



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Desarrollo de Aplicaciones en Geomática

Claudio Álvarez (Cátedra)

Ignacio Yañez (Laboratorio)



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Objetivo

Esta asignatura busca como objetivo entregar al **Ingeniero Civil en Geografía (ICG)** las herramientas y metodologías de **gestión de proyectos geoinformáticos** necesarios, para la construcción de aplicaciones basadas en **Sistemas de Información Geográfica (SIG o GIS*)**, con el objetivo de generar productos de información que apoyen la **toma de decisiones sobre el territorio**.

* Acrónimo en inglés de *Geographic Information System*



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Arquitectura GIS Organizacional



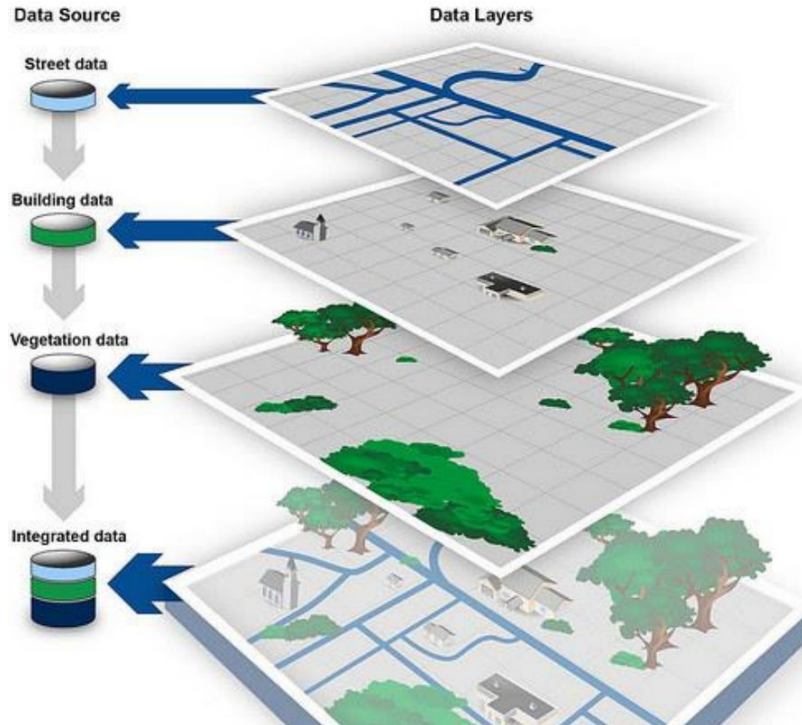
Objetivo

El presente contenido tiene por finalidad introducir al estudiante en el uso de tecnologías de información y datos espaciales en un contexto organizacional a distinta escalas funcionales, contexto administrativo, expertis y necesidades del usuario final, pudiendo diferenciar y proponer arquitectura de desarrollo según la toma de decisiones requeridas.

Contenido:

- 1: Arquitectura de Sistema desde ESRI y Open Source
- 2: ICG y su aporte en la explotación de las tecnologías GIS (Arquitectura y Soluciones)

¿Qué es un Dato Geográfico?



Fuente: Researchgate

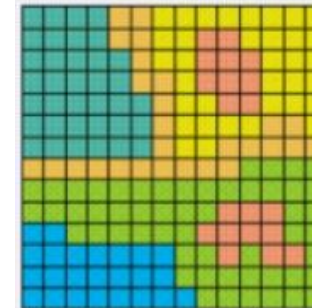
Entidad espacio-temporal que cuantifica la distribución, estado y los vínculos de los fenómenos u objetos naturales y sociales.

- Posición absoluta: sobre un sistema de coordenadas (x, y, z).
- Posición relativa: frente a otros elementos del paisaje.
- Pertenece a un sistema de proyección de coordenadas.
- Es Representado por un: punto, línea, polígono.
- Posee atributos que lo describen.
- En algunos casos, estas entidades poseerán atributos temporales $[f(x,y,z,t)]$



Tipos de datos Geográficos

Ítem	Vectorial	Raster
Volumen de almacenamiento	Baja	Alta
Precisión en cálculo de superficies y distancias	Alta	Media - Baja
Precisión en representatividad geométrica	Alta (entidades geométricas)	Media - Baja (tamaño de píxel)
Complejidad de estructura	Alta	Baja
Dimensiones geográficos	2d - 3d	2d - 3d - 4d*
Representación gráfica de de los elementos en la realidad	Depende del nivel de abstracción del usuario	Alta (imagen satelital)
Procesamiento	Fácil (geoprocesamiento y cálculo de atributos)	Complejo (cálculo matricial y combinaciones de bandas)



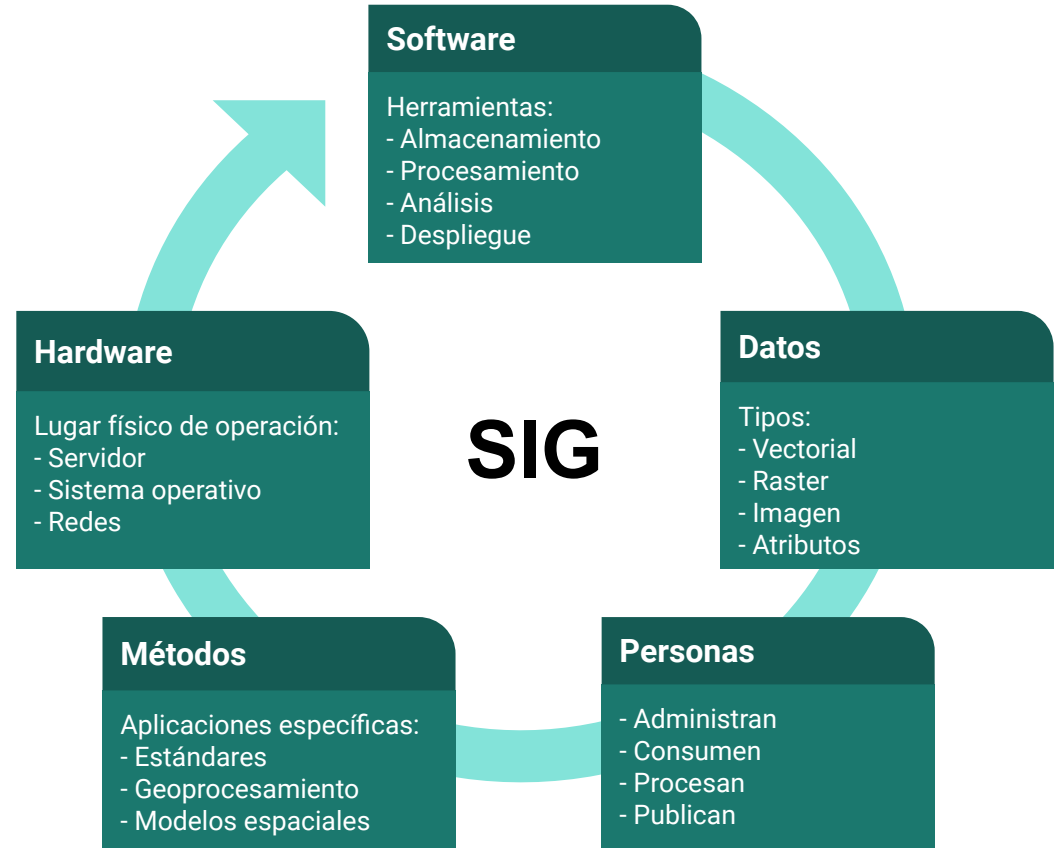
1	1	1	1	2	2	4	3
1	1	2	2	2	4	3	6
1	2	2	2	5	4	6	6
2	2	2	5	4	3	6	6
2	2	5	2	4	4	6	6
2	5	5	2	5	4	4	3
5	4	4	5	2	5	4	4
4	4	4	4	4	2	5	4

Fuente: ArcGIS - Raster fundamentals



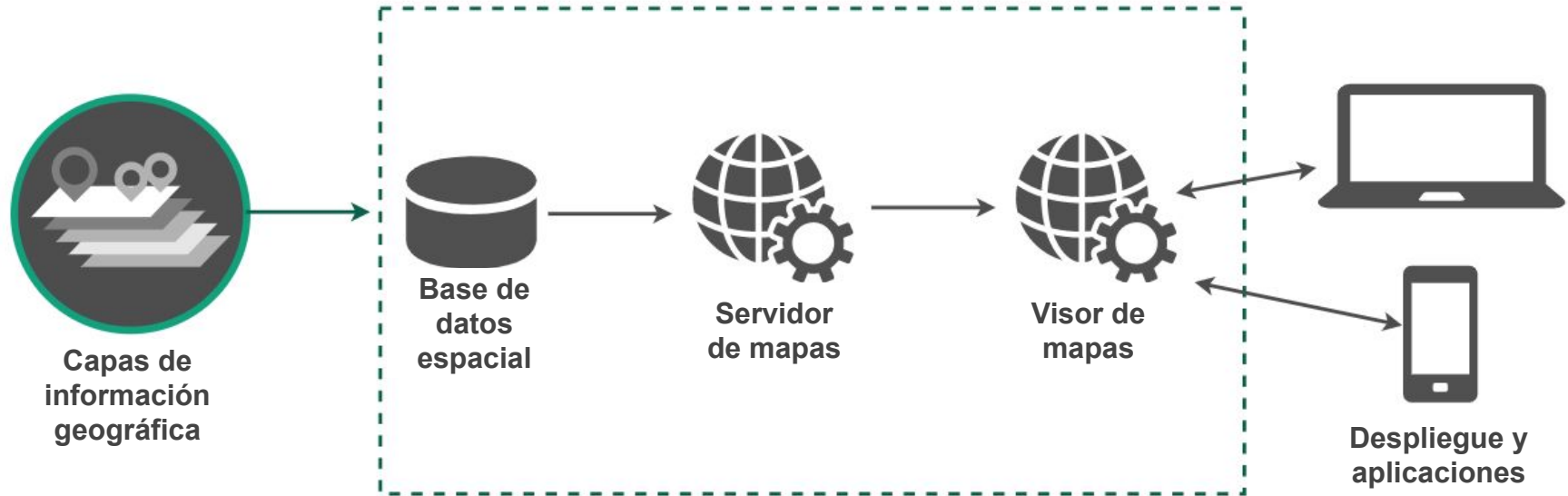
Sistema de Información Geográfica (SIG)

Es un sistema compuesto de hardware y software, que junto con métodos y recursos humanos relacionados, permite la visualización, análisis, e interpretación de información geográfica para comprender relaciones, patrones y tendencias a través de la generación de un producto cartográfico.





Arquitectura de un SIG

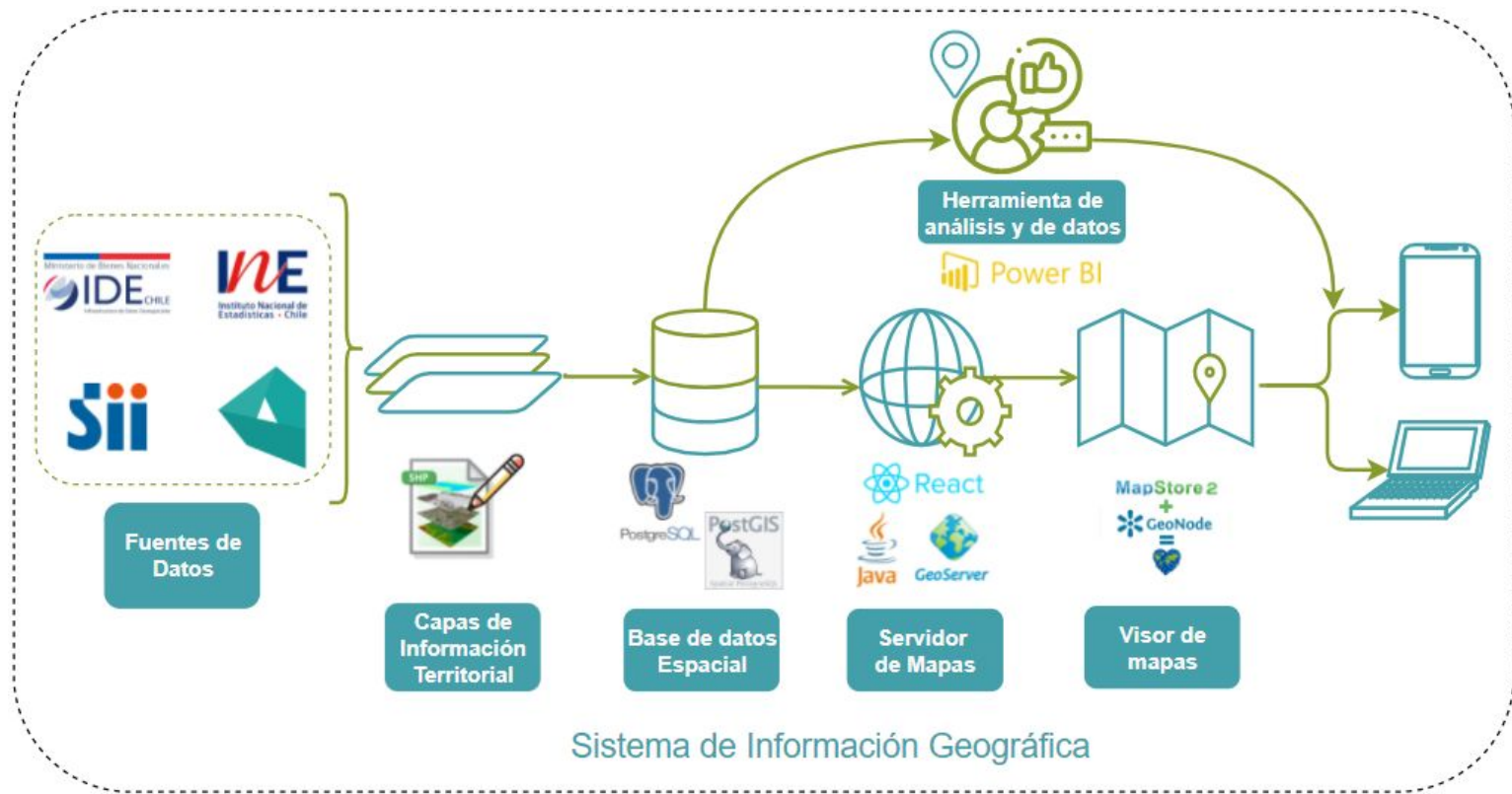


Fuente: Elaboración propia - CITIAPS USACH



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Arquitectura de un SIG





Arquitectura de un SIG

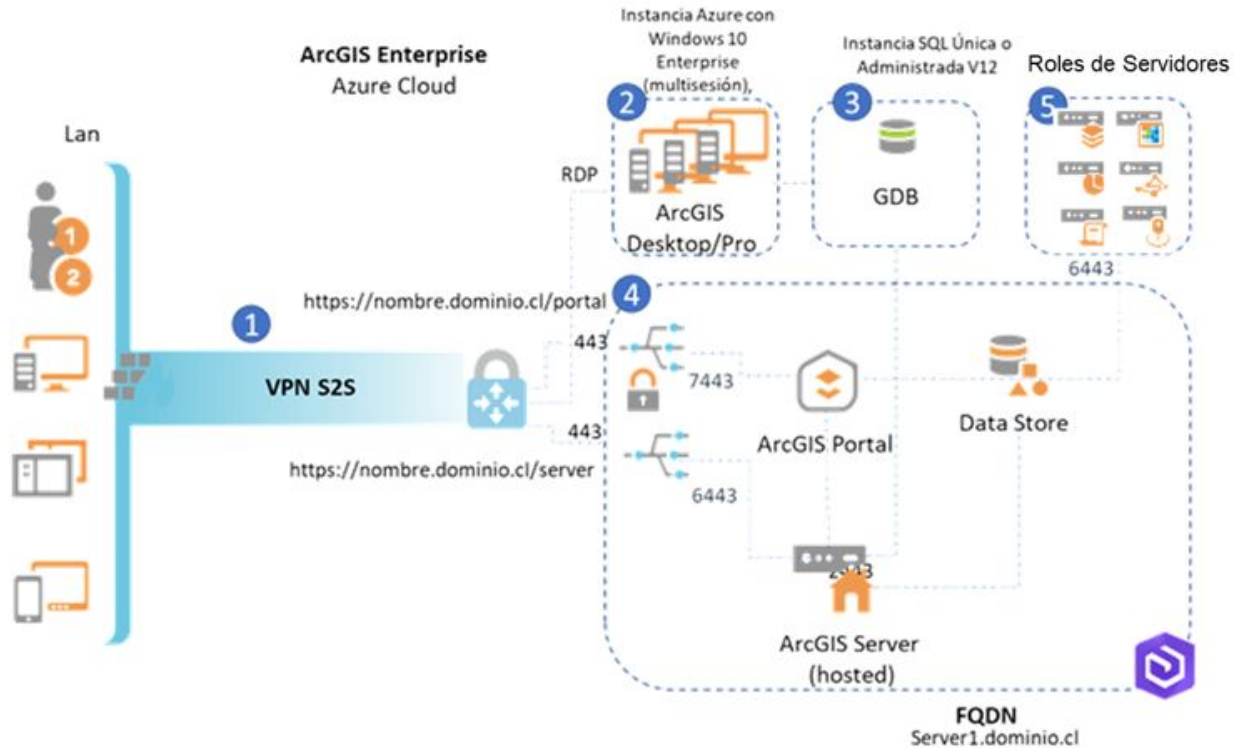




Diagrama conceptual de un SIG

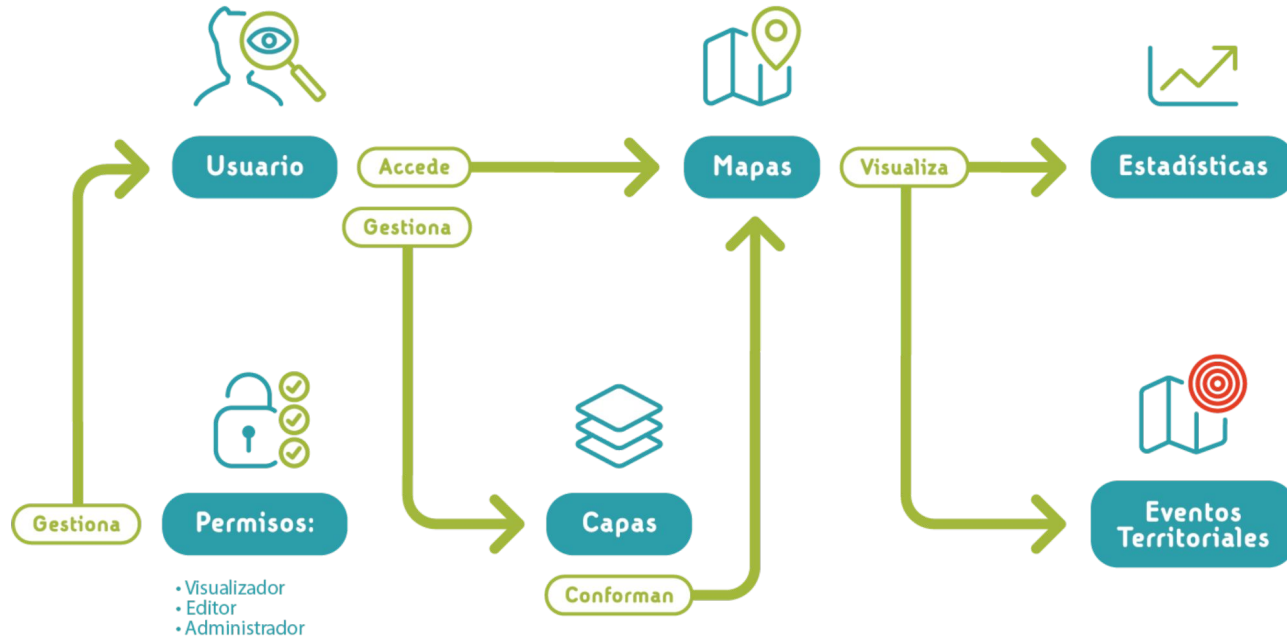
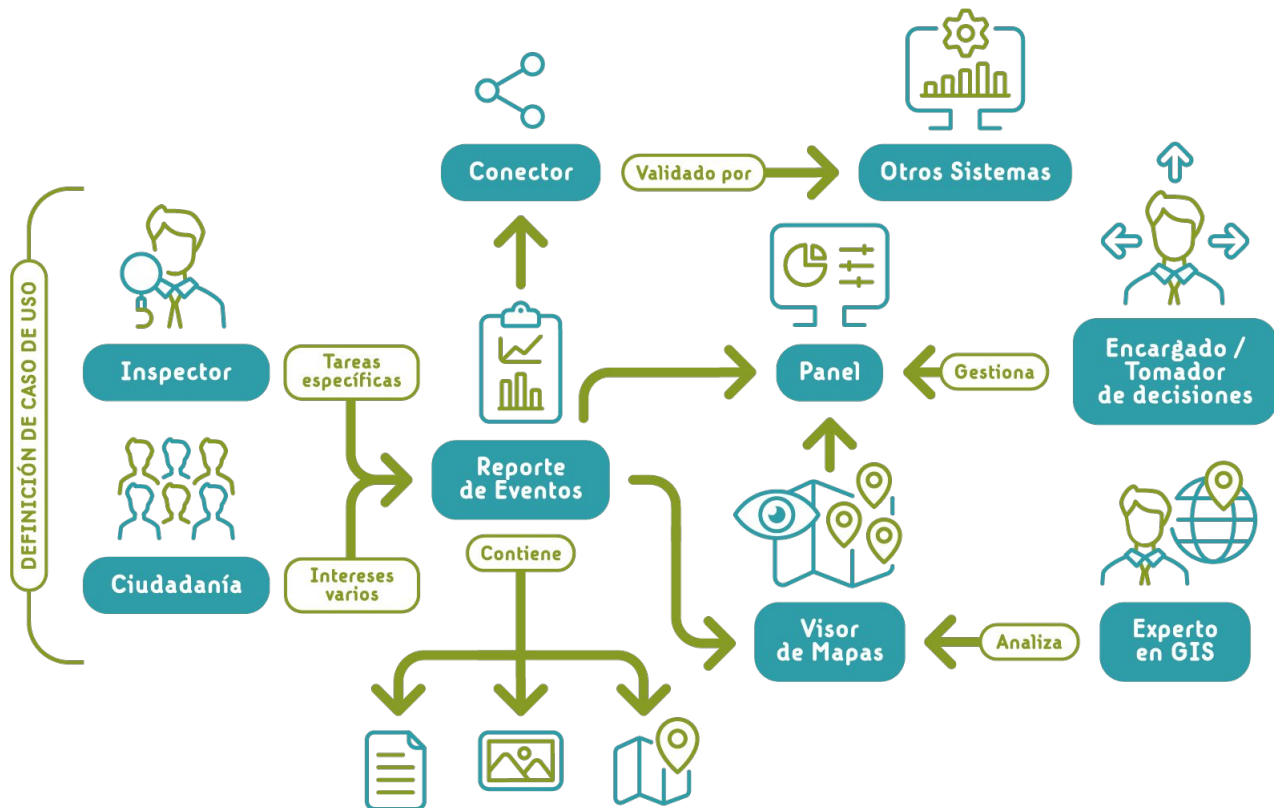




Diagrama conceptual de un SIG





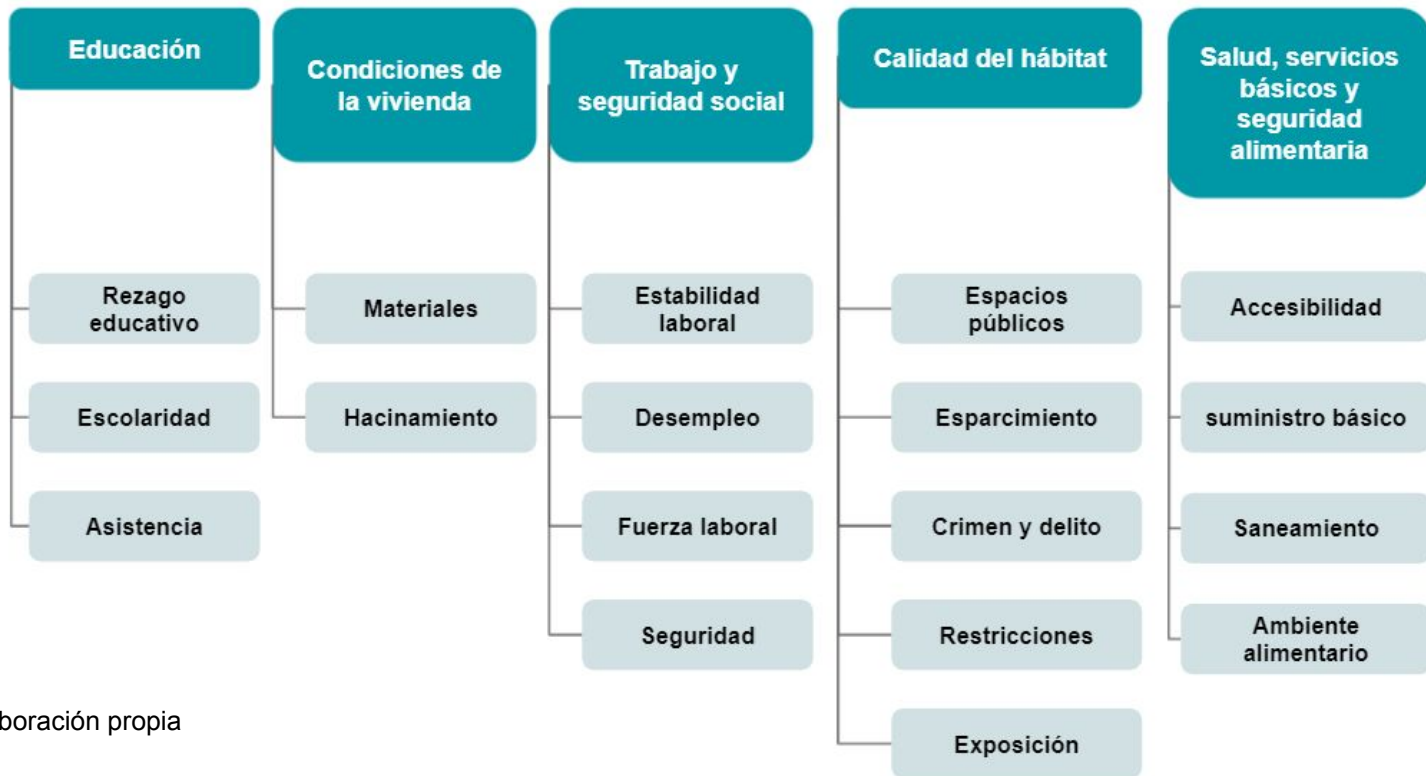
Para reflexionar

1. ¿Cómo se relaciona el ICG con respecto a su aporte en la arquitectura SIG y la soluciones que soporta?
2. Realizar cuadro comparativo de las arquitecturas ESRI y Open Source desde un enfoque económico, técnico y necesidades del usuario final (Realizar una vez visto el contenido desde cátedra y laboratorio)
3. Realizar un listado de actividades y procesos considerados dentro del diseño, e implementación de un SIG en las cuales cree que aportaría usted desde su disciplina, considerando roles y otras especialidades de trabajo.



Para reflexionar

Enfoque multidimensional



Fuente: Elaboración propia



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Bases de Datos



Objetivo

El presente contenido tiene por finalidad introducir al estudiante en el uso de bases de datos, como entorno de trabajo de la información geoespacial, en donde se desarrollará la estructuración de ésta como modelo de datos, junto con el establecimiento de reglas y comportamientos, que definirán su funcionalidad dentro de los aplicativos. Además se profundizará sobre el lenguaje SQL y su uso dentro de las bases de datos.

Contenido:

1. BD: Open Source vs ESRI (GDB)
2. SQL
3. Fundamentos del Lenguaje
4. Consultas SQL
5. Modelos de datos
6. Dominios y Subtipos
7. Topología
8. Relaciones
9. Vistas
10. Versionamiento y respaldo
11. Esquema de usuarios BD
12. Esquema XML
13. Esquema SQL y Backup

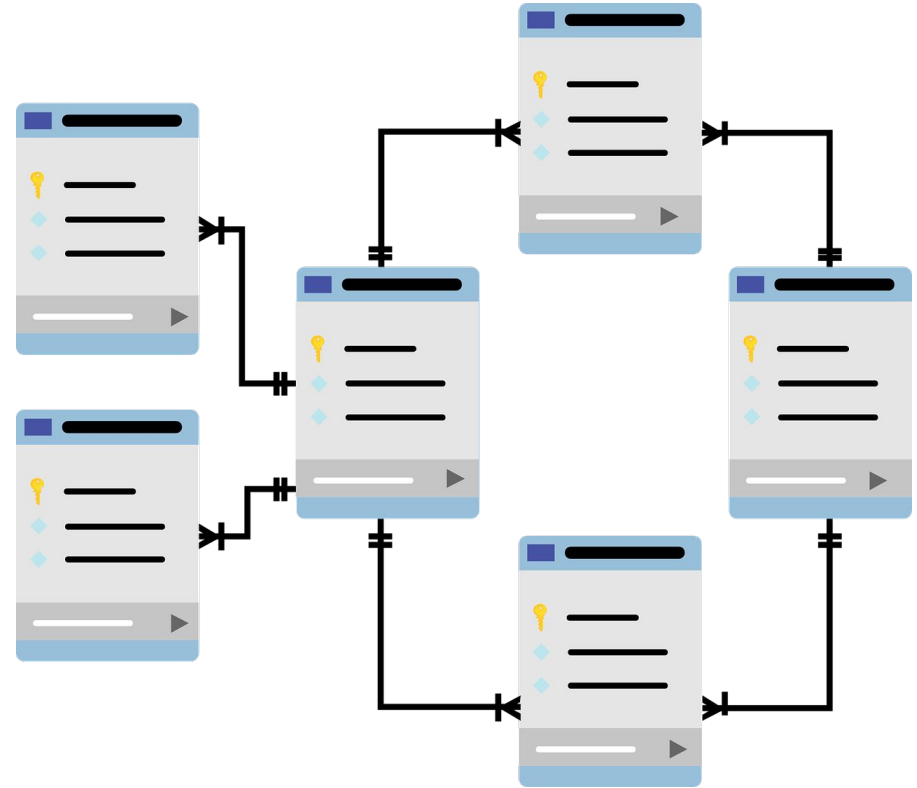


Introducción a las BD

Una **base de datos** es una colección de tablas organizadas en forma lógica por un DBMS.

Una **tabla** es un conjunto de **registros** relacionados

Un **registro** corresponde a una observación, y queda caracterizado por **atributos**

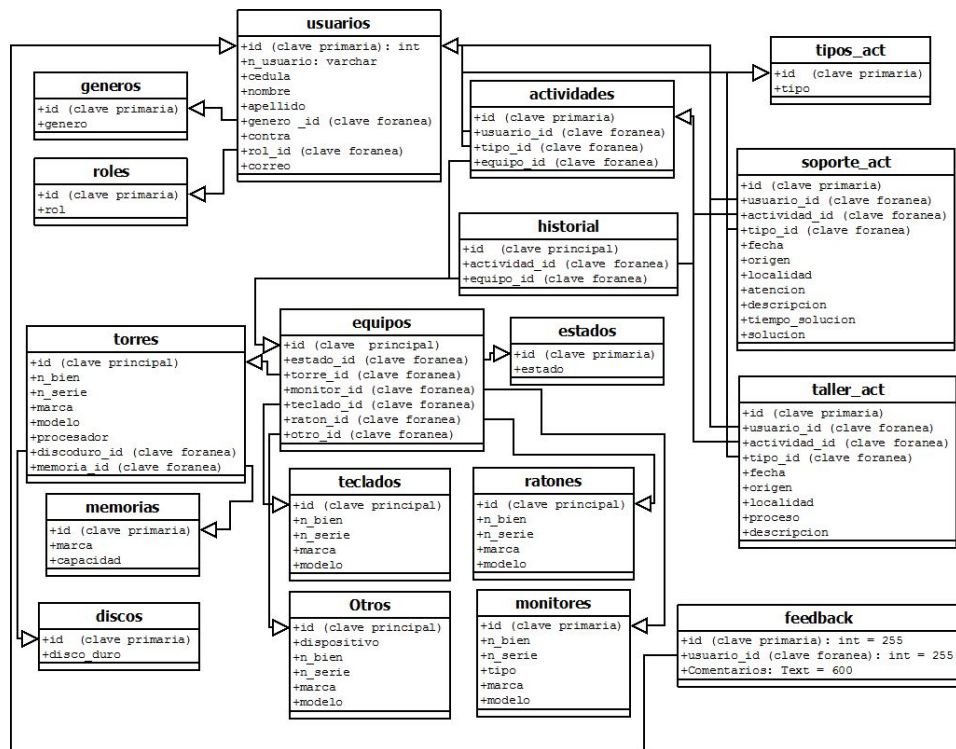




UNIVERSIDAD
DE SANTIAGO
DE CHILE

Tipos de bases de datos

Modelo de Datos



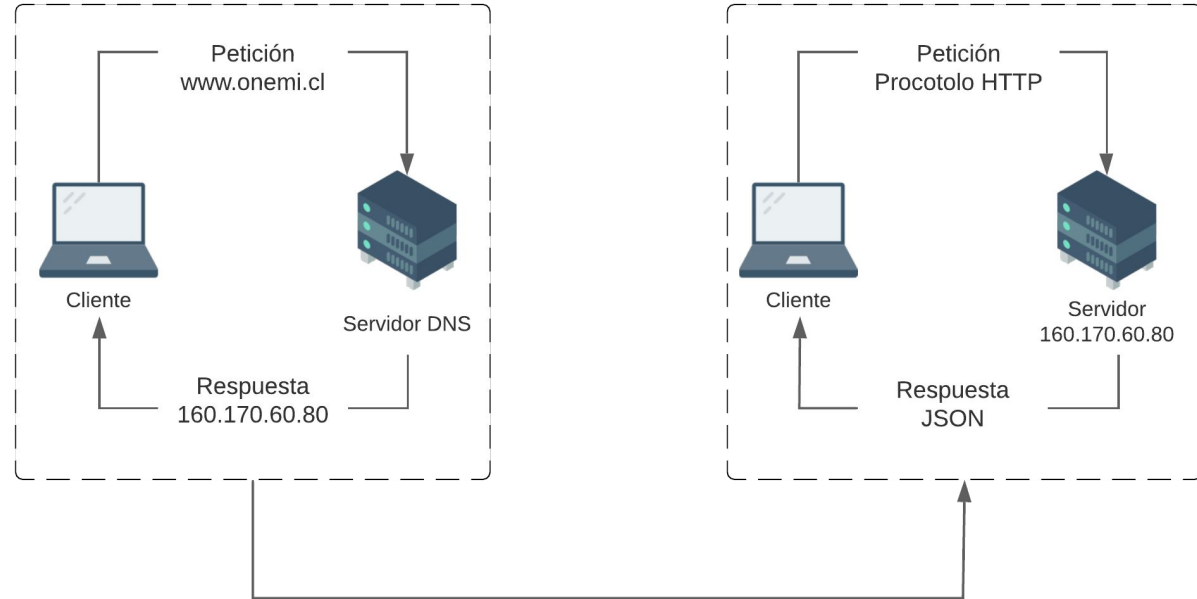
SGBD	Características	Ventajas	Inconvenientes
ACCESS	Perteneciente a Microsoft. Es muy gráfico. Métodos simples y directos, con formularios, para trabajar con la información.	Asequible para personas con poco manejo con las bases de datos. Crea varias vistas para una misma información.	No es multiplataforma. No funciona con bases de datos grandes, tanto para registros como para usuarios.
SQLite	Los tipos de datos se asignan a valores individuales y no a la columna como la mayoría de los SGBD.	Multiplataforma. No requiere configuración. Acceso muy rápido. No requiere servidor.	El dinamismo de los datos hace que no se portable a otras bases de datos. Falta de clave foráneas.
SQL SERVER	Software propietario. El lenguaje es TSQL.	Multiplataforma, aunque pertenezca a Microsoft. Transacciones.	Utiliza mucha RAM. Tamaño de página fijo y pequeño. Relación calidad/precio inferior a Oracle.
MYSQL	Pertenece a Oracle. Licencia GPL/Licencia comercial.	Agrupación de transacciones. Distintos motores de almacenamiento. Instalación sencilla.	No tiene soporte. Capacidad limitada.
POSTGRESQL	Tiene la extensión POSTGIS para bases de datos espaciales.	Código abierto y gratuito, multiplataforma. Gran volumen de datos. Transacciones, disparadores y afirmaciones.	Respuesta lenta. Requiere hardware. No es intuitivo.
ORACLE	Dispone de su propio lenguaje PL/SQL. Soporta bases de datos de gran tamaño.	Es el más usado a nivel mundial. Multiplataforma. Es intuitiva y fácil de usar.	Precio muy elevado. Elevado coste de la información, tratado por trabajadores formados por Oracle.

Fuente: Stackoverflow



Modelo de cliente servidor

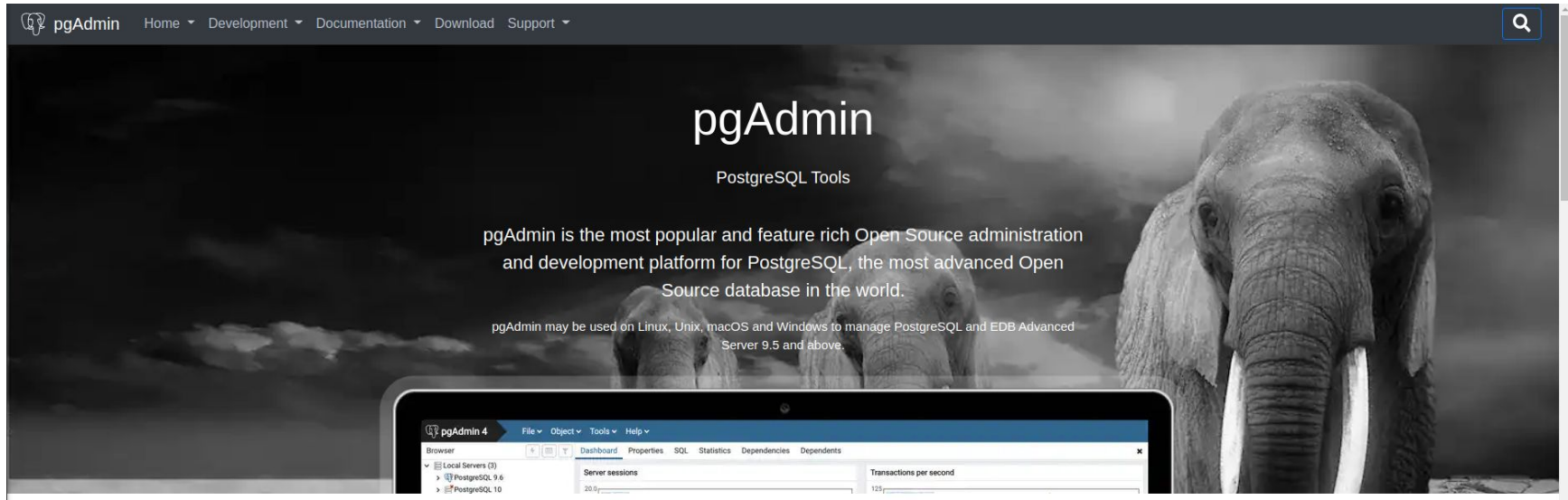
Cliente
Petición
Servidor
DNS
Puerto
Respuesta





pgAdmin

Herramienta que nos permite **administrar** y **desarrollar** en una base de datos PostgreSQL. De uso gratuito, levanta un pequeño servidor al cual nos conectamos una vez instalado.

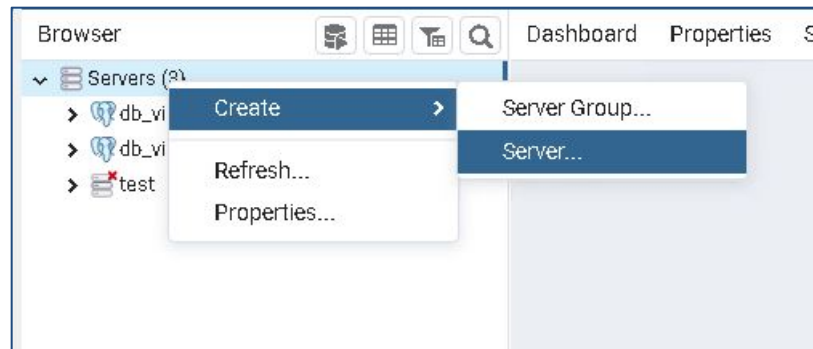
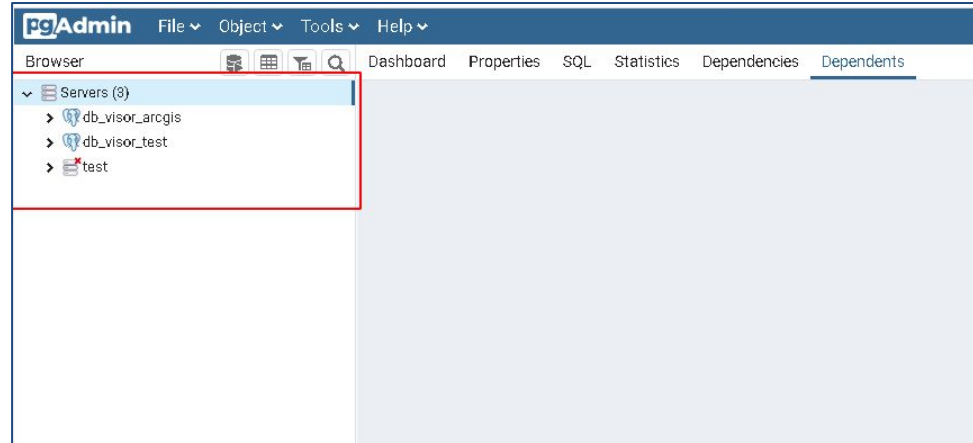


Fuente: pgAdmin



pgAdmin

- A la derecha encontramos las conexiones a las bases de datos ya establecidas.
- Para crear una nueva conexión solo se debe hacer clic derecho en “Server” y darle a Create -> Server.

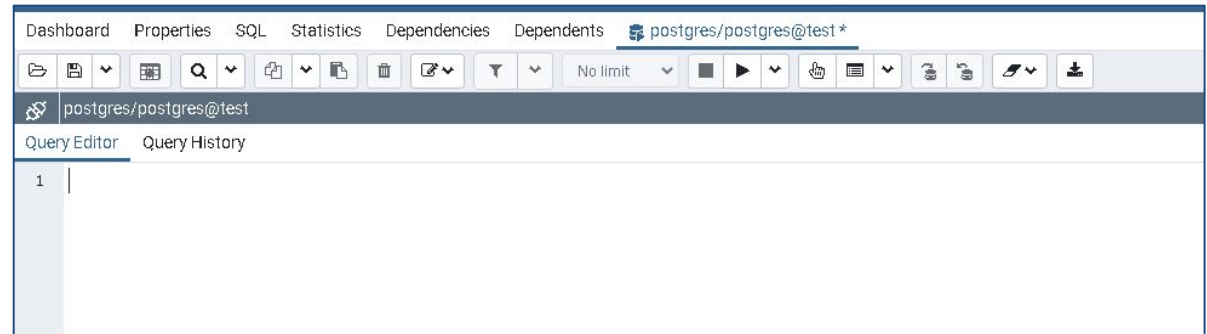
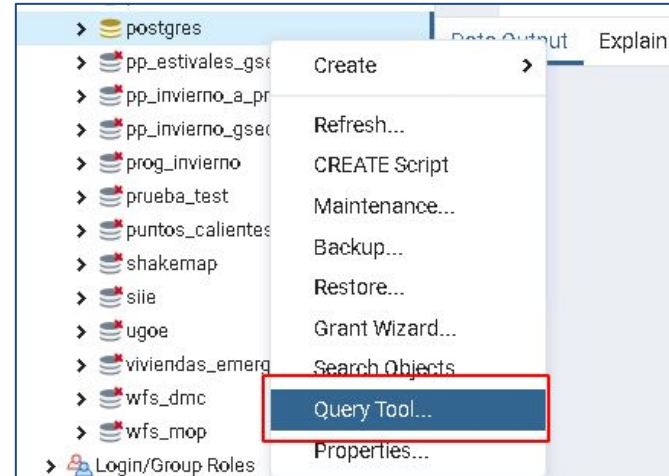




pgAdmin

Para ejecutar comandos SQL, nos debemos ir a la base de datos de donde queremos ejecutar el *script*, y darle a “Query Tool”.

Esto abre una pestaña en donde podemos interactuar con la base de datos mediante SQL.





Diseño de bases de datos

Diseño conceptual

Entidades y relaciones
existentes entre ellas

Diseño lógico

Proceso de normalización de
las tablas para evitar
duplicado de información

Diseño físico

Escribir código en SQL para
plasmar el diseño en el motor
de bases de datos



Diagrama conceptual

Mientras diseñamos nuestra base de datos utilizamos el diagrama conceptual para plasmar las entidades y sus relaciones.

- Entidad: Representa un “objeto” independiente.
- Atributos: Características que posee la entidad.
- Relación o Cardinalidad: Conjunto de relación de las cuales participan las entidades. Estas pueden ser:
 - Uno a Uno (1...1)
 - Uno a Varios (1...m)
 - Varios a Uno (m...1)
 - Varios a Varios (m...m)

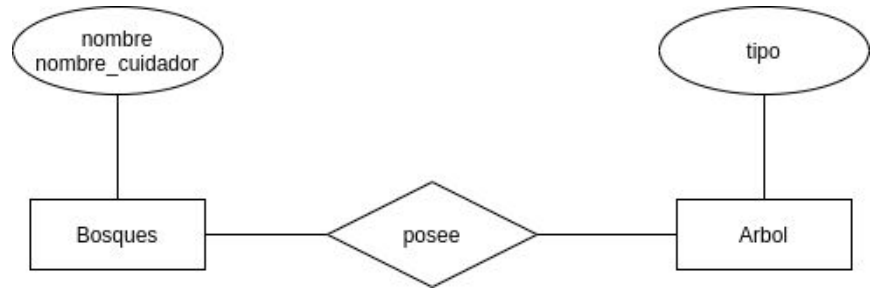




Diagrama lógico

Durante esta fase, debemos crear las tablas que representan nuestra entidades encontradas. Establecemos las relaciones y añadimos las llaves correspondientes para cumplir con el diseño.

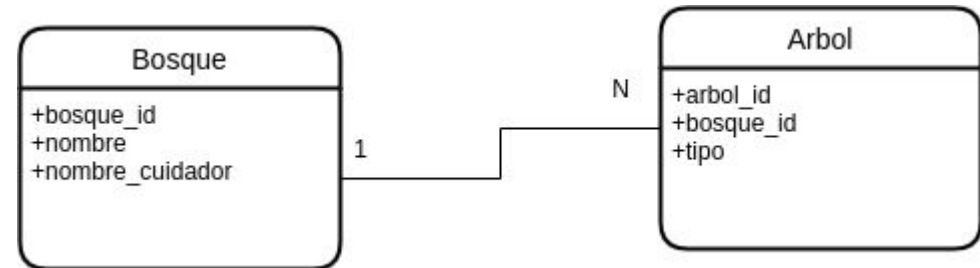




Diagrama físico

Es nuestra última fase e implica el crear y ejecutar nuestro código SQL para crear la base de datos, tablas, columnas, relaciones, etc.

```
CREATE TABLE bosques (  
    bosque_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(250) NOT NULL,  
    nombre_cuidador VARCHAR(250) NOT NULL,  
  
);  
  
CREATE TABLE arbol (  
    arbol_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    bosque_id INT NOT NULL REFERENCES bosques(bosque_id),  
    tipo VARCHAR(250) NOT NULL  
);
```



Respaldo y restauración

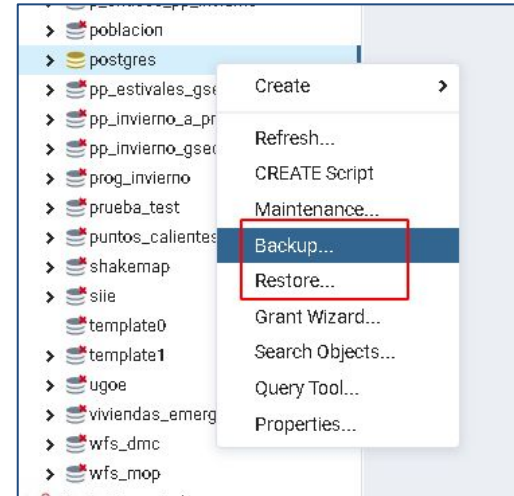
Una de las funciones más útiles de las bases de datos, es su capacidad de replicarse.

Esto es, tomar los datos en una base de datos y llevarlos a otra base de datos e incluso a otro servidor.

Para realizar estas operaciones tenemos dos alternativas comunes:

- Utilizar las herramientas que vienen integradas en la instalación de PostgreSQL.
- Utilizar herramientas externas como PgAdmin.

Primero nos vamos a la base de datos que deseamos respaldar/restaurar:





Respaldo y restauración

Si escogemos la opción de respaldo o “Backup”, nos abre un modal, en la cual nos pide diferentes datos, por ejemplo, dónde y cómo se llamará el archivo de backup, el formato, etc.

The screenshot shows the 'Backup (Database: postgres)' dialog box in pgAdmin. It has two tabs: 'General' and 'Dump options'. The 'General' tab is active, showing fields for 'Filename', 'Format' (set to 'Custom'), 'Compression ratio', 'Encoding' (set to 'Select an item...'), 'Number of jobs', and 'Role name' (set to 'Select an item...'). At the bottom, there are buttons for 'i', '?', 'Cancel', and 'Backup'.



Respaldo y restauración

Si escogemos la opción de restauración o “Restore”, nos abre un modal, en la cual nos pide diferentes datos, por ejemplo, dónde y cómo se encuentra el archivo de backup, el formato, etc.

Restore (Database: postgres)

General Restore options

Format Custom or tar

Filename ...

Number of jobs

Role name Select an item...

Cancel Restore



UNIVERSIDAD
DE SANTIAGO
DE CHILE

SQL

(Conceptos básicos genéricos ó con PostgreSQL)



SQL (Structured Query Language)

Nos permite realizar operaciones de consulta, actualización, creación y eliminación de datos de una base de datos relacional.

SQL contiene su propio conjunto de palabras claves, entre ellas están:

- INSERT
- UPDATE
- CREATE
- DELETE
- WHERE
- FROM

Entre muchas más.



SQL - Notación

Las sentencias SQL **siempre** deben terminar con “;”, esto debido a que podemos escribir muchas sentencias en un solo archivo .sql, por lo que el intérprete debe ser capaz de diferenciar entre un comando y otro.

```
SELECT * FROM temblores WHERE magnitud > 5.0;
```

Es importante notar que los comandos son “case-insensitive”, es decir, no importa si escribimos SELECT o select. Lo que siempre debe estar bien escrito son los atributos o nombres de las tablas de la base de datos a la cual estamos consultando.



SQL - Comentarios

Para dejar comentarios en nuestros *scripts* utilizamos la notación “--” para comentarios de una sola línea:

```
-- Seleccionamos todos los datos de la tabla amenazas
```

```
/* */
```

```
SELECT * FROM amenazas;
```

Para comentarios de más de una línea se utiliza “/* Comentario de muchas líneas */”.



SQL - Crear base de datos

Para crear una base de datos utilizamos:

```
CREATE DATABASE nombre_base_datos;
```

Además, es posible añadir más configuraciones al momento de crear la base de datos, por ejemplo, asignar un dueño de la base de datos con:

```
CREATE DATABASE nombre_base_datos WITH OWNER = nombre_rol;
```



SQL - Crear tabla en base de datos

Para crear una tabla en una base de datos utilizamos:

```
CREATE TABLE servicios_arcgis (  
  
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  
    nombre_servicio VARCHAR(250) NOT NULL,  
  
    url_servicio VARCHAR(250) NOT NULL  
  
);
```

Así, en un solo *script* podemos crear todas las tablas que necesitemos.



SQL - Insertar datos

Supongamos una tabla con las siguientes columnas:

```
CREATE TABLE servicios_arcgis (  
  
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  
    nombre_servicio VARCHAR(250) NOT NULL,  
  
    url_servicio VARCHAR(250) NOT NULL  
  
);
```

Pregunta: ¿Qué tipos de datos conoce?



SQL - Insertar datos

Para insertar datos en la tabla anterior utilizamos la siguiente sentencia:

```
INSERT INTO servicios_arcgis (nombre_servicio,url_servicio)
VALUES ('Poblacion_Chilena',
       https://visor.cl/arcgis/rest/services/Poblacion\_Chilena/MapServer);
```

Notar que no ingresamos el id, esto debido a las características de que presenta esa columna, ya que se auto incrementa automáticamente, así tenemos un identificador único por fila en la tabla.



SQL - Seleccionar datos

Para seleccionar datos en la tabla anterior utilizamos la siguiente sentencia:

```
SELECT nombre_columna1, nombre_columna2, nombre_columnaN FROM nombre_tabla;
```

Para seleccionar todas las columnas:

```
SELECT * FROM servicios_arcgis;
```

Para seleccionar solo las urls, utilizamos:

```
SELECT url_servicio FROM servicios_arcgis;
```

Para seleccionar todo, pero que el id sea cierto valor, utilizamos:

```
SELECT * FROM servicios_arcgis WHERE id = 17;
```



SQL - Seleccionar datos

Para seleccionar datos en la tabla anterior utilizamos la siguiente sentencia:

```
SELECT nombre_columna1, nombre_columna2, nombre_columnaN FROM nombre_tabla;
```

Para seleccionar todas las columnas:

```
SELECT * FROM servicios_arcgis;
```

Para seleccionar solo las urls, utilizamos:

```
SELECT url_servicio, FROM servicios_arcgis;
```

Para seleccionar todo, pero que el id sea cierto valor, utilizamos:

```
SELECT * FROM servicios_arcgis WHERE id = 17;
```



SQL - Actualizar datos

Para actualizar datos en la tabla anterior utilizamos la siguiente sentencia:

```
UPDATE nombre_tabla  
SET nombre_columna1 = valor1, nombre_columna2 = valor2,...  
WHERE condicion;
```

Por ejemplo, para actualizar el nombre que ingresamos en la tabla “servicios_arcgis”:

```
UPDATE servicios_arcgis  
SET nombre_servicio = "Poblacion Nacional"  
WHERE id=1;
```




SQL - Borrar datos

Para borrar datos en la tabla anterior utilizamos la siguiente sentencia:

```
DELETE FROM nombre_tabla WHERE condicion;
```

Por ejemplo:

```
DELETE FROM servicios_arcgis WHERE id=1;
```

Importante: Si no se especifica la condición “WHERE”, **todos los datos** de la tabla serán borrados permanentemente. Se recomienda no borrar datos de esta manera.



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Caso Práctico con SQLITE y DB Browser



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Antes de comenzar

Sitio Oficial: [DB Browser for SQLite \(sqlitebrowser.org\)](http://sqlitebrowser.org)

Spatialite: [Index of /gaia-sins/windows-bin-amd64-prev \(gaia-gis.it\)](http://index.of/gaia-sins/windows-bin-amd64-prev)

Cheat Sheet: [SQLite Cheat Sheet \(sqlitetutorial.net\)](http://sqlitetutorial.net)

Datos para practicar: [SQLite Sample Database And Its Diagram \(in PDF format\) \(sqlitetutorial.net\)](http://sqlitetutorial.net)

Actividad: Generar base de datos censal a nivel de zona local considerando cantidad de personas y como mínimo 6 variables



Caso Práctico

Con los datos para practicar:

¿Cuántos Álbumes hay por artista? considerando:

1. Orden descendente
2. Los 3 con más producciones (Discos producidos)
3. Añadir nombres de artistas
4. Estilo Rock (Ayuda: desagregue la consulta)
 - 4.1. Unir tablas
 - 4.2. Solicitar campos
 - 4.3. Agrupando, ordenando y limitando
 - 4.4. Añadir filtro de estilo
 - 4.5. Que aparezca el nombre del estilo y parametrizar
5. Seleccionar los 10 estilos con más producciones, donde también, aparezca el nombre del artista (también considerar ratio entendido como producciones / total producciones * 100)

Actividad: ¿Cuál es el playlist con mayor duración?

¿Cuál es el playlist de mayor valor?

¿Quién es el mejor cliente y vendedor?

¿Qué grupo ha tenido la mejor venta?

¿Cuáles son los 3 países con mayores ventas por estilo?



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Consultas Espaciales con SQL



Resumen de Funciones

Spatial relationships and measurements (topología)

- ST_Centroid: returns the the geometric center of a geometry
- ST_ClusterKmeans: returns integer ID for the cluster each input geometry is in.
- ST_Contains: return TRUE if and only if no points of B lie in the exterior of A
- ST_Crosses: returns TRUE if the supplied geometries have some, but not all, interiors points in common
- ST_Covers: returns TRUE(1) if no point in geometry B is outside geometry A
- ST_CoveredBy: returns TRUE(1) if no point in Geometry A is outside Geometry B.
- ST_LineCrossingDirection: given 2 linestrings, returns a number between -3 and 3 denoting what kind of crossing behaviour. 0 is no crossing
- ST_Disjoint: returns TRUE if geometries do not spatially intersect
- ST_Distance: for geometry returns cartesian distance between two geometries in projected units. for geography type returns minimum geodesic distance in meters.
- ST_MaxDistance: al revés de st_distance
- ST_Intersects: returns TRUE if the geometry spatially intersects
- ST_Length: return the length
- ST_Perimeter: returns length measurement of the boundary
- ST_Touches: returns TRUE if the geometries have at least one point in common, but their interiors do not intersect
- ST_Within: returns true if the geometry A is completely inside geometry B



Resumen de Funciones

Geometry Processing

- ST_Buffer: creates a buffer within the given radius, geom field has to be in UTM
- ST_Difference: erase arcgis
- ST_Intersection: creates intersection between two tables, if polyline or polygon, requires topology validation.
- ST_Dump: returns a set of geometry_dump rows, that make up a geometry d1.
- ST_MemUnion: dissolve (demora más pero consume menos)
- ST_Union: dissolve (consume más recursos)
- ST_RemoveRepeatedPoints: returns a version of the given geometry with duplicated points removed
- ST_IsValid: Returns TRUE if the geometry is well formed.
- ST_Y: Returns the y coordinate of the point, or NULL if not available input must be a point.
- ST_X: Same as Y.
- ST_AsLatLonText: returns the degree, minutes, seconds representation of the given point
- ST_Area: returns the area of the surface if its a polygon
- ST_Subdivide: Returns a set of geometry where no geometry in the set has more than the specified number of vertices.
- ST_MakeValid: Attempts to make an invalid geometry valid without losing vertices.
- ST_Simplify: simplifies geometry based on given tolerance
- ST_SimplifyPreserveTopology: simplifies geometry based on a given tolerance preserving its topology (prevents oversimplification).
- ST_CollectionExtract: Given a (multi)geometry, return a (multi)geometry consisting only of elements of the specified type. 1 == POINT, 2 == LINESTRING, 3 == POLYGON.



Ejemplos

Agrupación de puntos

```
--crea una tabla si no existe
create table if not exists esquema.tablaFinal as
select round(campo/10000) as nombreCampo1,
       sum(campo) as nombreCampo2,
       st_setsrid(st_makepoint(
         avg(st_x(st_centroid(the_geom))),
         avg(st_y(st_centroid(the_geom)))
       ), 4326) as the_geom
from esquema.tabla
group by round(manzent/10000000);

CREATE INDEX if not exists tablaFinal_gix on
esquema.tablaFinal USING GIST (the_geom);
```

Intersección de puntos con polígonos (notar que sólo utiliza validación topológica)

```
create table if not exists esquema.tablaFinal as
select *
from esquema.capa1 as shp1
join esquema.capa2 as shp2
on ST_INTERSECTS(shp1.the_geom, shp2.the_geom);
```




Ejemplos

Intersección entre polígonos/polilíneas (se pueden mezclar) (notar que necesita validación topológica y hacer la intersección vectorial)

```
create table if not exists esquema.tablaFinal as
select shp2.nombreCampo as nombre,
ST_INTERSECTION(ST_MakeValid(shp1.the_geom),ST_MakeValid(shp2.the_geom)) as the_geom
from esquema.capa1 as shp1
join esquema.capa2 as shp2
on ST_INTERSECTS(shp1.the_geom, shp2.the_geom);
```

Enlaces

- [1] SQL basics; <https://www.dataquest.io/blog/sql-basics/>
- [2] postGis; <https://postgis.net/docs/manual-2.0/>
- [3] pgRouting; <https://docs.pgrouting.org/2.4/en/index.html>
- [4] Spatialite; [Spatialite SQL functions reference list \(gaia-gis.it\)](#)



Actividad: Generar Traza de Líneas en el ejemplo que sigue (Caso de Aplicación Real)



Caso de Aplicación Real

```
/*--subdividir temperatura (3s)*/  
create table if not exists multi_amen.sub_temp as  
select shp."GRIDCODE" as ctemp,  
st_subdivide(shp.the_geom) as the_geom  
from var_amb.valid_temp as shp;
```

```
CREATE INDEX if not exists sub_temp_gix on  
multi_amen.sub_temp USING GIST (the_geom);
```

```
/*--subdividir humedad (5s)*/  
create table if not exists multi_amen.sub_hum as  
select shp."GRIDCODE" as chum,  
st_subdivide(shp.the_geom) as the_geom  
from var_amb.valid_hum as shp;
```

```
CREATE INDEX if not exists sub_hum_gix on  
multi_amen.sub_hum USING GIST (the_geom);
```



Caso de Aplicación Real

```
/*-- intersectar temperatura y humedad segun indice de calor (8s)
-- 1: bajo [1]
-- 2: precaucion (medio) [2]
-- 3: extrema precaucion (medio) [2]
-- 4: Peligro (medio alto) [3]
-- 5: Peligro extremo (alto) [3]*/
create table if not exists multi_amen.heatwave as
select
case
    when (shp1.ctemp <= 6 and shp2.chum <= 4) then 1
    when (shp1.ctemp <= 6 and shp2.chum >= 5) then 2
    else 3
end as code, st_intersection(st_makevalid(shp1.the_geom), st_makevalid(shp2.the_geom)) the_geom
from multi_amen.sub_temp as shp1
join multi_amen.sub_hum as shp2
on st_intersects(shp1.the_geom, shp2.the_geom);

CREATE INDEX if not exists heatwave_gix on
multi_amen.heatwave USING GIST (the_geom);
```



Caso de Aplicación Real

```
/*--Subdividir pri (40s)
-- 1: bajo pri
-- 2: medio pri
-- 3: alto pri*/
create table multi_amen.sub_pri as
select
case
  when shp."GRIDCODE" <= 30 then 1
  when (shp."GRIDCODE" > 30 and shp."GRIDCODE" <= 60) then 2
  when (shp."GRIDCODE" > 60) then 3
end as cpri, st_subdivide(st_makevalid(shp.the_geom)) as the_geom
from var_amb.valid_pri shp;
```

```
CREATE INDEX if not exists sub_pri_gix on
multi_amen.sub_pri USING GIST (the_geom);
```

```
/*--Subdividir heatwave (1s)*/
create table multi_amen.sub_heatwave as
select shp.code, st_subdivide(shp.the_geom) as the_geom
from multi_amen.heatwave as shp;
```

```
CREATE INDEX if not exists sub_heatwave_gix on
multi_amen.sub_heatwave USING GIST (the_geom);
```



Caso de Aplicación Real

```
/*-- Multiamenaza (25s)*/  
create table if not exists multi_amen.amenaza as  
select  
case  
    when (shp1.code = 1 and shp2.cpri = 1) then 1  
    when (shp1.code = 2 and shp2.cpri = 2) then 3  
    when (shp1.code = 3 and shp2.cpri = 3) then 5  
    when (shp1.code = 1 and shp2.cpri = 2) then 2  
    when (shp1.code = 1 and shp2.cpri = 3) then 3  
    when (shp1.code = 2 and shp2.cpri = 3) then 4  
    when (shp1.code = 2 and shp2.cpri = 1) then 2  
    when (shp1.code = 3 and shp2.cpri = 2) then 5  
    when (shp1.code = 3 and shp2.cpri = 1) then 5  
end as amen, st_intersection(shp1.the_geom, shp2.the_geom) as the_geom  
from multi_amen.sub_heatwave as shp1  
join multi_amen.sub_pri as shp2  
on st_intersects(shp1.the_geom, shp2.the_geom);  
  
CREATE INDEX if not exists amenaza_gix on  
multi_amen.amenaza USING GIST (the_geom);
```



Caso de Aplicación Real

```
/*--Raster de amenaza (20s)*/
create table if not exists multi_amen.raster_amen as
select shp.amen, st_asraster(shp.the_geom,100,100,'32BF', shp.amen) as the_geom
from multi_amen.amenaza as shp;

/*--Vectorizar amenaza (40s)*/
create table multi_amen.amen_pri_hw as
SELECT dp.val as amen, dp.geom as the_geom
FROM multi_amen.raster_amen, LATERAL ST_DumpAsPolygons(the_geom) AS dp
ORDER BY val;

CREATE INDEX if not exists amen_pri_hw_gix on
multi_amen.amen_pri_hw USING GIST (the_geom);

/*-- arreglar srid y tipo geom*/
ALTER TABLE multi_amen.amen_pri_hw
  ALTER COLUMN the_geom TYPE geometry(MULTIPOLYGON, 4326)
  USING st_multi(ST_SetSRID(the_geom,4326));

/*-- Eliminar lo que no sirve*/
DROP TABLE IF EXISTS multi_amen.sub_temp;
DROP TABLE IF EXISTS multi_amen.sub_pri;
DROP TABLE IF EXISTS multi_amen.raster;
DROP TABLE IF EXISTS multi_amen.class_temp_hum;
DROP TABLE IF EXISTS multi_amen.raster_temp;
```



UNIVERSIDAD
DE SANTIAGO
DE CHILE

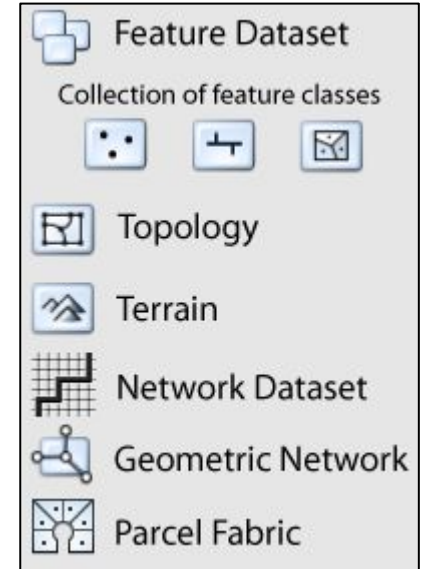
GEODATABASE



Tipos de datasets y datos específicos en GDBs

Dataset de entidad

- Colección de clases de entidad relacionadas que comparten un sistema de coordenadas común.
- Se utilizan para integrar espacial o temáticamente clases de entidad relacionadas. Su propósito primario es organizar clases de entidad relacionadas en un dataset común para generar:
 - Una topología
 - Un dataset de red
 - Un dataset de terreno
 - Una red geométrica.





Tipos de datasets y datos específicos en GDBs

The screenshot shows a file explorer for a GDB file named 'GDB_EJEMPLO.gdb'. It lists several datasets with their respective icons and counts. To the right of the list, there are five numbered red annotations explaining the types of datasets shown.

Dataset Name	Count	Type
Comuna_Santiago	1	1.- Feature class
Educación_ND	4	2.- Relación entre tabla y feature class
Educación_ND_Junctions	1	3.- Validación topológica
Establecimientos	1	4.- Network o red
Estaciones_Metro	1	5.- Tabla de atributos no espacial
GSE_2002	2	
Lineas_Metro	1	
Manzanas_Comuna_Santiago	1	
Paraderos_Transantiago	1	
Población_2002	2	
Red_Vial_Santiago	1	
SIMCE_2011	2	
SIMCE_2012	2	
SIMCE_2013	2	
Topologia	3	
Viajes_Educación	1	
Establecimientos_ATTACH	5	
Establecimientos_ATTACHREL	2	
GSE	5	
POB_MANZ	5	

Como estructura básica solamente se requiere un feature class dentro de un feature dataset



Tipos de datasets y datos específicos en GDBs

CreateFeatureDataset_management(out_dataset_path, out_name, {spatial_reference})

#Importación de bibliotecas

```
import arcpy  
from arcpy import env
```

Definir espacio de trabajo

```
env.workspace = "C:/data"
```

Definir variables locales

```
out_dataset_path = "C:/output/HabitatAnalysis.gdb"  
out_name = "analysisresults"
```

Crear referencia espacial desde el archivo prj de una capa. También se puede integrar uno por código

```
sr = arcpy.SpatialReference("C:/data/studyarea.prj")
```

Crear alguna GDB para almacenar dataset. Necesario de no existir una GDB

```
#arcpy.CreateFileGDB_management("C:/output", "HabitatAnalysis.gdb")
```

Crear Dataset

```
arcpy.CreateFeatureDataset_management(out_dataset_path, out_name, sr)
```



Tipos de datasets y datos específicos en GDBs

```
CreateFeatureclass_management(out_path, out_name, {geometry_type}, {template}, {has_m},  
{has_z}, {spatial_reference}, {config_keyword}, {spatial_grid_1}, {spatial_grid_2}, {spatial_grid_3})
```

Importación de bibliotecas

```
import arcpy  
from arcpy import env
```

Definir espacio de trabajo

```
env.workspace = "C:/data"
```

Definir variables locales

```
out_path = "C:/output"  
out_name = "habitatareas.shp"  
geometry_type = "POLYGON"  
template = "study_quads.shp"  
has_m = "DISABLED"  
has_z = "DISABLED"
```

Obtener referencia espacial desde la función describir de una SHP. ** [Explorar Función Describe](#) **

```
spatial_reference = arcpy.Describe("C:/workspace/studyarea.shp").spatialReference
```

Crear Feature Class

```
arcpy.CreateFeatureclass_management(out_path, out_name, geometry_type, template, has_m, has_z, spatial_reference)
```



Tipos de datasets y datos específicos en GDBs

¿Qué otros tipos de datos y herramientas existen para la GDB que puedan ser utilidad? Explorar dentro del catálogo:

- Data Management Tools
 - Attachments
 - Distributed Geodatabase
 - Domains
 - Feature Class
 - Features
 - Fields
 - Geodatabase Administration
 - Joins
 - Relationship Class
 - Subtypes
 - Table
 - Topology
 - Versions
 - Workspace

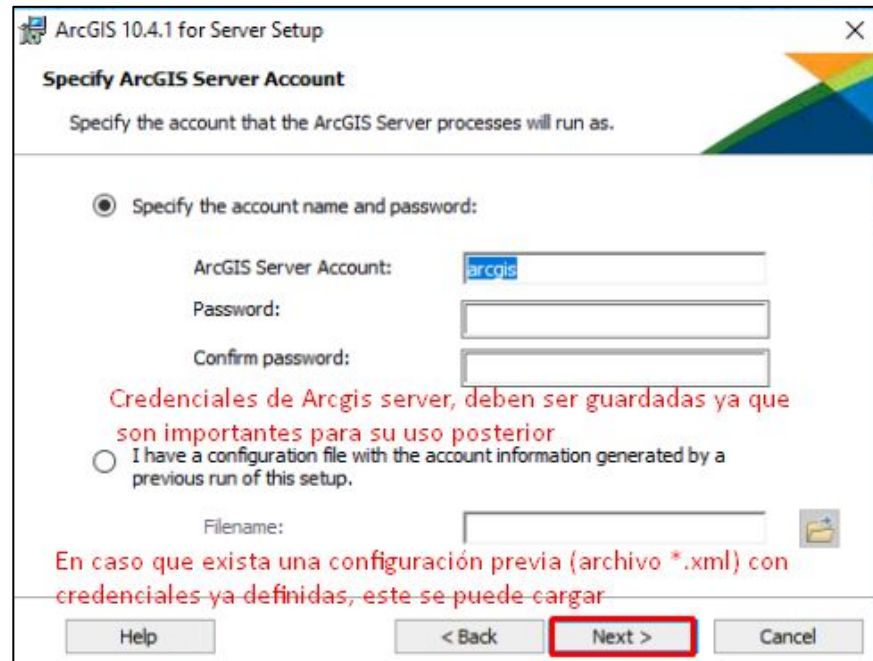
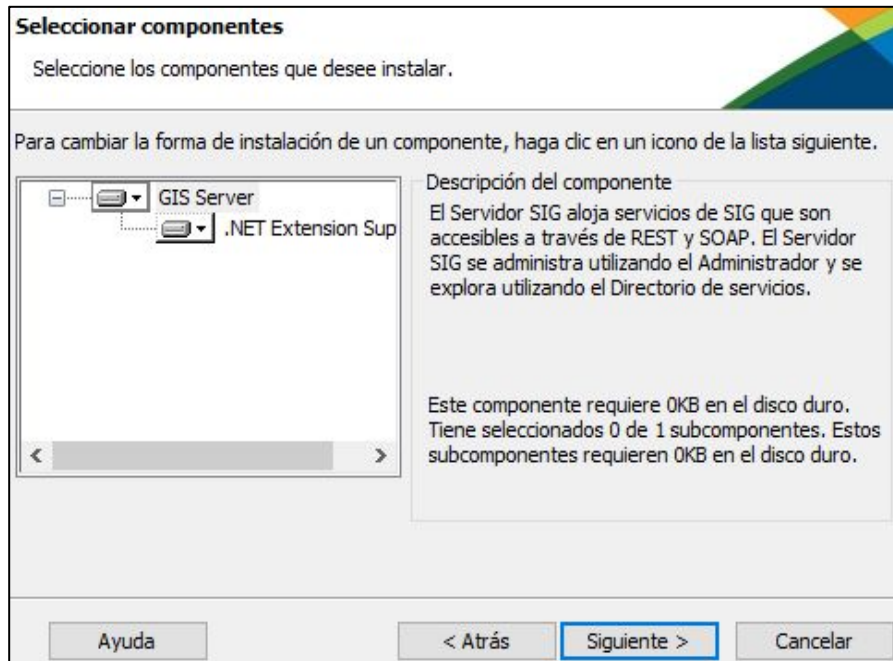


UNIVERSIDAD
DE SANTIAGO
DE CHILE

Creación y gestión de GDBs corporativas



Instalación de Arcgis Server



Nota: Requiere tener previamente instalado Arcgis Desktop

Fuente: Elaboración propia.



Instalación de Arcgis Server

Software Authorization Wizard

Authorization Options
You must authorize the software prior to use. Select from the options below.

Authorization Options

Al igual que Arcgis Desktop, se requiere un archivo de autorización, en este caso, particular para server

☐ I have installed my software and need to authorize it.

☐ I have already authorized the software and need to authorize additional extensions.

☒ I have received an authorization file from Esri and am now ready to finish the authorization process.

C:\Users\Administrador.WIN-1GHMO7QLH1\Desktop\10.4\ Browse...

< Atrás **Siguiente >** Cancelar

Software Authorization Wizard

Authorization Method
Select the method you want to use to authorize the software.

☒ Authorize with Esri now using the Internet.
(This automatic method is the easiest way to authorize. It requires an Internet connection.)

☐ Authorize at Esri's website or by email to receive your authorization file.

< Atrás **Siguiente >** Cancelar

Authorizing Software

✓ Connecting to ESRI...

✓ Sending authorization information...

✓ Receiving authorization file...

✓ Authorizing software...

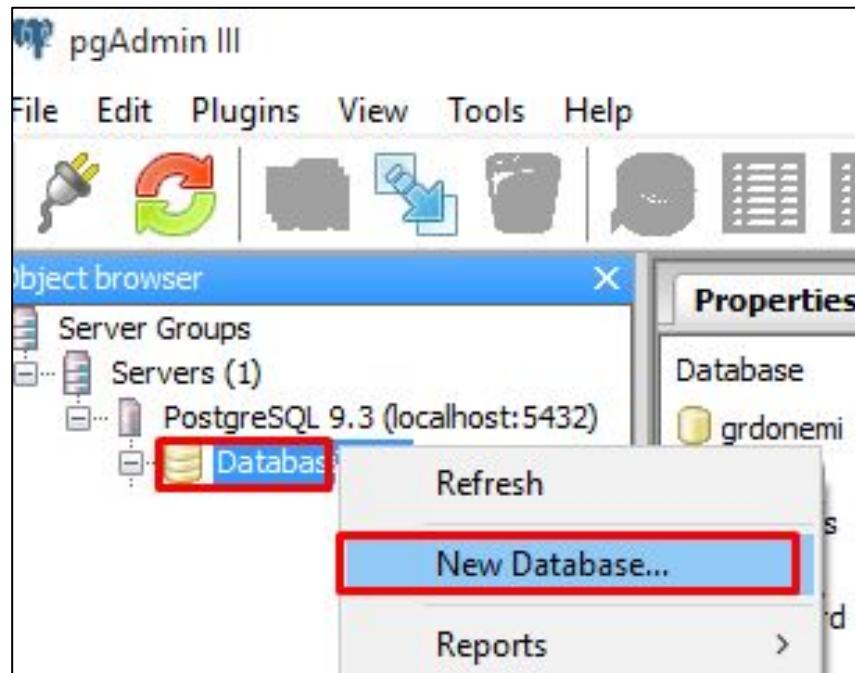
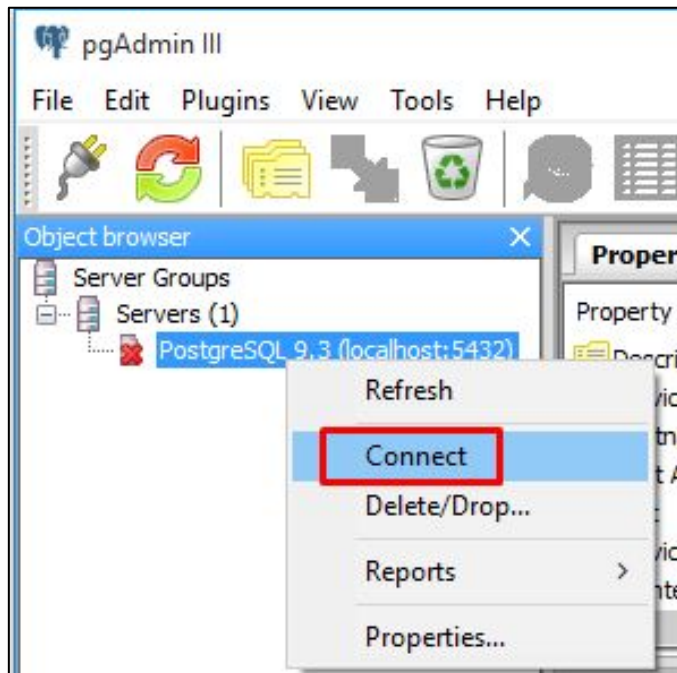
Congratulations, your software has been authorized and is now ready for use.

< Atrás **Finalizar** Cancelar

Nota: Requiere tener previamente instalado Arcgis Desktop

Fuente: Elaboración propia.

Habilitación de Postgis

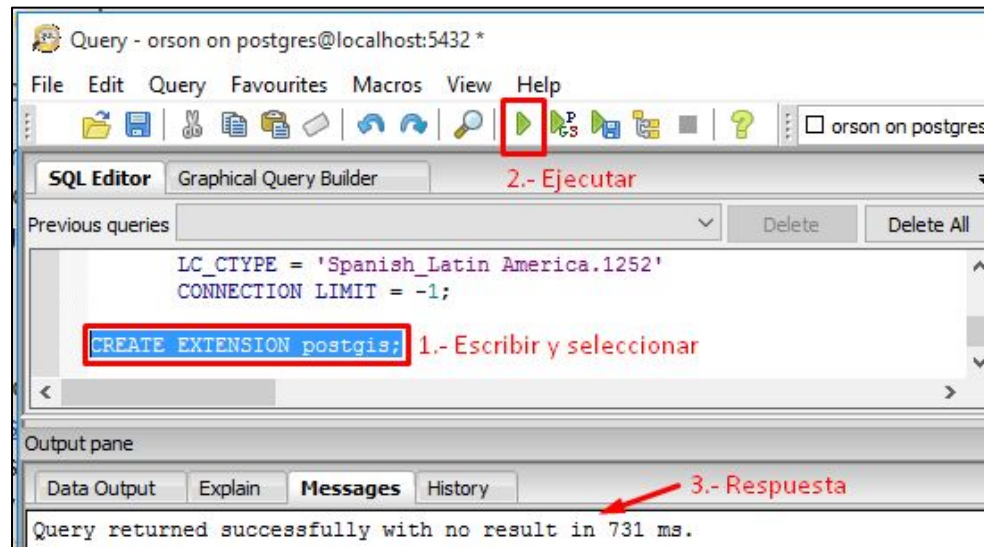
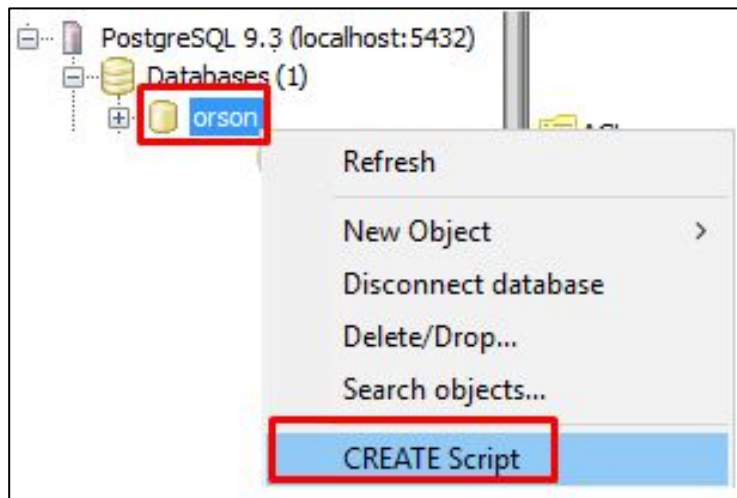


Nota: Requiere tener previamente instalado PostgreSQL

Fuente: Elaboración propia.



Habilitación de Postgis



Nota: Requiere tener previamente instalado PostgreSQL

Fuente: Elaboración propia.



Habilitación de soporte Arcgis/PostgreSQL

De forma de vincular tanto del almacenamiento como procesamiento de la información geométrica entre la base de datos corporativa de Arcgis y PostgreSQL. Arcgis entrega un archivo de soporte (**st_geometry**) de base de datos que debe ser pegado en el directorio de PostgreSQL.

Por defecto se encuentra en el directorio según sea la versión de PostgreSQL, en este caso se ejemplifica con 9.3.

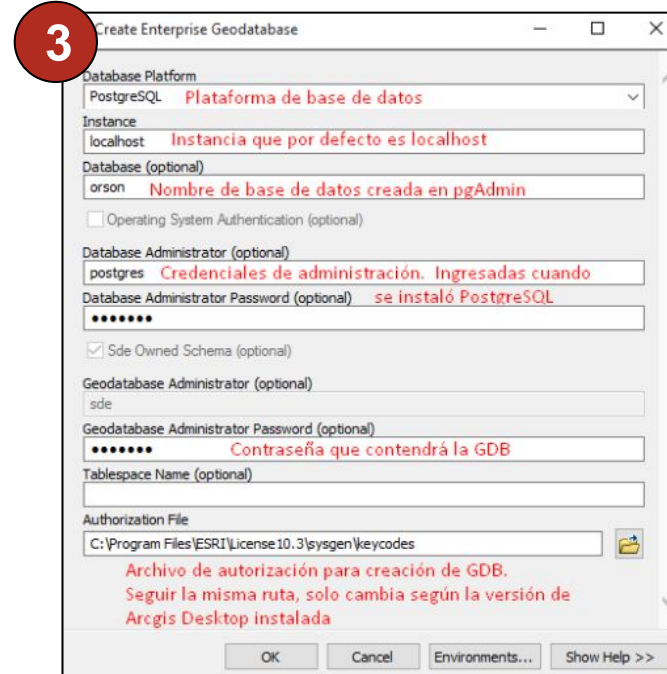
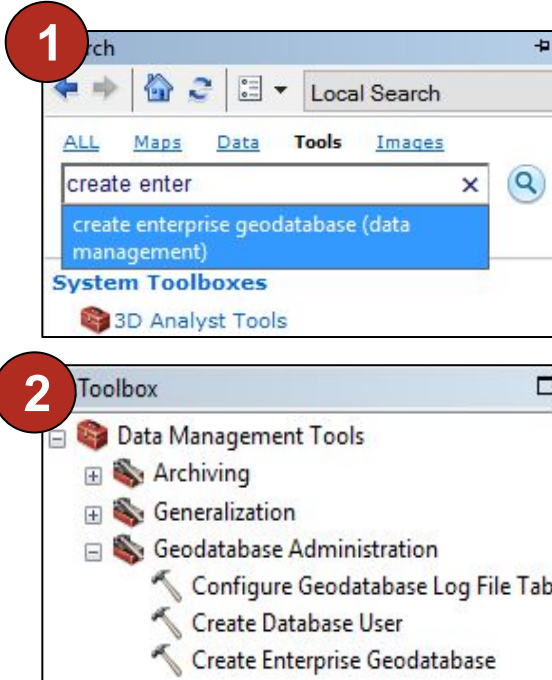
C:\Program Files

(x86)\ArcGIS\Desktop10.3\DatabaseSupport\PostgreSQL\9.3\Windows64

Y deben ser pegados en el directorio

C:\Program Files\PostgreSQL\9.3\lib

Creación de GDB Corporativa



Nota: Requiere tener previamente instalado PostgreSQL

Fuente: Elaboración propia.



Conexión a GDB

Database Connection

Seleccionar plataforma

Database Platform: PostgreSQL

Por defecto la instancia es localhost o 127.0.0.1

Instance: localhost

Las credenciales de autenticación corresponden a las ingresadas cuando fue creada la GDB

Authentication Type: Database authentication

User name: sde

Password:

☒ Save user name and password

Database: orson

Seleccionar base de datos a la cual se establecerá una conexión

[About Database Connections](#) OK Cancel

Toolboxes
Database Servers
Database Connections
Add Database Connection
GIS Servers
Add ArcGIS Server
Add ArcIMS Server
Add WCS Server
Add WMS Server
Add WMTS Server
My Hosted Services
Ready-To-Use Services

Nota: Requiere tener previamente instalado PostgreSQL

Fuente: Elaboración propia.



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Gestión de usuarios



Buenas prácticas en manejo y acceso de datos

Usuarios y Roles

Los privilegios determinan qué es lo que el usuario está autorizado a hacer con los datos y la base de datos.

Estos se Asignan según el tipo de trabajo que realiza la persona dentro de la organización respondiendo a:

- ¿El usuario participa de la administración de la geodatabase?
- ¿El usuario necesita editar o crear datos?
- ¿El usuario sólo debe consultar los datos?

Rol	Descripción	Privilegios sobre Datasets
Administrador	Es el propietario de los datos de una GDB (superusuario). Puede crear usuarios y roles	<ul style="list-style-type: none">• SELECT• INSERT• UPDATE• DELETE
Creador	Usuario encargado de crear esquemas, y dentro de ellos, crear tablas, vistas y clases de entidad. También es el encargado de administrar usuarios y roles en un Dataset.	<ul style="list-style-type: none">• CREATE TABLE• CREATE PROCEDURE• CREATE VIEW
Editor	Usuario encargado de editar tablas de atributos.	<ul style="list-style-type: none">• SELECT• INSERT• UPDATE• DELETE
Visualizador	Usuario con permisos de lectura de tablas de atributos.	<ul style="list-style-type: none">• SELECT



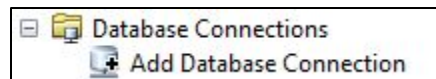
Buenas prácticas en manejo y acceso de datos

Creación de usuarios

Desde el Catálogo de Arcmap dirigirse a conexión de bases de datos y añadir una nueva. Si la GDB corporativa creada es de tipo "sde", en caso que no se haya especificado otro nombre por defecto el superusuario será "sde". La forma de conectarse fue vista en el módulo anterior de Arcpy.

```
import arcpy
arcpy.CreateDatabaseConnection_management("Database Connections",
    "Prueba.sde",
    "POSTGRESQL",
    "5432",
    "DATABASE_AUTH",
    "usuario",
    "contraseña",
    "SAVE_USERNAME",
    "garfield",
    "nombre_base_datos")
```

Fuente: Elaboración propia.



Database Connection

Database Platform: PostgreSQL

Instance: 127.0.0.1

Authentication Type: Database authentication

Por defecto el superusuario se llama "sde"

User name: sde

Password:

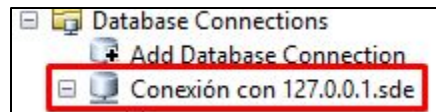
☒ Save user name and password

Database: sde

Por defecto la base de datos es "sde"

[About Database Connections](#)

OK Cancel





Buenas prácticas en manejo y acceso de datos

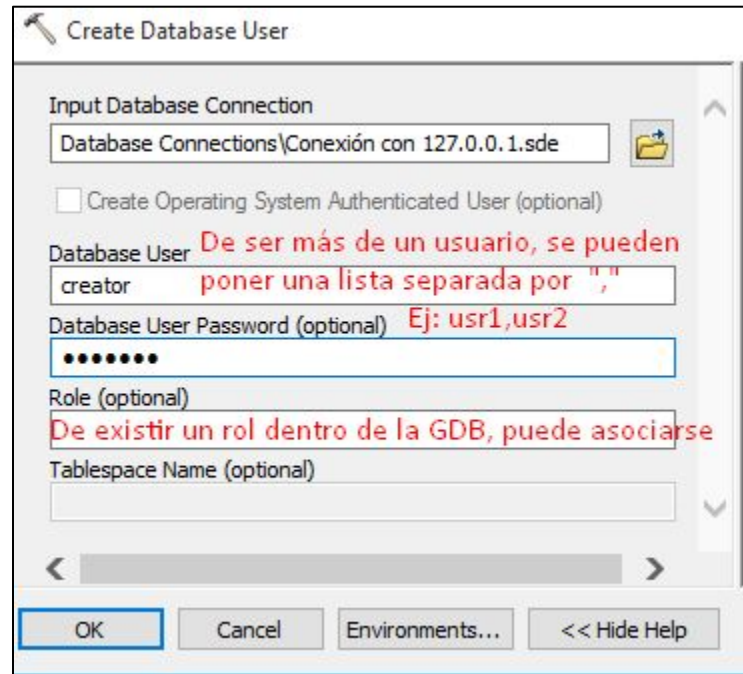
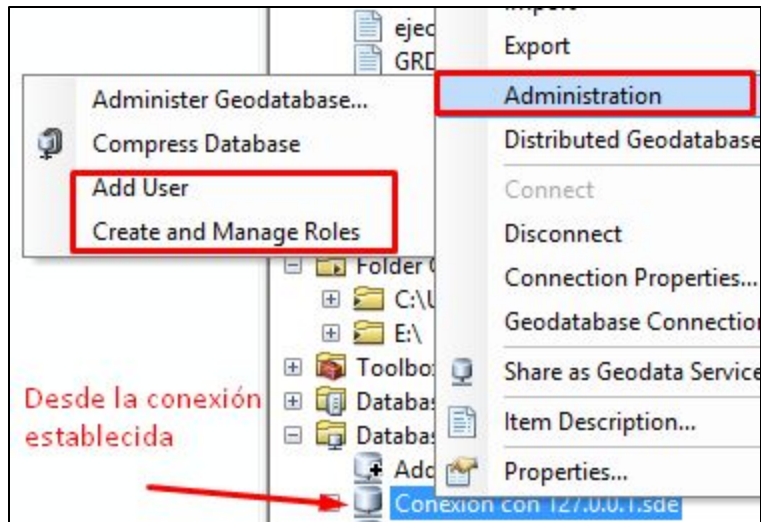
Creación de usuarios y roles

Establecida la conexión a la GDB corporativa se puede acceder al panel de administración para la creación de usuarios y roles. Se debe tener en cuenta que cada usuario tiene asociado un rol, es decir, varios usuarios pueden pertenecer a un rol en particular.

Rol	Usuario
Creador	creator
Editor	editor_1
	editor_2
Visualizador	visual_1
	visual_2
	visual_3

Fuente: Elaboración propia.

Buenas prácticas en manejo y acceso de datos





Buenas prácticas en manejo y acceso de datos

```
CreateDatabaseUser_management(input_database,    {user_authentication_type},    user_name,  
{user_password}, {role}, {tablespace_name})
```

Importar biblioteca

```
import arcpy
```

Conectar a GDB

```
arcpy.CreateDatabaseConnection_management("C:/connections",    "pgconn.sde",    "POSTGRESQL",    myserver,  
mypgdb, "DATABASE_AUTH", "ela", "3L@pwd", "SAVE_USERNAME")
```

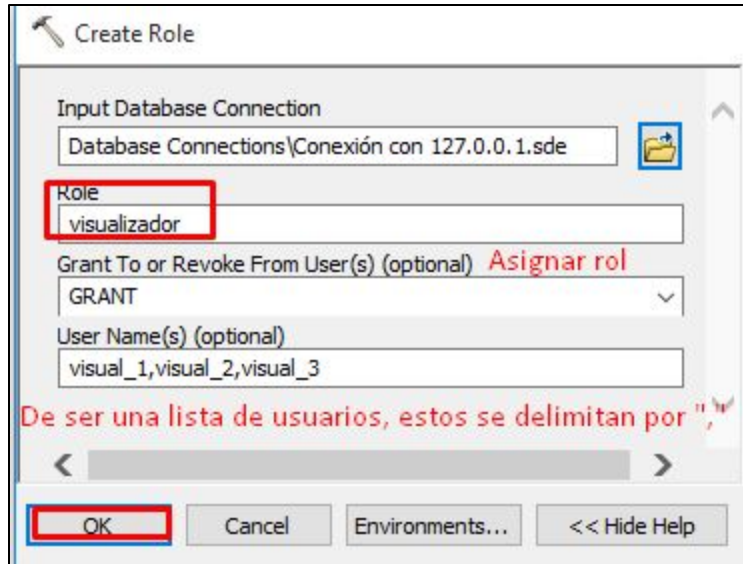
Crear Usuario

```
arcpy.CreateDatabaseUser_management("C:/connections/pgconn.sde", "DATABASE_USER", "dataowner", "N0look")
```

[Revisar Documentación](#)



Buenas prácticas en manejo y acceso de datos



CreateRole_management(input_database, role,
{grant_revoke}, {user_name})

Ejemplo 1: Importar biblioteca
import arcpy

```
arcpy.CreateRole_management("C:\\dbconnections\\db_postg.  
.sde", "drafters", "GRANT", "eng1,eng2")
```

Ejemplo 2: Importar biblioteca
import arcpy

```
arcpy.CreateRole_management("D:\\myconnectionfiles\\mygdb  
.sde", "creators", "REVOKE", "intern1,intern2,intern3")
```

¿En qué se diferencian?

[Revisar Documentación](#)

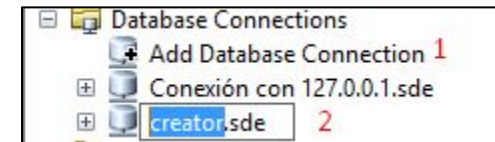


Buenas prácticas en manejo y acceso de datos

Asignación de privilegios

- Se probará estableciendo una nueva conexión con las credenciales correspondientes a un usuario **creator**.
- Con este usuario se realiza la creación de esquemas.
- Para ello se debe tener en cuenta:
 - El usuario **creator** podrá crear, eliminar información y esquemas, además de administrar privilegios de usuarios.
 - El administrador **SDE** podrá crear, eliminar información y esquemas administradas por el **creator**, pero **no** administrar privilegios de otros usuarios.

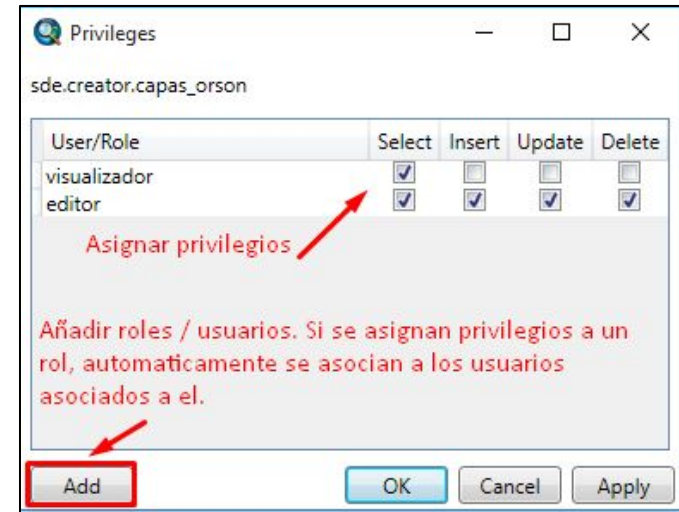
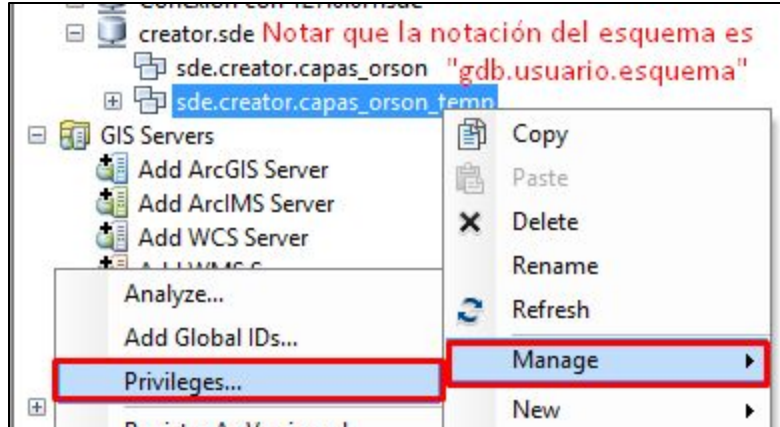
Rol	Usuario
Creador	creator
Editor	editor_1
	editor_2
Visualizador	visual_1
	visual_2
	visual_3





Buenas prácticas en manejo y acceso de datos

Asignación de privilegios





Buenas prácticas en manejo y acceso de datos

Asignación de privilegios

A través de arcpy podemos asignar los diferentes niveles de acceso que tendrán los usuarios a los datasets que se encuentren dentro de nuestras GDB, estos se dividen en:

- **AS_IS:** Mantiene los privilegios existentes.
- **GRANT:** Permite al usuario visualizar el dataset
- **REVOKE:** Quita los privilegios de acceso al dataset

```
import arcpy

datasetName = "C:/Conexiones/gdb@produccion.sde/produccion.GDB.IncendiosFeature"

arcpy.ChangePrivileges_management(datasetName,"Usuario1","GRANT","GRANT")
```



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Trabajo con Sistema de versionamiento en GDB



Transacciones

Corresponde a unidades de trabajo definidas por la aplicación realizada contra una base de datos (Cambios a una base de datos). El proceso de una transacción se define por:

- Inicio de una transacción.
- Realización de modificaciones contra la base de datos.
- Confirmación o no de la transacción.
- De ser confirmada, los cambios se hacen visibles para otros usuarios y aplicaciones.



Transacciones

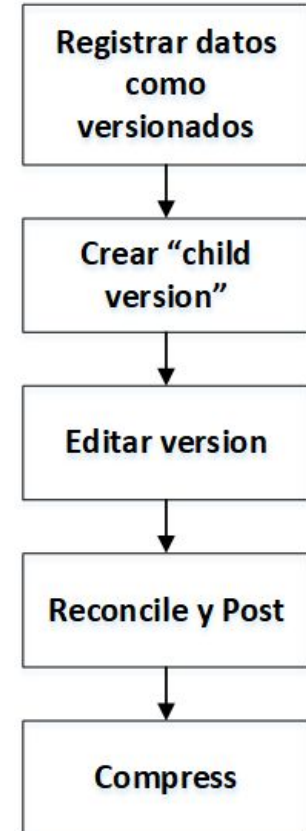
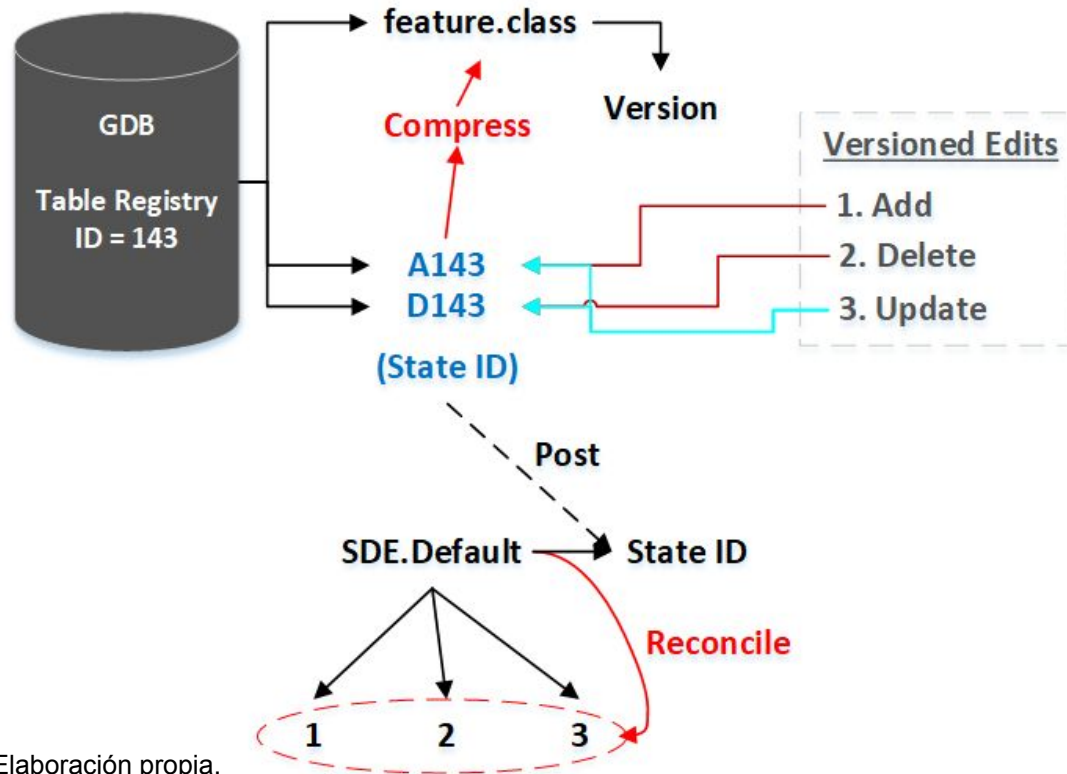
Propiedades de las transacciones:

- **Atómica:** una transacción exhibe un comportamiento todo o nada. Si se confirma, todos los cambios se aplican a la base de datos. Si se deshace, ninguno de sus cambios se aplica.
- **Coherente:** una transacción deja la base de datos en un estado coherente.
- **Aislada:** una transacción puede aislar sus cambios de otras transacciones hasta que los confirme. Otros usuarios no ven el trabajo interno de la transacción mientras está en curso.
- **Duradera:** una vez que una transacción se confirma, sus resultados son persistentes.

Para lograr estas propiedades, los sistemas de administración de bases de datos utilizan una variedad de mecanismos de bloqueo para garantizar que varias transacciones simultáneas se blinden o se aíslen entre sí.



Vista General de Versionamiento





Ejemplo

El siguiente ejemplo muestra el uso de versionamiento de GDB mediante la edición de feature class desde un usuario de jerarquía menor al administrador.

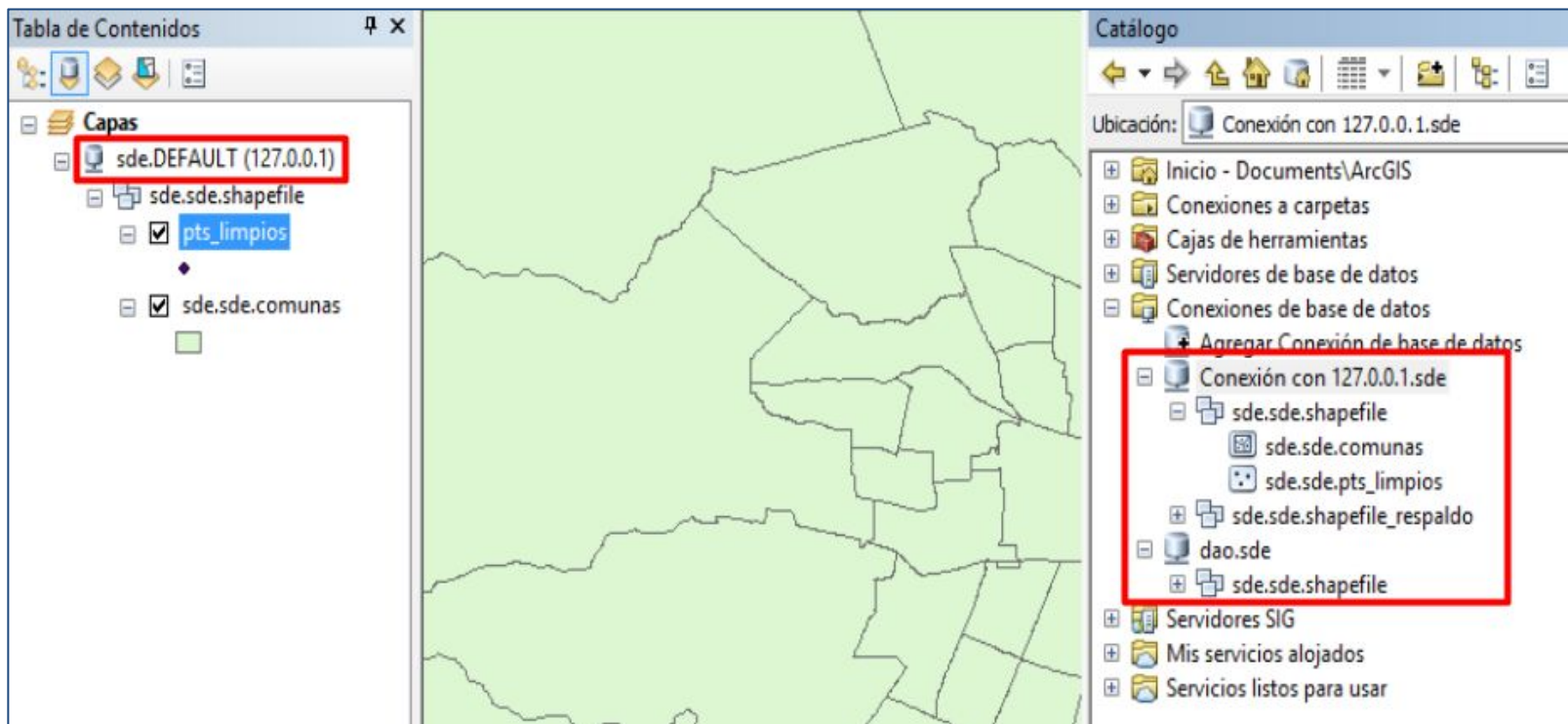
Inicialmente se disponen de 2 conexiones a la misma GDB:

- 1) Desde SDE: Administrador y dueño de GDB.
- 2) Desde usuario_dao: Usuario con permisos de edición sobre un esquema en particular.

Viendo ambas conexiones. ¿Qué observa?



Ejemplo

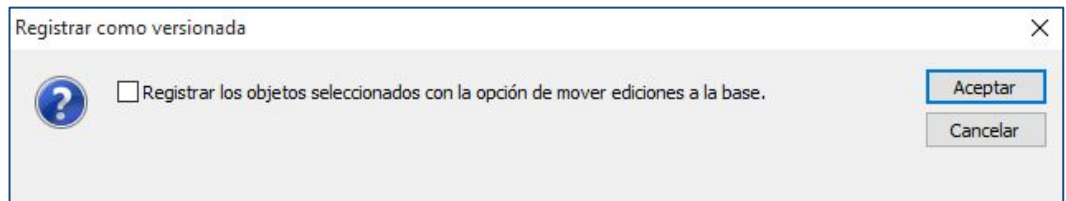
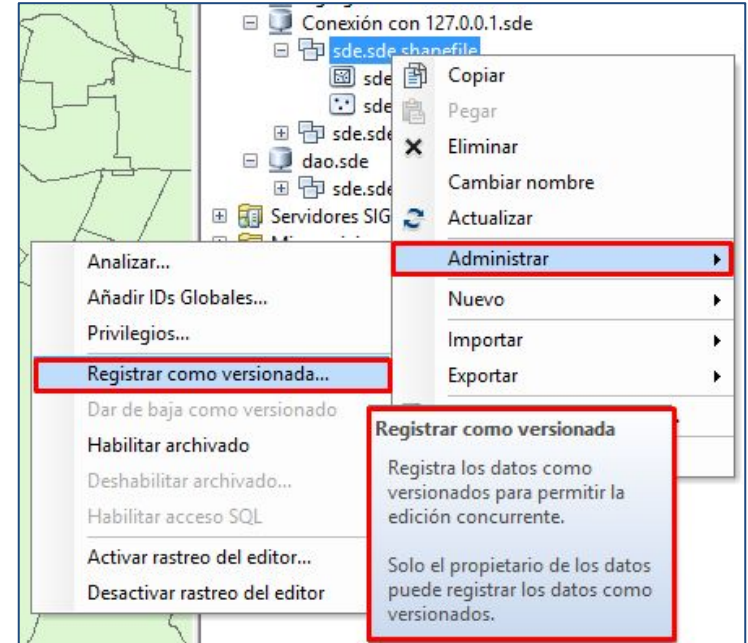




Ejemplo

Vista desde SDE

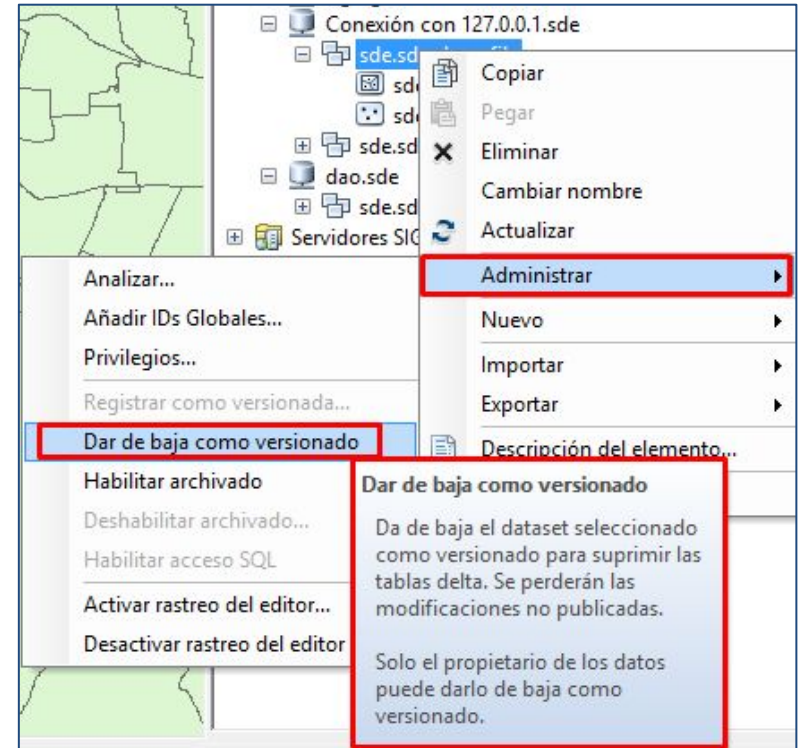
- Para el dueño o administrador de esquema, se encuentran habilitadas las opciones de “Administrar”.
- Para habilitar edición concurrente es necesario registrar un esquema como versionado.



Ejemplo

Vista desde SDE

- De la misma forma, se puede desactivar la edición, dando de baja el versionamiento del esquema.
- Esto implica que se suprimen tablas Deltas, perdiendo modificaciones que no han sido publicadas por otras versiones de la raíz.





Ejemplo

Vista desde otro usuario

- Visto en la imagen de la lámina siguiente:
 - Al conectarse como otro usuario, en este caso como usuario de otra dirección “usuario_dao”; las opciones de Administración se encuentran deshabilitados. ¿Por qué?
 - La tabla de contenidos contiene cargado esquemas desde el usuario SDE y usuario_dao. ¿Qué tienen en común? ¿En qué se diferencian?



Ejemplo

Tabla de Contenidos

Capas

- sde.DEFAULT (127.0.0.1)
 - sde.sde.shapefile
 - pts_limpios (DAO)
 - sde.sde.comunas
- sde.DEFAULT (127.0.0.1)
 - sde.sde.shapefile
 - pts_limpios (SDE)
 - sde.sde.comunas

Catálogo

Ubicación: sde.sde.shapefile

- Inicio - Documents\ArcGIS
- Conexiones a carpetas
- Cajas de herramientas
- Servidores de base de datos
- Conexiones de base de datos
 - Agregar Conexión de base de datos
 - Conexión con 127.0.0.1.sde
- dao.sde
 - sde.sde.shapefile
 - sde.sde
 - sde.sde
- Servidores SIG
- Mis servicios alojados
- Servicios listos para usar

Analizar...
Añadir IDs Globales...
Privilegios...
Registrar como versionada...
Dar de baja como versionado
Habilitar archivado

Copiar
Pegar
Eliminar
Cambiar nombre
Actualizar
Administrar
Nuevo
Importar
Exportar
Descripción del elemento
Propiedades...

Ejemplo

Herramienta de versionado: Esta se habilita desde el menú de herramientas



Administrador de versiones

Abre la ventana Administrador de versiones para crear nuevas versiones, cambiar el nombre de las versiones existentes, eliminar versiones y alterar propiedades de versión.

Requiere una licencia estándar o avanzada y es solo para lectura con una licencia básica.



Crear nueva versión

Crea una nueva versión secundaria de la base de datos donde la versión principal es la versión actual. En el momento en que se crea la nueva versión, ésta es idéntica a la versión de la que deriva.

Requiere una licencia estándar o avanzada y está deshabilitado con una licencia básica.



Actualizar

Actualiza la conexión de base de datos para todas las versiones y vuelve a dibujar el mapa de modo que pueda ver todo cambio que pueda haberse conciliado y publicado recientemente.



Ejemplo

Herramienta de versionado: Esta se habilita desde el menú de herramientas



Cambiar Versión

Cambia a una versión de geodatabase diferente.



Conciliar

Extrae cambios de la versión principal a la versión secundaria al fusionar todos los datasets, clases de entidad y tablas modificados entre la versión actual que se está editando y la versión de destino.

Requiere una licencia estándar o avanzada y está deshabilitado con una licencia básica.



Publicar

Inserta cambios de una versión secundaria en su versión principal al aplicar la sesión de edición actual a la versión de destino conciliada. Es necesario utilizar Conciliar versión para elegir una versión de destino antes de publicar.

Requiere una licencia estándar o avanzada y está deshabilitado con una licencia básica.

Ejemplo

Herramienta de versionado: Esta se habilita desde el menú de herramientas



Conflictos

Identifica y resuelve conflictos entre la versión de edición actual y la versión de destino contra la que se concilió. Si una entidad se editó en ambas versiones, la entidad se encuentra en conflicto.

Requiere una licencia estándar o avanzada y está deshabilitado con una licencia básica.



Cambios de versión

Permite ver de qué manera cambió una versión con respecto a otra versión antecesora en la geodatabase.

Requiere una licencia estándar o avanzada y está deshabilitado con una licencia básica.

Comprendiendo la herramienta de versionado; ¿Cómo se relaciona con el esquema general de versionamiento?

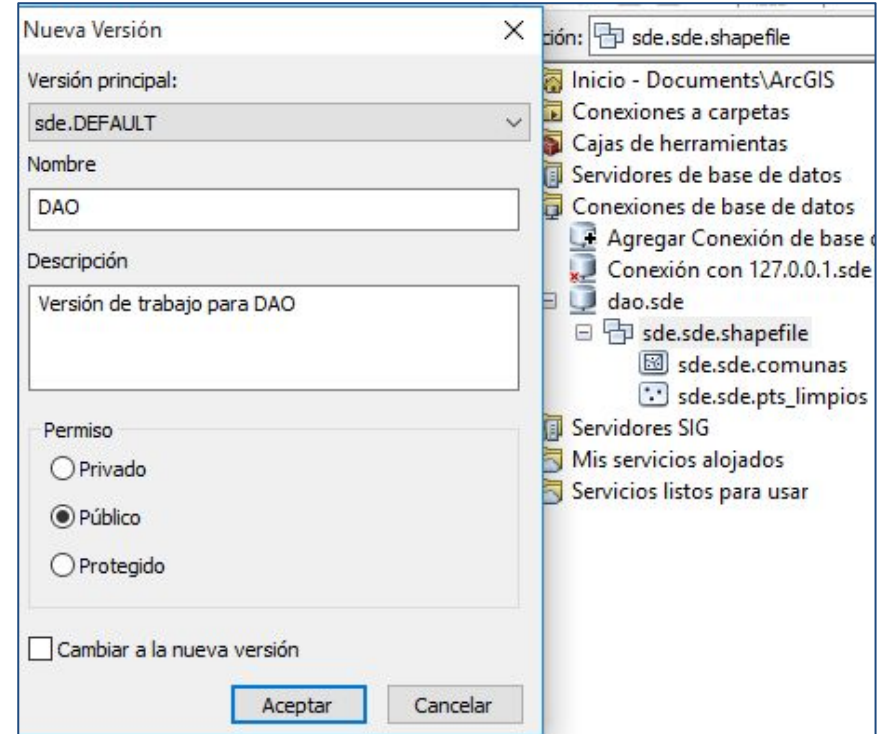


Ejemplo

Vista desde otro usuario

- Desde la herramienta, el primer paso es crear una nueva versión.
- Para ello es necesario seleccionar desde qué versión se realizará esta nueva.
- Seguido de nombre, descripción y permiso.

¿Qué significa cada permiso?





Ejemplo

Vista desde otro usuario

- Desde la administración de versiones, podemos identificar:
 - Descripción y permisos ingresados al momento de crear la versión.
 - Propietarios de cada versión.
 - Última fecha y hora en que se trabajó en cada versión.

Fuente: Elaboración propia.

Administración de la geodatabase (usuario_dao@DESKTOP-LF067QO/sde)

Versiones Bloqueos

Filtrado

Nombre: Propietario:

Nombre	Propietario	Modificado
DEFAULT	sde	19/9/2020 18:29:29
DAO	sde	19/9/2020 18:29:29
DAO_INTERNO	usuario_dao	19/9/2020 18:29:29

Actualizar 3 de 3 Versiones en 20/9/2020 11:44:08

Transaccional Vista del árbol Histórico

Propiedades

Nombre: DEFAULT

Propietario: sde

Padre:

Descripción: Versión predeterminada de la instancia.

Acceso: Público

Creado: 19/11/2018 08:56:04

Modificado: 19/9/2020 18:29:29

Está bloqueando: False

Es réplica: False

Está bloqueado: False

Administración de la geodatabase (usuario_dao@DESKTOP-LF067QO/sde)

Versiones Bloqueos

Filtrado

Nombre: Propietario:

Nombre	Propietario	Modificado
DEFAULT	sde	19/9/2020 18:29:29
DAO	sde	19/9/2020 18:29:29
DAO_INTERNO	usuario_dao	19/9/2020 18:29:29

Actualizar 3 de 3 Versiones en 20/9/2020 11:44:08

Transaccional Vista del árbol Histórico

Propiedades

Nombre: DAO_INTERNO

Propietario: usuario_dao

Padre: sde.DAO

Descripción: Modificaciones Internas de DAO

Acceso: Privado

Creado: 20/9/2020 11:43:18

Modificado: 19/9/2020 18:29:29

Está bloqueando: False

Es réplica: False

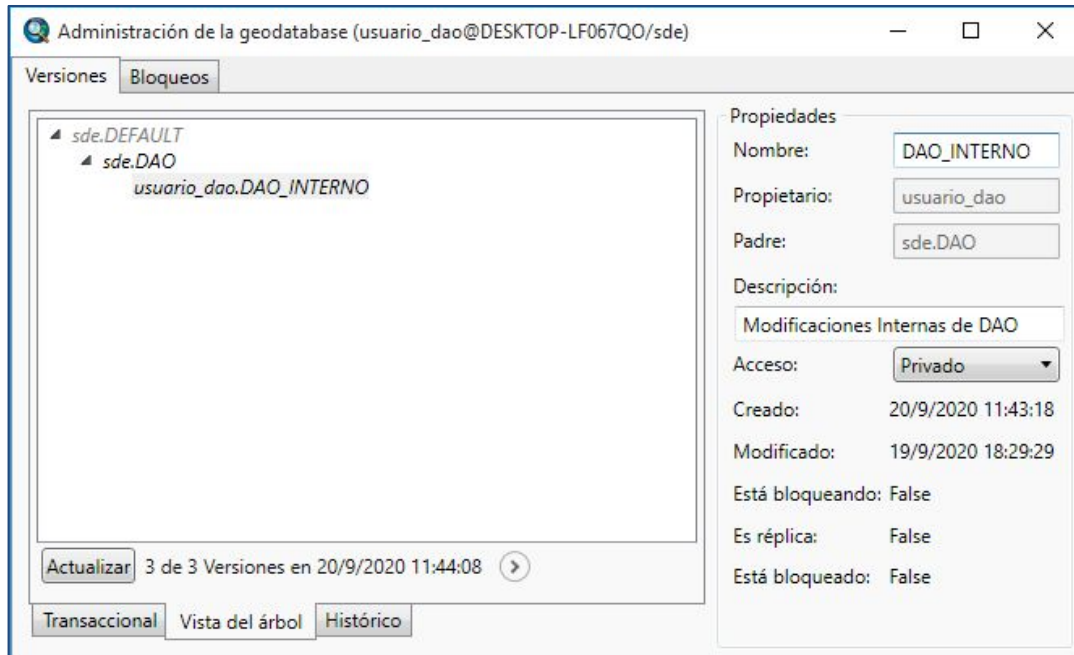
Está bloqueado: False



Ejemplo

Vista desde otro usuario

- Desde la Vista de árbol, es posible entender la ramificación que tiene cada versión, y por consecuencia en qué orden se deberían ir validando y publicando las ediciones.



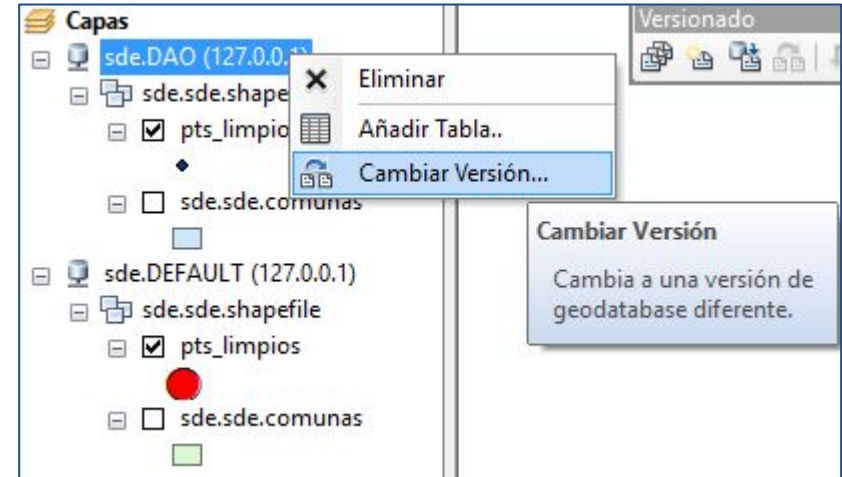


Ejemplo

Vista desde otro usuario

- Desde la tabla de contenidos y estando conectados sólo como usuario_dao. Cambiamos la versión a la creada para la unidad.

Recordar que la primera instancia de carga en la tabla corresponde a la sesión del usuario.



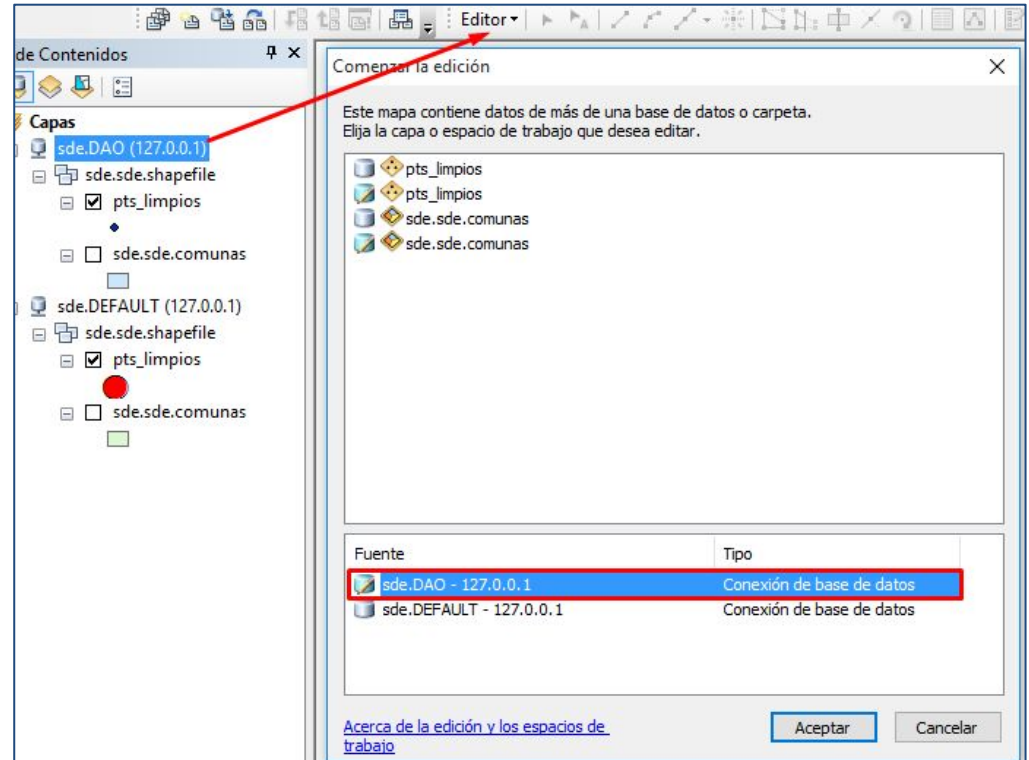
Ejemplo

Vista desde otro usuario

- Para comenzar a editar el contenido de la capa de pts_limpios, es necesario habilitar el modo editor.

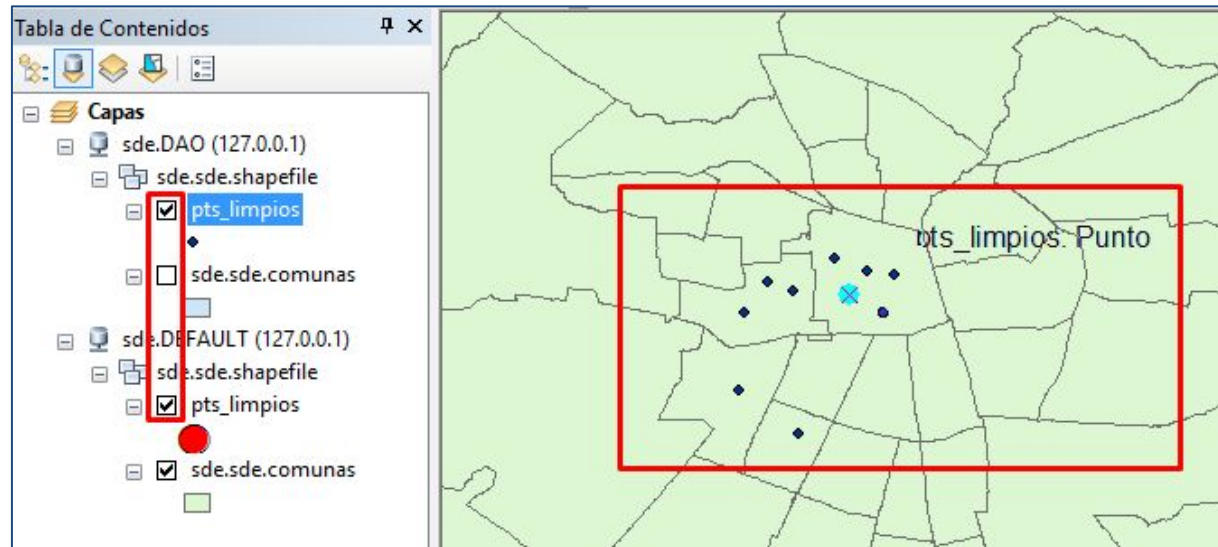
¿Qué tiene de relevante el modo editor?

¿Qué se observa en la imagen?



Ejemplo

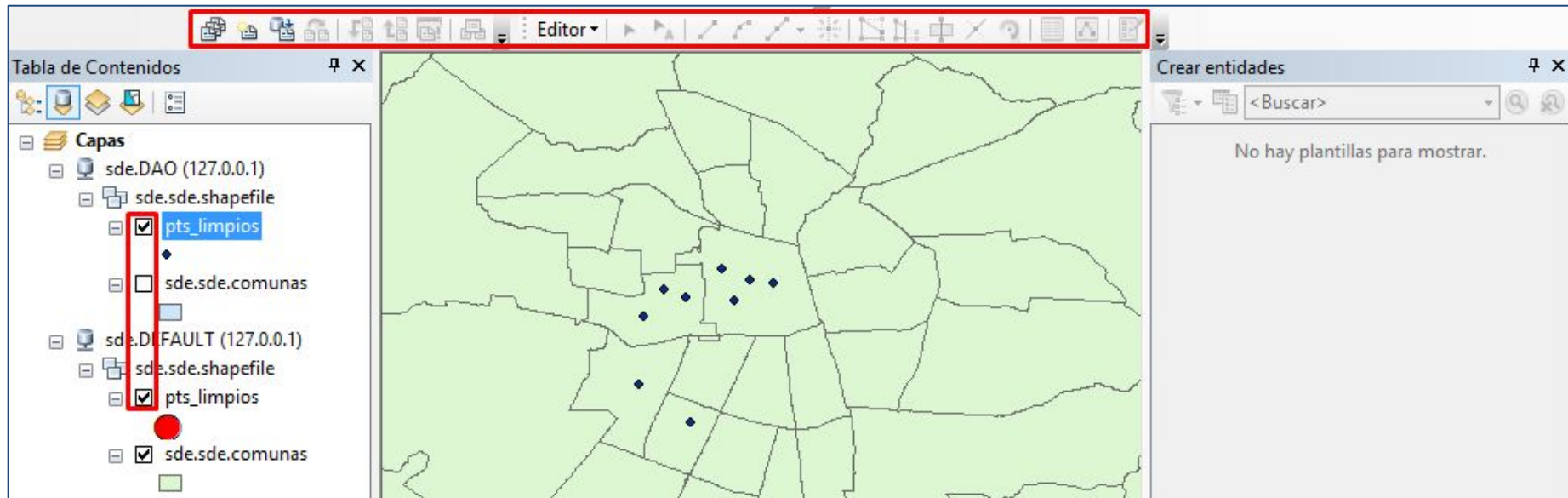
Desde el usuario_dao se insertan entidades de puntos, sin embargo. ¿Qué se observa desde la tabla de contenido y despliegue de capas?



Fuente: Elaboración propia.

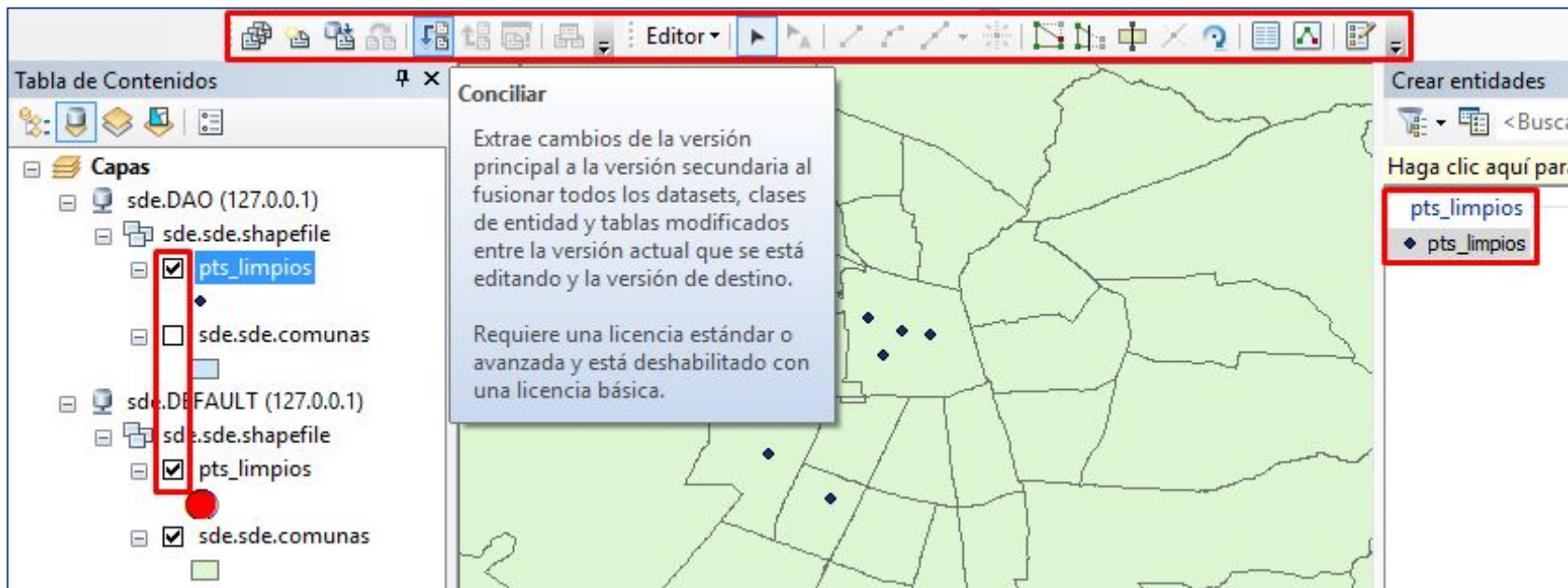
Ejemplo

Se deben conciliar cambios para ser publicados, sin embargo, esto no es posible. ¿Por qué?



Ejemplo

Recordar que para conciliar y publicar, es necesario tener habilitado el modo editor. Así como se editan entidades y atributos; también se editan versiones.





Ejemplo

Vista desde otro usuario

- Desde la opción de cambios de versión es posible analizar los cambios.

The screenshot shows a GIS application window titled 'Cambios de versión'. The interface includes a tree view on the left, a table of properties in the center, and two map views at the bottom.

Tree View:

- Cambios (10)
 - pts_limpios(10)
 - Inserta(10)
 - 5673
 - 5674
 - 5675
 - 5676
 - 5677
 - 5678
 - 5679
 - 5680
 - 5681
 - 5682

Table of Properties:

Propiedad	sde.DAO	sde.DEFAULT	Antepasado común
OBJECTID	5678		
lat	0		
lng	0		
status			
type			
address_ty			
address_na			
address_nu			
region_id	0		
commune_id			
owner			
manager			

Map Views:

- Left Map View: Labeled 'sde.DAO' in the bottom left corner. It shows a map of a city area with several red circular markers.
- Right Map View: Labeled 'sde.DEFAULT' in the bottom right corner. It shows the same map area but without the red markers.

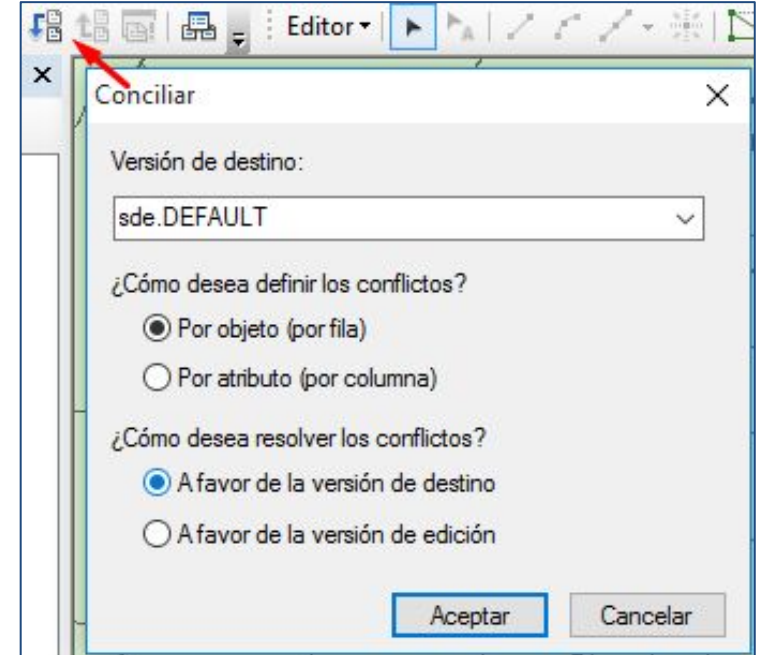
Fuente: Elaboración propia.

Ejemplo

Vista desde otro usuario

- Desde la opción de Conciliar, se validan los cambios a una raíz o rama de destino y se define como se definen y resuelven conflictos.

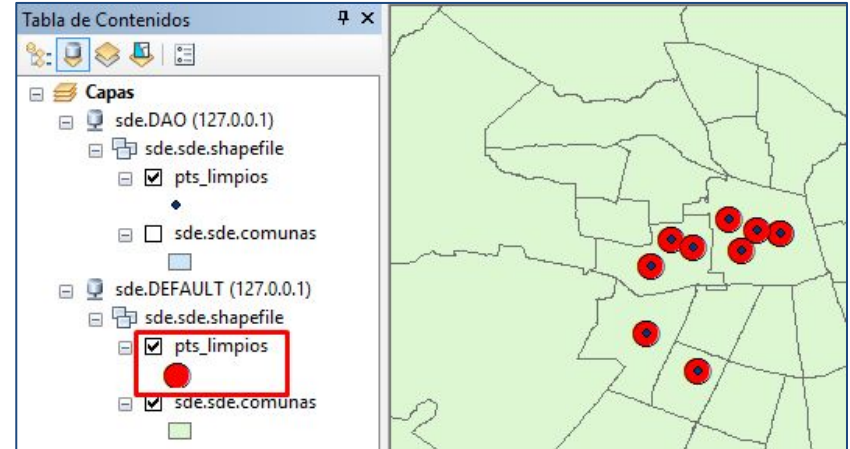
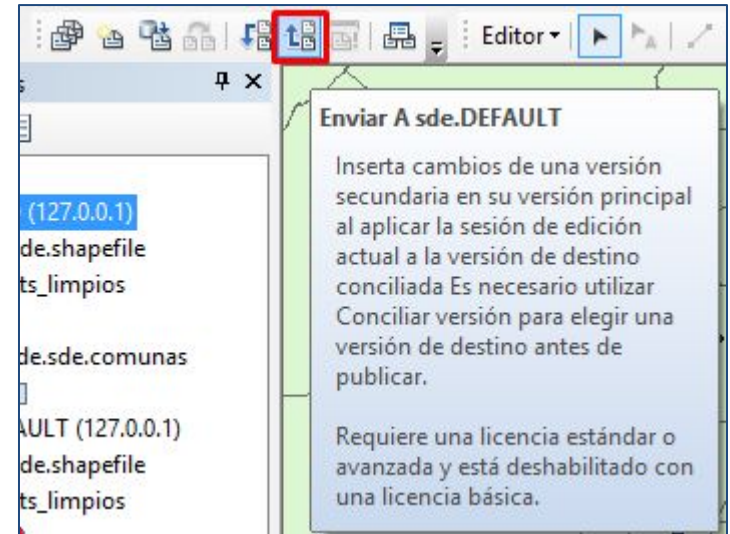
¿Cuándo podría existir un conflicto?





Ejemplo

- Conciliados los cambios, estos pueden enviarse y aplicarse a la versión de jerarquía mayor.
- Los cambios serán publicados y visibles en la versión primaria.

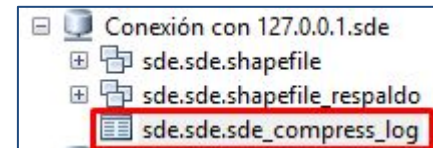
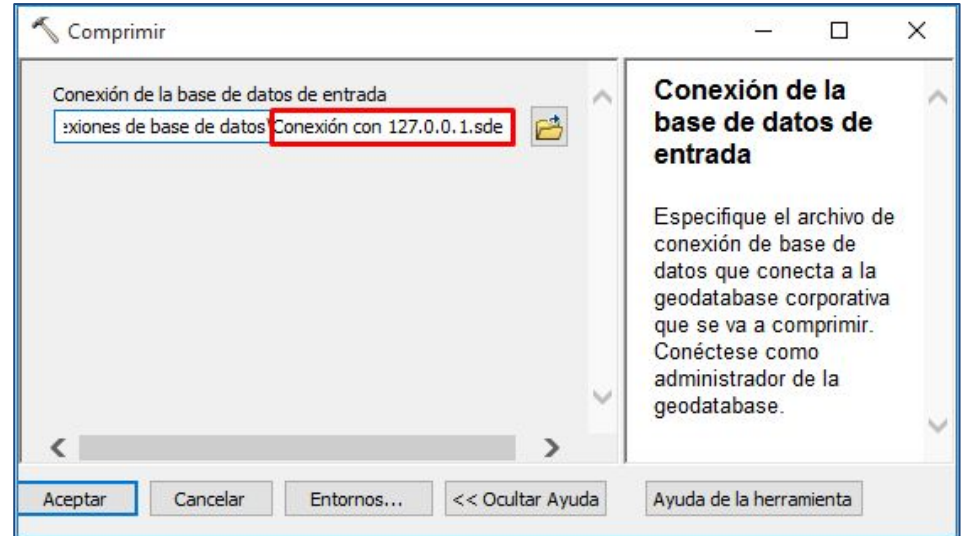




Ejemplo

- Finalmente se comprimen las ediciones.
- Esto implica que se analizan las tablas deltas y se elimina toda redundancia.

¿Qué tipo de redundancia?





UNIVERSIDAD
DE SANTIAGO
DE CHILE

Servicios de Mapas



Objetivo

El presente contenido tiene por finalidad introducir al estudiante sobre los tipos de servicios de mapas, sus capacidades y configuraciones de uso, y publicación.

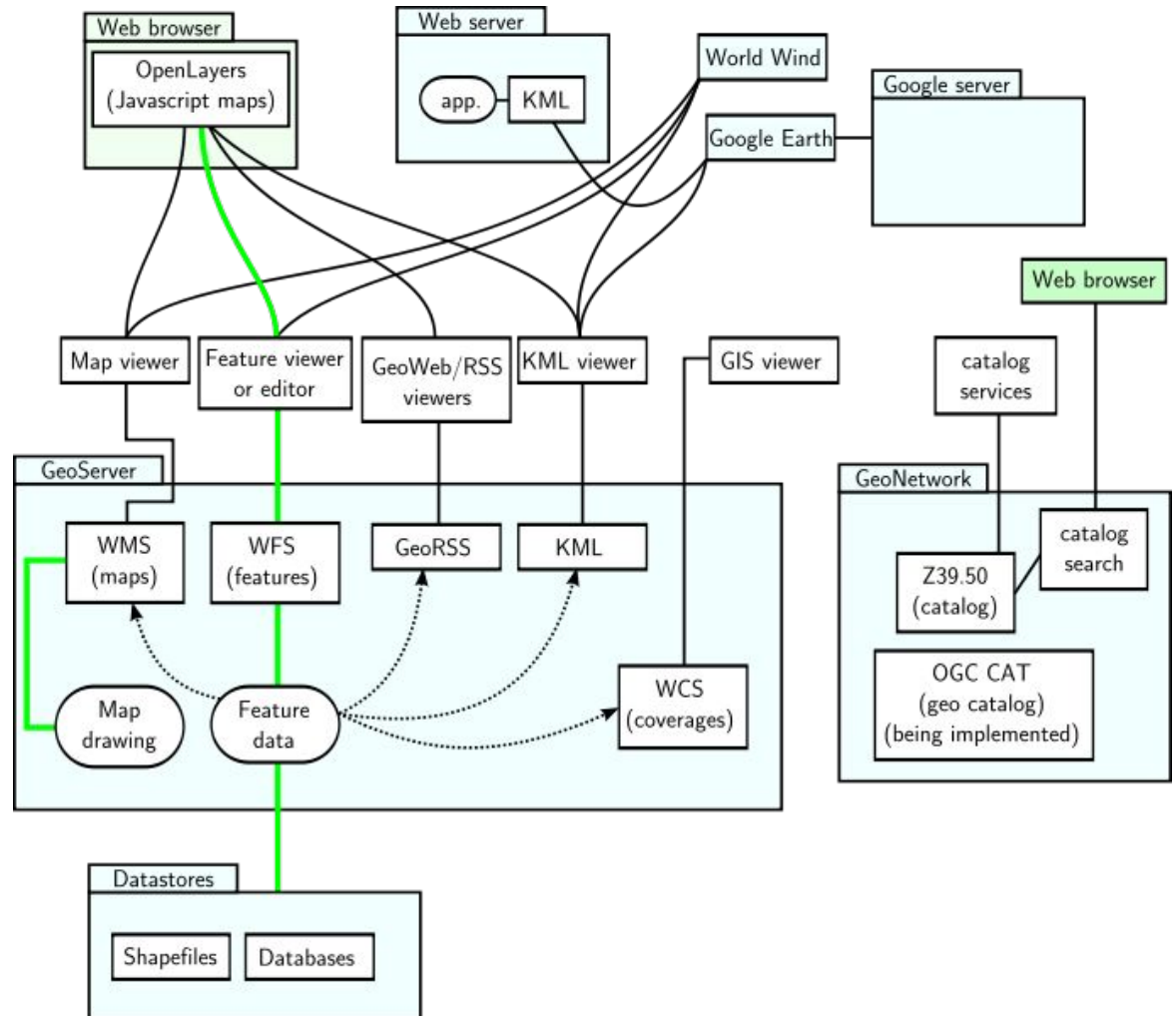
Contenido:

- 1: Introducción a servicios de mapas, tipos y estándares (ej: WMS, WFC, WTMS, CSW, WPS, ECQL / MapService, FeatureService, Geoprocessing Service)
- 2: Ejemplos y casos de usos de tipos de servicios
- 3: Publicación y consumo de servicios (protocolos de generación y solicitud)



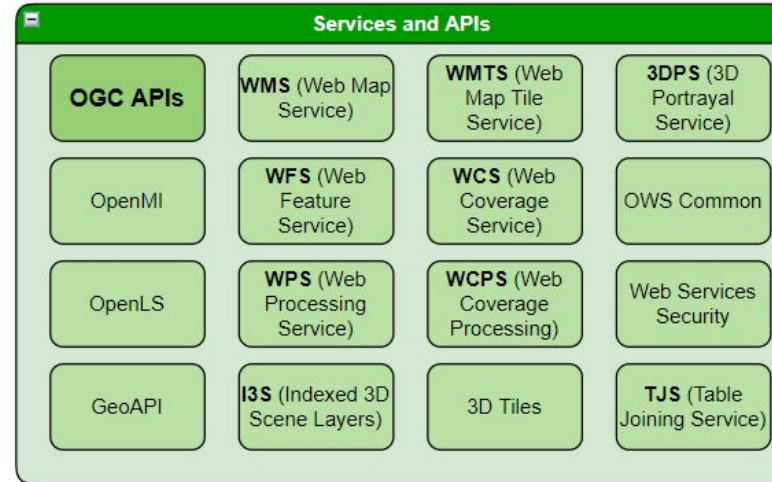
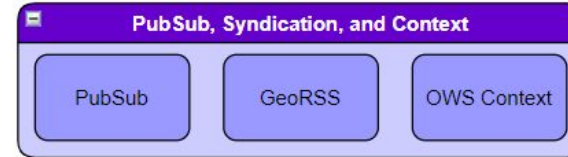
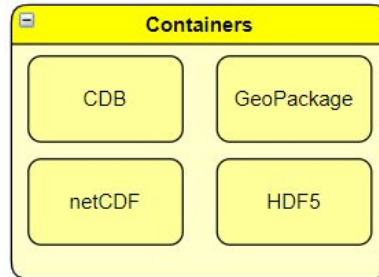
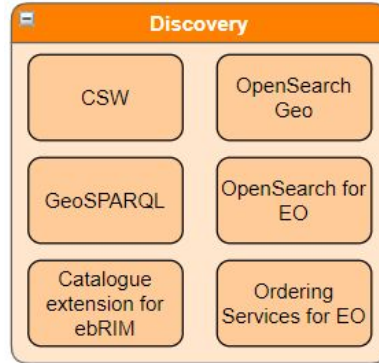
Diagrama Conceptual

[OGC Standards](#) | [OGC](#)





Agrupación de Estándares OGC

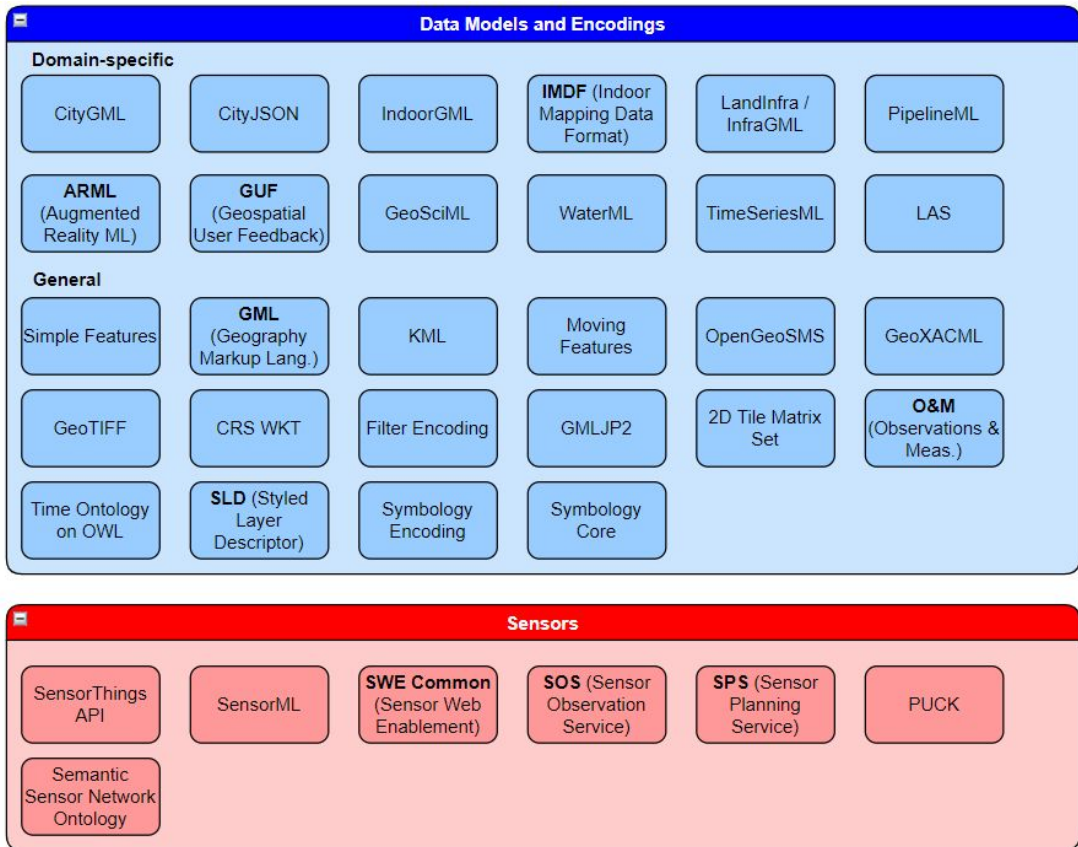


[OGC Standards | OGC](#)

Fuente: OGC



Agrupación de Estándares OGC



[OGC Standards](#) | [OGC](#)

Fuente: OGC



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Aplicativos SIG



Objetivo

El presente contenido tiene por finalidad introducir al estudiante la creación de aplicaciones en entornos de trabajo de Escritorio, Web y móviles, con la finalidad de generar una plataforma geo-informática que resuelva las necesidades de los usuarios en cuanto a herramientas y funcionalidades requeridas.

Contenido:

1. Aplicativos Escritorio: lenguaje Arcpy, automatización de geoprocesos
2. Aplicativos Web: ArcGIS Online (Web AppBuilder, Dashboard, Storymaps)
3. Aplicativos Móviles: Collector, Survey123, Explorer, FieldMaps
4. Caso de aplicación: Explicación de caso de uso en contexto GRD



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Ver Presentación: Introducción a Python y Arcpy