

Universidad de las Ciencias Informáticas

Facultad 6



**Módulo de análisis espacial para el
SIG GeoQ-Guardián 2.0**

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

Autor: Fidel Méndez Hernández

Tutor: Ing. Nilberto Caridad Chavez Marquez

Co-Tutor: Ing. Lewis Rodríguez Fuentes

Junio 2013

"Dicen que los pesimistas ven el vaso medio vacío; los optimistas, en cambio, lo ven medio lleno. Los ingenieros, por supuesto, ven que el vaso es el doble de grande de lo que sería necesario"

Bob

Lewis

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

Fidel Méndez Hernández

Autor

Ing. Nilberto C. Chavez Marquez

Tutor

Ing. Lewis Rodríguez Fuentes

Co-Tutor

DATOS DE CONTACTO

Tutor:



Nombre: Nilberto Caridad Chavez Marquez.

Institución: Universidad de las Ciencias Informáticas.

Título: Ingeniero en Ciencias Informáticas.

Categoría Docente: Profesor Asistente.

Correo electrónico: nchavez@uci.cu

Graduado de la UCI, con cuatro años de experiencia en el Centro de Desarrollo Geoinformática y Señales Digitales y analista principal del proyecto SIG-Desktop en la Universidad de las Ciencias Informáticas.

Co-Tutor:



Nombre: Lewis Rodríguez Fuentes.

Institución: Universidad de las Ciencias Informáticas.

Título: Ingeniero en Ciencias Informáticas.

Correo electrónico: lfuentes@uci.cu

AGRADECIMIENTOS

Un famoso clérigo inglés llamado Charles Caleb Colton, dijo en una ocasión que: "...Ningún metafísico ha experimentado nunca tantas dificultades para expresarse, como el hombre agradecido...". Jamás imaginé que fuera tan complicado para mí: no, agradecer (porque es algo que sale del corazón y da un inmenso placer hacerlo y uno mayor recibirlo), sino por el temor de que en mi calidad de ser humano, cometa el error de olvidar a alguna de las tantas personas que han tenido que ver con que este día haya sido posible.

*Gracias a **mi familia en Cienfuegos**, siempre pendiente de mí, por todos los años juntos, por el amor incondicional. A mis padres por siempre estar a mi lado en todo momento, por darme apoyo y amor. Son todo para mí, por ellos es que me supero cada día, soy mejor persona, los admiro, los amos.*

*Gracias a **mi familia en la UCI**, esa tropa de amigos y compañeros que hemos compartido, coincidido, interpuestos en el camino en diferentes años de la carrera y los que venían ya de antes: en los estudios, en la beca; a todos les estoy agradecido.*

*Gracias a **mi novia** Yaniris, que estuvo siempre a mi lado sin importar cuán difícil fuera el momento y me dió las fuerzas y el amor necesario para salir adelante.*

*Gracias a **mis maestros**, que tan indulgentes y cariñosos se han mostrado conmigo, para quienes han trabajado por cada nuevo conocimiento que he adquirido y que ahora es para mí motivo de satisfacción y de sano orgullo, en especial a mis tutores Nilberto y Lewis.*

*Gracias a **la UCI**, la escuela que me tomó de sus brazos cuando apenas sabía programar y me devuelve ahora a la revolución, convertido en un profesional.*

"Lo que hizo más importante tu rosa, es el tiempo que pasaste con ella..." -dijo el zorro al principito en el libro de Antoine De Saint Exupéry-. Así pudiera decir yo: fue el tiempo que pasé con ustedes, lo que los hizo más importantes. Gracias por todo.

DEDICATORIA

Quiero dedicar este trabajo a todas las personas que han estado a mi lado aun cuando no los necesitaba.

A mi madre.

A mi madre, esa fiel compañera que supo ser mi familia entera aun sabiendo que ya la tenía, que ha sido mi más fiel amiga, mi confidente, mi confesora, por inculcarme quizás a propósito quizás sin querer, el deseo de ser alguien en la vida y de intentar hacer correctamente cada cosa que hiciera.

A mi padre.

A mi padre, por ser también más que un padre, un amigo; por ser aquella persona que siempre tuvo la palabra o la frase perfecta para hacerme entrar en razón, e inculcarme desde pequeño las ansias de aprender. Gracias por decir en algún momento que estudiara mucho.

A mi hermano.

Porque todos los niños tienen un héroe y el mío es mi hermano; siempre he admirado sus logros y he seguido sus pasos

A mis abuelos, Marcelo, Josefa y Luis, y mi tío Roberto, que de seguro están orgullosos de mí, donde quiera que estén.

A mi abuelita Hilda y mi tía Melva por brindarme siempre su apoyo, por quererme mucho y por estar siempre presentes.

En fin, esto va dedicado a todos aquellos que han confiado en mí y a los que no también, pues su actitud me ayudó a demostrarles que sí se puede.

RESUMEN

La Dirección de Seguridad y Protección de la Universidad de las Ciencias Informáticas (UCI) se encuentra enfrascada en la implementación de un sistema de guardia integral, que permita detener los diferentes hechos delictivos que se cometen. Para ello han diseñado un sistema de guardia donde se involucren todas las áreas en aras de hacerlo efectivo. Dada la cantidad de áreas, recursos humanos y materiales con los cuales cuenta la institución, se dificulta el análisis espacial de los activos asociados a los procesos que se llevan a cabo en el sistema de guardia obrera-estudiantil.

El objetivo del trabajo es implementar un módulo de análisis espacial para el SIG GeoQ-Guardián 2.0, que permita analizar espacialmente la información de los procesos de planificación y control de la guardia obrera-estudiantil. Este permitirá calcular la distancia entre las áreas, postas, zonas vulnerables; representar sobre el mapa la coordenada media de una zona dada, entre otras funciones. En la investigación se emplearon los métodos científicos: analítico-sintético, análisis histórico-lógico, la modelación, observación y la entrevista. Se revisaron las tendencias y tecnologías actuales aplicables al proceso.

Como resultados se obtuvieron la documentación técnica del proceso de desarrollo de software y la solución propuesta. Con el módulo, el análisis de la información sobre el mapa es de forma automática, el control de la guardia se realiza más rápido, identificándose aquellas zonas donde hay carencia de personal brindando diversas ventajas en el proceso de toma de decisiones de la guardia obrera-estudiantil.

Palabras clave:

Análisis espacial, control de guardia, guardia obrera-estudiantil, información, módulo, seguridad.

ÍNDICE

INTRODUCCIÓN..... 1

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA SOLUCIÓN PROPUESTA..... 8

1.1 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA 8

Dirección de Seguridad y Protección 8

Guardia 8

Sistema de guardia 8

Servicio 9

Servicio de guardia..... 9

Activo 9

Activo del sistema de guardia 9

Zona 9

Vulnerable 9

Zona vulnerable..... 10

Datos Espaciales..... 10

Mapa..... 10

Mapa Digital 10

Georreferenciación..... 11

Cartografía 11

Sistema..... 11

Sistemas de Información Geográfica (SIG)..... 11

1.2 OBJETO DE ESTUDIO 12

Descripción General 12

Descripción actual del dominio del problema 15

1.3 SOLUCIONES EXISTENTES..... 17

Quantum GIS..... 18

ArcGIS..... 19

GvSig 21

Conclusiones del epígrafe..... 22

1.4 CONCLUSIONES PARCIALES 23

CAPÍTULO 2. SOLUCIÓN TÉCNICA.....	24
2.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE	24
<i>Proceso Unificado de Desarrollo de Software</i>	25
2.2 LENGUAJE DE MODELADO.....	26
<i>Lenguaje Unificado de Modelado (UML)</i>	27
2.3 HERRAMIENTA CASE.....	28
<i>Visual Paradigm</i>	28
2.4 LENGUAJE DE PROGRAMACIÓN	29
<i>Lenguaje de Programación: C++</i>	29
2.5 FRAMEWORK DE DESARROLLO	30
<i>Framework de Desarrollo: Quantum GIS</i>	30
2.6 ENTORNO DE DESARROLLO INTEGRADO (IDE).....	31
<i>Entorno de Desarrollo Integrado: Qt Creator</i>	32
2.7 CONCLUSIONES PARCIALES.....	33
CAPÍTULO 3. CARACTERÍSTICAS DEL SISTEMA.....	34
3.1 MODELO DE DOMINIO	34
<i>Glosario de términos del dominio</i>	35
<i>Descripción del dominio</i>	36
3.2 REQUISITOS FUNCIONALES	36
3.3 REQUISITOS NO FUNCIONALES	37
3.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO	40
<i>Descripción de los Actores</i>	40
<i>Descripción de casos de uso del Sistema</i>	40
<i>Descripción textual de los casos de uso del Sistema</i>	41
3.5 CONCLUSIONES PARCIALES.....	46
CAPÍTULO 4. IMPLEMENTACIÓN Y VALIDACIÓN DE LOS RESULTADOS.....	47
4.1 PATRÓN ARQUITECTÓNICO	47
<i>Modelo Vista Controlador (MVC)</i>	48
4.2 PATRONES DE DISEÑO	49
<i>Patrones GRASP</i>	50

<i>Patrones GoF (Gang of Four)</i>	51
4.3 MODELO DE DISEÑO.....	51
<i>Diagrama de Clases del Diseño</i>	51
4.4 GENERALIDADES DE LA IMPLEMENTACIÓN.....	53
<i>Modelo de Implementación</i>	53
<i>Modelo de Despliegue</i>	54
4.5 PRUEBAS.....	54
<i>Pruebas de Caja Negra</i>	55
4.6 CONCLUSIONES PARCIALES.....	60
CONCLUSIONES	61
RECOMENDACIONES	62
BIBLIOGRAFÍA	63
GLOSARIO	66

ÍNDICE DE FIGURAS

<i>Figura 1: Función Optimización de análisis espacial</i>	14
<i>Figura 2: Resultados de la utilización de análisis espacial</i>	15
<i>Figura 3: Interfaz de Quantum Gis</i>	18
<i>Figura 4: Interfaz de ArcGIS</i>	20
<i>Figura 5: Interfaz de GvSig</i>	21
<i>Figura 6: Fases y flujos que utiliza RUP</i>	26
<i>Figura 7: Modelo de Dominio</i>	35
<i>Figura 8: Diagrama de Casos de Uso del Sistema.</i>	41
<i>Figura 9: Interrelación entre los elementos del patrón MVC</i>	49
<i>Figura 10: Diagrama de Clases del Diseño CU_ Calcular Distancia entre Puntos</i>	52
<i>Figura 11: Diagrama de Componentes CU_ Calcular Distancia entre dos Puntos</i>	53
<i>Figura 12: Modelo de Despliegue</i>	54

ÍNDICE DE TABLAS

<i>Tabla 1: Descripción del Actor.....</i>	<i>40</i>
<i>Tabla 2: Descripción del CU Calcular la distancia entre dos puntos.</i>	<i>45</i>
<i>Tabla 3: Secciones a probar en Calcular Matriz Distancia.....</i>	<i>58</i>
<i>Tabla 4: Descripción de las variables en Calcular Matriz Distancia.....</i>	<i>59</i>
<i>Tabla 5: Matriz de datos Calcular Matriz Distancia</i>	<i>60</i>

INTRODUCCIÓN

El hombre siempre ha tenido la necesidad de transmitir sus experiencias. Desde el año 2300 antes de Cristo, los mapas se elaboraron sobre pergaminos y fueron escritos utilizando plumas de aves. Con la invención de la brújula, la humanidad pudo desarrollar mapas de diferentes escalas y con mayor grado de exactitud, posteriormente el surgimiento de la imprenta, el cuadrante¹ y el nonio² posibilitó el incremento de la producción de mapas (Martín , 2010). El telescopio y otros dispositivos que permiten hacer un estudio más exacto de la tierra; brindaron a las personas encargadas de elaborar los mapas la posibilidad de encontrar la latitud midiendo ángulos a partir de la estrella polar durante las noches y por el día con la ayuda del Sol. Sobre los mapas se puede representar distintos tipos de información, que van desde el orden social y económico, hasta los fenómenos naturales que ocurren en la Tierra.

La cartografía es la ciencia que se encarga del estudio y elaboración de los mapas geográficos, ha abarcado desde los primeros trazos realizados en la arena, hasta el uso de novedosas técnicas geodésicas³, fotogramétricas⁴, de teledetección⁵ y servicios de mapas en Internet (Cali, 2011).

A medida que se fue desarrollando la cartografía, los mapas se hicieron más complejos a causa de la gran cantidad de símbolos y datos que encerraban en sus formatos, convirtiéndose en un obstáculo para aquellos que no eran expertos en su manipulación. Creando la necesidad de utilizar herramientas capaces de agilizar los procesos en los cuales se maneja información geográfica. Las Tecnologías de la Información y las Comunicaciones (TIC) han facilitado el manejo de este tipo de información, denominándose como el “conjunto de procesos y productos derivados de las nuevas herramientas (hardware y software), soportes de la información y canales de comunicación relacionados con el almacenamiento, procesamiento y transmisión digitalizados de la información” (GONZÁLEZ, 2006).

¹**Cuadrante:** Antiguo instrumento utilizado para medir ángulos en astronomía y navegación.

²**Nonio:** Segunda escala auxiliar que tienen algunos instrumentos de medición.

³**Técnicas Geodésicas:** Mediciones técnicas para unir puntos de referencia de diferentes continentes.

⁴**Técnicas Fotogramétricas:** Con el uso de imágenes fotográficas se determinan las propiedades geométricas de los objetos.

⁵**Técnicas Teledetección:** Recoger información a través de diferentes dispositivos de un objeto concreto o un área.

Las TIC son técnicas de gestión e innovación que se basan en sistemas o productos que son capaces de captar información multidimensional, almacenarla, elaborarla, transmitirlos, difundirlas y hacerlas inteligibles, accesibles y aplicables en correspondencia con el fenómeno a transformar. Su singularidad radica en la constante innovación que posibilitan y su mayor capacidad de tratamiento de la información.

Con el desarrollo de las TIC, se ha posibilitado responder satisfactoriamente a las necesidades planteadas por la geografía, donde la informática ha ayudado a una mayor comprensión y valoración de la misma. Con el surgimiento de la Internet y las comunicaciones, se ha potencializado el trabajo con los mapas, dando origen a los Sistemas de Información Geográfica (SIG). Los SIG surgen a partir de la observación que se realiza de las diferentes coberturas sobre la superficie de la tierra concluyendo que no eran independientes entre sí, sino que guardaban algún tipo de relación, haciéndose latente la necesidad de evaluarlos de una forma integrada y multidisciplinaria. Una manera sencilla de hacerlo era superponiendo copias transparentes de mapas de coberturas sobre mesas iluminadas y encontrar puntos de coincidencia en los distintos mapas de los diferentes datos descriptivos (Hernández, 2012).

Según David Rhind, “un SIG es un sistema de hardware, software y procedimientos, diseñados para soportar la captura, manejo, manipulación, análisis, modelado y despliegue de datos espacialmente referenciados (geo-referenciados), para la solución de problemas complejos del manejo y planeamiento territorial” (Rhind, 2005).

En las últimas décadas, los SIG evolucionaron, alcanzando un impacto significativo en distintas esferas de la vida social; con su aplicación, las instituciones administrativas y centros de investigación, los han convertido en una herramienta imprescindible para solucionar conflictos, utilizando información proveniente de un rango amplio de disciplinas.

Los SIG pueden dar soluciones a disímiles problemas; ejemplo de ello es en un mapa de peligro sísmico, donde se logra indicar la ubicación y extensión de áreas, para evitar fuertes inversiones de capital y deliberar sólo las actividades apropiadas donde incurran terremotos, tsunamis o volcanes. En el transporte, se pueden utilizar para obtener los poblados que estén siendo interceptados por alguna vía principal y así considerar estas áreas para determinadas situaciones. También se utilizan estos sistemas para obtener ciudades que se encuentren a una distancia determinada, ya sea de tiendas, empresas, hospitales o lugares específicos de interés; al igual que para conocer si alguna entidad se encuentra

completamente dentro de otra, pueden municipios y países que estén completamente dentro de países y continentes respectivamente (Hernández, 2012).

En Cuba, numerosas empresas e instituciones hacen uso de los SIG en busca de lograr más eficiencia, eficacia y aumento de la calidad de vida de la población, se pueden mencionar como ejemplos: el Instituto de Geografía de la Academia de Ciencias de Cuba "El SIG de Cuba" que tiene como propósito actualizar el Atlas Nacional de Cuba; el Instituto Cubano de Hidrología desarrolla TeleMap, una herramienta destinada al diseño de SIG y el Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA).

La Universidad de las Ciencias Informáticas (UCI) como parte del programa de informatización de la sociedad cubana incluyó desde el curso 2006–2007 el estudio y desarrollo de los SIG, pretendiendo poner en el mercado soluciones software nacionales que garanticen la representación de datos espaciales en los diferentes escenarios que puedan presentarse en el ámbito nacional y para la exportación. El centro de desarrollo Geoinformática y Señales Digitales (GEySED) perteneciente a la Facultad 6, es el que agrupa y desarrolla la línea de investigación y desarrollo de los SIG donde se ha creado un proyecto de investigación y desarrollo (I+D) llamado SIG-DESKTOP, que tiene como objetivo el desarrollo de SIG de escritorio para diferentes esferas según las necesidades del país.

En la UCI existe hoy un número elevado de recursos que permiten a los estudiantes y trabajadores satisfacer sus necesidades, permitiendo una mejor convivencia en la institución. Desde su surgimiento se hizo imprescindible velar por la seguridad de estos recursos por lo que se conformó un sistema de guardia capaz de cubrir las áreas de la residencia estudiantil, el área docente y el perímetro que rodea el recinto universitario, con el objetivo de prevenir, detectar y neutralizar la ocurrencia de amenazas, así como reducir los niveles de riesgos que puedan originar hechos delictivos de carácter común o que atenten contra los principios de la nación.

Para este sistema de guardia se hizo apremiante contar con el apoyo de los estudiantes y trabajadores de la universidad quienes realizan la guardia en puntos estratégicos, planificada por un personal designado por los jefes de áreas a los que pertenecen.

La información asociada a este proceso se controla de forma manual mediante documentos impresos. Además, no se encuentra en una fuente centralizada a la que el personal encargado tenga acceso, haciendo difícil el control físico y la obtención de reportes con los resultados del proceso.

La Dirección de Seguridad y Protección (DSP) tiene entre sus funciones la planificación y control de la guardia obrera-estudiantil en dicha institución. La misma no cuenta con un mecanismo que le permita realizar un análisis en la distribución de los puntos estratégicos para la protección de los recursos, presentando limitantes como son:

La ubicación y el conocimiento de cuántas postas, puestos de mandos y zonas vulnerables están ubicados en un área determinada. El análisis que se realiza actualmente, muchas veces trae consigo errores que conllevan a imprecisiones en la toma de decisiones, fundamentalmente en la ubicación estratégica de las postas en cada área que existe en la universidad. Debido a esto, ubican al personal implicado en el servicio de guardia en postas que no son efectivas, ya que se encuentran unas en frente de otras e inciden en el mismo radio de acción.

El cálculo de la distancia existente entre dos puestos de mando, entre un puesto de mando y una zona vulnerable, y entre varias postas. La ubicación de los puestos de mando más cercanos donde se pueda acudir en el momento en el que se presente un incidente durante el servicio de guardia, se convierte en un proceso complejo. El análisis se realiza de manera manual sobre mapas impresos generados por aplicaciones.

El análisis correcto y oportuno sobre aquellas zonas consideradas vulnerables en el sistema de guardia, dificulta la correcta distribución de los recursos humanos asociados al servicio de guardia. Así como el poco análisis y control que se efectúa de los turnos de guardia dificulta la planificación debido a que puede ocurrir que el personal implicado en el servicio de guardia, haya realizado seguidamente un mismo turno de guardia, lo que trae consigo inconformidades y una reestructuración de la planificación de la misma. En muchas ocasiones no se tiene en cuenta la cantidad de personas necesarias en una posta determinada, siendo prioridad las zonas consideradas de alto riesgos.

La DSP, tras el análisis realizado, no cuenta con un mecanismo de análisis espacial en el desarrollo de la planificación y control de los activos del sistema de guardia en la UCI, lo que dificulta el proceso de toma de decisiones.

Tras analizar la problemática antes descrita se identifica como **problema a resolver** de esta investigación: ¿Cómo favorecer la distribución y el control de la guardia obrera estudiantil, para agilizar el proceso de

toma de decisiones en la Dirección de Seguridad y Protección en la Universidad de las Ciencias Informáticas?

Se define como **objetivo general** del presente trabajo de diploma: Desarrollar el módulo de análisis espacial para el Sistema de Información Geográfica GeoQ-Guardián 2.0.

Defendiéndose como **objeto de estudio** el proceso de análisis espacial en los Sistemas de Información Geográfica, específicamente en la automatización del proceso de análisis espacial en el Sistema de Información Geográfica GeoQ-Guardián 2.0 lo que constituye el **campo de acción** de la investigación.

Para dar cumplimiento al objetivo planteado se trazan las siguientes **tareas de la investigación**:

- Caracterizar los procesos del sistema de guardia obrera estudiantil.
- Realizar una revisión de las tendencias y tecnologías actuales aplicables.
- Modelar los artefactos que genera la solución propuesta.
- Implementar los requisitos funcionales de la solución.
- Probar el correcto funcionamiento de la solución.

A lo largo de esta investigación se tuvo en cuenta los **métodos científicos** que se utilizaron en la realización de la misma, a continuación se presentan:

Teóricos:

- **Histórico–Lógico:** Se utilizaron con el objetivo de realizar un estudio sobre la evolución y el desarrollo que han tenido los SIG, lo que permitió desarrollar una valoración sobre en qué etapa se encuentran dichos sistemas y cuál es la tecnología más factible a utilizar para el desarrollo de los mismos. Examinar los procesos de análisis espacial que se utilizaron para asegurar la protección de los recursos humanos y materiales.
- **Modelación:** La modelación es justamente el proceso mediante el cual se crean abstracciones con vista a explicar la realidad, por lo que es fundamental en el proceso de construcción del software. Se utilizará para realizar los modelos ingenieriles de la solución que se plantea.
- **Analítico-Sintético:** Se utilizó para sintetizar y analizar la documentación y bibliografías de diferentes autores, para sustentar desde el punto de vista teórico y práctico los elementos que se relacionan con el proceso de análisis espacial del sistema de guardia obrera-estudiantil.

Empíricos:

- **Entrevista:** Permite establecer el alcance que tuvo el presente trabajo, el objetivo, además del flujo actual de la información, así como la comunicación con otras personas, para obtener los requerimientos a implementar en la solución propuesta.

Se realizó una entrevista informativa y no estructurada centralizada en el tema de análisis espacial. La población seleccionada fueron especialistas del DSP, tomándose como muestra seis personas y como unidad de estudio al especialista general de dicho departamento. La técnica utilizada para el trabajo con la muestra seleccionada fue muestreo no probabilístico, específicamente el muestro intencional garantizando que la selección sea más lógica y racional.

Como parte de esta investigación se propone la siguiente **idea a defender**:

Si se desarrolla el módulo de análisis espacial para el Sistema de Información Geográfica GeoQ-Guardián 2.0, se agilizará el proceso de toma de decisiones de la planificación de la guardia obrera-estudiantil en la Universidad de las Ciencias Informáticas.

Con la culminación de todas las tareas desarrolladas y del proceso investigativo se espera obtener los siguientes resultados:

- Documentación técnica del proceso de desarrollo de la solución propuesta.
- Módulo de análisis espacial del SIG GeoQ-Guardián 2.0.

El presente trabajo está dividido en 4 capítulos:

Capítulo 1: Fundamentación teórica de la solución propuesta

En este capítulo se describen brevemente los conceptos fundamentales relacionados con el dominio del problema. Se realiza un estudio del arte de los SIG y de los sistemas existentes para el análisis espacial de la planificación de la guardia, además de hacer un análisis de las soluciones existentes.

Capítulo 2: Solución técnica

En este capítulo se seleccionan las tecnologías y herramientas a utilizar en el proceso de desarrollo del software. Se abordan las características presentes en el lenguaje de programación, lenguaje modelado, la metodología, el entorno de desarrollo y la herramienta case utilizadas en la elaboración de la solución.

Capítulo 3: Características del sistema

En este capítulo se describe las características que va a constar el sistema, planteándose el modelo del dominio, los requerimientos funcionales y no funcionales, así como la descripción de los casos de uso y los actores del sistema.

Capítulo 4: Implementación y validación de los resultados

En este capítulo se describen los patrones de diseños, los diagramas de clases del análisis y los diagramas de clases del diseño, la arquitectura a utilizar, así como los diagramas de componentes, el diagrama de despliegue y el diseño de casos de prueba de la solución.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA SOLUCIÓN PROPUESTA

En el presente capítulo, para una mejor comprensión del problema, se introducen una serie de conceptos básicos para el trabajo con SIG, tales como información geográfica, mapas y datos espaciales. Se hace una reseña de los SIG con funciones de análisis espacial más utilizadas a nivel internacional, haciendo énfasis en sus características, lo que permitirá seleccionar la que se ajuste más a las necesidades de trabajo.

1.1 Conceptos asociados al dominio del problema

Los conceptos asociados al dominio del problema se toman a partir de las definiciones más actualizadas; estos conceptos son útiles para el entendimiento del problema y ayudan a entender las relaciones que existen entre ellos.

Dirección de Seguridad y Protección

Organismo encargado de asesorar, dirigir y controlar el funcionamiento del sistema de guardia en una determinada institución y de la capacitación del personal que trabaja en lugares o áreas expuestas a riesgos, manteniendo informada a la institución de la situación existente y las medidas derivadas de las inspecciones y controles realizados por las inspecciones de dicho cuerpo (Castro, 1998).

Guardia

Conjunto de soldados o gente armada que asegura la defensa de una persona o de un puesto, encomendado a una o más personas (RAE, 2012).

Sistema de guardia

Es autor de la presente investigación considera como un sistema de guardia un conjunto de activos interrelacionados (áreas, puestos de mandos, postas, zonas vulnerables, agrupaciones, recursos humanos y la información asociada a la planificación de la guardia) que permiten disminuir los riesgos ante hechos delictivos.

Servicio

Organización y personal destinados a cuidar intereses o satisfacer necesidades del público o de alguna entidad oficial o privada (RAE, 2012).

Servicio de guardia

El autor de la presente investigación considera como un servicio de guardia, el apoyo voluntario que brinda un determinado recurso humano en función de la seguridad y protección de una institución o recurso humano asociado en un momento dado. El mismo se realiza diurno, nocturno o en ambos horarios en dependencia de las necesidades existentes, por lo que se realiza una previa planificación para su correcto funcionamiento.

Activo

Conjunto de todos los bienes y derechos con valor monetario que son propiedad de una empresa, institución o individuo, y que se reflejan en su contabilidad (RAE, 2012).

Activo del sistema de guardia

No es más que todo lo que posee una determinada institución en términos de áreas, puestos de mando, postas, recursos humanos y zonas vulnerables, que son zonas de alto riesgo para la ocurrencia de hechos delictivos. Cada una de las instituciones por si solas presentan zonas que son más vulnerables que otras, por lo que hace que la cantidad de recursos humanos varíe en determinados casos.

Zona

Extensión considerable de terreno que tiene forma de banda o franja (RAE, 2012).

Vulnerable

Se aplica a la persona, al carácter o al organismo que es débil o que puede ser dañado o afectado fácilmente porque no sabe o no puede defenderse (RAE, 2012).

Zona vulnerable

El autor de la presente investigación considera como zona vulnerable un lugar, espacio o sitio que aparece expuesta a eventos naturales o causados por el hombre (hechos delictivos que puedan ocurrir), que puede ser afectado de forma directa o indirecta, pudiendo ocasionarle un daño físico o un perjuicio económico.

Datos Espaciales

Los datos espaciales o datos cartográficos digitales son informaciones sobre la localización y las formas de un objeto geográfico y las relaciones entre ellos, normalmente con coordenadas y topología, es el tipo de datos necesarios para crear mapas y estudiar relaciones espaciales, son el componente fundamental de los SIG. Se refieren a entidades o fenómenos que cumplen los siguientes principios básicos:

- Posición absoluta: Sobre un sistema de coordenadas (x, y, z).
- Posición relativa: Frente a otros elementos del paisaje (topología).
- Figura geométrica que representan (punto, línea, polígono).
- Atributos que describen (características del elemento o fenómeno).

“Es evidente 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” (Lanero, 2011).

Mapa

Un mapa es la representación gráfica a escala de la Tierra o parte de ella en una superficie. Es un conjunto de puntos, líneas y áreas, que están definidos tanto por su colocación en el espacio con respecto a un sistema de coordenadas, como por sus atributos no espaciales. La leyenda del mapa es la clave que enlaza los atributos no espaciales con las entidades espaciales (Capdevila, 2002).

Mapa Digital

Un mapa digital es un conjunto de datos que representan información espacial y atributos, almacenados en el ordenador. Constituye el almacenamiento de información espacial como dibujos electrónicos hechos a base de elementos gráficos sencillos (líneas, puntos, círculos organizados en capas, con el objetivo de

una salida impresa o por pantalla (Arteaga, 2004).

Georreferenciación

La georreferenciación es la asignación de coordenadas terrestres a objetos de interés, ya sean naturales o artificiales. Permite definir la ubicación de un objeto, de manera espacial, mediante el registro de su longitud y latitud en un sistema de coordenadas específico.

Cartografía

Se conoce como cartografía a la ciencia que se dedica al estudio y elaboración de mapas que sirven para la navegación, para la ubicación del ser humano. La palabra cartografía viene del griego *grapho* y significa 'la escritura de mapas' (Alonso, 2007).

Es una herramienta que mediante elementos gráficos ordenados en una secuencia lógica aporta información. Está conformada por varios elementos como la escala, cuadrícula, leyenda, líneas de marco y sistemas de numeración de hojas. Consiste en la representación, lo más exacta posible, de parte o toda la superficie de la tierra y cualquier cuerpo sobre una superficie plana.

Sistema

Es un todo integrado, aunque compuesto de estructuras diversas, interactuantes y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos pueden variar ampliamente de un sistema a otro. Un sistema ejecuta una función imposible de realizar por cualquiera de las partes individuales. La complejidad de la combinación está implícita (Bunge, 2009).

Es una colección organizada de hombres, máquinas y métodos necesaria para cumplir un objetivo específico. Un módulo ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí.

Sistemas de Información Geográfica (SIG)

Muchos autores han definido en la última década lo que es un SIG, entre las definiciones dadas se encuentran las siguientes:

- Los SIG se definen como el conjunto de herramientas para reunir, introducir, almacenar, recuperar, transformar y cartografiar datos espaciales sobre el mundo real para un conjunto particular de objetivos (Sendra, 2002).
- Los SIG son el conjunto de instrumentos y métodos especialmente dispuestos para capturar, almacenar, transformar y presentar información geográfica o territorial referenciada al mundo real (Buzai, 2011).
- SIG es el conjunto de mapas de la misma porción del territorio, donde un lugar concreto tiene la misma localización en todos los mapas incluidos en el sistema de información, resultando posible realizar análisis de sus características espaciales y temáticas para obtener un mejor conocimiento de esa zona (Jiménez, 2008)
- Los SIG son un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos. Son una tecnología que permiten gestionar y analizar la información espacial, que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato (Thompson, 2010).

1.2 Objeto de Estudio

Descripción General

Comprender la distribución espacial de los datos de los fenómenos que ocurren en el espacio constituye hoy un gran reto para el esclarecimiento de cuestiones centrales en diversas disciplinas. Con el uso y aplicación de las Tecnologías de Información Geográficas (TIG⁶) se establece un marco de intercambio que permite visualizar los distintos procesos que ocurren en el espacio, en un tiempo determinado, los cuales permiten la visualización espacial de variables tales como las poblaciones individuales, la calidad de vida o los índices de ventas en una región, mediante el uso de mapas. Es por esto que la percepción

⁶ **TIG:** Están formadas por un conjunto de técnicas y métodos clásicos y modernos en torno a la Cartografía (lectura, interpretación y elaboración de mapas).

visual de la distribución espacial del fenómeno, es muy útil para traducir los patrones existentes en consideraciones objetivas y medibles.

El análisis espacial incluye un conjunto de herramientas que amplían las capacidades del análisis estadístico tradicional para abordar aquellos casos en los que la distribución espacial de los datos tiene influencia sobre las variables medidas y esta se considera relevante. La georreferenciación de los datos permite manejar un conjunto de conceptos nuevos como son los de distancia (entre dos puntos), adyacencia (entre dos polígonos o dos líneas), interacción y vecindad (entre puntos) (Rogerson, 2009).

Algunas interrogantes que busca resolver el uso del análisis espacial son:

- ¿Cómo realizar la distribución espacial de fenómenos?
- ¿Cómo incluir patrones espaciales en un mapa?
- ¿Cómo llevar a cabo asociaciones y concentración de elementos geográficos?
- ¿Cómo estimar o predecir los fenómenos geográficos?
- ¿Cómo seleccionar las variables geográficas?

El análisis de datos con SIG tiene por finalidad descubrir estructuras espaciales, asociaciones y relaciones entre los datos, así como modelar fenómenos geográficos. El proceso convierte los datos en información útil para conocer un problema determinado.

Los SIG cuentan con funciones de análisis espacial (ver Figura 1) para actuar sobre los datos desde la captura y edición hasta la visualización, identificándose como (Rogerson, 2009):

- **Interrogaciones:** Se pueden utilizar criterios temáticos o espaciales y combinar ambos; las preguntas pueden ser simples y bien definidas (¿Qué hay en tal localización? ¿Cuántos objetos de un tipo se encuentran dentro a menos de 1 km de un punto determinado?) o más vagas (¿cuál es la localidad más cercana a un punto?).
- **Medidas:** Estas funciones producen valores numéricos que describen algunas propiedades esenciales de los objetos, como su longitud, área, forma, pendiente, orientación o la distancia y dirección entre dos o más entidades.
- **Transformaciones:** Comprenden un conjunto de métodos simples de análisis espacial que transforman las entidades originales, mediante comparaciones o combinaciones. Estas funciones utilizan principios y reglas geométricas, aritméticas o lógicas, y también operadores de conversión

de datos vectoriales a ráster y viceversa. Este grupo incluye la creación de corredores (buffer), las operaciones de “punto en polígono”, superposición de polígonos e interpolación espacial.

- **Sumarios:** Son funciones que resumen los datos en unos cuantos indicadores. Los más usados son los de estadística descriptiva (media aritmética, mediana, moda, desviación típica, varianza) y sus equivalentes espaciales, como el centro de gravedad y la desviación típica de las distancias.
- **Optimización:** Se agrupan aquí diversas técnicas de naturaleza normativa cuya finalidad es la de seleccionar localizaciones que cumplen determinados criterios. Son muy utilizadas en los estudios de mercado y en la planificación de los equipamientos públicos, para estimar la localización óptima de establecimientos con arreglo a determinados supuestos. También se incluyen los métodos de análisis de redes, como la búsqueda de rutas óptimas entre dos o más localizaciones en una red.

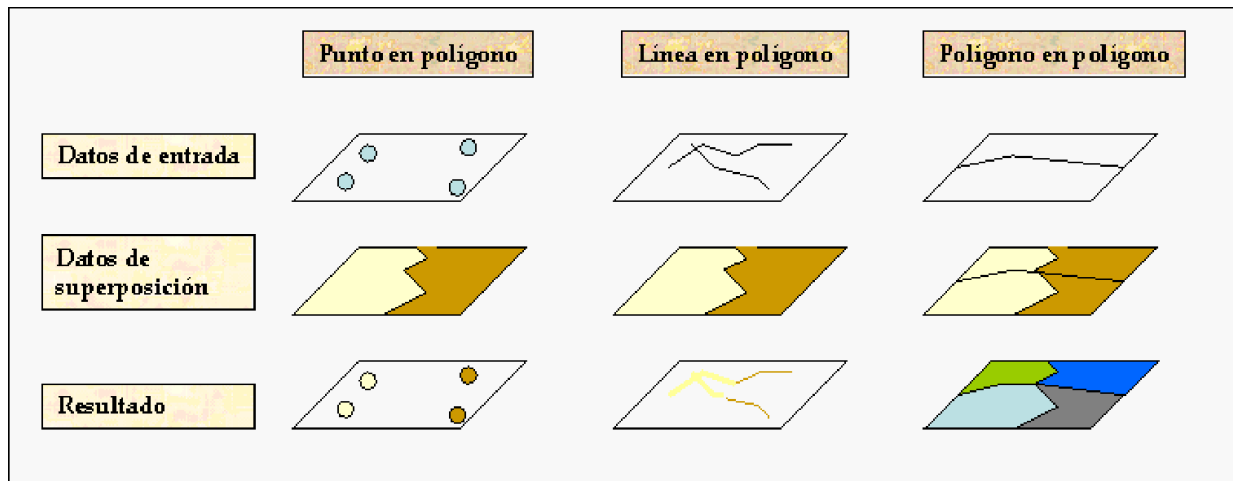


Figura 1: Función Optimización de análisis espacial (Longley, 2001)

El proceso de análisis espacial cuenta con varias etapas para su realización como son (Rogerson, 2009):

1. **Definición de objetivos:** Especificación clara de las preguntas e hipótesis que se van a investigar.
2. **Selección de los datos:** El tipo de elemento (punto, línea, polígono) y su representación (vectorial, ráster) así como la escala de medida de los datos condicionan los métodos y transformaciones de los mismos.

3. **Determinación de los métodos de análisis:** En función de los objetivos y de los datos disponibles es posible que existan varias alternativas metodológicas entre las que se deberá decidir.
4. **Aplicación de las funciones necesarias en un programa SIG:** Los modos de aplicación y las funciones de análisis varían en cada programa; sin embargo, es necesario conocer los conceptos y transformaciones que se aplican a los datos.
5. **Presentación y valoración de los resultados:** Los resultados se pueden comunicar en forma de mapas, tablas, gráficos u otros documentos, cuya utilidad deberá ser valorada (ver Figura 2).

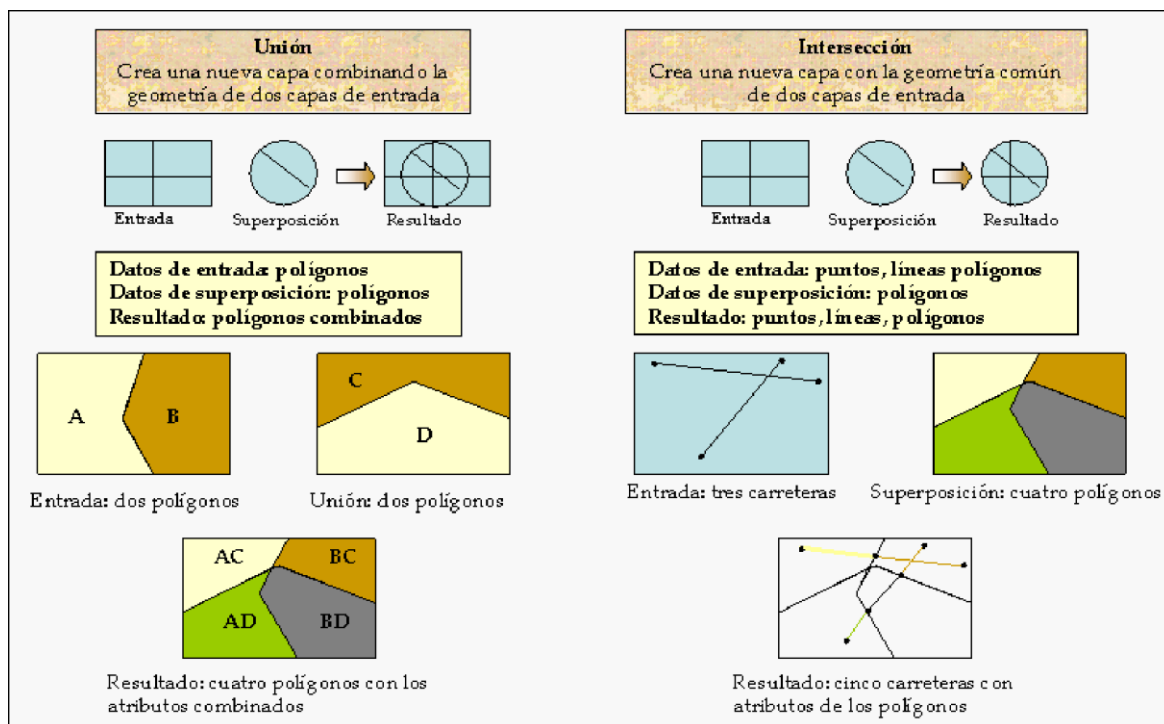


Figura 2: Resultados de la utilización de análisis espacial (Longley, 2001)

Descripción actual del dominio del problema

El proceso de planificación de la guardia obrera de forma manual es complejo, esto se debe a la gran cantidad de información que manejan sus planificadores, el número de reportes que deben ser generados y la dificultad que surge para la realización de una buena planificación.

Para el desarrollo de este proceso fueron definidos cuatro tipos de guardia (guardia obrera de trabajadores internos, guardia obrera de trabajadores externos, guardia obrera de vicerrectoría de logística y guardia obrera de oficiales operativos). La guardia de los trabajadores se cumple fuera del horario laboral y se planifica por el área donde el trabajador está contratado y comprometido. Las posiciones se ubican por decisión del director de seguridad y protección y en todas las áreas debe designarse un responsable que atienda la organización y realización de la guardia, que se identifique y trabaje con el puesto de mando de la guardia de la UCI.

Las personas de guardia para cada día del mes se planifican antes del día veinte y dicha planificación se entrega al puesto de mando de la guardia obrera central entre el veinte al veinticinco de cada mes. Entre los días veinte al veinticinco se realiza la reunión de la guardia UCI con la participación de las personas que atienden la tarea en cada área. El especialista de seguridad y protección entre el veinticinco y treinta y uno de cada mes, elabora los listados con las personas de guardia para cada día y se los entrega al organizador de la guardia obrera de seguridad y protección que entra de guardia.

La guardia obrera de trabajadores internos cuenta con diecisiete posiciones en la residencia de profesores y se organiza mediante dos turnos, teniendo presente la condición de internos. Esta guardia es planificada por el subdirector de residencia de profesores en coordinación con los dirigentes de las áreas donde laboran los internos.

Por su parte la guardia obrera de trabajadores externos tiene veinte posiciones en los docentes, IP, casona, piscina y gimnasio; se realiza en tres turnos particularizada por sexos. Esta es planificada por las personas responsables de la guardia en las áreas, en dos variantes (días fijo o consecutivos) y con la cantidad de veinte compañeras y cuarenta hombres diarios. Cada una de estas áreas tiene al mes días asignados para la guardia obrera de trabajadores externos.

Para la guardia obrera de vicerrectoría de Logística se posee trece posiciones en los complejos cocina-comedores y se efectúa en dos turnos. La planificación de esta guardia es realizada por cada uno de los directores de los complejos.

La guardia obrera de oficiales operativos es realizada por los cuadros y dirigentes que no hacen de rector en funciones. Cuenta para su desarrollo con dos trabajadores diarios, en el horario de 7 PM a 7 AM y es planificada por los dirigentes de las áreas.

Una vez planificada la guardia obrera las personas son responsables directas de su cumplimiento. Para ello deben conocer, con quince días de anticipación como mínimo, el día que le corresponde.

El puesto de mando controla la asistencia y realización de la guardia obrera, siendo responsable de comunicar semanalmente las incidencias a los dirigentes, factores y organizaciones de las áreas, con copia a los directores de seguridad y protección y de capital humano. Todo este proceso manual de planificación y control de la guardia obrera hace que resulte más compleja la generación de los reportes. Los posibles cambios organizacionales de la guardia exigen además, una constante actualización en cada una de las planificaciones, lo que implica volver a imprimir cada reporte.

La DSP no cuenta con un mecanismo que le permita analizar los activos del sistema de guardia de la UCI, pues los reportes y controles se efectúan de forma manual resultando muy compleja la realización de dicho procedimiento. Además posee debilidades presentes en la planificación de la guardia obrera como son:

- Conocer la distancia mínima y máxima que existe entre dos puntos de un área.
- Identificar el puesto de mando más cercano a un hecho que se produzca.
- Conocer las postas que cubren áreas comunes.
- Conocer cuántas postas, zonas vulnerables y puestos de mandos están ubicados en un área determinada.
- Conocer si los recorridos planificados se interceptan en algún punto específico.

1.3 Soluciones existentes

En la actualidad existen muchas soluciones de SIG que usan análisis espacial dentro de su proceso de investigación y exploración de los datos geográficos, algunos ejemplos son:

Quantum GIS



Figura 3: Interfaz de Guantum Gis (Bosch, 2011)

Quantum GIS (QGIS) es un SIG de código libre, con grandes potencialidades para la edición de mapas, para plataformas GNU/Linux, Unix, Mac OS y Microsoft Windows. Permite manejar formatos ráster y vectoriales, así como bases de datos (Dawson, 2007).

Características:

QGIS tiene muchas funciones y características comunes a todos los SIG. Las características son divididas en elementos del núcleo y complementos.

Elementos del núcleo

- Soporte ráster y vectorial.
- Soporte para PostgreSQL⁷ con tablas espaciales utilizando PostGIS⁸.
- Diseño de mapas.
- Panel de vista general.

⁷**PostgreSQL:** Un Sistema Gestor de Bases de Datos relacional orientado a objetos y libre.

⁸**PostGIS:** Módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL.

- Guardar y recuperar proyectos.
- Cambiar simbología vectorial y ráster.
- Arquitectura extensible con complementos.

Complementos de análisis espacial:

- Añadir capas WFS⁹ y WMS¹⁰.
- Añadir capas de texto delimitado.
- Decoración (etiqueta de copyright, flecha de Norte y barra de escala)
- Georreferenciación.
- Herramientas GPS¹¹.
- Integración con GRASS GIS¹² y uso de las herramientas disponibles.
- Uso de R (programa de estadísticas avanzado).
- Consola para la programación de Python¹³.
- Acceso a GPS en formato gpx.
- Acceso a OpenstreetMap¹⁴.
- Creación de datos para Mapserver.¹⁵

ArcGIS

El ArcGIS Desktop es un conjunto de aplicaciones integradas: ArcMap¹⁶, ArcCatalog¹⁷ y ArcToolbox¹⁸, los cuales posibilitan el análisis espacial, la administración de datos y el mapeo. Este fue desarrollado y comercializado por ESRI, el mismo se agrupa en varias aplicaciones para la captura, edición, manejo,

⁹**WFS:** Permite interactuar con los mapas servidos por el estándar WMS.

¹⁰**WMS:** Produce mapas de datos referenciados espacialmente.

¹¹**GPS:** Es un sistema que sirve para determinar nuestra posición con coordenadas de Latitud, Longitud y Altura.

¹²**GRASS GIS:** Es un software SIG licencia GPL (software libre).

¹³**Python:** Es un lenguaje de programación de alto nivel.

¹⁴**OpenstreetMap:** Es un proyecto colaborativo para crear mapas libres y editables

¹⁵**Mapserver:** Es un entorno de desarrollo en código abierto (Open Source Initiative) para la creación de aplicaciones SIG en Internet/Intranet.

¹⁶**ArcMap:** Es un software de Sistema de Información Geográfico (SIG) creado por ESRI para mapeo digital.

¹⁷**ArcCatalog:** Es un software de Sistema de Información Geográfico (SIG) creado por ESRI diseñado para explorar y administrar los datos.

¹⁸**ArcToolbox:** Es un software de Sistema de Información Geográfico (SIG) creado por ESRI para usar en el geoprocesamiento de datos.

análisis, tratamiento, diseño, publicación e impresión de información geográfica. El análisis espacial es la funcionalidad más poderosa del “software” ArcGIS (Dangermond, 2008).

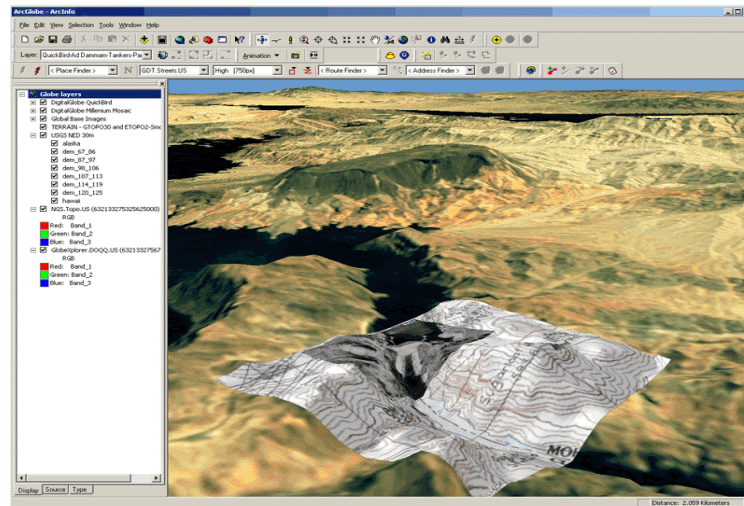


Figura 4: Interfaz de ArcGIS (Dangermond, 2008)

Funciones de análisis espacial de ArcGIS:

- **Clip:** Permite cortar una capa de información según el perímetro definido por el contorno de otra capa. El resultado se almacena en un nuevo archivo. El archivo de entrada puede ser de puntos, líneas o polígonos. El archivo utilizado para cortar, debe ser de polígonos.
- **Interceptar:** Permite obtener un archivo nuevo a partir de dos que se solapan. El nuevo fichero sintetizará los datos espaciales del área interceptada; en su tabla de atributos almacenará los campos de ambos archivos correspondientes a cada área de intersección. En el caso de la intersección de dos archivos de polígonos, los polígonos del archivo de entrada serán recortados según los límites de los polígonos del archivo de intersección cuando éstos no coincidan.
- **Unir:** Permite superponer dos archivos de polígonos para obtener un archivo nuevo, que almacenará toda la información espacial y atributiva contenida en los archivos fuente.
- **Buffer:** Es la determinación de proximidad espacial o cercanía de diversas características geográficas. El área se establece mediante la definición de una distancia específica, o en función

de los datos almacenados en un campo de la tabla de atributos del archivo analizado. El archivo sometido a análisis puede ser de puntos, líneas o polígonos.

- **Agregar:** Permite crear un archivo nuevo a partir de varios archivos de entrada de iguales características, adyacentes desde el punto de vista espacial. Los campos de las tablas de atributos deben estar configurados de igual forma en todos los archivos implicados al objeto, siempre manteniendo la consistencia en el resultado; de lo contrario algunos datos se perderán.
- **Disolver:** Fusiona los polígonos cuyos valores son iguales en el campo de la tabla de atributos que haya sido seleccionado para ejecutar esta tarea.

GvSig

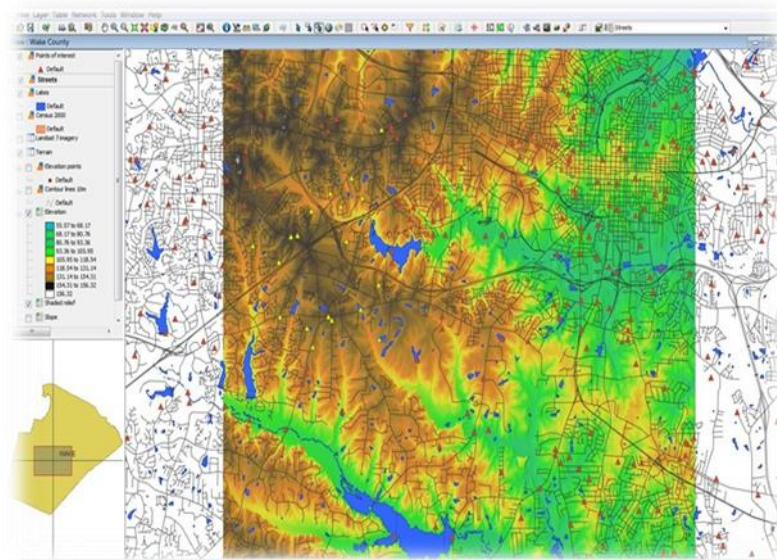


Figura 5: Interfaz de GvSig (Terry, 2009)

GvSig es un SIG libre y de código abierto, maneja información geográfica vectorial y ráster, soporta diferentes formatos vectoriales y ráster. Puede integrar datos en el sistema a partir de servidores de mapas con especificaciones del OGC¹⁹, en otras palabras que contenga servicios de WMS (Servicio de mapas en la Web), WFS (Servicio de Características en la Web), WCS (Servicio de Cobertura Web),

¹⁹**OGC:** Es una consorcio que se encarga de la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la World Wide Web.

Servicio de Catálogo y Servicio de Nomencladores (Terry, 2009).

Funciones de análisis espacial de GvSig:

- **Área de Influencia:** Crea una nueva capa vectorial de polígonos, generados como zona de influencia alrededor de la geometría de los elementos vectoriales de una capa de entrada. Por ejemplo:
 - ¿Qué zonas urbanas no tienen paradas de autobuses en un radio de 500m?
 - ¿Qué zonas urbanas carecen de colegios en un radio de 1000m?
 - ¿Qué pozos incumplen la normativa al no respetar la distancia mínima entre dos consecutivos?
- **Recortar:** Permite limitar el ámbito de trabajo de una capa vectorial (ya sea de puntos, líneas o polígonos), extrayendo una zona de interés. Para ello, el usuario deberá proporcionar una capa de entrada (la capa de la que se quiere extraer la zona) y una de recorte, de forma que la unión de las geometrías incluidas en la capa de recorte definirá el ámbito de trabajo. Es de gran utilidad para limitar el ámbito de trabajo a una zona concreta de detalle, pero toda la cartografía es de carácter general.
- **Disolver:** Actúa sobre una sola capa de entrada, cuyo tipo de geometría ha de ser forzosamente de polígonos. El proceso analiza cada polígono de la capa de entrada, de tal forma que fusionará en un solo polígono aquellos polígonos que tomen idéntico valor para un campo específico. Permite introducir el criterio espacial en la decisión de fusionar varios polígonos.

Conclusiones del epígrafe

QGIS implementa varias funcionalidades que apoyan el proceso de análisis espacial, brindándole escalabilidad a la arquitectura que presenta esta herramienta. Permite incorporar nuevas funcionalidades a partir de desarrollo de plugins²⁰ en python garantizando así mejores prestaciones de la aplicación a partir de nuevas exigencias de los clientes.

La familia de ArcGis al ser de software propietario no permite acceder a su código fuente, por lo que no se puede adaptar al negocio específico a desarrollar, quedando descartada la posibilidad de su utilización.

²⁰**Plugin:** Es un complemento para un programa, añadiéndole una funcionalidad al mismo.

GvSig no se pudo utilizar porque es una aplicación fundamentalmente para realizar análisis complejos, esta herramienta presenta disímiles funciones, prácticamente todas sus funciones tienen que ser ejecutadas a base de comandos. Además no cumple con las políticas a tener en cuenta para el trabajo con la información geográfica tratadas en el proyecto.

Por lo explicado anteriormente se escoge como herramienta base para desarrollar la solución propuesta QGIS, pues además de ser una aplicación desarrollada por la comunidad de software libre; permite extraer de su código fuente las bibliotecas más importantes para adaptarlas a un negocio específico enfocándose principalmente en el trabajo con la información geográfica.

1.4 Conclusiones parciales

Durante el desarrollo de este capítulo se realizó un análisis del objeto de estudio y de la situación problemática, así como de los principales SIG existente a nivel mundial que utiliza análisis espacial, por lo que se concluye que:

- El estudio realizado sobre el proceso de planificación y control de la guardia que utiliza la DSP en la UCI, determinó las insuficiencias del proceso análisis espacial de los activos, la no determinación del como ubicar correctamente los puestos de guardia y la carencia de elementos para el conocimiento de cómo es que se realiza la guardia en dicha institución.
- Se selecciona la herramienta QGIS por las ventajas que genera su utilización en comparación con otras herramientas existentes a nivel mundial, disponiendo de un conjunto de bibliotecas que facilitarán el trabajo con la información geográfica. Además cumple con las políticas tecnológicas llevadas a cabo por el proyecto SIG-Desktop.

CAPÍTULO 2. SOLUCIÓN TÉCNICA

En este capítulo se realiza un análisis de los siguientes elementos: metodología de software, lenguaje de modelado, herramienta de modelación, lenguaje de programación y framework²¹ de desarrollo que se empleará en la implementación. Se permitirá conocer las herramientas seleccionadas en el desarrollo de la solución propuesta.

2.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software definen Quién debe hacer qué, cuándo y cómo debe hacerlo para obtener los distintos productos en un determinado proceso de software. Su objetivo principal es guiar a los desarrolladores en el diseño e implementación de nuevas aplicaciones de probada calidad. Su necesidad viene dada por la engorrosa tarea de la creación de aplicaciones de software que respondan a las habilidades y restricciones de los estándares de software en el mundo. Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo, las cuales se pueden clasificar en dos grupos (Carrillo, 2008.):

- **Las metodologías pesadas:** Están orientadas al control de los procesos, estableciendo rigurosamente la definición de roles, herramientas, actividades, artefactos y notaciones para el modelado y documentación detallada. Prestan mayor énfasis a la planificación y control del proyecto, en especificar de forma precisa los requisitos y el modelado.
- **Las metodologías ágiles:** Proporcionan una serie de pautas, principios y técnicas pragmáticas que aseguran la entrega del proyecto con menos complicación y satisfactoriamente. Esto lo cumple a través de muestra de versiones parcialmente funcionales del software al consumidor en cortos periodos de tiempo para que se pueda evaluar y sugerir cambios en el producto según se va desarrollando.

²¹**Framework:** Un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular.

Proceso Unificado de Desarrollo de Software

RUP (*Rational Unified Process* según sus siglas en inglés) es una metodología de desarrollo de software pesada, la cual se centra en la definición detallada de los procesos, tareas a realizar y las herramientas a utilizar. Pretende prever todo el desarrollo del producto de antemano por lo que requiere una extensa documentación. Jacobson define la metodología de la siguiente manera: “Se puede definir como un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos (CHACÓN, 2006).

La metodología RUP cuenta con 3 características esenciales:

- **Dirigida por casos de uso:** Los casos de uso constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo del software.
- **Centrada en la arquitectura:** Se establece una arquitectura candidata al inicio del desarrollo del sistema que funcionará como guía y se irá perfeccionando en las restantes fases de desarrollo.
- **Iterativa e incremental:** El desarrollo iterativo garantiza la corrección de los errores en cada iteración, brindando la posibilidad de que los elementos sean integrados continuamente, lo que garantiza un producto más robusto y de mayor calidad.

RUP se repite a lo largo de una serie de ciclos de desarrollo que constituyen la vida de un sistema, donde cada ciclo concluye con una versión del producto. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Todas las fases se subdividen en n iteraciones y terminan con un hito (ver Figura 6).

Las nociones de casos de uso y de escenarios utilizadas en RUP han demostrado ser una manera excelente de capturar los requerimientos funcionales y asegurarse que direccionan el diseño, la implementación y la prueba del sistema, logrando así que el sistema satisfaga las necesidades del usuario. Una característica a destacar de RUP es que describe cómo controlar, rastrear y monitorizar los cambios, en ambientes en los cuales el cambio es inevitable, para permitir un desarrollo iterativo exitoso.

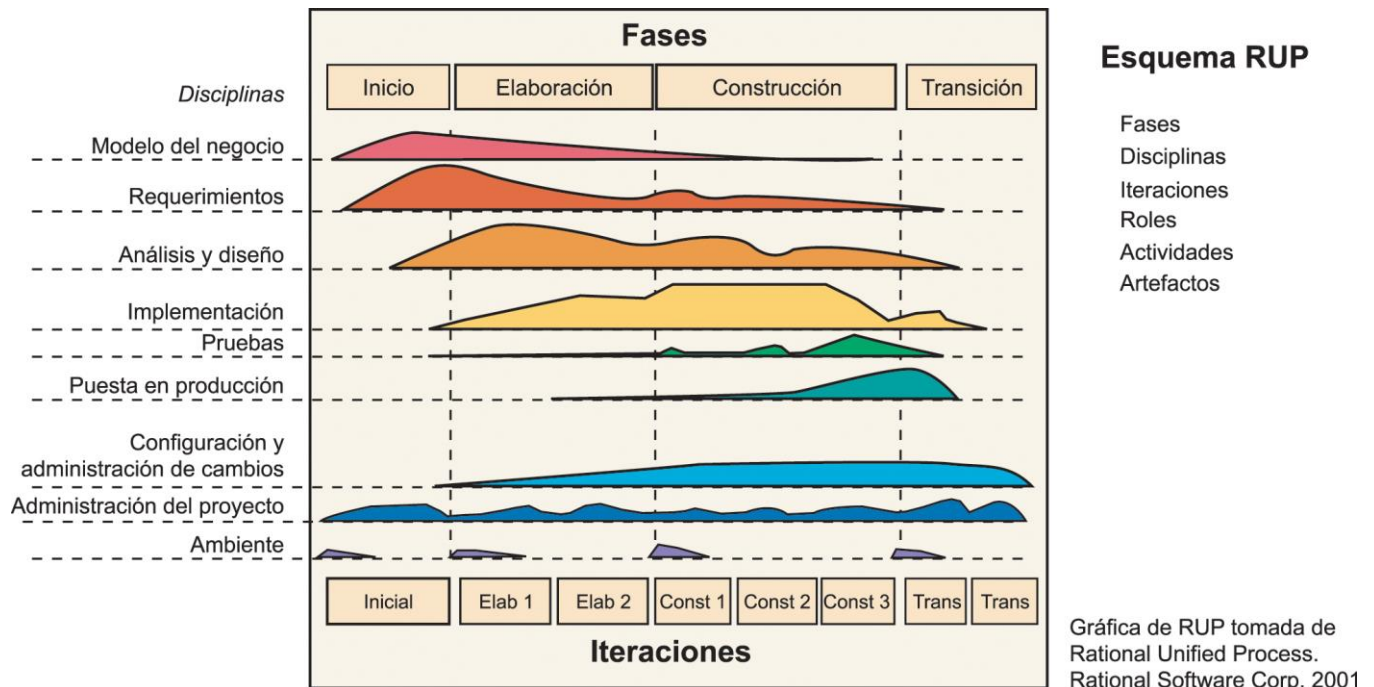


Figura 6: Fases y flujos que utiliza RUP

RUP se propone como metodología para la realización de la solución propuesta, teniendo en cuenta que el SIG GeoQ-Guardián 2.0 utiliza esta metodología para que los profesionales que intervienen en su producción (que son muy variables debido a las políticas por las que se rigen en la universidad respecto al vínculo docencia-producción) posean la documentación necesaria para la interacción con la solución. Se garantiza que con su aplicación se genere la documentación y artefactos imprescindibles para que los futuros clientes tengan como base una guía para su entendimiento, así como los futuros técnicos que contará el proyecto.

2.2 Lenguaje de Modelado

La falta de estandarización en la forma de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos desarrolladores. Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema.

Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. (Pockin, 2008). Surge por la necesidad de unificar a las industrias, para crear diseños sólidos, y compartir diseños gráficos realizados entre distintos diseñadores y servir de apoyo en los procesos de análisis de un problema. Su objetivo es lograr una aplicación de software robusta, flexible y escalable.

Los diseños logrados usando UML se pueden realizar en cualquier lenguaje orientado a objetos ya que este se aplica a una multitud de diferentes tipos de sistemas, dominios, y métodos o procesos. UML ayuda a los usuarios a entender la realidad desde un punto de vista de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades de dinero en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de los módulos que construirán el software (ORALLO, 2002).

Entre los rasgos principales que han contribuido a hacer de UML el estándar de la industria en la actualidad, se puede mencionar (Jacobson, 2000):

- Puede ser utilizado en las diferentes fases de un proyecto.
- Permite modelar sistemas utilizando técnicas orientadas a objetos.
- Ampliamente utilizado en la industria del software.
- Realiza una verificación y validación del modelo realizado.
- Reemplaza a decenas de notaciones empleadas por otros lenguajes.
- Puede ser aplicado a diferentes tipos de sistemas, dominios y métodos o procesos.

UML en su versión 2.1 se seleccionó como lenguaje de modelado para el desarrollo del módulo puesto que es el lenguaje estándar de modelado para crear planos de software, además de visualizar, especificar, construir, documentar y detallar los artefactos del sistema. Es un lenguaje que permite modelar estructuras complejas utilizando técnicas orientadas a objetos.

2.3 Herramienta CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) en inglés (*Computer Aided Software Engineering*) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y costo (CASE, 2011)

Visual Paradigm

Visual Paradigm (VP) es una herramienta profesional capaz de soportar todo el ciclo de vida del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Está orientada a UML que proporciona apoyo al modelado visual, el cual ayuda a una más rápida construcción de aplicaciones con mejor calidad y a un menor coste. Es definido por los usuarios como necesaria para la captura de requisitos, planificación de software, modelado de clases entre otras funcionalidades. Permite dibujar mayoritariamente todos los diagramas de clases además de admitir código inverso, generación de código desde diagramas y generación de la documentación. Entre las características más notables se pueden encontrar:

- Editor de Detalles de Casos de Uso: entorno para la especificación de los detalles.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Generación de bases de datos.
- Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Distribución automática de diagramas.
- Reorganización de las figuras y conectores de los diagramas UML.

Durante el proceso de desarrollo del software, las herramientas CASE juegan un papel fundamental en este proceso facilitando el trabajo a los analistas e ingenieros en un equipo de desarrollo, para ello se utilizará como herramienta para la representación de diagramas UML: Visual Paradigm en su versión 8.0, el cual presenta características que son factibles a utilizar durante el proceso de desarrollo de software. Es una herramienta que utiliza UML como lenguaje de modelado, así como la creación de diagramas UML, siendo capaces de guiar todo el proceso de desarrollo de software. Visual Paradigm es una herramienta desarrollada con tecnologías libres y multiplataforma, lo que facilita su uso en los diferentes sistemas operativos existentes a nivel mundial.

2.4 Lenguaje de Programación

Un lenguaje de programación es una técnica de comunicación estandarizada para expresar las instrucciones a una computadora. Se trata de un conjunto de reglas sintácticas y semánticas utilizadas para definir los programas de ordenador (INFORMÁTICA, 2006).

Los lenguajes de programación se encargan de relacionar a los seres humanos con los ordenadores, y así manipularlos para automatizar tareas y llevar a cabo instrucciones donde se controla su comportamiento y se realizan diferentes operaciones. Estos lenguajes le permiten al programador especificar con precisión los datos que una computadora tomará sobre una decisión determinada, dichos datos pueden ser almacenados o transmitidos.

Lenguaje de Programación: C++

C++ es un lenguaje de programación de propósito general diseñado con la intención de extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos llegando a ser un lenguaje híbrido. Es, por ende, un lenguaje de programación multiparadigma, poseedor de programación genérica, que se suma a los otros dos paradigmas ya existentes para ese entonces: programación estructurada y programación orientada a objetos (JALÓN, 2008).

Los compiladores de C++ generan código nativo con un alto grado de optimización en memoria y velocidad, lo que lo convierte en uno de los lenguajes más eficientes, siendo a la vez un lenguaje muy flexible que permite programar con múltiples estilos. Como lenguaje de programación mantiene una considerable potencia para programación a bajo nivel pudiéndose utilizar esta característica para realizar implementaciones óptimas, y escribir software de bajo nivel, como drivers y componentes de sistemas operativos, para el desarrollo rápido de aplicaciones, según el marco de trabajo con el que se disponga, aunque se le han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción.

Las características ofrecidas por C++ lo han convertido en un lenguaje de programación popular, utilizado en la construcción de los más variados sistemas informáticos. El SIG GeoQ-Guardián 2.0 ha sido implementado con C++, ofreciendo mayor seguridad y estandarización al poder compilar el mismo código fuente en diversas plataformas. Constituye de mucha ayuda en el tratamiento de la información geográfica

pues permite una compilación de toda la información en un menor tiempo de ejecución. Por tales motivos, el módulo de análisis espacial propuesto será construido con el lenguaje de programación C++ buscando mantener la compatibilidad y garantizar la portabilidad de la solución.

2.5 Framework de Desarrollo

Se puede definir como una estructura tecnológica de soporte, definido con artefactos o módulos de software concretos, con el objetivo de crear las bases para la mejor organización y desarrollo de los proyectos de software. Los framework de desarrollo en la informática han marcado un hito significativo, pues poseen significativas ventajas que simplifican el proceso de implementación y estandarización de los procesos en alguna medida. Refiere ser toda una tecnología o modelo de programación que contiene máquinas virtuales, compiladores, bibliotecas de administración de recursos en tiempo de ejecución y especificaciones de lenguajes, además de incluir una biblioteca de componentes reutilizables. Fueron diseñados con el fin de facilitar el desarrollo de software posibilitando a los proyectos identificar los requerimientos y prestar menos atención a los detalles de programación, quizás de bajo nivel, para proveer un sistema funcional (Willian, 2008).

Luego del estudio de las aplicaciones prácticas que desarrolla y las diferentes conceptualizaciones del término, se pueden precisar como las principales ventajas de la utilización de un framework:

- El desarrollo rápido de aplicaciones. Los componentes incluidos en un framework constituyen una capa que libera al programador de la escritura de código de bajo nivel.
- La reutilización de componentes de software. Los framework son los paradigmas de la reutilización.
- El uso y la programación de componentes que siguen una política de diseño uniforme.

Un framework orientado a objetos logra que los componentes sean clases que pertenezcan a una gran jerarquía de clases, lo que genera como resultado bibliotecas más fáciles de aprender a usar.

Framework de Desarrollo: Quantum GIS

El framework Quantum Gis es un Sistema de información Geográfica de Código abierto, multiplataforma, desarrollado en QT Toolkit y C++. Se publica bajo Licencia Pública General (GNU General Public License).

“QGIS puede ver y superponer datos vectoriales y ráster en diferentes formatos y proyecciones sin conversión a un formato interno o común” (Dawson, 2007).

Los formatos admitidos incluyen tablas de PostgreSQL con capacidad espacial, archivos en formatos ráster e imágenes admitidas por la biblioteca GDAL (Geospatial Data Abstraction Library), datos ráster y vectoriales de GRASS de bases de datos entre otros. Es capaz de diseñar mapas y explorar datos espaciales de forma interactiva con una interfaz amigable. Permite además crear, editar, administrar y exportar mapas vectoriales en varios formatos.

Ofrece herramientas de digitalización para formatos admitidos por OGR²² y capas vectoriales de GRASS, crear y editar archivos shape y capas vectoriales de GRASS, geocodificar imágenes con el complemento Georreferenciador, herramientas GPS²³ para importar y exportar formato GPX²⁴ y convertir otros formatos GPS a GPX, crear capas PostGIS a partir de archivos shape con el complemento SPIT, guardar capturas de pantalla como imágenes georreferenciados.

QGIS actualmente ofrece herramientas de análisis vectorial, muestreo, geoprocésamiento, geometría y administración de bases de datos. También puede usar las herramientas de GRASS integradas, que incluyen la funcionalidad completa de GRASS de más de 350 módulos. QGIS también se puede usar como cliente WMS o WFS y como servidor WMS además de permitir exportar datos a un archivo Mapfile y publicarlos en Internet usando un servidor web con UMN MapServer instalado (OSGeo, 2009).

Con la utilización del framework Quantum Gis en su versión 1.8 se disponen de herramientas para el tratamiento y análisis de la información geográfica, contando con librerías que permiten el trabajo con la información geográfica.

2.6 Entorno de Desarrollo Integrado (IDE)

Un Entorno de Desarrollo Integrado o en inglés, *Integrated Development Environment* (IDE), es un ambiente para escribir la lógica de aplicación y el diseño de interfaces de aplicaciones. Se puede caracterizar como el programa informático compuesto de un conjunto de herramientas que utilizan los

²² **OGR**: Es una librería que permite abrir y trabajar con ficheros vectoriales en bastantes formatos GIS.

²³ **GPS**: Es un sistema satelital de posicionamiento.

²⁴ **GPX**: Es un esquema XML pensado para transferir datos GPS entre aplicaciones.

programadores para generar código (CORPORATION, 2011). Estos pueden ser multiplataforma, además soportan diversos lenguajes de programación, tener un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de forma amigable.

Entorno de Desarrollo Integrado: Qt Creator

Qt Creator constituye una plataforma que se ajusta a las necesidades de los desarrolladores. Se centra en proporcionar características que ayudan a los nuevos usuarios de Qt a aprender y comenzar a desarrollar rápidamente, también aumenta la productividad de los desarrolladores con experiencia en Qt (CORPORATION, 2011).

Es un IDE que admite desarrollar aplicaciones implementadas con C++ de manera sencilla y rápida, soportado por Windows, Linux y MacOSX. Presenta toda la documentación de Qt, ejemplos a través de la ayuda y tiene su código fuente bajo la licencia GNU GPL, de manera que los proyectos desarrollados en él pueden ser desarrollados bajo esta licencia.

Dentro de las múltiples funcionalidades que brinda se encuentran:

- Editor de código con soporte para C++ y QML²⁵, proporcionando ayuda sensible al contexto.
- Permite la integración con la mayoría de los sistemas de control de versiones más difundidos internacionalmente.
- Herramientas para la rápida navegación del código.
- Resaltado de sintaxis y auto-completado de código.
- Soporte para refactorización de código.

La selección del IDE Qt Creator en su versión 4.8.1 para el desarrollo de la aplicación proporciona todas las herramientas para diseñar y desarrollar, permitiendo crear aplicaciones de escritorio para diversos Sistemas Operativos haciendo uso del CMAKE²⁶, herramienta multiplataforma que se utiliza para generar y/o automatizar código, construir, probar y empaquetar software independientemente de la plataforma en la que se desarrolle.

²⁵**QML:** Lenguaje declarativo de Meta-Objetos basado en JavaScript que permite diseñar interfaces de usuario.

²⁶**CMAKE:** Cross Platform Make (make multiplataformas)

2.7 Conclusiones Parciales

Para el desarrollo del sistema propuesto se utilizará como metodología de desarrollo del software RUP, utilizando como lenguaje de modelado a UML en su versión 2.1 y Visual Paradigm en su versión 8.0 para la modelación. El sistema se desarrollará soportado en una arquitectura orientada a objetos. Se utiliza además para el desarrollo de la aplicación como framework de desarrollo QGIS en su versión 1.8, utilizando como lenguaje de programación C++ y como IDE de desarrollo QtCreator en su versión 2.5.0.

La selección de estas herramientas y tecnologías utilizadas se lleva a cabo teniendo en cuenta que cumplan con los principios establecidos por la universidad, siendo software libre y desarrollados bajo la licencia GNU GPL, además de que las mismas son utilizadas por el equipo de desarrollo del proyecto SIG-Desktop, lo que facilita la continuación de su desarrollo garantizando a gran medida el proceso de desarrollo y la obtención de la documentación técnica asociada al sistema.

CAPÍTULO 3. CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se describen las características de la solución propuesta y se realiza el proceso de modelación del módulo propuesto, para una mejor comprensión del contexto en que se desarrolla. Para ello se define el modelo de dominio, se especifican los requisitos funcionales y no funcionales que se consideran necesarios, se identifican los actores y se describen los principales casos de uso del sistema, además de la elaboración de los diagramas correspondientes para su correcto entendimiento.

3.1 Modelo de Dominio

En el estudio realizado y en correspondencia con el problema a resolver no se definen claramente los procesos elementales del negocio, debido a que existen restricciones en las exigencias espacio-temporales para la obtención del producto. Además no existe conocimiento de causa en la caracterización de la población objeto (cliente del producto) lo que permitió sustentar la elaboración del mismo en la utilización de un modelo de dominio en lugar de requerir una modelación completa del negocio.

“El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Esto es así ya que cuando se realiza la programación orientada a objetos, se supone que el funcionamiento interno del software va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de dominio constituye una primera versión del sistema” (GARCERANT, 2008).

El modelo de dominio es una representación visual estática del entorno real del sistema expresado a través de un diagrama, en este se describen las entidades implicadas en el sistema y sus relaciones. Se crea para documentar los conceptos dominantes y la terminología del sistema; se centra en una parte del negocio, la relacionada con el ámbito; tiene como objetivo principal ayudar a comprender los conceptos que utilizan los usuarios y con los que deberá trabajar la aplicación.

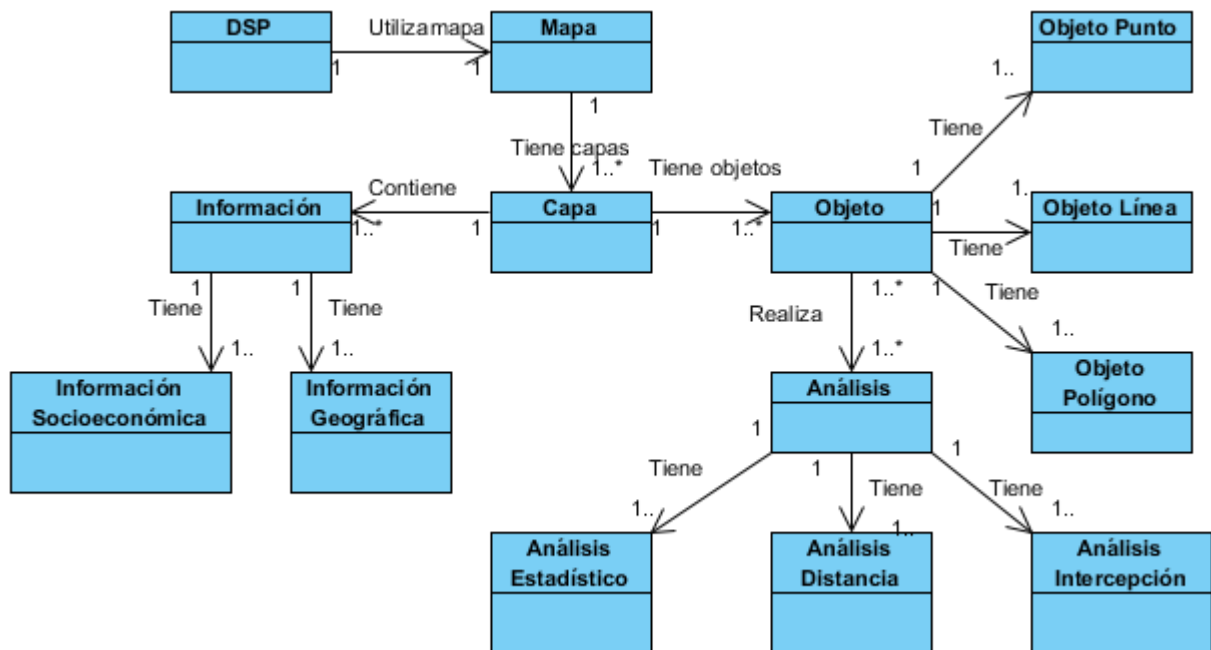


Figura 7: Modelo de Dominio

Glosario de términos del dominio

DSP: Dirección de Seguridad y Protección de la UCI que necesita utilizar el software para realizar análisis espacial a la información geográfica que posee.

Mapa: Representación geográfica de una parte de la superficie terrestre, en la que se da información relativa a una ciencia determinada.

Capa: Se utilizan para controlar la visibilidad de la geometría. Estas se superponen y logran la representación de un mapa aportando información.

Objeto: Elementos contenidos en una o muchas capas de información.

Línea, Punto y Polígono: Objetos contenidos en las capas.

Información: Datos informativos de una capa determinada.

Información geográfica: Datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales.

Información socio-económica: Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier centro o lugar de interés.

Operación: Acción que desee realizar el cliente para relacionar objetos entre sí, que se encuentren en una u otras capas.

Intersección: Operación que desea realizar el cliente. Para que exista una intersección los objetos que se relaciones entre sí, deben de tener un conjunto de elementos de coincidencia entre ellos.

Distancia: Operación que desea realizar el cliente. Se realiza la operación distancia entre dos puntos específicos.

Estadístico: Operación que desea realizar el cliente. Brinda datos estadísticos acerca de un objeto determinado.

Descripción del dominio

La DSP utiliza los mapas para la planificación de la guardia obrera-estudiantil. Estos están compuestos por capas que contienen información socioeconómica y geográfica. Las capas poseen objetos que pueden ser puntos, líneas o polígonos; con los cuales se puede realizar análisis estadístico, de distancia o de intercepción.

3.2 Requisitos Funcionales

Los requisitos funcionales especifican acciones que debe poder realizar un sistema, sin tener en cuenta las restricciones físicas (PRESSMAN, 2005), además definen su comportamiento de salida y entrada. A continuación se presentan los requerimientos funcionales que el sistema debe cumplir en su implementación.

El sistema debe ser capaz:

- RF 1: Calcular la distancia entre dos puntos.
- RF 2: Calcular el centro de un polígono.
- RF 3: Calcular la coordenada media entre varios puntos.
- RF 4: Calcular la coordenada media de un polígono.
- RF 5: Calcular la coordenada media de una recta.
- RF 6: Calcular la cantidad de puntos presentes en un polígono
- RF 7: Calcular el punto de intersección entre dos líneas.
- RF 8: Calcular la intercepción de una recta a un polígono.
- RF 9: Calcular la desviación estándar en una capa puntos.
- RF 10: Calcular la desviación estándar de un polígono.
- RF 11: Calcular la desviación estándar de una línea.
- RF 12: Visualizar los atributos de una recta.
- RF 13: Visualizar los atributos de un polígono.
- RF 14: Visualizar los atributos de un punto.
- RF 15: Calcular la media de una recta.
- RF 16: Calcular la media de un polígono.
- RF 17: Calcular la mediana de una recta.
- RF 18: Calcular la mediana de un polígono.
- RF 19: Calcular la mediana de un punto.
- RF 20: Calcular el coeficiente de variación de una recta.
- RF 21: Calcular el coeficiente de variación de un polígono.
- RF 22: Calcular el coeficiente de variación de un punto.
- RF 23: Mostrar los nodos presentes en un polígono.
- RF 24: Mostrar los nodos presentes en una recta.
- RF 25: Transformar un polígono a polilíneas.
- RF 26: Transformar polilíneas en un polígono.

3.3 Requisitos no Funcionales

Los Requisitos No Funcionales (en lo adelante RNF) son propiedades o cualidades que el producto debe tener. Normalmente están vinculados a requisitos funcionales, es decir una vez se conozca lo que el

sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser (PRESSMAN, 2005).

Usabilidad

- El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.
- Las funcionalidades principales del sistema estarán orientadas a iconos para un mayor reconocimiento por parte del usuario.

Fiabilidad

- Debido a la arquitectura que presenta el sistema, siendo más robusto al no tratarse de un sistema de gestión que requiera mantenimiento y optimización en el almacenamiento, se estima un tiempo promedio de 6 meses entre posibles fallas.
- El tiempo medio de reparación, en caso de un fallo es de 7 días.

Soporte

- La aplicación recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y los clientes.

Restricciones de diseño e implementación

- Diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.

Requisitos para la documentación de usuario en línea y ayuda del sistema

- El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.

Interfaz

Interfaz de Usuario

El sistema debe:

- Tener una apariencia profesional y un diseño gráfico sencillo.

Interfaces de Hardware

Para las PCs clientes:

- Se requiere al menos 1GB de memoria RAM.
- Se requiere al menos 40 GB de disco duro.
- Procesador 512 MHz como mínimo.

Interfaces de Software

La construcción de la aplicación funcionará bajo los conceptos del patrón arquitectónico Modelo Vista Controlador. Las PCs clientes:

- El módulo de análisis espacial es un sistema multiplataforma que puede ejecutarse en los siguientes sistemas operativos:
 - GNU/Linux.
 - Windows.
 - Windows Server 2000 o superior.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGis como extensión de PostgreSQL para el soporte de datos espaciales.

Interfaz de Comunicación

- El módulo de análisis espacial garantizará mediante su interfaz la configuración del entorno de trabajo mediante funcionalidades propias como ocultar y mostrar paneles, así como elementos para cambiar las vistas, las escalas y las capas que serán visibles en la interacción.

Requisitos de Licencia

- El módulo de análisis espacial 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.

Requisitos Legales, Derecho del Autor y otros

- El módulo de análisis espacial 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, el módulo de análisis espacial se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

3.4 Descripción del sistema propuesto

Un caso de uso constituye una técnica utilizada para describir el comportamiento del sistema, a través de un documento narrativo que define la secuencia de acciones que obtienen resultados de valor para un actor que utiliza un sistema para completar un proceso, sin importar los detalles de la implementación.

Los actores se definen como los roles que puede tener un usuario; pueden ser humanos, otros sistemas, máquinas, hardware, etc. que interactúan con un sistema para de esta forma intercambiar datos.

Descripción de los Actores

Actor	Descripción
Usuario	Cualquiera persona que interactúe con la aplicación y desee realizar análisis espacial utilizando información geográfica.

Tabla 1: Descripción del Actor

Descripción de casos de uso del Sistema.

A continuación se muestra el Diagrama de Casos de Uso del Sistema donde están representados los distintos actores que interactúan con el sistema y luego la descripción textual de cada uno de estos.

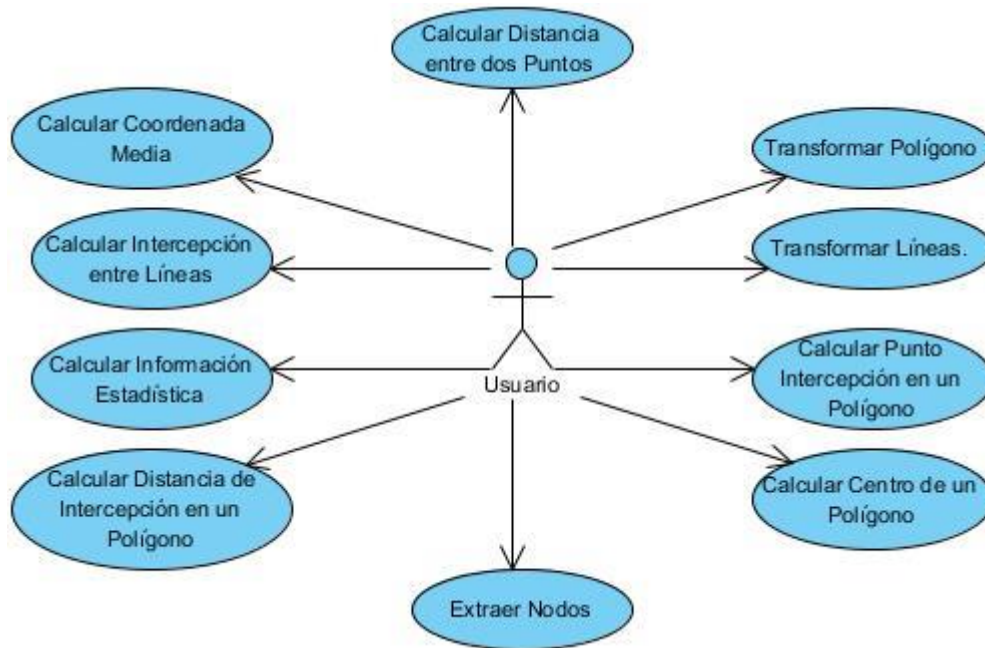
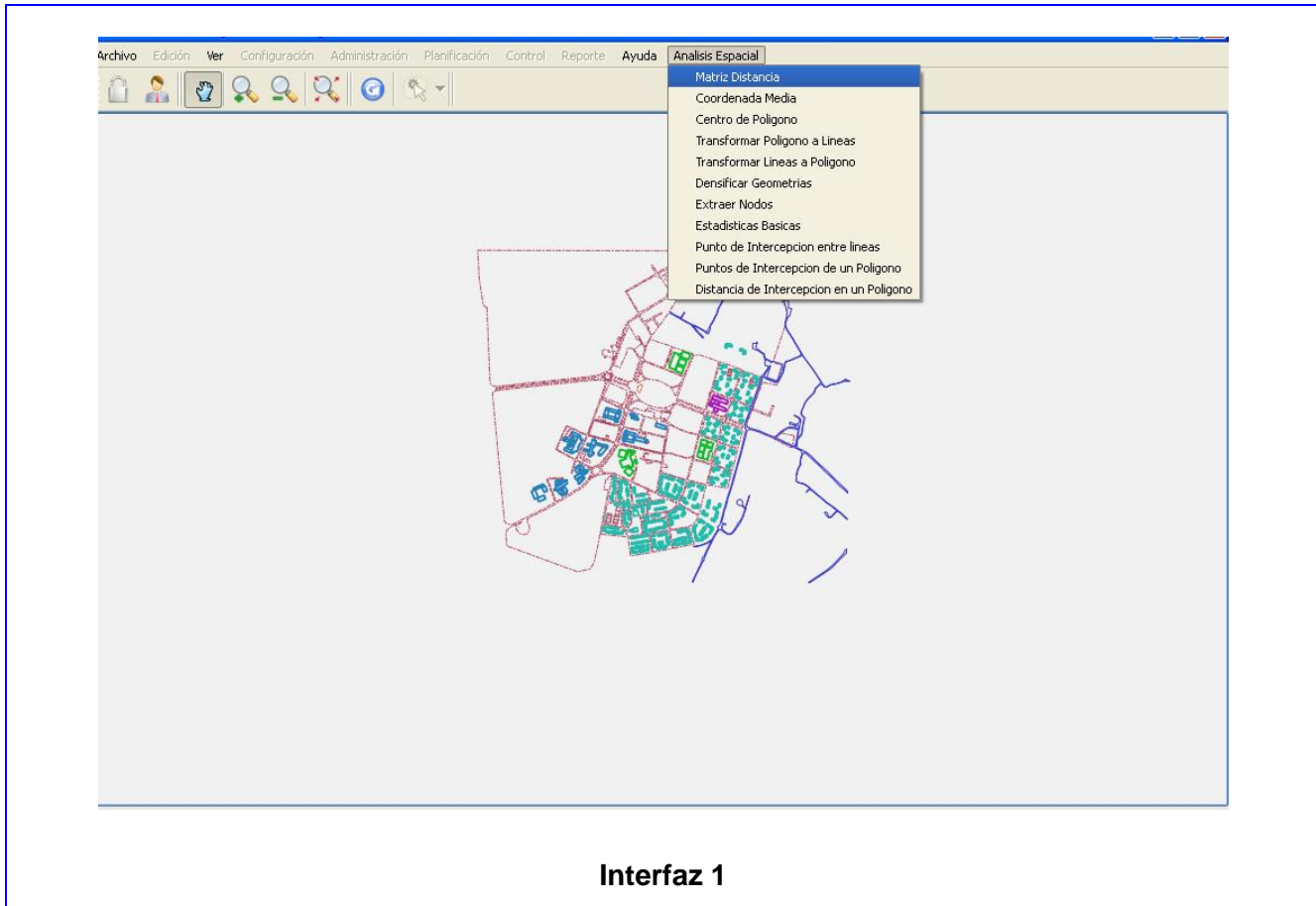


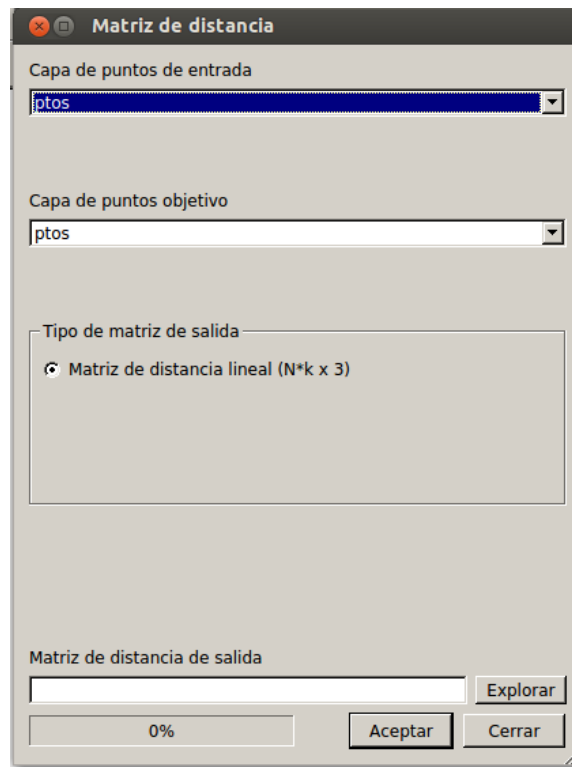
Figura 8: Diagrama de Casos de Uso del Sistema.

Descripción textual de los casos de uso del Sistema

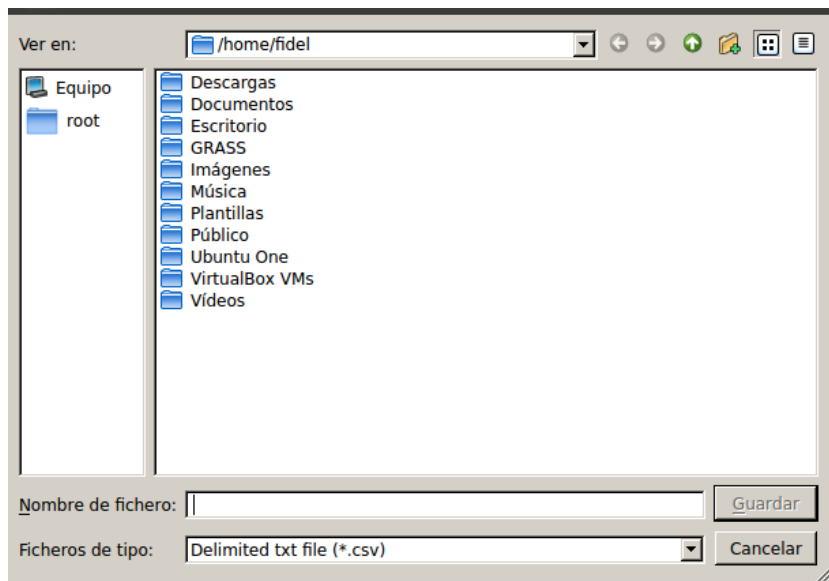
Caso de uso	Calcular Distancia entre dos Puntos
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción Matriz Distancia del menú Análisis Espacial. El sistema muestra una interfaz con la lista de las capas de puntos cargadas en el panel de capas. El usuario selecciona la capa de puntos y el sistema permite elegir la dirección para guardar, concluyendo así el caso de uso.
Precondiciones	Debe existir al menos una capa de puntos cargada en el panel de capas.
Referencias	RF 1
Prioridad	Crítico
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso se inicia cuando el usuario selecciona la opción Matriz Distancia en el menú Análisis Espacial. Ver interfaz 1.</p>	<p>2. El sistema muestra una interfaz con una lista de las capas de puntos cargadas en el panel de capas. Ver Interfaz 2.</p>
<p>3. El usuario selecciona las capas de puntos y hace clic izquierdo sobre el botón Explorar.</p>	<p>4. El sistema muestra una interfaz que brinda la opción de selección de la dirección y el nombre para guardar el fichero. Ver Interfaz 3.</p>
<p>5. El usuario selecciona la dirección del fichero, entra el nombre del fichero y hace clic izquierdo en el botón Guardar.</p>	<p>6. El sistema guarda la dirección y el nombre, regresa al paso 2.</p>
<p>7. El usuario hace clic izquierdo en el botón Aceptar</p>	<p>8. El sistema guarda el resultado en la dirección seleccionada, terminando el caso de uso.</p>
<p>Prototipo de Interfaz</p>	





Interfaz 2



Interfaz 3

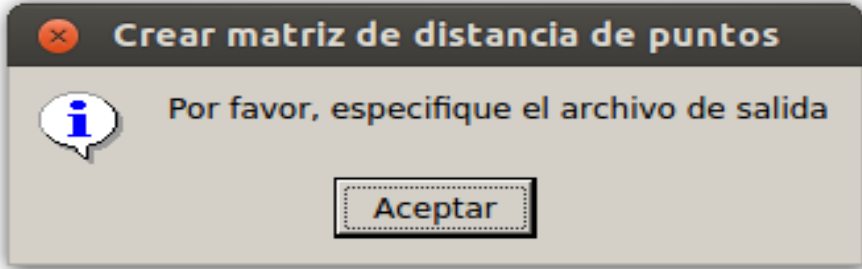
Flujo Alterno	
5. El usuario hace clic izquierdo en el botón Cancelar. Ver Interfaz 3.	6. El sistema cierra la interfaz para la selección del fichero y regresa al paso 2.
	8. El sistema muestra un mensaje de información "Por favor, especifique el archivo de salida". Ver interfaz 4.
9. El usuario hace clic izquierdo en el botón Aceptar y regresa al paso 2.	
<p>Prototipo de Interfaz</p>  <p>Interfaz 4</p>	
7. El usuario hace clic izquierdo en el botón Cancelar. Ver interfaz 2.	8. El sistema no guarda el resultado en la dirección seleccionada, terminando el caso de uso.
Poscondición	Se guarda en un fichero la distancia entre los puntos.

Tabla 2: Descripción del CU Calcular la distancia entre dos puntos.

Las descripciones de casos de uso se encuentran en el documento Modelo del Sistema del expediente de proyecto.

3.5 Conclusiones Parciales

En el presente capítulo se describió la solución propuesta, caracterizando los procesos del negocio, procediendo a su modelado y describiendo los casos de uso del negocio. Se esbozaron los requisitos funcionales y no funcionales, así como los diagramas de clases persistentes, concluyendo que:

- El modelado del dominio permitió tener un conocimiento acerca del funcionamiento interno del software, así como de los actores que interactúan con él, permitiendo describir las entidades implicadas en el sistema y sus relaciones.
- Los requisitos funcionales y no funcionales brindan una breve descripción de las características que presenta el sistema, lo cual permitió la creación del Diagrama de Caso de Uso del Sistema.
- La descripción de los casos de usos brindaron de forma gráfica una información más detallada de lo que se desea desarrollar, así como tener un punto de partida para el desarrollo del sistema.

CAPÍTULO 4. IMPLEMENTACIÓN Y VALIDACIÓN DE LOS RESULTADOS

Como resultado del flujo de trabajo de requisitos se tiene una vista externa del sistema, expresado en el lenguaje del cliente, describiendo lo que se espera de él a través de casos de uso. En este capítulo se llevará a cabo una vista interna del lenguaje descrito por los desarrolladores a través de los flujos de trabajo diseño, implementación y prueba. En cada uno de estos flujos de trabajo se verán reflejados cada uno de los artefactos llevados a cabo en la construcción del software (diagramas de clases del diseño, diagramas de componente y despliegue), así como los patrones arquitectónicos y de diseños a utilizar durante el diseño del software. Una vez realizada la implementación del sistema se realizarán los diseños de casos de pruebas, con el objetivo de verificar que el sistema cumpla con las descripciones realizadas por los analistas y con las necesidades principales del cliente.

4.1 Patrón arquitectónico

Un patrón arquitectónico se define como: "familia de sistemas en términos de un patrón de organización estructural. Más específicamente, un estilo arquitectónico determina el vocabulario de los componentes y conectores que se pueden utilizar en casos de ese estilo, junto con un conjunto de restricciones sobre cómo pueden ser combinados. Estos pueden incluir restricciones topológicas en las descripciones arquitectónicas (por ejemplo, sin ciclos). Otras restricciones, por ejemplo, que tienen que ver con la semántica de la ejecución, también podría ser parte de la definición de estilo" (GRACIA, . 2005).

Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Propone que son plantillas para arquitecturas de software concretas, que especifican las propiedades estructurales de una aplicación - con amplitud de todo el sistema - y tienen un impacto en la arquitectura de subsistemas. La selección de un patrón arquitectónico es, por lo tanto, una decisión fundamental de diseño en el desarrollo de un sistema de software (BUSCHMANN, 2006).

Realizado el estudio de las características principales y análisis de las definiciones de los patrones arquitectónicos, se determinó la utilización del patrón arquitectónico Modelo Vista Controlador basado en

la arquitectura que posee el framework de desarrollo identificado para desarrollar la solución. A continuación se argumentan teóricamente los criterios asumidos para la selección y los principios básicos presente en la misma.

Modelo Vista Controlador (MVC)

El patrón arquitectónico MVC fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales están dadas por el hecho de que, el modelo, las vistas y los controladores se tratan como entidades separadas (ver Figura 9); esto hace que cualquier cambio producido en el modelo se refleje automáticamente en cada una de las Vistas (López, 2010).

Definición de las partes

El modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo.

La vista es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa preferentemente con el controlador, pero es posible que trate directamente con el modelo a través de una referencia al propio modelo.

El controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo, centra toda la interacción entre la vista y el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo.

Este modelo de arquitectura presenta varias ventajas:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.

- Interfaz de Programación de Aplicaciones API (*Application Programming Interface*) muy bien definida; cualquiera que use el API, podrá reemplazar el modelo, la vista o el controlador, sin aparente dificultad.
- Conexión entre el modelo y sus vistas dinámicas; se produce en tiempo de ejecución, no en tiempo de compilación.

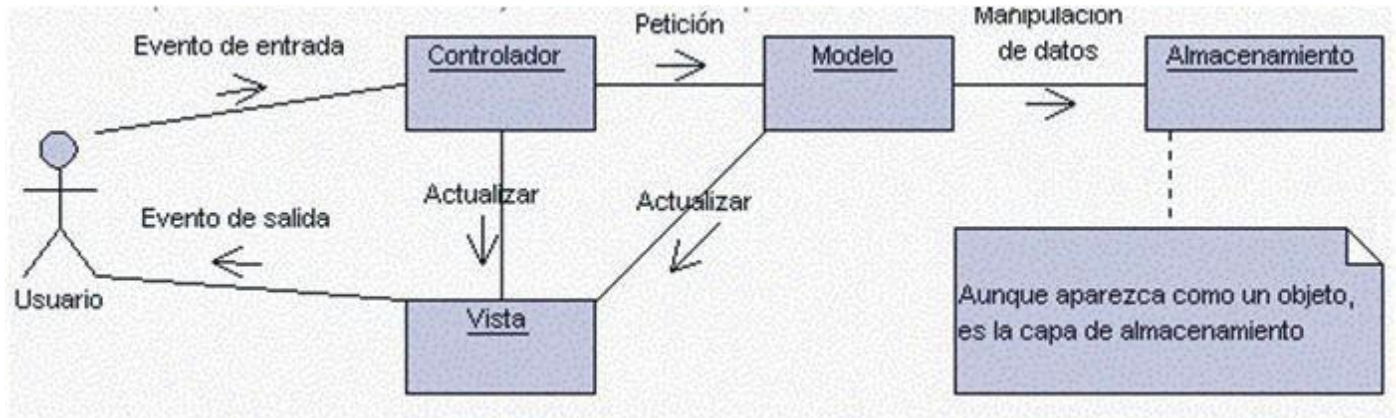


Figura 9: Interrelación entre los elementos del patrón MVC (López, 2010)

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si uno de los componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas. Este escenario contrasta con la aproximación monolítica típica de muchos programas de pequeña y mediana complejidad. Todos tienen un Frame que contiene todos los elementos, un controlador de eventos, un cúmulo de cálculos y la presentación del resultado. Ante esta perspectiva, hacer un cambio aquí no es nada trivial.

4.2 Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Tratan los problemas del diseño que se repiten y que se presentan en situaciones particulares, con el fin de proponer soluciones a ellas. Estos son soluciones exitosas a problemas comunes y están basados en la

recopilación del conocimiento de los expertos en desarrollo de software, con el fin de ayudar a no cometer los mismos errores.

Patrones GRASP

Existe una serie de buenas prácticas de recomendable aplicación en el diseño de software entre las que se encuentran los patrones GRASP (por sus siglas en inglés *General Responsibility Assignment Software Patterns*). Son los patrones generales de software para asignación de responsabilidades. Se ha hecho un esfuerzo por incorporar al diseño del módulo de estructura y composición los cinco más usados (Larman, 2001):

Experto

Asigna una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide (Visconti, 2011). Es la base para el bajo acoplamiento y la alta cohesión (patrones que se analizarán luego).

Creador

Asigna a una clase B la responsabilidad de crear una instancia de una clase A. (Visconti, 2011) Supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. Por solo citar un ejemplo de su utilización: la clase `QgsVectorLayer` crea instancias de la clase `QgsFeature` porque contiene objetos de ella.

Bajo acoplamiento

Una clase con bajo acoplamiento no depende innecesariamente de muchas otras. Por lo que una clase con esta característica no se afecta por cambios de otros componentes y es fácil de entender y reutilizar. En el módulo de estructura y composición las clases son lo más independientes posible, lo cual las vuelve más fáciles de reutilizar. Un cambio en una no implica cambios en otras.

Alta cohesión

Una alta cohesión garantiza que las clases con responsabilidades estrechamente relacionadas no realicen un trabajo enorme. Mejoran la claridad y la facilidad con que se entiende el diseño, se simplifican el mantenimiento y las mejoras en funcionalidad, a menudo se genera un bajo acoplamiento, soporta una

mayor capacidad de reutilización. En el módulo de análisis espacial, cada clase tiene responsabilidades moderadas en su área funcional y colabora con las otras para llevar a cabo las tareas.

Controlador

Un Controlador es un objeto de interfaz no destinado al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. Genera un mayor potencial de los componentes reutilizables: garantiza que los procesos de dominio sean manejados por la capa de los objetos del dominio y no por la de la interfaz, como sucede con la clase `QgsVectorLayer`.

Patrones GoF (Gang of Four)

Observador (Observer)

Este patrón (de comportamiento) define una dependencia entre un objeto y otros de manera que cuando un objeto cambia de estado notifica a los otros el nuevo valor de los datos de interés para cada uno de ellos con el fin de que realicen en sus respectivas áreas las actualizaciones necesarias (Gamma, 2005).

4.3 Modelo de Diseño

El modelo de diseño se muestra como una abstracción de la implementación del sistema, debido a que con el modelo de diseño se pretende crear un plano del modelo de implementación capturando los requisitos o subsistemas individuales, interfaces y clases; el mismo procura descomponer el trabajo de implementación de forma entendible mediante la comprensión de aspectos involucrados con requisitos funcionales y no funcionales incorporados a las limitaciones del lenguaje usado, tecnologías y sistemas operativos, en fin constituye una entrada o paso fundamental en las actividades de la implementación.

Diagrama de Clases del Diseño

Durante el diseño de software el diagrama de clases de diseño permite describir gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Este se elabora para tener en cuenta los detalles concretos de la implementación del sistema, describiendo la realización física de los casos de uso que este contiene.

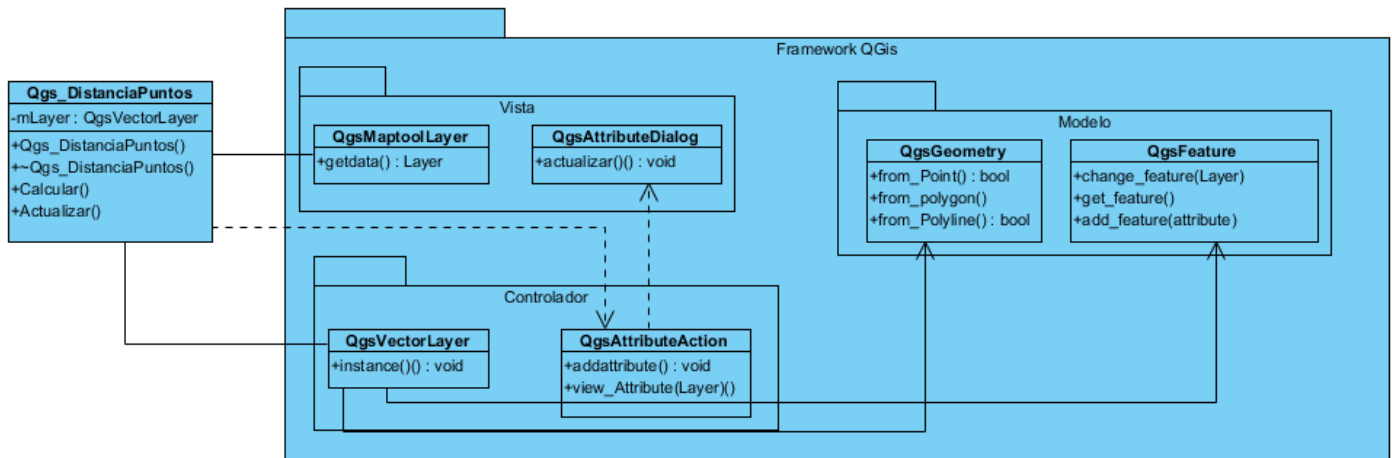


Figura 10: Diagrama de Clases del Diseño CU_ Calcular Distancia entre Puntos

Los diagramas de clases del diseño se encuentran en el documento Modelo de Diseño del expediente de proyecto.

A continuación se ofrecerán las descripciones de las clases que componen el sistema.

QgsAttributeDialog: Clase interfaz del Framework Qgis, representa en un cuadro de diálogo los atributos de un elemento seleccionado.

QgsMapToolLayer: Clase interfaz del framework Qgis, permite capturar los atributos presente en una capa seleccionada.

QgsVectorLayer: Clase controladora del framework Qgis encargada de manipular elementos de una capa vectorial. Se encarga de adicionar, modificar, eliminar, obtener un listado de los elementos de una capa.

QgsFeatureAction: Clase controladora de las acciones que se realizan sobre una capa vectorial, utilizada para las operaciones de inserción y modificación de elementos.

QgsFeature: Clase entidad del framework Qgis, contiene un mapa con los atributos de la tabla de la base de datos para una capa específica.

QgsGeometry: Clase entidad del framework Qgis, permite realizar acciones geométricas con respecto a una figura seleccionada.

4.4 Generalidades de la Implementación

Modelo de Implementación

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema, en este se modela la vista estática del sistema, se muestra organización y dependencias lógicas existentes entre sus componentes. Un componente generalmente está compuesto por clases, pero se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. En fin un modelo de implementación brinda una visión general de lo que tiene que ser implementado.

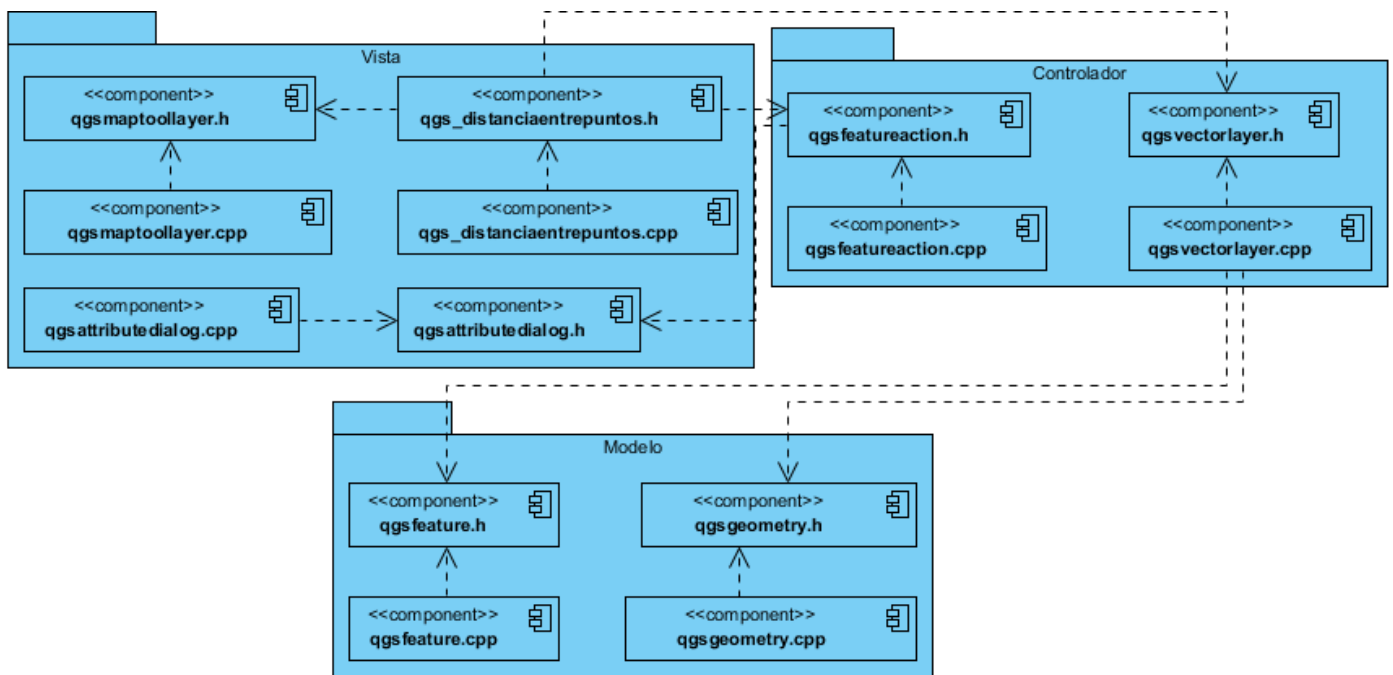


Figura 11: Diagrama de Componentes CU_ Calcular Distancia entre dos Puntos

Los diagramas de componentes se encuentran en el Modelo de Implementación que se encuentra en el expediente de proyecto.

Modelo de Despliegue

Los elementos de diseño al nivel del despliegue indican cómo se ubicará la funcionalidad y los subsistemas del entorno computacional físico que soportará al software (PRESSMAN, 2005).

El diagrama de despliegue muestra la disposición física de los nodos que componen el sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional.

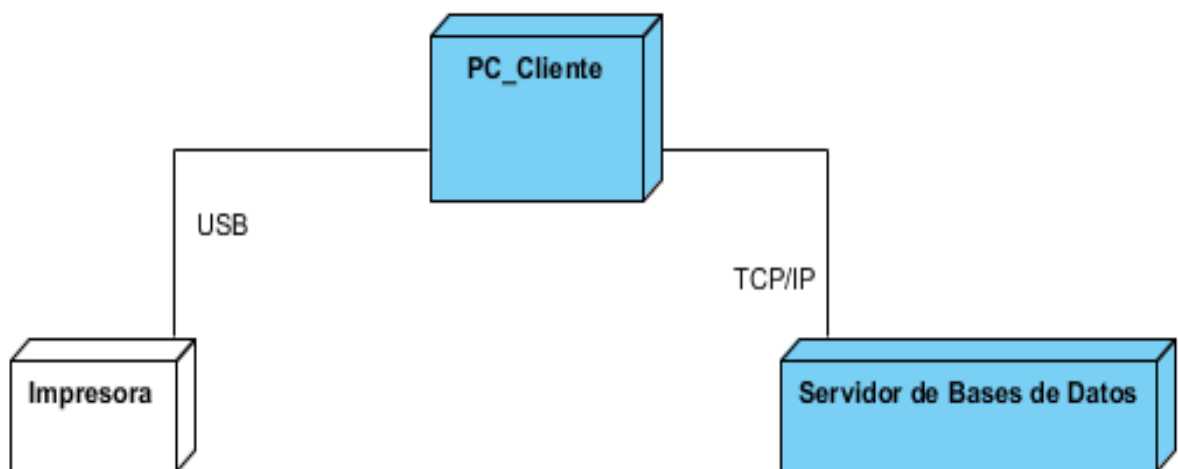


Figura 12: Modelo de Despliegue

4.5 Pruebas

Los errores dentro del proceso de desarrollo del software pueden venir dados desde su creación durante las fases de inicio y análisis hasta su explotación durante el despliegue. La identificación errónea de requisitos o un mal diseño de clases pueden provocar faltas graves en el buen funcionamiento de una aplicación. De ahí que se realicen a las aplicaciones de software numerosas pruebas con el objetivo de descubrir fallas no detectadas hasta ese momento.

La prueba es el proceso de ejecución de un programa con la intención de descubrir un error. Las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y la codificación. La creciente percepción del software como un

elemento del sistema y la importancia de los costes asociados a un fallo del propio sistema, están motivando la creación de pruebas minuciosas y bien detalladas (PRESSMAN 2005).

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 los 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.
- 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.

Pruebas de Caja Negra

“El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados validos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica” (PRESSMAN, 2005).

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.

A continuación se muestra el caso de prueba para el caso de uso “Calcular Distancia entre dos Puntos” utilizando la técnica de Partición de Equivalencia.

Caso de uso: Calcular Distancia entre dos Puntos.

Descripción general del caso de uso: El caso de uso se inicia cuando el usuario selecciona la opción Matriz Distancia del menú Análisis Espacial. El sistema muestra una interfaz con la lista de las capas de puntos cargadas en el panel de capas. El usuario selecciona la capa de puntos y el sistema permite elegir la dirección para guardar, concluyendo así el caso de uso.

Condiciones de Ejecución: Debe existir al menos una capa de puntos cargada en el panel de capas.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: “Calcular Matriz Distancia”	EC 1.1: Calcular distancia exitosamente.	El caso de uso se inicia cuando el usuario selecciona la opción Matriz Distancia. El sistema muestra una interfaz donde el usuario selecciona los datos: <ul style="list-style-type: none"> ➤ Capa de Puntos de Entrada. ➤ Capa de Puntos 	Módulo de Análisis Espacial de GeoQ-Guardián 2.0. <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Análisis Espacial. 3. Clic en la opción Matriz Distancia.

		<p align="center">Objetivo.</p> <p>El usuario selecciona las capas de puntos que desea saber la distancia entre sus puntos y hace clic izquierdo en el botón Explorar.</p> <p>El sistema muestra una interfaz donde el usuario selecciona los datos para guardar el resultado:</p> <ul style="list-style-type: none"> ➤ Dirección del fichero. ➤ Nombre del fichero. <p>El usuario entra la dirección y el nombre donde desea guardar el resultado. Luego hace clic izquierdo en el botón Guardar.</p> <p>El sistema guarda los cambios, visualiza una interfaz y el usuario hace clic izquierdo en el botón Aceptar, terminando el caso de uso.</p>	<ol style="list-style-type: none"> 4. Clic en la opción Capa de Puntos de Entrada. 5. Clic en la opción Capa de Puntos Objetivo. 6. Clic en la opción Explorar. 7. Clic en el botón Guardar. 8. Clic en el botón Aceptar.
<p>EC 1.2: Calcular Distancia sin éxito.</p>		<p>El sistema muestra un mensaje de información “Especifique el archivo de salida”. El usuario hace clic izquierdo en el botón Aceptar para cerrar la</p>	<p>Módulo de Análisis Espacial de GeoQ-Guardián 2.0.</p> <ul style="list-style-type: none"> ➤ Página principal. ➤ Clic en el menú Análisis Espacial.

		ventana que contiene el mensaje de error.	<ul style="list-style-type: none"> ➤ Clic en la opción Matriz Distancia. ➤ Clic en la opción Capa de Puntos de Entrada. ➤ Clic en la opción Capa de Puntos Objetivo. ➤ Clic en la opción Explorar. ➤ Clic en el botón Cancelar. ➤ Clic en el botón Aceptar. ➤ Clic en el botón Aceptar para cerrar la ventana de error.
E.C 1.3: No Calcular Distancia.	No	El usuario hace clic izquierdo en el botón Cerrar y el sistema termina el caso de uso.	<p>Módulo de Análisis Espacial de GeoQ-Guardián 2.0.</p> <ol style="list-style-type: none"> 1. Página principal. 2. Clic en el menú Análisis Espacial. 3. Clic en la opción Matriz Distancia. 4. Clic en la opción Cerrar.

Tabla 3: Secciones a probar en Calcular Matriz Distancia

Descripción de variables

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Capa de puntos de entrada	Lista de Selección	No	El usuario selecciona de la lista la capa que desea saber su distancia.
2	Capa de Puntos Objetivo	Lista de Selección	No	El usuario selecciona de la lista la capa que desea saber su distancia.
3	Dirección del fichero	Campo de texto	No	El usuario selecciona la dirección del fichero.
4	Nombre del fichero	Campo de texto	No	El usuario escribe el nombre del fichero.

Tabla 4: Descripción de las variables en Calcular Matriz Distancia

Matriz de Datos SC 1: “Calcular Matriz Distancia”

Id del escenario	Capa de Puntos de Entrada	Capa de Puntos Objetivo	Dirección del fichero	Nombre del fichero	Respuesta del sistema	Resultado de la prueba
EC 1.1: Calcular Distancia exitosamente.	V “puntos1”	V “puntos2”	V “Datos”	V “ejemplo”	El sistema permite conocer la distancia entre los puntos de la capa de puntos mediante el fichero seleccionado.	Satisfactorio

EC 1.2: Calcular Distancia sin éxito.	V "puntos1"	V "puntos2"	I "vacío"	I "vacío"	El sistema muestra un mensaje de información "Especifique el archivo de salida".	Satisfactorio
EC 1.3: No Calcular Distancia.	N/A	N/A	N/A	N/A	El sistema termina el caso de uso sin calcular distancia.	Satisfactorio

Tabla 5: Matriz de datos Calcular Matriz Distancia

Las descripciones de los casos de prueba del resto de los casos de usos se encuentran en el documento Diseño de Casos de Prueba.

4.6 Conclusiones Parciales

En el presente capítulo se diseñó el sistema propuesto y se procedió a la construcción en términos del diseño, se llevó a cabo la modelación de los diferentes artefactos correspondientes a los flujos de trabajo diseño, implementación y prueba. Por lo que se puede concluir que:

- Con la obtención de los diagramas de clases del diseño se logró una mejor visión de la estructura en términos de clases del sistema propuesto.
- Se obtuvieron los diagramas de componentes permitiendo conocer los componentes internos de la solución, brindando una visión general de lo que se implementó.
- Se obtuvo el diagrama de despliegue brindando la disposición físicas de los nodos que componen el sistema.
- Se diseñaron los casos de pruebas utilizando el método de caja negra, probando el sistema y obteniendo resultados satisfactorios.

CONCLUSIONES

Con el desarrollo de la presente investigación se obtuvo un módulo de análisis espacial para la seguridad y protección de la Universidad de las Ciencias Informáticas que permite analizar espacialmente la planificación y control de la guardia obrera-estudiantil en dicha institución, por lo que se concluye que:

- Las herramientas y tecnologías utilizadas en el desarrollo de la aplicación obedecen a criterios de selección de tecnologías libres y multiplataforma, adecuándose a las políticas que impulsa la institución y el país.
- Todos los requisitos funcionales previstos fueron implementados satisfactoriamente teniendo en cuenta las restricciones no funcionales especificadas a lo largo de la investigación.
- El diseño de las pruebas de caja negra permitió validar el cumplimiento de los requisitos funcionales capturados y comprobar que el subsistema efectúa las acciones que debe realizar en el momento esperado.
- Se logró analizar la planificación de la guardia obrera estudiantil y demás activos del sistema de guardia de la institución, logrando una mayor seguridad de los mismos y contribuyendo a mejorar los resultados en la toma de decisiones.
- Se logró documentar todo el proceso de desarrollo del subsistema para posteriores estudios y modificaciones.

RECOMENDACIONES

En correspondencia con los resultados obtenidos y la experiencia acumulada a lo largo de todo el proceso investigativo se proponen las siguientes recomendaciones:

- Desarrollar la funcionalidad Unión para poder lograr unir varias zonas y tratarlas como una sola, facilitando el trabajo de análisis en función de lograr una correcta planificación de la guardia obrera-estudiantil.
- Desarrollar la funcionalidad Intersección que permita interceptar varias zonas para obtener una según la intercepción establecida y de esta forma obtener una opción que facilite el proceso de toma de decisiones.
- Desarrollar la funcionalidad Diferencia para obtener la diferencia entre varias zonas interceptadas.

BIBLIOGRAFÍA

- Hernández , Ana Luisa. 2012.** *Subsistema de Selección por Localización para el Sistema GeoQ.* Cuba : Univerisidad de Ciencias informáticas, 2012.
- M Visconti. 2011.** Fundamentos de Ingeniería de Software. 2011.
- Alonso, Miguel. 2007.** Definición ABC. [Online] 2007.
<http://www.definicionabc.com/ciencia/cartografia.php>.
- Arteaga, Luis. 2004.** Las Nuevas Geografías. Barcelona : Salvat Editores, 2004.
- Bosch, Charles. 2011.** Quantum GIS. [Online] 2011. <http://www.qgis.org/>.
- Bunge, Mario. 2009.** Diccionario de filosofía. México : s.n., 2009.
- BUSCHMANN, F. 2006.** Pattern – Oriented Software Architecture. A System of Pattern. Inglaterra : s.n., 2006.
- Cali. 2011.** [Online] 2011. . <http://www.cali.gov.co/planeacion/publicaciones.php?id=33568>. Glosario A-C de la Infraestructura de Datos Espaciales de Santiago de Cali..
- Capdevila, Joan. 2002.** Mapas y civilización. Historia de la cartografía en su contexto cultural y social. 2002.
- Carrillo, Isaías. 2008..** Metodología de Desarrollo de Software. 2008.
- CASE, Tools. 2011.** Case Tools. [Online] 2011. <http://case-tools.org..>
- Castro, Fidel. 1998.** Ministerio de Cultura de la República de Cuba. *Dirección de Seguridad y Protección.* [Online] Gaceta Oficial de la República de Cuba, Junio 14, 1998. [Cited: Diciembre 1, 2012.] <http://www.min.cult.cu/loader.php?sec=instituciones&cont=direccionseguridad>.
- CHACÓN, JULIO CÉSAR. 2006.** APLICACIÓN DE LA METODOLOGÍA RUP PARA EL DESARROLLO RÁPIDO DE APLICACIONES. Guatemala : s.n., 2006.
- CORPORATION. 2011.** Qt Creator IDE and tools. [Online] 2011. <http://qt.nokia.com/products/developer-tools..>
- Dangermond, Jack. 2008.** Esri española. [Online] 2008. <http://www.esri.es>.
- Dawson, John. 2007.** Osgeo. [Online] 2007. <http://www.osgeo.org/qgis>.
- G Buzai. 2011.** Análisis Socioespacial con Sistemas de Información Geográfica Tomo 1: Perspectiva Científica. Buenos Aires : s.n., 2011.
- Gamma, Erich. 2005.** Design Patterns - Elements of Reusable Object-Oriented Software. 2005.

- GARCERANT, I. 2008.** Modelo de Dominio. Tecnología y Synergix. . [Online] Visión de Synergix de los Sistemas de Información y la Ingeniería del Software, 2008. <http://synergix.wordpress.com/?s=modelo+de+dominio..>
- GONZÁLEZ, A. 2006.** Las nuevas tecnologías en la formación ocupacional: retos y posibilidades. Sevilla : s.n., 2006.
- GRACIA, J. . 2005.** Patrones de diseño. *Diseño de Software Orientado a Objetos*. [Online] . 2005. <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php..>
- INFORMÁTICA. 2006.** Secretaría de Planificación Estratégica. *Estándares para el uso de herramientas de desarrollo y plataformas de aplicaciones web*. Cuba : s.n., 2006.
- Jacobson, I. 2000.** El Proceso Unificado de Desarrollo de Software. 2000.
- JALÓN, Javier GARCÍA DE. 2008.** Aprende C++ como si estuviera en primero. Universidad de Navarra : s.n., 2008.
- Jiménez, A. Moreno. 2008.** Sistemas y análisis de la información geográfica. España : s.n., 2008.
- Lanero, Agustin. 2011.** 2011. Topografía, cartografía y cinecias afines.
- Larman, Craig. 1999.** *UML y patrones*. 1999.
- Longley, P. 2001.** Geographic Information Systems and Science. Reino Unido : s.n., 2001.
- Martín , Rafael . 2010.** Venezuela : EVOLUCIÓN DEL PENSAMIENTO CIENTÍFICO, 2010, Vol. PRIMERAS OBSERVACIONES DE LOS FENÓMENOS ELÉCTRICOS Y MAGNÉTICOS.
- Martinez, Carlos A. 2010.** s.l. : Definiendo los SIG, 2010.
- ORALLO, E. H. 2002.** El Lenguaje Unificado de Modelado (UML). 2002.
- PRESSMAN, Roger. 2005.** Ingeniería de Software. . [book auth.] Mc Graw Hill. *Un enfoque práctico*. 2005.
- RAE. 2012.** Real Academia Española. [Online] 2012. [Cited: Febrero 25, 2013.] <http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>.
- Rhind, David. 2005.** Consideraciones conceptuales de los sistemas de informacion geografica. Reino Unido : s.n., 2005.
- Rogerson, P. 2009.** Spatial anlysis and GIS. 2009.
- Sendra, J. Bosque. 2002.** Sistemas de Información Geográfica. Madrid : s.n., 2002.
- Silva, José Luis. 2011.** s.l. : Aplicación de Sistemas de Información Geográfica en Cuba, 2011. Revista Internacional de Ciencias de la Tierra.
- Terry, Steven. 2009.** GvSIG. [Online] 2009. <http://www.gvsig.org/web/>.

Torres, Jordi. 2009. *Edición y visualización de información vectorial en aplicaciones SIG*. San Sebastián : s.n., 2009.

Visconti, M. and Astudillo. 2011. *Fundamentos de Ingeniería de Software*. 2011.

Willian, G.D. 2008. Web coders in the bo. [Online] 2008. <http://www.codebox.es/glosario..>

GLOSARIO

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

CASE: (Computer Aided Software Engineering). Constituye una herramienta que ayuda al ingeniero de software a desarrollar y mantener software.

Compilador: Un compilador es un programa que permite traducir el código fuente de un programa en lenguaje de alto nivel, a otro lenguaje de nivel inferior

Datos Espaciales: “Los datos geográficos se definen como cualquier información sobre objetos o fenómenos que tengan una ubicación relativa con respecto a la superficie de la Tierra”.

Georreferenciación: La georreferenciación para un determinado sistema de coordenadas no es más que la posición con la que se define la localización de un objeto espacial, (esto puede ser a través de vectores, puntos, aéreas).

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

Paradigma: Procede del griego *paradeigma*, que significa “ejemplo” o “modelo”. En principio, se aplicaba a la gramática (para definir su uso en cierto contexto) y a la retórica (para referirse a una parábola o fábula). A partir de la década del '60, comenzó a utilizarse para definir a un modelo o patrón en cualquier disciplina científica o contexto epistemológico.

Plugins: Un software plug-in es un complemento para un programa que añade funcionalidad a la misma.

Software: “Producto que los ingenieros de software construyen y después mantienen en el largo plazo. Incluyen los programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, el contenido que se presenta conforme los programas se ejecuten y los documentos, tanto físicos como virtuales, que engloban todas las formas de medios electrónicos” (PRESSMAN, 2005).

Software libre: Es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente.