

PROYECTO FINAL DE CARRERA

Aplicativo Web para manipulación
de entidades geográficas



Alumno:

Bacigalupo, Joaquín Andrés

Director del Proyecto:

Ing. Carlos Giorgetti

Director del Proyecto:

Ing. Arriondo Pedro

2016

PÁGINA EN BLANCO

[ANÁLISIS DE DOMINIO](#)

[Problemática](#)

[ARQUITECTURA](#)

[Geoserver](#)

[Ventajas](#)

[Justificación de elección](#)

[PHP](#)

[Ventajas](#)

[Justificación de elección](#)

[Apache](#)

[Ventajas](#)

[Justificación de elección](#)

[Apache Tomcat](#)

[Ventajas](#)

[Justificación de elección](#)

[PostGis, PostgreSQL](#)

[Ventajas](#)

[Justificación de elección](#)

[JavaScript](#)

[Ventajas](#)

[Justificación de elección](#)

[Openlayers](#)

[Ventajas](#)

[Justificación de elección](#)

[GeoExt – ExtJS Sencha 3.4](#)

[Ventajas](#)

[Justificación de elección](#)

[Debian](#)

[Ventajas](#)

[Desventajas](#)

[Justificación de elección](#)

[Herramientas](#)

[Justificación herramientas Open Source y Libres](#)

[NetBeans IDE](#)

[Subversion Tortoise Tigris](#)

[Ventajas](#)

[Instalación de servidores](#)

[PostgreSQL](#)

[PostgreSQL Configuración:](#)

[Subversion](#)

[Apache Tomcat](#)

[Instalación Geoserver](#)

[Instalación SSH](#)

[Instalación de la aplicación](#)

http://ip.de.mi.server/map_viewer/DESARROLLO DE LA SOLUCIÓN

Metodologías de desarrollo

Desarrollo prototipado evolutivo

Primer Prototipo

Segundo prototipo

Tercer prototipo

Cuarto prototipo

Funcionalidad restante

Quinto prototipo

Sexto prototipo

Funcionalidad restante

Descripción de los Procesos Automáticos

Árbol de capas por usuario

Persistencia de datos geográficos

Elecciones de desarrollo

Tree panel - geoExt

lecciones aprendidas

Selección de tipo de capa

Manejo de estilos:

Estilo para modificación de entidades

Exportar capas vectoriales a shapes

Control de región habilitada para dibujar

Manejo de estados en WFS-T

FeatureNS

Renderers

Base de Datos

atributos por capa

capas asociadas

estilo_x_capa_x_usuario

limites_x_capa_x_usuario

usuarios

CONCLUSIÓN

BIBLIOGRAFÍA

ANEXOS

Preguntas Frecuentes

¿Cómo se representa la curvatura de la Tierra en mapas?

GEOIDE:

ELIPSOIDE:

DATUM:

¿A que se le denomina SRE o CRS?

¿Que es WGS 84 o EPSG:4326?

El World Geodetic System 1984 (WGS84) o su equivalente EPSG:4326.

¿Qué "elipsoide de referencia" usa Argentina?

¿Qué es un Sistema de Coordenadas?

¿Que sistema de coordenadas se emplea en Argentina?

¿Qué es una base de datos geoespacial?

Manual de Usuario

Interfaz de Edición:

Árbol de capas

[Tipos de Icono:](#)

[Tipo de Capa:](#)

[Menú Contextual de Capas](#)

[Edición de capas:](#)

[Edición de Estilos:](#)

[Panel Central:](#)

[Edición de Entidades Geográficas:](#)

[Edición de Puntos:](#)

[Insertar nuevos puntos](#)

[Modificar puntos existentes](#)

[Eliminar puntos existentes](#)

[Seleccionar Puntos](#)

[Control Snap para edición e inserción](#)

[Edición de Líneas o Segmentos:](#)

[Insertar nuevos Segmentos](#)

[Modificar Segmentos existentes](#)

[Eliminar Línea existente](#)

[Seleccionar Líneas](#)

[Control Snap para edición e inserción](#)

[Edición de Polígonos:](#)

[Insertar nuevo Polígono](#)

[Modificar Polígono existente](#)

[Eliminar Polígono existente](#)

[Seleccionar Polígonos](#)

[Control Snap para edición e inserción](#)

[Generación de Huecos](#)

[Herramientas adicionales](#)

[Medición por segmentos](#)

[Medición por Área](#)

[Medir caja contenedora.](#)

[Panel Superior](#)

[Edición de Perfil](#)

[Configuración](#)

[Ayuda](#)

[Salir](#)

[Otras opciones de menú Contextual:](#)

[Transparencia](#)

[Exportar a Shapefile](#)

ANÁLISIS DE DOMINIO

Problemática

Actualmente la IDESF (Infraestructura de Datos Espaciales de Santa Fe) cuenta con servicios de mapas Web (basados en el estándar WMS “Web Map Services”), este estándar definido por el Open Geospatial Consortium (OGC), define tres operaciones básicas:

- Devolver metadatos del nivel de servicio.
- Devolver un mapa cuyos parámetros geográficos y dimensionales han sido bien definidos.
- Devolver información de características particulares mostradas en el mapa (opcionales).

El estándar citado no permite la edición de los mapas con que cuenta actualmente la IDESF, por lo que la carga de datos debe ser realizada mediante programas externos, con los inconvenientes que esto conlleva para la carga distribuida y masiva de datos, a lo largo de la provincia.

Por esto es que la IDESF se ve en la necesidad de desarrollar una aplicación que de solución a esta problemática, que cumpla con los lineamientos legales de la Ley Provincial N° 12.360 de la ciudad de Santa Fe, y le permita ajustarse de esta forma a la Directiva Inspire¹ que ha definido de forma exhaustiva la tipología de servicios que deben ofrecer los nodos y geoportales de cualquier IDE.

En uno de sus artículos enumera varias categorías de servicios:

- Servicios de localización (mostrar el contenido de los metadatos).
- Servicios de visualización (mostrar, navegar, acercarse o alejarse).

¹ Infrastructure for Spatial Information in Europe (<http://www.idee.es/web/guest/europeo-inspire>)

- Servicios de descarga (descargar copias).
- Servicios de transformación y servicios de “acceso a servicios”

Si las dos primeras funcionalidades se enmarcan en el concepto de “navegación” propio de la ‘web 1.0’, la transformación y los servicios agregados abren una nueva perspectiva.

Con su efectiva implementación se daría paso a una interacción directa con los datos espaciales, más que con los mapas. La apertura de servicios de transformación situaría a la red de IDE en el marco de la ‘web2.0’.

En esta nueva perspectiva, una IDE de segunda generación tendría que implementar funcionalidades, que se ajusten a lo que InspirE define como **“servicios de transformación”** y **“servicios de acceso a servicios”**, mediante los cuales el usuario adquiera capacidades de edición sobre los datos espaciales.

El concepto de transformación ha de entenderse no sólo como migración de los datos a distintos formatos, modelos o sistemas geodésicos, sino como la facultad de interacción con la información para crearla, editarla, actualizarla, bloquearla y publicarla.

Una IDE realmente colaborativa y fundada en la participación de usuarios productores organizados en comunidades requiere de nuevos estándares que permitan compartir los datos espaciales, con garantías de interoperabilidad.

Los protocolos para compartir datos están implícitos en la especificación “WFS-T” (“Servicio Transaccional de entidades vectoriales”), aunque estos estén en constante evolución.

La capacidad transaccional abre las posibilidades para colaboraciones a través de Internet. Los usuarios ya no necesitarán permisos de acceso a la misma base de datos espacial al usar el estándar WFS-T. Esto permite gestionar datos geográficos realmente abiertos, al igual que el trabajo en red y las

aplicaciones de gestión permitieron el desarrollo del movimiento de software libre”.

El acceso compartido a la misma base de datos de forma estandarizada asegura una edición conjunta de la información, esencial en cualquier proyecto colaborativo.

Por acceso a los datos debe entenderse no sólo la posibilidad de consultar (búsqueda, filtrado o simbolización) sino la **edición** de esos datos, tal y como se realiza en un entorno SIG (Sistema de información Geográfico), pero ahora de forma remota a través de la red.

Estas funcionalidades de edición están contempladas entre las operaciones ‘Transaccionales’, incluidas en la especificación WFS-T. Conforme a la definición del OGC “la operación de transacción es usada para describir las operaciones de transformación aplicadas a elementos accesibles vía web”.

Un servicio Web de elementos puede realizar una operación de transacción directamente o traducirla al lenguaje del repositorio de datos con el que se conecta, para posteriormente realizar la transacción. Las operaciones ‘Transaccionales’ se descomponen a su vez en tres elementos básicos: <Insert>, <Update> y <Delete>.

El aplicativo generado será diseñado para ejecutar sobre cualquier navegador con soporte JavaScript en el lado Cliente, e interactuará con el servidor de mapas Geoserver que brinda las herramientas para la lectura y escritura de datos geográficos.

La estructura propuesta es la que se detalla en el gráfico siguiente (Figura 001).

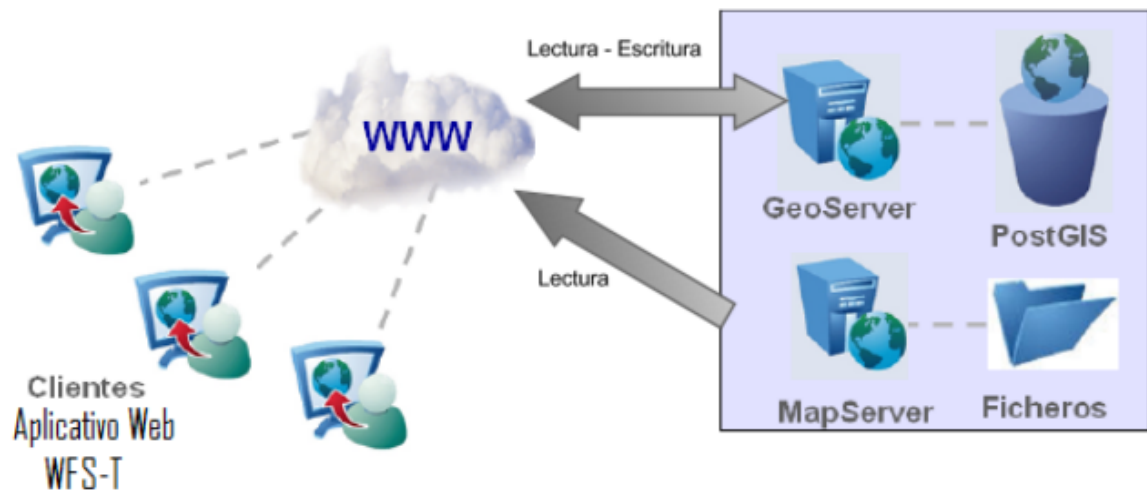


Figura 001

ARQUITECTURA

Enmarcado en la Ley Provincial N° 12.360 se estudiaron herramientas de software libre para la implementación y desarrollo de este producto, de entre las cuales se seleccionaron las siguientes:

- Geoserver
- PHP
- Apache
- Tomcat
- PostGis
- Javascript
- OpenLayers
- GeoExt
- Ext JS
- Debian Server

Geoserver

Geoserver es un Servidor de mapas de código abierto escrito en Java que les permite a los usuarios compartir y editar datos geoespaciales. Este está diseñado para interoperabilidad, ya que permite publicar datos geográficos desde la mayoría de las fuentes de datos espaciales basadas en estándares de código abierto de más amplia difusión.

Este se ejecuta sobre un servidor Tomcat, el cual es una implementación libre de Java Servlet y JavaServer Pages technologies.

Geoserver es desarrollado, probado y soportado por un grupo diverso de individuos y organizaciones a lo largo de todo el mundo.

Es referencia de implementación y está certificado para los estándares Web Feature Service (WFS) y, Web Map Service (WMS), definidos por el OGC..

VENTAJAS

- Es multiplataforma.
- ***Es un verdadero servidor de mapas a diferencia de su competidor más directo Mapserver que es simplemente una aplicación enmascarada detrás de un servidor web apache.***
- Capacidad de conexión con la mayoría de los manejadores de base de datos espaciales más reconocidos, principalmente PostGis que es una base de datos geoespacial basada en el motor de datos PostgreSQL, como así también el soporte de los más importantes formatos de archivos geoespaciales, que se utilizan en la actualidad
- Provee los dos servicios fundamentales para el desarrollo de esta aplicación web, que son WMS y WFS y WFS-T.
- Posee una amplia documentación en su página oficial, en la cual se explican todos los pormenores del sistema desde su instalación hasta su puesta en producción, cubriendo aspectos técnicos y no técnicos.
- Es libre, bajo licencia GNU, por lo que se ajusta a la Ley Provincial 12.360 de software libre.

- El servidor se ejecuta sobre un Servidor Tomcat por lo que permite ocultar por medio de un proxy sus funcionalidades y estructuras brindando gran seguridad a los servicios.
- Permite la utilización de cualquier sistema de referencia geoespacial o datum
- Posee una interfaz web de configuración que facilita la gestión de las capas y servicios publicados
- Maneja el estándar Styled Layer Descriptor (SLD) para el manejo de estilos.

JUSTIFICACIÓN DE ELECCIÓN

Principalmente, por ser un Software open source, por lo que se ajusta a la Ley Provincial Nº 12.360 de software libre, cuenta con la certificación de OGC. Para los servicios que este proporciona.

Se ajusta perfectamente a las librerías OpenLayers que son las que dan soporte para la visualización y gestión de las entidades geográficas desde el aplicativo web.

Se cuenta con una extensa documentación en la URL del proyecto, y por ser un software de gran difusión se puede encontrar en toda la web documentación y ejemplos de implementaciones.

El proyecto está en constante evolución y actualización.

Permite la utilización en modo balanceado montando múltiples servidores Geoserver para dar respuesta a las solicitudes y permitir redundancia.

Soporta todos los tipos de datum estándar y además la definición de nuevos sistemas de referencia de ser necesarios.

El servidor cuenta con una interfaz web de configuración que permite ajustar todos los parámetros de funcionamiento de los servicios y facilita la carga de capas y su publicación.

Los cambios y ajustes pueden realizarse de modo en línea, sin necesidad de reinicios programados ni modificaciones engorrosas de archivos de definición de mapas.

Brinda la posibilidad de securizar el servidor, restringir el acceso a las capas o a los servicios mediante perfiles o roles de usuarios completamente configurables desde su interfaz web.

Entre los servidores evaluados, Mapserver y Geoserver se optó principalmente por este último, ya que provee como módulo incorporado el servicio WFS-T, en contraposición con Mapserver que requiere de un módulo separado llamado Tiny que debe ser instalado de forma separada de Mapserver

PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting).

Es un lenguaje ampliamente utilizado y puede ser embebido dentro de código HTML, permite la interacción plena con Javascript (Lenguaje del lado cliente).

Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. PHP se encuentra instalado en más de 20 millones de sitios Web y en más de un millón de servidores.

VENTAJAS

- Es un lenguaje multiplataforma.
- Soporte para la conexión con la mayoría motores de base de datos que se utilizan en la actualidad, principalmente PostGis
- Gran potencial de expansión mediante la utilización de módulos (extensiones).
- Amplia documentación en su página web oficial, entre la cual se destaca que todas las funciones están explicadas, ejemplificadas y presentadas en varios idiomas.
- Es libre, por lo que se ajusta al requerimiento legal que se exige para la IDESF.
- Está en constante evolución y cuenta con actualizaciones regulares que lo convierten en el producto de software más difundido.

JUSTIFICACIÓN DE ELECCIÓN

PHP al ser una herramienta open source, hace que el desarrollador tenga a su disposición inmediatamente, y de forma gratuita, gran cantidad de recursos, el lenguaje es simple de ejecutar como así también es simple de configurar y poner en funcionamiento los elementos para soporte del mismo, ya sean servidores web o la posibilidad de integrarse con bases de datos, como también interactuar con otros lenguajes de programación. Esta abundancia de código libremente disponible ayuda aún más en el proceso de desarrollo.

Se cuenta además con una enorme comunidad de desarrolladores, soporte oficial existente y su licencia gratuita hacen que PHP resulte el lenguaje base indicado para el desarrollo de esta aplicación.

Para hacer las cosas aún más simples se seleccionó una arquitectura CLIENTE-SERVIDOR pura, que diferencia claramente el servicio del lado servidor, y los aplicativos Web del lado cliente.

Apache

El servidor Web Apache es de código abierto y multiplataforma. Ya que existen distribuciones disponibles para múltiples sistemas operativos tales como Linux, UNIX, Windows, Macintosh y otros, es además ampliamente aceptado y difundido, por su gran versatilidad y facilidad de configuración y expansión.

El servidor está estructurado en módulos. La configuración de cada módulo se hace mediante la inclusión de directivas que están contenidas dentro del mismo módulo.

Estos se pueden clasificar en tres categorías:

- *Módulos Base*: Módulo con las funciones básicas del Apache
- *Módulos Multiproceso*: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones
- *Módulos Adicionales*: Cualquier otro módulo que le añada una funcionalidad al servidor, que en el caso de esta aplicación son particularmente dos.

mod_proxy - Este módulo implementa un proxy/Gateway.

mod_proxy_http - Permite realizar proxy a nivel de solicitudes http.

VENTAJAS

- *Personalizable.* La arquitectura modular permite construir un servidor hecho a medida
- *Administración.* Los archivos de configuración son en ASCII, tienen formato simple y pueden ser editados usando cualquier editor de texto.
- *Eficiencia.* El gran esfuerzo realizado por toda la comunidad que da soporte a Apache en refinar su codificación, dio como resultado un servidor veloz y que consume menos recursos de sistema que cualquier otro servidor.
- *Portabilidad.* Apache corre en una amplia variedad de sistemas operativos, principalmente en cualquier sistema operativo Linux o Unix.
- *Estabilidad/Confiabilidad.* El código de Apache está abierto al público. Cuando se detectan errores, éstos son rápidamente comunicados y corregidos. Cuenta con actualizaciones periódicas. Esto hace de Apache un producto estable y confiable a largo plazo.
- *Soporte.* Existe una muy grande y dedicada comunidad de usuarios alrededor del mundo que dan soporte tanto al mantenimiento como a la configuración de servidores Apache.

JUSTIFICACIÓN DE ELECCIÓN

Se eligió Apache como servidor Web para este aplicativo, ya que es el servidor Web por excelencia más utilizado en Internet, posee un alto grado de acoplamiento con el lenguaje de desarrollo seleccionado para la aplicación (PHP, Javascript), y un soporte exhaustivo existente en libros, manuales, tutoriales, y además factor muy importante es el servidor web que actualmente utiliza la IDESF.

Apache Tomcat

Apache Tomcat es una implementación de software a código abierto de Java Servlet y JavaServer Pages technologies. Las especificaciones de Java Servlet y JavaServer Pages están desarrolladas según Java Community Process².

Tomcat está desarrollado en un ambiente abierto y participativo, realizado bajo la licencia [Apache License version 2](#), y es un producto registrado de Apache Software Foundation.

Este es un intento por ser un proyecto basado en el aporte de los mejores esfuerzos de todos los desarrolladores que participan de él a lo largo de todo el mundo.

Brinda actualmente soporte a gran cantidad de aplicaciones de gran escala de todo tipo de industria y organizaciones, entre ellas y por nombrar solo algunas WallMart y The Weather Chanel.

VENTAJAS

- Es un producto probado y con gran aceptación y difusión.

² <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

- Es totalmente compatible con Geoserver, es también el servidor recomendado por el desarrollador de este y sobre el cual certifican el funcionamiento correcto del producto.
- Tiene el soporte y el respaldo de Apache Software Foundation.
- Cuenta con una extensísima documentación y actualizaciones periódicas.

JUSTIFICACIÓN DE ELECCIÓN

Principalmente por ser el servidor java recomendado por el desarrollador de Geoserver que es la plataforma de servicio de mapas seleccionada para soportar los servicios WFS y WFS-T que se precisan para este aplicativo.

Es además el servidor java con el que ya cuenta la IDESF el sus servidores de producción, y su acople con Apache es nativo fue diseñado para integrarse con este.

PostGis, PostgreSQL

Es el soporte geográfico para la base de datos postgresQL, en efecto PostGis habilita espacialmente al motor Postgre.

Las estructuras y las definiciones de datos se ajustan a las especificaciones de OGC, dadas en su normativa “Simple Features Specification for SQL”³, cuenta además con la certificación oficial del OGC.

Es un proyecto bajo licencia GNU como lo es también el motor de datos PostgreSQL.

³ <http://www.opengis.org/docs/99-049.pdf>

Cuenta con actualizaciones periódicas como, también con una extensa documentación, manuales y ejemplos que simplifican su utilización y configuración.

Aporta un conjunto de herramientas para la manipulación de entidades geográficas que lo hacen muy atractivo para su utilización.

Es la base de datos espacial bajo licencia GNU por excelencia, y se integra perfectamente con los demás elementos seleccionados para el desarrollo de este proyecto.

VENTAJAS

- Se distribuye bajo licencia GNU por lo que se ajusta a la Ley Provincial 12.360
- Cuenta con actualizaciones periódicas y una extensa documentación en línea tanto oficial como de listas de correos.
- Soporta todos los tipos de datum que se utilizan en la IDESF.
- Ofrece funciones integradas para el manejo y manipulación de entidades geográficas, que simplifican muchos cálculos y gestiones, sobre las mismas.
- Es el motor de datos actualmente utilizado por la IDESF por lo que la integración del nuevo servicio no representaría una gran complicación a nivel estructural.

JUSTIFICACIÓN DE ELECCIÓN

Principalmente por ser el motor de datos geográficos utilizado por la IDESF, y que se encuentra actualmente en producción.

Cuenta con licencia GNU, y se ajusta perfectamente a los requerimientos del servicio nuevo a brindar.

Se integra perfectamente con los demás elementos que se precisan para la puesta en funcionamiento de este producto.

El número de funcionalidades integradas es muy importante y permiten realizar un sin fin de manipulaciones sobre las entidades en forma ágil y precisa.

Es una base de datos relacional, no un archivo secuencial de datos como lo son los archivos shapefile.

Un shapefile ⁴ es un tipo de archivo definido por ESRI ⁵, los mismos, fueron y son ampliamente utilizados hoy día, pero con el advenimiento de las bases de datos relacionales están siendo poco a poco reemplazados por las bases de datos GIS.

Otra característica fundamental es la performance lograda a la hora de manipular grandes cantidades de datos geográficos.

⁴ <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

⁵ Empresa especializada en Sistemas geográficos, que como pionera de la industria es referente en lo que a estándares se refiere.

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas

VENTAJAS

- Se integra perfectamente con PHP y HTML como así también con las librerías que se utilizan para edición y gestión de los mapas.
- El código Javascript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido; un script ejecutado en el servidor, sin embargo, sometería a éste a duras pruebas y los servidores de capacidades más limitadas se podrían resentir ante una continua solicitud por un mayor número de usuarios.
- el lenguaje de scripting es seguro y fiable.

JUSTIFICACIÓN DE ELECCIÓN

Javascript es hoy el lenguaje más difundido en aplicaciones que necesitan dinámica en la visualización del lado cliente, las librerías que se seleccionaron para esta aplicación están programadas en Javascript y se integran perfectamente unas con otras y con los productos servidores que se seleccionaron como así también con PHP (lenguaje de lado cliente).

Openlayers

OpenLayers es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores web.

OpenLayers ofrece una API para acceder a diferentes fuentes de información cartográfica en la red: Web Map Services (WMS), Mapas comerciales (tipo Google Maps, Bing, Yahoo), Web Features Services (WFS), compatible con los servidores de mapas más difundidos y perfectamente compatible con Geoserver, y principalmente brinda soporte a Web Features Services – Transactional (WFS-T), el protocolo de manipulación de entidades geográficas que nos permite realizar las funciones que en este proyecto son necesarias.

VENTAJAS

- Simplicidad de su uso.
- Independiente de servidores de mapas concretos.
- Acceso a servicios WMS, WFS y WFS-T de forma transparente para el usuario y desarrollador.
- Acceso a mapas de Google Maps, Yahoo Maps, Open Strits y Bing.

JUSTIFICACIÓN DE ELECCIÓN

Esta librería se ajusta perfectamente a las necesidades de este proyecto, es de código abierto lo cual la vuelve una elección natural para la utilización en el ámbito de la IDESF, y principalmente por la integración de los servicios necesarios para gestión de entidades geográficas.

GeoExt – ExtJS Sencha 3.4

GeoExt es una librería escrita en JavaScript que unifica el know how de OpenLayers con el dominio de ExtJS Sencha que permite crear interfaces gráficas web con gran dinamismo y uniformidad.

VENTAJAS

- Simplicidad de su uso.
- Integración directa para las estructuras de Openlayers y Javascript
- Independiente de los navegadores, e integración con los mismos.
- Gran potencia para la realización de interfaces claras y limpias para los usuarios finales.

JUSTIFICACIÓN DE ELECCIÓN

Esta librería se ajusta de manera ideal a las necesidades de este proyecto, es de código abierto, por lo cual se vuelve una buena elección para la utilización en el ámbito de la IDESF, y principalmente por la integración de los servicios necesarios para gestión de entidades geográficas, visualización, y administración de las mismas a nivel de capas y mapas.

Debian

Debian El Proyecto Debian es una asociación de personas que han hecho causa común para crear un sistema operativo (SO) libre.

El Proyecto Debian es una organización voluntaria con 3 documentos fundadores:

El Contrato Social de Debian, define las bases por las cuales el proyecto y sus desarrolladores tratan los asuntos.

Las Directrices de software libre de Debian, definen los criterios del Software libre y dictan qué software es aceptable para la distribución, según lo referido al contrato social. Estas pautas también se han adoptado como base de la definición del Open Source.

La Constitución de Debian, describe la estructura de la organización para la toma de decisiones de manera formal dentro del proyecto. Enumera el poder y las responsabilidades del Líder de Proyecto Debian, de la secretaría, y de los desarrolladores en general.

Actualmente, el proyecto incluye más de mil desarrolladores. Cada uno de ellos posee algún lugar en el proyecto ya sea relacionado con los paquetes: mantenimiento, documentación, control de calidad, o relacionado con la infraestructura del proyecto: coordinación de lanzamientos, traducciones de web, entre otros.

El proyecto mantiene listas de correo y un sistema que permite a toda la comunidad leer y reportar bugs. De esta forma, trabajan de manera conjunta usuarios y desarrolladores.

Existen otros medios de comunicación con colaboradores y usuarios, principalmente por medio del IRC y particularmente en el servidor Freenode. Todos estos medios poseen una gran concurrencia, siendo frecuentados por una multitud de usuarios y desarrolladores.

Además, los desarrolladores pueden tomar decisiones generales obligatorias para una resolución o una elección general. Toda votación es realizada por el método Schulze⁶.

⁶ Sistema de votación que selecciona un ganador basado en las preferencias de los votantes
https://es.wikipedia.org/wiki/M%C3%A9todo_Schulze

VENTAJAS

- Es libre y gratuito
- Es el OS de base para todos los servidores de la provincia y para los cuales se da soporte de infraestructura.
- Tiene una de las comunidades más grandes dando soporte y fomentando su constante mejora.
- Ofrece actualizaciones periódicas y brinda soporte gratuito durante un largo periodo, entre versiones.
- Su instalación es muy simple como así su actualización.
- Se integra perfectamente con las herramientas necesarias para este proyecto.
- Cuenta con soporte para las versiones más actualizadas del software necesario para esta aplicación.

DESVENTAJAS

- Su uso en modo consola requiere de un manejo pormenorizado de los comandos y de las estructuras de directorios del sistema operativo, por lo que requiere una adaptación ante primeros usos, no se recomienda esta modalidad para usuarios finales.
- No todos los dispositivos de hardware son soportados, por lo que hay que ser realmente cuidadoso a la hora de configurar el hardware para instalaciones de Debian en modo servidor.

JUSTIFICACIÓN DE ELECCIÓN

Este Sistema operativo está ampliamente difundido y es un fuerte competidor de las versiones de Linux más tradicionales y que en realidad no son libres ni gratuitas, en cambio Debian si lo es y además da soporte gratuito, y cuenta con una comunidad muy seria de colaboradores.

Soporta ampliamente las versiones del software necesario para la realización de este proyecto por lo cual se convierte en una elección ideal.

La gestión de los paquetes de actualizaciones y de software son muy simples de gestionar, y su estructura organizativa interna es simple de entender y administrar.

Es además, la distribución que se utiliza actualmente en la provincia, lo que permitirá que el despliegue de este proyecto sea muy simple sobre servidores ya instalados.

HERRAMIENTAS ASOCIADAS

Herramientas

JUSTIFICACIÓN HERRAMIENTAS OPEN SOURCE Y LIBRES

Software Libre es aquel que respeta la libertad de los usuarios y su comunidad, y se entiende por libertad el hecho de que el mismo se puede ejecutar, copiar, estudiar, distribuir y modificar libremente. Es decir software libre es una cuestión de libertad y no de precios, para dejar más en claro aún la idea de libre se debe interpretar como libre expresión, y no como barra libre.

Las razones para la elección de herramientas de desarrollo que sean open source y libres son muchas, entre las principales podemos mencionar:

- **Flexibilidad.** Si el código fuente está disponible, los desarrolladores pueden aprender y modificar los programas a su antojo, adaptándolo para realizar tareas específicas. Además, se produce un flujo constante de ideas que mejoran la calidad de los programas.
- **Fiabilidad y seguridad.** Con varios programadores a la vez mirando el mismo trabajo, los errores se detectan y corrigen antes, por lo que el producto resultante es más fiable y eficaz que el comercial.
- **Rapidez de desarrollo.** Las actualizaciones y ajustes se realizan a través de una comunicación constante vía Internet. Menores tiempos de desarrollo debido a la amplia disponibilidad de herramientas y librerías.
- **Relación con el usuario.** El programador se acerca mucho más a la necesidad real de su cliente, y puede crear un producto específico.
- **Libre.** Es de libre distribución, cualquier persona puede regalarlo, venderlo o prestarlo, y todo lo creado es propiedad del autor, sin necesidad del pago de licencias.
- **Piratería.** Combate efectivamente la piratería de software.

NETBEANS IDE

NetBeans es un entorno de desarrollo integrado libre, hecho originalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems⁷ fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

Su integración con los lenguajes utilizados para programar este proyecto lo vuelven una elección casi natural por su fiabilidad, integración y soporte disponibles.

Es además multiplataforma permitiendo que diferentes desarrolladores con distintos perfiles, en cuanto a su preferencia de entorno de base, puedan encontrar una herramienta común y con gran integración.

Instalación del entorno

Su instalación es simple y prácticamente desatendida tanto sobre Debian como sobre Windows, utilizando los paquetes provistos por Sun Microsystems los cuales pueden ser descargados desde la página oficial de este.

Ventajas

- Editor de texto.
- Resaltado de sintaxis.

⁷ Empresa Informática especializada en Hardware, y software principalmente Java, actualmente fue adquirida por ORACLE corporation

- Compilación en tiempo real.
- Control de versiones con SVN.
- Integración con PHP HTML y JavaScript.
- Entorno simple y claro para la gestión de proyectos.
- Refactorización.

Desventajas

- Alto consumo de memoria en el ordenador

SUBVERSION TORTOISE TIGRIS

VENTAJAS

- Se puede ver el historial de cada uno de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- Se envían sólo las diferencias en ambas direcciones.
- Permite realizar control de versiones.
- Cuenta con un integrador de diferencias simple e intuitivo.
- Cuenta con soporte e integración con NetBeans

Instalación de servidores

La instalación se realizó toda sobre un Debian 7 en versión servidor, sobre el mismo se desplegaron dos servidores Apache uno tradicional el cual hace las veces de servidor web para la aplicación, y un Apache Tomcat el cual aloja y da soporte al Geoserver quien es el servidor de mapas y nos da soporte al protocolo WFS-T.

Se instaló luego una base de datos Postgres con su correspondiente paquete de extensiones Postgis, y se desplegó y configuró la aplicación

phpPgAdmin⁸ para permitir la administración de la base de datos utilizando el entorno WEB ya disponible en el servidor.

Para la instalación del servidor postgresql se deben seguir los siguientes pasos.

POSTGRESQL

```
apt-get install postgresql
```

POSTGRESQL CONFIGURACIÓN:

Desde un terminal abrir la ruta

```
/etc/postgresql/8.4/main/postgresql.conf  
(Asumiendo que 8.4 es la versión instalada en Debian)
```

Por defecto toda conexión al servidor viene deshabilitada, para lo cual deberemos modificar algunos archivos del servidor para permitir su manipulación remota.

Realizando los siguientes cambios

Cambiar

```
#listen_addresses = localhost
```

Por (remover el numeral y especificar la ip deseada o un *)

```
listen_addresses = 192.168.0.101
```

y

```
#password_encryption = on
```

Por

⁸ Aplicación web para administración de bases de datos Postgres

```
password_encryption = on
```

Ahora deberemos especificar el usuario y password para el servidor

NOTA: Ejecutar todos los comandos como el usuario postgres
(para ingresar en el tipear: **su postgres**)

Creación de un usuario

```
createuser
```

Este comando lo guiará por algunos pasos para crear el usuario

```
Enter name of role to add: unUsuario
```

```
Shall the new role be a superuser? (y/n) n
```

```
Shall the new role be allowed to create databases? (y/n) n
```

```
Shall the new role be allowed to create more new roles? (y/n) n
```

Para manipular el estado del servidor o verificar su estado utilizar los siguientes comandos

Start the service : /etc/init.d/postgresql start

Stop the service : /etc/init.d/postgresql stop

Know the status : /etc/init.d/postgresql status

Restart the service : /etc/init.d/postgresql restart

NOTA: Lo más adecuado sería poder manipular los estados y las bases desde un sitio web como phpPgAdmin.

Para instalar los paquetes de Postgis 1.5 compatible con postgresql 8.4 se deben seguir los siguientes pasos.

```
postgis apt-get install proj postgresql-server-dev-8.4 libxml2-dev libgeos-dev
```

Una vez instalados los paquetes anteriores debemos descargar postgis 1.5 y compilarlo (la ruta web utilizada puede ser cualquiera que contenga una imagen de postgis 1.5)

```
wget http://download.osgeo.org/postgis/source/postgis-1.5.8.tar.gz  
tar xfvz postgis-1.5.8.tar.gz
```



```
cd postgis-1.5.8
./configure
make
```

Luego como **root**

```
make install
```

Los archivos de destino quedarán en

```
/usr/share/postgresql/8.4/contrib //recordar la ruta la utilizaremos
posteriormente
```

Para poder crear una base de datos utilizable con Postgis deberemos ejecutar en modo postgres (usuario postgres) los siguientes comandos:

```
createdb yourDataBase
createlang plpgsql yourDataBase
psql -d yourdatabase -f
/usr/share/postgresql/8.4/contrib/postgis-1.5/postgis.sql
psql -d yourdatabase -f
/usr/share/postgresql/8.4/contrib/postgis-1.5/spatial_ref_sys.sql
```

Asignar permisos a algún usuario con grant y create user o utilizar el gestor WEB para terminar la manipulación y asociación de usuarios y permisos

Para crear tablas dentro de esta base de datos utilizaremos las funciones almacenadas que se crearon gracias a los comandos anteriores, y que son parte del paquete postgis.

Para crear una tabla de puntos primero creamos una tabla estándar utilizando cualquier gestor de base de datos y luego a la misma le asociamos una nueva columna del tipo geográfica.

```
Select AddGeometryColumn ('public','puntos','the_geom',22185,'POINT',2);
```

Asumimos en este comando que contamos con un esquema del tipo public, que el DATUM será 22185, la capa será del tipo POINT, y será el mismo de dos dimensiones "2D".

```
text AddGeometryColumn(varchar schema_name,
                        varchar table_name,
```

```
varchar column_name,  
integer srid,  
varchar type,  
integer dimension,  
boolean use_typmod=true);
```

SUBVERSION

Para realizar el versionado se utiliza el servidor de versionado “subversión” el cual es nativo en Debian, de no estar ya instalado utilizar el comando

```
# apt-get install subversion
```

Luego debe ser configurado para permitir los accesos vía apache y los medios para brindar seguridad en su operación, para esto se deberá seguir los pasos detallados a continuación.

Primero y para hacer accesible subversión a través de HTTP es necesario instalar en Apache un módulo específico que permite la integración entre los mismos, este se denomina apache-svn

```
apt-get install libapache2-svn
```

Luego se debe seleccionar un lugar para colocar el repositorio. Para este proyecto se utilizó la ruta:

```
mkdir /srv/svn
```

Se ingresa a la misma tipeando.

```
cd /srv/svn
```

A partir de este punto se comienza con la creación y configuración del repositorio

Se debe crear un directorio para alojar el mismo

```
mkdir nombre_del_repositorio
```

Asociamos el versionado a dicho directorio

```
svnadmin create /srv/svn/nombre_del_repositorio
```

Se ajustan los permisos para habilitar a Apache a realizar solicitudes y cambios sobre estos directorios.

```
chown -R www-data nombre_del_repositorio  
chmod -R g+rws nombre_del_repositorio
```

Luego realizamos el agregado de usuarios y passwords para el ingreso por webDAV⁹

Habilitamos primero las librerías

```
a2enmod dav  
a2enmod dav_fs
```

Reiniciamos apache

```
service apache2 restart
```

Configuramos ahora por lo menos un usuario para poder acceder al repositorio

```
htpasswd -c /etc/apache2/webdav.password nombreDeUsuario
```

El comando del sistema solicitará se ingrese el password y su confirmación asociados al usuario en creación.

Ahora podemos indicarle a Apache finalmente cómo utilizar el repositorio y cómo validar los usuarios que desean ingresar al mismo, para esto debemos agregar en la ruta.

```
/etc/apache2/sites-enabled/000-default
```

⁹ **Web Distributed Authoring and Versioning** (<http://www.webdav.org/>)

Ajustando según sea necesario los nombres y rutas agregamos al final del archivo y antes del tag `</VirtualHost>` lo siguiente.

```
<Location /srv/svn/map_viewer>
    DAV svn
    SVNPath /srv/svn/map_viewer
    AuthType Basic
    AuthName "Subversion Repository Map_Viewer"
    AuthUserFile /etc/apache2/webdav.password
    Require valid-user
</Location>
```

Reiniciar apache

```
service apache2 restart
```

Para ingresar al repositorio podremos utilizar cualquier herramienta que sea compatible con el estándar SVN.

Para poder mantener una copia de trabajo y una de producción, se creó el repositorio en el directorio que se indicó con anterioridad

```
/srv/svn/nombre_del_repositorio
```

Y luego se realizó un checkout del mismo desde el directorio `/var/www` del apache corriendo sobre Debian.

Y otro checkout en el equipo host que se utilizó para realizar la programación y testeo de la aplicación

De esta forma una vez editada la aplicación y verificada su estabilidad y funcionalidad en el host de edición se realiza un commit, y luego de finalizado un update en el directorio de Apache para mantener la versión de producción sincronizada con el estado de avance de la aplicación.

APACHE TOMCAT

Para poner en funcionamiento este servidor de Java Servlet y JavaServer Pages technologies, deberemos descargar la versión última del sitio oficial

<http://tomcat.apache.org/download-80.cgi>

Para mayor comodidad descargamos la versión en formato .ZIP, y la copiamos en alguna ruta de nuestro servidor que dispongamos para emplazar

los servidores y algunos otros elementos que serán necesarios para la ejecución del server, en este caso se seleccionó la ruta.

```
/srv/apache-tomcat-8.0.8
```

Esto es todo lo que debemos hacer ya que viene pre configurado para trabajar en modo localhost que es lo que precisamos, luego se pueden realizar configuraciones de seguridad extra pero no hacen al alcance de este proyecto.

INSTALACIÓN GEOSERVER

Para desplegar e instalar el servidor de mapas Geoserver deberemos primero que nada descargar el mismo del sitio oficial <http://geoserver.org/release/stable/>

De este sitio bajaremos la versión compilada y empaquetada, que se encuentra bajo el apartado “*Web Archive*”.

Para desplegar el mismo solo deberemos descomprimir el archivo .zip que se descarga del sitio en el directorio del servidor Apache Tomcat, en nuestro caso bajo la ruta

```
/srv/apache-tomcat-8.0.8/webapps
```

Una vez descomprimido debemos configurar algunos archivos para que todo funcione correctamente.

La primer configuración y la más importante, es indicarle a Tomcat donde está la máquina virtual JAVA en nuestro equipo, para esto descargamos de la página oficial de Oracle la versión jre-7.0 en su formato “*tar.gz*”

Descomprimos este paquete en el mismo directorio de Apache Tomcat para nuestra comodidad, quedando:

```
/srv/jre1.7.0
```

Editamos entonces el archivo

```
/srv/apache-tomcat-8.0.8/bin/catalina.sh
```

En el mismo agregamos lo siguiente luego de los comentarios y como primeras líneas ejecutables del archivo (siempre asumimos las rutas de archivos que describimos):

```
JRE_HOME=/srv/jre1.7.0
export JRE_HOME;
```

Para operar nuestro servidor ejecutamos los comandos

Para iniciar Tomcat

```
/srv/apache-tomcat-8.0.8/bin/startup.sh
```

(Deberíamos obtener una salida similar a la siguiente)

```
Using CATALINA_BASE: /srv/apache-tomcat-8.0.8
Using CATALINA_HOME: /srv/apache-tomcat-8.0.8
Using CATALINA_TMPDIR: /srv/apache-tomcat-8.0.8/temp
Using JRE_HOME: /srv/jre1.7.0
Using CLASSPATH:
/srv/apache-tomcat-8.0.8/bin/bootstrap.jar:/srv/apache-tomcat-8.0.8/bin/tom
cat-juli.jar
Tomcat started.
```

Para detener Tomcat

```
/srv/apache-tomcat-8.0.8/bin/shutdown.sh
```

Una vez arrancado el servidor verificamos que funcione nuestra aplicación ingresando desde algún navegador a la dirección web

<http://ip.de.mi.server:8080/geoserver/>

Deberíamos visualizar el back office de geoserver!

Para poder utilizarlo desde el navegador y con nuestra aplicación y operar con el protocolo WFS-T agregaremos en el archivo proxy.conf lo siguiente.

```
ProxyRequests On
ProxyPreserveHost On
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>
ProxyPass /geoserver http://localhost:8080/geoserver
ProxyPassReverse /geoserver http://localhost:8080/geoserver
```

Si este archivo no existe le pediremos a apache que lo habilite ejecutando

```
a2enmod proxy
a2enmod proxy_connect
a2enmod proxy_http
```

De esta manera todas las peticiones dirigidas a /geoserver serán enrutadas internamente a localhost:8080/geoserver lo que permitirá.

Ahora agregaremos un archivo extra que permitirá procesar los requerimientos WFS y WFS-T debido a que se producirá un error causado por una característica de seguridad del comando XMLHttpRequest que está diseñado para procesar solamente respuestas de solicitudes provenientes solamente desde dominios iguales al solicitante.

Como las solicitudes provienen ahora desde sitios remotos, e intentan ingresar a localhost se determina que el origen no coincide y se produce un error.

Una manera de subsanar esto es creando un archivo proxy.cgi que se consigue en la siguiente dirección web

```
http://trac.osgeo.org/openlayers/browser/trunk/openlayers/examples/proxy.c  
gi
```


A este archivo lo guardaremos en el directorio cgi-bin de nuestro servidor Apache.

```
/usr/lib/cgi-bin
```

NOTA: Asegurarse de salvar el archivo con permisos de ejecución.

Ajustamos el archivo proxy.cgi configurando como sigue

```
allowedHosts = ['192.168.0.101', '192.168.0.101:80']
```

De esta manera permitiremos que solo nuestro servidor pueda acceder a este proxy y nadie más distinto de nosotros.

Para habilitar nuestro visualizador a trabajar con este proxy le indicaremos a OpenLayers que lo haga, especificando el siguiente comando en las primeras líneas de nuestro visualizador.

```
OpenLayers.ProxyHost = '/cgi-bin/proxy.cgi?url=';
```

Otra opción es mediante la definición de alias en el servidor colocar tanto el aplicativo web como el servidor de mapas bajo el mismo dominio.

INSTALACIÓN SSH

Ejecutar el comando en consola

```
apt-get install ssh
```

INSTALACIÓN DE LA APLICACIÓN

La aplicación se ejecuta directamente desde el directorio www de nuestro servidor Apache principal solo deberemos habilitar su acceso y tener instalado php5.

En caso de no tenerlo ejecutamos el comando

```
apt-get install php5
```

Si esta todo correcto e instalado como se especificó con anterioridad copiando el directorio de la aplicación en la ruta

`/var/www/`

En esta instancia debería funcionar todo correctamente, lo cual podremos verificar ingresando a

`http://ip.de.mi.server/map_viewer/`

DESARROLLO DE LA SOLUCIÓN

Metodologías de desarrollo

DESARROLLO PROTOTIPADO EVOLUTIVO

PRIMER PROTOTIPO

El primer prototipo fue el más básico y simple de todos ya que en este solo se presentó una máquina virtual configurada y funcionando donde se presentaron las herramientas seleccionadas instaladas y funcionando, en su configuración más básica para comenzar el proceso de desarrollo, se definieron lineamientos para el próximo prototipo y los alcances del mismo.

Este incluía Geoserver instalado y funcionando, con acceso a la consola de administración (Figura 002).



Figura 002

Postgres-PostGis instalado y funcionando con acceso remoto y algunas capas de mapa precargadas (Figura 003).

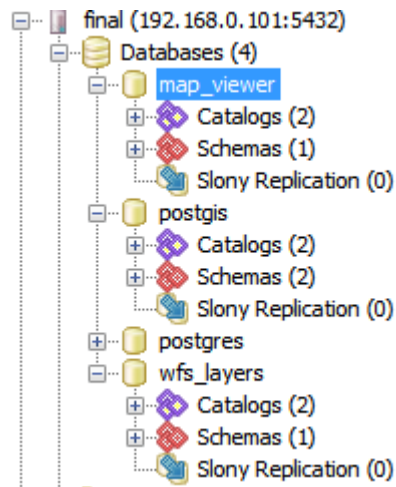


Figura 003

map_viewer: es la base de datos que gestiona y administra las propiedades del editor

postgis: es un ejemplo con capas importadas de IPEC que contiene un gran número de capas como por ejemplo ejes, manzanas radios censales, para poder realizar pruebas.

wfs_layers: son capas dedicadas para el editor, configuradas para testeo de funcionalidades.

Servidor SVN configurado y funcionando con acceso remoto y la raíz para el proyecto map_viewer generada.

Para el siguiente prototipo se definió la necesidad de generar un visualizador que permitiera la visualización de los diferentes tipos de capas necesarias para el resto del proyecto.

SEGUNDO PROTOTIPO

En este prototipo se entregó un visualizador básico (Figura 004) el cual contaba con una estructura de capas en formato de árbol en su lateral izquierdo y un cuerpo principal que contenía el mapa que se seleccionaba desde el árbol de capas. Esto implicó la integración de todas las herramientas necesarias para el resto del proyecto, como Openlayers para la visualización de los tipos de capa WMS y WFS en sus formatos más simples y en modo de visualización solamente, PostGIS para el almacenamiento y recuperación de las capas vectoriales para el formato WFS, y, finalmente, Geoserver para la gestión y publicación de los servicios y capas disponibles.

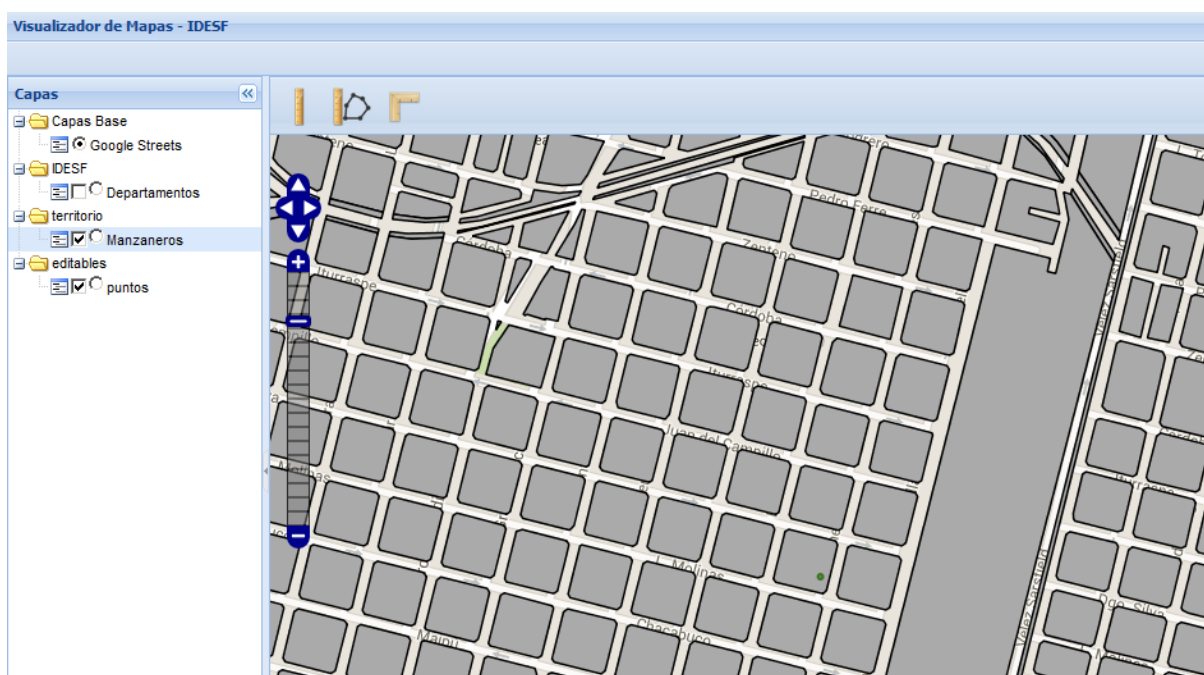


Figura 004

Para el siguiente prototipo se requirió el agregado de seguridad para el acceso al visualizador, discriminado por usuario y la capacidad de asignación de capas específicas para los mismos ya sea bajo el protocolo WMS o WFS.

TERCER PROTOTIPO

Este prototipo cuenta con una pantalla de autenticación inicial (Figura 005) donde el usuario que desea realizar gestiones sobre capas debe ingresar sus credenciales, esto implicó la creación de nuevas estructuras en base de datos para almacenar la información específica de cada usuario, sus capas asociadas, y los formatos con los cuales puede visualizar las mismas.

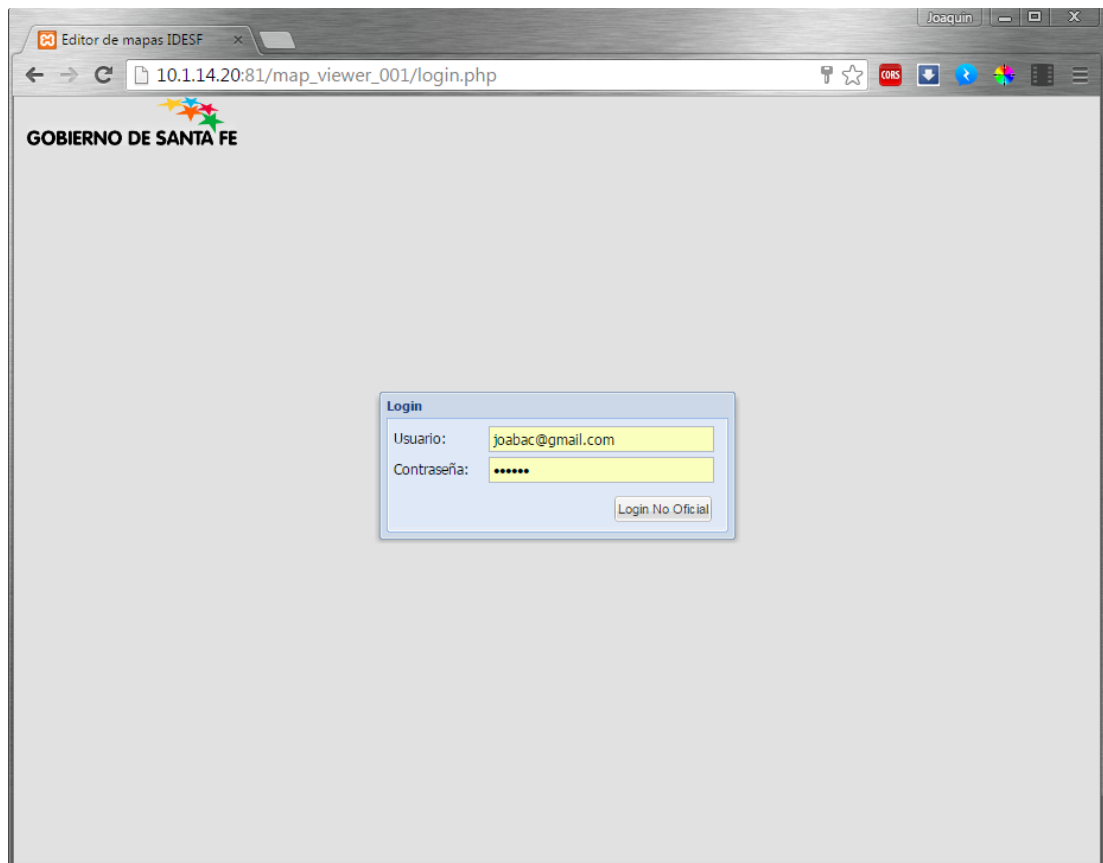


Figura 005

Estas credenciales ingresadas son validadas contra el servidor, y si todo es correcto el servidor retorna el visualizador configurado según las propiedades de cada usuario, esto quiere decir que para diferentes usuarios podemos tener diferentes configuraciones de capas, ya sea en número como en formato.

Por ejemplo para dos usuarios diferentes, podemos elegir que una capa se habilite para uno de ellos en formato WFS para edición, y para el otro solo en formato de visualización en WMS.

Usuario 1: (Figura 006)

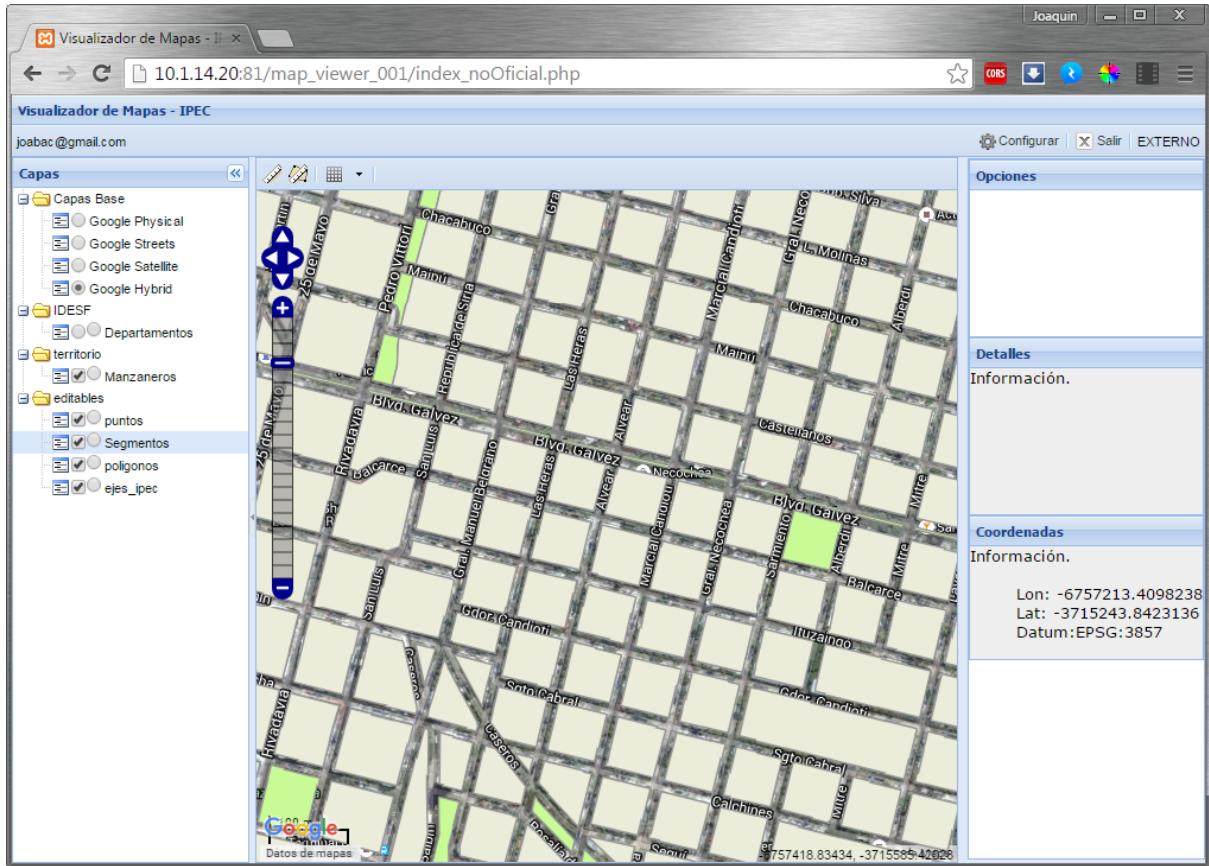


Figura 006



Usuario 2: (Figura 007)

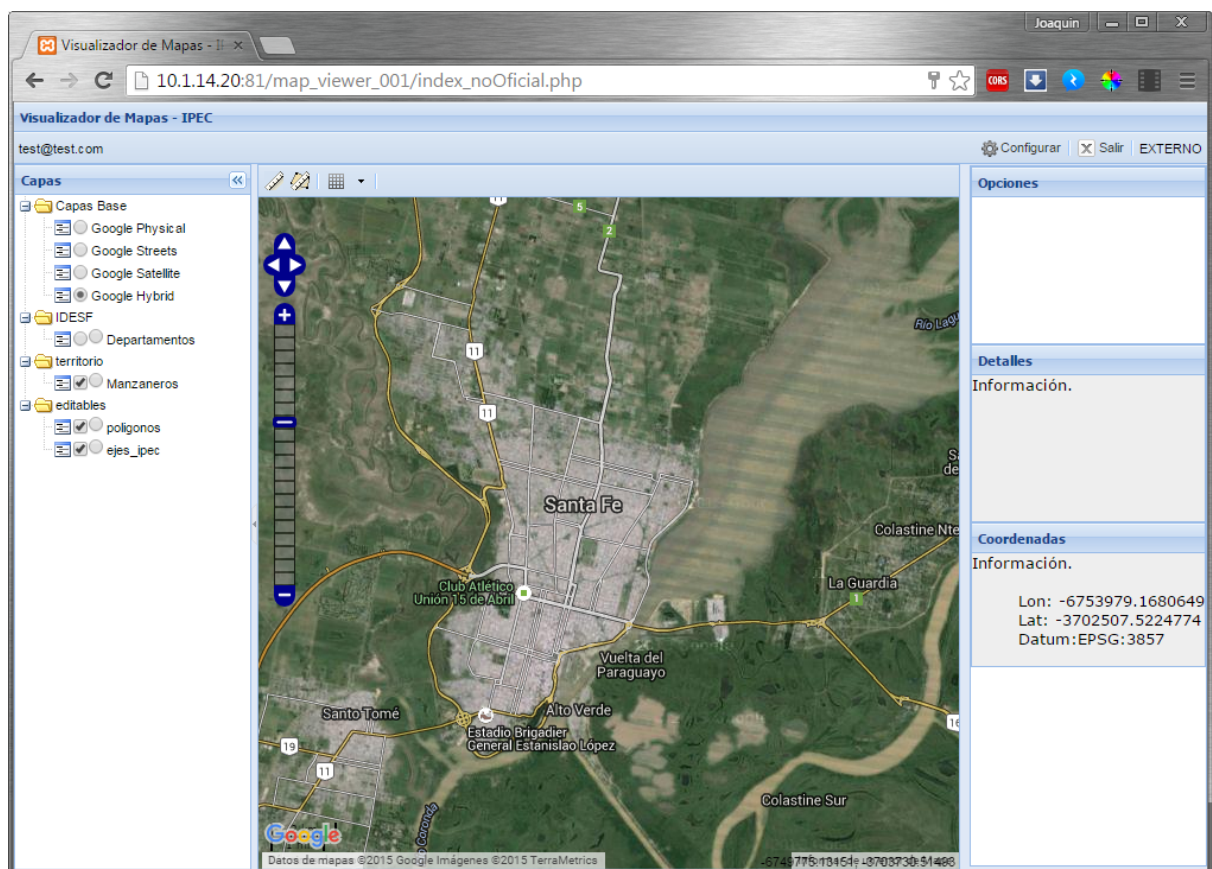


Figura 007

CUARTO PROTOTIPO

En este prototipo se presentó la capacidad de edición de las capas vectoriales WFS mediante el uso del protocolo WFS-T.

Esta funcionalidad de edición permite para cada tipo de capa vectorial, (POLIGONO, LINEA, PUNTO) tanto la inserción como la edición y remoción de los mismos, esta característica se habilita de a una capa por vez, asemejando la forma de trabajo que se utiliza en gvSIG o QGis *(realizando un click derecho sobre una capa del listado de capas y seleccionando el modo de edición)* (Figura 008)

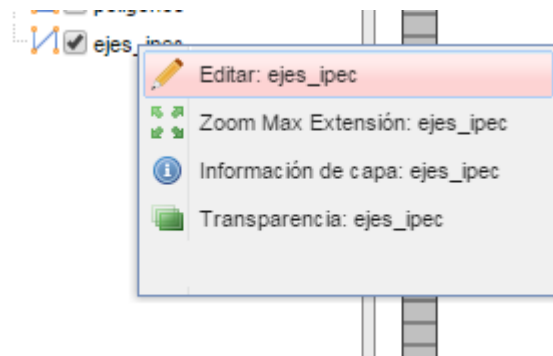


Figura 008

Edición de Puntos: (Figura 009)

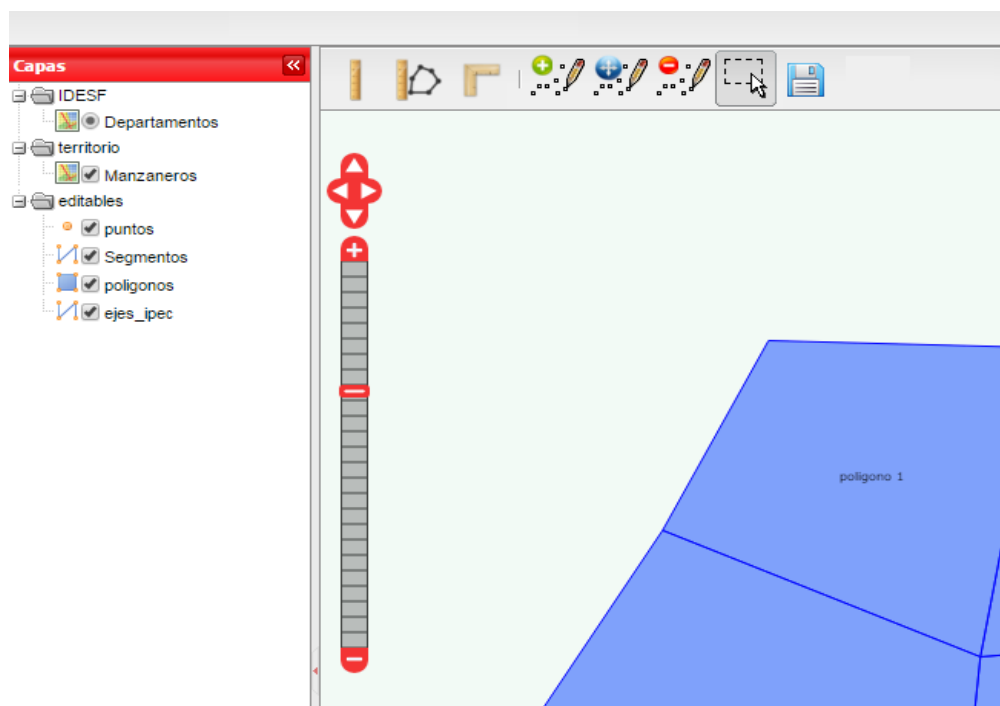


Figura 009

Edición de Segmentos: (Figura 010)

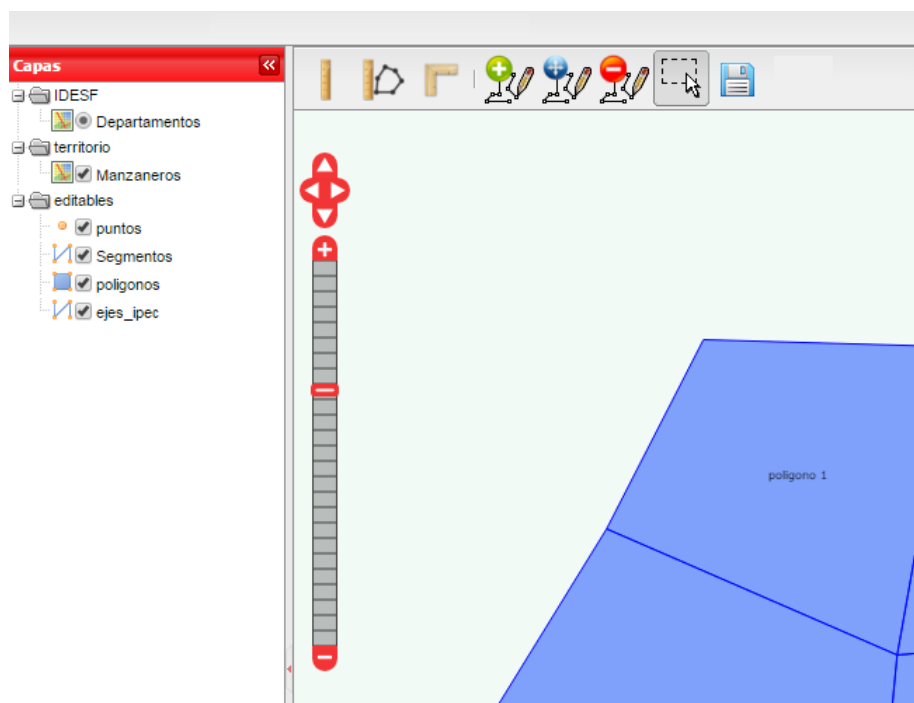


Figura 010

Edición de Polígonos: (Figura 011)

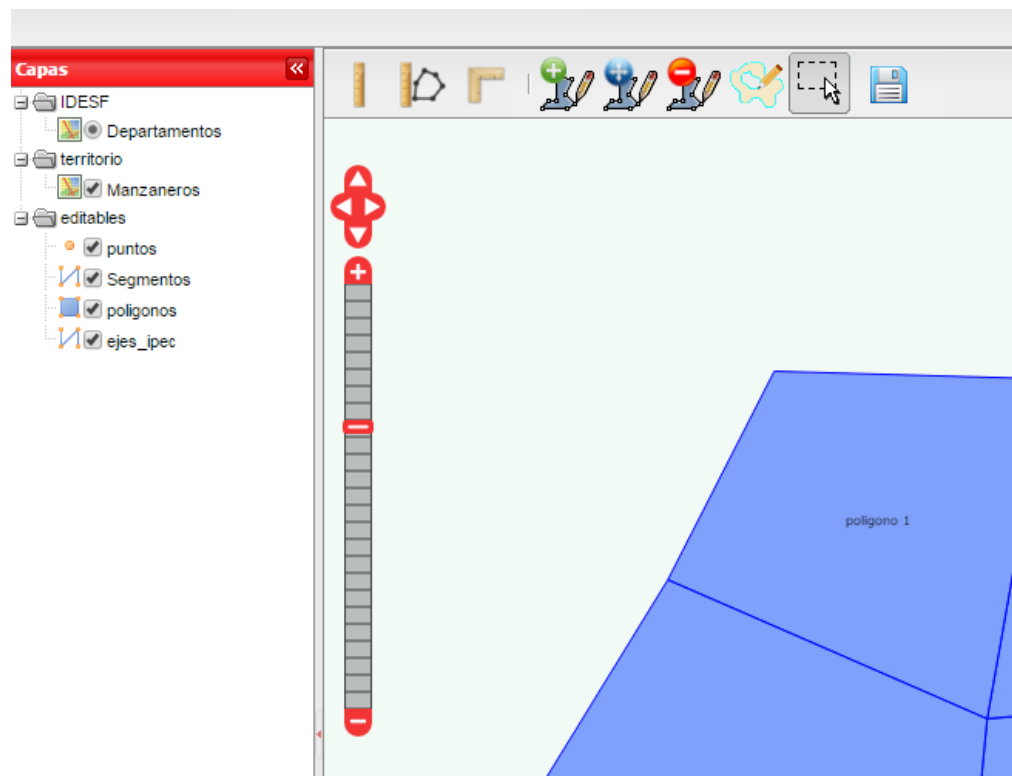


Figura 011

Esta capacidad , hizo que surja la necesidad de diferenciar lógicamente el tipo de capa que se está editando, para lo cual se debieron realizar cambios en el código que permiten la diferenciación o identificación del tipo de capa lógicamente en tiempo de ejecución.

Se planteó para un próximo prototipo la posibilidad voluntaria o automática de exportar las capas lo que genera un requerimiento nuevo a ser presentado en el nuevo prototipo.

Funcionalidad restante

Se debería mejorar el estilo de capas WFS y proporcionar un gestor de estilos local para cada usuario más acabado y refinado y que brinde la posibilidad de persistir los estilos por usuario y por capa según sea necesario.

QUINTO PROTOTIPO

Se presenta en este prototipo el editor de estilos solicitado y algunas funcionalidades adicionales que facilitan la edición de las capas independientemente del tipo de capa.

Estas son:

- Control Snap (magnetismo a entidades geográficas, segmentos y puntos)(Figura 012)

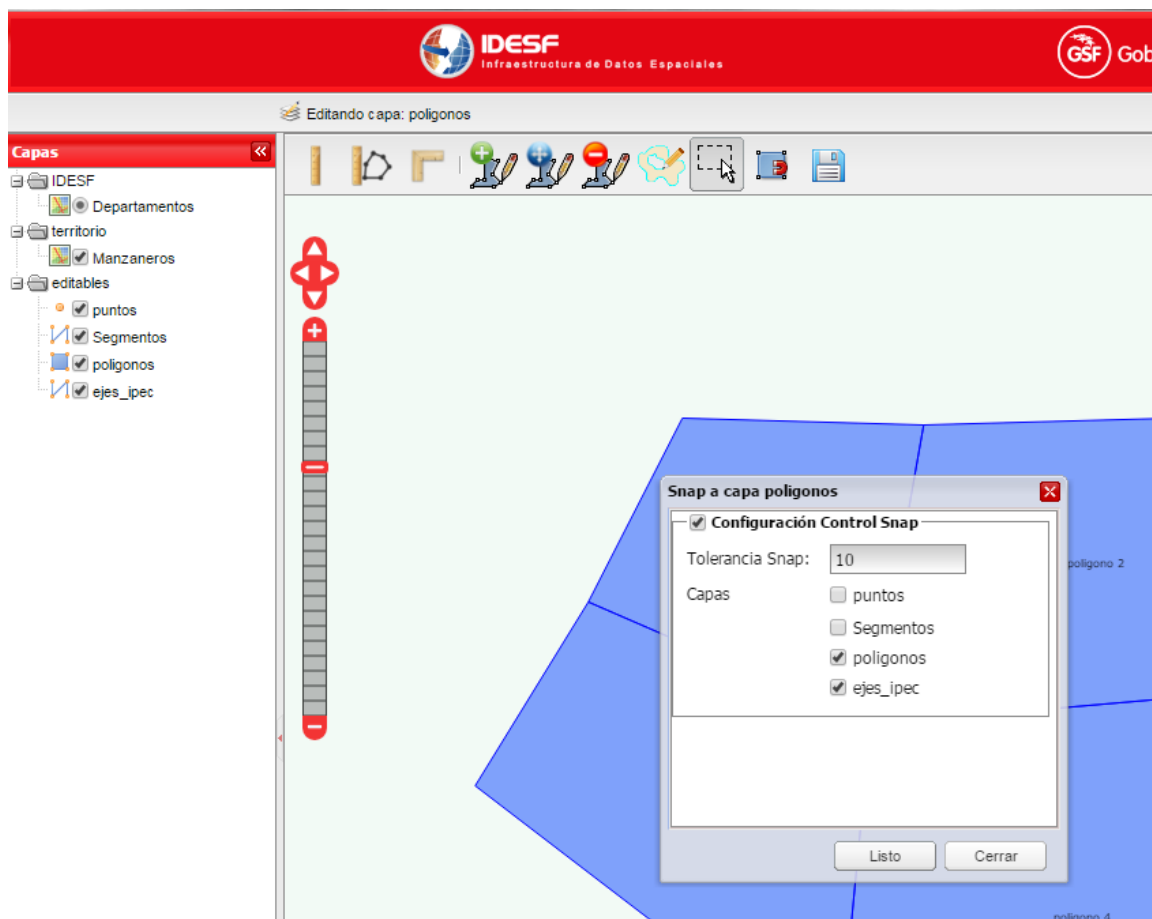


Figura 012

Esta configuración permite, que al editar cualquier tipo de geometría, las mismas se adhieran a los nodos o segmentos de las capas que se elijan como si estos fuesen magnéticos, la ventaja de esta configuración es que de esta manera podemos asegurar el cumplimiento de restricciones de topología, por ejemplo que dos lados de un polígono estén en contacto, o que dos segmentos nazcan de un mismo nodo.

- Estilos ajustados a los formatos provinciales (Figura 013).

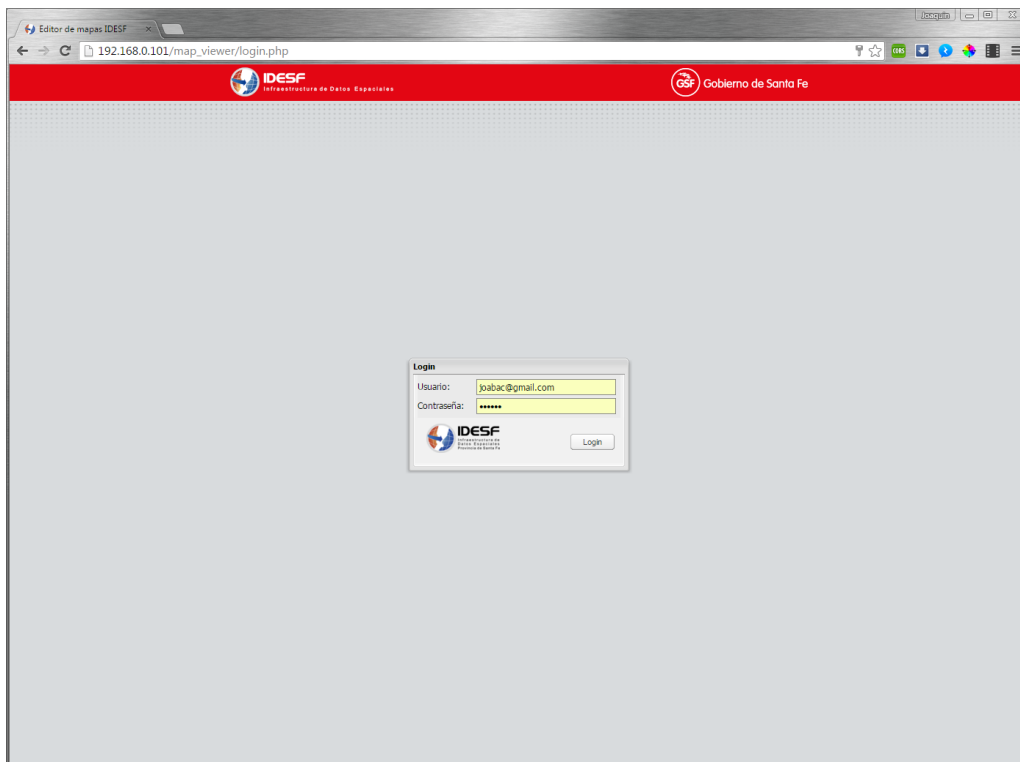


Figura 013

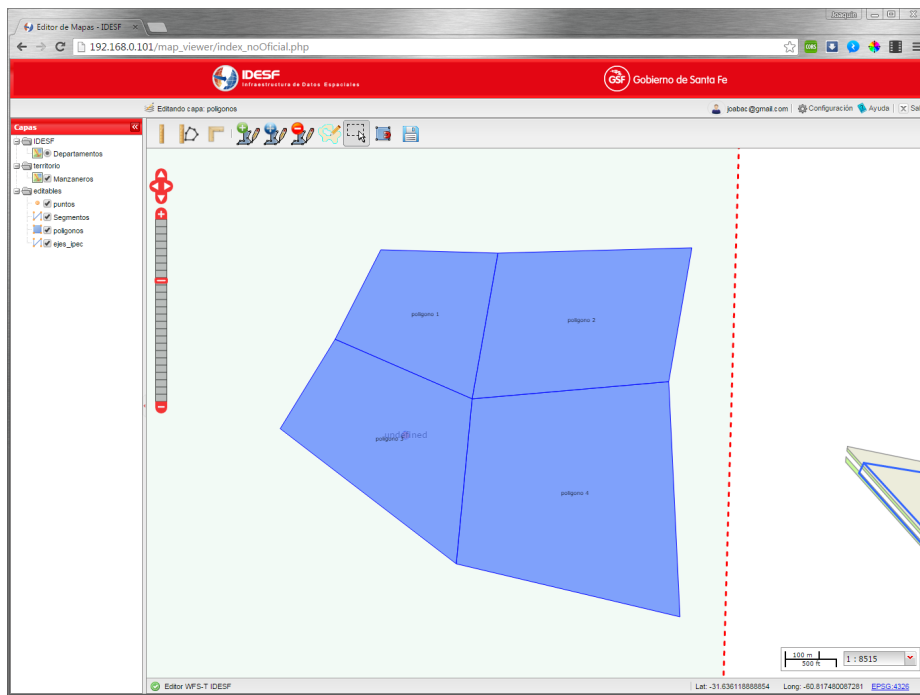


Figura 014

Se agregaron logos nuevos y ajustados a los lineamientos provinciales, y se ajustaron los colores para respetar el formato provincial (Figura 014)

- Edición del perfil de usuario (Figura 015)

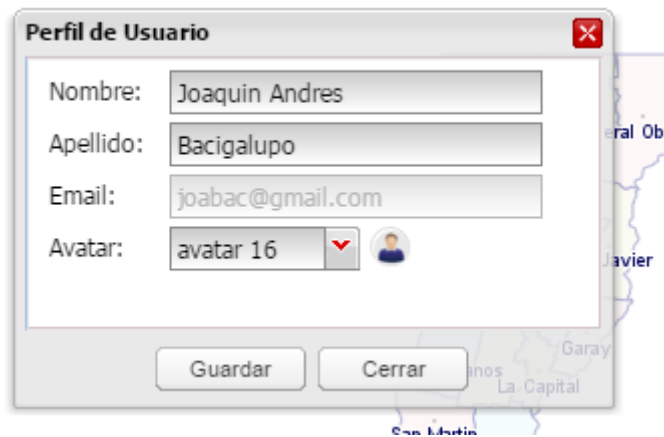


Figura 015

Se permite realizar una edición básica sobre los atributos del perfil de usuario, de todas formas el alta física del usuario se realiza desde base de

datos. Se permite además seleccionar un icono avatar para personalizar aún más el estilo de la interfaz.

- Edición de estilos por capa y por usuario (Figura 016)

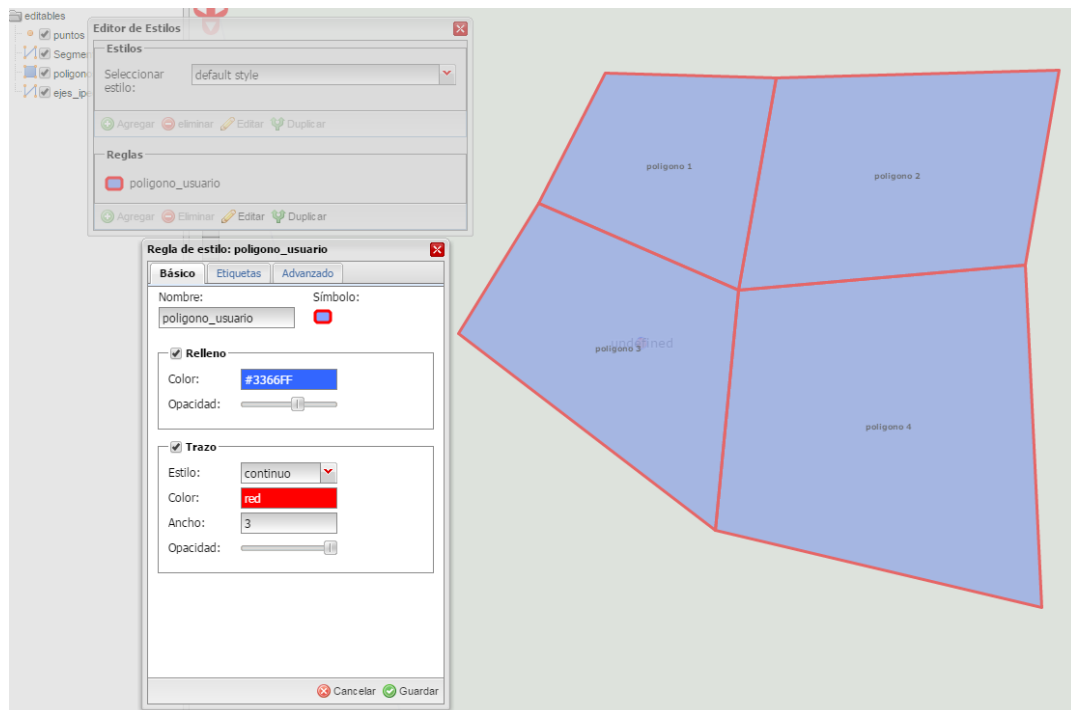


Figura 016

Se permite realizar ajustes, sobre el estilo básico de la geometría, como por ejemplo color de relleno y transparencia o color de trazo, su opacidad y su grosor Figura 017, Figura 018 y Figura 019.

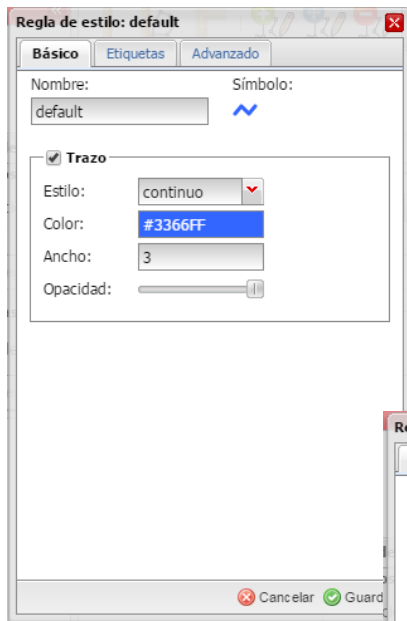


Figura 017



Figura 018

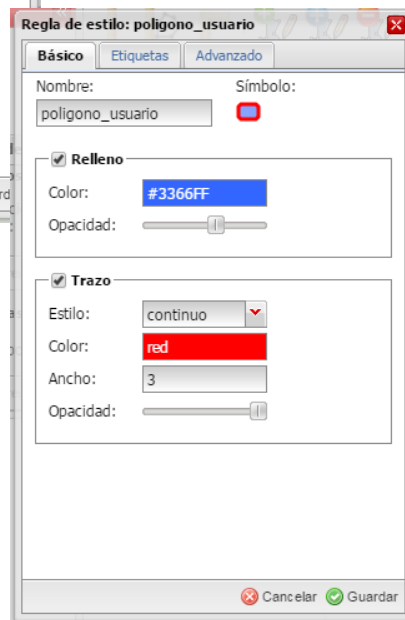


Figura 019

Se puede, además, ajustar las propiedades relacionadas con la visualización de las etiquetas de las geometrías (Figura 020)

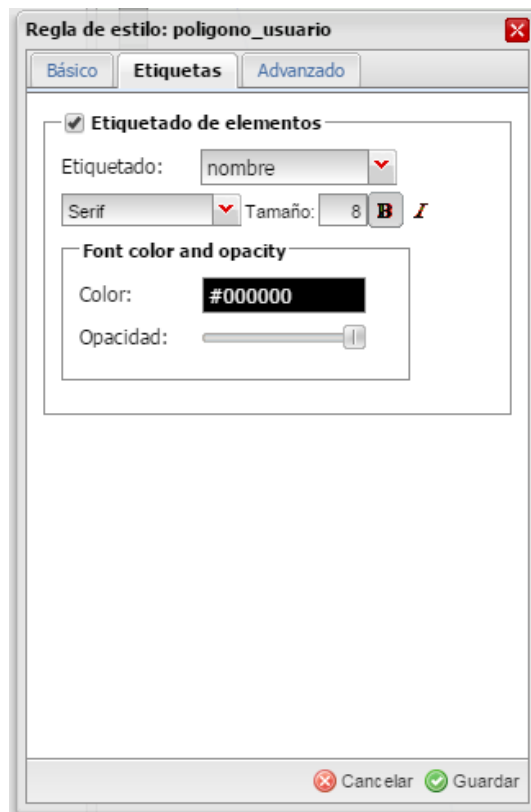


Figura 020

Estas configuraciones se persisten por usuario y por capa, por lo cual cada usuario podrá definir de forma independiente la visualización que se ajuste mejor a sus necesidades, y esta configuración estará disponible la próxima vez que cargue su sesión.

- Límites de edición por usuario y por capa

Cada usuario contará con una región habilitada para edición y no se permite editar fuera de ella o por sobre regiones no habilitadas

La región habilitada se presenta con un color verde claro y un nivel de transparencia que no entorpece la visualización de las demás capas. (Figura 021)

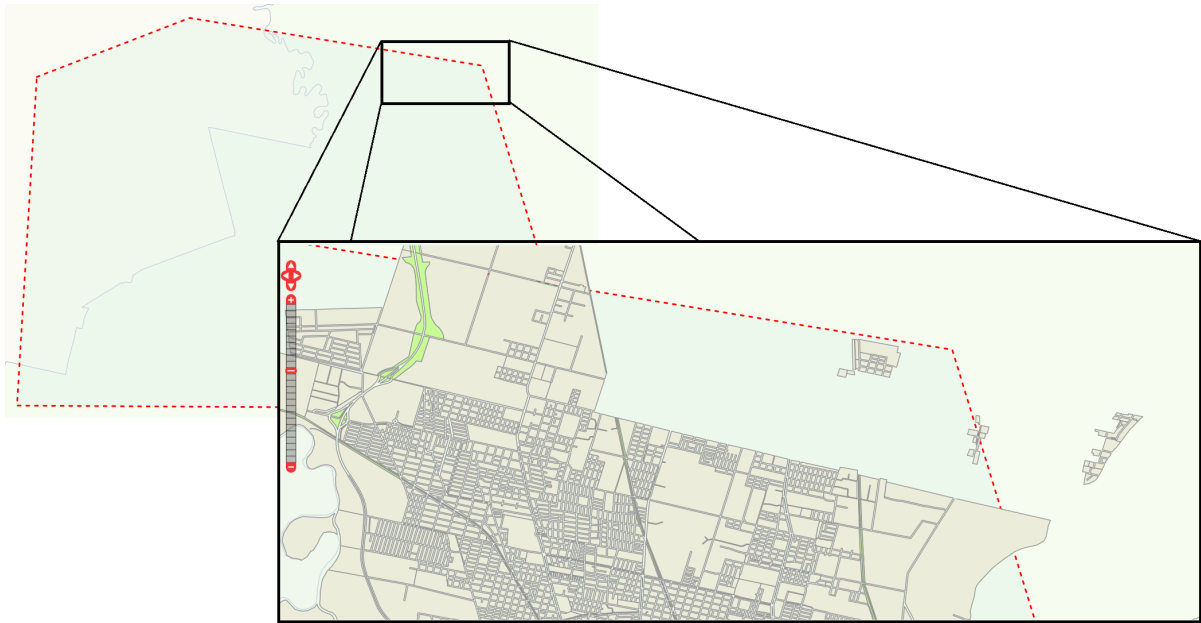


Figura 021

En la siguiente imagen se presentan dos casos que ejemplifican la funcionalidad de la región habilitada (Figura 022)

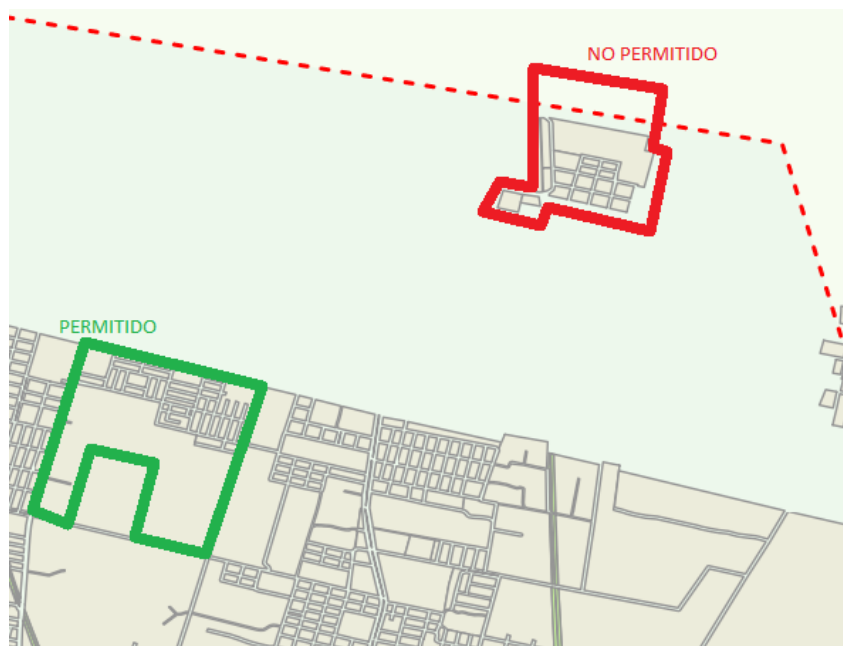


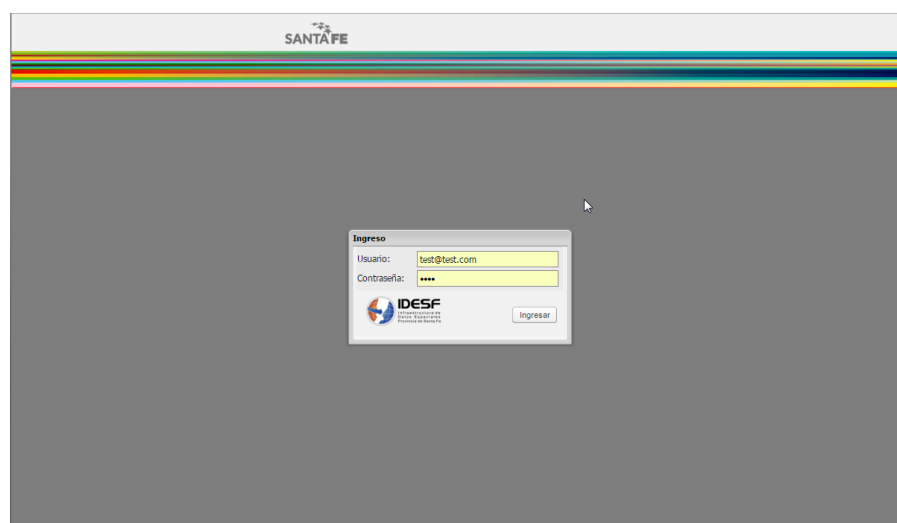
Figura 022

SEXTO PROTOTIPO

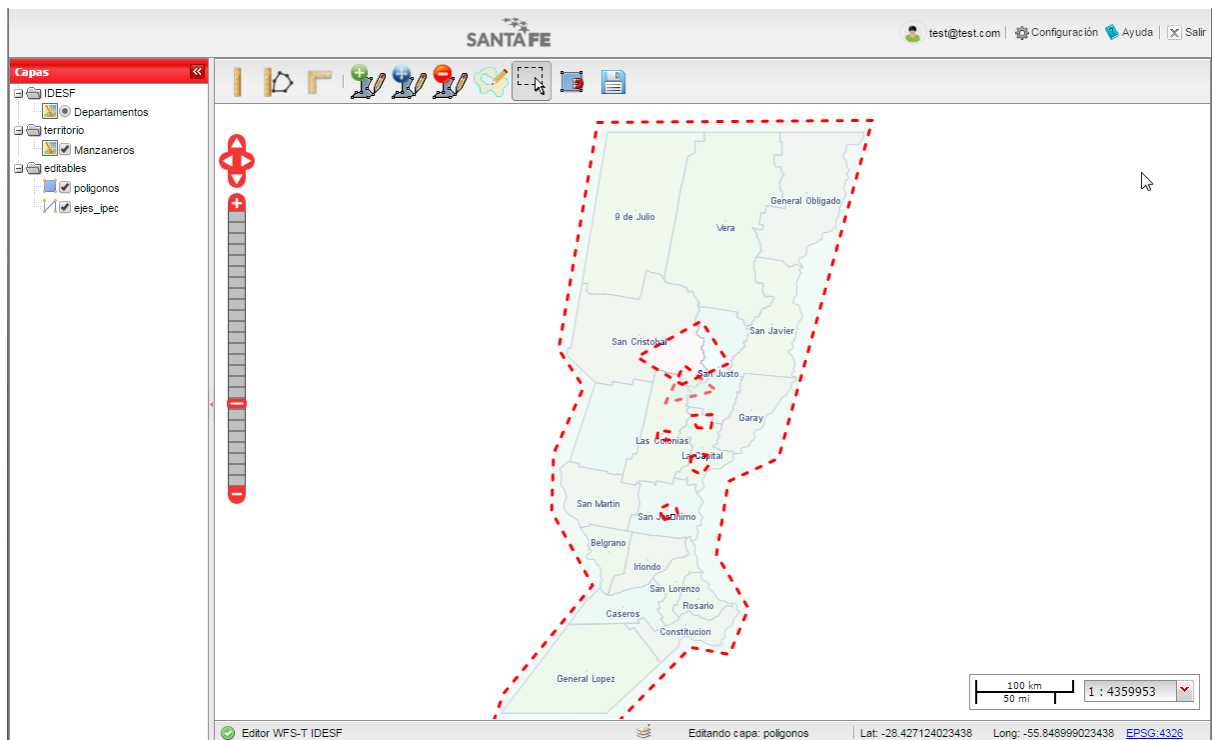
Ante un cambio de dirigencia política se debió ajustar los estilos nuevamente para permitir el prototipo se ajuste al nuevo lineamiento de estilos provincial.

Se agregaron además características de filtrado WFS por capa. y se mejoraron cuestiones de configuración, relativas a archivos de conexión y configuración para los diferentes entornos de la infraestructura.

Pantalla de autenticación:



El panel principal se modificó para permitir una mejor distribución del contenido y volver genérico el editor para cualquier productor.



Funcionalidad restante

Generar herramientas adicionales configurables por usuario para diferentes tipos de geometrías, generar configuración parametrizable por base de datos de filtros y escalas.

Se deberá definir una capa adicional de software que permita implementar el protocolo SOAP entre el código PHP actual de acceso a datos y el lado cliente.-

Descripción de los Procesos Automáticos

ÁRBOL DE CAPAS POR USUARIO

Se desarrolló un método utilizando recursión para satisfacer esta necesidad, el árbol de capas está codificado en cada capa en un atributo de base de datos respetando el formato

```
root.<nombreCapa|directorio>[.<nombreCapa|directorio>]
```

Al inicializar la aplicación web una vez logueado el usuario, el servidor inicializa las variables pertenecientes al mismo, y del lado cliente se procesan las capas habilitadas generando el árbol de nodos y ramas parseando para cada una la estructura explicada.

Esto nos permite definir de manera independiente para cada usuario las capas que puede utilizar.

Los elementos utilizados para manipular los nodos son todas estructuras definidas en la librería GeoExt y ExtJS:

❑ Ext.tree.TreeNode

Nodo básico de árbol para estructuras Ext.tree

<https://docs.sencha.com/extjs/3.4.0/#!/api/Ext.tree.TreeNode>
de

❑ GeoExt.tree.LayerNode

Es una subclase de Ext.tree.TreeNode , la diferencia radica en que este nodo está conectado a una capa de Openlayers.Layer, lo cual nos habilita la manipulación directa de la misma desde el nodo del árbol.

❑ GeoExt.data.LayerStore

Clase que permite almacenar y agrupar capas de Openlayers.

❏ GeoExt.tree.LayerContainer

Subclase de Ext.tree.AsyncTreeNode, esta clase colecta toda la información de capas que tienen su propiedad de displayInLayerSwitcher seteada en true.

PERSISTENCIA DE DATOS GEOGRÁFICOS

La persistencia se realiza utilizando una clase definida en OpenLayers que básicamente su objetivo es gestionar la persistencia de los elementos editados, el nombre del mismo es SaveStrategy

```
new OpenLayers.Strategy.Save()
```

La misma puede ser parametrizada para realizar la persistencia de forma automática o de forma manual únicamente.

para esto deberemos configurar los parámetros:

“Si el respaldo es solo manual”

```
saveStrategy.auto = false
```

“El respaldo se realizará automáticamente cada int (ms)”

```
saveStrategy.auto = int (ms)
```

ELECCIONES DE DESARROLLO

Tree panel - geoExt

Se eligió generar por código el árbol ya que es necesario agrupar las capas por nodos y subnodos, lo cual no se puede realizar directamente con geoext. Para ese fin se utilizaron las estructuras treeNode, de geoext, anidándolas, y realizando búsquedas recursivas para el anidado de nuevas capas como hojas de dichos nodos.

para el anidado directo se permiten sólo 2 niveles y las capas deben estar declaradas como grupos.

```
// create a group layer (with several layers in the "layers" param)
// to show how the LayerParamLoader works
new OpenLayers.Layer.WMS("Tasmania (Group Layer)",
    "http://demo.opengeo.org/geoserver/wms", {
    layers: [
        "topp:tasmania_state_boundaries",
        "topp:tasmania_water_bodies",
        "topp:tasmania_cities",
        "topp:tasmania_roads"
    ],
    transparent: true,
    format: "image/gif"
    }, {
    isBaseLayer: false,
    buffer: 0,
    // exclude this layer from layer container nodes
    displayInLayerSwitcher: false,
    visibility: false
    }
```

Luego se le solicita a geoext que levante todas las capas del listado de capas, lo cual no es correcto para este proyecto

LECCIONES APRENDIDAS

Se precisaba una estructura más compleja de anidamiento y definición de niveles de capas, que además debe ser completamente dinámico y no puede ser hardcodeado

Agregado de librería GXP para reutilización de código (OpenGeo SDK Client API Reference) funcionalidades de edición, estilado.

Lección importante se deben definir correctamente los atributos mediante el uso del parámetro schema al crear las capas vector, ya que este atributo es quien describe la capa en detalle y es utilizado por las herramientas de edición de entidades.

Selección de tipo de capa

Se realiza mediante el método javascript definido en el cuerpo del código principal. La metodología que se utiliza realiza un GET solicitando un **DescribeFeatureType** contra Geoserver indicando, además, la respuesta en formato json, de esta manera nos permite analizar su contenido en forma rápida y sin más transformaciones que un simple parseo del string recuperado.

```
var describeFeatureTypeRequest = OpenLayers.Request.GET({ url:
capa.protocol.url+'?SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&outputFormat=application/json&TYPENAM
ME='+capa.protocol.featureType,
  async: false
});
```

y se analiza el atributo específico que describe el tipo de geometría.

```
capa_json = JSON.parse(describeFeatureTypeRequest.responseText);
var tipo_geometria = getValues(getObjects(capa_json, 'name', 'the_geom'), 'localType');
```

luego se devuelve el tipo de geometría

```
switch (tipo_geometria[0])
{
  case 'Point':
    iconoCapa = "gxp-icon-symbolgrid-point";
    break;
  case 'LineString':
```

```

        iconoCapa = "gxp-icon-symbolgrid-line";
        break;
    case 'Polygon':
        iconoCapa = "gxp-icon-symbolgrid-polygon";
        break;
    default :
        iconoCapa = "gxp-tree-rasterlayer-icon";
        break;
}

```

Manejo de estilos¹⁰:

La recuperación del estilo en cada capa, se realiza copiando los objetos que contienen los atributos específicos de estilo, de la variable “styles” definida en cada capas.

El estilo por defecto *se define por reglas*.

Recuperamos el estilo de la capa específica a persistir usando

```

var estilo_default_feature_json =
JSON.stringify(map.getLayersByName(nombreCapa)[0].styleMap.styles.default.rules[0].symbolizer);

```

Para restaurar el mismo se debe forzar el estilo recuperado y setear el mismo en tiempo de carga de la capa y definirlo en dos objetos que permitirán luego reutilizar los componentes de estilos, gxp

1 - Primero recuperamos el estilo desde base de datos:

```

function getEstilos(id_layer)
{
    var estiloCapa = "";
    var id_capa_estilo = id_layer;
    var parametros = {Parametro:'getEstilo',id_capa:id_capa_estilo,user_name:userName};

    $.ajax({
        type: "POST",
        async: false,
        url: "scripts/utiles.php",
        dataType: "json",
        success: function(msg) {
            if(msg.length > 0)
            {
                estiloCapa = msg[0].estilo;
            }
        }
    });
}

```

¹⁰ <http://www.peterrobins.co.uk/it/olstyle.html>
http://docs.openlayers.org/library/feature_styling.html

```

    }
  },
  error: function() { //error de transacción

    mensajeError("Error", "Error al recuperar el estilo");
  },
  data: parametros
});

if(estiloCapa != "")
{
  var estilo_json = JSON.parse(estiloCapa);

  return estilo_json;
}
else
{
  return "";
}

```

2 - Luego debemos inicializar las capas específicas con los estilos por defecto definiendo el estilo base y las reglas que se aplicarán.

else{//capas WFS

```

var estilo_grafico = objetoCapa.estilo_wfs;
var estilo_json = getEstilos(objetoCapa.id_capa);
var estilo_inicial;
var symbolizer;
var symbolizerText;
var sldConfig;
var sld;
var simbolizer_temporal;

if(estilo_json != null){

  if(typeof estilo_json.Point != 'undefined')
  {
    symbolizer = new OpenLayers.Symbolizer.Point(estilo_json.Point);

    estilo_inicial = {
      strokeColor : estilo_json.Point.strokeColor,
      strokeOpacity : estilo_json.Point.strokeOpacity,
      strokeWidth : estilo_json.Point.strokeWidth,
      strokeLinecap : estilo_json.Point.strokeLinecap,
      strokeDashstyle : estilo_json.Point.strokeDashstyle,
      fillColor:estilo_json.Point.fillColor,
      fillOpacity: estilo_json.Point.fillOpacity ,
      pointRadius:estilo_json.Point.pointRadius,
      graphicName:estilo_json.Point.graphicName,
      rotation:estilo_json.Point.rotation,

      //se puede agregar la opcion del grafico
    };

    //strokeColor      {String} Color for line stroke.
    //strokeOpacity     {Number} Stroke opacity (0-1).
    //strokeWidth       {Number} Pixel stroke width.
    //strokeLinecap     {String} Stroke cap type ("butt", "round", or "square").
    //strokeDashstyle   {String} Stroke dash style according to the SLD spec.
    //fillColor         {String} RGB hex fill color (e.g.
    //fillOpacity       {Number} Fill opacity (0-1).
    //pointRadius       {Number} Pixel point radius.
    //externalGraphic   {String} Url to an external graphic that will be used for
    rendering points.

    //graphicWidth     {Number} Pixel width for sizing an external graphic.
    //graphicHeight     {Number} Pixel height for sizing an external graphic.
    //graphicOpacity    {Number} Opacity (0-1) for an external graphic.

```

```

//graphicXOffset    {Number} Pixel offset along the positive x axis for
displacing an external graphic.
//graphicYOffset    {Number} Pixel offset along the positive y axis for
displacing an external graphic.
//rotation    {Number} The rotation of a graphic in the clockwise direction about
its center point (or any point off center as specified by graphicXOffset and graphicYOffset).
//graphicName

    simbolizer_temporal = {"Point":symbolizer,"Text":{}};
}

if(typeof estilo_json.Line != 'undefined')
{
    symbolizer = new OpenLayers.Symbolizer.Line( estilo_json.Line);
    estilo_inicial = {
        strokeColor : estilo_json.Line.strokeColor,
        strokeOpacity : estilo_json.Line.strokeOpacity,
        strokeWidth : estilo_json.Line.strokeWidth,
        strokeLinecap : estilo_json.Line.strokeLinecap,
        strokeDashstyle : estilo_json.Line.strokeDashstyle

    };
    //      strokeColor {String} Color for line stroke.
    //      strokeOpacity    {Number} Stroke opacity (0-1).
    //      strokeWidth      {Number} Pixel stroke width.
    //      strokeLinecap    {String} Stroke cap type ("butt", "round", or "square").
    //      strokeDashstyle

    simbolizer_temporal = {"Line":symbolizer,"Text":{}};
}

if(typeof estilo_json.Polygon != 'undefined')
{
    symbolizer = new OpenLayers.Symbolizer.Polygon(estilo_json.Polygon);
    estilo_inicial = {
        strokeColor : estilo_json.Polygon.strokeColor,
        strokeOpacity : estilo_json.Polygon.strokeOpacity,
        strokeWidth : estilo_json.Polygon.strokeWidth,
        strokeLinecap : estilo_json.Polygon.strokeLinecap,
        strokeDashstyle : estilo_json.Polygon.strokeDashstyle,
        fillColor : estilo_json.Polygon.fillColor,
        fillOpacity : estilo_json.Polygon.fillOpacity
    };
    //      strokeColor {String} Color for line stroke.
    //      strokeOpacity    {Number} Stroke opacity (0-1).
    //      strokeWidth      {Number} Pixel stroke width.
    //      strokeLinecap    {String} Stroke cap type ("butt", "round", or "square").
    //      strokeDashstyle  {String} Stroke dash style according to the SLD spec.
    //      fillColor      {String} RGB hex fill color (e.g.
    //      fillOpacity

    simbolizer_temporal = {"Polygon":symbolizer,"Text":{}};
}

if(typeof estilo_json.Text != 'undefined')
{
    symbolizerText = new OpenLayers.Symbolizer.Text(estilo_json.Text);

    estilo_inicial.fontColor = estilo_json.Text.fontColor;
    estilo_inicial.fontOpacity = estilo_json.Text.fontOpacity;
    estilo_inicial.fontSize = estilo_json.Text.fontSize;
    estilo_inicial.label = estilo_json.Text.label;
    estilo_inicial.fontFamily = estilo_json.Text.fontFamily;
    estilo_inicial.zIndex = estilo_json.Text.zIndex;
}

```

```
        simbolizer_temporal.Text = symbolizerText;
    }
    else
    {
        if(typeof estilo_json.Line != 'undefined' || typeof estilo_json.Point != 'undefined' || typeof
estilo_json.Polygon != 'undefined'){
            simbolizer_temporal.Text = new OpenLayers.Symbolizer.Text({CLASS_NAME:
"OpenLayers.Symbolizer.Text",graphicName: "circle",rotation: 0,vendorOptions: {},zIndex: 0 });
        }
    }
}

var temporary_style = new OpenLayers.Style( {
    cursor: "inherit",
    fillColor: "#F5D223",
    fillOpacity: 0.4,
    fontColor: "#000000",
    hoverFillColor: "#ffffff",
    hoverFillOpacity: 0.8,
    hoverPointRadius: 1,
    hoverPointUnit: "%",
    hoverStrokeColor: "#ff0000",
    hoverStrokeOpacity: 1,
    hoverStrokeWidth: 0.2,
    label:"",
    labelAlign: "cm",
    labelOutlineColor: "#ffffff",
    labelOutlineWidth: 3,
    pointRadius: 6,
    pointerEvents: "visiblePainted",
    strokeColor: "#FF7700",
    strokeDashstyle: "solid",
    strokeLinecap: "round",
    strokeOpacity: 1,
    strokeWidth: 2
});
var selected_style = new OpenLayers.Style({
    cursor: "pointer",
    fillColor: "#0033ff",
    fillOpacity: 0.4,
    fontColor: "#000000",
    hoverFillColor: "#ffffff",
    hoverFillOpacity: 0.8,
    hoverPointRadius: 1,
    hoverPointUnit: "%",
    hoverStrokeColor: "#ff0000",
    hoverStrokeOpacity: 1,
    hoverStrokeWidth: 0.2,
    label:"",
    labelAlign: "cm",
    labelOutlineColor: "#ffffff",
    labelOutlineWidth: 3,
    pointRadius: 6,
    pointerEvents: "visiblePainted",
    strokeColor: "#003377",
    strokeDashstyle: "solid",
    strokeLinecap: "round",
    strokeOpacity: 1,
    strokeWidth: 2
});
```

```

var estiloMapa = new OpenLayers.StyleMap({
  'default': new OpenLayers.Style(
    estilo_inicial,
    {
      defaultsPerSymbolizer: false,
      description: "estilo por defecto",
      isDefault: true,
      layerName: objetoCapa.nombre_capa,
      name: "default",
      rules: [
        new OpenLayers.Rule({
          title : estilo_json.title,
          context: null,
          description: null,
          elseFilter: false,
          filter: null,
          maxScaleDenominator: null,
          minScaleDenominator: null,
          name: null,
          symbolizer: simbolizer_temporal
        })
      ]
    }
  ),
  'temporary':temporary_style,
  'select': selected_style
});

var schemaURI = objetoCapa.url +
'/DescribeFeatureType?version='+objetoCapa.version_wfs+'&typename='+objetoCapa.nombre_capa;
var layer = new OpenLayers.Layer.Vector(objetoCapa.nombre_fantasia,
{
  protocol: new OpenLayers.Protocol.WFS({
    version: objetoCapa.version_wfs,
    srsName: datumWfs ,
    url: objetoCapa.url,
    featureType: objetoCapa.nombre_capa,
    featureNS:'http://192.168.0.101/geoserver/idesf',
    //TODO: agregar atributo URI en gestion de capas
    geometryName: "the_geom",
    renderers: ["SVG",'Canvas'],
    schema: schemaURI
  }),
  isBaseLayer: false,
  visibility: objetoCapa.visibility,
  displayInLayerSwitcher: false,
  strategies: [new OpenLayers.Strategy.BBOX()],
  styleMap: estiloMapa
});

capas.push(layer);

```

3 - Definimos la nueva capa con el estilo por defecto y las reglas asociadas para reutilizar luego con gxp

```

var estiloMapa = new OpenLayers.StyleMap({
  'default': new OpenLayers.Style(
    estilo_inicial,
    {
      defaultsPerSymbolizer: false,
      description: "estilo por defecto",

```

```

        isDefault: true,
        layerName: objetoCapa.nombre_capa,
        name: "default",
        rules: [
            new OpenLayers.Rule({
                title : estilo_json.title,
                context: null,
                description: null,
                elseFilter: false,
                filter: null,
                maxScaleDenominator: null,
                minScaleDenominator: null,
                name: null,
                symbolizer: simbolizer_temporal
            })
        ]
    },
    'temporary': temporary_style,
    'select': selected_style
});

```

```

var schemaURI = objetoCapa.url +
'/DescribeFeatureType?version='+objetoCapa.version_wfs+'&typename='+objetoCapa.nombre_capa;
var layer = new OpenLayers.Layer.Vector(objetoCapa.nombre_fantasia,
{
    protocol: new OpenLayers.Protocol.WFS({
        version: objetoCapa.version_wfs,
        srsName: datumWfs ,
        url: objetoCapa.url,
        featureType: objetoCapa.nombre_capa,
        featureNS:'http://192.168.0.101/geoserver/idesf',
        //TODO: agregar atributo URI en gestion de capas
        geometryName: "the_geom",
        renderers: ['SVG','Canvas'],
        schema: schemaURI
    }),
    isBaseLayer: false,
    visibility: objetoCapa.visibility,
    displayInLayerSwitcher: false,
    strategies: [new OpenLayers.Strategy.BBOX()],
    styleMap: estiloMapa
});

```

En el bloque resaltado se observa la definición del estilo con la variable `estilo_inicial` que renderiza los elementos en la carga inicial, y los `symbolizers` asociados que permitirán generar las reglas que serán utilizadas luego con `gxp` como, así también, el nombre que el usuario le asignó a la regla.

En los casos para los cuales la capa no posee estilo definido en base de datos se dejará el estilo por defecto de `openlayers`.

NOTA: El dato más importante a tener en cuenta es el uso de los objetos temporales el cual según el tipo de estilo recuperado se formará para poder luego incorporarlo al estilo según corresponda.

```
simbolizer_temporal = {"Point":symbolizer,"Text":{}};  
simbolizer_temporal = {"Line":symbolizer,"Text":{}};  
simbolizer_temporal = {"Polygon":symbolizer,"Text":{}};
```

Estilo para modificación de entidades

Se debe definir en el control openlayers el estilo que se desea utilizar mediante la variable `vertexRenderIntent`

Exportar capas vectoriales a shapes

La metodología utilizada reutiliza un recurso de Geoserver que permite generar un archivo zip que luego puede ser importado en cualquier software que soporte formato ESRI (Shapefiles).

La metodología utilizada exige que solo se pueda exportar datos que están persistidos en el servidor.

```
http://192.168.0.101:8080/geoserver/IDESF/ows?service=WFS&version=1.0.0  
&request=GetFeature&typeName=IDESF:'+layer.name+'&outputFormat=SHAPE  
-ZIP
```

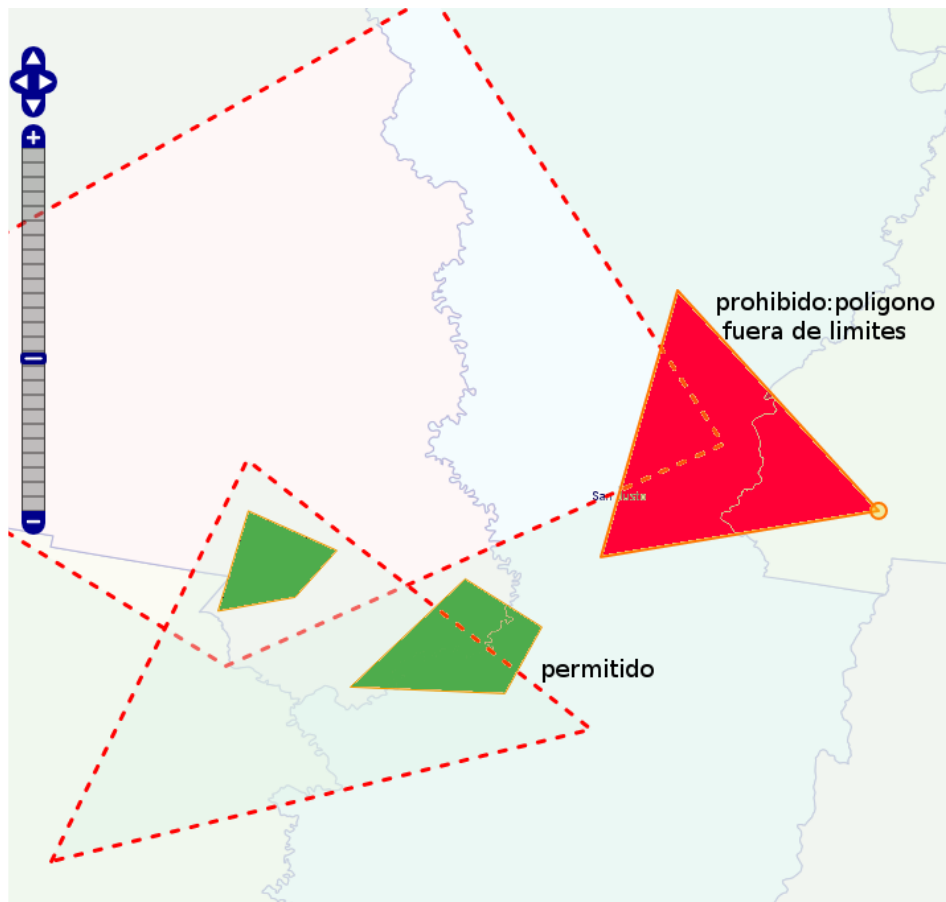
Control de región habilitada para dibujar

Se definirá una región habilitada para cada usuario utilizando un polígono, el cual será definido por un usuario administrador, y cargará el mismo en la base de datos en formato geoJSON y con Datum EPSG:4326.

El polígono que se define sera por capa y por usuario y será cargado cada vez que se desee editar una capa, el mismo no se mostrará en el listado de capas sólo será visible sobre el mapa y tendrá un estilo predefinido (contorno rojo de trazos, relleno verde claro con una opacidad que no entorpece la visualización de las capas de base).

La metodología utilizada será validar, la pertenencia de los puntos polígonos o líneas en dicha región, si no está contenido el polígono, punto o línea, en la región será informado y no se permitirá su persistencia, se validará

también la intersección con los límites de la región para asegurar que esté la geometría contenida realmente dentro del límite y evita que trazos o trayectos intersecten los límites, como se muestra en la imagen.



Para controlar la pertenencia se utiliza un método implementado en openlayers llamado intersects, y para controlar si la geometría está contenida en el polígono límite se decidió utilizar una librería para manejo de topología llamada JSTS (Javascript Topology Suite

<http://bjornhartell.github.io/jsts/doc/api/symbols/jsts.geom.Geometry.html> .

```

function testea_limite(geometria)
{
  var points = geometria.geometry.getVertices();
  var contenido = new Array();
  var contiene = new Array();
  var total_puntos = points.length;

  var testeo = false;

  $(layer_limite.features).each(function (idx, feature)
  {
    contenido[idx] = 0;
    var suma_parcial = 0;

    $(points).each(function (idx, punto){
      contains = feature.geometry.containsPoint(punto);
      if(contains === true )
      {
        suma_parcial += 1//si no esta estrictamente contenido en algun poligono limite
      }
    });

    contenido[idx] = suma_parcial;

    var parser = new jsts.io.OpenLayersParser();
    contiene[idx] = parser.read(feature.geometry).contains( parser.read(geometria.geometry));

  });

  $(contenido).each(function (idx, feature_test)
  {
    if(feature_test === total_puntos && contiene[idx] === true)
      //todos los puntos incluido y ninguna interseccion con los limites
    {
      testeo = true;
    }
  });

  return testeo;
}

```

Para todos los tipos de geometría se aplica el mismo algoritmo.

Manejo de estados en WFS-T

El manejo del estado de las entidades geográficas es importante para los casos en que se desea manipular los mismos para editarlos, eliminarlos, o insertar nuevos elementos por código.

Cada Feature tiene asociado un estado en el objeto

`OpenLayers.Feature.Vector.state` donde para cada situación donde se edite actualice o inserte un feature se deberá indicar la operación que luego deberá realizar el protocolo WFS-t al procesar las capas.

Los estados asociados son :

```
OpenLayers.State.INSERT  
OpenLayers.State.UPDATE  
OpenLayers.State.DELETE  
OpenLayers.State.UNKNOWN
```

Un ejemplo de este uso es en el borrado de entidades geográficas:

```
var DeleteFeature = OpenLayers.Class(OpenLayers.Control, {  
  initialize: function(layer, options) {  
    OpenLayers.Control.prototype.initialize.apply(this, [options]);  
    this.layer = layer;  
    this.handler = new OpenLayers.Handler.Feature(  
      this, layer, {click: this.clickFeature}  
    );  
  },  
  clickFeature: function(feature) {  
    if(feature.fid == undefined) {  
      this.layer.destroyFeatures([feature]);  
    } else {  
      feature.state = OpenLayers.State.DELETE;  
      this.layer.events.triggerEvent("afterfeaturemodified",  
        {feature: feature});  
      feature.renderIntent = "select";  
      this.layer.drawFeature(feature);  
    }  
  },  
  setMap: function(map) {  
    this.handler.setMap(map);  
    OpenLayers.Control.prototype.setMap.apply(this, arguments);  
  }, CLASS_NAME: "OpenLayers.Control.DeleteFeature"});
```

La porción de código resaltada ejemplifica este uso, donde se marca el feature en estado DELETE, y luego se redibuja el feature dando lugar a que la misma se oculte y queda agendada para su eliminación.

FeatureNS

Esta variable de configuración de las capas WFS, merece una explicación aparte ya que es muy importante a la hora de visualizar o editar los elementos geográficos.

Esta variable hace referencia al Name Space¹¹ (espacio de nombres), para cada Workspace de Geoserver se genera un NS el cual contendrá los elementos geográficos, este parámetro se configura al crear el Espacio de trabajo en Geoserver.

Una vez creado el espacio de trabajo se puede verificar el NS generado ingresando a la siguiente URL.

<https://<URL o IP>/geoserver/rest/namespaces>

En el listado de namespaces mostrado, seleccionamos el de nuestro interés y allí veremos el URL que deberemos especificar en la configuración de la capa WFS.

Este parámetro se configura en la variable featureNS del objeto Openlayers “`OpenLayers.Protocol.WFS`”¹² ”.

```
protocol: new OpenLayers.Protocol.WFS({
  version: objetoCapa.version_wfs,
  srsName: datumWfs ,
  url: objetoCapa.url,
  featureType: objetoCapa.nombre_capa,
  featureNS: objetoCapa.featureNS,
                //'nombre exacto de geoserver para el name space',
  geometryName: "the_geom",
  schema: schemaURI
}),
```

¹¹ Un **espacio de nombres XML** es una recomendación **World Wide Web Consortium** para proporcionar elementos y atributos con nombre único en un archivo XML

¹² <http://dev.openlayers.org/docs/files/OpenLayers/Protocol/WFS-js.html>

Renderers¹³

Las dos opciones más difundidas son SVG y Canvas.

Canvas (Lienzo en inglés) es un elemento HTML incorporado en HTML5 que permite la generación de gráficos dinámicamente por medio del scripting. Permite generar gráficos estáticos y animaciones.

SVG (Scalable Vector Graphics) que es una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados.

Se seleccionó **SVG** ya que es el renderer por defecto utilizado por Openlayers y con el cual se obtuvo mejores resultados en los dos visualizadores para los cuales se optimizó el diseño Chrome y Firefox.

Para lograr un renderizado de las capas vector en todo el rango de zoom permitido deberemos especificar la siguiente instrucción en el código javascript

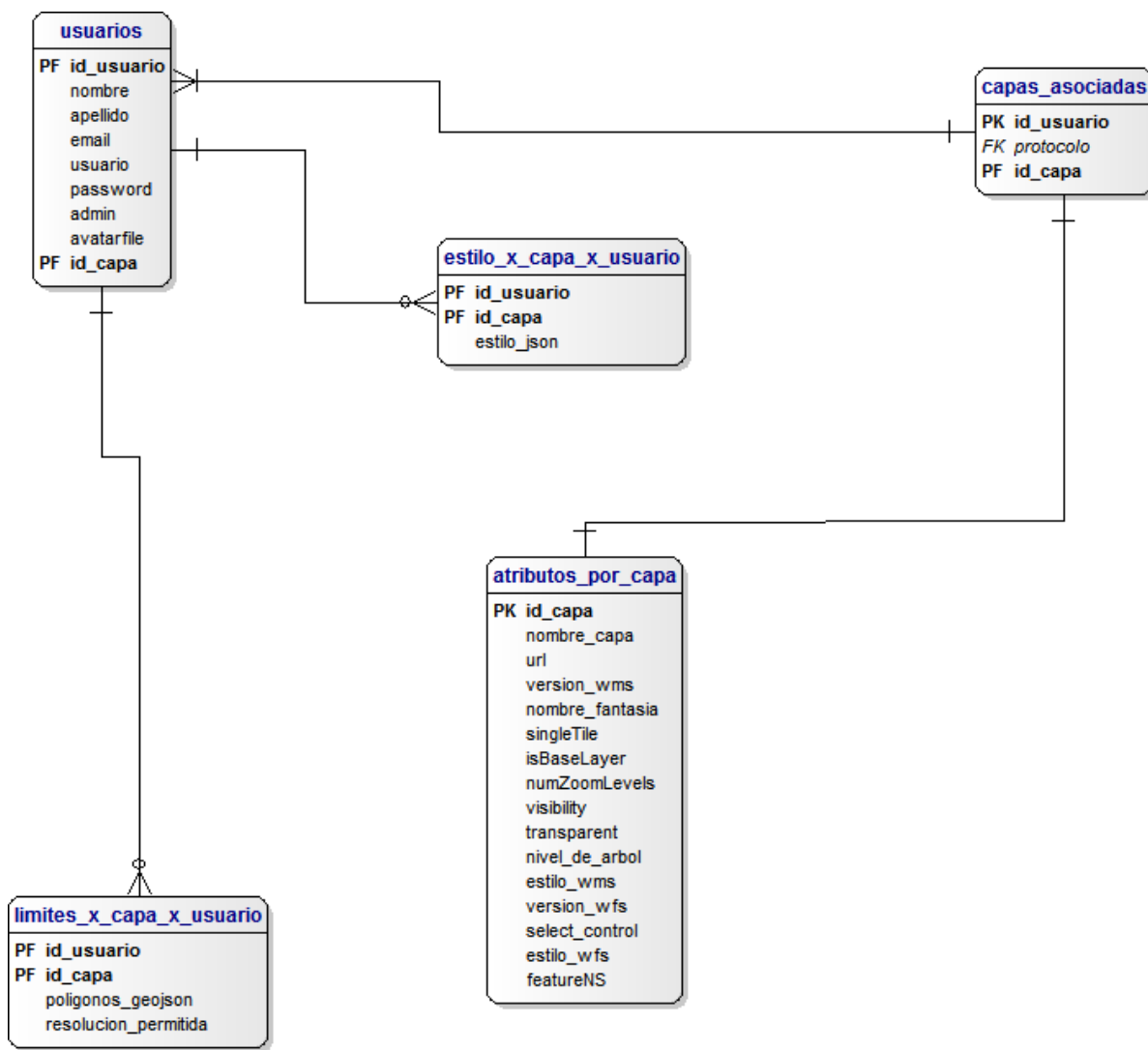
```
OpenLayers.Renderer.SVG.prototype.MAX_PIXEL = Number.MAX_VALUE
```

De este modo las capas vectoriales se mostrarán en todo el rango de escalas habilitadas para el editor.

¹³ Rendering es el proceso de generar imágenes partiendo de un modelo 2D o 3D

Base de Datos

En el presente esquema se observan las relaciones y vínculos entre las entidades básicas definidas



ATRIBUTOS_POR_CAPA

En esta tabla se definen los atributos específicos de cada capa entre ellos se pueden enumerar los más importantes

nombre_capa : Nombre físico de la capa como se publica en Geoserver

url : dirección WFS o WMS donde se encuentra Geoserver

nombre_fantasia: Nombre que deseamos mostrar para la capa

nivel_de_arbol : cadena de texto separada por puntos que define la jerarquía donde se posicionará la capa;

por ejemplo root.editables indica que la capa estará bajo la rama principal y dentro del nodo editables

featureNS: URL que define el nombre del esquema XML bajo el cual se encuentra la capa

CAPAS_ASOCIADAS

En esta tabla se define la relación entre la tabla usuarios, las capas que tendrá asociadas y el protocolo con que podrá visualizarlas.

ESTILO_X_CAPA_X_USUARIO

En esta tabla se define, para cada capa que tenga un usuario asociada, el estilo con el cual será cargada cada vez que este inicie sesión, dicho estilo puede ser cambiado desde el editor, y guardado para que esta configuración persista.

LIMITES_X_CAPA_X_USUARIO

En esta tabla se almacenan los límites definidos para cada usuario respecto de las capas que tenga asociadas.

USUARIOS

Tabla general de usuario, encontramos aquí datos de perfil y contraseñas

CONCLUSIÓN

La realización del proyecto fue una experiencia enriquecedora, tanto a nivel personal como profesional.

A nivel personal, me permitió ponerme en contacto con un sin número de profesionales de IDESF que me brindaron un apoyo incondicional para poder llevar adelante este proyecto, y principalmente a nivel laboral ya que hoy formo parte de este equipo, y trabajar de mi profesión es invaluable.-

Al comienzo del proyecto tanto para IDESF, quienes tenían la inquietud y necesidad sobre esta tecnología, como para mi, era un gran mar de interrogantes, poco a poco, y con cambios de versiones y tecnologías de por medio, se pudo ir avanzando y puliendo la aplicación que hoy ya se encuentra desplegada en entornos de desarrollo en la infraestructura provincial, proyectando su uso para proyectos actuales y futuros, lo cual me llena de orgullo y da valor a todo el esfuerzo puesto en esta tarea.

Respecto de las decisiones tomadas en cuanto a las tecnologías y productos seleccionados, algunas fueron inducidas por la infraestructura existente en la provincia, y sobre la cual hay que desplegar las aplicaciones, un ejemplo de esto son los sistemas operativos de base para los servidores, que cambiaron en el transcurso del desarrollo de este proyecto de Suse a Debian.

La arquitectura Cliente - Servidor seleccionada para el desarrollo del proyecto, fue desde mi punto de vista un acierto, que simplificó considerablemente la codificación.

Como así también la metodología de prototipado evolutivo, que si bien en algunos momentos exige la realización de reingenierías del código, brinda flexibilidad a la hora de adaptar la aplicación a nuevos requerimientos.

Lo más importante de todo el proceso es el conocimiento adquirido en las tecnologías específicas, Geoserver, WFS, WFS-T, WMS, WMTS, etc.

El protocolo WFS-T se ha vuelto hoy un estándar para la geocodificación, en los últimos años ha evolucionado mucho, tanto que ha sido incorporado en aplicaciones como Mapserver, que al comienzo de este proyecto, informaban de forma explícita en sus sitios web, que no tenían proyectos definidos que incluyeran WFS-T o intenciones de incluir el protocolo, hoy sin embargo ofrecen productos que agregan esta funcionalidad.

El punto más débil del protocolo estudiado es la **seguridad**, que queda delegada en los servidores de mapas o en las implementaciones específicas, y no se ofrece ninguna característica estandarizada que resuelva esta cuestión, es sin dudas el punto más importante a tener más en cuenta, y que puede sin dudas ser todo un tema a desarrollar o estudiar en un futuro proyecto de IDESF.

Otro punto importante aprendido y a considerar es el tamaño y la granularidad de los datos, que deberá ser analizada para cada caso con mucho detenimiento para evitar cuellos de botella en las redes y para evitar bloqueos en los navegadores, debido a que los archivos transferidos si no se manejan con cuidado pueden llegar a generar transferencias de hasta 6 Mb por petición.

La interfaz se diseñó para que sea intuitiva y minimalista, de modo que sea simple y de fácil uso, permitiendo además que el aprendizaje de la herramienta para los usuarios finales sea lo menos costoso posible.

Resumiendo y analizando todo lo investigado y aprendido en el transcurso de la realización de este proyecto, no me queda más que sumar cosas positivas, ver como crecía la aplicación, y como iba tomando forma, o el nivel de comprensión y entendimiento sobre el código de las librerías utilizadas, que permitió incluso realizar ajustes puntuales o adaptaciones en

algunas de ellas para que satisfagan requerimientos específicos del proyecto, como la oportunidad que se generó de concursar en la provincia y hoy formar parte del plantel laboral de la misma.

BIBLIOGRAFÍA

Bibliografía

Libros:

Tema	Recurso
GeoServer Beginner's Guide	Stefano Iacovella, Brian Youngblood February 2013 ISBN 139781849516686 Paperback350 pages
GeoServer Cookbook	Stefano Iacovella November 2014 ISBN 139781783289615 Paperback280 pages
Mastering GeoServer	Colin Henderson November 2014 ISBN 139781783287697 Paperback420 pages
OpenLayers Cookbook	Antonio Santiago Perez August 2012 ISBN 139781849517843 Paperback300 pages
OpenLayers 2.10 Beginner's Guide	Erik Hazzard March 2011 ISBN 139781849514125 Paperback372 pages
Learning Ext JS 3.2	Shea Frederick, Colin Ramsay, Steve 'Cutter' Blades, Nigel White October 2010 ISBN 139781849511209 Paperback432 pages
Arquitecturas de Software	Fowler, Martin. Patterns of Enterprise Application Architecture. 2002.
PostGIS Cookbook	Paolo Corti, Stephen Vincent Mather, Thomas J Kraft, Bborie Park January 2014 ISBN 139781849518666 Paperback484 pages
Fundamentos de bases de datos	Silberschatz, Abraham, Korth, Henry F.,Sudarshan, S. McGraw-Hill ISBN 84-481-3654-3

Elementos de cartografía	Robinson, Arthur H. [y otros] ISBN:84-282-0768-2
Larman, Craig	UML y patrones ISBN: 84-205-3438-2

Documentación Web:

Tema	Recurso
PHP	http://www.php.net/docs.php
Javascript - JQuery	http://librosweb.es/javascript/ http://api.jquery.com
SVN Tortoise	http://tortoisesvn.net/support.html
Apache Web Server	http://httpd.apache.org/docs/
PostgreSQL	http://www.postgresql.org/docs/
Postgis	http://postgis.net/stuff/postgis-1.5.pdf
GeoExt	http://geoext.org/docs.html
ExtJS Sencha 3.4.0	http://docs.sencha.com/extjs/3.4.0/#!/api
GXP	http://gxp.opengeo.org/master/doc/

ANEXOS

PREGUNTAS FRECUENTES

¿Cómo se representa la curvatura de la Tierra en mapas?

La respuesta a esta pregunta surge de enlazar los siguientes conceptos:

GEOIDE:

Es una forma ideal y teórica que tiene la tierra (cuerpo de forma casi esférica aunque con un ligero achatamiento en los polos). Corresponde a una superficie irregular.

Trabajar con esta superficie irregular genera dificultades matemáticas para poder ubicar las posiciones en la tierra sobre un mapa.

Se busca trabajar con un cuerpo geométrico que se parezca a la Tierra pero que sea regular.

ELIPSOIDE:

Es la figura geométrica más simple que se ajusta a la forma de la Tierra.

El elipsoide biaxial es una figura tridimensional generada por rotación de una elipse sobre su eje más corto. Este eje coincide aproximadamente con el eje de rotación de la Tierra. Los países han adoptado diferentes "elipsoides de referencia" que se ajustan a las características de cada lugar.

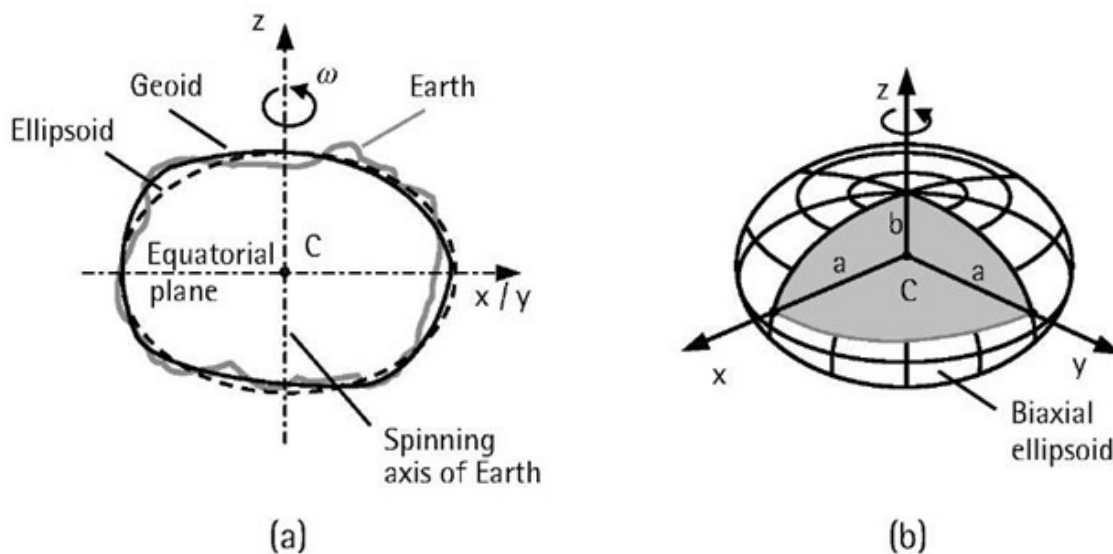


imagen: <http://www.albireotopografia.es/wp-content/uploads/2015/01/Topografia-geoide-y-elipsoide.jpg>

La verdadera forma de la tierra es un geoide, la cual no se define geoméricamente sino físicamente.

La figura geométrica que más se asemeja a la verdadera forma de la tierra es un elipsoide de revolución.

DATUM:

Cada país trata de que la superficie de su elipsoide coincida con el geoide.

El ajuste se hace determinando el llamado “punto fundamental” en el que el elipsoide elegido como referencia es tangente al geoide. Al conjunto de parámetros que definen ese punto fundamental se lo llama Datum y define entre otras cosas, la posición de origen y la orientación de las líneas de latitud y longitud del sistema de coordenadas.

Todos los Datum están basados sobre un elipsoide de referencia.

El Datum puede ser:

- **Local:** es el datum que mejor aproxima el tamaño y forma de una parte determinada de la tierra a nivel del mar. El centro de su esfera no coincidirá con el centro de masa de la tierra. (Ejemplo: POSGAR 94).
- **Geocéntrico:** es el datum que mejor aproxima el tamaño y forma de la tierra como un todo. El centro de su esfera coincide con el centro de masa de la tierra.(Ejemplo: WGS 84)

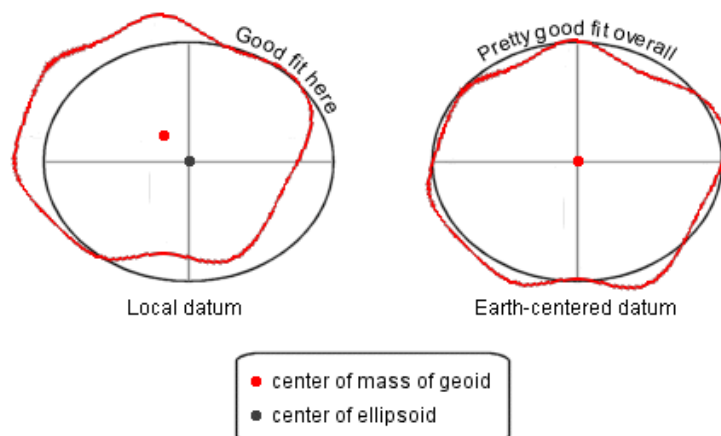


imagen:http://www.geography.hunter.cuny.edu/~jochen/GTECH361/lectures/lecture04/concepts/Datums/Components%20of%20a%20datum_files/image004.gif

Los conceptos antepuestos describen la existencia de Sistemas de Referencias que permiten representar la información espacial de la mejor manera posible en función de la zona geográfica real.

¿A que se le denomina SRE o CRS?

Los SRE, SRS o CRS (por sus siglas en inglés SRS: Spatial Reference System, CRS : Coordinate Reference System) , como su nombre lo indica, son Sistemas de Referencia Espacial usados para poder ubicar los objetos geográficos en un mapa. Existe una gran cantidad de sistemas, de los cuales destacan los siguientes:

SRE Organización

WGS World Geodesic System

NSRS National Geodetic Survey

EPSG European Petroleum Survey Group

Estos pueden ser:

- **Indirectos:** Un sistema indirecto usa referencias a objetos cuya posición conocida, por ejemplo límites estatales, direcciones o monumentos.
- **Directos:** Un sistema directo debe de hacer referencia al datum de algún SRE, el datum se representa por medio de un identificador estándar, por ejemplo: WGS84 o EPSG:22185.

¿Que es WGS 84 o EPSG:4326?¹⁴

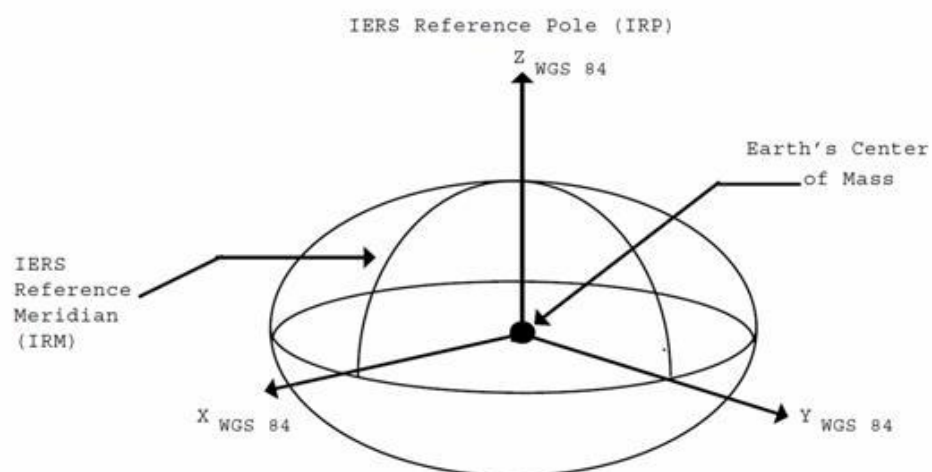
El World Geodetic System 1984 (WGS84) o su equivalente EPSG:4326.

Desde 1987, el GPS utiliza el World Geodetic System WGS-84, que es un sistema de referencia terrestre único para referenciar las posiciones y vectores.

Se estableció este sistema utilizando observaciones Doppler al sistema de satélites de navegación NNSS o Transit, de tal forma que se adaptara lo mejor posible a toda la Tierra.

Se define como un sistema cartesiano geocéntrico del siguiente modo:

- Origen, centro de masas de la Tierra, incluyendo océanos y atmósfera.
- Eje Z paralelo a la dirección del polo CIO o polo medio definido por el BIH¹⁵, época 1984.0 con una precisión de 0,005".
- El eje X la intersección del meridiano origen, Greenwich, y el plano que pasa por el origen y es perpendicular al eje Z, el meridiano de referencia coincide con el meridiano cero del BIH en la época 1984.0 con una precisión de 0,005". Realmente el meridiano origen se define como el IERS Reference Meridian (IRM).
- El eje Y ortogonal a los anteriores, pasando por el origen.
- Terna rectangular dextrosum.



¹⁴ <http://www.ign.es/ign/layoutIn/actividadesGeodesiaStmagd.do>

¹⁵ La International Time Bureau (Oficina Internacional de la Hora en español)

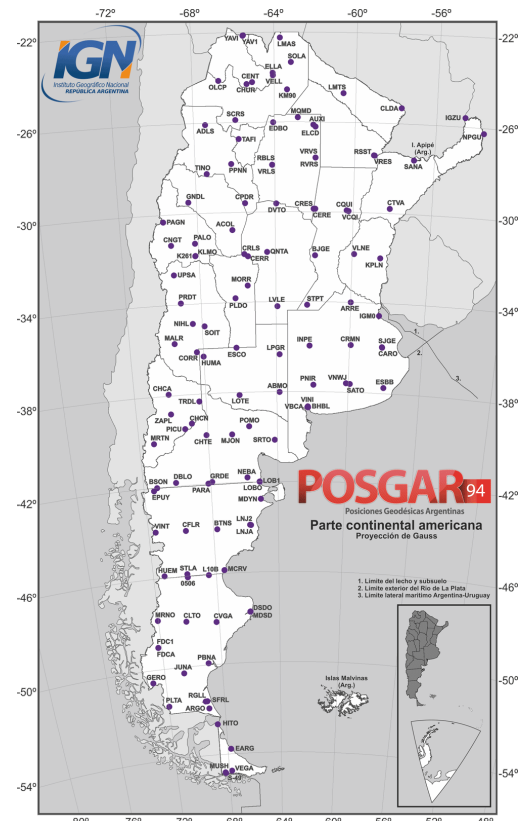
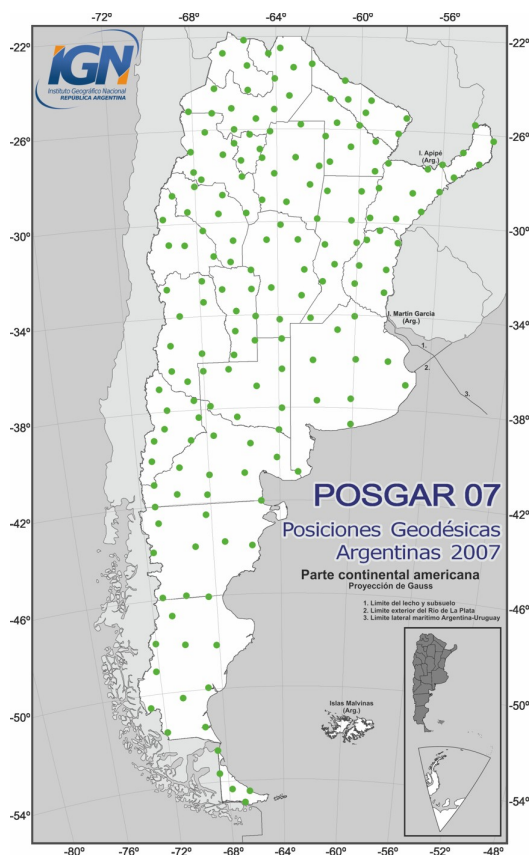
¿Qué “elipsoide de referencia” usa Argentina?

El Datum geodésico utilizado en Argentina se denomina POSGAR 94 (Posiciones Geodésicas Argentinas) definido por primera vez en 1994 por el Instituto Geográfico Militar de la República Argentina para abarcar el ámbito de mapas topográficos y levantamiento geodésico tanto en tierra como en mar.

Nuestra ciudad se ajusta con mayor precisión a EPSG 22185.

POSGAR 94 hace referencia al elipsoide WGS 84 y al meridiano de Greenwich, dando origen a una red geodésica de alta precisión de 127 puntos medidos sobre la base de WGS 84 definido por el Sistema Geodésico Nacional (Sistema Geodésico Nacional).

POSGAR 98 es superior técnicamente, pero no está impuesto legalmente, por sobre POSGAR 94, hasta mayo de 2009, cuando



POSGAR 2007 fue aceptada oficialmente y se sustituye oficialmente POSGAR 94 por esta nueva versión, sin embargo su uso no ha caducado.



Casa de Piedra, Pcia. de Catamarca

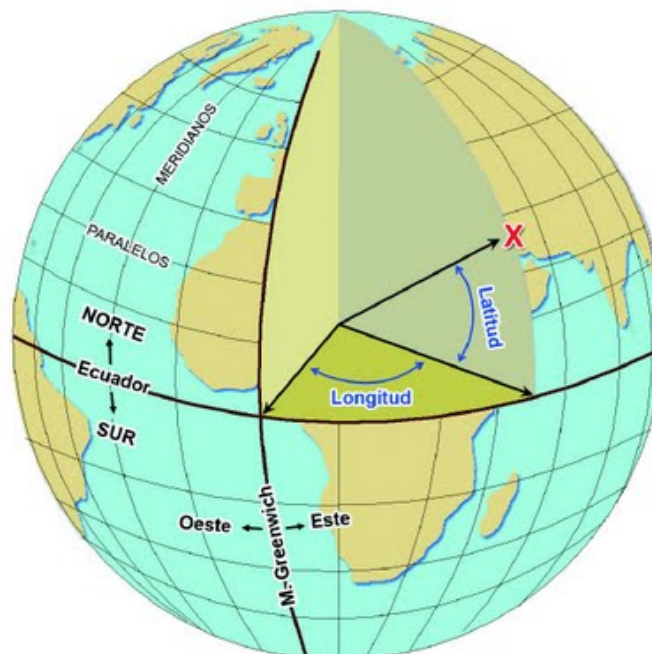
¿Qué es un Sistema de Coordenadas?

Los sistemas de coordenadas permiten a los datasets¹⁶ geográficos utilizar ubicaciones comunes para la integración. Se trata de sistemas que se utilizan para representar la ubicación de entidades geográficas, imágenes y observaciones (como las localizaciones GPS – sistemas de posicionamiento global) dentro de un marco geográfico común.

Tipos de sistemas de coordenadas:

En los sistemas de información geográfica (SIG) se utilizan habitualmente dos tipos de sistemas de coordenadas:

Los sistemas de coordenadas globales o esféricas, basado por ejemplo en latitud (Norte-Sur) – longitud (Este-Oeste). Estas dos coordenadas angulares medidas desde el centro de la Tierra son de un sistema de coordenadas esféricas que están alineadas con su eje de rotación. La definición de un sistema de coordenadas geográficas incluye un datum, meridiano principal y unidad angular. Estas coordenadas se suelen expresar en grados sexagesimales. Generalmente, éstos son llamados sistemas de coordenadas geográficas.



Los sistemas
coordenadas

de
proyectadas

¹⁶ Un "Conjunto de datos" o "dataset" es una colección de datos normalmente tabulada. Por cada elemento se indican varias características.

(proyecciones cartográficas) surgen como alternativa ante la dificultad de los sistemas angulares en medir distancias constantes. Las proyecciones cartográficas se utilizan para definir un sistema coordenado plano, el cual es una grilla sobre la que se dibuja un mapa en dos dimensiones.

Es decir, se puede definir a una proyección como la transformación de un espacio tridimensional en uno bidimensional.

Ejemplo de Proyecciones pueden ser : Mercator, Mercator Transversal (Sistema Gauss Krüger, UTM), Lambert.

Los sistemas de coordenadas (geográficas o proyectadas) ofrecen un marco en el que definir las ubicaciones del mundo real.

¿Que sistema de coordenadas se emplea en Argentina?

fuelle: instituto geografico nacional

Se puede definir una proyección diciendo que es un sistema plano de meridianos y paralelos sobre el cual puede dibujarse sobre un mapa.

Según la deformación producida por el pasaje de la esfera al plano, los sistemas de proyección pueden ser: equivalentes¹⁷, conformes¹⁸ o afiláctica¹⁹.

También pueden ser clasificadas según su modo de obtención atribuyéndose por su complejidad figuras geométricas capaces de aplanarse para representar la tierra: cilíndricos, cónicos y acimutales o planas.

Disposición gráfica de las 7 fajas sobre la parte continental americana de la República Argentina

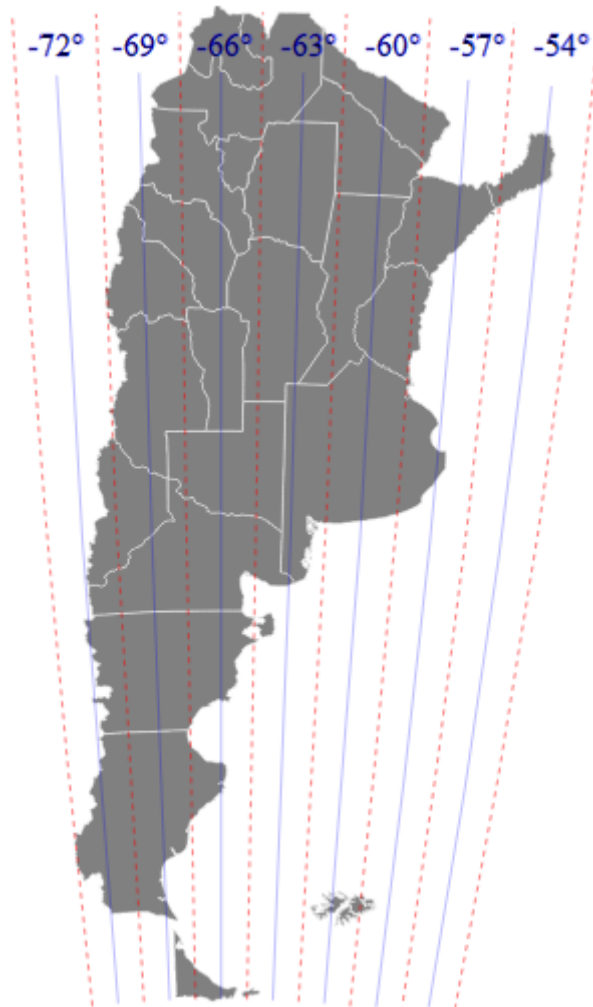
Los sistemas de proyección utilizados nuestro país y empleado por el Instituto Geográfico Nacional (IGN) es la proyección GAUSS-KRÜGER para la confección de las cartas topográficas nacionales, divide a la República

¹⁷ Son aquellas que conservan las superficies (áreas) de terreno, aunque las formas dejen de ser semejantes.

¹⁸ Son aquellas que mantiene los ángulos que forman dos líneas sobre la superficie terrestre.

¹⁹ Son aquellas que no conservan distancias, ángulos ni superficies, pero donde las deformaciones son mínimas.

Argentina (sector continental e Islas Malvinas) en 7 fajas meridianas de Oeste a Este.



Cada faja de la grilla Gauss-Krüger mide 3° de ancho (longitud) por 34° de largo (latitud) y tiene como propio origen la intersección del POLO SUR con el meridiano central de cada faja.

Al igual que en la proyección utilizada en otros países, la UTM (Mercator Transversal Universal), y con el objeto de evitar coordenadas negativas, se le asigna al meridiano central de cada faja el valor arbitrario de 500.000 metros y al POLO SUR el valor de cero metros.

Cabe señalar que en esta proyección el origen de las ordenadas "X" es el POLO SUR y son positivas hacia el ECUADOR. Su valor expresa la distancia en metros del polo al punto, según la dirección del meridiano central de faja a la

cual pertenece el punto. El origen de las abscisas “Y” es el meridiano central de cada faja.

Faja Meridiana	Meridiano Central de Faja	Meridiano Límite de Faja
1	-72°	-73° 30', -70° 30'
2	-69°	-70° 30', -67° 30'
3	-66°	-67° 30', -64° 30'
4	-63°	-64° 30', -61° 30'
5	-60°	-61° 30', -58° 30'
6	-57°	-58° 30', -55° 30'
7	-54°	-55° 30', -52° 30'

En la ciudad de santa fe se utiliza la faja 5 “EPSG:22185 ” para la representación física de los datos pero los mismos serán visualizadas o proyectados en el editor utilizando EPSG:4326, por compatibilidad con otras herramientas como google u OpenStreet.

¿Qué es una base de datos geoespacial?

“Una base de datos geográfica es una colección de datos organizados de tal manera que sirvan efectivamente para una o varias aplicaciones SIG.

Esta base de datos comprende la asociación entre sus dos principales componentes: datos espaciales y atributos o datos no espaciales”. [ESRI, 1998]

En este tipo de bases de datos es imprescindible establecer un cuadro de referencia (un SRE, Sistema de Referencia Espacial) para definir la localización y relación entre objetos, ya que la información tratada en estos almacenes tienen un valor relativo, no es un valor absoluto.

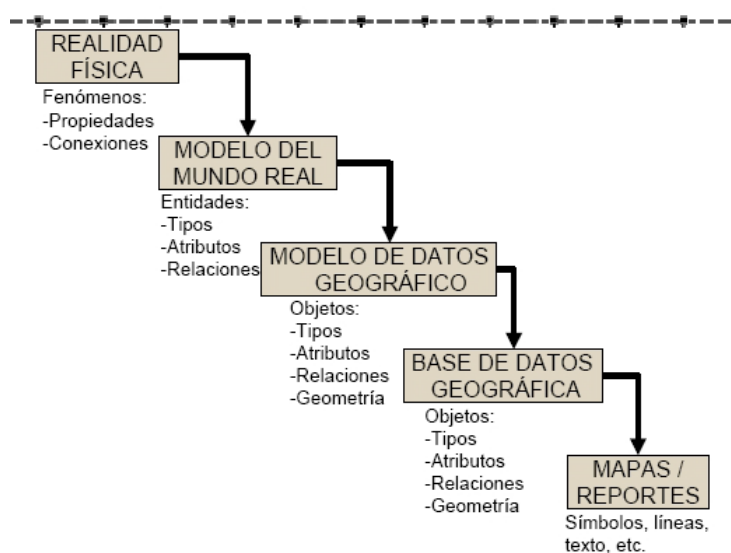
La mayoría de los sistemas de información geográfica adoptan una arquitectura en la cual los datos son administrados a través de dos modelos de datos heterogéneos:

Un administrador de bases de datos (en inglés database management system , abreviado DBMS) para los datos descriptivos y Un módulo específico para la administración de datos espaciales.

Los administradores de bases de datos espaciales (en inglés spatial database management system, abreviado SDBMS) permiten la adopción de una arquitectura integrada, en la cual el administrador de datos es extendido para almacenar tanto la descripción de los objetos como su geometría. El lenguaje de consulta SQL (por sus siglas en inglés structured query language o lenguaje estructurado de consulta) es extendido para manejar nuevos tipos de datos (puntos, líneas y polígonos) y son incorporadas nuevas funciones que permiten la selección y recuperación de los datos, no sólo por criterios alfanuméricos, sino que también aplicando criterios espaciales a través de relaciones topológicas, de orientación, medición, entre otras.

La construcción de una base de datos geográfica implica un proceso de abstracción para pasar de la complejidad del mundo real a una representación simplificada que pueda ser procesada por el lenguaje de las computadoras actuales.

Diseño y Planeamiento de la BD de un SIG



Proceso de diseño y planeamiento de una base de datos de un sistema de información geográfica.

[Fuente: apuntes de la cátedra electiva Sistemas de Información Geográfica-FRSFUTN.]

MANUAL DE USUARIO

Editor de entidades Geográficas

Ingreso al sistema:

El ingreso a la aplicación de edición se realiza mediante un panel de autenticación FIG 01, donde cada usuario ingresa sus credenciales y luego realiza un clic derecho sobre el botón Ingresar.

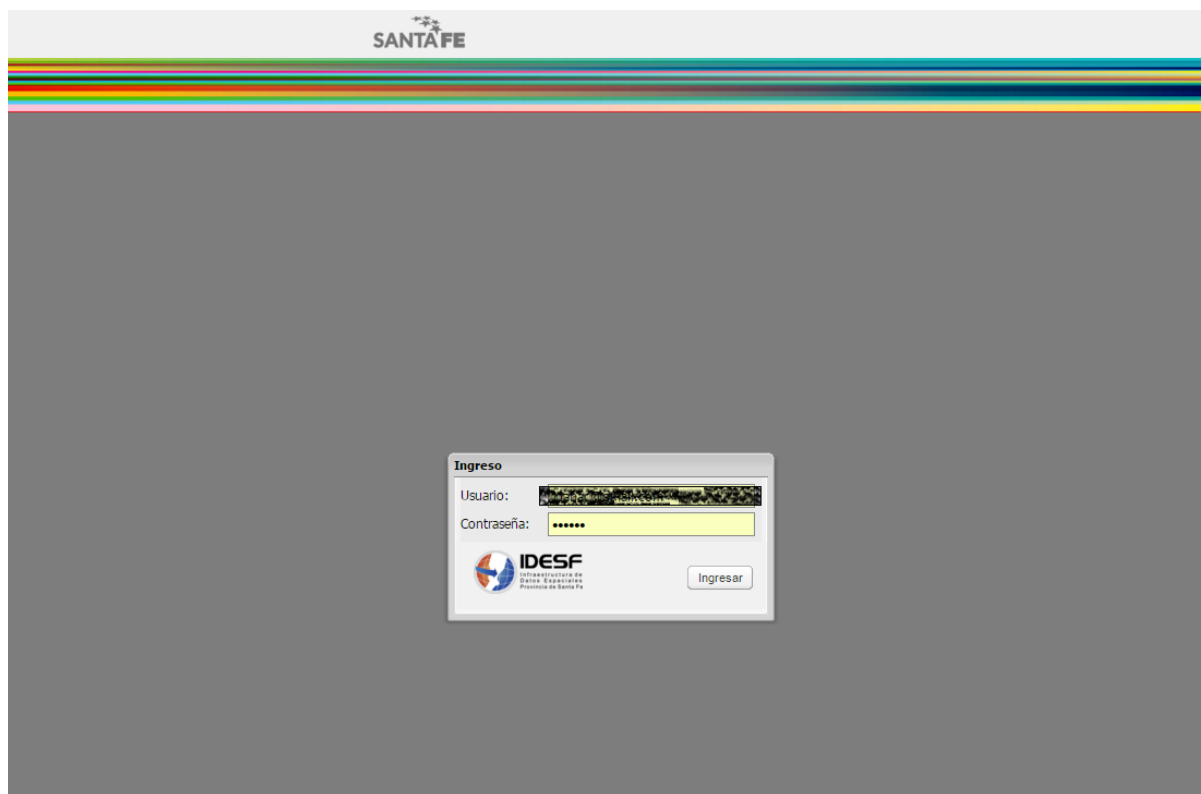


FIG 01

Interfaz de Edición:

La interfaz presenta un diseño simple y limpio para el usuario que pueda visualizar claramente los elementos del editor FIG 02.

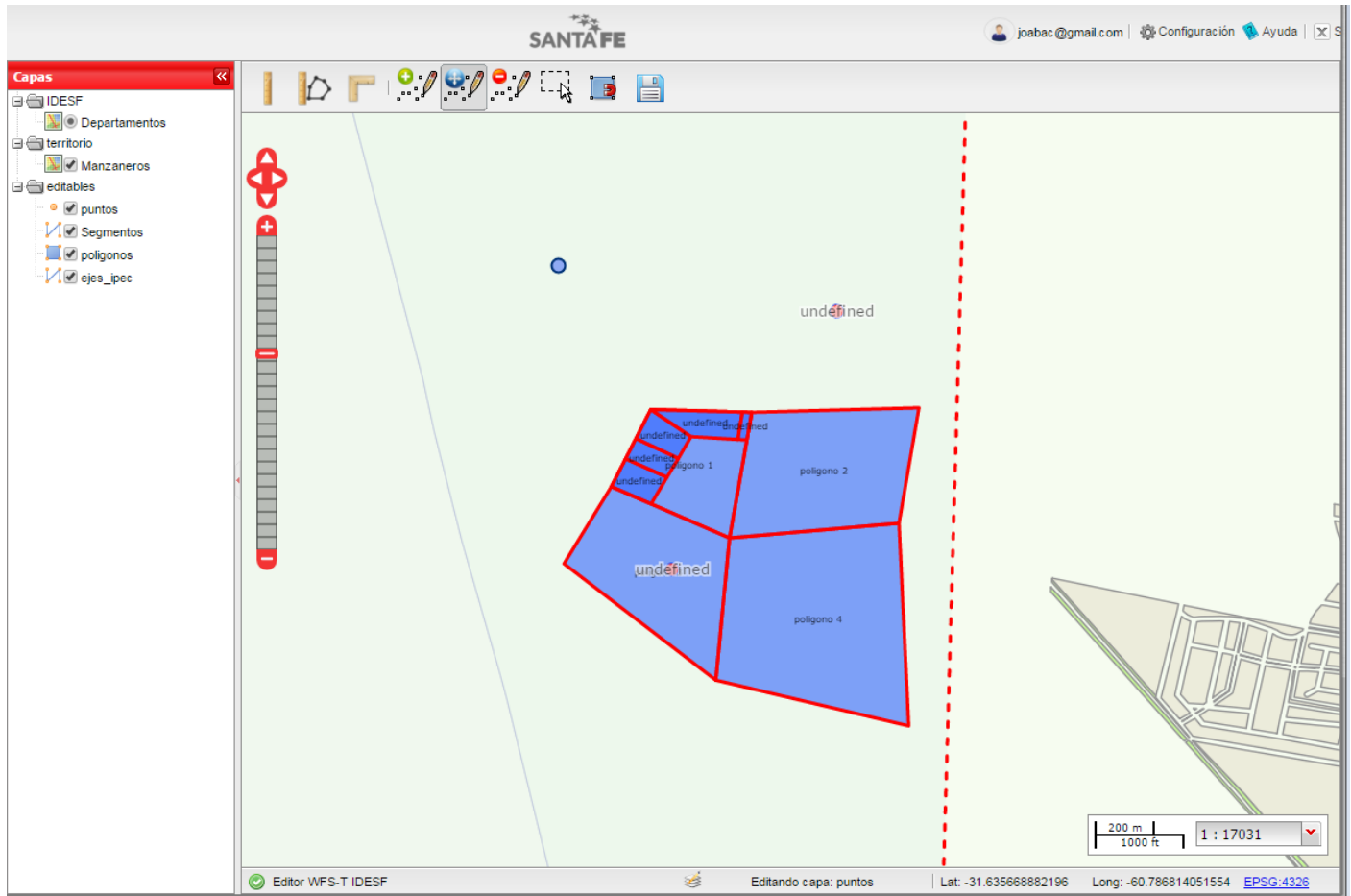


FIG 02

Básicamente se presentan tres regiones:

- Izquierda: Árbol de capas
- Panel Central: Mapa y herramientas de edición.
- Panel Superior: Barra de información y perfil de usuario

Árbol de capas

Un árbol de capas típico se puede observar en la FIG 03

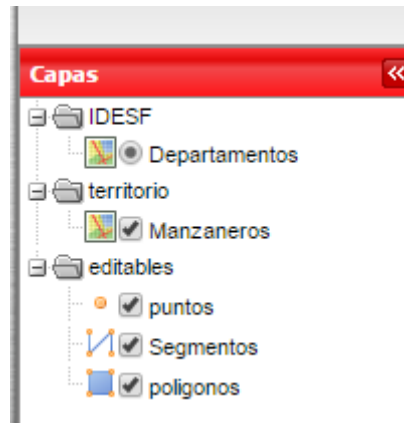






FIG 03

Cada usuario tendrá su conjunto de capas habilitadas, por lo que cada árbol generado podría ser diferente.

Se pueden observar diferentes tipos de capas e iconos que brindan información al usuario para discernir las capas de base de las editables o de superposición:

Tipos de Icono:

	Capa Raster o de Imagen
	Capa editable de Puntos
	Capa editable de Segmentos
	Capa editable de Polígonos

Tipo de Capa:

Según el elemento de selección RadioButton o Checkbox

<input type="radio"/>	Capa de base
<input checked="" type="checkbox"/>	Capa de superposición o editable (según icono asociado)

Menú Contextual de Capas

Las capas del tipo editable permiten se realice clic derecho sobre las mismas, a lo cual despliega un menú contextual FIG 04 con opciones para editar las mismas, cambiar su estilo y cambiar sus características.

Para cada capa, el menú se ajusta indicando que capa se está por gestionar.

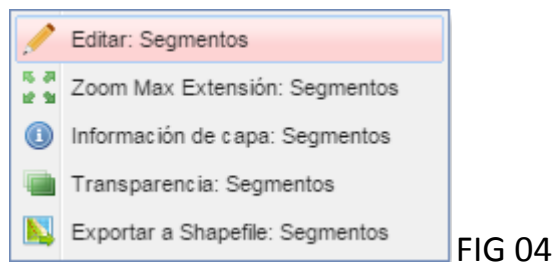


FIG 04

Edición de capas:

Al seleccionar esta opción del menú contextual se interrogará al usuario si realmente desea iniciar la edición FIG 05.

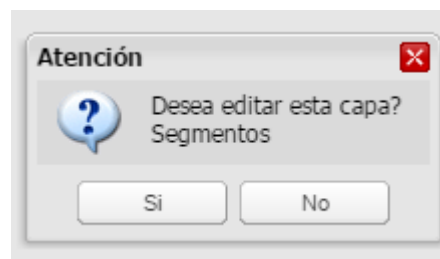
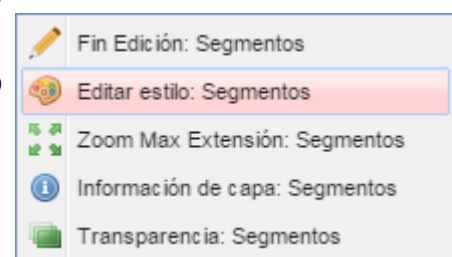


FIG 05

La siguiente vez que realice un clic derecho sobre la capa el menú contextual ofrecerá la opción de finalizar el proceso de edición FIG 06 en caso que existan cambios sin guardar se ofrecerá la opción de continuar o guardar antes de finalizar.

FIG 06



Edición de Estilos:

Se puede además para cada capa editable definir un estilo específico y que será persistente para cada usuario, esto se puede realizar seleccionando la opción del menú contextual “*Editar Estilo: <nombre capa>*”

Al seleccionar esta opción se presentará una ventana de configuración o edición de reglas FIG 07 que serán aplicadas a la capa y que definirán el estilo con que será visualizada.

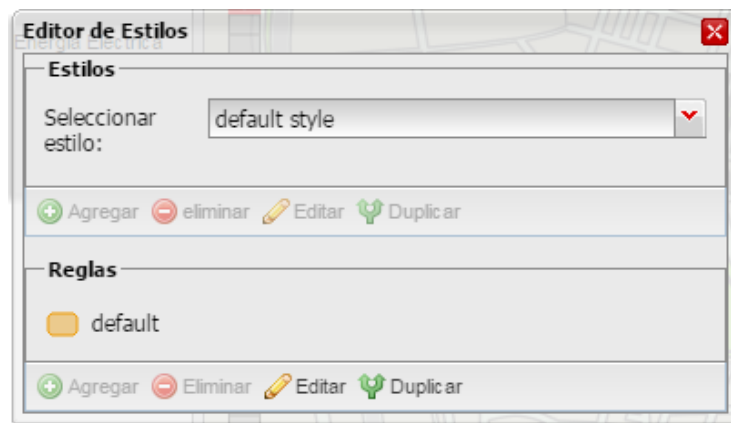


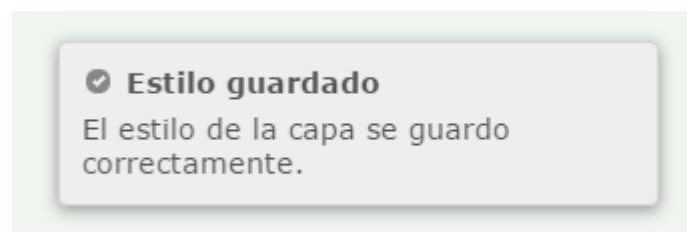
FIG 07

Aquí podremos seleccionar la regla a editar, y realizando un clic izquierdo sobre el botón *Editar*, se presentará una ventana de edición específica para la regla.

Para cualquier cambio que se realice en los estilos, se deberá luego realizar un guardado manual, si se desea persistir el mismo para futuras aperturas, utilizando el disquete que se presenta en la barra de menú.



Si el estilo se guardó con éxito se informará mediante un mensaje emergente.



En este panel podremos editar el estilo básico, de las entidades geográficas (colores, grosores de línea, estilos de trazado, formato de iconos) FIG 08.

También se podrá definir las etiquetas o labels que se mostrarán sobre las entidades geográficas, como así también su estilo (Tipo de letra tamaño, color) FIG 09.

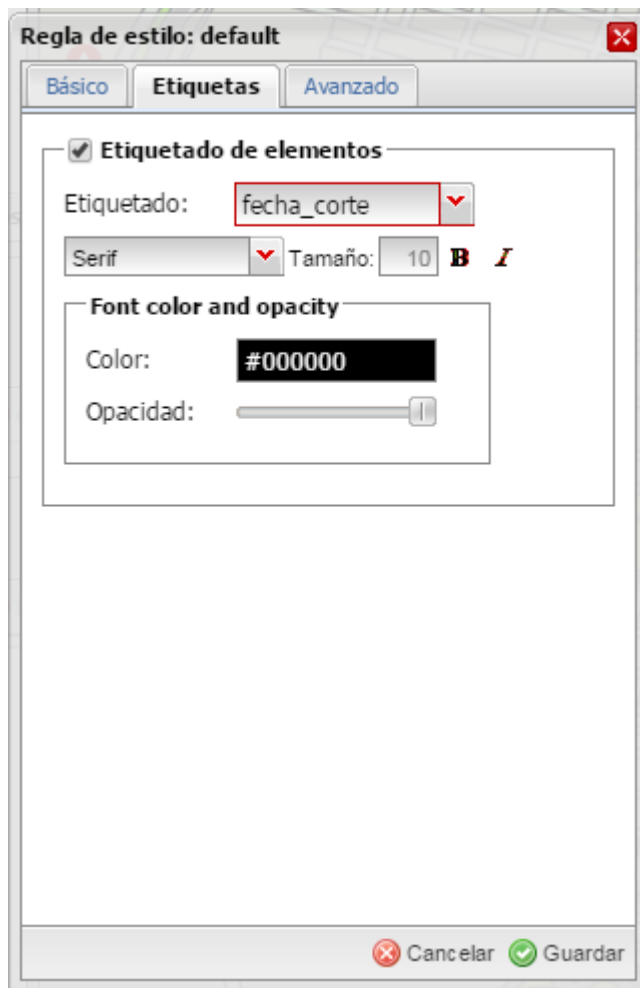


FIG 09

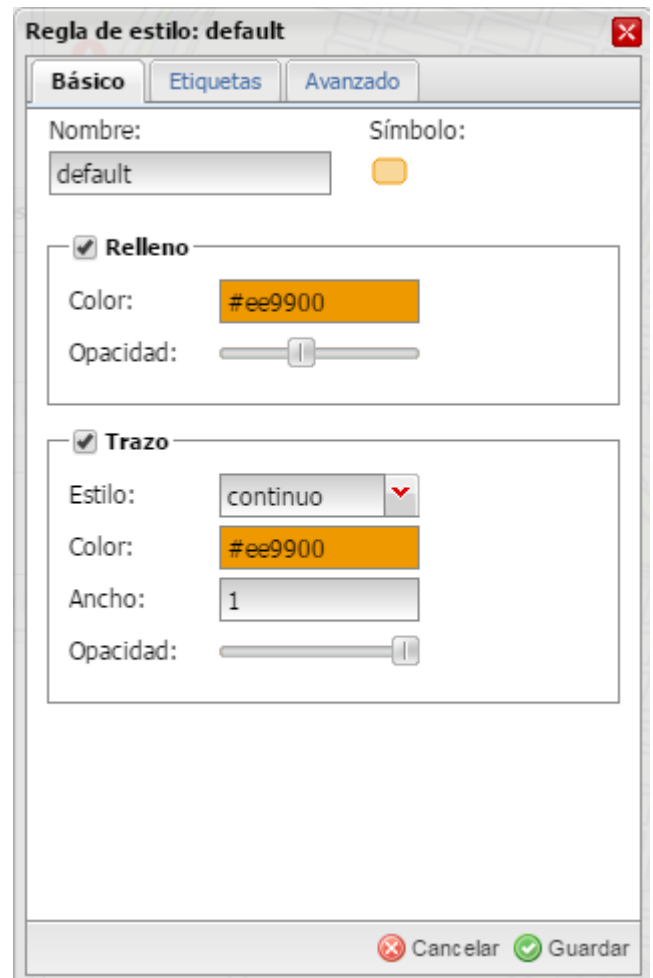


FIG 08

Se pueden configurar también atributos avanzados FIG 10, como ser la visualización limitada por la escala, o reglas de visualización y filtrado con reglas del tipo mayor o igual que, menor que o con grupos lógicos del tipo cumplir todas : AND lógico, cumplir cualquiera : OR lógico, o ninguna de : NOT lógico .

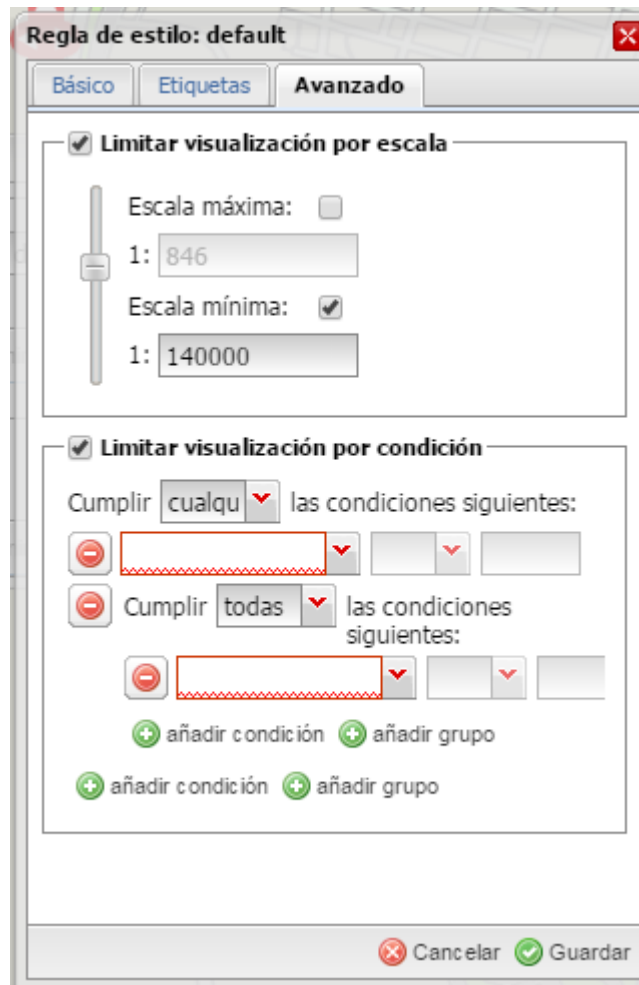


FIG 10

Panel Central:

Edición de Entidades Geográficas:

Llamaremos de aquí en más al proceso de Edición, tanto a la inserción, modificación como eliminación de elementos geográficos (puntos, segmentos, polígonos) sobre un mapa.

Edición de Puntos:

Este tipo de geometrías nos permiten definir posiciones o ubicaciones específicas de objetos, que a modo de ejemplo y enumerando solo algunas opciones podrían ser :

- Domicilios
- Monumento
- lugares de interés
- Ubicación donde se cometió un delito.

Para realizar la edición de una capa de puntos primero iniciar la edición de la misma, haciendo clic izquierdo sobre la capa de puntos a editar FIG 11.

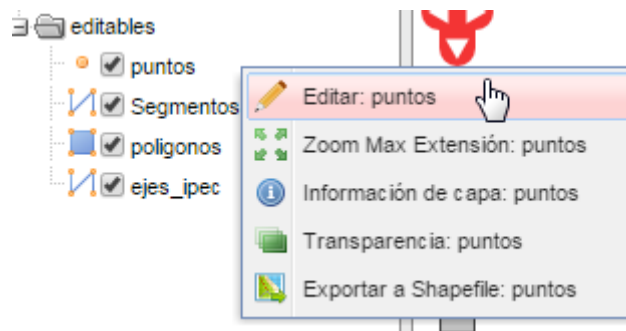


FIG 11

Si el mapa se encuentra en los límites de escala mínimos definidos para esa capa, se habilita entonces la barra de edición en la parte superior del panel central y se dibujara la región habilitada sobre el mapa , sino se informará mediante un mensaje de error dicha situación FIG 12

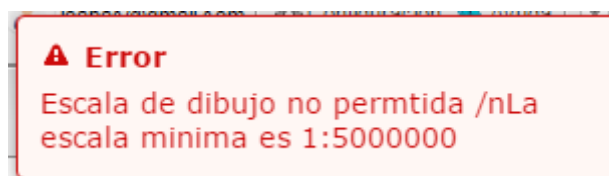


FIG 12

Si se pudo ingresar al modo de edición veremos algo similar a lo mostrado en la FIG 13.

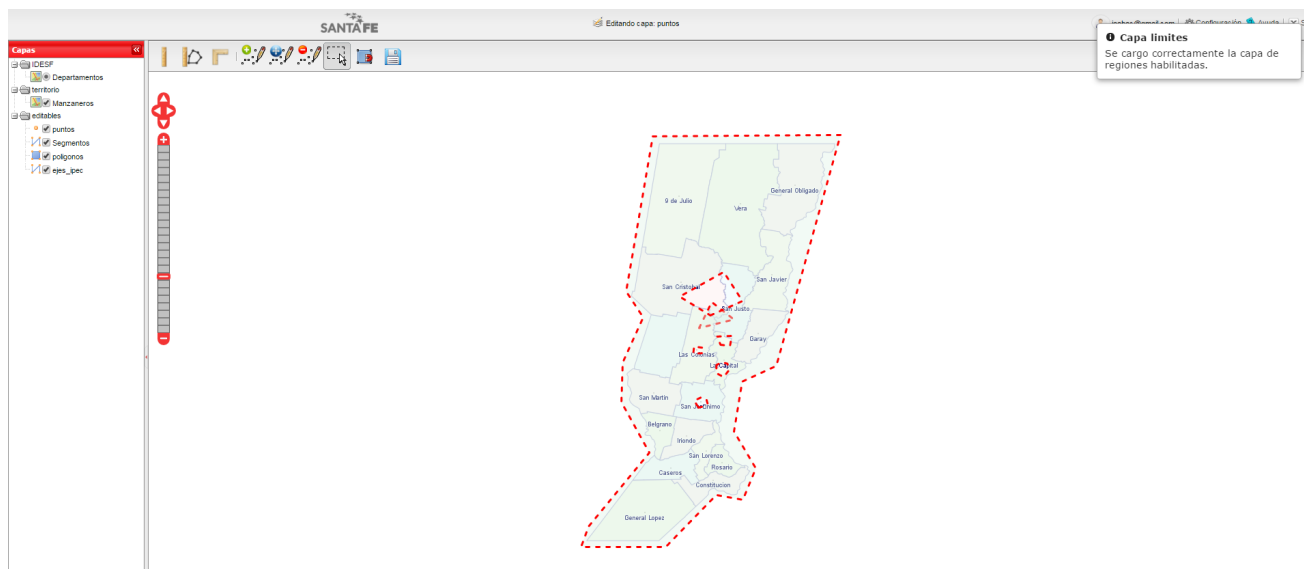








FIG 13

En la barra de herramientas se presentarán los siguientes iconos:

	<p>Insertar nuevos puntos</p>
	<p>Modificar puntos existentes</p>
	<p>Eliminar puntos existentes</p>
	<p>Seleccionar Puntos</p>
	<p>Control Snap para edición e inserción</p>
	<p>Salvar Cambios y estilos</p>

Insertar nuevos puntos

Para realizar esta operación hacer un clic izquierdo sobre la herramienta con el icono :



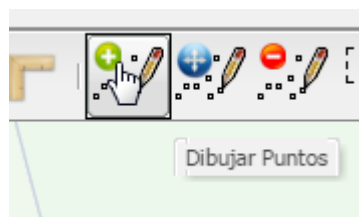
El editor, pasará al modo de inserción y bastará con posicionar el cursor del mouse sobre la región deseada y realizar un clic izquierdo.

El punto quedará dibujado con el estilo que tenga definido el usuario para la capa en edición y luego guardar los cambios haciendo clic izquierdo en el icono del disquete .

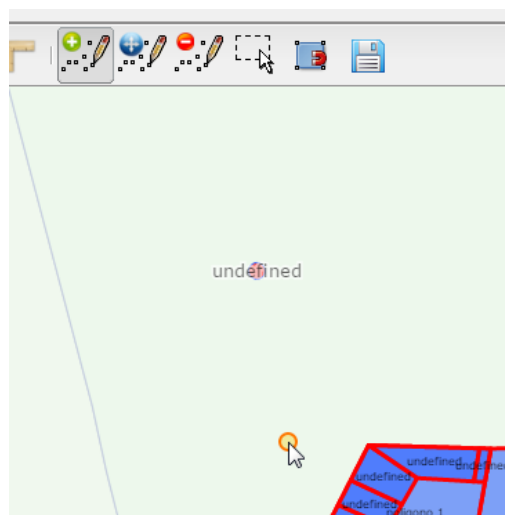
Resumen de pasos:

- 1 - Seleccionar herramienta.
- 2 - Posicionar puntero de mouse.
- 3 - Clic izquierdo.
- 4 - Guardar cambios haciendo clic.

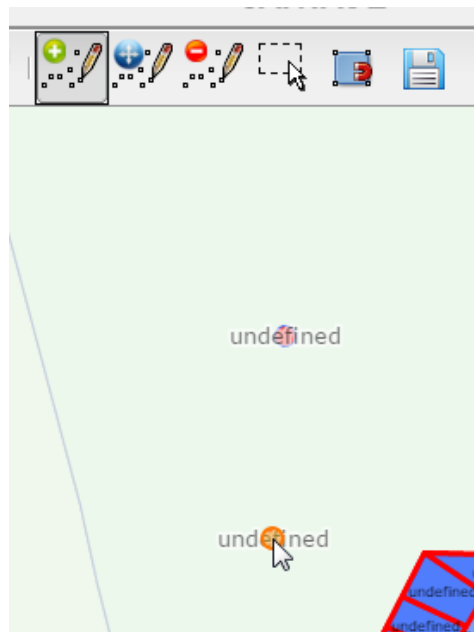
paso 1



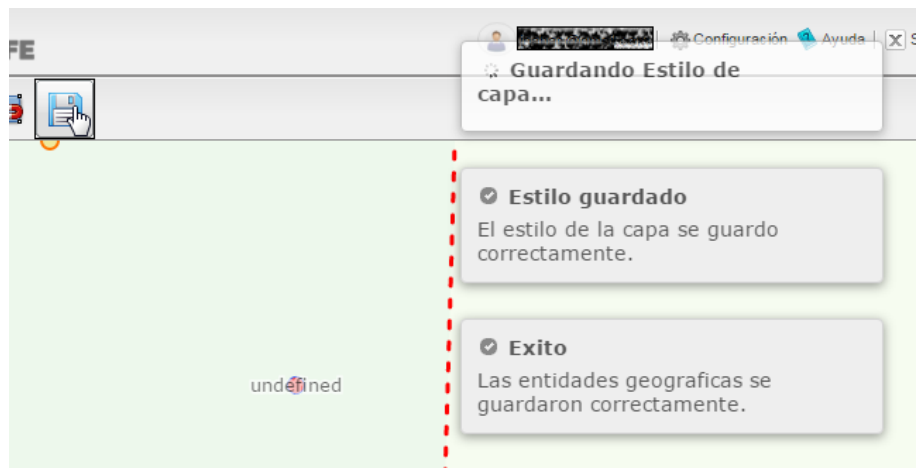
paso 2



paso 3



paso 4



Al finalizar el proceso se informará que los cambios se guardaron correctamente.

Modificar puntos existentes

Para realizar esta operación hacer un clic izquierdo sobre la herramienta con el icono :



El editor, pasará al modo de modificación, se deberá entonces seleccionar el punto que se desea mover, haciendo un clic izquierdo sobre el mismo.

El punto quedará dibujado con un estilo diferente, que representa el estado de seleccionado.

Para mover o modificar su posición, ubicar el cursor sobre el punto seleccionado y manteniendo presionado el botón derecho del mouse desplazar hasta la posición deseada y liberar el botón.

Resumen de pasos:

- 1 - Seleccionar herramienta.
- 2 - Clic izquierdo para seleccionar el punto a modificar.
- 3 - Mantener presionado botón izquierdo del mouse sobre el punto seleccionado y desplazar hasta posición deseada.
- 4 - Liberar botón del mouse.
- 5 - Guardar los cambios

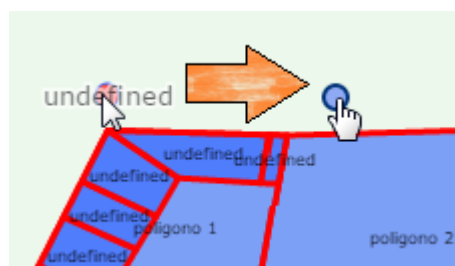
paso 1



paso 2



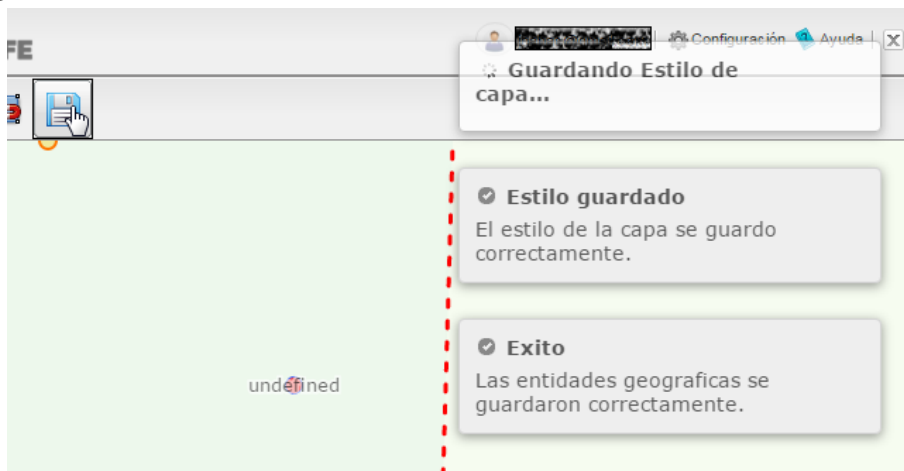
paso 3



paso 4



paso 5



Eliminar puntos existentes

Para realizar esta operación hacer un clic izquierdo sobre la herramienta con el icono :



El editor, pasará al modo de eliminación, se deberá entonces hacer clic sobre el punto que se desea eliminar, haciendo un clic izquierdo sobre el mismo.

El usuario será alertado sobre que la operación es irreversible.

Si selecciona Sí, la misma será eliminada, de otra manera el proceso será descartado.

Resumen de pasos:

- 1 - Seleccionar herramienta.
- 2 - Clic izquierdo para eliminar el punto deseado.
- 3 - Seleccione Si, para eliminar el punto, o No, para descartar el proceso
- 4 - Si seleccionó Si deberá Guardar los cambios .

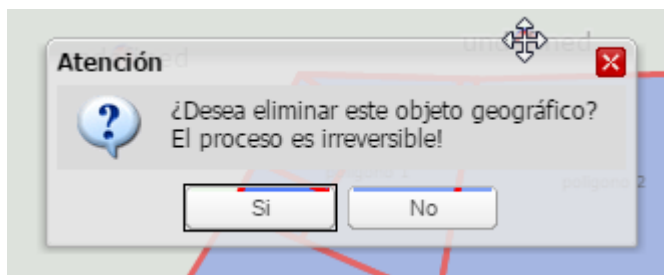
paso 1



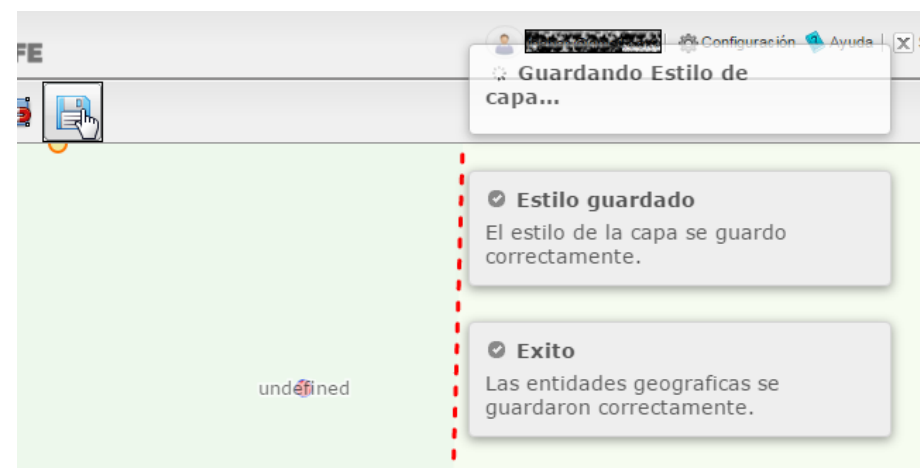
paso 2



paso 3



paso 4



Seleccionar Puntos

Esta herramienta permite seleccionar puntos específicos o un conjunto de puntos, fue implementada para dar solución a requerimientos específicos de productores que deseen realizar operaciones particulares sobre algún punto o un conjunto de los mismos. Actualmente la herramienta no posee una funcionalidad específica.

Algunos ejemplos podrían ser:

- La selección de un conjunto de puntos y la impresión de los mismos.
- Selección de puntos a exportar a archivo de exportación.

Control Snap para edición e inserción

Esta herramienta es de gran utilidad a la hora de insertar o editar elementos geográficos ya que habilita la opción de magnetismo sobre las capas seleccionadas, El efecto visible y sensible será que al aproximar el cursor con el elemento a insertar o a desplazar, el mismo realizará un salto hacia los vértices, laterales o puntos de las capas que se hayan seleccionado como elemento de referencia para el Snap²⁰

Para realizar esta operación, realizar un clic izquierdo sobre la herramienta con el icono :



Se mostrará entonces una ventana de configuración para esta herramienta FIG 14

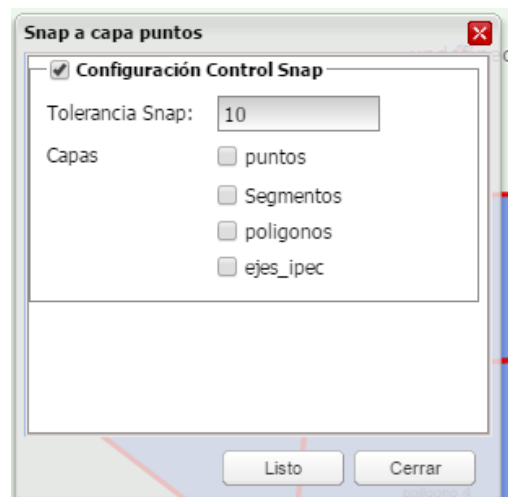


FIG 14

²⁰ [https://en.wikipedia.org/wiki/Snap_\(computer_graphics\)](https://en.wikipedia.org/wiki/Snap_(computer_graphics))

La tolerancia Snap FIG 15 hace referencia a la cantidad de píxeles que serán utilizados para calcular el punto desde el cual se producirá el salto del elemento hacia el vértice, lateral o punto .

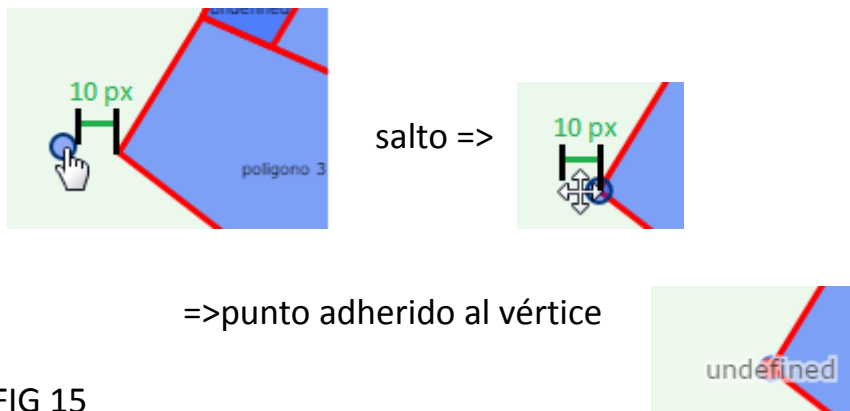


FIG 15

Esto nos asegura que el punto estará exactamente posicionado en las mismas coordenadas que el vértice del polígono utilizado como referencia, quedando así asegurado el contacto si deseamos realizar luego alguna verificación de topología²¹

Las capas sobre las cuales se puede realizar el Snap aparecerán disponibles como un listado seleccionable mediante cajas de selección las cuales se podrán ir activando o desactivando según necesidad.

²¹ La **topología geoespacial** estudia las relaciones espaciales entre los diferentes elementos gráficos

Edición de Líneas o Segmentos:

Este tipo de geometrías nos permiten definir rasgos lineales, a modo de ejemplo se puede enumerar algunas opciones:

- Tendidos eléctricos.
- Recorridos de transportes públicos.
- Ejes de calles.
- Cauces de Ríos.

Para realizar la edición de una capa de líneas, primero se debe iniciar la edición de la misma haciendo clic izquierdo sobre la capa de líneas a editar FIG 16.

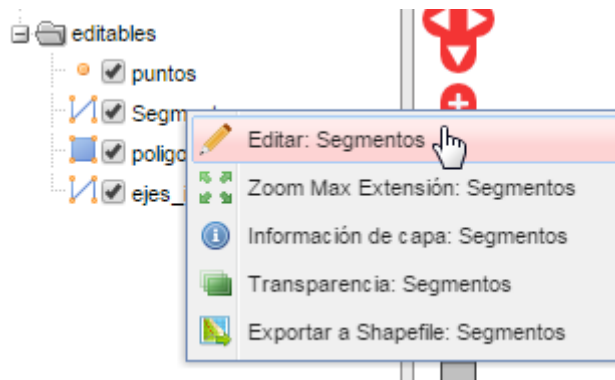


FIG 16

Si el mapa se encuentra en los límites de escala mínimos definidos para esa capa, se habilita entonces la barra de edición en la parte superior del panel central y se dibujara la región habilitada sobre el mapa , sino se informará mediante un mensaje de error dicha situación FIG 17

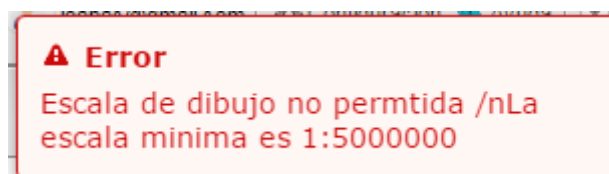


FIG 17

Si se pudo ingresar al modo de edición veremos algo similar a lo mostrado en la FIG 18.

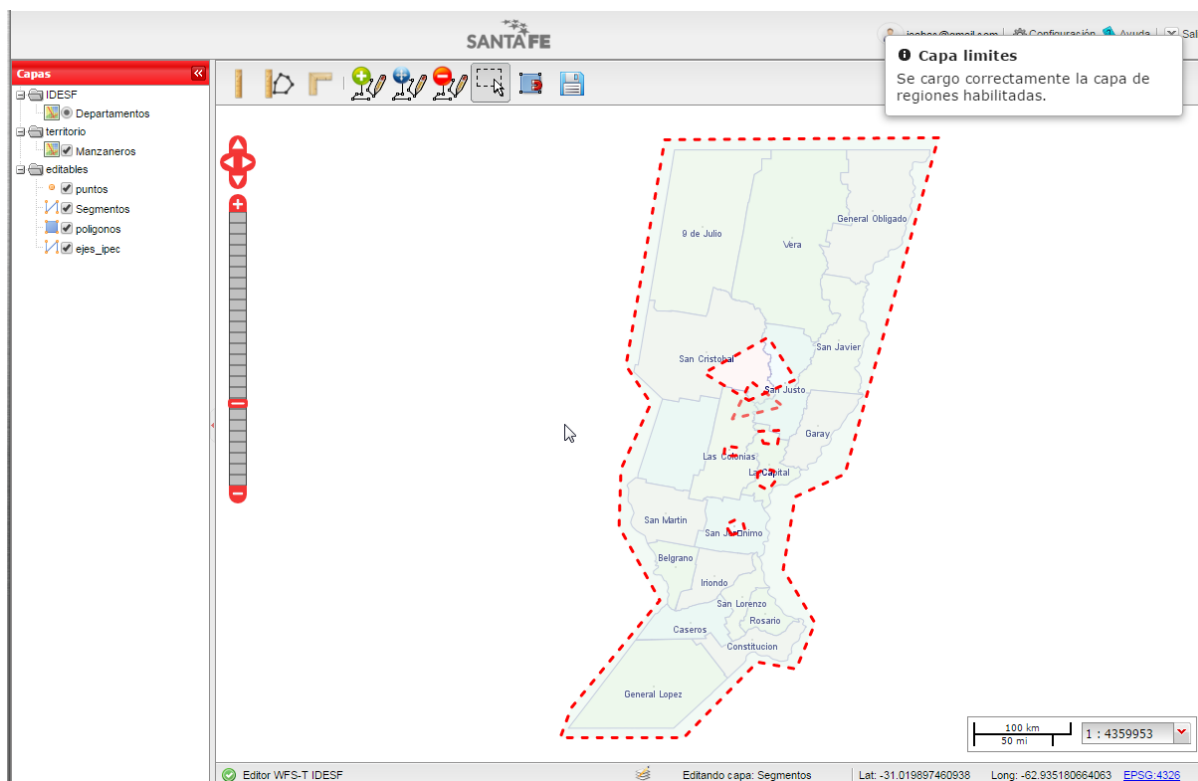








FIG 18

En la barra de herramientas se presentarán los siguientes iconos:

	Insertar nuevos segmentos
	Modificar segmentos existentes
	Eliminar segmentos existentes
	Seleccionar segmentos
	Control Snap para edición e inserción
	Salvar Cambios y estilos

Insertar nuevos Segmentos

Para realizar esta operación bastará con hacer un clic izquierdo sobre la herramienta con el icono :



El editor, pasará al modo de inserción y bastará con posicionar el cursor del mouse sobre la posición inicial deseada y realizar un clic izquierdo.

El punto inicial quedará dibujado y un segmento adherido al cursor del mouse se visualizará y podrá desplazarse visualizando su ubicación, lo que habilita el posicionamiento del mismo, una vez posicionado el eje en ángulo y longitud haciendo un clic izquierdo se genera un vértice nuevo desde el cual se podrá nuevamente extender un segmento, **para finalizar la edición bastará con realizar un doble clic**, finalizada la edición el segmento quedará graficado con el estilo que tenga definido el usuario para la capa en edición, y finalmente se deberá persistir los cambios haciendo clic izquierdo en el icono del disquete

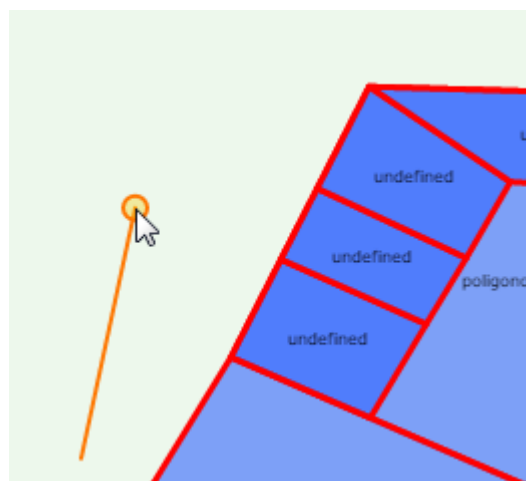
Resumen de pasos: (Polilínea de 2 segmentos)

- 1 - Seleccionar herramienta.
- 2 - Posicionar puntero de mouse.
- 3 - Clic izquierdo.
- 4 - Posicionar primer segmento y realizar un clic izquierdo
- 5 - Finalizar edición del segmento realizando doble clic.
- 6 - Guardar cambios

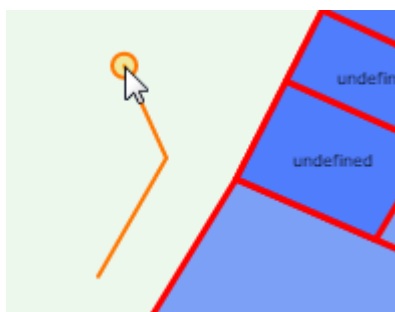
paso 1



paso 2 y 3



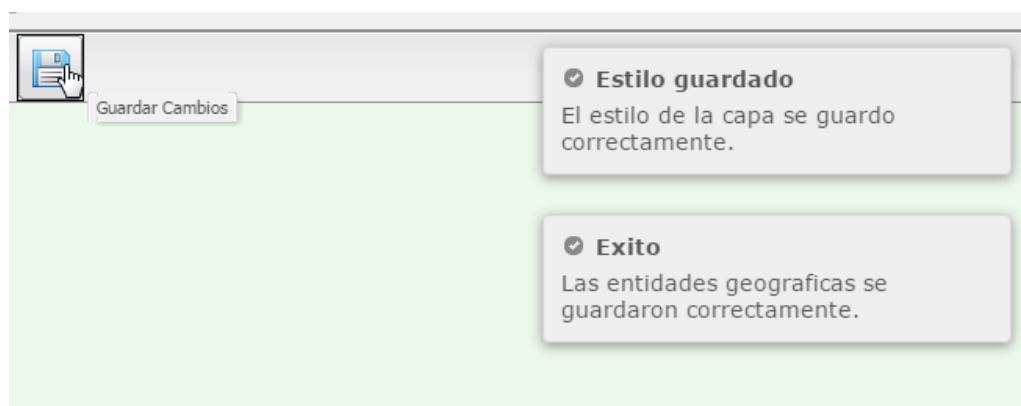
paso 4



paso 5



paso 6



Al finalizar el proceso se informará que los cambios se guardaron correctamente o se mostrará un mensaje de error indicando el error detectado.

Modificar Segmentos existentes

Para realizar esta operación se debe hacer un clic izquierdo sobre la herramienta con el icono :



El editor, pasará al modo de edición , se debe entonces seleccionar el segmento que se desea modificar, haciendo un clic izquierdo sobre el mismo.

El segmento quedará dibujado con un estilo diferente, que representa el estado de seleccionado para edición.

Sobre los segmento podremos, modificar la posición de cualquiera de sus vértices o crear nuevos quebrando los segmentos desde su punto medio.

Para modificar un vértice la operación es similar a la edición de un punto, se debe posicionar el cursor sobre el vértice a mover y mantener entonces presionado el botón izquierdo del mouse, desplazar el vértice y liberar el mouse una vez re posicionado el mismo.

La operación de creación de nuevos vértices es similar a desplazar, un vértice solo difiere en que se debe realizar seleccionando el punto medio de un segmento y desplazando el mismo hasta la posición deseada.

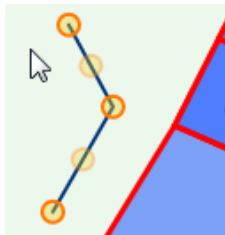
Resumen de pasos: (desplazar un vértice y quebrar un segmento)

- 1 - Seleccionar herramienta.
- 2 - Seleccionar línea a editar.
- 3 - Posicionar cursor sobre vértice a editar y realizar un clic sin liberar el botón, luego posicionar el vértice y liberar botón.
- 4 - Seleccionar Línea
- 5 - Posicionar cursor sobre punto medio de segmento a quebrar y realizar un clic sin liberar el botón, luego posicionar el vértice nuevo en la posición deseada y liberar botón.
- 6 - Guardar los cambios

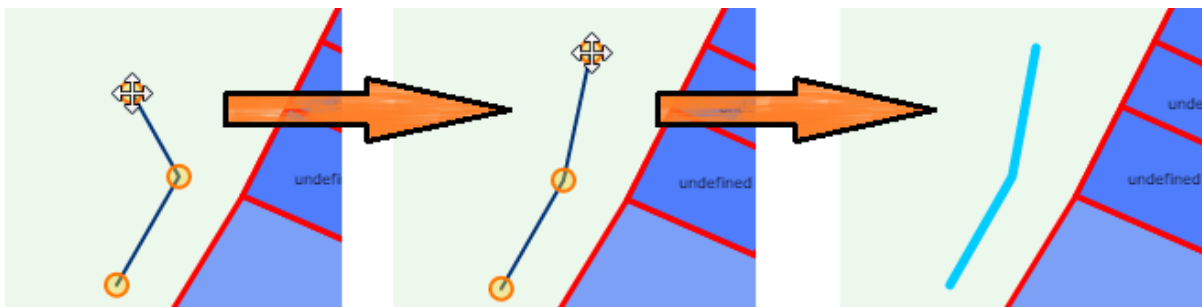
paso 1



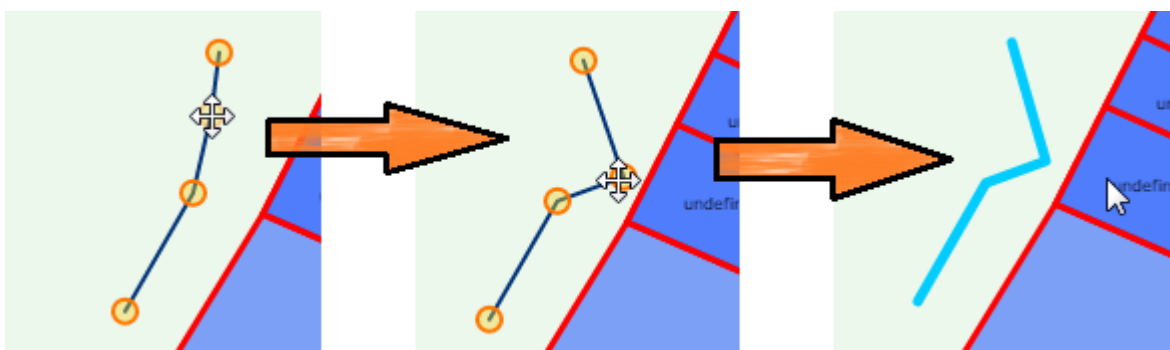
paso 2



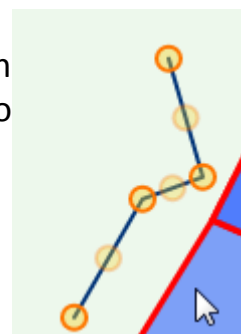
pasos 3



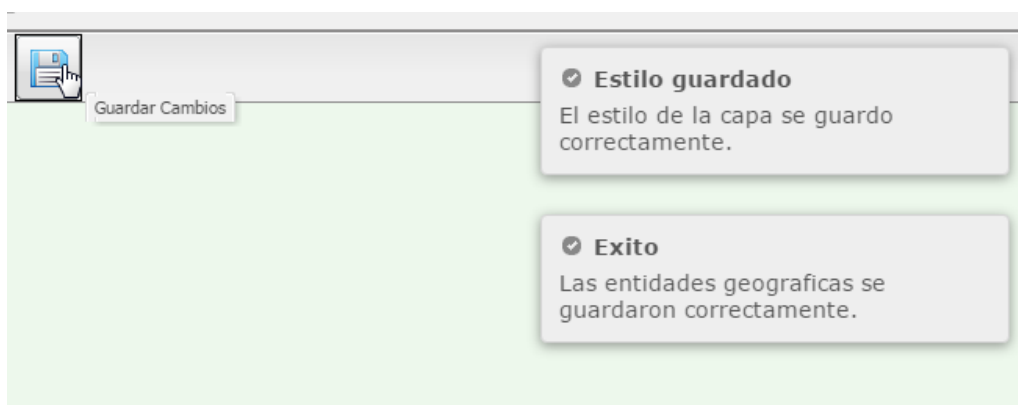
pasos 4 y 5 - "Nuevo vértice"



NOTA: La línea editada ahora contará con un nuevo vértice y en consecuencia un nuevo segmento.



pasos 6



Eliminar Línea existente

Para realizar esta operación realizar un clic izquierdo sobre la herramienta con el icono :



El editor, pasará al modo de eliminación. Para eliminar una línea, realizar un clic sobre cualquier punto de la extensión de la misma.

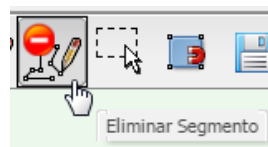
El usuario será alertado sobre que la operación es irreversible.

Si selecciona Sí, la misma será eliminada, de otra manera el proceso será descartado.

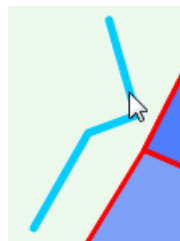
Resumen de pasos:

- 1 - Seleccionar herramienta.
- 2 - Clic izquierdo para eliminar la línea deseada.
- 3 - Seleccione Si, para eliminar la línea , o No, para descartar el proceso
- 4 - Si seleccionó Si deberá Guardar los cambios .

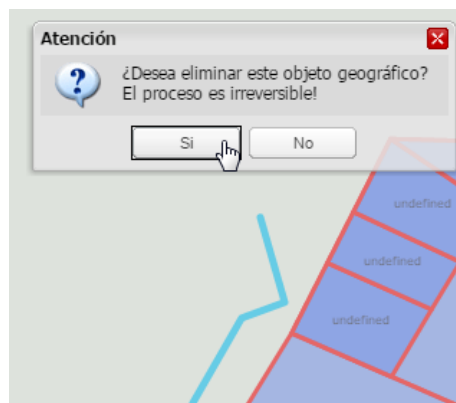
paso 1



paso 2



paso 3 -"Eliminación"



paso 4



Seleccionar Líneas

Esta herramienta permite seleccionar líneas específicas o un conjunto de las mismas, fue implementada para dar solución a requerimientos específicos de productores que deseen realizar operaciones particulares sobre alguna línea o un conjunto de estas. Actualmente la herramienta no posee una funcionalidad específica.

Algunos ejemplos de operaciones sobre líneas que requieran selección podrían ser:

- La selección de un conjunto de líneas para la realización de algún cálculo de topología.
- Generación de un anillo, de forma automática.

Control Snap para edición e inserción

La operatoria es idéntica, a las opciones presentadas en edición de puntos.

El Snap es aplicable tanto, para los vértices existentes como para los nuevos, y puede ser relativo a cualquiera de las capas vectoriales disponibles en el editor.

Edición de Polígonos:

Estas Geometrías son áreas cerradas (figuras de muchos lados) que representan la forma y la ubicación de entidades homogéneas.

Algunos ejemplos pueden ser:

- Parcelas.
- Manzanas.
- Lagunas.
- Región Sembrada.

Para realizar la edición de una capa de polígonos, primero se debe iniciar la edición de la misma haciendo clic izquierdo sobre la capa a editar FIG 19.

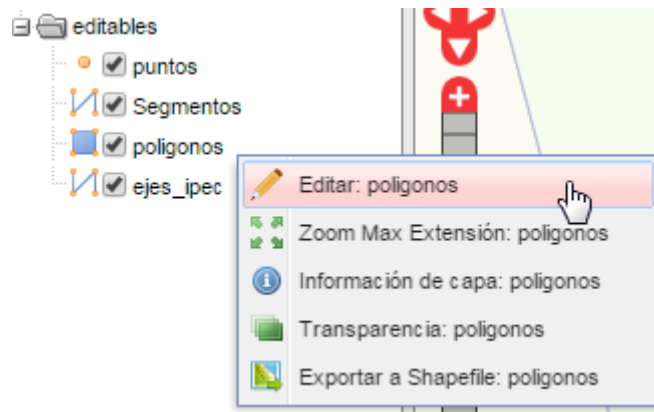


FIG 19

Si el mapa se encuentra en los límites de escala mínimos definidos para esa capa, se habilitará la barra de edición en la parte superior del panel central y se dibujara la región permitida sobre el mapa , sino, se informará mediante un mensaje de error FIG 20

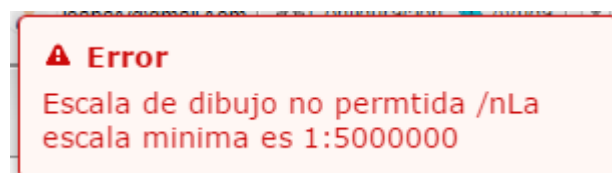


FIG 20

Si la escala es correcta y se ingreso al modo de edición veremos algo similar a lo mostrado en la FIG 21.

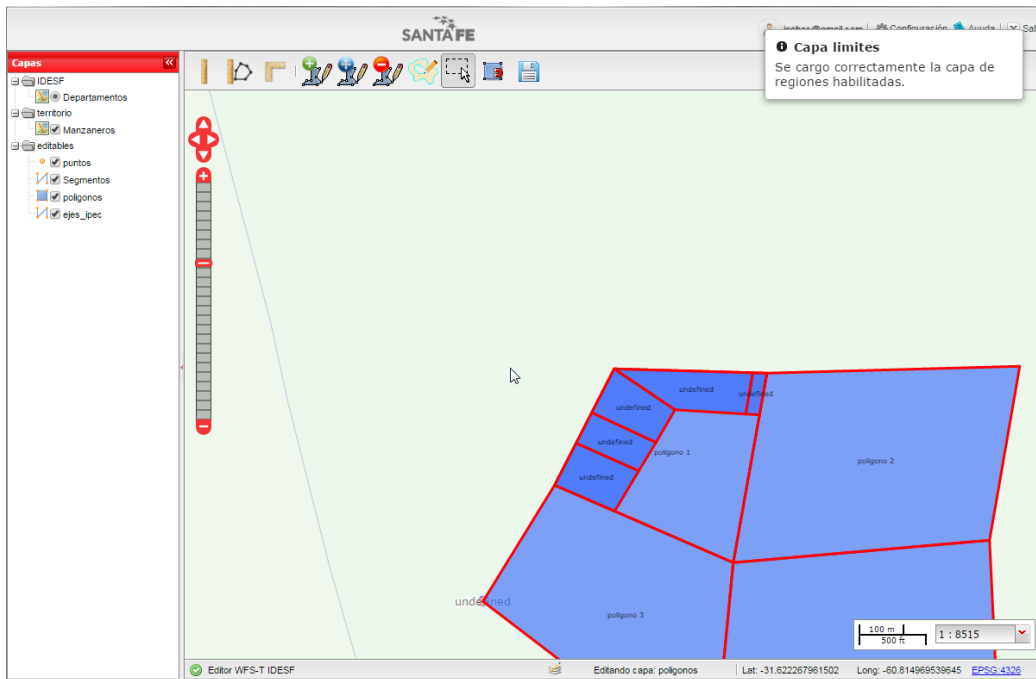


FIG 21

En la barra de herramientas se presentarán los siguientes iconos:

	Insertar nuevos polígono
	Modificar polígono existente
	Eliminar polígono existente
	Dibujar hueco en polígono
	Seleccionar Polígonos
	Control Snap para edición e inserción
	Salvar Cambios y estilos

Insertar nuevo Polígono

Para realizar esta operación bastará con hacer un clic izquierdo sobre la herramienta con el icono :



El editor, pasará al modo de inserción y bastará con posicionar el cursor del mouse sobre la posición inicial deseada y realizar un clic izquierdo.

El punto inicial quedará dibujado y un segmento adherido al cursor del mouse se visualizará y podrá desplazarse visualizando su ubicación, lo que habilita el posicionamiento del mismo, una vez posicionado el segmento en ángulo y longitud y realizando un clic izquierdo se genera un vértice nuevo desde el cual se podrá nuevamente extender un segmento para cerrar el polígono, esta operación podrá realizarse tantas veces como vértices sean necesarios, **para finalizar la edición bastará con realizar un doble clic**, finalizada la edición el polígono quedará dibujado con el estilo que tenga definido el usuario para la capa en edición, y finalmente se deberá persistir los cambios haciendo clic izquierdo en el icono del disquete .

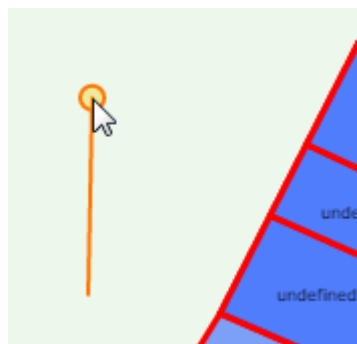
Resumen de pasos: (polígono de 3 vértices)

- 1 - Seleccionar herramienta.
- 2 - Posicionar puntero de mouse en posición inicial.
- 3 - Clic izquierdo.
- 4 - Posicionar segundo vértice y realizar un clic izquierdo
- 5 - Posicionar el cursor en la posición seleccionada para el tercer vértice , y realizar un doble clic, de modo que quede cerrado el polígono.
- 6 - Guardar cambios

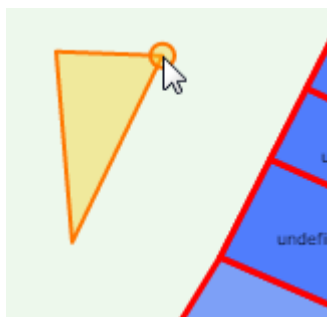
paso 1



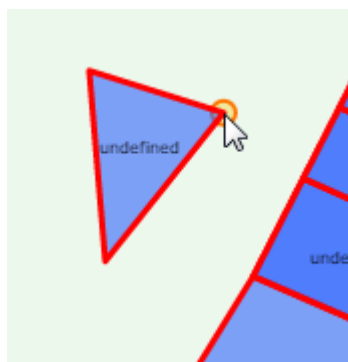
paso 2 y 3



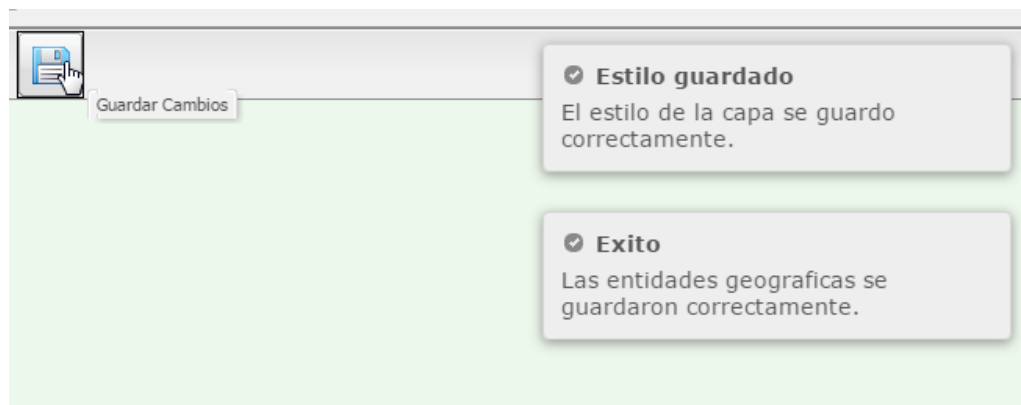
paso 4



paso 5



paso 6



Al finalizar el proceso se informará que los cambios se guardaron correctamente o se mostrará un mensaje de error indicando el error detectado.

Modificar Polígono existente

Para realizar esta operación se debe hacer un clic izquierdo sobre la herramienta con el icono :



El editor, pasará al modo de edición , se debe entonces seleccionar el polígono que se desea modificar, haciendo un clic izquierdo sobre el mismo.

El polígono se visualizará con un estilo diferente, que representa el estado de, seleccionado para edición.

Sobre los polígono podremos, modificar la posición geográfica de cualquiera de sus vértices o crear nuevos quebrando los segmentos desde su punto medio.

Para modificar un vértice la operación es similar a la edición de un punto o una línea, se debe posicionar el cursor sobre el vértice a mover y manteniendo presionado el botón izquierdo del mouse, desplazar el vértice y liberar el botón una vez re posicionado el mismo.

La operación de creación de nuevos vértices es idéntica a la descrita para los segmentos, se debe seleccionar el punto medio de un segmento y desplazar el mismo hasta la posición deseada.

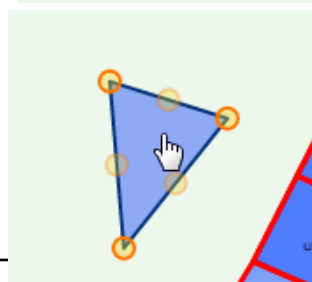
Resumen de pasos: (desplazar un vértice y agregar uno nuevo)

- 1 - Seleccionar herramienta.
- 2 - Seleccionar polígono a editar.
- 3 - Posicionar cursor sobre el vértice a editar y realizando un clic izquierdo sin liberar el botón, posicionar el vértice y liberar entonces el botón.
- 4 - Seleccionar Polígono
- 5 - Posicionar cursor sobre punto medio de segmento a quebrar y realizar un clic sin liberar el botón, posicionar el vértice nuevo en la ubicación deseada y liberar botón.
- 6 - Guardar los cambios

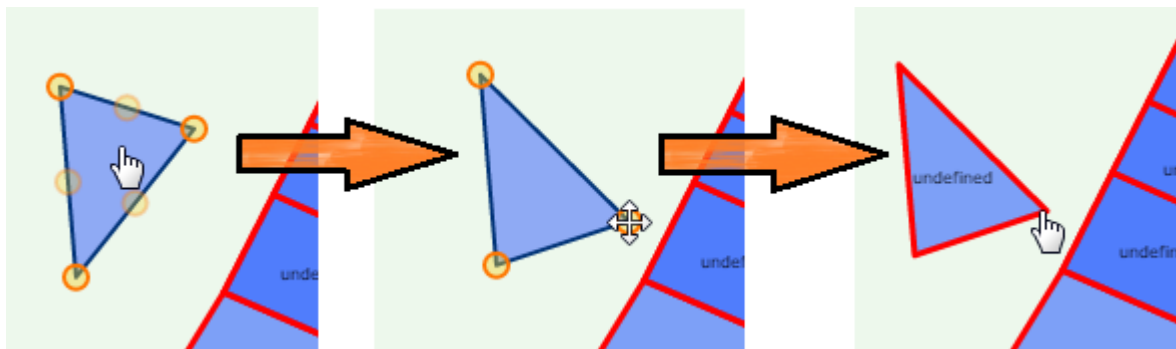
paso 1



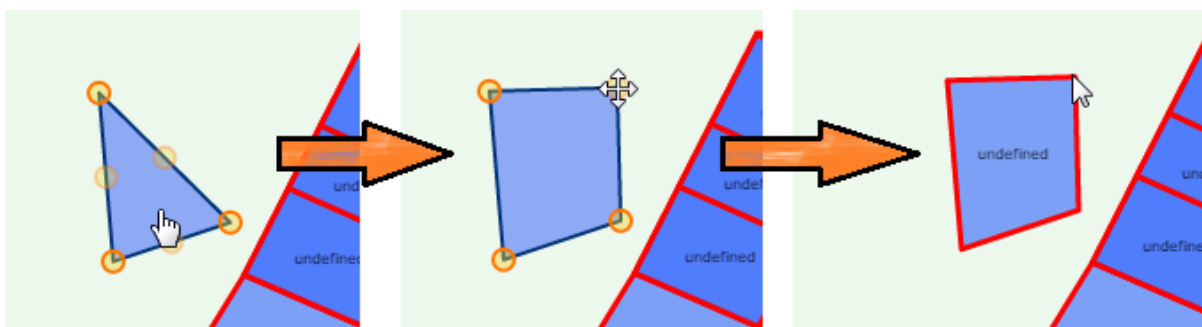
paso 2



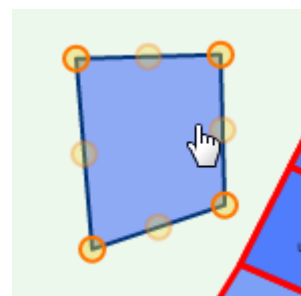
passo 3



passo 4 y 5 - "Nuevo vértice"



NOTA: El polígono ahora contará con un nuevo vértice y en consecuencia un nuevo segmento.



passo 6

Guardar Cambios

- ✔ **Estilo guardado**
El estilo de la capa se guardo correctamente.
- ✔ **Exito**
Las entidades geograficas se guardaron correctamente.

Eliminar Polígono existente

Para realizar esta operación realizar un clic izquierdo sobre la herramienta con el icono :



El editor, pasará al modo de eliminación. Para eliminar una polígono, realizar un clic sobre cualquier punto que esté contenido en el área del mismo.

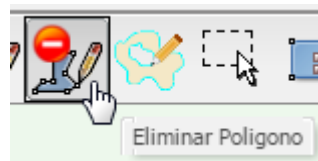
El usuario será alertado sobre que la operación es irreversible.

Si selecciona Sí, la misma será eliminada, de otra manera el proceso será descartado.

Resumen de pasos:

- 1 - Seleccionar herramienta.
- 2 - Clic izquierdo para eliminar el polígono deseado.
- 3 - Seleccione Si, para eliminar, o No, para descartar el proceso
- 4 - Si seleccionó Si deberá Guardar los cambios .

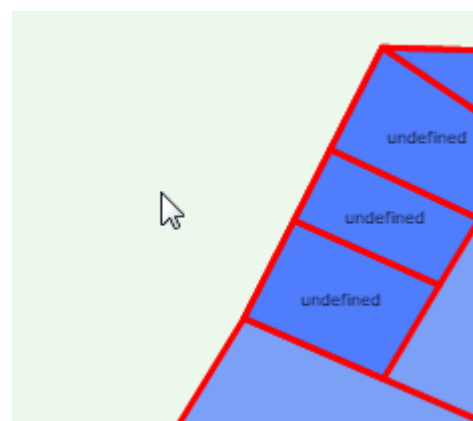
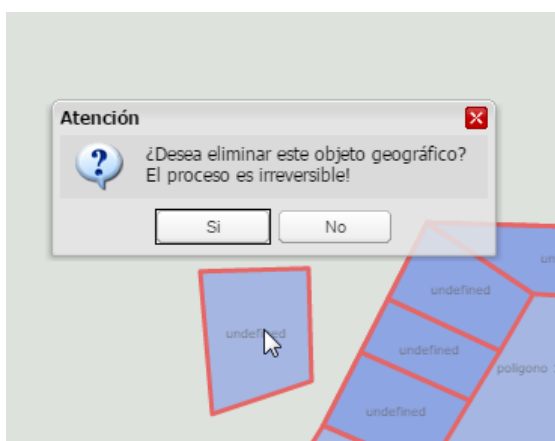
paso 1



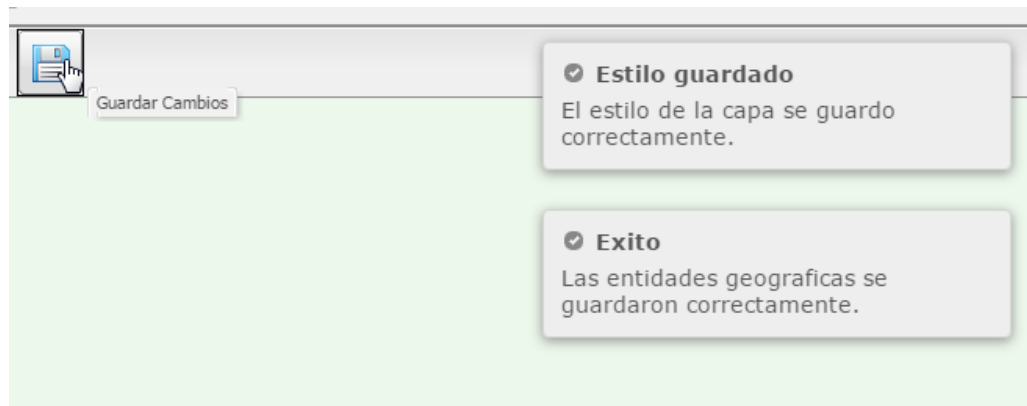
paso 2



paso 3 -"Eliminación"



paso 4 - "solo si se seleccionó Si"



Seleccionar Polígonos

Esta herramienta permite seleccionar polígonos específicos o un conjunto de estos, fue incorporada para dar solución a requerimientos específicos de productores que deseen realizar operaciones particulares sobre algún polígono o un conjunto de estos.

Actualmente la herramienta está asociada sólo con la generación de huecos, pudiendo estar relacionada con un sin fin de otras funcionalidades.

Algunos ejemplos de operaciones sobre polígonos que requieran selección podrían ser:

- Uniones, Intersecciones, restas.
- Cálculos topológicos; pertenencia de otras geometrías, segmentos o vértices en Contacto, entre otras.

Control Snap para edición e inserción

La operatoria es idéntica, a las opciones presentadas en edición de puntos y líneas.

El Snap es aplicable tanto, para los vértices existentes como para los nuevos generados, y puede ser relativo a cualquiera de las capas vectoriales disponibles en el editor.

Generación de Huecos

Para realizar esta operación, deberemos primero, dibujar un polígono de corte sobre el polígono al cual deseamos generar un hueco.

Luego y en un orden específico se deberá seleccionar:

1. Polígono sobre el cual se generará el hueco.



2. Presionando tecla Shift , hacer clic sobre el polígono de referencia para el Corte

Una vez seleccionados estos hacer clic en la herramienta de generación de hueco.



El editor Cortara el polígono de base, tomando la referencia el de corte y dibujara el nuevo con el hueco.

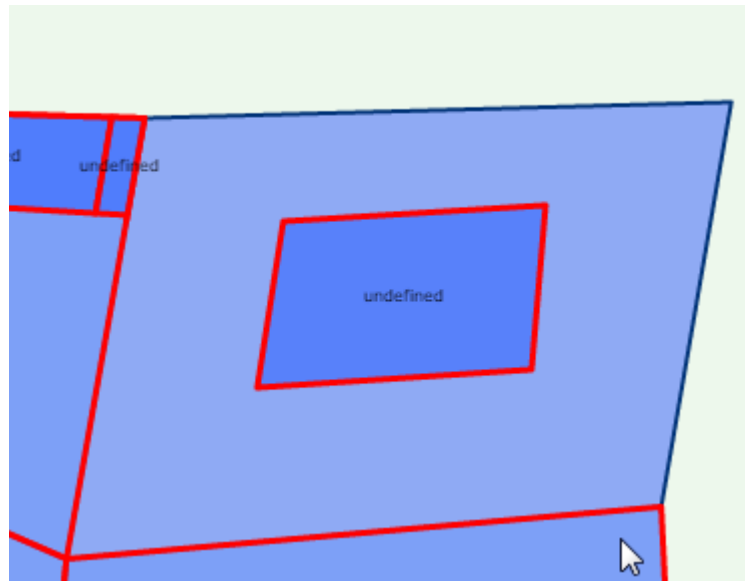
Resumen de pasos:

- 1 - Dibujar polígono de referencia para corte
- 2 - Seleccionar polígono base (polígono a ser cortado).
- 3 - Seleccionar polígono de referencia para el corte
- 4 - Clic izquierdo en herramienta de corte.
- 5 - Guardar Cambios.

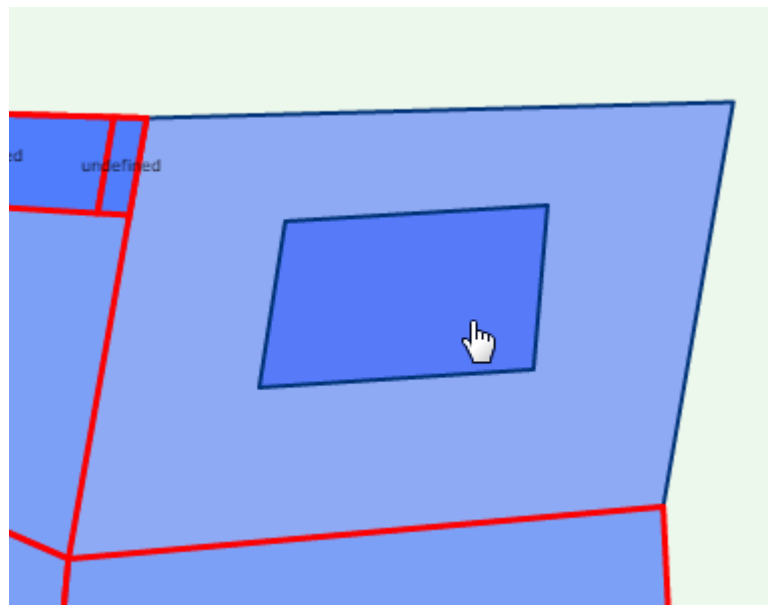
paso 1



paso 2



paso 3



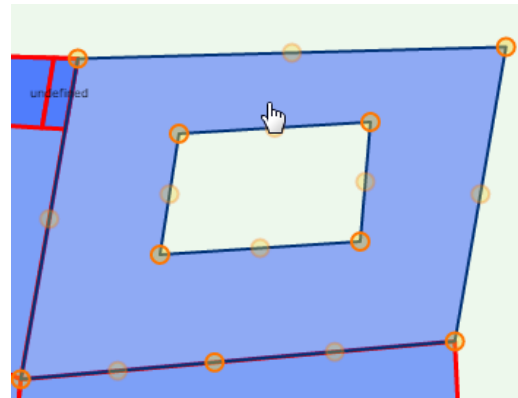
paso 4 - "clic en herramienta de hueco"



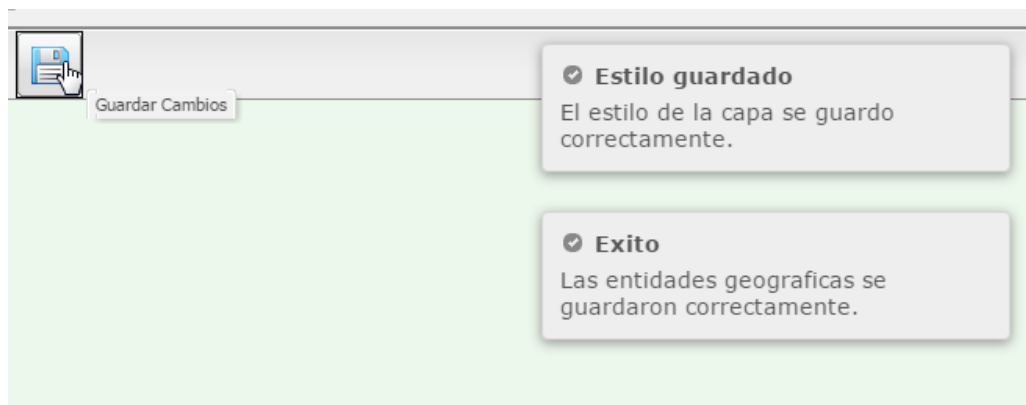
Resultado: Polígono con hueco



Ahora el polígono cuenta con nuevos vértices y segmentos editables en su interior






paso 5



Herramientas adicionales

En la barra de herramientas se ofrecen tres herramientas adicionales, de gran utilidad para la evaluación de los elementos graficados o por graficar:

	Medición por segmentos.
	Medición por área.
	Medir caja contenedora.

Medición por segmentos

Seleccionar la herramienta de medición la cual está disponible tanto en modo de edición, como en modo de visualización:



Realizar un clic izquierdo en el punto inicial de medición, extender el segmento que se presenta adherido al cursor del mouse hasta el punto que se desea incluir en la medición, e ir realizando un clic para cada punto que se desee utilizar como referencia de medición para el segmento, cuando se desea finalizar la medición realizar un doble clic izquierdo. el resultado se presentará en la barra de estado en la parte inferior del panel central, expresando la longitud medida en m o Km según la escala lo amerite.

Resumen de pasos: (ejemplo de medición con 3 segmentos)

- 1 - Seleccionar herramienta de medición.
- 2 - Clic en punto inicial de medición, extender hasta próximo punto.
- 3 - Clic izquierdo, y repetir para el tercer segmento de medición
- 4 - Doble clic finaliza medición.

paso 1



paso 2



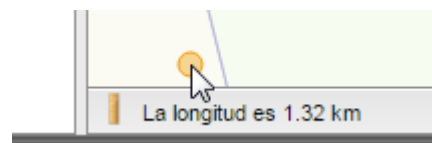
paso 3



paso 4



La longitud obtenida en este ejemplo es de 1.32 Km.



Medición por Área

Seleccionar la herramienta de medición de áreas la misma está disponible tanto en modo de edición, como en modo de visualización:



Realizar un clic izquierdo en el punto inicial de medición, extender el segmento que se presenta adherido al cursor del mouse hasta el punto del polígono de medición que se desea incluir, luego extender el nuevo segmento y realizar un clic para definir cada vértice, cuando se desea finalizar la medición realizar un doble clic izquierdo. el resultado se presentará en la barra de estado en la parte inferior del panel central, expresando la longitud medida en m o Km según la escala lo amerite.

Resumen de pasos: (ejemplo de medición con 4 segmentos o 4 vértices)

- 1 - Seleccionar herramienta de medición.
- 2 - Clic en punto inicial de medición, extender hasta próximo punto.
- 3 - Clic izquierdo, y repetir para el tercer vértice.
- 4 - Doble clic finaliza medición.



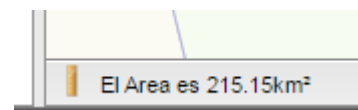
paso 3



paso 4



El área medida es en presente ejemplo es de 215.15 Km².



Medir caja contenedora.


Esta característica nos permitirá calcular de forma automatizada la caja contenedora para un área seleccionada, obteniendo en formato de cadena de texto separada por comas “,” las latitudes y longitudes de las esquinas inferior izquierda y superior derecha FIG 22.



FIG 22

Para realizar esta medición seleccionar la herramienta de medición de caja contenedora:



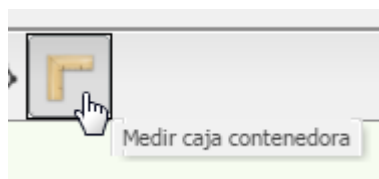
Presionando la tecla Ctrl  marcar la esquina manteniendo presionado el botón izquierdo del mouse, extender la caja que se dibuja sobre el mapa, hasta cubrir el área que se desea calcular, y sin liberar la tecla Ctrl , liberar el botón del mouse.

Se mostrará una alerta en pantalla desde la cual se podrá copiar o anotar las coordenadas de los puntos.

Resumen de pasos:

- 1 - Seleccionar herramienta de Cálculo de caja Contenedora.
- 2 - Mantener presionada tecla Ctrl, y realizar Clic en punto inicial de medición, extender caja hasta próximo punto.
- 3 - Sin liberar tecla Ctrl, liberar botón del mouse.
- 4 - utilizar combinación de teclas Ctrl + C para copiar resultado

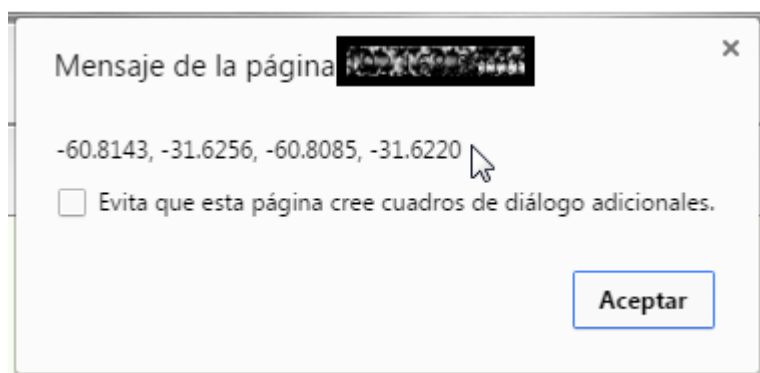
paso 1



paso 2



paso 3



paso 4 - "Coordenadas Obtenidas"

-60.8146, -31.6259, -60.8080, -31.6220

Panel Superior

Edición de Perfil

Cada usuario cuenta con un conjunto de datos básicos que pueden ser editados.

Para llevar a cabo esta tarea se debe hacer clic sobre el avatar presentado en la barra superior FIG 23.



FIG 23

El editor presentará una ventana de configuración básica de usuario FIG24.

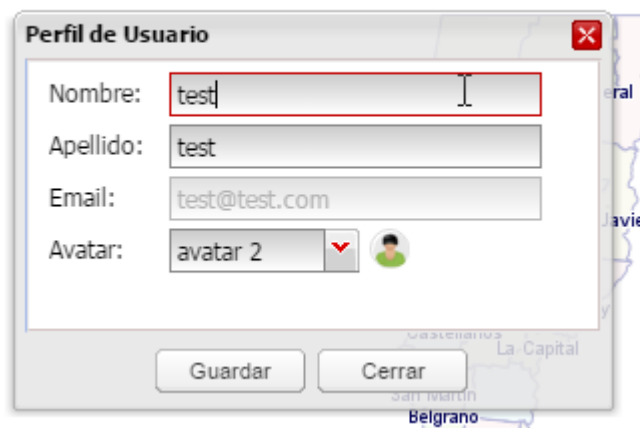


FIG 24

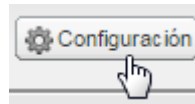
Actualmente este perfil permite solo configurar el nombre y apellido del usuario y su avatar asociado, dentro de un grupo de 18 imágenes fijas.

La edición sobre los campos es directa se modifica su contenido y se debe seleccionar Guardar para confirmar los cambios, de otro modo cualquier cambio ser descartado.

Configuración

Este panel actualmente ofrece la configuración de la persistencia automática de la capa en edición.

Para acceder al mismo se debe realizar un clic izquierdo sobre el botón con el icono del engranaje:



El panel presentado cuenta con una solapa principal, donde en caso de estar en modo de visualización, se visualizará deshabilitada FIG 25.



FIG 25

Si se está en modo de edición, este panel se mostrará activo FIG 26, pudiendo habilitar el auto guardado haciendo clic sobre la caja de selección con la etiqueta Autoguardar, y configurando entonces el intervalo de tiempo en segundos para realizar esta operación.

Para cada procedimiento de guardado se informará el éxito o fracaso de la operación con mensajes emergentes , **“no bloqueantes”**

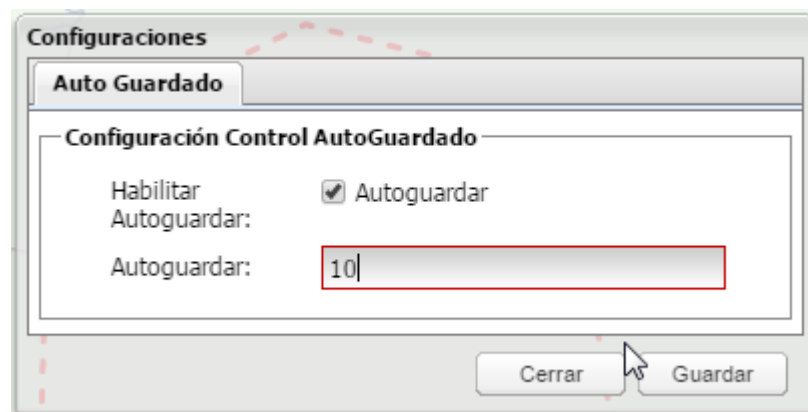


FIG 26

Se debe tener en cuenta que el proceso de autoguardado solo persiste los cambios realizados sobre las geometrías de la capa en edición.

Para persistir los cambios en los estilos asociados a la capa en edición, se debe realizar un guardado manual , mayores referencias en sección “Edición de Estilos”

Ayuda

Se lanza una solapa nueva con el presente documento de ayuda.

Salir

Finaliza la sesión del usuario y regresa al panel de autenticación, se informará al usuario que el navegador será refrescado perdiendo todos los cambios que no se hayan persistido ya sea manual o automáticamente.

Otras opciones de menú Contextual:

Información de Capa:

Al seleccionar la opción Información de capa FIG 27

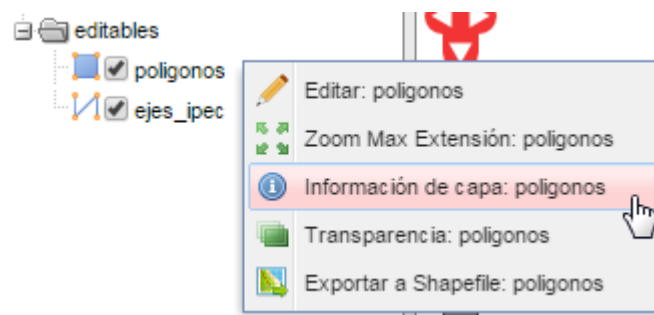


FIG 27

El editor presentará un panel con información básica sobre la capa FIG28.

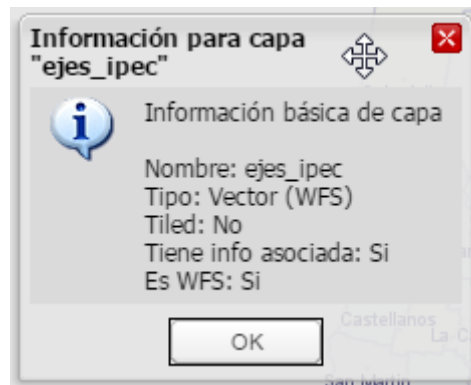


FIG 28

Transparencia

Esta opción del menú habilitará un control deslizante que permitirá modificar la transparencia de la capa seleccionada FIG 29.

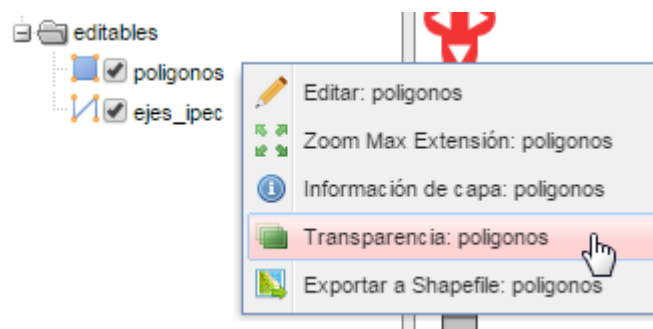


FIG 29

Ejemplo de transparencia configurada al 65 % . FIG 30

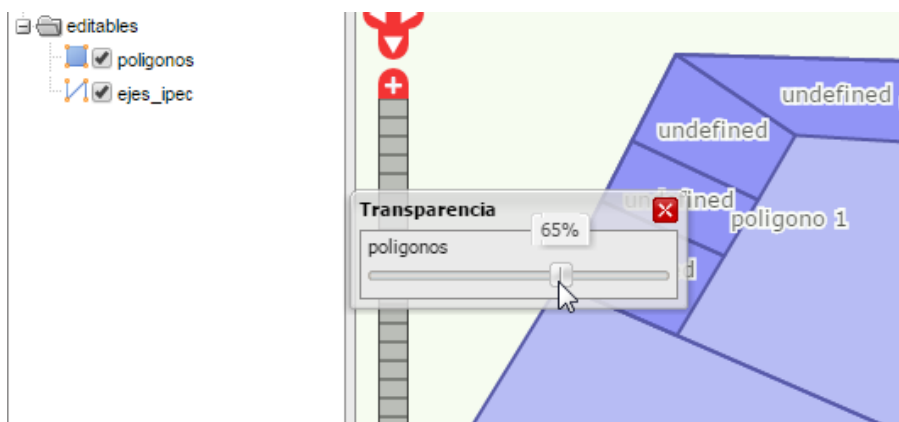


FIG 30

Exportar a Shapefile²²

Esta opción del menú contextual, inicia el proceso de exportación hacia un archivo externo de la capa seleccionada FIG 31.

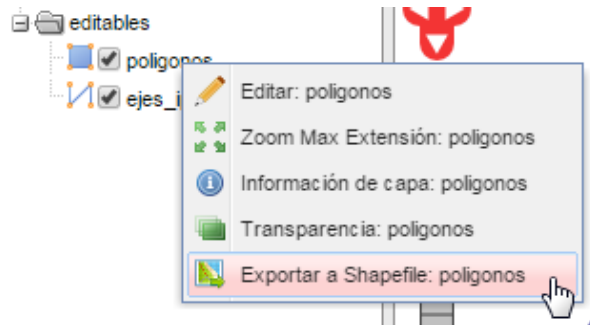


FIG 31

Serán exportados solo los elementos que estén guardados , todo cambio realizado que no se haya guardado no será incluido en el archivo exportado.

El archivo obtenido será descargado por el navegador en formato ZIP ²³.

Este contendrá 6 archivos separados, de los cuales 5 definen según el estándar para archivos shp la capa exportada (geometrías y atributos).

- **.shp** - es el archivo que almacena las entidades geométricas de los objetos.
- **.shx** - es el archivo que almacena el índice de las entidades geométricas.
- **.dbf** - es la base de datos, donde se almacena la información de los atributos de los objetos.
- **.prj** - Es el archivo que guarda la información referida al sistema de coordenadas en formato WKT²⁴.
- **.txt** - Es el archivo que almacena la consulta realizada para recuperar el presente archivo.
- **.cst** - Archivo que contiene la información de la codificación de caracteres de los atributos de la presente capa.

²² Un shapefile es un formato vectorial de almacenamiento digital donde se guarda la localización de los elementos geográficos y los atributos asociados a ellos

²³ En informática, **ZIP** o **zip** es un formato de compresión sin pérdida, muy utilizado para la compresión de datos como documentos, imágenes o programas.

²⁴ La representación **Well Known Text** o de *texto conocido* (también llamado **WKT** en su acrónimo inglés) es una codificación o sintaxis en formato ASCII estandarizada diseñada para describir objetos espaciales expresados de forma vectorial.