

Infraestrutura Nacional de Dados Espaciais – INDE

Introdução aos Geoserviços



29 a 31 de Maio
São Paulo (SP) Brasil

Instrutor: Wesley Silva Fernandes
e-mail: wesley.fernandes@ibge.gov.br

- **Conceitos de IDE**

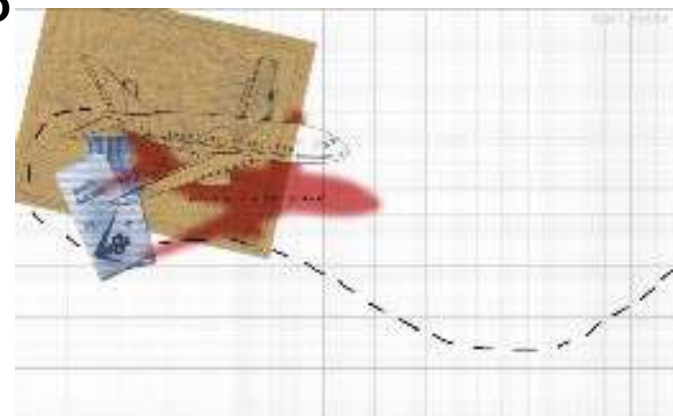
- **INDE - a IDE do Brasil**
- **Como funcionam as IDE**
- **A importância dos padrões nas IDE**
- **Esforços mundiais para padronização de IG – ISO e OGC**
- **Esforços nacionais para padronização – e-ping / CONCAR**

- **O que são geoserviços**

- **Padrões OGC**

- **Geoserver - conceitos**

- **Implementação de geoserviços WMS/ WFS**





Uma das primeiras perguntas que se faz quando ocorre um evento ?

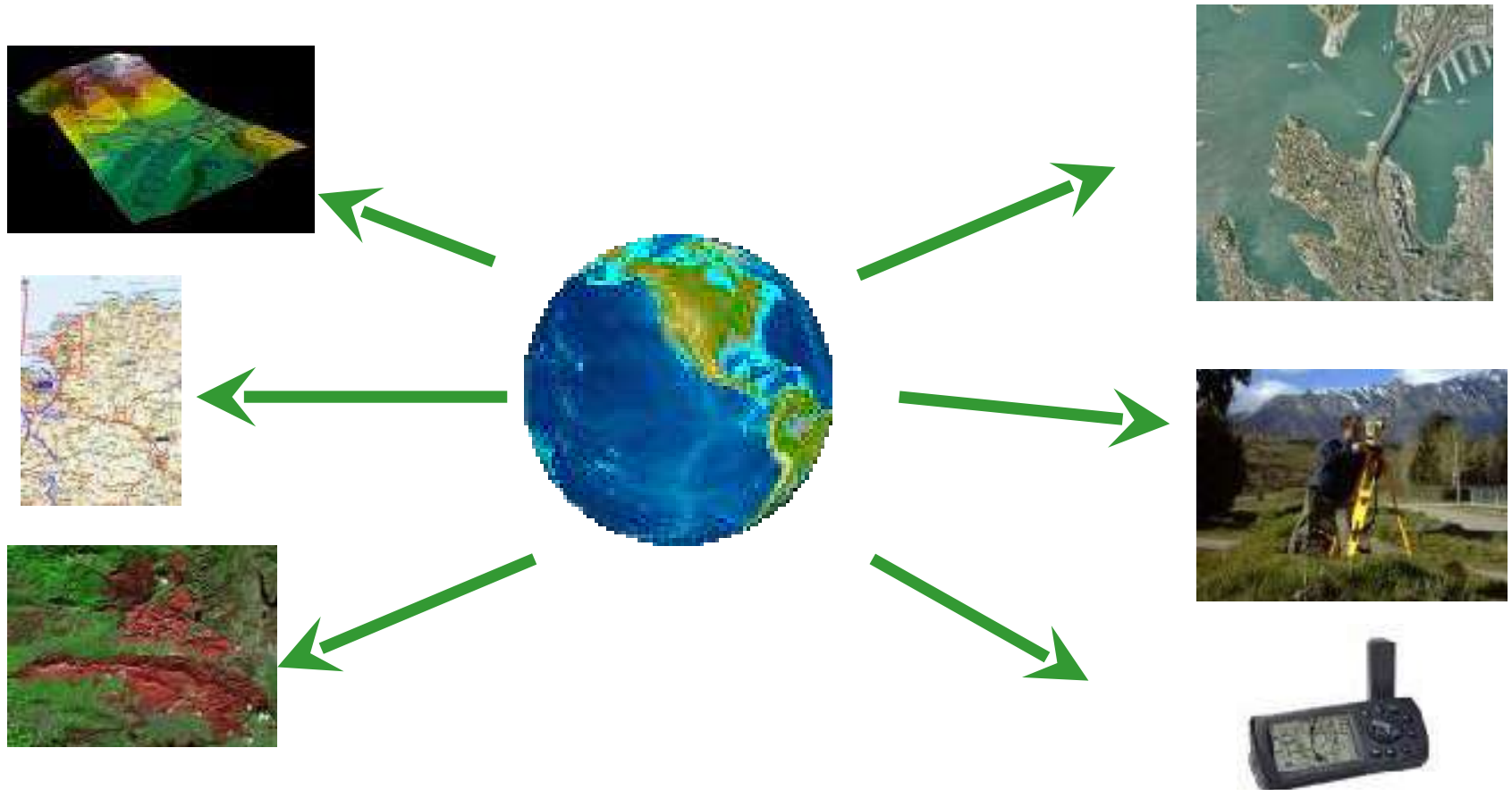
Onde isto aconteceu?

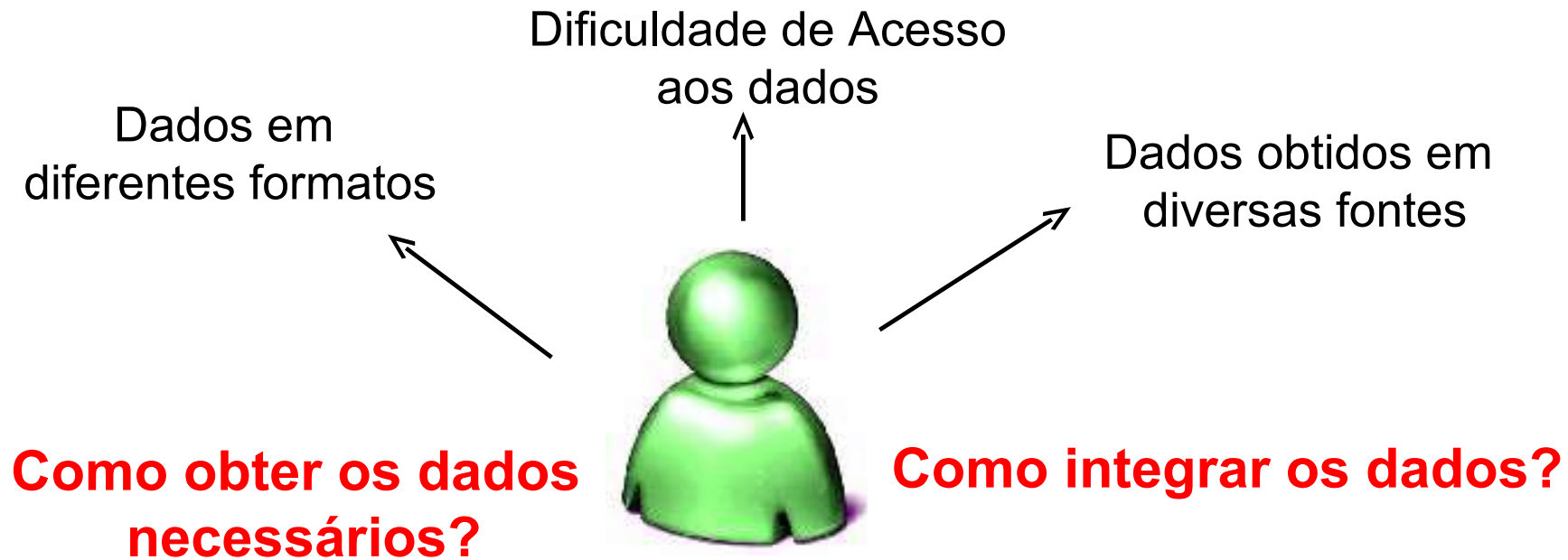


Nas respostas – informações geoespaciais

Informação Geoespacial – o que é

Aquela que se distingue pela sua componente espacial, que associa a cada entidade ou fenômeno uma localização na Terra, traduzida por sistema geodésico de referência, podendo ser derivado, das tecnologias de levantamento, associadas a sistemas globais de posicionamento apoiados por satélites, bem como de mapeamento ou de sensoriamento remoto – Decreto da INDE





Geoprocessamento sem Interoperabilidade

- Desde o início dos SIG, muitos métodos de aquisição, armazenamento, processamento, análise e visualização de dados geoespaciais foram desenvolvidos independentemente um do outro.

- Um sistema de sistemas
 - Composto por um conjunto de recursos:
 - dados, metadados, serviços, *hardware*, *software*, pessoal técnico, usuários, organizações, marcos legais, padrões, acordos...
- Pertencente e administrado por uma **Comunidade**
- Esse sistema é harmonizado e coordenado, para que a sociedade possa compartilhar e usar IG em rede
- Baseado em serviços web
- **COLABORATIVO**



Conjunto integrado de tecnologias; políticas; mecanismos e procedimentos de coordenação e monitoramento; padrões e acordos, necessário para facilitar e ordenar a geração, o armazenamento, o acesso, o compartilhamento, a disseminação e o uso dos dados geospaciais de origem federal, estadual, distrital e municipal

Decreto nº 6.666/2008



2008

INDE/Brasil



2006

IEDG/Equador, NSDI/EUA – revisada

2004

IDEMEX/México, SNIT/Chile

2003

INSPIRE/Europa, IDERC/Cuba



2002

IDEE/Espanha

2001

CGDI/Canadá, ANZLIC/Austrália e Nova Zelândia

2000

ICDE/Colômbia



1996

NSDI/EUA

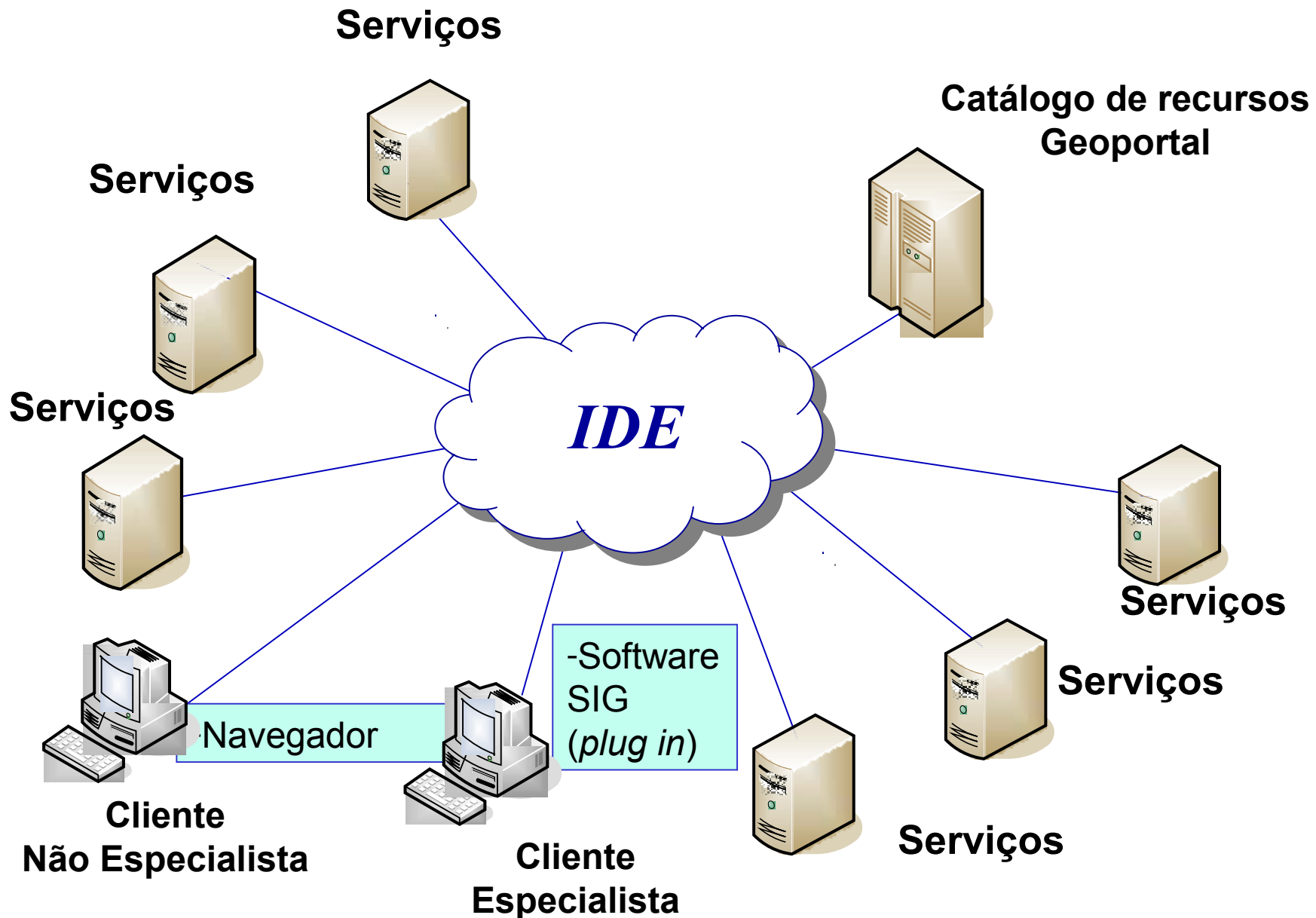
1995

SNIG/Portugal





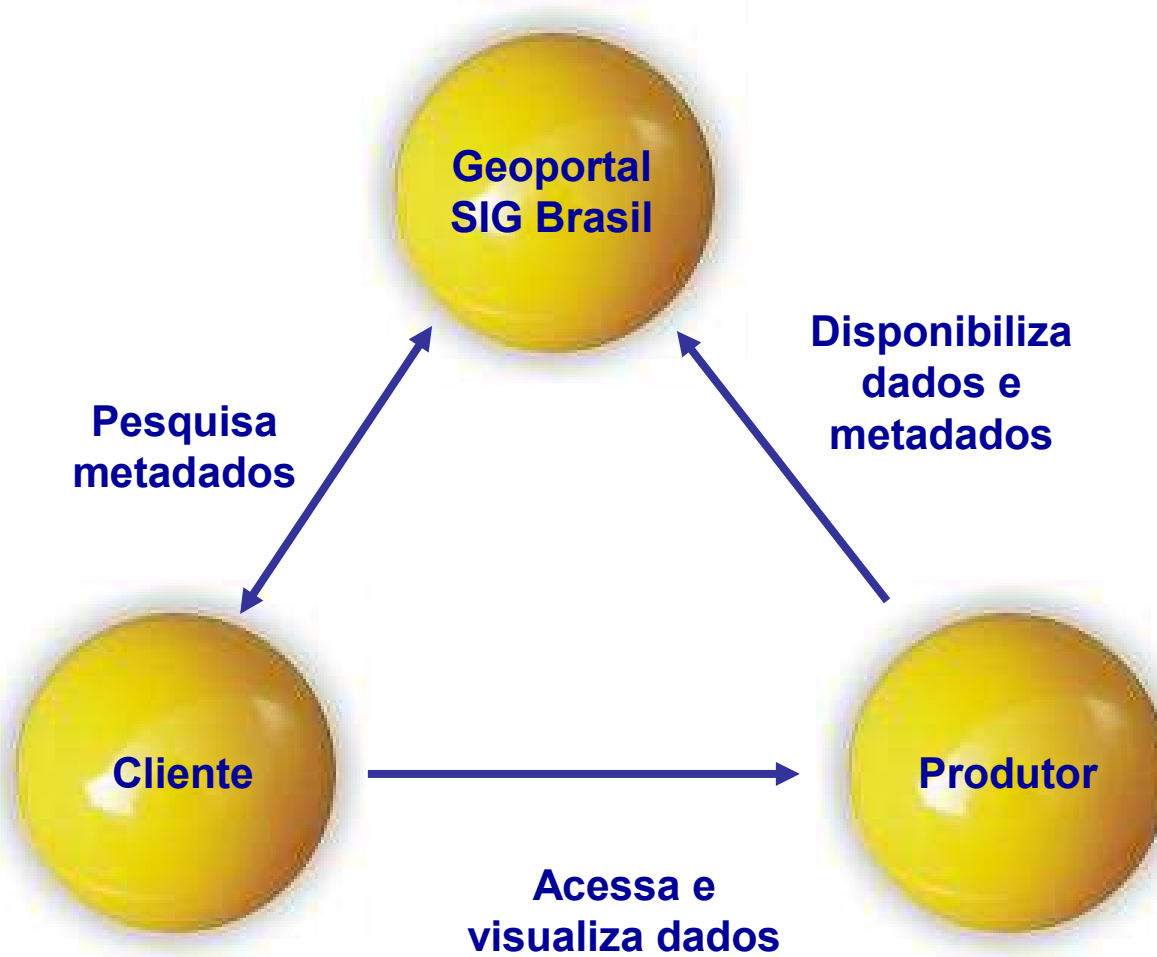
Arquitetura de uma IDE

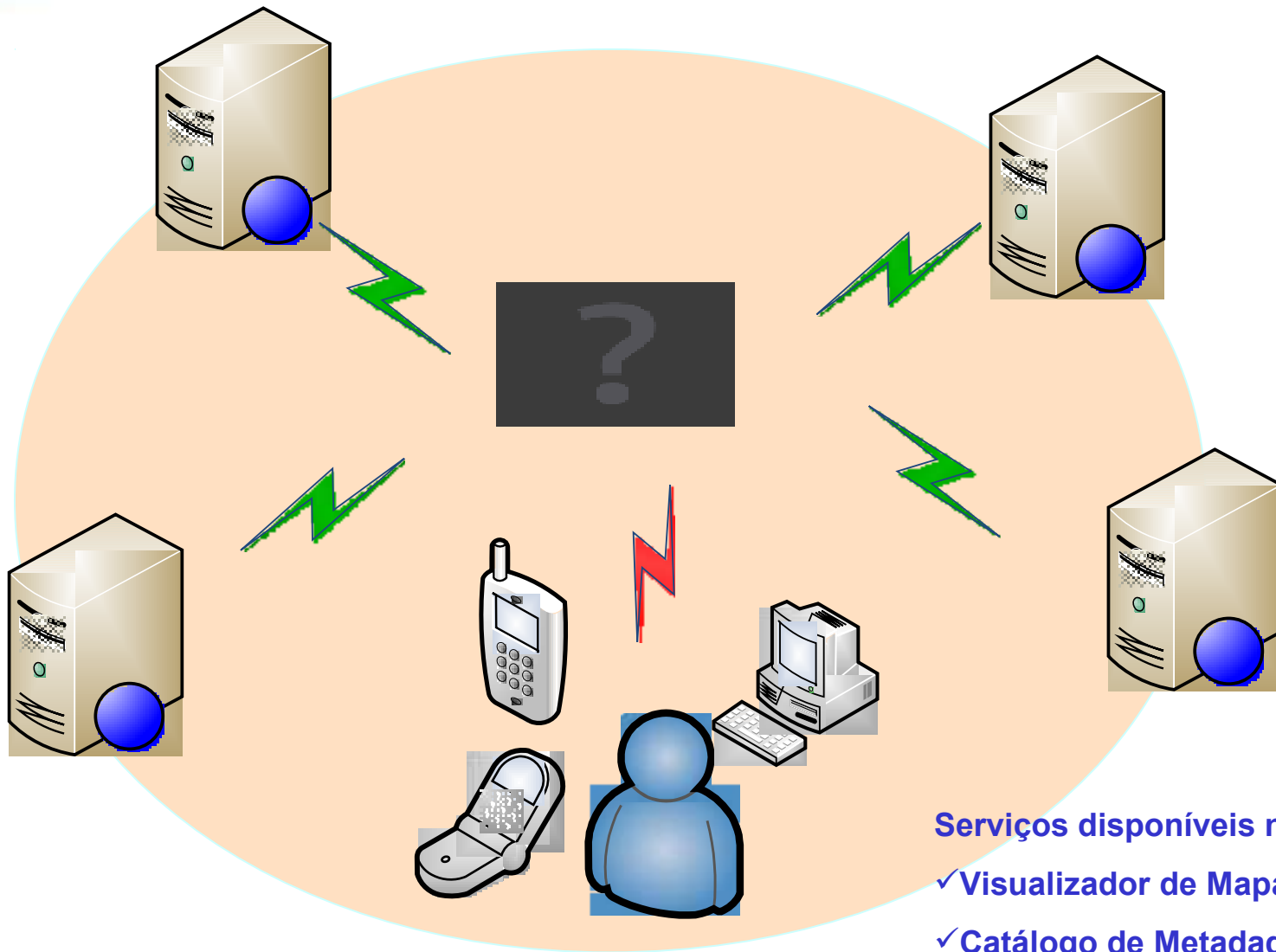




Arquitetura Orientada a Serviços

Modelo de Serviços Web da INDE

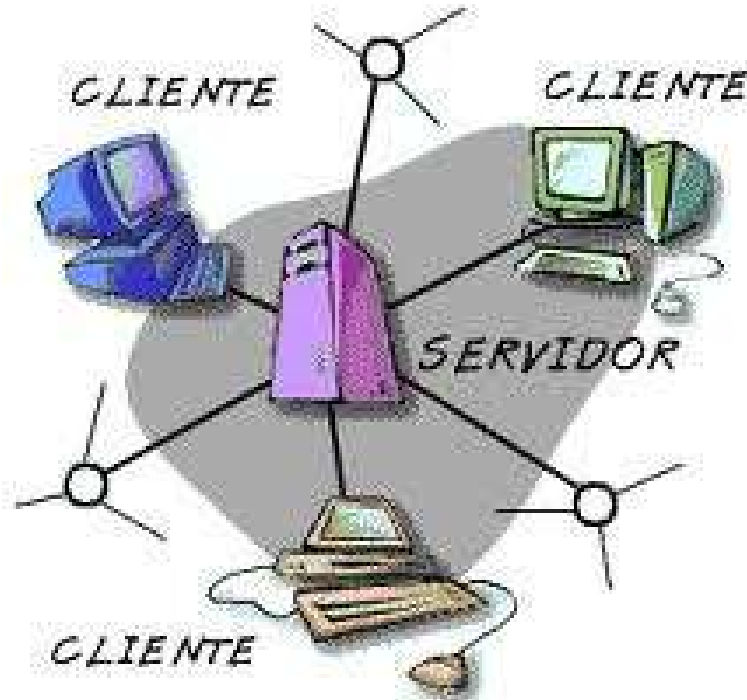


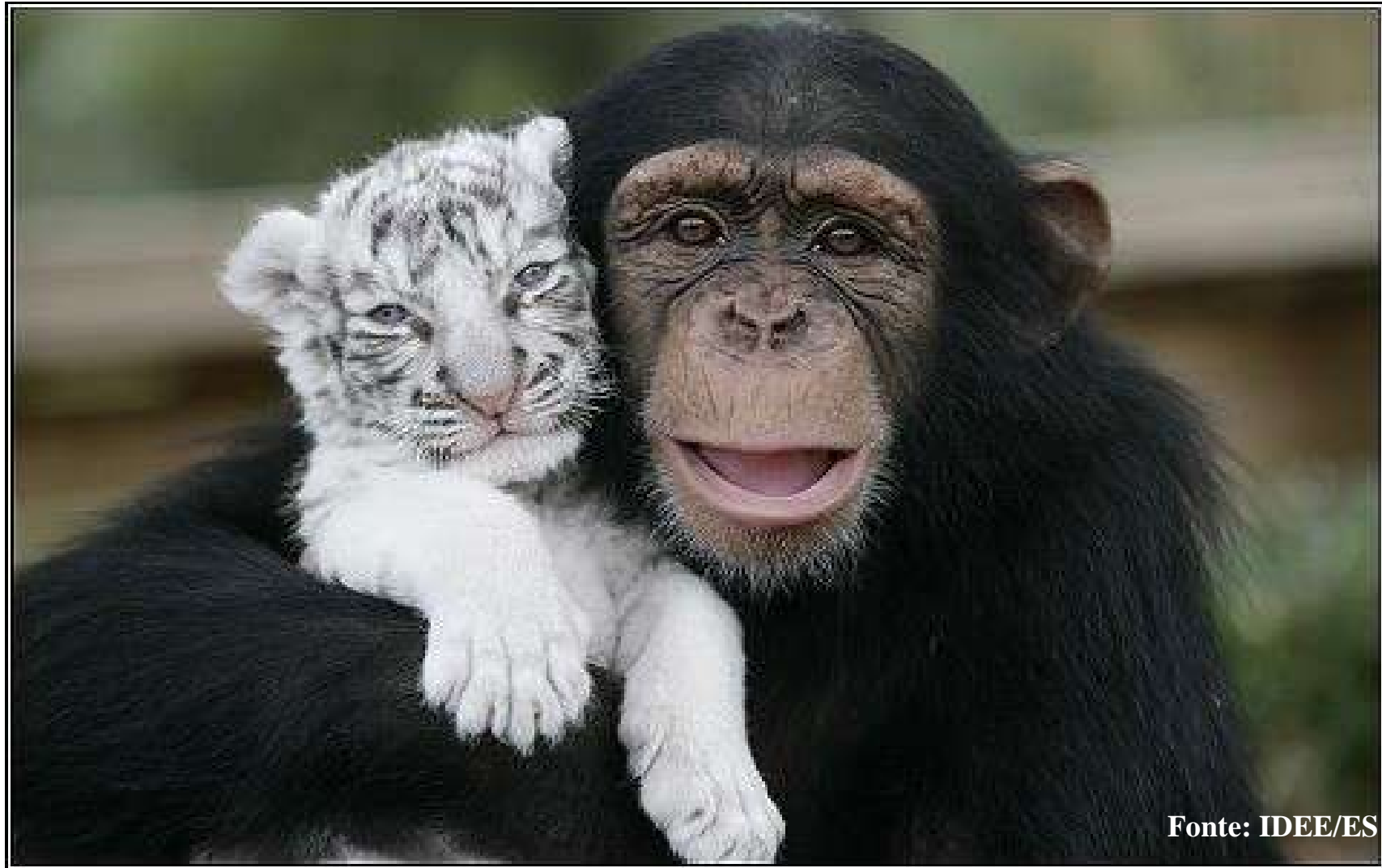


Serviços disponíveis no Portal:

- ✓ Visualizador de Mapas
- ✓ Catálogo de Metadados
- ✓ DBDG
- ✓ Catálogo de Serviços

- **Permite**
 - 1) Que os usuários façam as suas consultas
 - 2) Que os sistemas se comuniquem
 - 3) Que se gerem aplicações e serviços de valor agregado

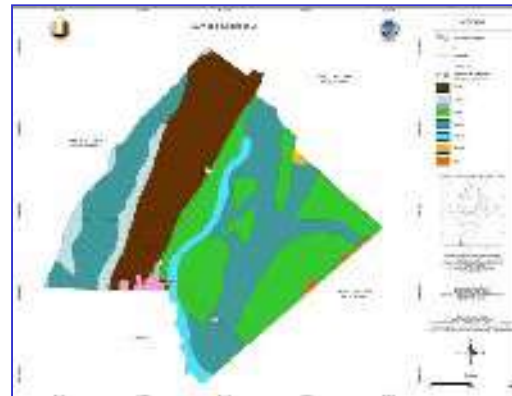




Fonte: IDEE/ES

Dificuldades IG – Diversidade de Fontes

- Será necessário utilizar dados de diversas fontes, como de:
 - Imagens de satélite;
 - Diferente tipos de Banco de Dados;
 - CAD;
 - Diferentes Plataformas SIG;
 - GNSS, etc.
- Certamente esses dados serão adquiridos de órgãos diferentes...



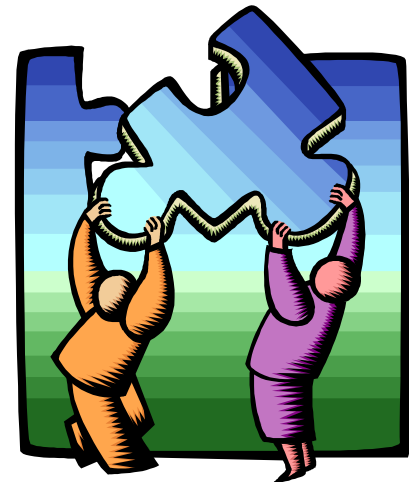
- Problemas que serão encontrados:
 - Integrar os dados;
 - Custo envolvido neste processo;
 - Tempo necessário para integração;
 - Acesso aos Dados necessários;
 - Inexistência de Metadados;
 - Possível perda de dados, etc.
- Nesta situação, ainda que o acesso aos dados fosse fácil, a tarefa de integrá-los seria árdua, propensa a erros e algumas vezes inviável.



- A maior parte dos problemas está relacionado com a ausência de interoperabilidade entre os dados;
- As dificuldades relacionadas a falta de interoperabilidade são encontradas em diferentes níveis:
 - Falta de correspondência entre diferentes conjuntos de dados (ex.: Rio x Curso D'Água);
 - Dados produzidos por aplicações de diferentes setores;
 - Entre tipos de dados diferentes (*Raster* x *Vetor*)
 - Diferenças entre mesma classe de feições produzidas por fontes e processos diferentes;



- Como resolver estes problemas??????
- A Solução está em três componentes básicos:
 - Tecnologia;
 - Adoção de um conceito comum de *Dados de Referência*;
 - Política, para apoiar as implementações básicas necessárias;
- Dois componentes da solução proposta está relacionado à **Padronização**.



- A definição de Dados de Referência tem como objetivo facilitar o compartilhamento de dados entre os usuários;
- Cada feição geoespacial poderá ser fornecida por um produtor de dados diferente;
- Apesar de existirem muitos produtores de dados, o conjunto dos dados deverá ser integrado para criar um repositório de dados principal;
- Com a disponibilização deste conjunto de dados o problema da produção redundante será resolvida, ou minimizada, diminuindo o custo dos dados geoespaciais, pois o mesmo será dividido entre os produtores;



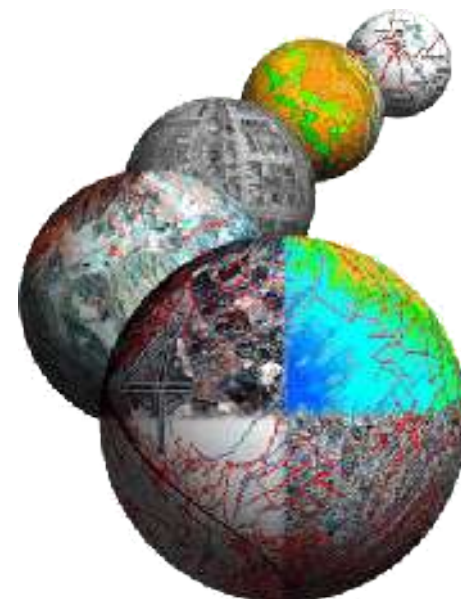
- A definição dos dados de Referência inclui um modelo de dados básico que deve ser seguido pelos produtores dos dados que serão disponibilizados no repositório central;
- Os usuários dos dados poderão estender esta modelagem para atender suas necessidades;
- No Brasil, temos a Especificação Técnica para Dados Geospaciais Vetoriais (EDGV) elaborada pela CONCAR.



- Os benefícios da adoção de uma definição comum de dados geoespaciais são:
 - Os produtores de dados irão produzir os dados de sua especialidade, melhorando a qualidade dos mesmo;
 - A atualização dos dados será realizada com mais frequência, devido a produção compartilhada;
 - Os usuários irão compartilhar os dados com mais facilidade



- Devido a relevância do assunto, a ISO resolveu discutí-lo no que trata da padronização no campo das Informações Geográficas/Geomática
- Pensando em dar uma solução ao problema da falta de Interoperabilidade a ISO/TC211 definiu uma série de especificações, como por exemplo a ISO 19119 (“Geographic Information – Services”).



Padronização – ISO 19119

“A ISO 19119 fornece uma estrutura para os desenvolvedores criarem *Softwares* que habilitam as usuários acessarem e processarem dados geográficos de uma variedade de origens através de uma interface genérica de *Software* dentro de um ambiente tecnológico aberto”

Extended OSE Service Category	Relevant ISO 19100 series standard
Geographic human interaction services	19117: Geographic Information - Portrayal
	19128: Geographic Information - Web Map server interface
	19107: Geographic Information - Spatial schema
	19110: Geographic Information - Methodology for feature cataloguing
	19111: Geographic Information - Spatial referencing by coordinates
	19112: Geographic Information - Spatial referencing by geographic identifiers
	19115: Geographic Information - Metadata
	19123: Geographic Information - Schema for coverage geometry and functions
	19125-1: Geographic Information - Simple feature access - Part 1: Common architecture
19128: Geographic Information - Web Map server interface	
Geographic model/Information management services	
Geographic Workflow/Task management services	(no relevant ISO 19100 series standards)
Geographic processing service	19107: Geographic Information - Spatial schema
	19108: Geographic Information - Temporal schema
	19109: Geographic Information - Rules for application schema
	19111: Geographic Information - Spatial referencing by coordinates
	19116: Geographic Information - Positioning services
	19123: Geographic Information - Schema for coverage geometry and functions
Geographic communication services	(no relevant ISO 19100 series standards)
Geographic system management services	(no relevant ISO 19100 series standard)

Os serviços geográficos da ISO 19119 foram especificados com os seguintes propósitos:

- Fornecer uma estrutura abstrata que permite o desenvolvimento coordenado de serviços específicos;
- Disponibilizar serviços interoperáveis através de padronização de interfaces;
- Habilitar o uso de dados de serviços de fornecedores diferentes a partir da sobreposição dos mesmos;
- Definir uma estrutura abstrata que pode ser implementada de diversos modos.

Muitos termos são amplamente utilizados na ISO 19119, como:

- **Serviço** – Partes distintas de funcionalidades que são fornecidas por entidades por meio de interfaces. Geralmente fornecido por um software na rede para um software cliente;
- **Interface** – Conjunto de operações com um nome que caracterizam o comportamento de uma entidade;
- **Operação** – Definição de uma transformação ou consulta que um objeto deve ser chamado para executar. Tem um nome e uma lista de parâmetro.



Serviços são acessíveis através de um conjunto de interfaces que são um conjunto de operações.

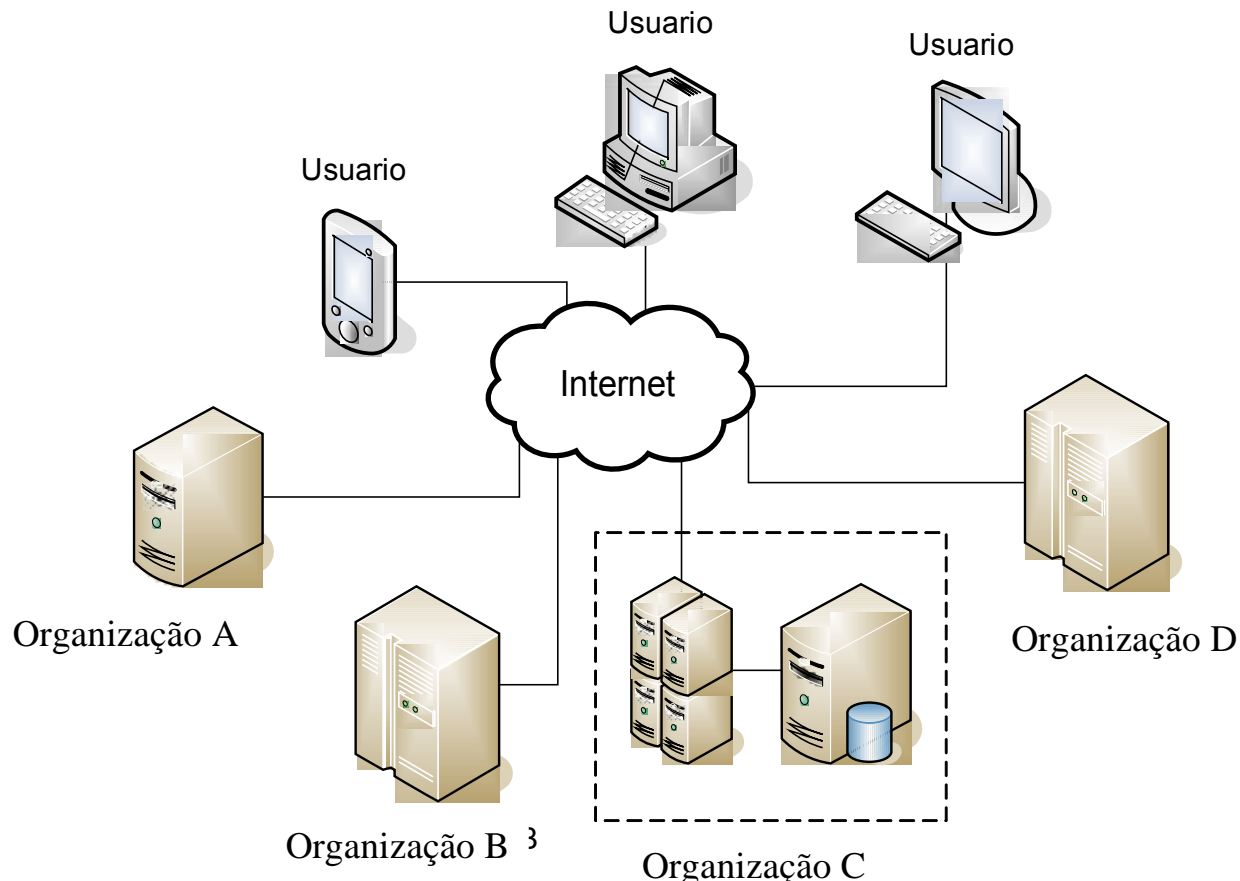
O objetivo das interfaces na ISO 19119 é alcançar a interoperabilidade.

- **Interoperabilidade:**

- Capacidade de comunicar, executar ou transferir dados entre diferentes unidades funcionais sem que o usuário precise ter nenhum conhecimento (conhecimentos específicos) ou muito pouco (conhecimentos genéricos, padrão) sobre as características específicas dessas unidades.



- A ISO 19119 incorporou as mudanças importantes que estavam ocorrendo na computação, como utilização de uma arquitetura baseada em serviços;
- A mudança permitiu um uso mais extensivo da Internet.



Esforços Mundias de Padronização

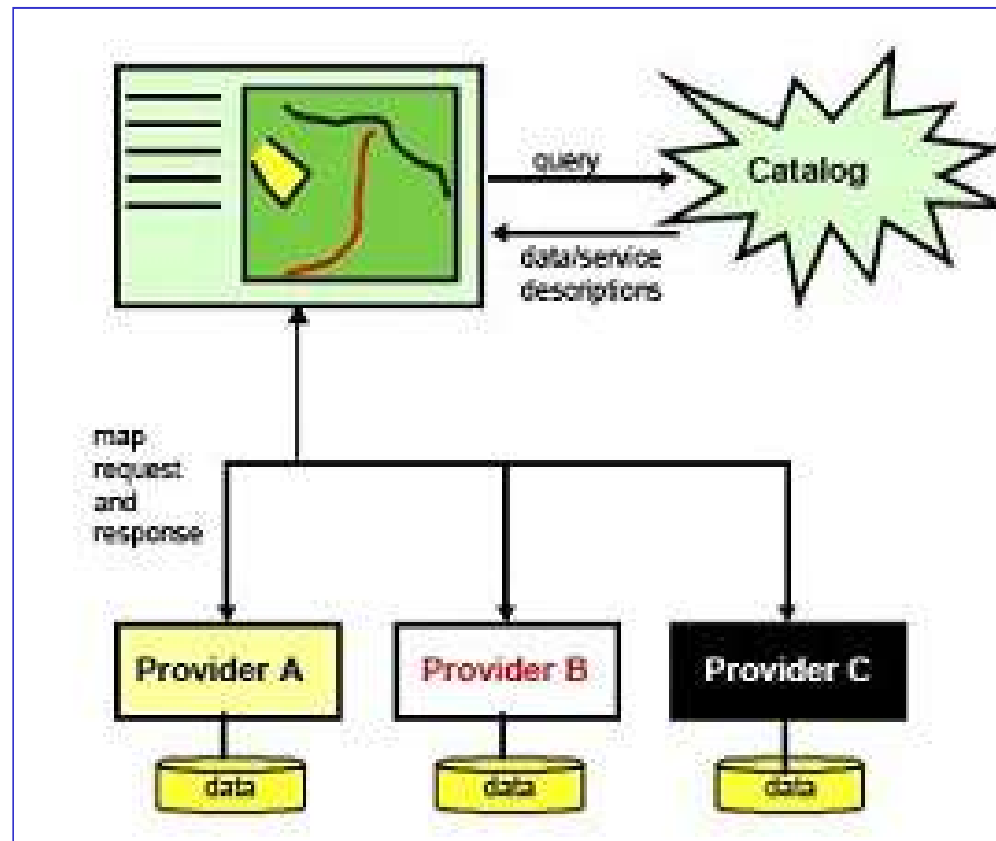
A implementação da Norma ISO referente a IG foi desenvolvida pela OGC, um consórcio de empresas relacionadas a IG, governos e universidades



- Exemplos de interação com serviços baseados na ISO 19119:
 - **Visualização de Catálogo:** serviço que permite ao usuário interagir com um catálogo para localizar, navegar e gerenciar metadados sobre dados geográficos ou geoserviços;
 - **Visualização de Dados Geográficos:** Serviço que permite ao usuário visualizar uma um mais coleções de feições ou coberturas. Permite ao usuário interagir com os dados do mapa;
 - **Editor de Feições Geográficas:** Visualizador que permite ao usuário editar as feições.



Cenário de Interação de usuário em um Ambiente Baseado em Serviços



- **Serviço de Acesso a Feições:** Serviço que fornece a um cliente acesso e gerenciamento a uma base de feições geográfica;
- **Serviço de Acesso a Mapa:** Serviço que fornece a um cliente acesso a figuras de feições geográficas;
- **Serviço de Acesso a Coberturas:** Serviço que fornece um cliente acesso e gerenciamento a uma base de coberturas
- **Serviço de Catálogo:** Serviço que possibilita a descoberta e gerenciamento de serviços por meio de uma base de metadados;



- As Normas da ISO relativas a Informação Geoespacial são conceituais, cabendo a implementação das mesmas pelas organizações interessadas;
- Havia a necessidade de uma implementação de referência;
- Solução???

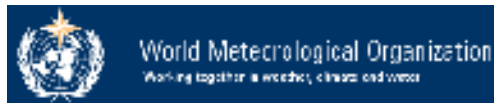


Open GIS Consortium – OGC;

- Consórcio internacional fundado em 1994, sem fins lucrativos:
 - Mais de 300 empresas (ESRI, Intergraph, MapInfo, Autodesk, Oracle...), universidades, administrações ...
- Missão:
 - Promover o desenvolvimento e a utilização de normas técnicas e de sistemas abertos no domínio da IG
- Método:
 - Definir por consenso especificações que padronizem uma ARQUITETURA pública, e INTERFACES PADRÕES para garantir a interoperabilidade.



Parceiros



E outros...

Membros



E outros...

- A padronização não seria obtida pelas empresas isoladamente;
- As empresas reconheceram a importância estratégica da padronização;
- A OGC reúne as organizações relacionadas à Dados Geoespaciais e busca um consenso para o bem comum;
- Como a implementação foi obtida através de um consenso, todos seguiram, alcançando a padronização.





EDGV

Especificações Técnicas para Estruturação de Dados Geoespaciais Digitais Vetoriais

Para Incrementar :

- o entendimento
- a disponibilidade
- o acesso,
- a Integração e
- o uso

das Informações Geoespaciais



OGC[®]
Open Geospatial Consortium, Inc.

Nomes Geográficos

Flujo **natural** de agua **fresca** fluyendo regularmente o intermitentemente sobre un lecho y confinando dentro de las orillas de un canal. Fluye hacia el **mar**, lago, **pantano** u otro río . El **punto de origen** es su fuente. El camino que sigue es su **curso**. La parte donde entra al **mar**, lago o pantano es su boca. Donde se encuentra con otro río es una confluencia. Al menos una **parte del año** es mayor que un arroyo y fluye.



International
Organization for
Standardization



Normas e Padrões na INDE



(Decreto 6.666/08)

- Art. 1º Fica instituída, no âmbito do Poder Executivo federal, a Infra-Estrutura Nacional de Dados Espaciais - INDE, com o objetivo de:
....
- II - promover a utilização, na produção dos dados geoespaciais pelos órgãos públicos das esferas federal, estadual, distrital e municipal, **dos padrões e normas** homologados pela Comissão Nacional de Cartografia - CONCAR; e
- Art. 2º Para os fins deste Decreto, entende-se por:
....
- III - Infra-Estrutura Nacional de Dados Espaciais - INDE: conjunto integrado de tecnologias; políticas; mecanismos e procedimentos de coordenação e monitoramento; **padrões e acordos**, necessário para facilitar e ordenar a geração, o armazenamento, o acesso, o compartilhamento, a disseminação e o uso dos dados geoespaciais de origem federal, estadual, distrital e municipal;
- Art. 4º Os órgãos e entidades do Poder Executivo federal deverão:
 - I - na produção, direta ou indireta, ou na aquisição dos dados geoespaciais, obedecer **aos padrões estabelecidos para a INDE e às normas relativas à Cartografia Nacional**; e



(Decreto 6.666/08)

Art. 6º Compete à CONCAR:

I - estabelecer os procedimentos para a avaliação dos novos projetos de que trata o inciso II do art. 4º;

II - homologar os padrões para a INDE e as normas para a Cartografia Nacional, nos termos do Decreto-Lei nº 243, de 28 de fevereiro de 1967, e do Decreto nº 89.817, de 20 de junho de 1984;

IV - garantir que o DBDG seja implantado e mantido em conformidade com os Padrões de Interoperabilidade de Governo Eletrônico (e-ping), mantidos pela Secretaria de Logística e Tecnologia da Informação, do Ministério do Planejamento, Orçamento e Gestão;

V - promover o desenvolvimento de soluções em código aberto e de livre distribuição para atender às demandas do ambiente de servidores distribuídos em rede, utilizando o conhecimento existente em segmentos especializados da sociedade, como universidades, centros de pesquisas do País, empresas estatais ou privadas e organizações profissionais;



Padrões de Interoperabilidade de Governo Eletrônico (e-PING)

Seguem um conjunto de políticas gerais :

- Alinhamento com a Internet;
- Adoção de navegadores (*browsers*) como principal meio de acesso;
- Adoção de metadados para os recursos de informação do governo;
- Escalabilidade;
- Transparência;
- Adoção Preferencial de Padrões Abertos.



OBS: No caso de dados relativos à área de geoprocessamento, o e-PING define um conjunto de padrões abertos que devem ser utilizados. Esses padrões estão baseados principalmente nas definições do OGC (*Open Geospatial Consortium* – <http://www.opengeospatial.org/>).



CSW – Para catálogo de Metadados

GML - padrões de arquivos para intercâmbio entre estações de trabalho



WCS – *Web Coverage Service* - acessar informações georreferenciadas que possuem valores em todo o espaço considerado, sem fronteiras bem definidas (geo-campos).

WFS – *Web Feature Service*: especificação OGC que define a interface de um serviço que permite acessar e manipular dados geográficos codificados em GML na Internet (HTTP).

WFS Básico (WFS): implementa operações somente leitura, que permitem obter os dados espaciais.

WFS Transacional (WFS-T): implementa as operações transacionais, usadas para manipular os dados remotamente.

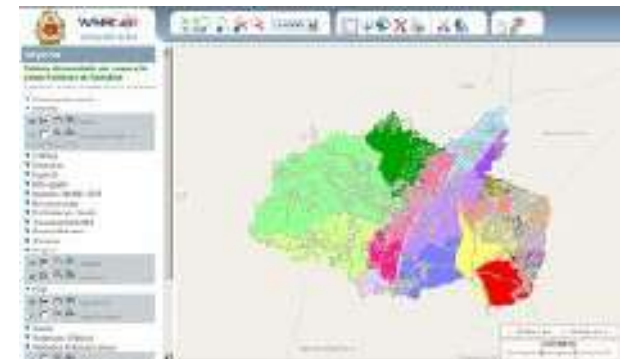
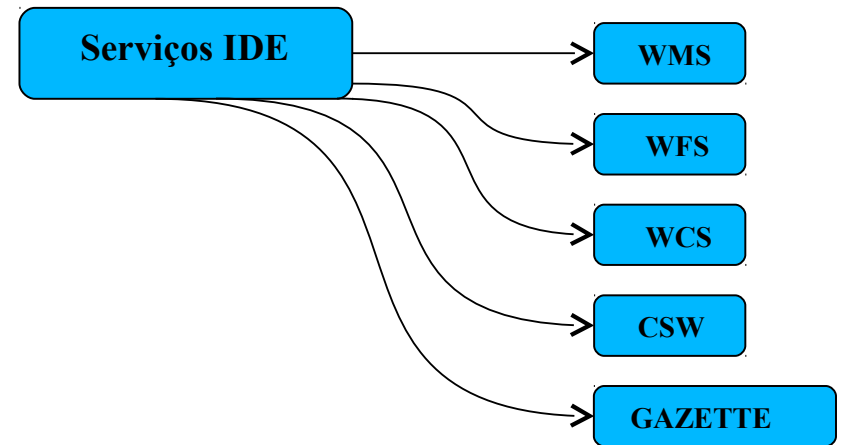
WMS – Web Map Service: disponibilizar mapas (dados geográficos editados) ou imagens na Internet (HTTP).



O que são Geoserviços

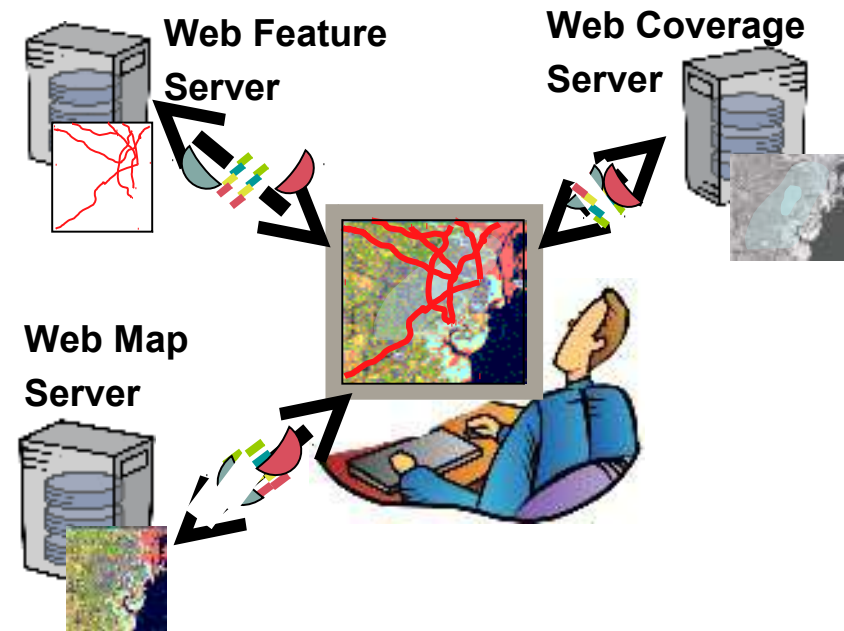
São as **funcionalidades**, acionadas mediante um navegador de Internet, que uma IDE oferece aos usuários para acessar dados e metadados geoespaciais.

Se organizam em serviços de visualização de mapas, de download, de consulta, de conversão





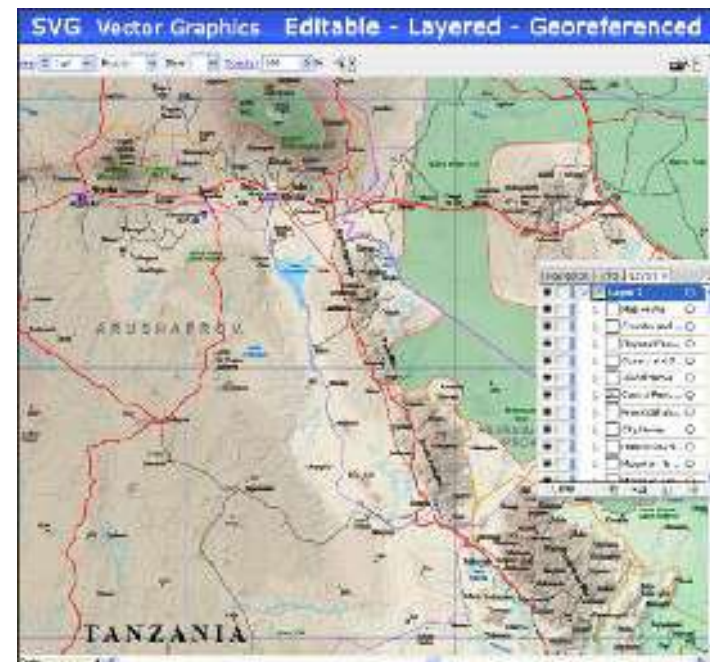
- Web Map Service (WMS)
- Catalog Service Web (CSW)
- Gazetteer (Gaz)
- Web Feature Service (WFS, WFS-T)
- Web Coverage Service (WCS)
- Web Map Context (WMC)
- Filter
- Sensor Web Enablement (SWE)
- Geolinked Data Access Service (GDAS)
- Web Processing Service (WPS)
- Web Coordinate Transformation System (WCTS)
- Style Layer Descriptor (SLD)





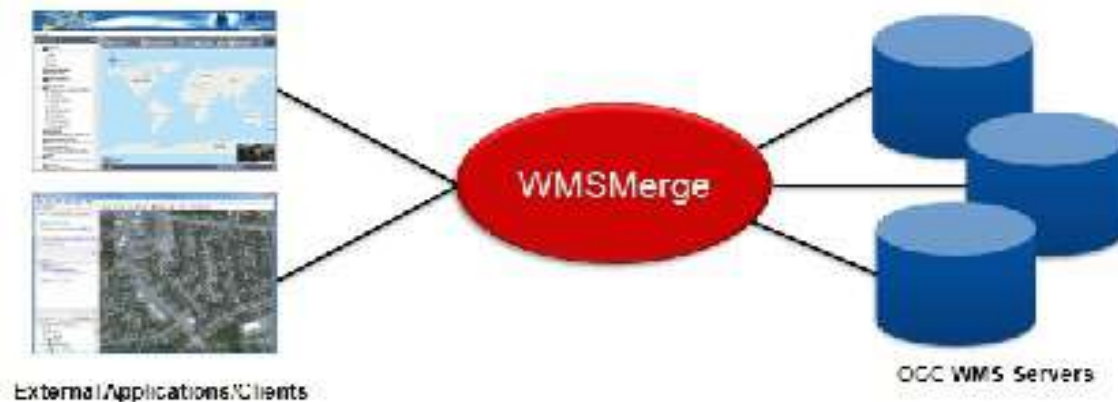
Define um "mapa" como uma representação de informação geográfica na forma de um arquivo digital de imagens adequada para exibição na tela de um computador

Os mapas produzidos pelos serviços WMS normalmente são uma imagem no formato PNG, GIF ou JPEG, e ocasionalmente gráficos vetoriais no formato SVG (Scalable Vector Graphics) ou WebCGM (Computação Gráfica Web Metafile)





- Pode-se realizar consultas WMS a partir de qualquer navegador Web através de requisições Http
- Benefícios:
 - Fornece uma interface padrão para acessar dados geoespaciais
 - Clientes podem realizar requisições a diferentes servidores de dados espaciais e combinar o resultado obtendo uma visão integrada dos dados
 - Suportado por diversos servidores e clientes



WMSMerge Processing Diagram



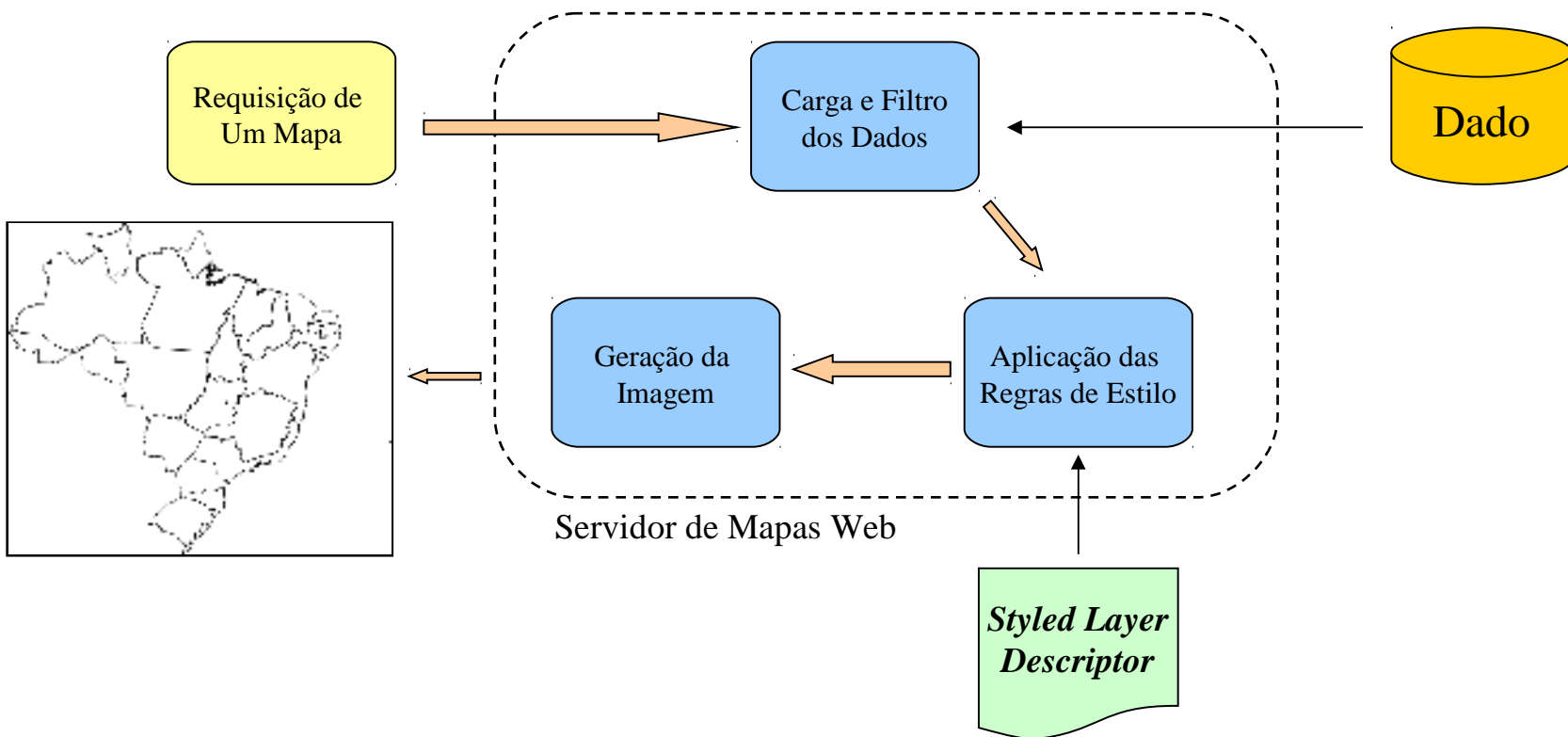
- Http suporta dois métodos de solicitação: GET e POST. O WMS básico só é definido para a solicitação GET;
- A forma de uma requisição HTTP GET é:
`http://host[:port]/path?{name[=value]&;}`

Exemplo:

`http://200.255.94.99:8080/geoserver/wms?
SERVICE=WMS&REQUEST=GetCapabilities`

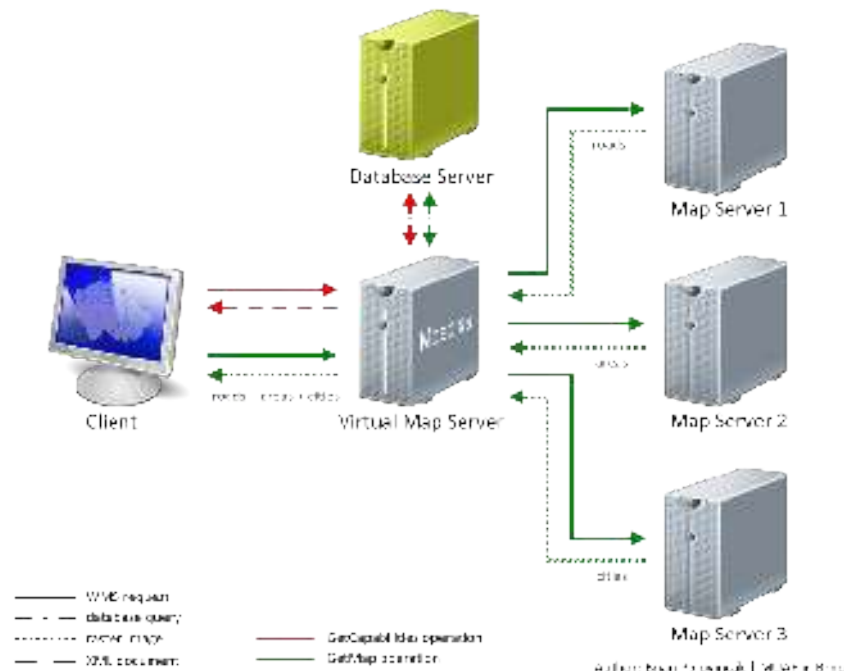


WMS - Solicitação e Resposta Http





- *GetCapabilities*: Retorna metadados do servidor;
- *GetMap*: Retorna um mapa geográfico no formato de imagem;
- *GetFeatureInfo* (opcional): Retornar atributos alfanuméricos associados as feições do mapa;
- *GetLegendGraphic* (opcional): Retorna uma legenda do mapa gerado;





- **Versão**
 - O formato é do tipo x.y.z;
 - A versão aparece no XML de resposta do GetCapabilities;
- **Request**
 - É o nome de uma das operações oferecidas pelo serviço WMS;
- **Formato**
 - É o formato de saída da resposta de uma operação. O XML de resposta da operação GetCapabilities define os formatos que cada operação permite realizar;

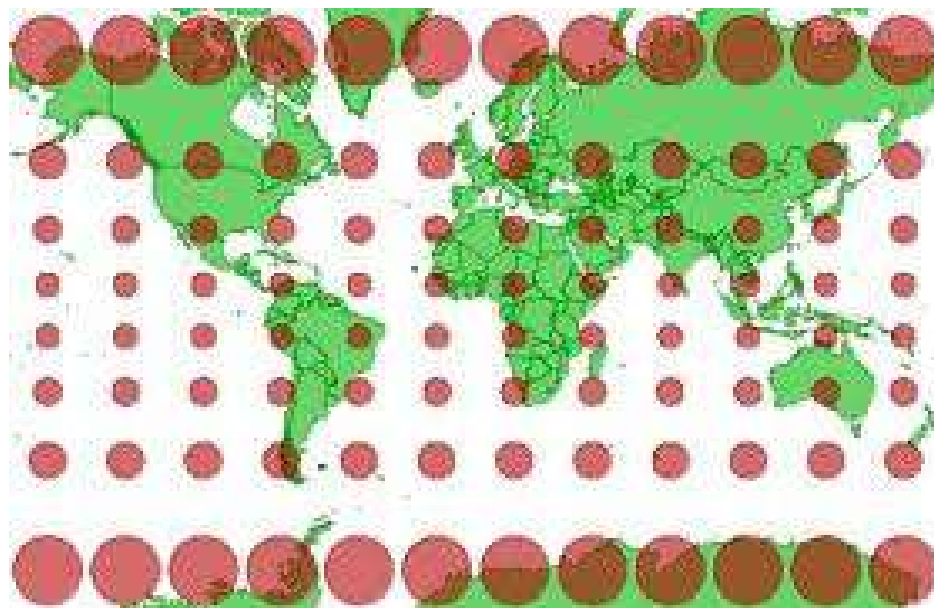


- **SRS**
 - O Sistema de Referência Espacial é o parâmetro que designa o código de um sistema de referência de coordenadas horizontais. Geralmente utiliza os códigos da EPSG.
- **BBOX**
 - É um conjunto de quatro coordenadas, com as unidade do SRS utilizado, que definem um retângulo envolvente da área de interesse, com o X mínimo, Y mínimo, X máximo e Y máximo.
- **Width e Height**
 - Largura e altura da figura que representa o mapa, respectivamente, em pixels.
- **Style**
 - Permite associar um estilo alternativo ao *layer* corrente.



EPSG

O **EPSG** é uma codificação definida pelo European Petroleum Survey Group que associa uma codificação numérica a um sistema de coordenadas cartográficas. Por exemplo, EPSG:29193 corresponde a SAD69 / UTM zona 23 . Para se pesquisar os diversos códigos EPSG disponíveis e sua referência cartográfica, basta ir ao site [Spatial References \(http://spatialreference.org/ref/epsg/\)](http://spatialreference.org/ref/epsg/).

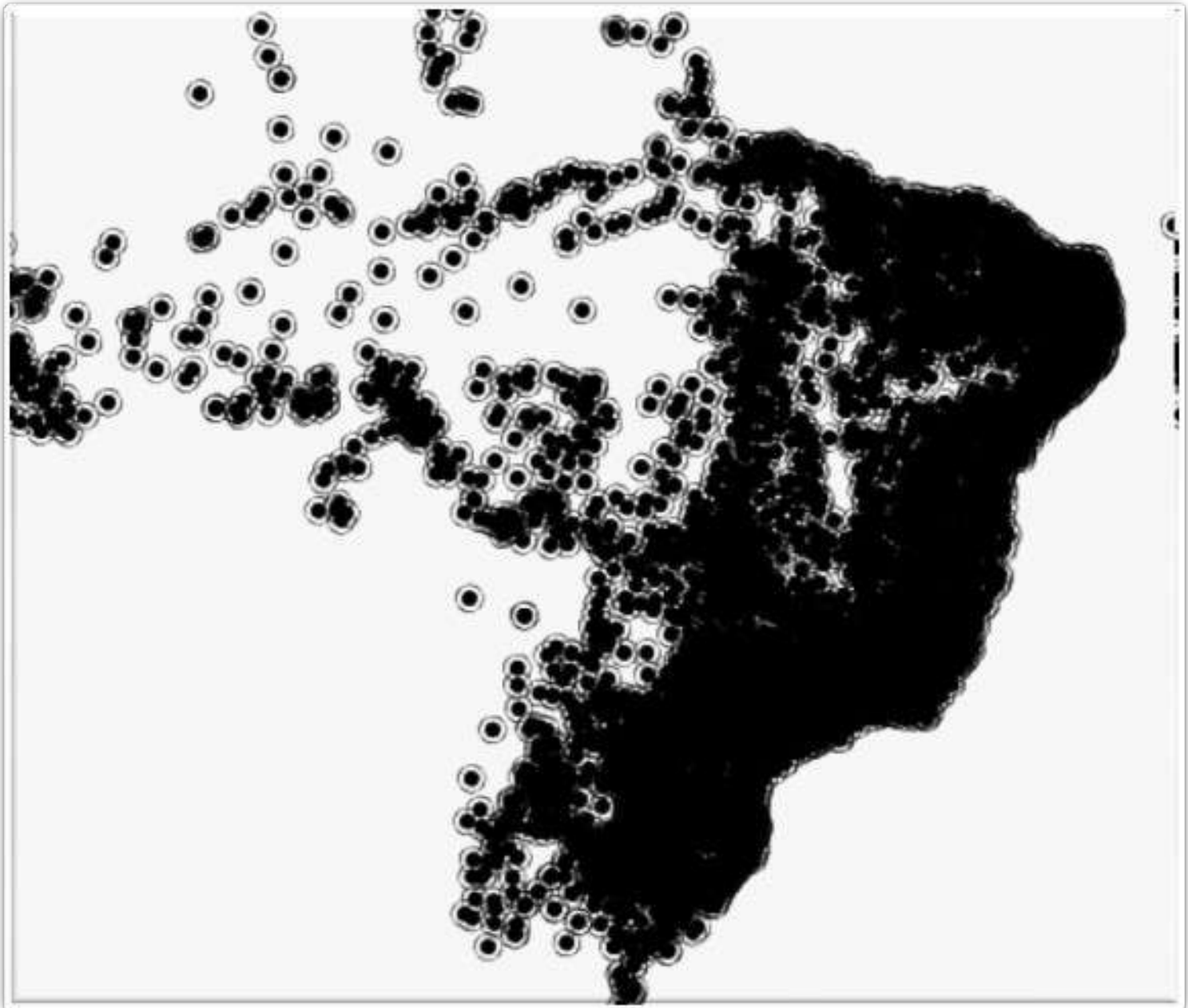




```
http://www.geoservicos.ibge.gov.br/geoserver/gn/wms?  
service=WMS&  
version=1.1.0&  
request=GetMap&  
layers=IBGE:e1000_cidade&  
Styles=&  
bbox=-77.922,-34.204,-32.411,4.603&  
width=512&  
height=436&  
srs=EPSG:4326&  
format=image/png
```



WMS - Resposta



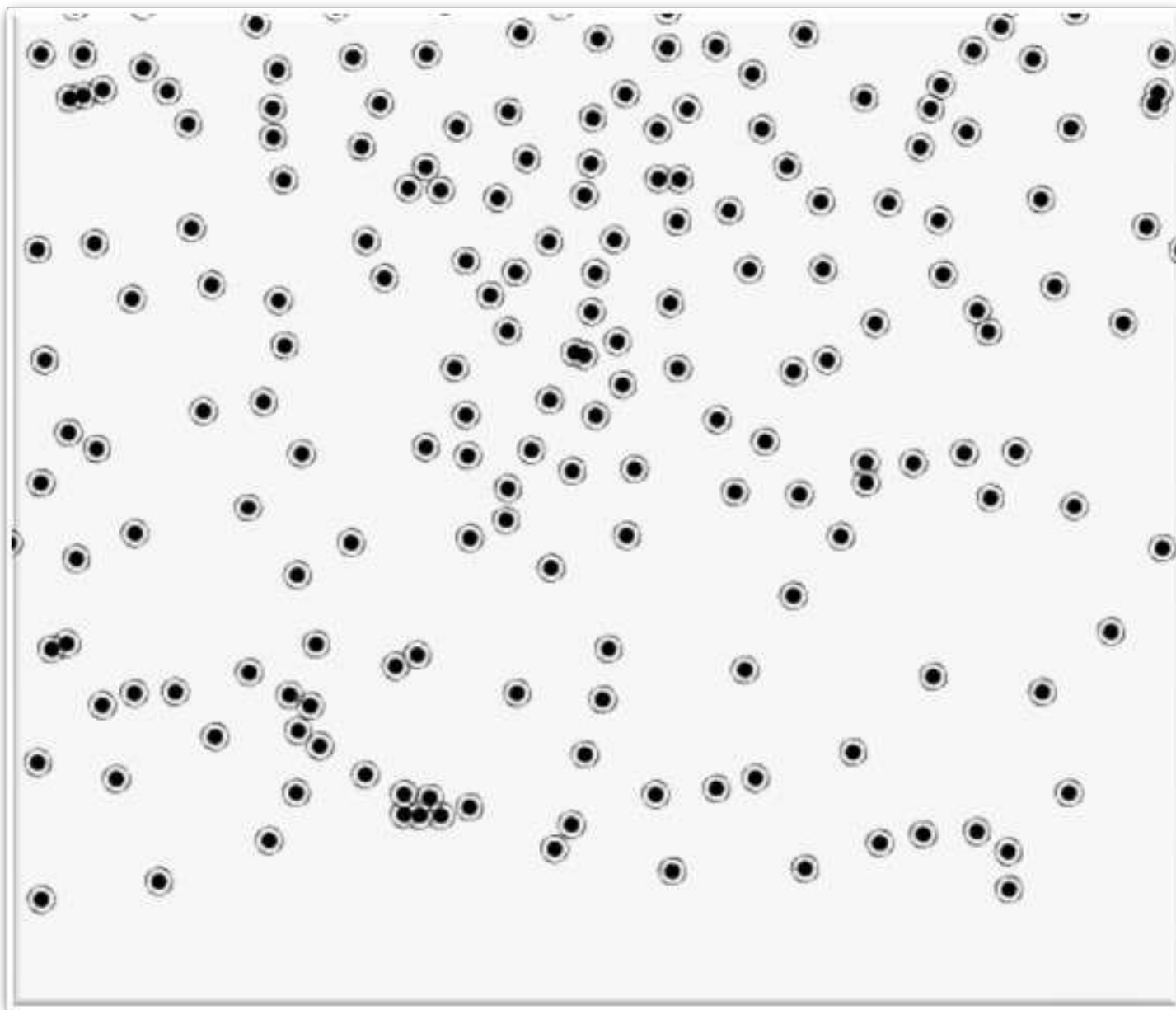


```
http://www.geoservicos.ibge.gov.br/geoserver/gn/wms?  
service=WMS&  
version=1.1.0&  
request=GetMap&  
layers=IBGE:e1000_cidade&  
styles=&  
bbox=-44.394,-23.256,-41.622,-20.928&  
width=512&  
height=436&  
srs=EPSG:4326&  
format=image/png
```



Resposta:

IBGE





- A resposta a uma solicitação GetCapabilities é um arquivo XML com informações sobre operações e serviços oferecidos pelo servidor

Parâmetros da solicitação	Obrigatório/ Opcional	Descrição
VERSION	Opcional	Versão solicitada
SERVICE	Obrigatório	Tipo de serviço
REQUEST	Obrigatório	Nome da solicitação
UPDATESEQUENCE	Opcional	Número da sequência ou cadeia para o controle cachê
FORMAT	Opcional	Formato de saída dos metadatos do serviço



- **GetCapabilities**
 - SERVICE
 - Deve assumir o valor WMS, pois no mesmo servidor pode haver outros serviços diferentes;
 - REQUEST
 - Deve-se usar o valor “GetCapabilities” para esta versão



Solicitação GetCapabilities

http://200.255.94.99:8080/geoserver/wms?
service=WMS&
version=1.1.1&
request=GetCapabilities

The screenshot shows a web browser window displaying the XML response for a GetCapabilities request. The URL in the address bar is `http://200.255.94.99:8080/geoserver/wms?service=WMS&version=1.1.1&request=GetCapabilities`. The XML content is partially visible, showing the root element `GetCapabilitiesResponse` and several `Layer` elements. The first `Layer` element is for the `WMS` service, and the second is for the `WMS` service. The `Layer` elements contain information about the service, including the `name`, `title`, `abstract`, `keywords`, `isPublic`, `isQueryable`, `isGetCapabilities`, and `isGetMap` attributes. The `isGetMap` attribute is set to `true` for both layers.

Resposta GetCapabilities



Principais seções do documento Capabilities:

- **Service** – contém informações básicas como nome e tipo de serviço. Indica a versão do WFS suportada;
- **Request** – Descreve as operações que o servidor reconhece assim como os parâmetros e formatos de saída de cada operação;
- **Layer** – Lista os Layer e projeções disponíveis no servidor.

- Através da operação GetCapabilities o usuário irá acessar os metadados do Geoserviço, assim a resposta desta operação deve conter dados como sistema de coordenadas, retângulo envolvente, dados disponíveis no serviço, etc...

- Os dados obtidos no documento recuperado pela operação GetCapabilities, são utilizados na solicitação de um mapa através da operação GetMap.



Solicitação GetFeatureInfo



- A operação GetFeatureInfo é utilizada pelas aplicações que acessam os serviços WMS, para obter dados alfanúmericos dos elementos geográficos disponíveis.

– Formato Suportado:

- TEXT – Saída em arquivo texto simples. Padrão.
- HTML – Usa *templates* HTML definidos no servidor.

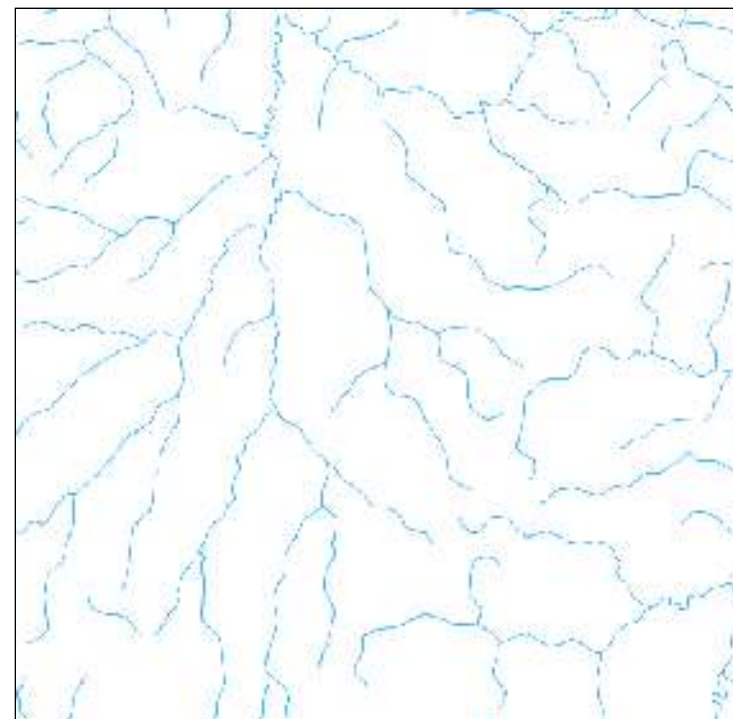


Tipo de solo: Urbano
Código: 25841



- Implementação da OGC que descreve a linguagem utilizada para produzir estilos relacionados a apresentação de feições geográficas;
- Documento visto como uma extensão do WMS ;
- Produzido com a linguagem XML para personalizar a aparência de um mapa.
- Os dados disponibilizados no servidor de Geoserviços possuem um estilo de apresentação padrão, que pode ser configurado através de um arquivo SLD;

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0"
  ....">
<NamedLayer>
  <Name>CURSO_DAGUA</Name>
  <UserStyle>
    <Name>CURSO_DAGUA_style</Name>
    <Title>Rios</Title>
    <FeatureTypeStyle>
      <Rule>
        <LineSymbolizer>
          <Stroke>
            <CssParameter name="stroke">#00CCFF</CssParameter>
            <CssParameter name="stroke-opacity">1</CssParameter>
            <CssParameter name="stroke-width">0.2</CssParameter>
          </Stroke>
        </LineSymbolizer>
      </Rule>
    </FeatureTypeStyle>
  </UserStyle>
</NamedLayer>
</StyledLayerDescriptor>
```



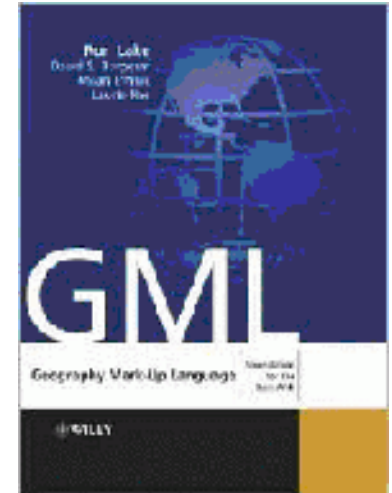


- Existem dois tipos de serviço WFS, o básico e o transacional;
- As operações fundamentais do WFS básico são: GetCapabilities, DescribeFeatureType e GetFeature. É um serviço Web só de leitura de feições;
- O WFS transacional engloba todas as operações do WFS básico mais operações de transação;
- Com o WFS transacional é possível atualizar e remover as feições no servidor, ou seja, o acesso é de leitura e escrita;
- O WFS transacional também é conhecido como WFS-T

- As operações realizadas no servidor WFS-T são atômicas, ou seja, caso ocorra alguma falha durante a transação os elementos permanecerão inalterados, no mesmo estado de antes do início da operação;
- Com o WFS-T é possível realizar o bloqueio (*lock*) de feições de uma área de trabalho onde o usuário está realizando edições.



- **GML – Geographic Markup Language**
 - Geographic Markup Language é um documento XML para a modelagem (escrito em XML Schema) , transporte e armazenamento de informação geográfica.
 - GML é uma camada semântica sobre o XML.
 - GML fornece um conjunto de classes de objetos para descrever elementos feições geográficas, sistemas de referência espacial, geometria, topologia, tempo, unidades de medida e valores gerais.



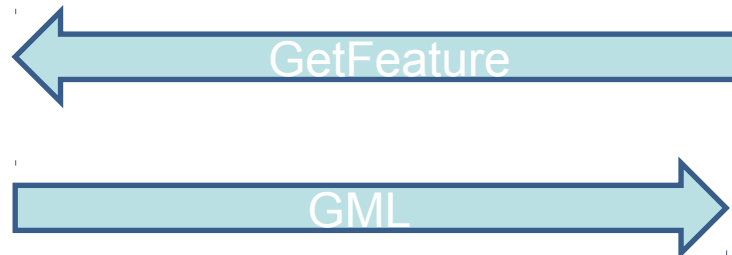
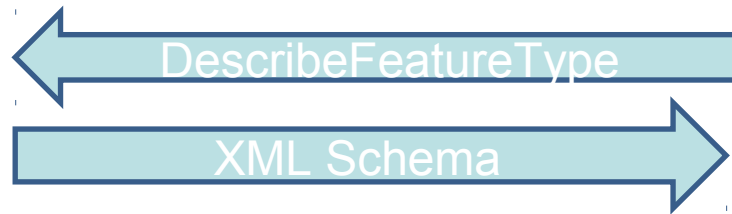
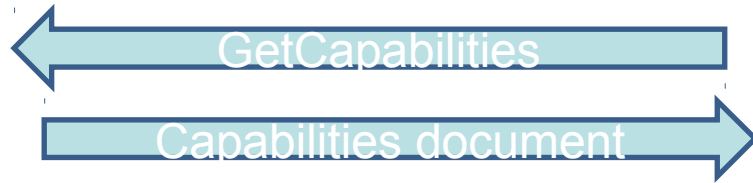


WFS - Operações

A norma define as seguintes operações:

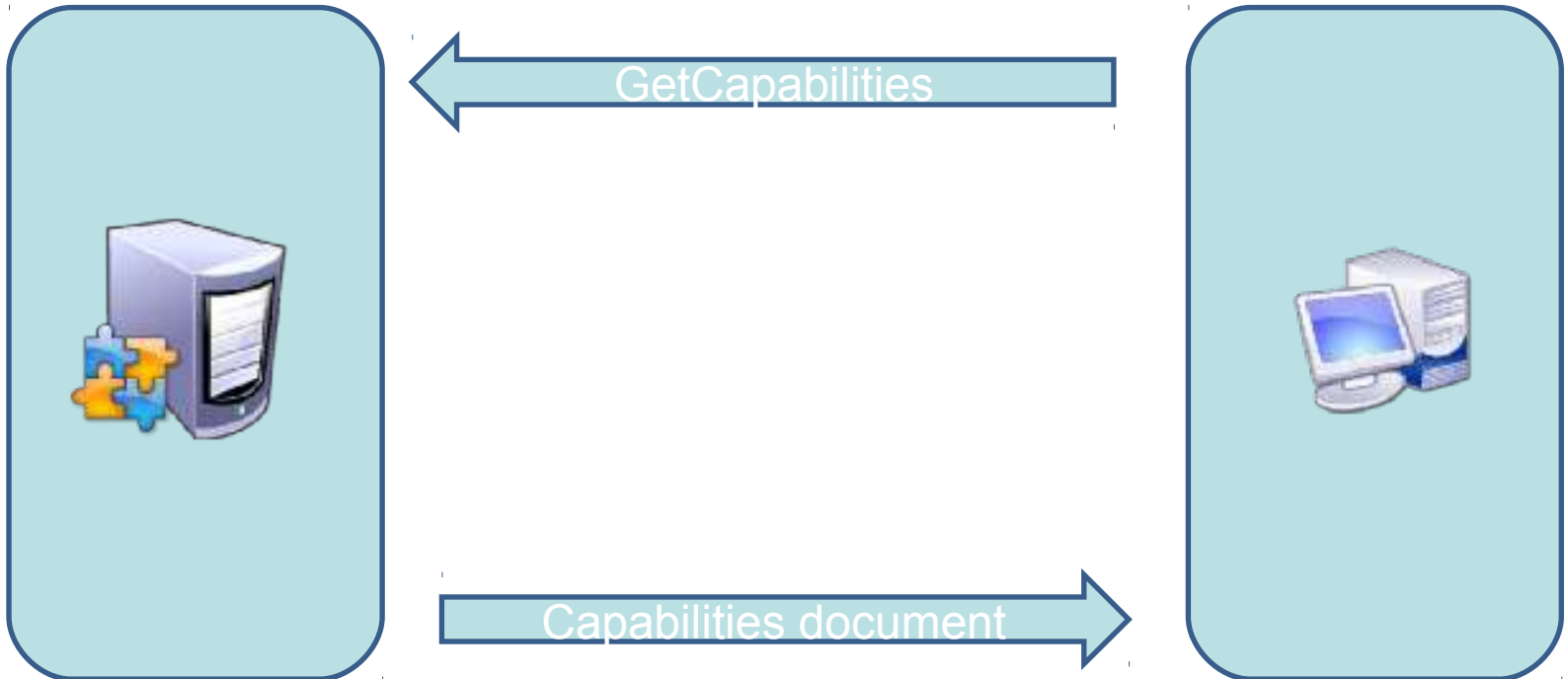
- *GetCapabilities*: Retorna metadados do servidor;
- *DescribeFeatureType* : Retorna informações e atributos de um conjunto de dados;
- *GetFeature*: Retorna o dado desejado, incluindo a geometria e os atributos;
- *LockFeature*: Bloqueia a edição da feição;
- *Transaction*: Permite a edição da feição com inclusão, exclusão e atualização;
- *GetGMLObject* (versão 1.1.0) – Retorna feições geoespaciais;







```
<?xml version="1.0" ?>  
<GetCapabilities  
  service="WFS"  
  xmlns="http://www.opengis.net/wfs"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd"/>
```





```
<?xml version="1.0" ?>  
<DescribeFeatureType  
version="1.1.0"  
service="WFS"  
xmlns="http://www.opengis.net/wfs"  
xmlns:ns01="http://www.server01.com/ns01"  
xmlns:ns02="http://www.server02.com/ns02"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd">  
<TypeName>ns01:TreesA_1M</TypeName>  
<TypeName>ns02:RoadL_1M</TypeName>  
</DescribeFeatureType>
```



DescribeFeatureType



XML Schema

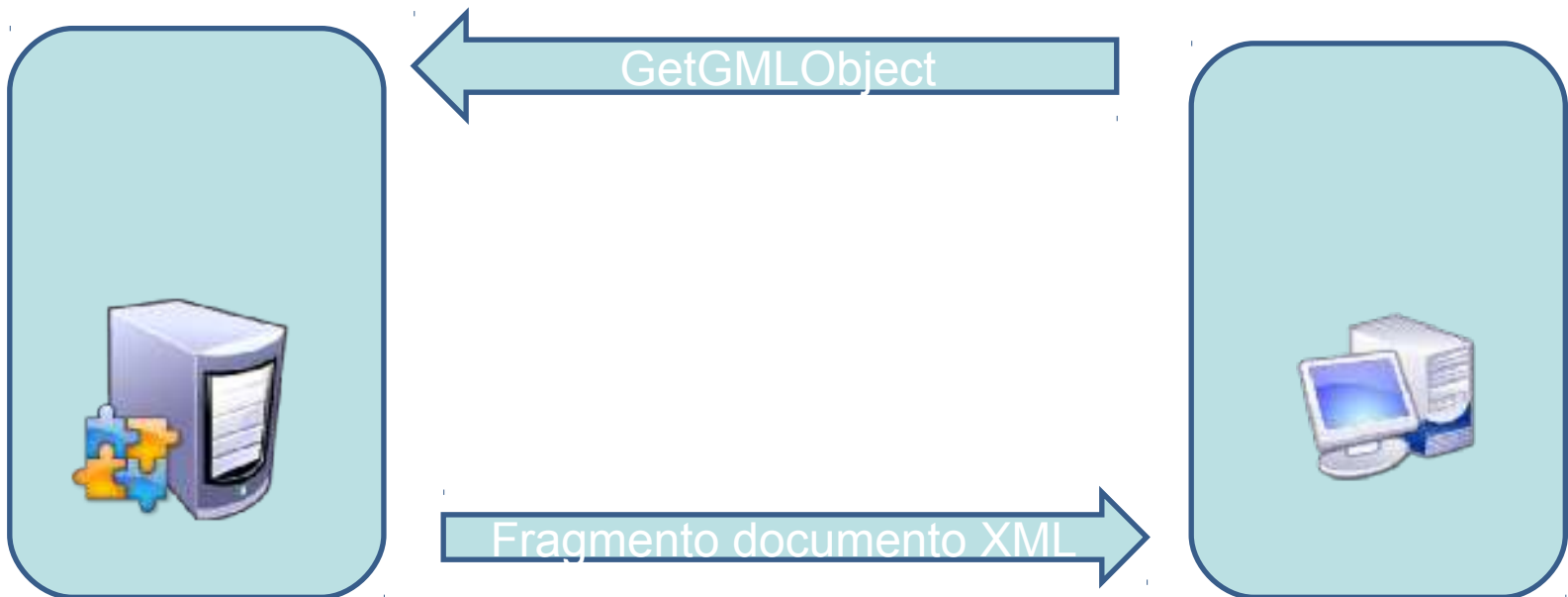


```
<?xml version="1.0" ?>  
<GetFeature  
  version="1.1.0"  
  service="WFS"  
  handle="Example Query"  
  xmlns="http://www.opengis.net/wfs"  
  xmlns:ogc="http://www.opengis.net/ogc"  
  xmlns:gml="http://www.opengis.net/gml"  
  xmlns:myns="http://www.someserver.com/myns"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd">  
  <Query typeName="myns:ROADS">  
    <wfs:PropertyName>myns:path</wfs:PropertyName>  
    <wfs:PropertyName>myns:lanes</wfs:PropertyName>  
    <wfs:PropertyName>myns:surfaceType</wfs:PropertyName>  
    <ogc:Filter>  
      <ogc:Within>  
        <ogc:PropertyName>myns:path</ogc:PropertyName>  
        <gml:Envelope srsName="EPSG:63266405">  
          <gml:lowerCorner>50 40</gml:lowerCorner>  
          <gml:upperCorner>100 60</gml:upperCorner>  
        </gml:Envelope>  
      </ogc:Within>  
    </ogc:Filter>  
  </Query>  
</GetFeature>
```



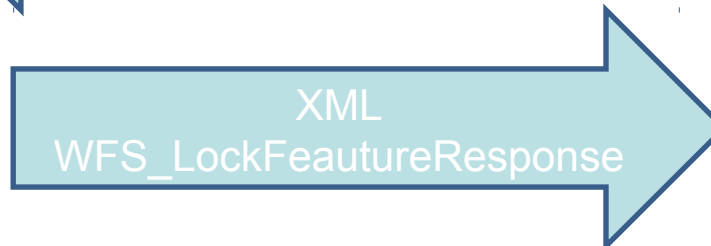


```
<?xml version="1.0"?>  
<wfs:GetGmlObject  
  xmlns:wfs="http://www.opengis.net/wfs"  
  xmlns:ogc="http://www.opengis.net/ogc"  
  xmlns:gml="http://www.opengis.net/gml"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd"  
  service="WFS"  
  version="1.1.0"  
  outputFormat="text/xml; subtype=gml/3.1.1"  
  traverseXlinkDepth="2"  
  traverseXlinkExpiry="1">  
  <ogc:GmlObjectId gml:id="t1"/>  
</wfs:GetGmlObject>
```





```
<?xml version="1.0" ?>
<LockFeature
version="1.1.0"
service="WFS"
expiry="4"
lockAction="SOME"
xmlns="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml"
xmlns:myns="http://www.someserver.com/myns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd">
<Lock handle="LOCK1" typeName="myns:BuiltUpA_1M">
<ogc:Filter>
<ogc:Within>
<ogc:PropertyName>BuiltUpA_1M/wkbGeom</ogc:PropertyName>
<gml:Polygon gid="1"
srsName="http://www.opengis.net/gml/epsg.xml#63266405">
<gml:exterior>
<gml:LinearRing>
<gml:posList>-95.7 38.1 -97.8 38.2 ...</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</ogc:Within>
</ogc:Filter>
</Lock>
</LockFeature>
```





```
<?xml version="1.0"?>
<wfs:Transaction
version="1.1.0"
service="WFS"
xmlns="http://www.someserver.com/myns"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.someserver.com/myns
http://www.someserver.com/wfs/cwwfs.cgi?
request=describefeaturetype&typename=InWaterA_1M.xsd
http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd">
<wfs:Insert idgen="UseExisting">
<InWaterA_1M gml:id="INW1">
<wkbGeom>
<gml:Polygon gml:id="P1"
srsName="http://www.opengis.net/gml/srs/epsg.xml#63266405">
<gml:exterior>
<gml:LinearRing>
<gml:posList>-98.54 24.26 ...</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</wkbGeom>
<id>150</id>
<f_code>ABCDE</f_code>
<hyc>152</hyc>
<tileId>250</tileId>
<facId>111</facId>
</InWaterA_1M>
<InWaterA_1M gml:id="INW2">
</wfs:Insert>
</wfs:Transaction>
```



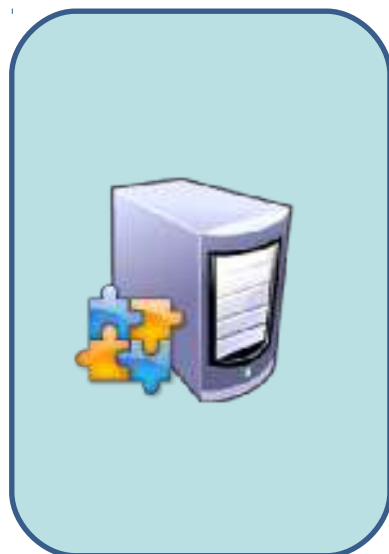
Transaction (INSERT)

XML TransactionResponse

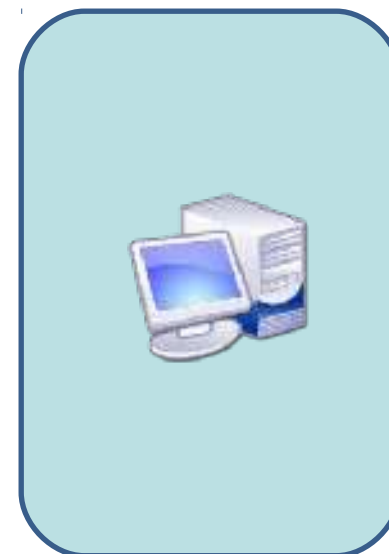




```
<?xml version="1.0" ?>
<wfs:Transaction
version="1.1.0"
service="WFS"
xmlns="http://www.someserver.com/myns"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd">
<wfs:Update typeName="BuiltUpA_1M">
<wfs:Property>
<wfs:Name>populationType</wfs:Name>
<wfs:Value>CITY</wfs:Value>
</wfs:Property>
<ogc:Filter>
<ogc:GmlObjectId gml:id="BuiltUpA_1M.1013"/>
<ogc:GmlObjectId gml:id="BuiltUpA_1M.34"/>
<ogc:GmlObjectId gml:id="BuiltUpA_1M.24256"/>
</ogc:Filter>
</wfs:Update>
```



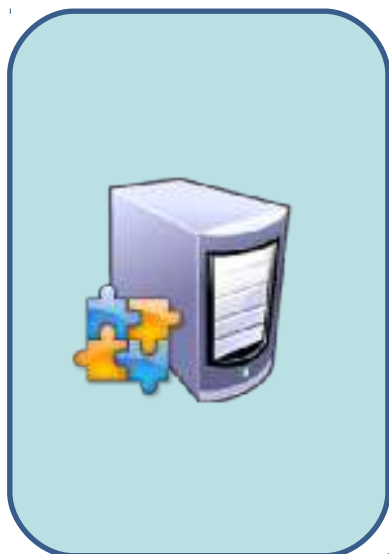
Transaction (UPDATE)



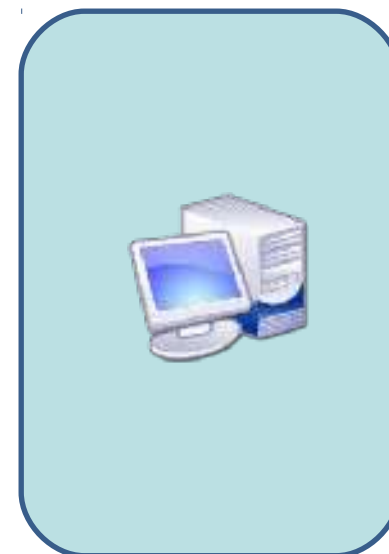
XML TransactionResponse



```
<?xml version="1.0" ?>
<wfs:Transaction
version="1.1.0"
service="WFS"
xmlns:myns="http://www.someserver.com/myns"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd">
<wfs:Delete typeName="myns:InWaterA_1M">
<ogc:Filter>
<ogc:GmlObjectId gml:id="InWaterA_1M.1013"/>
<ogc:GmlObjectId gml:id="InWaterA_1M.10"/>
<ogc:GmlObjectId gml:id="InWaterA_1M.13"/>
<ogc:GmlObjectId gml:id="InWaterA_1M.140"/>
<ogc:GmlObjectId gml:id="InWaterA_1M.5001"/>
<ogc:GmlObjectId gml:id="InWaterA_1M.2001"/>
</ogc:Filter>
</wfs:Delete>
</wfs:Transaction>
```



Transaction (DELETE)



XML TransactionResponse



```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=GetCapabilities
```

Principais seções do documento Capabilities:

- A resposta é um documento XML;
- **ServiceIdentification** – contém informações básicas como nome e tipo de serviço. Indica a versão do WFS suportada;
- **ServiceProvider** – Informações de contato da instituição responsável pelo servidor como telefone, email, endereço, etc.
- **OperationsMetadata** – Descreve as operações que o servidor reconhece e os respectivos parâmetros;
- **FeatureTypeList** – Lista os tipos de feições disponíveis no servidor. Também informa a projeção padrão e o área geográfica do dado.
- **Filter_Capabilities** – Lista os filtros disponíveis para pesquisar os dados (Operadores Espaciais – *Equals*, *Touches*, Operadores de Comparação – *LessThan*, *GreaterThan* e outros)



- WFS 1.1.0 retorna GML3 como padrão enquanto WFS 1.0.0 retorna GML2;
- WFS 1.1.0 especifica o Sistema de Referência Espacial como urn:x-ogc:def:crs:EPSG:XXXX enquanto na WFS 1.0.0 <http://www.opengis.net/gml/srs/epsg.xml#XXXX>.
- WFS 1.1.0 permite a reprojeção dos dados *on-the-fly* ao contrário do WFS 1.0.0;
- WFS 1.0.0 retorna coordenada geográfica como longitude/latitude (x/y) enquanto WFS 1.1.0 latitude/longitude (y/x);
- O Geoserver adota a seguinte convenção:
 - SRS - EPSG:xxxx - latitude/longitude ;
 - SRS - <http://www.opengis.net/gml/srs/epsg.xml#xxxx> - longitude/lat
 - SRS - urn:x-ogc:def:crs:EPSG:xxxx - latitude/longitude



O objetivo desta operação é obter informações de um tipo de feições antes de solicitar os dados.

Formas de Solicitação:

1. Retorna todos os tipos de feições do servidor

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=DescribeFeatureType
```

2. Retorna um tipo de feições específico:

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=DescribeFeatureType&  
typeName=namespace:featuretype
```



Solicitação GetFeature

1. Forma mais simples de solicitação *GetFeature*, sem nenhum argumento:

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=GetFeature&  
typeName=namespace:featuretype
```

Retorna todo o conjunto de dados

2. Solicitação *GetFeature* para recuperar um elemento específico :

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=GetFeature&  
typeName=namespace:featuretype&  
featureID=feature
```

**Retorna uma feição, indicada pelo seu ID no parâmetro
featureID**



3. Solicitação *GetFeature* para recuperar um número máximo de elementos:

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=GetFeature&  
typeName=namespace:featuretype&  
maxFeatures=N
```

Caso o ID do elemento não seja conhecido e deseja-se limitar a quantidade de elementos retornados



4. Solicitação *GetFeature* para recuperar um número máximo de elementos ordenados em função de um atributo (o padrão é ordenação ascendente) :

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=GetFeature&  
typeName=namespace:featuretype&  
maxFeatures=N  
sortBy=property
```

Se os dados não forem ordenados, os mesmos serão recuperados aleatoriamente, em função da organização interna da fonte de dados



5. Solicitação *GetFeature* para recuperar um número máximo de elementos ordenados em função de um atributo (ordenação descendente) :

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=GetFeature&  
typeName=namespace:featuretype&  
maxFeatures=N  
sortBy=property+D
```

Alguns servidores exigem a indicação do tipo de ordenação, assim para ordenação ascendente basta substituir o D pelo A. Não é obrigatório o uso dos parâmetros maxFeatures e sortBy ao mesmo tempo



6. Solicitação *GetFeature* para recuperar um tipo de feição com um ou mais atributos específicos :

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=GetFeature&  
typeName=namespace:featuretype&  
propertyName=property
```

Com o parâmetro *propertyName* podemos restringir a quantidade de atributos nas feições recuperadas. Útil nos casos em que não é necessário a recuperação dos elementos com todos os atributos. Para passar mais de um atributo devemos separar os nomes por vírgulas (*propertyName=property1,property2*) .



7. Podemos limitar nossa solicitação a uma região específica com o parâmetro *bbox* (bounding box – retângulo envolvente):

```
http://www.exemplo.com/wfs?  
service=wfs&  
version=1.1.0&  
request=GetFeature&  
typeName=namespace:featuretype&  
bbox=a1,b1,a2,b2
```

Os valores a1,b1,a2,b2 são as coordenadas dos cantos do retângulo envolvente, sendo que o canto inferior deve vir primeiro, independente de aparecer da esquerda para direita ou o contrário.



- Codificação de filtros –
 - Uma expressão *Filter Encoding* (filtro de expressão) é uma construção usada para determinar os valores dos atributos de um tipo de objeto, com o objetivo de selecionar um subconjunto de instâncias em que se pretendem operar a qualquer momento;
 - Esta especificação visa descrever um catálogo de codificação XML *Query Language* da OGC, como um sistema independente de representação de predicado de consultas;
 - É utilizada em conjunto com a SLD para selecionar um conjunto específico de objetos sobre os quais deseja-se aplicar um estilo.

O XML é um formato para a criação de documentos com dados organizados de forma hierárquica, sendo recomendado para uso em documentos de **texto formatados, imagens vetoriais ou bancos de dados.**

Pela sua portabilidade, já que é um formato que não depende das plataformas de hardware ou de software, um banco de dados pode, através de uma aplicação, escrever em um arquivo XML, e um outro banco distinto pode ler então estes mesmos dados.





GOVERNO
FEDERAL

e-PING

Interoperabilidade de:

Tecnologia

Dados

Organização e
Intercâmbio de
Informações

Informações

e-PING



Organização e Intercâmbio de Informações

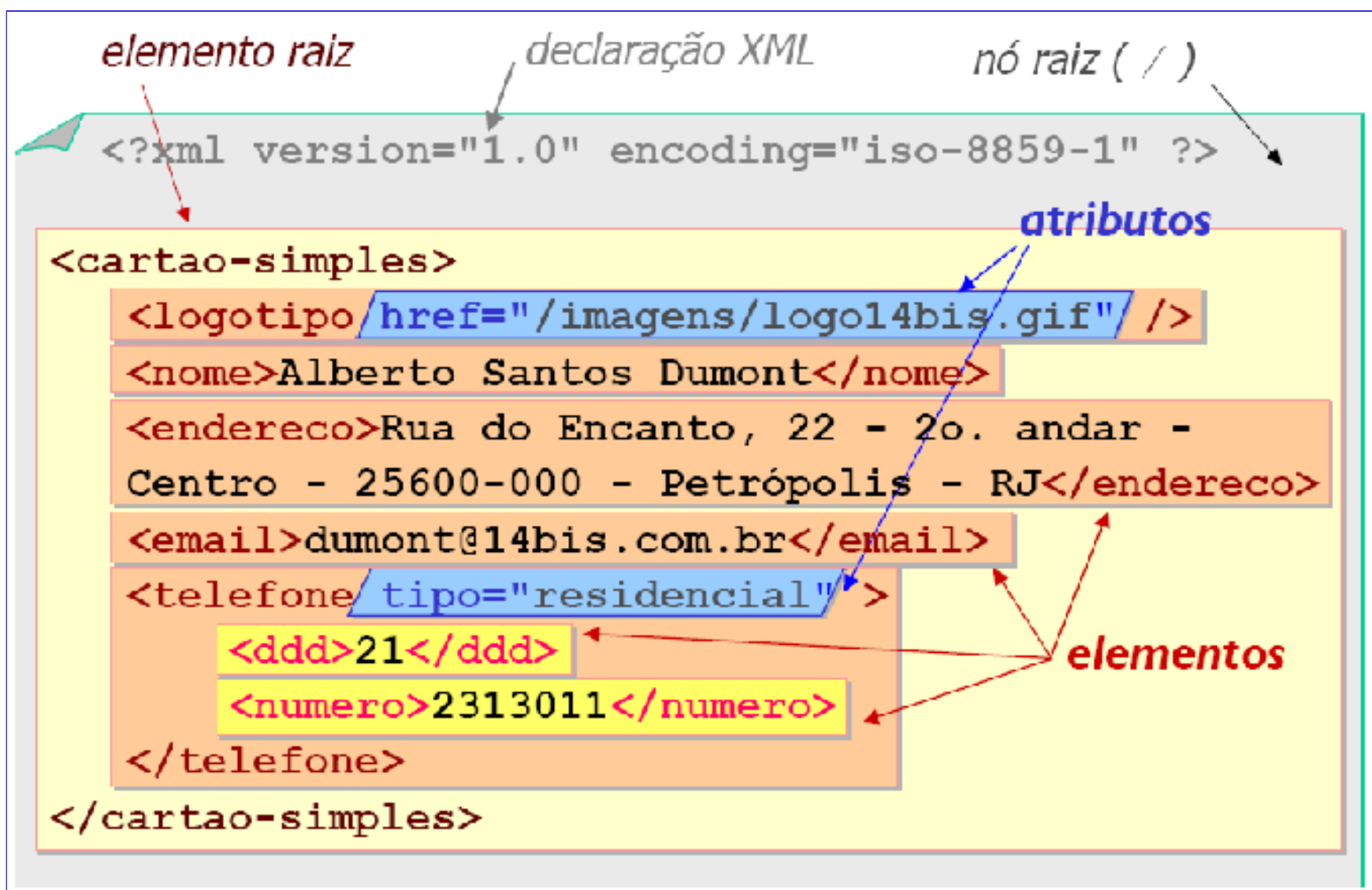
Uso de XML para intercâmbio de dados.

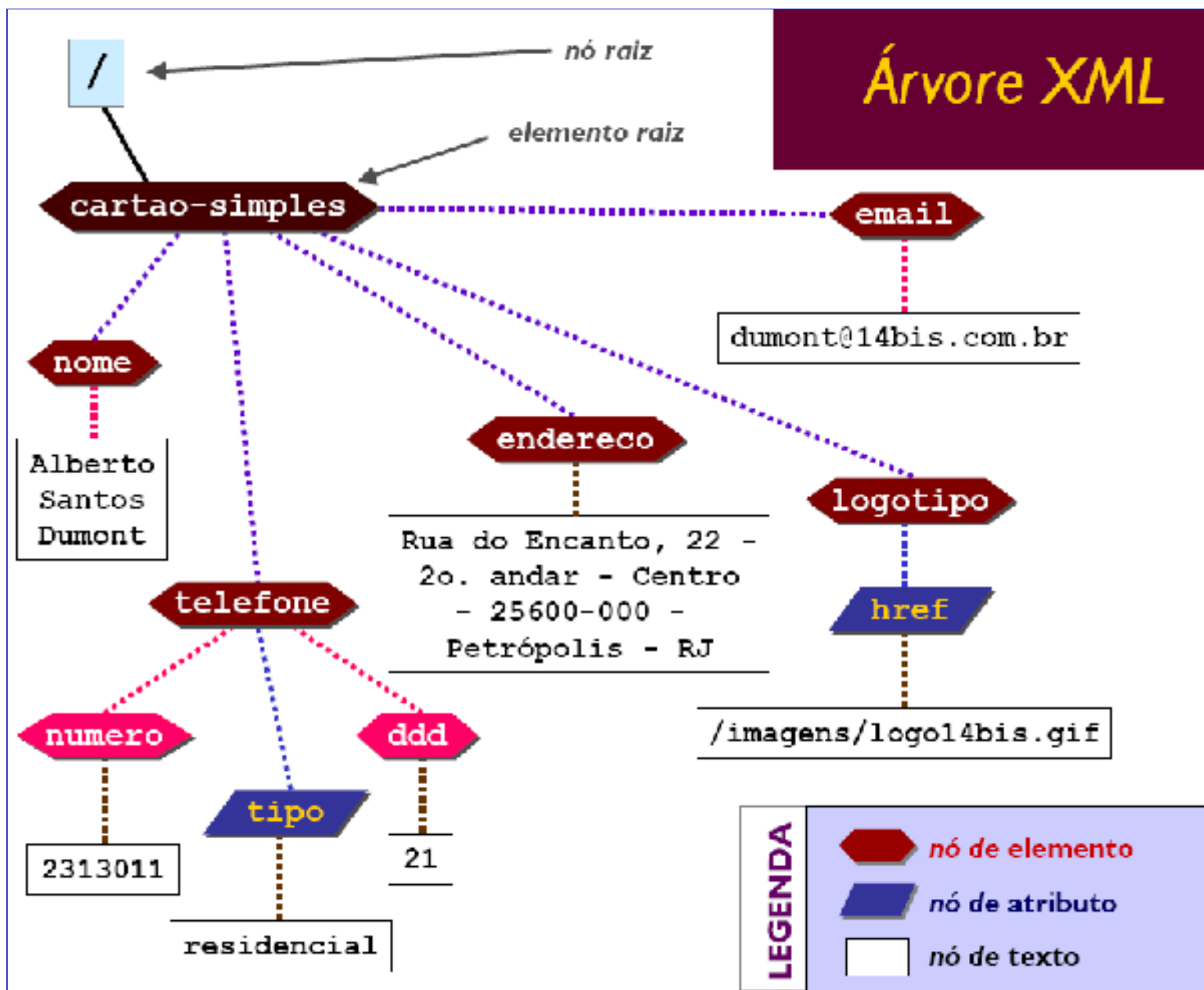
Uso de XML Schema e da UML (quando for o caso) para definição dos dados para intercâmbio.

Uso de XSL para transformação de dados.

Uso de um padrão de metadados para a gestão de conteúdos eletrônicos.

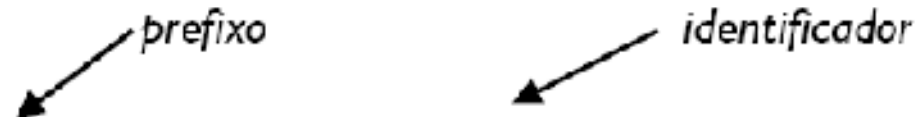
Partes de um Documento XML






- **Limita o escopo de elementos**
 - Evita conflitos quando duas linguagens se cruzam no mesmo documento
- **Consiste da associação de um *identificador* a cada elemento/atributo da linguagem, que pode ser**
 - *herdado* através do escopo de uma sub-árvore
 - *atribuído explicitamente através de um **prefixo***
- **Exemplo**

```
<cadastro xmlns:firma="01.234.567/0001-99">  
  <nome>Severino Severovitch</nome>  
  <firma:nome>Sibéria Informática Ltda.</firma:nome>  
  <email>bill@norte.com.br</email>  
</cadastro>
```



Este elemento <nome> pertence a outro namespace



- Porque é um **padrão aberto**
 - Facilidade para converter para formatos proprietários
- Porque é **texto**
 - Fácil de ler, fácil de processar, menos incompatibilidades
- Porque promove a **separação** entre estrutura, conteúdo e apresentação
 - Facilita geração de dados para visualização dinâmica
 - Evita repetição de informação / simplifica manutenção
- Porque permitirá **semântica** na Web
 - Elementos HTML não carregam significado, apenas dicas de formatação: mecanismos de busca ficam prejudicados
 - Solução com XML dependerá de suporte dos clientes

- Um esquema é essencial para que haja **comunicação usando XML**
 - Pode ser estabelecido "informalmente" (via software)
 - Uso formal permite validação usando ferramentas genéricas de manipulação de XML
- Soluções

DTD

```
<!ELEMENT contato  
    (nome, email, telefone)>  
<!ATTLIST contato  
    codigo NMTOKEN #REQUIRED>
```

*Simple mas não é XML
Não suporta namespaces
Limitado quando a tipos de dados*

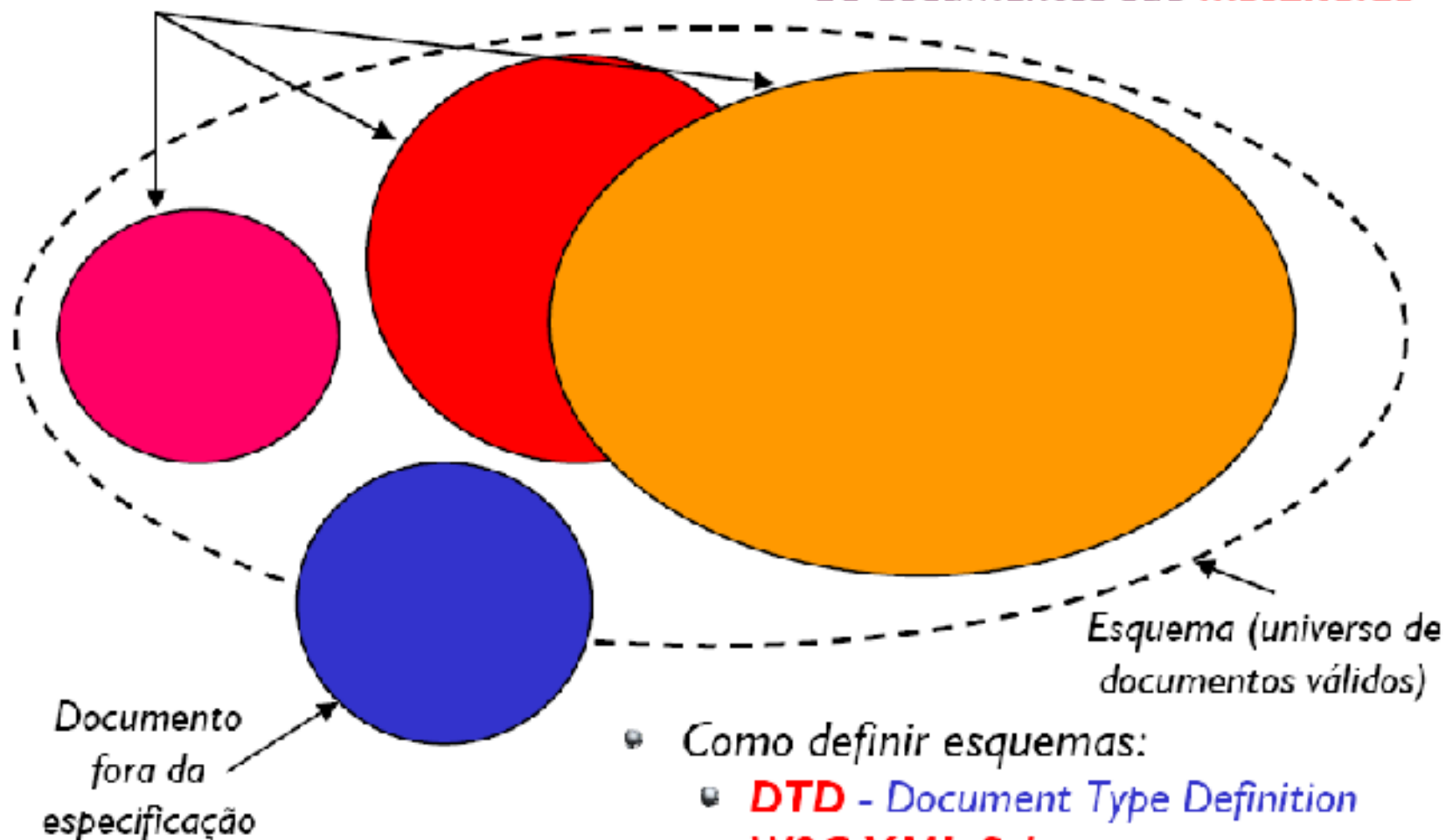
XSchema

```
<xsd:schema  
    xmlns:xsd=".../XMLSchema">  
<xsd:element name="contato">  
    <xsd:complexType>  
        <xsd:attribute name="codigo"  
            use="required">
```

*É XML, porém mais complexo
Suporta namespaces
Permite definição de tipos*

Documentos que aderem à especificação (válidos)

- O esquema representa uma **classe**
- Os documentos são **instâncias**



- *Para que possa ser manipulado como uma árvore, um documento XML precisa ser **bem formado***
 - *Documentos que não são bem formados não são documentos XML*
- *Documentos bem-formados obedecem as regras de construção de documentos XML genéricos*
- *Regras incluem*
 - *Ter um, e apenas um, elemento raiz*
 - *Valores dos atributos estarem entre aspas ou apóstrofes*
 - *Atributos não se repetirem*
 - *Todos os elementos terem etiqueta de fechamento*
 - *Elementos estarem corretamente aninhados*

- *Um XML bem construído pode não ser **válido** em determinada aplicação*
- *Aplicação típica pode esperar que*
 - *elementos façam parte de um **vocabulário** limitado,*
 - *certos atributos tenham **valores** e **tipos** definidos,*
 - *elementos sejam organizados de acordo com uma determinada estrutura **hierárquica**, etc.*
- *É preciso **especificar** a linguagem!*
 - ***Esquema**: modelo que descreve todos os elementos, atributos, entidades, suas relações e tipos de dados*
- *Um documento XML é considerado válido **em relação a um esquema** se obedecer todas as suas regras*

- *XML não estabelece nenhum vocabulário mas apenas regras mínimas de estrutura*
- *Para criar o menor arquivo XML possível:*
 - *Abra um editor de textos*
 - *Salve o arquivo com extensão .xml*
 - *Escreva um elemento raiz vazio*

<x/>

- *Salve o arquivo*
- *Abra no Internet Explorer*

```
<nome> Fulano de Tal </nome>
```

Elemento raiz

- *Um elemento*

```
<nome> Fulano de Tal </nome>
```

Etiqueta
inicial

Conteúdo do
Elemento

Etiqueta
final



`<nome></nome>`

The diagram shows a light blue rectangular box with a folded top-left corner. Inside the box is an orange horizontal bar containing the text `<nome></nome>` in blue font.



`<nome />`

The diagram shows a light blue rectangular box with a folded top-left corner. Inside the box is an orange horizontal bar containing the text `<nome />` in blue font.

fulano_de_tal.xml

```
<contato>  
  <nome>Fulano de Tal</nome>  
  <email>fulano@site.com.br</email>  
  <telefone>  
    <ddd>11</ddd>  
    <numero>8765 4321</numero>  
  </telefone>  
</contato>
```

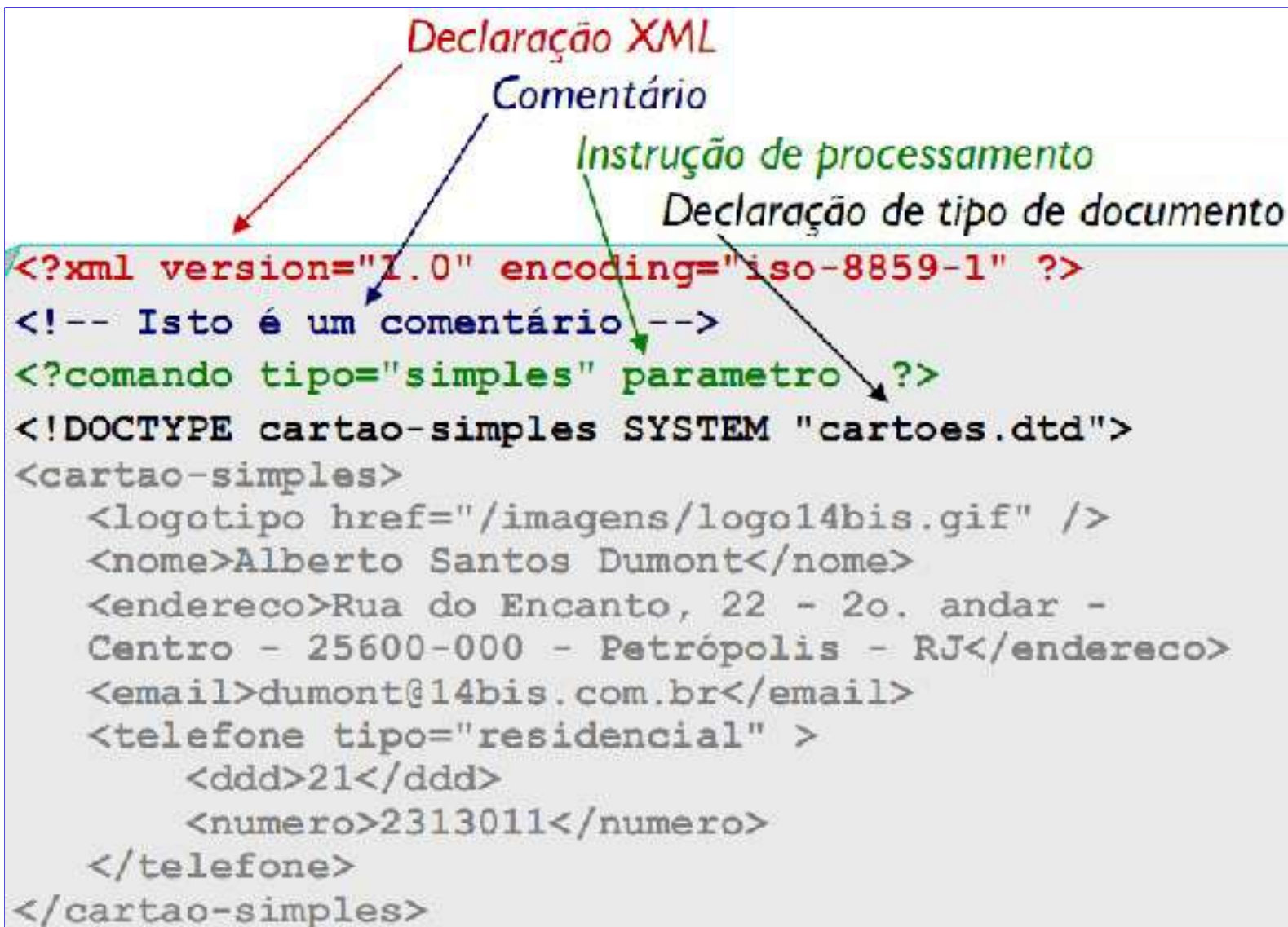

- *Árvore XML*
 - nós,
 - raiz,
 - galhos e
 - folhas
- *Prólogo*
- *Comentários*
- *Instruções de processamento*
- *Elementos*
- *Atributos*
- *Nós de texto*
- *Entidades*

Um documento XML pode ser representado como uma árvore.
A estrutura é formada por vários nós

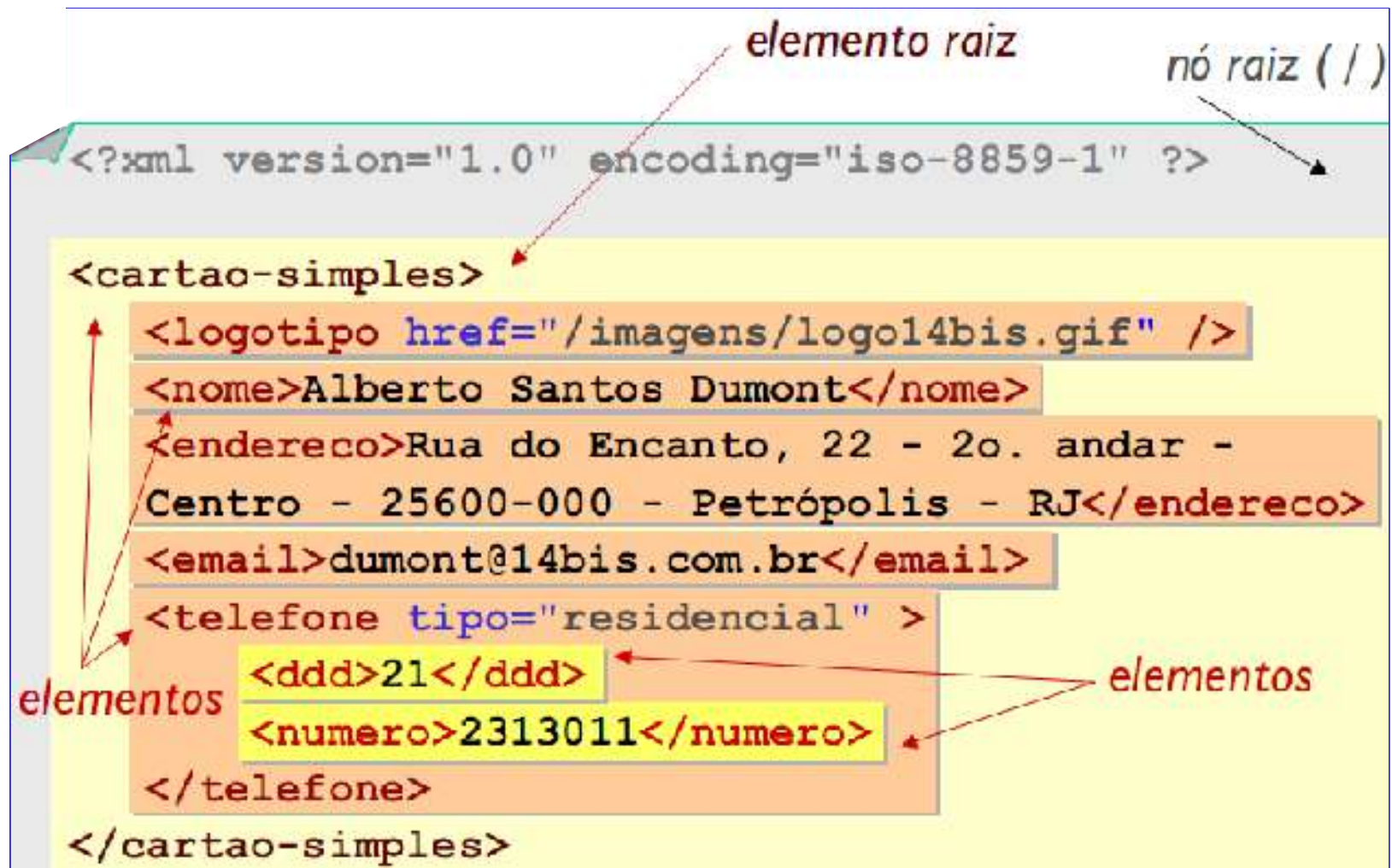
```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!-- Isto é um comentário -->
<cartao-simples>
  <logotipo href="/imagens/logo14bis.gif" />
  <nome>Alberto Santos Dumont</nome>
  <endereco>Rua do Encanto, 22 - 2o. andar -
  Centro - 25600-000 - Petrópolis - RJ</endereco>
  <email>dumont@14bis.com.br</email>
  <telefone tipo="residencial">
    <ddd>21</ddd>
    <numero>2313011</numero>
  </telefone>
</cartao-simples>
```

informações usadas pela processador XML

um "nó" pode ser ...
um elemento,
um atributo,
um bloco de texto
um comentário,
uma instrução,
uma declaração,
uma entidade, ...



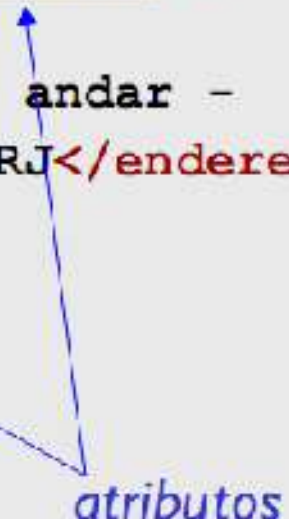
Nó raiz e elementos



```
<?xml version="1.0" encoding="iso-8859-1" ?>

<cartao-simples>
  <logotipo href="/imagens/logol4bis.gif" />
  <nome>Alberto Santos Dumont</nome>
  <endereco>Rua do Encanto, 22 - 2o. andar -
Centro - 25600-000 - Petrópolis - RJ</endereco>
  <email>dumont@l4bis.com.br</email>
  <telefone tipo="residencial" />
    <ddd>21</ddd>
    <numero>2313011</numero>
  </telefone>
</cartao-simples>
```

atributos

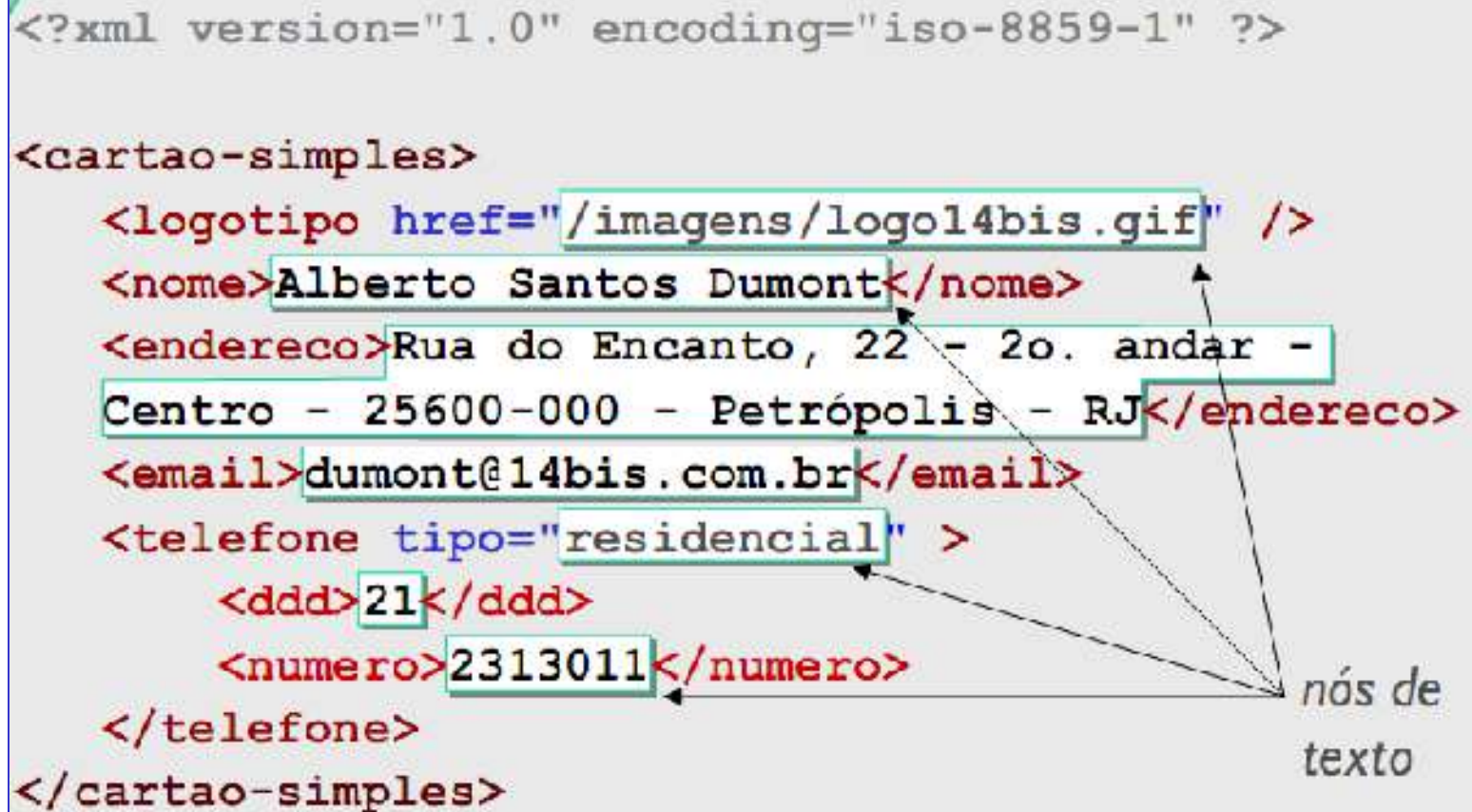


- Não podem ter descendentes (são as folhas da árvore)

```
<?xml version="1.0" encoding="iso-8859-1" ?>

<cartao-simples>
  <logotipo href="/imagens/logol4bis.gif" />
  <nome>Alberto Santos Dumont</nome>
  <endereco>Rua do Encanto, 22 - 2o. andar -
  Centro - 25600-000 - Petrópolis - RJ</endereco>
  <email>dumont@14bis.com.br</email>
  <telefone tipo="residencial" >
    <ddd>21</ddd>
    <numero>2313011</numero>
  </telefone>
</cartao-simples>
```

nós de texto



- São constantes associadas a um valor de texto
 - Podem aparecer em qualquer lugar do documento
- Substituídas **durante** o processamento do documento
- Sintaxe:
 - &ENTIDADE;
- Exemplo:
 - &data_de_hoje;
- Entidades pré-definidas:
 - **<** que corresponde a <
 - **>** que corresponde a >
 - **&** que corresponde a &
 - **"** que corresponde a "
 - **'** que corresponde a '

▪ *Regras básicas*

- *Etiqueta inicial e final têm que ter o mesmo nome (considerando diferença de maiúscula e minúscula)*
- *Não pode haver espaço depois do < nas etiquetas iniciais nem depois do </ nas finais*
- *Atributos têm sempre a forma nome="valor" ou nome = 'valor':*
 - *aspas podem ser usadas entre apóstrofes e apóstrofes podem ser usados entre aspas*
 - *aspas e apóstrofes não podem ser neutralizados mas sempre podem ser representados pelas entidades ' e "*
- *Não pode haver atributos na etiqueta final*

- *Elementos mal formados*

```
<Profissão>Arquiteto</profissão>
```

```
<TR><TD>item um</td></tr>
```

```
<ÃÁÃÈÉ>139.00</ãáãèé>
```

- *Há várias maneiras de representar a mesma informação em XML*

```
<data>23/02/1998</data>
```

```
<data dia="23" mes="02" ano="1998" />
```

```
<data>
```

```
  <dia>23</dia>
```

```
  <mes>02</mes>
```

```
  <ano>1998</ano>
```

```
</data>
```

- *Nomes de atributos e elementos*
- *Podem **conter***
 - *qualquer caractere alfanumérico ou ideograma*
 - *. (ponto)*
 - *- (hífen)*
 - *_ (sublinhado)*
- *Não podem **começar** com*
 - *ponto,*
 - *hífen ou*
 - *número*

- *Ignora efeitos especiais dos caracteres*

```
<titulo>Curso de XML</titulo>
```

```
<exemplo>Considere o seguinte trecho de  
XML:
```

```
<![CDATA[
```

```
    <empresa>
```

```
        <nome>João & Maria S/A</nome>
```

```
    </empresa>
```

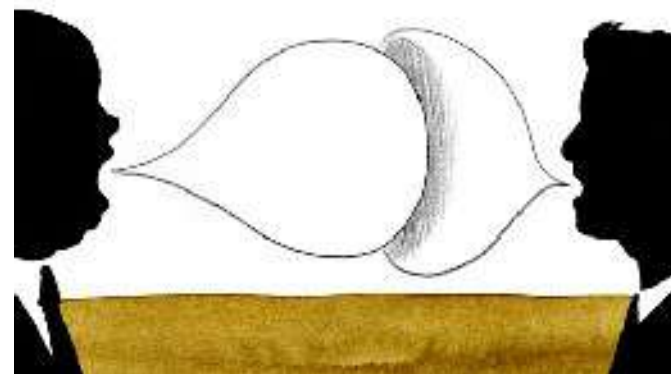
```
]]>
```

```
</exemplo>
```

- *Iguais aos comentários HTML*

```
<!-- Isto é um comentário -->
```

```
<!-- isto é um erro -- sério! -->
```



- *Opcional (exceto quando o conjunto de caracteres usado for diferente de UTF-8)*

```
<?xml version="1.0"  
      encoding="iso-8859-1"  
      standalone="yes" ?>
```


- *ter um único elemento raiz*
- *etiquetas iniciais e finais combinam (levando em conta que caracteres maiúsculos e minúsculos são diferentes)*
- *elementos bem aninhados*
- *valores de atributos entre aspas ou apóstrofes*
- *atributos não repetidos*
- *identificadores válidos para elementos e atributos*
- *comentários não devem aparecer dentro das etiquetas*
- *sinais < ou & nunca devem ocorrer dentro dos valores dos atributos ou nos nós de texto do documento.*



- *Permite que elementos de mesmo nome de diferentes aplicações sejam misturados sem que haja conflitos*
- *Um namespace (universo de nomes) é declarado usando atributos reservados*
 - *xmlns="identificador" (namespace default)*
 - *associa o identificador com todos os elementos que não possuem prefixo. Ex: <nome>*
 - *xmlns:prefixo="identificador"*
 - *associa o identificador com os elementos e atributos cujo nome local é precedido do prefixo. Ex <prefixo:nome>*
- *O prefixo é arbitrário e só existe dentro do documento*
- *O identificador (geralmente uma URI) deve ser reconhecido pela aplicação*

Exemplo

Vale para todo o elemento `<cartao>`

Esta URI está associada a este prefixo

```
<ct:cartao
  xmlns:ct="01.234.567/0001-89/cartoes">
  <ct:nome>Alberto Santos Dumont</ct:nome>
  <ct:endereco>Rua do Encanto, 22 - Centro
  25600-000 - Petrópolis - RJ</ct:endereco>
  <ct:email>dumont@14bis.com.br</ct:email>
  <ct:telefone tipo="residencial">
    <ct:ddd>21</ct:ddd>
    <ct:numero>2313011</ct:numero>
  </ct:telefone>
</ct:cartao>
```


Exemplo com 3 namespaces

```
<departamento
  xmlns:ct="01.234.567/0001-89/cartoes"
  xmlns="01.234.567/0001-89/empresa"
  xmlns:html="http://www.w3.org/1999/xhtml">

  <ct:nome>Fulano de Tal</ct:nome>
  <nome>Contabilidade</nome>
  <endereco>Rua Projetada, 33</endereco>
  <html:a href="web.html">
    <html:strong>link negrito HTML</html:strong>
  </html:a>
  <urgencia><ct:numero>2313011</ct:numero></urgencia>
</departamento>
```

Namespace default

URI padrão XHTML

- *Para se ter uma linguagem precisa-se de um **esquema***
 - *Linguagem implica comunicação.*
 - *Não há comunicação eficiente sem uma convenção entre as partes sobre vocabulários, regras de formação, etc.*



- *Documentos não válidos são "individualistas"*
 - *Um esquema representa um conjunto de documentos, que existem e que virão a existir*
 - *É possível fazer muitas coisas com UM documento não válido. É difícil automatizar os processos sem considerar uma CLASSE de documentos*
- *Um esquema é uma formalidade necessária*
 - *Se você tem uma grande coleção de documentos que foram construídos segundo determinadas regras, você já tem, informalmente, um esquema*
 - *Para validar documentos de acordo com suas convenções, é preciso ter um esquema*

- *Um esquema define uma classe de documentos*
 - *Os documentos que quiserem fazer parte dessa classe devem aderir ao esquema*
- *Um documento pode pertencer a várias classes*
 - *Um documento pode ser válido em vários esquemas*

- *Documentos válidos contém*
 - *Declaração de tipo de documento (para DTD), ou*
`<!DOCTYPE bilhete SYSTEM "bilhete.dtd">`
 - *Declaração de namespace e schema (para XML Schema)*
`<bilhete codigo="ZMIKT8"
xmlns="urn:123456789"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:123456789 bilhete.xsd">`
- *Para validar*
 - *Use um validador (com suporte a Schema)*



- *Um vocabulário*
 - *Elementos, atributos*
- *Uma gramática*
 - *Relacionamentos*
- *Uma coleção de entidades*
 - *No caso dos DTDs*

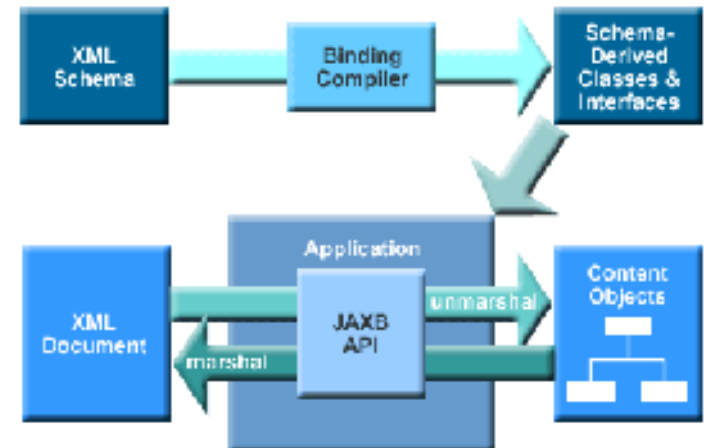


XML Schema é uma linguagem baseada no formato XML para definição de regras de validação ("esquemas") em documentos no formato XML.

Foi a primeira linguagem de esquema para XML a obter o status de recomendação por parte do W3C.

Foi amplamente utilizado para desenvolvimento da NF-e (Nota Fiscal Eletrônica) Brasileira.

- **XML Schema é uma alternativa ao DTD baseada em XML;**
- **Um esquema XML descreve a estrutura de um documento XML;**
- **A linguagem XML Schema também é chamada de XML Schema Definition (XSD).**



O propósito de um XML Schema é definir os blocos de construção permitidos em um documento XML, como um DTD.

Um XML Schema define:

- elementos que podem aparecer em um documento;
- atributos que podem aparecer em um documento;
- que elementos são elementos filhos;
- a ordem dos elementos filhos;
- o número de elementos filhos;
- se um elemento é vazio ou pode incluir texto;
- tipos de dados para elementos e atributos;
- valores padrão e fixos para elementos e atributos.

Um documento XML simples

```
<?xml version="1.0">  
<nota>  
<para>João</para>  
<de>José</de>  
<tipo>Lembrete</tipo>  
<corpo>Não esquecer a reunião amanhã!</corpo>  
</nota>
```



```
<?xml version="1.0"?>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.w3schools.com">

    <xs:element name="nota">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="para" type="xs:string"/>
          <xs:element name="de" type="xs:string"/>
          <xs:element name="tipo" type="xs:string"/>
          <xs:element name="corpo" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

  </xs:schema>
```

O elemento nota é dito um tipo complexo porque contém outros elementos.

Os outros elementos são ditos tipos simples porque não contém outros elementos.

Você vai aprender mais sobre tipos simples e complexo mais adiante.



- **Este documento contém uma referência a um XML Schema:**

```
<?xml version="1.0"?>
  <nota
    xmlns="http://www.w3schools.com"
    xsi:schemaLocation="http://www.w3schools.com nota.xsd">
    <para>João</para>
    <de>José</de>
    <tipo>Lembrete</tipo>
    <corpo>Não esquecer a reunião amanhã!</corpo>
  </nota>
```

O elemento <schema> é o elemento raiz de todo XML Schema:

```
<?xml version="1.0"?>
```

```
<xs:schema>
```

```
...
```

```
...
```

```
</xs:schema>
```

O elemento <schema> pode conter alguns atributos. Uma declaração de Schema geralmente parece com isto:

```
<?xml version="1.0"?>  
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="http://www.w3schools.com"  
xmlns="http://www.w3schools.com"  
elementFormDefault="qualified">  
  ...  
  ...  
  
</xs:schema>
```


- O seguinte fragmento:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

indica que os elementos e tipos de dados usados no esquema (schema, element, complexType, sequence, string, boolean, etc.) vêm do namespace "http://www.w3.org/2001/XMLSchema". Ele também especifica que os elementos e tipos de dados que vêm de "http://www.w3.org/2001/XMLSchema" devem ser prefixados com xs: !!!

- Este fragmento:

```
targetNamespace=http://www.w3schools.com
```

indica que os elementos definidos por este esquema (nota, para, de, tipo, corpo) vêm do namespace "http://www.w3schools.com" .

- Este fragmento:

```
xmlns="http://www.w3schools.com"
```

indica que o namespace padrão é "http://www.w3schools.com".

- E o fragmento:

```
elementFormDefault="qualified">
```

indica que todo elemento usado por uma instância de documento XML que foi declarado neste esquema deve ser qualificado pelo namespace.

```
<?xml version="1.0"?>
  <note xmlns="http://www.w3schools.com"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.w3schools.com nota.xsd">

    <para>João</para>
    <de>José</de>
    <tipo>Lembrete</tipo>
    <corpo>Não esquecer a reunião amanhã!</corpo>
  </note>
```

`xmlns="http://www.w3schools.com"` - Namespace Padrão

`xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"` -

Declara o prefixo o Namespace que trata de instâncias de esquemas

`xsi:schemaLocation="http://www.w3schools.com note.xsd">`- este atributo possui 2 valores :

o 1° é o namespace usado e o segundo é o esquema propriamente dito

XML Schemas define os elementos de um documento XML.

Um elemento simples é um elemento XML que contém apenas texto. Ele não pode conter outros elementos ou atributos.

O texto pode ser de diferentes tipos:

- Boolean
- String
- Date
- Etc.

Ou um tipo definido por você

```
<xs:element name="xxx" type="yyy"/>
```

Exemplos:

```
<xs:element name="para" type="xs:string"/>
```

```
<xs:element name="de" type="xs:string"/>
```

```
<xs:element name="tipo" type="xs:string"/>
```

```
<xs:element name="corpo" type="xs:string"/>
```

XML Schema tem vários tipos de dados próprios. Aqui está uma lista dos mais comuns:

- `xs:string`
- `xs:decimal`
- `xs:integer`
- `xs:boolean`
- `xs:date`
- `xs:time`

- **Exemplos:**

Padrão

```
<xs:element name="color" type="xs:string" default="red"/>
```

Fixo

```
<xs:element name="color" type="xs:string" fixed="red"/>
```

- Todos atributos são declarados como tipos simples.
- Apenas elementos complexos tem atributos!

Importante!!!!

Elementos simples não podem ter atributos. Se um elemento tem atributos, ele é considerado do tipo complexo. Mas atributos são declarados como tipos simples. Isso significa que um elemento com atributos sempre tem uma definição do tipo complexo.

A sintaxe é a seguinte

```
<xs:attribute name="xxx" type="yyy"/>
```

Exemplo:

```
<lastname lang="EN">Smith</lastname>
```

Definição:

```
<xs:attribute name="lang" type="xs:string"/>
```

Assim como elementos, atributos também aceitam valores padrão e fixo

- Todos atributos são opcionais por padrão. Para especificar explicitamente que um atributo é opcional, utilize o atributo "use":

```
<xs:attribute          name="lang"          type="xs:string"  
  use="optional"/>
```

Para fazer um atributo obrigatório:

```
<xs:attribute          name="lang"          type="xs:string"  
  use="required"/>
```

- Falamos até aqui do básico da linguagem XML, para atender a nossa necessidade;
- Em alguns casos, um conhecimento mais aprofundado desse assunto será necessário;

- <http://www.w3schools.com/schema/default.asp>





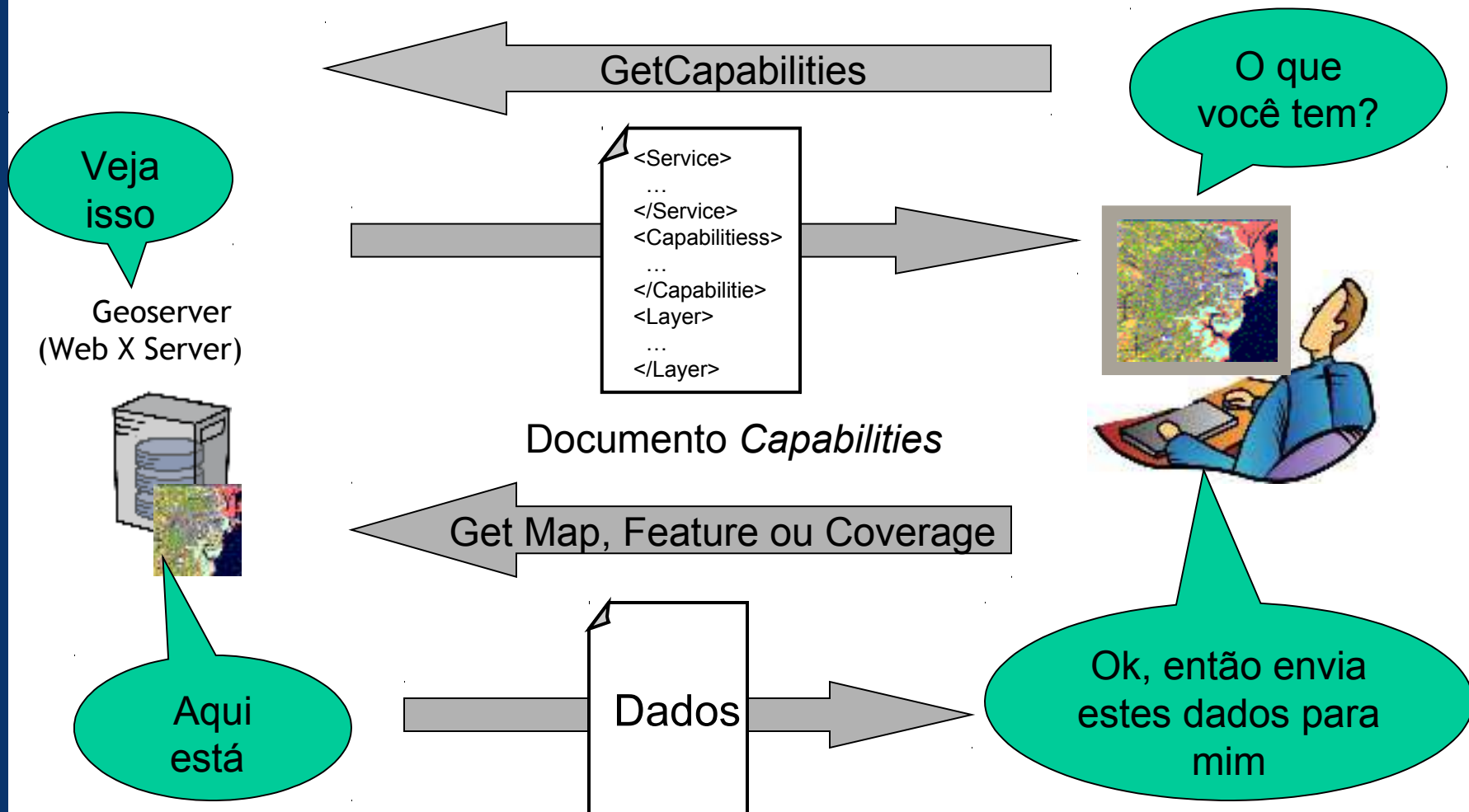
<http://geoserver.org>

Geoserver é uma ferramenta *open source* que permite compartilhar e editar informação geoespacial mediante o uso de padrões, segundo as especificações da OGC.

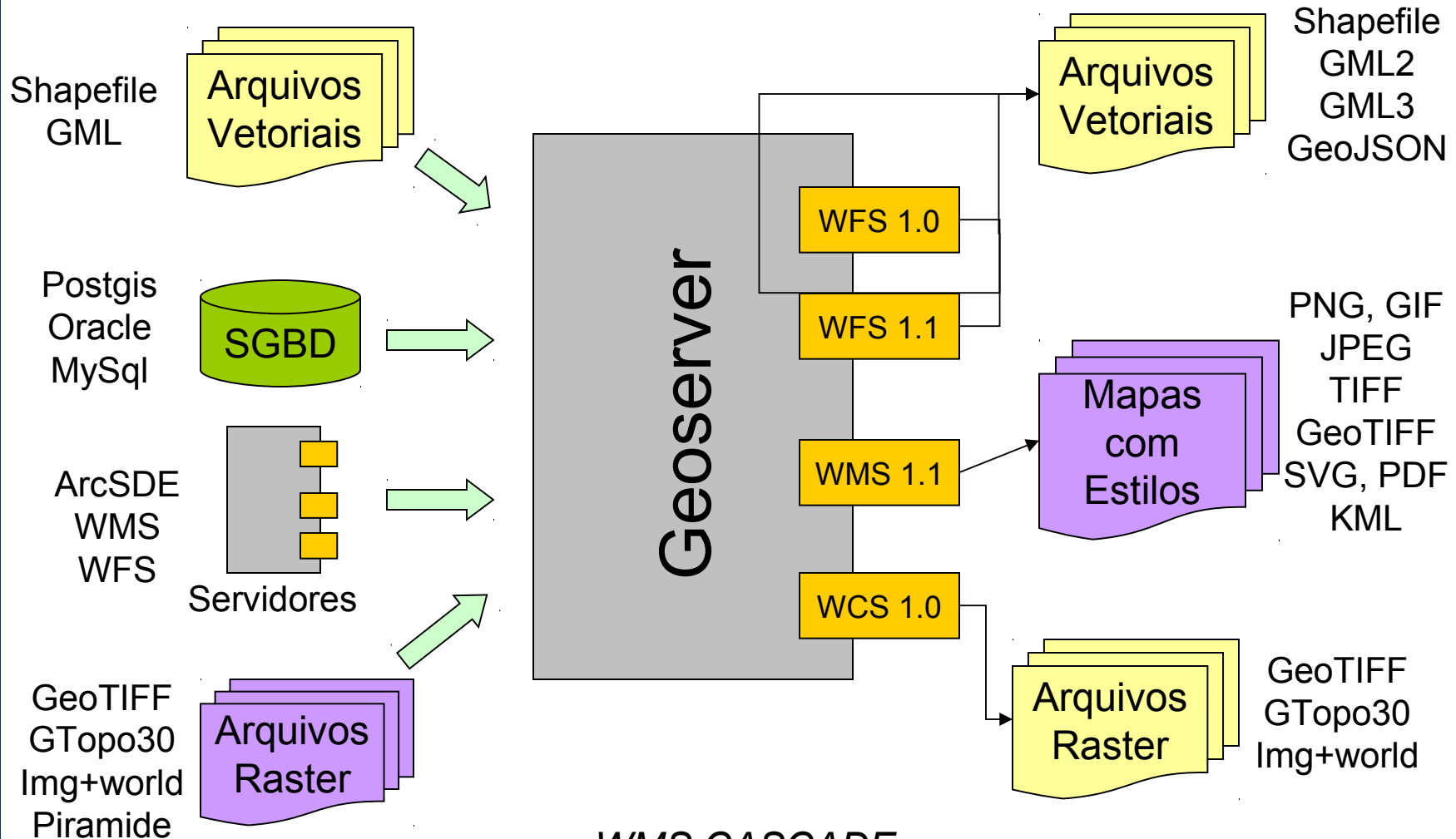
Geoserver está escrito em Java - multiplataforma (execução em múltiplos sistemas operacionais).

- O **Geoserver** Implementa as seguintes especificações OGC:
 - WMS 1.1.1 (Certificação de Alta Performance)
 - WFS 1.0 (Implementação de referência)
 - WCS 1.0 (Implementação de referência)

Visão Geral : Padrão OGC Web Services (“W*S”)



Dados Suportados



WMS CASCADE

A nova versão do Geoserver permite adicionar serviços WMS de outros servidores

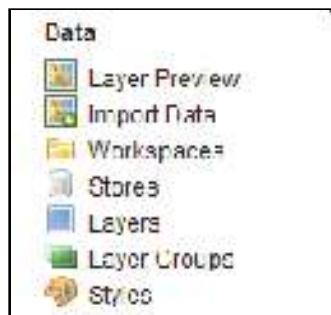


Workspace é um recurso lógico utilizado para agrupar os dados que possui alguma similaridade.

Store é onde são informados os parâmetros relativos aos dados geoespaciais que serão disponibilizados como serviços



Layer é uma coleção de feições geográficas, normalmente representadas por uma primitiva geométrica (ponto, linha ou polígono), ou uma imagem georreferenciada.



Um estilo (**Style**) tem parâmetros de apresentação de feições geográficas como cor, forma, tamanho e regras de apresentação que permitem restringir a apresentação das feições geográficas em função do Zoom, etc.

Todos os **Layers** devem ter, pelo menos, um estilo associado.

Na Página de *download* do Geoserver existem várias opções de arquivo de instalação:

Utilizaremos o arquivo war (*Web Archive*) para fazer a instalação



The screenshot shows the 'Stable' section of the Geoserver 2.1.2 download page. It includes the version number 'GeoServer 2.1.2', the release date 'Released on October 6, 2011', and two columns: 'Download format' and 'Documentation'. The 'Download format' column lists: Binary (OS independent), Web Archive, Windows Installer, Mac OS X Installer, and Source Code. The 'Documentation' column lists: HTML Documentation, PDF Documentation, API Documentation, and Change Log.

Download format	Documentation
 Binary (OS independent)	 HTML Documentation
 Web Archive	 PDF Documentation
 Windows Installer	 API Documentation
 Mac OS X Installer	 Change Log
 Source Code	

Geoserver - Instalação

Manager

[List Applications](#) [LTM Manager Jobs](#) [Manager Help](#)

Applications

Path	Display Name	Running	Resources	Commands
/	Webstore 1.0 Context	true	0	Start Stop Reload Undeploy Deploy operations with idle 300 minutes
/store	Tomcat Documentation	true	0	Start Stop Reload Undeploy Deploy operations with idle 300 minutes
/contextos	Knowledge 1.0 Context	true	0	Start Stop Reload Undeploy Deploy operations with idle 300 minutes
/store/contextos	WebContext	true	0	Start Stop Reload Undeploy Deploy operations with idle 300 minutes
/geoserver	Geo Server	true	0	Start Stop Reload Undeploy Deploy operations with idle 300 minutes
/geoserver/manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy Deploy operations with idle 300 minutes
/manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy Deploy operations with idle 300 minutes

Deploy

Define a directory or WAR file location on server

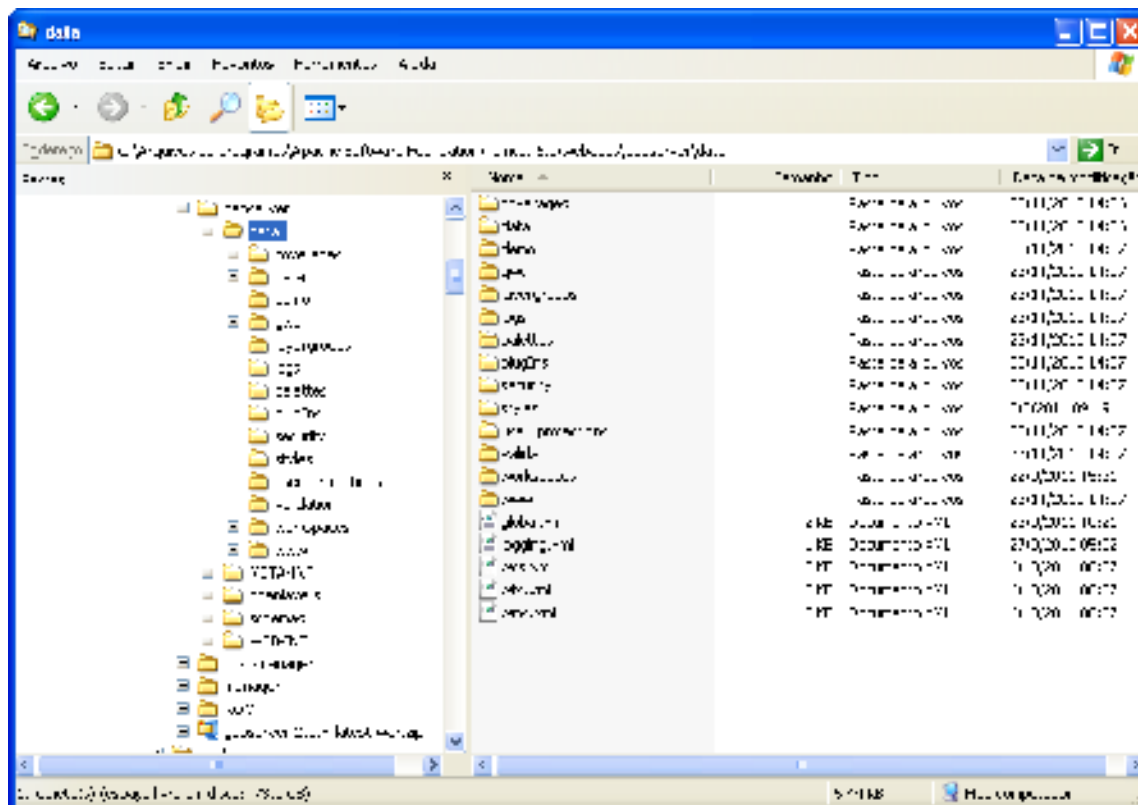
Context Path (required): _____
XML Configuration File URL: _____
WAR or Directory URI: _____

WAR file to deploy

Selected WAR file to upload: _____

Geoserver Data Directory:

O Geoserver *Data Directory* é o caminho onde são armazenados as configurações (como os arquivos SLD) e os dados (*shape* e imagens, por exemplo) do servidor.



Geoserver Data Directory:

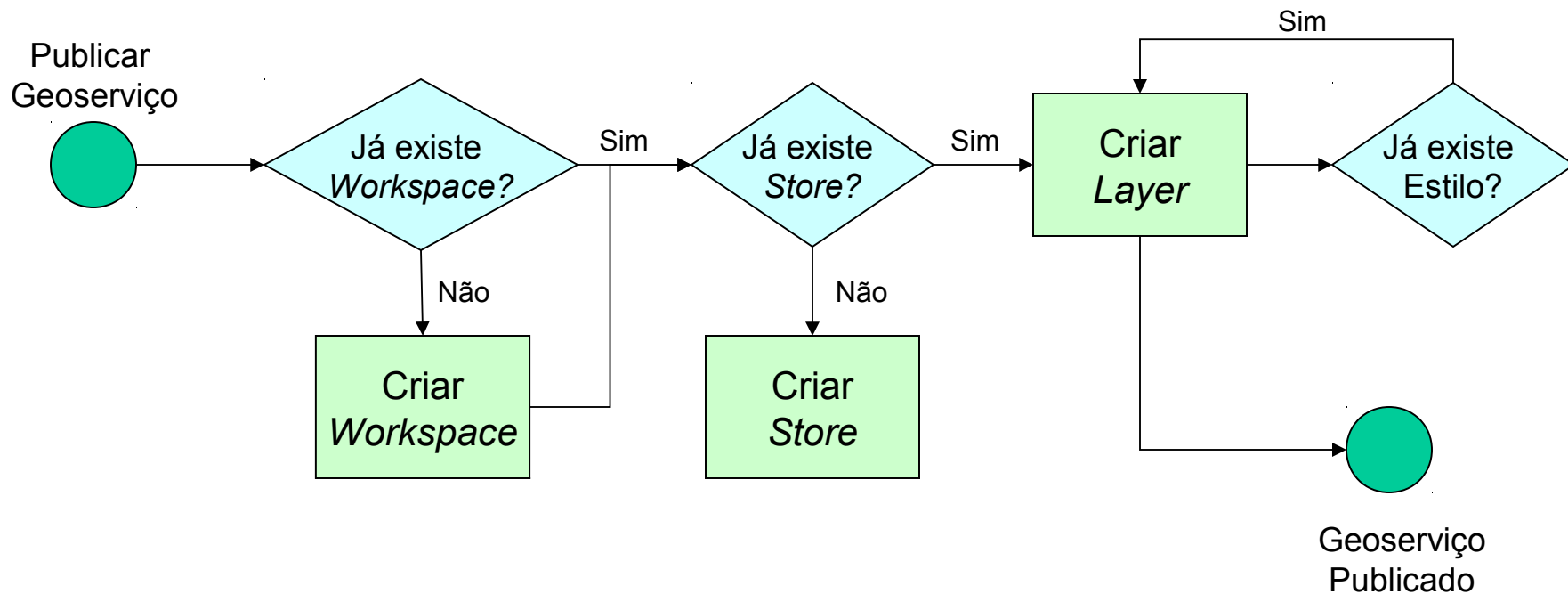
É recomendável a criação de um diretório em local diferente do caminho onde será instalado o Geoserver, para facilitar eventuais atualizações da aplicação.

Para alterar a configuração original do diretório de dados deve-se editar o arquivo web.xml, encontrado no diretório WEB-INF do caminho de instalação do Geoserver.

```
<context-param>  
  <param-name>GEOSERVER_DATA_DIR</param-name>  
  <param-value>C:\GeoserverData</param-value>  
</context-param>
```


Configuração de Geoserviços

A criação de Geoserviços no Geoserver segue um seqüência lógica:



Geoserver - Interface de Administração

Service Capabilities

Service	Version
WFS	2.0.0
WFS	2.1.1
WFS	2.0.0
WFS	2.1.0
WMS	2.1.1
WMS	2.1.0

Acesso a administração dos principais serviços de configuração

Links para o Documento Capabilities De cada tipo de serviço

- Para que o usuário possa definir suas próprias regras de estilo foi definida uma linguagem padrão: **Symbology Encoding (SE)**
- SE se usa para simbolizar as respostas de WMS, WFS, WCS.
- O **perfil WMS-SLD (Styled Layer Descriptor)** descreve como se o SE pode ser usado junto com WMS.
- WMS-SLD se trata de uma extensão que permite a simbolização pelo usuário.
- OGC publicou a versão SLD 1.0.0 em 2002 e a última versão SLD 1.1.0 em 2007:

<http://www.opengeospatial.org/standards/sld>

- Perfil WMS que permite configurar como os dados serão apresentados;
- Permite alterar o estilo padrão do Layer com solicitações HTTP-GET e POST;
- Até a versão 1.0 a especificação também tratava da linguagem de apresentação dos dados, após isso a OGC dividiu a especificação, deixando a parte da linguagem com a especificação SE



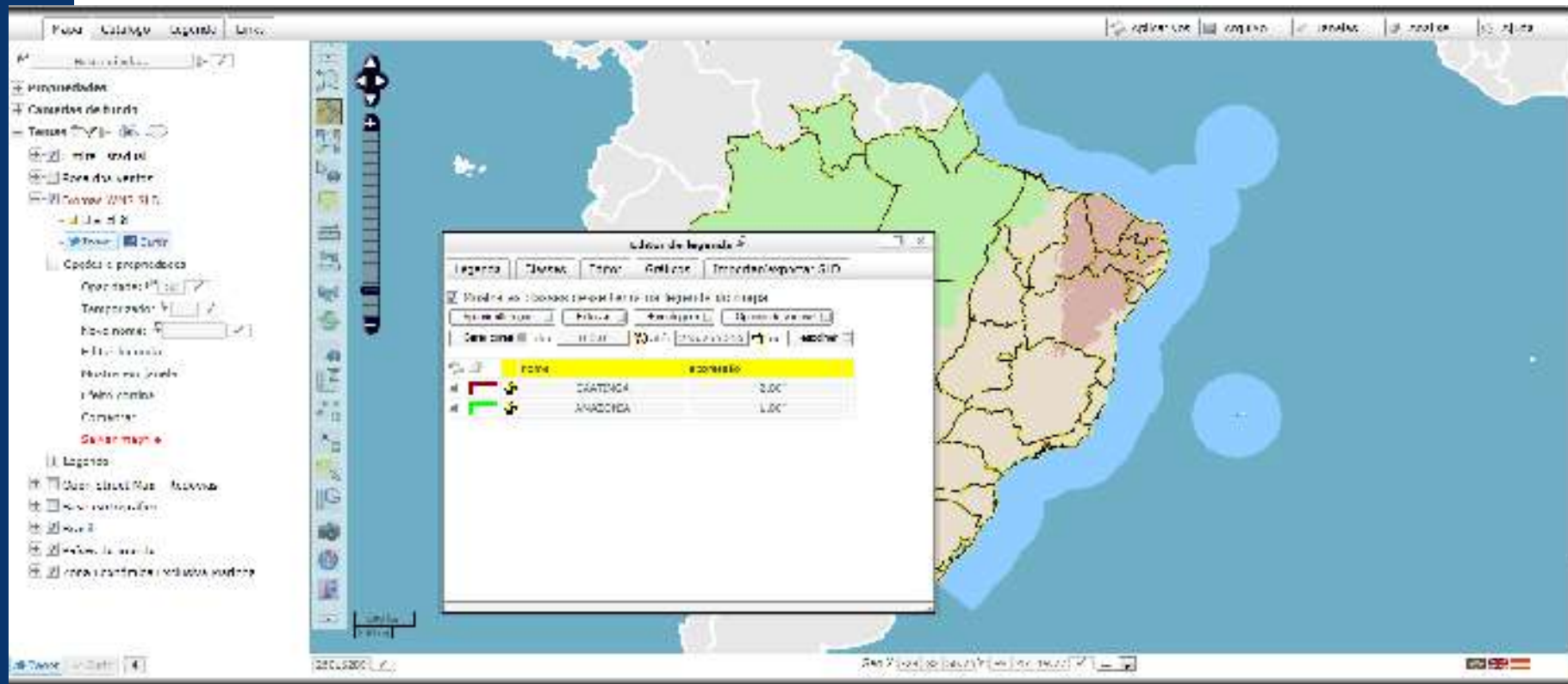
Exemplo de solicitação HTTP-GET a um servidor WMS

```
http://servidorWms.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
```

Neste exemplo estamos
Atribuindo estilos a 3
Layers diferentes.

SE - Symbology Encoding

Linguagem XML orientada a informações de estilo que pode ser aplicado em feições cartográficas vetoriais e raster;



```
<xsd:element name="FeatureTypeStyle" type="se:FeatureTypeStyleType">
</xsd:element>
<xsd:complexType name="FeatureTypeStyleType">
  <xsd:sequence>
    <xsd:element ref="se:Name" minOccurs="0"/>
    <xsd:element ref="se:Description" minOccurs="0"/>
    <xsd:element ref="se:FeatureTypeName" minOccurs="0"/>
    <xsd:element ref="se:SemanticTypeIdentifier" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="se:Rule"/>
      <xsd:element ref="se:OnlineResource"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="version" type="se:VersionType"/>
</xsd:complexType>
```



```
<xsd:element name="Rule" type="se:RuleType">
</xsd:element>
<xsd:complexType name="RuleType">
  <xsd:sequence>
    <xsd:element ref="se:Name" minOccurs="0"/>
    <xsd:element ref="se:Description" minOccurs="0"/>
    <xsd:element ref="se:LegendGraphic" minOccurs="0"/>
    <xsd:choice minOccurs="0">
      <xsd:element ref="ogc:Filter"/>
      <xsd:element ref="se:ElseFilter"/>
    </xsd:choice>
    <xsd:element ref="se:MinScaleDenominator" minOccurs="0"/>
    <xsd:element ref="se:MaxScaleDenominator" minOccurs="0"/>
    <xsd:element ref="se:Symbolizer" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Estrutura de um arquivo SLD

Um arquivo SLD contém a seguinte estrutura hierárquica:

- Header(cabeçalho)
 - FeatureTypeStyles
 - Rules
 - Symbolizers

O header contém informações de metadados XML como Namespaces, Schema, etc;

O elemento FeatureTypeStyles é um grupo de regras de Estilo;

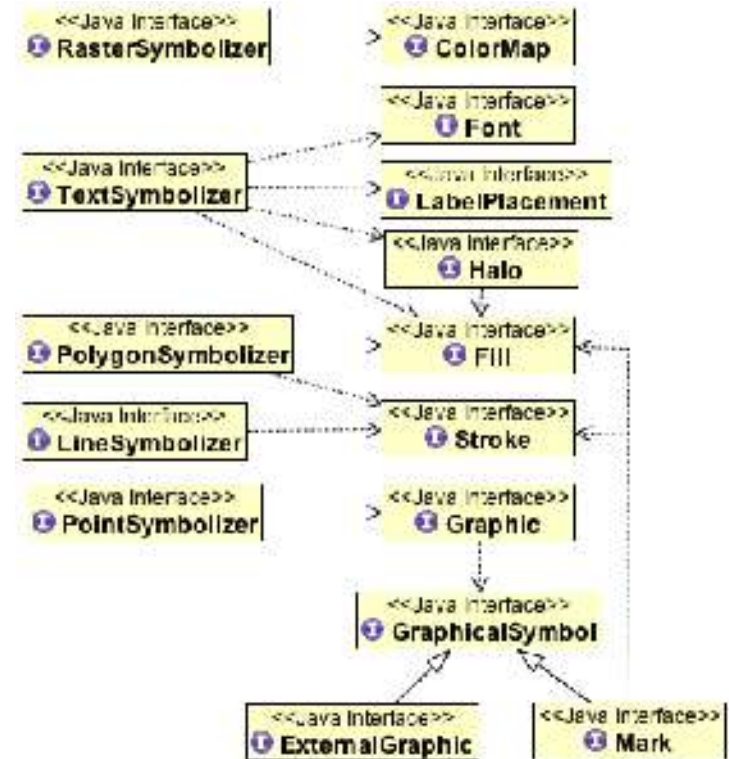
Uma Rule é uma diretiva de estilo que pode ser aplicada a Todo conjunto de dados ou pode conter filtros;

Um Symbolizer é uma instrução de estilo;

Symbolizers

Existem 5 Symbolizers:

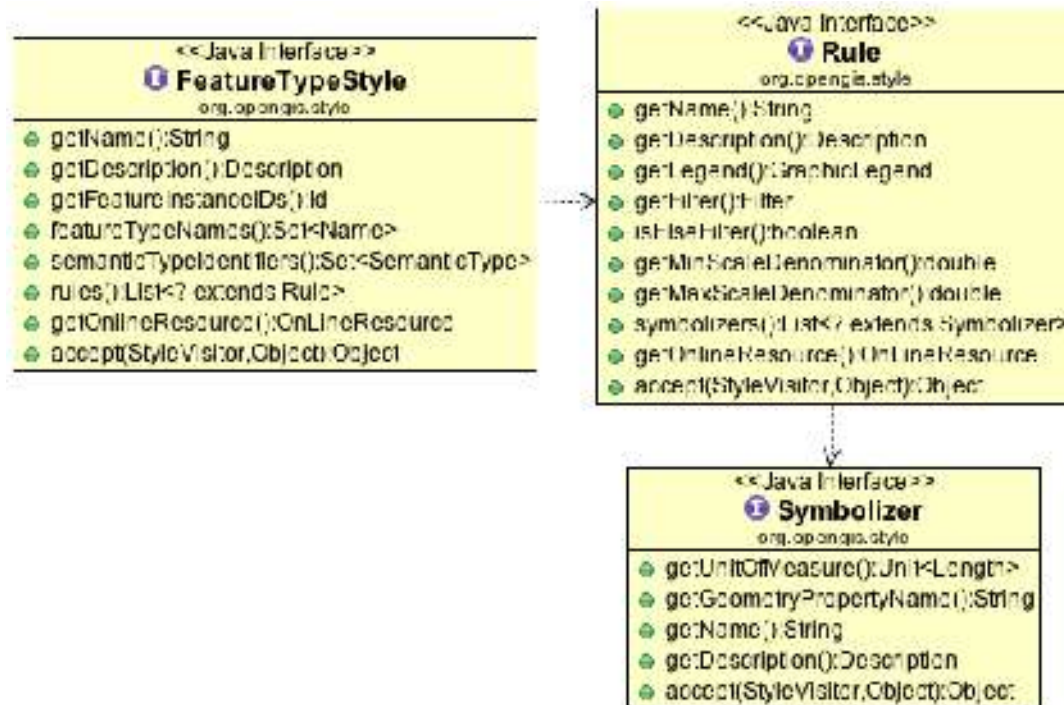
- *PointSymbolizer*
- *LineSymbolizer*
- *PolygonSymbolizer*
- *RasterSymbolizer*
- *TextSymbolizer*



Estrutura de um arquivo SLD

Pode existir um ou mais por arquivo;
Uma ou mais *Rule* por *FeatureTypeStyles*;
Um ou mais *Symbolizer* por *Rule*;

IMPORTANTE: No caso de haver mais de um *FeatureTypeStyles* em um arquivo SLD, a ordem desses elementos irá afetar a ordem de desenho das feições, assim as feições correspondentes ao primeiro *FeatureTypeStyles* serão desenhadas primeiro em seguida serão desenhadas as feições do segundo *FeatureTypeStyles* e assim sucessivamente



Sintaxe:

Tag	Obrigatório?	Descrição
<ExternalGraphic>	Não	Especifica uma imagem para ser usada como símbolo
<Mark>	Não	Especifica uma forma para ser usada como símbolo
<Opacity>	Não	Determina a transparência dos símbolos. Valores entre 0 e 1 (completamente opaco). Padrão 1
<Size>	Sim	Determina o tamanho do símbolo em pixel
<Rotation>	Não	Determina a rotação do gráfico em graus. Segue o sentido horário de rotação

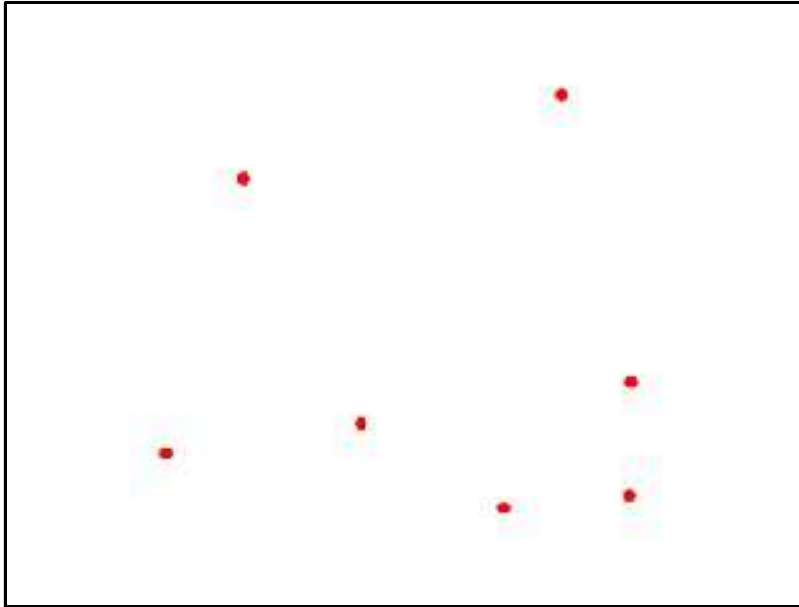
Elementos contidos no *ExternalGraphic* :

Tag	Obrigatório?	Descrição
<OnlineResource>	Sim	Endereço do arquivo de imagem
<Format>	Sim	Especifica uma forma para ser usada como símbolo. Pode ser uma URL ou caminho relativo ao arquivo

Elementos contidos no *ExternalGraphic* :

Tag	Obrigatório?	Descrição
<WellKnownName>	Sim	O nome da forma. Pode ser circle, square, triangle, star, cross, ou x. Padrão é square
<Fill>	Não	Especifica como o símbolo será preenchido. As opções são <CssParameter name="fill"> com uma cor no formato #RRGGBB ou <GraphicFill> para utilizar um gráfico como preenchimento
<Stroke>	Não	Determina como o contorno será desenhado. As opções são <CssParameter name="fill"> para especificar uma cor ou <GraphicStroke> para utilizar um gráfico

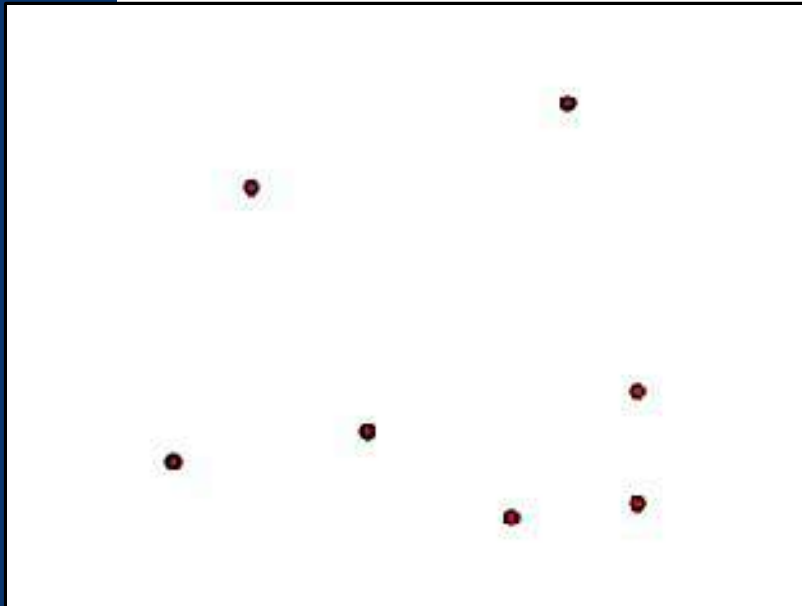
Simbolizando Pontos – Exemplo 1



```
<FeatureTypeStyle>
  <Rule>
    <PointSymbolizer>
      <Graphic>
        <Mark>
          <WellKnownName>circle</WellKnownName>
          <Fill>
            <CssParameter name="fill">#FF0000</CssParameter>
          </Fill>
        </Mark>
        <Size>6</Size>
      </Graphic>
    </PointSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Aqui temos uma *Rule* em um *FeatureTypeStyle*, situação mais simples. O estilo do ponto foi configurado no *PointSymbolizer*, onde o elemento *Mark* determinou a forma como um círculo, o elemento *Fill* o preenchimento vermelho e size o tamanho de 6 pixels

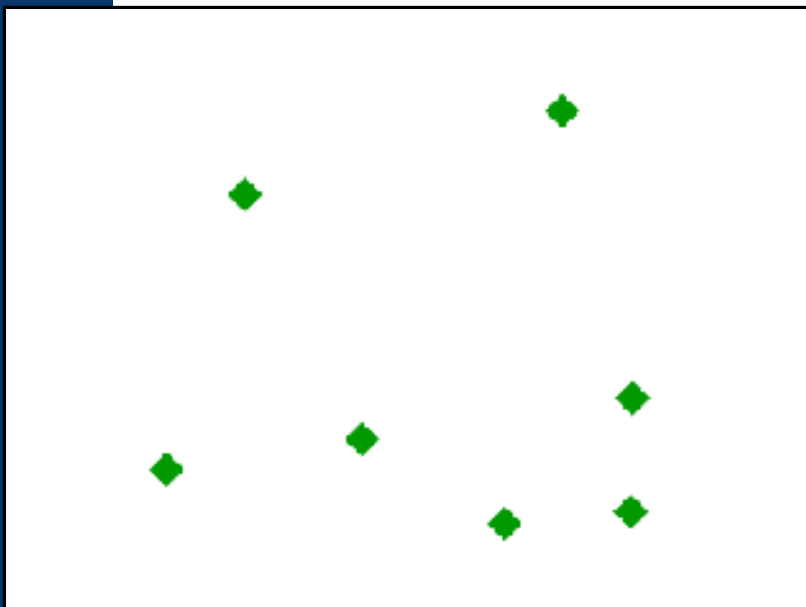
Simbolizando Pontos - Exemplo 2



```
<FeatureTypeStyle>
  <Rule>
    <PointSymbolizer>
      <Graphic>
        <Mark>
          <WellKnownName>circle</WellKnownName>
          <Fill>
            <CssParameter name="fill">#FF0000</CssParameter>
          </Fill>
          <Stroke>
            <CssParameter name="stroke">#000000</CssParameter>
            <CssParameter name="stroke-width">2</CssParameter>
          </Stroke>
        </Mark>
        <Size>6</Size>
      </Graphic>
    </PointSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

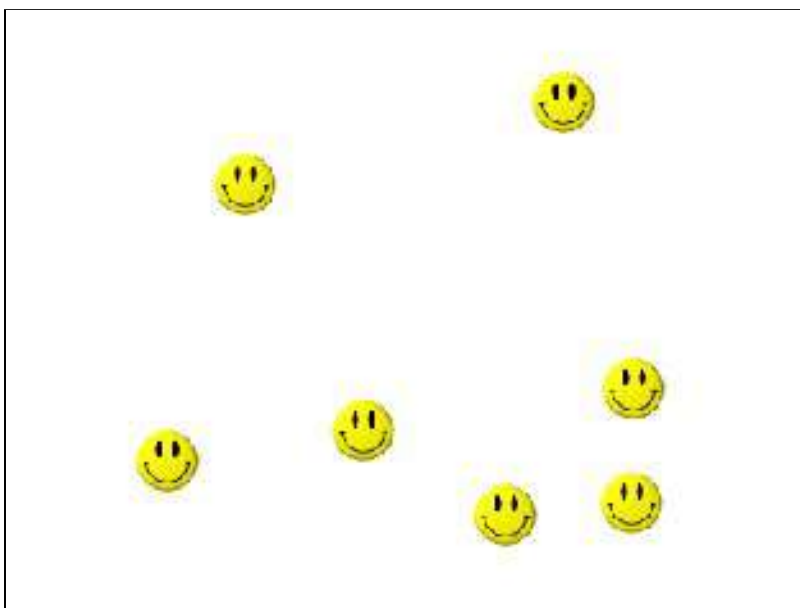
Neste exemplo foi adicionado um contorno preto ao ponto com o elemento *Stroke*. O contorno é preto (#000000) com espessura de 2 *pixels* (*stroke-width*)

Simbolizando Pontos- Exemplo 3



```
<FeatureTypeStyle>
  <Rule>
    <PointSymbolizer>
      <Graphic>
        <Mark>
          <WellKnownName>square</WellKnownName>
          <Fill>
            <CssParameter name="fill">#009900</CssParameter>
          </Fill>
        </Mark>
        <Size>12</Size>
        <Rotation>45</Rotation>
      </Graphic>
    </PointSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Agora os pontos estão simbolizados com quadrados (square) rotacionados (45°)



```
<FeatureTypeStyle>
  <Rule>
    <PointSymbolizer>
      <Graphic>
        <ExternalGraphic>
          <OnlineResource
            xlink:type="simple"
            xlink:href="smileyface.png" />
          <Format>image/png</Format>
        </ExternalGraphic>
        <Size>32</Size>
      </Graphic>
    </PointSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Neste exemplo os pontos estão sendo simbolizados com uma figura. O caminho da figura deve ser indicada no elemento *ExternalGraphic*. Neste caso o arquivo está no mesmo caminho do arquivo SLD

Criando o SLD

1 – A partir de um arquivo existente:

The screenshot displays the GeoServer Styles management interface. On the left, the 'Styles' panel shows a list of existing styles: 'bug', 'capitals', 'cite_lakes', 'dem', 'giant_polygon', 'grass', and 'green'. A red box highlights the 'Add a new style' button. On the right, the 'New style' form is visible. The 'Name' field contains 'cursoGeoservicos'. Below it, the 'Copy from existing style' dropdown menu is open, showing a list of style names. The 'point' style is currently selected in the dropdown. The 'Validate' button is also visible at the bottom of the form.

Styles
Manage the Styles published by GeoServer

[Add a new style](#)
Removed selected style(s)

<< < 7 > >> Results 1 to 21 (out of 21 items)

<input type="checkbox"/>	Style Name
<input type="checkbox"/>	bug
<input type="checkbox"/>	capitals
<input type="checkbox"/>	cite_lakes
<input type="checkbox"/>	dem
<input type="checkbox"/>	giant_polygon
<input type="checkbox"/>	grass
<input type="checkbox"/>	green

New style
Type a new SLD definition, or use an existing one as a template, or upload a real file. Use the "validate" button to verify the style is a valid SLD document.

Name:

Copy from existing style: [Copy ...](#)

- Escolha
- bug
- capitals
- cite_lakes
- dem
- giant_polygon
- grass
- green
- line
- poi
- point**
- poly_landmarks
- polygon
- pophatch
- population
- rain
- raster
- restricted
- simple_made

Criando o SLD

New style

Type a new SLD definition, or use an existing one as a template, or upload a ready-made style from your file system. The editor can provide syntax highlight and be brought the "validate" button to verify the style is a valid SLD document.

Name

Copy from existing style

currit

12pt

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc"
3   xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:se="http://www.opengis.net/se" http://schemas.opengis.net/sld/1.0.0/StyleLayerDescriptor.xsd">
5   <NamedLayer>
6     <Name>Default Point</Name>
7     <IsStyle>
8       <Title>Default point</Title>
9       <Abstract>A sample style that just prints out a 6px wide red square</Abstract>
10
11     <FeatureTypeStyle>
12       <! -- FeatureTypeName --><FeatureTypeName -->
13       <Title>
14         <Title>Red square</Title>
15         <FillSymbolizer>
16           <Graphic>
17             <Mark>
18               <Uc_KnownMark>square</WellKnownName>
19             </>
20             <CssParameter name="fill">#ff0000</CssParameter>
```

A partir daqui podemos fazer alterações conforme nossa necessidade

- 1 – Criar um novo arquivo SLD relativo a pontos e associar a um Layer com uma requisição *GetMap*;



Simbolizando Pontos – Ex. 5 Adicionando Labels



```
<FeatureTypeStyle>  
  <Rule>  
    <PointSymbolizer>  
      (...)  
    </PointSymbolizer>  
    <TextSymbolizer>  
      <Label>  
        <ogc:PropertyName>name</ogc:PropertyName>  
      </Label>  
      <Fill>  
        <CssParameter name="fill">#000000</CssParameter>  
      </Fill>  
    </TextSymbolizer>  
  </Rule>  
</FeatureTypeStyle>
```

Podemos ainda adicionar Labels (Rótulos) aos pontos com o elemento *TextSymbolizer*. O Elemento *Label* indica o atributo que contém os valores que serão utilizados como rótulo. O elemento *Fill* indica o cor da fonte.

Sintaxe do *TextSimbolizer*

Tag	Obrigatório?	Descrição
<Label>	Sim	Especifica o conteúdo do rótulo
	Não	Especifica informações sobre a fonte do rótulo
<LabelPlacement>	Não	Determina o posicionamento do Rótulo
<Halo>	Sim	Cria uma cor de fundo em torno do rótulo
<Fill>	Não	Determina a cor de preenchimento da fonte

Cada *tag* possui *sub-tags*.

Tag

Dentro de *Font* pode haver *tags* <CssParameter>, que pode ter 4 tipos de parâmetros.

Tag	Obrigatório?	Descrição
name="font-family"	Não	Determina o nome da família da fonte. Padrão: Times.
name="font-style"	Não	Especifica o estilo da fonte. Opções: normal, italic, e oblique. Padrão: normal.
name="font-weight"	Não	Determina o peso da fonte. Opções: normal e bold. Padrão: normal
name="font-size"	Não	Determina o tamanho da fonte em pixel. Padrão: 10.

```
Ex.: <Font>  
  <CssParameter name="font-family">Arial</CssParameter>  
  <CssParameter name="font-size">12</CssParameter>  
  <CssParameter name="font-style">normal</CssParameter>  
  <CssParameter name="font-weight">bold</CssParameter>  
</Font>
```

Tag <Halo>

Dentro de *Halo* pode haver duas *Tags* :

Tag	Obrigatório?	Descrição
<Radius>	Não	Determina o raio da cor de fundo em pixel. O Padrão é 1.
<Fill>	Não	Determina a cor do preenchimento na forma #RRGGBB. O Padrão é Branco #FFFFFF

Tag <Placement>

Determina a localização do texto:

Tag	Obrigatório?	Descrição
<PointPlacement>	Não	Utilizado com geometria do tipo Ponto.
<LinePlacement>	Não	Utilizado com geometria do tipo Linha

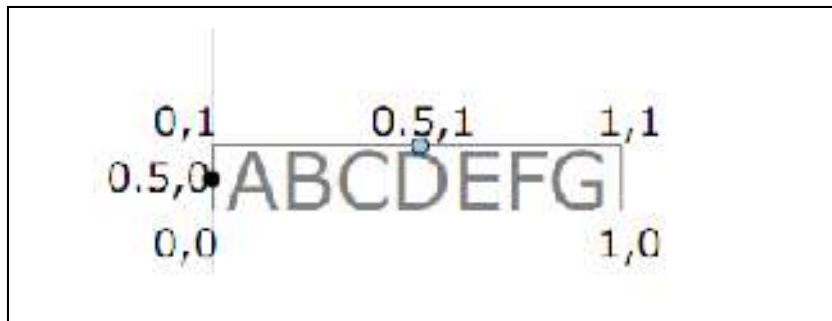
Tag <PointPlacement>

Tag	Descrição
<AnchorPoint>	Determina a posição relativa do rótulo em relação ao ponto. Ex.: Centralizado no ponto, acima do ponto e do lado esquerdo...
<Displacement>	Distância em pixel do rótulo em relação ao ponto.
<Rotation>	Determina o valor da rotação, no sentido horário, do <i>Label</i> . Valor em graus.

AnchorPoint

O valor padrão é $x=0,y=0.5$

Rótulo no lado direito do ponto



<PointPlacement>

```
<AnchorPoint>
```

```
<AnchorPointX>
```

```
0.5
```

```
</AnchorPointX>
```

```
<AnchorPointY>
```

```
0.5
```

```
</AnchorPointY>
```

```
</AnchorPoint>
```

```
</PointPlacement>
```

Tag <Displacement>

Os valores são em pixel



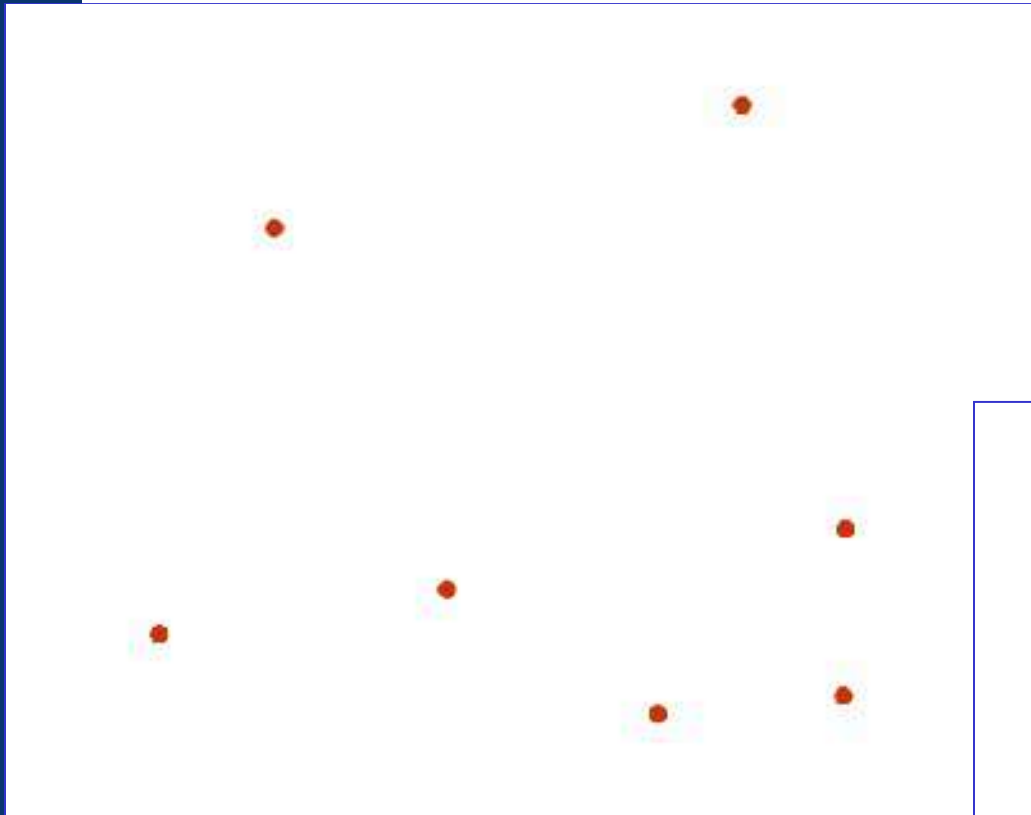
```
<PointPlacement>  
<Displacement>  
  <DisplacementX>  
    10  
  </DisplacementX>  
  <DisplacementY>  
    0  
  </DisplacementY>  
</Displacement>  
</PointPlacement>
```


Tag < LabelPlacement >

Ex.:

```
<LabelPlacement>
  <PointPlacement>
    <AnchorPoint>
      <AnchorPointX>0.5</AnchorPointX>
      <AnchorPointY>0.0</AnchorPointY>
    </AnchorPoint>
    <Displacement>
      <DisplacementX>0</DisplacementX>
      <DisplacementY>5</DisplacementY>
    </Displacement>
  </PointPlacement>
</LabelPlacement>
```

Pontos em Função da Escala



Pontos em Função da Escala

```
<FeatureTypeStyle>
  <Rule>
    Rule - 1
  </Rule>
  <Rule>
    Rule - 2
  </Rule>
  <Rule>
    Rule - 3
  </Rule>
</FeatureTypeStyle>
```

```

  <Rule>
    <Name>Grande</Name>
    <MaxScaleDenominator>160000000</MaxScaleDenominator>
    <PointSymbolizer>
      <Graphic>
        <Mark>
          <WellKnownName>circle</WellKnownName>
          <Fill>
            <CssParameter name="fill">#CC3300</CssParameter>
          </Fill>
        </Mark>
        <Size>12</Size>
      </Graphic>
    </PointSymbolizer>
  </Rule>
```

Rule 1

Pontos em Função da Escala

Rule 2

```
<Rule>
  <Name>Medio</Name>
  <MinScaleDenominator>160000000</MinScaleDenominator>
  <MaxScaleDenominator>320000000</MaxScaleDenominator>
  <PointSymbolizer>
    <Graphic>
      <Mark>
        <WellKnownName>circle</WellKnownName>
        <Fill>
          <CssParameter name="fill">#CC3300</CssParameter>
        </Fill>
      </Mark>
      <Size>8</Size>
    </Graphic>
  </PointSymbolizer>
</Rule>
```

Pontos em Função da Escala

Rule 3

```
<Rule>
  <Name>Pequeno</Name>
  <MinScaleDenominator>320000000</MinScaleDenominator>
  <PointSymbolizer>
    <Graphic>
      <Mark>
        <WellKnownName>circle</WellKnownName>
        <Fill>
          <CssParameter name="fill">#CC3300</CssParameter>
        </Fill>
      </Mark>
      <Size>4</Size>
    </Graphic>
  </PointSymbolizer>
</Rule>
```

Descrição das *Rules*

Rules	Nome	Denominador de escala
1	Grande	1:160.000.000 ou menos
2	Médio	1:160.000.000 a 1:320.000.000
3	Pequeno	Maior que 1:320.000.000

Simbolizando Pontos – Exemplo 7

Symbolizer Dinâmico

```
<FeatureTypeStyle>
  <Rule>
    <PointSymbolizer>
      <Graphic>
        <ExternalGraphic>
          <OnlineResource xlink:type="simple"
xlink:href="http://meusite.com/tn_${ESTADO_DO_BRASIL}.png"/>
          <Format>image/png</Format>
        </ExternalGraphic>
        <Size>32</Size>
      </Graphic>
    </PointSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

O Geoserver permite o uso de expressões CQL nos elemento **WellKnownName** e **OnlineResource/@xlink:href**. Neste exemplo estamos passando o nome de um Atributo, para ser substituído pelo nome do arquivo que será usado pelo Symbolizer

Sintaxe:

Tag	Obrigatório?	Descrição
<Stroke>	Sim	Determina a visualização da linha e pode conter outras 3 tags

Tags que podem ser incluídas em *Stroke*:

Tag	Obrigatório?	Descrição
<GraphicFill>	Não	Desenha a linha com um padrão repetido
<GraphicStroke>	Não	Desenha a linha com um gráfico repetido
<CssParameter>	Não	Determina os parâmetros de estilo da linha

Tag < CssParameter >

Existem diversos parâmetros que podem ser acrescentados a *tags* <CssParameter>

Tag	Obrigatório?	Descrição
name="stroke"	Não	Determina a cor da linha na forma #RRGGBB. Padrão: Preto #000000.
name="stroke-width"	Não	Especifica a largura da linha em pixel. Padrão: 1
name="stroke-opacity"	Não	Determina a transparência da linha. Valores entre 0 e 1. Padrão 1 (Opaco)
name="stroke-linejoin"	Não	Determina como as linhas são desenhadas nas interseções. Valores: <i>mitre</i> , <i>round</i> , e <i>bevel</i> . Padrão <i>mitre</i>
name="stroke-linecap"	Não	Determina como as linhas são desenhadas no final do segmento. Valores: <i>butt</i> , <i>round</i> e <i>square</i> . Padrão <i>butt</i>
name="stroke-dasharray"	Não	Configura a apresentação da linha com padrão de espaços. Usa um par de valores: o primeiro o comprimento da linha desenhada e o segundo o comprimento do espaçamento.
name="stroke-dashoffset"	Não	Determina o espaçamento em pixel onde o padrão com espaços era começar. Em pixel

Stroke-linejoin e stroke-linecap

stroke-linejoin



'miter' join



'round' join



'bevel' join

stroke-linecap



'butt' cap

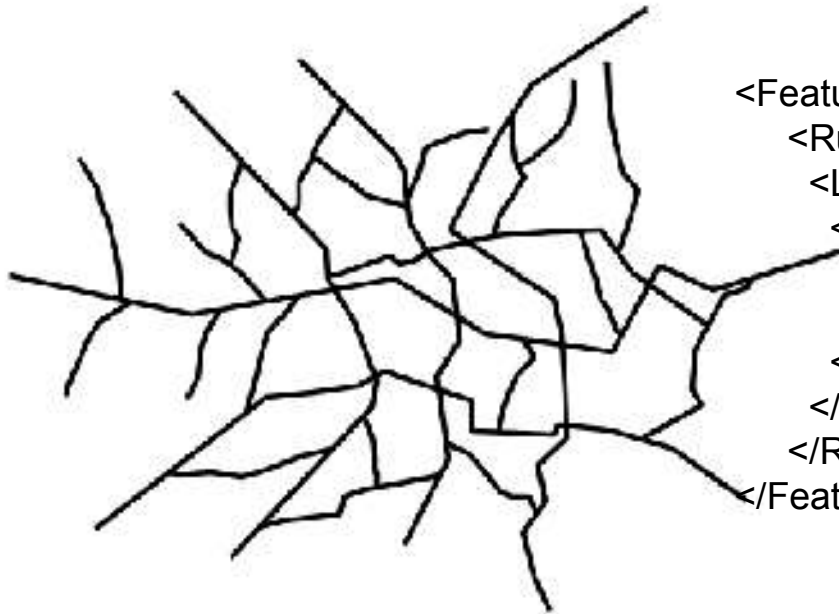


'round' cap



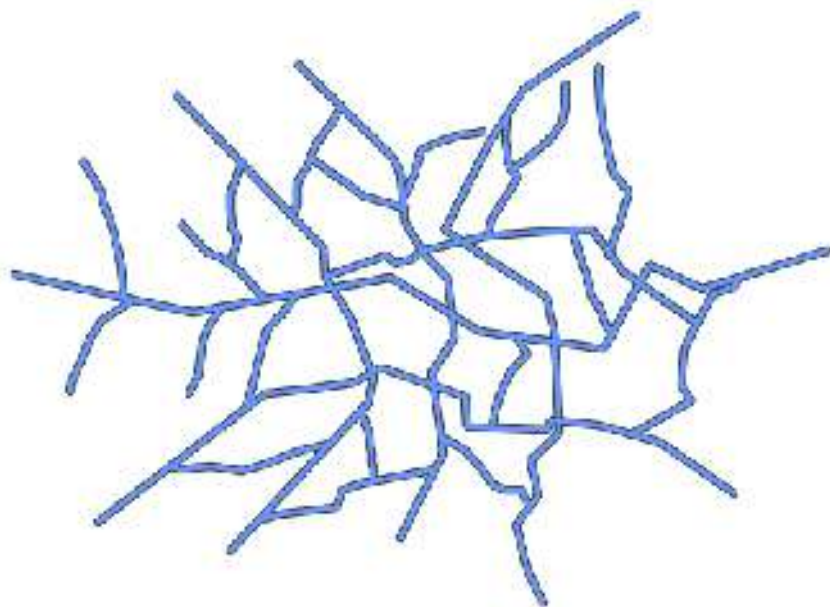
'square' cap

Linha Simples



```
<FeatureTypeStyle>  
  <Rule>  
    <LineSymbolizer>  
      <Stroke>  
        <CssParameter name="stroke">#000000</CssParameter>  
        <CssParameter name="stroke-width">3</CssParameter>  
      </Stroke>  
    </LineSymbolizer>  
  </Rule>  
</FeatureTypeStyle>
```

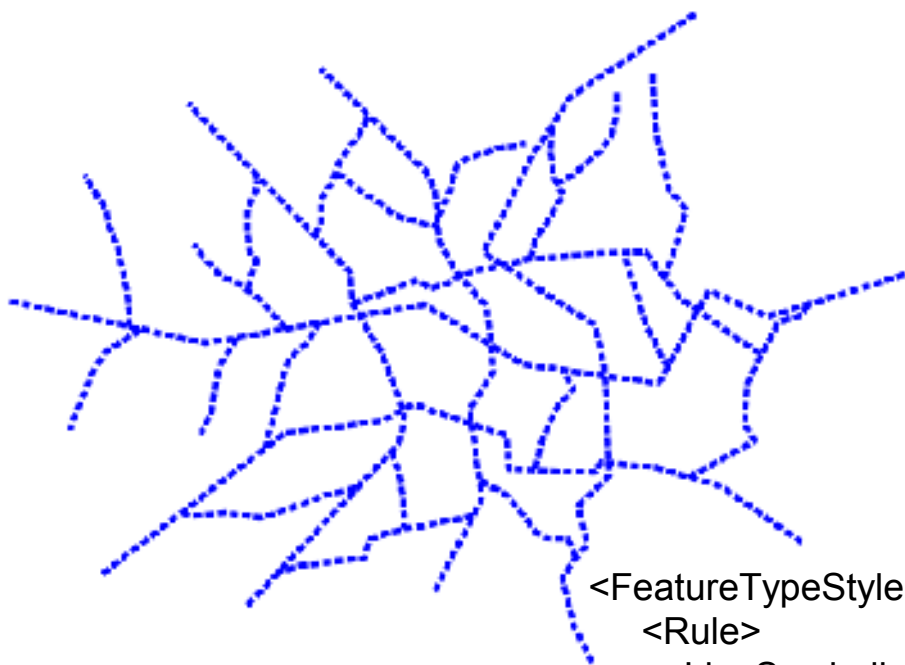
Linha com Borda



```
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#333333</CssParameter>
        <CssParameter name="stroke-width">5</CssParameter>
        <CssParameter name="stroke-linecap">round</CssParameter>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#6699FF</CssParameter>
        <CssParameter name="stroke-width">3</CssParameter>
        <CssParameter name="stroke-linecap">round</CssParameter>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

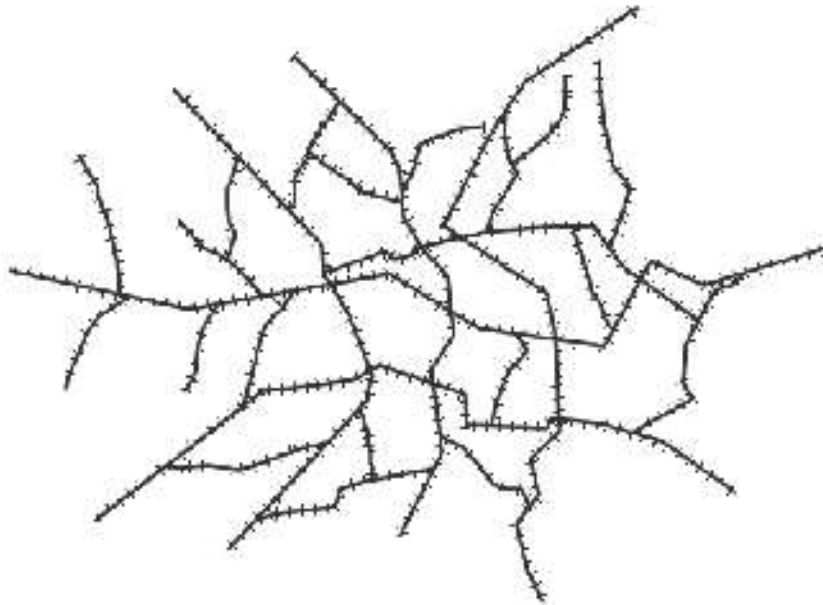
Outra opção seria utilizar dois estilos no mesmo Layer

Linha Pontilhada



```
<FeatureTypeStyle>  
  <Rule>  
    <LineSymbolizer>  
      <Stroke>  
        <CssParameter name="stroke">#0000FF</CssParameter>  
        <CssParameter name="stroke-width">3</CssParameter>  
        <CssParameter name="stroke-dasharray">5 2</CssParameter>  
      </Stroke>  
    </LineSymbolizer>  
  </Rule>  
</FeatureTypeStyle>
```

Linha de Trem



Rule 1:

Linha preta com largura de 3 Pixel

```
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#333333</CssParameter>
        <CssParameter name="stroke-width">3</CssParameter>
      </Stroke>
    </LineSymbolizer>
  </Rule>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <GraphicStroke>
          <Graphic>
            <Mark>
              <WellKnownName>shape://vertline</WellKnownName>
              <Stroke>
                <CssParameter name="stroke">#333333</CssParameter>
                <CssParameter name="stroke-width">1</CssParameter>
              </Stroke>
            </Mark>
            <Size>12</Size>
          </Graphic>
        </GraphicStroke>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Rule 2:

Linha preta sem preenchimento com padrão de linhas verticais pretas com 1 pixel de largura com 12 pixels de comprimento

Símbolos *Shape*

Tag	Descrição
shape://vertline	Linha vertical
shape://horline	Linha Horizontal
shape://slash	Linha diagonal inclinada para frente (barra)
shape://backslash	Mesmo que o anterior, com inclinação contrária (contrabarra)
shape://dot	Um círculo muito pequeno com espaço em volta
shape://plus	Símbolo de adição (+). Adequado para preenchimento
shape://times	Símbolo de multiplicação (×) Adequado para preenchimento
shape://oarrow	Um símbolo de seta aberta. Utilizada para final de linha
shape://carrow	Um símbolo de seta fechada. Utilizada para final de linha

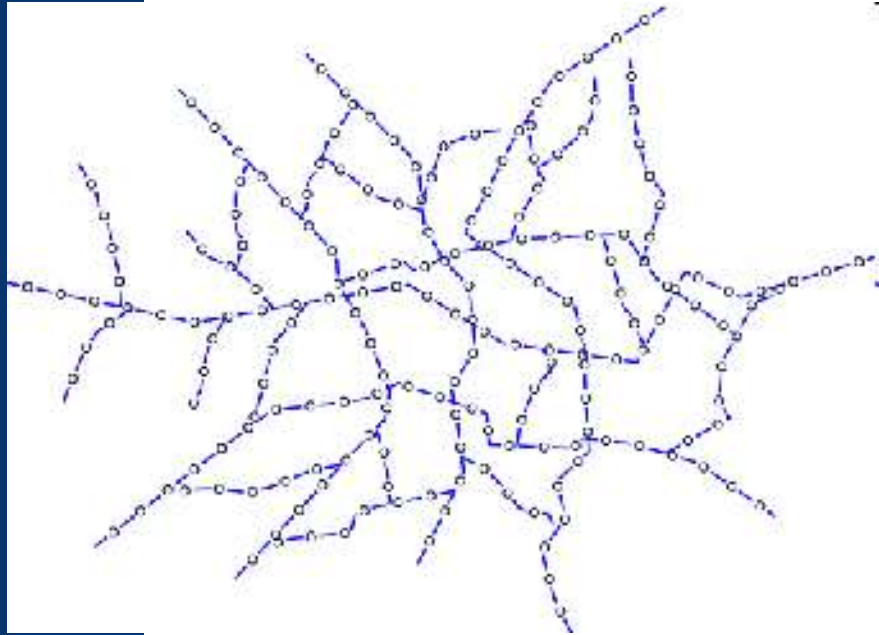
Linha com espaços e gráficos repetidos



```
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <GraphicStroke>
          <Graphic>
            <Mark>
              <WellKnownName>circle</WellKnownName>
              <Stroke>
                <CssParameter name="stroke">#333333</CssParameter>
                <CssParameter name="stroke-width">1</CssParameter>
              </Stroke>
              <Fill>
                <CssParameter name="stroke">#666666</CssParameter>
              </Fill>
            </Mark>
            <Size>4</Size>
            <CssParameter name="stroke-dasharray">4 6</CssParameter>
          </Graphic>
        </GraphicStroke>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Combina um gráfico com o padrão *stroke-dasharray*

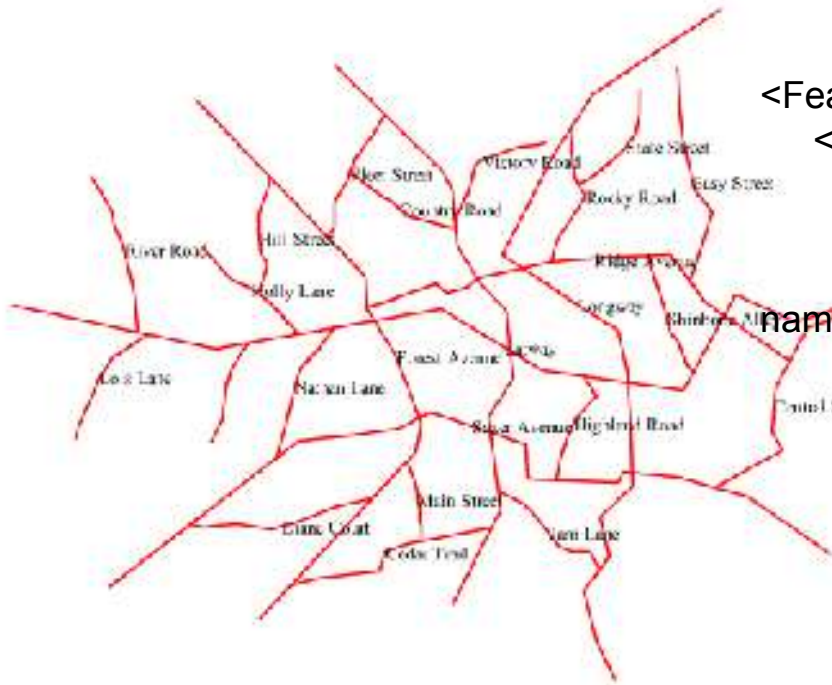
Alternando gráficos e espaços



Elemento GraphicStroke :
Linha com padrão de gráficos
repetidos

```
<FeatureTypeStyle>
  <Rule>
    - <LineSymbolizer>          Linha com espaços
      <Stroke>
        <CssParameter name="stroke">#0000FF</CssParameter>
        <CssParameter name="stroke-width">1</CssParameter>
        <CssParameter name="stroke-dasharray">10 10</CssParameter>
      </Stroke>
    </LineSymbolizer>
    <LineSymbolizer>
      <Stroke>
        <GraphicStroke>
          <Graphic>
            <Mark>
              <WellKnownName>circle</WellKnownName>
              <Stroke>
                <CssParameter name="stroke">#000033</CssParameter>
                <CssParameter name="stroke-width">1</CssParameter>
              </Stroke>
            </Mark>
            <Size>5</Size>
            <CssParameter name="stroke-dasharray">5 15</CssParameter>
            <CssParameter name="stroke-dashoffset">7.5</CssParameter>
          </Graphic>
        </GraphicStroke>
      </Stroke>
    </LineSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

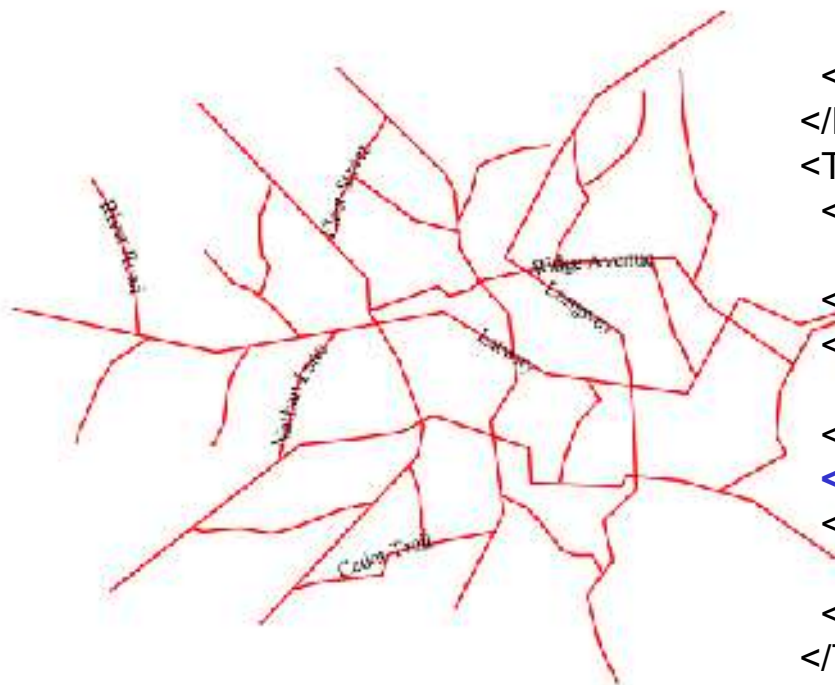
Linha com rótulo



```
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter
name="stroke">#FF0000</CssParameter>
      </Stroke>
    </LineSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>name</ogc:PropertyName>
      </Label>
      <Fill>
        <CssParameter name="fill">#000000</CssParameter>
      </Fill>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

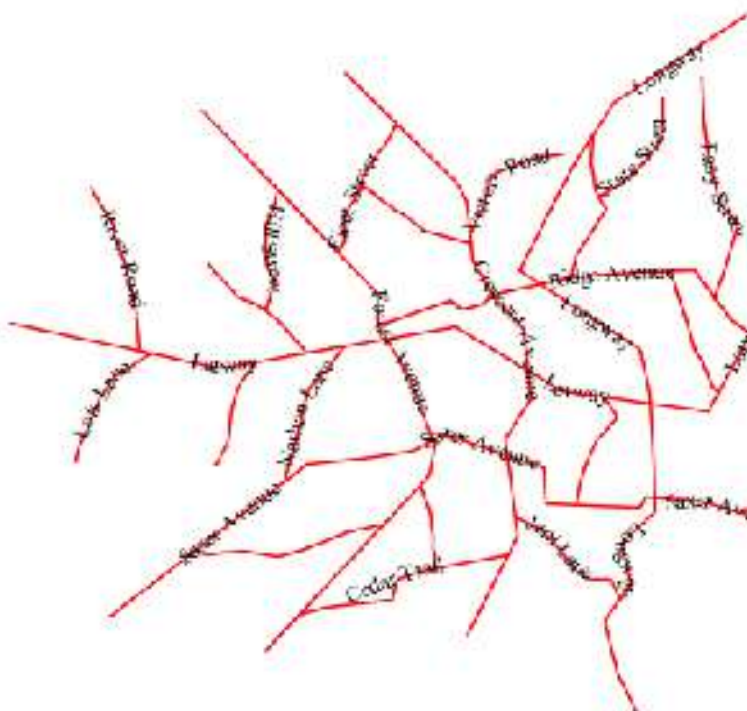
Linhas com rótulo

```
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#FF0000</CssParameter>
      </Stroke>
    </LineSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>name</ogc:PropertyName>
      </Label>
      <Fill>
        <CssParameter name="fill">#000000</CssParameter>
      </Fill>
      <VendorOption name="followLine">true</VendorOption>
      <LabelPlacement>
        <LinePlacement />
      </LabelPlacement>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
```



Importante: este recurso não faz parte da especificação SLD. É um recurso Oferecido pela ferramenta

Otimizando a colocação dos rótulos nas Linhas



```
<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#FF0000</CssParameter>
      </Stroke>
    </LineSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>name</ogc:PropertyName>
      </Label>
      <Fill>
        <CssParameter name="fill">#000000</CssParameter>
      </Fill>
      <VendorOption name="followLine">>true</VendorOption>
      <VendorOption name="maxAngleDelta">90</VendorOption>
      <VendorOption name="maxDisplacement">400</VendorOption>
      <VendorOption name="repeat">150</VendorOption>
      <LabelPlacement>
        <LinePlacement />
      </LabelPlacement>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Importante: este recurso não faz parte da especificação SLD. É um recurso Oferecido pela ferramenta

Elementos de Otimização:

Tag	Obrigatório?	Descrição
< maxAngleDelta>	Não	Ângulo máximo permitindo para rotular a linha
< maxDisplacement>	Não	Este parâmetro é utilizado pelo Geoserver para resolver conflito. O valor indica a distância máxima permitida para deslocar um rótulo na resolução de conflitos.
< repeat>	Não	Determina a distância a partir da qual o rótulo será repetido

Sintaxe:

Os dois elementos mais externos são:

Tag	Obrigatório?	Descrição
<Fill>	Não (Quando <i>Stroke</i> ocorrer)	Determina o estilo do preenchimento do Polígono
<Stroke>	Não (Quando <i>Fill</i> ocorrer)	Determina o estilo do contorno do Polígono

OBS.: O elemento *LineSymbolizer* pode simbolizar polígonos. Eles podem conter informações sobre o contorno do polígono.

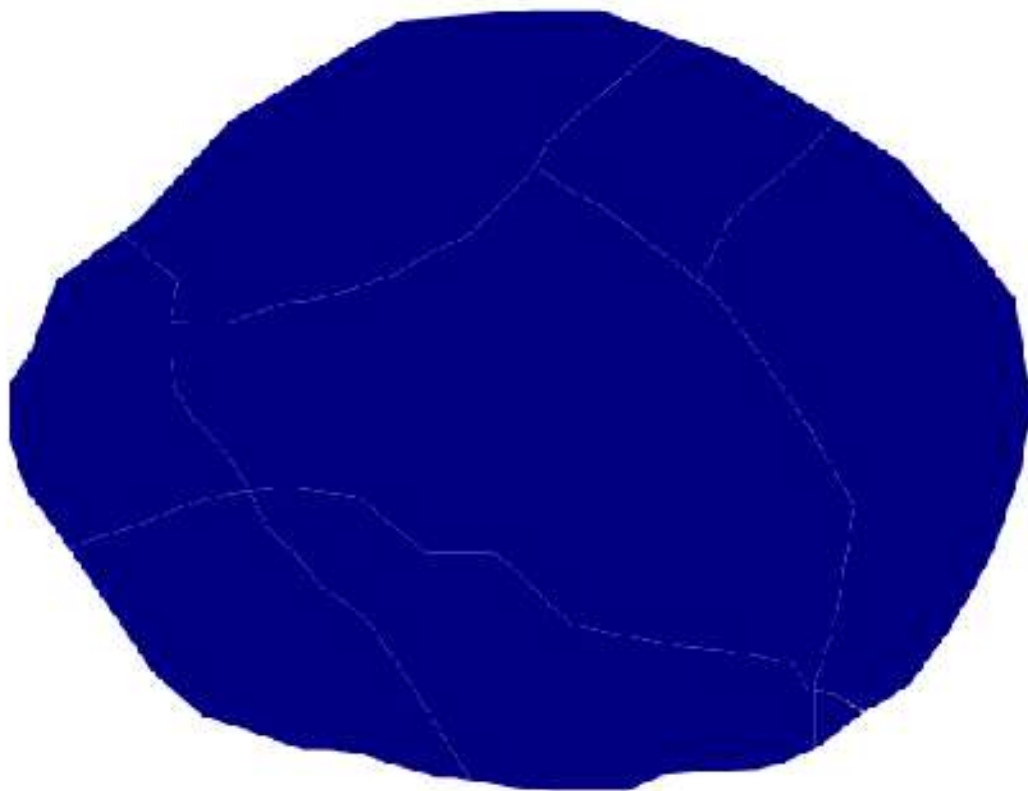
Sintaxe:

O elemento *Fill* pode conter dois elementos

Tag	Obrigatório?	Descrição
<GraphicFill>	Não	Desenha o preenchimento do polígono com um padrão repetido
<CssParameter>	Não	Determina os parâmetros do preenchimento do polígono

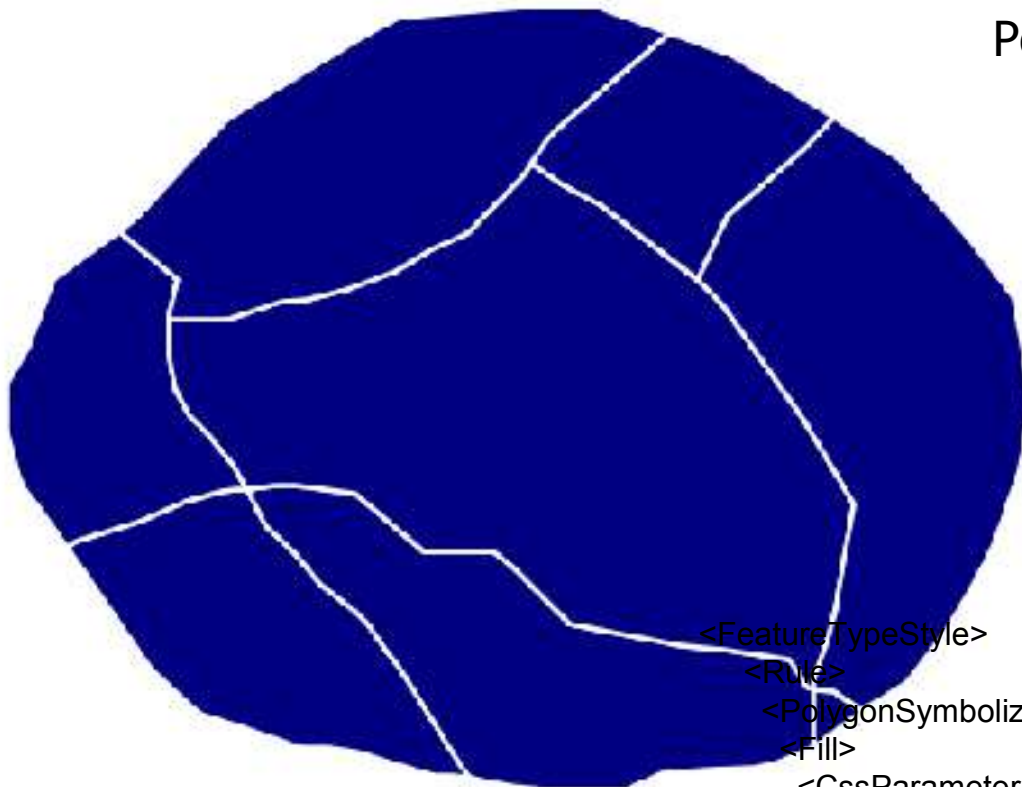
O elemento *CssParameter* pode conter dois elementos

Tag	Obrigatório?	Descrição
name="fill"	Não	Determina a cor do preenchimento do polígono na forma #RRGGBB. O padrão é Cinza (#808080)
name="fill-opacity"	Não	Determina a transparência do preenchimento do polígono. Valores variam entre 0 e 1. Padrão 1 (completamente opaco)



```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter
name="fill">#000080</CssParameter>
      </Fill>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

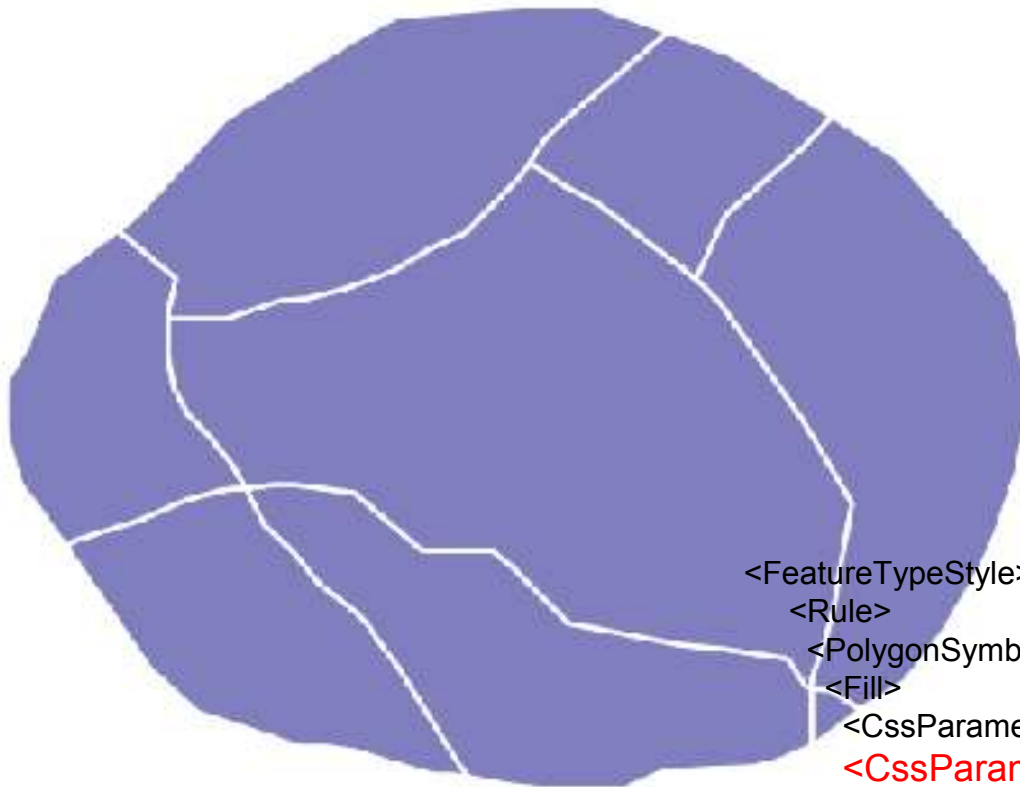

Polígono Simples com Contorno



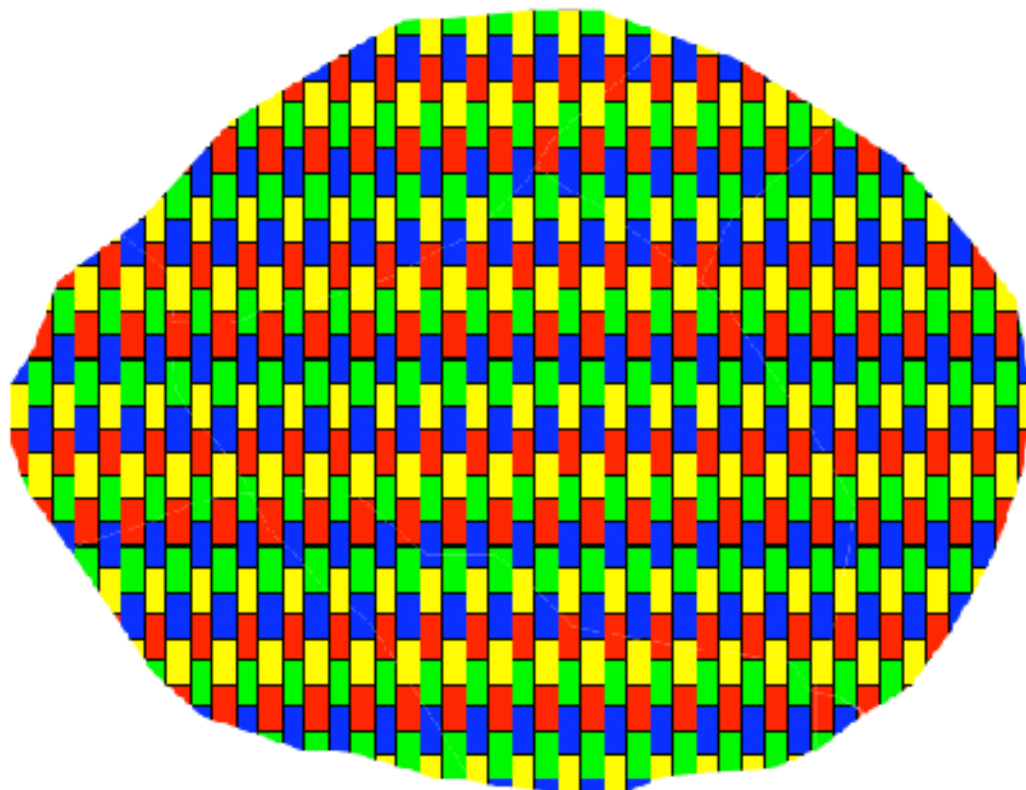
```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">#000080</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">#FFFFFF</CssParameter>
        <CssParameter name="stroke-width">2</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Simbolizando Polígonos - Exemplo 3

no Simples Transparente



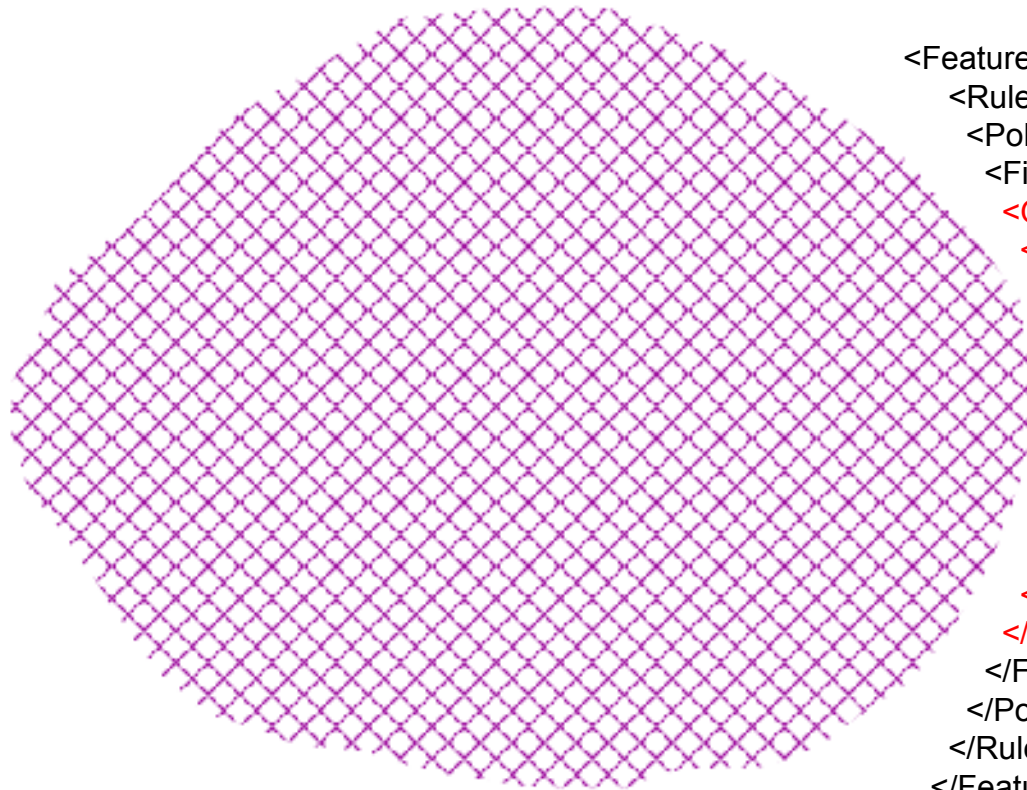
```
<FeatureTypeStyle>  
  <Rule>  
    <PolygonSymbolizer>  
      <Fill>  
        <CssParameter name="fill">#000080</CssParameter>  
        <CssParameter name="fill-opacity">0.5</CssParameter>  
      </Fill>  
      <Stroke>  
        <CssParameter name="stroke">#FFFFFF</CssParameter>  
        <CssParameter name="stroke-width">2</CssParameter>  
      </Stroke>  
    </PolygonSymbolizer>  
  </Rule>  
</FeatureTypeStyle>
```



Polígono Simples
Preenchido com figura

```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <GraphicFill>
          <Graphic>
            <ExternalGraphic>
              <OnlineResource
                xlink:type="simple"
                xlink:href="colorblocks.png" />
              <Format>image/png</Format>
            </ExternalGraphic>
            <Size>93</Size>
          </Graphic>
        </GraphicFill>
      </Fill>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Polígono Simples Preenchido com hachura



```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <GraphicFill>
          <Graphic>
            <Mark>
              <WellKnownName>shape://times</WellKnownName>
              <Stroke>
                <CssParameter name="stroke">#990099</CssParameter>
                <CssParameter name="stroke-width">1</CssParameter>
              </Stroke>
            </Mark>
            <Size>16</Size>
          </Graphic>
        </GraphicFill>
      </Fill>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Polígono Simples Com Rótulo



```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">#40FF40</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">#FFFFFF</CssParameter>
        <CssParameter name="stroke-width">2</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>name</ogc:PropertyName>
      </Label>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Rótulo com *Halo*



```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">#40FF40</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">#FFFFFF</CssParameter>
        <CssParameter name="stroke-width">2</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>name</ogc:PropertyName>
      </Label>
      <Halo>
        <Radius>3</Radius>
        <Fill>
          <CssParameter name="fill">#FFFFFF</CssParameter>
        </Fill>
      </Halo>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
```


Simbolizando Polígonos - Exemplo 8



Polígono com Rótulo Estilizado

```
<FeatureTypeStyle>
  <Rule>
    <PolygonSymbolizer>
      (Fill e Stroke)
    </PolygonSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>name</ogc:PropertyName>
      </Label>
      <Font>
        <CssParameter name="font-family">Arial</CssParameter>
        <CssParameter name="font-size">11</CssParameter>
        <CssParameter name="font-style">normal</CssParameter>
        <CssParameter name="font-weight">bold</CssParameter>
      </Font>
      <LabelPlacement>
        <PointPlacement>
          <AnchorPoint>
            <AnchorPointX>0.5</AnchorPointX>
            <AnchorPointY>0.5</AnchorPointY>
          </AnchorPoint>
        </PointPlacement>
      </LabelPlacement>
      <Fill>
        <CssParameter name="fill">#000000</CssParameter>
      </Fill>
      <VendorOption name="autoWrap">60</VendorOption>
      <VendorOption name="maxDisplacement">150</VendorOption>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Elementos de Otimização:

Tag	Obrigatório?	Descrição
< autoWrap>	Não	Determina o comprimento máximo da Rótulo, após o qual este será quebrado em outra linha. Valor em pixel.
< maxDisplacement>	Não	Este parâmetro é utilizado pelo Geoserver para resolver conflito. O valor indica a distância máxima permitida para deslocar um rótulo na resolução de conflitos. No caso do polígono permite que o valor seja exibido fora de seus limites

Sld Cook Book

<http://docs.geoserver.org/stable/en/user/styling/sld-cookbook/index.html>

