

# **MANUAL DE USUARIO DE VCSGIS PARA GVSIG DESKTOP**



*Asociación gvSIG  
[www.gvsig.com](http://www.gvsig.com)*

 <b>www.gvsig.com</b>	Calle Ángel Guimerá, 61 – Puerta 3, 46008 Valencia, España Información general: <a href="mailto:info@gvsig.com">info@gvsig.com</a> Teléfono: (+34) 961919938
<p>Listas de distribución:</p> <p>Existen diversas listas de distribución con el objeto de facilitar la comunicación entre todos los interesados en el proyecto gvSIG, tanto usuarios como desarrolladores.</p> <p><a href="http://www.gvsig.com/es/comunidad/listas-de-correo">http://www.gvsig.com/es/comunidad/listas-de-correo</a></p>	
<p><i>Todos los nombres propios de programas, sistemas operativos, equipo hardware etc., que aparecen en este curso son marcas registradas de sus respectivas compañías u organizaciones.</i></p>	
<p><i>Este documento se distribuye con la licencia Creative Commons Reconocimiento-CompartirIgual 3.0 Unported</i> <i>(<a href="http://creativecommons.org/licenses/by-sa/3.0/deed.es">http://creativecommons.org/licenses/by-sa/3.0/deed.es</a>)</i></p>	

# Índice

<b>1 INTRODUCCIÓN.....</b>	<b>5</b>
1.1. Conceptos básicos del control de versiones.....	5
Repositorio.....	6
Modelos de versionado.....	7
Copias de trabajo.....	10
Flujo de trabajo (Modelo Copiar-Modificar-Fusionar).....	10
Revisiones.....	11
Estados de tablas y registros.....	12
Conexión con el repositorio.....	13
<b>2 UTILIZACIÓN BÁSICA.....</b>	<b>14</b>
Introducción.....	14
Creación de un repositorio.....	14
Iniciar servidor VCSGIS.....	15
Creación de una copia de trabajo.....	16
Añadir una capa al repositorio.....	17
Añadir una capa del repositorio.....	21
Ciclo de trabajo básico.....	22
Gestión de las Revisiones.....	27
Exportar datos desde el repositorio.....	29
<b>3 AÑADIR CAPA VCSGIS USANDO EL DIÁLOGO “AÑADIR CAPA” DE GVSIG DESKTOP.....</b>	<b>31</b>
<b>4 IMPORTACIÓN O CARGA DE HISTORIAL DE DATOS.....</b>	<b>33</b>
<b>5 TOPOLOGÍA EN VCSGIS.....</b>	<b>37</b>
<b>6 ASIGNACIÓN DE RECURSOS A UNA ENTIDAD.....</b>	<b>42</b>
Introducción.....	42
Tabla de recursos.....	42
Ejemplo asignación recursos.....	42
<b>7 MODELOS DE DATOS.....</b>	<b>57</b>
Introducción.....	57
Configuración del modelo de datos en VCSGIS.....	58
Conectarnos a un modelo de datos.....	63
Consideraciones especiales sobre los modelos de datos.....	64

**8 HERRAMIENTAS.....65**

    Herramienta mostrar cambios.....65

# 1 INTRODUCCIÓN

Este documento presenta una introducción resumida al control de versiones **VCSGIS** o **Version Control System Gis**. En él se tratan los conceptos generales de un sistema de control de versiones genérico y estos mismos en el caso concreto de *VCSGIS*. Además el documento pretende que tras su lectura el usuario no solo conozca las partes o elementos que componen este tipo de sistemas sino que pueda utilizarlo o trabajar con *VCSGIS*. Para llevar a cabo la segunda idea esta documentación presenta una guía de uso más ejemplos básicos.

Antes de empezar a describir las partes que componen un sistema de control de versiones hay que definir claramente que son estos y por tanto que es *VCSGIS*. Un sistema de control de versiones es un programa o software basado en la centralización de información para compartir entre usuarios que a diferencia de un servidor normal, recuerda los cambios que hayan sido realizados en sus datos. Cabe destacar de nuevo que no es un servidor al uso, ya que no solo almacena información, sino que almacena la información así como las modificaciones que los usuarios realizan sobre ella, siendo igual de importante la gestión de esos cambios como la propia información.

También hay que destacar que los sistemas de control de versiones no se centran en ningún tipo de dato en concreto, pudiendo existir sistemas que controlan archivos fuente, imágenes... dependiendo de la necesidad de los usuarios. En el caso de *VCSGIS* al estar orientado a un perfil técnico especializado en información geográfica. Este gestiona información sobre tablas y capas (tablas con información geométrica).

Sin más dilación comenzamos a detallar las diferentes partes que componen un sistema de control de versiones tomando como ejemplo a *Version Control System Gis (VCSGIS)*.

## 1.1. Conceptos básicos del control de versiones

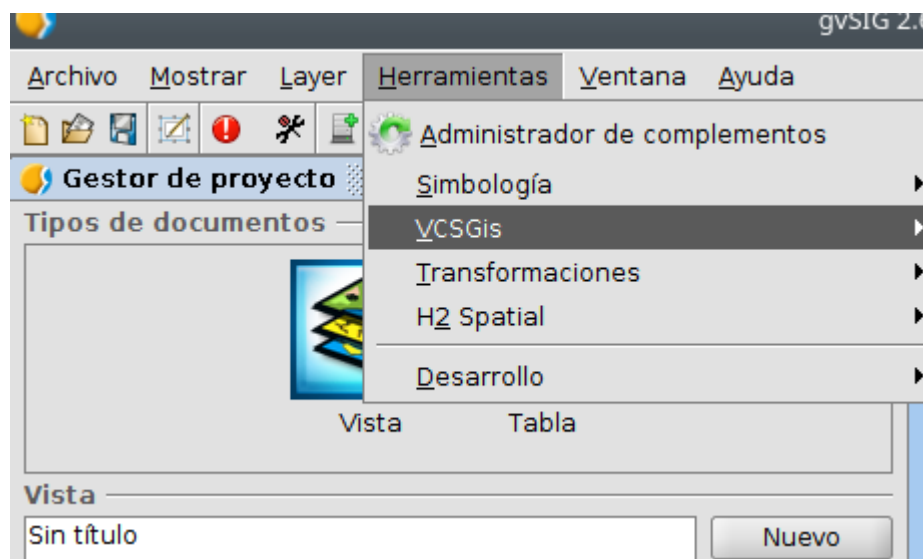
Antes de que entremos a ver como funciona *VCSGIS* es importante tener una visión general de cómo funciona un *Sistema de Control de Version (VCS)* y los términos que se utilizan.

- El **repositorio** *VCSGIS* usa una base de datos central que contiene todos los datos cuyas versiones se controlan y sus respectivas historias. Ésta base de datos se conoce como el **repositorio**. El repositorio normalmente está en un servidor de base de datos, que provee a pedido el contenido a los clientes de *VCSGIS*. Si solo puede hacer una copia de seguridad de una sola cosa, hágala del repositorio, ya que es la copia maestra de toda su información.
- **Copia de trabajo**

Aquí es donde se realiza el trabajo en serio. Cada usuario tiene su propia copia de trabajo, comúnmente conocida como *caja de arena* en su ordenador local. Usted puede obtener la última versión del *repositorio*, trabajar en ella localmente sin perjudicar a nadie, y cuando haya terminado con los cambios que ha realizado puede *confirmar* (commit) sus cambios en el *repositorio*.

Una *copia de trabajo* de VCSGIS no contiene la historia de los datos, pero sí contiene una copia de los datos que existían en el repositorio antes que comience a hacer cambios. Esto significa que es fácil verificar qué cambios ha realizado.

También necesita saber donde encontrar VCSGIS dado que no hay mucho para ver en los menús y herramientas de gvSIG desktop. Esto se debe a que VCSGIS es una complemento de gvSIG desktop, así que primero inicie el gvSIG desktop. Haga clic en el menú “Herramientas” y debería ver una entrada nueva **VCSGIS**:



## Repositorio

Como se dijo en el apartado anterior VCSGIS es un sistema centralizado para compartir información, estando en su núcleo un **repositorio**, es decir un almacén central de datos, que salva la información en forma de tablas. Este almacén no es más que un sistema de base de datos que tiene tanto la información que modifican los usuarios del software, en el caso de VCSGIS tablas o capas, como las tablas que almacenan los cambios producidos en los anteriores.

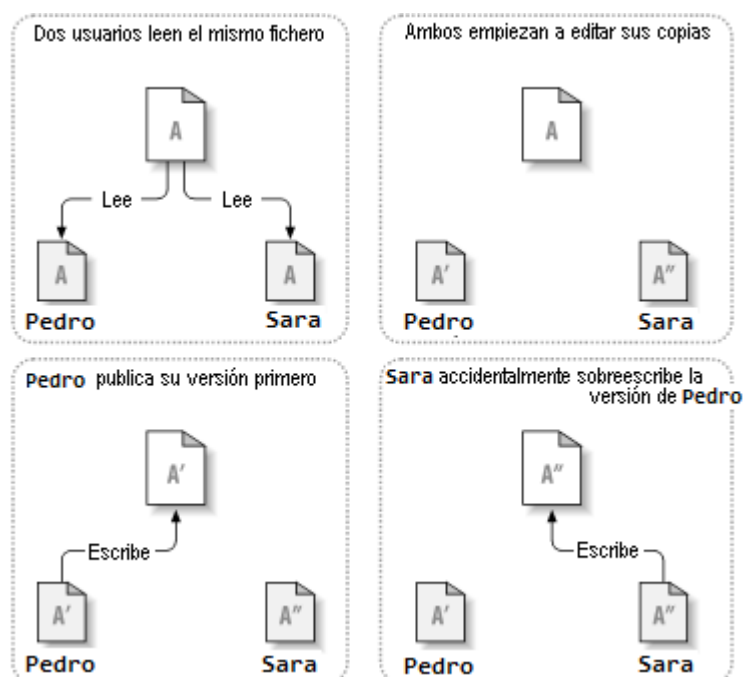
Cada tabla del usuario mantendrá su propio control de versiones de forma independiente a las demás tablas del repositorio, llevando un control de los registros que se van insertando, modificando o borrando en ella. La unidad mínima de información que sigue o controla VCSGIS es el registro de una tabla, sin hacer seguimiento de los cambios en los campos de cada registro.

Cuando un cliente accede a los datos del repositorio en VCSGIS, normalmente ve la última versión de este, es decir la información actual con las capas y tablas. Sin embargo el cliente también tiene la posibilidad de ver los estados previos del repositorio. Por ejemplo, un cliente podría preguntarse *¿Qué contenía esta tabla este miércoles?* o *¿Quién fue la última persona en editar esta capa y qué cambios realizó?* Estas son el tipo de posibilidades que diferencian a los sistemas de control de versiones: sistemas diseñados para guardar y registrar las modificaciones de los datos a lo largo del tiempo.

## Modelos de versionado

Tras la definición de los sistemas de control de versiones y más concretamente *VCSGIS*, así como el núcleo de estos o repositorio, puede intuirse la existencia de un punto crítico en este tipo de software. Este punto no es otro que el intercambio de información actualizada entre usuarios, evitando que los cambios de estos no se pisen unos a otros, ya que parece sencillo pensar que los usuarios de forma accidental pueden sobrescribir cambios de los demás al almacenar la información de los suyos en el repositorio.

Para entender mejor lo anterior consideramos el siguiente escenario. Suponga que tiene dos compañeros de trabajo, Pedro y Sara. Cada uno decide editar la misma tabla del repositorio a la vez. Si Pedro guarda sus cambios en el repositorio primero, es posible que, unos momentos después, Sara pueda accidentalmente sobrescribirlos con su propia versión. Mientras que la versión de la tabla de Pedro no se ha perdido para siempre, porque el sistema recuerda cada cambio, cualquier cambio que Pedro hizo no estará en la versión nueva de la tabla de Sara, porque ella nunca vio los cambios de Pedro. De lo anterior podemos decir que el trabajo de Pedro está efectivamente perdido, o al menos falta en la última versión de la tabla.



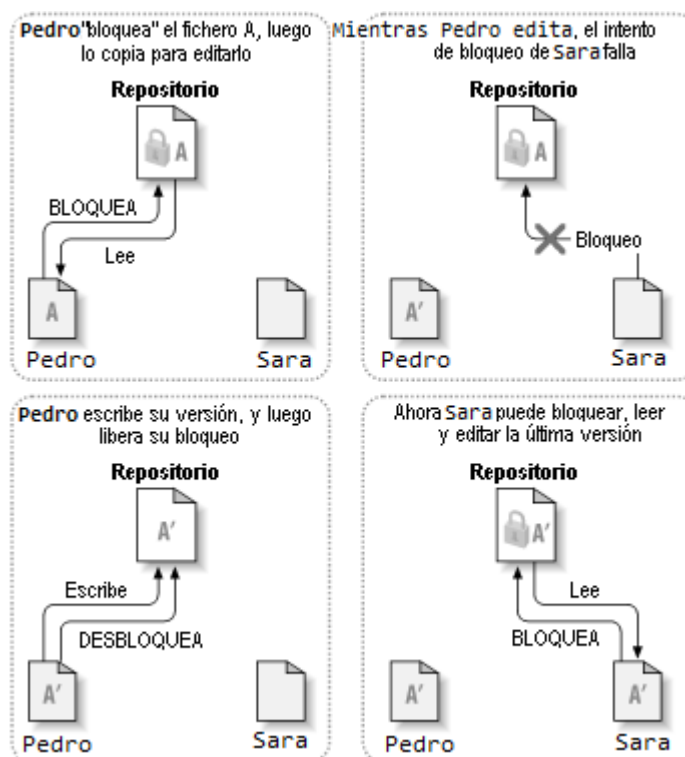
Para solucionar ese problema existen dos modelos;

- Bloquear-Modificar-Desbloquear
- Copiar-Modificar-Fusionar

## Bloquear-Modificar-Bloquear

Muchos sistemas de control de versiones utilizan un modelo Bloquear-Modificar-Desbloquear para enfrentarse al problema anterior, lo cual es una solución muy simple. En estos sistemas, el

repositorio sólo permite que una persona cambie un archivo. Pedro primero debe bloquear el archivo antes que pueda empezar a hacer cambios en él. Si Pedro ha bloqueado un archivo, entonces Sara no puede hacer ningún cambio en él. Si ella intenta bloquear el archivo, el repositorio le denegará la petición. Todo lo que ella puede hacer es leer el archivo, y esperar a que Pedro termine sus cambios y libere el bloqueo. Después que Pedro desbloquee el archivo es el turno de Sara para bloquear y editar.



El problema con el modelo bloquear-modificar-desbloquear es que es un poco restrictivo, y a menudo se convierte en una molestia para los usuarios:

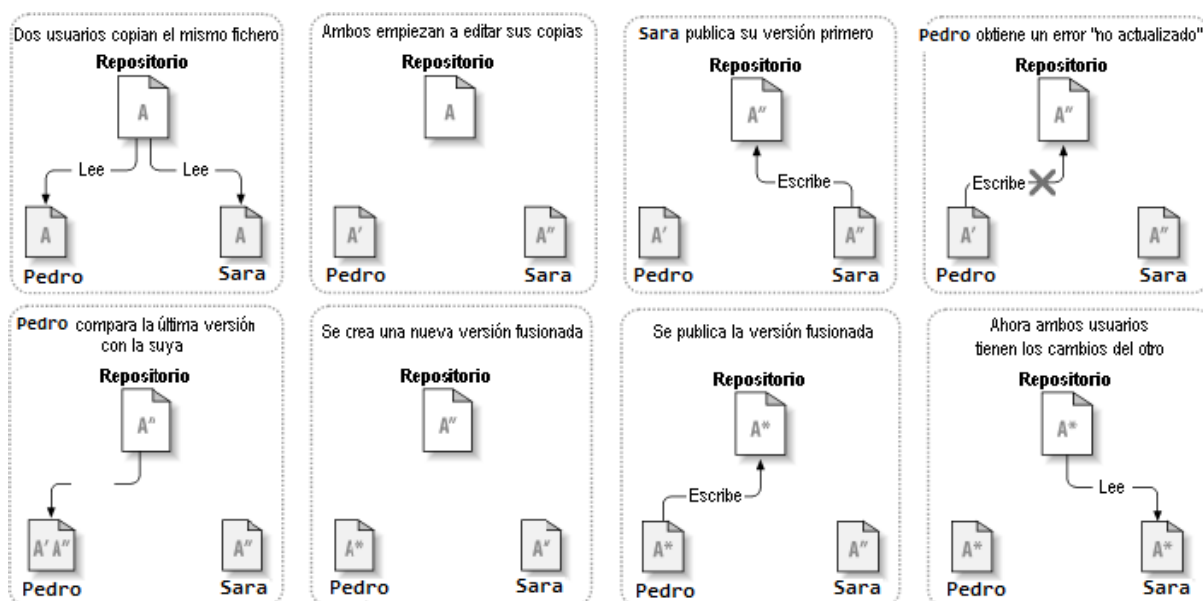
- *El bloqueo causa muchos problemas administrativos.* A veces los usuarios bloquean archivos y se olvidan de desbloquearlos, dejando sin acceso a la edición de estos por parte de los demás usuarios. Para solucionar esto se tiene que gestionar el desbloqueo por parte de un administrador lo cual hace que la situación cause un montón de retraso y pérdidas de tiempo innecesarias.
- *El bloqueo puede causar procesos en serie innecesarios.* ¿Qué ocurre si un usuario está editando el inicio de un archivo de texto, y otro simplemente quiere cambiar la parte final del mismo archivo? Esos cambios no se superponen en absoluto y por tanto se podría editar el archivo de forma simultánea, no existiendo la necesidad de tomar turnos en este tipo de situaciones.
- *El bloqueo puede causar una falsa sensación de seguridad.* Imagine que un usuario bloquea y edita el archivo A, mientras otro usuario distinto simultáneamente bloquea y edita el archivo B. Pero supongamos también que los archivos A y B dependen uno del otro, y que los cambios hechos a cada uno son incompatibles. De repente A y B ya no funcionan juntos y el sistema de bloqueo no tiene forma de prevenir este problema, sin embargo, de alguna forma este sistema dio una sensación de falsa seguridad. Es fácil para los

usuarios imaginar que al bloquear los archivos están realizando una tarea segura y aislada, y la realidad es que no, inhibiéndoles de discutir si sus cambios son compatibles entre sí.

## Copiar-Modificar-Fusionar

VCSGIS y otros sistemas de control de versiones usan el modelo Copia-Modificación-Fusión como alternativa al modelo de bloqueo. En este modelo, cada cliente de los usuarios lee el repositorio y crea una **copia de trabajo** personal de las capas o tablas de modo que los usuarios trabajan en paralelo, modificando sus copias privadas. Finalmente, las copias privadas son unificadas conjuntamente en una nueva versión final mediante el asesoramiento del sistema de control de versiones, pero es un humano en última instancia el responsable de hacer esta acción de manera correcta.

Para entender mejor lo anterior consideramos el siguiente escenario; Digamos que tanto Pedro como Sara crean copias de trabajo de la misma capa, copiadas del repositorio. Ellos trabajan simultáneamente, y hacen cambios al mismo archivo A dentro de sus copias. Sara es la primera en grabar sus cambios en el repositorio. Cuando Pedro intenta grabar sus cambios más tarde, el repositorio le informa que su archivo A está desactualizado. En otras palabras, que el archivo A en el repositorio ha cambiado de alguna forma desde la última vez que lo copió. Por lo que Pedro le pide a su cliente que fusione cualquier nuevo cambio del repositorio dentro de su copia de trabajo del archivo A. Lo más seguro es que los cambios de Sara no se superpongan a los suyos; por lo que una vez que ambos conjuntos de cambios se han integrado, él graba su copia de trabajo de nuevo en el repositorio.



¿Pero qué ocurre si los cambios de Sara se superponen a los cambios de Pedro? ¿Qué hacemos entonces? La situación se denomina un **conflicto**, y normalmente no es problema. Cuando Pedro le pide a su cliente que fusione los últimos cambios del repositorio en su copia de trabajo A, se da el caso anterior, la copia de trabajo de Pedro marca que está en un estado de conflicto. En ese estado de conflicto Pedro es capaz de ver ambos conjuntos de cambios conflictivos, y

manualmente podrá elegir entre ellos cual es el más acertado. Tenga en cuenta que el software no puede resolver conflictos automáticamente, ya que sólo los humanos son capaces de entender y hacer las elecciones necesarias de forma correcta. Una vez que Pedro haya resuelto manualmente los cambios que se superponían, puede guardar de forma segura el archivo fusionado al repositorio.

El modelo Copiar-Modificar-Fusionar puede parecer un poco caótico, pero en la práctica, funciona extremadamente bien. Los usuarios pueden trabajar en paralelo, sin que tengan que esperar nunca uno por otro, y cuando trabajan en los mismos archivos, resulta que la mayoría de los cambios concurrentes no se superponen en absoluto, siendo los conflictos muy poco frecuentes. Además, el tiempo que lleva resolver conflictos es mucho menor que el tiempo perdido por un sistema que implementa el modelo de bloqueo.

Como curiosidad decir que hay una situación donde el modelo Bloquear-Modificar-Desbloquear resulta mejor que este, y es cuando se tiene archivos no-fusionables, por ejemplo imágenes. Si dos personas cambian una imagen a la vez, no hay forma de fusionar esos cambios, y uno de ellos perderá sus cambios.

## **Copias de trabajo**

A modo de recordatorio decir que el sistema de control de versiones VCSGIS utiliza el modelo Copia-Modificación-Fusión como modelo de versionado, por lo que utiliza **copias de trabajo**.

Una copia de trabajo en VCSGIS esta formada por una base de datos en su sistema local, la cual contiene una colección de tablas o capas que pueden ser modificados sin ningún tipo de problema. Su copia de trabajo es su área de trabajo privada y VCSGIS nunca incorporará los cambios de otra gente, ni hará que sus cambios estén disponibles para los demás, a menos que se lo pida expresamente. Básicamente la copia de trabajo es una captura/fotografía/snapshot del repositorio en un instante concreto.

Una copia de trabajo de VCSGIS también contiene algunas tablas extra, creados y mantenidos por VCSGIS, para ayudar a llevar a cabo la gestión de cambios. En estos se almacenan los numero de revisión, un sistema de numeración única que identifica de manera inequívoca los diferentes cambios registrados en el repositorio. Estos codigos de revision, únicos para cada capa o tabla, ayudan al software a reconocer qué capas o tablas contienen cambios no publicados o qué capas o tablas contienen contenidos desfasados respecto a la información del repositorio.

## **Flujo de trabajo (Modelo Copiar-Modificar-Fusionar)**

Este apartado define la estructura o forma de trabajar teórica con un sistema de control de versiones que presenta el modelo Copiar-Modificar-Fusionar como podría ser VCSGIS. Especificar que estos mismos procesos serán explicados desde un punto de vista práctico para VCSGIS en el apartado siguiente.

**Paso 1:** Creación de repositorio. Se lleva a cabo por el administrador y con el se da inicio al proceso.

**Paso 2:** Creación de copia de trabajo. Proceso realizado por los diferentes usuarios o clientes del repositorio y como se dijo en el apartado anterior consiste en la captura del repositorio en un instante determinado. Esa captura completamente funcional permite cualquier acción de edición o modificación y los cambios realizados en esta serán propuestos por el usuario para formar parte del repositorio.

**Paso 3:** Realización de cambios en la información de su copia de trabajo y verificación que funcionan correctamente

**Paso 4:** Publicación de cambios. El sistema de control de versiones VCSGIS en este caso provee al usuario de comandos para publicar sus cambios en el repositorio por tanto estar estos disponibles para todos los usuarios registrado es en ese repositorio. En el caso de que los demás usuarios hayan publicado antes sus propios cambios, el software le provee de comandos para fusionar esos cambios dentro de su copia de trabajo tras leer el repositorio. La fusión puede presentar dos caso bien diferenciados:

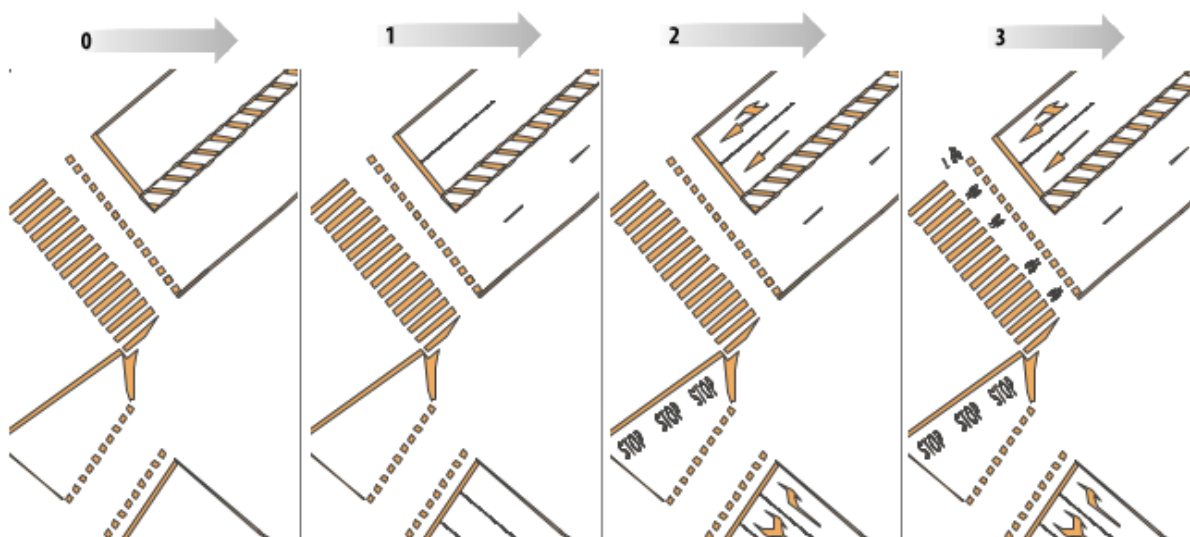
- Los cambios del usuario afectan a elementos no modificados anteriormente por los demás usuarios y por tanto se mezclan y fusionan las tablas.
- El usuario ha realizado cambios sobre elementos modificados anteriormente por otros usuarios y ya registrados en el repositorio, en ese caso se entra en conflicto y el usuario decide que cambios son los que se quedan finalmente en el repositorio los propios o los ya presentes.

Al acto de publicar los datos en el servidor se le denomina hacer **confirmar**, *commit*, y al proceso de actualizar la copia de trabajo con la información del repositorio se le denomina **actualizar**, *update*.

## Revisiones

Volviendo al proceso de publicar o *confirmar* los datos en el servidor, hacer *commit*, este nos genera como resultado una **revisión** a la cual esta ligada cualquier cambio realizado en la tabla de la copia de trabajo modificada; cambiar el contenido de las tablas, crear, borrar, renombrar y copiar registros o geometrías... Siendo solo registrada en el repositorio si todos los cambios de esa tabla pueden realizarse (transacción atómica). En el caso de que algún cambio de una tabla no pueda hacerse, ninguno de los demás cambios de la tabla referentes a esa revisión se realizará. Las revisiones presentan un identificador único de la revisión y en el caso de VCSGIS cada tabla de la copia de trabajo presentara un control de la revisión individual con identificadores únicos propios.

Para ilustrar lo anterior se presenta el siguiente esquema del funcionamiento de las revisiones;



En la primera revisión, revisión 0, se han creado las líneas base de la calzada, en la siguiente revisión, revisión 1, se añaden las separaciones entre carriles y medianeras, en la revisión 2 se añaden las flechas indicadores del sentido y señales de stop, y por ultimo, en la revisión número 3 se han añadido las señales del carril bici.

Tras la explicación anterior parece lógico pensar que las revisiones y su número o identificador único son el mecanismo que utilizan los sistemas de control de versiones y por tanto *VCSGIS* para gestionar el estado de la copia de trabajo con respecto al repositorio y viceversa.

## Estados de tablas y registros

Los estados que puede presentar la copia de trabajo y las diferentes capas y tablas que los componen son las siguientes;

- *Sin cambios y actualizado.* La tabla no se ha cambiado en la copia de trabajo y no se han confirmado cambios a esa tabla en el repositorio desde su revisión de trabajo. Una acción *commit* de esa tabla no hará nada, y una actualización de ella tampoco.
- *Cambiado localmente y actualizado.* La tabla ha sido cambiado en la copia de trabajo y no se ha confirmado ningún cambio a esa tabla en el repositorio desde su revisión base. Hay cambios locales que no se han confirmado en el repositorio, por lo que al hacer *commit* la tabla se confirmarían sus cambios actualizando el repositorio. Si se procede a hacer una actualización o *update* de la tabla no se realizará ya que la copia de trabajo actual posee la versión mas moderna de esta presente en el repositorio.
- *Sin cambios y desactualizado.* La tabla no ha sido cambiado en la copia de trabajo, pero ha sido cambiado en el repositorio. La tabla deberá ser actualizado en algún momento para presentar el mismo contenido que la versión del repositorio. Un comando *commit* sobre la tabla no hará nada ya que no hay cambios en local y el comando actualizar o *update* traerá los últimos cambios del repositorio a su copia de trabajo.

- *Cambiado localmente y desactualizado.* La tabla se ha cambiado tanto en la copia de trabajo como en el repositorio. No podrá ejecutar un *commit* sobre la tabla ya que necesita actualizar en primer instancia la copia de trabajo con el repositorio. Para ello necesita realizar una actualización o *update* de la copia de trabajo que intentará fusionar los cambios del repositorio con los cambios locales. Si *VCSGIS* no puede completar la fusión de una forma plausible automáticamente, le dejará al usuario la tarea de resolver los conflictos. Por ultimo, una vez actualizado, se deberá realizar una petición *commit* para registra los cambios locales en el repositorio y terminará el proceso.

## Conexión con el repositorio

El sistema de control de versiones VCSGIS dispone de dos opciones a la hora de realizar la conexión con el repositorio;

- Conexión a una BBDD de nuestra red local.
- Conexión a un servidor VCSGIS.

La primera opción y más usual consiste realizar la conexión a un repositorio situado en un servidor local y se realiza identificando el fichero de la BBDD en la estructura de carpetas. La segunda opción al ser online permite la conexión al repositorio mediante URL, ampliando la operabilidad del software.

## 2 UTILIZACIÓN BÁSICA

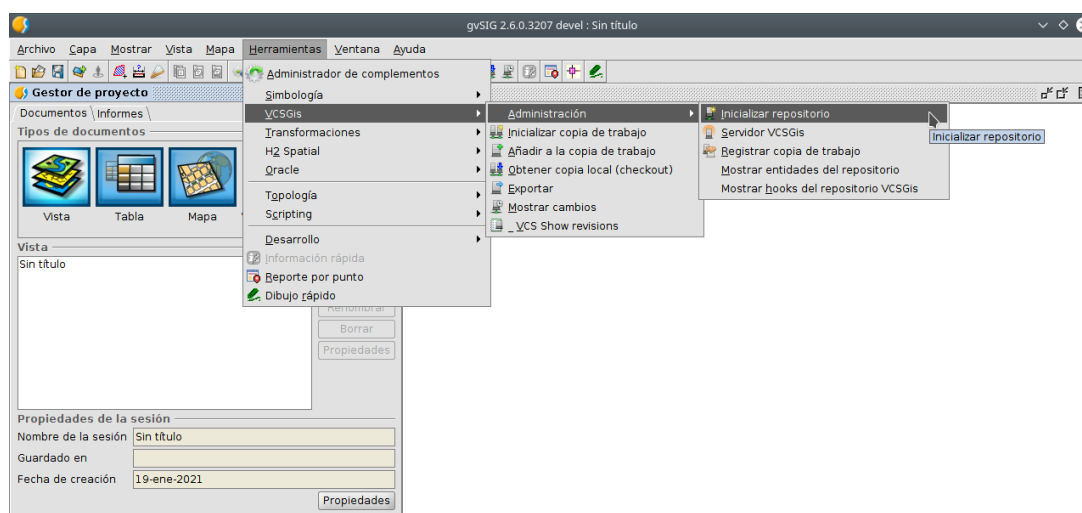
### Introducción

A continuación, como se mencionó en el Apartado *Flujo de trabajo (Modelo Copiar-Modificar-Fusionar)* se realiza un aproximación más práctica a la manera o forma de trabajar que tiene el software VCSGIS. Para llevar a cabo lo anterior se seguirá el flujo de trabajo indicado en el apartado antes mencionado, adaptando los puntos a nuestro sistema de control de versiones.

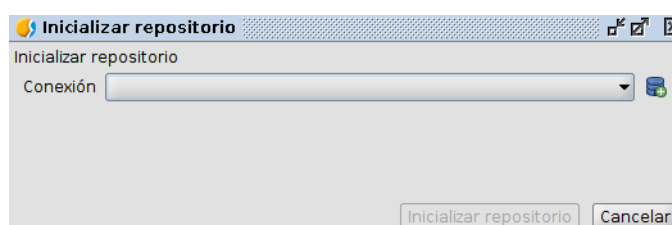
### Creación de un repositorio

Para la creación del repositorio tal y como se destaca anteriormente en el documento disponemos de dos opciones, conexión local a un BBDD como repositorio o conexión a un servidor o repositorio remoto (online). En esta guía nos inclinaremos por la primera opción o conexión local.

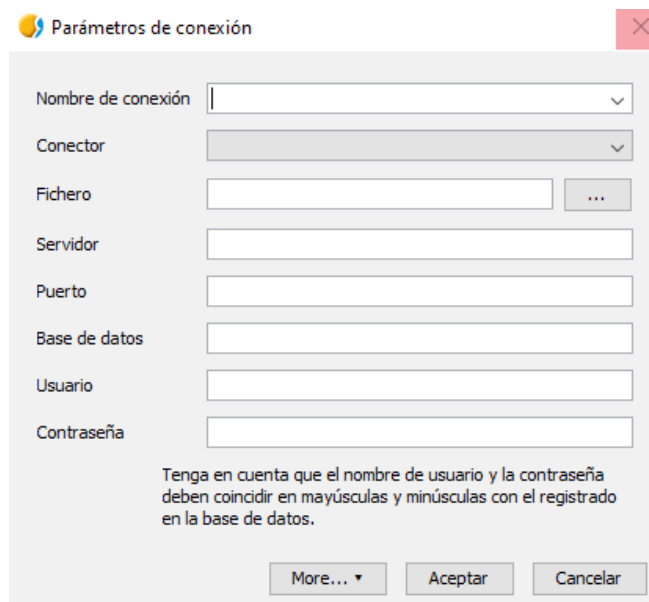
Para inicial la conexión al repositorio hay que dirigirse a la pestaña *Herramientas*, desplegarla y buscar la opción *VCSGIS*. Una vez dentro seleccionaremos la opción *Administración* y dentro de esta *Inicializar repositorio*. Puede ver gráficamente lo mencionado anteriormente en la siguiente ilustración.



Tras pulsar se obtiene la siguiente ventana la cual nos permite seleccionar una conexión a una base de datos ya existente o generar una nueva.



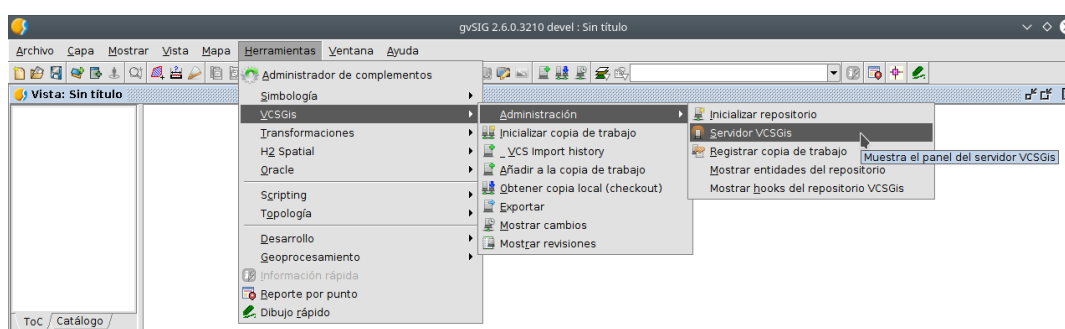
En el caso de que se busque crear una nueva conexión hay que hacer clic sobre el botón situado a la derecha del desplegable en la ventana anterior, acción que iniciará el proceso de creación de una nueva conexión a una BBDD. Esta nueva conexión se realiza mediante la ventana genérica de conexiones a este tipo de datos de gvSIG Desktop, ver la figura siguiente.



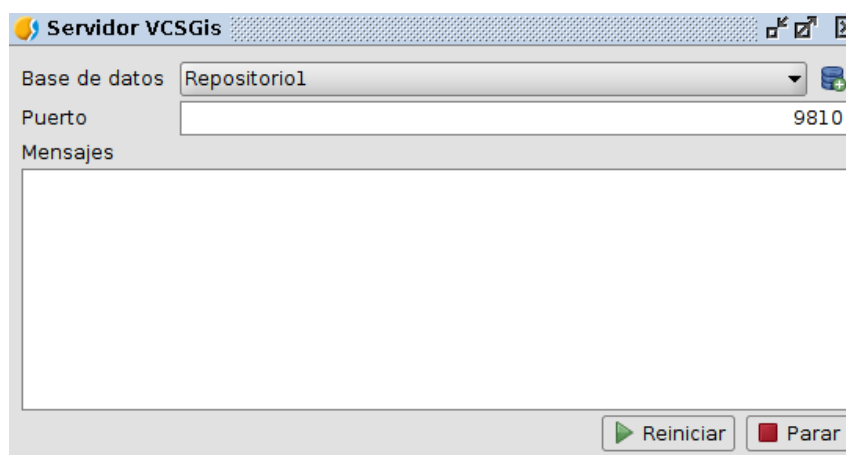
Como se detallo en la explicación teórica, la creación es un proceso destinado a ser ejecutado por el administrador, un usuario regular no debería llevar a cabo este proceso.

## Iniciar servidor VCSGIS

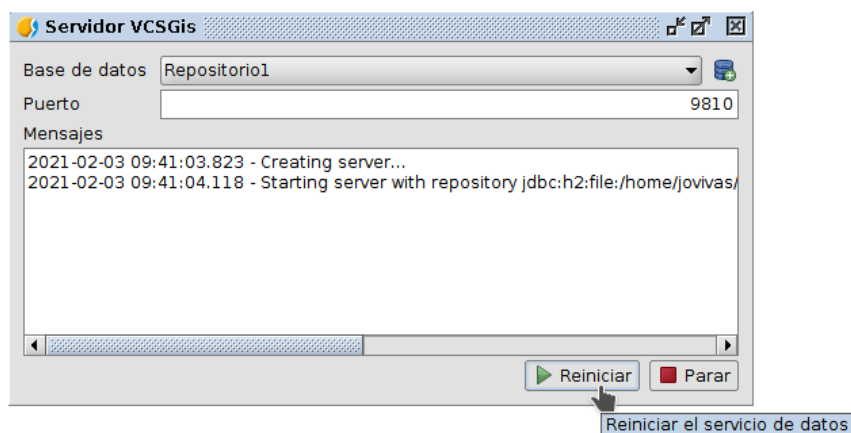
Como se dijo en el apartado anterior, la herramienta de control de versiones VCSGIS dispone de la posibilidad de suministrar un repositorio local a otras máquinas mediante la inicialización de este en un servidor propio en gvSIG desktop. En otras palabras VCSGIS permite que un único equipo almacene el repositorio y este suministre la información a los demás equipos mediante una conexión (URL) al repositorio remoto. Esta opción se puede configurar ejecutando el comando *Servidor VCSGIS* situado dentro de *Herramientas*, desplegable *VCSGIS*, desplegable *Administración*.



Tras la ejecución del comando anterior se obtiene el siguiente interfaz.

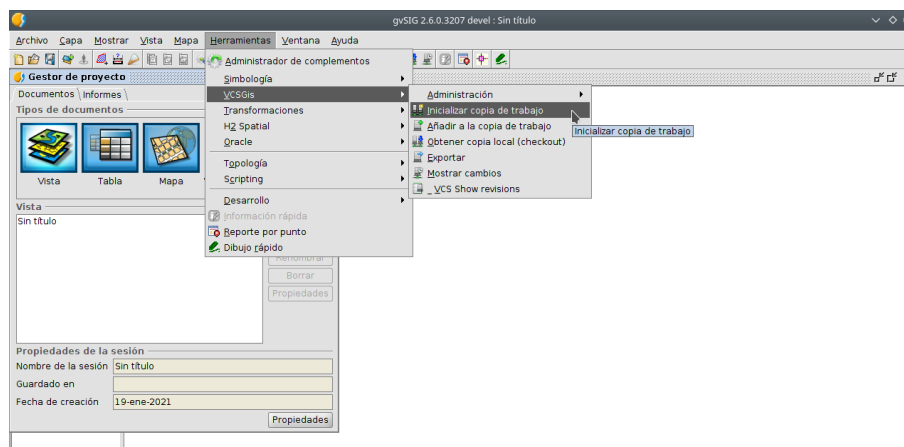


En esta ventana hay que especificar la base de datos o repositorio que va a ser suministrado mediante conexión remota así como el puerto por el cual se va a enviar la información. Una vez especificado lo anterior solo queda iniciar el proceso pulsando el botón *Reiniciar* situado en la zona inferior de la ventana. Puede también pararse el suministro de dicha información con el botón *Parar* adyacente al anterior. Tras iniciar el proceso la ventana en la zona central muestra las peticiones realizadas al servidor.

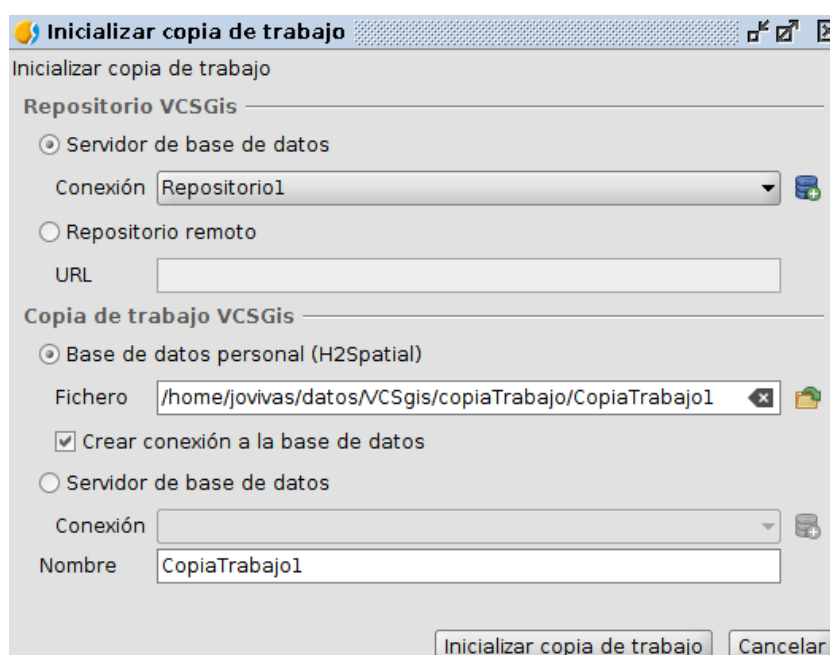


## Creación de una copia de trabajo

El siguiente paso es crear la copia de trabajo, proceso realizado al ejecutar el comando *Inicializar copia de trabajo* dentro de la pestaña *VCSGIS* situada en la opción *Herramientas* de gvSIG Desktop.



Una vez ejecutado lo anterior se obtiene la siguiente venta cuyos componentes se dividen en dos apartados, *Repositorio de VCS* y *Copia de trabajo de VCS*.

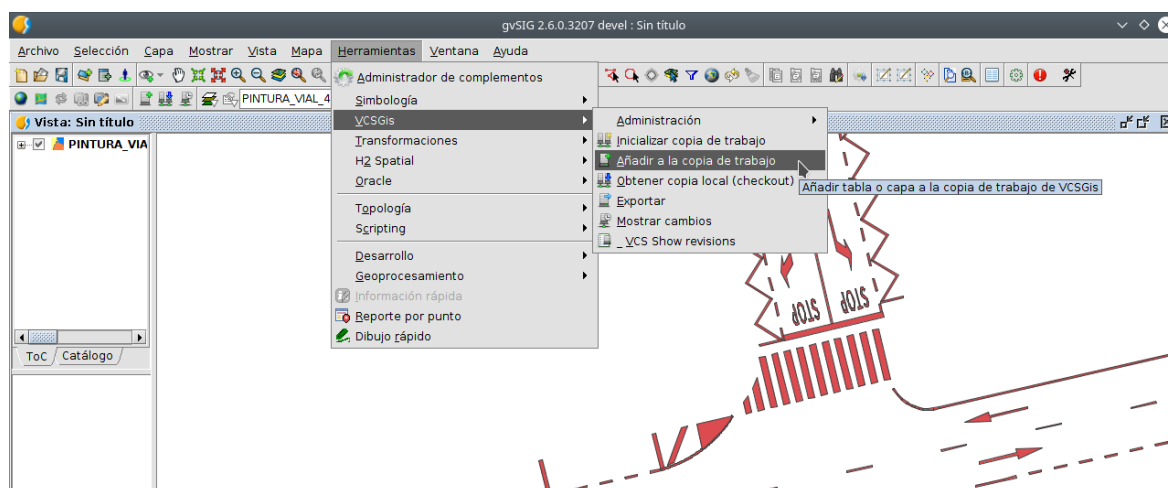


En el apartado *Repositorio del VCS* se selecciona la conexión al repositorio, pudiendo ser local u online, mientras que el apartado *Copia del trabajo VCS* se introducen los parámetros para generar la nueva copia de trabajo. Los parámetros necesarios son el nombre de la copia y la situación de esta, la cual puede ser local, genera un BBDD personal en formato H2Spatial, u online. Si se selecciona la opción local, marcar la opción *Crear conexión a la base de datos* como recomendación.

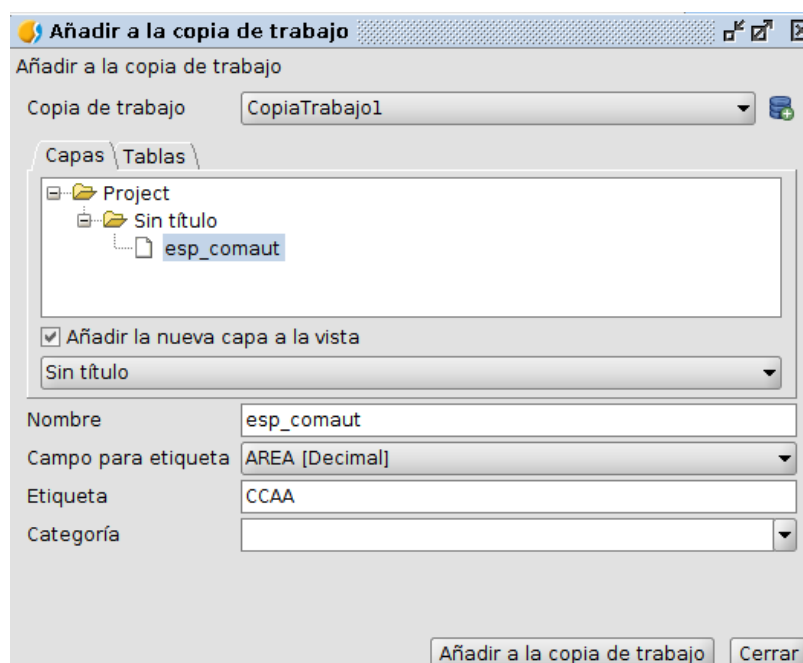
## Añadir una capa al repositorio

Tras cerciorarnos de la existencia de un repositorio y de una copia de trabajo se procede a actualizar este con nueva información. Para realizarlo hay que cumplir dos pasos.

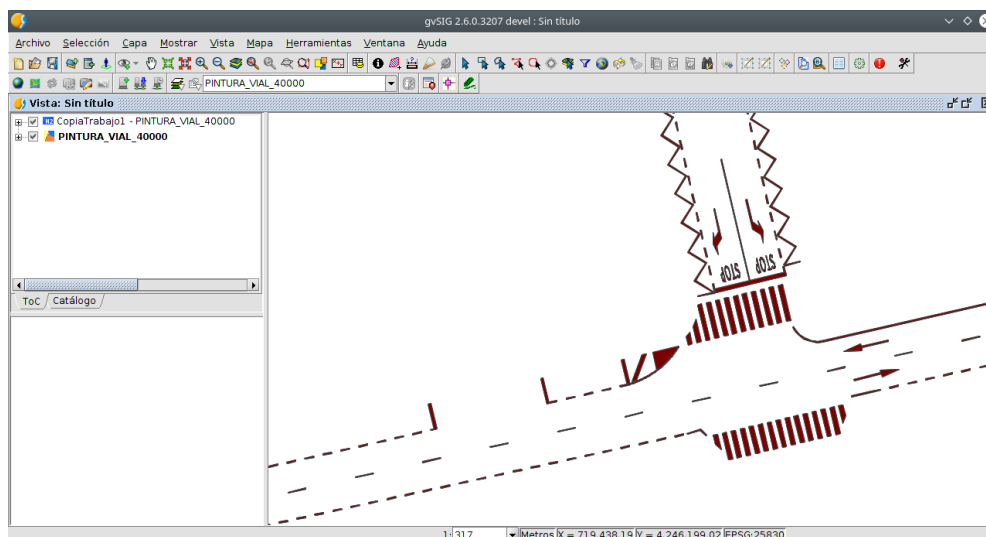
El paso 1 consiste en cargar dicha nueva información o capa en la vista. El segundo paso se basa en ir a la opción *Herramientas* del menú de gvSIG Desktop, pestaña *VCSGgis* y pestaña *Añadir a la copia de trabajo*.



Tras pulsar el comando *Añadir a la copia de trabajo* se desplegará la ventana siguiente donde se selecciona la copia de trabajo donde queremos añadir la capa, la estructura de carpetas actual de gvSIG donde se selecciona la información a añadir y una serie de opciones.

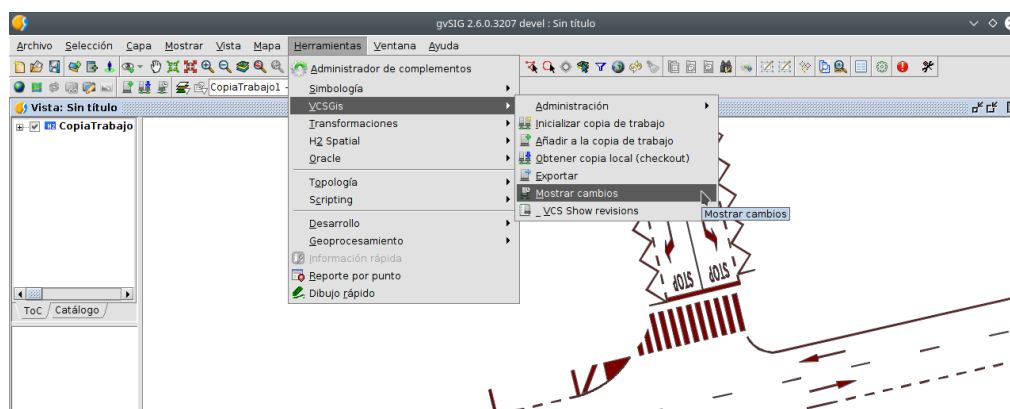


Entre las opciones destaca *Añadir esa capa a la vista*, opción recomendada para iniciar el proceso de edición utilizando el control de versiones. Hay que hacer incapie en la idea anterior, la capa añadida recientemente es sobre la que hay que trabajar ya que es la que tiene un control de versiones asociado, la capa inicial usada para introducir el dato puede eliminarse de la vista ya que los cambios sobre ella no se registran en VCSGIS.

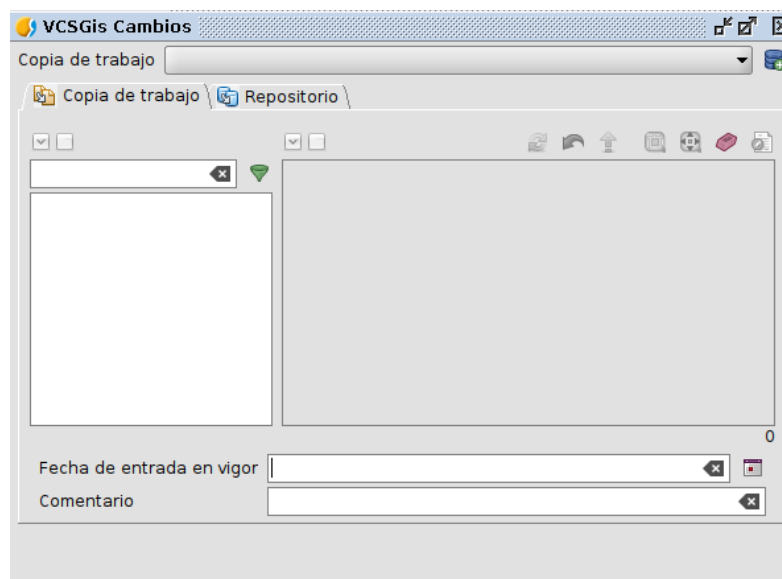


Las opciones restantes son nombre, campo para etiquetas, etiqueta y categoría. Estas opciones permiten respectivamente renombrar la capa en el árbol de la vista, seleccionar que campo queremos que sea representado como campo de etiquetas respectivamente, crear una etiqueta referida a esa capa y asignar la capa a una categoría definida por el usuario.

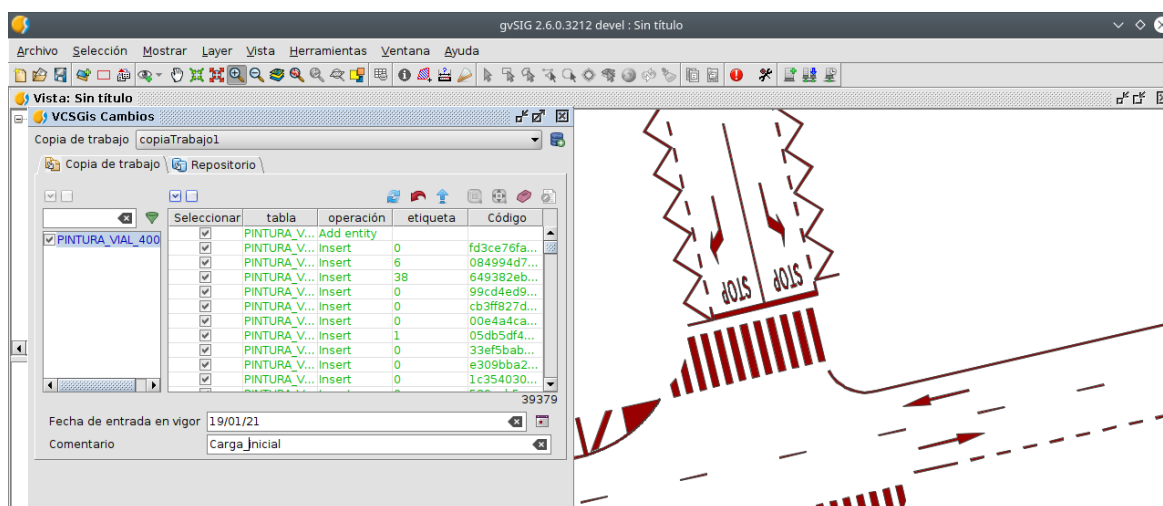
El proceso de añadir la nueva capa al repositorio termina cuando tras realizar la carga de información anterior se ejecuta la herramienta *Mostrar cambios* situado en la pestaña *VCSGIS* dentro de la opción *Herramientas* de del software.



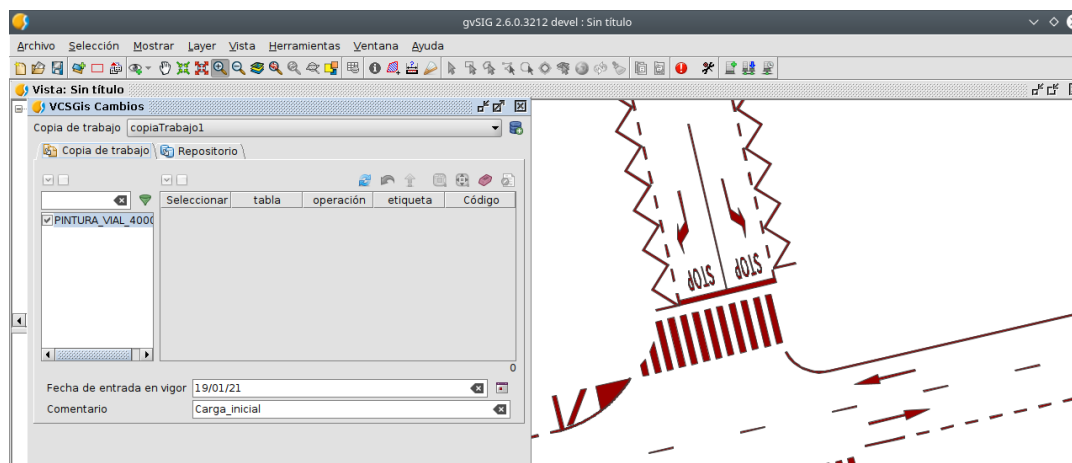
La ejecución de lo anterior genera la siguiente ventana.



Tras ejecutar la herramienta de *Mostrar cambios* y al pulsar la opción *Local* en el área destinada a estos aparecen registros correspondientes a los elementos de la capa a añadir, ver siguiente imagen. Esto se debe a que hay diferencias entre la copia de trabajo y el repositorio, presentando la copia de trabajo una serie de nuevos elementos, una capa, que el repositorio carece. El proceso de añadir una nueva capa al repositorio finaliza si seleccionamos esos registros y pulsamos el botón que realiza un *commit*, componente 8 de la ventana. Con esa acción el repositorio se actualiza con la información de la copia de trabajo y tendría por tanto la nueva capa a su disposición.



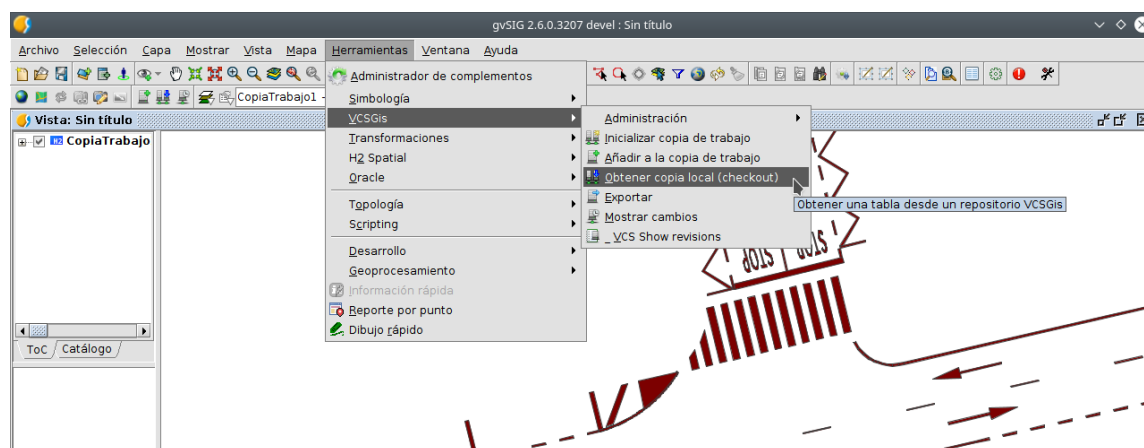
A modo de comprobación tras lo anterior si se selecciona la pestaña *Local* y se pulsa el botón de actualizar o refrescar área de visualización (componente 6 de la ventana), esta aparecerá vacía.



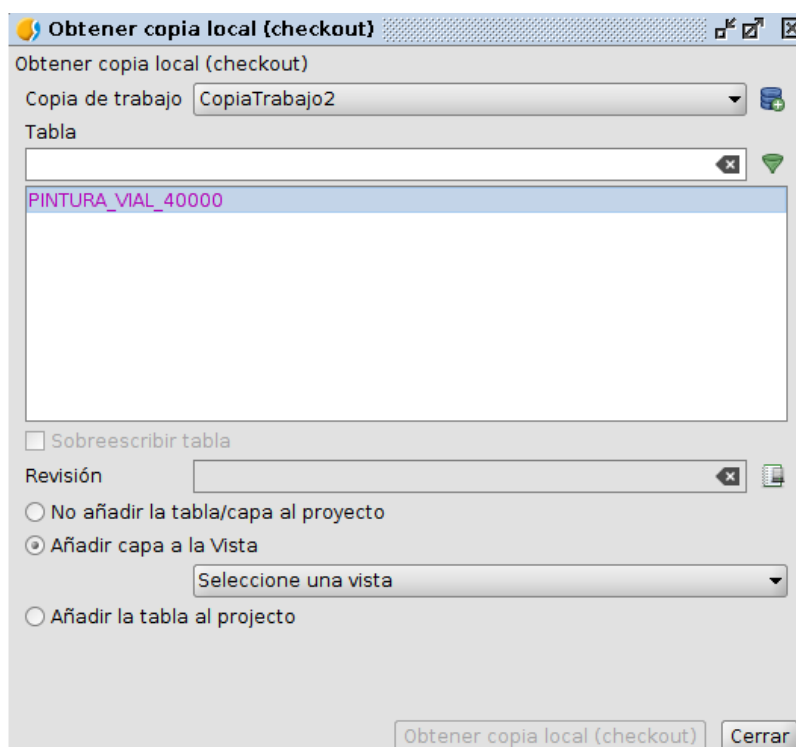
## Añadir una capa del repositorio

El procedimiento de añadir una capa del repositorio a la copia de trabajo puede llevarse a cabo de dos maneras diferentes.

La primera forma, muy similar a *Añadir capa al repositorio*, se realiza ejecutando el comando *Obtener copia local* (checkout) situado en la opción Herramientas del menú de *gvSIG Desktop*, pestaña *VCSGIS*.



Como resultado de la ejecución anterior se obtiene la siguiente ventana:



En la ventana hay que indicar la copia de trabajo a la que se quiere añadir la capa procedente del repositorio y la capa en cuestión, seleccionando esta de la lista presente. Además de lo anterior, la ventana presenta varias opciones en función si la tabla/capa ya se encuentra en la *Copia de Trabajo*.

Si la Tabla/capa se encuentra ya en la copia en cuestión, se puede utilizar la opción *Sobreescribir tabla*. dicha opción borra la información local de la tabla y vuelve a descarga la tabla desde el repositorio. En el caso de que la tabla nunca haya existido en la copia la opción sobrecribir aparecerá deshabilitada y se podrá cargar esta cumplimentando el resto de parámetros; *No añadir la tabla/capa al proyecto*, *Añadir capa a la vista* y *Añadir la tabla al proyecto*.

Otra cosa a destacar es la posibilidad de cargar la capa en la copia para una determinada revisión, para ello se dispone de un parámetro específico.

Hacer hincapié también en una idea importante, si en el proceso de obtener una copia local, es decir una capa del repositorio no se marca la opción *Añadir capa a la vista*, seleccionando la vista deseada, la capa se almacena en la copia de trabajo pero no se representará en la vista. Llegado a este punto se tendría que cargar la capa mediante mediante el diálogo estándar de *Añadir capa* de gvSIG Desktop.

La segunda manera de realizar el proceso se realiza con el diálogo genérico de *Añadir capa* de gvSIG Desktop. Esta forma de proceder puede verse en el apartado **Añadir capa VCSGIS usando el diálogo "Añadir Capa" de gvSIG Desktop**.

## Ciclo de trabajo básico

## Introducción

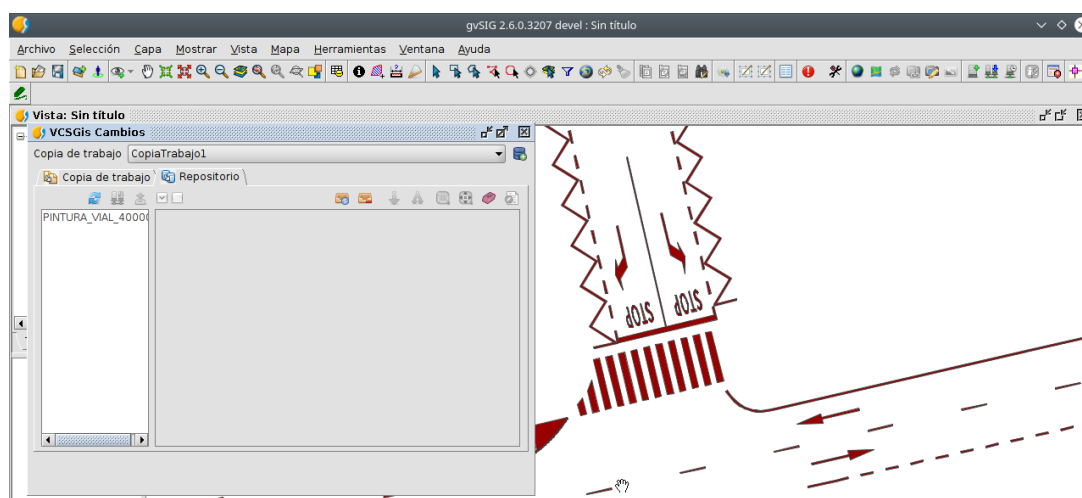
Una vez que tenemos una copia de trabajo con las tablas del repositorio que necesitemos ya podemos comenzar a trabajar. Los pasos típicos que se siguen cuando se trabaja el bajo control de versiones son:

- Actualizar la copia de trabajo (update).
- Realizar cambios. Es decir, trabajar con las capas normalmente.
- Revisar los cambios.
- Arreglar tus errores.
- Mezclar las capas con los cambios que se hayan realizado mientras trabajabas. Posiblemente haya que arreglar conflictos.
- Enviar los cambios al repositorio.

Y cada vez que se vuelve al trabajo, se repite este ciclo.

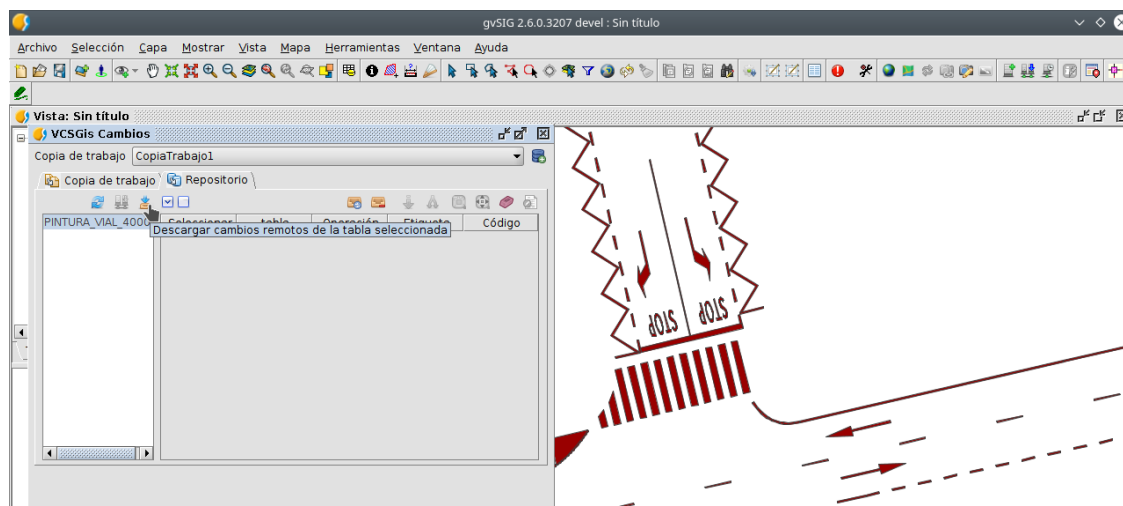
## Actualizar la copia de trabajo

Para actualizar la copia de trabajo hay que ejecutar el comando *Mostrar cambios* y una vez en la ventana con el mismo nombre seleccionar la pestaña *Remoto*; quedando la ventana en cuestión como en la siguiente figura.



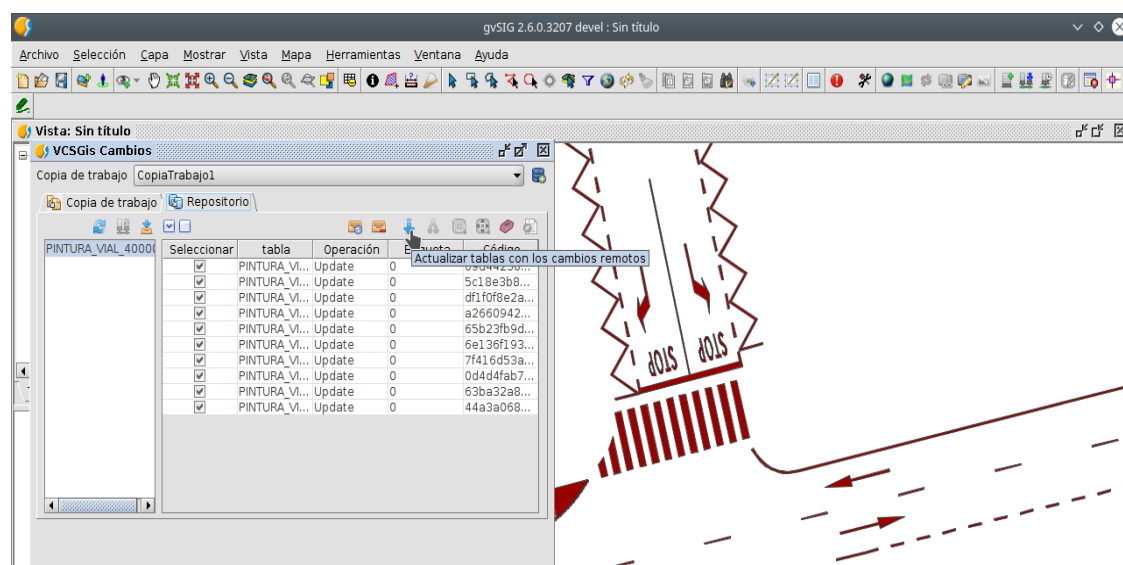
Si no existen cambios en el repositorio con respecto a la copia de trabajo, las capas de la lista de capas de la *Copia de trabajo* aparecerán con una tipografía sin negrita en el componente 3 de la ventana (área con lista de capas de la *Copia de trabajo*) indicando que la *Copia de trabajo* está actualizada; pero si existen cambios, estas aparecerán en negrita estando la copia desactualizada.

Para actualizarla hay que seleccionar la capa en negrita y pulsar el botón para descargar los cambios o diferencias del repositorio frente a la *Copia de trabajo*, componente 5 de la ventana.



La acción anterior provocará que los cambios se descarguen del repositorio y se muestren en el área de visualización de cambios, componente 6. Estos cambios aun no se habrán aplicado sobre las tablas del usuario. Para realizar esto deberán usarse los botones de **actualizar** (Update) o **mezclar** (Merge) cambios en la *Copia de trabajo*, componentes 10 y 11 de la ventana.

!!! note *Actualizar (Update)*; Se actualizarán las tablas locales con los cambios del repositorio. Si hay cambios locales se perderán. *Mezclar (Merge)*; Se mantendrán los cambios remotos para los registros seleccionados, y los cambios locales para los registros no seleccionados.

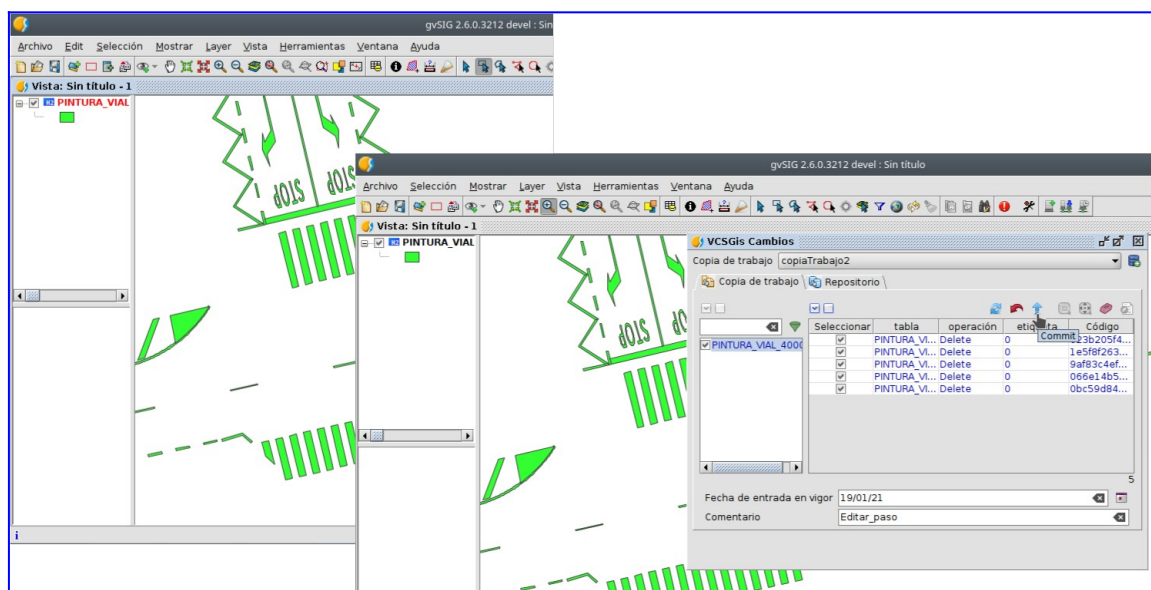


## Revisando y enviando los cambios realizados

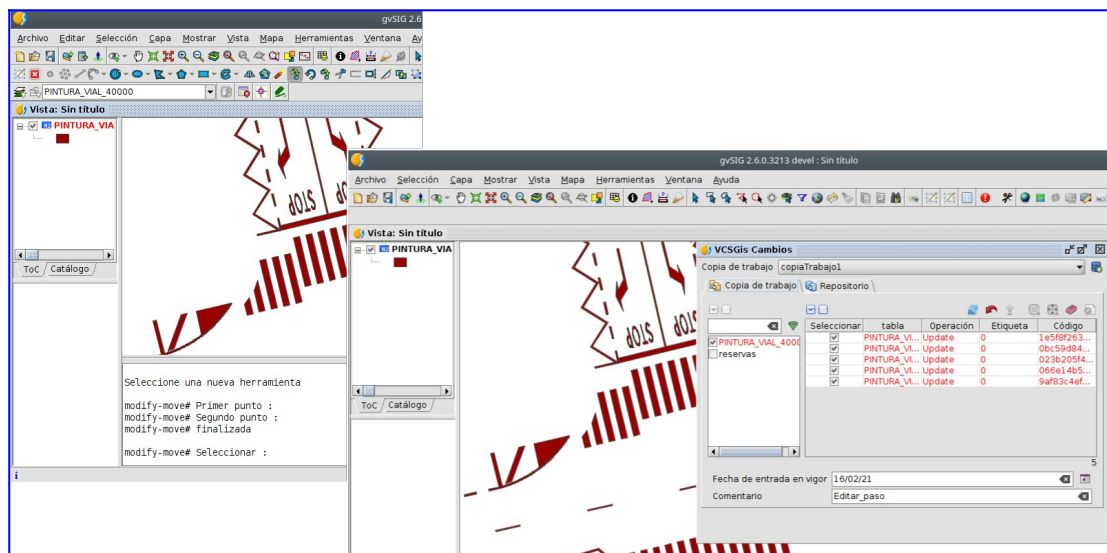
Para explicar mejor este comportamiento se va a usar el siguiente ejemplo:

Un usuario (Pedro) modifica una serie de elementos de una capa de su copia de trabajo (CopiaTrabajo2) llamada PINTURA\_VIAL, tras terminar edición (eliminar elementos) ejecuta el comando Mostrar cambios. Como resultado al seleccionar la opción Local de la ventana Mostrar

cambios, se muestran una serie de registros correspondientes a los citados cambios y solo tendría que pulsar el botón confirmación (Commit) para actualizar esos cambios de su copia de trabajo en el repositorio. Proceso que termina realizando.

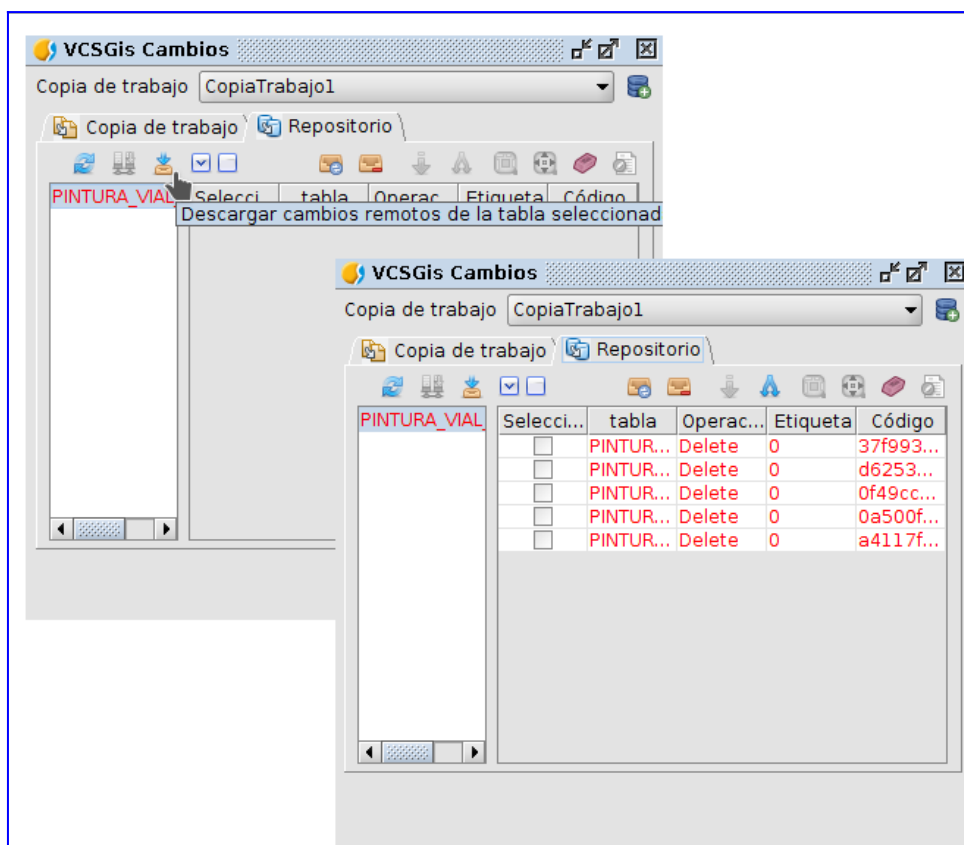


Paralelo al trabajo del usuario Pedro, otro usuario (Sara) trabaja en su copia de trabajo (CopiaTrabajo1) en la misma capa que Pedro y justamente en los mismos elementos que este modificando su posición. Tras terminar la edición procede a ejecutar el comando Mostrar cambios para subir sus cambios al repositorio, pero Pedro ya ha actualizado el repositorio con sus cambios previamente. De modo que el resultado que arroja la ventana de Mostrar cambios es el siguiente;

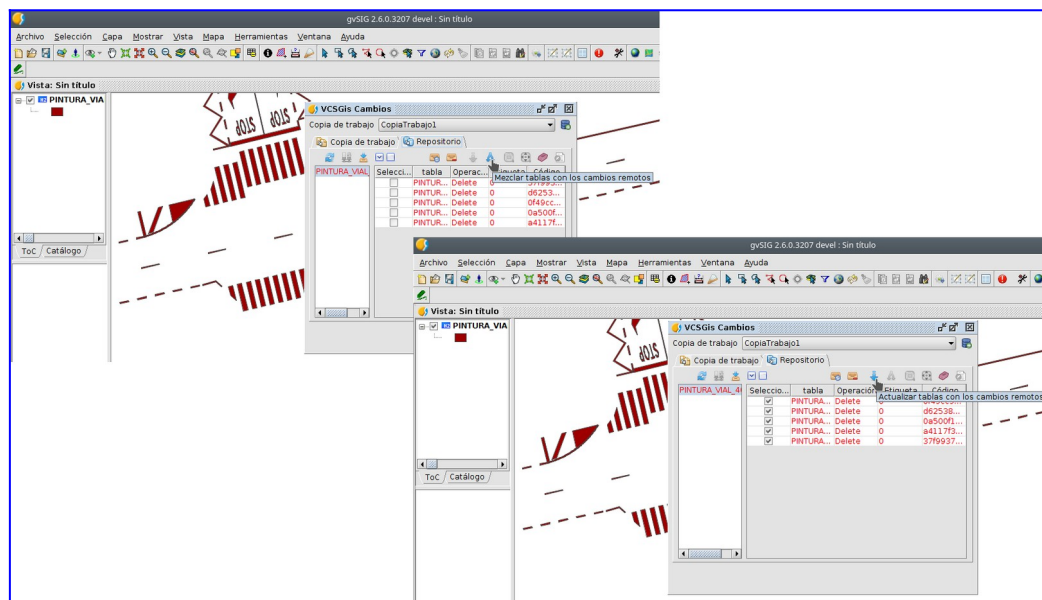


En primer lugar Sara tiene que actualizar la copia de trabajo con respecto al repositorio tal y como se indica en el apartado anterior y tras eso hacer un Commit, pero existe un problema sus cambios entran en conflicto con los de Pedro, de ahí el color rojo de estos. Si no entrara en conflicto tras la actualización de la copia de trabajo podría realizar el Commit sin problema.

Para solucionar el problema Sara tiene que hacer clic sobre la pestaña Repositorio, descargar los cambios del repositorio.

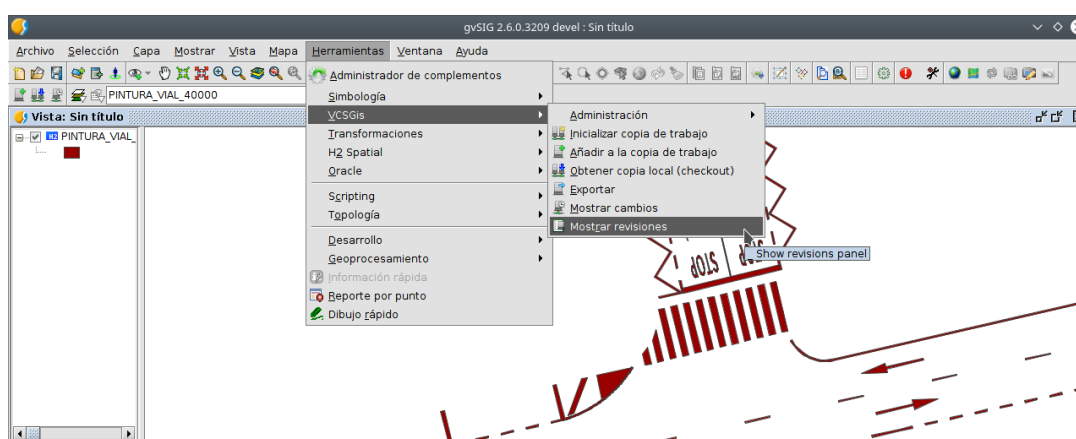


La ventana Mostrar cambios contiene ahora los cambios del repositorio, es decir los de Pedro y por tanto Sara tiene que decidir que cambios de este aceptar como buenos y cuales no, marcando con el check, mediante la opción mezclar, componente 11. Es importante destacar que también puede descartar los cambios de Pedro totalmente si realiza un Merge sin marcar ninguno de los cambios de la lista de cambios del repositorio o los suyos si marca todo los cambios del repositorio y realiza un Merge o si realiza simplemente un Update.

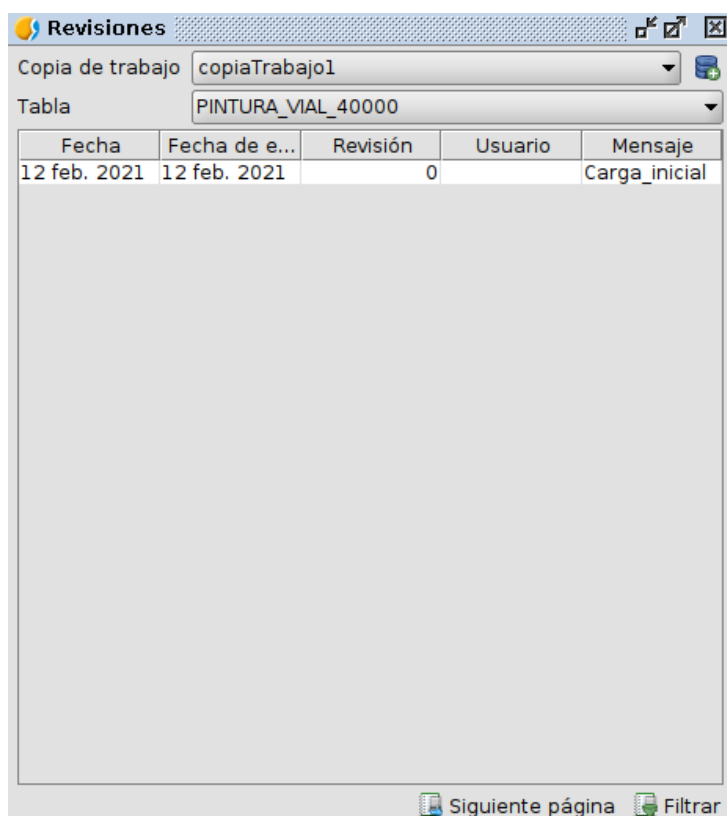


## Gestión de las Revisiones

Tal y como se menciona al inicio del documento, las revisiones o cambios que sufren los datos en un software de control de versiones presentan tanta o más importancia que los propios datos. Haciendo hincapié en la idea anterior, la herramienta software VCSGIS para el control de versiones dispone de un módulo para la correcta gestión de dicha información. Para la gestión de las revisiones ir a la opción *Herramientas* del menú de gvSIG Desktop, pestaña *VCSGgis* y ejecutar *Mostrar revisiones*.

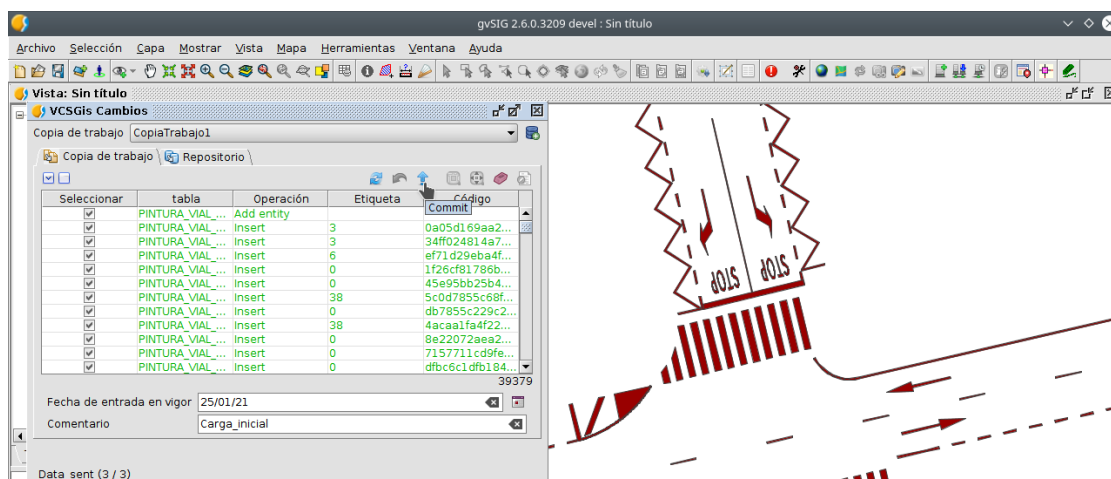


Como resultado de realizar la anterior acción se muestra la siguiente ilustración.



La ventana presenta dos elementos donde hay que seleccionar la *Copia de Trabajo* y la *Tabla/Capa* en cuestión. Una vez seleccionadas en la zona central de la ventana se muestran las revisiones que presenta el elemento seleccionado anteriormente en forma de tabla de datos. Para la gestión de las revisiones en dicha ventana se dispone en su zona inferior de dos iconos que permiten aumentar las revisiones presentes en la zona de visualización de esta o filtrarlas para realizar una búsqueda sobre estas.

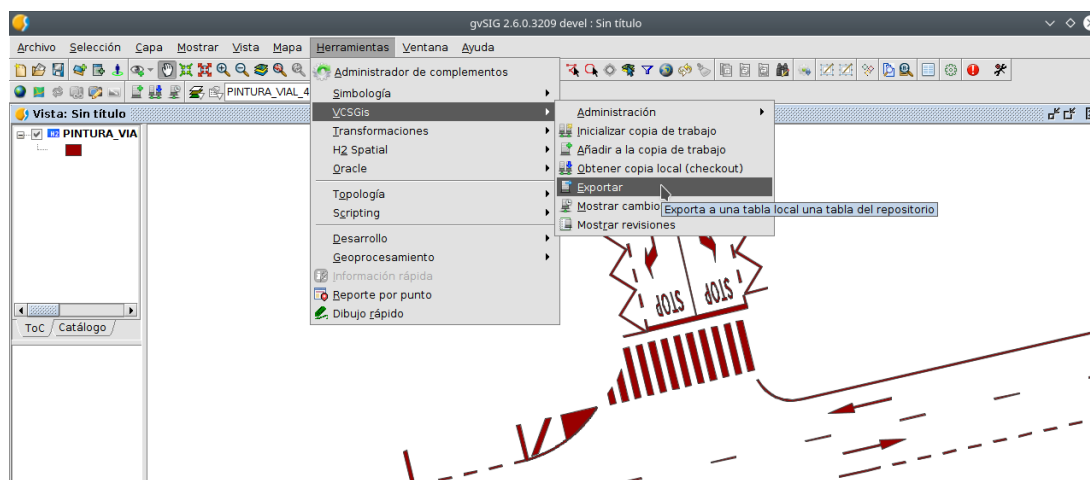
A continuación se muestra en la siguiente composición un ejemplo de como se crea una revisión y como aparece esta en la ventana anteriormente definida.



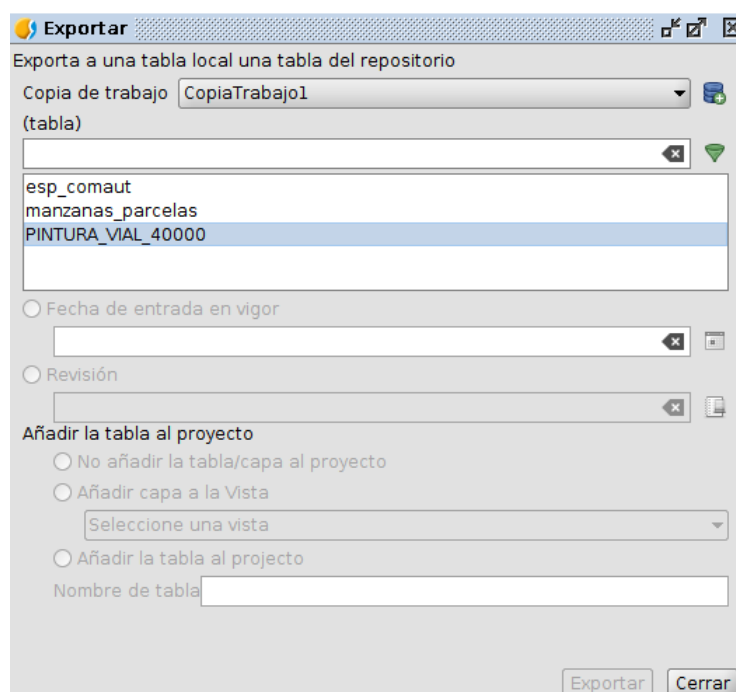
## Exportar datos desde el repositorio

La herramienta de control de versiones VCSGIS dispone de la posibilidad de exportar una capa/tabla de la copia de trabajo como un archivo. Este nuevo fichero posee la característica de ser un conjunto de datos normal, es decir, sin control alguno de sus cambios y por tanto de sus revisiones.

Para realizar el proceso anterior se tiene que ejecutar el comando *Exportar* situado en la opción *Herramientas* del menú de gvSIG Desktop, pestaña *VCSGIS*.



Tras la ejecución de comando se presenta la siguiente ventana.



El funcionamiento de esta es simple, en primer lugar hay que seleccionar la *Copia de Trabajo* de la cual obtener la capa que se busca exportar. Tras esto aparecerán en la ventana de capas la lista de capas de la copia en cuestión, debiendo seleccionar la deseada.

Concretado lo anterior solo queda indicar a que fecha se quiere la capa exportada. Para ello disponemos de dos opciones:

- Fecha de entrada en vigor. Esta opción proporciona un imagen de la cartografía a una determinada fecha indicada por el usuario.
- Revisión. Con esta opción se descarga la capa/tabla con los cambios presentes hasta la citada revisión. La gestión para la selección de revisión se realiza de igual manera que en apartado anterior.

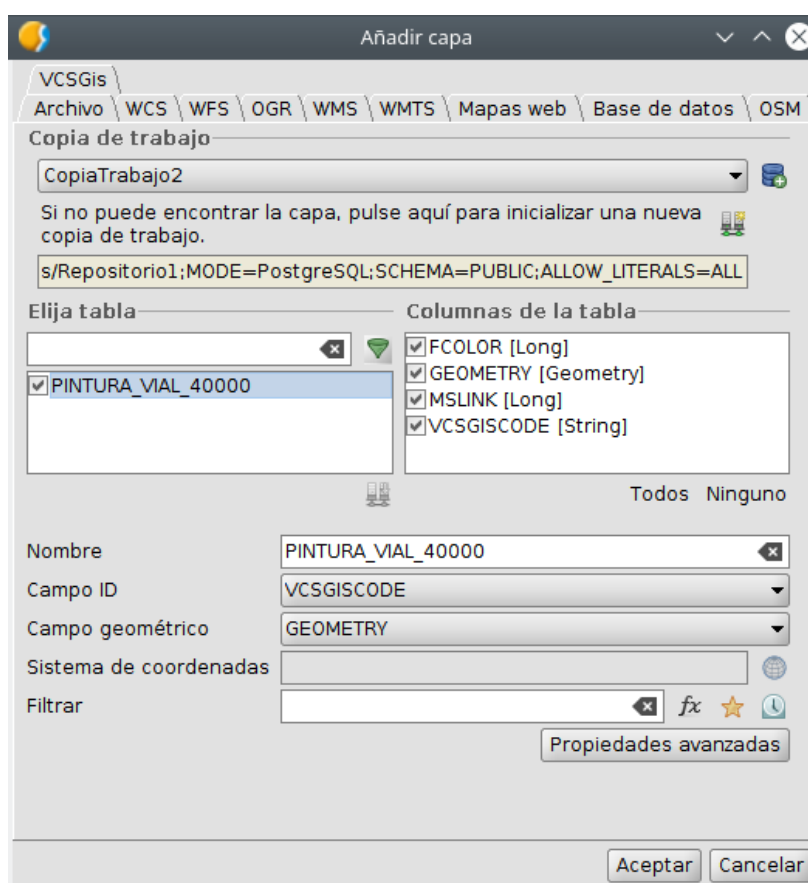
Tras selección de la capa y la “fecha” de la información que queremos exportar de esta solo queda en la ventana una serie de parámetros que permiten la visualización y gestión de esta en las vistas.

### 3 AÑADIR CAPA VCSGIS USANDO EL DIÁLOGO “AÑADIR CAPA” DE GVSIG DESKTOP

Una vez en la ventana de añadir capa existen 2 maneras de proceder para cargar un capa/tabla con control de versiones VCSGIS;

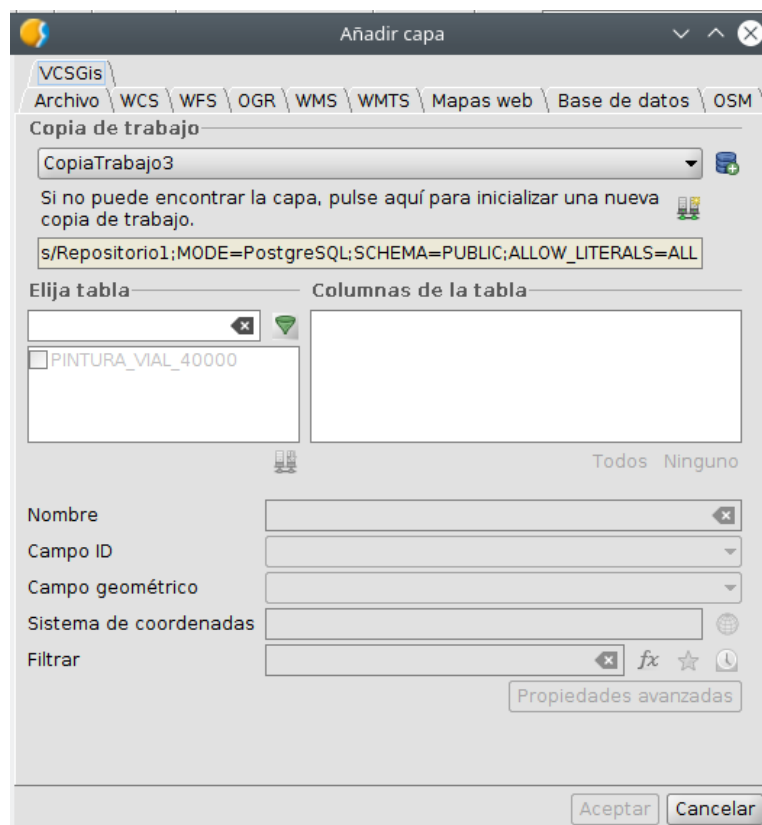
- Seleccionar la capa como si fuera un archivo de base de datos genérico utilizando la conexión a la base de datos asociada a la *Copia de trabajo*.
- Utilizar la pestaña VCSGIS que permite cargar de manera automática este tipo de archivos con control de versiones. Opción más lógica.

A continuación, se muestra el aspecto de dicha ventana al seleccionar la pestaña *VCSGIS* y cargar una capa la cual ya estaba en local pero que fue eliminada de la vista.

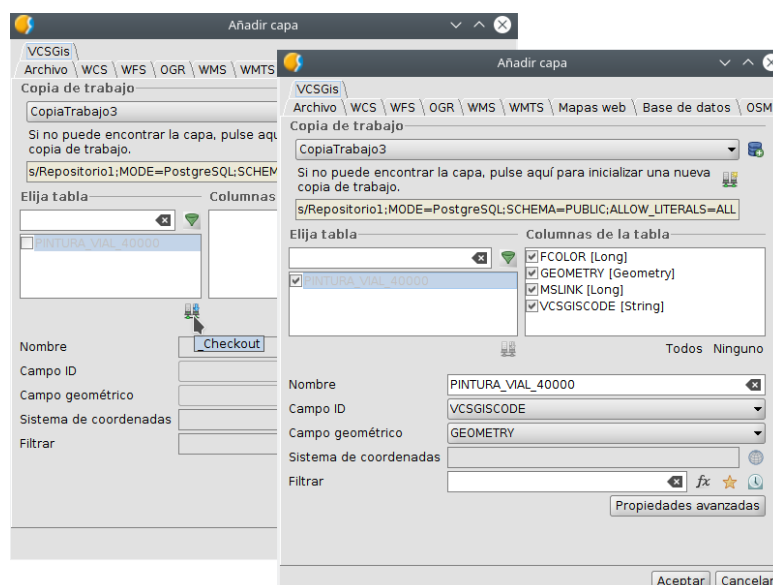


Como se puede apreciar hay que identificar la copia de trabajo y tras eso seleccionar ( marcar el check) en el área de elección de tablas la tabla/capa en cuestión. A parte de lo anterior, es importante cumplimentar la información de campos de información presentes.

En el caso de que la copia de trabajo nunca halla presentado la capa que se pretende cargar, la ventana muestra un comportamiento diferentes. Ver siguiente ilustración.



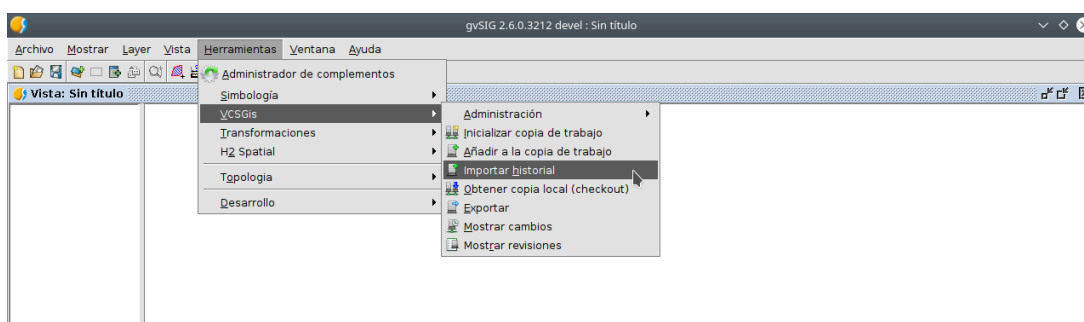
Se puede ver que la capa aparece deshabilitada, para poder cargarla hay que seleccionarla, lo cual habilita el icono que obtiene la copia local (checkout). Tras ejecutar dicho proceso al pulsar dicho icono la ventana presenta el mismo aspecto que si la copia local hubiera tenido en algún momento dicha tabla/capa. Solo queda pulsar *Aceptar* para terminar el proceso.



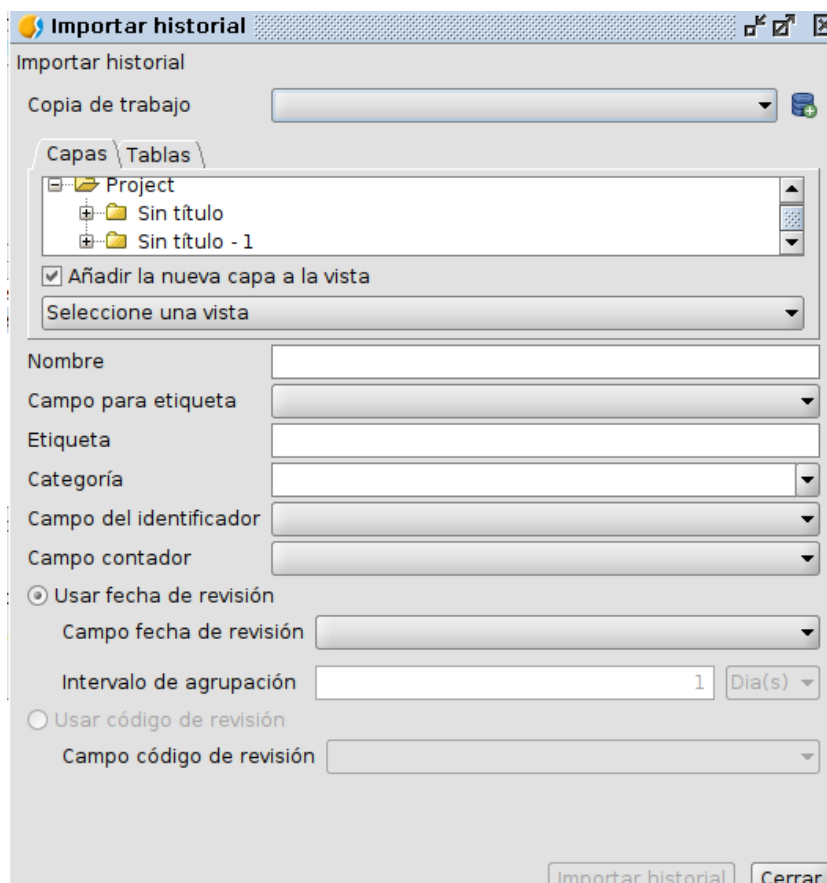
## 4 IMPORTACIÓN O CARGA DE HISTORIAL DE DATOS

Otra funcionalidad presente en el control de versiones *VCSGIS* de *gvSIG Desktop* es la de realizar el proceso de carga o importación de datos históricos siempre y cuando estos presenten un campo de carácter temporal que indique cualquier modificación de estado del elemento. Un ejemplo de esos estados sería su fecha de implantación, modificación, actualización, eliminación...

La opción *Importar historial* se encuentra en la opción *Herramientas* del menú de *gvSIG Desktop*, pestaña *VCSGIS*.

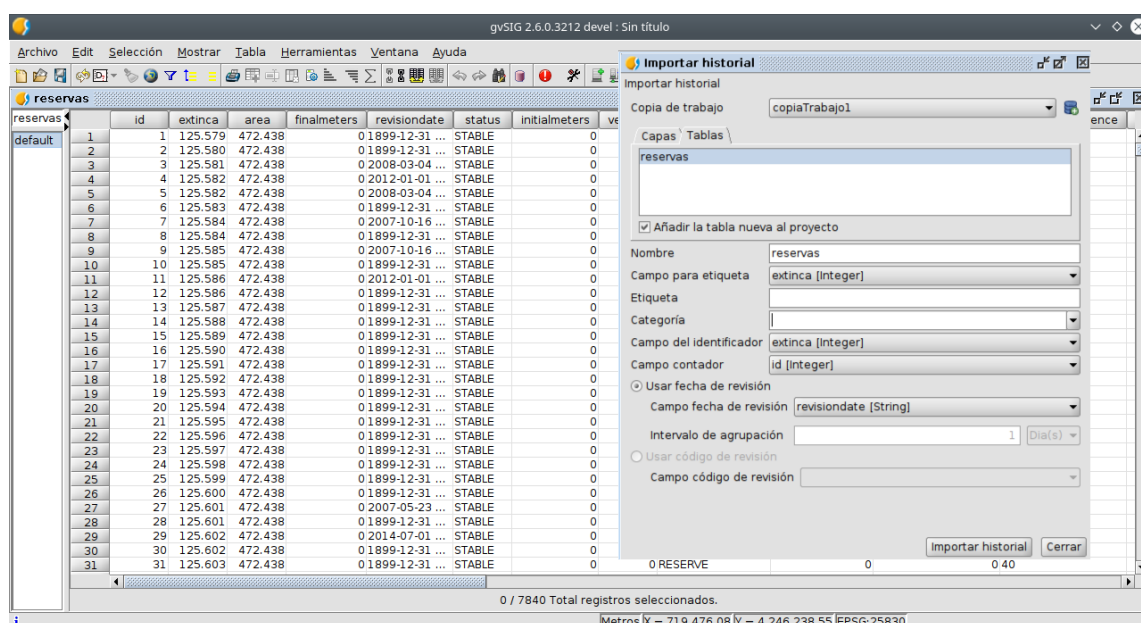


La ventana o interfaz resultante de ejecutar dicho comando se muestra en la siguiente ilustración.



En dicha ventana hay que especificar la copia de trabajo donde se almacenará el histórico en cuestión mediante un desplegable. Una vez seleccionada la copia de trabajo en la ventana hay que identificar el archivo donde se encuentra el historial, el cual debe estar cargado de antemano en *gvSIG Desktop* ya sea como tabla o capa. Tras lo anterior, la ventana dispone de una serie de campos y desplegables a completar, de los cuales hay que destacar el campo identificador el cual hace referencia a la clave primaria de la tabla y los campos que referentes a la inclusión de los datos según fecha de revisión o código de revisión. Referente a estos campos últimos, existe la posibilidad de realizar las diferentes revisiones según fecha de revisión especificando un intervalo de agrupación o usando un código de revisión, previa identificación de este campo.

Como ejemplo para ilustrar el proceso anterior se muestra como la tabla *Reservas* de una base de datos ajena a *VCSGIS* pasa a formar parte del control de versiones tras ejecutar el comando *Importar historial*.



Tras finalizar el proceso se puede comprobar que el proceso de carga fu correcto visualizando las diferentes revisiones creadas ejecutando el comando *Mostrar revisiones* dentro de la opción *Herramientas* del menú de *gvSIG Desktop*, pestaña *VCSGIS*.

**Revisiones**

Copia de trabajo: copiaTrabajo1

Tabla: reservas

Fecha	Fecha de e...	Revisión	Usuario	Mensaje
31 dic. 1899	31 dic. 1899	2		
1 ene. 1990	1 ene. 1990	3		
17 sept. 20...	17 sept. 20...	4		
22 oct. 2002	22 oct. 2002	5		
24 oct. 2002	24 oct. 2002	6		
25 oct. 2002	25 oct. 2002	7		
28 oct. 2002	28 oct. 2002	8		
29 oct. 2002	29 oct. 2002	9		
30 oct. 2002	30 oct. 2002	10		
31 oct. 2002	31 oct. 2002	11		
5 nov. 2002	5 nov. 2002	12		
3 mar. 2003	3 mar. 2003	13		
11 mar. 2003	11 mar. 2003	14		
12 mar. 2003	12 mar. 2003	15		
15 mar. 2003	15 mar. 2003	16		
19 mar. 2003	19 mar. 2003	17		
21 mar. 2003	21 mar. 2003	18		
26 mar. 2003	26 mar. 2003	19		
29 abr. 2003	29 abr. 2003	20		
7 may. 2003	7 may. 2003	21		
14 may. 20...	14 may. 20...	22		
16 may. 20...	16 may. 20...	23		
19 may. 20...	19 may. 20...	24		
20 may. 20...	20 may. 20...	25		
22 may. 20...	22 may. 20...	26		

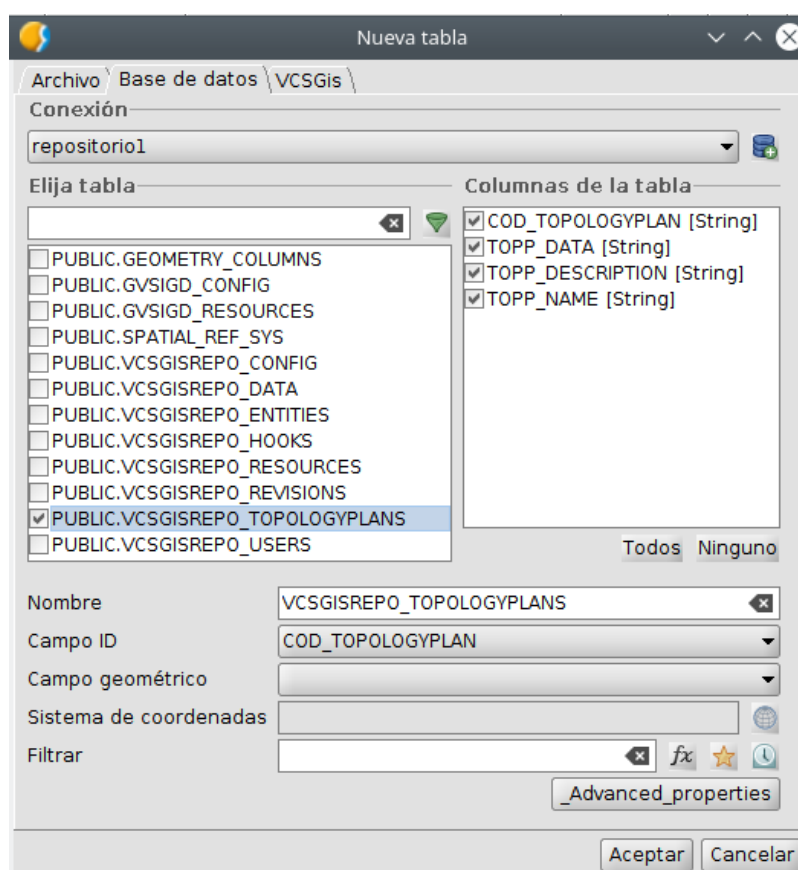
Siguiente página   Filtrar



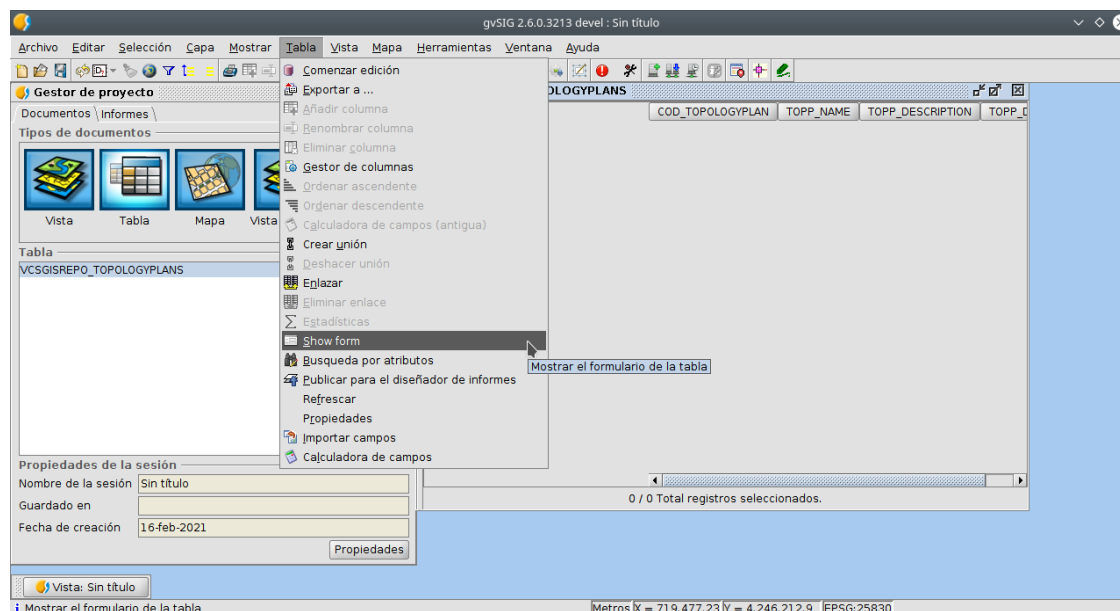
## 5 TOPOLOGÍA EN VCSGIS

El sistema de control de versiones *VCSGIS* de *gvSIG Desktop* permite incluir revisiones topológicas a aplicar a la información que este maneja si el usuario lo desea. En otras palabras, garantiza que la información gráfica gestionada por el control de versiones cumpla las reglas topológicas especificadas por el usuario. Para realizar lo anterior hay que configurar varios aspectos del repositorio como son la tabla de planes topológicos y la tabla de entidades.

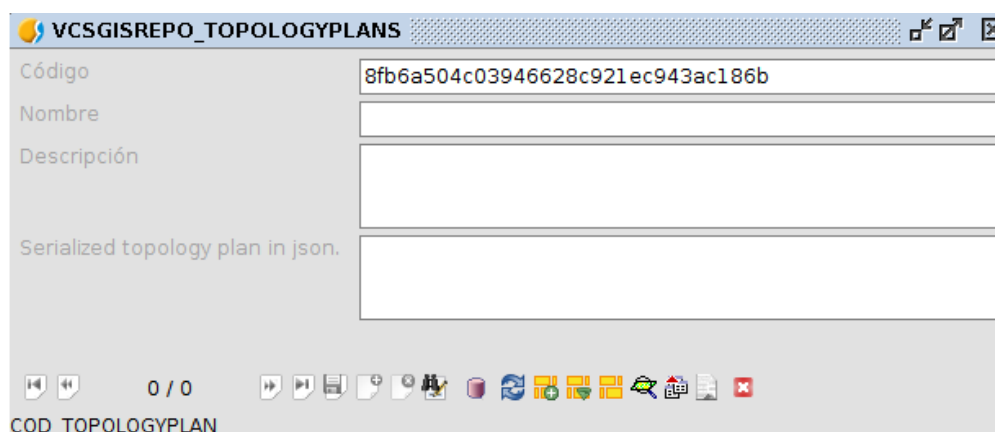
Como se cita anteriormente la configuración topológica se realiza modificando varias tablas. En primer lugar hay que crear en la tabla de planes topológicos, **VCSGISREPO\_TOPOLOGYPLANS**, una nueva entrada. La tabla originalmente aparece vacía y sin ningún elemento. Para acceder a ella hay que abrir la tabla desde el gestor de proyectos.



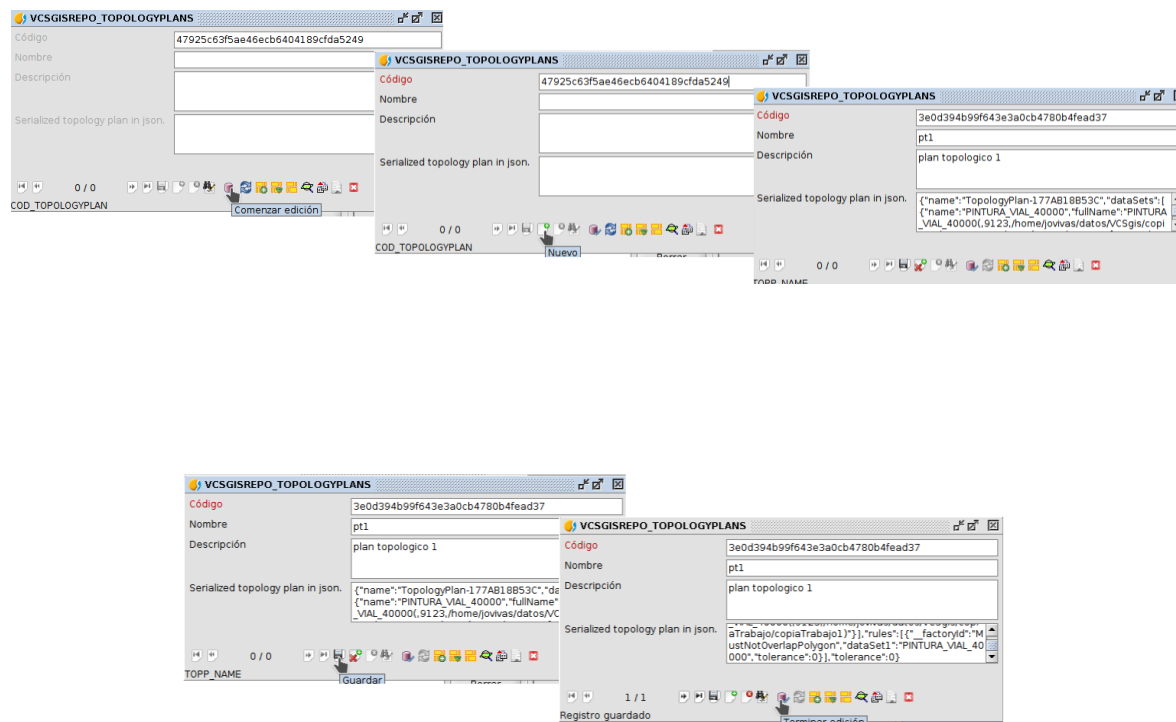
Tras haber accedido a ella hay que ejecutar el comando Show form, el cual nos permite de forma cómoda crear un nuevo plan.



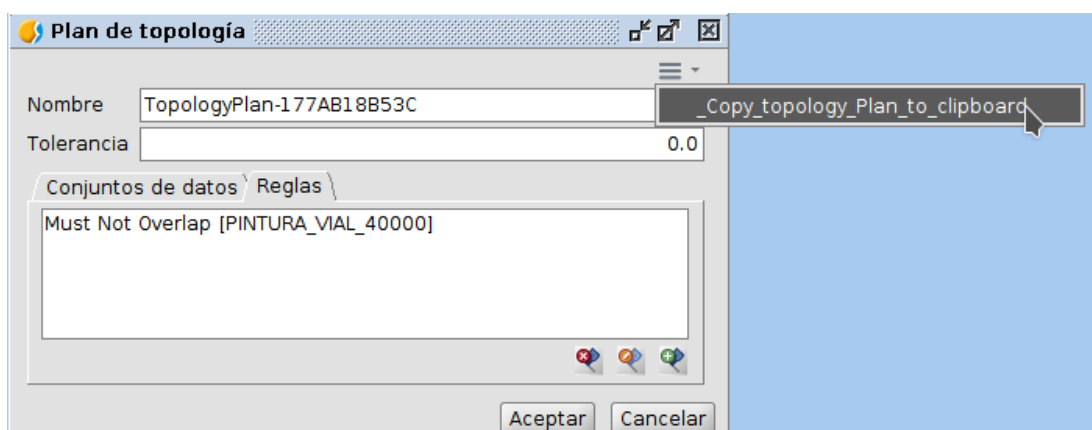
Como resultado de ejecutar el comando anterior se obtiene el siguiente formulario.



Para crear un nuevo plan solo tendremos que poner la tabla en edición, crear un nuevo plan, rellenar la información del plan (Código, Nombre, Descripción y plan topológico en JSON), guardar dicho plan y terminar la edición de la tabla. Dicho proceso se detalla en la siguiente ilustración.



Para obtener la información sobre el plan topológico en formato JSON hay que crear este previamente mediante el plugin de *Topología* de *gvSIG Desktop*. Este se puede obtener en ese formato solo al pulsar el botón de opciones situado en la esquina superior derecha y seleccionar la opción *Copiar plan topológico al portapapeles*.



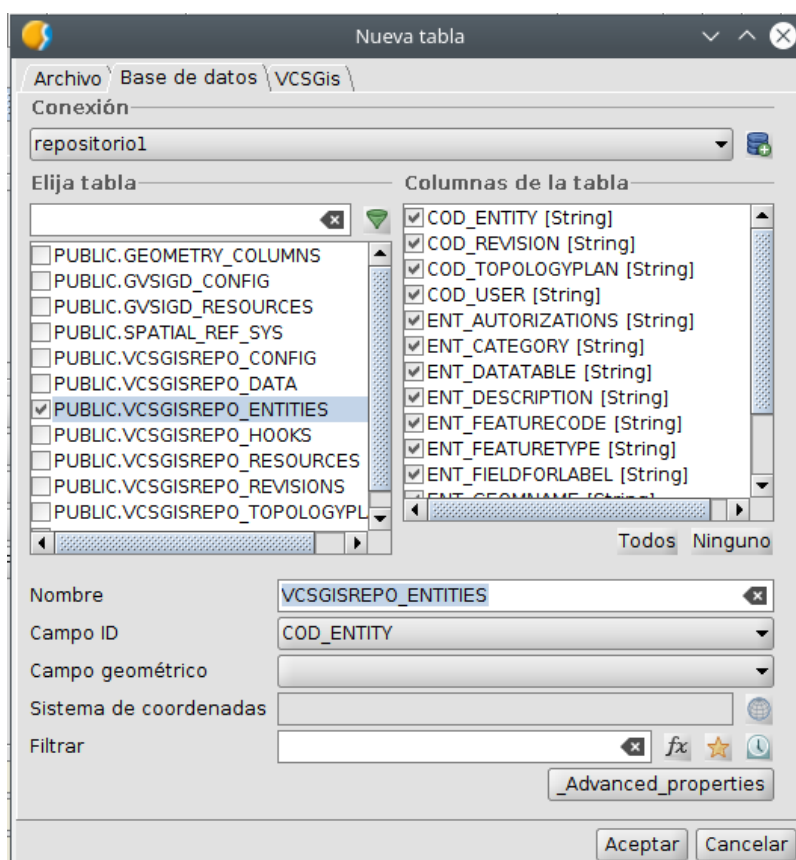
Esa cadena de caracteres es la información a incluir en la definición de un plan topológico de la tabla **VCSGISREPO\_TOPOLOGYPLANS**.

La siguiente ilustración muestra la tabla con el registro correspondiente al plan topológico creado en el ejemplo.

VCSGISREPO_TOPOLOGYPLANS			
COD_TOP...	TOPP_NA...	TOPP_DESCRIPTION	TOPP_DATA
1	3e0d394...	pt1	plan topologico 1
{ "name": "TopologyPlan-177AB18B53C", "dataSets": [ { "name": "P...			

0 / 1 Total registros seleccionados.

Tras la creación de una nueva entrada en la tabla de planes topológicos, solo queda asignar ese plan a las tablas del repositorio sobre las cuales se busca que cumplan topología. Para realizar lo anterior hay que abrir la tabla **VCSGISREPO\_ENTITIES** donde se almacenan las capas/tablas disponibles en el repositorio. El acceso a esta tabla se realiza de igual manera que la anterior.

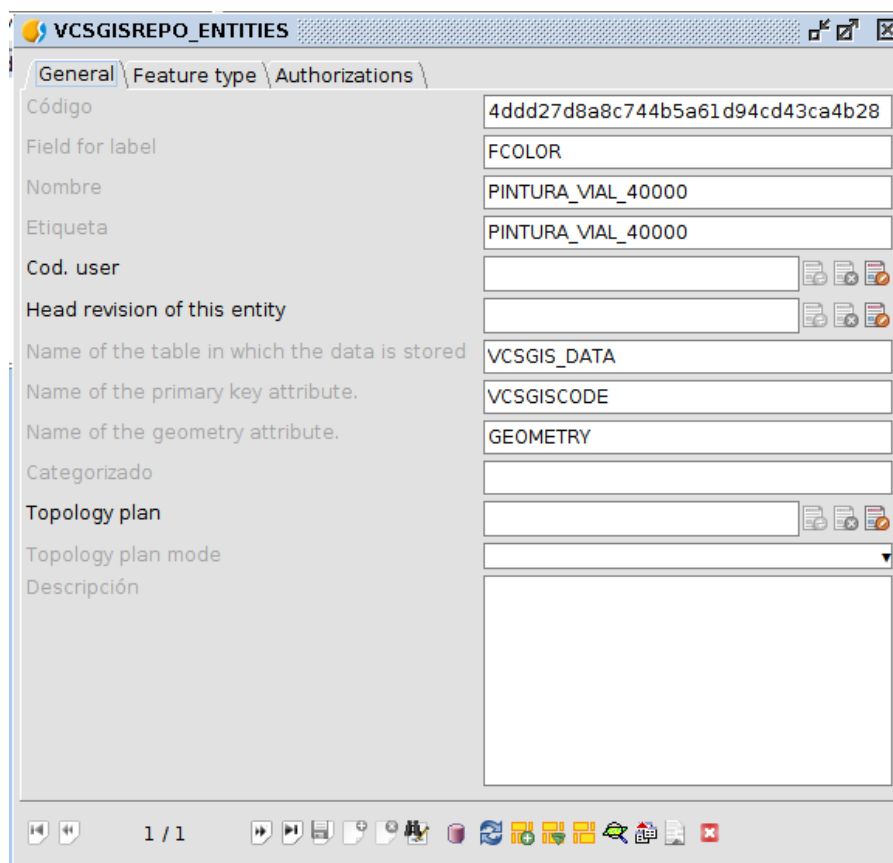


En la siguiente imagen se muestra que la tabla de entidades del repositorio ejemplo solo almacena una capa. Como se puede apreciar no presenta ningún plan de topología asociado.

VCSGISREPO_ENTITIES						
VCSGISREPO_ENTITIES	COD_ENTITY	COD_REVISION	COD_USER	COD_TOPOLOGYPLAN	ENT_NAME	ENT_DATATABLE
default	1	4ddd27d8a...	ab500d240e0...		PINTURA_VI...	VCSGIS_DATA

0 / 1 Total registros seleccionados.

Una vez allí hay que obtener su formulario asociado al igual que con la tabla de planes topológicos mediante ella ejecución del comando *Show form*. Una vez desplegado, en el formulario aparecen todas las tablas disponibles del repositorio, las cuales mediante edición permite asignar un plan de topología definidos anteriormente en la tabla correspondiente.



The screenshot shows the 'VCSGISREPO\_ENTITIES' form with the 'General' tab selected. The form contains the following fields and values:

Field	Value
Código	4ddd27d8a8c744b5a61d94cd43ca4b28
Field for label	FCOLOR
Nombre	PINTURA_VIAL_40000
Etiqueta	PINTURA_VIAL_40000
Cod. user	
Head revision of this entity	
Name of the table in which the data is stored	VCSGIS_DATA
Name of the primary key attribute.	VCSGISCODE
Name of the geometry attribute.	GEOMETRY
Categorizado	
Topology plan	
Topology plan mode	
Descripción	

The form also includes a status bar at the bottom showing '1 / 1' and various navigation icons.

La edición del formulario se realiza de la misma manera que en el formulario de planes topológicos. Durante la edición de la entidad hay que asignar el plan correspondiente a la tabla en la pestaña *General*, desplegable *Plan Topológico*.

## 6 ASIGNACIÓN DE RECURSOS A UNA ENTIDAD

### Introducción

En *gvSIG desktop* podemos asociar *recursos* a una capa o tabla. Estos recursos, cuando estamos trabajando con tablas de tipo fichero, como puede ser un shape, se almacenan junto a este. Así por ejemplo, podemos asociar a un shape, un fichero “.cpg”, con la información sobre la codificación a usar para los textos, un fichero “.gvsleg”, para que tenga una leyenda por defecto cuando lo carguemos, un fichero “.gvslab o “.sld” para que tenga asociado un etiquetado por defecto, “.frm” en el que especifiquemos que aspecto debe tener el formulario asociado a la tabla, incluso podríamos asociar informes personalizados, “.report”. Todos estos tipos de ficheros con información complementaria a los datos de la tabla se consideran recursos asociados a ella. Siendo los recursos citados solo una muestra de los que podríamos asociar a una tabla.

Ahora bien, cuando estamos trabajando con VCSGIS, las tablas se encuentran almacenadas en un repositorio remoto, y se descargan a una base de datos local para trabajar con ellas, entonces... ¿podemos asociar recursos a estas tablas que están en un base de datos bajo un sistema de control de versiones ?

VCSGIS provee mecanismos para poder asociar esos recursos, y almacenarlos en el propio control de versiones de forma que estos se descarguen automáticamente a la copia local cuando se descargan las tablas. Para hacer esto, es preciso la intervención del administrador del repositorio de versiones, y una vez configurado como gestionar los recursos de cada tabla, los usuarios ya podrán acceder a ellos de forma transparente. Con esto se garantiza que todos los usuarios ya no solo dispongan de los datos de una misma tabla, sino también del modo de representar esta así como de los ficheros de recursos que estén asociados a ella.

### Tabla de recursos

En VCSGIS, La tabla de recursos es una tabla formada por dos campos, *nombre* y *value*. El campo *nombre* de tipo String, identifica de manera única el recurso y la capa/tabla a la cual esta asociada, mientras que el campo *value* de tipo byteArray almacena el recurso en sí, es decir el archivo. Para realizar la asignación de este recurso, previa a su creación y actualización con los diferentes recursos, hay que realizar modificaciones en la entidades del repositorio de la tabla **VCSGISREPO\_ENTITIES** asignándole a cada una de de las tablas o entidades del repositorio en que tabla estan almacenados los recursos de ella.

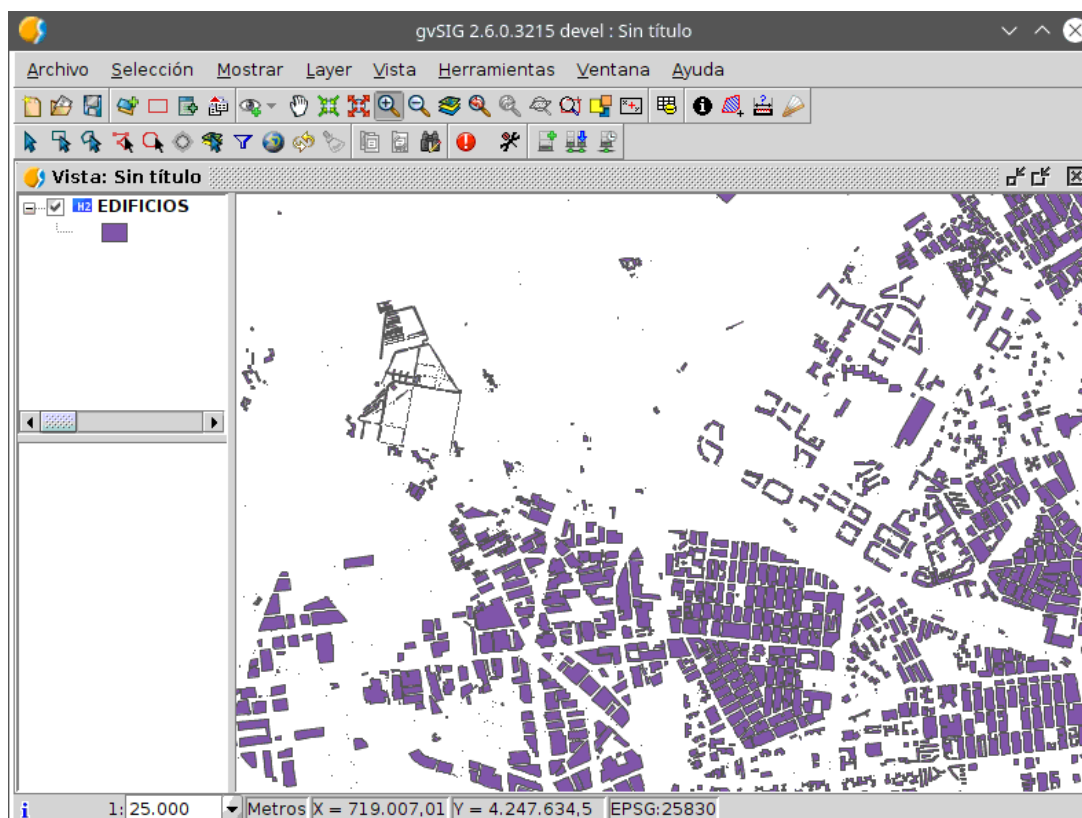
### Ejemplo asignación recursos

Para facilitar la comprensión del proceso de asignación de recursos a una determinada capa a continuación se presenta un ejemplo. En este se va a asociar una leyenda a una capa presente en un repositorio.

La capa utilizada para la realización del ejemplo llamada *EDIFICIOS* esta almacenada en una capa de polígonos con la geometría de los edificios de la ciudad de Alicante y que ya esta bajo el control de versiones. Esta capa presenta las siguientes características;

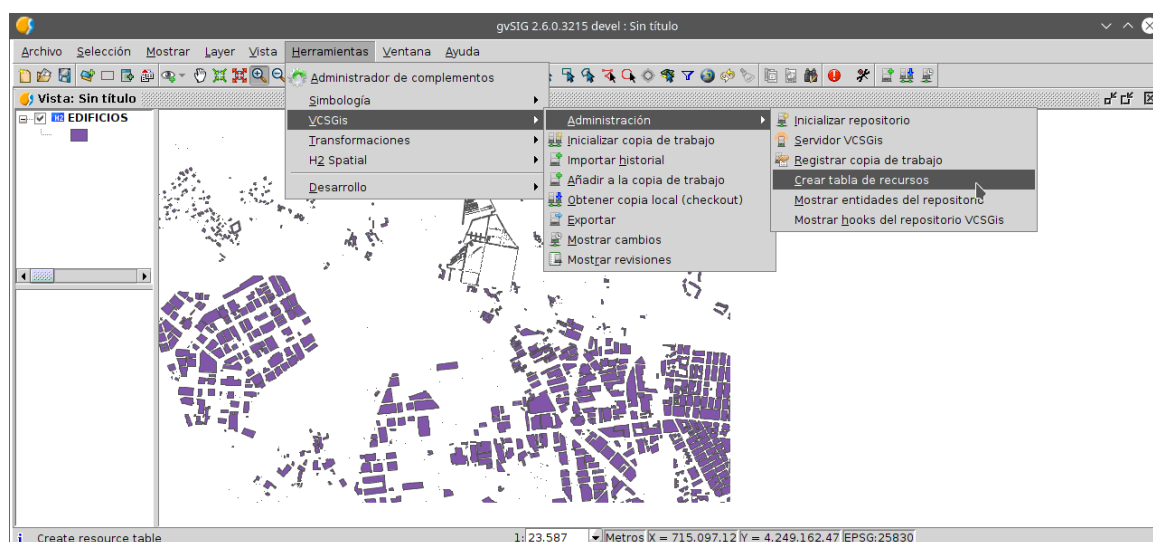
- Repositorio: *aytoALC*
- Copia de trabajo: *usuario1*
- Categoría: *BASE*

La siguiente ilustración muestra la capa cargada en *gvSIG Desktop*.

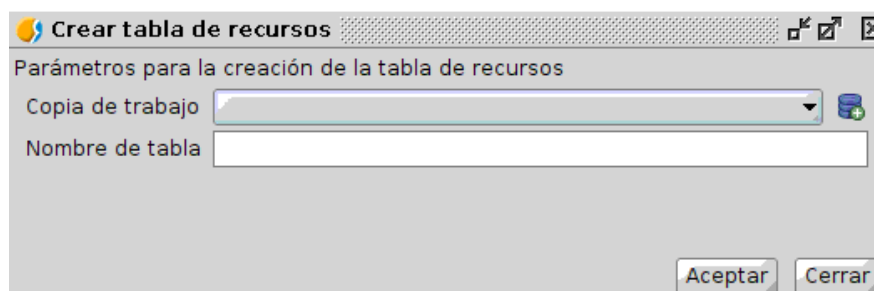


Tal y como se puede ver, la simbología presenta una leyenda por defecto. Para cambiarla y registrarla en el repositorio de modo que siempre que se utilice dicha capa se presente con la leyenda correcta hay que registrar la nueva leyenda como recurso.

En primer lugar hay que crear la tabla de recursos. Dicho proceso se realiza mediante la opción *Crear tabla de recursos* situada en el menú *Herramientas*, submenú *VCSGIS*, submenú *Administración*.



Realizar lo anterior muestra el siguiente cuadro de diálogo.



En la ventana hay que seleccionar en el primer desplegable llamado *Copia de trabajo* la copia de trabajo donde esta la capa, *usuario1*.

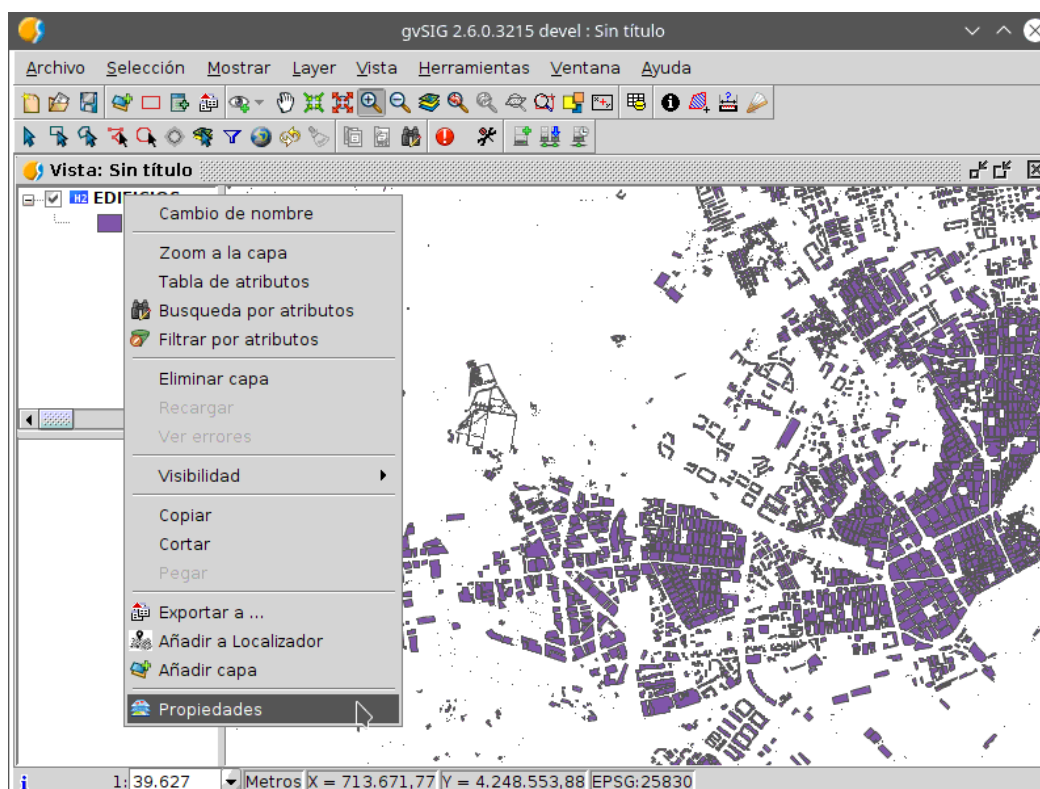
En el segundo componente hay que detallar el nombre de la tabla de recursos. Se nombra la nueva tabla de recursos como *BASE\_RESOURCES* ya que en este ejemplo vamos a almacenar todos los recursos de la categoría *BASE* en dicha tabla.

Hay que tener en cuenta que se pueden crear tantas tablas de recursos como se deseen. De modo que se puede crear una única tabla de recursos que almacene todos ellos o varias de estas que almacenen los recursos de determinadas categorías de datos por ejemplo. Normalmente agruparemos las tablas del repositorio de forma que grupos lógicos de tablas usen la misma tabla de recursos, así balancearemos la carga de esta tabla y los usuarios que no trabajen con las capas de un bloque lógico no precisaran descargarse los recursos de esta.

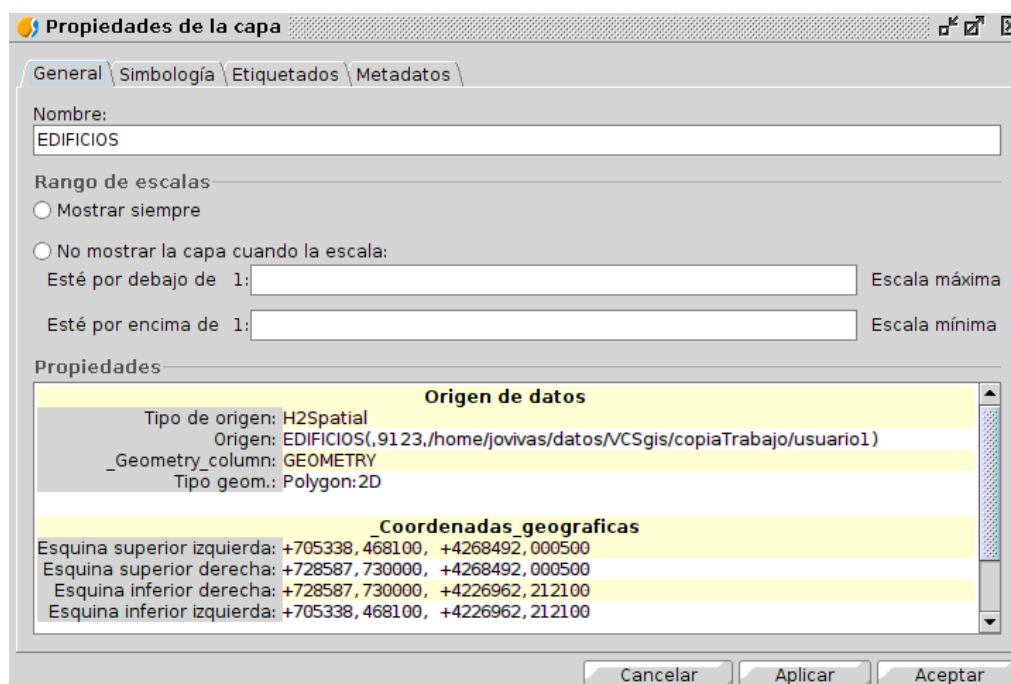
El proceso de creación finaliza pulsando el botón *Aceptar* en la esquina inferior derecha de la ventana.

El siguiente paso a seguir es almacenar un recurso en la tabla recientemente creada. Para ello es necesario disponer de él o crearlo desde cero. En el caso del ejemplo se procede a crear de cero la leyenda para la capa *EDIFICIOS*.

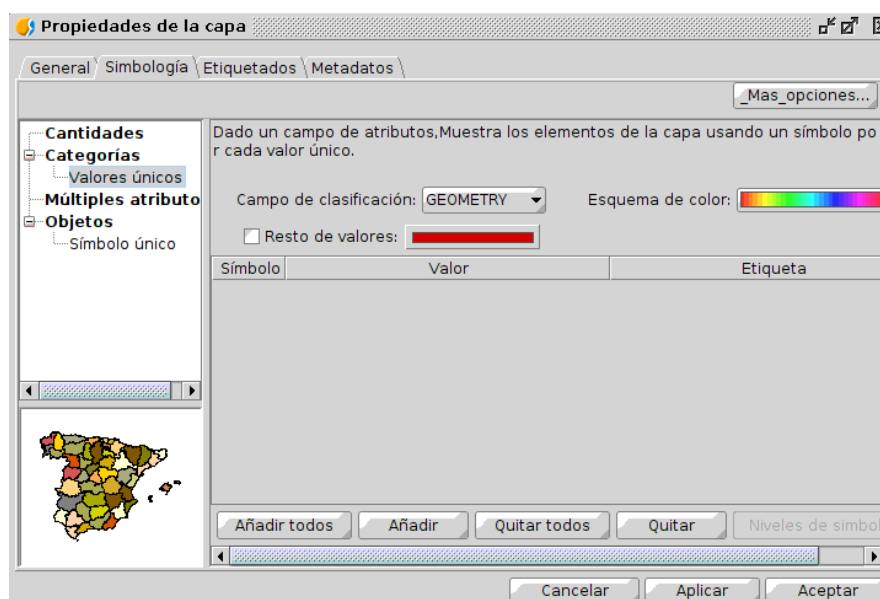
Para crear una leyenda hay que ir al árbol de capas situado en el *Toc* de *gvSIG Desktop* y tras selección de la capa en cuestión pulsar botón derecho del mouse y seleccionar la opción *Propiedades*.



La opción anterior presenta la siguiente ventana.

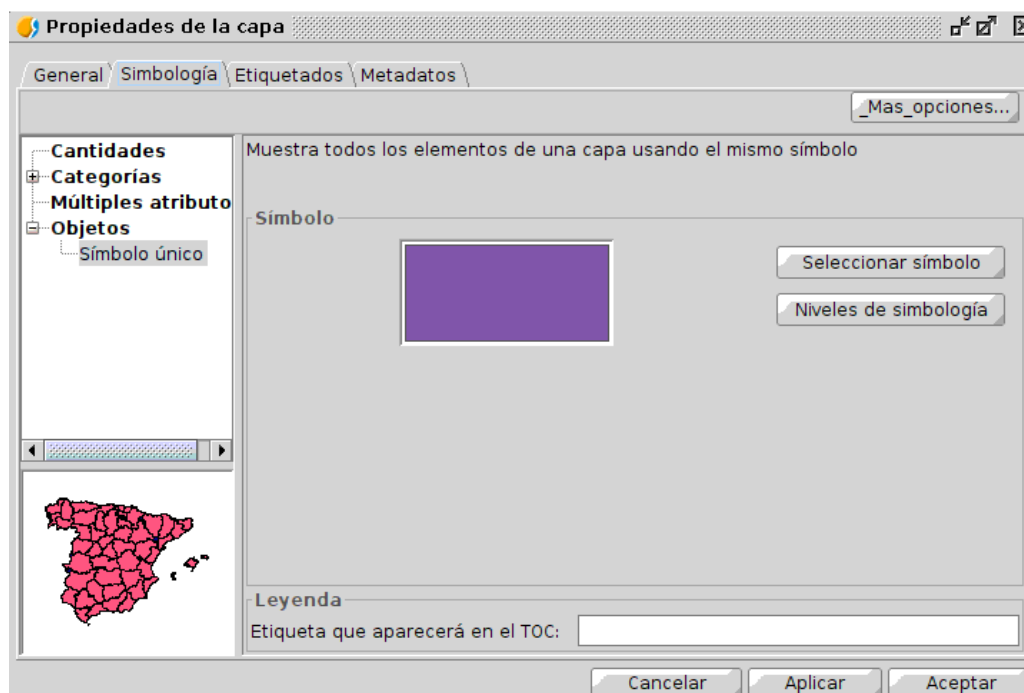


Se selecciona la pestaña *Simbología* pues en esta se especifica todo lo referente a la representación gráfica de la capa. El panel de de la pestaña anterior es el siguiente.

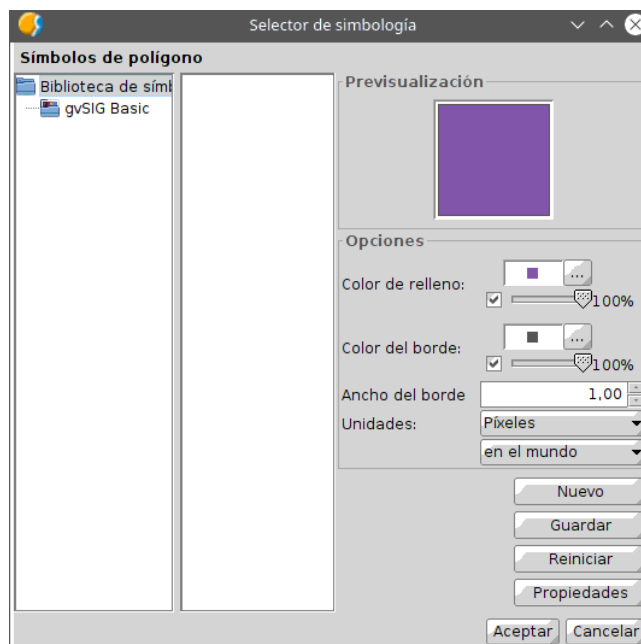


Una vez allí y continuando con el caso del ejemplo, se selecciona la opción *Símbolo único* de dentro de *Objetos* del cuadro de opciones de simbología situado en la parte izquierda del panel.

Dicha selección habilita el siguiente panel en la ventana.



En el panel se inicia el proceso de definición de nuestra leyenda pulsando el botón *Selección símbolo* el cual presenta el cuadro de diálogo *Selector de simbología*.



En esta nueva ventana hay que definir las características gráficas de los elementos de la capa, en este caso polígonos. Por lo tanto hay que definir como se van a representar sus bordes e interior. Los parámetros para los bordes e interior de los polígonos de la capa edificios se detallan a continuación.

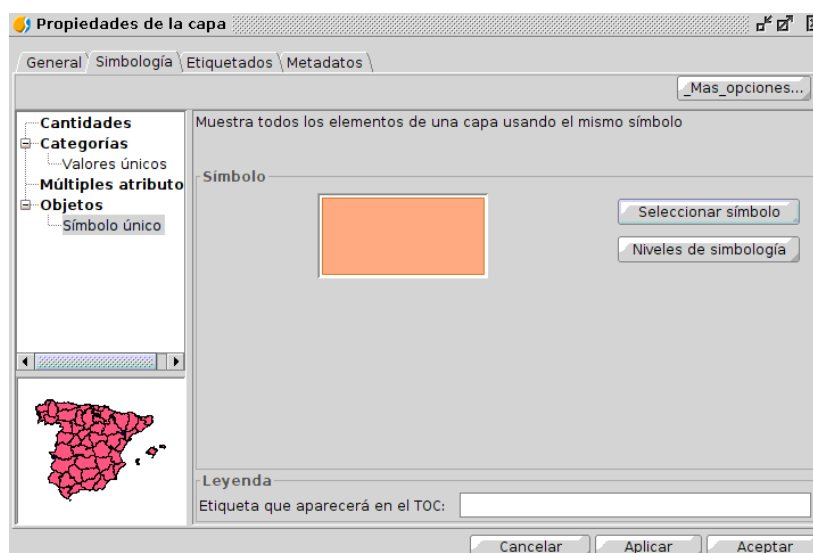
### Componente Color(RGB) Opacidad Ancho (píxel)

Borde 231 120 58 100% 1

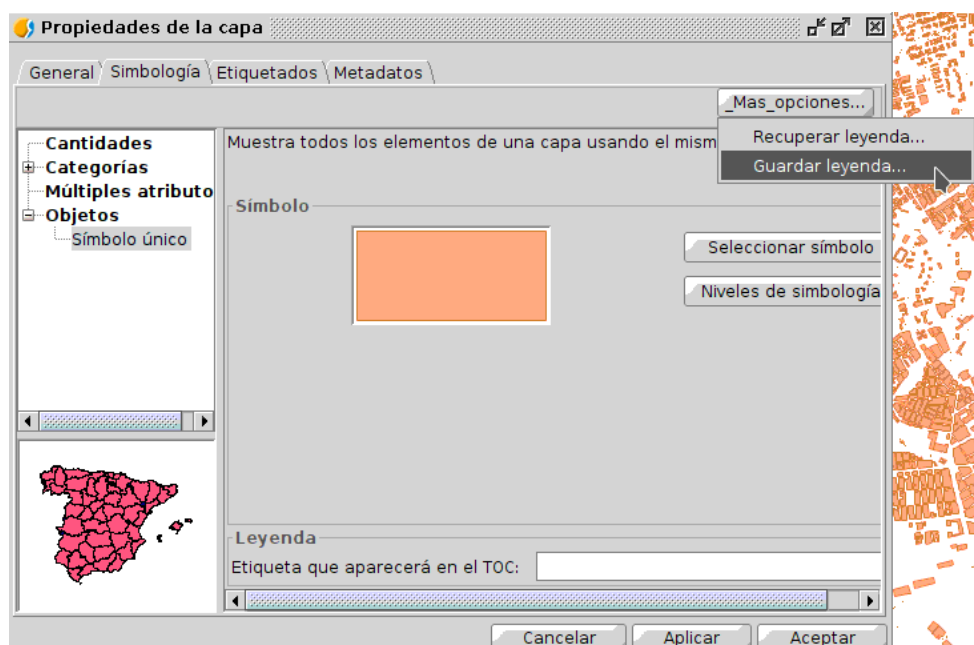
Interior 248 190 132 100% -

Una vez especificados se aceptan los cambios pulsando el botón *Aceptar* y se cierra el cuadro de diálogo

Tras lo anterior ya de nuevo en el panel de la pestaña *Simbología* de la ventana *Propiedades* de la capa *EDIFICIOS* se puede ver que el símbolo único ha cambiado según la definición especificada anteriormente.



Solo queda ahora almacenar esa simbología pulsando el botón *Más opciones* situado en la zona superior derecha de la ventana y seleccionar a continuación la opción *Guardar leyenda*.

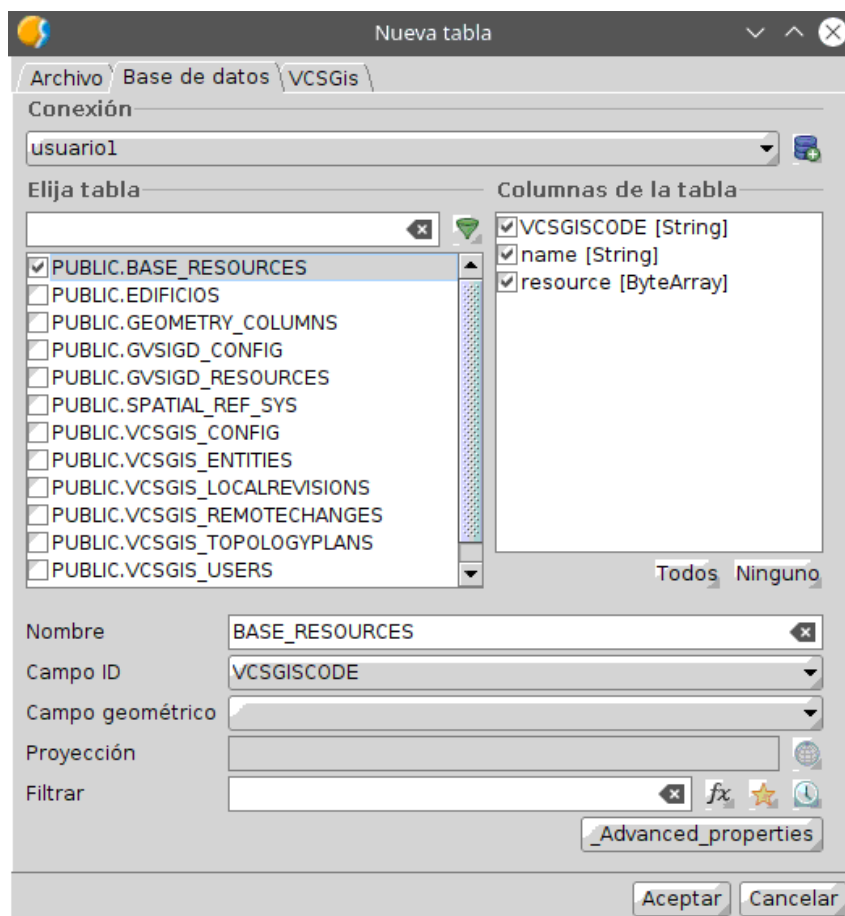


La opción anterior presenta un cuadro de diálogo que nos permite indicar la ruta donde almacenar la leyenda y el nombre del fichero que almacenará esta. En el caso del ejemplo se almacena en una carpeta destinada a almacenar leyendas y el nombre del archivo es el mismo que la capa con la extensión `.gvsleg`, es decir `EDIFICIOS.gvsleg`. Hay que indicar que el archivo con el recurso siempre tiene que llamarse de la misma manera que la capa de la que es recurso.

Una vez guardada la leyenda, pulsamos los botones *Aplicar* y *Aceptar* para terminar el proceso de definición de la simbología de la capa.

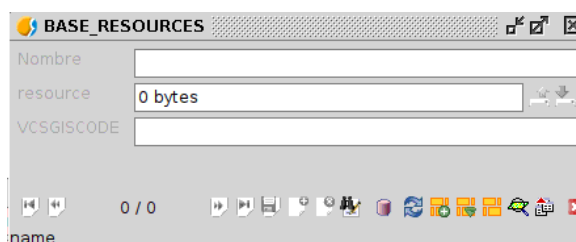
Con el recurso ya creado hay que introducir este en la tabla de recursos, `BASE_RECURSOS`.

En primer lugar hay que abrir la tabla *BASE\_RESOURCES*. Para abrir la tabla hay que realizarlo desde el *Gestor de proyectos* situado en el menú *Mostrar* de *gvSIG Desktop*. El proceso de abrir una tabla es el genérico a abrir cualquier archivo, primero se selecciona *Tabla* como tipo de datos a abrir, se selecciona la opción de *Nuevo*, lo que habilita una ventana donde se tiene que seleccionar la pestaña *Base de datos*. Esa pestaña muestra en su zona superior un desplegable donde hay que especificar la base de datos donde se encuentra la tabla, *usuario1*. Una vez seleccionada la base de datos, en la lista de tablas de esta hay que marcar la tabla en cuestión y pulsa el botón *Aceptar*.

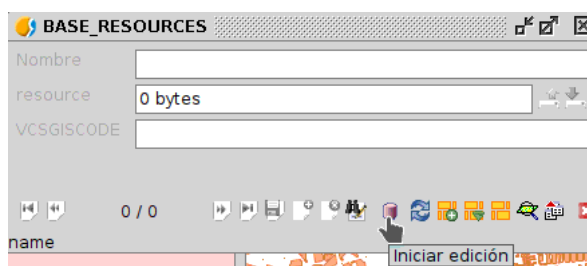


Como resultado se obtiene la tabla vacía. Para rellenarla hay que obtener su formulario, seleccionando la opción *Show form* situada en el menú *Tabla* de *gvSIG Desktop* siempre y cuando la tabla este abierta y seleccionada.

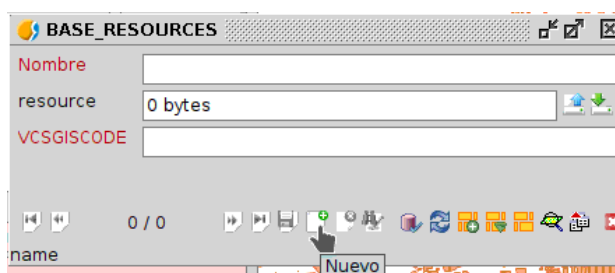
El formulario de la tabla *BASE\_RESOURCES* es el siguiente.



Para introducir un valor en la tabla hay que poner esta en edición ya sea desde el mismo desplegable que se mencionó anteriormente para obtener el formulario, o desde el mismo formulario utilizando el botón *Comenzar edición*.

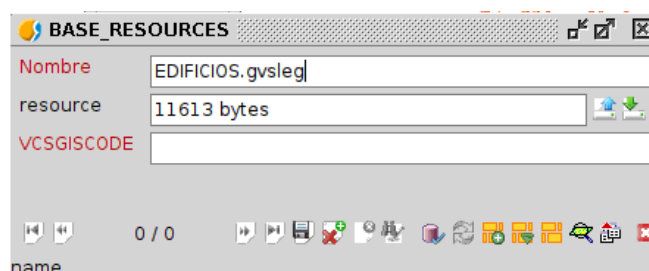


Una vez en edición hay que crear un nuevo elemento mediante el botón *Nuevo* situado en la zona inferior del formulario e indicado en la imagen siguiente.

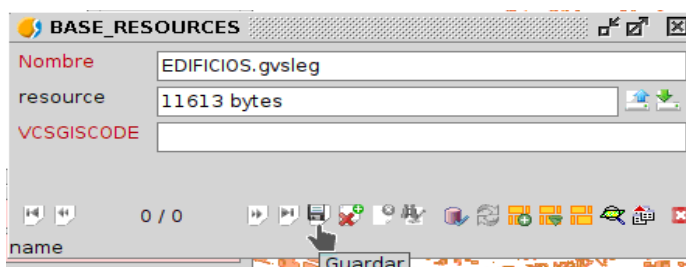


Especificamos en el campo *nombre* el nombre de la tabla más la extensión asociada al recurso que se desea almacenar y en el campo *value* gracias al botón izquierdo se puede subir a la tabla dicho fichero.

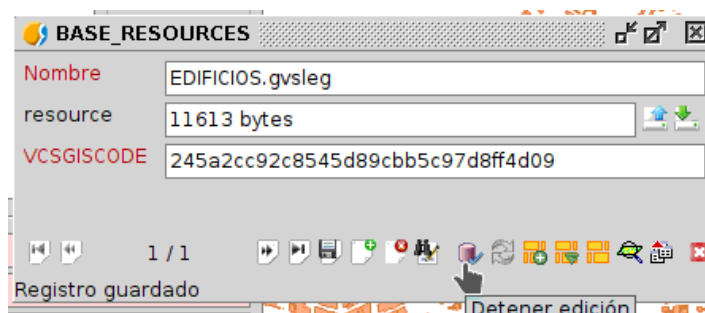
En el caso del ejemplo la configuración del formulario es la siguiente.



Una vez creado el nuevo elemento de la tabla solo queda guardar.

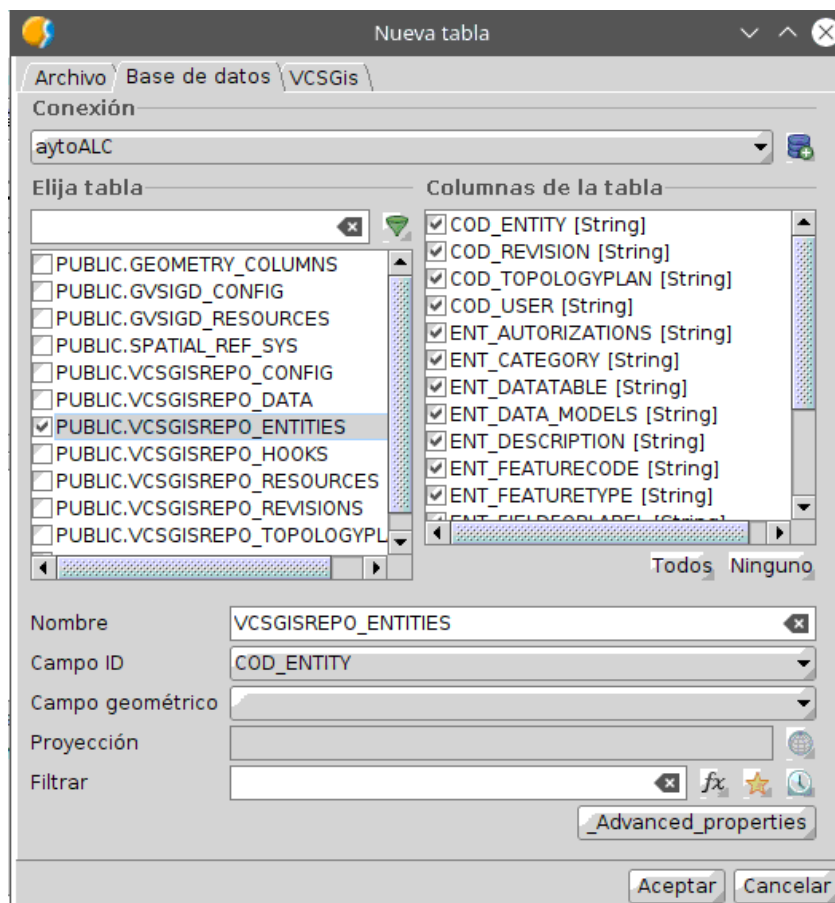


Y posteriormente finalizar la edición de la tabla.



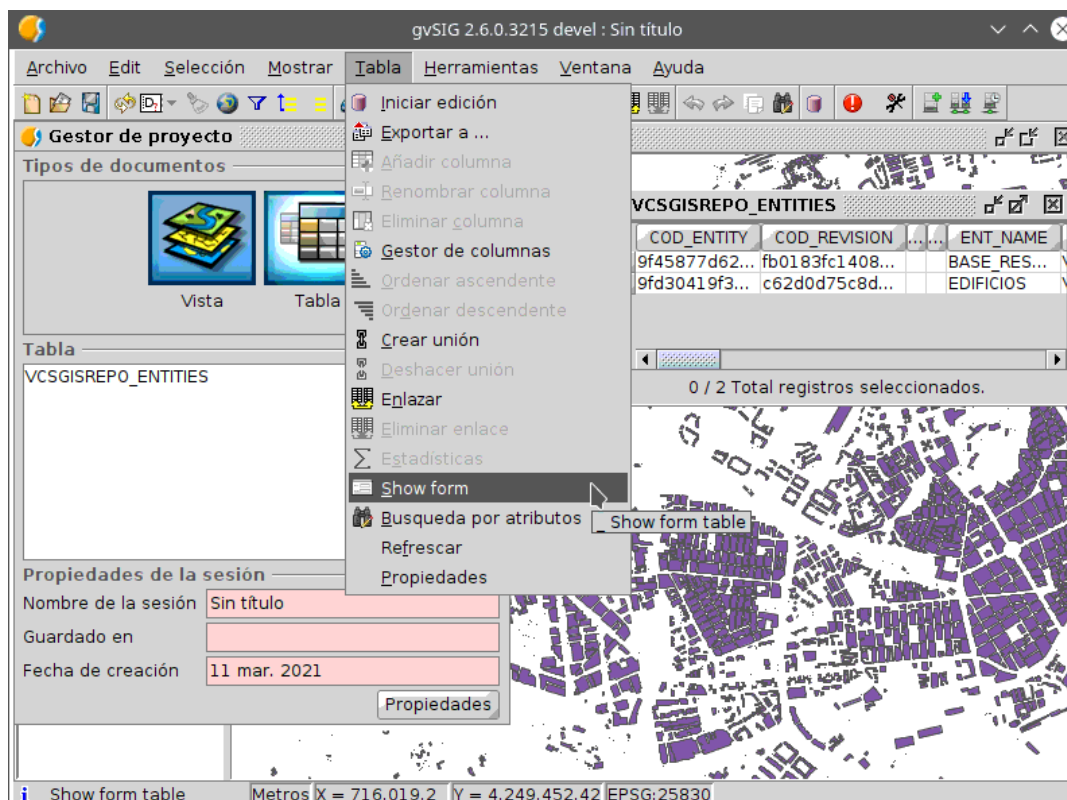
Tras crear y rellenar la tabla hay que subir esta al repositorio. Para hacer lo anterior hay que añadir la tabla a la copia local, opción *Añadir a la copia de trabajo* del menú *Herramientas*, submenú *VCSGIS*, y posteriormente realizar un *commit* tras ejecutar la opción *Mostrar cambios* situada en el menú *Herramientas*, submenú *VCSGIS*.

Tras la correcta definición de la tabla recursos hay que asignar a la capa *EDIFICIOS* del repositorio dicha tabla como su tabla de recursos. Para ello hay que abrir la tabla *VCSGISREPO\_ENTITIES* situada en el repositorio *aytoALC*. Para abrir la tabla hay que realizarlo desde el *Gestor de proyectos* situado en el menú *Mostrar* de *gvSIG Desktop*. El proceso de abrir una tabla es el genérico a abrir cualquier archivo, primero se selecciona *Tabla* como tipo de datos a abrir, se selecciona la opción de *Nuevo*, lo que habilita una ventana donde se tiene que seleccionar la pestaña *Base de datos*. Esa pestaña muestra en su zona superior un desplegable donde hay que especificar la base de datos donde se encuentra la tabla, *aytoALC*. Una vez seleccionada la base de datos, en la lista de tablas de esta hay que marcar la tabla en cuestión y pulsa el botón *Aceptar*.



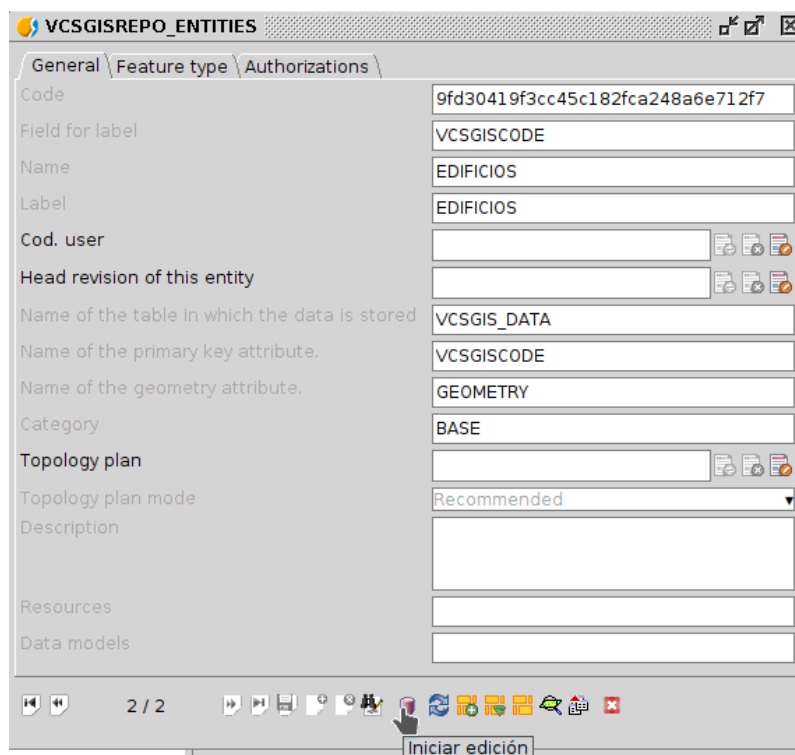
La siguiente imagen muestra la tabla *VCSGISREPO\_ENTITIES* donde por el momento hay dos entidades, la capa/tabla *EDIFICIOS* y la tabla *BASE\_RECURSOS*.

El proceso de asignación de los recursos se realiza modificando la entidad a la que se desea asignar el recurso, en este caso la capa *EDIFICIOS*. Para ello es necesario obtener el formulario asociado a la capa *VCSGISREPO\_ENTITIES*. Para obtener el formulario de la tabla seleccionaremos la opción *Show form* situada en el menú *Tabla* de *gvSIG Desktop* siempre y cuando la tabla este abierta y seleccionada.



El formulario de la tabla es el siguiente.

Una vez en el formulario se identifica el elemento que hace referencia a la tabla/capa *EDIFICIOS* y se inicia la edición de la tabla para la modificación de este. Este proceso se puede realizar desde el mismo desplegable que se mencionó anteriormente para obtener el formulario, o desde el mismo formulario utilizando el botón *Comenzar edición*.



VCSGISREPO\_ENTITIES

General | Feature type | Authorizations

Code: 9fd30419f3cc45c182fca248a6e712f7

Field for label: VCSGISCODE

Name: EDIFICIOS

Label: EDIFICIOS

Cod. user:

Head revision of this entity:

Name of the table in which the data is stored: VCSGIS\_DATA

Name of the primary key attribute: VCSGISCODE

Name of the geometry attribute: GEOMETRY

Category: BASE

Topology plan:

Topology plan mode: Recommended

Description:

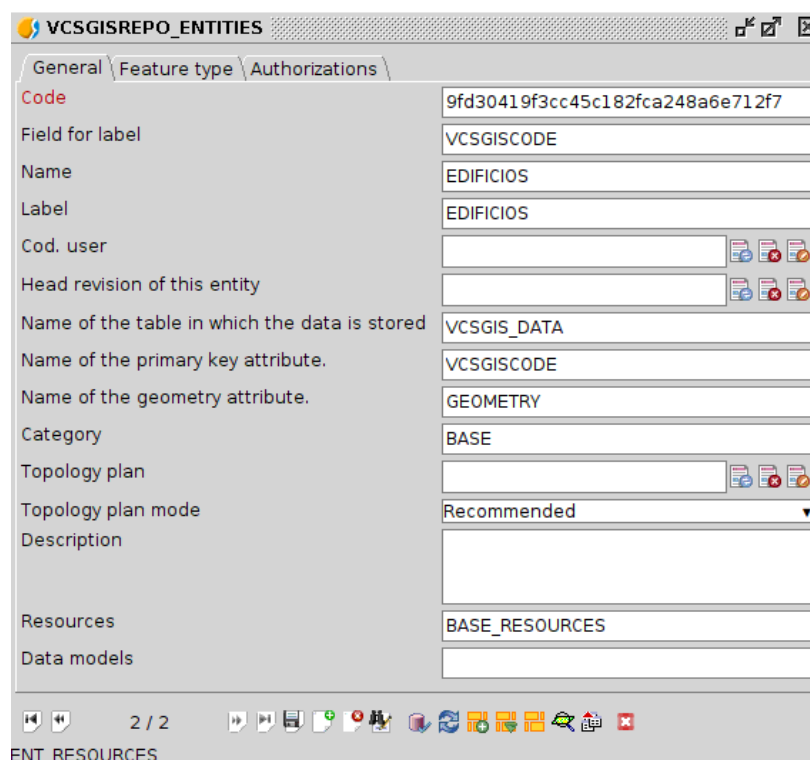
Resources:

Data models:

2 / 2

Iniciar edición

De los diferentes campos del formulario hay que identificar el referente a los recursos, llamado, *Resources*. En este hay que especificar el nombre de nuestra tabla de recursos, *BASE\_RESOURCES*, ya que la tabla/capa edificios pertenece a la categoría *BASE*.



VCSGISREPO\_ENTITIES

General | Feature type | Authorizations

Code: 9fd30419f3cc45c182fca248a6e712f7

Field for label: VCSGISCODE

Name: EDIFICIOS

Label: EDIFICIOS

Cod. user:

Head revision of this entity:

Name of the table in which the data is stored: VCSGIS\_DATA

Name of the primary key attribute: VCSGISCODE

Name of the geometry attribute: GEOMETRY

Category: BASE

Topology plan:

Topology plan mode: Recommended

Description:

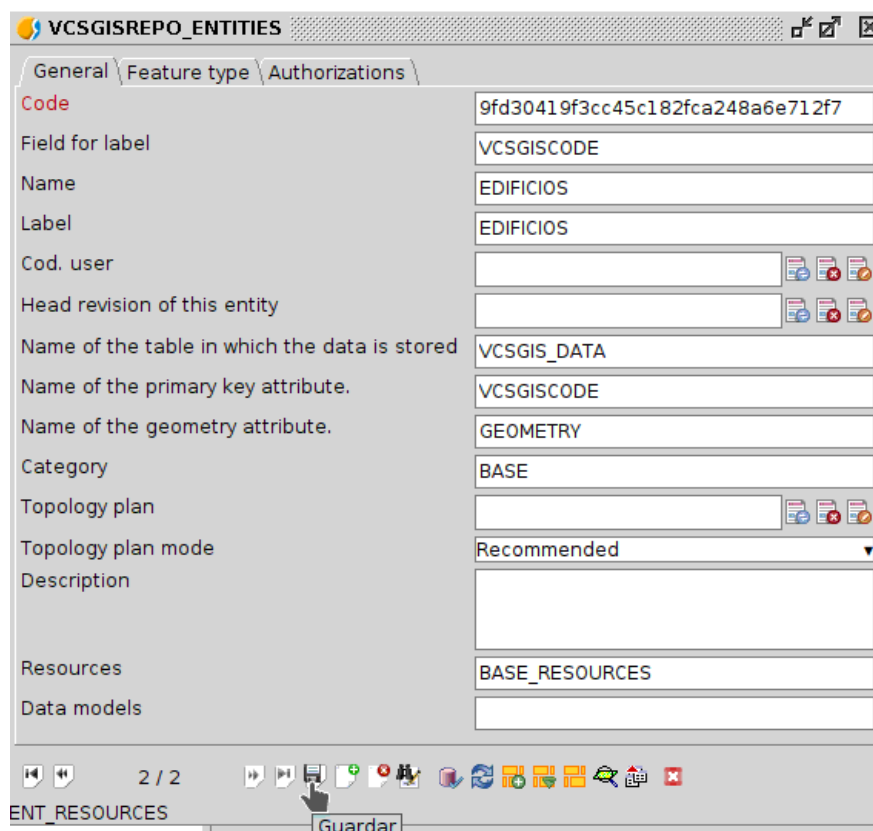
Resources: BASE\_RESOURCES

Data models:

2 / 2

ENT RESOURCES

Solo queda guardar los cambios en la entidad.



**VCSGISREPO\_ENTITIES**

General | Feature type | Authorizations

Code: 9fd30419f3cc45c182fca248a6e712f7

Field for label: VCSGISCODE

Name: EDIFICIOS

Label: EDIFICIOS

Cod. user: [empty]

Head revision of this entity: [empty]

Name of the table in which the data is stored: VCSGIS\_DATA

Name of the primary key attribute: VCSGISCODE

Name of the geometry attribute: GEOMETRY

Category: BASE

Topology plan: [empty]

Topology plan mode: Recommended

Description: [empty]

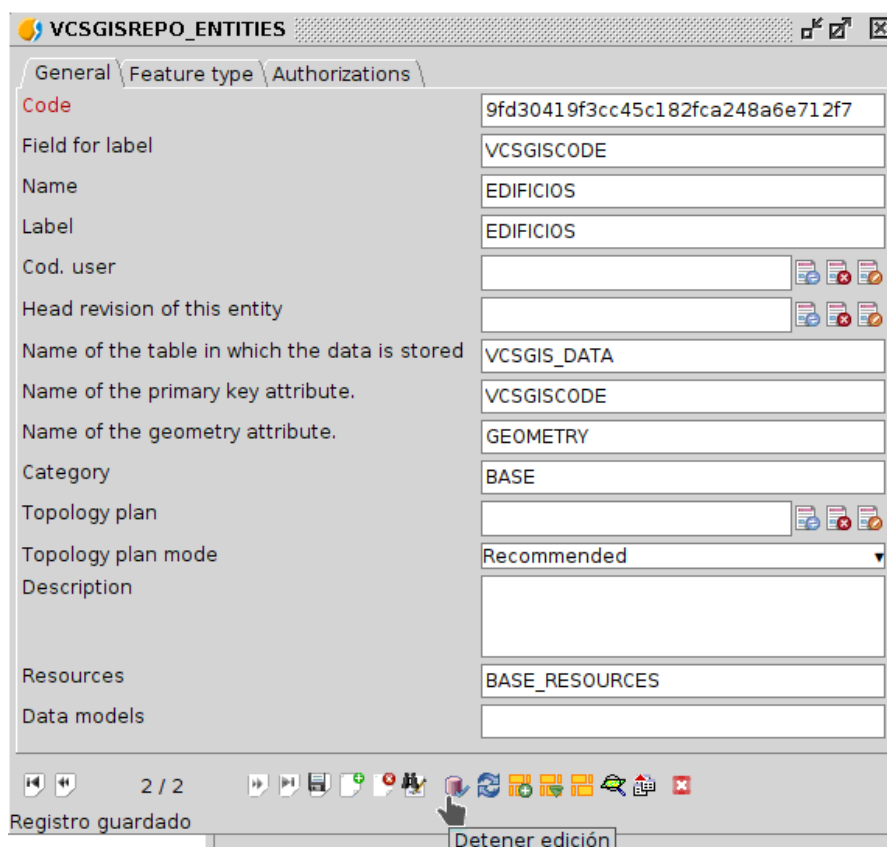
Resources: BASE\_RESOURCES

Data models: [empty]

2 / 2

Guardar

Y terminar la edición de la tabla.



**VCSGISREPO\_ENTITIES**

General | Feature type | Authorizations

Code: 9fd30419f3cc45c182fca248a6e712f7

Field for label: VCSGISCODE

Name: EDIFICIOS

Label: EDIFICIOS

Cod. user: [empty]

Head revision of this entity: [empty]

Name of the table in which the data is stored: VCSGIS\_DATA

Name of the primary key attribute: VCSGISCODE

Name of the geometry attribute: GEOMETRY

Category: BASE

Topology plan: [empty]

Topology plan mode: Recommended

Description: [empty]

Resources: BASE\_RESOURCES

Data models: [empty]

2 / 2

Registro guardado

Detener edición

Tras lo anterior la asignación de los recursos mediante la tabla de recursos *BASE\_RESOURCES* a la capa *EDIFICIOS* ha concluido.

Para confirmar el proceso de asignación de un recurso de la tabla de recursos a una capa solo hay que hacer checkout de esta tabla en cuestión, tal y como se indica en el apartado [Añadir una capa del repositorio](#).

Tras todo lo realizado anteriormente cualquier usuario que realice una descarga del repositorio de la tabla *BASE\_RESOURCES* y la capa *EDIFICIOS* presentará la misma leyenda asociada.

## 7 MODELOS DE DATOS

### Introducción

Trabajar con problemas complejos muchas veces precisa realizar un análisis adecuado, y definir un modelo de datos para representarlos de manera correcta. Normalmente el modelo de datos se materializa en un conjunto de tablas, registros, relaciones y restricciones de una base de datos relacional.

Con *gvSIG Desktop* se puede acceder a esas tablas, realizar consultas, visualizar sus datos y si tienen información geográfica representarla en un mapa. De modo que, es posible ver cada una de las tablas que conforman el modelo de forma independiente o ver ese conjunto de tablas como un modelo de datos coherente, como un todo. Si se adopta la última opción, hay que añadir a *gvSIG Desktop* información para que el software sea consciente de las relaciones que hay entre tablas, las restricciones establecidas para algunos de sus atributos, así como cual es la forma de acceder a cada una de las tablas del modelo de datos. Normalmente se emplean dos herramientas de *gvSIG Desktop* para realizar lo anterior, el *gestor de columnas* y los *repositorios de datos* o *espacios de trabajo de base de datos*.

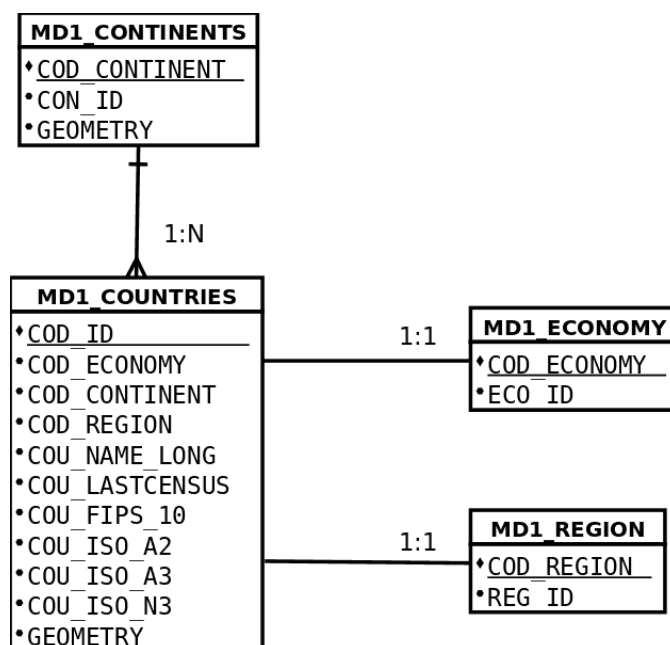
A través del *gestor de columnas* se declaran que relaciones existen entre las tablas. Por ejemplo, si se tiene una tabla de *facturas* y otra de *líneas de factura*, se define una relación entre ambas, indicando como obtener las líneas de una factura, o como obtener la factura asociada a una línea de factura. *GvSIG Desktop* permite declarar este tipo de relaciones entre las distintas tablas o entidades del modelo de datos, y luego cuando se precisa presentar esta información a través de formularios, adapta la presentación para que se pueda navegar entre las relaciones declaradas. Además, es posible usar estas relaciones para poder realizar búsquedas entre las distintas tablas del modelo de forma sencilla y más o menos transparente para el usuario.

Pero para poder “navegar” entre las distintas tablas de un modelo de datos, no solo es necesario declarar las relaciones que existen entre ellas; también se debe declarar como se puede acceder a cada una de las tablas que componen el modelo. Para esto hay que definir un *repositorio de datos*, en el que se indica donde esta almacenada cada tabla del modelo y como acceder a ella.

Centrando la atención en *VCSGIS*, se ha integrado una herramienta que ofrece la posibilidad de cargar desde el repositorio del control de versiones que tanto la información relacionada con las restricciones y relaciones tablas así como en donde se encuentran cada una de las tablas del modelo. De esta manera se puede almacenar la definición del modelo de datos para tablas ya integradas en el repositorio, pudiendo conectarse a ese modelo de datos, descargarlo y registrarlo de forma cómoda para el usuario.

Para ilustrar como configurar la parte de configuración de modelos de datos en el contexto de *VCSGIS* vamos a trabajar un modelo sencillo, con solo unas pocas tablas.

Las tablas seleccionadas para realizar el ejemplo son *countrie*, *continent*, *region* y *economy*. Estas tablas representan países, continentes, subregión y el tipo de región según su economía respectivamente. Las relaciones establecidas entre capas se detallan en la siguiente ilustración:



## Configuración del modelo de datos en VCSGIS

Configurar un modelo de datos en *VCSGIS* requiere cierto análisis y organización. No está pensado para crearse sin una planificación previa. Antes de añadir las tablas que forman el modelo se deben tener bien claras y declaradas las relaciones entre ellas, así como las restricciones que se buscan aplicar a los atributos de estas. Adicionalmente, también hay que declarar como se deben presentar los datos de las distintas tablas al usuario, normalmente en forma de *formularios*.

Para realizar toda esta tarea hay que apoyarse en las herramientas de *gvSIG desktop* de *Crear nueva tabla o capa* y el *Gestor de columnas*. Una vez declarada la información que define el modelo de datos, es posible añadir las tablas que lo componen al repositorio de la herramienta de control de versiones *VCSGIS*.

Ahora solo queda indicar a *VCSGIS* que tablas de las que hay en el repositorio forman parte de un modelo de datos. Para ello se necesita disponer de acceso de escritura a la base de datos en la que reside el repositorio donde están las tablas que forman el modelo. Cargaremos en el proyecto de *gvSIG desktop* la tabla *PUBLIC.VCSGISREPO\_ENTITIES* y, bien directamente sobre la tabla o en el formulario asociado a ella, indicaremos para cada "entidad" a que modelo de datos pertenece.

Para facilitar la comprensión de la configuración de modelos de datos en la herramienta de control de versiones *VCSGIS* de *gvSIG Desktop* se utiliza el siguiente ejemplo. El ejemplo se basa en 4 tablas; **MD1\_CONTINENTS**, **MD1\_COUNTRIES**, **MD1\_ECONOMY** y **MD1\_REGION**, siendo el conjunto de ellas el modelo de datos **MD1**.

MD1_COUNTRIES								
	COU_NAM...	COU_LAS...	COD_ECO...	COU_FIPS...	COU_ISO...	COU_ISO...	COU_ISO...	COD_CONT...
1	Aruba	2010.0 6. Develo...	AA	AW	ABW	533	North Ame...	Latin Am...
2	Afghanist...	1979.0 7. Least ...	AF	AF	AFG	004	Asia	South As...
3	Angola	1970.0 7. Least ...	AO	AO	AGO	024	Africa	Sub-Saha...
4	Anguilla	-99.00 6. Develo...	AV	AI	AIA	660	North Ame...	Latin Am...
5	Albania	2001.0 6. Develo...	AL	AL	ALB	008	Europe	Europe &
6	Aland Isla...	-99.00 2. Develo...	-99	AX	ALA	248	Europe	Europe &
7	Andorra	1989.0 2. Develo...	AN	AD	AND	020	Europe	Europe &
8	United Ar...	2010.0 6. Develo...	AE	AE	ARE	784	Asia	Middle E...
9	Argentina	2010.0 5. Emergi...	AR	AR	ARG	032	South Ame...	Latin Am...

1 / 255 Total registros seleccionados.

MD1_CONTINENTS	
CON_ID	COD_CONTINENT
1	Antarctica
2	Asia
3	Africa
4	South America
5	Seven seas (ope...
6	Europe
7	North America
8	Oceania

0 / 8 Total registros seleccionados.

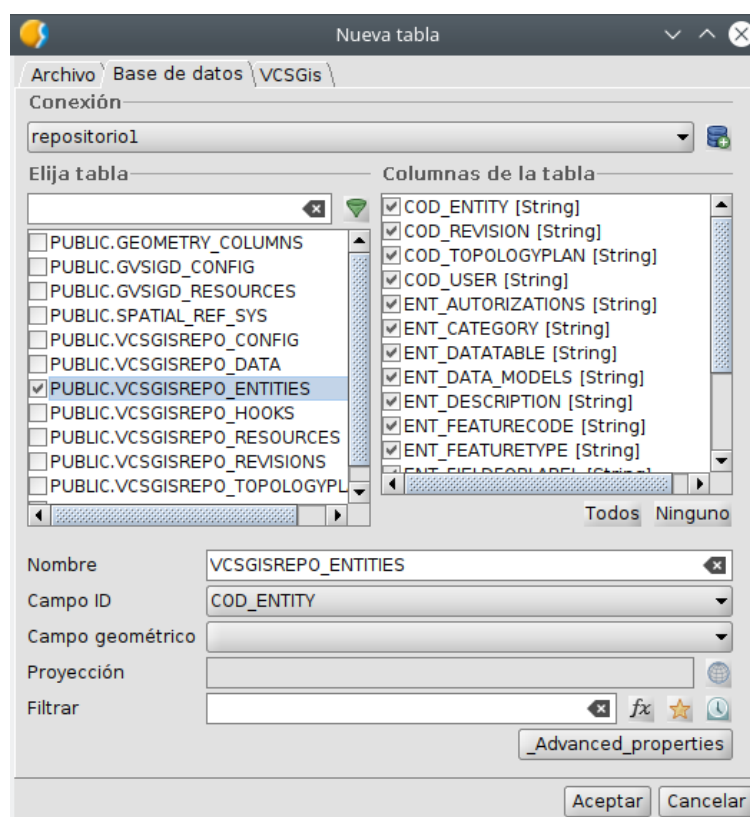
MD1_ECONOMY	
COD_ECONOMY	ECO_ID
1	6. Developing r...
2	7. Least develo...
3	2. Developed r...
4	5. Emerging re...
5	3. Emerging re...
6	1. Developed r...
7	4. Emerging re...

0 / 7 Total registros seleccionados.

MD1_REGION	
COD_REGION	REG_ID
1	Antarctica
2	East Asia & ...
3	Europe & Ce...
4	Latin Americ...
5	Middle East ...
6	North America
7	South Asia
8	Sub-Saharan...

0 / 8 Total registros seleccionados.

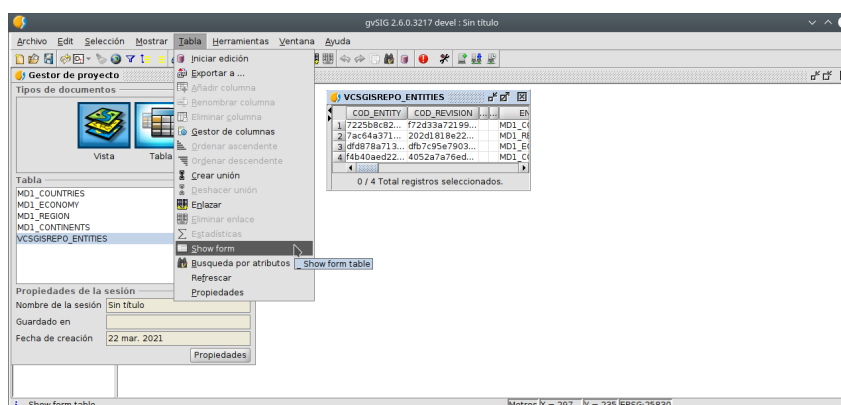
Para empezar la configuración hay que abrir la tabla del repositorio *PUBLIC.VCSGISREPO\_ENTITIES* desde el *Gestor de proyectos* situado en el menú *Mostrar* de *gvSIG Desktop*. El proceso de abrir una tabla es el genérico a abrir cualquier archivo, primero se selecciona *Tabla* como tipo de datos a abrir, se selecciona la opción de *Nuevo*, lo que habilita una ventana donde se tiene que seleccionar la pestaña *Base de datos*. Esa pestaña muestra en su zona superior un desplegable donde hay que especificar la base de datos donde se encuentra la tabla. Una vez seleccionada la base de datos, en la lista de tablas de esta hay que marcar la tabla en cuestión y pulsa el botón *Aceptar*.



De lo anterior se obtiene la siguiente tabla.

VCSGISREPO_ENTITIES						
	COD_ENTITY	COD_REVISION	...	ENT_NAME	ENT_DATATABLE	ENT_FEATURECODE
1	7225b8c82...	f72d33a72199...		MD1_COUNTRIES	VCSGIS_DATA	VCSGISCODE
2	7ac64a371...	202d1818e22...		MD1_REGION	VCSGIS_DATA	VCSGISCODE
3	dfb78a713...	dfb7c95e7903...		MD1_ECONOMY	VCSGIS_DATA	VCSGISCODE
4	f4b40aed22...	4052a7a76ed...		MD1_CONTINENTS	VCSGIS_DATA	VCSGISCODE
0 / 4 Total registros seleccionados.						

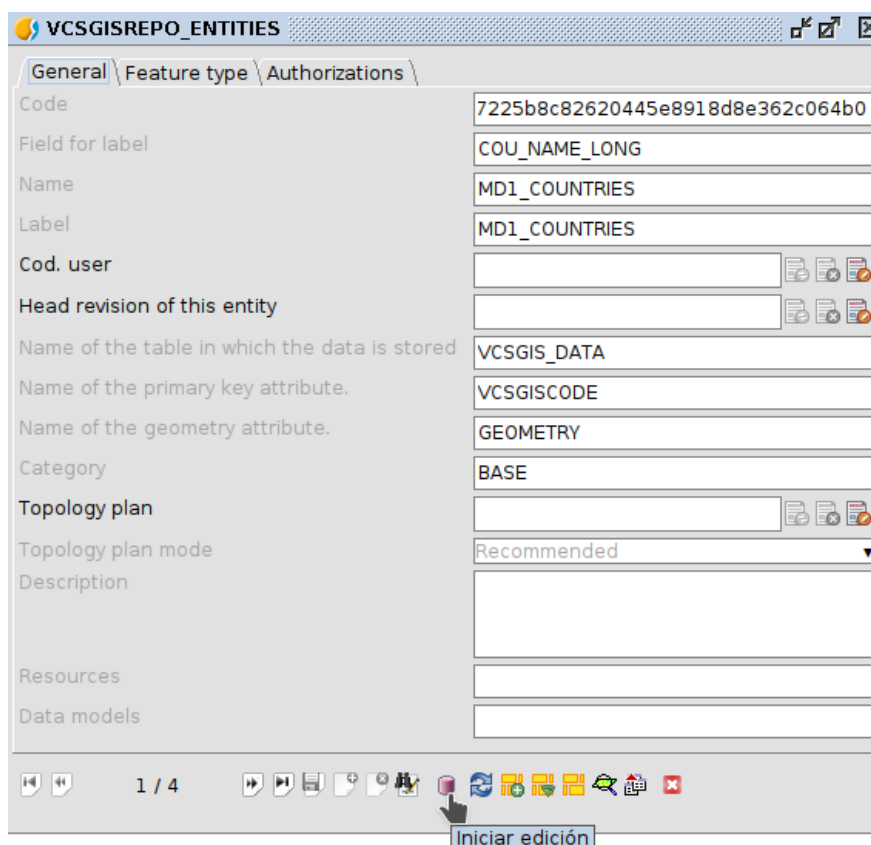
En la tabla se muestran las cuatro tablas anteriores cargadas, pero hay que especificar, que aún no conforman un modelo de datos. Para definirlo se tiene que modificar y añadir en ellas a que modelo de datos pertenecen. Para realizar lo anterior hay que editar cada tabla de manera manual o mediante su formulario. Para obtener el formulario se ejecuta la opción *Show form* situada en el menú *Tabla* de *gvSIG Desktop* siempre y cuando la tabla este abierta y seleccionada.



El formulario de la tabla *PUBLIC.VCSGISREPO\_ENTITIES* es el siguiente.

VCSGISREPO_ENTITIES	
General \ Feature type \ Authorizations \	
Code	7225b8c82620445e8918d8e362c064b0
Field for label	COU_NAME_LONG
Name	MD1_COUNTRIES
Label	MD1_COUNTRIES
Cod. user	
Head revision of this entity	
Name of the table in which the data is stored	VCSGIS_DATA
Name of the primary key attribute.	VCSGISCODE
Name of the geometry attribute.	GEOMETRY
Category	BASE
Topology plan	
Topology plan mode	Recommended
Description	
Resources	
Data models	

Para introducir un valor en la tabla hay que poner esta en edición ya sea desde el mismo desplegable que se mencionó anteriormente para obtener el formulario, o desde el mismo formulario utilizando el botón *Comenzar edición*.



VCSGISREPO\_ENTITIES

General | Feature type | Authorizations

Code: 7225b8c82620445e8918d8e362c064b0

Field for label: COU\_NAME\_LONG

Name: MD1\_COUNTRIES

Label: MD1\_COUNTRIES

Cod. user: [empty]

Head revision of this entity: [empty]

Name of the table in which the data is stored: VCSGIS\_DATA

Name of the primary key attribute: VCSGISCODE

Name of the geometry attribute: GEOMETRY

Category: BASE

Topology plan: [empty]

Topology plan mode: Recommended

Description: [empty]

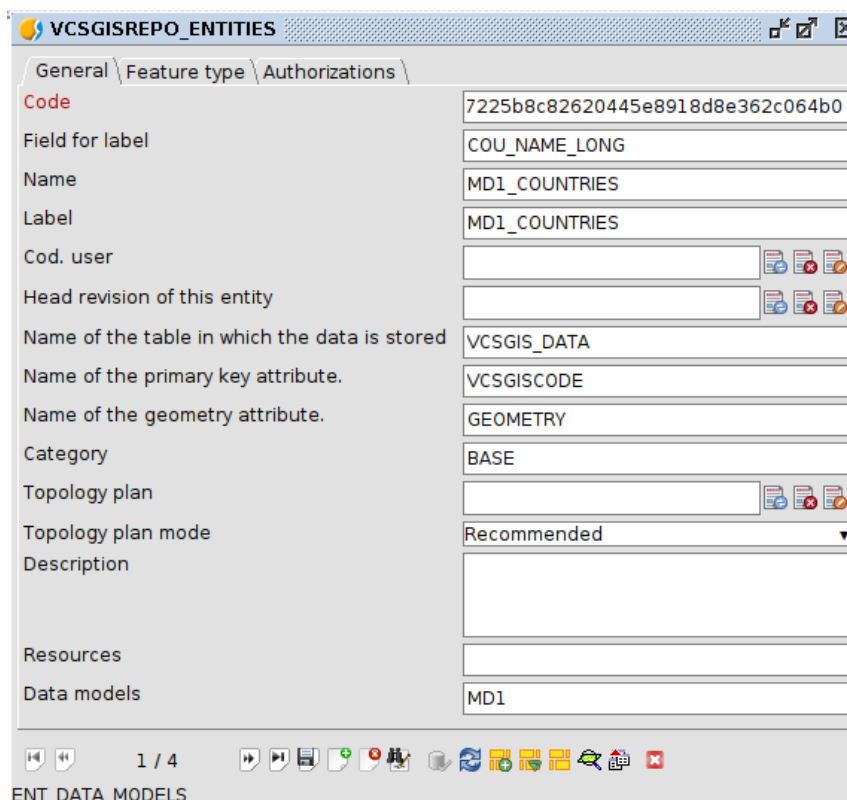
Resources: [empty]

Data models: [empty]

1 / 4

Iniciar edición

Tras esto se introduce en el campo *Data models* el modelo de datos al cual pertenece cada tabla o capa, en este caso *MD1*.



**VCSGISREPO\_ENTITIES**

General | Feature type | Authorizations

**Code** 7225b8c82620445e8918d8e362c064b0

Field for label COU\_NAME\_LONG

Name MD1\_COUNTRIES

Label MD1\_COUNTRIES

Cod. user

Head revision of this entity

Name of the table in which the data is stored VCSGIS\_DATA

Name of the primary key attribute. VCSGISCODE

Name of the geometry attribute. GEOMETRY

Category BASE

Topology plan

Topology plan mode Recommended

Description

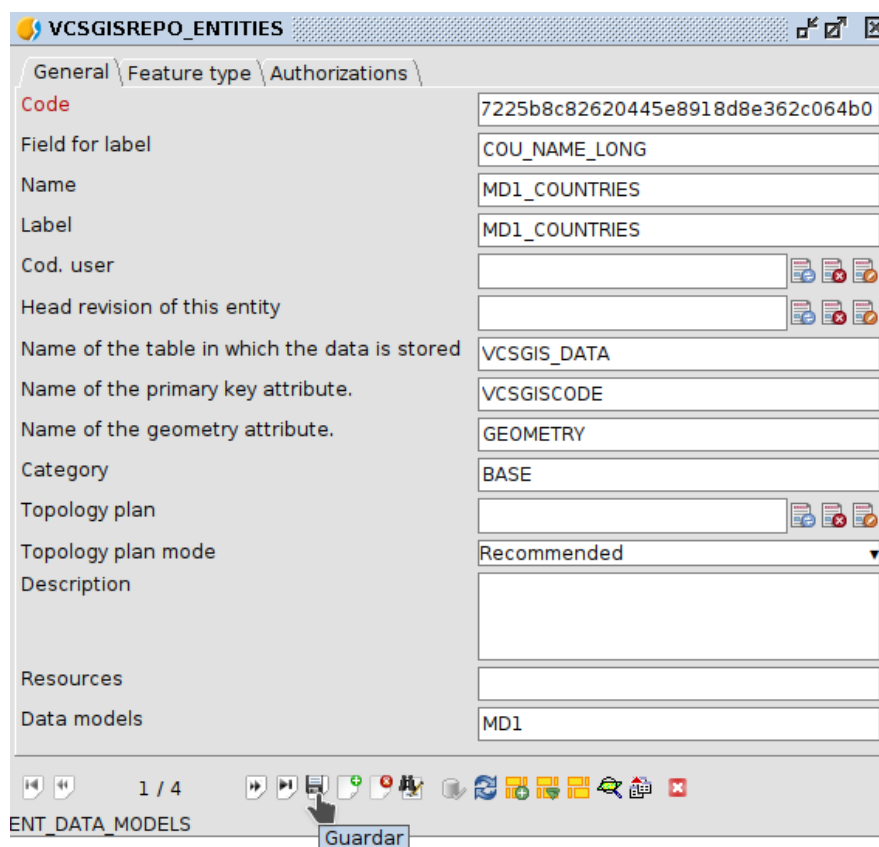
Resources

Data models MD1

1 / 4

ENT DATA MODELS

Una vez asignado el modelo de datos se guarda dicho cambio.



**VCSGISREPO\_ENTITIES**

General | Feature type | Authorizations

**Code** 7225b8c82620445e8918d8e362c064b0

Field for label COU\_NAME\_LONG

Name MD1\_COUNTRIES

Label MD1\_COUNTRIES

Cod. user

Head revision of this entity

Name of the table in which the data is stored VCSGIS\_DATA

Name of the primary key attribute. VCSGISCODE

Name of the geometry attribute. GEOMETRY

Category BASE

Topology plan

Topology plan mode Recommended

Description

Resources

Data models MD1

1 / 4

ENT DATA MODELS

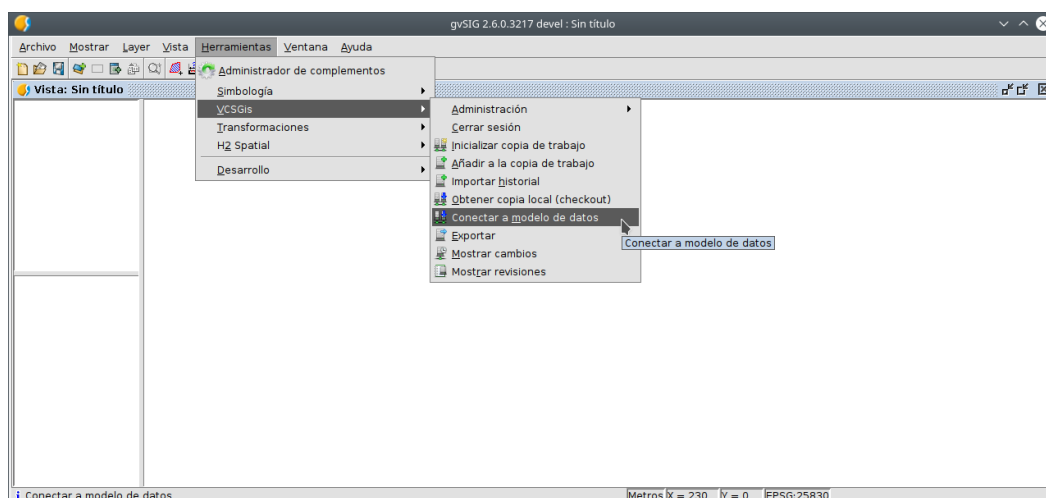
Guardar

Tras lo anterior hay que repetir el proceso con todas las tablas del modelo en cuestión y finalizar la edición de la tabla *PUBLIC.VCSGISREPO\_ENTITIES*

## Conectarnos a un modelo de datos

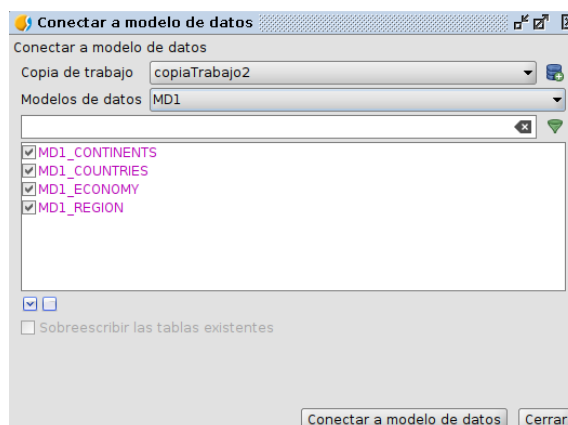
Una vez definido y cargado en el repositorio un modelo de datos, se sigue el flujo de trabajo habitual con *VCSGIS*. Crear una copia de trabajo y descargar en ellas las tablas con las que se busca trabajar. Ahora bien, se dispondrá de una herramienta que permite “conectar” a un modelo de datos. Esta herramienta presenta para una copia local determinada cuales son los modelos de datos que hay definidos en ella pudiendo seleccionar uno. Tras la selección se muestran las tablas que componen dicho modelo de datos, y si tras conectarse a el, se descargará las tablas del modelo y las registrará para que *gvSIG Desktop* pueda acceder a ellas aunque no estén cargadas como tablas o capas del proyecto actual.

Dicha herramienta se encuentra en el menú *Herramientas*, submenú *VCSGIS*, opción *Conectar a modelo de datos*.



Esa opción despliega el siguiente cuadro de diálogo en cual tras seleccionar una copia de trabajo y el modelo de datos, nos descarga en esta las capas que conforman dicho modelo.

La siguiente ilustración muestra la ventana que tras pulsar el botón *Conectar a modelo de datos* cargará en la nueva copia de trabajo *copiaTrabajo2* las capas que conforman el modelo de datos *MD1*. La lista de capas puede verse en dicha ventana.



## Consideraciones especiales sobre los modelos de datos

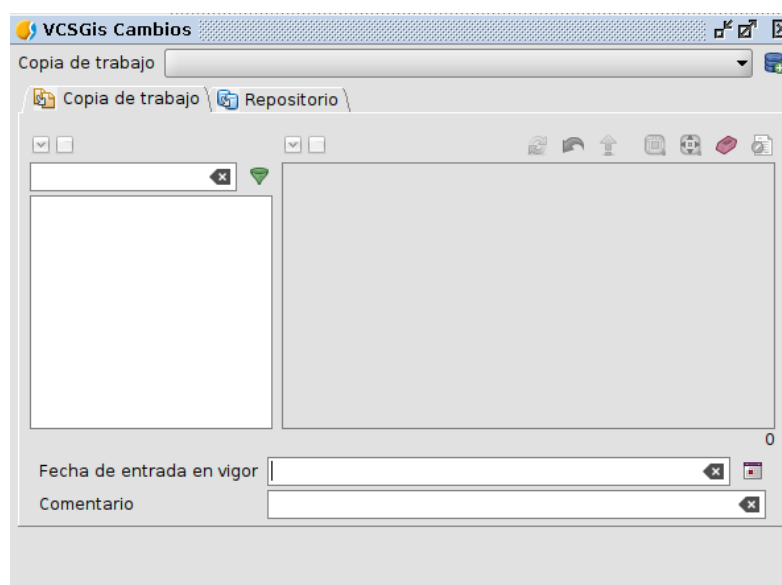
El sistema de “repositorio de datos” que tiene *gvSIG Desktop* para declarar donde esta cada tabla, así como la forma en que se definen las relaciones entre estas, usa un modelo “plano” de espacio de nombres. Es decir, no tiene esquemas o subespacios de nombres. Todas las tablas que se cargan en *gvSIG Desktop* comparten el mismo espacio de nombres. Esto implica que si se realiza una conexión simultanea a dos modelos de datos que contengan tablas con el mismo nombre estos colisionaran, pisándose unas tablas a otras.

Para evitar problemas de este tipo, sobre todo con modelos de datos que conviven en el mismo repositorio de *VCSGIS*, es recomendable que los nombres de tablas de un modelo lleven un prefijo que identifique a ese modelo. De esta forma se pueden cargar en el control de versiones varios modelos sin que colisionen entre ellos.

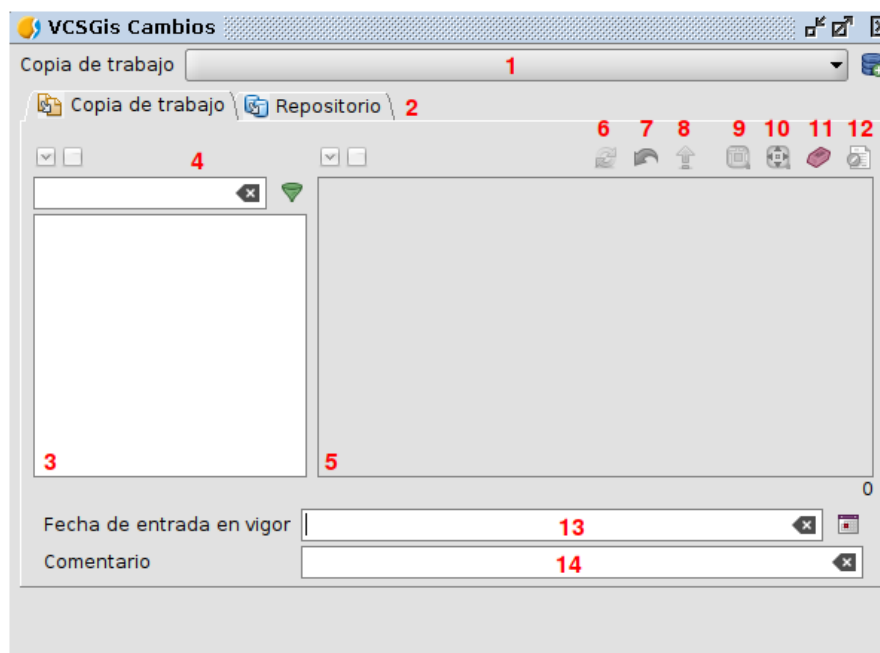
## 8 HERRAMIENTAS

### Herramienta mostrar cambios

La herramienta de *Mostrar cambios* es una de las más importante de VCSGIS. Es la encargada como su propio nombre indica de mostrar los cambios entre la copia de trabajo actual y el repositorio. Los cambios entre información se detectan seleccionando la copia de trabajo en cuestión y mediante la gestión de las pestañas *Local* y *Remoto*. Al seleccionar la pestaña *Local* se muestran los *cambios existentes en la copia de trabajo frente al repositorio*. Si por el contrario se selecciona la opción *Remoto* se muestran los *cambios del repositorio frente a la copia de trabajo*. Además de las pestañas, la ventana permite identificar la *Fecha de entrada en vigor* así como un apartado *Comentarios* asociados a los cambios que se van a enviar al repositorio.

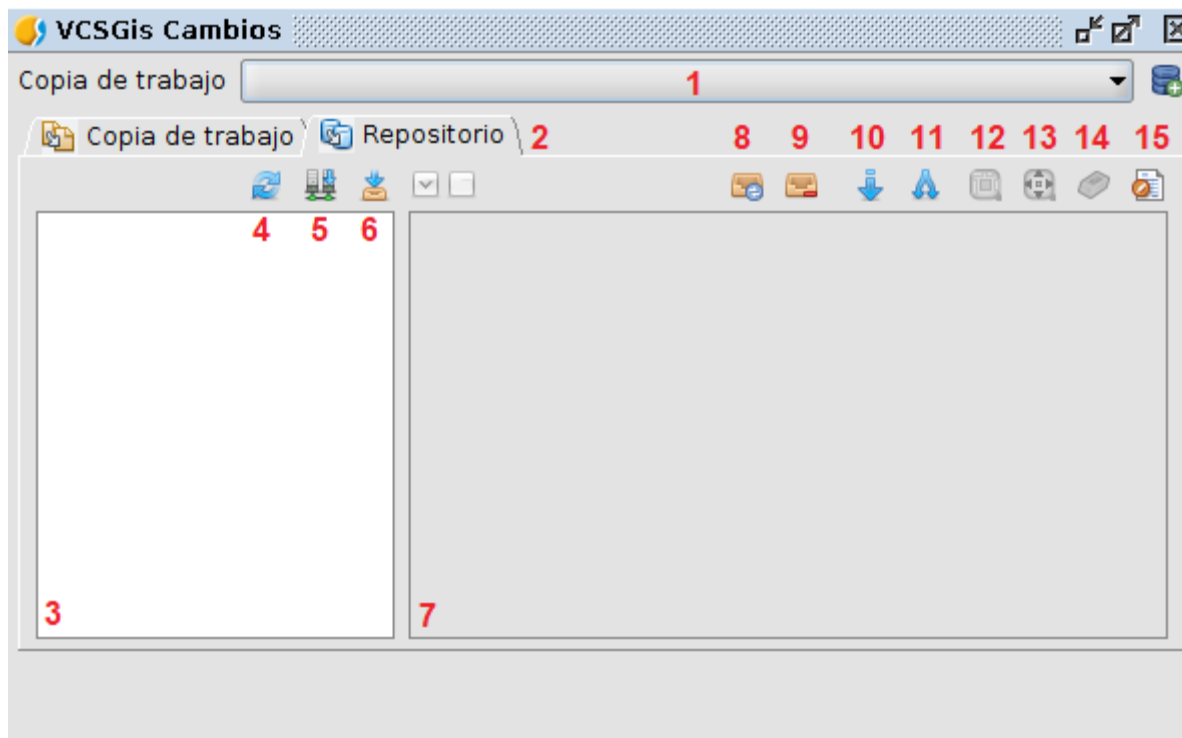


Los componentes seleccionada la opción *Local* de esta se listan a continuación:



1. Desplegable para selección de la *Copia de trabajo* sobre la que ver los cambios.
2. Pestañas Local/Remoto.
3. Área de visualización de capas presentes en la copia local. Dependiendo del color que presente en esta zona, las capas presentarán cambios locales o no.
4. Elementos de filtrado para realizar búsquedas de capas.
5. Área de visualización de cambios.
6. Botón de refrescar el área de visualización.
7. Botón *revert*. Esta función permite deshacer los cambios locales presentes en el área de visualización de cambios.
8. Botón para enviar o *confirmar* los cambios locales al repositorio (commit).
9. Botón que marca la geometría con cambios locales.
10. Botón que centra la vista en la geometría con cambios locales.
11. Botón que elimina la marca de la geometría con cambios locales, generada con el componente 6.
12. Botón que muestra un formulario con los datos del registro seleccionado.
13. Campo para indicar la fecha de entrada en vigor de los cambios que van a ser enviados al repositorio.
14. Campo para introducir un comentario a los cambios que van a ser enviados al repositorio.

Los componentes seleccionada la opción *Remoto* de esta se listan a continuación:



1. Desplegable para selección de la *Copia de trabajo* sobre la que ver los cambios.
2. Pestañas Local/Remoto.
3. Área con lista de capas de la *Copia de trabajo*.
4. Botón de refrescar la lista de capas de la *Copia de trabajo*.
5. Botón que permite hacer de manera directa una copia local (checkout) de la capa seleccionada.
6. Botón para descargar los cambios que han habido en la capa seleccionada en el repositorio desde la ultima vez que se actualizó esta en la *Copia de trabajo*. Esta operación puede ser pesada dependiendo de la cantidad de cambios que hayan en el repositorio desde la ultima actualización de la *Copia de trabajo*.
7. Área de visualización de cambios. Muestra los cambios que han habido en el repositorio en relación a las capas de la *Copia de trabajo*.
8. Actualiza el área de visualización de cambios, releyendo estos de la información almacenada en la *Copia de trabajo*. No accede al repositorio para actualizarlos.
9. Elimina la lista de cambios de la *Copia de trabajo*. El usuario deberá descargarlos de nuevo del repositorio en caso de que desee verlos.
10. Botón *Update*. Se actualizaran las tablas locales, *Copia de trabajo*, con los cambios del repositorio. Si hay cambios locales se perderán. Esta opción no esta disponible por

defecto si existen conflictos entre los cambios del repositorio y los cambios realizados en la *Copia de trabajo*. En el caso que existiendo conflictos se busque prevalecer la información remota frente a la local, se tendría que marcar todos los cambios del repositorio, proceso que habilita dicho botón en la ventana *Mostrar cambios*.

11. Botón *Merge*. Se mezclan o fusionan los cambios de la capa del usuario de la *Copia de trabajo* con los cambios del repositorio. El proceso es simple, los cambios remotos que no tengan conflicto, cambios de color diferente a rojo, se actualizan en la *Copia de trabajo*, al igual que los cambios con conflictos, cambios de color rojo, con el check seleccionado. Marcar el check de los cambios en conflicto implica que prevalecerán los cambios del repositorio para esos determinados elementos frente a los existentes en la *Copia de trabajo* o capa local. Si lo que se busca es mezclar la capa local y remota asegurando que los cambios locales sean los finales, no hay que marcar ningún cambio en conflicto y realizar la operación *Merge*. Esta operación solo estará activa si se han detectado conflictos entre la *Copia de trabajo* y el repositorio.
12. Botón que marca la geometría con cambios en el repositorio.
13. Botón que centra la vista en la geometría con cambios en el repositorio.
14. Botón que elimina la marca de la geometría con cambios en el repositorio, generada con el componente 12.
15. Botón que muestra un formulario con los datos del registro seleccionado.