

**Universidad de las Ciencias Informáticas**  
**FACULTAD 6**



**Título:** Módulo de Edición para el Sistema GeoQ  
Versión 2.0

**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**

**Autor:** Leandro Manuel Sainz Lobet

**Tutor:** Ing. Nilberto Caridad Chavez Marquez

La Habana, 26 de junio 2012

# *Declaración de Autoría*

---

## *Declaración de Autoría*

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Geoinformática de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Leandro Manuel Sainz Lobet (autor)

Nilberto Caridad Chavez Marquez (tutor)

---

---

## *Datos de Contacto*

**Tutor:** Ing. Nilberto Caridad Chavez Marquez

**Formación Académica:** Ingeniero en Ciencias Informáticas.

**Centro Laboral:** Universidad de las Ciencias Informáticas (UCI).

**Correo electrónico:** [nchavez@uci.cu](mailto:nchavez@uci.cu)

*Dedicatoria*

*A mi mamá y mi papá que son el faro que guía mis pasos.  
A mis abuelas que fueron mi luz y a ellas les debo todo.*

## *Agradecimientos*

*A toda mi familia por su apoyo y preocupación y especialmente a mi mamá y mi papá porque han sido mi guía, mi orgullo y mi inspiración y de no ser por ellos no hubiese llegado hasta aquí.*

*A todos mis profes por transmitirme sus valores y conocimientos que me han formado como profesional.*

*A todos mis compañeros de los grupos por los que he pasado por soportarme durante todos estos años.*

*A mi tutor por haberme brindado sus conocimientos y haber confiado en mí.*

*Y en general a todos aquellos que de una forma u otra han contribuido a mi formación.*

## Resumen

En la actualidad el proyecto de desarrollo SIG-Desktop de la Universidad de las Ciencias Informáticas, no cuenta con la capacidad de brindar un cómodo espacio de trabajo en el proceso de edición de cartografías digitales. La UCI como universidad de avanzada, propone al Centro de Geoinformática y Señales Digitales el desarrollo de una nueva versión de dicho proceso. Este sistema será capaz de realizar nuevas funcionalidades de edición de cartografías, brindando un cómodo espacio de trabajo para usuarios y fortaleciendo las ya existentes. La realización de este nuevo sistema se toma como punto de partida para el resto de los procesos de personalización del producto del proyecto SIG-Desktop.

El presente trabajo de diploma aborda la investigación de los procesos de edición cartográfica en Sistemas de Información Geográfica (SIG), con el objetivo de desarrollar un sistema que realice el proceso antes mencionado para el módulo de Edición del Sistema de Información Geográfica GeoQ. El desarrollo de la aplicación se llevó a cabo aplicando la arquitectura orientada a objetos y basada en componentes. En este documento se recogen las principales funcionalidades del proceso mencionado y se muestran las dependencias de los componentes que lo integran a través del modelo de implementación. Se describen las herramientas utilizadas y se muestran las pruebas realizadas al sistema.

**Palabras Claves:** capas, datos espaciales, GeoQ, SIG, TIC.

## Índice de Tablas

|  |    |
|--|----|
| Tabla 1 Requisitos Funcionales del sistema .....         | 34 |
| Tabla 2 DCU_Conmutar edición a múltiples capas .....     | 39 |
| Tabla 3 DCU_ Añadir objetos espaciales a una capa .....  | 41 |
| Tabla 4 DCP CU_Añadir objetos espaciales a una capa..... | 56 |

## Índice de Figuras

|   |    |
|---|----|
| Fig. 1 Antecedentes de los SIG.....   | 3  |
| Fig. 2 Formatos de datos en los SIG.....  | 11 |
| Fig. 3 ArcGIS .....   | 15 |
| Fig. 4 Funcionamiento de la opción Snap .....   | 16 |
| Fig. 5 Conversión de Polilíneas a Polígonos.....                                      | 17 |
| Fig. 6 Opción Copiar/Pegar.....   | 18 |
| Fig. 7 Flujos y fases de trabajo de RUP .....   | 27 |
| Fig. 8 Flujos de trabajo en un proyecto XP.....                                       | 29 |
| Fig. 9 Modelo de Dominio .....  | 33 |
| Fig. 10 Diagrama de Casos de Usos del Sistema .....                                   | 39 |
| Fig. 11 Diagrama de Clases del Análisis CU_Añadir objetos espaciales a una capa ..... | 45 |
| Fig. 12 Diagrama de Colaboración CU_Añadir objetos espaciales a una capa .....        | 46 |
| Fig. 13 Diagrama de clases del diseño CU_Añadir objetos espaciales a una capa.....    | 49 |
| Fig. 14 Diagrama de Secuencia CU_Añadir objetos espaciales a una capa .....           | 50 |
| Fig. 15 Diagrama de Componentes CU_Añadir objeto espaciales a una capa.....           | 53 |
| Fig. 16 Diagrama de Despliegue.....   | 54 |



## Índice de Contenido

|   |    |
|---|----|
| Introducción .....  | 1  |
| Capítulo I: Fundamentación Teórica .....                            | 8  |
| 1.1 Introducción .....  | 8  |
| 1.2 Conceptos relacionados al dominio del problema .....            | 8  |
| 1.3 Proceso de edición cartográfica .....                           | 12 |
| 1.3.1 Referentes teóricos del proceso de edición cartográfica ..... | 12 |
| 1.3.2 Definiciones del proceso de edición cartográfica.....         | 13 |
| 1.4 Análisis de otras soluciones existentes .....                   | 15 |
| 1.5 Conclusiones Parciales.....                                     | 19 |
| Capítulo II: Tendencias y Tecnologías actuales a desarrollar .....  | 20 |
| 2.1 Introducción .....  | 20 |
| 2.2 Lenguaje Unificado de Modelado (UML) .....                      | 20 |
| 2.3 Herramientas CASE.....  | 21 |
| 2.3.1 Rational Rose .....   | 21 |
| 2.3.2 Visual Paradigm.....  | 21 |
| 2.4 Lenguaje de Programación .....                                  | 22 |
| 2.4.1 Lenguaje de Programación a utilizar: C++ .....                | 23 |
| 2.5 Framework de Desarrollo.....                                    | 24 |
| 2.6 Entorno de Desarrollo Integrado .....                           | 25 |
| 2.6.1 Entorno de Desarrollo a utilizar: Qt Creator.....             | 25 |
| 2.7 Metodología de Desarrollo de Software .....                     | 25 |
| 2.7.1 Proceso Unificado de Desarrollo (RUP) .....                   | 26 |
| 2.7.2 Programación Extrema o XP.....                                | 29 |

|  |    |
|--|----|
| 2.7.3 Microsoft Solution Framework (MSF) .....           | 30 |
| 2.8 Conclusiones Parciales.....                          | 30 |
| Capítulo III: Presentación de la solución propuesta..... | 32 |
| 3.1 Introducción .....                                   | 32 |
| 3.2 Modelo del Negocio .....                             | 32 |
| 3.3 Modelo de Dominio .....                              | 33 |
| 3.3.1 Glosario de términos del dominio .....             | 33 |
| 3.4 Requisitos Funcionales .....                         | 34 |
| 3.5 Requisitos no funcionales .....                      | 35 |
| 3.6 Casos de Usos del Sistema .....                      | 38 |
| 3.6.1 Especificación de Casos de Uso .....               | 39 |
| 3.7 Modelo de Análisis.....                              | 44 |
| 3.7.1 Diagramas de interacción.....                      | 45 |
| 3.8 Modelo de Diseño.....                                | 47 |
| 3.8.1 Patrones empleados en el diseño .....              | 47 |
| 3.8.2 Diagramas de clases del diseño .....               | 49 |
| 3.8.3 Diagramas de interacción.....                      | 50 |
| 3.9 Conclusiones parciales .....                         | 50 |
| CAPÍTULO IV: Construcción de la solución propuesta.....  | 52 |
| 4.1 Introducción .....                                   | 52 |
| 4.2 Generalidades de la implementación .....             | 52 |
| 4.3 Modelo de Implementación .....                       | 53 |
| 4.4 Modelo de Despliegue .....                           | 53 |
| 4.5 Validación del sistema propuesto.....                | 54 |
| 4.5.1 Prueba de Caja Negra .....                         | 55 |

|   |    |
|---|----|
| 4.5.2 Diseño de Caso de Prueba Caja Negra. Partición equivalente..... | 56 |
| 4.6 Conclusiones Parciales.....                                       | 58 |
| Conclusiones Generales.....   | 59 |
| Recomendaciones.....  | 60 |
| Referencias Bibliográficas.....                                       | 61 |
| Bibliografía Citada.....  | 63 |

## *Introducción*

El hombre, en la medida que ha evolucionado, se ha encontrado en la necesidad de almacenar toda la información con la cual trabaja o que recopila para utilizarla posteriormente. La información es la base de toda entidad, sin ella no sería posible desarrollar muchas de las actividades que se encuentran concebidas dentro de estas y sobre la cual se toman muchas decisiones. Esta puede ser procesada y transformada en conocimiento, lo que permite que la misma sea de gran utilidad en diferentes esferas de la vida.

En la actualidad el trabajo que realiza el hombre se ve condicionado en muchos aspectos por la utilización de las Tecnologías de la Informática y las Comunicaciones (TICs). Las mismas proporcionan herramientas que facilitan, en gran medida, su trabajo en distintas actividades tales como: procesamiento de datos, almacenamientos digitales de información, automatización de tareas y además una mejor comunicación entre las personas.

El siglo XX fue el escenario propicio para el inicio de la computación. A partir del estallido de la segunda Guerra Mundial en 1945, se realizaron importantes estudios y descubrimientos científicos y tecnológicos que propiciaron el surgimiento y desarrollo de esta nueva tecnología. La computación, desconocida hasta ese momento y que posteriormente se convertiría en un importante recurso para el desarrollo económico y social a nivel mundial. Muchos han sido los retos, desafíos y obstáculos encontrados a lo largo del camino, sin embargo ha sido asombrosa la velocidad con que se han desarrollado las TICs.

Cuba ha reconocido la importancia que tienen las TICs y actualmente se encuentra en un proceso de digitalización de la información y automatización de procesos que dada las necesidades en diferentes sectores de la sociedad se hace necesario automatizar. Dicho proceso está enfocado a priorizar sectores como la salud, los servicios, la educación, las investigaciones y la gestión económica.

La inclusión de la informática en los sectores anteriormente señalados ha provocado que grandes volúmenes de información hayan sido digitalizados y que muchos procesos se ejecuten de forma automática. Grandes cúmulos de datos son almacenados en archivos informáticos, estos cada vez alcanzan mayor proporción y necesitan de grandes dispositivos para ser almacenados, por consiguiente,

la gestión de los mismos se torna cada vez más difícil. Ante la creciente cantidad de datos que manejan hoy las empresas, los profesionales de las tecnologías de la información enfrentan el desafío de administrar datos heterogéneos, guardados en diferentes lugares de las redes corporativas, y también en una diversa variedad de dispositivos. Es por eso que los profesionales de estas tecnologías tienen la necesidad de adquirir las herramientas para administrar y gestionar la información de una forma segura y eficaz.

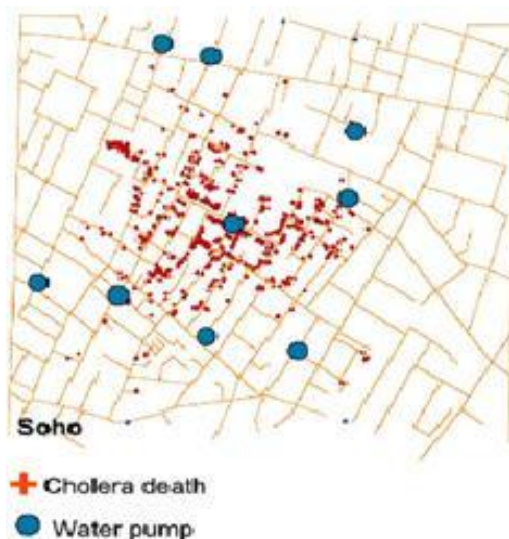
La implementación de sistemas que procesan información y la representan de una forma más descifrable, constituyen uno de los principales aportes de las TICs. Dichos sistemas a su vez contribuyen al ahorro de tiempo, recursos y al progreso socioeconómico del país.

Con la ayuda del desarrollo continuo de las TICs para mejorar los procesos, el estudio de la información geográficamente referenciada toma auge en un marco de globalización. El estudio de estas nuevas temáticas se fortalece como consecuencia de que diversas decisiones a nivel mundial requieren de la existencia, el procesamiento y la posibilidad de compartir dicha información, como datos en laboratorios de investigación, instituciones académicas, la industria privada y las instituciones militares y públicas. Por tal motivo se hizo necesaria la creación de una tecnología capaz de analizar y procesar información geográficamente referenciada. Con este propósito uno de los retos de la sociedad actual es lograr la difusión de dicha información y sus tecnologías asociadas, lo cual conlleva una serie de estrategias que van desde capacitar a la sociedad en general en el uso de esta nueva tecnología hasta evidenciar sus ventajas para el proceso de la toma de decisiones.

Uno de los antecedentes más conocidos del uso de la lógica de esta nueva tecnología se encuentra en el caso del doctor Snow y la epidemia del caso del cólera en Londres en el siglo XIX, el cual a través del mapeo de las muertes por cólera y las fuentes de agua logró identificar el origen de la epidemia de cólera que afectaba a Londres, y con esto logró controlarla y vencerla. (**Fig. 1**) (Gordon, 2008)

Ya en los años 1960 y 1970 emergieron nuevas tendencias en las formas de utilizar los mapas para la valoración de recursos y planificación, el uso de computadoras progresó rápidamente en el manejo de información cartográfica y se afinaron mucho de los sistemas informáticos para distintas aplicaciones cartográficas. (Gordon, 2008)

Ya en la década de los '80 se concreta esta nueva tecnología con la creación de los Sistemas de Información Geográfica (SIG, GIS por sus siglas en inglés) como una solución mejor elaborada y con más funcionalidades para el análisis, manipulación, obtención o visualización de datos geográficos. Ellos permiten la representación, en un mapa digital, de una región con todos sus datos geográficamente referenciados que brindarán información al usuario que los consulte. Los mismos cuentan con herramientas dotadas de grandes capacidades de procesamiento alfanumérico y gráfico, que poseen aplicaciones y procedimientos capaces de realizar la captura, el almacenamiento, el análisis y exposición de los datos representados. (Gordon, 2008)



**Fig. 1 Antecedentes de los SIG**

No solo la producción de hardware ha demostrado ser una vía importante de desarrollo económico. El desarrollo de software ha manifestado ser un factor importante para la economía de cualquier país o institución, Cuba ha alcanzado un avance en este sector en los últimos años. La creación de la Universidad de las Ciencias Informáticas (UCI), primera universidad construida en Cuba al calor de la batalla de ideas, es protagonista en el desarrollo de este sector, aprovechando las posibilidades que ofrecen las TICs para brindar servicios y desarrollar soluciones de software con el fin de informatizar los procesos y proporcionar un avance en la rama de la economía.

En la facultad número 6 de la UCI se encuentra el centro de desarrollo de software de Geoinformática y Señales Digitales (GEySED) compuesto por dos departamentos. En el departamento de Geoinformática se desarrollan varios proyectos de desarrollo de software, entre los cuales se puede apreciar el proyecto de desarrollo de SIG para plataformas de escritorio SIG-Desktop. Este proyecto posee como misión la creación de SIG, basándose en tecnologías de software libre para su uso en entidades nacionales o extranjeras. Dentro de los objetivos del proyecto se encuentra desarrollar un SIG que cumpla con todos los requerimientos de una plataforma para la creación de SIG aplicables a negocios específicos: Sistema de Información Geográfica para plataforma de escritorio (GeoQ).

Hoy, cuando el número de SIG crece en el mercado internacional, la mayoría de ellos no permiten la integración con otros sistemas existentes que puedan prescindir de sus servicios. Tal desventaja se hace notable para GeoQ, una herramienta de arquitectura extensible pero el bajo nivel de usabilidad de las funcionalidades de su módulo de Edición no propicia a usuarios que realicen con la herramienta un cómodo y sustentado trabajo.

A medida que el producto GeoQ vaya tomando madurez y se logre un mayor interés de los clientes por el desarrollo de un SIG, trae como resultado, el aumento de objetos espaciales a editar dentro de las cartografías digitales a realizar. Como consecuencia, los cartógrafos que utilizan la herramienta tendrán que realizar un proceso de edición cartográfica sumamente engorroso por la debilidad en sus funcionalidades de edición, lo que pudiera generar resultados de análisis incorrectos y provocar atrasos en la producción, este último, factor fundamental que influye en el desarrollo de un proyecto.

Teniendo en cuenta lo expuesto anteriormente surge el siguiente **problema de la investigación**: Las funcionalidades de edición en GeoQ se encuentran en un estado inicial de desarrollo que no facilita su utilización para un trabajo sostenido de edición cartográfica.

Para darle solución al problema de la investigación identificado se presenta el siguiente **objetivo general**: Desarrollar la segunda versión del módulo de edición del Sistema de Información Geográfica GeoQ.

Basándose en el problema definido se tiene como **objeto de estudio**: Proceso de edición de cartografías en Sistemas de Información Geográfica.

El objetivo trazado concreta el siguiente **campo de acción**: Módulo de edición del Sistema de Información Geográfica GeoQ.

La **idea a defender** en la presente investigación es: El desarrollo de la segunda versión del módulo de edición del Sistema de Información Geográfica GeoQ, dotará al sistema de nuevas funcionalidades de edición de cartografías, brindando un cómodo espacio de trabajo para usuarios y fortaleciendo las ya existentes.

Con el fin de resolver el problema de la investigación y darle cumplimiento al objetivo planteado con anterioridad de forma sistemática y ascendente se exponen las siguientes **tareas de la investigación**:

1. Realizar un estudio del estado del arte del proceso de edición de cartografías en los Sistemas de Información Geográfica.
2. Caracterizar el proceso de desarrollo de los Sistemas de Información Geográfica.
3. Valorar las herramientas a utilizar en la construcción de la solución.
4. Seleccionar y argumentar la metodología de desarrollo de software a usar en el proceso.
5. Identificar las nuevas funcionalidades que debe brindar el módulo de edición en su segunda versión.
6. Diseñar e implementar la solución propuesta.
7. Realizar las pruebas a la solución propuesta.

Se esperan como **posibles resultados**:

1. Segunda versión del módulo de edición del sistema GeoQ.
2. Documentación técnica del proceso de desarrollo del módulo.
3. Documento de tesis.

Para la realización de la investigación se utilizaron los siguientes métodos de la investigación:

Métodos empíricos:



- Observación: Para registrar y descubrir directamente las características más importantes en el proceso de desarrollo del módulo de edición de herramientas especializadas en la edición cartográfica y así entender las necesidades reales de la herramienta GeoQ.
- Entrevista: Se realizara a los especialistas del proyecto SIG-Desktop y al especialista en cartografía el Lic. Daniel Echevarría González para definir los posibles requisitos a implementar, así como la obtención de propuestas de soluciones.
- Experimento: Para realizar las pruebas finales que tributen a las nuevas funcionalidades propuestas que debe brindar el módulo de edición en su segunda versión.

## Métodos Teóricos:

- Análisis y síntesis: Para conocer, reflexionar y aumentar los conocimientos acerca del proceso de edición de cartografías en los Sistemas de Información Geográfica basándose en la bibliografía consultada, y luego, la síntesis para arribar a conclusiones de la investigación.
- Histórico y lógico: Para determinar las tendencias actuales del desarrollo de los Sistemas de Información Geográfica basado en el estudio y evolución de las mismas.
- Modelación: Para expresar la realidad mediante modelos que facilitan el proceso investigativo. Esto facilita la creación de los modelos ingenieriles.

En la presente investigación se desarrolla el tema “Módulo de Edición para el sistema GeoQ Versión 2.0”. El informe de la investigación está conformado por cuatro capítulos, los cuales brindan en conjunto la solución al problema de la investigación. En el documento se encuentran definidos el proceso de edición de cartografías en los SIG así como su proceso de desarrollo. Una vez definidos y analizados estos procesos se identifican las nuevas funcionalidades que debe brindar el módulo de edición en su segunda versión, así como las funcionalidades que constituyen una estructura idónea y completa a reutilizar. La selección de las funcionalidades está basada en argumentos teóricos y prácticos que son expuestos a medida que avanza la investigación.

El documento se encuentra estructurado de la siguiente manera:

En el **capítulo I** se analizan los conceptos relacionados con la investigación, se realiza una descripción del objeto de estudio identificado, se detalla la situación problemática actual y las principales conclusiones que se obtuvieron luego de que se valoraran las soluciones existentes.

En el **capítulo II** se detalla las principales tecnologías, lenguajes de programación y herramientas que se utilizarán para la construcción de la solución propuesta.

En el **capítulo III** se describe propiamente la solución propuesta en términos de modelo de dominio, requisitos funcionales y no funcionales y los casos de uso del sistema, todo esto unido a los correspondientes diagramas que lo modelan así como una explicación detallada del proceso de construcción de la solución propuesta en función de diagramas de clases y estándares del diseño.

En el **capítulo IV** se plantea la construcción de la solución propuesta en el capítulo III en función de las generalidades de la implementación, modelo de despliegue y modelo de implementación, finalizando el capítulo con los diseños de casos de pruebas que validen la implementación.

## Capítulo 1: Fundamentación Teórica

### 1.1 Introducción

El presente capítulo contiene conceptos y elementos teóricos sobre la investigación donde el conocimiento de los mismos es de vital importancia para el lector, para lograr así una mayor comprensión de la investigación. Se argumenta el estado actual del objeto de estudio, campo de acción, su seguimiento y utilización en la actualidad.

### 1.2 Conceptos relacionados al dominio del problema

Para lograr una correcta asimilación de los temas a tratar en el presente informe de la investigación, se hace necesaria la especificación de diversos conceptos que permiten el cumplimiento de los objetivos y la realización de la investigación.

La naturaleza de los Sistemas de Información (SI) radica en la asociación de datos en formato digital con herramientas informáticas para un posterior estudio de los mismos en dependencia de la necesidad de las entidades a través de objetivos concretos trazados con anterioridad. Un caso particular de los SI son los SIG, donde la información en formato digital no son más que datos geográficamente referenciados de alguna zona en mapas digitales, los cuales actúan como herramientas informáticas.

*“Los Sistemas de Información Geográfica son 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.”*  
(Araya, 2011)

Una definición más corta y puntual sería: Un SIG es un sistema que por medio de computadoras y datos geográficos ayuda a nuestro mejor entendimiento del mundo en que vivimos y nos permite resolver los problemas que diariamente afrontamos. (Ramos, 2000)

La información que se maneja en los SIG tiene como elemento fundamental de los mismos, los datos espaciales o datos cartográficos espaciales, estos representan las informaciones y formas de todo objeto espacial, así como la relación entre ellos. En la mayoría de los casos, estos datos se encuentran representados en coordenadas y topologías, constituyendo así la forma o tipo de dato de representación para la creación de los mapas.

*“Un dato espacial es una variable asociada a una localización en el espacio, por tanto, se refieren a entidades o fenómenos que cumplen los siguientes principios básicos: (Caldas, 2011)*

- *Tienen posición absoluta: Sobre un sistema de coordenadas (x, y, z).*
- *Tienen una posición relativa: Frente a otros elementos del paisaje (topología: incluido, adyacente, cruzado, entre otros).*
- *Tienen una figura geométrica que las representan (punto, línea, polígono).*
- *Tienen atributos que lo describen (características del elemento o fenómeno).”*

Es de vital importancia la información o dato espacial dentro de un SIG, esta puede ser coleccionada entre otras vías a través de información recopilada en el campo o en el área que se piense representar, por sensores remotos (también conocida esta técnica como percepción remota), información publicada acerca de la zona o cartas temáticas y por medio de encuestas, censos y entrevistas. Es de destacar *“que los datos son el principal activo de cualquier sistema de información. Por ello el éxito y la eficacia de un SIG se miden por el tipo, la calidad y vigencia de los datos con los que opera”.* (Caldas, 2011)

En los SIG el proceso encargado de transformar los datos geográficamente referenciados en información útil es el referente a las funciones de análisis espacial. Este proceso va enfocado principalmente a que a través de transformaciones y otras tantas funciones descubrir estructuras u objetos espaciales, asociaciones y relaciones entre los datos, modelado de fenómenos geográficos, entre otros parámetros presentes en el mundo real.

*“El análisis espacial estudia las relaciones de proximidad-distancia de los elementos en el espacio, optimizando su ubicación y ayudando a la correcta toma de decisiones.”* (Geaintec, 2011)

Como aspecto positivo del proceso de análisis espacial se debe mencionar que los resultados arrojados por el efecto en los datos de cada una de las funciones, añaden valor económico y sobre todo

información y conocimiento a los datos informáticos. Dentro de las principales funciones de análisis espacial aplicable en los SIG se encuentran las transformaciones, medidas, interrogaciones, sumarios, optimización, entre otras.

De conjunto con los datos, uno de los principales componentes de la Infraestructura de Datos Espaciales (IDE) son los metadatos geográficos, los cuales son los encargados de permitir la descripción, clasificación y recuperación de la información geográficamente referenciada. El término metadatos proviene del griego “*meta*” que significa cambio, afiliándolo al término “*datos*”, esto conlleva a que son un conjunto de propiedades descriptivas mediante las cuales se rastrean los cambios hacia los datos.

*“Un metadato es una información que describe entre otras cosas, la calidad, distribución, actualidad y referencia espacial de un conjunto de datos. Muchos especialistas llaman a los metadatos como “datos acerca de los datos”.”* (Caldas, 2010)

*“La palabra cartografía tiene su origen en los vocablos “carta” del Latín que significa papel que sirve para comunicarse y “grapho” del griego que significa descripción, estudio o tratado. La cartografía es la rama del grafismo que se ocupa de los métodos e instrumentos utilizados para exponer y expresar ideas, formas y relaciones en un espacio bi o tridimensional. La cartografía parte del principio de que los seres vivos, los fenómenos físicos y sus interrelaciones ocurren en un contexto temporal y espacial y que por lo tanto es posible mapearlos.”* (Rica, 2010)

*“La cartografía constituye un conjunto de operaciones que permiten a partir de observaciones y mediciones, la representación de una parte o la totalidad de la Tierra. La representación de toda o parte de la superficie terrestre en una superficie plana constituye un mapa”.* (Rico, 2000)

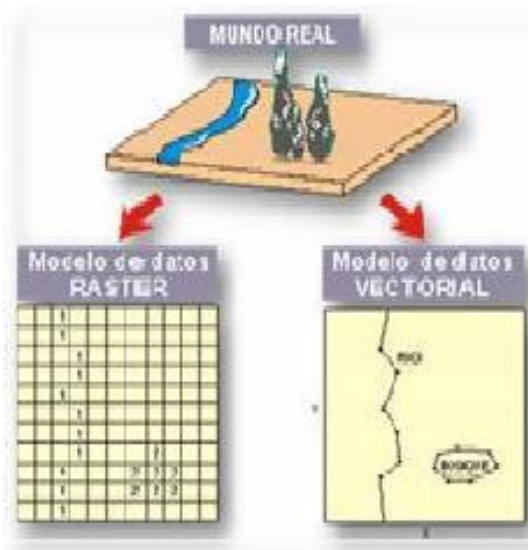
En los SIG los datos son representaciones de objetos de la vida real, estos se clasifican en objetos discretos (casas, arboles, entre otros) y en objetos continuos, los cuales mayormente están dados por los fenómenos atmosféricos (porción de lluvias, elevaciones montañosas, entre otros). La representación de estos objetos puede realizarse a través de formatos vectorial, formato ráster o como en algunos SIG donde se utilizan ambos formatos de representación. **(Fig. 2)**

Los SIG que se centran en el manejo de datos en formato vectorial son más difundidos en el mercado. Los SIG ráster son utilizados en estudios que requieran la generación de capas continuas, necesarias en

fenómenos no discretos; también en estudios medioambientales donde no se requiere una excesiva precisión espacial (contaminación atmosférica, distribución de temperaturas, localización de especies marinas, análisis geológicos, entre otros.). (Rico, 2000)

*“El formato vectorial define objetos geométricos (puntos, líneas y polígonos) mediante la codificación explícita de sus coordenadas. Los puntos se codifican en formato vectorial por un par de coordenadas en el espacio, las líneas como una sucesión de puntos conectados y los polígonos como líneas cerradas (formato orientado a objetos) o como un conjunto de líneas que constituyen las diferentes fronteras del polígono (formato Arco/nodo)”.* (Murcia, 2006)

*“Los datos ráster son una abstracción de la realidad, representan ésta como una rejilla de celdas o píxeles, en la que la posición de cada elemento es implícita según el orden que ocupa en dicha rejilla. En el modelo ráster el espacio no es continuo sino que se divide en unidades discretas. Esto le hace especialmente indicado para ciertas operaciones espaciales como por ejemplo las superposiciones de mapas o el cálculo de superficies.”* (Geomática, 2001)



**Fig. 2 Formatos de datos en los SIG**

Una cartografía se define en su totalidad por la superposición de capas, también se les conoce como cobertura o layer. Las capas se ubican desde la parte inferior hasta la superior donde cada una de estas

capas en correspondencia con la siguiente presenta elementos comunes en dependencia del área u objeto que representan, lo que facilita la transposición de cada una de ellas.

*“Una capa (o layer) es un soporte que contiene información de objetos y formas ya sean líneas, sólidos o cualquier elemento que le asignemos a ella. La ventaja de las capas es que podemos relacionar ciertos elementos para realizar variadas operaciones ya sea seleccionándolos, ocultándolos, bloqueándolos o diferenciándolos, para así hacer más ordenado y eficiente el procedimiento de modelado.”* (Larenas, 2010)

## **1.3 Proceso de edición cartográfica**

Dentro del proceso de desarrollo en los SIG, el proceso de edición cartográfica se evidencia en el módulo de edición donde se realizan todas las funcionalidades que permiten a los usuarios realizar operaciones para la creación y corrección de mapas a través de la gestión de objetos geográficos. De forma general, el proceso antes mencionado como parte del proceso de desarrollo de los SIG está encaminado a permitir editar determinados objetos de una geometría del mapa, la misma se manobra a través de puntos, líneas y figuras asimétricas, definiendo las propiedades particulares de cada objeto en cada uno de los casos.

### **1.3.1 Referentes teóricos del proceso de edición cartográfica**

En la actualidad es muy común observar un mapa y a simple vista poder distinguir rasgos distintivos de algún territorio u objeto. Pero no siempre fue así, en la antigüedad los primeros seres humanos vieron la necesidad de comunicar sus conocimientos acerca de los territorios y con el paso del tiempo perpetuaron sus conocimientos en pinturas en las paredes de las cuevas. Dada la obligación de llevar consigo la información acerca de los territorios surgieron los croquis, elaborados inicialmente en pieles de animales.

Se cree que la primera cartografía que representaba el mundo conocido fue realizada en el siglo VI antes de Cristo por el filósofo griego Anaximandro. Tenía forma circular y mostraba el mundo conocido agrupado en torno al mar Egeo y rodeado por el océano. (Gordon, 2008)

Una de las cartografías más famosas de la época clásica fue trazada por el geógrafo griego Eratóstenes hacia el año 200 antes de Cristo. Representaba el mundo conocido desde Gran Bretaña, al noroeste; la desembocadura del río Ganges, al este; y hasta Libia al sur. Esta cartografía fue la primera en la que aparecieron líneas paralelas y transversales para señalar los puntos con la misma latitud. También aparecían algunos meridianos, pero éstos tenían una separación irregular. (Gordon, 2008)

Con el avance de la ciencia y a medida que las civilizaciones crecían surge la necesidad de representar la forma que iban tomando los territorios. Es desde aquí el punto de partida donde nace y se reconoce la cartografía como ciencia.

En pocas palabras, la cartografía ha facilitado la vida de los seres humanos por las notables comodidades que brinda este proceso. Entre otras cosas, los habitantes reconocen a la perfección sus territorios así como otros de los cuales conocen algún dato de referencia y posibilita con seguridad conocimiento acerca de la obtención de recursos naturales, áreas seguras para la construcción de viviendas, entre otras.

En los SIG el proceso de edición cartográfica es de vital importancia el diseño de los mapas, pues se ha de garantizar que la información que se obtiene de los objetos georeferenciados en la cartografía aparezca de manera sintetizada de la información global del objeto o territorio que se pretende representar. En dicho proceso se realizan operaciones que permiten a los usuarios insertar nuevos objetos geográficos definidos por líneas y polígonos a capas temporales, además de otras funciones como la activación de puntos.

Los antecedentes históricos de la ejecución del primer proceso de edición cartográfica en un SIG datan de la confección del primer SIG creado en Canadá en el año 1965, el cual llevaba como propósito realizar un censo a cerca de la flora y fauna del país. (Gordon, 2008)

## **1.3.2 Definiciones del proceso de edición cartográfica**

La edición cartográfica siempre se ha entendido como un proceso, por las disímiles funcionalidades o técnicas que se encuentran contenidas dentro de la misma. El proceso para SIG varía de un caso a otro en dependencia de las perspectivas y objetivos de la herramienta, pero, a grandes rasgos los procesos de edición cartográfica en SIG presentan pautas similares o funcionalidades básicas.



El proceso de edición cartográfica en SIG se define como un conjunto de técnicas y habilidades de diseño gráfico con el fin de crear un mapa digital dibujando y simplificando la información mediante símbolos, líneas y colores. Un buen desempeño en la realización de un proceso de edición cartográfica dotará a los SIG de una legible representación de la información, para que lectores con bajo nivel de habilidades en estas herramientas logren interpretar con claridad la información. (Araya, 2011)

El lenguaje cartográfico para un SIG define componentes o símbolos similares para la representación de objetos o territorios. La simbología cartográfica está definida por puntos, trazados o líneas y tonalidad, los cuales ejemplifican zonas puntuales, lineales y zonales. Esta simbología está definida a través de valores o variables visuales como la forma, tamaño, color, punteado, entre otras.

Como definición de proceso, la edición cartográfica define un modelo compuesto por un conjunto de métricas o pasos que sirven de guía y posibilitan que el amontonamiento o desorden de la simbología representativa de la información sea mínima y el resultado final sea comprensible. A pesar de constituir una guía, estos pasos no se definen como una especificidad del proceso de edición cartográfica en SIG, dado los propósitos de la herramienta será la ejecución de cada uno de los pasos. Un resumen de este modelo se define a continuación: (savgis.org, 2011)

- La simbología cartográfica automatizada en base a una consulta.
- Elección de una simbología cartográfica a representar.
- Asociación atributo descriptivo - atributo gráfico (símbolo, implantación, variables visuales).
- Posicionamiento automático de las etiquetas descriptivas en objetos representados.
- Filtrado y generalizaciones de propiedades de objetos.

De forma resumida, los procesos de edición cartográfica ejecutados en los SIG en la actualidad sirven de gran ayuda para la construcción de soluciones ya que ofrecen a los usuarios funcionalidades necesarias que facilitan la implementación de proyectos de una forma sencilla. Por otra parte, como aspectos positivos, sintetizan el trabajo a través de la disminución del tiempo significativamente empleado en el proceso de elaboración de la solución y dotan a los SIG de habilidades como la resistencia a cambios y actualizaciones en cualquier instante dentro o fuera del proceso de construcción de la solución.

## 1.4 Análisis de otras soluciones existentes

Uno de los principales problemas que presenta el SIG GeoQ se encuentra centrado en la realización de personalizaciones a otros productos, donde la cantidad de tiempo empleado en el proceso de edición cartográfica de la herramienta es considerable. A continuación se describen un conjunto de herramientas que servirán de guía para la construcción de la solución propuesta y así reducir el tiempo en la realización de ediciones cartográficas.

### ArcGis

ArcGIS es un completo SI que permite crear, analizar, almacenar y difundir datos, modelos, mapas y globos en 3D, poniéndolos a disposición de todos los usuarios según las necesidades de la organización. Como SI, ArcGIS es accesible desde clientes desktop, navegadores web, y terminales móviles que se conectan a servidores departamentales o corporativos. Para los desarrolladores, ArcGIS proporciona herramientas que les permitirán crear sus propias aplicaciones. (Fig. 3) (España, 2010)

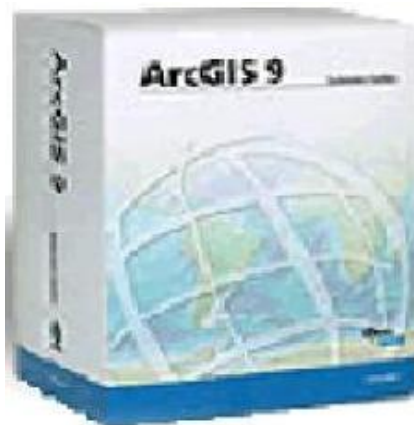


Fig. 3 ArcGIS

### Arcview

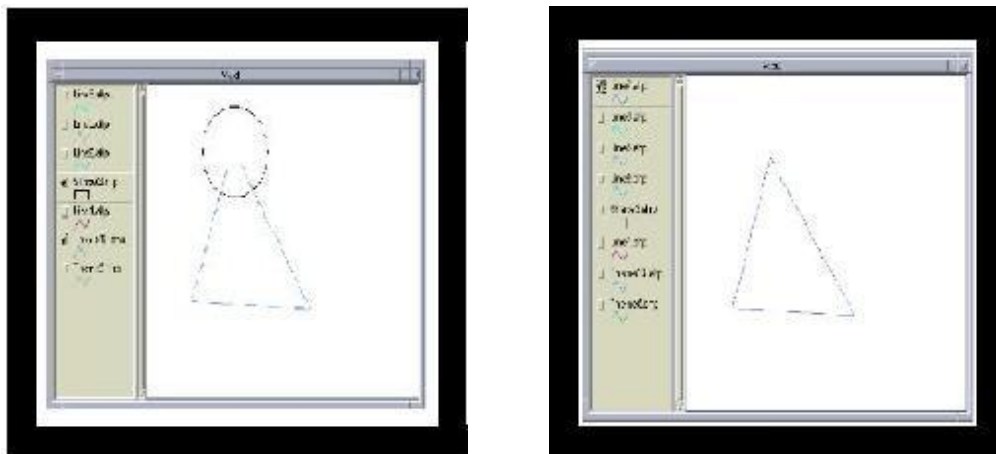
ArcView es una aplicación proporcionada por ArcGIS para el desarrollo de Sistemas de Información Geográfica Desktop, es un software de arquitectura extensible y escalable a diferentes plataformas. Esta nueva arquitectura permite adicionar módulos extendiendo considerablemente las capacidades funcionales del mismo. (Ambiental, 2011)

La herramienta descrita a continuación es una extensión de Arcview, la cual reducirá el tiempo en la realización de correcciones o ediciones de mapas y simplifica la elaboración de proyectos en el contexto de los SIG. La herramienta interactúa con el usuario final el cual no tendrá que contar necesariamente con conocimientos profundos en el uso de SIG ni de lenguajes de programación al trabajar con una interfaz gráfica que le facilitará el proceso de creación de un SIG en el aspecto de edición de datos geográficos. (Puebla, 2011)

A continuación se definen un conjunto de funcionalidades definidas por esta herramienta:

Opción para unir dos o varias líneas (Snap) de manera automática:

Esta nueva funcionalidad es una herramienta de Arcview que permite hacer la unión de dos o más líneas de una manera inmediata y automática. Su funcionamiento se basa en generar un círculo en el cual deben quedar dentro las líneas a unir. La línea o líneas que estén más alejadas del centro del círculo serán las que se unirán a la línea que quede más cercana al centro del círculo. Soporta varias líneas las cuales sus extremos se encuentren dentro del círculo. (Fig. 4)

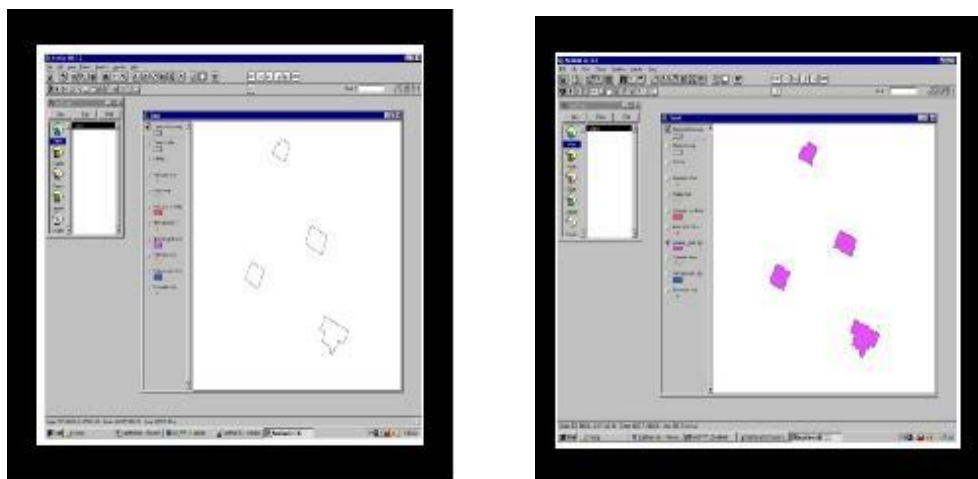


**Fig. 4 Funcionamiento de la opción Snap**

Opción para la creación de nuevos temas haciendo la conversión de gráficas de Polilíneas o Líneas a Polígonos:

Esta funcionalidad de la herramienta, para que funcione correctamente, debe pedir al usuario que las líneas o polilíneas a ser convertidas formen figuras cerradas, de otra manera sería imposible formar

polígonos. Una vez que se comprueba que existen figuras cerradas, se procede a hacer la conversión y crear un nuevo tema. (Fig. 5)



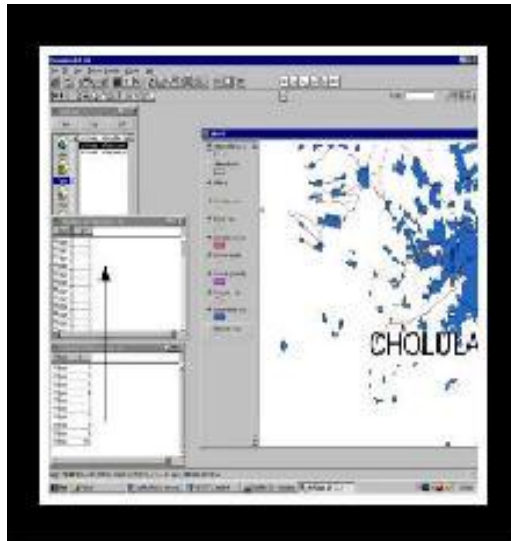
**Fig. 5 Conversión de Polilíneas a Polígonos**

Opción de Copiar/Pegar que también transfiere atributos de las gráficas:

Arcview cuenta con la función de Copiar/Pegar. El problema es que, al hacer un copiado de gráficas, si éstas cuentan con atributos en su tabla, éstos se perderán al hacer la transferencia. La nueva opción de Copiar/Pegar transfiere también los atributos, dando solución a un problema que ocasiona mucha pérdida de tiempo. Esta funcionalidad ayuda mucho en las situaciones donde se requiere transferir grandes cantidades de datos descriptivos de una capa a otra. Se requiere que la capa origen y la capa destino tengan los mismos campos en la tabla. (Fig. 6)

Acceso rápido a la opción de Iniciar/Detener Edición por medio de una funcionalidad:

Con el fin de agrupar las funcionalidades que hacen más cómodo el proceso de edición de cartografía, se agregó la función Iniciar/Detener edición como componente de la extensión en una sola funcionalidad. Esta función activa y desactiva el modo de edición en un tema para que se pueda trabajar en él. La mayoría de las funcionalidades que componen esta herramienta requieren un tema activo antes de poder desempeñar su tarea.



**Fig. 6 Opción Copiar/Pegar**

## **Arc-Gis ArcEditor**

Para la edición y gestión avanzada de información, ArcEditor una herramienta de la familia ArcGIS, con esta modalidad puedes editar, manipular y gestionar datos de naturaleza geográfica. Tiene toda la funcionalidad de Arcview agregando más herramientas que puedan crear, editar y asegurar el proceso de explotación de datos con una calidad a toda prueba. (Geoinfo, 2010)

Estas aplicaciones desarrollan un sistema completo para la edición de cartografías, pero específicamente la extensión de ArcView presentada cumple las expectativas del sistema a implementar, aunque no está diseñado para soporte libre.

## **CoreIDRAW 10**

Una de las herramientas más versátiles para la edición cartográfica es el administrador de objetos, con la gran potencia y facilidad que brinda en el uso de capas. Por cuanto se sabe que para representar en un mapa todos los elementos, como son, ríos, lagos, lagunas, carreteras, vías de trenes, ciudades, escuelas, iglesias, aeropuertos, entre otros, es lógico tenerlos diferenciados por capas, cada cual con sus características especiales, como son los símbolos correspondientes y orden real de acuerdo a su jerarquía, como a su posición real sobre la superficie terrestre. Uno de los administradores de objetos representativo a nivel mundial es el CoreIDRAW 10. (Corel, 2011)

Aplicaciones como CorelDRAW 10 sirven de base para la creación de productos similares. Una desventaja de este sistema, es su dependencia de sistemas privativos, poniéndolo en una situación costosa y de difícil adquisición.

## **1.5 Conclusiones Parciales**

En el presente capítulo se han expuesto diversos conceptos y elementos teóricos que sustentan el problema de la investigación, asociados al proceso de edición de cartografías en SIG, facilitando algunos argumentos que permitan comprender mejor los términos a manejar en la investigación. Se describió el objeto de estudio en general y el dominio del problema así como el análisis de otras soluciones existentes en el mundo, las cuales ofrecen la solución más completa para el problema planteado en esta investigación, pero el hecho de haber sido desarrollado en los Estados Unidos, lo cual lo ubica bajo las leyes del bloqueo, y su carácter privativo, impiden su utilización en el proyecto SIG-Desktop. Lo que no impide que sirvan de guía para la construcción de la solución propuesta. De esta forma se espera dar una visión al lector de los aspectos teóricos manejados en el presente trabajo, necesarios para el posterior desarrollo de la investigación.

# *Tendencias y Tecnologías actuales a desarrollar*

---

## *Capítulo 11: Tendencias y Tecnologías actuales a desarrollar*

### **2.1 Introducción**

La salida al mercado de nuevas TICs con muy poco tiempo entre una y otra, la informatización de la sociedad y la automatización de procesos en empresas de todo tipo, ha obligado a la industria del software a establecer técnicas, metodologías y herramientas para lograr una aceleración en el proceso de desarrollo de soluciones informáticas. Estos instrumentos a medida que aceleran el proceso deben garantizar la calidad y la actualización de las soluciones así como su soporte.

Este capítulo tiene como objetivo seleccionar las tecnologías, herramientas y metodologías correctas que garanticen construir una solución de alta calidad y que se ajuste a las necesidades del proyecto. El estudio que se brinda a continuación está basado en dar a conocer cuál de ellas se usará y el por qué.

### **2.2 Lenguaje Unificado de Modelado (UML)**

Es un lenguaje unificado de modelado orientado a objetos estándar de la industria para especificar, visualizar, construir y documentar los elementos de los sistemas de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto los elementos conceptuales, tales como procesos del negocio y funciones del sistema, como los elementos concretos, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables. (Fundación, 2011)

Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos. Además UML es desde finales de 1997, orientado a objeto de acuerdo a Object Management Group, siendo utilizado diariamente por grandes empresas de desarrollo de software como Microsoft, Oracle, Rational. (Fundación, 2011)

Las ventajas que posibilita la utilización de UML son varias, por un lado el uso de lenguajes visuales facilita su asimilación y entendimiento por parte de cualquier equipo de trabajo, el tiempo invertido en el desarrollo de la arquitectura se minimiza y la trazabilidad y documentación de la solución se confecciona de una forma ordenada y guiada por los casos de uso. Pero si hay una ventaja que destaca sobre todas las demás, es la similitud a la realidad y garantía que se consigue en labores de diseño arquitectónico y

# *Tendencias y Tecnologías actuales a desarrollar*

---

mantenimiento haciendo uso de UML frente a la realización de las mismas tareas en ausencia de modelos.

## **2.3 Herramientas CASE**

Las herramientas CASE (Computer Aided Software Engineering, del español Ingeniería de Software Asistida por Computadoras) modelan la información de negocios cuando ésta se transfiere entre distintas entidades organizativas en el seno de una compañía. El objetivo primordial de las herramientas de esta categoría consiste en representar objetos de datos de negocios, sus relaciones, y ayuda a comprender mejor la forma en que fluyen estos objetos de datos entre distintas zonas de negocio en el seno de la compañía. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas no satisfacen las necesidades de la organización. (Murcia, 2011)

Existen varias herramientas que soportan UML como lenguaje que posibilitan el modelado visual de soluciones informáticas o productos de software, guiando la comunicación entre los integrantes del equipo de desarrollo. Algunos ejemplos son Rational Rose y Visual Paradigm.

### **2.3.1 Rational Rose**

Rational Rose es una de las herramientas más potentes de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de construirlo y cubre todo el ciclo de vida de un proyecto abarcando la concepción y familiarización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. La herramienta cuenta con un soporte UML incomparable, ingeniería inversa o de ida y vuelta multilenguaje, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria, facilidad de uso e integración optimizada. (IBM, 2011)

### **2.3.2 Visual Paradigm**

Visual Paradigm es una herramienta profesional multiplataforma que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue a través de la representación de todo tipo de diagramas. La misma propicia un conjunto de ayudas para el



# *Tendencias y Tecnologías actuales a desarrollar*

---

desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. (EcuRed, 2011)

Características de Visual Paradigm:

- Ofrece un entorno de creación de diagramas para UML versión 8.0.
- Generación de código.
- Generación de bases de datos.
- Transformación de diagramas de entidad.
- Relación en tablas de base de datos.
- Está disponible en múltiples plataformas.
- Diseño centrado en casos de usos y enfocado al negocio lo que genera un software de mayor calidad.
- Capacidad de ingería directa e inversa.
- Soporta aplicaciones Web.

Ambas herramientas estudiadas tienen grandes prestaciones para el modelado del sistema. Pero teniendo en cuenta que Visual Paradigm es una herramienta multiplataforma, que brinda gran ayuda para la representación de las interfaces y con una amplia facilidad de uso para los usuarios. Además de que su licencia proporciona grandes ventajas para el uso en aplicaciones con fines comerciales, posee una amplia bibliografía tanto en materiales audiovisuales como documentación digital y como una de las características fundamentales la posibilidad de integrarse con los servidores de controles de versiones (Subversion), se define su uso para el modelado de la solución, especialmente la versión 8.0 Enterprise Edition, descartándose así al Rational Rose.

## **2.4 Lenguaje de Programación**

Las computadoras en general necesitan de una serie de instrucciones para posibilitar la comunicación entre sus operaciones y los humanos. Para lograr tal interactividad existen los lenguajes de programación que están conformados por una serie de reglas sintácticas y semánticas que serán utilizadas por el programador y a través de las cuales creará un programa o subprograma. También los lenguajes de programación se clasifican según su nivel de abstracción en bajo nivel (lenguaje

# *Tendencias y Tecnologías actuales a desarrollar*

---

perfectamente interpretado por las máquinas, se conoce como lenguaje máquina), lenguaje de medio nivel (es una composición entre lenguaje de máquina y el lenguaje natural) y el lenguaje de alto nivel (que está compuesto por lenguaje humano). (Lanzillotta, 2005)

El desarrollo de una plataforma de escritorio está encaminado a la búsqueda de fáciles alternativas de programación en distribuciones operativas libres, lo que es de destacar que el lenguaje de programación a utilizar en la solución, por los requerimientos del proyecto, posibilita la escritura de instrucciones de una forma fácil y entendible.

## **2.4.1 Lenguaje de Programación a utilizar: C++**

El lenguaje C++ se comenzó a desarrollar en 1980. Su autor fue Bjerne Stroustrup, de la AT&T (Americana de Telefonía y Telegrafía). Al comienzo era una extensión del lenguaje C que fue denominada C con clases. Este nuevo lenguaje comenzó a ser utilizado fuera de la AT&T en 1983. El nombre C++ es también de ese año, y hace referencia al carácter del operador incremento de C (++). Ante la gran difusión y éxito que iba obteniendo en el mundo de los programadores, la AT&T comenzó a estandarizarlo internamente en 1987 y ya para 1989 se estandarizó a nivel americano e internacional. (Virtual, 2011)

C++ es una versión ampliada del lenguaje C. Incluye todo lo que forma parte de C y añade soporte para la programación orientada a objetos (POO). Además, contiene muchas mejoras y características que sencillamente lo convierten en C mejor, independientemente de la programación orientada a objetos. Una particularidad del C++ es la posibilidad de redefinir los operadores (sobrecarga de operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales. C++ permite trabajar tanto a alto como a bajo nivel. Es importante señalar que aunque todo lo que se conozca sobre el lenguaje C se puede aplicar a C++, comprender sus características más avanzadas requerirá una importante inversión de tiempo y esfuerzo. De cualquier manera las recompensas de programar en C++ justificarán de sobra el esfuerzo realizado. (Systems, 2011)

El lenguaje ha experimentado un notable y extraordinario éxito desde su creación. De hecho, muchos sistemas operativos, compiladores e intérpretes han sido escritos en C++, haciendo mención entre tantos de ellos, al propio Windows y Java. Una de las razones de su éxito es ser un lenguaje de propósito general que se adapta a múltiples situaciones.

# *Tendencias y Tecnologías actuales a desarrollar*

---

Tanto sus fervientes seguidores como aquellos que critican el éxito del lenguaje, han reconocido el gigantesco salto de C++ dentro del ámbito de la programación, ambos bandos reconocen la alta complejidad del mismo. Vale destacar que C++ ha servido de elemento base y justificación para el diseño de otros lenguajes que hoy día intentan eliminar sus fallas al tiempo que mantienen sus ventajas, ejemplo de este caso son los lenguajes C# y Java. Por otra parte como aporte de este lenguaje y dando origen a un nuevo paradigma de la programación, aplica la implementación de las plantillas como una de sus últimas incorporaciones (meta programación).

Para la selección del lenguaje a utilizar en el desarrollo del sistema, se tiene en cuenta en primer lugar la existencia de un Entorno de Desarrollo Integrado de programación estandarizado y calificado en sistemas operativos libres. C++ es un lenguaje que brinda la posibilidad de interactuar perfectamente con periféricos por lo que se identificó como lenguaje principal para la creación del sistemas propuesto, tomando en cuenta además, que es el lenguaje nativo del framework de desarrollo Qt, el cual brinda un conjunto de funcionalidades que facilitan y agilizan el desarrollo de la aplicación.

## **2.5 Framework de Desarrollo**

Qt es una biblioteca multiplataforma de software que desarrolla Nokia para crear interfaces gráficas de usuario y el desarrollo de programas sin interfaz gráfica como herramientas de la consola y servidores. El SIG Google Earth, la aplicación para llamadas telefónicas por internet Skype, el gestor de imágenes Adobe Photoshop Album y el software de virtualización Virtual Box hacen uso de esta biblioteca, la cual posee numerosos módulos que permiten trabajar en bases de datos, gráficas, imágenes y sonido. (EcuRed, 2011)

Qt ofrece un conjunto de funciones que facilitan el desarrollo de la solución que se propone. Una de ellas es el mecanismo de señales y slots (ranuras), las señales son emitidas cuando ocurre un evento y el slots es una función que se ejecuta en respuesta a una señal. El desarrollo y acoplamiento de módulos de software o hardware para añadir características o servicios específicos (pluggins) es otra de las funciones que aporta esta estructura conceptual y tecnológica de soporte (framework), permitiendo al sistema adaptarse a cambios futuros con la inclusión de nuevas funcionalidades y el desarrollo de nuevas versiones sin la necesidad de hacer modificaciones al código fuente del sistema.

## **2.6 Entorno de Desarrollo Integrado**

Un entorno de desarrollo integrado o IDE (del inglés Integrated Development Environment) es un programa compuesto por un conjunto de herramientas que proveen un marco de trabajo amigable para los lenguajes de programación. Dentro de las partes fundamentales que lo integran se encuentra el editor de código, compilador, depurador y un constructor de interfaz gráfica de usuario. (EcuRed, 2011)

### **2.6.1 Entorno de Desarrollo a utilizar: Qt Creator**

Qt Creator es un entorno de desarrollo multiplataforma muy completo integrado a la biblioteca Qt y ofrecido por Nokia. Compatible con los sistemas operativos Windows, Linux y Mac OS X y es distribuido bajo tres tipos de licencia: las licencias públicas Qt GNU LGPL v. 2.1, Qt GNU GPL v. 3.0 y la licencia comercial de desarrollo Qt Commercial Developer Licence. (Qt, 2010)

Contiene un sofisticado editor de código que proporciona completamiento de código y ayuda de contexto para los lenguajes C++ y Java Script. Tiene integrado los sistemas de control de versiones Subversion, CVS (Control Versión System), Perforce y Mercurial. Para el desarrollo gráfico cuenta con los editores Qt Designer para el diseño de interfaces gráficas y Qt Quick Designer para el desarrollo de interfaces animadas con el lenguaje Java Script. Entre otras cosas esta herramienta soporta la construcción de aplicaciones de escritorio y dispositivos móviles.

Teniendo en cuenta que Qt Creator garantiza el uso de las funcionalidades del framework Qt, es importante resaltar que el entorno de desarrollo escogido cumple con todas las restricciones de desarrollo que la UCI y el país está respaldando, seguir el principio de utilizar herramientas desarrolladas en entornos libres para garantizar una independencia tecnológica. Por otra parte, el IDE es multiplataforma, se distribuye bajo licencias públicas y se acopla perfectamente a la medida de las necesidades de la solución propuesta. Qt Creator es el IDE seleccionado para el desarrollo de la aplicación propuesta dadas las características descritas anteriormente.

## **2.7 Metodología de Desarrollo de Software**

Las Metodologías de Desarrollo de Software definen quién, cuándo y cómo debe hacerse qué para obtener productos en un determinado proceso de desarrollo. Guían a los desarrolladores en el diseño e implementación del producto y se dividen en dos grandes grupos metodológicos:

# *Tendencias y Tecnologías actuales a desarrollar*

---

- **Ágiles o Ligeras:** Orientadas a la interacción con el cliente y el desarrollo incremental. Muestra versiones al consumidor en cortos períodos de tiempo para la evaluación y sugerencia de cambios en el producto a medida que avanza el desarrollo.
- **Tradicional o pesadas:** Orientadas al control estricto de los procesos. Establecen rigurosamente las actividades a desarrollar, herramientas a manejar y notaciones que se usarán.

Los dos grandes grupos de clasificación en términos metodológicos exteriorizan grandes diferencias. Es por ello que responden a desemejantes procesos de desarrollo, en dependencia de la complejidad del producto y los requisitos que establezca el cliente.

## **2.7.1 Proceso Unificado de Desarrollo (RUP)**

La metodología RUP, por sus siglas en inglés es una metodología de desarrollo de software basada en componentes e interfaces bien definidas con lo cual se podrá contar con guías para poder documentar e implementar de una manera fácil y eficiente. Junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Es un proceso que puede especializarse para una gran variedad de sistemas de software, en disímiles áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. RUP no es un sistema con sendas firmemente establecidas, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

RUP como proceso define en su modelación un conjunto de elementos como guía al trabajo en el proceso de desarrollo:

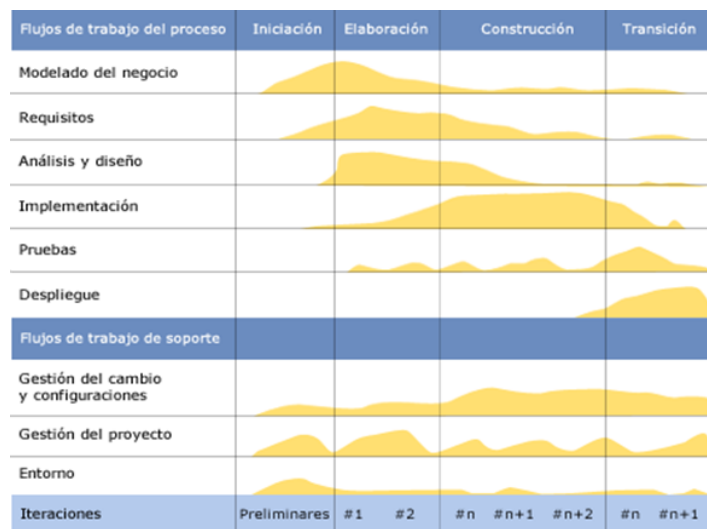
- **Trabajadores** (quién): Definen el comportamiento y responsabilidades de un individuo, los cuales se encargan de realizar las actividades y son los responsables de un conjunto de artefactos.
- **Actividades** (cómo): Tarea con un propósito claro, ejecutada por un trabajador y manipula elementos.
- **Artefactos** (qué): Producto tangible producido, transformado y usado por las actividades.
- **Flujo de actividades** (cuándo): Secuencia de actividades ejecutada por trabajadores con un resultado de valor notorio.

# Tendencias y Tecnologías actuales a desarrollar

RUP es un proceso para el desarrollo de un proyecto de software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Significativo en esta metodología son sus tres características esenciales:

- **Dirigido por casos de usos:** Los casos de usos reflejan y orientan al proyecto a la importancia para los futuros usuarios de lo que necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo.
- **Iterativo e incremental:** Dividiéndose el proyecto en sub-proyectos donde una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo, los 6 primeros son conocidos como flujos de ingeniería y los 3 últimos de apoyo los cuales tienen lugar sobre 4 fases.



**Fig. 7 Flujos y fases de trabajo de RUP**

Flujos de trabajo de ingeniería:

# *Tendencias y Tecnologías actuales a desarrollar*

---

- **Modelado empresarial o del negocio:** Describe los procesos de negocio, identificando quién participa y las actividades que requieren automatización.
- **Requisitos o Requerimientos:** Define qué es lo que el sistema debe de hacer, se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de las funcionalidades pronosticadas y las restricciones impuestas, se indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes así como la estructura de capas de la aplicación.
- **Prueba o Testeo:** Busca identificar los defectos y corregirlos antes de la instalación final del sistema, se verifica la integración apropiada de los componentes y que se satisfacen los requerimientos de los clientes.
- **Despliegue o Instalación:** Realiza actividades como empaque, instalación, asistencia a usuarios, y otras tantas para entregar el software a los usuarios finales.

Flujos de trabajo de apoyo:

- **Gestión de cambios y configuración:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto.
- **Gestión de proyectos:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades del cliente.
- **Entorno o Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Cada una de las siguientes fases representa un ciclo de desarrollo en la vida de un producto de software:

- **Inicio o Concepción:** Se confecciona un plan de fases, se identifican los principales casos de usos y riesgos.
- **Elaboración:** Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

# Tendencias y Tecnologías actuales a desarrollar

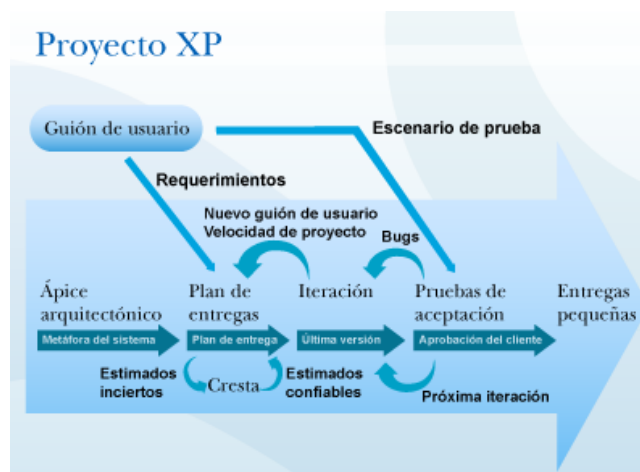
- **Transición:** Se centra la implementación del producto en el cliente y se entrena a los usuarios. Mayormente en esta fase surgen nuevos requisitos a ser analizados. (IBM, 2011)

## 2.7.2 Programación Extrema o XP

La programación extrema (XP) es el más destacado de los procesos ágiles de desarrollo de software. Basado en la simplicidad de las soluciones que se implementan, la comunicación entre los integrantes del proyecto y la retroalimentación del cliente y el equipo de desarrollo. Diseñada para proyectos de corto plazo y su objetivo principal es la programación rápida. Se centra en la reutilización de código, para lo cual se crean patrones o modelos estándares lo que la hace más flexible al cambio. (EcuRed, 2011)

Esta metodología está diseñada para facilitar la entrega del software cuando el cliente lo requiere. XP motiva a los desarrolladores a responder a los cambiantes requerimientos del cliente, incluso cuando el ciclo de vida del desarrollo se encuentre en su etapa final.

XP posee 5 principios básicos: realimentación rápida, asumir la simplicidad, cambio incremental, adherirse al cambio y trabajo de alta calidad. La puesta en práctica de estos principios así como la sucesiva secuencia de cada uno de sus flujos de trabajo se cumple metódica y extremadamente, de ahí el nombre de Programación Extrema.



**Fig. 8** Flujos de trabajo en un proyecto XP



## **2.7.3 Microsoft Solution Framework (MSF)**

Microsoft Solution Framework (MSF) es una metodología flexible e interrelacionada que brinda un enfoque disciplinado para proyectos basados en tecnología, define un grupo de principios, modelos, disciplinas, conceptos, guías y prácticas que pretenden facilitar la consecución de los objetivos durante el ciclo de vida del desarrollo del software.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación. Las disciplinas que rigen a MSF son Gestión de proyectos, Control de riesgos y Control de cambios. (Milagro, 2010)

La selección de la Metodología de desarrollo de Software estará encaminada por las características del proceso de desarrollo en el proyecto de desarrollo SIG-Desktop. Por ello queda descartada la posibilidad de la metodología MSF, la metodología idónea estará entre RUP y XP.

XP no es factible para el proyecto SIG-Desktop puesto que como metodología ágil solo se compromete con pequeñas soluciones y requiere una constante interacción con el cliente, lo que no es posible si este último no se encuentra en la misma nación, además, no define una arquitectura temprana y en cambio la va reconstruyendo en consecuencia de las versiones y requisitos que va presentando el usuario final, lo que para proyectos de gran escala sería un gasto de tiempo y recurso.

Por lo expuesto anteriormente queda como metodología a implantar RUP. Es un proceso configurable, se ajusta a pequeñas y grandes organizaciones de desarrollo. Se basa en una arquitectura de procesos simple y transparente que permite una uniformidad, sin embargo, se pueden variar para adaptarse a diferentes situaciones. Contiene un kit de desarrollo, proporcionando soporte para configurar el proceso y para adaptarse a las necesidades de una organización determinada.

## **2.8 Conclusiones Parciales**

En este capítulo se han presentado los instrumentos a utilizar en el proceso de desarrollo de la solución que se propone y sus características los hacen los más adecuados para el proceso de desarrollo. Se

# *Tendencias y Tecnologías actuales a desarrollar*

---

llegó a la conclusión de utilizar RUP como metodología de desarrollo por ser adaptable, está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental. Como lenguaje de modelado se aplica UML por ser de fácil asimilación y entendimiento, el tiempo invertido en el desarrollo de la arquitectura se minimiza y agiliza la detección y resolución de errores. UML se utiliza con la herramienta Visual Paradigm ya que soporta el ciclo de vida completo del desarrollo de software y es multiplataforma. El lenguaje de programación es C++ gracias a que está muy ligado al hardware con elementos que le permiten un estilo de programación con alto nivel de abstracción. Se utiliza como entorno de desarrollo integrado Qt Creator con las librerías correspondientes al framework Qt de Nokia por ser multiplataforma y ofrecer un conjunto de funcionalidades como el mecanismo de señales y la implementación de pluggins.

# *Presentación de la solución propuesta*

---

## *Capítulo 999: Presentación de la solución propuesta*

### **3.1 Introducción**

En este capítulo se describen las características de la solución propuesta, para una mayor comprensión del contexto en que se desarrolla el sistema a través de un estudio del negocio y el entorno donde se desarrolla, se identifican los principales conceptos del dominio del problema y las relaciones entre estos, se identifican los requisitos funcionales y no funcionales y se modela el diagrama de Casos de Usos del Sistema describiéndose textualmente.

Realizar el diseño de una aplicación es muy importante, con ello se puede desarrollar la programación del sistema de forma más eficiente, pues permite un acercamiento al lenguaje de programación en el cual será desarrollada. En el presente capítulo se modela el diseño del sistema propuesto, con todos sus artefactos y diagramas basándose en distintos patrones de diseño y arquitectura.

El conjunto de actividades mencionadas garantiza el comienzo de los flujos de análisis y diseño del sistema, para lograr una evaluación interna del mismo que facilite el proceso de implementación que le continúa.

### **3.2 Modelo del Negocio**

El modelado del negocio logra crear una visión profunda de la organización a la que se le realiza la automatización, permitiendo definir los procesos, roles y responsabilidades en los modelos de casos de uso del negocio. Realizar un modelado del negocio tiene como ventaja que se puedan derivar los requerimientos del sistema que va a soportar la organización.

La presente investigación no está encaminada hacia las necesidades de un cliente específico ni tiene por objetivo la creación de una aplicación independiente en la que deban identificarse procesos a automatizar, la misma está guiada a la personalización de la aplicación GeoQ a través de la incorporación de nuevas funcionalidades en su módulo de Edición, presentándose como segunda versión. Como no están definidos los procesos de negocio, se procederá a trabajar en el desarrollo del Modelo del Dominio.

## 3.3 Modelo de Dominio

El modelo del dominio ayuda a comprender los conceptos más importantes asociados al dominio del problema. Es una representación visual estática del entorno real objeto del proyecto. Se llama dominio pues se centra en una parte del negocio, la relacionada con el ámbito del proyecto. Se utiliza el término Modelo de Dominio para distinguirlo del Modelo de Negocio ya que este último es mucho más abarcador.

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. El modelo de dominio se debe concebir como un diccionario visual de abstracciones que será utilizado en fases posteriores y cuya función principal es ayudar a comprender el problema a tratar. No se trata de un conjunto de diagramas que describen clases de software ni objetos de software con responsabilidades, sino más bien representa las clases conceptuales u objetos del mundo real en un dominio de interés.

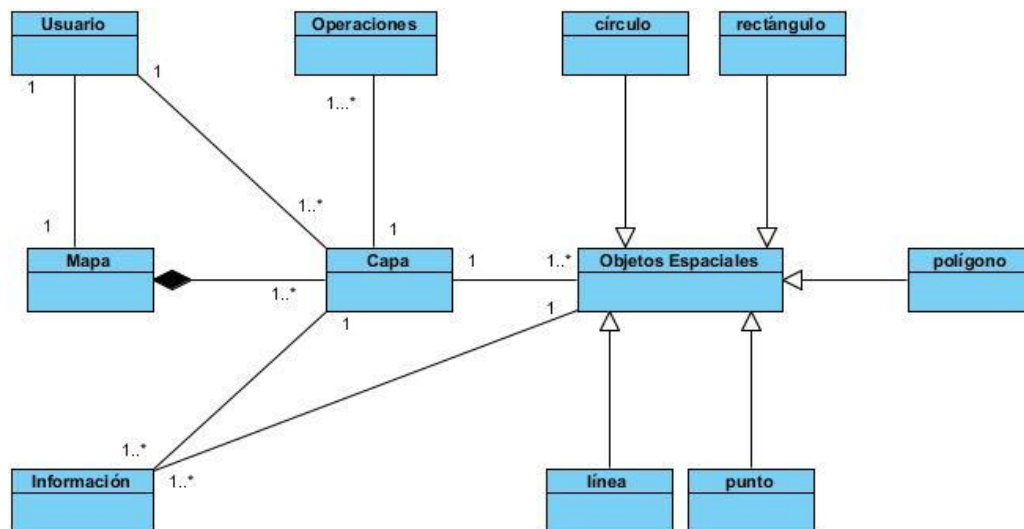


Fig. 9 Modelo de Dominio

### 3.3.1 Glosario de términos del dominio

- **Usuario:** Persona que trabaja en una empresa, organismo u organización que necesite utilizar GeoQ como principal fuente de información geográfica.
- **Mapa:** Representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los

# Presentación de la solución propuesta

globos terráqueos. Por sus propiedades métricas es posible tomar medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto.

- **Capa:** Elemento que superpuesto uno con el otro conforman un mapa, definen por separado la información a representar del mismo.
- **Objeto espacial:** Forma de representación de la realidad, entendida no sólo como algo físico sino también como un pensamiento.
- **Información:** Descripción elemental o una información.
- **Operaciones:** Acciones sobre las capas que permiten la representación de una parte o la totalidad de la Tierra.

## 3.4 Requisitos Funcionales

De acuerdo con RUP, el término requisito puede definirse como una condición que el sistema debe cumplir o capacidad que debe tener. Para que un requisito se considere satisfactorio debe cumplir las siguientes características: implementación independiente, consistente y no ambiguo, preciso, verificable, que pueda ser leído y modificable.

*Tabla 1 Requisitos Funcionales del sistema*

| Alias | Requerimiento                                    | Descripción  |
|-------|--|--|
| RF 1  | Desmarcar opciones de la barra del menú edición. | El sistema debe permitir al usuario desmarcar una opción de la barra del menú de edición seleccionada para realizar otras operaciones.                                   |
| RF 2  | Conmutar la edición a múltiples capas.           | El sistema debe permitir al usuario marcar la opción "Conmutar edición a múltiples capas" que permitirá realizar todo el proceso de edición en todas las capas del mapa. |
| RF 3  | Guardar la edición de múltiples capas.           | El sistema debe permitir al usuario marcar la opción "Guardar edición de múltiples capas" que permitirá guardar todo el proceso de edición en todas las capas            |

## *Presentación de la solución propuesta*

---

|      |                                       |  |
|------|---------------------------------------|--|
|      |                                       | del mapa.  |
| RF 4 | Añadir objetos espaciales a una capa. | El sistema debe permitir al usuario añadir objetos espaciales de tipo polígono y círculo sobre la capa del mapa que se esté trabajando.                        |
| RF 5 | Definir nodos de objetos espaciales.  | El sistema debe permitir definir los nodos de los objetos espaciales añadidos en las capas, para modificaciones que se quieran realizar en el objeto espacial. |
| RF 6 | Guardar objetos espaciales.           | El sistema debe permitir al usuario guardar un objeto espacial como fichero en la dirección especificada por el usuario.                                       |
| RF 7 | Cargar objetos espaciales.            | El sistema debe permitir al usuario cargar objetos espaciales como fichero para agregarlos a la capa seleccionada por el usuario.                              |

### **3.5 Requisitos no funcionales**

En este epígrafe se hará referencia a los requisitos no funcionales, que no son más que las propiedades o cualidades que el sistema debe tener. Para ello se tuvo en cuenta la importancia que poseen en función de que clientes y usuarios puedan valorar cuán usable, seguro, conveniente y agradable será el sistema.

#### **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 dependerán de la cantidad de información que tenga que

# *Presentación de la solución propuesta*

---

procesar la aplicación.

## **Usabilidad**

- El sistema debe ser concebido para ser utilizado por personas que tengan conocimientos básicos de informática y en el trabajo de gestión en SIG.
- Para que los usuarios puedan utilizar óptimamente el sistema, necesitarán un tiempo de adiestramiento y estudio de los conceptos principales asociados al dominio de la aplicación.
- La información deberá estar disponible en todo momento, limitada solamente por las restricciones establecidas como parte de las políticas de seguridad establecidas en el sistema.
- La interfaz debe ser amigable y entendible, fácil de acceder y manipular.
- Las funcionalidades principales del sistema estarán orientadas a iconos para un mayor reconocimiento por parte del usuario.

## **Restricciones de diseño e implementación**

- Diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.
- El producto de software final debe diseñarse sobre una arquitectura orientada a objetos y basada en componentes.
- Se debe lograr un sistema altamente configurable y extensible, teniendo en cuenta que se desarrollará sobre la Plataforma GeoQ y que constituye una plataforma de desarrollo para ser personalizada, pudiéndose incorporar a estas nuevas funcionalidades.

## **Apariencia o interfaz externa**

- El sistema debe tener una apariencia fácil de utilizar, que guíe al usuario paso a paso hasta alcanzar el objetivo deseado.

## **Requisitos de Software**

Las estaciones de trabajo deben de cumplir con los siguientes parámetros:

- Como Sistema Operativo cualquier distribución de UNIX/Linux, así como Windows 2000 o superior.

# *Presentación de la solución propuesta*

---

- SIG GeoQ en su versión 1.5
- Las librerías: libmap.dll, pdflib.dll, oci.dll, libfcgi.dll.
- El binding de QT Creator con Python (PyQt).
- PostGis como extensión de PostgreSQL para el soporte de datos espaciales.

## **Requisitos de Hardware**

Para las PC clientes:

- Se requiere al menos 128 MB de memoria RAM.
- Se requiere al menos 40 GB de disco duro.
- Procesador 512 MHz como mínimo.

## **Requisitos de Licencia**

- GeoQ se publica bajo la Licencia Pública General de GNU (GNU GPL). Desarrollar este producto bajo esta licencia quiere decir que se puede inspeccionar y modificar el código fuente y garantiza que nuestros usuarios siempre tendrán acceso a un programa SIG gratuito y que puede ser modificado libremente. Las modificaciones están sustentadas por requisitos legales, derecho de autor y otros.
- Los datos no están sujetos a ningún derecho o garantía. Los autores y editores no se hacen responsables de fallos y sus consecuencias. Siempre se es bienvenido para indicar posibles errores.

## **Requisitos Legales, Derecho del Autor y otros**

- El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.
- La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.
- Como producto, GeoQ se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias



# *Presentación de la solución propuesta*

---

Informáticas.

## **Confidencialidad**

- La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

## **Integridad**

- La información manejada por el sistema es objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma es considerada igual a la fuente o autoridad de los datos.
- Se incluye también mecanismos de chequeo de integridad y realización de auditorías por personal calificado de la entidad.

## **Disponibilidad**

- Los usuarios autorizados (autenticados por dominio y según su rol) se les garantizará el acceso a la información, los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.
- La información y las funcionalidades del sistema estarán disponibles y el usuario podrá acceder a ellas las 24 horas de los 7 días de la semana.

## **3.6 Casos de Usos del Sistema**

Los casos de uso no son artefactos orientados a objetos, son artificios narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

# Presentación de la solución propuesta

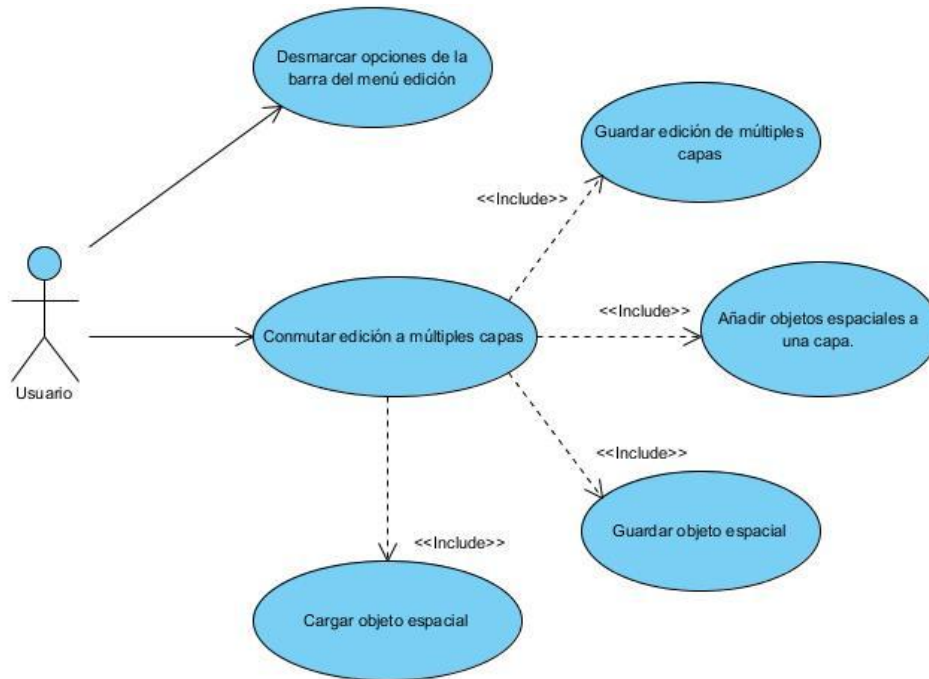


Fig. 10 Diagrama de Casos de Usos del Sistema

## 3.6.1 Especificación de Casos de Uso

En este epígrafe se muestra la descripción de los principales casos de uso del sistema del Módulo de edición para el sistema GeoQ versión 2.0. De cada uno de ellos se presenta el flujo normal de eventos y la interfaz de usuario donde se encuentran, los restantes se pueden encontrar en el expediente de proyecto de SIG-Desktop.

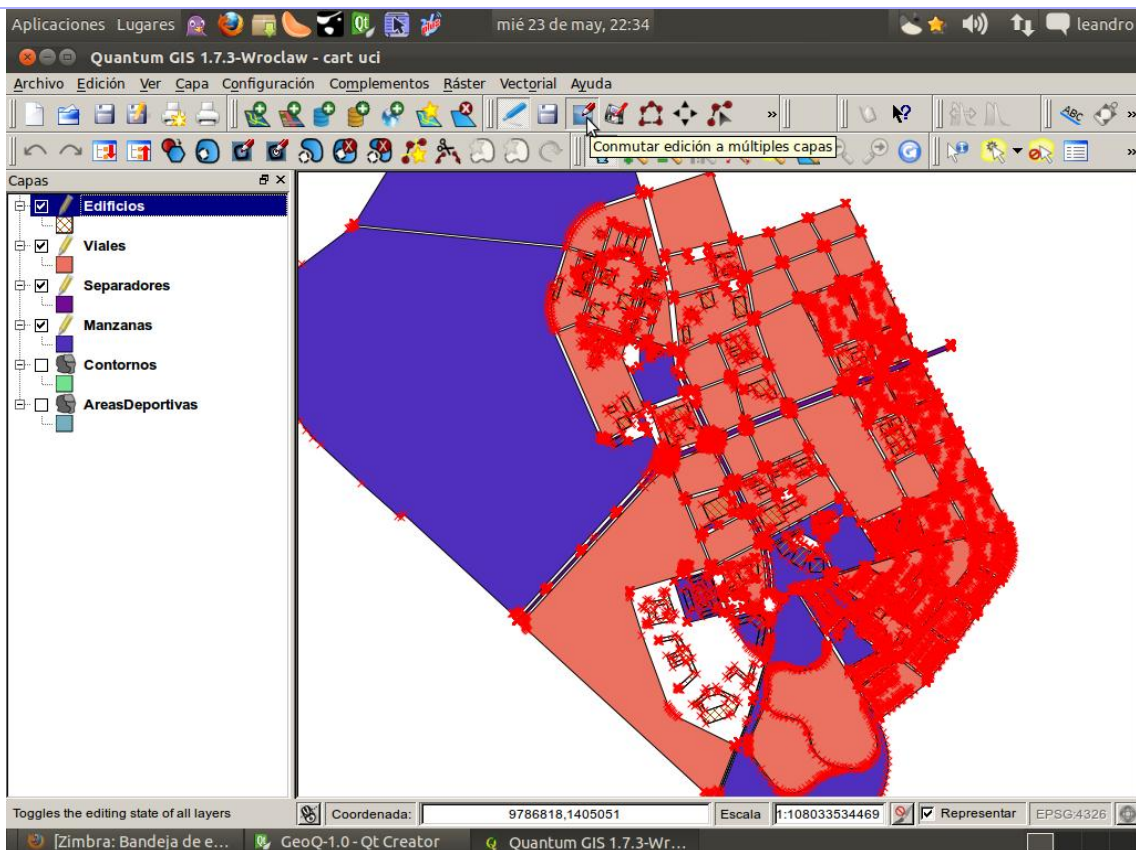
Tabla 2 DCU\_Conmutar edición a múltiples capas

|              |  |
|--------------|--|
| Caso de Uso: | Conmutar edición a múltiples capas.  |
| Actores:     | Usuario  |
| Resumen:     | El caso de uso se inicia cuando el usuario selecciona la opción “Conmutar edición a múltiples capas” el sistema permite al usuario aplicar modificaciones a todas las capas seleccionadas del mapa. El caso de uso termina cuando se ha conmutado la edición de todas las capas seleccionadas. |

# Presentación de la solución propuesta

|  |  |
|--|--|
| Precondiciones:  | Deben de existir más de una capa seleccionada sobre el mapa.                       |
| Referencias  | RF 2   |
| Prioridad  | Crítico  |
| Flujo Normal de Eventos  |  |
| Acción del Actor   | Respuesta del Sistema  |
| 1- El actor selecciona la opción “Conmutar edición a múltiples capas”. | 2- El sistema visualiza todas las capas seleccionadas como listas para la edición. |

## Prototipo de Interfaz



# Presentación de la solución propuesta

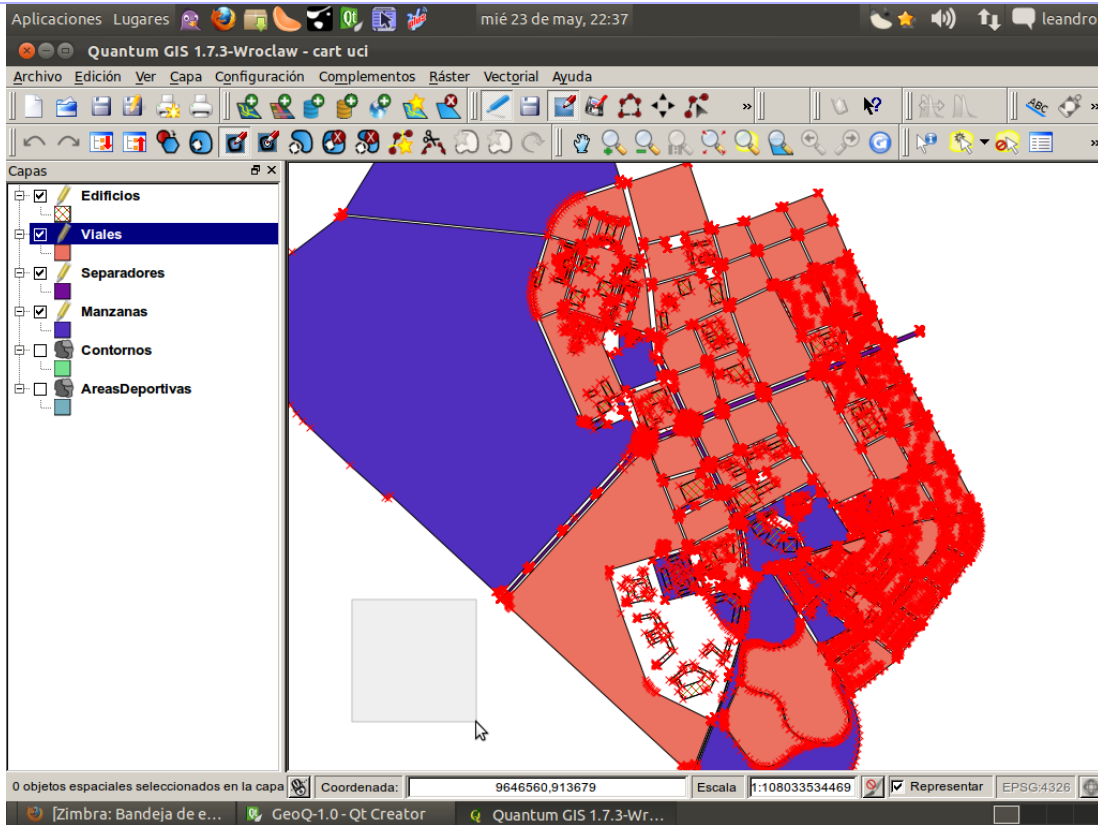
**Tabla 3 DCU\_ Añadir objetos espaciales a una capa**

|   |   |  |
|---|---|--|
| Caso de Uso:  | Añadir objetos espaciales a una capa.   |  |
| Actores:  | Usuario   |  |
| Resumen:  | El caso de uso se inicia cuando el usuario desea añadir un nuevo objeto en alguna de las capas seleccionadas. El sistema brinda las opciones de: "Añadir rectángulo" o "Añadir círculo". Permite modificar las nuevas figuras convirtiéndolas en polígonos. El caso de uso termina cuando se añade el nuevo objeto. |  |
| Precondiciones  | Debe estar seleccionada la opción "Conmutar edición" o "Conmutar edición a múltiples capas" y una capa de tipo polígono. Los valores a introducir deben de coincidir con los valores de entrada definidos en la capa seleccionada.  |  |
| Referencias   | RF 4, RF 5  |  |
| Prioridad   | Crítico   |  |
| <b>Flujo Normal de Eventos</b>  |   |  |
| <b>Acción del Actor</b>   | <b>Respuesta del Sistema</b>  |  |
| 1- El caso de uso se inicia cuando el usuario selecciona la opción "Conmutar edición" o "Conmutar edición a múltiples capas". | 2- El sistema habilita las opciones: "Añadir rectángulo" y "Añadir círculo".  |  |

# Presentación de la solución propuesta

|  |  |
|--|--|
| <p>3- El usuario selecciona alguna de las siguientes opciones:</p> <ol style="list-style-type: none"><li>1. Añadir rectángulo. Ver sección "Añadir rectángulo".</li><li>2. Añadir círculo. Ver sección "Añadir círculo".</li></ol> | <p>4- El sistema se remite a la opción seleccionada por el usuario.</p>  |
| <b>Sección "Añadir rectángulo".</b>  |  |
| <b>Acción del Actor</b>  | <b>Respuesta del Sistema</b>   |
| <p>1. El usuario presiona clic izquierdo en el lugar donde desea añadir el rectángulo y arrastra el cursor hasta soltar el clic.</p>   | <p>2. El sistema muestra un nuevo formulario dándole la posibilidad al usuario de ingresar los valores de identificación del objeto, del tipo de dato especificado por el usuario en la creación de la capa.</p> |
| <p>3. El usuario llena los campos correspondientes.</p>  | <p>4. El sistema añade el nuevo objeto adquiriendo las propiedades de la capa seleccionada.</p>  |
| Prototipo de Interfaz  |  |

# Presentación de la solución propuesta

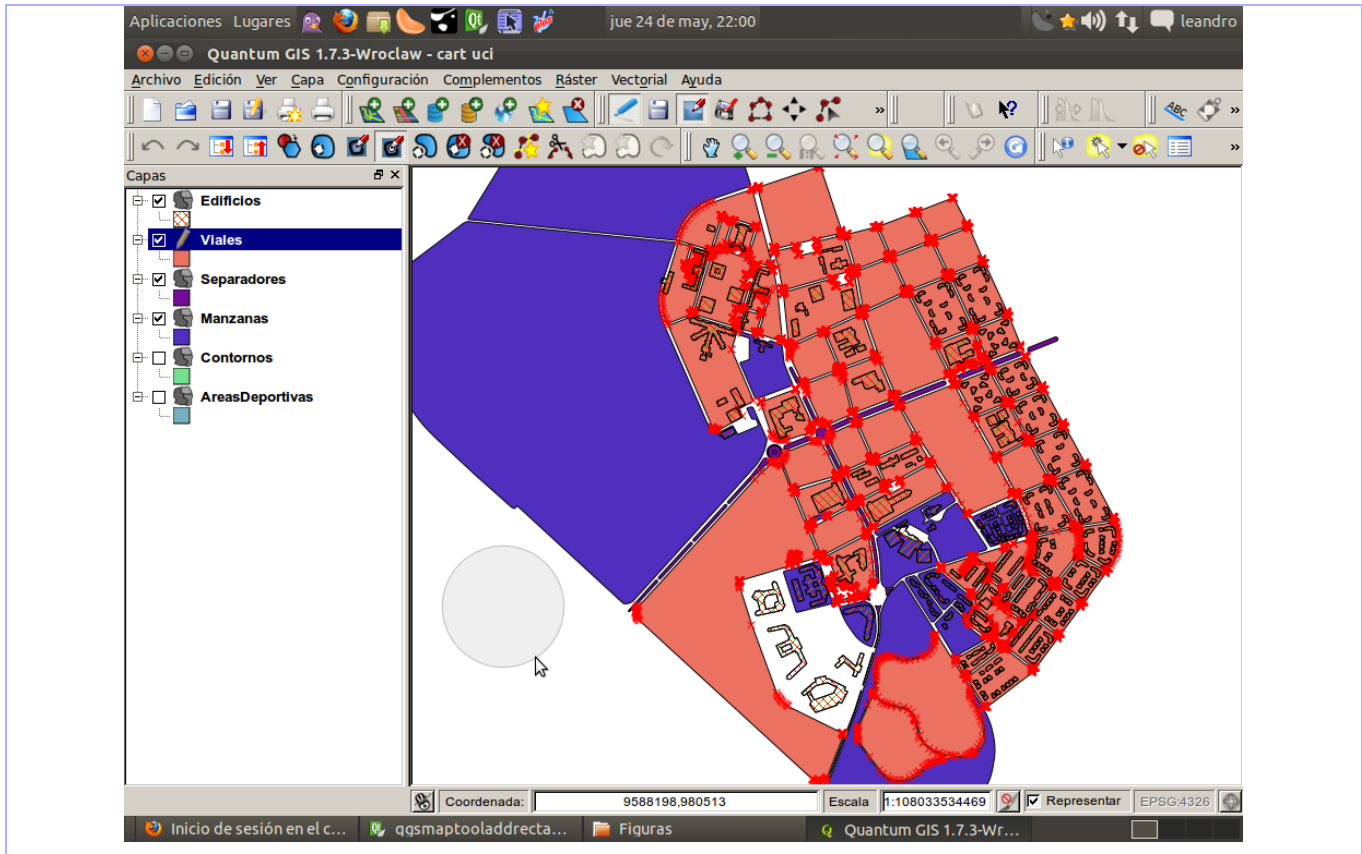


## Sección "Añadir círculo".

| Acción del Actor   | Respuesta del Sistema   |
|--|---|
| 1. El usuario presiona clic izquierdo en el lugar donde desea añadir el círculo y arrastra el cursor hasta soltar el clic. | 2. El sistema muestra un nuevo formulario dándole la posibilidad al usuario de ingresar los valores de identificación del objeto, del tipo de dato especificado por el usuario en la creación de la capa. |
| 3. El usuario llena los campos correspondientes.   | 4. El sistema añade el nuevo objeto adquiriendo las propiedades de la capa seleccionada.  |

Prototipo de Interfaz

# Presentación de la solución propuesta

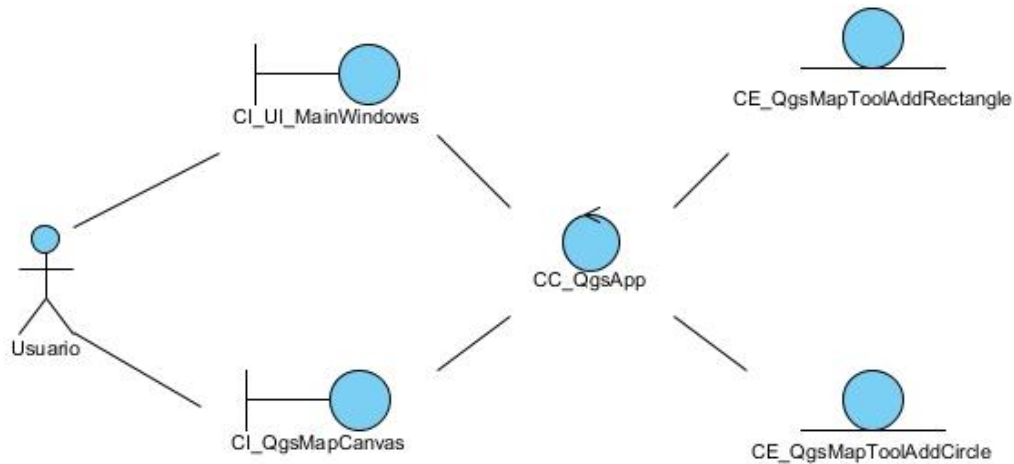


## 3.7 Modelo de Análisis

Durante el análisis de los requisitos se genera la especificación de las características operacionales del sistema así como las restricciones a las que este debe responder, generándose el artefacto diagrama de clases del análisis donde se representan los conceptos de dominio del problema. Incorpora la actividad del mundo real su funcionamiento, no la implementación automatizada del mismo.

El diagrama de clases es la representación principal del diseño y el análisis para un sistema. En él, la estructura de clases se especifica a través de relaciones entre ellas y las distintas estructuras de herencia que se establecen. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal, en el diseño, se parte del mismo diagrama y se modifica para satisfacer los detalles de la implementación. (Centers, 2011)

# Presentación de la solución propuesta



**Fig. 11 Diagrama de Clases del Análisis CU\_Añadir objetos espaciales a una capa**

En la Fig. 11 se puede apreciar el diagrama de clases del análisis para el caso de uso Añadir objetos espaciales a una capa, identificándose la clase **CI\_UI\_MainWindows** donde a través de ella se realiza la petición del resultado de la acción que desea realizar el usuario. Aparecen las clases **CE\_QgsMapToolAddRectangle** y **CE\_QgsMapToolAddCircle** encargadas de generar los objetos espaciales. El contenido será interpretado por la clase principal de GeoQ, la **CC\_QgsApp** y mostrado en la **CI\_QgsMapCanvas**. Los restantes diagramas se pueden encontrar en el expediente de proyecto de SIG-Desktop.

## 3.7.1 Diagramas de interacción

Los diagramas de interacción o comportamiento muestran la interacción entre los objetos del sistema, la forma en que se comunican y su comportamiento de modo general. Existen dos tipos de diagramas de interacción, que se realizan fundamentados en la misma información, sin embargo, cada uno de acuerdo a su objetivo hace mayor énfasis en un aspecto particular. (Centers, 2011)

### 3.7.1.1 Diagramas de secuencia

Un diagrama de secuencia muestra una interacción ordenada según la secuencia temporal de eventos, los objetos que participan en la interacción y los mensajes que estos intercambian ordenados según su secuencia en el tiempo. Para la modelación se utiliza un eje vertical que representa el tiempo, este fluye

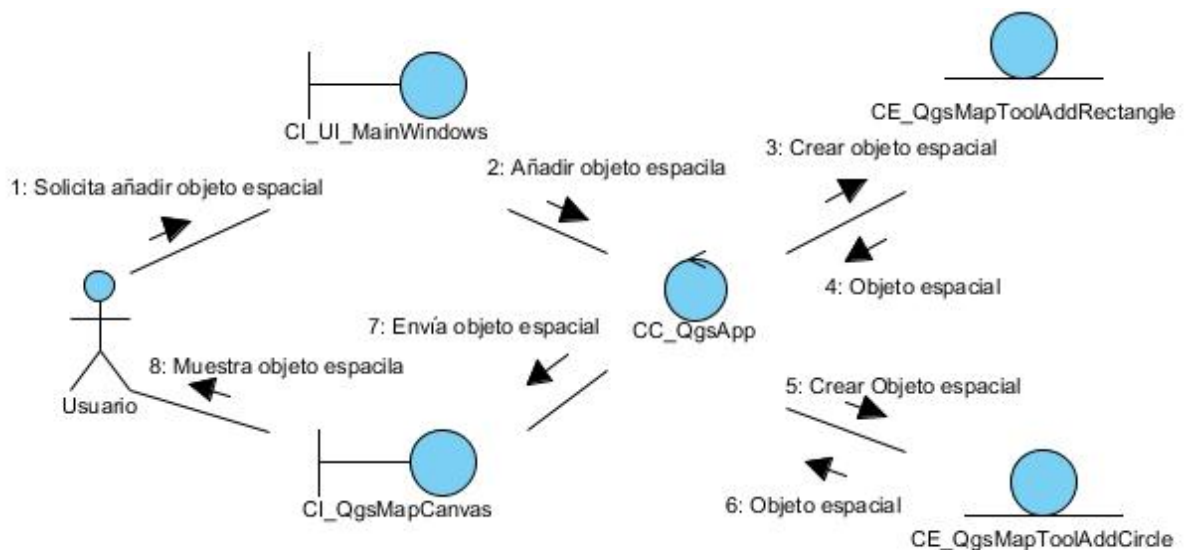


# Presentación de la solución propuesta

de arriba hacia abajo, mientras en el eje horizontal se colocan los objetos y actores participantes en la interacción. En este tipo de diagramas los mensajes se clasifican de acuerdo a cómo se producen en el tiempo: simples, síncronos, y asíncronos. (Centers, 2011)

## 3.7.1.2 Diagramas de colaboración

Un diagrama de colaboración muestra la interacción entre los objetos y los enlaces existentes entre ellos, indicando el flujo de mensajes entre los elementos del modelo. Durante la ejecución de un diagrama de colaboración se crean y destruyen objetos y enlaces. A diferencia de los diagramas de secuencia los diagramas de colaboración muestran la relación entre los objetos no la secuencia en el tiempo en que se producen los mensajes. (Centers, 2011)



**Fig. 12 Diagrama de Colaboración CU\_Añadir objetos espaciales a una capa**

En la Fig. 12 se puede apreciar el diagrama de colaboración para el caso de uso Añadir objetos espaciales a una capa. Los restantes diagramas se pueden encontrar en el expediente de proyecto de SIG-Desktop.

## **3.8 Modelo de Diseño**

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizado como entrada fundamental de las actividades de implementación. (Centers, 2011)

### **3.8.1 Patrones empleados en el diseño**

Para un correcto entendimiento de la plataforma GeoQ se necesita centrar la investigación en su arquitectura, enfocada para el trabajo con mapas. El modelo arquitectónico del SIG GeoQ está confortado bajo la aplicación diversos patrones que deben respetarse a la hora de realizar la personalización de dicha plataforma.

#### **Patrones de Arquitectura**

**Patrón de arquitectura orientada a objetos:** Los componentes del estilo se basan en principios orientados a objetos: encapsulamiento, herencia y polimorfismo. Son así mismo las unidades de modelado, diseño e implementación, los objetos y sus interacciones son el centro en el diseño de la arquitectura en la estructura de la aplicación.

**Patrón de arquitectura basada en componentes:** Las características principales de este patrón son la modularidad, la reusabilidad y compatibilidad. En la arquitectura basada en componentes también se requiere robustez ya que los componentes han de operar en entornos mucho más heterogéneos y diversos. Su premisa es que los componentes cumplan con alta cohesión y bajo acoplamiento. (Centers, 2011)

#### **Patrones de diseño**

##### **Patrón Creador**

- La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos.

# *Presentación de la solución propuesta*

---

- En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella.
- El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento.

## **Patrón Observador**

Este patrón define una dependencia entre un objeto y un conjunto de ellos, de modo que los cambios en el primero se vean reflejados en los otros, es decir, cuando un objeto cambie de estado se notifica y actualizan automáticamente todos los objetos que dependen de él.

**Patrón Bajo Acoplamiento:** Acoplamiento bajo significa que una clase no depende de muchas clases.

El patrón brinda los siguientes beneficios:

- No se afectan las clases por cambios realizados a otros componentes.
- Fácil de entender por separado.

Mejora la reutilización de las clases.

**Patrón Singleton (Solitario):** En el diseño de clases es necesario aplicar la solución del patrón Singleton que no es más que garantizar el acceso único a una clase mediante una única instancia. De esta forma se controla el acceso a las clases. (Centers, 2011)

Como principio de diseño del sistema se tienen:

- Una apariencia fácil de utilizar de manera que pueda ser usado por personas con conocimiento básico en SIG.
- Una iconografía descriptiva, es decir, que cada ícono se ajuste a la funcionalidad que represente.
- Los nombres de los atributos comenzarán con letra minúscula y los métodos con letra mayúscula.

## 3.8.2 Diagramas de clases del diseño

El diagrama de clases del diseño presentado a continuación está basado en la implementación que se realiza en el SIG GeoQ con el framework Qt, además de estar diseñado bajo una arquitectura orientada a objetos. Debido a la complejidad y la extensión de la herramienta, incluyendo específicamente todas las clases contenidas en el framework, y siguiendo la premisa de ajustarse al flujo propio de la solución y no a los detalles del equipo de desarrollo, se decidió representar únicamente las clases que tuvieran que ver directamente con la construcción de la solución, sin dejar de incluir las partes fundamentales del framework que ayuden a su comprensión.

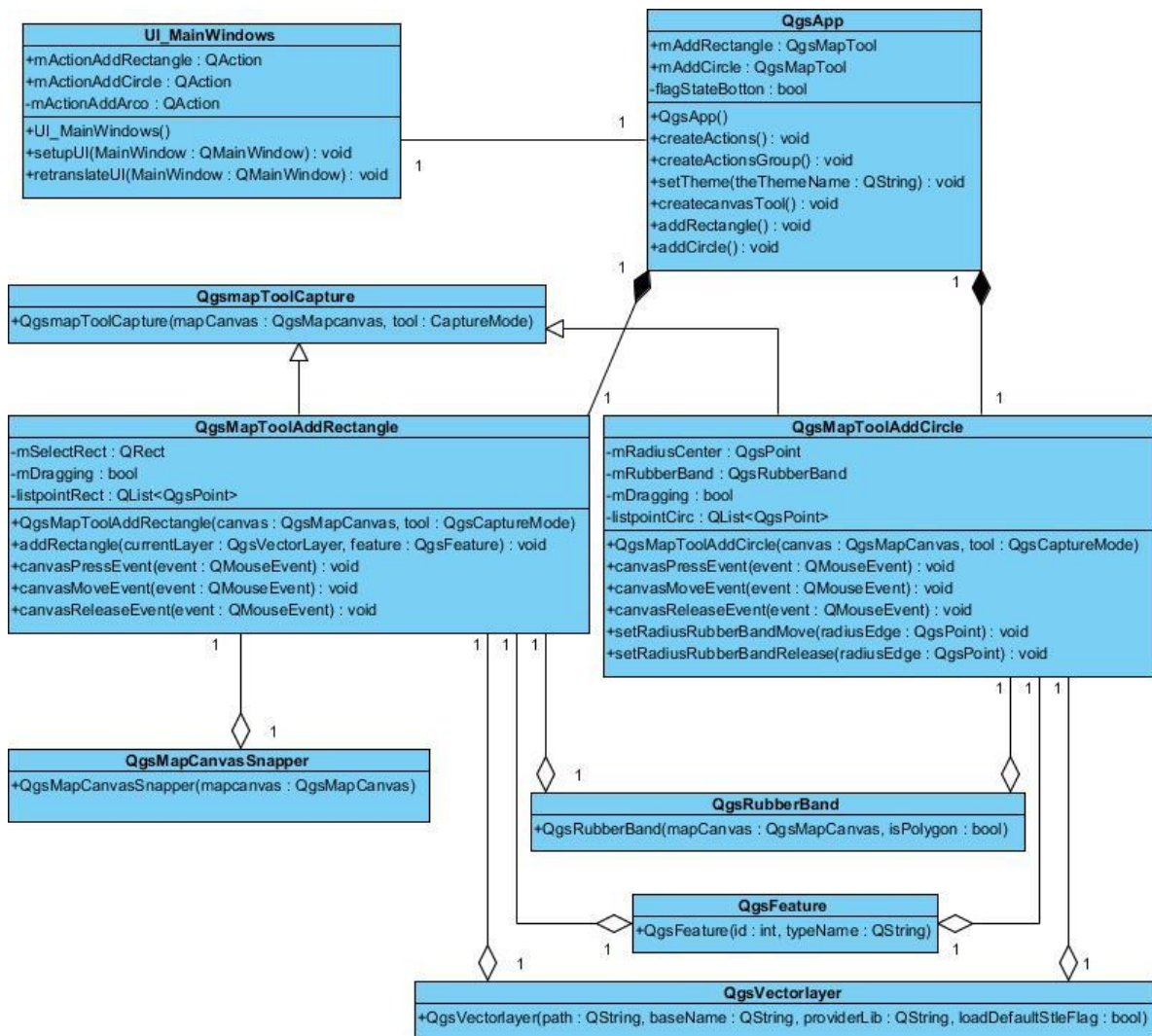
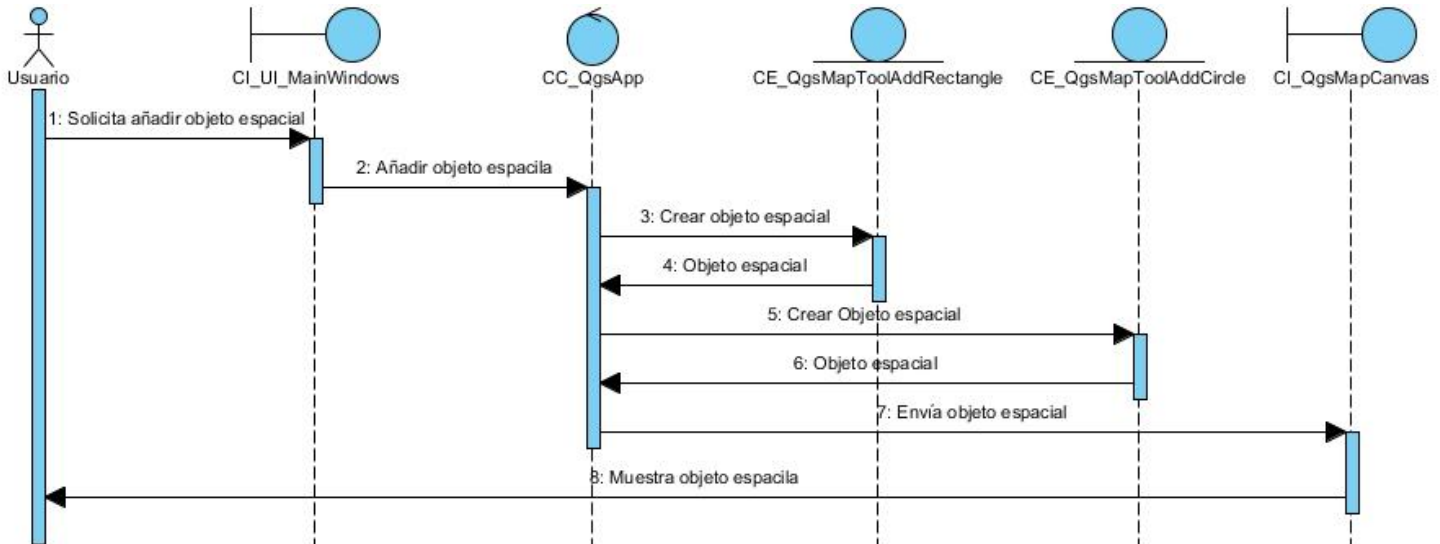


Fig. 13 Diagrama de clases del diseño CU\_Añadir objetos espaciales a una capa

# Presentación de la solución propuesta

En la Fig. 13 se puede apreciar el diagrama de clases del diseño para el caso de uso Añadir objetos espaciales a una capa. Los restantes diagramas se pueden encontrar en el expediente de proyecto de SIG-Desktop.

## 3.8.3 Diagramas de interacción



**Fig. 14 Diagrama de Secuencia CU\_Añadir objetos espaciales a una capa**

En la Fig. 14 se puede identificar el diagrama de secuencia para el caso de uso Añadir objetos espaciales a una capa. Aparecen representados en el diagrama los objetos que intervienen en la realización del mismo y la interacción entre estos así como el intercambio de mensajes que realizan. Los restantes diagramas se pueden encontrar en el expediente de proyecto de SIG-Desktop.

## 3.9 Conclusiones parciales

En este capítulo se trata la descripción de la solución propuesta, para ello se desarrolla un modelo de dominio para aplicaciones que no tienen procesos de negocio definido. El modelo de dominio permitió representar de forma más específica las entidades que se encuentran en el lugar donde tiene desarrollo el sistema. La descripción de los requisitos y de casos de usos del sistema permitió definir las necesidades objetivas de la investigación así como los recursos necesarios para la misma. Estas necesidades fueron convertidas a funcionalidades del sistema y descritas para un mejor entendimiento.

## *Presentación de la solución propuesta*

---

Se expusieron los diagrama de clases y los diagrama de interacción correspondientes al análisis y diseño y se identificó la arquitectura ajustada para el desarrollo del sistema, que va a contribuir a la adecuada implementación del mismo.

# *Construcción de la solución propuesta*

---

## *CAPÍTULO IV: Construcción de la solución propuesta*

### **4.1 Introducción**

En este capítulo se abordan temas fundamentales para la construcción del sistema propuesto. Se recogen las generalidades del Flujo de Trabajo de Implementación así como se presentan las pruebas realizadas al sistema desarrollado.

### **4.2 Generalidades de la implementación**

Los constantes avances en la rama de las TIC a la par de las crecientes necesidades de sus usuarios, han provocado que la industria de software se plantee nuevos mecanismos para lograr desarrollar aplicaciones en menor tiempo y con un mayor nivel de escalabilidad.

El sistema a desarrollar necesita ser escalable, es por ello que para su implementación se utiliza un mecanismo integrado a una arquitectura orientada a objetos y basada en componentes. De igual forma se hizo necesaria la utilización de varios patrones de diseño, es el caso del patrón Singleton: el sistema tiene un único punto de entrada utilizando para ello la misma instancia de la clase controladora para todas las peticiones; el patrón Creador: el sistema permite conectar con el objeto producido en cualquier evento, permitiendo captar dinámicamente las dependencias entre objetos, de tal forma que un objeto notificará a los objetos dependientes de él cuando cambia su estado, siendo actualizados automáticamente.

Para garantizar la reutilización y entendimiento de los códigos desarrollados se realizó la programación siguiendo estándares de codificación previamente definidos, donde por ejemplo, el nombre de métodos y clases comienzan en minúscula (`addRectangle`), para el caso de las variables se identifica primero el tipo de dato y luego un nombre referente a la información para el que están destinados (`vlayer`). Las clases se encuentran distribuidas por carpetas, separadas de acuerdo a la estructura definida por capas en el modelo de clases del diseño, logrando así la utilización y optimización de códigos y clases implementadas sin importar el módulo al cual correspondan.

## 4.3 Modelo de Implementación

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas y mostrar las relaciones entre sus elementos. Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. (Alba, 2010)

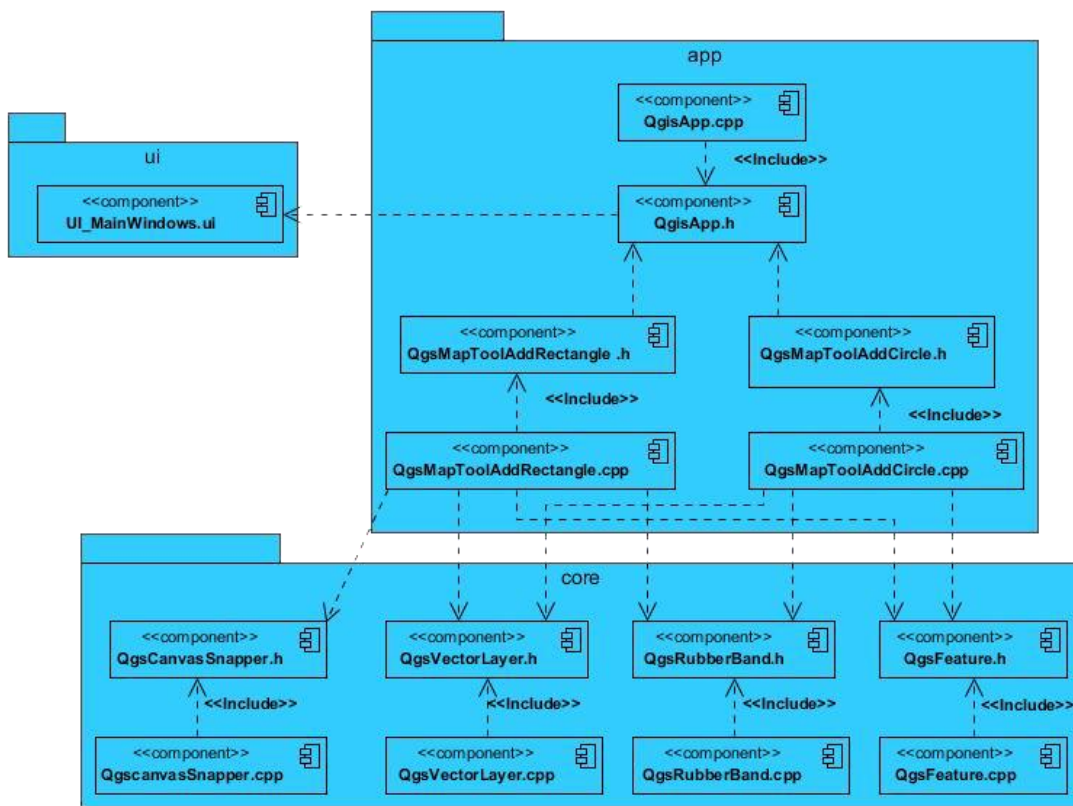


Fig. 15 Diagrama de Componentes CU\_ Añadir objeto espaciales a una capa

En la Fig. 15 se puede apreciar el diagrama de componentes para el caso de uso Añadir objetos espaciales a una capa. Los restantes diagramas se pueden encontrar en el expediente de proyecto de SIG-Desktop.

## 4.4 Modelo de Despliegue

El Modelo de Despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla los requisitos de hardware y otra



# Construcción de la solución propuesta

información relacionada al despliegue del sistema propuesto. Muestra dónde y cómo se desplegarán los componentes. Es un mapa específico de la instalación física del sistema. Un diagrama de despliegue ilustra el despliegue físico del sistema en un ambiente de producción (o prueba). Muestra dónde se ubicarán los componentes, en qué máquinas o hardware. (Ltd, 2007)



**Fig. 16 Diagrama de Despliegue**

## 4.5 Validación del sistema propuesto

La prueba es un elemento crítico para la garantía de la calidad del software, es el proceso que permite verificar y revelar la calidad de este tipo de producto. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. Con la realización de estas pruebas se pretende encontrar y documentar los defectos que puedan afectar la calidad del sistema desarrollado y validar los requisitos que debe cumplir el mismo y a su vez verificar si estos fueron implementados correctamente.

Las pruebas no aseguran que el software presenta ausencia de defectos pero si demuestra que el software puede tener defectos. Existen diferentes tipos de pruebas que se le pueden aplicar al software para identificar fallos en la aplicación, como son:

- ✓ Pruebas de Integración: Comprueban la compatibilidad y funcionalidad de las interfaces entre los distintos elementos que componen un sistema.
- ✓ Pruebas de Validación: Son realizadas sobre un software completamente integrado para evaluar el cumplimiento con los requisitos especificados.

# *Construcción de la solución propuesta*

---

- ✓ Prueba de Caja Blanca: Se basa en el diseño de casos de prueba que usen la estructura de control del diseño procedimental para derivarlos, en otras palabras se analiza la estructura lógica del programa.
- ✓ Prueba de Caja Negra: Se centra principalmente en los requisitos funcionales del software reflejados en su interfaz sin tener en cuenta el funcionamiento interno de la aplicación, no considera la codificación dentro de los parámetros a evaluar. Se basa en que las entradas sean aceptadas de forma adecuada y se reciba una salida correcta demostrando que cada función es completamente operativa.
- ✓ Prueba de Aceptación: Es la prueba final basada en las especificaciones del usuario. Su objetivo principal es demostrarle el cumplimiento del requisito de software al usuario. Puede estar asociado tanto a requisitos funcionales como no funcionales y cada requisito puede tener una o más pruebas de aceptación asociada.

Las pruebas a llevar a cabo sobre el sistema desarrollado responden al método de prueba de caja negra utilizando específicamente la técnica de partición de equivalencia, cuyo propósito es evaluar los requisitos funcionales del programa. Se realiza para verificar el comportamiento del sistema sin tener en cuenta cómo se implementa.

## **4.5.1 Prueba de Caja Negra**

Para desarrollar la prueba de caja negra existen varias técnicas entre ellas:

- ✓ Técnica de la Partición de Equivalencia: Divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- ✓ Técnica del Análisis de Valores Límites: Prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- ✓ Técnica de Grafos de Causa-Efecto: Permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

# Construcción de la solución propuesta

## 4.5.2 Diseño de Caso de Prueba Caja Negra. Partición equivalente

En este epígrafe se muestra la descripción del diseño de caso de prueba para el caso de uso del sistema Añadir objetos espaciales a una capa. Las restantes descripciones se pueden encontrar en el expediente de proyecto de SIG-Desktop.

**Nombre del Caso de Uso:** Añadir objetos espaciales a una capa.

**Descripción general:** El caso de uso se inicia cuando el usuario desea añadir un nuevo objeto en alguna de las capas seleccionadas. El sistema brinda las opciones de: “Añadir rectángulo” o “Añadir círculo”. El caso de uso termina cuando se añade el nuevo objeto.

**Condiciones de ejecución:** Debe estar seleccionada una capa de tipo polígono. Los valores a introducir solo pueden ser los valores definidos como entrada en la capa.

**Secciones a probar en el Caso de Uso.**

**Tabla 4 DCP CU\_ Añadir objetos espaciales a una capa**

| Nombre de la Sección                          | Escenarios de la Sección     | Descripción de la Funcionalidad  | Flujo Central  |
|---|------------------------------|--|--|
| SC 1: “Añadir objetos espaciales a una capa”. | EC 1.1: “Añadir rectángulo”. | El usuario presiona el clic izquierdo en el lugar donde desea añadir el rectángulo y arrastra el cursor hasta soltar el clic. El sistema muestra un nuevo formulario dándole la posibilidad al usuario de ingresar los valores de identificación del objeto. El usuario llena los campos correspondientes. El sistema añade el nuevo objeto con las propiedades de la capa seleccionada. | Módulo Edición.<br>1- Seleccionar la opción “Añadir rectángulo”. |

# Construcción de la solución propuesta

|  |   |   |   |
|--|---|---|---|
|  | EC 1.2: "Añadir círculo".                                 | El usuario presiona el clic izquierdo en el lugar donde desea añadir el círculo y arrastra el cursor hasta soltar el clic. El sistema muestra un nuevo formulario dándole la posibilidad al usuario de ingresar los valores de identificación del objeto. El usuario llena los campos correspondientes. El sistema añade el nuevo objeto con las propiedades de la capa seleccionada. | Módulo Edición.<br>1- Seleccionar la opción "Añadir círculo".                       |
|  | EC 1.3: "Añadir objeto espacial fuera del área del mapa". | El usuario presiona el clic izquierdo en el lugar donde desea añadir el objeto espacial y arrastra el cursor hasta soltar el clic, seleccionando el área fuera del mapa. El sistema no responde ante la situación anterior.   | Módulo Edición.<br>1- Seleccionar la opción "Añadir rectángulo" o "Añadir círculo". |

## Matriz de datos.

**SC1:** Opciones de búsqueda.

| ID del escenario | Escenario | Respuesta del sistema | Resultado de la prueba |
|------------------|-----------|-----------------------|------------------------|
|------------------|-----------|-----------------------|------------------------|

# Construcción de la solución propuesta

|        |   |  |                |
|--------|---|--|----------------|
| EC 1.1 | Añadir rectángulo                               | El sistema muestra un nuevo formulario dándole la posibilidad al usuario de ingresar los valores de identificación del rectángulo. El sistema añade el nuevo objeto adquiriendo las propiedades de la capa seleccionada. | Satisfactorio  |
| EC 1.2 | Añadir círculo                                  | El sistema muestra un nuevo formulario dándole la posibilidad al usuario de ingresar los valores de identificación del círculo. El sistema añade el nuevo objeto adquiriendo las propiedades de la capa seleccionada.    | Satisfactorio. |
| EC 1.3 | Añadir objeto espacial fuera del área del mapa. | El sistema no responde ante la situación.  | Satisfactorio. |

## 4.6 Conclusiones Parciales

Durante el capítulo se abordaron las generalidades para el desarrollo de la aplicación, teniendo en cuenta la arquitectura seleccionada y los componentes utilizados, constatando que la aplicación de patrones de diseño asegurará la escalabilidad y robustez de la solución. Además, con la creación del modelo de despliegue se obtuvo una vista de la distribución física del sistema logrando una mayor explotación de sus funcionalidades. Se realizaron pruebas de caja negra al sistema arrojando resultados satisfactorios y dos no conformidades resueltas satisfactoriamente.

## *Conclusiones Generales*

En la presente investigación se profundizó en los conocimientos sobre los procesos de edición de cartografías en los SIG, procedentes de fuentes diversas en diferentes publicaciones digitales logrando organizarlos en función de la fundamentación del objetivo propuesto.

Las pruebas realizadas a la aplicación con el objetivo de validar los requisitos funcionales determinados para el desarrollo de la segunda versión de su Módulo de Edición, reveló un sistema donde se arrojaron resultados favorables y no conformidades.

Los resultados de este trabajo posibilitan la factibilidad tecnológica de la utilización de la segunda versión del módulo de edición del SIG GeoQ en el proyecto de desarrollo SIG-Desktop, perteneciente al departamento de Geoinformática de la UCI.

Se logró desarrollar la segunda versión del módulo de edición del SIG GeoQ, con la característica de ser más abarcador que la versión anterior. Además se asegura la escalabilidad del sistema por la integralidad de sus funcionalidades y las técnicas empleadas en el desarrollo de las mismas, lo cual permite actualizar el software de manera dinámica.

### *Recomendaciones*

En aras de lograr un manejo eficaz del sistema desarrollado se recomienda la inclusión en el manual de ayuda para el usuario de la nueva versión del módulo de edición, así como capacitar al personal para la utilización del mismo.

Desarrollar nuevas funcionalidades de edición cartográfica con la utilización directa de librerías de Qt, con el propósito de fortalecer la usabilidad de la aplicación y agregar nuevas funcionalidades al sistema.

Socializar la solución propuesta a la comunidad de desarrollo en la universidad.

## Referencias Bibliográficas

1. **Gordon, Carlos. 2008.** slideshare. [En línea] Octubre de 2008. [Citado el: 21 de Octubre de 2011.] <http://www.slideshare.net/carlosgordonb/introduccion-sig>.
2. **Araya, Melvin Lizano. 2011.** Scribd. [En línea] Julio de 2011. [Citado el: 24 de Octubre de 2011.] <http://es.scribd.com/doc/51138833/Definicion-SIG>.
3. **Ramos, Edgar Loyo. 2000.** CIRIA. [En línea] 1 de Septiembre de 2000. [Citado el: 24 de Octubre de 2011.] [catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/...l.../capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/...l.../capitulo4.pdf).
4. **Caldas, Universidad Distrital Francisco José de. 2011.** Sistema de Informacion Geografica. [En línea] 12 de Julio de 2011. [Citado el: 24 de Octubre de 2011.] [http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/datos\\_espaciales.htm](http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/datos_espaciales.htm).
5. **Geaintec. 2011.** GEAINTEC. [En línea] 2011. [Citado el: 24 de oCTUBRE de 2011.] <http://www.geaintec.cl/servicios/informacion-sig/analisis-modelado/analisis-espacial/>.
6. **Caldas, Universidad Distrital Francisco José de. 2010.** Sistema de Informacion Geográfica. [En línea] 2 de Agosto de 2010. [Citado el: 25 de Octubre de 2011.] <http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/metadatos.htm>.
7. **Rica, San Jose - Costa. 2010.** Mapealo.com. [En línea] 2010. [Citado el: 25 de Octubre de 2011.] [www.mapealo.com/Costaricageodigital/.../intro\\_carto.pdf](http://www.mapealo.com/Costaricageodigital/.../intro_carto.pdf) - Costa Rica.
8. **Rico, José Lorenzo Herrero. 2000.** Dpto. tecnologías de la Informacion las Comunicaciones. [En línea] Marzo de 2000. [Citado el: 25 de Octubre de 2011.] [www.tic.udc.es/~nino/blog/documentos/cartografia-digital.pdf](http://www.tic.udc.es/~nino/blog/documentos/cartografia-digital.pdf).
9. **Murcia, Universidad de. 2006.** SIGMUR SIG y Teledetección en la Universidad de Murcia. [En línea] 2006. [Citado el: 28 de Octubre de 2011.] [http://www.um.es/geograf/sigmur/sigpdf/temario\\_8.pdf](http://www.um.es/geograf/sigmur/sigpdf/temario_8.pdf).
10. **Geomática, Universidad de Melbourne - Departamento de. 2001.** autoaprendizaje sig. [En línea] 7 de Marzo de 2001. [Citado el: 28 de Octubre de 2011.] [http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GIST\\_Raster..](http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GIST_Raster..)
11. **Larenas, Carlos Gonzáles. 2010.** MV Blog. [En línea] 29 de Noviembre de 2010. [Citado el: 28 de Octubre de 2011.] <http://hsutem.puntocl.com/tmv/2010/11/30/rhinoceros-layers/>.



12. **savgis.org. 2011.** savgis. [En línea] 7 de Abril de 2011. [Citado el: 1 de Noviembre de 2011.] <http://www.savgis.org/download/download.php>.
13. **España, Esri. 2010.** esri España. [En línea] 2010. [Citado el: 4 de Noviembre de 2011.] <http://www.esri.es/es/productos/arcgis/>.
14. **Ambiental, Secretaria de Desarrollo Sustentable y Política. 2011.** Scribd. [En línea] 2011. [Citado el: 4 de Noviembre de 2011.] <http://es.scribd.com/doc/57016229/Pinbn-Manual-Cartografia-Sig>.
15. **Puebla, Universidad de las Americas. 2011.** CIRIA UDLAP. [En línea] Noviembre de 2011. [Citado el: 4 de Noviembre de 2011.] [http://catarina.udlap.mx/u\\_dl\\_a/tales/](http://catarina.udlap.mx/u_dl_a/tales/).
16. **Geoinfo. 2010.** Softwaregis. [En línea] 2010. [Citado el: 4 de Noviembre de 2011.] <http://www.softwaregis.cl/arcgis.html>.
17. **Corel, Club Internacional de Usuarios de. 2011.** CORELCLUB.ORG. [En línea] 2011. [Citado el: 4 de NOVIEMBRE de 2011.] <http://www.corelclub.org/cartografia-con-coreldraw/>.
18. **Fundación, W. Lenguaje Unificado de Modelado. 2011.** EcuRed. [En línea] 2011. [Citado el: 2 de Diciembre de 2011.] [http://www.ecured.cu/index.php/Lenguaje\\_de\\_Modelaje\\_Unificado](http://www.ecured.cu/index.php/Lenguaje_de_Modelaje_Unificado).
19. **Murcia, Universidad de. 2011.** Universidad de Murcia. [En línea] 23 de Mayo de 2011. [Citado el: 2 de Diciembre de 2011.] [http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE\\_principales.html](http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html).
20. **IBM. 2011.** IBM. [En línea] 2011. [Citado el: 2 de Diciembre de 2011.] <http://www-01.ibm.com/software/awdtools/developer/rose/>.
21. **EcuRed. 2011.** EcuRed. [En línea] 2011. [Citado el: 2 de Diciembre de 2011.] [http://www.ecured.cu/index.php/Visual\\_Paradigm](http://www.ecured.cu/index.php/Visual_Paradigm).
22. **Lanzillotta, Analía. 2005.** MASTERMAGAZINE. [En línea] 12 de Febrero de 2005. [Citado el: 5 de Diciembre de 2011.] <http://www.mastermagazine.info/termino/5560.php>.
23. **Virtual, Pergamino. 2011.** Pergaminovirtual. [En línea] 2011. [Citado el: 5 de Diciembre de 2011.] <http://www.pergaminovirtual.com/definicion/C.html?PHPSESSID=9a75f4228bde94e4cf0cafade534bc89>.
24. **Systems, Zator. 2011.** Zator Systems. [En línea] 2011. [Citado el: 5 de Diciembre de 2011.] [http://www.zator.com/Cpp/E1\\_2.htm](http://www.zator.com/Cpp/E1_2.htm).
25. **EcuRed. 2011.** EcuRed. [En línea] 2011. [Citado el: 6 de Diciembre de 2011.] [http://www.ecured.cu/index.php/IDE\\_de\\_Programaci%C3%B3](http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3).

26. **Qt, Comunidad de usuarios de. 2010.** Zona Qt. [En línea] 25 de septiembre de 2010. [Citado el: 06 de diciembre de 2011.] <http://www.zonaqt.com/tutoriales/tutorial-b%C3%A1sico-de-qt-4>.
27. **IBM. 2011.** IBM. [En línea] Agosto de 2011. [Citado el: 07 de Diciembre de 2011.] <http://www-01.ibm.com/software/awdtools/rup/>.
28. **EcuRed. 2011.** EcuRed. [En línea] 22 de mayo de 2011. [Citado el: 07 de Diciembre de 2011.] [http://www.ecured.cu/index.php/Programaci%C3%B3n\\_Extrema\\_%28XP%29](http://www.ecured.cu/index.php/Programaci%C3%B3n_Extrema_%28XP%29).
29. **Milagro, Universidad Estatal de. 2010.** slideshare. [En línea] 28 de Julio de 2010. [Citado el: 07 de Diciembre de 2011.] <http://www.slideshare.net/bebeyom/metodologia-msf-4861508>.
30. **Centers, UML Value Added Trainig. 2011.** MILESTONE Consulting. [En línea] 23 de Mayo de 2011. [Citado el: 08 de Diciembre de 2011.] <http://www.milestone.com.mx/CursoArquitecturaSW.htm>.
31. **Alba, Eduardo Rivera. 2010.** Scribd. [En línea] 10 de Octubre de 2010. [Citado el: 09 de Diciembre de 2011.] <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
32. **Ltd, Sparx Systems Pty. 2007.** Enterprise Architect. [En línea] Sparx Systems Pty Ltd, 2007. [Citado el: 10 de Diciembre de 2011.] [http://www.sparxsystems.com.ar/resources/tutorial/physical\\_models.html](http://www.sparxsystems.com.ar/resources/tutorial/physical_models.html).

## Bibliografía Citada

1. Aprenda C++ San Sebastian Universidad de Navarra 1998
2. Business Modelin with UML Authors of UML Toolkit
3. El lenguaje de modelado unificado. Manual de referencia Addison Wesley
4. El proceso unificado de desarrollo de softwareAddison Wesley
5. Gordon, Carlos. 2008. slideshare. [En línea] Octubre de 2008. <http://www.slideshare.net/carlosgordonb/introduccion-sig>.
6. Araya, Melvin Lizano. 2011. Scribd. [En línea] Julio de 2011. <http://es.scribd.com/doc/51138833/Definicion-SIG>.
7. Ramos, Edgar Loyo. 2000. CIRIA. [En línea] 1 de Septiembre de 2000. [caterina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/...l.../capitulo4.pdf](http://caterina.udlap.mx/u_dl_a/tales/documentos/lis/...l.../capitulo4.pdf).

8. Caldas, Universidad Distrital Francisco José de. 2011. Sistema de Información Geográfica. [En línea] 12 de Julio de 2011. [http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/datos\\_espaciales.htm](http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/datos_espaciales.htm).
9. Geaintec. 2011. GEAINTEC. [En línea] 2011. <http://www.geaintec.cl/servicios/informacion-sig/analisis-modelado/analisis-espacial/>.
10. Caldas, Universidad Distrital Francisco José de. 2010. Sistema de Información Geográfica. [En línea] 2 de Agosto de 2010. <http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/metadatos.htm>.
11. Rica, San Jose - Costa. 2010. Mapealo.com. [En línea] 2010. [www.mapealo.com/Costaricageodigital/.../intro\\_carto.pdf](http://www.mapealo.com/Costaricageodigital/.../intro_carto.pdf) - Costa Rica.
12. Rico, José Lorenzo Herrero. 2000. Dpto. tecnologías de la Información las Comunicaciones. [En línea] Marzo de 2000. [www.tic.udc.es/~nino/blog/documentos/cartografia-digital.pdf](http://www.tic.udc.es/~nino/blog/documentos/cartografia-digital.pdf).
13. Murcia, Universidad de. 2006. SIGMUR SIG y Teledetección en la Universidad de Murcia. [En línea] 2006. [http://www.um.es/geograf/sigmur/sigpdf/temario\\_8.pdf](http://www.um.es/geograf/sigmur/sigpdf/temario_8.pdf).
14. Geomática, Universidad de Melbourne - Departamento de. 2001. autoaprendizaje sig. [En línea] 7 de Marzo de 2001. [http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GIST\\_Raster..](http://www.geogra.uah.es/gisweb/1modulosespanyol/IntroduccionSIG/GISModule/GIST_Raster..)
15. Larenas, Carlos Gonzáles. 2010. MV Blog. [En línea] 29 de Noviembre de 2010. <http://hsutem.puntocl.com/tmv/2010/11/30/rhinoceros-layers/>.
16. savgis.org. 2011. savgis. [En línea] 7 de Abril de 2011. <http://www.savgis.org/download/download.php>.
17. España, Esri. 2010. esri España. [En línea] 2010. <http://www.esri.es/es/productos/arcgis/>.
18. Ambiental, Secretaria de Desarrollo Sustentable y Política. 2011. Scribd. [En línea] 2011. <http://es.scribd.com/doc/57016229/Pinbn-Manual-Cartografia-Sig>.
19. Puebla, Universidad de las Américas. 2011. CIRIA UDLAP. [En línea] Noviembre de 2011. [http://catarina.udlap.mx/u\\_dl\\_a/tales/](http://catarina.udlap.mx/u_dl_a/tales/).
20. Geoinfo. 2010. Softwaregis. [En línea] 2010. <http://www.softwaregis.cl/arcgis.html>.
21. Corel, Club Internacional de Usuarios de. 2011. CORELCLUB.ORG. [En línea] 2011. <http://www.corelclub.org/cartografia-con-coreldraw/>.
22. Fundación, W. Lenguaje Unificado de Modelado. 2011. EcuRed. [En línea] 2011. [http://www.ecured.cu/index.php/Lenguaje\\_de\\_Modelaje\\_Unificado](http://www.ecured.cu/index.php/Lenguaje_de_Modelaje_Unificado).

23. Murcia, Universidad de. 2011. Universidad de Murcia. [En línea] 23 de Mayo de 2011.  
[http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE\\_principales.html](http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html).
24. IBM. 2011. IBM. [En línea] 2011. <http://www-01.ibm.com/software/awdtools/developer/rose/>.
25. EcuRed. 2011. EcuRed. [En línea] 2011. [http://www.ecured.cu/index.php/Visual\\_Paradigm](http://www.ecured.cu/index.php/Visual_Paradigm).
26. Lanzillotta, Analía. 2005. MASTERMAGAZINE. [En línea] 12 de Febrero de 2005.  
<http://www.mastermagazine.info/termino/5560.php>. [
27. Virtual, Pergamino. 2011. Pergaminovirtual. [En línea] 2011.  
<http://www.pergaminovirtual.com/definicion/C.html?PHPSESSID=9a75f4228bde94e4cf0cafade534bc89>.
28. Systems, Zator. 2011. Zator Systems. [En línea] 2011. [http://www.zator.com/Cpp/E1\\_2.htm](http://www.zator.com/Cpp/E1_2.htm).
29. EcuRed. 2011. EcuRed. [En línea] 2011. [http://www.ecured.cu/index.php/IDE\\_de\\_Programaci%C3%B3](http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3).
30. Qt, Comunidad de usuarios de. 2010. Zona Qt. [En línea] 25 de septiembre de 2010.  
<http://www.zonaqt.com/tutoriales/tutorial-b%C3%A1sico-de-qt-4>.
31. IBM. 2011. IBM. [En línea] Agosto de 2011. <http://www-01.ibm.com/software/awdtools/rup/>.
32. EcuRed. 2011. EcuRed. [En línea] 22 de mayo de 2011.  
[http://www.ecured.cu/index.php/Programaci%C3%B3n\\_Extrema\\_%28XP%29](http://www.ecured.cu/index.php/Programaci%C3%B3n_Extrema_%28XP%29).
33. Milagro, Universidad Estatal de. 2010. slideshare. [En línea] 28 de Julio de 2010.  
<http://www.slideshare.net/bebeyom/metodologia-msf-4861508>.
34. Centers, UML Value Added Trainig. 2011. MILESTONE Consulting. [En línea] 23 de Mayo de 2011.  
<http://www.milestone.com.mx/CursoArquitecturaSW.htm>.
35. Alba, Eduardo Rivera. 2010. Scribd. [En línea] 10 de Octubre de 2010.  
<http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
36. Ltd, Sparx Systems Pty. 2007. Enterprise Architect. [En línea] Sparx Systems Pty Ltd, 2007.  
[http://www.sparxsystems.com.ar/resources/tutorial/physical\\_models.html](http://www.sparxsystems.com.ar/resources/tutorial/physical_models.html).