, e o servidor HTTPD Apache 2.0.49 já faz parte do sistema operacional instalado .

Quanto ao servidor web necessário para executar as páginas JSP (APÊNDICE 4), o Apache-Tomcat-5.5.17, está embutido na aplicação Netbeans, do diretório **/opt/netbeans-5.5-beta2/enterprise3/apache-tomcat-5.5.17**. Portanto não precisa ser baixado da Internet.

4.6.1 METODOLOGIA

A metodologia aplicada ao desenvolvimento do protótipo resume-se a:

- Gerar as estruturas de dados;
- Criar as Telas do protótipo.

A) GERAÇÃO DAS ESTRUTURAS DE DADOS

Parte das estruturas de dados neste trabalho foi gerada a partir do Diagrama de Classe (FIG.4.3) mostrado em 4.4-B. A outra parte foi gerada a partir da base de dados fornecida pelo IME.

Para a primeira parte, usando o Modelador UML Umbrello, foram selecionados os botões SQL da linguagem ativa e o assistente de geração de código, ambos pressionados a partir do botão CÓDIGO da barra de ferramenta do modelador. Com a janela do assistente aberta, foram selecionadas as classes Gerente, Requisicao_Mudanca, Terminal, Usuario, Veiculo, Veiculo_Aquaviario, Veiculo_Ferroviano e Veiculo_Rodoviario.

Este procedimento gerou, automaticamente, todos os códigos SQL, que foram salvos no desktop do diretório principal.

Posteriormente, na linha de comando do Linux foram editados os códigos gerados em SQL e feitas algumas alterações compatíveis com o PostgreSQL, conforme mostrado abaixo:

```

CREATE TABLE Terminal (
    codigo_terminal serial ,
    tipo_terminal varchar(12) ,
    nome_terminal varchar(20) ,
    endereco_terminal varchar(50) ,
    capacidade_aceso varchar(10) ,
    velocidade_aceso varchar(10) ,
    capacidade_movimentacao_granel_solido varchar(10) ,
    capacidade_movimentacao_granel_liquido varchar(10) ,
    capacidade_movimentacao_carga_geral varchar(10) ,
    capacidade_movimentacao_container varchar(10) ,
    CONSTRAINT pk_terminal PRIMARY KEY(codigo_terminal)
);

```

```

CREATE TABLE Usuario (
    codigo_usuario serial ,
    nome_usuario varchar(30) ,
    funcao_usuario varchar(15) ,
    endereco_residencial varchar(50) ,
    nome_empresa varchar(30) ,
    endereco_comercial varchar(50) ,
    telefone_contato varchar(20) ,
    e_mail varchar(30) ,
    situacao_usuario varchar(8) ,
    login varchar(10) ,
    senha varchar (10) ,
    CONSTRAINT pk_usuario PRIMARY KEY(codigo_usuario)
);

```

```

CREATE TABLE Requisicao_Mudanca (
    codigo_requisicao_mudanca serial,
    codigo int,
    nome_sistema char(25) ,
    descricao_alteracao varchar(255) ,
    codigo_usuario int ,
    nome_usuario char(50),
    CONSTRAINT pk_mudanca PRIMARY KEY(codigo_requisicao_mudanca) ,
    CONSTRAINT fk_usuario FOREIGN KEY(codigo_usuario) REFERENCES usuario
(codigo_usuario)
);

```

```

CREATE TABLE Veiculo_Aquaviario (
    codigo_veiculo serial,
    tipo_veiculo varchar(10) ,
    potencia_veiculo varchar(10) ,
    tipo_combustivel varchar(10) ,
    consumo_combustivel varchar(10) ,

```

```

    custo_combustivel varchar(10) ,
    capacidade_carga varchar(10) ,
    codigo_terminal char(10) ,
    casco varchar(5) ,
    comprimento varchar(5) ,
    boca varchar(5) ,
    calado varchar(5) ,
    deslocamento_medio varchar(10),
    CONSTRAINT pk_veiculo_aquaviario PRIMARY KEY(codigo_veiculo)
);

```

```

CREATE TABLE Veiculo_Ferrovuario (
    codigo_veiculo serial,
    tipo_veiculo varchar(10) ,
    potencia_veiculo varchar(10) ,
    tipo_combustivel varchar(10) ,
    consumo_combustivel varchar(10) ,
    custo_combustivel varchar(10) ,
    capacidade_carga varchar(10) ,
    codigo_terminal char(10) ,
    bitola_linha varchar(5) ,
    tipo_carro_vagao varchar(10) ,
    numero_locomotiva_por_trem int ,
    numero_vagao_por_trem int ,
    custo_manutencao_carro_por_vagao varchar(5) ,
    rampa varchar(5) ,
    CONSTRAINT pk_veiculo_ferrovuario PRIMARY KEY(codigo_veiculo)
);

```

```

CREATE TABLE Veiculo_Rodoviario (
    codigo_veiculo serial,
    tipo_veiculo varchar(10) ,
    potencia_veiculo varchar(10) ,
    tipo_combustivel varchar(10) ,
    consumo_combustivel varchar(10) ,
    custo_combustivel varchar(10) ,
    capacidade_carga varchar(10) ,
    codigo_terminal char(10) ,
    fator_utilizacao varchar(5) ,
    carga_por_tipo_eixo varchar(10) ,
    poluente_motor varchar(10) ,
    custo_pneu varchar(5) ,
    custo_recapagem varchar(5) ,
    classe_vtr varchar(5) ,
    marca_modelo varchar(10) ,
    CONSTRAINT pk_veiculo_rodoviario PRIMARY KEY(codigo_veiculo)
);

```

Para inserir estes códigos no PostgreSQL, digitou-se o seguinte:

```
[postgres@tech ~]$ /usr/local/pgsql/bin/psql -d sitmmc -f arquivo_gerado.sql
```

onde: psql é o terminal interativo do PostgreSQL

-d especifica o nome do banco de dados que deve ser conectado, neste caso o SITMMC, que foi criado como indicado no item 4.6, na instalação do PostgreSQL.

-f executa o comando contido no arquivo (.sql), aqui exemplificado como arquivo_gerado.sql

Tech é o nome do localhost do sistema operacional Linux instalado.

Convém lembrar que este comando foi executado pelo usuário “postgres” após ter sido iniciado o servidor de banco de dados com o comando descrito abaixo:

```
[postgres@tech ~]$ /usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data > logfile 2>&1 &
```

Para a segunda parte, referente a geração da estrutura de dados a partir da base de dados fornecida pelo IME, que contém arquivos *shapefile* (.shp), foram feitos os seguintes procedimentos:

1. Utilizou-se o Programa shp2pgsql que é responsável pela conversão dos arquivos *shapefile* para arquivos com instruções em sql.:

```
[postgres@tech ~]$ /usr/local/pgsql/bin/shp2pgsql -s 4326 arquivo_shapefile nome_tabela >
arquivo.sql
```

onde: -s especifica o código epsg (4326) que a tabela possuirá, utilizado para análises topológicas e espaciais.

- arquivo_shapefile nome do arquivo *shape file* a ser lido. Deve-se colocar também o caminho onde se encontram os arquivos *shape file*;

- nome_tabela nome da tabela a ser gravada no banco de dados;

- arquivo.sql nome do arquivo onde conterá todos os comandos de criação da tabela e de inserção dos registros na tabela criada.

2. Repetir a etapa anterior para todas os demais arquivos *shape file* (*.shp), para criar as tabelas .sql a serem carregadas no banco de dados.
3. Para importar os arquivos criados anteriormente (*.sql) para o banco de dados SIG, devemos proceder de acordo com o exemplo abaixo (um para cada arquivo sql):

[postgres@tech ~]\$ /usr/local/pgsql/bin/psql -d gis -f nome_arquivo.sql

4. Se o usuário desejar consultar no banco de dados todas as importações realizadas, executar o comando [postgres@tech ~]\$ /usr/local/pgsql/bin/psql -l

B) CRIAÇÃO DAS TELAS DO PROTÓTIPO

As telas do protótipo proposto neste trabalho estão detalhadas no APÊNDICE 3. Elas foram criadas a partir do esquema de estrutura funcional (FIG.4.5) baseado na modelagem criada nesta dissertação e permitem ao usuário manipular os dados eficazmente.

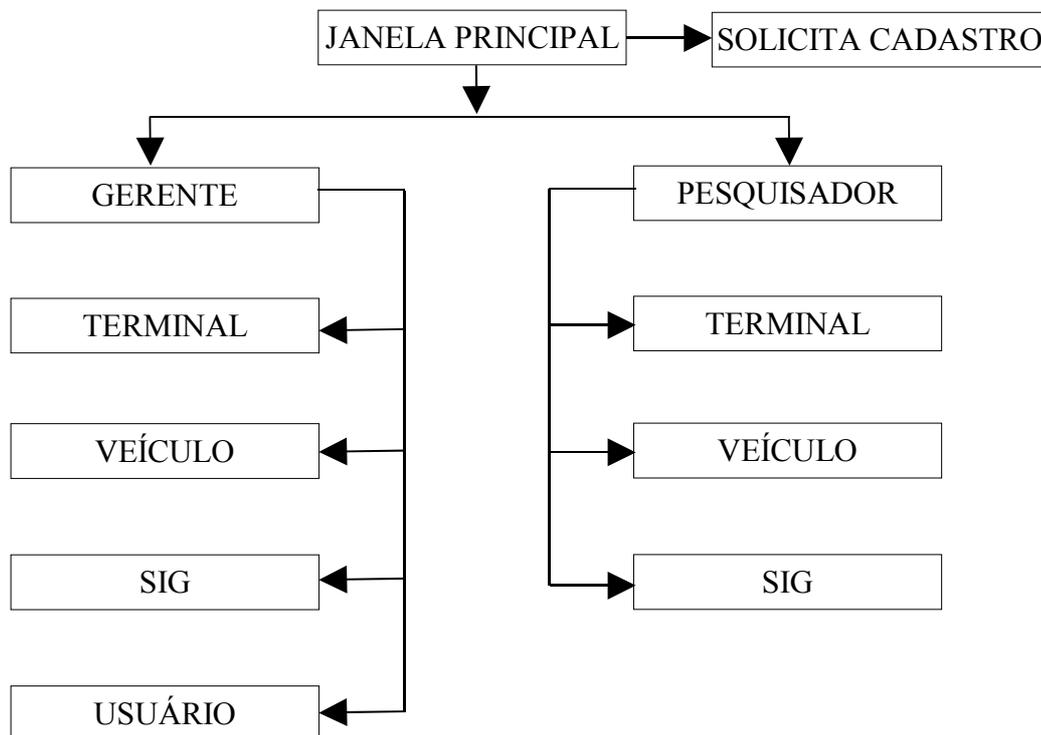


FIG.4.5: Esquema de Estrutura Funcional

A janela principal (FIG.4.6) do protótipo contém duas caixas de texto, uma para entrar o *login* e outra para a *senha*, além de um *link* para solicitar o cadastro.

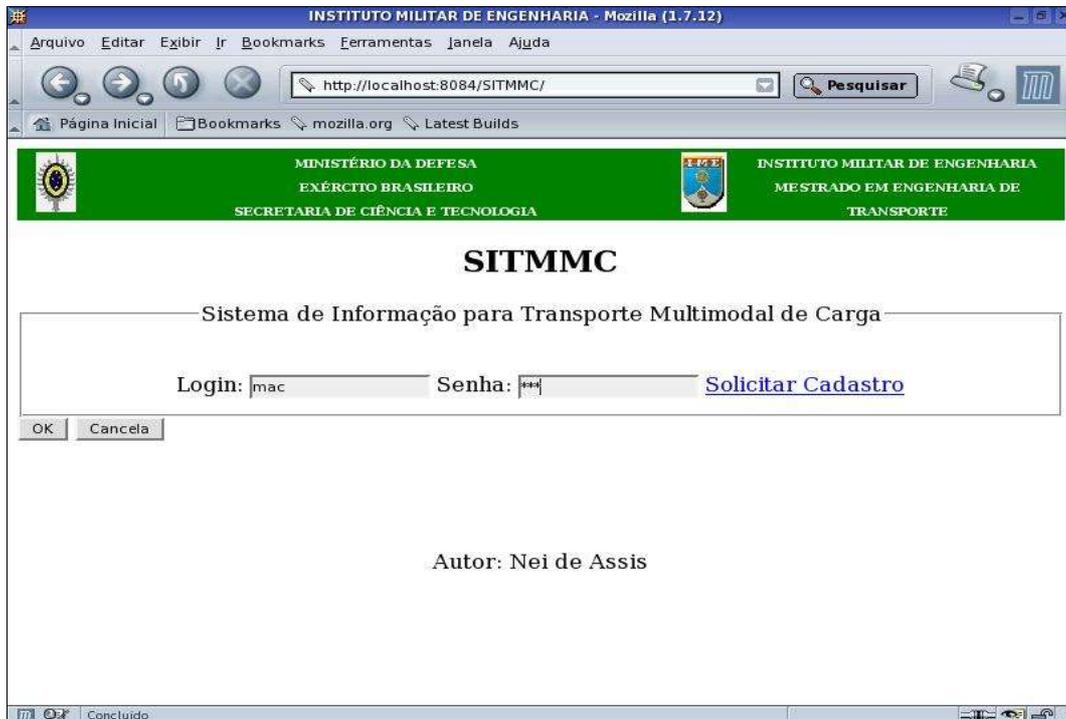


FIG.4.6: Janela Principal do Protótipo

De acordo com o Diagrama de Atividade (FIG.4.2) apresentado no item 4.3 deste trabalho, a primeira atividade do sistema é identificar o usuário, que pode ser o pesquisador ou o gerente do sistema.

Se o usuário não for cadastrado, ele seleciona o *link* que abrirá uma janela de cadastramento de usuário (FIG.4.7).

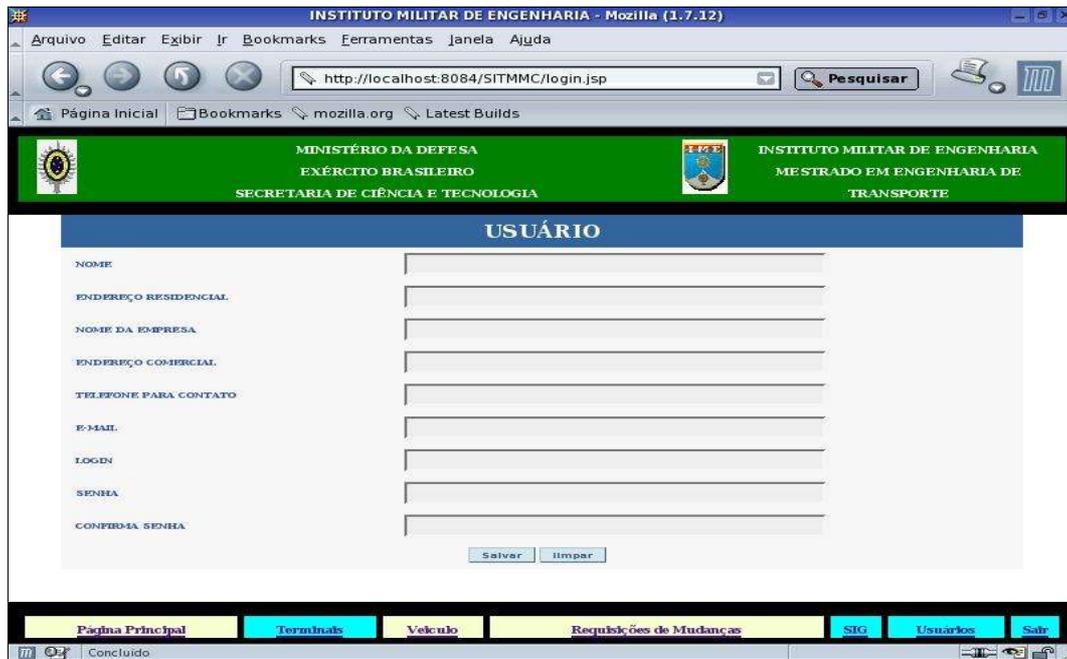


FIG.4.7: Solicitação de Cadastro

Uma vez identificado pelo sistema, o usuário abrirá a tela principal do gerente (FIG.4.8) ou a do pesquisador (FIG.4.9) e poderá a partir daí acessar os dados, selecionando os botões Terminal, Veículo, Requisição de Mudança, ou Usuário, que mostrará outras telas que foram criadas de acordo com a Descrição de Casos de Uso (ver APÊNDICE 1).



FIG.4.8: Janela do Gerente do Sistema



FIG.4.9: Janela do Pesquisador do Sistema

Por outro lado, quando o usuário selecionar o botão SIG, o sistema mostrará a tela C.14, que contém três janelas: uma janela que mostra mapas geográficos, outra que consulta os dados importados para o banco de dados GIS e outra que detalha o pedaço do mapa que está sendo consultado.

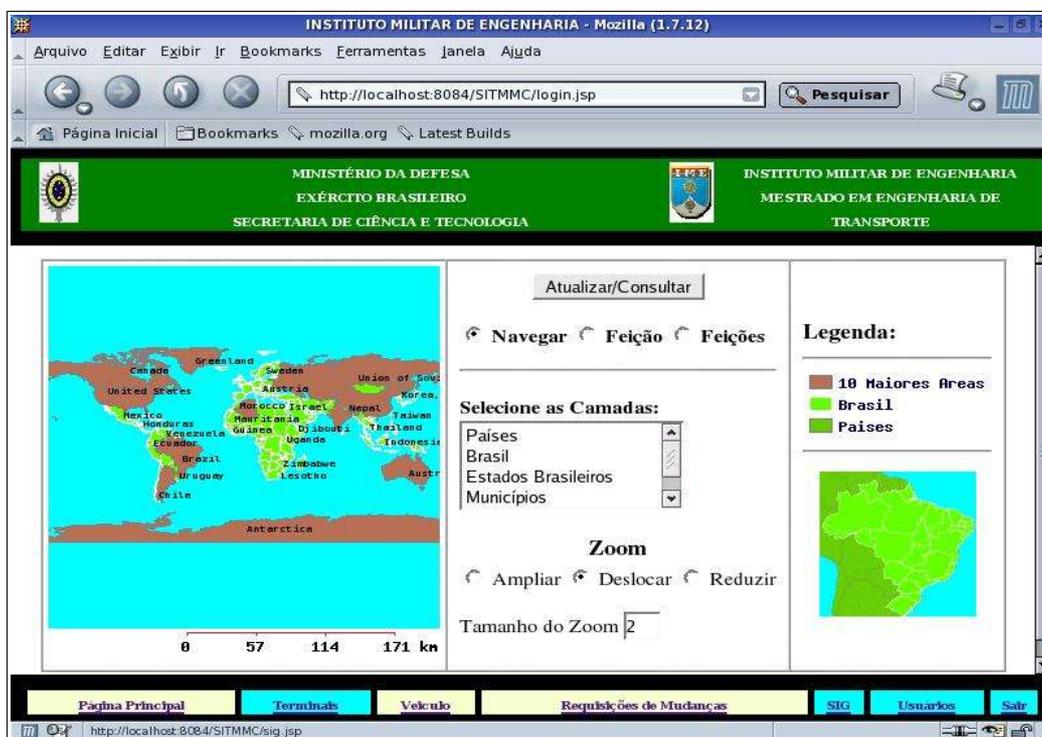


FIG.4.10: Janela de Localização das 10 Maiores Áreas da Terra

4.6.2 IMPLEMENTAÇÃO DO PROTÓTIPO

Com os recursos computacionais instalados e as etapas de geração de dados e a criação de telas terminada, inicia-se a fase de implementação do protótipo.

O Sistema de Informação de Transporte Multimodal de Carga – SITMMC foi criado na plataforma de desenvolvimento Java J2EE em duas camadas lógicas: Apresentação e Persistência.

Na camada de Apresentação foram incluídos os arquivos contendo os códigos JSP e Bean, respectivamente com as extensões (.jsp) e (.java).

O JSP (Java Server Pages) é uma tecnologia Web J2EE.

O Bean especifica o Java para a construção de objetos que serão lidos pelo servidor de

Web Jakarta-Tomcat.

A camada de Apresentação faz uma chamada à camada de Persistência que é a camada que acessa os dados do banco de dados.

Além dessas duas camadas, foi criado o *Value Object* (VO), que é o Padrão de Desenvolvimento (*Design Pattern*) que faz a comunicação entre as camadas de Apresentação e de Persistência.

Uma explicação mais detalhada sobre JSP, J2EE e camadas lógicas se encontram no APÊNDICE 4.

Não é o propósito desta dissertação o aprofundamento no conhecimento da linguagem Java. Entretanto, no APÊNDICE 4, foi apresentado o código Java escrito nas camadas de apresentação, de persistência e no *value object*, apenas para o módulo Terminal.

Quanto a parte referente ao GIS, a ferramenta usada para manipular os dados foi o MapServer. Conforme explicado no item 3.3.4, o MapServer permite o desenvolvimento de servidores de mapas, sendo o carro-chefe das aplicações na área de Geotecnologia, e fornece um CGI, o **mapserv**, com funcionalidade para o desenvolvimento de aplicações de SIG em ambiente WEB. O procedimento detalhado da instalação do MapServer está no item 4.6.

Neste trabalho, o arquivo **mapserv** foi copiado para o diretório **/srv/www/default/cgi-bin/**, que será lido pelo comando `<form method=GET action="mapserv">` escrito no arquivo **sig.html**, que está localizado no diretório **gis**.

O diretório **gis** foi criado no endereço do servidor httpd Apache 2.0.49, **/srv/www/default/html** que é o **http://localhost/** lido na janela do browser. Convém lembrar que o servidor Apache é iniciado no Contectiva 10 pelo comando **/sbin/service httpd start**.

No **/srv/www/default/html/gis** foram criados os arquivos **gis.html**, **gis.map** e **index.html** que permitem acessar os dados referentes aos mapas geográficos. Assim como foi dito para a linguagem Java, também não é o propósito desta dissertação o aprofundamento no conhecimento da linguagem do MapFile, mas, no APÊNDICE 5, são apresentados os códigos dos arquivos criados no diretório **.../gis**, para que se tenha uma visão do que vem a ser o sistema.

As tabelas geradas no PostgreSQL usando a metodologia apresentada no item 5.2 para o GIS foram as seguintes: estados, ferrovias, municípios, países, rios e rodovias. Estas tabelas foram geradas a partir dos *shapefiles* da base de dados fornecida pelo IME.

5 CONCLUSÕES

As conclusões finais desta dissertação têm como objetivo demonstrar e criticar os resultados alcançados pela pesquisa proposta no Capítulo 1 deste trabalho, fazendo sugestões futuras sobre a mesma.

Este trabalho apresentou, entre outros tópicos, o estado-da-arte da tecnologia da informação no desenvolvimento de um modelo UML e na criação de um protótipo para implementar este modelo, que foi chamado de SITMMC (Sistema de Informação para o Transporte Multimodal de Carga), com total aplicabilidade na Web.

O principal mérito desta pesquisa foi o fato de que o desenvolvimento do SITMMC agregou um valor qualitativo aos estudos existentes na área de Sistema de Informação de Transporte, pois ampliou os horizontes dos mesmos, tornando-os compatíveis com o que há de mais atualizado na Tecnologia da Informação.

O modelo criado teve como principal meta ser totalmente compatível com o desenvolvimento de um POO (Projeto Orientado a Objeto), o que possibilitou que ele pudesse ter sido criado numa plataforma de desenvolvimento Java J2EE, proporcionando um alto desempenho final para aplicações Web.

Outra meta atingida foi criar o protótipo em um Ambiente Livre, usando *Softwares Livres*, de Código Aberto, tais como: Conectiva 10, PostgreSQL, Driver JDBC, Postgis e MapServer. Os servidores de Web foram o Apache-Tomcat e o HTTPD Apache.

Portanto, esta pesquisa acadêmica desenvolvida para um Sistema de Informação de Transporte, disponibilizou uma grande quantidade de conhecimento que certamente poderá colaborar para o desenvolvimento de novos trabalhos que serão desenvolvidos por estudantes ou pesquisadores envolvidos na problemática do Sistema de Informação de Transporte, tanto em Ambiente Livre como em outros sistemas operacionais.

A principal crítica e sugestão para trabalhos futuros é que, caso haja interesse em ampliar este estudo, faça-se um pesquisa sobre como inserir o código Java do Jump ou do OpenJump no código do projeto do protótipo SITMMC para a leitura dos objetos GIS armazenados em banco de dados pelo PostGIS, no lugar do MapServer. O principal motivo desta sugestão é que embora o MapServer seja um estado-da-arte nas soluções de SIG (Apresentação de mapas, imagens e dados vetoriais) na Web, suporte diversos formatos vetoriais (ESRI® Shapefiles, PostGIS, etc.) e matriciais (TIFF/GeoTIFF, GIF, PNG, etc.),

ele é mais compatível com a linguagem C (nativa). Ele também pode ser acessado através de Python, Perl, PHP e Java. Entretanto, com relação às linguagens que permitem acesso ao API, o suporte à Java é deficiente.

Finalmente, o botão SIG, visto na barra inferior das telas do gerente e do pesquisador, quando pressionado, chama o arquivo `index.html`, que está localizado no diretório `/srv/www/default/html/gis`. Os comandos contidos neste arquivo darão início a leitura dos objetos GIS armazenados em banco de dados pelo PostGIS, após ter sido inicializado o servidor Apache.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- ANTT. **Concessões e Mapas**. Disponível em: www.antt.gov.br
- BALLOU, R.H.. **Logística empresarial**. São Paulo : Atlas, 1993.
- BASTOS, I.D. **Avaliação do Desempenho Logístico do Serviço de Transporte Rodoviário de Cargas – Um Estudo de Caso no Setor de Revestimentos Cerâmicos**. Dissertação de Mestrado. Florianópolis : Universidade Federal de Santa Catarina, 2003.
- BUSTAMANTE, J.C. **Terminais Multimodais de Carga**. Vitória, ES : UFES, 2001.
- CAIXETA-FILHO, J.V. e al. **Gestão Logística do Transporte de Cargas**. São Paulo : Atlas, 2001.
- CALKINS, H.W. E TOMLINSON, R.F. **Geographic Information Systems: Methods and Equipment for Land Use Planning**. Virginia, U.S.A. : IGUCGDSP : 1997.
- DEMARIA, M. **O Operador de Transporte Multimodal como fator de otimização da logística**. Dissertação de Mestrado. Santa Catarina : UFSC, 2004.
- DIOGO, F.J.A. **Sistema de Informação para os Transportes**. Dissertação de Mestrado. Rio de Janeiro : IME, 1994.
- FERREIRA FILHO, A.S. **Sistema de Informação para a Mobilização Militar dos Transportes – Modo Hidroviário**. Dissertação de Mestrado. Rio de Janeiro : IME, 1998.
- FIGUEIREDO NETO, E.S. **Roteirização de Veículos de Carga Utilizando a Experiência do Operador**. Dissertação de Mestrado. Rio de Janeiro : IME, 2005.
- FLEURY, P. **Intermodalidade: importância para a Logística e estágio atual no Brasil**. Disponível em: www.coppead.ufrj.br, 2003.
- GEIPOT. **Anuário Estatístico**. Disponível em: <http://www.geipot.org.br>.
- GÔMARA, A.R.B. **Brasil – 500 anos de Transporte**. Revista CLT, ano V, no 59, abril 2000.
- GUIA DE LOGÍSTICA. **Transporte Multimodal de Carga**. Disponível em: www.guiadelogistica.com.br
- GUIMARÃES, I.C.O. **Geração de base geográfica multimodal de transportes em sistemas de informação geográficos**. Rio de Janeiro : IME, 1999.
- HAY, W.W. **An Introduction to Transport Engineering**. 2. ed. New York, USA : John Wilwy & Sons Inc., 1977.

- HEUSER, C.A. **Projeto de banco de dados**. 2ª. Edição. Porto Alegre : Sagra Luzzatto, 1999.
- HIGGS, H.G. E GULLET, E.C.R. **The Management of Organization**. New York, U.S.A : McGraw-Hill Book Co., p. 527, 1976.
- JEPSON *et al.*, **Programando Aplicativos de Banco de Dados em Linux**. São Paulo : Makron Books, 2002.
- LAMBERT, D.M. E al. **Administração Estratégica da Logística**. São Paulo : Vantine consultoria, 1998.
- LARMAN, C. **Applying UML and Patterns**. New York : Prentice Hall, 1998.
- MENDONÇA, P.C.C.; KEEDI, S. **Transportes e seguros no comércio exterior**. São Paulo : Aduaneiras Ltda., 1997.
- MONTENEGRO, L.C.S. **Vantagens da utilização do transporte multimodal no comércio internacional brasileiro**. Disponível em: www.antt.gov.br
- MURTA, C.M. **Sistema de Informação Logística Para Distribuição de Carga Urbana**. Dissertação de Mestrado. Rio de Janeiro : IME, 1999.
- ORTENSI. L.O. **Software Livre**. São Paulo : 2006. Disponível em: www.ortensi.com
- PASSARI, A.F.L. **Marketplaces de cargas na otimização de recursos rodoviários de transporte: um estudo de caso**. Caderno de Pesquisas em Administração. São Paulo : v. 09, no 1, janeiro/março 2002.
- POMPILHO, S. **Análise Essencial**. 1ª. Edição. Rio de Janeiro : Infobook S.A., 1995.
- SILVA, R.O. **Modelo de apoio ao planejamento da mobilização militar dos transportes**. Dissertação de Mestrado. Rio de Janeiro : IME, 2001.
- TANAKA, A.K. **Notas de Aula do Curso de Projeto de Banco de Dados**. Rio de Janeiro : IME, 1997.
- THIRIET-LONGS, R.A. **Transporte Internacional de Carga – Uma Potencialidade Macroeconômica Brasileira**. Brasília : GEIPOT, 1982.
- TRANSPORTES. **Transportes e Mapas**. Disponível em: www.transportes.gov.br
- UCHÔA, H.N. **Geoprocessamento com Software Livre**. V 1.0. Rio de Janeiro : www.geolivre.org.br, 2004.
- ULLMAN, J.D. E WIDOM, J., **A First Course in Database System**. New Jersey, U.S.A. : Editora Prentice-Hall, 1997.

7 APÊNDICES

7.1 APÊNDICE 1: DESCRIÇÃO DE CASOS DE USO

1. TERMINAL

Nome: localizar dados de terminais

Atores: gerente e pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para consultar os dados dos terminais.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela para localização de terminais
2. o sistema exibe a tela de localização de terminais
3. o ator digita os dados:
 - código do terminal
 - nome do terminal
 - tipo do terminal (Aquaviário, ferroviário, rodoviário)
 - endereço do terminal
4. o ator dispara a consulta
5. o sistema retorna os dados do terminal encontrado

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados do terminal previamente cadastrado.

Nome: cadastrar terminal

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para incluir os dados dos terminais.

Descrição Detalhada:

• **Fluxo Principal:**

1. o ator chama a tela de cadastro de terminais
2. o sistema exibe a tela de cadastro de terminais
3. o ator digita os dados:
 - código do terminal
 - tipo do terminal (Aquaviário, ferroviário, rodoviário)
 - nome do terminal
 - endereço do terminal
 - capacidade de acesso
 - velocidade de acesso
 - capacidade de movimentação de granel solido
 - capacidade de movimentação de granel liquido
 - capacidade de movimentação de carga geral
 - capacidade de movimentação de container
4. o ator confirma a inclusão do terminal
5. o sistema salva os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

- 3.1) O código do terminal foi previamente cadastrado.
 - 3.1.1) O sistema exibe uma mensagem de erro informando que já existe cadastro com este código.
 - 3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.
- 3.2) O ator digita um valor inválido para o código.
 - 3.2.1) O sistema exibe uma mensagem de erro informando sobre a invalidez do código e o formato de entrada esperado.
 - 3.2.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes ao cadastro do novo terminal.

Nome: alterar terminal

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para alterar os dados dos terminais.

Descrição Detalhada:**• Fluxo Principal:**

1. o ator chama a tela de alteração de terminais
2. o sistema exibe a tela de alteração de terminais
3. o ator localiza os dados do terminal que ele deseja alterar
4. o ator digita os dados que deseja alterar
5. o ator confirma a alteração do terminal
6. o sistema salva os dados e exibe uma mensagem de confirmação

• Fluxo Alternativo:

- 3.1) O código do terminal não foi previamente cadastrado.
 - 3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.
 - 3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.
- 4.1) O ator digita um valor inválido para o código.
 - 4.1.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código e o formato de entrada esperado.
 - 4.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes a alteração do terminal.

Nome: excluir terminal

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para excluir os dados dos terminais.

Descrição Detalhada:**• Fluxo Principal:**

1. o ator chama a tela de exclusão de terminais
2. o sistema exibe a tela de exclusão de terminais
3. o ator localiza os dados do terminal que ele deseja excluir
4. o ator solicita a exclusão dos dados
5. o sistema exibe uma mensagem para confirmar a exclusão
6. o ator confirma a exclusão dos dados
7. o sistema exclui os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

3.1) O código do terminal não foi previamente cadastrado.

3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.

3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

6.1) O ator cancela a exclusão dos dados.

6.1.1) O sistema fecha a caixa de mensagem e retorna à tela principal

Pós-Condições: o sistema exclui os dados referentes ao terminal.

2. VEÍCULO - SISTEMA AQUAVIÁRIO

Nome: localizar dados de veículos sistema aquaviário

Atores: gerente e pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para consultar os dados dos veículos aquaviários.

Descrição Detalhada:

• **Fluxo Principal:**

1. o ator chama a tela para localização de veículos aquaviários
2. o sistema exibe a tela de localização de veículos aquaviários
3. o ator digita os dados:
-tipo de embarcação

4. o ator dispara a consulta
5. o sistema retorna os dados do veículo aquaviário encontrado

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados do veículo aquaviário previamente cadastrado.

Nome: cadastrar veículo sistema aquaviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para incluir os dados dos veículos aquaviários.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela de cadastro de veículos aquaviários
2. o sistema exibe a tela de cadastro de veículos aquaviários
3. o ator digita os dados:
 - código do veículo aquaviario
 - tipo de embarcação
 - força motriz
 - tipo de combustível
 - consumo de combustivel
 - custo de combustível
 - capacidade de carga
 - casco
 - comprimento
 - boca

-calado

-deslocamento médio

4. o ator confirma a inclusão do veículo aquaviário
5. o sistema salva os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

3.1) O código do veículo foi previamente cadastrado.

3.1.1) O sistema exibe uma mensagem de erro informando que já existe cadastro com este código.

3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

3.2) O ator digita um valor inválido para o código.

3.2.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código e o formato de entrada esperado.

3.2.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes ao cadastro do novo veículo.

Nome: alterar veículo sistema aquaviario

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para alterar os dados dos veículos aquaviários.

Descrição Detalhada:

• **Fluxo Principal:**

1. o ator chama a tela de alteração de veículos aquaviarios
2. o sistema exibe a tela de alteração de veículos aquaviarios
3. o ator localiza os dados do veículo que ele deseja alterar
4. o ator digita os dados que deseja alterar
5. o ator confirma a alteração do veículo
6. o sistema salva os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

3.1) O código do veículo não foi previamente cadastrado.

3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.

3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

4.1) O ator digita um valor inválido para o código.

4.1.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código e o formato de entrada esperado.

4.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes a alteração do veículo.

Nome: excluir veículo sistema aquaviario

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para excluir os dados dos veículos aquaviários.

Descrição Detalhada:

• **Fluxo Principal:**

1. o ator chama a tela de exclusão de veículos aquaviarios
2. o sistema exibe a tela de exclusão de veículos aquaviarios
3. o ator localiza os dados do veículo que ele deseja excluir
4. o ator solicita a exclusão dos dados
5. o sistema exibe uma mensagem para confirmar a exclusão
6. o ator confirma a exclusão dos dados
7. o sistema exclui os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

3.1) O código do veículo não foi previamente cadastrado.

3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.

3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

6.1) O ator cancela a exclusão dos dados.

6.1.1) O sistema fecha a caixa de mensagem e retorna à tela principal

Pós-Condições: o sistema exclui os dados referentes ao veículo aquaviário

3. VEÍCULO - SISTEMA RODOVIÁRIO

Nome: localizar dados de veículos sistema rodoviário

Atores: gerente e pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para consultar os dados dos veículos rodoviários.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela para localização de veículos rodoviários
2. o sistema exibe a tela de localização de veículos rodoviários
3. o ator digita os dados:
 - tipo de caminhão
4. o ator dispara a consulta
5. o sistema retorna os dados do veículo rodoviário encontrado

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados do veículo rodoviário previamente cadastrado.

Nome: cadastrar veículo sistema rodoviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema

2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para incluir os dados dos veículos rodoviários.

Descrição Detalhada:

• **Fluxo Principal:**

1. o ator chama a tela de cadastro de veículos rodoviários
2. o sistema exibe a tela de cadastro de veículos rodoviários
3. o ator digita os dados:
 - código do veículo rodoviário
 - tipo de caminhão
 - potência do caminhão
 - tipo de combustível
 - consumo de combustível
 - custo de combustível
 - capacidade de carga
 - fator de utilização
 - carga por tipo de eixo
 - poluentes do motor
 - custo do pneu
 - custo de recapagem
 - classe do vtr
 - marca e modelo
4. o ator confirma a inclusão do veículo
5. o sistema salva os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

- 3.1) O código do veículo foi previamente cadastrado.
 - 3.1.1) O sistema exibe uma mensagem de erro informando que já existe cadastro com este código.
 - 3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.
- 3.2) O ator digita um valor inválido para o código.

3.2.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código e o formato de entrada esperado.

3.2.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes ao cadastro do novo veículo.

Nome: alterar veículo sistema rodoviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para alterar os dados dos veículos rodoviários.

Descrição Detalhada:

• **Fluxo Principal:**

1. o ator chama a tela de alteração de veículos rodoviários
2. o sistema exibe a tela de alteração de veículos rodoviários
3. o ator localiza os dados do veículo que ele deseja alterar
4. o ator digita os dados que deseja alterar
5. o ator confirma a alteração do veículo
6. o sistema salva os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

3.1) O código do veículo não foi previamente cadastrado.

3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.

3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

4.1) O ator digita um valor inválido para o código.

4.1.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código e o formato de entrada esperado.

4.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes a alteração do veículo.

Nome: excluir veículo sistema rodoviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para excluir os dados dos veículos rodoviários.

Descrição Detalhada:

• **Fluxo Principal:**

1. o ator chama a tela de exclusão de veículos rodoviários
2. o sistema exibe a tela de exclusão de veículos rodoviários
3. o ator localiza os dados do veículo que ele deseja excluir
4. o ator solicita a exclusão dos dados
5. o sistema exibe uma mensagem para confirmar a exclusão
6. o ator confirma a exclusão dos dados
7. o sistema exclui os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

- 3.1) O código do veículo não foi previamente cadastrado.
 - 3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.
 - 3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.
- 6.1) O ator cancela a exclusão dos dados.
 - 6.1.1) O sistema fecha a caixa de mensagem e retorna à tela principal

Pós-Condições: o sistema exclui os dados referentes ao veículo rodoviário

4. VEÍCULO - SISTEMA FERROVIÁRIO

Nome: localizar dados de veículos sistema ferroviário

Atores: gerente e pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para consultar os dados dos veículos ferroviários.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela para localização de veículos ferroviários
2. o sistema exibe a tela de localização de veículos ferroviários
3. o ator digita os dados:
 - tipo de locomotiva
4. o ator dispara a consulta
5. o sistema retorna os dados do veículo ferroviário encontrado

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados do veículo ferroviário previamente cadastrado.

Nome: cadastrar veículo sistema ferroviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para incluir os dados dos veículos ferroviários.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela de cadastro de veículos ferroviários
2. o sistema exibe a tela de cadastro de veículos ferroviários
3. o ator digita os dados:
 - código do veículo ferroviário
 - tipo de locomotiva

- potência da locomotiva
- tipo de combustível
- consumo de combustível
- custo de combustível
- capacidade de carga
- bitola da linha
- tipo de carro vagão
- numero de locomotiva por trem
- numero de vagões por trem
- custo de manutenção de carros por vagão
- rampa

4. o ator confirma a inclusão do veículo

5. o sistema salva os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

3.1) O código do veículo foi previamente cadastrado.

3.1.1) O sistema exibe uma mensagem de erro informando que já existe cadastro com este código.

3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

3.2) O ator digita um valor inválido para o código.

3.2.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código e o formato de entrada esperado.

3.2.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes ao cadastro do novo veículo.

Nome: alterar veículo sistema ferroviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para alterar os dados dos veículos ferroviários.

Descrição Detalhada:**• Fluxo Principal:**

1. o ator chama a tela de alteração de veículos ferroviários
2. o sistema exibe a tela de alteração de veículos ferroviários
3. o ator localiza os dados do veículo que ele deseja alterar
4. o ator digita os dados que deseja alterar
5. o ator confirma a alteração do veículo
6. o sistema salva os dados e exibe uma mensagem de confirmação

• Fluxo Alternativo:

- 3.1) O código do veículo não foi previamente cadastrado.
 - 3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.
 - 3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.
- 4.1) O ator digita um valor inválido para o código.
 - 4.1.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código e o formato de entrada esperado.
 - 4.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes a alteração do veículo.

Nome: excluir veículo sistema ferroviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para excluir os dados dos veículos ferroviários.

Descrição Detalhada:**• Fluxo Principal:**

1. o ator chama a tela de exclusão de veículos ferroviários
2. o sistema exibe a tela de exclusão de veículos ferroviários
3. o ator localiza os dados do veículo que ele deseja excluir

4. o ator solicita a exclusão dos dados
5. o sistema exibe uma mensagem para confirmar a exclusão
6. o ator confirma a exclusão dos dados
7. o sistema exclui os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

- 3.1) O código do veículo não foi previamente cadastrado.
 - 3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.
 - 3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.
- 6.1) O ator cancela a exclusão dos dados.
 - 6.1.1) O sistema fecha a caixa de mensagem e retorna à tela principal

Pós-Condições: o sistema exclui os dados referentes ao veículo ferroviário

5. USUÁRIO

Nome: solicitar cadastro

Atores: gerente e pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para solicitar cadastro de usuário.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela de cadastro de usuário
2. o sistema exibe a tela de cadastro de usuário
3. o ator digita os dados:
 - nome do usuário
 - função do usuário (gerente ou pesquisador)
 - endereço residencial
 - nome da empresa
 - endereço comercial

-telefone para contato

-e_mail

4. o ator dispara a consulta
5. o sistema retorna os dados de solicitação de cadastro

● **Fluxo Alternativo:**

3.1) O ator preenche um dado incorreto.

3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.

3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados de solicitação de cadastro.

Nome: localizar dados do usuário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para consultar os dados dos usuários.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela para localização de usuários
2. o sistema exibe a tela de localização de usuários
3. o ator digita os dados:
 - código do usuário
4. o ator dispara a consulta
5. o sistema retorna os dados do usuário encontrado

● **Fluxo Alternativo:**

3.1) O ator preenche um dado incorreto.

3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.

3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados do usuário previamente cadastrado.

Nome: aprovar cadastro usuário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para aprovar e incluir os dados dos usuários.

Descrição Detalhada:**• Fluxo Principal:**

1. o ator chama a tela de cadastro de usuários
2. o sistema exibe a tela de cadastro de usuários
3. o ator digita os dados:
 - código do usuário
 - nome do usuário
 - função do usuário
 - endereço residencial
 - nome da empresa
 - endereço comercial
 - telefone para contato
 - email
4. o ator confirma a inclusão do usuário
5. o sistema salva os dados e exibe uma mensagem de confirmação

• Fluxo Alternativo:

- 3.1) O código do usuário foi previamente cadastrado.
 - 3.1.1) O sistema exibe uma mensagem de erro informando que já existe cadastro com este código.
 - 3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.
- 3.2) O ator digita um valor inválido para o código.
 - 3.2.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código.
 - 3.2.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

Pós-Condições: o sistema armazena os dados referentes ao cadastro do novo usuário.

Nome: alterar usuário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para alterar os dados dos usuários.

Descrição Detalhada:**• Fluxo Principal:**

1. o ator chama a tela de alteração de usuários
2. o sistema exibe a tela de alteração de usuários
3. o ator localiza os dados do usuário que ele deseja alterar
4. o ator digita os dados que deseja alterar
5. o ator confirma a alteração do usuário
6. o sistema salva os dados e exibe uma mensagem de confirmação

• Fluxo Alternativo:

- 3.1) O código do usuário não foi previamente cadastrado.
 - 3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.
 - 3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.
- 4.1) O ator digita um valor inválido para o código.
 - 4.1.1) O sistema exibe uma mensagem de erro informando sobre a invalidade do código e o formato de entrada esperado.
 - 4.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela

Pós-Condições: o sistema armazena os dados referentes a alteração do usuário.

Nome: excluir usuário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para excluir os dados dos usuários.

Descrição Detalhada:**• Fluxo Principal:**

1. o ator chama a tela de exclusão de usuários
2. o sistema exibe a tela de exclusão de usuários
3. o ator localiza os dados do usuário que ele deseja excluir
4. o ator solicita a exclusão dos dados
5. o sistema exibe uma mensagem para confirmar a exclusão
6. o ator confirma a exclusão dos dados
7. o sistema exclui os dados e exibe uma mensagem de confirmação

• **Fluxo Alternativo:**

3.1) O código do usuário não foi previamente cadastrado.

3.1.1) O sistema exibe uma mensagem de erro informando que não existe cadastro com este código.

3.1.2) O sistema limpa todos os campos e posiciona o cursor no primeiro campo da tela.

6.1) O ator cancela a exclusão dos dados.

6.1.1) O sistema fecha a caixa de mensagem e retorna à tela principal

Pós-Condições: o sistema exclui os dados referentes ao usuário.

6. SIG

Nome: localizar mapa geográfico com dados de aquavias, rodovias e ferrovias

Atores: gerente e pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para consultar mapa geográfico com dados de aquavias, rodovias e ferrovias.

Descrição Detalhada:

• **Fluxo Principal:**

1. o ator chama a tela do mapa geográfico
2. o sistema exibe a tela do mapa
3. o ator dispara a consulta
4. o sistema retorna os dados encontrados

● **Fluxo Alternativo:**

Pós-Condições: O sistema exibe o mapa e os dados cadastrados.

7. REQUISIÇÃO DE MUDANÇA

Nome: gerar requisição de mudança

Atores: pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para gerar requisição de mudança.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela de requisição de mudança
2. o sistema exibe a tela de requisição de mudança
3. o ator digita os dados:
 - nome do sistema (Terminal, veículo aquaviário, ferroviário ou rodoviário)
 - código (do veículo ou do terminal)
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: gerar requisição de mudança de veículo aquaviário

Atores: pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para gerar requisição de mudança de veículo aquaviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela de requisição de mudança
2. o sistema exibe a tela de requisição de mudança
3. o ator digita os dados:
 - nome do sistema (Terminal, veículo aquaviário, ferroviário ou rodoviário)
 - código (do veículo ou do terminal)
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: gerar requisição de mudança de veículo ferroviário

Atores: pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para gerar requisição de mudança de veículo ferroviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela de requisição de mudança
2. o sistema exibe a tela de requisição de mudança
3. o ator digita os dados:
 - nome do sistema (Terminal, veículo aquaviário, ferroviário ou rodoviário)
 - código (do veículo ou do terminal)

4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: gerar requisição de mudança de veículo rodoviário

Atores: pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para gerar requisição de mudança de veículo rodoviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela de requisição de mudança
2. o sistema exibe a tela de requisição de mudança
3. o ator digita os dados:
 - nome do sistema (Terminal, veículo aquaviário, ferroviário ou rodoviário)
 - código (do veículo ou do terminal)
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: gerar requisição de mudança de terminal

Atores: pesquisador

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para gerar requisição de mudança de terminal

Descrição Detalhada:**● Fluxo Principal:**

1. o ator chama a tela de requisição de mudança
2. o sistema exibe a tela de requisição de mudança
3. o ator digita os dados:
 - sistema (Terminal aquaviário, ferroviário ou rodoviário)
 - código do terminal
 - descrição da alteração
 - código de usuário
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● Fluxo Alternativo:

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: localizar requisição de mudança

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para localizar requisição de mudança.

Descrição Detalhada:**● Fluxo Principal:**

1. o ator chama a tela para localização de requisição de mudança
2. o sistema exibe a tela de localização de requisição de mudança

3. o ator digita os dados:
 - nome do sistema (Terminais, veículos aquaviários, ferroviários ou rodoviários)
 - código (do veículo ou do terminal)
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: localizar requisição de mudança de veículo aquaviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para localizar requisição de mudança de veículo aquaviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela para localização de requisição de mudança
2. o sistema exibe a tela de localização de requisição de mudança
3. o ator digita os dados:
 - nome do sistema (Terminais, veículos aquaviários, ferroviários ou rodoviários)
 - código (do veículo ou do terminal)
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: localizar requisição de mudança de veículo ferroviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para localizar requisição de mudança de veículo ferroviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela para localização de requisição de mudança
2. o sistema exibe a tela de localização de requisição de mudança
3. o ator digita os dados:
 - nome do sistema (Terminais, veículos aquaviários, ferroviários ou rodoviários)
 - código (do veículo ou do terminal)
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: localizar requisição de mudança de veículo rodoviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para localizar requisição de mudança de veículo rodoviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela para localização de requisição de mudança
2. o sistema exibe a tela de localização de requisição de mudança
3. o ator digita os dados:
 - nome do sistema (Terminais, veículos aquaviários, ferroviários ou rodoviários)
 - código (do veículo ou do terminal)
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

- 3.1) O ator preenche um dado incorreto.
 - 3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: localizar requisição de mudança de terminal

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para localizar requisição de mudança de terminal

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator chama a tela para localização de requisição de mudança
2. o sistema exibe a tela de localização de requisição de mudança
3. o ator digita os dados:
 - nome do sistema (Terminais, veículos aquaviários, ferroviários ou rodoviários)
 - código (do veículo ou do terminal)
4. o ator dispara a consulta
5. o sistema retorna os dados da requisição de mudança encontrada

● **Fluxo Alternativo:**

3.1) O ator preenche um dado incorreto.

3.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.

3.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O sistema localiza e exibe os dados da requisição de mudança.

Nome: aprovar requisição de mudança

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para aprovar requisição de mudança.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator localiza os dados referentes a requisição de mudança
2. o ator aprova a requisição de mudança
3. o sistema salva os dados modificados e exibe uma mensagem de confirmação

● **Fluxo Alternativo:**

2.1) O ator cancela a aprovação.

2.1.1) O sistema retorna e exibe a tela de requisição de mudança.

Pós-Condições: O sistema armazena os dados modificados.

Nome: aprovar requisição de mudança de veículo aquaviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para aprovar requisição de mudança de veículo aquaviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator localiza os dados referentes a requisição de mudança
2. o ator aprova a requisição de mudança

3. o sistema salva os dados modificados e exibe uma mensagem de confirmação

● **Fluxo Alternativo:**

2.1) O ator cancela a aprovação.

2.1.1) O sistema retorna e exibe a tela de requisição de mudança.

Pós-Condições: O sistema armazena os dados modificados.

Nome: aprovar requisição de mudança de veículo ferroviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para aprovar requisição de mudança de veículo ferroviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator localiza os dados referentes a requisição de mudança
2. o ator aprova a requisição de mudança
3. o sistema salva os dados modificados e exibe uma mensagem de confirmação

● **Fluxo Alternativo:**

2.1) O ator cancela a aprovação.

2.1.1) O sistema retorna e exibe a tela de requisição de mudança.

Pós-Condições: O sistema armazena os dados modificados.

Nome: aprovar requisição de mudança de veículo rodoviário

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para aprovar requisição de mudança de veículo rodoviário

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator localiza os dados referentes a requisição de mudança
2. o ator aprova a requisição de mudança
3. o sistema salva os dados modificados e exibe uma mensagem de confirmação

● **Fluxo Alternativo:**

- 2.1) O ator cancela a aprovação.
 - 2.1.1) O sistema retorna e exibe a tela de requisição de mudança.

Pós-Condições: O sistema armazena os dados modificados.

Nome: aprovar requisição de mudança de terminal

Atores: gerente

Pré-Condições:

1. estar conectado na tela principal do sistema
2. ser cadastrado no sistema e ter perfil de gerente

Descrição Geral: o ator interage com o sistema para aprovar requisição de mudança de terminal

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator localiza os dados referentes a requisição de mudança
2. o ator aprova a requisição de mudança
3. o sistema salva os dados modificados e exibe uma mensagem de confirmação

● **Fluxo Alternativo:**

- 2.1) O ator cancela a aprovação.
 - 2.1.1) O sistema retorna e exibe a tela de requisição de mudança.

Pós-Condições: O sistema armazena os dados modificados.

8. CONECTAR / DESCONECTAR DO SISTEMA

Nome: conectar no sistema

Atores: gerente e pesquisador

Pré-Condições:

1. estar na tela de entrada
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para conectar-se.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator digita o login e senha
2. o sistema verifica se pesquisador foi previamente cadastrado
3. o sistema abre conexão e exibe a tela principal

● **Fluxo Alternativo:**

- 1.1) O ator digita um dado incorreto.
 - 1.1.1) O sistema exibe uma mensagem de erro indicando qual dos valores é inválido.
 - 1.1.2) O sistema limpa os dados e posiciona o cursor no primeiro campo.

Pós-Condições: O pesquisador se conecta e o sistema exibe a tela principal.

Nome: desconectar do sistema

Atores: gerente e pesquisador

Pré-Condições:

1. estar conectado
2. ser cadastrado no sistema

Descrição Geral: o ator interage com o sistema para desconectar.

Descrição Detalhada:

● **Fluxo Principal:**

1. o ator seleciona a opção de sair
2. o sistema fecha conexão e exibe a tela de entrada

Pós-Condições: O sistema desconecta o pesquisador.

7.2 APÊNDICE 2: REFERÊNCIAS DE SQL

Este apêndice contém informação gerais de referência para os comandos SQL suportados pelo PostgreSQL. Informações detalhadas de cada comando com relação ao padrão podem ser encontradas nas respectivas páginas de referência em www.javalinux.com.br .

A) SUMÁRIO

ABORT -- interrompe a transação corrente

ALTER AGGREGATE -- altera a definição de uma função de agregação

ALTER CONVERSION -- altera a definição de uma conversão de codificação

ALTER DATABASE -- altera um banco de dados

ALTER DOMAIN -- altera a definição de um domínio

ALTER FUNCTION -- altera a definição de uma função

ALTER GROUP -- altera um grupo de usuários

ALTER LANGUAGE -- altera a definição de uma linguagem procedural

ALTER OPERATOR CLASS -- altera a definição de uma classe de operadores

ALTER SCHEMA -- altera a definição de um esquema

ALTER SEQUENCE -- altera a definição de um gerador de seqüência

ALTER TABLE -- altera a definição de uma tabela

ALTER TRIGGER -- altera a definição de um gatilho

ALTER USER -- altera uma conta de usuário do banco de dados

ANALYZE -- coleta estatísticas sobre o banco de dados

BEGIN -- inicia um bloco de transação

CHECKPOINT -- força um ponto de controle no registro de transação

CLOSE -- fecha o cursor

CLUSTER -- agrupa a tabela de acordo com um índice

COMMENT -- define ou muda o comentário sobre um objeto

COMMIT -- efetiva a transação corrente

COPY -- copia dados entre um arquivo e uma tabela

CREATE AGGREGATE -- cria uma função de agregação

CREATE CAST -- cria uma conversão de tipo de dado

CREATE CONSTRAINT TRIGGER -- cria um gatilho de restrição
CREATE CONVERSION -- cria uma conversão de codificação
CREATE DATABASE -- cria um banco de dados
CREATE DOMAIN -- cria um domínio
CREATE FUNCTION -- cria uma função
CREATE GROUP -- cria um grupo de usuários
CREATE INDEX -- cria um índice
CREATE LANGUAGE -- cria uma linguagem procedural
CREATE OPERATOR -- cria um operador
CREATE OPERATOR CLASS -- cria uma classe de operadores
CREATE RULE -- cria uma regra de reescrita
CREATE SCHEMA -- cria um esquema
CREATE SEQUENCE -- cria um gerador de seqüência
CREATE TABLE -- cria uma tabela
CREATE TABLE AS -- cria uma tabela a partir dos resultados de uma consulta
CREATE TRIGGER -- cria um gatilho
CREATE TYPE -- cria um tipo de dado
CREATE USER -- cria uma conta de usuário do banco de dados
CREATE VIEW -- cria uma visão
DEALLOCATE -- remove um comando preparado
DECLARE -- define um cursor
DELETE -- exclui linhas de uma tabela
DROP AGGREGATE -- remove uma função de agregação
DROP CAST -- remove uma conversão de tipo de dado
DROP CONVERSION -- remove uma conversão de codificação
DROP DATABASE -- remove um banco de dados
DROP DOMAIN -- remove um domínio
DROP FUNCTION -- remove uma função
DROP GROUP -- remove um grupo de usuários
DROP INDEX -- remove um índice
DROP LANGUAGE -- remove uma linguagem procedural

DROP OPERATOR -- remove um operador
DROP OPERATOR CLASS -- remove uma classe de operadores
DROP RULE -- remove uma regra de reescrita
DROP SCHEMA -- remove um esquema
DROP SEQUENCE -- remove uma seqüência
DROP TABLE -- remove uma tabela
DROP TRIGGER -- remove um gatilho
DROP TYPE -- remove um tipo de dado
DROP USER -- remove uma conta de usuário do banco de dados
DROP VIEW -- remove uma visão
END -- efetiva a transação corrente
EXECUTE -- executa um comando preparado
EXPLAIN -- mostra o plano de execução de um comando
FETCH -- traz linhas de uma consulta usando um cursor
GRANT -- define privilégios de acesso
INSERT -- cria novas linhas na tabela
LISTEN -- ouve uma notificação
LOAD -- carrega ou recarrega um arquivo de biblioteca compartilhada
LOCK -- bloqueia uma tabela
MOVE -- posiciona o cursor
NOTIFY -- gera uma notificação
PREPARE -- prepara um comando para execução
REINDEX -- reconstrói índices
RESET -- redefine o valor de um parâmetro em tempo de execução com seu valor padrão
REVOKE -- revoga privilégios de acesso
ROLLBACK -- interrompe a transação corrente
SELECT -- retorna linhas de uma tabela ou de uma visão
SELECT INTO -- cria uma tabela a partir dos resultados de uma consulta
SET -- muda um parâmetro de tempo de execução
SET CONSTRAINTS -- define os modos de verificação da restrição na transação corrente
SET SESSION AUTHORIZATION -- define o identificador do usuário da sessão e o

identificador do usuário corrente, da sessão corrente.

SET TRANSACTION -- define as características da transação corrente

SHOW -- mostra o valor de um parâmetro de tempo de execução

START TRANSACTION -- inicia um bloco de transação

TRUNCATE -- esvazia a tabela

UNLISTEN -- pára de escutar uma notificação

UPDATE -- atualiza linhas de uma tabela

VACUUM -- limpa e opcionalmente analisa um banco de dados

7.3 APÊNDICE 3: TELAS DO PROTÓTIPO



FIG.7.1: Janela de Localização de Terminais

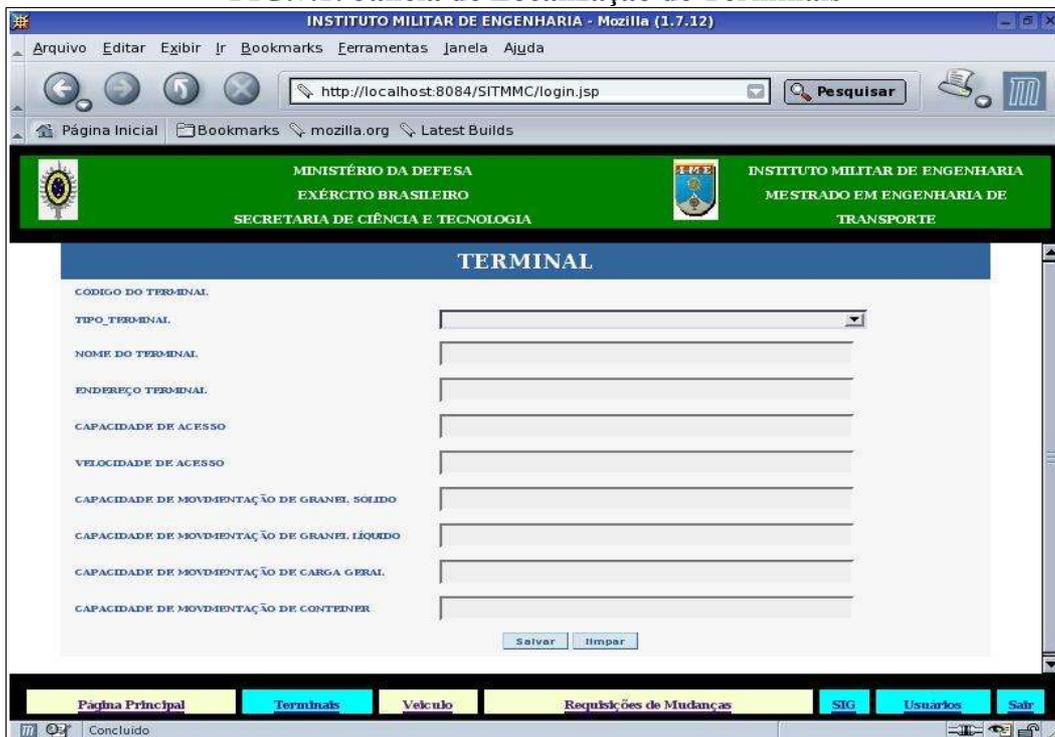


FIG.7.2: Janela Cadastro de Terminais



FIG.7.3: Janela de Seleção de Veículos

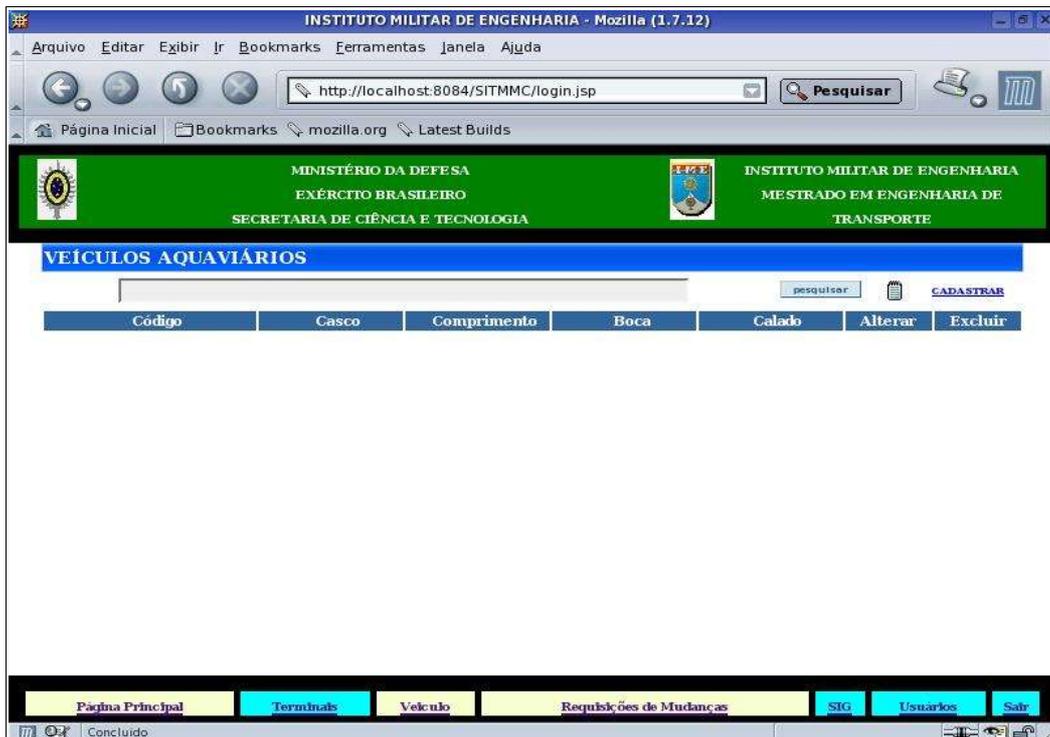


FIG.7.4: Janela de Localização de Veículos Aquaviários

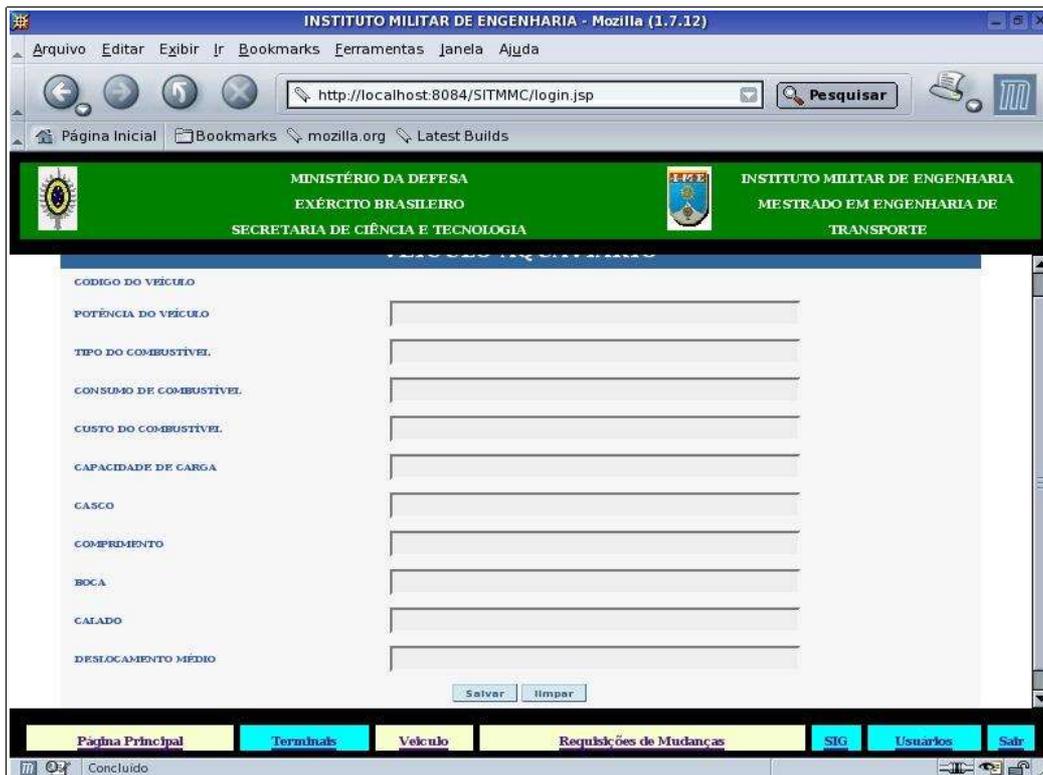


FIG.7.5: Cadastramento de Veículos Aquaviários

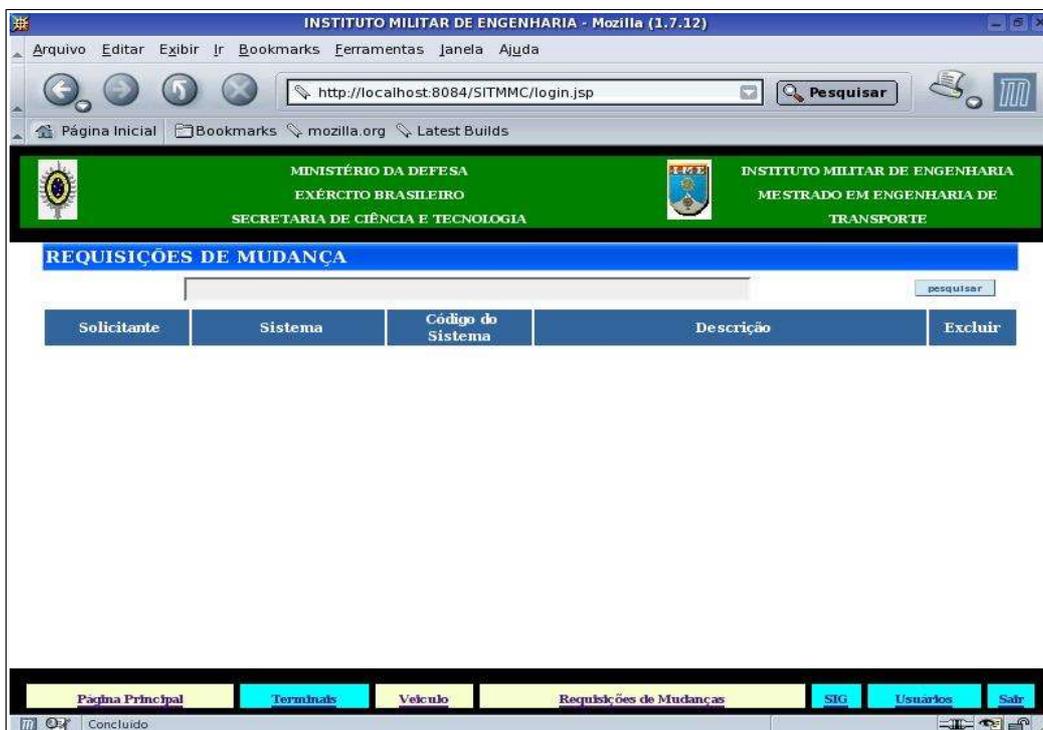


FIG.7.6: Janela de Requisição de Mudanças

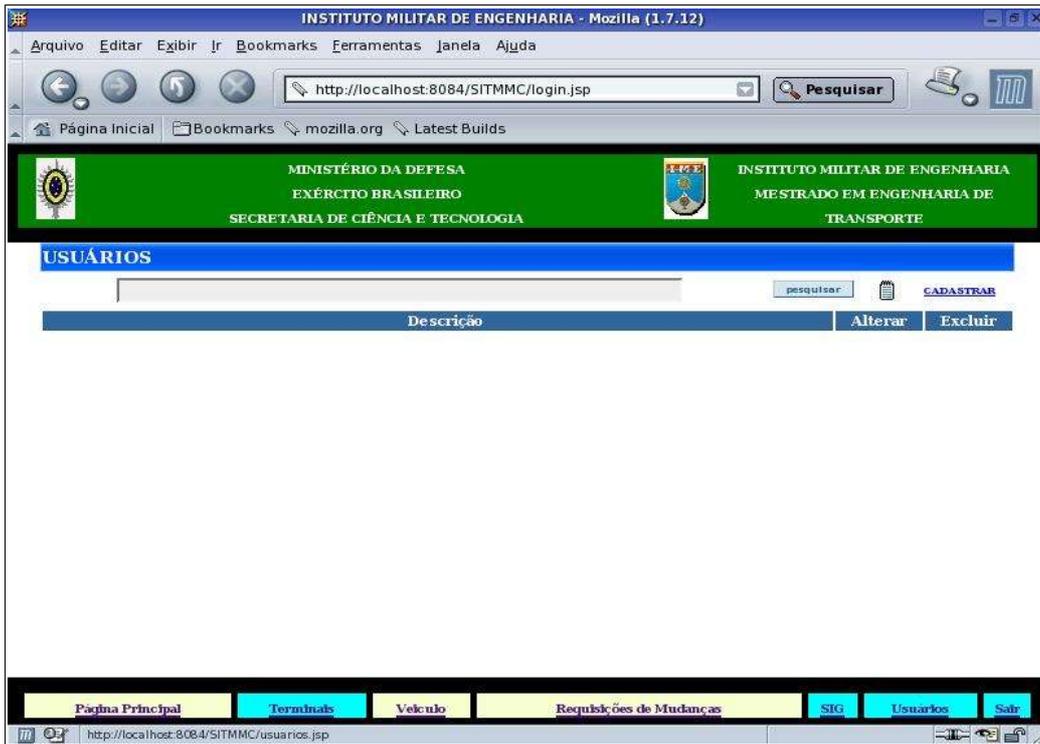


FIG.7.7: Janela de Localização de Usuários

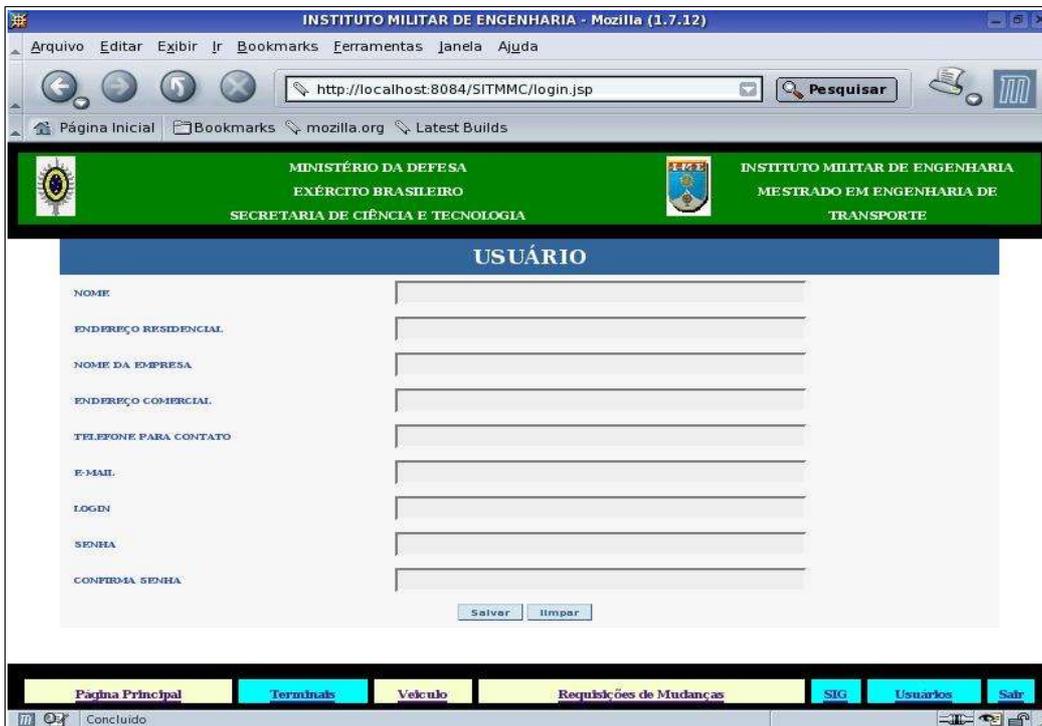


FIG.7.8: Janela de Cadastro de Usuários

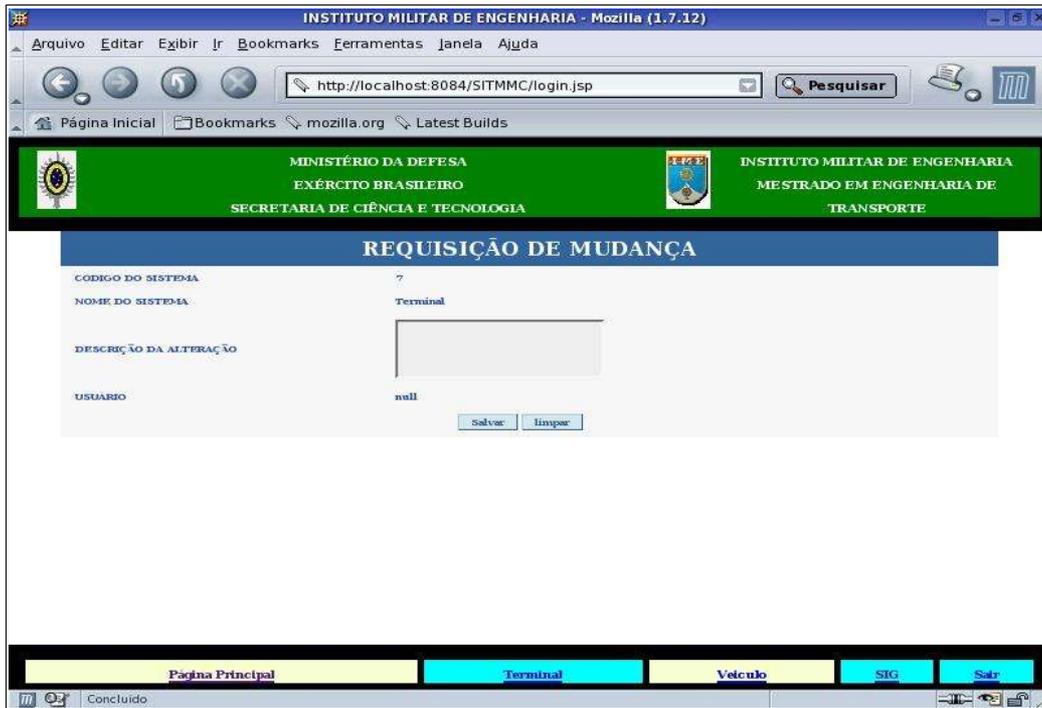


FIG.7.9: Janela de Requisição de Mudança pelo Usuário

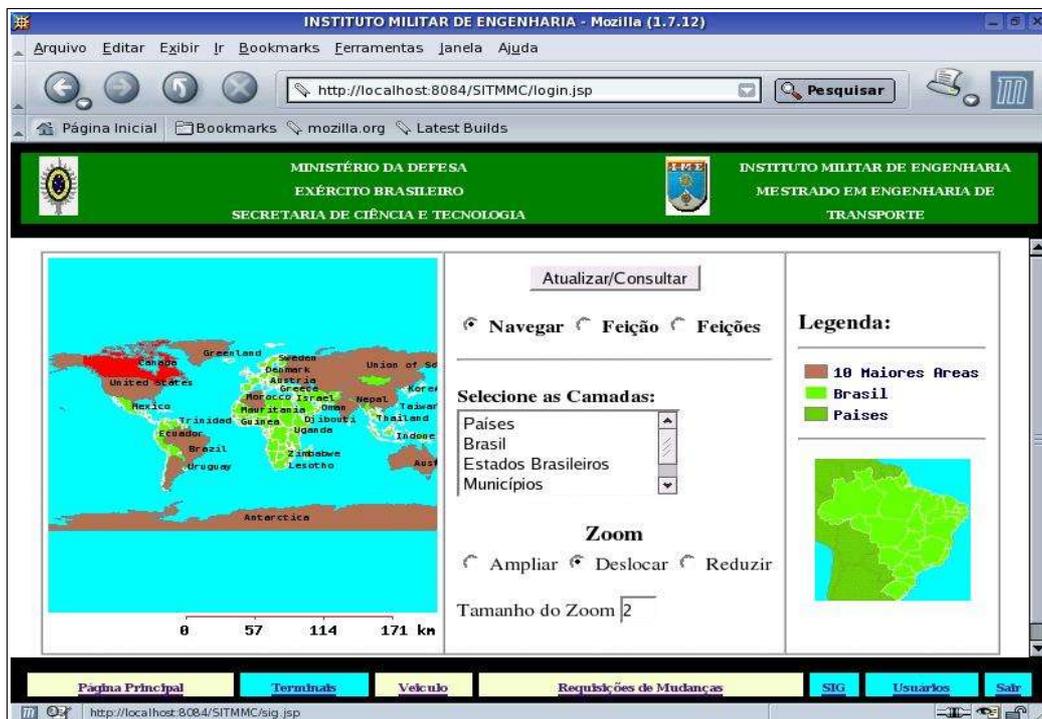


FIG.7.10: Janela de Localização de um País

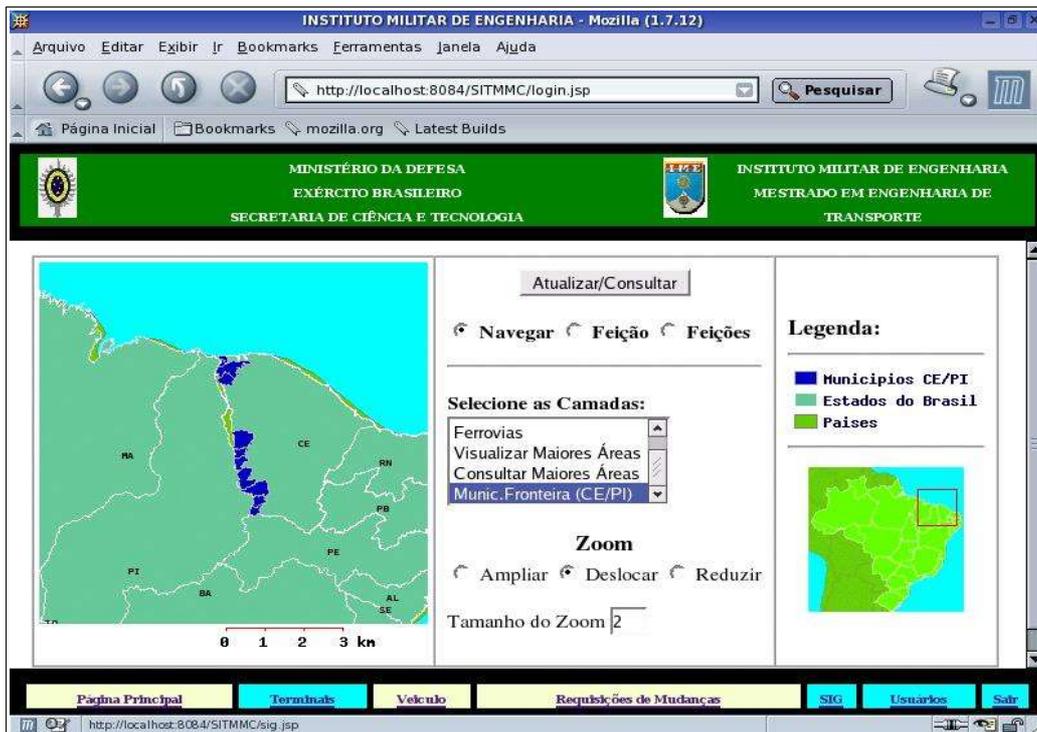


FIG.7.11: Janela de Municípios entre Ceará e Piauí

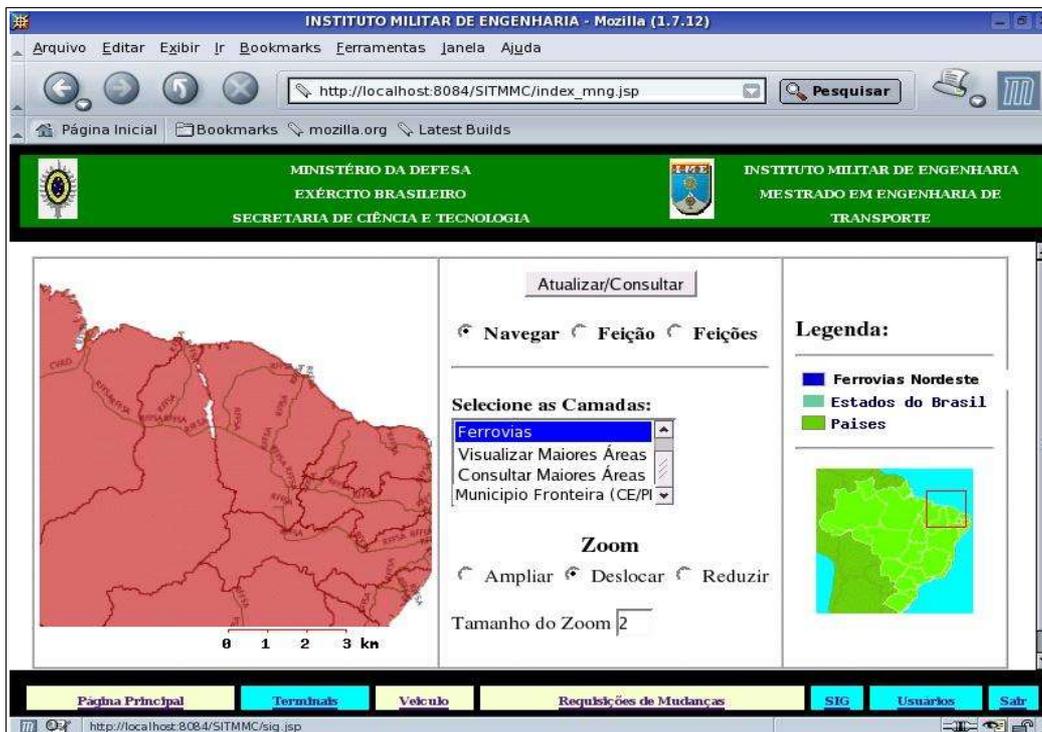


FIG.7.12: Janela de Ferrovias do Nordeste

7.4 APÊNDICE 4: JSP/JAVA

O JSP (Java Server Pages) é uma tecnologia Web J2EE que utiliza o conceito de container, responsável por gerar conteúdo dinâmico via protocolo HTTP e fornecer o runtime para aplicações, entre outras tarefas.

As principais características do JSP são:

- Separação do conteúdo estático do dinâmico: a lógica de geração de conteúdo dinâmico é mantida separada das tags HTML, responsáveis pela interface para o usuário. A parte lógica é encapsulada em componentes Java do tipo JavaBeans, que são utilizadas pelas páginas JSP através de scriptlets, códigos Java puro inseridos dentro da página JSP, tendo as vantagens tanto da OO quanto de Java ou tags especiais, chamadas taglibs.
- Roda em qualquer plataforma: como a tecnologia JSP é uma extensão da plataforma Java, as páginas JSP têm a vantagem da independência de plataforma.
- Diversos formatos: possibilidade de implementação de linguagens como HTML, XML, etc.

A) SERVIDOR JSP TOMCAT

Para que as páginas JSP sejam executadas, é preciso instalar na máquina um servidor JSP.

Existem vários servidores JSP espalhados na Web, alguns deles, dispendo de versões para plataformas Windows e Linux. Um dos mais famosos é o Tomcat, do projeto Jakarta, que pode ser baixado do site <http://jakarta.apache.org/tomcat>.

B) PLATAFORMA JAVA J2EE

“O estado da arte no desenvolvimento de sistemas corporativos está no desenvolvimento. Temos três camadas lógicas, a saber: camada de apresentação, camada de negócio e camada de persistência. Atualmente existem no mercado, algumas plataformas de desenvolvimento de software corporativo em camadas, dentre as quais se destacam a

plataforma Java, a plataforma .NET e a plataforma CORBA (Common Object Request Broker Architecture). Cada uma dessas plataformas possui características próprias que impactam no processo de desenvolvimento, no desempenho final, etc. A plataforma CORBA é mais complexa, porém mais versátil, podendo ser usada com várias linguagens. Ainda apresenta alguns problemas de produtividade, mas possui um ótimo desempenho, principalmente devido ao fato de comunicar com outras plataformas como, por exemplo, Java.

A plataforma Java J2EE1.4 oferece uma base de desenvolvimento ampla, estável, bem testada e com alto desempenho final. No entanto essa versatilidade acarreta um certo prejuízo no tempo de desenvolvimento do software. Comparativamente a plataforma .NET oferece uma maior produtividade no desenvolvimento, no entanto é uma tecnologia nova e instável, apresentando vários problemas no desenvolvimento de software.”

“A plataforma J2EE1.4 é uma plataforma multicamada que pode estar com as camadas distribuídas em várias máquinas ou em uma só máquina. Geralmente conhecida como 3 camadas, mas pode ter mais camadas:

- Camada cliente que executa na máquina do cliente;
- Camada Web que executa no servidor J2EE
- Camada de negócio que executa no servidor J2EE
- Camada EIS (Enterprise Information System) software que executa no servidor EIS.

A plataforma trabalha com conceito de containers de componentes. Um container fornece serviços aos componentes, tais como: ciclo de vida, segurança, etc. Tem vários tipos de container, tais como:

- Servidor Java EE;
- Container EJB (Enterprise JavaBeans);
- Container Web;

- Container Aplicação cliente;
- Container Applet;

Esta plataforma não oferece grande produtividade devido ao seu container EJB. Além de criar várias classes, há códigos complexos para um programador inexperiente se aventurar na programação em Java utilizando banco de dados com os EJB's de entidade.

Para se criar um bean de sessão na plataforma J2EE 1.4 tem-se:

- classe de bean de sessão;
- interface de base;
- interface remota;
- interface local;

Para criar uma conexão com o banco de dados usando beans de entidade, tem-se que criar:

- classe de bean de entidade;
- interface EntityBean;
- Chamada do Banco de dados;
- interface local;
- interface remota.

Todos esses métodos e interfaces têm que ser feitos na mão utilizando códigos complicados ou são gerados por uma IDE (Integrated Development Environment). Para adicionar uma classe de entidade com bean gerenciando persistência deve ter estes requisitos:

- A classe é definida como public;
- A classe não pode ser definida como abstract ou final;
- Ele contém um construtor vazio;
- Ele não faz implementação do método finalize.”

C) ESTRUTURA DO J2EE 1.4

“Na plataforma J2EE1.4 a comunicação entre as estruturas de camadas é mais complexa de se programar, dificultando o desenvolvimento de software, mesmo que depois de finalizado tenha um bom desempenho.

A página web se comunica com a camada web, que é constituída pelo container de Servlets e página JSP. Esse Servlet se comunica com a fábrica de EJB de Sessão, que retorna com os objetos EJB para a Servlet. O container EJB possui os Beans de Sessão e os Beans de Entidades.

Os dados enviados de um cliente html são recebidos no servlet que cria um objeto de valor (value object), ou seja, um objeto com os dados recebidos. Nesse ponto o servlet precisa usar um bean de sessão. Isso é feito invocando a fábrica de bean de sessão, ou seja, o servlet precisa conhecer explicitamente a fábrica de beans de sessão e solicitar um bean de sessão para a fábrica.

Esse processo envolve duas interfaces: a interface de fábrica e a interface do bean de sessão. Analogicamente o bean de sessão necessita de beans de entidade que são obtidos, explicitamente, de uma consulta à fábrica de beans de entidade.” (Fonte: http://www.linhadecodigo.com.br/artigos_impressao.asp?id_ac=1085, 2007)

D) CÓDIFICAÇÃO JSP / JAVA

1. CAMADA DE APRESENTAÇÃO (JSP + BEAN)

1.1 JSP

addTerminal.jsp

```
<%@ page errorPage="/erro.jsp" %>
<%@ page import="java.util.*,sitmmc.*,sitmmc.vo.*"%>
<%response.setHeader("Cache-Control","no-cache");%>
<jsp:useBean id="terminal" scope="page" class="sitmmc.bean.BeanTerminal"/>
<jsp:setProperty name="terminal" property="*" />
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Sistema de Informação Para Transporte Multimodal de
Carga</TITLE>
<link rel="stylesheet" type="text/css" href="includes/estilo.css">
<script language="javascript" src="includes/jscripts.js"></script>
```

```

<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<BODY bgColor=white topMargin=3>
<%
    terminal.inserir();
%><br><br><br>
<table width="62%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr class="main_tool">
    <td height="22" align="center" valign="middle">SUCESSO</td>
  </tr>
  <tr class="tbarea">
    <td height="85" align="center" valign="middle">O <strong>TERMINAL</strong>
    foi cadastrado com sucesso!</td>
  </tr>
  <tr class="lnk">
    <td height="19" align="center" valign="middle"><a href="terminais.jsp"
target="_self">VOLTAR</a></td>
  </tr>
</table>
</BODY>
</HTML>

```

cngTerminal.jsp

```

<%@ page errorPage="/erro.jsp" %>
<%@ page import="java.util.*,sittmmc.*,sittmmc.vo.*"%>
<%response.setHeader("Cache-Control","no-cache");%>
<jsp:useBean id="terminal" scope="page" class="sittmmc.bean.BeanTerminal"/>
<jsp:setProperty name="terminal" property="*" />
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Sistema de Informação Para Transporte Multimodal de
Carga</TITLE>
<link rel="stylesheet" type="text/css" href="includes/estilo.css">
<script language="javascript" src="includes/jscripts.js"></script>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<BODY bgColor=white topMargin=3>
<%
    terminal.alterar();
%><br><br><br>
<table width="62%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr class="main_tool">
    <td height="22" align="center" valign="middle">SUCESSO</td>
  </tr>
  <tr class="tbarea">
    <td height="85" align="center" valign="middle">O <strong>TERMINAL</strong>
    foi alterado com sucesso!</td>
  </tr>

```

```

<tr class="lnk">
  <td height="19" align="center" valign="middle"><a href="terminais.jsp"
target="_self">VOLTAR</a></td>
</tr>
</table>
</BODY>
</HTML>

```

delTerminal.jsp

```

<%@ page errorPage="/erro.jsp" %>
<%@ page import="java.util.*,sitmmc.*,sitmmc.vo.*"%>
<%response.setHeader("Cache-Control","no-cache");%>
<jsp:useBean id="terminal" scope="page" class="sitmmc.bean.BeanTerminal"/>
<jsp:setProperty name="terminal" property="*" />
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Sistema de Informação Para Transporte Multimodal de
Carga</TITLE>
<link rel="stylesheet" type="text/css" href="includes/estilo.css">
<script language="javascript" src="includes/jscripts.js"></script>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<BODY bgColor=white topMargin=3>
<%
  terminal.excluir();
%><br><br><br>
<table width="62%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr class="main_tool">
    <td height="22" align="center" valign="middle">SUCESSO</td>
  </tr>
  <tr class="tbarea">
    <td height="85" align="center" valign="middle">O <strong>TERMINAL</strong>
    foi removido com sucesso!</td>
  </tr>
  <tr class="lnk">
    <td height="19" align="center" valign="middle"><a href="terminais.jsp"
target="_self">VOLTAR</a></td>
  </tr>
</table>
</BODY>
</HTML>

```

1.2 BEAN

BeanTerminal.java

```
package sitmmc.bean;

import sitmmc.*; import sitmmc.bean.*; import sitmmc.db.*; import sitmmc.vo.*;
import java.sql.*;
import java.util.*;

/**
 * Title: Sistema de Informação Para Transporte Multimodal de Carga
 */

public class BeanTerminal {
    private String codigo_terminal="";
    private String tipo_terminal="";
    private String nome_terminal="";
    private String endereco_terminal="";
    private String capacidade_acesso="";
    private String velocidade_acesso="";
    private String capacidade_movimentacao_granel_solido="";
    private String capacidade_movimentacao_granel_liquido="";
    private String capacidade_movimentacao_carga_geral="";
    private String capacidade_movimentacao_container="";

    public String getCodigo_terminal(){
        return this.codigo_terminal;
    }
    public void setCodigo_terminal(String codigo_terminal){
        this.codigo_terminal = codigo_terminal;
    }
    public void setTipo_terminal(String tipo_terminal){
        this.tipo_terminal = tipo_terminal;
    }
    public String getTipo_terminal(){
        return this.tipo_terminal;
    }
    public void setNome_terminal(String nome_terminal){
        this.nome_terminal = nome_terminal;
    }
    public String getNome_terminal(){
        return this.nome_terminal;
    }
    public void setEndereco_terminal(String endereco_terminal){
```

```

    this.endereco_terminal = endereco_terminal;
}
public String getEndereco_terminal(){
    return this.endereco_terminal;
}
public void setCapacidade_aceso(String capacidade_aceso){
    this.capacidade_aceso = capacidade_aceso;
}
public String getCapacidade_aceso(){
    return this.capacidade_aceso;
}
public void setVelocidade_aceso(String velocidade_aceso){
    this.velocidade_aceso = velocidade_aceso;
}
public String getVelocidade_aceso(){
    return this.velocidade_aceso;
}
public void setCapacidade_movimentacao_granel_solido(String
capacidade_movimentacao_granel_solido){
    this.capacidade_movimentacao_granel_solido = capacidade_movimentacao_granel_solido;
}
public String getCapacidade_movimentacao_granel_solido(){
    return this.capacidade_movimentacao_granel_solido;
}
public void setCapacidade_movimentacao_granel_liquido(String
capacidade_movimentacao_granel_liquido){
    this.capacidade_movimentacao_granel_liquido =
capacidade_movimentacao_granel_liquido;
}
public String getCapacidade_movimentacao_granel_liquido(){
    return this.capacidade_movimentacao_granel_liquido;
}
public void setCapacidade_movimentacao_carga_geral(String
capacidade_movimentacao_carga_geral){
    this.capacidade_movimentacao_carga_geral = capacidade_movimentacao_carga_geral;
}
public String getCapacidade_movimentacao_carga_geral(){
    return this.capacidade_movimentacao_carga_geral;
}
public void setCapacidade_movimentacao_container(String
capacidade_movimentacao_container){
    this.capacidade_movimentacao_container = capacidade_movimentacao_container;
}
public String getCapacidade_movimentacao_container(){
    return this.capacidade_movimentacao_container;
}
public void inserir() throws SitmmcException{

```

```

    Terminal terminal = new Terminal(0, this.tipo_terminal, this.nome_terminal,
this.endereco_terminal, this.capacidade_ acesso, this.velocidade_ acesso,
this.capacidade_ movimentacao_ granel_ solido,
this.capacidade_ movimentacao_ granel_ liquido, this.capacidade_ movimentacao_ carga_ geral,
this.capacidade_ movimentacao_ container);
    PerTerminal perterminal = new PerTerminal();
    perterminal.inserir(terminal);
}
public void alterar() throws SitmmcException {
    Terminal terminal = new Terminal(Integer.parseInt(this.codigo_terminal),
this.tipo_terminal, this.nome_terminal, this.endereco_terminal, this.capacidade_ acesso,
this.velocidade_ acesso, this.capacidade_ movimentacao_ granel_ solido,
this.capacidade_ movimentacao_ granel_ liquido, this.capacidade_ movimentacao_ carga_ geral,
this.capacidade_ movimentacao_ container);
    PerTerminal perterminal = new PerTerminal();
    perterminal.alterar(terminal);
}
public void excluir() throws SitmmcException {
    PerTerminal perterminal = new PerTerminal();
    perterminal.excluir(Integer.parseInt(this.codigo_terminal));
}
public void getTerminal() throws SitmmcException {
    PerTerminal perterminal = new PerTerminal();
    Terminal terminal = (Terminal) perterminal.consultar(Integer.parseInt
(this.codigo_terminal));
    this.codigo_terminal=String.valueOf(terminal.getCodigo_terminal());
    this.tipo_terminal=terminal.getTipo_terminal();
    this.nome_terminal=terminal.getNome_terminal();
    this.endereco_terminal=terminal.getEndereco_terminal();
    this.capacidade_ acesso=terminal.getCapacidade_ acesso();
    this.velocidade_ acesso=terminal.getVelocidade_ acesso();
    this.capacidade_ movimentacao_ granel_ solido=terminal.getCapacidade_ mo
vimentacao_ granel_ solido();
    this.capacidade_ movimentacao_ granel_ liquido=terminal.getCapacidade_ m
ovimentacao_ granel_ liquido();
    this.capacidade_ movimentacao_ carga_ geral=terminal.getCapacidade_ movi
mentacao_ carga_ geral();
    this.capacidade_ movimentacao_ container=terminal.getCapacidade_ movime
ntacao_ container();
}
public Collection getTerminais() throws SitmmcException {
    PerTerminal perterminal = new PerTerminal();
    return perterminal.consultarterminais(this.nome_terminal);
}
}
}

```

2. CAMADA DE PERSISTÊNCIA

PerTerminal.java

```
package sitmmc.db;

import sitmmc.*; import sitmmc.db.*; import sitmmc.vo.*;
import java.sql.*;
import java.util.*;

/**
 * Title: Sistema de Informação Para Transporte Multimodal de Carga
 */

public class PerTerminal {

    private Database db;
    private Connection con;
    private PreparedStatement pst;
    private Statement st;
    private ResultSet rs;
    private String sql;

    public PerTerminal(){
        db = new Database();
    }

    public void inserir(Terminal terminal) throws SitmmcException{
        try{
            con = db.getConn();
            pst = con.prepareStatement("insert into terminal(tipo_terminal,
nome_terminal," +
                "endereco_terminal, capacidade_aceso, velocidade_aceso," +
                "capacidade_movimentacao_granel_solido," +
                "capacidade_movimentacao_granel_liquido," +
                "capacidade_movimentacao_carga_geral,
capacidade_movimentacao_container) values(?, ?, ?, ?, ?, ?, ?, ?, ?)");
            pst.setString(1, terminal.getTipo_terminal());
            pst.setString(2, terminal.getNome_terminal());
            pst.setString(3, terminal.getEndereco_terminal());
            pst.setString(4, terminal.getCapacidade_aceso());
            pst.setString(5, terminal.getVelocidade_aceso());
            pst.setString(6, terminal.getCapacidade_movimentacao_granel_solido());
            pst.setString(7, terminal.getCapacidade_movimentacao_granel_liquido());
            pst.setString(8, terminal.getCapacidade_movimentacao_carga_geral());
```

```

        pst.setString(9, terminal.getCapacidade_movimentacao_contenedor());
        pst.executeUpdate();
    } catch (SQLException e) {
        throw new SitmmcException("Ocorreu um erro durante a inclusão solicitada. Informe ao
administrador do sistema a seguinte mensagem: "+ e.getMessage());
    } finally {
        db.freeConn();
    }
}

```

```

public void alterar(Terminal terminal) throws SitmmcException {
    try {
        con = db.getConn();
        pst = con.prepareStatement("update terminal set tipo_terminal=?, " +
            " nome_terminal=?, endereco_terminal=?, capacidade_acesso=?, " +
            " velocidade_acesso=?, capacidade_movimentacao_granel_solido=?, " +
            " capacidade_movimentacao_granel_liquido=?,
capacidade_movimentacao_carga_geral=?, " +
            " capacidade_movimentacao_contenedor=? where codigo_terminal=?");
        pst.setString(1, terminal.getTipo_terminal());
        pst.setString(2, terminal.getNome_terminal());
        pst.setString(3, terminal.getEndereco_terminal());
        pst.setString(4, terminal.getCapacidade_acesso());
        pst.setString(5, terminal.getVelocidade_acesso());
        pst.setString(6, terminal.getCapacidade_movimentacao_granel_solido());
        pst.setString(7, terminal.getCapacidade_movimentacao_granel_liquido());
        pst.setString(8, terminal.getCapacidade_movimentacao_carga_geral());
        pst.setString(9, terminal.getCapacidade_movimentacao_contenedor());
        pst.setInt(10, terminal.getCodigo_terminal());
        pst.executeUpdate();
    } catch (SQLException e) {
        throw new SitmmcException("Ocorreu um erro durante a alteração solicitada. Informe ao
administrador do sistema a seguinte mensagem: "+ e.getMessage());
    } finally {
        db.freeConn();
    }
}

```

```

public Terminal consultar(int codigo_terminal) throws SitmmcException {
    Terminal terminal = null;
    try {
        con = db.getConn();
        sql = "select * from terminal where codigo_terminal=" + codigo_terminal;
        st = con.createStatement();
        rs = st.executeQuery(sql);
        if (rs.next()) {
            terminal = new Terminal(rs.getInt("codigo_terminal"), FormatText.cleanNull(rs.getString

```

```

("tipo_terminal"),FormatText.cleanNull(rs.getString("nome_terminal")),
FormatText.cleanNull(rs.getString("endereco_terminal")),FormatText.cleanNull(rs.getString
("capacidade_aceso")),FormatText.cleanNull(rs.getString("velocidade_aceso")),
FormatText.cleanNull(rs.getString("capacidade_movimentacao_granel_solido")),
FormatText.cleanNull(rs.getString("capacidade_movimentacao_granel_liquido")),
FormatText.cleanNull(rs.getString("capacidade_movimentacao_carga_geral")),
FormatText.cleanNull(rs.getString("capacidade_movimentacao_containeer")));
    }
    }catch(SQLException e){
        throw new SitmmcException("Ocorreu um erro durante a consulta solicitada. Informe ao
administrador do sistema a seguinte mensagem: "+ e.getMessage());
    }finally{
        db.freeConn();
    }
    return terminal;
}

```

```

public Collection consultarterminais(String nome_terminal) throws
SitmmcException {
    ArrayList terminals = new ArrayList();
    try{
        con = db.getConn();
        if(nome_terminal.equals("")){
            sql = "select * from terminal order by nome_terminal";
        }else{
            sql = "select * from terminal where UPPER(nome_terminal) like UPPER('%" +
nome_terminal + "%') order by nome_terminal";
        }
        st = con.createStatement();
        rs = st.executeQuery(sql);
        Terminal terminal = null;
        while (rs.next()){
            terminal = new Terminal(rs.getInt("codigo_terminal"),FormatText.cleanNull(rs.getString
("tipo_terminal"),FormatText.cleanNull(rs.getString("nome_terminal")),
FormatText.cleanNull(rs.getString("endereco_terminal")),FormatText.cleanNull(rs.getString
("capacidade_aceso")),FormatText.cleanNull(rs.getString("velocidade_aceso")),
FormatText.cleanNull(rs.getString("capacidade_movimentacao_granel_solido")),
FormatText.cleanNull(rs.getString("capacidade_movimentacao_granel_liquido")),
FormatText.cleanNull(rs.getString("capacidade_movimentacao_carga_geral")),
FormatText.cleanNull(rs.getString("capacidade_movimentacao_containeer")));
            terminals.add(terminal);
        }
    }catch(SQLException e){
        throw new SitmmcException("Ocorreu um erro durante a consulta solicitada. Informe ao
administrador do sistema a seguinte mensagem: "+ e.getMessage());
    }finally{
        db.freeConn();
    }
}

```

```

    }
    return terminals;
}

public void excluir(int codigo_terminal) throws SitmmcException{
    try{
        con = db.getConn();
        pst = con.prepareStatement("delete from terminal where codigo_terminal=?");
        pst.setInt(1,codigo_terminal);
        pst.executeUpdate();
    }catch(SQLException e){
        throw new SitmmcException("Ocorreu um erro durante a exclusão solicitada. Informe ao
administrador do sistema a seguinte mensagem: "+ e.getMessage());
    }finally{
        db.freeConn();
    }
}
}
}
}

```

3. VALUE OBJECT (VO)

Terminal.java

```

package sitmmc.vo;

/**
 * Title: Sistema de Informação Para Transporte Multimodal de Carga
 */

public class Terminal {
    private int codigo_terminal;
    private String tipo_terminal;
    private String nome_terminal;
    private String endereco_terminal;
    private String capacidade_aceso;
    private String velocidade_aceso;
    private String capacidade_movimentacao_granel_solido;
    private String capacidade_movimentacao_granel_liquido;
    private String capacidade_movimentacao_carga_geral;
    private String capacidade_movimentacao_container;

    public Terminal() {
    }
}

```

```

public Terminal(int codigo_terminal, String tipo_terminal, String nome_terminal, String
endereco_terminal, String capacidade_aceso, String velocidade_aceso, String
capacidade_movimentacao_granel_solido, String capacidade_movimentacao_granel_liquido,
String capacidade_movimentacao_carga_geral, String capacidade_movimentacao_container)
{
    this.codigo_terminal=codigo_terminal;
    this.tipo_terminal=tipo_terminal;
    this.nome_terminal=nome_terminal;
    this.endereco_terminal=endereco_terminal;
    this.capacidade_aceso=capacidade_aceso;
    this.velocidade_aceso=velocidade_aceso;
    this.capacidade_movimentacao_granel_solido=capacidade_movimentacao_granel_solido;
    this.capacidade_movimentacao_granel_liquido=capacidade_movimentacao_granel_liquid
o;
    this.capacidade_movimentacao_carga_geral=capacidade_movimentacao_carga_geral;
    this.capacidade_movimentacao_container=capacidade_movimentacao_container;
}
public int getCodigo_terminal(){
    return this.codigo_terminal;
}
public void setCodigo_terminal(int codigo_terminal){
    this.codigo_terminal = codigo_terminal;
}
public void setTipo_terminal(String tipo_terminal){
    this.tipo_terminal = tipo_terminal;
}
public String getTipo_terminal(){
    return this.tipo_terminal;
}
public void setNome_terminal(String nome_terminal){
    this.nome_terminal = nome_terminal;
}
public String getNome_terminal(){
    return this.nome_terminal;
}
public void setEndereco_terminal(String endereco_terminal){
    this.endereco_terminal = endereco_terminal;
}
public String getEndereco_terminal(){
    return this.endereco_terminal;
}
public void setCapacidade_aceso(String capacidade_aceso){
    this.capacidade_aceso = capacidade_aceso;
}
public String getCapacidade_aceso(){
    return this.capacidade_aceso;
}
}

```

```

public void setVelocidade_aceso(String velocidade_aceso){
    this.velocidade_aceso = velocidade_aceso;
}
public String getVelocidade_aceso(){
    return this.velocidade_aceso;
}
public void setCapacidade_movimentacao_granel_solido(String
capacidade_movimentacao_granel_solido){
    this.capacidade_movimentacao_granel_solido =
capacidade_movimentacao_granel_solido;
}
public String getCapacidade_movimentacao_granel_solido(){
    return this.capacidade_movimentacao_granel_solido;
}
public void setCapacidade_movimentacao_granel_liquido(String
capacidade_movimentacao_granel_liquido){
    this.capacidade_movimentacao_granel_liquido =
capacidade_movimentacao_granel_liquido;
}
public String getCapacidade_movimentacao_granel_liquido(){
    return this.capacidade_movimentacao_granel_liquido;
}
public void setCapacidade_movimentacao_carga_geral(String
capacidade_movimentacao_carga_geral){
    this.capacidade_movimentacao_carga_geral =
capacidade_movimentacao_carga_geral;
}
public String getCapacidade_movimentacao_carga_geral(){
    return this.capacidade_movimentacao_carga_geral;
}
public void setCapacidade_movimentacao_container(String
capacidade_movimentacao_container){
    this.capacidade_movimentacao_container = capacidade_movimentacao_container;
}
public String getCapacidade_movimentacao_container(){
    return this.capacidade_movimentacao_container;
}
}
}

```

7.5 APÊNDICE 5: CÓDIGOS DE OPERAÇÕES COM O MAPSERVER

A) GIS.HTML

```
<html>
<head><title>MapServer</title></head>
<body bgcolor=#ffffff>

<center><h2>Interface do MapServer</h2></center>
<form method=GET action="[program]">

<center>
<table border=1 cellspacing=0 cellpadding=4 bgcolor="#ffffff">
<tr>
<td valign="top" align=center>
<table cellpadding="0" cellspacing="0" border="0">
<tr><td colspan="2"><INPUT NAME="img" TYPE="image" SRC="[img]" width=300
height=300 border=0></td></tr>
<tr>
<td align="right"></td>
</tr>
</table>
</td>
<td valign="top" bgcolor=#ffffff>
<table cellpadding="5" cellspacing="0" border="0" bgcolor="#ffffff">
<tr><td>
<center><input type="submit" value="Atualizar/Consultar"></center>
<p>

<input type="radio" name="mode" value="browse" checked> <b>Navegar</b>
<input type="radio" name="mode" value="query"> <b>Feicao</b>
<input type="radio" name="mode" value="query"> <b>Feicoes</b>

<hr>

<p>
<b>Selecione as Camadas: </b><br>
<select multiple name="layer" size=4>
<option value="contry92" [contry92_select]> Países
<option value="brasilpl" [brasilpl_select]> Brasil
<option value="estadospl" [estadospl_select]> Estados Brasileiros
<option value="munipl1" [munipl1_select]> Municipios
<option value="riosln" [riosln_select]> Rios
<option value="rodoviasln" [rodoviasln_select]> Rodovias
<option value="ferroviasln" [ferroviasln_select]> Ferrovias
```

```

    <option value="contry92a" [contry92a_select]> Visualizar Maiores Areas
    <option value="areas" [areas_select]> Consultar Maiores Areas
    <option value="municons" [municons_select]> Munic.Fronteira (CE/PI)
</select>

<p><center><font size=4><b>Zoom</b></font><br>
  <input type=radio name=zoomdir value=1 [zoomdir_1_check]> Ampliar
  <input type=radio name=zoomdir value=0 [zoomdir_0_check]> Deslocar
  <input type=radio name=zoomdir value=-1 [zoomdir_-1_check]> Reduzir
</center><p>
Tamanho do Zoom <input type=text name=zoomsize size=2 value=[zoomsize]>
</center><p>
</td></tr>
</table>
</td>
<td>
  <table cellpadding="5" cellspacing="0" border="0" bgcolor="#ffffff">
    <tr>
      <td>
        <font size=+1><b>Legenda:</b></font><br><hr>
        <hr>
        <p>
          <center><INPUT NAME="ref" TYPE="image" SRC="[ref]"
border="0"></center>
        </td>
      </tr>
    </table>
  </td>
</table>
</center>
<input type="hidden" name="imgxy" value="149.5 149.5"> <!-- metade da resolucao -0.5 -->
<input type="hidden" name="imgext" value="[mapext]">
<input type="hidden" name="map" value="[map]">
<input type="hidden" name="savequery" value="true">

<input type="hidden" name="program" value="[program]">
<input type="hidden" name="map_web_imagepath" value="[map_web_imagepath]">
<input type="hidden" name="map_web_imageurl" value="[map_web_imageurl]">

</form>

</body></html>

```

B) MAPFILE (GIS.MAP)

```
# Start of map file
#
NAME GIS
STATUS ON
SIZE 300 300 # resolucao do mapa
EXTENT -74.33 -34.53 -33.9 5.69 # coordenada espacial
# para enquadramento do mapa - consultar no jump-
UNITS KILOMETERS
FONTSET "/srv/www/default/html/gis/font.list"
SHAPEPATH "/srv/www/default/html/gis/data"
IMAGECOLOR 0 254 255 # pano de fundo do mapa verificar cores no gimp

# you may need to change this to match your MapServer build
IMAGETYPE JPG # tipo de imagem que sera gerada

# Projection definition, consult the PROJ.4 documentation for parameter discussion
#

PROJECTION
  "init=epsg:4326" #codificacao epsg que pode ser verificada ,, no caso do Rio SUL 23
# endereço http://mapserver.gis.umn.edu/doc36/proj.html SAD69/UTM zone 23 sul
# cod 29183
END

# Start of web interface definition (including WMS enabling metadata)
#
WEB
  HEADER gis_header.html # cabeçalho
  TEMPLATE gis.html # pagina que disponibiliza a interface
  FOOTER gis_footer.html # rodape
  IMAGEPATH "/srv/www/default/html/temp/" # onde estao sendo gravados os arquivos
temporarios
  IMAGEURL "http://localhost/temp/"
  METADATA
    WMS_TITLE "UMN MapServer gis Demo"
    WMS_ABSTRACT "This is the UMN MapServer application for Brazil."

    # change this value to match your setup
    WMS_SRS "EPSG:4326"
  END
END

QUERYMAP # Mapa de consulta
  SIZE 300 300
```

```

STATUS ON
STYLE HILITE # destaca a feica em estudo
COLOR 255 0 0 # cor de destaque do hilite
END

# Start of reference map (Mapa de Referencia)
#
REFERENCE # mapa do canto inferior direito
IMAGE graphics/gis.jpg # onde esta o mapa de referÃancia

# EXTENT 393234.393701263 5205405.16440722 495769.579718949 5307959.02579127
EXTENT -74.33 -34.53 -33.9 5.69
SIZE 120 120
STATUS ON
COLOR -1 -1 -1 # cor, nesse caso, transparente
OUTLINECOLOR 255 0 0 # cor do retangulo
END

# Start of legend
#
LEGEND
KEYSIZE 18 12 #Tamanho dos simbolos
LABEL # Objeto que esta dentro do objeto legend
TYPE BITMAP # Tipo da legenda
SIZE MEDIUM # Tamanho da legenda --manual do mapserver
COLOR 0 0 89 # Cor das letras
END
STATUS ON

END

# Start of scalebar
#
SCALEBAR # BARRA DE ESCALA
IMAGECOLOR 255 255 255 # PANO DE FUNDO
LABEL
COLOR 0 0 0
SIZE MEDIUM
END
STYLE 1
SIZE 200 2
COLOR 255 0 0
UNITS KILOMETERS
INTERVALS 3
TRANSPARENT FALSE
STATUS TRUE
END

```

```

# Start of symbol definitions (we're only using a few)
#
SYMBOL
  NAME 'circle'
  TYPE ELLIPSE
  POINTS 1 1 END
  FILLED TRUE
END

# Start of layer definitions
#

LAYER
  NAME contry92
  TYPE POLYGON
  STATUS DEFAULT
  CONNECTIONTYPE postgis
  CONNECTION "user=postgres dbname=gis host=localhost"
  DATA "the_geom from paises"

  CLASS # every other county in the state
  NAME 'Paises'
  OUTLINECOLOR 128 128 128
  COLOR 102 204 0
  END

  PROJECTION
    "init=epsg:4326"
    #codificacao epsg que pode ser verificada
  END
END

LAYER
  NAME estadospl
  TYPE POLYGON
  STATUS OFF
  CONNECTIONTYPE postgis
  CONNECTION "user=postgres dbname=gis host=localhost"
  DATA "the_geom from estados"

  LABELITEM "UF"

  CLASS #every other county in the state
  NAME 'Estados do Brasil'
  OUTLINECOLOR 255 255 255
  COLOR 102 200 150
  LABEL

```

```

        COLOR 0 0 0
        TYPE TRUETYPE
        FONT vera_mobd
        SIZE 6
        ANTIALIAS TRUE
        POSITION CR
        PARTIALS TRUE
    END
END
PROJECTION
    "init=epsg:4326"
END
END

LAYER
NAME munipl1
TYPE POLYGON
STATUS OFF
CONNECTIONTYPE postgis
CONNECTION "user=postgres dbname=gis host=localhost"
DATA "the_geom from municipios"

CLASS # every other county in the state
NAME 'Municipios'
OUTLINECOLOR 128 128 128
COLOR 255 255 204
END

PROJECTION
    "init=epsg:4326"
END
END

LAYER
NAME riosln
TYPE LINE
STATUS OFF
CONNECTIONTYPE postgis
CONNECTION "user=postgres dbname=gis host=localhost"
DATA "the_geom from rios"

CLASS # every other county in the state
NAME 'Rios'
OUTLINECOLOR 128 128 128
COLOR 102 153 255
END
PROJECTION

```

```

    "init=epsg:4326"
  END
END

LAYER
  NAME rodoviasln
  TYPE LINE
  STATUS OFF
  CONNECTIONTYPE postgis
  CONNECTION "user=postgres dbname=gis host=localhost"
  DATA "the_geom from rodovias"

  CLASS # every other county in the state
    NAME 'Rodovias'
    OUTLINECOLOR 128 128 128
    COLOR 255 0 51
  END

  PROJECTION
    "init=epsg:4326"
  END
END

LAYER
  NAME ferroviasln
  TYPE LINE
  STATUS OFF
  CONNECTIONTYPE postgis
  CONNECTION "user=postgres dbname=gis host=localhost"
  DATA "the_geom from ferrovias"

  CLASS # every other county in the state
    NAME 'Ferrovias'
    OUTLINECOLOR 128 128 128
    COLOR 255 204 204
  END

  PROJECTION
    "init=epsg:4326"
  END
END

END # Map File

```

C) INDEX.HTML

```
<html>
<head><title>MapServer</title></head>
<body bgcolor="#FFFFFF">

<center><h2>Interface do MapServer</h2></center>
<p><hr><p>
<center>
SITMMC<br>
IME - Engenharia de Transportes <br>
</center>
<p><hr><P>

<form method=GET action="/cgi-bin/mapserv">

<input type="hidden" name="map" value="/srv/www/default/html/gis/gis.map">
<input type="hidden" name="layer" value="contry92">
<input type="hidden" name="zoomsize" value=2>
<input type="hidden" name="program" value="/cgi-bin/mapserv">
<input type="hidden" name="map_web_imagepath" value="/srv/www/default/html/temp/">
<input type="hidden" name="map_web_imageurl" value="http://localhost/temp/">
<center><input type="submit" value="Inicializar"></center>
</form>
</body></html>
```

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)