



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Rogério Luís Ribeiro Borba

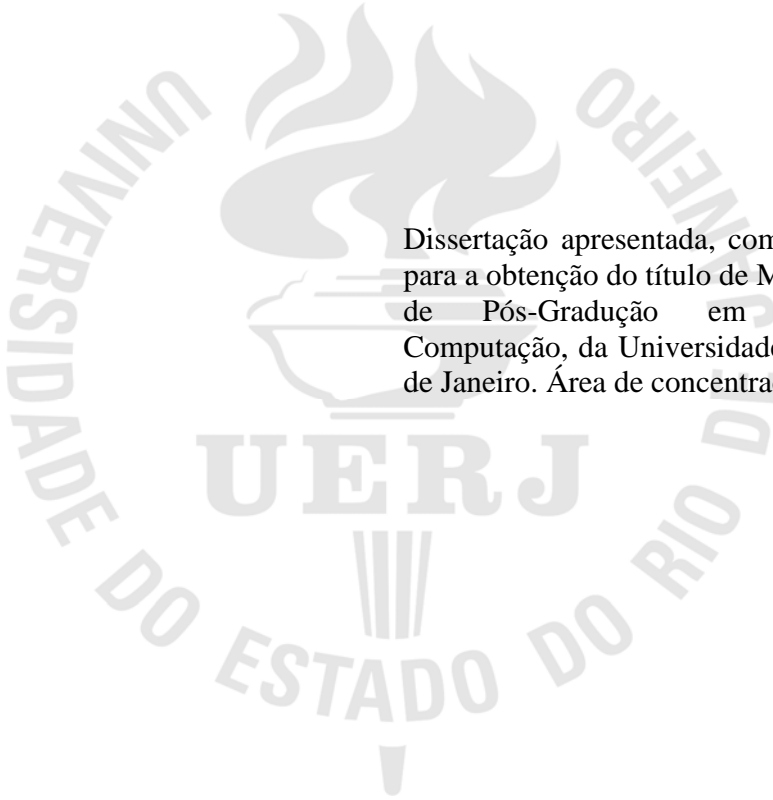
**Restituição fotogramétrica segundo o padrão da estruturação de dados
Geospaciais Vetoriais no ambiente E-FOTO**

Rio de Janeiro

2009

Rogério Luís Ribeiro Borba

**Restituição fotogramétrica segundo o padrão da estruturação de dados
Geospaciais Vetoriais no ambiente E-FOTO**



Dissertação apresentada, como requisito parcial para a obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia de Computação, da Universidade do Estado do Rio de Janeiro. Área de concentração Geomática.

Orientador: Prof. Dr. Jorge Luís Nunes Silva e Brito
Coorientador: Prof. Dr. Guilherme Lucio Abelha Mota

Rio de Janeiro

2009

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

B726 Borba, Rogério Luís Ribeiro.
Restituição fotogramétrica segundo o padrão da estruturação de dados geoespaciais vetoriais no Ambiente E-FOTO / Rogério Luís Ribeiro Borba. - 2009.
113 f.

Orientador: Jorge Luís Nunes Silva e Brito.
Coorientador: Guilherme Lucio Abelha Mota
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Fotogrametria – Teses. 2. Software livre – Teses. 3. Engenharia de Computação – Teses. I. Silva e Brito, Jorge Luís Nunes. II. Universidade do Estado do Rio de Janeiro. III. Título.

CDU 004.4:528.7

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação, desde que citada a fonte.

Assinatura

Data

Rogério Luís Ribeiro Borba

**Restituição fotogramétrica segundo o padrão da estruturação de dados
Geospaciais Vetoriais no ambiente E-FOTO**

Dissertação apresentada, como requisito parcial para a obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia de Computação, da Universidade do Estado do Rio de Janeiro. Área de concentração Geomática.

Aprovado em: 18 de março de 2009.

Banca Examinadora:

Prof. Dr. Jorge Luís Nunes Silva e Brito (Orientador)

Faculdade de Engenharia - UERJ

Prof. Dr. Guilherme Lucio Abelha Mota (Coorientador)

Faculdade de Engenharia - UERJ

Prof. Dr. Orlando Bernardo Filho

Faculdade de Engenharia - UERJ

Prof. Dr. Alberto Tavares da Silva

Centro Universitário Carioca - UniCarioca

Prof. Dr. Claudio João Barreto dos Santos

Instituto Brasileiro de Geografia e Estatística - IBGE

Rio de Janeiro

2009

DEDICATÓRIA

À minha esposa Jaqueline.

Ao meu pequeno Luís Felipe que nasceu durante este trabalho.

Aos meus pais.

Pela compreensão, paciência, incentivo e ajuda

AGRADECIMENTOS

Ao meu orientador Jorge Luís Nunes Silva e Brito

Ao meu co-orientador Guilherme Lucio Abelha Mota

Que sempre me orientaram com muita presteza, dedicação, competência e que tornaram esta dissertação possível.

À Universidade do Estado do Rio de Janeiro por mais esta oportunidade concedida para meu crescimento e aperfeiçoamento acadêmico e profissional.

Aos professores, funcionários e amigos do curso de pós-graduação em Engenharia de Computação área de concentração Geomática.

Ao professor Alexandre Corrêa que sempre vi como um grande exemplo.

Aos colegas de trabalho do IBGE que de alguma forma me ajudarem e motivaram a realizar este mestrado.

Ao meu irmão Marcos e cunhada Adriana pela ajuda dada.

RESUMO

Borba, Rogério Luís Ribeiro. *Restituição fotogramétrica segundo o padrão da estruturação de dados geoespaciais vetoriais no ambiente E-FOTO*. 113 f. Dissertação (Mestrado em Engenharia de Computação) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2009.

No mês de outubro de 2007 a Comissão Nacional de Cartografia (CONCAR) lançou uma norma cartográfica composta de especificações técnicas para Estruturação de Dados Geoespaciais Digitais Vetoriais (EDGV) para a realização do mapeamento topográfico sistemático no Brasil. No mês de novembro de 2008 foi publicado o Decreto nº 6666, que institui a Infra-Estrutura Nacional de Dados Espaciais e ratifica o uso das normas homologadas pela CONCAR, pelas instituições Federais. Este trabalho realiza dois requisitos, a saber: (a) fornece um esquema relacional alinhado as especificações da EDGV e (b) implementa um protótipo, com vistas a sua utilização no processo de restituição fotogramétrica nas instituições públicas responsáveis pelo mapeamento topográfico no Brasil. A implementação é realizada em ambiente de software livre. Também é importante salientar o caráter educacional da plataforma de software a ser implementada, para que o seu uso seja também direcionado para o ensino teórico e prático da fotogrametria digital nas instituições de ensino e pesquisa.

Palavras-chave: Fotogrametria; Restituição fotogramétrica; E-FOTO; EDGV; CONCAR; GNU/GPL – Software Livre.

ABSTRACT

The National Commission of Cartography (CONCAR) in October 2007 launched a pattern composed of standard specifications for Structuring of Digital Geospatial Data Vector (EDGV) for systematic topographic mapping production in Brazil. In November of 2008 it was published the Decree n° 6666, that established the National Spatial Data Infrastructure and ratified the use of standards approved by CONCAR. This work accomplishes two goals: (a) to provide a relational schema specification aligned with the EDGV and; (b) to provide an implementation of a prototype, for using in the photogrammetric restitution of topographic mapping in Brazil. The implementation is performed in an environment of free software, according to the GNU/GPL approach. It is also important to emphasize the educational character of the software to be implemented so that its use is also directed to the theoretical and practical studies of digital photogrammetry in the institutions of education and research.

Keywords: Digital photogrammetry; Photogrammetric mapping; E-FOTO; EDGV; CONCAR; GNU/GPL – Free Software.

LISTA DE ILUSTRAÇÕES

Figura 1 – Visão geral da proposta da dissertação	188
Figura 2 – Arquitetura de uma IDE.....	21
Figura 3 – Pacote com os subpacotes (categorias).....	22
Figura 4 – Diagrama de classes Administração Pública. Versão 2.0 da EDGV	24
Figura 5 – Representação de um açude	27
Figura 6 – Modelo geométrico do SFA.....	32
Figura 7 – Interface da classe Geometry	34
Figura 8 – Relacionamento entre dois polígonos	36
Figura 9 – Arquitetura para tipos pré-definidos da SQL.....	39
Figura 10 – Feições com coordenadas.....	40
Figura 11 – Feições com tipos geométricos	42
Figura 12 – Tabelas no PostgreSQL/PostGIS e o esquema da SFA	45
Figura 13 – Classe de: (a) Feição, (b) Agregadora e (c) Domínio	49
Figura 14 – Mapeamento de uma classe para tabela	49
Figura 15 – Atributos de uma classe	51
Figura 16 – Mapeamento do atributo para coluna.....	51
Figura 17 – Mapeamento de uma hierarquia de classes em uma única tabela	53
Figura 18 – Mapeamento de uma classe concreta para tabela.....	54
Figura 19 – Mapeamento de classe por tabela.....	55
Figura 20 – Abordagem Orientada a Metadados.....	56
Figura 21 – Mapeamento de uma hierarquia com herança múltipla para uma tabela	59
Figura 22 – (a) Associação simples; (b) agregação e; (c) composição	60
Figura 23 – Relacionamento espacial.....	60
Figura 24 – Multiplicidade e direcionalidade.....	62
Figura 25 – Implementação dos diversos tipos de associação segundo o modelo relacional.....	63
Figura 26 – Relacionamento espacial entre Trecho_Drenagem e Massa_Dagua	64
Figura 27 – Relacionamento espacial entre trecho de drenagem e corredeira	65

Figura 28 – Classe Area_lazer com representação de polígono.....	66
Figura 29 – Elemento com mais de uma representação geométrica	67
Figura 30 – A Falta do tipo <i>Geometry</i>	68
Figura 31 – A Classe Recife mapeada para três tabelas.....	69
Figura 32 – Elemento agregador com seus agregados	70
Figura 33 – Relacionamento entre as tabelas agregadora e suas agregadas.....	70
Figura 34 – Classe de Domínio com seus valores e tabela com suas linhas	71
Figura 35 – Diagrama de caso de uso da Restituição fotogramétrica Digital na EDGV	75
Figura 36 – Diagrama de classes de integração com o modelo de dados.....	77
Figura 37 – Classes de visão e classes de metamodelo	78
Figura 38 – Serviços oferecidos pela classe FactoryManagerDatabase	79
Figura 39 – Classe MetaTable.....	80
Figura 40 – Classe MetaColumn	81
Figura 41 – Classe WKT	82
Figura 42 – Modelo Estereoscópico Carregado no E-FOTO	83
Figura 43 – Relação dos elementos cartográficos restituídos no E-FOTO	85
Figura 44 – Passos para classificação. (a) EDGVWindow e; (b) ConnectionDialog.....	85
Figura 45 – Resultado da consulta na tabela trecho_rodoviario.....	86
Figura 46 – Resultado da consulta na tabela ponto_rod_ferrovuario.....	86
Figura 47 – Resultado da consulta na tabela travessia_pedestre	86
Figura 48 – Resultado da consulta na tabela trecho_ferrovuario	86
Figura 49 – Resultado da consulta na tabela edif_metro_ferrovuario.....	87
Figura 50 – Resultado da consulta na tabela posto_combustivel	87
Figura 51 – Resultado da consulta na tabela edif_ensino.....	87
Figura 52 – Resultado da consulta na tabela arquibancada.....	87
Figura 53 – Resultado da consulta na tabela pto_est_med_fenomenos	87
Figura 54 – Feições armazenadas no PostgreSQL e Visualizadas no QGIS.....	88
Figura 55 – Travessia de pedestres que cruzam trecho ferroviário	88
Figura 56 – Distância entre edificação e posto de combustível	89
Figura 57 – Vias rodoviárias com seus respectivos trechos rodoviários.....	89

LISTA DE TABELAS

Tabela 1 – Informações da classe Sumidouro_Vertedouro na RCO	26
Tabela 2 – Matriz de Nove Interseções de Dimensionalidade Estendida (DE-9IM)	35
Tabela 3 – Representação textual dos elementos geométricos.....	37
Tabela 4 – Esquema de geometria normalizada	41
Tabela 5 – Esquema geometria binária.....	41
Tabela 6 – Mapeamento tipo do atributo para tipo SQL	50
Tabela 7 – Comparação de Estratégias para Mapeamento de Heranças	57
Tabela 8 – Elementos cartográficos a serem restituídos e suas classificações na EDGV	84

LISTA DE ABREVIATURAS

CAD	Computer Aided Design.
CEMND	Comitê Especializado para a Estruturação da Mapoteca Nacional Digital.
COM	Component Object Model.
CONCAR	Comissão Nacional de Cartografia.
CORBA	Common Object Request Broker Architecture.
DE-9IM	Matriz Modelo de Nove Interseções de Dimensionalidade Extendida.
DSG	Diretoria de Serviço Geográfico do Exército Brasileiro.
EFD	Estação Fotogramétrica Digital Educacional.
GML	Geography Markup Language.
GNU	General Public License.
IBGE	Fundação Instituto Brasileiro de Geografia e Estatística.
IDE	Infra-Estrutura de Dados Espaciais.
INDE	Infra-Estrutura Nacional de Dados Espaciais.
MMA	Ministério do Meio Ambiente.
MTD	Mapoteca Topográfica Digital.
OGC	Open Geospatial Consortium.
ODBMS	Object Database Management System.
OMT	Object Modeling Technique.
OMT-G	Object Modeling Technique for Geographic.
OO	Orientação a Objeto.
OpenGIS	Open Geospatial Consortium.
SGBD	Sistema Gerenciador de Banco de Dados.
SGBDE	Sistema Gerenciador de Banco de Dados Espaciais.
SGBD	Sistema Gerenciador de Banco de Dados Relacional.
SGBDOO	Sistema Gerenciador de Banco de Dados Orientado a Objeto.
SFA	Simple Feature Access.
SFA	Simple Features Interface Standard.
SQL	Structured Query Language.

SQL/CLI	SQL Call Level Interface.
TBCD	Tabelas da Base Cartográfica Digital.
UERJ	Universidade do Estado do Rio de Janeiro.
UDT	Tipos de Dados definidos pelo Usuário.
UML	Unified Modeling Language.
WFS	Web Feature Service.
WKT	Well-known Text.
WKB	Well-known Binary.
WMS	Web Map Service.

SUMÁRIO

	INTRODUÇÃO	15
1	FUNDAMENTOS BÁSICOS	20
1.1	Infra-estrutura nacional de dados espaciais	20
1.2	Estruturação de dados geoespaciais digitais vetoriais (EDGV)	21
1.3	Modelo conceitual orientado a objeto da EDGV	22
1.3.1	<u>Diagrama de Classes</u>	23
1.4	Modelagem conceitual de dados geográficos (OMT-G)	24
1.4.1	<u>Diagrama de classes na OMT-G</u>	25
1.4.2	<u>Relacionamentos na OMT-G</u>	25
1.5	Relação de Classes e Objetos (RCO) da EDGV	25
1.6	Restituição	26
1.7	Sistema Gerenciador de Banco de Dados Espaciais	28
2	ARCABOUÇO TECNOLÓGICO	29
2.1	Projeto E-FOTO	29
2.2	Padrões da Open Geospatial Consortium (OGC)	30
2.2.1	<u>OGC</u>	30
2.2.2	<u>Simple Feature Access</u>	31
2.2.2.1	Parte 1 - Arquitetura Comum.....	31
2.2.2.1.1	Modelo Geométrico	32
2.2.2.1.2	Interface da classe Geometry	33
2.2.2.1.3	Operadores Topológicos	35
2.2.2.1.4	Formatos Texto e Binário	36
2.2.2.2	Parte 2 – Interface de acesso – implementação SQL	38
2.2.2.2.1	Tabelas de feições baseadas em tipos pré-definidos da SQL.....	39
2.2.2.2.2	Tabelas de feições com tipos geométricos	42
2.3	SGBD Postgre SQL	42
2.4	Cartucho Espacial Post GIS	43
2.4.1	<u>Esquema tabela de feições com geometria</u>	44

3	METODOLOGIA PARA REPRESENTAÇÃO DA EDGV EM BANCOS DE DADOS GEOGRÁFICOS	46
3.1	Seleção do sistema gerenciador de banco de dados	46
3.2	Mapeando do modelo orientado a objeto para o modelo relacional para EDGV	48
3.2.1	<u>Tipos de classe da EDGV</u>	48
3.2.2	<u>Mapeamento de classe para a tabela</u>	49
3.2.3	<u>Mapeamento dos tipos primitivos de atributos para tipos da SQL</u>	50
3.2.4	<u>Mapeamento dos atributos para colunas</u>	50
3.2.5	<u>Mapeamento de estruturas hierárquicas</u>	52
3.2.5.1	Mapeamento de toda a hierarquia de classe para uma única tabela.....	52
3.2.5.2	Mapeamento de cada classe concreta para a sua própria tabela	53
3.2.5.3	Mapeamento de cada classe para a sua própria tabela	55
3.2.5.4	Mapeamento das classes em uma estrutura genérica de tabelas	56
3.2.5.5	Comparação entre as estratégias de mapeamento	57
3.2.5.6	Herança múltipla	58
3.2.6	<u>Mapeamento dos relacionamentos</u>	59
3.2.7	<u>Relacionamentos espaciais</u>	64
3.2.8	<u>Mapeamento das classes de feição</u>	65
3.2.8.1	Feição com uma única representação.....	66
3.2.8.2	Feição com mais de uma representação	66
3.2.8.3	A falta do conceito da abstração de alguns softwares.....	67
3.2.9	<u>Mapeamento das classes agregadoras</u>	69
3.2.10	<u>Mapeamento das classes de domínio</u>	71
3.2.11	<u>Geração de scripts para criação das tabelas</u>	72
3.3	Considerações adicionais sobre a abordagem relacional da EDGV	72
4	O MÓDULO DE RESTITUIÇÃO FOTOGRAMÉTRICA DIGITAL DO E-FOTO	74
4.1	Novas demandas e novos desafios	74
4.2	Novos requisitos para o módulo estereoplotter	74
4.2.1	<u>Requisitos funcionais</u>	75

4.2.2	<u>Requisitos não-funcionais</u>	75
4.2.3	<u>Modelo de classes da extensão</u>	76
4.2.3.1	O papel de cada classe	78
4.3	Prova de conceito	82
5	CONCLUSÃO	90
	REFERÊNCIAS	92
	APÊNDICE - Modelo conceitual da EDGV simplificado.....	95

INTRODUÇÃO

No Brasil há duas instituições públicas responsáveis pelo mapeamento topográfico sistemático terrestre básico; uma militar, a Diretoria de Serviço Geográfico (DSG) do Exército Brasileiro e outra civil, a Fundação Instituto Brasileiro de Geografia e Estatística (IBGE). Até o início dos anos 90, a chamada cartografia convencional ou analógica era executada pelas duas organizações de forma padronizada, segundo especificações técnicas obrigatórias, de acordo com resoluções definidas pelos órgãos responsáveis, em obediência à legislação específica (DELOU, 2006).

A partir dos anos 90, quando a metodologia de produção adotou a tecnologia digital, com o grande crescimento dos Sistemas de Informação Geográfica (SIG), que demandavam bases cartográficas digitais, houve uma ruptura com o processo padronizado de produção cartográfico que seguia especificações técnicas comuns às duas instituições. Se por um lado a cartografia digital trouxe mais flexibilidade e grandes avanços para área, por outro lado gerou uma dissonância no processo de produção cartográfica digital. Cada instituição criou o seu próprio modelo para estruturar os elementos cartográficos contidos nos arquivos digitais na estrutura de vetor, que representavam as cartas topográficas do Mapeamento Topográfico Sistemático Terrestre. Enquanto a DSG baseava a estruturação dos elementos cartográficos nas Tabelas de Base Cartográfica Digital – TBCD, o IBGE baseava a estruturação dos elementos cartográficos na Mapoteca Topográfica Digital – MTD.

A falta de padronização dos modelos gerou diversos problemas, conforme mencionado por Delou (2006). A título de exemplo, registra-se: (a) dificuldade de compartilhamento entre cartas vetorizadas, (b) dificuldade de integrar projetos de mapeamento entre as duas instituições, (c) dificuldade de usar um SIG com dados das duas instituições e (d) baixo nível de semântica nestes modelos.

Entre os anos de 1998 e 2004, a DSG e o IBGE realizaram várias reuniões, objetivando a junção dos seus modelos MTD e TBCD. Esses esforços configuraram o início da tentativa de obtenção de um padrão único para a estrutura de dados geoespaciais vetoriais. Porém a falta de investimentos impediu nesta oportunidade a obtenção de um padrão único (CONCAR, 2007).

O Ministério do Meio Ambiente (MMA), no final de 2004, fez contato com a Comissão Nacional de Cartografia (CONCAR) com a finalidade de usar uma base cartográfica digital contínua da área da Amazônia Legal, em um de seus projetos. Devido à falta de padronização entre as cartas elaboradas pelo IBGE e pela DSG, não foi possível naquele momento disponibilizar uma base contínua da região da Amazônia Legal.

Com objetivo de criar uma base contínua da Amazônia Legal surgiu uma proposta de convênio do MMA com a DSG e o IBGE, para obtenção da base, mencionada anteriormente, em área contínua e em um mesmo padrão de estrutura de dados, ou seja, seguindo um mesmo modelo. Ciente da demanda por um modelo único e da necessidade do estabelecimento de normas e padrões para a criação de uma Infra-Estrutura Nacional de Dados Espaciais (INDE), a CONCAR solicitou que esse trabalho fosse coordenado e executado no âmbito do Comitê Especializado para a Estruturação da Mapoteca Nacional Digital (CEMND).

Contando com o apoio financeiro do MMA, no transcorrer do ano de 2005, os integrantes da DSG e do IBGE, que passaram a fazer parte CEMND, retomaram os esforços para a criação de um padrão único. Os trabalhos foram iniciados tomando-se por base no paradigma da orientação a objeto. A meta era produzir um modelo conceitual que incorporava características da TBCD (versão 1997) e da MTD (versão 1999). Preliminarmente, foi desenvolvido um modelo que resultou em uma proposta inicial de Estrutura de Dados Geoespaciais Vetoriais (EDGV).

A EDGV tornou-se uma proposta para a Mapoteca Nacional Digital (MND), versão 2005, no âmbito do CEMND. A proposta foi apresentada em reunião da plenária da CONCAR, em abril de 2006. Com este fato, a CONCAR deliberou pela sua aprovação e homologou esta versão como norma provisória a ser adotada, até que uma nova versão mais abrangente que atendesse também às necessidades de outras instituições, fosse elaborada.

Durante o ano de 2006 e até agosto de 2007, o Comitê trabalhou, com a participação de várias outras instituições, para criação de um modelo conceitual com base na versão 2005. As especificações técnicas da EDGV foram lançadas em setembro de 2007 como uma norma Nacional. Trata-se de um novo marco no mapeamento topográfico sistemático terrestre básico com um modelo único, cuja adoção pelas instituições responsáveis pelo mapeamento topográfico no Brasil é obrigatória (CONCAR, 2007).

É importante ressaltar que esta norma cartográfica faz parte da INDE, que corresponde ao conjunto de esforços para garantir bases para descoberta, avaliação e utilização de dados espaciais.

O problema

O CEMND elaborou as especificações técnicas para a EDGV, que consiste em um modelo conceitual orientado a objeto. Além de possibilitar o compartilhamento, a interoperabilidade e a disseminação de dados, ela cria novos elementos cartográficos e homogeneiza a estruturação desses elementos para o formato vetorial.

Uma vez divulgado e disponibilizado este novo padrão, as duas grandes organizações produtoras de cartografia (DSG e IBGE), assim como outras instituições, precisam realizar suas atividades de produção seguindo o modelo da EDGV. O problema é que não há ainda em suas linhas de produção plataforma de software, aplicativos e procedimentos operacionais digitais que estejam em conformidade com esta norma cartográfica. Setores de produção relacionados à fotogrametria digital e ao sensoriamento remoto necessitam de um novo arcabouço tecnológico que dê sustentação às suas atividades cotidianas. Particularmente o processo de restituição fotogramétrica necessita estar acoplado a EDGV e, para que isto seja possível, novos métodos para representar os objetos gerados a partir da restituição fotogramétrica digital precisam ser desenvolvidos.

Objetivos da Dissertação

Objetivo principal da presente dissertação é estender o módulo de restituição fotogramétrica 3D da Estação Fotogramétrica Digital (EFD) E-FOTO, a fim de viabilizar a restituição fotogramétrica digital, em consonância com o modelo conceitual da EDGV. As

feições restituídas serão armazenadas num banco de dados geográficos. A visão geral da proposta de extensão é apresentada na figura 1.

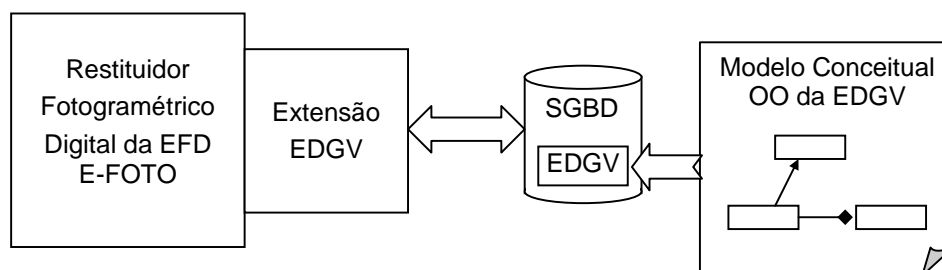


Figura 1 – Visão geral da proposta da dissertação

Na figura 1 é possível observar a extensão para a EDGV do restituidor fotogramétrico do E-FOTO, que faz a comunicação e o armazenamento das feições restituídas no SGBD. Note-se também que o modelo conceitual orientado a objeto da EDGV foi mapeado e implementado no SGBD.

A fim de realizar a restituição com qualidade no padrão da EDGV é necessário contemplar também os seguintes objetivos:

- Armazenar as feições restituídas em banco de dados geográficos relacional.
- Modularizar a arquitetura do protótipo de forma a isolar o módulo de restituição fotogramétrica da estrutura do banco de dados, trazendo flexibilidade à proposta quanto à modificações na EDGV.
- Produzir um modelo de dados aderente ao modelo conceitual orientado a objeto da EDGV para Bancos de Dados Geográficos Relacionais.
- Utilizar padrões de informação geoespacial aderentes a OGC para que seja possível ampliar o intercâmbio de dados.

Deve ser mencionado o caráter inédito da presente proposta, que torna o E-FOTO a primeira EFD adaptada à EDGV.

Organização da Dissertação

Esta dissertação está organizada da seguinte forma: o Capítulo 1 apresenta termos e conceitos que fazem parte do contexto desta dissertação; o Capítulo 2 traz os elementos tecnológicos utilizados no desenvolvimento deste trabalho. O Capítulo 3 discute e apresenta a metodologia para representação da EDGV em bancos de dados geográficos relacionais. O Capítulo 4 aborda a adaptação do módulo de restituição fotogramétrica digital do E-FOTO para conformidade com a EDGV. O Capítulo 5 traz as conclusões da pesquisa.

1 FUNDAMENTOS BÁSICOS

Neste capítulo serão abordados alguns aspectos básicos, como por exemplo, em que contexto a EDGV se insere; como esta norma está organizada e quais os elementos que fazem parte ou são utilizados na sua estruturação.

1.1 Infra-Estrutura Nacional de Dados Espaciais

O decreto nº 6.666, de 27 de novembro de 2008 (Brasil, 2008), instituiu no âmbito do Poder Executivo Federal, a Infra-Estrutura Nacional de Dados Espaciais (INDE). Uma INDE reúne: (a) um conjunto integrado de tecnologias; (b) políticas; (c) mecanismos e procedimentos de coordenação e monitoramento; (d) padrões e acordos, necessários para facilitar e ordenar a geração, o armazenamento, o acesso, o compartilhamento, a disseminação e o uso dos dados geoespaciais de origem federal, estadual, distrital e municipal.

Ainda, segundo este Decreto, compete a CONCAR a homologação dos padrões para a INDE e as normas para a Cartografia Nacional.

Segundo o Instituto Geográfico Nacional da Espanha (IGN) uma Infra-estrutura de Dados Espaciais (IDE) inclui: (a) os dados e suas descrições (metadados); (b) as tecnologias de busca e acesso; (c) as normas para sua produção, gestão e disseminação e; (d) os acordos entre produtores e entre os usuários (IGN, 2008).

A EDGV é uma norma para produção, gestão e disseminação, de informações geoespaciais, e está em conformidade com os elementos de uma IDE definidos pelo IGN. Portanto, dentre os elementos, a serem disponibilizados por uma IDE, a EDGV é sem dúvida um importante componente. A figura 2 exibe uma arquitetura de uma IDE esboçada pelo IGN.

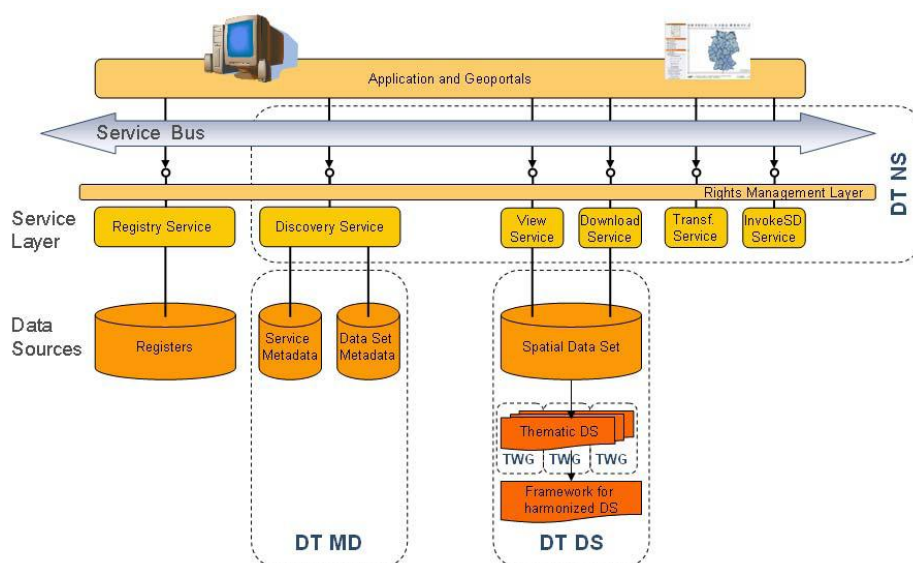


Figura 2 – Arquitetura de uma IDE
Fonte: IGN, 2008

A figura 2 está organizada em camadas. A primeira é propriamente um geportal ou uma aplicação. A segunda camada é a de serviços. Alguns tipos de serviços são: (a) de localização; (b) de visualização; (c) de download; (d) de transformação e; (e) que permitem acesso a serviços de dados espaciais. A terceira camada armazena e disponibiliza os dados de uma maneira geral.

1.2 Estruturação de Dados Geoespaciais Digitais Vetoriais (EDGV)

A EDGV é parte componente da Mapoteca Nacional Digital (MND). A MND é entendida como o conjunto de dados geoespaciais (vetoriais e matriciais) estruturados, seus banco de dados, seus metadados, armazenados em repositórios distribuídos e compartilhados (CONCAR, 2007).

A EDGV foi elaborada para padronizar estruturas de dados que viabilizem o compartilhamento de dados, a interoperabilidade e a racionalização de recursos entre os produtores e consumidores de dados e informação cartográfica (CONCAR, 2007). Um exemplo bem claro da falta de padronização é a forma de trabalho que é empregada na produção das cartas pelos dois grandes centros produtores, o IBGE e a DSG. Enquanto o primeiro utiliza a Mapoteca

Topográfica Digital (MTD), o segundo usa a Tabelas da Base Cartográfica Digital (TBCD), ambos os modelos para representação cartográfica e que não são compatíveis nem interoperáveis (DELOU, 2006).

A EDGV foi modelada em conformidade com o paradigma da orientação a objeto seguindo as especificações da OMT-G e aderente à UML. Uma das especificações é o modelo conceitual, derivado do processo de abstração dos elementos e fenômenos geográficos, que foi realizado com base na análise da descrição da natureza, da terra e dos fenômenos naturais do espaço geográfico brasileiro. Este modelo conceitual foi organizado em pacotes (categorias), conforme apresentado na figura 3. Cada pacote se refere a uma categoria e contém classes que representam os elementos cartográficos, formando a base para a representação cartográfica. Há também pacotes que contêm outros pacotes.

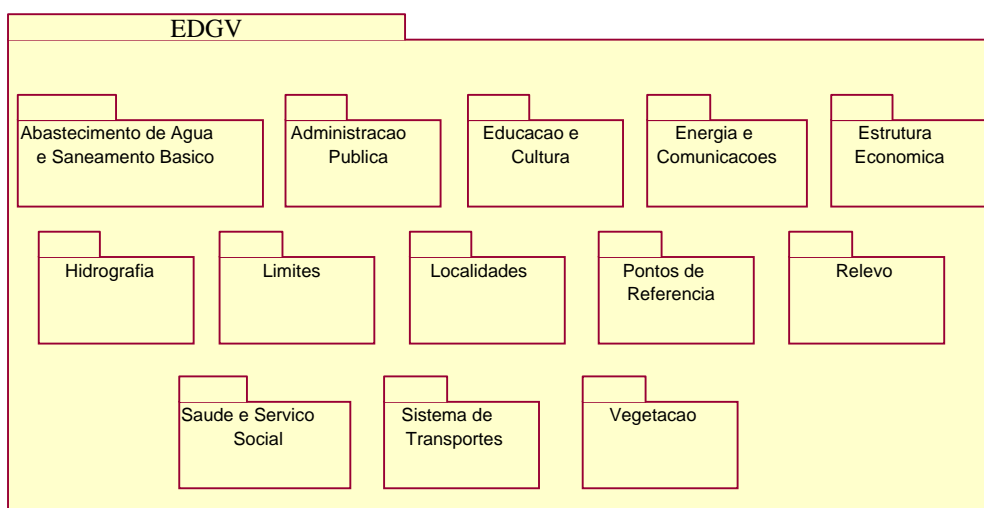


Figura 3 – Pacote com os subpacotes (categorias)

1.3 Modelo Conceitual Orientado a Objeto da EDGV

Fowler (2003) afirma que um modelo conceitual é uma representação dos conceitos relevantes em um domínio de problema. Na EDGV ao invés de representar elementos cartográficos apenas como ponto, linha ou polígono, o modelo conceitual representa classes de feições cartográficas, com atributos alfanuméricos e geométricos. De acordo com Booch, Rumbaugh e Jacobson (2000), no contexto da orientação a objetos, classes são descrições de

conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

A título de exemplo, no modelo da EDGV, feições do tipo Ponte classificam-se em um determinado domínio de problema como objetos da classe Ponte. Para a classe Ponte as características são por exemplo: extensão, nome, carga máxima suportada etc. Mas o que difere a classe Ponte da classe Viaduto? Os objetos da classe Ponte devem necessariamente transpor a objetos da classe Massa D'água. Além disto, devem se relacionar a, pelo menos, dois objetos da classe Trecho Rodoviário. Por outro lado, os objetos da classe Viaduto não possuem o comportamento de transpassar objetos da classe Massa D'água. Em um contexto que se deseje ligar duas instâncias do tipo Trecho Rodoviário atravessando uma instância da classe Massa D'água, não seria possível utilizar objetos da classe Viaduto. Objetos da classe Viaduto, apesar de se relacionarem com objetos da classe Trecho Rodoviário, não são utilizados para a transposição de objetos da classe Massa D'água.

1.3.1 Diagrama de Classes

Segundo Booch, Rumbauch e Jacobson (2000), um Diagrama de Classes mostra um conjunto de classes, interfaces, colaborações e seus relacionamentos. Diagramas de Classe também exibem propriedades e operações de uma classe e suas restrições que, também, se aplicam aos respectivos objetos. É importante relatar que este artefato faz parte da UML.

A especificação, do modelo conceitual da EDGV, além de apontar os relacionamentos mais comuns, faz uma extensão para poder representar também os relacionamentos espaciais. Esta especificação seguiu duas linhas para expressar seu conteúdo e sua semântica: (a) OMT-G e; (b) UML. A primeira está presente na especificação. A segunda em um anexo, no padrão do software Rational Rose, que permite criar os artefatos da UML. Ambos os documentos podem ser obtidos através da CONCAR (<http://www.concar.ibge.gov.br>). Abaixo segue um exemplo de um diagrama de classes da UML, conforme a figura 4.

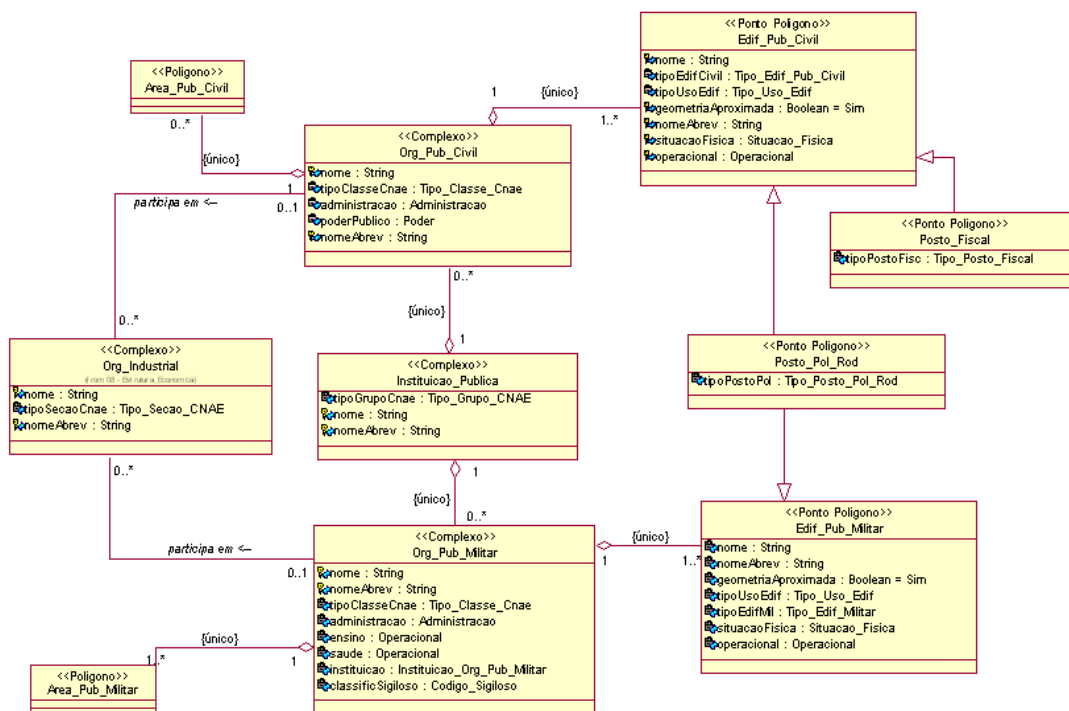


Figura 4 – Diagrama de classes Administração Pública. Versão 2.0 da EDGV
Fonte: CONCAR, 2007

1.4 Modelagem conceitual de dados geográficos (OMT-G)

OMT-G é uma técnica de modelagem baseada na Object Modeling Technique(OMT) e na Unified Modeling Language(UML) para a representação de elementos geográficos. “*Modelos de dados para aplicações geográficas têm necessidades adicionais, tanto com relação à abstração de conceitos e entidades, quanto ao tipo de entidades representáveis e seu inter-relacionamento*” (Casanova et al., 2005, p. 96).

Na especificação técnica da EDGV os diagramas de classes estão representados conforme a OMT-G. Este é o único artefato da OMT-G utilizado.

1.4.1 Diagrama de classes na OMT-G

Segundo Casanova et. al., o diagrama de classes na OMT-G é utilizado para descrever a estrutura e o conteúdo de um banco de dados geográfico. Ele contém elementos específicos da estrutura de um banco de dados, em especial classes de objetos e seus relacionamentos.

As classes no modelo OMT-G representam três grandes grupos de dados: (a) contínuos, (b) discretos e (c) não-espaciais. Esta classificação pode ser encontrada nas aplicações geográficas, proporcionando assim, uma visão integrada do espaço modelado. Classes, no modelo OMT_G, podem ser georreferenciadas ou convencionais (CASANOVA et al., 2005).

1.4.2 Relacionamentos na OMT-G


Este modelo representa três tipos de relacionamentos entre suas classes: associações simples, relacionamentos topológicos em rede e relacionamentos espaciais. As associações simples são indicadas por linhas contínuas, enquanto relacionamentos espaciais são indicados por linhas pontilhadas. Relacionamentos de rede são indicados por duas linhas pontilhadas paralelas (CASANOVA et al., 2005).

1.5 **Relação de Classes e Objetos (RCO) da EDGV**

A especificação da EDGV traz para cada classe seu nome e identificação, sua definição, sua primitiva geométrica, ou seja, se sua representação é um ponto, linha, polígono, combinação destas primitivas geométrica ou ainda um complexo. Cada atributo de uma classe também contém um nome, uma descrição, um tipo, um tamanho e, quando for o caso, a sinalização quanto ao fato de seu preenchimento ser obrigatório ou não. A RCO é parte da especificação da

EDGV que apresenta todas as informações da estrutura de cada classe do modelo conceitual. A título de exemplo a tabela 1 traz a representação da classe Sumidouro_Vertedouro na RCO.

Tabela 1 – Informações da classe Sumidouro_Vertedouro na RCO

Classe	Descrição			Código	Primitiva geométrica		
Sumidouro_Vertedouro	Local de infiltração ou afloramento (ressurgimento) de um curso d'água.			1.11			
Atributo	Tipo	Tamanho	Descrição	Domínio	Descrição	Requisito	Fotografia
nome	Alfanumérico	80	Nome completo da instância.	A ser preenchido	-	NULO	-
geometriaAproximada	Booleano	-	Indica que a geometria adquirida é aproximada em relação à escala prevista para o produto cartográfico.	Sim Não	-	NÃO NULO	-
tipoSumVert	Alfanumérico	12	Tipo da instância, se sumidouro ou vertedouro.	Sumidouro	Local onde um curso d'água passa a fluir de forma subterrânea.	NÃO	1.11_a
				Vertedouro	Local onde um curso d'água volta a fluir novamente sobre a superfície do terreno.	NULO	1.11_b
causa	Alfanumérico	16	Indica causa determinante do surgimento do vertedouro ou sumidouro.	Desconhecida	-	NÃO NULO	-
				Canalização	Canalização, tubulação e/ou conduto forçado de um curso d'água.		1.11_c
				Gruta ou Fenda	-		1.11_d
			Absorção	Absorção, em um terreno poroso (areia), da água.	1.11_e		
nomeAbrev	Alfanumérico	50	Nome ou abreviatura padronizada..	A ser preenchido	-	NULO	-

Fonte: CONCAR, 2007

1.6 Restituição

Para Coelho e Brito (2007), restituição é o processo de confecção de um mapa ou carta, ou parte dele, a partir de fotografias aéreas e informações de controle. A finalidade é a interpretação dos elementos cartográficos localizados no terreno, extraindo-os geograficamente referenciados (coordenadas no espaço-objeto), de forma a compor a base cartográfica de uma região, em determinada escala. A base cartográfica é o conjunto de objetos geograficamente referenciados a um determinado sistema de coordenadas. Estes objetos serão omitidos ou representados de diversas formas, de acordo com a escala empregada. A figura 5 traz a representação da feição açude.



Figura 5 – Representação de um açude

Fonte: T34-700, 1998

Coelho e Brito (2007) afirmam que a restituição digital nada mais é que a restituição executada em ambiente digital. O processo de restituição digital, normalmente, é realizado por software de CAD ou CADD (Computer Aided Design) com suporte a sistemas de coordenadas cartográficas.

O sistema CAD possui algumas limitações. Uma destas limitações é que poucas informações adicionais podem ser utilizadas além das primitivas geométricas como: ponto, linha e polígono. Para exemplificar, um trecho rodoviário e um trecho de drenagem são representados pela primitiva geométrica linha. Neste caso, para diferenciar o trecho rodoviário do trecho de drenagem, faz-se uso de uma simbologia para cada elemento em uma determinada escala. Assim são definidos os símbolos: nível, cor, estilo e peso.

A falta de semântica e conseqüentemente a distinção entre os elementos e seus relacionamentos são os grandes problemas em relação ao CAD, que trata cada elemento cartográfico como um ponto ou uma linha ou um polígono.

Uma outra limitação importante refere-se ao formato do arquivo normalmente utilizado para o armazenamento, que na maioria das vezes é proprietário e fechado, o que dificulta o compartilhamento e a interoperabilidade com outros sistemas.

Segundo Heipke (2001) existe uma tendência a que estações fotogramétricas digitais (EFD) venham a se tornar um “front end” para a aquisição e atualização de dados armazenados num SIG. Contudo, ainda segundo Heipke, para que todo o potencial de integração seja explorado é preciso haver uma maior integração entre o SIG e a EFD.

A EDGV fornece um novo paradigma no processo de restituição, uma vez que seu modelo conceitual, através do mecanismo de classificação dos elementos cartográficos, disponibiliza ao implementador, além dos elementos contemplados na TBCD e MTD, novos

elementos cartográficos. Os elementos da EDGV são na verdade objetos. Com isto se ganha também em semântica. Agora não se fala mais em primitivas geométricas como ponto, linha ou polígono, e sim em objetos, com geometria, que têm característica e comportamento. Neste cenário, executar tarefas de validação e consistência em um processo de restituição fotogramétrica digital é mais simples, pois os próprios objetos com suas semânticas e relacionamentos facilitam o processo de validação que torna os elementos consistentes e topologicamente corretos. Por um outro lado, no processo de restituição será necessário estruturar os elementos cartográficos (geo-objetos) conforme a EDGV.

1.7 Sistema Gerenciador de Banco de Dados Espaciais

De acordo com Tanaka (2007), um SGBD espacial é um sistema gerenciador de banco de dados que armazena e manipula objetos espaciais como quaisquer outros objetos no BD. Objetos espaciais ou geo-objetos são elementos que descrevem localizações ou formas geométricas, como por exemplo, estradas, rios, redes de esgotos, florestas, municípios, lagos etc. Em um computador, esses objetos são representados com pontos, linhas e polígonos. Além do mais, ainda conforme Tanaka (2007), funcionalidades para tratar objetos espaciais devem ser fornecidas. Dentre estas funcionalidades pode-se destacar: (a) suporte indexação espacial; (b) algoritmos eficientes para processamento de operações espaciais e; (c) regras específicas de domínio para otimização de consultas.

2 ARCABOUÇO TECNOLÓGICO

Estações fotogramétricas digitais, em geral, são como caixas pretas, inclusive quanto ao formato de armazenamento. Para romper com esta linha de ação, neste capítulo, serão abordados os elementos tecnológicos, para que se torne possível a realização da restituição fotogramétrica digital de uma forma diferente e aberta. Pretende-se utilizar o E-FOTO integrado a um SGBDE que esteja alinhado com os padrões da OGC para realizar a tarefa de restituição e em conformidade com a EDGV.

2.1 Projeto E-FOTO

De acordo com Coelho e Brito (2007), o projeto E-FOTO tem por objetivos a pesquisa, o desenvolvimento e a implementação de um ambiente para o ensino e auto-aprendizagem de fotogrametria digital, a partir de duas premissas básicas: (1) a liberdade dos programas componentes, em consequência da utilização da licença GNU/GPL e; (2) do livre uso da documentação associada, sob licença GNU/FDL. Atualmente este projeto encontra-se em curso na Faculdade de Engenharia no Programa de pós-graduação em Engenharia de Computação, área de concentração Geomática, da Universidade do Estado do Rio de Janeiro (UERJ).

De acordo com Brito et. al, (2006), o projeto E-FOTO envolve a construção e o gerenciamento de uma EFD livre. Em linhas gerais, EFDs executam diferentes tarefas do fluxo de trabalho fotogramétrico, cada uma realizada por um diferente subsistema da EFD. Este projeto desenvolve e disponibiliza o único ambiente para ensino, pesquisa e auto-aprendizado de Fotogrametria Digital livre e integrado por uma EFD educacional.

Por sua arquitetura aberta, a EFD do projeto E-FOTO, pode se inserir perfeitamente no contexto da INDE, algo desejável às instituições que são produtoras de elementos cartográficos e, conseqüentemente, necessitam de ferramentas adequadas para o desenvolvimento de suas atividades. Ainda no contexto deste projeto, o aperfeiçoamento e desenvolvimento de um dos seus módulos, permitirão realizar as atividades de restituição fotogramétrica digital em

consonância com a EDGV, parte componente da INDE, objetivo principal da presente dissertação. Cabe ainda ressaltar que, com o processo contínuo de desenvolvimento e aperfeiçoamento, o projeto E-FOTO está amadurecendo e isto certamente está incentivando seu uso em ambiente de produção.

Tecnologicamente, os módulos do E-FOTO foram desenvolvidos com a linguagem de programação C++, utilizando o paradigma da orientação a objeto. O framework utilizado para o seu desenvolvimento foi o QT, em virtude das suas bibliotecas e pelo fato de ser multiplataforma. Também é importante afirmar que entre as bibliotecas do QT existem *drivers* para a comunicação com os principais bancos de dados, incluindo aqueles que foram estendidos para suportar a componente geométrica.

2.2 Padrões da Open Geospatial Consortium (OGC)

2.2.1 OGC

No ano de 1994 foi fundado o consórcio Open Geospatial Consortium(OGC), que é uma organização voluntária internacional de padrões de consenso, com a finalidade de promover a interoperabilidade de informações espaciais. O consórcio internacional OGC conta com a participação de mais de 335 companhias, agências governamentais e universidades que através de um processo consensual se propõem a desenvolver padrões para interfaces de especificações públicas que permitam a interoperabilidade entre as diferentes geotecnologias (OGC, 2009).

As especificações da OGC seguem um modelo padrão, a saber: (i) especificações abstratas para apresentação de conceitos e servir de base para outras especificações; (ii) especificações de implementação; (iii) modelo de referência para apontamento dos trabalhos em andamento; (iv) documento de melhores práticas, que endereça boas práticas para implementação das especificações; (v) artigos afins para discussão, que são usados pelos grupos de trabalho, apesar de não representarem qualquer posição oficial. Dentre algumas especificações, podem ser

citadas: Web Map Service (WMS), Web Feature Service (WFS), Geography Markup Language (GML), Simple Feature Access (SFA) etc.

No contexto deste trabalho a especificação SFA será utilizada para permitir o armazenamento das feições restituídas pelo E-FOTO, conforme modelo conceitual da EDGV.

2.2.2 Simple Feature Access

Um dos padrões de interface de especificação, definido pela OGC, é o Simple Feature Access (SFA). Ele fornece um padrão para armazenamento e acesso a dados espaciais em banco de dados relacionais e objeto-relacionais. Por parte das aplicações, os dados podem ser utilizados por outras aplicações através de um modelo comum de armazenamento de dados e de uma interface de acesso para informação. O SFA está focado para o uso de elementos de dados vetoriais tais como: pontos, linhas e polígonos (OGC, 2009).

A norma SFA está dividida em duas partes: (1) arquitetura comum, que define o modelo geométrico e as operações espaciais e (2) implementação, que fornece uma interface de acesso a dados, através de uma determinada tecnologia de implementação. As opções de implementação disponíveis são SQL, CORBA e COM.

É oportuno relatar que em relação à parte 2, este trabalho endereça somente a opção, SFA para SQL, já que um dos objetivos desta dissertação, é o de prover uma solução para mapear a EDGV para bancos de dados relacionais geográficos no contexto do projeto E-FOTO.

2.2.2.1 Parte 1 - Arquitetura Comum

Esta especificação estabelece uma arquitetura comum e define termos de uso na arquitetura. Esta parte trata, sem detalhar, aspectos de implementação dos seguintes assuntos: (a) modelo geométrico; (b) a interface geometry; (c) operadores topológicos (d) texto de anotação (Annotation Text) e; (e) formatos textos e binários. (OGC, 2006)

2.2.2.1.1 Modelo Geométrico

Este modelo exhibe os tipos geométricos suportados, suas relações e o conjunto de operadores sobre os tipos geométricos. A figura 6 apresenta o modelo geométrico do SFA.

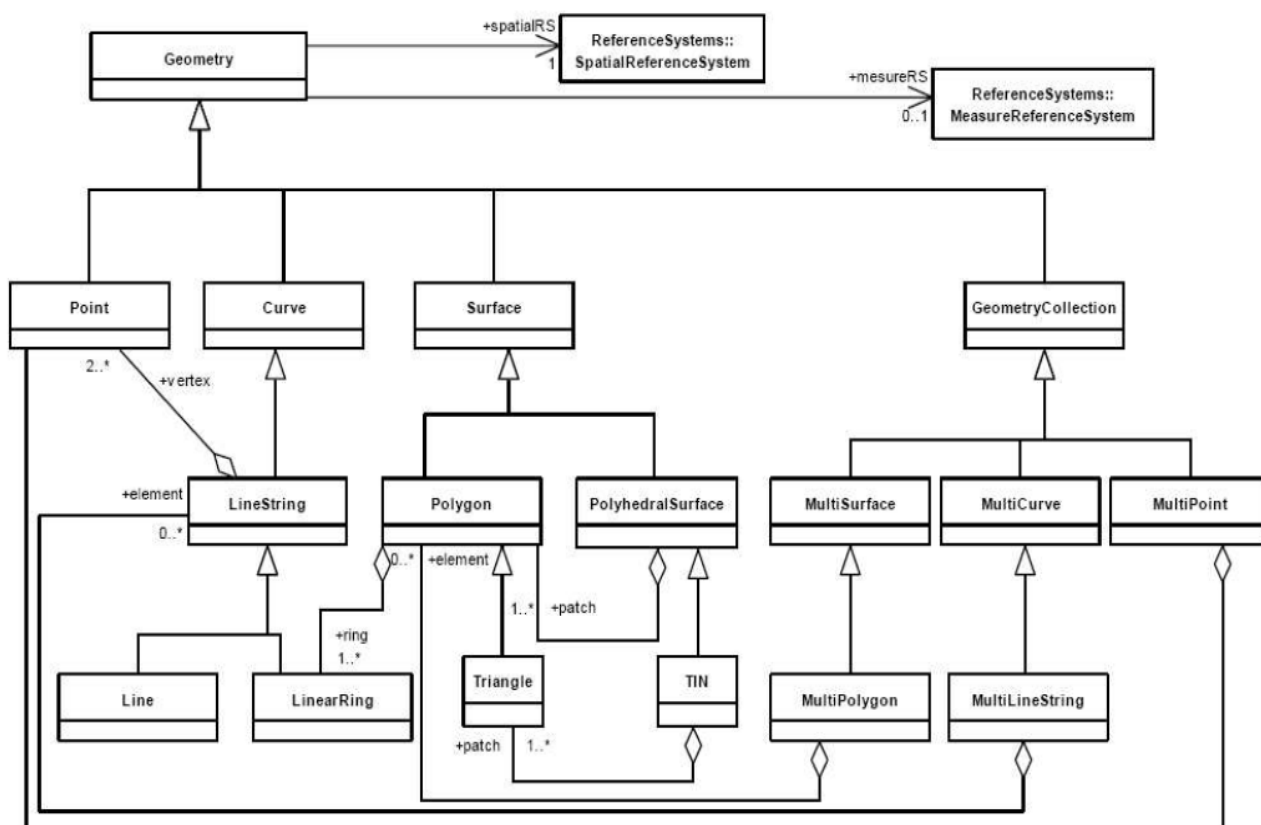


Figura 6 – Modelo geométrico do SFA

Fonte: OGC, 2006

Analisando a figura 6, é possível identificar a classe base abstrata **Geometry**, que possui as subclasses **Point**, **Curve**, **Surface** e **GeometryCollection**. Estas subclasses, por sua vez, também são especializadas em outras classes. Cada objeto geométrico é associado a um Sistema de Referência Espacial, que descreve as coordenadas do espaço geométrico em que o objeto é definido. Esta hierarquia de classes e relacionamentos possibilita a representação vetorial de qualquer elemento cartográfico.

Ainda sobre o modelo geométrico, em relação à dimensionalidade de uma figura geométrica, pode-se classificá-la como: (i) dimensionalidade 0 – ponto; (ii) dimensionalidade 1 – linha; (iii) dimensionalidade 2 – polígono. Em relação à dimensionalidade do espaço onde está inserida a figura geométrica, pode-se relacioná-la ao seguinte espaço objeto: (a) plano cartesiano 2D, (b) espaço tridimensional 3D, (d) plano cartesiano 2D com uma medida associada e (e) espaço tridimensional 3D com uma medida associada. A título de exemplo, um ponto com as coordenadas x e y encaixa-se no plano cartesiano 2D; um ponto com as coordenadas x , y , z e m encaixa-se no espaço tridimensional 3D com uma medida associada. Cabe ainda elucidar que a interpretação das coordenadas está sujeita ao sistema de referência associado ao ponto.

2.2.2.1.2 Interface da classe Geometry

De acordo com o diagrama de classes na figura 6, a classe `Geometry` ancestral de todas as outras, é abstrata oferecendo uma interface comum às operações espaciais. A figura 7 exibe a interface da classe `Geometry`.

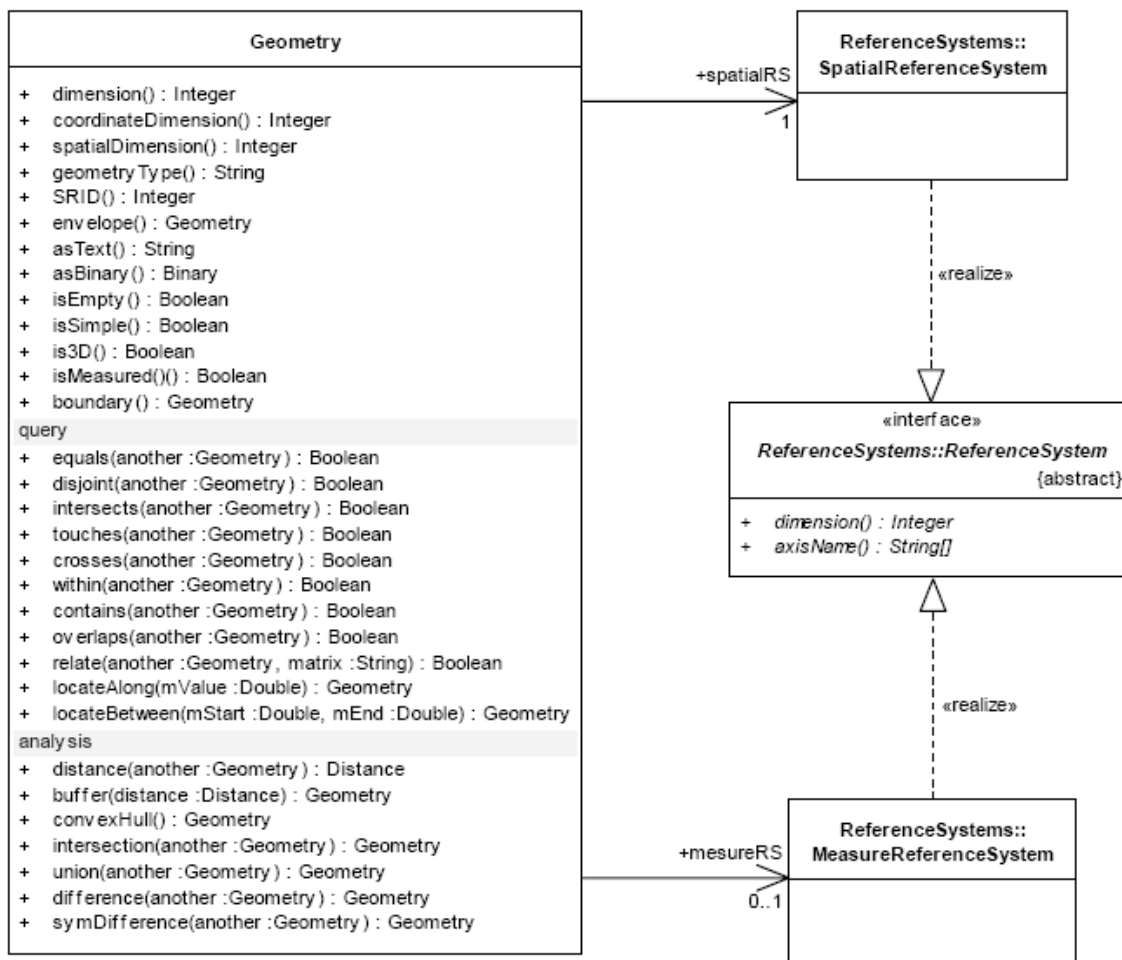


Figura 7 – Interface da classe Geometry

Fonte: OGC, 2006

Observando a figura 7, nota-se que as operações podem ser agrupadas da seguinte forma: (i) operações básicas, que informam sobre o tipo do elemento geométrico, sistema de coordenadas, se é uma geometria simples, seu perímetro, entre outras; (ii) operações topológicas para testar os relacionamentos espaciais entre objetos geométricos tais como: se são iguais, se não se tocam, se há interseção, entre outras; (iii) operações que dão suporte à análise tais como: a menor distância entre dois pontos, o elemento geométrico resultante de uma interseção, entre outras.

2.2.2.1.3 Operadores Topológicos

São baseados no conceito de Interior (I), Limite (L) e Exterior (E) de um elemento geométrico. Para exemplificar seja um objeto geométrico n com as seguintes definições: (1) $I(n)$ – representação do interior de n ; (2) $L(n)$ – representação do limite de n ; (3) $E(n)$ – representação do exterior de n . A interseção entre as representações, interior, limite e exterior de dois objetos geométricos pode resultar em um conjunto de elementos geométricos. Também pode responder consultas do tipo: a toca b ; a é igual a b ; a intercepta b ; a contém b ; entre outras. (OGC, 2006)

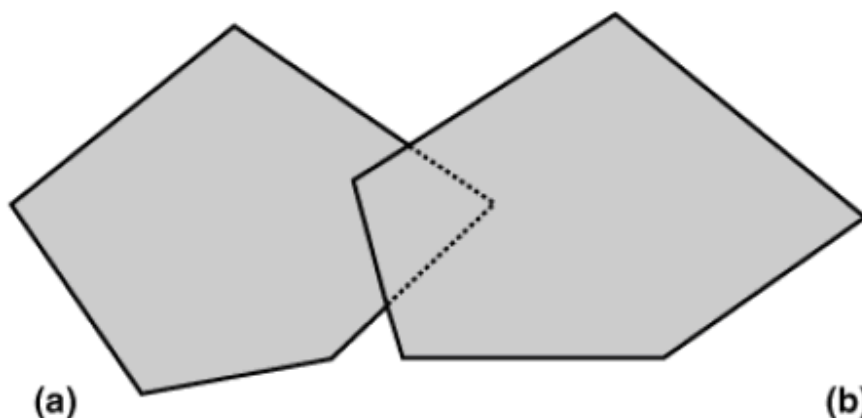
Os relacionamentos topológicos podem ser definidos com base em um modelo, chamado Matriz Modelo de Nove Interseções de Dimensionalidade Estendida (DE-9IM), conforme a tabela 2 ilustra as relações possíveis entre dois elementos geométricos.

Tabela 2 – Matriz de Nove Interseções de Dimensionalidade Estendida (DE-9IM)

	Interior	Limite	Exterior
Interior	$\dim(I(a) \cap I(b))$	$\dim(I(a) \cap L(b))$	$\dim(I(a) \cap E(b))$
Limite	$\dim(L(a) \cap I(b))$	$\dim(L(a) \cap L(b))$	$\dim(L(a) \cap E(b))$
Exterior	$\dim(E(a) \cap I(b))$	$\dim(E(a) \cap L(b))$	$\dim(E(a) \cap E(b))$

Fonte: Adaptado de (OGC, 2006)

Para exemplificar seja $\dim(n)$ o valor de retorno máximo de dimensão. Este valor de retorno corresponde a um dos seguintes valores: (-1, 0, 1, 2). No caso de valor de retorno -1, equivale a $\dim(n) = \text{vazio}$ ou $\dim(\emptyset)$. O valor 0 indica dimensionalidade de um ponto, o valor 1 indica dimensionalidade de uma linha e o valor 2 indica dimensionalidade de um polígono. A figura 8 exhibe o relacionamento entre dois polígonos.



	Interior	Limite	Exterior
Interior	2	1	2
Limite	1	0	1
Exterior	2	1	2

Figura 8 – Relacionamento entre dois polígonos
 Fonte: Adaptado de (OGC, 2006)

Observando a figura 8, a interseção entre o interior do polígono (a) com o interior do polígono (b) gera um outro polígono. A interseção entre o limite do polígono (a) com o interior do polígono (b) gera uma linha. O mesmo raciocínio é empregado para as outras relações.

2.2.2.1.4 Formatos Texto e Binário

Há duas formas de representar objetos geométricos. A primeira é no formato texto, que é referida como texto conhecido (Well-known Text-WKT). A segunda no formato binário, adotada como binário conhecido (Well-known Binary-WKB).

A representação WKT define como representar textualmente todas as diferentes instâncias de geometrias. A tabela 3 ilustra a representação textual dos elementos geométricos.

Tabela 3 – Representação textual dos elementos geométricos

Tipo de Geometria	Representação Literal	Comentário
Point	Point (10 10)	Um ponto
LineString	LineString (10 10, 20 20, 30 40)	Uma LineString com 3 pontos
Polygon	Polygon ((10 10, 10 20, 20 20, 20 15, 10 10))	Um polígono
Multipoint	MultiPoint ((10 10), (20 20))	Um MultiPoint com 2 pontos
MultiLineString	MultiLineString ((10 10, 20 20), (15 15, 30 15))	Um MultiLineString com duas linhas
MultiPolygon	MultiPolygon (((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)))	Um MultiPolygon com dois polígonos
GeomCollection	GeometryCollection (POINT (10 10), POINT (30 30), LINESTRING (15 15, 20 20))	Um GeometryCollection consistindo de dois Pontos e uma linha
PolyhedralSurface	PolyhedralSurface Z (((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0)), ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)), ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0)), ((1 1 0, 1 1 1, 1 0 1, 1 0 0, 1 1 0)), ((0 1 0, 0 1 1, 1 1 1, 1 1 0, 0 1 0)), ((0 0 1, 1 0 1, 1 1 1, 0 1 1, 0 0 1)))	Um cubo poliédrico
Tin	Tin Z (((0 0 0, 0 0 1, 0 1 0, 0 0 0)), ((0 0 0, 0 1 0, 1 0 0, 0 0 0)), ((0 0 0, 1 0 0, 0 0 1, 0 0 0)), ((1 0 0, 0 1 0, 0 0 1, 1 0 0)),)	Um tetraedro de 4 faces triangular

Tipo de Geometria	Representação Literal	Comentário
Point	Point Z (10 10 5)	Um ponto 3D
Point	Point ZM (10 10 5 40)	O mesmo ponto 3D com um valor de medida 40
Point	Point M (10 10 40)	Um ponto 2D como um valor de medida 40

Fonte: Adaptado de (OGC, 2006)

Nota-se na tabela 3 uma gramática que define como representar textualmente todas as diferentes instâncias de geometrias. Com esta representação, é possível inclusive realizar consultas e atualizações em banco de dados. Um exemplo de inserção em um SGBDE é apresentado a seguir:

```
insert into edgv.elemento_fisiografico ( id_elemento_fisiografico,geom, classe_do_objeto )
VALUES( 2, pointfromtext( 'POINT (56.17 16.12 )', 4291 ), 'Elemento_Fisiografico_Natural' ).
```

Observe que o trecho POINT(56.17 16.12) está na representação WKT.

A Representação WKB proporciona uma representação binária portátil de um objeto geométrico como um fluxo de bytes contínuo. Permite que aplicações clientes e o banco de dados troquem objetos geométricos no formato binário de maneira mais eficiente, principalmente quando comparado, com a representação WKT. No contexto desta dissertação somente a representação WKT será utilizada.

2.2.2.2 Parte 2 – Interface de acesso – implementação SQL

Este padrão especifica um esquema SQL que permite armazenamento, consulta e atualização de um elemento geométrico via uma interface de chamada da SQL (SQL Call Level Interface - SQL/CLI) (OGC, 2006). Define o conceito de tabelas de feição, ou seja, elementos

que contêm atributos alfanuméricos e geométricos. Define também tabelas de metadados para armazenar quais são as tabelas de feição e quais colunas contêm informação da componente espacial e, por último, define o formato das tabelas de feição.

Sua implementação é dependente da primeira parte, chamada de arquitetura comum. Basicamente existem as seguintes estratégias de implementação: (1) Tabelas de feições baseadas em tipos pré-definidos da SQL, usando tipos numéricos da SQL para armazenamento das geometrias e acesso via SQL; (2) Tabelas de feições baseadas em tipos pré-definidos da SQL, usando tipos binários da SQL para armazenar geometrias e acesso via SQL/CLI; (3) Tabelas de feições com tipos geométricos suportando acesso textual e binário as geometrias através da SQL. Para os dois primeiros casos a figura 9 exhibe a arquitetura/esquema.

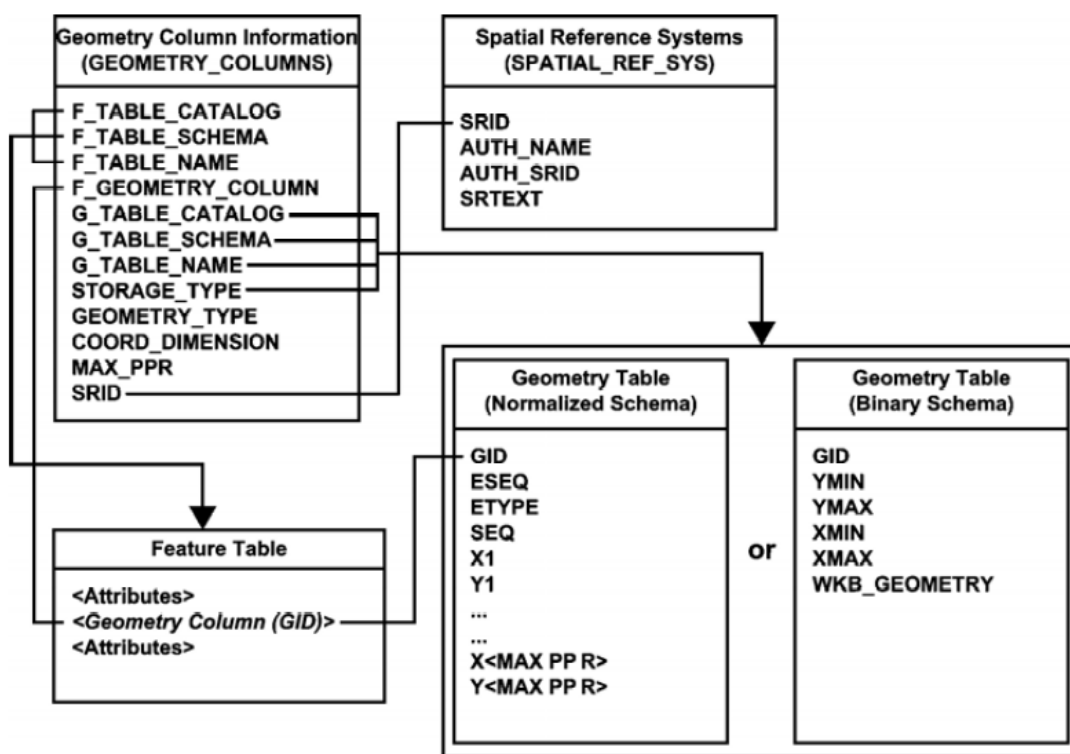


Figura 9 – Arquitetura para tipos pré-definidos da SQL
Fonte: OGC, 2006

2.2.2.2.1 Tabelas de feições baseadas em tipos pré-definidos da SQL

O esquema, apresentado na figura 9 dá suporte para a arquitetura de tipos pré-definidos da SQL. Cada tabela exerce um papel importante a saber: (a) a tabela **GEOMETRY_COLUMNS**

descreve as tabelas de feições que foram criadas no banco de dados com suas respectivas colunas geométricas; (b) a tabela `SPATIAL_REF_SYS` descreve o sistema de coordenadas e transformações para a geometria. (c) A tabela `FEATURE` é um modelo que representa um conjunto de tabelas que mapeiam entidades do mundo real que contenham uma geometria associada, ou seja, são as tabelas que foram criadas contendo atributos espaciais e não-espaciais. Na tabela `Feature` a coluna `Geometry (GID)` é uma chave estrangeira para a tabela `GEOMETRY`; (d) a tabela `GEOMETRY` armazena objetos geométricos e pode ser implementada usando o padrão de tipo numérico ou de tipo binário da SQL.

Dependendo do tipo de armazenamento especificado pela tabela `GEOMETRY_COLUMNS`, um objeto geométrico é armazenado ou como um conjunto de valores de coordenadas ou como um valor binário único. Conforme citado anteriormente a tabela `GEOMETRY` pode ser implementada de duas formas: (a) esquema de geometria normalizada e (b) esquema de geometria binária. No primeiro caso as coordenadas dos objetos geométricos são armazenadas como tipos de dados numéricos pré-definidos. No segundo esquema com o formato binário. A figura 10 traz um exemplo de duas feições com suas respectivas coordenadas.

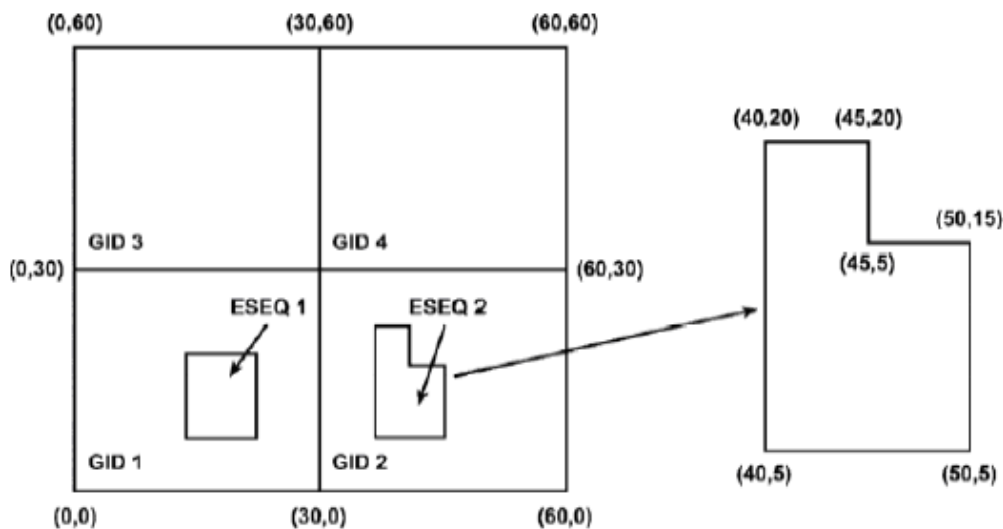


Figura 10 – Feições com coordenadas
Fonte: Adaptado de (OGC, 2006)

A tabela 4 mostra o esquema de geometria normalizada. A tabela 5 ilustra o esquema de geometria binária. Ambos guardam o retângulo envolvente da feição, conforme mostrado na figura 10.

Tabela 4 – Esquema de geometria normalizada

GID 1	ESEQ	ETYPE	SEQ	X0	Y0	X1	Y1	X2	Y2	X3	Y3	X4	Y4
1	1	3	1	0	0	0	30	30	30	30	0	0	0
1	2	3	1	10	10	10	20	20	20	20	10	10	10
2	1	3	1	30	0	30	30	60	30	60	0	30	0
2	2	3	1	40	5	40	20	45	20	45	15	50	15
2	2	3	1	50	15	50	5	40	5	NII	NII	NII	NII
3	1	3	1	0	30	0	60	30	60	30	30	0	30
4	1	3	1	30	30	30	60	60	60	60	30	30	30

Fonte: OGC, 2006

No esquema normalizado, tabela 4, cada *objeto geométrico* é identificado por uma chave (GID) e consiste de um ou mais elementos primitivos ordenados por um elemento seqüencial (ESEQ). Cada elemento primitivo no *objeto geométrico* é distribuído através de uma ou mais linhas na tabela Geometry, identificado por um tipo primitivo (ETYPE) e ordenado por um número seqüencial (SEQ). Nota-se também no esquema normalizado, que os tipos numéricos predefinidos da SQL são utilizados para as coordenadas e estes valores numéricos são obtidos a partir da tabela GEOMETRY até que *objeto geométrico* seja totalmente reconstruído.

Tabela 5 – Esquema geometria binária

GID	XMIN	YMIN	XMAX	YMAX	Geometry
1	0	0	30	30	< WKBGeometry >
2	30	0	60	30	< WKBGeometry >
3	0	30	30	60	< WKBGeometry >
4	30	30	60	60	< WKBGeometry >

Fonte: OGC, 2006

No esquema de geometria binária, tabela 5, a coluna GID é uma chave primária; a coluna Geometry armazena a informação espacial no formato binário, mais especificamente no padrão WKB.

2.2.2.2.2 Tabelas de feições com tipos geométricos

Este esquema leva em consideração o tipo geométrico da SQL, de maneira que é possível definir uma coluna com tipos geométricos como: Point, LineString, Polygon etc. Este modelo trabalha com as tabelas FEATURE, GEOMETRY_COLUMNS e SPATIAL_REF_SYS. No caso das tabelas FEATURE, que representam tabelas de feições, a coluna Geometry mapeia direto para um tipo geométrico. O SGBD que implementa este esquema disponibiliza para o usuário mais facilidade e flexibilidade de uso, já que os tipos geométricos são suportados diretamente. A figura 11 exibe o esquema.

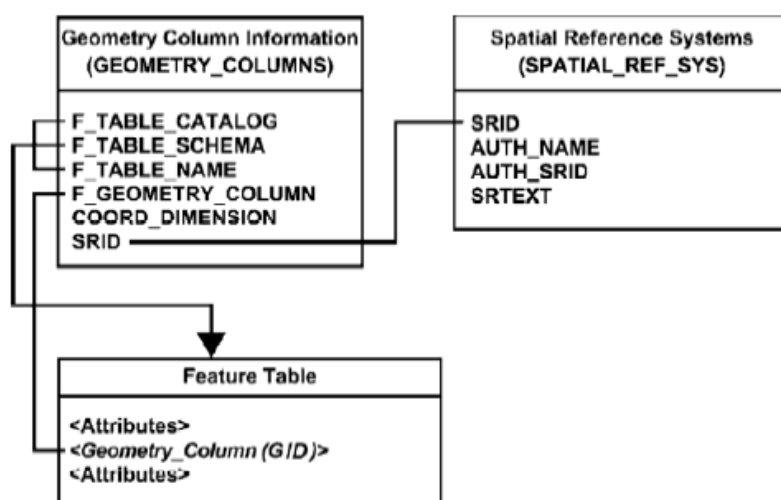


Figura 11 – Feições com tipos geométricos

Fonte: OGC, 2006

No esquema apresentado na figura 11 qualquer tabela de feição possui uma coluna geométrica que é de algum tipo conforme exibido na figura 6.

2.3 SGBD Postgre SQL

O PostgreSQL é um SGBD objeto-relacional de código aberto, maduro e ativo com mais de 15 anos de desenvolvimento. É robusto e confiável, além de ser flexível e rico em recursos. Ele é considerado objeto-relacional por implementar, além das características de um SGBD

relacional, possui também algumas características de orientação a objetos, tais como herança e tipos personalizados.

O PostgreSQL funciona nos principais sistemas operacionais, incluindo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e Windows. Ele inclui a maioria dos tipos de dados SQL92 e SQL99, incluindo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, e TIMESTAMP. Também suporta armazenamento de grandes objetos binários, incluindo imagens, sons ou vídeo. (PostgreSQL, 2007).

Alguns recursos do PostgreSQL são: (1) interface com linguagens procedurais (PL/pgSQL, PL/Tcl, PL/Perl, PL/Python, Ruby etc); (2) tipos de dados extensíveis no contexto da SQL (CREATE TYPE); (3) suporte a objetos complexos como arrays; (4) suporte ao princípio da herança da orientação a objeto, no contexto da SQL (INHERITS); (5) subsistema de regras de produção (TRIGGER e FUNCTION); (6) vários recursos de extensão, dentre os quais se encontra o cartucho espacial PostGIS, a ser descrito no item 2.4.

No contexto do projeto E-FOTO o PostgreSQL será o SGBD responsável pelo armazenamento das feições restituídas através do módulo de restituição fotogramétrica digital. Cabe aqui, uma observação que o critério para a seleção do SGBD está descrito no capítulo 3.

2.4 Cartucho Espacial PostGIS

O cartucho PostGIS é desenvolvido pela Refrations Research Inc, como um projeto de pesquisa tecnológica de dados espaciais. Refrations é uma empresa de consultoria em SIG e banco de dados em Victoria, British Columbia, Canadá, especializada na integração de dados e software customizados (POSTGIS, 2008). O PostGIS é disponibilizado sob licença General Public License (GNU).

Este cartucho adiciona suporte para objetos geográficos para o banco de dados objeto relacional PostgreSQL. Com efeito, a partir da instalação do PostGIS, o SGBD PostgreSQL pode trabalhar como um “backend” de dados espaciais para Sistemas de Informação Geográfica. Dentre suas características destacam-se as seguintes:

- (a) O armazenamento de objetos geométricos do tipo point, linestring, polygon, multipoint, multiline, multipolygon, e coleções geométricas.
- (b) Conformidade com a especificação SFA da OGC.
- (c) Uso das formas de representação WKB e WKT. Tanto WKT e WKB podem ser usadas e em conformidade com a especificação SFA.
- (d) Centenas de funções espaciais divididas da seguinte maneira: (i) gerenciamento; (ii) relacionamento topológicos; (iii) processamento; (iv) acesso; (v) criação.
- (e) Suporte para índices espaciais.

2.4.1 Esquema Tabela de Feições com Geometria

O consórcio OpenGIS define o padrão SFA, para tipos de objetos GIS, as funções necessárias para manipulá-los e um conjunto de tabelas de metadados. A fim de garantir a coerência dos metadados, operações como a criação e a remoção de colunas espaciais são realizadas através de procedimentos especiais definidos por esta entidade. No PostGIS há duas tabelas de metadados: `SPATIAL_REF_SYS` e `GEOMETRY_COLUMNS`. A tabela `SPATIAL_REF_SYS` detém os IDs numéricos e descrições textuais do sistemas de coordenadas espaciais utilizados na base de dados (POSTGIS, 2008). A tabela `GEOMETRY_COLUMNS` descreve as tabelas de feições disponíveis e suas colunas geométricas.

A figura 12 exibe duas tabelas criadas no Postgresql/PostGIS e o esquema definido na especificação SFA.

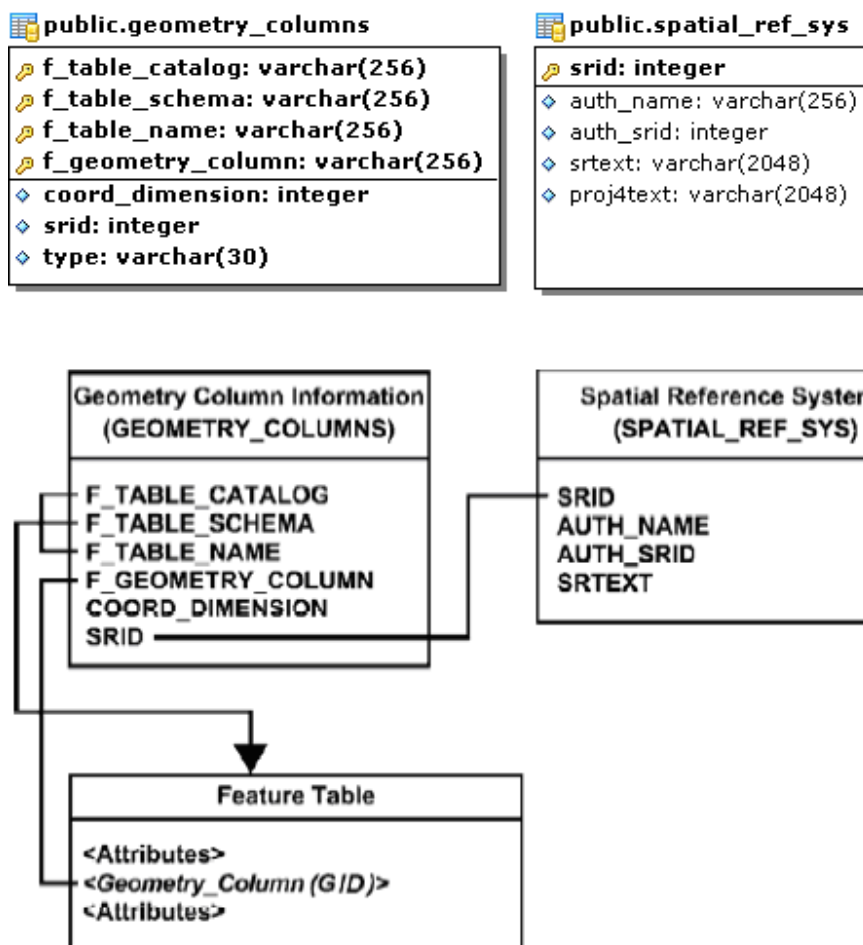


Figura 12 – Tabelas no PostgreSQL/PostGIS e o esquema da SFA

É interessante observar que as tabelas no PostgreSQL/PostGIS têm todos os campos que estão definidos no esquema. No caso da tabela `GEOMETRY_COLUMNS` existe uma coluna a mais, de nome `type`, para descrever o tipo da geometria. Na tabela `SPATIAL_REF_SYS` há uma coluna chamada `proj4text`, que é para prover transformações de coordenadas; esta coluna não está presente no esquema da SFA. Ainda é importante relatar que a tabela `FEATURE` do esquema SFA equivale no PostgreSQL/PostGIS a qualquer tabela que tenha uma coluna geométrica.

3 METODOLOGIA PARA REPRESENTAÇÃO DA EDGV EM BANCOS DE DADOS GEOGRÁFICOS

O modelo de classes da EDGV possui mais de trezentas classes. Estas classes são persistentes, necessitando de um SGBD para armazená-las. Este capítulo define uma tecnologia de banco de dados e um método para mapear as classes da EDGV para esta tecnologia.

3.1 Seleção do Sistema Gerenciador de Banco de Dados

De acordo Silberschartz, Henry e Sudarshan (2001), um Sistema Gerenciador de Banco de Dados (SGBD) é uma coleção de dados inter-relacionados e um conjunto de programas para acessar estes dados. Silberschartz, Henry e Sudarshan (2001) ainda afirmam que o principal objetivo de um SGBD é prover uma forma para armazenar e recuperar informações na base de dados de maneira conveniente e eficiente.

Sistemas Gerenciadores de Banco de Dados Relacionais (SGBDR) e Sistemas de Banco de Dados Orientados a Objeto (SGBDOO) possuem características distintas, mas basicamente endereçam ao mesmo propósito: armazenar dados necessários para a sustentação do negócio para o qual são aplicados, possibilitando a recuperação, atualização, comparação e tratamento desses elementos, a fim de produzir resultados palpáveis.

Segundo Silberschartz, Henry e Sudarshan (2001), um SGBDR é uma coleção de tabelas que compõe a base de dados, podendo estas tabelas estar inter-relacionadas. As tabelas podem ser acessadas seguindo critérios de segurança. Conceitos como integridade referencial de dados, que garantem que um dado referenciado em uma tabela esteja presente na tabela onde está sendo referenciada e chaves primárias estão presentes e garantem que um conjunto de informações possa ser representado de maneira consistente e independente da forma de acesso. Além disso, bancos de dados relacionais são baseados na teoria relacional.

De acordo com a ODBMS (2009), um SGBDOO é um sistema de gerenciamento de banco de dados que se apóia na criação e modelagem de dados como objetos. Um SGBDOO está fundamentado basicamente em três princípios: herança, polimorfismo e encapsulamento. Este modelo apresenta maior flexibilidade na manipulação de seu conteúdo. Cada objeto possui um identificador implícito único que é independente do valor desse objeto e de seu local físico de armazenamento. Os maiores problemas com esta tecnologia ainda são a falta da utilização do padrão definido e a fraca disseminação, ficando restrito a pequenos nichos.

Conforme apresentado anteriormente, a EDGV foi modelada segundo o paradigma da orientação a objetos. A princípio, tanto as aplicações quanto as bases de dados deveriam seguir o paradigma da Orientação a Objetos (OO). Contudo, no que se refere ao armazenamento dos dados, o modelo relacional é adotado na grande maioria dos sistemas, devido à sua maturidade, robustez, segurança e, sobretudo, seu formalismo matemático. Cabe ainda ressaltar que muitos dos SGBDR foram estendidos para suportar algumas características da orientação a objeto, são os chamados “Sistemas Gerenciadores de Banco de Dados Objeto Relacional (SGBDOR)”. É o caso do Oracle, DB2, PostgreSQL, entre outros, que normalmente implementam alguns conceitos do paradigma da orientação a objeto. Porém, mesmo com a padronização do SQL 1999, que introduz conceitos de orientação a objeto e novos tipos de dados, algumas destas tecnologias ainda não estão padronizadas ou não implementam totalmente as novas características definidas no SQL 1999, fato que pode ser constatado no PostgreSQL (POSTGRESQL, 2008).

Conforme mencionado anteriormente, o modelo relacional é maduro, seguro, robusto e fortemente disseminado. Optar por um SGBDOO seria um contra-senso, considerando os investimentos já realizados para a aquisição de sistemas gerenciadores de bancos de dados relacionais (SGBDR) e treinamento de pessoal, que teriam que ser refeitos para esta tecnologia. Há que se mencionar também o risco inerente à mudança de tecnologia e da necessidade de migração de base de dados. Ainda é importante destacar que grandes instituições públicas, produtoras de dados geoespaciais, como é o caso do IBGE, utilizam em suas linhas de produção, SGBDR como software corporativo. Por estas razões, este capítulo se dedica a apresentar a proposta de representação da EDGV numa estrutura de armazenamento relacional.

É preciso ainda destacar dois requisitos adicionais desta demanda. Primeiro, o fato de esta estratégia ter que lidar com bases de dados geoespaciais, portanto, considerando as componentes

geométricas. Segundo, a estratégia deve ser genérica de forma a permitir sua implementação em qualquer SGBDR.

Por último, é oportuno destacar que o protótipo do E-FOTO, que permitirá a execução da restituição fotogramétrica 3D na EDGV, terá como primeira plataforma tecnológica de banco de dados o SGBDOR PostgreSQL, em função deste possuir um cartucho espacial aderente aos padrões da OGC e ser um Software Livre.

3.2 Mapeando do Modelo Orientado a Objeto para o Modelo Relacional para EDGV

Eleito o modelo relacional como o paradigma de armazenamento, faz-se necessária a definição de uma estratégia para a implementação da EDGV para esse modelo.

O modelo conceitual da EDGV é organizado em treze pacotes: (1) Abastecimento de Água e Saneamento Básico; (2) Administração Pública; (3) Educação e Cultura; (4) Energia e Comunicações; (5) Estrutura Econômica; (6) Hidrografia; (7) Limites; (8) Localidades; (9) Pontos de Referência; (10) Relevo; (11) Saúde e Serviço Social; (12) Sistema de Transportes; (13) Vegetação. Cada pacote na EDGV contempla um diagrama de classes. Este modelo, depois de traduzido para o esquema relacional, poderá ser utilizado, por exemplo, no armazenamento de dados geográficos 3D adquiridos através do processo de restituição fotogramétrica digital. Portanto, é necessário que o armazenamento de informações alfanuméricas e espaciais seja uniforme. As seções a seguir tratam deste suporte por parte dos SGBDR.

3.2.1 Tipos de classe da EDGV

Existem basicamente três tipos de classes:

- Classes de Feição possuem geometria associada, que pode ser do tipo ponto, linha ou polígono.

- Classes Agregadoras não possuem geometria, somente agregam classes que possuem geometria e podem referenciar classes de domínio;
- Classes de Domínio são equivalentes aos *Code List* definidos na UML, não possuem geometria e nem agregam outras classes; simplesmente descrevem conjuntos de valores possíveis e podem ser referenciadas por classes do tipo Feição e Agregadora. A figura 13 ilustra um tipo de cada classe.

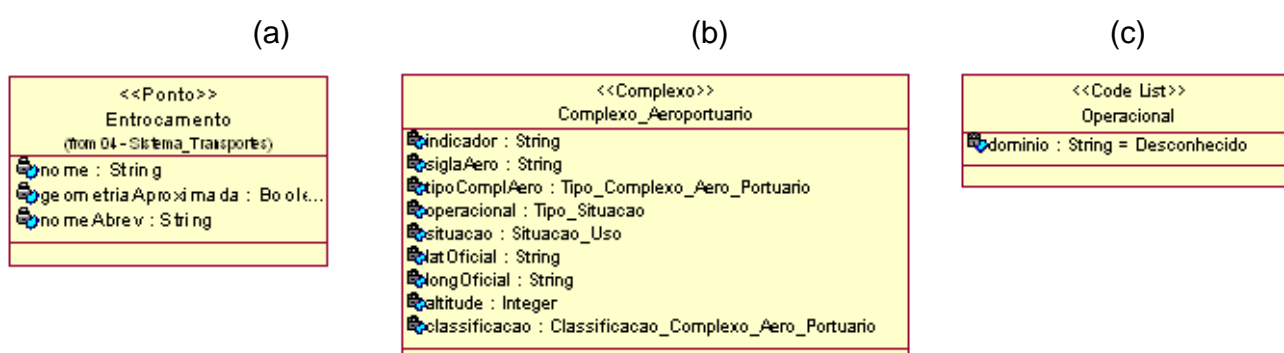


Figura 13 – Classe de: (a) Feição, (b) Agregadora e (c) Domínio

3.2.2 Mapeamento de Classe para a Tabela

Toda classe tem ao menos uma tabela correspondente. Para nomeá-la usa-se um padrão. Este padrão atribui à tabela o mesmo nome da classe desde que tenha no máximo 27 caracteres. Caso ultrapasse este limite, o nome será abreviado. Toda tabela terá uma chave primária artificial e seqüencial, do tipo “integer”, para corresponder ao identificador único intrínseco de cada objeto. A figura 14 exemplifica este mapeamento. O padrão para nomear a chave primária da tabela será: `id_nome_da_tabela`.

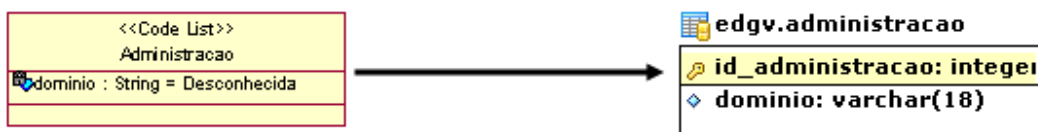


Figura 14 – Mapeamento de uma classe para tabela

3.2.3 Mapeamento dos Tipos Primitivos de Atributos para Tipos da SQL

Os atributos primitivos das classes da EDGV são os atributos primitivos que constam na maioria das linguagens de programação. Para mapeá-los para os tipos da SQL é necessário criar uma relação. A tabela 6 realiza este mapeamento. O nome da coluna no banco de dados terá o mesmo nome do atributo da classe, respeitando o limite máximo de 30 caracteres. Caso o limite de 30 caracteres seja atingido o nome da coluna será abreviado.

Tabela 6 – Mapeamento tipo do atributo para tipo SQL

Tipo do Atributo	Tipo da SQL
Alfanumérico	Char, Varchar
Booleano	Boolean, [char(1) – {S,N}]
Data	Date, Time, Timestamp
Decimal	Decimal
Inteiro	Integer
Flutuante	Float
Real	Real

3.2.4 Mapeamento dos Atributos para Colunas

Um atributo de uma classe pode ser mapeado para zero ou mais colunas em um banco de dados. Isto porque há casos no qual um atributo é derivado de um outro atributo. Há também a possibilidade de um atributo ser mapeado para mais de uma coluna. A figura 15 ilustra um atributo derivado e um composto.

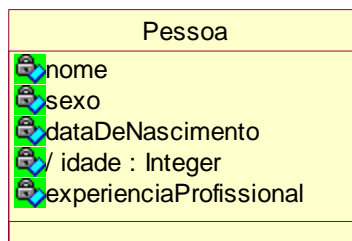


Figura 15 – Atributos de uma classe

Na figura 15 é fácil perceber que na classe Pessoa os atributos nome, sexo e dataDeNascimento podem ser mapeados para uma tabela pessoa cujas colunas são nome, sexo e dataDeNascimento. Em relação ao atributo idade, que é derivado, não há necessidade de armazená-lo. A idade é sempre calculada em função da data atual e da data de nascimento. Já em relação ao atributo experienciaProfissional, é importante perceber que é um par de informação, pois possui a forma valor e unidade de medida. Nesta situação o atributo experienciaProfissional poderia ser mapeado para duas colunas. Para exemplificar, uma pessoa poderia ter 10 anos de experiência e outra 5 meses. Por último é razoável citar que existem atributos que são transientes, ou melhor, não persistem no banco de dados.

Para nomear uma coluna, no caso de representar uma relação de um para um com o atributo, usa-se o mesmo nome do atributo. Para os casos de atributos compostos, utiliza-se a nomenclatura de cada elemento que o compõe para cada coluna. A figura 16 exibe este mapeamento.

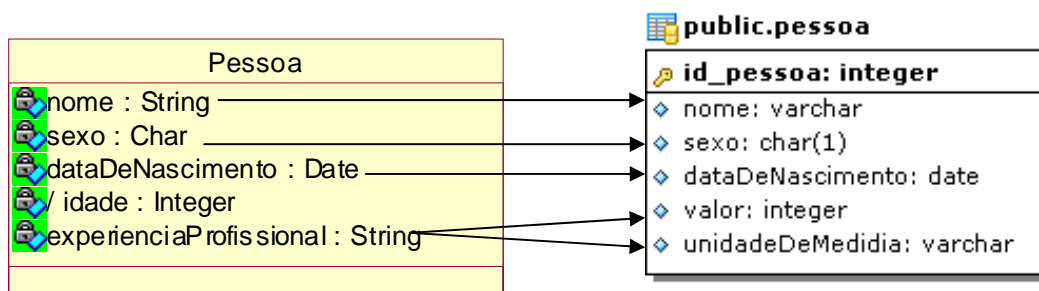


Figura 16 – Mapeamento do atributo para coluna

3.2.5 Mapeamento de Estruturas Hierárquicas

Generalização/Especialização, ou simplesmente herança, é um forma de relacionamento natural no modelo OO, porém no relacional não existe este mecanismo. Para simular o relacionamento de Generalização/Especialização no modelo relacional, de acordo com Ambler (2008) existem 3 principais soluções, além de uma quarta técnica, que vai além do mapeamento de herança. Estas técnicas são:

- Mapear toda a hierarquia de classes para uma única tabela;
- Mapear cada classe concreta para a sua própria tabela;
- Mapear cada classe para a sua própria tabela;
- Mapear as classes em uma estrutura genérica de tabelas;

Nas subseções a seguir são apresentadas brevemente cada uma destas estratégias.

3.2.5.1 Mapeamento de toda a hierarquia de classe para uma única tabela

Nesta estratégia todos os atributos de todas as classes que estão em uma hierarquia são mapeados para as colunas de uma única tabela (Ambler, 2008). A figura 17 exemplifica esta transformação. A hierarquia apresentada está presente no pacote Localidades da EDGV.

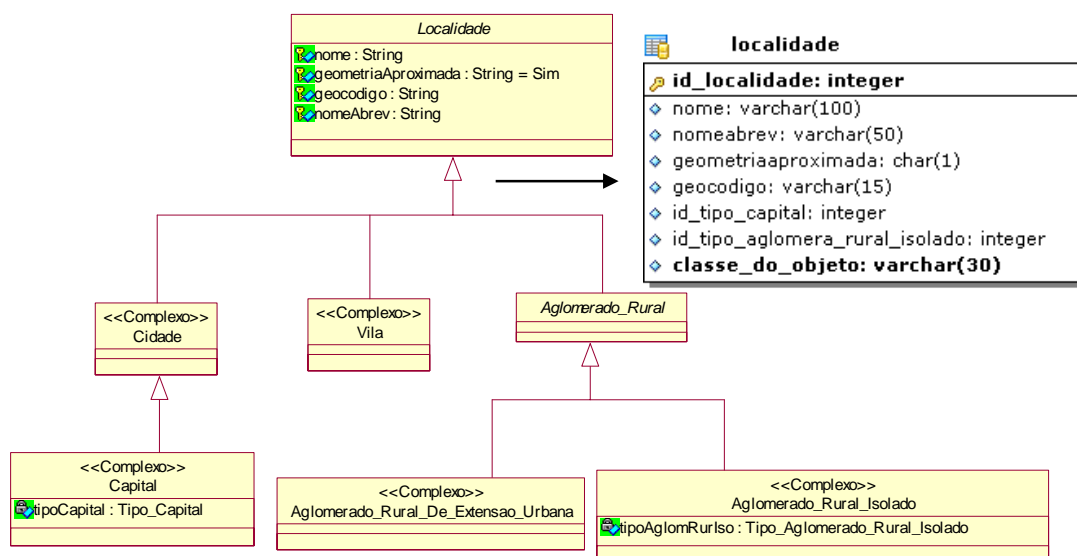


Figura 17 – Mapeamento de uma hierarquia de classes em uma única tabela
 Fonte: Adaptado das Especificações Técnicas da EDGV versão 2.0, 2007

Observando-se a figura 17, percebe-se certa diferença entre os atributos das classes e as colunas da tabela localidade. A primeira diferença é a coluna id_localidade que corresponde à chave primária, conforme mencionado na seção 3.2.2. A segunda diferença são as colunas id_tipo_capital e id_tipo_aglomera_rural_isolado, que expressam relacionamentos que serão apresentados na seção 3.2.6. A terceira diferença é a coluna denominada “classe do objeto” que é requerida para esta solução e tem como finalidade identificar o tipo de objeto utilizado, ou seja, a classe que originou a linha da tabela. Esta coluna deverá ser preenchida obrigatoriamente conforme as classes concretas da hierarquia.

3.2.5.2 Mapeamento de cada classe concreta para a sua própria tabela

De acordo Ambler (2008), nesta abordagem uma tabela é criada para cada classe concreta. Ainda segundo Fowler (2002), os atributos da classe ancestral são replicados para cada classe concreta na hierarquia. A figura 18 mostra esta situação

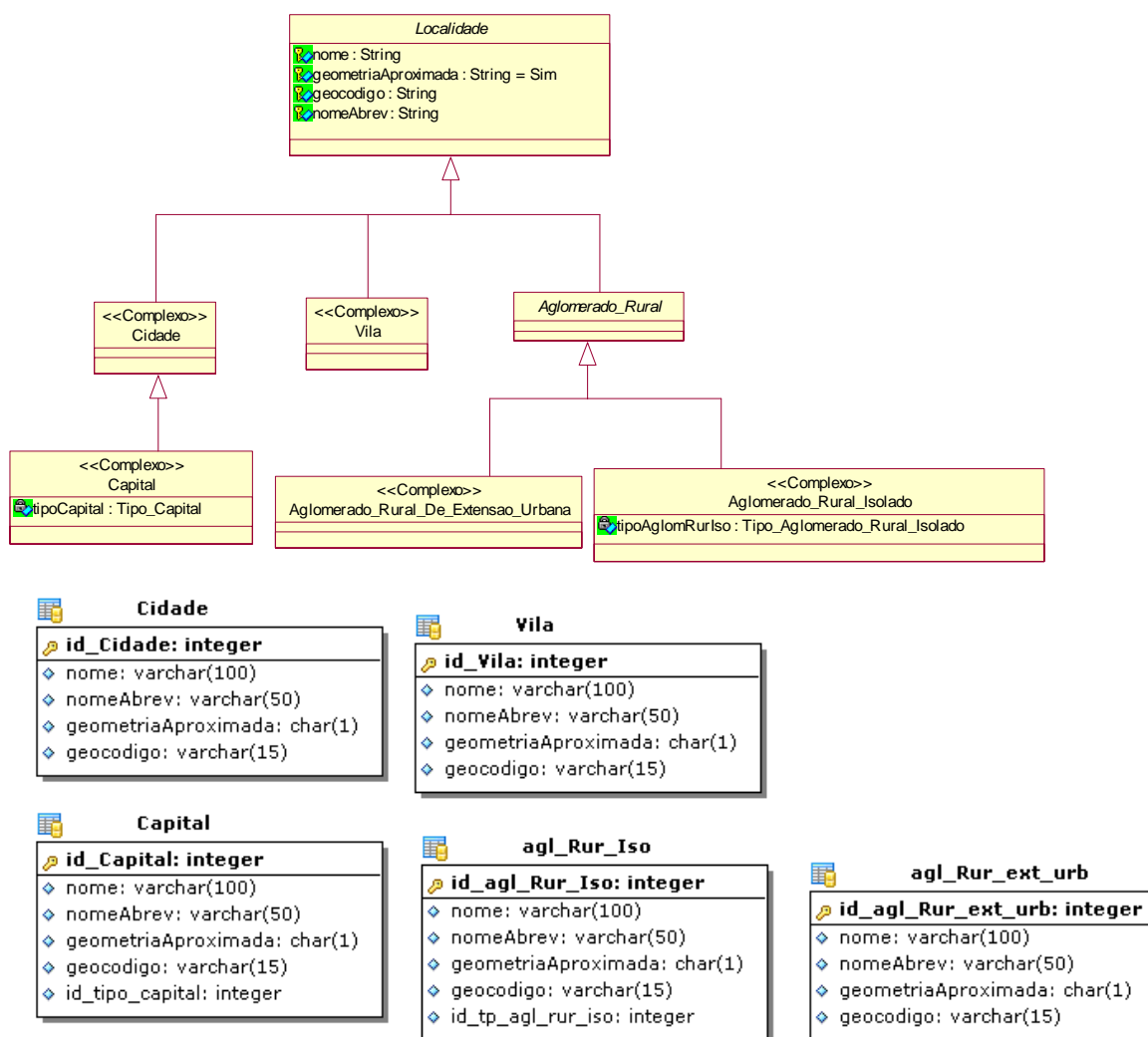


Figura 18 – Mapeamento de uma classe concreta para tabela
 Fonte: Adaptado das Especificações Técnicas da EDGV versão 2.0, 2007

Há que se observar que nesta modalidade foram criadas cinco tabelas correspondentes às cinco classes concretas da hierarquia. Note-se também que os atributos das classes ancestrais foram reproduzidos nas tabelas derivadas das classes concretas. No exemplo da figura 18, as únicas classes que não geraram tabela foram as classes *Localidade* e *Aglomerado_Rural*, por serem abstratas.

3.2.5.3 Mapeamento de cada classe para a sua própria tabela

Nesta solução para cada classe independentemente do fato de ser concreta ou abstrata, haverá uma tabela correspondente (Fowler, 2002). Desta forma os atributos de uma determinada classe estarão presentes somente em uma determinada tabela. Neste cenário, é imprescindível a junção entre tabelas (Fowler, 2002). A figura 19 ilustra esta situação.

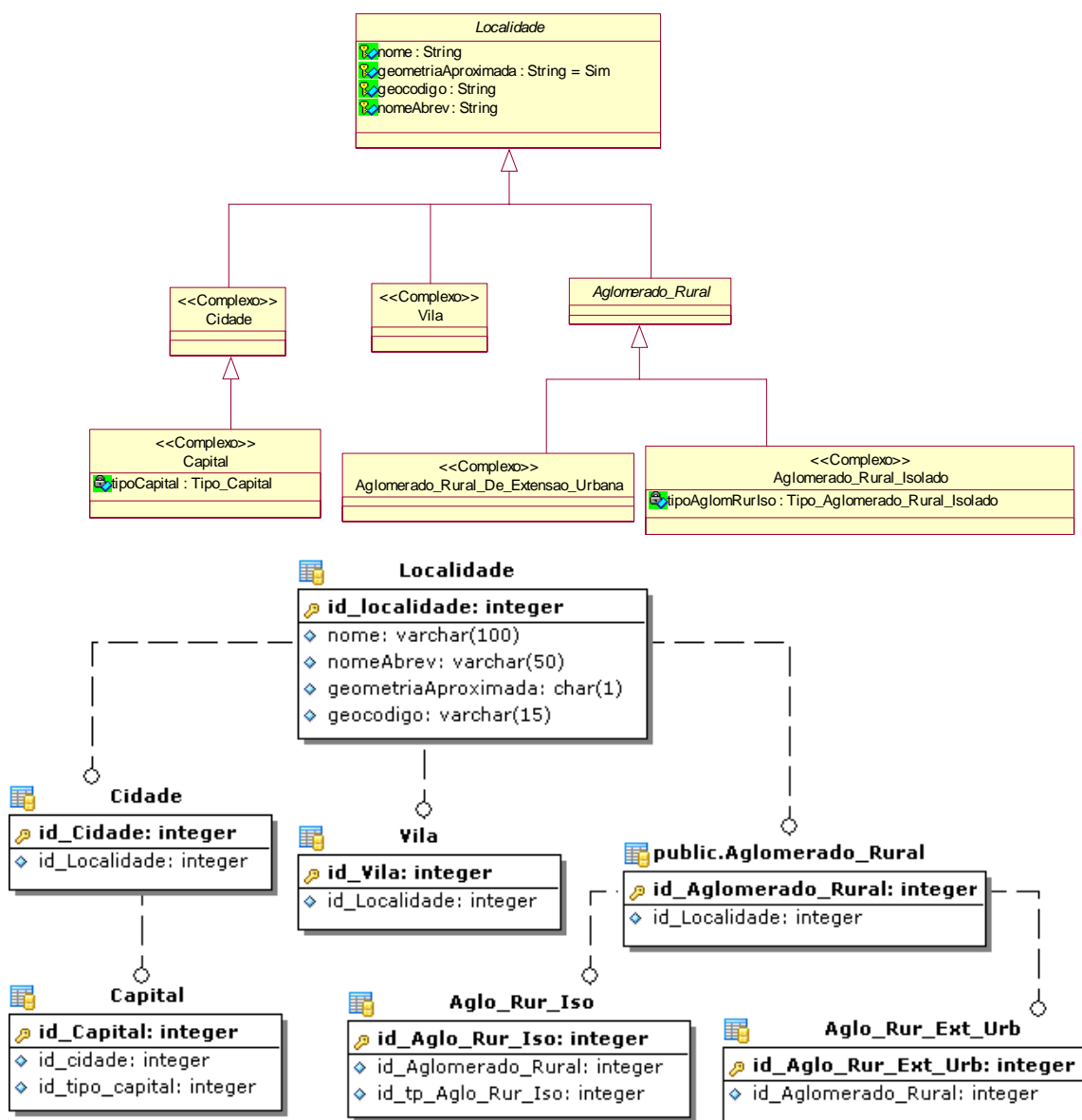


Figura 19 – Mapeamento de classe por tabela

Fonte: Adaptado das Especificações Técnicas da EDGV versão 2.0, 2007

Observe que foram criadas sete tabelas, uma para cada classe. Note também que para recuperar todas as informações de uma *Capital* é necessário fazer a junção da tabela *Capital* com a tabela *Cidade* e com a tabela *Localidade*.

3.2.5.4 Mapeamento das classes em uma estrutura genérica de tabelas

De acordo com Ambler (2008), uma quarta opção para o mapeamento das estruturas de herança em um banco de dados relacional é através de uma forma genérica, às vezes chamada de abordagem orientada a metadados (*meta-data driven approach*), para mapear suas classes. Esta abordagem não é específica à herança; ela suporta todas as outras formas de mapeamento para outras relações. A figura 20 representa esta modalidade.

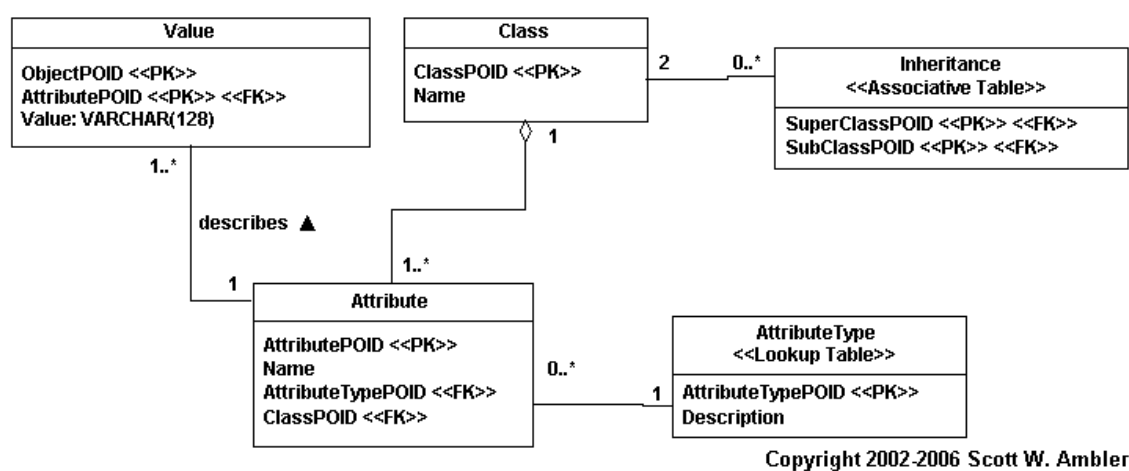


Figura 20 – Abordagem Orientada a Metadados
Fonte: Ambler, 2008

Nesta solução, as classes do modelo conceitual são mapeadas para um conjunto de tabelas de metadados, conforme a figura 20. Estas tabelas são organizadas conforme a estrutura de dados de uma classe, bem com os seus relacionamentos.

3.2.5.5 Comparação entre as estratégias de mapeamento

Nenhuma das estratégias de mapeamento anteriores se adequa a todas situações. Um estudo comparativo dessas estratégias é apresentada na tabela 7. Ambler(2008) e Fowler(2002) recomendam a seguinte opção: iniciar com a estratégia mapear toda a hierarquia de classe para uma única tabela e, depois, conforme necessidade, usar alguma outra estratégia para refatorar o modelo de dados.

Para o mapeamento do modelo conceitual da EDGV, no que se refere à generalização e especialização, será utilizado, inicialmente, o método de mapeamento denominado: “Mapear toda a hierarquia de classe para uma única tabela”.

Tabela 7 – Comparação de Estratégias para Mapeamento de Heranças

Estratégia	Vantagens	Desvantagens
Mapeamento toda a hierarquia de classes para uma única tabela	<p>Abordagem simples.</p> <p>Fácil de adicionar novas classes; só é preciso adicionar novas colunas na tabela.</p> <p>Fornecer suporte ao polimorfismo simplesmente mudando o tipo na linha da tabela.</p> <p>Acesso aos dados é rápido porque os dados estão em uma única tabela, sem a necessidade de junções.</p> <p>Uma consulta Ad-hoc é muito fácil, porque todos os dados são encontrados em uma tabela.</p> <p>O deslocamento de um atributo na hierarquia não afeta a tabela.</p>	<p>Acoplamento na hierarquia da classe é maior, porque todas as classes estão diretamente acopladas à mesma tabela.</p> <p>Uma mudança de uma classe pode afetar a tabela que pode afetar as outras classes na hierarquia.</p> <p>Potencialmente desperdício de espaço no banco de dados.</p> <p>Indicação do tipo se torna complexo quando existe sobreposição significativa entre os tipos.</p> <p>A tabela pode crescer rapidamente para grandes hierarquias.</p> <p>Os relacionamentos são condensados em uma tabela.</p>
Mapeamento de cada classe concreta para a sua própria tabela	<p>É fácil de fazer consultas ad-hoc, uma que vez todos os dados que são necessários sobre uma classe são armazenados em uma única tabela.</p> <p>Bom desempenho de acesso a um único objeto de dados.</p> <p>Cada tabela é acessada apenas quando essa classe é acessada, o que pode propagar o acesso à carga.</p>	<p>Quando se modifica uma classe precisa modificar a sua tabela e as tabelas de qualquer de suas subclasses.</p> <p>É difícil suportar vários papéis e ainda manter a integridade dos dados.</p> <p>Pode ser difícil manipular chaves primárias.</p> <p>Todos os relacionamentos da superclasse são condensados nas subclasses.</p>

Estratégia	Vantagens	Desvantagens
Mapeamento cada classe para a sua própria tabela	<p>A relação entre o modelo conceitual e o modelo do banco de dados é muito simples.</p> <p>Suporta polimorfismo muito bem; só é preciso ter registros nas tabelas adequadas para cada tipo.</p> <p>Muito fácil de modificar as superclasses e adicionar novas subclasses bastando modificar ou adicionar uma tabela.</p> <p>Os Dados crescem na proporção direta do crescimento do número de objetos.</p>	<p>É preciso tocar várias tabelas para carregar um objeto, o que significa uma junção ou múltiplas consultas. Potencialmente lento.</p> <p>A tabela pai pode tornar-se um gargalo porque precisa ser acessada com frequência.</p> <p>A alta normalização podem tornar difícil de compreender consultas ad hoc.</p> <p>Deslocamento de um atributo na hierarquia afeta a tabela.</p>
Mapear as classes em uma estrutura genérica de tabelas	<p>Funciona muito bem, quando o acesso de dados é encapsulado por uma sólida camada de persistência.</p> <p>Pode ser estendida para prover metadados e suportam uma vasta gama de mapeamentos, incluindo mapeamento de relacionamentos.</p> <p>É extremamente flexível, pois permite alterar rapidamente a maneira como os objetos são armazenados, bastando atualizar o meta modelo.</p>	<p>Técnica muito avançada que pode ser difícil de se implementar pela primeira.</p> <p>Funciona melhor para pequenas quantidades de dados, pois é necessário acessar muitas linhas de dados para construir um único objeto.</p> <p>Provavelmente será necessário construir uma pequena aplicação para manter a administração do meta modelo.</p> <p>Baixa Performance.</p>

3.2.5.6 Herança múltipla

Até este ponto, tem-se centrado sobre o mapeamento em hierarquias de herança simples. Segundo Ambler (1998) herança simples ocorre quando uma subclasse, como por exemplo, “Cidade” herda um ou mais atributos e ou operações diretamente, a partir de uma única classe pai, como “Localidade”. Herança múltipla ocorre quando uma subclasse possui diretamente duas ou mais superclasses.

Existem alguns casos de herança múltipla no modelo conceitual da EDGV. Para mapeá-los, também, basta seguir um ou mais dos métodos citados anteriormente. A figura 21 sinaliza esta situação usando o método de hierarquia por tabela. Esta hierarquia está presente no pacote “Hidrografia” da EDGV.

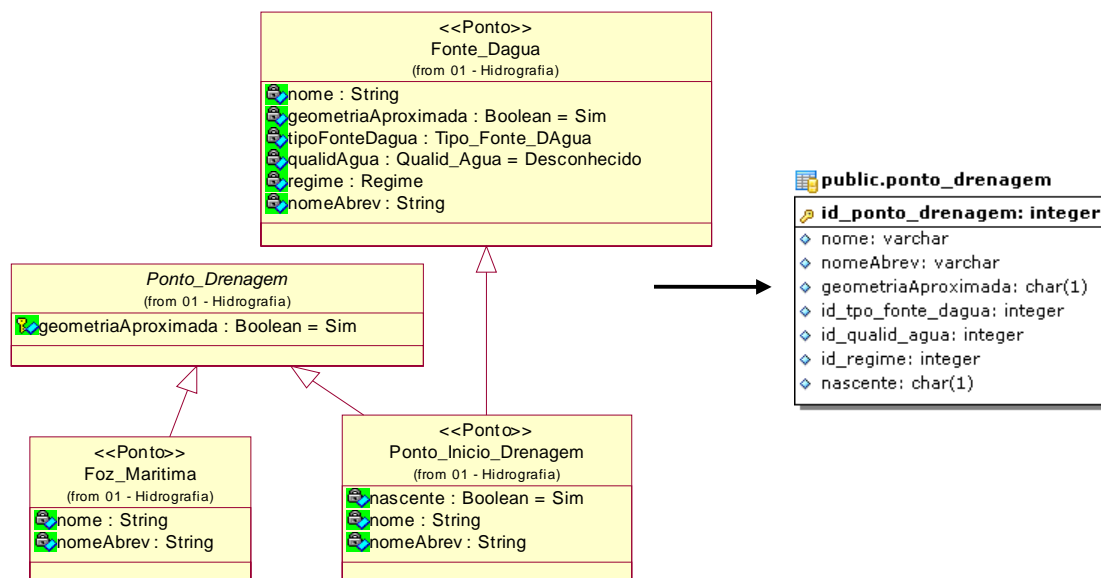


Figura 21 – Mapeamento de uma hierarquia com herança múltipla para uma tabela
 Fonte: Adaptado das Especificações técnicas da EDGV versão 2.0, 2007

3.2.6 Mapeamento dos Relacionamentos

De acordo com Booch, Rumbauch e Jacobson (2000) na modelagem orientada a objetos, as associações são relacionamentos estruturais entre instâncias. As associações podem ser de três tipos: associação simples, agregação e composição.

- Associação simples conceitualmente é apenas uma maneira de representar relacionamentos de forma normal entre um conjunto de vínculos de um modelo.
- Agregação é um tipo de relacionamento em que se diz: “parte de, composto de etc”. É referenciado com “Todo-Parte”. É inteiramente conceitual e nada faz além de diferenciar “o todo da parte”. Os objetos partes podem estar em mais de um todo.
- Composição é uma agregação mais forte. Os tempos de vida do todo e das partes são coincidentes. Não é possível compartilhar as partes com outros objetos. A figura 22 representa estes tipos de relacionamento.

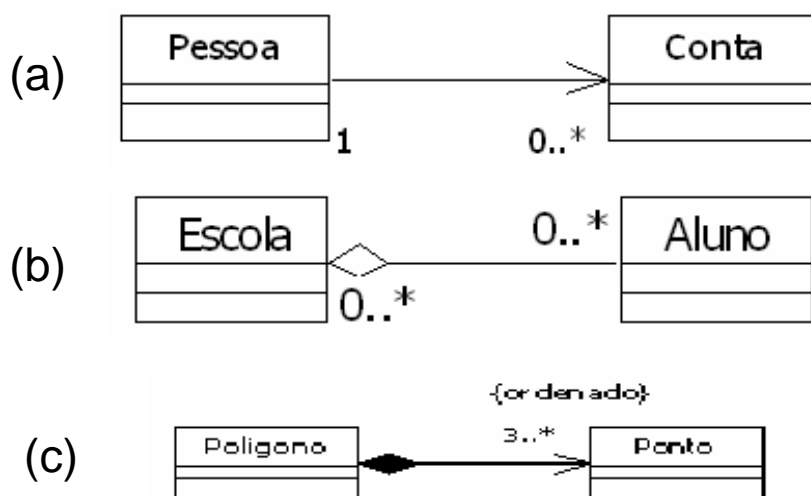


Figura 22 – (a) Associação simples; (b) agregação e; (c) composição.

É oportuno ainda lembrar que no modelo conceitual da EDGV existe um tipo especial de relacionamento que mapeia relacionamentos espaciais. Estes relacionamentos implementam relações topológicas. A figura 23 exhibe um relacionamento espacial. Note que o relacionamento está estereotipado com a palavra `<<espacial>>`.

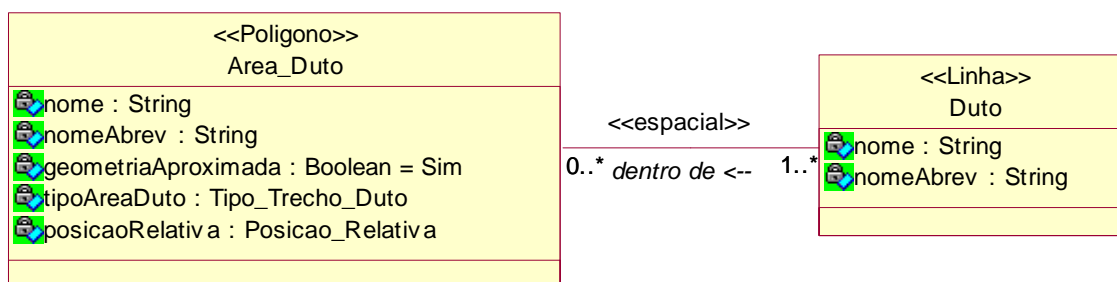


Figura 23 – Relacionamento espacial

Segundo Ambler (2008) relacionamentos no banco de dados são mantidos através do uso de chaves estrangeiras. Uma chave estrangeira é um atributo de dado que aparece em uma tabela que pode ser parte ou coincidente com a chave primária de outra tabela. Uma chave primária refere-se à tuplas (conjuntos) de um ou mais campos, cujos valores, considerando a combinação

de valores de todos os campos da tupla, nunca se repetem, nem são nulos. Uma chave primária, portanto, identifica univocamente uma linha em uma tabela.

Os relacionamentos entre tabelas no banco de dados relacional utilizam os mecanismos de restrição de integridade. De acordo com Elmasri e Navathe (2004), os principais tipos de restrições de integridade que ocorrem freqüentemente na modelagem de banco de dados convencionais são restrições de domínio, de chave, de entidade, de integridade referencial e de integridade semântica.

É preciso mapear os relacionamentos do modelo OO para o modelo relacional. Por exemplo, Associação simples, agregação e composição são formas de relacionamentos entre classes. Nos três casos, a implementação no banco de dados é idêntica. Porém, no caso mais específico da composição, onde as partes só existem enquanto o todo existir, no modelo relacional este mecanismo pode ser efetuado colocando na chave estrangeira uma restrição de integridade tal como: “*on delete cascade*”.

A multiplicidade de um relacionamento, tanto OO quanto relacional, pode ser: um para um, um para muitos (muitos para um) e muitos para muitos. Já em relação à direcionalidade do relacionamento do modelo OO, pode haver unidirecionalidade, quando um objeto tem conhecimento do outro que está na outra ponta. A recíproca não é verdadeira. Um relacionamento é bidirecional quando um primeiro objeto em uma ponta tem conhecimento de um segundo objeto na outra ponta e vice-versa. Ambler (2008) afirma que a direcionalidade do relacionamento no modelo relacional sempre é bidirecional, já que o mecanismo de chaves estrangeiras faz uso de junção de tabelas. A figura 24 aponta os relacionamentos com suas multiplicidades e direcionalidades do modelo orientado a objetos.

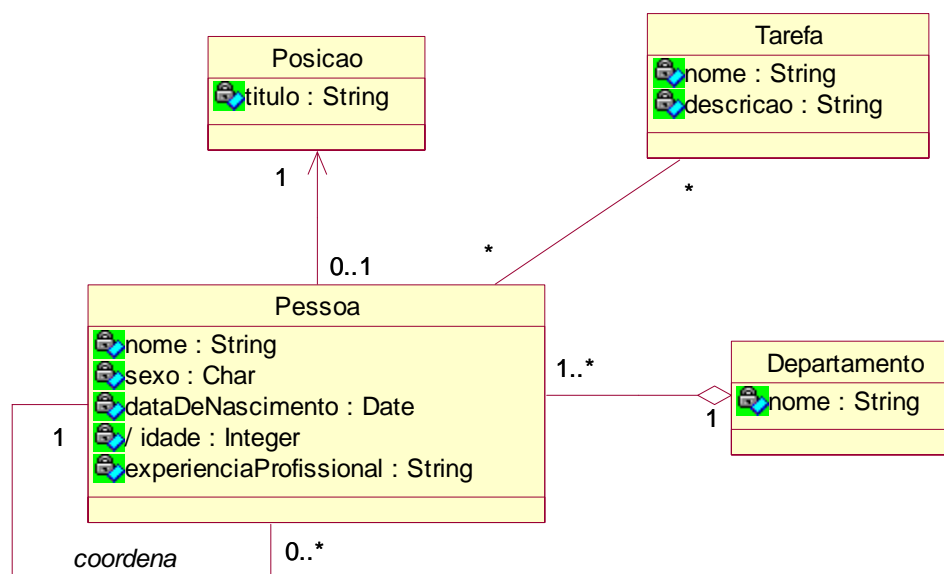


Figura 24 – Multiplicidade e direcionalidade

É fácil perceber através da figura 24 que a associação entre a classe Pessoa e Posicao tem multiplicidade um para um e é unidirecional, pois uma pessoa sabe sua posição, mas a posição não tem conhecimento da pessoa a que está ligada. O relacionamento entre a classe Pessoa e Departamento tem multiplicidade um para muitos e é bidirecional, já que uma pessoa conhece seu departamento e um departamento agrega pessoas a si. O relacionamento entre Pessoa e Tarefa tem multiplicidade muitos para muitos e é bidirecional. Por último há que se destacar um auto-relacionamento na classe Pessoa. Neste caso, uma pessoa pode coordenar várias pessoas e sabe quem são elas. Estas pessoas também sabem quem é o seu coordenador. A figura 25 apresenta este modelo implementado segundo a abordagem relacional.

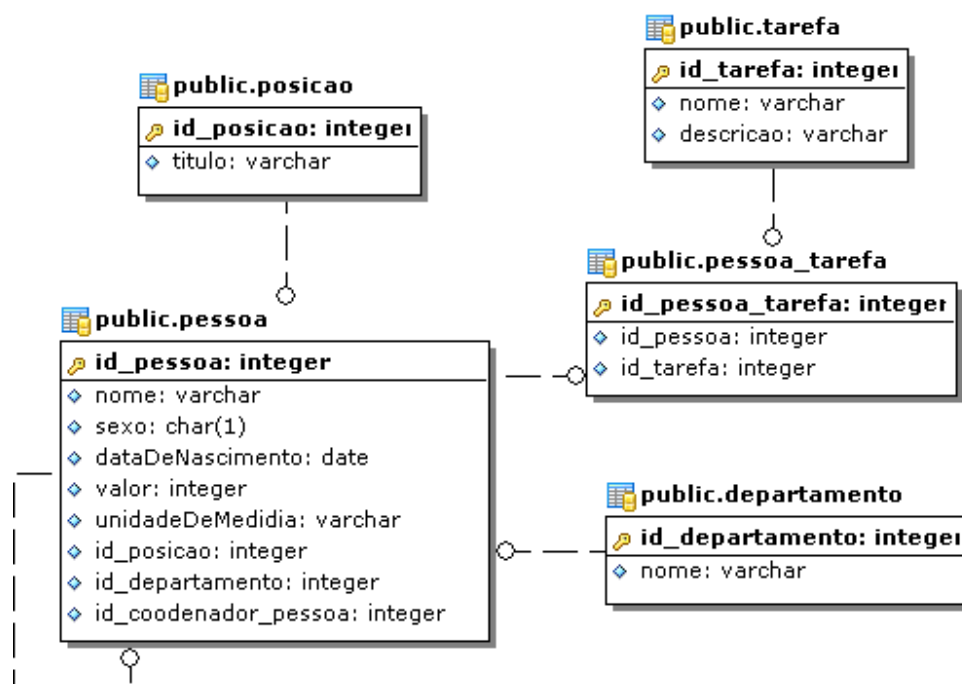


Figura 25 – Implementação dos diversos tipos de associação segundo o modelo relacional

O relacionamento entre as tabelas `pessoa` e `posicao` é criado através da chave estrangeira `id_posicao` da tabela `pessoa`. Há que se observar que, da mesma maneira, a coluna `id_departamento` da tabela `pessoa` é uma chave estrangeira para a tabela `departamento`. No caso do relacionamento entre `pessoa` e `tarefa` foi necessário criar uma tabela associativa, pois o modelo relacional não suporta naturalmente relacionamentos de muitos para muitos, necessitando de uma tabela intermediária. O auto-relacionamento da tabela `pessoa` é criado através da chave estrangeira `id_coordenador_pessoa` na própria tabela `pessoa`.

Cabe ainda mencionar sobre o padrão utilizado para nomear a chave estrangeira e a tabela associativa. Para a tabela que tem chave estrangeira utiliza-se o mesmo nome da chave primária da tabela que está sendo referenciada. No caso de mais de um relacionamento ou auto-relacionamento, compõe-se o nome da chave estrangeira com o papel exercido no relacionamento, respeitando-se o limite do tamanho do nome da coluna de até 30 caracteres. As figuras 24 e 25 explicitam a situação de um auto-relacionamento.

Para a tabela associativa, faz-se uma composição dos nomes das tabelas que realizam o relacionamento, lembrando o limite de 27 caracteres para o nome de tabela.

Os relacionamentos não-espaciais do modelo conceitual da EDGV são implementados no modelo relacional seguindo o padrão apresentado acima.

3.2.7 Relacionamentos Espaciais

No modelo da EDGV, além do tipo de relacionamento padrão da orientação a objeto, há o relacionamento espacial (estereótipo <<espacial>>). Este tipo de relacionamento pode ser implementado diretamente através das operações espaciais dos SGBD, sem necessitar da definição de relacionamento entre as respectivas tabelas. Um exemplo é apresentado na figura 26, onde há duas classes se relacionando espacialmente. Estas classes pertencem ao pacote Hidrografia.

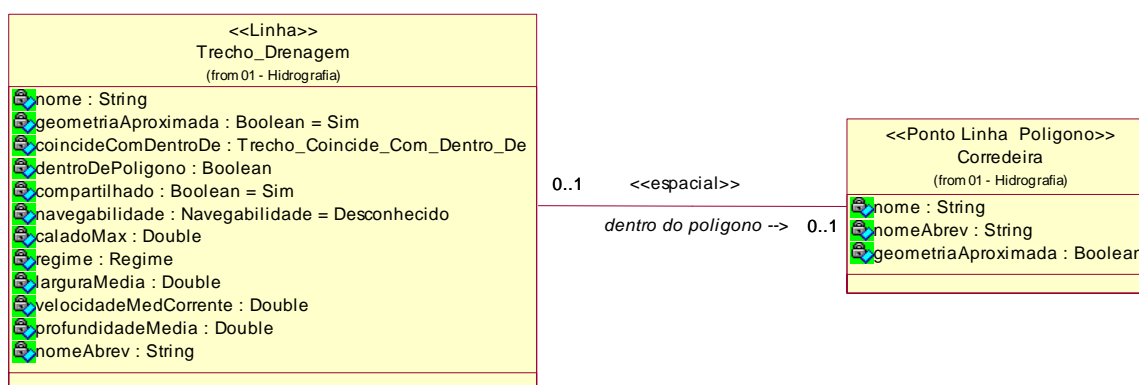


Figura 26 – Relacionamento espacial entre Trecho_Drenagem e Massa_Dagua
Fonte: Adaptado das especificações técnicas da EDGV versão 2.0, 2007

Neste exemplo, zero ou 1 trecho de drenagem pode estar dentro de uma corredeira. Observa-se que este relacionamento é espacial, ou seja, através de uma operação espacial é possível saber qual trecho de drenagem está dentro de uma corredeira. Assim, no banco de dados, não seria necessário criar o relacionamento entre as tabelas. A figura 27 mostra que a tabela trecho_drenagem não tem chave estrangeira para a tabela corredeira e vice-versa.

edgv.trecho_drenagem	edgv.corredeira
<p>id_trecho_drenagem: integer</p> <ul style="list-style-type: none"> ◆ nome: varchar(100) ◆ nomeabrev: varchar(50) ◆ geometriaaproximada: char(1) ◆ dentrodepoligono: char(1) ◆ compartilhado: char(1) ◆ caladomax: double precision ◆ larguramedia: double precision ◆ velocidademedcorrente: double precision ◆ profundidadedia: double precision ◆ id_trecho_coincide_com_dentro: integer ◆ id_navegabilidade: integer ◆ id_regime: integer ◆ id_trecho_curso_dagua: integer ◆ geom: Linestring 	<p>id_corredeira: integer</p> <ul style="list-style-type: none"> ◆ nome: varchar(100) ◆ nomeAbrev: varchar(50) ◆ geometriaAproximada: char(1) ◆ geom: public.geometry

Figura 27 – Relacionamento espacial entre trecho de drenagem e corredeira

É importante reparar que na tabela trecho_drenagem não há chave estrangeira para a tabela corredeira e vice-versa, o que ratifica que este relacionamento é implementado por uma operação espacial.

3.2.8 Mapeamento das classes de Feição

Os Sistemas Gerenciadores de Banco de Dados atuais estão disponibilizando mecanismos para que não somente a informação alfanumérica seja armazenada. Este é o caso, por exemplo, do SGBD PostgreSQL com o cartucho espacial PostGIS, conforme visto na capítulo 3, que disponibiliza meios para o armazenamento da parte geométrica, ou seja, pode-se armazenar dados alfanuméricos e geoespaciais em uma mesma estrutura de dados.

Conforme o modelo proposto pela OGC há uma variedade de elementos geométricos que pode ser armazenada. No caso do E-FOTO, no módulo de restituição fotogramétrica, existe a possibilidade de lidar com três tipos de geometria, a saber: Ponto, Linha e Polígono.

As classes de feição da EDGV foram modeladas com os tipos de geometria Ponto, Linha e Polígono. No modelo conceitual da EDGV, as classes de feição podem ser divididas em dois

grupos: Feição com uma única representação da geometria associada e Feição com mais de uma representação da geometria.

3.2.8.1 Feição com uma única representação

Como o próprio nome diz, a feição é representada com somente um tipo de geometria. Para mapear a classe de feição para um SGBD que suporta colunas geométricas, basta adicionar na tabela uma coluna geométrica correspondente ao tipo da geometria da feição. A figura 28 ilustra esta informação.

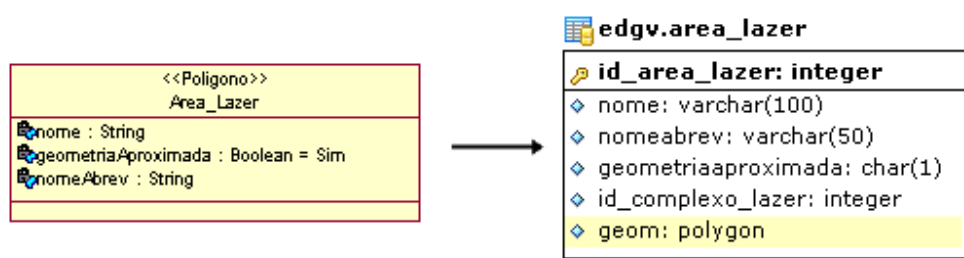


Figura 28 – Classe Area_lazer com representação de polígono

3.2.8.2 Feição com mais de uma representação

Existem certos elementos da EDGV que estão associados mais de uma representação geométrica. Para mapear estes elementos para um SGBD que suporta colunas geométricas, basta definir uma coluna geométrica como do tipo *Geometry*, conforme o modelo geométrico do SFA. A figura 29 ilustra esta situação.

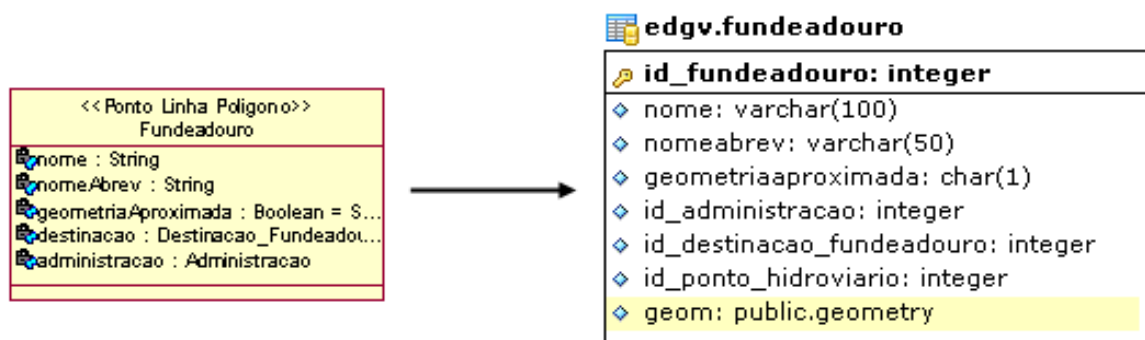


Figura 29 – Elemento com mais de uma representação geométrica

Observe-se que na classe Fundeadouro existe um estereotipo (<<Ponto Linha Poligono >>) indicando sua representação geométrica e que na tabela fundeadouro a coluna geom é definida com o tipo *geometry*, permitindo assim abstrair-se do elemento que será efetivamente armazenado, ou seja, pode-se armazenar qualquer geometria definida no modelo geométrico da SFA na seção 2.2.2.1.1.

3.2.8.3 A falta do conceito da abstração de alguns softwares

Alguns softwares, proprietários ou livres, não possuem a capacidade de lidar com mais de uma forma geométrica para uma dada feição. Este é o caso, por exemplo, do ArcGIS que, ao criar um elemento (Classe de feição), precisa definir concretamente que tipo de geometria (Ponto, Linha, Polígono ou outro tipo qualquer) será utilizado. Não é possível definir um tipo abstrato como, por exemplo, *Geometry*. Uma solução para resolver este problema seria criar uma tabela para representar somente os dados alfanuméricos e outras três representando somente a parte geométrica e, depois, criando-se o relacionamento entre a tabela alfanumérica e as tabelas de geometria. Este modelo não é o ideal, porém resolve o problema da representação geométrica. A figura 30 reflete esta situação.

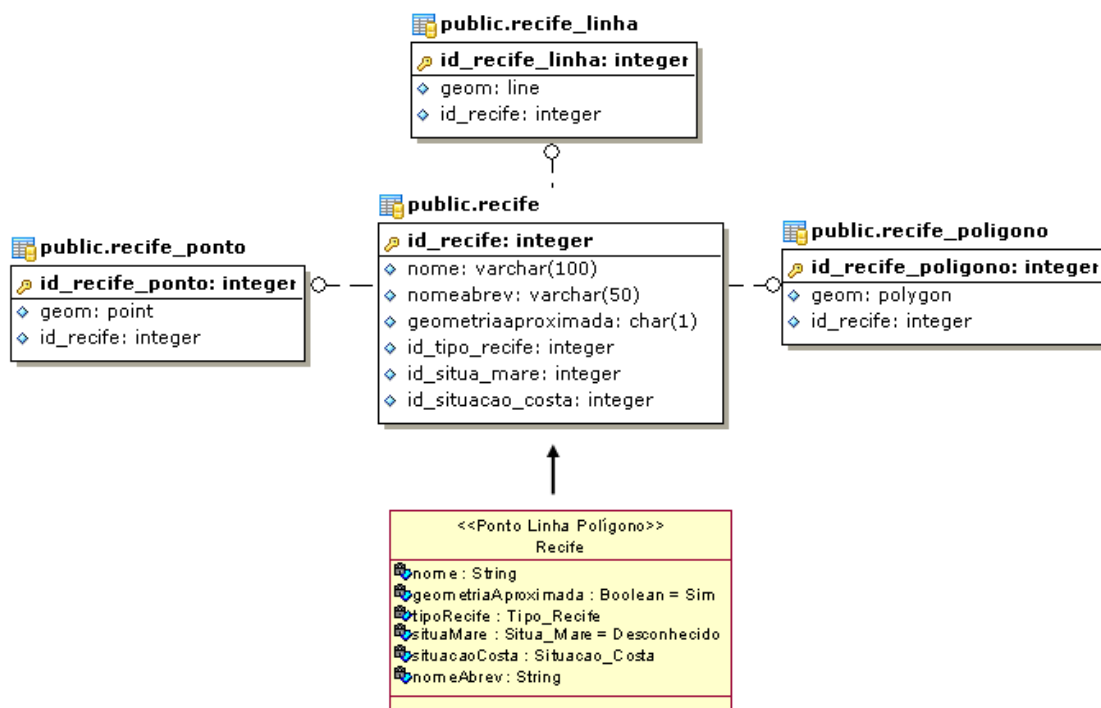


Figura 30 – A Falta do tipo *Geometry*.

Na figura 30 a classe Recife, que pode ter três representações, foi mapeada para uma tabela chamada recife e sem geometria. Para armazenar a componente geométrica foram criadas três tabelas, que foram relacionadas com a tabela recife.

Ainda há uma outra solução variante que seria apenas criar três tabelas com suas respectivas geometrias e repetir os atributos alfanuméricos para cada uma delas. A figura 31 ilustra esta situação.

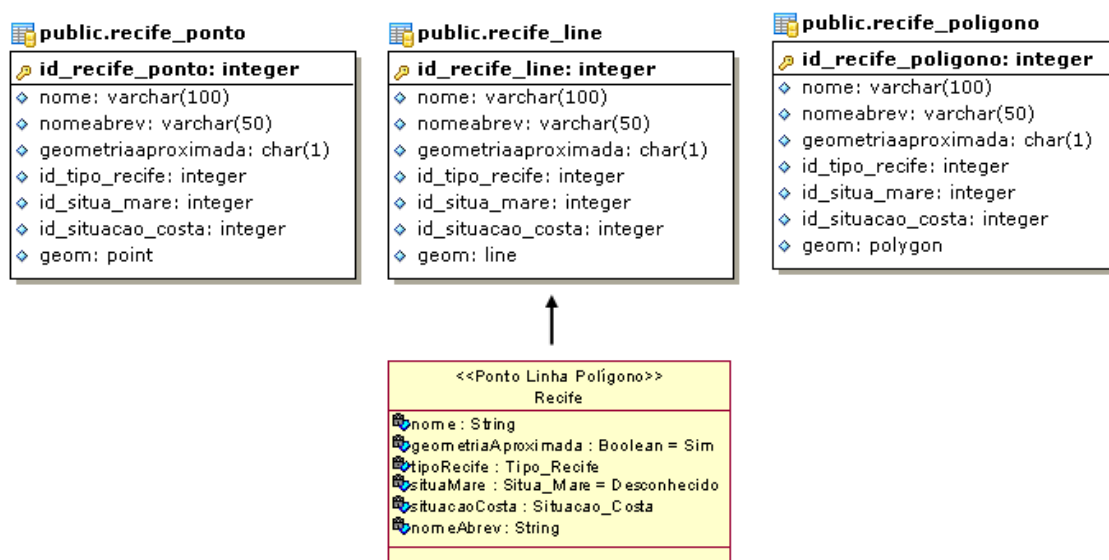


Figura 31 – A Classe Recife mapeada para três tabelas.

É importante perceber que, na solução apontada na figura 31 qualquer alteração na classe Recife impacta na alteração das três tabelas, seja a alteração de um atributo ou de um relacionamento.

3.2.9 Mapeamento das classes Agregadoras

Conforme explicado anteriormente, classes agregadoras não possuem geometria, mas são referenciadas por classes de feição e podem referenciar classes de domínio. A finalidade de uma classe agregadora é poder reunir um grupo de feições que, juntas, constituem um elemento maior e importante para a representação. A figura 32 exemplifica um caso de um elemento agregador com seus agregados. A figura 33 exibe o resultado do mapeamento para o modelo relacional.

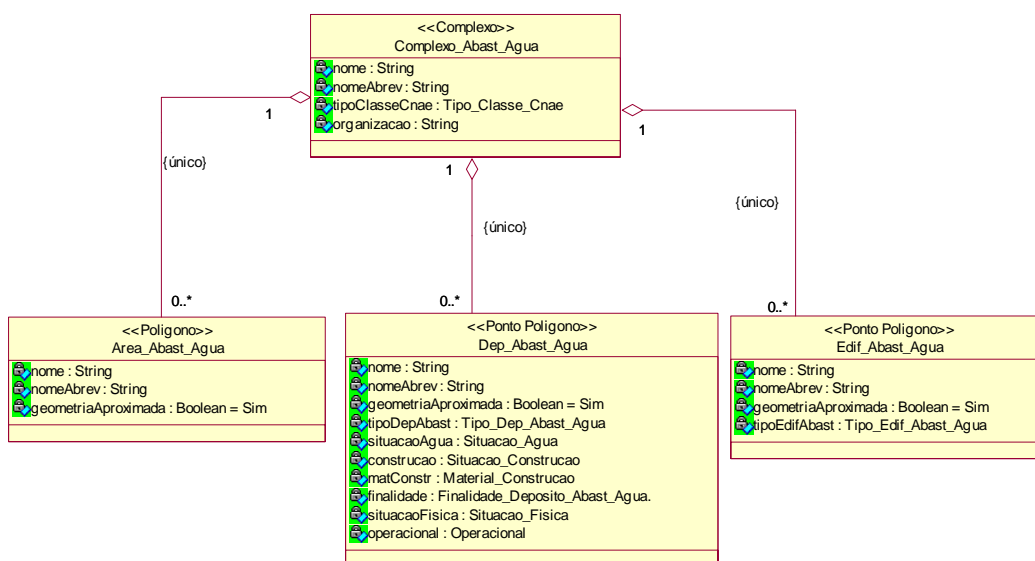
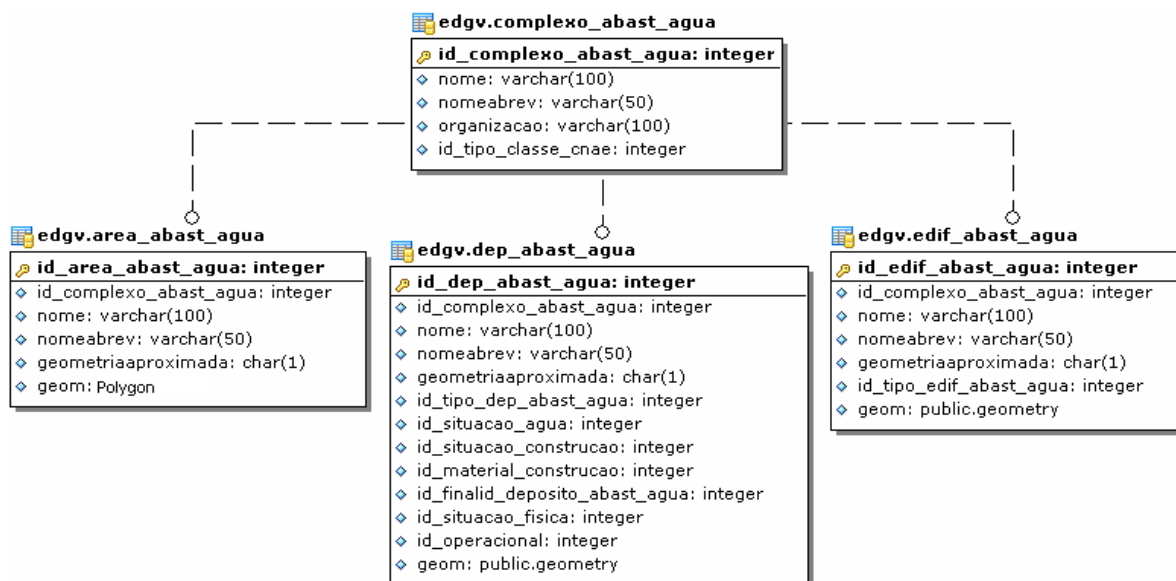


Figura 32 – Elemento agregador com seus agregados

Há que se destacar que a classe `Complexo_abast_agua`, elemento agregador, não possui geometria; porém as classes `Area_Abast_Agua`, `Dep_Abast_Agua` e `Edif_Abast_Agua` têm geometria e são consideradas elementos agregados. Note-se o estereotipo destas três últimas confirmando que são elementos com representação geométrica.

Figura 33 – Relacionamento entre as tabelas agregadora e suas agregadas



Observa-se na figura 33 que as tabelas que possuem uma coluna geométrica, geom, também têm uma chave estrangeira para a tabela, agregadora, complexo_abast_agua.

3.2.10 Mapeamento das classes de Domínio

As classes de Domínio descrevem um conjunto de valores que corresponde a um determinado intervalo de valores. Cada classe de domínio é mapeada para uma tabela. Por exemplo, a classe Operacional possui os seguintes valores de domínio: Sim, Não e Desconhecido. A tabela operacional terá para cada linha um valor corresponde ao domínio. A figura 34 exemplifica este caso.

Uma importante informação é que as classes de Domínio não estão explícitas nas especificações técnicas da EDGV. Elas estão presentes, de maneira implícita, na Relação de Classes de Objeto. A CONCAR disponibiliza no padrão do software Rational Rose, toda a modelagem conceitual, incluindo as classes de Domínio.

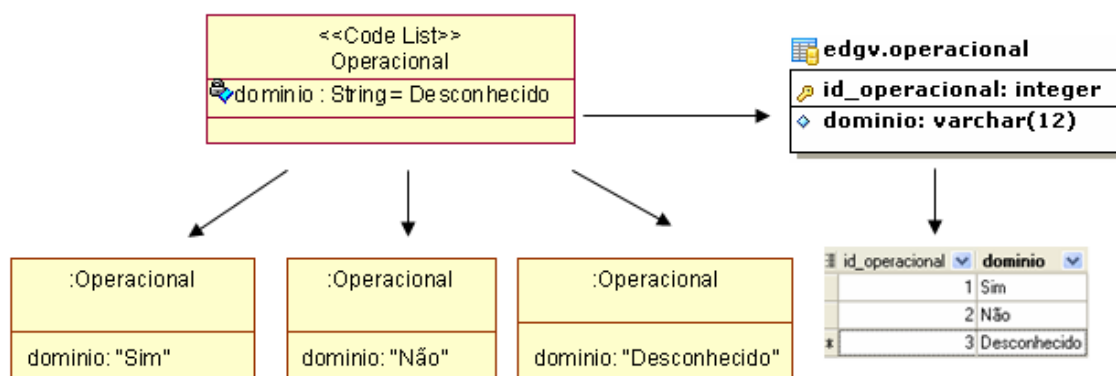


Figura 34 – Classe de Domínio com seus valores e tabela com suas linhas

Na figura 34 são exibidos três objetos da classe Operacional e as linhas correspondentes a cada objeto, na tabela operacional.

3.2.11 Geração de Scripts para Criação das Tabelas

Um pré-requisito para concretizar esta etapa foi uma análise detalhada das classes e dos relacionamentos do modelo conceitual da EDGV.

Uma vez definido o processo de mapeamento do modelo OO para o modelo relacional, volta-se o esforço para a criação de scripts em SQL para criação das tabelas no SGBD. Para criar as tabelas derivadas do modelo conceitual da EDGV, no PostgreSQL\PostGIS, os seguintes passos foram executados:

- (1) desenvolvimento e execução de um script para a criação das tabelas de feição; (2) adição de uma coluna, através da função *AddGeometryColumn*, para armazenar a geometria das feições nas tabelas criadas no passo 1. Esta função adiciona informações na tabela de metadados;
- (3) desenvolvimento e execução de um script para a criação das tabelas derivadas das classes agregadoras e de domínio;
- (4) desenvolvimento e execução de um script de restrição de integridade para gerar os relacionamentos entre as tabelas.

Este mesmo procedimento poderia ser adotado para qualquer outro SGBD. O material em mídia contém estes scripts.

3.3 **Considerações Adicionais sobre a Abordagem Relacional da EDGV**

Este capítulo abordou dois aspectos, o primeiro foi a seleção do SGBD, na qual o modelo relacional sagrou-se eleito. Ainda é importante mencionar que o SGBD PostgreSQL com o cartucho espacial PostGIS será utilizado no contexto do projeto E-FOTO para manter-se também alinhado com a filosofia de Software Livre (Licença GNU\GPL).

Um segundo aspecto muito importante foi a metodologia empregada para realizar o mapeamento do modelo orientado a objetos para o relacional. Para isto, apresentou-se uma

solução para mapear a EDGV no modelo relacional. Nesse contexto, os relacionamentos do tipo composição, agregação e associação simples do modelo orientado a objetos da EDGV tornaram-se, no modelo relacional, relacionamentos normais entre tabelas, com suas respectivas restrições de integridade.

Cada conjunto de classes do relacionamento de generalização/especialização condensou-se em uma tabela com uma coluna especial (`classe_do_objeto`) para identificar a classe correspondente. Ainda na generalização/especialização os relacionamentos das subclasses foram condensados nesta tabela.

Os relacionamentos da EDGV estereotipados de espacial não produziram qualquer efeito no modelo relacional, uma vez que são derivados de funções espaciais. O produto obtido foi um esquema de banco de dados geográficos que será utilizado no software E-FOTO para o armazenamento das feições obtidas por intermédio da restituição fotogramétrica digital.

Como consequência, este mapeamento disponibiliza um padrão de esquema em banco de dados relacional com extensão espacial, permitindo assim uma interoperabilidade em nível de base de dados. Acredita-se que esta solução de geração de esquema possa ser utilizada por qualquer instituição que pretenda realizar seus processos de mapeamento cartográfico em conformidade com a EDGV definida pela CONCAR.

4 O MÓDULO DE RESTITUIÇÃO FOTOGRAMÉTRICA DIGITAL DO E-FOTO

O projeto E-FOTO tem como um de seus objetivos disponibilizar uma EFD Educacional segundo o paradigma de software livre. Em linhas gerais, EFDs executam diferentes tarefas do fluxo de trabalho fotogramétrico, cada uma realizada por um diferente subsistema da EFD.

Uma tarefa importante para a produção de dados cartográficos é a Restituição Fotogramétrica Digital, que permite a extração de feições cartográficas tridimensionais a partir de um modelo estereoscópico. Esta operação engloba, basicamente, a representação das feições naturais ou artificiais presentes no modelo estereoscópico. Para a representação das feições de interesse, a EFD do E-FOTO dispõe de 3 primitivas gráficas: ponto, linha e polígono.

4.1 Novas Demandas e Novos Desafios

A promulgação das Especificações Técnicas para Estruturação de Dados Geoespaciais Vetoriais, Norma Cartográfica elaborada pela CONCAR e a falta de ferramentas que implementassem esta norma motivaram a extensão do módulo de restituição fotogramétrica digital 3D, para permitir realizar a restituição aderente a este novo padrão. Para atender a esta demanda, foram realizadas as seguintes atividades, na presente dissertação: (1) análise criteriosa da EDGV; (2) definição de novos requisitos para o módulo Restituição Fotogramétrica Digital; (3) a definição de um SGBD; (4) mapeamento do modelo conceitual da EDGV para um modelo de dados; (5) projeto e; (6) implementação de um protótipo.

4.2 Novos Requisitos para o Módulo Estereoplotter

Que requisitos são imprescindíveis para poder realizar a restituição em conformidade com a EDGV? As seções a seguir tratam deste assunto.

4.2.1 Requisitos Funcionais

Dentre os principais requisitos da extensão do módulo de restituição, o mais importante, sem dúvida, é: possibilitar que o E-FOTO execute a restituição fotogramétrica digital, no formato EDGV. A figura 35 explicita esta situação. Para que isto seja possível é preciso classificar os elementos geométricos restituídos no E-FOTO, conforme as primitivas geométricas (ponto, linha e polígono), consoante os elementos cartográficos pertencentes a EDGV, mais precisamente em seu modelo relacional. A EDGV também contempla outros tipos de objetos, como é o caso dos objetos derivados das classes de Domínio e Agregadoras. Neste sentido, um outro requisito é possibilitar aos elementos cartográficos que possuem geometria, associarem-se aos elementos do tipo Domínio e Agregador.

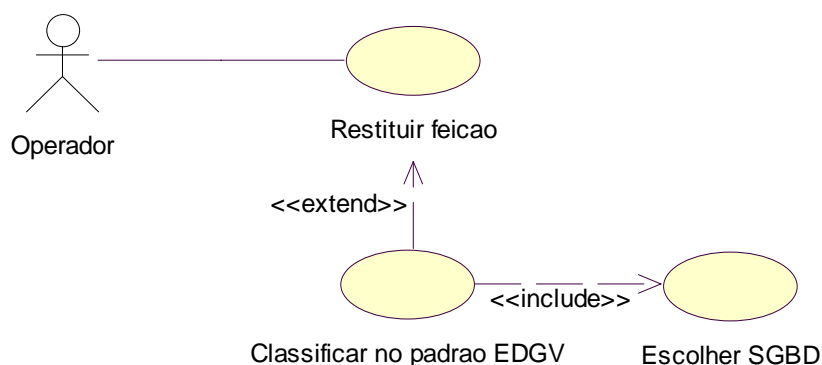


Figura 35 – Diagrama de caso de uso da Restituição fotogramétrica Digital na EDGV

4.2.2 Requisitos Não-Funcionais

No software E-FOTO os novos recursos não poderiam trazer como prejuízo, tornarem-se pré-requisito para a sua utilização. Então, um requisito muito importante é que as modificações efetuadas no módulo de restituição fotogramétrica digital 3D não acarretassem em uma restrição de seu uso.

Um outro fator é que o projeto E-FOTO está em curso, implicando em melhorias e atualizações constantes. Estas evoluções também não poderiam criar novas restrições para o uso da EDGV no software E-FOTO. Na verdade, o que se buscou no E-FOTO foi o baixo acoplamento com a EDGV, possibilitando assim a sua independência e evolução. Ainda é pertinente mencionar que a extensão deve ser capaz de continuar a realizar a restituição fotogramétrica mesmo com as mudanças que irão ocorrer na EDGV.

Um outro fator importante é que o modelo de dados da EDGV pode estar mapeado em tecnologia de SGBDR/SGBDOR diferente. Neste trabalho, o E-FOTO utiliza como primeira tecnologia de SGBD o PostgreSQL com o cartucho espacial PostGIS. Ainda é oportuno mencionar que a implementação foi realizada de maneira a possibilitar sua extensão para outros SGBDs, por exemplo, para o Oracle.

4.2.3 Modelo de Classes da Extensão

O primeiro requisito necessário para possibilitar ao E-FOTO realizar restituição fotogramétrica para EDGV, é ter o modelo conceitual da EDGV disponibilizado em um SGBD. Este primeiro requisito foi satisfeito, após uma análise minuciosa do modelo da EDGV e pela execução de todo o processo de mapeamento apresentado no capítulo 3. Nesta etapa foram desenvolvidos scripts em SQL que permitiram a criação do esquema no banco de dados PostgreSQL com o cartucho espacial PostGIS.

Um segundo requisito foi permitir ao E-FOTO reconhecer e interagir com esta base de dados que, agora, se encontra aderente ao modelo da EDGV, respeitando as diferenças entre um modelo orientado a objeto e o relacional. Para que isto se tornasse uma realidade, um conjunto de novas classes foi agregado ao E-FOTO. Este conjunto de classes é a parte nuclear que permite esta integração. A figura 36 ilustra o diagrama de classes nuclear (metamodelo).

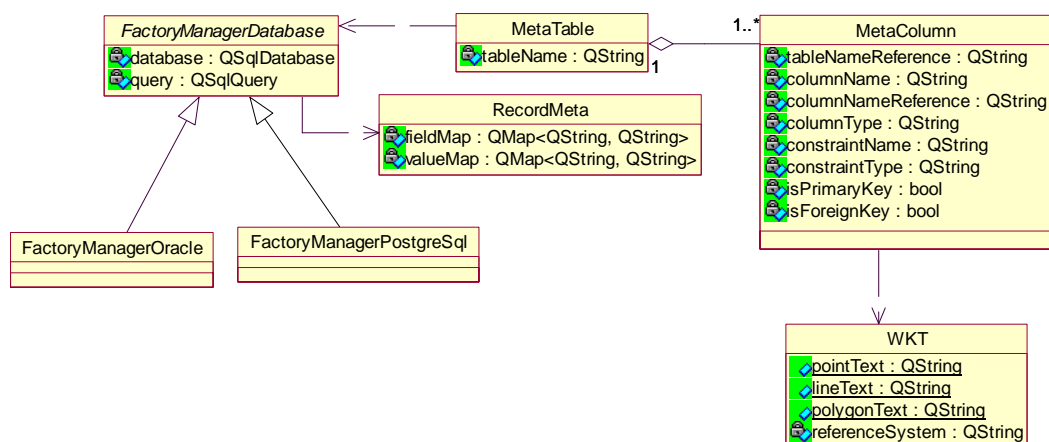


Figura 36 – Diagrama de classes de integração com o modelo de dados

Este diagrama de classes permite que o E-FOTO possa continuar seu processo de evolução sem depender do modelo de dados. Assim, mesmo que a EDGV continue evoluindo e, conseqüentemente, o modelo relacional da EDGV, derivado do modelo conceitual, também sofra atualizações, a EFD do E-FOTO não precisará de qualquer manutenção para enxergar estas modificações.

Um ponto importante a destacar nesta proposta é que ela permite que o E-FOTO se integre a outros modelos de dados diferentes da EDGV, desde que estes modelos respeitem os padrões estabelecidos pela OGC.

Há também as classes de visão, que permitem ao usuário interagir com o E-FOTO. Estas classes, a partir das interações com o usuário, se comunicam com as classes que realizam a classificação e o armazenamento no modelo de dados da EDGV. A figura 37 mostra um diagrama contendo classes de visão e de metamodelo.

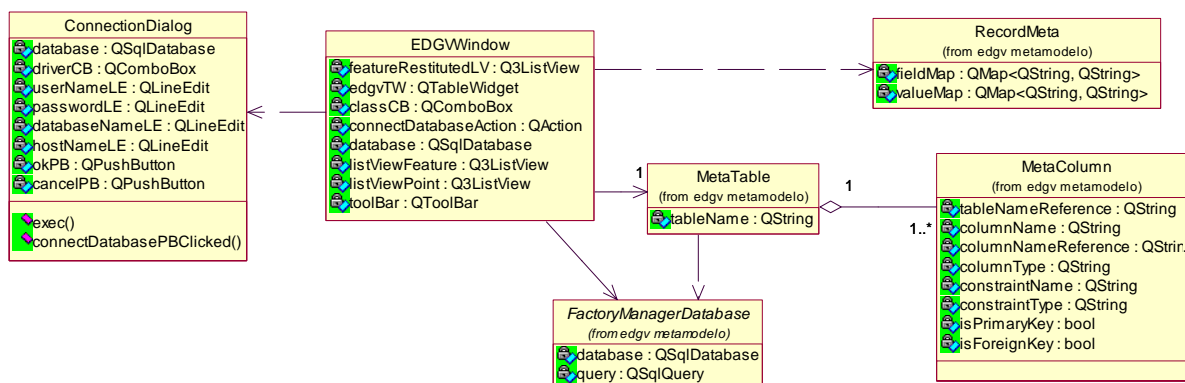


Figura 37 – Classes de visão e classes de metamodelo

4.2.3.1 O papel de cada classe

Cada classe exerce um importante papel para que a tarefa de restituição fotogramétrica digital na EDGV tenha sucesso. Os itens a seguir tratam de identificar o papel que cada classe desempenha.

- FactoryManagerDatabase

Esta classe abstrata tem como responsabilidade manter conexão, prover serviços para conversão de dados e armazenar os elementos restituídos no banco de dados. Ela define um conjunto de operações que suas subclasses precisam implementar de acordo com o SGBD escolhido para armazenar as feições. As informações necessárias para conhecimento de metadados das tabelas em um determinado SGBD são definidas nesta classe.

Uma observação sobre esta classe é que ela respeita o princípio do *Open Closed*, que afirma que uma classe deve estar aberta para extensões e fechada para modificações (MARTIN, 2006). A título de exemplo, suponha-se que o SGBD MySQL será responsável pelo armazenamento das feições utilizadas no processo de restituição para EDGV. Para tornar isto possível deve-se criar uma classe *FactoryManagerMySQL* herdando de *FactoryManagerDatabase* e implementando as operações abstratas. Há que se observar que não há alteração de qualquer outra classe, apenas a inclusão desta nova. A figura 38 exhibe os serviços oferecidos por esta classe. Os serviços em itálico são abstratos.

<i>FactoryManagerDatabase</i>	
✚	<i>namesFromTableWithGeometry()</i> : <i>QStringList</i>
✚	<i>valueAsSqlString(value : QString, type : QString)</i> : <i>QString</i>
✚	<i>selectSqlToMetaTableWithMetaColumns(tableName : QString)</i> : <i>QString</i>
✚	<i>valueAsDbString(value : QString, type : QString)</i> : <i>QString</i>
✚	<i>getNextVal(schemaName : QString, tableName : QString)</i> : <i>QString</i>
✚	<i>allNumericTypes()</i> : <i>QStringList</i>
✚	<i>allStringTypes()</i> : <i>QStringList</i>
✚	<i>allDateTypes()</i> : <i>QStringList</i>
✚	<i>allTimeStampTypes()</i> : <i>QStringList</i>
✚	<i>allTimeTypes()</i> : <i>QStringList</i>
✚	<i>allBooleanTypes()</i> : <i>QStringList</i>
✚	<i>allGeometryTypes()</i> : <i>QStringList</i>
✚	<i>getDatabase()</i> : <i>QSqlDatabase</i>
✚	<i>setDatabase(aQSqlDatabase : QSqlDatabase)</i> : <i>void</i>
✚	<i>getQuery()</i> : <i>QSqlQuery</i>
✚	<i>insertRecord(record : RecordMeta)</i> : <i>void</i>
✚	<i>valuesAsDbString(record : RecordMeta)</i> : <i>QString</i>
✚	<i>asDbDateString(aDateString : QString)</i> : <i>QString</i>
✚	<i>asDbTimeString(aTimeString : QString)</i> : <i>QString</i>
✚	<i>asDbTimeStampString(aTimestampString : QString)</i> : <i>QString</i>
✚	<i>asDbGeometryString(geometryString : QString, record : RecordMeta)</i> : <i>QString</i>
✚	<i>isGeometryType(aType : QString)</i> : <i>bool</i>
✚	<i>stringListAsCommaListString(stringList : QStringList)</i> : <i>QStringList</i>
🚩	<i>insertSqlString(record : RecordMeta)</i> : <i>QString</i>
🚩	<i>fieldNamesAsCommaListString(record : RecordMeta)</i> : <i>QString</i>
🚩	<i>fieldValuesAsCommaListString(record : RecordMeta)</i> : <i>QString</i>
🚩	<i>spatialReferenceSystemForTable(record : RecordMeta)</i> : <i>QString</i>
🚩	<i>stringBetweenSymbol(symbol : QString, aString : QString)</i> : <i>QString</i>
✚	<i>stringsEqual(str1 : QString, str2 : QString)</i> : <i>bool</i>

Figura 38 – Serviços oferecidos pela classe *FactoryManagerDatabase*

- *FactoryManagerOracle*

Para poder armazenar as feições no SGBD Oracle é necessário que esta classe implemente os serviços definidos pela sua superclasse *FactoryManagerDatabase*. Até o momento esta classe está inoperante. Está constando apenas para mostrar que o modelo está aberto para extensão sem a necessidade de alterações de código em qualquer outra classe.

- *FactoryManagerPostgreSql*

Esta classe herda de *FactoryManagerDatabase* e implementa os serviços necessários para manter a conexão, prover serviços para conversão de dados e armazenar os elementos restituídos no SGBD PostgreSQL. Ela permite ao E-FOTO trabalhar com o modelo de dados da EDGV.

Existe uma relação de dependência desta classe com o SGBD PostgreSQL, pois é necessário ter conhecimento de detalhes da estrutura deste SGBD para prover os serviços que necessitam ser desenvolvidos na classe.

- MetaTable

Esta classe tem como principal responsabilidade guardar informações de metadados de uma tabela. Ela utiliza a FactoryManagerDatabase para poder recolher estas informações de um determinado SGBD. Ela ainda mantém uma coleção de MetaColumn, que ajuda no armazenamento destes metadados. A figura 39 mostra esta classe.

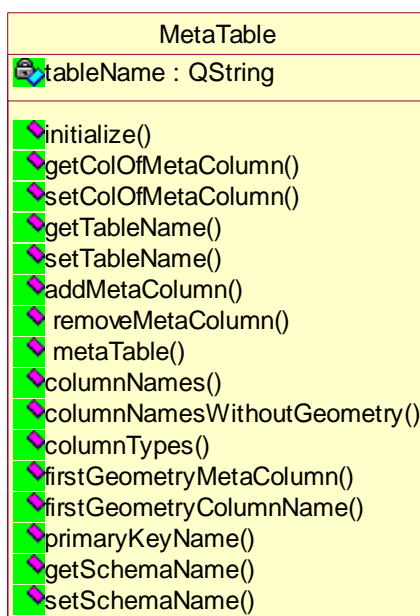


Figura 39 – Classe MetaTable

- MetaColumn

Esta classe tem a responsabilidade de manter todas as informações de uma coluna. Ela colabora com a classe MetaTable guardando as informações das colunas que pertencem a uma determinada tabela mantida pela classe MetaTable. Objetos desta classe são capazes de fornecer informações tais como: determinar se é uma chave primária; se é uma chave estrangeira e, neste caso, que tabela referencia; se a coluna é do tipo geométrica etc. A figura 40 ilustra esta classe.

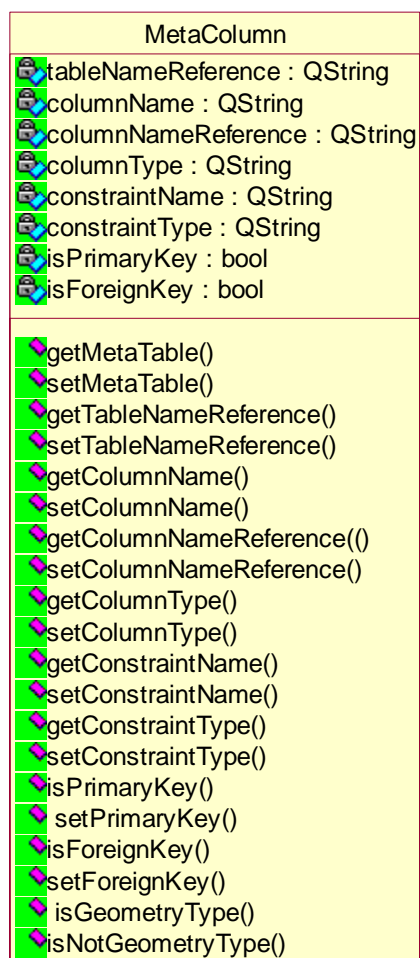


Figura 40 – Classe MetaColumn

- WKT

Esta classe é responsável pelos tipos de dados geométricos no formato estabelecido pela OGC. Ela é utilizada, por exemplo, pela classe `MetaColumn`. A figura 41 exibe sua estrutura.

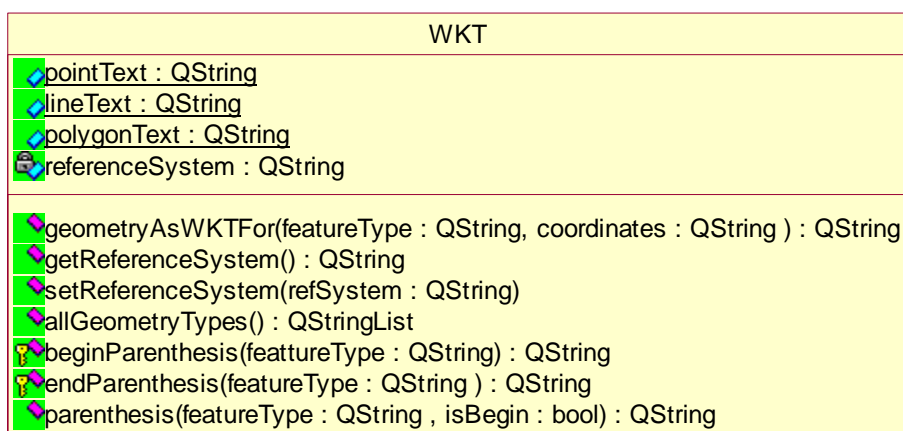


Figura 41 – Classe WKT

- EDGVWindow

Esta é a classe de visão que recebe as feições cartográficas que foram restituídas. Ela é a interface gráfica na qual o usuário interage para poder classificar os elementos geométricos de acordo com as feições da EDGV. Esta classe interage com as classes de metamodelo para realizar seu trabalho.

- ConnectionDialog

Esta é a classe de visão, que permite ao usuário se conectar com um banco de dados. Para isto, é necessário escolher o SGBD, usuário, senha, nome da base de dados e o local onde foi instalado o banco de dados. Ela colabora com a classe EDGVWindow.

4.3 Prova de Conceito

A partir da metodologia apresentada no capítulo 3, foi possível gerar um esquema de banco de dados no PostgreSQL com o cartucho espacial PostGIS. Este esquema gerado propiciou um modelo de dados compatível com a EDGV. O E-FOTO foi estendido para suportar a restituição fotogramétrica digital no padrão EDGV.

Para realizar esta prova, serão utilizadas as fotos 1997_016_300dpi.bmp e 1997_017_300dpi.bmp constantes do material didático disponível na internet (<http://www.efoto.eng.uerj.br/doku.php?id=pt-br:logo>). Também será necessário usar o arquivo 16_17_Uerj_oe.txt, que contém os parâmetros da orientação interior e exterior. Nesta avaliação será realizada a restituição de alguns elementos cartográficos nas imediações da UERJ, campus Maracanã, de acordo com as classes da EDGV. A figura 42 traz a tela principal do módulo de restituição 3D, com as imagens e parâmetros carregados no modo estereoscópico do tipo separação espacial.

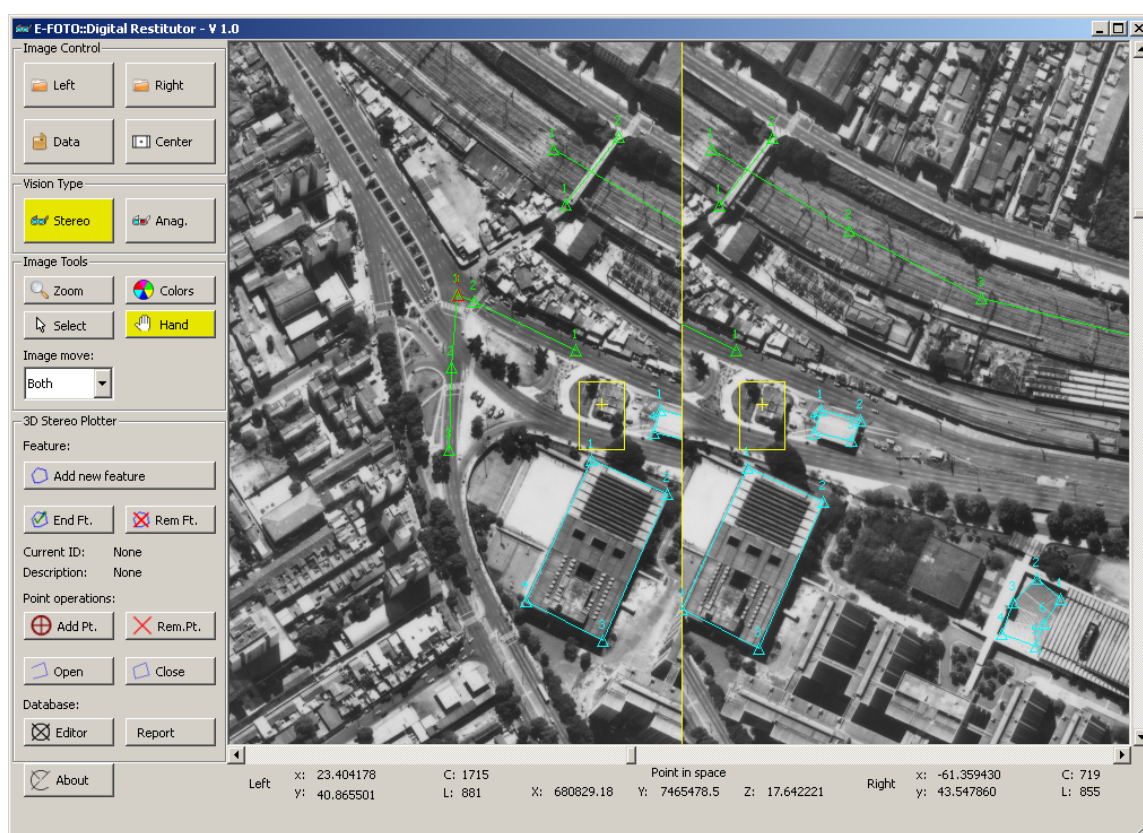


Figura 42 – Modelo Estereoscópico Carregado no E-FOTO

A figura 42 também exhibe alguns elementos cartográficos nas imediações da UERJ, já restituídos fotogrametricamente.

Em relação aos elementos restituídos, nesta avaliação, as seguintes feições foram selecionadas: (1) passarela da estação de trem da Mangueira; (2) trecho de ferrovia na Mangueira; (3) estação de metrô do Maracanã; (4) posto de combustível próximo à UERJ; (5) um trecho da Radial Oeste; (6) dois trechos da rua São Francisco Xavier; (7) um ponto rodoviário

entre a Radial Oeste e a São Francisco Xavier; (8) edificação de ensino de educação física da UERJ; (9) arquibancada do teatro da UERJ e; (10) uma estação de medição de fenômenos meteorológicos da UERJ. Os elementos de 1 a 7 pertencem a categoria Sistema de Transporte, os elementos 8 e 9 pertencem a categoria Educação e Cultura e o elemento 10 pertence à categoria Ponto de Referência. A tabela 8 traz a relação dos elementos a serem restituídos fotogrametricamente e suas respectivas classificações na EDGV.

Tabela 8 – Elementos cartográficos a serem restituídos e suas classificações na EDGV

Elemento Cartográfico	Classe	Geometria	Categoria
passarela da estação de trem da Mangueira	Travessia_Pedestre	Linha	Sistema de Transporte
trecho de ferrovia na Mangueira	Trecho_Ferrovuario	Linha	Sistema de Transporte
estação de metrô do Maracanã	Edif_Metro_Ferrovuaria	Polígono	Sistema de Transporte
posto de combustível próximo à UERJ	Posto_Combustivel	Polígono	Sistema de Transporte
um trecho da Radial Oeste	Trecho_Rodovuario	Linha	Sistema de Transporte
dois trechos da rua São Francisco Xavier	Trecho_Rodovuario	Linha	Sistema de Transporte
um ponto rodoviário entre a Radial Oeste e a São Francisco Xavier	Ponto_Rodovuario	Ponto	Sistema de Transporte
edificação de ensino de educação física da UERJ	Edif_Ensino	Polígono	Educação e Cultura
arquibancada do teatro da UERJ	Arquibancada	Polígono	Educação e Cultura
uma estação de medição de fenômenos meteorológicos da UERJ	Pto_Est_Med_Fenomenos	Ponto	Ponto de Referência

A figura 43 traz a relação destes elementos já restituídos fotogrametricamente, porém ainda como primitivas geométricas.

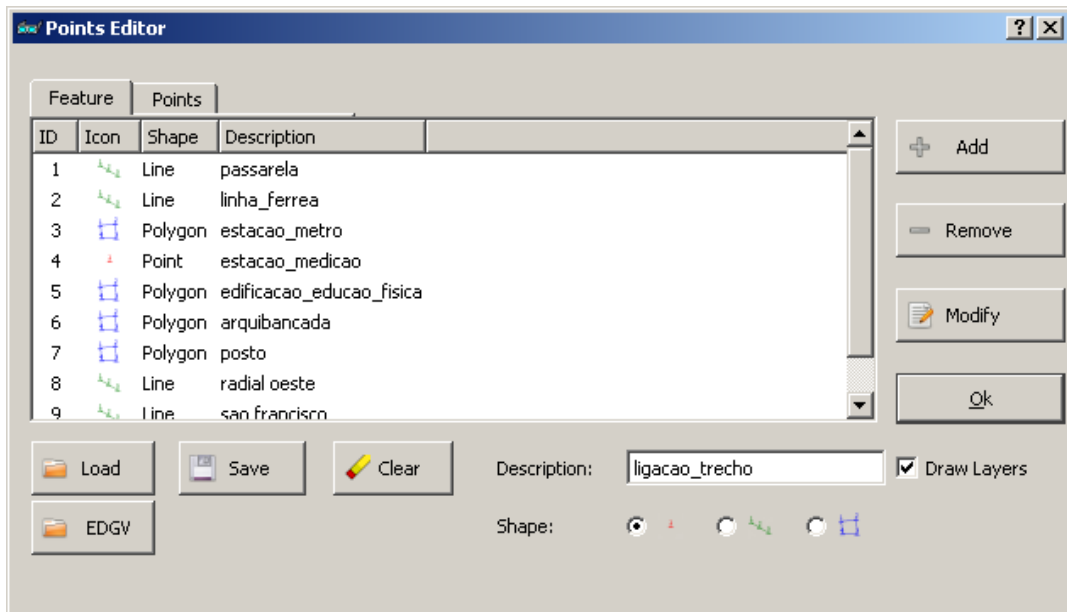


Figura 43 – Relação dos elementos cartográficos restituídos no E-FOTO

Uma vez que um elemento cartográfico tenha sido restituído, é necessário classificá-lo no padrão da EDGV. Para poder realizar esta classificação é necessário, antes, fazer a conexão com o banco de dados. A figura 44 ilustra esta situação.

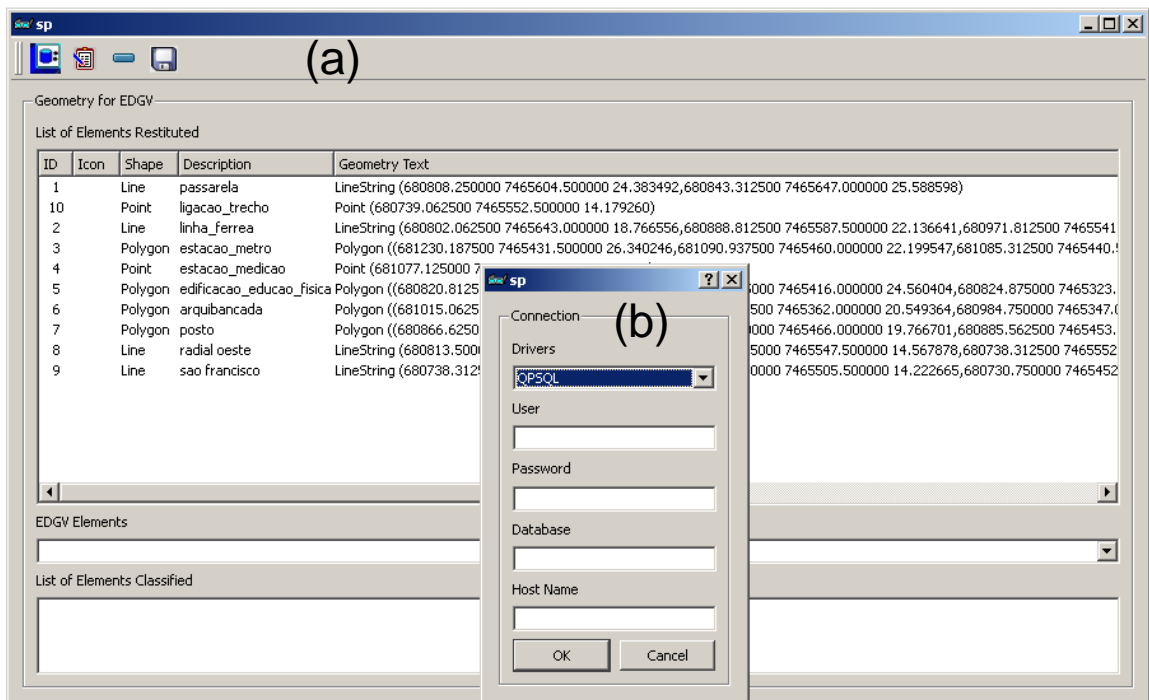


Figura 44 – Passos para classificação. (a) EDGVWindow e; (b) ConnectionDialog

Na figura 44 são listadas as feições a serem classificadas na EDGV. Para cada feição restituída é preciso classificá-la conforme a classe correspondente na EDGV e armazená-la no SGBD. Note-se também o formato geométrico de cada feição, com suas respectivas coordenadas de terreno.

Uma vez realizada a classificação e o armazenamento da feição no SGBD, o próximo passo é comprovar se as feições estão realmente armazenadas no banco de dados, no caso o PostgreSQL. Listam-se, a seguir algumas consultas com os respectivos resultados. Há que se observar que a coluna geom contém a geometria da feição no formato WKB.

(1) `select id_trecho_rodoviario, id_via_rodoviaria, nome, geom, asgml(geom) from edgv.trecho_rodoviario.` A figura 45 exibe o resultado.

id	id_via_rod	nome	geom	asgml
1	1	são francisco xavier	01020000A00572000003000000000000203AC	<gml:LineString srsName="EPSG:29189"><gml:coordinates>680733.0625,7465
2	2	radial oeste	01020000A005720000020000000000000003AC	<gml:LineString srsName="EPSG:29189"><gml:coordinates>680733,7465562,1
3	1	são francisco xavier	01020000A00572000002000000000000A03AC	<gml:LineString srsName="EPSG:29189"><gml:coordinates>680733.3125,7465:

Figura 45 – Resultado da consulta na tabela trecho_rodoviario

(2) `select id_ponto_rod_ferrovioario, nome, geom, asgml(geom) from edgv.ponto_rod_ferrovioario.` A figura 46 exibe o resultado.

id	nome	geom	asgml
1	ligacao_trecho	01010000A0057200000000002046C6244100000020947A5C4140F67AF7C75B2	<gml:Point srsName="EPSG:29189"><gml:coordinates>680739.062

Figura 46 – Resultado da consulta na tabela ponto_rod_ferrovioario

(3) `select id_travessia_pedestre, nome, geom, asgml(geom) from EDGV.travessia_pedestre.` A figura 47 exibe o resultado.

id	nome	geom	asgml
1	passarela mangueira	01020000A00572000002000000000000000000	<gml:LineString srsName="EPSG:29189"><gml:coordinates>680808.25,7465604.5,24.383492 6

Figura 47 – Resultado da consulta na tabela travessia_pedestre

(4) `select id_trecho_ferrovioario, nome, geom, asgml(geom) from edgv.trecho_ferrovioario.` A figura 48 exibe o resultado.

id	nome	geom	asgml
1	trecho_ferrovioario	01020000A005720000040000000000000020C4C62441000000C0AA7A5C41768D96033	<gml:LineString srsName="EPSG:29189"><gml:coordinates>

Figura 48 – Resultado da consulta na tabela trecho_ferrovioario

(5) select id_edif_metro_ferroviario, nome, geom, asgml(geom) from edgv.edif_metro_ferroviario. A figura 49 exibe o resultado.



id	nome	geom	asgml
1	estacao_metro	01030000A00572000001000000050000000	<gml:Polygon srsName="EPSG:29189"><gml:outerBoundaryIs><gml:LinearRing><gml:coordinates>6810

Figura 49 – Resultado da consulta na tabela edif_metro_ferroviario

(6) select id_posto_combustivel, nome, geom, asgml(geom) from edgv.posto_combustivel. A figura 50 exibe o resultado.



id	nome	geom	asgml
1	Posto UERJ	01030000A0057200000100000005000000000000000C045C7244100000040807A5C41D3A0681EC0BA3540000000C074C724	<gml:Polygon srsName="EF

Figura 50 – Resultado da consulta na tabela posto_combustivel

(7) select id_edif_ensino, nome, geom, asgml(geom) from edgv.edif_ensino. A figura 51 exibe o resultado.



id	nome	geom	asgml
1	edificacao_educacao_fisica	01030000A0057200000100000005000000000000040E8C6244100000020787A5C418E57207A52DA3640000	<gml:Polygon srsName="

Figura 51 – Resultado da consulta na tabela edif_ensino

(8) select id_arquibancada, nome, geom, asgml(geom) from edgv.arquibancada. A figura 52 exibe o resultado.



id	nome	geom	asgml
1	arquibancada_uerj	01030000A005720000010000000900000000000004053C82441000000805A7A5C41FE7E315BB23	<gml:Polygon srsName="EPSG:29189"><gml.out

Figura 52 – Resultado da consulta na tabela arquibancada

(9) select id_pto_est_med_fenomenos, nome, geom, asgml(geom) from edgv.pto_est_med_fenomenos. A figura 53 exibe o resultado.



id	nome	geom	asgml
1	estacao_medicao	01010000A005720000000000040E5C824410000000C0327A5C41	<gml:Point srsName="EPSG:29189"><gml:coordinates>681074.625,7465163,14.4

Figura 53 – Resultado da consulta na tabela pto_est_med_fenomenos

Uma vez comprovado o armazenamento no padrão da EDGV, a próxima etapa é confirmar a interoperabilidade dos dados através da visualização das feições guardadas no SGBD em outro software. A figura 54 mostra as feições restituídas no software QGIS.

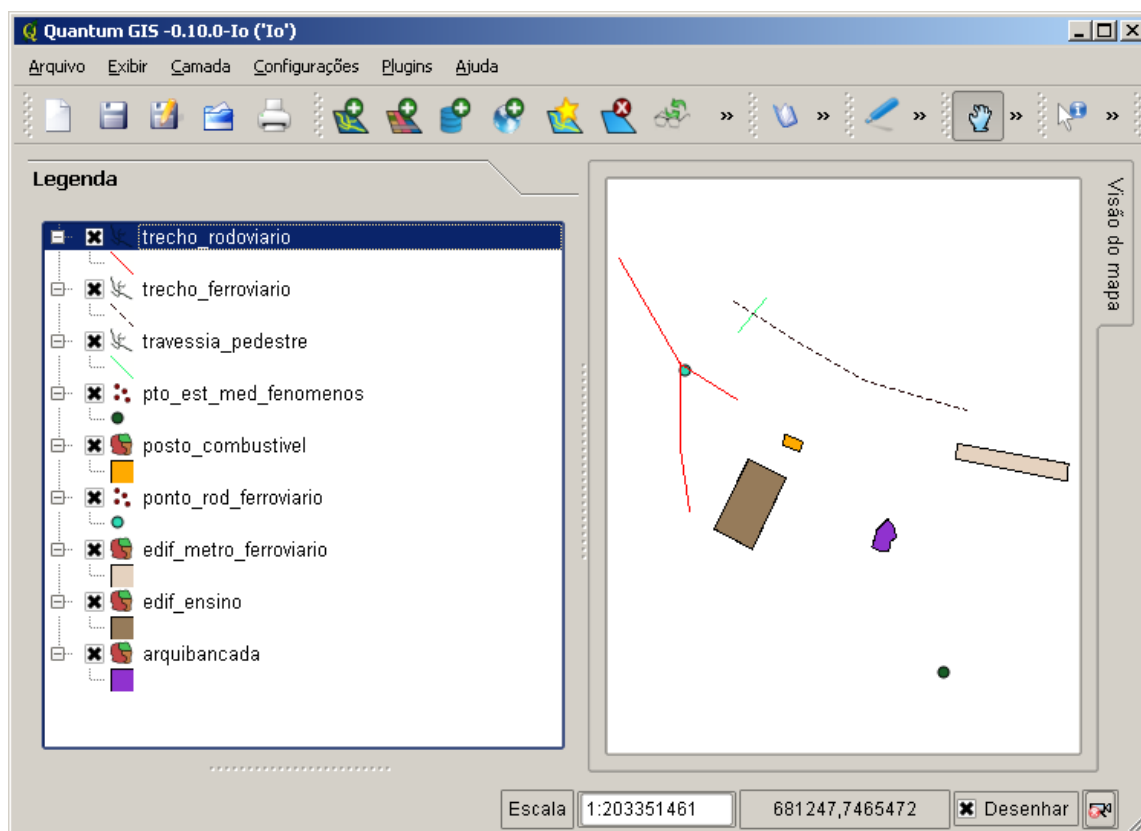


Figura 54 – Feições armazenadas no PostgreSQL e Visualizadas no QGIS

A figura 54 exibe, no software QGIS, as feições que foram restituídas e classificadas no padrão da EDGV no ambiente E-FOTO. Isto ratifica que foi possível realizar, com sucesso, a tarefa de restituição fotogramétrica digital para a EDGV em ambiente de software livre.

Uma vez armazenadas as feições no banco de dados, vários tipos de consulta, inclusive espaciais, podem ser feitas. A título de exemplo seguem algumas:

(a) saber quais travessias de pedestres, com suas respectivas extensões, cruzam trechos de ferrovia. A figura 55 traz o resultado da consulta.

```
SELECT EDGV.travessia_pedestre.nome, 'extensão: ' || ST_Length(EDGV.travessia_pedestre.geom),
EDGV.trecho_ferrovuario.nome
FROM EDGV.travessia_pedestre , EDGV.trecho_ferrovuario
WHERE ST_Crosses(EDGV.travessia_pedestre.geom, EDGV.trecho_ferrovuario.geom);
```

nome	?column?	nome_1
passarela mangueira	extensão: 55.0965416904727	trecho_ferrovuario

Figura 55 – Travessia de pedestres que cruzam trecho ferroviário

(b) saber qual a distância do posto de combustível até a edificação de ensino da UERJ. A figura 56 traz o resultado da consulta.

```
select st_distance(edgv.posto_combustivel.geom, edgv.edif_ensino.geom)
from edgv.posto_combustivel, edgv.edif_ensino
where (edgv.posto_combustivel.id_posto_combustivel = 1
and edgv.edif_ensino.id_edif_ensino = 1);
```

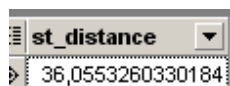


Figura 56 – Distância entre edificação e posto de combustível

(c) saber quais trechos rodoviários pertencem as vias rodoviárias. A Figura 57 traz o resultado da consulta.

```
SELECT trecho_rodoviario.id_trecho_rodoviario, via_rodoviaria.nome,
via_rodoviaria.nomeabrev AS nome_abrev, trecho_rodoviario.geom
FROM edgv.via_rodoviaria, edgv.trecho_rodoviario
WHERE (via_rodoviaria.id_via_rodoviaria = trecho_rodoviario.id_via_rodoviaria);
```

id_trecho	nome	nome_abrev	geom
3	Via São Francisco Xavier	São Francisco Xavier	01020000A00572000002000000000000A03AC62441000000C0967A5C41B6F2D93E1784E3840000000A09AC
1	Via São Francisco Xavier	São Francisco Xavier	01020000A00572000003000000000000203AC62441000000E0967A5C41B64DF1B6A8BE2640000000E038Cf
2	Via Radial Oeste	Radial Oeste	01020000A0057200000200000000000003AC6244100000080967A5C41E010A,AD4ECC1264000000080CDC

Figura 57 – Vias rodoviárias com seus respectivos trechos rodoviários

5 CONCLUSÃO

Não há dúvidas de que o projeto E-FOTO possui certa relevância para a comunidade acadêmica e científica por disponibilizar, de forma gratuita, um software voltado para a fotogrametria digital. Ele vem sendo desenvolvido visando, principalmente, ao ensino e à pesquisa. Com a introdução da EDGV neste processo, percebe-se, de uma maneira bem nítida, que um novo viés passa a existir. O projeto E-FOTO disponibilizou o primeiro software gratuito, aderente a uma Norma Cartográfica Nacional, cujo o uso, para as instituições que realizam produção voltada para o mapeamento topográfico sistemático terrestre básico, é obrigatório. Acredita-se, com isto, que novos caminhos foram criados e que podem gerar pesquisas e o uso do E-FOTO em ambiente de produção Cartográfica.

Este trabalho definiu como objetivo principal a possibilidade de realizar restituição fotogramétrica digital alinhada com as especificações técnicas para estruturação de dados geoespaciais vetoriais. Este objetivo foi alcançado através da extensão do módulo de restituição fotogramétrica digital do E-FOTO. Na perseguição desta meta, alguns objetivos específicos e intermediários também foram atingidos.

O primeiro objetivo específico atingido foi a criação de um esquema de um modelo de dados relacional aderente a EDGV para SGBDR/SGBDOR com extensão espacial. Este esquema foi utilizado, por exemplo, no SGBD PostgreSQL para armazenar as feições restituídas a partir do processo de restituição fotogramétrica digital. Ainda sobre este esquema é importante frisar que ele foi criado com base na EDGV 2.1, última versão homologada pela CONCAR em dezembro de 2008.

Um outro objetivo específico obtido foi, através dos elementos restituídos e armazenados no SGBD, o intercâmbio de dados. A prova de conceito realizada no capítulo 4, exemplificou este intercâmbio através da visualização no uso do software QGIS, dos elementos restituídos no E-FOTO e armazenados no SGBD PostgreSQL, com o cartucho espacial PostGIS.

Há que se ressaltar que a EDGV também está em evolução. Esta Norma Cartográfica, a princípio, atende às instituições que fazem mapeamento sistemático terrestre básico. A EDGV deverá incorporar nos próximos anos mapeamento cadastral e até temático. Isto possibilitará um alcance muito grande da EDGV no cenário Nacional. O E-FOTO, da maneira com está implementado, possibilitará a Restituição Fotogramétrica Digital também para estes mapeamentos,

sem alteração no seu código, desde que o esquema de banco de dados esteja disponível e em conformidade com os padrões da OGC.

Uma importante informação a destacar é que outros esquemas de banco de dados voltados para a parte geográfica, diferentes da EDGV, poderão ser utilizados sem a necessidade de ajustes. Por exemplo, se alguma Instituição tem uma base de dados geoespaciais no PostgreSQL, esta poderá ser utilizada sem a necessidade de qualquer alteração no E-FOTO, desde que respeite os padrões definidos pela OGC. Caso esta Instituição deseje usar um outro SGBD, desde que este siga os padrões da OGC, bastará desenvolver uma nova subclasse de `FactoryManagerDatabase` e implementar as operações abstratas para fazer uso no E-FOTO. Para isto se faz necessário recompilar o código fonte do software.

Dentre os resultados obtidos, encontra-se um conjunto de *scripts* SQL para a criação de um esquema em SGBDR ou SGBDOR que está em conformidade com a EDGV e um Software que foi estendido para permitir realizar a Restituição Fotogramétrica Digital de acordo com a EDGV.

Como sugestões para trabalhos futuros mencionam-se a inclusão de novas funcionalidades no E-FOTO, dentre as quais destacam-se:

- Realizar a validação topológica e a consistência lógica dos elementos da EDGV;
- Possibilitar ao E-FOTO carregar imagens acima de 100 Mb;
- Permitir ao E-FOTO carregar os elementos restituídos da EDGV através de um SGBDE.

REFERÊNCIAS

AMBLER, S. W., Disponível em:

<<http://www.agiledata.org/essays/mappingObjects.html#BasicConcepts>>. Acesso em: out. 2008.

BOOCH, G., RUMBAUGH, J., JACOBSON, I., *UML guia do usuário*. Rio de Janeiro, RJ, Editora Campus, 2000.

BORGES, K. A. V. OMT-G: An Object-Oriented Data Model for Geographic Applications. Holanda: *Jornal GeoInformatica*, 2001.

BRITO, J. L. N. et. al. Projeto E-FOTO: Disponibilizando uma Ferramenta educacional livre para a Fotogrametria Digital COBRAC 2006 · Congresso Brasileiro de Cadastro Técnico Multifinalitário · UFSC Florianópolis · 15 a 19 de Outubro 2006.

CASANOVA, M. et. al. Banco de Dados Geográficos. Curitiba: MundoGEO, 2005.

COELHO, L. e BRITO, J. L. N. *Fotogrametria Digital*. Rio de Janeiro, UERJ, 2007.

CONCAR (Brasil) - Especificações Técnicas para Estruturação de Dados Geoespaciais Digitais Vetoriais. Brasília, 2007.

CONCAR (Brasil) - RESOLUCAO_CONCAR_001-2007. Brasília, 2007

BRASIL. Decreto nº 6.666 de 27 de novembro de 2008. Institui a Infra Estrutura Nacional. Diário Oficial [da] República Federativa do Brasil, Poder Executivo, Brasília, DF, 28 nov. 2008. Seção 1, p.57.

DELOU, Alberto Luiz de Azevedo. Estruturação de Bases Cartográficas Para SIG: Um estudo comparativo dos modelos do IBGE e da DSG. 2006. 291 f. Dissertação (Mestrado em Engenharia de Computação) - Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2006.

DSG - TBCD(Tabelas da Base Cartográfica Digital). 2 ed. Brasília: Exército Brasileiro, 1998.

ELMASRI, R., NAVATHE, S. B., *Fundamentals of Database Systems*. Fourth Edition. Addison-Wesley. 2003, 1030p

FOWLER, M. Analysis Patterns Reusable Object Models. USA, Addison-Welsey, 2002.

FOWLER, M. Uml Distilled, 3Rd Ed. USA, Addison Wesley, 2003

FOWLER, M. et. al. Patterns of Enterprise Application Architecture . USA, Addison-Welsey, 2002.

HEIPKE, C., Digital Photogrammetric Workstations – a review of the state of the art for Topographic applications, Gim Internation (15) 4, 2001, p.35 -37.

IBGE. Noções Básicas de Cartografia: Manuais Técnicos em Geociências. Rio de Janeiro, 1999. P

INPE – Curso de Banco de Dados Geográficos - Cursos de Pós-Graduação Divisão de Processamento de Imagens – DPI. São Paulo.

INSTITUTO GEOGRÁFICO NACIONAL DA ESPANHA - Elementos de una IDE – Dezembro de 2008

LUNARDI, O. A. & AUGUSTO M. J. C. Infra-Estrutura dos Dados Espaciais Brasileira – Mapoteca Nacional Digital. COBRAC 2006 · Congresso Brasileiro de Cadastro Técnico Multifinalitário · UFSC Florianópolis · 15 a 19 de Outubro 2006.

MAPOTECA TOPOGRÁFICA DIGITAL: Documento Geral – Versão 3.0. IBGE, Diretoria de Geociências, Departamento de Cartografia. Rio de Janeiro, 1996.

MARTIN, C. R., MARTIN, M., Agile Principles, Patterns, and Practices in C#. Prentice Hall, 2006.

ODBMS. Disponível em: <<http://www.odbms.org/introduction.html>>. Acesso em: 18 jan. 2009.

ODBMS. Disponível em: <http://www.odbms.org/introduction_definition.html>. Acesso em: 18 jan. 2009.

OPEN GEOSPATIAL CONSORTIUM (OGC). Disponível em: <<http://www.opengeospatial.org>>. Acesso em: 11 jan. 2009.

OPENGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture. OGC, Outubro de 2006.

OPENGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL Option. OGC, Outubro de 2006.

POSTGIS 1.3.5 Manual. Disponível em: < <http://postgis.refrations.net/documentation>>. Acesso em: 5 jan. 2009.

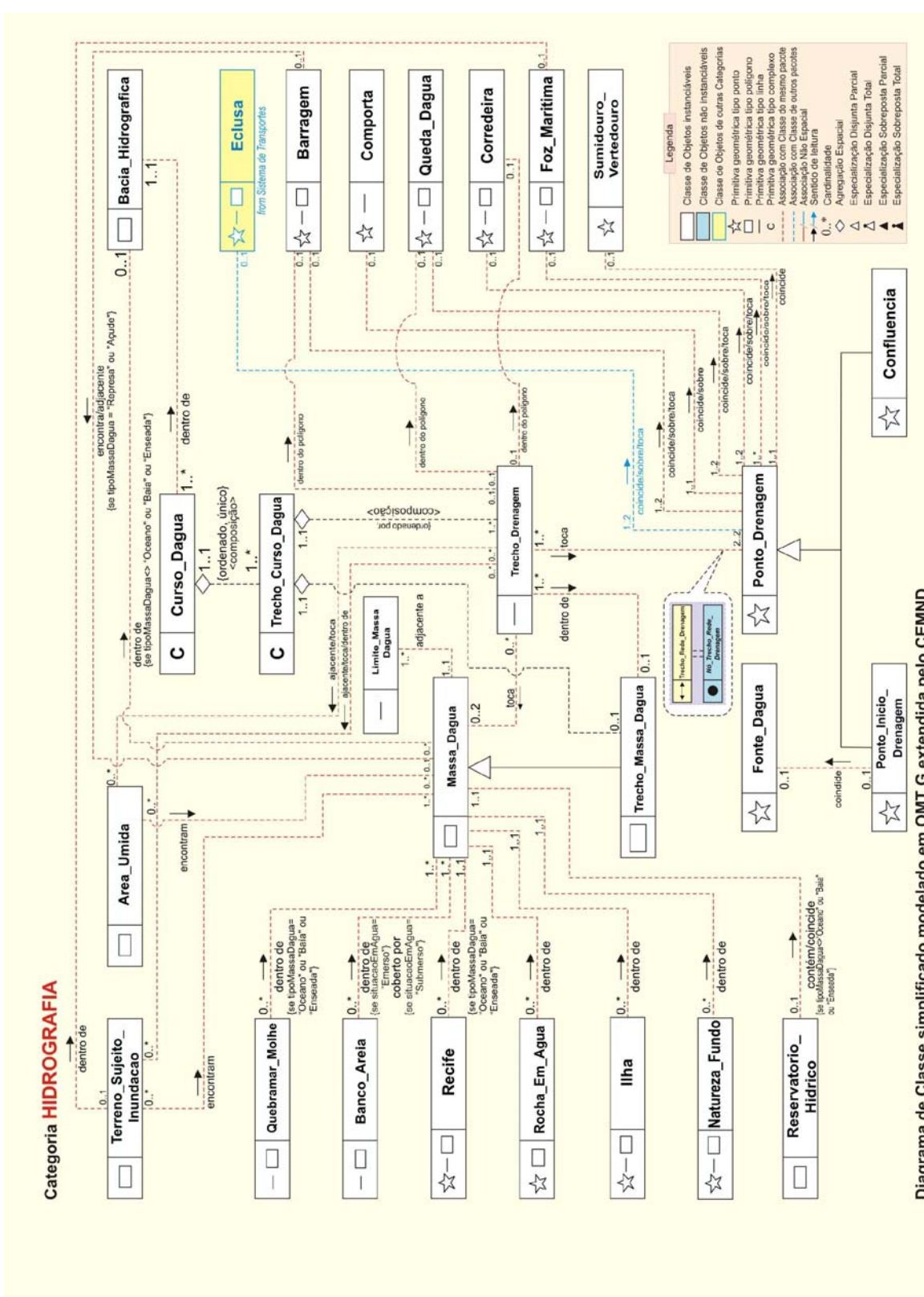
POSTGRESQL. Disponível em: < <http://www.postgresql.org/>>. Acesso em: 20 dez. 2008.

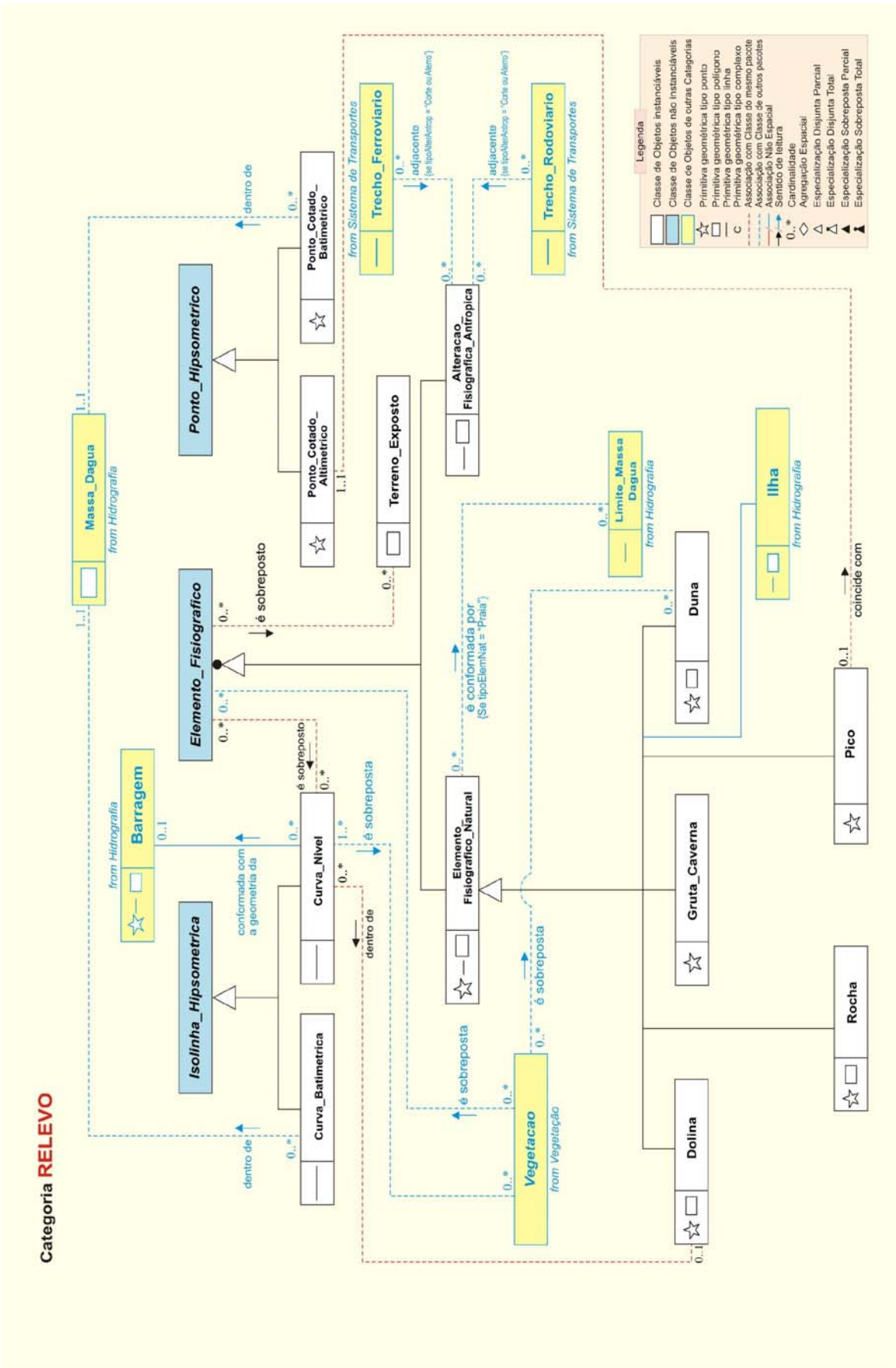
SILBERSCHARTZ, A., HENRY, F. K., SUDARSHAN, S., Database System Concepts Fourth Edition. McGRAW-HILL. 2001

POSTGRESQL, G. D. G., PostgreSQL 8.3.4 Documentation. 2008

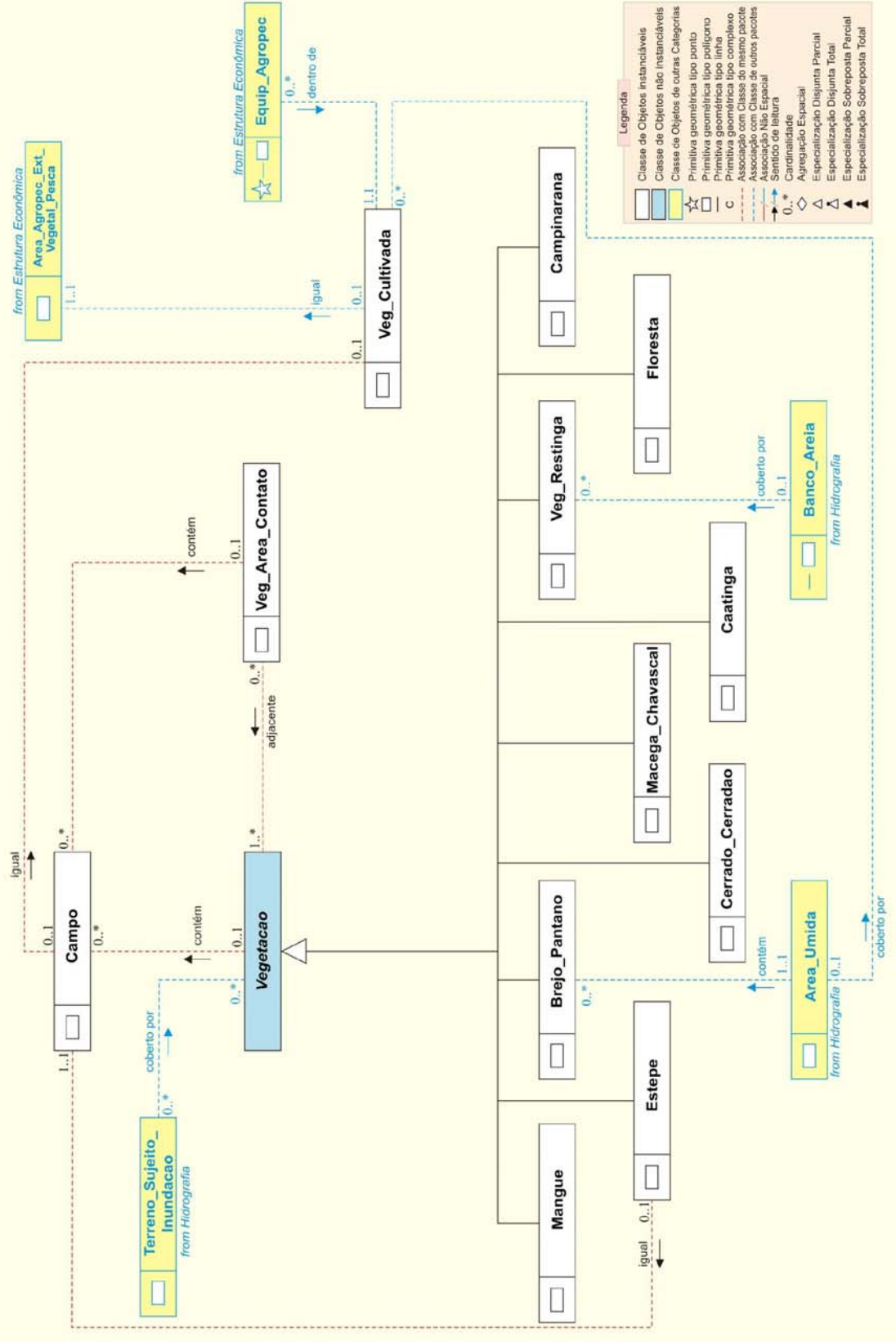
TANAKA, A., Banco de dados Espaciais. UNIRIO. 2007

APÊNDICE - Modelo conceitual da EDGV simplificado

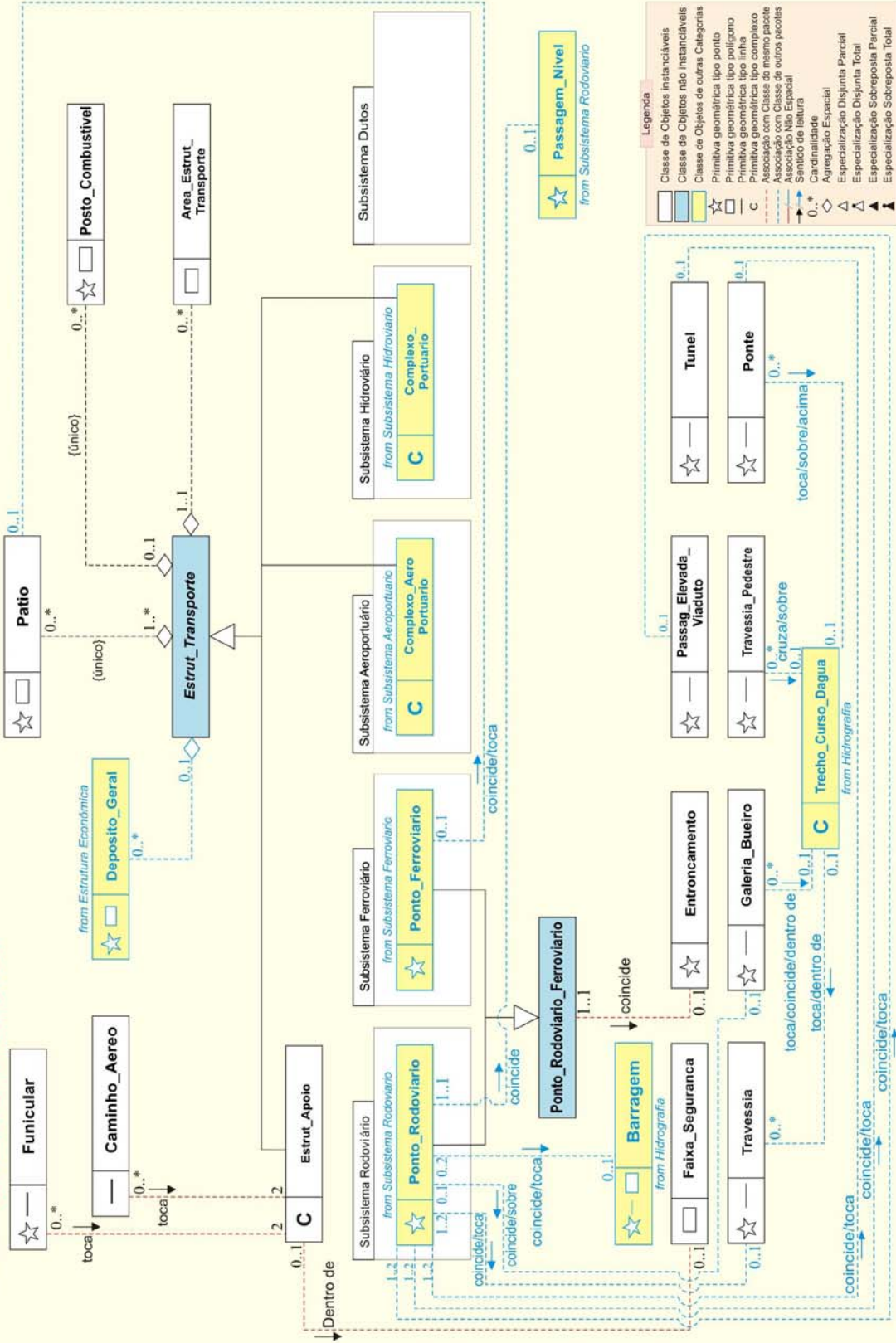




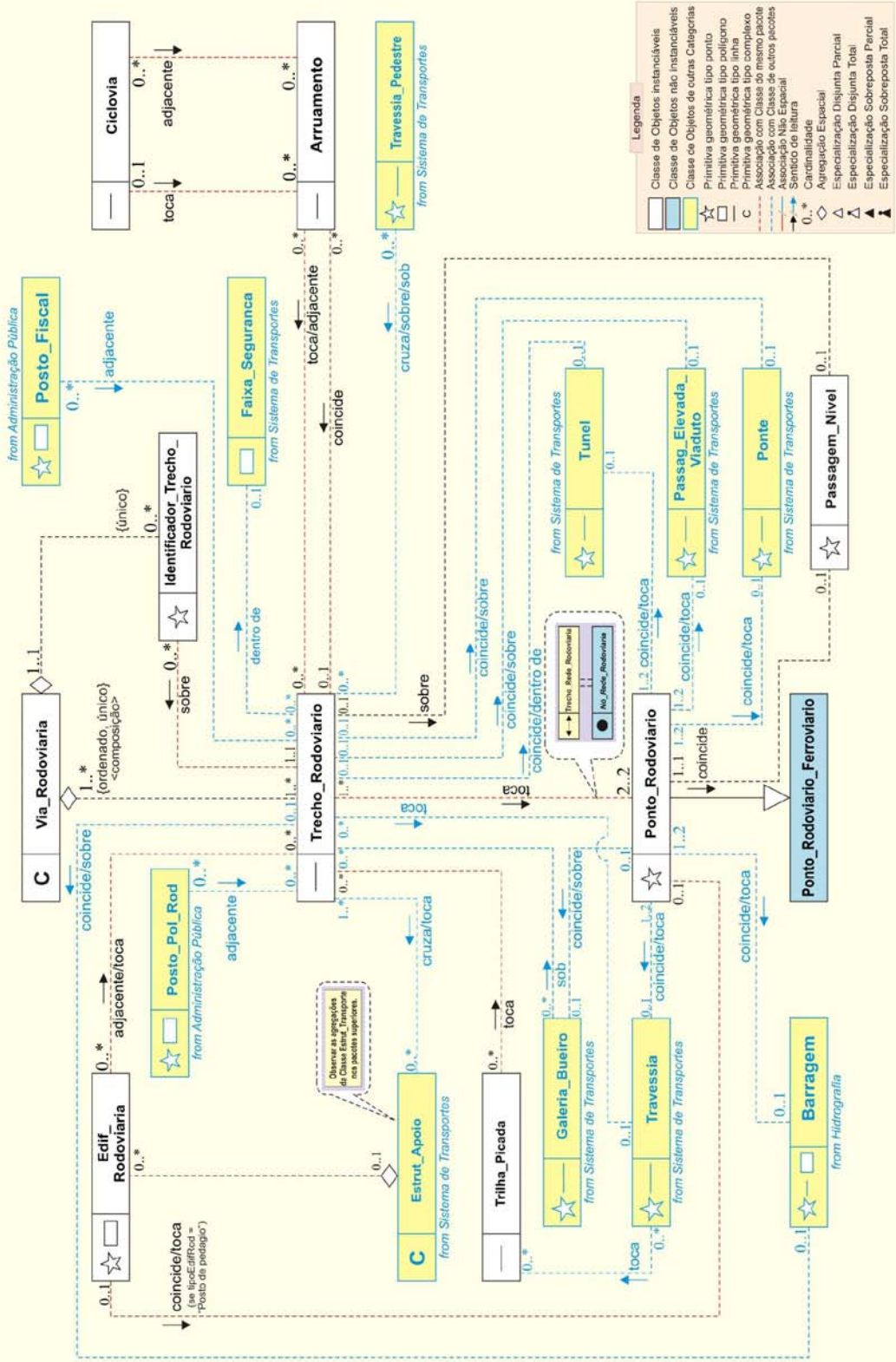
Categoria VEGETAÇÃO



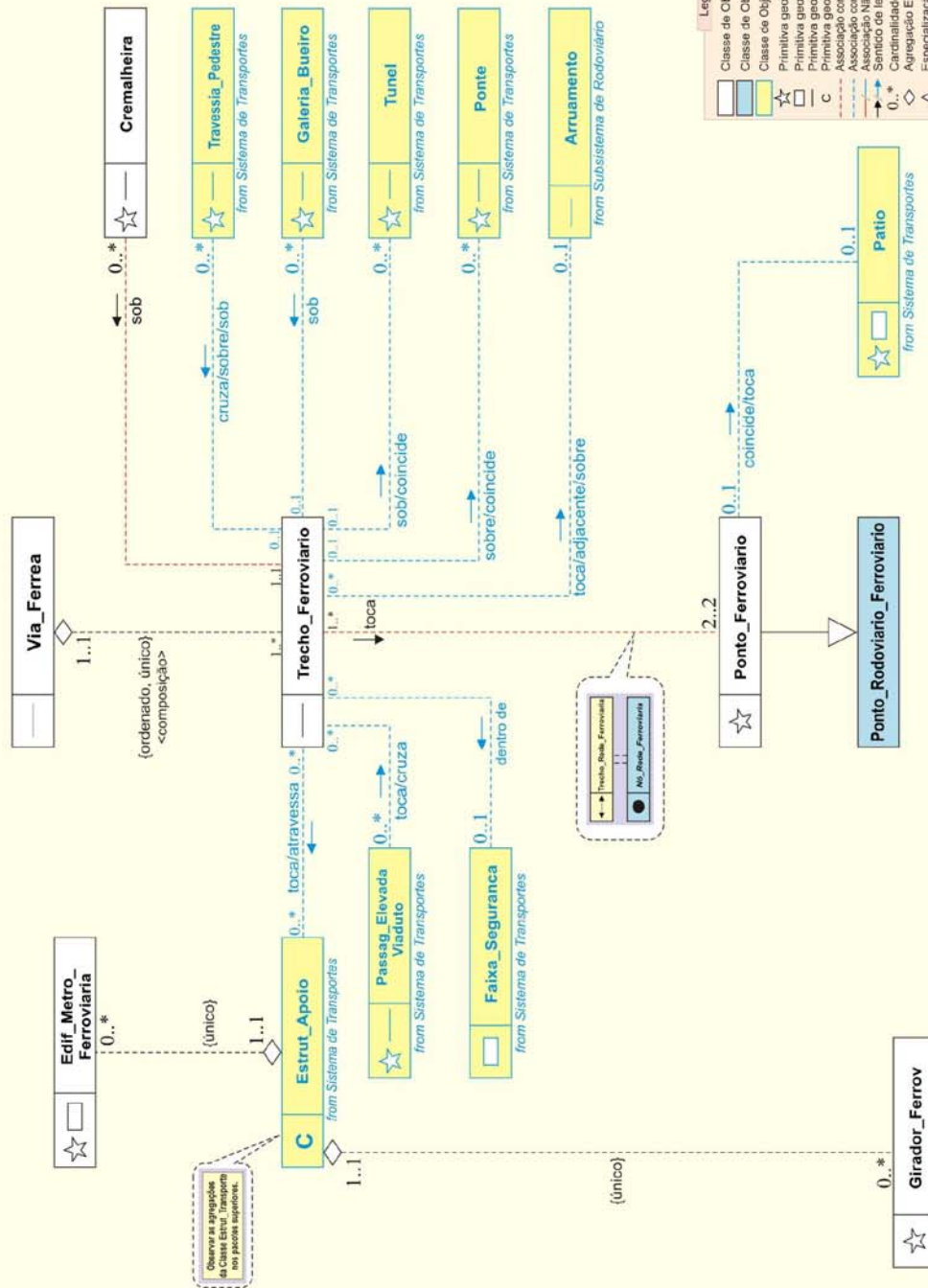
Categoria SISTEMA DE TRANSPORTES



Categoria SISTEMA DE TRANSPORTES - Subsistema Rodoviário



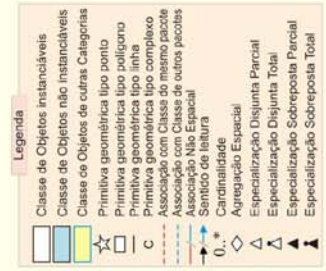
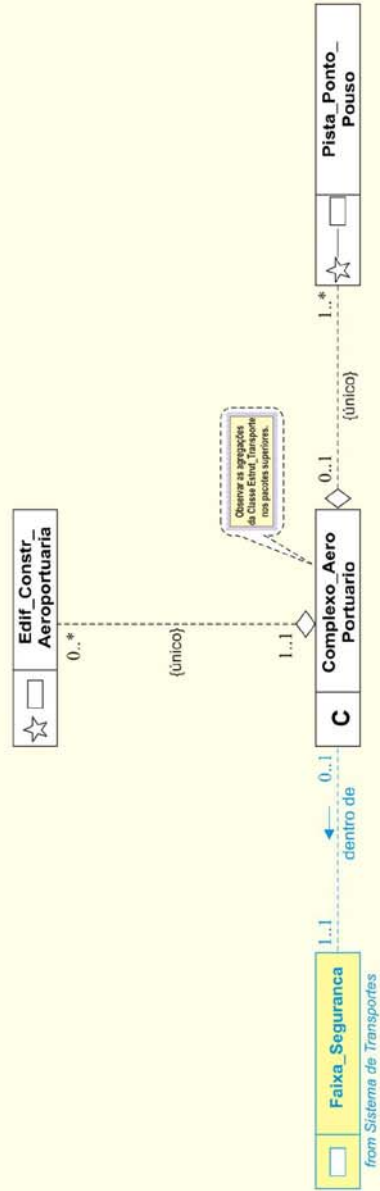
Categoria SISTEMA DE TRANSPORTES - Subsistema Ferroviário



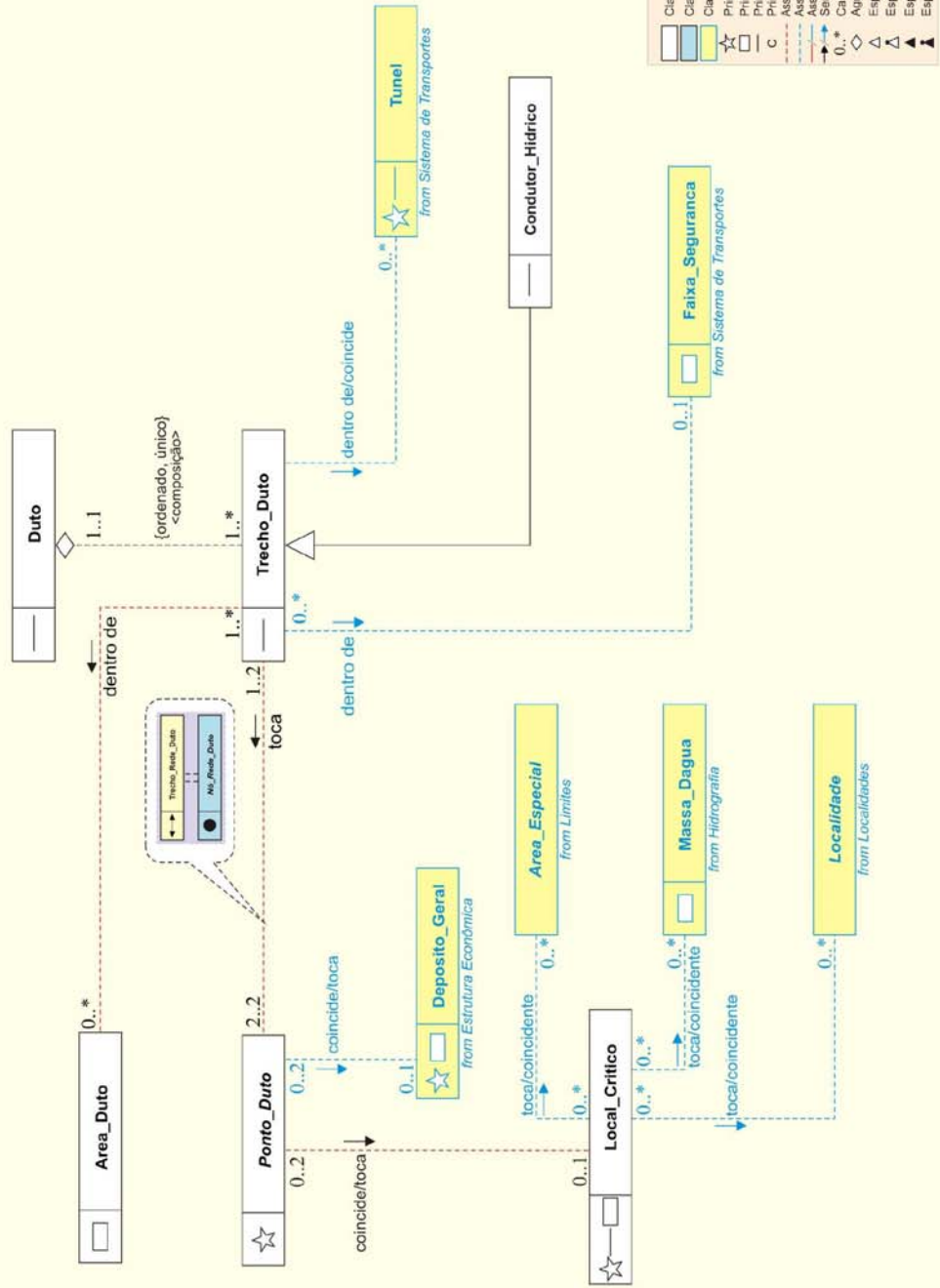
Legenda

- Classe de Objetos instanciáveis
- Classe de Objetos não instanciáveis
- Classe de Objetos de outras Categorias
- Primitiva geométrica tipo ponto
- Primitiva geométrica tipo polígono
- Primitiva geométrica tipo linha
- Primitiva geométrica tipo complexo
- Associação com Classe do mesmo pacote
- Associação com Classe de outros pacotes
- Associação Especial
- Sentido Inversa
- Cardinalidade
- Agregação Especial
- Especialização Disjunta Parcial
- Especialização Disjunta Total
- Especialização Sobreposta Parcial
- Especialização Sobreposta Total

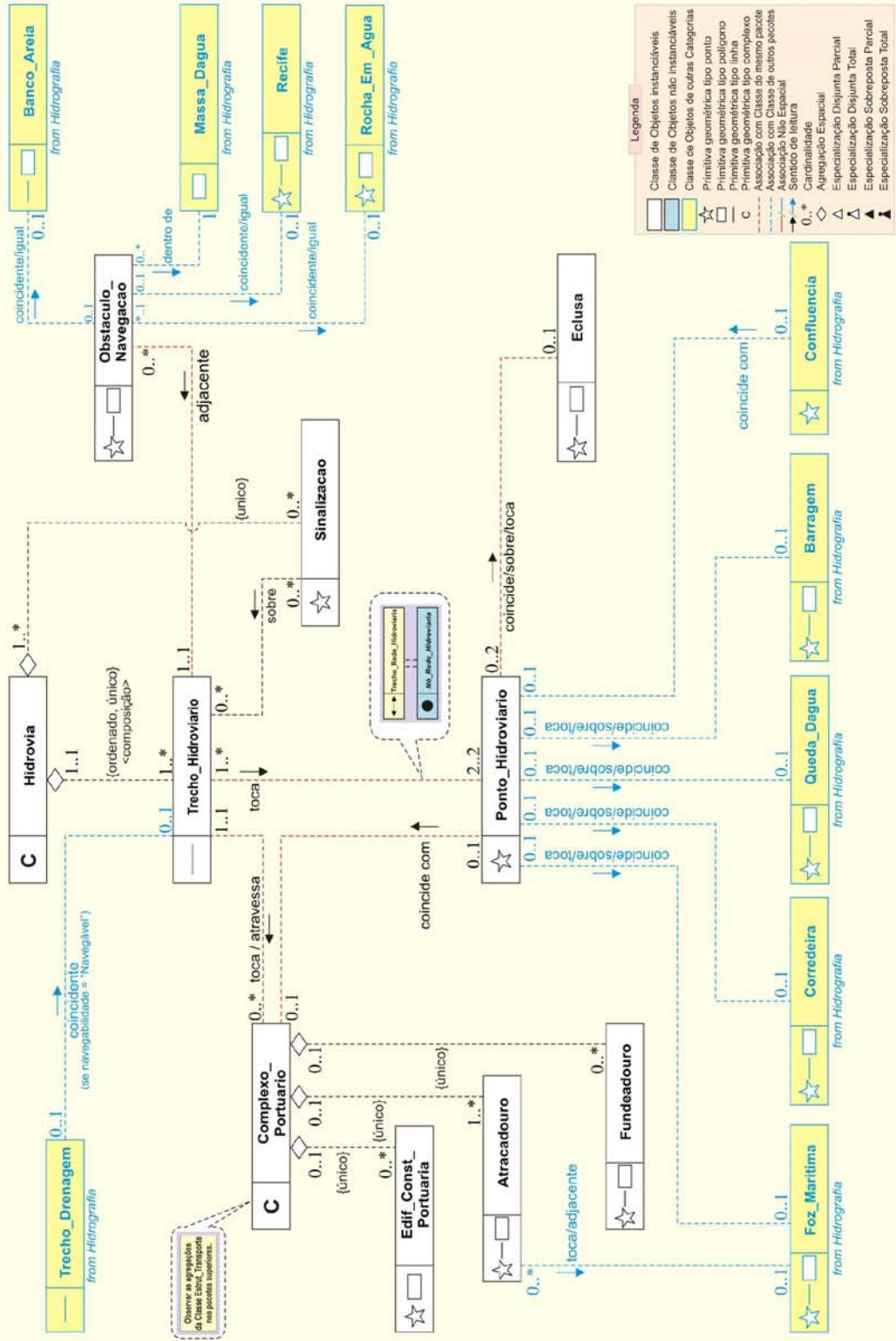
Categoria SISTEMA DE TRANSPORTES - Subsistema Aeroportuário



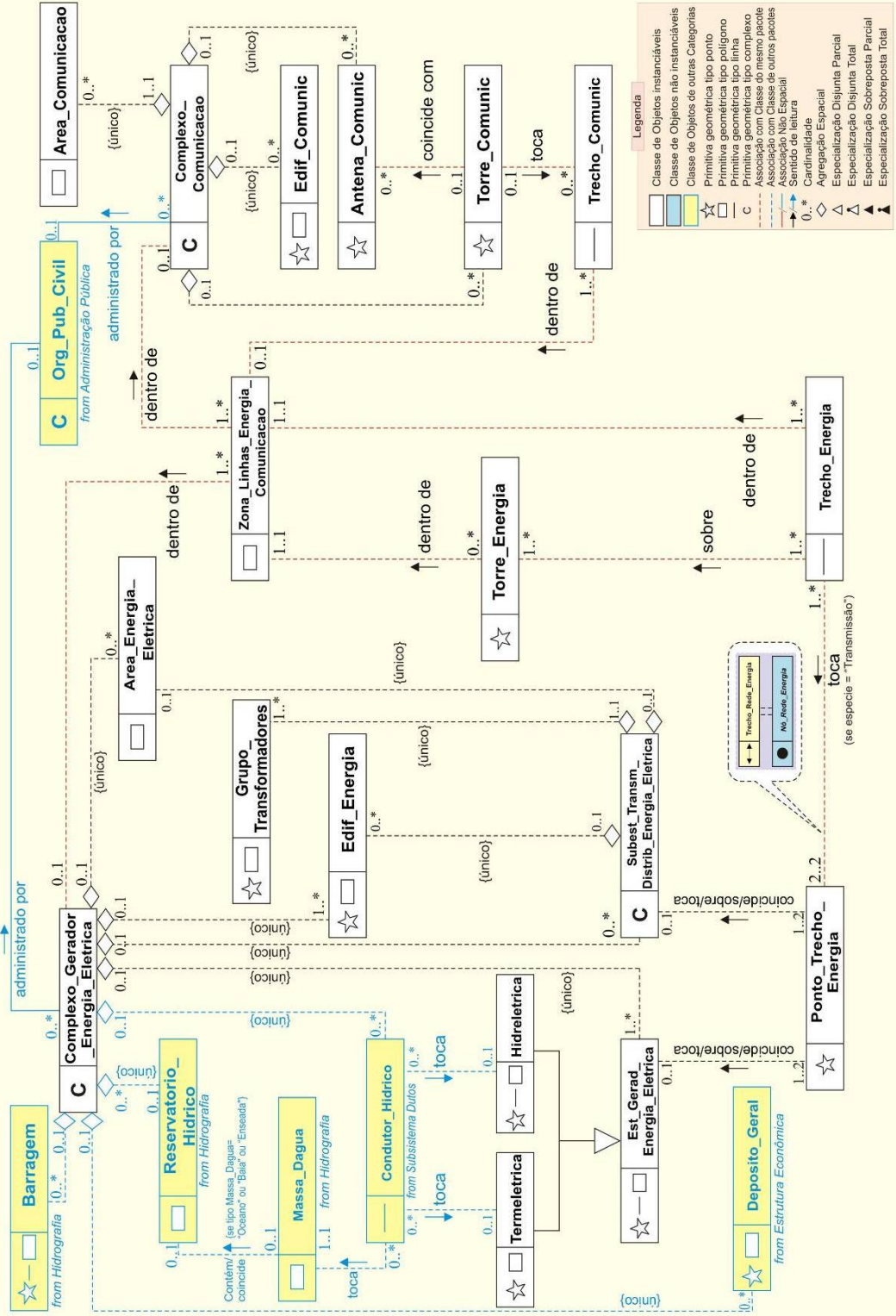
Categoria SISTEMA DE TRANSPORTES - Subsistema de Dutos



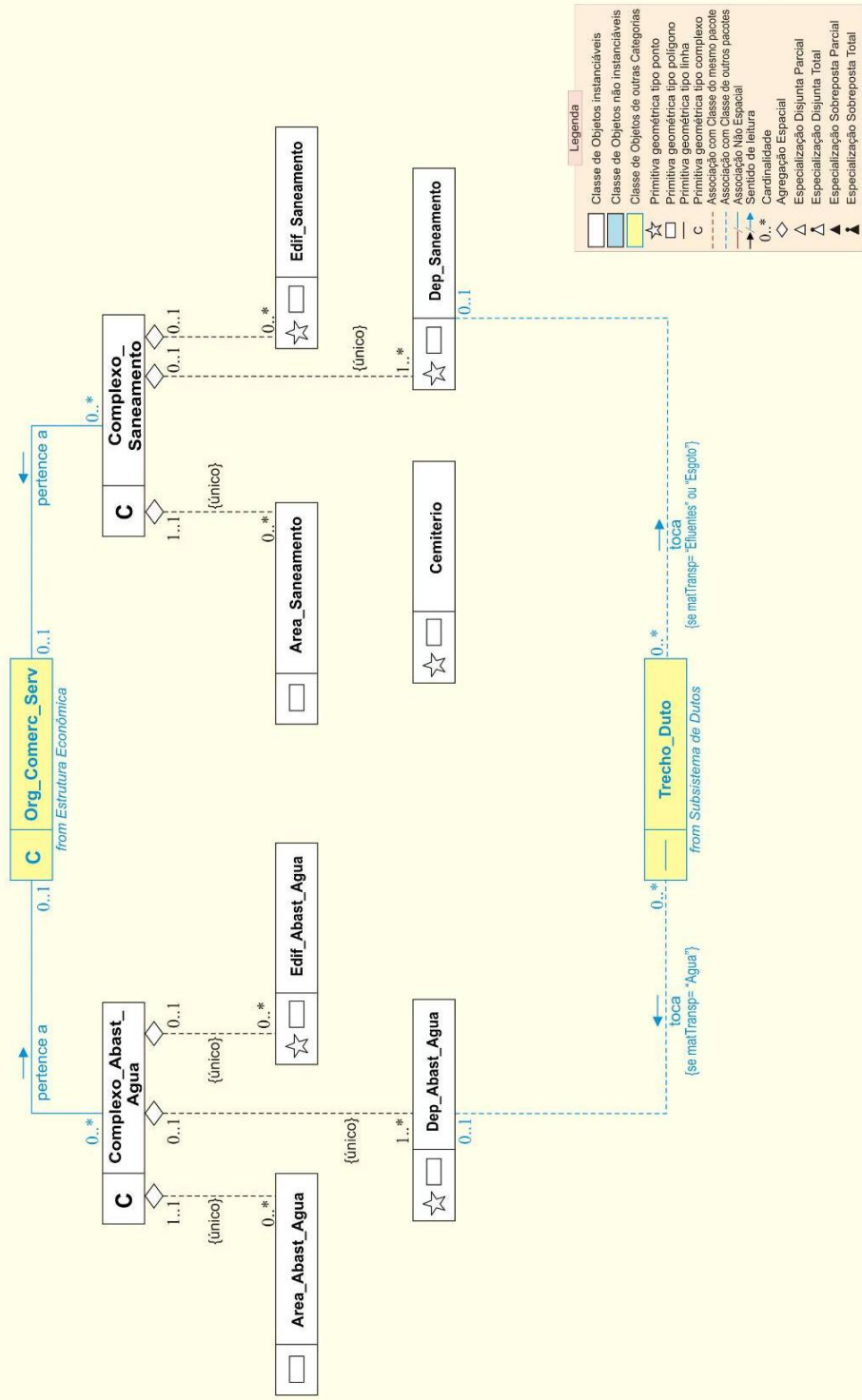
Categoria SISTEMA DE TRANSPORTES - Subsistema Hidroviário

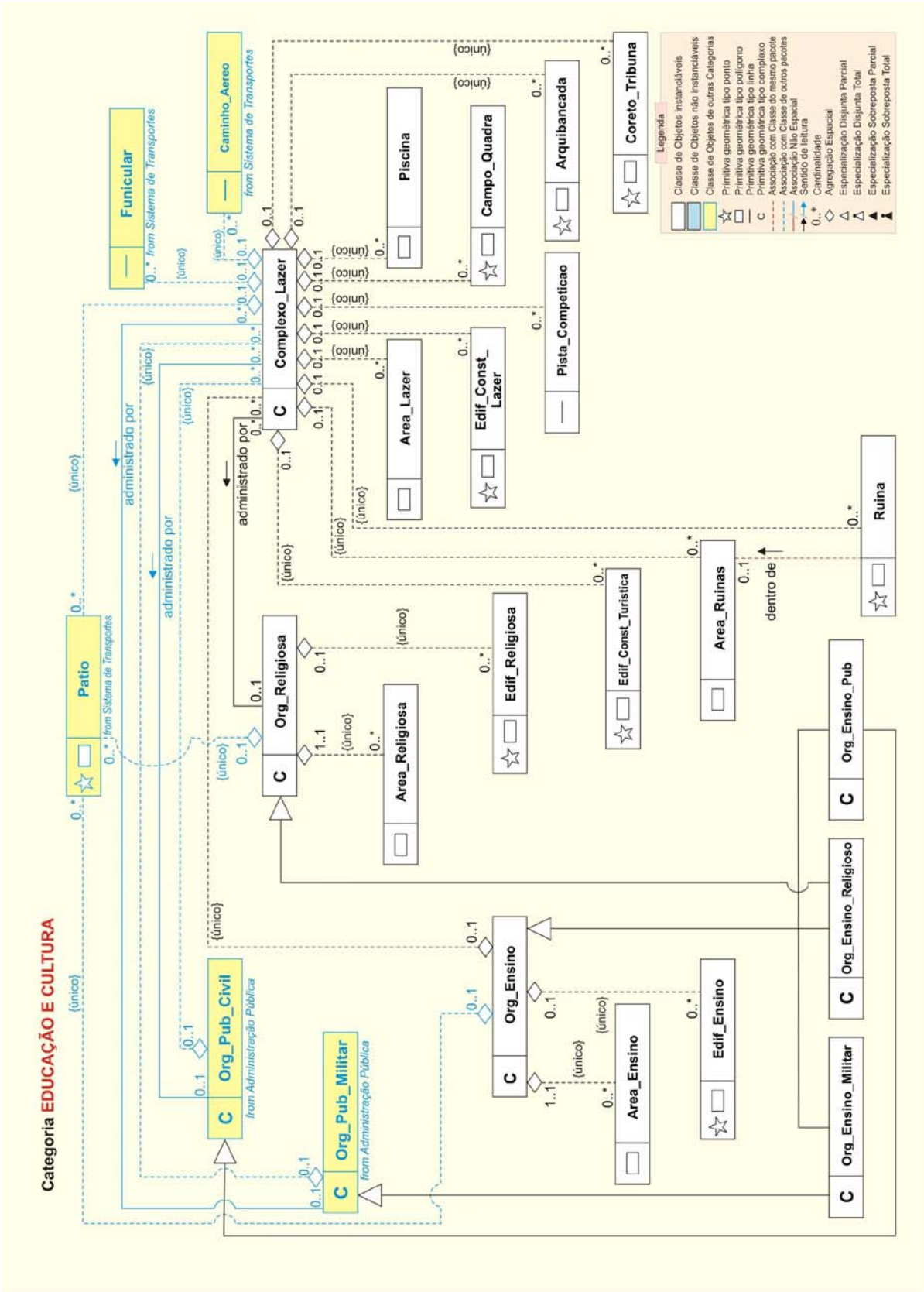


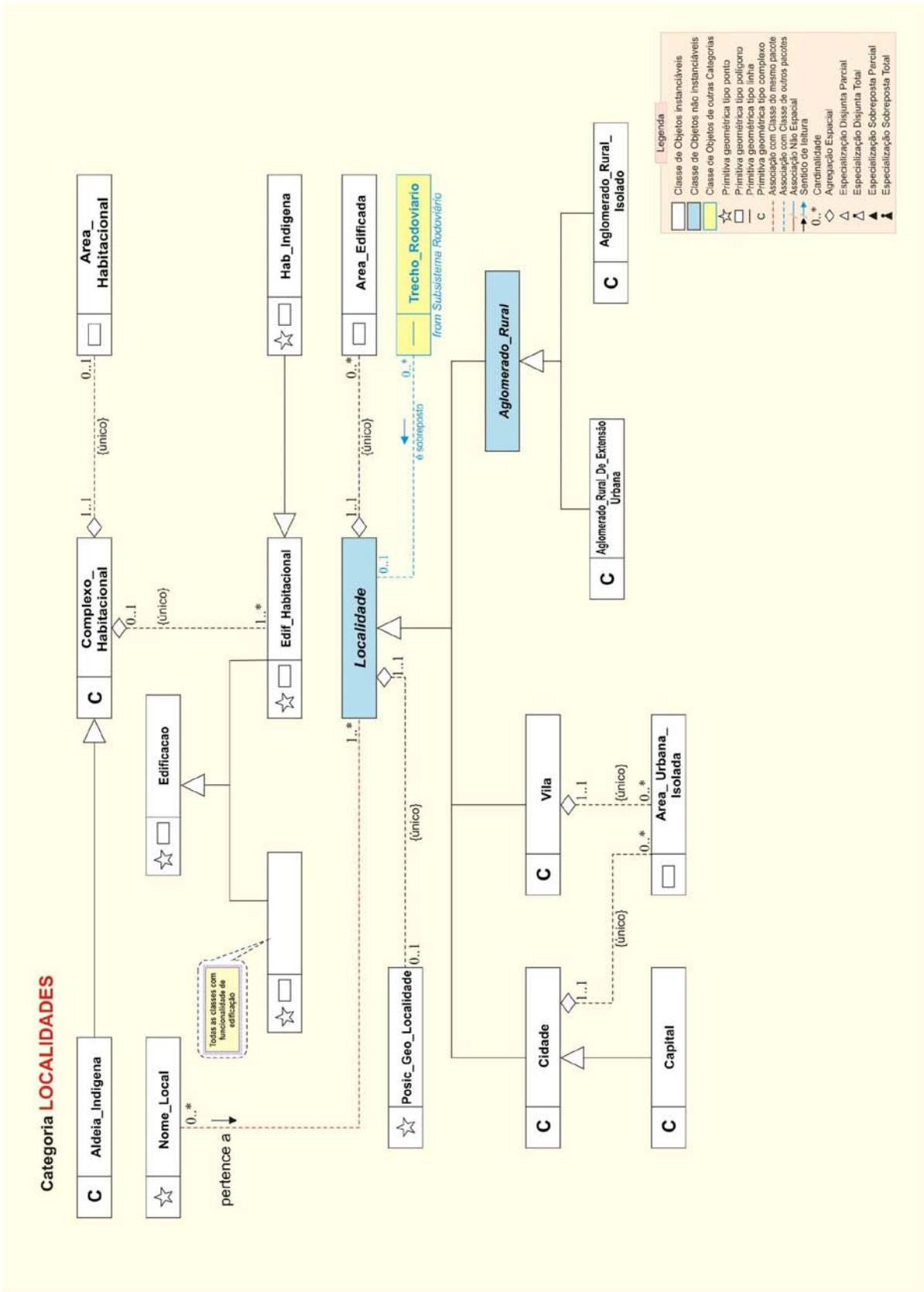
Categoria ENERGIA E COMUNICAÇÕES



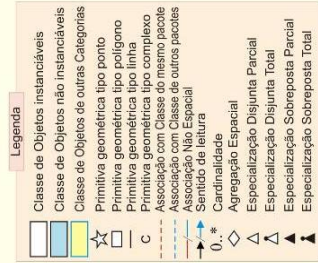
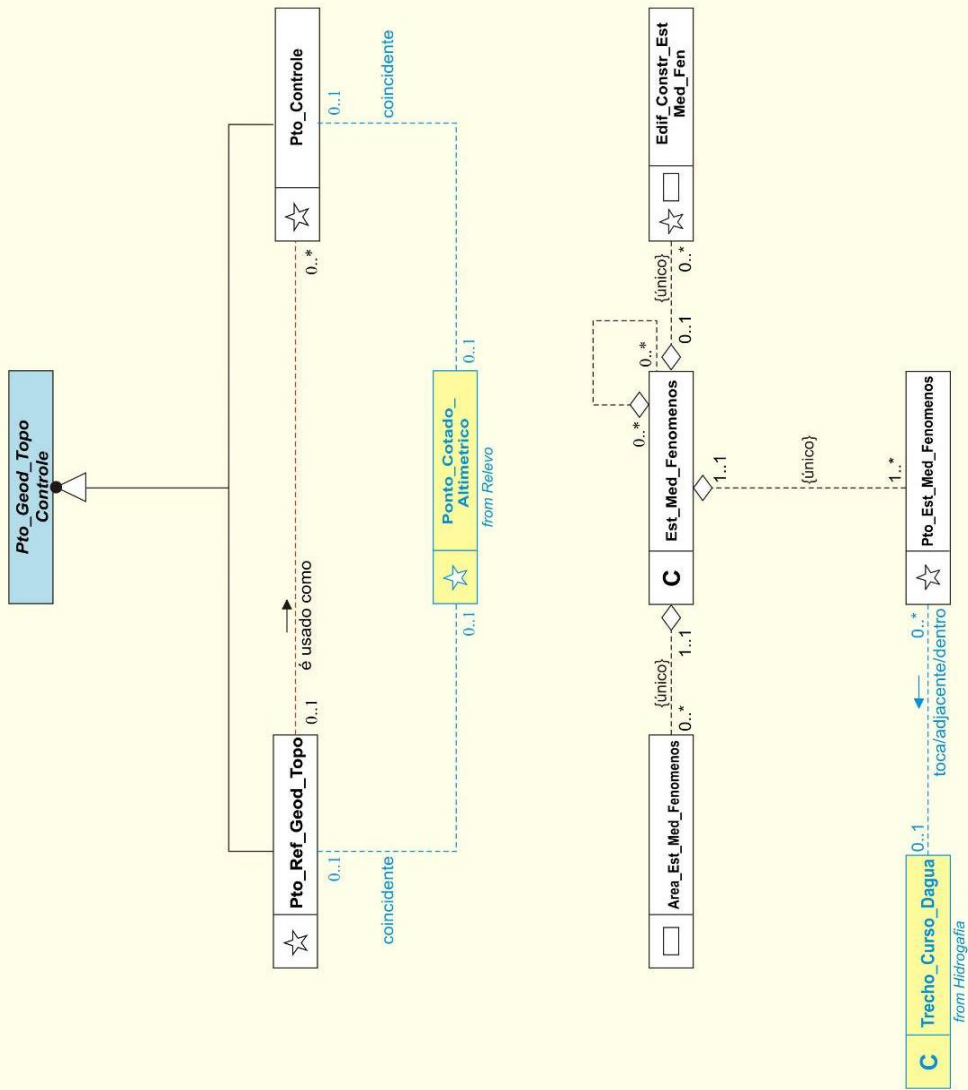
Categoria ABASTECIMENTO DE ÁGUA E SANEAMENTO BÁSICO

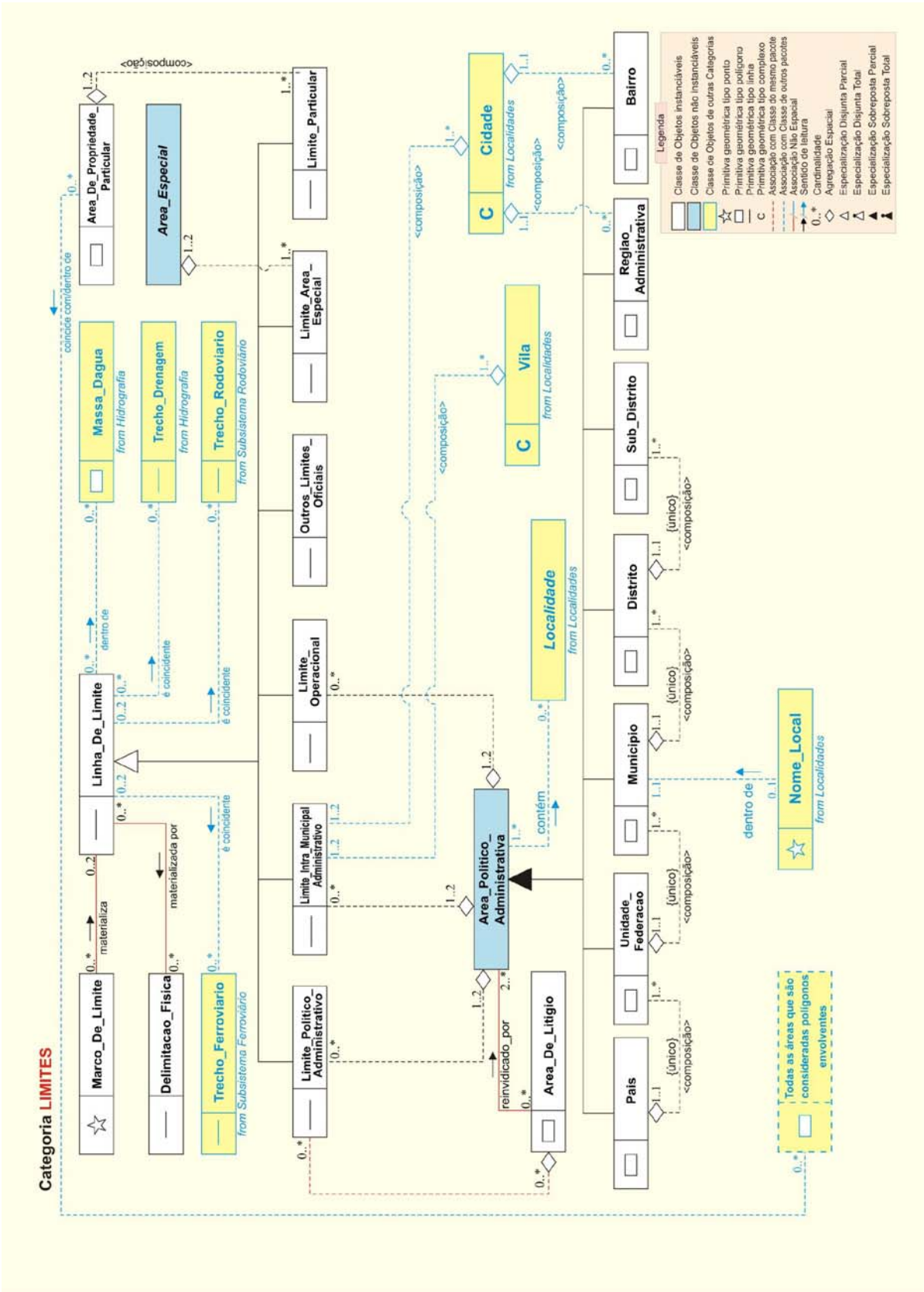




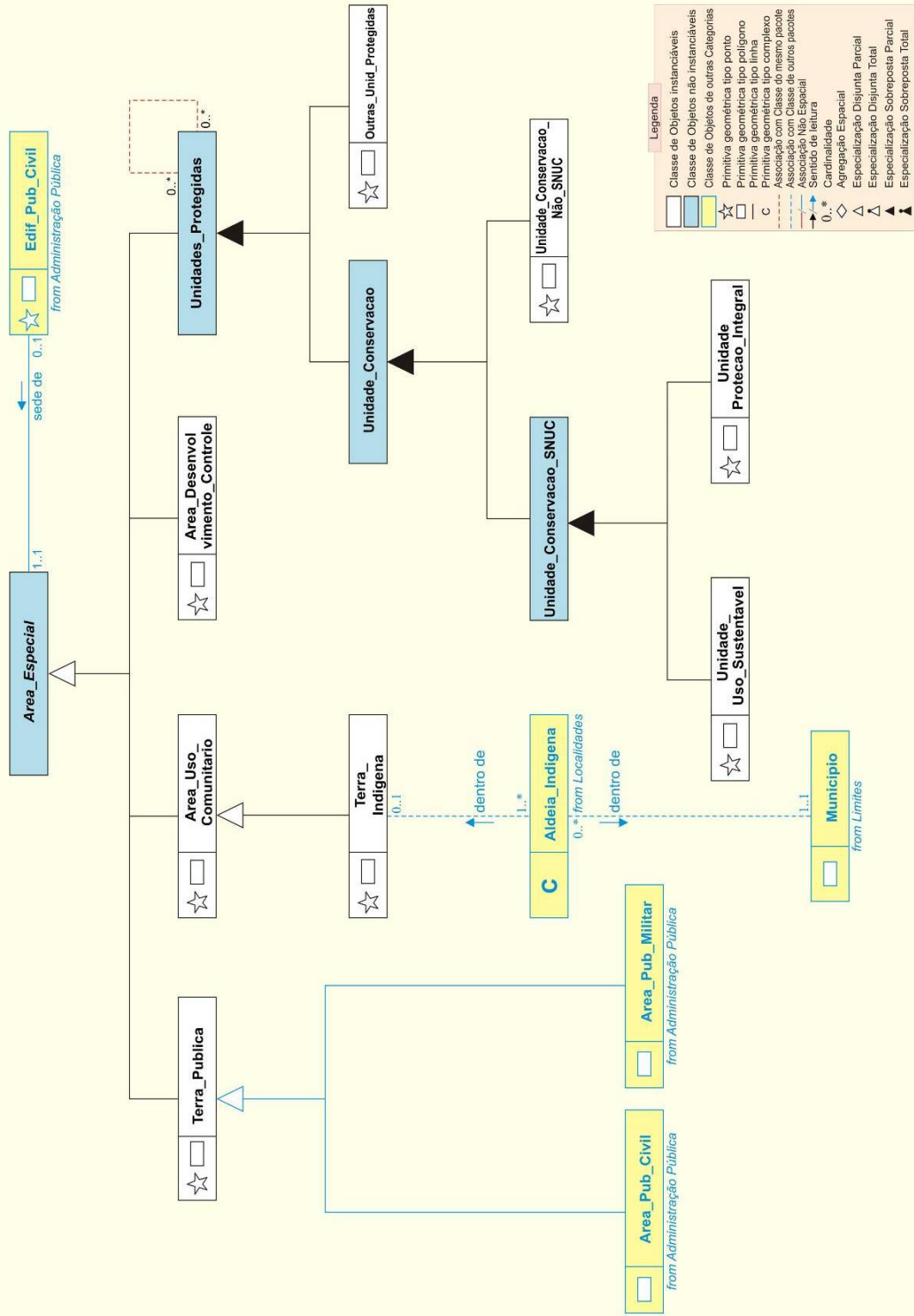


Categoria PONTOS DE REFERÊNCIA

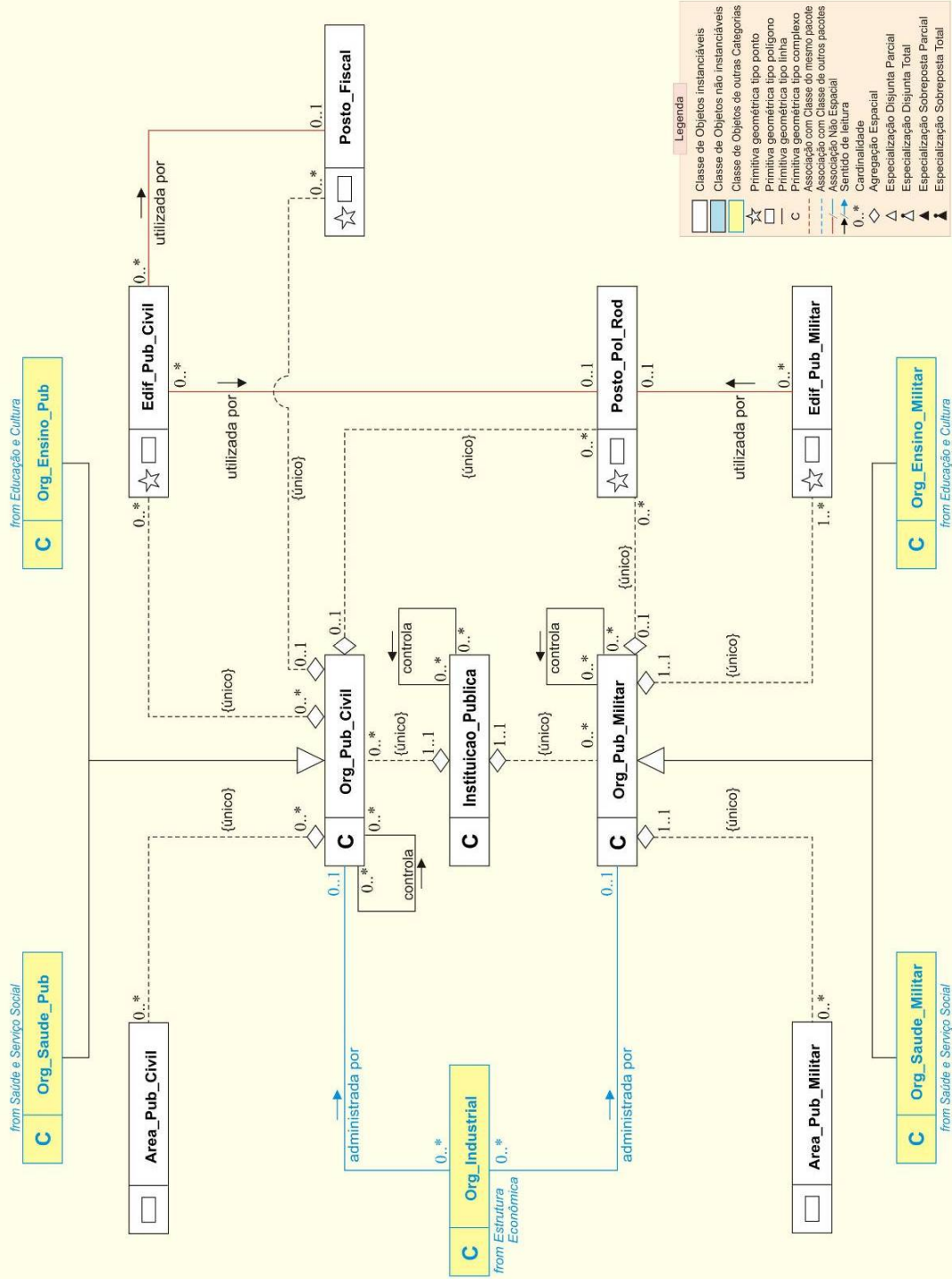




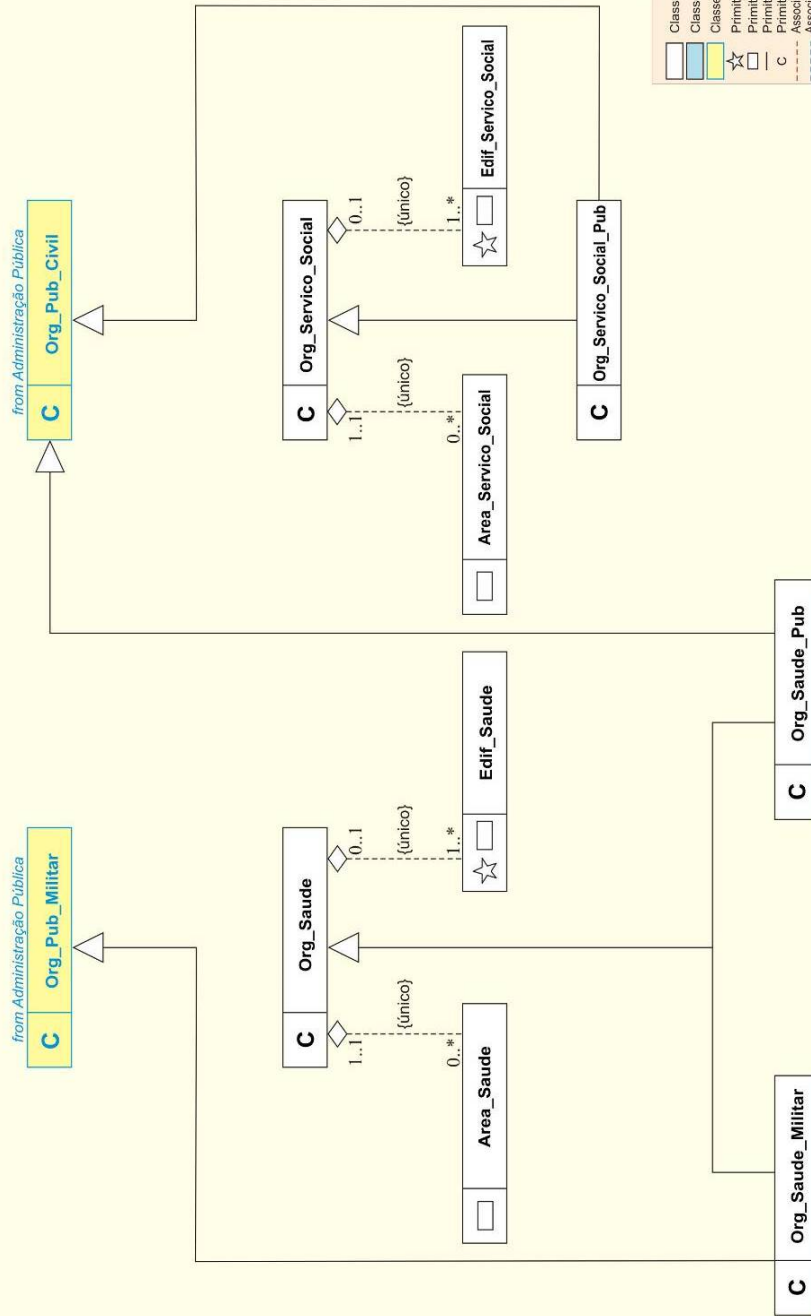
Categoria LIMITES



Categoria ADMINISTRAÇÃO PÚBLICA



Categoria SAÚDE E SERVIÇO SOCIAL



Legenda

- Classe de Objetos instanciáveis
- Classe de Objetos não instanciáveis
- Classe de Objetos de outras Categorias
- Primitiva geométrica tipo ponto
- Primitiva geométrica tipo polígono
- Primitiva geométrica tipo linha
- Primitiva geométrica tipo complexo
- Associação com Classe do mesmo pacote
- Associação com Classe de outros pacotes
- Sentido de leitura
- Cardinalidade
- Agregação Especial
- Especialização Disjunta Parcial
- Especialização Disjunta Total
- Especialização Sobreposta Parcial
- Especialização Sobreposta Total