

CLASE
6

Bases de datos espaciales

República Argentina
Ministerio de Educación
Programa Nacional Mapa Educativo

Curso de capacitación
Bases de datos espaciales

CLASE 6
Material de lectura. Versión 1

Temas de esta clase:

Administración de la base de datos espaciales. Roles, grupos, privilegios.
Gestión de proyectos de gvSIG con PostGIS.
Las bases de datos espaciales y las Infraestructuras de Datos Espaciales IDE.
Trabajo final de integración.

Referencias:

A lo largo del documento encontraremos íconos y recuadros que requieren de una especial atención de los lectores:



ACTIVIDADES: son consignas de actividades para realizar la práctica con gvSIG y PGAdmin acompañando la lectura. En la presente clase hay 2 actividades para resolver y un trabajo final integrador de todo el curso.



¡IMPORTANTE! Indica una actividad que no debe omitirse para poder desarrollar correctamente la práctica de la clase.

1 Administración de la base de datos espaciales.

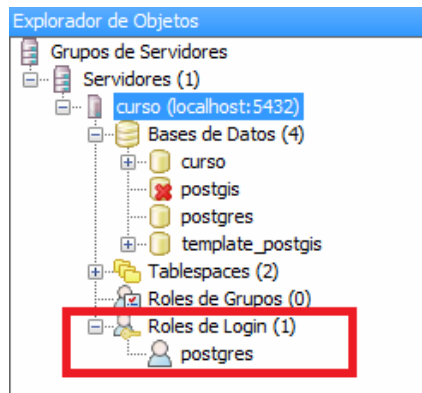
La administración de la base de datos espaciales es una tarea muy importante para resguardar la seguridad de la información, mantener los datos a salvo de errores involuntarios, establecer criterios y mecanismos de actualización, crear y dar permisos a diferentes usuarios, monitorear el rendimiento del motor, la integridad referencial, etc.

En un equipo de trabajo pequeño, seguramente serán las mismas personas que gestionan la cartografía quienes administran la base de datos espaciales. Pero lo ideal es que exista una persona asignada a esta tarea que tenga los conocimientos necesarios sobre la administración de una base de datos. Aquí veremos algunos aspectos básicos, sobre todo referidos a los distintos niveles de acceso a los datos por parte de los usuarios.

1.1 Roles

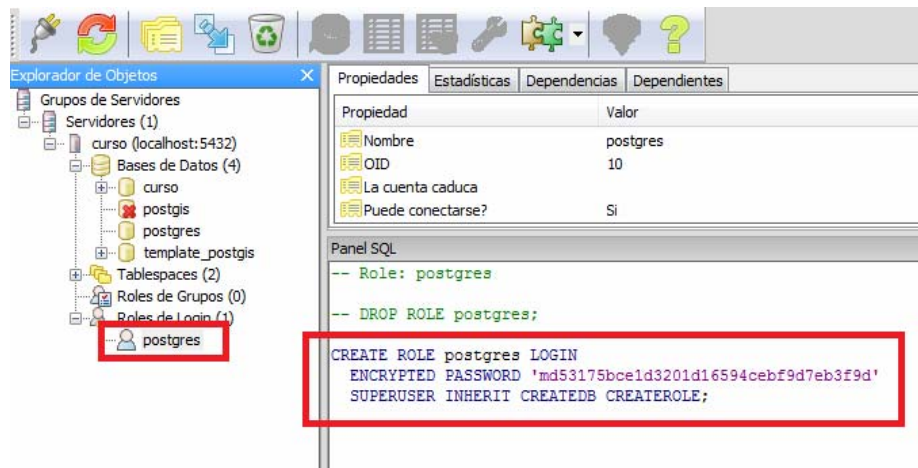
Los roles dentro de Postgres son similares a los usuarios de Windows, aunque son completamente independientes de estos. El manual de postgres, de todas maneras, recomienda utilizar los mismos nombres para simplificar el trabajo de las personas. Los usuarios pertenecen a un Cluster de base de datos, es decir a todas las bases de datos creadas a partir de una instalación del servidor Postgres; y no a una base de datos en particular.

Es por esto que dentro del árbol de objetos que vemos en PGAdmin, los roles están por fuera de todas las bases de datos.



Por ejemplo, podemos tener el rol de administrador, y tener permisos para crear, modificar y borrar bases de datos, tablas, funciones, roles, etc. Y también podemos crear roles para que otras personas dentro de la institución puedan visualizar los datos, o modificar solo una parte de ellos, o que a su vez puedan crear nuevos roles, etc. Un área que lleva adelante un SIG puede tener usuarios que solo consultan los datos, otros que actualizan solo la parte alfanumérica de los datos, y otros que corrigen las posiciones y la topología de los datos espaciales. También puede existir un usuario coordinador que necesite permisos para crear nuevos usuarios que visualicen los datos.

Para crear un nuevo usuario, accedemos al objeto "Roles de Login" en el árbol de objetos del panel izquierdo de PGAdmin. Allí vemos que, por defecto, se ha creado el usuario "postgres". Este es el usuario que hemos estado utilizando hasta ahora. Para conocer sus propiedades, que en el caso de los roles se denominan *atributos*, hacemos clic sobre "postgres" y vemos los atributos en el panel derecho.

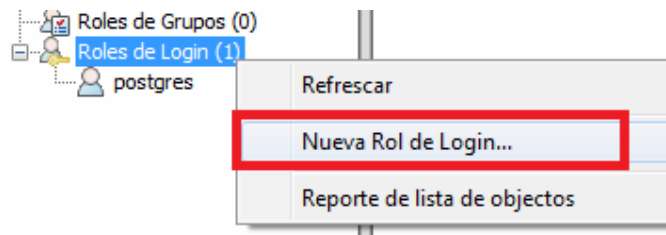


Aparecen cuatro atributos que veremos a continuación al crear un nuevo rol:

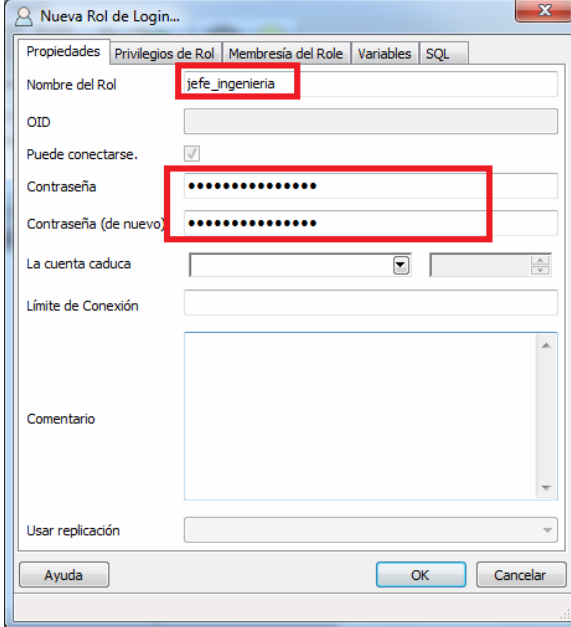
`SUPERUSER INHERIT CREATEDB CREATEROLE;`

Como ejercicio, vamos a suponer que estamos trabajando en un área de SIG que planifica la conectividad a Internet. Pero, como suele ocurrir, nuestra área se provee de información generada en oficinas vinculadas que trabajan específicamente con cada uno de los temas. En este caso el área de “Ingeniería” es quien analiza el alcance de cada una de las antenas de WIMAX. Para garantizarnos la actualización permanente de los datos, debemos dar acceso a esta oficina a la tabla “localidades_wimax” del esquema “salta”, para que modifiquen el dato referido al alcance de las antenas.

Para ello, en primer lugar, vamos a crear el rol “jefe_ingenieria” para asignárselo al coordinador de dicha oficina. Hacemos clic con el botón derecho sobre el objeto “roles de login” y seleccionamos la opción “Nuevo rol de login”.

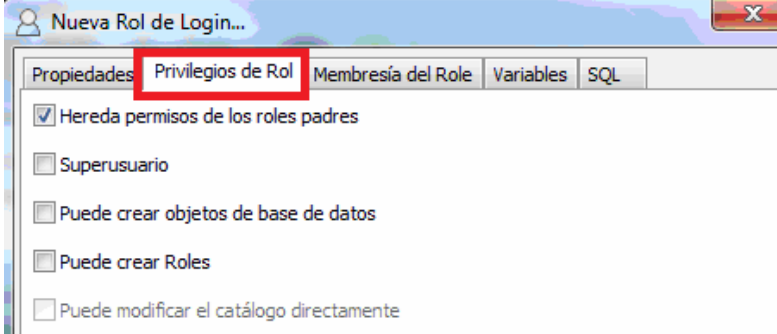


Se nos abre una ventana, donde colocamos como nombre “jefe_ingenieria”. Abajo colocamos una contraseña (sugerimos que para el ejercicio sea igual al nombre: “jefe_ingenieria”). Luego, volvemos a colocar la contraseña en el casillero de abajo.



Donde dice “la cuenta caduca” podríamos elegir una fecha y una hora, si se tratara de un permiso para hacer una tarea puntual. En este caso dejamos el campo en blanco.

En la pestaña “Privilegios de rol” encontramos los atributos similares a los que vimos en el rol “postgres”, y que se pueden habilitar.



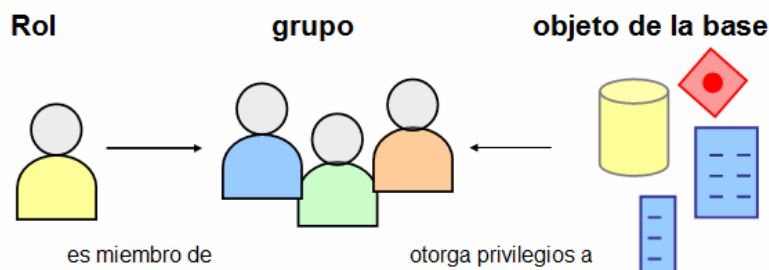
- Hereda permisos de sus roles padres. Veremos que se pueden crear roles dependientes de otros roles. En este caso dejamos habilitada la opción.
- Superusuario. Esto convierte al rol que estamos creando en un usuario que tiene permitido hacer todo. Por supuesto, en este caso no usaremos este atributo.
- Puede crear objetos en la base de datos. Si solo queremos que actualice los datos de “alcance” de las antenas WIMAX, no necesitará crear nuevos objetos.
- Puede crear Roles. En este caso no dejaremos que cree roles. Quizá en otro momento podríamos habilitar este atributo, ya que se trata del jefe de una oficina, y puede necesitar crear roles para las personas que trabajan en su área.

Por ahora cerramos las propiedades del usuario apretando el botón “OK”, y vemos como se ha agregado “jefe_ingenieria” dentro de “roles de login”.



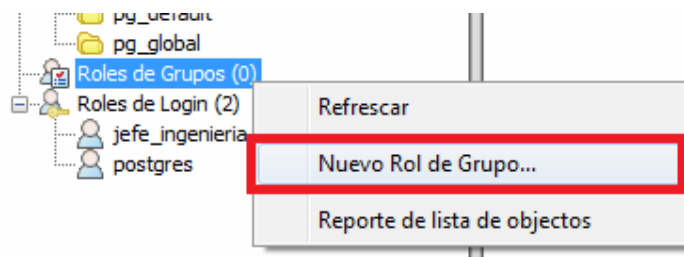
1.2 Roles de grupos

Postgres recomienda manejar los roles a través de grupos. Dichos grupos son, en realidad, roles de los cuales dependen otros roles. De esta manera, es más sencillo otorgar o revocar permisos a un conjunto de roles que tienen un factor en común que permiten agruparlos. Por ejemplo, se puede crear un grupo con roles que pertenecen a una misma oficina, o un grupo con roles que solo tienen permiso de lectura (SELECT) sobre las tablas, etc.

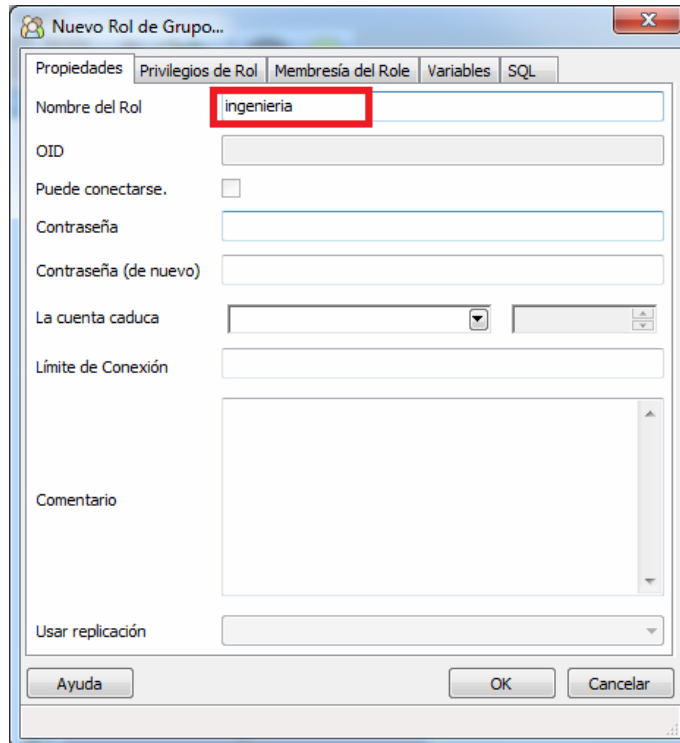


En el ejemplo, tenemos que conceder permisos al área de Ingeniería de nuestra institución, para modificar los datos del campo “alcance” de las antenas de WIMAX. Para ello crearemos el Rol de grupo “ingenieria” y haremos que el rol “jefe_ingenieria” se convierta en miembro de este grupo.

Hacemos clic derecho sobre “roles de grupos” y seleccionamos “nuevo rol de grupo”.



En el nombre, colocamos “ingenieria” y en contraseña no modificamos nada. Recordemos que el rol “jefe_ingenieria” se conectará usando su propia contraseña. El grupo es para asignarle permisos.

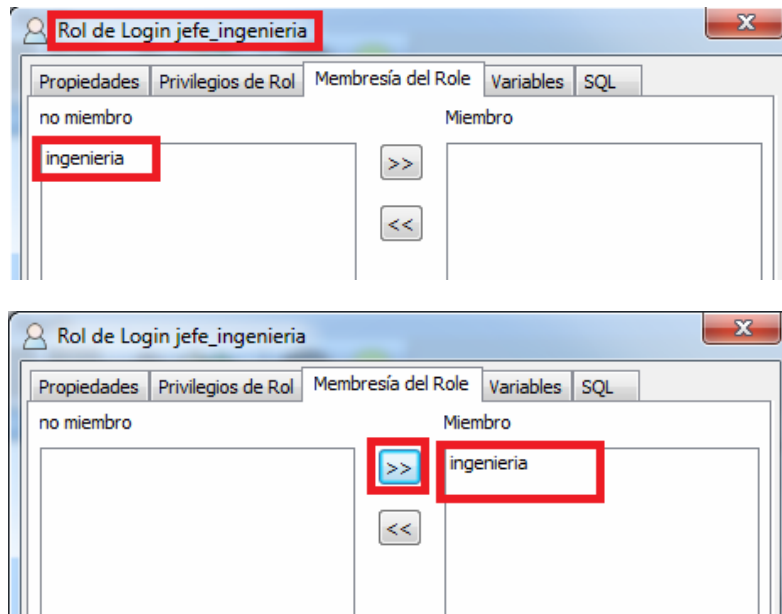


En la solapa “Privilegios de rol” dejamos los atributos que se habilitan por defecto, y en las demás solapas tampoco modificamos nada. Apretamos “OK”, y comprobamos que el rol de grupo ha sido creado.

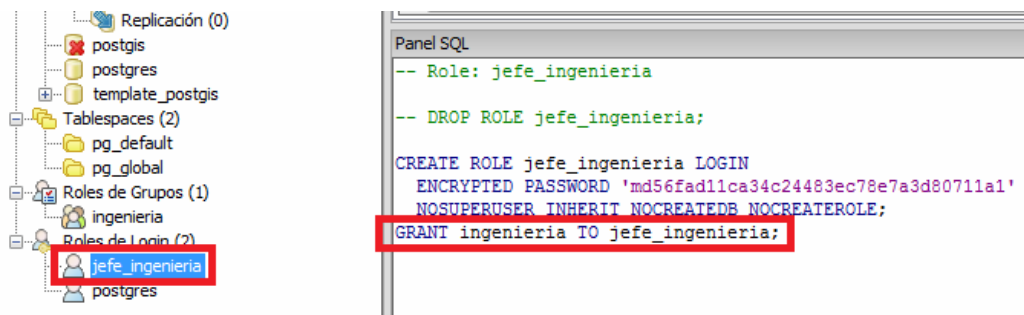


Para convertir al rol “jefe_ingenieria” en miembro del grupo que acabamos de crear, vamos a sus propiedades con un clic derecho sobre el objeto, y seleccionamos la opción “Propiedades”.

Allí, nos dirigimos a la solapa “Membresía de rol”, y vemos que en el recuadro de la izquierda aparecen listados los roles de grupos. En este caso, seleccionamos “ingenieria” y con el botón con las fechas hacia la derecha llevamos el grupo hacia el recuadro “miembro”.



Al apretar “OK”, vemos que la definición SQL del objeto “jefe_ingenieria” ha cambiado, y se ha agregado una nueva línea a la sintaxis:



```
CREATE ROLE jefe_ingenieria LOGIN
  ENCRYPTED PASSWORD 'md56fad11ca34c24483ec78e7a3d80711a1'
  NOSUPERUSER INHERIT NOCREATEDB CREATEROLE;
GRANT ingenieria TO jefe_ingenieria;
```

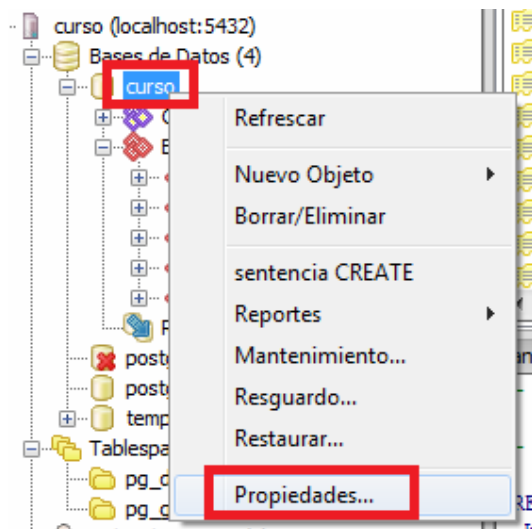
Ahora, el grupo “ingenieria” tiene un miembro que es “jefe_ingenieria”.

1.3 Privilegios

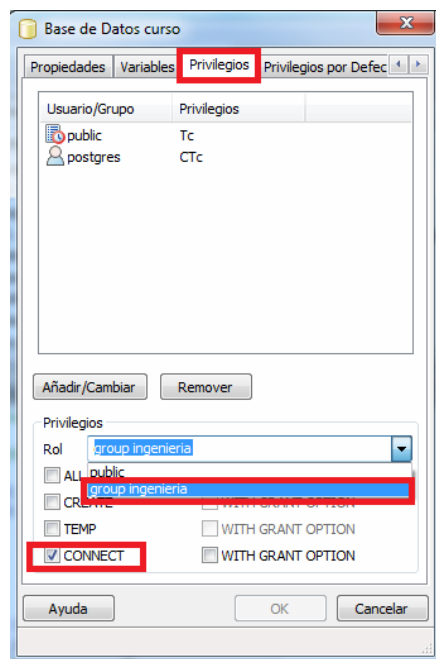
Una vez creado un grupo ya podemos asignarle privilegios, que son permisos especiales para realizar determinadas acciones sobre un objeto de la base. Estos objetos pueden ser: una base de datos, un esquema, una tabla, una columna, una función, etc.

Continuando con el ejercicio, tenemos que lograr que “ingenieria” pueda modificar la columna “alcance” de la tabla “salta.localidades_wimax”. Empezaremos por darle acceso a la base “curso” al rol de grupo “ingenieria”.

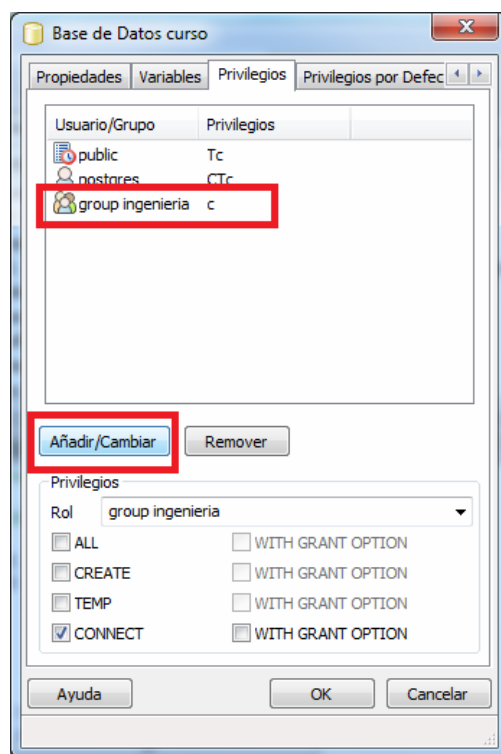
Entramos a las propiedades de la base “curso” (botón derecho sobre la base “curso”, opción “Propiedades”).



En la solapa “Privilegios” vemos que, por defecto, el rol “postgres” ya tiene privilegios asignados. Vamos a agregar el privilegio de poder conectarse a esta base al rol “ingeniería”. Seleccionamos “group ingeniería” del listado que aparece en “rol”, y luego dejamos habilitada solamente la opción “CONNECT”.

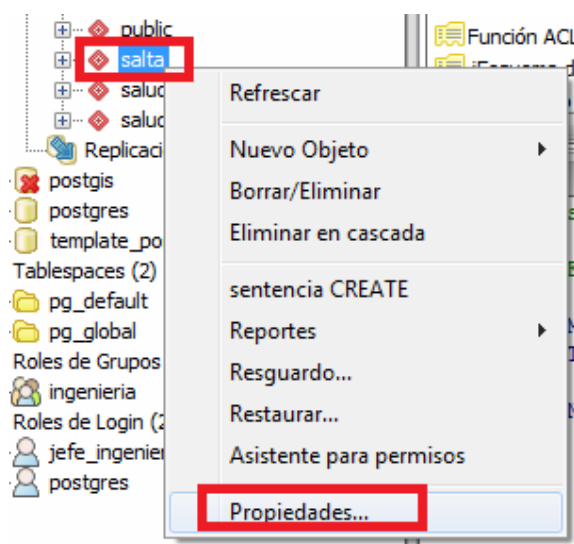


Una vez seleccionado el rol y el privilegio, apretamos el botón “Añadir/cambiar”, para agregar los privilegios.

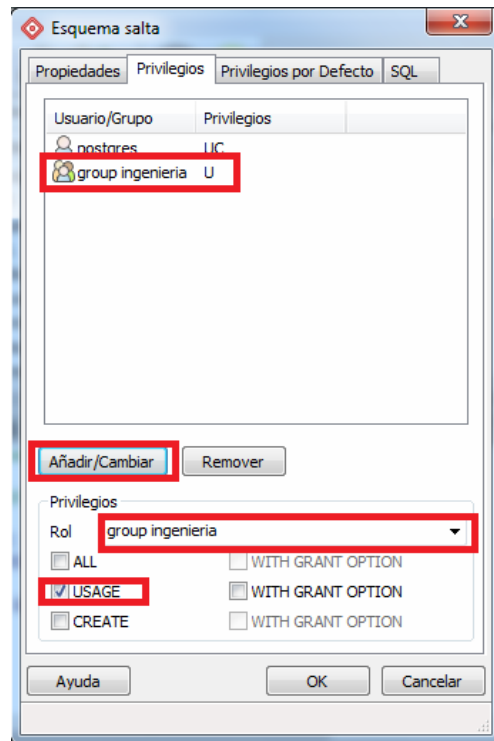


Ahora ya nos podemos conectar con el usuario “jefe_ingenieria” a la base “curso”.

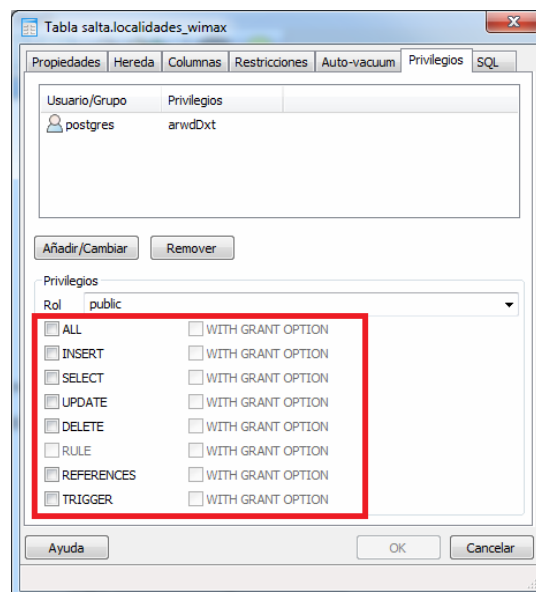
A continuación crearemos los privilegios sobre el esquema “salta”. Vamos a las propiedades del esquema “salta” y en la solapa “privilegios”, seleccionamos el rol “group ingenieria” y habilitamos la opción “USAGE”, que permite el uso del esquema, pero no habilita la creación de objetos ni otras acciones.



Presionamos “Añadir/Cambiar”, y luego “OK”.



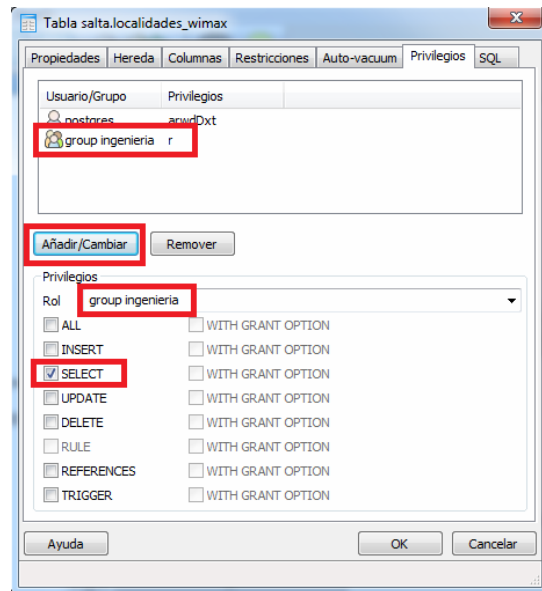
Ahora sí, podemos pasar a la tabla “salta.localidades_wimax”. Vamos a sus propiedades y en “Privilegios” seleccionamos el rol “group_ingenieria”. En el apartado de privilegios, en este caso encontramos opciones que nos resultan más familiares:



- Select
- update
- Insert
- delete, etc.

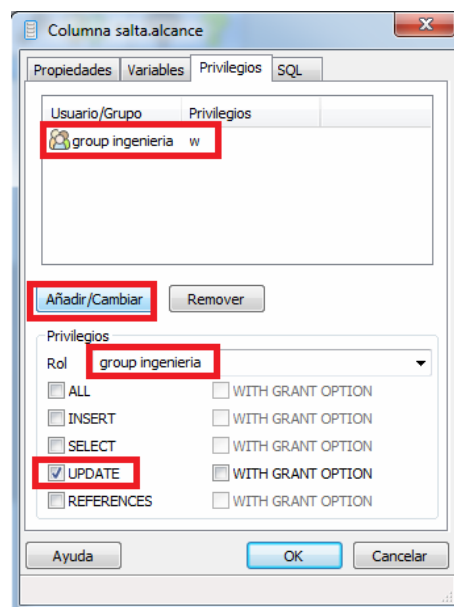
En este caso, aunque queremos que el departamento de ingeniería actualice la tabla, no usaremos la opción “UPDATE”, porque esto le daría permiso de escritura

sobre todas las columnas, cuando nosotros queremos que sólo actualice los datos de la columna “alcance”. En por esto que seleccionaremos solamente la opción “SELECT”.

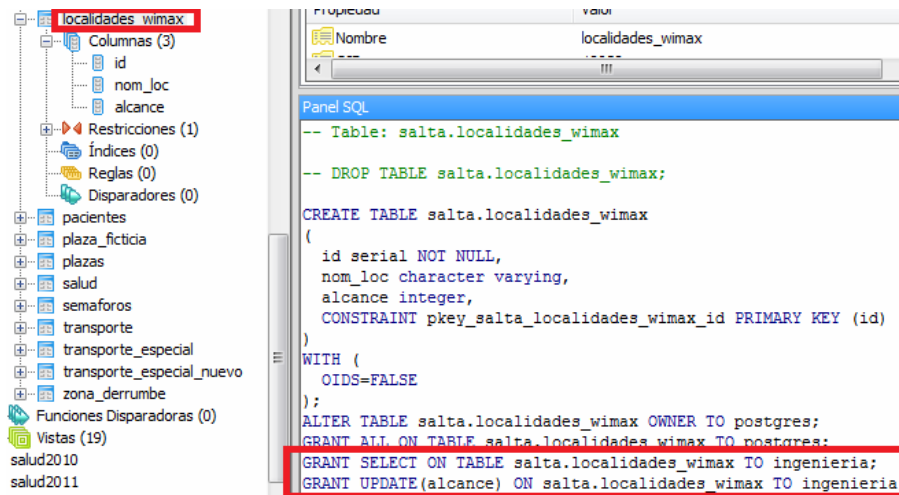


Apretamos “Añadir/cambiar” y luego “OK”.


Para terminar, iremos a las propiedades de la columna “alcance”, y ahí sí otorgaremos el permiso de escritura. En la solapa “privilegios”, como siempre en “rol” seleccionaremos “group ingenieria”. Pero en los privilegios optaremos por “UPDATE”, ya que necesitamos que “ingenieria” actualice estos datos.

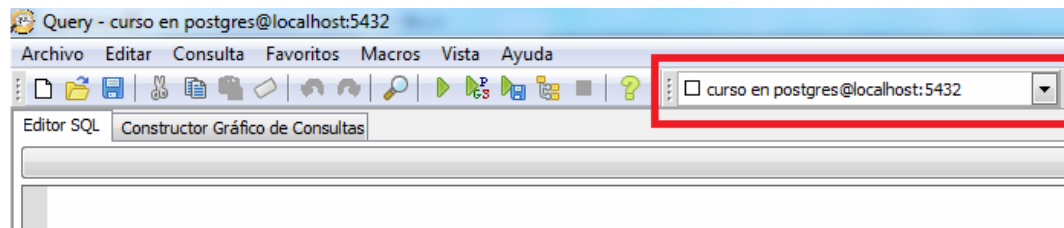


Apretamos “OK”, y procederemos a probar los cambios que hemos realizado. En primer lugar, podemos ver en el panel de la derecha que las propiedades de la tabla han cambiado:



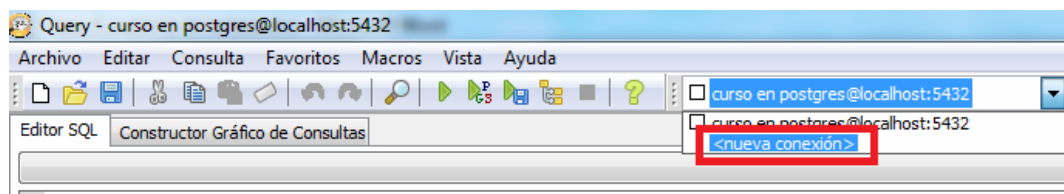
Pero ahora, vamos a comprobar los privilegios y limitaciones que tenemos

usando el constructor de consultas arbitrarias . Abrimos el constructor de consultas y vemos que en la parte superior, hay un elemento desplegable en donde figuran los parámetros de conexión a la base actuales.

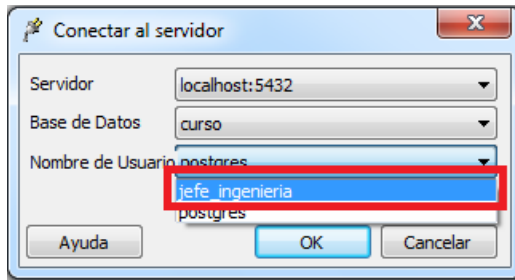


curso en **postgres**@localhost:5432

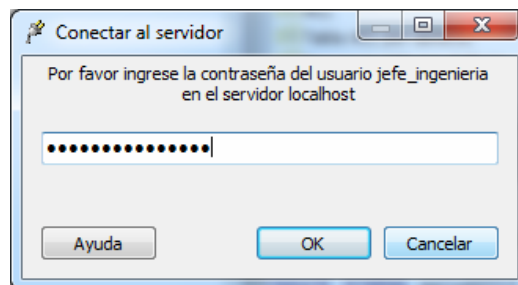
Donde “curso” es el nombre de la base, “postgres” es el nombre de usuario, “localhost” (o el nombre que corresponda) es el nombre del servidor donde se halla instalado Postgres, y “5432” es el puerto que usa Postgres para conectarse. Para cambiar estos parámetros y conectarnos usando el usuario “jefe_ingenieria”, desplegamos el listado y seleccionamos la opción “nueva conexión”.



Esto nos abre un cuadro de diálogo, donde podemos cambiar el nombre del usuario. Cambiamos “postgres” por “jefe_ingenieria”.



Vemos también que nos pide la contraseña de este nuevo usuario. Colocamos: “jefe_ingenieria” y luego presionamos “OK”.



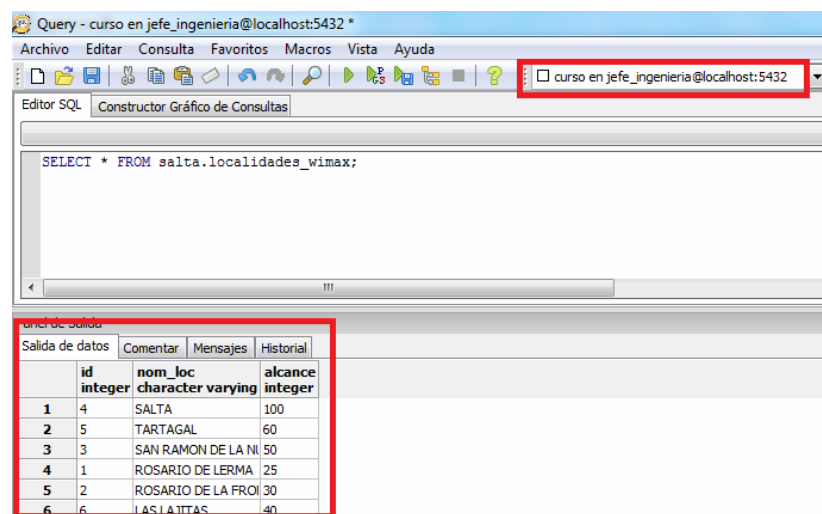
Ahora sí, podemos empezar a trabajar usando sólo los permisos otorgados al grupo “ingenieria” al cual pertenece nuestro usuario.

Recordemos que tenemos los siguientes privilegios:

- Acceso a la base de datos “curso”
- Uso del esquema “salta”
- Lectura de la tabla “salta.localidades_wimax”
- Escritura sobre la columna “alcance”.

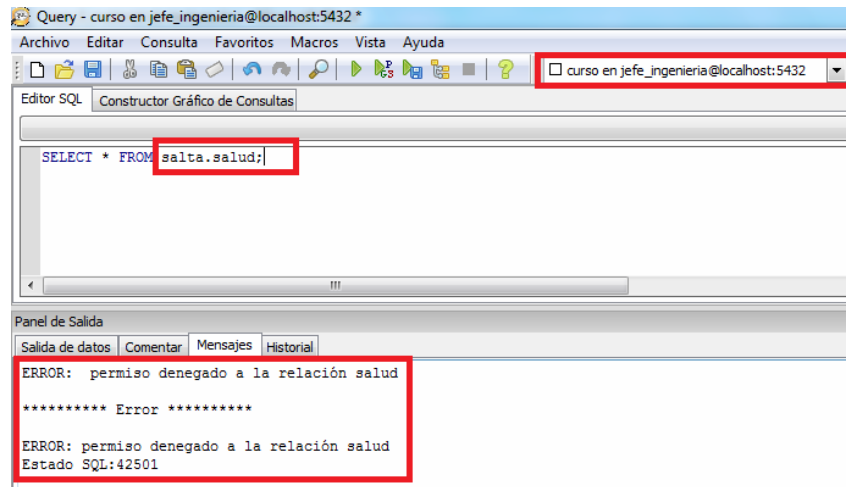
Si corremos una consulta de selección sobre “localidades_wimax”, obtendremos el resultado.

```
SELECT * FROM salta.localidades_wimax;
```



Pero si hacemos lo mismo con otra tabla sobre la cual no tenemos permiso de lectura:

```
SELECT * FROM salta.salud;
```



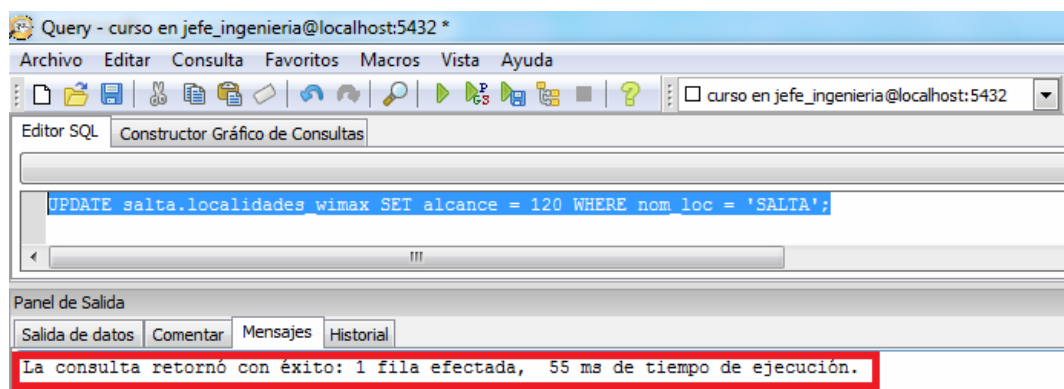
Nos aparecerá un mensaje de error:

```
ERROR: permiso denegado a la relación salud
```

```
***** Error *****
```

Con el usuario del área de Ingeniería de nuestra institución podemos actualizar un valor en el campo "alcance", pero no en el campo "nom_loc"

```
UPDATE salta.localidades_wimax SET alcance = 120 WHERE nom_loc = 'SALTA';
```



```
UPDATE salta.localidades_wimax SET nom_loc = 'CIUDAD DE SALTA' WHERE nom_loc = 'SALTA';
```

La segunda consulta nos tira el error relacionado con los permisos, porque no tenemos privilegios de UPDATE sobre el campo "nom_loc".

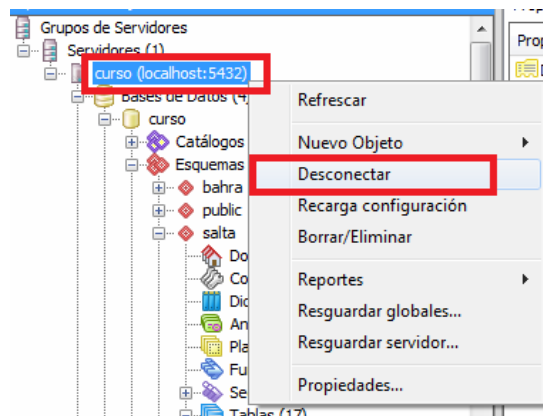
ACTIVIDAD 1

- Restaurar los archivos “clase6_public” y “clase6_salud” haciendo clic derecho sobre la base “curso”. Y correr las consultas de “clase6.sql”.
- Crear el usuario “tecnico_ingenieria” y hacerlo miembro del grupo “ingenieria”. Copiar al documento de texto, las consultas SQL que le corresponden en el panel de propiedades.
- Otorgar los privilegios necesarios para que la oficina de Ingeniería pueda insertar nuevos registros en la tabla “salta.salud”. Copiar al documento de texto, las consultas SQL que le corresponden a la tabla “salud” en el panel de propiedades.

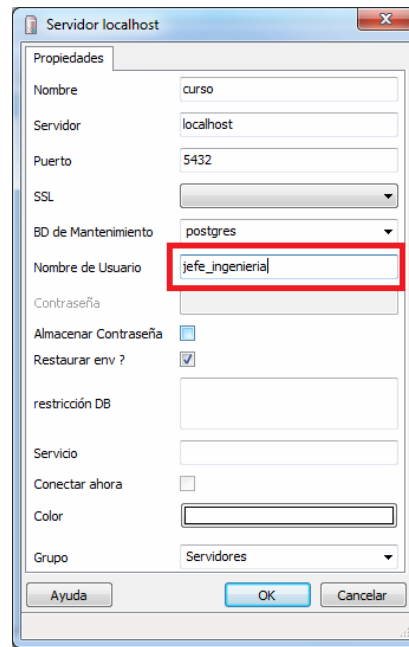
1.4 Acceder a Postgres con otro usuario

En este ejercicio, hemos probado como funcionan los privilegios usando el constructor de consultas. Pero también podemos entrar con el usuario y la clave de “jefe_ingenieria” y comprobar los accesos y limitaciones.

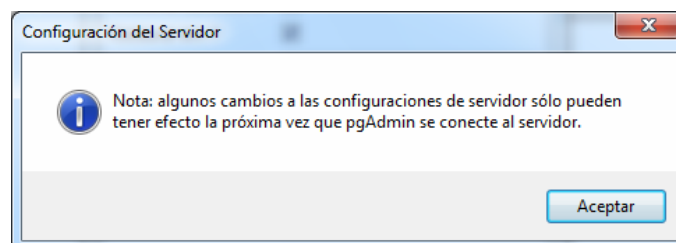
Primero debemos desconectarnos. Hacemos un clic derecho sobre el tercer objeto del árbol de objetos: curso en localhost:5432 (o el nombre del servidor y puerto que corresponda) y en las opciones seleccionamos “Desconectar”.



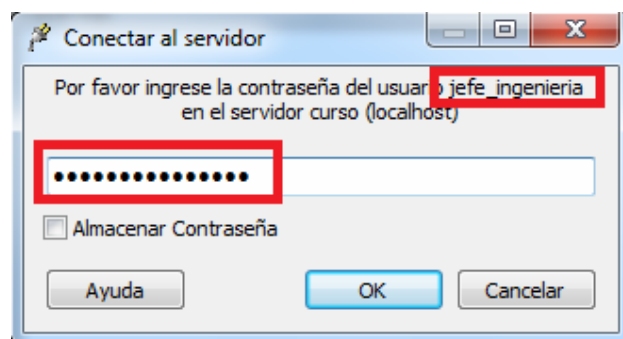
Una vez desconectados, vemos que el árbol de objetos se retrae, y nos queda solo el nombre de la conexión “curso” que creamos en la clase 1. Hasta aquí siempre nos hemos conectado con el usuario “postgres”, pero ahora lo haremos con el usuario “jefe_ingenieria”.



Clickeamos con el botón derecho sobre la conexión “curso (localhost:5432)” e ingresamos a las propiedades. Dentro de las propiedades, cambiamos el nombre de usuario por “jefe_ingenieria” y deshabilitamos el checkbox de “almacenar contraseña”. Finalmente apretamos “OK”, y vemos aparecer un mensaje que nos avisa que los cambios se efectuarán al volver a conectarnos. Aceptamos.

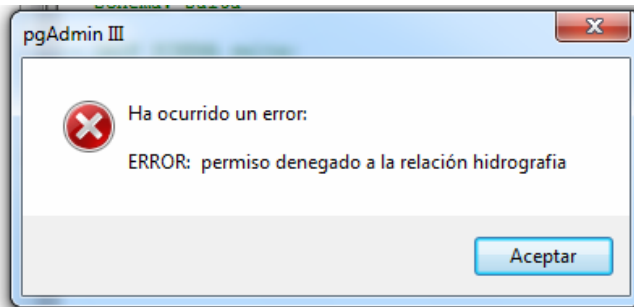
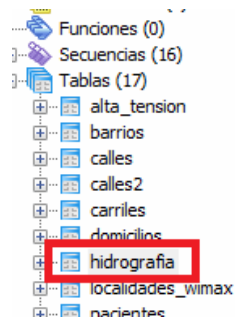


Esta acción guardó la configuración para acceder como “jefe_ingenieria”. Ahora accedemos haciendo doble clic sobre el mismo objeto, o con clic derecho y la opción “conectar”. Nos pide la contraseña, que también es “jefe_ingenieria” y luego nos abre el árbol de objetos.



Podemos explorar el árbol de objetos, pero al traer algunas propiedades de ciertos objetos, nos aparecerá un mensaje de permiso denegado. Lo que sucede es

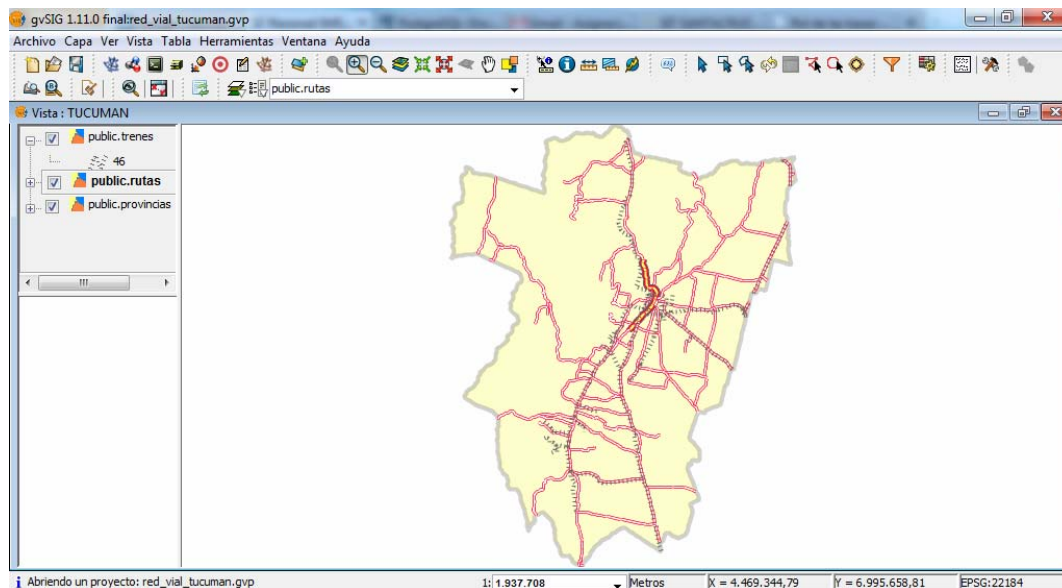
que para obtener las propiedades de esos objetos (como la cantidad de filas de una tabla), es necesario correr consultas a las que el usuario “jefe_ingenieria” no tiene acceso.



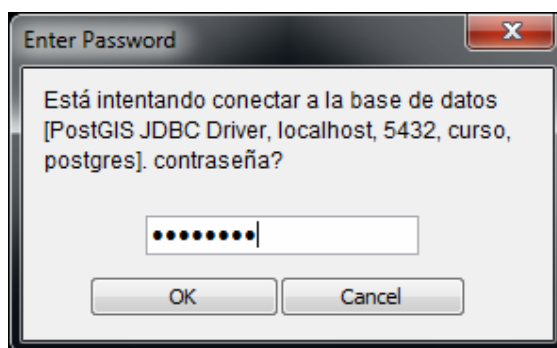
2 Gestión de proyectos de gvSIG con PostGIS

gvSIG utiliza archivos de extensión “.gvp” para almacenar los proyectos. Un proyecto de gvSIG consiste en un documento de texto en formato XML con instrucciones referidas a: la ruta de los archivos shapes utilizados, la simbología aplicada a cada capa, los métodos de clasificación utilizados, las proyecciones de vistas y capas, los nombres de las vistas, las conexiones a la bases de datos, las tablas visualizadas, los filtros, etc..

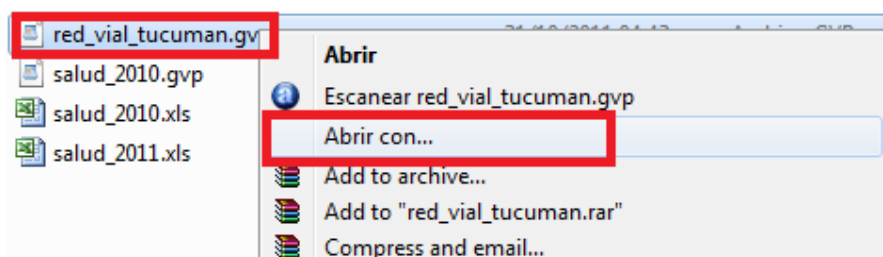
Tenemos, como ejemplo, el proyecto “red_vial_tucuman” en el que se visualizan tres capas de Postgres: provincias, rutas y trenes.



Al abrir el proyecto, gvSIG nos pedirá la contraseña de acceso a la base para el usuario “postgres”.



Estas tres capas están siendo consultadas desde Postgres aplicando un filtro para visualizar solamente los elementos que corresponden a la provincia de Tucumán.



Si abrimos el archivo “red_vial_tucuman.gvp” como archivo de texto plano (con WordPad, Block de notas, Gedit, etc.) veremos que en el encabezado aparecen los datos de configuración general de gvSIG, como la proyección por defecto, el color que se utiliza al seleccionar un elemento, etc.

```
value="com.iver.cit.gvsig.project.Project"/>
  <property key="VERSION" value="1.11.0"/>
  <property key="comments" value=""/>
  <property key="creationDate" value="11-oct-2011"/>
  <property key="isAbsolutePath" value="true"/>
  <property key="numExtents" value="0"/>
  <property key="numCameras" value="0"/>
  <property key="data-source-count" value="10"/>
  <property key="numDocuments" value="1"/>
  <property key="modificationDate" value="31-oct-2011"/>
  <property key="name" value="red_vial_tucuman.gvp"/>
  <property key="owner" value=""/>
  <property key="selectionColor" value="255,255,0,255"/>
  <property key="projection" value="EPSG:23030"/>
  <property key="followHeaderEncoding" value="true"/>
</xml-tag>
  <property key="type" value="sameDriverFile"/>
  <property key="gdbmsname" value="gdbms_82cc80e_
1335b43899e_7fff"/>
```

También vemos los datos de la vista creada “TUCUMAN”.

```
<xml-tag>
  <property key="className" value="ProjectView"/>
  <property key="comment" value=""/>
  <property key="creationDate" value="11/10/11 15:34"/>
  <property key="name" value="TUCUMAN"/>
  <property key="owner" value=""/>
  <property key="numViews" value="1"/>
  <property key="mapOverView" value="true"/>
</xml-tag>
  <property key="className"
```

Parámetros de conexión a la base, y muchos otros.

```

value='localhost' />
value="true" />
value='curso' />
<property key="SRID" value="22184" />
<property key="host"
<property key="port" value="5432" />
<property key="dbName" value="curso" />
<property key="literalDBName"
<property key="connName"
</xml-tag>

```

Al analizar este archivo, vemos que modificando algunos pocos datos, podemos crear un nuevo proyecto, similar al actual, pero en donde se visualicen los datos de otra provincia.

Por ejemplo, si buscamos y reemplazamos todas las veces que aparece:

```

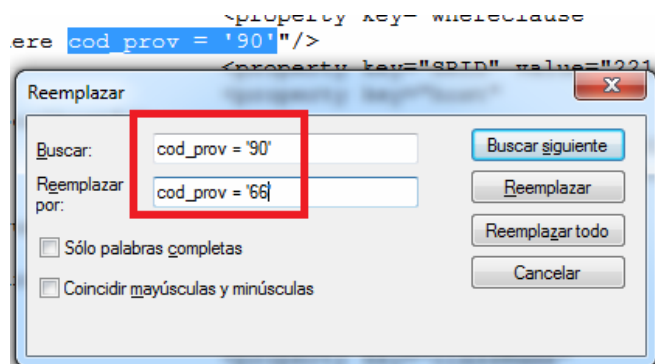
value="the_geom" />
value="where cod_prov = '90' />
value="localhost" />
<property key="whereclause"
<property key="SRID" value="22184" />
<property key="host"

```

`cod_prov = '90'`

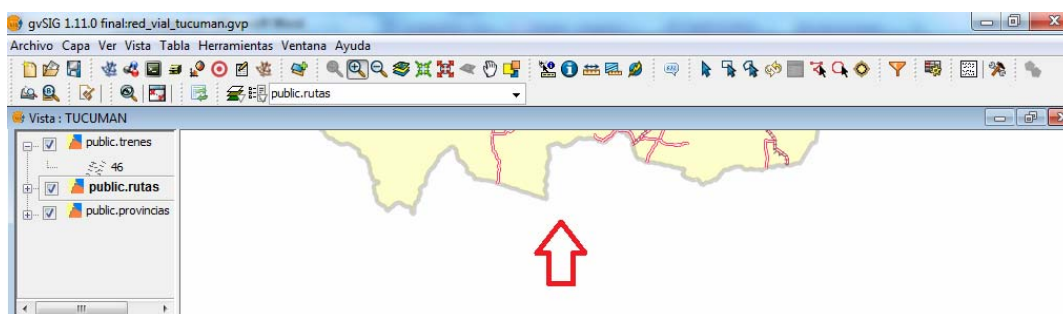
...que es el filtro para la provincia de Tucumán, por:

`cod_prov = '66'`

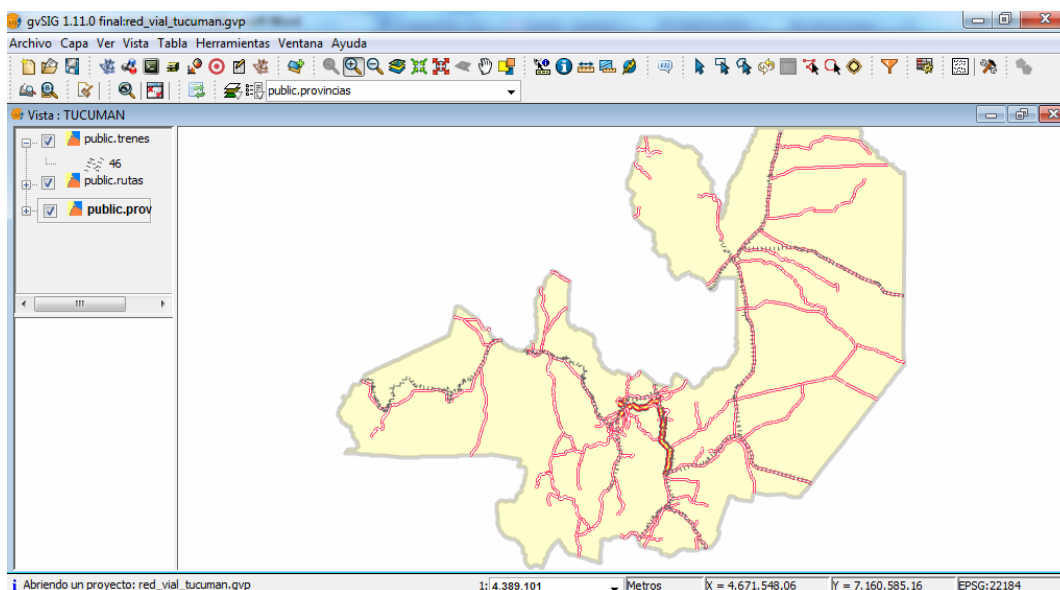


...para traer los datos de Salta, el proyecto dibujará el polígono de la provincia y las rutas y los trenes de Salta. Por supuesto, para no perder el archivo de Tucumán, en lugar de “guardar”, utilizaremos la opción “guardar como...” para crear un nuevo archivo.

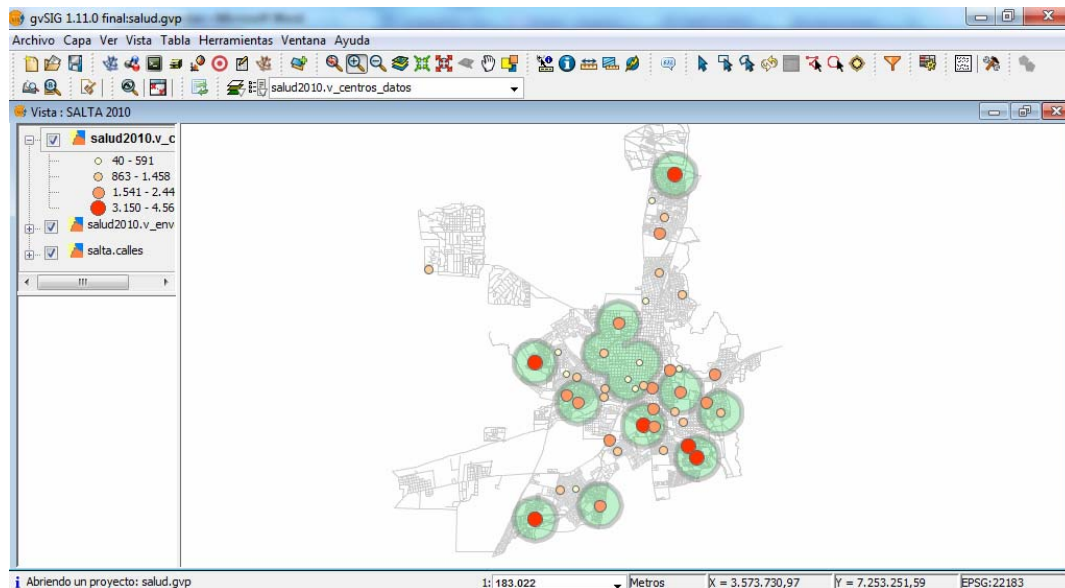
Ahora, al abrir el nuevo proyecto “red_vial_salta.gvp”, vemos como se dibujan los datos para esta provincia. El encuadre seguirá siendo el de Tucumán, por lo cual deberemos hacer un zoom a la capa usando, por ejemplo, la capa del límite provincial.



Lo mismo se puede hacer cuando tenemos un proyecto con datos que provienen de un esquema, y modificamos el archivo “.gvp” para que lea los datos de otro esquema que posee una estructura similar.



Si abrimos el proyecto “salud2010.gvp”, vemos que aparecen los centros de salud de Salta, clasificados según la cantidad de pacientes atendidos en 2010. También tenemos las envolventes de 1000 metros de los centros de salud que poseen ambulancia. Si queremos obtener esta misma vista para el año 2011, podemos aprovechar las similitudes de los esquemas salud2010 y salud2011.

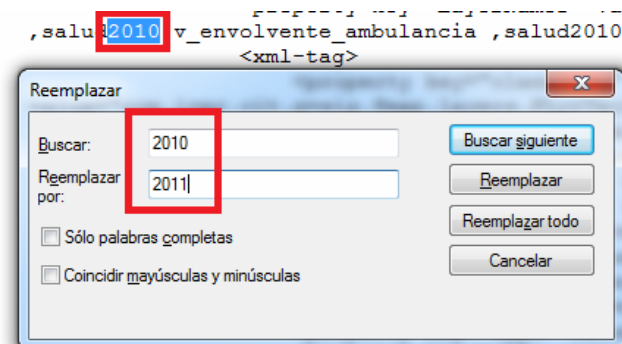


Los esquemas salud2010 y salud2011 tienen, cada uno, tablas llamadas “datos_basicos” con igual estructura. Si corremos los scripts que crean las vistas del esquema salud2010 “v_centros_datos” y “v_envolvente_ambulancia”, pero cambiando el nombre del esquema en cada caso, obtendremos esas mismas vistas para el esquema salud2011:

```
CREATE OR REPLACE VIEW salud2010.v_centros_datos AS
SELECT a.gid, a.the_geom, b.direct, b.pacientes, b.cod_csal,
b.internacion, b.ambulancia
FROM salta.salud a
JOIN salud2010.datos_basicos b ON a.centro = b.cod_csal;
```

```
CREATE OR REPLACE VIEW salud2010.v_envolvente_ambulancia AS
SELECT min (v_centros_datos.gid) AS gid,
st_union(buffer(v_centros_datos.the_geom, 1000)) AS the_geom
FROM salud2010.v_centros_datos
WHERE v_centros_datos.ambulancia = 'SI';
```

Ahora que tenemos las vistas que usa el proyecto salud2010.gvp, podemos crear el proyecto salud2011.gvp. Lo que haremos es abrir el archivo .gvp con WordPad y luego buscaremos y reemplazaremos “2010” por “2011”.



**ACTIVIDAD 2**

- Crear el proyecto “salud2011.gvp” para ver los datos de atención de pacientes y área de influencia de ambulancias para el año 2011.
- Crear una impresión de pantalla con la vista y pegarla en el documento de texto.

3 Las bases de datos espaciales y las Infraestructuras de Datos Espaciales IDE.

Una infraestructura de datos espaciales (IDE) es un conjunto articulado de **tecnologías, políticas, acuerdos institucionales, recursos y procedimientos estandarizados de trabajo**, cuya meta principal es asegurar la cooperación para hacer accesible la información geográfica.

Como vemos, lo tecnológico sólo es una parte de lo que implica el concepto de IDE, pero veremos que es allí donde principalmente entran en juego las bases de datos espaciales.

El surgimiento de una IDE responde principalmente, a dos necesidades. Por una lado, por parte de los productores, está la necesidad de **poner a disposición** de la comunidad los datos producidos, y por parte de los usuarios, la necesidad de **acceder** a estos mismos datos. En realidad la mayoría de los actores que forman parte de una IDE tienen ambos roles: de usuarios y productores, y es por ello que la sinergia necesaria para conformar una IDE encuentra su lugar. Otra necesidad es la de **reutilizar** los recursos generados para un propósito específico, en nuevos proyectos, de tal manera que se vuelva más eficiente la inversión en áreas que utilizan información geográfica.

Una IDE debe proveer los medios y los recursos necesarios para que los usuarios puedan encontrar, evaluar las condiciones y disponer de los datos geográficos. Para ellos se utilizan los **metadatos y los servicios de catálogo, visualización y descarga**.

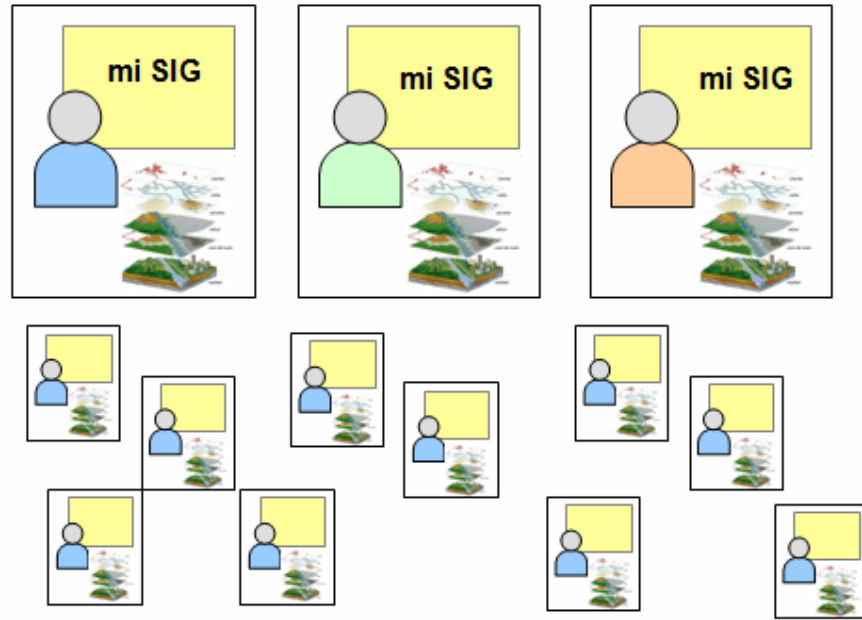
Adicionalmente, la IDE debe asegurarse de que los datos movilizados sean correctos, y que además estén permanentemente actualizados. Para ello es necesario que las áreas productoras de información sean provistas de los recursos necesarios (de todo tipo) para que puedan garantizar la mejor información.

La inclusión del concepto de **infraestructura** se debe a que, al igual cuando hablamos de caminos, autopistas, o redes de gas y agua, el armado de una IDE es un tejido complejo que se arma para luego dar soporte a infinidad de nuevas actividades, como las evaluaciones de impacto ambiental, el ordenamiento territorial, la planificación de políticas públicas, los diferentes servicios al ciudadano y a las actividades productivas, etc.

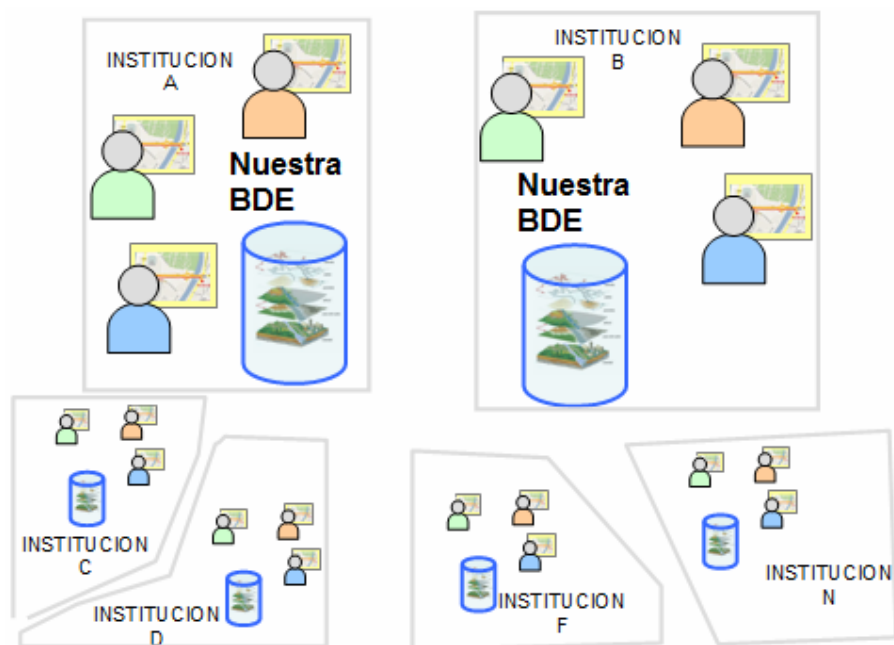
Una forma de comprender la relación de las bases de datos con las IDE, es analizando los diferentes esquemas de integración entre actores que nos llevan a salir

del SIG de escritorio, para integrarnos a toda la comunidad relacionada con la información geográfica.

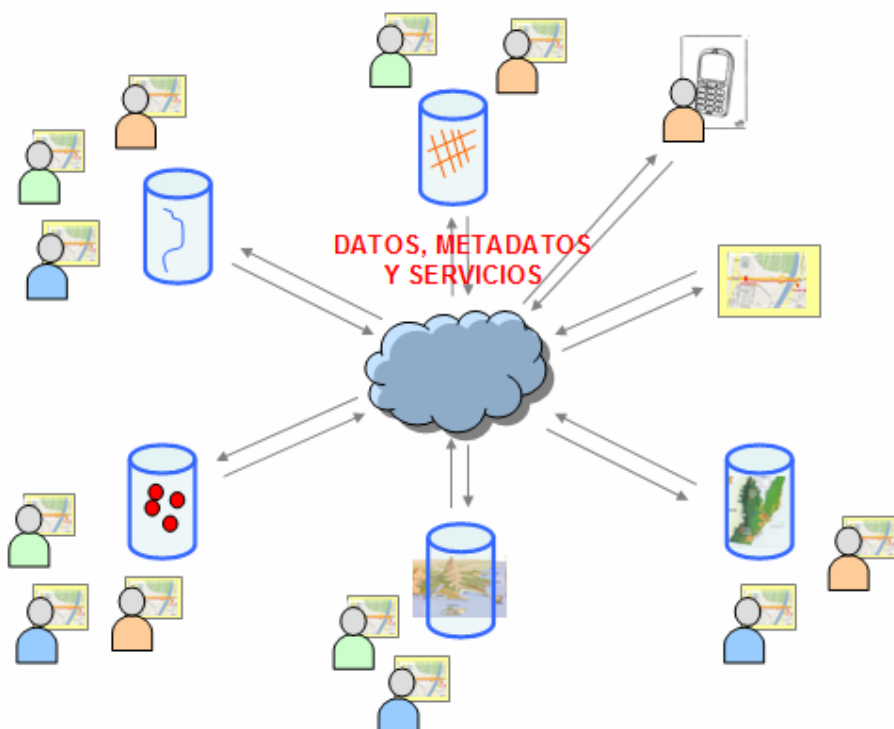
Esquemáticamente, podemos ver que en un primer momento, en nuestras oficinas predominó el llamado “**SIG monolítico**”, es decir, un SIG de escritorio que utiliza la información almacenada en la propia computadora donde se gestiona la cartografía.



Una segunda fase sería el esquema de trabajo que incluye a **las bases de datos espaciales**. En este caso se trabaja utilizando las ventajas de un base de datos espaciales (se evita la redundancia, acceso concurrente, integridad referencial, administración centralizada, etc.), pero la información que no es producida por la propia institución es traída en forma manual, muchas veces incumpliendo las normas, sin posibilidad de garantizar su actualización permanente, generando enormes gastos a la institución o al conjunto del Estado, etc.). Esto ocurre, generalmente, con la información de base, como rutas, hidrografía, relieve, etc. que son capas producidas por determinados organismos oficiales.



La tercera fase sería el trabajo en el contexto de una **Infraestructura de Datos Espaciales**, en donde cada institución trabaja utilizando la información de base de las instituciones productoras, pero a la que accede mediante servicios Web que garanticen su disponibilidad, confiabilidad y nivel actualización. La institución simplemente trae la información que utiliza como base, sabiendo que es la mejor que existe, y solo se dedica a producir la información “temática” que le corresponde. El conjunto de la comunidad, de esta manera, evita enormes gastos, y tiene disponible la mejor información geográfica.



Luego de leer este último párrafo, si volvemos a la definición inicial de IDE, vemos que el componente tecnológico es importante, pero mucho más lo son los **acuerdos institucionales** que se deben establecer para garantizar el funcionamiento de esta infraestructura.

Una base de datos espaciales es un gran paso en la integración de la información, los procedimientos, los códigos, los formatos, los sistemas de referencia, relacionados con la información geográfica dentro de una institución. Y también constituye una **fortaleza** frente a la oportunidad de integrar la institución a la IDE.

Además de los aspectos enumerados anteriormente, la base de datos cumple la función de almacenar los datos que se publican, guardar la configuración de los servicios IDE que se brindan, puede gestionar los usuarios de dichos servicios y sus permisos, puede almacenar y gestionar - junto a aplicaciones específicas - los metadatos, puede gestionar el contenido, las herramientas otros recursos de los geoportales, almacenar y gestionar la numerosa documentación y recursos que se generan en la interacción con otras instituciones, etc.

Para conocer más sobre IDEs: ¿Qué es una IDE? En el portal del Sistema de Información Territorial de la Provincia de Santa Cruz.

<http://www.sitsantacruz.gov.ar/iniciog.php?opcion=0502>



ACTIVIDAD 3

Actividad final de integración

Escribir un breve proyecto que pueda ser llevado a cabo en su institución y que implique la utilización de bases de datos espaciales.

1. Nombre de la institución.
2. Temáticas que aborda la institución
3. Nombre del proyecto
4. Síntesis.
5. Responder:
 - a. ¿Qué software usaría? Justificar en 3 renglones.
 - b. Este proyecto se podría realizar sin utilizar una base de datos espaciales? Justificar.
 - c. Qué elementos de la base utilizaría para garantizar la integridad referencial de los datos? Dar un ejemplo concreto relacionado con el proyecto.
 - d. Describir un caso en que se debería utilizar una multigeometría en el proyecto. (multipunto, multilínea, multipolígono).
 - e. Enumerar los geoprosos y procedimientos de geocodificación que se utilizarían en el proyecto. Desarrollar un ejemplo.
 - f. Definir un ejemplo de un grupo, con sus roles y atributos. Desarrollar un ejemplo en donde sea necesario otorgar privilegios de distintos tipos.
 - g. **Opcional:** qué papel cumpliría la base de datos espaciales generada a partir de este proyecto en relación a la IDE que le corresponde a su jurisdicción?

Índice

1 ADMINISTRACIÓN DE LA BASE DE DATOS ESPACIALES.....	1
1.1 ROLES	2
1.2 ROLES DE GRUPOS	5
1.3 PRIVILEGIOS	7
1.4 ACCEDER A POSTGRES CON OTRO USUARIO.....	15
2 GESTIÓN DE PROYECTOS DE GVSIG CON POSTGIS.....	17
3 LAS BASES DE DATOS ESPACIALES Y LAS INFRAESTRUCTURAS DE DATOS ESPACIALES IDE.....	22

Usted es libre de compartir - copiar, distribuir, ejecutar y comunicar públicamente y de hacer obras derivadas de este documento

Este documento es una obra compartida bajo la licencia Creative Commons.

Atribución-NoComercial-CompartirIgual 2.5 Argentina (CC BY-NC-SA 2.5)



Atribución — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).



No Comercial — No puede utilizar esta obra para fines comerciales.



Compartir bajo la Misma Licencia — Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Aviso: Al reutilizar o distribuir la obra, tiene que dejar muy en claro los términos de la licencia de esta obra. La mejor forma de hacerlo es enlazar a la siguiente página:

<http://creativecommons.org/licenses/by-nc-sa/2.5/ar/>

Programa Nacional Mapa Educativo
Ministerio de Educación
República Argentina

Teléfono/Fax: 54 11 4129-1408
Correo electrónico: mapaedu_nac@me.gov.ar
www.mapaeducativo.edu.ar

