



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**VICERRECTORADO DE INVESTIGACIÓN
INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA**

MAESTRÍA EN INGENIERÍA DE SOFTWARE

IV PROMOCIÓN

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE
MAGISTER EN INGENIERÍA DE SOFTWARE**

**TEMA: “CONSTRUCCIÓN DE UN FRAMEWORK PARA EL
DISEÑO DE UNA BASE DE DATOS ESPACIAL BASADOS EN
SHAPEFILES, APLICADO AL SISTEMA DE
INFRAESTRUCTURA VÍAL DE LA PROVINCIA DE
CHIMBORAZO”**

AUTOR: ING. JAIME EDUARDO ASQUI POMA

DIRECTOR: ING. LUCAS GARCÉS GUAYTA MSc.

LATACUNGA

2015



UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

MAESTRÍA EN INGENIERÍA DE SOFTWARE

CERTIFICACIÓN

Certifico que el trabajo de titulación, “CONSTRUCCIÓN DE UN FRAMEWORK PARA EL DISEÑO DE UNA BASE DE DATOS ESPACIAL BASADOS EN SHAPEFILES, APLICADO AL SISTEMA DE INFRAESTRUCTURA VÍAL DE LA PROVINCIA DE CHIMBORAZO.”, realizado por el Ing. JAIME EDUARDO ASQUI POMA, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo que cumple con los requisitos teóricos científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor JAIME EDUARDO ASQUI POMA para que lo sustente públicamente.

Latacunga, 19 de noviembre de 2015

Ing. Lucas Garcés Guayta MSc.
Director



UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

MAESTRÍA EN INGENIERÍA DE SOFTWARE

AUTORÍA DE RESPONSABILIDAD

Yo, JAIME EDUARDO ASQUI POMA, con cédula de identidad N°. 0602920621, declaro que este trabajo de titulación “CONSTRUCCIÓN DE UN FRAMEWORK PARA EL DISEÑO DE UNA BASE DE DATOS ESPACIAL BASADOS EN SHAPEFILES, APLICADO AL SISTEMA DE INFRAESTRUCTURA VÍAL DE LA PROVINCIA DE CHIMBORAZO.” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 19 de noviembre de 2015

Ing. Jaime Eduardo Asqui Poma

C.C.: 0602920621



UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

MAESTRÍA EN INGENIERÍA DE SOFTWARE

AUTORIZACIÓN

Yo, JAIME EDUARDO ASQUI POMA, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “CONSTRUCCIÓN DE UN FRAMEWORK PARA EL DISEÑO DE UNA BASE DE DATOS ESPACIAL BASADOS EN SHAPEFILES, APLICADO AL SISTEMA DE INFRAESTRUCTURA VÍAL DE LA PROVINCIA DE CHIMBORAZO,”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, 19 de noviembre de 2015

Ing. Jaime Eduardo Asqui Poma

C.C 0602920621

DEDICATORIA

Dedico el presente trabajo a todas aquellas personas que de una manera u otra ayudaron con el ánimo necesario para forjar en mí, el deseo de superación y aprendizaje continuo. Sobre todo a Jehová mi Dios que me ha proporcionado todo lo que tengo y soy y en particular a mi familia: mi esposa Jessy, a mis bebés gemelos, a mis padres Jaime y Olga ejemplo de trabajo y abnegación familiar, a mis hermanas Carolina y Abigaíl que me han regalado el mejor obsequio: mis sobrinos Eydan y Eymi; a mis cuñados Medardo y Leodan, a Jenny y Ariel mis primos hermanos.

Jaime Eduardo

AGRADECIMIENTO

A la UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE, esta gran Institución que nos brindó el saber y que ha sido la sede del conocimiento adquirido. De forma especial al Ing. Lucas Garcés, director de tesis, quien ha sabido guiarme con sus sabios conocimientos y paciencia hacia la culminación del presente trabajo de Investigación.

Jaime Eduardo

ÍNDICE DE CONTENIDOS

PORTADA	i
CERTIFICACIÓN.....	ii
AUTORÍA DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA.....	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDOS	vii
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS	xiii
RESUMEN	xv
ABSTRACT	xvi

CAPÍTULO I

DEFINICIÓN DEL PROBLEMA	1
1.1 Introducción.	1
1.2 Planteamiento del Problema.	1
1.3 Formulación del Problema.....	1
1.4 Objetivo General.	4

1.5	Objetivos Específicos.....	4	
1.6	Justificación e Importancia.....	4	
1.7	Planteamiento de la Hipótesis.....	6	
1.8	Hipótesis General.....	6	
1.9	Variables de la Investigación.....	6	
1.10	Conclusión del Capítulo.....	6	
CAPÍTULO II			
MARCO TEÓRICO			7
2.1	Introducción.....	7	
2.2	Antecedentes Históricos	7	
2.2.1	Primera Etapa Cronológica (Hasta 1980)	8	
2.2.2	Segunda Etapa Cronológica (1980 – 1995)	8	
2.2.3	Tercera Etapa Cronológica (1995 – Actualidad).....	10	
2.3	Antecedentes conceptuales y referenciales.....	12	
2.3.1	Caracterización Gnoseológica de los Sistema de Información Geográfica .	12	
2.3.2	Componentes de un SIG	13	
2.3.3	Sistema de Coordenadas	18	
2.3.4	Caracterización Tecnológica de los métodos de almacenamiento de datos espaciales.....	18	

2.3.5 Bases de datos Espaciales	24
2.3.6 El modelo relacional	31
2.3.7 Normalización	38
2.4 Antecedentes Contextuales	40
2.5 Conclusión del capítulo	44

CAPÍTULO III

DESARROLLO DEL FRAMEWORK	45
3.1. Introducción.	45
3.2. Representación Gráfica del Modelo	45
3.3. Análisis Previo	47
3.4. Desarrollo Modelo E-A-R Espacial.....	47
3.4.1. Identificar Entidades.....	49
3.4.2. Identificar Relaciones	50
3.4.3. Identificar Atributos de las entidades	51
3.4.4. Derivación de tablas	52
3.5. Velocidad de Respuesta	58
3.5.1 Índices.....	58
3.5.2 Selección de claves.....	63
3.5.3 División de tablas	63

3.5.4 Agrupación de tablas (Desnormalización).....	64
3.6. Definición del tamaño de la Base de Datos	64
3.7. Conclusión del capítulo	65

CAPÍTULO IV

IMPLEMENTACIÓN DEL FRAMEWORK.....66

4.1. Introducción.	66
4.2. Herramientas de Desarrollo	66
4.2.1. Sistema Operativo.....	66
4.2.2. Base de Datos Espacial	67
4.2.3. Cliente SIG.....	67
4.2.4. Servidor de Mapas	68
4.2.5. Servidor Web.....	68
4.3. Implementación del framework	68
4.3.1. Análisis Previo	69
4.3.2. Data Cleansing	71
4.3.3. Identificar Entidades.....	72
4.3.4. Identificar Relaciones	73
4.3.5. Identificar Atributos.....	75
4.3.6. Derivación de Tablas	77

4.4. Desarrollo del Aplicativo.....	80
4.4.1. Explicación.....	80
4.4.2. Conexiones.....	80
4.5 VALIDACION DEL FRAMEWORK	82
4.5.1. Comparativa referencial.....	82
4.5.2. Investigación de campo.....	86
4.5.3. Indicadores.....	88
4.5.4. Tabulación de resultados.....	89
4.5.5. Discusión de los resultados.....	92
4.6. Conclusión del Capítulo.....	92

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES.....	93
5.1. CONCLUSIONES.....	93
5.2. RECOMENDACIONES.....	94
BIBLIOGRAFÍA.....	95
ANEXOS.....	98

ÍNDICE DE TABLAS

Tabla 1 Grupo de Trabajo.....	2
Tabla 2 Variables	6
Tabla 3 Caracterización de las Etapas Cronológicas.....	12
Tabla 4 Comparativa shape vs B.D.....	84
Tabla 5 Consultas	85
Tabla 6 Resultados ítems almacenamiento	89
Tabla 7 Resultados ítems respuesta en búsquedas	90
Tabla 8 Resultados ítems generación informes.....	90
Tabla 9 Resultados ítems acceso a la información.....	91

ÍNDICE DE FIGURAS

Figura 1 Componentes del Objeto Espacial	16
Figura 2 Elementos que conforman un SIG.....	17
Figura 3 Intercambio de datos entre sistemas.....	20
Figura 4 Topología estrella para implementación de interfaces	21
Figura 5 Sistema de gestión de Bases de Datos Espaciales	25
Figura 6 Los SGBD crean información a partir de los datos	26
Figura 7 Representación de los Modelos	46
Figura 8 Secuencia de pasos AER	48
Figura 9 Secuencia de pasos AER	50
Figura 10 Entidades Espaciales	50
Figura 11 Relaciones Espaciales	52
Figura 12 Estructura Archivo Shape	69
Figura 13 Entidades Alfanumérica.....	72
Figura 14 Entidades Espaciales	73
Figura 15 Relaciones Identificadas.....	74
Figura 16 Atributos Identificados	75
Figura 17 Derivación de Tablas.....	78
Figura 18 Entidad cantón derivada.....	79
Figura 19 Entidad Parroquia derivada.....	79

Figura 20 Entidad capa_rodadura derivada.....	79
Figura 21 Entidad actualizada derivada	79
Figura 22 Entidad obra_arte derivada.....	80
Figura 23 Entidad orden derivada	80
Figura 24 Conexión Postgis	81
Figura 25 Líneas de Código	81
Figura 26 Tamaño shape.....	83
Figura 27 Tamaño B.D. Postgis	84
Figura 28 Contador de Visitas.....	86
Figura 29 Resultados Porcentuales ítems almacenamiento	89
Figura 30 Resultados Porcentuales ítems búsquedas.....	90
Figura 31 Resultados porcentuales ítems generación de informes.....	91
Figura 32 Resultados porcentuales ítems acceso a la información.....	91

RESUMEN

El fin del presente trabajo de investigación es desarrollar e implementar un framework para transformar un sistema no relacional de shapes vectoriales a un sistema relacional con base de datos espacial centralizado, aplicado en el Sistema de Infraestructura Vial de la Provincia de Chimborazo. En el capítulo uno trata de presentar un resumen de la problemática actual que se presenta en un sistema no relacional de shapefiles con los consecuentes inconvenientes así como el desperdicio de recursos y por otro lado las ventajas de utilizar un framework para transformarlo a un sistema relacional con una base de datos espacial. En el segundo capítulo se desarrolla el Marco Teórico, definiendo los antecedentes históricos con los cuales se determina la evolución que ha sufrido los métodos y técnicas de almacenamiento geoespacial. Se desarrolla los antecedentes conceptuales y se establecen los antecedentes contextuales para justificar la existencia del problema planteado en la presente investigación. El capítulo tres establece el modelo simplificado de la secuencia de pasos a seguir para el desarrollo del framework, trasladar desde un archivo shape a una base de datos espacial, iniciando desde el análisis previo, siguiendo el modelo EAR, posteriormente el análisis de la velocidad de respuesta así como el tamaño de la base de datos. En el capítulo cuatro se implementa el framework; siguiendo los pasos del capítulo precedente, al final del capítulo procedemos a validar el framework desarrollado y con ello a comprobar la hipótesis basada en una comparativa referencial e investigación de campo con el fin de medir cualitativa y cuantitativamente los indicadores. En el capítulo cinco se exponen las conclusiones y recomendaciones del trabajo de investigación.

PALABRAS CLAVES:

- **BASE DE DATOS ESPACIAL**
- **SISTEMA DE INFORMACIÓN GEOGRÁFICA**
- **MODELO EAR ESPACIAL**
- **INGENIERÍA DE SOFTWARE**

ABSTRACT

The purpose of this research is to develop and implement a framework to transform a non-relational vector system to a relational database system with centralized spatial data, the system applied in Road Infrastructure of the Province of Chimborazo shapes. In chapter one is a summary of the current problem that occurs in a non-relational system shapesfiles with consequent inconvenience and wasted resources on the other hand the benefits of using a framework to transform it to a relational system spatial data base. The second chapter develops the theoretical framework, defining the historical background to the evolution which has suffered the methods and techniques of geospatial storage is determined. It develops the conceptual background and contextual background are set to justify the existence of the problem raised in the present investigation. Chapter three provides the simplified model of the sequence of steps for the development of the framework, moving from a shape file to a spatial database, starting from the previous analysis, following the model EAR later analysis speed response and the size of the database. In chapter four the framework is implemented; following in the footsteps of the previous chapter, at the end of Chapter proceed to validate the developed framework and thereby check the hypothesis based on a comparative benchmark and field research in order to qualitatively and quantitatively measure the indicators. In chapter five the conclusions and recommendations of the research are presented.

KEYWORDS:

- **SPATIAL DATABASE**
- **SYSTEM INFORMATION GEOGRAPHIC**
- **EAR MODEL SPACE**
- **SOFTWARE ENGINEERING**

CAPÍTULO I

DEFINICIÓN DEL PROBLEMA

1.1 Introducción.

Este capítulo trata de presentar un resumen de la problemática actual que se presenta en un sistema no relacional de shapfiles con los consecuentes inconvenientes así como el desperdicio de recursos y por otro lado las ventajas de utilizar un framework para transformarlo a un sistema relacional con una base de datos espacial centralizado.

Adicionalmente se presentan justificativos del porqué es necesario el uso de Tecnologías que mejoren la forma de administrar la información levantada en el aspecto geográfico, con el fin de optimizar recursos.

1.2 Planteamiento del Problema.

El Gobierno Provincial de Chimborazo posee un inventario de la Infraestructura Vial en el ámbito de sus competencias; sin embargo, la información se encuentra almacenada en shapfiles en una estructura plana, debido al desconocimiento en el ámbito del diseño de una base de datos por parte de los técnicos y porque en realidad en su área no es imprescindible dicho conocimiento. Con esta forma de almacenar la información, que no es más que datos guardados en tablas en forma no relacional se han presentado una diversidad de problemas entre ellos un excesivo espacio de uso en disco, no se garantiza integridad referencial en los datos, se produce una deficiencia al momento de buscar y encontrar la información y finalmente no se encuentra centralizado el almacenamiento.

1.3 Formulación del Problema

La vialidad al ser competencia exclusiva de los GAD'S provinciales requieren datos idóneos y actualizados para una acertada y eficiente planificación, es por ello que el GAD de Chimborazo se ha visto en la necesidad de actualizar una completa base de datos que permita visualizar e identificar los requerimientos actuales, programar intervenciones, implementar programas de mantenimiento vial,

herramienta básica para la planificación integral en el ámbito de la vialidad y los servicios asociados a esta competencia.

La información geográfica en los últimos años ha tomado gran importancia en la planificación, la asignación correcta de los recursos y manejo adecuado de las entidades, viéndose el GADCH Provincial limitado en su accionar al no disponer de herramientas adecuadas para la toma de decisiones estratégicas.

La información que se obtenga del proceso denominado Inventario De La Infraestructura Vial, será un insumo para la elaboración del Plan Vial Participativo De Chimborazo, el mismo que permitirá la generación de programas y proyectos encaminados a dirigir la competencia de la vialidad en la provincia para el corto, mediano y largo plazo, como competencia exclusiva, la movilidad y su potencial aporte al desarrollo sustentable de la provincia y del país.

En tal virtud la Coordinación de Planificación del Gobierno Autónomo Descentralizado de la Provincia de Chimborazo, dentro de sus objetivos y empeñado en levantar tanto la línea base como el inventario de la Infraestructura Vial de la Provincia asignó el material técnico y humano para dicha labor, proceso que se efectuó durante los años 2012 y 2013.

La distribución del personal se lo hizo de acuerdo al siguiente cuadro:

Tabla 1

Grupo de Trabajo

TECNICO	RESPONSABILIDAD
Jefe de Equipo	Georeferenciación, Inventario, generación de datos, definición de red vial y codificación, Informe final
Ingeniero Civil	Georeferenciación, Inventario, generación de datos, informe final
Ingeniero Geógrafo	Generación de Mapas temáticos, codificación y responsable del sistema de información geográfica, responsable de la geodatabase.
Ingeniero Ambiental	Georeferenciación e Inventario.
Asistente de campo	Georeferenciación e Inventario.

Para el levantamiento de la Información en sitio, la Coordinación asignó el equipo necesario y suficiente, incluyendo toda la logística tanto en transporte como en alimentación. La georeferenciación y toma de información se realizó en formato shapes, y subido al ArcGis con licencia 10.1. lo que provocó los siguientes problemas:

Uso excesivo de disco.- La información almacenada en disco es más de lo que en forma óptima debería hacerlo, debido a que los registros o los campos se repiten innecesariamente, en los shapefile al ser una estructura lineal los datos deben ir en detalle, al ser un diseño no relacional, el espacio que ocupa es mayor del necesario, provocando exceso de datos, datos basura, duplicación de registros, entre otros inconvenientes propios del diseño no relacional.

No se garantiza la integridad referencial.- Así al eliminar un registro no se puede estar seguro de que dicho dato no afecte en forma sustancial al resto de la información, a la vez que de ninguna manera se garantiza que los datos puedan ser duplicados o corruptos.

Deficiencia en búsquedas.- No se pueden hacer búsquedas personalizadas rápidas a los datos para presentarlos a nivel gerencial. El tiempo para procesar los requerimientos de la alta gerencia es demasiado elevado, cuando esta información se la considera crítica y debería idealmente ser en tiempo real, y la persona que realice estos reportes no necesariamente debería ser un experto en el tema tratado.

Problemas en actualización.- Los shapefiles están almacenados en disco en cada computador de los técnicos que requieren de dicha información, por lo que los datos están disgregados y no se conoce exactamente cuál es el último archivo modificado y válido en su estructura. Debido a este inconveniente se tiene que realizar un levantamiento de la línea base por lo general cada dos años.

¿Cómo optimizar el almacenamiento en disco en el Sistema del Inventario Vial de la Provincia de Chimborazo?

1.4 Objetivo General.

Construir un framework para el diseño de una base de datos espacial basados en shapefiles que optimice el almacenamiento en disco, aplicado al Sistema de Infraestructura Vial de la Provincia de Chimborazo.

1.5 Objetivos Específicos.

- a. Construir el Marco Teórico que permita el análisis de la información de los Sistemas de Información Geográfica y los Métodos y técnicas de Almacenamiento de datos geoespaciales.
- b. Desarrollar el Framework para el almacenamiento de los shapefiles en una base de datos espacial en el Sistema de Inventario Vial de la Provincia de Chimborazo.
- c. Implementar el Framework para la migración de los shapefiles a la base de datos espacial, para centralizar la información.
- d. Validar el Framework mediante la creación de la geodatabase, que permita la manipulación, búsqueda, y acceso a la información almacenada en los discos.

1.6 Justificación e Importancia.

Con la vigencia de la Constitución 2008, el Plan de Gobierno “Segunda Minga Por Chimborazo” 2009-2014, fundamentó sus propuestas de desarrollo en las competencias exclusivas atribuidas a los Gobiernos Autónomos Descentralizados en el Art. 263 de la Constitución. Puntualmente, en lo referido en el literal 2 del mencionado artículo que señala como competencia del GAD Provincial:

“Planificar, construir y mantener el sistema vial de ámbito provincial, que no incluya las zonas urbanas”. (DEL TRABAJO, 2008)

El trabajo de investigación a la vez está alineado con el Plan del Buen Vivir del Gobierno Nacional 2013-2017 que entre las principales líneas de acción para la transformación de la matriz productiva expresa: “Mejorar y/o construir infraestructura de articulación vial, fluvial y aérea, a fin de facilitar la movilidad humana, el transporte y comercialización de productos, e integración de áreas productivas y mercados.” (Ecuador, Ecuador, & Secretaría Nacional de Planificación

y Desarrollo, 2013)

Está justificado en la Ley del COOTAD Código Orgánico de Organización Territorial, Autonomía y Descentralización, Capítulo IV del Ejercicio de las Competencias Constitucionales en el Artículo 129 del Ejercicio de la competencia de la vialidad en su apartado pertinente expresa: “Al gobierno autónomo descentralizado provincial le corresponde las facultades de planificar, construir y mantener el sistema vial de ámbito provincial, que no incluya las zonas urbanas.” (CHANCAY ZAMBRANO & ZAMBRANO ARIAS, 2015)

El objetivo atribuido a la competencia de vialidad en la Segunda Minga se centró en “disponer de un Sistema de Conectividad Provincial en óptimas condiciones para favorecer las actividades productivas, comerciales y sociales de la población, cuidando siempre del medio ambiente. Fomentando además la generación de empleo mediante la participación de microempresas comunitarias de operación y mantenimiento, con el fin de mejorar la calidad de vida de la población más necesitada”

Los Programas contemplados fueron:

1. Apertura de nuevas Vías Intercomunitarias que permitan la comunicación de Sectores Rurales actualmente aislados.
2. Mejoramiento de Caminos Vecinales Comunitarios cuyo afirmado se encuentra en estado natural.
3. Reconstrucción de Caminos vecinales Comunitarios cuyo afirmado se está perdiendo por falta de Mantenimiento Vial Rutinario.
4. Creación de Microempresas de Mantenimiento Vial Rutinario.

El marco legal señalado justifica la elaboración del Inventario Vial Provincial, como parte de las acciones de planificación que permitan disponer de una Base de Datos Georeferenciada, homogenizada e integrada en un SIG, para proporcionar información en detalle que permita realizar el análisis de los posibles modelos de gestión y la posterior generación del Plan de Gestión Vial Provincial de Chimborazo con la definición de políticas, programas y la priorización de proyectos viales.

1.7 Planteamiento de la Hipótesis

1.8 Hipótesis General.

Con el uso del framework construido para el diseño de una base de datos espacial entonces se optimiza el almacenamiento en disco en el Sistema de Infraestructura Vial de la Provincia de Chimborazo.

1.9 Variables de la Investigación.

Tabla 2

Variables

Tipo	Variable	Indicadores
Variable Independiente	Uso del framework	
Variable Dependiente	Optimizar el almacenamiento en disco	<ul style="list-style-type: none"> • Espacio de almacenamiento en disco. • Tiempo de respuesta en búsquedas. • Tiempo de generación de informes. • Acceso masivo a la información

1.10 Conclusión del Capítulo.

El trabajo de investigación se desarrollará en pasos notoriamente definidos, al iniciar con la revisión bibliográfica para la determinación del marco teórico y marco contextual, posteriormente vendrá el estudio y la aplicación de técnicas de data cleansing con el objetivo de poseer información válida para el ingreso al Sistema.

La parte medular de la investigación se refiere al desarrollo del framework para consolidar la base de datos espacial a partir de los shapefiles existentes dicho marco de trabajo estará en lo posible alineado con la realidad local; finalmente, la generación del diseño físico en el motor de Base de Datos espacial para su verificación y validación.

CAPÍTULO II

MARCO TEÓRICO

2.1 Introducción.

En el presente capítulo se desarrolla el Marco Teórico, definiendo los antecedentes históricos con los cuales se determina la evolución que ha sufrido los métodos y técnicas de almacenamiento geoespacial en los Sistemas de Información Geográfica.

Hecho el análisis histórico de manera cronológica, se desarrolla los antecedentes conceptuales, caracterizando los métodos y técnicas de almacenamiento geoespacial y los Sistemas de Información Geográfica y finalmente se establecen los antecedentes contextuales que permiten justificar la existencia del problema planteado en la presente investigación.

2.2 Antecedentes Históricos

Evolución de los métodos de almacenamiento geoespacial en los Sistemas de Información Geográfica.

Es tan importante en la actualidad los datos espaciales almacenados en un Sistema de Información Geográfico, así como los atributos de las entidades y la discusión de las distintas formas de almacenar los datos en una geodatabase, las diversas formas de acceder a las estructuras de los motores de bases de datos espaciales que manejan las diferentes herramientas; todo ello con el fin de que la Información llegue a ser masiva y brinde la ayuda necesaria para la toma de decisiones.

La evolución cronológica de los métodos y técnicas de almacenamiento en los Sistemas de Información Geográfica inició con mapas análogos, posteriormente los archivos shape hasta la actualidad el uso de geodatabase para el almacenamiento de los datos espaciales. La evolución cronológica con la descripción lógica se evidencia en las etapas que se muestra seguidamente.

2.2.1 Primera Etapa Cronológica (Hasta 1980)

En esta etapa cronológica se inicia el uso de los mapas, sin poder determinarse exactamente la fecha en la que se creó el primero; esencialmente lo usaban los mercaderes quienes viajaban grandes distancias y veían la necesidad de registrar su recorrido. Se caracteriza por el uso de mapas con **información análoga**. Dichos dibujos o mapas no tenían ningún tipo de relación geográfica.

En la década de 1940, nace el sistema creado por el Cuerpo de Ingenieros del Ejército de los Estados Unidos que se le denominó Sistema de Coordenadas Universal Transversal de Mercator (UTM por el inglés Universal Transverse Mercator). Posteriormente, se inicia con el manejo de archivos asistidos por Computadoras (CAD por el inglés Computer Aided Design). Se logran establecer distancias, escalas, que podían o no ser reales pero definitivamente no se establecen atributos para los mapas.

En la década de los 60 Roger Tomlinson, considerado como el padre del SIG, creó el CGIS (Canada Geographic Information System) para el Canadian Land Inventory. Este primer Sistema de Información Geográfica se consideró un sistema de medición de mapas computarizado. Más tarde, se establece el primer Laboratorio de Gráficos Informatizados y Análisis Espacial en la Universidad de Harvard y en 1969 dos de sus estudiantes forman la empresa ESRI, conocida actualmente por el desarrollo de proyectos SIG, esto se lo puede considerar como el inicio de los Sistemas de Información Geográfica, así como la primera etapa cronológica de la forma de manejar y almacenar información geográfica (Navarro, 2011).

2.2.2 Segunda Etapa Cronológica (1980 – 1995)

En esta etapa aparecen los sistemas de información geográfica (SIG) como tal; en los años 80 es cuando repunta el despliegue de los SIG influenciado especialmente por la disminución notable en el precio de las computadoras pudiendo hacer sostenible la industria del software. Los pioneros en este campo debido a su utilidad fueron empresas forestales y de recursos naturales, dadas las ventajas económicas que significaban tomar buenas decisiones utilizando herramientas SIG en comparación a los altos costos de los SIG, de ahí en adelante el precio del hardware

fue siendo más accesible y a su vez el del software con el poder de los SIG, este crecimiento no ha disminuido hasta la actualidad.

En el año de 1994 se crea el Open Geospatial Consortium (OGC) que se funda con el fin de agrupar a organizaciones de diferente índole pública y privada. En la década de los noventa se generan importantes avances en interfaces de usuario, interoperabilidad, con el fin de facilitar el uso extensivo de los SIG. En esta década se da inicio a la generación de estándares y normas que regulen la interoperabilidad entre sistemas abiertos y propietarios, y con ello conseguir el fácil intercambio de la información geográfica.

En esta generación la empresa ESRI (Environmental Systems Research Institute) desarrolla su producto ArcView GIS, dentro de esta herramienta los archivos se guardaban como .shp o también llamados shapefile, dada la popularidad de los productos ESRI este tipo de archivo llegó a convertirse en el formato para el intercambio de Información Geográfica. El shapefile es un tipo de formato de almacenamiento digital usado para información vectorial, llegando a ser el estándar de facto en el mundo de los GIS. (Comas & Ruiz, 1993).

Los datos en formato shapefiles se manipulan en una forma de multiarchivos pues para obtener la información completa se dividen entre 3 y 8 archivos que son independientes, en donde se almacena la información digitalmente, ubicándola de acuerdo a su categoría en cada uno de estos archivos independientes. Por ejemplo, los elementos que son geométricos se guardan en el .shp; los atributos y la información temática se guarda en un archivo aparte en formato dBase; pero a su vez cada elemento se interrelacionan mediante un código identificador, con lo que se consigue que un elemento conste de diversas partes. Ahora bien, los shape en realidad no almacenan topología de representaciones. Para lograr conservar la representación en un shapefile se debe crear otro archivo .lyr (Layer). Aunque se gana mucho en simplicidad y compatibilidad. Sin embargo, a la hora de compartir o tener centralizado la información no resulta útil este tipo de archivo, de ahí su limitación al momento de guardar y procesar la información con este tipo de archivos y su almacenamiento, pues enviar únicamente el archivo shp no es suficiente para lograr visualizar toda la información cartográfica; se hace necesario que todos los

archivos estén disponibles para mirar correctamente la información, además de que su nombre debe ser el mismo en cada una de las extensiones.

Con toda esta limitante en cuanto al manejo de los diferentes formatos que se tenían que usar para representar la información geográfica dio inicio a nuevas formas de guardar los datos espaciales con el fin de que nos permitan unificar toda la información en un mismo archivo con extensión de base de datos, así es que podemos catalogar la siguiente etapa cronológica que aparece más o menos por la década de los noventa.

2.2.3 Tercera Etapa Cronológica (1995 – Actualidad)

Ante el evidente problema que significa tener diferentes archivos con diferentes extensiones para almacenar información geográfica de un mismo elemento, se empieza a estudiar el uso de las geodatabase, es decir, que en un formato de base de datos nos permita consolidar los diferentes formatos y que a la vez sea interoperable y fácil de manejar la información.

Esta etapa cronológica que inicia en la década de los noventa tiene como característica sobresaliente para solucionar los problemas de la etapa anterior el uso de DBMS (Sistema Gestor de Base de Datos) y siendo más específicos las **Bases de Datos Espacial Relacional**. Las Bases de Datos Espaciales en realidad resuelven algunos problemas que no se podían solucionar con el tipo de archivos shape; pues proveen capacidades propias de un Gestor de Base de Datos. Con un DBMS se puede ya representar, almacenar, acceder y resolver problemas de acceso concurrente, manejar transacciones, centralizar la información, crear backups, recuperación ante fallas, proveer herramientas para el análisis espacial, ofrece mecanismos de visualización que permita a expertos analizar los datos.

Para el manejo de la Base de Datos se usa el Modelo Relacional, el mismo que inicia como una manifestación teórica previa a la aparición de cualquier software de base de datos relacional. En 1970 Edgar F. Codd, entonces científico de IBM, publicó un artículo en el que describió el modelo de base de datos Relacional, para ilustrar cómo pensaba él que debían diseñarse las bases de datos, partiendo de la teoría y lógica ya existentes. Los SGBD ‘relacionales’ de hoy en día son intentos de

implementar la visión teórica de Codd en un software del mundo real (**Sumathi & Esakkirajan, 2007**)

Al usar geodatabase en lugar de archivos, se permite escalar en tamaño y operabilidad pues admiten gran flujo de trabajo, así como la administración de información crítica y permite el versionamiento, replicación y archivo histórico de los datos. El mecanismo que se usa para el almacenamiento en una base de datos espacial es mediante tablas. Estas tablas se almacenan en el disco del servidor y consta de dos grupos de tablas, las denominadas tablas de dataset y las tablas del sistema.

Adicionalmente se usan triggers, funciones, procedimientos almacenados y otras funciones y tipos definidos por el usuario, dependiendo de cada motor de base de datos usado para la generación del geodatabase. La incorporación del uso de Bases de Datos Espaciales radica también en la urgente necesidad de manejar de forma conjunta los datos tanto geográficos como los descriptivos.

Al almacenar la información en un Gestor de este tipo se consigue ligar la información geométrica o espacial a la información alfanumérica; es otras palabras, a los componentes que maneja cualquier gestor de base de datos, ahora le unimos los espaciales. La característica común de estos gestores de bases de datos es también la capacidad relativamente sencilla de administrar grandes volúmenes de información geográfica.

Entre los sistemas Gestores de Bases de Datos Espaciales, se los puede categorizar en dos grandes grupos:

Propietarios.- Entre los Sistemas Propietarios más populares tenemos a Oracle Spatial, ESRI, Intergraph, Galdos Systems, SQL Server.

Open Source.- Entre los oferentes más conocidos en Open Source están PostgreSQL, Mysql, GraphDB.

En resumen podemos definir las tres etapas en la siguiente tabla.

Tabla 3**Caracterización de las Etapas Cronológicas**

	Etapa Cronológica I	Etapa Cronológica II	Etapa Cronológica III
Cronología	Hasta 1980	1980 – 1995	1995 Hasta la actualidad
Características	Papel CAD	Archivos Registro Campos No. de Campos No. de Registros Identificador Único Valores posibles	Relación Tupla Atributos Grado Cardinalidad Clave primaria Dominio
Tipo de Archivos	Papel	Shapefiles Datos Relacionales	Bases de Datos Espaciales Relacionales

2.3 Antecedentes conceptuales y referenciales**2.3.1 Caracterización Gnoseológica de los Sistema de Información Geográfica****Definición de SIG**

En su libro “Sistema de información geográfica. Prácticas con Arc View”, de Nieves Lantada Zarzosa define el SIG como: “Es un conjunto de elementos ordenadamente relacionados entre sí de acuerdo a ciertas reglas. Sus principales componentes son: contenido, equipo básico, equipo lógico, administrador y usuarios” (Zarzosa & Andrés, 2004).

Según Juan Peña Llopis en su libro “Sistemas de información geográfica aplicados a la gestión del territorio” lo define: “Los SIG son una nueva tecnología que permite gestionar y analizar la información espacial, y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato” (Llopis, 2006).

Según Geoffery J. Meaden y James M. Kapetsky, define “Los SIG constituyen una rama de la ciencia o disciplina que ha evolucionado y sigue evolucionando con

tal rapidez, que la definición de lo que es o de lo que hace cambia y se amplía sin cesar, hasta el punto que lo único cierto es que cualquier definición que demos de ella ahora ya no será válida dentro de 5 o 10 años.” (Meaden & Kapetsky, 1992)

De los conceptos investigados podemos concluir que un Sistema de Información Geográfica o también llamado SIG por sus siglas en Inglés (System Information Geographic) se puede definir como aquella técnica o método capaz de tratar la información geográfica para producir información adicional beneficiosa en la toma de decisiones. Para ello, es indispensable el uso tanto de hardware como de software en el desarrollo de herramientas informáticas para facilitar y operar estas tareas.

2.3.2 Componentes de un SIG

Son diferentes las aproximaciones que se pueden dar para definir los componentes que integran un SIG. Mucho depende del punto de vista para resaltar algunos componentes sobre otros, sin embargo, vamos a considerar los más esenciales para su funcionalidad. Por norma general, un distribuidor de software le dará importancia a las funcionalidades de su paquete SIG, mientras que un comerciante de hardware intentará vendernos el equipo más costoso y rápido para el funcionamiento de nuestro SIG. Un Sistema de Información Geográfica como su nombre mismo lo indica es un sistema, es decir, comprende una combinación de diferentes partes, que acoplados entre ellos forman un todo; los componentes principales son el hardware, software, datos espaciales, métodos y personal especializado.

Hardware.- Mientras que en la década de los ochenta y principios de los noventa las aplicaciones de SIG funcionaban en un equipo de trabajo o Workstation y resultaban altamente costosas, el desarrollo rápido en la industria y el crecimiento y mejora en el hardware llevó a que los costos se abaraten, lo que hace que hoy en día los computadores personales sean más que suficientes para administrar un SIG, a la vez que ha crecido drásticamente las características de hardware lo que ha hecho que un SIG no tenga problemas funcionando en una estación de trabajo gráfico potente. A pesar de la indiscutible importancia que tiene el hardware la cuestión del tipo de equipo ha pasado a ser un fundamento de fondo.

Actualmente ya no es necesario recurrir a una solución única ni completa de un solo oferente, sino que podemos hablar de soluciones consolidadas con gran número de componentes sintonizados mutuamente. Más aún, los sistemas actuales distribuidos en el Internet o Intranet, han cambiado la óptica de la Información Geográfica con acceso al mundo entero.

Software.- Para un sistema SIG el software es la parte medular pues gestiona las diferentes funciones y herramientas en la administración, el procesamiento y la visualización de los entes geográficos. La diferenciación de un producto software con el de un software de GIS, se podría detallar en las siguientes características: El precio de este tipo de software es más elevado del de un genérico; la funcionalidad al ser del tipo GIS es especializada y definitivamente, el área especial de uso se clasifica en CAD-GIS, procesamiento de imágenes, SIG vectorial, cartografía, análisis espaciales, simulaciones.

En la actualidad la diferenciación se establece especialmente en el campo de ofrecimientos que establecen los oferentes, correspondiente a las exigencias de los usuarios. En el área de software de estos productos se ofrecen ampliaciones o módulos adicionales al programa o también llamado como extensiones tratando de brindar especificaciones.

Hoy por hoy, la creciente oferta de productos Open Source ha generado grandes expectativas y se brinda especial interés en este tipo de tecnología con el fin del ahorro aspecto muy importante dado el elevado costo de estos productos. Open Source son programas de distribución libre y gratuita, dependiendo del tipo de licencia con el que fue creado, su código fuente está disponible y accesible para desarrolladores y dicho código puede ser modificado dependiendo de las necesidades propias de una empresa o institución.

Entre los software utilizados en Sistemas de Información Geográfico podemos mencionar a modo de ejemplo los siguientes: System form Automated Gescientific Analyses (SAGA), Geographic Resources Analysis Support System (GRASS), gvSIG, uDIG, ArcGIS (Distribuido por ESRI), GeoMeida Professional. En la actualidad se ha desarrollado herramientas Virtual Globes dentro de la geoinformática. Ejemplo de ellos es Google Earth, usado desde el browser de

cualquier navegador, que a su vez compró e integró una voluminosa cantidad de datos de satélites, de imágenes de ciudades. Este desarrollo es uno de los más representativos en cuestión de innovaciones importantes en estos tiempos.

Datos.- Un Sistema de Información Geográfica sin datos, no es un sistema. Una analogía útil que se puede usar para entender es un automóvil, la carrocería se podría representar como el hardware, el software como el motor, pero los datos representan el combustible que hace andar al automóvil; sin combustible no podemos llegar a ningún lugar aunque tuviéramos el automóvil y el motor; un SIG sin los datos espaciales correspondientes no sería útil.

Hay que mencionar por un lado que los SIG trabajan en base de datos específicos que presentan una referencia al espacio y que pueden ser localizados en él. Estos datos son geo referenciados o también llamados datos espaciales. Es importante la relación entre los datos y la información que necesitamos a lo que llamamos sus propiedades topológicas.

Los datos que usamos en los modelos geográficos son de dos tipos basados en el modelo vectorial y en el modelo raster. Si se desea importar los límites en una zona bastante grande se emplean datos tipo raster pues con este modelo de datos se logra tener una información continua. Sin embargo si los límites de las entidades geográficas son muy importantes para la aplicación y los elementos a recoger pueden ser de forma individual entonces se puede usar el modelo vectorial. **(Zarzosa & Andrés, 2004)**

En los dos tipos de estructuras es necesario las relaciones espaciales de los elementos que los conforman, o lo que se le llama como topología. En el área de la cartografía, esta topología es la relación que debe existir del objeto con su entorno, por ejemplo puede ser su conectividad, inclusión.

En el modelo de tipo vectorial se simplifica la representación a la realidad y se lo hace usando puntos, líneas y polígonos, en lo que para cada uno de ellos se guardan las coordenadas que lo conforman. En cambio, en el modelo raster, se usa una matriz de celdas para representar los datos espaciales. La posición espacial del elemento geográfico es única en el gran cuadro de celdas del Grid.

Los objetos espaciales tienen propiedades geométricas y atributos. Un ejemplo para entender este concepto: Para identificar una casa en el espacio necesitamos su valor de coordenadas que representan la información espacial, pero adicionalmente a esta información nos hace falta su color, su tamaño estos datos no son espaciales pero si son atributos de la casa. El posicionamiento de la casa en un sistema de información geográficos pueden ser representados pero a la vez pueden conectarse de forma única con sus atributos de tal modo que además de conocer en donde se encuentra ubicada en el mapa, también podemos conocer exactamente cuál es su color y cuál es su tamaño; al conectar estos dos tipos de datos los alfanuméricos y espaciales generamos un valor añadido a la información de la que disponemos. Este valor añadido a la información mediante conectar datos atributos con datos espaciales es uno de los motivos por los que usamos SIG y es lo que hace singulares este tipo de sistemas. Lo podemos representar en la siguiente figura:

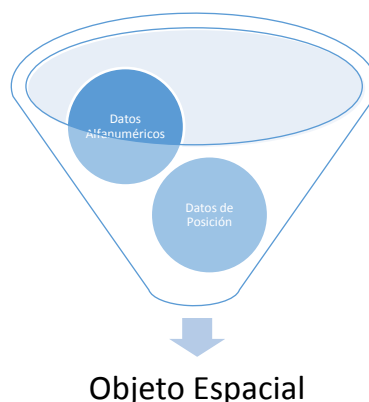


Figura 1 Componentes del Objeto Espacial

Si analizamos en realidad lo que le estamos dando a un objeto es un detalle mayor como información al usuario de los datos. En un sistema de catastro, normalmente se tiene información del terreno, de su propietario, su forma, etc., pero si a esto le añadimos su ubicación representada en un visor de mapas, y la forma del polígono que representa el terreno, la información es enriquecida por inteligencia, para con ello poder tratar y tomar decisiones importantes.

Personal.- Podemos disponer del equipo necesario tanto en hardware, en software y datos pero eso no es suficiente para que un Sistema de Información Geográfico funcione. Es necesaria la coordinación de los componentes; pero más importante aún es el ser humano como parte integrante de este funcionamiento. Los SIG realmente funcionan y son útiles cuando toda la infraestructura tecnológica

funciona pero está garantizada por la formación del personal para hacer buen uso de esta tecnología.

El éxito en un SIG exige planteamiento y planificación previa clara mirando desde la perspectiva del tiempo, técnica de recursos y de personal empleado. Se hace necesaria la planificación del proyecto que principalmente debe contener el análisis de necesidades, una versión piloto, estrategias de contingencia, y un análisis de requerimientos minucioso y exacto. Un elemento indispensable, y probablemente el inicial en la tarea del desarrollo de un SIG en una organización son las personas específicas que se dedicarán a cada tarea y que exige el trabajo con geo información. Pues a causa de la generalización y uso extendido de las aplicaciones SIG en el tiempo actual, no solo los especialistas entran en juego, sino también se han adicionado perfiles diferentes de colaboradores en el área afín a los SIG. Adicionalmente, y probablemente más importante es el personal que se dedicará a la observación y uso para la toma de decisiones de la información geo espacial, lo que en realidad provoca inteligencia en los datos representados y su uso masivo.

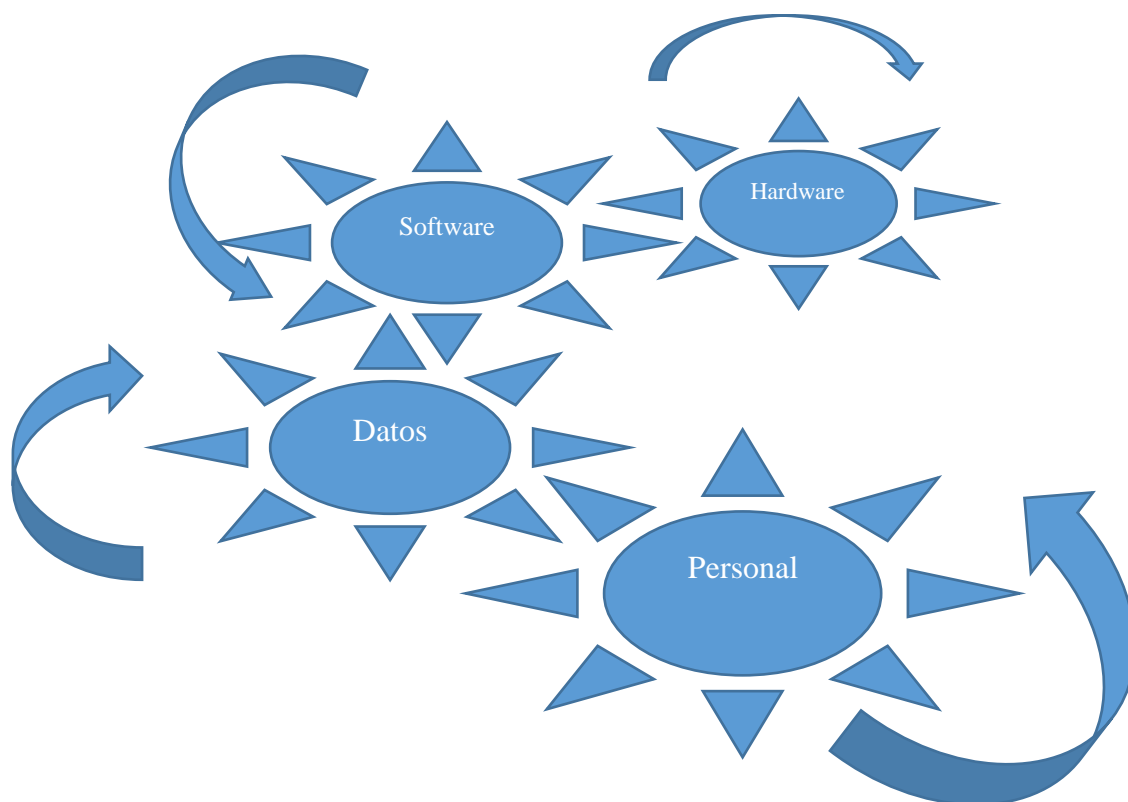


Figura 2 Elementos que conforman un SIG

2.3.3 Sistema de Coordenadas

Según Antoni Pérez Navarro un sistema de coordenadas es “una creación artificial que permite la definición analítica de la posición de un objeto o un fenómeno. Hay múltiples opciones para definir analíticamente la situación geométrica de un elemento y, por lo tanto, es posible escoger entre diferentes sistemas de coordenadas”. (Navarro, 2011).

Según Juan Peña Llopis se define al sistema de coordenadas como: “el proceso utilizado para relacionar la posición de un objeto o superficie en un plano o en archivo raster o vectorial con su posición en la superficie terrestre”. (Llopis, 2006).

Al ser nuestro principal objetivo la determinación de un punto sobre la superficie terrestre, desde el punto de vista matemático, los sistemas de coordenadas en su conjunto son aceptables, sin embargo, para seleccionar entre uno u otro simplemente prima la conveniencia o la simplicidad para manejar los datos. Para ser prácticos al elegir el sistema de coordenadas necesitamos que el objeto en cuestión se pueda determinar y visualizar en el espacio.

2.3.4 Caracterización Tecnológica de los métodos de almacenamiento de datos espaciales.

a. Concepto de almacenamiento de datos

Según Enrique Rivero Cornelio, Carlos Guardia Rivas, José Carlos Reig Hernández en su libro Bases de datos relacionales: diseño físico conceptualiza el almacenamiento de datos como “la forma transitoria o permanente que se utiliza en los ordenadores para guardar información”. (Cornelio, Rivas, & Hernández, 2004)

Según Enrique Rivero Cornelio, Carlos Guardia Rivas, José Carlos Reig Hernández en su libro Base de Datos relacionales; diseño físico establece diferencias entre memoria y almacenamiento mencionando que “Se llama memoria a todo dispositivo que se utiliza en los ordenadores para almacenar, de forma transitoria o permanente, cualquier información “[...], Tal como pueden comprobarse en los párrafos anteriores, existen dos vocablos, memoria y almacenamiento, que en el

entorno de las tecnologías de la información se pueden considerar equivalentes”- (Cornelio et al., 2004).

De acuerdo a C. J. Date en su libro *Introducción a los Sistemas de Base de Datos*, menciona refiriéndose al almacenamiento de datos y sus formas lo siguiente: “Un determinado archivo almacenado puede ser implementado físicamente en el almacenamiento en una amplia variedad de formas. Podría estar contenido completamente dentro de un solo volumen de almacenamiento [...]” (C. J. Date, 2001).

De los manifiestos anteriores el criterio asumido como concepto de almacenamiento se encuentra en el libro de Enrique Rivero Cornelio, Carlos Guardia Rivas, José Carlos Reig Hernández, debido a que cumple con las características necesarias para nuestro estudio.

b. Formato de datos y de intercambio

En el pasado, el desarrollo de formatos de intercambio se centraba en las necesidades del fabricante. Debido a que no había una institución que se ocupara de desarrollar formatos de intercambio, la perspectiva del fabricante era la única que el usuario podía aplicar para transferir datos de manera eficiente y sin pérdidas. Los desarrolladores de sistemas de software tenían los siguientes objetivos (naturalmente con diferentes prioridades):

- Transferencia de datos entre los productos de software propios con diferentes formatos internos.
- Transferencia de datos entre diferentes versiones y plataformas del mismo producto.
- Transformación de datos a un formato más adecuado para el almacenamiento de archivos, diversos portadores de datos y la comunicación remota.
- Recepción de datos con formatos pre-establecidos por fuentes públicas
- Recepción de datos de competidores (principalmente por importación, la exportación era mucho menos frecuente)

A pesar de ello si había consideraciones generales que debían cumplirse de acuerdo a procedimientos generales especificados por cada fabricante. (Pascoe, 1994).

El intercambio típico entre diferentes sistemas tiene lugar, en la mayoría de las veces, con el soporte de formatos externos específicos del sistema (aunque la aceptación de este tipo de formatos está reducida a unos cuantos que se utilizan para el intercambio entre sistemas de terceros, por ejemplo: DXF):

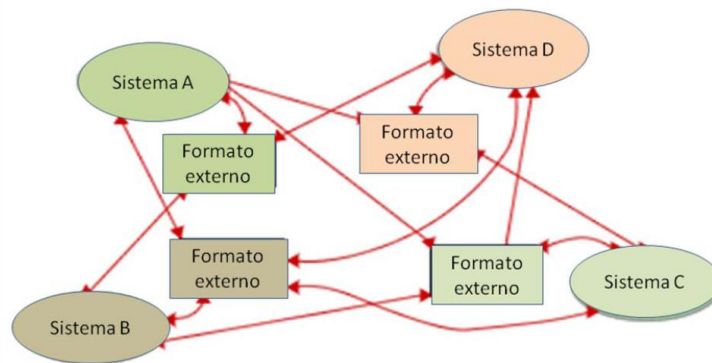


Figura 3 Intercambio de datos entre sistemas

La transmisión directa entre diversos sistemas excepcionalmente difícil en la mayoría de los casos, debido al carácter propietario de los formatos internos y su complejidad. A esto hay que sumarle la estabilidad de los formatos externos en el largo plazo que debe asegurar la continuidad en el intercambio entre los sistemas propios de diferentes tipos, versiones, plataformas, etc.

Los formatos externos poseen una representación completa de las estructuras internas por lo menos en lo que se refiere a los sistemas del respectivo fabricante. Como complemento de los formatos externos se puede hablar de interfaces primarias simples coordinadas en formatos ASCII libres. Estas interfaces sirven para capturar datos digitales provenientes de diferentes fuentes (sistemas de medición, GPS, sistemas digitales diversos, etc.) y que carecen de estructura.

Las diferencias esenciales entre los formatos externos está sujeta a la apertura del fabricante respectivo. Mientras que algunos de estos formatos son relativamente sencillos y parcialmente documentados de manera oficial, como por ejemplo:

Shapefile de ESRI en la descripción técnica; algunos otros con estructuras internas parecidas son consideradas propiedad intelectual del fabricante y no están documentadas para el público. Debido a que en la transmisión de datos se requieren estructuras simples, en la mayoría de las ocasiones es posible identificar, aún sin documentación, la estructura de un formato de transferencia ('reverse engineering') y así programar las interfaces requeridas.

El formato de exportación de ArcInfo es una buena instancia para ejemplificar el caso anterior. Este formato puede ser leído por otros productos, por lo menos en la versión ASCII sin compresión. Actualmente, también puede ser modificada, no sin algunos pleitos legales entre el fabricante y sus competidores. ESRI (fabricante de ArcInfo, el predecesor de ArcGIS) prohibía a sus competidores utilizar este formato para transferencias de datos. La consecuencia para uno de los países de la unión europea era que en las convocatorias para licitaciones gubernamentales en el área de sistemas geográficos había una cláusula que obligaba a los ofertantes a tener una interfaz del tipo 'Arc/Info Export', de no tenerla quedaban automáticamente excluidos de la licitación.

Si aparece un nuevo sistema en el mercado habrá que desarrollar para cada sistema existente una nueva interfaz para tratar de cubrir los requerimientos de los usuarios. Este esquema de trabajo se sigue con absurdo entusiasmo, debido a la creciente complejidad de los modelos de datos y estructuras dentro de los productos de software más que a la entrada de nuevos competidores al mercado.

Por esta razón, se propuso crear un concepto de interfaces de intercambio con topología de estrella, tal concepto propone crear un formato común como concentrador para la transferencia en cualquier dirección. De tal modo que cada nuevo producto requiere solamente una interfaz adicional:



Figura 4 Topología estrella para implementación de interfaces

Junto al hecho de que en este esquema se necesitan dos en lugar de una transformación para cada intercambio bilateral, este esquema de operación está plagado además de diversos problemas, a pesar de sus ventajas obvias y su atractivo. Los problemas a los que nos referimos yacen en particular en que las proporciones de las estructuras de la interfaz no solamente debe considerar las estructuras de dos sistemas, sino las prácticamente las estructuras de todos los sistemas, para poder garantizar un intercambio completo de datos.

Por último, la adopción de una arquitectura central para formatos de intercambio se complica demasiado sobre todo cuando se cuenta con la entrada de nuevos sistemas a la escena. Por dicha razón, la alternativa de un formato de intercambio estándar centralizado está plagada de problemas, ya que, no se pueden tomar en consideración estructuras específicas del sistema poco frecuentes.

Cabe señalar, que la iniciativa puesta en marcha desde principios de los años 80s con el propósito de crear estándares y un común denominador que reconozca las estructuras más importantes entre las marcas líderes en el mercado ha sido recientemente concluida dando como resultado una norma oficial. Con estos estándares oficiales continuaremos nuestro análisis en las secciones siguientes. A continuación presentamos un resumen de los formatos de intercambio específicos de los fabricantes que son los más importantes en la práctica.

c. Formatos Geográficos

Utilizaremos el nombre “formatos geográficos” para diferenciarlos de los formatos gráficos: los primeros indican por lo regular coordenadas en sistemas relacionales del espacio, los objetos son identificados claramente y la topología y atributos también pueden estar considerados. Por el contrario, los formatos gráficos típicos están orientados a entregar sistemas de coordenadas gráficas, los objetos están incluidos solamente para señalar partes de dibujo, los atributos y la topología no pueden ser transmitidos.

DXF –Drawing Interchange Format .- Como es natural, también podemos encontrar formas intermedias e híbridas, de tal forma, que en algunas ocasiones lo importante no es el formato sino su utilización. Así es que DXF (AutoCAD) es en

sus fundamentos un formato característico de transferencia de diseño gráfico (Drawing eXchange Format), sin embargo, también puede realizar tareas de transmisión de datos espaciales.

DXF es un formato de archivo desarrollado por la compañía Autodesk para el paquete de diseño asistido por computadora AutoCAD. Un archivo DXF contiene la descripción de un esquema o modelo basado en el estándar ASCII. Cada vez que aparece una nueva versión de AutoCAD vienen incluidas nuevas extensiones del formato DXF. Las versiones de DXF se denominan análogamente a las versiones de AutoCAD. DXF se puede importar a diferentes paquetes como un gráfico vectorial. DXF facilita también el intercambio de datos CAD entre programas de diferentes fabricantes, desafortunadamente, presenta algunas deficiencias. Frecuentemente, se pierde información o se representa de otras formas en los sistemas receptores (las diferencias se reflejan en los tipos de letra, dimensiones, símbolos externos).

DXF almacena típicamente múltiples tipos de entidades en niveles dentro de un archivo. Los elementos geométricos soportados son, por ejemplo, punto, línea, curvas, círculo, líneas de enlace, texto, símbolos externos y tipos de letras, bloques, dimensiones y otros. En el formato archivo de formas de ESRI, por el contrario, solo se permite un tipo de elemento geométrico por archivo. La eficiencia de una interfaz DXF se deriva del número de tipos de entidades DXF que se pueden representar en un sistema de información geográfica. Otro formato de Autodesk con la extensión .DWG es propietario y no está públicamente documentado.

GML— Geography Markup Language .- GML fue desarrollado como un lenguaje basado en XML con el objetivo de modelar, transportar y almacenar objetos con representaciones espaciales. Para cumplir con este propósito, atributos geométricos como técnicos deben estar soportados. En lo referente a la aceptación de GML como formato universal de intercambio de datos, parece que el tamaño actual del lenguaje impide su establecimiento definitivo. A través de la creación de perfiles, existe la posibilidad, de referirse únicamente a un subconjunto de constructos GML y de esta forma reducir la curva de aprendizaje del lenguaje.

KML - Keyhole Markup Language.- En abril del 2008, Keyhole Markup Language (in Version 2.2) fue elevado al rango de estándar OGC para la descripción

de datos geográficos. Este estándar fue desarrollado originalmente como lenguaje propietario para Google Earth. Al igual que GML, KML es un lenguaje de etiquetas (tags) con sintáxis muy parecida a la de XML y GML. Sin embargo, GML está definido mucho más ampliamente. Esto se demuestra en el hecho de que KML incorpora exclusivamente WGS84 como sistema de relaciones espaciales. KML es adecuado sobre todo en áreas donde se le da prioridad a la descripción de la visualización de información geográfica para representar contenido GML.

GDF – Geographic Data Files (ISO 14825:20044) .- Los archivos de datos geográficos 5(GDF, Geographic Data Files) son, por un lado, un modelo específico de datos, por el otro, un formato internacional de intercambio para la descripción y representación vectorial de entidades relacionadas con redes viales. Va más allá de una función de intercambio hasta establecer reglas para la adquisición de datos y la asignación de atributos. GDF describe objetos espaciales en 3 niveles: topología, objeto, objeto complejo. Por su estructura se trata de un formato de texto ASCII abierto. Tiene gran importancia, sobre todo en el área de mapas digitales viales o de navegación en vehículos automotrices. Sin embargo, no se puede hablar de una base de datos geográficos en GDF, más bien es el usuario quien lo transfiere en su sistema preferido.

DLG – Estándar del censo geológico de los EE.UU. Sirve para archivar y transmitir mapas digitales topográficos (solamente vectores).

GRASS – Formato del sistema de información geográfica GRASS (open source).

TIGER/DIME – Formato de transferencia vialidades y/o directorios desarrollado por la oficina de censos de los EE.UU.

2.3.5 Bases de datos Espaciales

Introducción

El método de base de datos ha evolucionado para solucionar, con bastante éxito, los problemas experimentados con los sistemas orientados a la aplicación. Aquí, la relación entre datos y aplicaciones se invierte. En el método de base de datos, en lugar de centrarnos en la aplicación y adaptar los archivos a dicha aplicación, el

centro de todo son los datos, y las aplicaciones trabajan sobre una base de datos central compartida, construida cuidadosamente y gestionada a conciencia. La base de datos de una organización se ve como un tesoro permanente, mientras que se reconoce que las aplicaciones pueden ser efímeras. Adicionalmente, al ser espacial, damos un plus a la información, porque ya no son solo datos vacíos los representados sino que le decimos en donde se encuentra dichos datos, pues los representamos en el mapa.

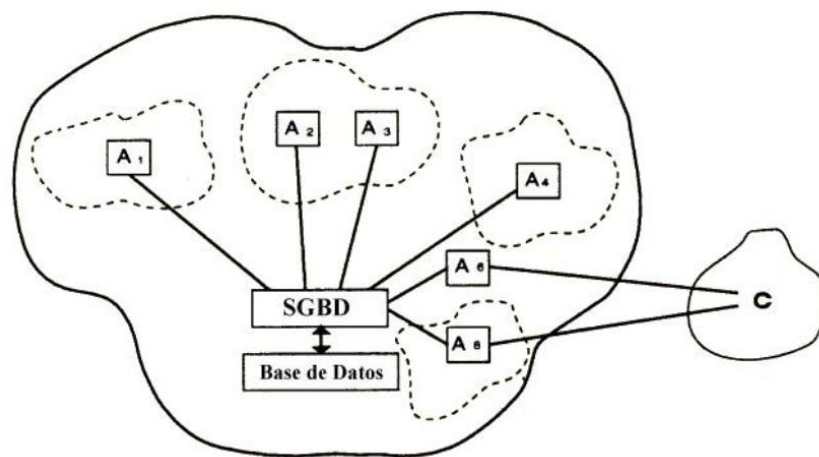


Figura 5 Sistema de gestión de Bases de Datos Espaciales

La Figura resume la estructura básica del método de base de datos para el tratamiento digital de la información. En este caso, en lugar de tener un gran número de archivos de datos enlazados a aplicaciones específicas, los datos de la organización se almacenan en una base de datos central. Se establece un Sistema de Gestión de Base de Datos (SGBD o DBMS - DataBase Management System) para mantener la base de datos y los programas de aplicación individuales acceden a los datos a través del SGBD; dentro de los SGBD también se incluyen los datos espaciales, llegando a formar parte integral del sistema, representado en la figura como un elemento adicional fuera del concepto tradicional de un gestor de base de datos.

Seguidamente vamos a estudiar algunos de los elementos clave del método de base de datos y sus definiciones.

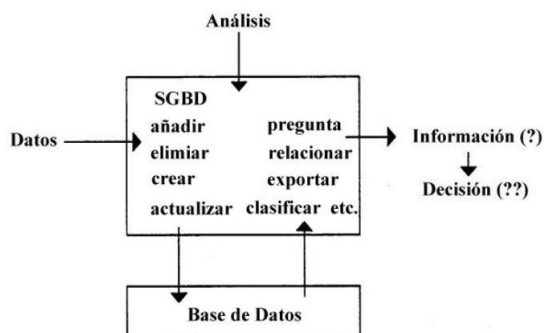
a. Datos

Los Datos son hechos puros u observaciones que se almacenan en una base de datos. Los datos, por sí mismos, tienen un valor directo pobre ya que no tienen ni estructura ni contexto ni, por lo tanto, posibilidad de ser interpretados.

Una Base de Datos es un cuerpo de datos relacionados almacenados de forma estructurada. Las bases de datos no tienen, necesariamente, que estar informatizadas. Un archivador con datos organizados con una secuencia lógica es una base de datos (como por ejemplo, un sistema de organización en fichas).

La Información es el resultado de la explotación de los datos. Los datos a menudo necesitan ser ordenados, o resumidos, o reducidos, o combinados con otros datos para hacer aparente la información que está sumergida en el gran volumen de datos sin estructurar. La información es el patrón que los seres humanos esperan reconocer en los datos: un satélite recoge datos, pero sólo los seres humanos pueden interpretar los patrones de puntos como un tipo particular de entidad geográfica. Los ordenadores pueden ayudarnos a cribar los datos, pero la información, al fin y al cabo, es una cosa subjetiva. Es una visión determinada construida a partir de los datos. La toma de decisiones necesita información, no datos.

Las bases de datos son dispositivos que facilitan nuestra búsqueda de información a partir de los datos. Mediante la interrogación (o consulta) de la base de datos esperamos obtener información.



Base de Datos + SGBD + Análisis → Información (?) → Decisión (??)

Figura 6 Los SGBD crean información a partir de los datos

En la actualidad incluso se han desarrollado herramientas y técnicas que permiten analizar ciertas tendencias de los datos almacenados en una base de datos, con el fin de extraer cierto conocimiento que permita distinguir patrones que no se lo puede interpretar de primera vista, pero que al disponerlos se vuelven una fuente rica de información sobre todo relacionado con tendencias, que servirán a las grandes organizaciones para establecer estrategias de negocios, estas técnicas forman parte del Data Mining (Colecta de Datos), como otra posibilidad de explotación de conocimiento a partir de los datos, proceso que también se puede aplicar en los datos geográficos y establecer patrones de funcionamiento.

Un Sistema de Gestión de Bases de Datos (SGBD o DBMS) es, esencialmente, un software de propósito general que mejora la manipulación de una base de datos con el fin de “proveen facilidades para la manipulación de grandes volúmenes de datos” (Medina Narváez & Morán Cabrera, 2011)

Las bases de datos no informatizadas tienden a ser voluminosas, lentas y poco flexibles. Como ejemplo, los primeros sistemas de organización de las bibliotecas consistían en un sistema de fichas. Normalmente, una biblioteca necesitaba mantener varias indexaciones de las fichas: una ordenada por autor, otra por tema y quizá otra por número de adquisición. Evidentemente, hay una considerable duplicación de información y una considerable posibilidad de que los tres índices de fichas sean inconsistentes.

“Los beneficios que produce el almacenamiento de datos rugen del uso que se le haga de éstos. El empleo que se vaya a hacer de los datos puede ser desconocido o casi desconocido a priori, pero es necesario evaluar y medir, al menos, su probabilidad.” (Capote, 2008)

“Se le llama **base de datos** a los bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto” («Base de datos», 2015).

b. Sistema de Gestión de la Base de Datos, SGBD

“Sistema de Gestión de Base de Datos (SGDB) que es el software que controla y gestiona el acceso a la base de datos, y cuyo papel es cada vez más importante en el correcto funcionamiento de las aplicaciones actuales.” (Lope, 2015).

“Un SGDB proporciona del mismo modo a usuarios finales como a administradores y personal cualificado en general los mecanismos para definir, consultar y mantener la información existente en la base de datos” (García, 2015)

El SGBD utiliza el poder de la informática para solucionar las limitaciones de las bases de datos en formato papel. Un SGBD de una biblioteca, por ejemplo, almacenaría los detalles de un libro sólo una vez, pero sería capaz de visualizar dichos detalles como si hubiesen sido almacenados por autor, tema, número de adquisición o, por supuesto, en cualquier otro orden, según se requiera. El SGBD también permitiría una mayor flexibilidad en el análisis, resumen y sobre todo la velocidad en la presentación de datos de lo que permitiría un sistema de fichas. Actualmente los SGDB espaciales, tienen la funcionalidad de hacer el mismo tipo de consultas y flexibilidad de la información, pero no solo en datos alfanuméricos sino también lo hace en datos espaciales; así por ejemplo, digamos que tenemos una base de datos en la que también tenemos información de los predios urbanos de una determinada ciudad, pues con un SGDB espacial podemos ubicarlos en el mapa, pero también podemos organizarlo por extensión, por planimetría, por sectores, etc.

c. Funciones de un SGBD

Mientras que las aplicaciones se centran en ofrecer soluciones específicas a tareas concretas, los SGBD esperan ofrecer servicios generales valiosos para los creadores de bases de datos, sin tener en cuenta el contexto. Según María Victoria Nevado Cabello en su libro “Introducción a Las Bases de Datos Relacionales”, básicamente las funciones ofrecidas por cualquier SGBD pueden ser resumidas de la siguiente manera:

- Crear, modificar y borrar estructuras de datos: Cualquier SGBD debe permitir crear las definiciones de campo, las estructuras de ficheros y registros necesarias para recibir los datos de la organización. Como los requerimientos evolucionarán

durante la vida de la base de datos, el SGBD también tiene que tener la suficiente flexibilidad para modificar las estructuras de datos. Quizá añadiendo campos adicionales, introduciendo nuevos índices o borrando partes de la estructura de datos que ya no son necesarios.

- Añadir, actualizar y borrar registros: Habiendo ya establecido la estructura de datos, el SGBD debe facilitar la entrada de datos, sea mediante la importación de registros de datos o mediante la entrada por teclado. También tiene que permitir editar los registros existentes y borrar los innecesarios.
- Extraer información de los datos: Debe tener mecanismos para extraer información de la base de datos de manera comprensible. Debería ser posible consultar registros individuales, ordenar los registros según los criterios que se requieran, resumir los datos estadísticamente, producir informes tabulados y producir gráficos. También debería ser posible exportar los datos a otros sistemas (compatibilidad)
- Mantener la seguridad e integridad: Deben limitar el acceso a algunos datos y asegurar la integridad lógica de los datos almacenados.
- Construir Aplicaciones: Además de mantener la base de datos, la mayoría de SGBD deben incluir un lenguaje de programación o ser compatibles con alguno de los lenguajes existentes (C, PHP, Python, Perl, entre otros). Esto permitirá que las aplicaciones puedan tener acceso a la base de datos a través del SGBD.

Los SGBD son un potente conjunto de herramientas para los diseñadores de bases de datos. Como cualquier conjunto de herramientas, la calidad del producto final depende de la pericia de la persona que utiliza estas herramientas. La destreza del o la analista de bases de datos está en utilizar las herramientas de un SGBD para diseñar bases de datos que cumplan, eficientemente, con las especificaciones del cliente. (Cabello, s. f.)

d. Ventajas del método de base de datos

Las ventajas del método de base de datos encajan, a grandes rasgos, con los inconvenientes del método orientado a la aplicación que antes comentábamos:

Reducción de la redundancia de datos: Utilizando el método de base de datos, los diseñadores de bases de datos pueden eliminar por completo la duplicación de

datos. En la mayoría de las situaciones, una organización almacenará cada elemento una vez, y sólo una vez, en la base de datos central. Esto implica, claramente, un considerable ahorro de recursos (espacio de almacenamiento).

Aumento de la integridad de datos: Gracias a que los datos normalmente se almacenan una sola vez y a las reglas de integridad del software SGBD modernos, se mantiene la validez y consistencia de los datos. La posibilidad de que diferentes partes de una organización trabajen en copias diferentes de una misma serie de datos se reduce.

Independencia de datos: Como los usuarios y las aplicaciones se basan en la visión conceptual, los profesionales de la base de datos pueden alterar las realidades físicas de ésta sin que los usuarios se den cuenta de los cambios. Las inversiones hechas por los usuarios para familiarizarse con la estructura conceptual de su base de datos y los esfuerzos en desarrollar aplicaciones que operen en el nivel Conceptual o de Visión de Usuario, pueden generar beneficios a largo plazo. La independencia de datos ayuda a ofrecer un entorno informático estable para los usuarios.

Seguridad de los datos: Utilizar un SGBD permite a los expertos controlar quién tiene permiso para acceder a determinadas partes de la base de datos de la organización.

También significa que pueden mejorarse los procedimientos de creación de copias de seguridad.

Mantenimiento eficiente: Como todos los datos se almacenan centralmente y todas las aplicaciones extraen la información de la base de datos, el mantenimiento es mucho más sencillo. Imagina, por ejemplo, que la Oficina de Correos decide añadir un dígito más a su estructura de código postal. Si contamos con un sistema orientado a la aplicación con diez aplicaciones que utilizan el código postal, el personal de mantenimiento necesitará hacer las alteraciones diez veces. En el método de base de datos, sólo deberán modificar la definición de 'código postal' almacenada en el Diccionario de Datos del SGBD.

Visión Corporativa/Central: En el método de base de datos, la base de datos es un recurso clave para la organización, y su estado y futuro desarrollo deberán ser

cuidadosamente planificados. Con este método parece más fácil desarrollar estrategias de planificación corporativas.

2.3.6 El modelo relacional

El Modelo Relacional, existió como una manifestación teórica previa a la aparición de cualquier software de base de datos relacional. Según la historia señalada por José Manuel Cabello García en su libro “Almacenamiento de la información e introducción a SGBD. IFCT0310”, en 1970 Edgar F. Codd, entonces científico de IBM, publicó un artículo en el que describió el modelo de base de datos Relacional, para ilustrar cómo pensaba él que debía diseñar las bases de datos, partiendo de la teoría y lógica ya existentes. Los SGBD ‘relacionales’ de hoy en día (SGBDR o RDBMS – Relational DataBase Management System) son intentos de implementar la visión teórica de Codd en un software del mundo real. (García, 2015)

Raras veces una idea puede tener un efecto tan inmediato y profundo sobre la industria. Desde mediados de los setenta hasta principios de los noventa, la historia del desarrollo de bases de datos consistía, básicamente, en desarrolladores de software que producían SGBDR operacionales basándose en, al menos, algunos elementos del Modelo Relacional teórico. Para grandes ordenadores (mainframes), IBM introdujo, a principios de los ochenta, SQL/DS y DB2. Ahora, hay otros SGBDR para mainframe que incluyen Oracle, Sybase, Ingres, Informix, Empress, Unify y muchos otros. La popularidad del método relacional es tal que prácticamente todos los sistemas de bases de datos afirman ser ‘relacionales’. Algunos sistemas ampliamente utilizados que, en cierta medida, siguen los principios relacionales son PostgreSQL, MS SQL Server, Paradox, Oracle, FoxPro, Access y MySQL.

Como muchos de los SIG actuales fueron inicialmente desarrollados durante los ochenta, cuando el Modelo Relacional estaba a la orden del día, no es sorprendente que la mayoría de ellos asuman el método relacional en lo referente a base de datos. Varios SIG utilizan una base de datos relacional convencional para almacenar los datos alfanuméricos. El software ARC/INFO GIS de ESRI, por ejemplo, utilizó el paquete INFO, uno de los primeros SGBDR, para almacenar datos alfanuméricos. Posteriormente, ESRI ha enlazado su software ARC con SGBDR más modernos, como Oracle, para el tratamiento de los datos alfanuméricos. IDRISI para Windows

utiliza una versión limitada de la base de datos MS Access para el tratamiento de los atributos (datos alfanuméricos). El software Atlas GIS, MapInfo y ArcView también utilizan un modelo relacional restringido para almacenar los atributos basados en archivos dBase estructurados. El sistema Geo/SQL utiliza, de forma similar, un SGBD relacional.

Algunos SIG utilizan bases de datos relacionales no sólo para organizar los datos alfanuméricos sino también para almacenar los datos geográficos. Por ejemplo, VISION*GIS utilizó el SGBDR Oracle para gestionar los datos alfanuméricos y geográficos, como lo hizo ESRI con su Spatial Database Engine. Cuando se trata de acceder a bases de datos externas, los SIG acostumbran a asumir que éstas son relacionales. De la misma manera, y no menos importante, a tal punto que se compara a Oracle, Postgresql ha evolucionado tanto su desarrollo, que con su módulo PostGIS especializado en el manejo de información espacial, que unido a los datos alfanuéricos, hacen de éste SGDB el ideal para desarrolladores expertos en SIG.

Si echamos una mirada más detallada de la estructura desarrollada de los SIG, veremos que uno de los objetivos de estos es el almacenamiento de información espacial, que en realidad son datos, que luego de mantener una estructura adecuada a su objetivo necesitarán ser consultados.

a. Los tres elementos del Modelo de Datos Relacional

De acuerdo (Galván Salinas & Jiménez Alfaro, s. f.) en su artículo “Fundamentos de la teoría relacional para extracción de datos MultiValores del RDBMS jBase” El Modelo Relacional de Codd tiene tres elementos:

- a) Un elemento estructural que describe la forma en que pueden guardarse los datos.
- b) Un elemento de manipulación que describe un número de operadores que permiten a los usuarios procesar los datos guardados en formato relacional
- c) Un elemento de integridad que propone reglas para asegurarnos que los datos se gestionan con validez y consistencia

b. El elemento estructural

Superficialmente, el elemento estructural del Modelo Relacional no podría ser más sencillo, ya que todos los datos parecen estar almacenados en tablas bidimensionales. Así, para los usuarios, el Modelo Relacional es un modelo conceptual muy familiar, permitiéndoles creer que los datos se guardan dentro del ordenador de una forma muy parecida a las tablas en papel con las que estamos tan familiarizados.

Sin embargo, en la realidad las cosas no son tan simples, ya que las tablas contenidas en un SGBDR deben cumplir unos requisitos específicos para que el software funcione correctamente. Aunque el modelo conceptual de un usuario puede, simplemente, consistir en tablas de almacenamiento de datos, el diseñador de bases de datos debe ser consciente de las restricciones que puso Codd sobre el formato de las tablas relacionales y tiene que diseñar una estructura de datos que se adapte a estas restricciones.

Una tabla relacional debe tener, especialmente, las siguientes características:

Todas las columnas deben tener nombres distintos: Los nombres de las columnas se utilizan como referencia para acceder a los valores que hay en ellas. Si usamos el mismo nombre para dos columnas distintas nuestro sistema no sabrá cuál de ellas debe utilizar, por eso los SGBD no permiten generar dos columnas con el mismo nombre en la misma tabla.

Todas las entradas de una columna deben ser del mismo tipo: La mayoría de las personas que utilizan el ordenador estarán familiarizadas con la idea de que las variables pueden ser declaradas como Reales (es decir, un número con una parte decimal), Enteras (un número sin decimales) o como Cadenas de caracteres (es decir, cualquier símbolo alfanumérico) y sabemos que si intentamos poner un real, digamos 4.34, en una variable que ha sido declarada como entera el ordenador se quejará. El Modelo Relacional toma esta idea y define que cada columna de una tabla sólo puede tener valores del mismo dominio (se llama dominio al conjunto de todos los valores posibles que el elemento puede adoptar). Una columna que se llame Día, por ejemplo, tendría un dominio de números enteros positivos que irían sólo de 1 a 31. Si

intentamos introducir los números 32 o -1 en la columna Día causaríamos un error. Una columna llamada ‘Capital de Ciudad’ sólo puede tener entradas pertenecientes al conjunto de capitales de ciudad. En el modelo de Codd sólo se pueden hacer comparaciones entre columnas que utilizan datos del mismo dominio. No tiene sentido en el mundo real preguntarse, por ejemplo, si el Día es igual a la Distancia, aunque los dos pueden ser representados como números enteros. El concepto de dominio es un intento de incorporar este tipo de conocimiento del mundo real en el Modelo Relacional. El concepto de dominio, por lo tanto, intenta evitar la entrada de datos ilegales y de prevenir comparaciones ilógicas.

Aunque el concepto de dominio es una construcción teórica valiosa, es uno de los elementos de Codd que, a menudo, no se implementa en un software comercial.

Todas las entradas en una tabla relacional deben ser ‘atómicas’: A veces, en las tablas en papel permitimos que una celda contenga más de un valor. Por ejemplo, en un formulario donde se consulta la nacionalidad deberíamos permitir que alguien con doble nacionalidad comprima los dos nombres en un único cuadro del formulario en papel. En los sistemas relacionales esto está prohibido. En las tablas relacionales cada celda debe tener una única entrada.

El orden de las columnas y filas no tiene importancia: En las tablas en papel a menudo utilizamos el mismo orden en que entramos los datos para tratar la información. Podríamos, por ejemplo, entrar los datos ordenándolos por año o almacenar detalles personales ordenándolos por apellido. En una tabla relacional esto no está permitido. Se tiene que asumir que los datos entrados en las columnas y filas están ordenados al azar.

Cada fila de la tabla debe ser diferente de las demás: En una tabla relacional no puede haber filas duplicadas. En otras palabras, debe haber una columna (o combinación de columnas) que identifique de forma única cada fila de datos, y aquí es donde nos encontramos por primera vez el concepto de ‘clave’. La clave es una columna (o combinación de columnas) con un valor único en cada fila. En ningún momento puede haber dos filas con el mismo valor. Por lo tanto, puede utilizarse para identificar de forma individual a cada fila. Si una tabla llamada ‘DetallesEmpleados’, por ejemplo, contiene una columna llamada

‘NumNacionalIdent’, entonces ésta sería una buena clave porque cada persona tiene un número de identidad nacional personal y único.

Es posible utilizar más de una columna (o combinación de columnas) para almacenar los valores que son únicos para cada fila. Si la tabla DetallesEmpleados anterior también contiene un NumEmpleado que es único para cada empleado, entonces tendremos dos campos que actúan como clave (NumNacionalIdent y NumEmpleado). Cuando hay más de una clave, cada una de ellas se llamará clave candidata. El modelo relacional exige, entonces, que se escoja una de las claves candidatas para acceder a las filas de la tabla. La clave candidata seleccionada pasará a llamarse clave primaria o principal, pasando la otra a denominarse clave secundaria o alternativa.

Cuando un conjunto de datos no tiene, de forma natural, ninguna clave candidata (no hay ninguna columna en la que los datos sean únicos para cada fila) es una práctica común (y esencial) añadir una columna adicional, como NumEmpleado, que actuará como identificador único para cada fila de la tabla.

En las bases de datos de un SIG, la localización a menudo actúa como clave primaria, partiendo del argumento (o requisito) que dos entidades del mismo tipo nunca pueden ocupar el mismo espacio. Al proceso de añadir un identificador espacial único a los datos alfanuméricos (atributos) se le llama geocodificación o referenciación espacial. Este proceso ofrece un gran número de temas interesantes. Por ejemplo, en una base de datos de propiedades, podríamos identificar cada una de las propiedades mediante la referencia UTM del centroide1 de esta propiedad.

Como alternativa, las administraciones municipales a veces asignan a las propiedades individuales unos códigos numéricos. Recientemente, también se adopta el sistema de código postal como clave (con un número de propiedad y su código postal identificamos la mayoría de las propiedades). Cada uno de estos métodos tiene ventajas e inconvenientes. La geocodificación, es decir, utilizar el espacio para identificar de forma única una entidad, es el caso especial de los SIG para establecer claves primarias dentro de la base de datos. El especialista está en la posibilidad de elegir sus propias claves siempre y cuando se ajusten a las condiciones planteadas para su selección

Valores nulos: El modelo relacional utiliza el concepto de valor nulo para indicar que hay un valor actual todavía desconocido o que no hay una entrada apropiada para esa celda. Supongamos que una tabla tiene una columna llamada NumTelefono, y nos encontramos con una vivienda sin teléfono. En algunos software, si dejamos la celda en blanco el programa interpretaría ese ‘en blanco’ como un valor 0, lo que no sería correcto. En el Modelo Relacional, la entrada apropiada sería Nulo para indicar que ese valor es ‘desconocido’. El valor Nulo no significa lo mismo que el valor 0 o que dejar la celda en blanco.

Desde el principio nos hemos referido a las estructuras del Modelo Relacional con el nombre de tablas, pero debido a su base matemática el modelo tiene una terminología formal distinta. En este módulo preferimos usar los comúnmente utilizados términos de tabla como fila, columna, etc., pero deberías ser consciente de la existencia de la terminología alternativa. A una tabla como la anterior se la llama relación, a las filas tupla y a las columnas atributo. El número de filas de una tabla es su cardinalidad y el número de columnas su grado.

c. El elemento de manipulación (álgebra relacional)

El elemento de manipulación de un modelo de base de datos es el conjunto de comandos mediante el cual podemos operar sobre los datos almacenados en el formato especificado por este elemento. Un aspecto distintivo del modelo relacional es que los comandos de manipulación operan sobre toda la tabla en lugar de hacerlo registro a registro. La entrada de cada comando es una o más tablas y la salida siempre es una nueva tabla. El funcionamiento del elemento de manipulación en la teoría de bases de datos relacionales se basa en las operaciones utilizadas en la Teoría de Conjuntos. Cada tabla puede ser contemplada como un conjunto y las operaciones crean nuevos conjuntos como combinaciones de los conjuntos existentes.

En el Modelo Relacional de Codd el elemento de manipulación consiste en un conjunto de operadores conocido como álgebra relacional. Los usuarios de un software de BD relacional, sin embargo, no utilizan esta álgebra directamente. El álgebra relacional existe para declarar, matemáticamente, lo que es posible en el

modelo relacional: ofrece un criterio con el que juzgar los lenguajes SGBDR implementados, como el SQL (Structured Query Language).

Como nuestro interés se centra, básicamente, en obtener un uso práctico de las bases de datos relacionales, la descripción del álgebra relacional que se ofrece aquí está reducida deliberadamente.

El álgebra relacional es un elemento sorprendentemente sencillo una vez hemos entendido su sentido básico, pero su simplicidad puede oscurecerse debido a los términos matemáticos que pueden utilizarse para describirla. Un buen modelo mental que podemos utilizar consiste en simular que tenemos un número de tablas en papel convencionales a partir de las cuales queremos crear nuevas tablas. Para hacer esto manualmente necesitaríamos llevar a cabo un conjunto de operaciones de ‘cortar, pegar y copiar’ hasta haber creado las nuevas tablas que queríamos. El álgebra relacional puede ser vista como un servicio que nos ofrece el equivalente electrónico de unas tijeras, pegamento y fotocopidora para trabajar sobre las tablas electrónicas contenidas en el SGBDR.

Hay tres operadores relacionales fundamentales:

- **Select**

Selección, crea un subconjunto con todas las filas de una tabla que cumplen con un criterio determinado.

- **Join**

Join es el operador crucial que permite que los datos almacenados en tablas separadas se ‘peguen’ juntos temporalmente en una columna compartida.

- **Union**

Unión, añade dos tablas en una sola.

- **Difference**

Diferencia resta dos tablas.

- **Intersection**

Intersección: crea, a partir de dos tablas, una nueva tabla con los registros comunes. En otras palabras, la tabla creada con este operador contendría todos los registros que están presentes en ambas tablas.

d. El elemento integridad

La integridad de una base de datos se refiere a la exactitud y grado de corrección de los datos que están almacenados en ella. En el modelo relacional de Codd se proponen dos tipos de integridad: integridad de entidades e integridad referencial.

Integridad de Entidades.- Ésta hace referencia a la clave primaria de la tabla y simplemente consiste en que cada fila en la tabla relacional tendrá una entrada válida (es decir, no nula) y única para los atributos de la clave primaria. Si no se impusiera esta regla podría suceder que las filas de una tabla existieran, pero no fueran accesibles. La mayoría de los sistemas relacionales incorporan este elemento de integridad.

Integridad Referencial.- Esta regla requiere que los valores de una clave externa existan como clave primaria en otra tabla. Si esta regla no existiera sería posible encontrar una fila con una clave externa que apunta a una fila de otra tabla que no existe.

2.3.7 Normalización

La normalización es otra técnica de modelado de datos para el desarrollo de un diseño lógico que, a diferencia del modelado EAR, evolucionó directamente del Modelo Relacional, por lo que un conjunto de tablas que hayan sido normalizadas con éxito tienen la garantía de cumplir con las expectativas del Modelo Relacional.

Las características de la Normalización son:

- **Método de abajo hacia arriba**

Si EAR es un método de arriba hacia abajo, es decir, de lo más general a lo más detallado, la Normalización empieza con el detalle de los atributos y aplica una serie de transformaciones para crear una estructura general partiendo del detalle.

- **Formal**

Si EAR es un método visual con el cual uno puede ‘garabatear’ el camino hacia una solución, la Normalización es un método que sigue un conjunto de reglas que deben ser aplicadas en una secuencia particular de forma que se asegure la creación de las tablas apropiadas. (Kendall, 1997) señala que la Normalización es la transformación de vistas de usuario complejas y almacenes de datos a un conjunto de estructuras de datos estables más pequeñas y sencillas”. Algunos textos parecen olvidar esta simplicidad y presentan exposiciones del tema muy rigurosas. Intentaremos encontrar el punto medio.

- a. Reunir una lista con todos los atributos**

Si la técnica de Normalización se usa independientemente de cualquier otra técnica, entonces el punto de partida sería establecer una lista de todos los atributos asociados con un sistema. Teóricamente, esto puede entenderse como el paso de reunir una única tabla prototipo que incluya todos los atributos.

- b. Identificar las tablas iniciales**

El paso siguiente a la tabla prototipo la primera tabla con todos los atributos, consiste en reunir una serie de tablas provisionales cada una de las cuales parece agrupar atributos que pertenecen a la misma entidad. Si reconocemos que en una base de datos real puede haber varios cientos de atributos, entonces sabremos que la tarea inicial del diseñador para identificar entidades y asignar atributos a las entidades puede ser muy dura. Por esta razón, muchos diseñadores usarían una versión informal del modelado EAR que les ayudaría a producir un conjunto de tablas provisionales sobre las cuales podrían aplicar después las técnicas de normalización.

c. Primera Forma Normal (1FN)

La Primera Forma Normal implica simplemente la retirada de los grupos repetidos de una tabla. Como ya hemos encontrado el problema de los grupos repetidos en el modelado EAR, podemos tratar esto de forma rápida.

d. Segunda Forma Normal (2FN)

La Segunda Forma Normal se usa para eliminar un problema que sólo puede aparecer en tablas en las que la clave primaria es una combinación de atributos.

e. Tercera Forma Normal (3FN)

Ahora ya hemos creado tres tablas más pequeñas a partir de nuestra tabla inicial. De nuevo, la duplicidad de datos es el signo de alarma. Si aún existen tablas o datos repetidos, la única forma de solucionarlos es volver a dividirlos.

f. Cuarta y Quinta Formas Normales

Las formas normales Cuarta y Quinta tratan casos especiales que parecen existir básicamente en los sueños de los Informáticos. También existe un método alternativo a la normalización llamado Normalización de Boyce-Codd.

Como nuestro propósito es simplemente saber que las tablas que diseñamos funcionarán adecuadamente podemos ignorar tranquilamente estas otras formas de normalización, bastándonos saber de su existencia.

g. Comprobar las relaciones

Como hemos visto, 1FN, 2FN y 3FN se concentran en refinar una tabla concreta en una serie de tablas más pequeñas y eficaces. Podemos esperar que esta familia de tablas poseerá las relaciones de clave externa necesarias para enlazarlas correctamente cada una de sus tablas individuales.

2.4 Antecedentes Contextuales

El 05 de Junio de 1957 por decreto ejecutivo No. 98287 se crea el Honorable Consejo Provincial de la Provincia de Chimborazo, en la ciudad de Riobamba;

inicialmente sus labores las efectuaban en el edificio otorgado en las calles 10 de Agosto y España, posteriormente en el año 2008 el Gobierno Central otorga la titularidad del edificio donde funciona actualmente en las calles Primera Constituyente y Carabobo esquina.

El Gobierno Provincial es netamente una Institución que forma parte del estado ecuatoriano, con autonomía económica y administrativa, regido por la COOTAD especialmente.

De acuerdo al PDOT (Programa de Desarrollo y Ordenamiento Territorial) Provincial establece lo siguiente:

“La entrada en vigencia del COOTAD y la creación del Consejo Nacional de Competencias, dentro del proceso de descentralización que vive actualmente el Ecuador, ha dejado en evidencia la necesidad de avanzar en forma acelerada en el proceso de descentralización. Existe 8 competencias exclusivas asignadas a los gobiernos provinciales en el Artículo 263 de la Constitución vigente, entre las que están: el riego, fomento productivo (incluyendo al sector turismo) y fomento agropecuario, con el mandato de asumirlas a corto plazo en el territorio. Por lo dispuesto en el artículo 263 de la Constitución no se debe pensar que la rectoría del desarrollo económico y el ejercicio de la competencia del fomento productivo es exclusiva de un sólo nivel de gobierno, en este caso del Gobierno Autónomo Provincial. Efectivamente, en los artículos 262 y 267 de la Constitución y el 42 del COOTAD, asignan como una competencia exclusiva, pero a su vez de gestión concurrente, a los gobiernos regionales y a los gobiernos parroquiales, aspecto que es ratificado en el COOTAD en sus artículos 32 (competencias exclusivas de los gobiernos regionales) y 65 (gobiernos parroquiales)”

La propuesta de desarrollo provincial, se configura sobre la base de cuatro pilares:

- Sistema de Fomento Productivo
- Sistema de Gestión Ambiental
- Sistema de Equidad e Inclusión
- Sistema de Gobernabilidad

Para la justificación del problema planteado en el presente estudio, se ha elaborado un instrumento de investigación el cual es una encuesta aplicado a 10 miembros del equipo de trabajo del Departamento de Planificación de Gobierno Provincial de Chimborazo, de los cuales se obtuvieron los siguientes resultados:

Encuesta:

1.) Como guarda usted los archivos georeferenciados tomados en campo del inventario vial?

Resultado: Luego de la aplicación de la encuesta se obtuvo un 90% de encuestados que usan shapefiles para almacenar la geo información, un 10% lo hace en archivos Excel, nadie mencionó una geodatabase ni ningún otro tipo de almacenamiento.

2.) A tenido inconvenientes al momento de obtener el Sistema de Inventario Vial actualizado?

Resultado: Un 100% de los encuestados respondieron afirmativamente a la pregunta de si tuvieron inconvenientes al acceder al Sistema del Inventario Vial actualizado, con estas respuestas confirmamos que en realidad existe un problema en la actualización de la información del Inventario Vial.

3.) Con qué frecuencia considera que los datos del Sistema de Inventario Vial tiene que ser actualizado?

Resultado: El 100% de personas encuestadas respondió que una vez al año debería ser la frecuencia para actualizar el Inventario Vial; de los resultados se puede establecer que la frecuencia de actualización es muy importante.

4.) Cómo realiza las búsquedas de Información de un campo en el Inventario Vial?

Resultado: 8 de las 10 personas encuestadas eligieron la opción b, es decir que para buscar cierta información utiliza los shape desplegando todo el contenido; mientras que 2 de las 10 personas lo hace con un editor de cliente abriendo las capas del documento. De estos resultados se hace evidente el

problema que significa buscar algún campo específico de la información del Inventario Vial.

5.) Ha encontrado datos duplicados o información repetida en los shapefiles del Inventario Vial?

Resultado: El 100% de los encuestados respondieron que si encuentran datos duplicados en el Inventario Vial. De los resultados se hace notorio el problema que existe en cuestión de integridad de datos en el Sistema de Inventario Vial.

6.) Considera útil centralizar la Información del Inventario Vial en una geodatabase?

Resultado: La totalidad de los encuestados el 100% considera útil la centralización de la Información del Inventario Vial en una geodatabase. Aquí podemos claramente notar una posible solución al problema presentado en el Inventario Vial.

7.) Cree usted que la administración del Inventario Vial estaría mejor administrada en una geodatabase en comparación de shapefiles individuales como actualmente los maneja?

Resultado: El 100% de encuestados responde afirmativamente a la pregunta de si estaría mejor administrada el Inventario Vial en una geodatabase.

8.) Cree útil un Sistema de Información del Inventario Vial publicado en la web?

Resultado: El 100% de los encuestados respondieron que si sería útil la publicación de la Información del Inventario Vial en la web.

9.) Apoyaría la creación de una geodatabase con el fin de que la información se actualice en tiempo real y sea pública?

Resultado: De los encuestados el 100% apoyaría la creación de una geodatabase para tener actualizado el Sistema de Inventario Vial y a la vez la información del mismo sea masiva.

Analizado los resultados provenientes de la encuesta, el 90% de los miembros del equipo técnico del Departamento de Planificación coinciden que tienen problemas con el manejo de los shapefiles en el inventario vial de la Provincia de Chimborazo dados especialmente por la no centralización de los archivos, la lentitud en las búsquedas de información, la inconsistencia de datos entre los diferentes archivos.

De los resultados de la encuesta podemos decir con certeza que el 100% de los encuestados apoyarían la solución a este problema con el desarrollo de una Geodatabase que centralice la información del Inventario Vial.

2.5 Conclusión del capítulo

La evolución histórica que ha tenido los métodos y técnicas de almacenamiento en los Sistemas de Información Geográfica ha ido contribuyendo a la mejora de la administración de la información y consecuentemente con ello la optimización en la toma de decisiones de la alta gerencia.

Los métodos de almacenamiento usados en los GIS han ahorrado tiempo y recursos en las empresas; que a pesar de su elevado costo, implementarlo resulta más económico que no tenerlo.

Conocer a detalle los métodos y técnicas de almacenamiento así como los conceptos de GIS, mejoran la conceptualización en el desarrollo de este documento de investigación, paso importante y esencial para la consecución del trabajo.

CAPÍTULO III

DESARROLLO DEL FRAMEWORK

3.1. Introducción.

El objetivo del capítulo es establecer el modelo simplificado de la secuencia de pasos a seguir para el desarrollo del marco de trabajo, y con ello trasladar desde un archivo shape a una base de datos espacial, iniciando desde el análisis previo, siguiendo el modelo EAR, posteriormente el análisis de la velocidad de respuesta así como el tamaño de la base de datos, hasta su finalización. De esa forma, comenzamos a desglosar los componentes en cuestión.

3.2. Representación Gráfica del Modelo

El Modelo EAR define claramente pasos a seguir para el análisis y diseño de la Base de Datos, dicho modelo hemos generalizado y acoplado para desarrollar el framework; de tal modo que podemos llamar a nuestro modelo EAR Espacial, con algunas diferencias dadas por el tipo de datos espaciales con los que se va a trabajar. Para el Modelo EAR Espacial establecemos diferencias en cada uno de sus etapas y una notación diferente para la representación del modelo.

Es muy importante diferenciar dos aspectos clave dados especialmente por la experiencia en el desarrollo de este tipo de bases de datos, y estos tienen que ver con la velocidad de respuesta y la proyección con la que se va a incrementar la Base de Datos; dentro del aspecto espacial y la administración que va a tener el SGDB que manipule los datos espaciales se debe considerar que debe ser el acceso lo suficientemente rápido como para que el usuario siga haciendo uso del sistema y mucho de esa velocidad depende del tamaño de las tablas a las que se va a consultar, así como del número de join que vamos a tener dentro de la consulta; de ahí que se debe ser sumamente cuidadoso con el diseño de Base de Datos Espaciales, de tal modo que si es necesario se debe subdividir tablas categorizadas por ejemplo por años para no acumular en una sola tabla demasiada información que al ser espacial tardaría demasiado tiempo en realizar una consulta y representar los datos.

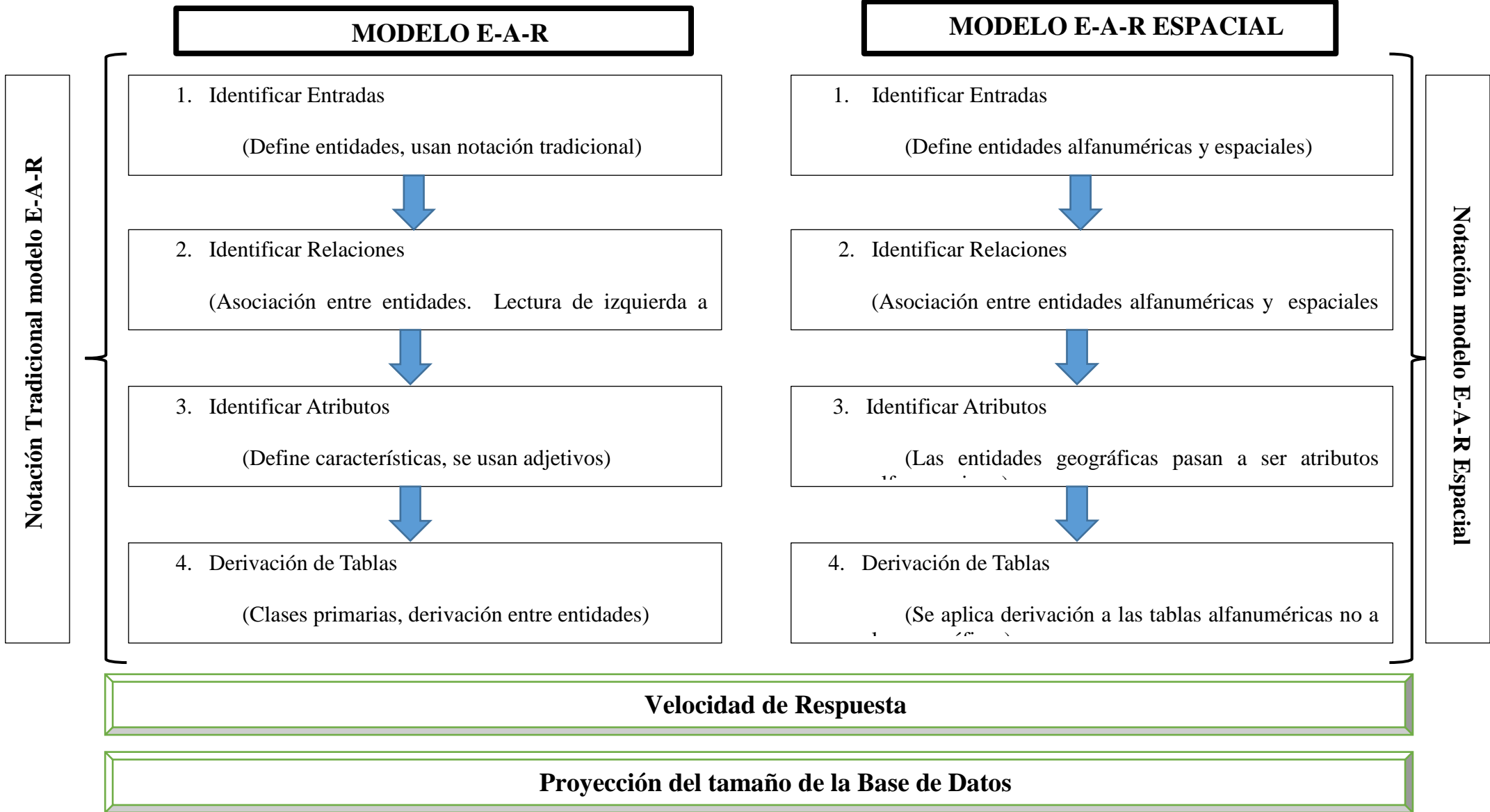


Figura 7 Representación de los Modelos

3.3. Análisis Previo

Es imposible ser dogmático sobre el punto de inicio del ciclo de diseño de bases de datos. En cualquier ejercicio de base de datos habrá un proceso de investigación previo a tomar la decisión que decidirá su construcción.

El grado de formalidad con que se llevará esta investigación variará considerablemente de un proyecto a otro. Para grandes proyectos de base de datos, probablemente se dediquen recursos considerables a la producción de justificaciones de los costes y beneficios, estudios de viabilidad, análisis de los sistemas, informes sobre las necesidades de los usuarios, etc. A veces, se utilizan los estudios piloto y los prototipos para ayudar a los directivos a tomar decisiones de una forma más escalonada.

En nuestra propia área de interés, por ejemplo, ha habido numerosos ejercicios durante los últimos años para examinar el proceso de implantación de bases de datos SIG en administraciones municipales y otras organizaciones. Es importante tener en cuenta la importancia de un buen análisis, ya que de esto depende de cómo se estructuren los pasos para la creación del proyecto de base de datos espacial, el entender las raíces del problema real y como este será llevado a un modelo computacional, no escatimar reparos en el tiempo de análisis, seguro es conveniente detenerse en el análisis hasta lograr una plena comprensión del problema que tener que realizar muchos cambios en los diseños posteriores.

Para ayudar a los analistas a estructurar sus ideas, se han desarrollado una serie de técnicas de modelamiento estudiadas en el capítulo anterior, entre ellas podemos listar las más importantes que son el Modelo Entidad – Atributo – Relación, y el modelamiento por Normalización, para el presente desarrollo del framework usamos el Modelado Entidad-Atributo-Relación (EAR), debido a la facilidad que presenta en el desarrollo de Bases de Datos y las ventajas que ofrece para acoplarse al modelo espacial.

3.4. Desarrollo Modelo E-A-R Espacial

Para el desarrollo del framework se va a utilizar el método EAR que es el

modelo utilizado para el diseño de una base de datos y lo generalizaremos a una base de datos espacial. Sus ventajas son las siguientes:

Método de arriba-abajo: la técnica funciona de lo general hasta el detalle, y ésta es una forma sencilla de proceder cuando se plantea cómo tratar un problema., se va a aplicar esta técnica analizando en primer lugar los datos alfanuméricos y posteriormente los de geoposicionamiento.

Visual: El modelado EAR consiste esencialmente en una serie de diagramas que representan la estructura de un conjunto de datos, cada uno de los tipos van a ser representados con notación especial para diferenciar lo alfanumérico de lo espacial.

Simplicidad: Aunque un análisis completo EAR puede resultar bastante exigente, el significado de los diagramas EAR simples se puede explicar de forma fácil a personas sin formación técnica, y especialmente útil para el uso de personas que no son expertas en el diseño de bases de datos; y como es el caso en el Gobierno Provincial su labor realmente es al trabajo con GIS. A veces, los diseñadores utilizan los diagramas EAR en discusiones preliminares con usuarios como un medio para comunicar ideas, bastante útil para el fin que pretendemos, lograr que profesionales de otras áreas no especializadas en software, puedan elaborar bases de datos espaciales.

Podemos secuenciar el modelado EAR en los siguientes pasos:

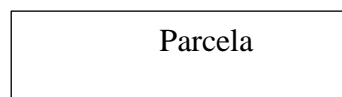


Figura 8 Secuencia de pasos AER

3.4.1. Identificar Entidades

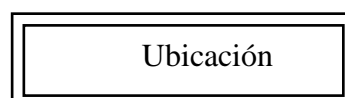
Hemos definido anteriormente las entidades como ‘objetos’ o ‘cosas’, es decir, cualquier cosa identificable con existencia independiente y sobre la cual la organización necesita recoger información. La primera acción en el modelado EAR consiste en examinar la información recogida en las investigaciones preliminares y elaborar una lista de todas las ‘entidades’ importantes. En una descripción escrita, los sustantivos que aparecen en nuestra lista son a menudo candidatos a ser entidades.

En los diagramas EAR las entidades se dibujan normalmente como cajas que contienen el nombre de la entidad. Así, si estuviéramos modelando una base de datos sobre el catastro de usuarios del sistema de riego de la Provincia de Chimborazo, podríamos tener como entidad la siguiente:



Dentro del modelo tradicional durante esta primera etapa definimos entidades candidatas y para de ellas elegir las entidades que formarán parte del diseño de la Base de Datos; mientras que en el modelo espacial, es necesario definir entidades pero necesitamos realizar una clara categorización entre entidades alfanuméricas y espaciales; refiriéndonos a las alfanuméricas a todas las entidades del modelo tradicional y las espaciales las que están dadas por ubicaciones geo referenciadas.

Como notación para entidades espaciales para efectos del framework en desarrollo vamos a usar el mismo símbolo pero con doble línea esta notación la escogemos dado que algunos autores usan esta particularidad para llamar a las entidades débiles en nuestro caso resulta ser así pues a continuación se desecha esta entidad, es decir, como entidad podría ser la siguiente:



3.4.2. Identificar Relaciones

Una relación es una asociación del mundo real entre dos objetos. Por ejemplo, un marido y una mujer están ‘casados’, una compañía ‘emplea’ a un empleado, o un Censo ‘contiene’ Distritos. En descripciones escritas, a menudo son los verbos los que dan alguna pista sobre las relaciones. En los diagramas EAR, las relaciones se representan con una línea entre las cajas que indica el tipo de relación. Por ejemplo si establecemos la relación existente entre una parcela y su propietario lo representaríamos de la siguiente manera:

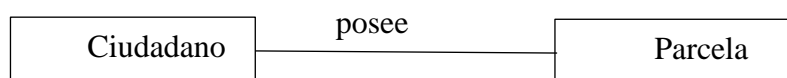


Figura 9 Secuencia de pasos AER

Puede ayudar a comprender si establecemos una buena relación entre las entidades, haciendo una lectura lógica de derecha a izquierda y viceversa:

Ciudadano es propietario de Parcela.- Una persona posee una parcela

Parcela es propiedad de Ciudadano.- El ciudadano tiene una parcela

Pueden existir muchas variantes en las lecturas, sin embargo como podemos ver debe existir una relación lógica entre las entidades a través de su relación, esto ayudará a evitar complicaciones en la decisión de establecer las relaciones entre las entidades. En el modelo espacial, las relaciones que se establecen entre entidades las vamos a leer de arriba hacia abajo en lugar de izquierda a derecha y la relación entre las dos será una flecha de doble línea. Ejemplo de lo dicho:

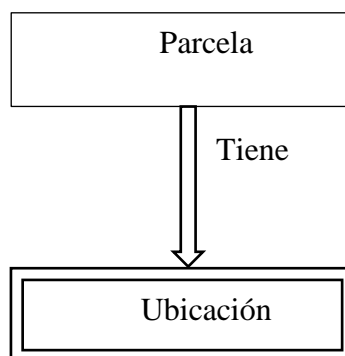


Figura 10 Entidades Espaciales

La relación que se establece entre las dos entidades, podría ser, la parcela tiene una ubicación. Cabe anotar que la relación no es en doble sentido, es decir la relación se establece de arriba hacia abajo y no viceversa. Un aspecto importante del modelado EAR es determinar el grado de relación. Aquí, grado se refiere al número de instancias (registros) que pueden participar en la relación. Las relaciones pueden ser bien uno-a-uno (1:1), uno-a-muchos (1:M) o muchos-a-muchos (M:N). Sin embargo, en el modelo espacial estas relaciones no son útiles, y simplemente se establece la que define de uno a uno; esto debido a que una ubicación geoespacial se corresponde exactamente a un solo lugar.

Uno-a-uno (1:1): Existe una relación de uno-a-uno cuando una instancia de una entidad puede tener una relación con una (y sólo una) instancia de otra; o sea, la relación es entre pares de entidades. La relación entre marido y mujer parece un buen ejemplo, un marido sólo puede tener una mujer, y viceversa. En el caso espacial solo se puede aplicar este tipo de relación.

3.4.3. Identificar Atributos de las entidades

Habiendo establecido entidades y relaciones, nos queda la tarea de asignar atributos a las entidades. En las descripciones escritas, los atributos aparecerán a menudo como adjetivos de los sustantivos. Por su parte, el modelado EAR sigue la regla de sentido común que dice que los atributos deberían asignarse a las entidades (sustantivos) a las que describen. En un diagrama, los atributos pueden mostrarse como elementos que salen de las entidades a las cuales pertenecen. Sin embargo, los diagramas EAR de grandes dimensiones pueden saturarse fácilmente si queremos mostrar todos los atributos; por ello, a menudo los diagramas EAR se dibujan mostrando sólo las entidades y las relaciones y luego se producen sub-diagramas que muestran sólo partes del diagrama principal con los atributos adjuntos.

En el diseño de una base de datos espacial, es indispensable establecer y diferenciar los atributos que son alfanuméricos y los geoespaciales, aquí radica una de las más importantes diferencias y motivos del presente framework.

A modo de ejemplo, analizando el mismo caso de la parcela, una parcela posee una extensión, ubicación, área regada, área sin riego, área productiva, forma. De los atributos enlistados anteriormente clasificamos los que son alfanuméricos, entre los que tenemos, la extensión, área regada, área sin riego, área productiva; y entre los atributos que son espaciales los podemos considerar a la ubicación. Para continuar con el desarrollo del framework, entonces todas las entidades en el apartado anterior consideradas espaciales, pasan a ser atributos de las entidades alfanuméricas, es decir, si en el apartado anterior yo conseguí definir las entidades de acuerdo a la Figura 10, entonces ahora la entidad ubicación y la relación que tiene de uno a uno, se eliminan; y pasan a formar un atributo espacial de la entidad alfanumérica; para representar los atributos espaciales usamos la notación de un ovalo con doble línea. Entonces me quedaría de la siguiente forma:

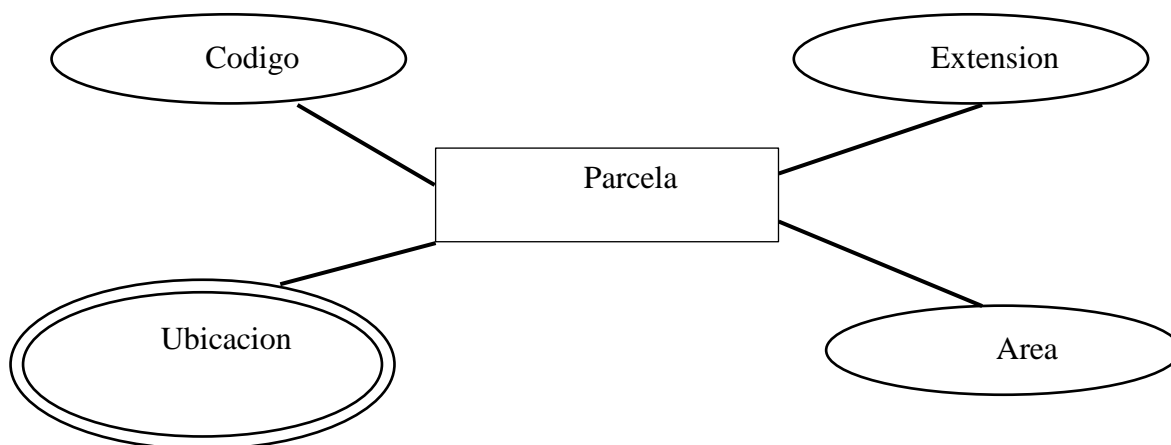


Figura 11 Relaciones Espaciales

De la derivación anterior definitivamente podemos concluir que no existen entidades espaciales, sino atributos espaciales. Las entidades espaciales pasaron a ser atributos espaciales de las entidades alfanuméricas. Los atributos espaciales pueden ser de tres tipos: punto, línea y polígono, que son los tipos de datos nativos que se usan en la georeferenciación, y que en el diseño físico de una base de datos se usa para expresar el tipo de dato que formará parte de la tabla de la base de datos.

3.4.4. Derivación de tablas

Las listas de atributos asociados con entidades son una reminiscencia de la definición de cabecera del Modelo Relacional que presentamos anteriormente.

Aunque a la larga queremos derivar definiciones tabulares de los diagramas EAR, en este punto no podemos asumir que exista una relación directa entre entidades EAR y tablas en el Modelo Relacional, dado que existen todavía algunos problemas que deberíamos resolver antes. Aquí vamos a hacer referencia a tres extensiones del modelo básico EAR Espacial:

- Claves primarias o principales
- Datos Espaciales
- Relaciones entre entidades
- Grupos repetidos

a. Claves primarias o principales.

El Modelo Relacional insiste en que cada fila de una tabla debe ser identificada de forma única mediante una clave primaria. Esto no forma parte explícitamente del proceso EAR Espacial, así que es necesario asegurarse de que dentro de los atributos asignados a cada entidad exista una clave primaria adecuada, esto quiere decir que en lo posible la clave primaria no debe ser un atributo espacial, dado el espacio de memoria usado y la dificultad de identificar dentro de la tabla dicho elemento; téngase en cuenta que mientras mayor sea el tamaño de la clave primaria que ocupa el espacio en memoria, mayor será el tiempo que tarde el SGDB tanto en indexar dicho elemento como en operar con él. En los casos en que no exista tal clave, el analista debe crear una clave ‘artificial’ en la lista de atributos. A menudo, estas claves impuestas serán simplemente una secuencia única de numeración, para efectos de nuestro framework esta recomendación es válida y sumamente útil para hacer eficaz los parámetros básicos en la operatividad de la Base de Datos.

b. Atributos Espaciales.

En el modelo que tratamos de representar debemos definir y establecer de forma clara, los atributos que se refieren a la ubicación en el espacio, los tipos de datos primitivos para el diseño de una base de datos espacial son el punto, la línea y el polígono, cada uno de ellos con capacidades especiales para ser representados en el mapa y con ello proveer un valor añadido a la entidad dentro del mundo real.

c. Relaciones entre entidades.

El Modelo Relacional utiliza atributos compartidos (claves externas) para relacionar una tabla con otra tabla. De nuevo, esto no forma parte explícitamente del modelo EAR básico y por ello debemos asegurarnos que las claves adecuadas estén en su sitio dentro del modelo conceptual para que este modelo se ajuste a los principios relacionales.

Afortunadamente, las reglas para transformar relaciones EAR a formato relacional son bastante directas:

Relaciones Uno-a-uno.- Cuando existe una relación uno-a-uno, es posible reunir las dos entidades involucradas en una sola tabla relacional y ello realmente es más notorio en diseño de base de datos espaciales. Supongamos, por ejemplo, una parcela posee una ubicación, en este caso podemos crear dos entidades con una relación uno a uno, dado que una parcela posee una sola ubicación geográfica y la ubicación geográfica hace referencia a una sola parcela. Por principio básico para el diseño de la base de datos en cuestión, las entidades geográficas se reducen en una tabla única y van como atributos dentro de la entidad a la que hace referencia, el ejemplo mencionado anteriormente quedaría de la siguiente manera:

PARCELA(IdParcela, Extensión, Area, ...)

UBICACION (IdUbicacion, Puntosxy_Referencia)

En el caso de una base de datos espacial se debe incluir ambas entidades en una tabla las entidades espaciales deberán pasar a ser atributos dentro de una entidad. Entonces en lugar de tener como en el ejemplo anterior dos entidades con el nombre PARCELA y UBICACIÓN, aplicando este principio en el modelo espacial, la tabla quedaría de la siguiente manera:

PARCELA(IdParcela, Extensión, Area, Ubicacionxy, Referencia)

Relaciones Uno-a-Muchos.- Supongamos que en una universidad cada estudiante tiene un (único) tutor personal, pero que los profesores pueden actuar como tutores personales de más de un alumno

Aquí, los atributos de los profesores pueden almacenarse en una tabla de PROFESOR:

PROFESOR (IdProfesor, Asignatura, Salario, etc.)

Esto nos permite almacenar detalles sobre los profesores aun cuando éstos no actúen como tutores personales.

Los atributos de los estudiantes pueden almacenarse en:

ESTUDIANTE (IdEstudiante, Curso, IdProfesor)

Se tiene que observar que el Número de Identificación del PROFESOR se ha incluido en esto ESTUDIANTE como clave externa. Significa que es posible identificar para cada estudiante qué profesor es su tutor personal. En términos generales, cualquier relación 1:M puede dibujarse en forma tabular añadiendo la clave principal del uno en la tabla del muchos, o sea, creando una clave externa. En el diseño de nuestra base de datos espacial, este tipo de relación es bastante útil, dado que el archivo shape que normalmente es levantado en campo por parte de los técnicos expertos en GIS lo hacen linealmente; un ejemplo de ello es en el levantamiento de la Infraestructura Vial.

En la tabla donde se encuentra la información, consta de los siguientes atributos:

Vias(tramo, longitud1, actualizado, fecha, canton....)

Hecho el análisis, un cantón puede tener varios tramos de vías, por lo tanto el diseño correcto debería ser de la siguiente manera:

Vias(tramo, longitud1, actualizado, fecha, IdCanton,....)

Canton(IdCanton, NombreCanton, UbicacionCanton)

En este caso el IdCanton en la tabla Vias es la clave secundaria de la tabla Cantón, mientras que en la tabla Cantón es la clave principal; realmente con esto lo que se está realizando es una normalización de la base de datos.

Relaciones Muchos-a-Muchos: En nuestro EAR Espacial así como en el Modelo Relacional no se puede acomodar directamente relaciones muchos-a-muchos. Sin embargo, las relaciones M:M SIEMPRE pueden ser descompuestas en dos relaciones 1:M, usando la clave primaria de cada tabla original como atributos de la tabla intermedia. Así, si volvemos a la relación Estudiante-Curso (los estudiantes toman varios cursos y los cursos tienen varios estudiantes). Las cabeceras de las tablas necesarias para implementar la relación de forma relacional son:

ESTUDIANTE (IdEstudiante, Nombre,...)

CURSO (IdCurso, Titulo,...)

MATRICULADO (IdEstudiante, IdCurso)

En este ejemplo se ilustra tanto el cambio en el diagrama EAR, que resulta de insertar una tabla intermedia, como la estructura de la tabla. La tabla intermedia MATRICULADO actúa como referencia cruzada entre las dos tablas originales. Las tablas que usamos para implementar una relación M:M siempre tendrán una clave primaria que es una combinación de las claves primarias de las dos tablas que enlazamos.

Puntualicemos dos aspectos de la descomposición de relaciones M:M. Primero, respecto al Modelo Relacional, vemos ahora que las tablas pueden usarse para representar RELACIONES así como entidades, nuestra tabla de MATRICULADO representa aquí la relación 'estudiante está matriculado a curso'. Segundo, la tabla MATRICULADO puede extenderse para albergar atributos sobre el estado de la relación entre estudiantes y cursos. En el Modelo EAR Espacial, se debe ser lo suficientemente cuidadoso de no incluir atributos espaciales dentro de las claves primarias o secundarias del modelo; dado el tiempo y peso que eso significaría a la operatividad de la Base de Datos, y a la vez ser sigilosos en cuanto al lugar de la tabla en el que ubicamos nuestros atributos espaciales, que por lo general no debería ser en la tabla que rompe la relación M:M.

d. Grupos Repetidos

Una tercera extensión al método EAR necesaria para acercar los diagramas EAR a nuestro modelo Espacial es la forma relacional es el tratamiento de los grupos repetidos. Una de las restricciones en el Modelo Relacional es que las entradas de las tablas relacionales deben ser atómicas, es decir cada celda puede albergar solo un único valor. Sin embargo, a menudo permitimos en las tablas de papel la entrada de varios valores del mismo tipo en una sola celda de la tabla. En la jerga del Modelo Relacional es un grupo repetido, y es necesario eliminarlos de los modelos conceptuales que deben presentarse de forma relacional. Pongamos un ejemplo para clarificar las cosas. Imagina que estamos creando una base de datos de las vías a partir de una tabla con la siguiente cabecera:

VIAS (tramo, longitud1 ,(CapaRodadura))

Donde los paréntesis en CapaRodadura indican la posibilidad de que una vía tenga más de una capa de rodadura, de modo que CapaRodadura representa un grupo repetido dentro de la tabla. Algunos modelos de base de datos pueden manejar este tipo de estructura sin dificultad, pero en Bases de Datos Espaciales y por ende en el modelo EAR Espacial, de las estructuras tabulares simples significa que debemos tomar medidas para eliminar los grupos repetidos.

Pero ¿quizá podamos simplemente añadir una columna extra llamada ‘segunda caparodadura’? Esto tampoco es una buena idea. Si la mayoría de las vías resultan tener sólo una capa de rodadura, habrá mucho espacio malgastado y aún peor, ¿qué ocurrirá cuando una vía llegue a tener una tercera capa de rodadura?. Quizá podamos añadir filas extra a la tabla, pero querría decir duplicar el nombre y la dirección de las familias tantas veces como coches posean y duplicar datos en una base de datos sin una buena razón activa la señal de alarma.

La única solución satisfactoria es crear una nueva tabla específicamente para albergar los grupos repetidos, de esta forma:

VIAS (Id, tramo, longitud1,....)

CAPARODADURA (Id, CapaRodadura)

Esto resuelve el problema sin crear campos null. Así pues, la regla general es que la repetición de campos se resuelve estableciendo una tabla separada para los campos repetidos. Esto se traduce en una entidad adicional y una relación 1:M en los diagramas EAR Espacial. Los pasos que hemos hecho hasta ahora no garantizan que las tablas producidas se ajusten completamente a las expectativas del Modelo Relacional. Sin embargo, hemos mostrado lo suficiente del método como para ilustrar de qué forma el modelado EAR Espacial puede ser muy útil para ayudar a los diseñadores de datos espaciales a ‘pensar cómo tratar un problema’.

3.5. Velocidad de Respuesta

Una fase crítica de las bases de datos relacionales es que son lentas comparadas con otros métodos y la culpa recae sobre el hecho de haber sacrificado la funcionalidad a cambio de la generalización y facilidad de uso de los SGBDR. Esto puede ser cierto o no, pero sí está claro que para los diseñadores de grandes bases de datos como es el caso de los SIG es una gran preocupación durante el proceso del diseño asegurarse que los SGBDR funcionan con la mayor rapidez posible, por lo que se han desarrollado una serie de técnicas de mejora de la velocidad:

- Índices
- Selección de claves
- División de tablas
- Agrupación de tablas (Desnormalización)
- Optimización SQL

3.5.1 Índices

Los índices son el método por excelencia que utilizan los diseñadores que trabajan con SGBDR para obtener tiempos de respuesta adecuados. En realidad, sin los Índices el Método Relacional no sería operativo. Los índices son tan importantes en SGBDR que merecen que les dediquemos nuestra atención. El concepto de un índice nos resulta muy familiar: usamos índices en los libros para encontrar las páginas que contienen la información que queremos, y usamos los índices de las bibliotecas para localizar los libros que queremos. Piensa en lo lento que sería el proceso de encontrar un libro en una biblioteca si no pudieras usar el sistema de

índices. Tendrías que recorrer secuencialmente todos los estantes de la biblioteca hasta dar con el libro que querías. Podemos intuir que los índices en los sistemas informáticos de bases de datos sirven para el mismo propósito que los índices en los libros o las bibliotecas. Existen para ahorrar tiempo.

En la mayoría de SGBDR se implementa físicamente una tabla “base” como un archivo adicional. Esta tabla “base” se actualiza con cada actualización de la base de datos. Si no hubiera índices y un usuario pidiera al SGBD que recuperara una fila de datos, el sistema debería leer secuencialmente el archivo hasta llegar a la localización física del registro que guarda la fila de datos. Si el archivo de este ejemplo tuviera varios miles de registros y el registro objeto de la búsqueda estuviera hacia el final, habría un retraso considerable en la respuesta. Si, en cambio, hacemos la consulta sobre un atributo para el cual se ha creado un índice, el SGBDR podrá ‘buscar’ la localización del registro pedido en el índice y dirigirse directamente al punto del archivo donde se encuentra.

Los índices son valiosos para mejorar los tiempos de respuesta cuando los usuarios quieren extraer información de una única tabla, pero son esenciales para operaciones que requieren extraer información de dos o más tablas. Tal como vimos en la técnica de normalización, es probable que un modelo de datos correctamente diseñado para una BD relacional esté formado por muchas tablas, y en consecuencia muchas de las consultas que se le pueden hacer requerirán que las tablas estén enlazadas temporalmente. Desarrollar estas operaciones de unión sin índices sería terriblemente lento. Imagina, por ejemplo, que en nuestra base de datos de las vías del Gobierno Provincial de Chimborazo quisiéramos saber qué cuál es la vía asfaltada que cruza por el cantón Colta. Para conseguir esto, sin índices, el SGBDR debería:

Buscar secuencialmente en la tabla de Tipo de Vías hasta que encuentre el tipo de vía ‘asfaltada’ en la columna TipoVia y reconozca el número deCodigoTipoVia para asfaltada. Luego Buscar secuencialmente en la tabla Vias, identificando el CodigoTipoVia hasta encontrar coincidencias del cantón Colta en la fila que contenga el valor de la extensión de la vía asfaltada. Incluso con una base de datos pequeña, esta sería una tarea extremadamente tediosa. Tan solo con índices

declarados en la columna de los atributos clave se pueden llevar a cabo con éxito preguntas de este tipo.

Los paquetes SGBDR implementan índices de formas diferentes, pero en términos generales un índice consiste físicamente en un archivo con dos columnas. Una columna contiene los valores del atributo usado para construir el índice. La segunda columna contiene los números de registro que señalan la posición física de los registros en el archivo de la tabla. De esta forma, para extraer una fila de información de una tabla utilizando un atributo indexado, un SGBDR seguiría los siguientes pasos:

Buscaría en el archivo de índice el valor de atributo especificado, posteriormente leería el valor del número de registro del registro deseado y finalmente accedería directamente al registro deseado desde el archivo de tabla. Hipotéticamente hablando los registros en la tabla Vias están en el orden en que se ingresó la información, de forma que el cantón Riobamba fue el primer registro ingresado y Colta el último. Obtener detalle de la vía que pasa por el cantón Colta significaría que deba el SGDB leer aproximadamente 200001 registros antes de encontrar el registro buscado. Sin embargo, utilizando el índice el SGDB puede rápidamente escanear el archivo índice, y detectar que el cantón Colta está ubicado en el registro 200002 e ir directamente allí. Esta notoria diferencia en velocidad al realizar una consulta, se hace mucho más evidente y crítico al manejar Sistema de Información Geográfica, debido a la magnitud de datos que tiene que consultar y mostrar.

Los índices pueden ser creados en orden ascendente o descendente de la clave índice. También pueden declararse como únicos o no únicos, siendo el primero el que no acepta entradas dobles en el atributo sobre el cual se ha creado el índice.

Si los índices son tan valiosos para incrementar la velocidad de consulta, ¿por qué los diseñadores no construyen índices para todos los atributos? Hay dos respuestas a esto. Primero, los archivos de índices, aunque pequeños comparados con los archivos de tablas convencionales, consumen espacio. Segundo, aunque los índices aceleran significativamente las operaciones de consulta, frenan las operaciones de inserción y borrado. Cada vez que se inserta o se suprime un registro de una tabla, cualquier índice relacionado con ella debe ser actualizado. Las

operaciones de inserción en tablas que tienen índices únicos pueden ser especialmente lentas, ya que para cada nuevo registro el SGBDR debe comprobar, antes de poder aceptar el registro, que la condición de exclusividad no ha sido violada para el atributo.

Por lo tanto, la decisión de dónde colocar índices requiere un equilibrio, por parte del diseñador, entre las demandas en conflicto. Afortunadamente, existen algunas reglas bastante claras que pueden usarse como guía:

- La clave primaria de cualquier tabla siempre debería estar indexada de forma única.
- Todas las claves externas deberían ser indexadas, ya que son los atributos que se usan para unir tablas.
- Un atributo que se use frecuentemente para acceder a una tabla debería ser indexado, aunque puede que no sea un campo clave. Un índice sobre un atributo que no sea clave primaria ni externa, y que sea usado para acceder directamente a la tabla, es conocido como índice secundario.

Como ya apuntábamos anteriormente, una base de datos geográfica puede llegar a tener un tamaño muy grande y realizar según qué operaciones geométricas sin un índice creado sobre la geometría puede resultar extremadamente lento. Además, las operaciones sobre campos alfanuméricos acostumbran a ser relativamente simples: comparar el valor de una celda con un valor dado, leer el valor,... Determinar si un valor es mayor que 1000, para poner un ejemplo, es bastante simple. Por el contrario, las operaciones sobre campos geométricos pueden ser mucho más complejas, determinar si un punto cae dentro de un polígono, cálculo de distancias, etc.

Repetir una operación de comparación sobre un campo alfanumérico 10 millones de veces requiere tiempo, pero hacer lo mismo sobre un campo geométrico requerirá aún mucho más. Por ejemplo, en una capa cartográfica que consistía en una malla de puntos espaciados cada 30 metros. Dichos puntos tan sólo tenían 3 campos: Identificador, Geometría y Altitud. Al tener un punto cada 30 metros, la tabla contenía alrededor de unos 38 millones de registros.

Dado un punto aleatorio en el terreno ejecutamos una consulta contra la base de datos para obtener los 4 puntos de la malla más cercanos a nuestro punto. Sin el uso de un índice implicaría:

1. Calcular la distancia entre el punto dado y cada uno de los 38 millones de puntos.
2. Ordenar ascendentemente la tabla de acuerdo con la distancia calculada.
3. Mostrar los 4 primeros registros.

Este proceso en PostGIS tardó alrededor de unos 2 minutos (113,891 milisegundos).

Si, por el contrario, creamos un índice sobre el campo geometría, el proceso que debe seguir el sistema es:

1. Buscar en la tabla índice cuáles son las posiciones (en la tabla que alberga los 38 millones de puntos) de los registros (puntos) que están geográficamente cerca de nuestro punto.
2. Calcular la distancia entre los puntos encontrados a través del índice y nuestro punto.
3. Ordenar ascendentemente los resultados según la distancia.
4. Mostrar los 4 primeros registros.

Se observa que el paso 1 limita (y lo hace enormemente) el número de puntos sobre los cuales vamos a tener que calcular las distancias. Este proceso en PostGIS tardó exactamente 31 milisegundos. En este caso, la misma consulta con índice es alrededor de 3600 veces más rápida que la misma consulta sin índice.

Un último punto sobre los índices. Aunque el diseñador de BD puede especificar dónde se crearán los índices, el software determinará por sí mismo cuándo usarlos. Todos los paquetes SGBD tienen rutinas de optimización de las consultas que determinan cómo ejecutar mejor un comando. En línea con el principio de la independencia de los datos, los usuarios no necesitan conocer la existencia de los índices.

3.5.2 Selección de claves

Como hemos visto cualquier tabla puede tener varias claves candidatas de las cuales debemos seleccionar una clave primaria. La selección de una clave primaria debería basarse en consideraciones de eficiencia. Generalmente, las características que harán de un atributo una clave eficiente son:

- Una clave corta es más atractiva que una larga (o sea, 1 mejor que 10000000), ya que reducirá el tamaño del archivo índice y además facilitará el acceso del SGBDR por experiencia, es mucho más útil crear una clave primaria auto numérica para identificar el registro.
- Las claves numéricas son preferibles a las alfanuméricas, porque los números ocupan menos espacio que los caracteres alfanuméricos y a su vez son mejores que las espaciales, pues dichos campos ocupan mayor espacio que las claves numéricas y alfanuméricas.
- Es recomendable que el atributo clave no sea volátil; cada vez que una clave primaria se cambia, significa que el SGBD tendrá que cambiar no sólo un valor en una tabla, sino en todas las otras tablas que usan la clave primaria como clave externa. Esto es un ejemplo del 'efecto cascada'.

Podemos argumentar que el mejor camino es utilizar claves primarias que no tengan significado (normalmente una secuencia única de números), ya que de esta forma no habrá incentivos para actualizarlas. En realidad, muchos SGBDR proporcionan herramientas de auto numeración para generar claves primarias ideales. En resumen, cortas, numéricas, sin significado y por ello permanentes.

3.5.3 División de tablas

Los SGBD procesan mejor las tablas pequeñas que las grandes. Si observas que dentro de una misma tabla tus usuarios/as utilizan un conjunto de atributos más a menudo que el resto, podrías considerar dividir esta tabla en dos. Hay que separar los atributos consultados asiduamente de los que rara vez se utilizan y establecer una relación uno-a-uno entre las dos tablas. Si fuera necesario, podría recuperarse la información completa de la tabla inicial con una operación de unión. De esta forma, las consultas sobre los atributos usados frecuentemente irían más rápido en esta

nueva tabla más ‘pequeña’ y de ‘uso frecuente’.

3.5.4 Agrupación de tablas (Desnormalización)

Paradójicamente, en algunas circunstancias, en lugar de dividir las tablas para obtener rapidez, podemos agruparlas para ganarla. Si encuentras que los usuarios/as hacen consultas frecuentes que combinan atributos concretos de tablas separadas, puede ser ventajoso combinar tablas para evitar tener que hacer operaciones de unión repetidamente. Por ejemplo, si en la base de datos del sistema vial encontramos que el nombre del cantón se usa casi siempre asociado con los detalles sobre las vías, podría ser aconsejable recombinar VIAS con CANTON para evitar operaciones innecesarias de unión.

Debe quedar claro que si recombinas tablas, que por otro lado las técnicas de normalización indican que deben estar separadas, estarás Desnormalizando tu base de datos. Y puedes empezar a experimentar las anomalías de actualizar, insertar y suprimir comentadas en la tercera forma normal. También aumentarás el tamaño de la base de datos al permitir la duplicación de datos. Desnormalizar una BD es un paso que debe tomarse sólo después de haberlo considerado con calma y probablemente haberlo pensado dos o incluso más veces.

e) Optimización de sentencias SQL

Como se ha explicado, para usar SOL el usuario sólo debe crear una sentencia/expresión lógica y el software de base de datos genera la respuesta. Sin embargo, los expertos en SQL creen que pueden mejorar el rendimiento especificando sentencias SQL de una forma determinada. Para hacer esto, es necesario un conocimiento detallado de los procesos internos del SGBDR, conocimiento que la mayoría de los usuarios no posee. Para operaciones importantes, los profesionales de las bases de datos analizarán qué forma de sentencia SQL alcanzará un tiempo de respuesta óptimo.

3.6. Definición del tamaño de la Base de Datos

Está claro que una de las mayores preocupaciones del diseño físico es determinar qué tamaño final puede alcanzar la base de datos y así saber qué

capacidad de almacenamiento se requerirá.

Afortunadamente, el tamaño de una BD puede estimarse de forma aproximada con un procedimiento sencillo:

- Calcular el tamaño de cada tabla: La longitud de cada fila de datos puede determinarse fácilmente agregando el número de bytes (caracteres) que se han permitido para cada atributo y después multiplicando dicha cifra por el número de registros en la tabla.
- Calcular archivos índices: El tamaño real de cada archivo de índice variará dependiendo del método de indexación usado y de la longitud del atributo clave. Como regla general, sin embargo, puedes calcular que cada índice en una tabla será un 20% de la tabla.

Suma de tablas más previsión de crecimiento: El tamaño total de la BD será la suma de todas las tablas y los archivos de índice. Después, es prudente permitir al menos otro 20% para expansiones futuras.

3.7. Conclusión del capítulo

Desarrollar el framework EAR Espacial fue la tarea específica del presente trabajo de investigación. Parte importante de este capítulo fue el análisis previo, no importa el tiempo que haga falta en hacerlo, el éxito o fracaso del framework depende en gran medida de este paso, para determinar globalmente lo que se desea lograr.

Los 4 pasos que constituyen el corazón del desarrollo del framework pueden resumirse en identificar entidades, relaciones, atributos y finalmente derivación de tablas. No debe olvidarse, sin embargo, lo esencial de la velocidad de respuesta y la definición del tamaño de crecimiento en un GIS.

CAPÍTULO IV

IMPLEMENTACIÓN DEL FRAMEWORK

4.1. Introducción.

A lo largo de este capítulo vamos a evidenciar la implementación del Framework desarrollado en el capítulo precedente; para ello usando herramientas de desarrollo Open Source, secuenciamos los pasos a seguir para el traslado práctico de un shape que contiene la información del Sistema de Infraestructura Vial de la Provincia de Chimborazo a una Base de Datos Espacial, y finalmente, la validación del Framework desarrollado, con la correspondiente comprobación de la Hipótesis.

4.2. Herramientas de Desarrollo

Las herramientas que se eligieron para el desarrollo de todo el aplicativo son Open Source, de acuerdo al Decreto 1014 emitido por parte de la Secretaría de Informática del Gobierno Central, en el que se menciona textualmente en el Artículo 1. “Establecer como política pública para las Entidades de la Administración Pública Central la utilización de Software Libre en sus sistemas y equipamientos informáticos.” (Castillo, 2010).

A continuación explicamos brevemente las herramientas utilizadas y sus ventajas:

4.2.1. Sistema Operativo

Centos (Sistema Operativo Empresarial Comunitario) es un sistema operativo basado en la estructura de Linux. El Sistema Operativo usado para el aplicativo es Centos 6.5 de 64 bits. Entre las principales características tenemos:

- Estabilidad
- Velocidad
- Confiabilidad
- Seguridad

4.2.2. Base de Datos Espacial

PostGIS es una extensión que forma parte del Motor de Base de Datos PostgreSQL en una Base de Datos Espacial. Es la solución perfecta para el almacenamiento, mantenimiento y gestión de Bases de Datos espaciales. Especialmente por lo que está construido a partir de PostgreSQL, PostGIS hereda todo la potencia de esta herramienta, así como los estándares abiertos. Entre las principales características tenemos las siguientes:

- Es software libre, posee licencia GNU General Public License (GPL)
- Está creado con compatibilidad para los estándares OGC
- Soporta tipos de datos, índices y tiene cientos de funciones espaciales.
- Importa y exporta datos con herramientas conversoras
- Tiene compatibilidad con clientes SIG de escritorio para trabajar con datos espaciales.
- Trabaja con servidores de mapas web.
- Es estable y rápido, convirtiéndose en la base de datos espacial de código abierto más usada.

4.2.3. Cliente SIG

De los clientes SIG existente en el mercado se eligió Quantum GIS que es un Sistema de información Geográfica open para plataformas GNU/Linux, Unix, Mac OS y Windows. Permite trabajar con formatos ráster y vectoriales a través de bases de datos espaciales y bibliotecas especializadas que es en realidad donde está su mayor potencial. Entre las características principales tenemos:

- Interfaz Flexible
- Interoperabilidad
- Conexión a base de datos
- Gran cantidad de complementos
- Integración con GRASS

4.2.4. Servidor de Mapas

Como Servidor Web de mapas se escogió GeoServer que tiene como funcionalidad la de servir mapas y datos de diferentes formatos para aplicaciones Web, tanto para programas GIS Desktop o clientes web ligeros. En el nivel más simple, solo hace falta un Web Browser para ver exactamente tus mapas como se desee. Adicionalmente implemente todos los estándares geo espaciales, tales como Open Geospatial Consortium (OGC), Web Feature Service (WFS) y Web Coverage Service (WCS); y está certificado como implementación de alto rendimiento de Web Map Service (WMS). Entre las principales ventajas tenemos:

- Compatible con todas las especificaciones de la OGC
- Facilidad de uso
- Soporta varios formatos de entrada
- Soporta varios formatos de salida
- Basado en servlets JAVA
- Incluye un cliente integrado OpenLayers para pre visualizar capas de datos.

4.2.5. Servidor Web

Para el aplicativo siguiendo la línea de Open Source se desarrolló con el Apache Web Server, de código abierto, la mayoría de sus extensiones son personalizadas y están disponibles de forma gratuita. Entre las principales ventajas citamos las siguientes:

- Costo
- Funcionalidades
- Soporte
- Portabilidad

4.3. Implementación del framework

Siguiendo la secuencia de pasos que el modelo EAR Espacial desarrollado en el capítulo anterior establece, pasamos a la ejecución:

4.3.1. Análisis Previo

Analizamos el archivo shape formado por los técnicos de la Unidad de Planificación, para el Sistema de Infraestructura Vial de la Provincia de Chimborazo, dicho archivo tiene la extensión shp y es el utilizado por los técnicos para todo el trabajo y la información topográfica. El archivo tiene la siguiente estructura.

TRAMO	LONGITUD_1	ACTUALIZAD	FECHA	AÑO	CANTON	OBRA_ARTE	CAPA_RODAD	CALIF_CAPA	LISTO	ORDEN	TIPO
TOTORAS-OZOG...	10.380000000	CRISTIAN PALAC...	NULL	2012	ALAUSSI	NULL	LASTRADO	MALO	1	4 ORDEN	NULL
TOTORAS-OZOG...	9.680151455	CRISTIAN PALAC...	NULL	2012	ALAUSSI	NULL	LASTRADO	MALO	1	4 ORDEN	NULL
GUAMOTE MATRI...	4.015002466	IVAN ROJAS	NULL	2012	GUAMOTE	NULL	TIERRA	MALO	1	4 ORDEN	NULL
ZANJAPAMBA-VI...	3.134796162	EDGAR DAMIAN	NULL	2012	GUANO	NULL	TIERRA	MALO	1	4 ORDEN	NULL
VIA GARCÍA MO...	0.820790228	EDGAR DAMIAN	NULL	2012	GUANO	NULL	TIERRA	MALO	1	4 ORDEN	NULL
SILVERIA - VIA C...	2.231498441	EDGAR DAMIAN	NULL	2012	GUANO	NULL	TIERRA	MALO	1	4 ORDEN	NULL
PANAMERICANA ...	1.409625301	IVAN ROJAS	NULL	2012	RIOBAMBA	ALCANTARITLAD...	ASFALTO	BUENO	1	1 ORDEN	PANAMERICANA
LINLIN COLEGI...	0.503575627	FERNANDO ZAB...	NULL	2012	COLTA	NULL	TIERRA	REGULAR	1	4 ORDEN	NULL
INTERVENID	COD_TRAMO	COD_VIA	EJES	CUNETA	ALCANTARIL	PUENTE	PARROQUIA	OBSERVACIO	X	Y	re_class
NULL	1428	NULL	0	NO	NO	NO	ACHUPALLAS	NULL	759419.1167790...	9751382.247889...	Suelto/No pavim...
NULL	1	NULL	0	NO	NO	NO	ACHUPALLAS	NULL	758778.4112279...	9751787.802239...	Suelto/No pavim...
NULL	2	NULL	0	NO	NO	NO	GUAMOTE	NULL	753223.8876920...	9785698.864490...	Suelto/No pavim...
NULL	3	NULL	0	NO	NO	NO	SAN JUAN	NULL	747385.1061119...	9827534.980949...	Suelto/No pavim...
NULL	4	NULL	0	NO	NO	NO	SAN ANDRES	NULL	749847.3352159...	9828520.091930...	Suelto/No pavim...
NULL	5	NULL	0	NO	NO	NO	SAN ANDRES	NULL	749955.4883270...	9829598.222020...	Suelto/No pavim...
NULL	7	NULL	1	SI	SI	SI	SANTIAGO DE Q...	NULL	751679.8965799...	9801577.612150...	Duro/Pavimento
NULL	8	NULL	1	NO	NO	NO	COLUMBE	NULL	749367.3452469...	9793552.369869...	Suelto/No pavim...

Figura 12 Estructura Archivo Shape

Inicialmente en el archivo shape formado, vamos a hacer un análisis de las columnas ahí especificadas, de acuerdo a conversaciones mantenidas con las personas que lo crearon y llenaron la información. En primer lugar tenemos una columna que se llama TRAMO, esta columna especifica el lugar en el que inicia y el lugar en el que termina la vía que se ha levantado de acuerdo a algún lugar de referencia este puede ser comunidad, parroquia, cantón o vía. La información de este campo no es tan exacta, más bien sirve de información adicional.

El siguiente campo se denomina LONGITUD_1, este campo está dado en metros y establece la medida del tramo geo referenciado. A continuación existe el campo con el nombre de ACTUALIZAD, en esta columna del shape se encuentra el nombre del técnico que hizo el levantamiento de la información o que lo actualizó, es un campo tipo varchar sin ningún tipo de referencia. El siguiente campo es la fecha en la que se levantó la información determinada por el día y el mes separado por el guion (-), no todas las filas tiene fecha.

A continuación se encuentra la columna del año en el que se levantó la información o se procedió a actualizar. En este campo que está dado teóricamente

por cuatro caracteres que deberían ser el que determine el año, existen datos incoherentes, tales como que el ingreso se ha hecho con el valor 21012 o con 2103; suponemos que es un error en la digitación y corresponden al año 2012 y 2013 respectivamente. El siguiente campo es el CANTON, aquí se ha ingresado información correspondiente al cantón por el que pasa la vía en su mayoría de extensión, como referencia para luego las búsquedas. Entre los valores que ha tomado esta columna están: Alausí, Chambo, Chunchi, Colta, Cumandá, Guamote, Guano, Riobamba, Pallatanga, Penipe, y existen celdas vacías.

Luego tenemos el campo OBRA_ARTE, entre los posibles valores que puede tomar este campo están: alcantarillado, cuneta, puente y una variación de éstas. Sin embargo, revisados los datos debido a que se lo ingresaba de forma manual, existen datos erróneos, tal es el caso que en lugar de estar alcantarillado, se ha ingresado como alcantarillado solo por poner un ejemplo. Posteriormente tenemos el campo CAPA_RODAD, que se refiere al tipo de capa de rodadura de la vía, en el cual se identifican los siguientes tipos: lastrado, tierra, asfalto, empedrado, hormigón, adoquinado. En el siguiente se encuentra una evaluación a la capa de rodadura del campo anterior, la valoración se lo hace entre bueno, regular, malo y algunas celdas vacías.

El campo que le sigue se denomina LISTO, y tiene los posibles valores de 0, 1 y vacío. El siguiente determina el tipo de vía, categorizado por vías de 1 orden, 3 orden y 4 orden ubicados en la columna denominada ORDEN. El siguiente campo se denomina TIPO y simplemente tiene el valor de panamericana o vacío. A continuación tenemos un campo que se llama INTERVENID en donde los valores son intervenid* y vacío. En la columna subsiguiente se encuentra el COD_TRAMO, este campo identifica cada uno de los tramos levantados, aunque tiene el problema que los números no son únicos sino que existen repetidos.

Se ha creado un campo denominado EJES especialmente para el tratamiento de la información de manera interna dentro del Departamento de Planificación, su valor es 0 por defecto, a futuro piensa cambiarse de acuerdo a las necesidades. A continuación tenemos el campo CUNETAS con dos posibles valores SI, y NO; con lo

cual se determina si existe o no cuneta en el tramo de vía levantado. El mismo principio se aplica al siguiente campo que se denomina PUENTE.

Posteriormente existe un campo que se denomina PARROQUIA, bastante útil para la administración de la información, pues aquí se llena la parroquia en donde se encuentra el tramo de vía levantado. Posteriormente tenemos OBSERVACION, campo puesto para las observaciones que se vayan a agregar en el tramo de vía levantado. En el final del shape, tenemos los tipos de datos geoespaciales, se encuentra en un campo el punto X y el punto Y del inicio del tramo de la vía. Y finalmente, el campo RE_CLASS en donde se puede tener los valores Suelto/No Pavimentado, y Duro/Pavimento.

4.3.2. Data Cleansing

Hecho el análisis previo en la Base de Datos claramente determinamos en primer lugar, antes de continuar con el proceso del framework, la limpieza de datos. Descubrimos, corregimos y eliminamos datos erróneos existentes en nuestro shape, tomando en cuenta que al ser un archivo geo, son bastantes los datos que son inconsistentes.

Vamos a identificar datos incompletos, incorrectos, inexactos, no pertinentes, etc. y substituirlos o modificarlos todos los datos sucios encontrados. Posterior a la limpieza consideramos que los datos y la base podrían ser considerados para continuar con el desarrollo del framework. Corregimos errores sintácticos que luego del análisis son los que se han presentado en el archivo shape.

Iniciando el análisis desde la izquierda hacía la derecha en la columna ACTUALIZAD se procedieron a limpiar los registros 2705 y 2706 en donde se presentan la fecha en un formato incluido el año, que no es consistente con el resto de registros; por lo tanto quitamos el año 2013 y dejamos en el mismo formato que el resto de registros. En el registro 2701 existe el año ingresado como 21012, obviamente es un error de sintaxis, por lo tanto cambiaron a 2012.

Enseguida, procedemos a revisar los datos en la columna OBRA_ARTE, en donde existen problemas de inconsistencia sintáctica al escribir de forma errónea los

mismos datos; concluimos aquello dado que tenemos registros como ALCANTARILLADO y otros como ALACANTARILLADO, lo mismo pasa en ALCANTARILLADO CUNETA y otros como ALACANTARILLADO CUNETA y otros cuantos como ALCANTARILLADO – CUNETA, y otros como ALCANTARIILADO - CUNETA; en donde obviamente los registros deberían tener el mismo dato. Por comodidad y facilidad toda la limpieza de datos se lo hizo desde el motor de base de datos con sentencias de actualización desde postgresql.

4.3.3. Identificar Entidades

Posterior al análisis minucioso de los datos y luego del proceso de data cleansing aplicamos el primer paso formal del framework, analizamos y determinamos las siguientes entidades tanto Alfanuméricas como Espaciales.

Entidades Alfanuméricas

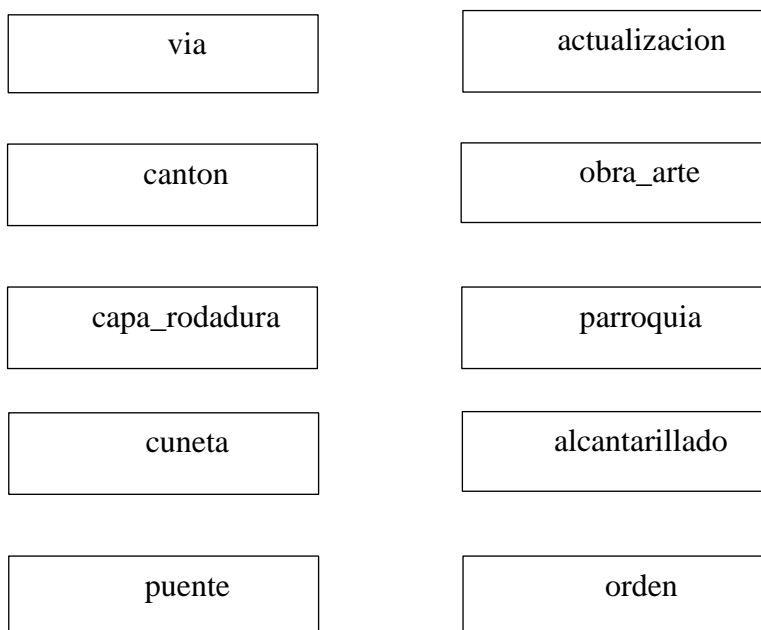


Figura 13 Entidades Alfanumérica

Entidades Espaciales

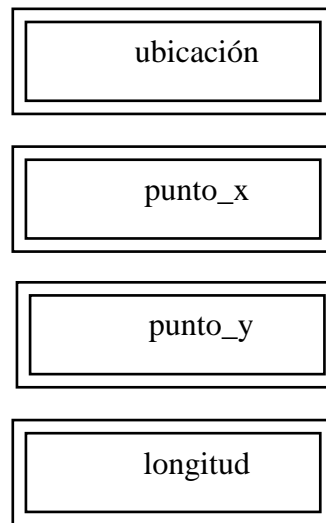


Figura 14 Entidades Espaciales

4.3.4. Identificar Relaciones

Ahora establecemos las relaciones existentes entre cada una de las entidades, tanto alfanuméricas y espaciales. Si nos fijamos para nuestro desarrollo de la Base de Datos Espacial necesitamos tener muy claro cuál será la estructura que se va a manejar así como las relaciones existentes entre cada uno de las entidades. Desarrollado las relaciones con las entidades existentes el diseño nos quedaría de la siguiente manera:

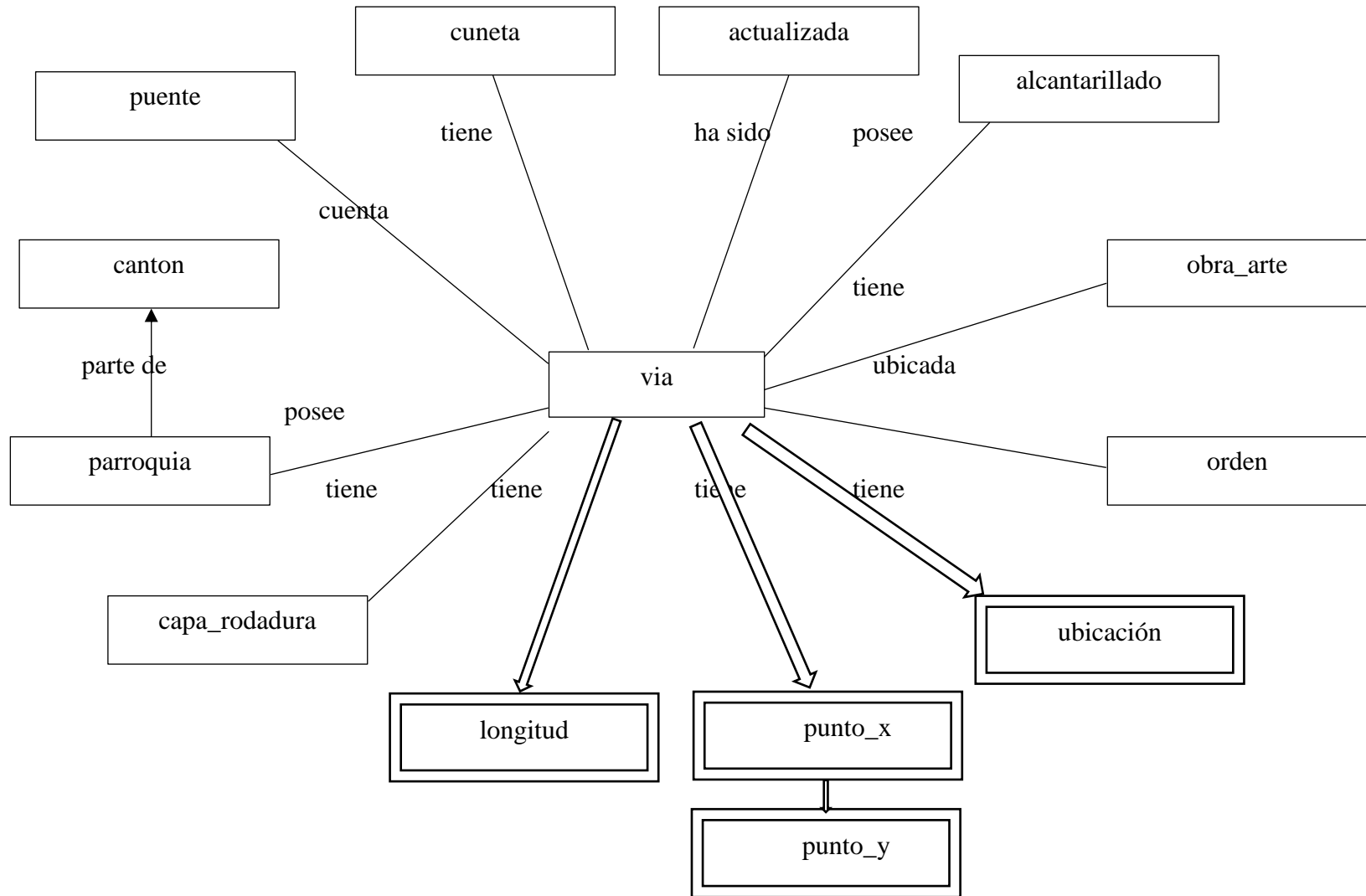


Figura 15 Relaciones Identificadas

4.3.5. Identificar Atributos

a) Entidad vía

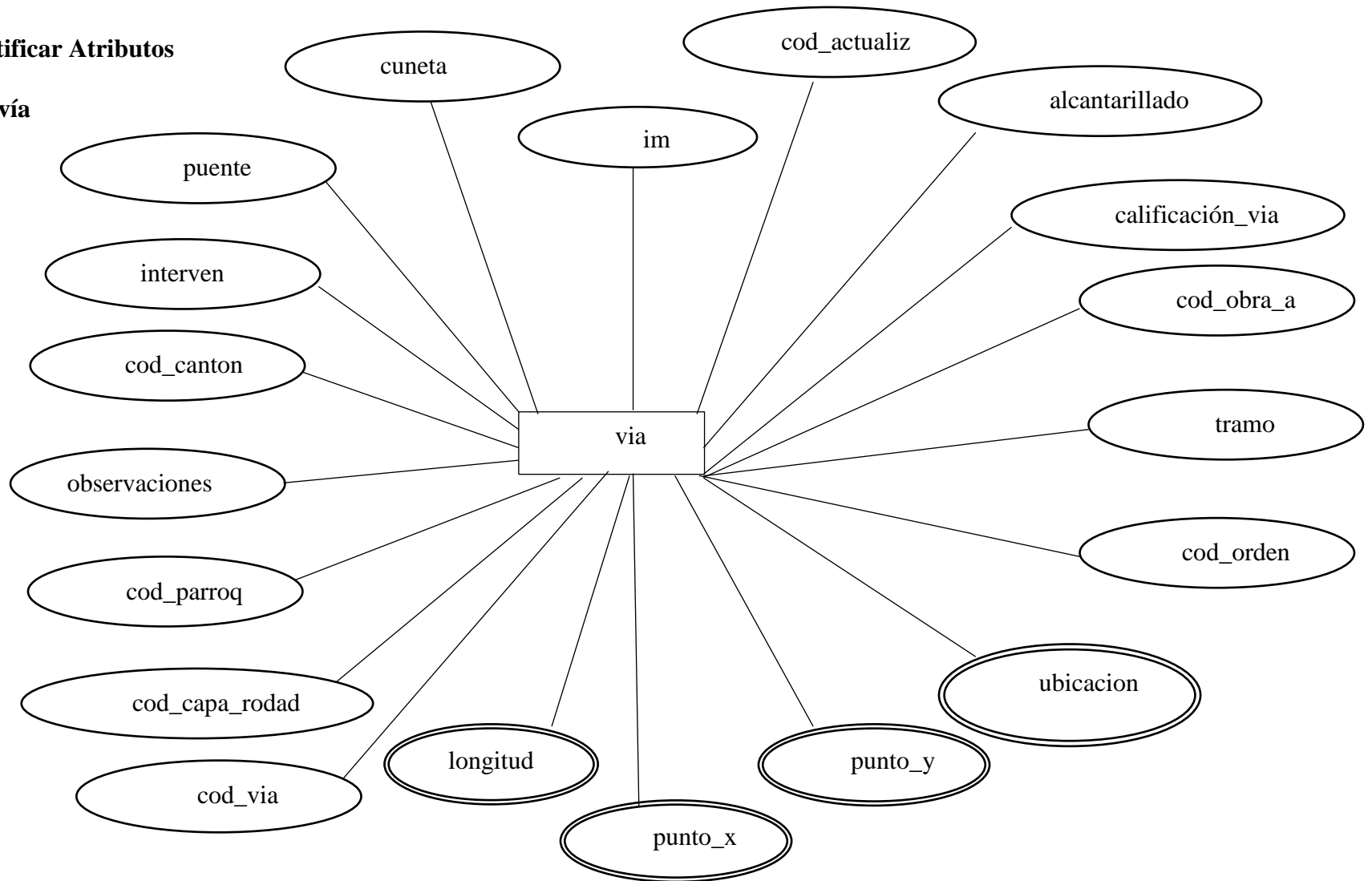
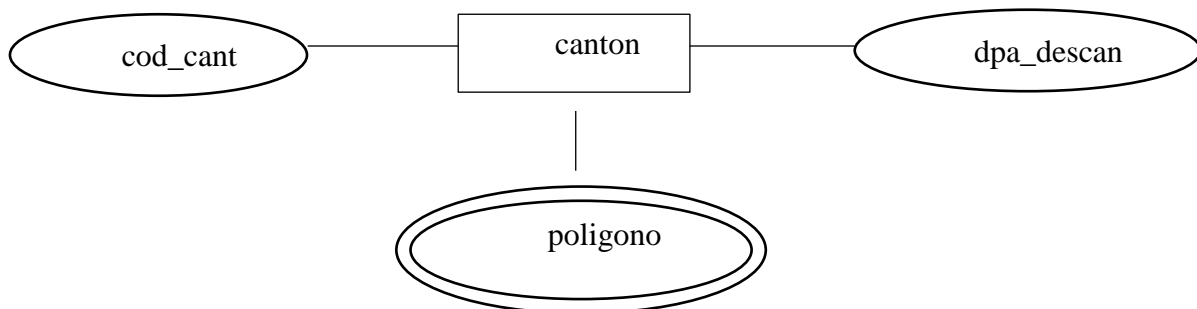
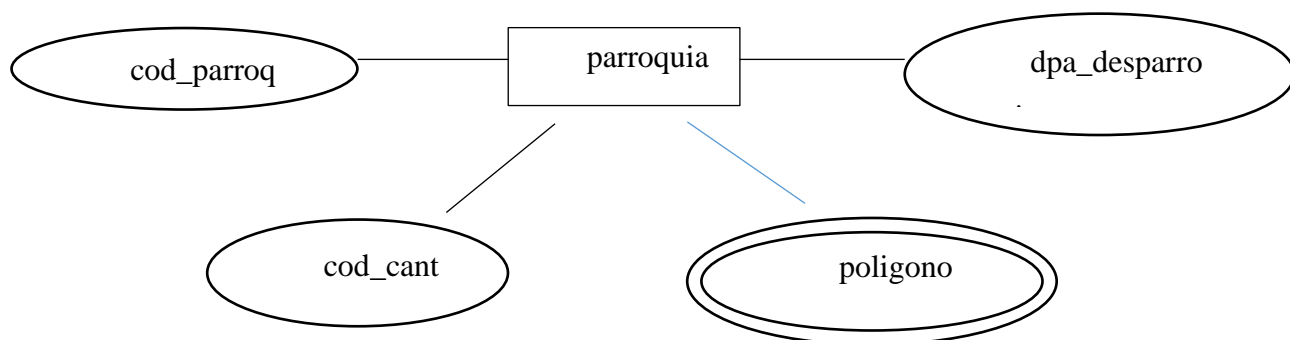
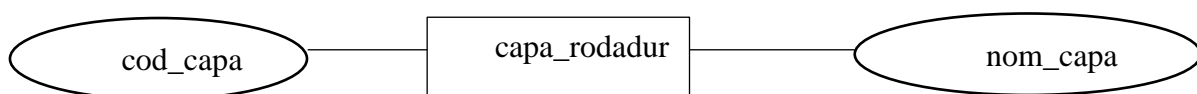
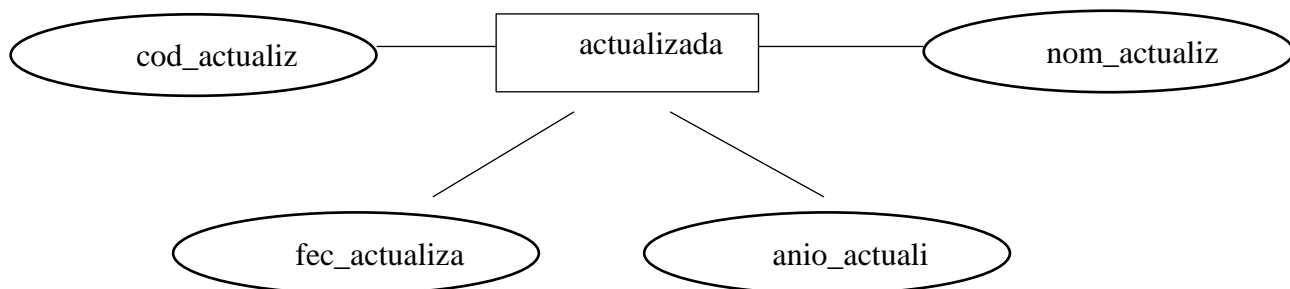
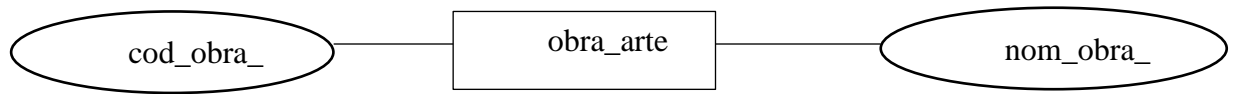


Figura 16 Atributos Identificados

b) Entidad cantón**c) Entidad parroquia****d) Entidad capa_rodadura****e) Entidad actualizada**

f) Entidad obra_arte**g) Entidad orden****4.3.6. Derivación de Tablas**

En el presente apartado lo que hacemos es definir claramente las claves primarias y las claves secundarias en el diseño; así como establecer claramente las entidades espaciales dentro de nuestro diseño.

Por notación vamos a agregar al inicio del atributo cuando sea clave primaria las letras PK y cuando sea una clave secundaria FK; los atributos espaciales no hacen falta una notación especial, dado que las mismas ya se definieron anteriormente con una línea doble en el óvalo que la presenta.

Entonces el diseño quedaría de la siguiente manera:

a) Entidad vía

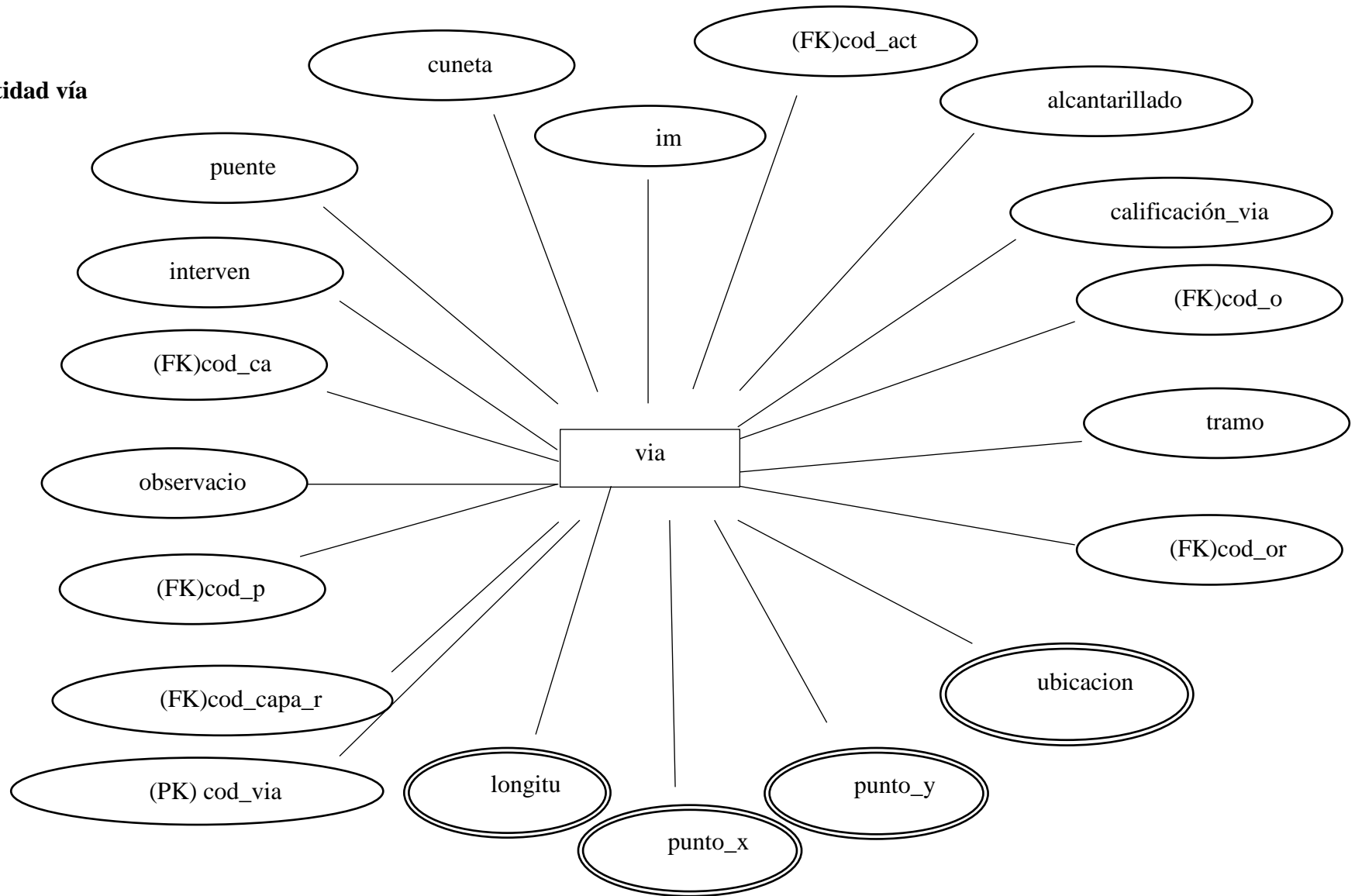
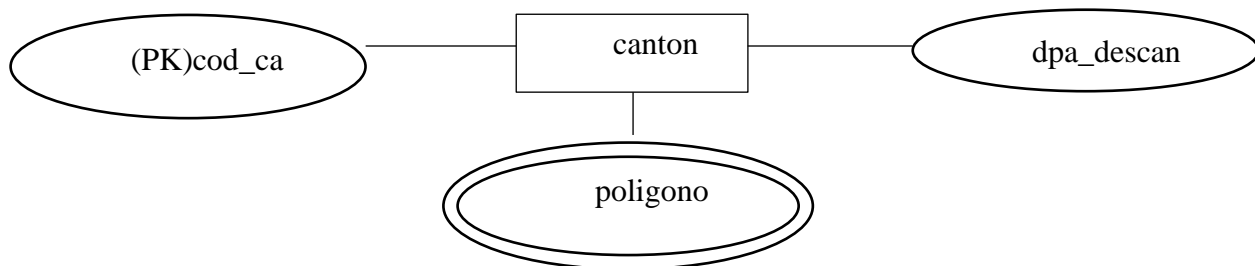
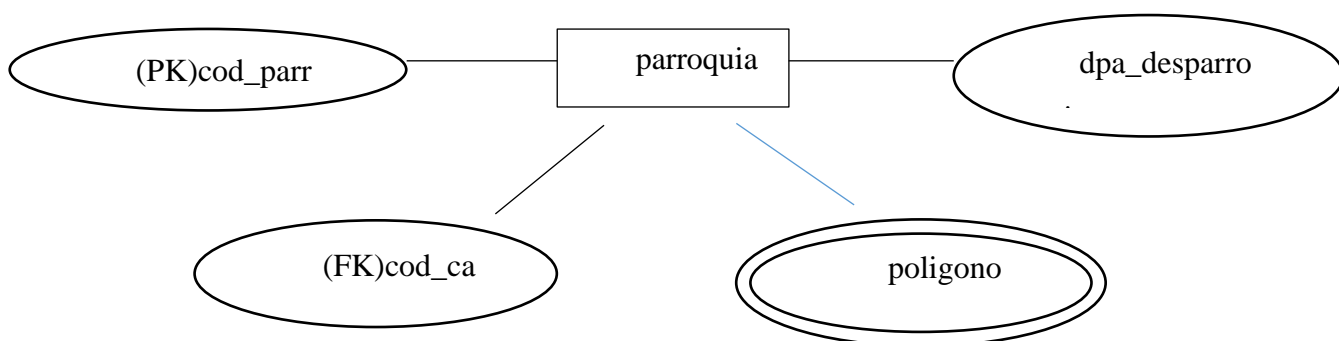
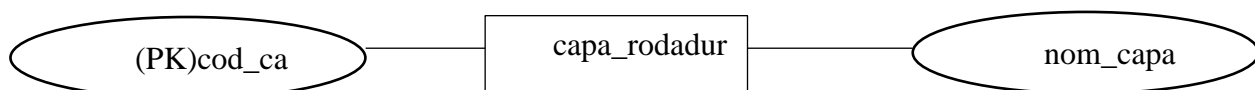
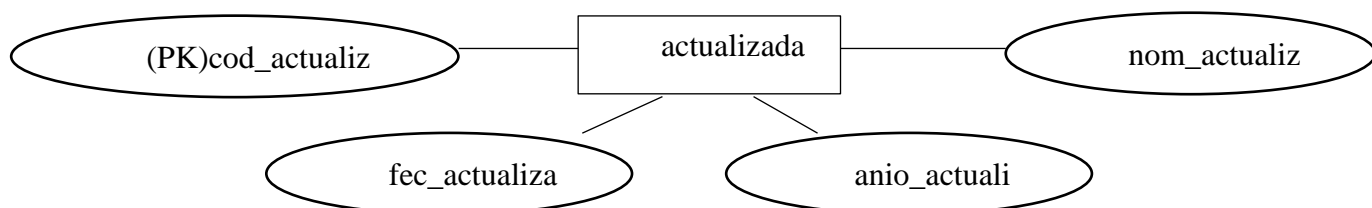
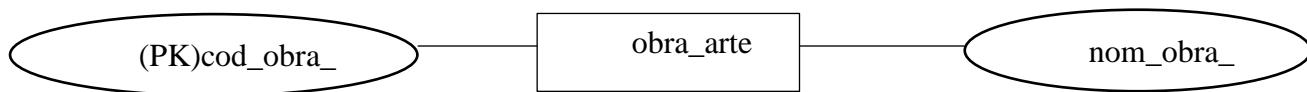
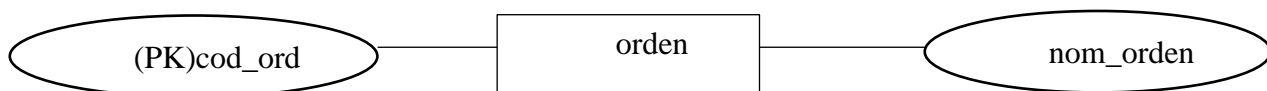


Figura 17 Derivación de Tablas

b) Entidad cantón**Figura 18 Entidad cantón derivada****c) Entidad parroquia****Figura 19 Entidad Parroquia derivada****d) Entidad capa_rodadura****Figura 20 Entidad capa_rodadura derivada****e) Entidad actualizada****Figura 21 Entidad actualizada derivada**

f) Entidad obra_arte**Figura 22 Entidad obra_arte derivada****g) Entidad orden****Figura 23 Entidad orden derivada****4.4. Desarrollo del Aplicativo.**

El aplicativo que se ha desarrollado utiliza librerías propias de php para la representación de los mapas, con la herramienta geoserver como servidor de mapas. Todas las herramientas usadas son open source, tratando de ser lo más amigable posible en su interfaz, se usa estilos css, con interface potenciada con javascript.

4.4.1. Explicación

El aplicativo que se denomina SIP (Sistema de Infraestructura Provincial de Chimborazo), en su primera fase presenta los datos subidos que se refieren al Sistema de Infraestructura Vial de la Provincia, En el anexo 1 se presenta la interface y un manual básico del uso del sistema.

4.4.2. Conexiones

Para lograr llevar desde el shape a la base de datos postgresql se utilizaron algunas librerías nativas de php, así como nativas de postgresql. La conexión con el módulo de postgis, requiere cierta configuración, que en nuestro caso se estableció un usuario postgres, la clave como root, el servidor localhost, el puerto 5432 y la base de datos en postgis con el nombre de framework.

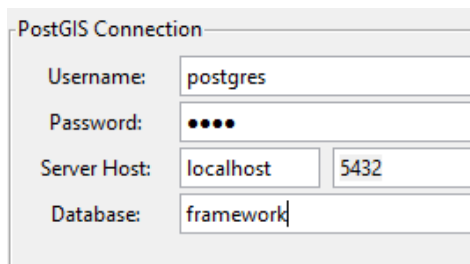


Figura 24 Conexión Postgis

El código para generar la librería shp2pgsql desde php se especifica en el siguiente código usado.

```

1 <?php
2 $path_shape="d:\\";
3 $path_salida="d:\\";
4 shell_exec("C:\Program Files\PostgreSQL\9.4\bin\postgis\shp2pgsql-gui.exe");
5 shell_exec("C:\Program Files\PostgreSQL\9.4\bin\shp2pgsql.exe -I -s 4326 d:\DICE_DPA2011_chimborazo_POR_PARROQUIAS.shp parroquias > d:\parroquias.sql");
6 shell_exec("C:\Program Files\PostgreSQL\9.4\bin\shp2pgsql.exe -I -s 4326 d:\GEO_DPA_CHIMBORAZO_POR_CANTONES_2010.shp cantones > d:\cantones.sql");
7 shell_exec("C:\Program Files\PostgreSQL\9.4\bin\shp2pgsql.exe -I -s 4326 d:\vias_50000_GADPCH_2012.shp vias > d:\vias.sql");
8 shell_exec("C:\Program Files\PostgreSQL\9.4\bin\shp2pgsql.exe -I -s 4326 d:\DICE_DPA2011_chimborazo_POR_PARROQUIAS.shp gis.parroquias | psql -U postgres -d framework");
9 shell_exec("C:\Program Files\PostgreSQL\9.4\bin\shp2pgsql.exe -I -s 4326 d:\GEO_DPA_CHIMBORAZO_POR_CANTONES_2010.shp gis.cantones | psql -U postgres -d framework");
10 shell_exec("C:\Program Files\PostgreSQL\9.4\bin\shp2pgsql.exe -I -s 4326 d:\vias_50000_GADPCH_2012.shp gis.vias | psql -U postgres -d framework");
11
12 ?>

```

Figura 25 Líneas de Código

En estas líneas de código se resume la interfaz para el llamado de la librería usada para la transformación de los shape a postgis, no siendo la única es una de las más útiles por ser nativa de postgresql. Se usa una sentencia shell_exec con el fin de llamar a una ejecutable, posteriormente se llamada desde el ejecutable de shp2pgsql se establecen los parámetros a usarse en primer lugar va el parámetro -I con el fin de que se cree un índice espacial de la columna de geometrías tanto de las parroquias, como de los cantones y de las vías, que son las tres entidades en donde se tienen datos geoespaciales.

Enseguida se especifica -s para indicar el sistema de referencias de coordenadas en el que está expresada la capa que se usa, que en nuestro caso es el 4326, dado que el sistema de coordenadas está basado en WGS 84. Si hiciera falta también en este

lugar se agrega el modo de codificación con –W, para indicarle si es UTF-8 o Latin1 o cualquiera que sea el método que se use.

4.5 VALIDACION DEL FRAMEWORK

Para validar el framework desarrollado se han utilizado instrumentos útiles con tal fin. Dicha validación se sustenta con métodos evaluativos cuantificables y de investigación, que son los siguientes:

- Comparativa referencial.
- Investigación de campo.
- Tabulación de resultados.
- Discusión de los resultados.

4.5.1. Comparativa referencial.

El framework desarrollado es aplicable en una etapa de producción, por lo que un procedimiento de mediciones se puede realizar comparando los diferentes indicadores establecidos para la comprobación de la hipótesis que van a ser evaluados sin el uso del framework cuando se usa exclusivamente archivos shape y el contraste con el uso del framework en una Base de Datos Espacial.

Establecimiento de métricas

Las medidas que establezcamos para comprobar que se han ejecutado los indicadores y que realmente sean capaces de definir el comportamiento exitoso de lo que queremos efectivamente comprobar debería satisfacer los siguientes principios:

- Objetivas
- Sencillas
- Obtenibles
- Válidas y
- Robustas

El framework requiere mediciones sobre el producto terminado, y para ello vamos a usar la Base de Datos Espacial, que es el resultado de aplicar el marco de

trabajo desarrollado, por lo que seleccionamos métricas para establecer los indicadores claramente definidos en el inicio de la investigación.

Entre los tipos de medidas establecidas para un producto Software o de almacenamiento, escogemos solo aquellas que realmente son prácticas y que cumplen las características mencionadas en una métrica. Por lo tanto vamos a realizar medidas que nos proporcionan cuantitativamente resultados para compararlos y posteriormente los cualitativos con el fin de concluir en la comprobación de la hipótesis planteada. Las métricas seleccionadas son las siguientes.

a.) Almacenamiento en disco.

Vamos a usar como métrica el tamaño que se usa para el almacenamiento en disco, en una comparativa basada en el archivo shape, y el valor que ocupa la Base de Datos Espacial. Con el fin de tomar referencias reales revisamos el tamaño que ocupa de espacio en disco los archivo shape antes de cualquier tipo de procesamiento y los comparamos con el tamaño de espacio que ocupa luego de aplicado el framework en la Base de Datos Espacial, a continuación los resultados.

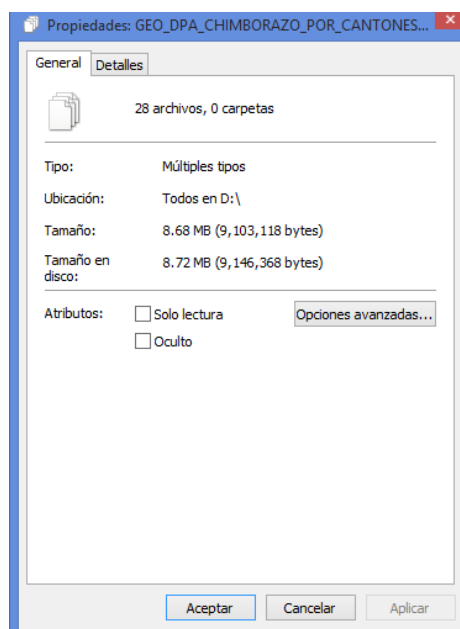
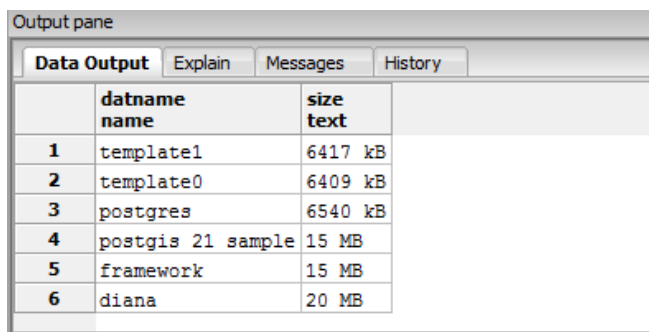


Figura 26 Tamaño shape

Con el comando: `SELECT pg_database.datname, pg_size_pretty(pg_database_size(pg_database.datname)) AS SIZE FROM pg_database;` revisamos el tamaño real que ocupan las bases de datos en postgresql de donde se obtuvo el siguiente resultado.



	datname	size
1	template1	6417 kB
2	template0	6409 kB
3	postgres	6540 kB
4	postgis 21 sample	15 MB
5	framework	15 MB
6	diana	20 MB

Figura 27 Tamaño B.D. Postgis

Tabla 4

Comparativa shape vs B.D.

Archivos Shape	Base de Datos Postgresql
8.68 MB	15MB
10 Usuarios	1 centralizado
86.8 MB	15 MB

De la tabla anterior el archivo shape ocupa un espacio en almacenamiento menor casi en un 50% al de la base de datos después de aplicar el framework; sin embargo, se debe considerar que el archivo se distribuye para al menos 10 técnicos expertos en el Departamento de Planificación, por lo tanto el resultado real que ocupa el archivo shape sería de:

$$8.68 \text{ MB} * 10 (\text{usuarios}) = 86.8 \text{ MB}$$

Que si comparamos los 86.8 MB a los 15 MB que ocupa la base de datos espacial, el ahorro del almacenamiento en disco es realmente exponencial.

b.) Tiempo de respuesta en búsquedas.

Aunque en la práctica están varios factores ligados al resultado tiempo que tarda una sentencia sql en ejecutarse, tales como variables que tienen que ver con el diseño de la base de datos, velocidad de procesamiento, tamaño de la tabla. Sin embargo, hemos tomado como referente una consulta estándar a uno o dos tablas indexadas para una comparativa sencilla entre el tiempo que se demora una consulta sql y el tiempo que se demora la misma consulta en un estimativo al hacerlo en el archivo shape o en el Excel generado por la herramienta o dentro del SIG Cliente.

Aplicamos una serie de 10 búsquedas vía consultas sql para el cálculo de nuestra tabla comparativa, las consultas que se hicieron se elaboraron de acuerdo al criterio técnico de un experto en GIS del Departamento de Planificación en el que se escogió los parámetros más comunes que se necesitan para revisar la información. Además de lo expuesto, estimamos que de acuerdo al número de registros existentes se tiene que tardar en buscar cada registro un estimativo de 2 mili segundos por cada tupla, por lo tanto por cada dato que busquemos en el shape determinamos los mili segundos que nos demoramos en mostrarlo. Las consultas hechas se encuentran en el anexo 2. De lo expuesto se obtuvo los siguientes resultados:

Tabla 5

Consultas

Consulta	T. postgresql	T. shape
Por cantón Riobamba	820 ms	735 * 2 = 1470
Por cantón Alausí	647 ms	329 * 2 = 658
Por cantón Chambo	142 ms	137 * 2 = 274
Por capa de rodadura Lastrado	937 ms	854 * 2 = 1708
Por capa de rodadura Tierra	773 ms	1049 * 2 = 2098
Por capa de rodadura Asfalto	297 ms	586 * 2 = 1172
Por calificación de capa Bueno	438 ms	757 * 2 = 1514
Por calificación de capa Regular	657 ms	854 * 2 = 1708
Por calificación de capa Malo	859 ms	1092 * 2 = 2184
Por orden 4 orden	1459 ms	1839 * 2 = 3678
Por orden 3 orden	430 ms	691 * 2 = 1382
Por parroquia Columbe	94 ms	145 * 2 = 290
Por parroquia San Andrés	125 ms	177 * 2 = 354
Por parroquia Achupallas	78 ms	42 * 2 = 84
Por longitud <5	1460 ms	2537*2= 5074
Por longitud >=5	500 ms	169 * 2 = 338

c.) Tiempo de generación de informes.

La importancia para generar informes o reportes de la información, es en sí realmente lo más crítico desde el punto de vista del usuario; para él debería ser transparente el trabajo que el desarrollador o expertos en GIS hace por detrás para conseguir un resultado, y dado la orientación gerencial que tiene el sistema propuesto, hemos considerado de vital consideración, el cálculo del tiempo que toma emitir un reporte desde el sistema, en una comparativa con el tiempo que toma generarlo manualmente, ya sea de la parte geográfica como de la alfanumérica. Basados en el mismo tiempo que tarda una sentencia sql para generar el informe, podemos determinar el mismo tiempo adscrito en la tabla anterior es la misma al generar los informes correspondientes, de la misma manera con el tiempo que demora en generar el reporte en el shape.

d.) Acceso masivo a la información

El acceso masivo a la información, se lo puede brindar mediante la publicación de los resultados en la web; basados en el número de visitas que tiene la página web institucional por día, podríamos concluir que ahora la información no es un privilegio de un grupo reducido de técnicos o personas, sino tiene un tinte público y masivo. El acceso a la página web institucional se da de acuerdo al registro de visitas como se muestra en la siguiente figura:



Figura 28 Contador de Visitas

4.5.2. Investigación de campo.

Finalmente vamos a realizar un análisis no solo cuantitativo sino también cualitativo orientado a la definición de los resultados de las métricas que se han

establecido. Alineado con el método científico, la investigación de campo compone las prácticas y procedimientos para el levantamiento de conocimientos con el fin de determinar las realidades, necesidades, dificultades y practicidad de la investigación aplicada. Para ello usamos una encuesta online que se publicó dentro de la Intranet Institucional orientada a medir los resultados obtenidos con el desarrollo de la Base de Datos Espacial usando el framework, esta encuesta orientada al personal experto, y también a los que hacen uso del sistema en base a los indicadores establecidos en la variable dependiente.

Población y Muestra

Dada la naturaleza de la investigación, y tomando en cuenta que el framework beneficia realmente a las personas expertas en el Área de Planificación; debido al número de usuarios directos del framework se ha tomado una muestra no probabilística resultado de la fórmula para población finita, basados en un margen del 0,05% de error para 30 empleados que son los que van a responder la encuesta, y son las personas del Departamento de Planificación del Gobierno Provincial de Chimborazo.

Cálculo de la Muestra.

$$n = \frac{Z^2 P Q N}{(N - 1) E^2 + Z^2 P Q}$$

En donde

n= Tamaño de muestra

Z= Valor Z curva normal (1.96) 95%

P=Probabilidad de éxito (0.5)

Q=Probabilidad de fracaso (0.5)

N=Población (30)

E=Error muestral (0.05)

Nos quedaría:

$$n = \frac{1.96^2(0.5)(0.5)(30)}{(29)(0.05)^2 + 1.96^2(0.5)(0.5)}$$

N=28

4.5.3. Indicadores.

Son conceptos que se puede medir o cuantificables en el ámbito de la demostración de la hipótesis, para estimar el cumplimiento de nuestra tesis. Los indicadores considerados son el espacio en disco, la rapidez en búsquedas e informes y el acceso masivo a la información. Basados en los indicadores y con el fin de demostrarlos en el levantamiento de las encuestas establecemos 8 preguntas orientadas a responder y cualificar los resultados obtenidos con el Sistema de Infraestructura Provincial, y de este modo demostrar la hipótesis planteada.

Las preguntas que se usaron para la encuesta tienen dos posibles opciones de SI y NO y son las siguientes:

1. ¿Considera eficiente centralizar la información de los shape en una Base de Datos Espacial?
2. ¿Considera que ha mejorado la administración del Sistema de Infraestructura Vial al usar la base de Datos Espacial?
3. ¿Las búsquedas de información son más rápidas con el Sistema de Infraestructura Vial?
4. ¿Son intuitivas las búsquedas en el Sistema Generado?
5. ¿Le han ahorrado tiempo la generación de informes con el Sistema Generado?
6. ¿Considera útil la herramienta de generación de Informes del Sistema Generado?

7. ¿Cree usted que el visto de datos espaciales generado sea útil para el acceso masivo a la información?

8. ¿Considera fácil de acceder y usar a una persona sin conocimientos especiales al Sistema Generado?

4.5.4. Tabulación de resultados.

a.) Almacenamiento en disco.

Tabla 6

Resultados ítems almacenamiento

No.	Pregunta	Opción	Resp.	Total	%
1	¿Considera eficiente centralizar la información de los shape en una Base de Datos Espacial?	SI	28	28	100%
		NO	0		0%
2	¿Considera que ha mejorado la administración del Sistema de Infraestructura Vial al usar la base de Datos Espacial?	SI	28	28	100%
		NO	0		0%

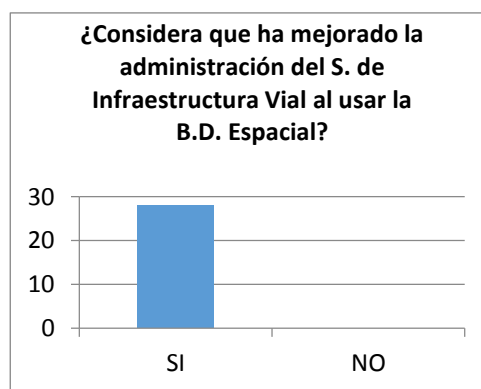
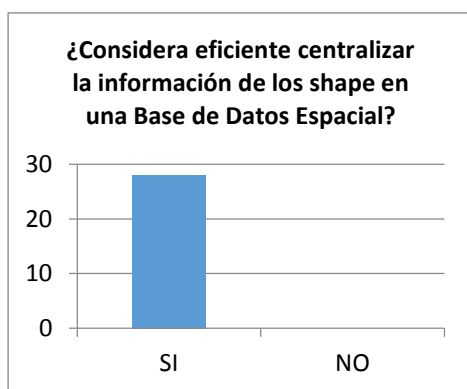


Figura 29 Resultados Porcentuales ítems almacenamiento

b.) Tiempo de respuesta en búsquedas.

Tabla 7

Resultados ítems respuesta en búsquedas

No.	Pregunta	Opción	Respuestas	Total	%
3	¿Las búsquedas de información son más rápidas con el Sistema de Infraestructura Vial?	SI	28	28	100%
		NO	0		0%
4	¿Son intuitivas las búsquedas en el Sistema Generado?	SI	28	28	100%
		NO	0		0%

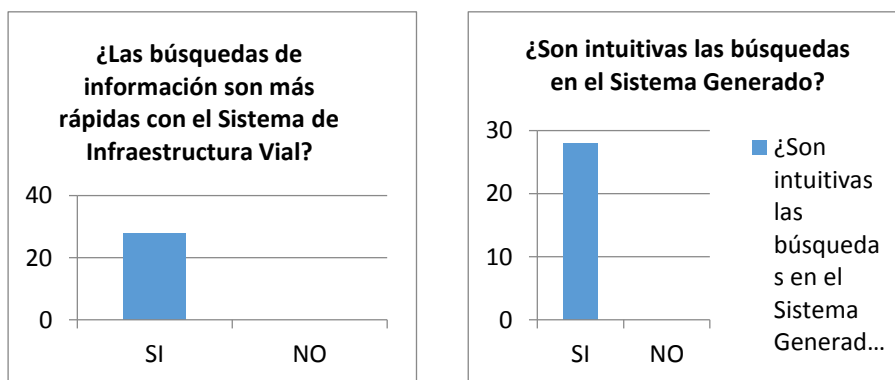


Figura 30 Resultados Porcentuales ítems búsquedas

c.) Tiempo de generación de informes.

Tabla 8

Resultados ítems generación informes

No.	Pregunta	Opción	Respuestas	Total	%
5	¿Le han ahorrado tiempo la generación de informes con el Sistema Generado?	SI	28	28	100%
		NO	0		0%
6	¿Considera útil la herramienta de generación de Informes del Sistema Generado?	SI	28	28	100%
		NO	0		0%

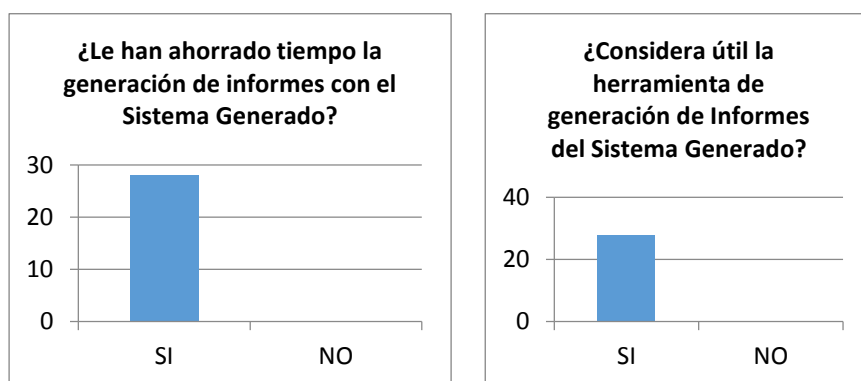


Figura 31 Resultados porcentuales ítems generación de informes

d.) Acceso masivo a la información

Tabla 9

Resultados ítems acceso a la información

No.	Pregunta	Opción	Respuestas	Total	Porcentaje
7	¿Cree usted que el visto de datos espaciales generado sea útil para el acceso masivo a la información?	SI	28	28	100%
		NO	0		0%
8	¿Considera fácil de acceder y usar a una persona sin conocimientos especiales al Sistema Generado?	SI	28	28	100%
		NO	0		0%

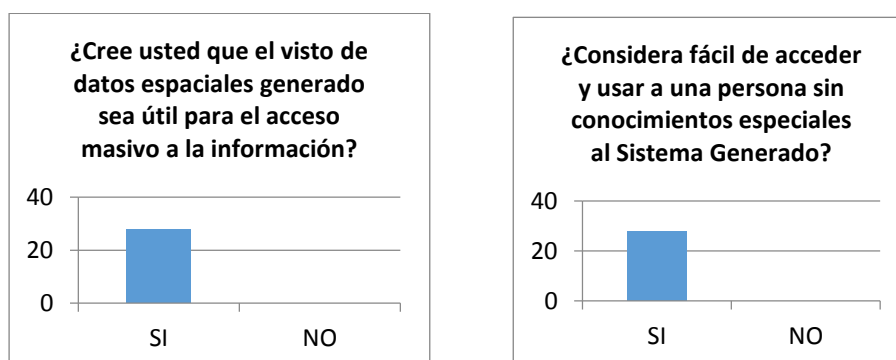


Figura 32 Resultados porcentuales ítems acceso a la información

4.5.5. Discusión de los resultados.

Acorde a los indicadores y las métricas establecidas claramente se puede afirmar que se ha optimizado el espacio de almacenamiento en disco, al centralizar la información del shape en una Base de Datos Espacial, usando el framework desarrollado en el presente trabajo de investigación; basta comparar los 15 MB usados en la Base de Datos creada en postgres vs los 86.8 MB que es el estimativo de uso en el archivo shape. También es importante y notorio la velocidad de respuesta en las consultas sql realizadas como prueba; como referencia se asumió el doble en una búsqueda manual, ya sea en un archivo Excel, en el archivo shape o en el cliente SIG. Y finalmente, basados en las estadísticas a la página web de la semana precedente a la recopilación de la información, es evidente que se masifica la información cuantitativamente.

A su vez, los resultados obtenidos en la encuesta realizada al personal experto del Departamento de Planificación, en los cuatro puntos centrales que se refieren a los indicadores y las 8 preguntas orientadas a responder dichos indicadores; claramente se nota como el conjunto de encuestados respondió afirmativamente en todas las preguntas efectuadas, lo que evidentemente comprueba la hipótesis planteada al inicio del presente trabajo de investigación en el que se demuestra que con el uso del framework para el diseño de la Base de Datos Espacial se optimiza el almacenamiento en disco en el Sistema de Infraestructura Vial de la Provincia de Chimborazo.

4.6. Conclusión del Capítulo.

Durante la aplicación no debe dejarse de lado, los problemas lógicos en cuestión de instalar herramientas, realizar conexiones, desarrollar interfaces, etc., que aunque no forman parte del desarrollo, sí constituyen parte de la solución. Finalmente, la validación del framework que es el paso que avala la solución y demuestra la hipótesis planteada al inicio de la investigación. Como no mencionar los indicadores determinados al inicio del presente trabajo, y que se busca la manera de medirlos cuantitativamente como cualitativamente, para concluir con la optimización de almacenamiento en disco.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La alta gerencia para la toma de decisiones, necesita información amigable e intuitiva para el desenvolvimiento de sus actividades; que un Sistema de Información Geográfica puede proveerlo.
- El costo de desarrollar un Sistema de Información Provincial, es bastante elevado; pero resulta más económico que no tenerlo.
- Para la correcta consecución de cualquier Sistema Informático, depende en gran medida del tiempo que se le asigne al análisis previo; mucho más fácil es corregir en fases tempranas que en las posteriores.
- Las cuatro fases para el desarrollo del framework son claramente definidas y explicadas; sin embargo, no menos importante es la consideración adicional que debe tener en cuanto a velocidad de respuesta y futura expansión de la base de datos espacial.
- El planteamiento de indicadores claros y medibles desde el inicio de la investigación, ayudaron a que podamos cualitativamente y cuantitativamente de manera sencilla demostrar la hipótesis planteada.
- El framework traslada un shape a Base de Datos Espacial. Sin embargo, es preferible el desarrollo de la Base de Datos Espacial en el mismo inicio del proyecto de georeferenciación para en lo posible disminuir los archivos shape.
- Concluimos que el framework de desarrollo EAR Espacial si optimiza el almacenamiento en disco, agiliza las búsquedas, genera informes de manera rápida y permite un acceso masivo a la información.

5.2. RECOMENDACIONES

- Se sugiere que los datos espaciales sean manejados con bastante prolijidad, dado que son el plus en cualquier tipo de aplicación.
- Es necesario que se determine muy bien los requerimientos de la Base de Datos Espacial, en la etapa de análisis previo con el fin de evitar errores posteriores.
- Se recomienda el uso de técnicas de data cleansing en los datos alfanuméricos y espaciales al momento de usar el framework.
- Dada la complejidad de estos sistemas se sugiere una planificación bien establecida y correctamente desarrollada cuando se trate de implementar sistemas de información geográfica.
- De las herramientas estudiadas, se sugiere el uso de software open; por cuestiones de escalabilidad, seguridad, costos, independencia, entre otros.

BIBLIOGRAFÍA

- Base de datos. (2015, septiembre 18). En *Wikipedia, la enciclopedia libre*. Recuperado a partir de https://es.wikipedia.org/w/index.php?title=Base_de_datos&oldid=85204784
- Cabello, M. V. N. (s. f.). *Introduccion a Las Bases de Datos Relacionales*. Editorial Visión Libros.
- Capote, O. P. (2008). *INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS*. Editorial Paraninfo.
- Castillo, C. R. (2010). *Sistema Operativo GNU con Linux - Trisquel* (Primera, Vol. 1). Ecuador: Martin Iturbide.
- CHANCAY ZAMBRANO, M., & ZAMBRANO ARIAS, M. J. (2015). *PROPUESTA DE UN MODELO DE GESTIÓN DE CONTROL INTERNO PARA EL GAD PARROQUIAL ÁNGEL PEDRO GILER DEL CANTÓN TOSAGUA*. Recuperado a partir de <http://181.196.143.6/handle/123456789/766> (2015, Mayo 20)
- C. J. Date. (2001). *Introducción a los sistemas de bases de datos*. Pearson Educación.
- Comas, D., & Ruiz, E. (1993). *Fundamentos de los sistemas de información geográfica*. Ariel.
- Cornelio, E. R., Rivas, C. G., & Hernández, J. C. R. (2004). *Bases de datos relacionales: diseño físico: Orientada al DB2 para z/OS de IBM*. Univ Pontifica Comillas.

- DEL TRABAJO, S. G. D. R. (2008). Constitución de la República del Ecuador. Recuperado a partir de <http://www.prosigma.com.ec/pdf/nlegal/Constitucion-de-la-Republica-del-Ecuador.pdf> (2008, Junio 20)
- Ecuador, Ecuador, & Secretaría Nacional de Planificación y Desarrollo. (2013). *Buen vivir: plan nacional 2013-2017 : todo el mundo mejor*.
- Galván Salinas, A. M., & Jiménez Alfaro, A. J. (s. f.). Fundamentos de la teoría relacional para extracción de datos MultiValores del RDBMS jBase. Recuperado a partir de <http://repositoriodigital.academica.mx/jspui/handle/987654321/82149> (2015, Junio 15)
- García, J. M. C. (2015). *Almacenamiento de la información e introducción a SGBD. IFCT0310*. IC Editorial.
- Kendall, J. E. (1997). *Análisis y diseño de sistemas*. Pearson Educación.
- Llopis, J. P. (2006). *Sistemas de información geográfica aplicados a la gestión del territorio: entrada, manejo, análisis y salida de datos espaciales : teoría general y práctica para ESRI ArcGIS 9*. Editorial Club Universitario.
- Lope, R. Á. P. de. (2015). *SGBD e instalación. IFCT0310*. IC Editorial.
- Meaden, G. J., & Kapetsky, J. M. (1992). *Los sistemas de información geográfica y la telepercepción en la pesca continental y la acuicultura*. Food & Agriculture Org.

- Medina Narváez, G. B., & Morán Cabrera, E. I. (2011). Implementación de un sitio web turístico para el complejo La Piedra Grande ubicado en el recinto La Victoria del cantón Cumandá. Recuperado a partir de <http://190.95.144.28/handle/123456789/88> (2011, Julio 30)
- Navarro, A. P. (2011). *Introducción a los sistemas de información geográfica y geotelemática*. Editorial UOC.
- Pascoe, R. T. (1994). Construction of interfaces for the transfer of data between geographical information systems. Recuperado a partir de <http://ir.canterbury.ac.nz/handle/10092/8421> (2015, Abril 25)
- Sumathi, S., & Esakkirajan, S. (2007). *Fundamentals of Relational Database Management Systems*. Springer Science & Business Media.
- Zarzosa, N. L., & Andrés, M. A. N. (2004). *Sistemas de información geográfica. Prácticas con Arc View*. Univ. Politèc. de Catalunya.

ANEXOS

Anexos 1

El Sistema de Infraestructura Vial del Gobierno Provincial de Chimborazo es el inicio de una Plataforma completa de servicios denominado Sistema de Información Provincial, en donde se van a incluir no solo el sistema Vial, sino también y posteriormente los ejes de Ambiente, Producción y Turismo en una consolidada plataforma georeferenciada para el servicio a la Comunidad de la Provincia y del País.

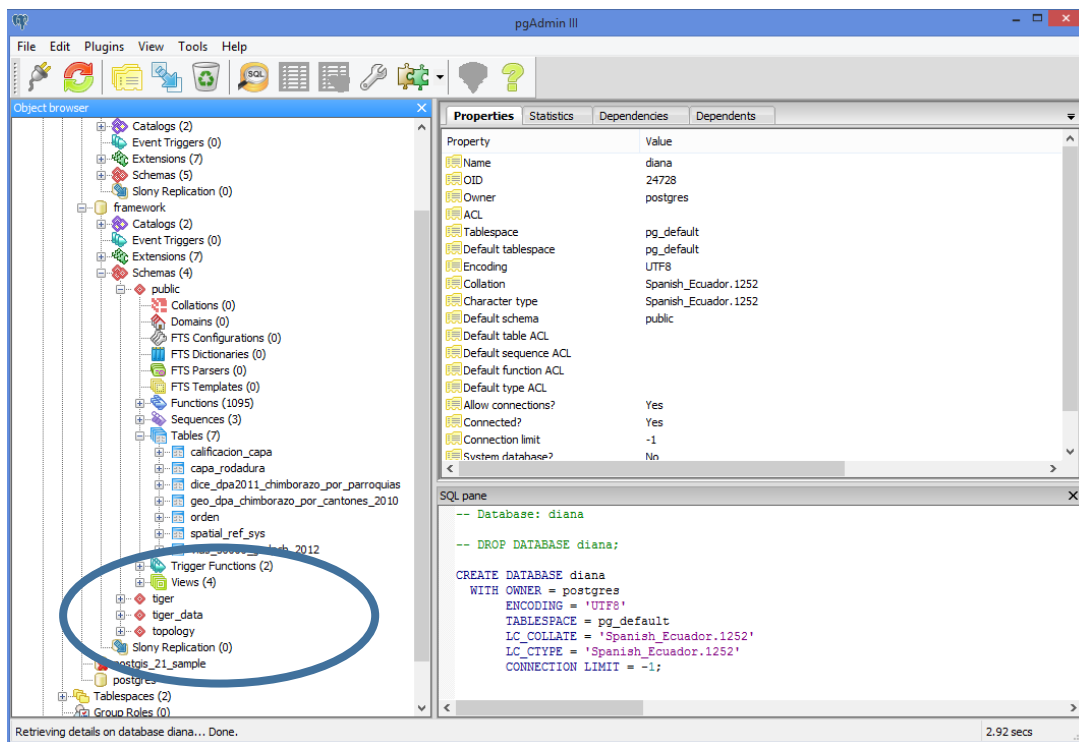
Como inicio de las tareas se subió la Base de Datos Espacial, siguiendo una serie de pasos descritos a continuación:

1. Se instaló la versión 9 de Postgresql, incluyendo el módulo del Postgis.
2. Se subieron los datos desde el shape al postgis, utilizando librerías y sentencias, tal como

```
shp2pgsql -s 4326 -I -W LATIN1 d:\vias_50000_GADPCH_2012 vias > d:\vi.sql
```

3. Al ejecutar el pgAdmin, se procedió a realizar algunas sentencias con el fin de limpiar los datos

La Base de Datos Espacial después de las modificaciones efectuadas quedó de la siguiente manera:



Como se puede observar los nombres de las tablas que conforman la Base de Datos coinciden con los diseñados en el framework, los nombres importadores directamente desde los shape conservan la misma denominación.

Ya dentro de la Base de Datos en Postgis, es sumamente fácil trabajar con los datos y hacer consultas de cualquier tipo a los datos georeferenciados; pues se aplica el lenguaje SQL para las consultas.

A continuación sentencias sql para recoger datos, enlazando tablas y con ello trabajar con un sistema relacional.

a. Pregunta por el cantón guano, entre la tabla de vías y la de cantón

```
SELECT
```

```
*
```

```
FROM
```

```
vias_50000_gadpch_2012, geo_dpa_chimborazo_por_cantones_2010
```

```
where
```

```
vias_50000_gadpch_2012.canton=geo_dpa_chimborazo_por_cantones_2010.dpa_descan and  
geo_dpa_chimborazo_por_cantones_2010.dpa_descan='GUANO';
```

The screenshot shows a PostgreSQL SQL Editor window with the following query:

```
SELECT *
FROM
vias_50000_gadpch_2012, geo_dpa_chimborazo_por_cantones_2010
where
vias_50000_gadpch_2012.canton=geo_dpa_chimborazo_por_cantones_2010.dpa_descan and geo_dpa_chimborazo_por_cantones_2010.dpa_descan='GUANO';
```

The output pane displays the following table:

	tramo character varying(150)	longitud_1 double precision	actualizad character varying(50)	fecha character varying(10)	año character varying(5)	canton character varying(15)	obra_arte character varying(15)
1	ZAHJAFAMBA-VIA TAMBOHUASHA	3.134796162	EDGAR DAMIAN		2012	GUANO	
2	SILVERIA - VIA CALSHI HIERBA BUENA	2.231498441	EDGAR DAMIAN		2012	GUANO	
3	SAN ANDRES-JOSEFINA	1.178942153	EDGAR DAMIAN		2012	GUANO	
4	LA JOSEFINA - STA LUCIA DE TAMBO - SAN ISIDRO	0.699130225	EDGAR DAMIAN		2012	GUANO	
5	PICHAN-SAN PABLO	6.303906655	EDGAR DAMIAN		2012	GUANO	
6	PICHAN CARLOS FIN DE CAMINO	2.979329302	EDGAR DAMIAN		2012	GUANO	
7	TRAMO 6	0.357340787	EDGAR DAMIAN		2012	GUANO	
8	PICHAN SAN CARLOS - SANTA LUCIA DE TEMBO	7.97266064	EDGAR DAMIAN		2012	GUANO	
9	VIA SAN JACINTO-SAN GERARDO	0.128753255	ALVARO DELLI		2012	GUANO	
10	TRAMO 8	1.470535911	EDGAR DAMIAN		2012	GUANO	
11	BARRIO SAN RAFAEL-SAN VICENTE LLINGUINDE	2.01374022	EDGAR DAMIAN		2012	GUANO	
12	SAN ISIDRO-BARRIO SAN RAFAEL	0.439826413	EDGAR DAMIAN		2012	GUANO	
13	SAN ISIDRO-DELICIA-TUTUCFALA	1.661959613	EDGAR DAMIAN		2012	GUANO	

b. Pregunta por el tipo de capa 'REGULAR' enlazado con las tablas

SELECT

*

FROM

vias_50000_gadpch_2012, calificacion_capa

where

vias_50000_gadpch_2012.calif_capa=calificacion_capa.cod_calificacion
and calificacion_capa.nom_calificacion='REGULAR';

The screenshot shows a PostgreSQL SQL Editor window with the following query:

```
SELECT *
FROM
vias_50000_gadpch_2012, calificacion_capa
where
vias_50000_gadpch_2012.calif_capa=calificacion_capa.cod_calificacion and calificacion_capa.nom_calificacion='REGULAR';
```

The output pane displays the following table:

	tramo character varying(150)	longitud_1 double precision	actualizad character varying(50)	fecha character varying(10)	año character varying(5)	canton character varying(15)	obr cha
1	SAGUAZO LA UNIÓN- CAHUAJI ALTO-CHAZO ALTO - TRAMO 1	0.895494374	FERNANDO VASCONEZ		2012	GUANO	
2	GUANO - SAN ANDRES	3.653564653	ADMINISTRADOR		2012	GUANO	
3	LA JOSEFINA - STA LUCIA DE TAMBO - SAN ISIDRO	0.517566483	EDGAR DAMIAN		2012	GUANO	
4	SECAO TROJE SAN BARTOLO CASTUS ALTO	1.926969796	ALVARO DELLI	05-07	2012	COLTA	ALC
5	ALACAO - SANTA ANA DE GUANO	0.656967594	ALVARO DELLI		2012	GUANO	FUE
6	ALACAO - SANTA ANA DE GUANO	0.213613166	ALVARO DELLI		2012	GUANO	FUE
7		0.428005792	TRAMO 08		2012	RIOBAMBA	
8	4 ESQUINAS - BATZACON -SIGSIPAMBA	1.391694375	EDGAR DAMIAN		2012	GUANO	
9	VIA LLAMALLON	3.347624799	FERNANDO VASCONEZ	18-08	2012	CHUNCHI	
10	VIA MAGNA CHUNCHI	3.268727365	FERNANDO VASCONEZ	09-08	2012	CHUNCHI	CUN
11	SAN VICENTE-TAMBO BAJO	15.5	EDGAR DAMIAN		2012	COLTA	
12	PICHAN-SAN PABLO	6.303906655	EDGAR DAMIAN		2012	GUANO	
13	RUMICRUZ-CATEQUILLA-TITATCON	0.373478491	FERNANDO VASCONEZ		2012	CHAMBO	ALC
14	VIA MAGNA CHUNCHI	1.550833936	FERNANDO VASCONEZ	09-08	2012	CHUNCHI	CUN

c. Consulta por tipo de capa de rodadura, consulta TIERRA, enlazado entre tablas

SELECT

*

FROM

vias_50000_gadpch_2012, capa_rodadura

where

vias_50000_gadpch_2012.capa_rodad=capa_rodadura.cod_capa and
capa_rodadura.nom_capa='TIERRA';

The screenshot shows a PostgreSQL SQL Editor window with the following query:

```
FROM
vias_50000_gadpch_2012, capa_rodadura
where
vias_50000_gadpch_2012.capa_rodad=capa_rodadura.cod_capa and capa_rodadura.nom_capa='TIERRA';
```

The output pane displays the following data:

Tramo character varying(150)	longitud_1 double precision	actualizad character varying(50)	fecha character varying(10)	año character varying(5)	canton character varying(15)	obra_arte character varyin...
1 LA JOSEFINA - STA LUCIA DE TAMBO - SAN ISIDRO	0.517566483	EDGAR DAMIAN		2012	GUANO	
2 ALACAO - SANTA ANA DE GUANO	0.656967594	ALVARO DELLI		2012	GUANO	FUENTE
3 ALACAO - SANTA ANA DE GUANO	0.213613166	ALVARO DELLI		2012	GUANO	FUENTE
4	0.428005792	TRAMO 08		2012	RIOBAMBA	
5 URBINA - LLIO	4.791299427	EDGAR DAMIAN		2012	GUANO	
6 TRAMO 71	1.231411091	EDGAR DAMIAN		2012	RIOBAMBA	
7 TRAMO 179	1.462978238	EDGAR DAMIAN	12-07	2012	GUANO	
8 LIGLIG-LA VAQUERIA	1.994846894	FERNANDO ZABALA		2012	COLTA	ALCANTARILLADO
9 LA SILVERIA CAMPAMENTO	3.48640327	IVAN ROJAS		2012	GUANO	
10 VIA LA PUNTILLA-PANAMERICANA SUR 2	0.716986347	FERNANDO ZABALA		2012	COLTA	
11 VIA A LAS ROSAS DESVIO A GABUIN GRANDE	0.368627169	CRISTIAN PALACIOS	31-07	2012	FALLATANGA	
12 PUSÑAG JESUS DEL GRAN POSER - LA DELICIA - TRAMO 1	1.437045573	EDGAR DAMIAN		2012	GUANO	
13 DESVIO EL MANZANO-CROGLONTIS	1.344031203	CRISTIAN PALACIOS		2012	PERIPE	
14 PUSÑAG VIA VALPARAISO	3.801601768	EDGAR DAMIAN		2012	GUANO	

d. Consulta por tipo de orden, consulta 1 ORDEN, enlazado entre tablas

SELECT

*

FROM

vias_50000_gadpch_2012, orden

where

vias_50000_gadpch_2012.orden=orden.cod_orden and orden.nom_orden='1
ORDEN';

	tramo character varying(150)	longitud_1 double precision	actualizad character varying(50)	fecha character varying(10)	año character varying(5)	canton character varying(15)	obra_arte character varying(30)	capa_rodad character varying(30)	calif_cai charact
1	LIMITE PROVINCIAL SAN ANDRES RIOBAMBA	1.592087876	IVAN ROJAS		2012	GUANO	ALCANTARILLADO CUNETA	3	1
2	RED ESTATAL RIOBAMBA -PENIPE-BAÑOS	0.811135887	CRISTIAN PALACIOS		2012	PENIPE	ALCANTARILLADO CUNETA	3	1
3	PANAMERICANA - COLTA	0.246278401	IVAN ROJAS		2012	RIOBAMBA	ALCANTARILLADO CUNETA PUENTE	3	1
4	RED ESTATAL RIOBAMBA -PENIPE-BAÑOS	2.132814234	CRISTIAN PALACIOS		2012	PENIPE	ALCANTARILLADO CUNETA	3	1
5	PANAMERICANA - COLTA	1.374609875	IVAN ROJAS		2012	RIOBAMBA	ALCANTARILLADO CUNETA PUENTE	3	1
6	PANAMERICANA - COLTA	0.395744655	IVAN ROJAS		2012	RIOBAMBA	ALCANTARILLADO CUNETA PUENTE	3	1
7	LIMITE PROVINCIAL SAN ANDRES RIOBAMBA	3.992567252	IVAN ROJAS		2012	GUANO	ALCANTARILLADO CUNETA	3	1
8	LIMITE PROVINCIAL SAN ANDRES RIOBAMBA	2.508455023	IVAN ROJAS		2012	GUANO	ALCANTARILLADO CUNETA	3	1
9	LIMITE PROVINCIAL SAN ANDRES RIOBAMBA	3.992567252	IVAN ROJAS		2012	GUANO	ALCANTARILLADO CUNETA	3	1
10	LIMITE PROVINCIAL SAN ANDRES RIOBAMBA	1.386367004	IVAN ROJAS		2012	GUANO	ALCANTARILLADO CUNETA	3	1
11	VIA JOYASHI (PANAMERICANA)	2.539802691	FERNANDO VASCONEZ	13-08	2012	CHUNCHI	ALCANTARILLADO CUNETA	3	1
12	VIA JOYASHI (PANAMERICANA)	1.572615277	FERNANDO VASCONEZ	13-08	2012	CHUNCHI	ALCANTARILLADO CUNETA	3	1
13	VIA JOYASHI (PANAMERICANA)	0.949044318	FERNANDO VASCONEZ	13-08	2012	CHUNCHI	ALCANTARILLADO CUNETA	3	1
14	CHUNCHI-RIOBAMBA II PARTE	0.946671243	IVAN ROJAS		2012	ALAUSI	ALCANTARILLADO CUNETA PUENTE	3	1

- Posteriormente se instaló el Geoserver.
- A continuación se instaló el QGis, y se hizo la conexión con el almacén de datos de postgis
- Finalmente, se subió el aplicativo para la representación de los datos en sus respectivos mapas.

Aplicativo

Al aplicativo que está colgado en la web se lo puede acceder desde la página web de la Institución:

www.chimborazo.gob.ec en la parte inferior derecha con el título. Sistema de Información Provincial.

El aplicativo tiene las siguientes pantallas:

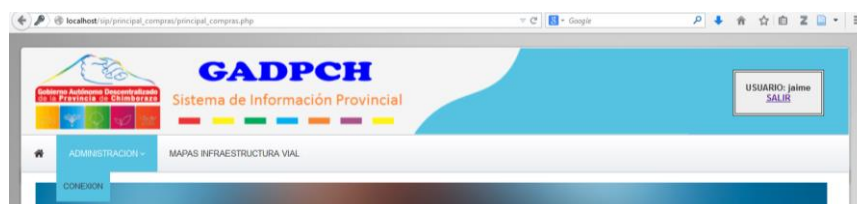
Pantalla de Inicio



Cuando se logea como invitado, tiene acceso a los reportes y cuando se logea como administrador, tiene acceso a las conexiones y claves.

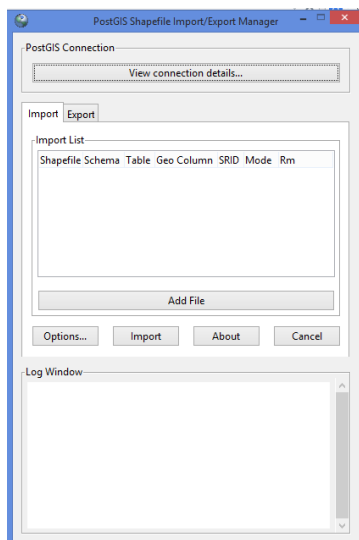
El usuario Administrador tiene el nombre admin y la clave utics; el usuario invitado tiene nombre admin y clave admin.

El administrador tiene las siguientes opciones:

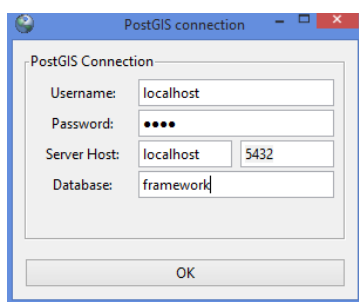


Iniciando con el menú, encontramos en la administración la opción de conexión. Si escogemos en esta opción se abre la librería correspondiente para la configuración de la conexión y los parámetros con los que se va a comunicar con postgresql y el módulo de postgis.

La pantalla que se abre es la siguiente:



Dentro de la cual establecemos los parámetros de comunicación que debemos realizar para la conexión con postgis, tal como se muestra en la siguiente figura:



Dentro de las opciones que disponemos encontramos la importación y exportación desde archivos shape, o desde una tabla.

Menú de Mapas

También tenemos dentro del sistema un menú para mostrar los mapas generados desde el geoserver, entre los que disponemos se encuentran:



Podemos elegir entre mapas caracterizados por cantón, por tipo de rodadura, por estado de la vía y por clasificación de orden.

Generados de mapas geo server

Para disponer de los mapas que se van a representar necesitamos de un servidor de mapas, para el presente trabajo se escogió geoserver; en este mapa configuramos lo que tiene que ver con los datos en los cuales incluye la comunicación con el motor de base de datos postgis.



Generamos las capas que necesitamos para la representación de acuerdo al principio de búsqueda. Se puede personalizar las sentencias sql que vamos a generar en el mapa.


Anexo 2

La encuesta realizada se encuentra en el siguiente link, que se publicó en la intranet de la Institución, brindando un acceso directo a todos los interesados.

https://docs.google.com/forms/d/1R7H1V9I502egfOFI80DkCanR1e56A1klhbq-nWNKbss/viewform?usp=send_form

Las consultas realizadas junto con el tiempo que se tardó cada una de ellas se presenta a continuación.

- **Por cantón Riobamba**



The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor Graphical Query Builder
```

Previous queries

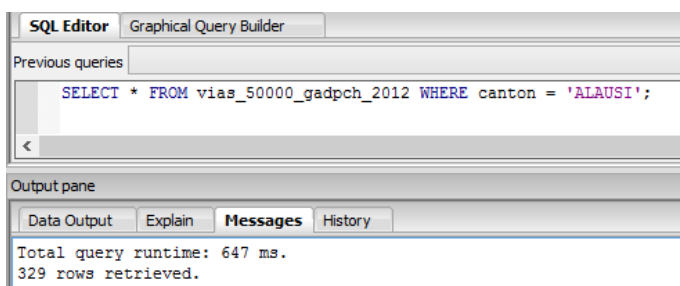
```
SELECT * FROM vias_50000_gadpch_2012 WHERE canton = 'RIOBAMBA';
```

Output pane

Data Output Explain Messages History

```
Total query runtime: 820 ms.  
735 rows retrieved.
```

- **Por cantón Alausí**



The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor Graphical Query Builder
```

Previous queries

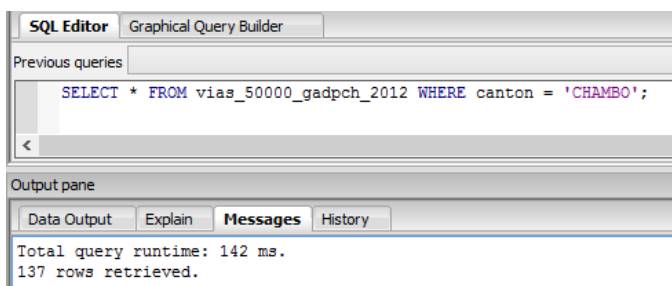
```
SELECT * FROM vias_50000_gadpch_2012 WHERE canton = 'ALAUSSI';
```

Output pane

Data Output Explain Messages History

```
Total query runtime: 647 ms.  
329 rows retrieved.
```

- **Por cantón Chambo**



The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor Graphical Query Builder
```

Previous queries

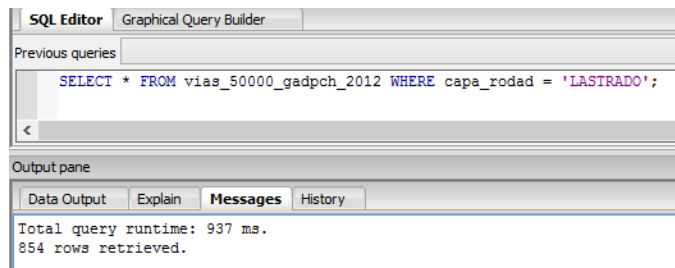
```
SELECT * FROM vias_50000_gadpch_2012 WHERE canton = 'CHAMBO';
```

Output pane

Data Output Explain Messages History

```
Total query runtime: 142 ms.  
137 rows retrieved.
```

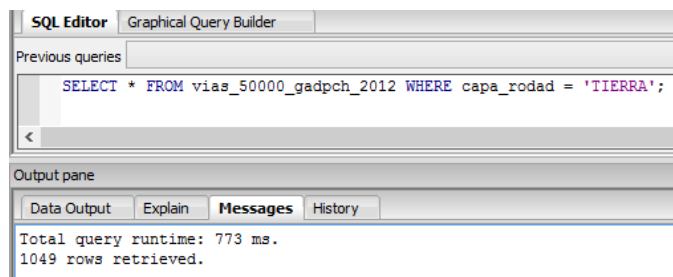
- **Por capa de rodadura Lastrado**



The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor | Graphical Query Builder  
Previous queries  
SELECT * FROM vias_50000_gadpch_2012 WHERE capa_rodad = 'LASTRADO';  
<  
Output pane  
Data Output | Explain | Messages | History  
Total query runtime: 937 ms.  
854 rows retrieved.
```

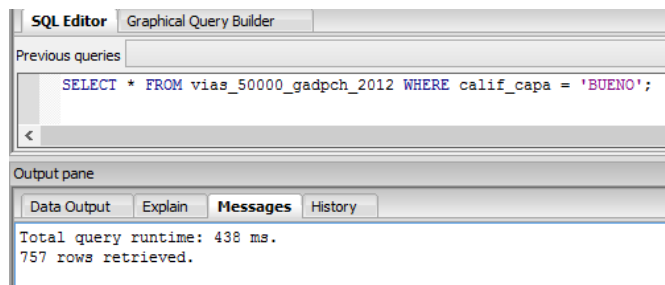
- **Por capa de rodadura Tierra**



The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor | Graphical Query Builder  
Previous queries  
SELECT * FROM vias_50000_gadpch_2012 WHERE capa_rodad = 'TIERRA';  
<  
Output pane  
Data Output | Explain | Messages | History  
Total query runtime: 773 ms.  
1049 rows retrieved.
```

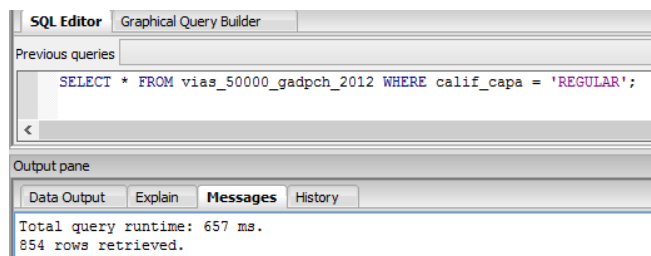
- **Por calificación de capa Bueno**



The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor | Graphical Query Builder  
Previous queries  
SELECT * FROM vias_50000_gadpch_2012 WHERE calif_capa = 'BUENO';  
<  
Output pane  
Data Output | Explain | Messages | History  
Total query runtime: 438 ms.  
757 rows retrieved.
```

- **Por calificación de capa Regular**



The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor | Graphical Query Builder  
Previous queries  
SELECT * FROM vias_50000_gadpch_2012 WHERE calif_capa = 'REGULAR';  
<  
Output pane  
Data Output | Explain | Messages | History  
Total query runtime: 657 ms.  
854 rows retrieved.
```

- **Por calificación de capa Malo**

The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor | Graphical Query Builder
```

Previous queries

```
SELECT * FROM vias_50000_gadpch_2012 WHERE calif_capa = 'MALO';
```

Output pane

Data Output | Explain | **Messages** | History

Total query runtime: 859 ms.
1092 rows retrieved.

- **Por orden 4 orden**

The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor | Graphical Query Builder
```

Previous queries

```
SELECT * FROM vias_50000_gadpch_2012 WHERE orden = '4 ORDEN';
```

Output pane

Data Output | Explain | **Messages** | History

Total query runtime: 1459 ms.
1839 rows retrieved.

- **Por orden 3 orden**

The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor | Graphical Query Builder
```

Previous queries

```
SELECT * FROM vias_50000_gadpch_2012 WHERE orden = '3 ORDEN';
```

Output pane

Data Output | Explain | **Messages** | History

Total query runtime: 430 ms.
691 rows retrieved.

- **Por parroquia Columbe**

The screenshot shows the SQL Editor interface with the following content:

```
SQL Editor | Graphical Query Builder
```

Previous queries

```
SELECT * FROM vias_50000_gadpch_2012 WHERE parroquia = 'COLUMBE';
```

Output pane

Data Output | Explain | **Messages** | History

Total query runtime: 94 ms.
145 rows retrieved.

- **Por parroquia San Andres**

SQL Editor Graphical Query Builder

Previous queries

```
SELECT * FROM vias_50000_gadpch_2012 WHERE parroquia = 'SAN ANDRES';
```

Output pane

Data Output Explain **Messages** History

Total query runtime: 125 ms.
177 rows retrieved.

- **Por parroquia Achupallas**

SQL Editor Graphical Query Builder

Previous queries

```
SELECT * FROM vias_50000_gadpch_2012 WHERE parroquia = 'ACHUPALLAS';
```

Output pane

Data Output Explain **Messages** History

Total query runtime: 78 ms.
42 rows retrieved.

- **Por longitud <5**

SQL Editor Graphical Query Builder

Previous queries

```
SELECT * FROM vias_50000_gadpch_2012 WHERE longitud_1 <5
```

Output pane

Data Output Explain **Messages** History

Total query runtime: 1460 ms.
2537 rows retrieved.

- **Por longitud >=5**

SQL Editor Graphical Query Builder

Previous queries

```
SELECT * FROM vias_50000_gadpch_2012 WHERE longitud_1 >=5
```

Output pane

Data Output Explain **Messages** History

Total query runtime: 500 ms.
169 rows retrieved.

