

**DESARROLLO DE UN LENGUAJE DE DOMINIO ESPECIFICO (DSL) GRAFICO  
PARA EL MODELAMIENTO DE BASES DE DATOS ESPACIALES**

**TESIS DE MAESTRIA**

**YENNY ESPINOSA GOMEZ  
CODIGO: 20122395005**

**DIRECTOR:  
Msc JAVIER FELIPE MONCADA SANCHEZ**

**UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS  
FACULTAD DE INGENIERIA  
MAESTRIA EN CIENCIAS DE LA INFORMACION Y LAS COMUNICACIONES  
ENFASIS EN GEOMATICA  
BOGOTA D.C., 2019**

*A mis padres por todo su amor, enseñanzas y entrega incondicional.*

*A mis hijas que son el motor de mi vida y mi inspiración cada día.*

*A mi esposo por toda su ayuda en estos años, creciendo juntos.*

*A mi hermana por ser mi amiga y apoyo.*

## **AGRADECIMIENTOS**

Agradezco a Dios, la Naturaleza, al Universo por la vida y por todo lo que he realizado.

A todas las personas que me ayudaron y aportaron de una u otra manera a realizar este proyecto.

A mi Director de Tesis por su total ayuda, orientación, asesoría y paciencia durante todo el desarrollo de este trabajo y a mis Docentes Jurados porque con su fundamental, valiosa y gran ayuda se realizó este proyecto.

A mi familia, amigos, compañeros, profesores que me animaron para concluir este trabajo y esta etapa.

A mi alma máter, la Universidad Distrital Francisco José de Caldas por ser mi formadora en lo académico, profesional y personal.

## Contenido

Contenido .....	4
RESUMEN .....	5
PALABRAS CLAVES .....	5
1. INTRODUCCION .....	6
2. ESTADO DEL ARTE .....	12
3. METODOLOGIA .....	21
➤ TRANSFORMACION DE MODELOS .....	23
➤ ANALISIS DE RESULTADOS .....	24
4. CARACTERIZACION DEL DISEÑO Y MODELADO DE BASES DE DATOS ESPACIALES .....	26
5.1 DISEÑO Y MODELADO DE UNA BASE DE DATOS ESPACIAL: .....	26
6. DISEÑO DEL METAMODELO PARA BASES DE DATOS ESPACIALES. ....	47
7. SINTAXIS CONCRETA DEL METAMODELO PARA BASES DE DATOS ESPACIALES .....	49
➤ TRANSFORMACION DE MODELOS .....	52
➤ ANALISIS DE RESULTADOS .....	57
8. CONCLUSIONES .....	62
9. RECOMENDACIONES .....	63
10. REFERENCIAS BIBLIOGRAFICAS .....	64

## **RESUMEN**

El modelado de Bases de Datos Espaciales se desarrolla actualmente requiriendo que los usuarios tengan conocimiento en algunas herramientas y lenguajes de propósito general lo que en ocasiones dificulta el proceso de diseño e implementación de estas en entornos computacionales pues se requiere de un gran aprendizaje y conocimiento de esos lenguajes y de procedimientos complejos y engorrosos. Los Lenguajes de Dominio Específico ofrecen una alternativa para facilitar muchos de estos procesos, por ello se evidencia necesario desarrollar una propuesta de metamodelo y su respectivo DSL gráfico que facilite el modelamiento de este tipo de Bases de Datos Espaciales Relacionales para los usuarios finales. Para ello se partirá de una propuesta abstracta para luego llevarla a una sintaxis concreta y posteriormente poder transformar los modelos en código para las extensiones Open Source como PostGIS y MySQL Spatial con el fin que los usuarios finales puedan emplear la propuesta de DSL gráfico y experimentar mejoras en el proceso de modelado de su Bases de datos.

## **PALABRAS CLAVES**

Dato Geográfico, Sistemas de información geográfica, Base de Datos Espacial, Análisis Espacial, DSL, MDE, Modelo, Metamodelo, Geodatabase, SIRIUS.

# 1. INTRODUCCION

## 1.1. ANTECEDENTES DEL PROBLEMA

Las Bases de Datos Espaciales (BDE) se utilizan comúnmente dentro de los Sistemas de Información Geográfica (GIS) en áreas como la geomática, geología, percepción remota, ordenamiento territorial, entre muchas otras, para almacenar y analizar datos geográficos o espaciales y cartográficos y ayudar en la toma de decisiones[1], [2].

Las Bases de Datos Espaciales expanden las capacidades de las Bases de Datos Relacionales facilitándoles el almacenar, consultar y analizar datos espaciales o con componentes geográficos[3], [4]. Permiten realizar búsquedas espaciales y alfanuméricas, logrando realizar consultas tan simples como la determinación de objetos geométricos dentro de una determinada región. Proporcionan la posibilidad de realizar análisis espaciales para probar si dos geometrías se superponen o se cruzan entre muchas otras funcionalidades.[5] Para reducir el tiempo de respuesta de las consultas espaciales, se utilizan muchas técnicas entre ellas la indexación espacial.[5]

Los Lenguajes de Dominio Especifico (DSLs) ofrecen una solución al reto de conseguir que la colaboración multidisciplinaria sea eficiente, efectiva y lo más productiva posible dentro del desarrollo un proyecto, de igual manera son importantes los DSLs dado que aumentan flexibilidad y control del diseño y desarrollo de los mismos[6]. El desarrollo de un Dominio Específico completado a tiempo, posiblemente, por parte de desarrolladores ajenos al proyecto puede ayudar a múltiples usuarios a conseguir resultados en un tiempo más adecuado, sin depender de terceros[7]. De igual manera facilitan el uso, aumentan la expresividad y usabilidad de los diferentes tipos de usuarios de los DSLs[8]. Los DSLs gráficos son lenguajes de dominio específico que a diferencia de los DSL textuales, no usan un conjunto ordenado de sentencias, como los lenguajes de propósito general, estos se basan en entidades (figuras) y relaciones[9].

El desarrollo de aplicaciones requiere constantemente y en la mayoría de ocasiones el uso de un dominio concreto, la solución más simple, pero no la más rápida, para dar respuesta a estos problemas es el uso de un GPL (General Purpose Language) como Python, Java o C++[10], pero recurrir a la utilización de un lenguaje específico de dominio (DSL - Domain Specific Language)[11], que no es más que un pequeño lenguaje restringido y definido concretamente para satisfacer los requerimientos de un dominio concreto[12] puede resultar sencillo para un usuario no especializado.

Lograr el uso por parte de los usuarios de un DSL pasa por suministrar herramientas para trabajar con él, para el caso de DSL visuales es de vital importancia la utilización y disponibilidad de editores gráficos, que permitan y faciliten la elaboración de modelos utilizando el DSL, en este proyecto se pretende posiblemente el uso de marcos de trabajo que soporten la generación de editores gráficos, partiendo de un metamodelo, como es el caso de Sirius[13].

## **1.2. FORMULACION DEL PROBLEMA DE INVESTIGACION**

El diseño de Bases de Datos Espaciales es un procedimiento complejo que está compuesto por varios procesos y actividades a seguir, que en ocasiones algunos de estos procesos pueden resultar un poco engorrosos para los usuarios.

Uno de los inconvenientes que un usuario puede encontrar al intentar crear una Base de Datos Espacial Relacional está en el paso del Modelo Relacional (modelo lógico) al Sistema Gestor de Bases de Datos, para esto el usuario debe ayudarse de un lenguaje como el SQL (Lenguaje de dominio específico textual), pero adicionalmente exigiendo al usuario que conozca de forma muy precisa los tipos de datos y comandos específicos de la extensión espacial utilizada por el Sistema Gestor de Bases de Datos.

Aunque estas extensiones son compatibles con los estándares del Open Geospatial Consortium (OGC) y amplían las capacidades de los sistemas gestores de bases de datos, adicionando tipos, funciones geoespaciales y sistemas de referencia, muchos de sus comandos no son similares; como es el caso de PostGIS y MySQL Spatial,

convirtiendo la creación de una base de datos espacial en algo muy complejo para un usuario, así tenga buenas bases de conocimiento de SQL.

Esto a su vez genera que el rango de disciplinas y usuarios que pueden manipular estas herramientas se disminuya y aumente el consumo de factores preciados como el tiempo y dinero.

### **1.3. OBJETIVOS DE LA INVESTIGACION**

#### **1.3.1. Objetivo General:**

Desarrollo de un Lenguaje de Dominio Especifico (DSL) gráfico, para el modelamiento de bases de datos espaciales, que permita a usuarios finales optimizar recursos y crear estructuras fácilmente con las extensiones Open Source como PostGIS y MySQL Spatial.

#### **1.3.2. Objetivos Específicos:**

A fin de alcanzar el objetivo planteado, se proponen los siguientes objetivos específicos:

- Realizar un diagnóstico de las herramientas Open Source disponibles para la creación de bases de datos espaciales, su uso e implementación, y establecer un estado del arte con relación a estas herramientas.
- Especificar las herramientas tecnológicas y los requerimientos necesarios para la creación del metamodelo que soporte ambientes gráficos.
- Construir un prototipo de metamodelo que permita el modelado de bases de datos espaciales para usuarios finales.
- Realizar la transformación del modelo a código SQL para las extensiones Open Source PostGIS y MySQL Spatial.

### **1.4. PREGUNTA DE INVESTIGACION E HIPOTESIS**

#### **1.4.1. PREGUNTA DE INVESTIGACION**

La pregunta que se busca responder es: ¿Un Lenguaje de Dominio Especifico gráfico aplicado al modelamiento de Bases de Datos Espaciales generará grandes beneficios a usuarios finales al permitir el aprovechamiento y la optimización de algunos de los recursos que para el proceso de diseño e implementación se encuentran disponibles en las extensiones Open Source PostGIS y MySQL Spatial?

#### **1.4.2. HIPOTESIS**

El desarrollo de un Lenguaje de Dominio Específico (DSL) gráfico aplicado al modelamiento de Bases de Datos Espaciales con su respectivo prototipo transformado para las extensiones Open Source PostGIS y MySQL Spatial permitirá a los usuarios finales el aprovechamiento y la optimización de algunos de los recursos durante el proceso de diseño e implementación de este tipo de Bases de Datos.

#### **1.5. ALCANCE Y LIMITACIONES**

##### **1.5.1. ALCANCE**

Con el propósito de realizar la aplicación del DSL gráfico en el contexto del modelamiento de Bases de Datos Espaciales, se recopilarán algunos datos (información geográfica digital vectorial e información no espacial) de una zona específica de la ciudad de Bogotá como caso de estudio y así poder hacer la aplicación práctica o validación de la hipótesis propuesta.

Un DSL gráfico con su transformación a las extensiones PostGIS y MySQL Spatial permitirá a los usuarios finales el aprovechamiento y la optimización de algunos de los recursos durante el proceso de diseño e implementación de una bases de datos espacial ya que les podría facilitar el paso del Modelo Relacional (Modelo Lógico) al Sistema Gestor de Bases de Datos y podría ser aplicable a otro tipo de zonas y/o otros casos de estudios.

Las diversas instituciones que diseñan e implementan bases de datos espaciales y la ciudadanía o usuarios en general podrían utilizar los resultados de la investigación para sus propios proyectos, propósitos y fines. Estas entidades podrían iniciar proyectos de investigación encaminados a estudiar y profundizar en temas relacionados con la aplicación y uso este tipo de alternativas computacionales en cooperación con estudiantes investigadores y docentes de los programas académicos de pregrado y postgrado de la Universidad Distrital Francisco José de Caldas.

### **1.5.2. LIMITACIONES**

La principal limitación identificada es la disponibilidad y el acceso a los algoritmos internos propios de las extensiones PostGIS y MySQL Spatial. Otra posible limitación es la disponibilidad y el acceso a datos (información geográfica digital e información no espacial) de un caso de estudio de prueba, para poder aplicar el Lenguaje de Dominio Específico (DSL) gráfico propuesto, junto con su respectivo prototipo para las extensiones ya mencionadas.

Ante esta eventual limitación en el acceso a datos espaciales se puede optar por las posibles soluciones que se presentan a continuación:

- Buscar varios proveedores de datos de este tipo de información: como el Instituto Geográfico Agustín Codazzi, la Unidad Administrativa de Catastro Distrital, el Servicio Geológico, etc.
- Descargar alguna información que ya se encuentra disponible del portal de la Infraestructura de Datos Espaciales del Distrito Capital (IDEC@), que aunque no está completa totalmente puede acondicionarse y completarse manualmente.
- Descargar alguna información que ya se encuentra disponible del portal de la Infraestructura Colombiana de Datos Espaciales (ICDE) y/o del portal del Sistema de Información Geográfica para el Ordenamiento Territorial Nacional (SIGOT), que aunque no está completa totalmente puede acondicionarse y completarse manualmente.
- Simular o inventar algunos datos que cumplan las condiciones necesarias para este proyecto.

### **1.6. JUSTIFICACION**

Actualmente las Bases de Datos Espaciales (BDE) se utilizan ampliamente como parte de los Sistemas de Información Geográfica (GIS) y su aplicación se da en diversas áreas como la geomática, geología, percepción remota, ordenamiento territorial, el catastro, la administración de tierras, el análisis espacial. Se han venido utilizando para almacenar, gestionar, procesar y analizar datos espaciales con el propósito de entender un fenómeno, resolver algunos interrogantes y ser soporte para la toma de decisiones.

El desarrollo de Bases de Datos Espaciales presenta algunos problemas en los procesos de diseño y la implementación por la dificultad de contar con un gran conocimiento en el manejo y uso de herramientas computacionales así como de lenguajes de programación de propósito general, lo que significa una alta curva de aprendizaje de estas herramientas y lenguajes y por ende se requiere contar con personal especializados en los mismos que en muchas ocasiones por temas presupuestales o por el grado de delimitación de los proyectos de bases de datos espaciales no se puede cubrir, dificultando finalmente el correcto y eficiente desarrollo de los mismos.

Por ello, una solución alterna podría ser el contar con un lenguaje de dominio específico para el desarrollo de Bases de Datos Espaciales que le permitan a los diseñadores, integrantes y usuarios de este tipo de proyectos dedicarse a resolver el problema que se pretende solucionar con la Base de Datos y no tengan que dedicar mucho tiempo al aprendizaje de un lenguaje y de la estructura interna de alguna herramienta, permitiendo así un mejor aprovechamiento y optimización de algunos de los recursos necesarios para la creación de una Base de Datos Espacial-

Y si un posible DSL se planteara desde el punto de vista gráfico, permitiría la posterior transformación del Modelo a código SQL para extensiones como PostGIS y MySQL Spatial, con el fin de que los usuarios puedan emplear dicho DSL gráfico para facilitarles crear la estructura de la Base de Datos espacial reduciendo esfuerzo cognitivo y computacional, así como tiempo de trabajo en este tipo de proyectos.

El desarrollo de este tipo de investigaciones, también pretende aportar al conocimiento, desarrollo y aplicación de otro tipo de DSLs en otros contextos como una alternativa para reducir la curva de aprendizaje de herramientas y lenguajes computacionales de propósito general y permitirle a los usuarios y a las personas en general tener una experiencia de desarrollo de sus proyectos de forma más fácil, agradable y sin requerir muchos esfuerzos, conocimientos y herramientas engorrosas que hasta ahora han dificultado el desarrollo de muchos proyectos en diferentes ámbitos.

## **2. ESTADO DEL ARTE**

### **MARCO TEORICO**

#### **2.1 SISTEMAS DE INFORMACION GEOGRAFICA**

Los Sistemas de Información Geográfica (SIG) se han convertido en nuevos elementos de identidad territorial en el ámbito de una nueva *Sociedad de la Información*. Su valor más esencial reside en las posibilidades que ofrecen para mostrarnos la realidad geográfica de la cual dependen la mayoría de las actividades del Hombre. El análisis de la información geográfica describe los elementos espaciales en función de su posición en la superficie terrestre, de sus atributos temáticos y de las relaciones entre ellos en un tiempo concreto, permitiendo a nuestra sociedad disponer de los suficientes elementos de juicio para afrontar decisiones de futuro con el rigor necesario[14].

Otra definición de SIG, bastante aceptada por la comunidad de las ciencias de la información geográfica, es: Un sistema de hardware-software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión[15].

Los sistemas de información geográfica tienen cinco componentes básicos (ver figura 1) que permiten su desarrollo, utilización y manipulación: Recurso Humano, Recurso Técnico (Hardware y Software), la Información Geográfica, los Procesos y Procedimientos y la Organización.



Figura 1. Componentes Básicos de un SIG

Los Sistemas de Información Geográfica son muy importantes porque permiten resolver diversos problemas territoriales. El uso práctico de un SIG es tan variado que va “desde el inventario de los recursos naturales y humanos hasta el control y la gestión de los datos catastrales y de propiedad urbana y rústica (catastro multipropósito), la planificación y la gestión urbana y de los equipamientos, la cartografía y el control de grandes instalaciones, el marketing geográfico, etc. En realidad un Sistema de Información Geográfica es útil en cualquier área donde sea necesario el manejo de información espacial”[16].

Los SIG son una ciencia, una disciplina, una tecnología y una metodología para resolver problemas aplicados[4].

## 2.2 DATO GEOGRAFICO:

Los datos geográficos son entidades espacio–temporales que cuantifican la distribución, el estado y los vínculos de los distintos fenómenos u objetos naturales y sociales. Un dato geográfico se caracteriza por tener[17]:

- Posición absoluta: ubicación o localización del objeto o entidad o dato geográfico respecto de un sistema de referencia o de coordenadas (x, y, z).
- Posición relativa: ubicación o localización del objeto o entidad o dato geográfico respecto de otros elementos del paisaje o entorno geográfico (topología,

relaciones espaciales como: incluido, adyacente, cruzado, conectado, entre otros).

- Figura geométrica: es la representación del objeto o entidad o dato geográfico, o su geometría (punto, línea, polígono).
- Atributos que lo describen: información temática o alfanumérica que describe, define y/o distingue al objeto o entidad o dato geográfico (características del elemento o fenómeno).

Los datos geográficos o datos espaciales o elementos espaciales constan de dos componentes: el componente espacial (la observación o la entidad) y el componente temático (la variable o atributo temático). Los SIG deben ser capaces de gestionar ambos componentes o elementos[18].

## **2.3 MODELO DE DATOS INTERNO EN EL AMBITO DE BASES DE DATOS ESPACIALES**

### **2.3.1 Base de datos Espacial:**

Conjunto de información interrelacionada, almacenada digitalmente que incluye datos sobre la localización y los atributos de los elementos geográficos que han sido codificados como puntos, nodos, líneas, polígonos grid o píxeles[19], [20].

En la actualidad, los sistemas de bases de datos espaciales se utilizan en formas relativamente sofisticadas que van mucho más allá que la simple gestión, visualización y análisis de información geográficamente referenciada. Un típico sistema de base de datos espacial es mayor en sus capacidades para manipular nuevos tipos de datos y modelos, estructuras de datos complejas incluyendo indexación espacial y sofisticados algoritmos y operadores para tratar los datos de manera eficiente. En este enfoque no sólo se articula el uso de la información espacial en los dominios de aplicación convencional, sino que también proporciona el puente necesario que permite incorporar aplicaciones interoperables en el marco de la tecnología de la información (Yeung & Hall, 2007).

Las Bases de Datos Espaciales juegan un papel importante en muchas aplicaciones, incluidos los sistemas de información geográfica, apoyo a la decisión, transporte inteligente, y asignación de recursos. La característica clave que hace que una base de datos espacial se haya convertido en una herramienta poderosa es su capacidad para manipular (por ejemplo, modelo, índice y consulta, etc.) y no almacenar simplemente, objetos espaciales (por ejemplo, puntos, líneas, polígonos, etc.).[21]

### **2.3.2 Modelo de datos:**

Método formal de ordenación de los datos almacenados, dirigido a recrear las condiciones de los elementos del mundo real a que hacen referencia los mismos. Un Modelo de Datos debe integrar reglas que regulen el tipo de datos, las relaciones y las operaciones entre éstos.

Conjunto de directrices para la representación de una organización lógica de los datos en una base de datos. Consiste en unidades lógicas de datos interrelacionadas entre sí (Tsichritzis & Lochovsky, 1977). Estos modelos de datos son conocidos también como modelos de datos internos que permiten el desarrollo y la manipulación de bases de datos y para este caso de bases de datos espaciales.

El papel o rol que juega el Modelo de Datos Interno en un SIG, se presenta en la figura 2, en ella se puede apreciar que se parte de la realidad y de una situación particular, luego se plantea un modelo en el que se pueda entender la dinámica de los datos (geográficos y no geográficos), para luego llevarlos al computador con el fin de tener un almacenamiento, procesamiento y análisis más eficiente y automatizado que permita obtener información con valor agregado que posteriormente será analizada, interpretada, explicada, confrontada por las personas y/o expertos de acuerdo con la dinámica que se esté analizando. Con el fin último de que dicha información y sus respectivos análisis sirvan de base y soporte para la toma de decisiones más acertada y sustentada que va a revertirse o volver nuevamente a la realidad por medio de planes, acciones y/o políticas en beneficio y mejoramiento de la situación particular estudiada.

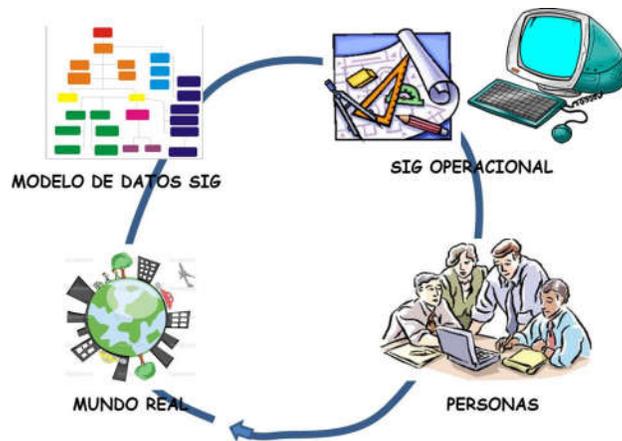


Figura 2. Rol de un Modelo de Datos en un SIG

Fuente: Propia

Al trabajar en un SIG es necesario introducir la información geográfica en un sistema de información y realizar una modelización de la realidad para representar los datos geográficos de forma digital por medio un modelo de datos, siendo los modelos de datos más empleados el vectorial y el ráster[22]. Así en función del tipo de representación (modelo de datos) que se utilice, se distinguen dos tipos de SIG: vectorial y ráster[23].

Si se requiere de información de una amplia zona sin importar los límites de los objetos, se emplea el modelo de datos ráster, ya que con este tipo de datos se dispone de información continua. Si por el contrario es importante conocer el límite de las entidades geográficas y se requiere tener los elementos de forma individual, se trabajará con el modelo de datos vectorial[22].

### 2.3.3 Metamodelo:

Un modelo es una abstracción de la realidad, que la presenta de una manera simplificada y generalizada. Un metamodelo es un modelo que describe un Lenguaje de Modelado (LM), con el que se describen otros modelos.

La noción de metamodelo se basa en la arquitectura de metadatos adoptada por el consorcio Object Management Group (OMG) en la especificación del Meta-Object Facility (MOF) [24]

Esta figura muestra la diferencia entre información, modelos, metamodelos y meta-metamodelos. MOF nombra cada una de estas capas o niveles con los nombres M0, M1,

M2 y M3 respectivamente, como se puede ver en la figura 3. De ahora en adelante, se usarán los términos M0, M1, M2 y M3 para referirse a las capas de MOF. Es importante fijarse en que estos conceptos están en capas diferentes y poseen significados diferentes, aunque usen la misma notación.

El nivel más alto es el nivel de los meta-metamodelos, M3. Algunos ejemplos de lenguajes en este nivel son MOF [24] usado por OMG, [25] usado por EMF, y GOPRR (Grafo, Objeto, Propiedad, Relación, y Rol) [26]. Las instancias de los lenguajes de M3 son los metamodelos, que definen LMs y corresponden al nivel M2. Un ejemplo bien conocido de metamodelos es la definición de UML [27]. Luego están los modelos que representan, por ejemplo, diseños de aplicaciones, estos son instancias de los metamodelos y constituyen el nivel M1, cualquier diseño concreto de una aplicación en UML podría servir como ejemplo. Por último, se encuentra el nivel M0, que sería la implementación de los diseños, por decirlo en términos de programación Orientada a Objetos, en el nivel M0 se considerarían los objetos de las clases en ejecución y en el nivel M1 estarían las definiciones de las clases de objetos.

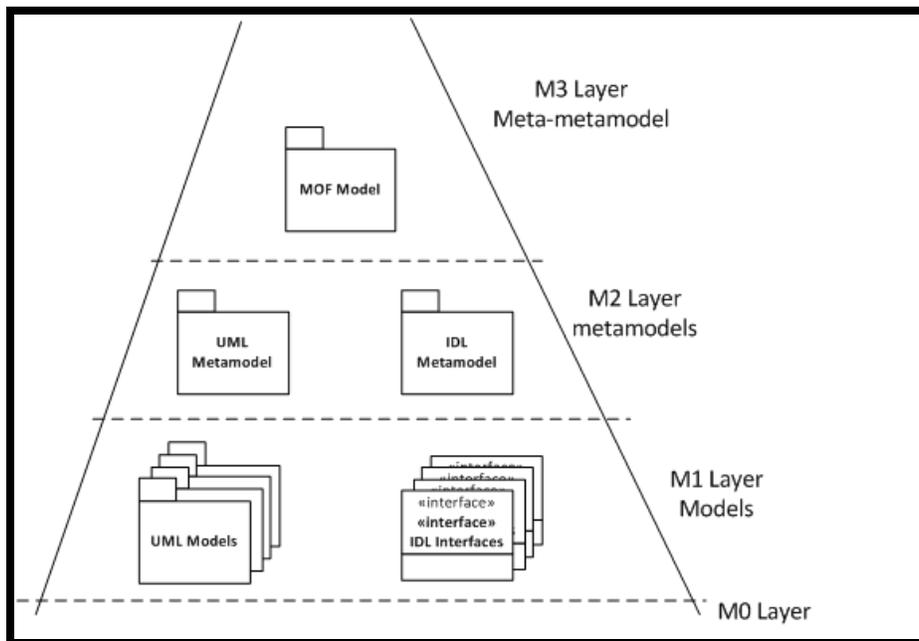


Figura 3 Arquitectura de Metadatos en MOF.[28]

Se emplea el término "metamodelado" para generalizar los problemas que surgen en diferentes industrias en todas las etapas: recopilación de datos, procesamiento, gestión de objetos con fines de automatización, simplificación y ordenamiento, con el fin de formar una estructura común que aún tenga un cierto nivel de flexibilidad y extensibilidad.[29]

Dos propiedades importantes en las que se basa el metamodelado son:

- Proximidad - trabajando con el modelo formal elegido no iremos más allá de sus límites;
- La integridad del sistema formal elegido significa que dentro de su marco es posible describir todos los objetos del conjunto dado. [29]

Los sistemas basados en los principios de metamodelado también deben tener las propiedades de capacidad de expansión (adaptabilidad) y universalidad (comunidad).[29]

#### **2.3.4 Lenguaje de Dominio Específico:**

Los lenguajes de dominio específico (Domain Specific Languages DSL) pueden ser considerados como argumentos escritos en un lenguaje de programación más general. El lenguaje de programación “real” ejecuta el parser sobre el código del DSL, para luego trabajar sobre él. Generalmente, las funciones del DSL sólo se centran en lo que se quiere hacer, y el sistema más grande resuelve el cómo hacerlo.

Son muchos los DSL existentes actualmente, suficiente como para cubrir gran parte de las aplicaciones que se puedan necesitar, especialmente en el campo de las Bases de Datos Espaciales. Los DSL más populares incluyen todos los lenguajes de consulta (query), todos los lenguajes plantilla, Shell scripts, lenguajes de almacenamiento e intercambio de datos como XML, o lenguajes para documentos como LaTeX, CSS o HTML[30].

Tres ejemplos destacados de DSL son:

- » Structure Query Language – SQL:

Este DSL aparece por primera vez en 1974, como sucesor de SEQUEL (Structured English Query Language), el lenguaje de IBM para acceder a datos basado en el cálculo de predicados. Fue diseñado por Donald D. Chamberlin en los laboratorios de IBM y lanzado oficialmente al mercado en 1986, año en el cual

fue estandarizado por ANSI y un año después por ISO, creando así la primera versión del lenguaje. Se desarrolla con el fin de poder acceder a bases de datos relacionales y especificar diferentes tipos de operaciones en ellas. Se caracteriza por el uso de álgebra y cálculo relacional para recuperar y modificar información en las bases de datos[31].

» Unified Modeling Language – UML:

Es el ejemplo más claro de lenguajes de dominio específico gráficos. Se utiliza para modelar sistemas de software, y en los últimos años modelos de bases de datos relacionales y orientadas a objetos. Permite la visualización, especificación, construcción y documentación de un sistema. Algunos miembros de la comunidad científica no consideran a UML como un DSL debido a que sus funciones son netamente de modelado y no sigue los ideales de la programación estructurada. Sin embargo, sus funciones de modelado son lo que lo hace precisamente un lenguaje de dominio específico, y son de gran importancia para implementar los procesos de la ingeniería de requerimientos y en la programación orientada a objetos. En 2005 se lanza como un estándar aprobado por la ISO, respaldado por el OMG (Object Management Group), aunque su primera aparición es en 1997 como propuesta de “Los tres amigos” (James Rumbaugh, Grady Booch, Ivar Jacobson) para el OMG[32].

» Extensible Markup Language – XML:

Es un lenguaje de marcas utilizado para almacenar datos de manera legible. Permite definir la gramática de lenguajes específicos, al igual que HTML, para estructurar documentos, con la gran diferencia que XML soporta bases de datos, permitiendo la comunicación de aplicaciones e integración de información. Fue desarrollado por el W3C (World Wide Web Consortium) como subconjunto del SGML (Standard Generalized Markup Language), la normalización del GML (Generalized Markup Language) de IMB hecha por ISO en 1986. XML ha servido como base para la creación de más lenguajes como XSL, XLINK y de tecnologías como Xades, Xpath, XQuery y XLT, entre otros[33].

#### **2.3.4 Sistemas de Generación de Código:**

Para completar el ciclo de vida de un proceso de transformación que obtenga como producto una aplicación ejecutable es necesario ejecutar una transformación de modelo a texto. Es decir, a partir de los modelos transformados es necesario obtener el código fuente final que se ejecutará sobre la plataforma de destino.

Los generadores de código tienen que obtener la información de los modelos y generar artefactos (p.e., código fuente) a partir de ellos. En los casos más sencillos se mapea cada símbolo del lenguaje a un determinado fragmento de código; el código generado también podría ser variable dependiendo de los valores de las entrada. La idea es que a partir del modelo se generen artefactos sin necesidad de intervención manual[28].

### **3. METODOLOGIA**

Con este proyecto se desarrolló un Lenguaje de Dominio Específico (DSL) gráfico para el modelamiento de Bases de Datos Espaciales que facilite el proceso de migración del Modelo Relacional al Sistema Gestor de Bases de Datos, el cual permitirá a un usuario final realizar dicha tarea sin tener unos conocimientos amplios y especializados en el funcionamiento, sintaxis y semántica de un lenguaje de propósito general que se requiere para este tipo de proyectos, sino que se podrá emplear un DSL gráfico.

Por tanto, en el presente capítulo se va a presentar la metodología que se pretende realizar para alcanzar el objetivo propuesto.

#### **METODO DE LA INVESTIGACIÓN**

El tipo de investigación de esta propuesta de proyecto de Tesis se considera como una investigación aplicada, en la que se seguirá el método hipotético-deductivo, ya que se ha postulado una hipótesis por medio de la intuición o de un supuesto del investigador, luego se pondrá a validación o verificación dicha hipótesis creando un DSL gráfico que facilite el diseño e implementación de Bases de Datos Espaciales Relacionales, aplicándolo a una zona de estudio de la ciudad de Bogotá, con el fin último de presentar las bondades y sus diferentes alternativas de uso o empleo de este tipo de alternativa que no requiere de un gran conocimiento en la herramienta y en el lenguaje reduciendo esfuerzo en el proceso.

#### **DISEÑO DE LA INVESTIGACION**

Para esta propuesta de investigación se requiere contar con algunos insumos dentro de los cuales se encuentran: un conjunto de datos muestra de alguna información geográfica digital vectorial de una zona de estudio de Bogotá para ser gestionada y administrada en una Base de Datos Espacial, unos recursos físicos como computador con acceso a

internet y software, tanto básico como especializado, unos materiales básicos como papelería, medios de almacenamiento, algunos recursos bibliográficos como libros y artículos de consulta.

## **ESCENARIO**

El escenario en el que se va a realizar la investigación está dado en el contexto de cualquier organización o empresa o entidad que administra y gestiona diferentes datos tanto espaciales como no espaciales y que requiera que estos sean almacenados, georreferenciados, manipulados, gestionados y analizados espacialmente mediante el empleo de un Base de Datos Espacial Relacional.

## **INSTRUMENTACION**

Dentro de las posibles herramientas tecnológicas (hardware y software) que se requieren para realizar el desarrollo del DSL gráfico, se encuentran:

- Computador con acceso a Internet y programas básicos de oficina.
- Software o Aplicaciones o Framework o Especificaciones Informáticas y/o Especializadas, posiblemente como: PostgreSQL, PostGIS, MySQL, MySQL Spatial, IDE Eclipse, Sirius, entre otros.

Al desarrollar el proyecto completamente se indicará si los datos son confidenciales o no, así como los derechos de autor y/o la propiedad intelectual según el caso.

## **PROCEDIMIENTO**

Las actividades a desarrollar en este trabajo y que presentan el proceso de desarrollo de un DSL para facilitar el diseño y desarrollo de una Base de Datos Espacial Relacional, son:

### ➤ *LEVANTAMIENTO DE LA INFORMACION*

- Actividad 1 Recopilación de información General: Generar una base documental que soporte la elaboración de la investigación en las áreas de: diseño de base de datos espaciales relacionales, modelamiento de Bases de datos espaciales, estructura de

una Bases de Datos Espacial Relacional, ontología y construcción de un metamodelo, construcción de DSLs gráficos basados en metamodelos.

- Actividad 2 Recopilación de Información general y detallada sobre las extensiones Open Source PostGIS y MySQL Spatial para comprender su estructura, algoritmos, sintaxis y semántica.

➤ *DISEÑO DEL METAMODELO PARA BASES DE DATOS ESPACIALES.*

- Actividad 1: Diseño del metamodelo para la sintaxis abstracta del DSL.  
En esta etapa se diseñará el metamodelo correspondiente que representará la sintaxis abstracta del DSL para Bases de datos espaciales que permita representar relaciones (tablas), geometría, objetos, referencia espacial, relaciones entre objetos, etc.,
- Actividad 2 Sintaxis Abstracta del Metamodelo.  
En esta etapa se representa el metamodelo en diferentes formatos, posiblemente como un diagrama de clases, para ello posiblemente se utilice el framework EMF/Ecore dentro de la IDE Eclipse. Para diseñar y representar esta sintaxis abstracta del Metamodelo en forma gráfica se puede utilizar un Diagrama Ecore de la herramienta Ecore Tools.

➤ *SINTAXIS CONCRETA DEL METAMODELO PARA BASES DE DATOS ESPACIALES*

- Actividad 1: Sintaxis concreta del metamodelo con un Editor Textual:  
En esta actividad se desarrollará un Editor textual que permita crear una sintaxis concreta del metamodelo anterior, con el fin de escribir las instrucciones y la gramática en el lenguaje de propósito específico definido, para ello se podrá utilizar Xtext.
- Actividad 2: Sintaxis concreta del metamodelo con un Editor Gráfico:  
En esta actividad se desarrollará un Editor textual que permita crear una sintaxis concreta del metamodelo anterior, con el fin de escribir las instrucciones y la gramática en el lenguaje de propósito específico definido, para ello se podrá utilizar Sirius.

➤ *TRANSFORMACION DE MODELOS*

- Actividad 1: Realizar las transformaciones de modelos para obtener el despliegue de

una herramienta que facilite el diseño e implementación de Bases de Datos Espaciales Relacionales.

- **Actividad 2: Generación de Código:**

En esta actividad se seleccionará un sistema generador de código o parser a partir del modelo para generar un lenguaje de programación de dominio específico para facilitar el diseño e implementación de bases de datos espaciales con las extensiones PostGIS y MySQL Spatial, para ello se podrá utilizar Acceleo, Java.

- *ANALISIS DE RESULTADOS*

- **Actividad 1: Documentación de los procesos realizados durante el desarrollo del proyecto.** (Esta actividad se desarrolla durante todo el proyecto ya que es transversal desde el inicio del proyecto)

- **Actividad 2: Análisis de Resultados:**

En esta actividad se analizará los resultados obtenidos al desarrollar el DSL gráfico para facilitar el diseño e implementación de bases de datos espaciales

- *PRESENTACIÓN DE INFORMES DE LAS ACTIVIDADES REALIZADAS*

Esta actividad se desarrollará a lo largo del proyecto para documentar lo realizado, los resultados obtenidos y las conclusiones o lecciones aprendidas que se vayan determinando.

- *PRESENTACIÓN DE UN ARTÍCULO DE INVESTIGACIÓN QUE PUEDA SER PUBLICABLE*

Esta actividad consiste en redactar, revisar y consolidar un artículo de investigación publicable en una revista de investigación relacionada con la temática de este proyecto, como mecanismo de difusión y divulgación de los resultados o conclusiones obtenidas al realizar esta propuesta de tesis.

- *DEFENSA DE TESIS.*

Esta actividad consiste en realizar todo el procedimiento tendiente a presentar formalmente el desarrollo de esta propuesta de Tesis con fines a obtener el título de Maestría en Ciencias de la Información y las Comunicaciones, énfasis Geomática.

La duración estimada para realizar este trabajo es de seis meses, que podrían extenderse según el avance y desarrollo de esta propuesta, así como de las fortalezas y de las dificultades que se puedan presentar sobre la marcha.

## PROCESAMIENTO Y ANALISIS DE DATOS

El procesamiento y análisis de los datos se realizará de acuerdo con los procedimientos planeados anteriormente; todo esto permitirá determinar si el uso de un Lenguaje de Dominio Específico (DSL) gráfico aplicado al modelamiento de Bases de Datos Espaciales con su respectivo prototipo transformado para las extensiones Open Source PostGIS y MySQL Spatial permitirá a los usuarios finales el aprovechamiento y la optimización de algunos de los recursos al alcance.

Los resultados obtenidos con este proyecto permitirán validar si la hipótesis planteada en la propuesta de proyecto de tesis es cierta o falsa.

La metodología aplicada en este proyecto se puede apreciar en el siguiente esquema (ver figura 4)

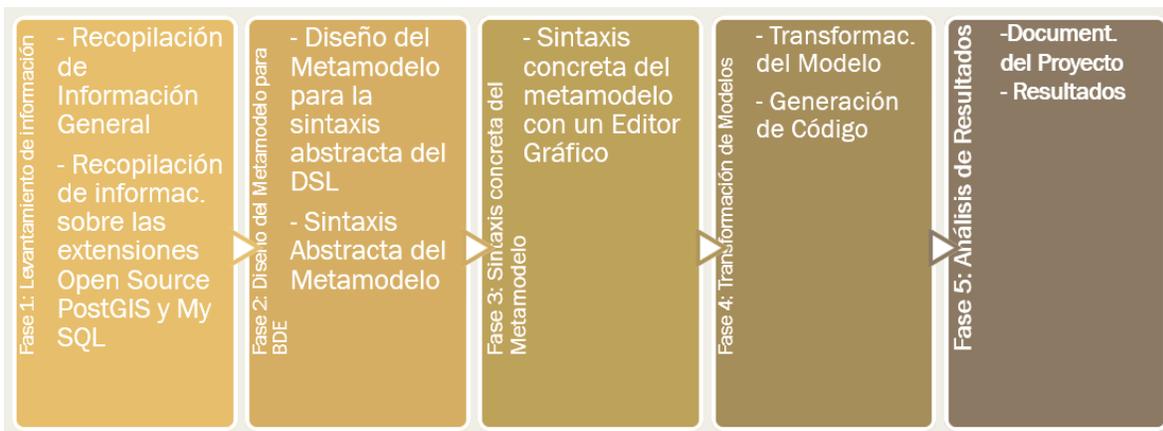


Figura 4. Metodología del Proyecto

Fuente: Elaboración Propia

## 4. CARACTERIZACION DEL DISEÑO Y MODELADO DE BASES DE DATOS ESPACIALES

### 5.1 DISEÑO Y MODELADO DE UNA BASE DE DATOS ESPACIAL:

Las etapas de desarrollo de una Base de datos Espacial son (ver figura 5):

- Conceptualización (Formulación del Problema – Definición de Requerimientos)
- Diseño
- Implementación
- Evaluación

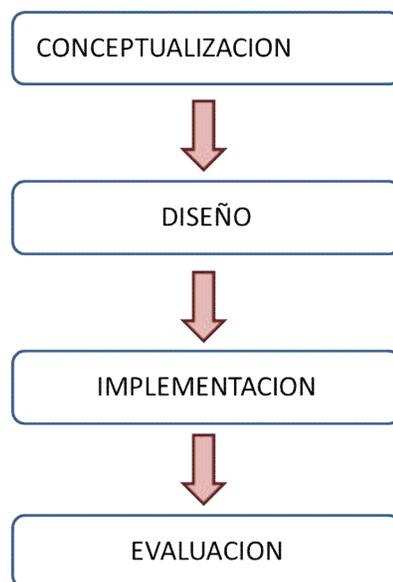


Figura 5. Etapas de desarrollo de una Base de datos Espacial

#### 5.1.1. CONCEPTUALIZACION

Es el proceso que busca conceptualizar todo lo relacionado con la Base de datos Espacial a desarrollar. Es un proceso que pretende dar claridad inicial de la Bases de Datos Espacial antes de implementarlo, o de realizar cualquier actividad en alguna herramienta para manejar Bases de Datos Espaciales (Software)

Tiene las siguientes actividades:

- Formulación del Problema.
- Definición de requerimientos.
- Determinar el problema a ser solucionado con el desarrollo de la Base de Datos Espacial.
- Definir claramente los objetivos de la Base de Datos Espacial, es decir las metas propuestas (esto busca delimitar el proyecto a desarrollar)
- Determinar los posibles usuarios de la Base de Datos Espacial (Es importante que en el proceso de Contextualización se tenga presente el contexto de la Organización que soporta la Base de datos espacial. Entendiendo como Organización uno de los Componentes Básicos de la Base de Datos Espacial)
- Definir los requerimientos de los usuarios (Sus necesidades, estas necesidades de los usuarios van a dar orientaciones iniciales a la definición de las posibles funciones u operaciones que debe tener la Base de Datos Espacial).

Esto conlleva al Análisis funcional de la Base de Datos Espacial:

- Establecer la información requerida y evaluar las posibles fuentes de información. Es decir, identificar el flujo de información.
- Definir una Lista de Entidades con sus posibles atributos y presentar esto en un cuadro resumen. Como por ejemplo (ver tabla 1):

<b>Entidad</b>	<b>Geometría</b>	<b>Atributos</b>
Predio	Polígono	<ul style="list-style-type: none"> <li>• Cedula_Catastral</li> <li>• Matricula_Inmobiliaria</li> <li>• Código_Predial</li> </ul>
Manzana	Polígono	<ul style="list-style-type: none"> <li>• Código_Barrío</li> <li>• Código_Manzana</li> </ul>
Propietario	No Espacial	<ul style="list-style-type: none"> <li>• Cedula_Ciudadania</li> <li>• Nombre</li> <li>• Género</li> </ul>
Vía	Línea	<ul style="list-style-type: none"> <li>• Nombre_Vía</li> <li>• Tipo</li> </ul>

Tabla 1. Muestra de Listado de Entidades con sus atributos

- Reglas que rigen procesos:  
En esta actividad se listan las reglas que rigen los procesos, de acuerdo con las relaciones o la dinámica que se da entre los objetos geográficos y no geográficos.

Por ejemplo:

- Un Barrio contiene una o más Manzanas.
  - Una Manzana está contenida (o localizada) en un Barrio.
  - Una Manzana contiene uno o más Predios.
  - Un Predio debe estar localizado en una manzana.
  - Un Predio puede contener una o más Construcciones.
  - Una Construcción debe estar localizada en un predio.
  - Un Propietario debe tener uno o más Predios.
  - Un Predio debe ser poseído por uno o más propietarios.
- Se realiza un diagrama que plasme en papel el “Modelo Conceptual” aunque teóricamente sabemos que el modelo conceptual es un modelo mental; pero este se materializa o se vuelve tangible para poderlo revisar, chequear, corregir y que sea soporte a la documentación de la Base de Datos Espacial. Por ejemplo (ver figuras 6 y 7):

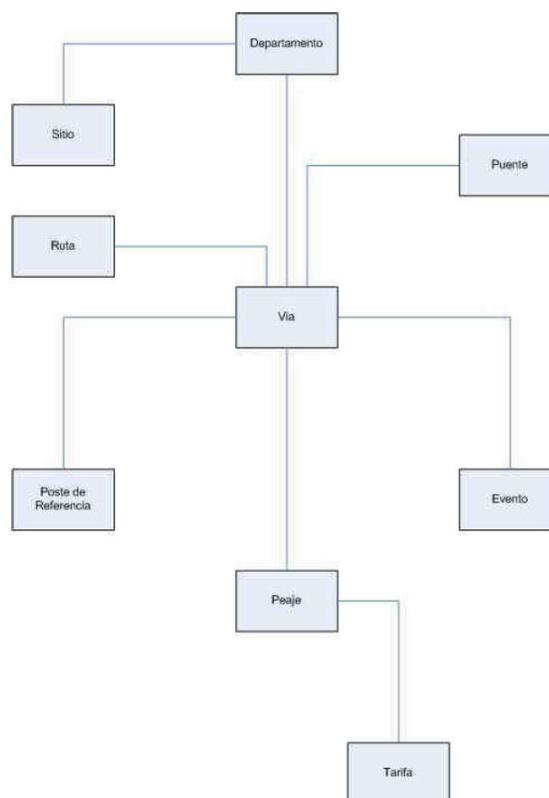


Figura 6. Ejemplo No. 1 de Modelo Conceptual

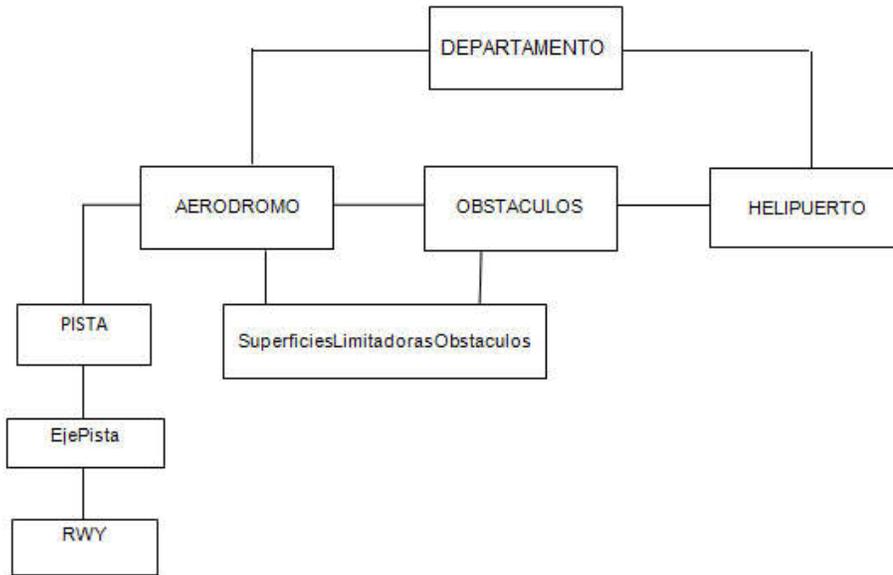


Figura 7. Ejemplo No. 2 de Modelo Conceptual

### 5.1.2. DISEÑO

En esta etapa se diseñan los modelos que luego se van a implementar ya que esta etapa corresponde al Diseño de la Base de Datos Espacial. En general el proceso es similar al proceso de diseño de una Base de datos Tradicional.

Primero realizamos el Modelo Lógico y luego realizamos el Modelo Físico.

Si estamos diseñando bajo un enfoque o una orientación Relacional, nuestro Modelo Lógico es un modelo Entidad – Relación y el Modelo Físico es un Modelo Relacional.

- Modelo Lógico (corresponde con un Modelo Entidad – Relación) El modelo Entidad – Relación presenta las relaciones existentes entre las diferentes entidades. Es importante recordar que las relaciones pueden ser: Uno a Uno, Uno a Muchos y Muchos a Muchos. Las Relaciones tienen Cardinalidad y Obligatoriedad.
- Si seguimos mirando, nuestro ejemplo anterior, nuestro modelo Entidad – Relación para ese caso y teniendo en cuenta las reglas que rigen los procesos definidas previamente, sería:

Algunos ejemplos de Modelos Entidad – relación se aprecian en las figuras 7 y 8:

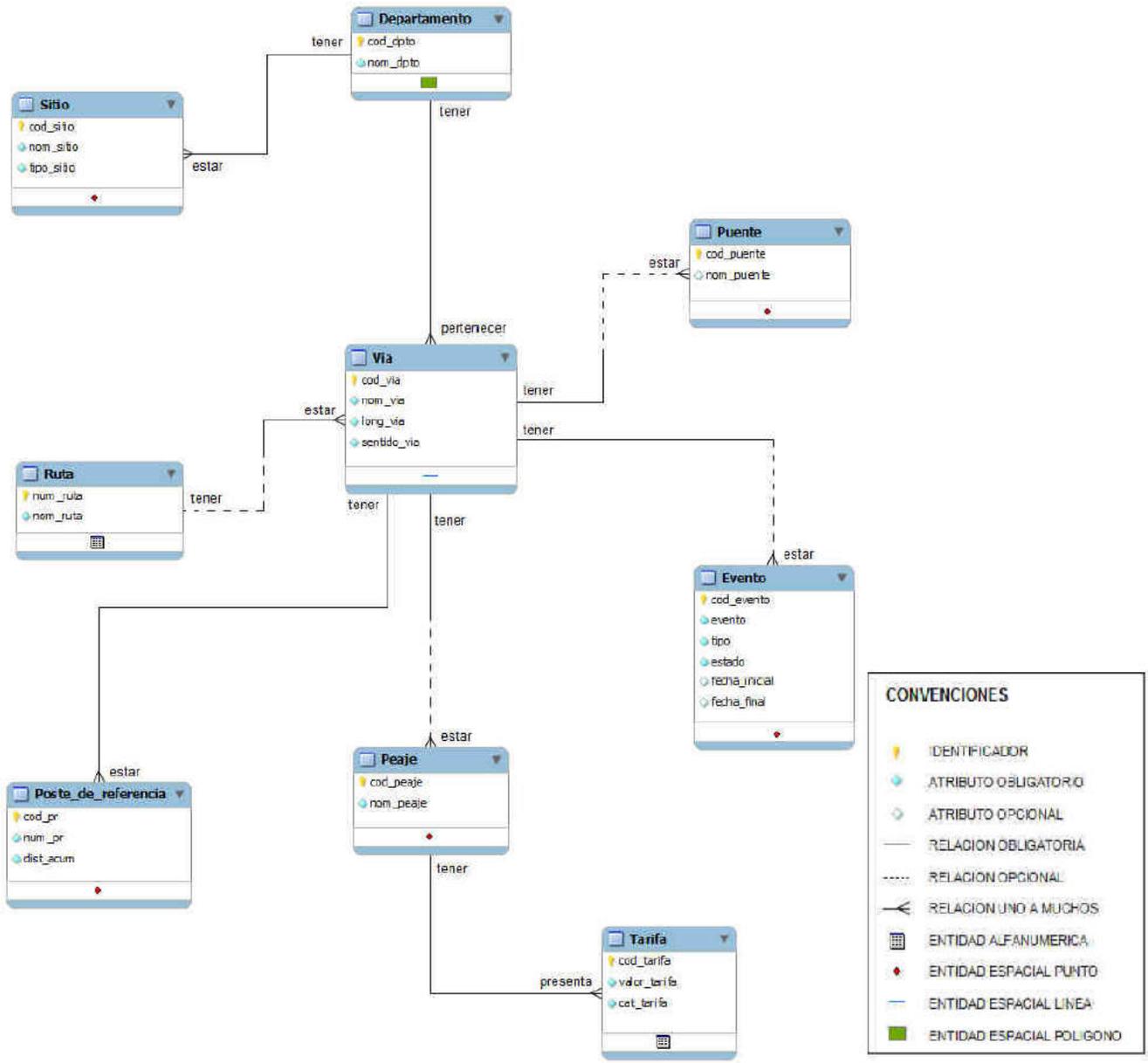


Figura 7. Ejemplo No. 1 de un Modelo Entidad - Relación.

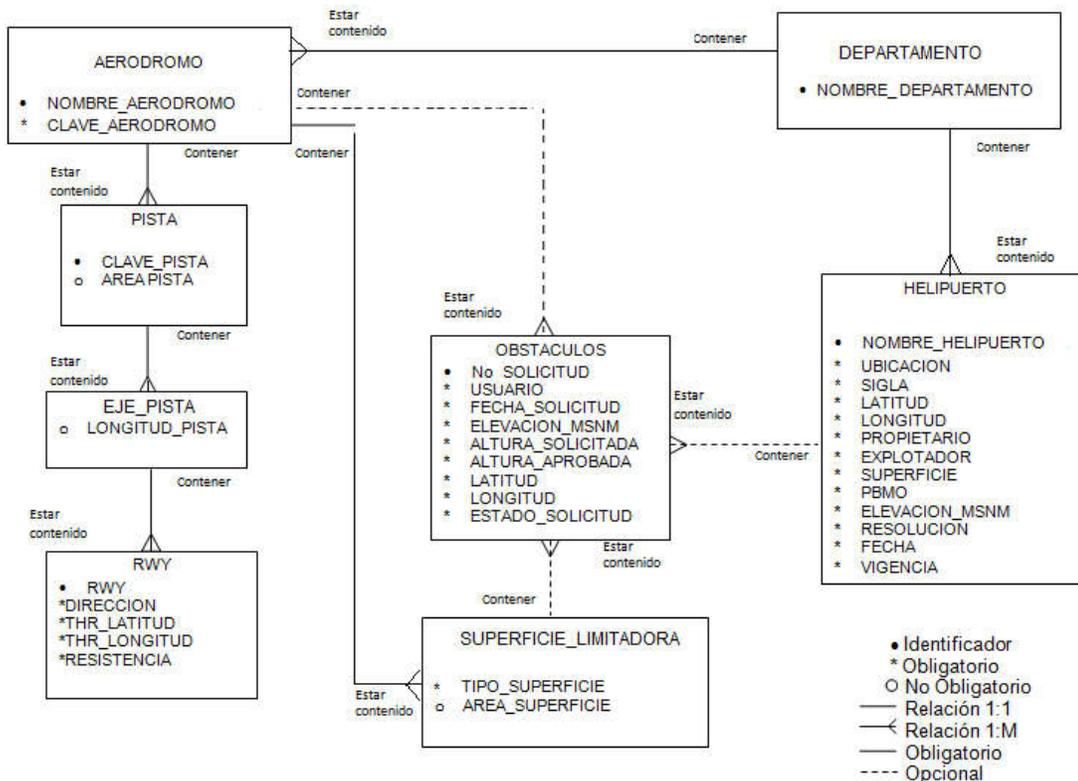


Figura 8. Ejemplo No. 2 de un Modelo Entidad - Relación.

- Modelo Físico (corresponde con un Modelo Relacional)

El Modelo Relacional es un modelo que busca presentar el diseño de las tablas que se requieren elaborar en la implementación. Por tanto se basa en el Modelo E-R, realizado previamente.

Este modelo involucra un “Esqueleto de las Tablas” en el que se especifican aspectos como restricciones semánticas e integridad referencial.

El planteamiento de este modelo es similar a como se diseñaba para una base de datos tradicional. Por ejemplo: Miremos la Tabla de Predio.

Tabla Predio:

Atributo	Pre_Cod	Direccion	Area	Man_cod	Bar_cod	Mat_Inmob
Tipo de Dato	Text	Text	Double	Text	Text	Text
Tamaño	3	50	6	3	6	12
Tipo de Llave	PK	-	-	FPK	FPK	-
Nulidad	NN	NN	NN	NN	NN	NN
Unicidad	-	U	-	-	-	U
Ejemplo	28	Kr 8 # 10-50	342	19	201514	0502835C

Tabla No. 2 1er Ejemplo de un Mod. Relacional (o Modelo de Datos Relacional MDR)

PK: Primary Key    FK: Foreign Key    FPK: Foreign Primary Key    NN: No Nulo (Obligatorio)    U: Único

Observe que si usted hiciera la Tabla para la Entidad Barrio, en ese caso, el Atributo (Columna) Bar\_Cod es PK, NN y es U (Único). Mientras que en la Tabla de Predio el atributo Bar\_Cod no es un atributo único. Tenga en cuenta que de acuerdo con el tipo de relación se heredan o no atributos de otras entidades.

En las siguientes figuras se aprecia un segundo Ejemplo de un Modelo Relacional (o Modelo de Datos Relacional MDR) (ver figura 9):

Departamento

Departamento		
NOMBRE ATRIBUTO	COD_DPTO	NOM_DPTO
TIPO DE DATO	Char	Char
TAMAÑO	2	25
LLAVE	PK	
OBLIGATORIO	SI	SI
UNICIDAD	SI	SI

Tabla 2. Modelo relacional para la tabla de departamento

Sitio de interés

Sitio				
NOMBRE ATRIBUTO	COD_SITIO	TIPO_SITIO	NOM_SITIO	COD_DPTO
TIPO DE DATO	Char	Char	Char	Char
TAMAÑO	8	50	50	2
LLAVE	PK			PFK
OBLIGATORIO	SI	SI	SI	SI
UNICIDAD	SI	NO	NO	NO

Tabla 3. Modelo relacional para la tabla de sitio

Poste de Referencia

Pr						
NOMBRE ATRIBUTO	COD_PR	NUM_PR	DIST_ACUM	COD_VIA	COD_DPTO	NUM_RUTA
TIPO DE DATO	Char	Char	Double	Char	Char	Char
TAMAÑO	17	3		14	2	2
LLAVE	PK			PFK	PFK	PFK
OBLIGATORIO	SI	SI	SI	SI	SI	SI
UNICIDAD	SI	NO	NO	NO	NO	NO

Tabla 6. Modelo relacional para la tabla de poste de referencia

Estación de Peaje

Peaje					
NOMBRE ATRIBUTO	COD_PEAJE	NOM_PEAJE	COD_VIA	COD_DPTO	NUM_RUTA
TIPO DE DATO	Char	Char	Char	Char	Char
TAMAÑO	16	25	14	2	2
LLAVE	PK		PFK	PFK	PFK
OBLIGATORIO	SI	SI	SI	SI	SI
UNICIDAD	SI	NO	NO	NO	NO

Tabla 7. Modelo relacional para la tabla de peaje

Figura 9. Ejemplo No. 2 de un Modelo de Datos Relacional

También es necesario realizar el DICCIONARIO DE DATOS (o llamado también Diccionario de Objetos y Atributos) como parte del diseño de la Base de Datos Espacial.

Ejemplo de un Diccionario de Datos (o Diccionario de Objetos y Atributos) (ver figura 10 a y 10 b):

DEPARTAMENTO:

Es la entidad territorial principal por la cual se divide administrativa y políticamente el territorio nacional. Estos a su vez forman regiones geográficas, culturales y económicas.

Atributos:

- COD\_DPTO: es el identificador del departamento.  
Tipo de dato: Char (2). Tiene un tamaño de 2, ya que este corresponde a la codificación del Invias.
  
- NOM\_DPTO: corresponde al nombre del departamento.  
Tipo de dato: Char (25).

SITIO:

Lugar que representa una ubicación de interés, así como un sitio geográfico y que es tenido en cuenta como una referencia importante.

Atributos:

- COD\_SITIO: es el identificador del lugar.  
Tipo de dato: Char (8).
  
- TIPO\_SITIO: hace referencia a la clase de lugar de interés.  
Tipo de dato: Char (50).
  
- NOM\_SITIO: corresponde al nombre del lugar.  
Tipo de dato: Char (50).
  
- COD\_DPTO: es el identificador del departamento al que pertenece el lugar de interés.  
Tipo de dato: Char (2). Tiene un tamaño de 2, ya que este corresponde a la codificación del DANE.

Figura 10 a. Ejemplo de Diccionario de Datos

## PR:

Puntos con coordenadas conocidas ubicados a los extremos derecho e izquierdo de la vía, los cuales se encuentran representados por postes y el cual en su conjunto constituyen el marco del sistema de referencia.

## Atributos:

- COD\_PR: es el identificador del punto de referencia.  
Tipo de dato: Char (17). El cual está compuesto por 6 dígitos del código de la vía, más 3 dígitos del número del PR.
  
- NUM\_PR: corresponde al número del PR.  
Tipo de dato: Char (3).
  
- DIST\_ACUM: hace referencia a la distancia acumulada desde el principio de una vía específica hasta ese punto.  
Tipo de dato: Double.
  
- COD\_VIA: es el identificador de la vía al que pertenece el punto de referencia.  
Tipo de dato: Char (14).
  
- COD\_DPTO: es el identificador del departamento al que pertenece el punto de referencia.  
Tipo de dato: Char (2). Tiene un tamaño de 2, ya que este corresponde a la codificación del DANE.
  
- NUM\_RUTA: es el identificador de la ruta al que pertenece el punto de referencia.  
Tipo de dato: Char (2).

Figura 10 b. Continuación Ejemplo de Diccionario de Datos

## **Importante:**

La variación importante entre el diseño del Modelo Físico en una Base de Datos Tradicional frente a una Base de Datos Espacial está dada por:

Las Relaciones Muchos a Muchos entre entidades, el resto de relaciones se solucionan de forma similar cuando se pasa de un Modelo Entidad – Relación al Modelo de Datos Relacional.

En relaciones muchos a muchos podemos tener:

- Una Entidad no espacial y otra entidad no espacial: en este caso la relación se soluciona con una Tabla de Paso.
- Una Entidad No Espacial y otra Entidad Espacial: en este caso la relación se soluciona con una Tabla de Paso.

Ejemplo: Predios y Propietarios.

- Una Entidad Espacial y otra Entidad Espacial: en este caso la relación no se soluciona con una Tabla de Paso, sino que esta relación se resuelve a través de procesos de superposición.

Ejemplo:

- Vías (Lineal) y Localidad (Poligonal) – Superposición Línea en Polígono.
- Estrato (Poligonal) y Localidad (Poligonal) – Superposición Polígono en Polígono.

#### **Otras Actividades de la Etapa de Diseño son:**

- Luego de Diseñar la Base de Datos Espacial que tendrá el SIG, se debe realizar una Clasificación y Codificación de los elementos de acuerdo con la Clasificación que tienen los Modelos de Datos o Catálogos de Objetos CO-025 y CO-02000 que tiene el Instituto Geográfico Agustín Codazzi (IGAC) para objetos geográficos a escalas 1:25000 (Zona Rural) y 1:2000 (Zona Urbana) respectivamente. Con esto lo que se pretende es organizar y codificar la información teniendo en cuenta unos marcos de referencia o estándares definidos previamente.
- Luego se realiza un diseño de la Simbolización que va a tener cada uno de los elementos para organizar como estos se presentarán en una posible salida gráfica (mapa o ploteo).

Para los símbolos se definen grosores de líneas, colores, especificando medidas.

Es necesario realizar el diseño del Catálogo de Objetos y Símbolos, realizando la respectiva clasificación de los objetos o elementos (ver figura 11):

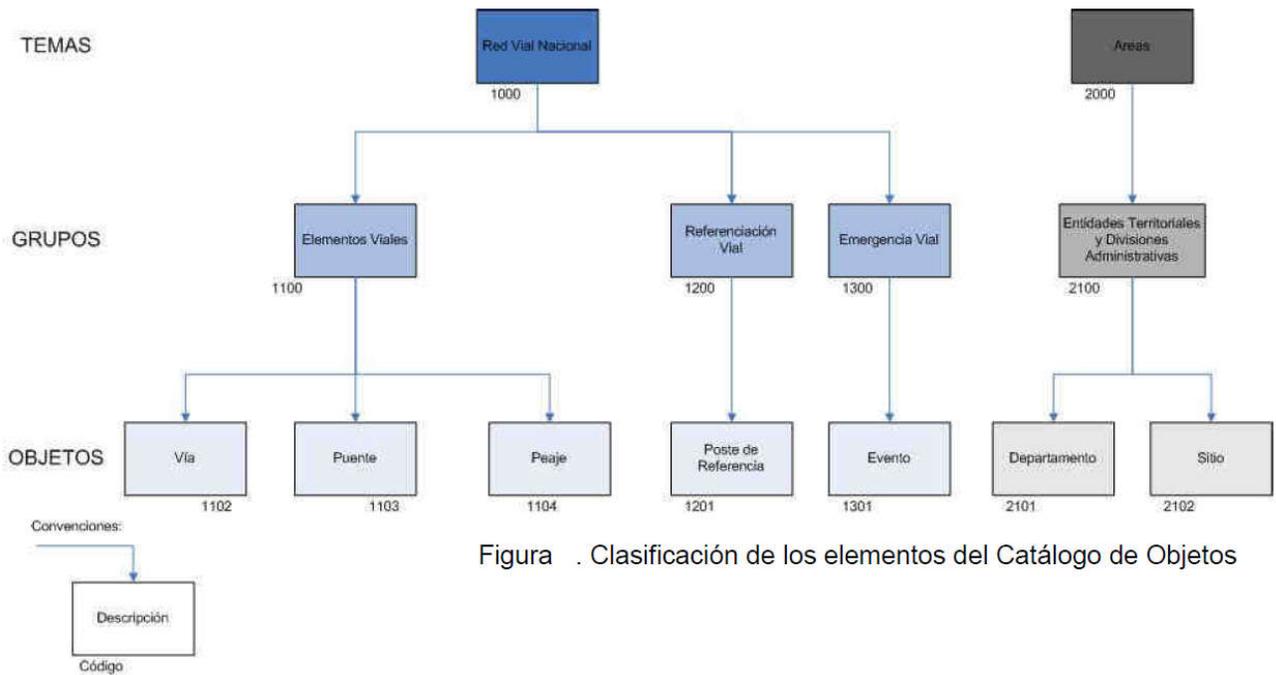


Figura . Clasificación de los elementos del Catálogo de Objetos

Figura 11. Ejemplo de la Clasificación de los elementos del Catálogo de Objetos:

Ejemplo de la plantilla del Catálogo de Objeto y de Símbolo para un elemento (en este caso para el elemento u objeto *Peaje*). (Ver tabla 3)

CATÁLOGO DEL OBJETO			
<p><b>TEMA:</b> Red vial nacional.</p> <p><b>GRUPO:</b> Elementos viales.</p> <p><b>NOMBRE DEL OBJETO:</b> Peaje.</p> <p><b>CODIGO DEL OBJETO:</b> 1104.</p> <p><b>TIPO DE OBJETO:</b> Punto.</p> <p><b>DEFINICION:</b> Caseta con una posición geográfica destinada al cobro de dinero, con el cual se otorga derecho para poder circular por una determinada vía.</p>			
<p><b>ATRIBUTOS DEL OBJETO:</b></p> <ul style="list-style-type: none"> <li>• <b>COD_PEAJE:</b> es el identificador del peaje. Tipo de dato: Char (8). El cual está compuesto por 6 dígitos del código de la vía, mas 2 dígitos del numero del peaje.</li> <li>• <b>NOM_PEAJE:</b> corresponde al nombre del peaje. Tipo de dato: Char (25).</li> <li>• <b>COD_VIA:</b> es el identificador de la vía al que pertenece el peaje. Tipo de dato: Char (6).</li> <li>• <b>COD_DPTO:</b> es el identificador del departamento al que pertenece el peaje. Tipo de dato: Char (2).</li> <li>• <b>NUM_RUTA:</b> es el identificador de la ruta al que pertenece el peaje. Tipo de dato: Char (2).</li> </ul>			
CATÁLOGO DEL SIMBOLO			
SIMBOLO	ESPECIFICACIONES	CODIGO DEL SIMBOLO	FUENTE
<input checked="" type="checkbox"/>	Grosor: 1 mm Color: negro R: 0 G: 0 B: 0	1104	Instituto Nacional de Vías

Tabla . Diseño plantilla del catálogo de objeto y de símbolo para Peaje

Tabla 3. Ejemplo Diseño plantilla del Catálogo de Objeto y de símbolo para Peaje.

Ejemplo de la plantilla del Catálogo de Objeto y de Símbolo para un elemento (en este caso para el elemento u objeto Vía) (ver Tabla 4).

CATÁLOGO DEL OBJETO			
<b>TEMA:</b> Red vial nacional.			
<b>GRUPO:</b> Elementos viales.			
<b>NOMBRE DEL OBJETO:</b> Vía.			
<b>CODIGO DEL OBJETO:</b> 1102.			
<b>TIPO DE OBJETO:</b> Línea.			
<b>DEFINICION:</b> Sistema de comunicación que conduce de un lugar a otro, por medio de áreas destinadas al tránsito vehicular y que comprende tanto troncales como transversales.			
<b>ATRIBUTOS DEL OBJETO:</b>			
<ul style="list-style-type: none"> <li>• <b>COD_VIA:</b> es el identificador de la vía. Tipo de dato: Char (6).</li> <li>• <b>NOM_VIA:</b> corresponde al nombre de la vía. Tipo de dato: Char (50).</li> <li>• <b>LONG_VIA:</b> hace referencia a la longitud de la vía. Tipo de dato: Double.</li> <li>• <b>SENTIDO_VIA:</b> corresponde al sentido de digitalización que tiene la vía. Tipo de dato: Char (1).</li> <li>• <b>COD_DPTO:</b> es el identificador del departamento al que pertenece la vía. Tipo de dato: Char (2). Tiene un rango de 2, ya que este corresponde a la codificación del DANE.</li> <li>• <b>NUM_RUTA:</b> es el identificador de la ruta al que pertenece la vía. Tipo de dato: Char (2).</li> </ul>			
CATÁLOGO DEL SIMBOLO			
SIMBOLO	ESPECIFICACIONES	CODIGO DEL SIMBOLO	FUENTE
—	Grosor: 1 mm Color: negro R: 0 G: 0 B: 0	1102	Instituto Nacional de Vías

Tabla . Diseño plantilla del catálogo de objeto y de símbolo para vía.

Tabla 4. Ejemplo Diseño plantilla del Catálogo de Objeto y de símbolo para Peaje.

- Luego se listan los posibles análisis y modelamientos que se van a realizar en la implementación, con base en los requerimientos de los usuarios y en las funciones del sistema, conceptualizadas previamente.

Para cada uno de los posibles análisis y modelamientos se debe realizar en lo posible un flujograma o diagrama que presente cuales son los datos iniciales necesarios, cuales son los procesos u operaciones de análisis a aplicar, cuáles son los productos o datos intermedios (si los hay) y cuáles son los productos finales. En dichos flujogramas o diagramas es necesario que Ud. sea muy claro en qué tipo de convenciones o notación empleó.

#### Modelamiento Espacial:

Un primer Ejemplo de un Modelamiento Espacial (Tomado del trabajo de Rodolfo Franco) se puede apreciar en la figura No. 12:

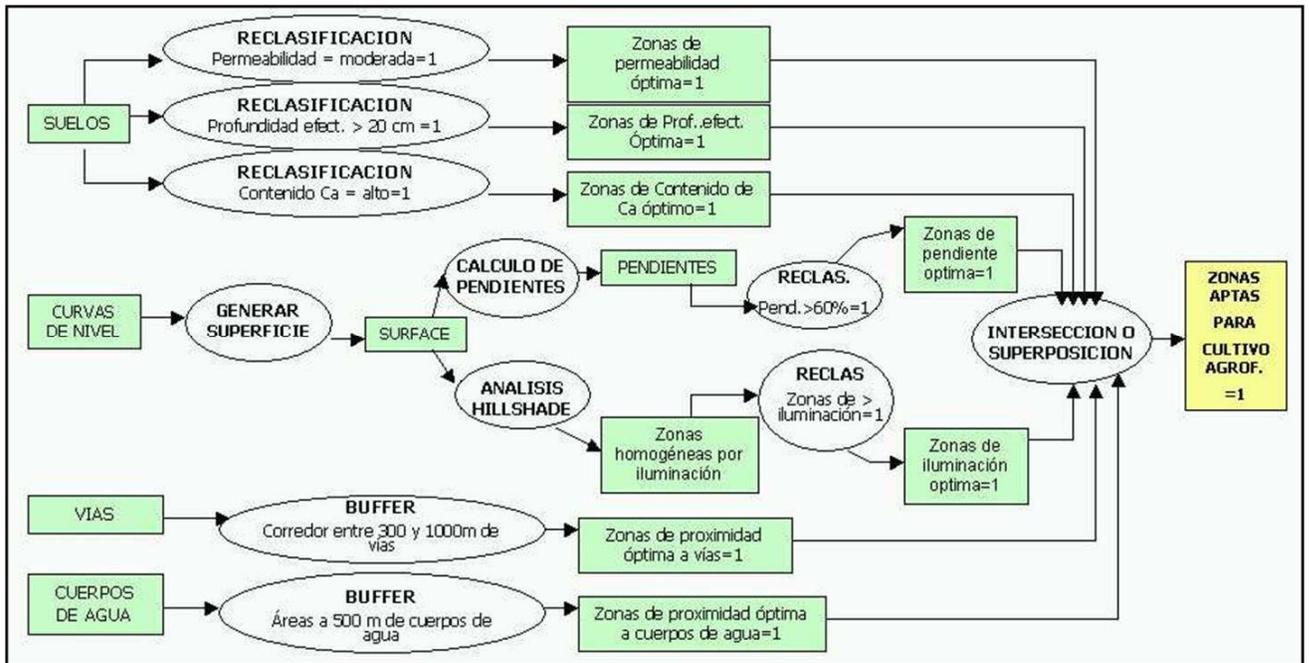


Figura No. 12 Flujograma Operacional del Modelamiento Espacial (Autor: Rodolfo Franco)

NOTA: A este modelo le faltan las convenciones o la notación apropiada para facilitar su lectura y comprensión.

Un segundo ejemplo de un Modelamiento Espacial (ver figura No. 13)

**Mapa de amenazas naturales**

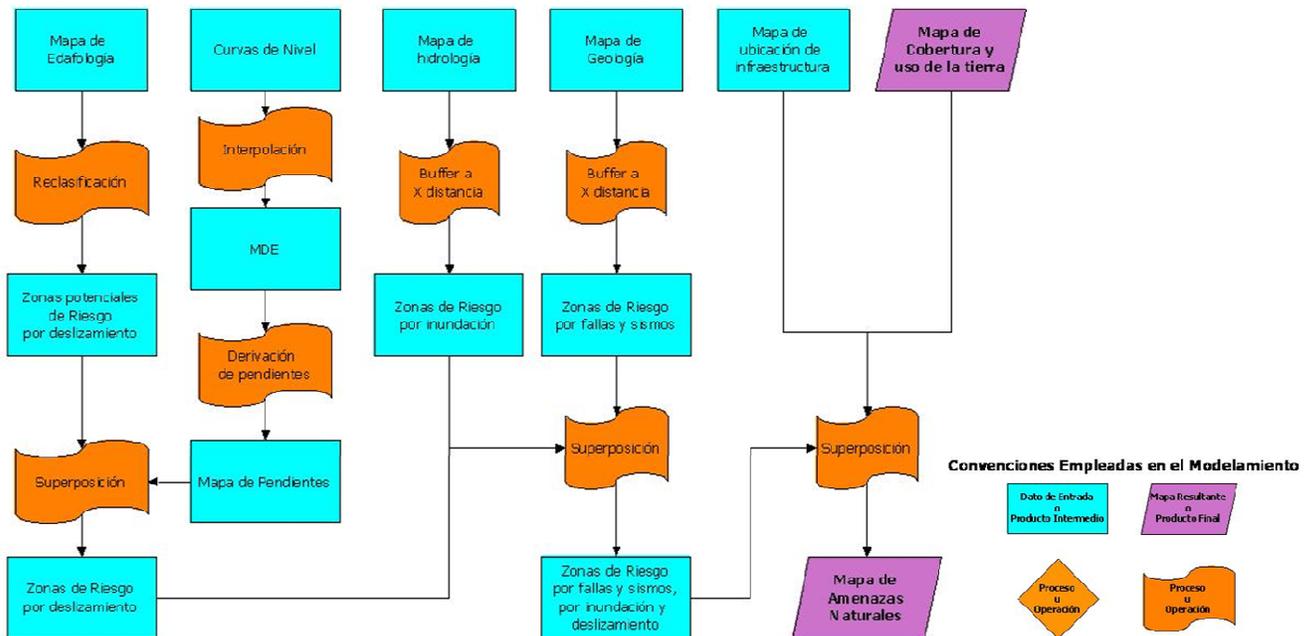


Figura No. 13 Flujograma Operacional del Modelamiento Espacial

Un tercer Ejemplo de un Modelamiento Espacial (ver figura No. 14)

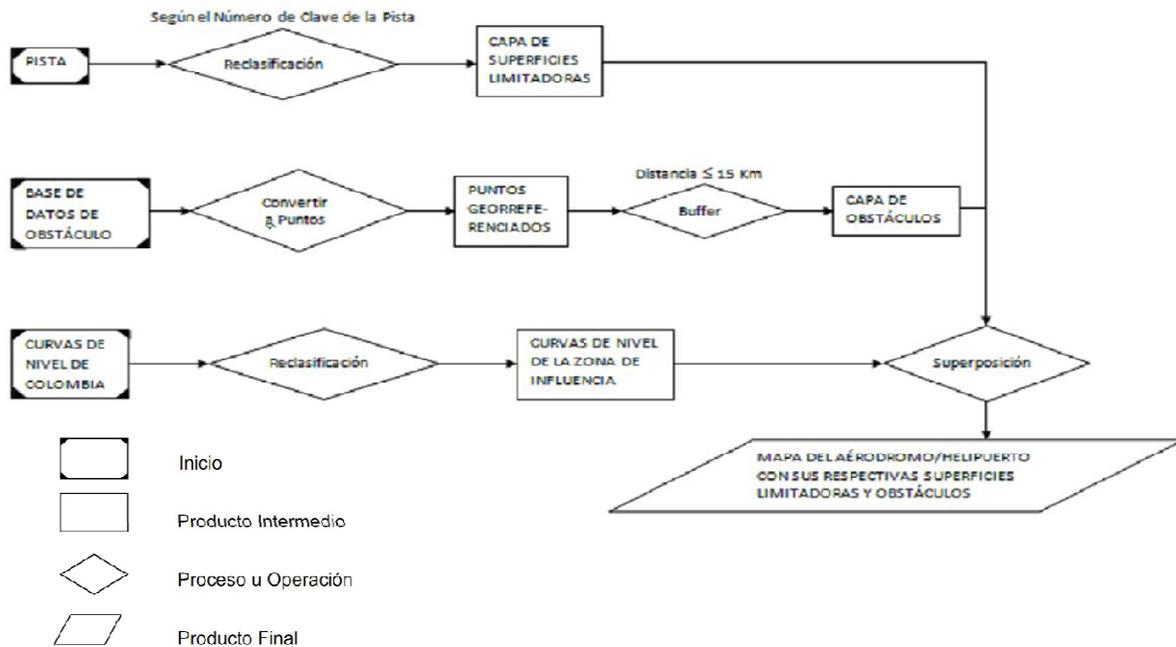


Figura no. 14 Ejemplo No. 3. Diagrama de Flujo para la generación de un Mapa Cartográfico

- Posteriormente, se listan las posibles salidas, resultados y/o productos de la Base de Datos Espacial. Es decir, se indican cuáles serían los posibles productos que se pueden obtener luego de implementar el Sistema.

Tales posibles productos, resultados o salidas pueden ser: salidas gráficas, mapas análogos temáticos, reportes, estadísticas, gráficas, productos obtenidos y/o generados mediante la aplicación de Análisis y Modelamiento Espacial, archivos con algunos resultados o coberturas.

Adicionalmente en esta actividad del Diseño se debe indicar si se va a realizar una interface de usuario final para la Base de Datos Espacial con el fin de facilitar el uso del sistema por parte de los diferentes usuarios. Si dicha interfaz se piensa realizar es necesario delimitar cuál será su alcance y sus funciones, cómo se va a desarrollar, es decir se debe presentar un conciso diseño de tal aplicación.

- Aspecto para tener en cuenta (Importante):  
Es importante nunca perder de vista que tanto durante el desarrollo de la Etapa de Conceptualización como de la Etapa de Diseño no se ha implementado nada en un

software especializado como arcgis, arcview, gvSIG, arcinfo, idrisi, ilwis, qgis, etc., todo se ha desarrollado en papel como resultado de unas discusiones y acuerdos entre el equipo de trabajo que elabora la Base de Datos Espacial. Lo máximo que se ha podido realizar en términos de una sistematización es ir plasmando los resultados y/o conclusiones de estas etapas en un informe escrito realizado con un procesador de palabras como Word, Start Office, Writer, etc.

Es decir, todavía no se ha realizado nada desde el punto de vista práctico u operativo. Todavía no se ha “cacharreado” con algún software o hardware especializado para Bases de Datos Espaciales.

### 5.1.3. IMPLEMENTACION

En esta etapa se toman todos los resultados, modelos, listas de posibles análisis, productos, etc., y a partir de todo lo desarrollado en las dos etapas anteriores: Conceptualización y Diseño, ahora sí se empieza la operativización de la Base de Datos Espacial. Es decir, ahora sí se inicia la implementación de la Base de Datos Espacial desde el punto de vista práctico, operativo, real.

Para hacer la implementación de la Base de Datos Espacial, es decir hacerlo una realidad, algo tangible, es necesario que se empiecen a abordar los Subsistemas de una Base de Datos Espacial: Captura, Almacenamiento, Análisis y Modelamiento y Salida.

De acuerdo con esto lo primero que se debe hacer es mirar que información se requiere de acuerdo con las necesidades de los usuarios y el objetivo de la Base de Datos Espacial y se van realizando en resumen las siguientes actividades (que corresponden con cada uno de los aspectos que abordan los diferentes subsistemas):

- Captura de la Información necesaria.
- Estructuración y almacenamiento adecuado de la información.
- Implementación y Montaje de la Base de Datos Espacial.
- Clasificación y Codificación de los elementos de acuerdo con lo definido previamente.
- Aplicación práctica de la Simbolización propuesta en el Diseño.
- Desarrollo práctico es decir operativo o tangible, de los posibles análisis y modelamientos propuestos para encontrar las soluciones o resultados a las problemáticas detectadas y que se suponía iba a solucionar la Base de Datos Espacial.

- Generación u obtención de las salidas, productos y resultados propuestos en el Diseño.
- Desarrollo e Implementación de la Interfaz Gráfica de Usuario de la Base de Datos Espacial o Interface Final de Usuario de la Base de Datos Espacial (en el caso de que se realice) indicando o realizando el manual de Usuario de dicha aplicación. Este tipo de interfaces requieren para su desarrollo de emplear algún tipo de programación, en un lenguaje público como: C++, Java o un lenguaje propietario y amarrado a un software para SIG como: avenue para arcview 3.X, arcobjects para arcgis, etc.
- Elaboración del Informe o Documento de Soporte de la Base de Datos Espacial. Dicho documento contendrá toda la información y los soportes de las etapas del desarrollo de la Base de Datos Espacial. Allí se indica qué se hizo, cómo se hizo, qué se tuvo en cuenta, para que si alguna persona consulta la documentación de la Base de Datos Espacial tenga la suficiente claridad de lo realizado.

En esta etapa si se emplean o utilizan los materiales, equipos y softwares necesarios para el desarrollo práctico de la Base de Datos Espacial como: cartografía, fotografías aéreas, imágenes satelitales, mesas digitalizadoras, plotters, escáneres, software como arcview, arcinfo, ilwis, geomedia, autocad, Access, Oracle, visio, arcgis, qgis, gvSIG, etc.

En todo el proceso de implementación de la Base de Datos Espacial se debe ir realizando, paralelamente, un control de calidad de los procesos para buscar que estos se realicen adecuada y eficientemente. Estos controles y medidas de calidad también deben documentarse en el informe final de la Base de Datos Espacial.

#### 5.1.4. EVALUACION – VALIDACION DEL SISTEMA

Esta etapa, aun cuando se coloca como la última, se va a desarrollar permanentemente pues tiene como propósito revisar el funcionamiento de la Base de Datos Espacial y corregir o modificar algunas cosas que pudiesen estar mal durante la realización y desarrollo de las etapas anteriores junto con sus actividades, para garantizar que todo quede correcto y adecuado en cada una de ellas y en general en todo el proceso.

*Mantenimiento y Uso del sistema:*

- Definición de zona piloto
- Evaluación y control de los modelos

- Confrontación con la realidad
- Depuración de errores
- Control y definición de estándares
- Personalización de las interfases
- Documentación
- Distribución

## 5.2 PostgreSQL

¿Qué es PostgreSQL? PostgreSQL es un sistema de gestión de base de datos relacional de objetos (ORDBMS) basado en POSTGRES, Versión 4.2, desarrollado en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley. POSTGRES fue pionero en muchos conceptos que solo estuvieron disponibles en algunos sistemas de bases de datos comerciales mucho más tarde. PostgreSQL es un descendiente de código abierto de este código original de Berkeley. Es compatible con una gran parte del estándar SQL y ofrece muchas características modernas: consultas complejas llaves extranjeras desencadena vistas actualizables integridad transaccional control de concurrencia multiversión Además, el usuario puede ampliar PostgreSQL de muchas maneras, por ejemplo, agregando nuevos tipos de datos funciones operadores Funciones agregadas métodos de índice lenguajes de procedimiento Y debido a la licencia liberal, PostgreSQL puede ser utilizado, modificado y distribuido por cualquier persona de forma gratuita para cualquier propósito, ya sea privado, comercial o académico. [34]

### 5.2.1. Sintaxis SQL

#### Estructura léxica

La entrada SQL consiste en una secuencia de comandos. Un comando se compone de una secuencia de tokens, terminada por un punto y coma (;). El final de la secuencia de entrada también termina un comando. Los tokens que son válidos dependen de la sintaxis del comando en particular.

Un token puede ser una palabra clave, un identificador, un identificador entre comillas, un literal (o constante), o un símbolo de carácter especial. Normalmente, los tokens están separados por espacios en blanco (espacio, tabulador, nueva línea), pero no es necesario

que no existan ambigüedades (lo que generalmente ocurre solo si un carácter especial es adyacente a algún otro tipo de token).

Por ejemplo, lo siguiente es una entrada de SQL válida (sintácticamente):

```
SELECT * FROM MY_TABLE;  
ACTUALIZAR MY_TABLE SET A = 5;  
Insertar en mis valores de tabla (3, 'hola');
```

Esta es una secuencia de tres comandos, uno por línea (aunque no es necesario; puede haber más de un comando en una línea, y los comandos pueden dividirse entre líneas).

Además, los comentarios pueden ocurrir en la entrada de SQL. No son fichas, son efectivamente equivalentes a espacios en blanco.

La sintaxis SQL no es muy consistente en cuanto a qué tokens identifican los comandos y cuáles son operandos o parámetros. Los primeros tokens son generalmente el nombre del comando, por lo que en el ejemplo anterior generalmente hablamos de un "SELECCIONAR", un "ACTUALIZACIÓN" y un comando "INSERTAR". Pero, por ejemplo, el comando ACTUALIZAR siempre requiere que aparezca un token SET en una posición determinada, y esta variación particular de INSERT también requiere VALORES para poder completarse.

Expresiones de valor Las expresiones de valor se utilizan en una variedad de contextos, como en la lista de destino del comando SELECT, como nuevos valores de columna en INSERTAR o ACTUALIZAR, o en condiciones de búsqueda en varios comandos. El resultado de una expresión de valor a veces se denomina escalar, para distinguirlo del resultado de una expresión de tabla (que es una tabla). Por lo tanto, las expresiones de valor también se denominan expresiones escalares (o incluso expresiones simples). La sintaxis de expresión permite el cálculo de valores a partir de partes primitivas mediante operaciones aritméticas, lógicas, de conjuntos y otras.

### 5.3. PostGIS

PostGIS es un extensor de base de datos espacial para la base de datos relacional de objetos PostgreSQL. Agrega soporte para objetos geográficos permitiendo que las consultas de ubicación se ejecuten en SQL.

Además del conocimiento de ubicación básico, PostGIS ofrece muchas características que rara vez se encuentran en otras bases de datos espaciales de la competencia, como Oracle Locator / Spatial y SQL Server.

#### 5.4. MySQL

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

MySQL se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, MySQL se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP. LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos (a veces, Perl o Python se utiliza en lugar de PHP).

MySQL, que fue concebido originalmente por la compañía sueca MySQL AB, fue adquirida por Oracle en 2008. Los desarrolladores todavía pueden usar MySQL bajo la Licencia Pública General de GNU (GPL), pero las empresas deben obtener una licencia comercial de Oracle.

#### 5.5 Comparación entre Postgre SQL y My SQL

Al comparar "MySQL vs PostgreSQL, ambos sistemas de bases de datos, si bien en primera instancia pareciera que funcionan y realizan las mismas tareas, difieren en muchas cosas. Desde la forma en la que el desarrollador debe interactuar con ellos, hasta la eficiencia e incluso el propósito.

En muchos casos utilizamos MySQL por comodidad, o por el hecho de que es el gestor más popular y conocido, pero sin duda hay en muchas ocasiones en las cuales el uso de PostgreSQL no solamente es bienvenido sino que es totalmente recomendado, tengamos en cuenta que no es por menos que es visto como el sistema de gestión de bases de datos más avanzado, además de que es totalmente libre.

Aunque para aquellos casos en los cuales lo que necesitaremos es más que nada leer las tablas, y el volumen de datos no es demasiado, MySQL sin duda es la mejor de las opciones.” [35]

A continuación, se muestra una tabla (No. 5) resumen de la comparación entre PostgreSQL y MySQL [36]

<b>Característica</b>	<b>PostgreSQL</b>	<b>MySQL</b>
<i>Código abierto</i>	Completamente de código abierto	De código abierto, pero propiedad de Oracle y ofrece versiones comerciales
<i>Cumplimiento de ACID</i>	Cumple completamente con ACID	Algunas versiones cumplen
<i>Cumplimiento de SQL</i>	Cumple casi completamente	Algunas versiones cumplen
<i>Soporte para concurrencia</i>	Implementación de MVCC soporta múltiples peticiones sin bloquear lecturas	Soporte en algunas versiones
<i>Seguridad</i>	Seguro desde las bases con soporte SSL	Soporte SSL en algunas versiones
<i>Soporte para NoSQL/JSON</i>	Varias funcionalidades soportadas	Solo soporte para datos JSON
<i>Métodos de acceso</i>	Soporta todos los estándares	Soporta todos los estándares
<i>Replicación</i>	Múltiples tecnologías de replicación disponibles: <ul style="list-style-type: none"> <li>• Un solo maestro a un standby</li> <li>• Un solo maestro a varios standbys</li> <li>• Replicación de flujo/hot standby</li> <li>• Replicación bidireccional</li> <li>• Replicación lógica por flujo de logs</li> </ul>	Replicación estándar maestro-standby: <ul style="list-style-type: none"> <li>• Un solo maestro a un standby</li> <li>• Un solo maestro a varios standbys</li> <li>• Un solo maestro a un standby a uno o más standbys</li> <li>• Replicación circular (A hacia B hacia C y de vuelta hacia A)</li> <li>• Maestro a maestro</li> </ul>
<i>Vistas materializadas</i>	Soportado	No soportado
<i>Tablas temporales</i>	Soportado	Soportado
<i>Datos geoespaciales</i>	Soportado	Soportado
<i>Lenguajes de programación</i>	Soportado	No soportado
<i>Sistema de tipos extensible</i>	Soportado	No soportado

Tabla 4. Comparación entre PostgreSQL y MySQL [36]

## **6. DISEÑO DEL METAMODELO PARA BASES DE DATOS ESPACIALES.**

### 6.1 Diseño del metamodelo para la sintaxis abstracta del DSL

En esta etapa se diseñará el metamodelo correspondiente que representará la sintaxis abstracta del DSL para Bases de datos espaciales que permita representar relaciones (tablas), geometría, objetos, referencia espacial, relaciones entre objetos, etc.,

Para iniciar con el desarrollo del DSL visual fue necesario partir de un proyecto EMF (Eclipse Modeling Framework) en la IDE Eclipse, y crear dos elementos importantes para el desarrollo, un .ecore (modelo de dominio definido usando Ecore) y un .aird (representación de punto de vista), estos dos elementos están totalmente asociados.

En el modelo .aird se define el modelo EClases, en este se realizó la abstracción de los elementos básicos del problema del modelamiento de bases de datos espaciales sobre herramientas libres (PostgreSQL-PostGIS y MySQL-MySQL Spatial).

En el caso de la herramienta los usuarios pueden tener todos los sistemas de referencia definidos por la OGC, la creación de relaciones (tablas) tanto alfanuméricas como espaciales y sus respectivas relaciones

Después de definidas las EClases (Entidades), fue necesario definir los atributos pertinentes de cada una de ellas y la forma de relacionarse entre ellas.

### 6.2 Sintaxis Abstracta del Metamodelo.

En esta etapa se representa el metamodelo en diferentes formatos, posiblemente como un diagrama de clases, para ello posiblemente se utilice el framework EMF/Ecore dentro de la IDE Eclipse. Para diseñar y representar esta sintaxis abstracta del Metamodelo en forma gráfica se puede utilizar un Diagrama Ecore de la herramienta Ecore Tools.

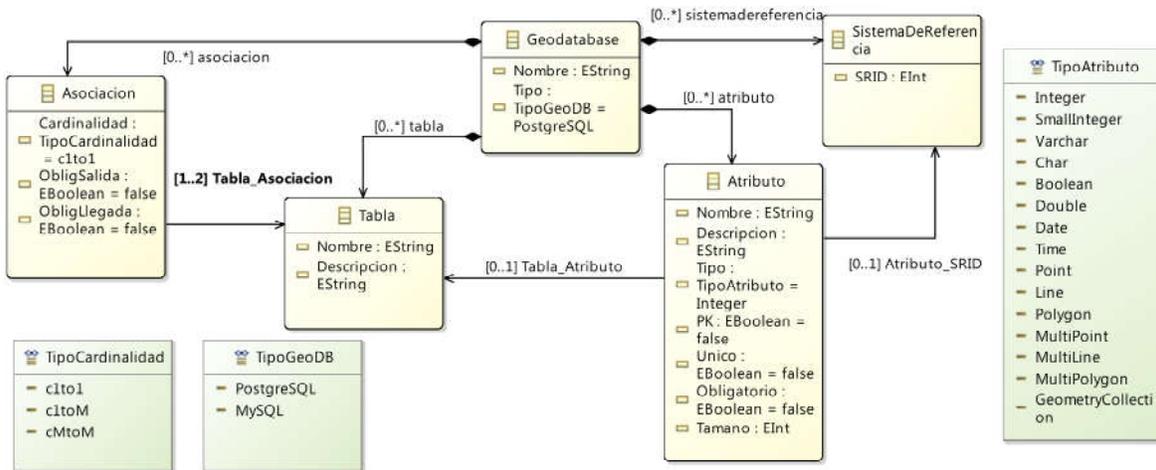


Figura 15. Diagrama de EClases. (Fuente: Elaboración propia)

En dicho diagrama (figura 15) se presentan las diferentes EClases con los demás atributos definidos para cada una de ellas. También se presentan las diferentes asociaciones o interrelaciones entre las Eclases (entidades) con su respectiva multiplicidad o cardinalidad, este diagrama emplea la notación definida por UML.

## **7. SINTAXIS CONCRETA DEL METAMODELO PARA BASES DE DATOS ESPACIALES**

### **7.1 Sintaxis concreta del metamodelo con un Editor Gráfico:**

En esta actividad se desarrollará un Editor textual que permita crear una sintaxis concreta del metamodelo anterior, con el fin de escribir las instrucciones y la gramática en el lenguaje de propósito específico definido, para ello se utilizó Sirius; y en particular se empleó OBEO Designer que cuenta con la posibilidad de trabajar con Eclipse + Sirius de forma integrada.

La herramienta SIRIUS fue utilizada en la creación de Viewpoint del metamodelo, para ello fue necesario agregar las dependencias del metamodelo. El nuevo Viewpoint cuenta un conjunto de elementos de representación que ayudarán visualmente al entendimiento y trabajo del usuario final, como son el caso de los diagramas. Se agregó una representación del tipo de diagrama y se estableció las clases dominio. Sirius se apoya en layers o capas que exponen los elementos relevantes del modelo, como son los nodos y las relaciones. Sirius puede apoyarse en la utilización de elementos gráficos, figuras simples o imágenes que agregan mayor grado de usabilidad e intuición al diagrama. Finalmente, Sirius permite la adición de herramientas de creación de nodos y relaciones con el fin de que el usuario pueda agregar en tiempo de ejecución las entidades necesarias para el modelamiento. El DSL preliminar se puede ver en la figura 16, que corresponde a la vista funcional del mismo.



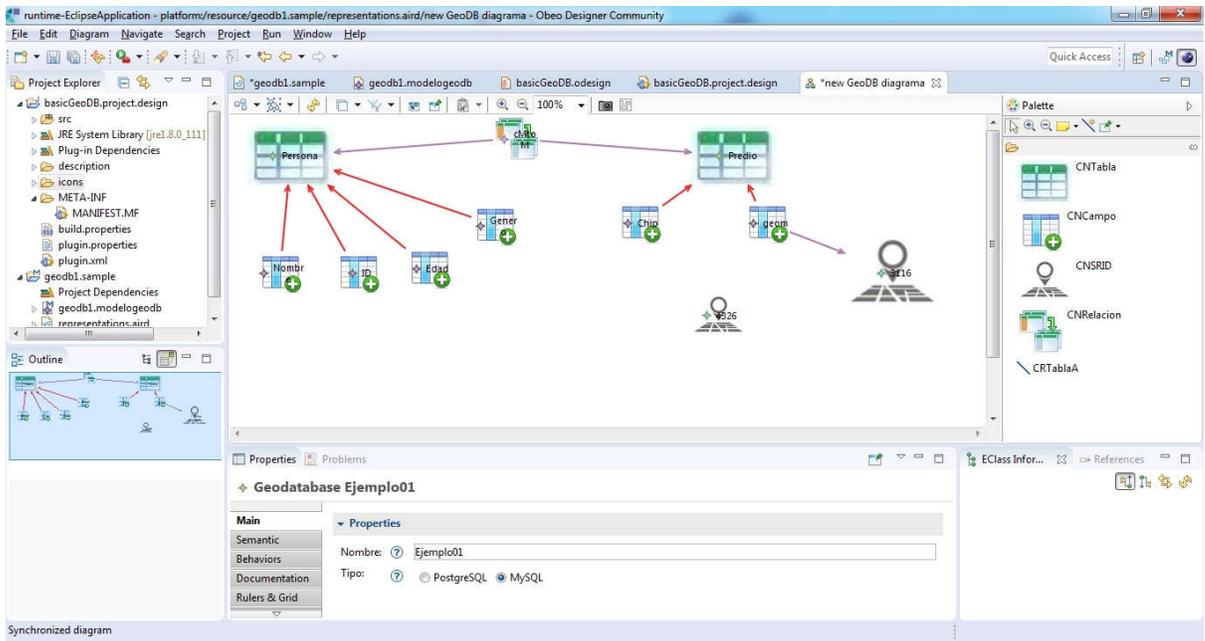


Figura 18. Vista funcional del DSL GeoDB Diagrama en Sirius, junto con su paleta respectiva. (Fuente: Elaboración propia)

Dentro de los nodos definidos en el DSL Geodatabase, se tienen (ver tabla 5):

Nodo	Elemento gráfico asociado
Relación (Tabla)	 CNTTabla
Atributo (Campo)	 CNCampo
Asociación (Relación)	 CRTTablaA
Sistema de Referencia	 CNSRID

Tabla 5. Nodos del DSL Geodatabase. (Fuente: Elaboración propia)

En Sirius se definieron diferentes relaciones entre dichos nodos para el DSL propuesto.

La herramienta permite crear relaciones (tablas), relaciones espaciales, metadatos, sistemas de referencia y análisis espaciales concretos, todos establecidos en el

metamodelo.

➤ *TRANSFORMACION DE MODELOS*

- Actividad 1: Realizar las transformaciones de modelos para obtener el despliegue de una herramienta que facilite el diseño e implementación de Bases de Datos Espaciales Relacionales.

GENERACION DEL EDITOR VISUAL:

Para poder obtener un editor visual de nuestro DSL es necesario llevar a cabo una serie de acciones. Éstas se listan a continuación:

- Abrimos un entorno Eclipse y seleccionamos el espacio de trabajo (workspace) donde se encuentran los diferentes proyectos de nuestro DSL.
- En Sirius (OBEO Designer (Eclipse + Sirius)), luego de emplear el DSL se genera un archivo llamado en este caso *geobde.modelogb* que corresponde con un archivo de tipo XML que consta de todos los elementos generados o creados con el DSL gráfico *GeoDB*, como: tipo de base de datos según el Sistema Gestor de la Base de Datos – SGBD deseado (bien sea PostgreSQL o MySQL), las Relaciones (tablas) que se requieran, sus atributos y asociaciones (relaciones), entre otras.
- Dicho archivo XML es el que se debe transformar para generar el código SQL respectivo según el tipo de SGBD requerido.

La estructura del archivo *geodb1.modelogeodb*, que es de tipo XML, (ver figura 19) es:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <geodb:Geodatabase xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:geodb="http://modelogeodb" Nombre="Ejemplo01" Tipo="MySQL">
3   <tabla Nombre="Persona" Descripcion="Datos de la persona"/>
4   <tabla Nombre="Predio" Descripcion="Unidad Mininima Administrativa"/>
5   <tabla Nombre="miremos"/>
6   <asociacion Cardinalidad="cMtoM" ObligSalida="true" ObligLlegada="true" Tabla_Asociacion="//@tabla.1 //@tabla.0"/>
7   <sistemadereferencia SRID="3116"/>
8   <atributo Nombre="Nombre" Tipo="Varchar" Obligatorio="true" Tabla_Atributo="//@tabla.0"/>
9   <atributo Nombre="ID" Descripcion="Identificador de la Persona" FK="true" Unico="true" Obligatorio="true" Tabla_Atributo="//@tabla.0"/>
10  <atributo Nombre="Edad" Tabla_Atributo="//@tabla.0"/>
11  <atributo Nombre="Genero" Tipo="Boolean" Obligatorio="true" Tabla_Atributo="//@tabla.0"/>
12  <atributo Nombre="Chip" Descripcion="Chip del Predio" Tipo="Varchar" FK="true" Unico="true" Obligatorio="true" Tabla_Atributo="//@tabla.1"/>
13  <atributo Nombre="geom" Descripcion="geometria del predio" Tipo="Polygon" Obligatorio="true" Atributo_SRID="//@sistemadereferencia.0" Tabla_Atributo="//@tabla.1"/>
14  <atributo Nombre="id" FK="true" Unico="true" Obligatorio="true"/>
15 </geodb:Geodatabase>
16

```

Figura 19 Archivo XML geodb1.modelogeodb (Fuente: Elaboración propia)

Y su codificación (archivo geodb1.modelogeodb) es:

```

<?xml version="1.0" encoding="UTF-8"?>
<geodb:Geodatabase xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
  xmlns:geodb="http://modelogeodb" Nombre="Ejemplo01">
  <tabla Nombre="Persona" Descripcion="Datos de la persona"/>
  <tabla Nombre="Predio" Descripcion="Unidad Mininima Administrativa"/>
  <tabla Nombre="miremos"/>
  <asociacion Cardinalidad="cMtoM" ObligSalida="true" ObligLlegada="true"
    Tabla_Asociacion="//@tabla.1 //@tabla.0"/>
  <sistemadereferencia SRID="3116"/>
  <atributo Nombre="Nombre" Tipo="Varchar" Obligatorio="true"
    Tabla_Atributo="//@tabla.0"/>
  <atributo Nombre="ID" Descripcion="Identificador de la Persona" PK="true"
    Unico="true" Obligatorio="true" Tabla_Atributo="//@tabla.0"/>
  <atributo Nombre="Edad" Tabla_Atributo="//@tabla.0"/>
  <atributo Nombre="Genero" Tipo="Boolean" Obligatorio="true"
    Tabla_Atributo="//@tabla.0"/>

```

```

<atributo Nombre="Chip" Descripcion="Chip del Predio" Tipo="Varchar" PK="true"
  Unico="true" Obligatorio="true" Tabla_Atributo="//@tabla.1"/>
<atributo Nombre="geom" Descripcion="geometria del predio" Tipo="Polygon"
  Obligatorio="true" Atributo_SRID="//@sistemadereferencia.0"
  Tabla_Atributo="//@tabla.1"/>
<atributo Nombre="id" PK="true" Unico="true" Obligatorio="true"
  Tabla_Atributo="//@tabla.2"/>
</geodb:Geodatabase>

```

- Actividad 2: Generación de Código:

En esta actividad se seleccionará un sistema generador de código o parser a partir del modelo para generar un lenguaje de programación de dominio específico para facilitar el diseño e implementación de bases de datos espaciales con las extensiones PostGIS y MySQL Spatial, para ello se pudo haber utilizado Acceleo, Java, entre otros.

Para la generación de código, fue necesario desarrollar un parser, que en este caso se realizó en Java empleando la IDE de NetBeans con el lenguaje de programación Java.

El parser se llama parcerGeoDB, que toma el archivo XML resultante del DSL gráfico GeoDB y lo convierte o transforma al lenguaje SQL para alguno de los SGBD considerados (PostgreSQL o MySQL).

Dicho parser, ver figura 20, contiene toda la programación en Java (clases, estructuras de programación, sentencias de control) necesaria para cumplir el propósito definido, teniendo en cuenta como aspecto importante que se trata de una BDE y que los objetos presentes en la misma pueden tratarse de datos geográficos y de no geográficos.

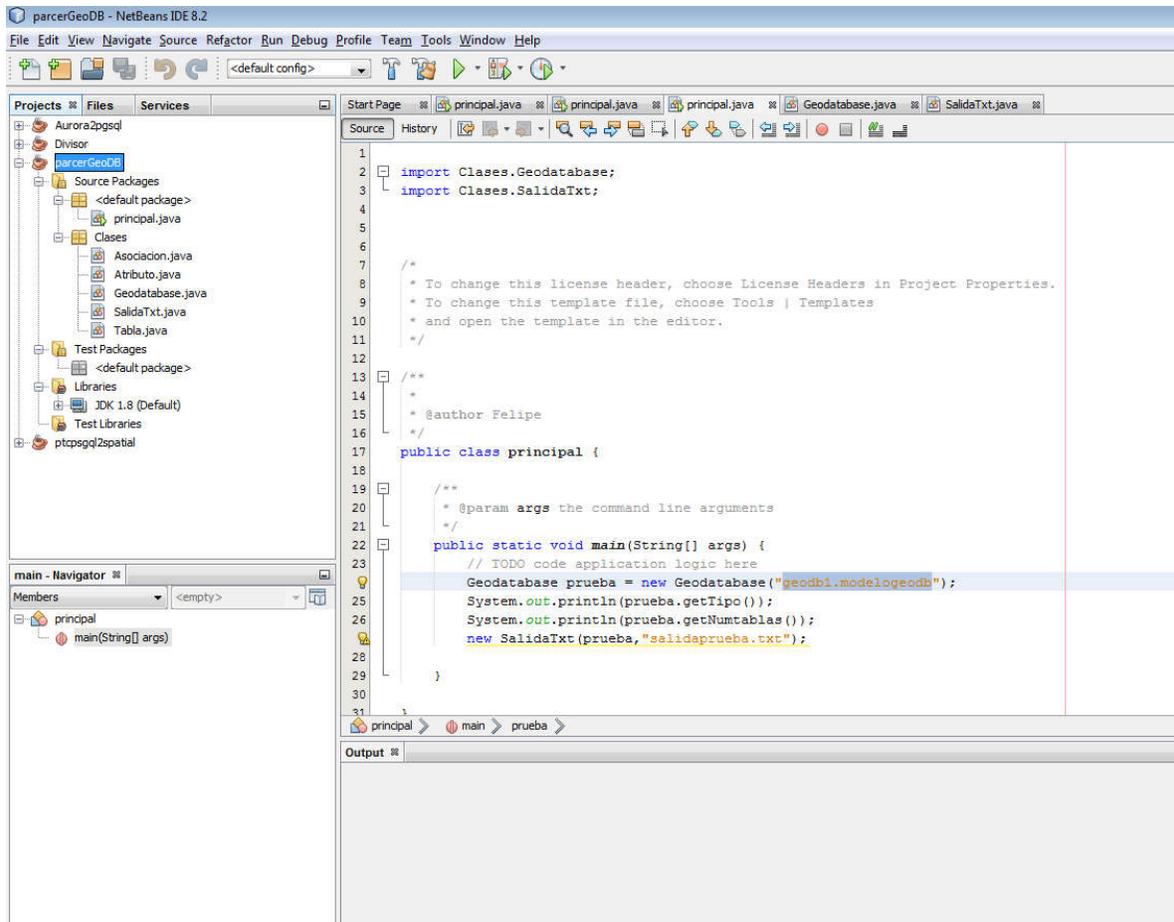


Figura 20. Vista general del Parser ParcerGeoDB, con sus clases.

Como resultado del parcer se genera un archivo llamado salida en formato plano (con extensión TXT o SQL) que puede ser llevado a cualquiera de los dos SGBD disponibles en este caso.

Un ejemplo de la estructura del archivo de salida en formato plano .TXT ó .SQL para el caso de que el SGBD sea PostgreSQL, se encuentra a continuación:

salidaprueba.txt ó salidaprueba.sql para PostgreSQL

```
CREATE TABLE Persona(
Nombre Varchar UNIQUE NULL,
ID Integer UNIQUE NOT NULL,
Edad Integer NULL,
Genero Boolean UNIQUE NULL);
ALTER TABLE Persona ADD PRIMARY KEY (ID);
```

```

CREATE TABLE Predio(
Chip Varchar UNIQUE NOT NULL);
ALTER TABLE Predio ADD PRIMARY KEY (Chip);
SELECT AddGeometryColumn('predio','geom',3116,'LINESTRING',2);

```

```

CREATE TABLE Predio_Persona(
Chip Varchar NOT NULL,
ID Integer NOT NULL);
ALTER TABLE Predio_Persona ADD PRIMARY KEY (Chip,ID);
ALTER TABLE Predio_Persona ADD FOREIGN KEY (Chip) REFERENCES
Predio(Chip);
ALTER TABLE Predio_Persona ADD FOREIGN KEY (ID) REFERENCES
Persona(ID);

```

Un ejemplo de la estructura del archivo de salida en formato plano .TXT ó .SQL para el caso de que el SGBD sea MySQL, se encuentra a continuación:

salidaprueba.txt ó salidaprueba.sql para MySQL

```

CREATE TABLE Persona(
Nombre Varchar(255) UNIQUE NULL,
ID Integer UNIQUE NOT NULL,
Edad Integer NULL,
Genero Boolean UNIQUE NULL);
ALTER TABLE Persona ADD PRIMARY KEY (ID);

```

```

CREATE TABLE Predio(
Chip Varchar(255) UNIQUE NOT NULL,
geom Linestring NULL);
ALTER TABLE Predio ADD PRIMARY KEY (Chip);

```

```

CREATE TABLE Predio_Persona(
Chip Varchar(255) NOT NULL,
ID Integer NOT NULL);
ALTER TABLE Predio_Persona ADD PRIMARY KEY (Chip,ID);

```

```
ALTER TABLE Predio_Persona ADD FOREIGN KEY (Chip) REFERENCES
Predio(Chip);
ALTER TABLE Predio_Persona ADD FOREIGN KEY (ID) REFERENCES
Persona(ID);
```

#### Prueba del parcer en el SGBD

- El archivo plano (TXT o SQL) generado en la actividad anterior se puede llevar a cualquiera de los SGBD y allí abrirlo directamente para convertirlo en archivo con extensión SQL.
- Al ejecutar el archivo SQL, lo que se obtiene es la base de datos espacial creada bien sea en PostgreSQL o en MySQL. Culminando así con el propósito y objetivo del desarrollo del DSL gráfico GeoDB.

Con el archivo que contiene el código transformado a SQL y generado con el parcer GeoDB, para cualquiera de los SGBD trabajados se procede a llevar dicho archivo de salida a cada uno de estos software y se ejecuta o corre el archivo allí en cada uno de los SGBD con el fin de tener de forma exitosa la creación de la base de datos con todos sus componentes. Esto fue posible gracias a que se trata de un sencillo archivo plano, facilitando el proceso de creación de las relaciones o tablas sin requerirse tener conocimientos específicos en el lenguaje propio del SGBD sino realizando el proceso con ayuda del DSL gráfico para BDE.

#### ➤ *ANALISIS DE RESULTADOS*

En esta actividad se analizó los resultados obtenidos al desarrollar el DSL gráfico para facilitar el diseño e implementación de bases de datos espaciales. Para ello previamente a esta actividad, se testeó el DSL gráfico GeoDB, luego el parcer con su archivo de resultado o salida en cada uno de los SGBD encontrando que se cumple con el propósito inicial definido.

#### EVALUACIÓN

La evaluación de la herramienta, proceso que se pretende realizar más adelante, está estrechamente ligada con el objetivo de ofrecer a la comunidad de usuarios de base

de datos espaciales un mecanismo eficiente, intuitivo y con calidad, que les ahorre tiempos de modelamiento y codificación. Para ello se espera realizar pruebas de rendimiento y encuestas de usabilidad.

El proceso de evaluación propuesto, a realizarse en un futuro cercano, para el Prototipo de un Lenguaje de Dominio Especifico para el Modelamiento de Bases de Datos Espaciales consiste en la comparación de tiempos empleados en la solución de un mismo escenario problema, usando DSL Geodatabase y el procedimiento manual que tradicionalmente se ha venido empleando con el fin de tomar tiempos de respuesta, comparar los resultados obtenidos y poder determinar el mejoramiento o no en el desarrollo del proceso de modelamiento de una Base de Datos Espacial, empleando una herramienta de tipo DSL como la propuesta anteriormente.

## Resultados

Se realizaron algunas pruebas y validación de la herramienta DSL gráfica GeoDB. Las pruebas midieron el tiempo y el esfuerzo (usabilidad) de los usuarios para crear tres tipos de modelos de base de datos en cada uno de los SGBD y la herramienta DSL gráfica GeoDB creada. Los modelos de base de datos tienen las características de:

1. Modelo de bases de datos para solo entidades u objetos no espaciales.
2. Modelo de bases de datos para entidades u objetos espaciales y entidades u objetos no espaciales.
3. Modelo de bases de datos para solo entidades u objetos espaciales.

El esfuerzo se puede definir como: *“cantidad de ocasiones en las que usuario selecciona o ingresa algún tipo de información al sistema”*, de acuerdo con Yamada, Hishitani, y Osaki (1993).

El tiempo y el esfuerzo fueron medidos para cada uno de los modelos, realizando el proceso en el mismo orden y con las mismas condiciones para cada uno de los tres modelos y en cada uno de las tres herramientas (PostgreSQL, MySQL y DSL gráfico GeoDB), así como también se estableció el tiempo y esfuerzo total en la creación de los tres modelos de bases de datos.

La figura 21 muestra los tiempos promedio en segundos, que los usuarios invirtieron en la

creación de cada uno de los modelos de base de datos con cada una de las herramientas. Allí se aprecia que con PostgreSQL se tarda más tiempo que con My SQL y a su vez se aprecia que el DSL gráfico mejora los tiempos de los dos anteriores.

La figura 22 muestra un comportamiento y una situación muy similar en cuanto al esfuerzo promedio de los usuarios en la creación de los tres modelos de base de datos. Allí se aprecia que el DSL gráfico en términos de esfuerzo (usabilidad) resulta ser más eficiente que al hacerlo empleando PostgreSQL o MySQL.

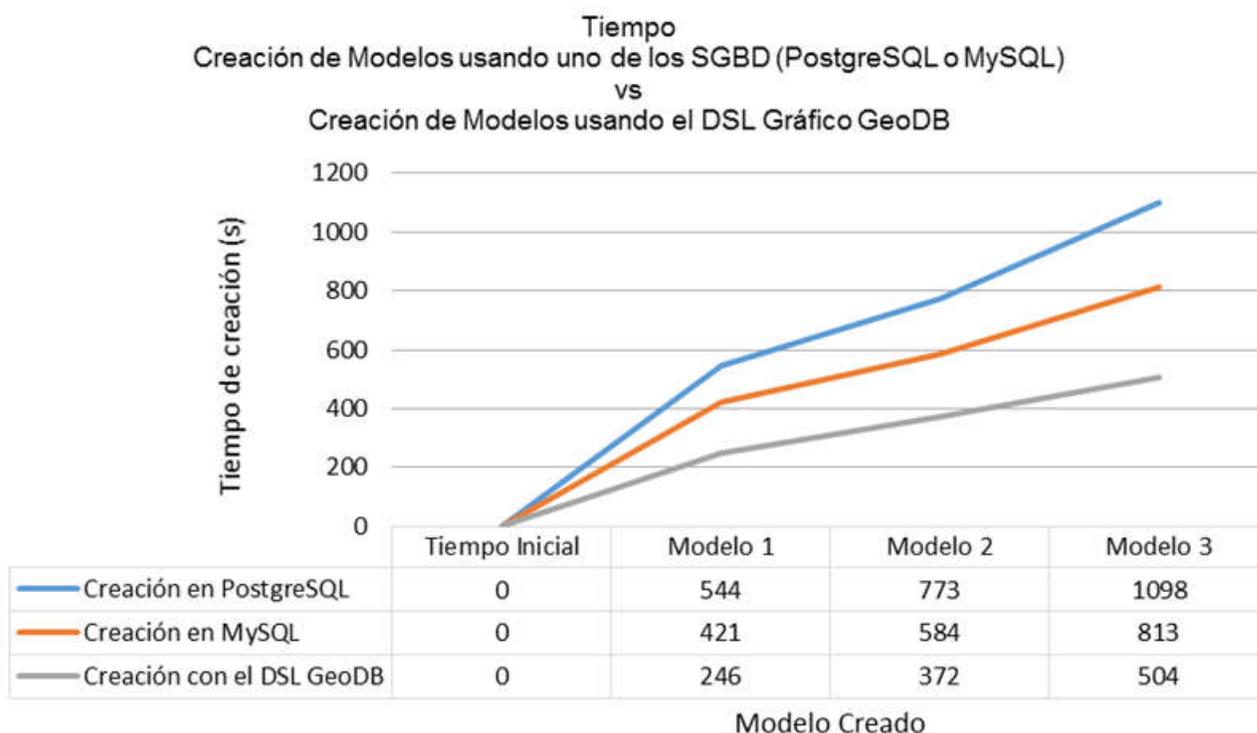


Figura 21: Comparación de tiempos en la Creación de Modelos de BD en los dos SGBD contra la herramienta DSL GeoDB creada.

Fuente: Elaboración propia

Usabilidad (Esfuerzo)  
 Creación de Modelos usando uno de los SGBD (PostgreSQL o MySQL)  
 vs  
 Creación de Modelos usando el DSL Gráfico GeoDB

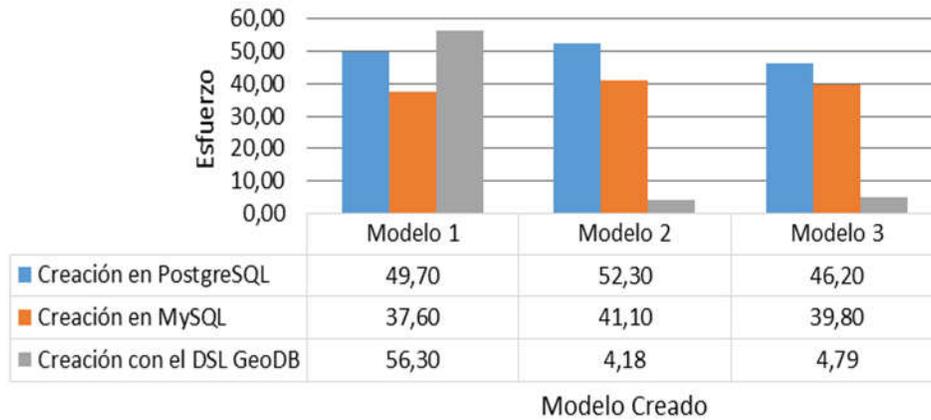


Figura 22: Comparación de esfuerzo en la Creación de Modelos de BD en los dos SGBD contra la herramienta DSL GeoDB creada.

Fuente: Elaboración propia

Las figuras 23 y 24, siguen corroborando la misma situación en cuanto a tiempo total y esfuerzo total que se emplea en la creación de los tres modelos de bases de datos, en cuanto a que es menor tanto el tiempo como el esfuerzo con el DSL gráfico GeoDB.

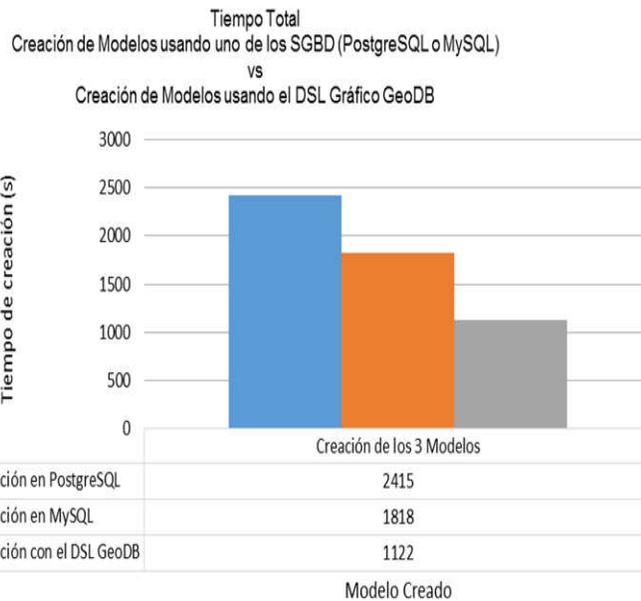


Figura 23: Comparación de Tiempo Total (seg) en la Creación de Modelos de BD en los dos SGBD contra la herramienta DSL GeoDB creada.

Fuente: Elaboración propia

Total de Usabilidad (Esfuerzo)  
 Creación de Modelos usando uno de los SGBD (PostgreSQL o MySQL)  
 vs  
 Creación de Modelos usando el DSL Gráfico GeoDB

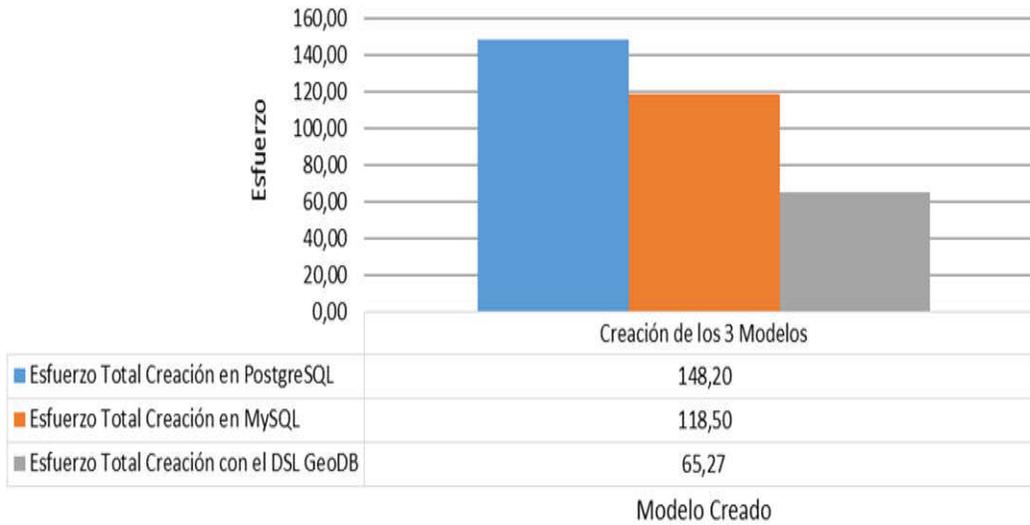


Figura 24: Comparación de Esfuerzo Total en la Creación de Modelos de BD en los dos SGBD contra la herramienta DSL GeoDB creada.

Fuente: Elaboración propia

## 8. CONCLUSIONES

Este trabajo presenta una herramienta de dominio específico para el modelado de base de datos espaciales sencillo, que soporta el desarrollo dirigido por modelos de editores GFM, con una curva de aprendizaje alta gracias a la integración de notación gráfica, mucho más intuitiva, cognitiva y eficiente. La herramienta permite la generación de modelos espaciales a los usuarios finales de una manera más transparente, menos compleja y semiautomática. Desvinculando al usuario con el manejo de lenguaje de dominio textual como el SQL y los comandos específicos de cada una de las extensiones espaciales.

Esta es la primera herramienta de dominio específico que permite modelar Bases de Datos Espaciales para ser implementados en las plataformas PostgreSQL-PostGIS y MySQL-MySQL Spatial, aportando un avance significativo en el área de la geomática y los Sistemas de Información Geográfica.

Es importante destacar que el desarrollo de este trabajo muestra nuevamente el aporte que pueden realizar los lenguajes de dominio específico, en especial los que tienen un componente gráfico, en el desarrollo de otras disciplinas. Además, gracias al uso de MDAs y DSLs, el usuario deja a un lado la preocupación por los detalles y por las diferencias entre un lenguaje y otro, ya que, por medio de MDA, es posible transformar un proyecto de un lenguaje específico a otro.

Las pruebas realizadas demuestran que al realizar la creación de modelos de base de datos con el DSL gráfico GeoDB, se disminuye el tiempo y el esfuerzo empleado comparado con los procesos realizados en cualquiera de los SGBD analizados (PostgreSQL y MySQL). A su vez para el DSL se tuvo en cuenta el metamodelo planteado para esta situación.

## 9. RECOMENDACIONES

La herramienta de dominio específico para el modelado de Base de Datos Espaciales expuesta en el presente artículo tiene la posibilidad de ampliar el número de aplicaciones específicas de Bases de Datos Espaciales, tal como Oracle-Oracle Spatial o ArcSDE entre otras. De igual manera se puede pensar en la adición de nuevos objetos como índices espaciales y rutinas o disparadores que ayudarían al usuario en el modelamiento de la Base de Datos Espacial.

Es necesario mejorar la evaluación de la herramienta y medir la robustez y usabilidad del DSL gráfico, así como la posibilidad de realizar el DSL en otras herramientas gráficas y comparar el rendimiento de estas y sus modificaciones.

### TRABAJO FUTURO

- Se podría pensar en ampliar y/o mejorar el *DSL gráfico GeoDB* en SIRIUS y su respectivo parcer *parcerGeoDB*, adicionándole o considerando los siguientes elementos:
  - Definición de dominios para los campos o atributos.
  - Definición de otros sistemas de referencia que puedan ser personalizados.
  - Considerar otros SGBD adicionales a los definidos en este trabajo.
  - Considerar el agregar o implementar una Interfaz Gráfica al parcer para definir los archivos de entrada y salida.
  - Considerar que el DSL gráfico y su parcer tuviese la posibilidad de una conexión al SGBD requerida para ejecutar directamente en el mismo Sistema Gestor la implementación del modelo de BDE junto con sus componentes, buscando una mayor usabilidad para el usuario final.

## 10. REFERENCIAS BIBLIOGRAFICAS

- [1] V. Olaya, *Sistemas de Información Geográfica*. 2011.
- [2] S. Aronoff, *Geographic Information System: a management perspective*. WDL Publications, Ottawa, Canada, 1989.
- [3] M. F. Goodchild and R. P. Haining, "GIS and Spatial Data Analysis: Converging Perspectives," no. 1, 2004.
- [4] P. A. Longley, M. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Systems and Science*. John Wiley & Sons, 2010.
- [5] H. Deeken, T. Wiemann, and J. Hertzberg, "Grounding semantic maps in spatial databases," *Rob. Auton. Syst.*, vol. 105, pp. 146–165, 2018.
- [6] S. Li, S. Dragicevic, F. A. Castro, M. Sester, S. Winter, A. Coltekin, C. Pettit, B. Jiang, J. Haworth, A. Stein, and T. Cheng, "Geospatial big data handling theory and methods: A review and research challenges," *ISPRS J. Photogramm. Remote Sens.*, vol. 115, pp. 119–133, 2016.
- [7] S. Mintchev, "User-defined rules made simple with functional programming," *Lect. Notes Bus. Inf. Process.*, vol. 176 LNBIP, pp. 229–240, 2014.
- [8] D. H. Ghosh, "DSL for the uninitiated," *Commun. ACM*, vol. 54, no. 7, pp. 44–50, 2011.
- [9] Pascual Espada Jordán, "DISEÑO DE OBJETOS VIRTUALES COLABORATIVOS ORIENTADOS A SERVICIOS EN EL MARCO DE INTERNET DE LAS COSAS," Universidad de Oviedo, 2012.
- [10] C. Gonz, "Tesis Doctoral – Mención Internacional," 2016.
- [11] T. Kosar, S. Bohra, and M. Mernik, "Domain-Specific Languages: A Systematic Mapping Study," *Inf. Softw. Technol.*, vol. 71, pp. 77–91, 2016.
- [12] C. Montenegro Marín, J. Cueva Lovelle,  Sanjuán Martínez, and P. Garcí, Gaonaa, "Desarrollo de un lenguaje de dominio específico para sistemas de gestión de aprendizaje y su herramienta de implementación 'KiwiDSM' mediante ingeniería dirigida por modelos," *Ingeniería*, vol. 15, no. 2, pp. 67–81, 2010.
- [13] V. Viyović, M. Maksimović, and B. Perišić, "Sirius: A rapid development of DSM

- graphical editor,” in *INES 2014 - IEEE 18th International Conference on Intelligent Engineering Systems, Proceedings*, 2014, pp. 233–238.
- [14] N. A., *Sistemas de Información Geográfica*. 2001.
- [15] P. G., M. . Leipnik, and Y. J., *A GLOSSARY OF GIS TERMINOLOGY*. North Dakota State University, 1992.
- [16] J. Bosque Sendra, *Sistemas de Información Geográfica*. Rialp, Madrid, 1992.
- [17] M. F. Goodchild, “Geographic Information Systems and disaggregate transportation modeling Geographical Systems,” 1998.
- [18] J. Bosque Sendra, *Sistemas de Información Geográfica*. Rialp, Madrid, 2000.
- [19] P. A. Burrough, *Principles of Geographical Information Systems. Spatial Information System and Geostatistics*. Oxford University Press, New York, U.S.A, 2000.
- [20] P. A. Burrough, *Principles of Geographical Information Systems for Land Resources Assessment*. Clarendon, Oxford, 1986.
- [21] Y. Gao, B. Zheng, G. Chen, and Q. Li, “Optimal-Location-Selection Query Processing in Spatial Databases,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 8, pp. 1162–1177, 2009.
- [22] N. Lantada Zarzosa and N. A. M. Amparo, *Sistemas de Información Geográfica, Prácticas con ArcView*. Alfaomega, 2004.
- [23] C. Ordoñez and R. Martinez, *Sistemas de Información Geográfica*. RA-MA, 2003.
- [24] O. M. GROUP, “MOF 2.0/XMI Mapping, Version 2.1.1,” 2007.
- [25] E. BUDINSKY, F., STEINBERG, D., PATERNOSTRO, M. & MERKS, *EMF: Eclipse Modeling Framework*. Addison-Wesley, 2009.
- [26] K. TOLVANEN, J. P., MARTTIIN, P. & SMOLANDER, “An integrated model for information systems modeling,” in *Proceeding of the Twenty-Sixth Hawaii International Conference*, 1993, pp. 470–479.
- [27] OMG, “OMG Unified Modeling Language Specification, Version 1.3.,” 1999.
- [28] C. Montenegro Marín, *Ingeniería dirigida por modelos (MDA) y casos prácticos*. 2017.
- [29] Y. Savchenko and V. Stepashko, “Metamodeling as a Way to Universalization of Inductive Modeling Tools,” in *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, 2018, vol. 1, pp. 444–447.
- [30] I. Reinhartz-Berger, K. Figl, and Ø. Haugen, “Investigating styles in variability modeling: Hierarchical vs. constrained styles,” *Inf. Softw. Technol.*, vol. 87, pp. 81–

102, 2017.

- [31] Y.-T. Liao, J. Zhou, C.-H. Lu, S.-C. Chen, C.-H. Hsu, W. Chen, M.-F. Jiang, and Y.-C. Chung, "Data adapter for querying and transformation between SQL and NoSQL database," *Futur. Gener. Comput. Syst.*, vol. 65, pp. 111–121, 2016.
- [32] H. Kilov, P. F. Linington, J. R. Romero, A. Tanaka, and A. Vallecillo, "The Reference Model of Open Distributed Processing: Foundations, experience and applications," *Comput. Stand. Interfaces*, vol. 35, no. 3, pp. 247–256, 2013.
- [33] R. Eito-Brun, "Chapter 1 - XML: The Basis of the Language," in *XML-based Content Management*, R. Eito-Brun, Ed. Chandos Publishing, 2018, pp. 1–30.
- [34] <https://www.postgresql.org/>
- [35] <https://blog.infranetworking.com/infografia-mysql-vs-postgresql/>
- [36] <https://www.2ndquadrant.com/es/postgresql/postgresql-vs-mysql/>