

JOHNATAN DANIEL FROMHOLZ LIMA

**UMA INTERFACE PARA A PERSISTÊNCIA DE FENÔMENOS
TERRESTRES REPRESENTADOS COMO OBJETOS**

**Dissertação apresentada como
requisito parcial para obtenção do
grau de Mestre em Informática –
Área de Engenharia de Software
ao Curso de Pós-Graduação em
Informática, Setor de Ciências
Exatas, Universidade Federal do
Paraná**

**Orientador: Prof. Dr. Carlos A. P.
de Carvalho**

CURITIBA

2004

Dedicatória

Dedico este trabalho a minha esposa Kathryn Zimmermann pelo apoio e motivação.

Agradecimentos

Ao Prof. Carlos Picanço pela oportunidade e pela orientação.

Ao Prof. Antônio Berutti pela ajuda e por compartilhar de seus conhecimentos.

A todos os colegas que me ajudaram e alguma forma durante esta jornada.

SUMÁRIO

LISTA DE FIGURAS.....	V
LISTA DE QUADROS.....	VI
LISTA DE TABELAS.....	VII
LISTA DE SIGLAS.....	VIII
RESUMO.....	IX
1 INTRODUÇÃO	1
2 CONTEXTUALIZAÇÃO.....	3
2.1 FATOS E FENÔMENOS TERRESTRES	3
2.2 ÁREA DE LEVANTAMENTO	6
2.3 ESCOPO DO TRABALHO.....	7
2.4 OBJETIVO GERAL.....	8
2.4.1 Objetivos específicos.....	9
3 REVISÃO BIBLIOGRÁFICA	10
3.1 TRABALHOS RELACIONADOS	10
3.2 PERSISTÊNCIA DE OBJETOS	12
3.3 FORMAS DE PERSISTÊNCIA.....	12
3.3.1 Banco de Dados Orientado a Objetos.....	12
3.3.2 Banco de Dados Objeto-Relacional	13
3.3.3 Serialização.....	15
3.3.4 Banco de Dados Relacional.....	15
3.3.4.1 Camada de Persistência	17
3.3.4.2 Mapeamento objeto-relacional.....	17
3.3.4.3 Identificadores de Objetos	18
3.3.4.4 Um Exemplo de Mapeamento Objeto-Relacional.....	20
3.3.4.5 Automação do processo de mapeamento.....	21
4 MODELO DOS FENÔMENOS TERRESTRES	23
4.1 CARACTERIZAÇÃO DOS FENÔMENOS COMO OBJETOS	24
4.1.1 Modelo de Representação dos Fenômenos como Objetos.....	24
5 O EXPERIMENTO.....	30
5.1 FERRAMENTAS UTILIZADAS	30
5.2 MÉTODO DE ABORDAGEM	31
5.2.1 Forma de persistência	31
5.2.2 Método para Persistência	32
5.2.3 Método de Localização De Fenômenos	33
5.3 MODELAGEM DA INTERFACE DE PERSISTÊNCIA EM UML.....	34
5.3.1 Diagrama De Classes.....	35
5.3.2 As Classes e seus Significados	38
5.3.3 Diagramas de Seqüência.....	39
5.3.4 Diagrama de seqüência armazena fenômenos	42
5.3.5 Diagrama de seqüência recupera fenômenos	42
5.4 FORMAS DE PERSISTÊNCIA PARA O MODELO DE REFERÊNCIA	43
5.4.1 O Modelo de Referência adaptado para SGBDR	45
5.4.1.1 Dicionário de dados.....	46
5.4.1.2 O Modelo de Referência adaptado para XML.....	49
6 UMA APLICAÇÃO COMO ESTUDO DE CASO	55
6.1 CASOS DE USO	55
6.1.1 Caso De Uso Armazenar Fenômeno.....	55
6.1.2 Caso De Uso Recuperar Fenômeno	56
6.2 ESPECIFICAÇÃO DE CASOS DE USO	57
6.2.1 Especificação do Caso de Uso Armazenar Fenômeno.....	57
6.2.2 Especificação do Caso de Uso Recuperar Fenômeno.....	59
6.2.3 CONSIDERAÇÕES	61

6.3	TESTANDO A APLICAÇÃO.....	62
6.3.1	Obtendo dados.....	62
6.3.2	Armazenando Fenômenos.....	64
6.3.3	Recuperando Fenômenos.....	64
6.3.4	Exportando e Importando em Formato XML.....	66
7	UMA INTEGRAÇÃO PARA TESTE	67
7.1	A APLICAÇÃO VFT.....	67
7.2	ADAPTAÇÕES NECESSÁRIAS	67
7.2.1	Classe BdFTBean.....	67
7.2.2	Alterações na classe BdFTBean	68
8	CONCLUSÃO, RESULTADOS E TRABALHOS FUTUROS.....	70
9	REFERÊNCIAS BIBLIOGRÁFICAS.....	72
10	DOCUMENTOS CONSULTADOS	73
	APÊNDICE I – ARTICULAÇÃO DE CARTAS DE FOLHAS TOPOGRÁFICAS	76
	APÊNDICE II – TUTORIAL SOBRE MAPEAMENTO OBJETO-RELACIONAL	86

LISTA DE FIGURAS

FIGURA 1 – VISÕES DO MESMO FATO	3
FIGURA 2 – SUPERFÍCIE TERRESTRE, TOPOGRÁFICA E CORPO HÍDRICO	4
FIGURA 3 – OS FENÔMENOS TERRESTRES NA NATUREZA	4
FIGURA 4 – FENOMENOS TERRESTRES DE FORMA ISOLADA	6
FIGURA 5 – UMA ÁREA DE LEVANTAMENTO EXTRAÍDA DA NATUREZA	7
FIGURA 6 – INTERFACE DE PERSISTÊNCIA DE FENÔMENOS TERRESTRES	7
FIGURA 7 – ESCOPO DO TRABALHO	8
FIGURA 8 – CAMADAS DE INFORMAÇÃO EM SIG	10
FIGURA 9 – CAMADAS DE INFORMAÇÃO ASSOCIADAS À UMA ÚNICA INFORMAÇÃO	11
FIGURA 10 – ESQUEMA PARA PERSISTÊNCIA	17
FIGURA 11 – ESTRURA DE CLASSES	20
FIGURA 12- ESTRUTURA DE TABELAS PARA O MAPEAMENTO	21
FIGURA 13 – CARACTERIZAÇÃO DE FENÔMENOS TERRESTRES	24
FIGURA 14 – MODELO DE REFERÊNCIA	25
FIGURA 15 - SUPERFÍCIE DO FENÔMENO TERRESTRE	27
FIGURA 16 – CLASSE ÁREA DE LEVANTAMENTO	28
FIGURA 17 – JANELA DE VISUALIZAÇÃO, ÁREA DE INTERESSE E ÁREA DE LEVANTAMENTO	29
FIGURA 18 – CLASSE JANELA DE VISUALIZAÇÃO	29
FIGURA 19 – CLASSE ÁREA DE INTERESSE	29
FIGURA 20 – MÉTODO DE PERSISTÊNCIA	33
FIGURA 21 – MÉTODO DE LOCALIZAÇÃO	34
FIGURA 22 - DIAGRAMA DE CLASSES	37
FIGURA 23 – DIAGRAMA DE SEQÜÊNCIA ARMAZENA FENÔMENOS	40
FIGURA 24 - DIAGRAMA DE SEQÜÊNCIA RECUPERA FENÔMENOS	41
FIGURA 25 – CLASSES PARA PERSISTÊNCIA EM MEIOS ALTERNATIVOS	45
FIGURA 26 – MODELO E/R NO BANCO DE DADOS	46
FIGURA 27 – ARQUITETURA JAXB	50
FIGURA 28 - UTILIZAÇÃO DO COMPILADOR XJC PARA GERAR CLASSES AUXILIARES	53
FIGURA 29 – CASO DE USO ARMAZENAR FENOMENO	56
FIGURA 30 – CASO DE USO RECUPERAR FENOMENO	57
FIGURA 31 - ÁREA DE LEVANTAMENTO VISTA NO MAXICAD	63
FIGURA 32 – INTERFACE COM USUÁRIO PARA DETERMINAR A JANELA DE VISUALIZAÇÃO	65
FIGURA 33 – INTERFACE COM USUÁRIO PRA DETERMINAR OS CRITÉRIOS SEMÂNTICOS	65

LISTA DE QUADROS

QUADRO 1 – PROCESSO DE SERIALIZAÇÃO.....	15
QUADRO 2- AUTOMAÇÃO DO PROCESSO DE MAPEAMENTO UTILIZANDO UMA FERRAMENTA	22
QUADRO 3 - ARQUIVO FENOMENOS.XSD.....	50
QUADRO 4 – CÓDIGO PARA O PROCESSO DE MARSHALL	54
QUADRO 5- CÓDIGO PARA O PROCESSO DE UNMARSHALL.....	54
QUADRO 6 – ESPECIFICAÇÃO DO CASO DE USO ARMAZENAR FENÔMENO	58
QUADRO 7 – ESPECIFICAÇÃO DO CASO DE USO RECUPERAR FENÔMENO	60
QUADRO 8 – TRECHO DE UMA ÁREA DE LEVANTAMENTO EM FORMATO TEXTO	63
QUADRO 9 - ARQUIVO FENOMENOS.XML.....	66
QUADRO 10 – TRECHOS DE CÓDIGO DA CLASSE BdFTBEAN ALTERADA	68

LISTA DE TABELAS

TABELA 1 – CLASSES E SEUS SIGNIFICADOS	38
TABELA 2 - MENSAGENS DO DIAGRAMA DE SEQÜÊNCIA ARMAZENA FENÔMENOS	42
TABELA 3 - MENSAGENS DO DIAGRAMA DE SEQÜÊNCIA RECUPERA FENÔMENOS.....	43
TABELA 4- ATRIBUTOS ESPECIAIS DO DICIONÁRIO DE DADOS	47
TABELA 5 – DICIONÁRIO DE DADOS.....	48

LISTA DE SIGLAS

API	Application Programming Interface
E/R	Entidade / Relacionamento
GIS	Geographic Information System
GML	Geography Markup Language
JAXB	Java Architecture for XML Binding
LDAP	Lightweight Directory Access Protocol
ODBMG	Object Database Management Group
ODL	Object Definition Language
OGIS	Consortium Open GIS
OID	Object Identifier
OQL	Object Query Language
SGBDOO	Banco de Dados Orientado a Objetos
SGBDR	Banco de Dados Relacional
SQL	Structure Query Language
UML	Unified Modeling Language
UTM	Universal Transverse Mercator
VRML	Virtual Reality Modeling Language
XML	Extensible Markup Language

RESUMO

Essa dissertação apresenta uma interface para a persistência de fenômenos terrestres representados como objetos. Essa interface dá suporte a operações como armazenamento, em banco de dados e arquivos XML, recuperação e pesquisa usando critérios de posicionamento espacial e propriedades descritivas dos fenômenos. Finalmente como resultado apresenta uma aplicação que utiliza a interface de persistência apresentada.

Palavras-chave: Persistência de Objetos; Representação de Fenômenos Terrestres; JAVA; SIG.

1 INTRODUÇÃO

O ponto principal dessa dissertação é a modelagem e construção de uma interface de persistência para fenômenos terrestres representados como objetos. Este trabalho está inserido num contexto de outros dois trabalhos relacionados, buscando uma solução específica para o modelo de representação de fenômenos terrestres como classes e objetos (VIEIRA 2004)(CAMARGO, 2003).

O desenvolvimento do trabalho apresentado está estruturado da seguinte maneira:

No capítulo 2, "Contextualização", é discutido o objeto principal do trabalho, o fenômeno terrestre, uma explicação sobre o ponto de vista cartográfico.

No capítulo 3, "Revisão Bibliográfica", é discutido outro tema principal do trabalho, a persistência de objetos. Os meios de persistência para objetos atuais que mais se destacam são discutidos. Além disso, há uma outra discussão sobre trabalhos relacionados na área.

No capítulo 4, "Modelo dos Fenômenos Terrestres", é exposto um modelo de referência, baseado no modelo Espaço-Temporal (VIEIRA, 2004) para a representação dos fenômenos como objetos. Com o modelo é possível representar os fenômenos terrestres computacionalmente de forma a persistilos.

No capítulo 5, "O Experimento", é demonstrado como foi modelada e construída a interface de persistência de fenômenos terrestres. É exposto um método de abordagem, a maneira pela qual as informações foram armazenadas, e uma explicação de como a interface foi modelada.

No capítulo 6, "Uma Aplicação como estudo de caso", é demonstrado utilização da interface de persistência no armazenamento e recuperação de

fenômenos utilizando para isso um protótipo exemplo.

No capítulo 7, "Uma integração para teste", é demonstrado como pode ser feita a integração da interface de persistência com outro trabalho que utiliza o mesmo modelo de representação dos fenômenos terrestres.

No capítulo 8, "Conclusão, Resultados e Trabalhos Futuros", são discutidos os resultados obtidos e também as perspectivas de trabalhos futuros, aprimoramentos e extensões da interface de persistência.

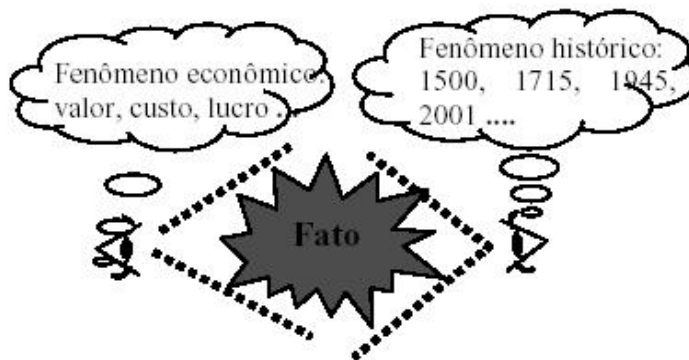
O estudo realizado referente às técnicas de mapeamento de objetos em um banco de dados relacional encontra-se no apêndice I. E o estudo realizado sobre o particionamento do espaço utilizando cartas topográficas encontra-se no apêndice II.

2 CONTEXTUALIZAÇÃO

2.1 FATOS E FENÔMENOS TERRESTRES

Fato é aquilo que existe ou acontece no mundo. Pode ser interpretado de diferentes formas. Por exemplo, observando a trajetória de um país o interesse pode estar voltado aos aspectos econômicos se os agentes observadores forem economistas, e aos aspectos sociais e históricos se os observadores forem sociólogos. A Figura 1 ilustra a relação entre fato e fenômeno.

FIGURA 1 – VISÕES DO MESMO FATO (VIEIRA, 2002)



Assim, fenômenos são interpretações de fatos ou o entendimento que se está tendo com relação aos fatos (VIEIRA, 2001) (VIEIRA et al, 2002).

Quando a Terra é observada na sua parte externa, percebe-se um fenômeno: a superfície terrestre. Coisas acontecem ou existem sobre essa superfície terrestre: São os fenômenos terrestres, e caracterizam-se por estarem situados sobre a superfície terrestre. Uma visão mais detalhada da superfície terrestre permite a observação de oceanos, lagos, rios entre tantos outros caracterizados pela presença da água. Assim, temos a superfície terrestre particionada em dois fenômenos: a superfície topográfica e corpo

hídrico. A Figura 2 apresenta a divisão da superfície terrestre em superfícies topográficas e corpos hídricos.

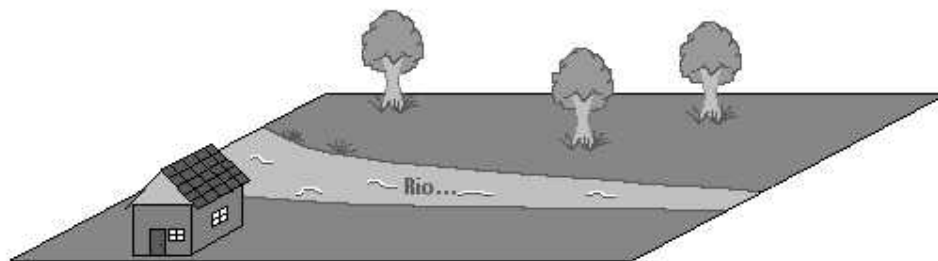
FIGURA 2 – SUPERFÍCIE TERRESTRE, TOPOGRÁFICA E CORPO HÍDRICO (VIEIRA, 2002)



Assim, pode-se considerar que fenômenos terrestres são entendimentos de fatos que ocorrem sobre a superfície terrestre. De forma mais específica, pode-se dizer que fenômenos topográficos são entendimentos de fatos que ocorrem sobre a superfície topográfica (VIEIRA et al, 2002).

Pode-se considerar que as manifestações que ocorrem sobre a superfície terrestre como sendo uma representação de algum tipo de fenômeno terrestre. Desse modo, pode-se notar que sobre a superfície topográfica estão inúmeros fenômenos terrestres. A figura 3, a seguir, ilustra um exemplo de manifestação de fenômenos terrestres na natureza (observe os fenômenos topográficos e corpo hídrico).

FIGURA 3 – OS FENÔMENOS TERRESTRES NA NATUREZA.



A princípio, os fenômenos topográficos foram divididos nas seguintes categorias para facilitar a análise (VIEIRA, 2004):

a) móvel - é toda região delimitada da superfície terrestre, cujo proprietário detém os direitos de usos e frutos sobre ele;

b) construção - pode ser de dois tipos (edificação ou artefato de engenharia):

b.1) edificação - é toda construção feita pelo homem, de caráter permanente, de um tipo de material, com um certo fim (serviço público, comercial, residencial ou comercial-residencial), delimitada por paredes e teto e contida em um imóvel;

b.2) artefato de engenharia - é toda construção feita pelo homem, de caráter permanente, mas não edificação, e é utilizada como apoio a alguma atividade humana ou como complementação de uma edificação;

c) via de acesso - é toda faixa da superfície terrestre a partir da qual se pode deslocar de um extremo para o outro;

d) afloramento rochoso - é todo tipo de rocha que está exposta sobre a superfície terrestre;

e) solo - é o extrato superficial que suporta a cobertura vegetal e as árvores;

f) depósito superficial - é todo tipo de material acumulado na superfície topográfica oriundo de degradação de rochas (exemplo: areia, cascalho etc.);

g) cobertura vegetal - é toda área de vegetação, para a qual se faz uma classificação estruturada. Nesta classificação se leva em conta somente a aparência e a natureza da vegetação predominante dentro da área;

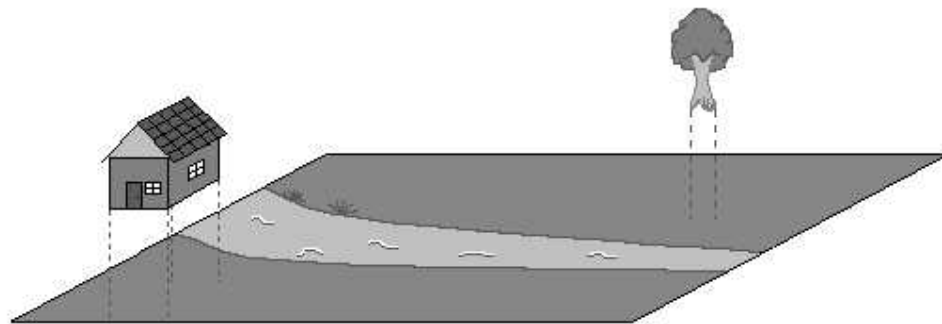
h) árvore - é todo tipo de vegetação que é identificada e levantada isoladamente;

i) topônimo - é o nome próprio que identifica um fenômeno terrestre.

Em alguns casos, há necessidade de trabalhar com fenômenos

terrestres específicos da natureza. Então, para facilitar a análise desses fenômenos eles são observados de forma isolada, independentemente uns dos outros, como objetos georreferenciados. A figura 4 ilustra a observação de fenômenos terrestres observados de forma isolada (retirados da figura 3).

FIGURA 4 – FENOMENOS TERRESTRES DE FORMA ISOLADA



2.2 ÁREA DE LEVANTAMENTO

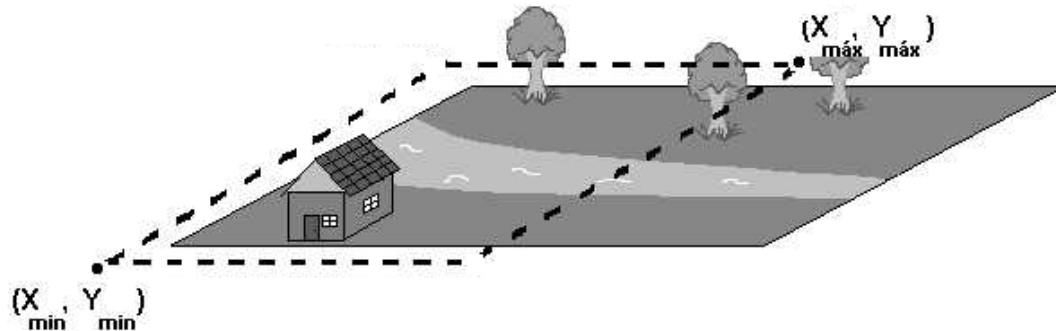
O ato de medir ou observar um conjunto de fenômenos para extrair dados em uma determinada região ou área da superfície terrestre caracteriza uma área de levantamento terrestre. No contexto desse trabalho, todos os fenômenos terrestres necessariamente precisam estar associados a pelo menos uma área de levantamento.

A região da superfície terrestre sobre a qual os dados foram medidos representa a área de levantamento. Assim, a área de levantamento é definida pelos seus pontos extremos (X máximo, Y máximo) e (X mínimo, Y mínimo) baseados em um sistema de coordenadas específico (neste trabalho será utilizado o sistema Geocêntrico) e representa uma seleção ou subconjunto de fenômenos terrestres que foram medidos em determinada época e que estão contidos em determinada região espacial delimitada pelos pontos extremos da área de levantamento.

A figura 5, a seguir, ilustra como uma área de levantamento (representada por uma região tracejada) e seus fenômenos podem ser

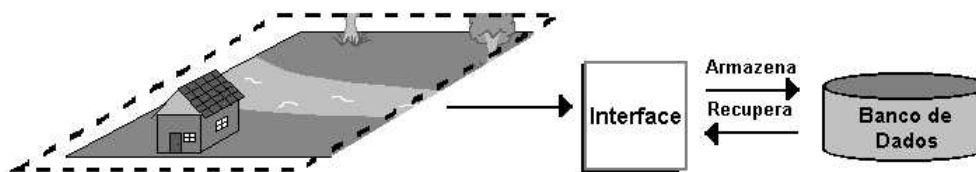
extraídos da natureza utilizando sua posição espacial.

FIGURA 5 – UMA ÁREA DE LEVANTAMENTO EXTRAÍDA DA NATUREZA



Em resumo, as áreas de levantamentos ocorrem em uma região da superfície terrestre, sobre a qual se encontram os fenômenos terrestres. Este trabalho propõe o desenvolvimento de uma interface que permita a persistência de fenômenos terrestres contidos em suas respectivas áreas de levantamento e que ambos estão modelados como objetos. A figura 6 ilustra como deve funcionar a interface de persistência de fenômenos terrestres:

FIGURA 6 – INTERFACE DE PERSISTÊNCIA DE FENÔMENOS TERRESTRES



2.3 ESCOPO DO TRABALHO

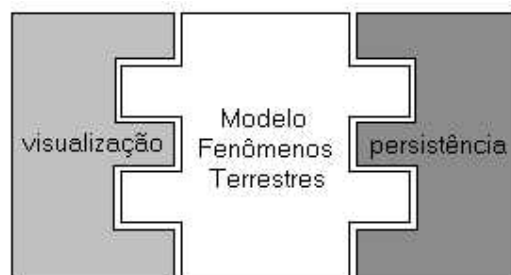
Esta dissertação está contextualizada dentro de um projeto que envolve outros dois trabalhos, como ilustra a figura 7. Os trabalhos são divididos em três modelos:

- Modelo de fenômenos terrestres – desenvolvido como tese de doutorado (VIERA, 2004) discute a modelagem dos fenômenos terrestres e áreas de levantamento

como objetos utilizando a UML em um modelo resultante chamado “Modelo Espaço-Temporal”.

- Modelo de visualização de fenômenos – desenvolvido como dissertação de mestrado (CAMARGO, 2003) consiste na elaboração de um arcabouço de programa para a visualização de fenômenos terrestres na Web por meio de cenários das áreas de levantamentos. Este trabalho utiliza como referência o Modelo Espaço-Temporal (VIEIRA,2004) para representação dos fenômenos.
- Modelo de persistência de fenômenos – é o trabalho do qual trata esta dissertação, consiste na utilização do Modelo Espaço-Temporal (VIEIRA, 2004) para representação dos fenômenos terrestres e construção de uma interface de persistência para persistência, procurando obter o máximo de independência de tecnologia e portabilidade(LIMA, 2003).

FIGURA 7 – ESCOPO DO TRABALHO



2.4 OBJETIVO GERAL

O objetivo deste trabalho é modelar e implementar uma interface para o armazenamento e recuperação de fenômenos terrestres representados como

objetos utilizando como referência o Modelo Espaço-Temporal(VIEIRA, 2004).

2.4.1 Objetivos específicos

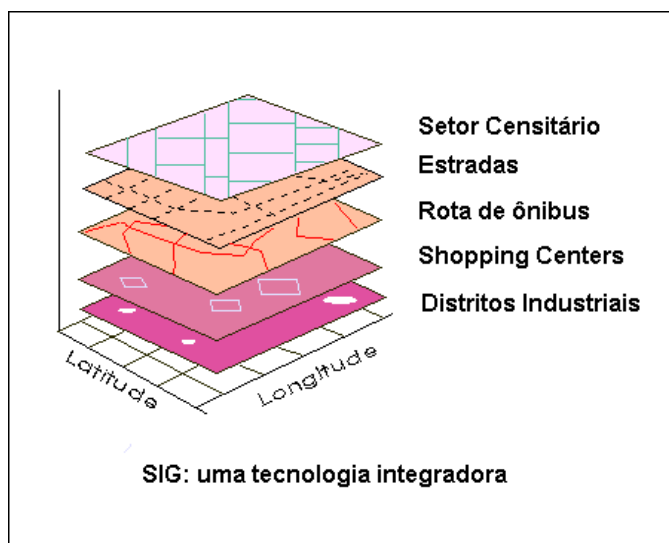
- Investigar e identificar métodos e técnicas para a persistência de objetos em sistemas de informação.
- Determinar o modelo de representação e métodos para a persistência de fenômenos terrestres.
- Implementar a Interface de Persistência.
- Desenvolver um protótipo modelo, estudo de caso, que demonstre a utilização da interface de persistência.

3 REVISÃO BIBLIOGRÁFICA

3.1 TRABALHOS RELACIONADOS

Considere o diagrama esquemático da figura 8. Imagine que o SIG nos permita organizar a informação sobre uma determinada região como um conjunto de mapas, cada um deles exibindo uma informação a respeito de uma característica da região. Cada um destes mapas temáticos individualmente é referenciado como uma camada, cobertura ou nível. Cada camada foi cuidadosamente sobreposta de forma que toda localização é precisamente ajustada às localizações correspondentes em todos os outros mapas. A camada de baixo deste diagrama é a mais importante, porque representa um reticulado com um sistema de referência (como latitude e longitude) ao qual todos os mapas foram precisamente referenciados.

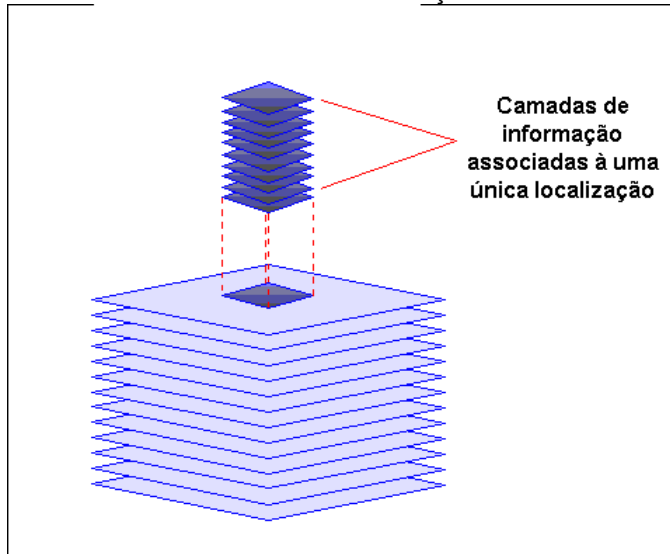
FIGURA 8 – CAMADAS DE INFORMAÇÃO EM SIG



Cada um destes mapas foram referenciados dentro de um mesmo

sistema locacional de referência para que informações exibidas nas diferentes camadas podem ser comparadas e analisadas em combinação, como mostra figura 9.

FIGURA 9 – CAMADAS DE INFORMAÇÃO ASSOCIADAS À UMA ÚNICA INFORMAÇÃO



Este trabalho tem o propósito de prover uma arquitetura independente de tecnologia, e que permita trabalhar com fenômenos modelados como objetos, dentre os sistemas de informação geográficos pesquisados ambos trabalho com o conceito trabalhar com fenômenos, também conhecidos como objetos geográficos na literatura, como mapas temáticos. Alguns exemplos deste tipo de abordagem são as ferramentas: TerraLib(TERRALIB,2004), GeoFrame(ROCHA, 2001) entre outras.

A forma de fazer a persistência é dependente da modelagem das informações, que neste caso, está fortemente enfocada na modelagem orientada a objetos, tema da modelagem espaço temporal o qual este trabalho é dependente. O que enfatiza a necessidade de uma solução específica para este modelo, dentre suas necessidades particulares. O que descartou a possibilidade de utilizar alguma solução pronta de persistência de fenômenos terrestres utilizada por outro SIG, já que esse não utiliza o Modelo Espaço-

Temporal (VIEIRA, 2004).

3.2 PERSISTÊNCIA DE OBJETOS

A persistência dos objetos pode ser vista como a capacidade de utilização do objeto fora dos limites da aplicação que o criou, ou seja, é a propriedade em que um objeto continua a existir, mesmo quando a aplicação que o criou terminar a execução. (Hill, 1996). Em geral a persistência é obtida gravando-se o objeto em um meio de armazenamento não volátil.

3.3 FORMAS DE PERSISTÊNCIA

Dentre as técnicas e ferramentas existentes para realizar a persistência de objetos em sistemas de informação, destacaram-se as seguintes:

- Banco de dados orientado a objetos
- Banco de dados objeto-relacional
- Serialização
- Banco de dados relacional

3.3.1 Banco de Dados Orientado a Objetos

Os Sistemas de Gerenciamento de Bancos de Dados Orientados a Objetos (SGBDOO) tiveram início nos anos 80 quando a tecnologia de objetos estava surgindo. Atualmente encontra-se em desenvolvimento a padronização pela *Object Database Management Group* (ODBMG,2003). Enquanto isso o ODMG produziu um padrão para uma linguagem de consultas para SGBDOO, sob o nome OQL. As principais linguagens em processo de padronização para SGBDOO são:

- ODL - *Object Definition Language*.

- OQL - *Object Query Language*, declarativa, não procedural, para consulta e atualização do conteúdo da base de dados baseada no SQL.

Em um SGBDOO, a noção de objeto é usada no nível lógico e possui características não encontradas nas linguagens de programação tradicionais, como operadores de manipulação de estruturas, gerenciamento de armazenamento, tratamento de integridade e persistência dos dados. Atualmente, os SGBDOO não são muito utilizados pelas organizações devido a questões principalmente de desempenho e portabilidade das informações entre diferentes SGBDOO (REIS, 2002) além de atualmente ter linguagens de consulta pouco padronizadas (CELKO, 1997).

3.3.2 Banco de Dados Objeto-Relacional

O paradigma objeto-relacional surgiu com a proposta de suprir carências do modelo relacional para armazenamento de dados complexos em tabelas, por exemplo, uma hierarquia de classes. Esses bancos de dados oferecem interfaces tanto para as aplicações que foram desenvolvidas no paradigma relacional quanto às orientadas a objetos, dispensando assim a necessidade de dois bancos de dados, um para as aplicações relacionais e outro para aplicações orientadas a objetos (STONEBRAKER et al, 1996). Um banco de dados deve prover algumas características básicas para ser considerado um SGDB objeto-relacional (ORACLE, 2000):

a) Estender os tipos básicos – pode-se criar um novo tipo de dados para facilitar a modelagem e organização das informações. Faz-se isso definindo o nome do tipo, as informações sobre a forma de armazenamento do tipo, e as rotinas que convertem o novo tipo em ASCII e de ASCII para o novo tipo. Um exemplo disso é a sentença:

```
Create type NOVO_TIPO (
```

```
Internallegth = 8,
Input = FUNCAO_DE_ENTRADA
Output = FUNCAO_DE_SAIDA);
```

b) Implementar objetos complexos – entende-se por objetos complexos aqueles compostos de múltiplos objetos básicos ou definidos pelo usuário. Um tipo, que é parte de um tipo composto, também pode ser um tipo composto. Um exemplo de um tipo composto é mostrado a seguir:

```
Create type TIPO_COMPOSTO(
        NOME          varchar(10),
        OBJETO    NOVO_TIPO);
Create table TABELA of TIPO_COMPOSTO
```

Neste exemplo o nome do novo tipo composto é TIPO_COMPOSTO, que foi definido como uma composição de um tipo básico NOME, que é um “varchar” , e de um tipo OBJETO que foi definido no exemplo anterior. A segunda sentença cria uma tabela, cujo nome é TABELA, que aceitará instâncias do novo tipo recém criado.

c) Permitir herança de dados e funções – são características semelhantes com as linguagens atuais de programação orientadas a objeto. Um exemplo de herança é mostrado a seguir, baseado no item anterior :

```
Create type TIPO_COMPOSTO_HERANCA(
        NOME          varchar(10)) UNDER TIPO_COMPOSTO;
```

O banco de dados objeto-relacional embora ofereça algumas facilidades para a representação de objetos atualmente ,na maioria dos casos, ainda tem um suporte ao modelo de objetos restrito faltando por exemplo suporte a herança múltipla e/ou interfaces (BAIAO, 2003).

A persistência utilizando Spatial Types utilizam uma modelagem de objetos geográficos que não é focado em fenômenos terrestres, ou seja, não

foram construídas baseadas no Modelo Espaço Temporal que é a referência deste trabalho.

3.3.3 Serialização

A serialização de objeto significa simplesmente gravar os valores de todos os seus campos de dados como seqüência de bytes. Representando o estado do objeto no momento em que foi persistido (SUN, 2002).

Para a serialização de um objeto em Java, por exemplo, é necessário definir a classe do objeto como uma classe que pode ser serializada e utilizar as classes *ObjectOutputStream* ou *ObjectInputStream* para manipular os objetos. O exemplo a seguir (Quadro 1) ilustra o processo de serialização de um objeto:

QUADRO 1 – PROCESSO DE SERIALIZAÇÃO

Ex. código

```
public class testeSer implements Serializable
...
testeSer obj = new testeSer();
ObjectOutputStream ou = new ObjectOutputStream(new
FileOutputStream("arquivo.ser"));
ou.writeObject(obj);
ObjectInputStream in = new ObjectInputStream(new
FileInputStream("arquivo.ser"));
obj = (testeSer)in.readObject();
```

Como se pode observar no exemplo anterior, o processo de serialização é de fácil utilização além de permitir a persistência de um objeto ou de um conjunto deles de forma transparente ao programador. O inconveniente é que normalmente a persistência do objeto serializado ocorre em um arquivo seqüencial o que geralmente prejudica o desempenho das aplicações.

3.3.4 Banco de Dados Relacional

Para permitir um processo de persistência de sistemas baseados em objetos em bases de dados relacionais, foram propostas diversas idéias que convergiram para um conjunto de padrões de mapeamento de classes em tabelas, alguns destes padrões são citados a seguir:

- *Simple Table Agregation, One Class one Table, Objects in BLOBs, etc* (KELLER W.,1997).
- *Persistent Object, OID, Cursor, etc* (AMBLER, 2000)
- Divididos em 3 tipos Comportamentais, Estruturais e Metadados: *Identify Map, Query Object, Dependent Mapping, Serialized Lob, Lazy Load, etc* (FOWLER et al, 2002).

Além do mapeamento dos objetos em tabelas, são necessárias classes responsáveis pelo acesso aos dados. Para diminuir o acoplamento, relacionamento direto entre o acesso ao banco de dados e o restante da aplicação, surge a opção de construir uma camada de acesso aos dados separando as classes que manipulam os dados das outras classes que armazenam e restauram os dados de forma que as alterações no modelo de dados requeiram modificações apenas nas classes de acesso aos dados, restringindo o impacto de mudanças(AMBLER,2000).

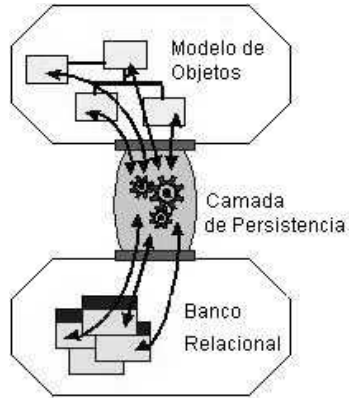
A principal função de uma camada de acesso aos dados é garantir a independência entre o modelo de objetos e o esquema de dados do banco, permitindo que detalhes do esquema de dados sejam substituídos diminuindo o esforço de manutenção.

Em resumo, para representar objetos em um modelo relacional é necessário fazer o mapeamento dos objetos em tabelas, ou seja, transformar um diagrama de classes em um modelo entidade-relacionamento. Após isto, constrói-se um conjunto de objetos responsável pela ligação entre objetos do domínio do problema e o modelo relacional que os representa, este conjunto representa a camada de persistência (ou camada de acesso aos dados).

A figura 10 a seguir ilustra o relacionamento entre os elementos de

uma camada de persistência.

FIGURA 10 – ESQUEMA PARA PERSISTÊNCIA



3.3.4.1 Camada de Persistência

Uma camada de persistência de objetos pode ser descrita como um conjunto de classes que permite a realização do processo de armazenamento e manutenção do estado de objetos em algum meio não-volátil, como um banco de dados, de forma transparente (AMBLER, 2002).

3.3.4.2 Mapeamento objeto-relacional

É um processo de transformação entre o objeto e a modelagem relacional e entre os sistemas que suportam esta característica (KELLER, 1998). O processo de mapeamento requer conhecimento da modelagem de objetos e modelagem relacional, no que eles se parecem e no que diferem.

Os modelos orientados a objetos e relacional utilizam diferentes paradigmas: o modelo relacional que é baseado em tabelas e o modelo orientado em objetos que é baseado em classes. Quando os objetos precisam ser armazenados em bancos de dados relacionais precisa ser preenchida a

lacuna existente entre estes paradigmas. Segundo Fussel (1997), alguns conceitos precisam ser mapeados para o paradigma relacional:

- Agregação
- Herança e polimorfismo
- Associações entre as classes

O mapeamento do objeto em tabelas pode ser feito de várias maneiras, mas que podem ser agrupadas em algumas formas básicas (RATIONAL, 2002). O apêndice II aborda um tutorial mais detalhado sobre este assunto:

- Uma classe mapeada para uma ou mais tabelas: uma simples classe pode ser mapeada em muitas tabelas. A camada de persistência deve compor o objeto construindo uma junção das tabelas. Este modo de mapeamento tem implicações no desempenho e também em atualizações do modelo.
- Uma tabela mapeia uma ou mais classes: esta abordagem melhora a performance, mas viola os princípios da normalização de tabelas. Os atributos de uma classe podem ser mapeados em uma ou mais colunas de uma tabela. Às vezes, combinações de colunas são usadas para calcular o valor de um atributo.
- Os relacionamentos entre os objetos são implementados por meio de referência entre os objetos e as tabelas relacionadas por chaves estrangeiras.

3.3.4.3 Identificadores de Objetos

Identificadores de Objetos, *Object Ids* (OIDs), são identificadores únicos de elementos no esquema de dados. Eles funcionam de forma similar à chave primária em banco de dados relacionais, mas em sistemas orientados a

objetos funcionando mantendo relacionamentos entre os objetos e garantindo a unicidade dos objetos no esquema (AMBLER, 2000).

Os OIDs são importantes no processo de mapeamento objeto-relacional pois são os responsáveis pela criação das chaves primárias (identificadores) no modelo relacional para representação dos objetos mapeados. Alguns mecanismos mais conhecidos utilizados para a geração de OIDs são (AMBLER, 2002):

- Utilização do operador Max da linguagem SQL, em uma tabela para obter o maior valor utilizado como OID e utilizá-lo, acrescido de uma constante, como a próxima chave para um objeto armazenado. Esta técnica garante unicidade apenas em nível de tabela.
- Adoção de uma tabela de valores-chave, na qual o sistema mantém o último valor utilizado como chave por qualquer elemento no banco de dados. Sempre que um novo elemento é inserido, este valor deve ser acrescido de um. Esta técnica garante unicidade em nível de banco de dados.
- Utilização do algoritmo “High/Low” - um tipo de variante do modelo de tabelas de valores-chave. Neste caso, o sistema acessa a tabela que armazena valores-chave apenas uma vez, mantém o valor encontrado (*high*) em memória e forma identificadores para objetos juntamente com um contador auxiliar (*low*) gerado em memória – os dois valores são concatenados na hora de fornecer à aplicação um valor de chave único, para inclusão no banco de dados. Esta técnica também garante unicidade ao nível de banco de dados, sendo que o overhead decorrente da geração de chaves é o menor dentre as três técnicas, já que o banco de dados é menos acessado para gerar valores de chave.

Além dos mecanismos genéricos mostrados anteriormente, também são comuns implementações que permitem ao desenvolvedor utilizar-se de mecanismos de geração de chaves proprietários do banco de dados em uso – a técnica conhecida como GUID, por exemplo, é um mecanismo de geração de OID embutidos em bancos de dados de fabricantes específicos, como por exemplo, o *Microsoft SQL Server* que tenta gerar identificadores únicos por meio da concatenação de informações como o endereço físico da placa de rede, hora atual, IP da máquina e um número aleatório.

3.3.4.4 Um Exemplo de Mapeamento Objeto-Relacional

Nesta seção será exposto um padrão utilizado no mapeamento de objetos para bancos de dados relacionais baseado em Keller, (1997):

FIGURA 11 – ESTRURA DE CLASSES

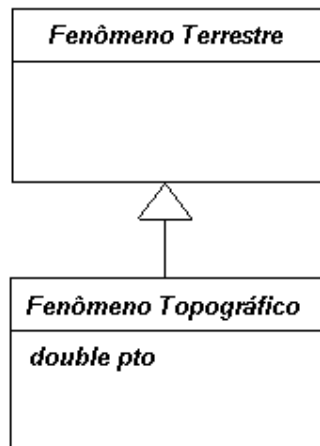
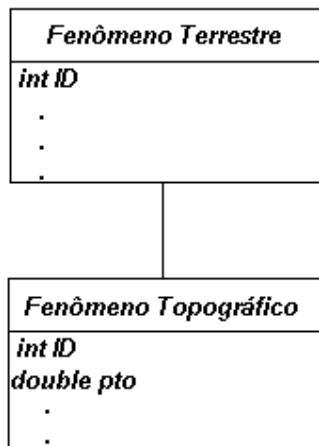


FIGURA 12- ESTRUTURA DE TABELAS PARA O MAPEAMENTO



A figura 11 mostra classes que possuem um relacionamento do tipo herança e a figura 12 um modo de mapeamento dessas classes em um modelo relacional. Note a necessidade do campo ID em ambas as tabelas que servirá como um identificador para os objetos mapeados nessas tabelas.

3.3.4.5 Automação do processo de mapeamento

Atualmente existem ferramentas que auxiliam o processo de mapeamento de objetos em tabelas e vice-versa. Essas ferramentas podem ser de uso comercial ou gratuito. Algumas destas ferramentas podem ser encontradas em (SERVICE, 2004).

As funcionalidades básicas destas ferramentas incluem o mapeamento de tabelas em um diagrama de classes ou ao contrário. Algumas funcionam também como um *framework* de persistência, como por exemplo o Hibernate (HIBERNATE, 2004). Geralmente esses *frameworks* gravam a correspondência entre as classes e as tabelas em um arquivo utilizando uma linguagem descritiva, como XML. É utilizando este arquivo que o *framework* consegue persistir os objetos nas tabelas correspondentes. O quadro a seguir ilustra um trecho de código do mapeamento de uma tabela chamada *Pontos* para uma classe *Ponto* na ferramenta Hibernate:

QUADRO 2- AUTOMAÇÃO DO PROCESSO DE MAPEAMENTO UTILIZANDO UMA FERRAMENTA

```
<class name="Ponto" table="pontos">
  <id name="pontoid" type="long" column="pontoid">
    <generator class="assigned" />
  </id>
  <property
    name="x"
    type="long"
    column="x"
    length="10" />
</class>
```

4 MODELO DOS FENÔMENOS TERRESTRES

O modelo de fenômenos terrestres é um modelo conceitual formal para descrição de fenômenos terrestres baseado em VIEIRA (2004). A descrição do modelo é feita pelo uso do conceito de classes de objetos. Neste modelo, um fenômeno terrestre é descrito por dois grupos de atributos. Um grupo diz respeito à semântica do fenômeno e o outro ao espaço ocupado pelo fenômeno e sua superfície.

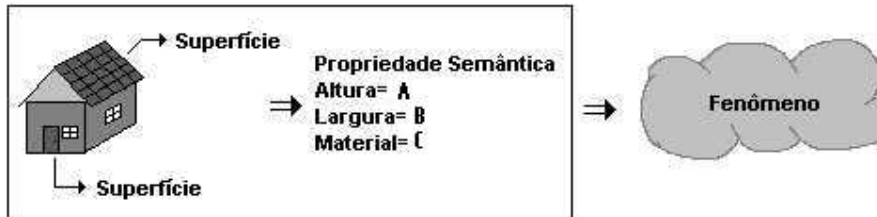
A semântica caracteriza o tipo de fenômeno terrestre dentro da categoria que ele pertence, por exemplo: árvore, edificação, solo, etc. A cada tipo está associado um conjunto de atributos que caracterizam os objetos daquele tipo. Por exemplo, os objetos do tipo edificação podem ter como atributos: registro e proprietário.

O espaço ocupado por um fenômeno terrestre é o espaço ocupado pela sua superfície e esta é composta por um conjunto de faces planas conexas. Isto se verifica quando se deseja descrever um fenômeno terrestre em sua forma tridimensional (3D). Entretanto, existem situações em que se pode admitir que um fenômeno terrestre tem uma dimensionalidade, 0D (ponto), 1D (linha) ou 2D (plano). Por exemplo, é possível se adotar uma dimensionalidade 1D para representar uma rede de drenagem. Entretanto, em uma outra interpretação, pode ser necessário ter, além das conexões e caminhos, as margens dos rios representadas em verdadeira grandeza, na dimensionalidade 2D ou mesmo em 3D. Uma dimensionalidade 0D pode ser usada para representar uma árvore ou um poste. Neste caso, com apenas um ponto geográfico e os atributos semânticos, podem caracterizar o espaço e o tempo.

Em resumo, a caracterização de um fenômeno terrestre consiste em

determinar a sua superfície e suas propriedades semânticas. A figura 13, a seguir, ilustra a caracterização dos fenômenos terrestres extraídos da natureza para o modelo de fenômenos terrestres.

FIGURA 13 – CARACTERIZAÇÃO DE FENÔMENOS TERRESTRES



4.1 CARACTERIZAÇÃO DOS FENÔMENOS COMO OBJETOS

4.1.1 Modelo de Representação dos Fenômenos como Objetos

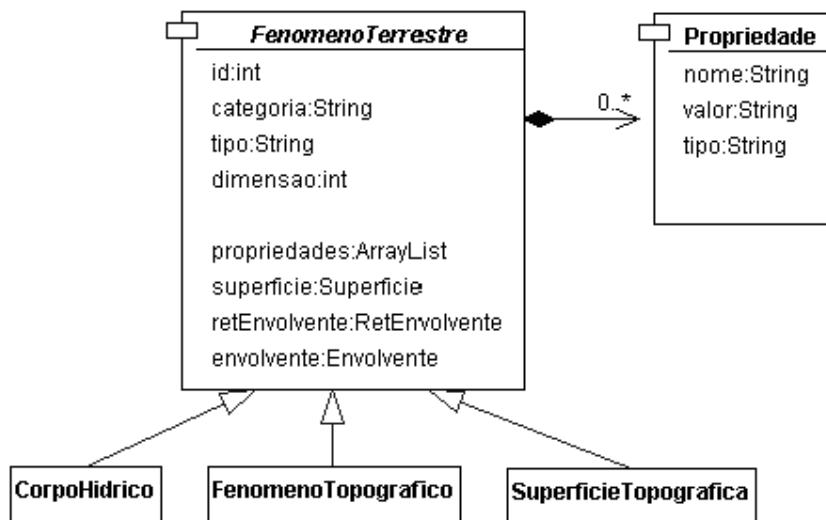
Os fenômenos terrestres podem ser definidos como sendo de 3 tipos básicos. O primeiro deles é a superfície terrestre que representa a parte externa do planeta. Sobre a superfície terrestre existem fenômenos que se caracterizam por não possuir forma definida e em que a superfície terrestre fica submersa, esses são os corpos hídricos. Há também as regiões da superfície terrestre que se caracterizam por não estarem submersas chamadas superfície topográficas. O último tipo básico é fenômeno topográfico que abrange o restante dos fenômenos que estão sobre a superfície topográfica, por exemplo, árvores, edificações, pavimentações etc.

Essa visão da realidade sempre acontece segundo uma concepção ou um histórico da formação do observador ou conforme um modelo que existe na mente do observador. Assim, para se descrever um fenômeno, ou seja, o entendimento que se tem de um fato que ocorre sobre a superfície terrestre, adota-se um modelo e um conjunto de propriedades e atributos. Para esse conjunto de propriedades e atributos são feitas observações ou atribuições de

valores que caracterizam ou interpretam o fato corretamente, no espaço e no tempo.

O modelo de representação de fenômenos como objetos, chamado de *Modelo de Referência*, é uma formalização da representação dos fenômenos terrestres que foi adotada utilizando a UML e baseia-se em (VIEIRA, 2004). Esse modelo descreve como os fenômenos terrestres se comportam. A figura 14, a seguir ilustra o modelo de referência:

FIGURA 14 – MODELO DE REFERÊNCIA



Observando o modelo, nota-se que a super classe mais geral *FenômenoTerrestre*, que representa os fenômenos terrestres, descreve o fenômeno baseado nas seguintes propriedades:

- **Atributos:** representa um conjunto de atributos que um fenômeno pode possuir para descrever suas características.
- **Dimensão:** indica a dimensionalidade da forma espacial do fenômeno, isto é, se a sua forma geométrica é composta de 0 (ponto), 1 (linha) , 2 (face) ou 3 dimensões (múltiplas faces). Essa propriedade representa a premissa básica do modelo de referência que todo fenômeno se apresenta na forma de

superfícies planas.

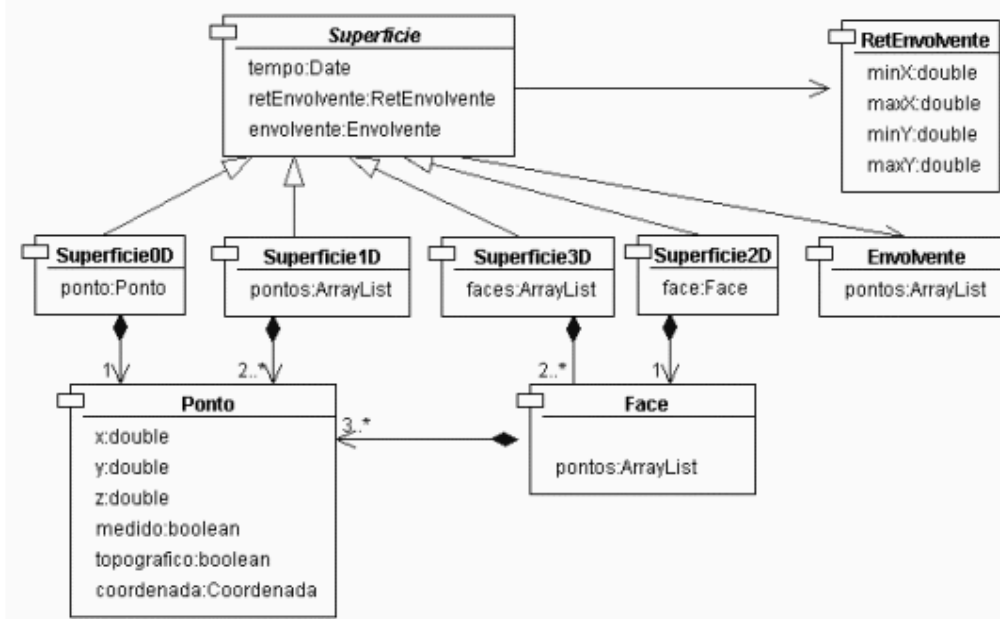
- Superfície: é a descrição geométrica da forma espacial do fenômeno associada a um instante de observação que a define no tempo.
- Tipo: define a semântica que o fenômeno assume: rua, imóvel, árvore, etc.

O modelo de referência divide os fenômenos terrestres em 3 categorias:

- CorpoHídrico – representa os fenômenos terrestres de forma líquida. Por exemplo: oceanos, lagos, rios etc.
- SuperfícieTopografica – representa a parte da superfície terrestre que não está coberta por um corpo hídrico.
- FenomenoTopografico – representa os fenômenos terrestres restantes que se caracterizam por estar situados sobre a superfície topográfica.

O Modelo de Referência baseia-se na construção de fenômenos a partir de suas superfícies. A superfície de um fenômeno terrestre é representada como um objeto complexo com várias classes e relacionamentos que definem as possíveis superfícies de um fenômeno. A figura 15, a seguir, ilustra como a superfície é representada no modelo de referência:

FIGURA 15 - SUPERFÍCIE DO FENÔMENO TERRESTRE



As dimensionalidades estão representadas pelas classes Superficie0D, Superficie1D, Superficie2D e Superficie3D. A classe Superficie0D representa as superfícies pontuais definidas por uma instância da classe Ponto. A classe Superficie1D representa as superfícies lineares definidas por uma seqüência de duas ou mais instâncias da classe Ponto. A classe Superficie2D representa as superfícies planas definidas por uma instância da classe Face. A classe Superficie3D representa as superfícies multifacetadas definidas por um conjunto de duas ou mais instâncias da classe Face. A classe Face representa uma região plana definida por uma seqüência de instâncias da classe Ponto as quais determinam os vértices da face.

A área de levantamento que é a região onde se encontram os fenômenos, representando o levantamento de campo, é representada pela classe mostrada na figura 16.

FIGURA 16 – CLASSE ÁREA DE LEVANTAMENTO

Area Levantamento
-tempo:Date
-nome:String
-retEnvolvente:RetEnvolvente
-fenomenos>List
-escala:double
+getAreas(x1:int,y1:int,x2:int,y2:double):AreaLevantamento[]

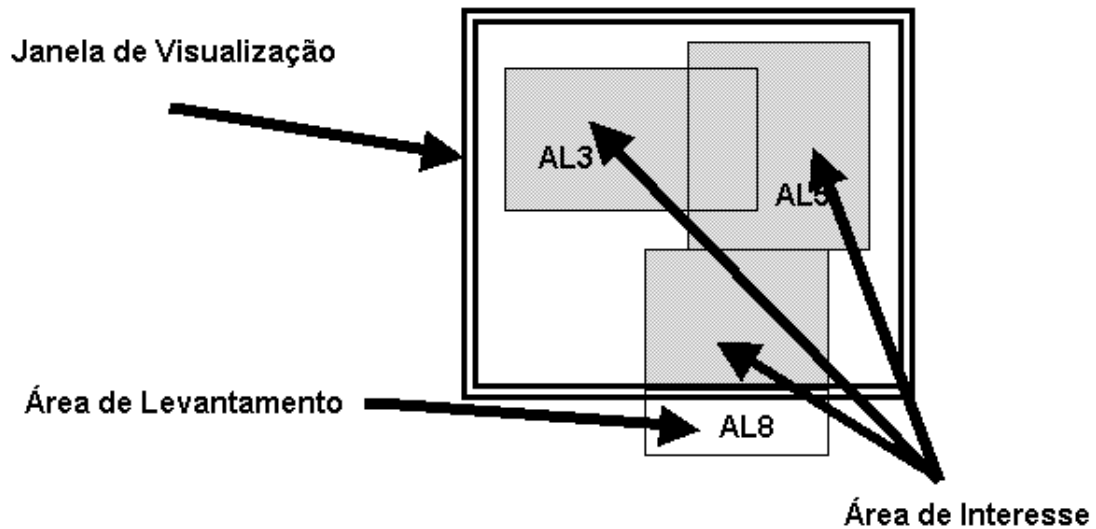
Criou-se uma classe que representa uma região delimitada por 2 pontos na qual o usuário pode visualizar os fenômenos que estejam contidos nessa região facilitando a utilização da interface de persistência de fenômenos (LIMA, 2003) por uma aplicação de visualização de fenômenos terrestres (CAMARGO, 2003) . A este objeto que representaria uma região espacial para a visualização dos fenômenos deu-se o nome de *janela de visualização*.

A janela de visualização é uma região retangular delimitada por coordenadas geográficas em seus pontos extremos. Definindo assim, uma região da superfície topográfica na qual estão os fenômenos terrestres de interesse do usuário.

A interface de persistência considera que qualquer fenômeno está vinculado com uma área de levantamento. No momento da recuperação dos fenômenos, para compor a janela de visualização, a área de levantamento restaurada é transformada em outro elemento chamada *área de Interesse*.

A Área de Interesse representa um subconjunto de fenômenos da Área de Levantamento, os fenômenos que compõem este subconjunto são escolhidos baseados em sua posição geográfica com relação a uma janela de visualização. A principal diferença entre a área de levantamento e a área de interesse é que essa última possui seus limites derivados da intersecção dos limites da área de levantamento original com a janela de visualização utilizada. A figura 17, a seguir ilustra a relação entre janela de visualização, área de interesse e área de levantamento.

FIGURA 17 – JANELA DE VISUALIZAÇÃO, ÁREA DE INTERESSE E ÁREA DE LEVANTAMENTO



A Janela de Visualização é uma região retangular definida pelo usuário por meio das coordenadas agrupando todos os fenômenos contidos nela. A Janela de Visualização é representada pela classe a seguir, figura 18:

FIGURA 18 – CLASSE JANELA DE VISUALIZAÇÃO

Janela Visualizacao
-fenomenos:List
-retEnvolvente:RetEnvolvente
+cria(x1:int,x2:int,y1:int,y2:double):v
+adiciona(a:AreaInteresse):void

A Área de Interesse é um conjunto de fenômenos terrestres que pertencem às áreas de levantamento atingidas pela janela de visualização. A Área de Interesse é representada pela classe a seguir, figura 19:

FIGURA 19 – CLASSE ÁREA DE INTERESSE

Area Interesse
-fenomenos:List
-retEnvolvente:RetEnvolvente
+cria(x1:int,y1:int,x2:int,y2:double):v

5 O EXPERIMENTO

Na realização do experimento, implementação da Interface de Persistência de Fenômenos Terrestres, as seguintes atividades precisaram ser feitas:

1. Desenvolvimento de um método de abordagem que determine como armazenar os fenômenos terrestres
2. Desenvolvimento de um método que permita a estruturação da base de dados facilitando a localização dos fenômenos terrestres
3. Construção do diagrama de classes definindo um conjunto de classes e o papel que cada classe terá que realizar na interface de persistência.
4. Construção de diagramas de seqüência que possibilitassem identificar os fluxos de mensagens que foram trocadas entre as classes para organizar a estrutura da interface.
5. Construção do Diagrama de Entidade-Relacionamento que represente como está organizado o banco de dados da aplicação, identificado cada entidade e seus atributos.
6. Elaboração do Dicionário de Dados para documentar as características e o significado de cada tabela e coluna presente no banco de dados.

5.1 FERRAMENTAS UTILIZADAS

Os recursos utilizados para a execução deste projeto são:

1 Computador com:

1. Ferramenta de Modelagem UML – Together.

2. Java Standard Developer Kit 1.4.0 ou superior;
3. Java Web Service Developer Kit Patch 1.1 ou superior;
4. Biblioteca GeoVRML, utilizada para conversão de coordenadas geográficas.
5. Banco de dados Microsoft SQL-Server 7 ou superior
6. Conjunto de dados representados por áreas de levantamento e fenômenos terrestres em formato Maxicad.

Os softwares e equipamentos utilizados são os que estão disponíveis no laboratório do Curso de Pós-Graduação em Geociências do Departamento de Geomática da Universidade Federal do Paraná.

5.2 MÉTODO DE ABORDAGEM

O método de abordagem do experimento consiste em definir de que forma os propósitos da interface de persistência podem ser implementados. Estes propósitos da interface de persistência de fenômenos terrestres podem ser demonstrados em duas funcionalidades básicas:

- Um método de fácil acesso para armazenamento e busca de fenômenos terrestres.
- Facilitar a localização de fenômenos geograficamente utilizando coordenadas.

5.2.1 Forma de persistência

Para a implementação da interface de persistência de fenômenos terrestres será utilizado um banco de dados relacional. Os principais fatores que motivaram a escolha deste modelo de persistência foram:

- O modelo de banco de dados disponível como recurso para o desenvolvimento do projeto;
- O modelo relacional ter maior flexibilidade para migração entre

os diversos SGBDR.

- Consolidação de padrões, o que possibilita a construção da Interface de persistência de forma genérica para qualquer sistema gerenciador de banco de dados relacional.

5.2.2 Método para Persistência

Para o desenvolvimento do experimento foi necessário extrair um conjunto de características de cada objeto identificado, no caso os fenômenos terrestres. É descrita como *superfície* a característica que possibilita visualizar o objeto e *semânticas* as características descritivas, codificadas alfanumericamente, que contêm informações adicionais sobre o objeto.

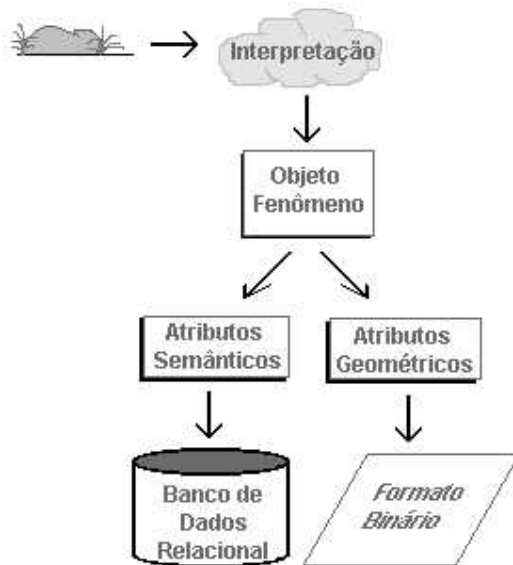
A estratégia para persistência dos fenômenos adotada foi separar as características da superfície e as semânticas em estruturas separadas. As principais razões para essa separação estão relacionadas com o desempenho da aplicação e para a complexidade da modelagem.

A *superfície* do fenômeno, isto é, as características geométricas são armazenadas em uma estrutura de dados proprietária melhorando o desempenho para a recuperação da informação geométrica, já que não são necessários múltiplos acessos ao banco de dados e diminuindo o número de tabelas necessárias na modelagem da interface de persistência.

As características *semânticas* do fenômeno são incluídas em tabelas do SGBDR, já que as características semânticas precisam de uma busca mais rápida e não demandam tantos recursos computacionais para serem recuperadas. A figura 20, a seguir, demonstra o esquema utilizado para o experimento do método de persistência.

As características geométricas xxxx.

FIGURA 20 – MÉTODO DE PERSISTÊNCIA

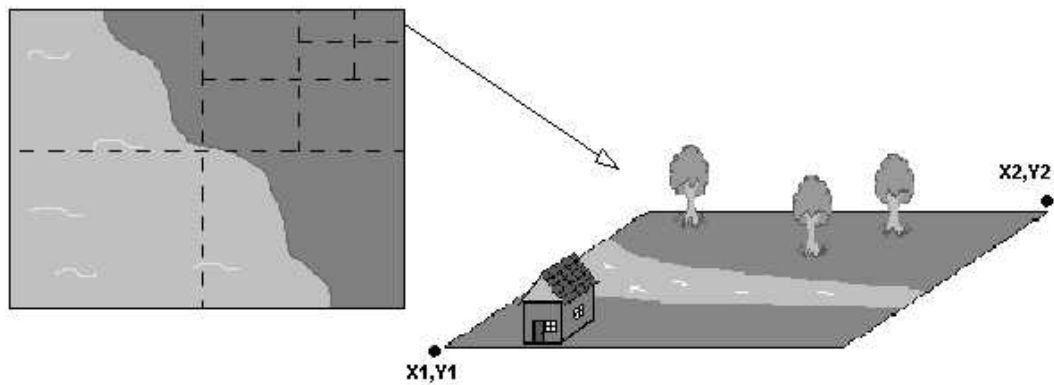


5.2.3 Método de Localização De Fenômenos

A alternativa encontrada para localização de objetos georreferenciados utiliza-se do particionamento do espaço por meio de cartas topográficas. Em algumas referências bibliográficas, as cartas topográficas também são chamadas de mapas.

O processo de particionamento do espaço utilizando cartas topográficas baseia-se na divisão de uma carta topográfica, gerando uma nova carta em maior escala, essa divisão pode ocorrer sucessivas vezes até se encontrar a carta mais apropriada para conter a área de levantamento e seus fenômenos. Este processo ajuda a encontrar áreas de levantamento próximas, pois elas estão na mesma carta topográfica. O apêndice I explica com maiores detalhes o método de particionamento das cartas utilizado. A figura 21, a seguir, ilustra como funciona em linhas gerais a articulação de uma carta topográfica para encontrar uma área de levantamento.

FIGURA 21 – MÉTODO DE LOCALIZAÇÃO



5.3 MODELAGEM DA INTERFACE DE PERSISTÊNCIA EM UML

A Interface de persistência de fenômenos terrestres foi modelada utilizando-se a UML. A seguir estão descritos os seguintes diagramas gerados na modelagem da Interface de persistência :

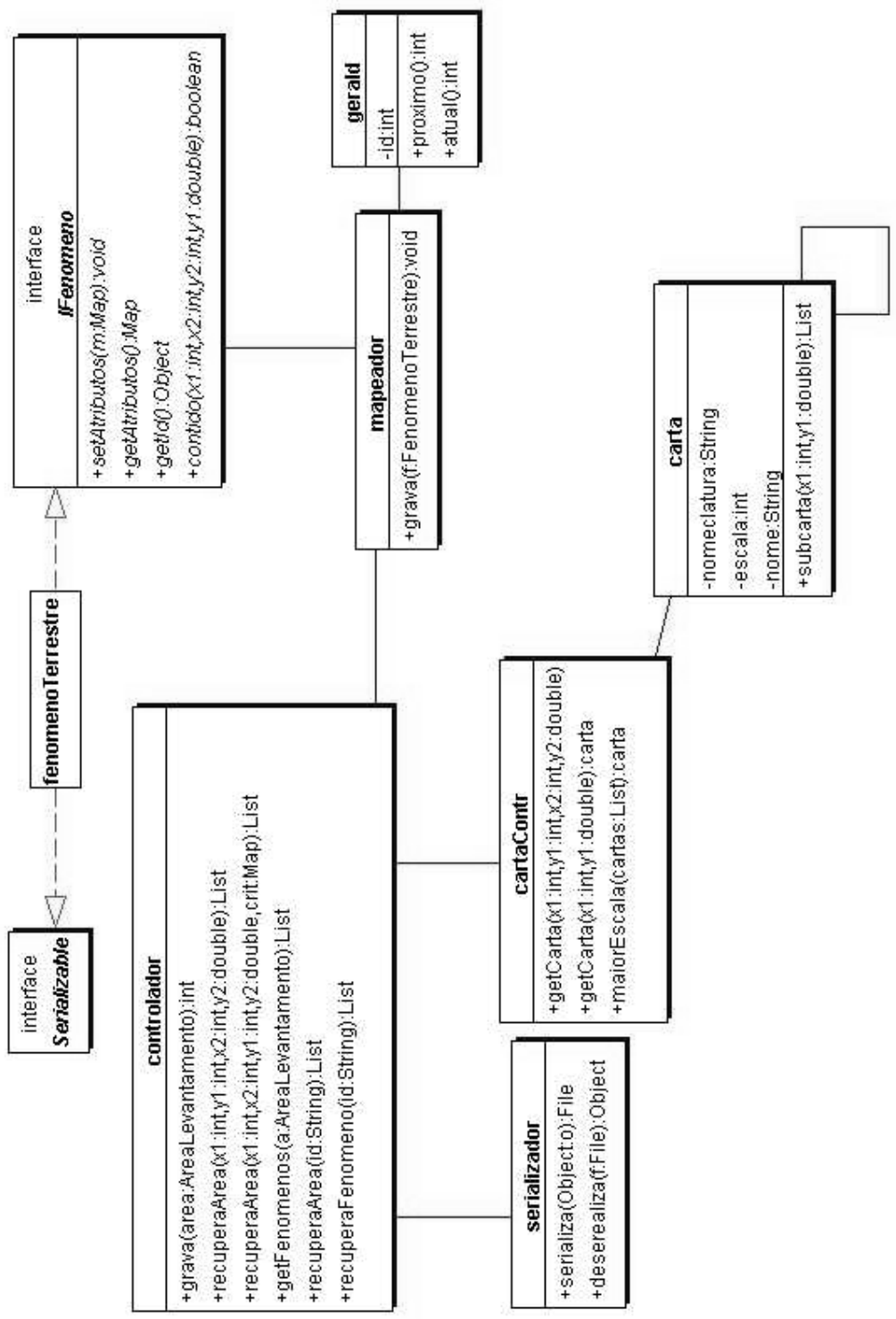
- Diagrama de Classes – demonstra o relacionamento estático e a estrutura dos objetos da Interface de Persistência, assim como a ligação com o Modelo de Referência.
- Diagrama de Seqüência – demonstra a interação entre os objetos para realizar as operações de armazenamento e recuperação das áreas de levantamento e fenômenos terrestres.

5.3.1 Diagrama De Classes

Um diagrama de classes é um diagrama que mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Ele é usado para fazer a modelagem estática do sistema, visualização, especificação e documentação de modelos estruturais. (BOOCH, 1999)

A figura 22, a seguir, ilustra o diagrama de classes da interface de persistência de fenômenos terrestres, esse diagrama de classes será aplicado sobre o Modelo de Referência representado no diagrama pela classe *FenomenoTerrestre*.

FIGURA 22 - DIAGRAMA DE CLASSES



5.3.2 As Classes e seus Significados

Todas as classes no diagrama de classes exposto anteriormente têm alguma responsabilidade para a persistência dos fenômenos. A tabela 1, a seguir, ilustra a(s) responsabilidade(s) para cada classe do diagrama.

TABELA 1 – CLASSES E SEUS SIGNIFICADOS

Classe/Interface	Significado
Serializable	Interface padrão da linguagem Java para representar objetos que podem ser serializados.
IFenomeno	Interface que obriga a classe que o implementa implementar métodos utilizados para acessar suas propriedades semânticas.
FenomenoTerrestre	Classe base do Modelo de Referência
Serializador	Classe responsável pelo armazenamento e recuperação de objetos serializados.
Carta	Classe que representa uma carta topográfica.
CartaContr	Classe responsável por encontrar uma carta topográfica apropriada a posição requerida.
Mapeador	Classe responsável por fazer o mapeamento das propriedades semânticas do fenômeno em um modelo relacional.
GeralID	Classe responsável pela geração de identificadores (ID) para os objetos mapeados no banco de dados.
Controlador	Classe controladora que faz a ligação das ações entre os diversos objetos do modelo.

5.3.3 Diagramas de Seqüência

O diagrama de seqüência ilustra a interação entre os objetos através de mensagens. Os objetos são colocados ao longo do eixo X e as mensagens que esses objetos enviam e recebem são colocadas ao longo do eixo Y em ordem crescente de tempo, de cima para baixo.(BOOCH, 1999).

Para o melhor entendimento da interação entre os objetos para solução de persistência foram criados dois diagramas de seqüência, um que representa o armazenamento de fenômenos terrestres, representado pela figura 23, e outro para a recuperação dos fenômenos terrestres, representado pela figura 24.

FIGURA 23 – DIAGRAMA DE SEQÜÊNCIA ARMAZENA FENÓMENOS

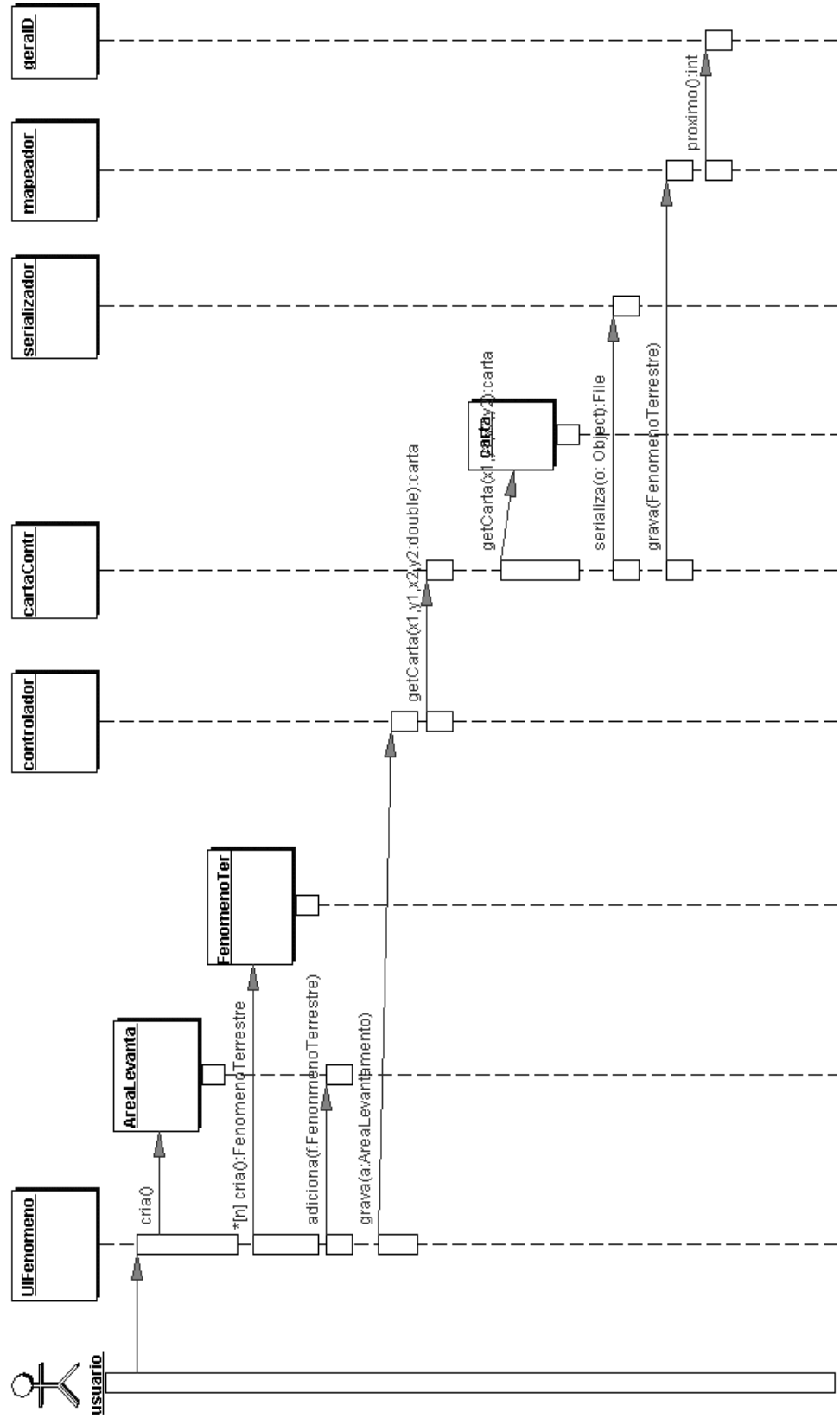
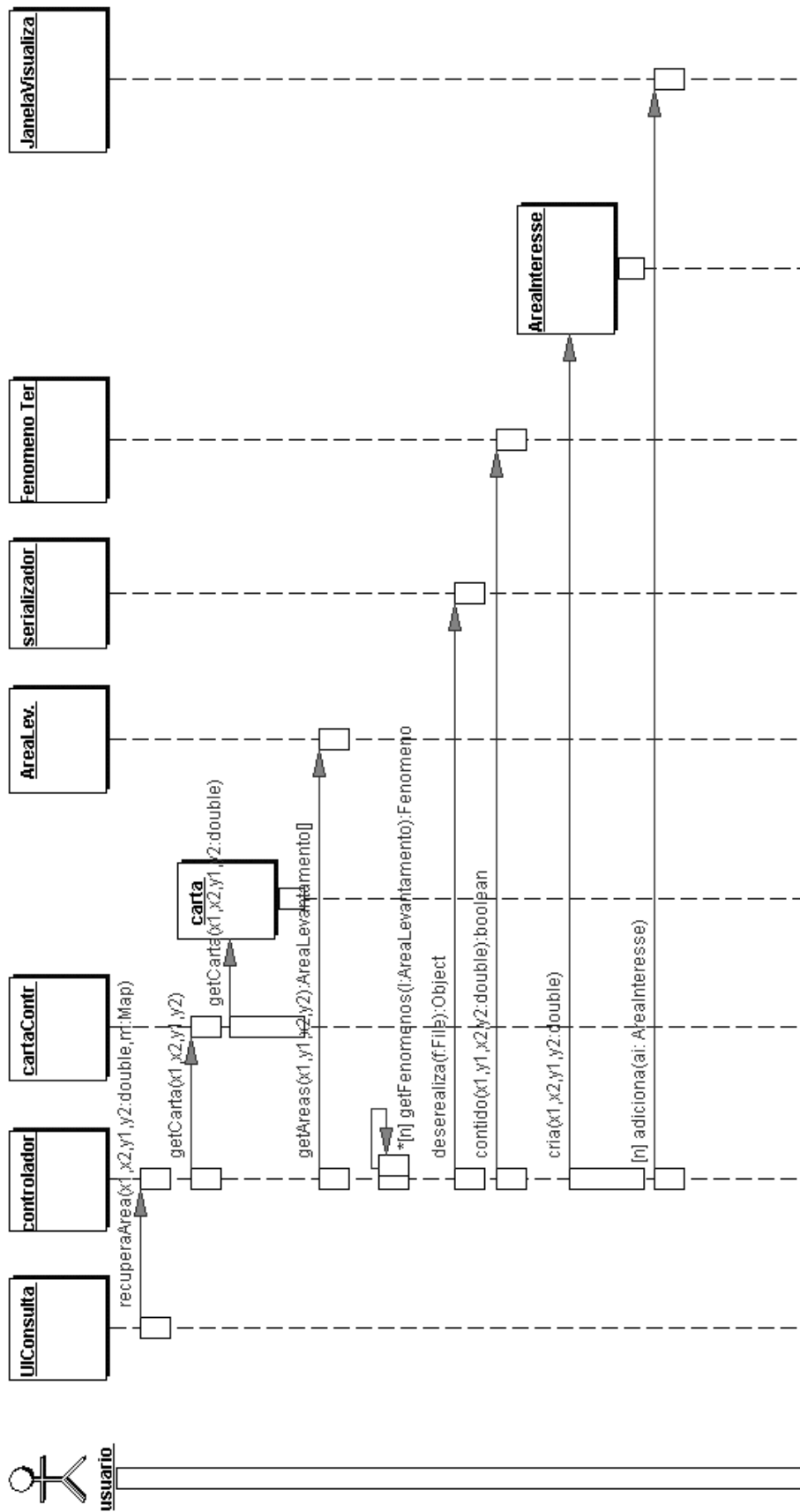


FIGURA 24 - DIAGRAMA DE SEQUÊNCIA RECUPERA FENÔMENOS



5.3.4 Diagrama de seqüência armazena fenômenos

Este diagrama de seqüência inicia quando o usuário solicita por meio de uma janela de interface qualquer, representada pela classe UIFenomeno, a leitura das informações topográficas levantadas. A partir dessas informações, uma nova área de levantamento é criada e fenômenos são atribuídos a ela. A classe controladora (controlador) é responsável por acionar o mecanismo de armazenamento que persiste a área de levantamento e seus fenômenos no banco de dados. A tabela 2 mostra as principais mensagens de cada classe usada no diagrama e seus significados.

TABELA 2 - MENSAGENS DO DIAGRAMA DE SEQÜÊNCIA ARMAZENA FENÔMENOS

Classe	Mensagem	Significado
FenomenoTerrestre	cria	criação de um novo fenômeno terrestre
AreaLevantamento	adiciona	adição de um fenômeno terrestre à área de levantamento
Controlador	grava	gravação de uma área de levantamento e seus fenômenos terrestres
CartaContr	getCarta	recuperação de uma carta topográfica que contenha a área em questão
Serializador	serializa	serializa o objeto
Mapeador	grava	grava um mapeamento das propriedades do fenômeno
geralD	proximo	gera um identificador para o fenômeno

5.3.5 Diagrama de seqüência recupera fenômenos

Este diagrama de seqüência inicia quando o usuário informa por meio de uma janela de interface qualquer, representada pela classe UIConsulta, a

região de interesse quais as características semânticas que os fenômenos de seu interesse possuem. Uma nova janela de visualização é criada, as áreas de levantamento contidas na janela de visualização são localizadas e os fenômenos que corresponderem aos critérios de busca serão armazenados em suas áreas de interesse respectivas, que são incluídas por sua vez dentro da janela de visualização. A tabela 3 mostra as principais mensagens de cada classe usada no diagrama e seus significados.

TABELA 3 - MENSAGENS DO DIAGRAMA DE SEQÜÊNCIA RECUPERA FENÔMENOS

Classe	Mensagem	Significado
controlador	RecuperaArea	solicita a recuperação das áreas de levantamento
	GetFenomenos	recupera os fenômenos da área de levantamento
AreaLevantamento	GetAreas	recupera as áreas de levantamento que estão contidas em determinada região
CartaContr	getCarta	recuperação de uma carta topográfica que contenha a área em questão
FenomenoTerrestre	contido	Informa se o fenômeno pertence a região
janelaVisualizacao	adiciona	adiciona uma área de interesse
	cria	criação da janela
AreaInteresse	Cria	criação da área de interesse
Serializador	deserializa	Deserializa o fenômeno terrestre

5.4 FORMAS DE PERSISTÊNCIA PARA O MODELO DE REFERÊNCIA

É interessante prover à interface de persistência um modo alternativo para facilitar a troca de informações entre as possíveis aplicações que utilizem

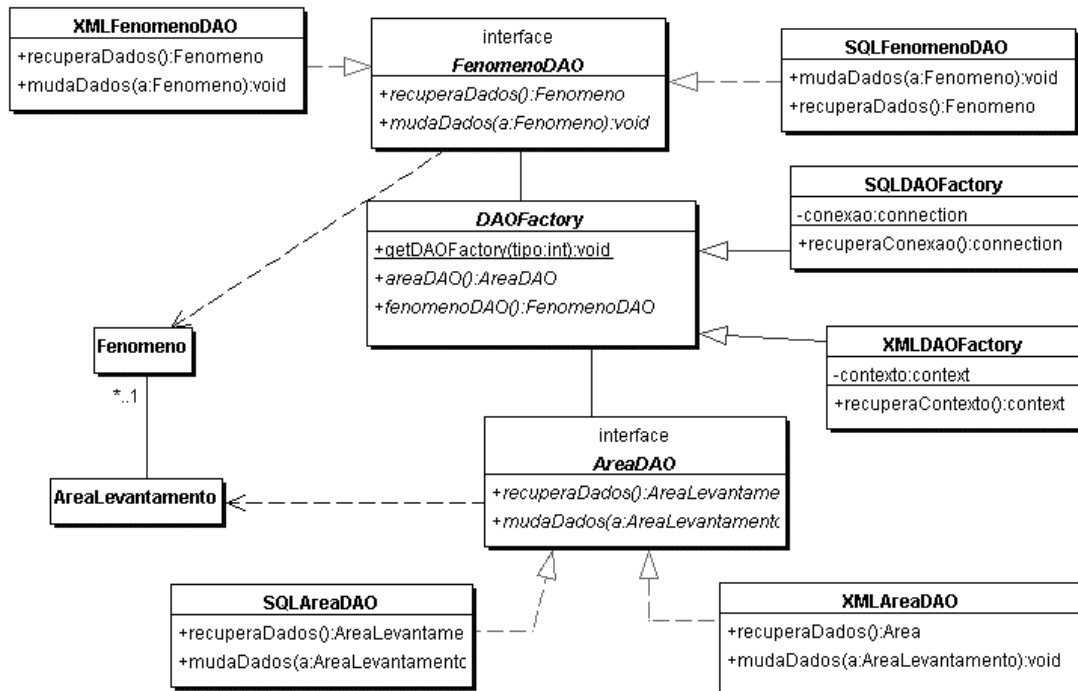
os fenômenos terrestres como fonte de dados. A solução escolhida foi expressar os fenômenos de maneira descritiva através de arquivos XML (*Extensible Markup Language*), dada a flexibilidade que eles permitem para estruturar informações.

Com o intuito de prover a persistência do Modelo de Referência em diversas formas, projetou-se uma estrutura de classes que utiliza um padrão de projeto, solução já consagrada no projeto de software orientado a objetos, específico para auxiliar o acesso a diversas fontes de dados de forma independente, esse padrão é o *DAO Factory* (ALUR, 2001).

O padrão de projeto *DAO Factory* é uma combinação do padrão DAO, *Data Access Objects*, (ALUR, 2001) e do padrão *Factory* (GAMMA et al, 1995). O padrão DAO provê uma abstração para acesso a fontes de dados, enquanto o padrão *Factory* fornece uma interface para instanciar um objeto deixando a responsabilidade de decidir qual objeto deve ser criado para as subclasses.

O padrão *DAO Factory* utiliza uma classe abstrata que pode construir vários tipos de *DAO Factory* concretas, cada uma delas suportando um tipo diferente de implementação para persistência. A figura 25, a seguir, ilustra um diagrama de classes que implementa o padrão *DAO Factory* para Fenômenos Terrestres, facilitando a utilização de formas alternativas de persistência:

FIGURA 25 – CLASSES PARA PERSISTÊNCIA EM MEIOS ALTERNATIVOS

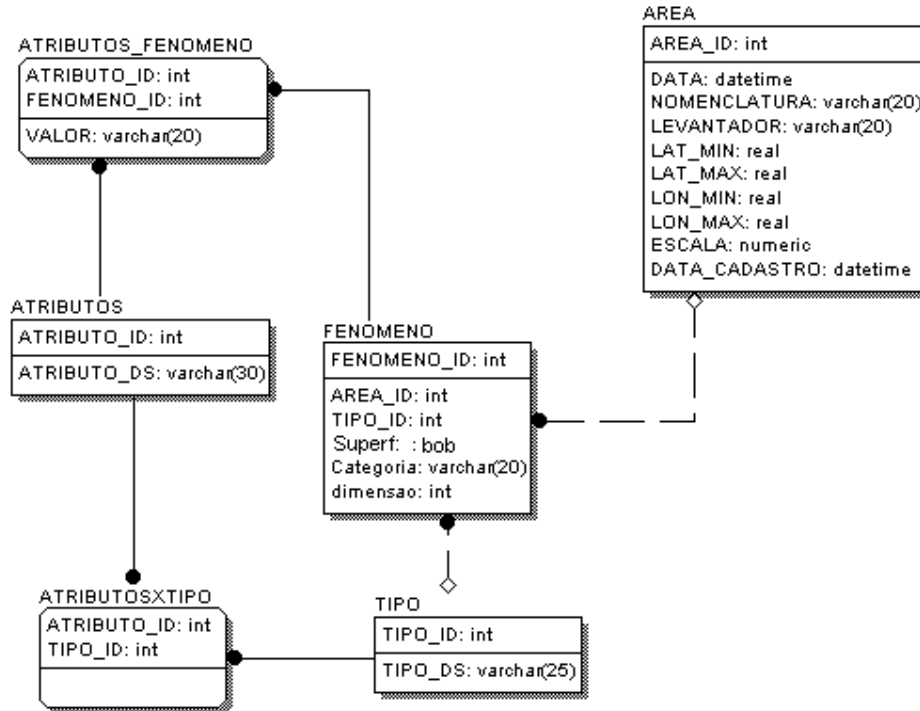


A interface de persistência de fenômenos terrestres utilizou-se do padrão “DAO Factory”, para implementar duas formas de persistência para o Modelo de Referência: uma em Banco de Dados Relacional (classes com prefixo SQL) e outra em que a área de levantamento e os fenômenos podem ser persistidos na forma de um documento em XML (classes com prefixo XML). Para implementar esse novo padrão, a classe DAOFactory substitui a classe serializador (ver diagramas de seqüência).

5.4.1 O Modelo de Referência adaptado para SGBDR

O modelo relacional do banco de dados utilizado que permite a persistência das informações semânticas dos fenômenos terrestres e de sua área de levantamento é mostrado na figura 26.

FIGURA 26 – MODELO E/R NO BANCO DE DADOS



O modelo mostra o relacionamento dos fenômenos terrestres, tabela FENOMENO, com a área de levantamento, tabela AREA, e também o relacionamento do fenômeno com seus atributos semânticos representados pela tabela TIPO. O que significa um conjunto de atributos para cada tipo de fenômeno.

5.4.1.1 Dicionário de dados

O Dicionário de Dados é um documento que expressa o significado para cada elemento no diagrama entidade-relacionamento (tabelas e colunas). Do conjunto de atributos das tabelas utilizadas no mapeamento dos objetos, apenas duas possuem características especiais. Esses dois atributos ou colunas são mostrados na Tabela 4.

TABELA 4- ATRIBUTOS ESPECIAIS DO DICIONÁRIO DE DADOS

Tabela	Atributo	Instância
AREA	nomenclatura	SF23-XC-V3NO-XIINO-V
FENOMENO	Superf	0xACED000573720...

O atributo *nomenclatura* representa um código gerado para identificação do desdobramento da carta topográfica original, particionamento do espaço, gerando uma outra carta que é um subconjunto da primeira. Para maiores detalhes sobre a geração do código de nomenclatura de cartas, consulte o apêndice I.

O atributo *imagem* representa um campo binário em que está representada a superfície do fenômeno terrestre, parte geométrica, do fenômeno terrestre de forma serializada.

A tabela 5, a seguir, mostra o dicionário de dados completo para o esquema do banco de dados desenvolvido:

TABELA 5 – DICIONÁRIO DE DADOS

Tabela	Coluna da Tabela	Tipo de Dado	Campo Nulo	Campo Chave Primária	Campo Chave Estrangeira	Significado
AREA	AREA_ID	inteiro	Não	Sim	Não	Identificador da área de Levantamento no banco de dados
	DATA	data	Não	Não	Não	Data quando foi feito o levantamento
	NOMENCLATURA	caractere	Sim	Não	Não	nomenclatura da carta topográfica associada
	LEVANTADOR	caractere	Sim	Não	Não	Nome do cartógrafo responsável
	LAT_MIN	real	Sim	Não	Não	Latitude mínima registrada da área
	LAT_MAX	real	Sim	Não	Não	Latitude máxima registrada da área
	LON_MIN	real	Sim	Não	Não	Longitude mínima registrada da área
	LON_MAX	real	Sim	Não	Não	Longitude máxima registrada da área
	ESCALA	Inteiro	Sim	Não	Não	Escala em que foram levantados os dados
	DATA_CADASTRO	data	Sim	Não	Não	Data de cadastro dos dados
ATRIBUTOS	ATRIBUTO_ID	Inteiro	Não	Sim	Não	Identificador do atributo no banco de dados
	ATRIBUTO_DS	caractere	Sim	Não	Não	Descrição do nome do atributo (propriedade)
ATRIBUTOS_FENOMENO	ATRIBUTO_ID	Inteiro	Não	Sim	Sim	Identificador do atributo a que se refere
	FENOMENO_ID	Inteiro	Não	Não	Sim	Identificador do fenômeno terrestre ao qual o atributo se refere
	VALOR	caractere	Sim	Não	Não	Valor do atributo para o fenômeno associado
ATRIBUTOSXTIPO	ATRIBUTO_ID	inteiro	Não	Sim	Sim	Identificador do atributo para o tipo ao qual ele será associado
	TIPO_ID	Inteiro	Não	Não	Não	Identificado para o tipo de fenômenos.
FENOMENO	SUPERF	binário	Sim	Não	Não	Representa as características geométricas (superfície) do fenômeno.
	AREA_ID	inteiro	Sim	Não	Não	Identificador da área ao qual o fenômeno pertence.
	FENOMENO_ID	inteiro	Sim	Não	Sim	Identificador do fenômeno no banco de dados
TIPO	TIPO_ID	inteiro	Não	Sim	Não	Identificar para o tipo de fenômeno no banco de dados
	TIPO_DS	caractere	Sim	Não	Não	Nome do tipo de fenômeno associado (rua, edificação etc)

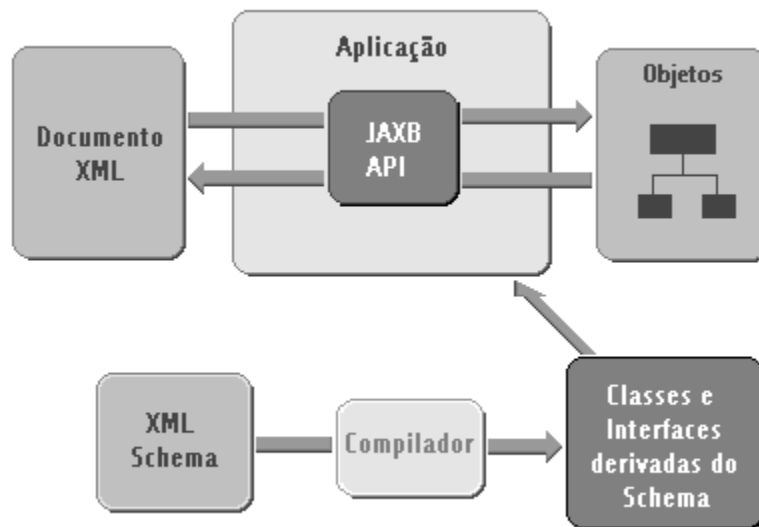
5.4.2 O Modelo de Referência adaptado para XML

Com a linguagem Java, a manipulação de arquivos XML pode ser feita a partir de duas API básicas:

- a) *Document Object Model* (DOM) – todo documento XML é lido e transformado em uma estrutura tipo árvore, onde os “nós” representam os elementos do documento XML, seja ele *tag*, atributo ou trecho de texto. Neste modo de manipulação, o consumo de memória ao ler arquivos grandes é sensível.
- b) *Simple API for XML* (SAX) – durante a leitura do arquivo XML são gerados eventos que devem ser tratados pelo programador. Deste modo, o consumo de memória é pequeno, mas para algumas estruturas de documentos XML complexos fica difícil de se trabalhar.

A alternativa mais eficiente encontrada para a conversão de objetos em documentos XML e vice-versa, na linguagem Java, foi a API JAXB. A API JAXB (*Java Architecture for XML Binding*) é uma ferramenta que permite o mapeamento bidirecional entre objetos Java e documentos XML. Com a definição do *XML-Schema*, documento de definição da estrutura do XML, um compilador gera um conjunto de classes que permitem ao desenvolvedor manipular e criar documentos XML, sem se preocupar com a estrutura dos dados (TAI-WEI LIN, 2003). A figura 27, a seguir, ilustra como funciona, em linhas gerais, a arquitetura JAXB.

FIGURA 27 – ARQUITETURA JAXB



Para a utilização do JAXB sobre os fenômenos terrestres necessitou-se criar um arquivo XML-Schema que se chamou de *fenomenos.xsd*. Esse arquivo contém a definição básica para os objetos: *janelaVisualizacao*, *AreaInteresse* e *FenomenoTerrestre* baseada no Modelo de Referência, para conversão em XML. O arquivo *fenomenos.xsd* é mostrado a seguir:

QUADRO 3 - ARQUIVO FENOMENOS.XSD

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="janela" type="JanelaVisualizacaoType"/>
<xsd:complexType name="JanelaVisualizacaoType">
  <xsd:sequence>
    <xsd:element name="areas" type="AreaInteresseType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="minX" type="xsd:string" />
  <xsd:attribute name="minY" type="xsd:string" />
  <xsd:attribute name="maxX" type="xsd:string" />
  <xsd:attribute name="maxY" type="xsd:string" />
</xsd:complexType>
<xsd:complexType name="AreaInteresseType">
  <xsd:sequence>
    <xsd:element name="fenomenos" type="FenomenoType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="minX" type="xsd:string" />

```

```

<xsd:attribute name="minY" type="xsd:string" />
<xsd:attribute name="maxX" type="xsd:string" />
<xsd:attribute name="maxY" type="xsd:string" />
<xsd:attribute name="nome" type="xsd:string" />
<xsd:attribute name="descricao" type="xsd:string" />
<xsd:attribute name="tempo" type="xsd:string" />
</xsd:complexType>
<xsd:complexType name="FenomenoType">
  <xsd:sequence>
    <xsd:element name="atributo" type="AtributoType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="superficie0D" type="Superficie0DType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="superficie1D" type="Superficie1DType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="superficie2D" type="Superficie2DType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="superficie3D" type="Superficie3DType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="fenomenoID" type="xsd:string" />
  <xsd:attribute name="areaLevantamentoID" type="xsd:string" />
  <xsd:attribute name="categoria" type="xsd:string" />
  <xsd:attribute name="dimensao" type="xsd:string" />
  <xsd:attribute name="nome" type="xsd:string" />
  <xsd:attribute name="descricao" type="xsd:string" />
</xsd:complexType>
<xsd:complexType name="Superficie0DType">
  <xsd:sequence>
    <xsd:element name="pontos" type="PontoType" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="xmax" type="xsd:string" />
  <xsd:attribute name="xmin" type="xsd:string" />
  <xsd:attribute name="ymax" type="xsd:string" />
  <xsd:attribute name="ymin" type="xsd:string" />
</xsd:complexType>
<xsd:complexType name="Superficie1DType">
  <xsd:sequence>
    <xsd:element name="pontos" type="PontoType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="xmax" type="xsd:string" />
  <xsd:attribute name="xmin" type="xsd:string" />
  <xsd:attribute name="ymax" type="xsd:string" />
  <xsd:attribute name="ymin" type="xsd:string" />
</xsd:complexType>
<xsd:complexType name="Superficie2DType">
  <xsd:sequence>
    <xsd:element name="pontos" type="PontoType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="faces" type="FaceType" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="xmax" type="xsd:string" />
  <xsd:attribute name="xmin" type="xsd:string" />
  <xsd:attribute name="ymax" type="xsd:string" />
  <xsd:attribute name="ymin" type="xsd:string" />

```

```

</xsd:complexType>
<xsd:complexType name="Superficie3DType">
  <xsd:sequence>
    <xsd:element name="pontos" type="PontoType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="faces" type="FaceType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="xmax" type="xsd:string" />
  <xsd:attribute name="xmin" type="xsd:string" />
  <xsd:attribute name="ymax" type="xsd:string" />
  <xsd:attribute name="ymin" type="xsd:string" />
</xsd:complexType>
<xsd:complexType name="FaceType">
  <xsd:sequence>
    <xsd:element name="pontos" type="PontoType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AtributoType">
  <xsd:attribute name="nome" type="xsd:string"/>
  <xsd:attribute name="valor" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="PontoType">
  <xsd:attribute name="seq" type="xsd:string"/>
  <xsd:attribute name="x" type="xsd:string"/>
  <xsd:attribute name="y" type="xsd:string"/>
  <xsd:attribute name="z" type="xsd:string"/>
  <xsd:attribute name="medido" type="xsd:string" />
  <xsd:attribute name="topografico" type="xsd:string"/>
</xsd:complexType>
</xsd:schema>

```

As principais *tags*, ou marcações, utilizadas no desenvolvimento desde modelo, de representação de fenômenos em XML baseado no modelo de referência, foram:

- ComplexType – Define a criação de um novo tipo.
- Sequence – Define uma sequência de elementos, ou grupo, que um determinado tipo pode conter.
- Element – Define um novo elemento sendo de um determinado tipo, pode ocorrer 0 ou mais ocorrências do mesmo elemento.
- Attribute – Define um atributo, propriedade, para o elemento.

Para que se consiga utilizar o JAXB na transformação de objetos em

XML e vice-versa é necessário gerar um conjunto de classes auxiliares utilizando o compilador *xjc*, que faz parte da API. Esse compilador é utilizado sobre o *XML-Schema* (fenomenos.xsd) para que sejam geradas as classes auxiliares de transformação. A figura 28 ilustra a criação das classes para um pacote chamado “sogis.xml”.

FIGURA 28 - UTILIZAÇÃO DO COMPILADOR XJC PARA GERAR CLASSES AUXILIARES

```
C:\temp>xjc -p sogis.xml fenomenos.xsd
parsing a schema...
compiling a schema...
sogis\xml\AreaInteresse.java
sogis\xml\AreaInteresseType.java
sogis\xml\Atributo.java
sogis\xml\AtributoType.java
sogis\xml\Atributos.java
sogis\xml\AtributosType.java
sogis\xml\Categoria.java
sogis\xml\Dimensao.java
sogis\xml\Fenomeno.java
sogis\xml\FenomenoType.java
sogis\xml\Id.java
sogis\xml\JanelaVisualizacao.java
sogis\xml\JanelaVisualizacaoType.java
sogis\xml\Nome.java
sogis\xml\ObjectFactory.java
sogis\xml\Ponto.java
sogis\xml\PontoType.java
sogis\xml\Superficie.java
sogis\xml\SuperficieType.java
sogis\xml\Valor.java
sogis\xml\X.java
sogis\xml\Y.java
sogis\xml\Z.java
sogis\xml\bgm.ser
sogis\xml\jaxb.properties
sogis\xml\impl\AreaInteresseImpl.java
sogis\xml\impl\AreaInteresseTypeImpl.java
sogis\xml\impl\AtributoImpl.java
sogis\xml\impl\AtributoTypeImpl.java
sogis\xml\impl\AtributosImpl.java
sogis\xml\impl\AtributosTypeImpl.java
sogis\xml\impl\CategoriaImpl.java
sogis\xml\impl\DimensaoImpl.java
sogis\xml\impl\FenomenoImpl.java
sogis\xml\impl\FenomenoTypeImpl.java
sogis\xml\impl\IdImpl.java
sogis\xml\impl\JanelaVisualizacaoImpl.java
sogis\xml\impl\JanelaVisualizacaoTypeImpl.java
sogis\xml\impl\NomeImpl.java
sogis\xml\impl\PontoImpl.java
sogis\xml\impl\PontoTypeImpl.java
sogis\xml\impl\SuperficieImpl.java
sogis\xml\impl\SuperficieTypeImpl.java
sogis\xml\impl\ValorImpl.java
sogis\xml\impl\XImpl.java
sogis\xml\impl\YImpl.java
sogis\xml\impl\ZImpl.java
```

Após a geração das classes, estas são compiladas e já podem ser utilizadas para a transformação de um fenômeno em XML, ou de um XML para

fenômeno. O processo de transformação de um objeto para um documento XML é chamado pela API JAXB de *marshall*. A seguir é feita uma demonstração do processo de *marshall* de um objeto *janelaVisualizacao* em um arquivo chamado *fenomenos.xml*.

QUADRO 4 – CÓDIGO PARA O PROCESSO DE MARSHALL

```
File f = new File("fenomenos.xml");
JAXBContext Context = JAXBContext.newInstance("sogis.xml");
Marshaller m = jc.createMarshaller();
FileWriter fout = new FileWriter(f);
m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT,new Boolean(true));
m.marshal(janelaVisualizacao,fout);
fout.close();
```

Para fazer a restauração dos objetos a partir de um documento XML é necessário fazer a transformação inversa do documento XML em um objeto (fenômeno terrestre). Este processo é conhecido na API JAXB como *unmarshal*. O quadro a seguir ilustra o código necessário para realizar o *unmarshal*.

QUADRO 5- CÓDIGO PARA O PROCESSO DE UNMARSHALL

```
File f = new File("fenomenos.xml");
JAXBContext Context = JAXBContext.newInstance("sogis.xml");
Unmarshaller u = jc.createUnmarshaller();
janelaVisualizacao janela = (janelaVisualizacao)u.unmarshal(new FileInputStream(f));
```

6 UMA APLICAÇÃO COMO ESTUDO DE CASO

Para demonstrar o funcionamento da interface de persistência de fenômenos, optou-se pela criação de uma aplicação experimental, que servirá como um estudo de caso para o experimento. Esta aplicação será discutida a seguir na forma de casos de uso, ou seja, através das funcionalidades que apresenta.

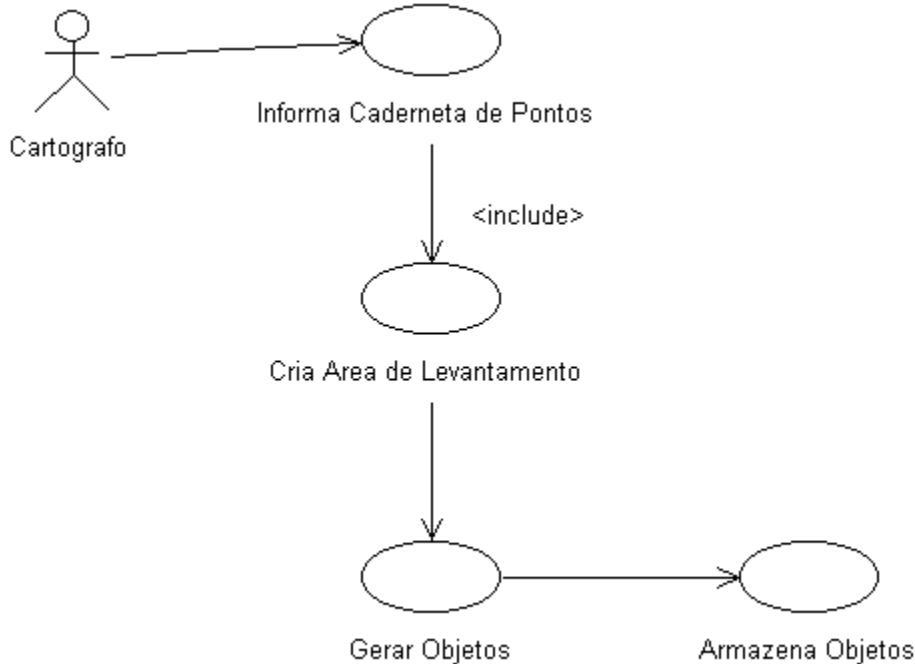
6.1 CASOS DE USO

A seguir, discute-se dois casos de uso básicos para demonstrar uma aplicação de persistência de fenômenos: o Armazenamento e a Recuperação. Estes representam as operações básicas da interface de persistência: gravação e leitura de fenômenos, respectivamente.

6.1.1 Caso De Uso Armazenar Fenômeno

Este caso de uso representa as ações necessárias para que o fenômeno possa ser incorporado no sistema. Os fenômenos são gerados utilizando os pontos fornecidos no levantamento de campo e então vinculados a uma área de levantamento. A figura 29, a seguir, demonstra este caso de uso em notação UML.

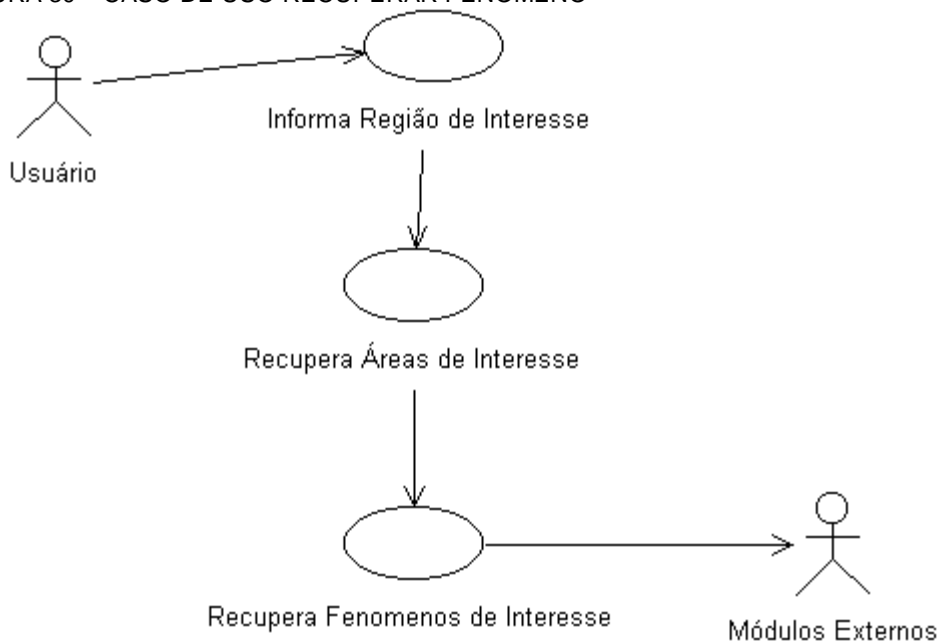
FIGURA 29 – CASO DE USO ARMAZENAR FENOMENO



6.1.2 Caso De Uso Recuperar Fenômeno

Este caso de uso representa as ações necessárias para que ocorra a recuperação de áreas de observação, e seus respectivos fenômenos, pelo sistema. A área de levantamento é encontrada utilizando sua posição geográfica e os fenômenos contidos na área encontrada podem ser restaurados por uma seleção baseada em suas propriedades semânticas. Por exemplo: Pode ser necessário recuperar uma área de levantamento que contenha as seguintes coordenadas, utilizando o sistema geodésico, latitude -25.4529 e longitude 49.2336 e que os fenômenos contidos nesta área do tipo “árvore” tenham altura igual a um metro. A figura 30, a seguir, demonstra este caso de uso em notação UML.

FIGURA 30 – CASO DE USO RECUPERAR FENOMENO



6.2 ESPECIFICAÇÃO DE CASOS DE USO

6.2.1 Especificação do Caso de Uso Armazenar Fenômeno

O caso de uso armazenar fenômeno representa uma seqüência de passos necessários para inserir os fenômenos terrestres na base de dados. A especificação a seguir ilustra o curso básico, alternativo e de exceções que são previstos de ocorrer neste caso de uso:

QUADRO 6 – ESPECIFICAÇÃO DO CASO DE USO ARMAZENAR FENÔMENO

1 Objetivo	Permitir ao cartógrafo responsável importar fenômenos terrestres registrados em uma caderneta para uma base de dados.
2 Pré-Condições	Não se aplica.
3 Pós-Condições	Não se aplica.
4 Curso de Eventos	
4.1 Curso Básico	
	<p>Se a região informada pertencer a região de trabalho (Brasil) então</p> <p style="padding-left: 40px;">Lê os pontos topográficos da caderneta e determina os limites da área de levantamento.</p> <p style="padding-left: 40px;">O sistema cria uma área de levantamento.</p> <p style="padding-left: 40px;">O sistema apresenta os níveis de informação extraídos da caderneta.</p> <p style="padding-left: 40px;">Se o usuário não selecionar nenhum nível e selecionar a opção gerar então <i>nível não selecionado</i> (curso de exceção)</p> <p style="padding-left: 40px;">Se o usuário selecionar um nível ou mais e escolher a opção gerar então <i>Gerar objetos</i> (curso alternativo)</p> <p style="padding-left: 40px;">Se o usuário selecionar a opção sair então <i>Gravar Objetos</i> (curso alternativo).</p>
4.2 Curso Alternativo	

Gerar Objetos:

Se algum nível selecionado

então o sistema gera um fenômeno topográfico.

O sistema associa o fenômeno à área de levantamento.

Gravar Objetos:

Se existirem objetos associados a área de levantamento

então Grave os objetos

4.3 Curso de Exceções*Nível não selecionado*

O sistema mostra a mensagem 'nenhum nível foi selecionado'

5 Observações

Não se aplica.

6.2.2 Especificação do Caso de Uso Recuperar Fenômeno

O caso de uso recuperar fenômeno tem uma seqüência de passos para se restaurar os fenômenos terrestres da base de dados. A especificação a seguir ilustra o curso básico, alternativo e de exceções que são previstos de ocorrer neste caso de uso:

QUADRO 7 – ESPECIFICAÇÃO DO CASO DE USO RECUPERAR FENÔMENO

1 Objetivo	Permitir a recuperação de fenômenos terrestres pela aplicação.
2 Pré-Condições	Não se aplica.
3 Pós-Condições	Não se aplica.
4 Curso de Eventos	
4.1 Curso Básico	
	<p>Se o usuário informar uma região dentro dos limites da aplicação então <i>mostrar áreas</i> (curso alternativo).</p> <p style="padding-left: 40px;">O sistema apresenta as áreas de levantamento contidas na área de interesse do usuário.</p> <p style="padding-left: 40px;">O usuário seleciona qual área de levantamento e quais fenômenos devem ser recuperados.</p> <p style="padding-left: 40px;">O sistema recupera os fenômenos que obedecerem aos critérios de seleção.</p> <p style="padding-left: 40px;">Se houver um ou mais fenômenos resultantes da seleção então o sistema repassa os fenômenos para uma aplicação externa.</p> <p>Senão</p> <p style="padding-left: 40px;">O sistema apresenta a exceção <i>fenômenos não encontrados</i></p> <p>Senão</p> <p style="padding-left: 40px;">O sistema apresenta a exceção <i>região fora dos limites da aplicação</i></p>
4.2 Curso Alternativo	

<i>Mostrar áreas:</i>	
O sistema lista as áreas de levantamento registradas que estão contidas dentro da área de interesse do usuário.	
4.3 Curso de Exceções	
<i>Região fora dos limites da aplicação</i>	
O sistema mostra a mensagem 'Esta região está fora dos limites para esta aplicação'.	
<i>Fenômenos não encontrados</i>	
O sistema mostra a mensagem 'Nenhum Fenômeno encontrado'.	
5 Observações	Não se aplica.

6.2.3 CONSIDERAÇÕES

Não é do escopo da aplicação, estudo de caso, aspectos fora da persistência tais como:

- A maneira pela qual os fenômenos são criados na sua forma digital, ou seja, os algoritmos e estruturas de dados utilizados para a criação do fenômeno terrestre como objeto não são importantes, contanto que o objeto esteja de acordo com o Modelo de Referência;
- Validações relacionadas à origem ou consistência dos dados utilizados como fonte para criação dos objetos;

6.3 TESTANDO A APLICAÇÃO

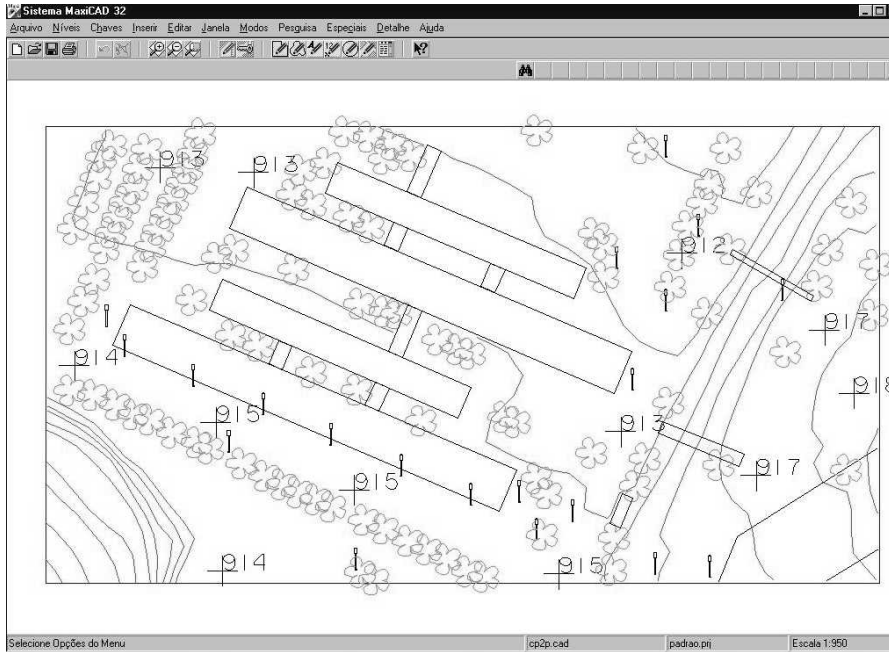
A criação e armazenamento dos fenômenos baseiam-se em um arquivo de entrada onde estes dados são repassados à aplicação de persistência que se encarrega de registrá-los na base de dados.

Na recuperação dos fenômenos, estes são localizados e extraídos da base de dados e ficam contidos em um objeto *JaneladeVisualizacao* que funciona como um *container* onde ficam disponíveis para outra aplicação ou a partir do qual é gerado um documento XML.

6.3.1 Obtendo dados

Para a reprodução do experimento utilizou-se um conjunto de dados reais que representam áreas de levantamento e fenômenos presentes no Centro Politécnico da Universidade Federal do Paraná. Esses dados atualmente encontram-se disponíveis em arquivos no formato proprietário do software Maxicad®. A figura 31, a seguir, ilustra como uma área de levantamento é observada utilizando o Maxicad:

FIGURA 31 - ÁREA DE LEVANTAMENTO VISTA NO MAXICAD



O quadro, a seguir, ilustra um trecho de arquivo fonte de dados a partir do qual as informações sobre os fenômenos podem ser lidas. A região delimitada pelos pontos máximos e mínimos contidos no arquivo fonte de dados representa uma área de levantamento.

QUADRO 8 – TRECHO DE UMA ÁREA DE LEVANTAMENTO EM FORMATO TEXTO

```
; MaxiCad 32 - Versao For Windows
; c:\ano2002\tese\prog\l5\cp2c1.cad 11/24/02 17:50:14.0
;[padrao]
M 1.0051SSU25
N000001Area do Desenho      *
E000004000
1 677600.270000 7183713.200000 0.00 01 #
1 677600.270000 7183832.420000 0.00 01 #
1 677817.600000 7183832.420000 0.00 01 #
1 677817.600000 7183713.200000 0.00 01 #
1 677600.270000 7183713.200000 0.00 01 #
D0000
1 677600.270000 7183713.200000 0.00 01 #
1 677600.270000 7183832.420000 0.00 01 #
1 677817.600000 7183832.420000 0.00 01 #
1 677817.600000 7183713.200000 0.00 01 #
1 677600.270000 7183713.200000 0.00 01 #
```


N052115PontosAltimetricos *
.824598 914.150024 #
S0800120000

6.3.2 Armazenando Fenômenos

Na leitura do arquivo fonte de dados, as informações sobre os fenômenos são extraídas e um novo fenômeno terrestre é instanciado. Após a instanciação do objeto, suas características semânticas são mapeadas para o Modelo E/R proposto pela interface de persistência. Enquanto suas características geométricas são gravadas em um arquivo.

6.3.3 Recuperando Fenômenos

Antes da recuperação do fenômeno terrestre é necessário localizá-lo espacialmente. Na aplicação utilizou-se uma pequena janela de interface com o usuário que permite informar as coordenadas máximas e mínimas de interesse. Com essas informações, um objeto *JaneladeVisualizacao* é criado e, em seguida, inicia-se a recuperação na base de dados das Áreas de Levantamento que possuam fenômenos dentro do retângulo da *JaneladeVisualizacao*. A figura 32 ilustra uma interface com o usuário usada em testes para que se determine a janela de visualização desejada:

FIGURA 32 – INTERFACE COM USUÁRIO PARA DETERMINAR A JANELA DE VISUALIZAÇÃO

Consulta de fenômenos:

	Min.	Max.
Latitude	-25.45292	-25.45182
Longitude	-49.23366	-49.2315

Janela de Visualização(*.xml):

Áreas Disponíveis

2) | -49.23366165161133,-49.231502532958984,-25.4529209136962
 1)q| -25.452917098999023,-25.451818466186523,-49.23366165161

Áreas(s) encontrada(s):

Encontrar Carta(s) Atributos Semânticos OK

Além das propriedades espaciais, durante a pesquisa pode-se escolher critérios de seleção baseados nas propriedades semânticas dos fenômenos. Por exemplo, superfície terrestre em que tipo do solo seja arenoso. A figura 33 ilustra a interface com o usuário que permite a seleção dos critérios semânticos para pesquisa dos fenômenos terrestres.

FIGURA 33 – INTERFACE COM O USUÁRIO PARA DETERMINAR OS CRITÉRIOS SEMÂNTICOS

nome	valor	pesquisar
Largura		<input type="checkbox"/>
TIPO SOLO		<input type="checkbox"/>

6.3.4 Exportando e Importando em Formato XML

Após armazenar e posteriormente recuperar os fenômenos da base de dados é interessante que se possa ilustrar o resultado do experimento de alguma forma. Por esse motivo, utilizando-se a interface de persistência, gerou-se um documento XML que representa uma área de levantamento. O quadro a seguir ilustra o resultado da “exportação” de um fenômeno para um arquivo XML.

QUADRO 9 - ARQUIVO FENOMENOS.XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<areas nome="nome" maxX="-25.45182" descricao="descArea" minY="-
49.23366165161133" tempo="tempo" maxY="-49.231502532958984" minX="-
25.45292">
<fenomenos dimensao="1" categoria="DTM" nome="DTM" descricao="DTM"
fenomenoID="1">
<atributo valor="q" nome="teste1"></atributo>
<atributo valor="q" nome="TIPO SOLO"></atributo>
<superficie1D>
<pontos medido="N" x="0.0" topografico="S" z="0.0" y="0.0" seq="0"></pontos>
...
</superficie1D>
</fenomenos>
</areas>
```

7 UMA INTEGRAÇÃO PARA TESTE

7.1 A APLICAÇÃO VFT

Com o propósito de demonstrar a utilização da interface de persistência por uma aplicação que utiliza os fenômenos terrestres, optou-se por utilizar a aplicação VFT (CAMARGO,2003), para que utilize a interface de persistência.

A aplicação VFT , Visualização de Fenômenos Terrestres, utiliza também o Modelo de Referência para representar os fenômenos terrestres e transformar áreas de levantamentos e fenômenos a partir de um arquivo XML em uma cena VRML.

7.2 ADAPTAÇÕES NECESSÁRIAS

Algumas adaptações foram necessárias na aplicação VFT para que esta possa utilizar a interface de persistência. Essas adaptações são citadas nas seções a seguir.

7.2.1 Classe BdFTBean

Para realizar a persistência, a VFT utiliza uma classe BdFTBean responsável pelo armazenamento e recuperação dos fenômenos. Para obter maiores detalhes sobre a classe consulte os apêndices da dissertação (CAMARGO, 2003).

7.2.2 Alterações na classe BdFTBean

Para que a classe BdFTBean passe a utilizar a interface de persistência, necessitou-se apenas substituir as chamadas diretas ao banco de dados por chamadas a métodos da classe *controler* (da interface de persistência). O quadro a seguir mostra os trechos de código alterados para a integração:

QUADRO 10 – TRECHOS DE CÓDIGO DA CLASSE BDFTBEAN ALTERADA

```
public AreaLevantamento loadAreaLevantamento(String areald)
throws BdFTException {
    AreaLevantamentoParser parser;
    areald=areald.trim();

    // new Line
    AreaLevantamento area=controler.getArea(areald);
    ....
}

public ArrayList findAreasLevantamentoBD(
    String minX,
    String minY,
    String maxX,
    String maxY )
{
    try {
        // new Line
        AreaLevantamento areas=controler.getAreas(minX,minY,maxX,maxY); }
    ...
}
```

```
public AreaLevantamentoBD findAreaLevantamentoBD(String areald) {  
    AreaLevantamentoBD areaBD=null;  
    try {  
        AreaLevantamento areas=controler.getAreas(areald);  
    ...  
    }  
}
```

Não foram necessárias outras alterações uma vez que a própria aplicação VFT já se encarregava de criar os fenômenos como objetos, sendo de responsabilidade da interface apenas o armazenamento e recuperação dos fenômenos.

Para a integração com a aplicação VFT foi necessário utilizar apenas métodos da classe *controler*, controladora dos eventos, já que neste ponto da integração a própria aplicação já se encarregou de criar os objetos, ou seja, a área de levantamento e os fenômenos.

No protótipo da interface de persistência também foi implementada uma solução de integração por meio dos arquivos XML gerados pela própria aplicação VFT. Desta forma, a interface de persistência pode importar e exportar áreas de levantamento descritas em arquivos XML utilizadas pela aplicação VFT (CAMARGO, 2003).

8 CONCLUSÃO, RESULTADOS E TRABALHOS FUTUROS

Este trabalho teve o objetivo de propor uma interface para armazenar, recuperar e disponibilizar fenômenos terrestres persistidos na forma serializada e em XML. A diversidade de tecnologias envolvidas na construção da interface exigiu um grande esforço de aprendizagem, no entanto, o uso de tecnologias centradas em JAVA permitiu que o aprendizado obtido em uma tecnologia fosse reaproveitado em outra.

Os fenômenos utilizados para experimento são reais e observados no Centro Politécnico – Universidade Federal do Paraná. Outras fontes de dados também podem ser utilizadas já que este trabalho não está vinculado a um software específico, mas a uma representação de objetos dos fenômenos, Modelo de Referência, sobre o qual foi desenvolvida a interface de persistência de fenômenos.

A interface de persistência apresentou facilidade de uso principalmente por apresentar um baixo acoplamento com o acesso aos dados e também pelo controlador que centraliza os serviços principais. O maior esforço na construção da interface deu-se durante a sua modelagem que teve diversas iterações para se chegar a um modelo simples e significativo.

Da maneira como foi modelada e com os padrões de projeto utilizados, entre eles o *DAO Factory*, facilitou-se a diversificação de implementação de meios de persistência, por exemplo, SGBDR e XML. Esta característica é importante pois diminui significativamente o trabalho de manutenção do código, permitindo que novas características sejam adicionadas com maior facilidade.

Este trabalho procurou desenvolver uma interface de persistencia que não fosse dependente de tecnologia, por isso foi modelada de forma a poder se

adaptar a outras formas de persistência e de não estar relacionada a nenhum produto específico colocando toda a lógica de para a persistência na camada de persistência. E o fato de ter sido codificada em java lhe proporcionou ainda maior portabilidade.

A seguir estão algumas sugestões de aperfeiçoamentos que podem ser seguidos para a interface de persistência de fenômenos terrestres:

- Compatibilização do Modelo de Referência com o modelo proposto pela OGIS, consórcio de empresas e instituições ligadas a sistemas de informação geográfica, utilizando a Linguagem GML como formato de intercâmbio de dados para permitir a troca de informações.
- Estender a interface implementando o armazenamento dos fenômenos utilizando outro meio de persistência como por exemplo LDAP.

9 REFERÊNCIAS BIBLIOGRÁFICAS

ALUR, D. ; CRUPI, J; MALKS, D. **Core J2EE Patterns**. Best Practices and Design Strategies. Prentice-Hall, 2001.

AMBLER S.W. **Mapping Objects to Relational Databases**. Ronin International. 2000.

AMBLER S.W. **The Design of a Robust Persistence Layer For Relational Databases**. Ronin International. 2000.

BAIAO, F. **Bancos de Dados Orientados a Objetos e Relacionais-Objetos**. Disponível em <http://www.cos.ufrj.br/~baiao>, Acesso em 26 Out. 2003.

BOOCH G., RUMBAUGH J., JACOBSON I. **The Unified Modeling Language User Guide**. Addison-Wesley, 1999.

CAMARGO, N. F. **VFT: UMA ESTRUTURA DE APLICAÇÃO PARA VISUALIZAÇÃO DE FENÔMENOS TERRESTRES EM 3D VIA WEB**. Dissertação de Mestrado. Curitiba, Paraná. Curso de Pós Graduação em Informática da Universidade Federal do Paraná. 2003.

CELKO, Joe; CELKO, Jackie. **Verdades e Mentiras sobre banco de dados de objetos**. Byte Brasil, São Paulo, v.6, n. 10, p. 86-89, out. 1997..

FUSSEL. **Foundations of Object Relational Mapping**, Sunnyvale CA, 1997.

HIBERNATE. **Hibernate Framework**. Disponível em: <http://www.hibernate.org>. Acesso em: 30 mai de 2004.

HILL D. R. C. **Object-Oriented Analysis and Simulation**. Publishing Company, Inc., 1996.

KELLER W.; COLDEWEY J.: **Relational Database Access Layers: A Pattern Language**, in Collected Papers from the PLoP'96 and EuroPLOP'96 Conferences, Washington University, Department of Computer Science, 1997.

KELLER W.; **Mapping Objects to Tables**, EA Generali, Neusetzgasse 1, A 1100 Wien, Austria, 1998

LIMA, J. D.; CARVALHO, C. A. P. de; VIEIRA, A. J. B.; CAMARGO, N. F. **Persistência de Fenômenos Terrestres Representados como Objetos**. In: Anais do III Colóquio Brasileiro de Ciências Geodésicas. Curitiba, Paraná. Curso de Pós Graduação em Ciências Geodésicas da Universidade Federal do Paraná. 2003.

FOWLER M., RICE D., FOEMMEL M., HIEATT E., MEE R., STAFFORD R. **Patterns of Enterprise Application Architecture**, Addison Wesley., 2002

ODMG. **Object Database Management**. Disponível em <http://odmg.org>, Acesso em 26 fev. 2003.

ORACLE Corporation. **Manuais do SGBD Objeto-Relacional Oracle 8i**. 2000.

RATIONAL Corporation, **Integrating Object and Relational Technologies**, Rational Whitepaper 2002.

REIS C. A. L. **Apostila da Disciplina Banco de Dados orientados a Objetos**, Universidade Federal do Pará, 2002.

ROCHA, L.V.; IOCHPE, C.; EDELWEISS, N. O. **Framework Conceitual GeoFrame - versão 2.0**. 2001. Relatório de Pesquisa. PPGC da UFRGS. (RP-309).

STONEBRAKER M., BROWN P. **Object-Relational Database Systems**. Morgan Kaufmann, 1996.

SERVICE Architecture. **Object-relational mapping product vendors**. Disponível em: http://www.service-architecture.com/products/object-relational_mapping.html. Acesso em 30 Mai. 2004.

SUN Microsystems. **The Java™ Tutorial** A practical guide for programmers, 2002.

SUN Microsystems. **Object Serialization**. Disponível em: <http://www.java.sun.com/products/jdk/1.3/docs/guide/serialization/index.html>. Acesso em 24 fev., 2003.

VIEIRA, A. J. B.; **Modelagem espaço-temporal de fenômenos topográficos**. Plano de Tese apresentado em Seminário de Qualificação do Curso de Pós Graduação em Ciências Geodésicas da Universidade Federal do Paraná. Curitiba, 2001.

VIEIRA, A. J. B.; **Fenômenos Terrestres representados como classes e objetos**. Tese de Doutorado do Curso de Pós Graduação em Ciências Geodésicas da Universidade Federal do Paraná. Curitiba, 2004.

VIEIRA J.B., CARVALHO C.A.P, SLUTER C. R. **Modelagem Espaço-Temporal de Fenômenos Topográficos: Uma Revisão Didática**. Anais do Simpósio Brasileiro de Geomática, Presidente Prudente-SP.2002.

Tai-Wei Lin. **Java Architecture for XML Binding (JAXB): A Primer**. Disponível em: <http://developer.java.sun.com/developer/technicalArticles/jaxb/>. Acesso em 24 Ago., 2003.

TerraLib. **TerraLib**. Disponível em: <http://terralib.dpi.inpe.br/home.htm>. Acesso em 30 Ago., 2004.

10 DOCUMENTOS CONSULTADOS

GAMMA E, HELM R., JOHNSON R., VLISSIDES JOHN. **Design Patterns: Elements of Reusable Object-Oriented Software**. Reading, Massachusetts: Addison-Wesley, 1995.

LARMAN C. **Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design**, Prentice-Hall 1999.

LARMAN C. **J2EE & EJB Design Patterns**. Disponível em www.craiglarman.com. Acesso em 23 mar. 2003.

MICROSOFT CORPORATION. **Microsoft SQL Server: Getting Started With SQL Server 7.0**. SQL Server Books Online, 1998.

SUN MICROSYSTEMS. **Java™ 2 SDK, Standard Edition Documentation**. Versão 1.3. Disponível em: <http://java.sun.com/products/jdk/1.3/index.html>. Acesso em: 20 ago 2002.

UNIVERSIDADE FEDERAL DO PARANÁ, Sistema De Bibliotecas. Série **Normas Para Apresentação De Documentos Científicos**. Curitiba, 2000.

VIEIRA, A. J. B;CASTRO L. C – Textos didáticos: **Conceitos Importantes da Cartografia Digital**. Editora da UFPR. Curitiba, 2000

**APÊNDICE I – PESQUISA SOBRE ARTICULAÇÃO DE FOLHAS DE CARTAS
TOPOGRÁFICAS:**

1 INTRODUÇÃO

A articulação de cartas topográficas é o processo que procura determinar qual a carta topográfica em maior escala que contém determinado ponto. Este processo ajuda a determinar a nomenclatura das cartas topográficas (código identificador da carta).

2 ARTICULAÇÃO DE CARTAS DE FOLHAS TOPOGRÁFICAS

O Sistema Cartográfico Nacional adota o Esquema de Articulação de Folhas de Cartas cujas medidas são múltiplas uma da outra da maior para a menor escala. Esta articulação, que é de Lei nas escalas entre 1/1.000.000 e 1/25.000 foi ampliada até 1/500, aprovada pela Cartografia Aeronáutica – DEPV.

A convenção de Londres de 1909, ao abordar o problema de articulação de folhas da Carta Internacional, recomendou o Sistema atualmente em uso no Brasil, uma vez que adotado e regulamentado pela Diretoria de Serviço Geográfico do Exército (DSG), conta de seu Manual de Normas Gerais.

O esquema de articulação, partindo da Carta Mundial ao Milionésimo (CIM) se estende até a escala 1:25.000 limite de escala da cartografia sistemática nos termos do Decreto-Lei nº 243/67. Escalas maiores pertencem ao grupo de cartografia especial a ser regulada oportunamente, nos termos do mesmo Decreto-Lei.

Em vista disso, cartas e plantas em escalas maiores que 1:25.000, não têm articulação de folha regulamentada. Isso faz com que os órgãos produtores de cartas e plantas adotem cada um seu próprio sistema de articulação de folhas, ocorrendo dificuldade quando se necessita interligar folhas produzidas por fontes diferentes.

Os sistemas cartográficos metropolitanos vêm adotando um sistema de articulação que varia do formato quadrado até 1:25.000 para tornar-se

retangular, terminado novamente, na maior escala em formato quadrado.

Na área fundiária verifica-se atualmente a incidência de folhas, na mesma escala, de formato retangular justapostas a folhas de formato quadrado, o que dificulta a cobertura sistemática do território cartografado.

A articulação é apresentada na configuração alfanumérica e numérica bastando para tanto a entrada com as coordenadas de um ponto (latitude e longitude); o cálculo fornecerá a identificação da folha que contém o ponto desde a articulação equivalente à escala 1/1.000.000 até 1/500.

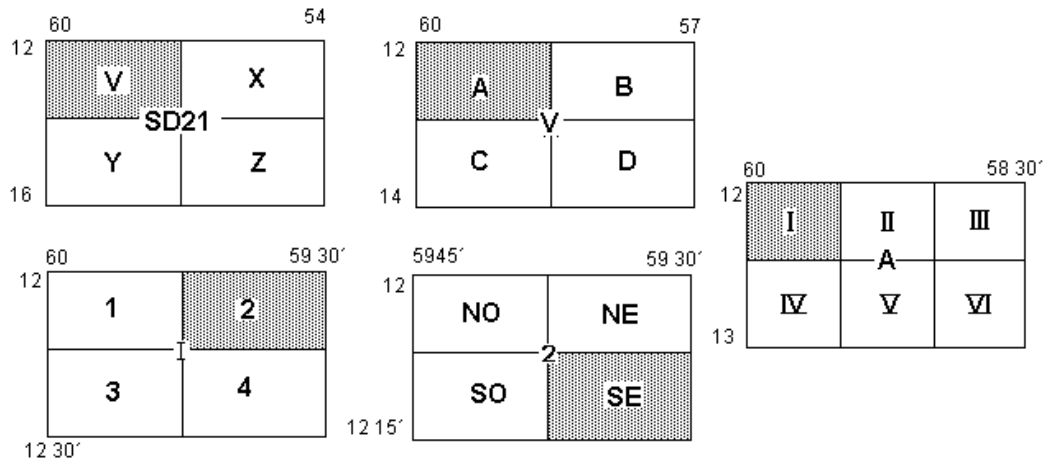
Note-se a importância desta determinação: em vez de se enquadrar um ponto numa folha de carta, especifica-se a carta ou série de cartas que contém esse ponto. A articulação de folhas de carta é única e imutável para determinado ponto, em qualquer área do globo terrestre.

FIGURA 1 - 46 CARTAS MILIONÉSIMO QUE COBREM O BRASIL

	18	19	20	21	22	23	24	25	
	78	72	66	60	54	48	42	36	30
NB			Roraima						4
NA		Pico da Neblina	Pico da Vista	Imbuém	Macapá				0
SA		Iça	Manaus	Santarém	Belém	São Luís	Fortaleza		4
SB	Javari	Juruá	Purus	Tapajós	Araguaia	Teresina	Jaguaribe	Natal	8
SC	Contamana	Rio Branco	Porto Velho	Jurema	Tocantins	Rio São Francisco	Aracaju	Recife	12
SD			Guaporé	Cuiabá	Goiás	Brasília	Salvador		16
SE				Columbá	Goiânia	Belo Horizonte	Rio de Janeiro		20
SF				Rio Apa	Pernambuco	Rio de Janeiro	Viçosa		24
SG				Assunção	Curitiba	Guape			28
SH				Uruguai	Porto Alegre				32
SI					Lagoa Mirim				

A figura a seguir ilustra o processo de articulação de uma carta ao milionésimo qualquer até a escala de 1:25.000. Note que cada região da carta tem um símbolo para indicar a posição do ponto que se deseja determinar.

FIGURA 2 - ARTICULAÇÃO DE UMA CARTA AO MILIONÉSIMO ATÉ 1/25.000



De forma resumida, os passos da figura anterior geraram cartas com as seguintes nomenclaturas:

Escala	Nomenclatura
1:1.000.000	SD21
1:500.000	SD21-V
1:250.000	SD21-V-A
1:100.000	SD21-V-A-I
1:50.000	SD21-V-A-I-2
1:25.000	SD21-V-A-I-2-SE

Veja com maiores detalhes o esquema de articulação conforme as normas de cartografia sistemática:

- A quadrícula básica corresponderá à da carta internacional ao milionésimo, com 4° de latitude a partir do Equador por 6° e longitude, tendo os mesmos meridianos centrais do sistema UTM.
- A carta de escala 1/500.000 resulta da divisão da quadrícula básica em 4 quadrículas de 2° x 3°.
- A carta de escala 1/250.000 resulta da divisão da quadrícula anterior em 4 quadrículas de 1° x 1,5°.
- A carta de escala 1/100.000 resulta da divisão da quadrícula anterior em 6 quadrículas de 30° x 30°.
- A carta de escala 1/50.000 resulta da divisão da quadrícula

anterior em 4 quadrículas de $15^{\circ} \times 15^{\circ}$.

- A carta de escala 1/25.000 resulta da divisão da quadrícula anterior em 4 quadrículas de $7,5^{\circ} \times 7,5^{\circ}$.

A norma de articulação vai somente até a escala 1:25.000 para escalas maiores não existe uma norma que diga como fazer a articulação. Mas há um modelo de articulação para chegarmos até uma escala 1:500 ou maior.

A quadrícula de $30' \times 30'$ (escala 1:100.000), pode ser dividida em 25 quadrículas $6' \times 6'$, dando origem à cartas de 1/20.000 cujas folhas são numeradas consecutivamente, da esquerda para a direita e de cima para baixo.

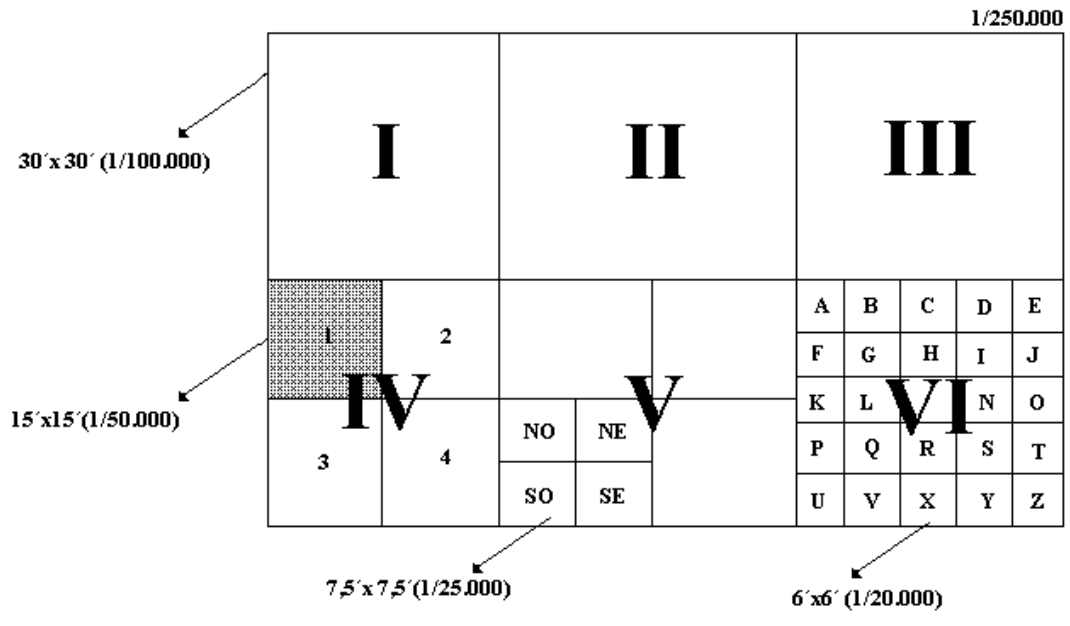
Cada quadrícula de $6' \times 6'$ (escala 1:20.000) se desmembrará em 4 quadrículas de $3' \times 3'$ correspondendo à folha de 1/10.000, a qual desmembrada em 4 outras quadrículas de $1,5' \times 1,5'$, dará a folha de 1/5.000.

Para esquematização da folha de escala 1/2.000, a quadrícula de $3' \times 3'$ (1/10.000) ou seja $180'' \times 180''$ se desmembrará em 25 folhas de $36'' \times 36''$, numeradas consecutivamente da esquerda para a direita e de cima para baixo.

Cada quadrícula de $36'' \times 36''$ se desmembrará em 4 de $18'' \times 18''$ correspondendo à folha de 1/1.000, a qual, desmembrada em 4 quadrículas de $9'' \times 9''$ dará a folha de 1/500.

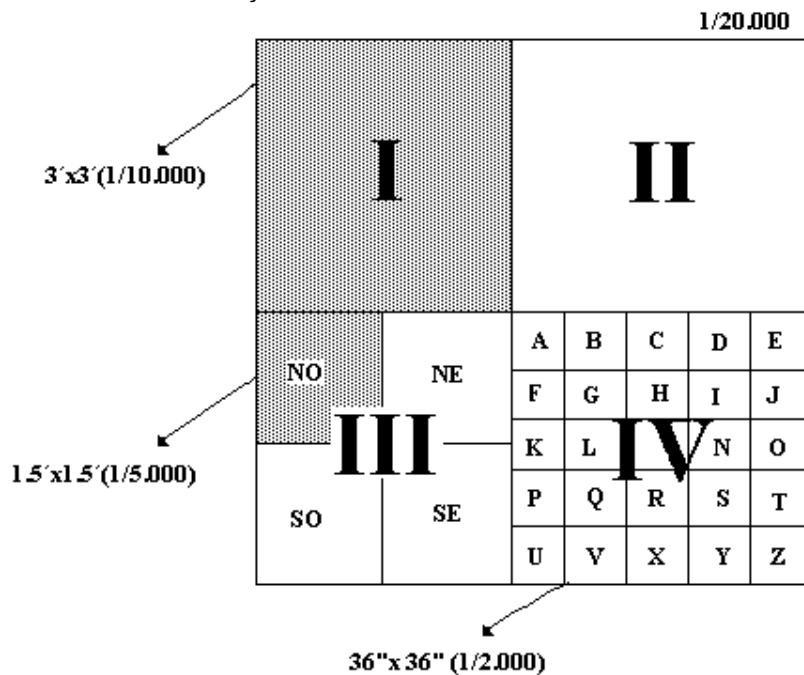
A figura a seguir mostra o esquema de articulação de uma carta na escala 1/250.000 até uma carta 1/20.000

FIGURA 3 – ARTICULAÇÃO DE 250.000 À 20.000



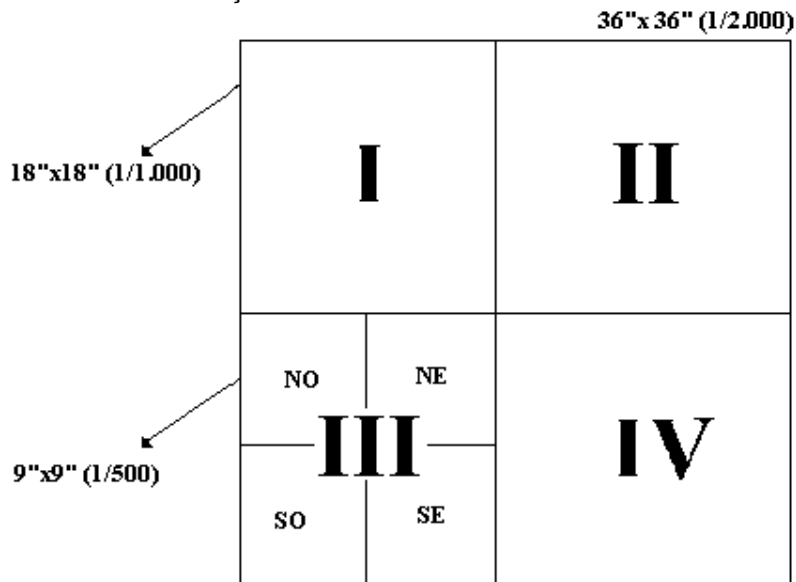
Em seguida, a próxima figura mostra o esquema de articulação de uma carta na escala 1/20.000 até 1/2.000:

FIGURA 4 – ARTICULAÇÃO DE 1/20.000 À 1/2.000



Finalmente, a próxima figura mostra a articulação de uma carta na escala 1/2.000 até a escala 1/500 (comumente usada em plantas de engenharia).

FIGURA 5 – ARTICULAÇÃO DE 1/2.000 ATÉ 1/500



Analisando o processo de articulação das cartas percebemos uma característica comum na articulação das cartas: Para dividir uma carta utilize um fator de divisão f_x (eixo x) e um fator de divisão f_y (eixo y). E então para descobrir o fator de divisão que será utilizado para encontrar a escala da nova carta, aplique a seguinte fórmula: $F = (f_x + f_y) / 2$.

Por exemplo, considerando a figura 2: Suponha que exista uma carta na escala 1/250.000 se dividirmos esta carta em 6 partes distribuídas em 3 colunas (f_x) e 2 linhas (f_y). O fator de divisão é 2,5; como temos uma carta na escala 1/250.000 o resultado será 6 cartas na escala 1/100.000.

O processo de articulação de cartas é utilizado como base para determinar a nomenclatura de uma carta topográfica. A tabela a seguir mostra um pequeno roteiro para determinar a nomenclatura de uma carta, dependendo da escala que ela se encontra.

Folha	Como fazer a nomenclatura
1/1.000.000	1 grupo de 4 dígitos
	Exemplo: SF23

1/500.000	<p>1 grupo de 4 dígitos</p> <p>1 traço seguido de uma letra(V,X,Y,Z)</p> <p>Exemplo: SF23-X</p>
1/250.000	<p>1 grupo de 4 dígitos</p> <p>1 traço seguido de uma letra(A,B,C,D)</p> <p>Exemplo: SF23-XC</p>
1/100.000	<p>4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondente à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 1 dígito (I,II, ...) correspondente à articulação 1/100.000.</p> <p>Exemplo: SF23-XC-V</p>
1/50.000	<p>1 grupo de 4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondente à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 1 dígito (I,II, ...) correspondente à articulação 1/100.000.</p> <p>1 grupo de um dígito (1,2,...) correspondente à articulação de 1/50.000</p> <p>Exemplo: SF23-XC-V3</p>
1/25.000	<p>1 grupo de 4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondentes à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondentes à articulação 1/100.000 e 1/50.000.</p> <p>2 dígitos correspondentes à articulação 1/25.000</p>

	(NO,NE,SO,SE) Exemplo: SF23-XC-V3NO
1/20.000	<p>1 grupo de 4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondentes à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 4 dígitos correspondentes à articulação 1/100.000, 1/50.000 e 1/25.000.</p> <p>1 traço seguido de um grupo de 1 dígito correspondente à articulação 1/20.000</p> <p>Exemplo: SF23-XC-V3NO-X</p>
1/10.000	<p>1 grupo de 4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondentes à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 4 dígitos correspondentes à articulação 1/100.000, 1/50.000 e 1/25.000.</p> <p>1 traço seguido de um grupo de 3 dígitos correspondentes à articulação 1/20.000 e 1/10.000 (I,II,III,...).</p> <p>Exemplo: SF23-XC-V3NO-XII</p>
1/5.000	<p>1 grupo de 4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondentes à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 4 dígitos correspondentes à articulação 1/100.000, 1/50.000 e 1/25.000.</p> <p>1 traço seguido de um grupo de 5 dígitos correspondentes à articulação 1/20.000, 1/10.000 e 1/5.000(NO,NE,SO,SE).</p> <p>Exemplo: SF23-XC-V3NO-XIINO</p>

1/2.000	<p>1 grupo de 4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondentes à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 4 dígitos correspondentes à articulação 1/100.000, 1/50.000 e 1/25.000.</p> <p>1 traço seguido de um grupo de 5 dígitos correspondentes à articulação 1/20.000, 1/10.000 e 1/5.000.</p> <p>1 traço seguido de um grupo de 3 dígitos correspondentes à articulação 1/2.000 (I,II,III,...).</p> <p>Exemplo: SF23-XC-V3NO-XIINO-V</p>
1/1.000	<p>1 grupo de 4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondentes à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 4 dígitos correspondentes à articulação 1/100.000, 1/50.000 e 1/25.000.</p> <p>1 traço seguido de um grupo de 5 dígitos correspondentes à articulação 1/20.000, 1/10.000 e 1/5.000.</p> <p>1 traço seguido de um grupo de 3 dígitos correspondentes à articulação 1/2.000 e 1/1.000 (1,2,3...).</p> <p>Exemplo: SF23-XC-V3NO-XIINO-V1</p>
1/500	<p>1 grupo de 4 dígitos correspondentes à identificação da carta ao milionésimo.</p> <p>1 traço seguido de um grupo de 2 dígitos correspondentes à articulação 1/500.000 e 1/250.000.</p> <p>1 traço seguido de um grupo de 4 dígitos correspondentes à articulação 1/100.000, 1/50.000 e 1/25.000.</p> <p>1 traço seguido de um grupo de 5 dígitos correspondentes à</p>

articulação 1/20.000, 1/10.000 e 1/5.000.

1 traço seguido de um grupo de 4 dígitos correspondente à
articulação 1/2.000, 1/1.000 e 1/500 (NO,NE,SO,SE).

Exemplo: SF23-XC-V3NO-XIINO-V1NE

3 REFERÊNCIAS

VIEIRA A. J. B.; CARVALHO C. A. P.; SLUTER C. R. **Modelagem Espaço-Temporal De Fenômenos Topográficos: Uma Revisão Didática**. Anais do Simpósio Brasileiro de Geomática, Presidente Prudente - SP, 9-13 de julho de 2002.

LISBOA FILHO, J. **Introdução ao SIG – Sistema de Informações Geográficas**. Porto Alegre: CPGCC da UFRGS, 1995.

CHRISMAN, N. **Exploring Geographic Information Systems**. New York: John Wiley & Sons, 1997.

CARVALHO F. R. **Articulação Sistemática de Folhas de Cartas**. INFORMATIVO COCAR. ANOIII, nº especial CGP05, 1985.

NATIONAL CENTER FOR GEOGRAPHIC INFORMATION AND ANALYSIS **NCGIA Core Curriculum**. Edited by Michael F. Goodchild and Karen K Kemp, Santa Barbara: University of California, 1990

APÊNDICE II – TUTORIAL SOBRE MAPEAMENTO OBJETO-RELACIONAL

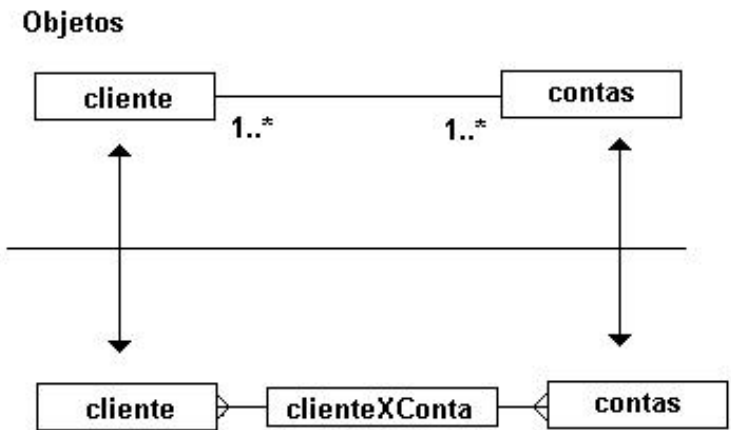
INTRODUÇÃO

APRESENTAÇÃO

Atualmente o mapeamento de objetos para um banco de dados relacional é uma tarefa comum e um tanto tedioso. Já que o desenvolvedor teria que “refazer” a modelagem que foi feita orientada a objetos para uma modelagem relacional. É para facilitar esta tarefa e também ter facilidade de testes e integração que se utiliza os conhecidos padrões. Existem vários tipos de padrões, os mais conhecidos são os padrões de projeto, mas existem padrões de análise, padrões de arquitetura, etc. Os padrões facilitam pois já foram utilizados por outros desenvolvedores, testados e tem suas vantagens e desvantagens já conhecidas.

O maior problema no mapeamento de objetos em tabelas é o mapeamento dos relacionamentos entre os objetos. Principalmente devido à diversidade de forma que um objeto pode assumir desde um simples atributo até uma lista de objetos complexos.

Algumas vezes quando representamos objetos em tabelas o modelo pode ser alterado dependendo do tipo de relacionamento entre os objetos. Por exemplo uma associação entre dois objetos em que ambos podem ter várias ocorrências em ambos os lados.



Relacional

FIG. 1 – DIFERENÇA ENTRE AS DUAS MODELAGENS

As diferenças surgem no processo de mapeamento devido às diferenças paradigmáticas entre estas duas representações:

Tabelas	Classes
Definem registros	Definem Objetos
Registros são compostos de campos	Objetos são compostos de atributos
Usa chave estrangeira para referenciar outros registros	Usa ponteiros para referenciar outros objetos
Os Tipos são estáticos	Os Tipos são Dinâmicos
Dificuldade para representar relacionamentos complexos	Facilidade para representar relacionamentos complexos
Possuem campos chave	Não possuem chave, mas ID
Todas as linhas têm os mesmos atributos	Suporta representações heterogêneas

Visto isto como representar objetos em tabelas ? Geralmente para representar um objeto em uma tabela siga os seguintes passos básicos:

1. Defina uma tabela para cada classe persistente
2. Defina colunas correspondentes aos tipos de dados da classe
3. Se a classe tiver relacionamentos, defina colunas para as chaves estrangeiras das tabelas que contém os outros objetos

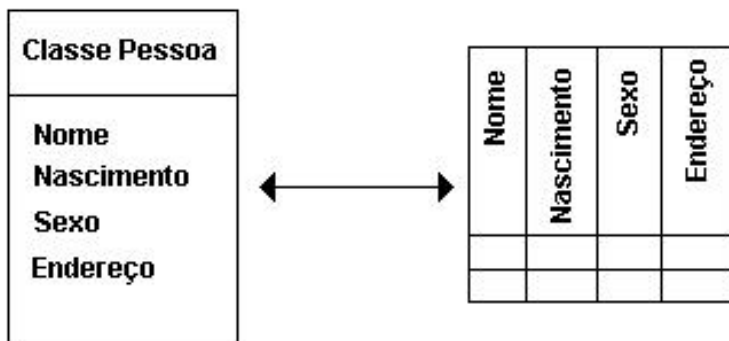


FIGURA 2 – MAPEAMENTO TRIVIAL

A próxima fase é conseguir manter a identidade dos objetos em banco de dados relacional e conseqüentemente identificar uma chave primária para a tabela. A solução mais simples criar um identificador (ID) para cada objeto e utilizá-lo como chave primária. Geralmente utiliza-se uma seqüência numérica.

ID	NOME	NASCIMENTO	SEXO	ENDEREÇO
6	JOHNATAN	28/06/80	M	HENRIQUE CORREIA, 440
7	DANIEL	26/07/81	F	JOSE MERHY, 1680

Uma prática comum na programação orientada objetos é a utilização de tipos complexos, ou seja, tipos além dos tipos básicos. Isto é um problema na representação de objetos em tabelas. Para resolver este problema geralmente utiliza-se um OID (Object Identifier).

Geralmente uma String em linguagem de programação é mapeada para o tipo *Char* em banco de dados. Para tipos complexos siga os seguintes passos:

1. Represente o tipo em uma tabela
2. Associe cada objeto desta tabela a um identificador (OID)
3. Adicione uma coluna para cada tipo que não for do tipo básico
4. Nesta coluna guarde o OID do que referencia o objeto complexo

5. Declare esta coluna como chave estrangeira

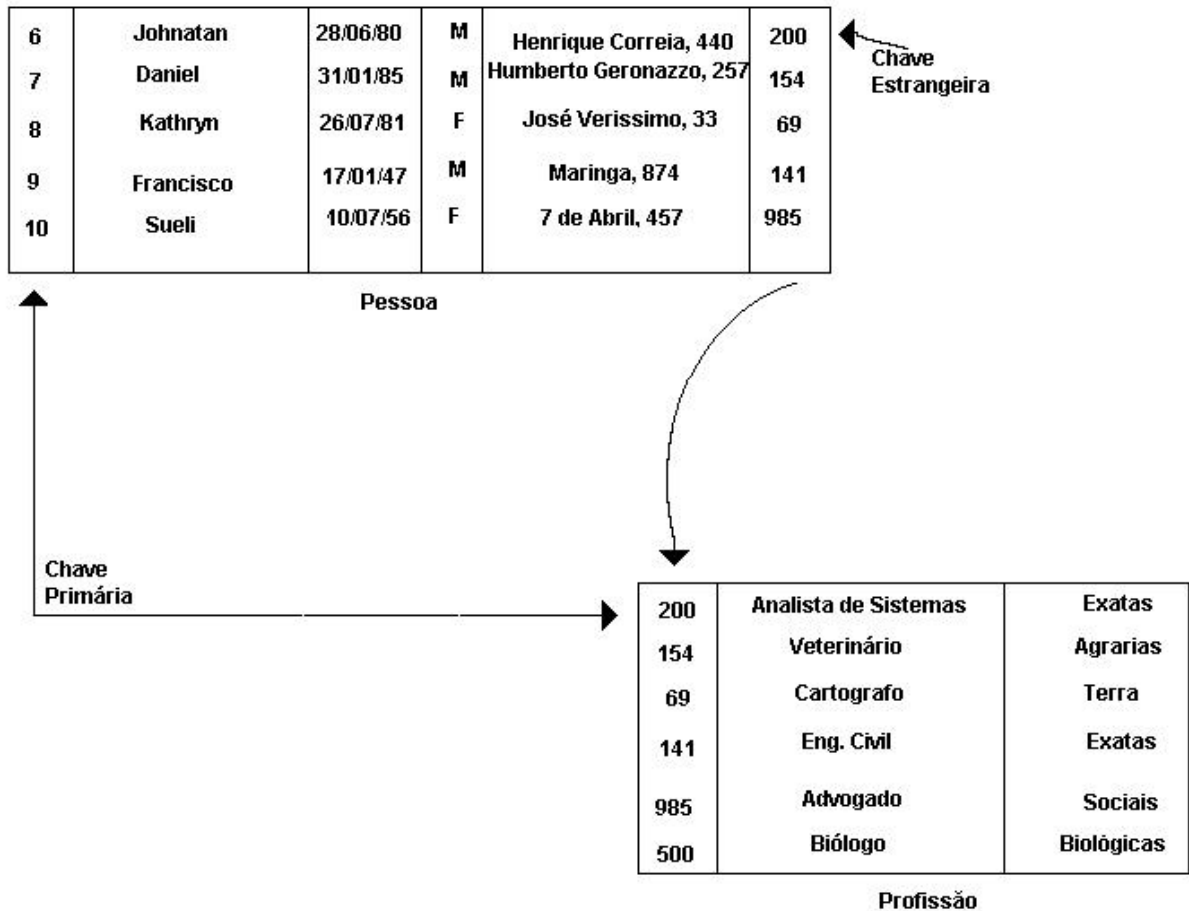


FIGURA 3 – UTILIZAÇÃO DE OID

O último passo e talvez o mais complexo é a representação dos relacionamentos dos objetos em um modelo relacional. Um relacionamento pode variar tanto em tipo como herança, composição e associação quanto na multiplicidade dos relacionamentos como 1 para 1, 1 para muitos e muitos para muitos.

3.1 RELACIONAMENTO UM PARA UM

É o mais trivial de ser mapeado. Simplesmente utiliza chave estrangeira na tabela. Para referenciar os objetos. No relacionamento um para um apenas um objeto ou registro destino é relacionado para cada objeto ou

registro destino.

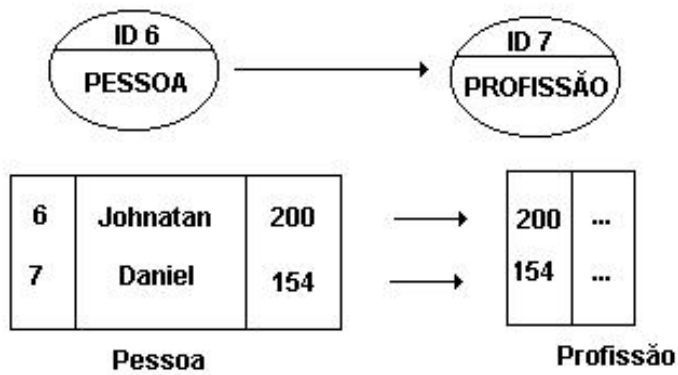


FIGURA 4 – RELACIONAMENTO 1 PARA 1

3.2 RELACIONAMENTO UM PARA MUITOS

O relacionamento um para muitos é similar ao relacionamento um para um. A única diferença é que a ocorrência de valores no destino pode ser uma coleção de objetos ou de registros.

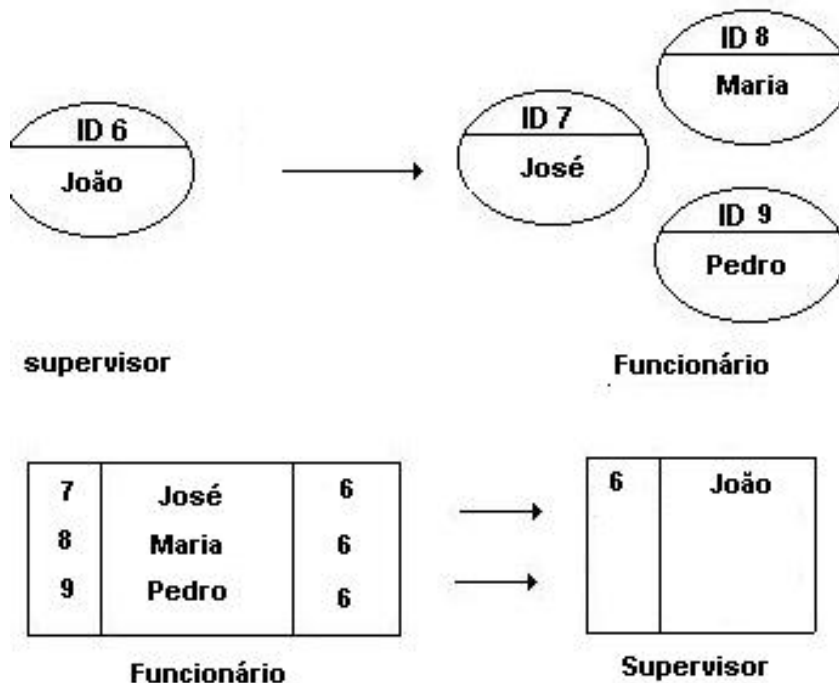


FIGURA 5 – RELACIONAMENTO 1 PARA MUITOS

3.3 RELACIONAMENTO MUITOS PARA MUITOS

O relacionamento muitos para muitos é o que mais diferencia o modelo objetos do modelo relacional, isto pelo aparecimento de uma nova entidade que representa a associação. Esta nova entidade contém principalmente as chaves primárias das tabelas associadas. Cada registro na tabela associativa representa uma combinação entre cada membro dos relacionamentos.

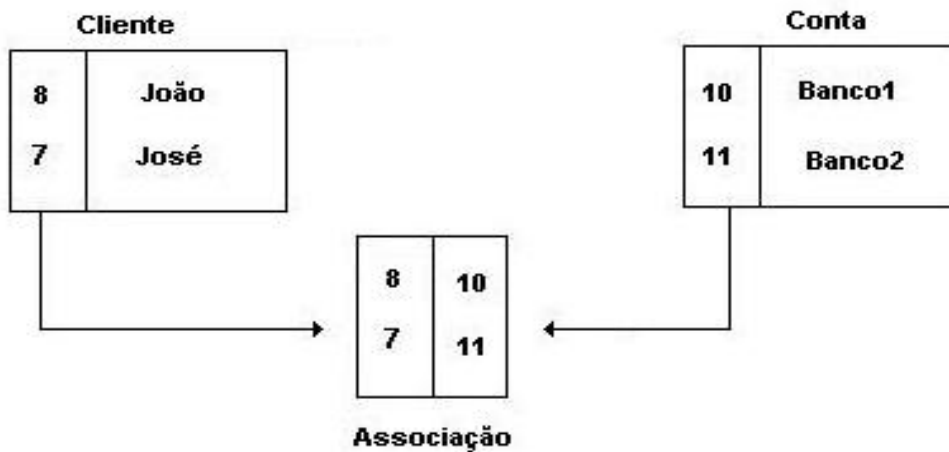


FIGURA 6 – RELACIONAMENTO MUITOS PARA MUITOS

3.4 COLEÇÕES HETEROGÊNEAS

Outro problema na representação de objetos para relacional é a capacidade dos objetos de utilizar coleções heterogêneas de dados, ou seja, de tipos diferentes de dados. Uma solução é criar um relacionamento que contenha o tipo do objeto que contém o dado e o identificador do objeto.

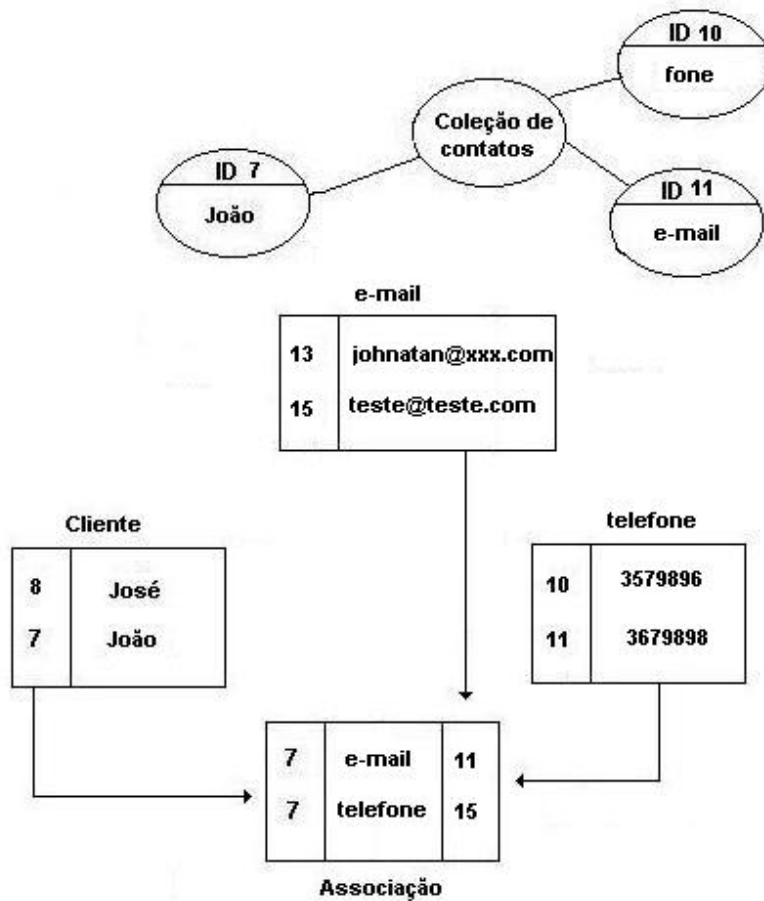


FIGURA 6 – COLEÇÕES HETEROGÊNEAS

3.5 HERANÇA

O relacionamento de herança é um tipo de relacionamento especial entre os objetos em que todas as características de uma classe são copiadas para suas subclasses.

A representação de herança de objetos em banco de dados relacional geralmente segue 3 modelos básicos:

- Vertical
- Horizontal
- Com Filtros

No mapeamento no modelo vertical siga os passos básicos:

- Cada classe deve ser mapeada em uma tabela.

- Adicione um campo identificador(ID) para cada tabela.
- Nas tabelas que representam as subclasses deve-se adicionar uma coluna que armazenará uma referência (chave estrangeira) para a tabela que representa a super classe

Veja o exemplo a seguir de como executar os passos anteriores:

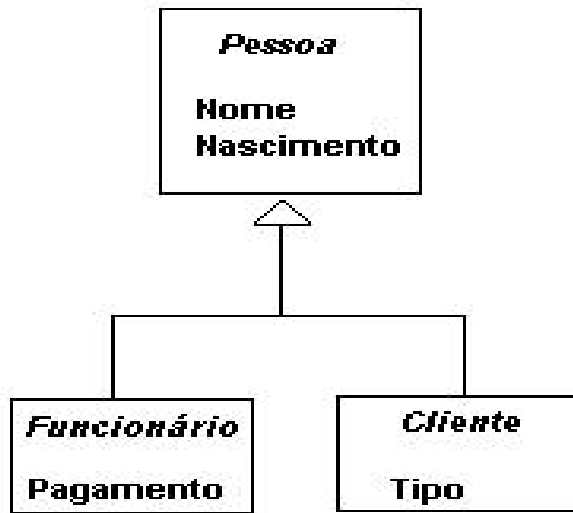


FIGURA 7 – ARVORE DE HERANÇA

A chave primária da tabela que representa a superclasse é copiada para as tabelas que representam as subclasses como uma chave estrangeira (campo Pessoa_ID).

Pessoa

Pessoa_ID	Int
Nome	Char(20)
Nascimento	Date/Time

Funcionário

ID	Int
Pessoa_ID	Int
Pagamento	Real

Cliente

ID	Int
Pessoa_ID	Int
Tipo	Char(20)

A vantagem de utilizar esta estratégia é que uma alteração na tabela que representa a super classe, por exemplo a tabela pessoa se reflete para todas as outras tabelas que a utilizam, economizando esforço na alteração da base de dados.

A desvantagem desta estratégia é que as consultas ficam mais complexas para a reconstrução do objeto. Sendo necessário a utilização de junções entre as tabelas (*joins*). O código abaixo ilustra o comando SQL necessário para recuperar os atributos de um funcionário que possua o identificador igual a 100.

```
SELECT p.ID, p.Nome, p.Nascimento, f.Pagamento
FROM PESSOA p, FUNCIONÁRIO f
WHERE f.Pessoa_ID=p.Pessoa_ID AND p.Pessoa_ID=100
```

O mapeamento horizontal é similar ao mapeamento vertical. A diferença é que os atributos da superclasse são replicados em suas subclasses no mapeamento. No nosso exemplo os campos da tabela Pessoa são replicados para a tabela Funcionário e Cliente. Então a tabela Pessoa é removida já que não é mais necessária.

O mapeamento vertical elimina a necessidade de junções para recompor objetos, mas em contra partida o esforço de manutenção da base é maior em comparação com o mapeamento vertical

Funcionário

ID	Int
Nome	Char(20)
Nascimento	Date/Time
Pagamento	Real

Cliente

ID	Int
Nome	Char(20)
Nascimento	Date/Time
Tipo	Char(20)

A outra estratégia conhecida como mapeamento com filtros. Segue o mesmo princípio do mapeamento horizontal. No mapeamento com filtros existe apenas uma tabela que contém todos os atributos da super classe e das suas sub classes. Mas além de todos os atributos das classes existe um campo chamado *Tipo* (no nosso exemplo *Tipo_Pessoa*) que indica que tipo de objeto está sendo representado por aquele registro.

Pessoa

ID	Int
Nome	Char(20)
Nascimento	Date/Time
Pagamento	Real
Tipo_Cliente	Char(20)
<i>Tipo_Pessoa</i>	Int

A desvantagem desta abordagem é que a tabela está totalmente desnormalizada, isto é, a redundância dos dados é alta. A vantagem é que a reconstrução de objetos é extremamente simples.

Utilizando esta abordagem é necessário incluir em todos os comandos SQL que utilizem esta tabela um filtro que indica que tipo de objeto deseja-se recuperar, já que nesta tabela existem vários tipos de objetos semelhantes, mas não iguais. No nosso exemplo convencionou-se que o valor '1' para o campo Tipo_Pessoa indica uma pessoa que é Funcionária.

```
SELECT ID, Nome, Nascimento, Pagamento  
FROM Pessoa WHERE Tipo_pessoa=1
```

3.6 COMPOSIÇÃO

O relacionamento de composição é freqüente em um modelo de objetos e precisa ser mapeado para o modelo relacional. A estratégia básica para o mapeamento da agregação de objetos fundamenta-se no uso de chaves estrangeiras entre as tabelas.

Para ilustrar a utilização da estratégia se utilizará um exemplo básico para explicar o uso de agregação. O "Exemplo da Nota Fiscal" é fácil de ser entendido, um modelo de classes é mostrado abaixo para ilustrar isso:



FIGURA 8 – EXEMPLO DE AGREGAÇÃO

Para fazer o mapeamento do exemplo acima siga os seguintes passos:

- Cada classe é mapeada em uma tabela
- Crie um identificador (ID) para cada tabela
- Crie uma coluna para chave estrangeira nas tabela em que a multiplicidade for maior que 1 (no nosso caso as tabelas Pedido e ItemDoPedido)
- Guarde na coluna criada o ID do objeto que tem multiplicidade igual a 1 (no nosso caso o valor de ID_Cliente na tabela Pedido e de ID_Pedido em ItemDoPedido)

Cliente

Cliente_ID	Int
Nome	Char(30)

Pedido

Pedido_ID	Int
Cliente_ID	Int
Vendedor	Char(30)
Data	Date/Time

Item do Pedido

Pedido_ID	Int
Item_ID	Int
Descrição	Char(40)
Quantidade	Real
Preço	Real

É necessário observar antes de fazer o mapeamento a multiplicidade das associações. Dependendo dos relacionamentos pode ser necessário aplicar técnicas de mapeamento como: 1 para 1, 1 para muitos ou muitos para muitos. Vistos anteriormente.

3.7 UTILIZANDO CAMPOS BLOB

Uma técnica utilizada atualmente é utilizar campos do tipo BLOB (Binary Large Objects). Este padrão serve para representar relacionamentos como associações, agregação e herança. Esta estratégia é usada em problemas não convencionais de mapeamento de objetos.

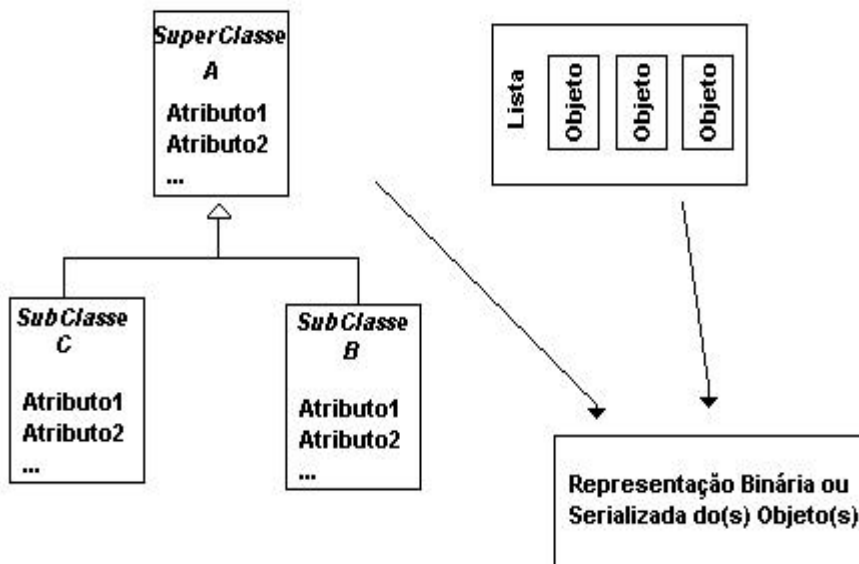


FIGURA 9 – REPRESENTAÇÃO DE OBJETOS EM FORMATO BINÁRIO

A estratégia baseia-se em dois passos principais:

1. representação da classe e sua estrutura no formato binário, geralmente utilizando um arquivo
2. Importar esta representação para um campo do tipo BLOB em uma tabela

É importante combinar campos convencionais na tabela que irá conter a representação binária para facilitar a localização do objeto que está sendo representado.

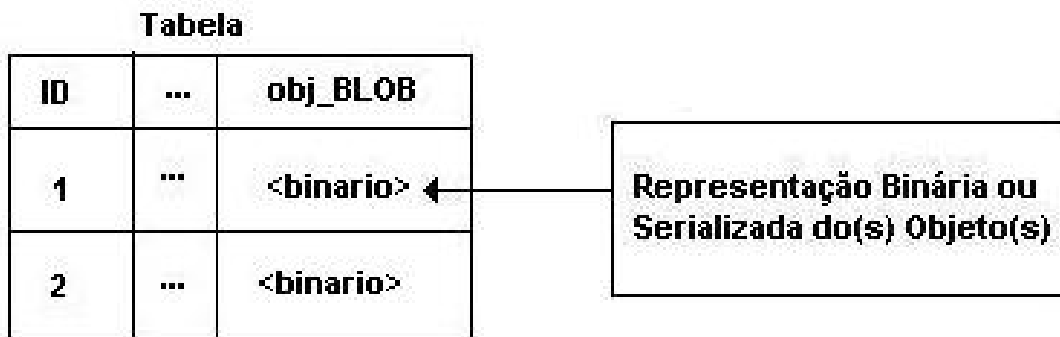


FIGURA 10 – ARMAZENAMENTO DE ARQUIVOS BINÁRIOS EM BANCO DE DADOS RELACIONAIS

As desvantagens desta estratégia são principalmente a perda de performance ocorrida pela operação de leitura e gravação do campo BLOB e também pelo aumento do espaço significativo no banco de dados principalmente se o sistema gerenciador não possuir a capacidade de campos BLOB de tamanho variável.

As principais vantagens desta abordagem são simplificação do mapeamento ocasionando também a diminuição do processamento exigido pelas consultas e junções complexas no banco de dados. E também a combinação de propriedades OODB com banco de dados relacionais dada a facilidade do campo BLOB de armazenar estruturas de objetos complexas.

REFERÊNCIAS BIBLIOGRÁFICAS

RONIN INTERNATIONAL. **Mapping Objects to Relational Databases**. Disponível em: < <http://www.ambysoft.com/mappingobjects.pdf>>. Acesso em 27 jan. 2003.

KELLER, Wolfgang. **Mapping Objects to Tables**, A Pattern Language. Disponível em: < <http://www.objectarchitects.de/>>. Acesso em 27 jan. 2003.

BROWN, K. e Whitenack, B.1996. **Crossing Chasms. Pattern Languages of Program Design** vol. 2. Reading, MA. Addison-Wesley

OBJECTMATTER,Inc. **Object-Relational Mapping Basics & Strategies**. Disponível em: < <http://www.objectmatter.com/>>. Acesso em 28 jan. 2003.