

# Infraestructuras de datos espaciales. Conceptos básicos

**Miguel Rodríguez Luaces**

Laboratorio de Bases de Datos  
Universidade da Coruña



## ¿Quiénes somos?

- Laboratorio de Bases de Datos. Universidade da Coruña
  - Área de Sistemas de Información Geográfica
    - Amplia experiencia en el desarrollo de aplicaciones GIS:
      - Proyecto EIEL (Diputación de A Coruña)
        - <http://www.dicoruna.es/webeiel>
      - Viaxe Virtual de la Biblioteca Virtual Galega
        - <http://bvg.udc.es>
      - Proyecto de desarrollo de Plataforma y Herramientas GIS OpenSource (Xunta de Galicia)
    - Principales áreas de desarrollo de aplicaciones GIS
      - Desarrollo de aplicaciones para la gestión de información geográfica
      - Construcción de infraestructuras de datos espaciales

## Objetivos del curso

- ¿Qué vamos a aprender?
  - ¿Qué es una infraestructura de datos espaciales (IDE)?
  - ¿Qué componentes forman una IDE?
  - ¿Cómo recuperamos información de una IDE?
  - ¿Cómo recuperamos información de objetos geográficos?
  - ¿Cómo recuperamos información de atributos del espacio?
  - ¿Cómo generamos cartografía en una IDE?
  - ¿Cómo representamos y obtenemos metadatos?
  - ¿Cómo obtenemos información de topónimos?
  - ¿Qué software utilizamos para implantar una IDE?
  - ¿Qué software utilizamos a nivel de usuario para utilizar una IDE?

## Contenidos

- **Infraestructuras de datos espaciales**
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Contenidos

- **Infraestructuras de datos espaciales**
  - **Conceptos básicos**
    - Definición de infraestructuras de datos espaciales
    - Arquitectura y componentes
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

## IDE. Conceptos básicos

- **Sistemas de referencia espacial**
  - Es necesario definir la manera en que se asignan coordenadas referidas a la superficie terrestre. Para ello hay que definir el *espacio geográfico*. Consiste en:
    - Definición matemática del espacio
    - Sistema de referencia de coordenadas
  - La definición matemática del espacio determina sus propiedades.
    - Espacios euclídeos: El espacio tradicional que se enseña en geometría
    - Espacios topológicos: Espacio en el que sólo están definidos los conceptos de conectividad y continuidad
    - Lo más común es considerar el espacio euclídeo continuo en 2 o 3 dimensiones modelado mediante el conjunto  $R^2$  ó  $R^3$

## IDE. Conceptos básicos

- Sistemas de referencia espacial
  - A continuación, hay que definir como se ajusta el espacio matemático a la superficie terrestre. Esto es el *sistema de referencia de coordenadas*
  - Un sistema de referencia de coordenadas consiste en:
    - Una aproximación de la superficie de la Tierra mediante el geoide de referencia
    - Una transformación de coordenadas desde el espacio matemático a coordenadas sobre el geoide
  - Lo más común en España:
    - Geoide WGS-84 y coordenadas latitud, longitud, altura
    - Geoide ED50 y coordenadas UTM

## IDE. Conceptos básicos

- Sistemas de referencia espacial. Ejemplo:
  - Definición matemática del espacio
    - Espacio euclídeo continuo en 2 dimensiones ( $R^2$ )
    - Nuestras coordenadas son de la forma (43.30, -8.15)
  - Geoide de referencia (WGS84)
    - Semieje mayor: 6378137.0 metros
    - Aplanamiento:  $1/298.257223563$  ( $\approx 3.35 \text{ ‰}$ )
  - Transformación de coordenadas
    - Se define que (0,0) está en el cruce del meridiano de Greenwich con el Ecuador
    - La primera coordenada es latitud (distancia norte-sur a ese punto medida en grados)
    - La segunda coordenada es longitud (distancia este-oeste a ese punto medida en grados)
    - Coordenadas Facultad Informática UDC: (43.332709, -8.410517)

## IDE. Conceptos básicos

- Sistemas de referencia espacial. Otro ejemplo:
  - Definición matemática del espacio
    - Espacio euclídeo continuo en 2 dimensiones ( $R^2$ )
    - Nuestras coordenadas son de la forma (43.30, -8.15)
  - Geoide de referencia (ED50)
    - Semieje mayor: 6378388.0 metros
    - Aplanamiento: 1/297
  - Transformación de coordenadas
    - Se define que (0,0) está en el cruce del meridiano 9° Oeste con el Ecuador
    - La primera coordenada son metros de distancia al norte de ese punto
    - La segunda coordenada son metros de distancia al este de ese punto
    - Coordenadas Facultad Informática UDC: (4797931, 547787)

## IDE. Conceptos básicos

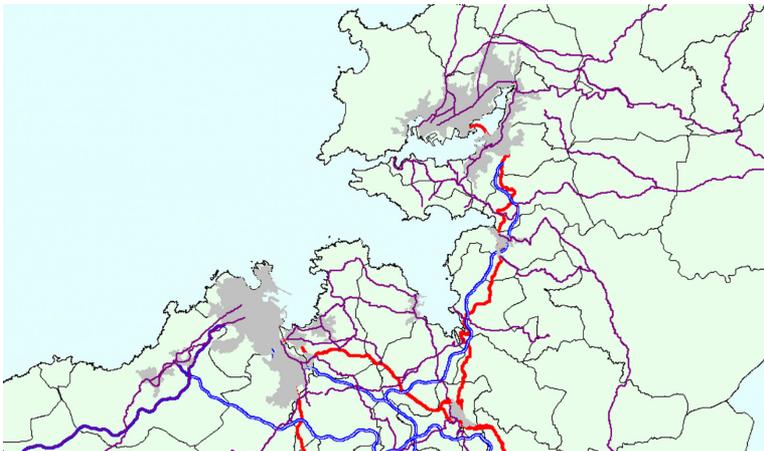
- Sistemas de referencia espacial
  - Existe un estándar internacional para denominar los sistemas de referencia espacial
  - Propuesto por el *European Petroleum Survey Group* (EPSG)
  - Asigna un nombre único (EPSG:xxxx) a cada sistema de referencia espacial
  - Ejemplos:
    - EPSG:4326 → Geoide WGS84 y coordenadas geográficas
    - EPSG:23029 → Geoide ED50 y coordenadas UTM zona 29
  - Dos conjuntos de datos, en diferentes espacios geográficos no se pueden utilizar simultáneamente
    - Ejemplo: en Galicia, hay 150 metros de diferencia entre el mismo punto entre WGS84 y ED50

## IDE. Conceptos básicos

- Objetos geográficos y atributos del espacio
  - Objetos geográficos: son subconjuntos del espacio que se usan para representar la posición o extensión de otras entidades. Por ejemplo:
    - La superficie de una carretera
    - La posición de un manantial
  - Atributos del espacio: son funciones que asocian a cada punto del espacio un valor. Por ejemplo:
    - Temperatura
    - Pendiente del terreno

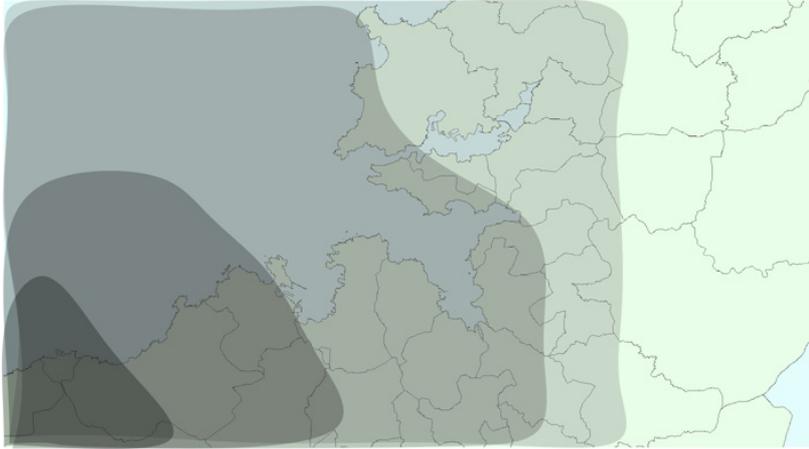
## IDE. Conceptos básicos

- Ejemplo de objetos geográficos



## IDE. Conceptos básicos

- Ejemplo de atributos del espacio



## IDE. Conceptos básicos

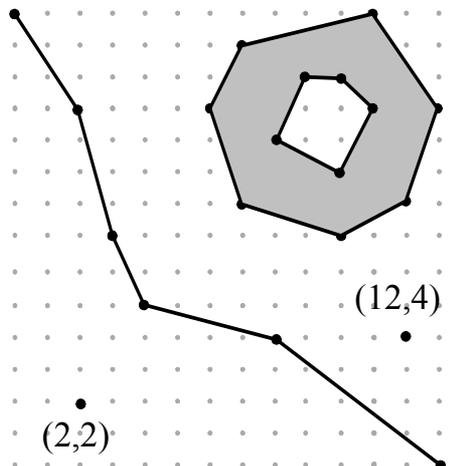
- Objetos geográficos y atributos del espacio
  - Ambos puntos de vista del espacio deben estar presentes en el modelo conceptual
  - Los objetos geográficos son útiles para analizar estructuras hechas por el hombre:
    - Redes de carreteras
    - Divisiones territoriales
    - Catastro
  - Los atributos del espacio son más apropiados para información natural o recolectada por sensores:
    - Meteorología
    - Geología
    - Fotos aéreas (ortofotos, satélite)

## IDE. Conceptos básicos

- Modelo vectorial y modelo raster
  - Modelo vectorial
    - Representa la información geográfica utilizando puntos y segmentos
    - De cada uno se almacenan sus coordenadas usando tipos de datos del ordenador
  - Modelo raster
    - Representa la información geográfica utilizando un array bidimensional de celdas
    - Cada celda almacena los atributos de ese punto del espacio
  - Ambos modelos definen una aproximación del espacio geográfico del modelo conceptual al espacio de almacenamiento del ordenador

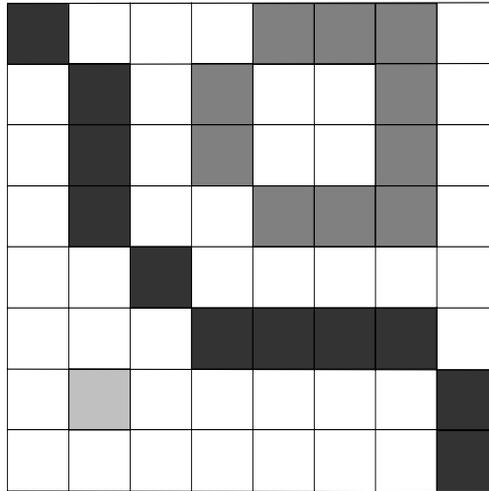
## IDE. Conceptos básicos

- Modelo vectorial



## IDE. Conceptos básicos

- Modelo raster



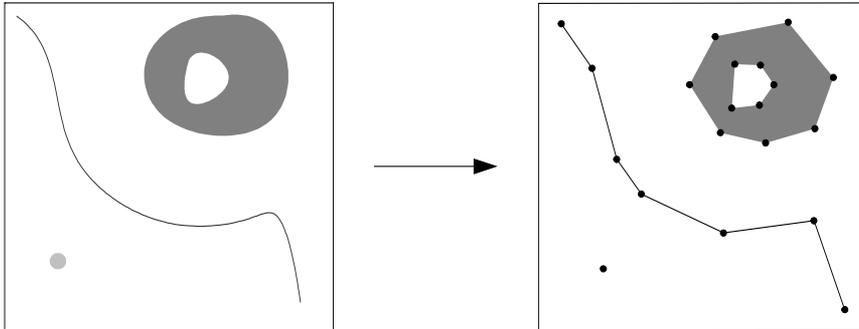
## IDE. Conceptos básicos

- Modelo vectorial y modelo raster

- Cualquier modelo lógico puede usarse para representar tanto objetos geográficos como atributos del espacio.
  - Modelo vectorial
    - Objetos geográficos: discretización de los objetos
    - Atributos del espacio: poligonización de la función
  - Modelo raster
    - Objetos geográficos: pixelización de los objetos
    - Atributos del espacio: discretización de la función
- La elección de uno o de otro depende de la aplicación específica

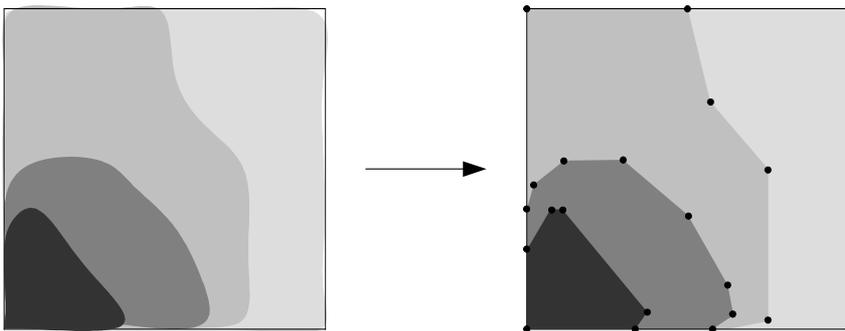
## IDE. Conceptos básicos

- Objetos geográficos en un modelo vectorial



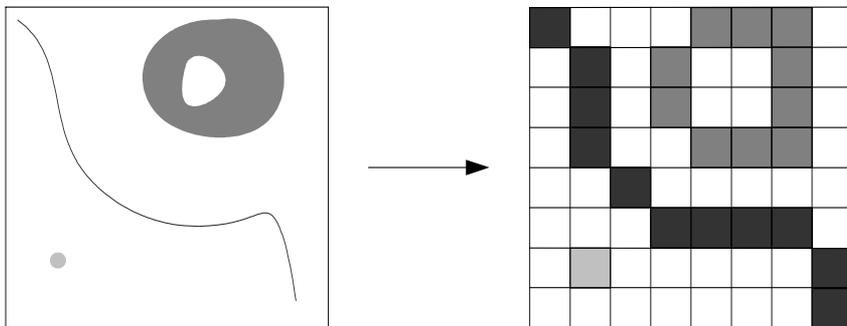
## IDE. Conceptos básicos

- Atributos del espacio en un modelo vectorial



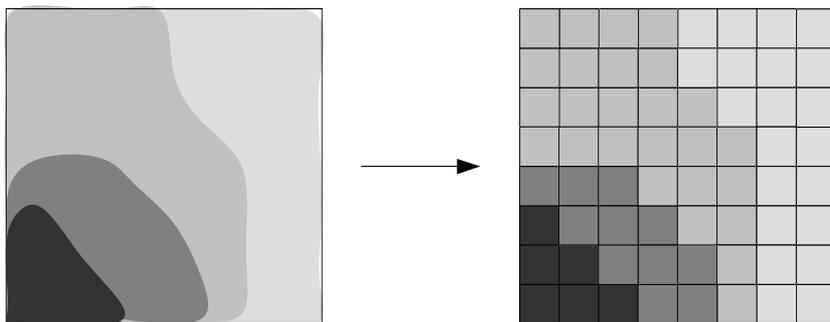
## IDE. Conceptos básicos

- Objetos del espacio en un modelo raster



## IDE. Conceptos básicos

- Atributos del espacio en un modelo raster



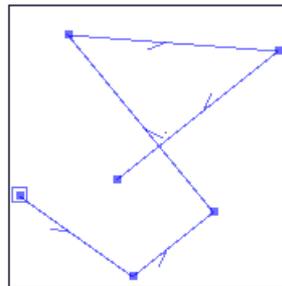
## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - Con los tipos de datos tradicionales los valores se visualizan de forma similar a como se almacenan.
    - Ej: una tabla de enteros se visualiza como una tabla
  - Sin embargo, para la visualización de información geográfica se necesitan abstracciones diferentes a las usadas en la representación en la base de datos
  - Para ello, hay que considerar los siguientes aspectos:
    - Definición de objetos cartográficos
    - Definición y aplicación de estilos
    - Utilización de múltiples representaciones (resolución, tipo)
    - Control de la densidad de la información

## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - Ej: los valores de los tipos de datos requieren representaciones gráficas

```
LINestring(  
  POINT(27.2, 115.8)  
  POINT(33.5, 134.0)  
  .  
  .  
  .  
  POINT(43.3, 13.4)  
)
```



## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - Ej: las relaciones con información geográfica requieren ser representadas como mapas

### CONCELLOS

Nombre	Geometria
Ferrol	g1
A Coruña	g2
Teo	g3
.....	.....
Cee	g315



## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - Se denomina *objeto cartográfico* a la representación gráfica de un objeto geográfico o atributo del espacio.
  - El objeto cartográfico se obtiene a partir de la aplicación de un estilo de visualización al objeto geográfico o atributo del espacio.
  - El estilo de visualización determina las propiedades gráficas de los objetos geográficos o atributos del espacio.
  - Se han definido estándares internacionales para la definición de estilos:
    - OpenGIS: Styled Layer Descriptor [SLD]
    - ISO 19117: Geographic Information – Portrayal

## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - El estilo de visualización debe permitir:
    - Mostrar un objeto cartográfico únicamente a una determinada escala de visualización
      - Ej: los puntos de alumbrado público solo se visualizan a muy alta escala
    - Mostrar objetos cartográficos de diferente resolución en función de la escala visualización
      - Ej: en escalas bajas no se deben mostrar todos los puntos del objeto geográfico
    - Mostrar objetos cartográficos de diferente tipo en función de la escala de visualización
      - Ej: en escalas bajas, las poblaciones se muestran como puntos, en escalas altas, como superficies.
    - Mostrar objetos cartográficos con diferente estilo en función de atributos del objeto geográfico (mapas temáticos).

## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - En el siguiente ejemplo el objeto cartográfico usado para cada objeto geográfico varía en función de la escala



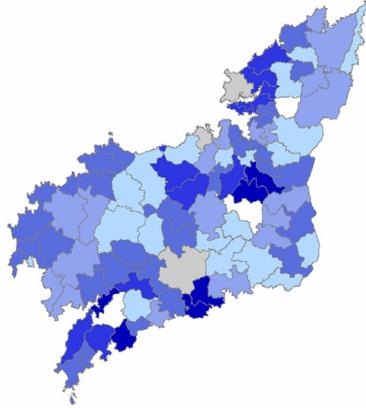
Mapa de escala baja

Mapa de escala media

Mapa de escala alta

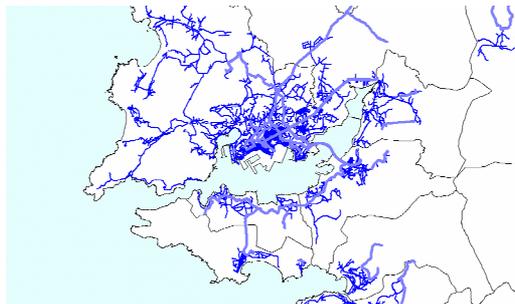
## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - Ejemplo de mapa temático: el estilo del objeto cartográfico depende del atributo *estado de la red de agua*



## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - Dado que el espacio de visualización es limitado, debe existir un método de *control de la densidad de información*.
  - Ej: en este mapa de la red de abastecimiento hay demasiada información ya que no se ven los objetos.

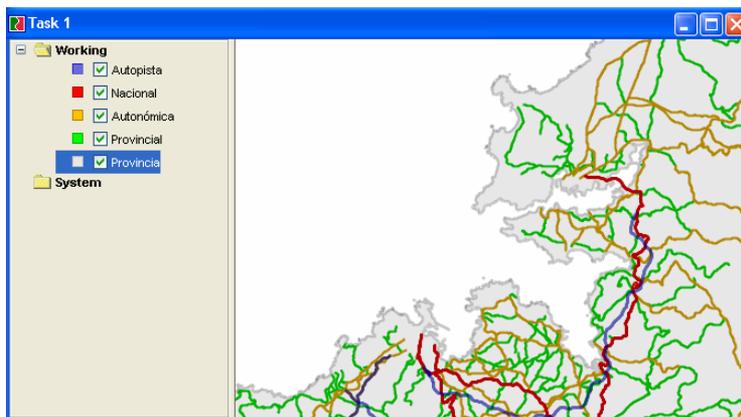


## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - Para visualizar información geográfica en una interfaz de usuario es necesario definir una metáfora de visualización.
  - Esta metáfora describe cómo se organiza, manipula y visualiza la información geográfica.
  - La metáfora más utilizada es la que hace que el interfaz de usuario sea como un mapa en papel.
    - Los objetos cartográficos se organizan en capas de visualización con un estilo común
    - Las capas de visualización se organizan en una pila de capas. Los objetos de la capa superior se colocan sobre los de las inferiores.
  - Hay una diferencia fundamental con los mapas en papel. La información visualizada no es estática.

## IDE. Conceptos básicos

- Objetos cartográficos y capas de visualización
  - El ejemplo muestra un mapa con cinco capas. El orden determina cual oculta a cual.



## IDE. Conceptos básicos

### ■ Tecnología Web

#### □ Servicio web

- Aplicación que permite al usuario interactuar con ella utilizando los protocolos de la Web (HTTP)
- Ejemplo: Google Maps es un servicio web porque permite interactuar con la aplicación no solo con un navegador web, sino también mediante peticiones directas

#### □ Protocolo HTTP

- Conjunto de normas para intercambiar información con un servidor web
- Peticiones GET
  - Toda la información se envía en la URL
- Peticiones POST
  - Alguna información se envía en forma de archivo adjunto

## Contenidos

### ■ Infraestructuras de datos espaciales

#### □ Conceptos básicos

#### □ Definición de infraestructuras de datos espaciales

#### □ Arquitectura y componentes

- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

## IDE. Definición

- Infraestructura de información espacial
  - Metadatos
  - Conjuntos de datos espaciales
  - Servicios de datos espaciales
  - Servicios y tecnologías de red
  - Acuerdos sobre puesta en común, acceso y utilización
  - Mecanismos, procesos y procedimientos de coordinación y seguimiento establecidos

*DIRECTIVA 2007/2/CE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 14 de marzo de 2007 por la que se establece una infraestructura de información espacial en la Comunidad Europea (Inspire)*

*Diario Oficial de la Unión Europea, 25.4.2007, L108/1*

## IDE. Definición

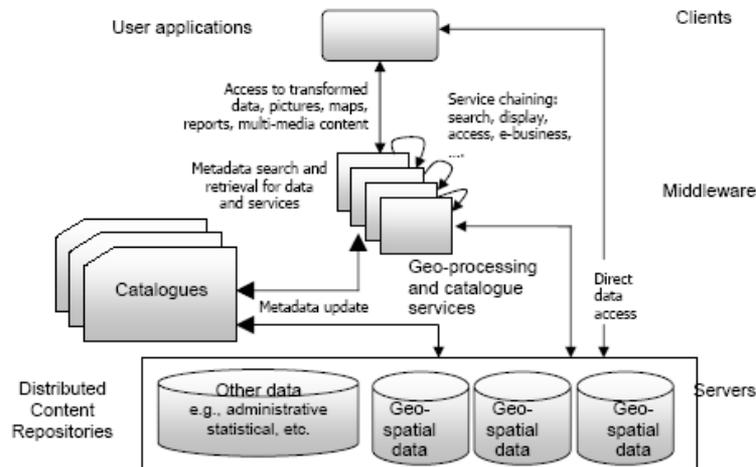
- No es un sistema de información geográfica (SIG)
  - Un SIG es una parte de una IDE. Una IDE implica el uso de estándares y acuerdos de interoperabilidad.
- No es una herramienta SIG
  - Una IDE no se vende en una caja, no es Geomedia o ArcGIS. Estas son herramientas para su implantación.
- En este curso vamos a abordar cuestiones tecnológicas. En una implantación real hay que abordar otras muchas cuestiones

# Contenidos

- **Infraestructuras de datos espaciales**
  - Conceptos básicos
  - Definición de infraestructuras de datos espaciales
  - **Arquitectura y componentes**
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

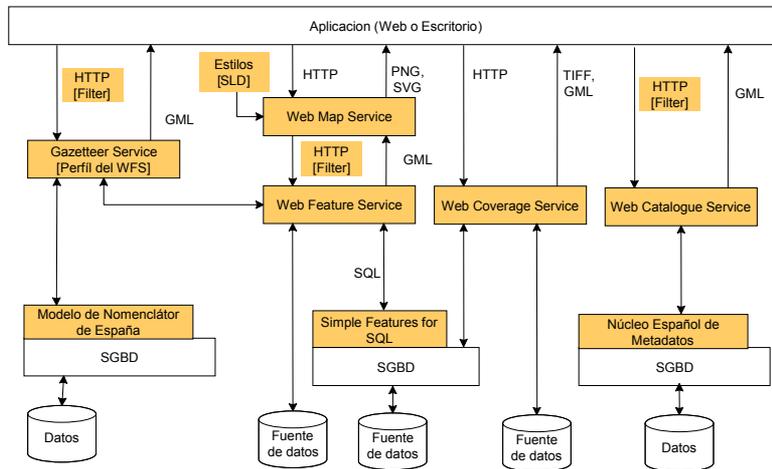
# Arquitectura y componentes

- **Modelo de referencia de INSPIRE**



# Arquitectura y componentes

## ■ Contenido del curso

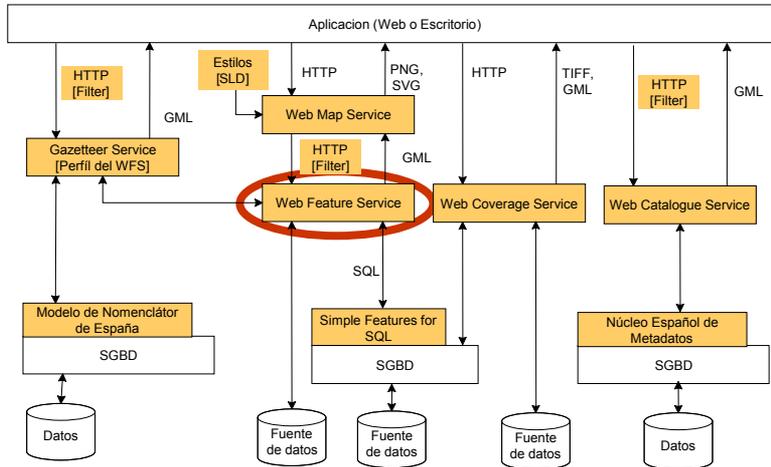


# Contenidos

- Infraestructuras de datos espaciales
- **Recuperación de objetos geográficos [WFS]**
  - Definición
  - Funcionamiento
  - Filtrado de información [Filter]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

## ■ Contenido del curso



# WFS. Definición

- Disponible en:  
<http://www.opengeospatial.org/standards/wfs>
- Define un servicio web para la consulta y modificación de información geográfica.
  - Recibe peticiones usando el protocolo HTTP
  - Devuelve o modifica los objetos geográficos
- Se puede utilizar como mediador
  - Define un interfaz común para acceder a datos que provengan de diferentes fuentes.
- Pueden construirse dos tipos de WFS
  - WFS Básico: Funcionalidad de consulta
  - WFS Transaccional: Funcionalidad de modificación de datos.

## WFS. Definición

- Operaciones definidas en el estándar
  - GetCapabilities: devuelve información del servicio
  - GetFeature: devuelve un conjunto de objetos geográficos
  - DescribeFeatureType: devuelve el esquema del feature type
  - Transaction: cambia la información en el servidor (insertar, borrar o modificar)
  - LockFeature: permite bloquear la información antes de ser modificada.
  - GetFeatureWithLock: integra LockFeature y GetFeature

## Contenidos

- Infraestructuras de datos espaciales
- **Recuperación de objetos geográficos [WFS]**
  - Definición
  - **Funcionamiento**
  - Filtrado de información [Filter]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

## WFS. Funcionamiento

- Ejemplo de GetCapabilities
  - Hacemos la petición por HTTP Get  
**http://url.del.servicio/wfs?request=GetCapabilities**
  - La respuesta es un fichero XML que contiene:
    - Información del servicio
      - Nombre, información de contacto, ...
      - Operaciones soportadas
      - Lenguajes soportados
    - Metadatos de la información
      - Lista de feature types disponibles
      - Sistemas de coordenadas de cada uno
      - Operaciones soportadas

## WFS. Funcionamiento

- Ejemplo de respuesta de GetCapabilities

```
<WFS_Capabilities version="1.0.0" updateSequence="null">
  <Service> ... </Service>
  <Capability> ... </Capability>
  <FeatureTypeList>
    <Operations> ... </Operations>
    <FeatureType>
      <Name>Rivers</Name>
      <Keywords/>
      <SRS>EPSG:4326</SRS>
      <LatLonBoundingBox minx="-21.0"
                                miny="36.0"
                                maxx="62.0"
                                maxy="70.0"/>
    </FeatureType>
  </FeatureTypeList>
</WFS_Capabilities>
```

## WFS. Funcionamiento

- Ejemplo de GetFeature
  - Hacemos la petición por HTTP Post
    - **http://url.del.servicio/wfs?**
  - Adjuntamos un fichero XML que contiene:
    - Nombre del feature type que queremos
    - Propiedades del feature type en el resultado
    - Un filtro (expresado usando Filter Encoding)
    - El formato del resultado

## WFS. Funcionamiento

- Ejemplo de petición GetFeature

```
<GetFeature outputFormat="GML2" ...>
  <Query typeName="Rivers">
    <Filter>
      <BBOX>
        <PropertyName>/Rivers/Geometry</PropertyName>
        <gml:Box>
          <gml:coord><gml:X>-8</gml:X><gml:Y>20</gml:Y></gml:coord>
          <gml:coord><gml:X>32</gml:X><gml:Y>76</gml:Y></gml:coord>
        </gml:Box>
      </BBOX>
    </Filter>
  </Query>
</GetFeature>
```

## WFS. Funcionamiento

- Ejemplo de Transaction
  - Hacemos la petición por HTTP Post
    - **http://url.del.servicio/wfs?**
  - Adjuntamos un fichero XML que contiene un conjunto de ordenes *insert*, *delete* o *update*
    - *Insert*: indicamos la nueva *feature*
    - *Delete*: indicamos con un filtro la *feature* a borrar
    - *Update*: indicamos la *feature* a cambiar y su valor.

## WFS. Funcionamiento

- Ejemplo de petición Transaction

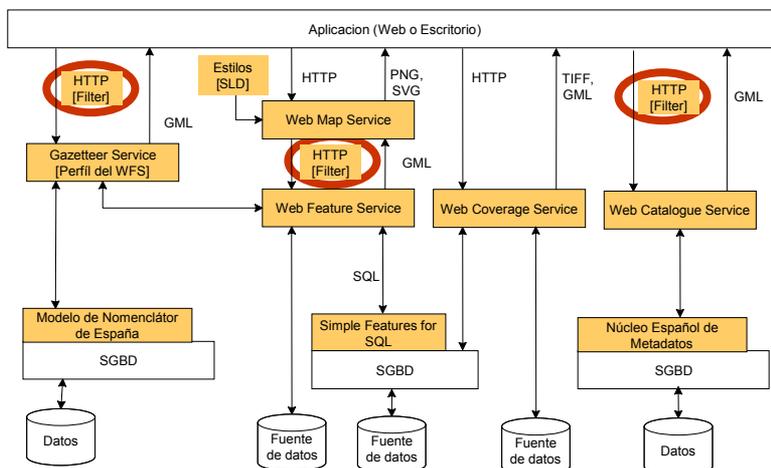
```
<Transaction ...>
<Insert>
  <Cities fid="ID2"> ... </Cities>
</Insert>
<Delete typeName="Cities">
  <ogc:Filter> ... </ogc:Filter>
</Delete>
<Update typeName="Cities">
  <Property>
    <Name>Cities/Name</Name>
    <Value>updated city name</Value>
  </Property>
  <ogc:Filter> ... </ogc:Filter>
</Update>
</Transaction>
```

# Contenidos

- Infraestructuras de datos espaciales
- **Recuperación de objetos geográficos [WFS]**
  - Definición
  - Funcionamiento
  - **Filtrado de información [Filter]**
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

- Contenido del curso



## WFS. Filter encoding

- Disponible en:
  - <http://www.opengeospatial.org/standards/filter>
- Define un lenguaje para definir filtros de consultas.
  - Operaciones del WFS
  - Peticiones a catálogos
- Define un conjunto de operadores espaciales y alfanuméricos y un lenguaje XML para representarlos

## WFS. Filter encoding

- Operaciones
  - Predicados espaciales (de Simple Features for SQL)
    - *equals, disjoint, touches, within, overlaps, crosses, intersects, contains, bbox*
  - Predicados métricos
    - *dwithin*: la distancia es menor que una cantidad
    - *beyond*: la distancia es mayor que una cantidad
  - Comparación (de SQL)
    - *isEqualTo, isLike, isBetween*, etc.
  - Operadores lógicos (booleanos)
    - *and, not, or*
  - Expresiones, funciones, operaciones aritméticas, etc.

## WFS. Filter encoding

- Ejemplo de filtro:
  - Devuelve aquellas *features* cuya geometría esté contenida en un rectángulo.

```
<Filter>
  <BBOX>
    <PropertyName>/Rivers/Geometry</PropertyName>
    <gml:Box>
      <gml:coord><gml:X>-8</gml:X><gml:Y>20</gml:Y></gml:coord>
      <gml:coord><gml:X>32</gml:X><gml:Y>76</gml:Y></gml:coord>
    </gml:Box>
  </BBOX>
</Filter>
```

## WFS. Filter encoding

- Ejemplo de filtro complejo

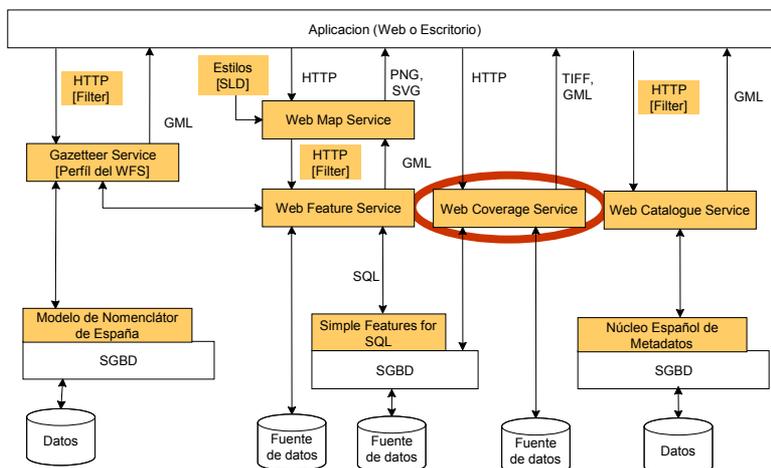
```
<ogc:Filter>
  <ogc:And>
    <ogc:PropertyIsLike wildCard="*" singleChar="?" escape="\">>
      <ogc:PropertyName>/Europe/City</ogc:PropertyName>
      <ogc:Literal>London</ogc:Literal>
    </ogc:PropertyIsLike>
    <ogc:BBOX>
      <ogc:PropertyName>/Europe/Border</ogc:PropertyName>
      <gml:Box>
        <gml:coord><gml:X>1</gml:X><gml:Y>40</gml:Y></gml:coord>
        <gml:coord><gml:X>12</gml:X><gml:Y>56</gml:Y></gml:coord>
      </gml:Box>
    </ogc:BBOX>
  </ogc:And>
</ogc:Filter>
```

# Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- **Recuperación de atributos del espacio [WCS]**
  - Definición
  - Funcionamiento
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

- Contenido del curso



## WCS. Definición

- Disponible en:  
<http://www.opengeospatial.org/standards/wcs>
- Define:
  - Un servicio web para consultar atributos del espacio (coberturas).
- Proporciona las siguientes operaciones:
  - GetCapabilities: devuelve información del servicio
  - DescribeCoverage: devuelve información de una cobertura específica.
  - GetCoverage: devuelve los datos de una cobertura.

## Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- **Recuperación de atributos del espacio [WCS]**
  - Definición
  - **Funcionamiento**
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

## WCS. Funcionamiento

- Ejemplo de GetCoverage
  - Hacemos la petición por HTTP Get  
`http://url.del.servicio/wcs?  
REQUEST=GetCoverage&  
LAYER=europe&  
SRS=EPSG:4326&  
BBOX=-10,42,-6,44&  
WIDTH=600&HEIGHT=400&  
FORMAT=jpg`
  - Indicando en la petición
    - Nombre de la cobertura.
    - Sistema de coordenadas de los datos
    - Área del mapa deseada
    - Tamaño de la imagen resultante
    - Formato de la imagen

## WCS. Funcionamiento

- Ejemplo de GetCoverage
  - El resultado es:

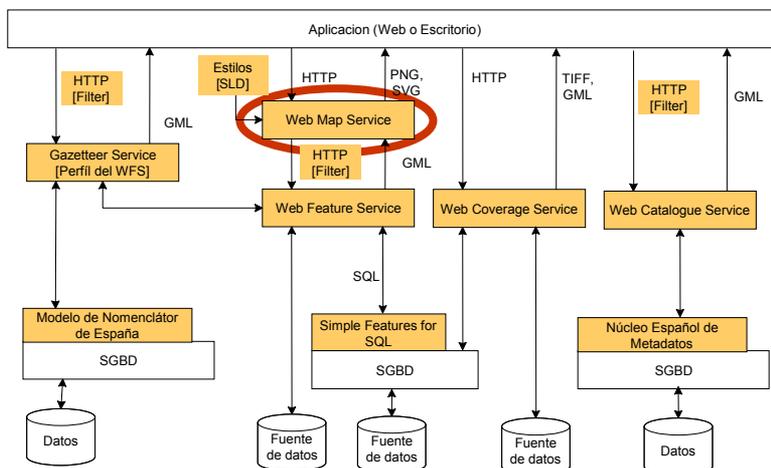


# Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- **Creación de cartografía [WMS]**
  - Definición
  - Funcionamiento
  - Definición de estilos de visualización [SLD]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

- Contenido del curso



## WMS. Definición

- Disponible en:  
<http://www.opengeospatial.org/standards/wms>
- Define un servicio web que genera cartografía a partir de información geográfica
- Mediante una petición HTTP permite indicar:
  - Capas a visualizar
  - Estilos a utilizar
  - Formato de imagen
  - Resolución de la imagen

## WMS. Definición

- Existen dos tipos de servicios
  - *WMS Básico*: los estilos están predefinidos y no se pueden cambiar.
  - *WMS con SLD*: los estilos se definen utilizando el estándar SLD (*SLD-enabled WMS*)
- Las operaciones que debe soportar cualquier WMS son:
  - *GetCapabilities*: devuelve información del servicio.
  - *GetMap*: construye y devuelve una mapa como una imagen.
  - *GetFeatureInfo*: devuelve información acerca de los objetos representados en un pixel de la imagen.

## WMS. Definición

- El servicio WMS permite la producción de cartografía como imágenes y como formatos vectoriales.
- Los formatos de imagen usualmente soportados son PNG, JPG, etc...
- Los formatos vectoriales utilizan descripciones geométricas en lugar de píxeles.
- El más común es SVG (*Scalable Vector Graphics*), un lenguaje basado en XML propuesto por el W3C.
- No confundir GML con SVG.
  - GML representa objetos geográficos (información).
  - SVG representa objetos cartográficos (mapas).

## Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- **Creación de cartografía [WMS]**
  - Definición
  - **Funcionamiento**
  - Definición de estilos de visualización [SLD]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

## WMS. Funcionamiento

- Ejemplo de GetCapabilities
  - Hacemos la petición por HTTP Get  
**http://url.del.servicio/wms?request=GetCapabilities**
  - La respuesta es un fichero XML que contiene:
    - Información del servicio
      - Nombre, información de contacto, ...
      - Operaciones soportadas
      - Lenguajes soportados
    - Metadatos de la información
      - Lista de *layers* disponibles
      - Sistemas de coordenadas de cada uno
      - Área geográfica cubierta por la capa
      - Estilos aplicables a cada capa

## WMS. Funcionamiento

- Ejemplo de respuesta de GetCapabilities

```
<WMT_MS_Capabilities version="1.1.1" updateSequence="1.1.0">
  <Service> ... </Service>
  <Capability>
    <Layer queryable="1" opaque="0" ...>
      <Name>europe:country</Name>
      <SRS>EPSG:31466</SRS>
      <LatLonBoundingBox minx="-31.27" miny="32.56"
        maxx="68.92" maxy="81.86"/>
      <Style>
        <Name>default:europe:country</Name>
        <LegendURL width="20" height="20"> ... </LegendURL>
      </Style>
      <ScaleHint max="1.7976931348623157E308" min="0.0"/>
    </Layer>
  </Capability>
</WMT_MS_Capabilities>
```

## WMS. Funcionamiento

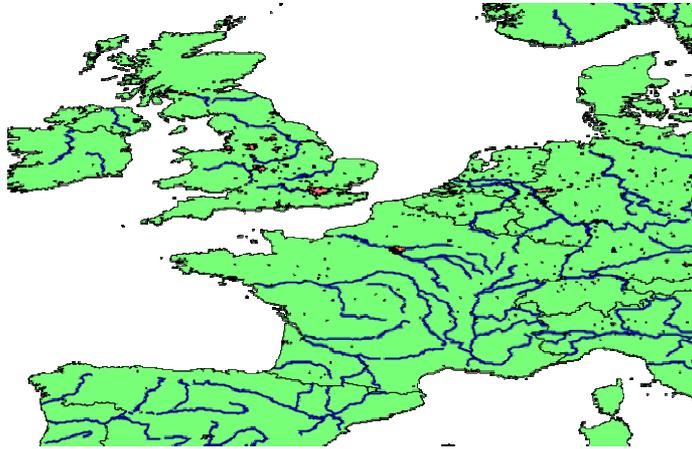
- Ejemplo de GetMap
  - Hacemos la petición por HTTP Get,
    - **http://url.del.servicio/wms?**  
**REQUEST=GetMap&**  
**LAYERS=europe:country,**  
**europe:major\_rivers,**  
**europe:major\_urban\_places&**  
**STYLES=default,default,default&**  
**SRS=EPSG:4326&**  
**BBOX=-10,40,12,60&**  
**WIDTH=640&HEIGHT=410&**  
**FORMAT=image/png&**  
**BGCOLOR=0xaaaaff&**  
**TRANSPARENT=true&**  
**EXCEPTIONS=application/vnd.ogc.se\_inimage**

## WMS. Funcionamiento

- Ejemplo de GetMap
  - Indicando:
    - Lista de capas que deben componer el mapa.
      - La primera de la lista es la de más abajo en el mapa.
    - Lista de estilos para cada capa
    - Sistema de proyección a utilizar
    - Área del mapa deseada
    - Tamaño de la imagen resultante
    - Formato de la imagen
    - Color de fondo, y si el fondo es transparente
    - Formato de las excepciones

## WMS. Funcionamiento

- Ejemplo de GetMap
  - Y el resultado es:

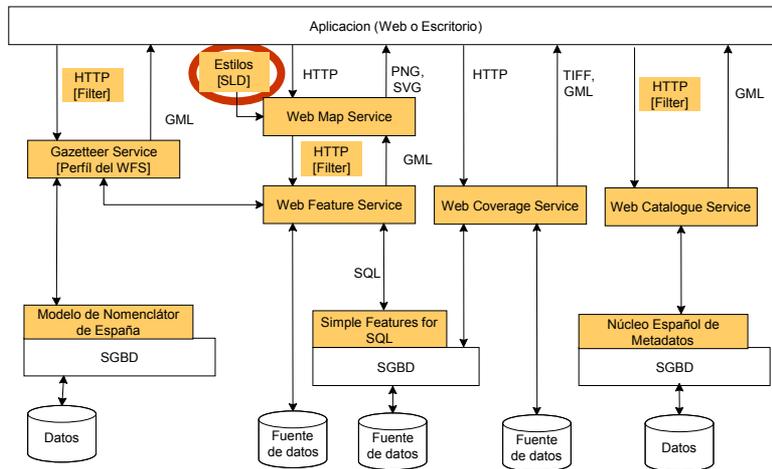


## Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- **Creación de cartografía [WMS]**
  - Definición
  - Funcionamiento
  - **Definición de estilos de visualización [SLD]**
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

## ■ Contenido del curso



Octubre 2007

IDEs. Conceptos básicos

75

## WMS. SLD

- La especificación WMS proporciona:
  - La definición de un interfaz común para acceder a servicios de generación de cartografía.
  - La posibilidad de integrar cartografía de diferentes fuentes utilizando imágenes de fondo transparente.
- Sin embargo, ciertos requerimientos no son satisfechos:
  - De los estilos sólo se conoce el nombre, pero no su definición.
  - No existe un estándar para la definición de estilos.
  - Los clientes del servicio no pueden definir sus propios estilos para la información geográfica.

Octubre 2007

IDEs. Conceptos básicos

76

## WMS. SLD

- Para solucionar estos problemas, se ha definido el lenguaje SLD (*Styled Layer Descriptor*). Disponible en: <http://www.opengeospatial.org/standards/sld>
- Un WMS con SLD proporciona esta funcionalidad:
  - Utilizar el lenguaje SLD como una biblioteca.
    - El desarrollador define el estilo usando SLD.
    - El cliente puede consultar la definición del estilo.
  - Utilizar el lenguaje para definir nuevos estilos
    - El cliente muestra la información con un estilo propio.
    - El servicio puede almacenar los nuevos estilos.
  - Implementar un servicio *WMS en cascada*
    - El lenguaje SLD permite indicar en el estilo la fuente de datos. El WMS se usa como un *pintor* de información geográfica.

## WMS. SLD

- Las operaciones de un WMS con SLD son:
  - Operaciones ampliadas
    - *GetCapabilities*: debe proporcionar información adicional referente a la funcionalidad SLD soportada.
    - *GetMap*: debe permitir indicar los estilos definidos por el usuario.
  - Operaciones nuevas
    - *DescribeLayer*: devuelve información acerca de las capas definidas en el servicio.
    - *GetLegendGraphic*: devuelve el icono que representa al estilo.
    - *GetStyles*: permite consultar los estilos.
    - *UpdateStyles*: permite modificar los estilos definidos.



## WMS. SLD

- Define un lenguaje XML para representar la definición de estilos de visualización.
  - Capas y origen de los datos
  - Apariencia gráfica de los estilos
- Una capa puede ser predefinida (*NamedLayer*) o definida por el usuario (*UserLayer*).
- Para cada capa, se pueden definir filtros (*LayerFeatureConstraints* y *FeatureTypeConstraint*).
- Una capa definida por el usuario puede provenir de una fuente de datos remota (*RemoteOWS*).
- Una capa puede tener un estilo predefinido (*NamedStyle*) o definido por el usuario (*UserStyle*).



## WMS. SLD

- Un estilo definido por el usuario está compuesto estilos definidos para cada *feature type* (*FeatureTypeStyle*).
- Para cada uno de ellos, se puede definir un conjunto de reglas que determinan las condiciones en las que se usa el estilo (*Rule*).
- Cada regla tiene una escala máxima y mínima de visualización, y un filtro que determina a que objetos se aplica la regla.
- Además, cada regla tiene asociado uno o varios simbolizadores que determinan como se pintan los objetos (*Symbolizer*).



---

## WMS. SLD

- Existe un simbolizador para cada tipo de estilo.
- Cualquier objeto puede pintarse con cualquier simbolizador.
- El simbolizador de puntos permite asociar símbolos a los objetos.
- El simbolizador de líneas permite representar los bordes de los objetos.
- El simbolizador de polygons permite rellenar los objetos.
- El simbolizador de textos permite colocar etiquetas a los objetos.
- El simbolizador de raster permite ajustar como se muestran las coberturas.



---

## WMS. SLD

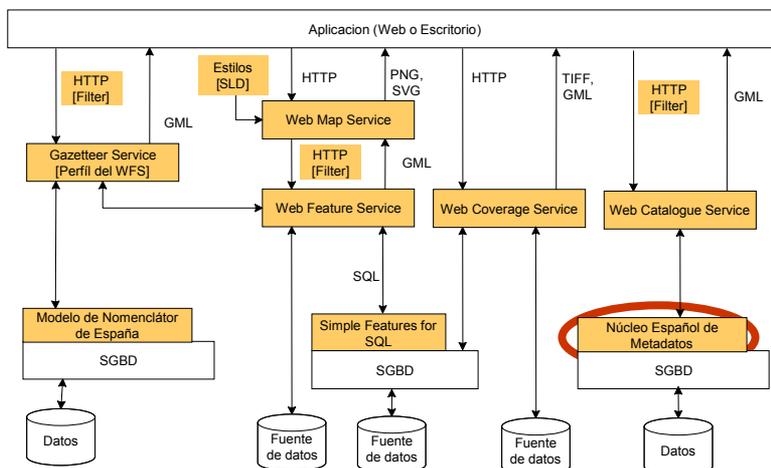
- Cómo se pueden hacer cosas con SLD
  - Mostrar un objeto cartográfico en función de la escala.
    - Se utiliza el rango de escalas de las reglas del estilo.
  - Mostrar objetos de distinta resolución en función de la escala.
    - Se utilizan las reglas de los estilos y distintos atributos geográficos del objeto.
  - Mostrar objetos de distinto tipo en función de la escala.
    - De igual manera que el anterior.
  - Mostrar mapas temáticos.
    - Se utilizan las reglas y filtros para seleccionar los objetos a los que se aplica cada estilo.

# Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- **Representación y consulta de metadatos [CS-W]**
  - **Representación de metadatos [NEM, ISO19115]**
  - Servicio de catálogo de metadatos
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

- Contenido del curso

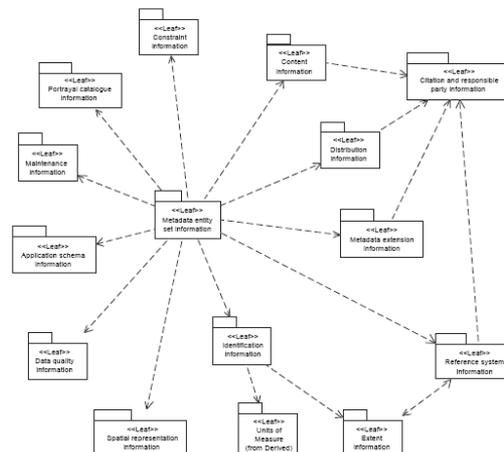


# Metadatos. NEM

- ISO 19915
  - Norma internacional de metadatos perteneciente a la familia ISO 19100
  - Desarrollada por el Comité Técnico 211 de la Organización de Estandarización Internacional (ISO) (<http://www.isotc211.org/>)
  - Proporciona un modelo de metadatos
  - Establece un conjunto común de terminología, definiciones y procedimientos de ampliación para metadatos

# Metadatos. NEM

- ISO 19115





## Metadatos. NEM

- ISO 19115. Problemas:
  - Muy amplia: tiene 409 elementos
  - Muy voluminosa: es un documento de 140 páginas
  - Muy compleja: en cuanto a la terminología usada para describir sus elementos
  - Demasiado general: trata tantos aspectos que es difícil de implementar si no se acota seleccionando ciertos parámetros, posibilidades y variaciones que se definen como opcionales.



## Metadatos. NEM

- Nucleo español de Metadatos
  - Recomendación definida por el Grupo de Trabajo de la IDEE (<http://www.idee.es/>)
  - Establecida en forma de perfil de ISO19115
  - Formado por la ampliación del Núcleo (Core) de la Norma ISO 19115 de Metadatos con los elementos necesarios de otras normas referentes a los metadatos, como son Dublín Core Metadata

## Metadatos. NEM

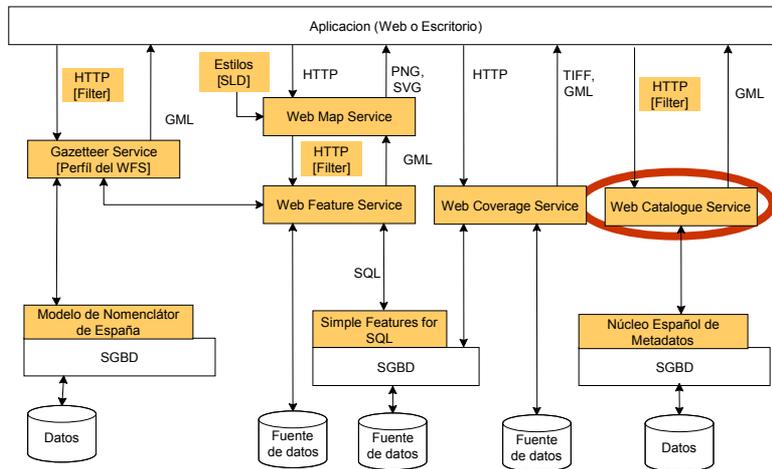
- Creación de metadatos
  - CatMDEdit (<http://catmdedit.sourceforge.net/>)
  - Aplicación de escritorio para la creación y edición de metadatos siguiendo ISO19115 y el NEM
  - Desarrollada por el consorcio español TeIDE
  - Es software libre (código abierto), multilingüe, y trabaja con dos interfaces visuales, una detallada según el estándar ISO19115 y otra ajustada al NEM
  - Compatible con otros estándares como Dublin Core
  - Posee posibilidad de importación/exportación en XML y en otros formatos, como Excel.
  - Facilita la edición de palabras clave a través de tesauros

## Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- **Representación y consulta de metadatos [CS-W]**
  - Representación de metadatos [NEM, ISO19115]
  - **Servicio de catálogo de metadatos**
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

## ■ Contenido del curso



# Metadatos. CS-W

## ■ Disponible en:

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

## ■ Define un servicio web para consultar metadatos

## ■ Proporciona las siguientes operaciones:

- GetCapabilities: describe el servicio
- DescribeRecord: describe la estructura de los registros devueltos por el servicio
- GetDomain: describe el dominio de los valores que pueden tomar los atributos de los registros
- GetRecords: permite realizar consultas al servicio
- GetRecordByID: permite realizar consultas al servicio conociendo el identificador del metadato deseado

## Metadatos. CS-W

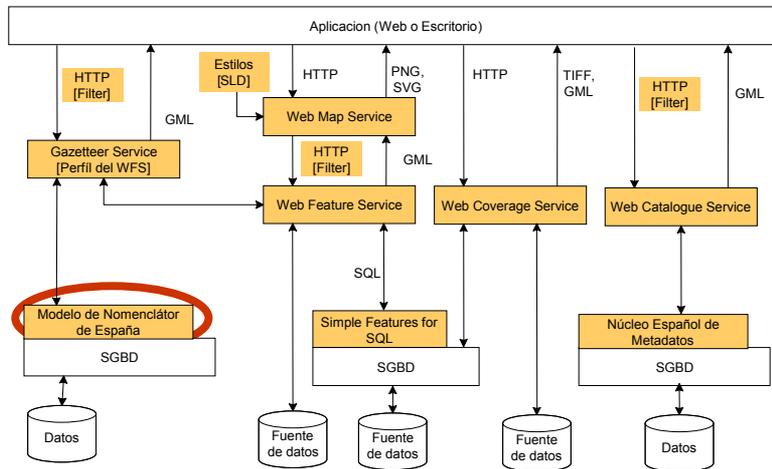
- Las operaciones anteriores son de consulta.
- Opcionalmente, el servicio puede implementar operaciones de actualización
  - Transaction: permite insertar, borrar o modificar un registro
  - Harvest: permite extraer registros de una fuente externa
- La diferencia entre las operaciones es que con la primera el usuario crea el nuevo registro, mientras que con la segunda los registros se extraen de otra fuente de datos

## Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- **Representación y consulta de topónimos [WFS-G]**
  - **Representación de topónimos [MNE]**
  - Servicio de nomenclátor
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

## ■ Contenido del curso

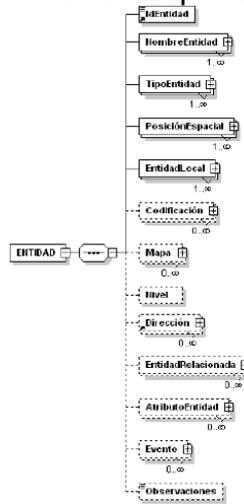


# Nomenclátor. MNE

- Modelo de Nomenclátor de España (MNE):
  - Definido por el Consejo Superior Geográfico
  - Estructura de datos cuya finalidad es el almacenamiento y gestión de los nombres geográficos
  - Define un modelo de datos adaptado a la realidad de España
  - Establece el conjunto de atributos que se consideran fundamentales para caracterizar a un topónimo
  - Compatible con otras normas y especificaciones

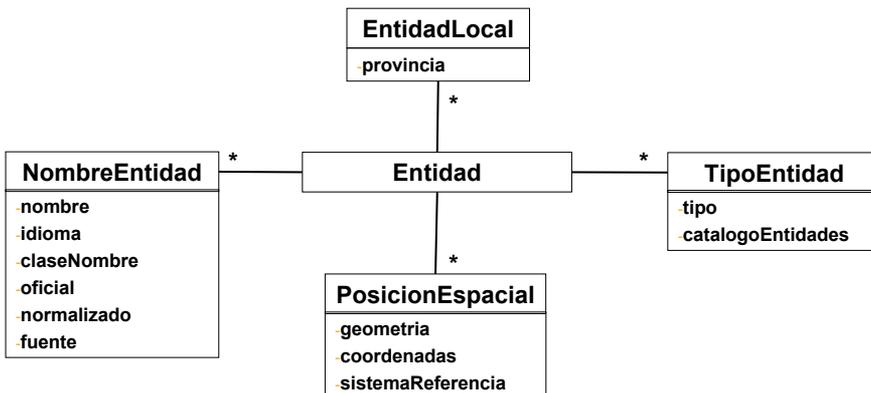
# Nomenclátor. MNE

- Modelo de Nomenclátor de España (MNE):



# Nomenclátor. MNE

- Modelo de Nomenclátor de España (MNE):
  - Detalle del modelo de datos

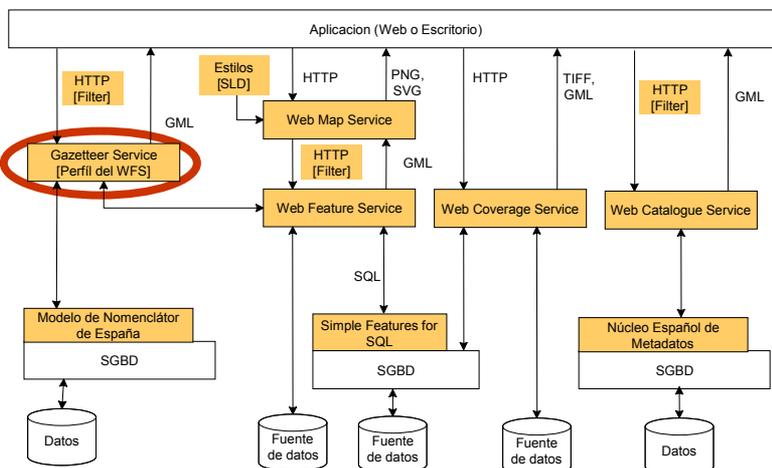


# Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- **Representación y consulta de topónimos [WFS-G]**
  - Representación de topónimos [MNE]
  - **Servicio de nomenclátor**
- Componentes para la implementación de una IDE
- Herramientas cliente para una IDE

# Arquitectura y componentes

- Contenido del curso





## Nomenclator. WFS-G

- La especificación aún no es pública
- Define un servicio web para consultar un nomenclator
- Define un modelo básico de nomenclator (con menos atributos que el MNE)
- Proporciona las siguientes operaciones:
  - GetCapabilities: describe el servicio
  - DescribeFeature: permite conocer la estructura de un tipo de entidad
  - GetFeature: permite realizar una consulta al servicio de nomenclátor



## Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- **Componentes para la implementación de una IDE**
- Herramientas cliente para una IDE



## Componentes

- Existe una gran cantidad de software libre y comercial para la implantación de una IDE
- Realizar una descripción detallada de cada componente software está fuera de los objetivos de este curso
- Realizaremos una breve descripción de cada componente siguiendo este esquema:
  - Almacenamiento de los datos
  - Servicios web
  - Matriz servicios / software



## Componentes

- Almacenamiento de los datos
  - La base de la IDE son los datos almacenados
  - Los datos pueden estar en muchos formatos diferentes
    - Formatos de herramientas CAD (DWG, DXF, DGN)
    - Formatos de herramientas SIG (Shapefile)
    - Formatos raster (GeoTIFF, ECW)
    - Sistemas gestores de bases de datos
  - Los servicios WFS y WCS van a servir de intermediario proporcionando un método de acceso común
  - Además, estos servicios van a permitir integrar información de diferentes unidades funcionales de la organización



## Componentes

- Almacenamiento de los datos
  - Con respecto a los sistemas gestores de bases de datos, hay muchas alternativas
    - Software comercial: Oracle spatial, IBM DB2
    - Software libre: PostgreSQL, MySQL
  - PostgreSQL
    - Es más robusto y posee más funcionalidad
    - PostGIS implementa de forma mucho más completa la especificación de OpenGIS
  - MySQL
    - Es más rápido y está mucho más extendido
  - ¿Cuál elegir? Depende de si necesitamos eficiencia o funcionalidad



## Componentes

- Servicios web
  - Hay dos tipos. En función del tipo van a necesitar un servidor web distinto
    - Basados en Java Server Pages (JSP). Necesitan un contenedor de aplicaciones web JSP
      - Apache Tomcat (<http://tomcat.apache.org/>)
      - Jetty (<http://www.mortbay.org/>)
      - Jboss (<http://www.jboss.org/products/jbossas>)
    - Basados en (Common Gateway Interface) CGI. Necesitan un servidor web tradicional
      - Apache Web Server (<http://httpd.apache.org/>)
      - Internet Information Services (<http://www.iis.net/>) (Software comercial. Microsoft)

# Componentes

- Servicios web
  - Basados en Java Server Pages
    - Geoserver (<http://docs.codehaus.org/display/GEOS/Home>)
      - Es muy fácil de configurar
      - Gran base de desarrolladores
      - Es el menos rápido y el menos robusto
    - Deegree (<http://www.deegree.org/>)
      - Es rápido y robusto
      - Es muy complicado de configurar
      - Es el esfuerzo de un único equipo de trabajo
  - Basados en CGI
    - MapServer (<http://mapserver.gis.umn.edu/>)
      - Es el más rápido
      - Es difícil de configurar, y sobre todo, de poner a funcionar (desarrollado para Linux)

# Componentes

- Matriz servicios / software

	WFS	WFS-T	WCS	WMS	SLD	CSW	WFS-G
Geoserver	Sí	Sí	Sí	Sí	Sí	No	No
Deegree	Sí	Sí	Sí	Sí	Sí	Sí	Sí
MapServer	Sí	No	Sí	Sí	Sí	No	No



---

## Contenidos

- Infraestructuras de datos espaciales
- Recuperación de objetos geográficos [WFS]
- Recuperación de atributos del espacio [WCS]
- Creación de cartografía [WMS]
- Representación y consulta de metadatos [CS-W]
- Representación y consulta de topónimos [WFS-G]
- Componentes para la implementación de una IDE
- **Herramientas cliente para una IDE**



---

## Herramientas cliente

- Para utilizar una IDE se necesitan herramientas que soporten la parte cliente de los servicios de la IDE
- Existen una gran variedad de herramientas, tanto en software libre como en software propietario
- Vamos a describir brevemente tres
  - gvSIG
  - uDIG
  - openJUMP



## Herramientas cliente

- gvSIG (<http://www.gvsig.gva.es/>)
  - Financiado por la Generalitat Valenciana
  - Ventajas
    - Internacionalización
    - Cliente de un gran número de estándares IDE
    - Grupo de desarrollo con mucha inercia
  - Inconvenientes
    - Muy complejo a la hora de extender
    - Documentación escasa



## Herramientas cliente

- uDIG (<http://udig.refractor.net/confluence/display/UDIG/Home>)
  - Desarrollado por el mismo grupo que PostGIS
  - Ventajas
    - Potencia y experiencia del grupo desarrollador
    - Facilidad de extensión
    - Cliente de un gran número de estándares IDE
  - Inconvenientes
    - Desarrollo extranjero
    - Voluminoso

## Herramientas cliente

- openJUMP (<http://openjump.org>)
  - Basado en JUMP (Desarrollado por VividSolutions)
  - Desarrollado por un grupo heterogéneo
  - Ventajas
    - Sencillez de la implementación
  - Inconvenientes
    - Es el menos completo de los tres
    - No soporta sistemas de referencia

## Bibliografía

- Miguel R. Luaces (2007). *Apuntes de la asignatura Bases de Datos 3*. Universidad de A Coruña.  
<http://lbd.udc.es/docencia/bd3>
- European Petroleum Survey Group.  
<http://www.epsg.org/>
- Open Geospatial Consortium.  
<http://www.opengeospatial.org/>
- Directiva INSPIRE. <http://www.ec-gis.org/inspire/>



---

## Bibliografía

- Laurini; Thompson (1999). *Fundamentals Of Spatial Information Systems*. Academic Press
- Longley; Goodchild; Maguire; Rhind (2001). *Geographic Information Systems And Science*. Wiley
- Manolopoulos; Papadopoulos; Vassilakopoulos (2005). *Spatial Databases. Technologies, Techniques and Trends*. Idea Group
- Miguel R. Luaces (2004). *A Generic Architecture For Geographic Information Systems*. Tese Doutoral. Universidade da Coruña



---

## Bibliografía

- Rigaux; Scholl; Voisard (2002). *Spatial Databases. With Application To GIS*. Academic Press
- Shekhar; Chawla (2003). *Spatial Databases. A Tour*. Pearson Education
- Taboada González; Cotos Yáñez (eds) (2005). *Sistemas de Información Medioambiental*. Netbiblo
- Worboys (2000). *GIS. A Computing Perspective*. Taylor & Francis

# Infraestructuras de datos espaciales. Conceptos básicos

**Miguel Rodríguez Luaces**

Laboratorio de Bases de Datos  
Universidade da Coruña

