

Práctica 2: Configuración básica de una base de datos espacial y un motor de servicios web geográficos

Laboratorio de Ingeniería del Software

Grado en Ingeniería Informática

Especialidad en Ingeniería del Software

Rubén Béjar

Universidad de Zaragoza

Escuela de Ingeniería y Arquitectura

Departamento de Informática e Ingeniería de Sistemas

Área de Lenguajes y Sistemas Informáticos

1 Objetivos

En esta práctica os familiarizaréis con la base de datos espacial PostgreSQL/PostGIS, y con el motor de servicios web geográficos Geoserver.

2 Aspectos prácticos

2.1 Máquina virtual

Tenéis a vuestra disposición una máquina virtual Linux (Xubuntu 12.04) con PostgreSQL (incluyendo la extensión PostGIS), Tomcat 7 y Geoserver ya instalados. Es la misma que se distribuyó para la práctica 1, si ya la tenéis no hace falta volver a descargarla. **Esta máquina virtual podríais usarla para el despliegue del proyecto si queréis.**

3 Tareas

3.1 Configurar y probar la instalación de la base de datos espacial

Obligatoria

La máquina virtual tiene instalado un gestor de bases de datos con soporte para datos geográficos (base de datos espacial). Concretamente es PostgreSQL 9.1 con la extensión espacial PostGIS 2.0. El proceso de instalación es muy dependiente de la plataforma y de las versiones (tanto la versión de la plataforma como la de la base de datos); si os interesa saber más, podéis empezar mirando en <http://postgis.net/>, pero la máquina virtual que tenéis ya lo lleva todo instalado.

Durante el proceso de instalación se crea un servicio de Linux llamado `postgresql` que es el servidor de base de datos. Si en algún momento tenéis algún problema y queréis probar a relanzar el servicio, estos son los comandos para lanzar (aunque se lanza automáticamente al arrancar la máquina virtual), parar y relanzar el servicio a través del shell:

```
$sudo service postgresql [start|stop|restart]
```

Para consultar el estado de los servicios podéis usar el siguiente comando (si `postgresql` sale con un (+) en la lista de servicios es que está lanzado).

```
$service --status-all
```

También tenéis instalada una aplicación gráfica para la administración de la base de datos llamada “pgAdmin III” (versión 1.14). Lanzadla. Veréis que tenéis creada una conexión con el servidor de base de datos en `localhost:5432` llamada `geospatialdbserver`. Conectaos a ella. Como esa conexión se hace con el usuario `postgres` (cuya contraseña es `postvm`) tenéis pleno acceso a todo en el gestor de bases de datos.

3.2 Configuración de la B.D. espacial para el proyecto

Obligatoria

Primero vais a crear un **usuario** (un *Login Role* en terminología de PostgreSQL) para el

proyecto¹:

Role name: proyecto

Password: proyectovm

Role privileges: "Inherits rigths from parent roles" y "Can create database objects".

Todo lo demás lo dejáis por defecto.

Ahora vais a crear un **tablespace** para que se almacenen allí los datos. Primero necesitaréis crear el directorio (cread /home/labis/proyecto/tablespace) y tenéis que cambiarle tanto el propietario como el grupo a postgres (\$sudo chown postgres:postgres /home/labis/proyecto/tablespace). Luego desde pgadmin cread un nuevo Tablespace:

Name: TS_PROYECTO

Owner: proyecto

Location: /home/labis/proyecto/tablespace

Todo lo demás lo dejáis por defecto.

El siguiente paso es crear una **base de datos**. Creadla en pgadmin:

Name: proyectodb

Owner: proyecto

Tablespace: TS_PROYECTO

Todo lo demás lo dejáis por defecto.

Luego cread un **esquema** dentro de la base de datos que acabáis de crear:

Name: proyecto

Owner: proyecto

Todo lo demás lo dejáis por defecto.

La base de datos recién creada debe ser **habilitada para que soporte datos y consultas espaciales**. Desde el menú de contexto de Extensions en la bd en pgadmin elegid New Extension... y allí añadid la extensión de nombre postgis al esquema proyecto (sin tocar nada más). Luego volvéis a hacer lo mismo pero añadiendo la extensión postgis_topology (va ligada a la anterior, no hace falta especificar de nuevo el esquema).

En este punto hay un problema. Si vais al esquema proyecto que acabáis de crear, veréis que hay una tabla spatial_ref_sys. Esta tabla se ha creado automáticamente

¹ No es buena idea usar el usuario administrador para todo. Por otra parte, en una instalación realista seguramente tendríais más roles con distintos niveles de privilegios

al añadir la extensión `postgis`. Fijaos que tiene como propietario a `postgres` (no nos ha dado la opción de elegirlo al instalarla) y si vais a sus propiedades veréis (pestaña `Privileges`) que nuestro usuario `proyecto` no tiene ningún privilegio sobre esta tabla. Esto va a impedirnos trabajar con este usuario con datos espaciales porque como mínimo necesitamos poder leer cosas de esta tabla (y también acceder a las funciones, vistas, etc. que se han añadido cuando hemos instalado la extensión `postgis` y que también tiene como propietario a `postgres`). Lo que podemos hacer es darle privilegios al usuario `proyecto` sobre todos estos objetos. Podríamos hacerlo uno por uno, pero son muchos (hay casi 900 funciones), así que probaremos otra cosa:

- Haced click derecho en el esquema `proyecto` y elegid `Grant Wizard...` seleccionad todos los objetos y en la pestaña `Privileges` seleccionad todos los privilegios y el rol `proyecto`, click en `Add/Change` y luego click en `Aceptar`. Con esto le daríamos privilegios plenos a nuestro rol sobre todos los objetos del esquema y solucionaríamos el problema.

Aquí **podemos encontrarnos con otro problema**, y es que el desplegable de los roles no nos muestre el rol `proyecto`. La solución es la siguiente: haced lo mismo que antes con uno de los roles que nos aparezca (`public`) **pero no hagáis click en Aceptar**. Abrid la pestaña `SQL` y seleccionad y copiad todo el código que aparece en ella **y luego hacéis click en Cancelar**. Después, os aseguráis de que el esquema `proyecto` esté seleccionado y pulsáis el botón `Execute arbitrary SQL queries` de la barra de herramientas principal. En el editor `SQL` que aparece pegáis el código y buscáis y sustituís el texto `TO public` por el texto `TO proyecto`. Cuando lo tengáis, ejecutad las sentencias y comprobad que no os dan errores. Si ahora volvéis a consultar las propiedades de la tabla `spatial_ref_sys`, por ejemplo, veréis como `proyecto` ahora tiene privilegios.

3.3 Prueba de la B.D. espacial para el proyecto

Obligatoria

Vais a comprobar que todo funciona, que se pueden añadir datos espaciales y que se pueden hacer consultas SQL utilizando los operadores espaciales que proporciona PostGIS.

Aunque PostGIS nos proporciona herramientas de línea de comandos para cargar datos espaciales en la base de datos, y que también se pueden crear directamente en SQL, vamos a utilizar QGIS para crear una tabla con datos geográficos procedentes de un fichero SHP. Lanzad QGIS y elegid “Añadir capas PostGIS”, creáis una nueva conexión:

Nombre: `proyecto`

Servidor: `localhost`

Base de datos: `proyectodb`

Nombre de usuario: `proyecto`

Contraseña: `proyectovm`

En este punto hacéis click en `Probar conexión`. Si da fallo, algo no está bien en la configuración, tendréis que repasarla. Si da éxito, marcad `Guardar nombre de usuario`, `Guardar contraseña`² y `Listar también tablas sin geometría`.

² Esta opción es insegura y os avisará de ello. Pero para el caso actual os ahorrará introducir la contraseña cuando os

Todo lo demás lo dejáis por defecto.

Hacéis click en **Conectar** (como de momento no hay datos, no nos añadirá ninguna tabla a la lista pero si que os debe mostrar el esquema proyecto) y luego en **Cerrar**. Ahora vais al menú principal **Base de datos > Administrador de BBDD > Administrador de BBDD**, seleccionáis el **esquema proyecto** dentro de la conexión proyecto e importáis **capa/archivo**. En **Entrada** localizad el fichero SHP con las comarcas de Aragón, en **Tabla** escribid el nombre que queráis para la tabla (p.ej. **comarcas_aragon**, mejor todo minúsculas), aseguraos de que el **esquema** es **proyecto**, marcad **Crear nueva tabla** y **Crear índice espacial** (esto es **absolutamente crítico** si los datos son un poco grandes y queréis buenas prestaciones) y en **SRID de origen** indicáis el código EPSG de los datos (en este caso pondréis 23030). Si todo va bien os dirá que la importación ha tenido éxito y si actualizáis el esquema veréis que hay una tabla nueva llamada **comarcas_aragon**. Seleccionadla y en la pestaña **Vista preliminar** podéis verla³.

3.3.1 Consultas SQL espaciales

Opcional. Pero es importante que la hagáis si tenéis tiempo, porque es la forma de ver cómo se trabaja con la base de datos espacial

Una base de datos espacial debe permitir hacer análisis espaciales. Vais a probar algunos ejemplos con PostGIS para que os hagáis una idea de lo que se puede hacer (tendréis que estudiar la documentación de PostGIS para descubrir todas sus posibilidades).

Primero de todo, cargad la capa de núcleos de población de Aragón en la base de datos usando QGIS (de esta forma tendréis dos capas: Comarcas y Núcleos de población). Llamad a la tabla **nucleos_aragon**.

Luego, desde el propio administrador de bases de datos de QGIS, seleccionad la base de datos proyecto y lanzad la **Ventana SQL**. Probad a ejecutar las siguientes consultas. El resultado lo podéis ver en la ventana SQL, o podéis cargarlo como capa nueva en el mapa (tendréis que indicar qué columna de los resultados es un identificador y qué columna es la geometría) que es lo que os recomiendo si queréis apreciar bien el resultado de las consultas espaciales.

“Comarca cuyo nombre es Zaragoza” (esta consulta no tiene nada de espacial, es para que os aseguréis de que todo os funciona y para que veáis que sigue siendo una base de datos relacional normal).

```
SELECT * FROM comarcas_aragon WHERE d_comarca = 'Zaragoza';
```

“Núcleos de población de Aragón a menos de 25 kilómetros del centro de Zaragoza”

```
SELECT * FROM nucleos_aragon WHERE ST_Distance(geom,  
ST_GeomFromText('POINT(675000 4613000)', 23030)) < 25000
```

ST_Distance es una función de PostGIS que nos permite calcular la distancia mínima entre dos geometrías.

ST_GeomFromText es una función de PostGIS que permite construir un objeto

conectéis a la base de datos espacial desde QGIS.

3 Si los datos son muy grandes es posible que no tengáis una vista previa. No pasa nada: estarán bien cargados, solo es que por prestaciones QGIS ha decidido no mostrar vista previa.

Geometry⁴ de PostGIS a partir de texto. En este caso construimos un punto a partir de sus coordenadas en EPSG:23030.

“Área total de los núcleos de población de Aragón a menos de 25 kilómetros del centro de Zaragoza”

```
SELECT SUM(ST_Area(geom))/1000000 AS km2_totales FROM
nucleos_aragon WHERE ST_Distance(geom,
ST_GeomFromText('POINT(675000 4613000)', 23030)) < 25000
```

ST_Area es una función de PostGIS que nos devuelve el área de una geometría en las unidades del sistema de referencia de esta (puesto que estamos trabajando en UTM, las unidades son metros, por eso el área nos la devolverá en metros cuadrados y tenemos que dividir por 1000000 para tener kilómetros cuadrados).

“Núcleos de población de Aragón contenidos en la comarca de Zaragoza”

```
SELECT n.id, n.geom FROM comarcas_aragon AS c, nucleos_aragon AS n
WHERE ST_Contains(c.geom, n.geom) AND c.d_comarca='Zaragoza';
```

ST_Contains es una función de PostGIS que nos devuelve cierto si la segunda geometría está totalmente contenida en la primera. Fijaos que esta consulta es un ejemplo de *spatial join* (cruzamos dos tablas por sus geometrías).

3.4 Configuración de Geoserver

Obligatoria

Geoserver es un motor de servicios geográficos en web, libre y desarrollado en Java. Podéis descubrir más cosas sobre este proyecto en (<http://geoserver.org/>). Tenéis la versión 2.4.x de Geoserver instalada y desplegada en un servidor web Tomcat 7. Para lanzar, parar y relanzar Tomcat se usan los siguientes comandos:

```
$sudo service tomcat7 [start|stop|restart]
```

El acceso a la consola de administración de Tomcat se hace a través de <http://localhost:8080/manager> (usuario:labis, password: labisvm).

Comprobad que Geoserver funciona (desde el navegador, id a <http://localhost:8080/geoserver>). Viene con algunos datos cargados por defecto: id a Data > Layer Preview y dadles un vistazo.



Si os fijáis, hay un montón de “formatos” en los que Geoserver puede darnos acceso a los datos que hemos cargado. **¿Reconocéis alguno? ¿Podéis determinar alguna de las diferencias entre ellos?**

Desde esa misma web se pueden realizar tareas de administración de Geoserver. Para ello, primero hay que conectarse (usuario: admin, password: geoserver). Lo primero que vais a hacer es publicar unos datos de PostGIS/PostgreSQL a través de Geoserver.

El primer paso será crear un *workspace* (llamadlo *proyecto*). Los *workspaces* son

⁴ PostGIS añade dos tipos de dato principales a PostgreSQL: Geometry (para geometrías planas) y Geography (para geometrías esféricas). Si trabajamos con datos geográficos proyectados (por ejemplo en UTM) querremos trabajar con Geometry. El tipo Geography queda para los datos que tengamos sin proyectar (coordenadas longitud/latitud sin más). Como al cargar los datos de comarcas y núcleos de Aragón hemos indicado que el sistema de referencia era el EPSG:23030 (UTM 30N ED50), QGIS los ha cargado como Geometry en la base de datos.

formas de agrupar capas relacionadas en Geoserver. Si hace falta concretar, podemos referirnos a una capa concreta de un workspace concreto como `workspace:capa`. Tendréis que darle un identificador en forma de URI. Puede ser cualquier URI válida, pero es interesante hacer un esfuerzo para que sea única y tenga algún sentido (por ejemplo podéis poner <http://eina.unizar.es/labis/CURSOACTUAL/eXX/prueba> donde eXX es vuestro identificador de equipo y CURSOACTUAL puede ser 1314 o el que corresponda).

Desde la interfaz de administración de Geoserver, tenéis que crear un nuevo `store` de tipo PostGIS (un `store` de Geoserver es un contenedor de datos que puede ser un simple fichero o una base de datos espacial). Creadlo dentro del `workspace` prueba, y llamadlo `proyectodb`. Especificad los parámetros de conexión a PostgreSQL (para la B.D. `proyectodb` que habéis creado hace un momento) y guardadlo.

Luego tenéis que crear una `layer` de Geoserver. Para ello añadid una nueva para el `store` recién creado y elegid `comarcas_aragon`. Generad los *bounding boxes* de la capa usando las opciones automáticas que se ofrecen (un *bounding box* es un rectángulo definido como dos pares de coordenadas que engloba la capa entera; tenerlo predefinido ayudará a Geoserver a hacer algunas operaciones más rápidamente). Guardadlo. Podéis previsualizar la capa en la administración web de Geoserver para comprobar que se ve bien.

Aunque Geoserver ha asignado un símbolo por defecto a la capa recién creada, lo normal es que queráis mejorar esa simbología. Para ello tendréis que crear un `style`. Si vais a hacerlo desde las opciones de administración de Geoserver, veréis que os pide que creéis un documento SLD (Styled Layer Descriptor). SLD es un estándar para la descripción de estilos cartográficos en XML. Podéis buscar su especificación en Internet y consultar la documentación de Geoserver para ver ejemplos. Para ejemplos sencillos, y con algo de experiencia no tan sencillos, se puede crear el SLD directamente en un editor de texto o modificando algún ejemplo anterior. Pero vamos a ver una forma más visual usando Quantum GIS:

Abrid QGIS y cargad la capa `comarcas_aragon` desde la base de datos espacial. Simbolizadla como queráis y, desde la misma ventana de creación de simbología, guardad el estilo como SLD.

Luego, en la creación de estilos de Geoserver, cargad el fichero SLD, validadlo⁵ y guardadlo, dándole el nombre que queráis y metiéndolo en el `workspace` del proyecto. En la sección de `Layers`, pestaña `Publishing`, elegid este estilo como estilo por defecto para la capa de `comarcas_aragon` y guardad. En la previsualización de la capa deberíais ver ahora el estilo gráfico que habéis creado con QGIS. **Atención:** SLD es un estándar que soporta la creación de estilos muy complejos, y QGIS tiene bastantes posibilidades para la creación de estilos. Esta combinación implica que **es posible que creéis un estilo con QGIS que luego no se vea exactamente igual en Geoserver**, o incluso que creéis algo con QGIS que no sea un SLD totalmente correcto, o que no sea un SLD que Geoserver entienda. En esos casos podéis probar a simplificar el estilo en QGIS, o bien a retocar manualmente el SLD.

3.4.1 Carga del servicio de mapas Geoserver en QuantumGIS

Obligatoria

⁵ Es más fiable si se valida, pero a veces funcionará incluso aunque la validación del SLD nos de algún fallo.

Finalmente, vais a ver cómo conectar al servicio web de Geoserver para ver los datos que habéis cargado y simbolizado. Para ello, añadid una capa de tipo WMS a QGIS (igual que en la práctica 1) pero esta vez apuntad a esta URL: <http://localhost:8080/geoserver/wms>

Una vez conectados, veréis que os sale una larga lista de capas (Geoserver viene con bastantes datos de ejemplo). Elegid las comarcas de Aragón que acabáis de cargar en Geoserver y cargadlas en el mapa. Obviamente esto no es muy interesante, puesto que tenéis acceso a esos mismos datos localmente. Lo interesante es notar que ahora se pueden ver esos datos, con el estilo gráfico que habéis configurado, desde cualquier computador conectado a Internet⁶, y desde cualquier aplicación cliente que entienda el protocolo WMS.

⁶ Bueno, dada la configuración de red de la máquina virtual, esto no será así, pero alterando esa configuración de red, podríais acceder al servicio WMS con las comarcas de Aragón desde cualquier otro equipo en Internet.