

**Universidad de las Ciencias Informáticas  
Facultad 6**



# **GeoQ: Desarrollo del módulo de interrelación de múltiples tablas y capas**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS  
INFORMÁTICAS**

## **AUTORES:**

**Dunia Rodríguez Parra**

**Maidy Emily Cancio Fernández**

## **TUTOR:**

**Ing. Vladimir Martell Fernández**

**La Habana, 21 de junio de 2012  
“Año 54 de la Revolución”**

## *Dedicatoria*

### **De Maily:**

A mi abuelita **Cira** por sus horas de desvelo orándole a **Dios** para que lograra este sueño que se ha vuelto realidad, y por sus sabios consejos.

A mis padres **José René Cancio Arredondo** y **Magalys Fernández Hernández** mis preciados tesoros y eternos amores, quienes me han apoyado siempre en todas mis decisiones.

A mi hermana **Yoanka** por ser mi ejemplo a seguir, por sorprenderme siempre por los momentos preciosos que paso con ella, y por ayudarme tanto siempre que lo necesito.

A mi novio **Raydel** por ayudarme en todo y ayudarme a ser una mejor persona y por hacerme sentir tan especial.

### **De Dunia:**

A mis padres **Adeyvis Parra Benítez** y **Alberto Julio Rodríguez Cazañas** por representar mi hilo conductor durante todos estos años, brindándome un apoyo incondicional, y por demostrarme que no existe la palabra imposible.

A mi hermana **Mileidys** (Miki), por devolverme la alegría cuando la necesitaba, siempre me haces reír, por poder contar con ella en todo momento.

A mi sobrinita **Keyla**, por ser esa personita tan especial en mi vida y capaz de hacerme olvidar los problemas con tan una sonrisita.

# *Agradecimientos*

## **De Maily:**

A **Raydel Raúl Viñolo Sosa**, mi ángel de la guarda por ayudarme desde el principio de esta jornada de cinco años, por estar siempre, siempre, dispuesto a apoyarme y acompañarme cuando más lo necesito, a él, hoy mi novio, gracias por estar a mi lado y ser la persona especial que has sido durante este quinto año.

A **Magalys**, mi mamita peleona que la quiero con la vida y a mi papito que adoro: **José René**, un soñador de la vida (Atagerge), a ustedes gracias por apoyarme tanto, por ayudarme en todo, por darme y hacerme alcanzar hasta lo imposible, a ustedes por ser la luz que me ha guiado siempre.

A mi **abuela** por todo, por sus sorpresas y su ternura.

A mi hermana **Yoanka**, por el apoyo y preocupación de mi futuro, siempre al tanto.

A mis tíos **Arnaldo** y **Alexis** y mi tía **Tamara**, por el apoyo que siempre me han dado.

A mis padrinos **Maribel** (Marita) y mi tío **Felix**, por su preocupación.

A **Leyanis**, **Yory**, **Dayana**, **Emilio**, **Walfrido**, **Richard** y **Octavio** gracias por ayudarme a superar el primer año.

A **Maidelyn** y **Claudia** por hacerme reír, por los buenos momentos.

# *Agradecimientos*

## **De Dunia:**

A mi novio **Pedro Enrique Palau Isaac** por demostrarme todas las cosas lindas que hace por mí, por no dejar de apoyarme en cada momento que lo necesito y estar siempre a mi lado brindándome fuerza y seguridad.

A mis padres **Adeyvis Parra Benítez** y **Alberto Julio Rodríguez Cazañas**, por ser las personas más importantes en mi vida, por compartir conmigo cada momento y por tratar de que nunca me falte nada.

A mi hermana **Mileidys** y mi sobrinita **Keyla**, por darme un cariño y afecto incondicional.

A mi **San Lázaro** por cuidarme y protegerme en todos estos años de mi vida.

A mi **familia**, por darme ánimos para seguir adelante.

# *Agradecimientos*

## **De las dos:**

Al Ing. **Vladimir Martell Fernández**, por ser un ejemplo de tutor, al cual le damos las gracias por guiarnos y ayudarnos a realizar este sueño, por enfrentar junto a nosotras cada obstáculo que desde el principio tuvimos, de los cuales gracias a su empeño y dedicación salimos gloriosas y seguras de poder seguir adelante.

A nuestros compañeros de batalla **Palau** y **Viñolo**, porque sin ellos esta tesis no hubiera sido posible, gracias por su esmero, esfuerzo y tiempo dedicado.

A nuestras **Amistades** por compartir los buenos y malos momentos de nuestras vidas, por ayudar siempre que lo necesitamos y saber que se puede contar con ellos, en especial **Yory, Viana, Leyanis** y **Leidy** y a todos los del grupo **9102**.

A todos **nuestros profesores**, por contribuir a nuestra formación humana y profesional a lo largo de estos cinco años.

A **nuestro equipo de trabajo** siempre dispuesto a ayudarnos.

A la **Universidad de las Ciencias Informáticas** por educarnos y convertirnos en profesionales comprometidos con nuestra sociedad y prepararnos para la vida.

**¡A todas las personas que siempre nos apoyaron durante este largo periodo!**

## *Declaración de autoría*

Declaramos por este medio que nosotras, Dunia Rodríguez Parra y Maily Emily Cancio Fernández, con carné de identidad 89041617772 y 89021728414 respectivamente somos las autoras de este trabajo y que autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste, firmamos la presente declaración jurada de autoría en La Habana a los 21 días del mes de junio del año 2012.

---

Dunia Rodríguez Parra  
**Autor**

---

Maily Emily Cancio Fernández  
**Autor**

---

Ing. Vladimir Martell Fernández  
**Tutor**

# *Resumen*

El constante desarrollo de los sistemas de información geográfica ha traído consigo la creación de novedosas y ventajosas funcionalidades que brindan mejoras a la manipulación y análisis de la información geográfica y socioeconómica. La mayoría de las empresas en la actualidad optan por el uso de estos sistemas, los cuales apoyan el proceso de la toma de decisiones.

La investigación que se presenta consiste en el análisis, diseño e implementación de un módulo para la interrelación de múltiples tablas y capas pues apoya la gestión de la información que se maneja. Pretende disminuir las limitaciones de usabilidad y de análisis de información geográfica y socioeconómica del SIG GeoQ.

El desarrollo del módulo propuesto expone las ventajas para el análisis de la información que a continuación se mencionan: maneja la información socioeconómica de conjunto con la información geográfica, disminuye la posibilidad de entrada de errores con la carga de información socioeconómica de fuentes externas y permite realizar complejos análisis de la información con grandes volúmenes de datos.

***Palabras clave:*** GeoQ, SIG, interrelación de tablas y capas.

# Índice

<b>Introducción</b>	<b>1</b>
<b>Capítulo 1: “Fundamentación Teórica”</b>	<b>5</b>
1.1 Introducción	5
1.2 Conceptos asociados al dominio del problema	5
1.2.1 Aplicaciones informáticas	5
1.2.2 Gestión de la información	5
1.2.3 Información socioeconómica	6
1.2.4 Información geográfica	7
1.2.5 Cartografía	7
1.2.6 Base de Datos Espacial	8
1.2.7 Consultas Espaciales	8
1.2.8 Interrelación de tablas y capas	8
1.2.9 Sistemas de Información Geográfica	9
1.3 Objeto de Estudio	11
1.3.1 Descripción General	11
1.4 Análisis de otras soluciones existentes	13
1.4.1 ArcGIS	13
1.4.2 GvSig	14
1.4.3 La interrelación de múltiples tablas y capas en las soluciones anteriores	15
1.5 Conclusiones Parciales	16
<b>Capítulo 2: “Tendencias, tecnologías y herramientas a utilizar”</b>	<b>17</b>
2.1 Introducción	17
2.2 Metodología de desarrollo de software	17
2.3 Lenguaje de unificado de modelado: UML	19
2.4 Visual Paradigm como herramienta CASE	19
2.5 Lenguaje de programación a utilizar	20
2.6 Entorno Integrado de Desarrollo (IDE)	21
2.7 Sistema Gestor de Base de Datos	22
2.8 Arquitectura de software	23
2.8.1 Patrones Arquitectónicos	24
2.9 Conclusiones Parciales	25



<b>Capítulo 3: “Análisis de la solución propuesta”</b>	<b>26</b>
3.1 Introducción	26
3.2 Modelo de Dominio	26
3.2.1 Descripción de las clases del dominio	26
3.3 Requisitos	27
3.3.1 Requisitos Funcionales	27
3.3.2 Requisitos no Funcionales	29
3.4 Descripción del sistema propuesto	30
3.4.1 Descripción de los actores del sistema	30
3.4.2 Diagrama de Casos de Uso del Sistema	30
3.4.3 Descripción textual de los Casos de Uso del Sistema	31
3.5 Conclusiones Parciales	35
<b>Capítulo 4: “Implementación y pruebas de la solución”</b>	<b>37</b>
4.1 Introducción	37
4.2 Patrones de Diseño	37
4.3 Modelo de Diseño	39
4.4 Modelo de implementación	39
4.5 Diagrama de componentes	41
4.6 El proceso de pruebas	41
4.6.1 Diseños de casos de prueba	43
4.7 Conclusiones parciales	53
<b>Conclusiones Generales</b>	<b>54</b>
<b>Recomendaciones</b>	<b>55</b>
<b>Referencias Bibliográficas</b>	<b>56</b>
<b>Bibliografía</b>	<b>59</b>
<b>Glosario de Términos</b>	<b>62</b>

# Índice de tablas y figuras

## Tablas

Tabla 1: Definición de los actores. -----	30
Tabla 2: Descripción textual del Caso de Uso Gestionar Relación. -----	35
Tabla 3: Secciones a probar del Caso de Uso del Sistema Gestionar Relación.-----	48
Tabla 4: Descripción de las variables. -----	49
Tabla 5: Matriz de Datos SC 1 Crear Relación. -----	51
Tabla 6: Matriz de Datos SC 2 Modificar Relación.-----	53
Tabla 7: Matriz de Datos SC 3 Eliminar Relación. -----	53

## Figuras

Figura 1: Clasificación de las capas. ....	10
Figura 2: Interfaz de ArcGIS .....	13
Figura 3: Interfaz de GvSig .....	14
Figura 4: Interrelación de tablas ArcGIS .....	15
Figura 5: Interrelación de tablas GvSig. ....	16
Figura 6: Fases y flujos de trabajo en RUP .....	18
Figura 7: Diagrama de Dominio .....	26
Figura 8: Diagrama de Casos de Uso del Sistema.....	31
Figura 9: Diagrama de Clases del Diseño del CU Gestionar Relación. ....	40
Figura 10: Diagrama de Componentes del CU Gestionar Relación.....	42

# *Introducción*

Desde el comienzo de la humanidad el hombre sintió la necesidad de ubicarse respecto al contorno geográfico en que se encontraba, siendo vital para la supervivencia conocer los terrenos en los que se encontraba y comunicar a través de distancias y direcciones los conocimientos sobre dónde se localizaban las regiones de factibilidad para la caza, la pesca o la ubicación de sus enemigos. Con el transcurso de los años y el incremento de los conocimientos, ante los descubrimientos territoriales, surgió la necesidad de representar, por su utilidad, las superficies y los ángulos de territorios para su posterior utilización, dando paso al surgimiento de los mapas y con ello una nueva ciencia: “la cartografía”.

Además de plasmar en los mapas las características de las regiones, a medida que fue evolucionando, el propio hombre necesitó documentar en ellos distintos tipos de información referente a las actividades humanas, así como proteger la misma para su perduración y manejo. Debido a esta necesidad el estudio y la superación constante de la humanidad se ha encargado de buscar soluciones que realicen tareas enfocadas a la vinculación de la información y los elementos de un mapa en una misma representación geográfica, con el objetivo de disminuir el esfuerzo del trabajo.

En la actualidad, la informática y las nuevas tecnologías han logrado un desarrollo significativo debido a las facilidades de automatización y administración de la información que proveen. Esto ha propiciado su uso en la rama de la cartografía, permitiendo la utilización de ordenadores y sistemas informáticos para el almacenamiento y acceso a mayor cantidad de información, en menor tiempo y con mayor calidad.

Estos sistemas informáticos dedicados al manejo, análisis y representación de la información geográfica son denominados *Sistemas de Información Geográfica* (SIG), y son utilizados en múltiples esferas de la vida social, económica y política. Se caracterizan por gestionar grandes volúmenes de información, utilizándose bases de datos especializadas en el manejo de contenidos y consultas espaciales. Hoy día, los SIG son cada vez más utilizados a nivel mundial producto del vertiginoso avance de la humanidad y la necesidad de hacer uso de toda la información generada en este proceso. Cuba, inmersa en la industria del software, ha comenzado a incursionar en el uso y producción de estos sistemas.

La Universidad de las Ciencias Informáticas (UCI) una de las mayores productoras de software del país posee el Centro de Geoinformática y Señales Digitales (GEySED) y dentro de este el Departamento de Geoinformática, el cual tiene como misión desarrollar productos, servicios y soluciones informáticas en el campo de la Geoinformática, contribuyendo a la formación integral de profesionales que respondan a las necesidades del progreso científico técnico y socioeconómico, permitiendo un posicionamiento en el mercado nacional (GEOINFORMÁTICA 2011).

La línea de desarrollo de software, SIG-DESKTOP, perteneciente al mencionado Departamento de Geoinformática tiene como objetivo fundamental desarrollar productos y soluciones informáticas en ambiente de escritorio que sirvan de apoyo a la toma de decisiones y contribuyan al seguimiento y control de la información socioeconómica de cualquier sector de la sociedad a partir de su representación y análisis espacial. GeoQ es un resultado concreto de lo anterior. Llevado a cabo por el grupo de desarrollo de la línea, se erige como un SIG para ambientes de escritorio y cuenta con varias funcionalidades que apoyan la toma de decisiones y contribuyen al manejo y análisis de la información geográfica y socioeconómica definida para un negocio específico.

No obstante lo anterior, existen múltiples empresas que no manejan la información socioeconómica de conjunto con la información geográfica, siendo muy común su correspondiente gestión en partes separadas. La versión actual de este SIG no incluye la posibilidad anterior, pues gestiona únicamente la parte geoespacial de la información, no admite la inclusión de datos socioeconómicos que se encuentren almacenados en algún formato, y que puedan o no estar asociados a una cartografía, lo cual limita el análisis de información geográfica y socioeconómica en las mencionadas empresas nacionales e internacionales y la usabilidad del propio sistema, pues sus usuarios estarían forzados a introducir los datos manualmente, aumentando considerablemente la posibilidad de errores.

Aún cuando lo anterior se materialice como una mejora del sistema, la situación no queda resuelta completamente pues GeoQ realiza solamente análisis muy simples de la información geográfica asociada a la cartografía lo que limita su análisis en algunas áreas de negocios donde el contenido de los datos a analizar pudiera ser complejo o el volumen de información fuera muy grande.

A partir de la situación problemática anterior se define el siguiente problema a resolver ***¿Cómo disminuir las limitaciones de usabilidad y de análisis de información geográfica y socioeconómica del Sistema de Información Geográfica GeoQ?***

Se define como **objeto de estudio** de la investigación el Sistema de Información Geográfica GeoQ, siendo el **objetivo general** desarrollar un módulo para la interrelación de múltiples tablas y capas que contribuya al análisis avanzado de la información geográfica y socioeconómica y aumente la usabilidad del sistema GeoQ. Derivado del objeto y el objetivo general se delimita la automatización de la interrelación de tablas y capas en GeoQ como el **campo de acción** de la investigación.

Para cumplir con el objetivo general propuesto se desarrollan una serie de tareas de la investigación que serán descritas a continuación:

1. Caracterizar el estado del arte asociado a los SIG en ambiente de escritorio.
2. Caracterizar el SIG GeoQ.
3. Valorar las soluciones existentes que resuelven el problema parcial o totalmente.
4. Fundamentar la utilización de herramientas, tecnologías y lenguajes a utilizar en el desarrollo de la propuesta de solución.
5. Realizar el análisis diseño e implementación de la solución según la metodología de desarrollo de software seleccionada.

Los autores de la investigación **defienden** la siguiente **idea**: el desarrollo de un módulo para la interrelación de múltiples tablas y capas contribuye a disminuir las limitaciones de usabilidad y de análisis de información geográfica y socioeconómica del SIG GeoQ.

En el transcurso de la investigación se emplean los métodos teóricos que se describen a continuación:

**Histórico-Lógico:** Se utiliza en la realización de estudios previos de SIG, enfocándose fundamentalmente en la interrelación de tablas y capas, apoyándose en los principales conceptos que relacionan los elementos mencionados anteriormente.

**Analítico-Sintético:** Se emplea para comprender la teoría y las definiciones vinculadas a la interrelación de múltiples tablas y capas, permitiendo obtener los elementos fundamentales para formular una propuesta de solución a la situación planteada.

La investigación se estructura de la siguiente forma:

**Capítulo 1:** En este capítulo se especifican y se explican los principales conceptos asociados a los SIG. Se caracteriza el sistema GeoQ como objeto de estudio y se valoran y critican las soluciones actuales que de alguna manera resuelven el problema de la investigación.

**Capítulo 2:** En este capítulo se definen, argumentan y valoran las principales herramientas, tecnologías, metodologías, lenguajes y arquitecturas que se utilizan para la construcción de la solución.

**Capítulo 3:** En este capítulo se comienza la construcción de la solución según la metodología de desarrollo seleccionada, se definen los requerimientos funcionales y no funcionales y se describen los Casos de Uso del Sistema.

**Capítulo 4:** En este capítulo se concluye la construcción de la solución y se especifican los artefactos referidos a las etapas de diseño, implementación y pruebas.

# Capítulo 1: “Fundamentación Teórica”

## 1.1 Introducción

En este capítulo se describen los principales aspectos que fundamentan la investigación como parte del aseguramiento teórico, se caracteriza el estado actual de los SIG y los conceptos más significativos. Se define el Sistema de Información Geográfica GeoQ y, finalmente, se valoran las posibles soluciones existentes que responden al problema planteado de alguna manera en la actualidad.

## 1.2 Conceptos asociados al dominio del problema

### 1.2.1 Aplicaciones informáticas

La Real Academia Española define el concepto de aplicación informática como un *programa preparado para una utilización específica, como el pago de nóminas, formación de un banco de términos léxico* (ESPAÑOLA 2011f).

Por otra parte, (ALEGSA 2007) lo considera un programa informático que permite a un usuario utilizar una computadora con un fin específico. Las aplicaciones son parte del software de una computadora, y a menudo, suelen ejecutarse sobre el sistema operativo del que se dispone. Una aplicación de software suele tener un único objetivo, por ejemplo: navegar en la web, revisar el correo, explorar el disco duro, editar textos, jugar, (un juego es un tipo de aplicación), editar fotos o videos familiares, entre otros.

Los autores de esta investigación definen **aplicación informática** como un programa que ejecuta acciones en un ordenador para facilitar el trabajo del usuario en un entorno determinado de la ciencia la técnica o la sociedad.

### 1.2.2 Gestión de la información

Del latín *gestio*, el concepto de gestión hace referencia a la *acción y al efecto de gestionar o de administrar. Gestionar es realizar diligencias conducentes al logro de un negocio o de un deseo cualquiera* (DEFINICIÓN.DE 2011).

El diccionario de la lengua española define la gestión como *el conjunto de trámites que se llevan a cabo para resolver un asunto* (ESPAÑOLA 2011c). Se considera que la gestión es la actividad o tarea que se encarga de la administración, tratamiento y manipulación de los recursos para solucionar una situación.

El Diccionario de la Real Academia de la Lengua Española plantea que la información es la *acción y efecto de informar o informarse* (ESPAÑOLA 2011g). La información es el análisis o experiencia que el hombre obtiene sobre el entorno en el que se desarrolla, son datos procesados según el significado de las cosas.

En la actualidad la información cada vez es mayor, el desarrollo de las tecnologías ha permitido facilitar la gestión, tratamiento, manipulación y conservación de la misma. Los datos son un aspecto fundamental en el avance de algunas organizaciones, los cuales se pueden obtener de diferentes fuentes, gestionando el acceso de estos sobre los usuarios para lograr proteger los activos que se encuentran en una entidad.

En las instituciones se aplica la gestión de la información integrando algunas funciones como: el establecimiento de políticas, aplicación de estrategias de seguridad, y la aplicación correcta de la información con el fin de automatizar los procesos que incrementan el control de los sistemas.

Se define en el contexto de esta investigación la **gestión de la información** como el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de la organización; es la actividad que se encarga de manipular y administrar datos mediante el uso de sistemas informáticos, con lo cual el hombre obtiene provecho para solucionar situaciones.

### **1.2.3 Información socioeconómica**

La Información económica es el *conjunto de datos que proporcionan los elementos necesarios para medir y evaluar las repercusiones de las políticas de ingreso, gasto y endeudamiento públicos en el contexto económico y social del país* (DEFINICIÓN.ORG 2012).

Los autores de esta investigación consideran que información socioeconómica es el resultado del análisis de los conocimientos obtenidos en la práctica para evaluar los efectos en el ámbito político, económico y social de un país.



### 1.2.4 Información geográfica

La Información geográfica o espacial *describe la ubicación física de los objetos y la relación entre los objetos. La industria de la información espacial es un componente especializado del sector de tecnologías de la información más amplia y tiene vínculos con muchas otras disciplinas tales como la planificación, la gestión de los recursos naturales, ingeniería y servicios de salud. La información espacial se utiliza para desarrollar aplicaciones tan diversas como los sistemas de gestión de incendios forestales, servicios de ambulancia de despacho y los planes de construcción de la encuesta (ESPATIAL 2007).*

Los autores consideran que la información geográfica es un tipo de información específica la cual posee un componente espacial definido en un sistema de coordenadas determinado. Constituye una fuente de información para el usuario con el objetivo de aumentar su conocimiento geográfico, lo cual provoca la satisfacción ante determinadas necesidades sobre el análisis de un territorio, ya que la información puede ser consultada, transferida, transformada, superpuesta o procesada en cualquier momento.

### 1.2.5 Cartografía

Un **mapa** es una *representación geográfica de la Tierra o parte de ella en una superficie plana / Es la representación geográfica de una parte de la superficie terrestre, en la que se brinda información relativa a una ciencia determinada / Mapa lingüístico, topográfico, demográfico (ESPAÑOLA 2011h).* Los mapas permiten conocer cómo se encuentra estructurado el espacio geográfico, lo cual contribuye al proceso de análisis y comprensión del mismo por el ser humano.

Según el Diccionario de la Lengua Española, cartografía se define como *el arte de trazar cartas cartográficas (ESPAÑOLA 2011a).* La International Cartographic la define como *el arte, la ciencia y la técnica de hacer mapas y su estudio tanto en documentos científicos como en obras de arte (AGRÍCOLAS 1975).*

La cartografía es la *ciencia que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales y demás.* Por extensión, también se denomina cartografía a un *conjunto de documentos territoriales referidos a un ámbito concreto de estudio (AGUASCALIENTES 2012).* En el marco de esta investigación se considera que el concepto anterior brinda todos los elementos necesarios asociados al término cartografía.

### 1.2.6 Base de Datos Espacial

La Lic. Rosa María Mato García señala como una base de datos al *conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. Puede considerarse una colección de datos variables en el tiempo* (GARCÍA 2009).

Por otra parte, una base de datos es un *conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada* (DATE 2001).

Según Alexander Rodríguez Torres los datos espaciales son *la información del mundo real traducida en capas temáticas. Un modelo de datos geográficos es una abstracción del mundo real que utiliza un conjunto de objetos dato, para dar soporte al despliegue de mapas, consultas, edición y análisis* (TORRES 2009).

*Una base de datos geográfica o espacial es una colección de datos organizados de tal manera que sirvan efectivamente para una o varias aplicaciones SIG. Esta base de datos comprende la asociación entre sus dos principales componentes: datos espaciales y atributos o datos no espaciales* (TORRES 2009).

Las bases de datos espaciales son bases de datos con una componente geográfica utilizadas para modelar la información geo-referenciada procedente del mundo real de forma que pueda ser empleada por SIG, esta abstracción es lograda a partir de la representación de capas.

### 1.2.7 Consultas Espaciales

Según el diccionario de la Lengua Española una consulta es la *búsqueda de datos que se realiza en un libro, periódico o fichero para informarse sobre un asunto* (ESPAÑOLA 2011b). Por lo anteriormente definido se plantea que las consultas espaciales son el método que permite acceder a la información existente en una base de datos espacial para analizar y gestionar la misma.

### 1.2.8 Interrelación de tablas y capas

Para lograr una definición de interrelación se considera necesario comprender el significado del concepto relación. Según la Real Academia Española **relación** no es más que: *Conexión, correspondencia de algo con otra cosa* (ESPAÑOLA 2011e).

Sobre interrelación, la Real Academia Española considera que se refiere a: *Relación mutua entre personas, cosas o fenómenos* (ESPAÑOLA 2011d).

Los autores definen la interrelación entre dos tablas y capas como la interconexión entre ellas mediante la información que se encuentra almacenada en las mismas, se logra a través de un identificador único, de esta forma cualquier cambio en una tabla o capa, afecta en alguna medida a la otra.

### 1.2.9 Sistemas de Información Geográfica

El Centro Nacional para el Análisis de la Información Geográfica (NCGIA) define un SIG como un *Sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión* (ALCALÁ 2001).

Por su parte, Rodríguez Pascual los define como el *conjunto integrado de medios y métodos informáticos, capaz de recoger, verificar, intercambiar, almacenar, gestionar, actualizar, manipular, recuperar, transformar, analizar y mostrar datos espaciales referenciados a la Tierra*.

Otro concepto de Pascual lo define como un *modelo informatizado del mundo real, descrito en un sistema de referencia ligado a la Tierra, establecido para satisfacer necesidades de información respondiendo a un conjunto de preguntas concretas*.

Finalmente Huxhold y Levishohn los consideran una *colección de tecnología de la información, datos y procedimientos de captación de información, almacenamiento, manipulación, análisis y presentación en mapas y estadísticas sobre características que pueden ser representadoras en mapas* (SANTOS Sin año).

Los autores consideran que un SIG se compone de la unión de hardware, software e información con una ubicación en el espacio.

Constituyen una herramienta que permite a los usuarios crear, consultar, editar y analizar datos de forma espacial sirviendo de apoyo a la toma de decisiones y contribuyendo al seguimiento y control de la información socioeconómica de cualquier sector de la sociedad a partir de su representación geográfica y su análisis.

## Formas de representación de la información geográfica

La información geográfica es un elemento esencial en el funcionamiento de los SIG, teniendo en cuenta la forma en que se desarrollan se hace necesario realizar una modelación de los datos existentes, para esto se cuenta con dos modelos: modelo vectorial y modelo raster.

El modelo vectorial está definido por tres elementos: puntos, líneas y polígonos, se utilizan para representar fenómenos de un territorio, basándose en las normas topológicas y la localización de este.

- Los puntos representan la ubicación de una zona a pequeña escala, transfieren poca información y no permiten realizar mediciones.
- Las líneas representan los fenómenos lineales en un territorio como caminos, calles y ríos, permiten medir distancias entre dos puntos.
- Los polígonos representan elementos territoriales de un área en específico como provincias, ciudades, municipios, y edificios, transmiten mayor información que las líneas y los puntos, operaciones como la medición de perímetros y áreas de regiones se pueden realizar con estos.

El modelo raster se utiliza para almacenar, procesar y visualizar datos geográficos, representa la realidad en una malla o rejilla donde cada uno de los elementos tiene un orden implícito en esta. Está orientado a las propiedades del espacio más que a la precisión de la localización.

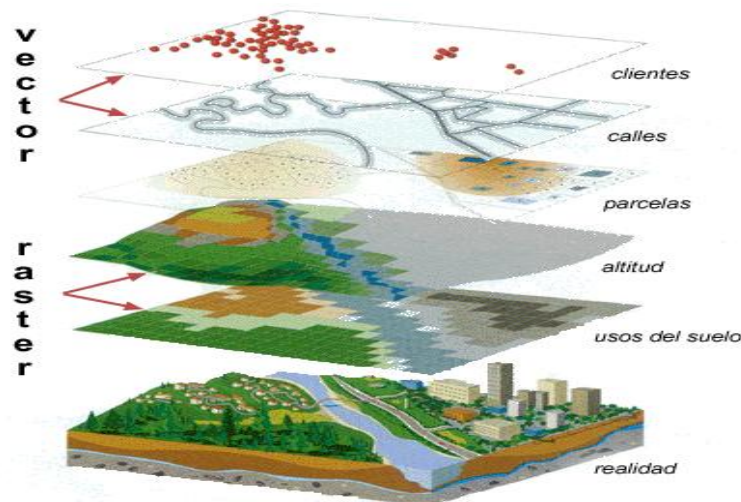


Figura 1: Clasificación de las capas.

## **Funcionamiento de los SIG**

En el funcionamiento de los SIG, las bases de datos con información geográfica están enlazadas por un dato en común a un mapa digital, constituyen un elemento esencial ya que permiten que se obtenga información de una región especificada por el usuario.

### **Fases del funcionamiento de un SIG**

1. Entrada de datos: toda la información que se maneja en el sistema puede encontrarse de manera digitalizada o no, esta información está clasificada en datos vectoriales o raster.
2. Almacenamiento y actualización de las bases de datos geográficamente: como aspecto primordial se geo-referencia la información existente mediante coordenadas de latitud y longitud, es la forma de estructurarla en el SIG, teniendo en cuenta la relación de los datos con su ubicación geográfica.
3. Análisis e interpretación de los datos geo-referenciados: genera una nueva información tomando como base el análisis y la interpretación de los datos previamente geo-referenciados.
4. Salida de la información en forma de productos diferentes: depende de los requisitos que necesite el usuario (ANDALUCÍA 2010).

## **1.3 Objeto de Estudio**

### **1.3.1 Descripción General**

GeoQ es un producto de la línea SIG-Desktop que surge en el contexto productivo del Centro de Desarrollo GEySED en el año 2010, nace a partir de la necesidad de disponer de una aplicación informática en ambiente de escritorio que realizara funcionalidades asociadas a los Sistemas de Información Geográfica para la comercialización y desarrollo de soluciones ajustándose a las exigencias de futuros negocios.

El sistema está desarrollado sobre los lenguajes de programación Python y C++, puede ser ejecutado en los sistemas operativos (SO) GNU/Linux en su distribución Ubuntu v9.10, Windows XP-SP2 y Mac OSX. Brinda servicios tales como escala de zoom, panel de leyenda, gestión de apariencias. Para el manejo de datos genera un conjunto de capas (colores, símbolos), que son utilizadas por el usuario en la realización de consultas, de análisis y representación geo-espacial de la información referente a un determinado negocio.

Se encuentra bajo la autoridad Pública GNU/GPL versión 3 (General Public License), trabaja con información de diferentes formatos (SHP, KML, TAB, TIFF), permite realizar un conjunto de acciones como: crear, editar y eliminar nuevas capas y generar, imprimir y exportar mapas. El sistema GeoQ presenta una arquitectura de plugins extensible por lo que es configurable según los requerimientos que se definan en un momento determinado (GEOQ 2010).

### **Funcionalidades de GeoQ:**

1. Vector de superposición y de datos de trama en diferentes formatos y proyecciones, sin conversión a un formato interno o común. Este vector soporta los siguientes formatos:
  - Espacio dispuesto para las tablas PostgreSQL con PostGIS y Spatialite, los formatos de tipo vectorial cuentan con el apoyo de la biblioteca OGR9.
  - Formatos de mapa de bits con el apoyo de la biblioteca de GDAL10, ejemplo: modelos de elevación digital, la fotografía aérea, lugares de GRASS11 y mapsets (LORES 2011).
2. Crear mapas y explorar interactivamente los datos espaciales con una interfaz gráfica de usuario. Muchas de las herramientas disponibles en la interfaz gráfica de usuario incluyen:
  - Compositor de impresión.
  - Panel de vista.
  - Marcadores espaciales.
  - Identificar y seleccionar las características.
  - Editar / Ver / atributos de búsqueda.
  - Características de etiquetado.
  - Superposición de diagrama de vectores.
  - Vector de cambio y la simbología de trama.
  - Añadir una capa de retícula.
  - Decorar el mapa con una flecha al norte, la barra de escala y la etiqueta de derecho de autor.
  - Guardar y restaurar los proyectos.
3. Crear, editar y exportar datos espaciales usando:
  - Herramientas para la digitalización de GRASS y formatos shapefile.
  - Herramientas de GPS para la importación y exportación en formato GPX, convertir otros formatos de GPS para GPX, o bajar / subir directamente a una unidad de GPS.
4. Realizar el análisis espacial utilizando el plugin para fToolsShapefiles o el plugin de GRASS integrado, incluyendo: álgebra de mapas, análisis del terreno, modelos hidrológicos y análisis de redes (LORES 2011).

## 1.4 Análisis de otras soluciones existentes

En el contexto nacional e internacional existen varios SIG que en alguna manera facilitan el procesamiento y visualización de la información geográfica, propiciando mayores garantías en el proceso de toma de decisiones.

A continuación se muestran los ejemplos de SIG más representativos de la afirmación anterior:

### 1.4.1 ArcGIS

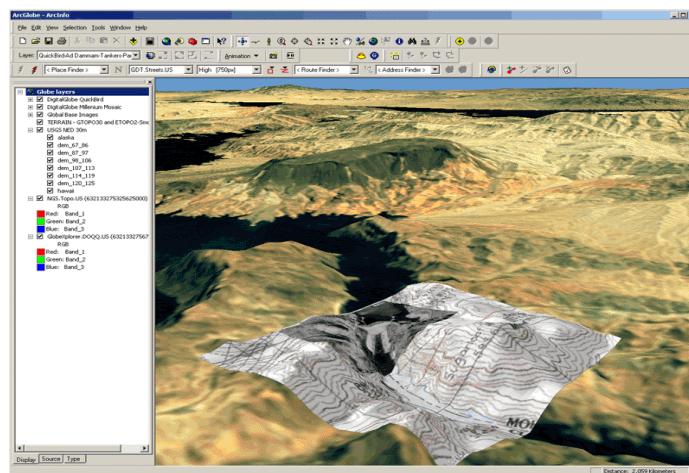


Figura 2: Interfaz de ArcGIS

ArcGIS constituye un producto para la creación, análisis y manipulación de la información geográfica, contiene diferentes aplicaciones por separadas aunque funcionalmente se encuentran integradas.

#### Módulos comprendidos en el formato ArcView de ArcGIS

**ArcMap:** Este es el módulo principal de ArcGIS, está basado en el trabajo con los mapas, el análisis, realización de consultas, visualización de los datos geográficos, edición de cartografía, importación y exportación a formatos CAD, EXCEL, ACCESS, TAB y DWG.

**ArcCatalog:** Está centrado en la organización y documentación de los datos geográficos. Mediante esta aplicación se tendrá la posibilidad de crear nuevas capas vacías y editarlas en ArcMap.

**ArcToolbox:** Está destinado a la realización de operaciones de conversión, edición y análisis de los datos, haciendo posible la creación de nueva información mediante el uso de herramientas que



contiene, permite, además, la sobre-posición de capas de información, conversión de datos, re-proyección, unión y separación de información (CENTER 2011).

### 1.4.2 GvSig

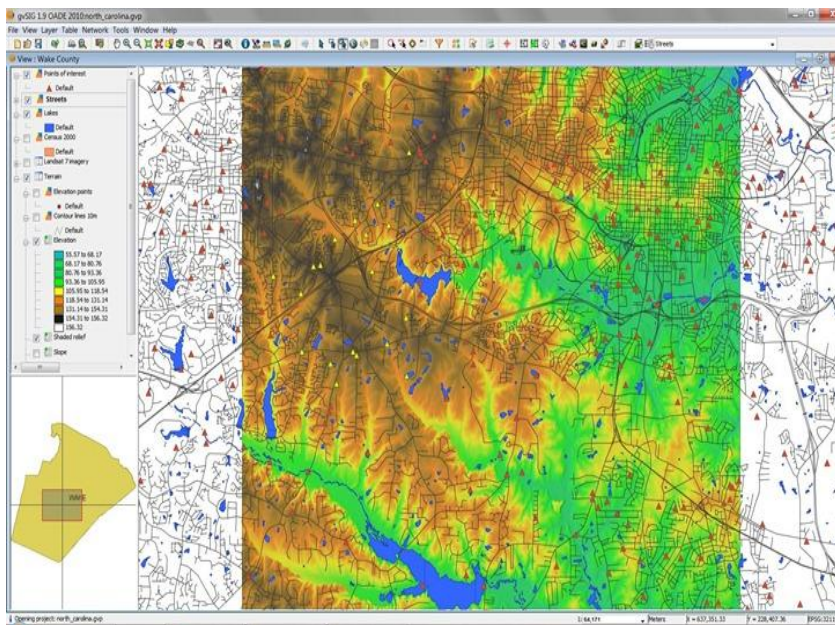


Figura 3: Interfaz de GvSig

GvSig es un SIG libre y de código abierto, maneja información geográfica vectorial y raster, soporta los formatos: GML, SHP, DXF, DWG, DGN, KML (vectoriales) y los formatos: MrSID, GeoTIFF, ENVI o ECW (raster). Puede integrar datos en el sistema a partir de servidores de mapas con especificaciones del OGC<sup>1</sup>, en otras palabras que contenga servicios de WMS (Web Map Service), WFS (Web Feature Service), WCS (Web Coverage Service), Servicio de Catálogo y Servicio de Nomencladores (GVSIG 2011).

#### Características principales de GvSig:

- Acceso a formatos vectoriales: pueden ser de diferentes formas como: SHP, GML, KML, DXF, DWG y DGN.
- Acceso a formatos raster: pueden ser de diferentes formas como BMP, GIF, TIF, TIFF, JPG, JPE y PNG.

<sup>1</sup> Consorcio internacional formado por un conjunto de empresas, agencias gubernamentales y universidades, tienen como fin la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica.



- Acceso a servicios remotos: OGC (WMS, WFS, WCS, WFS-T, WPS).
- Acceso a bases de datos y tablas: PostGIS, MySQL, ArcSDE, Oracle, JDBC y CSV.
- Selección: se realiza mediante puntos, rectángulos, polígonos, capas, atributos.
- Tablas: se realiza un conjunto de operaciones como: filtros, enlazar, unir, exportar e importar campos (GVSIG 2011).

### 1.4.3 La interrelación de múltiples tablas y capas en las soluciones anteriores

Los sistemas descritos anteriormente realizan una correcta gestión de la información geográfica, no obstante, no todos tienen integradas las funcionalidades para lograr la interrelación de múltiples tablas y capas de manera satisfactoria, relacionar tablas consiste en que los datos asociados no se anexan a la tabla de atributos de la capa como en el caso del *join*. Lo que sucede es que se puede acceder a los datos relacionados cuando se trabaje con los atributos de una capa.

#### En ArcGIS

La funcionalidad Relate del SIG ArcGIS es una potencia en este sistema ya que permite realizar grandes análisis de datos pues permite la interrelación entre múltiples tablas, permitiendo mantener relacionada la información en el momento que se está utilizando. La relación de tabla en ArcMap es bidireccional, esto implica que ambas tablas implicadas podrán utilizar la relación sin tener en cuenta qué tabla posee la relación.

Si dos capas de un mapa apuntan al mismo shapefile, ambas se ven afectadas y se puede utilizar una relación que posea una de ellas. Dados que las relaciones se pueden almacenar en un archivo de capa específico, una capa no puede eliminar una relación que sea propiedad de otra capa, incluso aunque ambas capas hagan referencia a los mismos datos (CENTER 2011). El sistema no se puede utilizar como propuesta de solución pues se encuentra bajo licencia propietaria Arcgis 9.3.

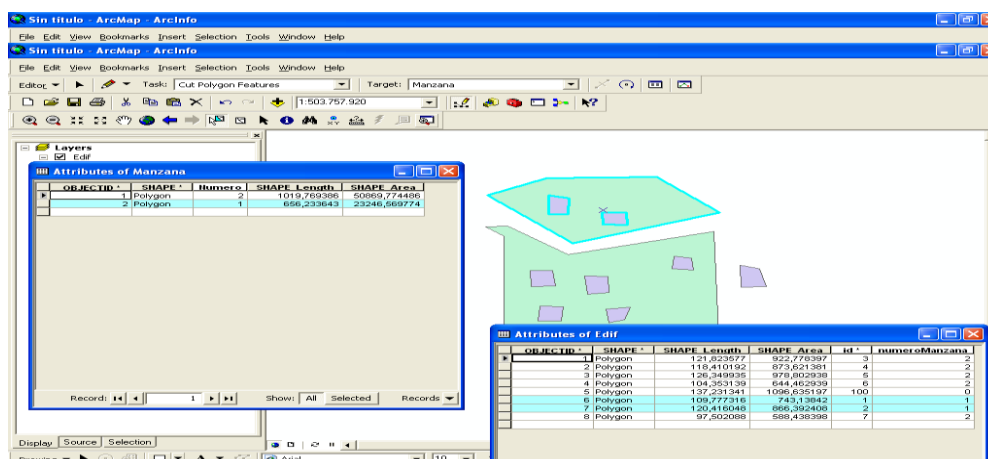


Figura 4: Interrelación de tablas ArcGIS

## En GvSig

Una de las acciones que se pueden hacer con el sistema GvSig es relacionar tablas, este proceso se manifiesta con un atributo en común, y trae como resultado que al seleccionar un campo de la tabla origen se seleccionan los campos que tienen en común con la tabla destino y de esta forma quedarían marcados los campos que se encuentran relacionados (relación de uno a muchos) (GVSIG 2011).

Este sistema no es conveniente utilizarlo para la propuesta de solución, pues se encuentra desarrollado en el lenguaje de programación Java y rompe con la arquitectura de GeoQ que se basa en el SIG Quantum GIS.

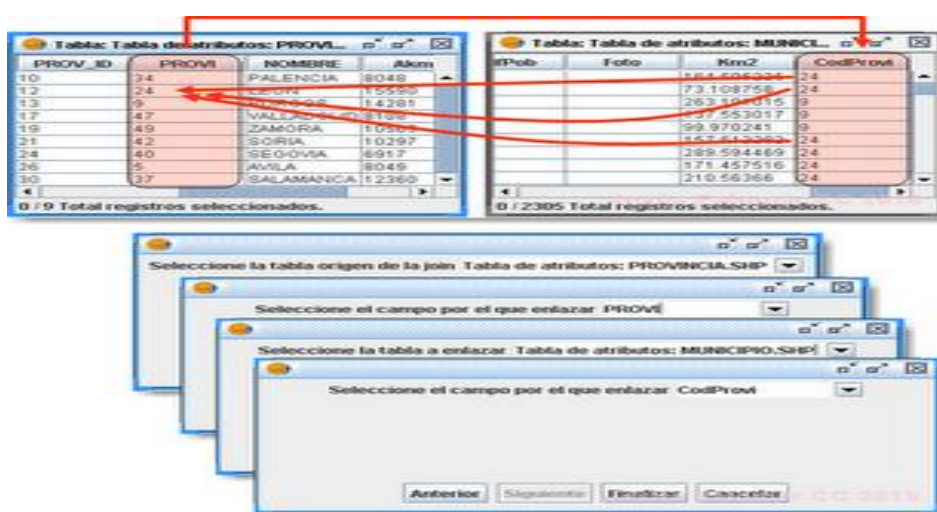


Figura 5: Interrelación de tablas GvSig.

### 1.5 Conclusiones Parciales

Luego de realizado el estudio del estado del arte en el área de los SIG y la interrelación de múltiples tablas y capas se concluye que:

1. Ninguno de los SIG analizados ofrecen una solución completa a la problemática planteada pues no realizan correctamente el proceso de interrelación de múltiples tablas y capas.
2. No obstante lo anterior, el SIG ArcGIS sí realiza el proceso de interrelación descrito pero su carácter propietario y el interés de la universidad de utilizar y desarrollar tecnologías libres permitió que se rechazara como posible solución.
3. El SIG ArcGIS sirvió de base teórica y conceptual para el desarrollo de la propuesta de solución.

## **Capítulo 2: “Tendencias, tecnologías y herramientas a utilizar”**

### **2.1 Introducción**

En este capítulo se definen, argumentan y valoran las principales herramientas, tecnologías, metodologías y lenguajes que se utilizan para la construcción de la solución.

### **2.2 Metodología de desarrollo de software**

Durante el desarrollo del ciclo de vida de un software, se deben realizar determinadas tareas para obtener como resultado final un producto con la calidad que se requiere. En este proceso, intervienen algunos componentes que pasan por diferentes fases o etapas, utilizando herramientas específicas y aplicando un conjunto de normas, ocasionando así el uso de una metodología de desarrollo de software las cuales “surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software”.

Las metodologías de desarrollo de software se clasifican en dos, las metodologías pesadas: se encuentran destinadas para el control de los procesos, definiendo con rigor todas las actividades que se desarrollarán, así como las herramientas que se utilizarán, y por otro lado las metodologías ligeras: se encuentran orientadas a la interacción del cliente y la evolución incremental del producto. Se muestran las versiones que se obtienen del software en cortos intervalos de tiempo, con el propósito de poder realizar cambios según las necesidades del cliente. Dentro de estas clasificaciones de metodología se pueden mencionar como ejemplos al Proceso Unificado de Desarrollo (RUP), como metodología pesada y Extreme Programming (XP), CRYSTAL y SCRUM como metodología ágiles (SCRIBD 2012a).

#### **El Proceso Unificado de Desarrollo (RUP) como metodología a utilizar**

Clasificándose dentro de las metodologías de desarrollo pesadas, es la más utilizada en el análisis, la implementación y documentación de sistemas orientados a objetos. Utiliza Lenguaje Unificado de Modelado (UML) para describir un sistema determinado, optimizando la productividad del equipo de desarrollo y proporcionando mejores resultados en el desarrollo del software basándose en la calidad del mismo. RUP basa su funcionamiento en tres características fundamentales, iterativo e incremental, centrado en la arquitectura y dirigido por casos de uso (SCRIBD 2012a).

Fases de desarrollo de RUP:

*Fase de inicio:* tiene como objetivo fundamental la comprensión de los requerimientos y límites del proyecto que se desarrollará. En ella se incluye al menos una arquitectura candidata, la fase de análisis y diseño, se estima el costo y el tiempo de creación del producto. Esta fase culmina con los objetivos del ciclo de vida.

*Fase de elaboración:* en esta fase se analiza el dominio del problema, desarrollando un plan de trabajo donde estén presentes las actividades necesarias del proceso. Se especifica el diseño de la arquitectura que evoluciona en conjunto con las iteraciones sucesivas hasta convertirse en el sistema final. Se contienen los casos de uso más críticos identificados en la fase de inicio.

*Fase de construcción:* en el desarrollo de esta fase es donde todos los componentes y requerimientos deben de ser implementados, integrados y probados con el objetivo de obtener una versión aceptada del proyecto.

*Fase de transición:* esta fase tiene como objetivo la entrega del producto a los clientes. Se incluyen una serie de acciones como entrenamiento, manufactura, soporte y mantenimiento del producto hasta lograr la satisfacción del cliente (SCRIBD 2012a).

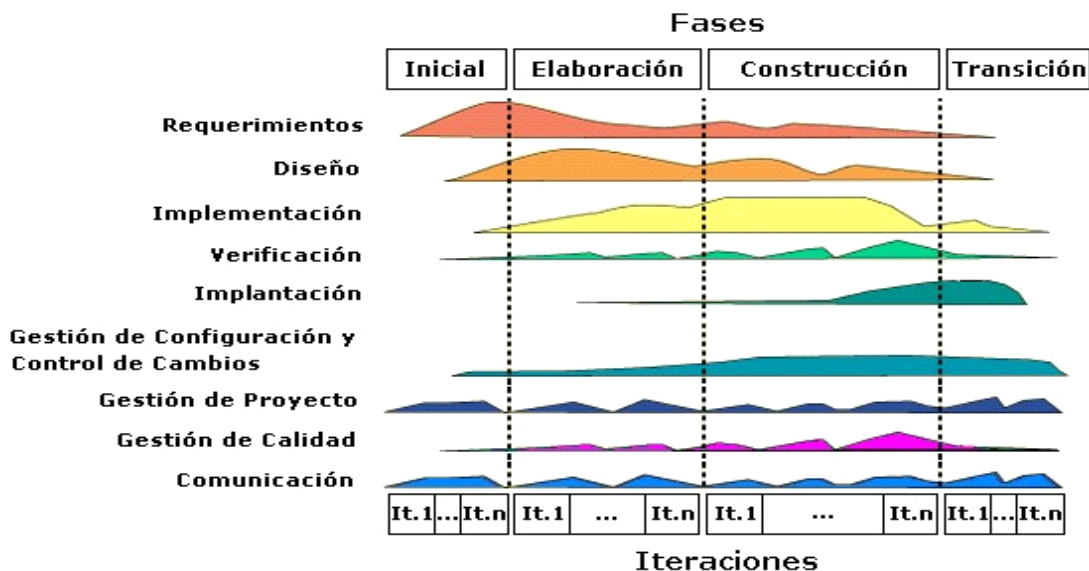


Figura 6: Fases y flujos de trabajo en RUP

### **2.3 Lenguaje de unificado de modelado: UML**

UML es un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Según sus autores: Jacobson, Booch y Rumbaugh es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso, más bien es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, hasta la implementación y configuración con los diagramas de despliegue.

Sus principios fundamentales son fáciles de entender y de aprender. Hoy día, es el lenguaje que complementa la ingeniería de software. No es utilizado solamente para la especificación de un sistema, además, se utiliza también para propósitos de comunicación entre el personal involucrado en el desarrollo de un sistema o para la documentación de software existente (BERCIAL 2006).

Una característica a resaltar es su flexibilidad. UML es extensible (es decir, capaz de mejorarse con nuevas características) e independiente de cualquier proceso de análisis y diseño orientado a objetos específico. Los modeladores de UML tienen la libertad de diseñar sistemas utilizando varios procesos, pero todos los desarrolladores pueden ahora expresar esos diseños con un conjunto de notaciones gráficas estándar (AGUASCALIENTES 2012).

### **2.4 Visual Paradigm como herramienta CASE<sup>2</sup>**

Una herramienta CASE se define como un conjunto de programas y ayudas que tienen como propósito asistir a los analistas, ingenieros de software y desarrolladores durante todo el ciclo de vida de un sistema. Realizar un buen diseño es un elemento esencial que la caracteriza, siendo una fuente importante para el desarrollo de un producto con la calidad requerida. Existen muchos tipos de CASE como: System Architect, SilverRun Rational Rose y una de las más utilizadas es la herramienta Visual Paradigm.

---

<sup>2</sup> Computer Aided Software Engineering (Ingeniería de Software Asistida por Computadora).

## **Visual Paradigm**

En el modelado de los diagramas se emplea Visual Paradigm For UML en su versión 8.0, porque es una herramienta potente y multiplataforma. Proporciona la construcción de aplicaciones con mayor calidad de manera rápida y eficiente. Contiene un conjunto de tutoriales y ayudas para la creación de sistemas informáticos. Visual Paradigm es una herramienta fácil de utilizar, apoya a la ingeniería reversa y la generación de UML del código.

Se encuentra destinada para soportar el ciclo de vida completo en el desarrollo de un sistema, teniendo como base la representación de los diferentes tipos de diagramas. Soporta las últimas versiones del mismo: Lenguaje Unificado de Modelado y la Notación y Modelado de Procesos de Negocio. Esta herramienta ayuda a los equipos de desarrollo de software a desplegar el proceso de evolución de un sistema. Contiene una conexión con a la herramienta Rational Rose en sus archivos de proyecto (.MDL / .CAT) los cuales pueden ser importados a Visual Paradigm UML (PRESSMAN 2001).

### Características de Visual Paradigm:

- Desarrollada bajo estándares de software libre.
- Una de sus licencias es gratuita.
- Se encuentra disponible en múltiples plataformas.
- Entorno de creación de diagramas para UML.
- Se puede realizar una ingeniería directa e inversa.
- Contiene un diseño centrado en los casos de uso.
- Fácil de instalar y actualizar.

## **2.5 Lenguaje de programación a utilizar**

En la informática, un lenguaje de programación es concebido como una herramienta empleada para indicar a una máquina determinadas acciones que se realizarán basándose en situaciones reales. Constituyen un medio de comunicación entre los usuarios y las máquinas. Se pueden citar C++, C# y Java como ejemplos de lenguajes.

## Lenguaje C++

C++, es un lenguaje de programación orientado a objetos. Surgió como resultado del trabajo realizado por Bjarne Stroustrup. Proviene del lenguaje C, donde en la década de los años 70 era empleado para la programación en sistemas Unix. Su creciente utilización propició una ampliación de sus posibilidades en el marco de la programación orientada a objetos.

### Ventajas que proporciona:

- Ofrece un conjunto de mejoras facilitando su usabilidad en aspectos tales como la entrada y salida de datos y definición de tipos.
- Es portable y multiplataforma, lo que significa que un programa escrito en este lenguaje se puede compilar en cualquier sistema operativo sin necesidad de cambiar el código fuente.
- Es versátil, esto significa que se pueden programar sistemas de diferentes niveles de complejidad.
- Es un lenguaje multi-nivel, por lo que brinda la posibilidad de ser utilizado para programar directamente el hardware, teniendo en cuenta el sistema operativo, además de programar para aplicaciones tipo Windows.
- Es un lenguaje libre y muy recomendable para la creación de aplicaciones de escritorio.
- Es muy utilizado, por lo que existen numerosos tutoriales y códigos fuentes abiertos que se pueden estudiar para aprender este lenguaje.
- Existen muchos algoritmos escritos en este lenguaje, cuyo pseudocódigo puede ser tomado y modificado a una solución determinada.
- Es didáctico, como consecuencia, sirve de base para aprender con facilidad otros lenguajes (SCRIBD 2012b).

## 2.6 Entorno Integrado de Desarrollo (IDE)

Un entorno de desarrollo integrado (del inglés *Integrated Development Environment* o *IDE*) es un programa compuesto por una serie de herramientas que utilizan los desarrolladores para escribir el código. Puede soportar varios lenguajes de programación o puede estar diseñada para uno únicamente. Existen muchos ejemplos de IDE lo constituyen: NetBeans, Delphi de Borland, MS Visual Studio .NET de Microsoft y Qt Creator (OVIEDO 1999).

## **Qt Creator**

Qt Creator es un Entorno de Desarrollo Integrado que permite la creación de aplicaciones complejas de escritorio en C++ de manera rápida. Se encuentra basado en la framework Qt, se caracteriza por ser abierto, gratuito y muy eficiente. Proporciona herramientas para el diseño y desarrollo de sistemas.

### Principales características de Qt Creator:

- Utiliza el lenguaje de programación orientado a objetos C++.
- Soporta otros lenguajes de programación como: C# / .NET, Python: PyQt y PySide, Ada, Pascal, Perl, PHP y Ruby.
- Es posible utilizarlo sobre múltiples entornos de trabajo: Linux, Mac OSX, Windows, Windows CE.
- Se encuentra distribuido bajo tres tipos de licencias: Commercial Developer License, Qt GNU LGPL v. 2.1 y Qt GNU GPL v. 3.0 (ATELIER 2009).

## **2.7 Sistema Gestor de Base de Datos**

Un sistema gestor de base de datos (SGBD) permite al usuario realizar un conjunto de acciones vinculadas al manejo de los datos contenidos en una base de datos, algunas de estas acciones son procesar, describir, administrar y recuperar. Emplea un lenguaje basado en consultas. Tiene como aspecto esencial la integración de los datos. Algunos ejemplos de SGBD son: MySQL, Oracle, MS SQL Server, PostgreSQL y su extensión PostGIS.

### **PostgreSQL**

Fue creado en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley. Es considerado el gestor de base de datos de código abierto más avanzado en la actualidad. Soporta un lenguaje SQL92/SQL3, integridad de transacciones, y extensibilidad de tipos de datos. Almacena no solo la información que se maneja en base de datos, tablas y columnas, sino también sobre funciones y métodos de acceso, esto es conocido como sistema de catálogo. Contiene un modelo de datos que está dado por una colección de relaciones con nombres, los cuales emplean atributos de un tipo específico. El sistema incorpora clases, herencia, tipos y funciones, brindándoles la posibilidad a los usuarios de extenderlo (LOCKHART 1996).



Ventajas que ofrece PostgreSQL:

- Proporciona la capacidad de almacenar información en la base de datos, la cual el usuario puede acceder a ella y actualizarla.
- Estabilidad
- Alto rendimiento
- Flexibilidad
- Gran compatibilidad con otros sistemas.
- Permite crear o migrar aplicaciones desde Access, Visual Basic, Visual C/C++ y Delphi para utilizarlo como servidor de base de datos (SCRIBD 2012c).

**2.8 Arquitectura de software**

La Arquitectura de Software(AS) es la organización fundamental de un sistema. Se representa a través de sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. Se concentra en requerimientos no funcionales, que son satisfechos mediante los modelos y diseños de la aplicación. Un diseño correcto de la Arquitectura del sistema es esencial para el éxito o fracaso del proyecto.

La AS no es el software operativo, en cambio, es una representación que permite que un ingeniero de software: 1) analice la efectividad del diseño para cumplir con los requisitos establecidos, 2) considere opciones arquitectónicas en una etapa en que aún resulta relativamente fácil hacer cambios al diseño y 3) reduzca los riesgos asociados con la construcción del software (PRESSMAN 2005).

Según David Garlan establece que la AS constituye un puente entre el requerimiento y el código, ocupando el lugar que en los gráficos antiguos se reservaba para el diseño (REYNOSO 2004).

Son muchas las definiciones redactadas por expertos del tema. Esta investigación se registró por la que aparece redactada en el documento de IEEE Std 1471-2000:

“La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución” (REYNOSO 2004).

### 2.8.1 Patrones Arquitectónicos

Teniendo como base el estudio realizado sobre las características principales de los patrones arquitectónicos existentes, se determina que es más recomendable aplicar patrones orientados a objetos y basados en componentes.

Se define como patrón arquitectónico a los que definen la estructura de un sistema software, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema Arquitectura Orientada a Objetos (PBWORKS 2009).

“Los componentes de un sistema encapsulan los datos y las operaciones que deben aplicarse para manejar los datos. La comunicación y coordinación entre componentes se consigue mediante el paso de mensajes” (PRESSMAN 2005).

#### Algunos principios claves de la arquitectura orientada a objetos

*Abstracción:* propiedad que permite representar las características esenciales de un objeto, sin preocuparse de las restantes características (no esenciales). Se encuentra centrada en la vista externa de un objeto, con el propósito de separar el comportamiento esencial de un objeto de su implementación.

*Encapsulamiento:* proceso de ocultar todos los secretos de un objeto que no contribuyen a sus características esenciales. Es definido como la capacidad de contener y controlar el acceso a un grupo de elementos asociados. En este principio se tendrá un control de los datos y procedimientos que son utilizados.

*Modularidad:* es la encargada de dividir la aplicación en pequeñas partes, donde cada una de ellas debe de funcionar de la forma más independiente posible, sin tener en cuenta la aplicación. Esta división se encuentra centrada en módulos que puedan ser ejecutados de manera separada.

*Polimorfismo:* permite referirse a objetos de clases diferentes mediante el mismo elemento de programa y realizar la misma operación de diferentes formas, según sea el objeto que se referencia en ese momento (LORES 2011).

## **Arquitectura Basada en Componentes**

La arquitectura basada en componentes consiste en una rama de la Ingeniería de Software en la cual se trata con énfasis la descomposición del software en componentes funcionales. Esta descomposición permite convertir componentes pre-existentes en piezas más grandes de software. Este proceso de construcción de una pieza de software con componentes ya existentes, da origen al principio de reutilización del software, mediante el cual se promueve que los componentes sean implementados de tal manera que permita posteriormente su utilización funcional sobre los diferentes sistemas en el futuro (LORES 2011).

### **2.9 Conclusiones Parciales**

1. Luego de conocer y evaluar las características más significativas de las herramientas, tecnologías y tendencias a utilizar, se asegura que sus elementos satisfacen las necesidades existentes para el desarrollo del módulo de interrelación de tablas y capas.
2. La utilización de las herramientas propuestas facilita y fortalece el trabajo aportando rapidez y eficacia a lo largo del ciclo de vida del desarrollo, garantizando una mayor usabilidad de la aplicación.
3. Las herramientas seleccionadas impulsan el desarrollo del software libre, muy a tono con el reclamo nacional y universitario actual.

## Capítulo 3: “Análisis de la solución propuesta”

### 3.1 Introducción

En este capítulo se comienza la construcción de la solución según la metodología de desarrollo seleccionada, se definen los requerimientos funcionales y no funcionales y se describen los Casos de Uso del Sistema.

### 3.2 Modelo de Dominio

Un modelo conceptual explica los conceptos significativos en un dominio del problema; es el artefacto más importante a crear durante el análisis orientado a objetos. Es una representación de conceptos del dominio del problema (LARMAN 1999).

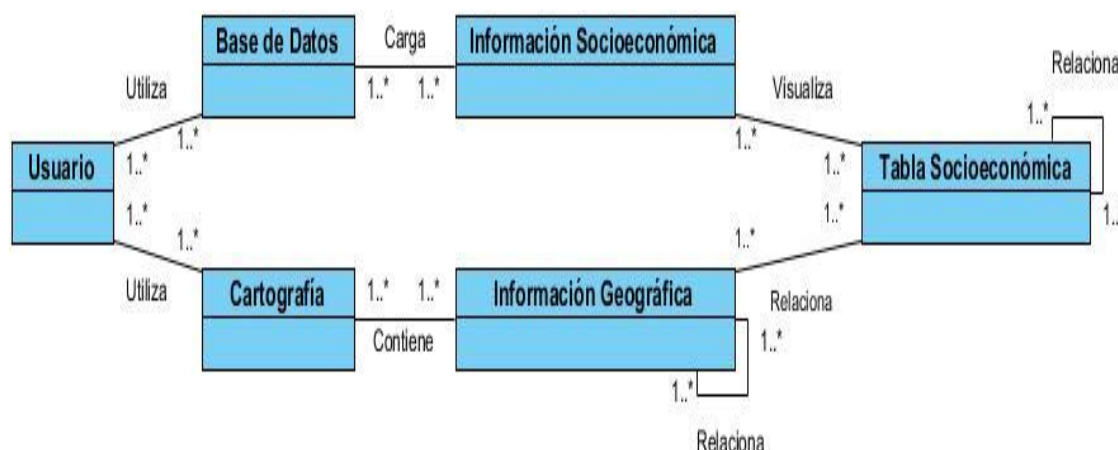


Figura 7: Diagrama de Dominio

#### 3.2.1 Descripción de las clases del dominio

**Usuario:** Cualquier persona que interactúa con el sistema.

**Base de Datos:** Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

**Información Socioeconómica:** Información asociada y referida a la actividad económica política o social de una región determinada.

**Cartografía:** Se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales.

**Información Geográfica:** Información que tiene asociada una referencia en un mapa.

**Tabla Socioeconómica:** Entidad que contiene datos sin referencia geográfica.

### 3.3 Requisitos

#### 3.3.1 Requisitos Funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer y dependen del tipo de software que se desarrolle. Los requisitos funcionales describen con detalle la función del sistema, sus entradas, salidas y excepciones (SOMMERVILLE 2005b).

El Sistema debe ser capaz de:

1. *Crear una nueva relación:* se debe poder crear una relación con cardinalidad de uno a muchos entre tablas y/o capas a partir de parámetros solicitados: tabla o capa con la que se relaciona, campo de tabla o capa con la que se relaciona, campo de la tabla o capa que se le realiza la relación, nombre de la relación.
2. *Eliminar una relación existente:* se debe poder eliminar permanentemente una relación existente de la lista de relaciones.
3. *Modificar una relación existente:* se debe poder modificar una relación existente de la lista de relaciones, cambiándole cualquiera de los campos que componen la relación.
4. *Listar las relaciones creadas:* se debe poder visualizar el listado de las relaciones una vez creadas, de la tabla socioeconómica o la capa.
5. *Especificar parámetros de la conexión a la base de datos en PostgreSQL:* se debe poder ejecutar una conexión a la base de datos a partir de parámetros solicitados: nombre de la conexión, servidor, puerto, nombre de usuario y contraseña.
6. *Modificar una conexión existente:* se debe poder modificar una conexión existente, cambiándole cualquiera de los campos que componen la conexión.
7. *Eliminar una conexión existente:* se debe poder eliminar permanentemente una conexión existente.
8. *Agregar una tabla socioeconómica de una Bases de Datos PostgreSQL:* se debe poder agregar la o las nuevas tablas con información socioeconómica al sistema proveniente de una Base de Datos en PostgreSQL.
9. *Visualizar una tabla añadida al sistema:* una vez añadida la tabla debe ser visualizada en el sistema.
10. *Mostrar los elementos de una tabla socioeconómica:* se debe poder mostrar la información correspondiente a una tabla seleccionada por el usuario.
- 10.1. *Seleccionar o deseleccionar todas las filas de la tabla socioeconómica:* se debe poder seleccionar o deseleccionar las filas seleccionadas en una tabla existente.

- 10.2. *Mover las filas seleccionadas hacia arriba del todo en la tabla socioeconómica:* se debe poder ubicar en las n primeras filas las n filas seleccionadas.
- 10.3. *Invertir la selección de las filas en la tabla socioeconómica:* se debe poder invertir la selección realizada, de manera que una vez accedida a la opción queden seleccionadas las filas que no lo estaban y deseleccionadas las que sí.
- 10.4. *Conmutar a modo de edición la tabla socioeconómica:* se debe poder modificar la información de la fila seleccionada en la tabla por cualquier valor válido.
  - 10.4.1. *Agregar una nueva columna a la tabla socioeconómica:* se debe poder agregar una nueva columna a la tabla.
  - 10.4.2. *Borrar una columna de la tabla socioeconómica:* se debe poder eliminar una columna existente en la tabla.
  - 10.4.3. *Abrir calculadora de campo:* se debe poder realizar cálculos con la información de la tabla socioeconómica.
- 10.5. *Buscar elementos dentro de la tabla socioeconómica:* se debe poder buscar un valor según un atributo especificado.
- 10.6. *Realizar una búsqueda avanzada dentro de la tabla socioeconómica:* se debe poder realizar una búsqueda mediante una consulta SQL.
- 10.7. *Seleccionar las relaciones asociadas a la tabla socioeconómica o capa:* se debe poder mostrar un listado de las relaciones asociadas a la tabla de atributos para su selección.
- 10.8. *Visualizar las relaciones de la tabla socioeconómica o de la capa:* se debe poder visualizar una o más relaciones creadas por el usuario según la o las filas seleccionadas en la tabla de atributo.
- 10.9. *Mostrar solo las filas seleccionadas de la tabla socioeconómica:* se debe poder visualizar en la tabla de atributos solamente las filas seleccionadas por el usuario.
- 10.10. *Buscar solo en las filas seleccionadas de la tabla socioeconómica:* se debe poder realizar búsquedas solamente en las filas seleccionadas por el usuario.
11. *Eliminar tabla socioeconómica:* se debe poder eliminar permanentemente una tabla socioeconómica existente.
12. *Cambiar nombre de la tabla socioeconómica:* se debe poder especificar un nuevo nombre a la tabla socioeconómica existente.
13. *Realizar una consulta SQL a la tabla socioeconómica:* se debe poder ejecutar una consulta SQL según las operaciones especificadas.

### 3.3.2 Requisitos no Funcionales

Los requisitos no funcionales restringen el sistema en desarrollo y el proceso de desarrollo que se debe utilizar. Pueden ser requerimientos del producto, organizacionales o externos. A menudo están relacionados con las propiedades emergentes del sistema y, por lo tanto, se aplican al sistema (SOMMERVILLE 2005b).

Los requisitos no funcionales que se detallan a continuación fueron descritos a partir de la arquitectura definida en la línea de desarrollo donde se ejecuta esta investigación. La propuesta que se presenta se adecua completamente a las especificaciones desarrolladas en la mencionada línea.

#### **Usabilidad**

- Las funcionalidades principales del módulo serán mostradas mediante iconos para lograr un mayor reconocimiento por parte del usuario.
- Las interfaces del módulo deben permitir completar las acciones necesarias en la menor cantidad de clic.

#### **Eficiencia**

- El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento.
- Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación, en todos los casos, dependerán de la cantidad de información que se procese en cada momento.

#### **Requisitos de Licencia**

El módulo se encuentra desarrollado bajo la Licencia Pública General de GNU (GNU GPL): General Public License. El desarrollo de este módulo bajo esta licencia significa que es posible inspeccionar y modificar el código.

#### **Restricciones de diseño e implementación**

- El diseño será sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar las nuevas funcionalidades de módulo.
- Las funcionalidades deben ser altamente configurables y extensibles pues se desarrollarán sobre la Plataforma personalizable QGIS.

## Interfaz

### Interfaces de Usuario:

Las interfaces de usuario poseerán una apariencia profesional similar a la de GeoQ, con un diseño gráfico sencillo, además de ser intuitiva de forma que el usuario logre la interacción con las nuevas funcionalidades.

### Interfaces de Hardware para las PCs clientes:

Para el funcionamiento del módulo de interrelación de múltiples tablas y capas se requiere al menos 128 MB de memoria RAM, y, como mínimo 40 GB de disco duro y un procesador a 512 MHz.

### Interfaces de Software para las PCs clientes:

Las funcionalidades del módulo de interrelación de múltiples tablas y capas se pueden ejecutar en los sistemas operativos. GNU/Linux, Windows y Mac OS.

## 3.4 Descripción del sistema propuesto

El Modelo de Casos de Uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. Se puede contar con diferentes tipos de usuarios, donde cada uno se representa mediante un actor que define un rol en la aplicación. Los actores modelan el entorno del sistema (BERCIAL 2006).

### 3.4.1 Descripción de los actores del sistema

Los actores constituyen un tipo de usuario de un sistema, puede ser una persona, un sistema informático o una organización que se encuentre interactuando con el sistema (CHILE 2010).

Actor	Descripción
Usuario	Cualquier persona que interactúe con el sistema.

Tabla 1: Definición de los actores.

### 3.4.2 Diagrama de Casos de Uso del Sistema

Un diagrama de Casos de Uso del Sistema representa gráficamente los procesos y su interacción con los actores, por su parte, un Caso de Uso no es más que una tarea que debe poder llevarse a cabo



con el apoyo del sistema que se está desarrollando. Cada Caso de Uso debe detallarse, habitualmente mediante una descripción textual (CHILE 2010).

A continuación se presenta el Diagrama de Casos de Uso del Sistema de la solución propuesta:

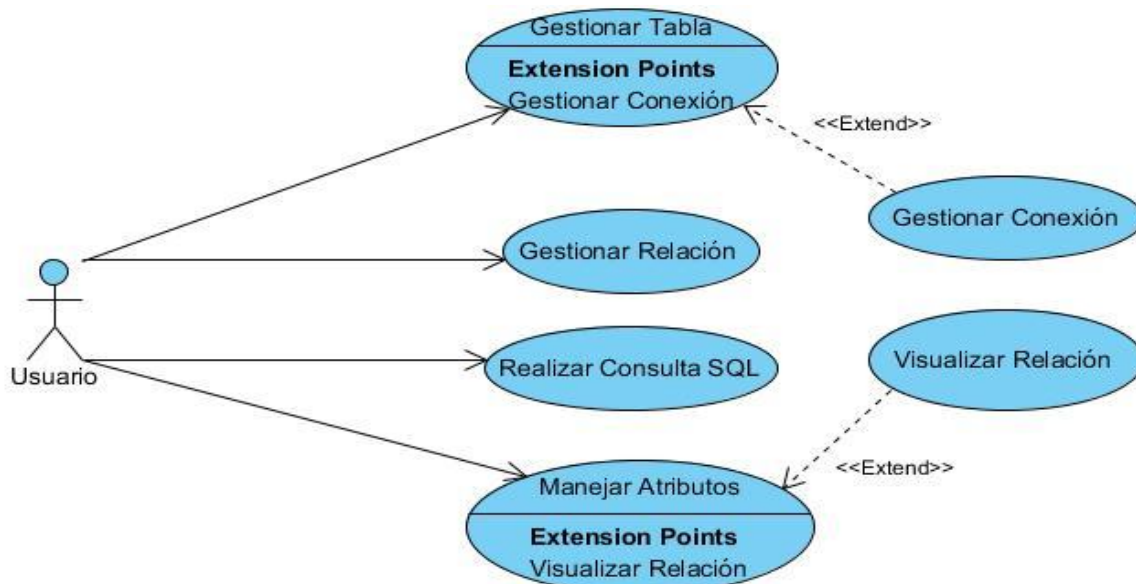


Figura 8: Diagrama de Casos de Uso del Sistema

### 3.4.3 Descripción textual de los Casos de Uso del Sistema

#### Descripción textual del Caso de Uso Gestionar Relación

<b>Caso de Uso:</b>	Gestionar Relación.
<b>Actores:</b>	Usuario.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción “Propiedades” de la capa, elige la opción “Relaciones”, y termina cuando el usuario cree, elimine o modifique una relación.
<b>Precondiciones:</b>	El sistema debe tener cargada más de una capa y/o más de una tabla.
<b>Referencias:</b>	RF1, RF2, RF3, RF4.
<b>Prioridad:</b>	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción “Propiedades de la capa”.	2. El sistema muestra la interfaz “Propiedades de la Capa” con las siguientes opciones: Estilo, Etiquetas, Campos, General, Metadatos, Acciones, Uniones, Diagramas y

	Relaciones.
3. El usuario selecciona la opción "Relaciones".	4. El sistema muestra la interfaz "Relaciones" que le permite al usuario realizar las siguientes acciones: a) Crear Relación, Modificar Relación o Eliminar Relación.
5. El usuario selecciona la opción deseada.	6. Si el usuario selecciona: Crear Relación. Ver Sección "Crear Relación". Modificar Relación. Ver Sección "Modificar Relación". Eliminar Relación. Ver Sección "Eliminar Relación".
7. El usuario selecciona la opción "Aceptar".	8. El sistema guarda las acciones realizadas, terminando así el Caso de Uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1.1. El usuario selecciona la opción "Relaciones" de una tabla socioeconómica.	1.2. El sistema ejecuta la acción 4 del Flujo Normal de Eventos y continúa normalmente con los puntos sucesivos.
Sección "Crear Relación"	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción "Crear Relación".	2. El sistema muestra la interfaz "Crear Relación" donde el usuario debe llenar los siguientes campos: <ul style="list-style-type: none"> <li>• Capa / tabla</li> <li>• Atributo de capa / tabla</li> <li>• Atributo seleccionado</li> <li>• Nombre de relación</li> </ul>
3. El usuario introduce los datos correspondientes y selecciona la opción "Aceptar".	4. El sistema crea la relación listándolas en la interfaz "Relaciones", terminando así el Caso de Uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1. El usuario no introduce el nombre de la relación y selecciona la opción "Aceptar".	3.2. El sistema muestra la interfaz "Campo vacío" con el mensaje de error "Debe introducir un nombre. La relación no será creada.", selecciona la opción "Aceptar".
	3.3. El sistema vuelve a la interfaz "Relaciones".
3.1. El usuario introduce los mismos	3.2. El sistema no crea la relación y muestra la interfaz

datos de una relación existente, con el campo nombre de la relación modificado y selecciona la opción "Aceptar".	"Conflicto de Relación" con el mensaje de error "Esta capa ya posee una relación con esa estructura. La relación no será creada" en caso que sea una capa. Si es una tabla muestra el mensaje de error "Esta tabla ya posee una relación con esa estructura. La relación no será creada.", selecciona la opción "Aceptar".
	3.3. El sistema vuelve a la interfaz "Relaciones".
3.1. El usuario introduce los datos de una relación donde el campo nombre de la relación es igual a una existente y selecciona la opción "Aceptar".	3.2. El sistema no crea la relación y muestra la interfaz "Conflicto de Relación" con el mensaje de error "Esta capa ya posee una relación con nombre "nombre de la relación". La relación no será creada" en caso que sea una capa. Si es una tabla muestra el mensaje de error "Esta tabla ya posee una relación con nombre "nombre de la relación". La relación no será creada.", selecciona la opción "Aceptar".
	3.3. El sistema vuelve a la interfaz "Relaciones".
4.1. El usuario selecciona la opción "Cancelar".	4.2. El sistema no crea la relación, terminando así el Caso de Uso.
<b>Sección "Modificar Relación"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la relación a modificar.	
2. El usuario selecciona la opción "Modificar Relación".	3. El sistema muestra la interfaz "Modificar Relación" con los datos de la relación a modificar.
4. El usuario modifica los datos deseados y selecciona la opción "Aceptar".	5. El sistema guarda los cambios, terminado así el Caso de Uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.1. El usuario no selecciona la relación a modificar.	
1.2. El usuario selecciona la opción "Modificar" y selecciona la opción	1.3. El sistema muestra la interfaz "Información" con el mensaje "Debe seleccionar una relación".

"Aceptar".	
	1.4. El sistema vuelve a la interfaz "Relaciones".
4.1. El usuario introduce los mismos datos de una relación existente, con el campo nombre de la relación distinto y selecciona la opción "Aceptar".	
	4.2. El sistema no modifica la relación y muestra la interfaz "Conflicto de Relación" con el mensaje de error "Esta capa ya posee una relación con esa estructura. La relación no será modificada" en caso que sea una capa. Si es una tabla muestra el mensaje de error "Esta tabla ya posee una relación con esa estructura. La relación no será modificada.", selecciona la opción "Aceptar".
	4.3. El sistema vuelve a la interfaz "Relaciones".
4.1. El usuario modifica el campo nombre de la relación con el mismo nombre de una relación existente y selecciona la opción "Aceptar".	
	4.2. El sistema no modifica la relación y muestra la interfaz "Conflicto de Relación" con el mensaje de error "Esta capa ya posee una relación con nombre "nombre de la relación". La relación no será modificada" en caso que sea una capa. Si es una tabla muestra el mensaje de error "Esta tabla ya posee una relación con nombre "nombre de la relación". La relación no será modificada.", selecciona la opción "Aceptar".
	4.3. El sistema vuelve a la interfaz "Relaciones".
4.1. El usuario modifica el campo nombre de la relación dejándolo vacío y selecciona la opción "Aceptar".	

	4.2. El sistema muestra la interfaz “Campo vacío” con el mensaje de error “Debe introducir el nombre de la relación. La relación no será modificada.”
	4.3. El sistema vuelve a la interfaz “Relaciones”.
4.1. El usuario selecciona la opción “Cancelar”.	4.2. El sistema no modifica la relación, terminando así el Caso de Uso.
Sección “Eliminar Relación”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la relación a eliminar.	
2. El usuario accede a la opción “Eliminar Relación”.	3. El sistema elimina la relación del listado de relaciones.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1.1. El usuario no selecciona la relación a eliminar.	
1.2. El usuario accede a la opción “Eliminar Relación”.	1.3. El sistema no elimina la relación y muestra la interfaz “Información” con el mensaje “Debe seleccionar una relación”, selecciona la opción “Aceptar”.
1.4. El usuario selecciona la opción “Aceptar”.	1.5. El sistema vuelve a la interfaz “Relaciones”.
<b>Poscondición:</b>	Una relación es creada, modificada o eliminada.

Tabla 2: Descripción textual del Caso de Uso Gestionar Relación.

El resto de las descripciones textuales de los Casos de Uso del Sistema pueden ser consultadas en el Anexo 1.

### 3.5 Conclusiones Parciales

Los autores concluyen:

1. Con la modelación del dominio y el sistema se logra un entendimiento común entre el equipo de desarrollo de manera que las etapas sucesivas del desarrollo de la solución pueden ser comprendidas y ejecutadas sin dificultad por los demás miembros.

2. La documentación generada producto de la realización de las actividades anteriores asegura la continuidad y la ampliación de la solución en algún momento posterior a esta investigación.
3. Los requisitos funcionales y no funcionales ubican al equipo de desarrollo en los objetivos de la solución y proporcionan una guía para su desarrollo y posterior validación.

## **Capítulo 4: “Implementación y pruebas de la solución”**

### **4.1 Introducción**

En este capítulo se concluye la construcción de la solución, se enuncian los principales patrones de diseño tenidos en cuenta y se especifican los artefactos referidos a las etapas de diseño e implementación, a saber, el diagrama de clases del diseño por cada Caso de Uso y el diagrama de componentes. Finalmente, se desarrollan los Diseños de Casos de Prueba necesarios para la validación de la propuesta.

### **4.2 Patrones de Diseño**

Craig Larman en su libro UML y Patrones describe un patrón de diseño como el amplio repertorio de principios generales y de expresiones formados por expertos que sirven de guía al crear un software. El patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas (LARMAN 1999).

Los patrones de diseño, de manera general, constituyen soluciones estándar que brindan respuesta a un problema común durante el diseño de un software. Una vez que se ha desarrollado el modelo de análisis, el diseñador puede examinar una representación detallada del problema que debe resolver y las restricciones que impone el problema (PRESSMAN 2005).

#### **Patrones generales de software para asignar responsabilidades (GRASP)**

En el diseño de la solución de la presente investigación se emplean los patrones que a continuación se describen:

**Experto:** Según Craig Larman el patrón denominado Experto consiste en que una clase debe poseer la información necesaria para cumplir con sus responsabilidades. Ofrece una analogía con el mundo real, consiste en asignarle responsabilidades a una clase que dispone de la información necesaria para llevar a cabo una tarea, expresa que cada objeto realice operaciones relacionadas con la información que posea.

Con este patrón se conserva el encapsulamiento, pues los objetos se valen de su propia información para hacer lo que se les pide. Soporta un bajo acoplamiento, esto contribuye a tener un sistema robusto y de fácil mantenimiento.

**Creador:** Plantea asignarle a una clase la responsabilidad de crear una instancia de otra siempre que se cumpla que una clase B contenga a una clase A, que exista una agregación entre ellas, que la clase B inicialice los datos de la clase A, o que la clase B registre la clase A.

Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se conecte con el objeto producido en cualquier evento. Al escogerlo como creador brinda soporte de bajo acoplamiento, lo que provee mayor reutilización.

**Bajo Acoplamiento:** Plantea asignar las responsabilidades de modo que se mantenga bajo acoplamiento, responde a la pregunta ¿Cómo dar soporte a poca dependencia y a una mayor reutilización?

El acoplamiento es una medida de la fuerza con una clase está conectada a otras clases con que las conoce y con que recurre a ellas. Una clase con bajo acoplamiento no depende de muchas otras.

El bajo acoplamiento estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca cambios negativos, como al realizar cambios en las clases afines pues se ocasionan cambios locales, es más difícil de entender una clase cuando se encuentra aislada, además de que su reutilización es más difícil porque se requiere la presencia de otras clases de las que depende.

**Alta Cohesión:** Plantea asignar las responsabilidades de modo que se mantenga una alta cohesión. Responde a la pregunta ¿Cómo mantener controlable la complejidad?

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza una clase con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

El patrón alta cohesión define que una clase tiene responsabilidades moderadas en un área funcional moderada con las otras para llevar a cabo las tareas. Una clase que presente este patrón garantiza que sea más fácil darle mantenimiento, conservarla, entenderla y reutilizarla.

Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización.



La asignación de responsabilidades constituye la esencia de un buen diseño de clases. Los patrones no deben verse de forma independiente pues tiene relaciones que mantienen el equilibrio entre clases, pero sobre todo el Bajo Acoplamiento y la Alta Cohesión ya que estos logran que la asignación de responsabilidades se lleve a cabo de forma coherente y que las clases se creen con un buen diseño donde los objetos interactúan (LARMAN 1999).

### 4.3 Modelo de Diseño

Los modelos de diseño muestran los objetos o clases en un sistema y, donde sea apropiado, los diferentes tipos de relaciones entre estas entidades. Los modelos de diseño son esencialmente el diseño mismo. Son el puente entre los requerimientos y la implantación del sistema (SOMMERVILLE 2005a).

#### Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Los diagramas de clases permiten modelar la vista de diseño del sistema y el mejoramiento del modelo conceptual o de dominio.

A continuación se muestra en el Diagrama de Clases del Diseño del Caso de Uso del Sistema *Gestionar Relación*. Los diagramas de clases del diseño correspondientes a los demás Casos de Uso del Sistema pueden ser consultados en el Anexo 2.

### 4.4 Modelo de implementación

Un modelo de implementación incluye suficiente información para construir el sistema. Debe incluir, no solamente la semántica lógica del sistema y los algoritmos, las estructuras de datos y los mecanismos que aseguran funcionamiento apropiado, sino también las decisiones de organización sobre los artefactos del sistema que son necesarios, permitiendo así el trabajo cooperativo de las personas y el procesamiento por parte de las herramientas.

Esta clase de modelo debe incluir las construcciones para empaquetar el modelo, para la comprensión de la persona, y para la conveniencia de la computadora. Estas no son las características de la aplicación en sí misma. En realidad, son características del proceso de construcción (JACOBSON 2000).

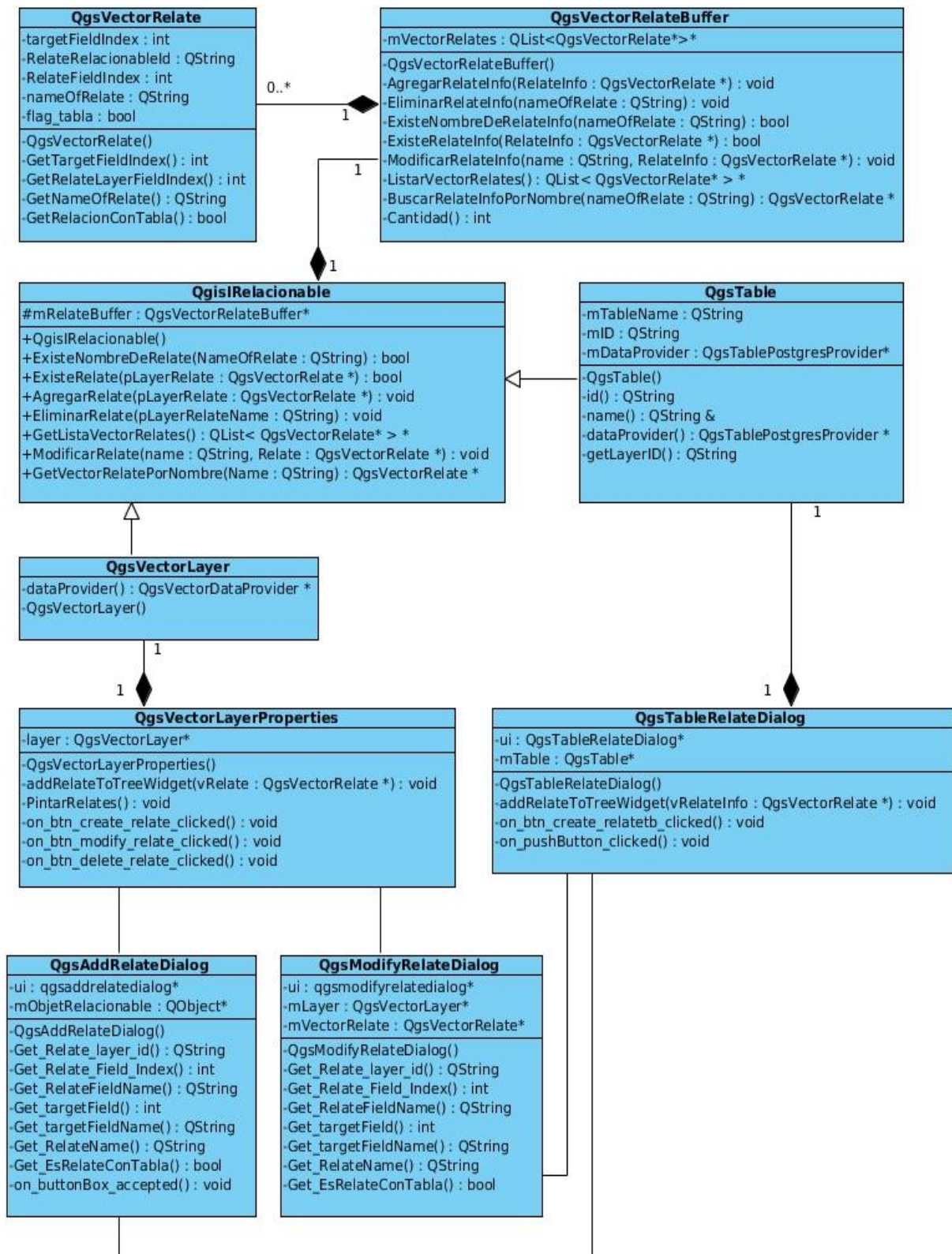


Figura 9: Diagrama de Clases del Diseño del CU Gestionar Relación.

#### 4.5 Diagrama de componentes

Un componente es una unidad física de implementación con interfaces bien definidas pensada para ser utilizada como parte reemplazable de un sistema. Cada componente incorpora la implementación de ciertas clases del diseño del sistema.

Un diagrama de componentes muestra las organizaciones y las dependencias entre tipos de componentes de software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Cada componente ofrece algunas interfaces y utiliza otras. Si las dependencias entre componentes se hacen a través de interfaces, los componentes se pueden sustituir por otros componentes que realicen las mismas interfaces. El diagrama de componentes muestra clasificadores de componentes, las clases definidas en ellos, y las relaciones entre ellas (JACOBSON 2000).

Otra definición refiere que los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Un diagrama de componentes representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables (IVARJACOBSON 2004).

En la figura 10 se presenta el Diagrama de Componentes del Caso de Uso del Sistema Gestionar Relación, el resto de los Diagramas de Componentes pueden ser consultados en el Anexo 3.

#### 4.6 El proceso de pruebas

Según la Real Academia Española *Prueba* se define como la *razón, argumento, instrumento u otro medio con que se pretende mostrar y hacer patente la verdad o falsedad de algo / Someterlo a determinadas situaciones para averiguar o comprobar sus cualidades o comportamientos / Ensayo o experimento que se hace de algo, para saber cómo resultará en su forma definitiva / Para referirse a lo que por su perfecta construcción, firmeza y solidez, es capaz de resistir* (RAE 2001).

Según Pressman una prueba es un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática al software. Las pruebas de software se realizan para descubrir errores cometidos sin darse cuenta al realizar su diseño y construcción, con la utilización de un conjunto de pasos en que se puedan incluir técnicas y métodos específicos del diseño de caso de prueba (PRESSMAN 2005).

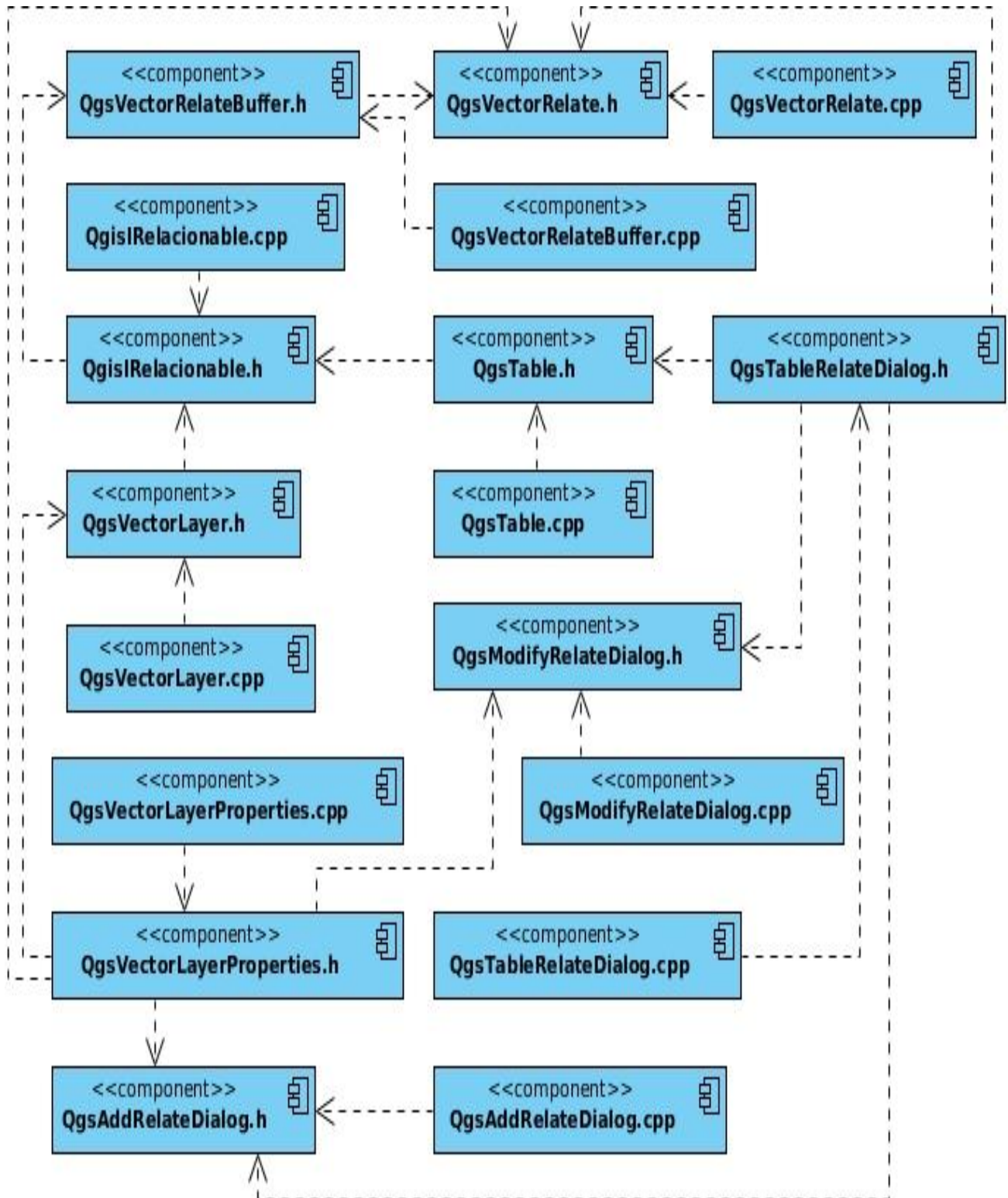


Figura 10: Diagrama de Componentes del CU Gestionar Relación.

### Pruebas de Caja Blanca

Las pruebas de caja blanca se aplican si se conoce el funcionamiento interno del producto, para realizar un examen detallado del sistema. Tienen como propósito verificar que las operaciones internas se realizan de acuerdo con las especificaciones planteadas y que se han probado todos los componentes internos de manera adecuada. Se proporcionan casos de prueba que ejerciten conjuntos específicos de condiciones, bucles o ambos (PRESSMAN 2005).

### Pruebas de Caja Negra

Las pruebas de caja negra se aplican a la interfaz del software, cuando se conoce bien la función específica del producto, para demostrar que cada funcionalidad del mismo es plenamente operacional. Es decir, este tipo de pruebas se ejecutan para examinar aspectos funcionales, tiene poca relación con la estructura interna del software.

Para la verificación y validación del módulo desarrollado se decidió utilizar la *técnica de pruebas de caja negra* pues el mayor interés de los investigadores radica en identificar la mayor cantidad de defectos en las funcionalidades definidas para el sistema, con el objetivo de erradicarlos antes de que el producto sea entregado al cliente, logrando, con esto, aumentar la calidad del producto y satisfacer las expectativas del cliente.

La técnica de prueba a ejecutar está dirigida hacia:

- Funciones que estén incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

#### 4.6.1 Diseños de casos de prueba

El diseño de las pruebas se basa en la creación de casos de prueba cuya ejecución permitirá observar posibles síntomas de defectos. Se puede definir un caso de prueba como “el conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular como, por ejemplo, ejercitar un camino concreto de un programa o verificar el cumplimiento de un determinado requisito” (BERCIAL 2006).



Para la ejecución del proceso de pruebas de la solución, se desarrollaron, por cada Caso de Uso, las secciones a probar y la matriz de datos a utilizar en las pruebas.

**Secciones a probar del Caso de Uso del Sistema Gestionar Relación**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
<p><b>SC 1: Crear Relación.</b></p>	<p>EC 1.1: Crear una relación a una capa con éxito.</p>	<p>Una vez que el usuario selecciona la opción “Propiedades”, y seleccione la opción “Relaciones”, escoge la opción “Crear Relación”, luego el sistema muestra la interfaz “Crear Relación” donde debe llenar los campos de la nueva relación. El usuario introduce los datos correspondientes y accede a la opción “Aceptar”. El sistema realiza las verificaciones pertinentes, crea la relación y vuelve a la interfaz “Relaciones” donde se lista la relación creada.</p>	<p>Clic derecho ir a la opción “Propiedades” de una capa/ Clic en la opción “Relaciones” de una capa/ Clic en la opción “Crear Relación”/ Introducir todos los datos/ Clic en el botón “Aceptar”.</p>
	<p>EC 1.2: Crear una relación a una tabla con éxito.</p>	<p>Una vez que el usuario selecciona la opción “Relaciones”, escoge la opción “Crear Relación”, el sistema muestra la interfaz “Crear Relación” donde el usuario debe llenar los campos de la nueva relación. El usuario introduce los datos correspondientes y accede a la opción “Aceptar”. El sistema realiza las verificaciones pertinentes, crea la relación y vuelve a la interfaz “Relaciones” donde se lista la relación creada.</p>	<p>Clic derecho ir a la opción “Relaciones” de una tabla/ Clic en la opción “Crear Relación”/ Introducir todos los datos/ Clic en el botón “Aceptar”.</p>
	<p>EC 1.3: Crear una relación con el campo nombre de la relación vacío.</p>	<p>El usuario selecciona la opción “Relaciones” y selecciona la opción “Crear Relación”, no introduce el nombre de la relación y selecciona la opción “Aceptar”, el sistema muestra la interfaz “Campo Vacío” con el mensaje de error “Debe introducir un nombre. La relación no será creada.”. El usuario selecciona la opción “Aceptar” y el sistema vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones” /Clic en la opción “Crear Relación”/ Introducir los datos de la relación menos el nombre de la relación/ Clic en el botón “Aceptar”/ Clic en la opción “Aceptar”.</p>

	<p>EC 1.4: Crear una relación con el mismo nombre de una relación existente.</p>	<p>El usuario selecciona la opción “Relaciones” y selecciona la opción “Crear Relación”, introduce los datos de una relación con el campo nombre de la relación igual al de una relación existente y accede a la opción “Aceptar”.</p> <p>El sistema no crea la relación y muestra la interfaz “Conflicto de Nombre” con el mensaje de error: Esta capa ya posee una relación con nombre: <i>nombre</i>. La relación no será creada.” en caso que sea una capa. Si es una tabla muestra el mensaje de error “Esta tabla ya posee una con nombre: <i>nombre</i>. La relación no será creada.”. El usuario selecciona la opción “Aceptar” y el sistema vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones” /Clic en la opción “Crear Relación”/ Introducir los datos de la relación donde el campo nombre de la relación es igual a una existente/Clic en el botón “Aceptar”/ Clic en el botón “Aceptar”.</p>
	<p>EC 1.5: Crear una relación con los mismos datos de una relación existente, y el campo nombre de la relación distinto.</p>	<p>El usuario selecciona la opción “Relaciones” y la opción “Crear Relación”, introduce los mismos datos de una relación existente con el campo nombre modificado, accede a la opción “Aceptar”, el sistema no crea la relación y muestra la interfaz “Conflicto de Relación” con el mensaje de error “Esta capa ya posee una relación con esa estructura. La relación no será creada.” en caso que sea una capa.</p> <p>Si es una tabla muestra el mensaje de error “Esta tabla ya posee una relación con esa estructura. La relación no será creada.”. El usuario selecciona la opción “Aceptar” y el sistema vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones” / Clic en la opción “Crear Relación”/ Introducir los datos de una relación existente, donde el campo nombre de la relación sea distinto/Clic en el botón “Aceptar”/ Clic en el botón “Aceptar”.</p>
	<p>EC 1.6: Cancelar la opción Crear una relación.</p>	<p>Si el usuario selecciona la opción “Relaciones” y la opción “Crear Relación”, introduce o no los datos de una relación y selecciona la opción “Cancelar”, el sistema no crea la relación y vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones” /Clic en la opción “Crear Relación”/ El usuario introduce o no los datos/ Clic en el botón “Cancelar”.</p>

<b>SC 2: Modificar Relación.</b>	<p>EC 2.1: Modificar una relación de una capa con éxito.</p>	<p>Una vez que el usuario selecciona la opción “Propiedades” de una capa y escoge la opción “Relaciones”, selecciona la relación a modificar de la lista de relaciones, luego selecciona la opción “Modificar Relación”.</p> <p>El sistema muestra la interfaz “Modificar Relación” con los datos de la relación a modificar, el usuario modifica los datos deseados y accede a la opción “Aceptar”.</p> <p>El sistema realiza las verificaciones pertinentes, modifica la relación y vuelve a la interfaz “Relaciones”.</p>	<p>Clic derecho ir a la opción “Propiedades” de una capa /Clic en la Opción “Relaciones” de una capa/ Clic en la relación a modificar/ Clic en la opción “Modificar Relación” / Introducir los nuevos datos de los campos a modificar/ Clic en el botón “Aceptar”.</p>
	<p>EC 2.2: Modificar una relación de una tabla con éxito.</p>	<p>Una vez que el usuario selecciona la opción “Relaciones” de una tabla, escoge la relación a modificar de la lista de relaciones, selecciona la opción “Modificar Relación”.</p> <p>El sistema muestra la interfaz “Modificar Relación” con los datos de la relación a modificar, el usuario modifica los datos deseados y accede a la opción “Aceptar”.</p> <p>El sistema realiza las verificaciones, modifica la relación y vuelve a la interfaz “Relaciones”.</p>	<p>Clic derecho ir a la opción “Relaciones” de una tabla / Clic en la relación a modificar/ Clic en la opción “Modificar Relación”/ Introducir los nuevos datos de los campos a modificar/ Clic en el botón “Aceptar”.</p>
	<p>EC 2.3: Modificar una relación sin haber seleccionado una relación.</p>	<p>El usuario selecciona la opción “Relaciones”, no selecciona la relación y selecciona la opción “Modificar Relación”, el sistema muestra la interfaz “Información” con el mensaje “Debe seleccionar una relación”.</p> <p>El usuario selecciona la opción “Aceptar” y el sistema vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones” / Clic en la opción “Modificar Relación”/ Clic en el botón “Aceptar”.</p>



	<p>EC 2.4: Modificar una relación con el campo nombre de la relación vacío.</p>	<p>El usuario selecciona la opción “Relaciones” y luego selecciona la relación a modificar de la lista de relaciones y escoge la opción “Modificar Relación”, modifica los campos deseados y el campo nombre de la relación lo deja vacío y selecciona la opción “Aceptar”, el sistema muestra la interfaz “Campo Vacío” con el mensaje de error “Debe introducir un nombre. La relación no será modificada.”. El usuario selecciona la opción “Aceptar” y el sistema vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones”/ Seleccionar una relación/ Clic en la opción “Modificar Relación”/ Modificar los datos /Clic en el botón “Aceptar”/ Clic en el botón “Aceptar”.</p>
	<p>EC 2.5: Cancelar la opción Modificar Relación.</p>	<p>Si el usuario selecciona la opción la opción “Relaciones” y luego selecciona la relación a modificar de la lista de relaciones y selecciona la opción “Modificar Relación”, y selecciona luego de modificar o no algún campo la opción “Cancelar”, el sistema no modifica la relación y vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones”/ Seleccionar una relación/Clic en la opción “Modificar Relación”/ El introduce o no los nuevos datos/Clic en el botón “Cancelar”.</p>
	<p>EC 2.6: Modificar los datos de una relación existente, con el campo nombre de la relación distinto.</p>	<p>El usuario selecciona la opción “Relaciones” y luego selecciona la relación a modificar de la lista de relaciones y escoge la opción “Modificar Relación”, introduce los datos de una relación existente, aún cuando tenga el nombre de la relación distinto y accede a la opción “Aceptar”, el sistema no modifica la relación y muestra la interfaz “Conflicto de Relación” con el mensaje de error “Esta capa ya posee una relación con esa estructura. La relación no será modificada” en caso que sea una capa. Si es una tabla muestra el mensaje de error “Esta tabla ya posee una relación con esa estructura. La relación no será modificada.” El usuario selecciona la opción “Aceptar” y vuelve a la interfaz “Relaciones”.</p>	<p>Relación”/ Introducir los datos /Clic en el botón “Aceptar”/ Clic en el botón “Aceptar”.</p>

	<p>EC 2.7: Modificar los datos de una relación con el campo nombre de la relación con el mismo nombre de una relación existente.</p>	<p>Si el usuario selecciona la opción “Relaciones” y luego selecciona la relación a modificar de la lista de relaciones y escoge la opción “Modificar Relación”, y modifica los datos de una relación con el campo nombre de la relación igual al de una relación existente y accede a la opción “Aceptar”.</p> <p>El sistema no modifica la relación y muestra la interfaz “Conflicto de Nombre” con el mensaje de error “Esta capa ya posee una relación con nombre “nombre de la relación”. La relación no será modificada”. Si es una tabla muestra el mensaje de error “Esta tabla ya posee una relación con nombre “nombre de la relación”. La relación no será modificada.” El usuario selecciona la opción “Aceptar” y vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones”/ Seleccionar una relación Clic en la opción “Modificar Relación”/ Modificar los datos de la relación donde el campo nombre de la relación igual /Clic en el botón “Aceptar”/ Clic en el botón “Aceptar”.</p>
<p><b>SC 3: Eliminar Relación.</b></p>	<p>EC 3.1: Eliminar una relación de una capa con éxito.</p>	<p>Una vez que el usuario selecciona la opción “Propiedades” de una capa y escoge la opción “Relaciones”, selecciona la relación a eliminar de la lista de relaciones selecciona la opción “Eliminar”, el sistema elimina la relación del listado.</p>	<p>Clic derecho ir a la opción “Propiedades” de una capa/ Clic en “Relaciones”/ Seleccionar una relación/ Clic en la opción “Eliminar Relación”.</p>
	<p>EC 3.2: Eliminar una relación de una tabla con éxito.</p>	<p>Una vez que el usuario selecciona la opción “Relaciones”, selecciona la relación a eliminar de la lista de relaciones selecciona la opción “Eliminar”, el sistema elimina la relación del listado.</p>	<p>Clic derecho ir a la opción “Relaciones”/ Seleccionar una relación/ Clic en la opción “Eliminar Relación”.</p>
	<p>EC 3.3: Eliminar una relación sin éxito.</p>	<p>Si el usuario no selecciona la relación a eliminar y accede a la opción “Eliminar Relación”, el sistema no elimina la relación y muestra la interfaz “Información” con el mensaje “Debe seleccionar una relación”. El usuario selecciona la opción “Aceptar”. El sistema vuelve a la interfaz “Relaciones”.</p>	<p>Clic en la opción “Relaciones”/ Clic en la opción “Eliminar Relación”/ Clic en el botón “Aceptar”.</p>

Tabla 3: Secciones a probar del Caso de Uso del Sistema Gestionar Relación.

**Descripción de las variables**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Capa/Tabla	Lista desplegable	No	En este campo se admiten solamente una de las capas o tablas que se especifican en el módulo.
2	Atributo Capa/Tabla	Lista desplegable	No	En este campo se admiten solamente el atributo de las capas o tablas que se especifican en el módulo.
3	Atributo Seleccionado	Lista desplegable	No	En este campo se admiten solamente el atributo de la capa o tabla que se especifican en el módulo.
4	Nombre de la Relación	Campo de texto	No	En este campo debe introducirse un nombre único para la relación.

**Tabla 4: Descripción de las variables.**

**Matriz de Datos**

ID del escenario	Escenario	Variable 1 Capa/Tabla	Variable 2 Atributo Capa/Tabla	Variable 3 Atributo Seleccionado	Variable 4 Nombre de la Relación	Respuesta del Sistema	Resultado de la Prueba
EC:1.1	Crear una relación con éxito a una capa.	V/	V/	V/	V/	El sistema crea la relación y vuelve a la interfaz "Relaciones" donde se lista la relación creada.	
EC 1.2:	Crear una relación con éxito a una tabla.	V/	V/	V/	V/	El sistema crea la relación y vuelve a la interfaz "Relaciones" donde se lista la relación creada.	

ID del escenario	Escenario	Variable 1 Capa/Tabla	Variable 2 Atributo Capa/Tabla	Variable 3 Atributo Seleccionado	Variable 4 Nombre de la Relación	Respuesta del Sistema	Resultado de la Prueba
EC 1.3:	Crear una relación con el campo nombre de la relación vacío.	V/	V/	V/	I/	El sistema no crea la relación y muestra la interfaz "Campo Vacío" con el mensaje "Debe introducir un nombre. La relación no será creada."	
EC 1.4:	Crear una relación con el mismo nombre de una relación existente.	V/	V/	V/	I/	El sistema no crea la relación y muestra la interfaz "Conflicto de Nombre" con el mensaje "Esta capa/tabla ya posee una relación con nombre "nombre de la relación". La relación no será creada".	

ID del escenario	Escenario	Variable 1 Capa/Tabla	Variable 2 Atributo Capa/Tabla	Variable 3 Atributo Seleccionado	Variable 4 Nombre de la Relación	Respuesta del Sistema	Resultado de la Prueba
EC 1.5:	Crear una relación con los mismos datos de una relación existente, y el campo nombre de la relación distinto.	I/	I/	I/	V/	El sistema no crea la relación y muestra la interfaz "Conflicto de Relación" con el mensaje "Esta capa/tabla ya posee una relación con esa estructura. La relación no será creada".	
EC 1.6:	Cancelar la opción Crear una relación.	N/A	N/A	N/A	N/A	El sistema vuelve a la interfaz "Relaciones".	

Tabla 5: Matriz de Datos SC 1 Crear Relación.

ID del escenario	Escenario	Variable 1 Capa/Tabla	Variable 2 Atributo Capa/Tabla	Variable 3 Atributo Seleccionado	Variable 4 Nombre de la Relación	Respuesta del Sistema	Resultado de la Prueba
EC 2.1:	Modificar una relación de una capa con éxito.	V/	V/	V/	V/	El sistema modifica la relación y vuelve a la interfaz "Relaciones".	
EC 2.2:	Modificar una relación de una tabla con éxito.	V/	V/	V/	V/	El sistema modifica la relación y vuelve a la interfaz "Relaciones".	

EC 2.3:	Modificar una relación sin haber seleccionado o una relación.	N/A	N/A	N/A	N/A	El sistema muestra la interfaz "Información" con el mensaje "Debe seleccionar una relación".
EC 2.4:	Modificar una relación con el campo nombre de la relación vacío.	V/	V/	V/	I/	No se modifica la relación y muestra la interfaz "Campo Vacío" con el mensaje "Debe introducir un nombre. La relación no será modificada.".
EC 2.5:	Cancelar la opción Modificar	N/A	N/A	N/A	N/A	El sistema vuelve a la interfaz Relaciones".
EC 2.6:	Modificar los datos de una relación existente, con el campo nombre de la relación distinto.	I/	I/	I/	V/	El sistema no modifica la relación y muestra la interfaz "Conflicto de Relación" con el mensaje "Esta capa/tabla ya posee una relación con esa estructura. La relación no será modificada".

EC 2.7:	Modificar los datos de una relación con el campo nombre de la relación con el mismo nombre de una relación existente.	V/	V/	V/	I/	El sistema no modifica la relación y muestra la interfaz "Conflicto de Nombre" con el mensaje "Esta capa/tabla ya posee una relación con nombre: <i>nombre</i> . La relación no será modificada".
---------	---	----	----	----	----	---

Tabla 6: Matriz de Datos SC 2 Modificar Relación.

ID del escenario	Escenario	Respuesta del Sistema	Resultado de la Prueba
EC 3.1:	Eliminar una relación de una capa con éxito.	El sistema elimina una relación permanentemente del listado de relaciones en la interfaz "Relaciones".	
EC 3.2:	Eliminar una relación de una tabla con éxito.	El sistema elimina una relación permanentemente del listado de relaciones en la interfaz "Relaciones".	
EC 3.3:	Eliminar una relación con éxito.	El sistema elimina una relación permanentemente del listado de relaciones en la interfaz "Relaciones".	

Tabla 7: Matriz de Datos SC 3 Eliminar Relación.

#### 4.7 Conclusiones parciales

1. La utilización de patrones de diseño en la implementación de la solución propicia mayor organización, comprensión y menor tiempo de desarrollo.
2. El diseño gráfico propuesto se adecua al diseño del software GeoQ, lo que propicia mayores y mejores resultados en la usabilidad y la comprensión de las funcionalidades implementadas.
3. Las pruebas que se seleccionan se consideran acertadas y adecuadas pues buscan el correcto funcionamiento de la aplicación desde la experiencia del usuario.

## *Conclusiones Generales*

Luego de finalizado el proceso investigativo, los autores arriban a las conclusiones siguientes:

1. Las soluciones existentes analizadas no responden totalmente al problema pues no realizan correctamente el proceso de interrelación de múltiples tablas y capas que constituye el aporte esencial de esta investigación o presentan alguna otra limitación para su utilización en GeoQ.
2. El SIG ArcGIS sirvió de referencia para lograr la implementación de la solución, su limitación principal radica en el hecho de constituir una herramienta propietaria de alto costo en el mercado internacional.
3. La utilización de herramientas y tecnologías de plataforma abierta garantiza la soberanía tecnológica impulsada por la universidad y el país.
4. La solución que se propone, igualmente que las herramientas utilizadas para su implementación, es multiplataforma y de código abierto, hecho que amplía considerablemente la cantidad de usuarios que la utilicen.
5. La modelación del dominio y el sistema propicia la ampliación y escalabilidad futura de la herramienta al contar con la documentación técnica según la metodología seleccionada.
6. Los requisitos funcionales, una vez convenidos con el cliente, proporcionan una guía segura para su posterior satisfacción.
7. La utilización de patrones de diseño en la implementación de la solución propicia mayor organización, comprensión y menor tiempo de desarrollo.
8. El diseño gráfico propuesto se adecua al diseño del software GeoQ, lo que propicia mayores y mejores resultados en la usabilidad y la comprensión de las funcionalidades implementadas.
9. Las pruebas que se seleccionan se consideran acertadas y adecuadas pues buscan el correcto funcionamiento de la aplicación desde la experiencia del usuario.

Finalmente:

10. El desarrollo del módulo para la interrelación de múltiples tablas y capas contribuye a disminuir las limitaciones de usabilidad y de análisis de información geográfica y socioeconómica del SIG GeoQ.



## *Recomendaciones*

Los autores de este trabajo de diploma recomiendan:

1. Incluir la posibilidad de importar información socioeconómica desde otros Sistemas Gestores de Bases de Datos como MySQL, SQLite y SQL Server.
2. Implementar el JOIN entre tablas y entre tablas y capas de la misma manera que GeoQ lo realiza para las capas.
3. Ejecutar las pruebas propuestas a partir de los diseños especificados.

## Referencias Bibliográficas

1. AGRÍCOLAS, I. I. D. C. *Generalidades sobre Avalúos de Fincas*. Caracas, 1975. p.
2. AGUASCALIENTES, U. A. D. *Historia de UML*, 2012. [2012]. Disponible en: [www.ingenieriasw.com](http://www.ingenieriasw.com)
3. ALCALÁ, U. D. *Introducción a los SIG*, 2001. [Disponible en: <http://www.geogra.uah.es/qisweb/1modulosespanyol/IntroduccionSIG/GISModule/GISModule.htm>
4. ALEGSA. *¿Qué es una aplicación?*, 2007. [2011]. Disponible en: [www.alegsa.com.ar/Dic/aplicacion.php](http://www.alegsa.com.ar/Dic/aplicacion.php)
5. ALEMAN, Y. R. S. Y. G. N. *Transferencia Tecnológica del Portal Inter-Nos Señales Digitales*. La Habana, Universidad de las Ciencias Informáticas, 2007. 154. p.
6. ANDALUCÍA, C. D. E. D. *Sistemas de Información Geográfica, tipos y aplicaciones empresariales*, 2010. [2012]. Disponible en: [http://sig.cea.es/como\\_funcionan](http://sig.cea.es/como_funcionan)
7. ATELIER, G. L. *Qt Creator, desarrollando aplicaciones rápidamente*, 2009. [Disponible en: <http://www.glatelier.org/2009/05/qt-creator-desarrollando-aplicaciones-rapidamente/>
8. BERCIAL, L. S. Y. P. J. L. *Universidad Europea de Madrid: Taller sobre Pruebas en Ingeniería del Software.*, 2006. 7.
9. CENTER, A. R. *Aspectos básicos de la relación de tablas*, 2011. [2012]. Disponible en: <http://help.arcgis.com/es/arcgisdesktop/10.0/help/index.html#//005s0000002t000000>
10. CHILE, D. D. C. D. L. C. U. D. *Casos de Uso*, 2010. [2012]. Disponible en: <http://www.dcc.uchile.cl/~psalinas/uml/casosuso.html>
11. DATE. *Introducción a los Sistemas de Bases de Datos*. 2001. p.
12. DEFINICIÓN.DE. *Definición de Gestión*, 2011. [2011]. Disponible en: <http://definicion.de/gestion>
13. DEFINICIÓN.ORG. *Definición de información económica*, 2012. [Disponible en: <http://www.definicion.org/informacion-economica>
14. ESPAÑOLA, D. D. L. L. *Definición de cartografía*, 2011a. [2011]. Disponible en: <http://www.wordreference.com/definicion/cartograf%C3%ADa>
15. ---. *Definición de Consulta*, 2011b. [2011]. Disponible en: <http://www.wordreference.com/definicion/consulta>
16. ---. *Definición de Gestión*, 2011c. [2011]. Disponible en: <http://www.wordreference.com/definicion/gesti%C3%B3n>
17. ---. *Definición de Interrelación*, 2011d. [Disponible en: <http://buscon.rae.es/>
18. ---. *Definición de Relación*, 2011e. [2011]. Disponible en: [http://buscon.rae.es/drae/SrvltConsulta?TIPO\\_BUS=3&LEMA=relaci%C3%B3n](http://buscon.rae.es/drae/SrvltConsulta?TIPO_BUS=3&LEMA=relaci%C3%B3n)

19. ESPAÑOLA, R. A. *Definición de Aplicación Informática*, 2011f. [2011]. Disponible en: <http://buscon.rae.es/drae/SrvltGUIBusUsual?LEMA=aplicaci%C3%B3n&origen=RAE>
20. ---. *Definición de Información*, 2011g. [2011]. Disponible en: <http://buscon.rae.es/>
21. ---. *Definición de Mapa*, 2011h. [2011]. Disponible en: <http://buscon.rae.es/>
22. ESPATIAL, S. *What is Spatial Information?*, 2007. [Disponible en: [http://www.sicom.sa.gov.au/about\\_spatial.html](http://www.sicom.sa.gov.au/about_spatial.html)]
23. FERNÁNDEZ, P. L. Y. L. *Generación de casos de prueba a partir de especificaciones UML*, 2001. 11.
24. GARCÍA, D. C. *La edición en TV: diferencias entre los sistemas lineal y no lineal.*, 2009.
25. GEOINFORMÁTICA. *Departamento de Geoinformática*. La Habana, UCI, 2011. 12.
26. GEOQ, P. *Documento Visión*. La Habana, 2010. p. *Expediente de Proyecto*.
27. GÓMEZ, A. V. *SERVICIO DE MEDIA STREAMING PARA LA WEB. PORTAL INTER-NOS. MÓDULOS: TELECLASES Y TV*. Señales Digitales. La Habana, Universidad de las Ciencias Informáticas, 2006. 90. p.
28. GVSIG. *Funcionalidades*, 2011. [Disponible en: <http://www.gvsig.org/web/projects/gvsig-desktop/funcionalidades>]
29. IVARJACOBSON, J. R. Y. G. B. *El Lenguaje Unificado de Modelado*. 2004. p. *Parte\_3\_Referencia\_Enciclopedia\_de\_terminos\_parte\_1.pdf*.
30. JACOBSON, I. *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*. 2000. 435 p.
31. LARMAN, C. *UML y Patrones*
32. *Introducción al análisis y diseño orientado a objetos*. México, 1999. 536 p.
33. LOCKHART, T. *Manual del usuario de PostgreSQL*, 1996.
34. LORES, Y. V. *Módulo de Seguridad para el Sistema de Información Geográfica Quantum GIS*. Geoinformática. La Habana, Universidad de las Ciencias Informáticas, 2011. 85. p.
35. OSGEOLIVE. *Quantum GIS*, 2010. [Disponible en: [http://live.osgeo.org/es/overview/ggis\\_overview.html](http://live.osgeo.org/es/overview/ggis_overview.html)]
36. OVIEDO, U. D. *Entornos de Desarrollo Integrado*, 1999. 5.
37. PBWORKS. *Patrones Arquitectónicos*, 2009. [2012]. Disponible en: <http://isg3.pbworks.com/w/page/7624479/Patrones%20Arquitect%C3%B3nicos>
38. PRESSMAN, R. *Ingeniería del Software. Un enfoque práctico*. 5ta. 2001. p.
39. ---. *Ingeniería del Software: Un enfoque práctico*. Sexta edición. 2005. 927 p.
40. RAE. *Concepto de Prueba*, 2001. [2012]. Disponible en: <http://buscon.rae.es/drae/>
41. REYNOSO, C. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*, 2004. 73.
42. SANTOS, M. A. S. *Definición y Desarrollo Histórico del SIG*, Sin año.

43. SCRIBD. *Metodología del Desarrollo del Software*, 2012a. [Disponible en: [http://es.scribd.com/leo\\_michael\\_1/d/38989975-Metodologias-de-desarrollo](http://es.scribd.com/leo_michael_1/d/38989975-Metodologias-de-desarrollo)
44. ---. Ventajas de C++. en: *Algoritmos y programas*. 2012b. 192.p.
45. ---. *Ventajas y desventajas de PostgreSQL*, 2012c.
46. SIGDESKTOP, L. *Manual de Usuario*. La Habana, 2011. 26 p.
47. SOMMERVILLE, I. Diseño. en: *Ingeniería del Software*. 2005a. 145.p.
48. ---. Requerimientos. en: *Ingeniería de Software*. 2005b. 122.p.
49. TORRES, A. R. *Bases de datos espaciales: Spatial Databases*. *Revista Internacional de la Tierra MAPPING Interactivo*, 2009.

## Bibliografía

1. AGRÍCOLAS, I. I. D. C. *Generalidades sobre Avalúos de Fincas*. Caracas, 1975. p.
2. AGUASCALIENTES, U. A. D. *Historia de UML*, 2012. [2012]. Disponible en: [www.ingenieriasw.com](http://www.ingenieriasw.com)
3. ALCALÁ, U. D. *Introducción a los SIG*, 2001. [Disponible en: <http://www.geogra.uah.es/qisweb/1modulosespanyol/IntroduccionSIG/GISModule/GISModule.htm>
4. ALEGSA. *¿Qué es una aplicación?*, 2007. [2011]. Disponible en: [www.alegsa.com.ar/Dic/aplicacion.php](http://www.alegsa.com.ar/Dic/aplicacion.php)
5. ALEMAN, Y. R. S. Y. G. N. *Transferencia Tecnológica del Portal Inter-Nos Señales Digitales*. La Habana, Universidad de las Ciencias Informáticas, 2007. 154. p.
6. ANDALUCÍA, C. D. E. D. *Sistemas de Información Geográfica, tipos y aplicaciones empresariales*, 2010. [2012]. Disponible en: [http://sig.cea.es/como\\_funcionan](http://sig.cea.es/como_funcionan)
7. ATELIER, G. L. *Qt Creator, desarrollando aplicaciones rápidamente*, 2009. [Disponible en: <http://www.glatelier.org/2009/05/qt-creator-desarrollando-aplicaciones-rapidamente/>
8. BERCIAL, L. S. Y. P. J. L. *Universidad Europea de Madrid: Taller sobre Pruebas en Ingeniería del Software.*, 2006. 7.
9. CENTER, A. R. *Aspectos básicos de la relación de tablas*, 2011. [2012]. Disponible en: <http://help.arcgis.com/es/arcgisdesktop/10.0/help/index.html#/005s0000002t000000>
10. CHILE, D. D. C. D. L. C. U. D. *Casos de Uso*, 2010. [2012]. Disponible en: <http://www.dcc.uchile.cl/~psalinas/uml/casosuso.html>
11. DATE. *Introducción a los Sistemas de Bases de Datos*. 2001. p.
12. DEFINICIÓN.DE. *Definición de Gestión*, 2011. [2011]. Disponible en: <http://definicion.de/gestion>
13. DEFINICIÓN.ORG. *Definición de información económica*, 2012. [Disponible en: <http://www.definicion.org/informacion-economica>
14. ESPAÑOLA, D. D. L. L. *Definición de cartografía*, 2011a. [2011]. Disponible en: <http://www.wordreference.com/definicion/cartograf%C3%ADa>
15. ---. *Definición de Consulta*, 2011b. [2011]. Disponible en: <http://www.wordreference.com/definicion/consulta>
16. ---. *Definición de Gestión*, 2011c. [2011]. Disponible en: <http://www.wordreference.com/definicion/gesti%C3%B3n>
17. ---. *Definición de Interrelación*, 2011d. [Disponible en: <http://buscon.rae.es/>
18. ---. *Definición de Relación*, 2011e. [2011]. Disponible en: [http://buscon.rae.es/drae/SrvltConsulta?TIPO\\_BUS=3&LEMA=relaci%C3%B3n](http://buscon.rae.es/drae/SrvltConsulta?TIPO_BUS=3&LEMA=relaci%C3%B3n)

19. ESPAÑOLA, R. A. *Definición de Aplicación Informática*, 2011f. [2011]. Disponible en: <http://buscon.rae.es/drae/SrvltGUIBusUsual?LEMA=aplicaci%C3%B3n&origen=RAE>
20. ---. *Definición de Información*, 2011g. [2011]. Disponible en: <http://buscon.rae.es/>
21. ---. *Definición de Mapa*, 2011h. [2011]. Disponible en: <http://buscon.rae.es/>
22. ESPATIAL, S. *What is Spatial Information?*, 2007. [Disponible en: [http://www.sicom.sa.gov.au/about\\_spatial.html](http://www.sicom.sa.gov.au/about_spatial.html)]
23. FERNÁNDEZ, P. L. Y. L. *Generación de casos de prueba a partir de especificaciones UML*, 2001. 11.
24. GARCÍA, D. C. *La edición en TV: diferencias entre los sistemas lineal y no lineal.*, 2009.
25. GEOINFORMÁTICA. *Departamento de Geoinformática*. La Habana, UCI, 2011. 12.
26. GEOQ, P. *Documento Visión*. La Habana, 2010. p. *Expediente de Proyecto*.
27. GÓMEZ, A. V. *SERVICIO DE MEDIA STREAMING PARA LA WEB. PORTAL INTER-NOS. MÓDULOS: TELECLASES Y TV*. Señales Digitales. La Habana, Universidad de las Ciencias Informáticas, 2006. 90. p.
28. GVSIG. *Funcionalidades*, 2011. [Disponible en: <http://www.gvsig.org/web/projects/gvsig-desktop/funcionalidades>]
29. IVARJACOBSON, J. R. Y. G. B. *El Lenguaje Unificado de Modelado*. 2004. p. *Parte\_3\_Referencia\_Enciclopedia\_de\_terminos\_parte\_1.pdf*.
30. JACOBSON, I. *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*. 2000. 435 p.
31. LARMAN, C. *UML y Patrones*
32. *Introducción al análisis y diseño orientado a objetos*. México, 1999. 536 p.
33. LOCKHART, T. *Manual del usuario de PostgreSQL*, 1996.
34. LORES, Y. V. *Módulo de Seguridad para el Sistema de Información Geográfica Quantum GIS*. Geoinformática. La Habana, Universidad de las Ciencias Informáticas, 2011. 85. p.
35. OSGEOLIVE. *Quantum GIS*, 2010. [Disponible en: [http://live.osgeo.org/es/overview/qgis\\_overview.html](http://live.osgeo.org/es/overview/qgis_overview.html)]
36. OVIEDO, U. D. *Entornos de Desarrollo Integrado*, 1999. 5.
37. PBWORKS. *Patrones Arquitectónicos*, 2009. [2012]. Disponible en: <http://isg3.pbworks.com/w/page/7624479/Patrones%20Arquitect%C3%B3nicos>
38. PRESSMAN, R. *Ingeniería del Software. Un enfoque práctico*. 5ta. 2001. p.
39. ---. *Ingeniería del Software: Un enfoque práctico*. Sexta edición. 2005. 927 p.
40. RAE. *Concepto de Prueba*, 2001. [2012]. Disponible en: <http://buscon.rae.es/drae/>
41. REYNOSO, C. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*, 2004. 73.
42. SANTOS, M. A. S. *Definición y Desarrollo Histórico del SIG*, Sin año.

43. SCRIBD. *Metodología del Desarrollo del Software*, 2012a. [Disponible en: [http://es.scribd.com/leo\\_michael\\_1/d/38989975-Metodologias-de-desarrollo](http://es.scribd.com/leo_michael_1/d/38989975-Metodologias-de-desarrollo)
44. ---. Ventajas de C++. en: *Algoritmos y programas*. 2012b. 192.p.
45. ---. *Ventajas y desventajas de PostgreSQL*, 2012c.
46. SIGDESKTOP, L. *Manual de Usuario*. La Habana, 2011. 26 p.
47. SOMMERVILLE, I. Diseño. en: *Ingeniería del Software*. 2005a. 145.p.
48. ---. Requerimientos. en: *Ingeniería de Software*. 2005b. 122.p.
49. TORRES, A. R. *Bases de datos espaciales: Spatial Databases*. *Revista Internacional de la Tierra MAPPING Interactivo*, 2009.
50. ALONSO, J. Curso de metodología de la investigación. Santa Clara, 1998. p.
51. ÁLVAREZ, I. Curso de Investigación Científica. Santa Clara Cuba, 1997. p.
52. ANALYSIS, N. C. F. G. I. A. *Los Sistemas de Información Geográfica*, 2010.
53. BIOGRAPHIES, T. Robert William Bemer, 2006. [2012]. Disponible en: [http://www.thocp.net/biographies/bemer\\_bob.htm](http://www.thocp.net/biographies/bemer_bob.htm)
54. PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico*. 5. Mc Graw-Hill/Interamericana de España, S.A, 2002. p.

## **Glosario de Términos**

**Contorno geográfico:** es un concepto utilizado por la ciencia geográfica para definir el espacio organizado por la sociedad. Es el espacio en el que se desenvuelven las personas en su interrelación con el medio ambiente.

**Ordenadores:** es una máquina electrónica que recibe y procesa datos para convertirlos en información útil.

**Software:** Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación. Equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

**Multiplataforma:** término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

**Tecnología:** es conocida como la aplicación práctica del conocimiento generado por la ciencia, así como el conjunto de saberes, habilidades, destrezas y medios necesarios para llegar a un fin predeterminado mediante el uso de objetos artificiales.

**Artefacto:** Es una información que es utilizada o producida mediante un proceso de desarrollo de software.

**Automatización de la información:** es la disciplina responsable de plasmar en la práctica técnico-informática la definición y planificación de los procesos y actividades. Es la base para la creación de los sistemas informáticos y tecnológicos sobre los que se asientan los servicios de los centros de información.

**Administración de la información:** es el proceso de planificar, organizar, dirigir y controlar el uso de los recursos y las actividades de trabajo con el propósito de lograr los objetivos o metas de la organización de manera eficiente y eficaz.

**Tablas:** se refiere al tipo de modelado de datos, donde se guardan los datos recogidos por un programa, para ser organizados y agrupados según las características o principios comunes.



**Capa:** se define a cada una de las partes superpuestas que forman un todo.

**Plugins:** es un módulo de hardware o software que añade una característica o un servicio específico a un sistema.

**Shapefile:** es un formato vectorial de almacenamiento digital donde se guarda la localización de los elementos geográficos y los atributos asociados a ellos.