**Práctica Modulo 4.2. Introducción a las bases de datos espaciales: postgresql /postgis**

En esta práctica se va a utilizar el gestor de libre distribución **Postgres** que dispone de una extensión denominada **PostGIS** para el manejo de información espacial.

**PostgreSQL/PostGIS** es una base de datos gratuita soportada por el OpenGis Consortium que sigue dicho estándar y el estándar SQL-MM común a todas las bases de datos espaciales.

**Archivos utilizados:** Andalucia.shp, shx……

**Metodología de desarrollo de la práctica**

1. Crear una base de datos espacial vacía

2. Insertar datos en una base de datos espacial desde un fichero .shp

3. Consultas a la base de datos espacial:

* Selección por distancia
* Selección por distancia, teniendo en cuenta el índex espacial.
* Selección por punto-en-Bounding BOX
* Selección por punto-en-polígono.
* Selección por polígono.
* Selección por polígono y solape de Bounding\_BOX.
* Selección por área.
* Selección por proximidad (buffer).
* Función ST\_Touches que selecciona los polígonos que tocan a uno dado.

4. Acceso mediante gvSIG

5. Conexión con un WMS

**1. Crear una base de datos espacial (vacía).**

* Arranque **psql** con la base de datos por defecto con uno de los procedimientos empleados en el punto anterior.
* Teclee:

**CREATE DATABASE andalucia;** (no olvide “;” al final)

para crear una nueva base de datos de nombre “andalucia”. Dicha base de datos tendrá como usuario “postgres” y la clave de acceso de administración.

* Salga de PostgreSQL con **\q**
* Vuelva a entrar en PostgreSQL, pero en la BBDD de “andalucia”

**psql andalucia postgres**

* Ahora, hay que hacer algunos cambios para que la base de datos “andalucia” acepte la extensión PostGIS.

En primer lugar, instalar la extensión del lenguaje sql. Para ello, teclee:

**CREATE LANGUAGE plpgsql;**

Normalmente, responderá que ya está instalado.

* Salga de psql con **\q**.
* A continuación, hay que crear un par de tablas: “geometry\_columns”, para almacenar un conjunto de datos sobre las tablas de geometría espacial que se creen, y “spatial\_ref\_sys”, que contiene los identificadores de los sistemas de referencia y sus datos (SRID’s). Para ello, teclee en la ventana de comandos:

**cd C:\Archivos de programa\PostgreSQL\9.6\share\contrib\postgis-2.4**

**psql –d andalucia –f postgis.sql “postgres” (pedirá password)**

**psql –d andalucia –f spatial\_ref\_sys.sql “postgres” (pedirá password)**

* Para verificar que el proceso de creación fue realizado correctamente, conecte de nuevo con el servidor tecleando

**psql andalucia “postgres”**

* Teclee el comando

**\d**

Podrá comprobar que se han creado las tablas antes reseñadas. La primera de ellas estará vacía, pero la segunda tiene datos.

* Teclee los siguientes comandos para comprobarlo:

**\d spatial\_ref\_sys**

**SELECT postgis\_full\_version();**

**SELECT \* FROM spatial\_ref\_sys WHERE srid=23030;**

El primero de ellos, nos muestra las columnas de la tabla “spatial\_ref\_sys”. El último el contenido de las columnas de la fila correspondiente al SRID 23030 (ED50/UTM zona 30N, normativo en España).

**2. Insertar datos en una base de datos espacial desde un fichero shp.**

Vamos a cargar ahora una base de datos desde un fichero shp. Existe un programa denominado **shp2pgsql** que prepara un script SQL para crear las tablas necesarias e insertar los contenidos adecuadamente.

* Localice los archivos Andalucia.shp, Andalucia.shx, etc en la carpeta Datos del Moodle. Desde el directorio en que haya descargado los ficheros. Para que se creen los datos y tablas en la BBDD de nombre “andalucia”, teclee:

**shp2pgsql –s 23030 .\Andalucia\Andalucia.shp Andalucia > andalucia.sql**

Se creará un fichero con comandos SQL denominado “andalucia.sql”. Si quiere, puede verlo con un editor, aunque es muy grande.

* Para ejecutar dicho script en la base de datos “andalucia”, teclee:

**psql –f andalucia.sql –d andalucia “postgres”**

e introduzca la clave de acceso. El script crea una tabla denominada igualmente “andalucía” que contiene la información espacial y los atributos del fichero shape.

* Verifique el contenido de la base de datos, tecleando

**psql andalucia “postgres”**

y ejecutando los comandos **\d** y **\d andalucia**. Note que la tabla contiene las columnas “nombre”, “provincia”, “gid” y “codigo”. La información espacial está contenida en una columna denominada “the\_geom” (podríamos cambiarle el nombre si quisiéramos) cuyo tipo es MULTIPOLYGON, ya que los municipios pueden estar compuestos de varios polígonos (¡En Andalucía es muy frecuente!).

* Como la Base de Datos ya es bastante grande, interesa introducir una estructura adicional de indexado (por defecto tiene un *b-tree*, que es válido para los atributos no espaciales). Cree un índice espacial mediante el comando:

**CREATE INDEX andalucia\_bbox\_index ON andalucia USING gist (geom);**

El método **gist** crea un R-TREE mejorado, similar al explicado en teoría.

* Como es natural, el árbol es una jerarquía de rectángulos envolventes. Compruebe las dimensiones del rectángulo envolvente del municipio ‘BUJALANCE’ mediante el comando:

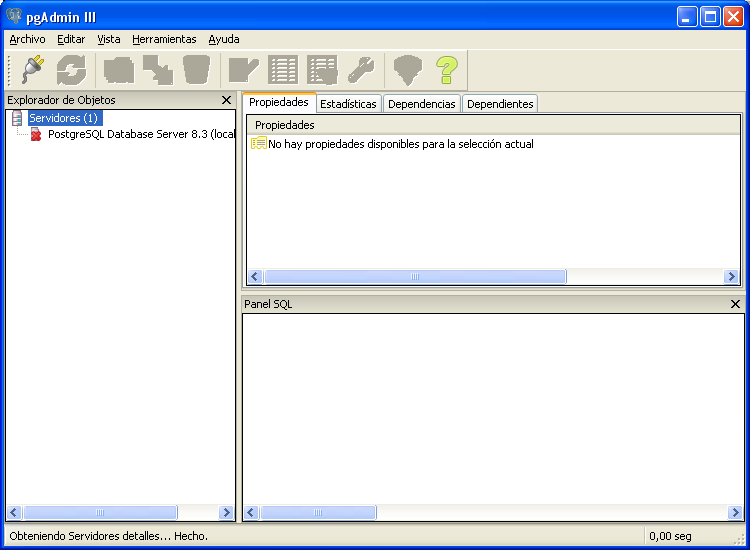
**SELECT nombre, ST\_extent(geom) FROM andalucia WHERE nombre=’BUJALANCE’ GROUP BY nombre;**

El empleo de la clausula “GROUP BY” es obligatorio al tratarse de un tipo de geometría MULTIPOLYGON.

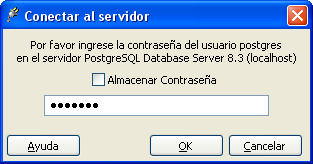
**3. Consultas a la base de datos espacial**

La extensión PostGIS de PostgreSQL permite hacer numerosas operaciones con los datos espaciales almacenados siguiendo los estándares OpenGIS y SQL-MM (derivado del anterior). Para comprobar algunas de ellas utilizaremos la herramienta PgAdminIII, más cómoda que el programa psql.

Para arrancar, seleccione “Inicio” > “Todos los programas” >”PosgreSQL 9.6” > “PgAdmin III”

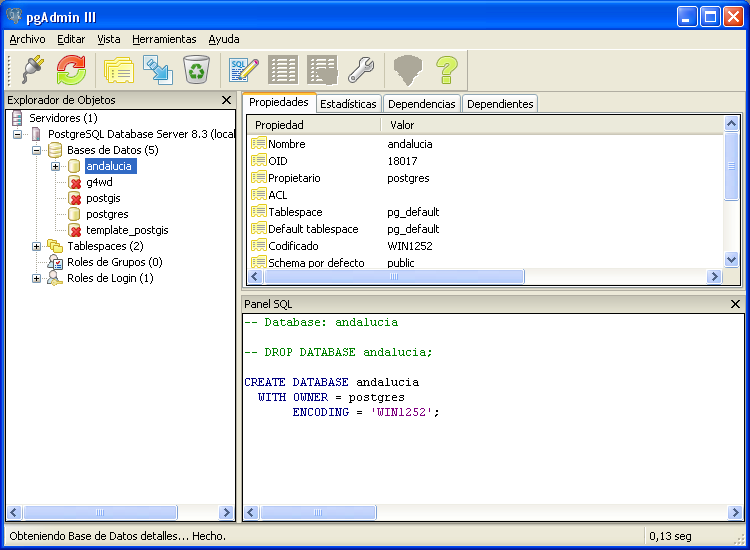


* Haga doble clik sobre “PostgreSQL Data Base Server 9.6 (localhost)”

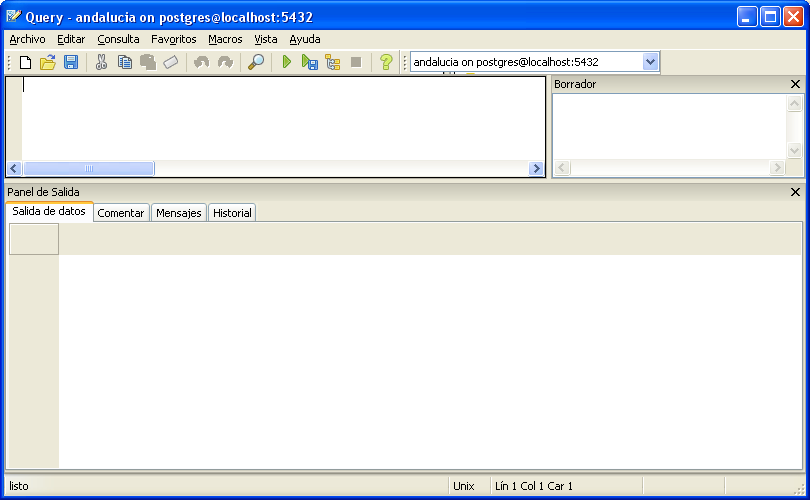


Introduzca su clave de acceso.

* Seleccione la Base de Datos “andalucia”



* Elija en el menú el icono correspondiente a “Ejecutar consultas SQL arbitrarias” y aparecerá una nueva ventana:



* En el cuadro de texto superior podrá teclear los comandos SQL. Los resultados se mostrarán en el panel de salida después de pulsar “Consulta”>”Ejecutar”.

**Nota**: ¡Cuidado! Si copia y pega los comandos que aparecen a continuación, corrija las comillas simples tecleándolas de nuevo, pues Word corrigió automáticamente dichas comillas y las cambió por las tildes correspondientes a “comilla abierta” y “comilla cerrada”.

**Consultas:**

* **1. Selección por distancia.** Búsqueda de los polígonos situados a una distancia < 1000 metros del punto de coordenadas (285090, 3977061) en el SRID 23030.

**SELECT nombre FROM andalucia WHERE ST\_Distance(geom, ST\_GeomFromText(‘POINT(285090 3977061)’, 23030)) < 1000 ;**

* **2. Selección por distancia, teniendo en cuenta el índex espacial.** En este ejemplo, se da una ventana de búsqueda (BOX3D) de 2000x2000 metros para seleccionar previamente los polígonos cuyo rectángulo envolvente interseca con ella (operador &&). La selección por distancia se hace con los resultantes.

**SELECT nombre FROM andalucia WHERE geom && ST\_setSRID(‘BOX3D(284090 3976061, 285090 3978061)’:: box3d, 23030) AND ST\_Distance(geom, ST\_GeomFromText(‘POINT(285090 3977061)’, 23030)) < 1000 ;**

* **3. Selección por punto-en-Bounding BOX.** Este ejemplo, es una alternativa que selecciona aquellos polígonos cuyo rectángulo evolvente contiene al punto dado.

**SELECT nombre FROM andalucia WHERE ST\_GeomFromText(‘POINT(285090 3977061)’, 23030) && geom;**

* **4. Selección por punto-en-polígono**. Ésta es una alternativa más precisa pues, además, se exige que el punto esté dentro del polígono.

**SELECT nombre FROM andalucia WHERE ST\_GeomFromText(‘POINT(285090 3977061)’, 23030) && geom**

**AND ST\_within(ST\_GeomFromText(‘POINT(285090 3977061)’, 23030), geom);**

* **5. Selección por polígono.** En este caso, se emplea un polígono en lugar de un punto y se seleccionan los polígonos de la base de datos que lo cortan :

**SELECT nombre FROM andalucia WHERE ST\_Intersects(geom, ST\_GeomFromText(‘POLYGON((285000 3976000, 286000 3976000, 286000 3976500, 285500 3977000, 285000 3976500, 285000 3976000))’, 23030));**

* **6. Selección por polígono y solape de Bounding\_BOX.** Este ejemplo es una simplificación del anterior en el que se seleccionan los polígonos cuyos rectángulos envolventes cortan al polígono dado.

**SELECT nombre FROM andalucia WHERE geom && ST\_GeomFromText(‘POLYGON((285000 3976000, 286000 3976000, 286000 3976500, 285500 3977000, 285000 3976500, 285000 3976000))’, 23030);**

* **7. Selección por área.** Se seleccionan los polígonos de menos de 500 Ha de superficie.

**SELECT nombre, ST\_Area(geom) / 10000 as hectareas FROM andalucia WHERE ST\_Area(geom) / 10000 < 500;**

* **8. Selección por proximidad (buffer).** En este ejemplo se pretende hallar los municipios (o polígonos) limítrofes con el municipio de Málaga. Para ello, se buscan todos los polígonos que cortan a otro trazado a partir del polígono del municipio “MÁLAGA” creciendo alrededor 0.5 metros (buffer).

**SELECT nombre FROM andalucia WHERE ST\_Intersects(geom, ST\_Buffer((SELECT geom FROM andalucia WHERE nombre='MÁLAGA'),0.5));**

* **9. Existe otra función, denominada ST\_Touches que selecciona los polígonos que tocan a uno dado.** Sin embargo, no funciona correctamente en el caso de “Málaga”, como podrá comprobar más tarde con GvSIG. Quizá se deba a que consta de dos polígonos disjuntos o por algún otro problema con los datos de entrada. Puede probar esta función con “Antequera”, que sólo tiene un polígono.

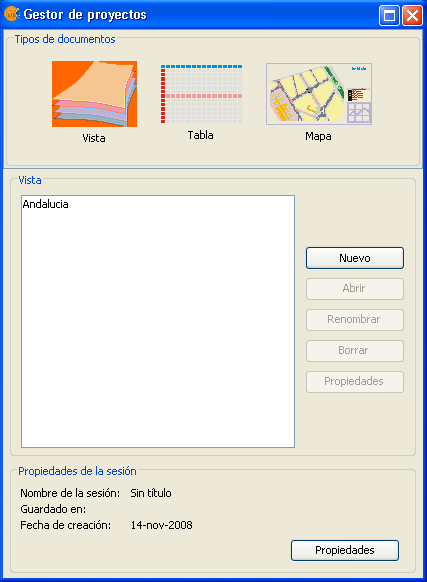
**SELECT nombre FROM andalucia WHERE ST\_Touches(geom, (SELECT geom FROM andalucia WHERE nombre='ANTEQUERA'));**

Si desea saber más, consulte el manual PostGIS 1.3.3 Documentation.

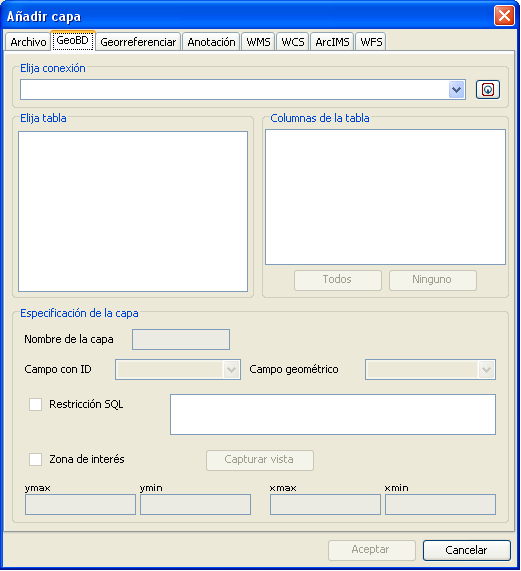
**4. Acceso a la Base de Datos PostgreSQL/PostGIS mediante gvSIG**

En este apartado se va a acceder a la Base de Datos de Andalucía desde **gvSIG**.

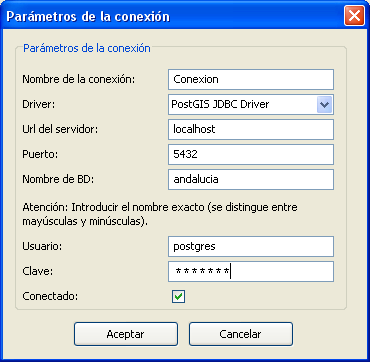
* Cree un nuevo proyecto desde la ventana “Gestor de proyectos” y llámelo “Andalucia”



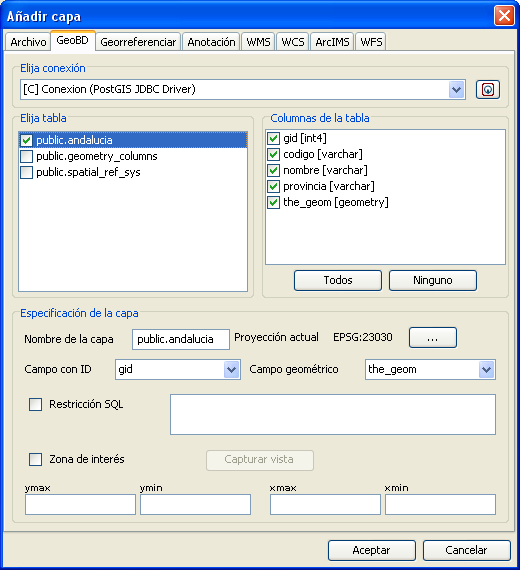
* Desde la Vista Andalucía, elija “Añadir capa” y seleccione la pestaña GeoBD



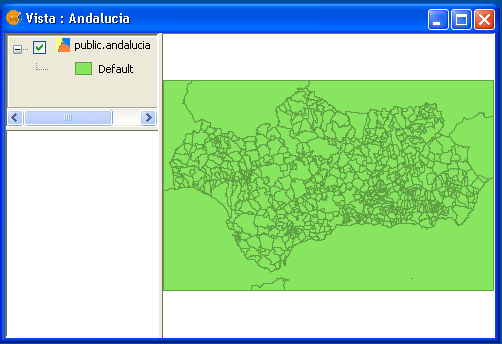
* Pulse el botón situado a la derecha del cuadro de texto “Elija conexión”. Rellene los campos de la ventana emergente como se muestra en la figura y pulse “Aceptar”.



* Si todo ha ido bien, se mostrarán en la ventana “Añadir capa” las tablas existentes en la Base de Datos “andalucia”.
* Seleccionar la tabla denominada “andalucia”. Pulsar “Aceptar”

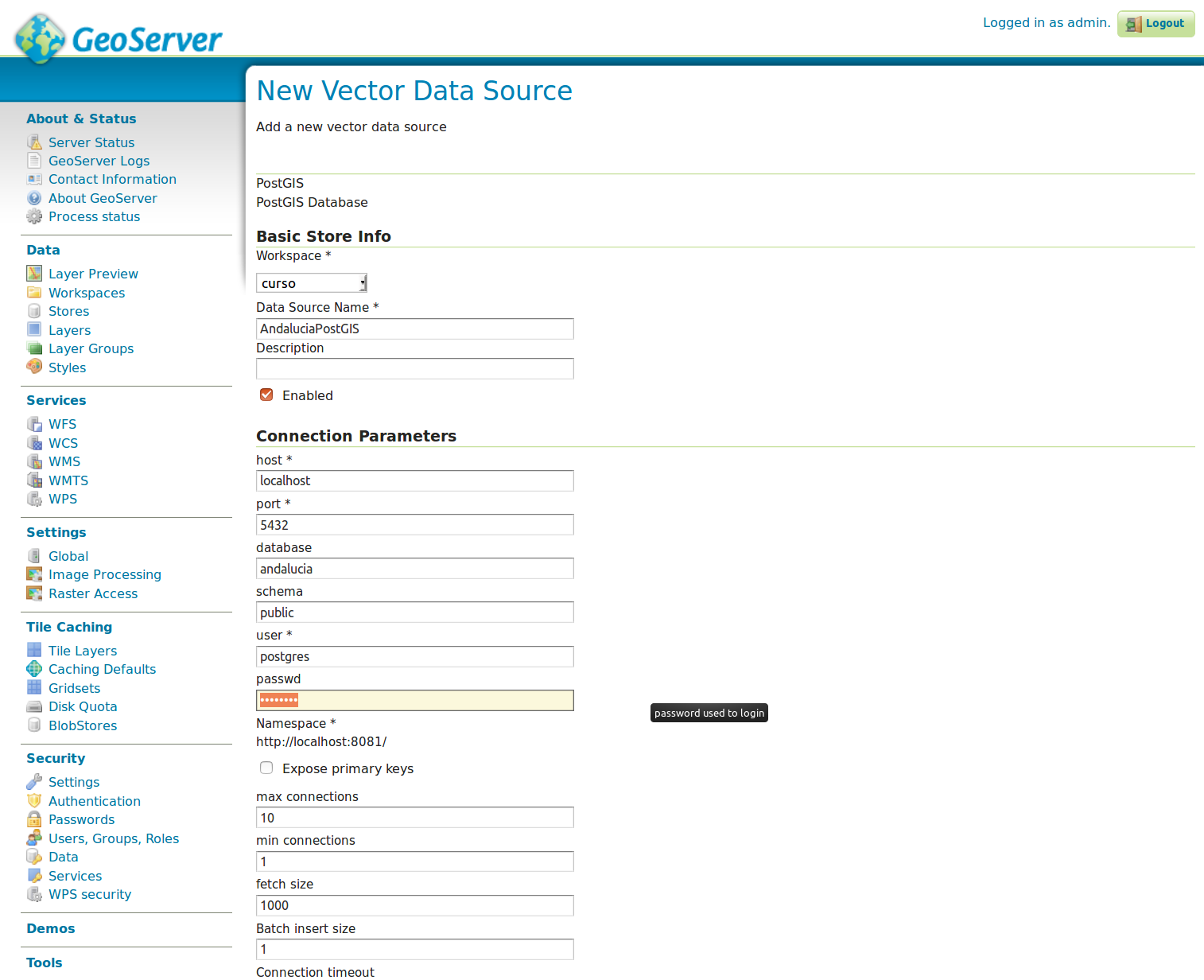


* Aparecerá en la vista la capa cargada

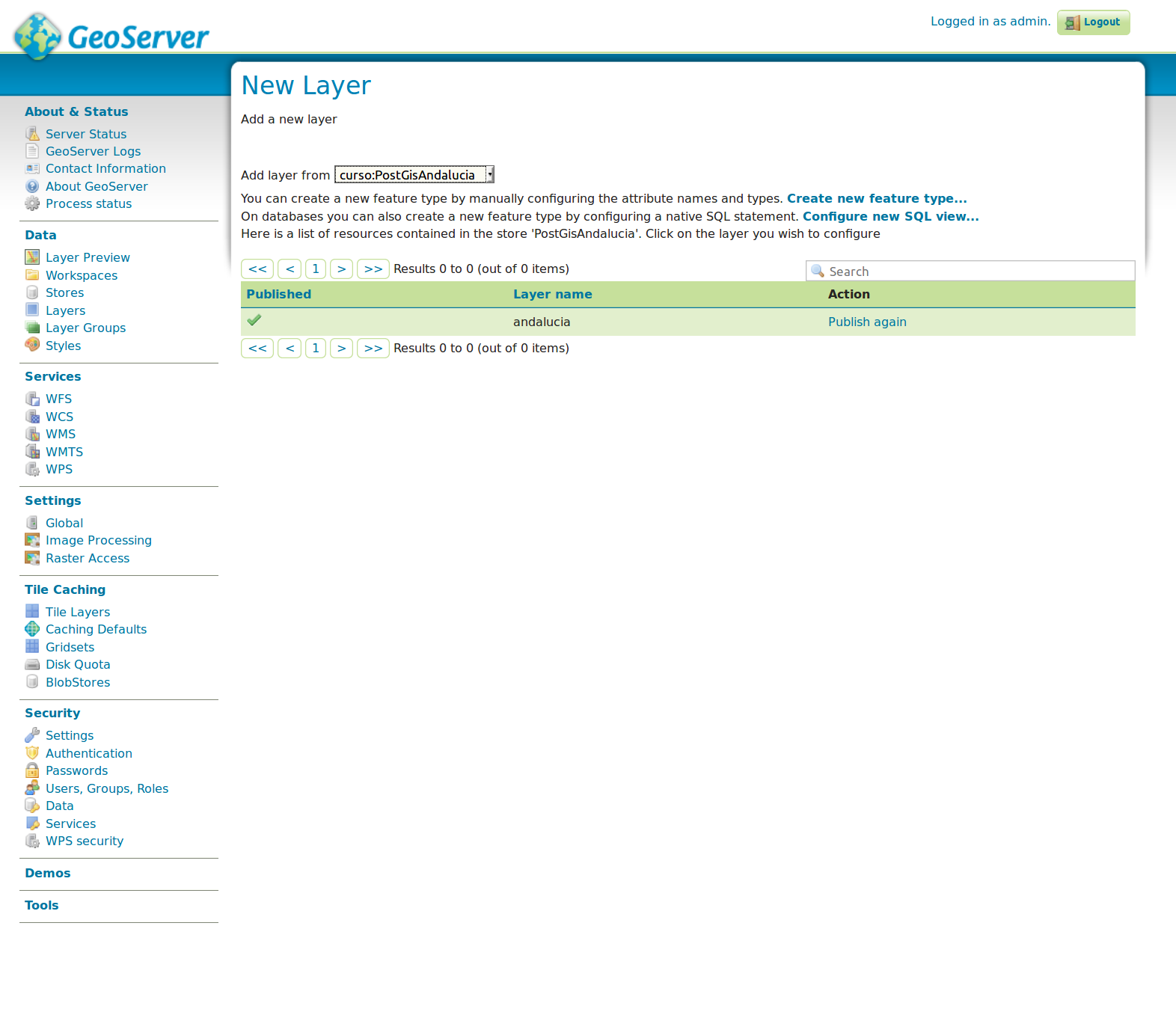


**5. Conexión a un WMS**

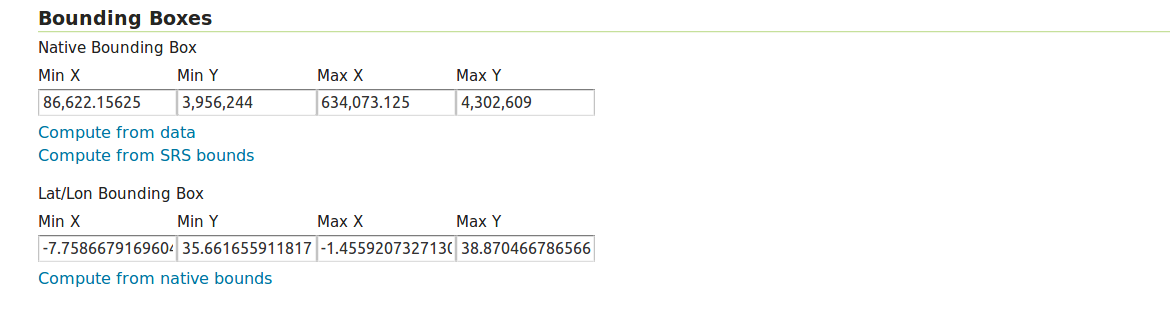
Para conectar desde GeoServer con una BBDD PostGIS será necesario crear un **nuevo Almacén de Datos** (Store). Ver práctica 4.1 para detalles. El tipo de Almacén será **PostGIS - PostGIS Database.** El siguiente paso será introducir los datos de conexión con la BBDD “andalucia” que ya hemos creado en nuestro PostGIS.



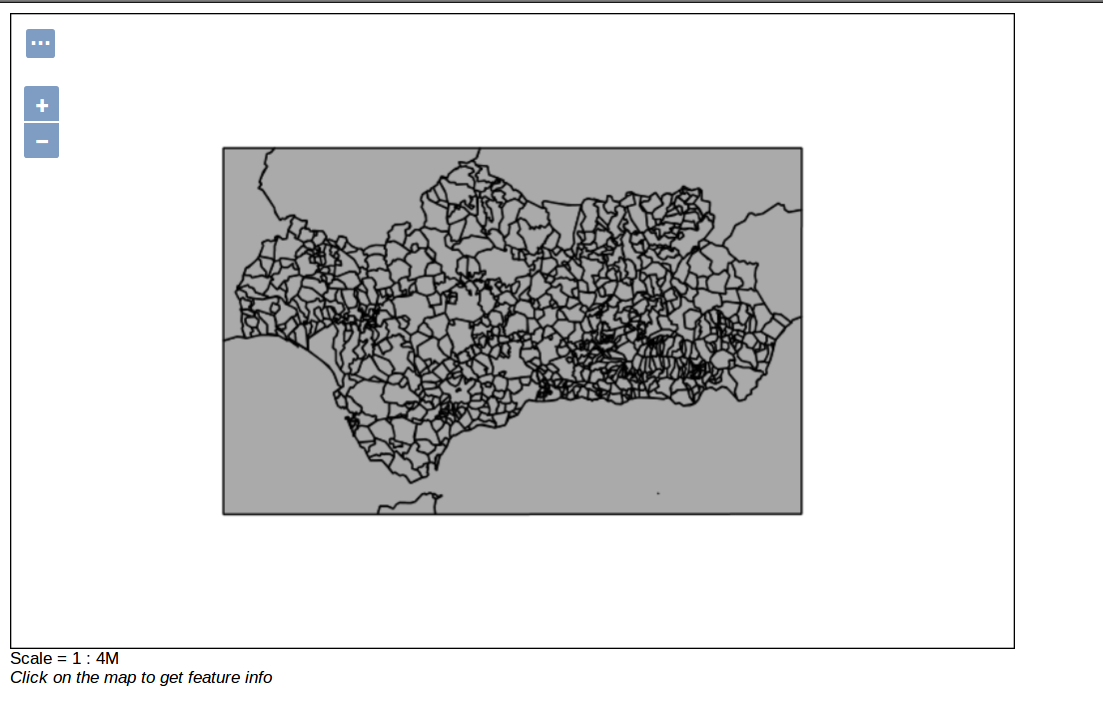
Ahora tenemos que crear una **nueva capa** seleccionando el almacén que hemos creado en el punto anterior.

****

Seleccionamos **Publicar.** En lapertaña “Datos”, establecemos los límites a visualizar con “**Compute from data**” y “**Compute from native Bounds**” y guardamos los datos de publicación.

****

Ahora previsualizaremos la capa con OpenLayer accediendo a “**Previsualización de Capas**”, buscamos la capa en el listado y seleccionamos “”**OpenLayers** para que muestre la capa. Deberíamos conseguir algo como esto:



Recordad que, al igual que en la práctica 4.1, también se puede usar gvSIG para acceder a la información publicada en el WMS.