

Universidad de las Ciencias Informáticas

Facultad 6



**Sistema de Información Geográfica para la  
Dirección de Seguridad y Protección en la  
Universidad de las Ciencias Informáticas**

Trabajo de Diploma para optar por el Título de  
Ingeniero en Ciencias Informáticas

Autores: Yainelis Gallardo Hidalgo  
Miguel Milán Isaac

Tutor: Ing. Nilberto C. Chavez Marquez

La Habana, Junio/2013

*Declaración de Auditoría*

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2013.

Autor(es):

---

Yainelis Gallardo Hidalgo

---

Miguel Milán Isaac

Tutor:

---

Ing. Nilberto C. Chavez Marquez

## *Datos del Contacto*

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

Profesión: Ingeniero en Ciencias Informáticas

Categoría docente: Asistente.

Año de graduado: 2007

Correo electrónico: nchavez@uci.cu

Profesor destacado por su preparación docente metodológica ha impartido las asignaturas de Práctica Profesional I y III, Ética Informática, Historia de la Informática, Formación Pedagógica, Metodología de la Investigación Científica, Debate Histórico Contemporáneo. Líder de proyecto de las líneas de Geofísica y Control de Flota del Departamento de Geoinformática del Centro GEySED. Tutor de diez trabajos de investigaciones aplicadas los cuales han obtenido la máxima calificación. Fue subdirector de formación del centro de desarrollo. Analista principal del proyecto SiG-Desktop.

## *Dedicatoria*

### *Dedicatoria de Miguel*

*A mi mamá por estar conmigo durante todo este tiempo y brindarme todo el apoyo que necesité para lograr este y otros resultados en la vida.....*

*A mi hijo porque ha sido la razón por la cual me he sacrificado durante todo este tiempo, cosa que me enseñó mi mamá....luchar por los hijos.....*

*A mi novia por soportarnos mutuamente durante estos tres años y darme el apoyo que necesité aquí en la escuela.....*

*A toda mi familia por apoyarme en mi formación y educación.....*

*A mis amigos por soportarme, y por siempre tener un tema para formar una discusión.....*

*A todos muchísimas gracias....*

### *Dedicatoria de Yainelis*

*A mi mamá y a mi papá por brindarme todo el apoyo que necesité durante estos cinco años, por el cariño que siempre me han dado, por confiar en mí y creer que podía llegar aquí a pesar de todo....*

*A mi hermana por darme apoyo, por escucharme y por estar siempre en las buenas y en las malas...*

*A mi novio por ser paciente durante este tiempo, por ayudarme, apoyarme y comprenderme....*

*A mis amigos, por de una forma u otra y por cualquiera de las vías existentes darme el ánimo que necesité, preocuparse por mis cosas y estar ahí siempre....*

*A todos muchísimas gracias....*

## *Agradecimientos*

### *Agradecimientos de Miguel*

*A mi mamá por preocuparse por mí, por ser además de madre, un padre para mí y demostrarme que todo puede ser posible, por confiar en mí....*

*A Frank por ayudarme durante todos estos años y que siempre se ha preocupado por mis cosas.....*

*A mi hijo porque me dio las fuerzas que necesité en los momentos más difíciles para seguir adelante....*

*A mi novia por ayudarme, preocuparse por mí y por mis cosas.....*

*A mis tías, mis tíos, mi abuela, mi tío Abelito... que siempre me han querido y se han preocupado por mí.....*

*A mis primas y primos que han sido los hermanos que no he tenido.....*

*A mi amigo Nestor por apoyarme de forma incondicional durante estos cinco años....*

*A mi tutor por aguantar esta carga tan pesada, ayudarme a controlar un poco mis emociones y obtener este resultado.....*

*Al jefe de proyecto todo lo que me enseñó de GeoQ y Qt, y ayudarme cuando explotaba algo en el código que muchas veces decía que “en su máquina no pasaba”, pero al final encontrábamos una solución.....*

*A Sandor que nos brindó todo el conocimiento que tiene sobre GeoQ y nos apoyaba con el código....*

*A mi tribunal por todos sus consejos para que este trabajo saliera con la calidad requerida ....*

*A todos los profesores del proyecto y de curso que me aportaron parte de su conocimiento y permitieron que estuviera hoy aquí..*

*A todos muchas gracias.....*

### *Agradecimientos de Yainelis*

*A mi mamá por cada vez que me sentía mal hacerme sentir un poco mejor, por estar siempre cuando la necesité. Por hacerme esa comida que me duraba toda la semana. Por saber cómo tratarme cuando tenía los días malos....*

*A mi papá por ayudarme con las conclusiones, darme apoyo, por ser el mensajero cuando había que traerme la comida y por enseñarme las cosas que fuera de lo común debe saber una mujer....*

*A mi hermana por darme consejos, por permitirme dárselo también, por escucharme cuando necesité que alguien me escuchara, por estar siempre en las malas y en las buenas.....*

*A mis tíos, por preocuparse por mí y saber el cómo me iban las cosas.....*

*A mi novio por ser tan paciente, por darme cariño, apoyo. Por aguantarme o al menos intentarlo cuando yo estaba imposible, por comprenderme, por permitirme ser yo misma.....*

*A Audrey, Yankiel, Alaín y Yarita por ser los amigos con los cuales he compartido una parte de mi vida, que puedo contar con ellos para los buenos y malos momentos, tienden a ser un poco falsos pero están ahí.....*

*A Margelis por demostrar ser una persona que valora mucho la amistad y a pesar de no vernos hace casi dos años o más, siempre se ha mantenido al tanto de mis cosas, preocupándose por mí (siendo mi hermanita color chocolate).....*

*A mi tutor por ayudarme en todo lo que estaba a su alcance, por darme la energía que necesité para seguir adelante, por ser más que mi tutor, mi amigo....*

*Al Peto, al Lachy, a Robe, a Eduard, a Frank, a Alexis, a Comikito, por estar siempre atentos, pendientes de cada uno de mis pasos. Dándome chuchos con cada uno de los mundiales. A Pedro por demostrarme de una forma peculiar que estaba ahí para lo que necesitara (un pisotón en el pie).....*

*A las chicas del apartamento, incluyendo a Humbe, por quitarme comida, por despertarme cuando estaba durmiendo, por soportar mis malos genios, por comprenderme y apoyarme. Por estar en los buenos y malos momentos durante estos cinco años.....*

*A Yeni por ser mi profesora y su acompañante durante estos dos últimos años....*

*A mi tribunal por decirme que estábamos embarcados a cada rato, pero que nos permitió con sus críticas que este trabajo quedara con la calidad requerida....*

*A mis amistades y personas que he conocido aquí, que a muchas no voy a volver a ver nunca, pero que van a estar en un rinconcito de mi corazón.*

*A todos los profesores que me dieron el conocimiento para ser hoy por hoy quien soy....*

*“Todo aquel que está seriamente comprometido con el cultivo de la ciencia, llega a convencerse de que en todas las leyes del universo está de manifiesto un espíritu infinitamente superior al hombre, y ante el cual, nosotros con nuestros poderes debemos sentirnos humildes.”*

*Albert Einstein*

## *Resumen*

Con el surgimiento de las empresas e instituciones y el continuo desarrollo de las Tecnologías de la Información y las Comunicaciones, aparecieron nuevas interrogantes y soluciones con el fin de satisfacer las necesidades de protección y de seguridad en las mismas. Implantando el uso de sistemas informáticos, a partir del uso de cámaras de seguridad, Sistemas de Posicionamiento Global y Sistemas de Información Geográficos. La Universidad de las Ciencias Informáticas, es una de estas instituciones que necesita proteger los recursos humanos y materiales que posee, para ello cuenta con la Dirección de Seguridad y Protección, encargada de velar por la seguridad de la institución, así como de la representación de la planificación y el control de la Guardia Obrera – Estudiantil sobre mapas. Las representaciones que se realizan actualmente, muchas veces traen consigo errores en cuanto a la representación espacial de los activos asociados al servicio de guardia, así como del control de la asistencia durante el servicio de guardia, por la realización manual de dicha representación sobre mapas en formato impreso.

El objetivo del presente trabajo de diploma persigue desarrollar un Sistema de Información Geográfica que permita representar espacialmente la información de los procesos de planificación y control de la Guardia Obrera-Estudiantil en la Dirección de Seguridad y Protección de la Universidad de las Ciencias Informáticas. La presente investigación identifica y analiza integralmente la situación problemática, además de materializar los resultados de investigaciones anteriores. Se realiza un análisis de las diferentes herramientas y tecnologías actuales para el modelado y desarrollo del sistema propuesto. Además se describe y modela la solución propuesta.

**Palabras Claves:** Activos del sistema de guardia, Dirección de Seguridad y Protección, Sistema de Información Geográfica, Tecnologías de la Información y las Comunicaciones.



## *Abstract*

With the emergence of enterprises and institutions and the continued development of the Information Technology and Communications, were new questions and solutions to meet the needs of protection and security in them. Implementing the use of computer systems, from the use of security cameras, Global Positioning Systems and Geographic Information Systems. The University of Informatics Sciences, is one such institution that needs to protect human and material resources it has, for it has the Directorate of Security and Protection, responsible for ensuring the safety of the institution, representation as well as the planning and control of the Guard Workers - Student maps. The representations that are made today, often bring with errors as to the spatial representation of the assets associated call service and assistance control for guard duty, by manually performing such representation on maps print.

The purpose of this diploma work aims to develop a Geographic Information Systems that allows information spatially represent the planning and control of the Directorate of Security and Protection Guard Workers - Student at University of Informatics Sciences. This research identifies and analyzes comprehensively the problem situation, in addition to realizing the results of previous research. An analysis of the different tools and technologies for modeling and development of the proposed system. It also describes and models the proposed solution.

**Keywords:** Directorate of Security and Protection, Geographic Information Systems, guard system assets, Information Technology and Communications.

## Índice

Introducción .....	1
Capítulo I: Fundamentación teórica de la solución propuesta. ....	6
1.1 Conceptos asociados al dominio del problema .....	6
1.2 Objeto de estudio.....	8
1.2.1 Descripción general.....	8
1.2.2 Descripción actual del dominio del problema.....	8
1.2.3 Situación problemática .....	9
1.3 Análisis de otras soluciones existentes.....	10
1.3.1 TURNex (Planificación y Control de Turnos de Trabajo) .....	10
1.3.2 SiCaTu (Sistema de Cálculo de Turnos) .....	11
1.3.3 SP-Expert (Software de Gestión y Planificación Inteligente del Personal) .....	12
1.3.4 Valoración general .....	13
1.4 Conclusiones del capítulo .....	13
Capítulo II: Tendencias y tecnologías actuales a utilizar .....	14
2.1 GeoQ.....	14
2.2 Metodologías de desarrollo de software .....	14
2.2.1 Proceso Unificado de Desarrollo (RUP(Rational Unified Process)) .....	15
2.2.2 ¿Por qué RUP?.....	15
2.3 Lenguaje de modelado .....	16
2.3.1 Lenguaje Unificado del Modelado (UML).....	16
2.3.2 ¿Por qué lenguaje UML? .....	16
2.4 Herramientas CASE .....	17
2.4.1 Visual Paradigm .....	17
2.4.2 ¿Por qué Visual Paradigm?.....	17
2.5 Lenguaje de programación .....	18
2.5.1 C++ .....	18
2.5.2 ¿Por qué C++?.....	18
2.6 Entorno de Desarrollo Integrado (IDE).....	19

2.6.1 QtCreator .....	19
2.6.2 ¿Por qué QtCreator? .....	19
2.7 Gestores de bases de datos .....	20
2.7.1 PostgreSQL .....	20
2.7.2 ¿Por qué PostgreSQL? .....	20
2.8 Herramientas de administración .....	21
2.8.1 PgAdmin .....	21
2.8.2 ¿Por qué pgAdmin? .....	22
2.9 Conclusiones del capítulo .....	22
Capítulo III: Presentación de la solución propuesta .....	23
3.1 Modelo del negocio.....	23
3.1.1 Actores y trabajadores del negocio .....	23
3.1.2 Diagrama de actividades.....	26
3.1.3 Descripción de los casos de uso del negocio .....	27
3.2 Requisitos.....	29
3.2.1 Requisitos funcionales .....	29
3.2.2 Requisitos no funcionales.....	31
3.3 Descripción del sistema propuesto .....	34
3.3.1 Descripción de los actores .....	34
3.3.2 Diagrama de casos de uso del sistema.....	34
3.3.3 Descripción de casos de uso del sistema.....	35
3.4 Conclusiones del capítulo .....	39
Capítulo IV: Construcción de la solución propuesta.....	41
4.1 Análisis .....	41
4.1.1 Modelo del análisis.....	41
4.1.2 Diagrama de clases del análisis .....	42
4.1.3 Diagrama de interacción .....	43
4.1.3.1 Diagrama de colaboración .....	43
4.1.3.2 Diagrama de secuencia .....	44
4.2 Diseño .....	44

4.2.1 Patrones.....	44
4.2.1.1 Patrones arquitectónicos .....	45
4.2.1.2 Patrones de diseño .....	45
4.2.1.3 Patrones GRASP (Patrón de Asignación de Responsabilidades) .....	46
4.2.2 Diagrama de clases del diseño .....	46
4.2.3 Diseño de la base de datos .....	48
4.2.3.1 Diagrama de clases persistentes .....	49
4.2.3.2 Modelo Entidad-Relación .....	50
4.3 Implementación .....	50
4.3.1 Modelo de Despliegue.....	50
4.3.2 Diagrama de componentes .....	51
4.4 Prueba.....	52
4.4.1 Pruebas de caja negra .....	52
4.5 Resultado de las pruebas .....	60
4.6 Conclusiones del capítulo .....	60
Conclusiones generales .....	62
Recomendaciones .....	63
Referencias bibliográficas.....	64

### *Introducción*

Las Tecnologías de la Información y las Comunicaciones (TICs), surgieron con el objetivo de informatizar muchos de los procesos que se estaban llevando a cabo en la sociedad, así como gestionar la información, permitiendo almacenarla y procesarla. Muchos de estos procesos estaban estrechamente relacionados con las necesidades que poseía la sociedad de garantizar la seguridad de los recursos humanos y materiales. Por ello y gracias al continuo desarrollo de las TICs se ha logrado informatizar muchos de los procesos que tienen lugar en la actualidad, garantizando el acceso a información de todo tipo. El desarrollo de los diferentes canales de comunicación y el almacenamiento de grandes volúmenes de datos, ha permitido gestionar las fuentes de información, posibilitando mayor precisión, evitando así posibles errores en el trabajo manual.

Para las empresas e instituciones que manejan gran cantidad de recursos humanos y materiales, garantizar la seguridad de los mismos se convierte en una ardua tarea, en la cual se invierten cuantiosos recursos, en el uso de nuevas tecnologías con estos fines. Con las TICs surgieron nuevas soluciones, con el fin de satisfacer las necesidades elementales de las instituciones, en aras de preservar los recursos humanos y materiales que posee, estableciendo el uso de sistemas para la seguridad y protección. Muchos de estos sistemas de seguridad tienen implantado cámaras de seguridad, Sistema de Posicionamiento Global (GPS) y Sistema de Información Geográfica (SIG), siendo capaces de visualizar con exactitud y en tiempo real las operaciones de planificación y control de la guardia, entre otras.

La Universidad de las Ciencias Informática (UCI) es una de estas instituciones que necesita garantizar la seguridad y protección de los recursos humanos y materiales que posee. Para ello cuenta con la Dirección de Seguridad y Protección (DSP) la cual se encarga de velar por la seguridad de la institución, así como de la representación de la planificación y control de la guardia obrera-estudiantil en el centro, implementando las políticas a seguir con estos fines. En la actualidad se realiza la planificación y control de la Guardia Obrera Estudiantil (GOE), en aras de prevenir cualquier hecho delictivo que suceda. La misma se realiza de forma manual, ya que la universidad no cuenta con una herramienta que permita representar espacialmente la información asociada a la planificación y control de la guardia obrera-estudiantil.

La DSP de la UCI se encuentra enfrascada en la implementación de un sistema de guardia integral para la institución que permita detener los diferentes hechos delictivos que se dan en la misma. 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 la representación espacial de los activos asociados a los procesos que se llevan a cabo en el sistema de guardia obrera-estudiantil.

Las representaciones que se realizan actualmente, muchas veces traen consigo errores que conllevan a imprecisiones en la toma de decisiones, tras ubicar sobre mapas en formato impreso las áreas existentes en la universidad y las postas por cada una de estas áreas. Por la poca visualización y por errores que se cometen en la planificación, en ocasiones la distribución del personal y la ubicación de las postas no es la más efectiva, ya que se encuentran unas frente a otras e incidiendo en el mismo radio de acción. En dichas áreas se encuentran zonas que son consideradas vulnerables por la DSP, las mismas no se encuentran representadas correcta y oportunamente en el sistema de guardia.

Esta representación errónea dificulta la ubicación de los puestos de mandos más cercanos cuando se produzca un incidente durante el servicio de guardia. De la misma forma trae problemas a la hora de distribuir los recursos humanos asociados al servicio de guardia, ya que no cumplen en muchas de las ocasiones con algunas de las premisas a llevar a cabo para este proceso, ubicando a personal del sexo femenino en zonas de altas incidencias de hechos delictivos, poniendo en riesgo la integridad de esas personas.

En ocasiones no se tiene en cuenta la cantidad de personas necesarias en una posta determinada, siendo la zona por su ubicación considerada de alto riesgo. Todo lo anteriormente planteado dificulta consultar la información ya que la representación de la misma se realiza de manera manual sobre mapas impresos generados por otras aplicaciones. Al personal implicado en la planificación de la guardia se le dificulta la ubicación sobre el terreno de los diferentes activos existentes en el sistema de guardia.

El oficial de guardia es el encargado de llevar a cabo el control de la asistencia en el servicio de guardia y conocer el estado en el que se encuentra una posta en cada turno, evitando que en alguna posta falte uno o todo el personal implicado, estando propensa a la ocurrencia de hechos delictivos. En ocasiones, el control de la asistencia se hace tedioso ya que no se tiene una visualización previa de los estudiantes y profesores que se encuentran en las diferentes postas que componen el área, ni de los ausentes para la realización de una nueva reestructuración de la planificación para impedir la ocurrencia de cualquier situación inadecuada a lo que está establecido.

Todo lo anteriormente planteado, trae un alto grado de errores en cuanto a la representación espacial de los activos del sistema de guardia en la Dirección de Seguridad y Protección de la Universidad de las Ciencias Informáticas, así como del control del estado de la asistencia en un turno de guardia, dificultándose la toma de decisiones, dado la realización manual de esta representación.

Tras el análisis de la situación existente se identifica el siguiente **problema a resolver** ¿Cómo representar y ubicar espacialmente la información asociada al proceso de planificación y control del sistema de guardia obrera-estudiantil, en la Universidad de las Ciencias Informáticas? Para dar solución al problema anteriormente planteado se propone como **objetivo general**, desarrollar un Sistema de Información Geográfica, que permita representar espacialmente la información de los procesos de planificación y control de la guardia obrera-estudiantil en la Dirección de Seguridad y Protección de la Universidad de Ciencias Informáticas. Identificándose como **objeto de estudio** en la presente investigación los procesos de planificación y control del sistema de guardia obrera – estudiantil en la Universidad de las Ciencias Informáticas, teniendo como **campo de acción** la representación espacial de la información asociada al sistema de guardia obrera-estudiantil de la Universidad de las Ciencias Informáticas.

En la presente investigación se propone la siguiente **idea a defender**, si se implementa un Sistema de Información Geográfica que permita representar espacialmente la información de los procesos de planificación y control de la guardia obrera-estudiantil, se agilizará el proceso de toma de decisiones en la Dirección de Seguridad y Protección de la Universidad de las Ciencias Informáticas.

Para dar solución al problema planteado se proponen las siguientes tareas de la investigación:

- ✚ Caracterizar los procesos del sistema de guardia obrera-estudiantil en la Universidad de las Ciencias Informáticas.
- ✚ Realizar una revisión de las tendencias y tecnologías actuales aplicables al proceso del negocio.
- ✚ Modelar la solución propuesta.
- ✚ Implementar la solución propuesta.
- ✚ Probar la solución propuesta.

Como resultado del cumplimiento de las tareas investigativas se espera obtener:

- ✚ La documentación técnica del proceso de desarrollo de software.
- ✚ El Sistema de Información Geográfica para la Seguridad y Protección en la Universidad de las Ciencias Informáticas (GeoQ-Guardián).

Para el desarrollo de esta investigación se utilizan los siguientes métodos:

### **Métodos Teóricos:**

**Analítico – sintético:** Se utiliza para estudiar y analizar la documentación y bibliografías de diferentes autores y extraer la información que permita sustentar desde el punto de vista teórico y práctico los elementos que se relacionan con el proceso de planificación del sistema de guardia obrera – estudiantil en la Universidad de las Ciencias Informáticas.

**Análisis histórico-lógico:** Se utiliza con el objetivo de realizar un estudio sobre la evolución y el desarrollo que han tenido los Sistemas de Información Geográfica, lo que permite 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. Analizar el proceso de planificación y control de los recursos humanos y materiales de los sistemas de seguridad y protección y realizar una valoración sobre el estado del arte del objeto de estudio para realizar la proyección futura en el campo de acción escogido.

**Modelación:** La modelación es justamente el proceso mediante el cual se crean modelos con vista a investigar 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.

### **Métodos Empíricos:**

**Observación:** Permite realizar valoraciones y obtener información a partir de lo observado. Esto se utilizará principalmente cuando se realicen observaciones sobre el funcionamiento de Sistemas de Información Geográfico con estos fines, lo que dará una visión de cómo tiene que ser el sistema.

**Entrevista:** Permite establecer el alcance que tendrá el presente trabajo, el objetivo, además del flujo actual de la información, así como la comunicación con otras personas, para obtener los requerimientos que contendrá la aplicación a desarrollar.

La investigación está dividida en cuatro capítulos fundamentales.

**Capítulo 1:** 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 la representación espacial de la planificación de la guardia, así como del control de la asistencia, además de realizar un análisis de otras situaciones existentes.



**Capítulo 2:** Se argumenta el porqué de la selección de las tecnologías y herramientas a utilizar en el proceso de desarrollo del software, lenguajes de programación y modelado, así como la arquitectura a utilizar, la metodología y el entorno de desarrollo.

**Capítulo 3:** En este capítulo se da a conocer la solución propuesta con todos los argumentos, el modelo del negocio, los requerimientos funcionales y no funcionales, así como la descripción de los casos de uso, los actores del negocio y del sistema.

**Capítulo 4:** En este capítulo se plasmarán los patrones a utilizar, los diagramas de clases del análisis y los diagramas de clases del diseño, así como los diagramas de componentes, el diagrama de despliegue y el diseño de casos de prueba de la solución.

*Capítulo I: Fundamentación teórica de la solución propuesta.*

En el presente capítulo se describirán los conceptos asociados al dominio del problema, así como la descripción del objeto de estudio. Se analizarán los avances tecnológicos asociados a los sistemas existentes, así como, la descripción actual del dominio del problema para dar solución a la situación problemática en la Dirección de Seguridad y Protección de la Universidad de las Ciencias Informáticas.

*1.1 Conceptos asociados al dominio del problema*

A continuación se plantean un conjunto de definiciones, las cuales están estrechamente relacionadas con el objeto de estudio y la problemática tratada en la investigación, para que sea de fácil comprensión y entendimiento.

**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)

**Sistema**

Conjunto de partes o elementos organizados y relacionadas que interactúan entre sí para lograr un objetivo. (Cuesta, 2008)

**Guardia**

Personal especializado en asegurar la defensa de una o varias personas o un puesto determinado.(RAE, 2012)

**Sistema de guardia**

Los autores de la presente investigación consideran 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**

Los autores de la presente investigación consideran 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)

### **Zona vulnerable**

Los autores de la presente investigación consideran 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.

### **Sistema de Información Geográfica**

El término SIG procede del acrónimo de Sistema de Información Geográfica. Técnicamente se puede definir un SIG como una tecnología de manejo de información geográfica formada por equipos electrónicos (hardware) programados adecuadamente (software) que permiten manejar una serie de datos espaciales

(información geográfica) y realizar análisis complejos con éstos, siguiendo los criterios impuestos por el equipo científico (personal). (Sanabria, 2011)

### *1.2 Objeto de estudio*

#### *1.2.1 Descripción general*

El proceso de planificación y control de la guardia obrera-estudiantil es realizado por el Vicedecano de Residencia o Extensión Universitaria y Administradores de Áreas respectivamente, encargados de cada una de las áreas existentes en la universidad. Debido a la cantidad de recursos humanos y materiales con los cuales cuenta la institución, la realización de la planificación de la GOE en ocasiones se hace tediosa, ya que las mismas son realizadas sobre ficheros Excel<sup>1</sup> y sin una previa visualización de las información de los diferentes activos sobre mapas, para la realización de valoraciones sobre la planificación y control de la guardia que se está llevando a cabo. Por lo general y por la poca visualización que se tiene de los activos existentes en la institución, conlleva a un alto grado de errores en la representación espacial de la información asociada a los procesos de planificación y del control de la guardia obrera-estudiantil e inconformidades por parte del personal implicado.

El objetivo de la representación espacial del proceso de planificación y control de la guardia, es asegurar que cada una de las postas, áreas, puestos de mando, la información de la planificación, recursos humanos y zonas vulnerables con las cuales cuenta la universidad se encuentren bajo constante vigilancia en los horarios establecidos por la DSP.

#### *1.2.2 Descripción actual del dominio del problema*

La DSP de la UCI realiza la GOE, que unidades constituyen el eslabón fundamental en la protección interna de la universidad. La GOE tiene como objetivo fundamental, la protección y cuidado de las personas y medios materiales con los que cuenta la institución, de conjunto con los demás servicios contratados por la universidad de Seguridad y Protección. Para ello la DSP elaboró un grupo de principios que tienen como objetivo fundamental la correcta planificación, así como el control de la GOE en la institución, estableciendo las áreas consideradas de mayor riesgo en la universidad.

---

<sup>1</sup> Aplicación distribuida por Microsoft Office para hojas de cálculo.

La realización de la planificación de la GOE cuenta con un grupo de lineamientos establecidos por la DSP, lo que hace posible que la guardia se realice en el tiempo requerido y se efectúe fuera del horario laboral, basándose en la Ley No 75 de la Defensa Nacional para la realización de la guardia obrera-estudiantil. El personal que realiza la guardia tendrá un previo aviso (con tres meses de antelación) del lugar, día y turno donde realizará dicha tarea; por parte del jefe encargado, cuáles son los implicados en cada una de las postas existentes y el horario en el cual lo efectuará.

Las áreas definidas por la DSP de la UCI, cuentan con un responsable, el cual es el encargado de la planificación y control de la guardia obrera-estudiantil, así como de los cambios que se realicen en la misma, ante cualquier problema que pueda existir, haciéndose necesario el cambio de día o turno de guardia al cual está dirigido. Una vez conocido las postas, los estudiantes, profesores y trabajadores se presentan en sus respectivos puestos de mando o donde se encuentran los jefes de equipos (turnos), diez minutos antes del horario establecido.

El cumplimiento de la GOE se registra en el libro de incidencias, el cual es controlado por los responsables de cada una de las áreas con las cuales cuenta la universidad. Esto permite controlar la asistencia al servicio de guardia, con el objetivo de conocer los estudiantes y profesores que se encuentran ausentes, así como para su posterior análisis.

### *1.2.3 Situación problemática*

La DSP de la UCI se ve en la necesidad de llevar a cabo la implementación de un servicio de guardia integral para la institución, el cual permita detener los diferentes hechos delictivos que puedan ocurrir en la misma. Para ello han diseñado un sistema de guardia que involucre todas las áreas en aras de hacerlo efectivo. Dada la cantidad de áreas, recursos humanos y materiales con que cuenta la institución se dificulta la representación espacial de los activos asociados a los procesos que se llevan a cabo en el sistema de GOE.

En estos momentos la representación de los activos y de la planificación del sistema de guardia, requiere de gran esfuerzo para su realización, debido a la cantidad de recursos humanos y materiales, postas y áreas a cubrir en cada período. De igual manera se dificulta llevar a cabo el control de la asistencia en el servicio de guardia, así como conocer el estado en el que se encuentra una posta determinada, debido a que no se tiene una visualización previa de los estudiantes que se encuentran en las diferentes postas, ni

de los ausentes existentes para la realización de una nueva reestructuración de la planificación, en caso de que ocurra que una posta quede total o parcialmente vulnerable.

Es de vital importancia para la institución tener representadas y bien protegidas aquellas zonas que representan o están en riesgo potencial, las cuales fueron definidas por la DSP como zonas vulnerables. En la actualidad no es así, lo que trae consigo errores en la distribución de los recursos humanos en las diferentes postas establecidas. Muchos de estos errores están dados por la incorrecta ubicación geográfica de dichas zonas y las dificultades para localizarlas sobre los mapas que se elaboran en grandes láminas de papel.

El sistema de postas existentes y su respectiva distribución por áreas, no permiten sobre el terreno tener una visión general de los recorridos de los diferentes servicios de guardia, así como el área geográfica que cubre cada posta, lo que trae consigo que no se realicen planificaciones acorde a las necesidades de la institución y que queden lugares más protegidos que otros.

Hasta este momento la información referente al funcionamiento del servicio de guardia en un período determinado no se conoce y debe analizarse de manera diferida, por lo que muchas veces se pierde información o se hacen análisis poco efectivos que van en detrimento de la labor educativa que debe realizarse en cada una de las áreas con los ausentes, que llegan tarde o cometen alguna irregularidad durante el servicio de guardia. De igual manera no se conservan ninguno de los reportes asociados a los históricos del sistema de guardia. Los rectores en funciones no tienen conocimiento en tiempo real de cuál es el estado del sistema de guardia en la universidad, ni de cuál es el área más vulnerable durante su servicio de guardia.

### *1.3 Análisis de otras soluciones existentes*

De las principales soluciones existentes a nivel mundial, los autores del presente trabajo de diploma analizaron las funcionalidades y características que pudieran tenerse en cuenta en el desarrollo de la propuesta de solución.

#### *1.3.1 TURNex (Planificación y Control de Turnos de Trabajo)*

TURNex es un producto diseñado por Solex (filiar de Sonda), este está especialmente diseñado para programar y planificar los turnos de los empleados en forma fácil y eficiente. Su foco se centra en resolver complejas asignaciones y cambios en los respectivos turnos. El mismo está orientado a todas aquellas

empresas que prestan servicios de personal a sus clientes y que trabajan en base a turnos, como por ejemplo: guardias de seguridad, empresas de aseo, cajeros, entre otros.

Permite reducir los tiempos de reprogramación de turnos, mejorar la utilización del personal, beneficiar la atención y satisfacción de los clientes, e incrementar la utilidad de la empresa. TURNex en su diseño no presenta un componente que sea capaz de representar espacialmente los procesos de planificación y control de los turno de guardia, impidiendo conocer con exactitud la ubicación de dicho procesos.(Solex, 2012)

Dentro de sus principales características, se destacan las siguientes:

- ✚ Producto 100% Web.
- ✚ Soporta diferentes plataformas.
- ✚ Es escalable y modular.
- ✚ Garantiza servicios profesionales de soporte y mejora continua ante cualquier problema.
- ✚ Permite definir múltiples: zonas, sucursales, clientes e instalaciones.
- ✚ Perfil de Usuario: operadores, supervisores, planificadores.
- ✚ Conceptos estándar de control de asistencia: empleados, calendarios, horas contratadas, bonificaciones, hojas de vida, justificaciones de ausencias.
- ✚ Ciclo de vida de turnos: patrones, rotaciones, asignaciones, validaciones.
- ✚ Integración de los recursos humanos: Políticas de remuneraciones.

### 1.3.2 SiCaTu (*Sistema de Cálculo de Turnos*)

SiCaTu se desarrolla para dar solución a la planificación de las guardias de un Departamento de Cardiología. Con el objetivo de buscar una solución para visualizar el proceso de asignación de guardias de los empleados. Dicho proceso consiste en varios tipos de guardias que incluyen horas extras, fines de semana, puentes, entre otros activos.(Molina, 2012)

El mismo utiliza un algoritmo matemático de conjunto con las premisas que se especifican, se confecciona una planificación que se adapta en un 100% a las necesidades del departamento. La idea de SiCaTu es confeccionar una interfaz de usuario sencilla, de fácil manipulación, que detallara en todo lo posible todos los elementos implicados y presentara la solución en forma de informe:

- ✚ Personal.

- ✚ Vacaciones de cada persona.
- ✚ Días festivos y puentes.
- ✚ Tipos de guardias.
- ✚ Posible pre-asignaciones (guardias impuestas antes del cálculo).
- ✚ Calendarios calculados.
- ✚ Distintos tipos de informes.

### 1.3.3 SP-Expert (*Software de Gestión y Planificación Inteligente del Personal*)

SP-Expert, permite un enfoque proactivo para la gestión del tiempo de trabajo óptimo, generando los modelos de tiempo de trabajo previos a la planificación, que posteriormente se aplican optimizando los turnos y el calendario de trabajo. La contabilización del tiempo de trabajo anual se puede gestionar en base a cuentas positivas o negativas y subdivididas en diferentes intervalos durante el año (mensual, trimestral, entre otros). El alcance de la previsión puede ser ampliado o reducido según se desee. Por tanto, siempre es posible ver los efectos a corto plazo de las decisiones de planificación sobre el saldo de horas a final de año.

El gestor de planificación puede trabajar con modelos flexibles de tiempo de trabajo basándose en los límites mínimo y máximo de horas de trabajo, con parámetros que son definibles por el usuario. Las funciones de contabilización online del tiempo de trabajo están completamente integradas en el software de planificación y muestran de inmediato los efectos del programa previsto en las cuentas de tiempo de los empleados (saldos de horas). SP-Expert pone en evidencia, con antelación, que las costosas horas extraordinarias deberían ser aprovechadas para producir. (Interflex, 2009)

Esta solución cuenta con un grupo de funcionalidades entre las cuales se encuentra:

- ✚ Vista de la Planificación específica para cada usuario según agrupación y clasificación.
- ✚ Control de cuentas online (vacaciones, horas contrato, entre otros).
- ✚ Necesidades operativas de personal.
- ✚ Selección de turnos.
- ✚ Vistas personalizables de la planilla y de las cuentas (horas de contrato, vacaciones, entre otros).
- ✚ Espacio para observaciones.
- ✚ Horas de inicio y final de turno.



- ✚ Selecciones especiales (tiempos bonificados).
- ✚ Comprobación de reglas de contrato y/o convenio.

### 1.3.4 Valoración general

Luego de analizar las soluciones anteriormente propuestas, los autores de la presente investigación consideran que las mismas poseen características similares a lo que se desea implementar, pero ninguna de estas se adecua a las características de los procesos que se desarrollan en la DSP. Estos sistemas son desarrollados en un entorno Web y no cuentan con componentes SIG que le permitan geo-referenciar la información que se maneja en la DSP de la institución. Además, todas estas soluciones pertenecen a empresas y no permiten la modificación de las funcionalidades presentes, lo que impediría adecuarlas a las necesidades existentes.

### 1.4 Conclusiones del capítulo

Durante el desarrollo de este capítulo se realizó un análisis del objeto de estudio y de la situación problemática, así como de los principales sistemas de planificación y control de la guardia existente a nivel mundial, 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 para la ubicación geográfica de los activos, la no determinación del estado de la asistencia al servicio de guardia y la carencia de bases históricas para el conocimiento de cómo es que se realiza la guardia en dicha institución.
- ✚ Los sistemas de planificación y control de la guardia estudiados, no utilizan un componente para la representación espacial de los diferentes activos con los cuales cuenta la institución, así como de la información asociada al proceso de planificación, por lo que no cumple con las necesidades del cliente; no obstante, cuentan con funcionalidades que pueden ser utilizadas como apoyo para dar solución a la problemática existente.

### *Capítulo II: Tendencias y tecnologías actuales a utilizar*

En este capítulo se realizará un análisis sobre las principales metodologías y herramientas a utilizar en la solución propuesta. Para ello se llevará a cabo una breve descripción de algunas de las principales arquitecturas de software, metodologías de desarrollo de software, lenguajes de modelado, herramientas CASE, lenguajes de programación, entornos de desarrollo integrado, gestores de bases de datos y herramientas de administración, con el objetivo de establecer una comparación entre ellas; seleccionando las que han de utilizarse para el desarrollo de la solución propuesta.

#### *2.1 GeoQ*

GeoQ es una plataforma del Centro de Desarrollo de Software GEySED de la Universidad de las Ciencias Informáticas, basado en Quantum GIS (QGIS), es de código abierto y se encuentra bajo la Licencia Pública General (GNU/GPL). Es una solución desarrollada en C++<sup>2</sup> y de carácter multiplataforma (Linux, Unix, Mac OSX y Windows) que trabaja con distintos orígenes de datos tales como PostGIS y SpatiaLite, utiliza los formatos ráster y vectorial. GeoQ proporciona un número cada vez mayor de las capacidades proporcionadas por las funciones básicas y plugins<sup>3</sup>. Permite visualizar, gestionar, editar, analizar datos y componer mapas para imprimir. (Escobar, 2012)

La plataforma GeoQ permite el desarrollo de nuevas soluciones mediante personalizaciones, en este caso, para representar espacialmente los activos del sistema de guardia en la Universidad de las Ciencias Informáticas.

#### *2.2 Metodologías de desarrollo de software*

La construcción de un software es guiado por metodologías de desarrollo, las cuales tienen como objetivo fundamental guiar todos los procesos de desarrollo del software. Las metodologías se dividen en dos grandes grupos de acuerdo al énfasis realizado en la documentación del desarrollo; estas pueden ser ágiles o robustas. Para el desarrollo de sistemas que permitan la representación espacial de los procesos de planificación y control del sistema de guardia obrera-estudiantil se considera entre los principales exponentes de los dos grupos: RUP (Proceso Unificado de Desarrollo) y XP (Programación Extrema).

---

<sup>2</sup> Lenguaje C mejorado

<sup>3</sup> Módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

### 2.2.1 Proceso Unificado de Desarrollo (RUP(Rational Unified Process))

El Proceso Unificado de Desarrollo es un proceso de desarrollo de software que constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. (Álvarez, 2012)

La misma tiene presente como principios de desarrollo:(Álvarez, 2012)

**Adaptar el proceso:** El proceso deberá adaptarse a las necesidades del cliente siendo importante interactuar con él. Las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico.

**Equilibrar prioridades:** Los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un equilibrio que satisfaga los deseos de todos.

**Demostrar valor iterativamente:** Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y la calidad del producto y se refina la dirección del proyecto así como también los riesgos involucrados.

**Colaboración entre equipos:** Debe haber una comunicación fluida para coordinar requisitos, desarrollo, evaluaciones, planes, resultados, etc.

**Elevar el nivel de abstracción:** Motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcas de referencia. Esto evita que los ingenieros vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la mejor manera los requisitos sin comenzar desde un principio pensando en la reutilización del código.

**Enfocarse en la calidad:** El control de calidad no debe realizarse al finalizar en cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

### 2.2.2 ¿Por qué RUP?

Tras el análisis de la metodología anteriormente mencionada, se puede concluir que RUP es la metodología idónea para llevar a cabo el proceso de desarrollo de software, ya que a pesar de ser caracterizada por una metodología compleja, presenta características que pueden ser empleadas en diferentes situaciones. La misma es una metodología utilizada en proyectos a largo plazo como el que se desea desarrollar y que requiere de un equipo de desarrollo para su correcto funcionamiento. RUP a su vez requiere de una amplia

documentación, permitiendo desarrollar aplicaciones, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software. Es una metodología guiada por casos de uso, centrada en la arquitectura e iterativa incremental, lo cual permite una mejor organización y agregar nuevas funcionalidades a la solución que se desea desarrollar, sin que la misma presente problemas en la solución existente.

### 2.3 Lenguaje de modelado

#### 2.3.1 Lenguaje Unificado del Modelado (UML)

El Lenguaje Unificado del Modelado (UML) es “un lenguaje estándar para escribir planos de software”. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema intensivo de software. (Pressman, 2010)

Los rasgos principales que han contribuido a hacer de UML el estándar de la industria en la actualidad, se mencionan a continuación: (Rumbaugh, 2004)

- ✚ Permite modelar sistemas utilizando técnicas orientadas a objetos.
- ✚ Adecuado a las necesidades de conectividades actuales y futuras.
- ✚ Es un lenguaje muy expresivo que cubre las vistas necesarias para desarrollar y luego desplegar los sistemas.
- ✚ Ampliamente utilizado por la industria del software.
- ✚ Reemplaza a decenas de notaciones empleadas por otros lenguajes.
- ✚ Modela estructuras complejas.
- ✚ Comportamiento del sistema: casos de usos, diagramas de secuencia, de colaboración, que sirven para evaluar el estado de las máquinas.

#### 2.3.2 ¿Por qué lenguaje UML?

El análisis realizado, se puede concluir que el lenguaje de modelado UML es el lenguaje idóneo para visualizar, especificar, construir y documentar los artefactos pertenecientes al proceso de desarrollo de software, debido a que permite el modelado de sistemas utilizando técnicas orientadas a objetos. UML es un lenguaje fácil de utilizar por los usuarios, permitiendo la creación de diagramas los cuales son utilizados a lo largo del ciclo de vida del proceso de desarrollo del software.

### 2.4 Herramientas CASE

Las herramientas CASE se definen como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores durante el ciclo de vida del desarrollo de un software. La realización de un nuevo software requiere que las tareas sean organizadas y completadas de forma correcta y eficiente. Las herramientas CASE fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo del software. (Murillo, 1999)

#### 2.4.1 Visual Paradigm

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios. (Rumbaugh, 2004)

El sistema presenta un conjunto de características que hace posible su utilización para el desarrollo de software:

- ✚ Entorno de creación de diagramas para UML.
- ✚ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✚ Uso de un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación.
- ✚ Capacidades de ingeniería directa (versión profesional) e inversa.
- ✚ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✚ Disponibilidad de múltiples versiones, para cada necesidad.
- ✚ Disponibilidad de integrarse en los principales IDEs<sup>4</sup>.
- ✚ Disponibilidad en múltiples plataformas.

#### 2.4.2 ¿Por qué Visual Paradigm?

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, el cual presenta características que son factibles a utilizar durante el proceso de desarrollo de software. Es una herramienta

---

<sup>4</sup> Entorno de desarrollo integrado

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.

### *2.5 Lenguaje de programación*

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. (Marvin, 2008)

#### *2.5.1 C++*

C++ es un lenguaje versátil, potente y general. C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original. (García, 1998)

C++ es a la vez un lenguaje procedural (orientado a algoritmos) y orientado a objetos. Como lenguaje procedural se asemeja al C y es compatible con este presentando ciertas ventajas. Como lenguaje orientado a objetos se basa en una filosofía completamente diferente, que exige del programador un completo cambio de mentalidad. Las características propias de la Programación Orientada a Objetos.

#### *2.5.2 ¿Por qué C++?*

Se utiliza C++ como lenguaje de programación debido que es un lenguaje que presenta funcionalidades proporcionadas por bibliotecas que junto con Qt (Quarks Technologies) facilita su uso, para dar solución al sistema que se desea desarrollar, siendo un lenguaje orientado a objetos y con características propias de la programación orientada a objetos. C++ es una herramienta usada en el desarrollo de aplicaciones de escritorio, además de ser utilizado como lenguaje de programación en el desarrollo de aplicaciones en el proyecto SIG-Desktop, como la plataforma de desarrollo GeoQ.

### 2.6 Entorno de Desarrollo Integrado (IDE)

#### 2.6.1 QtCreator

QtCreator es un IDE multiplataforma que se ajusta a las necesidades de la compañía que soporta a Qt<sup>5</sup>, por lo que propicia una mayor integración y aumenta el número de funcionalidades. (Qt Project Hosting, 2012)

QtCreator 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.

A continuación se presentan algunas de las características principales de QtCreator:

- ✚ Editor de código con soporte para C++, QML<sup>6</sup> y ECMAScript<sup>7</sup>.
- ✚ Herramientas para la rápida navegación del código.
- ✚ Control estático de código y estilo a medida que se escribe.
- ✚ Soporte para refactorización de código.
- ✚ Ayuda sensitiva al contexto.
- ✚ Paréntesis coincidentes y modos de selección.

#### 2.6.2 ¿Por qué QtCreator?

Se realizó un análisis sobre el IDE QtCreator concluyendo que: es el IDE idóneo para el desarrollo de un sistema para la representación espacial de los activos del sistema de guardia y la información asociada a la planificación y el control de la asistencia al servicio de guardia, debido a que es un IDE multiplataforma para el desarrollo de aplicaciones en C++ de forma sencilla. QtCreator brinda un mejor trabajo en cuanto a la creación de interfaces gráficas, web, multihilos (múltiples hilos de ejecución) y bases de datos, además de lo anteriormente planteado se utiliza como IDE para el desarrollo de aplicaciones de escritorio en el proyecto SIG-Desktop y se utilizó para el desarrollo del framework de desarrollo GeoQ formando parte del

---

<sup>5</sup> Biblioteca multiplataforma para desarrollar interfaces gráficas de usuario

<sup>6</sup> Lenguaje de Modelado Qt.

<sup>7</sup> Lenguaje de tipo dinámico inspirado en Java y en otros lenguajes.

conjunto de herramientas a utilizar en el desarrollo de la aplicación para dar solución a la problemática planteada.

### 2.7 Gestores de bases de datos

#### 2.7.1 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD<sup>8</sup> y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (Martínez, 2010)

Dentro de los componentes del sistema PostgreSQL se encuentran:

- ✚ Aplicación cliente.
- ✚ Demonio postmaster.
- ✚ Ficheros de configuración.
- ✚ Procesos hijos postgres.
- ✚ Sección anticipada de escritura (WAL).
- ✚ Disco del núcleo caché.
- ✚ Disco.

Entre sus principales características se encuentran:

- ✚ Copias de seguridad en caliente.
- ✚ Múltiples métodos de autenticación.
- ✚ Completa documentación.
- ✚ Licencia BSD.
- ✚ Disponible para Linux y UNIX en todas sus variantes y Windows 32/64bit.

#### 2.7.2 ¿Por qué PostgreSQL?

Tras realizar un análisis de uno de los gestores de bases de datos más usados, se concluye que, dado las características que presenta PostgreSQL, es el gestor de base de datos a utilizar para la gestión de los datos asociados a los activos del sistema de guardia, siendo un gestor fácil de administrar, multiplataforma,

---

<sup>8</sup> Sistema de Base de Datos



de sintaxis SQL estándar y fácil de aprender, con mejor soporte para triggers y procedimientos en el servidor, y ofreciendo soluciones en campos en las que MySQL no podría, siendo eficiente para el trabajo con grandes volúmenes de datos, como son los datos de la información asociada a los activos del sistema de guardia. PostgreSQL contiene replicasiones de datos y cuenta con un soporte empresarial disponible. Posee una gran escalabilidad, siendo capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, permitiéndole soportar una mayor cantidad de peticiones simultáneas de manera correcta, llegando a soportar el triple de carga de lo que soporta MySQL. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.





### 2.8 Herramientas de administración

#### 2.8.1 PgAdmin

PgAdmin es un diseño integral de interfaces para la gestión de base de datos PostgreSQL, diseñado para funcionar con la mayoría de los sistemas operativos, dicho software está escrito en C++ y utiliza un conjunto de herramientas de plataforma cruzada. PgAdmin es una aplicación nativa, se ejecuta en código binario (no en una máquina virtual), por lo que ofrece un excelente rendimiento.(pgAdmin, 2009)

PgAdmin está diseñado para responder a las necesidades de todos los usuarios, desde escribir simples consultas SQL hasta crear complejas bases de datos. La interfaz gráfica soporta todas las características de PostgreSQL y facilita la administración. La aplicación también incluye un resaltado de sintaxis SQL, un editor de código del lado del servidor, un agente de programación y apoyo para el motor de replicación Slonyl<sup>9</sup>. La conexión con el servidor se puede hacer a través de TCP / IP o Unix Domain Sockets y puede ser encriptado SSL para la seguridad. No se requieren controladores adicionales para comunicarse con el servidor de base de datos.(pgAdmin, 2009)

Entre las características principales de pgAdmin se encuentran:

-  Multiplataforma.
-  Diseñado para múltiples versiones de PostgreSQL y derivados.
-  Amplia documentación.
-  Interfaz multilingüe.

---

<sup>9</sup>Slonyl: Sistema de replicación para PostgreSQL.

- ✚ Acceso a datos.
- ✚ Mantenimiento de rutina.
- ✚ El acceso a todos los objetos de PostgreSQL.
- ✚ Multibyte de apoyo.

### 2.8.2 ¿Por qué pgAdmin?

Se realizó un análisis sobre algunas de las herramientas de administración existentes y se llegó a la conclusión que pgAdmin se utilizará como herramienta de administración, debido a las características propias, que permite un mejor trabajo y diseño de interfaces para la gestión de base de datos PostgreSQL. Dicha herramienta permite el desarrollo de complejas bases de datos a través de consultas SQL, siendo factible dada la gran cantidad de volúmenes de datos a manejar para la realización de la aplicación a desarrollar, además de que posee acceso a todos los objetos de PostgreSQL.

### 2.9 Conclusiones del capítulo

Para el desarrollo del sistema propuesto se utilizará como metodología de desarrollo RUP, utilizando como lenguaje de modelado a UML en su versión 2.1 y Visual Paradigm para UML en su versión 8.0 para la modelación. Se utiliza además para el desarrollo de la aplicación como framework de desarrollo GeoQ en su versión 1.0, utilizando como lenguaje de programación C++ y como IDE de desarrollo QtCreator en su versión 2.5.0. Se utiliza como gestor de base de datos PostgreSQL en su versión 9.1 y como herramienta de administración pgAdmin en su versión 1.14.0.

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 del proceso de desarrollo.

### Capítulo III: Presentación de la solución propuesta

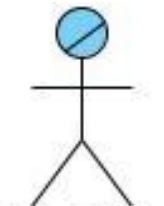

En este capítulo se realizará una descripción de la solución propuesta en términos de procesos del negocio, actores y trabajadores del negocio, así como la justificación de su uso. Se expone el diagrama de casos de uso del negocio y de actividades, así como la descripción de los casos de uso. Seguidamente se definirán los actores del sistema, mostrando el diagrama de casos de uso del sistema, así como su respectiva descripción.

#### 3.1 Modelo del negocio




El modelado del negocio es una técnica para comprender los procesos del negocio de la organización. Está soportada por dos tipos de modelo UML: modelos de casos de uso y modelos de objetos. Ambos se definen en la extensión UML relativa al negocio. Un modelo de casos de uso del negocio describe los procesos del negocio de una empresa en términos de casos de uso del negocio y actores del negocio y se corresponden con los procesos del negocio y los clientes respectivamente. (Rumbaugh, 2004)

##### 3.1.1 Actores y trabajadores del negocio

Un actor del negocio representa a un individuo, grupo, organización, máquina o sistema que interactúa con el negocio y se beneficia de su existencia. (Rumbaugh, 2004)

Actor	Descripción
 Jefe del Área	Los días quince de cada mes inicializa el proceso del negocio de planificación de la guardia obrera-estudiantil, así como el control de la misma.
 Representante del Área	El representante del área o el oficial de guardia se encargan de llevar a cabo la representación de la información asociada al sistema de guardia sobre el mapa.

Los trabajadores del negocio representan personas o sistemas que están involucrados en uno o más procesos del negocio, que participan en ellos, pero no obtiene ningún resultado de valor. (Rumbaugh, 2004)

Trabajadores del Negocio	Descripción
 <p data-bbox="321 636 656 701">&lt;&lt;Trabajador del Negocio&gt;&gt; Planificador</p>	<p data-bbox="862 531 1430 663">Es el encargado de llevar a cabo el proceso de planificación de la guardia obrera-estudiantil.</p>
 <p data-bbox="321 909 656 974">&lt;&lt;Trabajador del Negocio&gt;&gt; Oficial de Guardia</p>	<p data-bbox="862 814 1430 900">Se encarga de controlar la asistencia al servicio de guardia obrera-estudiantil.</p>
 <p data-bbox="310 1188 667 1262">&lt;&lt;Trabajador del Negocio&gt;&gt; Especialista</p>	<p data-bbox="862 1087 1430 1220">Se encarga de llevar a cabo la representación sobre mapas de la planificación de la guardia.</p>

A continuación se mostrará el **Modelo de Objetos**, el cual indica lo que un proceso de negocio va a hacer, esto sirve como una abstracción del como los trabajadores del negocio y las necesidades de las entidades empresariales están vinculadas, y del como colaboran para llevar a cabo el negocio. (Rosas, 2011)



Figura 1: Modelo de Objetos

La siguiente figura representa el diagrama de **Casos de Uso del Negocio (CUN)** que describe las relaciones que existe entre un caso de uso del negocio y un actor del negocio. (Rumbaugh, 2004)

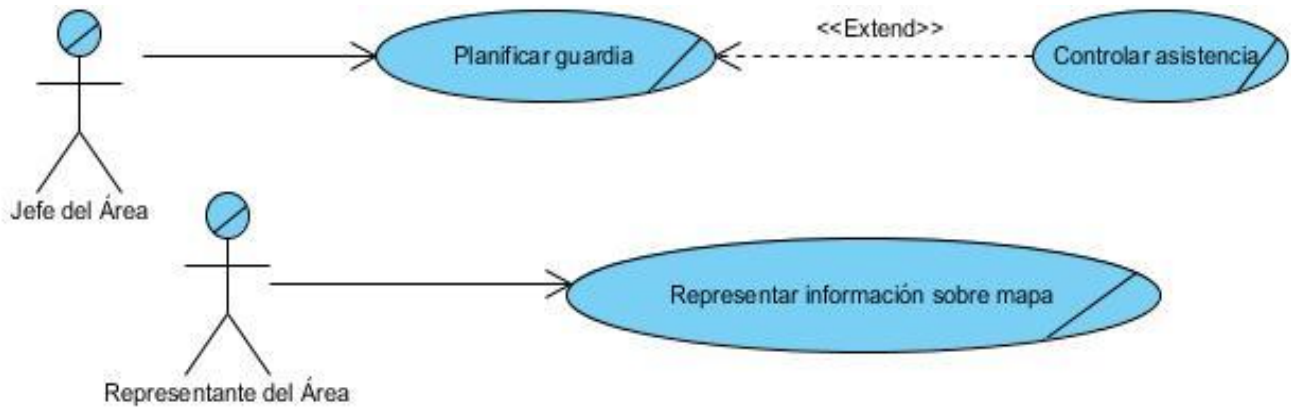


Figura 2: Diagrama de casos de uso del negocio

### 3.1.2 Diagrama de actividades

Los diagramas de actividades se encargan de describir en detalle lo que pasa dentro del negocio y permite examinar los roles específicos que juegan los trabajadores del negocio y las actividades que realizan, también identifica que funciones deberá asumir el producto de software y quiénes serán los actores del sistema.(Rumbaugh, 2004)

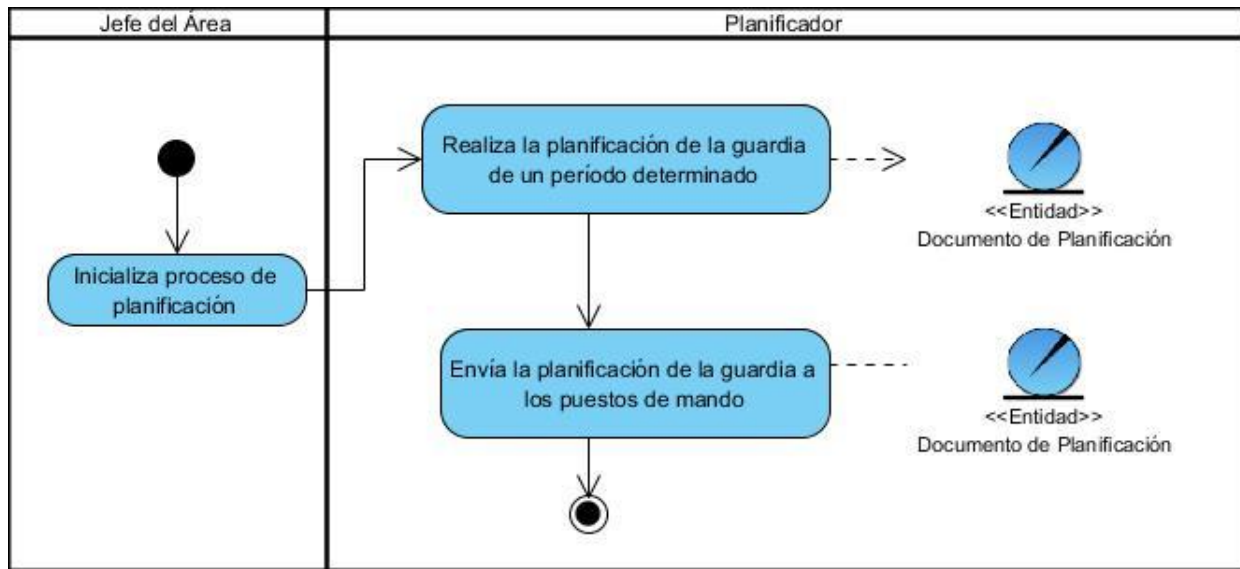


Figura 3: Diagrama de actividades. CU Planificar guardia

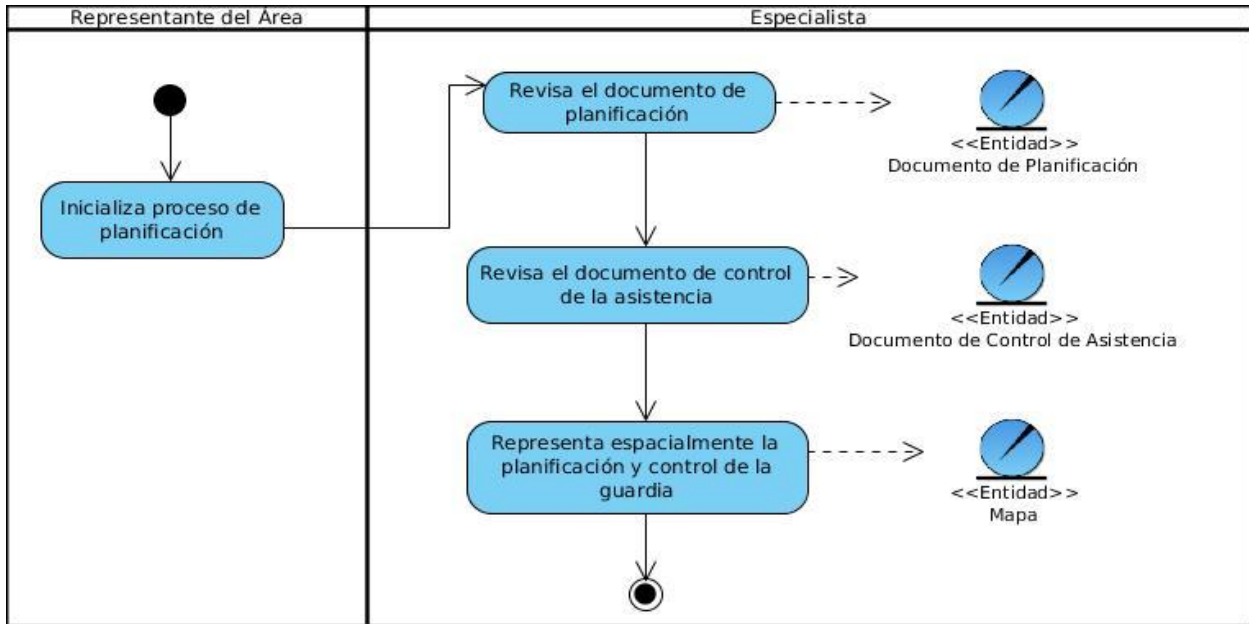


Figura 4: Diagrama de actividades. CU Controlar asistencia

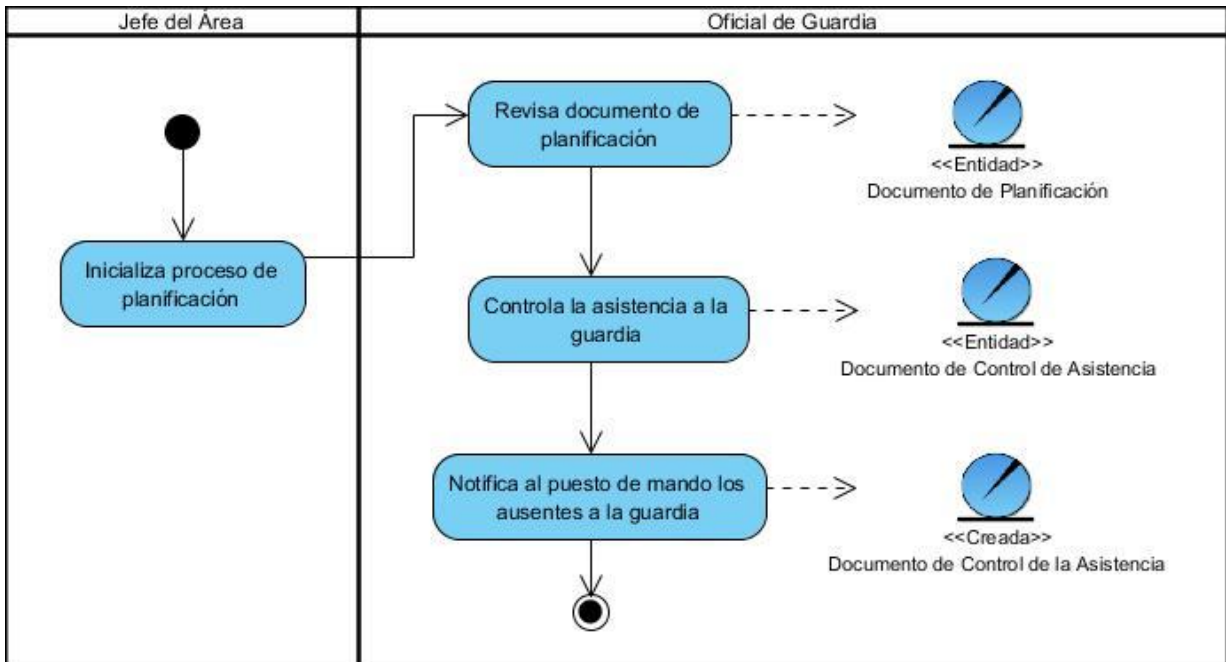


Figura 5: Diagrama de actividades. CU Representar información sobre mapa

### 3.1.3 Descripción de los casos de uso del negocio

#### Descripción del Caso de Uso “Planificar guardia”

<b>Caso de uso del Negocio</b>	Planificar guardia
<b>Actores</b>	Jefe del Área
<b>Trabajadores</b>	Planificador
<b>Resumen</b>	El caso de uso inicia cuando el Jefe del Área inicializa el proceso del negocio de planificación de la guardia obrera-estudiantil y termina cuando se ha realizado el proceso de planificación.
<b>Casos de usos asociados</b>	Controlar asistencia.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Proceso del Negocio</b>
1. El caso de uso inicia cuando el actor inicializa el proceso del negocio al planificar la guardia obrera-estudiantil.	2. El Planificador realiza el proceso de planificación de la guardia obrera-estudiantil en la universidad para un período determinado, concluyendo así el caso de uso.
<b>Pos condición</b>	Se realiza el proceso de planificación de la guardia de un período determinado para cada una de las áreas con las cuales cuenta la universidad.

Descripción del Caso de Uso “**Controlar asistencia**”

<b>Caso de uso del Negocio</b>	Controlar asistencia
<b>Actores</b>	Jefe del Área
<b>Trabajadores</b>	Oficial de Guardia
<b>Resumen</b>	El caso de uso inicia cuando el Jefe del Área inicializa el proceso del negocio de planificación de la guardia obrera-estudiantil y termina cuando se realiza el control de la asistencia.
<b>Casos de usos asociados</b>	Ninguno
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Proceso del Negocio</b>



1. El caso de uso inicia cuando el actor inicializa el proceso del negocio al planificar la guardia obrera-estudiantil.	2. El Oficial de Guardia controla la asistencia de los estudiantes y profesores en un turno determinado.
	3. El Oficial de Guardia entrega el documento de control de la asistencia al puesto de mando.
<b>Pos condición</b>	Se realiza el control de la asistencia.

Descripción del Caso de Uso “**Representar información sobre mapa**”

<b>Caso de uso del Negocio</b>	Representar información sobre mapa
<b>Actores</b>	Representante del área
<b>Trabajadores</b>	Especialista
<b>Resumen</b>	El caso de uso inicia cuando el tiempo inicializa el proceso del negocio de planificación de la guardia obrera-estudiantil y termina cuando se realiza la representación de la información sobre el mapa.
<b>Casos de usos asociados</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Proceso del Negocio</b>
1. El caso de uso inicia cuando el actor inicializa el proceso del negocio al planificar la guardia obrera-estudiantil.	2. El Especialista representa en el mapa la información asociada a la guardia obrera-estudiantil.
<b>Pos condición</b>	Queda representada la información sobre mapas de la guardia obrera-estudiantil.

### 3.2 Requisitos

#### 3.2.1 Requisitos funcionales

Los requisitos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir. (Pressman, 2005) Como requisitos funcionales de la solución propuesta se tienen:

- RF.1: Añadir una nueva área al sistema de guardia.
- RF.2: Editar la información de un área determinada.
- RF.3: Eliminar un área del sistema de guardia.
- RF.4: Añadir una nueva posta al sistema de postas existentes.
- RF.5: Editar la información de una posta determinada.
- RF.6: Eliminar una posta del sistema de guardia.
- RF.7: Añadir una zona vulnerable al sistema de guardia.
- RF.8: Editar información de una zona vulnerable.
- RF.9: Eliminar una zona vulnerable del sistema de guardia.
- RF.10: Adicionar nuevo turno de guardia en un área determinada.
- RF.11: Modificar el horario de un turno de guardia.
- RF.12: Eliminar un turno de guardia en un área determinada.
- RF.13: Registrar un recurso humano en un sistema de guardia.
- RF.14: Editar los datos de un recurso humano en el sistema.
- RF.15: Eliminar los datos de un recurso humano en el sistema de guardia.
- RF.16: Añadir rol a un usuario.
- RF.17: Eliminar rol de un usuario determinado.
- RF.18: Importar datos de los recursos humanos de los servicios Web de la UCI.
- RF.19: Agregar agrupación al sistema de guardia.
- RF.20: Modificar agrupación del sistema de guardia.
- RF.21: Eliminar agrupación del sistema de guardia.
- RF.22: Crear un nuevo rol de usuario.
- RF.23: Modificar un rol.
- RF.24: Eliminar un rol establecido.
- RF.25: Asignar un rol a un usuario.
- RF.26: Quitar permiso de rol.
- RF.27: Buscar activo del sistema de guardia.
- RF.28: Identificar activo sobre el mapa.

RF.29: Publicar guardia planificada.

RF.30: Registrar asistencia al servicio de guardia.

RF.31: Eliminar registro de asistencia.

RF.32: Resaltar visualmente las postas cuyo personal de guardia esté incompleto.

RF.33: Exportar planificación del servicio de guardia.

RF.34: Exportar los datos de los activos del servicio de guardia de un área.

RF.35: Exportar mapa como imagen

RF.36: Representar la planificación de la guardia.

### 3.2.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Normalmente están relacionados con los requisitos funcionales, es decir una vez que 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. Se emplearán componentes que indiquen al usuario el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable.
- ✚ El software tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.
- ✚ El Sistema tendrá una correcta Arquitectura de la Información, a partir de un estudio de usuarios para su etiquetado e indexado.
- ✚ Las funcionalidades principales del sistema estarán orientadas a iconos para un mayor reconocimiento por parte del usuario.

#### Fiabilidad

- ✚ La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores.
- ✚ La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- ✚ 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.

### **Eficiencia**

- ✚ El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento. Para un mapa con extensión municipal, el tiempo de respuesta promedio es menor que 1 segundo. Para una cartografía de extensión de datos igual a Cuba, por cada capa podría demorarse hasta 2 y 3 segundos dependiendo necesariamente de la cantidad de objetos de la misma.
- ✚ Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

### **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.
- ✚ El producto de software final debe diseñarse sobre una arquitectura modelo-vista-controlador.
- ✚ Se deben emplear los estándares establecidos (diseño de interfaces, base de datos y codificación). Se debe lograr un producto altamente configurable y extensible, teniendo en cuenta que se desarrollará sobre la Plataforma Quantum GIS y que constituye una plataforma de desarrollo para ser personalizada como aplicaciones a la medida, pudiéndose incorporar a estas nuevas funcionalidades.

### **Interfaz de Usuario**

El sistema debe:

- ✚ Tener una apariencia profesional y un diseño gráfico sencillo.
- ✚ Posibilitarle al usuario la configuración del entorno de trabajo.
- ✚ Ser intuitivo.

### **Interfaces de Hardware**

Para las PCs clientes:

- ✚ Se requiere al menos 128 MB 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 de arquitectura modelo-vista-controlador centrada en los datos. Las PCs clientes:

- ✚ El sistema GeoQ-Guardián es un sistema multiplataforma que puede ejecutarse en los siguientes sistemas operativos:
  - ✓ GNU/Linux.
  - ✓ Mac OS.
  - ✓ 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 producto GeoQ-Guardián 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 Legales, Derecho del Autor y otros**

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

### **Confidencialidad**

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

### **Integridad**


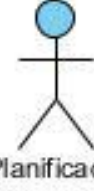


La información manejada por el sistema es objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma es considerada igual a la fuente o autoridad de los datos. Se incluye también mecanismos de chequeo de integridad y realización de auditorías por el personal calificado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

### **Disponibilidad**

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

### 3.3 Descripción del sistema propuesto

#### 3.3.1 Descripción de los actores

Actores del Sistema	Descripción
 <p>Usuario</p>	<p>Se encarga de exportar la planificación, exportar los datos de un activo, así como exportar el mapa como imagen.</p>
 <p>Planificador</p>	<p>Se encarga de llevar a cabo el proceso de planificación de la guardia obrera-estudiantil.</p>
 <p>Oficial de Guardia</p>	<p>Se encarga de controlar el estado de la asistencia durante el servicio de guardia.</p>
 <p>Especialista</p>	<p>Se encarga de representar espacialmente los activos del sistema de guardia, así como la información asociada al proceso de planificación.</p>

#### 3.3.2 Diagrama de casos de uso del sistema

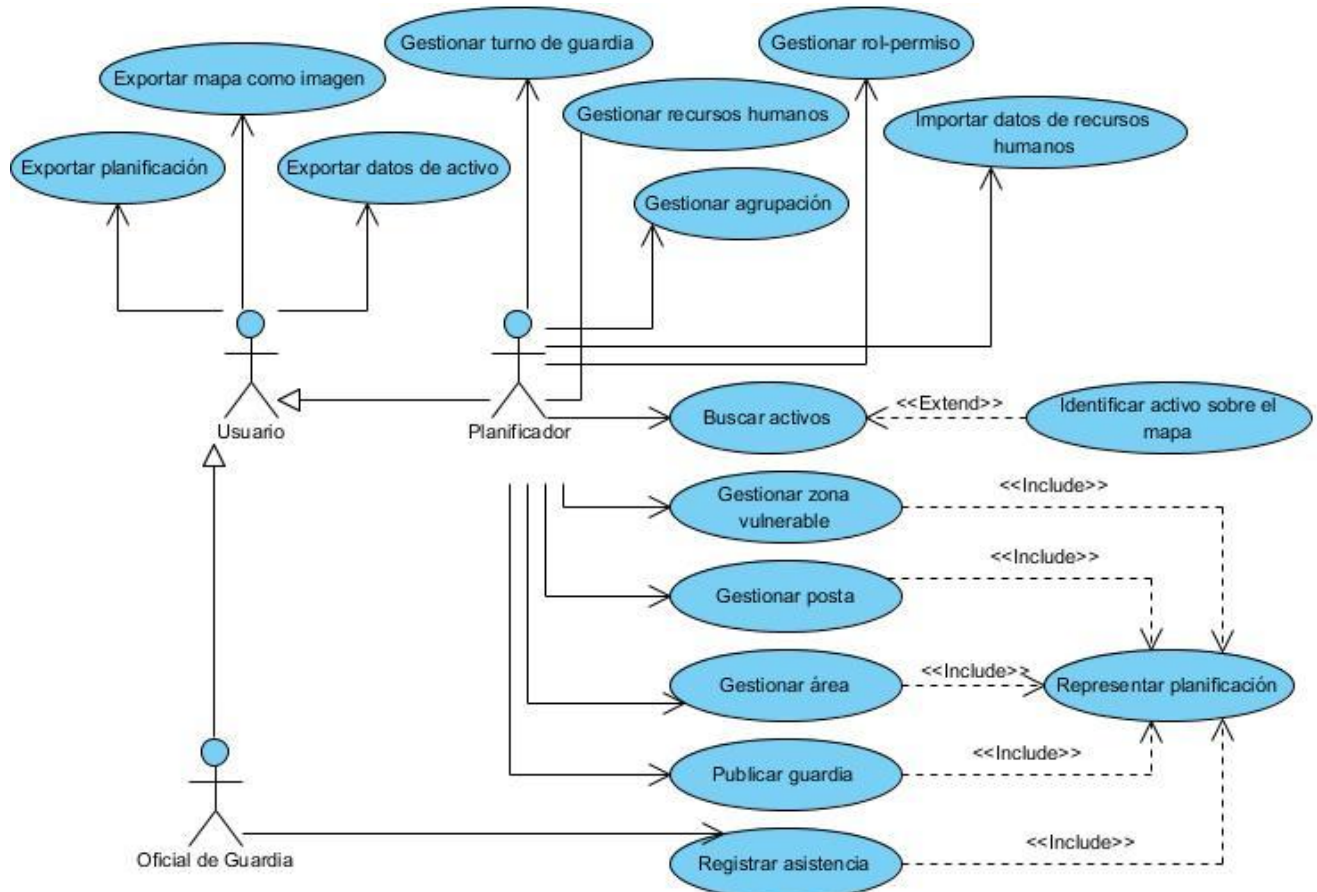


Figura 6: Diagrama de Casos de Uso del Sistema (DCUS)

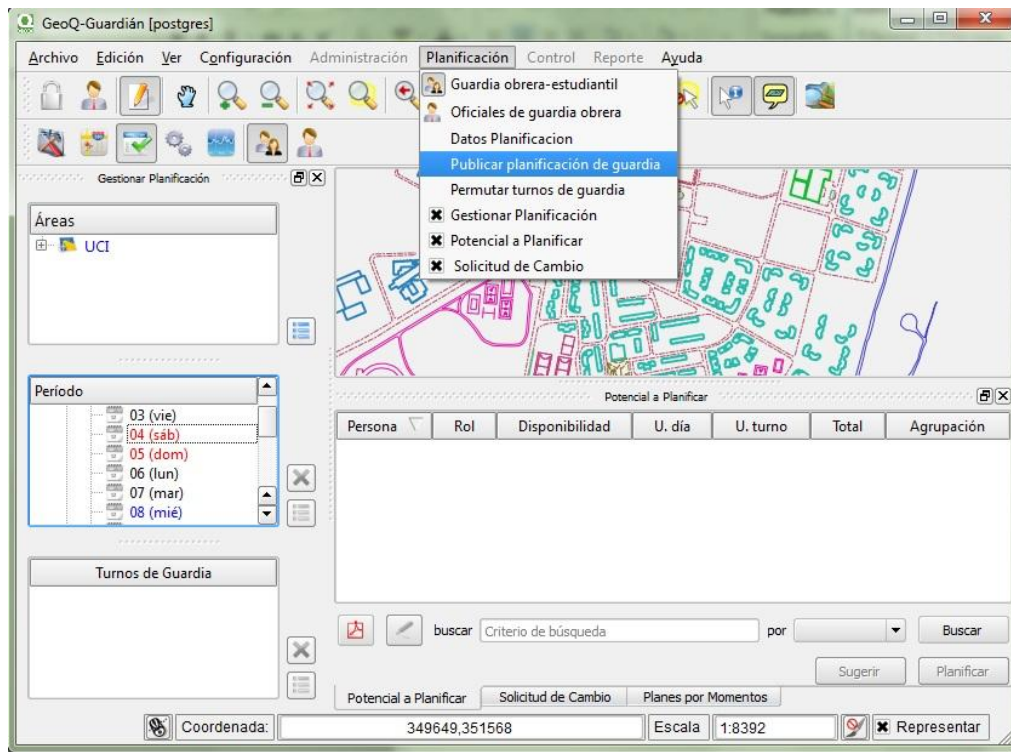
### 3.3.3 Descripción de casos de uso del sistema

A continuación se muestra la descripción textual de uno de los principales Casos de Uso del Sistema, Publicar guardia.

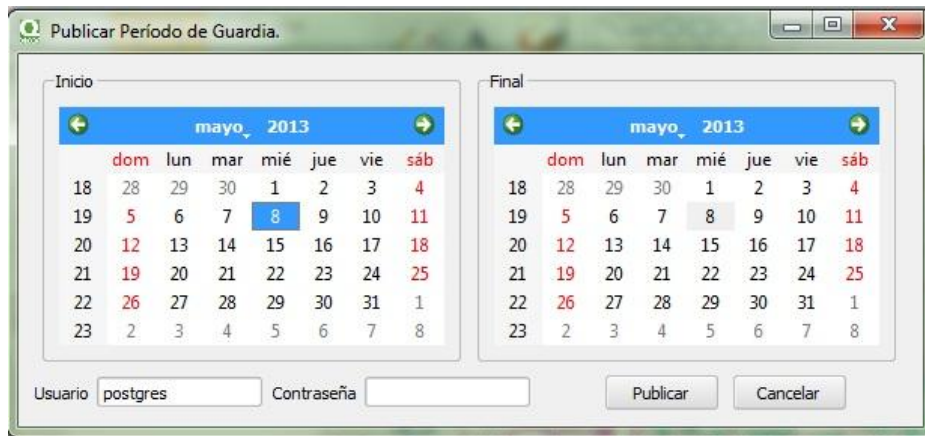
<b>Caso de uso</b>	Publicar guardia
<b>Actores</b>	Planificador
<b>Resumen</b>	El caso de uso inicia cuando el usuario selecciona la opción Publicar planificación de la guardia del menú Planificación, el sistema muestra una interfaz para especificar el período a publicar. El caso de uso termina cuando el sistema envía por correo avisos de la planificación de la guardia.

<b>Precondiciones</b>	Debe haberse seleccionado el área donde se va a planificar la guardia. El usuario debe estar autenticado.	
<b>Referencias</b>	<b>RF.29, RF.36</b>	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El caso de uso inicia cuando el usuario selecciona del menú Planificación la opción Publicar planificación de guardia. (Ver interfaz 55.)	2. El sistema muestra una nueva interfaz (Ver interfaz 56) que muestra la guardia planificada para un período determinado.	
3. El usuario hace clic izquierdo la fecha de inicio y fin de la planificación y, luego introduce el usuario y la contraseña.	4. El sistema notifica por correo a los usuarios correspondientes la planificación de la guardia, concluyendo así el caso de uso.	
5. El usuario da clic izquierdo en el botón Publicar.	6. El sistema muestra un mensaje “El proceso de publicación de la planificación ha terminado”, concluyendo así el caso de uso.	
<b>Prototipo de Interfaz</b>		





Interfaz 55

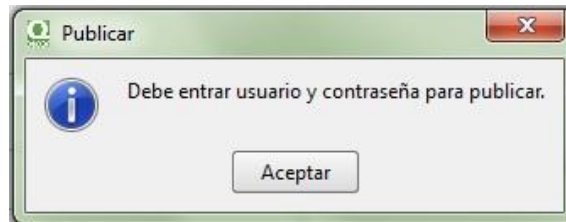


Interfaz 56

Flujo Alterno

5. El usuario hace clic izquierdo en el botón Cancelar.	6. El sistema no guarda los cambios de la planificación y termina el caso de uso.
5. El usuario introduce incorrectamente el usuario o la contraseña.	6. El sistema muestra el siguiente mensaje “Debe entrar usuario y contraseña para publicar”. (Ver interfaz 57).
7. El usuario hace clic izquierdo en el botón Aceptar y regresa al paso 2.	

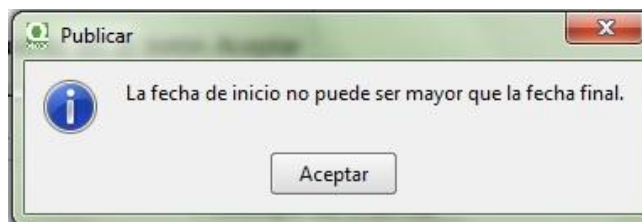
**Prototipo de Interfaz**



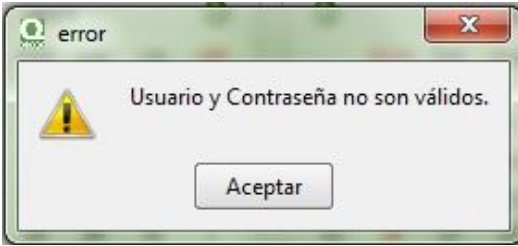
**Interfaz 57**

	6. El sistema muestra un mensaje de información “La fecha de inicio no puede ser mayor que la fecha final”. (Ver interfaz 58).
7. El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de error y regresa al paso 2.	

**Prototipo de Interfaz**



**Interfaz 58**

	6. El sistema muestra un mensaje de error "Usuario y Contraseña no son válidos".
7. El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de error y regresa al paso 2.	
<p><b>Prototipo de Interfaz</b></p>  <p><b>Interfaz 59</b></p>	
<b>Pos condición</b>	Se publica la planificación de la guardia y se notifica a los usuarios por correo electrónico.

La descripción textual de los restantes casos de uso del sistema se puede encontrar en el **Anexo 1**.

### 3.4 Conclusiones del capítulo

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 Negocio permitió tener un amplio conocimiento de las necesidades del negocio, así como de los actores y trabajadores que interactúan con él, permitiendo delimitar cuales serán cada una de las funcionalidades específicas para cada uno de los trabajadores que interactúan con el negocio.
- ✚ 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.

- ✚ Los requisitos funcionales 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 IV: Construcción de la solución propuesta*

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 de análisis, 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 análisis, diseño, interrelación, modelo entidad relación, 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 Análisis*

El análisis de requisitos genera la especificación de características operacionales de software; indica la interfaz del software en otros elementos del sistema, y establece las restricciones que debe tener el software. Este permite que el ingeniero de software se extienda sobre requerimientos básicos establecidos durante tareas anteriores a la ingeniería de requisitos y construya modelos que representen escenarios del usuario, actividades funcionales, clases de problemas y sus relaciones, el comportamiento de las clases y el sistema y, a medida que se transforma el flujo de datos. (Pressman, 2008)

##### *4.1.1 Modelo del análisis*

El modelo de análisis ofrece una visión conceptual precisa y unificada de alternativas para la implementación de los sistemas. La estructura impuesta por el modelo de análisis se define mediante una jerarquía y su estructura se representa mediante un sistema de análisis que denota el paquete de más alto nivel del modelo. Este modelo crece incrementalmente conforme se analizan más casos de uso, de manera que el sistema se construye como una estructura de clases de análisis y relaciones entre dichas clases. (Pressman, 2002)

### 4.1.2 Diagrama de clases del análisis

El Modelo de Clases de Análisis es un artefacto que representa la abstracción de una o varias clases y subsistemas del diseño del sistema. Los casos de uso se describen mediante clases del análisis. El mismo presenta las siguientes características:

- ✚ Está centrado en los requisitos funcionales.
- ✚ Su comportamiento está definido mediante responsabilidades.
- ✚ Define atributos que son conceptuales y reconocibles en el dominio del problema.
- ✚ Encaja en uno de los tres estereotipos básicos: de interfaz, de control y de entidad.

**Entidad:** Se utiliza para modelar información que posee una vida larga y que a menudo es persistente.

**Interfaz:** Se utiliza para modificar la interacción entre el sistema y sus actores, lo que implica recibir y representar informaciones y peticiones de los usuarios y sistemas externos.

**Control:** Se usa para encapsular el control de un caso de uso en concreto. También se utiliza para representar derivaciones y cálculos complejos.

En este epígrafe se llevará a cabo la representación de los principales diagramas de clases del análisis. Los restantes diagramas del análisis se establecerán en el **Anexo 2**.

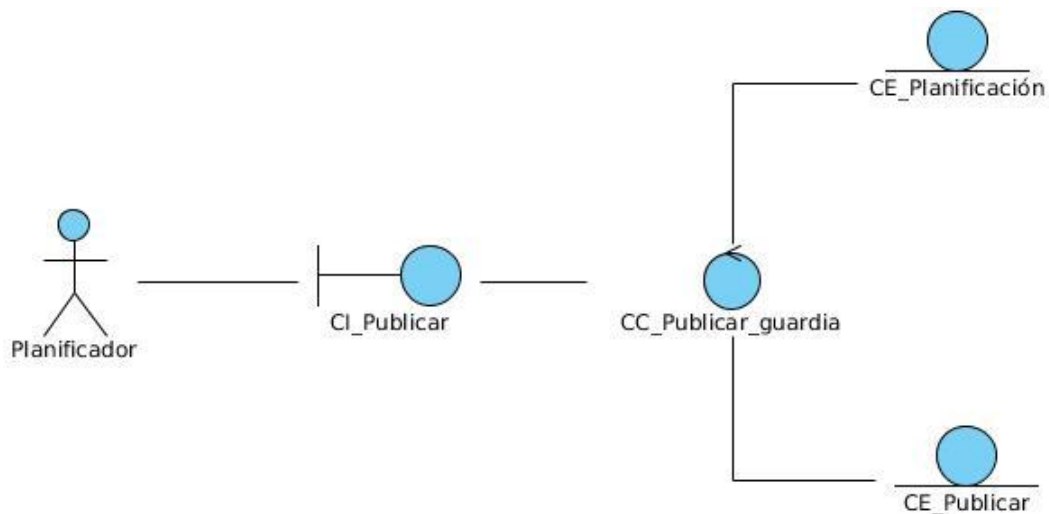


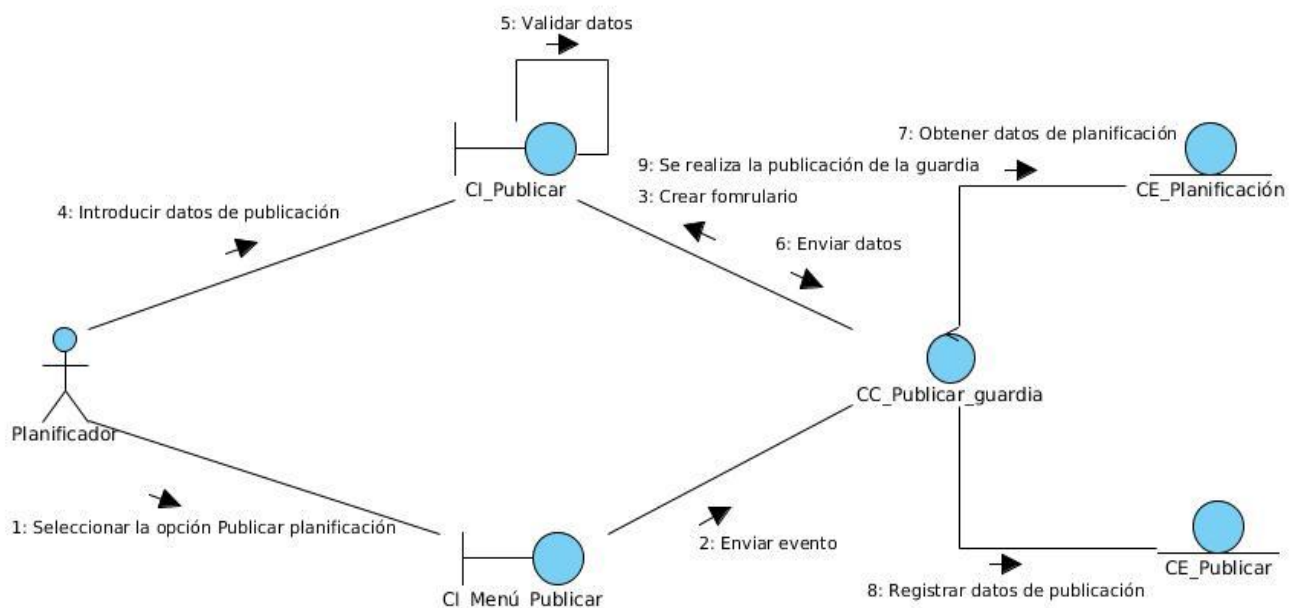
Figura 7: Diagrama de clases del análisis CU Publicar guardia

### 4.1.3 Diagrama de interacción

Los diagramas de interacción son diagramas que describen cómo grupos de objetos colaboran para conseguir algún fin. Estos diagramas muestran objetos, así como los mensajes que se pasan entre ellos dentro de un caso de uso. Los diagramas de interacción capturan el comportamiento de los casos de uso. Esto no solo es importante para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa. (Pressman, 2002)

#### 4.1.3.1 Diagrama de colaboración

El diagrama de colaboración es un diagrama que muestra las interacciones organizadas alrededor de los roles. Estos no muestran el tiempo como una dimensión aparte, por lo que se hace necesario etiquetar con números tanto la secuencia de mensajes, como los hilos concurrentes. Un diagrama de colaboración es también un diagrama de clases que contiene roles de clasificador y roles de asociación en lugar de solo clasificador y asociaciones, estos describen la configuración de los objetos y de los enlaces que pueden ocurrir cuando se ejecuta una instancia de la comunicación. (Pressman, 2005)



**Figura 8: Diagrama de colaboración CU Publicar guardia**

En este epígrafe se llevará a cabo la representación de los principales diagramas de colaboración. Se mostrarán los restantes casos de uso en el **Anexo 3**.

### 4.1.3.2 Diagrama de secuencia

Los diagramas de secuencia modelan la interacción entre objetos en un sistema, permitiendo describir gráficamente la interacción de un conjunto de objetos en una aplicación a través del tiempo. (Pressman, 2002) En este epígrafe se hará una breve descripción de los diagramas de secuencia más significativos y se mostrarán los demás referenciados en el **Anexo 4**.

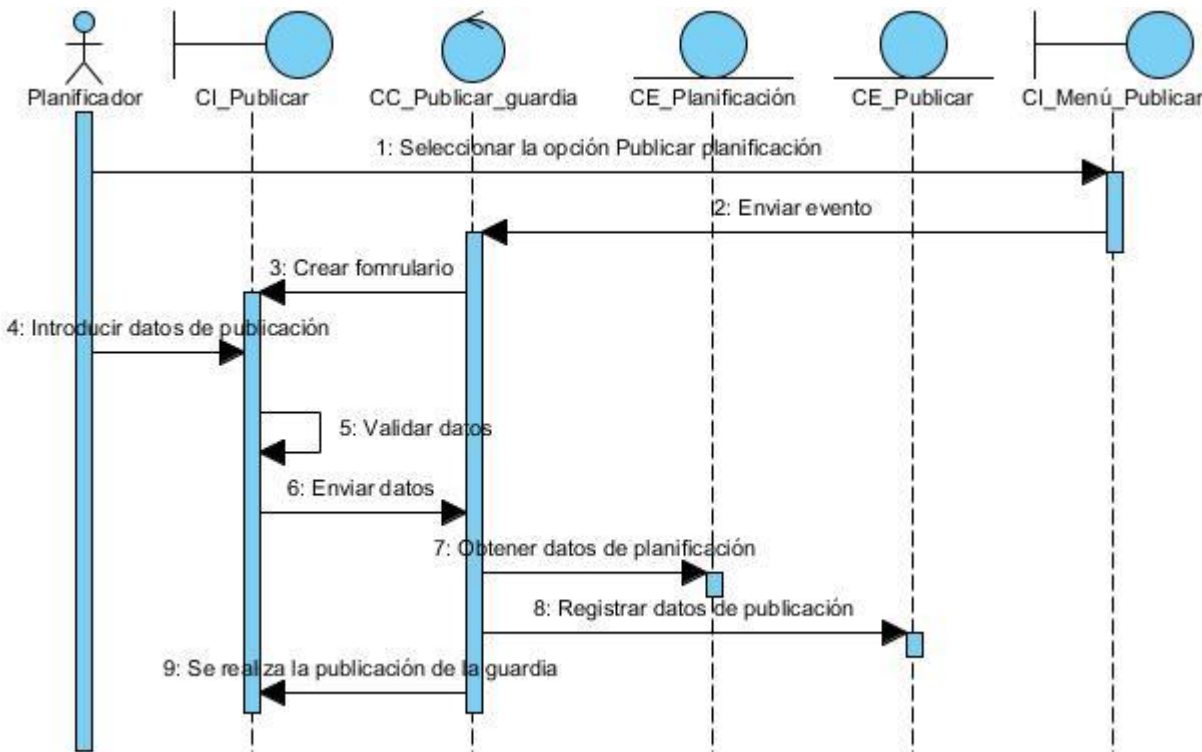


Figura 9: Diagrama de secuencia CU Publicar guardia

## 4.2 Diseño

El flujo de trabajo de diseño crea una representación o modelo de software que se encarga de proporcionar detalles acerca de la estructura de datos, las arquitecturas, las interfaces y los componentes del software que son necesarios para implementar el sistema. (Pressman, 2002)

### 4.2.1 Patrones

Un patrón involucra una descripción general de una solución concurrente a un problema recurrente con soluciones recurrentes con diversos objetivos y restricciones. Los patrones cubren diferentes rangos de



escala de abstracción. Algunos ayudan a estructurar un sistema en subsistemas, otros a refinar otros subsistemas y sus relaciones, otros solucionan una implementación en particular. (Rosánigo, 2000)

Los patrones dados las características que posee se dividen en dos grandes grupos, los patrones arquitectónicos y los patrones de diseño.

### 4.2.1.1 Patrones arquitectónicos

Los patrones arquitectónicos describen los principios fundamentales de la arquitectura de un sistema de software. Este a su vez identifica los subsistemas, define sus responsabilidades y establece las reglas y guías para organizar la relación entre ellos. (Rosánigo, 2000)

Entre los patrones arquitectónicos utilizados en el proceso de desarrollo del software decidimos utilizar el patrón Modelo-Vista-Controlador:

**Modelo-Vista-Controlador:** Este patrón consiste en separar los intereses propios de cada una de las distintas unidades que define. Estas unidades son Modelo, Vista y Controlador: la Vista se encarga de todo lo referente a la presentación de la aplicación, como son las imágenes a mostrar, cajas de texto, distribución de los elementos en la pantalla, colores, formatos, entre otros. El Modelo se encarga de la lógica o reglas del negocio y del acceso a datos. El Controlador actúa como el interlocutor entre la Vista y el Modelo, llevando las solicitudes del usuario al Modelo y notificando a la Vista según sea necesario. (Paredes, 2009)

Los beneficios de usar MVC son: interfaz de usuario intercambiable fácilmente, reusabilidad de componentes del Modelo en distintas Vistas, facilidad de soportar nuevos tipos de clientes y facilitar las pruebas de unidad. Los problemas que produce el uso del MVC son: complejidad creciente de desarrollo y acoplamiento desde la Vista hacia el Modelo.

### 4.2.1.2 Patrones de diseño

Los patrones de diseño proveen un esquema para refinar los subsistemas y las relaciones entre los componentes de un sistema. Describen una estructura recurrente de comunicación entre componentes para resolver un problema general de diseño dentro de un contexto particular. Se encuentran en un nivel intermedio de granularidad entre los patrones arquitectónicos y las expresiones idiomáticas. Tienden a operar independientemente de un paradigma particular de programación o lenguaje de programación. (Rosánigo, 2000)

En la actualidad existen tres grandes grupos de patrones de diseño los cuales son de **creación**, de **estructura** y de **comportamiento**. Para el desarrollo de este sistema se utilizarán los patrones de diseño de creación y de comportamiento, específicamente los que se mostrarán a continuación.

**Patrones de creación:** este patrón implica el proceso de instanciar objetos. Ayudan a que el sistema sea independiente de cómo se crean, componen y representa los objetos. (Escuela Politécnica Superior, 2009)

- ✚ **Singleton (Instancia única):** Asegura que una clase tenga una única instancia y proporciona un punto de acceso global a dicha instancia.

**Patrones de comportamiento:** caracterizan las formas en las que interactúan y reparten responsabilidades las distintas clases u objetos.

- ✚ **Observador:** Define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos.

### 4.2.1.3 Patrones GRASP (Patrón de Asignación de Responsabilidades)

A continuación se presentan los patrones de diseño GRASP utilizados para el desarrollo de la aplicación. Estos patrones se usan para asignar responsabilidades.

- ✚ **Experto en información:** Asignar la responsabilidad de realizar una labor a la clase que contiene toda la información necesaria para cumplir la tarea encomendada.
- ✚ **Creador:** La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos.
- ✚ **Controlador:** Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

### 4.2.2 Diagrama de clases del diseño

Los diagramas de clases se utilizan para modelar la visión estática de un sistema. Esta visión soporta los requisitos funcionales del sistema, en concreto, los servicios que el sistema debería proporcionar a sus

usuarios finales. Normalmente contienen: clases, interfaces y relaciones entre ellas: de asociación, de dependencia y/o generalización.

Los diagramas de clases también pueden contener a paquetes o subsistemas, que se usan para agrupar elementos del modelo en partes más grandes (paquetes que a su vez contienen a varios diagramas de clases). (Pressman, 2005)

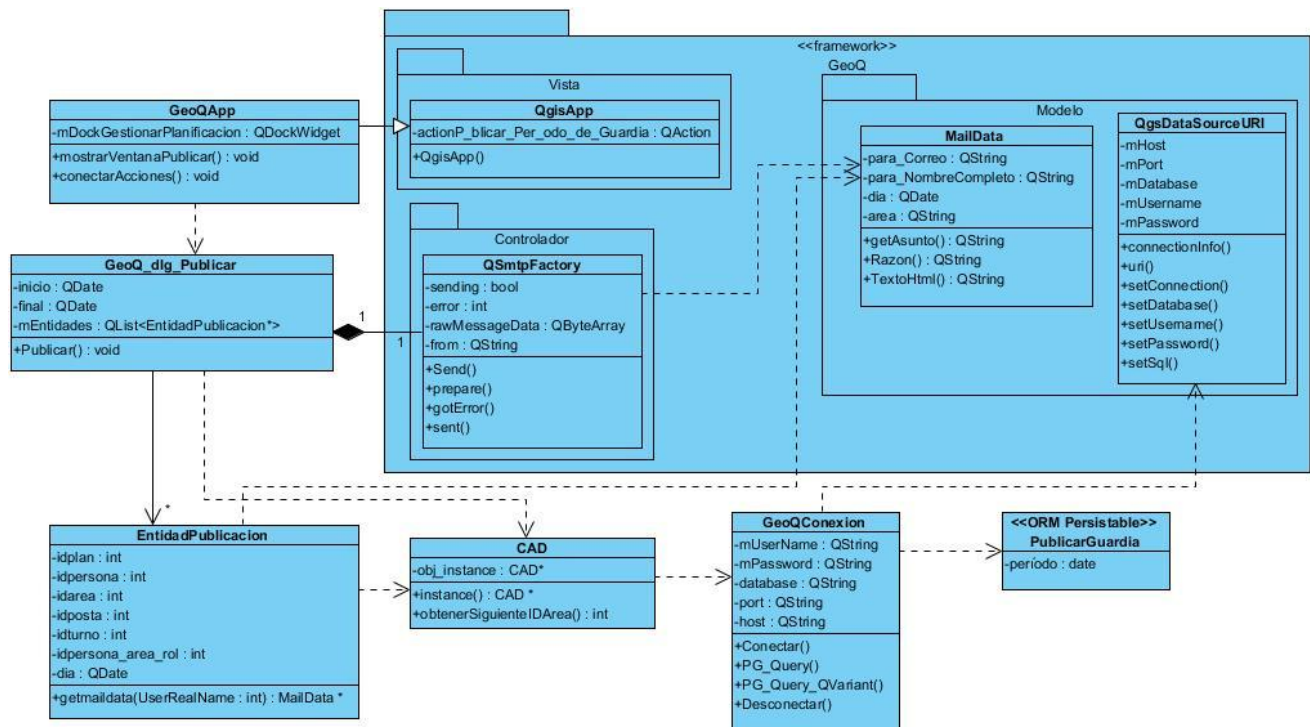


Figura 10: Diagrama de clases del diseño. CU Publicar guardia

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

**QgsAttributeTableDialog:** Clase interfaz perteneciente al framework GeoQ, presenta las acciones correspondientes para adicionar, modificar, eliminar y buscar un listado de mail elementos.

**QgsAttributeDialog:** Clase interfaz del Framework GeoQ, representa en un cuadro de diálogo los atributos de un elemento seleccionado en el listado que contiene QgsAttributeTableDialog.

**GeoQ\_dlg\_Gestionar:** Clase interfaz que permite listar los datos de los activos del sistema de guardia (áreas, postas, recursos humanos, zonas vulnerables), también brinda opciones para la gestión de los mismos, como pueden ser: adicionar, eliminar, modificar y buscar un activo determinado.

**CAD:** Clase de acceso a datos. En la misma se encuentran las operaciones que permiten acceder a los datos almacenados en la base de datos de GeoQ-Guardián.

**GeoQConexion:** Clase conexión que contiene los atributos necesarios para establecer la conexión con la base de datos, y realizar consultas, para adicionar, modificar, eliminar y listar datos.

**QgsVectorLayer:** Clase controladora del framework GeoQ encargada de manipular elementos de una capa vectorial. Se encarga de adicionar, modificar, eliminar y 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 GeoQ, contiene un mapa con los atributos de la tabla de la base de datos para una capa específica, es una forma de abstracción para manipular de forma genérica los elementos de cualquier tabla de la base de datos.

**QgsPostgresProvider:** Proveedor de acceso a la base de datos que proporciona el framework GeoQ.

**QgsDataSourceURI:** Clase del framework GeoQ que proporciona la cadena de conexión a una base de datos.

En este epígrafe se llevará a cabo la representación de los principales diagramas de clases del diseño. Se mostrarán los restantes casos de uso en el **Anexo 5**.

### *4.2.3 Diseño de la base de datos*

El diseño de una base de datos consiste en definir la estructura de los datos que debe de tener una base de datos de un sistema de información determinado. En el caso relacional, esta estructura será un conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, entre otros. (Costal, 2009)

4.2.3.1 Diagrama de clases persistentes

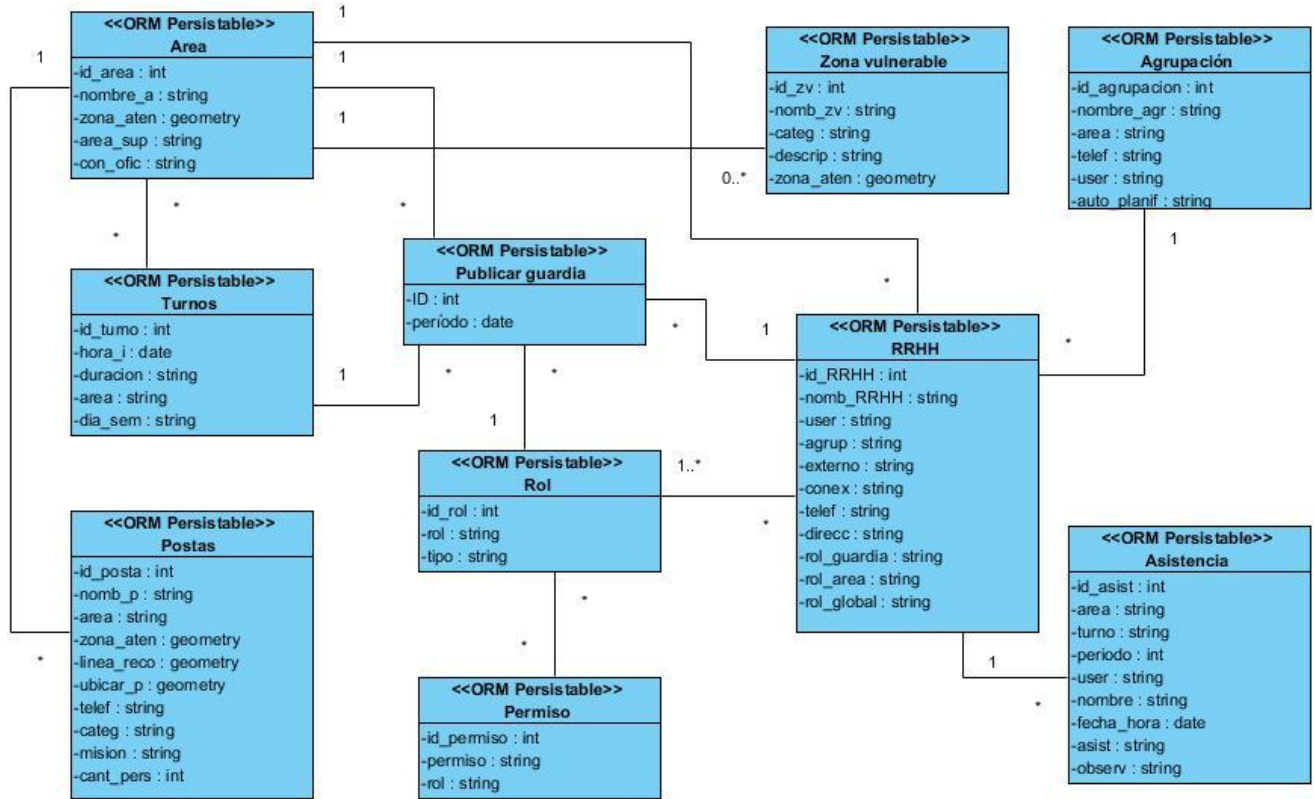


Figura 11: Diagrama de clases persistentes

### 4.2.3.2 Modelo Entidad-Relación

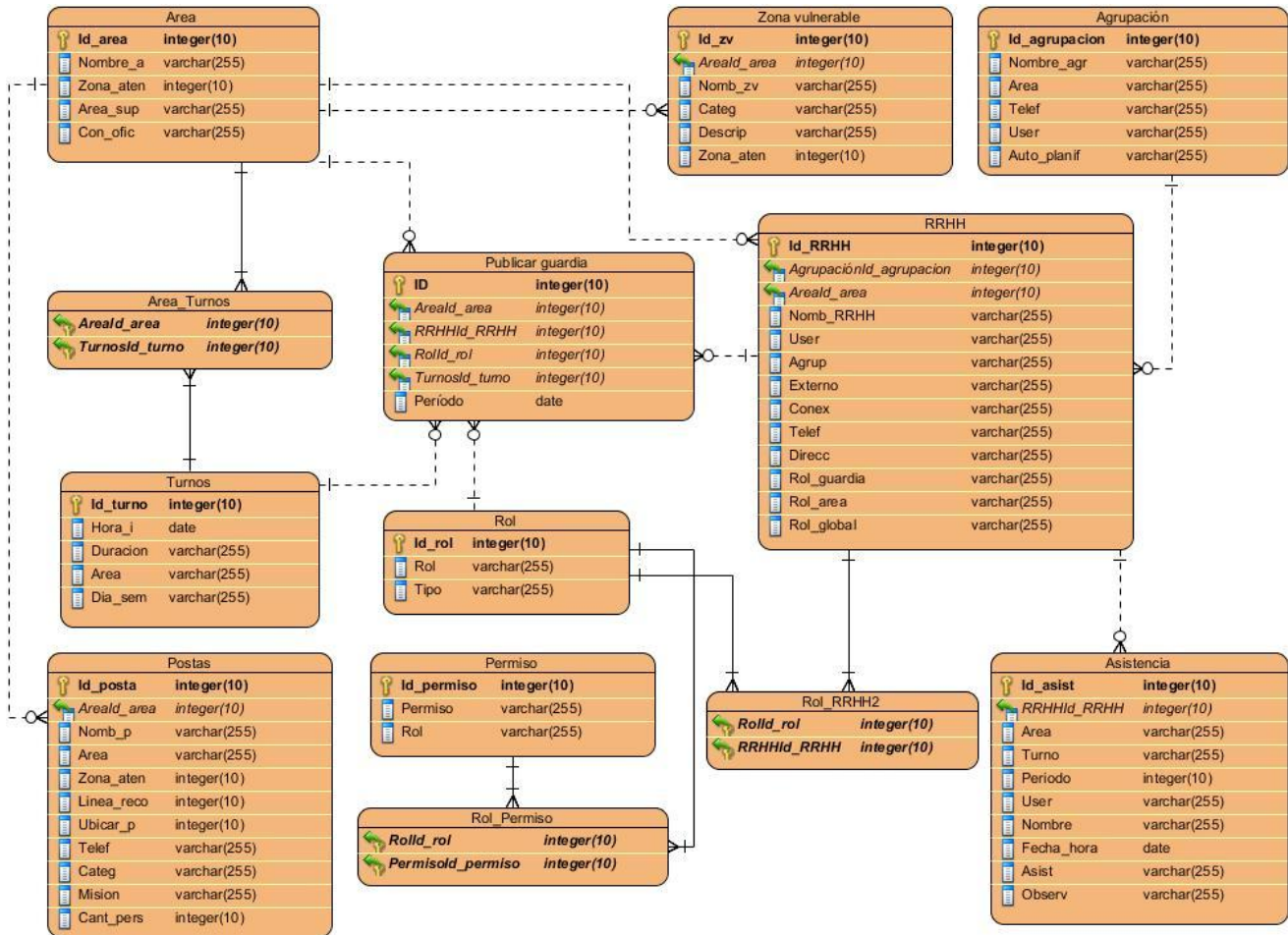


Figura 12: Modelo Entidad-Relación

### 4.3 Implementación

El flujo de trabajo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Además de describir cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado y el cómo dependen los componentes unos de otros. (Rumbaugh, 2004)

#### 4.3.1 Modelo de Despliegue

Modelo de Despliegue que es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes



de software en los nodos físicos. El mismo está compuesto por nodos, dispositivos y conectores. (Pressman, 2005)

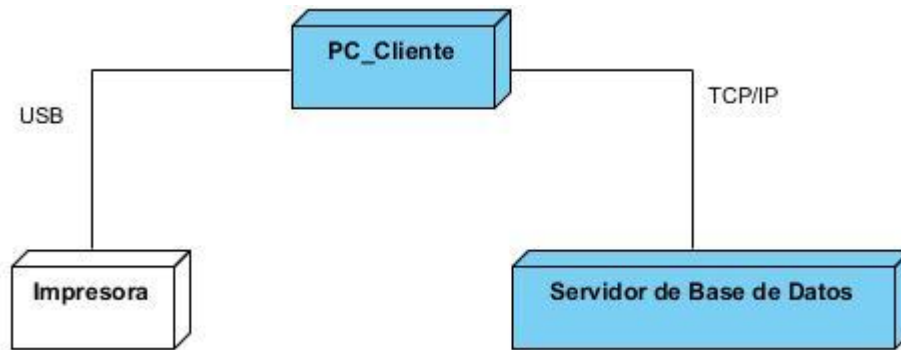


Figura 13: Diagrama de despliegue

### 4.3.2 Diagrama de componentes

Los Diagramas de Componentes modelan la vista estática de un sistema. Se representa como un grafo de componentes unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. Estos tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. (Pressman, 2005)

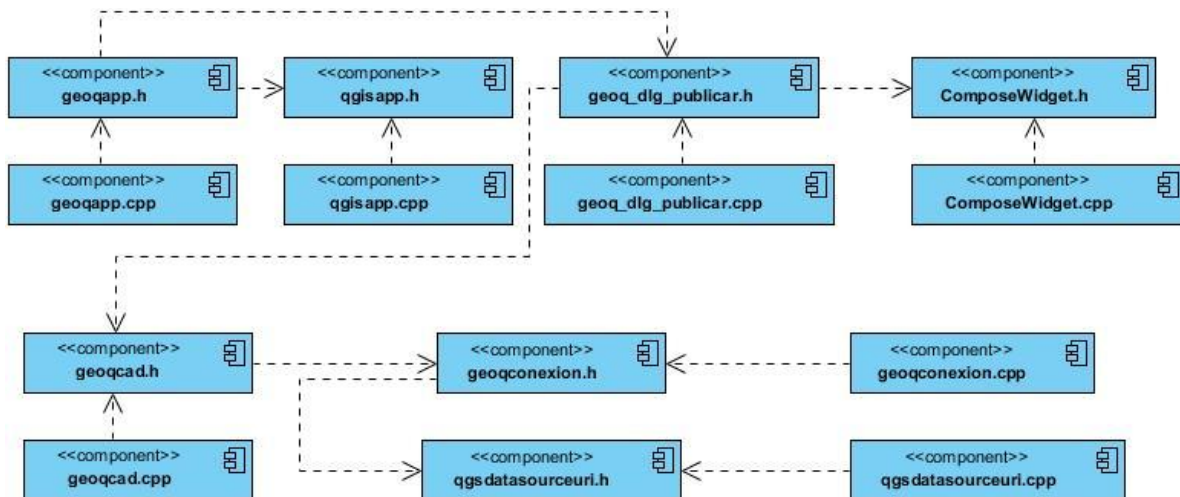


Figura 14: Diagrama de componentes. CU Publicar guardia

Los restantes diagramas de componentes se especifican en el **Anexo 6**.

### 4.4 Prueba

El flujo de trabajo de prueba es el proceso de ejecución de un programa con la intención de descubrir un error. Las pruebas de 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. (Pressman, 2005)

- ✚ **Pruebas de Verificación:** Se comprueba el cumplimiento de las especificaciones del diseño.
- ✚ **Pruebas de Validación:** Se encargan de velar por el cumplimiento de los requisitos del análisis.
- ✚ **Pruebas de Caja Blanca:** Se conoce el código y se trata de ejecutar cada uno de los elementos del mismo.
- ✚ **Pruebas de Caja Negra:** Solamente se conoce la interfaz y se trata de probar cada uno de los elementos que componen a la misma.

Cualquier tipo de prueba tiene el objetivo principal de encontrar errores o defectos en el software. Para probar el sistema propuesto se utilizará la prueba de caja negra.

#### 4.4.1 Pruebas de caja negra

La prueba de Caja Negra, se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que la salida producida es correcta, ya que no es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar. (Pressman, 2005)

Para el desarrollo de las pruebas de caja negra existen varias técnicas, entre las que se encuentran:

- ✚ **Técnica de la Partición de Equivalencia:** Esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del sistema.
- ✚ **Técnica del Análisis de Valores Límites:** Esta técnica prueba la habilidad del sistema para manejar datos que se encuentran en los límites aceptables.
- ✚ **Técnica de Grafos de Causa-Efecto:** Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el sistema, descubre



de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico.

A continuación se realizarán las pruebas de caja negra utilizando la técnica de partición de equivalencia, las demás pruebas realizadas a la aplicación se darán a conocer en los artefactos generados, específicamente los **Diseños de Casos de Prueba**.

### Caso de Uso Gestionar Área

#### Descripción general:

El caso de uso se inicia cuando el usuario selecciona la opción Activos del Área en el submenú Modos del menú Configuración, el sistema muestra el listado de las áreas registradas en el sistema, y las opciones de agregar una nueva área, modificar los datos de un área y eliminar un área existente; el usuario selecciona una de las opciones que le brinda el sistema agregando, actualizando o eliminando un área del sistema concluyendo así el caso de uso.

#### Condiciones de ejecución:

Debe estar activada la opción de edición y un área seleccionada. El usuario debe estar autenticado.

#### Secciones a probar en el caso de uso:

##### Sección “Añadir nueva área”

Escenarios de la sección	Descripción de la funcionalidad	Flujo central
EC 1.1: Añadir nueva área exitosamente.	<p>El sistema muestra una interfaz donde el usuario añade los datos de la nueva área:</p> <ul style="list-style-type: none"><li>- Nombre.</li><li>- Área Superior.</li><li>- Zona de atención.</li><li>- Con oficiales.</li></ul> <p>El usuario entra los datos pertenecientes al área que se desea añadir y hace clic izquierdo en el botón Aceptar. El sistema guarda los cambios, visualiza en el mapa el área con el color</p>	<p>Caso de uso Gestionar área.</p> <ol style="list-style-type: none"><li>1. Página principal.</li><li>2. Clic en el menú Configuración.</li><li>3. Clic en la opción Modos.</li><li>4. Clic en la opción Activos de Área.</li><li>5. Clic en la opción Añadir un área de guardia.</li><li>6. Selecciona en el mapa la zona de atención.</li></ol>

	predefinido por el sistema y termina el caso de uso.	7. Clic en el botón Aceptar.
EC 1.2: Añadir nueva área sin éxito.	El sistema muestra un mensaje de información "Complete todos los campos". El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de error.	Caso de uso Gestionar área <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la opción Añadir un área de guardia.</li> <li>6. Clic en el botón Aceptar.</li> <li>7. Clic en el botón Aceptar para cerrar la ventana de error.</li> </ol>
EC 1.3: Cancelar petición.	El usuario hace clic izquierdo en el botón Cancelar. El sistema no guarda los cambios y termina el caso de uso.	Caso de uso Gestionar área: <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la opción Añadir un área de guardia.</li> <li>6. Clic en el botón Cancelar.</li> </ol>
EC 1.4: Añadir área sin éxito por la existencia de la misma.	El sistema muestra un mensaje de error "El área especificada ya existe". El usuario hace clic izquierdo en el botón Aceptar para cerrar la ventana que contiene el mensaje de información.	Caso de uso Gestionar área: <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la opción Añadir área de guardia.</li> <li>6. Clic en el botón Aceptar.</li> </ol>

		7. Clic en el botón Aceptar para cerrar la ventana de error.
--	--	--

**Sección “Modificar un área”**

<p>EC 2.1: Modificar área exitosamente.</p>	<p>El usuario hace clic izquierdo sobre el encabezado del área que desea modificar y selecciona con clic izquierdo la opción Modificar un área de guardia. El sistema muestra una interfaz con los datos del área seleccionada a modificar:</p> <ul style="list-style-type: none"> <li>- Nombre.</li> <li>- Área Superior.</li> <li>- Zona de atención.</li> <li>- Con oficiales.</li> </ul> <p>El usuario modifica los datos pertenecientes al área seleccionada y hace clic en el botón Aceptar.</p> <ul style="list-style-type: none"> <li>- Nombre.</li> <li>- Área Superior.</li> <li>- Zona de atención.</li> <li>- Con oficiales.</li> </ul> <p>El sistema guarda los cambios, se modifica el área en el mapa y termina el caso de uso.</p>	<p>Caso de uso Gestionar área:</p> <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la cabecera de la tabla del área a modificar.</li> <li>6. Clic en la opción Modificar área de guardia.</li> <li>7. Selecciona en el mapa el área de atención.</li> <li>8. Clic en el botón Aceptar.</li> </ol>
<p>EC 2.2: Modificar área sin éxito.</p>	<p>El sistema muestra un mensaje de error “El área especificada ya existe”. El usuario hace clic izquierdo en el botón Aceptar</p>	<p>Caso de uso Gestionar área:</p> <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la cabecera de la tabla del área a modificar.</li> <li>6. Clic en la opción Modificar área de guardia.</li> <li>7. Clic en el botón Aceptar.</li> </ol>

		8. Clic en el botón Aceptar para cerrar la ventana de error.
EC 2.3: Cancelar petición.	El usuario hace clic izquierdo en el botón Cancelar. El sistema no guarda los cambios y termina el caso de uso.	Caso de uso Gestionar área: <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la cabecera de la tabla del área a modificar.</li> <li>6. Clic en la opción Modificar área de guardia.</li> <li>7. Clic en el botón Cancelar.</li> </ol>
EC 2.4: Modificar área sin éxito por ser el área, área superior de sí mismo.	El sistema muestra un mensaje de información “El área superior no puede ser área superior de sí misma”. El usuario hace clic izquierdo sobre el botón Aceptar.	Caso de uso Gestionar área: <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la cabecera de la tabla del área a modificar.</li> <li>6. Clic en la opción Modificar área de guardia.</li> <li>7. Clic en el botón Aceptar.</li> <li>8. Clic en el botón Aceptar para cerrar la ventana de error.</li> </ol>
EC 2.5: Modificar área sin éxito por ser área superior hija del área.	El sistema muestra un mensaje de información “No puede poner como área superior a otra que sea hija de la seleccionada”. El usuario hace clic izquierdo en el botón Aceptar.	Caso de uso Gestionar área: <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> </ol>

		<ol style="list-style-type: none"> <li>5. Clic en la cabecera de la tabla del área a modificar.</li> <li>6. Clic en la opción Modificar área de guardia.</li> <li>7. Clic en el botón Aceptar.</li> <li>8. Clic en el botón Aceptar para cerrar la ventana de error.</li> </ol>
--	--	---

**Sección “Eliminar un área”**

EC 3.1: Eliminar área exitosamente.	El usuario hace clic izquierdo sobre el área o las áreas que desee eliminar y selecciona la opción Eliminar área de guardia. El sistema muestra un mensaje de información “Desea eliminar “#” de área/áreas seleccionadas” área de guardia”. El usuario hace clic izquierdo sobre el botón aceptar. El sistema elimina de la lista el área o las áreas seleccionadas y termina el caso de uso.	<p>Caso de uso Gestionar área:</p> <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la cabecera de la tabla del área a eliminar.</li> <li>6. Clic en la opción Eliminar área de guardia.</li> <li>7. Clic en la opción Aceptar.</li> </ol>
EC 3.2: Eliminar área sin éxito.	El usuario hace clic izquierdo sobre el área que desea eliminar y selecciona la opción Eliminar área de guardia. El sistema muestra un mensaje de información “Desea eliminar “#” de área/áreas seleccionadas” área de guardia.” El usuario hace clic izquierdo sobre el botón Cancelar. El sistema no elimina de la lista el área o las áreas seleccionadas y termina el caso de uso.	<p>Caso de uso Gestionar área:</p> <ol style="list-style-type: none"> <li>1. Página principal.</li> <li>2. Clic en el menú Configuración.</li> <li>3. Clic en la opción Modos.</li> <li>4. Clic en la opción Activos de Área.</li> <li>5. Clic en la cabecera de la tabla del área a eliminar.</li> <li>6. Clic en la opción Eliminar área de guardia.</li> <li>7. Clic en el botón Cancelar.</li> </ol>

**Descripción de variables:**

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nombre	Campo de texto	No	El usuario escribe el nombre de la nueva área.
2	Zona de atención	Objeto espacial polígono	No	El usuario marca en el mapa la zona de atención de la nueva área.
3	Área superior	Lista de selección	Si	El usuario selecciona de la lista el área a la que se subordina.
4	Con oficial	Lista de selección	No	El usuario selecciona de la lista si va a existir oficial o no.

**Matriz de datos:**

**Sección “Añadir nueva área”**

Id del escenario	Nombre	Zona de atención	Área superior	Con oficial	Respuesta del sistema	Resultado de la prueba
EC 1.1: Añadir nueva área exitosamente.	V “93”	V “dibujar un círculo en el mapa”	V “Residencia 6”	V “si”	El sistema permite añadir una nueva área al sistema de guardia anteriormente preestablecida por la dirección de seguridad y protección.	Satisfactorio
EC 1.2: Añadir nueva área sin éxito.	I “vacío”	V “dibujar un círculo en el mapa”	V “Residencia 6”	V “si”	El sistema muestra un mensaje de información “Complete todos los campos”.	Satisfactorio
EC 1.2: Añadir nueva área sin éxito.	V “93”	I “vacío”	V “Residencia 6”	V “si”	El sistema muestra un mensaje de información “Complete todos los campos”.	Satisfactorio
EC 1.2: Añadir nueva área sin éxito.	V “93”	V “dibujar un círculo en el mapa”	I “vacío”	V “si”	El sistema muestra un mensaje de información “Complete todos los campos”.	Satisfactorio
EC 1.2: Añadir nueva área sin éxito.	V “93”	V “dibujar un círculo en el mapa”	V “Residencia 6”	I “vacío”	El sistema muestra un mensaje de información “Complete todos los campos”.	Satisfactorio
EC 1.3: Cancelar petición.	N/A “Residencia 6”	N/A “dibujar un círculo en el mapa”	N/A “Residencia 6”	N/A “si”	El sistema no guarda los cambios y regresa a la página principal.	Satisfactorio
EC 1.4: Especificación de área.	I “48”	V “dibujar un círculo en el mapa”	V “Residencia 6”	V “si”	El sistema muestra un mensaje “El área especificada ya existe”.	Satisfactorio

**Sección “Modificar un área”**

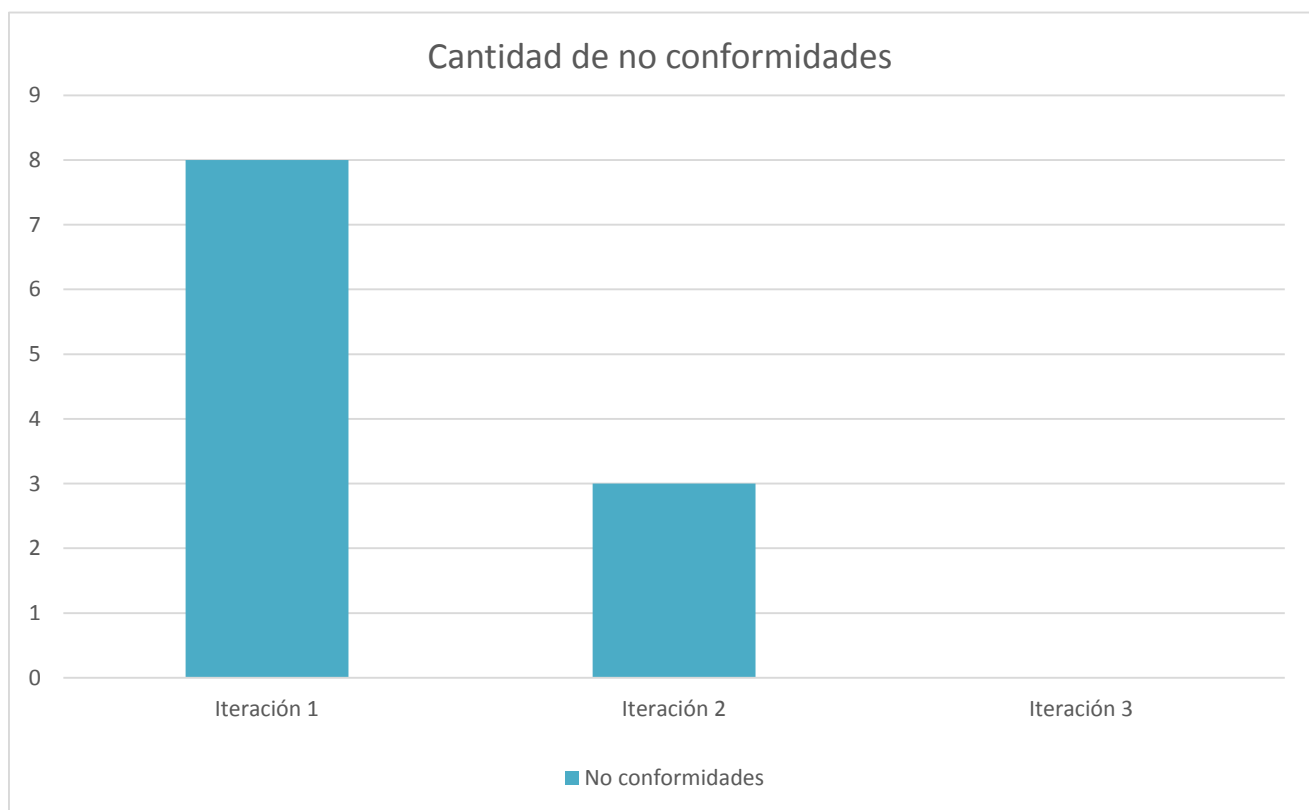
Id del escenario	Área	Nombre	Zona de atención	Área superior	Con oficial	Respuesta del sistema	Resultado de la prueba
EC 2.1: Modificar área exitosamente.	V "Residencia 6"	V "Residencia 8"	V "dibujar un círculo en el mapa"	V "Residencia 4"	V "si"	El sistema guarda los cambios y se modifica el área en el mapa.	Satisfactorio
EC 2.2: Modificar área sin éxito.	V "Residencia 6"	I "Residencia 6"	V "dibujar un círculo en el mapa"	V "Residencia 4"	V "si"	El sistema muestra un mensaje de error "El área especificada ya existe".	Satisfactorio
EC 2.3: Cancelar petición.	N/A "Residencia 6"	N/A "Residencia 8"	N/A "dibujar un círculo en el mapa"	N/A "Residencia 4"	N/A "si"	El sistema no guarda los cambios realizados.	Satisfactorio
EC 2.4: Modificar área sin éxito por ser el área, área superior de sí mismo.	N/A "Residencia 6"	N/A "Residencia 6"	N/A "dibujar un círculo en el mapa"	N/A "Residencia 4"	N/A "si"	El sistema muestra un mensaje de error "El área seleccionada no puede ser área superior de sí misma."	Satisfactorio
EC 2.5: Modificar área sin éxito por ser área superior hija del área	N/A "UCI"	N/A "Facultad 6"	N/A "dibujar un círculo en el mapa"	N/A "Residencia 4"	N/A "si"	El sistema muestra un mensaje de error "No puede poner como área superior a otra área que sea hija de la seleccionada."	Satisfactorio

**Sección "Eliminar un área"**

Id del escenario	Área	Respuesta del sistema	Resultado de la prueba
EC 3.1: Eliminar área exitosamente.	V "Residencia 6"	El sistema elimina de la lista el área o las áreas seleccionadas.	Satisfactorio
EC 3.2: Eliminar área sin éxito.	I "vacío"	El sistema muestra un mensaje de información "Seleccione un área de la lista"	Satisfactorio

### 4.5 Resultado de las pruebas

Se le realizaron un grupo de pruebas de calidad al sistema, con el fin de verificar que el sistema cumpliera con los requisitos funcionales identificados tras el análisis realizado de la situación existente. Una vez realizada una primera iteración se detectó ocho no conformidades, a las cuales se les dio solución, tras una segunda iteración se detectaron tres no conformidades y ya en una tercera no se detectaron no conformidades en el sistema, garantizando la calidad óptima del software. A continuación se muestra una gráfica con la relación de las no conformidades identificadas por el grupo de calidad.



### 4.6 Conclusiones del capítulo

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 análisis, diseño, implementación y prueba. Por lo que se puede concluir que:



- ✚ Se obtuvieron los diagramas de clases del análisis, teniendo una mayor visión del sistema en términos de clases del análisis, detallando las responsabilidades de cada una de las funcionalidades con las cuales cuenta el sistema.
- ✚ Se obtuvieron los diagramas de interacción (colaboración y secuencia) que permitieron valorar el comportamiento de los casos de uso a través de mensajes de envío y respuesta, así como de interacción con el usuario.
- ✚ 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 diseñaron los casos de pruebas utilizando el método de caja negra, probando el sistema y obteniendo resultados satisfactorios.

### *Conclusiones generales*

Con el desarrollo de la presente investigación se obtuvo un Sistema de Información Geográfica para la Seguridad y Protección de la Universidad de las Ciencias Informáticas que permite representar 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 representar la planificación de la guardia obrera-estudiantil y demás activos del sistema de guardia de la institución, logrando una mejor visibilidad del mismo 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

Una vez concluido el trabajo de diploma se recomienda lo siguiente:

- ✚ Integrar a la solución un módulo de análisis espacial que facilite el proceso de toma de decisiones en la dirección de seguridad y protección.
- ✚ Desarrollar algoritmos que permitan automatizar el proceso de planificación de la guardia obrera estudiantil e integrarlos a la solución.
- ✚ Incorporar nuevos servicios de guardia como la cuartelería.

### Referencias bibliográficas

- Álvarez, Gonzálo. 1999.** ¿ Qué es Java ? *Características del lenguaje*. [En línea] 1999. [Citado el: 8 de Enero de 2013.] <http://www.iec.csic.es/criptonomicon/java/quesjava.html>.
- Álvarez, Marcos Manuel. 2012.** *Tecnologías de la Información*. 2012.
- Belmonte, Oscar. 2004.** *Introducción al lenguaje de programación Java: Una guía básica*. 2004.
- Carrillo, Isaías. 2008.** *Metodología del Desarrollo del Software*. 2008.
- Castro, Fidel. 1998.** Ministerio de Cultura de la República de Cuba. *Dirección de Seguridad y Protección*. [En línea] Gaceta Oficial de la República de Cuba, 14 de Junio de 1998. [Citado el: 1 de Diciembre de 2012.] <http://www.min.cult.cu/loader.php?sec=instituciones&cont=direccionseguridad>.
- Costal, Dolores. 2009.** *Introducción al diseño de bases de datos*. 2009.
- Cuesta, Sandra Milena. 2008.** *Definiciones de sistema*. Bogotá : s.n., 2008.
- De Nobrega, María. 2012.** Herramientas CASE: Rational Rose. [En línea] 2012. [Citado el: 10 de Diciembre de 2012.] [http://curso\\_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobre](http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobre).
- EMS Database Management. 2012.** *EMS SQL Manager*. 2012.
- Escobar, Sandor. 2012.** *Sistema de Información Geográfica para automatizar la gestión de la distribución de las redes eléctricas en la Unión Nacional Eléctrica*. Ciudad de la Habana : s.n., 2012.
- Escuela Politécnica Superior. 2009.** *Patrones de Diseño*. 2009.
- García, Javier. 1998.** *Aprenda C++ como si estuviera en primero*. España : s.n., 1998.
- IBM. 2011.** Introducción a Business Process Management (BPM). [En línea] 29 de Abril de 2011. [Citado el: 28 de Febrero de 2013.] <http://www.ibm.com/developerworks/ssa/local/websphere/introduccion-bpm/index.html>.
- Interflex. 2009.** Software selection. *SP-Expert*. [En línea] 2009. [Citado el: 7 de Enero de 2013.] <http://www.softwareseleccion.com/sp+expert-p-2755>.
- Martínez, Rafael. 2010.** Sobre PostgreSQL. [En línea] 2 de Octubre de 2010. [Citado el: 6 de Febrero de 2013.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).

- Marvin, David. 2008.** Definición de lenguaje de programación. Tipos. Ejemplos. [En línea] 16 de Octubre de 2008. [Citado el: 11 de Diciembre de 2012.] <http://catedraprogramacion.foroactivos.net/t83-definicion-de-lenguaje-de-programacion-tipos->.
- Molina, David. 2012.** SiCaTu. *Sistema de Cálculo de Turnos*. [En línea] 9 de Febrero de 2012. [Citado el: 15 de Octubre de 2012.] <http://SicaTu – Sistema de Cálculo de Turnos « molda.es.htm>.
- Murillo, Félix. 1999.** *Herramientas CASE*. 1999.
- Oracle. 2012.** Bienvenido a NetBeans y [www.netbeans.org](http://www.netbeans.org). *¿Qué es NetBeans?* [En línea] Oracle Corporation, 2012. [Citado el: 11 de Diciembre de 2012.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
- . **2011.** Why MySQL? [En línea] 2011. [Citado el: 7 de Febrero de 2013.] <https://www.mysql.com/documentation/index.html>.
- Paredes, Simón Andrés. 2009.** *Diseño e Implementación de la tercera versión del framework Java para aplicaciones Web de la empresa DYBOX*. Santiago de Chile : s.n., 2009.
- Pérez, Liset. 2008.** *Estrategia para el desarrollo de requisitos funcionales de software en proyectos productivos de la UCI*. Ciudad de la Habana : s.n., 2008.
- pgAdmin. 2009.** pgAdmin PostgreSQL tools. [En línea] 26 de Marzo de 2009. [Citado el: 7 de Febrero de 2013.] <http://www.pgadmin.org/licence.php>.
- Prado, Edison. 2009.** *Herramienta inteligente para la captura de requisitos no funcionales*. Ciudad de la Habana : s.n., 2009.
- Pressman, Roger. 2008.** Capítulo 8: Modelado del Análisis. *Ingeniería de Software*. New York : s.n., 2008.
- . **2002.** *Ingeniería de Software. Un enfoque práctico. 5ta Edición*. México : s.n., 2002.
- Pressman, Roger S. 2005.** *Ingeniería del Software*. 2005.
- Pressman, Roger. 2010.** *Software Engineering: A Practitioner's Approach*. New York : s.n., 2010.
- Qt Project Hosting. 2012.** Qt Project. *Qt Creator*. [En línea] 5 de Enero de 2012. [Citado el: 13 de Diciembre de 2012.] [http://qt-project.org/wiki/Category:Tools::QtCreator\\_Spanish](http://qt-project.org/wiki/Category:Tools::QtCreator_Spanish).
- RAE. 2012.** Real Academia Española. [En línea] 2012. [Citado el: 25 de Febrero de 2013.] <http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm>.
- Reynoso, Carlos. 2004.** *Introducción a la arquitectura de software*. Buenos Aires : s.n., 2004.

- Rosánigo, Zulema Beatriz. 2000.** *Maximizando reuso en software para ingeniería estructural: Modelos y Patrones.* Argentina : s.n., 2000.
- Rosas, Gonzalo. 2011.** *Modelado del Negocio con UML.* 2011.
- Rumbaugh, James. 2004.** *El proceso unificado de desarrollo de software.* La Habana : Félix Varela, 2004.
- Sanabria, Javier Manrique. 2011.** *Conceptos de SIG y Cartografía.* Bogotá : s.n., 2011.
- Solex. 2012.** *Solex. Planificación y Control de Turnos de Guardia.* [En línea] 8 de Octubre de 2012. [Citado el: 21 de Octubre de 2012.]
- Tabares, Marta Silvia. 2011.** *Arquitectura de Software Parte 1.* [En línea] 4 de Septiembre de 2011. [Citado el: 10 de Diciembre de 2012.] <http://www.slideshare.net/mstabare/arquitecturas-de-software-parte-1>.