

### 3 Serviços definidos pelo OGC e a TerraLib

#### 3.1. Introdução

O Open GIS Consortium (OGC ou OpenGIS) é um consórcio industrial com mais de 250 companhias, agências governamentais e universidades cujos membros trabalham num processo de consenso colaborativo para prover e aumentar interoperabilidade entre tecnologias envolvendo informações e localizações espaciais [20]. A motivação do OGC é um mundo onde todos disponibilizem informações e serviços geográficos em qualquer rede, aplicação ou plataforma. O OGC tem como missão prover uma interface de compartilhamento de informações espaciais e codificar especificações abertas e publicá-las para que fiquem disponíveis para uso global.

No Programa de Especificação do OGC, o Comitê Técnico e o Comitê de Planejamento trabalham num processo de consenso formal para disponibilizar as especificações OpenGIS.

O Programa de Interoperabilidade OGC é uma série de iniciativas de engenharia disponíveis para acelerar o desenvolvimento e a aceitação das especificações OpenGIS.

O OGC possui um documento chamado Modelo de Referência OGC (ORM). Este documento serve de embasamento técnico para o OGC. Ele provê uma arquitetura de *framework* para o desenvolvimento de trabalhos no OGC. Existe uma camada dedicada exclusivamente a Web Service: OpenGIS Web Service Framework.

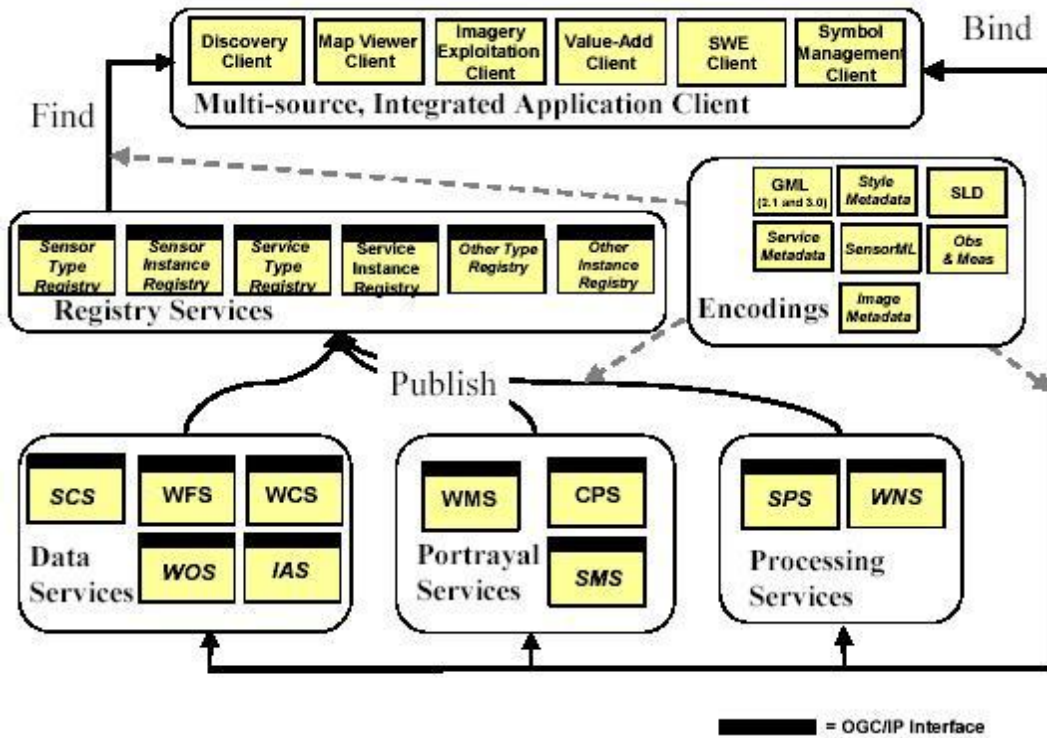


Figura 3 - OpenGIS Web Service Framework [22]

Os diferentes tipos de serviços (*Data Services*, *Portrayal Services* e *Processing Services*) implementam as operações que manipulam informações geográficas propriamente ditas. Estas operações retornam informações nos formatos representados na Figura 3 pela caixa *Encodings*. Os serviços podem ser publicados nos *Registry Services*. Assim, os clientes podem procurar por serviços nos *Registry Services* e, uma vez encontrados, estabelecer uma conexão com estes serviços.

A arquitetura que compreende os serviços e as aplicações OpenGIS pode ser melhor visualizada na Figura 4.

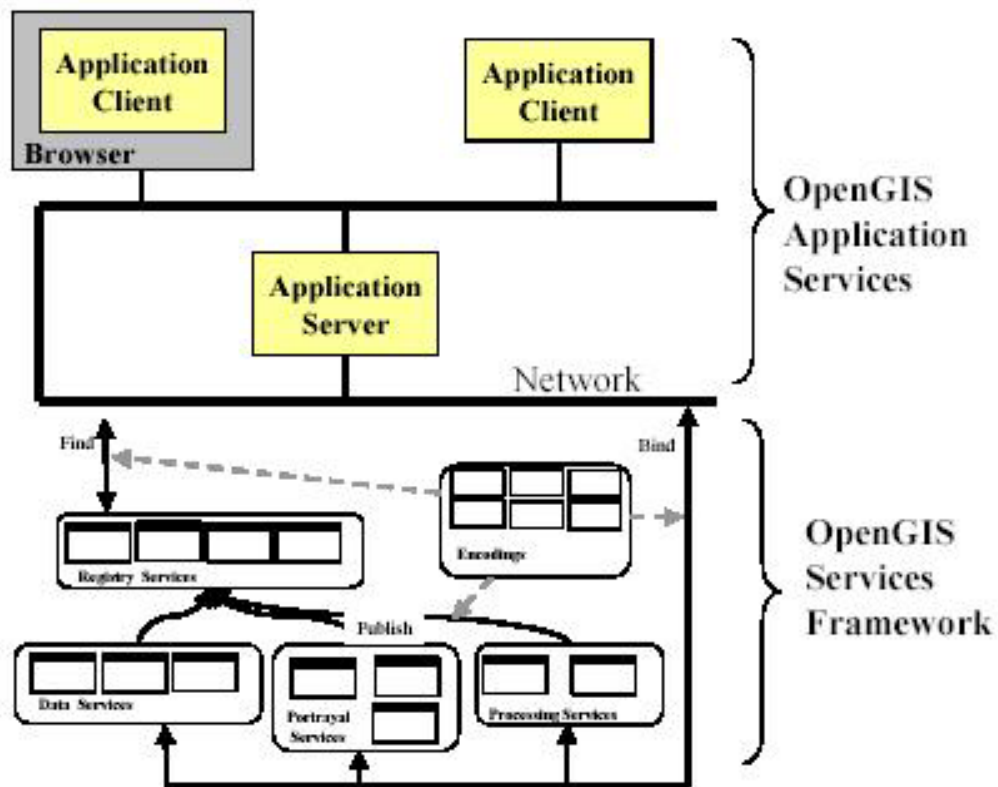


Figura 4 - Aplicações e Serviços OpenGIS [19]

### 3.2. Web Services OpenGIS

Como visto na Figura 3, o OGC possui vários Web Services definidos. Pode-se destacar dois deles como os mais difundidos e implementados:

- Web Map Service (WMS)
- Web Feature Service (WFS)

A seguir será feita uma breve discussão sobre cada um destes serviços. Logo após, serão apresentadas algumas decisões deste trabalho com relação a estes serviços.

#### 3.2.1. Web Map Service

O Web Map Service (WMS) é uma forma de produzir mapas a partir de dados georreferenciados e realizar consultas aos seus atributos. O mapa é a

representação visual das informações geoespaciais, e não a informação propriamente dita.

No WMS, a aplicação cliente não necessita implementar muitas operações. Um navegador Web padrão pode fazer o papel de aplicação cliente. Basta que ele envie os parâmetros corretamente para o servidor, via o protocolo HTTP. As informações trafegadas pela rede consistem basicamente de arquivos XML e imagens gráficas, que podem ser nos formatos GIF, JPEG, PNG entre outros.

### **3.2.2. Web Feature Service**

Assim como o WMS, o Web Feature Service (WFS) é uma forma de permitir a aplicações cliente visualizar informações geográficas na Internet. A grande diferença é a forma na qual estes dados geográficos são transportados do servidor para o cliente.

Enquanto que, no WMS, os dados geográficos são transportados como uma imagem gráfica, no WFS os dados são transportados na forma de GML (*Geography Markup Language*) [17]. Como o próprio nome indica, o GML é um markup XML para dados geográficos.

O cliente WFS já não é tão simples quanto o cliente WMS pois precisa decodificar GML e transformar estes documentos XML em informações geográficas [21].

### **3.2.3. Porque WMS?**

O serviço WMS foi o escolhido para este trabalho pelos seguintes motivos:

1. O WMS foi considerado como o primeiro passo rumo ao mundo OGC de padronização de serviços geográficos. Assim, seria iniciada a implementação da arquitetura base dos Web Services OGC, incluindo o mapeamento dos dados em TerraLib para o formato padrão OGC.

2. Um cliente WMS, em geral, é simples e relativamente fácil de se construir. Portanto, torna-se fácil e rápido construir uma aplicação para visualizar dados armazenados em TerraLib pelo WMS.
3. Existem muitos clientes WMS implementados atualmente e a maioria das aplicações de SIG, inclusive as *stand-alone*, possuem *plugins* que acessam serviços WMS, como por exemplo o ESRI ArcView.

#### **3.2.4. Modelo de Dados**

O modelo de dados adotados no WMS baseia-se nas seguintes noções.

Um *objeto geográfico* (*simple feature* no modelo de referência do OGC) é uma representação (computacional) de um fenômeno geográfico. O Rio São Francisco e o Rio Amazonas são exemplos de fenômenos geográficos. Tipicamente, mas não obrigatoriamente, um objeto geográfico possui um ou mais atributos que representam a localização espacial do fenômeno geográfico, em um dado sistema de referência espacial.

A localização espacial por sua vez é um objeto geométrico, como um ponto, linha, polígono ou outro objeto geométrico mais complexo.

Há inúmeros sistemas de referência espaciais, sendo os mais conhecidos baseados nas noções de latitude e longitude.

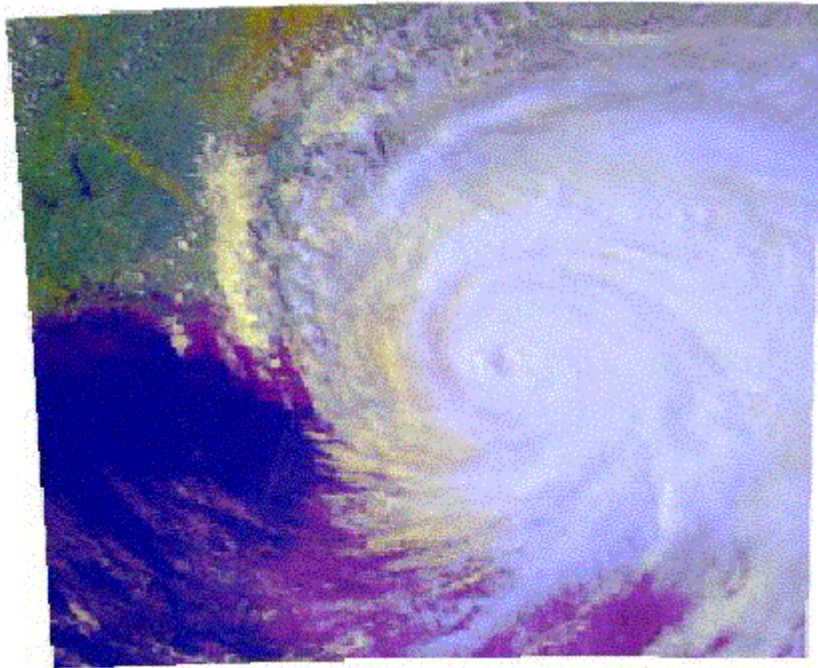
Um *layer* (*feature collection* no modelo de referência do OGC) é coleção de objetos geográficos. Por exemplo, a hidrografia de uma região pode ser representada em um *layer*, sendo cada rio, lago, etc... da região representado por um objeto geográfico pertencente ao *layer*.

Neste documento, a palavra *layer* será usada somente para se referir aos conceitos do WMS.

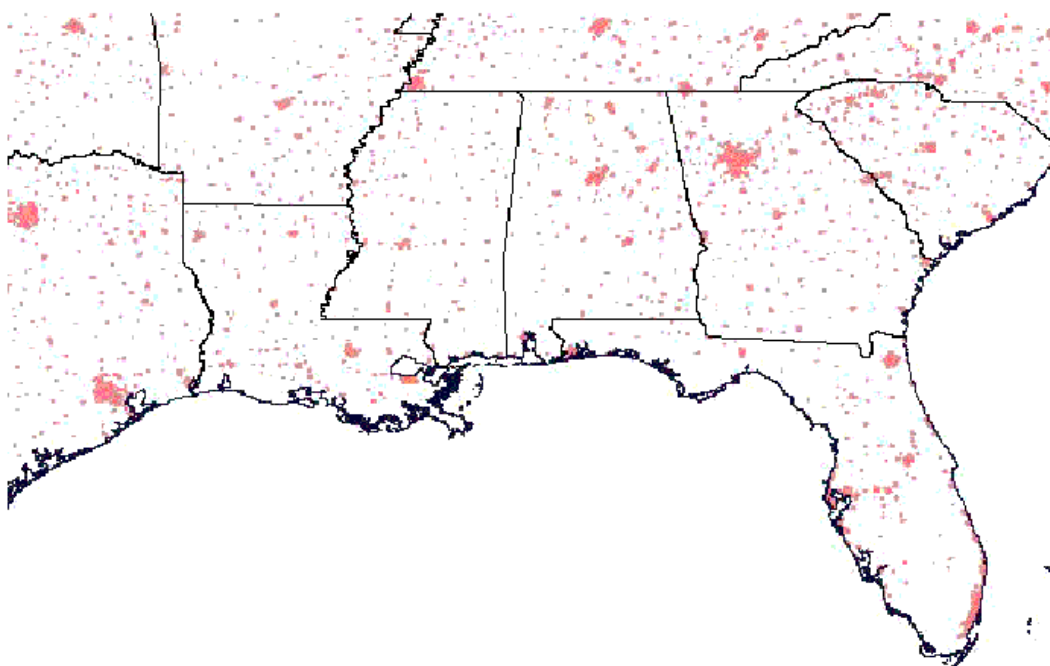
Um *mapa* é a representação gráfica de um ou mais *layers*. Note que os conceitos de *layer* e mapa são completamente distintos. Enquanto um *layer* refere-se a coleções de objetos geográficos, um mapa é uma representação gráfica de um conjunto de *layers*, tipicamente um arquivo de imagem.

Por exemplo, considere que o usuário deseja visualizar dados sobre a hidrografia brasileira e suponha que a base a ser consultada possui representações das bacias hidrográficas brasileiras em *layers* separados. Assim, o usuário pode requisitar a visualização somente das bacias que está interessado, escolhendo os *layers* apropriados. Esta requisição irá gerar um único mapa, ou seja uma única representação gráfica (um único arquivo GIF ou JPEG, por exemplo).

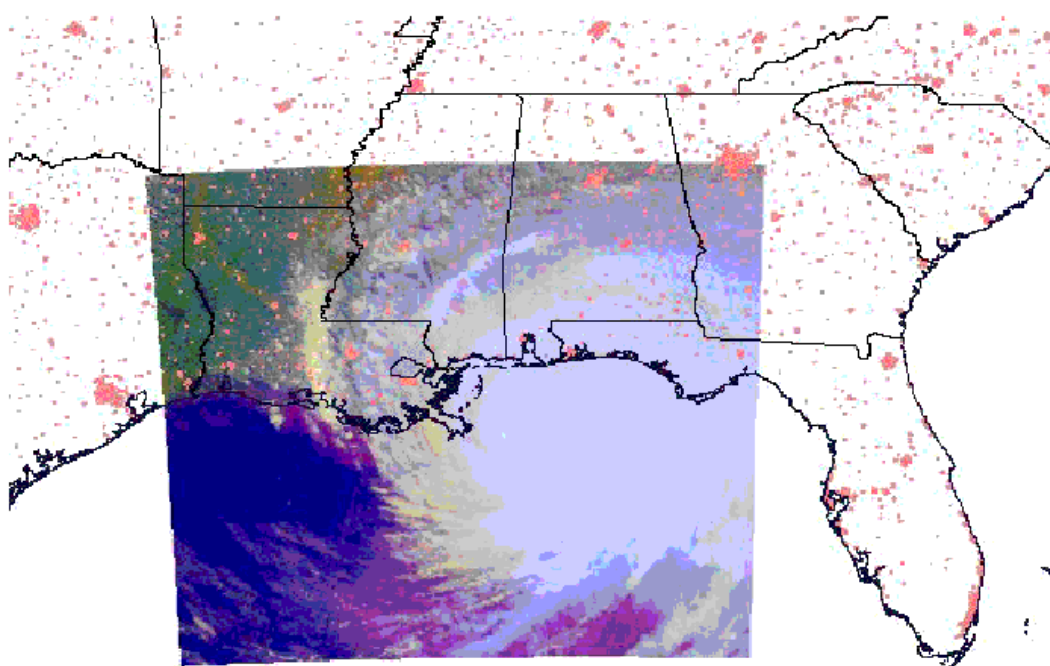
Na Figura 5, temos um mapa que representa o *layer* “Golfo do México”. Na Figura 6, está disposto o mapa que representa uma parte do *layer* “Estados Unidos Político”. Na Figura 7, está o mapa resultante da requisição de usuário que pede os *layers* “Golfo do México” e “Estados Unidos Político”. Quaisquer um dos três mapas mostrados (Figura 5, Figura 6 e Figura 7 ) pode ser resultado de uma requisição de usuário.



**Figura 5 - Imagem de Satélite do Golfo do México**



**Figura 6 - Estados Unidos Político**



**Figura 7 - Mapa de Composição de *layers***

### 3.3. A TerraLib

Analogamente às contribuições dadas pela GNU e pela comunidade de desenvolvimento do Linux, que disponibilizam não só as ferramentas mas também o código-fonte livremente, foi desenvolvida pelo DPI/INPE, TecGraf/PUC-Rio e pela FUNCATE a biblioteca **TerraLib**, que visa oferecer funcionalidades para a base de um software de GIS.

#### 3.3.1. Funcionalidade e Arquitetura

A TerraLib dá suporte a alguns componentes básicos para conjuntos de dados espaciais:

- tradução (ou importação e exportação de formatos) de dados: a biblioteca suporta importação e exportação para vários formatos existentes.
- representação de mapas e projeções cartográficas: a biblioteca suporta um núcleo básico de projeções cartográficas e um conjunto básico de técnicas de reprojeção e integração de dados espaciais.
- estruturas de dados geométricas; a biblioteca suporta estruturas vetoriais (pontos, linhas, polígonos), estruturas raster (matriciais) e estruturas relacionais (tabelas).
- algoritmos geométricos: a biblioteca oferece um núcleo básico de algoritmos de maior uso em geoprocessamento.
- modelo de dados sem impor grandes restrições de uso: o modelo de dados da TerraLib visa uma maior eficiência e uma adaptação aos algoritmos e às estruturas de dados. Este modelo pode ser concretizado em diferentes Sistemas de Gerenciamento de Banco de Dados (SGBD), como Oracle, MySQL e PostgreSQL, entre outros. Para cada um destes SGBDs existe um *driver* correspondente na TerraLib.
- consultas de acordo com três diferentes tipos de restrições: a biblioteca oferece consultas espaciais, temporais e convencionais (a atributos),



A TerraLib não serve somente para aplicações convencionais de SIG. Ela procura dar suporte a pesquisas inovadoras em algumas áreas de SIG, como:

- modelagem imprecisa: onde cada campo é acompanhado de informações de imprecisão espacial, que é propagada em operações algébricas no mapa.
- autômato celular: cujo uso em aplicações de SIG requer que cada célula possua seu próprio conjunto de atributos herdados que representam características físicas, ambientais, sociais, econômicas ou institucionais
- modelagem dinâmica: cuja implementação requer suporte para temporizadores e procedimentos interativos
- aplicações de Estatísticas Espaciais e de Modelos de Interações Espaciais: que requerem o uso de matrizes de proximidade espacial para suporte de relacionamento espacial entre elementos

O desenvolvimento da TerraLib não é feito em um único estilo de programação, pois usa dois paradigmas: orientado a objetos e procedimental. Para atender a esta exigência, a linguagem de programação escolhida foi C++.

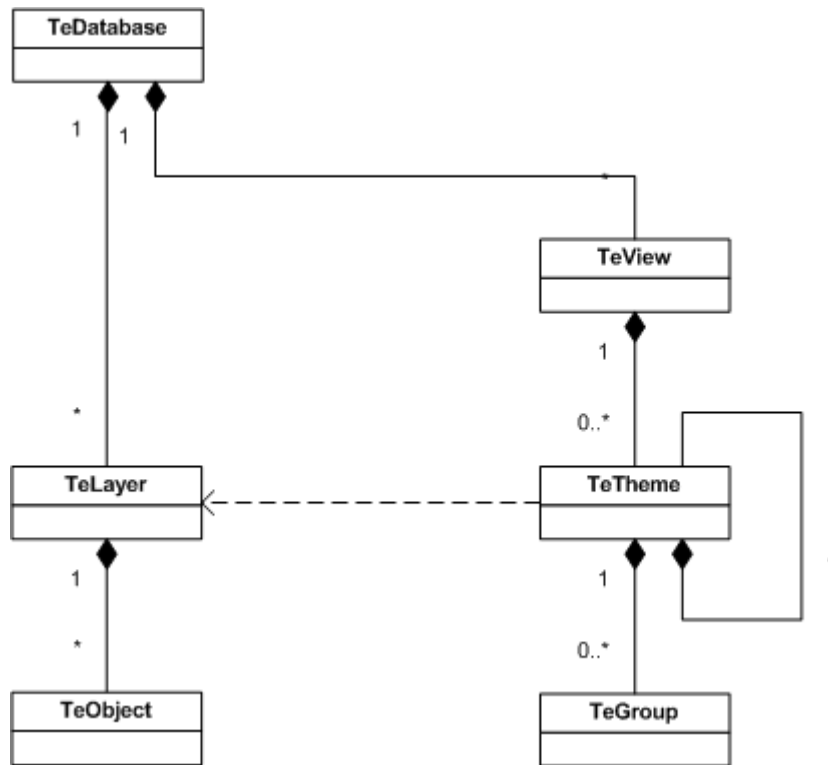
A opção por C++ facilita ainda o reuso de código. De fato, algoritmos implementados para o software **SPRING** (desenvolvido pelo DPI/INPE) foram adaptados para serem utilizados na TerraLib.

Por fim, a TerraLib foi desenvolvida em camadas:

- alto nível: oferece classes de dados abstratas para manipular e representar informações geográficas, baseadas no paradigma de campo/objeto (*Feature, Network, Surface, CellArray*)
- nível intermediário: oferece suporte a estruturas de dados geométricas, formato de dados, representação de mapas e algoritmos
- baixo nível: suporta indexação de estruturas espaciais para manipulação eficiente de grandes conjuntos de dados

**3.3.2.  
Modelo de Dados**

A Figura 8 mostra o modelo de dados da TerraLib.



**Figura 8 - Modelo de Dados da TerraLib**

Classe	Descrição do elemento da classe
TeDatabase	um conjunto de TeLayers
TeLayer	um conjunto de objetos espaço-temporais que compartilham a mesma projeção geográfica e o mesmo conjunto de atributos em um período de tempo.
TeObject	um objeto representando um fenômeno geográfico, cujo identificador é único.  Um objeto espaço-temporal tem uma representação espacial e uma representação convencional. Ambas podem, ou não, mudar de acordo com o passar do tempo.  A representação convencional é dependente da aplicação e está fora do escopo da TerraLib.
TeView	uma árvore de temas (TeTheme) que compartilham a mesma projeção geográfica
TeTheme	um subconjunto virtual de um <i>layer</i> (TeLayer).
TeGroup	um subconjunto virtual da <i>view</i> de um <i>layer</i> .

**Tabela 2 - Componentes do modelo de dados da TerraLib**

O mapeamento do modelo de dados da TerraLib para o modelo do WMS é bastante simples:

- cada “TeView” do modelo TerraLib corresponde a um *layer* contendo os TeThemes da TeView;
- cada “TeTheme” do modelo TerraLib, corresponde a um *layer*.

O *layer* correspondente a uma “TeView” herda desta os seguintes parâmetros:

- Nome (mapeado para <TITLE>)
- *Bounding Box* (mapeado para <BBOX>)
- Projeção (mapeado para <SRS>)

O mapeamento da projeção para o <SRS> não é uma tarefa imediata. Na implementação desta dissertação, todas as “TeView” requisitadas serão passadas para a projeção EPSG:4326 (LatLong), segundo a tabela do EPSG..

Cada *layer* correspondente a um “TeTheme” herda deste os seguintes parâmetros (ver Seção 3.1.3.1.2.1):

- Nome (mapeado para <TITLE>)
- ID (mapeado para <NAME>)
- *Bounding Box* (mapeado para <BBOX>)
- Escala mínima e máxima, quando existem (mapeada para <ScaleHint>)
- A materialização dos objetos geográficos pertencentes ao “TeTheme”.

### 3.4. O Framework Tdk

Como descrito na seção anterior, a TerraLib tem seu foco voltado para os algoritmos geométricos e para uma abstração dos dados. Entretanto, ela disponibiliza interfaces e serviços em uma nível relativamente baixo, dificultando o seu uso por programadores menos treinados. Mesmo para se criar uma aplicação simples, é necessário conhecer muitos detalhes da estrutura dos dados da biblioteca [26], além de contar com poucos componentes que auxiliam a construção da interface com o usuário.

Visando facilitar o uso da TerraLib e atender a funcionalidades comuns às aplicações de visualização de dados e alimentação de informações desenvolvidas sobre a biblioteca, o TecGraf/PUC-Rio propôs um *framework* chamado Tdk. Este *framework* funciona fornecendo uma interface mais simplificada, de nível de abstração mais alto e com diversos conceitos relacionados à interface com o usuário.

O Tdk é constituído por duas APIs, onde cada uma forma um subsistema distinto: API de Serviços e de API de Componentes.

### **3.4.1. API de Serviços**

A API de Serviços é responsável por funcionalidades como persistência, apresentação e operações espaciais. Esta API provê métodos bem definidos que encapsulam a complexidade de uma implementação direta sobre a TerraLib. As assinaturas dos métodos escondem do programador conceitos particulares da TerraLib, na medida do possível, oferecendo tipos primitivos ao invés das classes mais complexas, que constituem o modelo de dados da TerraLib.

#### **3.4.1.1. Serviço de Persistência**

Este serviço é responsável por oferecer uma interface de alto nível para a base de dados.

Suas principais funcionalidades são:

- encapsular lógicas complexas de inicialização do modelo TerraLib de Banco de Dados a diversas implementações de DBMS
- carregar, alterar, remover e salvar as entidades através de métodos especializados
- criar geometrias através de métodos especializados
- proporcionar uma forma simples de executar consultas diretamente no banco de dados
- oferecer um “fábrica” de SQL para encapsular e simplificar a criação dinâmica de consultas tornando o processo mais automático que o convencional na TerraLib.

#### **3.4.1.2. Serviço Gráfico**

O serviço gráfico do Tdk visa simplificar o manuseio das informações visuais dos dados. Para isto foi introduzido o conceito de *canvas* ao modelo.

Um *canvas* é uma estrutura de dados que permite ao desenvolvedor manipular dados visuais e exportar estes dados para a tela, para um arquivo ou para a impressora. Funções de redimensionamento (*zoom*, rotação) são implementadas pelo *canvas*.

A consulta espacial também pode ser feita através deste *canvas*. Se o usuário clicar com o *mouse* em um ponto da tela, as coordenadas deste ponto serão passadas para o *canvas*, para que este retorne os atributos dos objetos geográficos que estão dispostos naquele ponto.

Este serviço gráfico tem o objetivo de não ficar preso à biblioteca de visualização. Suas funcionalidades estão disponíveis quer o desenvolvedor use a biblioteca **Qt** ou a biblioteca **Iup** [27].

#### **3.4.1.3. Serviço de Cálculo**

Este serviço disponibiliza os algoritmos espaciais da TerraLib. Por exemplo, dado dois pontos, qual é a distância em quilômetros entre eles.

#### **3.4.1.4. Serviço de Apresentação**

O serviço de apresentação serve de complemento para o serviço gráfico. É o serviço que disponibiliza os componentes visuais para a tela do usuário como, por exemplo, Caixas de Diálogos, Legendas e o *Canvas* visual. Este *canvas* visual é onde os dados geográficos propriamente ditos são exibidos para o usuário da aplicação.

Análogo ao serviço Gráfico, este serviço tem como objetivo ser independente de bibliotecas de visualização.

### **3.4.2. API de Componentes**

Esta API define todos os componentes que serão utilizados para realizar as tarefas oferecidas pela API de Serviços.

Dentre as principais funcionalidades oferecidas pelos componentes implementados neste subsistema pode-se destacar:

- Componentes para encapsular as funcionalidades de legenda gráfica do mapa
- Componentes de auxílio a visualização de mapas
- Uma biblioteca para antever e configurar os resultados de impressão de mapas
- Uma biblioteca de estilos para exibição dos objetos espaciais
- Recursos de construção de gráficos estatísticos
- Diálogos diversos para interação com usuário