



**UNIVERSIDADE METODISTA DE PIRACICABA**  
**FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA**  
**MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**

**REALIDADE VIRTUAL APLICADA À  
VISUALIZAÇÃO DE INFORMAÇÕES  
VIA WORLD WIDE WEB**

**CARLOS EDUARDO DELLA COLETTA**

**ORIENTADOR: PROF. DR. NIVALDI CALONEGO JUNIOR**

**PIRACICABA, SP**  
**2006**



**UNIVERSIDADE METODISTA DE PIRACICABA**  
**FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA**  
**MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**

**REALIDADE VIRTUAL APLICADA À  
VISUALIZAÇÃO DE INFORMAÇÕES  
VIA WORLD WIDE WEB**

**CARLOS EDUARDO DELLA COLETTA**

**ORIENTADOR: PROF. DR. NIVALDI CALONEGO JUNIOR**

Dissertação apresentada ao Mestrado em Ciência da Computação, da Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP, como requisito para obtenção do Título de Mestre em Ciência da Computação.

**PIRACICABA, SP**  
**2006**

Della Coletta, Carlos Eduardo  
Realidade Virtual Aplicada à Visualização de Informações  
Via World Wide Web. Piracicaba, 2006.  
168p.

Orientador: Prof. Dr. Nivaldi Calonego Junor  
Dissertação (mestrado) – Programa de Pós-Graduação em Ciência da  
Computação – Universidade Metodista de Piracicaba

1-Realidade Virtual. 2- Visualização de Informações.

**REALIDADE VIRTUAL APLICADA À  
VISUALIZAÇÃO DE INFORMAÇÕES  
VIA WORLD WIDE WEB**

AUTOR: CARLOS EDUARDO DELLA COLETTA

ORIENTADOR: NIVALDI CALONEGO JUNIOR

Dissertação de Mestrado defendida e aprovada em 09 de março de 2006, pela  
Banca Examinadora constituída dos Professores:

Prof. Dr. Nivaldi Calonego Junior – UNIMEP (Orientador)

Profa. Dra. Regina Célia Coelho – UNIMEP

Prof. Dr. João Eduardo Ferreira - IME/USP

*Aos*

*Meus pais Carlos Roberto e Maria Rita pelo  
apoio e compreensão*

*Aos*

*Meus irmãos Cláudia e Ismael*

## **AGRADECIMENTOS**

Primeiramente, a Deus, que me criou, me amou e me preservou, em vitória, até este momento.

Ao meu amigo Husney Walter Daruge, pelas palavras de encorajamento, pela leitura crítica do trabalho e pelas sugestões no decorrer da escrita do mesmo.

À meu orientador, Prof. Dr. Nivaldi Calonego Junior, por ter acreditado em mim, ter me ajudado no decorrer deste trabalho, ter me exortado e incentivado, nas horas e medidas certas.

Aos professores do PPGCC da UNIMEP, por terem repassado os seus conhecimentos e experiências e pelo incentivo à busca do saber e do aprender a aprender.

À Brasil Ferrovias S.A, pela disponibilização das informações que foram necessárias para construção deste trabalho e obtenção deste título.

À amiga e irmã em Cristo, Edilaine Cristina Lisardo, pelo incentivo, apoio e encorajamentos em todos os momentos deste curso.

Finalmente e destacadamente, aos meus pais Carlos Roberto Della Coletta e Maria Rita Rodrigues Della Coletta pelo amor, paciência, compreensão, pelo incentivo e valorização dos meus estudos e suporte nos momentos de desânimo, tensão e cansaço, de quem foram subtraídos preciosos períodos de convivência.

“Ora, a fé é a certeza daquilo que esperamos e a prova das coisas que não vemos. Pois foi por meio dela que os antigos receberam bom testemunho. Pela fé entendemos que o universo foi formado pela palavra de Deus, de modo que aquilo que se vê não foi feito do que é visível.

Sem fé é impossível agradar a Deus, pois quem dEle aproxima precisa crer que Ele existe e que recompensa aqueles que o buscam.”

---

---

## RESUMO

O objetivo da pesquisa é a aplicação da Realidade Virtual na melhoria dos serviços oferecidos pela empresa Brasil Ferrovias S.A.. Esta empresa é uma *holding* que oferece serviços de transporte de cargas nas regiões sul, sudeste, oeste e centro-oeste, com sede na região sudeste, atendendo dezenas de outras empresas. Atualmente, ela não oferece serviço de acesso a dados *on-line* ou via Internet para os seus clientes, e os dados de seus relatórios encerram pouco tratamento no sentido de facilitar a interpretação. A pesquisa melhorou esses serviços com o uso de técnicas de desenvolvimento de aplicações para a Realidade Virtual e para a WEB, abrangendo a geração de páginas WEB geradas dinamicamente, sob demanda, a partir de dados armazenados no banco de dados da empresa. As solicitações dos clientes e funcionários da empresa podem gerar documentos em forma de tabela ou em forma de gráficos dinâmicos modelados em *Virtual Reality Modeling Language* (VRML). O texto VRML caracteriza um modelo de interface no qual os dados não são transportados, mas as informações a respeito desses dados. Isto permite a visualização de tais informações com dispositivos de Realidade Virtual, melhorando a interface com os usuários da empresa.

PALAVRAS-CHAVE: Realidade Virtual, Visualização de Informações, VRML e Web

---

---

---

---

## **REALIDADE VIRTUAL APLICADA À VISUALIZAÇÃO DE INFORMAÇÕES VIA WORLD WIDE WEB**

### ***ABSTRACT***

The aim of this research is the introduction of Virtual Reality for the improvement of services provided by Brasil Ferrovias S.A. This is a Holding Co. that provides cargo shipping services in the south, southeast, west and midwest of Brazil. Its head office is based in the country's southeast, and they have deals with many other companies. Currently it hasn't provided its clients with on-line data services or via Internet and its reports present few sense of facilitating the interpretation. This research improved of these services by applying development techniques for Virtual Reality and for the WEB, including the creation of WEB pages under demand, which can be done based on the company's stored data. The requests of the customers and employees of the company can generate documents in chart form or in form of dynamic graphs modeled in Virtual Reality Modeling Language (VRML). The VRML text features an interface model in which data are not sent; instead, the information about them. This will lead to the visualization of such information through Virtual Reality features and therefore will improve the interface with the users company.

**KEYWORDS:** Virtual Reality, Information Visualization, VRML e Web

---

---

## SUMÁRIO

LISTA DE FIGURAS .....	X
LISTA DE TABELAS.....	XI
LISTA DE ABREVIATURAS E SIGLAS .....	XII
<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1. PARQUE COMPUTACIONAL .....	5
1.2. SOFTWARE.....	8
1.3. ORGANIZAÇÃO DO TRABALHO .....	10
<b>2. BASES TEÓRICAS: ACESSO E VISUALIZAÇÃO DE INFORMAÇÕES REMOTAS ....</b>	<b>11</b>
2.1. INTERNET NO CONTEXTO DA EMPRESA.....	14
2.1.1. INTRANET .....	17
2.1.2. EXTRANET .....	20
2.2. ELEMENTOS DA APLICAÇÃO.....	21
2.2.1. O BANCO DE DADOS .....	21
2.2.1.1. STORED PROCEDURES.....	26
2.2.1.2. PACKAGES.....	28
2.2.2. CONEXÃO DE BANCO DE DADOS VIA OPEN DATABASE CONNECTIVITY.....	29
2.2.3. TECNOLOGIA DE DESENVOLVIMENTO E ACESSO A BANCO DE DADOS VIA WEB .....	32
2.2.3.1. ACTIVE SERVER PAGE - ASP .....	38
2.3. REALIDADE VIRTUAL .....	40
2.4. VISUALIZAÇÃO DE DADOS.....	43
2.4.1. CONSTRUÇÃO DE VISUALIZAÇÃO DE INFORMAÇÕES .....	45
2.4.2. VISUALIZAÇÃO DE INFORMAÇÕES.....	46
2.4.3. REPRESENTAÇÃO DE DADOS .....	47
2.4.4. MODELAGEM DE CENAS.....	53
2.4.5. CENAS DINÂMICAS .....	56
2.4.6. GRÁFICOS DINÂMICOS .....	61
<b>3. SISTEMA DE CONSULTA DE LOCALIZAÇÃO DE CARGAS FERROVIÁRIAS .....</b>	<b>64</b>
3.1. MODELO DO PROTÓTIPO.....	65
3.2. PLATAFORMA DE DESENVOLVIMENTO.....	74
3.3. RESULTADOS OBTIDOS.....	75
<b>4. CONSIDERAÇÕES FINAIS .....</b>	<b>79</b>
<b>5. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>81</b>
<b>ANEXO A – EXEMPLO DE SCRIPT EM VRML PARA GERAÇÃO DE GRÁFICOS .....</b>	<b>86</b>
<b>ANEXO B – MODELO DOS DADOS ATUAIS.....</b>	<b>139</b>
<b>ANEXO C – PROGRAMA FONTE JAVA.....</b>	<b>146</b>
<b>ANEXO D – STORED PROCEDURES .....</b>	<b>148</b>
<b>ANEXO E – PROGRAMA EM ASP CRIANDO UM ARQUIVO VRML .....</b>	<b>154</b>

## LISTA DE FIGURAS

FIGURA 1.1 - FATURAMENTO ANUAL (REAIS MIL) .....	2
FIGURA 1.2 - RELACIONAMENTO SIGEFER x E.R.P. – PROTHEUS NA BRASIL FERROVIAS S.A. ....	9
FIGURA 2.1 - EXEMPLO DE PILHA DE PROTOCOLOS TCP/IP (ÖZSU, 2001; P.76).....	11
FIGURA 2.2 - ESTRUTURA ACESSO INTERNET, INTRANET E EXTRANET .....	15
FIGURA 2.3 - ESTRUTURA DE UMA INTRANET.....	18
FIGURA 2.4 - ESTRUTURA SIMPLES DE SISTEMA DE BANCO DE DADOS (ELMASRI,2000;P.3).....	22
FIGURA 2.6 - ACESSO BANCO DE DADOS .....	25
FIGURA 2.8 - ARQUITETURA ODBC API (MICROSOFT, 2003B).....	30
FIGURA 2.9 - EXEMPLO DE REGISTRO DO B.D. ORACLE COMO FONTE DE DADOS DO SISTEMA .....	31
FIGURA 2.10 - TECNOLOGIAS DE PROCESSAMENTO USADAS NA CONSTRUÇÃO DE WEB SITE .....	33
FIGURA 2.11 - EXEMPLO DE PROGRAMA DE SERVIDOR COMPILADO – JAVA SERVLET .....	35
FIGURA 2.12 - EXEMPLO DE SCRIPT NO LADO DO SERVIDOR.....	36
FIGURA 2.13 - EXEMPLO DE PROGRAMA COMPILADO NO LADO DO CLIENTE – JAVA APPLET .....	37
FIGURA 2.14: ARQUITETURA ACTIVE SERVE PAGE (COPELAND,2000;P.23).....	40
FIGURA 2.15 - PROCESSO MAPEAMENTO VISUAL DE DADOS (CARD, 1999).....	45
FIGURA 2.16- REPRESENTAÇÃO GRÁFICA DE VENDA DE PRODUTOS NUM DIA .....	48
FIGURA 2.17- REPRESENTAÇÃO GRÁFICA DE VENDA DE PRODUTOS DE ALGUNS DIAS .....	49
FIGURA 2.18- REPRESENTAÇÃO DE HIPERTEXTO VENDA DE PRODUTOS NOS DIAS 11, 12, 13 .....	49
FIGURA 2.19 - GRÁFICO DE QUANTIDADE DE CHAMADOS.....	50
FIGURA 2.20- GRÁFICO QUANTIDADE DE CALORIAS ENERGÉTICAS DE ALIMENTOS .....	51
FIGURA 2.21- GRÁFICO PERFORMANCE DE VENDAS DE UMA EMPRESA.....	52
FIGURA 2.23 - EXEMPLO DE UM MODELO DE REPRESENTAÇÃO DE MUNDOS VIRTUAIS.....	53
FIGURA 2.24 - DINÂMICA DE OPERAÇÃO DO NAVEGADOR COM UTILIZAÇÃO DE PLUG-IN.....	55
FIGURA 2.25 - GRÁFICO 3D COM VRML FAZENDO ACESSO A JAVA .....	63
FIGURA 3.1 - MODELO LÓGICO PARA DESENVOLVIMENTO DO SISTEMA.....	65
FIGURA 3.2 - DIAGRAMA DE INTERAÇÃO – CONSULTA DE LOCALIZAÇÃO DE CARGAS TRANSPORTADAS .....	67
FIGURA 3.3 - DIAGRAMA ENTIDADE RELACIONAMENTO – CONSULTA DE CARGAS .....	70
FIGURA 3.4 – TELA INICIAL DA APLICAÇÃO .....	75
FIGURA 3.5 – TELA DE ESCOLHA DE MERCADORIA E ESTAÇÃO .....	76
FIGURA 3.6 – VISUALIZAÇÃO DE GRÁFICO EM VRML VIA PLUGIN CORTONA .....	77
FIGURA 3.7 – VISUALIZAÇÃO ATRAVÉ DE MAPA EM VRML VIA PLUGIN CORTONA .....	78

## LISTA DE TABELAS

TABELA 1.1 - MACRO EXCEL QUE IMPORTA AS INFORMAÇÕES DO BANCO DE DADOS ORACLE.....	4
TABELA 1.2 - SERVIDORES DA BRASIL FERROVIAS S.A.....	6
TABELA 1.3 - EXEMPLOS DE ESTAÇÕES DE TRABALHO DA BRASIL FERROVIAS S.A. ....	8
TABELA 2.1 – ESTRUTURA SINTÁTICA DE UM STORED PROCEDURE NO ORACLE .....	27
TABELA 2.2 – ESTRUTURA SINTÁTICA DE UM PACKAGE NO ORACLE .....	29
TABELA 2.3 - NÍVEIS DE IMERSÃO EM REALIDADE VIRTUAL.....	42
TABELA 2.4 – REPRESENTAÇÃO TEXTUAL DE VENDA DE PRODUTOS A.....	47
TABELA 2.5 – REPRESENTAÇÃO TABULAR DE VENDA DE PRODUTOS EM UM DIA .....	47
TABELA 2.6 – EXEMPLO DE UTILIZAÇÃO DE USE E DEF EM VRML .....	57
TABELA 2.7 – SINTAXE DO NÓ TIMESENSOR .....	58
TABELA 2.8 - CAMPOS DO NÓ TIMESENSOR .....	58
TABELA 2.9 – SINTAXE DO NÓ COLORINTERPOLATOR .....	59
TABELA 2.10 – SINTAXE DO NÓ COORDINATEINTERPOLATOR.....	59
TABELA 2.11 – SINTAXE DO NÓ NORMALINTERPOLATOR.....	59
TABELA 2.12 – SINTAXE DO NÓ ORIENTATIONINTERPOLATOR .....	59
TABELA 2.13 – SINTAXE DO NÓ POSITIONINTERPOLATOR .....	60
TABELA 2.14 – SINTAXE DO NÓ SCALARINTERPOLATOR.....	60
TABELA 2.15 – SINTAXE DO NÓ SCRIPT .....	61
TABELA 2.16 – CÓDIGO VRML ACESSANDO O PROGRAMA EM JAVA .....	62
TABELA 2.17 – CÓDIGO FONTE JAVA.....	63
TABELA 3.1 – CONEXÃO DE PÁGINA ASP COM BANCO DE DADOS ORACLE.....	71
TABELA 3.2 – STORED PROCEDURE NO BANCO DE DADOS ORACLE.....	71
TABELA 3.3 – CHAMADA DE STORED PROCEDURE DO BANCO DE DADOS ORACLE ATRAVÉS DE ASP .....	72
TABELA 3.4 – EXEMPLO DE CÓDIGO ASP CRIANDO UM ARQUIVO VRML (TRECHO DO PROGRAMA). 73	
TABELA 3.5 – EXEMPLO DE CÓDIGO ASP EXECUTANDO UM ARQUIVO VRML .....	73

## **LISTA DE ABREVIATURAS E SIGLAS**

ACT – AUTOMAÇÃO E CIRCULAÇÃO DE TRENS

ADS – ADVANTAGE DATABASE SERVER

B2B – BUSINESS TO BUSINESS

B2C – BUSINESS TO CONSUMER

BFDBF – NOME DA INSTÂNCIA DE PRODUÇÃO ORACLE (BF – BRASIL FERROVIAS)

C2C – CONSUMER TO CONSUMER

CCO – CENTRO CONTROLE OPERACIONAL

CGI – COMMON GATEWAY INTERFACE

D.E.R – DIAGRAMA ENTIDADE-RELACIONAMENTO

E.D.I – ELETRONIC DATA INTERCHANGE

E.R.P – ENTERPRISE RESOURCE PLANNING

FNDBD – NOME DA INSTÂNCIA DE DESENVOLVIMENTO ORACLE (FN – FERRONORTE)

FNDBP – NOME DA INSTÂNCIA DE PRODUÇÃO ORACLE (FN – FERRONORTE)

FTP – FILE TRANSFER PROTOCOL

GPS – GRAPHICAL POSITION SAT

IP – INTERNET PROTOCOL

M.E.R – MODELO ENTIDADE RELACIONAMENTO

ODBC – OPEN DATABASE CONNECTIVITY

SIGEFER – SISTEMA DE GERENCIAMENTO FERROVIÁRIO

SISAO – SISTEMA DE ACOMPANHAMENTO ORÇAMENTÁRIO

TCP – TRANSFER CONTROL PROTOCOL

TRANSFAT – SISTEMA DE TRANSFERÊNCIA DE FATURAMENTO

## 1. INTRODUÇÃO

A Realidade Virtual oferece meios para o tratamento e visualização de informação que facilitam a interface homem-computador. Essa premissa orientou a investigação da possibilidade de aplicar a Realidade Virtual ao ambiente comercial da empresa Brasil Ferrovias S.A., objetivando melhorar a apresentação e o processo de geração de relatórios administrativos utilizados por clientes ou pela gerência da empresa.

A viabilidade da modificação desse processo exigiu o levantamento detalhado da infra-estrutura e dos procedimentos atuais, resultando na contextualização da empresa. A Brasil Ferrovias S.A. é uma *holding* que atua no setor de transporte ferroviário, e controla as empresas Ferrobán – Ferrovias Bandeirantes S.A., Ferronorte S.A. – Ferrovias Norte Brasil, Ferrovia Novoeste e PortoFer. Essa *holding* oferece cerca de 3000 (três mil) empregos diretos, segundo a gerência de recursos humanos, distribuídos na sede (localizada na cidade de Campinas) e em filiais nos estados São Paulo, Mato Grosso e Mato Grosso do Sul.

A Brasil Ferrovias S.A. faz o transporte de produtos como aço, açúcar, adubo, alumínio, amônia, areia, arroz, bauxita, cal, carvão, cimento, cítricos, clínquer, contêineres, coque, diesel, farelo, feijão, ferro, fosfato, gasolina, madeira, milho, minério, óleo vegetal, óleos combustíveis, papel, soja e trigo. Essa diversidade de possibilidades de transporte de produtos é o resultado da política da empresa, e é sintetizada por “(...) quanto maior a diversidade de produtos, menor a dependência da empresa em relação a sazonalidade, provendo mais estabilidade do fluxo de transporte, conseqüentemente fluxo de caixa” (ATOR 1)<sup>1</sup>.

---

<sup>1</sup> Funcionário da Empresa que foi entrevistado

O fluxo de caixa está diretamente vinculado ao faturamento da empresa, conforme ilustra a figura 1.1. Esse montante abrange todas as empresas<sup>2</sup> da holding, apontando um valor de R\$ 687.585 milhões no ano de 2004. Esse volume de transações vem exigindo da empresa investimentos em tecnologia para o controle de tráfego e para melhorar a interação com seus usuários. Mas, esta condição impõe à empresa o desafio de tratar com a questão da dispersão geográfica dos setores produtivos, havendo a necessidade do uso de tecnologia que ofereça sistema de controle automatizado e confiável.

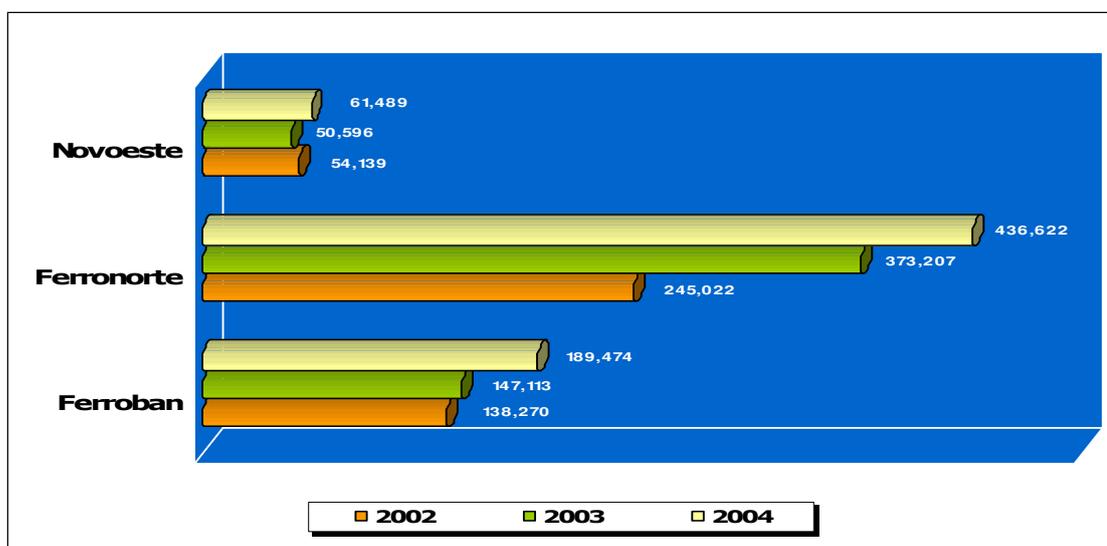


FIGURA 1.1 - FATURAMENTO ANUAL (REAIS MIL)

A *holding* Brasil Ferrovias S.A. possui seu Controle de Tráfego e Faturamento automatizado. Para o controle de tráfego existem dispositivos controladores instalados em locomotivas e vagões, que através do *Global Position System* (GPS) enviam sinais elétrico-eletrônicos ao Centro de Controle de Operações (CCO). No controle de faturamento existem estações de trabalho nas filiais conectados ao sistema, onde é efetuada a emissão do faturamento da empresa. Através desses sistemas o CCO toma decisões sobre o fluxo de trens na empresa e o departamento comercial gera informações para serem

<sup>2</sup> A quarta empresa da holding ("Portofer") não possui faturamento por se tratar de uma empresa sem fins lucrativos.

passadas aos clientes e/ou para serem utilizadas na construção de planilhas, em Microsoft Excel<sup>3</sup>, para o acompanhamento diário de transporte.

Os levantamentos iniciais desta pesquisa mostram que no início do ano de 2002 o envio de relatórios aos clientes demorava em média 8 (oito) horas, devido ao processo não automatizado de geração de formulários produzidos a partir de consultas ao banco de dados produzido no MS-Excel. No entanto havia a necessidade da ação do Departamento de Gerência de Sistemas e Tecnologia da Informação na exportação de dados, somente então se começava a trabalhar os dados. Porém, havia outra dificuldade no Microsoft Excel, que era transformar os dados que vinham em códigos para sua descrição. Com isso, o Departamento Comercial desenvolveu scripts em forma de macros comandos e fórmulas no Microsoft Excel para que o mesmo se tornasse mais rápido, diminuindo de 8 (oito) horas para 1 (uma) hora para o envio do relatório ao cliente. Mas, essas medidas não foram suficientes, dado que os clientes continuaram recebendo o relatório com posição de onde está a carga com defasagem de um dia.

A elaboração dos respectivos relatórios de situação das cargas dos clientes é noturna e consiste de consultas ao servidor de Banco de Dados, que geram uma tabela intermediária a ser utilizada pelo Departamento Comercial. Este utiliza a tabela juntamente com o Microsoft Excel, para a produção de uma planilha. Esta planilha implementa a formatação dos dados para que os usuários os entenda. Há, portanto, que se considerar no contexto da empresa a conexão da planilha com o banco de dados e a conexão da empresa com o cliente. A conexão entre a planilha e o banco de dados, ilustrada na Tabela 1.1, utiliza a interface *Open Database Connectivity* (ODBC) (MICROSOFT, 2003a). Atualmente, a empresa utiliza o servidor de banco de dados Oracle para o gerenciamento de dados da empresa. Essa plataforma foi adotada devido a fatores políticos e administrativos que extrapolam o escopo deste trabalho, e foram colocadas pela empresa como condição invariante no contexto deste trabalho.

---

<sup>3</sup> Microsoft Excel: Marca registrada de programa da empresa Microsoft.

No que concerne ao envio de relatórios aos clientes, a empresa utiliza o serviço de correio eletrônico (e-mail), não implementando um processo completamente automatizado que permita aos seus clientes acesso imediato aos seus dados. A planilha completa contém tabelas que ajustam os dados codificados no formato da Brasil Ferrovias S.A. para descrições inteligíveis por seus clientes, gerando, assim, uma planilha sumarizada para ser enviada através de e-mail ao cliente. Quando o cliente necessita de algo mais detalhado, ele solicita ao Departamento Comercial, que providencia a planilha e reenvia ao cliente.

**TABELA 1.1 - MACRO EXCEL QUE IMPORTA AS INFORMAÇÕES DO BANCO DE DADOS ORACLE**

```

Sub CarregaView()
'Declaração de Variáveis como Objeto
Dim OraSession As Object
Dim OraDatabase As Object
Dim EmpDynaset As Object
Dim flds() As Object
Dim fldcount As Integer

Set OraSession = CreateObject("OracleInProcServer.XOraSession")
Set OraDatabase = OraSession.OpenDatabase("sigefer", "sigefer_cons/conssgf", 0&)
Set EmpDynaset = OraDatabase.CreateDynaset("select * from sigefer.tgf_vagao_carregado", 0&)

Range("A1:H15").Select
Selection.ClearContents

'Declara e Cria os objetos para cada coluna
fldcount = EmpDynaset.Fields.Count
ReDim flds(0 To fldcount - 1)
For Colnum = 0 To fldcount - 1
    Set flds(Colnum) = EmpDynaset.Fields(Colnum)
Next

'Insere Cabeçalho nas Colunas
For Colnum = 0 To EmpDynaset.Fields.Count - 1
    ActiveSheet.Cells(5, Colnum + 1) = flds(Colnum).Name
Next

'Mostra os Dados
For Rownum = 6 To EmpDynaset.RecordCount + 1
    For Colnum = 0 To fldcount - 1
        ActiveSheet.Cells(Rownum, Colnum + 1) = flds(Colnum).Value
    Next
    EmpDynaset.DbMoveNext
Next
Range("A5:A5").Select
End Sub

```

No ano de 2002, o Departamento Comercial solicitou à Gerência de Sistemas e Tecnologia da Informação que desenvolvesse um aplicativo WEB capaz de oferecer à empresa e aos clientes acesso a informações mais precisas e no momento que desejassem. A solicitação prevê que o aplicativo deve permitir consultas *on-line* relativas a fretes e com diferentes possibilidades de visualização dos dados.

Esta nova situação exigiu o levantamento preliminar da infra-estrutura da Brasil Ferrovias S.A. no que concerne aos recursos de informática e distribuição geográfica, objetivando a obtenção de informações que posteriormente refletiram na arquitetura, no modelo e na implementação do software solicitado. Uma sinopse dos levantamentos elaborados durante o segundo semestre de 2002 é apresentada nos sub-itens 1.1 e 1.2, relatando as bases existentes à época e sobre as quais esses novos serviços foram experimentados, detalhando o parque computacional e os softwares instalados para o acesso às informações Empresa.

### **1.1. PARQUE COMPUTACIONAL**

No que tange ao hardware, a Brasil Ferrovias S.A. possui cerca de 490 usuários nomeados na Rede Corporativa, para um total de cerca de 342 Estações de Trabalho, o que corresponde à relação de quase 1,5 (um e meio) usuário por estação. Este índice é considerado superior às medidas do mercado, segundo estatística da FGV-SP (CAMANHO, 2001).

Mais recentemente, a empresa adotou por política a aquisição de produtos de melhor qualidade no segmento de hardware, tanto para Servidores quanto para Estações de Trabalho. Esses produtos oferecem maior confiabilidade em relação a problemas de ordem técnica que ocorrem com componentes de qualidade inferior. Esta atualização abarca servidores, com tecnologia desenvolvida pela IBM para plataforma AS-400, NetFinity (INTEL), RS/6000 F50 e a tecnologia HP.

A Tabela 1.2 sumariza as tecnologias em uso na empresa, apresentado o parque de servidores, nome do servidor, modelo, processador, memória RAM, capacidade do disco rígido, sistema operacional e uma breve descrição de suas responsabilidades (BRASIL FERROVIAS, 2002d).

**TABELA 1.2 - SERVIDORES DA BRASIL FERROVIAS S.A.**

SERVIDOR	PLATAFORMA	PROCESSADOR	RAM	DISCO RÍGIDO	SISTEMA OPERACIONAL	BREVE DESCRIÇÃO
FNENGNT	NET FINITY 3500 M10 – IBM	PENTIUM III - 633	256 MB	8 GB	WINDOWS 2000 SERVER	Domínio de usuários Ferronorte e Novoeste
FNPROTHEUSNT	NET FINITY 3500 M10 – IBM	PENTIUM III – 733 BI-PROCESSADO	1 GB	36 GB 36 GB	WINDOWS 2000 SERVER	Aplicativo E.R.P
FNADSNT	NET FINITY – IBM	PENTIUM III – 633	1 GB	36 GB 36 GB 36 GB 36 GB	WINDOWS 2000 SERVER	Banco de Dados E.R.P
FNENGENHARIANT	NET FINITY – IBM	PENTIUM III – 633	512 MB	09 GB 36 GB	WINDOWS 2000 SERVER	Dados Engenharia
CPS_NTTPRINTER	INTEL	PENTIUM	1280MB	10 GB	WINDOWS 2000 SERVER	Impressoras
CPS_NTTPROTHEUS	NET FINITY – IBM	PENTIUM III – 633	256 MB	08 GB 17 GB	WINDOWS 2000 SERVER	DHCP
FNCORPNT	NET FINITY – IBM	PENTIUM III – 633	512 MB	08 GB 08 GB 08 GB	WINDOWS 2000 SERVER	Cooperativo de Aplicações
NTTERMINAL	INTEL	PENTIUM III - 500	230 MB	03 GB	WINDOWS 2000 SERVER	Terminal Server
FNSERCOMNT	INTEL	PENTIUM III - 500	256 MB	08 GB 08 GB 04 GB	WINDOWS 2000 SERVER	MetaFrame
ORACLE	RS/6000 F50 – IBM		1 GB	08 GB 17 GB 17 GB 17 GB 17 GB 17 GB	AIX 4.3	Banco de Dados Oracle
FNSIGEFERNT	NET FINITY – IBM	PENTIUM III - 500	1 GB	17 GB	WINDOWS 2000 SERVER	Aplicativo SIGEFER
WEBFORMS	INTEL	PENTIUM III - 500	256 MB	19 GB	WINDOWS 2000 SERVER	Aplicativo WEB FORMS
FIREWALL	INTEL	PENTIUM III - 500	64 MB	30 GB	CONECTIVA LINUX 7.0	FIREWALL
FNPROXYLX	INTEL	PENTIUM III - 500	256 MB	30 GB	CONECTIVA LINUX 7.0	Proxy
ZEUS	NETFINITY – IBM	PENTIUM III	512 MB	08 GB 08 GB	WINDOWS 2000 SERVER	Domínio Brasil Ferrovias
CPSNT001	AS400				WINDOWS 2000 SERVER	Domínio Ferroban
FNWEBSRVNT	INTEL	PENTIUM III - 500	1 GB	30 GB	WINDOWS 2000 SERVER	Intranet e Extranet

O Servidor ORACLE, executando em plataforma IBM RS/6000 F50 e com o sistema Operacional AIX 4.3, contém o Sistema Gerenciador de Banco de Dados 8.1.7, os serviços FTP, TELNET, XWINDOW e SMTP.

O Servidor FNWEBSRVNT, onde estão armazenados a Intranet e a Extranet, utiliza o sistema operacional Windows 2000 Server<sup>4</sup>, o Microsoft Internet Information Services<sup>5</sup> (IIS 5.0), Client Oracle 8.1.7, o Forms & Reports Server, o Oracle WebDB, o Hypertext Processor - PHP e Microsoft Active Server Page

<sup>4</sup> Windows 2000 Server, sistema operacional fabricado Microsoft Corporation no ano de 2000.(MINASI,2000).

<sup>5</sup> O Internet Information Service (IIS) é um servidor web usado para publicação e distribuição de conteúdos WWW em navegadores web padrão (POWELL,1998).

– ASP (MICROSOFT, 2003c). Nesse servidor está instalado o PHP 4.0.6, não sendo possível à instalação de uma versão mais atual, pois existe um aplicativo, chamado Malha<sup>6</sup>, desenvolvido em PHP e utilizado na empresa que não funciona na versão atual (BRASIL FERROVIAS, 2002a).

O Servidor FNPROXYLX, utilizando sistema operacional Conectiva Linux 7.0, faz o cache das páginas Web da Internet para toda a rede corporativa da Empresa. O Servidor está programado para realizar tarefas regulares de geração de relatórios de acesso a Web e Gráficos de evolução dos equipamentos ativos que compõem a infra-estrutura da rede (BRASIL FERROVIAS, 2002b).

O Servidor FNPROXYLX, embora produzido em plataforma Unix, faz integração com o Diretório Ativo do Domínio Windows 2000 através de uma aplicação de consulta de pacotes Server Message Block (SMB). Devido ao fato de estar diretamente conectado com a Internet, o servidor também possui sistema de filtro de pacotes IP e NAT implementado.

Todos servidores estão localizados na sede, no departamento de Gerência de Tecnologia e Sistemas da Informação, e as estações de trabalho mais recentes são de tecnologia COMPAQ, ITAUTEC e ACER, e as mais antigas HP. Há ainda, porém, um significativo contingente de estações de trabalho montadas, e um conjunto de 20 (vinte) NoteBooks.

A Brasil Ferrovias S.A. possui uma grande quantidade de estações de trabalho instaladas tanto na matriz quanto nas filiais. A Tabela 1.3 relata uma amostra dessas estações de trabalho.

---

<sup>6</sup> Malha: Sistema desenvolvido pela SYFER, responsável pelo mapeamento dos trechos da Ferrovia, calculando distância entre as estações, velocidade apropriada para o trecho, entre outros.

**TABELA 1.3 - EXEMPLOS DE ESTAÇÕES DE TRABALHO DA BRASIL FERROVIAS S.A.**

PLATAFORMA	PROCESSADOR	RAM	DISCO RIGIDO	DVD/ CD-ROM	SISTEMA OPERACIONAL
HP VECTRA	486	16 MB	400 MB	NENHUM	WINDOWS 95
COMPAC	PENTIUM III 1Ghz	128 MB	20 GB	CD-ROM	WINDOWS 2000 PROFESSIONAL
ITAUTEC	CELEROM	128 MB	10 GB	CD-ROM	WINDOWS 2000 PROFESSIONAL
ACER	PENTIUM III 866 Mhz	128 MB	20 GB	CD-ROM	WINDOWS 2000
MONTADA	PENTIUM III 550 Mhz	128 MB	10 GB	CD-ROM	WINDOWS 98
NOTEBOOK – TOSHIBA	PENTIUM IV	256 MB	20 GB	DVD	WINDOWS XP
NOTEBOOK – COMPAC	AMD K6	256 MB	20 GB	CD-ROM	WINDOWS 98

Coloca-se em destaque que a empresa possui equipamentos HP Vectra plataforma 486. Estes equipamentos são utilizados na matriz e nas filiais da ferrovia para as consultas, impressão de faturas e despachos através do sistema SIGEFER.

## 1.2. SOFTWARE

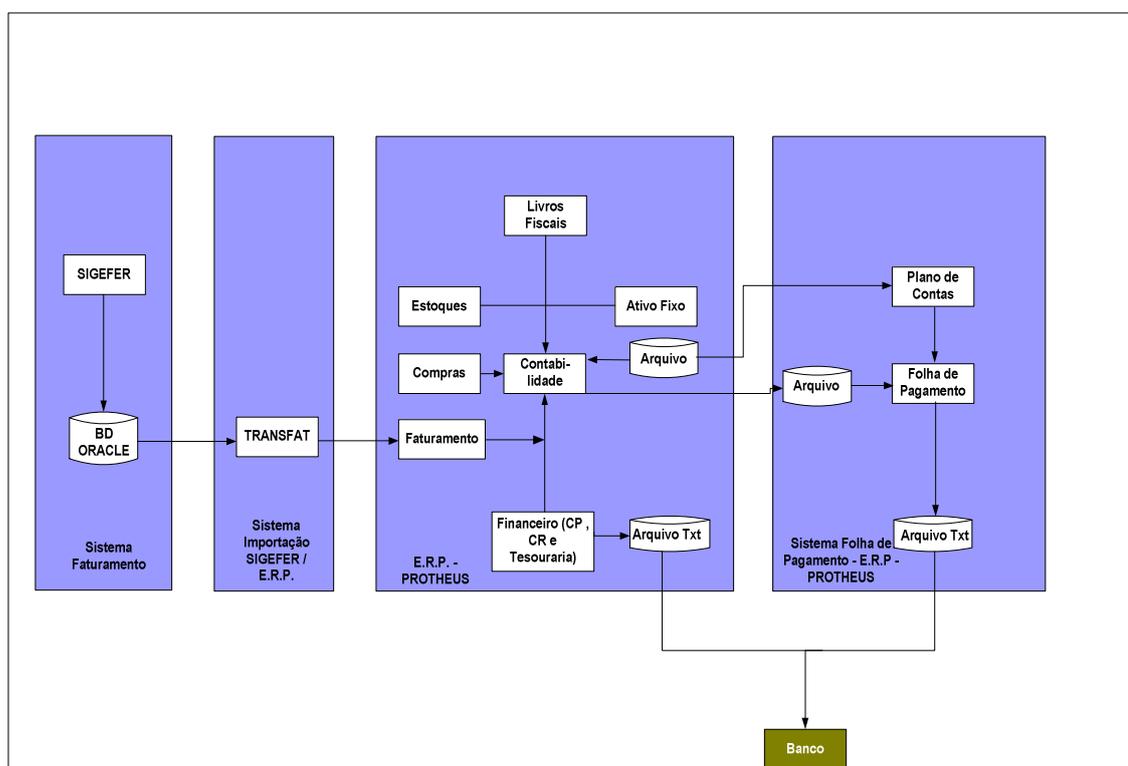
O Software instalado abrange subsistemas para controle de Contas a Pagar, Contas a Receber, Tesouraria, Faturamento, Estoque, Compras, Contabilidade, Livros Fiscais, Folha de Pagamento, subsistemas estes incorporados pelo *Enterprise Resource Planning* (ERP) – PROTHEUS, Automação de Circulação de Trens - ACT, Sistema de Gerenciamento de Equipagem - SGE, Controle de Fretes – SIGEFER e Orçamentos – SISAO (FERROBAN, 2002a).

A Figura 1.2 ilustra a integração do SIGEFER com o sistema *Enterprise Resource Planning* (ERP) – PROTHEUS<sup>7</sup>.

Sistemas ERP são programas de computador dedicados à gestão empresarial e controlam todos os processos e atividades da empresa, desde a emissão de um pedido de compra até o controle da folha de pagamento dos funcionários, ou seja, abrangendo setores como Produção, Venda, Manufatura, entre outros.

<sup>7</sup> Sistema ERP da Microsiga Software S.A.

Isto permite a divisão lógica da empresa em partes que serão controladas por módulos do ERP.



**FIGURA 1.2 - RELACIONAMENTO SIGEFER x E.R.P. – PROTHEUS NA BRASIL FERROVIAS S.A.**

O sistema de Automação de Circulação de Trens – ACT - é responsável pelo licenciamento de trens com utilização de painel sinóptico em monitores de vídeo com interface de despacho e gráfico automático de trens, permitindo a visualização do esquema unifilar de vias e pátios, das composições ao longo da malha e o status de licenciamento de cada trem (ENGESYS, 2002).

O Sistema de Gerenciamento de Equipagem – SGE é responsável por controlar e programar as escalas dos maquinistas, ajudantes e aprendiz de ajudantes (KS SISTEMAS, 2001).

O SISAO – sistema de orçamento - é responsável pelo controle orçamentário da empresa, fazendo o acompanhamento do orçado e do realizado.

O SIGEFER controla as informações de fretes efetuados pela empresa. Este sistema foi desenvolvido na plataforma Oracle<sup>8</sup> usando “Forms” do Oracle (BRASIL FERROVIAS, 2002c)

As tabelas necessárias e utilizadas na geração de consultas diárias sobre o transporte da empresa pertencem ao sistema SIGEFER.

Para importar os dados do SIGEFER para o PROTHEUS, sistema *Enterprise Resource Planning* (ERP) utilizado pela empresa, foi desenvolvido um aplicativo em Delphi chamado TRANSFAT, que faz a integração entre Oracle e arquivos DBF com índices CDX<sup>9</sup>, gerenciado pelo *Advantage DataBase Server*<sup>10</sup> (ADS). ADS é um gerenciador de banco de dados cliente/servidor escalonável de alto desempenho, para redes, mono-usuário (*standalone*), Internet e aplicações móveis.

### 1.3. ORGANIZAÇÃO DO TRABALHO

Este trabalho encontra-se organizado da maneira descrita a seguir. No capítulo 2 são apresentados os fundamentos teóricos necessários para a compreensão do modelo a ser apresentado. No capítulo 3 é apresentado o desenvolvimento do protótipo, que trata da implementação do modelo proposto de Visualização de Informações. No capítulo 4, as considerações finais são apresentadas. O Anexo A mostra um programa em VRML, onde é descrito um exemplo de script gerado para criação de gráficos. O Anexo B contempla o Modelo Atual dos Dados. O Anexo C ilustra um exemplo de programa na Linguagem Java interagindo com VRML para criação de gráficos. O Anexo D traz as Stored Procedures desenvolvidas para o sistema. O Anexo E descreve os programas em ASP criando um arquivo VRML.

---

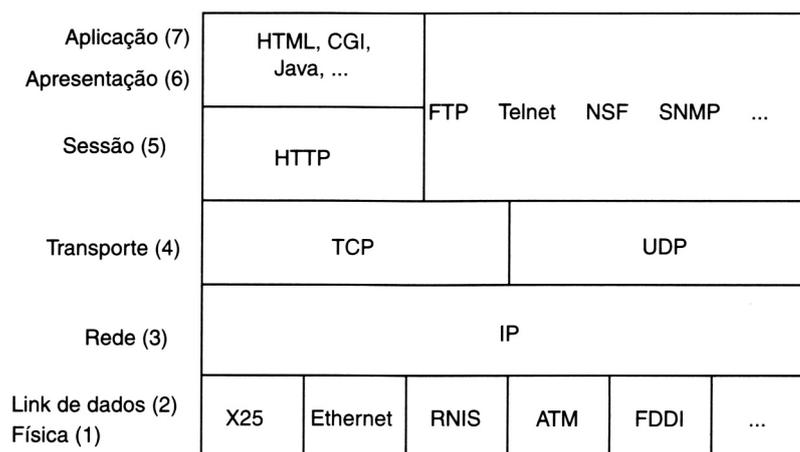
<sup>8</sup> Oracle: Marca registrada da empresa ORACLE Corporation.

<sup>9</sup> CDX: Funcionam como um unico arquivo de indice de arquivos DBF, contendo vários indices dentro, bastando apenas escolher a ordem do indice.

<sup>10</sup> ADS marca registrada da Extend Systems

## 2. BASES TEÓRICAS: ACESSO E VISUALIZAÇÃO DE INFORMAÇÕES REMOTAS

A necessidade de compartilhamento de recursos computacionais alavancou o desenvolvimento das redes de computadores nos anos 60. Neste contexto foi criada a ARPANET (rede militar norte americana) durante o período histórico denominado “guerra fria”, objetivando a comunicação abrangente e com garantia da sua funcionalidade. Criou-se, então, o conceito de “sistemas abertos” (TANEMBAUM, 1994), ou seja, redes de computadores interligadas por algum dispositivo de comunicação física que permite a troca de dados. Por se tratar de sistemas abertos, houve a necessidade da definição de padrões de interconexão em níveis hierárquicos de abstração, que tornam compatíveis os subsistemas de comunicação nos respectivos níveis dessa hierarquia. Essa hierarquia dos subsistemas de comunicação é denominada “pilha de protocolos”. A Figura 2.1 ilustra um exemplo de pilha de protocolos (ÖZSU, 2001)



**FIGURA 2.1 - EXEMPLO DE PILHA DE PROTOCOLOS TCP/IP (ÖZSU, 2001; P.76)**

A Internet é uma rede aberta cuja pilha de protocolos é conhecida por TCP/IP. Esta denominação associa o *Internet Protocol* (IP) ao *Transfer*

*Control Protocol (TCP)*, que determinam a identificação de máquinas e o como se dá à transferência de dados na Internet, respectivamente. Essa denominação genérica embute o protocolo *User Datagram Protocol (UDP)*, também utilizado para a comunicação na Internet. Analogamente ao TCP, o *User Datagram Protocol (UDP)* opera sobre o IP para oferecer um serviço mais rápido para a transferência de dados, porém não confiável (ARNETT, 1997).

A Internet, por ser uma rede aberta e com associação de protocolos possui uma variedade de serviços. Os principais serviços oferecidos na Internet são World Wide Web (ou WWW), correio eletrônico (KLENSIN, 2001), transferência de arquivos (POSTEL, 1985), acesso remoto (MURPHY JR., 2000) e comércio eletrônico. Os serviços WWW, correio eletrônico, transferência de arquivos e acesso remoto, caracterizam uma categoria de aplicações implementadas usando a arquitetura cliente-servidor. Este tipo de aplicação exige a implementação de um programa (denominado “servidor”) que executa numa dada máquina com finalidade única de atender requisições de outros programas (denominados “clientes”).

O comércio eletrônico é uma das aplicações que mais tem tido destaque na Internet, pois trata de quaisquer transações financeiras e é categorizado noutras modalidades de programas. Inicialmente, podem ser consideradas as aplicações que interligam consumidores do negócio diretamente ou aquelas que interligam apenas negócios (ALBERTIN, 2001).

Aplicações que interligam consumidores são denominadas C2C (*consumer-to-consumer*), como por exemplo, leilões convencionais e anúncios de classificados. Aplicações que interligam empresas são denominadas B2B (*business-to-business*), onde o comércio eletrônico dinamiza as transações comerciais entre empresas, proporcionando maior eficácia na distribuição, no acompanhamento do pedido e agilizando a produtividade, como por exemplo, no gerenciamento de fornecedores e pagamentos. Aplicações que interligam empresas e consumidores são denominadas B2C (*Business-to-Consumer*). O comércio eletrônico dinamiza as transações entre empresas e seus respectivos consumidores, e fornece ferramentas de relacionamento interativo mais

eficientes que garantem um “*feedback*” preciso nas ações de comunicação, tais como: propaganda; marketing; serviços *on-line*, como por exemplo, gerenciamento de finanças e informações e compra de produtos.

A implementação de software de comércio eletrônico para a Internet pressupõe o domínio de um conjunto de teorias, métodos, técnicas e ferramentas que tornam essa atividade complexa. O negócio determina o nível de confiabilidade, segurança, agilidade e flexibilidade que é desejado do software e do hardware a ele associados. Apesar do gerenciamento da complexidade de um sistema automatizado de auxílio ao negócio ser particionado, há elementos que não devem ser tratados de maneira disjunta, ou seja, há de ser considerada a plataforma já instalada, com suas compatibilidades e/ou incompatibilidades, e as possíveis migrações entre plataformas, devido à expansão ou retração de mercado, definindo a escalabilidade do sistema computacional.

Na empresa em questão devem ser consideradas a plataforma instalada, apresentada no capítulo 1, a infra-estrutura da rede e as novas necessidades da empresa para melhoria da qualidade do serviço, categorizado como B2C para esta proposta. No que tange à infra-estrutura de rede, utilizar-se-á A seção 2.1 para apresentar a arquitetura, mostrando o ambiente, com a finalidade de tornar claro que esta é uma condição invariante da proposta. Neste recorte, coloca-se em evidência a necessidade do aprofundamento nas questões teóricas que compreendem a produção de interface tridimensional para a visualização com técnicas de realidade virtual. Para isto serão abordadas a questão da produção de aplicações B2C e a questão da geração dinâmica de cenas. A seção 2.2, que trata de aplicações B2C, considera o projeto e acesso a bancos de dados, enfatizando o parque tecnológico instalado na empresa, ilustrando as possibilidades de uso da tecnologia para solucionar a questão. A seção 2.3, trata de conexão de dados utilizando a interface *Open Database Connectivity* (ODBC) e sua configuração. A Seção 2.4 discute tecnologias de desenvolvimento e acesso banco de dados através de páginas Web. A seção 2.5 discute as necessidades relativas à produção de

interface de realidade virtual, considerando a dinâmica das cenas devido às alterações intrínsecas do negócio e as solicitações dos usuários.

## 2.1. INTERNET NO CONTEXTO DA EMPRESA

No que se refere à arquitetura de rede, enfatiza-se a infra-estrutura instalada que a empresa possui para acesso a Internet, Intranet, Extranet e Correio Eletrônico, e Web sites da empresa.

A Figura 2.2 ilustra uma representação lógica da infra-estrutura de rede instalada na empresa. Nesta, observa-se a existência do servidor “FNPROXYLX”. Este minemônimo usa as duas primeiras letras para identificar a empresa, as duas últimas letras estão associadas ao sistema operacional existente no servidor e entre as duas primeiras e duas últimas está definido o servidor, neste caso servidor “Proxy”.

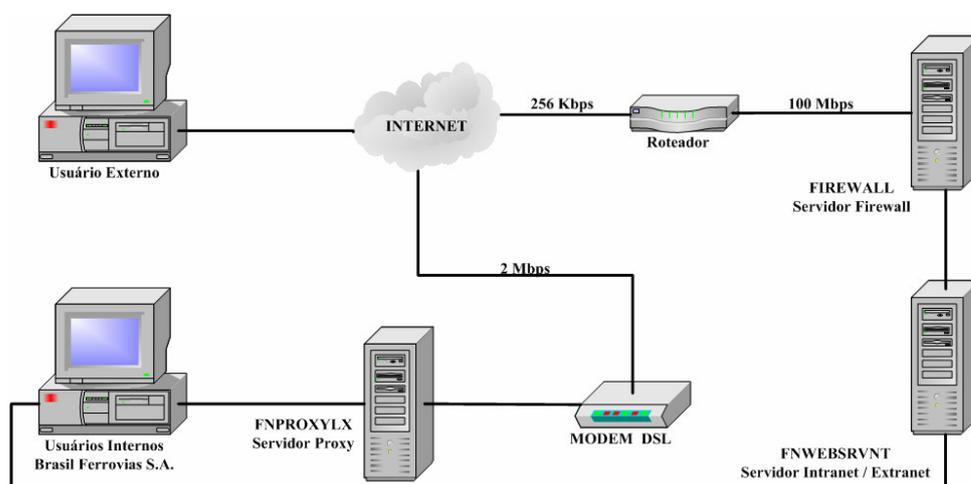
Um servidor “Proxy” atua como intermediário entre a estação de trabalho do usuário e a Internet, permitindo à empresa o controle das atividades administrativas, o aumento da segurança e o serviço de “*caching*”. Serviços obtidos através de conexão via modem DSL, o qual oferece acesso à Internet através de uma linha dedicada de 2 Mbps da telefônica, serviço este denominado Speedy Business.

Há duas maneiras distintas para o acesso aos dados da empresa. Em uma das maneiras, os dados são públicos e estão armazenados no servidor “<http://www.bestway.com.br>”, que é um servidor terceirizado. Outro modo permite o acesso ao subsistema interno à empresa, caracterizando uma intranet e uma extranet.

Caracteriza-se como utilização de serviços de extranet, os acessos que funcionários/clientes efetuam aos provedores de serviços www da empresa para consultas que são extraídas do sistema de informação, os quais são feitos através de dispositivos de roteamento e de segurança (“Firewall”). O

dispositivo de roteamento leva informações através de uma rede desde a origem até o destino, através de pelo menos 1 (um) nodo intermediário, fazendo o elo do usuário externo com o “Firewall”.

A Intranet é caracterizada por oferecer acesso restrito aos dados. Isto significa que apenas funcionários da empresa têm acesso a esses dados. A configuração deste subsistema de rede é dada através da configuração do software de browser nas estações de trabalho, configurando endereço do Proxy com endereço IP 10.1.1.50 e porta 2560 (por medida de segurança estão sendo usados endereçamentos fictícios), obrigando o funcionário a se identificar através de seu usuário e senha da rede quando iniciar o software do Browser.



**FIGURA 2.2 - ESTRUTURA ACESSO INTERNET, INTRANET E EXTRANET**

Em 2002, existiam apenas o web site de algumas das empresas que compõem a *holding*. São elas Ferronorte S.A. – Ferrovias Norte Brasil, Ferroban – Ferrovias Banderiantes S.A. e Portofer, cujos web site estavam hospedados num provedor fora da empresa.

O web site da Ferroban – Ferrovias Bandeirantes S.A., continha informações sobre a empresa, histórico de sua concessão, seus principais acionistas, seu organograma, mapa de sua malha, seus clientes, descrição de sua frota, detalhando tipos de vagões e locomotivas, seus serviços, tipo de produtos transportados, previsão de investimentos na malha e na frota e um formulário

onde se podiam pedir mais informações ou enviar novas sugestões (FERROBAN, 2002b).

O web site da Ferronorte S.A. – Ferrovias Norte Brasil, trazia informações sobre a empresa, histórico de sua formação e concessão, seu quadro acionário, suas sinergias envolvendo a Tenorte e a Portofer, sua frota, detalhando suas locomotivas e tipos de vagões, sua malha, e uma consulta sobre as composições dos trens em circulação, porém esta consulta está desabilitada (FERRONORTE, 2002).

O Web site da Portofer contém informações sobre a empresa, seu histórico de concessão, sua missão, seus objetivos, suas instalações e equipamentos, sua malha, o roteiro operado pela empresa, uma consulta de vagões, seus investimentos, gráficos operacionais, tipos de vagões, terminais no porto de Santos, pesagens de vagões, ferrovias que utilizam os serviços da empresa e notícias da empresa circulada na imprensa (PORTOFER,2002).

No segundo semestre de 2003 os Web sites da Ferrobán e Ferronorte foram retirados do ar e colocado o Web site da Brasil Ferrovias, que contém informações sobre a malha, a frota e o histórico das empresas pertencentes a *holding*, informações sobre a composição acionária, serviços, mapa do site, contatos, uma extranet com consulta de “onde está a sua carga”, um area restrita com acesso permitido apenas para funcionários da empresa, contendo o programa de WEBMAIL para acesso a e-mail, *link* paras as normas e fomulários padrões utilizasos na empresa, entre outros.

Como descrito anteriormente, a Figura 2.2 ilustra a infra-estrutura lógica da empresa, onde nota-se a existência de um servidor web que contempla os web sites, a Intranet e a Extranet da empresa. Utilizaremos o item 2.1.1 para descrever sobre a arquitetura da Intranet e o item 2.1.2 para a arquitetura da Extranet .

### 2.1.1. INTRANET

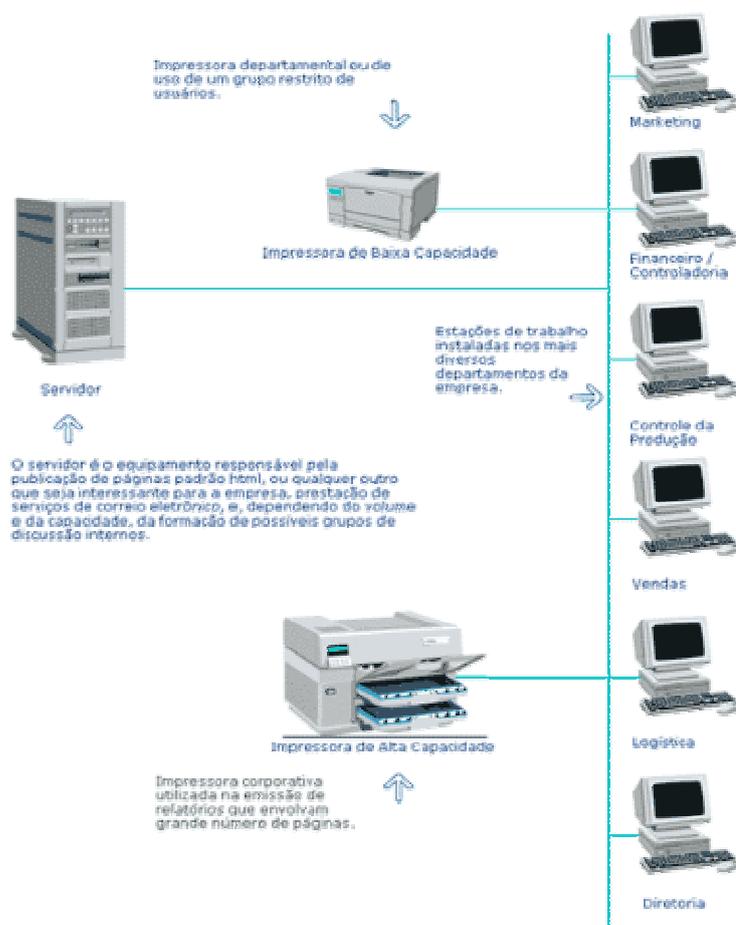
Intranet é um meio de comunicação, distribuição e compartilhamento de informações que utilizam padrões abertos de arquitetura Internet, infraestrutura de redes locais ( *Local Server NetWork* – LAN) e remotas (*Wide Server Net Worker* - WAN) e independência das plataformas de hardware e de software.

A Intranet é uma rede interna baseada no protocolo IP que se caracteriza pelo uso de todas as facilidades existentes na World Wide Web, como correio eletrônico, *File Transfer Protocol* - FTP, Telnet, *Network News* - NNTP, HTTP e interface comum dos browsers. Ela é composta por um servidor Web corporativo, tornando-se disponíveis para os usuários através de uma rede interna ou acesso discado privativo, fornecendo assim uma variedade de informações por meio de um único *front-end*, o navegador (browser) Web.

Além de incorporar toda a tecnologia Internet, as Intranets podem utilizar a estrutura de comunicação de dados da própria rede pública para se comunicar com filiais ou com qualquer empresa conectada à grande rede.

Normalmente elas são configuradas pelas empresas, num segmento isolado nos servidores que são usados para se ter acesso à Internet, para ligar seu sistema de informação internamente, dando acesso aos seus colaboradores (funcionários), sócios e quem quer que esteja autorizado a ter acesso ao conteúdo de suas páginas e atingem seu maior potencial quando são capazes de se conectar com os bancos de dados da empresa.

Normalmente, são requeridas uma senha de acesso e a identificação do usuário. Com isso as empresas não têm mais limitações geográficas de seus escritórios. A Figura 2.3 ilustra uma empresa na qual as estações de cada departamento estão conectadas a um servidor Intranet, onde estão publicadas as páginas HTML e/ou sistemas da empresa.



**FIGURA 2.3 - ESTRUTURA DE UMA INTRANET**

Dentro dos limites da empresa, tudo o que circula em forma de material impresso em papel pode ser colocado na Intranet de forma simples e objetiva: desde manuais e políticas de procedimento até informações de marketing, catálogos de venda de produtos, recursos humanos e catálogos telefônicos.

Intranets são usadas nas empresas para aproximação de departamentos e filiais, auxiliando a integração, troca de informações corporativas em tempo real. Com esta forma de infra-estrutura de distribuição de informações, os gerentes são capazes de definir suas estratégias computacionais e ter um melhor controle organizacional e uma melhor competitividade no ambiente de negócio (LAI, 2001).

O uso de Intranet permite a integração de dados corporativos para um crescimento da eficácia da organização com o mínimo custo, mas com bom resultado comparado ao uso de sistemas legado e aplicações de rede.

O aproveitamento do uso da Intranet pelos departamentos de uma empresa é diferenciado. Por exemplo, o Recursos Humanos é um departamento que se beneficia com o uso de Intranet, onde ele pode armazenar, distribuir, disponibilizar, manter e imprimir documentos corporativos eletronicamente.

A Intranet ajuda na quebra de barreiras existentes na comunicação e no gerenciamento, permitindo assim que os funcionários possam distribuir e comunicar as suas idéias mais rapidamente, tornado assim mais eficaz sua participação em tomadas decisões.

Os maiores benefícios com o uso de Intranet estão ligados a redução de custos e a o acesso a dados corporativos da empresa.

A intranet ajuda na quebra de barreiras existentes na comunicação e no gerenciamento, permitindo que os funcionários da empresa possam contribuir com idéias de maneira eficaz e mais rápida.

A Intranet da Brasil Ferrovias, está hospedada num servidor próprio da empresa e pode ser acessada tanto internamente como externamente pelos funcionários da empresa.

A Intranet da Brasil Ferrovias contém opções de HelpDesk, onde os funcionários podem efetuar chamados a área de Gerência de Tecnologia e Sistema de Informações, uma opção onde se encontra a Área de Transferência de Arquivos, utilizado para troca de arquivos, uma área onde se armazena os padrões de documentos, que contém as normas e padrões de documentos a serem utilizados na empresa, uma opção de WebMail utilizada para acessar as mensagens de e-mail, uma opção de acesso ao Sistema de Faturamento de Fretes – SIGEFER, uma opção de acesso ao Sistema de Orçamento da Empresa – SISAO, uma opção de acesso à planilha com o Boletim Diário de Transportes, opção de acesso a indicadores de transporte, opção de acesso aos projetos da engenharia (esta com acesso restrito ao departamento de engenharia), e uma opção de consulta de ramais e e-mail dos funcionários.

Concluindo, o conceito Intranet é enfatizado nesta proposta pelo fato de que a gerencia comercial terá acesso as consultas a serem geradas através da Internet.

### **2.1.2. EXTRANET**

Extranet é uma rede de negócios que permite a união de empresas parceiras através de suas Intranets. Também utiliza o protocolo de HTTP, tendo como consequência a facilidade de troca de conteúdos que a World Wide Web e uma Intranet possuem (GREER,1998).

Extranet possui um sistema de telecomunicação público para compartilhar parte de uma informação de negócio ou operações com provedores, vendedores, sócios, clientes, ou outros negócios.

Empresas podem utilizar extranet para troca de grandes volumes de dados usando *Electronic Data Interchange* (EDI), compartilhar catálogo de Produtos exclusivamente com seus atacadistas, colaborar com outras companhias em esforços de desenvolvimento em comum, desenvolvimento junto e uso de programas de treinamento com outras empresas, prover ou obter acesso a serviços oferecidos por uma empresa a um grupo de outras companhias. Por exemplo, uma aplicação bancária on-line administrada por uma empresa em nome de bancos associados, compartilhar notícias de interesse comum exclusivamente com sócios das empresas, entre outros serviços. A Extranet pode ser vista como parte da empresa que é estendida aos clientes para compartilhar informações de interesse comum e exclusivo. Para acessar a extranet as empresas não precisam ter o mesmo tipo de computador (hardware), sistema operacional, gerenciadores de banco de dados (software) ou browser para navegação. Uma extranet requer segurança e privacidade. Estas requerem gerenciamento de servidor Firewall, uso de certificados digitais ou meios semelhantes de autenticação de usuário e criptografia de mensagens.

A Brasil Ferrovias em 2002 estava iniciando o uso de Extranet. Possuía uma consulta que mostra a composição e localizações de trens em circulação pela

malha da empresa, consulta esta acessada através do web site da Ferronorte. Possuía uma consulta de posicionamento de vagões das ferrovias que utilizam os serviços da empresa, consulta acessada através do web site da Portofer. Para ter acesso a essas consultas necessita-se somente que a empresa cliente tenha um usuário e uma senha de identificação. Caso não possua basta solicitar via e-mail. As páginas de Extranet estão hospedadas num servidor da empresa, como apresentado anteriormente na Figura 2.2.

O conceito Extranet é enfatizado nesta proposta pelo fato de que os clientes acessarão as consultas a serem geradas através da Extranet.

Concluindo, neste item foram descritos os conceitos de Internet, Extranet e Intranet, suas arquiteturas e forma de acesso na Brasil Ferrovias. Como a proposta da pesquisa são consultas na Intranet e Extranet, no item 2.2 enfatizar-se-á a estrutura do banco de dados utilizado na empresa.

## **2.2. ELEMENTOS DA APLICAÇÃO**

No contexto da empresa descrito anteriormente, o desenvolvimento de um sistema B2C (*Business-to-Consumer*) usando interfaces de realidade virtual se fazem necessários os conhecimentos de Sistemas de Banco de Dados, Acesso a Base de dados via *Open Database Connectivity* (ODBC), Tecnologia de Desenvolvimento e Acesso a Banco de Dados via Web.

### **2.2.1. O BANCO DE DADOS**

Sistema de Banco de Dados (SBD) da empresa é o sistema responsável pela manutenção de registros no computador, como ilustra a Figura 2.4, e é composto por dados armazenados em banco de dados (BD), por software de gerenciamento de banco de dados (SGDB) e por programas de aplicação.

O sistema de banco de dados pode ser categorizado como relacional, entidade-relacionamento, rede, hierárquico ou orientado a objeto (DATE, 1991).

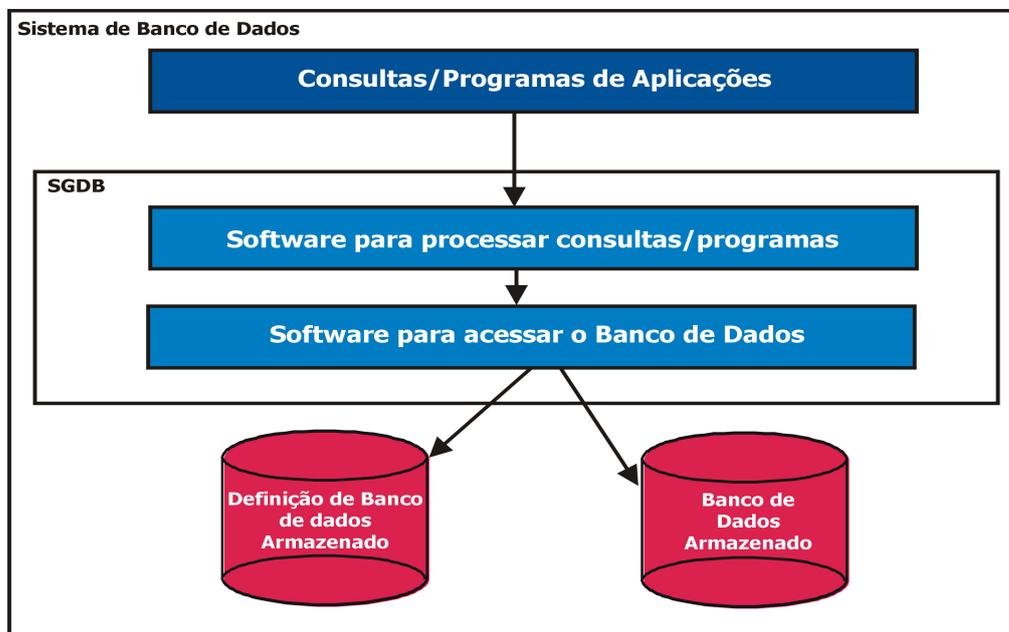


FIGURA 2.4 - ESTRUTURA SIMPLES DE SISTEMA DE BANCO DE DADOS (ELMASRI,2000;p.3)

Uma definição mais precisa de Banco de Dados não existe formalmente, contudo pode-se dizer que:

Banco de Dados representam aspectos restritos de uma parte restrita do mundo real, denominado de mini-mundo. Quaisquer alterações que ocorra no mini-mundo são refletidas no banco de dados.

Banco de Dados é uma coleção logicamente coerente de dados com um determinado significado próprio. Isto significa que um conjunto aleatório de dados não pode ser considerado um Banco de Dados.

Banco de Dados é projetado, construído e composto por um conjunto de dados para um propósito específico. Existe um grupo de usuário ou algumas aplicações pré-concebidas onde estes dados serão utilizados. (ELMASRI,1994, p.2).

Sistema Gerenciador de Banco de Dados (SGDB) é uma interface de software entre o banco de dados e o usuário. Um sistema de gerenciamento de banco de dados trata de solicitações do usuário para ações de bancos de dados e permite o controle centralizado da segurança e da integridade dos dados, tendo como suas principais funções: controle de redundância, compartilhamento de

dados, controle de acesso, controle de transações, possibilidade de múltiplas interfaces, representação de relacionamento complexo entre dados, reforçar restrições de integridade, providenciar “Backup” e restauração de dados e independência de dados. Existem vários SGDB no mercado, como por exemplo, o Oracle, o SyBase, o Microsoft SQL Server, o Informix e o DB2.

A Brasil Ferrovias, como já mencionado no Capítulo 1, utiliza o Sistema Gerenciador de Banco de Dados Oracle Server, instalado no Servidor ORACLE, onde estão armazenadas as tabelas do sistema SIGEFER.

No banco de dados Oracle, a cada inicialização, uma Área Global do Sistema (SGA) é alocada e processos de segundo plano (Background Process) são inicializados, visando controlar o banco (RAMALHO,1999).

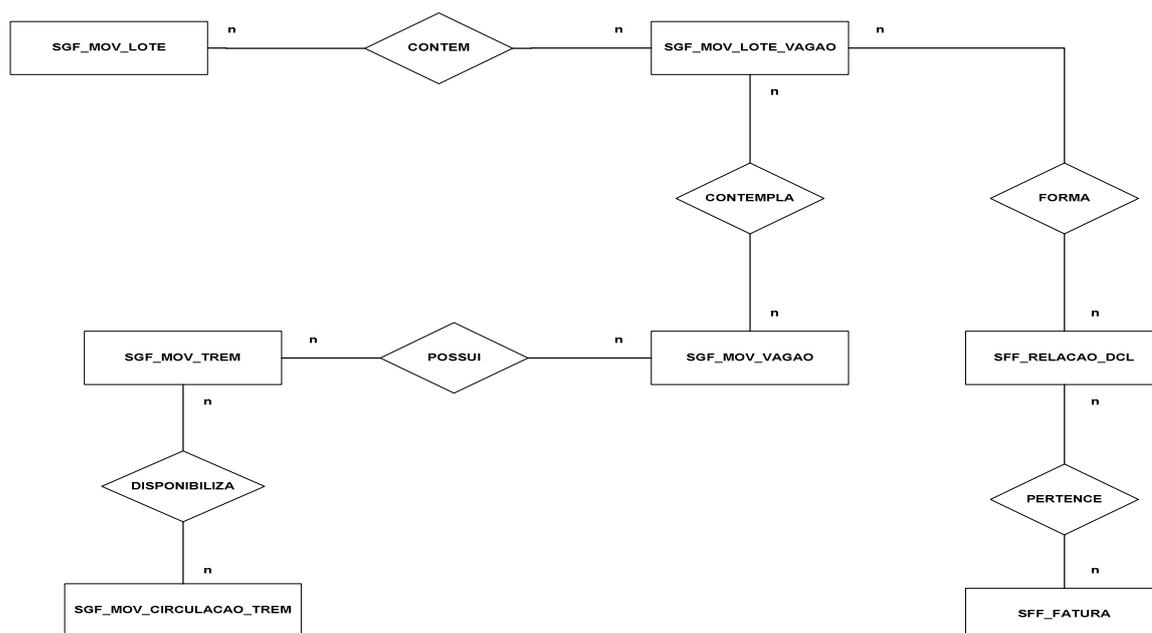
O conjunto da SGA (buffers da memória RAM) mais processos de segundo plano (Background), que juntos oferecem aos vários usuários o acesso ao banco de dados são denominados instância de Banco de Dados.

A empresa trabalha com um banco de dados de produção e outro de banco de dados para o desenvolvimento. Pode-se criar em um mesmo equipamento mais de uma instância. Para cada instância existe um banco de dados.

No Banco de Dados da empresa foram criadas várias instâncias. As instâncias existentes são: FNDBP, FNDBD e BFDBG, sendo FNDBP e FNDBD instâncias das bases de produção, e BFDBG, instância da base de desenvolvimento.

Quando falamos em metodologia para modelagem de banco de dados relacionais, uma das mais utilizadas é o Modelo de Entidade e Relacionamento (M.E.R.), que faz a modelagem do banco de dados, utilizando a representação de entidades externas, relacionamentos e atributos.

O Diagrama Entidade-Relacionamento (D.E.R.), ilustrado na Figura 2.5, modela parte das tabelas do sistema SIGEFER.



**FIGURA 2.5 - EXEMPLO DO D.E.R DO BANCO DE DADOS**

A Brasil Ferrovias não possui documentação da relação das tabelas do SIGEFER. A normalização dos dados deste banco não atende aos requisitos da Terceira Forma Normal (DATE, 1991), causando a replicação de dados e implicando em possíveis problemas de inconsistências.

O início da pesquisa exigiu a elaboração de uma engenharia reversa do banco com a finalidade de permitir a criação de uma visualização diferenciada dos dados. Nesse momento foi utilizada a ferramenta *AllFusion ErWin Data Modeler*<sup>11</sup> (ERWIN, 2005) para gerar a lista de tabelas e os seus respectivos relacionamentos, facilitando a elaboração do Diagrama Entidade-Relacionamento elaborado na Figura 2.5.

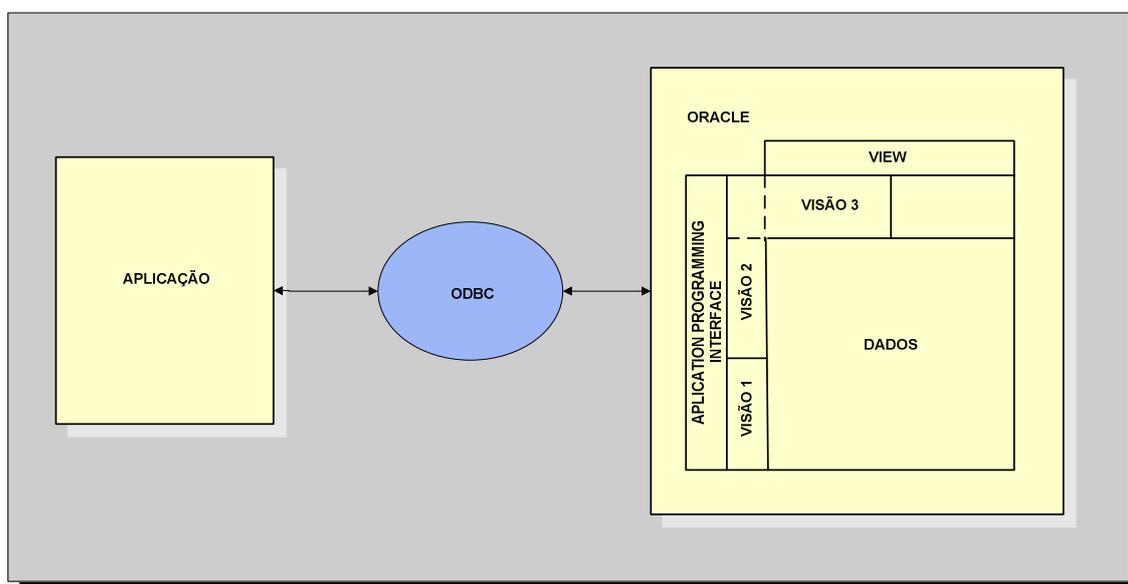
Constatou-se que o banco de dados da Brasil Ferrovias é composto por 400 tabelas, que representa a visão do *DataBase Administrator* (DBA). As tabelas relevantes para a elaboração desta proposta seguem no anexo A, e estão representadas na visão modelada na Figura 2.5.

<sup>11</sup> Marca registrada da empresa COMPUTER ASSOCIATES INTERNATIONAL.

O conceito de visões (views) consiste de um subconjunto de dados do BD, ou pode conter dados “virtuais” que são derivados dos existentes no BD, mas que não estão explicitamente armazenados. As *views* existem para satisfazer as necessidades dos usuários na apresentação apenas das informações importantes e ocultando aquelas desnecessárias (ÖZSU, 2001).

A geração dos relatórios enviados aos clientes e à gerência corresponde a visões diferenciadas de um mesmo banco de dados. Mais especificamente, no contexto atual, há uma visão referente aos transportes de produtos dos clientes e outra da movimentação diária de produto por cliente.

A Oracle oferece um módulo denominado “view” que auxilia a geração de relatórios baseados em diferentes visões. O uso desse aplicativo produz uma nova tabela, que pode ser manipulada por outras aplicações que utilizam o suporte de conexão ODBC<sup>12</sup>, que será detalhado no item 2.2.4, conforme ilustra a Figura 2.6.



**FIGURA 2.6 - ACESSO BANCO DE DADOS**

<sup>12</sup> ODBC : Open Data Base Connectivity

A aplicação ilustrada na Figura 2.6 pode fazer acesso ao banco de dados enviando uma cadeia de caracteres que encerra uma operação no banco de dados escrita na linguagem SQL<sup>13</sup> (URMAN, 2002). Este tipo de acesso permite quaisquer consultas, comprometendo a segurança do Banco de Dados. Além disso, a cadeia de caracteres tem de ser analisada do ponto de vista léxico, sintático e semântico, objetivando a sua compilação para o respectivo processamento no SGDB. Isto faz com que o acesso possa comprometer a performance do banco de dados (JERKE, 2001). Uma solução para esta questão é o uso de stored procedures, descritos nos itens 2.2.1.1 e 2.2.1.2 respectivamente.

#### **2.2.1.1. STORED PROCEDURES**

Um Stored Procedure (Procedimento Armazenado) é um programa SQL compilado e armazenado no banco de dados que pode ser executado na linha de comando da interface do Oracle. Cada consulta é um conjunto de comandos SQL que implementam um subprograma de acesso ao banco de dados para realizar uma determinada operação. É, geralmente, tratado como um registro do banco de dados. Isto faz com que o acesso usando banco a Stored Procedure seja controlado por mecanismos de segurança do servidor.

Uma Stored procedure aceita parâmetros de entrada, de forma que um único procedimento pode ser utilizado na rede por múltiplos clientes usando diferentes dados de entrada.

A Figura 2.7 ilustra usuários fazendo acesso ao banco de dados e uma mesma stored procedure chamada GETCLIENTE, resultando, assim, uma visão diferente para cada usuário.

---

<sup>13</sup> SQL: Structured Query Language - Linguagem de Consulta Estruturada.

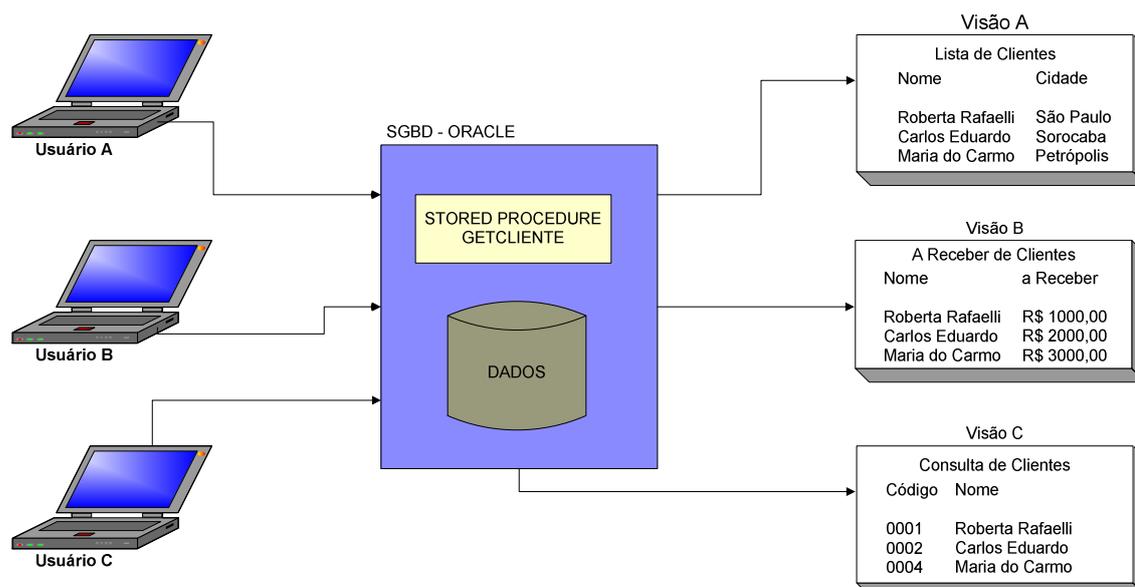


FIGURA 2.7 - EXEMPLO DE ACESSO A STORED PROCEDURE POR VÁRIOS USUÁRIOS

A Tabela 2.1 mostra a estrutura sintática de um Stored Procedure a ser criada para um banco de dados Oracle. Cabe salientar que os procedimentos armazenados podem ter sintaxe diferente e definidos pelo fabricante do SGBD.

TABELA 2.1 – ESTRUTURA SINTÁTICA DE UM STORED PROCEDURE NO ORACLE

```

CREATE [OR REPLACE] PROCEDURE <nome da procedure>
[parâmetro [{ IN | OUT | IN OUT }] tipo,

.....

[parâmetro [{ IN | OUT | IN OUT }] tipo, { IS/AS}
BEGIN
/* sessão executável */
EXCEPTION
/* sessão de exceção */

END <nome da procedure>

```

Stored Procedures residem no servidor de Banco de Dados. Uma vez executada no servidor seu plano de execução fica armazenado na memória, tornando, assim, os próximos acessos mais rápidos.

Uma Stored Procedure pode ser utilizada por sistemas desenvolvidos em várias linguagens, como, por exemplo, Delphi ou ASP. Essas linguagens precisam apenas efetuar a chamada da stored procedure, permitindo então que o código fique mais enxuto e caso haja mudança no banco de dados somente é preciso acertar a stored procedure não precisando manutenção nos sistemas que a utilizam.

#### **2.2.1.2. PACKAGES**

Package (Pacotes) é um objeto do banco de dados capaz de armazenar *procedures* e *functions* integradas, que podem ser executadas separadamente como se fossem parte de uma biblioteca ou à partir de uma execução provocar várias execuções encadeadas.

Package são áreas de armazenamento de sub-programas, constantes, variáveis e cursores em PL/SQL, linguagem SQL do Oracle, podendo compartilhar as informações da PACKAGE com outros aplicativos. Um package tem duas partes separadas: a especificação e o corpo.

Na especificação do Package há informações sobre o seu conteúdo, determinando os procedimentos, funções, tipos, cursor e constantes incluídas nele. Ela também especifica as variáveis de entrada e o tipo de saída de cada subprograma do package.

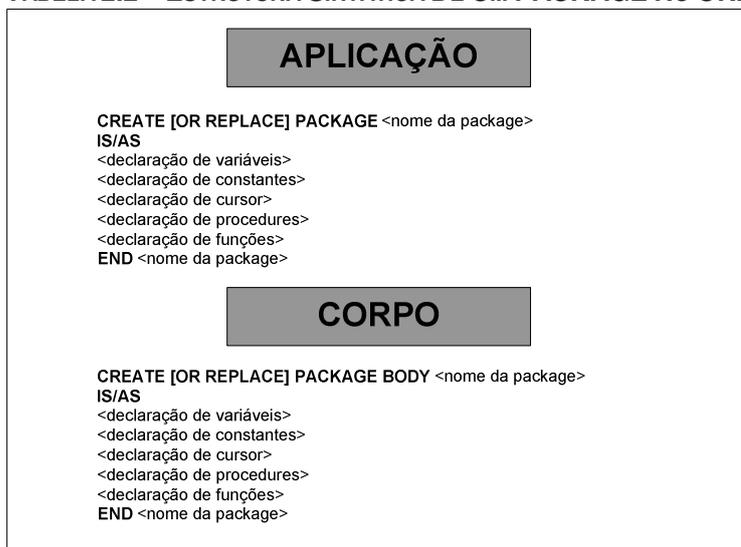
O corpo do package contém o código SQL para cada procedimento e/ou função que foram declarados na especificação do package.

Segundo URMAN (2002), o corpo do pacote é opcional caso não haja nenhuma procedure ou função (apenas declarações de variável, cursores, tipos e outros) na especificação do package.

Tanto a especificação do package como o Corpo precisam ser compilados, porém só conseguimos compilar o corpo se a especificação do pacote tiver sido compilada com sucesso.

A Tabela 2.2 mostra a sintaxe de um package a ser criada para um banco de dados Oracle. Cabe salientar que os packages podem ter sintaxe diferente e definida pelo fabricante do SGBD.

**TABELA 2.2 – ESTRUTURA SINTÁTICA DE UM PACKAGE NO ORACLE**

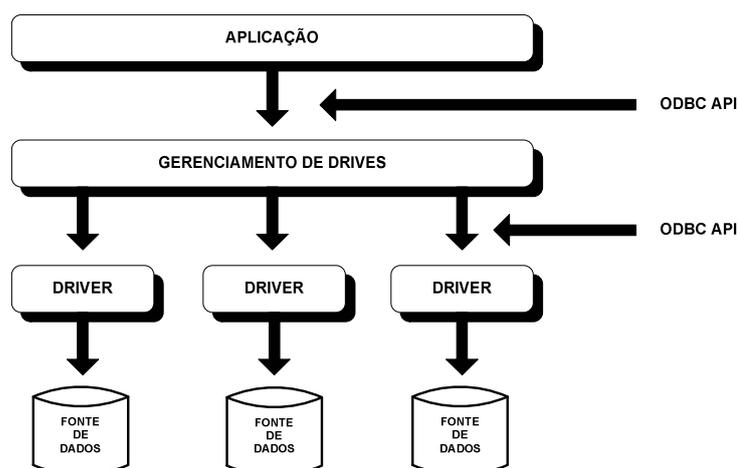


Qualquer declaração feita na especificação do Package deve ser elaborada no corpo da Package. Por exemplo, da mesma forma que uma função for declarada na especificação do package ela deve aparecer no corpo do package.

### 2.2.2. CONEXÃO DE BANCO DE DADOS VIA OPEN DATABASE CONNECTIVITY

Originalmente desenvolvida pela Microsoft, a interface *Open Database Connectivity* (ODBC) é a mais popular para acesso a banco de dados. Esta interface surgiu da necessidade de comunicação entre vários bancos de dados, pois não há uma interface única de programação para acesso a banco de dados. Cada fornecedor oferece uma interface própria e otimizada para interagir com o seu produto.

A interface ODBC faz uso da *Structured Query Language* (SQL) como linguagem de acesso ao banco de dados. As funções desta interface permitem o acesso a banco de dados de diversos fornecedores em diferentes plataformas. Dentre elas, destacam-se Oracle<sup>14</sup>, Sybase<sup>15</sup>, SQL Server<sup>16</sup>, DB2<sup>17</sup>, Interbase<sup>18</sup> e Informix<sup>19</sup>.



**FIGURA 2.8 - ARQUITETURA ODBC API (MICROSOFT, 2003B)**

A arquitetura da interface ODBC é composta de quatro componentes, ilustrados na Figura 2.8, que são:

*Aplicação*: é um programa que processa as requisições dos usuários, submete as instruções SQL através da chamada da interface ODBC e retorna se houver o resultado das instruções.

*Gerenciamento de Drivers*: responsável pela comunicação entre aplicações e drivers, onde são lidos e endereçados os dados em nome da aplicação. Executa as funções ODBC chamadas ou passa para um Driver executar.

<sup>14</sup> Oracle: Marca Registrada da Oracle Corporation.

<sup>15</sup> Sybase: Marca Registrada da Sybase Inc.

<sup>16</sup> SQL Server: Marca Registrada da Microsoft Corporation.

<sup>17</sup> DB2: Marca Registrada da IBM.

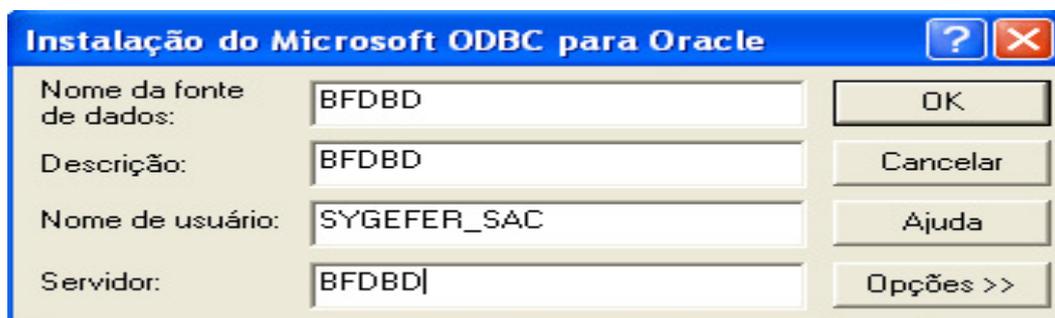
<sup>18</sup> Interbase: Marca Registrada da Borland Software Coporation.

<sup>19</sup> Informix: Marca Registrada da IBM.

*Driver:* são bibliotecas que implementam a interface ODBC, onde são processadas as funções ODBC para executar as instruções SQL no banco de dados e retornar o resultado para a aplicação.

*Fonte de Dados:* agrupa todos as informações técnicas para acessar os dados, como o nome do driver, endereço de rede, o software de rede, ou seja toda plataforma necessária para acessar o sistema de banco de dados.

Para a utilização da Interface ODBC é necessário primeiramente a instalação do gerenciador de drivers da interface ODBC no servidor Web onde serão executados os programas que acessam informações no servidor de banco de dados. Em seguida, é necessário instalar os drivers de sistemas do banco de dados e finalmente o banco de dados têm que ser registrado com uma fonte de dados do Sistema, como é ilustrado na Figura 2.9.



**FIGURA 2.9 - EXEMPLO DE REGISTRO DO B.D. ORACLE COMO FONTE DE DADOS DO SISTEMA**

No exemplo da Figura 2.9 ilustra-se como registrar um driver Microsoft para banco de dados Oracle, em que se utiliza como nome da fonte de dados e descrição BFDBD, informa-se o nome do usuário a acessar o banco como SYGEFER\_SAC e o nome do banco de dados como BFDBD, que se refere à instância a ser utilizada.

O exemplo foi implementado com Microsoft ODBC, para Oracle, porque a empresa possui o Banco de Dados Oracle, fazendo com que a proposta de implementação deste trabalho para a dissertação utilize esse driver para a conexão.

A Interface ODBC é também utilizada em páginas *Web* que fazem conexão a banco de dados. No item 2.2.3 são relatadas as tecnologias para desenvolvimento e acesso a Banco de dados via *Web*.

### **2.2.3. TECNOLOGIA DE DESENVOLVIMENTO E ACESSO A BANCO DE DADOS VIA WEB**

O conteúdo de páginas *Web* estáticas é determinado na criação da página, e toda vez que a página é acessada as informações são as mesmas. Numa página dinâmica, o conteúdo muda de acordo com as informações pedidas pelo usuário, as quais são processadas consultando os dados no banco de dados e o resultado é mostrado no browser da estação de trabalho do usuário.

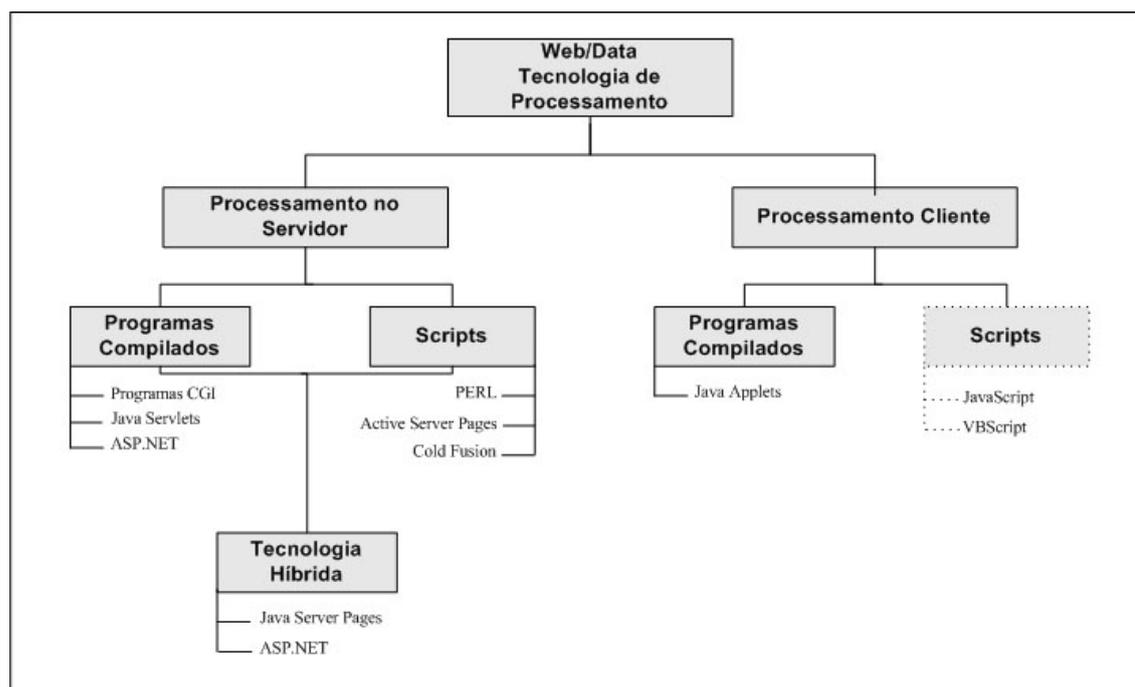
Com o grande crescimento de desenvolvimento de sistemas *Web*, e com o uso de sistemas *web* pelas empresas como ferramenta de negócio, torna-se primordial o bom conhecimento das tecnologias de desenvolvimento de páginas dinâmicas.

O desenvolvimento de sistemas *Web* se distingue do desenvolvimento tradicional de softwares pelo crescimento de requerimentos e pela contínua mudança em seu conteúdo de informação, determinado que o *Web Sites* são desenvolvidos para ser de fácil escalabilidade e manutenção (GINIGE, 2002).

Uma possibilidade de melhorar a manutenção de dados em *Web sites* é o uso de banco de dados para armazenamento desses dados e o uso de ferramentas que permitem alterar o conteúdo das páginas pré-definidas em tempo de acesso.

Existem várias tecnologias para o desenvolvimento de páginas *Web* com acesso a Banco de Dados. As tecnologias são divididas em “Processamento no Lado do Servidor”, “Processamento no Lado do Cliente” ou um conjunto das duas, denominado “Processamento Híbrido”. (MORRISON, 2002)

A Figura 2.10 ilustra as tecnologias de processamento de dados usadas na construção de web sites, no lado do servidor, no lado do cliente ou híbrida. O tracejado destaca tecnologias com processamento híbrido no lado do cliente (MORRISON, 2002).



**FIGURA 2.10 - TECNOLOGIAS DE PROCESSAMENTO USADAS NA CONSTRUÇÃO DE WEB SITE**

A terminologia “Processamento no Servidor Web” (ilustrado na Figura 2.10) significa que o programa que atende às solicitações dos usuários pode fazer uso de outros módulos de software para responder às requisições, além das suas funcionalidades. Esses módulos podem ser programas compilados, scripts ou híbridos.

Ao acessarem a páginas Web, os usuários fazem solicitações através de um formulário HTML, as quais são encaminhadas ao Servidor Web. No Servidor Web é feito o processamento das solicitações, gerando uma página Web dinâmica com o resultado e enviada para o Browser na estação de trabalho do usuário solicitante. Como o processamento das solicitações efetuadas pelo usuário é no servidor Web, o código do programa responsável pela geração da

página Web de resultado fica escondido do usuário, não permitindo cópia do programa, através do browser.

*Programas de Servidor Compilado.* Todas as requisições feitas pelo usuário, através de formulário HTML, são enviadas ao servidor Web, onde será processado por um programa de servidor compilado executado pelo servidor Web. O programa é um executável independente que roda no servidor Web, captura as requisições e processa, gerando uma página Web com saídas formatadas em tags HTML na memória do servidor Web. O Servidor Web envia esta página com o resultado para o browser do usuário.

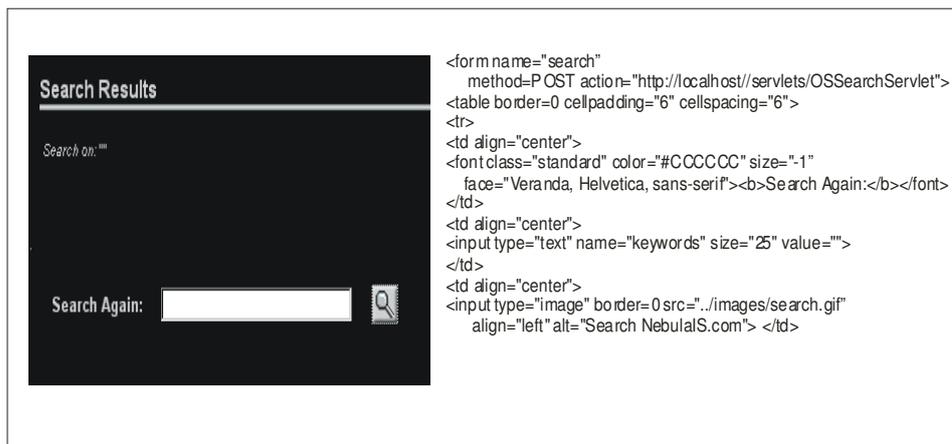
Linguagens mais comuns para a criação de programas de servidor compilados são Java<sup>20</sup>, Visual Basic<sup>21</sup> e C++<sup>22</sup>. No entanto, qualquer linguagem que cria programas executáveis pode ser usada, contanto que ela suporte comandos usados por um dos protocolos que estabelece normas de procedimentos para comunicação entre servidor Web e programas. O primeiro protocolo para uso com HTML foi o Common Gateway Interface (CGI) (CGI,2003). O uso do protocolo CGI possui uma desvantagem, pois para cada formulário enviado para o servidor Web uma cópia do programa de serviço é iniciada no servidor Web. Por exemplo, a Figura 2.11 ilustra uma rotina de busca, onde o usuário digita o texto a ser encontrado e clica na lupa para efetuar a busca. Então, um Java Servlet "OSSearchServlet" é utilizado para efetuar a busca e mostrar os resultados encontrados.

---

<sup>20</sup> Java marca registrada Sun

<sup>21</sup> Visual Basic marca registrada da Microsoft

<sup>22</sup> C++ marca registrada da Ansi



**FIGURA 2.11 - EXEMPLO DE PROGRAMA DE SERVIDOR COMPILADO – JAVA SERVLET**

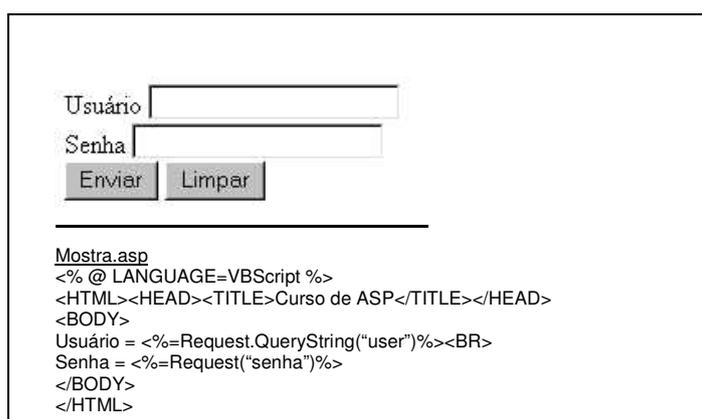
Com o crescimento e a popularidade de páginas Web, surgiram novas tecnologias para processar formulários de requisições, sem a necessidade de iniciar um serviço a cada vez que uma solicitação é enviada ao servidor. Exemplos destas tecnologias de comunicação de servidor Web incluem Java Servlets e Microsoft ASP.NET.

ASP.NET está incluído no modelo de processamento compilado e no modelo de processamento do lado do servidor híbrido, como mostra a Figura 2.10. O modelo compilado cria projeto de aplicações Web, ou coleção de arquivos provendo a necessidade para cada aplicação. O ambiente de desenvolvimento Visual Studio NET's, chamado VB.NET é semelhante ao ambiente de desenvolvimento provido para projeto de aplicações WEB, o qual possui integrado opções de *Debug* de programas, ferramentas gráficas e gerenciamento de projetos, onde cada um destas funções caracteriza a linguagem com processamento híbrido.

Programas compilados no lado do servidor oferecem duas principais vantagens: eles são primeiro compilados e armazenados no formato de linguagem de máquina; eles não precisam ser traduzidos para o formato de linguagem de máquina cada vez que são executados, então eles rodam mais rápido que scripts. Segundo, programas compilados são comumente desenvolvidos em ambiente de desenvolvimento integrado que contém utilitários de debug, tornando-se mais fácil localizar e corrigir erros.

*Script no lado do Servidor.* São programas armazenados no servidor Web para interagir com o ambiente Web. Esses scripts executam as funções requisitadas pelo usuário interagindo com o banco de dados, podendo consultar e/ou modificar o banco de dados, gerando páginas Web dinâmicas e envia-as ao browser da estação de trabalho do usuário. Sua arquitetura de processamento é a mesma arquitetura usada para programas de servidores compilados, diferenciando somente que o processamento no servidor Web é executado por um script de texto interpretado ao invés de um programa compilado.

Os scripts no lado do servidor podem ser desenvolvidos em *Common Gateway Interface* (CGI), que geralmente estão em arquivos separados e escritos em linguagem de script Practical Extraction and Report Language (PERL), que contém características herdadas do Unix, C e Basic. Por exemplo, uma página Web escrita em ASP, chamada MOSTRA.ASP, que mostra na tela o Usuário digitado e sua senha, é ilustrado na Figura 2.12. Neste exemplo o Browse através de uma página HTML, requisita ao servidor web o arquivo mostra.asp, o qual é interpretado pelo servidor Web e sua resposta enviada ao Browser do usuário.



**FIGURA 2.12 - EXEMPLO DE SCRIPT NO LADO DO SERVIDOR**

Além de PERL, existem o Cold Fusion da Macromedia, o Active Server Page (ASP) da Microsoft, o Hypertext Preprocessor (PHP) da Apache Software Foundation's, entre outros.

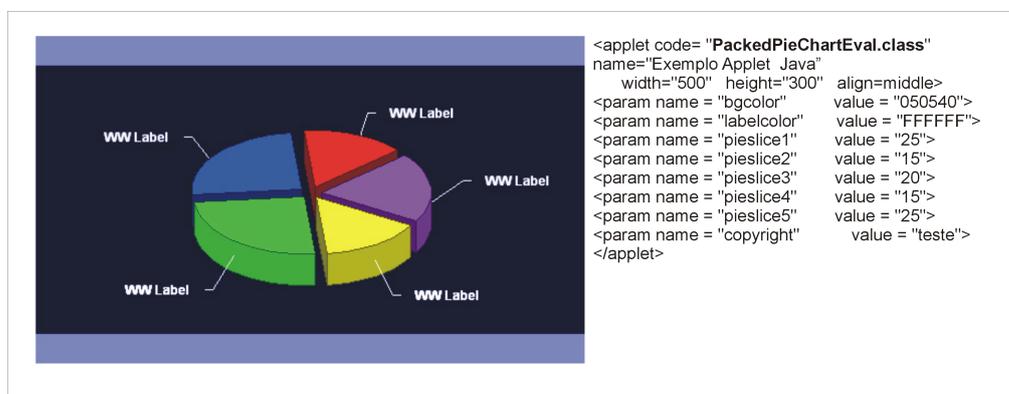
Um servidor Web pode executar uma variedade de scripts escritos em qualquer uma destas tecnologias. O servidor Web sabe qual interpretador de scripts carregar para interpretar estes scripts através da extensão dos arquivos de scripts. Arquivos Cold Fusion possuem extensão de arquivo "cfm", scripts ASP possuem extensão "asp", os scripts PHP possuem extensão "php" e os scripts Perl extensão "pl".

O processamento do lado do cliente é feito através de programas compilados, instalados e executado na estação do cliente por *scripts* criados em uma página Web com comandos HTML interpretados pelo browser do usuário.

O processamento no lado do cliente significa que ao utilizar o browser para solicitar páginas ao servidor, este interpreta scripts ou faz uso de programas executáveis para auxiliar a montagem da requisição no servidor,

O Java Applet é a tecnologia mais popular para criar programas que usuários possam baixar do Web site e instalar na sua estação de trabalho.

A Figura 2.13 ilustra o uso de um Java Applet para geração de um gráfico do tipo pizza numa página Web, onde são passados os valores de cada parte e o código *PackedPieChartEval.Class* monta o gráfico.



**FIGURA 2.13 - EXEMPLO DE PROGRAMA COMPILADO NO LADO DO CLIENTE – JAVA APPLET**

Embora scripts do lado do cliente não possam ser usados por uma página Web para interagir como banco de dados remoto, eles são com frequência usados para validar as solicitações em um formulário HTML enviado para um programa a ser executado no lado do servidor. Por exemplo, um script rodando num usuário pode checar as requisições feitas por um usuário numa página Web, verificando se os dados são válidos e contém valores corretos.

Scripts do lado do cliente também podem ser usados para criar recursos em páginas Web avançadas, incluindo imagens de mapa.

Outro exemplo é JavaScript, que é a linguagem de script do lado do cliente mais usada. Ela também suporta processamento do lado do servidor através de derivas de linguagens específicas de servidor Web.

Javascript provavelmente é melhor que VBScript, já que Javascript é uma linguagem de scripts capaz de rodar perfeitamente em qualquer browser.

*Processamento Híbrido no lado do Servidor* empenha-se em combinar as vantagens dos programas compilados no lado do servidor e as de scripts do lado do servidor; script do lado do servidor é criado, mas não compilado.

Exemplos de processamento híbrido é a tecnologia Java Server Page (JSP), introduzida pela a Sun Microsystems em 1997, e a tecnologia Microsoft ASP.NET.

Concluindo, as várias Tecnologias de Acesso e Desenvolvimento de Acesso à banco de dados via Web e verificando a infra-estrutura já existente na empresa, no sub-item 2.2.3.1 será enfatizada a linguagem *Active Server Page* (ASP).

### **2.2.3.1. ACTIVE SERVER PAGE - ASP**

Active Server Page (ASP) é um ambiente para programação por scripts no lado do servidor, que pode ser usada para criar páginas dinâmicas, interativas e de alta performance. Os scripts rodam no servidor Web e não na estação. É o

próprio servidor que transforma os scripts em HTML padrão, fazendo com que qualquer browser seja capaz de acessar um web site que use ASP. Ela traz para a linguagem HTML todo o poder do acesso à banco de dados, do acesso a arquivos texto, da captação de informações do formulário, da captação de informações sobre o visitante e sobre o servidor, do uso de variáveis, loops.

ASP surgiu juntamente com o lançamento do servidor *Web Internet Information Server* (IIS) 3.0. Esta é uma solução Microsoft, que exige que o servidor rode um sistema operacional Microsoft (MICROSOFT, 2002).

Scripts de ASP permitem aos desenvolvedores de páginas HTML a criação de páginas interativas de uma maneira simples. ASP contém um mecanismo simples para capturar informações do formulário HTML, personalizar um documento HTML e usar recursos HTML específicos do browser.

Scripts Active Server Pages (ASP) ao invés de iniciarem e terminarem com sinal de menor-que (<) e maior-que (>), eles iniciam com <% e terminam com %>. O tag <% é chamado de código de abertura e o %> de código de fechamento. Entre estes tags estão escritos os scripts de servidor. Estes scripts podem ser colocados em qualquer local da página web – mesmo dentro de elementos HTML.

ASP pode chamar componentes de ActiveX para executar tarefas como conectar a um banco de dados ou executar um cálculo (COPELAND, 2000).

Os scripts ASP podem ser escritos usando a linguagem de script *VBScript* ou a linguagem *Jscript*. Entretanto, numa aplicação ASP pode utilizar-se de muitas outras linguagens de criação de scripts existentes, e até mesmo misturar numa mesma aplicação, duas ou mais linguagens diferentes.

A Figura 2.14 ilustra uma arquitetura ASP, em que o browser do usuário requisita para o servidor Web a página *default.asp*. O servidor Web chama o ASP que interpreta o arquivo *default.asp* do início ao fim, executa todos os comando scripts VBScript ou Activex no servidor e envia uma resposta HTML para o browser do usuário. Caso haja necessidade de acesso ao banco de dados, o acesso é feito através da plataforma ODBC.

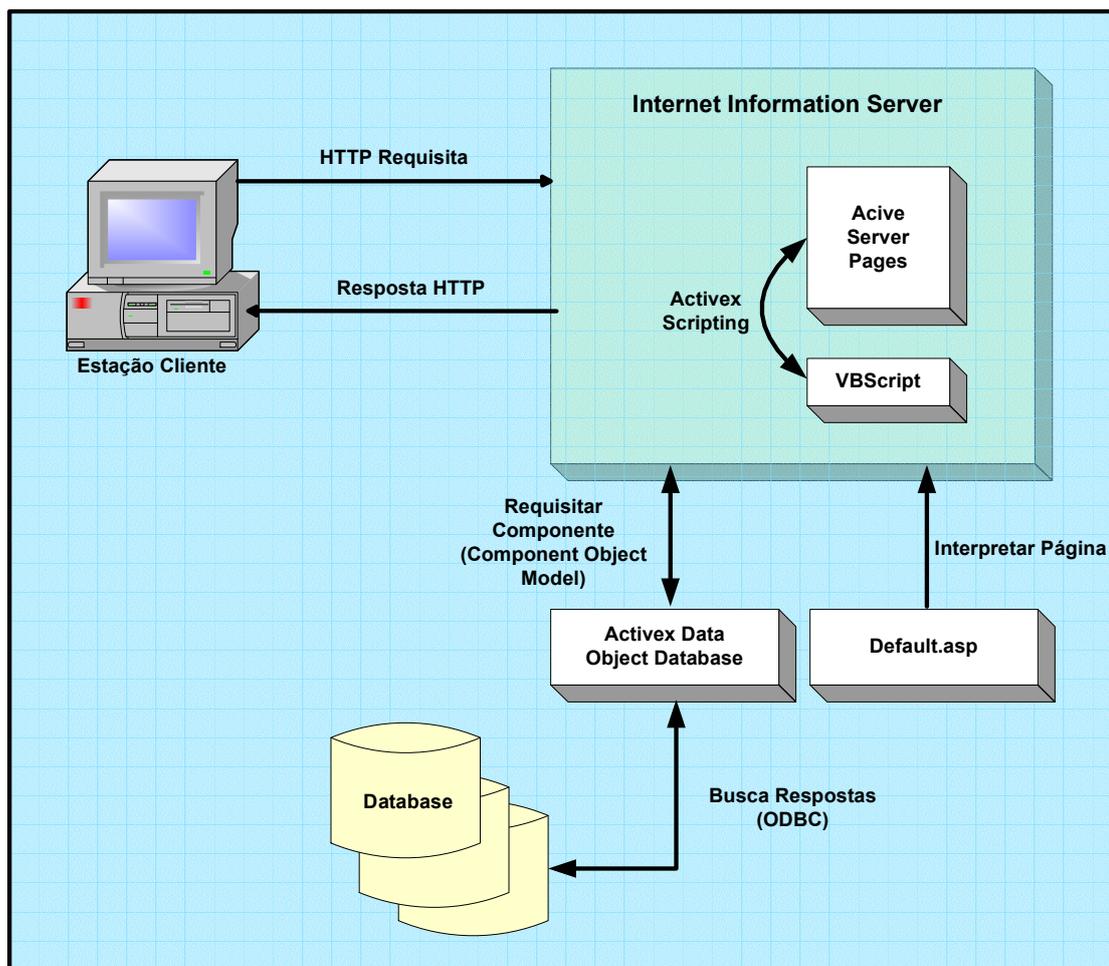


FIGURA 2.14: ARQUITETURA ACTIVE SERVE PAGE (COPELAND,2000;P.23)

Para o funcionamento de scripts ASP é necessário que o servidor possua instalado o servidor Web *Internet Infomation Server* (IIS) versão 3 ou superior. Este servidor deve ser instalado numa máquina NT Server 4 ou 2000. Para o IIS 3.0 ainda é preciso instalar um pacote adicional do ASP para que as aplicações funcionem. A partir da versão 4.0 este pacote já vem incorporado ao IIS.

### 2.3. REALIDADE VIRTUAL

Os termos Realidade e Virtual separadamente em informática não tem característica significativa.

Segundo o dicionário Aurélio (FERREIRA, 1999), a expressão *virtual* significa: “Que se resulta de, ou constitui de uma emulação, por programas de computador, de determinado objeto físico ou equipamento, de um dispositivo ou recurso, ou de certos efeitos ou comportamentos” e a expressão *realidade* significa: “Aquilo que existe efetivamente; real”.

Realidade Virtual pode ser definida de uma maneira simplificada como sendo a forma mais avançada de interface do usuário de computador até agora disponível (KIRNER, 2002b).

Segundo KIRNER (2002b apud AUKSTAKALNIS, 1992), uma definição um pouco mais refinada de realidade virtual é a seguinte : “(...) *uma forma das pessoas visualizarem, manipularem e interagirem com computadores e dados*”.

Realidade Virtual é considerada uma área que busca criar uma nova forma de interação com o computador. Trata-se de uma nova filosofia de interface com o usuário. Nesta nova filosofia, o usuário é colocado "dentro da interface" (KIRNER 2002b apud HED, 1995).

A realidade virtual é caracterizada pela coexistência integrada de três conceitos básicos: imersão, interação e envolvimento.

Segundo KIRNER (2002a), entende-se por imersão o sentimento de se estar dentro do ambiente. Um sistema imersivo é caracterizado pelo uso de capacete de visualização, luva eletrônica, entre outros, e são baseados no uso de salas com projeções das visões nas paredes, teto, e piso. Além dos sentidos visuais, também são importantes os dispositivos ligados aos demais sentidos, como som, posicionamento automático da pessoa e dos movimentos da cabeça, controles reativos, etc. A visualização tridimensional através de monitor é considerada não imersiva.

A Tabela 2.3 ilustra a classificação em níveis de imersão em Realidade Virtual (DINIZ, 1999).

**TABELA 2.3 - NÍVEIS DE IMERSÃO EM REALIDADE VIRTUAL.**

Cavernas: a imersão é total, como nos simuladores de vôo.	Maior
Realidade virtual projetada: é grande o suficiente, permitindo vários usuários.	
Realidade artificial: o usuário não tem contato físico com o sistema.	
Imersão completa do usuário: atingível através de capacete.	
De escritório: um exemplo bem claro é através do monitor do computador.	
Texto: requerimento mínimo, muito utilizado, pouco real.	Menor

Na classificação ilustrada na Tabela 2.3, as cavernas correspondem ao nível de imersão total, ou seja, o mais alto que um usuário pode experimentar. Nestes sistemas o mundo virtual é projetado em grandes superfícies onde é permitido a interação de vários usuários.

A Interação está ligada à capacidade reativa do sistema perante as ações do usuário, ou seja, o computador detecta as entradas do usuário e modifica instantaneamente o mundo virtual em função das ações efetuadas sobre ele. As pessoas são cativadas por uma boa simulação em que as cenas mudam em resposta aos seus comandos, que é característica mais marcante dos vídeos games. Para que um sistema de Realidade Virtual pareça mais realista, o ambiente virtual inclui objetos simulados. Outros artifícios para aumentar o realismo são empregados, por exemplo, a texturização dos objetos do ambiente e a inserção de sons tanto ambientais quanto de sons associados a objetos específicos. (VALERIO NETTO, 2002).

A idéia de envolvimento está ligada ao grau de motivação para o engajamento de uma pessoa em determinada atividade. O envolvimento pode ser passivo ou ativo. O envolvimento passivo ocorre gerando a interação do usuário com o sistema que acontece de maneira análoga a sua interação com um livro, a televisão ou cinema. Ou seja, o usuário tem minimizado os recursos de interação com o sistema, não podendo tomar decisões ou mudar uma cena pré-definida na programação. O envolvimento ativo ocorre de maneira análoga ao vídeo game, permitindo ao usuário a possibilidade de alterar as cenas ou tomar decisões e interferir no curso do programa. A Realidade Virtual tem potencial para os dois tipos de envolvimento ao permitir a exploração de um ambiente virtual e propiciar a interação do usuário com mundo virtual dinâmico.

KIRNER (2002b apud BURDEA, 1994; JACOBSON, 1991; KRUEGER, 1991), explica que Realidade Virtual é “(...) *uma técnica avançada de interface, onde o*

*usuário pode realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais”.*

Em Realidade Virtual podemos trabalhar com os sentidos de percepção visual, retorno auditivo, o tato e a força de reação.

Apesar desses sentidos, o trabalho trata da visualização de informações, por se tratar de tecnologia recente e atender aos requisitos da empresa.

#### **2.4. VISUALIZAÇÃO DE DADOS**

Empresas e Instituições têm feito uso intensivo de computadores para armazenamento de dados, ocasionando a necessidade de utilização de sistemas para visualização de informações. O fato do ser humano ter mais facilidade para analisar representações gráficas do que dados representados em tabelas ou textos repercutiu no impacto da visualização de dados. Eles são muito bons em processar cenas visuais e muito ruins em processar dados tabulares (MENDONÇA NETO, 2001).

Para descrevermos sobre Visualização de Dados e/ou Informações, antes introduziremos os conceitos Dados, Informação e Visualização.

Segundo WIRTH (1999) dados representam uma abstração da realidade, no sentido de que certas propriedades e características do objeto real são desprezadas, por serem inexpressivas ou irrelevantes para a solução do problema corrente em particular.

Informação é uma abstração informal (isto é, não pode ser formalizada através de uma teoria lógica ou matemática), que representa algo significativo para alguém através de textos, imagens, sons ou animação. Não é possível processar informação diretamente em um computador, ou seja, precisa-se primeiro reduzir a informação a dados (SETZER, 2001).

Até meados de 1999, visualização significava a construção de uma imagem visual na mente. Em 1999 seu significado tornou-se mais amplo, ou seja, uma representação gráfica de dados ou conceitos (WARE, 2000).

O fato de GERSHON (1998) descrever que Visualização consiste em transformar dados, informações e conhecimento na forma visual utilizando a capacidade de visualização natural do homem, entende-se que visualização proporciona a interação entre poderosos sistemas de processamento de informações – a mente humana e o computador moderno.

Segundo CARD (1999), a Visualização de Informações é o uso de representação visual interativa e suportada por computador, de dados abstratos para ampliar a cognição.

No item 2.4.1 é descrito como é feita a construção de visualização de informações, no item 2.4.2 é definido visualização de informações, no item 2.4.3 como é feito a representação dos dados e é citado exemplos de visualizações geradas a partir de um software existente para construção de visualização de informação e ferramentas existentes para utilização de conceitos de realidade virtual na construção de visualização de informações. No que se refere à Modelagem de Cenas utilizar-se-á o subitem 2.4.4 para relatar sobre como são implementados mundos virtuais, o conceito de grafo de cena, a linguagem *Virtual Reality Modeling Language* (VRML) e a realidade virtual como interface para facilitar a visualização de dados. O subitem 2.4.5 que trata de Cenas Dinâmicas, considera o conceito de nó em Realidade Virtual, enfatizando instanciação, Rotas (nó Route) que faz tratamento de eventos, temporizadores (nó TimeSensor ), os interpoladores utilizado para movimentos e o nó Script. O subitem 2.4.6 discute Gráficos dinâmicos, utilizando-se da apresentação de gráficos em Java ou JavaScripts.

### 2.4.1. CONSTRUÇÃO DE VISUALIZAÇÃO DE INFORMAÇÕES

Na construção de Visualização de Informações pode se fazer uma análise do conteúdo dos dados. Esta análise utiliza-se de exploração qualitativa de mensagens e informações, proporcionando uma compreensão aprofundada sobre seus significados.

No processo de construção de visualização de informações existem algumas etapas para transformar os dados em formas visuais como ilustra a figura 2.15.

As setas que partem da esquerda para a direita representam as etapas em que ocorrem a transformação de dados brutos e cenas visuais para interpretação humana. E, as setas que partem da direita para a esquerda representam modificações efetuadas pelo ser humano. Transformações de Dados mapeia os dados Brutos em formato adequado e particular às ferramentas que serão usadas para construir estruturas visuais, resultando em Tabelas de Dados (MENDONÇA NETO, 2001).

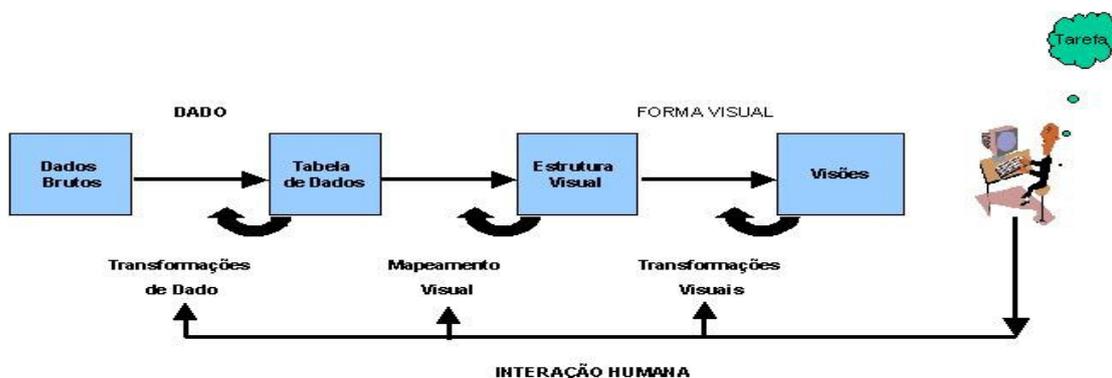


FIGURA 2.15 - PROCESSO MAPEAMENTO VISUAL DE DADOS (CARD, 1999)

O Mapeamento Virtual transforma a Tabela de Dados em Estruturas Visuais que combinam propriedades gráficas, pontos de referência na tela, e organização espacial destes dados. Finalmente, as Transformações Visuais criam Visões das estruturas montadas permitindo especificação de parâmetros gráficos como posição e escala de visualização. O Ser Humano pode interagir como parâmetro de controle para mudar ou restringir as visões de acordo com seu interesse.

## 2.4.2. VISUALIZAÇÃO DE INFORMAÇÕES

Segundo CARD (1999), a Visualização de Informações, pode ser construída utilizando formas espaciais e/ou formas estruturais. Como formas espaciais podemos mencionar as unidimensionais, bidimensionais, tridimensionais, multidimensionais e como as estruturais existem a temporal, hierárquica e em rede.

A Visualização de Informações unidimensional é baseada na utilização de um atributo, focando um ponto na reta (por exemplo, linha de tempo) ou 1 (um) elemento no monitor (por exemplo, conjunto de documentos). A Visualização de Informações bidimensional é baseada na utilização de 2 (dois) atributos, onde um atributo contém uma percepção da posição e outro atributo contém propriedades como altura, largura, área, cor, etc. (por exemplo, diagramas de barras e gráficos de pizza). A visualização de Informações tridimensional é baseada em três atributos, onde além dos atributos utilizados na visualização bidimensional, utiliza-se um terceiro atributo que incorpora freqüentemente a noção de volume. Aqui se tem a aplicação imediata de realidade virtual. Muitas vezes, pela facilidade da representação de um espaço bastante grande, usa-se realidade virtual para a visualização de informações bidimensionais, dispondo-se gráficos de barras, por exemplo, espalhados pelo cenário do mundo virtual. A visualização multidimensional se dá pela incorporação de quatro ou mais atributos aos dados, que pode usar três atributos espaciais e outros como cor, tonalidade, forma, etc., usando a visualização tridimensional com realidade virtual para navegar pelo cenário e interagir com as representações dos dados. As demais formas de visualização da informação, embora com grau de importância, não serão abordados neste trabalho, devido ao fato da ênfase do trabalho está em usar conceitos de realidade virtual para construção de visualização de informações e a utilização de dados da empresa em questão.

### 2.4.3. REPRESENTAÇÃO DE DADOS

Um dos ganhos com a utilização de visualização de Informações é a quantidade de informações que podem ser interpretadas de forma rápida se as mesmas forem muito bem apresentadas.

Segundo WARE (2000), existem vários tipos de representações, podendo ser dividido em: forma (ex: largura, tamanho, curvatura, orientação), cor (ex: tonalidade, intensidade), movimento (ex: piscar, direção do movimento), posição espacial (ex: côncavo/convexo, 2D, 3D), etc.

Na construção de representação de informações de forma unidimensional, utilizam-se textos para ilustrar notações matemáticas, símbolos, frases. A Tabela 2.4 ilustra um exemplo de uma representação de visualização de informações unidimensional utilizando notação matemática para a descrição das quantidades vendidas do produto A.

**TABELA 2.4 – REPRESENTAÇÃO TEXTUAL DE VENDA DE PRODUTOS A**

VPA = {10,20,30}
VPA = Vetor contendo as Vendas de Produtos A.

Para Representação de Visualização de Informações de forma bidimensional utiliza-se tabelas ou gráficos. A Tabela 2.5 ilustra um exemplo de Representação de Visualização de Informações utilizando tabelas, onde são descritas as quantidades vendidas do produto A, do produto B e do produto C, em um determinado dia.

**TABELA 2.5 – REPRESENTAÇÃO TABULAR DE VENDA DE PRODUTOS EM UM DIA**

	<b>Produto A</b>	<b>Produto B</b>	<b>Produto C</b>
<b>Dia 11</b>	10	20	50

A Figura 2.16 ilustra um exemplo de representação de visualização de informações bidimensionais através de gráficos, onde é construída a visão da quantidade vendida do produto A, do produto B e do produto C, em determinado dia.

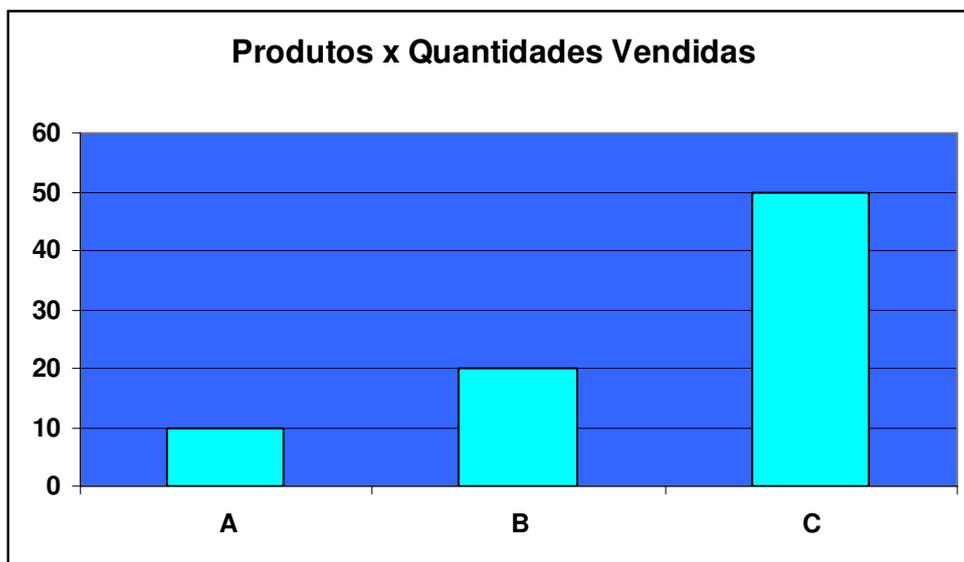


FIGURA 2.16- REPRESENTAÇÃO GRÁFICA DE VENDA DE PRODUTOS NUM DIA

A Visualização de Informação de forma tridimensional utiliza-se de gráficos. Os gráficos podem ser construídos em várias formas, entre elas citamos a forma de linha, barras, dispersão, pizza, cone, cilindro, pirâmide, etc.

A Figura 2.17 lustra um exemplo de representação de visualização de informações tridimensionais através de gráficos, onde é construída a visão da quantidade vendida do produto A, do produto B e do produto C, num período de três dias.

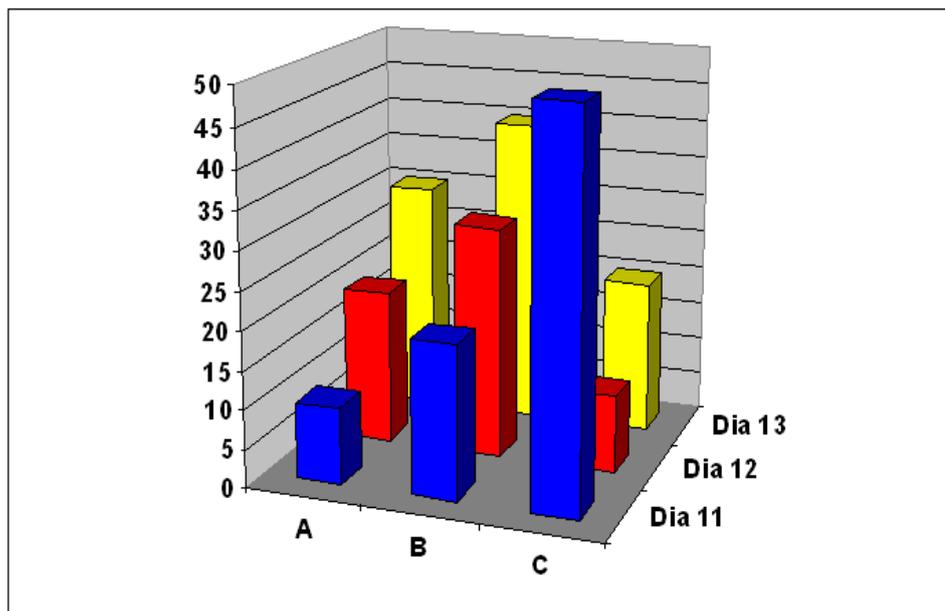


FIGURA 2.17- REPRESENTAÇÃO GRÁFICA DE VENDA DE PRODUTOS DE ALGUNS DIAS

Para a representação de Visualização de Informações pode-se fazer o uso da *World Wide Web* para a construção da Visualização de Informações.

A Visualização de Informações através da *World Wide Web*, ou seja, hipertexto, ocorre com a utilização de páginas HTML. Em páginas HTML podemos usar visualizações textuais, tabulares e gráficas. A figura 2.18 ilustra uma visualização de informações tridimensionais com a utilização de hipertexto.

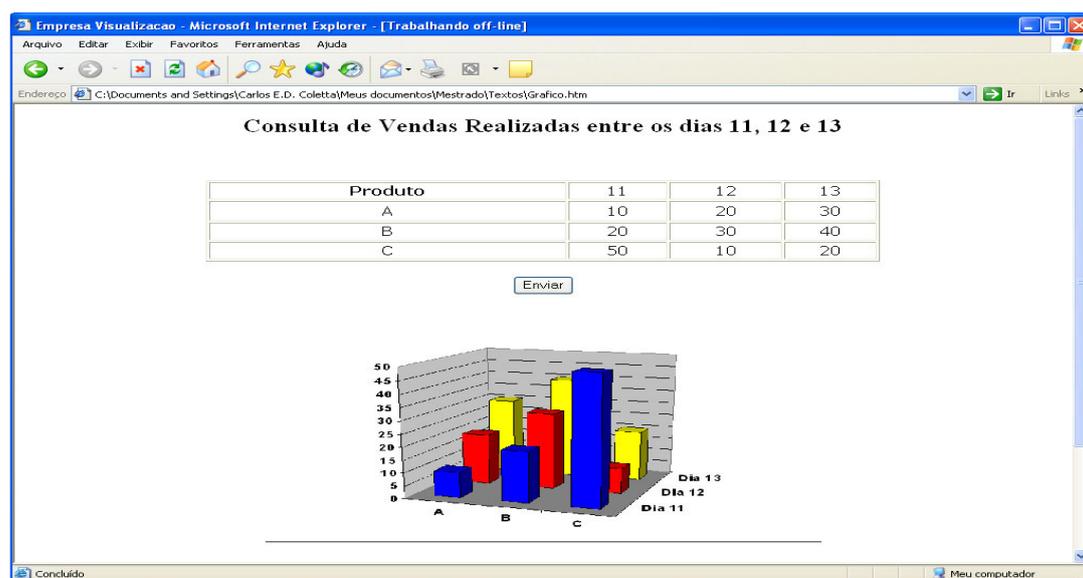


FIGURA 2.18- REPRESENTAÇÃO DE HIPERTEXTO VENDA DE PRODUTOS NOS DIAS 11, 12, 13

Existem algumas empresas de softwares que comercializam Softwares que auxiliam a Visualização de Informações através de Gráficos onde são gerados gráficos em várias dimensões, mas estático não permitindo a interação do usuário. A proposta deste trabalho é a geração de Visões utilizando softwares livres e/ou de baixo custo que permita a interação do usuário.

Utilizamos as Figuras 2.19, 2.20 e 2.21 para ilustrar exemplos de gráficos gerados com dois softwares existentes no mercado, porém vale destacar que ambos não são softwares livres e sim comercializados, um é o Visual Discovery ADVIZOR Solutions<sup>23</sup> e o outro é o OCULUS<sup>24</sup>.

A Figura 2.19 ilustra um gráfico, gerado pelo Visual Discovery, que proporciona a visualização dos chamados ocorridos no período de uma semana, permitindo assim visualizar qual o período que ocorre o maior número de chamados no helpdesk, ou seja, quando ocorre o *staff*.

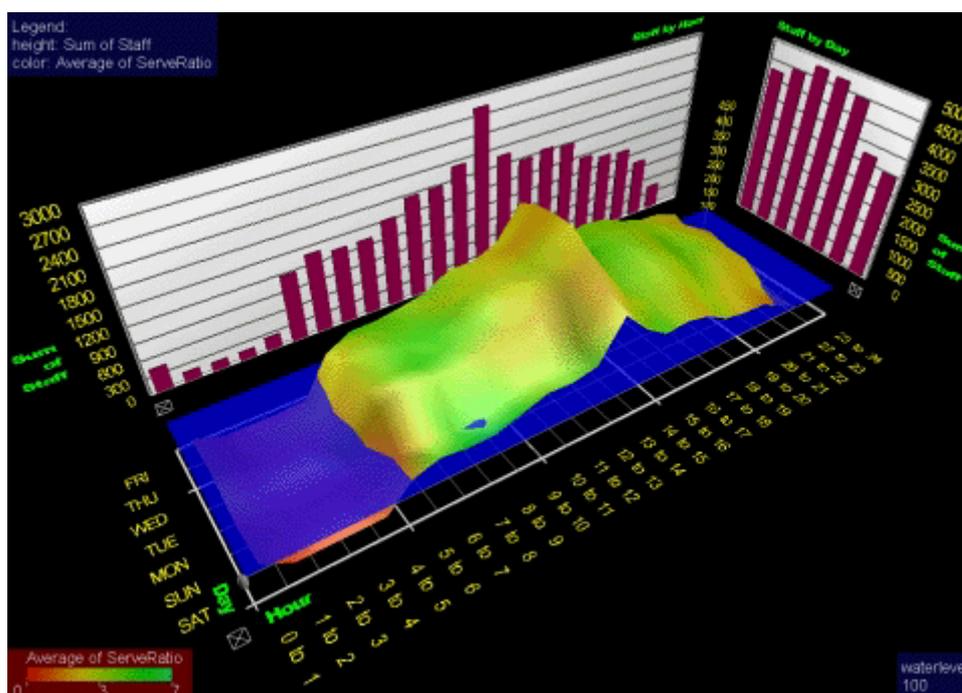


FIGURA 2.19 - GRÁFICO DE QUANTIDADE DE CHAMADOS

<sup>23</sup> Marca Registrada da ADVISOR SOLUTIONS, INC.  
<http://www.advizorsolutions.com>

<sup>24</sup> Marca Registrada da OCULUS INFO INC.  
<http://www.oculusinfo.com.com>

A Figura 2.20 ilustra um gráfico, gerado pelo Oculus, que proporciona a visualização dos da quantidade de calorias energéticas tem em cada alimento agrupados por grupo de alimentos.

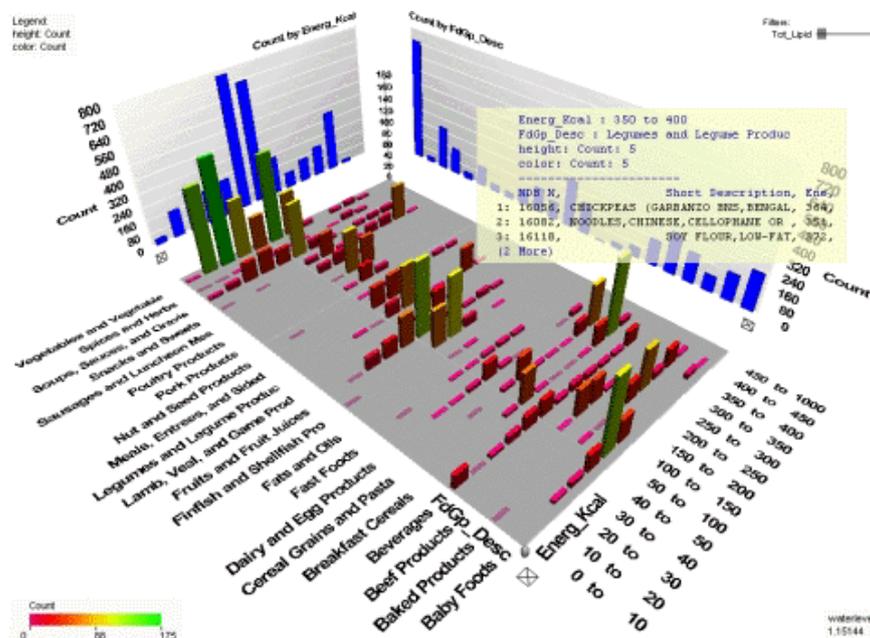


FIGURA 2.20- GRÁFICO QUANTIDADE DE CALORIAS ENERGÉTICAS DE ALIMENTOS

A Figura 2.21 ilustra um gráfico, gerado pelo Oculus, que proporciona a visualização dados dos valores de vendas de cada vendedor e suas respectivas comissões, determinado assim qual o vendedor com maior comissão e o vendedor com a menor comissão, mostrando assim a performance dos vendedores.

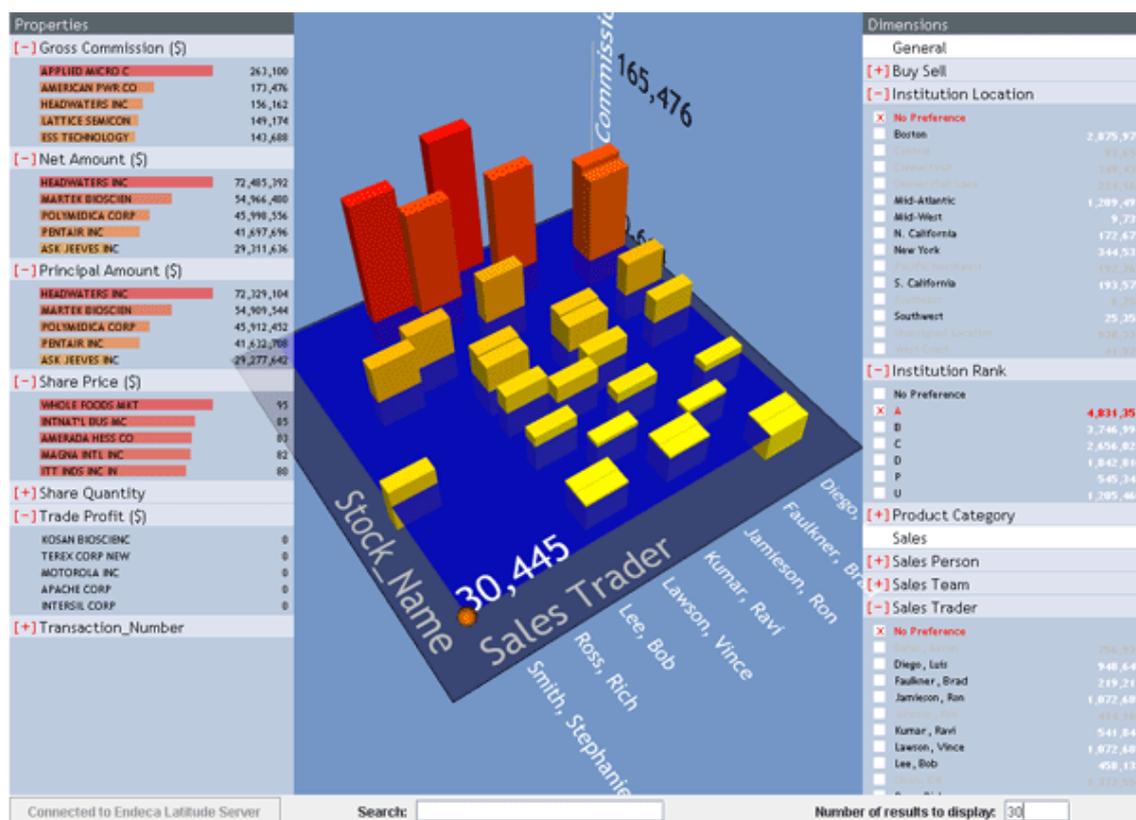


FIGURA 2.21- GRÁFICO PERFORMANCE DE VENDAS DE UMA EMPRESA

Na construção de um software para Visualização de Informações pode-se usar conceitos de Realidade Virtual, a qual faz o uso de cenas e permitir que o usuário faça interação com os dados da cena.

Ferramentas existentes para modelagem de dados utilizando realidade virtual são: VREAM, Superscape, Sense8 WorldToolkit, *Virtual Reality Modeling Language* (VRML), Spazz 3D, Shout 3D e Studio 3D Max.

O contexto deste trabalho está na utilização de visualização de informações utilizando realidade virtual em gráficos dinâmicos, utilizando a linguagem VRML. A Linguagem VRML utilizará os dados, pesquisados no banco de dados através da linguagem ASP, para efetuar a Modelagem de Cenas Dinâmicas gerando Gráficos, os quais serão usados tanto pela Gerência Comercial quanto pelos clientes para tomada de Decisão.

#### 2.4.4. MODELAGEM DE CENAS

Mundo virtual é um termo utilizado que diz respeito ao mundo digital criado a partir de técnicas de Computação Gráfica (VALERIO NETTO, 2002).

O Mundo virtual pode ser projetado em grandes superfícies onde é permitida a interação de vários usuários.

A criação de um mundo virtual é a expressão de componentes do mundo real através de um modelo matemático que permita representar os elementos bem como as suas interações. Esse modelo em Realidade Virtual é denominado Grafo de Cena, pois descreve os elementos do mundo virtual segundo um grafo (WASH, 2001).

O grafo de cena corresponde a uma degeneração do grafo, pois os elementos que constituem o mundo virtual devem ser organizados numa estrutura hierárquica, como ilustra a Figura 2.23.

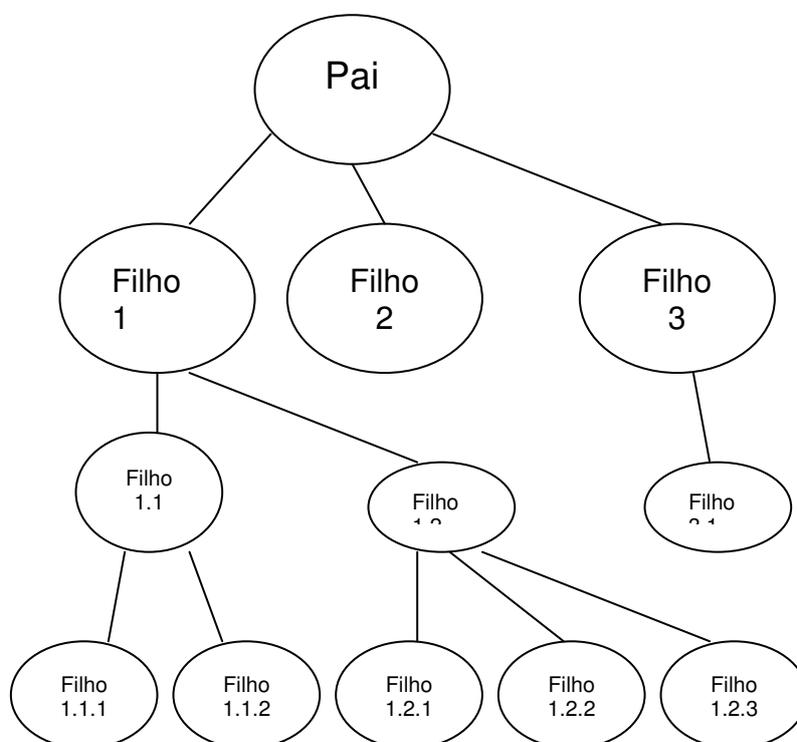


FIGURA 2.23 - EXEMPLO DE UM MODELO DE REPRESENTAÇÃO DE MUNDOS VIRTUAIS

A Figura 2.23 apresenta um modelo de representação de mundos virtuais. Para construção de um mundo virtual segundo este modelo devemos ter em mente que as características são definidas no percorrimento em pré-ordem da árvore gerada.

As interfaces de realidade virtual podem ser desenvolvidas em diferentes linguagens de programação, sendo a Linguagem VRML uma das possibilidades.

A linguagem *Virtual Reality Modeling Language* (VRML) é uma linguagem de programação que permite a produção de ambientes tridimensionais (3D) para *World Wide Web* (WWW). Em 1994, em Geneva – Suíça, na primeira conferência anual de WWW (*World Wide Web*) aconteceram às primeiras discussões acerca de interfaces 3D para a Internet. Durante uma sessão de discussão da necessidade de ferramentas para a especificação de mundos tridimensionais foi sugerido um termo *Virtual Reality Modeling Language* (VRML), em analogia à linguagem *Hyper Text Markup Language* (HTML).

As especificações da versão 1.0 da linguagem VRML foram discutidas em uma lista de e-mail, que levou a um consenso para esta primeira versão (WEB3D CONSORTIUM, 2003). O problema que se apresentava então era o uso de uma ferramenta disponível que pudesse ser adequada às necessidades da VRML 1.0. As análises e discussões relativas às diversas tecnologias disponíveis levaram ao consenso do uso do *Open Inventor*. Esta tecnologia, produzida pela Silicon Graphics Incorporation (SGI), utiliza arquivo de formato ASCII para a modelagem de elementos gráficos. O Formato do arquivo para o *Open Inventor* (*Inventor File Format*) oferece suporte à descrições de elementos tridimensionais através da utilização de objetos poligonais, de brilho, de materiais, de propriedades de ambiente e de efeitos reais. Assim, os objetos que constituem o *Inventor File Format*, com extensões pra conexão em redes de computadores, definem a base do VRML.

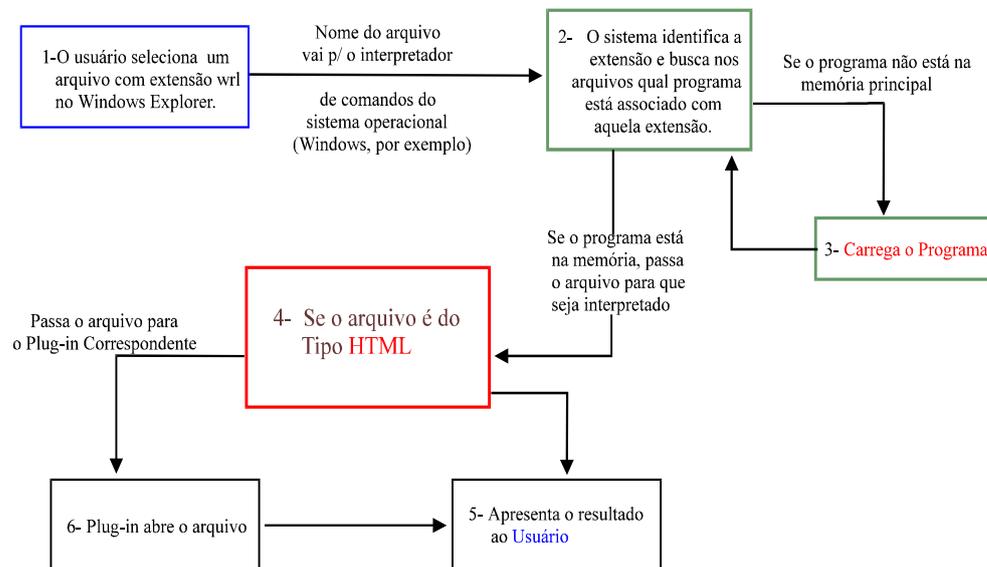
Em 1995 foram feitas pequenas modificações ao VRML 1.0, até ser constatada a necessidade de lançar uma nova versão com suporte à animação, interação e modelagem comportamental. O projeto da linguagem VRML 1.0 foi

modificado, mas ainda não atendia às necessidades de animação e interatividade inicialmente propostas. No encontro da SIGGRAPH 96, em New Orleans, em agosto de 1996, foi apresentada a versão VRML 2, feita conjuntamente pela Silicon Graphics, Sony e Mitra, o qual foi aprovado no mesmo ano pela *International Standards Organization* (ISO) como o padrão da linguagem VRML 2.

Os arquivos da Linguagem VRML possuem a extensão .wrl. Um arquivo VRML é composto de *header*, comentários, nós contendo campos e valores, rotas e protótipos.

A linguagem VRML pode ser combinada com HTML, Java, ou Javascript. Para utilização de VRML com HTML o browser precisa oferecer suporte à linguagem VRML. Quando o browser não oferece suporte à VRML tem de ser utilizado *Plugin*<sup>25</sup>.

A Figura 2.24 ilustra o conceito de uso de plug-in (visualização) pelo browser.



**FIGURA 2.24 - DINÂMICA DE OPERAÇÃO DO NAVEGADOR COM UTILIZAÇÃO DE PLUG-IN**

<sup>25</sup> *Plug-in* é um pequeno programa que, instalado num browser, permite que este desempenhe uma nova função. Alguns sites contêm informações, vídeos, jogos, arquivos criados em programas 3D que só podem ser acessados ou mostrados pelo navegador quando este possui o plug-in necessário instalado.

A Figura 2.24 ilustra o acesso a um mundo virtual quando há necessidade de uso de *plug-in*. Acrescenta-se a esse esforço a tarefa de busca do código a ser executado, considerando o desempenho do sistema, e a possibilidade da aplicação utilizar recursos multimídia.

A linguagem VRML oferece elementos da computação gráfica para a implementação das propriedades virtuais das interfaces interativas em mundo virtuais.

Os elementos de computação gráfica podem ser classificados em software e hardware. Devido ao interesse em modelagem de sistemas virtuais, consideram-se somente os elementos de software. Serão tratadas as formas e figuras geométricas, suas respectivas representações e transformações lineares de translação e rotação. Esses elementos de software são denominados nós.

#### **2.4.5. CENAS DINÂMICAS**

A construção fundamental de um arquivo VRML chama-se nó. Existem nós que são objetos – Cylinder, Cone, Box, SpotLight e nós que são usados como containers que armazenam nós com alguma relação lógica. Um nó Shape, por exemplo, contém um nó (geometry) que indica a forma do objeto e outro (appearance) que controla a sua aparência. Estes nós podem, por sua vez, conter outros, e assim por diante.

Em VRML existem definidos uma série de nós que possibilitam alguns recursos especiais, como ligações com outros arquivos (hyperlinking), inclusão de objetos definidos em outros arquivos, e detecção de colisões. Cada nó contém campos que mantêm os dados que o caracterizam como um elemento único dentro do arquivo.

Em VRML, podemos referenciar um nó várias vezes, chamado de instanciação, utilizando-se, para isso, as palavras DEF e USE, onde DEF define um nome e

cria um objeto do tipo especificado e USE indica que uma referência a um nó previamente definido será utilizada.

Um exemplo do uso das palavras USE e DEF é ilustrado na Tabela 2.6. No exemplo através do campo *children* se especifica-se outro nó e utiliza-se a palavra DEF para definir um nome como Computador5, no qual foi especificado o endereço de um outro objeto através do nó *inline*. Quando utilizado o USE ele chama esse objeto Computador5 fazendo só sua *translation*, não fazendo nenhuma alteração no objeto já criado.

**TABELA 2.6 – EXEMPLO DE UTILIZAÇÃO DE USE E DEF EM VRML**

```

Transform {
  Translation 12.0 0.0 -5.0
  Children DEF Computador5 Inline { url "computador5.wrl" }
},
Transform {
  Translation 0.0 0.0 -5.0
  Children USE Computador5
},
Transform {
  Translation 4.0 0.0 -5.0
  Children USE Computador5
},
Transform {
  Translation 8.0 0.0 -5.0
  Children USE Computador5
},

```

Uma forma básica para a criação de cenas dinâmicas é rotação de eventos entre campos de maneira arbitrária combinando nós interpoladores e nós sensores.

Um nó pode enviar e receber eventos. Um modo simples de definir o caminho entre o evento gerado por um nó e o nó que recebe o evento é o ROUTE. Um evento de um nó pode ser roteado para um evento do mesmo tipo de outro nó conforme mostra a sintaxe ilustrada a baixo.

#### Exemplo de Sintaxe de Route

**ROUTE** NodeName1.eventOutName\_changed **To** Nodename2.set\_eventInName,

sendo NodeName1 o nome do nó que envia o evento, eventOutName é o nome do evento de saída, NodeName2 o nome do nó que recebe o evento e eventInName o nome do evento de Entrada.

Quando um evento é roteado para outro, eles não precisam ter o mesmo nome, mas tem que ser, necessariamente, do mesmo tipo.

Os eventos dos nós na sintaxe ilustrado no exemplo acima precisam ter sido definidos antes no arquivo DEF.

O Nó TimeSensor dispara eventos conforme o tempo passa, sendo usado para muitos propósitos, como: animações e simulações, controle de atividades periódicas e também iniciar um simples evento de ocorrência, como um alarme de relógio. A Tabela 2.7 ilustra a sintaxe do nó TimeSensor.

**TABELA 2.7 – SINTAXE DO NÓ TIMESENSOR**

<b>Sintaxe:</b>		
TimeSensor {		
enabled	TRUE	
cicleInterval	0	
loop	FALSE	
startTime	0	
stopTime	0	
}		

A Tabela 2.8, apresenta os comentários sobre os campos utilizados no nó TimeSensor.

**TABELA 2.8 - CAMPOS DO NÓ TIMESENSOR**

Campo	Tipo	Comentário
Enabled	SFBool	liga e desliga este nó.
CicleInterval	SFFloat	O intervalo para um ciclo completo.
Loop	SFBool	define se o evento será repetitivo ou não.
StartTime	SFFLOAT	recebe um evento especificando quando deve começar cada ciclo
StopTime	SFFloat	recebe um evento especificando quando deve terminar cada ciclo

Os interpoladores são usados para definir um dicionário de valores para um *Key* variando de 0.0 a 1.0, para um *KeyValue*, o qual pode conter vários tipos de dados dependendo do Interpolador.

Os tipos de Interpoladores existentes são: *ColorInterpolator*, *CoordinateInterpolator*, *NormalInterpolator*, *OrietationInterpolator*, *PositionInterpolator* e *ScalarInterpolator*.

O Nó *ColorInterpolator* é um interpolador onde se coloca uma lista de valores RGB no campo *KeyValue*. A Tabela 2.9 ilustra sua sintaxe.

**TABELA 2.9 – SINTAXE DO NÓ COLORINTERPOLATOR**

```
ColorInterpolator {
    Key []
    KeyValue []
}
```

O nó *CoordinateInterpolator* é um interpolador onde se coloca valores 3D no campo *KeyValue*. A Tabela 2.10 ilustra sua sintaxe.

**TABELA 2.10 – SINTAXE DO NÓ COORDINATEINTERPOLATOR**

```
CoordinateInterpolator {
    Key []
    KeyValue []
}
```

O nó *NormalInterpolator* é um interpolador onde se coloca uma lista de vetores 3D no campo *KeyValue*. A Tabela 2.11 ilustra sua sintaxe.

**TABELA 2.11 – SINTAXE DO NÓ NORMALINTERPOLATOR**

```
NormalInterpolator {
    Key []
    KeyValue []
}
```

O nó *OrientationInterpolator* é um interpolador onde se coloca uma lista de valores de rotação, como coordenadas 3D de um ângulo, no campo *KeyValue*. A Tabela 2.12 ilustra sua sintaxe.

**TABELA 2.12 – SINTAXE DO NÓ ORIENTATIONINTERPOLATOR**

```
OrientationInterpolator {
    Key []
    KeyValue []
}
```

O nó *OrientationInterpolator* é usado na animação de objetos, rotacionando de acordo com os valores colocados no campo *KeyValue*.

O nó *PositionInterpolator* é um interpolador utilizado para intercalar frames, cada qual com sua posição. A Tabela 2.13 ilustra sua sintaxe.

**TABELA 2.13 – SINTAXE DO NÓ POSITIONINTERPOLATOR**

```
PositionInterpolator {
    Key []
    KeyValue []
}
```

O nó *ScalarInterpolator* é um interpolador onde se coloca uma lista de valores de pontos flutuantes no campo *KeyValue*. A Tabela 2.14 ilustra sua sintaxe.

**TABELA 2.14 – SINTAXE DO NÓ SCALARINTERPOLATOR**

```
ScalarInterpolator {
    Key []
    KeyValue []
}
```

O nó *Script* é usado para programar comportamento em uma cena. O nó recebe eventos que significam uma ação do usuário, executa um código que realiza algum tipo de cálculo em resposta ao evento e envia eventos para outras partes da cena.

Um script pode especificar uma ação cada vez que ele recebe um evento (use-se o nó *ROUTE* para enviar um evento a um *Script* como com qualquer outro nó). Além disso, você pode definir inicialização e procedimentos de paralisação. Em cada ação um script pode gerar múltiplos eventos.

Ao contrário de outros nós, no nó *Script* pode-se definir os eventos de recebimento e envio de nó. Um nó *Script* também pode ter campos. Os campos trabalham como variáveis locais persistentes ao *Script* e pode ser usado para definir um estado interno.

Um script pode receber um número ilimitado de eventos. Os valores destes eventos são armazenados em tipos de variáveis *eventIn*. O nome das variáveis especifica o nome dos eventos que o Script pode receber. Estas variáveis são somente de leitura, não se pode fixar um valor a uma destas variáveis, pode-se apenas ler seu valor atual. Também não há nenhum limite ao número de eventos que um Script pode gerar. Os valores associados aos eventos gerados são armazenados em tipos de variáveis *eventOut*. As variáveis e os campos são persistentes. A Tabela 2.15 ilustra a sintaxe de um nó Script.

**TABELA 2.15 – SINTAXE DO NÓ SCRIPT**

<pre> Script {   url []   directOutput FALSE   mustEvaluate FALSE   #and any number of :   eventIn type eventName   eventOut type eventName   field type fieldName initialValue } </pre>
--

O nó *Script* permite definir comportamentos usando linguagens de programação como a JAVA, JavaScript, ou o VRMLScript recentemente proposto pela SiliconGraphics.

#### 2.4.6. GRÁFICOS DINÂMICOS

Dentre as formas de visualização de informações, pode-se utilizar a linguagem VRML, combinando-a com Java ou JavaScript para criação de Gráficos Dinâmicos acessando banco de dados.

Nota-se que há dois modos para acessar um programa Java ou JavaScript através de VRML: um, denominado *script* interno, que se comunica com Java “dentro” das cenas VRML através do nó script e o outro, denominado *script* externo, onde *External Authoring Interface* (EAI) define uma interface Java ou JavaScript para applets que se comunicam através de um HTML Web Browser externo. Um applet EAI pode tanto passar quanto receber mensagens de cenas VRML embutidas em uma página HTML (BRUTZMAN, 1998).

Dentro do contexto da pesquisa dar-se-á ênfase ao modo *script* interno de acessar Java via VRML. A Tabela 2.16 ilustra um trecho do código VRML onde é definido um nó Script contendo as definições de variáveis do tipo *eventIn* e do tipo *eventOut* e fazendo chamada a uma rotina em Java chamada “Escala.class”. As variáveis do tipo *eventOut* referem se a valores de escala e posição para cada dia da semana

**TABELA 2.16 – CÓDIGO VRML ACESSANDO O PROGRAMA EM JAVA**

```

DEF S Script{
    eventIn SFFloat timeStamp
    eventOut SFVec3f segundaEscala
    eventOut SFVec3f segundaPosicao
    eventOut SFVec3f tercaEscala
    eventOut SFVec3f tercaPosicao
    eventOut SFVec3f quartaEscala
    eventOut SFVec3f quartaPosicao
    eventOut SFVec3f quintaEscala
    eventOut SFVec3f quintaPosicao
    eventOut SFVec3f sextaEscala
    eventOut SFVec3f sextaPosicao
    eventOut SFVec3f sabadoEscala
    eventOut SFVec3f sabadoPosicao
    eventOut SFVec3f domingoEscala
    eventOut SFVec3f domingoPosicao
    url "Escala.class"
}

```

Com a utilização do programa Java são criadas e atualizadas as cenas, com os dados obtidos no banco de dados. Esta atualização acontece em tempo real.

A Tabela 2.17 ilustra o programa fonte desenvolvido em Java. O Código completo se encontra no Anexo C.

**TABELA 2.17 – CÓDIGO FONTE JAVA**

```

// import com.buk.pacotes.SetaDados;
import java.awt.*;
import java.util.*;
import java.awt.event.*;
import vrmf.*;
import vrmf.node.*;
import vrmf.field.*;

public class Escala extends Script
{
    private SFVec3f escalaDiasDaSemana[];
    private SFVec3f posicaoDiasDaSemana[];

    // Variáveis de controle do Java que são associadas aos tipos do VRML
    // Para cada nó VRML há uma função Java que copia o valores
    // Escala para os dias da semana

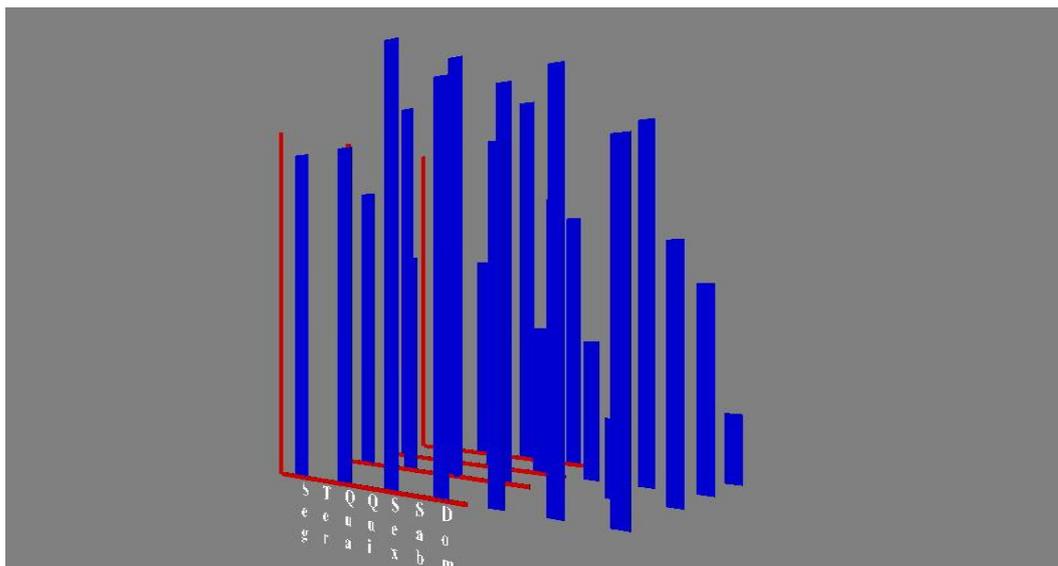
    // Lista de eventos que o Cortona envia para o Java quando o nó
    // Script do VRML é acionado

    // Obtém os valores iniciais da cena VRML
    // É chamado quando o Script é criado
    // Cada um dos campos recebe os valores iniciais definidos na cena

    // Valores buscados no BD
    // Atualiza Cena
    //
}

```

A Figura 2.25 ilustra um gráfico dinâmico gerado usando programa fonte desenvolvido em VRML fazendo chamada a um programa Java, ilustrado na tabela 2.17.

**FIGURA 2.25 - GRÁFICO 3D COM VRML FAZENDO ACESSO A JAVA**

### **3. SISTEMA DE CONSULTA DE LOCALIZAÇÃO DE CARGAS FERROVIÁRIAS**

Neste capítulo relata-se o desenvolvimento do protótipo de um Sistema de Realidade Virtual aplicada à Visualização de Informações onde é apresentado o projeto e a implementação de seus *scripts* e interfaces. Esta aplicação, destinada a fornecer aos administradores e/ou clientes da Brasil Ferrovias S.A. visões diferenciadas de informações. Utiliza-se de técnicas de Realidade Virtual para construção das visões e da interface *World Wide Web* para interação com o usuário..

A utilização de técnicas de Visualização de Informações juntamente com técnicas de Realidade Virtual permitiu a criação de um sistema que proporciona a construção de diferentes visões para cada usuário que acesse o sistema, trabalhando com informações *on-line* da base de dados da empresa, o que proporciona que o cliente tenha uma posição real de onde está a sua carga, e que os administradores da empresa consigam acompanhar todo o transporte e o trajeto das cargas. Estas visões podem ser montadas utilizando-se gráficos tridimensionais e dinâmicos que permitem que haja interação do usuário com estes gráficos.

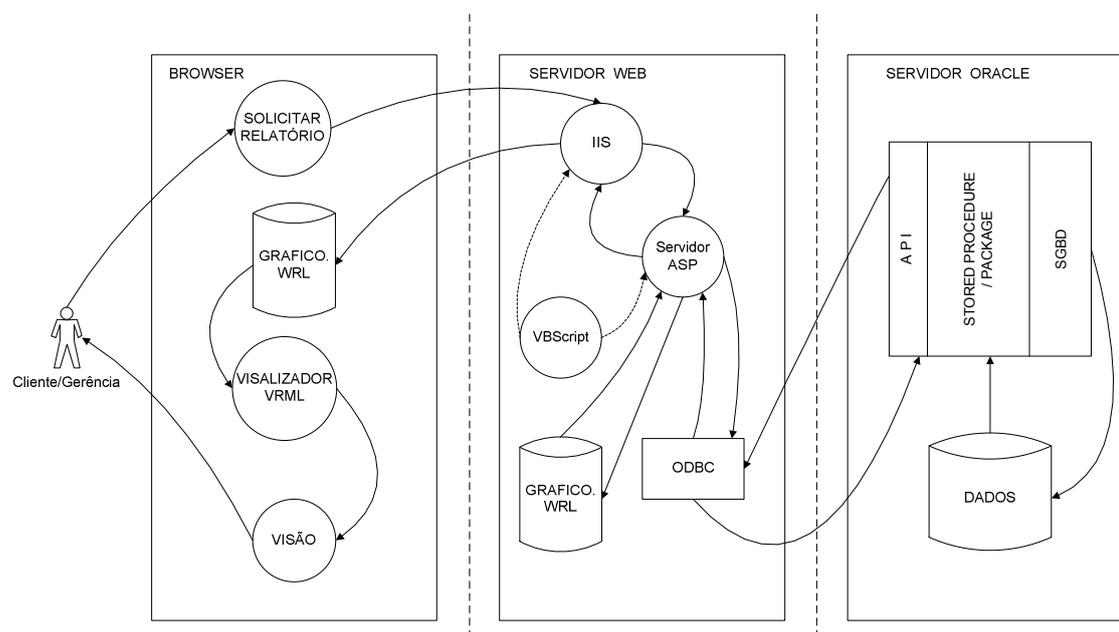
A construção deste sistema necessitou do conhecimento e do levantamento da situação computacional da empresa, de estudos das tecnologias que foram utilizadas na construção, assim como conceitos de Realidade Virtual em conjunto com conceitos de Visualização de Informações, conceitos estes discutidos neste trabalho.

Este protótipo foi desenvolvido sobre a plataforma *World Wide Web*, levando em consideração que os clientes que necessitam destas informações, não precisariam que a empresa enviasse as mesmas via e-mail para eles e sim que eles acessem o Web Site da empresa e consultem a posição de suas cargas, e também que se permita que o acesso seja feito de qualquer local, não sendo necessário que o administrador esteja na empresa para consultar.

No que tange ao desenvolvimento do protótipo, utilizar-se-á o item 3.1 para apresentar modelo do protótipo, mostrando como o mesmo foi desenvolvido. O item 3.2, trata da plataforma utilizada para o desenvolvimento do protótipo. O item 3.3, relata os resultados obtidos na construção do software.

### 3.1. MODELO DO PROTÓTIPO

Com o objetivo de se construir um software com interface amigável para o cliente e/ou administrador da Empresa e que este sistema proporcionasse uma grande interatividade do usuário e que os mesmos necessariamente não acessariam apenas de dentro da empresa, construiu-se uma aplicação com interface *World Wide Web*. Inicialmente apresentamos o modelo lógico ilustrado na Figura 3.1, colocando em destaque o uso de serviços integrados já disponíveis na empresa.



**FIGURA 3.1 - MODELO LÓGICO PARA DESENVOLVIMENTO DO SISTEMA**

A Figura 3.1 representa os três principais serviços a serem utilizados na implementação do trabalho. As fronteiras são representadas pelas linhas pontilhadas, permitindo assim que o desenvolvimento do projeto da aplicação

seja feito de forma modular, o que possibilita a implementação e o teste de cada uma das partes e serviços, facilitando a geração dos casos de testes e depuração.

A Figura 3.2 apresenta o diagrama de interação para uma consulta de cargas transportadas, detalhando os módulos apresentados na Figura 3.1. Esta consulta é utilizada pelo Departamento Comercial e demais clientes que tenham necessidade de informações sobre às cargas transportadas.

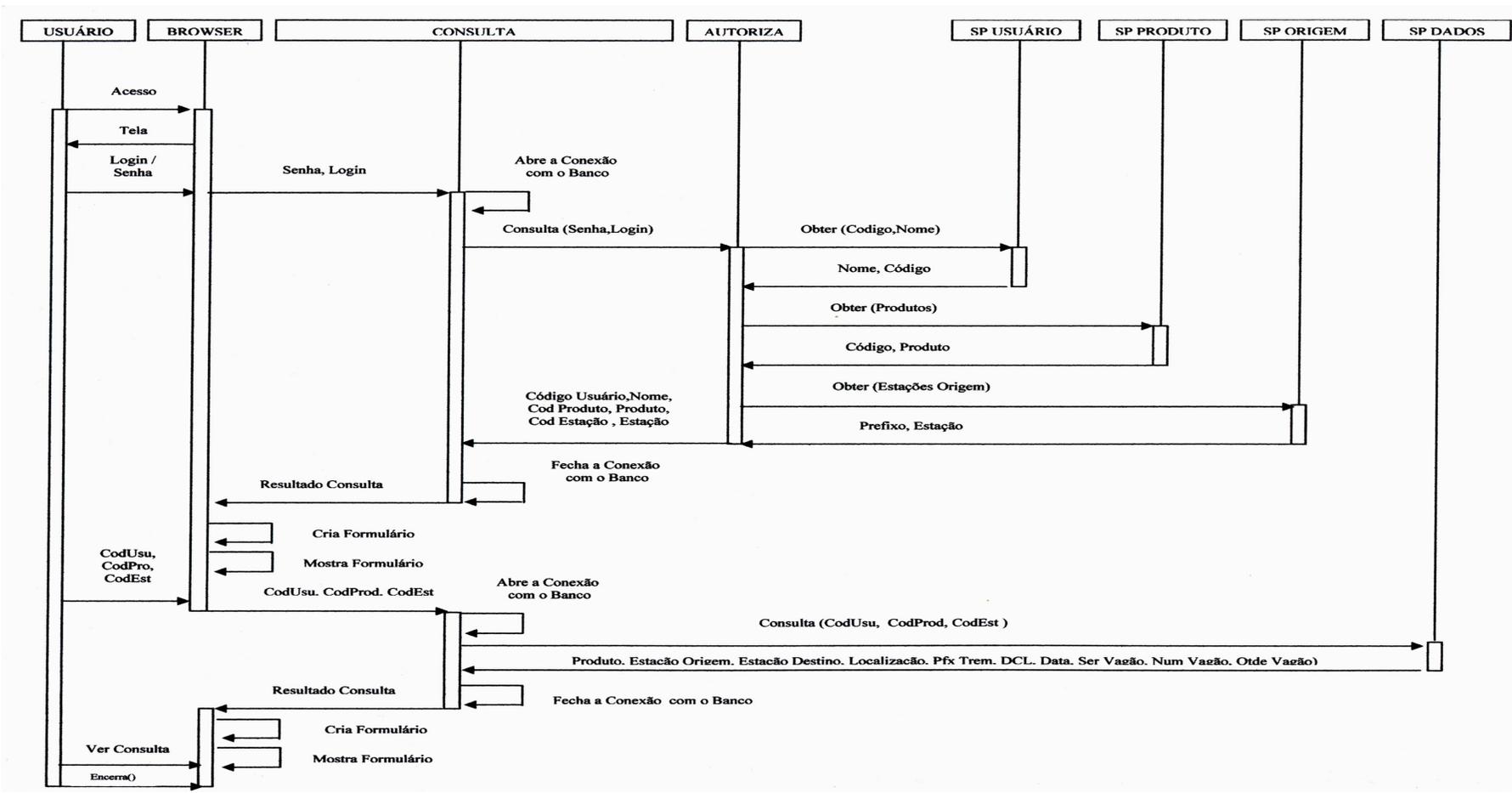


FIGURA 3.2 - DIAGRAMA DE INTERAÇÃO – CONSULTA DE LOCALIZAÇÃO DE CARGAS TRANSPORTADAS

O significado de “Usuário” na Figura 3.2 é mais amplo do que “Usuário” da Figura 3.1, no sentido de haver a possibilidade de clientes da empresa poderem ter acesso aos dados e saberem qual é a situação de suas respectivas cargas.

O módulo “Servidor WEB” apresentado na Figura 3.1 compreende as operações de autorização e de consulta ilustrados na Figura 3.2, evidenciando o fato do usuário não possuir meios de acesso direto ao banco de dados da empresa.

As consultas de cargas devem, portanto, diferenciar usuários externos dos internos. Os dados que usuários externos têm acesso deverão estar associados à Extranet da empresa e os dados que exigem maior sigilo (por exemplo: quantidade transportada) estarão associados à Intranet da empresa. Em ambos os casos, o modelo propõe que o usuário acessará o Browser e solicitará um relatório, bastando a diferenciação de servidor para caracterizar os serviços aos quais ele tem acesso.

Tendo em vista que o Banco de Dados utilizado pela empresa é o ORACLE, optou-se pela criação de um de Package (pacote) com as Stored Procedures para consulta ao banco.

Como o acesso aos dados ocorre via World Wide Web, utilizou-se a linguagem “ASP” para a produção das páginas sob demanda. Os scripts “ASP” permitem ao usuário selecionar o tipo de visão que deseja do banco de dados. Dependendo da seleção, o usuário obtém um formulário de dados, sob forma de página HTML, ou cena dinâmica.

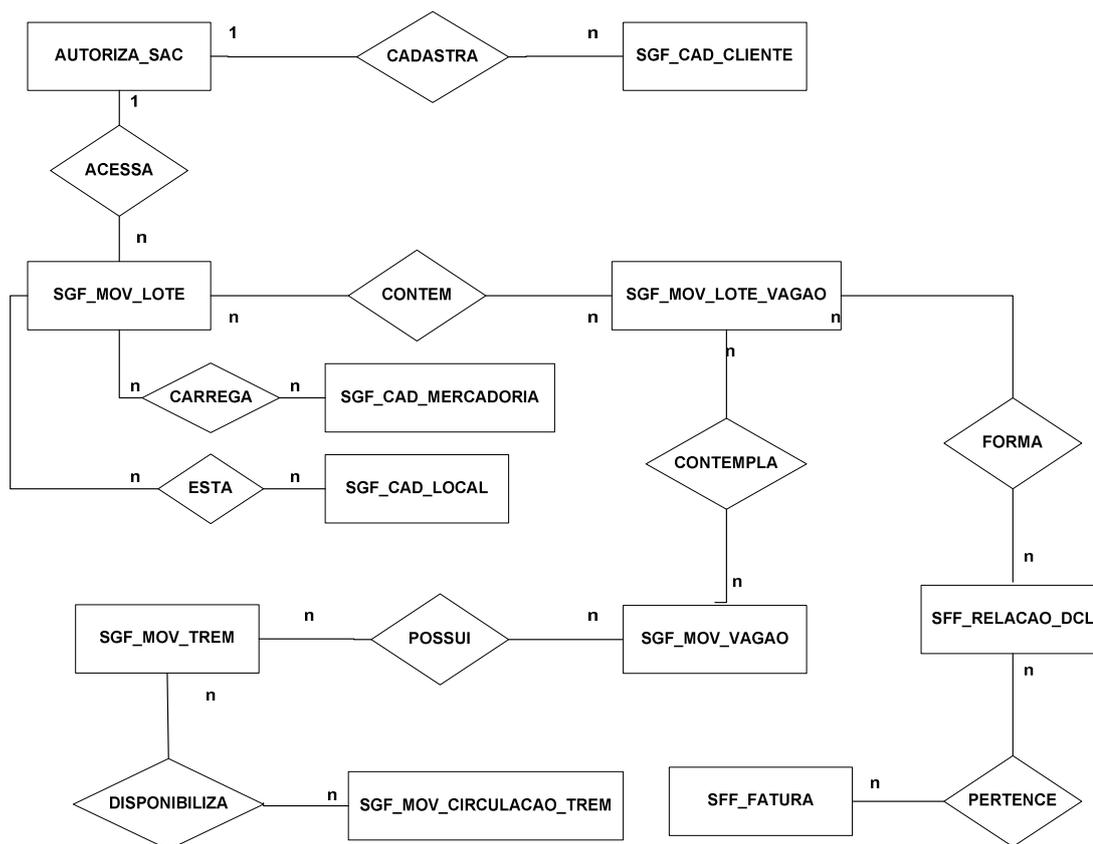
A cena dinâmica é caracterizada pelo transporte de uma descrição dos dados, exigindo que o software gere cenas baseadas em protótipos de gráficos modelados na linguagem Virtual Reality Modeling Language (ISO/IEC DIS 14772-1, 2003).

O Browser estabelece a interface com o usuário e transmite as solicitações ao servidor Active Server Page (ASP), que está hospedado no servidor Web da empresa. O módulo Servidor WEB é o responsável por traduzir as consultas

dos usuários em procedimentos de consulta ao banco de dados. Isto significa que este módulo é responsável por gerenciar as conexões/sessões de cada um dos usuários. Neste caso, pode haver diversas sessões de diversos usuários, causando problemas no servidor de banco de dados quando as sessões não são finalizadas a contento ou há falha no servidor WEB. Pode-se minimizar esse tipo de problema adotando-se solução análoga àquelas propostas em JERKE (2001). Isto faz com que o servidor ASP seja cliente do banco de dados, ao invés de permitir que o usuário crie a conexão com o banco. As consultas, portanto, criam sessões no banco de dados para cada um dos procedimentos armazenados a serem executados, caso o usuário tenha permissão de acesso aos dados por ele solicitados.

O Browser acessará o Servidor Web, através de páginas ASP enviando as requisições do usuário. O Servidor ASP acessará através da interface ODBC o servidor de Banco de Dados Oracle fazendo consultas por intermédio de Stored Procedures e/ou Package. Com o resultado será gerada uma página ASP e um Arquivo no padrão VRML que serão enviados ao Browser do usuário, que por sua vez executará um plugin e mostrará o gráfico gerado para o usuário.

A figura 3.3 mostra o Diagrama Entidade-Relacionamento que modela os elementos da consulta de posição das cargas.



**FIGURA 3.3 - DIAGRAMA ENTIDADE RELACIONAMENTO – CONSULTA DE CARGAS**

O Diagrama apresentado na Figura 3.3 foi obtido a partir da análise das tabelas apresentadas no Anexo B. Cabe lembrar que a documentação apresentada no Anexo B foi gerada exclusivamente para este trabalho.

Com base nesse diagrama, foi elaborado o *script* em linguagem VBScript (ver Tabela 3.1.), o respectivo procedimento armazenado (ver Tabela 3.2), a conexão com o banco de dados (ver Tabela 3.3) e o formato de página padrão a ser enviado ao Browser do usuário.

As páginas ASP fazem conexão com o Banco de Dados através de uma interface ODBC. Então é necessário que se registre uma Fonte de Dados Microsoft para Oracle. O comando `Set DB = Server.CreateObject("ADODB.Connection")` cria um cliente para o gerenciador ODBC, permitindo a solicitação de abertura de uma conexão ODBC com o banco de dados (através do comando `DB.Open`). As linhas de comando `Set CMD = Server.CreateObject("ADODB.Command")` e `Set CMD.ActiveConnection = DB` são necessárias para criar o elo entre a consulta e a

conexão com o banco de dados. Assim, o objeto CMD permite definir parâmetros, transmitir ou receber dados do banco de dados, conforme ilustra a Tabela 3.1.

**TABELA 3.1 – CONEXÃO DE PÁGINA ASP COM BANCO DE DADOS ORACLE**

```
<%
Dim CMD,RECTSET
Set DB = Server.CreateObject("ADODB.Connection")
DB.ConnectionString = "dsn=<instância_do_banco>;
                    uid=<identificador_usuario>;
                    pwd=<password_do_usuario>;"

DB.Open
Set CMD = Server.CreateObject("ADODB.Command")
Set CMD.ActiveConnection = DB
%>
```

As instruções de consulta ao Banco de Dados, estarão armazenadas no banco de dados e organizadas em *Packages*. O *Package* armazena todas os Stored Procedures necessários para execução da consulta. Um exemplo de procedimento armazenado é ilustrado na Tabela 3.2, que será aplicado para acesso ao banco de dados da Oracle. Todas as Stored Procedures necessárias para as consultas encontram-se relatadas no Anexo D.

**TABELA 3.2 – STORED PROCEDURE NO BANCO DE DADOS ORACLE**

```
CREATE OR REPLACE PROCEDURE GETNOMCLIENTE (CODCLIENTE
SGF_CAD_CLIENTE,COD_CLIENTE%TYPE)
IS
Begin
  Select NOM_RAZAO_SOCIAL into NOMCLI from SGF_CAD_CLIENTE
  Where COD_CLIENTE = CODCLIENTE;
End;
```

As páginas ASP, armazenadas no “Servidor WEB” (Figura 3.1), contêm scripts, escritas na linguagem “VBScript”, que fazem acesso ao banco de dados e solicitam a execução dos Stored Procedure, conforme ilustra a Tabela 3.3.

**TABELA 3.3 – CHAMADA DE STORED PROCEDURE DO BANCO DE DADOS ORACLE ATRAVÉS DE ASP**

```

CMD.CommandText = "{CALL GETDADOSA.GETPEDADOSA(?,?,?,{RESULTSET
100000,O_NOMLOC,O_PFXTREM,O_CODSIT,O_CODSER,O_NUMVAG,O_NUMDCL,O
_SERDCL,O_PFESTO,O_PFESTD,O_DSCMER,O_DATCAR,O_PFXLOC,O_SITREM,O
_DATFAT,O_PESTU}}}"
CMD.CommandType = adCmdText
CMD.Parameters.Append CMD3.CreateParameter("CLIENTE",adchar,adParamInput,4)
CMD.Parameters("Cliente") = fcliente
CMD.Parameters.Append CMD3.CreateParameter("PRODUTO",adchar,adParamInput,3)
CMD.Parameters("Produto") = cproduto
CMD.Parameters.Append CMD3.CreateParameter("ORIGEM",adchar,adParamInput,3)
CMD.Parameters("Origem") = corigem
Set RECTSET = CMD.Execute

```

Os dados retornados dos Stored Procedure são utilizados nos *scripts* do “Servidor WEB” para a geração de um arquivo que contém a cena dinâmica e o código HTML que faz com que esta cena seja transmitida ao Browser do usuário. Salienta-se que este arquivo possuirá um esqueleto, chamado “gráfico.wrl”, onde apenas serão acrescentados os dados retornados pelos Stored Procedure.

A Tabela 3.4 ilustra um protótipo em ASP que faz a criação de um arquivo com a extensão WRL. O código completo se encontra no Anexo E.

**TABELA 3.4 – EXEMPLO DE CÓDIGO ASP CRIANDO UM ARQUIVO VRML (TRECHO DO PROGRAMA)**

```

<html>
<head>
<title>Brasil Ferrovias</title>
</head>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<%
DIM cArquivo
DIM cCh
cCH = CHR(34)

cArquivo = "Grafico.wrl"

Set ObjFSO = CreateObject("Scripting.FileSystemObject")
ObjFSO.CreateTextFile(cArquivo)
Set objFile = ObjFSO.GetFile(cArquivo)
Set objStream = ObjFile.OpenAsTextStream(2,0)

objStream.WriteLine "#VRML V2.0 utf8"

objStream.WriteLine "Background {"
objStream.WriteLine "  skyColor 0.5 0.5 0.5"
objStream.WriteLine "}"

objStream.WriteLine "DEF TextSegunda Transform {"
.....
objStream.WriteLine "}"

ObjStream.close
Set ObjStream = nothing
%>
</body>
</html>

```

Como mostra a tabela 3.4, o programa em ASP gerou um arquivo chamado GRAFICO.WRL, que nada mais é do que uma visão gerada em VRML, arquivo este que contém a visão solicitada pelo usuário. A Tabela 3.5 ilustra como um programa em ASP faz para mostrar a visão criada para o usuário.

**TABELA 3.5 – EXEMPLO DE CÓDIGO ASP EXECUTANDO UM ARQUIVO VRML**

```

<html>
<head>
<title>Brasil Ferrovias</title>
</head>
<td width="500" valign="top" height="800"><embed src=""Grafico.wrl"> </td>
</body>
</html>

```

Como mostrado na Tabela 3.5, para que o Browser possa executar o arquivo com extensão WRL e mostrar o gráfico gerado, foi necessário a instalação de um plugin. No desenvolvimento desta aplicação foi utilizado o plugin Cortona (CORTONA, 2003). Este plugin permite que o usuário interaja com o gráfico.

### **3.2. PLATAFORMA DE DESENVOLVIMENTO**

O desenvolvimento deste protótipo ocorreu de maneira gradativa, levando em consideração, como dito anteriormente o parque computacional e as tecnologias existentes na empresa, como também observou-se a possibilidade de utilizar software livres e de baixo custo. Foi levado em consideração a popularização da Internet e então concluído como o título deste trabalho o desenvolvimento de um sistema de informação para Internet.

Quanto ao que se refere ao hardware, foi utilizado o Servidor Web da empresa que possui instalado o Sistema Operacional Windows 2000 Server, o servidor WEB IIS e que que esteja configurado o drive ODBC com acesso ao Servidor de Banco de Dados Oracle.

A aplicação foi desenvolvida utilizando a linguagem de programação Active Server Pages (ASP) e a linguagem de programação VRML, utilizando-se banco de dados oracle.

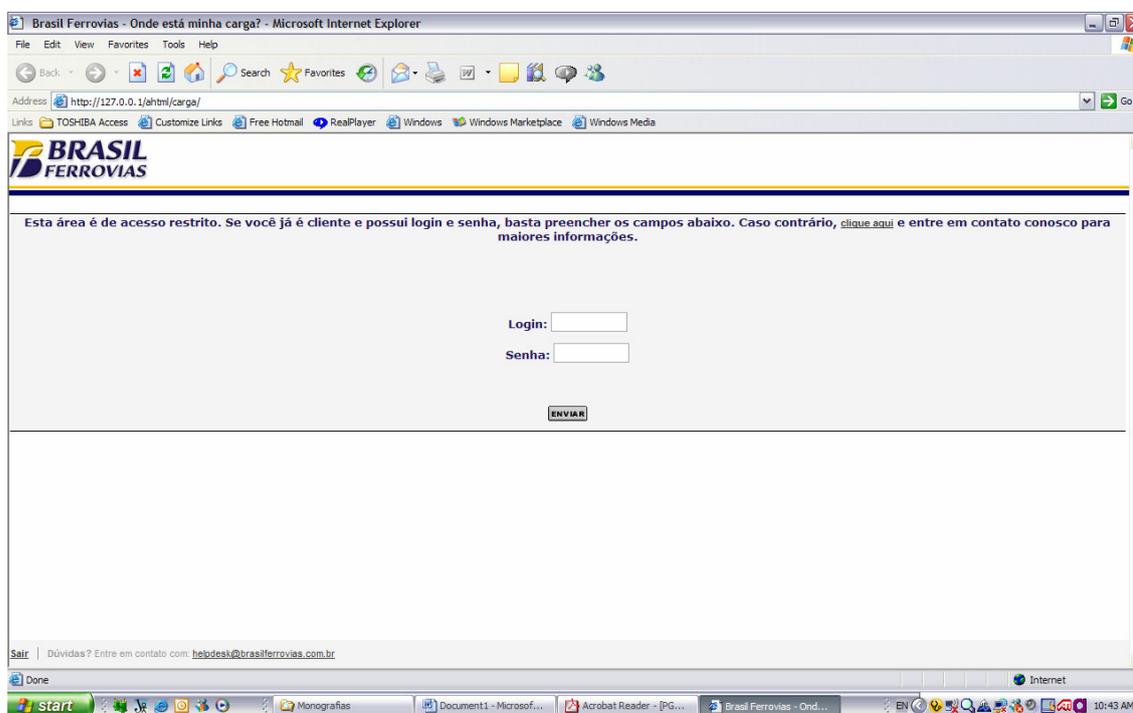
Na construção dos gráficos foi utilizado o editor VRMLPAD, editor próprio para linguagem VRML.

Na visualização dos gráficos gerados foi utilizado software como plug-in CORTONA.

### 3.3. RESULTADOS OBTIDOS

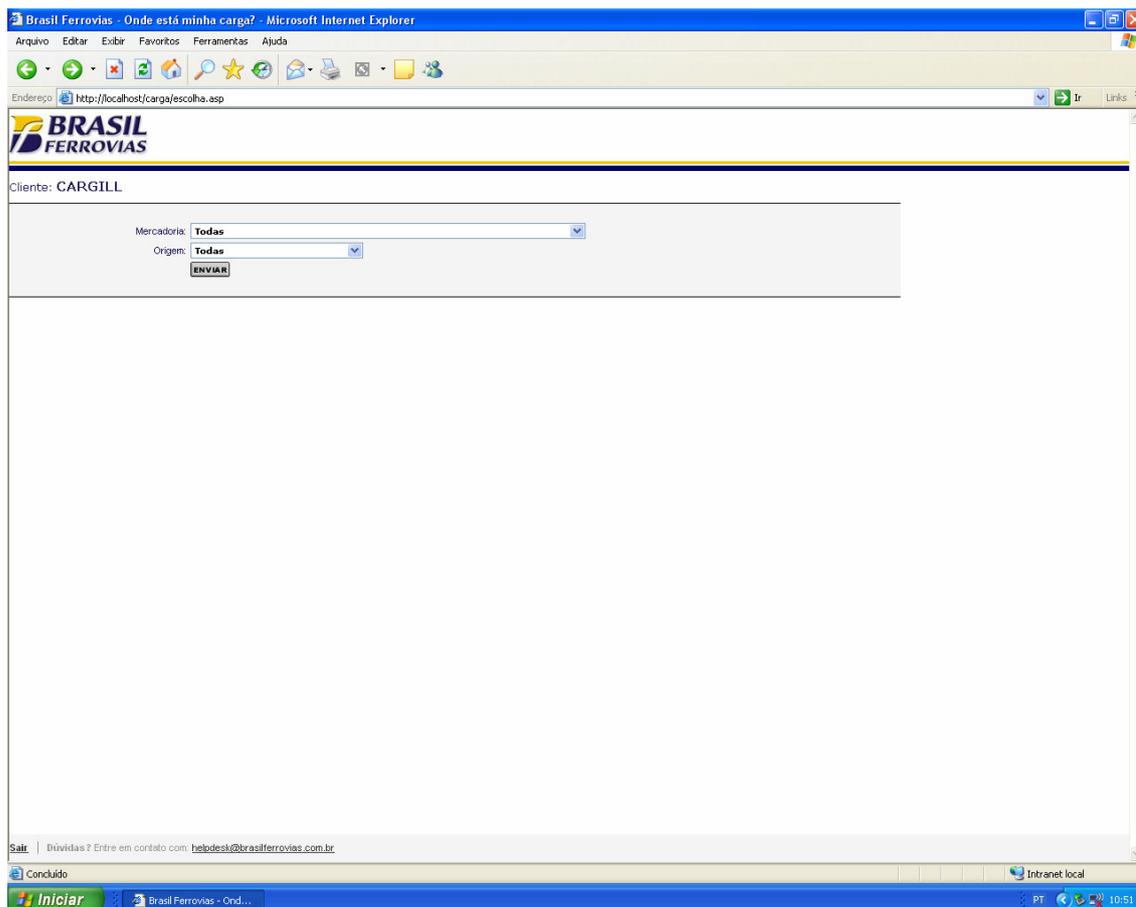
De maneira geral, os resultados obtidos foram satisfatórios, pois uma consulta que o cliente dependia da Brasil Ferrovias para ter a resposta e que sempre vinha com uma defazagem de 24 horas, passou estar disponível na Internet para que o próprio usuário consulte.

A Figura 3.4 ilustra a interface inicial em que o usuário, acessa a aplicação pelo Browse e se identificam através de um Login e uma Senha para ter prosseguir a consulta, a aplicação valida estas informações e disponibiliza para o usuário uma tela como ilustra a Figura 3.5.



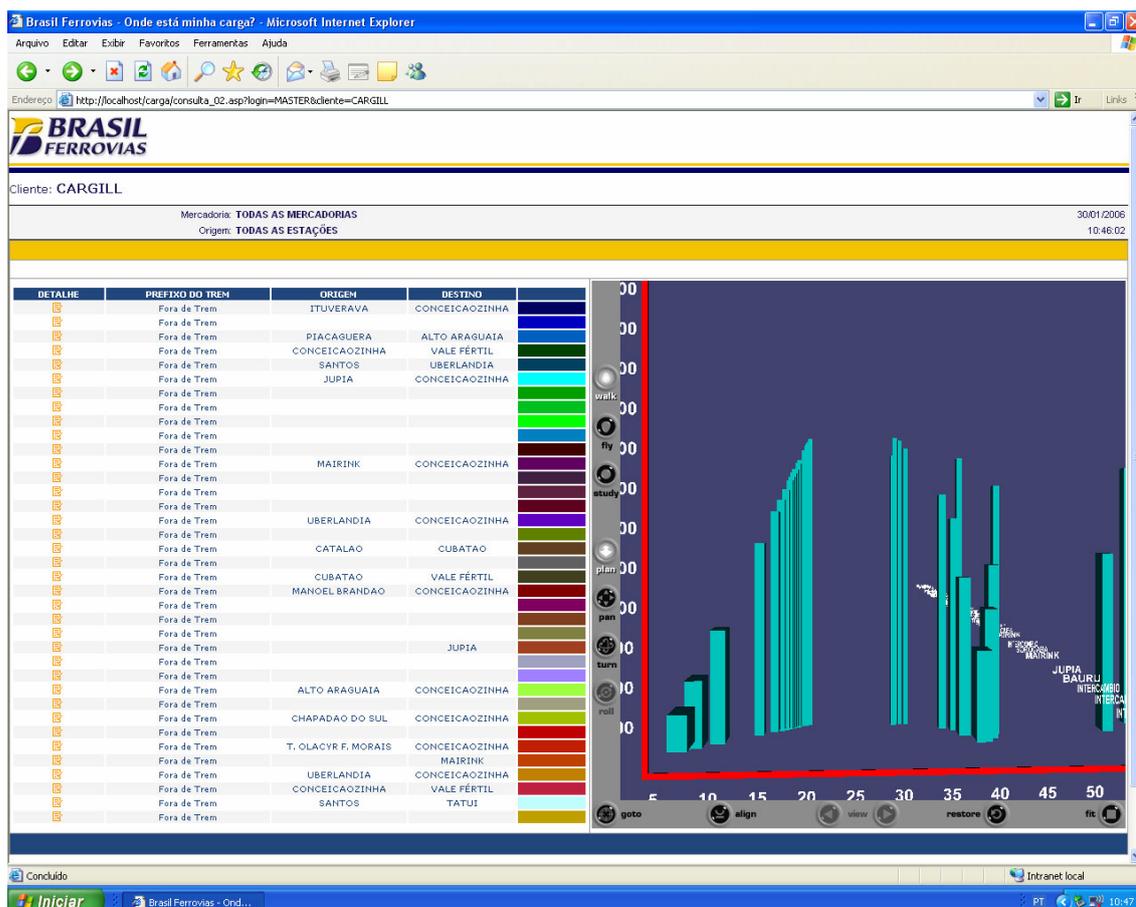
**FIGURA 3.4 – TELA INICIAL DA APLICAÇÃO**

A Figura 3.5 permite que o usuário escolha a Estação de Origem e o Produto transportado, dentre os já pré-definidos que o cliente contratou para transporte, para consulta, podendo o mesmo selecionar um ou todos, e o sistema mostrará uma ou todas as composições de carga pertencente a este usuário, dependendo de sua necessidade de consulta, o que resulta em várias visões de consulta.



**FIGURA 3.5 – TELA DE ESCOLHA DE MERCADORIA E ESTAÇÃO**

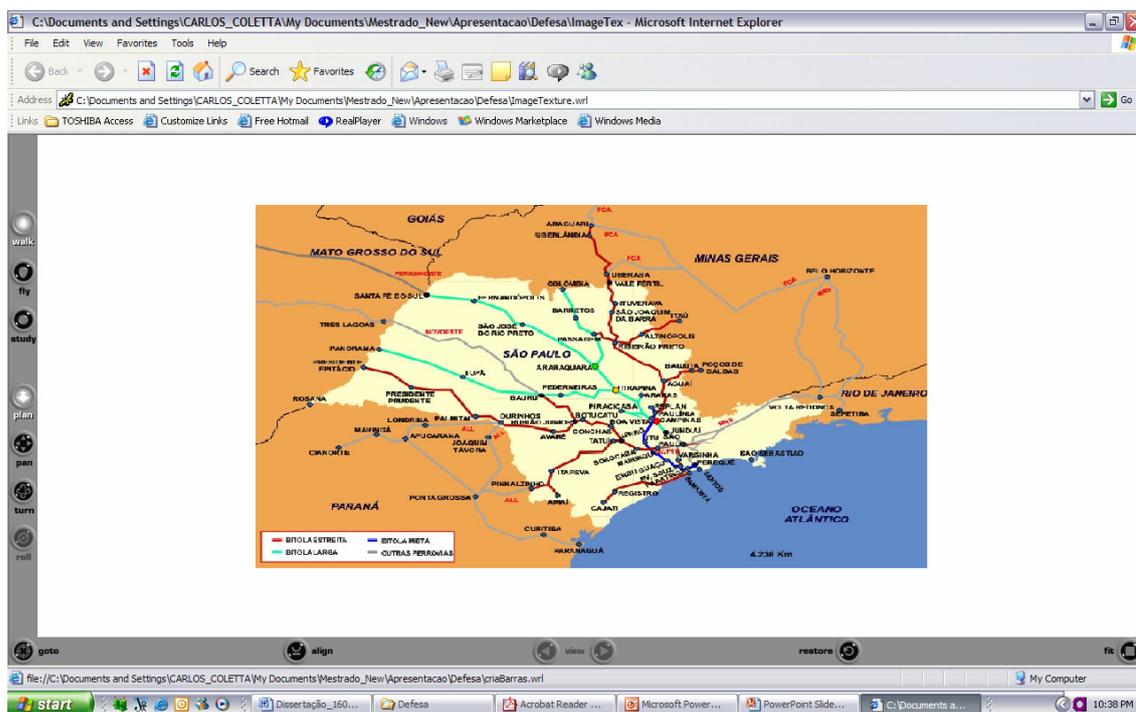
A Figura 3.6 ilustra uma visão de informações criada de acordo com os dados solicitados pelo usuário, mostrando o gráfico gerado através da linguagem VRML, e executado pelo plugin CORTONA. Nota-se que o usuário pode, se for preciso, movimentar o gráfico, buscando o melhor ponto para sua visualização e tomada de decisão. Esta opção de estudo do gráfico por parte do usuário só foi possível utilizando técnicas de Realidade Virtual. O gráfico no seu eixo X representa a quantidade de vagões na composição do trem, no eixo Y representa a quantidade (Toneladas) na composição do trem e o eixo Z corresponde a estação onde está o trem.



**FIGURA 3.6 – VISUALIZAÇÃO DE GRÁFICO EM VRML VIA PLUGIN CORTONA**

A Figura 3.7 ilustra uma outra visão de informações que o usuário pode ter, a qual é criada a partir de um mapa do estado de São Paulo, onde se pode verificar através do mesmo as composições de trem, a estação origem, estação destino e a localização atual do Trem. Tudo isto é gerado de acordo com os dados solicitados pelo usuário. Nota-se que o usuário pode, se for preciso, clicar na localização atual do trem e será mostrado um gráfico de barras mostrando a composição do trem, como por exemplo, quantidade transportada, número de vagões, entre outras.

O Código fonte do script em VRML do gráfico da figura 3.6 está descrito no Anexo A.



**FIGURA 3.7 – VISUALIZAÇÃO ATRAVÉ DE MAPA EM VRML VIA PLUGIN CORTONA**

No gráfico ilustrado na figura 3.7 as bolinhas vermelhas correspondem a estação de origem do trem, a verde a estação destino e a verde a estação atual do trem.

#### 4. CONSIDERAÇÕES FINAIS

Este trabalho apresentou, como proposta de dissertação, o desenvolvimento de um protótipo de Sistema de Realidade Virtual aplicada à Visualização de Informações via World Wide Web, um modelo de sistema baseado em ferramentas de baixo custo e com Softwares Livre, para atender uma necessidade da empresa de transportes ferroviários Brasil Ferrovias S.A. em disponibilizar para seus clientes e administradores uma consulta sobre a posição das cargas transportadas.

Inicialmente, através de entrevistas com o departamento comercial, foram apresentados como eram feitas estas consultas e como estavam sendo enviadas para os clientes. Para o desenvolvimento da aplicação foi necessário o levantamento das informações relativas ao parque computacional da empresa, e dos legados de sistemas que a empresa utiliza e o banco de dados utilizado. Com este levantamento descobriu-se em qual(is) sistema(s) e quais tabelas continham as informações que foram utilizadas para geração das consultas.

Durante o desenvolvimento do sistema alguns obstáculos foram encontrados como por exemplo, o Drive ODBC que seria utilizado, levando em consideração que para utilização de Packages alguns drives não ODBC funcionaram.

O desenvolvimento do sistema com a utilização de técnicas de Realidade Virtual, utilizando a linguagem VRML, permitiu que o usuário interagisse com os gráficos gerados e buscasse a melhor forma para consultar as informações que necessitavam. Cada gráfico gerado pode-se dizer que é uma visão que o usuário pode ter das cargas transportadas.

Um das vantagens da VRML é possibilitar a visualização de ambientes virtuais por qualquer pessoa que tenha acesso à Internet, independentemente da plataforma que esteja usando (Windows 95/98/ME/NT/2000/XP, Linux, Unix

etc). A existência de características de multimídia, interação com as linguagens HTML, Java, JavaScript, VRMLScript e a base de dados em SQL, fazem com que a VRML se torne uma poderosa ferramenta para aplicação em muitos ramos da ciência e atividades acadêmicas e comerciais.

Freqüentemente tem se pesquisado com grande interesse formas e técnicas de visualização de informações. Este trabalho teve como objetivo o desenvolvimento de uma aplicação para Visualização de Informações utilizando técnicas de Realidade Virtual, para além de demonstrar que é possível utilizar softwares Livre para a construção da aplicação, também suprir uma necessidade da empresa Brasil Ferrovias, que tinha como objetivo o desenvolvimeto de uma aplicação para que seus clientes e administradores consultassem os transportes de carga.

Concluimos com este trabalho que muito ainda pode ser explorado em Visualização de Informações utilizando técnicas de Realidade Virtual, podendo também se utilizar de outras técnicas como Realidade Aumentada, Virtualidade Aumentada, entre outras, visto que que estas técnicas possuem um campo de pesquisa bastante vasto. Pode-se também utilizar outros softwares como o JAVA 3D para construção de visões de informações não necessitando assim de nenhum *plugin* para visualização da mesma, dentre outras linguagens utilizadas para se desenvolver aplicações em Realidade Virtual

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

ALBERTIN, ALBERTO LUIZ. **Comércio Eletrônico**. São Paulo: Atlas, 3.ed, 2001. 276p.

ARNETT, MATTHEW FLINT; DULANEY, EMMETT; HARPER, ERIC. etc al. **Desvendando TCP/IP**. Trad. ARX Publicações. Rio de Janeiro: Atlas, 1997. 543p.

BRASIL FERROVIAS. **Documentação Servidor FNWEBSRVNT**. Campinas, 2002a, 3p. (Relatório Técnico)

BRASIL FERROVIAS. **Documentação Servidor FNPROXYLX**. Campinas, 2002b, 4p. (Relatório Técnico)

BRASIL FERROVIAS. **Manual SIGEFER**. Campinas, 2002c, 47p. (Relatório Técnico)

BRASIL FERROVIAS. **Relatório de Mapeamento de Servidores**. Campinas, 2002d, 20p. (Relatório Técnico)

BRUTZMAN, DON. **The Virtual Reality Modeling Language and Java**. Communications of ACM , v.41, p.57-64, Jul. 1998.

CAMANHO, ANTONIO LUIZ; ROCHA RICARDO S. **Plano de Tecnologia e Sistemas**. Campinas, 2001, 92p. (Relatório Técnico)

CARD, STUART K.; MACKINLAY, JOCK D.; SHNEIDERMAN, BEN. **Readings Information Visualization Using Vision to Think**. Morgan Kaufmann Pub. 1999. 686p.

COPELAND, DENNIS R.; CORBO, RAYMOND C.; FALKENTHAL, SUSAN A.; FISHER, JAMES L.; SANDLER, MARK N. **Which Development Tool is Right for you?**. IT Professional - IEEE, Vol 2(2), pp. 20-27, 2000.

CORTONA (2003). **Cortona Jet. Parallel Graphics, Inc.**  
<http://www.parallelgraphics.com/products/jet> (13 mai.)

- CGI (2003). **CGI: Common Gateway Interface**. <http://www.w3c.org/cgi> (13 mai.)
- DATE, C. J. **Introdução a Sistemas de Banco de Dados**. Trad. de Contexto Traduções. Rio de Janeiro: Campus, 4 ed., 1991. 674p.
- DINIZ, BÁRBARA SOARES PESSOA; SIQUEIRA, MOZART CAMPOS ARAÚJO FILHO. **Criação de Mundos Tridimensionais na Internet: Uma introdução operacional a VRML**. ETF'99 Escola de Tecnologia da Informação. Marília-SP, 1999.
- ELMASRI, RAMEZ; NAVATHE, SHAMKANT B. **Fundamentals of DataBase Systems**. New York: Addison-Wesley, 2 ed., 1994. 873p.
- ELMASRI, RAMEZ; NAVATHE, SHAMKANT B. **Fundamentals of DataBase Systems**. New York: Addison-Wesley, 3 ed., 2000. 980p.
- ENGESYS. **Automação de Controle de Trens**. Campinas, 2002, 31p. (Relatório Técnico)
- ERWIN (2005). **All Fusion Erwin Data Modeler**. <http://www3.ca.com/solutions/product.aspx?ID260>. (20 mai.)
- FERREIRA, AURÉLIO BUARQUE DE HOLANDA. **Novo Aurélio Século XXI: O Dicionário da Língua Portuguesa**. Rio de Janeiro: Nova Fronteira, 3 ed., 1999. 2128p.
- FERROBAN. **Resumo Sistemas**. Campinas, 2002a. 15p. (Relatório Técnico)
- FERROBAN (2002b). **Empresa**. [www.ferroban.com.br/empresa.htm](http://www.ferroban.com.br/empresa.htm) (27 mar.)
- FERRONORTE (2002). **Empresa**. [www.ferronorte.com.br/empresa/nossa.htm](http://www.ferronorte.com.br/empresa/nossa.htm) (27 mar.)
- GERSHON, NAHUM; EICK, STEPHEN G.; CARD STUART. **Information Visualization**. Interactions, v.5, Issue 3, p.09-15, Mar/Abr. 1998.

- GINIGE, ATHULA. **Web Engineering: Managing the Complexity of Web Systems Development**. Proceedings 14<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering, 2002
- GREER, TYSON. **Understanding Intranets**. Washington: Microsoft Express, 1998. 384p.
- HANSEN, STEVE. **Web Information Systems: The Changing Landscape of Management Models and Web Applications**. Proceedings 14<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering, 2002
- ISO/IEC DIS 14772-1. **VRML 97 Specification**.  
<http://www.web3d.org/technical/specifications/vrml97/index.html>, 2003.
- JERKE, NOEL. **Desenvolvendo E-Commerce com Visual Basic, ASP e SQL Server**. Trad. Maria da Penha. São Paulo: Makron Books, 1 ed., 2001. 524p.
- KIRNER, CLAUDIO. **Sistemas de Realidade Virtual**. 2002a.  
[http://www.realidadevirtual.com.br/publicacoes/tutorial\\_rv/tutrv.html](http://www.realidadevirtual.com.br/publicacoes/tutorial_rv/tutrv.html).
- KIRNER, CLAUDIO. **Realidade Virtual: Dispositivos e Aplicações**. 2002b.  
[http://www.realidadevirtual.com.br/publicacoes/apostila\\_rv\\_disp\\_aplicacoes/apostila\\_rv.html](http://www.realidadevirtual.com.br/publicacoes/apostila_rv_disp_aplicacoes/apostila_rv.html).
- KLENSIN, J., **RFC 2821, Simple Mail Transfer Protocol**. 2001.  
<http://www.ietf.org/rfc/rfc2821.txt?number=2821>.
- KS SISTEMAS. **Sistema de Gerenciamento de Equipagem**. Campinas, 2001, 57p. (Relatório Técnico)
- LAI, VICENTE S. **Intraorganizational Communication with Intranets**. Communications of the ACM, Vol 44(7), pp. 95-100, 2001.
- MASS, YOSI; HERZBERGER, AMIR. **VRCommerce – Eletronic Commerce in Virtual Reality**. Proceedings of the First ACM Conference on Eletronic Commerce, pp. 103-109, 1999.

MENDONÇA NETTO, MANOEL GOMES DE; ALMEIDA, MÁRCIO OLIVEIRA. **Uso de Interfaces Abundantes em Informação para Exploração Visual de Dados**. IV WorkShop sobre Fatores Humanos em Sistemas Computacionais, pp. 256-268, 2001.

MICROSOFT LIBRARY. **Microsoft Transaction Server and Internet Information Server: Technology for Web**", 2002.  
[http://msdn.microsoft.com/library/en-us/dnmts/html/msdn\\_mtsiis.asp](http://msdn.microsoft.com/library/en-us/dnmts/html/msdn_mtsiis.asp)

MICROSOFT LIBRARY. **What is ODBC ?**, 2003a.  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/hm/odch01pr\\_2.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/hm/odch01pr_2.asp)

MICROSOFT LIBRARY. **ODBC Architecture**, 2003b.  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/hm/odbcodbc\\_api\\_reference.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/hm/odbcodbc_api_reference.asp)

MICROSOFT LIBRARY. **ACTIVE SERVER PAGE (ASP)**, 2003c.  
<http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/iisref/aspguide.htm>

MINASI, MARK; ANDERSON CHRISTA; SMITH, BRIAN M.; TOOMBS, DOUG. **Dominando o Windows 2000 Server "A Bíblia"**. São Paulo: Makron Books, 2000. 1275p.

MORRISON, MIKE; MORRISON, JOLINE; KEYS, ANTHONY. **Integrating Web Sites and DataBases**. Communications of the ACM, Vol 45(9), pp. 81-86, 2002.

MURPHY JR., T.; RIETH, P.; STEVENS, J. **RFC 2877, Telnet Enhancements**. 2000.  
<http://www.ietf.org/rfc/rfc2877.txt?number=2877>

ÖZSU, M. TAMER; PATRICK VALDURIEZ. **Princípios de Sistemas de Banco de Dados Distribuídos**. Trad. de Vandenberg D. de Souza. Rio de Janeiro: Campus, 2 ed., 2001. 711p.

PORTOFER (2002). **Empresa**. [www.portofer.com.br](http://www.portofer.com.br) (27 mar.)

POSTEL, J.; REYNOLDS, J. **RFC 0959, File Transfer Protocol (FTP)**. 1985.

<http://www.ietf.org/rfc/rfc0959.txt?number=0959>

POWELL, MATT; BRAGINSKI, LEONID. **Running Internet Information Server**. Washington: Microsoft Express, 1998. 740p.

RAMALHO, JOSE ANTONIO. **Oracle 8i**. São Paulo: Berkeley, 1999. 494p.

SETZER, WALDEMAR W. **Coleção "Ensaio Transversais"**. São Paulo: Editora Escirturas, Vol. 10, 2001, 288p.

SPENCE, ROBERT. **Information Visualization**. England: ACM Press, 2001. 206p.

TANENBAUM, ANDREW S. **Redes de Computadores**. Trad. de Vandenberg D. de Souza. Rio de Janeiro: Campus, 2 ed., 1994. 786p.

URMAN, SCOTT. **Oracle 9i: Programação PL/SQL**. Trad. de Edson Furmankiewicz, Docware Traduções Técnicas. Rio de Janeiro: Campus, 2002. 528p.

VALÉRIO NETTO, ANTONIO; MACHADO, LILIANE DOS SANTOS; OLIVEIRA, MARIA CRISTINA F. DE. **Realidade Virtual: Fundamentos e Aplicações**. Florianópolis-SC: VisualBooks, 2002. 528p.

WARE, COLIN. **Information Visualization Perception for Design**. United States of America: Academic Press, 2000. 438p.

WASH, ARON E.; SÉVENIER, MIKAEL BOURGES. **Color Web3d**. USA: Prentice Hall Ptr, 2001. 1088p.

WEB3D CONSORTIUM (2003). **VRML 1.0 Specification**.

<http://www.web3d.org/technicalinfo/specifications/VRML1.0/index.html>

(15 mai.)

WITH, NIKLAUS. **Algoritmos e Estruturas de Dados**. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 1999. 256p.

## ANEXO A – EXEMPLO DE SCRIPT EM VRML PARA GERAÇÃO DE GRÁFICOS

```
#VRML V2.0 utf8
Background {
    skyColor 0.258824 0.258824 0.435294
}
DEF Planilha Group {
    children [
        DEF EixoY Transform{
            translation -3.0 0.4 0
            children [
                Shape {
                    appearance DEF CorEixo Appearance {
                        material Material {
                            diffuseColor 1 0 0
                        }
                    }
                    geometry Box {
                        size 0.1 8 0.01
                    }
                }
            ]
        }
        DEF EixoX Transform{
            translation 1.0 -3.55 0
            children [
                Shape {
                    appearance USE CorEixo
                    geometry Box {
                        size 8 0.1 0.01
                    }
                }
            ]
        }
        DEF Texty300 Transform {
            translation -3.6 -2.9 0
            children [
                Shape {
                    geometry Text {
                        string ["300"]
                        maxExtent 1
                        fontStyle FontStyle {
                            size 0.3
                        }
                    }
                }
            ]
        }
    ]
}
```

```

family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Texty600 Transform {
  translation -3.6 -2.3 0
  children [
  Shape {
  geometry Text {
  string ["600"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  }
  ]
  }
DEF Texty900 Transform {
  translation -3.6 -1.7 0
  children [
  Shape {
  geometry Text {
  string ["900"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE

```

```

language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Texty1200 Transform {
  translation -3.7 -1.1 0
  children [
  Shape {
  geometry Text {
  string ["1200"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  }
  ]
}
DEF Texty1500 Transform {
  translation -3.7 -0.5 0
  children [
  Shape {
  geometry Text {
  string ["1500"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  }
  ]
}

```

```

}
]
}
DEF Texty1800 Transform {
  translation -3.7 0.1 0
  children [
  Shape {
  geometry Text {
  string ["1800"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  ]
}
DEF Texty2100 Transform {
  translation -3.7 0.7 0
  children [
  Shape {
  geometry Text {
  string ["2100"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  ]
}
DEF Texty2400 Transform {
  translation -3.7 1.3 0

```

```

children [
Shape {
geometry Text {
string ["2400"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Texty2700 Transform {
translation -3.7 1.9 0
children [
Shape {
geometry Text {
string ["2700"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Texty3000 Transform {
translation -3.7 2.5 0
children [
Shape {
geometry Text {
string ["3000"]
maxExtent 1

```

```

fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Texty3300 Transform {
translation -3.7 3.1 0
children [
Shape {
geometry Text {
string ["3300"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Texty3600 Transform {
translation -3.7 3.7 0
children [
Shape {
geometry Text {
string ["3600"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE

```

```

leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Texty3900 Transform {
  translation -3.7 4.3 0
  children [
  Shape {
  geometry Text {
  string ["3900"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  }
  ]
}
DEF Texty4200 Transform {
  translation -3.7 4.9 0
  children [
  Shape {
  geometry Text {
  string ["4200"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1

```

```

}
}
}
]
}
DEF Texty4500 Transform {
  translation -3.7 5.5 0
  children [
  Shape {
  geometry Text {
  string ["4500"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  }
  ]
}
DEF Texty4800 Transform {
  translation -3.7 6.1 0
  children [
  Shape {
  geometry Text {
  string ["4800"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  }
  ]
}

```

```

DEF Texty5100 Transform {
  translation -3.7 6.7 0
  children [
  Shape {
  geometry Text {
  string ["5100"]
  maxExtent 1
  fontStyle FontStyle {
  size 0.3
  family "SANS"
  style "BOLD"
  horizontal TRUE
  leftToRight TRUE
  topToBottom TRUE
  language "pr"
  justify "FIRST"
  spacing 0.1
  }
  }
  ]
}

DEF Textx5 Transform {
translation -2 -2.70 3.0
children [
Shape {
geometry Text {
string ["5"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
]
}

DEF Textx10 Transform {
translation -1.5 -2.70 3.0
children [
Shape {
geometry Text {

```

```

string ["10"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
}
}
DEF Textx15 Transform {
translation -1 -2.70 3.0
children [
Shape {
geometry Text {
string ["15"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
}
}
DEF Textx20 Transform {
translation -0.5 -2.70 3.0
children [
Shape {
geometry Text {
string ["20"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"

```

```

style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx25 Transform {
translation 0 -2.70 3.0
children [
Shape {
geometry Text {
string ["25"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx30 Transform {
translation 0.5 -2.70 3.0
children [
Shape {
geometry Text {
string ["30"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"

```

```

justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx35 Transform {
translation 1 -2.70 3.0
children [
Shape {
geometry Text {
string ["35"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx40 Transform {
translation 1.5 -2.70 3.0
children [
Shape {
geometry Text {
string ["40"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}

```

```

]
}
DEF Textx45 Transform {
translation 2 -2.70 3.0
children [
Shape {
geometry Text {
string ["45"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx50 Transform {
translation 2.5 -2.70 3.0
children [
Shape {
geometry Text {
string ["50"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx55 Transform {
translation 3 -2.70 3.0
children [

```

```

Shape {
  geometry Text {
    string ["55"]
    maxExtent 1
    fontStyle FontStyle {
      size 0.2
      family "SANS"
      style "BOLD"
      horizontal TRUE
      leftToRight TRUE
      topToBottom TRUE
      language "pr"
      justify "FIRST"
      spacing 0.1
    }
  }
}
]
}
DEF Textx60 Transform {
  translation 3.5 -2.70 3.0
  children [
    Shape {
      geometry Text {
        string ["60"]
        maxExtent 1
        fontStyle FontStyle {
          size 0.2
          family "SANS"
          style "BOLD"
          horizontal TRUE
          leftToRight TRUE
          topToBottom TRUE
          language "pr"
          justify "FIRST"
          spacing 0.1
        }
      }
    }
  ]
}
DEF Textx65 Transform {
  translation 4 -2.70 3.0
  children [
    Shape {
      geometry Text {
        string ["65"]
        maxExtent 1
        fontStyle FontStyle {

```

```

size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx70 Transform {
translation 4.5 -2.70 3.0
children [
Shape {
geometry Text {
string ["70"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx75 Transform {
translation 5 -2.70 3.0
children [
Shape {
geometry Text {
string ["75"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE

```

```

topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx80 Transform {
translation 5.5 -2.70 3.0
children [
Shape {
geometry Text {
string ["80"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx85 Transform {
translation 6 -2.70 3.0
children [
Shape {
geometry Text {
string ["85"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}

```

```

}
}
]
}
DEF Textx90 Transform {
translation 6.5 -2.70 3.0
children [
Shape {
geometry Text {
string ["90"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
DEF Textx95 Transform {
translation 7 -2.70 3.0
children [
Shape {
geometry Text {
string ["95"]
maxExtent 1
fontStyle FontStyle {
size 0.2
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "FIRST"
spacing 0.1
}
}
}
]
}
]

```

```

}
DEF Textz1 Transform {
translation 5.8 -3.5 -0.5
children [
Shape {
geometry Text {
string ["PEREQUE"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -0.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br1 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 0.2912 0.5
}
}
]
}
]
}
DEF Textz2 Transform {

```

```

translation 5.8 -3.5 -1.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -1.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br2 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 0.5824 0.5
}
}
]
}
]
}
DEF Textz3 Transform {
translation 5.8 -3.5 -2.5
children [

```

```

Shape {
  geometry Text {
    string ["INTERCAMBIO"]
    maxExtent 1
    fontStyle FontStyle {
      size 0.3
      family "SANS"
      style "BOLD"
      horizontal TRUE
      leftToRight TRUE
      topToBottom TRUE
      language "pr"
      justify "MIDDLE"
      spacing 0.8
    }
  }
}
]
}
Transform {
  translation 2.65 -3.0 -2.5
  children [
    DEF qtde1 Transform{
      translation 0 0.01 0
      scale 1 2 1
      children [
        Shape {
          appearance DEF Br3 Appearance {
            material Material {
              diffuseColor 0 .35 .25
              specularColor 0 .31 .36
              emissiveColor 0 .13 .15
              ambientIntensity .0533
              shininess .32
            }
          }
          geometry Box {
            size 0.3 0.8736 0.5
          }
        }
      ]
    }
  ]
}
DEF Textz4 Transform {
  translation 5.8 -3.5 -3.5
  children [
    Shape {
      geometry Text {

```

```

string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
}
}
Transform {
translation -2.6 -3.0 -3.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br4 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
}
geometry Box {
size 0.3 1.1648 0.5
}
}
}
}
}
}
DEF Textz5 Transform {
translation 5.8 -3.5 -4.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1

```

```

fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 3.25 -3.0 -4.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br5 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 1.456 0.5
}
}
]
}
]
}
DEF Textz6 Transform {
translation 5.8 -3.5 -5.5
children [
Shape {
geometry Text {
string ["BAURU"]
maxExtent 1
fontStyle FontStyle {
size 0.3

```

```

family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 9.85 -3.0 -5.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br6 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 1.7472 0.5
}
}
]
}
]
}
DEF Textz7 Transform {
translation 5.8 -3.5 -6.5
children [
Shape {
geometry Text {
string ["JUPIA"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"

```

```

horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 3.05 -3.0 -6.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br7 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 2.0384 0.5
}
}
]
}
]
}
DEF Textz8 Transform {
translation 5.8 -3.5 -7.5
children [
Shape {
geometry Text {
string ["PORTOFER CONCEIÇÃOZINHA"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE

```

```

topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 4.05 -3.0 -7.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br8 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 2.3296 0.5
}
}
]
}
}
DEF Textz9 Transform {
translation 5.8 -3.5 -8.5
children [
Shape {
geometry Text {
string ["MAIRINK"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"

```

```

justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 7.65 -3.0 -8.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br9 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 2.6208 0.5
}
}
]
}
}
DEF Textz10 Transform {
translation 5.8 -3.5 -9.5
children [
Shape {
geometry Text {
string ["SOROCABA"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8

```

```

}
}
}
]
}
Transform {
translation -2.6 -3.0 -9.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br10 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 2.912 0.5
}
}
]
}
]
}
DEF Textz11 Transform {
translation 5.8 -3.5 -10.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}

```

```

}
]
}
Transform {
translation 26.05 -3.0 -10.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br11 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 3.2032 0.5
}
}
]
}
]
}
DEF Textz12 Transform {
translation 5.8 -3.5 -11.5
children [
Shape {
geometry Text {
string ["PORTOFER CONCEIÇÃOZINHA"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
]

```

```

}
Transform {
translation 14.85 -3.0 -11.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br12 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 3.4944 0.5
}
}
]
}
]
}
DEF Textz13 Transform {
translation 5.8 -3.5 -12.5
children [
Shape {
geometry Text {
string ["MAIRINK"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {

```

```

translation 3.85 -3.0 -12.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br13 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 3.7856 0.5
}
}
]
}
}
DEF Textz14 Transform {
translation 5.8 -3.5 -13.5
children [
Shape {
geometry Text {
string ["SOROCABA"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -13.5
children [

```

```

DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br14 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 4.0768 0.5
}
}
]
}
DEF Textz15 Transform {
translation 5.8 -3.5 -14.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 13.05 -3.0 -14.5
children [
DEF qtde1 Transform{
translation 0 0.01 0

```

```

scale 1 2 1
children [
Shape {
appearance DEF Br15 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 4.368 0.5
}
}
]
}
}
DEF Textz16 Transform {
translation 5.8 -3.5 -15.5
children [
Shape {
geometry Text {
string ["PEREQUE"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -15.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [

```

```

Shape {
appearance DEF Br16 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 4.6592 0.5
}
}
]
}
}
DEF Textz17 Transform {
translation 5.8 -3.5 -16.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 4.05 -3.0 -16.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br17 Appearance {

```

```

material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 4.9504 0.5
}
}
]
}
]
}
DEF Textz18 Transform {
translation 5.8 -3.5 -17.5
children [
Shape {
geometry Text {
string ["PEREQUE"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -17.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br18 Appearance {
material Material {
diffuseColor 0 .35 .25

```

```

specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 5.2416 0.5
}
}
]
}
]
}
DEF Textz19 Transform {
translation 5.8 -3.5 -18.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 6.85 -3.0 -18.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br19 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15

```

```

ambientIntensity .0533
shininess      .32
}
}
geometry Box {
size 0.3 5.5328 0.5
}
}
]
}
]
}
DEF Textz20 Transform {
translation 5.8 -3.5 -19.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -19.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br20 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess      .32

```

```

    }
}
geometry Box {
size 0.3 5.824 0.5
}
}
]
}
]
}
DEF Textz21 Transform {
translation 5.8 -3.5 -20.5
children [
Shape {
geometry Text {
string ["TRES LAGOAS"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -20.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br21 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
}
]
}
}

```

```

geometry Box {
size 0.3 6.1152 0.5
}
}
]
}
]
}
}
DEF Textz22 Transform {
translation 5.8 -3.5 -21.5
children [
Shape {
geometry Text {
string ["MAIRINK"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -21.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br22 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
}
]
}
geometry Box {
size 0.3 6.4064 0.5

```

```

}
}
]
}
]
}
DEF Textz23 Transform {
translation 5.8 -3.5 -22.5
children [
Shape {
geometry Text {
string ["SOROCABA"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -22.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br23 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 6.6976 0.5
}
}
}

```

```

]
}
]
}
DEF Textz24 Transform {
translation 5.8 -3.5 -23.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 15.25 -3.0 -23.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br24 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 6.9888 0.5
}
}
]
}
}

```



```

DEF Textz26 Transform {
translation 5.8 -3.5 -25.5
children [
Shape {
geometry Text {
string ["MANOEL BRANDAO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -25.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br26 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 7.5712 0.5
}
}
]
}
]
}
DEF Textz27 Transform {
translation 5.8 -3.5 -26.5

```

```

children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 6.65 -3.0 -26.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br27 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 7.8624 0.5
}
}
]
}
]
}
DEF Textz28 Transform {
translation 5.8 -3.5 -27.5
children [
Shape {

```



```

maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 45.65 -3.0 -28.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br29 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 8.4448 0.5
}
}
]
}
]
}
DEF Textz30 Transform {
translation 5.8 -3.5 -29.5
children [
Shape {
geometry Text {
string ["CHAPADAO DO SUL"]
maxExtent 1
fontStyle FontStyle {

```

```

size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -29.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br30 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 8.736 0.5
}
}
]
}
]
}
DEF Textz31 Transform {
translation 5.8 -3.5 -30.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"

```

```

style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation 3.85 -3.0 -30.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br31 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 9.0272 0.5
}
}
]
}
]
}
DEF Textz32 Transform {
translation 5.8 -3.5 -31.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE

```

```

leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -31.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br32 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 9.3184 0.5
}
}
]
}
]
}
DEF Textz33 Transform {
translation 5.8 -3.5 -32.5
children [
Shape {
geometry Text {
string ["T. OLACYR F. MORAIS"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE

```

```

language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -32.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br33 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 9.6096 0.5
}
}
]
}
DEF Textz34 Transform {
translation 5.8 -3.5 -33.5
children [
Shape {
geometry Text {
string ["PORTOFER CONCEIÇÃOZINHA"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"

```

```

spacing 0.8
}
}
}
]
}
Transform {
translation -2.6 -3.0 -33.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br34 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 9.9008 0.5
}
}
]
}
}
DEF Textz35 Transform {
translation 5.8 -3.5 -34.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
]
}
}

```

```

}
}
]
}
Transform {
translation 3.85 -3.0 -34.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br35 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 10.192 0.5
}
}
]
}
]
}
DEF Textz36 Transform {
translation 5.8 -3.5 -35.5
children [
Shape {
geometry Text {
string ["PARATINGA"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
}

```

```

]
}
Transform {
translation -2.6 -3.0 -35.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br36 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 10.4832 0.5
}
}
]
}
]
}
DEF Textz37 Transform {
translation 5.8 -3.5 -36.5
children [
Shape {
geometry Text {
string ["INTERCAMBIO"]
maxExtent 1
fontStyle FontStyle {
size 0.3
family "SANS"
style "BOLD"
horizontal TRUE
leftToRight TRUE
topToBottom TRUE
language "pr"
justify "MIDDLE"
spacing 0.8
}
}
}
]
}
]
}

```

```
Transform {
translation 3.65 -3.0 -36.5
children [
DEF qtde1 Transform{
translation 0 0.01 0
scale 1 2 1
children [
Shape {
appearance DEF Br37 Appearance {
material Material {
diffuseColor 0 .35 .25
specularColor 0 .31 .36
emissiveColor 0 .13 .15
ambientIntensity .0533
shininess .32
}
}
geometry Box {
size 0.3 10.7744 0.5
}
}
]
}
]
```

## ANEXO B – MODELO DOS DADOS ATUAIS

### SGF\_MOV\_TREM

Nome	Nulo?	Tipo
<b>PFX_TREM</b>	<b>NOT NULL</b>	<b>VARCHAR2(7)</b>
<b>DAT_FORMACAO</b>	<b>NOT NULL</b>	<b>DATE</b>
PFX_LOCAL_ORIGEM	NOT NULL	CHAR(3)
PFX_LOCAL_DESTINO	NOT NULL	CHAR(3)
COD_FERROVIA	NOT NULL	CHAR(2)
COD_CATEGORIA	NOT NULL	CHAR(1)
SEQ_OPERACAO	NOT NULL	NUMBER(3)
SEQ_OPERACAO_LOCAL	NOT NULL	NUMBER(3)
SEQ_LOCAL_PERCURSO	NOT NULL	NUMBER(3)
FLG_SENTIDO		CHAR(1)

### SGF\_MOV\_LOTE

Nome	Nulo?	Tipo
<b>NUM_LOTE</b>	<b>NOT NULL</b>	<b>CHAR(8)</b>
DAT_GERACAO	NOT NULL	DATE
COD_SITUACAO_LOTE	NOT NULL	CHAR(1)
DAT_SITUACAO_LOTE	NOT NULL	DATE
SER_DCL		CHAR(3)
NUM_DCL		NUMBER(6)
COD_MERCADORIA		CHAR(3)
PFX_ESTACAO_ORG		CHAR(3)
PFX_ESTACAO_DST		CHAR(3)
COD_CLIENTE_REMT		CHAR(4)
COD_CLIENTE_DEST		CHAR(4)
DAT_CARGA		DATE
DAT_DESCARGA		DATE
PFX_ESTACAO_ENCERRAMENT		CHAR(3)
DAT_ENCERRAMENTO		DATE

**SGF\_MOV\_CIRCULACAO\_TREM**

Nome	Nulo?	Tipo
PFX_TREM	NOT NULL	VARCHAR2(7)
DAT_FORMACAO	NOT NULL	DATE
SEQ_OPERACAO	NOT NULL	NUMBER(3)
SEQ_LOCAL_PERCURSO	NOT NULL	NUMBER(3)
PFX_LOCAL_PERCURSO		CHAR(3)
NUM_LINHA_PATIO		NUMBER(2)
NUM_MARCO_KM		NUMBER(4)
EXT_PERCURSO_PART_CHEG		NUMBER(4)
PFX_LOCAL_ANTERIOR		CHAR(3)
PFX_LOCAL_SEGUINTE		CHAR(3)
DAT_PARTIDA		DATE
DAT_CHEGADA	NOT NULL	DATE
FLG_EVENTO_OPERACIONAL	NOT NULL	CHAR(1)
COD_EVENTO_OPERACIONAL	NOT NULL	CHAR(3)
DAT_EVENTO_OPERACIONAL	NOT NULL	DATE
COD_SITUACAO_TREM	NOT NULL	CHAR(2)
DAT_SITUACAO_TREM	NOT NULL	DATE
COD_RESPONS_OPERACIONAL	NOT NULL	CHAR(1)
DAT_RESPONS_OPERACIONAL	NOT NULL	DATE
DAT_GRAVACAO	NOT NULL	DATE
MAT_DIGITADOR	NOT NULL	VARCHAR2(10)
COD_DOCUMENTO_DIGITACAO	NOT NULL	CHAR(2)
TXT_OBSERVACAO		VARCHAR2(80)
FLG_EXCLUSAO_MANUT		CHAR(1)
DAT_PREVISAO_CHEGADA		DATE

**SGF\_MOV\_VAGAO**

Nome	Nulo?	Tipo
NUM_VAGAO	NOT NULL	NUMBER(7)
LET_PROPRIEDADE	NOT NULL	CHAR(1)
EXT_COMPRIMENTO	NOT NULL	NUMBER(4,2)
COD_BITOLA	NOT NULL	CHAR(1)
COD_SERIE	NOT NULL	CHAR(3)
PES_TARA	NOT NULL	NUMBER(4,2)
PES_CAPACIDADE_UTIL	NOT NULL	NUMBER(5,2)
VOL_CAPACIDADE_UTIL	NOT NULL	NUMBER(8,2)
PES_TU	NOT NULL	NUMBER(6,3)
COD_EVENTO_OPERACIONAL	NOT NULL	CHAR(3)
DAT_EVENTO_OPERACIONAL	NOT NULL	DATE
COD_SITUACAO	NOT NULL	CHAR(2)
DAT_SITUACAO	NOT NULL	DATE
COD_RESPONS_OPERACIONAL	NOT NULL	CHAR(1)
DAT_RESPONS_OPERACIONAL	NOT NULL	DATE
NUM_VIAGEM	NOT NULL	NUMBER(6)
LET_INICIO_VIAGEM	NOT NULL	CHAR(1)
COD_ESTADO		CHAR(2)

DSC_ESTADO		VARCHAR2(40)
DAT_ESTADO		DATE
PFX_ESTACAO		CHAR(3)
NUM_LINHA		NUMBER(2)
PFX_DESVIO		CHAR(3)
PFX_OFICINA		CHAR(3)
DAT_INICIO_LOCAL		DATE
DAT_INTERCAMBIO		DATE
PFX_TREM		VARCHAR2(7)
DAT_FORMACAO		DATE
SEQ_TREM		NUMBER(3)
PFX_TREM_ANTERIOR		VARCHAR2(7)
DAT_FORMACAO_ANTERIOR		DATE
PFX_DESTINO		CHAR(3)
COD_CLIENTE_DEST		CHAR(4)
NUM_LOTE_CARGA		CHAR(8)
NUM_LOTE_ATUAL		CHAR(8)
DAT_ULT_REVISAO		DATE
TOT_KM_PERC_ULT_REVISAO		NUMBER(5)
DAT_GRAVACAO	NOT NULL	DATE
MAT_DIGITADOR	NOT NULL	VARCHAR2(10)
COD_DOCUMENTO_DIGITACAO	NOT NULL	CHAR(2)
TXT_OBSERVACAO		VARCHAR2(80)

### SGF\_MOV\_LOTE\_VAGAO

Nome	Nulo?	Tipo
<b>NUM_LOTE</b>	<b>NOT NULL</b>	<b>CHAR(8)</b>
<b>NUM_VAGAO</b>	<b>NOT NULL</b>	<b>NUMBER(7)</b>
PES_TU_CARREGADA		NUMBER(6,3)
DAT_TU_CARREGADA		DATE
PES_TU_DESCARGA		NUMBER(6,3)
DAT_TU_DESCARGA		DATE
DAT_INICIO_ESTADIA		DATE
DAT_FIM_ESTADIA		DATE
TMP_ESTADIA		NUMBER(4)
VAL_ESTADIA_CALCULADA		NUMBER(11,2)
VAL_ESTADIA_AUTORIZADA		NUMBER(11,2)
<b>SER_DCL</b>	<b>NOT NULL</b>	<b>CHAR(3)</b>
<b>NUM_DCL</b>	<b>NOT NULL</b>	<b>NUMBER(6)</b>
PFX_LOCAL_CARGA		CHAR(3)
FLG_EVENTO_CARGA		CHAR(1)
PFX_LOCAL_DESCARGA		CHAR(3)
FLG_EVENTO_DESCARGA		CHAR(1)

**SFF\_RELACAO\_DCL**

Nome	Nulo?	Tipo
SER_NOTA	NOT NULL	CHAR(2)
NUM_NOTA	NOT NULL	NUMBER(6)
SER_DCL	NOT NULL	CHAR(3)
NUM_DCL	NOT NULL	NUMBER(6)
DAT_DCL	NOT NULL	DATE
NUM_FXT	NOT NULL	CHAR(5)
DAT_INI_VIGENCIA	NOT NULL	DATE
DAT_FIM_VIGENCIA	NOT NULL	DATE
COD_CLI_CORR	NOT NULL	CHAR(4)
FILIAL_CORR		CHAR(2)
COD_MERC	NOT NULL	CHAR(3)
PFX_ORG	NOT NULL	CHAR(3)
PFX_DST	NOT NULL	CHAR(3)
COD_CLI_REMT	NOT NULL	CHAR(4)
COD_CLI_DEST	NOT NULL	CHAR(4)
PES_CALC	NOT NULL	NUMBER(6,3)
VAL_TARIFA	NOT NULL	NUMBER(6,2)
VAL_FRETE	NOT NULL	NUMBER(11,2)
VAL_TAXAS	NOT NULL	NUMBER(11,2)
QTD_VAG	NOT NULL	NUMBER(3)
COD_ICMS	NOT NULL	CHAR(1)
COD_NATUR_OPER		NUMBER(4)
VAL_BASE_ICMS_REAL	NOT NULL	NUMBER(11,2)
PER_ALIQ_ICMS	NOT NULL	NUMBER(4,2)
VAL_ICMS_REAL	NOT NULL	NUMBER(11,2)
SIT_REL	NOT NULL	CHAR(1)
COD_FERROVIA_GER	NOT NULL	CHAR(2)
COD_UF_ORG	NOT NULL	CHAR(2)
COD_UF_DST	NOT NULL	CHAR(2)

**SFF\_FATURA**

Nome	Nulo?	Tipo
SER_NOTA	NOT NULL	CHAR(2)
NUM_NOTA	NOT NULL	NUMBER(6)
COD_EMIT	NOT NULL	NUMBER(3)
DAT_EXPED	NOT NULL	DATE
VAL_NOTA	NOT NULL	NUMBER(11,2)
DAT_VENC	NOT NULL	DATE
PER_DESC		NUMBER(2)
DAT_DESC		DATE
COD_CLI	NOT NULL	CHAR(4)
FILIAL_CLI	NOT NULL	CHAR(2)
NOM_CLI		VARCHAR2(40)
NUM_FXT	NOT NULL	CHAR(5)
DAT_INI_VIGENCIA		DATE

DAT_FIM_VIGENCIA		DATE
VAL_TARIFA	NOT NULL	NUMBER(6,2)
COD_CGC_CLI		NUMBER(15)
NUM_INSCR_EST_CLI		CHAR(15)
COD_NATUR_OPER		NUMBER(4)
DSC_NATUR_OPER		VARCHAR2(40)
COD_TIP_PESSOA_CLI		CHAR(1)
END_CLI		VARCHAR2(40)
NOM_BAIRRO_CLI		VARCHAR2(40)
COD_MUNIC_CLI		NUMBER(6)
NOM_MUNIC_CLI		VARCHAR2(40)
COD_UF_CLI		CHAR(2)
COD_CEP_CLI		CHAR(8)
NOM_PRACA_PAGTO_CLI		VARCHAR2(40)
VAL_MULTA		NUMBER(8,2)
COD_ICMS		CHAR(1)
VAL_BASE_CALC_ICMS		NUMBER(11,2)
DSC_BASE_CALC_ICMS		VARCHAR2(20)
PER_ALIQ_ICMS		NUMBER(4,2)
VAL_ICMS		NUMBER(11,2)
SIT_NOTA	NOT NULL	CHAR(1)
DSC_VAL_EXT1		VARCHAR2(55)
DSC_VAL_EXT2		VARCHAR2(55)
DSC_VAL_EXT3		VARCHAR2(55)
DSC_ICMS_MSG1		VARCHAR2(50)
DSC_ICMS_MSG2		VARCHAR2(50)
DSC_ICMS_MSG3		VARCHAR2(50)
DSC_ICMS_MSG4		VARCHAR2(50)
DSC_MSG1		VARCHAR2(50)
DSC_MSG2		VARCHAR2(50)
FLG_SUBST_TRIB		CHAR(1)
SER_SELO_FISCAL		CHAR(2)
NUM_SELO_FISCAL		CHAR(8)
DAT_CANCEL		DATE
DSC_MOT_CANCEL		VARCHAR2(76)

## SGF\_CAD\_CLIENTE

Nome	Nulo?	Tipo
<b>COD_CLIENTE</b>	<b>NOT NULL</b>	<b>CHAR(4)</b>
SGL_CLIENTE		VARCHAR2(10)
NOM_RAZAO_SOCIAL		VARCHAR2(40)
NOM_FANTASIA		VARCHAR2(15)
FLG_REMUNERACAO		CHAR(1)

**SGF\_CAD\_MERCADORIA**

Nome	Nulo?	Tipo
<b>COD_MERCADORIA</b>	<b>NOT NULL</b>	<b>CHAR(3)</b>
DSC_MERCADORIA		VARCHAR2(65)
COD_PRODUTO		CHAR(3)
DEN_MERCADORIA		NUMBER(7,6)

**SGF\_CAD\_ESTACAO**

Nome	Nulo?	Tipo
<b>PFX_LOCAL</b>	<b>NOT NULL</b>	<b>CHAR(3)</b>
NOM_LOCAL		VARCHAR2(40)
COD_TIPO_LOCAL		CHAR(1)
COD_SITUACAO_LOCAL		CHAR(1)
COD_MUNICIPIO		NUMBER(6)

**SGF\_APURRA\_PROD\_GRUPO\_FLUXO\_ME**

Nome	Nulo?	Tipo
<b>DAT_EVENTO</b>	<b>NOT NULL</b>	<b>DATE</b>
<b>NUM_FLUXO</b>	<b>NOT NULL</b>	<b>VARCHAR2(5)</b>
<b>COD_FERROVIA</b>		<b>CHAR(2)</b>
<b>COD_MERCADORIA</b>		<b>CHAR(3)</b>
PFX_ESTACAO_ORG		CHAR(3)
PFX_ESTACAO_DST		CHAR(3)
COD_CLIENTE_REMT		CHAR(4)
COD_CLIENTE_DEST		CHAR(4)
TOT_VAGAO_CRG		NUMBER(4)
TOT_TU_CRG		NUMBER(13,3)
TOT_VAGAO_DCG		NUMBER(4)
TOT_TU_DCG		NUMBER(13,3)
TOT_TKU_CRG_FN		NUMBER(15,3)
TOT_TKU_CRG_FB		NUMBER(15,3)
TOT_TKU_CRG_NV		NUMBER(15,3)
TOT_TKU_CRG_MS		NUMBER(15,3)
RECEITA_TOTAL		NUMBER(13,2)
RECEITA_TOTAL_SICMS		NUMBER(13,2)
RECEITA_FN		NUMBER(13,2)
RECEITA_FN_SICMS		NUMBER(13,2)
RECEITA_FB		NUMBER(13,2)
RECEITA_FB_SICMS		NUMBER(13,2)
RECEITA_NV		NUMBER(13,2)
RECEITA_NV_SICMS		NUMBER(13,2)
RECEITA_MS		NUMBER(13,2)
RECEITA_MS_SICMS		NUMBER(13,2)
RECEITA_OF		NUMBER(13,2)

RECEITA_OF_SICMS		NUMBER(13,2)
ICMS_FN		NUMBER(13,2)
ICMS_FB		NUMBER(13,2)
ICMS_NV		NUMBER(13,2)
ICMS_MS		NUMBER(13,2)
ICMS_OF		NUMBER(13,2)
DISTANCIA_FN		NUMBER(5,1)
DISTANCIA_FB		NUMBER(5,1)
DISTANCIA_NV		NUMBER(5,1)
DISTANCIA_MS		NUMBER(5,1)
NUM_CONTRATO		VARCHAR2(15)
COD_ICMS		CHAR(1)
PER_ALIQUOTA_ICMS		NUMBER(4,2)

### SGF\_CAD\_CONTRATO\_VIG\_TAREFA

Nome	Nulo?	Tipo
NUM_CONTRATO	NOT NULL	VARCHAR(15)
DAT_INI_VIGENCIA	NOT NULL	DATE
DAT_FIM_VIGENCIA	NOT NULL	DATE
COD_FERROVIA	NOT NULL	CHAR(2)
VAL_TARIFA		NUMBER(8,2)
COD_TIPO_INTERCAMBIO	NOT NULL	CHAR(1)

### SGF\_CAD\_CONTRATO\_CLIENTE

Nome	Nulo?	Tipo
NUM_CONTRATO	NOT NULL	VARCHAR2(15)
COD_CLIENTE_REMT	NOT NULL	CHAR(4)
COD_FILIAL_REMT	NOT NULL	CHAR(2)
COD_CLIENTE_DEST	NOT NULL	CHAR(4)
COD_FILIAL_DEST	NOT NULL	CHAR(2)
COD_CLIENTE_CORR	NOT NULL	CHAR(4)
COD_FILIAL_CORR	NOT NULL	CHAR(2)
COD_COND_FRETE	NOT NULL	NUMBER(2)
COD_CLIENTE_FATUR		CHAR(4)
COD_FILIAL_FATUR		CHAR(2)

## ANEXO C – PROGRAMA FONTE JAVA

```
// import com.buk.pacotes.SetaDados;
import java.awt.*;
import java.util.*;
import java.awt.event.*;
import vrml.*;
import vrml.node.*;
import vrml.field.*;

public class Escala extends Script
{
    private SFVec3f escalaDiasDaSemana[];
    private SFVec3f posicaoDiasDaSemana[];

    // Variáveis de controle do Java que são associadas aos tipos do VRML
    // Para cada nó VRML há uma função Java que copia o valores

    float esc[]={1, 1, 1, 1, 1, 1, 1}; // Escala para os dias da semana

    // Lista de eventos que o Cortona envia para o Java quando o nó
    // Script do VRML é acionado
    static String[] eventosEscala = {"segundaEscala", "tercaEscala",
    "quartaEscala", "quintaEscala", "sextaEscala", "sabadoEscala",
    "domingoEscala"};
    static String[] eventosPosicao = {"segundaPosicao", "tercaPosicao",
    "quartaPosicao", "quintaPosicao", "sextaPosicao",
    "sabadoPosicao", "domingoPosicao"};

    // Obtém os valores iniciais da cena VRML
    // É chamado quando nó Script é criado
    // Cada um dos campos recebe os valores iniciais definidos na cena
    public void initialize (){
        System.out.println("initialize - begin ");
        escalaDiasDaSemana = new SFVec3f[7];
        posicaoDiasDaSemana = new SFVec3f[7];
        for (int i=0; i<7; i++){
            escalaDiasDaSemana[i] = (SFVec3f) getEventOut
            (eventosEscala[i]);
            posicaoDiasDaSemana[i] = (SFVec3f) getEventOut
            (eventosPosicao[i]);
        }
    }
}
```

```
        System.out.println("initialize - end ");
    }

    void leiaDados(){
        for (int i=0; i<7; i++){
            // Cria os valores que deveriam ser buscados no BD
            esc[i] = (float) (Math.random()*10);
        }
    }

    void atualizaGrafico(){
        for (int i=0; i<7; i++){
            // Atualiza cena
            escalaDiasDaSemana[i].setValue(1, esc[i], 1);
            posicaoDiasDaSemana[i].setValue((float)0.5+i,(esc[i]/2), 0);
        }
        System.out.println("Valor: " + esc[0] + " -- " + (esc[0]/2));
    }

    public void processEvent(vrml.Event e){
        String s =""+ e.getValue();
        try{
            Thread.sleep(10);
        } catch (java.lang.InterruptedExcepction err){

        }
        leiaDados();
        atualizaGrafico();
    }
}
```

## ANEXO D – STORED PROCEDURES

```
CREATE OR REPLACE PACKAGE GETCARGA
AS
TYPE TBLCODCLI IS TABLE OF SAC_CLIENTES.COD_CLIENTE%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLNOMCLI IS TABLE OF SAC_LOGIN.NOME%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLCODMER IS TABLE OF SGF_CAD_FLUXO.COD_MERCADORIA%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLDSCMER IS TABLE OF SGF_CAD_MERCADORIA.DSC_MERCADORIA%TYPE INDEX BY
BINARY_INTEGER;
TYPE TBLPFXEST IS TABLE OF SGF_CAD_FLUXO.PFX_ESTACAO_ORG%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLNOMEST IS TABLE OF SGF_CAD_LOCAL.NOM_LOCAL%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLPFXESTD IS TABLE OF SGF_CAD_FLUXO.PFX_ESTACAO_ORG%TYPE INDEX BY
BINARY_INTEGER;
TYPE TBLNOMESTD IS TABLE OF SGF_CAD_LOCAL.NOM_LOCAL%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLPFXESTO IS TABLE OF SGF_CAD_FLUXO.PFX_ESTACAO_ORG%TYPE INDEX BY
BINARY_INTEGER;
TYPE TBLNOMESTO IS TABLE OF SGF_CAD_LOCAL.NOM_LOCAL%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLPFXTREM IS TABLE OF SGF_MOV_VAGAO.PFX_TREM%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLNUMVAG IS TABLE OF SGF_MOV_VAGAO.NUM_VAGAO%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLTOTVAG IS TABLE OF NUMBER(12,2) INDEX BY BINARY_INTEGER;
TYPE TBLNUMDCL IS TABLE OF NUMBER(6) INDEX BY BINARY_INTEGER;
TYPE TBLSERDCL IS TABLE OF VARCHAR(10) INDEX BY BINARY_INTEGER;
TYPE TBLDTCARG IS TABLE OF SGF_MOV_LOTE.DAT_CARGA%TYPE INDEX BY BINARY_INTEGER;
TYPE TBLSERVAG IS TABLE OF VARCHAR(11) INDEX BY BINARY_INTEGER;
TYPE TBLQTDetu IS TABLE OF SGF_MOV_VAGAO.PES_TU%TYPE INDEX BY BINARY_INTEGER;

PROCEDURE GETLOGIN
(I_LOGIN IN SAC_CLIENTES.LOGIN%TYPE,
 O_CODCLI OUT TBLCODCLI);

PROCEDURE GETCLIENTE
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
 I_SENHA IN SAC_LOGIN.SENHA%TYPE,
 O_NOMCLI OUT TBLNOMCLI);

PROCEDURE GETMERCADORIA
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
 O_CODMER OUT TBLCODMER,
 O_DSCMER OUT TBLDSCMER);

PROCEDURE GETESTACAO
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
 O_PFXEST OUT TBLPFXEST,
 O_NOMEST OUT TBLNOMEST);

PROCEDURE GETCONSULTA
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
 I_CODMER IN SGF_CAD_MERCADORIA.COD_MERCADORIA%TYPE,
 I_CODEST IN SGF_CAD_LOCAL.PFX_LOCAL%TYPE,
 O_CODMER OUT TBLCODMER,
 O_DSCMER OUT TBLDSCMER,
 O_PFESTO OUT TBLPFXESTO,
 O_NOMLOO OUT TBLNOMESTO,
 O_PFESTD OUT TBLPFXESTD,
 O_NOMLOD OUT TBLNOMESTD,
 O_PFXLOC OUT TBLPFXEST,
 O_NOMLOC OUT TBLNOMEST,
 O_PFXTREM OUT TBLPFXTREM,
 O_TOTVAG OUT TBLTOTVAG);
```

```

PROCEDURE GETDETALHE
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
 I_CODMER IN SGF_CAD_MERCADORIA.COD_MERCADORIA%TYPE,
 I_PFESTO IN SGF_CAD_LOCAL.PFX_LOCAL%TYPE,
 I_PFESTD IN SGF_CAD_LOCAL.PFX_LOCAL%TYPE,
 I_PFXLOC IN SGF_CAD_LOCAL.PFX_LOCAL%TYPE,
 I_PFXTREM IN SGF_MOV_VAGAO.PFX_TREM%TYPE,
 O_SERDCL OUT TBLSERDCL,
 O_NUMDCL OUT TBLNUMDCL,
 O_DTCARG OUT TBLDTCARG,
 O_SERVAG OUT TBLSERVAG,
 O_NUMVAG OUT TBLNUMVAG,
 O_QTDEU OUT TBLQTDEU);

END GETCARGA;

CREATE OR REPLACE PACKAGE BODY GETCARGA
AS

PROCEDURE GETLOGIN
(I_LOGIN IN SAC_CLIENTES.LOGIN%TYPE,
 O_CODCLI OUT TBLCODCLI)
IS
CURSOR CURLOGIN IS
SELECT COD_CLIENTE FROM SAC_CLIENTES WHERE LOGIN = I_LOGIN;
RECORDCOUNT NUMBER DEFAULT 0;

BEGIN

FOR CUR_LOGIN IN CURLOGIN LOOP
    RecordCount := RecordCount + 1;
    O_CODCLI(RecordCount) := CUR_LOGIN.COD_CLIENTE;
END LOOP;

END GETLOGIN;

PROCEDURE GETCLIENTE
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
 I_SENHA IN SAC_LOGIN.SENHA%TYPE,
 O_NOMCLI OUT TBLNOMCLI)
IS
CURSOR CURNOME IS
SELECT NOME FROM SAC_LOGIN WHERE LOGIN = I_LOGIN AND SENHA = I_SENHA;
RECORDCOUNT NUMBER DEFAULT 0;

BEGIN

FOR CUR_NOME IN CURNOME LOOP
    RecordCount := RecordCount + 1;
    O_NOMCLI(RecordCount) := CUR_NOME.NOME;
END LOOP;

END GETCLIENTE;

```

```

PROCEDURE GETMERCADORIA
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
O_CODMER OUT TBLCODMER,
O_DSCMER OUT TBLDSCMER)
IS
CURSOR CUR_PESMERCA IS
SELECT DISTINCT F.COD_MERCADORIA, M.DSC_MERCADORIA FROM SGF_CAD_FLUXO
F,SGF_CAD_MERCADORIA M
WHERE M.COD_MERCADORIA = F.COD_MERCADORIA AND F.COD_CLIENTE_DEST IN (SELECT
COD_CLIENTE
FROM SAC_CLIENTES
WHERE LOGIN = I_LOGIN)
ORDER BY M.DSC_MERCADORIA;
RECORDCOUNT NUMBER DEFAULT 0;

BEGIN

FOR CURRECPESMERCA IN CUR_PESMERCA LOOP
RecordCount := RecordCount + 1;
O_CODMER(RecordCount) := CURRECPESMERCA.COD_MERCADORIA;
O_DSCMER(RecordCount) := CURRECPESMERCA.DSC_MERCADORIA;
END LOOP;
END GETMERCADORIA;

PROCEDURE GETESTACAO
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
O_PFXEST OUT TBLPFXEST,
O_NOMEST OUT TBLNOMEST)
IS
CURSOR CUR_PESESTA IS
SELECT DISTINCT F.PFX_ESTACAO_ORG, L.NOM_LOCAL
FROM SGF_CAD_FLUXO F,SGF_CAD_LOCAL L
WHERE L.PFX_LOCAL = F.PFX_ESTACAO_ORG AND F.COD_CLIENTE_DEST IN (SELECT COD_CLIENTE
FROM SAC_CLIENTES
WHERE LOGIN = I_LOGIN)
ORDER BY F.PFX_ESTACAO_ORG;
RECORDCOUNT NUMBER DEFAULT 0;

BEGIN

FOR CURRECPESESTA IN CUR_PESESTA LOOP
RecordCount := RecordCount + 1;
O_PFXEST(RecordCount) := CURRECPESESTA.PFX_ESTACAO_ORG;
O_NOMEST(RecordCount) := CURRECPESESTA.NOM_LOCAL;
END LOOP;

END GETESTACAO;

```

```

PROCEDURE GETCONSULTA
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
I_CODMER IN SGF_CAD_MERCADORIA.COD_MERCADORIA%TYPE,
I_CODEST IN SGF_CAD_LOCAL.PFX_LOCAL%TYPE,
O_CODMER OUT TBLCODMER,
O_DSCMER OUT TBLDSCMER,
O_PFESTO OUT TBLPFEXESTO,
O_NOMLOO OUT TBLNOMESTO,
O_PFESTD OUT TBLPFEXESTD,
O_NOMLOD OUT TBLNOMESTD,
O_PFXLOC OUT TBLPFEXEST,
O_NOMLOC OUT TBLNOMEST,
O_PFXTREM OUT TBLPFXTREM,
O_TOTVAG OUT TBLTOTVAG)
IS

CURSOR CUR_CONSULTA IS
SELECT DISTINCT
  lo.cod_mercadoria as codMerc,
  m.dsc_mercadoria as Merc,
  lo.pfx_estacao_org as Org,
  lorg.nom_local as DscOrg,
  lo.pfx_estacao_dst as Dst,
  ldst.nom_local as DscDst,
  tc.pfx_local_percurso as Local,
  lloc.nom_local as DscLoc,
  v.pfx_trem as Trem,
  COUNT(*) as TotVg
FROM sgf_mov_trem t,
  sgf_mov_circulacao_trem tc,
  sgf_mov_vagao v,
  sgf_mov_lote_vagao l,
  sgf_mov_lote lo,
  sgf_cad_mercadoria m,
  sgf_cad_local lorg,
  sgf_cad_local ldst,
  sgf_cad_local lloc
WHERE v.num_lote_carga = lo.num_lote
  AND l.num_vagao = v.num_vagao
  AND l.num_lote = v.num_lote_carga
  AND lo.cod_mercadoria = m.cod_mercadoria
  AND lo.pfx_estacao_org = lorg.pfx_local(+)
  AND lo.pfx_estacao_dst = ldst.pfx_local(+)
  AND tc.pfx_local_percurso = lloc.pfx_local(+)
  AND (v.pfx_trem = t.pfx_trem
  AND v.dat_formacao = t.dat_formacao
  AND t.seq_operacao = tc.seq_operacao
  AND tc.pfx_trem = v.pfx_trem
  AND tc.dat_formacao = v.dat_formacao)
  AND v.pes_tu > 0
  AND lo.cod_cliente_dest IN (SELECT cod_cliente FROM sac_clientes WHERE login = i_login)
GROUP BY m.dsc_mercadoria,
  lo.cod_mercadoria,
  lo.pfx_estacao_org,
  lorg.nom_local,
  lo.pfx_estacao_dst,
  ldst.nom_local,
  tc.pfx_local_percurso,
  lloc.nom_local,
  v.pfx_trem
UNION ALL

```

```

SELECT DISTINCT
  lo.cod_mercadoria as codmerc,
  m.dsc_mercadoria as Merc,
  lo.pfx_estacao_org as Org,
  lorg.nom_local as DscOrg,
  lo.pfx_estacao_dst as Dst,
  ldst.nom_local as DscDst,
  v.pfx_estacao as Local,
  lloc.nom_local as DscLoc,
  v.pfx_trem as Trem,
  COUNT(*) as TotVg
FROM sgf_mov_lote_vagao l,
  sgf_mov_vagao v,
  sgf_mov_lote lo,
  sgf_cad_mercadoria m,
  sgf_cad_local lorg,
  sgf_cad_local ldst,
  sgf_cad_local lloc
WHERE v.num_lote_carga = lo.num_lote
  AND l.num_vagao = v.num_vagao
  AND l.num_lote = v.num_lote_carga
  AND lo.cod_mercadoria = m.cod_mercadoria
  AND lo.pfx_estacao_org = lorg.pfx_local(+)
  AND lo.pfx_estacao_dst = ldst.pfx_local(+)
  AND v.pfx_estacao = lloc.pfx_local(+)
  AND v.pfx_trem is null
  AND v.pes_tu > 0
  AND lo.cod_cliente_dest IN (SELECT cod_cliente FROM SAC_CLIENTES WHERE login = i_login )
GROUP BY m.dsc_mercadoria,
  lo.cod_mercadoria,
  lo.pfx_estacao_org,
  lorg.nom_local,
  lo.pfx_estacao_dst,
  ldst.nom_local,
  v.pfx_estacao,
  lloc.nom_local,
  v.pfx_trem
ORDER BY 2, 3, 4, 5, 6;
RECORDCOUNT NUMBER DEFAULT 0;

BEGIN

FOR CURCONSULTA IN CUR_CONSULTA LOOP
  RecordCount := RecordCount + 1;
  O_CODMER(RecordCount) := CURCONSULTA.CODMERC;
  O_DSCMER(RecordCount) := CURCONSULTA.MERC;
  O_PFESTO(RecordCount) := CURCONSULTA.ORG;
  O_NOMLOO(RecordCount) := CURCONSULTA.DSCORG;
  O_PFESTD(RecordCount) := CURCONSULTA.DST;
  O_NOMLOD(RecordCount) := CURCONSULTA.DSCDST;
  O_PFXLOC(RecordCount) := CURCONSULTA.LOCAL;
  O_NOMLOC(RecordCount) := CURCONSULTA.DSCLOC;
  O_PFXTREM(RecordCount) := CURCONSULTA.TREM;
  O_TOTVAG(RecordCount) := CURCONSULTA.TOTVG;
END LOOP;

END GETCONSULTA;

```

```

PROCEDURE GETDETALHE
(I_LOGIN IN SAC_LOGIN.LOGIN%TYPE,
I_CODMER IN SGF_CAD_MERCADORIA.COD_MERCADORIA%TYPE,
I_PFESTO IN SGF_CAD_LOCAL.PFX_LOCAL%TYPE,
I_PFESTD IN SGF_CAD_LOCAL.PFX_LOCAL%TYPE,
I_PFXLOC IN SGF_CAD_LOCAL.PFX_LOCAL%TYPE,
I_PFXTREM IN SGF_MOV_VAGAO.PFX_TREM%TYPE,
O_SERDCL OUT TBLSERDCL,
O_NUMDCL OUT TBLNUMDCL,
O_DTCARG OUT TBLDTCARG,
O_SERVAG OUT TBLSERVAG,
O_NUMVAG OUT TBLNUMVAG,
O_QTDETU OUT TBLQTDETU)
IS

CURSOR CUR_DETALHE IS

SELECT DISTINCT
l.ser_dcl                AS SERDCL,
l.num_dcl                AS NUMDCL,
lo.dat_carga             AS DtCarrega,
v.cod_serie              AS Ser_Vagao,
v.num_vagao              AS Num_Vagao,
v.pes_tu                 AS TU
FROM sgf_mov_vagao v,
     sgf_mov_lote_vagao l,
     sgf_mov_lote lo,
     sgf_mov_trem t,
     sgf_mov_circulacao_trem tc,
     sff_relacao_dcl rd,
     sff_fatura f
WHERE v.num_lote_carga = lo.num_lote
      AND l.num_vagao = v.num_vagao
      AND l.num_lote = v.num_lote_carga
      AND l.ser_dcl = rd.ser_dcl(+)
      AND l.num_dcl = rd.num_dcl(+)
      AND rd.ser_nota = f.ser_nota(+)
      AND rd.num_nota = f.num_nota(+)
      AND (v.pfx_trem = t.pfx_trem
            AND v.dat_formacao = t.dat_formacao
            AND t.seq_operacao = tc.seq_operacao
            AND tc.pfx_trem = v.pfx_trem
            AND tc.dat_formacao = v.dat_formacao)
      AND v.pes_tu > 0
      AND lo.cod_cliente_dest IN (SELECT cod_cliente FROM sac_clientes WHERE login = I_login)
      AND lo.cod_mercadoria = i_codmer
      AND lo.pfx_estacao_org = i_pfesto
      AND lo.pfx_estacao_dst = i_pfestd
      AND v.pfx_trem = i_pfxtrem
      AND tc.pfx_local_percurso = i_pfxloc
ORDER BY 1;
RECORDCOUNT NUMBER DEFAULT 0;

BEGIN

FOR CURDETALHE IN CUR_DETALHE LOOP
  RecordCount := RecordCount + 1;
  O_SERDCL(RecordCount) := CURDETALHE.SERDCL;
  O_NUMDCL(RecordCount) := CURDETALHE.NUMDCL;
  O_DTCARG(RecordCount) := CURDETALHE.DTCARREGA;
  O_SERVAG(RecordCount) := CURDETALHE.SER_VAGAO;
  O_NUMVAG(RecordCount) := CURDETALHE.NUM_VAGAO;
  O_QTDETU(RecordCount) := CURDETALHE.TU;
END LOOP;

END GETDETALHE;

END GETCARGA;

```

## ANEXO E – PROGRAMA EM ASP CRIANDO UM ARQUIVO VRML

```
<html>
<head>
<title>Brasil Ferrovias</title>
</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">

<%
DIM cArquivo
DIM cCh
cCH = CHR(34)

cArquivo = "Grafico.wrl"

Set ObjFSO = CreateObject("Scripting.FileSystemObject")
ObjFSO.CreateTextFile(cArquivo)
Set objFile = ObjFSO.GetFile(cArquivo)
Set objStream = ObjFile.OpenAsTextStream(2,0)

objStream.WriteLine "#VRML V2.0 utf8"

objStream.WriteLine "Background {"
objStream.WriteLine "    skyColor 0.5 0.5 0.5"
objStream.WriteLine "}"

objStream.WriteLine "DEF TextSegunda Transform {"
objStream.WriteLine "    translation      0.5 0 0"
objStream.WriteLine "    children ["
objStream.WriteLine "        Shape {"
objStream.WriteLine "            geometry Text {"
objStream.WriteLine "                string [ "&cCh &"Seg" &cCh &"]"
objStream.WriteLine "                maxExtent 1"
objStream.WriteLine "                length 1"
objStream.WriteLine "                fontStyle FontStyle {"
objStream.WriteLine "                    size 0.5"
objStream.WriteLine "                    family "&cCh &"SERIF" &cCh
objStream.WriteLine "                    style "&cCh &"BOLD" &cCh
objStream.WriteLine "                    horizontal FALSE"
objStream.WriteLine "                leftToRight TRUE"
objStream.WriteLine "                topToBottom TRUE"
objStream.WriteLine "                language "&cCh &"pr" &cCh
objStream.WriteLine "                justify "&cCh &"BEGIN" &cCh
objStream.WriteLine "                spacing 0.1"
objStream.WriteLine "            }"
objStream.WriteLine "        }"
objStream.WriteLine "    ]"
objStream.WriteLine "}"

objStream.WriteLine "]"
objStream.WriteLine "}"

objStream.WriteLine "Background {"
objStream.WriteLine "    skyColor 0.5 0.5 0.5"
objStream.WriteLine "}"
objStream.WriteLine "DEF Planilha Group {"
objStream.WriteLine "    children ["
objStream.WriteLine "        DEF EixoY Transform{"
objStream.WriteLine "            translation      0 4 0"
```

