



**Prefeitura de Belo Horizonte  
PBH**

**Empresa de Informática e Informação do Município de Belo Horizonte S/A  
PRODABEL**

# **Geoprocessamento Corporativo**

**Manual de usuário  
Modelagem de banco de dados geográficos com o EA**

**Belo Horizonte - MG  
3 April, 2012**

## Sumário

1 Introdução .....	3
2 Modelagem de dados geográficos .....	3
2.1 Modelo OMT-G.....	3
2.2 Perfil UML.....	6
2.3 Model-Driven Architecture (MDA).....	6
3 Configuração da ferramenta para utilização da customização.....	7
3.1 – Configurações iniciais.....	7
3.2 - Modelando com o perfil OMT-G .....	12
4 Transformação de modelos e geração de <i>scripts</i> .....	20
4.1 Oracle Spatial .....	22
4.2 PostGIS .....	29
Referências bibliográficas .....	33

# 1 Introdução

A ferramenta CASE *Enterprise Architect* (EA) fornece suporte à modelagem de sistemas de informação utilizando a linguagem de modelagem UML nas versões mais atuais e conforme padrão publicado pelo *Object Management Group* (OMG). Dentre os vários recursos disponíveis na ferramenta, há a possibilidade de estender a linguagem UML através do mecanismo de perfil UML para adaptá-la a um domínio particular e gerar transformações de modelos de forma automática utilizando a abordagem *Model Driven Architecture* (MDA).

A customização da ferramenta para o domínio de dados geográficos torna possível a realização da modelagem conceitual de bancos de dados geográficos usando o modelo OMT-G e a geração dos *scripts* de forma automática usando os construtores do SGBD *Oracle Spatial* e também do *PostGIS*. Além de aumentar a produtividade e melhorar a documentação e manutenção no desenvolvimento de modelos de dados geográficos, a customização da ferramenta também contribui para o estabelecimento de um padrão para modelagem de dados geográficos na Prefeitura de Belo Horizonte.

## 2 Modelagem de dados geográficos

### 2.1 Modelo OMT-G

O modelo OMT-G foi proposto por Borges et al. (2001) para dar suporte à modelagem de dados geográficos. O modelo foi baseado, inicialmente, no OMT (Rumbaugh et al., 1991) e, posteriormente, adaptado para a notação da UML (OMG, 2007). Ele possui primitivas para representar as características geométricas e topológicas dos dados geo-espaciais, assim como os relacionamentos espaciais.

Conforme Casanova et al. (2005), o modelo OMT-G é baseado em três conceitos principais: classes, relacionamentos e restrições de integridade espaciais. A customização realizada para o EA suporta as classes e os relacionamentos do OMT-G. A Figura 2.1 mostra as classes *georreferenciadas* do modelo OMT-G.

## GEO-CAMPO



## GEO-OBJETO

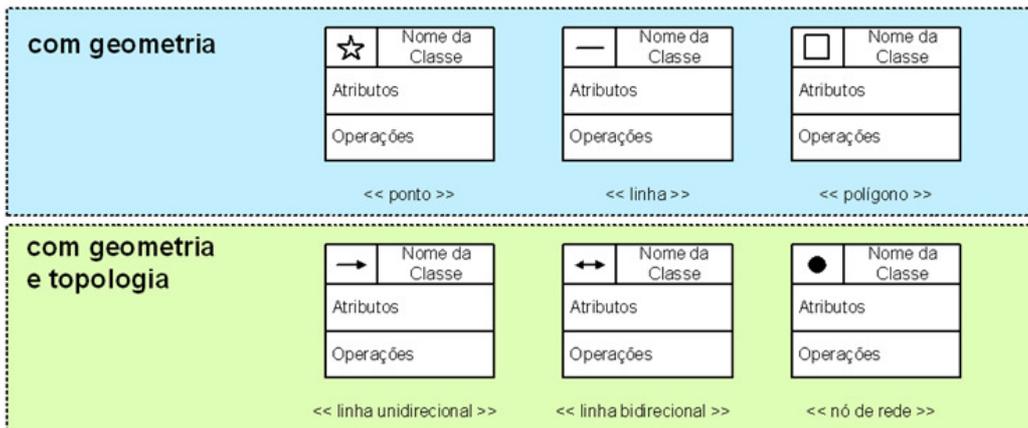


Figura 2.1 – Classes do modelo OMT-G

As classes podem ser *georreferenciadas* ou *convencionais*. As *convencionais* são as classes que não possuem propriedades geométricas e são simbolizadas como na UML. As classes *georreferenciadas*, as quais são representadas incluindo no canto superior esquerdo um retângulo que é usado para indicar a forma geométrica da representação, são especializadas em classes do tipo *geo-campo* (representam objetos e fenômenos distribuídos continuamente no espaço) e *geo-objeto* (representam objetos geográficos particulares, individualizáveis, associados a elementos do mundo real). As classes do tipo *geo-objeto* ainda são subdivididas em *geo-objeto com geometria* e *geo-objeto com geometria e topologia*. A classe *geo-objeto com geometria* representa objetos que possuem apenas propriedades geométricas, e é especializada nas classes: *Ponto*, *Linha* e *Polígono*. A classe *geo-objeto com geometria e topologia* representa objetos que possuem, além das propriedades geométricas, propriedades de conectividade topológica e é especializada nas classes: *Linha unidirecional*, *Linha bidirecional* e *Nó de rede*.

O modelo OMT-G dá suporte aos seguintes relacionamentos entre classes: *associações simples*, *relacionamentos espaciais* e *relações topológicas de rede*. Todos esses relacionamentos presentes no OMT-G são mostrados na Figura 2.2.

## RELACIONAMENTOS

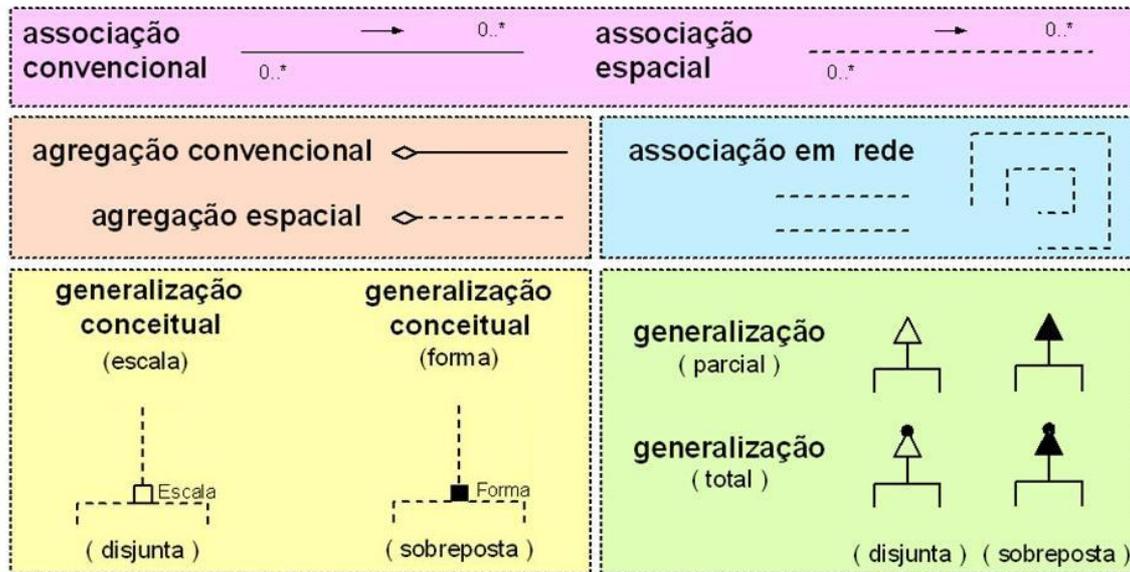


Figura 2.2 – Relacionamentos do modelo OMT-G

As associações simples ou convencionais são indicadas por linhas contínuas, enquanto relacionamentos espaciais (associação espacial) são indicados por linhas pontilhadas. Em relação à cardinalidade, é usada a mesma notação da UML. Os relacionamentos de rede (associação em rede) são relacionamentos entre objetos que estão conectados uns com os outros. Relacionamentos de rede são indicados por duas linhas pontilhadas paralelas.

Com relação à agregação, o modelo OMT-G segue a notação empregada na UML para a agregação convencional e quando a agregação ocorre entre classes *georreferenciadas* utiliza-se a *agregação espacial*, cuja diferença na representação em relação à convencional está na linha pontilhada. A agregação espacial indica que a geometria de cada parte deve estar contida na geometria do todo.

O modelo OMT-G possibilita, também, a definição de generalização, a qual também segue a notação da UML. Para diferenciar uma generalização total de uma parcial é utilizado um ponto no ápice do triângulo para denotar a totalidade e para diferenciar uma generalização disjunta de uma sobreposta é utilizado o triângulo preenchido para denotar a sobreposição. Existe ainda a generalização conceitual, utilizada para registrar a necessidade de representações diferentes para um mesmo objeto.

Maiores detalhes a respeito do modelo OMT-G podem ser vistos em Borges et al. (2001) e em Casanova et al. (2005).

## **2.2 Perfil UML**

A UML é uma linguagem visual de modelagem que pode ser usada em vários domínios de aplicações (OMG, 2007). Existem, porém, situações nas quais os construtores da UML não são capazes de expressar todos os conceitos de determinados domínios. Sendo assim, como cita Eriksson et al. (2004), para evitar que a UML se tornasse complexa demais, seus criadores tornaram-na extensível, ou seja, é possível adaptá-la a um domínio ou plataforma específica, por meio de seus mecanismos de extensão, que são: estereótipos, *tagged values* e *constraints*.

O conjunto desses mecanismos de extensão pode ser agrupado em um perfil UML. A intenção do mecanismo de perfil UML é fornecer um meio direto para adaptar um metamodelo existente com construtores que são específicos para um domínio, plataforma ou método em particular. O mecanismo de perfis é consistente com a *Meta Object Facility Specification* (MOF) (OMG, 2007).

## **2.3 Model-Driven Architecture (MDA)**

A principal característica da abordagem MDA (OMG, 2003) é a ênfase dada à modelagem. Porém, os artefatos produzidos nos diferentes níveis de abstração desta abordagem são modelos formais, isto é, modelos armazenados de tal forma que possam ser entendidos por computadores. Desta forma, é possível, por meio de regras de transformação e de uma linguagem específica, gerar modelos de níveis mais baixos de abstração e até mesmo código-fonte ou *scripts* de banco de dados, partindo de um nível conceitual.

Os níveis de abstração usados em MDA são o *Computational Independent Model* (CIM), *Platform Independent Model* (PIM) e o *Platform Specific Model* (PSM). Tanto o CIM como o PIM são modelos com um nível de abstração alto e independente de qualquer tecnologia de implementação. O PIM deve ser transformado em um PSM, que já leva em consideração detalhes de alguma tecnologia de implementação específica. O passo seguinte é a transformação do PSM para código-fonte ou *script*. Essa transformação é relativamente direta pelo fato do PSM ser ajustado a uma

tecnologia específica. A Figura 2.3 ilustra os principais níveis de abstração de modelos da abordagem MDA.

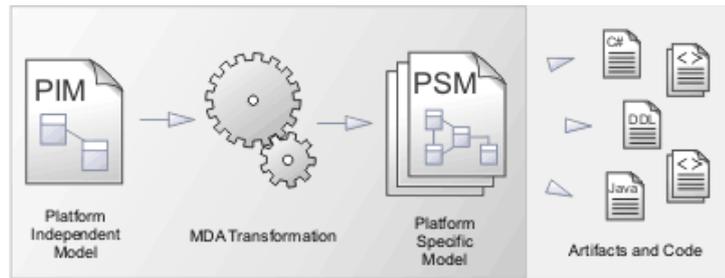


Figura 2.3 – Transformação de modelos em MDA  
(Fonte: Sparx System, 2011)

### 3 Configuração da ferramenta para utilização da customização

A customização do EA para modelagem de dados geográficos envolveu todos esses conceitos apresentados anteriormente. Inicialmente, foi criado um perfil UML, cujos estereótipos representam os elementos do modelo OMT-G. Posteriormente, foram criadas as regras de transformação automática de modelos, conforme a abordagem MDA, sendo possível gerar *scripts* para o *Oracle Spatial* e *PostGIS* por meio destas transformações. A abordagem MDA utiliza os termos CIM, PIM e PSM para separar os diferentes níveis de abstração de modelos. Porém, pode-se utilizar, também, a terminologia da abordagem clássica de projeto de banco de dados, a qual separa os níveis de abstração em modelo conceitual, lógico e físico. As seções seguintes mostram como configurar o EA para permitir a modelagem de dados geográficos utilizando a customização feita.

#### 3.1 – Configurações iniciais

Para fazer uso do perfil UML e dos demais recursos da customização, como a geração de *scripts* de banco de dados geográficos, em primeiro lugar, deve-se configurar a ferramenta para tornar o perfil disponível para realizar a modelagem. O perfil e as regras de transformação foram encapsulados em um arquivo XML, que foi gerado pela própria ferramenta, a qual denomina esse processo de *Model Driven Generation*

(MDG). Esse arquivo XML deve ser importado em um novo projeto criado na ferramenta. Os passos seguintes mostram como deve ser feita essa configuração:

1 – Abra a ferramenta e crie um novo projeto (File -> New Project). O projeto deve ser nomeado e salvo em alguma pasta usando a extensão .eap. Na janela “Select models”, que se abre após clicar em *Salvar*, basta clicar em OK.

2 – Clique em Tools -> Import Technology. A janela mostrada na Figura 3.1 é aberta. Em *Filename*, deve-se incluir o caminho do arquivo XML (OMTG\_MDG.xml). Ao incluir esse arquivo, algumas opções com *checkbox* se tornarão disponíveis. Mantenha essas opções marcadas, conforme mostrado na figura, e clique em *Import*.

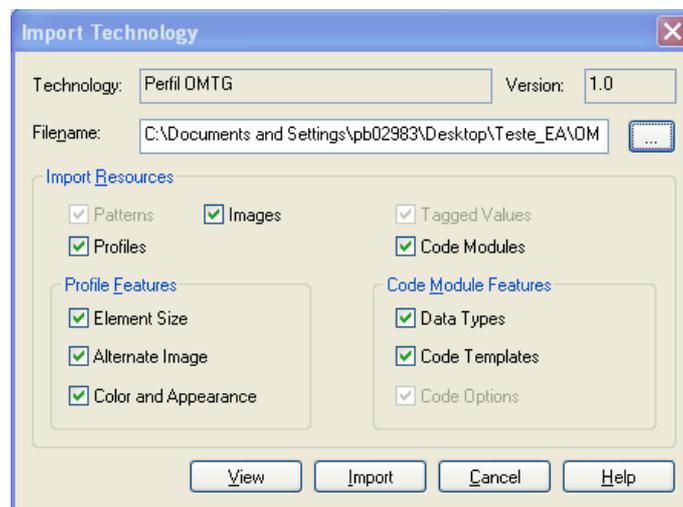


Figura 3.1 – Importando o arquivo MDG

3 – Clique em Settings -> MDG Technologies. A janela mostrada na Figura 3.2 é aberta. Nesse passo, o perfil será ativado para se tornar disponível na lista de *toolboxes* da ferramenta.

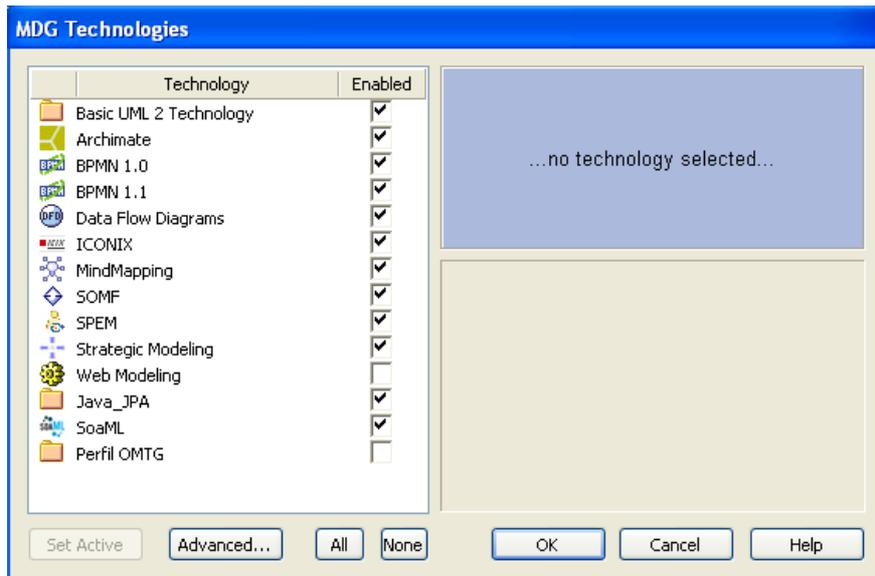


Figura 3.2 – Ativando o perfil

4 – Considerando ainda a janela mostrada na Figura 3.2, habilite o *checkbox* da coluna *Enabled* correspondente à opção *Perfil OMTG* da coluna *Technology*. Em seguida, clique no botão *Advanced*. A janela mostrada na Figura 3.3 é aberta. Clique em *Add* -> *Add Path* e inclua o caminho do diretório onde está o arquivo XML. Em seguida, clique em *OK*.

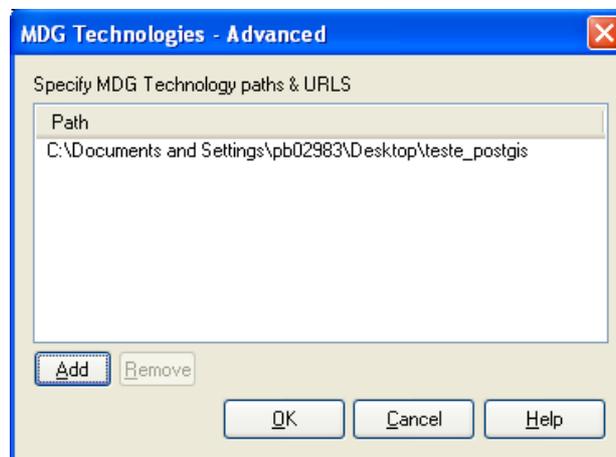


Figura 3.3 – Opções de ativação do arquivo MDG

5 – Será aberta uma caixa de diálogo (Figura 3.4) informando para reiniciar a ferramenta para que as mudanças sejam efetivadas. Clique em *OK* nesta caixa de diálogo e reinicie a ferramenta.



Figura 3.4 – Mensagem apresentada para reiniciar o EA

6 – Ao abrir novamente a ferramenta, selecione o projeto que foi criado anteriormente. Crie um diagrama de classes nesse projeto. Com as configurações realizadas anteriormente, o perfil já estará disponível, por meio da lista de *toolboxes*, bastando apenas torná-lo visível. Para isso, selecione a opção Perfil OMTG no campo *Profile* da barra de ferramentas, como mostrado na Figura 3.5.

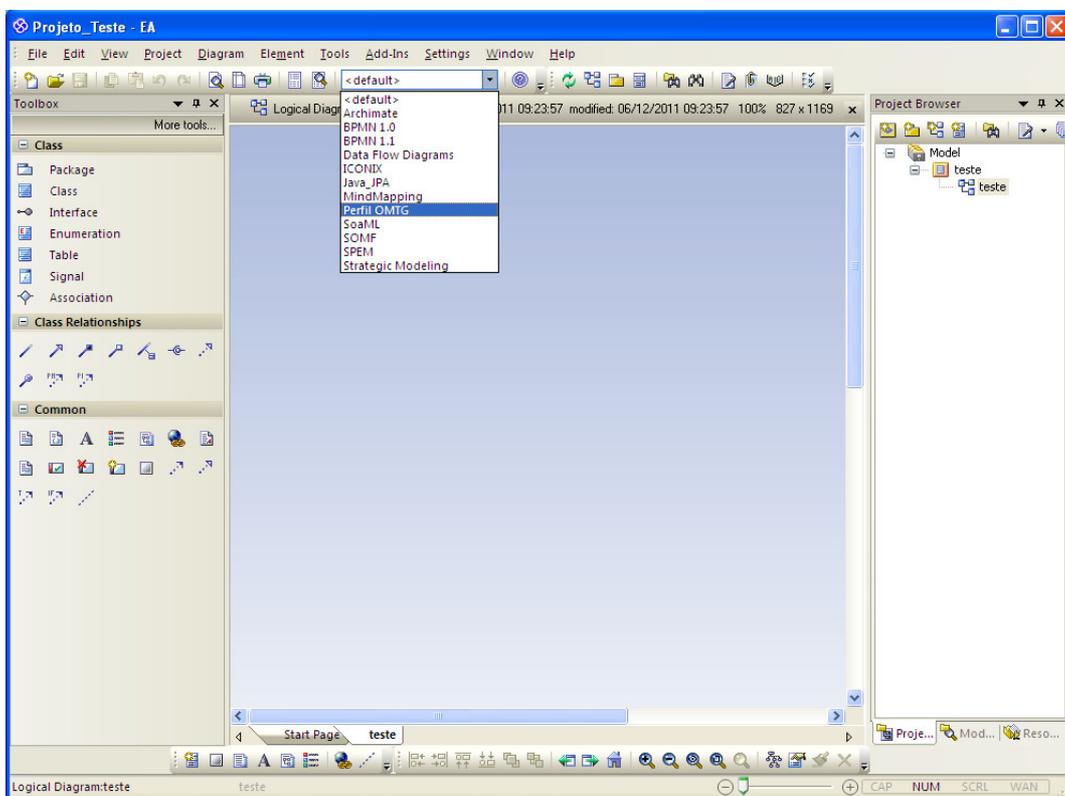


Figura 3.5 – Tornando o perfil visível na *toolbox* da ferramenta

7 – Clique em More Tools -> Set Toolbox Visibility. Será aberta a caixa de diálogo mostrada na Figura 3.6. Marque as opções da coluna *Visible* referente ao *namespace* Perfil OMTG, como mostrado na figura. Em seguida, clique em OK.

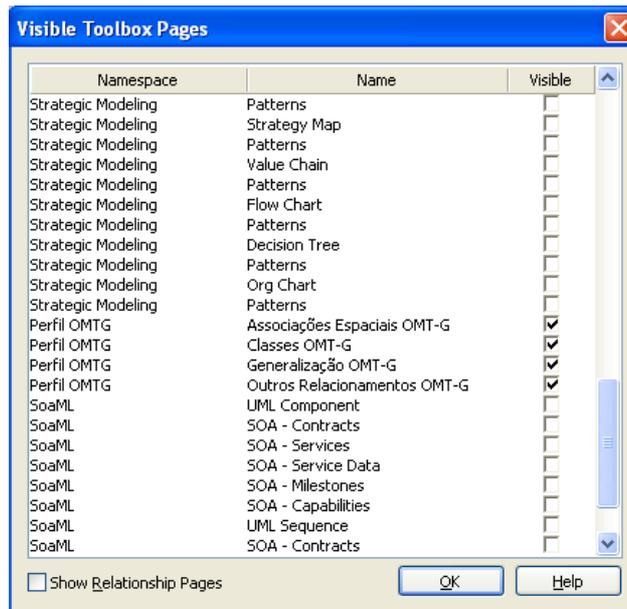


Figura 3.6 - Ativando o perfil

Após esses passos, o perfil aparecerá na lista de *toolboxes* da ferramenta. Possivelmente, as páginas do perfil estarão minimizadas. Para maximizar as quatro páginas do perfil ao mesmo tempo, clique em `More Tools -> OMT-G`. A aparência da ferramenta ficará como mostrado na Figura 3.7, com as quatro páginas do perfil OMT-G (Associações Espaciais OMT-G, Classes OMT-G, Generalização OMT-G e Outros Relacionamentos OMT-G) em destaque no *toolbox* da ferramenta. Para começar a modelar, basta selecionar o elemento desejado no *toolbox* e arrastá-lo ao diagrama.

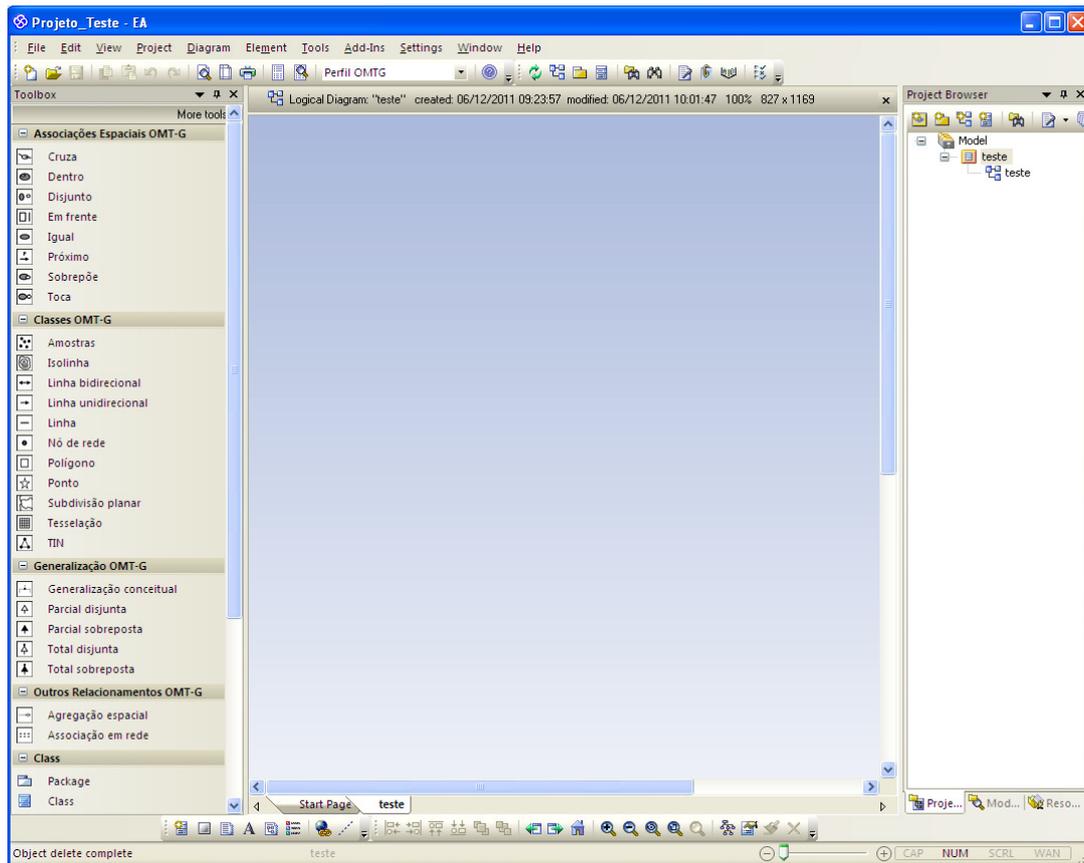


Figura 3.7– Perfil configurado na ferramenta

### 3.2 - Modelando com o perfil OMT-G

O *toolbox* do perfil OMT-G está dividido em quatro páginas, conforme pode ser visto na Figura 3.8. Na página *Classes OMT-G* estão as classes *georreferenciadas* presentes no modelo OMT-G, tanto geo-campo como geo-objetos. Na página *Associações Espaciais OMT-G* estão as possíveis formas de relacionamentos espaciais, os quais são representados pelos estereótipos *Cruza*, *Dentro*, *Disjunto*, *Em frente*, *Próximo*, *Sobrepõe* e *Toca*. Na página *Generalização OMT-G* estão os estereótipos que representam as formas de generalização do modelo OMT-G, a saber, *Generalização Conceitual*, *Generalização Parcial Disjunta*, *Parcial Sobreposta*, *Total Disjunta* e *Total Sobreposta*. E na página *Outros Relacionamentos OMT-G* estão representados os estereótipos *Agregação espacial* e *Associação em rede*.



Figura 3.8 – *Toolbox* com os elementos do perfil OMT-G

Ao se adicionar classes *georreferenciadas* no diagrama, os estereótipos com as características espaciais são mostrados na sua forma gráfica e textual, conforme mostrado na Figura 3.9, com as classes *Árvore* (Ponto), *Meio-fio* (Linha) e *Edificação* (Polígono). Para alterar essa propriedade, deixando apenas a forma gráfica do estereótipo, clique com o botão direito no diagrama e em *Properties*. A caixa de diálogo mostrada na Figura 3.10 será aberta. Selecione a aba *Elements* e em *Element Appearance*, desmarque a opção *Show Element Stereotypes*, deixando marcadas as opções *Use Stereotype Icons* e *Show Element Property String*, como mostrado na Figura 3.10. Em seguida, clique em *OK*.

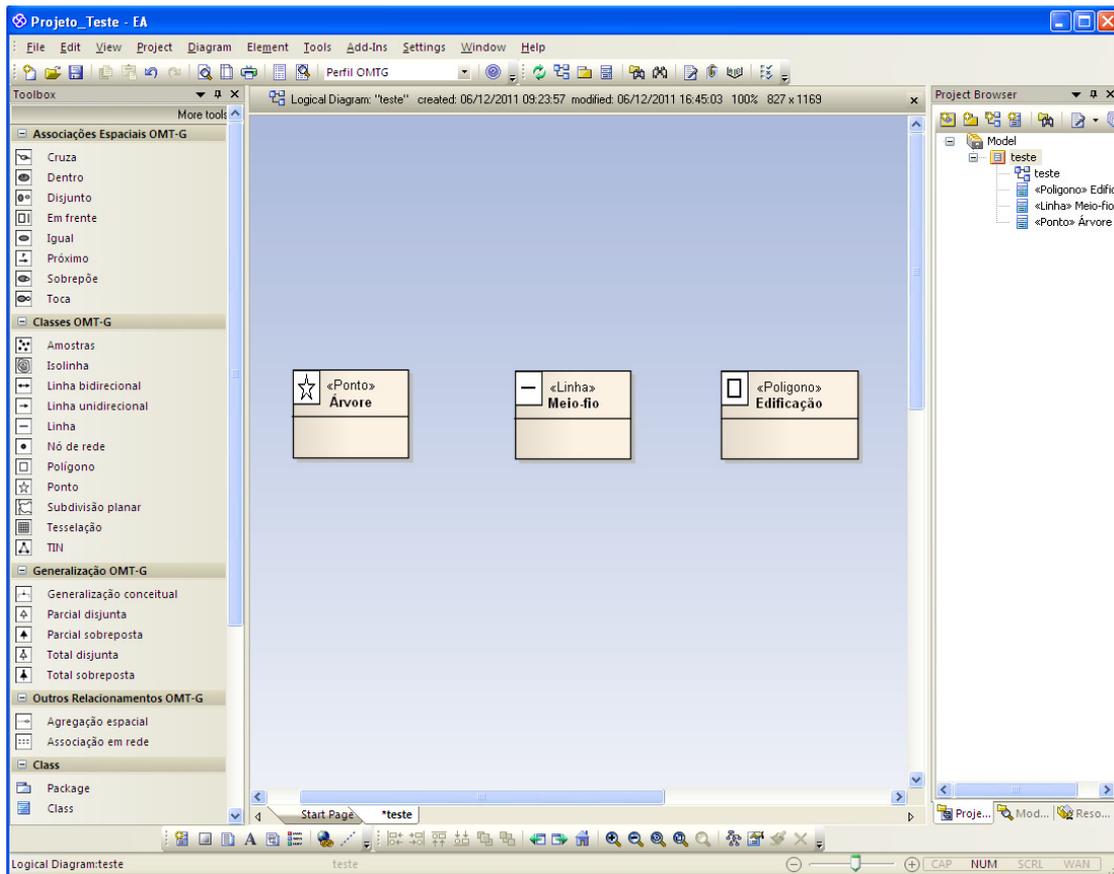


Figura 3.9 – Modelando com classes *georreferenciadas* do perfil OMT-G

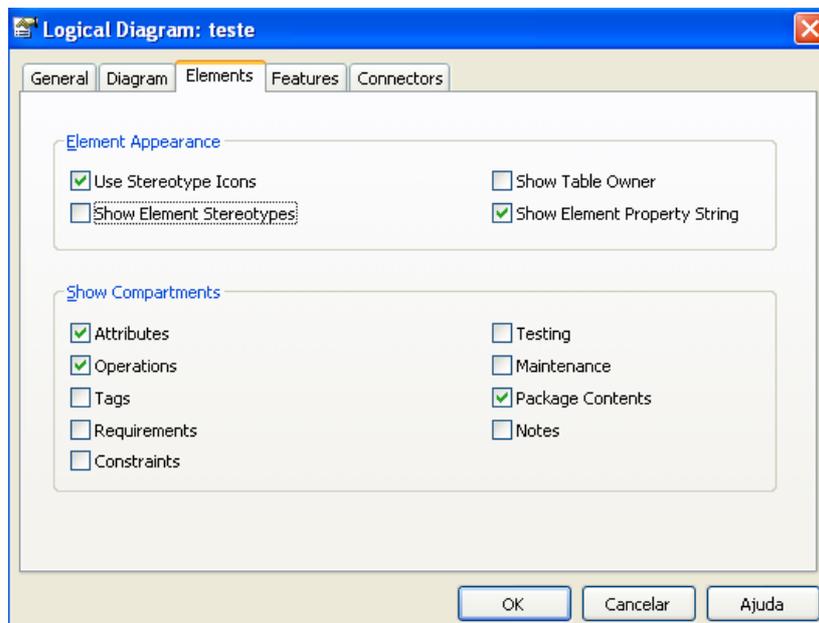


Figura 3.10 – Alterando propriedade de visualização dos estereótipos

Com as modificações realizadas nas propriedades do diagrama, as classes serão visualizadas como mostrado na Figura 3.11, com os estereótipos apenas na sua forma gráfica.

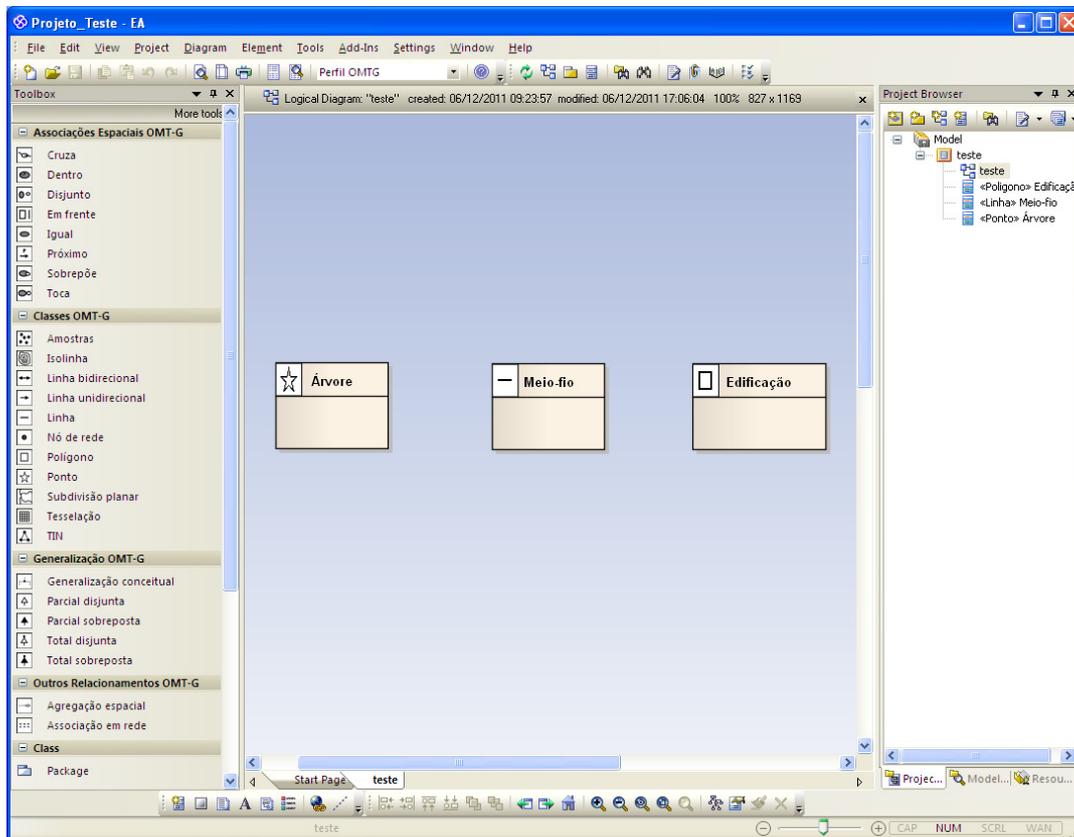


Figura 3.11 – Estereótipos do perfil OMT-G sendo visualizados apenas na forma gráfica

Para atribuir uma característica temporal à classe georreferenciada, abra as propriedades da classe em questão e selecione a aba *Tagged Values*, como pode ser visto na Figura 3.12. Clique no botão *New Tagged Value*. A janela mostrada na Figura 3.13 será aberta. Em *Tag*, digite *temporal*, como mostrado na figura. E em *Value*, entre com o valor desejado, o qual pode ser *instante* ou *intervalo*. Em seguida, clique em OK. O resultado é mostrado na Figura 3.14, a qual mostra duas classes com os dois tipos de características temporais.

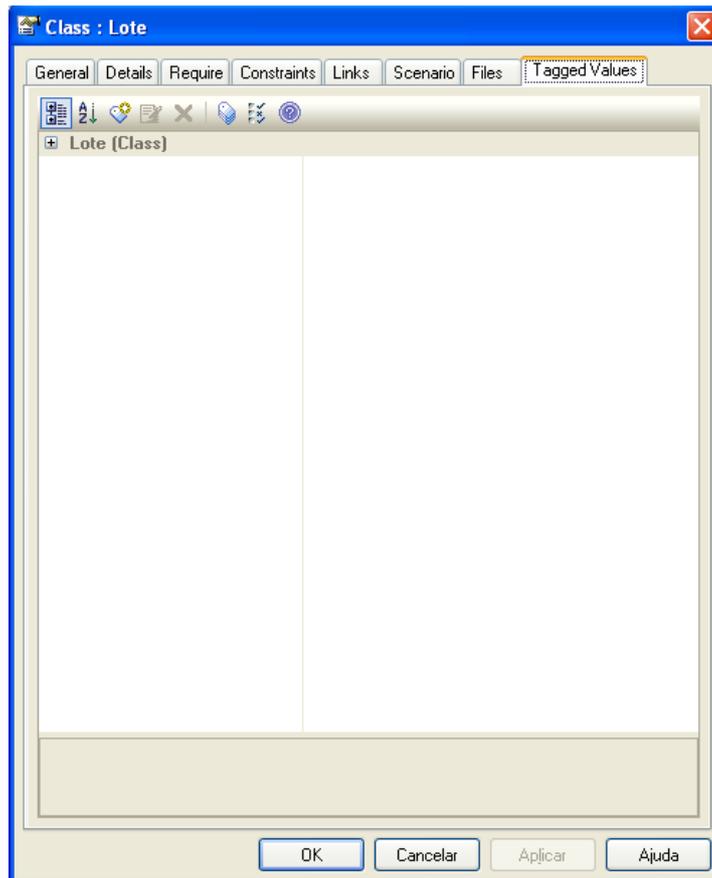


Figura 3.12 – Adicionando características temporais a classes georreferenciadas

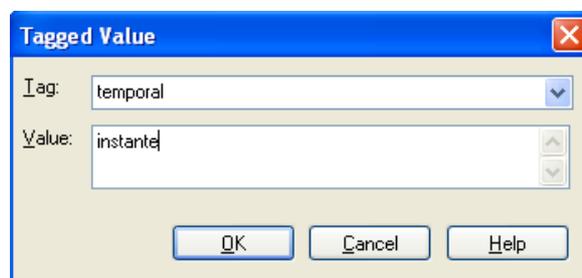


Figura 3.13 – Selecionando o tipo de característica temporal para a classe



Figura 3.14 – Classes com características temporais

Para fazer uso dos relacionamentos na modelagem, basta clicar no relacionamento desejado e ligar uma classe na outra. A Figura 3.15 ilustra a utilização

de associação espacial. Neste exemplo em particular, está sendo utilizado o estereótipo *Dentro* para conectar as classes *Edificação* e *Lote*.

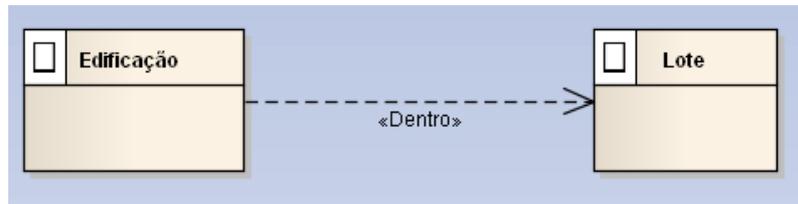


Figura 3.15 – Exemplo usando *Associação espacial*

A Figura 3.16a ilustra a utilização de associação em rede. Para ocultar o nome do estereótipo, clique com o botão direito no nome do estereótipo e selecione a opção *Hide Label*. Para adicionar um nome ao relacionamento, selecione o relacionamento, clique com o botão direito do mouse nele e escolha a opção *Association Properties*. Na aba *General* da janela que se abre, preencha o campo *Name* e no campo *Direction*, escolha a opção *Unspecified*. Em seguida, clique em OK. Após essas configurações, a aparência do relacionamento ficará como mostrado na Figura 3.16b.



Figura 3.16a – Associação em rede antes da configuração



Figura 3.16b – Associação em rede depois de configurada

A Figura 3.17 mostra um exemplo utilizando o relacionamento *Agregação espacial*. Da mesma forma como foi feito no passo anterior para a *Associação em rede*, nesta situação também foi utilizada a opção para ocultar o nome do estereótipo.

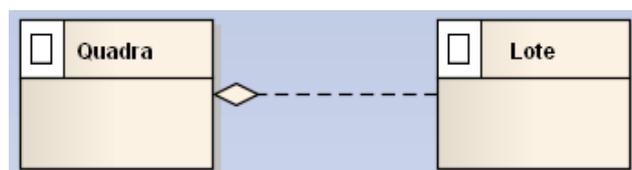


Figura 3.17 – Exemplo usando Agregação espacial

A Figura 3.18a ilustra um exemplo de modelagem usando o estereótipo *Generalização conceitual*. Da mesma forma que nos exemplos anteriores, deve-se ocultar o nome do estereótipo clicando com o botão direito no nome do estereótipo e selecionando a opção *Hide Label*. Para que o relacionamento seja visualizado como na Figura 3.18b, deve-se ainda alterar o estilo de linha do relacionamento. Para isso, clique com o botão direito no relacionamento, selecione *Line Style* e clique na opção *Tree Style - Vertical*.

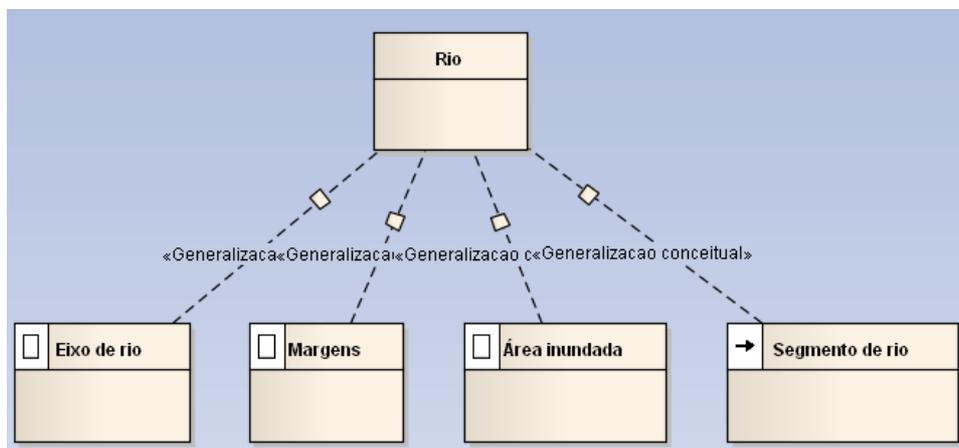


Figura 3.18a – Exemplo usando *Generalização conceitual* antes da configuração do relacionamento

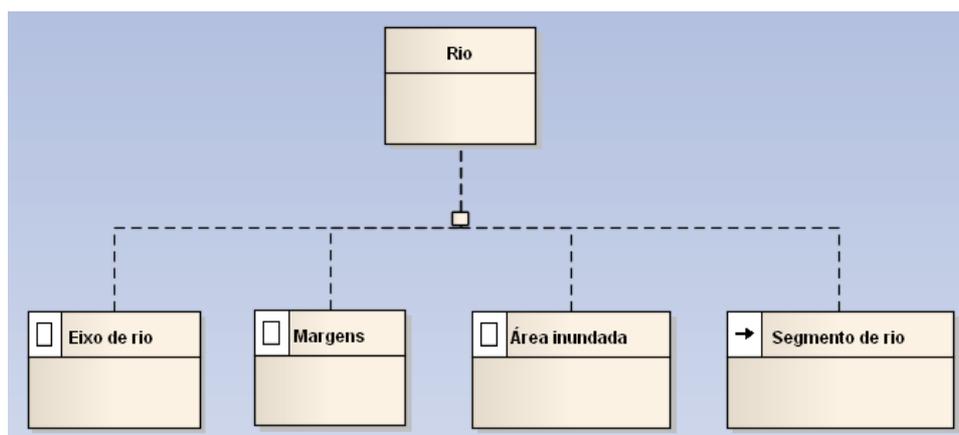


Figura 3.18b – Exemplo usando *Generalização conceitual* depois da configuração do relacionamento

A Figura 3.19 ilustra exemplos usando as outras possíveis formas de generalização do modelo OMT-G e que estão presentes no perfil, a saber, generalização parcial disjunta, generalização parcial sobreposta, generalização total disjunta e generalização total sobreposta.

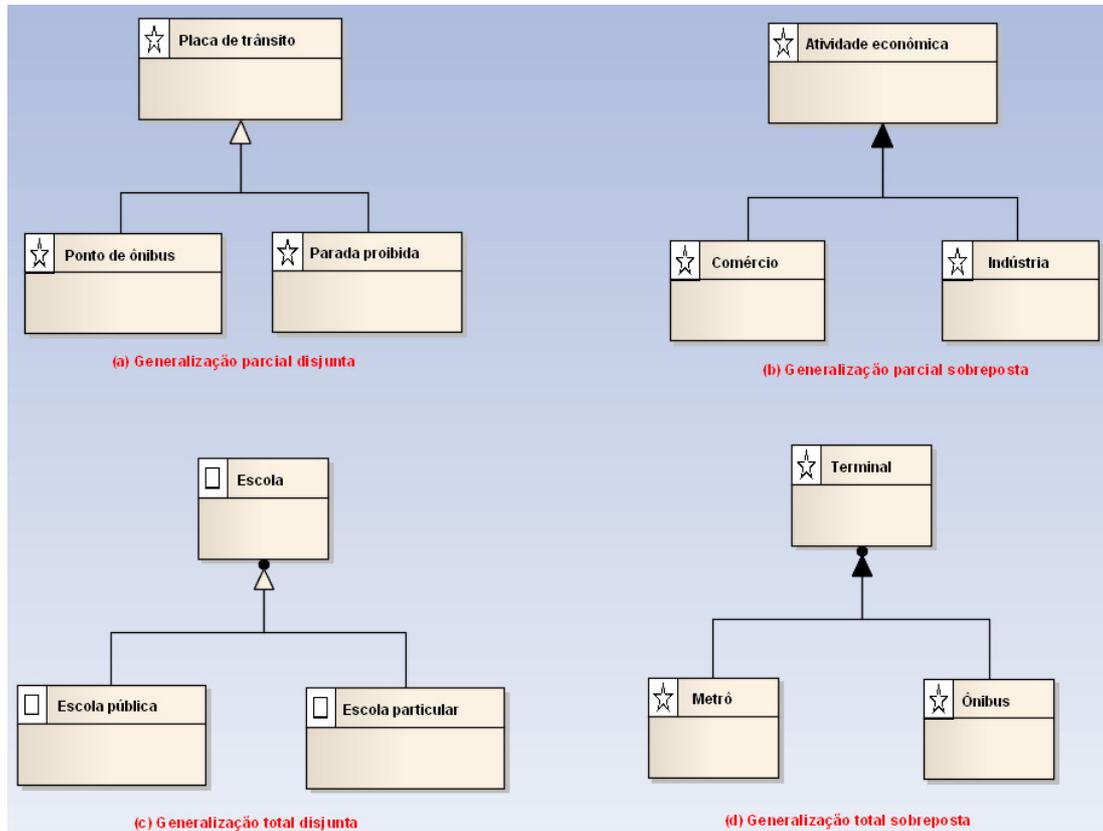


Figura 3.19 – Formas de generalização presentes no perfil OMT-G

### **OBSERVAÇÃO!**

Uma observação em relação à generalização deve ser verificada na hora de realizar transformações de modelos. Para realizar a transformação para o modelo de dados, o EA utiliza a estratégia em que é criada uma tabela para a superclasse, contendo todos os seus atributos e sua chave primária e é criada uma tabela para cada subclasse, usando a mesma chave primária da superclasse, e também estabelecendo-a como chave estrangeira em relação à tabela da superclasse; os atributos específicos são acrescentados na subclasse e a representação geográfica fica nas subclasses.

Por causa dessa estratégia, no modelo conceitual deve-se deixar a superclasse sem a representação geográfica para que o modelo de dados não seja gerado de forma

errada. Um exemplo de como se deve deixar o modelo conceitual nesse caso é mostrado na Figura 3.20.

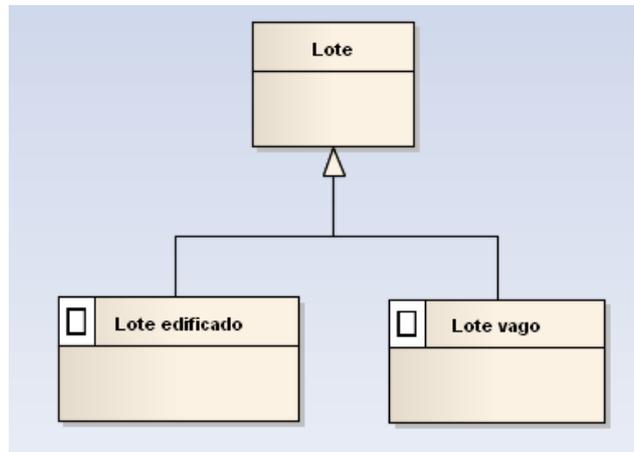


Figura 3.20 – Exemplo de generalização a ser usado para transformação de modelos

## 4 Transformação de modelos e geração de *scripts*

Uma das características deste projeto está na possibilidade de realizar transformações de forma automática de um modelo conceitual de banco de dados geográficos para modelos com detalhes específicos de uma tecnologia em particular e, posteriormente, gerar o *script* do banco. Foram feitas customizações para as tecnologias *Oracle Spatial* e *PostGIS*.

A Figura 4.1 ilustra o procedimento necessário para realizar a geração dos *scripts* de banco de dados geográficos. Inicialmente, é feita a modelagem conceitual utilizando o perfil UML criado para o modelo OMT-G. Na abordagem MDA, essa etapa é conhecida como CIM ou PIM e não leva em consideração detalhes de implementação. Posteriormente, esse modelo é transformado em outro modelo que já considera detalhes de plataformas específicas (PSM) e em seguida é feita a geração dos *scripts*.

Neste projeto, caso o projetista queira gerar os *scripts* para o *Oracle*, são gerados dois modelos intermediários, um com detalhes das tabelas e restrições de chave e outro com os índices espaciais e, em seguida, são gerados os *scripts* DDL. Caso queira gerar os *scripts* para o *PostGIS*, também são gerados dois modelos intermediários, um com detalhes das tabelas e chaves e outro com os índices espaciais e definição da coluna com geometria para as classes *georreferenciadas*. Em seguida, são gerados os *scripts* DDL.

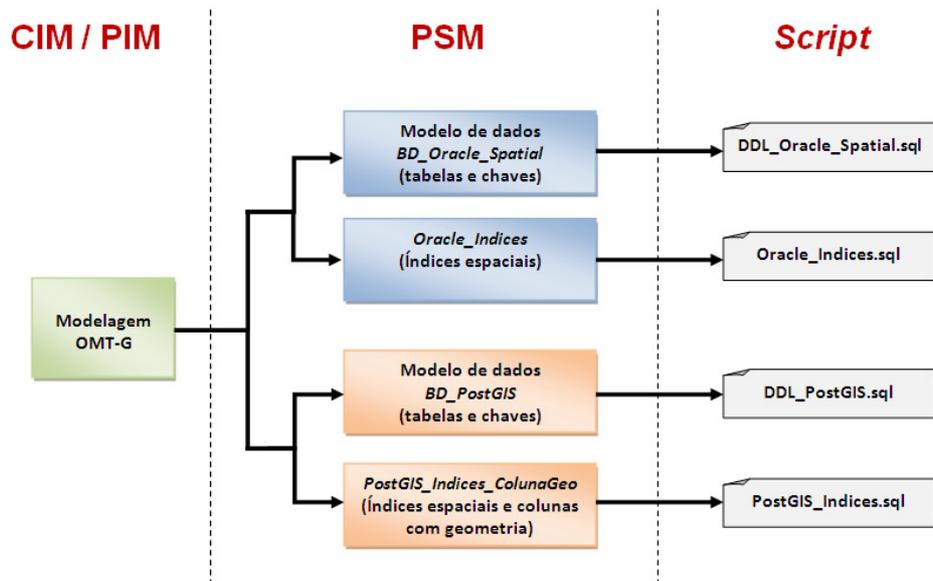


Figura 4.1 – Processo de transformação no EA

Para mostrar como é realizada a transformação de modelos e geração de *scripts* para essas duas plataformas, será utilizado o esquema conceitual mostrado na Figura 4.2, o qual consiste de quatro classes, sendo três delas classes *georreferenciadas* do tipo geo-objeto, com características espaciais de *Ponto* e *Polígono*. Neste exemplo foram utilizadas associações simples entre as classes.

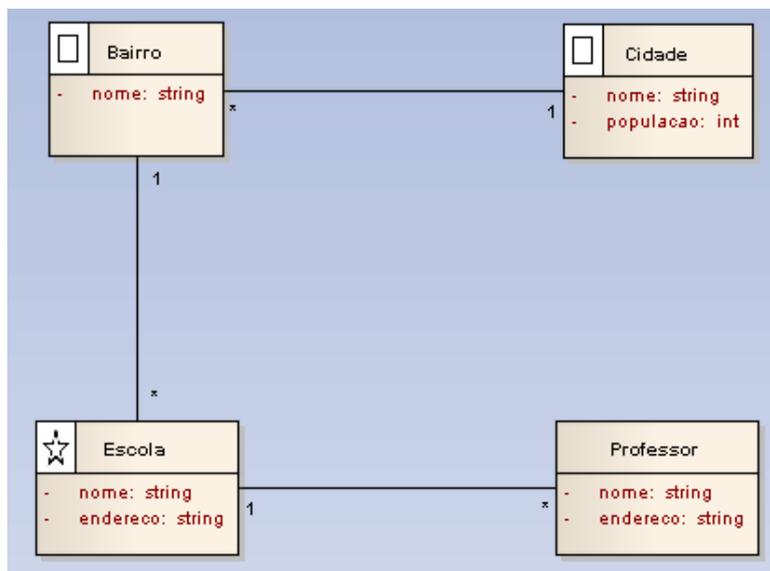


Figura 4.2 – Exemplo de esquema conceitual de banco de dados geográficos

## 4.1 Oracle Spatial

Antes de proceder com a transformação, sugere-se a criação de um novo pacote dentro do projeto atual para receber os modelos que serão provenientes da transformação. Para criar esse novo pacote, clique em *Add a Package* no *Project Browser* da ferramenta. Dê um nome ao pacote e clique em OK. Para este exemplo foi dado o nome `Modelo_Oracle` a esse novo pacote.

Para efetuar a transformação para a plataforma *Oracle Spatial*, clique com o botão direito no pacote do modelo conceitual e depois em *Transform Current Package*. A janela mostrada na Figura 4.3 será aberta com as opções para gerar a transformação. No lado direito da janela, em *Transformations*, estão as opções de plataformas para transformação e no lado esquerdo, em *Elements*, estão os elementos que serão transformados. Em *Transformations*, selecione as opções `BD_Oracle_Spatial` e `Oracle_Indices`. Ao selecionar as opções de transformações, é aberta a janela mostrada na Figura 4.4 para escolher o pacote para onde irão os elementos resultantes da transformação. Selecione o pacote criado anteriormente (`Modelo_Oracle`) tanto para a opção `BD_Oracle_Spatial` como para `Oracle_Indices` e clique em OK. Em seguida, clique em *Do Transform*.

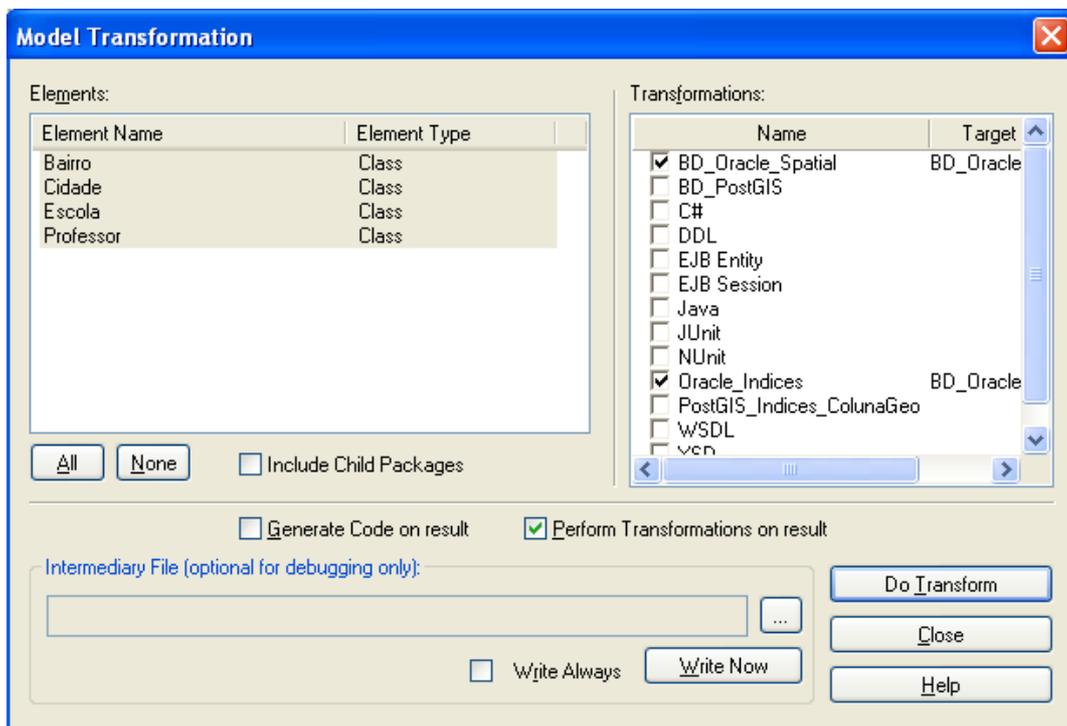


Figura 4.3 – Selecionando as opções de transformação de modelos para *Oracle Spatial*



Figura 4.4 – Selecionando o pacote que vai receber os modelos transformados

Como resultado desse processo de transformação, são criados dois sub-pacotes dentro do pacote `Modelo_Oracle`, como pode ser visualizado no *Project Browser* mostrado na Figura 4.5. O pacote denominado `BD_Oracle_Spatial` contém as definições de tabelas e restrições de chaves e o pacote `Indices_teste` será usado para gerar os índices espaciais.

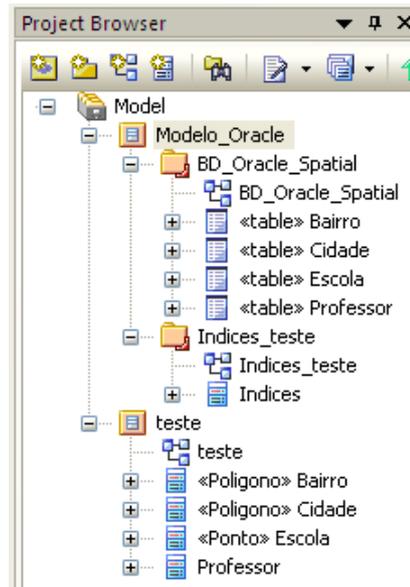


Figura 4.5 – *Project Browser* com os modelos resultantes da transformação

A Figura 4.6 ilustra o diagrama do pacote `BD_Oracle_Spatial`. Considerando a abordagem MDA, esse modelo é um PSM, pois já considera detalhes de uma plataforma particular. Neste modelo são criadas as chaves primárias e estrangeiras e para as classes georreferenciadas é criado um atributo do tipo `SDO_GEOMETRY`, o qual é específico do *Oracle Spatial*.

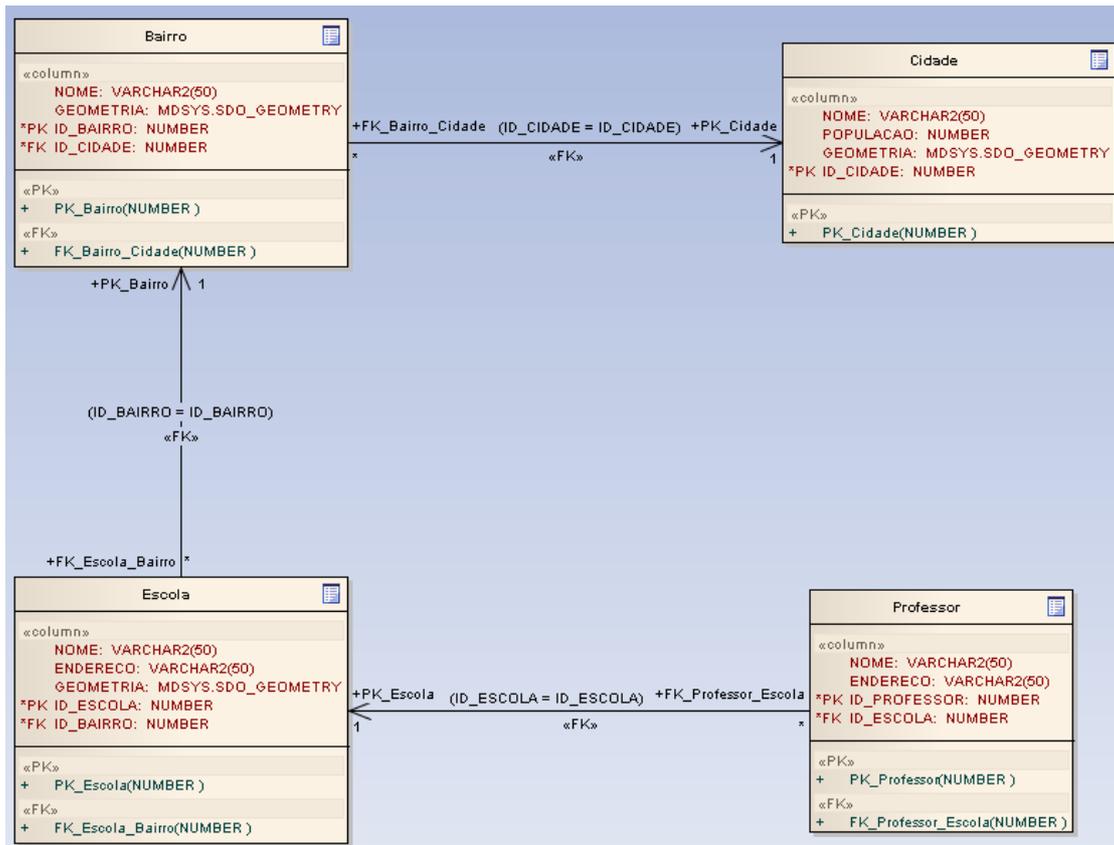


Figura 4.6 – Modelo resultante da transformação para a plataforma *Oracle Spatial*

A Figura 4.7 ilustra o diagrama do pacote `Indices_teste`, que será usado para a geração dos índices espaciais.



Figura 4.7 – Modelo para geração dos índices espaciais para o *Oracle Spatial*

Após essa transformação intermediária de modelos, pode-se, então, realizar a geração dos *scripts* de banco de dados geográficos. Isso será feito para cada um dos modelos gerados. Para gerar as tabelas e as restrições de chave, clique com o botão direito no pacote `BD_Oracle_Spatial` no *Project Browser* e selecione `Code Engineering -> Generate DDL`. Ao clicar em `Generate DDL`, será aberta a janela mostrada na Figura 4.8. Em *File Generation*, selecione a opção *Single File* e aponte o

caminho em que será gerado o arquivo com o *script*. Em seguida, clique em *Generate* e feche esta janela.

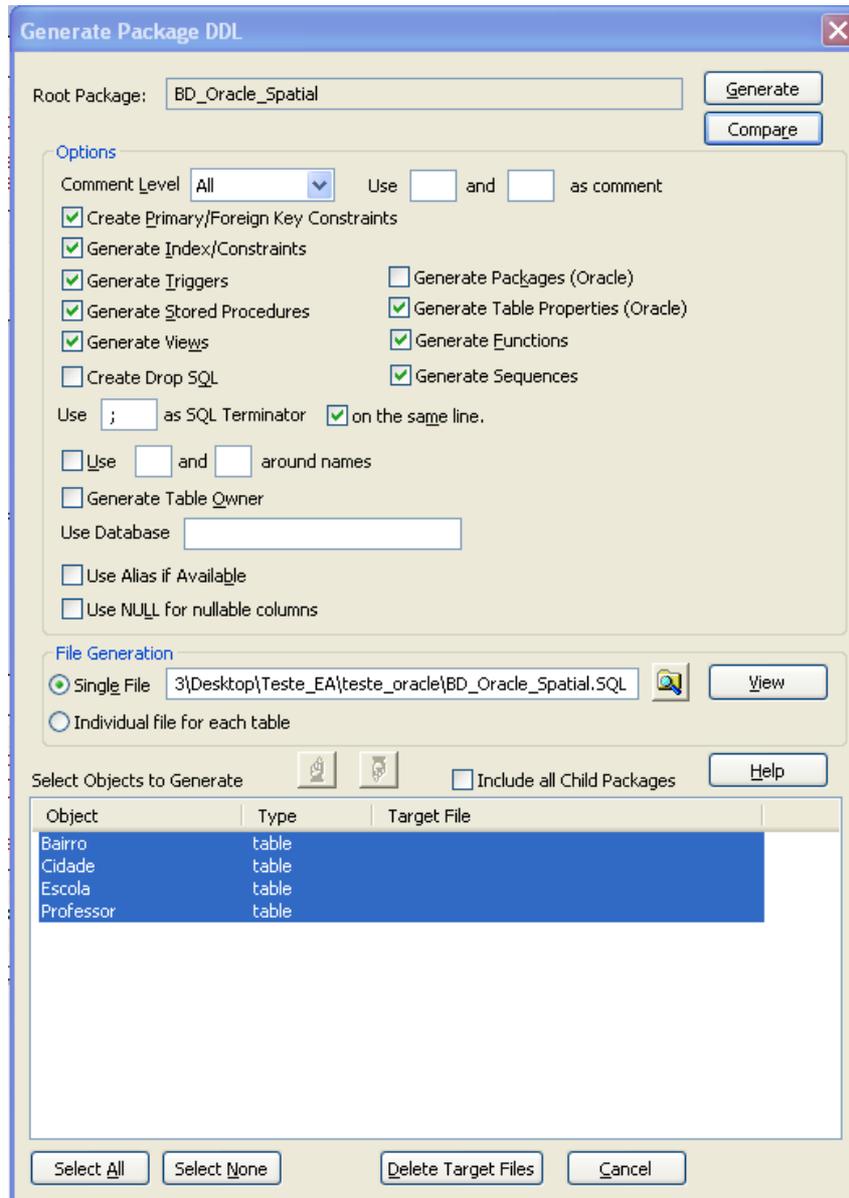


Figura 4.8 – Geração dos *scripts* de banco de dados geográficos para *Oracle Spatial*

Para gerar os índices espaciais, clique com o botão direito no pacote *Indices\_teste* no *Project Browser* e selecione *Code Engineering -> Generate Source Code*. Ao clicar em *Generate Source Code*, será aberta a janela mostrada na Figura 4.9. Clique em *Generate*. Em seguida, será aberta a caixa de diálogo mostrada na Figura 4.10 para indicar o caminho em que será gerado o arquivo com o *script*. Dê um nome para o arquivo com a extensão *.SQL*, clique em *Salvar* e depois em *Close* e feche a janela.

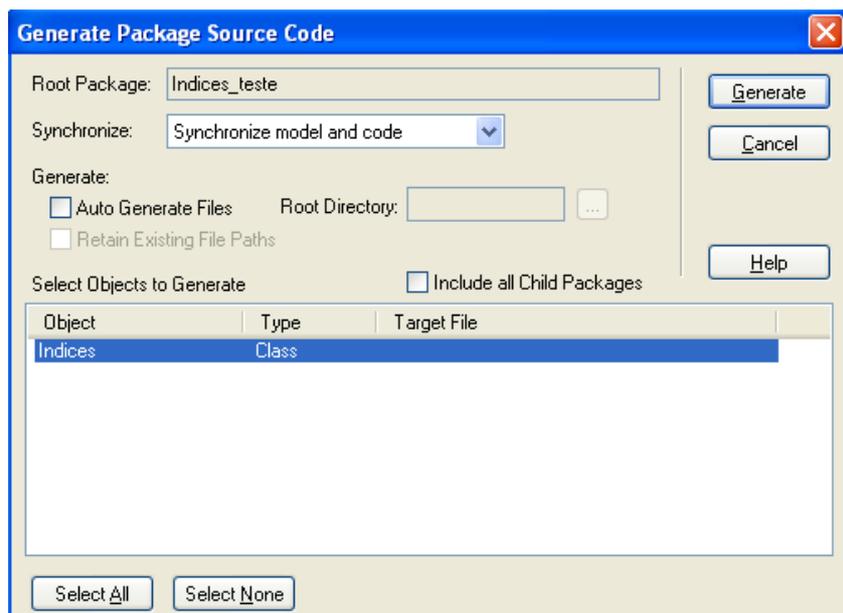


Figura 4.9 – Geração dos índices espaciais para o *Oracle Spatial*

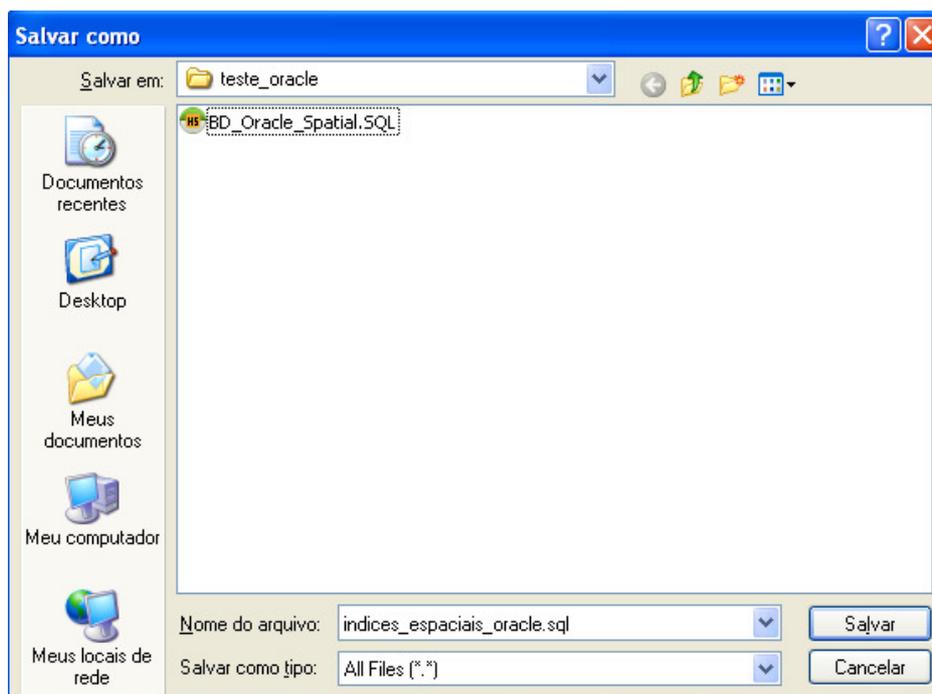


Figura 4.10 – Nomeando o arquivo com os índices espaciais

O quadro 1 mostra o *script* que foi gerado automaticamente referente ao arquivo `BD_Oracle_Spatial.sql` com as definições de tabelas e restrições de chave.

```

CREATE TABLE Bairro
(
    NOME          VARCHAR2(50),
    GEOMETRIA     MDSYS.SDO_GEOMETRY,
    ID_BAIRRO     NUMBER NOT NULL,
    ID_CIDADE     NUMBER NOT NULL
);

CREATE TABLE Cidade
(
    NOME          VARCHAR2(50),
    POPULACAO     NUMBER ,
    GEOMETRIA     MDSYS.SDO_GEOMETRY,
    ID_CIDADE     NUMBER NOT NULL
);

CREATE TABLE Escola
(
    NOME          VARCHAR2(50),
    ENDERECO      VARCHAR2(50),
    GEOMETRIA     MDSYS.SDO_GEOMETRY,
    ID_ESCOLA     NUMBER NOT NULL,
    ID_BAIRRO     NUMBER NOT NULL
);

CREATE TABLE Professor
(
    NOME          VARCHAR2(50),
    ENDERECO      VARCHAR2(50),
    ID_PROFESSOR  NUMBER NOT NULL,
    ID_ESCOLA     NUMBER NOT NULL
);

-- Create Primary Key Constraints
ALTER TABLE Bairro ADD CONSTRAINT PK_Bairro
    PRIMARY KEY (ID_BAIRRO)
    USING INDEX ;

ALTER TABLE Cidade ADD CONSTRAINT PK_Cidade
    PRIMARY KEY (ID_CIDADE)
    USING INDEX ;

ALTER TABLE Escola ADD CONSTRAINT PK_Escola
    PRIMARY KEY (ID_ESCOLA)
    USING INDEX ;

ALTER TABLE Professor ADD CONSTRAINT PK_Professor
    PRIMARY KEY (ID_PROFESSOR)
    USING INDEX ;

-- Create Foreign Key Constraints
ALTER TABLE Bairro ADD CONSTRAINT FK_Bairro_Cidade
    FOREIGN KEY (ID_CIDADE) REFERENCES Cidade (ID_CIDADE);

ALTER TABLE Escola ADD CONSTRAINT FK_Escola_Bairro
    FOREIGN KEY (ID_BAIRRO) REFERENCES Bairro (ID_BAIRRO);

ALTER TABLE Professor ADD CONSTRAINT FK_Professor_Escola
    FOREIGN KEY (ID_ESCOLA) REFERENCES Escola (ID_ESCOLA);

```

Quadro 1 – Script gerado para o arquivo BD\_Oracle\_Spatial.sql

O quadro 2 mostra as definições de índices espaciais que foram gerados no arquivo `indices_espaciais_oracle.sql`.

```
CREATE INDEX SX_Bairro ON Bairro(GEOMETRIA)
INDEXTYPE IS MDSYS.SPATIAL_INDEX
PARAMETERS ('SDO_INDX_DIMS=2 LAYER_GTYPE=POLYGON');

CREATE INDEX SX_Cidade ON Cidade(GEOMETRIA)
INDEXTYPE IS MDSYS.SPATIAL_INDEX
PARAMETERS ('SDO_INDX_DIMS=2 LAYER_GTYPE=POLYGON');

CREATE INDEX SX_Escola ON Escola(GEOMETRIA)
INDEXTYPE IS MDSYS.SPATIAL_INDEX
PARAMETERS ('SDO_INDX_DIMS=2 LAYER_GTYPE=POINT');
```

Quadro 2 – *Script* referente aos índices espaciais

## 4.2 PostGIS

Da mesma forma como foi feito para o *Oracle Spatial*, sugere-se também, neste caso, a criação de um novo pacote dentro do projeto atual para receber os modelos que serão provenientes da transformação. Para este exemplo foi dado o nome `Modelo_PostGIS` a esse novo pacote.

Os demais procedimentos também são semelhantes aos que foram feitos para o *Oracle Spatial*. Para efetuar a transformação para a plataforma *PostGIS*, clique com o botão direito no pacote do modelo conceitual e depois em *Transform Current Package*. Na janela que se abre (Figura 4.3), selecione as opções `BD_PostGIS` e `PostGIS_Indices_ColunaGeo`. Selecione o pacote `Modelo_PostGIS` para receber os elementos da transformação. Em seguida, clique em *Do Transform*.

Como resultado desse processo de transformação, são criados dois pacotes dentro do pacote `Modelo_PostGIS`, como pode ser visualizado no *Project Browser*, mostrado na Figura 4.11. O pacote denominado `BD_PostGIS` contém as definições de tabelas e restrições de chaves e o pacote `PostGIS_Indices_ColunaGeo_teste` será usado para gerar os índices espaciais e a coluna geométrica para as classes georreferenciadas.

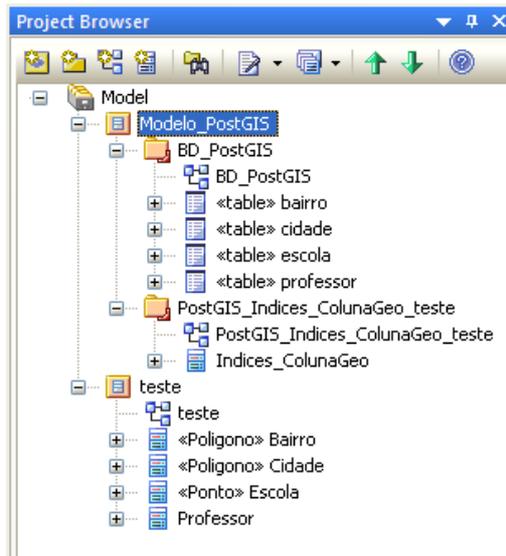


Figura 4.11 - *Project Browser* com os modelos resultantes da transformação para *PostGIS*

A Figura 4.12 ilustra o diagrama do pacote `BD_PostGIS` que foi gerado. Neste modelo são criadas as chaves primárias e estrangeiras. E a Figura 4.13 ilustra o diagrama do pacote `PostGIS_Indices_ColunaGeo_teste`, que será usado para a geração dos índices espaciais e da coluna geométrica para as classes georreferenciadas.

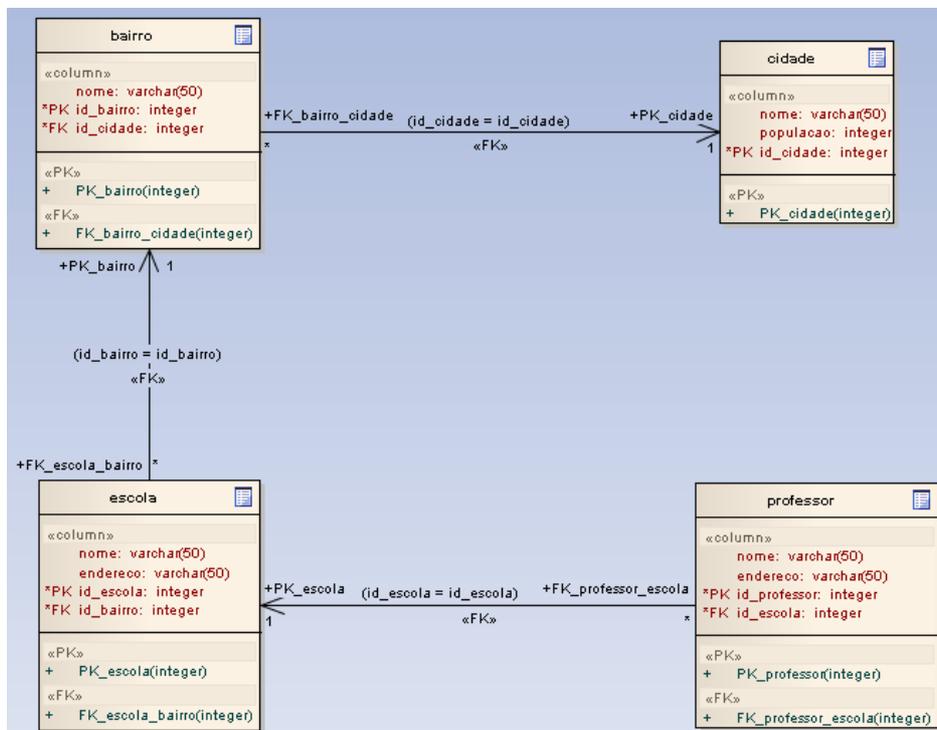


Figura 4.12 – Modelo resultante da transformação para *PostGIS*

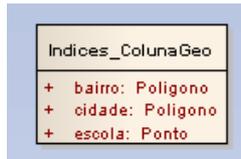


Figura 4.13 - Modelo para geração de índices espaciais e colunas geométricas no *PostGIS*

O procedimento para geração dos *scripts* de banco de dados é o mesmo do *Oracle Spatial*, o qual foi mostrado anteriormente, bastando mudar o nome dos arquivos gerados. O quadro 3 mostra os *scripts* gerados para definição de tabelas e chaves. E o quadro 4 mostra o *script* com os índices espaciais e a definição da coluna geométrica para cada classe georreferenciada do modelo.

```
-- Create Tables
CREATE TABLE bairro (
  nome varchar(50),
  id_bairro integer NOT NULL,
  id_cidade integer NOT NULL
);

CREATE TABLE cidade (
  nome varchar(50),
  populacao integer,
  id_cidade integer NOT NULL
);

CREATE TABLE escola (
  nome varchar(50),
  endereco varchar(50),
  id_escola integer NOT NULL,
  id_bairro integer NOT NULL
);

CREATE TABLE professor (
  nome varchar(50),
  endereco varchar(50),
  id_professor integer NOT NULL,
  id_escola integer NOT NULL
);

-- Create Primary Key Constraints
ALTER TABLE bairro ADD CONSTRAINT PK_bairro
  PRIMARY KEY (id_bairro);

ALTER TABLE cidade ADD CONSTRAINT PK_cidade
  PRIMARY KEY (id_cidade);

ALTER TABLE escola ADD CONSTRAINT PK_escola
  PRIMARY KEY (id_escola);

ALTER TABLE professor ADD CONSTRAINT PK_professor
```

```

PRIMARY KEY (id_professor);

-- Create Foreign Key Constraints
ALTER TABLE bairro ADD CONSTRAINT FK_bairro_cidade
FOREIGN KEY (id_cidade) REFERENCES cidade (id_cidade);

ALTER TABLE escola ADD CONSTRAINT FK_escola_bairro
FOREIGN KEY (id_bairro) REFERENCES bairro (id_bairro);

ALTER TABLE professor ADD CONSTRAINT FK_professor_escola
FOREIGN KEY (id_escola) REFERENCES escola (id_escola);

```

Quadro 3 - *Script* gerado para o arquivo BD\_PostGIS.sql

```

SELECT AddGeometryColumn('bairro', 'geometria', 29193, 'POLYGON', 2);
CREATE INDEX SX_bairro ON bairro USING GIST(geometria);

SELECT AddGeometryColumn('cidade', 'geometria', 29193, 'POLYGON', 2);
CREATE INDEX SX_cidade ON cidade USING GIST(geometria);

SELECT AddGeometryColumn('escola', 'geometria', 29193, 'POINT', 2);
CREATE INDEX SX_escola ON escola USING GIST(geometria);

```

Quadro 4 - *Script* referente aos índices espaciais e colunas geométricas para *PostGIS*

## Referências bibliográficas

BORGES, K. A. V.; DAVIS Jr., C. A.; LAENDER, A. H. F. OMT-G: An Object-Oriented Data Model for Geographic Applications. **GeoInformatica**, v.5, n.3, p. 221-260, set. 2001.

CASANOVA, M. et al. **Banco de Dados Geográficos**. Curitiba: MundoGeo, 2005. 506p.

ENTERPRISE ARCHITECT. Disponível em: <<http://www.sparxsystems.com>>. Acesso em: 01 jul. 2010.

ERIKSSON, H. et al. **UML 2 Toolkit**. Indianapolis: Wiley Publishing, 2004. 552p.

JENSEN, C.S et al. A consensus glossary of temporal database concepts. **ACM SIGMOD Record**, v. 23, n. 1, p. 52-64, mar. 1994.

NALON, F. R. **Adequação de um Perfil UML para Modelagem Conceitual de Bancos de Dados Geográficos aos Padrões ISO e OGC Usando MDA**. 2010. 75f. Dissertação (Mestrado em Ciência da Computação). Universidade Federal de Viçosa, Viçosa, 2010.

NALON, F. R. ; LISBOA FILHO, J. ; BORGES, K. A. V. ; ANDRADE, M. V. A. ; BRAGA, J. L. Using MDA and a UML Profile integrated with international standards to model geographic databases. In: SIMPÓSIO BRASILEIRO DE GEOINFORMÁTICA (GEOINFO), 2010, Campos do Jordão. **Anais...** São José dos Campos : MCT/INPE, 2010. p. 146-157

OBJECT MANAGEMENT GROUP. **Unified Modeling Language: Superstructure**, v. 2.1.2, OMG Document formal/2007-11-02 edition. Needham, MA, USA, 2007.

OBJECT MANAGEMENT GROUP. **MDA Guide**, v.1.0.1, OMG Document formal/2003-06-01 edition. Needham, MA, USA, 2003.

RUMBAUGH, J.; BLAHA, M.; PREMERLANI, W.; EDDY, F.; LORENSEN, W. **Object-oriented Modeling and Design**. Prentice-Hall, 1991.