



# OGC Web Services

Gilberto Ribeiro de Queiroz ([gribeiro@dpi.inpe.br](mailto:gribeiro@dpi.inpe.br))

Emerson Xavier ([emerson@dpi.inpe.br](mailto:emerson@dpi.inpe.br))

Vanessa Oliveira ([vanessa@dpi.inpe.br](mailto:vanessa@dpi.inpe.br))

Sérgio A. B. Cruz ([scruez@dpi.inpe.br](mailto:scruez@dpi.inpe.br))

Disponível em: <http://www.dpi.inpe.br/twsg/apresentacoes>

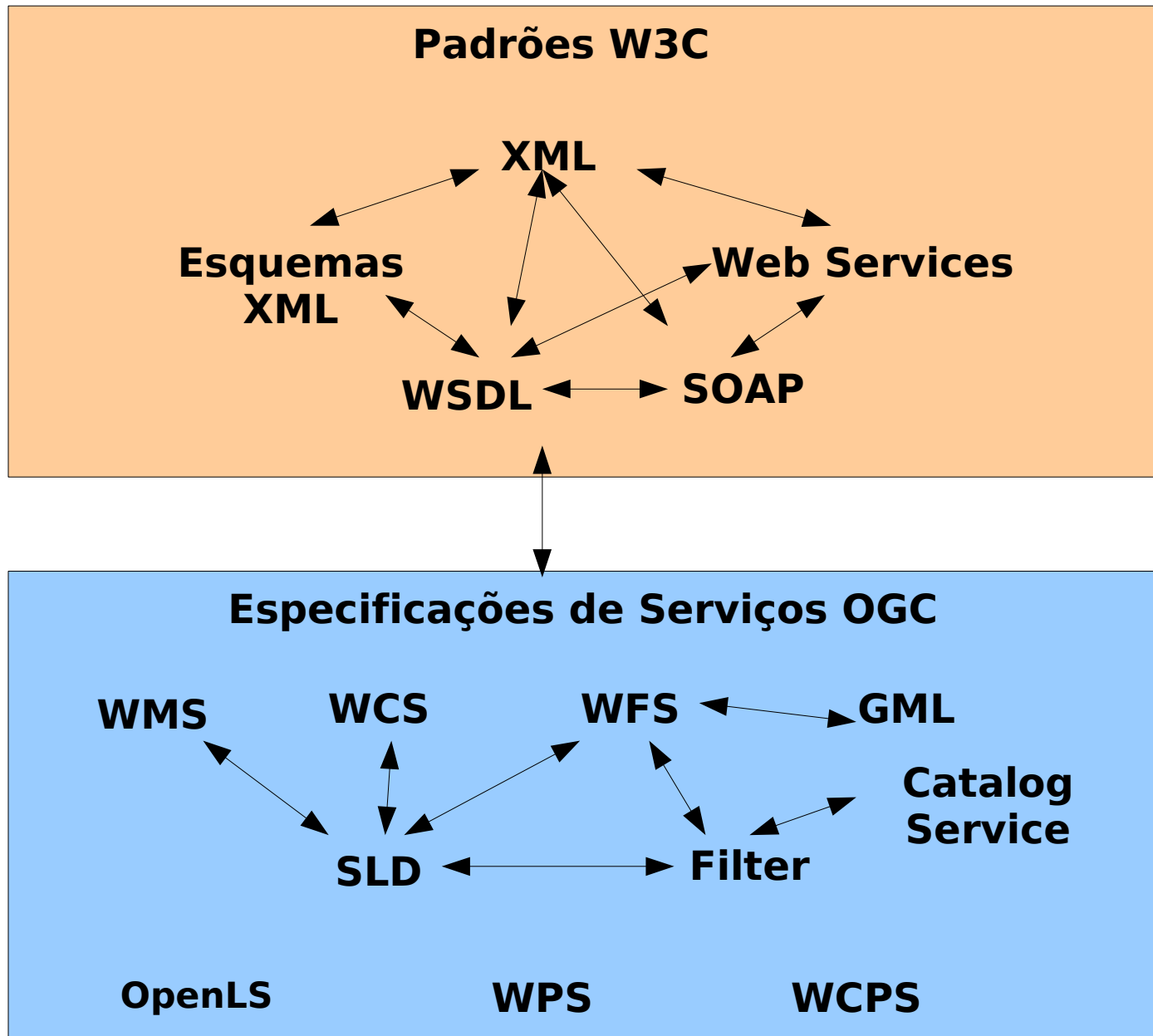


# Sumário



- XML e Esquemas
- Web Services
- WMS
- GML
- Filter
- WFS
- WCS
- SLD
- OLS
- WPS e WCPS
- CSW

# Motivação





**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA**  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# XML e Esquemas



# XML: Sumário



- Introdução
- Estrutura XML
- Namespaces
- Esquemas XML



# XML: Sumário



- Introdução ←
- Estrutura XML
- Namespaces
- Esquemas XML



# XML: Introdução



- Considerações iniciais
  - Web Services, por Wikipedia
    - “Os Web services são componentes que permitem às aplicações enviar e receber dados em formato *XML*. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, o formato *XML*.”
    - “As bases para a construção de um Web service são os padrões *XML* e *SOAP*”
  - Web Services, por W3C
    - “A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically *WSDL*). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an *XML* serialization in conjunction with other Web-related standards.”



# XML: Introdução



- Definição do XML
  - Linguagem p/ codificação de dados em texto puro
  - Derivada do SGML (Standard Generalized Markup Language), norma ISO 8879:1986
  - Sigla:
    - eXtensible Markup Language
  - Objetivo:
    - Facilitar o intercâmbio de dados através da Internet
  - Padronizada pelo W3C (World Wide Web Consortium)
    - Recomendação em sua 4ª edição (16 Ago 2006)
    - <http://www.w3.org/TR/xml>





# XML: Introdução



- Histórico do XML
  - 1993: primeiros trabalhos sobre adaptação das técnicas SGML à Web (Sperberg)
  - 1995: SGML adicionado às listas de trabalho do W3C (Connolly)
  - 1996: início dos trabalhos do XML Working Group (W3C), originalmente denominado SGML Editorial Review Board, presidido por Jon Bosak (Sun), com o apoio da Microsoft
  - 1998: versão 1.0 da recomendação W3C



# XML: Sumário



- Introdução
- Estrutura XML ←
- Namespaces
- Esquemas XML



# XML: Estrutura

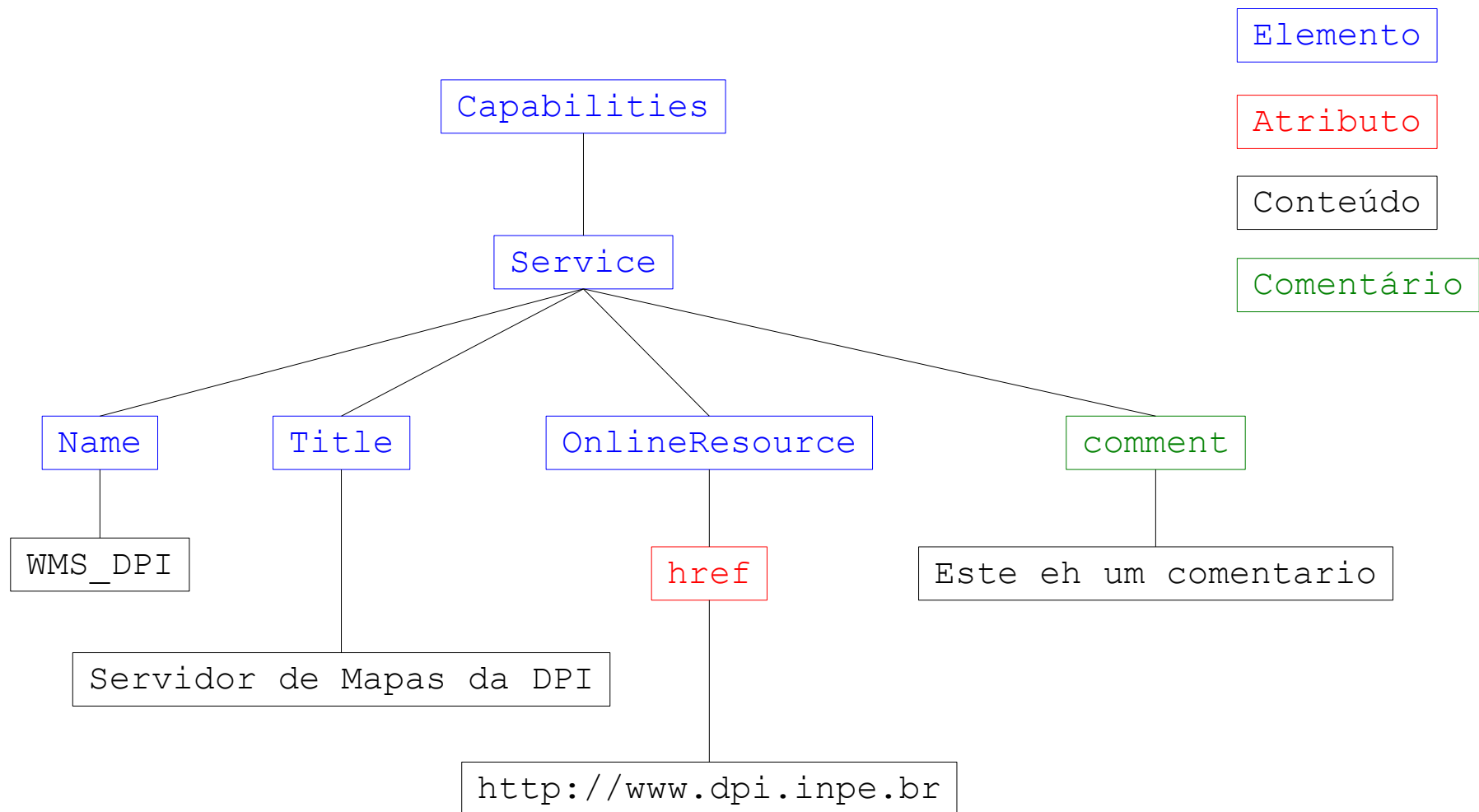


- Características
  - Separação conteúdo/formatação
  - Simplicidade e legibilidade
  - Possibilidade de criação de *tags* sem limitação
- Documento XML
  - Um dado pode ser considerado um documento XML se é **bem-formado**, de acordo com o que prescreve a norma
  - Adicionalmente, pode ser **válido** também
    - Essa validação ocorre através de um determinado mecanismo, normalmente DTD ou Esquemas XML

- Documento XML – componentes principais
  - Elemento: unidade fundamental, delimitado por *tags*
  - Atributo: usado p/ associar KVPs com elementos
  - Conteúdo: qq string que não contenha '<', ']]>', '&'
  - Comentário: informação não processada

```
<?xml version="1.0" encoding="UTF-8" ?>
<Capabilities>
  <Service>
    <Name>WMS_DPI</Name>
    <Title>Servidor de Mapas da DPI</Title>
    <OnlineResource href="http://www.dpi.inpe.br"/>
    <!-- Este eh um comentario -->
  </Service>
</Capabilities>
```

- Documento XML – componentes principais





# XML: Sumário



- Introdução
- Estrutura XML
- Namespaces ←
- Esquemas XML

- Considerações iniciais

```
<?xml version="1.0" encoding="UTF-8" ?>
<explo_Namespace>
  <!-- Situação comum no HTML -->
  <!-- tr = table row -->
  <!-- td = table data -->
    <table>
      <tr>
        <td>Banana</td>
        <td>Laranja</td>
      </tr>
    </table>

  <!-- Situação possível -->
  <!-- tr = tema recorrente -->
  <!-- td = tema dirigido -->
    <table>
      <tr>
        <td>XML</td>
        <td>C++</td>
      </tr>
    </table>

</explo_Namespace>
```

- Solução? Uso de Namespaces
  - Namespace default – os filhos herdam

```
<?xml version="1.0" encoding="UTF-8" ?>
<explo_Namespace>

    <table xmlns="http://www.w3.org/TR/html4/">
        <tr>
            <td>Banana</td>
            <td>Laranja</td>
        </tr>
    </table>

    <table xmlns="http://www.explo.org/temas/">
        <tr>
            <td>XML</td>
            <td>C++</td>
        </tr>
    </table>

</explo_Namespace>
```





# XML: Namespaces



- Definição

- Um namespace XML é identificado por uma referência URI, podendo ser aplicada à elementos e atributos
- Spec:
  - Recomendação em sua 2ª edição (16 Ago 2006)
  - <http://www.w3.org/TR/xml-names>
- Objetivo:
  - Prover elementos e atributos únicos num documento XML, eliminando uma possível ambigüidade
- URI – Universal Resource Identifier
  - URL – Universal Resource Locator (<http://www.inpe.br/>)
  - URN – Universal Resource Name (urn:oasis:dtd:xml:4.1.2)



# XML: Namespaces



- Conceitos
  - Nome local:
    - É o nome do elemento ou atributo sem o namespace
  - Nome do Namespace:
    - A URI que o identifica
  - Nome expandido:
    - É composto do nome local mais o nome do namespace
  - Nome qualificado
    - É um nome sujeito à interpretação do namespace
  - Prefixo:
    - É uma identificação reduzida de um namespace

## • Exemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:WFS_Capabilities
  xmlns:ows="http://www.opengis.net/ows"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  ...
  <wfs:FeatureTypeList>
    <ows:WGS84BoundingBox>
      <gml:LowerCorner>-180 -90<
/gml:LowerCorner>
      <gml:UpperCorner>180 90</gml:UpperCorner>
    </ows:WGS84BoundingBox>
  </wfs:FeatureTypeList>
  ...
</wfs:WFS_Capabilities>
```

<b>Local</b>	FeatureTypeList
<b>Namespace</b>	http://www.opengis.net/wfs
<b>Expandido</b>	http://www.opengis.net/wfs:FeatureTypeList
<b>Qualificado</b>	wfs:FeatureTypeList
<b>Prefixo</b>	wfs



# XML: Namespaces



- Outras informações
  - Quando um namespace (NS) é definido na Tag inicial de um elemento, todos os elementos filhos com o mesmo prefixo estão associados ao mesmo NS



# XML: Sumário



- Introdução
- Estrutura XML
- Namespaces
- Esquemas XML ←

- Considerações iniciais
  - Documento bem-formado vs válido

```
<Estado>  
  <Brasil/>  
  <Estado>MG</Estado>  
</Estado>
```

```
<Brasil>  
  <Estado>SP</Estado>  
  <Estado>MG</Estado>  
</Brasil>
```

- Bem-formado vs Válido
  - Bem-formado
    - O documento segue a sintaxe do XML (tags aninhadas, atributos únicos)
  - Válido
    - Se possui a declaração de uma estrutura que defina os tipos de elementos e atributos que estão no documento, ou seja, se obedece algumas normas descritas na sua gramática

- DTD vs Esquema
  - Formas de gramática mais conhecidas
  - DTD (Document Type Definition)
    - Primeira tentativa de criar uma estrutura p/ os XMLs
    - Consta da Spec original
  - Esquema (XML Schema)
    - Contornar as limitações do DTD na definição dos XMLs
    - Tem Specs próprias
- Quem usa esquemas para validar documentos?
  - W3C (SOAP, SOAP-Envelope)
  - OGC (WxS | x <- [M, C, F, P ...], GML)



# XML: Esquemas



- Propósito
  - Especificar a estrutura de documentos
    - “Uma linha possui pontos.”
  - Especificar os tipos de dados de cada elemento ou atributo
    - “Um ponto possui três coordenadas (x, y, z) como números reais.”
- Vantagens em relação ao DTD
  - É escrito em XML
  - Grande variedade de tipos de dados, inclusive criando subsets
    - “A coordenada x varia de 200.000 até 700.000.”
  - Uso de chaves





# XML: Esquemas



- Specs
  - XML Schema Part 0: Primer 2nd Ed
    - W3C Recommendation 28 October 2004
    - Não é normativo, mas fornece uma descrição simplificada
    - <http://www.w3.org/TR/xmlschema-0/>
  - XML Schema Part 1: Structures 2nd Ed
    - W3C Recommendation 28 October 2004
    - <http://www.w3.org/TR/xmlschema-1/>
  - XML Schema Part 2: Datatypes 2nd Ed
    - W3C Recommendation 28 October 2004
    - <http://www.w3.org/TR/xmlschema-2/>

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/wms"
  xmlns:wms="http://www.opengis.net/wms"
  xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <element name="Service">
    <complexType>
      <sequence>
        <element name="Name" type="string"/>
        <element name="KeywordList" minOccurs="0">
          <complexType>
            <sequence>
              <element name="Keyword" type="string" minOccurs="0"
                maxOccurs="unbounded"/>
            </sequence>
          </complexType>
        </element>
        <element name="OnlineResource">
          <complexType>
            <attribute name="href" type="anyURI"/>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>
</schema>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<Service xmlns="http://www.opengis.net/wms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="explo4.xsd">
  <Name>WMS</Name>
  <KeywordList>
    <Keyword>bird</Keyword>
    <Keyword>roadrunner</Keyword>
  </KeywordList>
  <OnlineResource href="http://hostname"/>
</Service>

```

- Analisando por partes
  - Cabeçalho

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/wms"
  xmlns:wms="http://www.opengis.net/wms"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
```

- Service

```
<element name="Service">
  <complexType>
    <sequence>
      <element name="Name" type="string"/>
      <element ref="wms:KeywordList" minOccurs="0"/>
      <element ref="wms:OnlineResource"/>
    </sequence>
  </complexType>
</element>
```

## Service

Name [1] : string  
KeywordList [0..1]  
OnlineResource [1] : URI

- Analisando por partes
  - KeywordList

```
<element name="KeywordList">  
  <complexType>  
    <sequence>  
      <element ref="wms:Keyword"  
        minOccurs="0" maxOccurs="unbounded"/>  
    </sequence>  
  </complexType>  
</element>
```

## Service

Name [1] : string  
KeywordList [0..1]  
OnlineResource [1] : URI

- Keyword

```
<element name="Keyword" type="string" />
```

## KeywordList

Keyword [0..n]: string



# XML: Esquemas



- Analisando por partes
  - OnlineResource

```
<element name="OnlineResource">  
  <complexType>  
    <attribute name="href" type="anyURI"/>  
  </complexType>  
</element>
```

## Service

Name [1] : string  
KeywordList [0..1]  
OnlineResource [1] : URI

## KeywordList

Keyword [0..n]: string

- Também podem ser criados tipos definidos

```
<element name="OnlineResource" type="wms:OnlineResourceType"/>  
  
<complexType name="OnlineResourceType">  
  <attribute name="href" type="anyURI"/>  
</complexType>
```

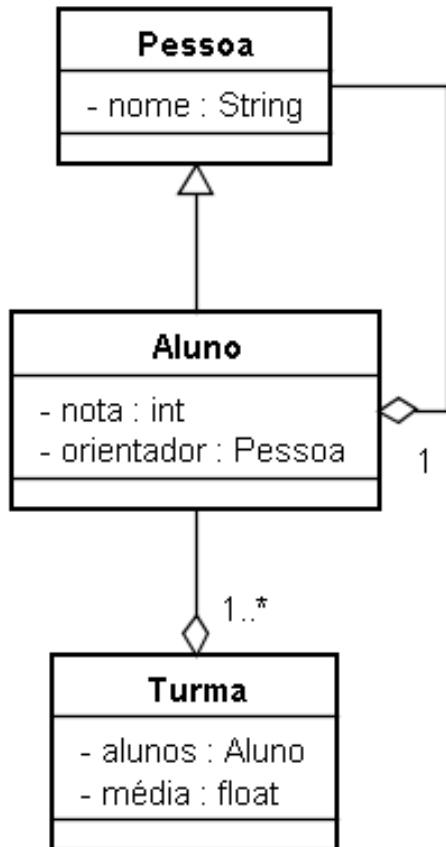


# XML: Esquemas



- Possibilidades
  - Agregação
    - “Uma quadra contém n lotes”
  - Herança
    - “SJ Campos é uma cidade”
  - Restrição de tipo
    - “Um telefone é da forma xx-yyyy-zzzz”
  - Poucas limitações
    - Acredito que não seja possível validar um CPF (será?)

## • Exemplos



```

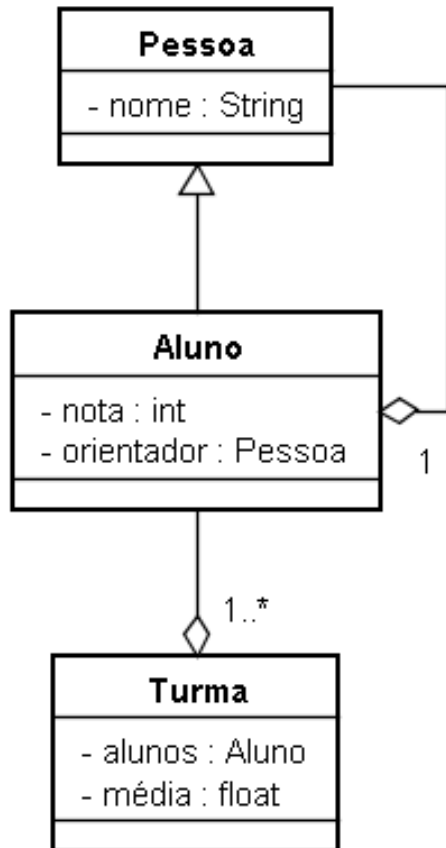
<xs:schema
  xmlns="http://www.terralib.org"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.terralib.org"
  elementFormDefault="qualified">

  <xs:complexType name="Pessoa">
    <xs:sequence>
      <xs:element name="nome" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="Aluno">
    <xs:complexContent>
      <xs:extension base="Pessoa">
        <xs:sequence>
          <xs:element name="nota" type="xs:integer"/>
          <xs:element name="orientador" type="Pessoa"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  ...
  
```

- Exemplos

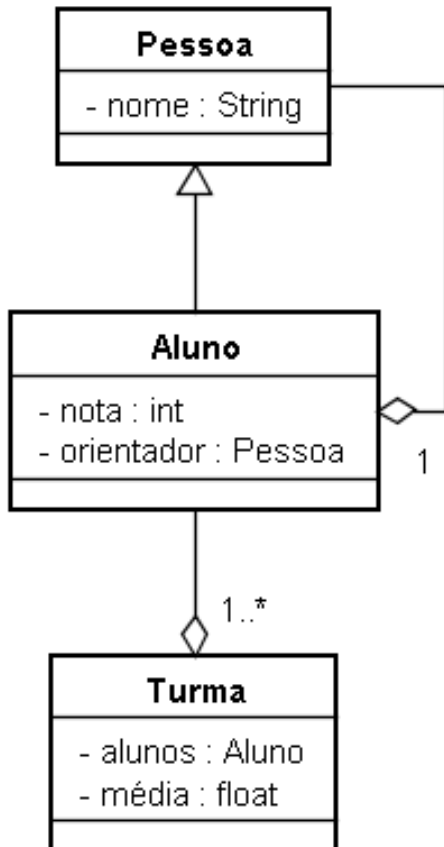


```
...
<xs:element name="Turma">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="aluno" type="Aluno"
        maxOccurs="unbounded"/>
      <xs:element name="media" type="xs:float"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```



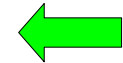
## • Exemplos



```

<aluno>
  <nome>Emerson</nome>
  <nota>8</nota>
  <orientador>
    <nome>Prof Miguel</nome>
  </orientador>
</aluno>

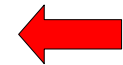
```



```

<aluno>
  <nota>8.5</nota>
  <nome>José</nome>
  <orientador>
    <nome>Prof Xavier</nome>
  </orientador>
</aluno>

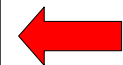
```



```

<aluno>
  <nome>Fulano</nome>
  <nota>8</nota>
  <orientador>Cicrano</orientador>
</aluno>

```



- DOM (Document Object Model)
  - Interface baseada em árvore: mapeia um documento XML em uma árvore
  - Oferece métodos p/ navegar nessa árvore
  - Interface especificada pelo W3C (3 níveis e specs)
    - <http://www.w3.org/DOM/DOMTR> (umas 10 specs)
- SAX (Simple API for XML)
  - API baseada em eventos: invoca certos métodos à medida que o documento vai sendo analisado
  - A aplicação define *handlers* p/ tratar cada tipo de evento
  - Desenvolvida por um grupo independente (XML-DEV)
    - <http://sax.sourceforge.net/>

- XML Pull

- API baseada em análise incremental: apenas o que a aplicação pede é analisado
- Só é executado enquanto constrói um elemento pedido pela aplicação
- Desenvolvido por um grupo independente
  - <http://www.xmlpull.org/>

- Comparação

API	Vantagem	Desvantagem
<b>DOM</b>	Acesso rápido aos dados	↑ memória, ↑ tempo
<b>SAX</b>	↓ memória, ↓ tempo	Difícil acesso a dados isolados Implementação complicada
<b>XML Pull</b>	↓ memória, ↓ tempo	Não suporta validação Pouco difundido



# XML: Fontes / Leituras adicionais



- Wikipedia
  - <http://en.wikipedia.org/wiki/XML>
- W3C
  - <http://www.w3.org/XML/>
  - <http://www.w3schools.com/>
- Hyperlink: Internet: XML
  - Vários links sobre o assunto
  - <http://www.mhavila.com.br/link/internet/xml.html>
- Material Prof Cláudio Baptista
  - <http://www.dsc.ufcg.edu.br/~baptista/cursos/BDInterr>



# XML: Fontes / Leituras adicionais



- Specs
  - Extensible Markup Language (XML) 1.0 (4th Edition)
    - W3C Recommendation 16 August 2006
    - <http://www.w3.org/TR/2006/REC-xml-20060816/>
  - Namespaces in XML 1.0 (2nd Edition)
    - W3C Recommendation 16 August 2006
    - <http://www.w3.org/TR/REC-xml-names/>
  - XML Schema Part 1: Structures 2nd Edition
    - W3C Recommendation 28 October 2004
    - <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/str>
  - XML Schema Part 2: Datatypes 2nd Edition
    - W3C Recommendation 28 October 2004
    - <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/dat>



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Web Services – WS



# WS: Web Services



- Definição (W3C):
  - Sistema de software projetado para suportar uma interação máquina-máquina interoperável sobre uma rede;
  - Interface é descrita por um documento em WSDL, processável por máquina;
  - Outros sistemas interagem com um Web Service na forma prescrita por sua descrição, usando mensagens SOAP tipicamente transportadas sobre o protocolo HTTP e XML serializados
- Padrões Relacionados:
  - WSDL 2.0 ( 26/06/2007 ), 1.2(11/06/2003)
  - SOAP 1.2 (27/04/2007), 1.1 (08/05/2000)

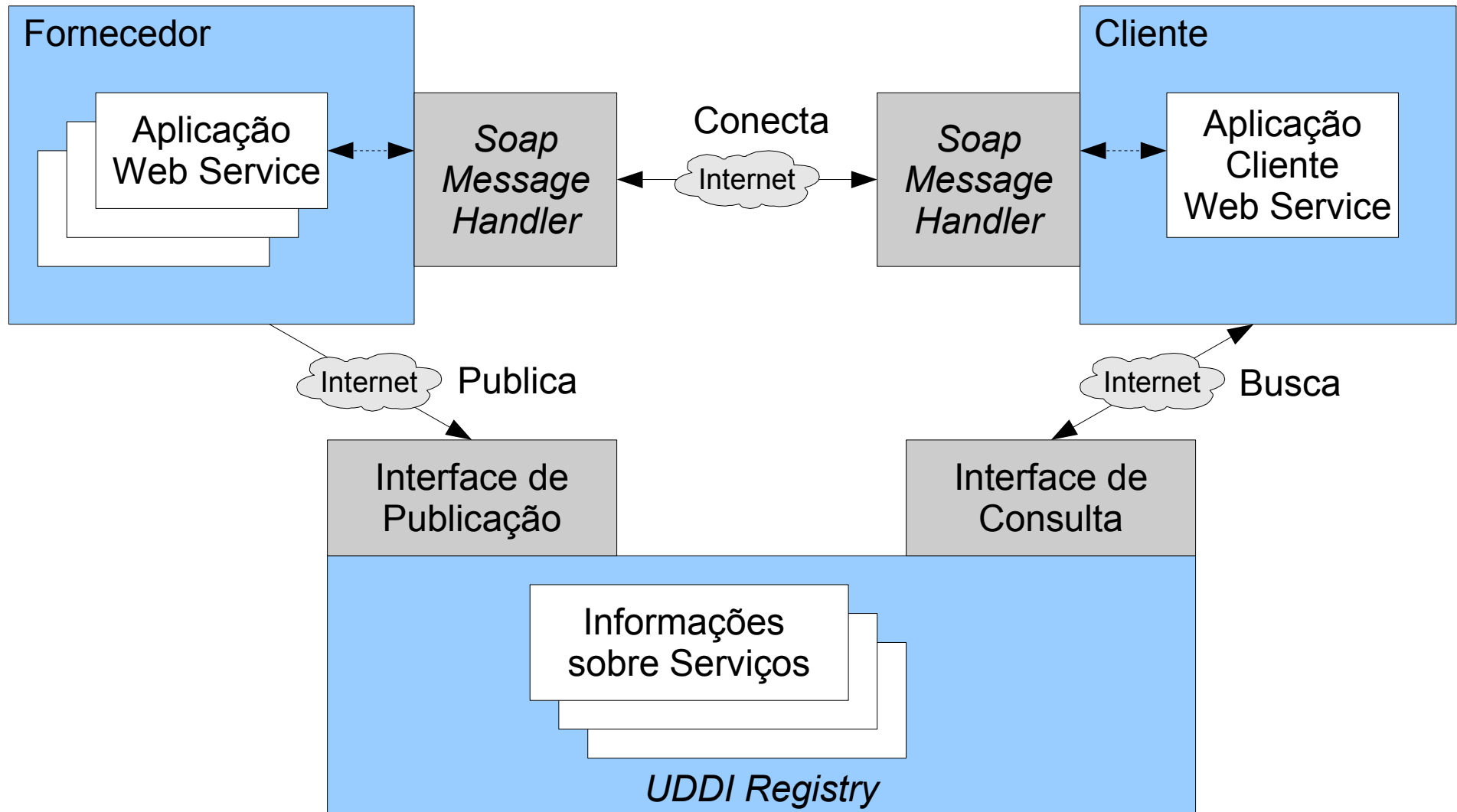


# WS: Web Services

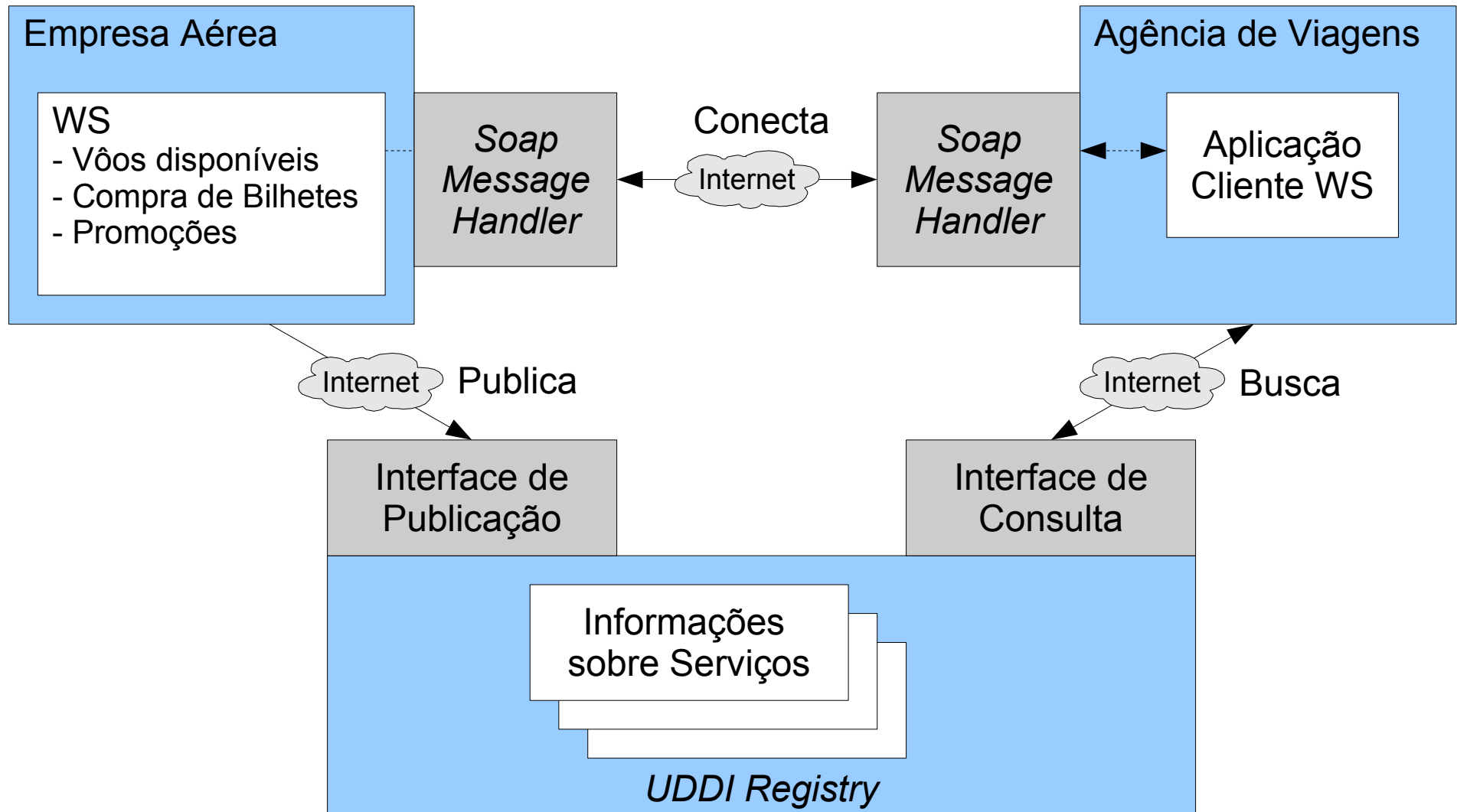


- Definição (IBM):
  - Um Web Service é uma interface formada por um conjunto de operações acessíveis pela rede através da troca de mensagens em formato XML padronizados.
  - A interface esconde os detalhes de implementação do serviço, permitindo que ele seja usado de maneira independente da plataforma de *software* ou *hardware* em que tenha sido implementado.

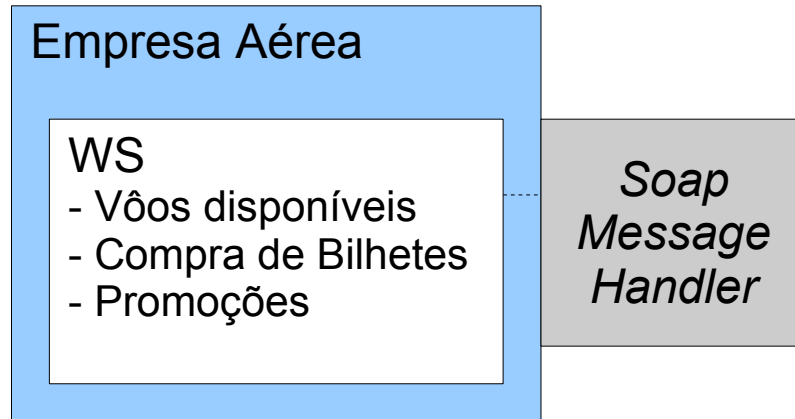




# WS: Exemplo



# Fornecendo um WS



## Fornecedor

- Implementa e/ou instala os componentes necessários a execução do WS
- Descreve o serviço utilizando o arquivo WSD (Web Service Description) em linguagem WSDL (Web Service Description Language)
- Publica informações sobre o serviço para que o mesmo possa ser utilizado (opcional)



# Fornecendo um WS - WSD



- Documento em formato XML descrevendo o comportamento do WS em termos de:
  - Formatos e tipos de dados utilizados nas mensagens de requisições de operações e respectivas respostas;
  - Especificação do protocolo de transporte das mensagens e formatos de serialização de dados utilizados entre o cliente e provedor do *Web Service*;
  - Endereços de rede onde o *Web Service* pode ser invocado.



# WSD - Web Services Description



- Estrutura

## Definição WSDL

### Descrição Abstrata

portType (interface  
2.0)  
operation  
message

### Descrição Concreta

binding  
port (endpoint 2.0)  
service



# WSD - Descrição abstrata



Descreve o Web Service independente da plataforma tecnológica em que ele esteja implementado.

- portType – descrição alto nível da interface do serviço. Define quais mensagens um serviço pode processar por grupos de funções conhecidos como operações. Cada operação representa uma ação que pode ser realizada pelo Web Service.
- Operação – análogo a um método de uma Classe , possui parâmetros de entrada e saída representados por mensagens.



# WSD - Descrição concreta



- binding – descreve a tecnologia (protocolo) de comunicação a ser utilizada para troca de mensagens. SOAP é o mais usado. O escopo pode variar de toda a interface à uma operação específica.
- port – identifica um endereço físico para acesso ao serviço



# Exemplo

- [tewscompat.wsdl](#)





# Exemplo



WSD

```
<?xml version="1.0"?>
<definitions name="TeWebServices" .... >
...
    <message name="searchWithinDistanceRequest">
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
    </message>
...
    <message name="searchWithinDistanceResponse">
        <part name="return" type="s:TeArrayOfstring"/>
    </message>
...
    <portType name="TeWebServices">
...
        <operation name="searchWithinDistance" parameterOrder="inputString">
            <input message="tns:searchWithinDistanceRequest"/>
            <output message="tns:searchWithinDistanceResponse"/>
        </operation>
...
    </portType>
...
</definitions>
```

## Código Fonte

```
axis2_array_list_t*
searchWithinDistance(const axis2_env_t *env, axis2_char_t* database,axis2_char_t* layer,axis2_char_t* criterio);
```



# Exemplo



WSD

```
<?xml version="1.0"?>
<definitions name="TeWebServices" .... >
...
    <message name="searchWithinDistanceRequest">
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
    </message>
...
    <message name="searchWithinDistanceResponse">
        <part name="return" type="s:TeArrayOfstring"/>
    </message>
...
    <portType name="TeWebServices">
...
        <operation name="searchWithinDistance" parameterOrder="inputString">
            <input message="tns:searchWithinDistanceRequest"/>
            <output message="tns:searchWithinDistanceResponse"/>
        </operation>
...
    </portType>
...
</definitions>
```

Código Fonte

```
axis2_array_list_t*
searchWithinDistance(const axis2_env_t *env, axis2_char_t* database,axis2_char_t* layer,axis2_char_t* criterio);
```



# Exemplo



WSD

```
<?xml version="1.0"?>
<definitions name="TeWebServices" .... >
...
    <message name="searchWithinDistanceRequest">
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
    </message>
...
    <message name="searchWithinDistanceResponse">
        <part name="return" type="s:TeArrayOfstring"/>
    </message>
...
    <portType name="TeWebServices">
        ...
        <operation name="searchWithinDistance" parameterOrder="inputString">
            <input message="tns:searchWithinDistanceRequest"/>
            <output message="tns:searchWithinDistanceResponse"/>
        </operation>
        ...
    </portType>
    ...
</definitions>
```

Código Fonte

```
axis2_array_list_t*
searchWithinDistance(const axis2_env_t *env, axis2_char_t* database,axis2_char_t* layer,axis2_char_t* criterio);
```



# Exemplo



WSD

```
<?xml version="1.0"?>
<definitions name="TeWebServices" .... >
...
    <message name="searchWithinDistanceRequest">
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
    </message>
    ...
    <message name="searchWithinDistanceResponse">
        <part name="return" type="s:TeArrayOfstring"/>
    </message>
    ...
    <portType name="TeWebServices">
    ...
        <operation name="searchWithinDistance" parameterOrder="inputString">
            <input message="tns:searchWithinDistanceRequest"/>
            <output message="tns:searchWithinDistanceResponse"/>
        </operation>
    ...
    </portType>
    ...
</definitions>
```

Código Fonte

```
axis2_array_list_t*
searchWithinDistance(const axis2_env_t *env, axis2_char_t* database,axis2_char_t* layer,axis2_char_t* criterio);
```

WSD

```
<?xml version="1.0"?>
<definitions name="TeWebServices" .... >
...
  <message name="searchWithinDistanceRequest">
    <part name="inputString" type="xsd:string"/>
    <part name="inputString" type="xsd:string"/>
    <part name="inputString" type="xsd:string"/>
  </message>
  ...
  <message name="searchWithinDistanceResponse">
    <part name="return" type="s:TeArrayOfstring"/>
  </message>
  ...
  <portType name="TeWebServices">
    ...
    <operation name="searchWithinDistance" parameterOrder="inputString">
      <input message="tns:searchWithinDistanceRequest"/>
      <output message="tns:searchWithinDistanceResponse"/>
    </operation>
  </portType>
  ...
</definitions>
```

Código Fonte

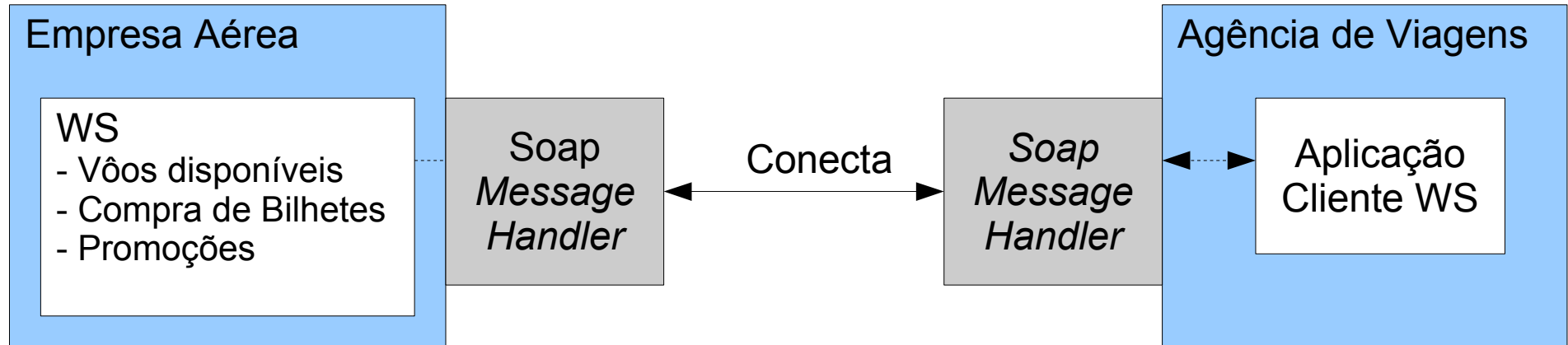
```
axis2_array_list_t*
searchWithinDistance(const axis2_env_t *env, axis2_char_t* database,axis2_char_t* layer,axis2_char_t* criterio);
```

WSD

```
<?xml version="1.0"?>
<definitions name="TeWebServices" .... >
...
    <message name="searchWithinDistanceRequest">
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
        <part name="inputString" type="xsd:string"/>
    </message>
    ...
    <message name="searchWithinDistanceResponse">
        <part name="return" type="s:TeArrayOfstring"/>
    </message>
    ...
    <portType name="TeWebServices">
        ...
        <operation name="searchWithinDistance" parameterOrder="inputString">
            <input message="tns:searchWithinDistanceRequest"/>
            <output message="tns:searchWithinDistanceResponse"/>
        </operation>
        ...
    </portType>
    ...
</definitions>
```

Código Fonte

```
axis2_array_list_t*
searchWithinDistance(const axis2_env_t *env, axis2_char_t* database,axis2_char_t* layer,axis2_char_t* criterio);
```



## Cliente

Implementa aplicação cliente capaz de interagir com o WS por meio de troca de mensagens SOAP de acordo com a descrição WSDL. Por meio do *SOAP Message Handler* a aplicação cliente:

- Conecta-se ao serviço (geralmente conexão HTTP);
- Gera requisições SOAP necessárias a invocação das operações disponíveis no WS;
- Recebe dados do WS.



# SOAP

- *Simple Object Access Protocol*
- Documento em formato XML
- Protocolo de Transporte utilizado para realizar a troca de mensagens XML entre os serviços.
- Os dados do SOAP são estruturados usando XML. Portanto, as mensagens podem ser processadas por quase todas as plataformas de *hardware*, sistemas operacionais e linguagens de programação.





# SOAP - Estrutura

<SOAP-ENV:envelope>

<!-- Elemento raiz do SOAP e define que essa é uma mensagem SOAP-->

<SOAP-ENV:header>

<!--Especifica informações específicas como autenticação (opcional)-->

</SOAP-ENV:header>

<SOAP-ENV:body>

<!--O elemento BODY contém o corpo da mensagem-->

<SOAP-ENV:fault>

<!--O elemento FAULT contém os erros que podem ocorrer-->

</SOAP-ENV:fault>

</SOAP-ENV:body>

</SOAP-ENV:envelope>



# SOAP – Exemplo Request



```
<soapenv:Envelope
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:inpe="http://inpe.org/">

  <soapenv:Body>

    <inpe:searchWithinDistance
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

      <inputString xsi:type="xsd:string">dbname=tutorial</inputString>

      <inputString xsi:type="xsd:string">distritos</inputString>

      <inputString xsi:type="xsd:string">x=-46 y=-23
        r=350000</inputString>

    </inpe:searchWithinDistance>

  </soapenv:Body>

</soapenv:Envelope>
```



# SOAP – Exemplo Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:ns1="http://inpe.br">
  <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsa:Action>base#searchWithinDistance</wsa:Action>
    <wsa:From>
      <wsa:Address>http://127.0.0.1:9090/axis2/services/TeWS</wsa:Address>
    </wsa:From>
    <wsa:MessageID>bddd5994-6896-4bc1-b577-e105be30ca07</wsa:MessageID>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:echoString xmlns:ns1="http://inpe.br">
      <x>Valinhos</x>
      <x>Campinas</x>
    </ns1:echoString>
  </soapenv:Body>
</soapenv:Envelope>
```



# UDDI Registry



- *Universal Discovery Description and Integration*
  - Mecanismo baseado em padrões que possibilita a classificação, catalogação e gerenciamento de WS, de modo que eles possam ser localizados e utilizados por outras aplicações.
  - Funcionam como “Páginas Amarelas” de Serviços.
  - Possibilita o compartilhamento de WS na implementação de sistemas de mais alto nível.
  - Integração dinâmica de WS.
  - Padrão definido dentro do *OASIS consortium*.

- Tipos XML utilizados na descrição de um WS

**businessEntity** – Informação sobre o fornecedor que publicou a informação sobre o WS.

Contem 1 ou mais

**businessService** – Informação descritiva sobre uma família de serviços.

Contem 1 ou mais

**bindingTemplate** – Informação técnica sobre a interface de um serviço, e suas especificações.

referência →

**tModel** – Descrição das especificações do serviço. (Pode conter WSDL).



# Referências



- Padrões W3C
  - <http://www.w3c.org>
- Web Services
  - <http://www.w3.org/2002/ws/>
- WSDL
  - <http://www.w3.org/2002/ws/desc/>
- SOAP
  - <http://www.w3.org/2000/xp/Group/>
- UDDI
  - <http://www.uddi.org/>



# WS: Ferramentas livres - Axis



- O que é ?
  - Apache Axis é uma *framework* para construção de processadores de mensagens SOAP, tais como programas clientes, servidores, gateways, etc ...;
  - Disponível nas linguagens Java, C++ (descontinuada) e C.
  - Projeto pertencente a lista dos projetos da Apache Software Foundation sob o esforço Web Services Project @ Apache. (<http://ws.apache.org>)



# WS: Ferramentas livres - Axis



## História

- Evolução do projeto originalmente criado pela IBM em 2000 de nome SOAP4J.
- Projeto Axis
  - Abordagem inicial na construção da ferramenta (release 1.0 lançado em 2002);
  - Versão não compatível com especificações WS-I (<http://www.ws-i.org>) . Implementação em C e C++ descontinuada.
- Projeto Axis2
  - Reengenharia da versão inicial Axis visando arquitetura mais modular e flexível, mais eficiência, suporte a especificações WS-I. (Java release 1.0 em 2005, C release 1.0 2007).





# WS: Ferramentas livres - Axis



## História

- Últimos *releases* ferramenta Axis2

Linguagem	Versão	Data	Características
Java	1.2	27/04/2007	Suporte a SOAP 1.1/1.2, WSDL 1.1/2.0
C	1.0	06/05/2007	Suporte a SOAP 1.1/1.2



# WS: Ferramentas livres - gSoap



- <http://www.cs.fsu.edu/~engelen/soap.html>



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Open Geospatial Consortium



# OGC: Open Geospatial Consortium



- Consórcio formado por empresas, universidades e agências governamentais de diversos países.
- Objetivo:
  - Promover o desenvolvimento de padrões que facilitem a interoperabilidade entre sistemas que envolvam informações geo-espaciais.
- Os produtos do trabalho do OGC são apresentados sob a forma de especificações de interfaces e padrões de intercâmbio.



# OGC: Especificações



- Especificações de Implementação:
  - Audiência mais técnica, considerada de nível de implementação
- Especificação Abstrata:
  - Fundamentos conceituais, sobre os quais as maiorias das especificações se apoiam
- Modelo de Referência
- Documentos de “Best Practices”:
  - Documentos discutindo práticas relacionadas ao uso ou implementação das especificações
- Artigos para Discussão:
  - Usado pelos grupos de trabalhos, não representando uma posição oficial do OGC



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Web Map Service – WMS



# WMS: Web Map Service



- Definição:
  - Especificação do OGC para serviços de produção de mapas na Web
- Versões da Especificação:
  - 1.0.0: 19/04/2000, 03-109r1, 45 páginas
  - 1.1.0: 21/01/2001, 01-068r3, 73 páginas
  - 1.1.1: 16/01/2002, 01-047r2, 82 páginas
  - 1.3.0: 15/03/2006, 00-028, 85 páginas



# WMS: Aplicação



Ministério Meio Ambiente  
(MMA): Florestas Nacionais



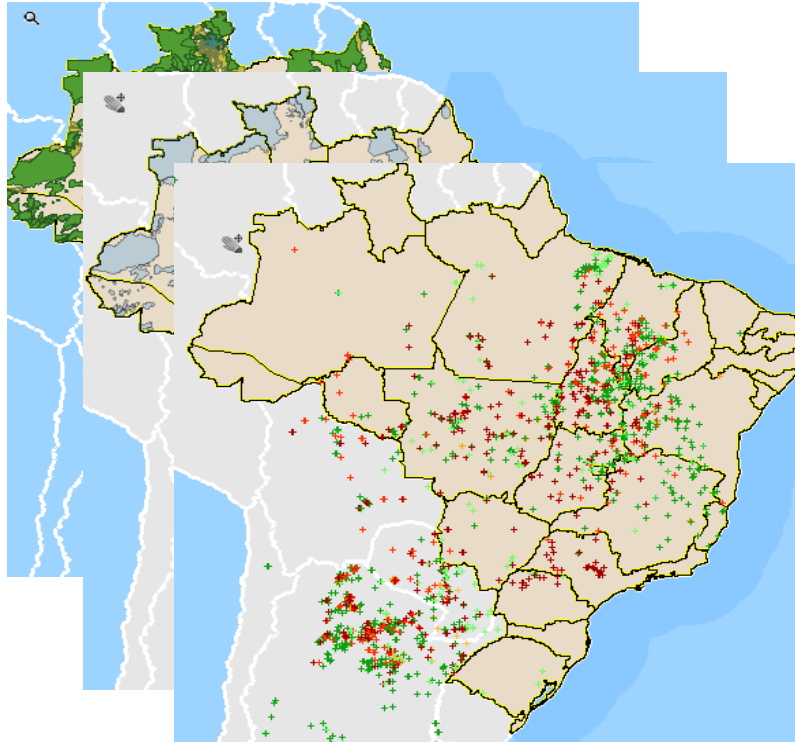


# WMS: Aplicação



FUNAI  
Terras Indígenas

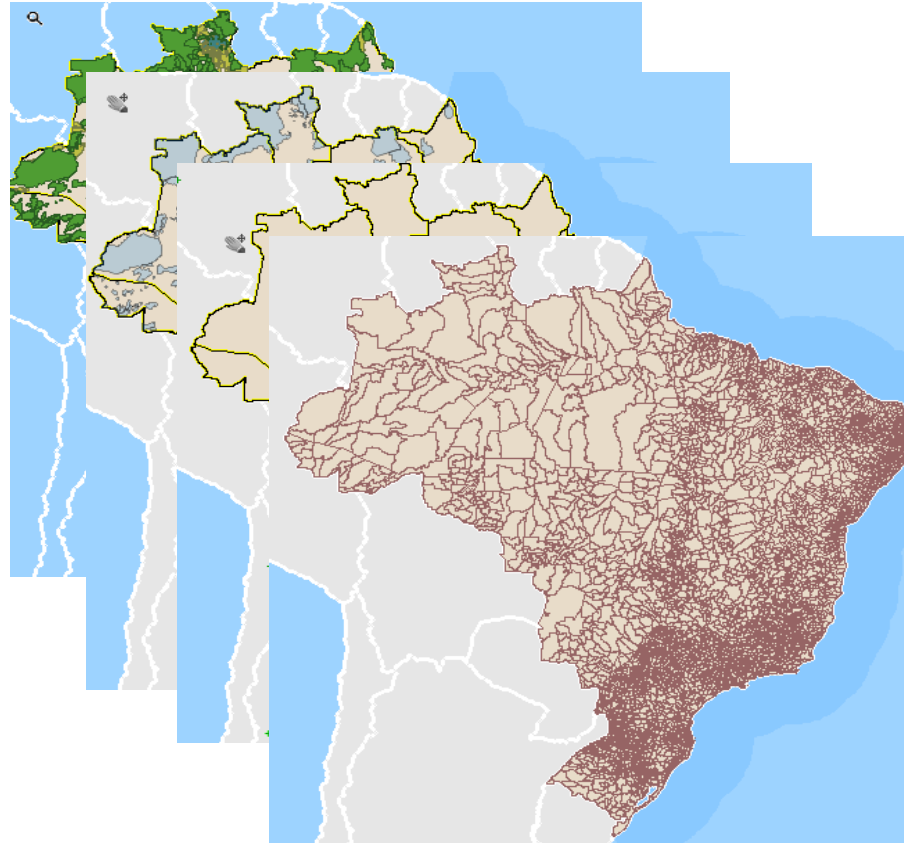
# WMS: Aplicação



INPE/CPTEC  
Focos Incêndio



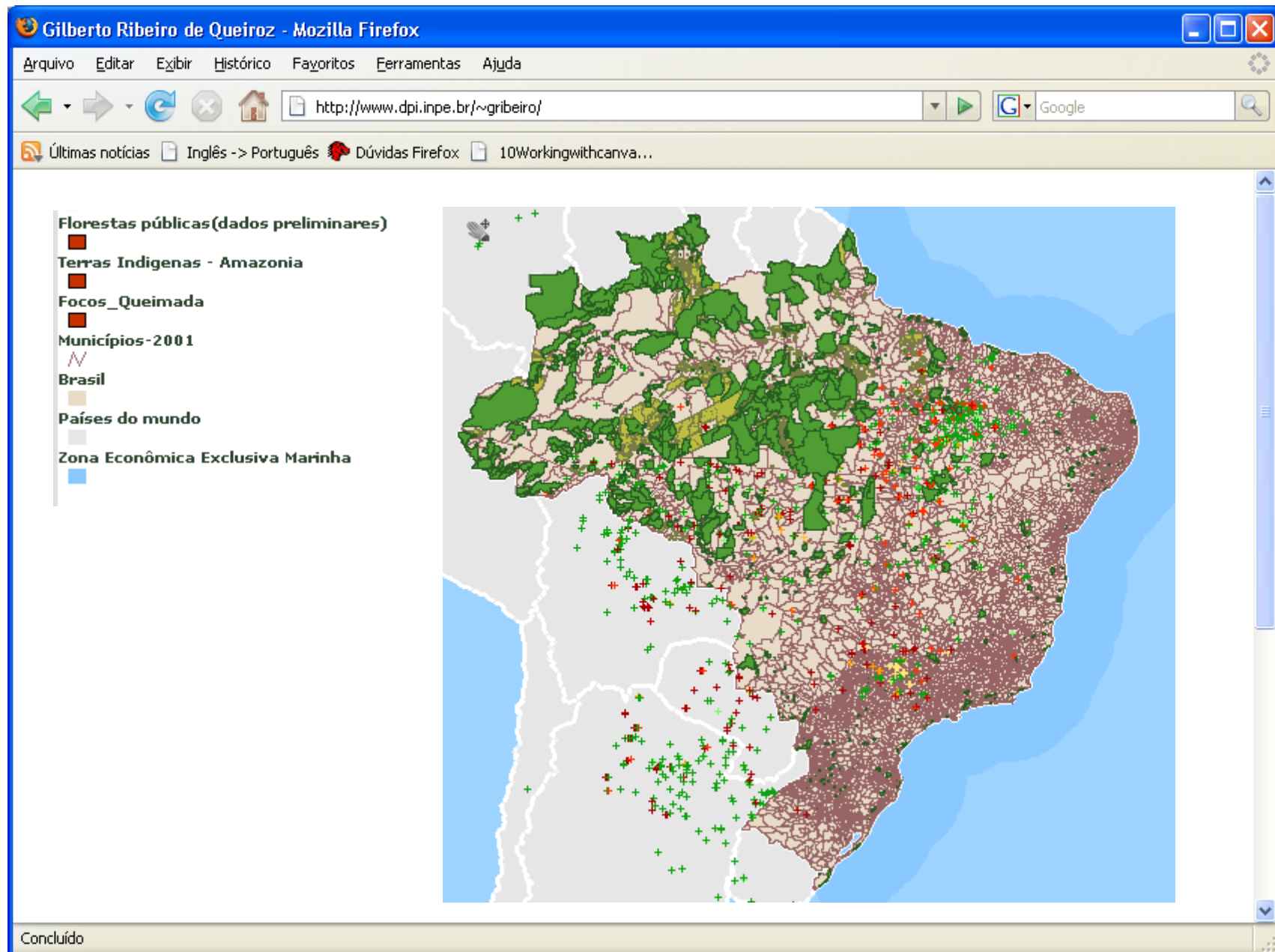
# WMS: Aplicação



IBGE  
Limites Municipais



# WMS: Aplicação





# WMS: Operações Suportadas



- **GetCapabilities:**

- Retorna um documento XML, descrevendo quais as interfaces suportadas pelo servidor, quais as camadas de informação que podem ser servidas e em quais formatos de imagem ele pode servir as camadas

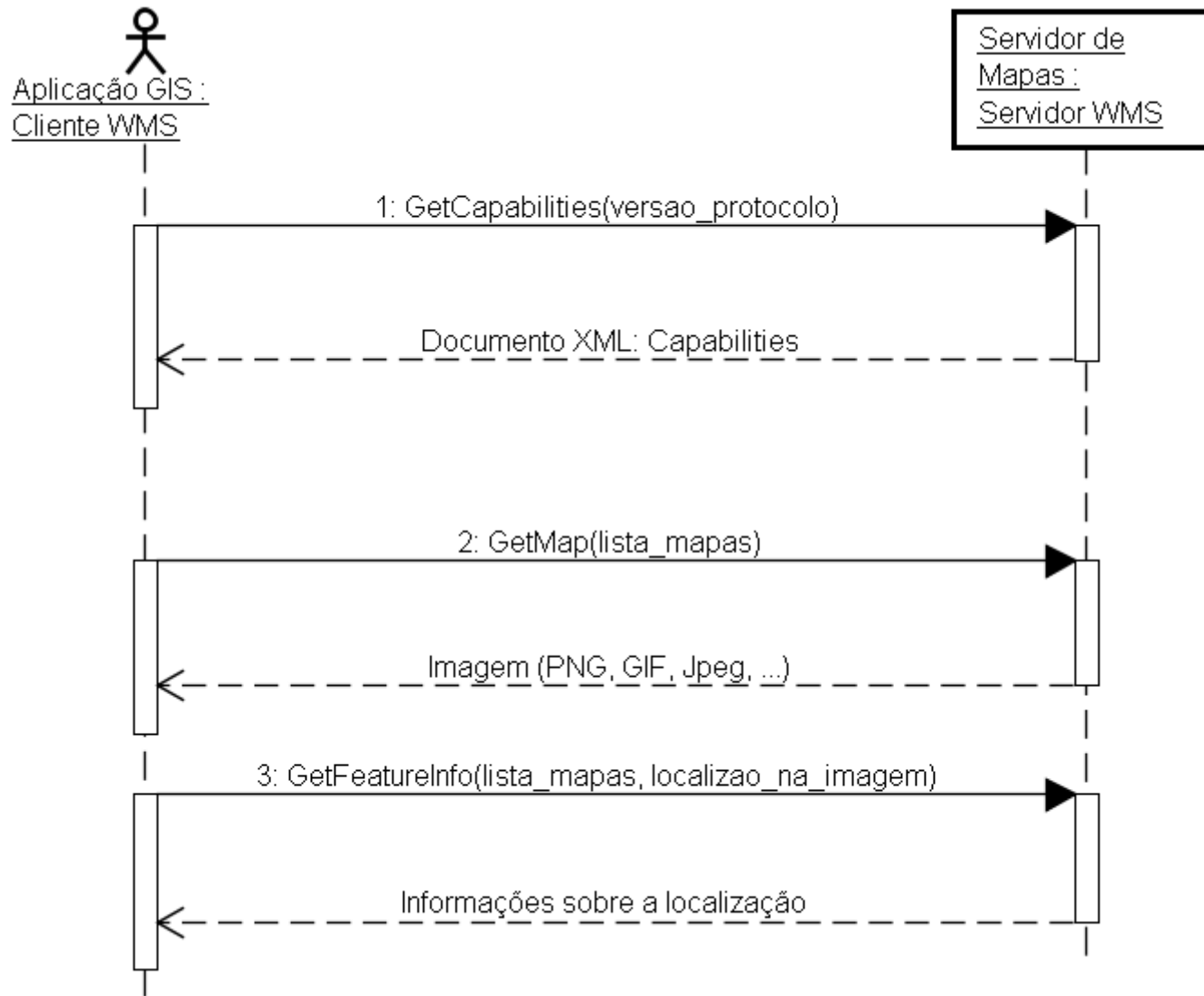
- **GetMap:**

- permite que os clientes solicitem “figuras” correspondentes aos mapas que podem ser servidos

- **GetFeatureInfo:**

- fornece maiores informações aos clientes a respeito de pontos específicos na imagem retornada pelo servidor

# WMS: Operações Suportadas





# WMS: Requisições



- Até a versão 1.3.0, está previsto que as requisições (*requests*) deverão ser realizadas por meio de URLs:
  - HTTP GET ou POST
- Os parâmetros de uma requisição são expressos em termos de pares:
  - nome/valor (kvp – key-value-pair)
  - Podemos incluir parâmetros adicionais na requisição



# WMS: Requisições



- Um navegador Web convencional pode ser utilizado para enviar solicitações a um servidor de mapas através de URLs:
  - **`http://mapas.mma.gov.br/cgi-bin/mapserv?`**  
`map=/opt/www/html/webservices/ucs.map&`  
**`REQUEST=GetCapabilities&SERVICE=WMS`**

[Ver no Navegador](http://mapas.mma.gov.br/cgi-bin/mapserv?map=/opt/www/html/webservices/ucs.map&REQUEST=GetCapabilities&SERVICE=WMS)





# WMS: GetCapabilities

Parâmetro	Valores	Obrigatório
VERSION	1.3.0	Não
REQUEST	GetCapabilities	Sim
SERVICE	WMS	Sim

- Solicitando o documento Capabilities de um servidor WMS:
  - **[http://mapas.mma.gov.br/cgi-bin/mapserv?](http://mapas.mma.gov.br/cgi-bin/mapserv?map=/opt/www/html/webservices/ucs.map&REQUEST=GetCapabilities&SERVICE=WMS)**  
**[map=/opt/www/html/webservices/ucs.map&](http://mapas.mma.gov.br/cgi-bin/mapserv?map=/opt/www/html/webservices/ucs.map&REQUEST=GetCapabilities&SERVICE=WMS)**  
**[REQUEST=GetCapabilities&SERVICE=WMS](http://mapas.mma.gov.br/cgi-bin/mapserv?map=/opt/www/html/webservices/ucs.map&REQUEST=GetCapabilities&SERVICE=WMS)**

**[Ver no Navegador](http://mapas.mma.gov.br/cgi-bin/mapserv?map=/opt/www/html/webservices/ucs.map&REQUEST=GetCapabilities&SERVICE=WMS)**



# WMS: GetCapabilities



- Retorna como resultado um documento XML que contém, entre outras coisas, as seguintes informações:
  - As camadas de informação disponíveis pelo serviço
  - Projeções em que a camada pode ser servida
  - O retângulo envolvente das camadas
  - Formatos disponíveis para o desenho dos mapas
  - Opcionalmente:
    - Os estilos disponíveis para desenho de cada camada
    - Figura da legenda
    - URL do dado “real”



# WMS: GetMap



Parâmetro	Valores	Obrigatório
VERSION	1.3.0	Sim
REQUEST	GetMap	Sim
LAYERS	Lista separada por “,”	Sim
STYLES	Lista separada por “,”	Sim
SRS	EPSG:4326	Sim
BBOX	xmin,ymin,xmax,ymax	Sim
WIDTH	800	Sim
HEIGHT	600	Sim
FORMAT	image/png	Sim
TRANSPARENT	TRUE ou FALSE	Não
BGCOLOR	0xFFFFFFFF	Não
EXCEPTIONS	INIMAGE	Não



# WMS: GetMap (Exemplo)



- Solicitando o mapa das Unidades de Conservação Federais e Estaduais:
  - **`http://mapas.mma.gov.br/cgi-bin/mapserv?`**  
`map=/opt/www/html/webservices/ucs.map&`  
**`SERVICE=WMS&VERSION=1.1.1&`**  
**`REQUEST=GetMap&`**  
**`LAYERS=ucsef&`**  
**`SRS=EPSG:4326&`**  
**`BBOX=-73.9909,-33.6039,-32.349,5.27222&`**  
**`STYLES=default&`**  
**`FORMAT=image/png&`**  
**`WIDTH=800&HEIGHT=600&`**  
**`TRANSPARENT=TRUE&`**  
**`EXCEPTIONS=text/xml`**

[Ver no Navegador](#)



# WMS: GetFeatureInfo



Parâmetro	Valores	Obrigatório
VERSION	1.3.0	Sim
REQUEST	GetFeatureInfo	Sim
Parâmetros da requisição GetMap	Lista separada por “,”	Sim
QUERY_LAYERS	Lista separada por “,”	Sim
I	Coluna do Pixel	Sim
J	Linha do Pixel	Sim
INFO_FORMAT	Default: MIME	Não
FEATURE_COUNT	Default: 1	Não



# WMS: GetFeatureInfo



- Solicitando informações sobre uma determinada localização no mapa de Unidades de Conservação Federais e Estaduais:
  - **`http://mapas.mma.gov.br/cgi-bin/mapserv?map=/opt/www/html/webservices/ucs.map&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetFeatureInfo&QUERY_LAYERS=ucsef&X=390&Y=158&INFO_FORMAT=application/vnd.ogc.gml&LAYERS=ucsef&SRS=EPSG:4326&BBOX=-73.9909,-33.6039,-32.349,5.27222&STYLES=default&FORMAT=image/png&WIDTH=800&HEIGHT=600&TRANSPARENT=TRUE&EXCEPTIONS=text/xml`**

[Ver no Navegador](#)



# WMS: Considerações



- Exceptions:
  - Caso o servidor WMS receba uma requisição incorreta, ele deve lançar uma exceção
  - Uma exceção é um documento que possui um formato específico:

```
<ServiceExceptionReport version="1.1.1">  
  <ServiceException code="LayerNotDefined">  
    WMS server error. Invalid layer(s) given in the LAYERS parameter.  
  </ServiceException>  
</ServiceExceptionReport>
```



# WMS: Considerações



- O servidor e o cliente negociam a versão do protocolo através da interface GetCapabilities
- Possibilidade de associar informações temporais e de dimensão às camadas (ver esquema do documento Capabilities)





# WMS: Considerações



- A definição do documento Capabilities encontra-se:
  - DTD: capabilities\_1\_0\_0.dtd (versão 1.0.0)
  - DTD: capabilities\_1\_1\_0.dtd (versão 1.1.0)
  - DTD: capabilities\_1\_1\_1.dtd (versão 1.1.1)
  - No esquema: capabilities\_1\_3\_0.xsd (versão 1.3.0)



# WMS: Considerações



- Na versão 1.1.0 existe uma interface opcional para serviços WMS com suporte a SLD:
  - DescribeLayer
- Na versão 1.1.1 existem interfaces opcionais para serviços WMS com suporte a SLD:
  - DescribeLayer
  - GetLegendGraphic
  - GetStyles
  - PutStyles



# WMS: Considerações



- Cascading Map Server:
  - Se comporta como um cliente de outro servidor de mapas e como servidor de mapas para seus clientes
  - Fornece um mecanismo conveniente para agregar recursos (capabilities) de servidores de mapas individuais em um único local lógico
  - Um Cascading Map Server pode realizar serviços adicionais:
    - Exemplo: Ele pode converter diferentes formatos de imagem (PNG, JPEG, etc. no formato GIF). Os clientes que só podem exibir GIFs podem continuar a se beneficiar da saída de servidores de mapa que produzem apenas JPEG ou PNG;



# WMS: Principais Mentores



- Autor:
  - Jeff de la Beaujardiere (NASA Goddard Space Flight Center)
  - Allan Doyle (International Interfaces, Inc.)
- Principais Organizações:
  - CubeWerx Inc., NASA, CadCorp, IONIC Software s.a., US Army Engineer Research and Development Center



# WMS: Padrões Relacionados



- Padrões que trabalham de forma complementar:
  - Styled Layer Descriptor (SLD)
  - Web Map Context



# WMS: Ferramentas



Nome	Tipo
deegree	C/S
GeoServer	S
MapServer	C/S
MapBender	C
Gaia	C
MapBrowser	C
Oracle MapViewer	S
Mapbuilder	C
Cadcorp SIS Map Server	S
Udig 1.1	C
JUMP 1.1.2 (Unified Mapping Platform)	C
ArcIMS	S



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Geography Markup Language – GML



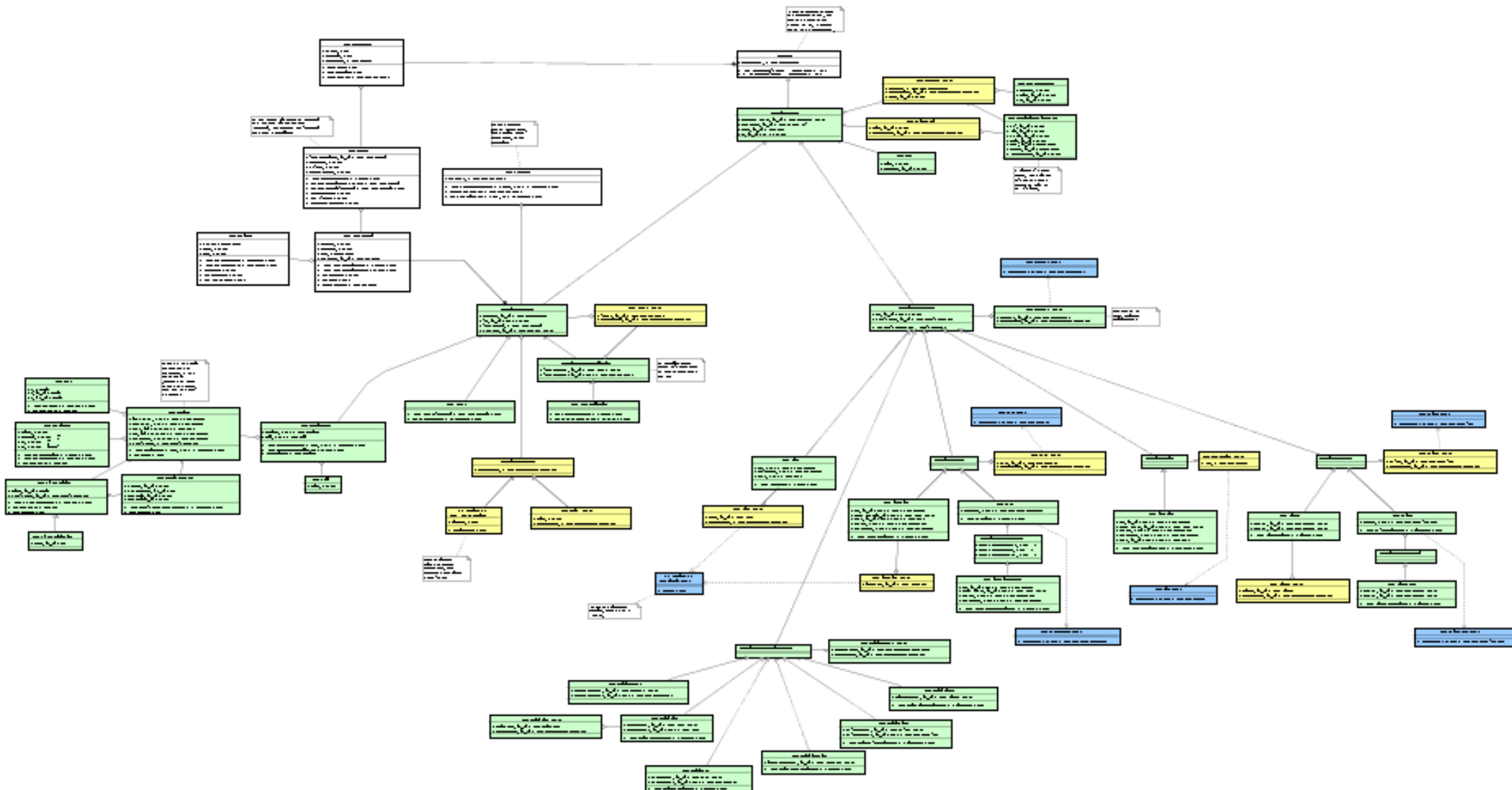
# GML: Geography Markup Language



- Definição
  - Especificação do OGC para codificar informação geográfica num arquivo XML
- Versões da Especificação
  - 3.1.1 (atual): apenas correções da 3.1.0
  - 3.1.0: OGC 03-105r1, 2004, 601 p.
  - 2.1.2: OGC 02-069, 2002, 66 p.



- Diagrama UML p/ feições simples





# GML: Componentes



- Abstratos

- Object

- Definição básica p/ qualquer objeto
    - Um documento pode ser um objeto GML, que por sua vez é um conjunto de objetos

- AbstractGMLType

- Qualquer objeto GML com uma identidade (nome, descrição, metadados e ID)

- Property

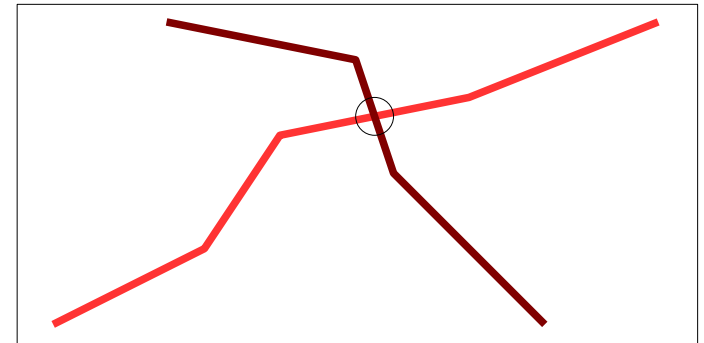
- Termo usado p/ referência a característica de um objeto GML
    - A propriedade é um nó filho do objeto
    - Os objetos podem ter um número infinito de propriedades
    - As propriedades podem ser passadas por valor ou por referência (xlink, usando seu ID)

- Exemplo referência x valor

```
<estrada>
  <nome>Estrada 1</nome>
  <TeGeometry>
    <gml:LineString>
      ...
      <gml:pointProperty>
        <gml:Point gml:id="point25">
          <gml:pos>360274.45 7412176.09</gml:pos>
        </gml:Point>
      </gml:pointProperty>
      ...
    </gml:LineString>
  </TeGeometry>
</estrada>

<estrada>
  <nome>Estrada 2</nome>
  <TeGeometry>
    <gml:LineString>
      ...
      <gml:pointProperty link:href="#point25">
        ...
      </gml:pointProperty>
    </gml:LineString>
  </TeGeometry>
</estrada>
```

## Cruzamento de estradas



**Propriedades em comum  
entre duas feições podem ser  
usadas por referência**



# GML: Componentes



- Coordenadas
  - Uma coordenada única
    - coord (GML2)
    - pos
  - Uma lista de coordenadas
    - coordinates (GML2)
    - posList

```
<gml:coord>
  <gml:X>100</gml:X>
  <gml:Y>50</gml:Y>
</gml:coord>

<gml:pos>100 50</gml:pos>

<gml:coordinates>-45.3, -22.8 -47.4, -22.1</gml:coordinates>

<gml:posList>-45.3 -22.8 -47.4 -22.1</gml:posList>
```



# GML: Componentes



- Coordenadas
  - Conjunto de atributos que identificam o sistema de coordenadas utilizado
  - srsName
    - Aponta para uma instância de um CRS codificado em GML
    - Normalmente é um código **EPSG**
  - srsDimension
    - A dimensão do sistema de coordenadas
  - Exemplo
    - 4 coordenadas em SAD69 / UTM zona 23

```
<gml:posList srsDimension="2" srsName="EPSG:29193">  
209875.37 7786108.83 209932.91 7785823.73 210330.08  
7783858.70 209966.27 7781855.12  
</gml:posList>
```

- Geometrias primitivas (0-1D)
  - Point
    - Um único ponto
  - Curve
    - Uma linha
  - LineString
    - Uma linha (GML2)
  - LinearRing
    - Um anel

```
<gml:Point>
  <gml:pos srsDimension="2">326358.80 7397825.65
</gml:pos>
</gml:Point>

<gml:Curve>
  <gml:segments>
    <gml:LineStringSegment>
      <gml:posList srsDimension="2">209875.37
7786108.83 209932.91 7785823.73 210330.08
7783858.70 209966.27 7781855.12
      </gml:posList>
    </gml:LineStringSegment>
  </gml:segments>
</gml:Curve>

<gml:LineString>
  <gml:coordinates>324576.16,7382767.53
324552.72,7382700.51 324504.21,7382669.83
324352.02,7382352.69</gml:coordinates>
</gml:LineString>
```



# GML: Componentes



- Geometrias primitivas (2D)
  - Surface
    - Um polígono
  - Polygon
    - Um polígono (GML2)

- Geometrias primitivas (2D) - exemplos

```
<gml:Polygon>
  <gml:outerBoundaryIs>
    <gml:LinearRing>
      <gml:coordinates>-46.5365637, -23.3630253 ...</gml:coordinates>
    </gml:LinearRing>
  </gml:outerBoundaryIs>
</gml:Polygon>

<gml:Surface>
  <gml:patches>
    <gml:PolygonPatch>
      <gml:exterior>
        <gml:LinearRing>
          <gml:posList srsDimension="2">335589.53 7356020.72 ...</gml:posList>
        </gml:LinearRing>
      </gml:exterior>
      <gml:interior>
        <gml:LinearRing>
          <gml:posList srsDimension="2">319687.00 7353007.00 ...</gml:posList>
        </gml:LinearRing>
      </gml:interior>
    </gml:PolygonPatch>
  </gml:patches>
</gml:Surface>
```

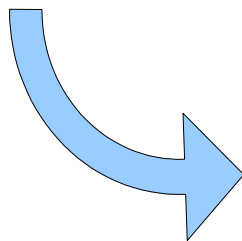
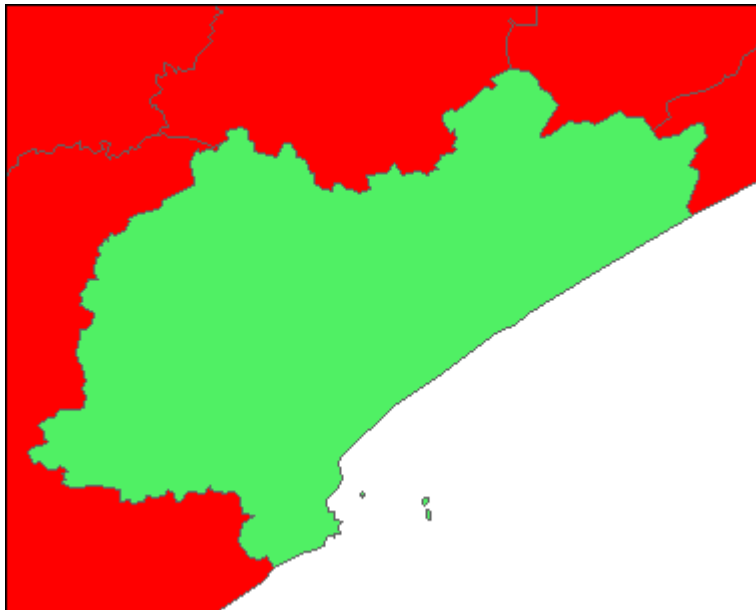




# GML: Componentes



- Geometrias compostas
  - MultiPoint
    - Vários pontos
  - MultiCurve
    - Várias linhas
  - MultiSurface
    - Vários polígonos



**Exemplo de geometria composta**

```
<microreg>
  <TeGeometry>
    <gml:MultiSurface srsName="EPSG:29193">
      <gml:surfaceMember>
        <gml:Surface>
          <gml:patches>
            <gml:PolygonPatch>
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList srsDimension="2">
                    263850.35 7309279.21 ...
                  </gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:PolygonPatch>
          </gml:patches>
        </gml:Surface>
      </gml:surfaceMember>
      <gml:surfaceMember> ...
        <gml:posList srsDimension="2">
          306642.38 7300712.22 ...
        </gml:posList>
      </gml:surfaceMember>
      ...
    </gml:MultiSurface>
  </TeGeometry>
  <nome>ITANHAEM</nome>
</microreg>
```



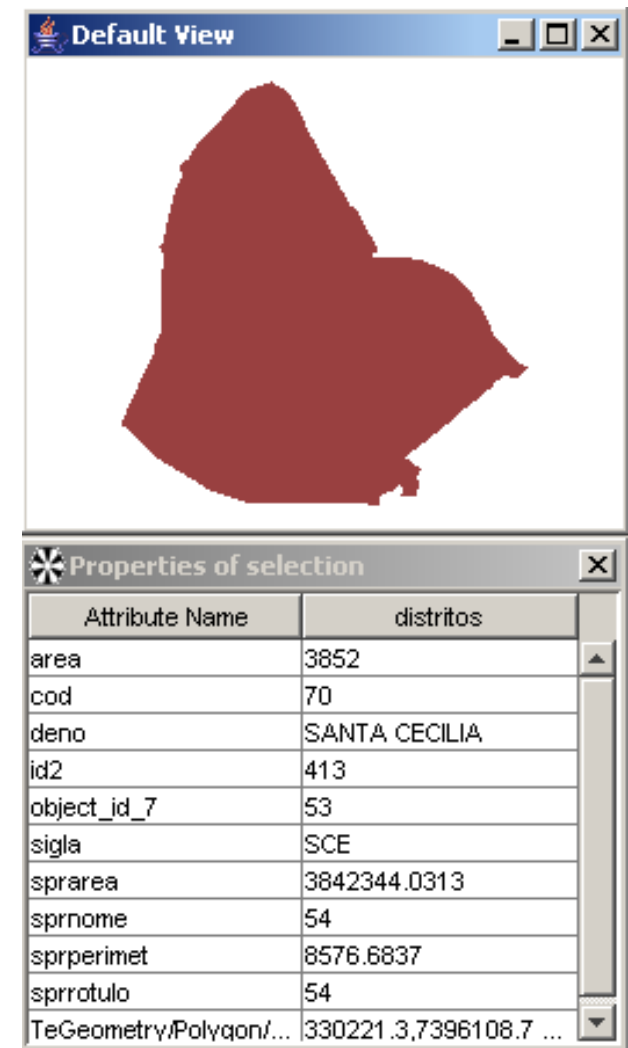
# GML: Componentes



- Feições
  - AbstractFeature
    - Qualquer coisa que pode ser considerada uma "feição GML", inclusive as compostas e seus componentes
    - Ex: lote, pessoa

- Feição simples

```
<distritos>
  <TeGeometry>
    <gml:Polygon srsName="EPSG:29193">
      <gml:outerBoundaryIs>
        <gml:LinearRing>
          <gml:coordinates>
            330221.3,7396108.7 ...
          </gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </TeGeometry>
  <sprarea>3842344.0313</sprarea>
  <sprperimet>8576.6837</sprperimet>
  <sprrotulo>54</sprrotulo>
  <sprnome>54</sprnome>
  <id2>413</id2>
  <area>3852</area>
  <cod>70</cod>
  <sigla>SCE</sigla>
  <deno>SANTA CECILIA</deno>
  <object_id_7>53</object_id_7>
</distritos>
```

The screenshot shows the TerraLib interface. The top window, titled "Default View", displays a map of a district (Santa Cecília) in a dark red color. Below the map is a window titled "Properties of selection" which contains a table of attributes for the selected district.

Attribute Name	distritos
area	3852
cod	70
deno	SANTA CECILIA
id2	413
object_id_7	53
sigla	SCE
sprarea	3842344.0313
sprnome	54
sprperimet	8576.6837
sprrotulo	54
TeGeometry/Polygon/...	330221.3,7396108.7 ...

- Feição composta



```
<Brasil>
  <gml:featureMember>
    <muníc_2001>
      <TeGeometry> ... </TeGeometry>
      <GEOCODIGO>3535606</GEOCODIGO>
      <NOME>Paraibuna</NOME>
      <LATITUDESE>-23,386</LATITUDESE>
      <LONGITUDES>-45,662</LONGITUDES>
      <AREA_TOT_G>809,794</AREA_TOT_G>
    </muníc_2001>
  </gml:featureMember>

  <TeGeometry> ... </TeGeometry>
  <Lingua>Português</Lingua>

</Brasil>
```

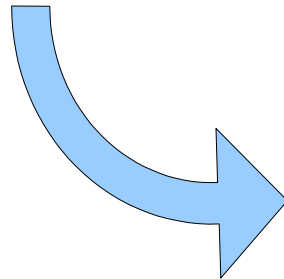
Attribute Name	muníc_2001	muníc_2001	muníc_2001
AREA_TOT_G	1099,613	751,443	809,794
GEOCODIGO	3549904	3518404	3535606
LATITUDESE	-23,179	-22,816	-23,386
LONGITUDES	-45,887	-45,193	-45,662
NOME	São José dos Campos	Guaratinguetá	Paraibuna
TeGeometry/MultiPolygo...	-46.0745847,-23.18019...	-45.4394659,-22.70283...	-45.7883270,-23.49139...
TeGeometry/MultiPolygo...	EPSG:4618	EPSG:4618	EPSG:4618



# GML: Aplicabilidade



```
testel=# \d mapa_distritos_sp
Table "public.mapa_distritos_sp"
  Column      |      Type      |
-----+-----+
sprarea       | double precision |
sigla         | character varying(3) |
deno          | character varying(25) |
```



**Convertendo uma  
representação no SGBD  
p/ um esquema GML**

## Descrição da tabela de atributos

### XML Schema p/ GML 2.1.2

```
<complexType name="distritos_Type">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="TeGeometry"
          type="gml:MultiPolygonPropertyType"/>
        <element name="sprarea" type="float"/>
        <element name="sigla">
          <simpleType>
            <restriction base="string">
              <maxLength value="3"/>
            </restriction>
          </simpleType>
        </element>
        <element name="deno">
          <simpleType>
            <restriction base="string">
              <maxLength value="25"/>
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```



# GML: Considerações relevantes



- Complexidade
  - Implementação complicada
  - Surgimento de novas specs, os *profiles*
- GML Profiles
  - Essencialmente as aplicações GML usam apenas um subconjunto dos muitos elementos definidos
  - Se algum subconjunto é especificado (e por vezes restrito), esse subconjunto é um *GML Profile*
  - 5 novas specs



# GML: Componentes



- Outros componentes
  - Geometrias 3D
    - Solid
  - Topologia
    - Node, Edge, Face
  - Coverages
    - Imagens, TINs
  - Estilos
    - Mecanismo para prover estilo (peso, cor) que pode ser colocado junto com os dados
  - E mais:
    - CRS (sistemas de coordenadas), informação temporal e feições dinâmicas, unidades, medidas, observações ...





# GML: Padrões relacionados



- Padrões que usam o GML diretamente
  - WFS
  - Filter
  - WCS
  - GeoRSS (não é OGC)



# GML: Ferramentas proprietárias



Nome	Ver	Tipo
Bentley Map 8.9.4	2 e 3	c
Bentley MicroStation Geospatial Extension 8.9.4	2 e 3	c
BILGI GIS GeoKIT	2 e 3	s/c
BILGI GIS GeoKIT WFS	2 e 3	s
Cadcorp GeognoSIS.NET 6.2	2 e 3	s/c
Cadcorp SIS 6.2	2 e 3	c
Cadcorp SIS Active Server Component 6.2	2 e 3	s/c
Cadcorp SIS Map Server 6.2	2 e 3	s/c
CodeSynthesis XSD 2.3.1	3	Encoder
IONIC RedSpider Catalog 2.3.5	2 e 3	s/c
IONIC RedSpider Enterprise 3.4	2 e 3	c
IONIC RedSpider Web 3.4	2 e 3	s/c
IONIC RedSpider Web 3.3	2 e 3	s/c
LuciadMap 6	2 e 3	s/c
Snowflake GO Loader 1.4	2 e 3	c
Snowflake GO Publisher Agent 1.2	2 e 3	s
Snowflake GO Publisher Desktop 1.2	2 e 3	c
Snowflake GO Publisher WFS 1.2	2 e 3	s
Social Change WebMap Composer 2.03	2 e 3	c
Carbon Project CarbonTools 2	2 e 3	COM Obj Lib
Carbon Project Gaia 2	2 e 3	c



# GML: Ferramentas livres



Nome	Ver	Tipo	Obs
lat/lon deegree iGeoPortal 2.0.0	2 e 3	c	
lat/lon deegree Web Feature Service 2.0.0	2 e 3	s	
lat/lon deegree Web Gazetteer Service 2.0.0	3	s	
OpenJUMP	2	c	
Quantum GIS 0.8	2	c	Via OGR
uDIG 1.1	2?	c	
GeoTools 2.3	2	c	Java ToolKit
GeoServer 1.5	2	s	
MapServer 4.10.2	2 e 3	s	
OGR	2	C++ Lib	Sem XSD



# GML: Ferramentas



- Considerações

- Nenhuma implementação proprietária leva o selo “GML Compliant”
- Em compensação, algumas têm o “WFS Compliant”, o que pode indicar uma boa implementação do GML
- As ferramentas livres têm basicamente duas implementações:
  - Java: pacote da VividSolutions (aumentada pelo GeoServer)
  - C++: pacote OGR, e escrita no estilo MapServer
- O pacote OGR faz parte da GDAL Lib
  - Independe do XML Schema, vai lendo e se adaptando



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Filter Encoding



# Filter: Filter Encoding



- Definição:
  - Especificação do OGC para codificar expressões de filtro (restrição) em XML
  - Pode ser usada por qualquer outro serviço que precise expressar predicados em XML
  - Pode ser transformada em outra linguagem alvo (cláusula WHERE da SQL ou XPath para consultas em documentos XML)
- Versões da Especificação:
  - 1.0.0: 17/05/2001, 02-059, 42 páginas
  - 1.1.0: 03/05/2005, 04-095, 40 páginas



# Filter: Aplicação



- Cenários:
  - “Estou acessando um servidor de mapas que me mostra todos os municípios brasileiros. Gostaria de ver apenas os de Minas Gerais”.
  - “Gostaria de salvar apenas os polígonos da classe desmatamento no formato GML. Nuvens e Agregados devem ser salvos em arquivos do formato *shp*”.



# Filter: Organização



- Propriedades (simples x complexas)
- Filtro (qualquer predicado válido)
- Operadores Espaciais
- Operadores de Relacionais
- Operadores Lógicos
- Expressões
- Operadores Aritméticos
- Literais
- Funções





# Filter: Integrando ao Serviço



- Se um serviço suporta o uso de Filter ele deve publicar isso em seu documento Capabilities:

```
<Filter_Capabilities>
<ogc:Filter_Capabilities
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/ogc
http://.../1.1.0/filterCapabilities.xsd">
  <ogc:Spatial_Capabilities>
    <ogc:GeometryOperands>
      <ogc:GeometryOperand>gml:Envelope</ogc:GeometryOperand>
      <ogc:GeometryOperand>gml:Point</ogc:GeometryOperand>
      <ogc:GeometryOperand>gml:LineString</ogc:GeometryOperand>
      <ogc:GeometryOperand>gml:Polygon</ogc:GeometryOperand>
      ...
    </ogc:GeometryOperands>
    <ogc:SpatialOperators>
      <ogc:SpatialOperator name="BBOX" />
      <ogc:SpatialOperator name="Equals" />
      <ogc:SpatialOperator name="Disjoint" />
      <ogc:SpatialOperator name="Intersect" />
      ...
    </ogc:SpatialOperators>
  </ogc:Spatial_Capabilities>
</ogc:Filter_Capabilities>
```

# Filter: Root Element

- Qualquer predicado válido, começando com o elemento <Filter>:

```
<xsd:element name="Filter" type="ogc:FilterType"/>
<xsd:complexType name="FilterType">
  <xsd:choice>
    <xsd:element ref="ogc:spatialOps"/>
    <xsd:element ref="ogc:comparisonOps"/>
    <xsd:element ref="ogc:logicOps"/>
    <xsd:element ref="ogc:_Id" maxOccurs="unbounded"/>
  </xsd:choice>
</xsd:complexType>
```



# Filter: Operadores Relacionais



```
<Filter>
  <PropertyIsEqualTo>
    <PropertyName>Idade</PropertyName>
    <Literal>30</Literal>
  </PropertyIsEqualTo>
</Filter>
```



# Filter: Operadores Espaciais



- A semântica é a mesma da [Simple Feature Specification](#)
- Ex: Identificar todas as feições que estão dentro de uma área (polígono) de interesse:

```
<Filter>
  <Intersects>
    <PropertyName>Geometry</PropertyName>
    <gml:Polygon srsName="63266405">
      <gml:outerBoundaryIs>
        <gml:LinearRing>
          <gml:posList> ... </gml:posList>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </Intersects>
</Filter>
```



# Filter: Property Names



- O elemento `<PropertyName>` é usado para codificar o nome de qualquer propriedade de um objeto:
  - Aplica-se a atributos escalares ou espaciais
  - Propriedades simples podem ser referenciadas pelo seu próprio nome
  - Propriedades complexas (agregadas) podem usar XPath



# Filter: Padrões Relacionados



- Padrões que trabalham de forma complementar:
  - WFS
  - SLD
  - Common Query Language (CQL) – OGC Catalog Interface V1.0
- Padrões W3C:
  - XML e Esquemas



# Filter: Principais Mentores



- Autor:
  - Panagiotis A. Vretanos (CubeWerx, Inc.)
- Principais Organizações:
  - CubeWerx Inc., Intergraph Corp., IONIC Software, Laser-Scan Ltd.
- Colaboradores:
  - NASA, Galdos System Inc., CadCorp Ltd, U.S. Census, ESRI



# Filter



Nome	Tipo
deegree	S
GeoServer	S
MapServer	S





**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Web Feature Service – WFS



# WFS: Web Feature Service



- Definição:
  - Especificação do OGC para acesso e manipulação de dados geográficos na Web
- Versões da Especificação:
  - 1.0.0: 17/05/2002, 02-058, 105 páginas
  - 1.1.0: 03/05/2005, 04-094, 131 páginas

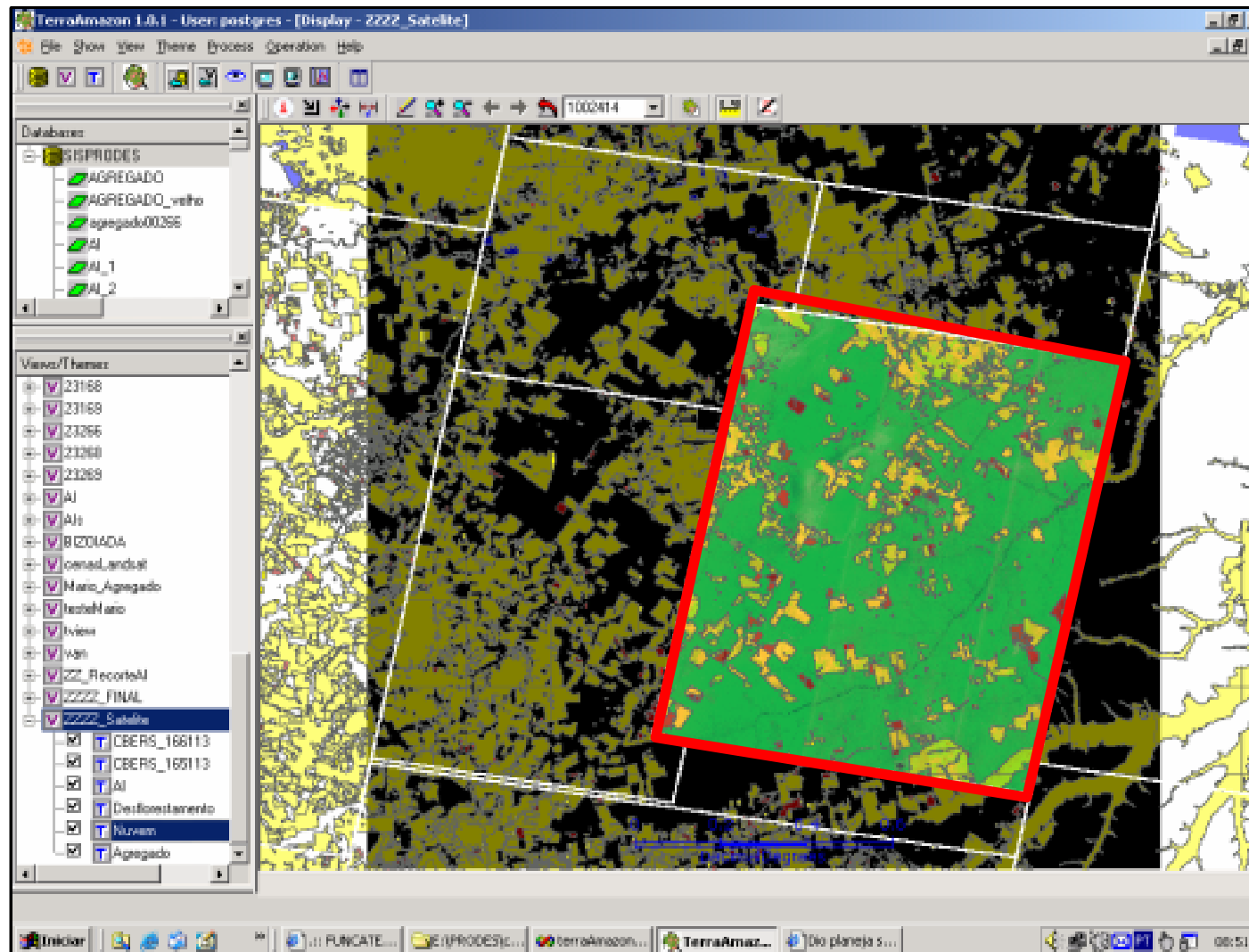


# WFS: Aplicação



- Cenário:
  - “Tenho um grande banco de dados geográficos com dados de desmatamento (polígonos). Gostaria de permitir o acesso a esses dados a outros pesquisadores, que poderiam, por exemplo, estudar padrões de desmatamento”.
  - Disponibilizando um servidor WFS como mediador para este banco de dados seria possível que os pesquisadores tivessem acesso aos dados, independente do formato de armazenamento do BD e com um protocolo bem definido.

- Cenário:



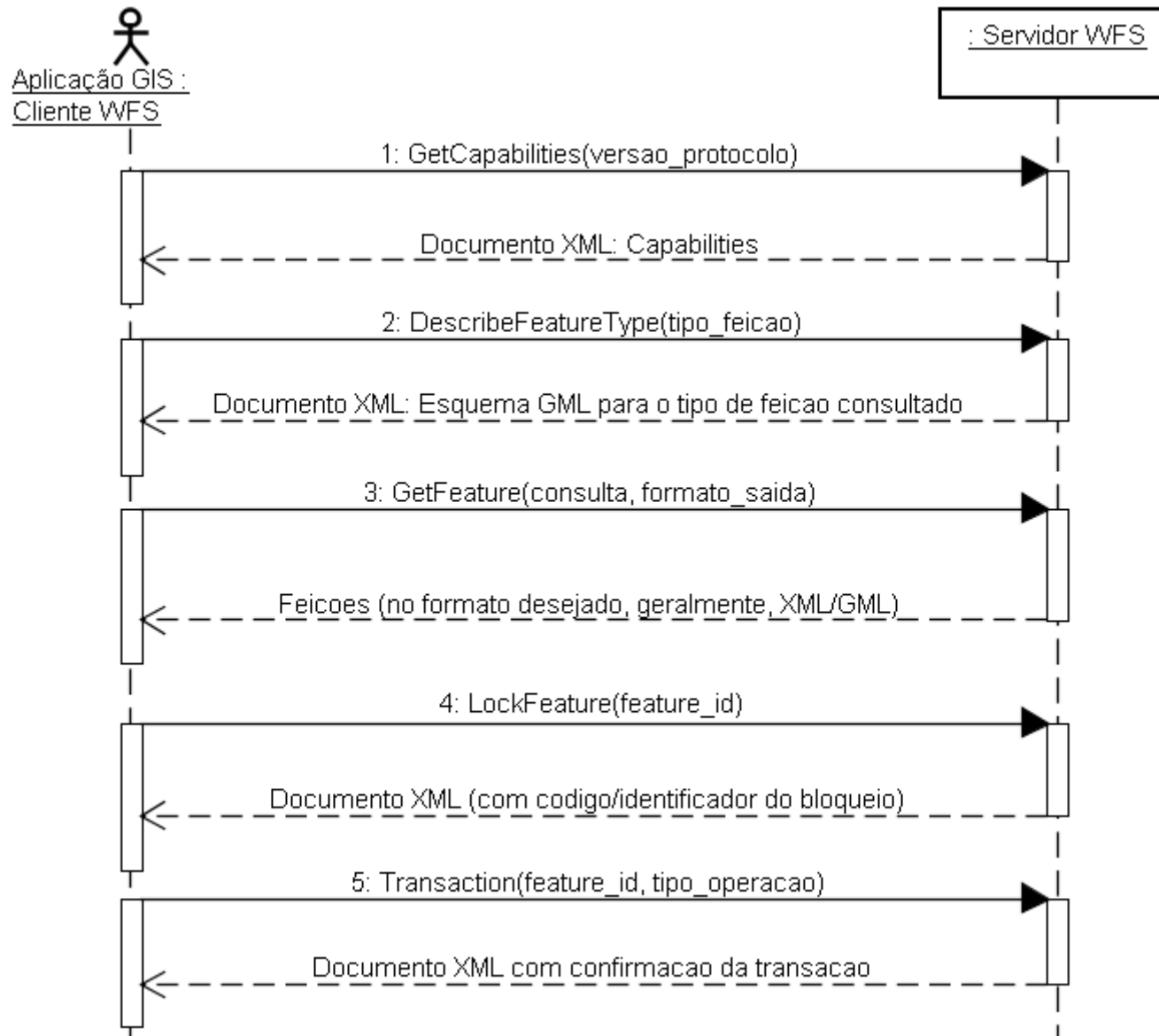


# WFS: Operações Suportadas



- WFS Básico:
  - GetCapabilities:
  - DescribeFeatureType
  - GetFeature
- WFS Transacional (WFS-T):
  - Transaction
  - LockFeature (opcional)

# WFS: Operações Suportadas





# WFS: Requisições



- Podemos enviar requisições via HTTP GET ou POST
- Os parâmetros de uma requisição são expressos em termos de pares (nome/valor) ou XML:
  - No caso de requisições codificadas em XML, não se usa o método GET (usa-se apenas POST)
  - Além disso, faz-se intensivo uso de XML nas requisições



# WFS: Requisições



- Pode-se usar SOAP:
  - O cliente simplesmente codifica a requisição WFS no conteúdo do elemento `<soap:Body>`, na mensagem de requisição





# WFS: GetCapabilities

Parâmetro	Valores	Obrigatório
VERSION	1.1.0	Não
REQUEST	GetCapabilities	Sim
SERVICE	WFS	Sim

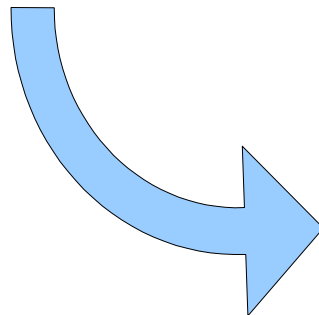
- Solicitando o documento Capabilities de um servidor WFS:
  - `http://localhost:8080/geoserver/wfs?`  
**VERSION=1.0.0&**  
**SERVICE=WFS&**  
**REQUEST=GetCapabilities**

[Ver no Navegador](#)

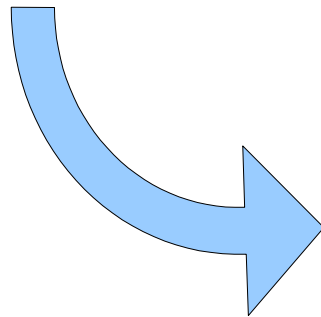
- Solicitando o esquema (GML) do tipo Municípios Brasileiros (IBGE, 2000):

```
testel=# \d mapa_municipios
      Table "public.mapa_municipios"
  Column      |      Type      |
  -----+-----+
  NOME_UF      | character varying(23) |
  REGIÃO       | character varying(15) |
  ...          | ...                |
  N_MUNIC1     | character varying(37) |
  COD_UF       | character varying(10) |
  POP_2000     | integer            |
  the_geom     | geometry           |
```

**Descrição da tabela de atributos**



- Solicitando o esquema (GML) do tipo Municípios Brasileiros (IBGE, 2000):
  - `http://localhost:8080/geoserver/wfs?`  
**SERVICE=WFS&**  
**VERSION=1.0.0&**  
**REQUEST=DescribeFeatureType&**  
**typeName=municipios:municipios\_2000**



[Ver no Navegador](#)

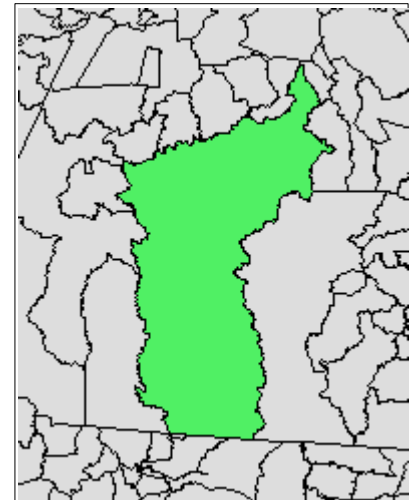
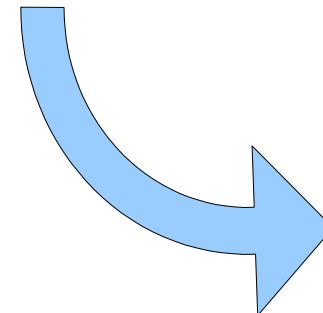
```
...  
<complexType name="municipios_2000_Type">  
  <complexContent>  
    <extension base="gml:AbstractFeatureType">  
      <sequence>  
        <element name="the_geom"  
          type="gml:MultiPolygonPropertyType"/>  
        <element name="POP_2000" type="long"/>  
        <element name="N_MUNIC1">  
          <simpleType>  
            <restriction base="string">  
              <maxLength value="37"/>  
            </restriction>  
          </simpleType>  
        </element>  
      </sequence>  
    </extension>  
  </complexContent>  
</complexType>  
...
```

# WFS: GetFeature

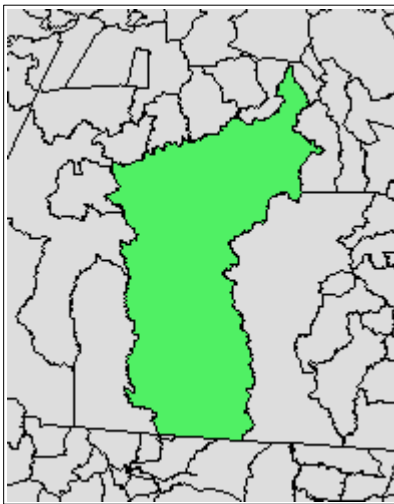
- De posse do esquema do tipo Municípios Brasileiros (IBGE, 2000), podemos solicitar feições (uma ou mais):

- `http://localhost:8080/geoserver/wfs?`  
**`SERVICE=WFS&`**  
**`VERSION=1.0.0&`**  
**`REQUEST=GetFeature&`**  
**`typeName=municipios:municipios_2000&`**  
**`maxFeatures=1&`**  
**`PropertyName=*&`**  
**`FILTER=<Filter>`**  
**`<PropertyIsEqualTo>`**  
**`<PropertyName>N_MUNIC1</PropertyName>`**  
**`<Literal>ALTAMIRÃ</Literal>`**  
**`</PropertyIsEqualTo>`**  
**`</Filter>`**

[Ver no Navegador](#)



- De posse do esquema do tipo Municípios Brasileiros (IBGE, 2000), podemos solicitar feições (uma ou mais):



```
...
<gml:featureMember>
  <municipios:municipios_2000 fid="municipios_2000.233">
    <municipios:the_geom>
      <gml:MultiPolygon ...">
        <gml:polygonMember>
          <gml:Polygon>
            <gml:outerBoundaryIs>
              <gml:LinearRing>
                <gml:coordinates ...>
                  -51.9035,-3.5902221
                  -51.903111,-3.5993586...
                </gml:coordinates>
              </gml:LinearRing>
            </gml:outerBoundaryIs>
          </gml:Polygon>
        </gml:polygonMember>
      </gml:MultiPolygon>
    </municipios:the_geom>
    <municipios:NOME_UF>PARA</municipios:NOME_UF>
    <municipios:N_MICRO>ALTAMIRA</municipios:N_MICRO>
  </municipios:municipios_2000>
</gml:featureMember>
...
```

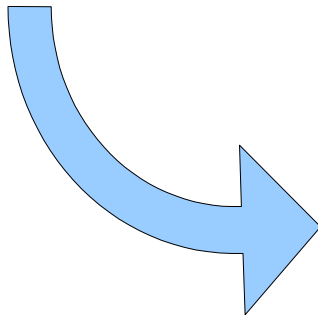
**Documento XML/GML  
2.1.2 com o resultado da  
consulta**



# WFS: LockFeature

- Bloqueando o município de altamira, para realizar uma alteração:
  - `http://localhost:8080/geoserver/wfs?`  
**SERVICE=WFS&**  
**VERSION=1.0.0&**  
**REQUEST=LockFeature&**  
**typeName=municipios:municipios\_2000&**  
**FEATUREID=municipios\_2000.233**

[Ver no Navegador](#)



```
<WFS_LockFeatureResponse xsi:schemaLocation="...">  
  <LockId>GeoServer_1d49ea30062bc95d</LockId>  
</WFS_LockFeatureResponse>
```

**Retorno da Operação LockFeature**

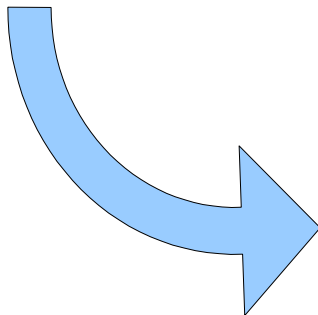


# WFS: Transaction

- Eliminando o município de Altamira do mapa de municípios:
  - `http://localhost:8080/geoserver/wfs?`  
**SERVICE=WFS&**  
**VERSION=1.0.0&**  
**REQUEST=Transaction&**  
**Operation=Delete&**  
**typeName=municipios:municipios\_2000&**  
**FEATUREID=municipios\_2000.233&**  
**RELEASEACTION=ALL&**  
**LockId=GeoServer\_1d49ea30062bc95d**

[Ver no Navegador](#)

Retorno da Operação Transaction



```
<wfs:WFS_TransactionResponse ...>
  <wfs:TransactionResult>
    <wfs:Status><wfs:SUCCESS/></wfs:Status>
  </wfs:TransactionResult>
</wfs:WFS_TransactionResponse>
```



# WFS: GetFeatureWithLock



- Similar a GetFeature
- No entanto, indica ao servidor WFS para bloquear as instâncias selecionadas, pois possivelmente haverá uma atualização em operações futuras
- O documento resultante conterá o valor da chave de bloqueio





# WFS: GetGmlObject



- Esta operação permite recuperar feições e elementos através do ID (gml:id)
- Interface pela qual o WFS pode ser consultado usando Xlinks
- Interface opcional

- Exceptions:
  - Caso o servidor WFS receba uma requisição incorreta ou ocorra algum erro no processamento de algum comando, ele deve lançar uma exceção
  - Uma exceção é um documento que possui um formato específico:

```
<?xml version="1.0" ?>
<ExceptionReport
version="1.1.0"
xmlns="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="owsExceptionReport.xsd">
  <Exception code="999" locator="INSERT STMT 01">
    <ExceptionText>parse error: missing closing tag for element wkbGeom
  </ExceptionText>
</Exception>
</ExceptionReport>
```



# WFS: Considerações



- O servidor e o cliente negociam a versão do protocolo através da interface GetCapabilities
- É possível restringir os tipos de operações sobre cada tipo de feição (inserção, atualização, remoção, criação, bloqueio...)
- “Todos” os aspectos de controle do servidor estão definidos:
  - Número máximo de feições que um cliente pode solicitar, se o servidor suporta SOAP, ...



# WFS: Considerações



- Namespaces normativos:
  - <http://www.opengeospatial.net/wfs>:
    - para o vocabulário da interface WFS
  - <http://www.opengeospatial.net/gml>:
    - para o vocabulário GML
  - <http://www.opengeospatial.net/ogc>:
    - para o vocabulário do Filter



# WFS: Considerações



- Atributos complexos podem ser consultados usando XPath
- O servidor pode retornar a resposta em outros formatos:
  - GML é obrigatório
  - Outros formatos poderiam ser: KML, SHP, SHP-zip, GML-zip
- Atributo *featureVersion*:
  - Previsto no elemento <Query>, pode ser usado para acomodar sistemas capazes de realizar versionamento de feições.



# WFS: Considerações



- Trabalhos Futuros:
  - Será que tipos topológicos da GML3 precisarão de operações especiais, além de get/lock/update?
  - Suporte a operações de merge/update que permitam atualizar as feições sem a necessidade de buscar toda a sua geometria (ex: adicionar vértices em uma geometria)



# WFS: Padrões Relacionados



- Padrões que trabalham de forma complementar:
  - Filter
  - GML
- Padrões W3C:
  - XML
  - XML Esquemas
  - WSDL
  - SOAP
  - XPath
  - Xlink



# WFS: Principais Mentores



- Autor:
  - Panagiotis A. Vretanos (CubeWerx, Inc.)
- Principais Organizações:
  - CubeWerx Inc., Intergraph Corp., IONIC Software
- Colaboradores:
  - Oracle, U.S. Census, NASA, Laser-Scan Ltd, Galdos System Inc., CadCorp Ltd, Polexis, ObjectFX, CSIRO, Social Change Online.





# WFS: Ferramentas



Nome	Tipo
deegree	C/S
GeoServer	S
MapServer	C/S
MapBender	C
Gaia	C
MapBrowser	C
Cadcorp SIS Map Server	S
Udig 1.1	C
JUMP 1.1.2 (Unified Mapping Plataform)	C



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Web Coverage Service – WCS



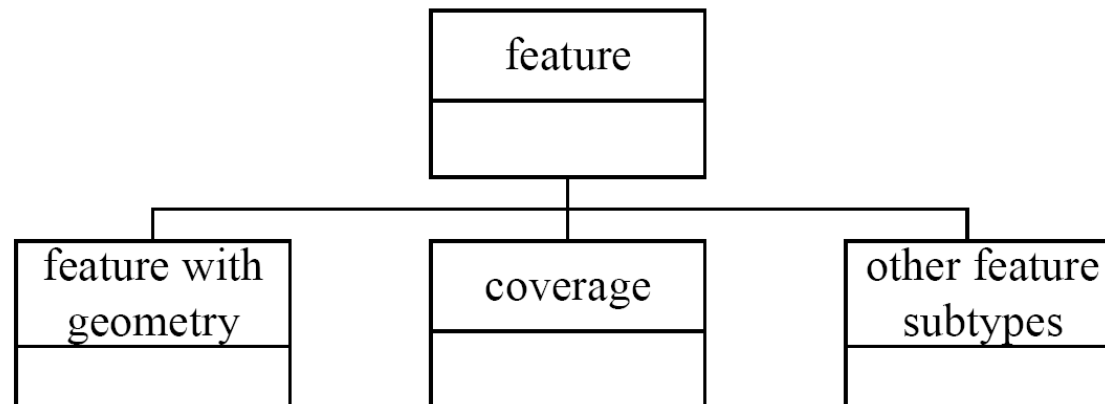
# WCS: Web Coverage Service



- Definição:
  - Web Service descrito pelo OGC, cujo objetivo é obtenção e troca de informações espaciais sob a forma de coverages.
- Retorna o dado com sua semântica original
- [Versões da Especificação:](#)
  - 1.0.0: 27/08/2003, 03-065r6, 67 páginas
  - 1.1.0: 17/10/2006, 06-083r8, 143 páginas

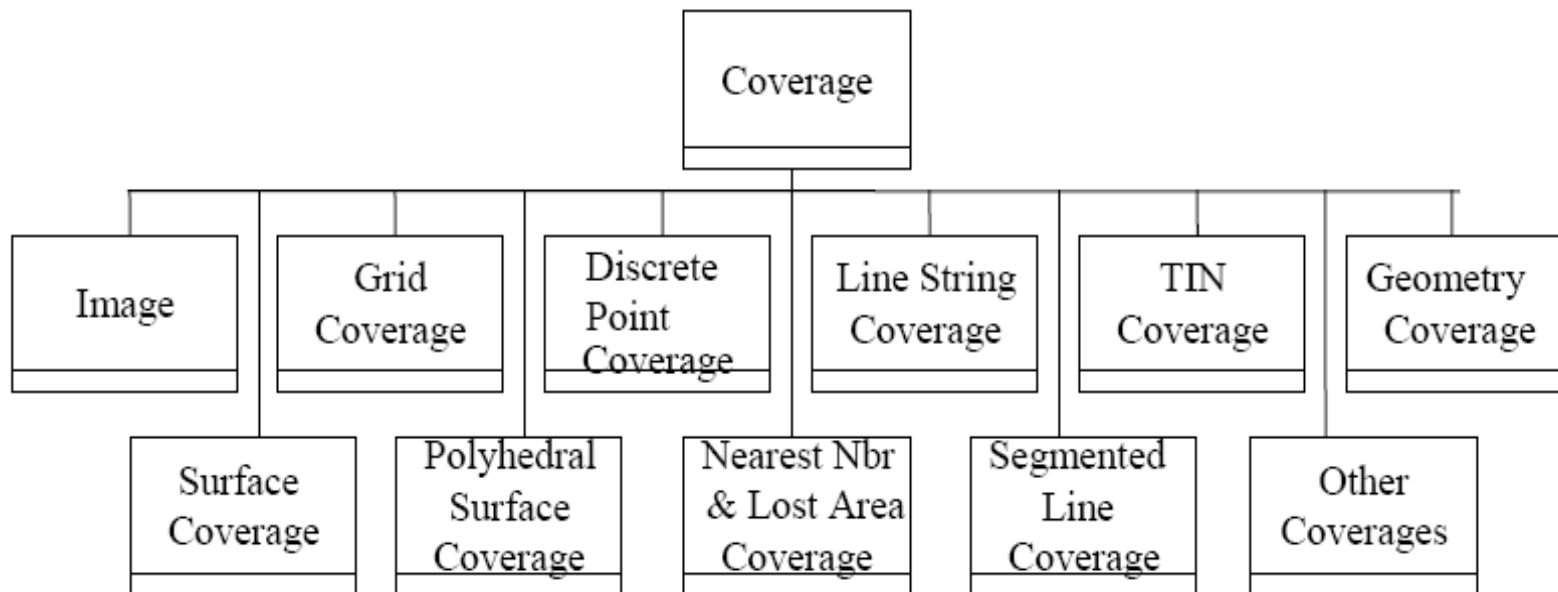
# WCS: Coverages

- Representação espacial que “cobre” uma área geográfica, divide a mesma em partições que podem ser regulares e irregulares, e designa um valor para cada partição.
- É um tipo de feature



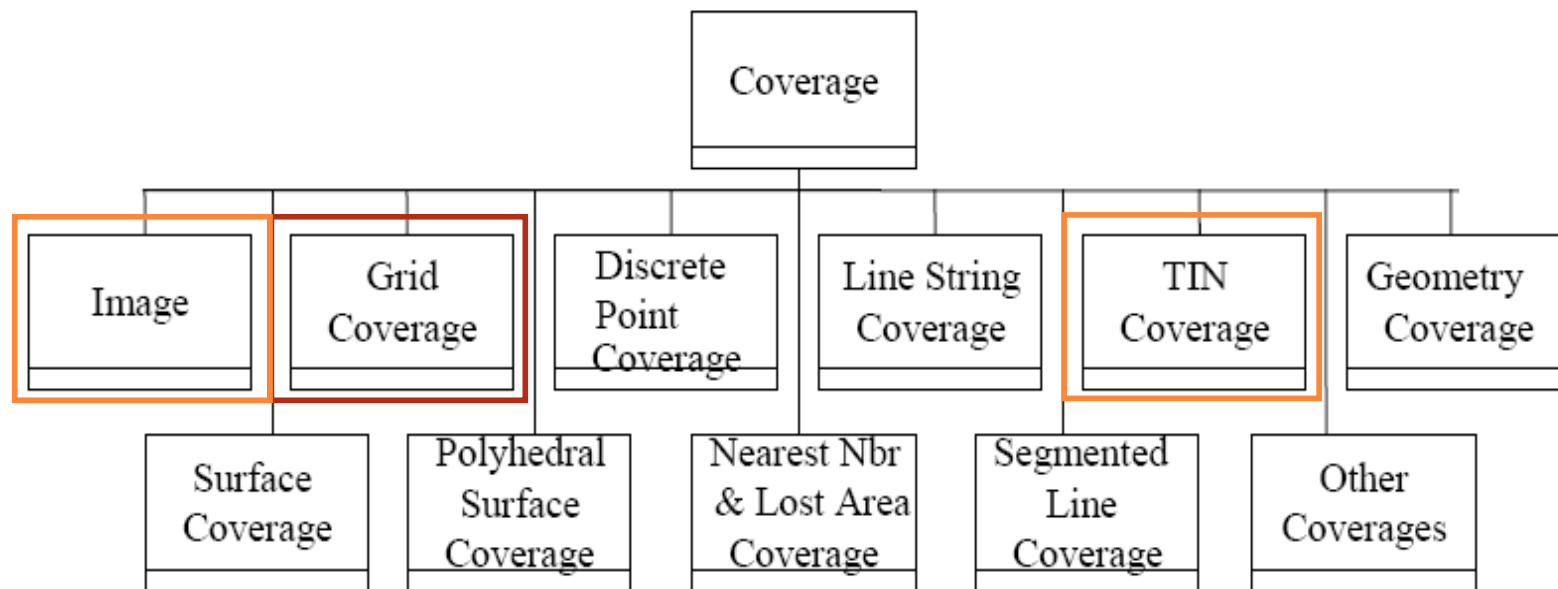
# WCS: A Coverage do OGC

- A coverage definida pelo OGC possui os seguintes sub-tipos.



# WCS: A Coverage do WCS

- A coverage usada no WCS até a versão atual é basicamente do tipo Grid. Imagens e grades TIN podem também ser pensadas como grids.



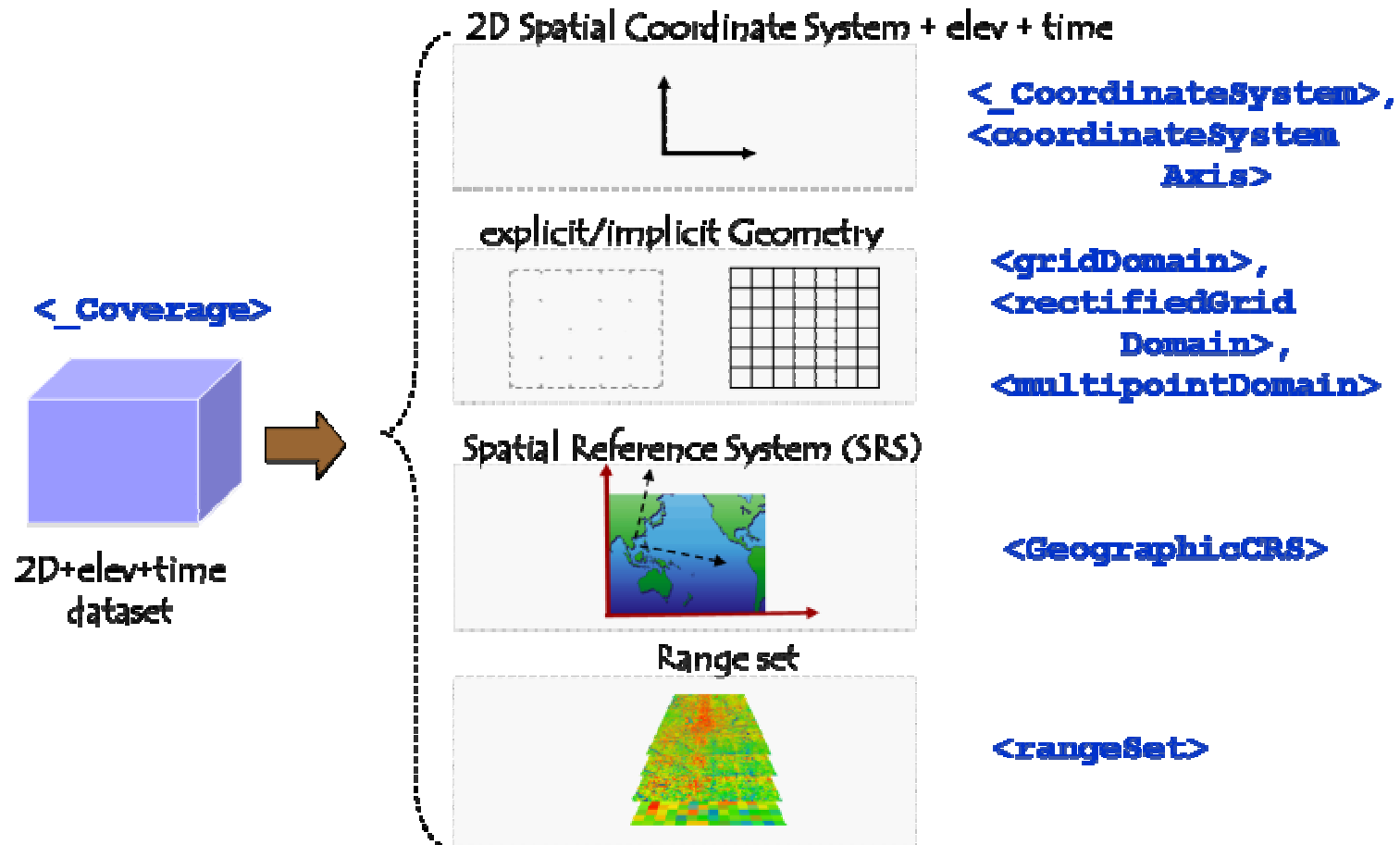


# WCS: A Coverage do WCS



- Grid coverages cujo domínio compreende localizações regularmente espaçadas e, opcionalmente, um componente tempo, o qual pode ser regular ou irregularmente espaçado:
  - Imagem de Satélite
  - Matriz de elevação digital
  - Mapa de temperatura
- Esse documento especifica coverages:
  - Open GIS Consortium. Topic 6: The coverage type and its subtypes

# WCS: A Coverage do WCS







# WCS: Aplicação



- Cenário:
  - “Tenho um grande banco de dados geográficos com imagens de desmatamento. Gostaria de permitir o acesso a esses dados a outros pesquisadores, que poderiam, por exemplo, estudar uma forma de detectar automaticamente padrões de desmatamento nestas imagens”.
  - Disponibilizando um servidor WCS como mediador para este banco de dados seria possível que os pesquisadores tivessem acesso a esses dados, independente do formato de armazenamento do BD e com um protocolo bem definido.



# WCS: Aplicação



- Visualização dos dados por meio de um cliente
- Obtenção de dados espaciais que representem fenômenos de distribuição contínua (coverages);
- Entrada em modelos científicos mais complexos, dado que as informações podem ser interpretadas, e não só retratadas, como no WMS.

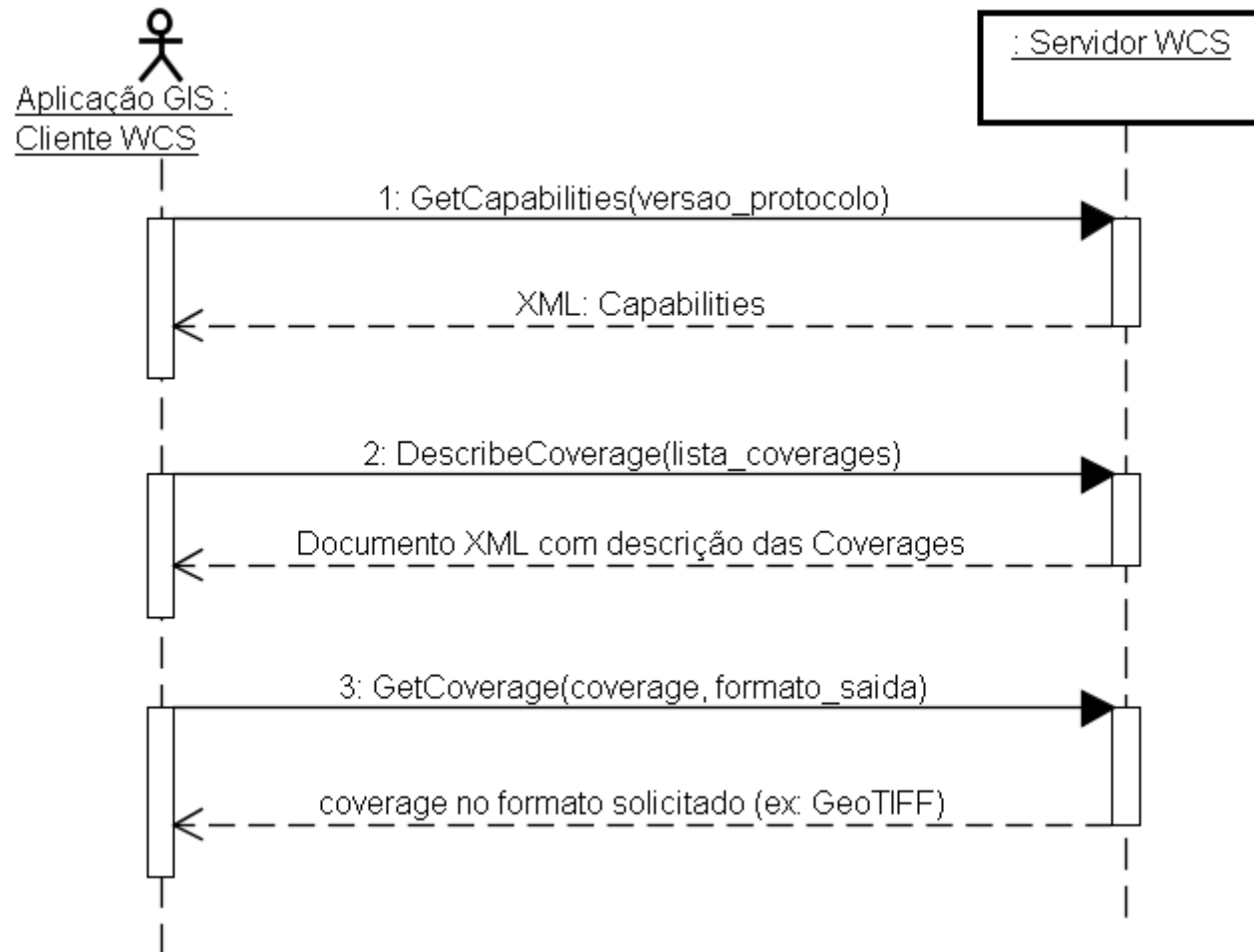


# WCS: Operações Suportadas



- GetCapabilities:
  - Xml com os metadados do serviço e dos layers (coverages) oferecidos
- DescribeCoverage:
  - Xml detalhado da coverage
- GetCoverage:
  - O dado propriamente dito, no formato especificado

# WCS: Operações Suportadas





# WCS: GetCapabilities



- É dividido em três sessões:
  - Service
    - Metadados do servidor
  - Capability
    - Descreve, entre outros, os requests que o serviço suporta, os formatos em que as exceções serão enviadas
  - ContentMetadata
    - Descreve minimamente as coverages disponíveis no servidor
- A requisição pode pedir uma das partes separadamente, ou o capabilities completo.



# WCS: GetCapabilities



- Parâmetros de Entrada

- Request

- Que deve ser igual a “GetCapabilities”

- Version

- 0.7
    - 1.0
    - 1.0.0

```
URL?REQUEST=GetCapabilities  
&VERSION=1.0.0  
&SERVICE=WCS  
&SECTION=/WCS_Capabilities/Service
```

- Service

- Que deve ser igual a “WCS”

- Section

- Opcional

- /WCS\_Capabilities/Service
      - /WCS\_Capabilities/Capability
      - /WCS\_Capabilities/ContentMetadata



# WCS: DescribeCoverage



- Detalha as coverages requeridas, apresentando informações como:
  - a localização no tempo e no espaço
  - os maiores e menos valores nos eixos x e y
  - Projeções suportadas
  - ▶ – Formatos de saída suportados (GeoTiff, HDF-EOS, DTED, NITF, GML)
  - Métodos de interpolação suportados
    - Vizinho mais próximo (default), Bilinear, Bicúbico, Menor área, Baricentro, Nenhuma

Qualquer formato é suportado, mas uma coverage deve estar em pelo menos um desses formatos.



# WCS: DescribeCoverage



- Parâmetros de Entrada

- Request

- Que deve ser igual a “DescribeCoverage”

- Version

- 0.7
    - 1.0
    - 1.0.0

```
URL?REQUEST=DescribeCoverage  
&VERSION=1.0.0  
&SERVICE=WCS  
&COVERAGE=coverage1,coverage2...,  
coverageN
```

- Service

- Que deve ser igual a “WCS”

- Coverage

- Lista das coverages requeridas para a operação, separadas por vírgula





# WCS: GetCoverage



- Retorna a Coverage requerida, no formato especificado
- A consulta pode ser feita apenas pelo retângulo envolvente ou pelo intervalo de tempo. Não está previsto o uso de filtros ou máscaras nessa especificação.



# WCS: GetCoverage



- Parâmetros de Entrada Obrigatórios
  - Request: Que deve ser igual a “GetCoverage”
  - Version
  - Service: Que deve ser igual a “WCS”
  - Coverage: Uma coverage apenas
  - CRS
  - Retângulo Envolvente(BBOX) ou Intervalo de tempo(Time)
  - Largura (Width), Altura(Height) e Profundidade (Depth - opcionalmente) ou ResX, ResY e ResZ(opcionalmente)
  - Formato de Saída (Format)



# WCS: GetCoverage



- Parâmetros de Entrada Opcionais
  - Response\_CRS
  - Parameter
    - Apenas para dados “compostos”, como imagem multiespectral
    - Ex: `PARAMETER=band1, band2, band5`
    - Interpolation
    - Exceptions

```
URL?REQUEST=GetCoverage
&VERSION=1.0.0&SERVICE=WCS
&COVERAGE=coverage1
&CRS=EPSG:32611
&BBOX=-116.9411,32.7485,
-116.4430,32.9745
&Format=GeoTIFF
```



# WCS: Endereço de Servidores



- <http://laits.gmu.edu/cgi-bin/NWGISS/NWGISS?>
- <http://laits.gmu.edu/cgi-bin/NWGISS/WCS1020?>
  - MODIS
  - ASTER
- <http://geobrain.laits.gmu.edu/cgi-bin/gdalwcs/gdalwcs?>
  - Global Landsat



# WCS: Considerações



- Na atual versão (1.0.0), o WCS encontra-se limitado à descrição e solicitação de coberturas do tipo simples;
- Em cada localização, se encontra ou um único valor (elevação), ou um conjunto de valores definidos de forma homogênea (reflectâncias electromagnéticas)
- Uma versão para WCS com suporte a WCS e SOAP está sendo especificada e encontra-se em sua versão 0.1.0.



# WCS: Padrões Relacionados



- Padrões que trabalham de forma complementar:
  - GML
  - SLD
- Padrões W3C:
  - XML
  - XML Esquemas
  - WSDL
  - SOAP
  - XPath
  - Xlink



# WCS: Principais Mentores



- Autor:
  - John D. Evans (NASA)
  - Arliss Whiteside (BAE Systems E&IS)
- Principais Organizações:
  - CubeWerx Inc., Intergraph Corp., IONIC Software, Oracle, NASA, Galdos System Inc., PCI Geomatics



# WCS: Ferramentas Certificadas



- Ionic (Ionic Image Archive)
  - <http://www.ionicssoft.com/ioniciasclient/homePage.do#target1>
- Degree (Degree WCS)
  - <http://deegree.sourceforge.net/src/demos.html#wcs>
- Luciad (LuciadMap 1.0)
  - [http://www.luciad.com/html/products/prod\\_map.htm](http://www.luciad.com/html/products/prod_map.htm)
- Geomatica (WebServer 1.0)
  - <http://www.pcigeomatics.com/products/webserver.html>





# WCS: Ferramentas Não Certificadas



- gvSIG
- MapServer



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Styled Layer Descriptor – SLD



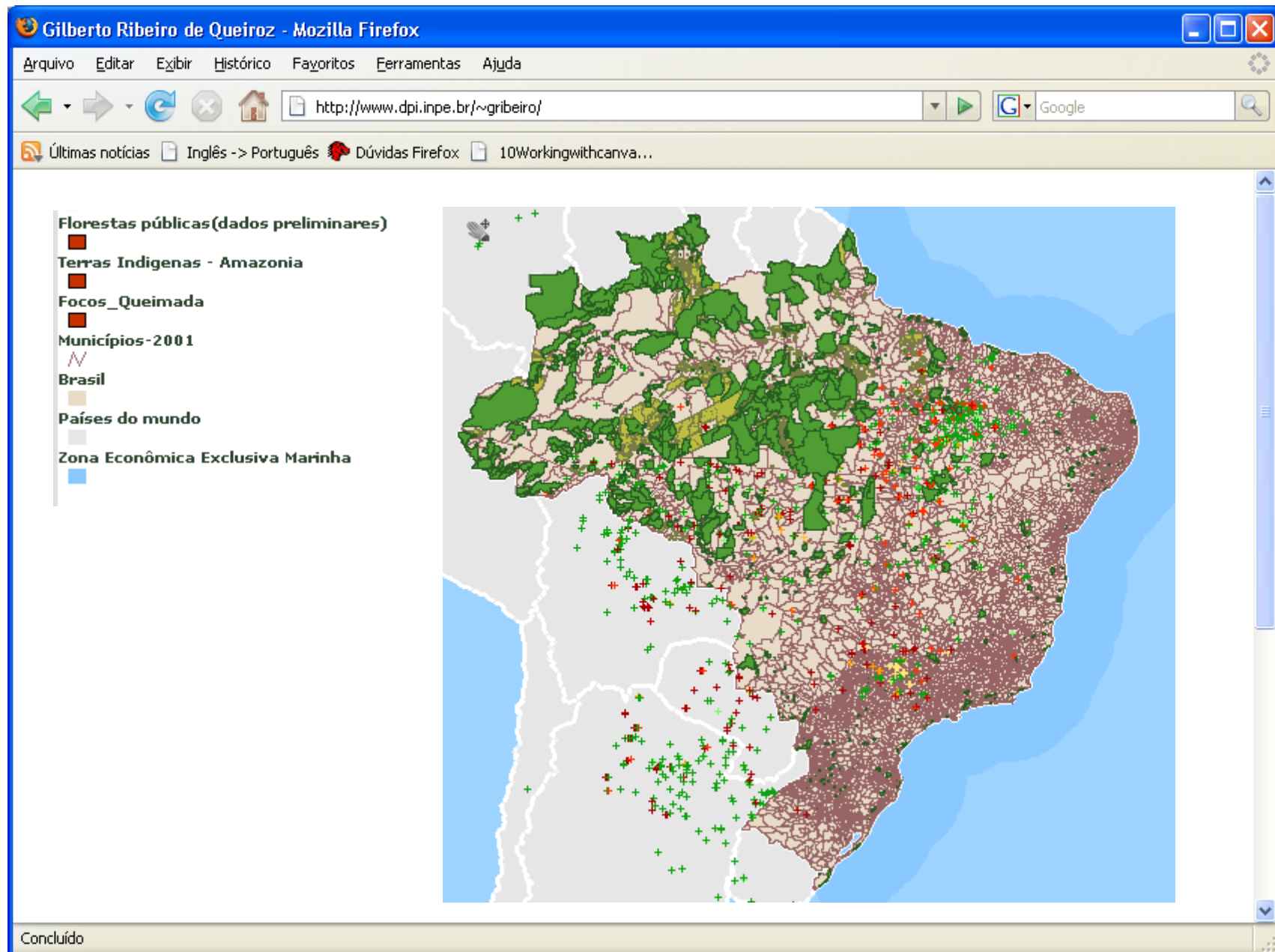
# SLD: Styled Layer Descriptor

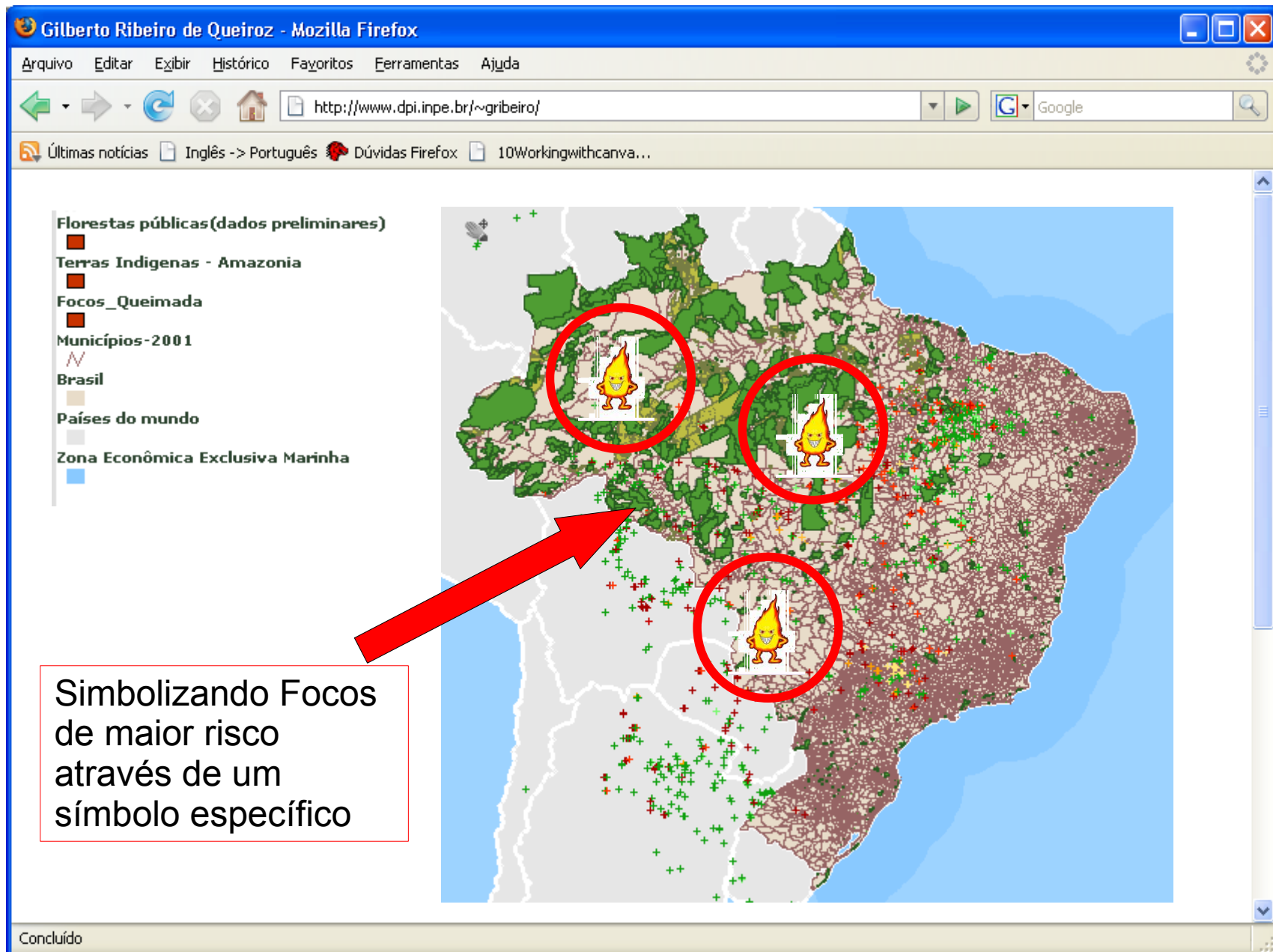


- Definição:
  - Formato de uma linguagem de estilos para produção de mapas;
  - A idéia é permitir o controle do visual dos mapas produzidos por um serviço WMS
- [Versões da Especificação:](#)
  - 1.0.0: 19/09/2002, 02-070, 117 páginas
  - 1.1.0 (draft): 19/10/2005, 05-078, 54 páginas
    - Complemento: Symbology Encoding (63 páginas)



# SLD: Aplicação







# SLD: Named Layers e Styles



- O WMS (padrão) lida com estilos e layers conhecidos e identificados por nomes:
  - `http://yourfavoritesite.com/WMS?VERSION=1.1.0&REQUEST=GetMap&BBOX=0.0,0.0,1.0,1.0&LAYERS=Rivers,Roads,Houses&STYLES=CenterLine,CenterLine,Outline`
- Esses estilos da especificação WMS são chamados de “named styles” e os layers de “named layers”



# SLD: Named Layers e Styles



- Podemos codificar “named layers” e “named styles” usando SLD:

```
<StyledLayerDescriptor version="1.0.0">
  <NamedLayer>
    <Name>Rivers</Name>
    <NamedStyle>
      <Name>CenterLine</Name>
    </NamedStyle>
  </NamedLayer>
  <NamedLayer>
    <Name>Roads</Name>
    <NamedStyle>
      <Name>CenterLine</Name>
    </NamedStyle>
  </NamedLayer>
  <NamedLayer>
    <Name>Houses</Name>
    <NamedStyle>
      <Name>Outline</Name>
    </NamedStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```



# SLD: Requisições WMS



- Um cliente pode usar SLD da seguinte forma:
  - Referenciando um documento SLD remoto em sua requisição:
    - Este documento deve estar num local acessível pelo servidor WMS
    - `http://yourfavoritesite.com/WMS?VERSION=1.0.5&REQUEST=GetMap&SRS=EPSG%3A4326&BBOX=0.0,0.0,1.0,1.0&SLD=http://myclientsite.com/mySLD.xml&WIDTH=400&HEIGHT=400&FORMAT=PNG`



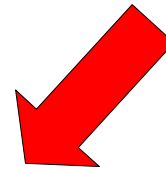


# SLD: Requisições WMS



- Um cliente pode usar SLD da seguinte forma:
  - Incluindo um documento XML/SLD na requisição (parâmetro SLD\_BODY)
  - `http://yourfavoritesite.com/WMS?VERSION=1.0.5&REQUEST=GetMap&SRS=EPSG%3A4326&BBOX=0.0,0.0,1.0,1.0&SLD_BODY=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22UTF8%22%3F%3E%3C!DOCTYPE+StyledLayerDescriptor+SYSTEM+%22http%3A%2F%2Fsom.site.com%2Fsld%2Fsld_072.xsd%22%3E%3CStyledLayerDescriptor+version%3D%221.0.0%22%3E%3CNamedLayer%3E%3CName%3ERivers%3C%2FName%3E%3CNamedStyle%3E%3CName%3ECenterLine%3C%2FName%3E%3C%2FNamedStyle%3E%3C%2FNamedLayer%3E%3CNamedLayer%3E%3CName%3ERoads%3C%2FName%3E%3CNamedStyle%3E%3CName%3ECenterLine%3C%2FName%3E%3C%2FNamedStyle%3E%3C%2FNamedLayer%3E%3CNamedLayer%3E%3CName%3EHouses%3C%2FName%3E%3CNamedStyle%3E%3CName%3EOutline%3C%2FName%3E%3C%2FNamedStyle%3E%3C%2FNamedLayer%3E%3C%2FStyledLayerDescriptor%3E`  
`WIDTH=400&HEIGHT=400&FORMAT=PNG`

Documento SLD  
codificado na URL





# SLD: Requisições WMS



- Um cliente pode usar SLD da seguinte forma:
  - Incluindo um documento XML/SLD na requisição (parâmetro SLD\_BODY) usando POST:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE GetMap SYSTEM "http://some.site.com/wms/GetMap.xsd">
<ogc:GetMap xmlns:ogc="http://www.opengis.net/ows"
xmlns:gml="http://www.opengis.net/gml"
env:encodingStyle="http://www.w3.org/2001/09/soapencoding"
version="1.2.0" service="WMS">
  <StyledLayerDescriptor version="1.0.0">
    <NamedLayer>
      <Name>Rivers</Name>
      <NamedStyle>
        <Name>CenterLine</Name>
      </NamedStyle>
    </NamedLayer>
    ...
    <BoundingBox srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      ...
    </BoundingBox>
    ...
  </ogc:GetMap>
```



# SLD: Integrando WMS, WFS e WCS



- A SLD pode ser usada para integrar um servidor WMS a servidores WFS/WCS:
  - `http://yourfavoritesite.com/WMS?VERSION=1.0.5&REQUEST=GetMap&SRS=EPSG%3A4326&BBOX=0.0,0.0,1.0,1.0&SLD=http://myclientsite.com/mySLD.xml&WIDTH=400&HEIGHT=400&FORMAT=PNG&REMOTE_OWS_TYPE=WFS&REMOTE_OWS_URL=http://anothersite.com/WFS?`



# SLD: “Component” x “Integrated”



- Introduz dois tipos de servidores, no que diz respeito a relação do servidor WMS com servidores WFS e WCS:
  - Component servers:
    - Servidor WMS fracamente acoplado, capaz de simbolizar feições/coverages de qualquer servidor WFS/WCS para o qual ele seja redirecionado.
  - Integrated servers:
    - Servidor WMS fortemente acoplado, capaz apenas de simbolizar feições/coverages de servidores WFS/WCS com o qual ele esteja integrado.



# SLD: Operações



- DescribeLayer
- GetLegendGraphic
- GetStyles
- PutStyles



# SLD: Operação DescribeLayer



- Definição:
  - Obtém informações a respeito das feições que queremos simbolizar (retorno é um XML)
  - Aplica-se a “named layers”
  - Liga os conceitos de layer do WMS com os conceitos de tipos de feição e coverage layer do WFS e WCS, respectivamente
  - Na prática evita que o documento Capabilities fique cada vez mais carregado com informações dos dados!
  - `http://yourfavoritesite.com/WMS?VERSION=1.1.0&REQUEST=DescribeLayer&LAYERS=Rivers,Roads,Houses`

- Exmplo de retorno desta operação:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WMS_DescribeLayerResponse SYSTEM "http://some.site.com/sld/DSR.dtd">
<WMS_DescribeLayerResponse version="1.1.0">
  <!-- 'Layer_A' comes from the wfs specified by
  the prefix "http://www.mywfs.com/WFS?" and has features
  of types 'Road_FT' and 'Route_FT' -->
    <LayerDescription name="Layer_A" wfs="http://www.mywfs.com/WFS?">
      <Query typeName="Road_FT" />
      <Query typeName="Route_FT" />
    </LayerDescription>
  <!-- 'Layer_B' cannot be described in terms of
  a WFS and so has no wfs attribute and no contents -->
    <LayerDescription name="Layer_B">
    </LayerDescription>
</WMS_DescribeLayerResponse>
```

Permite que o cliente invoque  
DescribeFeatureType



# SLD: User-defined Layers



```
<StyledLayerDescriptor version="1.0.0">
  <UserLayer>
    <Name>MyLayer</Name>
    <RemoteOWS>
      <Service>WFS</Service>
      <OnlineResource
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:type="simple"
        xlink:href="http://some.site.com/WFS?" />
      </RemoteOWS>
      <LayerFeatureConstraints>
        <FeatureTypeConstraint>
          <FeatureTypeName>RoadFeatures</FeatureTypeName>
        </FeatureTypeConstraint>
      </LayerFeatureConstraints>
      <UserStyle>
        [...]
      </UserStyle>
    </UserLayer>
  </StyledLayerDescriptor>
```





# SLD: User-defined Styles



```
<StyledLayerDescriptor version="1.0.0">
  <NamedLayer>
    <Name>Transportation</Name>
    <UserStyle>
      <Name>GS1</Name>
      <Title>GeoSym</Title>
      <Abstract>GeoSym style for transportation</Abstract>
      <FeatureTypeStyle>
        <Rule>
          <Filter>...[A = 1]...</Filter>
          <PolygonSymbol> ...[red]... </PolygonSymbol>
        </Rule>
        <Rule>
          <ElseFilter/>
          <PolygonSymbol> ...[gray]... </PolygonSymbol>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```



# SLD: Symbolizers

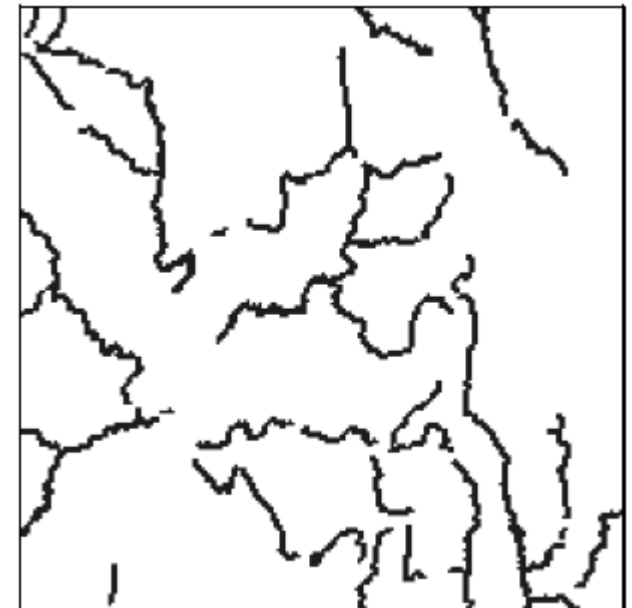


- Elementos presentes dentro das “Rules”
- Descreve como a feição irá “aparecer” no mapa
- Descreve:
  - Formato
  - Propriedades gráficas (cor, opacidade)
- Estão definidos 5 tipos de “symbolizers”:
  - Line
  - Polygon
  - Point
  - Text
  - Raster

# SLD: Symbolizers

```
<LineSymbolizer>
  <Geometry>
    <ogc:PropertyName>centerline</ogc:PropertyName>
  </Geometry>
  <Stroke>
    <CssParameter name="stroke">#0000ff</CssParameter>
    <CssParameter name="stroke-width">2</CssParameter>
  </Stroke>
</LineSymbolizer>
```

Parâmetros derivado  
de SVG/CSS2



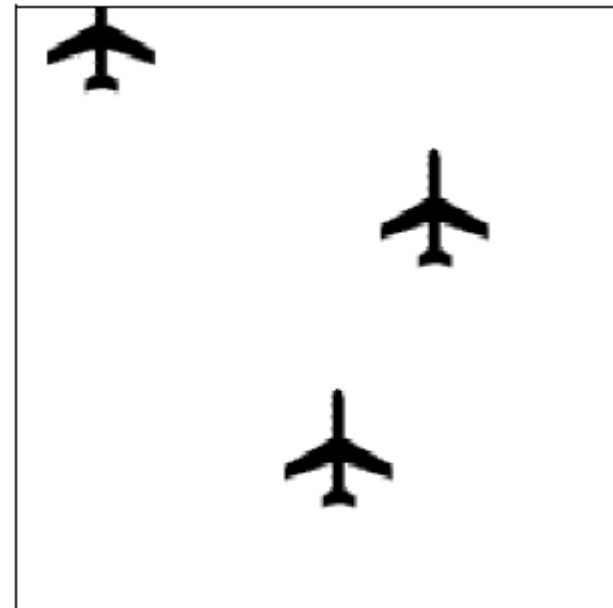
- Pode ser, por exemplo, um GIF ou SVG:

```
<PointSymbolizer>
  <Geometry>
    <ogc:PropertyName>locatedAt</ogc:PropertyName>
  </Geometry>
  <Graphic>
    <Mark>
      <WellKnownName>star</WellKnownName>
      <Fill>
        <CssParameter name="fill">#ff0000</CssParameter>
      </Fill>
    </Mark>
    <Size>8.0</Size>
  </Graphic>
</PointSymbolizer>
```



- Pode ser, por exemplo, um GIF ou SVG:

```
<PointSymbolizer>
  <Graphic>
    <ExternalGraphic>
      <OnlineResource
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:type="simple"
        xlink:href="http://www.vendor.com/geosym/2267.png"/>
      <Format>image/png</Format>
    </ExternalGraphic>
    <Mark/>
    <Size>15.0</Size>
  </Graphic>
</PointSymbolizer>
```





# SLD: Operação GetLegendGraphic



- Definição:
  - Forma de obter a legenda de um estilo
  - Exemplo:
    - `http://www.vendor.com/wms.cgi?VERSION=1.1.0&REQUEST=GetLegendGraphic&LAYER=ROADL_1M%3Alocal_data&STYLE=my_style&RULE=highwaysSLD=http://www.sld.com/styles/kpp01.xml&WIDTH=16&HEIGHT=16&FORMAT=image/gif`



# SLD: Gerenciamento de Símbolos



- GetStyles:
  - Usada para recuperar user-defined styles de um WMS.
- PutStyles:
  - Usada para armazenar user-defined styles e user-defined layers em um WMS
  - Podemos armazenar diversos estilos
  - Se a operação for bem sucedida, os estilos ficarão disponíveis no WMS para uso como “named styles”



# SLD: Considerações



- No documento Capabilities do servidor WMS deve constar informações avisando do suporte a SLD:
  - O cliente pode usar HTTP POST (com SLD embutido) e/ou HTTP GET (com parâmetros SLD ou SLD\_BODY)?
  - O WMS suporta user-defined layers?
  - O WMS suporta user-defined styles?
  - O WMS pode ser 'direcionado' a serviços OWS remotos? (i.e., o WMS suporta os parâmetros REMOTE\_OWS\_SERVICE e REMOTE\_OWS\_URL?)
  - O WMS suporta operação DescribeLayer?





# SLD: Considerações



- A simbolização pode levar em conta a escala:
  - Podemos solicitar o desenho de feições que estejam numa determinada escala
- Podemos ter símbolos a partir de bitmaps ou vetoriais:
  - Previsto informação de rotação
- Com a integração da especificação Filter, dá a possibilidade de fazer, praticamente, **“qualquer mapa”**!



# SLD: Considerações



- Casamento (parcial?) com outros padrões de simbolização:
  - GeoSym
  - MIL-STD-2525B



# SLD: Principais Mentores



- Autor:
  - William Lalonde (CubeWerx Inc.).
- Principais Organizações:
  - CubeWerx Inc (Editor), Syncline, Ionic Software s.a.
- Colaboradores:
  - ESRI, Compusult Ltd., Galdos Systems Inc., Intergraph, Laser-Scan, US Army ERDC



# SLD: Padrões Relacionados



- Padrões complementares:

- WMS
- WFS
- WCS
- Filter

- Padrões não OGC:

- SVG
- XML/XML esquemas



# SLD: Ferramentas



Nome	Tipo
GeoServer	S
MapServer	S



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# OpenGIS Location Services – OpenLS



# OpenLS



- Definição

- Especificação do OGC que define uma plataforma aberta para serviços baseados em localização (LBS)
  - Composto por cinco serviços essenciais
- Esta plataforma também é conhecida por GeoMobility Server (GMS)

- Specs

- Versão 1.1: OGC 05-016, 2005, 174 p.
  - [http://portal.opengeospatial.org/files/?artifact\\_id=8836](http://portal.opengeospatial.org/files/?artifact_id=8836)
- Versão 1.0: OGC 03-006r3, 2002, 66 p.
  - [http://portal.opengeospatial.org/files/?artifact\\_id=3418](http://portal.opengeospatial.org/files/?artifact_id=3418)



# OpenLS: Aplicabilidade



- Caso de uso
  - Uma pessoa quer encontrar restaurantes perto de onde se encontra. Seu celular não tem GPS.
    - Acessa o portal de sua operadora e solicita o serviço
    - O portal redireciona a requisição para um “Guia de restaurantes”
    - O serviço solicita a posição do usuário através de *Gateway Service*
    - Com a posição do usuário, um *Directory Service* é acionado para obter uma lista dos restaurantes a um certo raio dessa posição
    - Com a lista dos restaurantes, o usuário pode solicitar a um *Route Service* que trace o melhor caminho até o escolhido
    - Por fim, ele pede ao *Presentation Service* que o guie até o local de interesse, através de imagens





# OpenLS



- Definições

- Objetivo do OpenLS

- Definir o acesso aos serviços essenciais e aos tipos de dados abstratos (ADT) que compõem o GMS

- Abstract Data Type (ADT)

- Tipos de dados e estruturas bem-formadas para informação sobre localização
    - São implementados na forma de esquemas de aplicação, que são codificados em XLS
    - Tipos: Position ADT, Route ADT, POI ADT e Map ADT

- GeoMobility Server (GMS)

- Plataforma aberta que consiste dos serviços essenciais desenvolvidos sob a iniciativa do OpenLS
    - Oferece funções básicas para a construção de aplicativos LBS



# OpenLS



- Definições

- XML for Location Services (XLS)

- Método para codificar mensagens do tipo *request/response* e ADTs associados para o GMS

- Location-Based Service (LBS)

- Serviço de rede sem fio que usa informação geográfica para servir o usuário móvel
    - Também é conhecido como qualquer serviço de aplicação que explora a posição de um terminal móvel



# OpenLS



- Serviços essenciais (Core Services)
  - Directory Service
  - Gateway Service
  - Location Utility Service
  - Presentation Service
  - Route Service



# OpenLS: Core Services



- Directory Service
  - Definição
    - Serviço que provê acesso a um diretório *on-line* (páginas amarelas), para encontrar a localização de um produto ou serviço específico ou próximo
  - Pinpoint Directory Service
    - Permite aos assinantes encontrar a localização de um lugar, produto ou serviço específico
  - Proximity Directory Service
    - Permite que os assinantes acessem um diretório que informe o lugar, produto ou serviço mais próximo



# OpenLS: Core Services



- Directory Service
  - Exemplo
    - Onde fica o Center Vale Shopping?

```
<DirectoryRequest>  
  <POIProperties directoryType="White Pages">  
    <POIProperty name="POIName" value="Center Vale Shopping"/>  
  </POIProperties>  
</DirectoryRequest>
```



# OpenLS: Core Services



- Directory Service
  - Exemplo
    - Qual o bar (com chopp) mais próximo do HTO/CTA?

```
<DirectoryRequest>
  <POILocation>
    <Nearest>
      <POI ID="1">
        <POIAttributeList>
          <POIInfoList>
            <POIInfo name="POI Name" value="HTO/CTA"/>
          </POIInfoList>
        </POIAttributeList>
      </POI>
    </Nearest>
  </POILocation>
  <POIProperties directoryType="Yellow Pages">
    <POIProperty name="NAICS_type" value="Bar"/>
    <POIProperty name="NAICS_subType" value="Chopperia"/>
  </POIProperties>
</DirectoryRequest>
```



# OpenLS: Core Services



- Gateway Service

- Definição

- Serviço que busca a posição de um terminal móvel conhecido a partir da rede
    - É a interface entre o GMS e o servidor de localização que reside no GMLC ou MPC através do qual os serviços OpenLS obtém a posição do terminal móvel

- Interface modelada de forma similar ao protocolo MLP especificado por OMA 3.0 para SLIS

- Siglas

- GMLS – Gateway Mobile Location Center
    - MPC – Mobile Positioning Center
    - MLP – Mobile Location Protocol
    - OMA – [Open Mobile Alliance](#)
    - SLIS – Standard Location Immediate Service



# OpenLS: Core Services



- Gateway Service
  - Exemplo
    - Qual a posição do equipamento com IP 200.20.120.33?





# OpenLS: Core Services



- Location Utility Service
  - Definição
    - É a combinação de dois serviços: Geocoder e Reverse Geocoder
  - Geocoder Service
    - Serviço que transforma a descrição de uma localização (nome de lugar, rua, código postal), em um descrição normalizada com a geometria GML Point
  - Reverse Geocoder Service
    - Serviço que transforma uma dada posição numa descrição normalizada da localização de uma feição (endereço com ponto), onde o endereço pode ser definido como uma rua, um cruzamento, nome de lugar ou código postal



# OpenLS: Core Services



- Location Utility Service
  - Exemplos
    - Qual a coordenada do portão principal do INPE?
    - Pergunta clássica: onde estou?



# OpenLS: Core Services



- Presentation Service

- Definição

- Serviço que desenha um mapa feito a partir de um mapa base derivado de qualquer dado geoespacial e um conjunto de ADTs como camadas

- Similar ao WMS

- Possui alguns parâmetros extras, como o azimuth em que a imagem de resposta deve ser gerada
    - Tem estilo também (informativo, é trabalho futuro)
    - Manda uma resposta que pode conter um link para o dados ou esse dado codificado como base64 no XML



# OpenLS: Core Services



- Presentation Service
  - Exemplo
    - Onde está o INPE no mapa? (*request*)

```
<PortrayMapRequest>
  <Output width="240" height="240" format="image/png">
    <gml:Envelope>
      <gml:pos>-45.525 -23.125</gml:pos>
      <gml:pos>-45.505 -23.120</gml:pos>
    </gml:Envelope >
  </Output>

  <Basemap filter="Exclude"/> <!-- Todos os Layer disponiveis -->

  <Overlay>
    <Position>
      <gml:Point>
        <gml:pos>-45.514 -23.123</gml:pos> <!-- INPE -->
      </gml:Point>
    </Position>
  </Overlay>
</PortrayMapRequest>
```



# OpenLS: Core Services



- Presentation Service
  - Exemplo
    - Onde está o INPE no mapa? (*response*)

```
<PortrayMapResponse>
  <Map>
    <Content width="240" height="240" format="image/png">
      <URL>http://www.somesite.org/lbs/maps/hgtr837468.png</URL>
    </Content>
    <gml:Envelope>
      <gml:pos>-45.525 -23.125</gml:pos>
      <gml:pos>-45.505 -23.120</gml:pos>
    </gml:Envelope >
  </Map>
</PortrayMapResponse>
```



# OpenLS: Core Services



- Route Service

- Definição

- Serviço que determina rotas de percurso e informação sobre a navegação entre dois pontos (indicados pelo cliente)

- Considerações

- Ponto inicial: normalmente obtido pelo Gateway Service, mas ele pode indicar
    - Ponto final: qualquer localização, como um lugar que ele tem apenas o endereço ou o telefone, ou um ponto obtido a partir do Directory Service
    - O usuário também pode indicar pontos de passagem (mais rápido, mais curto, menos tráfego, mais paisagens etc) e o modo de transporte
    - O *request* leva alguns parâmetros do Presentation Service



# OpenLS: Core Services



- Route Service

- Exemplo

- Como ir de SJ Campos para Campos do Jordão passando pelo mercado de Quiririm?



# OpenLS: Considerações relevantes



- A Spec
  - Apesar de ser bem explicada com vários “casos de uso”, parece que ficou faltando algumas explicações mais claras sobre cada serviço em particular
  - Já vem com um *profile* SOAP (informativo)
- Navigation Service
  - Versão melhorada do Route Service
    - Vem com alguns próprios p/ navegação
  - Hoje está na seção *Deprecated Documents*
  - Spec
    - OpenGIS® Location Services (OpenLS): Part 6 – Navigation Service, Version 0.5.0, OGC 03-007r1
    - [http://portal.opengeospatial.org/files/?artifact\\_id=3840&vers](http://portal.opengeospatial.org/files/?artifact_id=3840&vers)





# OpenLS: Padrões relacionados



- Padrões usados pelo OpenLS
  - GML

- Proprietárias

- Nenhuma “Compliant”, pois não existem testes prontos para o OpenLS

Nome	Ver	Tipo
Autodesk LocationLogic XML Web Services 2	1.0	s
ESRI ArcIMS 9.0, 9.1	1.0 e 1.1	s
ESRI ArcWeb Services 2006	1.0 e 1.1	s
deCarta Drill Down Server Web Services	1.0 e 1.1	s
Geodan Movida 1.3	1.1	c/s
IONIC RedSpider Enterprise 3.4	1.0 e 1.1	c/s
MapInfo Envinsa 4.0	1.1	s

- Livre

- Se existe, está bem escondida



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



**Tendências: WPS e WCPS**

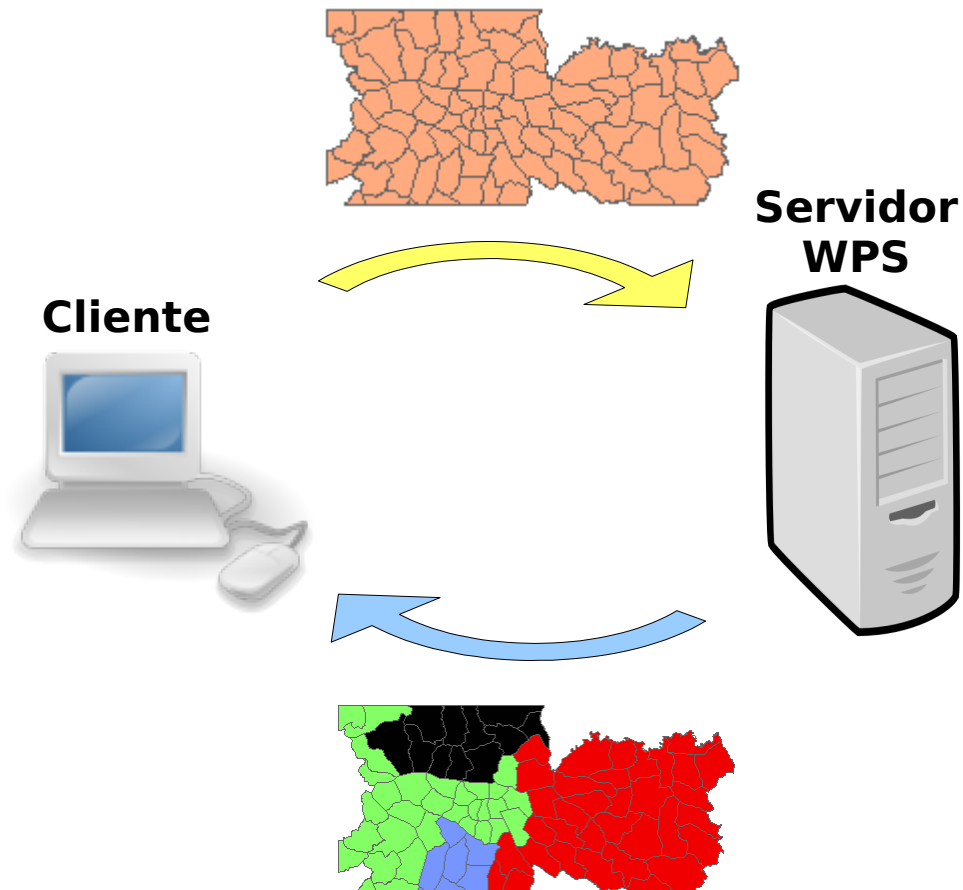


# WPS



- Definição
  - Web Processing Service
  - Serviço que pode oferecer qualquer tipo de funcionalidade SIG para clientes sobre uma rede, inclusive acesso a algoritmos ou modelos computacionais que operem em dados espaciais
  - Originalmente chamado de Geoprocessing Service (OGC 04-043)
- Specs
  - Version: 0.4.0, OGC 05-007r4, 2005, 86 p.
    - É um *Discussion Paper*
    - [http://portal.opengeospatial.org/files/?artifact\\_id=13149](http://portal.opengeospatial.org/files/?artifact_id=13149)

- Caso de uso
  - Regionalizar um dado através do algoritmo SKATER via rede





# WPS



- Considerações

- As possibilidades vão desde uma simples diferença entre atributos até um complicado modelo climático
- Sua interface provê mecanismos para identificar o dado geográfico, iniciar o processamento, e gerenciar a saída do resultado para o cliente
- Dados matriciais e vetoriais
- A spec **não**:
  - Traz quais cálculos serão realizados, mas provê um mecanismo genérico para possibilitar o processamento de geodados em um ambiente de rede
  - Especifica os dados de entrada ou saída
  - Especifica o arquivamento, catalogação, descoberta, ou obtenção da informação gerada pelo WPS



# WPS



- Interface
  - GetCapabilities
  - DescribeProcess
  - Execute



# WPS – Interface



- GetCapabilities

- Definição

- Permite que o cliente receba metadados do servidor, tais como identificação e atividades que podem ser executadas

- Atenção para a seção ProcessOfferings

- Contém uma breve descrição de cada um dos processos oferecidos pelo servidor
    - Não inclui os parâmetros de entrada e de saída
    - É basicamente um: identificador, título, resumo, metadados, versão

- O GetCapabilities do WPS está bem “amarrado” ao padrão OWS (OGC Web Services Common Specification)





# WPS – Interface



- GetCapabilities
  - Exemplo da seção ProcessOfferings

```
<ProcessOfferings>
  <Process processVersion="1.0">
    <ows:Identifier>buffer</ows:Identifier>
    <ows:Title>Buffer a polygon feature</ows:Title>
    <ows:Abstract>Buffer the polygon coordinates found in one
      GML stream by a given buffer distance, and output the results in
      GML.</ows:Abstract>
    <ows:Metadata xlink:title="buffer" />
    <ows:Metadata xlink:title="polygon" />
  </Process>
</ProcessOfferings>
```



# WPS – Interface



- DescribeProcess

- Definição

- Permite que o cliente obtenha informações mais detalhadas sobre uma ou mais operações que o servidor pode executar, tais como parâmetros de entrada e de saída

- Possíveis estruturas de dados

- Complex

- GMLs e arquivos complexos

- Literal

- Qualquer dado que pode ser representado numa string simples

- BoundingBox

- Retângulo envolvente

- Como não há limitações para tipos de serviços, a identificação acontece pela sua descrição



# WPS – Interface



- Execute

- Definição

- Permite que o cliente efetivamente execute o processamento desejado, desde que forneça os parâmetros necessários, recebendo a resposta pedida

- Considerações

- Pode enviar o resultado direto, como também pode informar ao cliente a situação do processamento (status URL)
    - No *request* ele pode especificar se o dado será armazenado ou não no servidor (o padrão é enviar, *store=false*), e se vai querer receber o status do processamento
    - Se *store==false*, então manda o dado direto para o cliente
    - Se *store==true*, então precisa armazenar o dado num formato XML próprio, o “ExecuteResponse” (inclusive as imagens devem ser codificadas no XML), e passa uma URL onde o dado de saída pode ser obtido



# WPS – Interface



- Execute
  - Exemplo de *request*

```
<Execute>
  <ows:Identifier>Buffer</ows:Identifier>
  <DataInputs>
    <Input>
      <ows:Identifier>InputPolygon</ows:Identifier>
      <ows:Title>Playground area</ows:Title>
      <ComplexValueReference ows:reference="http://foo.bar/some_WFS_request.xml"/>
    </Input>
    <Input>
      <ows:Identifier>BufferDistance</ows:Identifier>
      <ows:Title>Distance which people will walk to get to a playground</ows:Title>
      <LiteralValue uom="meters">400</LiteralValue>
    </Input>
  </DataInputs>
  <OutputDefinitions>
    <Output>
      <ows:Identifier>BufferedPolygon</ows:Identifier>
      <ows:Title>Area serviced by playground.</ows:Title>
      <ows:Abstract>Area within which most users of this playground will live.
      </ows:Abstract>
    </Output>
  </OutputDefinitions>
</Execute>
```



# WPS – Considerações relevantes



- OWS-4 Version: 0.9.1, OGC 06-182r1 (DP)
  - Definição
    - Documento que revê o material discutido durante o projeto OWS-4, descreve os WPSs desenvolvidos no trabalho, e oferece sugestões para que a OGC avance com o WPS
    - [http://portal.opengeospatial.org/files/?artifact\\_id=19424](http://portal.opengeospatial.org/files/?artifact_id=19424)
  - Sugerem uma série de melhorias (exemplos):
    - O WPS deveria considerar o uso de WSRF (WS Resource Framework)
    - Colocar mais algumas operações
  - Uma das premissas do projeto OWS-4 é estudar e melhorar o uso de WSDL em serviços OGC
    - Não há consenso sobre o uso de mecanismos SOAP e esta questão continua em aberto
  - Esperem a WPS 1.0 **para este ano**



# WPS – Considerações finais



- Padrões usados pelo WPS
  - OWS
- Ferramentas
  - Ainda não é uma Spec, logo não há cadastros no OGC
  - Algumas pessoas já estão fazendo os seus



# WCPS



- Definição

- Web Coverage Processing Service
- Suporta obtenção e processamento de dados espaciais na forma de *coverages*
  - **Coverage** → informação geoespacial representando fenômenos que variam no espaço, atualmente limitado a grades espaçadas regularmente

- Specs

- Version: 0.0.3, OGC 06-035r1, 2006, 67 p.
  - É um *Discussion Paper*
  - [http://portal.opengeospatial.org/files/?artifact\\_id=14022](http://portal.opengeospatial.org/files/?artifact_id=14022)



# WCPS – Aplicabilidade



- Caso de uso
  - Responder a uma consulta do tipo:
    - Eu quero uma imagem de satélite CBERS2-CDD, banda 5, órbita/ponto x-y, com os valores variando de 0 a 255 (histograma)





# WCPS



- Considerações
  - É baseado no modelo conceitual do WCS
  - Inclui as funcionalidades do WCS e as estende



# WCPS



- Interface
  - GetCapabilities
  - ProcessCoverage



# WCPS – Interface



- GetCapabilities
  - Idêntico ao do WCS, com as seguintes exceções:
    - Nomes de conjuntos de coverages, tipos de eixos e formato dos dados fazem parte do servidor, não da *coverage*



# WCPS – Interface



- ProcessCoverage
  - Definição
    - Essa operação estende o GetCoverage do WCS para permitir capacidade de processamento mais poderosa
  - Os clientes podem escolher se usam o sistema de coordenadas baseado nas células ou no terreno
  - Segue uma loooonga lista das expressões que podem ser usadas
  - O WCPS irá executar cada expressão, na ordem, uma de cada vez. Ex:
    - Pegue os dados da banda 3 da imagem tal, então some 10 p/ cada pixel



# Considerações relevantes



- A Spec
  - Meio “nebulosa”: carece de exemplos
- Trabalho futuro
  - Já prevê uma simplificação com o WPS
  - Provavelmente será uma perfil de aplicações WPS
- Quem fez
  - International University Bremen
  - rasdaman GmbH
  - Traduzindo: só um pequeno grupo abraçou a idéia



# WCPS – Considerações finais



- Padrões usados pelo WCPS
  - WCS
  - GML
- Ferramentas
  - Não é padrão (e pelo visto não vai ser), logo não tem ferramentas certificadas
  - Pelo visto, só o [Rasdaman](#) mesmo



**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Catalogue Services - CS



# Catálogos



- Catálogos ajudam usuários ou aplicações encontrarem informações localizadas em qualquer ambiente computacional distribuído.
- Publicam metadados e provêem mecanismos para consulta e recuperação de informações de repositórios distribuídos .
- Catálogo geoespacial atua sobre informações geográficas.





# Catálogos



- Catálogos não armazenam o dado geográfico em si, mas possuem uma referência para o seu local de armazenamento.
- São a estratégia mais simples de compartilhamento de dados geográficos entre aplicações



# Metadados



- Metadados descrevem os dados, com relação a seu conteúdo, localização, qualidade, criadores, disponibilidade, entre outros.
- Em geral, são descritos como dados sobre dados.
- Metadados para dados geográficos são chamados metadados espaciais.



# OGC Catalogue Services - CS



- Definição:
  - *Framework* que regulamenta um esquema específico para catálogos, incluindo a linguagem de consulta, o formato do conjunto de resultados e como o catálogo deve ser atualizado.
  - Define interfaces que devem estar disponíveis em aplicações de catálogo de metadados sobre dados e serviços geográficos.
- Versões da Especificação:
  - **1.0.0:** 1999, 09-51s, 187 páginas
  - **1.0.1:** 13/12/2002, 02-87r3, 231 páginas
  - **2.0.1:** 20/05/2005, 04-21r3, 205 páginas



# Modelo Abstrato do Catálogo



- Sintaxe para requisitar catálogos;
  - query language
- Lista dos metadados consultáveis;
  - Atributos que podem ser usados para realizar consultas, como título, formato, box;
- Lista mínima dos metadados retornados pelas consultas;
  - Atributos como título, data, formato, fonte,...



# OGC\_Common Catalogue Query Language - CQL



- Linguagem de consulta de metadados baseada na sintaxe SQL, especialmente na cláusula 'WHERE';
- Essa linguagem suporta consultas booleanas, operações de casamento textual, tipos de dados temporais e operadores geoespaciais;
- Toda especificação é extensível para acomodar heterogeneidade de ambientes e variedade de áreas de conhecimento;



# OGC\_Common Catalogue Query Language - CQL



- Está definida na própria especificação do catálogo;
- Usa o OGC Filter;

```
<csw:Query typeName="csw:dataset">
  <csw:ElementSetName>full</csw:ElementSetName>
  <csw:Constraint version="1.0.0">
    <ogc:Filter>
      <ogc:PropertyIsLike wildCard="%" singleChar="_" escape="\ ">
        <ogc:PropertyName>./Title</ogc:PropertyName>
        <ogc:Literal>Electrical%Transmission%</ogc:Literal>
      </ogc:PropertyIsLike>
    </ogc:Filter>
  </csw:Constraint>
</csw:Query>
```



# Protocol Bindings



- A especificação descreve implementações de serviços de catálogo para os seguintes protocolos:
  - Z39.50
  - CORBA/IIOP
  - HTTP
    - Também conhecido como Catalogue Services for the Web - CSW



# Interfaces



- Objetivo:
  - Acessar o catálogo;
- Interface de Serviço do Catálogo
- Interface de Consulta
- Interface de Transação
- Interface de Gerenciamento
- Interface de Intermediação





# sessionInfo



- Esse parâmetro é muito utilizado em várias interfaces da especificação, por isso está sendo apresentado separadamente.
- sessionID
  - Identificada unicamente uma seção cliente/servidor
  - obrigatório
- destinationID
  - Identifica o ‘alvo’ dessa mensagem e pode identificar um servidor, um serviço ou um processo
  - Condicional



# sessionInfo



- requestID
  - Identifica unicamente a mensagem
  - Pode ser usado para monitorar e controlar o processamento resultante de uma requisição
  - Opcional
- additionalInfo
  - Dados relevantes para o contexto de uma troca de mensagem específica
  - opcional



# Interface de Serviço do Catálogo



# Interface de Serviço do Catálogo



- Objetivo
  - Recuperar metadados do próprio serviço de catálogo
  - Implementação obrigatória
- Operações
  - getCapabilities



# getCapabilities



- Objetivo
  - Permitir ao cliente recuperar metadados que descrevam a instância de um catálogo;
- Recebe
  - Opcional parâmetro 'section', com o nome da sessão requerida do documento;
- Retorna
  - Documento com os metadados requeridos chamado *Service Metadata Document*



# getCapabilities Request



- Section
  - opcional



# getCapabilities Response



- ServiceIdentification
  - Metadados do serviço
- ServiceProvider
  - Metadados do provedor do serviço
- OperationMetadata
  - Operações implementadas pelo catálogo
- Content
  - Dados e 'fontes de dados' disponíveis pelo catálogo
- QueryLanguage
  - Linguagens de consulta suportadas pelo servidor
- **Todos opcionais**



# Interface de Consulta





# Interface de Consulta



- Objetivo

- Permitir que clientes descubram o conteúdo do catálogo
- Implementação obrigatória

- Operações

- query
  - present
  - describeRecordType
  - getDomain
- } Necessitam de uma seção



# query



- Objetivo
  - Permitir ao cliente executar uma consulta qualquer no catálogo;
  - Procura no conjunto de metadados disponível aqueles que atendam à consulta definida pelo usuário;
- Recebe
  - Parâmetros da consulta
- Retorna
  - Um documento contendo referências aos dados e serviços descritos pelos metadados selecionados, ou apenas o número de itens no conjunto;



# query Request



- 14 parâmetros
- queryExpression
  - queryLanguage
    - CQL
    - Filter, Type-1
  - predicate
    - String com a consulta propriamente dita
- resourceType
  - Data set
  - Data set collection
  - Service

Apresentados apenas os obrigatórios



# query Response



- hits
  - Número de registros recuperados na consulta
  - Obrigatório
- retrievedData (condicional)
  - Metadados organizados e formatados como especificado nos parâmetros de entrada.
  - Os valores default são:
    - Apresentação = brief
    - Formato = xml
    - Ordenação = descendente pelo campo ID



# present



- Objetivo
  - Retorna metadados selecionados referentes a dados previamente selecionados por uma consulta
- Recebe
  - Parâmetros de apresentação e ordenação
  - Conjunto de dados do qual requiere-se o detalhamento
- Retorna
  - Um documento contendo referências aos dados e serviços descritos pelos metadados selecionado;



# present Request



- 8 parâmetros
- targetID
  - Conjunto ou lista dos identificadores dos dados
  - Obrigatório
- responseSchema (Opcional)
  - FGDC
  - ISO-19115
  - ISO-19139



# present Response



- retrievedData
  - Metadados organizados e formatados como requerido
  - Obrigatório
- cursor
  - Último registro (result set resource) retornado pela consulta
  - Inteiro não-negativo
- hits
  - Número de registros retornado pela consulta
  - Inteiro não-negativo



# describeRecordType



- Objetivo
  - Recuperar a definição de tipo usados por metadados em uma ou mais fontes.
- Recebe
  - Parâmetros opcionais que identifiquem o tipo do qual se deseja a definição
  - Parâmetros opcionais de formatação (interface) da resposta
- Retorna
  - Um documento contendo a definição dos tipos requeridos, incluindo seu schema;





# describeRecordType Request



- sessionInfo
  - Condicional
- typeName
  - Nome do tipo de metadado para o qual requer a informação
  - Retorna todos os tipos quando omitido
- schemaLanguage
  - Default = XML Schema
- outputFormat



# describeRecordType Response



- typeName
- schemaLanguage
- sessionInfo (condicional)



# getDomain



- Objetivo
  - Retornar valores válidos para algum tipo de metadado;
- Recebe
  - Nome de um ou mais metadados ou parâmetros requeridos;
- Retorna
  - Descrição do domínio de um ou mais metadados ou parâmetros;



# GetDomain Request



- parameterName
  - Nome do metadado ou parâmetro
  - Único e obrigatório



# GetDomain Response



- parameterName
- Para qualquer 'parameter' simples, um dos três parâmetros abaixo pode ser, opcionalmente, usado para descrever o domínio de valores possíveis:
  - listOfValues
    - Domínio de valores possíveis
  - conceptualScheme
    - Referência confiável para o domínio de valores
  - rangeOfValues
    - Limite dos valores máximos e mínimos no domínio



# Interface de sessão



# Interface de Sessão



- Objetivo
  - Prover sessões interativas entre servidores e clientes
  - Implementação opcional
- Operações
  - initialize
  - close
  - status
  - cancel



# initialize



- Objetivo
  - Inicia uma transação e gera um identificador único usado em toda sessão;
- Recebe
  - Identificador da operação de request;
- Retorna
  - Dados descrevendo sucesso ou falha da operação, mais o identificador da sessão, quando a mesma for bem sucedida;





# initialize Request



- sessionInfo
  - condicional;



# initialize Response



- sessionInfo
  - condicional;



# close



- Objetivo
  - Encerra uma transação;
- Recebe
  - Identificador da sessão e a operação requisitada;
- Retorna
  - Identificadores opcionais da sessão;



# Close Request



- sessionInfo
  - condicional;



# Close Response



- sessionInfo
  - condicional;



# status



- Objetivo
  - Retorna o estado atual de uma transação iniciada anteriormente e que ainda está executando ou já terminada;
- Recebe
  - Identificador da sessão, a operação requisitada anteriormente e a requisição dessa operação;
- Retorna
  - Sessão ou IDs requisitados;



# status Request



- sessionInfo
  - condicional
- requestIDtoStatus
  - requestID da operação para a qual se deseja obter o estado
  - condicional



# status Response



- sessionInfo
  - condicional
- status
  - Mensagem contendo o estado da operação;
  - condicional;





# cancel



- Objetivo
  - Termina uma transação previamente iniciada;
- Recebe
  - Identificador da sessão, a operação requisitada anteriormente e a requisição dessa operação;
- Retorna
  - Dados descrevendo sucesso ou falha dessa operação;



# cancel Request



- requestIDtoCancel
  - requestID da operação, a qual se deseja cancelar
  - Obrigatório
- sessionInfo
  - condicional
- freeResources
  - Se 'FALSE', qualquer resultado parcial da operação não será apagado, até que o cliente termine a sessão.
  - condicional



# cancel Response



- **cancelledRequest**
  - requestID da operação, a qual se deseja cancelar;
  - Obrigatório;
- **sessionInfo**
  - condicional;
- **diagnostic**
  - Mensagem contendo o resultado da operação de cancelamento;
  - condicional;



# Interface de Gerenciamento



# Interface de Gerenciamento



- Objetivo
  - Atualização do catálogo, por meio de inserção, remoção e atualização/alteração de metadados.
  - Implementação opcional
- Operações
  - transaction
  - harvestRecords



# transaction



- Objetivo
  - Executar um conjunto especificado de operações de inclusão, alteração e atualização de metadados no catálogo;
- Recebe
  - O conjunto das operações “insert”, “update” e “delete”;
- Retorna
  - Um resumo dos resultados da transação;



# transaction Request



- insert
  - Cria novos metadados no catálogo;
  - Opcional;
- update
  - Modifica metadados existentes no catálogo;
  - Opcional;
- delete
  - Remove metadados do catálogo.
  - opcional;



# transaction Response



- transactionSummary
  - Resumo do resultado da operações que inclui o número de registros inseridos, atualizados ou removidos;
  - Obrigatório;
- insertResults
  - Breve descrição do registro criado pela operação, que deve incluir o identificador do registro;
  - opcional;





# transaction Response



- Exemplo de um transaction Response (CSW), indicando quantos registros foram inseridos, alterados ou apagados;

```
<?xml version="1.0" encoding="UTF-8" ?>

<csw:TransactionResponse version="2.0.0" xmlns:csw="http://www.opengis.net/cat/csw">

    <csw:TransactionSummary>

        <csw:totalInserted>0</csw:totalInserted>

        <csw:totalUpdated>1</csw:totalUpdated>

        <csw:totalDeleted>0</csw:totalDeleted>

    </csw:TransactionSummary>

</csw:TransactionResponse>
```



# harvestResource



- Objetivo
  - Adicionar ao repositório de metadados, um conjunto de metadados de uma nova fonte especificada;
- Recebe
  - Uma mensagem contendo a fonte dos metadados que se deseja adicionar ao repositório;
- Retorna
  - O reconhecimento de que a operação foi aceita e validada, ou um resumo dos resultados que identificam os novos registros adicionados ao catálogo;



# harvestResource Request



- **source**
  - URL da localização da fonte de dados;
  - Obrigatório;
- **resourceFormat**
  - Identifica o tipo de mídia que indica o formato da fonte recuperada;
  - Obrigatório;
- **resourceType**
  - Identifica o tipo da fonte, caso se saiba;
  - opcional;



# harvestResource Request



- responseHandler
  - URL utilizado para operações assíncronas;
  - opcional;
- harvestInterval
  - Intervalo de tempo entre a execução da operação harvestResource
  - opcional



# harvestResource Response



- acknowledgement
  - Resumo do resultado da operações , como o número total de registros afetados pela operação;
  - obrigatório;
- insertResults
  - Breve descrição do registro criado pela operação, que deve incluir o identificador do registro;
  - obrigatório;



# Interface de Intermediação



# Interface de Intermediação



- Objetivo
  - Ordenar e identificar as fontes que estão registradas no catálogo, mas não estão diretamente acessíveis para o cliente;
  - Implementação opcional
- Operações
  - order



# order



- Objetivo
  - Retorna um dado ou endereço de serviço identificado registrado no catálogo, mas não acessível ao usuário;
- Recebe
  - Identificadores dos registros a serem ordenados;
- Retorna
  - Produtos ordenados;





# order Request



- productId
  - Identificadores dos dados a serem ordenados;
  - Obrigatório;
- orderType
  - Tipo da ordenação requisitada
    - orderEstimate
    - orderQuoteAndSubmit
    - orderMonitor
    - orderCancel
- orderID
  - Identificador para uma ordenação;
  - Obrigatório;



# order Request



- orderInformation
  - Especifica a atual ordenação provida pelo cliente
  - Obrigatório;
- userInformation
  - Usado para prover informação sobre o usuário



# order Response



- productID
- orderType
- orderID
- orderStatus
  - Estado atual da ordenação
    - orderBeingEstimated
    - orderEstimated
    - orderBeingQuoted
    - orderBeingProcessed
    - orderCompleted
    - orderNotValid
    - orderCancelled



# order Response



- resourceEstimate
  - Estimativa dos recursos necessários para processar a requisição, como o custo da mesma
- orderInformation



# Catalogue Services for the Web CSW



# Interfaces Obrigatórias



- GetCapabilities
- DescribeRecord
- GetRecords

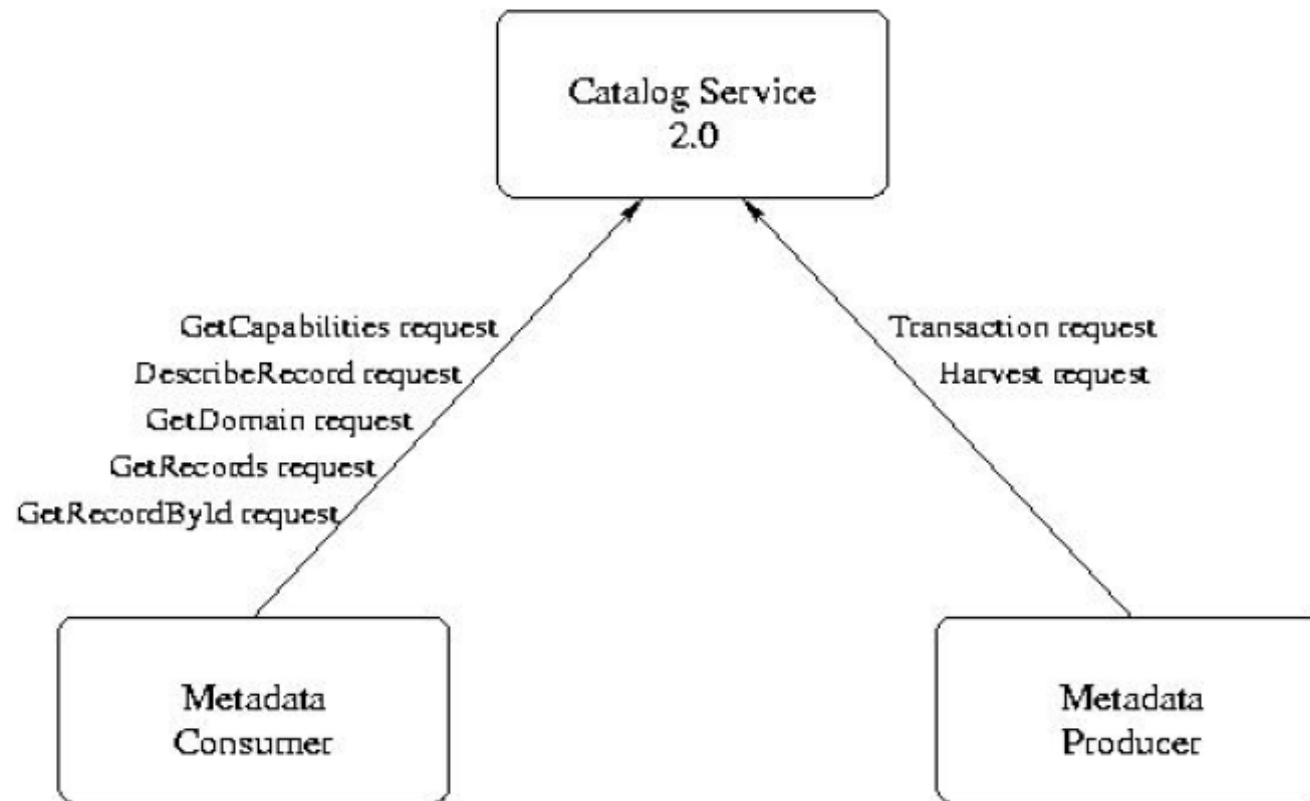


# Interfaces Opcionais



- GetRecordById
- GetDomain
- Harvest
- Transaction

- Arquitetura







# getCapabilities



- Parâmetros do request:

- Request

- Service

- Sections

- ServiceIdentification

- Service Provider

- OperationsMetadata

- Filter\_Capabilities

- Metadados sobre o filtro, caso o servidor implemente

- acceptVersions

- AcceptFormats

```
URL?REQUEST=GetCapabilities&  
SERVICE=CSW&  
ACCEPTVERSION=2.0.0,0.7.2
```



# describeRecord



- Permite um cliente descobrir elementos suportados por um catálogo;
- Permite que alguns ou todos elementos sejam descritos;



# describeRecord Request



- Request
- Service
- Version
- Namespace
  - Define exatamente quais tipos serão descritos
- typeName
  - Nomes dos tipos que serão descritos pelo catálogo
- outputFormat
  - default = application/XML
- schemaLanguage
  - default = xmlSchema



# describeRecord Response



```
<?xml version="1.0" encoding="ISO-8859-1"?>  
  <xs:simpleType name="SV_ParameterDirection_Type">  
    <xs:restriction base="xs:string">  
      <xs:enumeration value="in" />  
  
      <xs:enumeration value="out" />  
  
      <xs:enumeration value="in/out" />  
    </xs:restriction>  
  </xs:simpleType>
```



# getRecords



- Reúne as operações search e present
- 17 parâmetros de entrada



# getRecords Request



- Request
- Service
- Version
- typeNameNames
  - Nomes dos tipos que serão recuperados



# getRecords Response



```
<csw:RequestId>testingCSW</csw:RequestId>
```

```
<csw:SearchStatus status="none" timestamp="2007-07-31T07:31:25" />
```

```
<csw:SearchResults expires="2007-07-31T07:31:26"
```

```
  nextRecord="1"
```

```
  numberOfRecordsMatched="3"
```

```
  numberOfRecordsReturned="0"
```

```
  recordSchema="http://131.220.126.141:8080/deegree-  
    csw/services?service=CSW&version=2.0.0&request=DescribeRecord&typeName=csw:d  
    ataset"
```

```
/>
```



# CS: Principais Mentores



- Autor:
  - Douglas Nebert, Arliss Whiteside
- Principais Organizações:
  - ESRI, FGDC, NASA, ORACLE, MITRE
- Colaboradores:
  - Con terra GmbH, Cubewerx, Ionic Software





# SLD: Padrões Relacionados



- Padrões complementares:
  - Filter
  - GML
- Padrões não OGC
  - XML/XML esquemas



# Estatística de Implementação



Total	Comp.	Especificação	Versão
14	1	<a href="#"><u>Catalog Interface (1.0)</u></a>	CAT 1.0
9	0	<a href="#"><u>Catalog Interface (1.1.1)</u></a>	CAT 1.1.1
13	0	<a href="#"><u>Catalogue Service Implementation Specification (2.0.1)</u></a>	CAT 2.0.1
1	0	<a href="#"><u>Catalogue Service Implementation Specification (2.0.2)</u></a>	CAT 2.0.2
8	0	<a href="#"><u>Catalogue Services - ebRIM (ISO/TS 15000-3) profile of CSW (0.9.1)</u></a>	CAT2 AP ebRIM 0.9.1

Fonte: OGC



# Ferramentas Certificadas



- Catalog Server for Informix 1 (conTerra)
  - TerraCatalog
  - Versão 1.0 (2.0.2)
  - Cliente
  - <http://www.conterra.de/en/products/sdi/terracatalog/i>



# Ferramentas Não-Certificadas



- Degree
  - CSW cliente (2.0.0)
  - CSW servidor (2.0.0)
  - Demo
    - <http://www.deegree.org/deegree/portal/media-type/html/use>
  - generic OGC WebService client
    - <http://demo.deegree.org/deegree-csw/client/client.html>
- GvSig
  - <http://www.gvsig.gva.es/>



# Considerações Relevantes



- A especificação não prevê algo que identifique unicamente feições ou coleções, necessárias para checar a similaridade de conjuntos de dados, adquiridos de fontes diferentes. O que causaria, replicação de informações quando da integração dos catálogos.