

Drupal & GIS

Convirtiendo Drupal en un GeoCMS

DrupalCamp Spain Octubre 2011, Sevilla

Mapa de contenidos

1. ¿Qué es un GeoCMS?
2. ¿Es Drupal un GeoCMS?
3. El mundo Geo en Drupal
4. Repaso de algunos módulos
5. Gestión de la información geográfica
6. Las bases de datos espaciales
7. Las posibilidades que abren los GIS

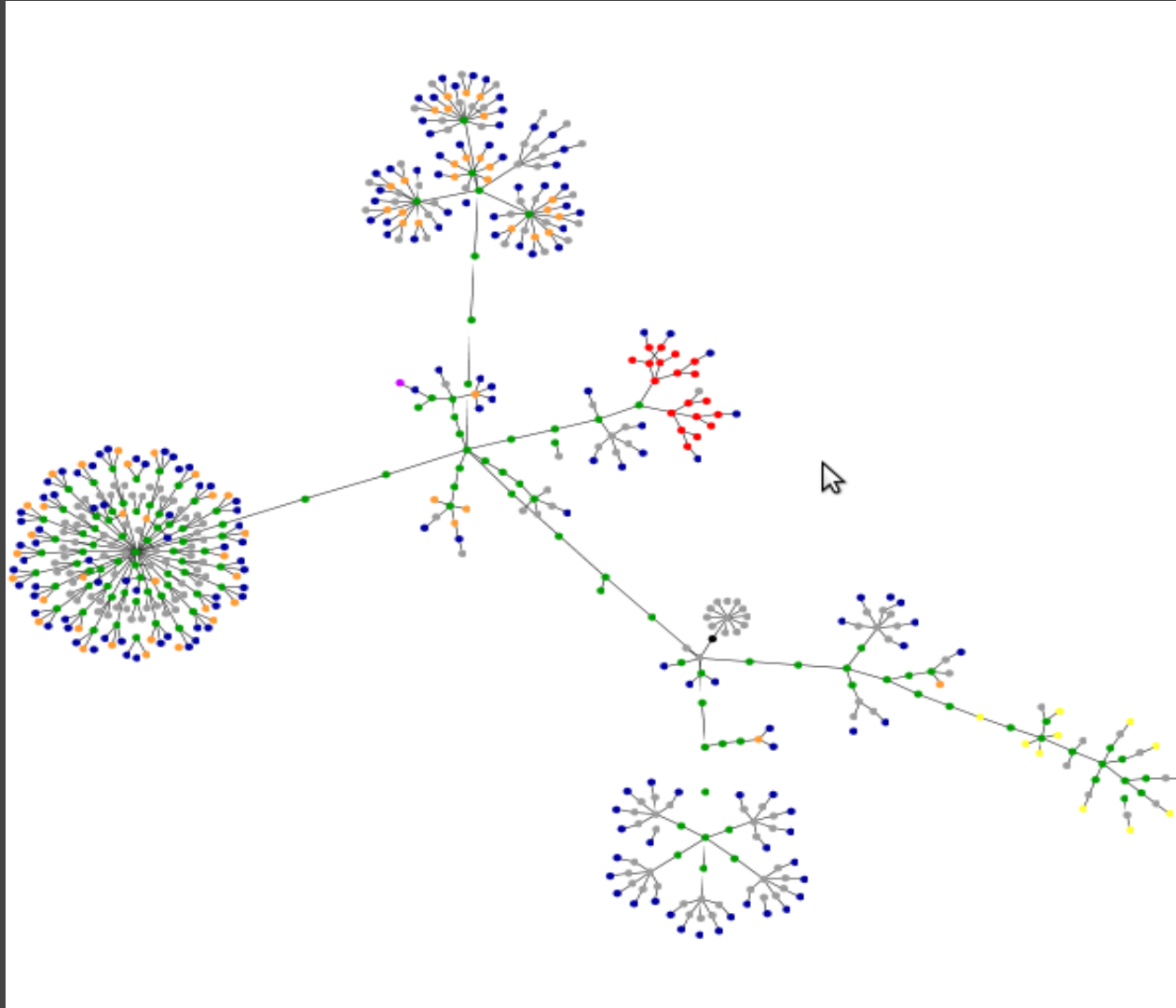
¿Qué es un GeoCMS?

Gestor de contenidos en el que la información está georreferenciada

La información geográfica es útil y se puede utilizar para establecer relaciones entre los contenidos

Es posible integrar fuentes externas de información geográfica que pueden aportar un valor añadido

¿Qué es un GeoCMS?



¿Qué es un GeoCMS?



¿Es Drupal un GeoCMS?

NO



¿Es Drupal un GeoCMS?

NO

Pero con matices...



¿Es Drupal un GeoCMS?

- No tiene soporte nativo para la georreferenciación
- No se pueden establecer relaciones espaciales entre los nodos
- No se puede integrar información geográfica de fuentes externas

¿Es Drupal un GeoCMS?

- Existen varios módulos que permiten georreferenciación
- No se pueden establecer relaciones espaciales entre los nodos
- No se puede integrar información geográfica de fuentes externas

¿Es Drupal un GeoCMS?

- Existen varios módulos que permiten georreferenciación
- Los nodos se pueden visualizar y relacionar por su posición geográfica
- No se puede integrar información geográfica de fuentes externas

¿Es Drupal un GeoCMS?

- Existen varios módulos que permiten georreferenciación
- Los nodos se pueden visualizar y relacionar por su posición geográfica
- Aparecen iniciativas para integrar fuentes de información geográfica

El mundo Geo en la comunidad

Campo con bastante movimiento

- 140 proyectos maduros y 192 en total con el tag Location
- Diferentes "familias"
 - Visualización de datos
 - Georreferenciación
 - Geolocalización de usuarios

El problema de la organización

- Múltiples iniciativas en busca de objetivos similares
- Cada *site* tiene unos niveles de exigencia diferentes
- ...cada maestrillo tiene su librillo...

Embebiendo mapas externos

- Sólo se busca meter un mapa dentro del contenido
- No se almacena ningún tipo de información geográfica
- Embed Google Maps Field, OpenLayers Filter, Mapstraction CCK Inline, Nodemap...

La familia Location & GMap

- Muy extendidos
- Cubren las necesidades básicas
- Limitado a puntos
- Las operaciones que se pueden realizar con la información almacenada es escasa
- Su uso está limitado a Google Maps

La familia OpenLayers

- Muy potente
- Se aprovecha de la gran flexibilidad que ofrece la librería
- Módulos auxiliares hacen que resulte muy útil, como por ejemplo OpenLayers Proximity o MapBox
- Procesa datos en formato geográfico básico
- El procesado de la información recae en Drupal

Otras opciones

- Módulos que tratan de ofrecer APIs comunes para diferentes clientes de mapas
 - Mapstraction/Mapstraction CCK
 - Mapping
- Módulos que tratan de integrar servicios externos para visualizarlos en Drupal
 - Mappingkit
- Utilización de servicios externos para dar valor añadido geoespacial
 - OpenCalais
 - Geonames

Procesando información geográfica

- Se trata de obtener información que almacenar por diferentes medios
 - Campos Lat/Lon
 - Direcciones
 - Mapas interactivos
 - Fotografías
- Para luego poder mostrarla asociada a los contenidos o poder asociarlos entre ellos
 - Widgets
 - Views

El procesamiento de la información geográfica

Los módulos actuales permiten almacenar esta información de diferentes maneras

- Coordenadas:
 - 40, -3
- Cadenas WKT (Well Known Text)
 - POINT(-3, 40)

GEOMETRYCOLLECTION(POINT(4 6),LINESTRING(4 6,7 10))

- Información binaria en campos espaciales (Well Known Binary)
 - 010100000000000000000000F03F000000000000F03F
(POINT(1,1))

Texto vs Binario

- El **texto** es legible
- No es necesario utilizar extensiones espaciales
- Es fácil de manipular

- Hay que crear queries complejas para hacer búsquedas
- En ocasiones hay que extraer datos y procesarlos en PHP (Geofield)
- No se puede manejar con proyecciones

Texto vs Binario

- El **binario** no se entiende
- Hay que utilizar extensiones espaciales
- No se puede manipular directamente

- Se pueden realizar operaciones geométricas directamente en BD
- Las operaciones pueden devolver objetos WKT
- Puede manejar diferentes proyecciones

Ejemplo de búsqueda espacial

Búsqueda Circular desde un punto:

- OpenLayers proximity:

- ```
SELECT node.nid AS nid, openlayers_proximity_filter_circle.distance AS
openlayers_proximity_filter_circle_distance, node.title AS node_title, node_data_field_ol_map.
field_ol_map_openlayers_wkt AS node_data_field_ol_map_field_ol_map_openlayers_wkt,
node_data_field_ol_map.delta AS node_data_field_ol_map_delta, node.type AS node_type,
node.vid AS node_vid FROM node node RIGHT JOIN openlayers_proximity
openlayers_proximity ON node.nid = openlayers_proximity.nid RIGHT JOIN (SELECT nid, lat,
lon, (6371.0 * ACOS(SIN((lat * RADIANS(1))) * SIN((41.6529434 * RADIANS(1))) + COS((lat *
RADIANS(1))) * COS((41.6529434 * RADIANS(1))) * COS((lon * RADIANS(1)) - (-4.7283811 *
RADIANS(1))))) AS distance FROM openlayers_proximity) openlayers_proximity_filter_circle
ON node.nid = openlayers_proximity_filter_circle.nid LEFT JOIN content_field_ol_map
node_data_field_ol_map ON node.vid = node_data_field_ol_map.vid WHERE (node.type in
('ol_geocoder')) AND ((openlayers_proximity_filter_circle.distance <= 20.000000)) GROUP BY
nid, openlayers_proximity_filter_circle_distance, node_title,
node_data_field_ol_map_field_ol_map_openlayers_wkt, node_data_field_ol_map_delta,
node_type, node_vid
```

# Ejemplo de búsqueda espacial

## Búsqueda Circular desde un punto:

- OpenLayers proximity:

- SELECT node.nid AS nid, openlayers\_proximity\_filter\_circle.distance AS

```
SELECT nid, lat, lon, (6371.0 * ACOS(SIN((lat * RADIANS(1))) * SIN((41.6529434 * RADIANS(1))) + COS((lat * RADIANS(1))) * COS((41.6529434 * RADIANS(1))) * COS((lon * RADIANS(1)) - (-4.7283811 * RADIANS(1))))) AS distance FROM openlayers_proximity) openlayers_proximity_filter_circle ON node.nid = openlayers_proximity_filter_circle.nid LEFT JOIN content_field_ol_map_node_data_field_ol_map ON node.vid = node_data_field_ol_map.vid WHERE (node.type in ('ol_geocoder')) AND ((openlayers_proximity_filter_circle.distance <= 20.000000)
```

node\_type, node\_vid



# Ejemplo de búsqueda espacial

## Búsqueda Circular desde un punto:

- PostGIS:

- SELECT state, zip  
FROM zipcode  
WHERE  
distance(  
transform(PointFromText('POINT(-116.768347 33.911404)', 4269),32661),geom)  
< 16093

# Ejemplo de búsqueda espacial

```
SELECT nid, lat, lon, (6371.0 * ACOS(SIN((lat * RADIANS(1))) * SIN((41.6529434 * RADIANS(1))) + COS((lat * RADIANS(1))) * COS((41.6529434 * RADIANS(1))) * COS((lon * RADIANS(1)) - (-4.7283811 * RADIANS(1))))) AS distance FROM openlayers_proximity openlayers_proximity_filter_circle ON node.nid = openlayers_proximity_filter_circle.nid LEFT JOIN content_field_ol_map node_data_field_ol_map ON node.vid = node_data_field_ol_map.vid WHERE (node.type in ('ol_geocoder')) AND ((openlayers_proximity_filter_circle.distance <= 20.000000)
```

VS

```
SELECT state, zip FROM zipcode WHERE distance(transform(PointFromText('POINT (-116.768347 33.911404)', 4269),32661),geom) < 16093
```

# Ejemplo de búsqueda espacial

```
SELECT nid, lat, lon, (6371 *
RADIANS(1))) + COS((lon -
* RADIANS(1)) - (-4.72813)) *
openlayers_proximity_filter_cir
LEFT JOIN content_filter_circle
node_data_field_openlayers_proxi
((openlayers_proximity_filter_cir
```

```
SELECT state, zip FROM
(-116.768347 33.91140)
```



```
W((41.6529434 *
DIANS(1))) * COS((lon -
openlayers_proximity)
nity_filter_circle.nid
e.vid =
AND
```

```
ntFromText('POINT
```

# Otras búsquedas espaciales



# Otras búsquedas espaciales



```
SELECT i.name, i.lat, i.lon
FROM items i, autop p
WHERE p.id = 66 AND
ST_DWithin(i.the_geom, p.the_geom,5000)
```

# El problema de la escalabilidad

Los clientes de mapas tienen una capacidad limitada

Al querer mostrar muchas geometrías se puede ver mermado su rendimiento

El procesamiento de muchas geometrías desde Drupal también puede hacer que el rendimiento caiga en picado.

¿Hemos encontrado un límite?

¿Cuáles son las posibles vías de escape?

# El problema de la escalabilidad

Los clientes de mapas tienen una capacidad limitada

Al querer mostrar muchas geometrías se puede ver mermado su rendimiento

El procesamiento de muchas geometrías desde Drupal también puede hacer que el rendimiento caiga en picado.

¿Habríamos encontrado un límite?  
¿Cuáles son las posibles vías de escape?

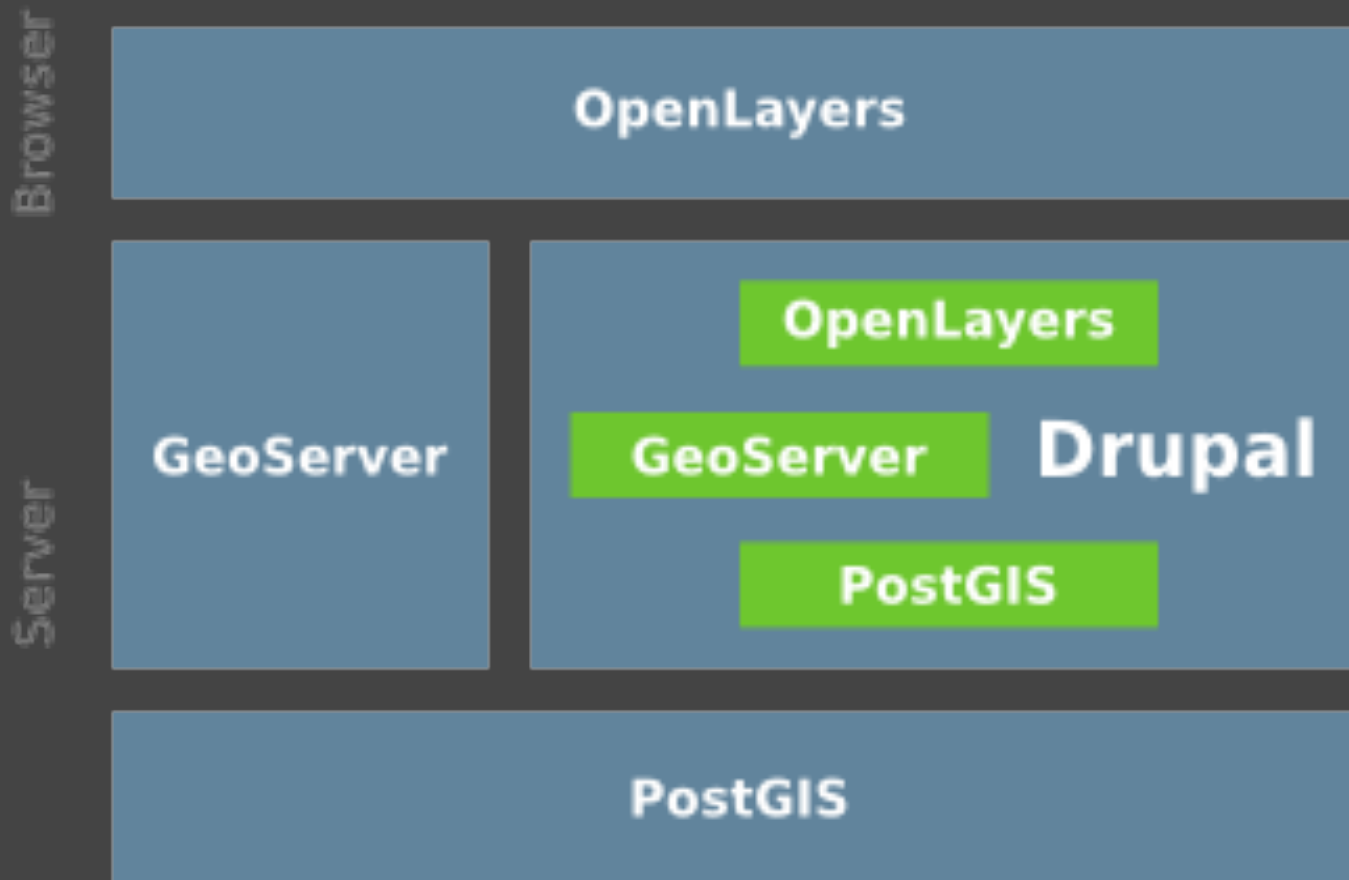
# GIS

# La integración de servicios GIS en Drupal

- Los GIS (Sistemas de Información Geográfica) son elementos que están orientados a procesar información geográfica
- Integrarlos con Drupal puede ser una vía para solventar los problemas que veíamos anteriormente
- Pueden liberar a Drupal de tareas costosas y llevarlas a cabo de forma más rápida



# La integración de servicios GIS en Drupal



# Nuevos caminos que se abren

La introducción de herramientas GIS dentro de Drupal ofrece un nuevo abanico de posibilidades, pero también nuevos retos

- Necesidad de familiarizar a los usuarios con los GIS
- Estandarización de los accesos
- Creación de familias de módulos capaces de integrar de forma sencilla las nuevas funcionalidades
- Comprobar que realmente el rendimiento de estas nuevas aproximaciones es tan bueno como se espera

# Tratemos que se fusionen en uno solo

- Seguir todos el mismo camino es la mejor manera de llegar a la meta
- Tomar parte en el debate dentro de la comunidad y tratar de tomar decisiones conjuntas

# Tratemos que se fusionen en uno solo

- Seguir todos el mismo camino es la mejor manera de llegar a la meta
- Tomar parte en el debate dentro de la comunidad y tratar de tomar decisiones conjuntas

Contribuir es la mejor manera de conseguir nuestros objetivos y conseguir que todos podamos beneficiarnos de ello

# Recursos

- <http://drupal.org>
- <http://groups.drupal.org/location-and-mapping>
- <http://idelab.uva.es>
- <http://openlayers.org>
- [https://github.com/developmentseed/openlayers\\_plus](https://github.com/developmentseed/openlayers_plus)
- <http://www.slideshare.net/ajturner/drupal-and-the-geospatial-web>  
<http://www.levelos.com/blog/2010/08/mapping-drupal>
- <http://www.geops.de/blog/64-spatial-data-and-drupal-7>

Muchas gracias

¡Nos encontramos en la web!

plopesc@gmail.com  
@plopesc