



SERVEI DE SISTEMES  
D'INFORMACIÓ GEOGRÀFICA  
I TELEDETECCIÓ  
Universitat de Girona

V JORNADAS DE SIG LIBRE

# Jaspa, la mejor alternativa libre a PostGIS

José Carlos Martínez, Marta González, Eloina Coll

*Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría*

Escuela Técnica Superior de Ingeniería Cartográfica  
Universidad Politécnica De Valencia



# Descripción

- Extensión espacial para bases de datos relacionales
- Ofrece cerca de 200 procedimientos almacenados SQL con una funcionalidad similar a PostGIS 1.4.
- Sigue los estándares *Simple Features* for SQL del OGC y SQL/MM.
- Actualmente soporta los SGBD PostgreSQL y H2.

# Antecedentes

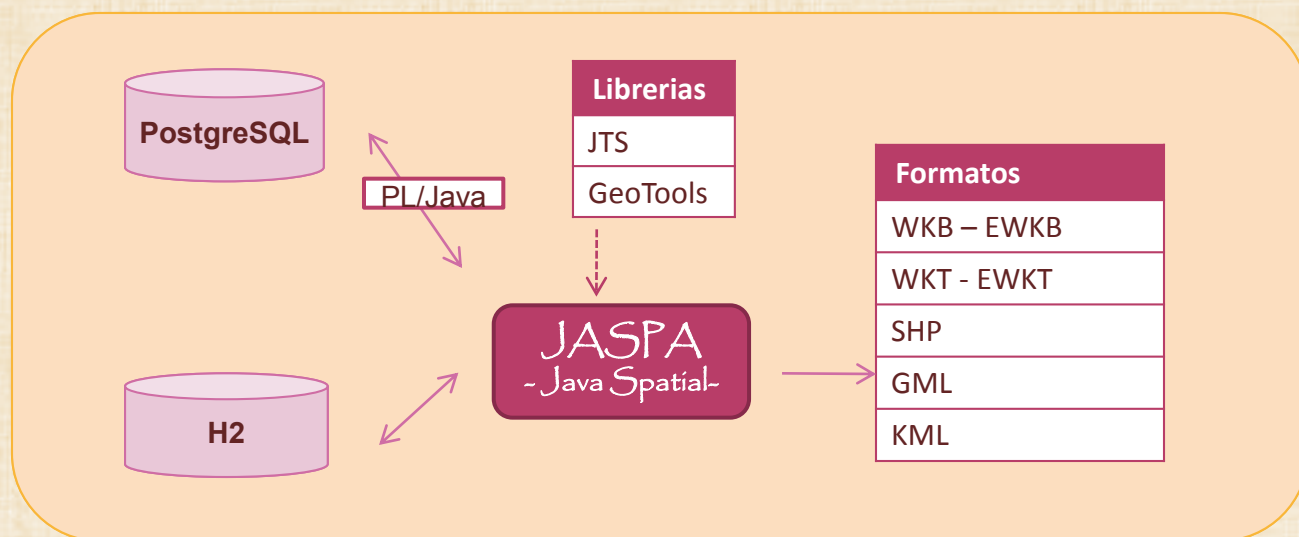
- Existen otras soluciones libres además de PostGIS como:
  - Spatial Box, H2 Spatial, Spatial Lite
  - MySQL Spatial, GeoDB
- Todas estas soluciones son soluciones parciales que adolecen de al menos uno de estos puntos si las comparamos con PostGIS:
  - Falta funcionalidad OGC/SQLMM, agregados, *arrays* de geometrías, etc.
  - No indexan espacialmente
  - No son soluciones cliente-servidor
- Por lo tanto a partir de ahora en esta presentación comparemos Jasper con PostGIS al ser la solución libre más avanzada.

# Objetivos

- Completar el puzle del software libre para información geográfica con una alternativa equiparable a PostGIS programada en Java.
- Potenciar la reutilización del código
- Producto multiplataforma que pueda utilizarse en diferentes SGBD
- Explotar las ventajas del lenguaje Java para el manejo de datos geográficos, especialmente en lo que respecta a la extensión del software por parte del usuario.

# Características generales de Jaspas

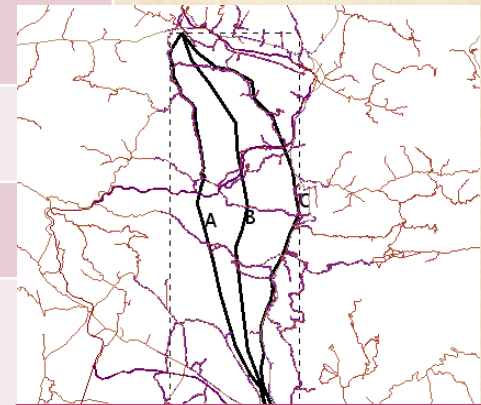
- Versión 0.1 (RC1) publicada en verano 2010
- Implementa por completo la funcionalidad de PostGIS 1.4 (funciones OGC, no OGC, *arrays*, agregados, etc.). No es un PORT de PostGIS. Implementado desde cero.
- Incorpora nuevas funciones propias (limpieza polígonos, autointersecciones, snap, etc.)





# Funcionalidad de Jaspá

Funciones	Ejemplos
Gestión	7 funciones (AddGeometryColumn, DropGeometryTable, Jaspá_Version ...)
Constructores de geometría	34 funciones (ST_GeomFromText , ST_MakeLine, ST_MakePolygon, ST_MakeGeomColl )
Acceso a geometrías	29 funciones (ST_CoordDim, ST_IsClosed, ST_IsValid ...)
Editores de geometría	27 funciones (ST_Affine, ST_Force_2D, ST_Multi, ST_NodeLine, ST_Transform ...)
Salida de geometrías	7 funciones (ST_AsBinary , ST_AsGML, ST_AsHEXEWKB, ST_AsText ...)
Relaciones espaciales y mediciones	30 funciones (ST_Centroid, ST_Contains, ST_Disjoint, ST_Distance, ST_Within ...)
Geoprocesamiento	19 funciones (ST_Buffer, ST_Collect, ST_Simplify, ST_Union
Referencia lineal	8 funciones (ST_Line_Interpolate_Point, ST_Line_Substring, ST_Locate_Along_Measure)
Otras funciones	17 funciones (ST_Accum, ST_Box2D, ST_Expand, ST_Extent, ST_Xmax, ST_XMin ...)



Posibilitan análisis espaciales complejos

- Funciones OGC, no OGC, *arrays*, agregados espaciales, etc.

# Extensión de la funcionalidad

1. Procedimientos almacenados en PL/pgSQL en un sistema PostgreSQL/Jaspa al igual que se utiliza en PostGIS.
2. Procedimientos almacenados en java en sistemas PostgreSQL/Jaspa o H2/Jaspa.
3. También se puede utilizar las funciones java de Jaspa directamente con objetos de tipo *Geometry* de JTS en lugar de bloques binarios, agilizando los procesos al no necesitar serializar y de-serializar cada vez las geometrías. Ejemplo: `ST_MinimumBoundingCircle()`.

# Algunos test (I) vs PostGIS 1.5.2

- Índices espaciales BD PostgreSQL

PostGIS	<pre>CREATE INDEX idx_suelos_geom on suelos USING gist(geom);  SELECT count(*) FROM suelos p1, suelos p2 WHERE (p1.geom &amp;&amp; p2.geom);</pre>	4600 ms
Jaspa	<pre>CREATE INDEX suelos_geom_gist ON suelos USING GIST (ST_PGBox(geom));  SELECT count(*) FROM suelos p1, suelos p2 WHERE (p1.geom &amp;&amp; p2.geom);</pre>	3900 ms

Suelos aprox. 25.000 polígonos.

625 millones de combinaciones en producto cruzado



# Algunos test (II) vs PostGIS 1.5.2

- Agregado espacial: ST\_Union

```
SELECT ST_AsText(geom) FROM (SELECT ST_Union(geom) as  
geom FROM suelos WHERE suelos.gid < 1200) AS foo;
```

BD	Tiempo
PostGIS	6150ms
Jaspa PostgreSQL	2950ms
Jaspa H2	3300ms

# Algunos test (III) vs PostGIS 1.5.2

```
SELECT ST_MinimumBoundingcircle(geom,48) FROM suelos  
WHERE suelos.gid < 10000;
```

BD	Tiempo
PostGIS (PLPGSQL)	9500 ms
JASPA4PG (PLJava)	11000 ms
JASPAPG (Jaspa stored procedure)	5900 ms
Jaspa H2 (Jaspa stored procedure)	6900 ms

# Algunos test (IV) vs PostGIS 1.5.2

## UMN Mapserver query

**Caso más desfavorable a Jaspa (conversión pljava C -> java):**

explain analyze select

```
count((asbinary(force_collection(force_2d(geom)),'NDR')))  
from suelos WHERE geom && setSRID('BOX3D(650000  
4474500,700000 4650000)')::BOX3D,  
find_srid('', 'suelos', 'geom') );
```

BD	Tiempo
PostGIS	60 ms
JASPA4PG (PLPGSQL)	200 ms
Jaspa H2	No spatial idx

# Análisis comparativo

Características	JASPA	PostGIS 1.4
Base de Datos	PostgreSQL y H2	PostgreSQL
Lenguaje de Programación	Java	C
Conectores SIG	Mapserver, Kosmo (PostgreSQL )	QGis, gvSig, GeoTools, Gdal, MapServer...
Índices espaciales	PostgreSQL: Gist H2: -	Gist
Tratamiento de geometrías vacías	NULL	GEOMETRYCOLLECTION EMPTY
Cajas de geometría (bounding boxes) tratadas como geometrías OGC	Sí	No. Utiliza tipos box2d, box3d

# Jaspa 0.2

- Disponible en abril de 2011
- Incorpora conectores para Kosmo utilizando Jaspa en PostgreSQL.
- Reorganización del código fuente. Nuevo formato interno blob para las geometrías.
- Sistema de topología de reglas: Reglas para controlar la corrección topológica de los datos (novedad en las extensiones espaciales libres).

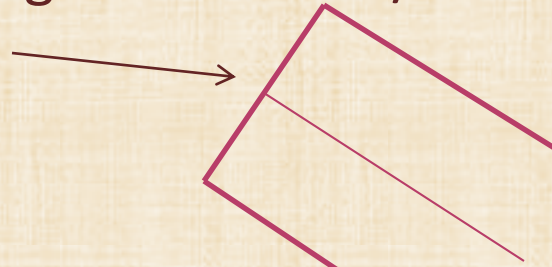


# Control de la topología

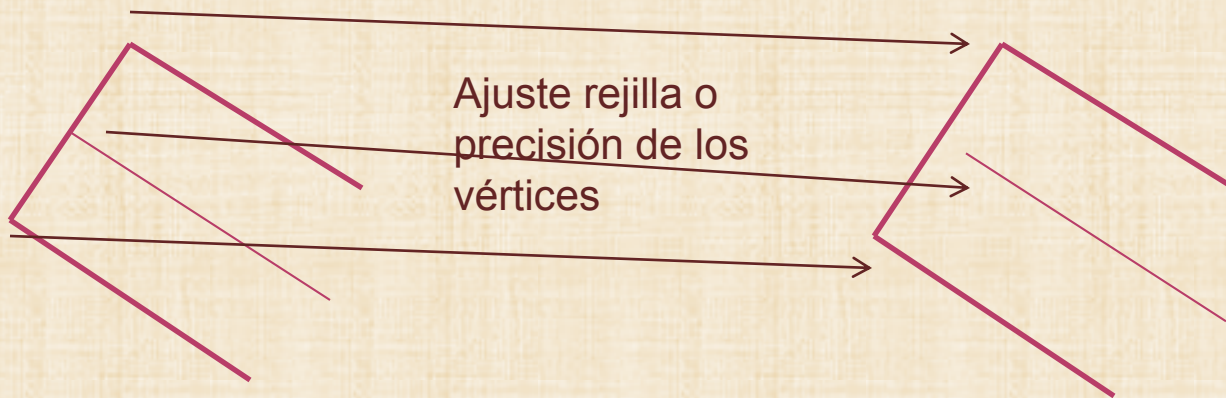
- Sistema de reglas de topología similar a ArcGIS 10
- Topología 2-D
- Basado en el modelo DE-9IM
- Implementadas en SQL espacial utilizando los propios procedimientos almacenados de Jaspá (usuario podría crear nuevas reglas de topología, de forma sencilla)
- 63 reglas en total: 28 para ser aplicadas en una capa y las otras 35 entre dos capas
- Tolerancia clúster
- Indexación espacial es vital: luego por el momento este sistema solo funciona con PostgreSQL y no en H2.

# Tolerancia / Cluster

- Problema de programas de SIG/CAD al capturar punto cercano.



- Problema de programas de SIG/CAD Desktop al capturar los vértices con una determinada precisión o ajuste a rejilla (como *MicroStation*).



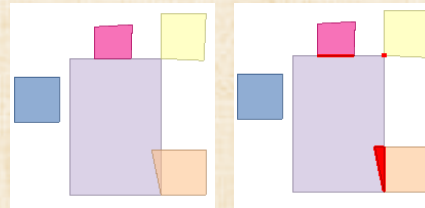
# Fundamentos del sistema topológico

El proceso de control de topología en Jaspa puede resumirse en tres pasos:

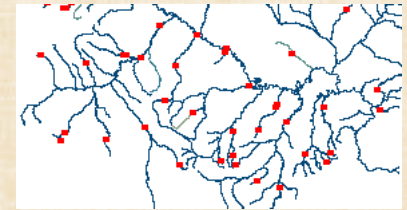
1. Creación del modelo de topología. Por cada modelo, se crean dos tablas de metadatos para almacenar las reglas y las capas.
2. Adición de capas al modelo y de las reglas que deben cumplir, especificando la tolerancia cluster para cada una de las capas.
3. Validación de las reglas: puede limitarse a una regla, a ciertas capas, o bien al modelo completo.

# Ejemplos

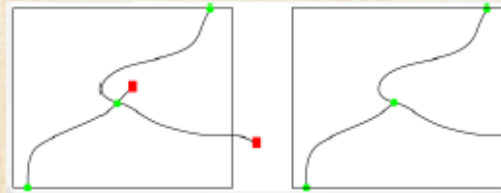
Geometry	Alias	Rule
<b>Point</b>	mbsp	Must be single Part
	mbd	Must be disjoint
	mbd_wt	Must be disjoint with tolerance
	mnbd	Must not be duplicated
	mnbd_wt	Must not be duplicated with tolerance
	mnhrp	Must not have repeated points
<b>Point + Point</b>	mbdw	Must be disjoint with
	mbdw_wt	Must be disjoint with, with tolerance
	mbcw	Must be coincident with
	mbcw_wt	Must be coincident with, with tolerance
<b>Point + Line</b>	mbi	Must be inside
	mbpi	Must be properly inside
<b>Point + Polygon</b>	mbi	Must be inside
	mbpi	Must be properly inside
	mbcbbo	Must be covered by boundary of
<b>Line</b>	mbsp	Must be single Part
	mnsi	Must not self intersect
	mnioti	Must not intersect or touch interior
	mnti	Must not touch interior
	mni	Must not intersect
	mbd	Must be disjoint
	mbd_wt	Must be disjoint, with tolerance
	mnbd	Must not be duplicated
	mno	Must not overlap
	mnhd	Must not have dangles
	mnhd_wt	Must not have dangles, with tolerance
	mnhp	Must not have pseudonodes
	mnhp_wt	Must not have pseudonodes, with tolerance
	mnhrp	Must not have repeated points



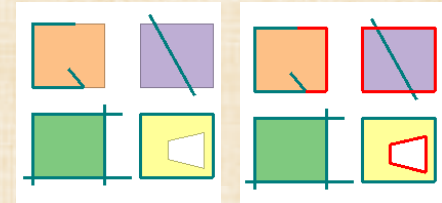
Must be disjoint



Must not have pseudo nodes



Must not have dangles



Boundary must be covered by layer



# Ventajas del sistema de reglas topológicas

- Una BD puede albergar diversos esquemas de topología.
- Posibilidad de establecer restricciones entre capas de multigeometrías.
- Validación solo de aquellas geometrías que han cambiado desde la última vez que se validó
- Las reglas pueden ser aplicadas a subselecciones de una capa que cumplan una condición.
- Control de errores: si el ajuste de tolerancia o los cálculos geométricos provocan errores (como generar geometrías no válidas) el proceso no se interrumpe y se da información precisa sobre la causa y las geometrías afectadas.
- Independencia del *frontend* gráfico utilizado



# Líneas futuras de desarrollo

- Funciones geodésicas
- Driver H2 para conectarse con SIG (Kosmo / gvSIG). Otros drivers: gvSIG, QGIS. Actualmente en la versión 0.2, existe driver para PostgreSQL/kosmo y PostgreSQL/UMN Mapserver.
- Soporte de Oracle y Hsqldb
- Conjunto de herramientas de ayuda para el análisis espacial al usuario.
- Topología híbrida (experimental)
- Almacenamiento Raster (experimental)

# Conclusiones

- En un año y medio se ha conseguido desarrollar un producto con similar funcionalidad que PostGIS en lenguaje Java, y se han implementado nuevas funciones espaciales específicas de Jaspa.
- Al estar programado en Java, puede incorporar las nuevas funcionalidades de la librería JTS/GeoTools de forma rápida.
- Se ha obtenido un producto fácil de extender con nuevas funcionalidades incluso para usuarios de SIG (que generalmente no son expertos en ciencias de la computación)
- Se está trabajando para lograr la portabilidad de Jaspa en otros SGBD como Oracle y HSQLDB, siendo actualmente posible con PostgreSQL y H2.
- La versión 0.2 incorpora un sistema avanzado de reglas de topología y tolerancia de coordenadas que es una novedad importante en bases de datos espaciales.



SERVEI DE SISTEMES  
D'INFORMACIÓ GEOGRÀFICA  
I TELEDETECCIÓ  
Universitat de Girona

V JORNADAS DE SIG LIBRE

# Gracias por la atención

Más información (descarga, manuales y talleres):

<http://forge.osor.eu/projects/jaspa/>

(Versión 0.2 disponible en abril 2011)

**Equipo Jaspa:**

José Carlos Martínez [jomarlla@cgf.upv.es](mailto:jomarlla@cgf.upv.es)

Marta González [margonal@aaa.upv.es](mailto:margonal@aaa.upv.es)

