

DESARROLLO DE UN SISTEMA DE INFORMACIÓN GEOESPACIAL CON USO DE SOFTWARE LIBRE.¹

Development of a Geospatial Information System using free software

Xavier Bustos Catari

RESUMEN

Desde hace mas de 15 años, las geotecnologías han provisto a los investigadores del Centro de Estudios Integrales del Ambiente (CENAMB) de la Universidad Central de Venezuela (UCV) de un apoyo fundamental en sus labores de docencia, investigación y extensión; por lo que, en el presente estudio se desarrolla un sistema geomático orientado a objetos siguiendo el Proceso Unificado de Desarrollo de Software e implementado con el uso de software libre como alternativa viable al software privativo. En este trabajo se identifican, también, las potencialidades y limitaciones encontradas en el uso de las herramientas libres y se describe un procedimiento para publicar la información espacial en entorno Web.

Palabras clave: Sistema, tecnología, software libre, información, proceso, geoespacial.

ABSTRACT

Since more than 15 years, the geo-technologies have provided to the researchers of the Environmental Integral Studies Center (CENAMB) of the Central University of Venezuela (UCV), a strong support in the areas of teaching, researching and services. In this study an object-oriented geomatic system is developed following the methodology of Rational Unified Process (RUP) and implemented using free software as a viable alternative to the proprietary software. In this work also the potentialities and limitations concerning the use of the free software tools are identified, and a procedure to publish spatial information on the web is described.

Key words: System, technology, free software, information, process, geospatial

¹ Recibido:10/04/2012
Aceptado:12/11/2013

INTRODUCCIÓN

Podemos definir un Sistema de Información Geográfica (SIG) como un sistema completo, formado por cinco componentes (personal capacitado, datos espaciales y descriptivos, métodos analíticos, hardware, software) (Zeiler, 1999), para el apoyo fundamental en áreas como la planificación ambiental, el análisis espacial, la producción cartográfica digital, entre otras. Tradicionalmente, el componente software de estos sistemas ha sido con licencia propietaria y entre los más utilizados con este tipo de licencias se cuentan el ArcGis de Esri y el Mapinfo de Pitney Bowes Software Inc. Con la aparición del software libre para computadores personales, algunos programas para desarrollo de SIG comienzan a ser liberados bajo este tipo de licencias y puestos a disposición de la comunidad de usuarios interesados; tal es el caso del programa Grass GIS y más recientemente los programas Quantum Gis y GvSig.

En el campo de las bases de datos estos desarrollos no se han quedado atrás y también contamos con Sistemas Gestores de Bases de Datos (SGBD), libres confiables y robustos (PGSQL, 2012), como es el caso de PostgreSQL que, conjuntamente con la extensión PostGis, permite a usuarios y desarrolladores la posibilidad de crear bases de bases de datos espaciales.

Paralelamente a todos estos desarrollos, no debemos obviar la posibilidad de trabajar conectados a la Web y explotar las ventajas existentes para consultar mapas o imágenes de sensores remotos dispuestos gratuitamente en la red, como por ejemplo, a través de los Servicios de Mapas para Web (Web Map Service), que permiten consultar información georeferenciada almacenada en servidores remotos.

En el presente trabajo, se plantean los pasos para la construcción de un sistema geoespacial utilizando la combinación de algunas de las herramientas con licencia libre, antes mencionadas y siguiendo lineamientos del Proceso Unificado de Desarrollo de Software. Al finalizar, se realizan algunas reflexiones sobre el uso de herramientas libres empleadas en el proceso de construcción.

EL SOFTWARE LIBRE Y LA GEOMÁTICA CON SOFTWARE LIBRE.

El término Software Libre se refiere, según la Fundación de Software Libre (FSF, 2009) a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades que se les permite a los usuarios del software:

- **Libertad 0:** “Usar el programa con cualquier propósito”. Se nos da la potestad de utilizarlo con cualquier fin, ya sea educativo, social, comercial, etc.
- **Libertad 1:** “Estudiar cómo funciona el programa, y adaptarlo a tus necesidades”. Implica que podemos estudiar su funcionamiento lo que nos permite entre otras cosas averiguar cómo se realizan tareas específicas, descubrir funciones ocultas, etc.
- **Libertad 2:** “Distribuir copias”; es permisible redistribuir el programa a quien queramos.
- **Libertad 3:** “Mejorar el programa y hacerlo público”. Libertad de hacer mejor el programa (para ello es necesario el código fuente), hacerlo más eficiente y redistribuir dichas mejoras a los demás, de modo que toda la comunidad se beneficie.

TIPOS DE LICENCIA LIBRES UTILIZADAS EN ESTE DESARROLLO.

Existen múltiples licencias libres en la actualidad; en esta sección se describen las utilizadas en el trabajo presentado.

La Licencia Pública General (GPL)

Mejor conocida por GNU/GPL (General Public License), es una licencia creada por Richard Stallman, a través de la Fundación de Software Libre (Free Software Foundation) a mediados de los 80. Está orientada principalmente a proteger la libre distribución, modificación y uso de software; su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación por manos inescrupulosas que restrinjan las libertades descritas anteriormente. (APESOL, 2005).

El proyecto GNU fue iniciado por Stallman con el objetivo de crear un sistema operativo completamente libre. Al anuncio original, siguieron otros ensayos escritos por Stallman como el "Manifiesto GNU", que establecieron sus motivaciones para realizar dicho proyecto, entre las que destaca "volver al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras". (APESOL, 2005).

La licencia GPL obliga a incluir el código fuente en su distribución, siendo imposible cambiar la licencia al programa, al distribuirlo tal cual o modificado.

La Distribución de Software Berkeley (Berkeley Software Distribution)

Según Lehey (2008), la Berkeley Software Distribution (BSD) se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California, en Berkeley, Estados Unidos.

En los primeros años del sistema Unix, sus creadores (los Laboratorios Bell de la compañía AT&T), autorizaron a la Universidad de California en Berkeley y a otras universidades a utilizar el código fuente y adaptarlo a sus necesidades. Durante la década de los setenta y los ochenta, Berkeley utilizó el sistema para sus investigaciones en materia de sistemas operativos. Cuando AT&T retiró el permiso de uso a la universidad por motivos comerciales, la universidad promovió la creación de una versión inspirada en el sistema Unix, utilizando los aportes que ellos habían realizado y permitiendo luego su distribución con fines académicos. Luego, al cabo de algún tiempo, se redujeron las restricciones para el copiado, la distribución o la modificación de este software.

La licencia BSD, si bien es Software Libre y cumple con las libertades nombradas, se diferencia de la GPL, en que es posible cambiar la licencia.

EL OPEN GEOSPATIAL CONSORTIUM (OGC)

El OGC, creado en 1994, es un consorcio internacional compuesto por compañías, agencias gubernamentales y universidades que participan en el desarrollo de soluciones conceptuales, públicamente disponibles, que pueden ser útiles para todo tipo de aplicaciones que manejen datos espaciales. Su fin es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la World Wide Web. Persigue acuerdos entre las diferentes empresas del sector que posibiliten la inter-operación de sus sistemas de geoprocesamiento y facilitar el intercambio de la información geográfica en beneficio de los usuarios. (OGC, 2010)

EL PROGRAMA QUANTUM GIS (QGIS)

Es un SIG de código libre que se ejecuta en las plataformas Linux, Unix, Mac OS y Microsoft Windows. QGIS está desarrollado utilizando el lenguaje C++, lo que hace que QGIS sea rápido y tenga una interfaz de usuario agradable y fácil de usar. (QGIS, 2010).

El objetivo inicial fue proporcionar un visor de datos SIG. QGIS ha alcanzado este punto en su evolución y está siendo utilizado por muchos para sus necesidades diarias de visualización de datos SIG. QGIS soporta un buen número de formatos ráster y vectoriales, con nuevos soportes fácilmente añadidos utilizando su arquitectura de complementos. QGIS se ha publicado bajo Licencia Pública (GNU General Public License). Desarrollar QGIS bajo esta licencia

implica que se puede inspeccionar y modificar el código fuente y las garantías que se tienen. (QGIS, 2010).

QGIS provee soporte para PostGIS (POGIS, 2010); por lo que, si este último se encuentra instalado, podemos cargar mapas directamente desde una base de datos PostGIS usando el botón "Add PostGIS layer". PostGIS es una extensión espacial para PostgreSQL para almacenar objetos (vector) espaciales.

Razones para seleccionar el SIG Quantum Gis:

- Amplia comunidad de usuarios de este programa y una gran cantidad de documentación en línea; se pueden descargar manuales completos desde sus sitios oficiales.
- Su interfaz gráfica es similar a la programa ArcGis, los usuarios familiarizados con este programa encontrarán funcionalidades similares en los mismos, lo que reducirá el tiempo de aprendizaje.

Rol: Este SIG tiene la función principal de conectarse a la base de datos PostGis; permite la realización de análisis espacial con apoyo del SQL, producir nuevas coberturas, generar salidas cartográficas, etc. Con este programa se puede generar los archivos MapFile; o que, conjuntamente con una aplicación desarrollada en código PHP/MapScript, crea el ambiente requerido para publicar mapas mediante un navegador Web.

EL MANEJADOR DE BASES DE DATOS OBJETO-RELACIONAL POSTGRESQL

Según el Grupo de Desarrollo Global de PostgreSQL (PGSQL, 2012), esta herramienta es un sistema gestor (manejador) de bases de datos objeto- relacional. Fue desarrollado por el Departamento de Ciencias de la Computación de la Universidad de Berkeley en California. PosgreSQL fue pionero en muchos conceptos que estuvieron disponibles en algunos sistemas de bases de datos comerciales mucho después.

De acuerdo con su sitio oficial (PGSQL, 2012), PostgreSQL implementa las características necesarias para competir con cualquier otra base de datos comercial, con la ventaja de tener una licencia de libre distribución BSD. La migración de bases de datos alojadas en productos comerciales hacia PostgreSQL se facilita, gracias a que soporta ampliamente el estándar SQL. PostgreSQL cuenta con una serie de características atractivas, como son la herencia de tablas (clases) y, un conjunto amplio de tipos de datos que incluyen arreglos, Binary

Large Object Block (BLOB), tipos geométricos y direcciones de red. PostgreSQL incluye también, el procesamiento de transacciones, integridad referencial y procedimientos almacenados.

POSTGIS. UNA EXTENSIÓN GEOESPACIAL PARA POSTGRESQL

POSTGIS es una extensión del sistema de base de datos PostgreSQL, que permite el almacenamiento y la manipulación de objetos geométricos vectoriales en bases de datos; incluyendo además, el Sistema de Referencia Espacial; lo que la convierte en una base de datos espacial para su utilización en Sistema de Información Geográfica. (Duke-Williams, 2006).

PostGIS ha sido desarrollado por la empresa canadiense Refraction Research Inc, especializada en productos "Open Source", como un proyecto de investigación de tecnologías de bases de datos espaciales. Está publicado bajo licencia GNU/GPL.

Un aspecto digno de tener en cuenta es que PostGIS ha sido certificado, en 2006 por el Open Geospatial Consortium (OGC), lo que garantiza la interoperabilidad con otros sistemas. PostGIS almacena la información geográfica en una columna del tipo GEOMETRY (diferente del homónimo utilizado por PostgreSQL).

La especificación OpenGis define dos formas estándares de expresar los objetos espaciales; estos son los formatos en Modo Texto (Well-Known Text -WKT) y en Modo Binario(Well-Known Binary-WKB). Ambos incluyen la información sobre el tipo de objeto y las coordenadas que conforman el objeto.

Razones para seleccionar el binomio PostgreSql y Postgis

- Se cataloga como el manejador de base de datos objeto-relacional libre más robusto existente en la actualidad, (PGSQL,2012).
- Soporte de datos espaciales y proyecciones a través de la extensión PostGis.
- Interoperabilidad con los programas Qgis, gvSIG, Grass, etc.; es decir la información espacial almacenada en la base de datos espaciales puede ser utilizada desde los programas nombrados.

Rol: Gestor de la base de datos espaciales a crear, en donde se almacena la información espacial y atributiva de los proyectos.

MAPSERVER

MapServer es una plataforma de desarrollo de código abierto (Open Source) creado a mediados de los 90 en la Universidad de Minnesota (USA), con el fin de publicar data espacial y aplicaciones de mapas interactivos en Internet/Intranet, mediante la tecnología IMS (Internet Map Server).

Se ejecuta sobre varios sistemas operativos (ej. Windows, Linux), permite la renderización de alta calidad y consulta de mapas en formato raster (JPG, GeoTIFF y otros); soporta la visualización y consulta de mapas en formato vectorial ESRI Shapefiles; además, su interfaz gráfica entre servidor y cliente puede ser creada utilizando el lenguaje HTML, combinada con otros como JavaScript, PHP, SQL, etc. Mapserver utiliza su propio lenguaje para el despliegue de capas de información geográfica, leyendas, escalas, simbología y la configuración de éstas (ancho, líneas, color, etiquetado, etc.); las cuales, se definen en el archivo de configuración o MapFile (Mapserver,2010).

Razones para seleccionar la Plataforma Mapserver

- Es conjuntamente con el Geoserver una de las plataformas más utilizada y difundida para publicar, visualizar, consultar mapas digitales utilizando un navegador WEB.
- Interpreta los archivos MapFile generados desde Quantum-Gis o gvSIG.
- Permite la conexión con bases de datos espaciales gestionadas desde el binomio PostGis –PostgreSql.

Rol: Servidor web; permite interactuar y desplegar mapas digitales sobre un visualizador Web de información contenida en una base de datos PostGis.

EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.

Para construir el sistema geoespacial, empleamos como metodología de desarrollo el “Proceso Unificado de Desarrollo de Software” también conocido como RUP por sus siglas en inglés (Rational Unified Process); éste se basa en una integración del trabajo de tres metodologías (I. Jacobson, G.Booch y J.Rumbaugh), quienes fueron reunidos por la corporación Rational para formar un marco de metodologías unificadas, cohesivas y comprensivas de desarrollo de sistemas de software. Su trabajo ha dado a lugar a importantes normas en la comunidad de desarrollo, incluida la aceptación general de los Casos de Uso y del Lenguaje de Modelado Unificado (Unified Modeling Language – UML). (RUP, 1998)

Flujos de Trabajo de la metodología RUP

Los flujos de trabajo desarrollados en el Proceso Unificado y que se utilizarán en la construcción del sistema propuesto, en el presente artículo, se describen a continuación.

Requisitos: Se persigue conformar el modelo de casos de uso; para ello, se “comienza identificando una necesidad desde el punto de vista del usuario y esta surge con un enfoque general para encontrar la solución, tomando en cuenta la tecnología disponible, recursos, enfoques existentes, entre otros” (Jacobson *et al.* 2000; 15). Los casos de uso son un enfoque formal, mediante los cuales se capturan rápidamente los requerimientos del sistema, desde el punto de vista del usuario; también, se indican el propósito del caso de uso, los actores involucrados, el evento inicial y las condiciones de terminación.

Análisis: Se examinan los requerimientos descritos en la fase de captura de requisitos, refinándolos y estructurándolos. El objetivo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos, que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura. (Jacobson *et al.* 2000)

Diseño: Se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos (incluyendo los no funcionales y otras restricciones) pertinentes. Una entrada en el diseño es el resultado del análisis, esto es, el modelo de análisis.

Implementación: En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Sus actividades incluyen la ejecución de la arquitectura, la puesta en práctica de las clases e interfaces, la aplicación de los subsistemas, documentación de software y la realización de unidades de prueba.

Pruebas.

En este flujo de trabajo se verifica el resultado de la implementación, probando cada construcción, lo que incluye, tanto las construcciones internas como las intermedias; así como también, las versiones finales del sistema. Se centra en la descripción de las actividades que se llevarán a cabo, para probar y asegurarse de que el software satisfaga las necesidades del usuario, y que sea fiable, entre otros aspectos clave. (Jacobson *et al.* 2000)

DESARROLLO DEL SISTEMA APLICANDO LA METODOLOGÍA RUP Y UTILIZANDO HERRAMIENTAS LIBRES

A continuación se describe el desarrollo del sistema de acuerdo a los flujos de trabajo del Proceso Unificado comenzando con el modelo de casos de uso y concretándose con el modelo de implementación. Es importante señalar que en la metodología RUP se utilizan conceptos del Enfoque Orientado a Objetos y el Lenguaje Unificado de Modelado (UML)². El siguiente cuadro muestra el proceso secuencial a seguir en el desarrollo; se señala que el producto de cada flujo de trabajo es un modelo que lleva el mismo nombre.

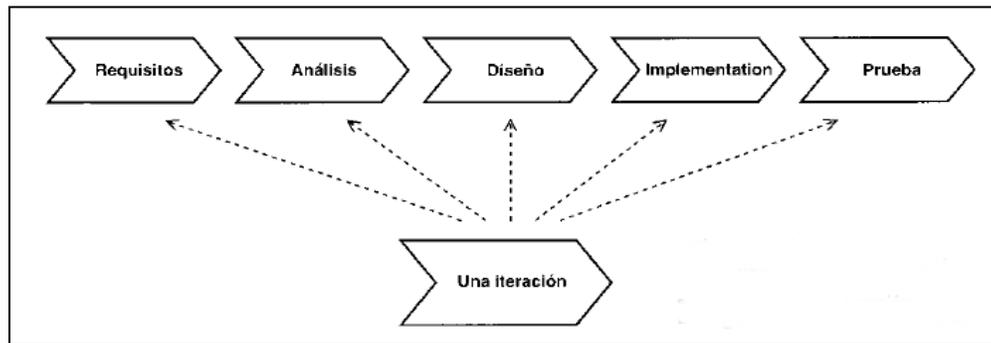


Figura 1. Cada iteración constituye un paso, a través de los cinco flujos de trabajo fundamentales

Fuente: Jacobson et al., 2000

REQUISITOS

Este flujo de trabajo comprende el dominio del problema planteado, y con ello, la realización del modelo de casos de uso.

Modelo de Casos de uso

Los “casos de usos” constituyen una descripción en lenguaje natural de la forma en que se va a operar el sistema. Para los efectos del trabajo que se presenta, los casos de uso fueron conformados, con base en las acciones necesarias para la gestión de datos espaciales generados en la ejecución de proyectos con componentes geomáticos, elaborados por el CENAMB (CENAMB, 2009)

² Ejemplos de aplicaciones, en los que se detallan esos temas, fueron publicados en las siguientes referencias “Una visión policéntrica del ambiente bajo el enfoque orientado a objetos”. (Bustos, 2008), y “Modelo de datos agrarios espaciales bajo el enfoque orientado a objetos (O-O) para el Instituto Nacional de Tierras (INTI). (Castellanos y Bustos, 2011).

Caso de uso 1: Registrar la información de un proyecto de investigación y su información asociada.

Caso de uso 2: Desplegar la información relacionada a los proyectos del CENAMB.

Caso de uso 3: Importar los archivos vectoriales en formato Shapefile asociados a la base de datos espaciales desarrolladas en PostgreSQL y Q-Gis.

Caso de uso 4: Desplegar, consultar, actualizar y representar cartográficamente las clases espaciales registradas en las bases de datos espaciales.

Caso de uso 5: Realizar análisis espacial de los datos almacenados en las bases de datos espaciales.

Caso de uso 6: Consultar información geoespacial de proyectos a través de un visualizador Web.

MODELO DE ANÁLISIS

El modelo de análisis a producir representa el proceso de comprensión y desarrollo de los requisitos ya establecidos en los casos de uso.

Diagrama de Clases

Se concreta con el diagrama de clases (UML), en el que se muestra una estructura de paquetes de análisis y clasificadores (clases espaciales y no espaciales) con sus atributos, y relaciones pertinentes entre las mismas.

Paquetes de análisis para el Proyecto: P_Avila_2009

Para conformar el modelo de análisis, se tomó como referencia la información geoespacial generada durante el desarrollo del proyecto titulado “La Gestión del Conocimiento integrado al Sistema de Información Geográfica como soporte de la Investigación y Gestión del Parque Nacional El Ávila.” (CENAMB, 2009), realizado en este centro de estudios durante los años 2008-2010, y que para efectos de simplificación, lo hemos llamado P_Avila_2009, año en que se generaron la mayoría de las coberturas vectoriales del mencionado estudio.

El Parque Nacional Waraira-Repano (antiguamente El Ávila) se localiza en el tramo central de la Cordillera de la Costa, el cual comprende un área montañosa, abrupta que se interpone entre las costas del Litoral Central y los Valles de Caracas, Guatire, Guarenas y parte de la llanura de Barlovento. Ocupa una superficie de 85.192 Ha; con alturas que varían desde los 120 msnm. a los 2765 msnm. (MARNR, 1992).

Las clases espaciales (coberturas vectoriales) que se generaron durante el mencionado proyecto fueron:

- av_abrae: (Ávila ABRAES)
- av_centros_poblados: (Ávila centros poblados)
- av_cuencas: (Ávila cuencas)
- av_curvas_nivel: (Ávila curvas de nivel)
- av_geologia: (Ávila geología)
- av_geomorfologia: (Ávila geomorfología)
- av_hidrografia: (Ávila hidrografía)
- av_mar (Ávila – y su ubicación respecto al mar Caribe)
- av_municipios: (Ávila municipios)
- av_parroquias_vargas: (Ávila y su relación con las parroquias del Edo. Vargas)
- av_uso_de_la_tierra: (Ávila y usos de la tierra)
- av_vegetacion: (Ávila y cobertura vegetal)
- av_vialidad: (Ávila y vialidad)

A continuación se visualizan algunas de las clases de análisis en notación UML (Unified Modeling Language).

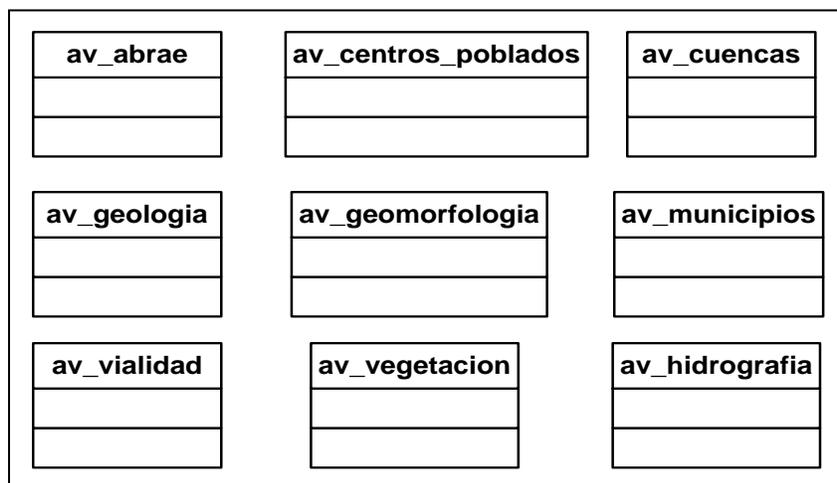


Figura 2. Modelo de Análisis. Ejemplos de clases espaciales del proyecto: P_Avila_2009
Fuente: Elaboración propia.

A continuación se muestran los atributos y métodos de algunas clases espaciales, incluyendo aquellos atributos de naturaleza espacial como son la geometría, sistema de referencia y su área o longitud (atributo que cambia de acuerdo al tipo de geometría).

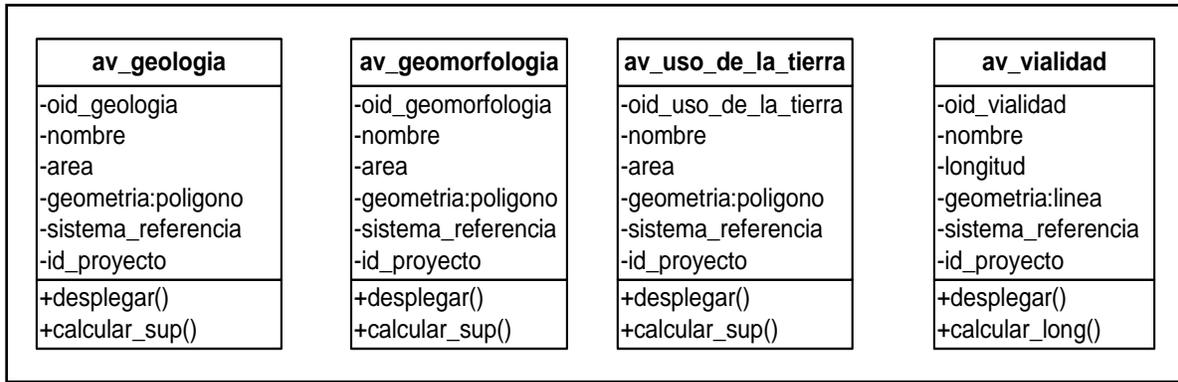


Figura 3. Modelo de Análisis. Clases espaciales con atributos y métodos. Elaboración propia.

Durante el análisis, se relacionan las clases espaciales y no espaciales; a continuación se muestra la clase no espacial Proyecto_c, conectada a las clases espaciales av_vegetacion y av_geomorfologia (notación UML). Las relaciones entre clases se leen de la siguiente manera: En un (1) Proyecto (por ejemplo P_Avila_2009) se generan muchos (*) polígonos de vegetación identificados por un código (oid_vegetacion); y cada polígono de vegetación identificado pertenece a un (1) solo proyecto (en este caso P_Avila_2009). Lectura análoga se realiza con la asociación Proyecto –geomorfología.

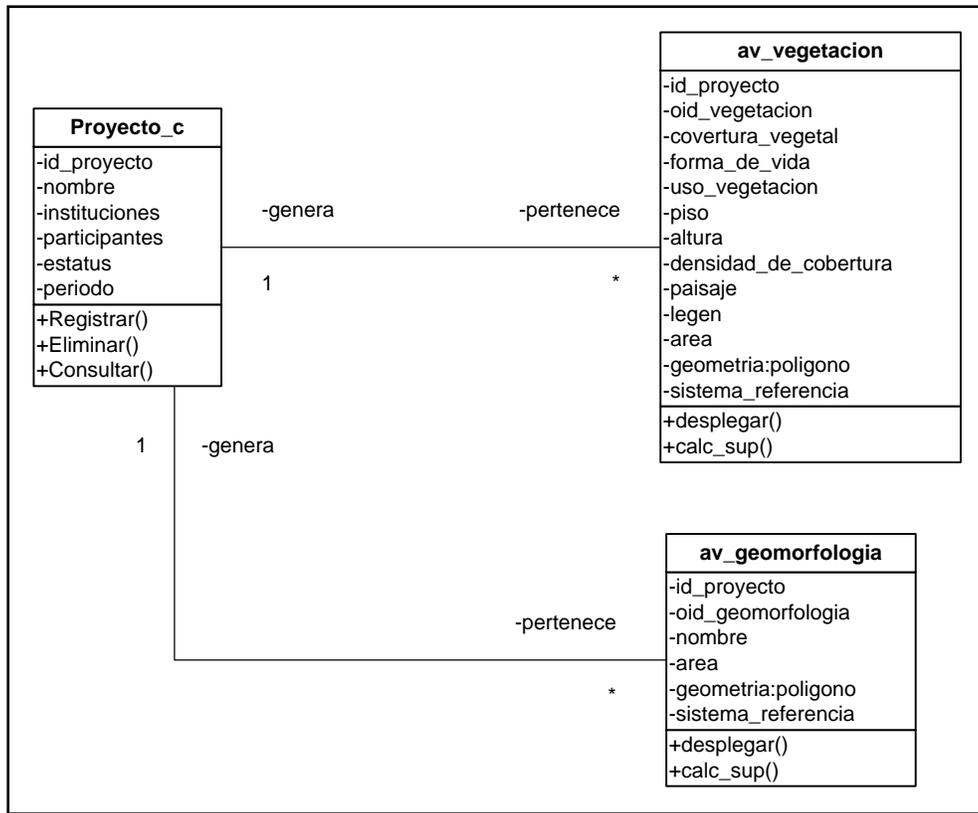


Figura 4. Modelo de Análisis. Relaciones entre clases de análisis. Proyecto P_Avila_2009
Fuente: Elaboración propia.

El proceso de creación y lectura de relaciones es similar para el resto de las clases espaciales restantes, con respecto al proyecto todas las asociaciones son del tipo 1 a varios (1--*) .De manera simplificada (sin atributos) se muestran la relación entre la clase no espacial Proyecto_c y el resto de las clases espaciales generadas para el proyecto P_Avila_2009.

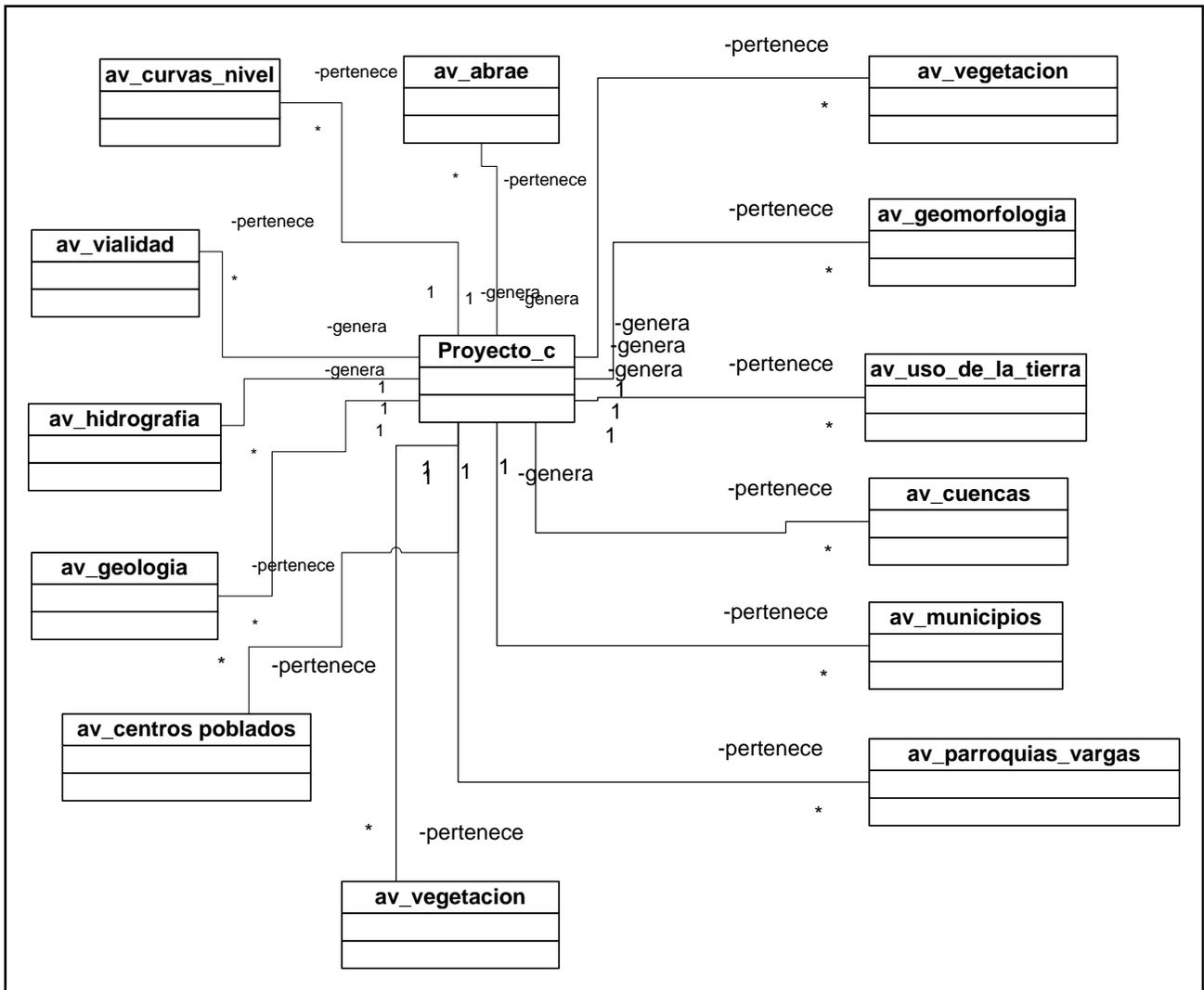


Figura 5. Modelo de Análisis. Relaciones entre clase no espacial Proyecto y clases espaciales
Fuente: Elaboración propia.

MODELO DE DISEÑO

En este modelo se describen las vistas arquitectónicas del sistema y se refina el modelo de análisis, con el uso de una notación común para llegar a un nivel de detalle que permita la implementación del sistema y las pruebas necesarias de la aplicación.

Para gestionar datos espaciales y no espaciales de los proyectos generados en el CENAMB, y conformar un repositorio común, se requirió el SGBD objeto-relacional PostgreSQL con su extensión Postgis; por lo que el diseño está supeditado a la arquitectura provista por dicha herramienta, que provee el soporte para la construcción de las bases de datos espaciales.

Asimismo el mecanismo de interfaz de usuario para visualizar y analizar dichos datos es provisto por la combinación de las plataformas Mapserver y Quantum Gis (Bustos, 2011).

Modelo de Diseño para las clases espaciales del Proyecto P_Ávila_2009.

En el diseño se utilizó la jerarquía de clases para soporte de datos geométricos propuesto por el Consorcio Geoespacial de Código Abierto (OGC por sus siglas en inglés), también llamado Consorcio Open Gis, el cual es tomado por Postgis en su implementación. Se contextualizaron las clases espaciales identificadas en el modelo de análisis dentro del marco (Framework) propuesto por el OGC; por lo que una primera aproximación al Modelo de Diseño para las clases espaciales pertenecientes al proyecto P_Ávila_2009, se muestra en la siguiente figura.

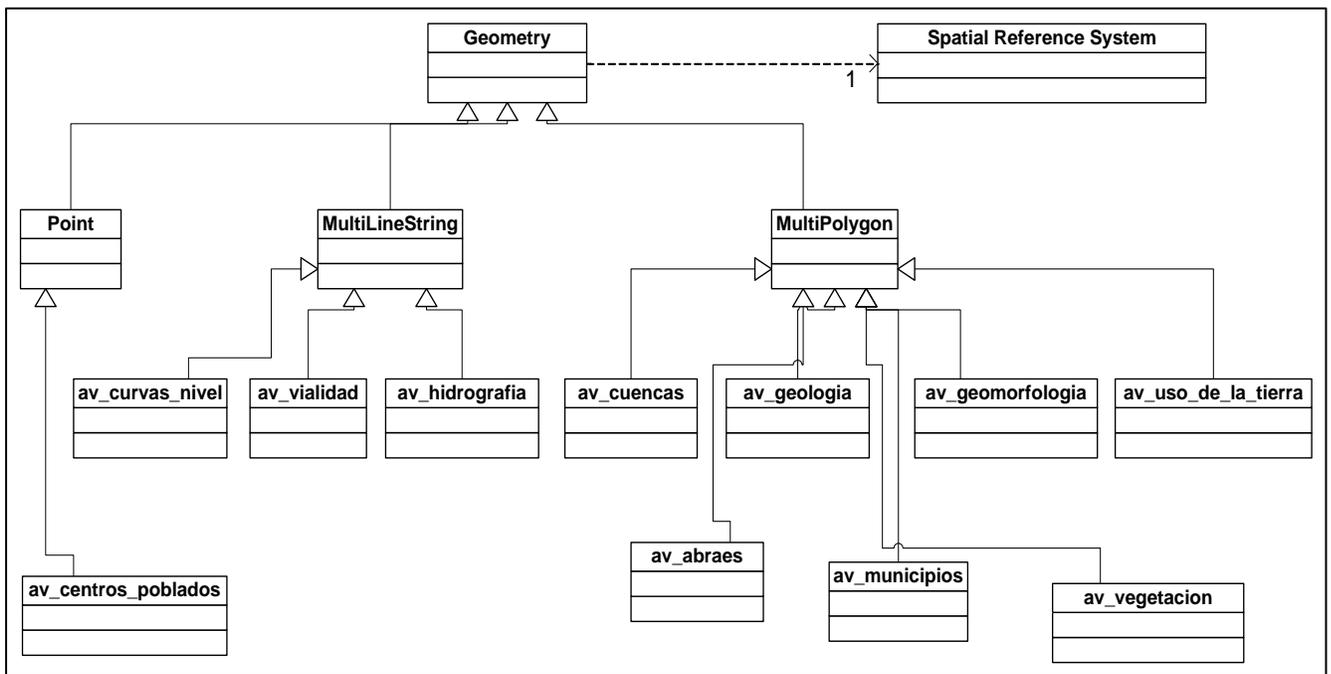


Figura 6. Modelo de Diseño Ubicación de clases espaciales de acuerdo a la especificación del OGC
Elaboración propia

Para los efectos de su comprensión, se han suprimido las relaciones entre las clases definidas en el modelo de análisis y se muestran solo las relaciones del tipo “herencia” entre las clases espaciales y el marco (framework) utilizado por PostGis. La lectura del modelo de diseño anterior puede interpretarse de la siguiente forma: Las clases (espaciales) generadas en el proyecto P_Ávila_2009 se conectan a dicho “Framework” y toman los métodos provistos por las clases básicas Punto (Point), Multilínea (MultiLineString) y Multipolígono (MultiPoligon),

los que a su vez, heredan los métodos de la superclase Geometría (Geometry), la cual permite el soporte de datos georeferenciados a través de la clase Sistema de Referencia Espacial (Spatial Reference System). (Bustos, 2011).

Descripción de las clases de diseño.

En esta sección se describe cómo se diseñaron las clases espaciales y su estructura equivalente en el SGBD PostgreSQL. Zeiler (1999) afirma que el diseño tradicional de bases de datos relacionales sigue dos pasos básicos: la articulación de un modelo lógico de datos y la implementación física de los modelos de bases de datos.

Por lo que, si se toma una clase espacial o no espacial de diseño, su equivalente bajo base de datos será una tabla; en el caso de una clase espacial tendrá una columna para especificar la geometría y el sistema de proyección. Por ejemplo si seleccionamos la clase espacial `av_centros_poblados`, su estructura física (tabla) de acuerdo a PostgreSQL- PostGis sería la mostrada en la siguiente figura.

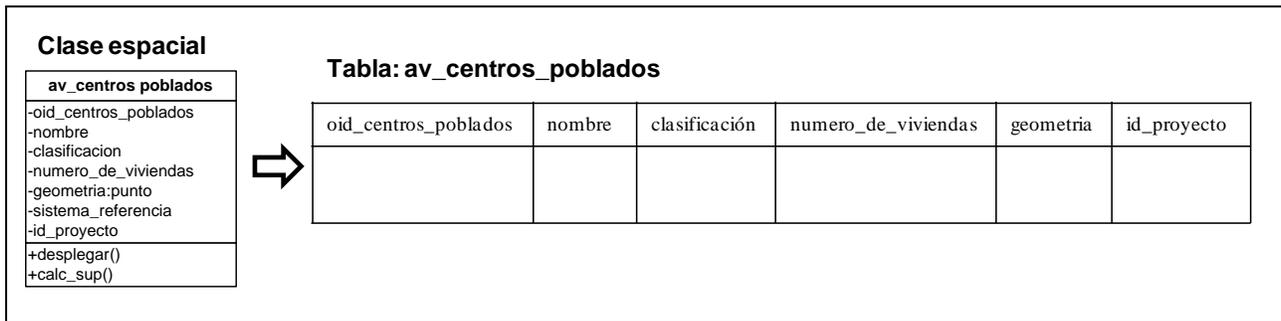


Figura 7. Equivalencia entre Clase espacial (O-O) y Tabla (Modelo Relacional).

Fuente: Elaboración propia.

En el modelo de implementación que a continuación se desarrolla se detallan las funciones de todas las herramientas empleadas.

MODELO DE IMPLEMENTACIÓN

Para implementar la base de datos espacial se ha seleccionado el gestor de bases de datos libre PostgreSQL con su extensión Postgis; se crea la base de datos espacial llamada `bde_cenamb` y se le importan todos los archivos vectoriales (con extensión SHP) generados durante la ejecución de proyectos, a través del comando Shapefile To Postgis Importer (Postgis versión 1.5). A continuación se muestran una vista parcial de la `bde_cenamb` que muestra los archivos

correspondientes al Proyecto P_Avila_2009 y dos tablas llamadas `geometry_columns` y `spatial_ref_sys` que son encargadas de proveer soporte geoespacial.

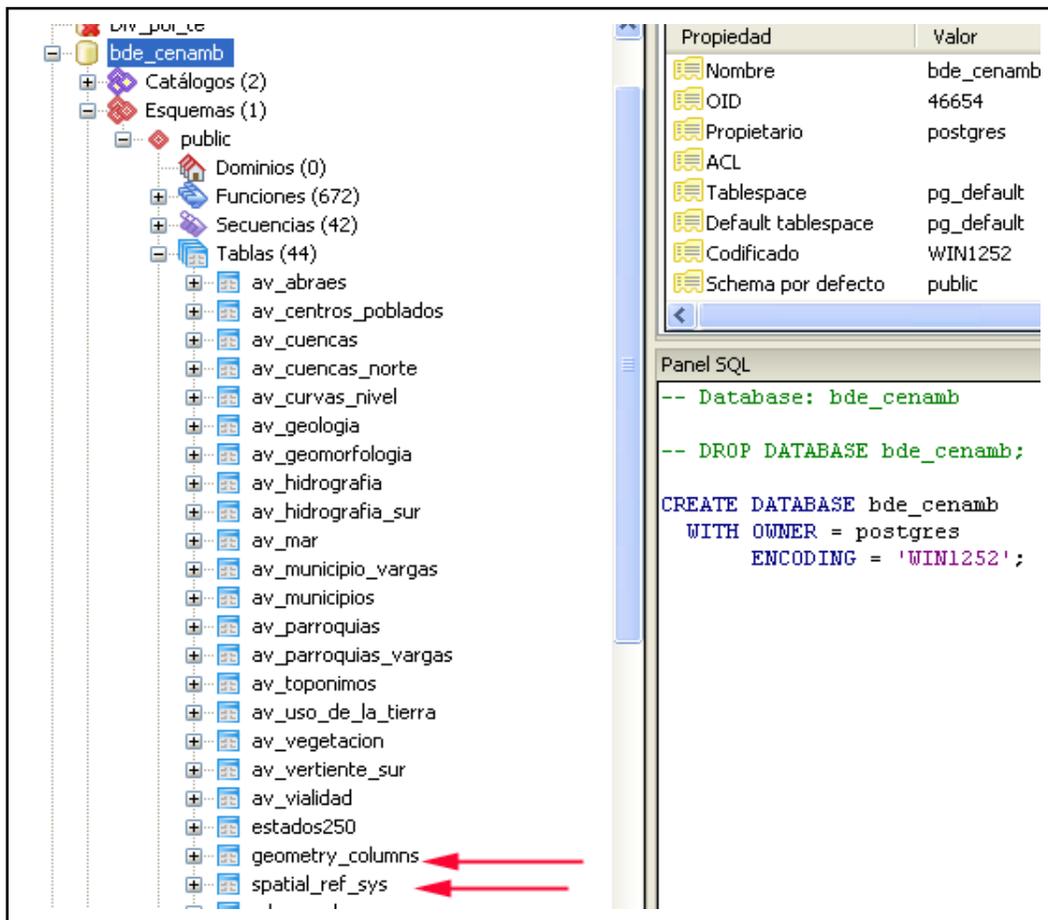


Figura 8. Base de datos espacial cenamb implementada sobre PostgreSQL-Postgis
Fuente: Elaboración propia.

Con el fin de explicar este procedimiento, se toma como caso de estudio la clase espacial `av_centros_poblados`, nombrada anteriormente y se procede a describir cómo se implementa sobre PostgreSQL. El proceso es similar para el resto de las clases lo que cambia es la geometría; que en este caso es “punto”.

Nótese que en el campo `the_geom` (geometry) indicado con la flecha, se almacena la información geoespacial; en este caso, la ubicación de los centros poblados, por razones de espacio, se almacenan en formato Well Known Binary (WKB) y, para visualizar su contenido, debe llevarse al formato Well Known Text (WKT); para ello, debemos ejecutar la siguiente instrucción:

“Select nombre_cpo,asewkt(the_geom) as the_geom,id_proye from av_centros_poblados”

El resultado parcial una vez ejecutada la consulta se muestra en la siguiente figura. Nótese que el campo titulado the_geom (la geometría) contiene la información del sistema de referencia empleado, que en este caso, se corresponde al SRID= 32616 (WGS-84- UTM Zona 19 Norte), y los puntos (POINT) indican un par de coordenadas (Lon, Lat) de acuerdo al mencionado sistema. La implementación utilizando PostGis permite, por tanto, trabajar con información georeferenciada y proyectada. (Bustos, 2011).

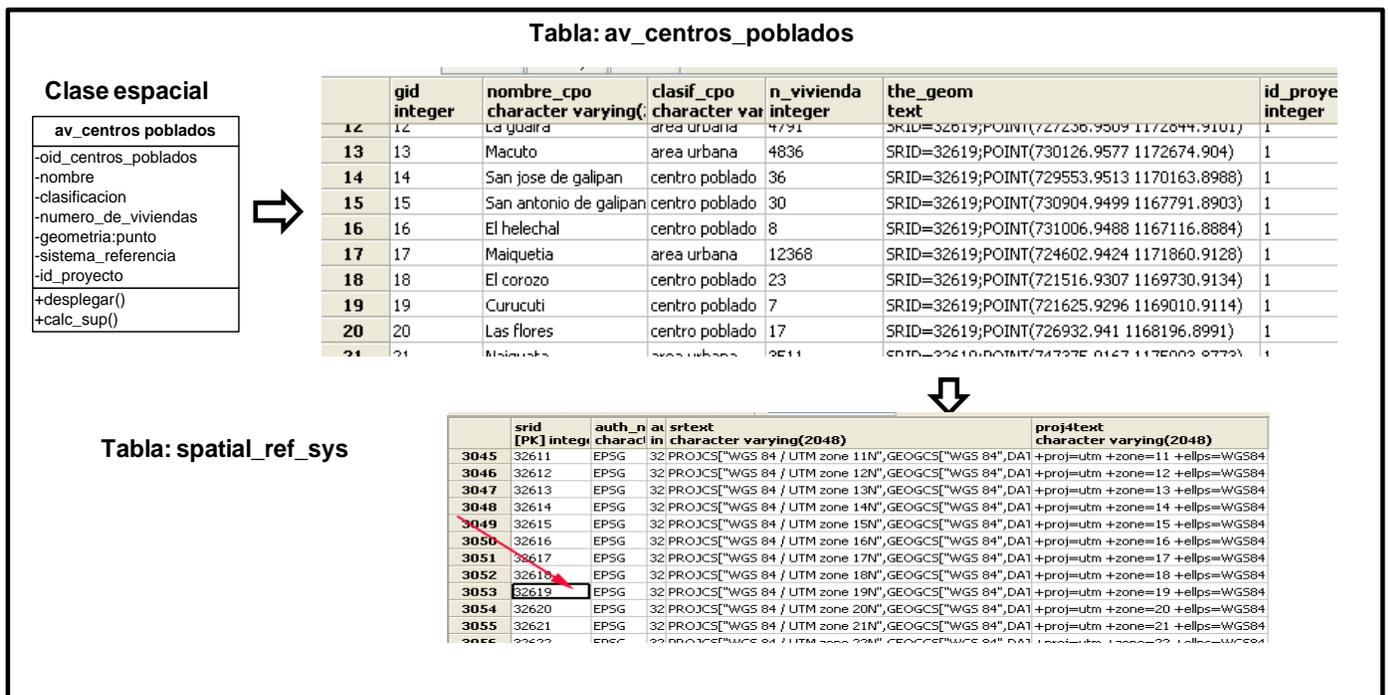


Figura 9. Modelo de implementación. Tabla PostgreSQL: av_centros_poblados
Elaboración propia.

Gestión de la información geoespacial con Quantum Gis.

Para visualizar los mapas digitales, realizar análisis espaciales, actualizar las coberturas vectoriales, agregar/eliminar atributos, producir salidas cartográficas, entre otras, se necesita el empleo de una herramienta SIG que nos permita tales requerimientos. En este trabajo, se ha conectado exitosamente la base de datos espacial creada con el SIG libre Qgis, que permite además de las funcionalidades antes nombradas, muchas otras mediante su conexión con el programa GRASS. El primer paso es crear la conexión entre una base de datos espacial PostgreSQL- PostGIS y Qgis (v. figura 10)

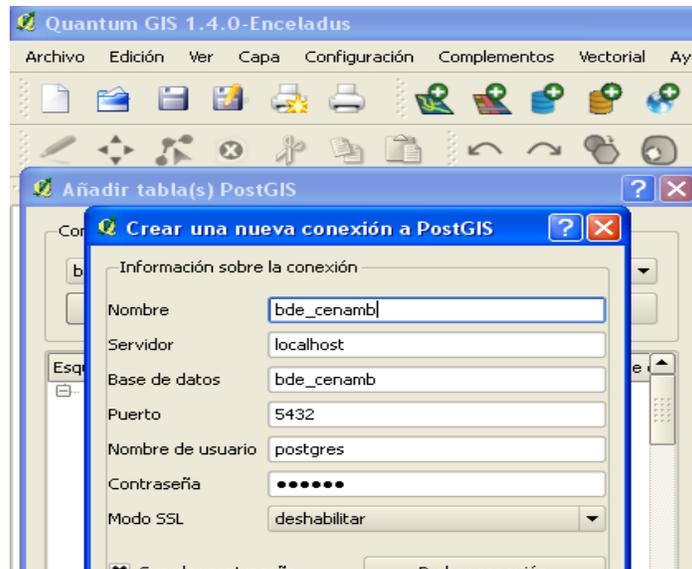


Figura 10. Conexión Postgis de la bde_cenamb con QGIS
Fuente: Elaboración propia.

Una vez realizada la conexión, como siguiente paso se procede al despliegue de la data espacial y sus atributos, si se quiere sea almacenada en la BD PostGIS. La siguiente figura muestra a través del programa Quantum Gis las coberturas correspondientes a municipios, hidrografía y centros poblados trabajados en el Proyecto P_Avila_2009.

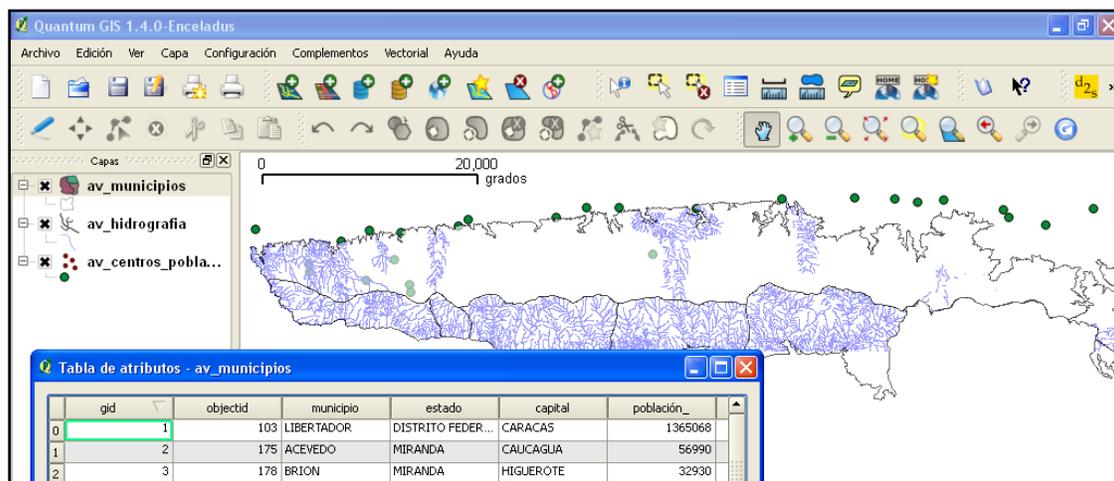


Figura 11. Despliegue de coberturas vectoriales con atributos con QGIS
Fuente: Elaboración propia.

Visualización de la información Geoespacial con Mapserver.

Hasta este punto, hemos visto como se almacena la información espacial y no espacial en tablas utilizando el gestor de base de datos postgreSql y su despliegue utilizando Q-Gis. A continuación, se describe cómo utilizar la plataforma Mapserver, y en particular la opción PHP/MAPSCRIPT, para acceder a la información de la base de datos espacial implementada y desplegar los mapas en un visualizador WEB, como es el caso del servicio Web Map Service (WMS).

Recordemos que la utilidad del Mapserver es proveer un servidor de mapas, que aunque poderoso, está limitado a la simple interfaz de usuario Web. MapScript es una interface que permite acceder a las funcionalidades subyacentes de Mapserver desde una variedad de lenguajes de programación. PHP/MapScript es una interfaz orientada a objetos que se define en términos de clases y métodos; el código fuente para el módulo PHP/MapScript se provee con la distribución Mapserver y por lo tanto es el usado en este trabajo. (Mapserver, 2010)

A continuación, se muestra una parte del archivo MapFile correspondiente al manejo de la cobertura de centros poblados utilizados en el proyecto P_Avila_2009.

```
av_cenpob.MAP
MAP
UNITS meters # unidades del mapa
EXTENT 678369.853125 1120726.556331 807565.396875 1212758.443669 #Area que engloba el mapa
SIZE 600 400
STATUS ON
IMAGECOLOR 255 255 255
FONTSET "fonts/fonts.txt"
SYMBOLSET "symbols/symbols.sym"
WEB
IMAGEPATH "C:/ms4w/Apache/htdocs/avila/tmp/" # Ubicación en el servidor.
IMAGEURL "tmp/"
END
LAYER
CONNECTIONTYPE postgis # conexión con la BD Postgis
NAME "centros poblados avila"
STATUS ON
CONNECTION "user=postgres password=**** dbname=bde_cenamb host=localhost" #conexión con al BD
DATA "the_geom FROM av_centros_poblados as av_centros_poblados using unique gid using SRID=32619"
TYPE POINT
CLASS
NAME "Centros poblados"
STYLE
SYMBOL star
SIZE 5
COLOR 0 34 125
OUTLINECOLOR 0 255 0
```

```
END  
END
```

De este archivo MapFile, que es un programa Script que contiene la información del mapa a desplegar, se resalta la instrucción CONNECTION (conexión), que permite acceder el mapa almacenado en la base de datos PostGis (av_centros_poblados). Los comentarios más relevantes se muestran seguidos del caracter #.

El archivo av_cenpob.MAP es llamado desde un programa desarrollado en lenguaje PHP, llamado base.phpm, del cual, solo hemos colocado dos instrucciones. El código completo se pueden consultar en Bustos (2011).

```
<?php  
.  
.  
$mapObject = ms_newMapObj("av_cenpob.map"); -> inclusión del archivo .map  
$mapImage = $mapObject->draw();  
.  
.  
?>
```

Mediante el uso de un navegador Web (en este caso Mozilla) y teniendo instalado el servidor de mapas Mapserver, se digita la siguiente dirección: <http://localhost/avila/base.php> y, se muestra el mapa que contiene la posición de los centros poblados.



Figura 12. Visualización con Mapserver de los centros poblados trabajados en el proyecto P_Avila_2009
Elaboración propia.

RESULTADOS OBTENIDOS

Una vez finalizada la implementación de estos desarrollos utilizando software libre, podemos afirmar que el repositorio de base de datos creado con el binomio PostgreSQL- Postgis provee un excelente soporte para el manejo de datos espaciales en estructuras de bases de datos. El Quantum Gis cumple con las expectativas planteadas en aspectos tales como, la rapidez en los análisis espaciales, interconexión con PostgreSQL-Postgis, visualización de coberturas tanto vectoriales como raster; con lo que demuestra ser un software confiable. El Mapserver permite desplegar los mapas en formato Web desde una intranet local o incluso, consultas desde Internet utilizando el servicio gratuito No-IP. Por lo que estas cuatro (4) herramientas conjugadas han demostrado que el potencial ofrecido por el software libre en el campo de la geomática es bastante amplio y completo, lo que permite el desarrollo de aplicaciones confiables en este tema; sin embargo, se advierte que al necesitarse la combinación e integración de varias de estas herramientas producidas por diferentes instituciones (PostgreSQL -Universidad de Berkeley, PostGis –Gis Refractions, Mapserver – Universidad de Minnesota) se requiere dedicar tiempo y esfuerzo en su aprendizaje.

CONCLUSIONES

- El proceso unificado de desarrollo de Software (RUP) constituye una metodología que se adapta perfectamente al desarrollo de sistemas geomáticos; la misma, permite llevar los requerimientos descritos en lenguaje natural, mediante los casos de uso hasta una implementación física con herramientas geomáticas libres.
- En cuanto a los manejadores de bases de datos libres, el binomio PostgreSql - Postgis utilizados para la conformación de la base de datos espacial demostró su potencialidad respecto a su robustez, soportes en el manejo de proyecciones y para importar archivos de formato Shapefile a PostGis.
- El entorno de desarrollo para Web Mapserver demostró su confiabilidad como herramienta de publicación sobre la Web, ya que no se detectaron fallas en intranets locales o en las pruebas realizadas con el servicio de dinámico gratuito NO-IP.com
- Con respecto a las herramientas libres utilizadas, aunque pueden ser empleadas en proyectos que manejen gran cantidad de información espacial, se concluye que deben ser

trabajadas de manera combinada entre ellas, e incluso pueden requerir el apoyo de otras para lograr los objetivos; esto se traduce en la necesidad de realizar búsquedas exhaustivas, con notable consumo de tiempo, en Internet (blogs, sitios Web, Foros en línea, etc.,) para obtener la solución.

RECOMENDACIONES

- La construcción de una base de datos espacial concebida sobre el enfoque orientado a objetos permite una gestión eficiente de la información geoespacial y no espacial. Sin embargo para crear una interfaz gráfica que interactúe con la misma, se recomienda el empleo de personal experto en los lenguajes de programación PHP o JAVA.
- La tecnología Mapserver- PHP/MapScript, utilizada con el fin de desplegar mapas digitales referenciados en formato WEB, puede complementarse con otras tecnologías existentes; se recomienda explorar la tecnología Openlayer y el uso de otros programas como Mapbender que permiten mejorar la interfaz para la visualización sobre dicho entorno.
- Una vez conformada la BD Espacial se recomienda explotar mejor las herramientas de análisis espaciales, sobre todo el uso del SQL atributivo y SQL espacial provisto por Postgis y herramientas libres que interactúan de forma transparente con dicha base de datos, ya que solo se realizaron análisis básicos en este campo.
- Finalmente, se recomienda adquirir y configurar un servidor web Mapserver con el fin de poner a disposición de los interesados, la información generada para su consulta.

AGRADECIMIENTO

Los datos utilizados en esta publicación fueron generados durante la ejecución del proyecto Financiado por el Consejo de Desarrollo Científico y Humanístico (CDCH) de la Universidad Central de Venezuela N°. PI-21-7385-2008-1

REFERENCIAS

- APESOL (2005). *Asociación Peruana de Software Libre*. Documento en Línea. Disponible en <http://www.apesol.org/softwarelibre.php>.

- Bustos, X. (2011). *Desarrollo de un sistema de información geoespacial bajo el enfoque orientado a objetos y su implementación usando software libre para proyectos ambientales*. Trabajo de ascenso para optar a la categoría de profesor Asistente. Universidad Central de Venezuela. Facultad de Humanidades y Educación. Escuela de Geografía.
- Bustos, X. (2008). *Una visión policéntrica del ambiente bajo el enfoque orientado a objetos*. Terra, Vol. XXIV, N° 36. Julio-diciembre 2008.
- Castellanos, A & Bustos, X (2011). *Modelo de datos agrarios espaciales bajo el enfoque orientado a objetos (O-O) para el Instituto Nacional de Tierras (INTI)*. Terra, Vol XXVII, N° 42. Julio – diciembre 2011.
- CENAMB (2009). *La Gestión del Conocimiento integrado al Sistema de Información Geográfica como soporte de la Investigación y Gestión del Parque Nacional El Ávila*. Centro de Estudios Integrales del Ambiente. Vicerrectorado Académico. Universidad Central de Venezuela. Proyecto Financiado por el CDCH N°. PI-21-7385-2008-1
- Duke-Williams, O. (2006). *Using PostGIS and MapServer in the Census Interaction Data Service Presentation to AGI Technical SIG 'Open-Source in GIS*. ESRC Award number H507 255177. Documento en Línea. Disponible en http://www.agi.org.uk/SITE/UPLOAD/DOCUMENT/Event_Presentations/TechSIG_20060427/OliverDukeWilliams.pdf.
- FSF. (2009). *Free Software Foundation (Fundación de Software Libre*. Documento en Línea. Disponible en <http://www.fsf.org/>
- Jacobson, I. Booch, G. Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*. Addison-Wesley Publishing Company.
- Lehey, G. (2008). *Qué es BSD*. Documento en Línea. Disponible en <http://www.freebsd.org/doc/es/articles/explaining-bsd/>
- Mapserver (2010). *Mapserver documentation. Release 5.6.1*. The MapServer Team. Disponible en: <http://mapserver.org/>
- M.A.R.N.R. (1992). *Áreas Naturales Protegidas de Venezuela*. Serie de Aspectos Conceptuales y Metodológicos. DGSPOA/ACM/01.
- OGC(2010). *Open Geospatial Consortium (El Consorcio Open GIS)*. Documento en Línea. Disponible en <http://www.opengeospatial.org/>

- PGSQL (2012). *Sitio oficial de PostgreSQL*. Documento en Línea. Disponible en <http://www.postgresql.org/>
- POGIS (2010). *PostGis. Extensión espacial para el SMDOR PostgreSQL*. Documento en Línea. Disponible en <http://postgis.refractor.net/>
- QGIS (2010). *The Quantum GIS Project. (El proyecto Quantum GIS)*. Documento en Línea. Disponible en <http://www.qgis.org/>
- RUP (1998). *Rational Unified Process. Best Practices for Software Development Teams*. Rational Software White Paper. TP026B. Documento en Línea. Disponible en http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- Zeiler, M. (1999). *Modeling our World. The ESRI Guide to Geodatabase Design*. Environmental System Research Institute, Inc. Redlands, California.

Xavier Bustos Catari. Licenciado en Ciencias de la Computación. UCV, 1994. MSc. en Ciencias de la Computación. Universidad Central de Venezuela (2000). Cursando Doctorado en Ciencias de la Computación- Universidad Central de Venezuela. Investigador- Docente del CENAMB y la Escuela de Geografía-FHE. Áreas de investigación: Desarrollo de Sistemas bajo el enfoque Orientado a Objetos, Desarrollos Web, Bases de Datos Geospaciales, Sistema de Información Geográfica, Educación a Distancia, Profesor de las materias “Computación para los Estudios Ambientales” en la Escuela de Geografía de la UCV (pregrado). e “Informática para los Estudios Ambientales” de las Maestrías en Análisis espacial y Gestión del Territorio –FHE y Planificación Integral del Ambiente (CENAMB). Correo electrónico: xavierbustos@gmail.com.