

**Universidad de las Ciencias Informáticas**  
**Facultad 6**



**Título:**

**Módulo de Información Geográfica para la Representación  
Espacial de las Cámaras en el Sistema Xilema Suria.**

**MIGREC-XS**

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

**Autor:** Luis Angel Blanco Correa

**Tutores:** MCs. Grethell Castillo Reyes

Ing. Cesar Santos Sanabria

La Habana, julio de 2016.

“Año 58 del Triunfo de la Revolución”

# DECLARACIÓN DE AUTORÍA

---

## Declaración de Autoría:

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

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

**Autor:**

\_\_\_\_\_

Luis Angel Blanco Correa

**Tutores:**

\_\_\_\_\_

MCs. Grethell Castillo Reyes

\_\_\_\_\_

Ing. Cesar Santos Sanabria



### Datos de Contacto:

**MSc. Grethell Castillo Reyes:** Graduado de Ingeniero en Ciencias Informáticas en La Universidad de las Ciencias Informáticas en el año 2012, MSc En Informática Aplicada en el año 2015. Trabajador del centro GEYSED en el departamento Integración de soluciones, donde desempeña el rol de Jefe de Proyecto de la Línea de Productos de Software Aplicativos SIG

Correo Electrónico: [gcreyes@uci.cu](mailto:gcreyes@uci.cu)

**Ing. Cesar Santos Sanabria:** Graduado de Ingeniero en Ciencias Informáticas en La Universidad de las Ciencias Informáticas en el año 2010. Trabajador del centro GEYSED en el departamento Desarrollo de Componentes, donde desempeña el rol de arquitecto en el proyecto Video-Vigilancia.

Correo Electrónico: [csanabria@uci.cu](mailto:csanabria@uci.cu)



### Agradecimientos:

Agradezco con todo mi cariño y mi amor a las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, a ustedes por siempre estar a mi lado, en especial agradezco:

A mis padres por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional, apoyo perfectamente mantenido a través del tiempo.

A mi hijo, que con su pequeño tamaño, me ha enseñado muchas y sabias lecciones para la vida.

A mi familia que me ha brindado su apoyo incondicional en cada momento de mi carrera y de mi vida, a mi otra mamá Maritza y a mi abuelita Onelia.

A mi tía Dayessi y a toda su familia que me han acogido, ayudado y querido como uno más de los suyos, gracias por todo.

A mis hermanos de corazón Carlito, Víctor, Yamil y Karel en los cuales he encontrado la compañía, el apoyo, la ayuda. Gracias por estar siempre conmigo.

A todos mis amigos de la UCI, en especial para Andy y Cuso que de cierta manera fueron mis compañeros de tesis, que me ayudaron, me apoyaron y compartieron conmigo largas noches de trabajo. A los limoneros, a Chandy mis compañeros de festejos. A todo el que compartió conmigo en las aulas y que de una forma u otra me hicieron crecer como profesionales.

A mi tutora Grethell, y mi amigo Álvaro.

En especial para todas y cada una de las personas que han hecho posible que hoy este aquí.

Los quiero y les agradezco.



## Dedicatoria:

### *A mi mamá:*

*Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, por ser mi mejor amiga y compañera, pero más que nada, por su amor.*

### *A mi papá.*

*Por los ejemplos de perseverancia, familiaridad, por representar un haz de guía que ilumina mis caminos oscuros, por el valor mostrado para salir adelante ante las dificultades, por sus enseñanzas en todas las esferas de la vida y en especial por su amor.*

### *A mi hijo.*

*Hijo, eres mi orgullo y mi gran motivación, libras mi mente de todas las adversidades que se presentan, y me impulsas a cada día superarme en vida profesional para ofrecerte siempre lo mejor. No es fácil, eso lo sé, pero tal vez sí no te tuviera, no habría logrado tantas grandes cosas, tal vez mi vida sería incompleta sin tí.*

### *A mi esposa.*

*Por tu paciencia y comprensión, por preferir sacrificar tu tiempo para que yo pudiera cumplir con el mío. Por tu bondad y sacrificio me inspiraste a ser mejor para tí, ahora puedo decir que esta tesis lleva mucho de tí, gracias por estar siempre a mi lado*

*A todas esas personas importantes en mi vida, que siempre estuvieron listas para brindarme toda su ayuda, "Mi Familia". Con todo mi cariño esta tesis se las dedico a ustedes.*

## RESUMEN

Los proyectos Video Vigilancia Suria y GeneSIG pertenecientes al centro de producción Geoinformática y Señales Digitales de la Universidad de las Ciencias Informáticas desarrollan actualmente sistemas de gran impacto para la sociedad, que contribuyen a las actividades de Video Vigilancia y representación de información geográfica, evidenciado en las aplicaciones Xilema Suria y SIG-UCI respectivamente. Estos sistemas actualmente a pesar de pertenecer a un mismo centro no presentan una forma de integración para aprovechar las ventajas de cada uno. En esta investigación se realizó el diseño, implementación y prueba de un Módulo de Información Geográfica para la Representación Espacial de Cámaras para Xilema Suria. Durante el desarrollo de la misma se patentizó que un módulo tal puede contribuir a aumentar la eficiencia de la gestión de las cámaras en sistemas de seguridad de Video Vigilancia y con ello garantizar una mejor protección a las áreas donde sea desplegado. Se enfatizó en los puntos ciegos u oscuros, zonas críticas, área barrida por las cámaras, obstáculos para la visión, representación en el espacio del campo de visión de las cámaras y observación del área geográfica donde se desarrolló un suceso determinado. Tal y como expresa la política del país y de la Universidad, la implementación de la aplicación está basada en tecnologías libres, que eviten el pago de licencias y uso de tecnologías privativas, que representan frenos en el desarrollo.

**Palabras claves:** cámara, georreferenciación, Video Vigilancia.



## ABSTRACT:

Video Surveillance and GeneSIG projects belonging to the production center Geo Informatics and Digital Signals from University of Informatics Sciences currently develop systems with great impact on society, contributing to the activities of video surveillance and representation of geographic information, evidenced in the Xilema Suria and Aplicativo SIG applications respectively. These systems currently despite belonging to the same center does not have an integration way. This research is the realization of the design, implementation and testing of a Geographic Information System for managing a Video Surveillance System. During the development of it has been confirmed that such a system can help increase management efficiency of the cameras in video surveillance security systems to ensure better protection to areas where it is deployed, emphasizing the blind or dark spots, critical areas, areas swept by cameras, obstacles to vision, representation in the space of sight space of the cameras, observing the geographic area where a particular event took place, among others. For developing it various methods and techniques that contributed to the theoretical and practical support system created are used. As expresses the policy of the country and the University, the application deployment is based on free technologies that avoid paying licensing and use of proprietary technologies, which represent brakes.

**Key words:** camera, video surveillance, georreferenciación.

## ÍNDICE

INTRODUCCIÓN .....	1
Capítulo 1. FUNDAMENTACION TEÓRICA .....	5
1.1. Introducción .....	5
1.2. Conceptos asociados al dominio del problema .....	5
1.2.1. Objeto espacial .....	5
1.2.2. Georreferenciación .....	5
1.2.3. Objeto bloqueante .....	5
1.2.4. Punto .....	6
1.2.5. Polígono .....	6
1.2.6. Línea .....	6
1.3. Descripción general del objeto de estudio de la investigación.....	6
1.4. Análisis de soluciones existentes.....	8
1.5. Herramientas, metodologías y tecnologías a utilizar para el desarrollo de la solución .....	10
1.5.1. Metodología de desarrollo .....	11
1.5.2. Rational Unified Process (RUP).....	11
1.5.3. Lenguaje de Modelado: UML.....	11
1.5.4. Herramienta de Modelado: Visual Paradigm.....	12
1.5.5. Lenguaje de programación .....	12
1.5.6. Framework y Entorno de Desarrollo Integrado (IDE) .....	12
1.6. Conclusiones parciales .....	13
Capítulo 2. ANÁLIS Y DISEÑO .....	14
2.1. Introducción .....	14
2.2. Modelo de Dominio .....	14
2.3. Requerimientos Funcionales.....	15
2.4. Requerimientos no Funcionales.....	19
2.5. Descripción del Sistema .....	21
2.5.1. Definición de los actores.....	21
2.5.2. Listado de los Casos de Uso. ....	21
2.5.3. Patrones de Caso de Uso.....	21
2.5.4. Diagrama de Casos de Uso del Sistema.....	22
2.5.5. Expansión de los Casos de Uso. ....	23



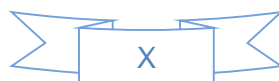
2.6. Patrones .....	40
2.6.1. Patrón Arquitectónico.....	40
2.6.2. Patrones de Diseño .....	41
2.7. Diagramas de Clase del Diseño.....	43
2.8. Utilización del Modelo de Diseño .....	43
2.9. Conclusiones Parciales.....	44
Capítulo 3. IMPLEMENTACIÓN Y PRUEBAS .....	45
3.1. Introducción .....	45
3.2. Modelo de Implementación.....	45
3.2.1. Diagramas de Despliegue.....	45
3.2.2. Diagramas de Componentes del Código Fuente.....	46
3.3. Pruebas .....	46
3.3.1. Diseño de Prueba de Caja Negra sobre MIGREC-XS.....	46
3.3.2. Resultados Obtenidos.....	49
3.3.3. Pruebas de Integración.....	50
3.4. Conclusiones Parciales.....	50
CONCLUSIONES GENERALES.....	51
RECOMENDACIONES .....	53
REFERENCIAS BIBLIOGRÁFICAS .....	54
BIBLIOGRAFÍAS CONSULTAS: .....	57

## ÍNDICE DE TABLAS:

Tabla. 1:	Análisis del uso de las soluciones existentes.....	10
Tabla. 1:	Definición del Actor del Sistema .....	21
Tabla. 2:	Listado de Casos de Uso.....	21
Tabla. 3:	Descripción Textual del Caso de Uso Conectarse al Módulo Gestor del Sistema Silema Suria. 23	
Tabla. 5:	Descripción Textual del Caso de Uso Autenticar Usuario. ....	25
Tabla. 6:	Descripción Textual del Caso de Uso Gestionar Capas.....	26
Tabla. 7:	Descripción Textual del Caso de Uso Manipular Mapa.....	29
Tabla. 9:	Descripción Textual del Caso de Uso Mostar la distribución de cámaras por zonas.....	32
Tabla. 10:	Descripción Textual del Caso de Uso Gestionar Cámaras.....	32
Tabla. 11:	Descripción Textual del Caso de Uso Manipular atributos de la cámara.....	35
Tabla. 12:	Descripción Textual del Caso de Uso Mostrar flujo de video captado por la cámara.....	37
Tabla. 13:	Descripción Textual del Caso de Uso Mostrar el área de influencia.....	37
Tabla. 14:	Caso de Prueba de CU. Gestionar cámara (Insertar) .....	47
Tabla. 15:	Caso de Prueba de CU. Gestionar cámara. (Modificar) .....	47
Tabla. 16:	Caso de Prueba de CU. Gestionar cámara. (Eliminar).....	48
Tabla. 17:	Caso de Prueba de CU. Gestionar cámara. (Visualizar) .....	49

## ÍNDICE DE FIGURAS:

Fig. 1:	Estructura de los SIG.....	7
Fig. 2:	Abanico de visión de la cámara. ....	7
Fig. 3:	Diagrama de Modelo de Dominio.....	15
Fig. 4:	Diagrama de Caso de Uso del Sistema .....	23
Fig. 5:	Arquitectura del Sistema.....	41
Fig. 6:	Diagrama de Clases del Diseño.....	43
Fig. 7:	Diagrama de Despliegue de MIGREC-XS.....	45
Fig. 8:	Gráfico de pruebas realizadas al MIGREC-XS.....	49



## INTRODUCCIÓN

En la actualidad la humanidad está colmada de diferentes tipos de amenazas y estas se extienden hasta los hogares, negocios o empresas. Por esta razón son diversos los sistemas empleados para Video Vigilancia, los cuales son conocidos como Sistemas de Circuito Cerrado de Televisión (CCTV). Estos sistemas han ganado una gran popularidad y efectividad debido a las funciones que realizan, tales como controladores de acceso, programas de reconocimiento facial, sensores de movimiento y guardias de seguridad. (Accesor Applications and Services, 2014) Funciones que han tomado mayor desarrollo y alcance con el transcurso del tiempo y los avances tecnológicos. Con la instalación y puesta en marcha de un CCTV no solo es posible mantener un mayor control sobre los sucesos significativos (actividades delictivas o fuera de lo correcto), sino que permiten contener y prevenir cualquier actividad que se desarrolle en los lugares donde estén implantados. (Novenca Security Systems, 2013)

Según (Salomón, 2004) la importancia y eficiencia de estos sistemas es muy alta, según arrojan las encuestas realizadas a especialistas en el tema de la seguridad y protección. Sin embargo son diversos los programas desarrollados en el mundo como por ejemplo las aplicaciones Ispy Connect y Axxon Next que ya integran la Video Vigilancia con los Sistemas de Información Geográfica (SIG). Los SIG, permiten representar información geográficamente referenciada, así como el manejo de mapas y planos, los cuales constituyen una herramienta muy poderosa en diferentes esferas de la sociedad. Estos no solo permiten representar territorios, puntos, líneas, lugares de interés u obras arquitectónicas realizadas por el hombre, sino que también sus utilidades pueden ser vinculadas a diversos sistemas informatizados (FAO, 2012). Aprovechando estas ventajas se hace posible mediante ellos georreferenciar las cámaras y otros dispositivos de seguridad pertenecientes a un Sistema de Video Vigilancia. De esta forma el operador que lo utilice conoce de una manera más detallada y explícita el ámbito en el que están ubicados cada uno de los elementos correspondientes a la seguridad, lo que facilita el análisis de cada uno de los sucesos ocurridos de una forma más clara.

En Cuba son muchos los esfuerzos realizados para lograr alcanzar un desarrollo tecnológico a la par de los existentes a nivel mundial, ya que es evidente que se cuenta con muchas instituciones que necesitan ser protegidas por sus intereses económicos, políticos y sociales. Para lograr esto, primeramente fue necesaria la contratación de servicios extranjeros debido a la poca experiencia que había en el tema en el país (DATYS, 2005). Luego como una estrategia de la Revolución, el Comandante en Jefe Fidel Castro Ruz decide crear la Universidad de las Ciencias Informáticas (UCI), en la que vincularía el estudio, la investigación y la producción, en aras de garantizar el desarrollo de aplicaciones informáticas en el país y con esto lograr formar profesionales comprometidos con la Patria que contribuyeran al desarrollo

tecnológico y a la sustitución de importaciones.

En la Facultad 6 de la UCI, se encuentra el Centro de Desarrollo Geoinformática y Señales Digitales (GEySED). En cuyo departamento existe un proyecto productivo llamado VVS (VVS), que trabaja en la aplicación Xilema Suria (XS), un sistema encaminado al proceso de Video Vigilancia; el cual permite fortalecer la seguridad en cualquier institución donde sea desplegado mediante un CCTV. Por las políticas del país y de la Universidad este proyecto migró totalmente a tecnologías libres e incluye la infraestructura tecnológica para soportar un conjunto de cámaras de seguridad, que se puedan gestionar dentro de una red de datos asegurando la visualización, almacenamiento y transmisión de los flujos de videos generados en cada uno de los dispositivos de adquisición.

En estos momentos el país cuenta con varios lugares públicos como hoteles, centros comerciales, plazas, parques, avenidas que tienen instalados estos servicios para mantener el orden con mayor eficiencia, detectando por esta vía cualquier infracción cometida. Ejemplo de esto se evidencia en la UCI donde se tiene instaurado un CCTV, para la constante vigilancia. Pero a pesar de que la Universidad y específicamente la Facultad 6 cuenta con este proyecto sus ventajas se ven limitadas al posicionar o direccionar las cámaras hacia lugares donde los elementos de su alrededor obstruyen su completa visión, dejando de esta forma puntos ciegos u oscuros en las áreas vigiladas. Los especialistas encargados de la manipulación de los dispositivos de seguridad no tienen representados geográficamente donde están ubicados cada uno de estos, lo que limita el apoyo visual. Por otra parte solo es posible analizar los hechos desde la perspectiva de la cámara y por tanto se deja de aprovechar toda la información que proporciona el terreno para la interpretación y análisis de cada suceso significativo y, por tanto, la seguridad.

La situación descrita anteriormente conlleva a plantearse el siguiente **problema a resolver**: ¿Cómo localizar la distribución de cámaras y su área de influencia en el Sistema Xilema Suria? Para darle solución al problema ante planteado se define como **objeto de estudio**: la georreferenciación de objetos en un mapa, enmarcado en el **campo de acción**: la georreferenciación en un mapa de las cámaras del Sistema de VVS.

Para limitar el alcance de la investigación y darle solución al problema planteado se concreta como **objetivo general**: Desarrollar un Módulo de Información Geográfica para la representación espacial de las cámaras en el Sistema Xilema Suria.

Para lograr el cumplimiento de estos objetivos se realizarán las siguientes **tareas investigativas**:

- ✚ Caracterización de las técnicas existentes para la georreferenciación de objetos.
- ✚ Selección de la metodología, tecnologías y herramientas para el desarrollo del MIGREC-XS.

- ✚ Identificación de los requisitos funcionales y no funcionales que debe cumplir el Sistema de Información Geográfica aplicado a VVS para definir las características que debe cumplir.
- ✚ Análisis y diseño del MIGREC-XS para determinar un hilo conductor para el desarrollo de la solución.
- ✚ Validación del correcto funcionamiento de las funcionalidades desarrolladas al MIGREC-XS para garantizar que la solución cumple con todos los requisitos identificados.

Los métodos de la investigación científica utilizados en el presente trabajo se muestran y explican a continuación:

## ➤ Métodos Teóricos

Con la utilización de estos métodos es posible estudiar las características esenciales del trabajo y tecnologías de cada uno de los proyectos implicados en la integración.

- ✚ **Histórico – Lógico:** Es aplicado para el análisis de los sistemas existentes, empleando un estudio sobre su evolución y desarrollo, en aras de buscar posibles soluciones.
- ✚ **Analítico – Sintético:** Utilizado para el estudio de bibliografías de diferentes autores, referentes a lenguajes de programación, tecnologías utilizadas, componentes principales y poder realizar una amplia investigación sobre los elementos que se relacionan con el objeto de estudio. (Chagoya, 2008)

## ➤ Métodos Empíricos

Estos métodos son los que permiten hacer una retroalimentación de trabajos realizados con anterioridad sobre un tema en específico, en este caso con las posibles propuestas para la integración de los dos proyectos productivos que permitirá aprovechar por parte de Video Vigilancia las ventajas que puede proporcionarle un SIG.

- ✚ **Entrevista:** Se utilizó principalmente en la parte inicial de la investigación, mediante la técnica de tormenta de ideas para determinar los requisitos de la aplicación y aclarar dudas acerca de las tecnologías. Además se utilizó para dar solución a preguntas surgidas durante el transcurso de la investigación sobre el funcionamiento de QGIS (Aplicación SIG).

A continuación se presenta la estructura que seguirá la investigación por capítulos:

**CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA:** Se explican los principales conceptos relacionados con SIG y sus formas de representación de la cartografía. Además se hace un recorrido de la evolución de estas tecnologías y un análisis del estado del arte de los sistemas existentes tanto a nivel nacional como internacional, percibiendo las características, así como las ventajas y desventajas. También se analiza y estudian las teorías recopiladas en la bibliografía y se extraen los elementos más importantes que se

relacionan con el objeto de estudio, donde se hace referencia al ángulo de visión de los dispositivos de seguridad.

**CAPÍTULO 2. ANÁLISIS Y DISEÑO:** En este capítulo se describe la metodología a utilizar para dar solución a la investigación, siendo esta el hilo conductor que guie el desarrollo del proyecto. Se caracterizan las tecnologías y las herramientas en aras de escoger las que brinden mejores prestaciones, teniendo en cuenta las definidas por el proyecto productivo VVS. También se desarrollará el modelo del dominio, así como los diagramas de clase del diseño. Se definirán y especificarán los requisitos tanto funcionales como los no funcionales, así como los casos de uso.

**CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS:** En este capítulo se trabajará en lo referente a la implementación del sistema y en las pruebas a aplicar para garantizar una solución de calidad y que cumpla con todas las especificaciones obtenidas del capítulo anterior. Se mostrarán los datos obtenidos con la aplicación de las pruebas y un resumen de los principales problemas detectados durante el proceso en caso de que hayan existido.

## Capítulo 1. FUNDAMENTACION TEÓRICA

### 1.1. Introducción.

En el presente capítulo se describen los elementos que constituyen el soporte teórico de la investigación. Se definen los principales conceptos relacionados con los SIG y Sistemas de Video Vigilancia. Además se hace un estudio de las principales aplicaciones en el ámbito nacional e internacional que hacen aportes al problema de investigación y que pueden contribuir al desarrollo de la solución. Se describen las herramientas, metodología y tecnologías a utilizar, logrando con esto un punto de partida para los resultados esperados del presente proyecto.

### 1.2. Conceptos asociados al dominio del problema.

Son varios los conceptos que se tienen en la actualidad sobre los SIG, algunos se refieren a sus componentes y funciones, otros a su capacidad para la toma de decisiones, otros a su función como base de datos, pero se puede resumir que un **SIG** se define como un sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. (Bravo, 200). Para la utilización de un SIG se hace necesario analizar los siguientes acápite, que describen algunos de sus componentes y funcionalidades.

#### 1.2.1. Objeto espacial

Es la representación de una información que cuenta con una referencia o localización geográfica y a su vez ésta con un atributo que define su forma geométrica. (Rivera, 2012)

#### 1.2.2. Georreferenciación

Es el posicionamiento de un objeto espacial (representado mediante punto, vector, área, o volumen) en un mapa, dado una proyección geoespacial o sistema de coordenadas y datos determinados. Para georreferenciar un objeto en el mapa es necesario inicialmente obtener el punto que determina su localización, al cual se le asocia un par de coordenadas, con lo que es posible ubicar el objeto en el mapa. (ARC Gis Resorces, 2014)

#### 1.2.3. Objeto bloqueante

Es la representación de un objeto espacial que constituye un agente inhibidor en el ángulo de visión de los dispositivos de seguridad. (Álvarez, 2014)



# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

## 1.2.4. Punto

Se denomina punto a la forma más simple de un símbolo de representación geométrica, el cual brinda la información referente a la localización de elementos en un mapa tales como registros de electricidad, pozos, etc. (Hudiel, 2008).

## 1.2.5. Polígono

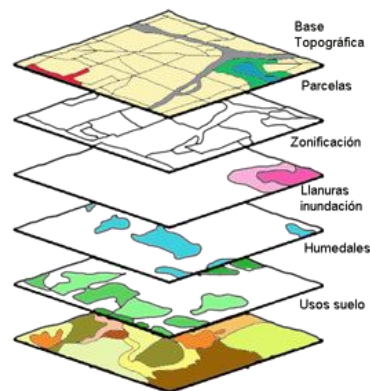
Los polígonos son formas bidimensionales. Están hechos con líneas rectas, y su forma es "cerrada". Todas las líneas están conectadas. Un polígono simple sólo tiene un borde que no se cruza con él mismo, uno complejo se interseca consigo mismo. (MathsIsFun, 2011)

## 1.2.6. Línea

Una línea es una colección de puntos infinitamente delgada, infinitamente larga extendiéndose en dos direcciones opuestas. (HotMath, 2013). Las líneas que se utilizarán en las representaciones geográficas del sistema servirán para mostrar elementos como viales, puertas y delimitaciones de áreas.

## 1.3. Descripción general del objeto de estudio de la investigación

Luego de haber analizado los conceptos asociados a la investigación podemos referirnos a que los SIG son sistemas de software y procedimientos, diseñados para soportar la captura, el manejo, la manipulación, el análisis, el modelado y el despliegue de datos espacialmente referenciados, para la solución de los problemas complejos del manejo y planeamiento territorial. (FAO, 2015) Estos son herramientas que permiten a los usuarios editar datos, mapas, analizar la información espacial, crear consultas interactivas y presentar los resultados de todas estas operaciones. La información en estos sistemas se almacena en capas temáticas que pueden ser de formato Raster o Vectorial. Cada capa agrupa objetos de un concepto determinado, como se puede observar en la Fig. 1. Este modelo implica un proceso de abstracción para obtener la forma final del mapa, sobreponiendo todas las capas de forma ordenada para obtener una sola imagen. (Gabriel Ortiz, 2003)



# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Fig. 1: Estructura de los SIG.

Teniendo en cuenta las utilidades de las representaciones gráficas se utilizarán los planos correspondientes a la institución o departamento donde se encuentre instaurado el sistema de Video Vigilancia para representar numerosas informaciones en ellos, tales como: líneas, que representen paredes, puertas o ventanas, y otros objetos que sean posibles obstáculos en la visión de las cámaras IP o también conocidas como cámaras Web o de Red.

Resulta importante destacar que como aporte fundamental y propósito de la presente investigación la representación de las cámaras IP se realizará en una capa denominada Capa de Cámaras donde se podrá obtener los metadatos y características correspondientes en dependencia de su tipo y marca, tales como:

- Dirección: Sentido de visualización de la cámara.
- Amplitud: Amplitud de barrido de las cámaras.
- Alcance: Alcance de visualización de las cámaras.

Dada las características de una cámara lo más adecuado para mostrar su ángulo de visión es en forma de abanico para lo cual se hace necesario la aplicación de fundamentos trigonométricos como:

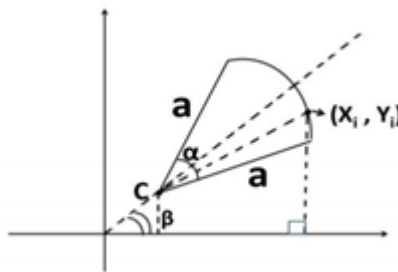


Fig. 2: Abanico de visión de la cámara.

a: alcance  $\alpha$ : amplitud  $\beta$ : dirección

$$X_i = a * \sin ((\beta - \alpha/2) + i)$$

$$Y_i = a * \cos ((\beta - \alpha/2) + i)$$

El ángulo o campo de visión con relación a un objetivo estándar en una cámara estática de 35 mm, expresado en grados, por ejemplo es 30°. En términos prácticos, es el área que puede abarcar un objetivo, donde el ángulo de visión se determina por la longitud focal del objetivo. Un objetivo gran angular tiene una longitud focal corta y abarca un ángulo de visión más amplio que los objetivos estándar o teleobjetivos, que tienen longitudes focales más largas. (Axis Communications, 2014)

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

A continuación se describe el pseudo-código de uno de los algoritmos existentes para la determinación de las coordenadas de los puntos correspondientes al área que ocupa el abanico de visualización de la cámara, el mismo será implementado para dar solución a este problema en la aplicación.

(Milord, 2012)

FUNCIÓN *Find\_point*(VAR a: TipoPuntoCámara): *TipoLista*;

ENTRADA: C objeto cámara

SALIDA: L lista de puntos que conforman el abanico de visualización

INICIO

Como primer punto de la lista tendrá las coordenadas de la cámara.

$i \leftarrow \text{dirección} - \text{amplitud}/2$

HACER DESDE  $i$  HASTA  $\text{dirección} + \text{amplitud}/2$  CON PASO 1

$L \leftarrow (\text{CoordenadaX} + (\text{alcance} * \sin(i)), \text{CoordenadaY} + (\text{alcance} * \cos(i)))$ ;

RETORNAR L;

FIN

La utilización de las ventajas de los SIG proporcionará una alternativa para analizar los hechos no solo desde la perspectiva de la cámara, donde el operador de los sistemas de seguridad podrá utilizar la información que aporta el terreno para la interpretación y análisis de cada suceso significativo y, por tanto, la seguridad.

## 1.4. Análisis de soluciones existentes.

Las posibilidades que ofrecen los sistemas de Video Vigilancia, dependerán en gran medida del nivel de la seguridad que demande el consumidor y de la solución adquirida por estos. A continuación se hace alusión a las aplicaciones y empresas del ámbito nacional e internacional respectivamente que trabajan en este campo, incluyendo es sus productos las ventajas de los SIG que hoy cumplen un papel fundamental en la disminución del tiempo de respuesta a los incidentes y al análisis de los mismos.

En el año 2012 en la UCI se realizó una investigación (Milord, 2012), y como parte de la misma se desarrolló el *plugins GeoVV*, una herramienta para la gestión de Video Vigilancia en GeoQ (Proyecto para la creación

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

de SIG de escritorio en el centro GEySED). Este trabajo está encaminado a mejorar la percepción remota y la confiabilidad que tienen los usuarios de GeoQ de la información de un área geográfica, para lo cual se apoya de Xilema Suria. Esta aplicación contiene funcionalidades como la determinación de puntos ciegos y área barrida por las cámaras. Sin embargo en este caso la Video Vigilancia solo constituye un elemento de apoyo a la información geográfica y no permite utilizarlo para desarrollar un correcto posicionamiento de las cámaras. Este trabajo se tomará como punto de partida para la investigación, haciendo uso de la documentación generada, la que será revisada y evaluada para el desarrollo de la nueva aplicación.

En el mismo año 2012 en la UCI se realizó otra investigación (Quintanilla, 2013), y como parte de la misma se desarrolló el **Componente para la representación espacial de dispositivos de seguridad (CREDS)**. Este es una aplicación de escritorio multiplataforma que se encarga de representar sobre un plano arquitectónico los dispositivos de seguridad registrados por el módulo Visor de la anterior versión del Xilema Suria. Este no permite la integración con la nueva versión del SXS, por no usar las mismas tecnologías, además de que solo permite representar las cámaras sobre un solo plano y no incorpora la adquisición del flujo de video de las cámaras.

**Ispy Connect:** Se puede descargar gratis desde su sitio web. Funciona sobre Windows y es muy fácil de instalar y configurar. Sin embargo, requiere conocimiento de administración de redes para conectar cámaras IP. Se pueden utilizar cámaras USB. Entre muchas características se pueden destacar: control de cámaras PTZ, integración mediante software con otros sistemas, acceso desde internet, notificación vía correo, *twitter* o SMS, detección de movimiento, detección de sonido, acceso desde *tablets* o *smartphones*, conexión con otros servidores, integración con planos y mapas. Existe la opción de una suscripción mensual de bajo costo, lo que permite administrar servidores desde la web del fabricante haciendo la vida mucho más fácil. (Hernandez, 2013)

**Axxon Next:** La empresa Axxon tiene una opción con excelentes capacidades. Se trata de su producto Next, cuya versión gratuita ofrece todas las funcionalidades del software, con las siguientes limitaciones: 1) hasta 1 TB de almacenamiento, 2) solo un servidor y 3) hasta 16 cámaras. La conexión con cámaras IP es muy sencilla. Posee capacidades avanzadas de análisis de video para detección de: movimiento, cruce de línea, objetos abandonados o sustraídos, entrada o salida de zonas y audio. Su sistema de archivo permite realizar un manejo eficiente de los discos mediante “archivos contenedores” de las grabaciones. En cuanto a lo visual, la interfaz de usuario es muy atractiva y fácil de usar y permite usar planos y mapas con vistas 3D. (Hernandez, 2013)

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

El **GV-GIS** es un Sistema de Información Geográfica basada en administración central de dispositivos Geo Visión. El seguimiento del vehículo y verificación remoto de la locación con el GV-GIS es posible instalarlo en el vehículo un Recibidor GPS (Sistema de Posicionamiento Global). Además de la localización en vivo de los vehículos, también se puede monitorear la locación de las cámaras y el estado de dispositivos E/S (Entrada y Salida) del DVR (Grabador de Vídeo Digital) y NVR (Grabador de Vídeo de Red) sin funciones GPS para vigilancia ciudadana. (Geo Vision, 2009)

**Netkrom:** El centro de control es un sistema donde convergen las tecnologías de información y comunicación (GPS, telefonía IP, GIS, Internet, radio, video IP, etc.), que sirven de apoyo a la toma de decisiones en seguridad ciudadana. Utiliza diferentes medios y canales de comunicación como el de telefonía, web, mensajes, radio, video, partes físicos, etc. Los cuales permiten automatizar el registro, tipificación, geo-referencia, seguimiento a las incidencias de inseguridad y mostrar indicadores geo-referenciados a manera de un observatorio de seguridad ciudadana. (Netkrom, 2010)

Tabla. 1: Análisis del uso de las soluciones existentes

Aplicaciones	Open Source	Integrable al entorno	Funcionalidades necesarias	Disponibilidad
GeoVV	SI	SI	Parcial	NO
CREDS	SI	NO	Parcial	NO
Ispy Conect	NO	NO	Total	SI
GV-GIS	NO	NO	Parcial	SI
Netkrom	SI	NO	Parcial	SI
Axxon Next	NO	NO	Parcial	SI

A pesar de los beneficios de las herramientas Axxon Next, El GV-GIS y Netkrom no fue posible su utilización como parte de la solución de la investigación debido a las problemáticas planteadas en la tabla anterior o cuentan con una interfaz que permita la integración con el SXS y no se conoce su código fuente por ser aplicaciones privativas. Por otra parte al ser aplicaciones privativas sus licencias hay que pagarlas propiciando gastos para el país.

## 1.5. Herramientas, metodologías y tecnologías a utilizar para el desarrollo de la solución

Para dar solución a la presente propuesta y con el fin de obtener un producto de software de alta calidad, eficacia y que brinde ventajosas funcionalidades al SXS se definieron las herramientas, metodologías y tecnologías que fueron de mayor utilidad para la implementación. En este caso se utilizarán las mismas que

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

se emplean en el proyecto, con el objetivo de garantizar la integración con los componentes implementados en este.

## **1.5.1. Metodología de desarrollo**

Las Metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software, con independencia de su temática o complejidad. (Vázquez, 2007) Actualmente estas metodologías son una guía en el proceso de desarrollo de las aplicaciones informáticas, permitiendo que se obtengan resultados con la mayor calidad, rapidez y eficiencia posible, para evitar cometer errores futuros.

## **1.5.2. Rational Unified Process (RUP)**

Una de las metodologías pesadas más conocidas y utilizadas es la Metodología RUP, la cual divide el desarrollo en cuatro fases que definen su ciclo de vida:

- ✚ Inicio: El objetivo es determinar la visión del proyecto y definir lo que se desea realizar.
- ✚ Elaboración: Etapa en la que se determina la arquitectura óptima del proyecto.
- ✚ Construcción: Se obtiene la capacidad operacional inicial.
- ✚ Transmisión: Obtener el producto acabado y definido.

Otros de los aspectos que propició se escogiera RUP como la metodología de desarrollo es que presenta dos características fundamentales, la primera es que está guiado por casos de usos que describen las funcionalidades del sistema que el usuario desea obtener, permitiendo la trazabilidad entre los artefactos generados en el proceso de desarrollo. La segunda característica es que esta metodología está centrada en la arquitectura, por lo cual permite una mejor organización y estructura de las partes del sistema, teniendo en cuenta elementos de calidad, rendimiento, reutilización y flexibilidad. (Jacobson, 2000).

## **1.5.3. Lenguaje de Modelado: UML**

El lenguaje unificado de modelado o notación (UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. Es el estándar de descripción de esquemas de software. Puede conectarse con lenguajes de programación (Ingeniería directa e inversa. Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.). Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar. (Larman, 1999)

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

## **1.5.4. Herramienta de Modelado: Visual Paradigm**

Las herramientas CASE (Computer-Aided Software Engineering o Ingeniería de Software Asistida por Computadora) son numerosas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Visual Paradigm v8.0, constituye una herramienta CASE que utiliza UML v8.0 (Lenguaje Unificado de Modelado), como lenguaje de modelado. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite realizar ingeniería tanto directa como inversa, tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la base de datos a partir del esquema de clases. Genera la documentación del proyecto automáticamente en varios formatos como Web y PDF (formato de documento portátil). Permite el control de versiones, exportar las imágenes, es multiplataforma, además es muy sencillo de usar, instalar y actualizar. Visual Paradigm admite compatibilidad con las demás versiones.

## **1.5.5. Lenguaje de programación**

El Lenguaje de Programación C++ es versátil, potente y general. Proviene del lenguaje C original, del cual mantiene sus ventajas en cuanto a riqueza de operadores y expresiones; ha eliminado algunas de las dificultades y limitaciones del mismo. El C++ es un lenguaje imperativo, orientado a objetos, el cual permite que un mismo código fuente puede ser compilado en diversas plataformas. Es muy potente en lo que se refiere a creación de sistemas complejo y actualmente, puede compilar y ejecutar código de C, ya que viene con librerías para realizar esta labor. Además resulta interesante destacar que existen varios tutoriales en línea, libros y códigos fuentes abiertos que permiten hacer el trabajo más satisfactorio y cómodo. (Tecnun, 2004)

## **1.5.6. Framework y Entorno de Desarrollo Integrado (IDE)**

Qt v.5.2.1 es una amplia plataforma de desarrollo que incluye clases, librerías y herramientas para la producción de aplicaciones de interfaz gráfica de usuario sin muchas complicaciones que pueden operar en varias plataformas. Qt5 no es un lenguaje, sino un grupo de clases que pueden ser utilizadas desde algún lenguaje de programación ya definido.

Un IDE es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, puede utilizarse para varios. Los IDE han sido empaquetados como programas de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de Interfaz Gráfica de Usuario (GUI). Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Qt Creator v.3.3, es una plataforma de entorno de desarrollo integrado disponible junto con las bibliotecas Qt. Es distribuido bajo tres tipos de licencias: Qt Commercial Developer License, Qt GNULGPL v. 2.1, Qt GNU GPL v. 3.0, está disponible para las plataformas: Linux, Mac OSX y Windows.

Principales características de Qt Creator:

- ✚ Posee un avanzado editor de código C++.
- ✚ Posee también una GUI integrada y diseñador de formularios.
- ✚ Ayuda sensible al contexto integrado.
- ✚ Depurador visual.
- ✚ Resaltado y auto completado de código (Carrero, Angel, 2010)

## 1.6. Conclusiones parciales

Los SIG son aplicaciones que representan gráficamente un determinado terreno y todas las características de interés del lugar. Los datos son estructurados comúnmente de dos formas, en formato *raster* y en modelos vectoriales, para esta investigación se utilizarán principalmente los modelos vectoriales. Por otra parte se evidenció que la información manejada es dividida por capas posibilitando la combinación de la misma de forma sencilla. En el capítulo también se presentaron diversos fundamentos teóricos que constituyen la base de esta investigación sobre los SIG y sus aplicaciones en la Video Vigilancia. Se analizaron algunas de las principales empresas del mundo en la fabricación de sistemas de este tipo y se realizó un estudio en cuanto a las desventajas, tecnologías y herramientas que serán utilizadas para el desarrollo del software propuesto.



## Capítulo 2. ANÁLISIS Y DISEÑO

### 2.1. Introducción

En este capítulo se caracterizan las herramientas, la metodología y las tecnologías a utilizar para el proceso de desarrollo del software. Se explican las ventajas que brindan y se valora el objetivo general que cumplen, reflejando la importancia de utilizarlas para el desarrollo de la solución propuesta; mediante una descripción detallada se gestionan conceptos y términos necesarios para entender la razón por la cual se eligen y qué consecuencia tiene realizar el software sobre esta base. Se realiza la propuesta del modelo del dominio, describiendo el flujo de relación entre los componentes más importantes y una detallada descripción de las clases presentes. Se especifican los requisitos funcionales y los no funcionales que la aplicación deberá contener y en el modelo del sistema se describe el actor del sistema junto con los diagramas de casos de uso y las descripciones de estos. Además se muestra el diagrama de clases del diseño y los artefactos generados con los diagramas de secuencia y colaboración.

### 2.2. Modelo de Dominio

El modelo de dominio captura los tipos de objetos más importantes en el contexto de un sistema; estos objetos del dominio representan los procedimientos o los eventos que suceden en el entorno en el que trabaja el sistema. Por tanto, para dar solución a la presente propuesta y con el fin de proporcionar a los usuarios, clientes, desarrolladores y otros interesados un vocabulario común se modeló el correspondiente diagrama.

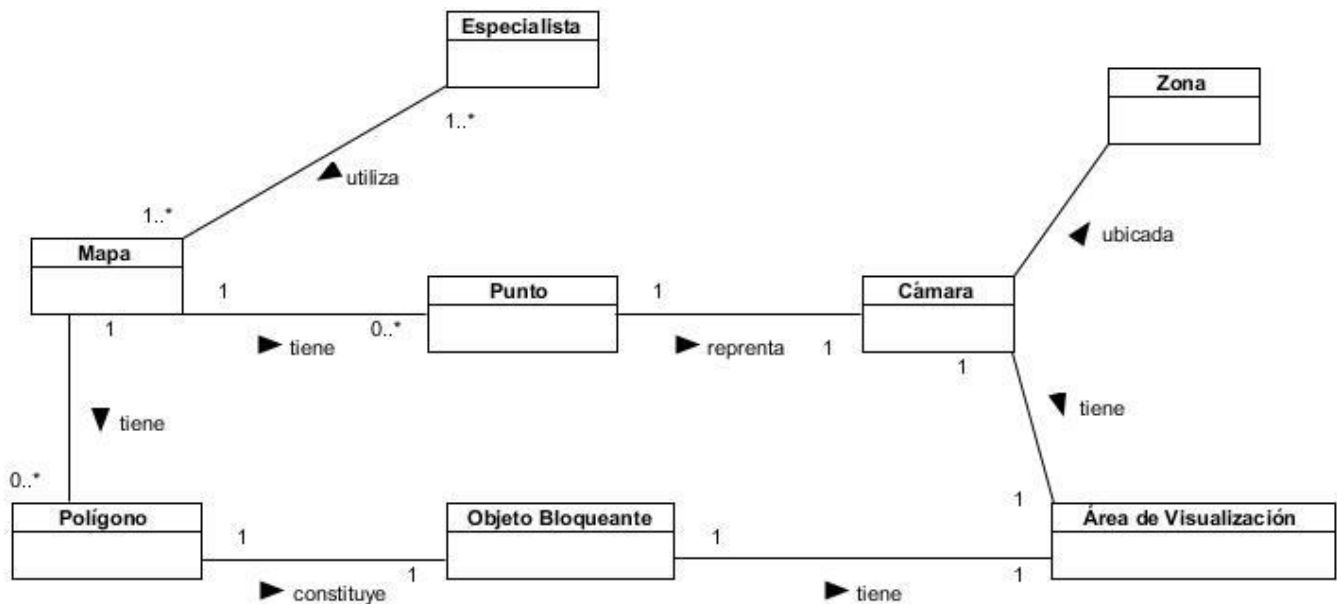


Fig. 3: Diagrama de Modelo de Dominio.

**Especialista** es la persona encargada de utilizar y manipular el **Mapa**, los cuales se componen de la integración de varias capas superpuestas que representan objetos de diferentes conceptos. Estas capas pueden ser de tipo **Punto**, **Línea** o **Polígono**, constituyendo estos **Objetos Bloqueantes** para la visualización de las **Cámaras** que son representadas en el **Mapa** mediante **Puntos**. Estas **Cámaras** a su vez poseen una **Zona** en la cual son ubicadas y al igual que los **Polígonos** invisten un **Área de Visualización** donde en muchas ocasiones una solapa a la otra.

- **Mapa:** Es la representación gráfica de un área determinada de forma vectorial, de imágenes o de formato Raster.
- **Punto:** Son elementos en una cartografía que georreferencian objetos puntuales. Por ejemplo cota fija, altura comprobada, monolito de distancia o nacimiento de un río en una coordenada específica.
- **Polígonos:** Son utilizados para la representación de objetos espaciales como edificaciones, plazas u otros elementos como embalses.
- **Zona:** Representa el nombre de un área del lugar donde está establecido el sistema de Video Vigilancia y a su vez puede contener otras zonas.
- **Cámara:** Objeto tecnológico que obtiene flujos de video desde el lugar donde está instalada y puede ser manejada remotamente.
- **Objeto Bloqueante:** Objeto que impide la completa visión de una cámara y a su vez representa objetos como edificios.
- **Área de Visualización:** Es la porción del mapa que ocupa un objeto bloqueante o agrupa el ángulo de visión de la cámara.

### 2.3. Requerimientos Funcionales

Una vez definido el modelo del dominio se realizó el levantamiento de requisitos. Los requisitos funcionales son la pieza fundamental en un proyecto de desarrollo de software, ya que constituyen la base que permite verificar si se alcanzaron o no los objetivos establecidos en él. Estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema, los cuales ofrecen una descripción completa del comportamiento del sistema desarrollado. Los requisitos funcionales del sistema propuesto son los descritos a continuación:

### **RF1. Conectar el MIGREC-XS al Módulo Gestor del SXS.**

El Sistema debe permitirle al usuario ingresar los datos para establecer la conexión con el módulo Gestor del SXS.

Entrada: IP y puerto.

Salida: Visualizar la ventana para loguearse como usuario del sistema.

### **RF2. Autenticar Usuario.**

El Sistema debe permitirle al usuario autenticarse para ingresar al sistema.

Entrada: Usuario y Contraseña.

Salida: Visualizar la aplicación.

### **RF3. Añadir la(s) capa(s) con la información cartográfica del lugar donde está instaurado el sistema de Video Vigilancia.**

El Sistema debe permitirle al usuario cargar la(s) capa(s) de un mapa que contenga la información cartográfica del lugar donde esta instaurado el sistema.

Entrada: Capa(s) Vectorial(es).

Salida: Capa añadida al área donde se muestran la lista de capas.

### **RF4. Eliminar capa.**

El Sistema debe permitirle al usuario eliminar la capa seleccionada del mapa.

Entrada: Capa(s) Vectorial(es) seleccionada(s).

Salida: Capa(s) eliminada(s) del área donde se muestran la lista de capas y del mapa.

### **RF5. Mostrar un mapa con la información cartográfica del lugar donde está instaurado el sistema de Video Vigilancia.**

El Sistema debe permitirle al usuario mostrar en el área de representación la información cartográfica del lugar donde está instaurado el sistema de Video Vigilancia. Solamente de la capa que está seleccionada.

Entrada: Capa(s) seleccionada(s).

Salida: Visualización del mapa.

### **RF6. Modificar capa.**

El Sistema debe permitirle al usuario modificar la capa seleccionada con la nueva información.

Entrada: Capa Vectorial seleccionada.

Salida: Capa modificada.

### **RF7. Realizar un ZOOM IN del mapa.**

Al ser seleccionada esta opción el sistema debe ser capaz de permitir acercar la vista del mapa en el área de representación hacia la zona determinada por el usuario.

Entrada: Mapa visualizado en el área de representación.

Salida: Modificación de la perspectiva del mapa más cerca a la vista del usuario.

### **RF8. Realizar un ZOOM OUT del mapa.**

Al ser seleccionada esta opción el sistema debe ser capaz de permitir alejar la perspectiva del mapa en el área de representación, para lograr una mayor panorámica de la zona determinada por el usuario.

Entrada: Mapa visualizado en el área de representación.

Salida: Modificación de la perspectiva del mapa más alejado a la vista del usuario.

### **RF9. Mover el mapa en la pantalla.**

El sistema debe permitir desplazar la vista del mapa hacia la zona determinada por el usuario.

Entrada: Mapa visualizado en el área de representación.

Salida: Modificación de la vista del mapa hacia el área determinada por el usuario.

### **RF10. Mostar la distribución de cámaras por zonas.**

El sistema debe permitirle al usuario obtener a través del módulo Gestor del SXS la distribución de las cámaras por zona y visualizarlas en un contenedor

Entrada: Lista con la información de las cámaras por zona.

Salida: Representación de la distribución de las cámaras por zona.

### **RF11. Añadir cámara en el mapa.**

El sistema debe permitirle al usuario añadir la una representación de cámara a una capa de cámaras.

Entrada: Cámara seleccionada.

Salida: Asignarle una posición en una de las Capas de Cámaras.

### **RF12. Eliminar cámara del mapa.**

El sistema debe permitirle al usuario eliminar la representación de la cámara de la Capa de Cámaras.

Entrada: Cámara seleccionada.

Salida: Quitar la referencia de la posición de la cámara en la Capa de Cámaras.

### **RF13. Mostrar en un mapa la posición de la cámara en el lugar donde está instaurado el sistema de Video Vigilancia.**

El Sistema debe permitirle al usuario mostrar en el mapa del lugar donde está instaurado el sistema de Video Vigilancia la representación de las cámaras. Solamente si la Capa de Cámaras está seleccionada.

Entrada: Capa de Cámaras seleccionada.

Salida: Visualización de las cámaras en el mapa.

### **RF14. Modificar en el mapa la posición de la cámara seleccionada.**

El Sistema debe permitirle al usuario modificar en el mapa del lugar donde está instaurado el sistema de Video Vigilancia la representación de las cámaras. Solamente si la Capa de Cámaras está seleccionada.

Entrada: Capa de Cámaras seleccionada.

Salida: Visualización de las cámaras en el mapa.

### **RF15. Visualizar atributos a una cámara.**

El sistema debe permitir mostrar las características de una cámara seleccionada.

Entrada: Objeto con las características de la cámara.

Salida: Mostrar en pantalla un cuadro de diálogo con las características de la cámara seleccionada.

### **RF16. Modificar atributos a una cámara.**

El sistema debe permitirle al usuario modificar las características cámara.

Entrada: Datos nuevos.

Salida: Atributos modificados.

### **RF17. Mostrar el flujo de video captado por una cámara**

Al ser seleccionada la cámara que se quiere visualizar, el sistema debe mostrar una ventana que contenga un reproductor con el flujo de video que está siendo captado por esta.

Entrada: ID de la cámara seleccionada.

Salida: Reproductor con el flujo de video de la cámara seleccionada.

### **RF18. Mostrar el área de influencia de una cámara.**

El sistema debe visualizar el área de influencia de una cámara seleccionada por el usuario teniendo en cuenta los objetos bloqueantes.

Entrada: ID de la cámara seleccionada, capas de objetos bloquenates.

Salida: Representación en el mapa del área de influencia de la cámara.

## **2.4. Requerimientos no Funcionales**

Los requerimientos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema. (Sosa, 2010) A continuación se hace referencia a los requisitos no funcionales tomados para esta solución:

### **Apariencia o interfaz externa:**

RNF1. El componente MIGREC-XS debe diseñarse con una interfaz amigable, ajustándose a la arquitectura de información establecida por el proyecto.

RNF2. Debe poseer colores claros y fácil navegabilidad para los operadores.

### **Usabilidad:**

RNF3. El MIGREC-XS podrá ser usado por cualquier usuario con conocimientos básicos sobre geografía e informática.

RNF4. Debe mostrar mensajes al usuario que le ayuden a llevar a cabo la tarea que realiza.

### **Seguridad:**

RNF5. Para conectarse al servidor el usuario debe conocer el IP y protocolo para realizar la conexión.

RNF6. El consumo de servicios brindados por el Módulo Gestor de XS solo será posible a partir de la correcta conexión del usuario de MIGREC-XS con el Módulo Gestor de XS.

RNF7. Se requiere de un usuario y contraseña para acceder al MIGREC-XS.

RNF8. Es necesario que durante todo el tiempo que se utiliza el MIGREC-XS exista conexión al Módulo Gestor de XS.

### **Software:**

RNF9. Debe estar instaurado en algún puesto de trabajo o servidor el Módulo Gestor de Xilema Suria.

RNF11. Códecs de video para la visualización del flujo de datos extraídos de las cámaras, además de hacer uso de la librería multimedia5 y multimedawidget.

RNF12. Instalar la librería zeromq para garantizar la comunicación.

RNF13. El MIGREC-XS debe utilizarse desde cualquier Linux, Windows o IOS.

### **Rendimiento:**

RNF14. El MIGREC-XS debe consumir como máximo 200 Mb de RAM.

### **Hardware:**

RNF14. El MIGREC-XS necesita un 1 Gb de RAM y un procesador Pentium 4, mayor de 1.8 GHz.

### 2.5. Descripción del Sistema

#### 2.5.1. Definición de los actores

Un Actor del sistema es quien interviene de alguna forma en algunos de los casos de uso, en dependencia del rol que tenga este en el sistema. En la presente investigación solamente existe un solo, el cual responde al nombre de **Operador del Representador Geográfico (Operador RG)**.

Tabla. 1: Definición del Actor del Sistema

Actores	Justificación
Operador RG	Es el encargado de realizar e inicializar todos los procesos del sistema.

#### 2.5.2. Listado de los Casos de Uso.

Los Casos de Usos (CU) que se relacionan a continuación son el resultado de la agrupación de los requisitos funcionales antes señalados. Teniendo en cuenta para esto la utilización de diversos patrones que permiten organizar y sintetizar el trabajo en actividades que dependen unas de otras.

Tabla. 2: Listado de Casos de Uso

Caso de Uso del Sistema	Prioridad
Conectar el MIGREC-XS al Módulo Gestor del SXS	Crítica
Autenticar Usuario	Crítica
Gestionar Capas	Crítica
Manipular Mapa	Crítica
Mostrar la distribución de cámaras por zonas	Crítica
Gestionar Cámara	Crítica
Manipular atributos de la cámara	Crítica
Mostrar flujo de video captado por la cámara	Crítica
Mostrar el área de influencia	Crítica

#### 2.5.3. Patrones de Caso de Uso

La experiencia en la utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con más precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Dado un contexto y un problema a resolver, estas técnicas han mostrado ser la solución adoptada en la comunidad del desarrollo de software. Estos patrones se enfocan hacia el diseño y las técnicas utilizadas en modelos de alta calidad, y no en cómo modelar usos específicos. Utilizando estos



patrones, arquitectos, analistas, ingenieros, y gerentes pueden lograr mejores resultados de forma más rápida. (Jacobson, 2000) En este caso se hizo uso de dos patrones, con sus respectivas especificaciones:

- ✚ Concordancia mediante Reuso y Adición. los cuales representan una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado. Ejemplo de estos se evidencian en el CU Autenticar usuario, que necesita de que primeramente el MIGREC-XS se conecte al módulo Gestor SXS.
- ✚ CRUD Total, el cual se basa en la fusión de casos de usos simple para formar una unidad conceptual. Este se evidencia en el CU Gestionar Cámaras.

### 2.5.4. Diagrama de Casos de Uso del Sistema

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. A continuación se muestra el diagrama de Caso de Uso correspondiente a este sistema.

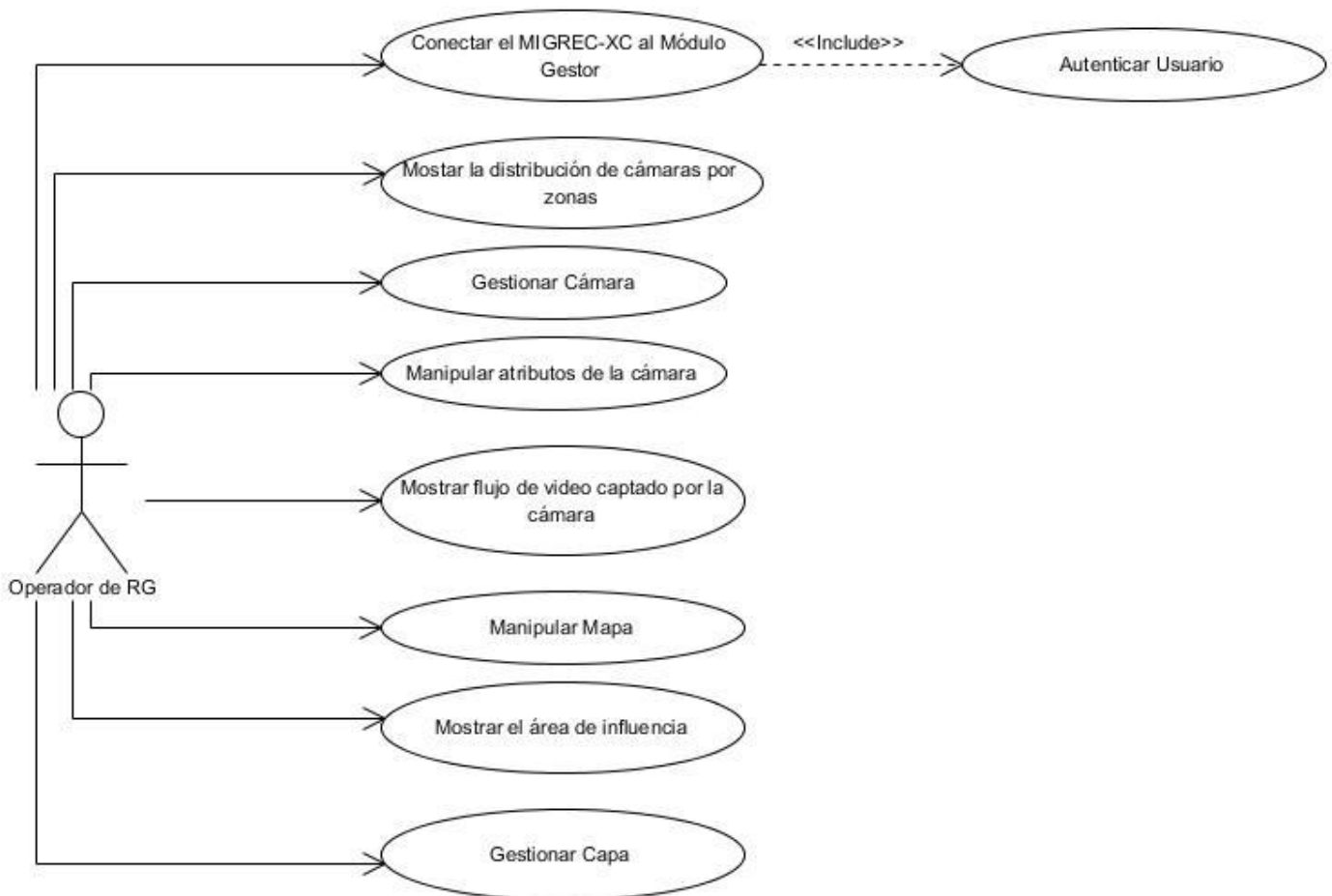


Fig. 4: Diagrama de Caso de Uso del Sistema

### 2.5.5. Expansión de los Casos de Uso.

Tabla. 3: Descripción Textual del Caso de Uso Conectarse al Módulo Gestor del Sistema Silema Suria.

<b>CU1</b>	Conectarse al Módulo Gestor del Sistema Silema Suria	
<b>Objetivo</b>	Iniciar la conexión con el Módulo Gestor del Sistema Silema Suria	
<b>Actores</b>	Operador RG	
<b>Resumen</b>	El Sistema debe permitirle al usuario ingresar los datos para establecer la conexión con el módulo Gestor del SXS	
<b>Referencias</b>	RF1	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Crítica.	
<b>Precondiciones</b>		
<b>Postcondiciones</b>	El sistema se conecta al Módulo Gestor del Sistema Silema Suria permitiendo usar sus funcionalidades.	
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Conexión disponible.		
	<b>Actor</b>	<b>Sistema</b>
1.	Abre la aplicación	
2.		Muestra una nueva ventana introducir los datos de la conexión al Gestor
3.	Ingresa IP y Puerto. Hace clic en el botón Aceptar	
4.		Muestra una nueva ventana para la autenticación del usuario. Termina el CU.
3.1	Hace click en "Cancelar" y termina el Caso de Uso.	

Flujos alternos. No hay conexión disponible.		
	Actor	Sistema
4.1		Muestra un mensaje de error con opciones para Cancelar o Intentar otra Conexión.
5.	Hace click en "Intentar otra conexión"	
6.		Vuelve a la paso 3 del Flujo Básico.
5.1	Hace clic en "Cancelar" y termina el Caso de Uso.	
Relaciones	CU Incluidos	No procede.
	CU Extendidos	No procede.

Tabla. 5: Descripción Textual del Caso de Uso Autenticar Usuario.

<b>CU2</b>	Autenticar Usuario	
<b>Objetivo</b>	Acceder a la aplicación con los permisos requeridos y tomar los datos necesarios de la base de datos del SXS.	
<b>Actores</b>	Operador RG	
<b>Resumen</b>	El Sistema debe permitirle al usuario autenticarse para ingresar al sistema.	
<b>Referencias</b>	RF2	
<b>Complejidad</b>	Alta.	
<b>Prioridad</b>	Crítica.	
<b>Precondiciones</b>	Debe de existir conexión con el Módulo Gestor del Sistema Silema Suria.	
<b>Postcondiciones</b>	Muestra la aplicación con la lista de cámara cargadas y los datos del proyecto existente, y en caso de que no halla ninguno, crea uno nuevo	
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Conexión disponible.		
	<b>Actor</b>	<b>Sistema</b>
1.	Ingresa Usuario y Contraseña. Hace click en el botón "Aceptar"	
2		Muestra la aplicación con la lista de cámara cargadas y los datos del proyecto existente, y en caso de que no halla ninguno, crea uno nuevo. Termina el CU.
1.1	Hace click en "Cancelar" y termina CU.	
<b>Flujos alternos.</b> No son correctos los datos.		
	<b>Actor</b>	<b>Sistema</b>
4.1		Muestra un mensaje de error con un botón Aceptar.
5.	Hace click en "Aceptar"	

6.		Vuelve a la paso 1 del Flujo de Básico.
<b>Relaciones</b>	<b>CU Incluidos</b>	CU1
	<b>CU Extendidos</b>	No procede.

Tabla. 6: Descripción Textual del Caso de Uso Gestionar Capas.

<b>CU3</b>	<b>Gestionar Capas</b>	
<b>Objetivo</b>	Realizar operaciones con las capas	
<b>Actores</b>	Operador RG	
<b>Resumen</b>	El Sistema debe permitirle al usuario Añadir, Eliminar, Modificar y Visualizar las Capas.	
<b>Referencias</b>	RF3, RF4, RF5, RF6	
<b>Complejidad</b>	Alta.	
<b>Prioridad</b>	Crítica.	
<b>Precondiciones</b>	Para Modificar y Eliminar: Capa Seleccionada. Para Visualizar: Exista alguna Capa añadida.	
<b>Postcondiciones</b>		
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Sección 1: Añadir Capa Vectorial		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Capa". Luego en "Añadir Capa Vectorial"	
2.		Muestra una ventana para introducir la fuente de la Capa Vectorial.
3.	Escoge la opción "Archivo" o "directorio", define la ruta haciendo	

## CAPÍTULO 2. ANÁLISIS Y DISEÑO

	clic en el botón "Explorar" y da click en "Abrir"	
4.		Carga las capas en el Panel de Capas. Termina el CU.
3.1	Escoge la opción "Base de Datos".	
4.1		Muestra una ventana para introducir los datos de la Base de
5.1	Ingresa los datos	
6.1		Carga las capas en el Panel de Capas. Termina el CU.
3.2	Escoge la opción "Protocolo". define la URI de Protocolo y da click en "Abrir"	
4.2		Carga las capas en el Panel de Capas. Termina el CU.
<b>Flujos alternos.</b> Datos incorrectos o espacios en Blanco		
	<b>Actor</b>	<b>Sistema</b>
6.1.1		Muestra un mensaje de error: Fuente de Entrada no Valida.
4.2.2		Muestra un mensaje de error: Fuente de Entrada no Valida.
<b>Flujos alternos.</b> Dirección de Archivo o Directorio incorrecta		
4a		Muestra un mensaje de error: Fuente de Entrada no Valida.
<b>Flujo básico:</b> Sección 2: Crear Capa		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Capa". Luego en "Crear Capa", y después en "Nueva capa de archivo shape"	
2.		Muestra una ventana para escoger el tipo de Capa Vectorial.

3.	Escoge la opción "Punto", "Línea" o "Polígono", y da clic en "Aceptar"	
4.		Muestra una ventana para poner el nombre y la URL donde se va a guardar la Capa Vectorial.
5.	Escribe el nombre, escoge la URL y da click en el botón "Aceptar"	
6.		Carga la nueva capa en el Panel de Capas. Termina el CU.
<b>Flujos alternos.</b> Cancelar la Operación		
	<b>Actor</b>	<b>Sistema</b>
5.1	Da click en "Cancelar" y termina el CU	
<b>Flujo básico:</b> Sección 3: Eliminar Capa		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Capa". Luego en "Eliminar Capa"	
2.		Muestra el Panel de Capas sin la Capa Vectorial seleccionada. Termina el CU
<b>Flujos alternos.</b> No hay capas seleccionadas.		
	<b>Actor</b>	<b>Sistema</b>
2.1		Muestra un mensaje de error: Fuente de Entrada no Valida.
<b>Flujo básico:</b> Sección 4: Crear Capa de Cámaras		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Capa". Luego en "Crear Capa de Cámara".	

2.		Carga la nueva capa en el Panel de Capas. Termina el CU.
<b>Flujo básico:</b> Sección 5: Modificar Capa Vectorial		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Capa". Luego en "Conmutar Edición"	
2.		Muestra el ícono de edición marcado, indicando que está lista para modificarse y habilita las opciones para insertar, eliminar y modificar los objetos de la capa.
3.	Escoge la opción deseada y modifica el contenido de la Capa	
4.		Muestra los cambios en el Área de Visualización. Termina el CU.
<b>Flujos alternos.</b> No hay capas seleccionadas.		
	<b>Actor</b>	<b>Sistema</b>
2.1		Muestra un mensaje de error: Fuente de Entrada no Valida.
<b>Relaciones</b>	<b>CU Incluidos</b>	No procede.
	<b>CU Extendidos</b>	No procede.

Tabla. 7: Descripción Textual del Caso de Uso Manipular Mapa.

<b>CU5</b>	Manipular Mapa
<b>Objetivo</b>	Visualizar el área del mapa deseada por el usuario
<b>Actores</b>	Operador RG
<b>Resumen</b>	El Sistema debe permitirle al usuario Acercar, Alejar y mover el Mapa, hacia cualquier zona
<b>Referencias</b>	RF8, RF9, R10



<b>Complejidad</b>	Media
<b>Prioridad</b>	Crítica.
<b>Precondiciones</b>	Tener activa alguna Capa en el Área de Visualización.
<b>Postcondiciones</b>	Mapa modificado de posición.

### Flujo de eventos

#### Flujo básico: Sección 1: Acercar el Mapa

	Actor	Sistema
1.	Hace click en el menú "Ver". Luego en "Acercar zum"	
2.		Muestra el ícono de "Acercar zum" marcado, indicando que está listo para usarse y habilita las opciones para insertar, eliminar y modificar los objetos de la capa.
3.	Hace click sobre la zona del mapa que quiere Acercar.	
4.		Muestra el mapa con la modificación visual requerida. Termina el CU.

#### Flujos alternos. Clic en zona equivocada

	Actor	Sistema
3.1	Hace click fuera del Mapa.	
4.1		No muestra ningún cambio.

#### Flujo básico: Sección 2: Alejar Mapa

	Actor	Sistema
1.	Hace click en el menú "Ver". Luego en "Alejar zum"	
2.		Muestra el ícono de "Alejar zum" marcado, indicando que está listo para usarse y habilita las opciones para insertar, eliminar y modificar los objetos de la capa.

3.	Hace click sobre la zona del mapa que quiere Alejar.	
4.		Muestra el mapa con la modificación visual requerida. Termina el CU.

**Flujos alternos. Cancelar la Operación**

	Actor	Sistema
3.1	Hace clic fuera del Mapa.	
4.1		No muestra ningún cambio.

**Flujo básico: Sección 3: Mover Mapa**

	Actor	Sistema
1.	Hace click en el menú "Ver". Luego en "Desplazar Mapa"	
2.		Muestra el ícono de "Desplazar Mapa" marcado, indicando que está listo para usarse y habilita las opciones para insertar, eliminar y modificar los objetos de la capa.
3.	Hace click sobre la zona del mapa que quiere mover y desplaza el mouse con el clic sostenido hacia la zona donde quiera desplazar la selección.	
4.		Muestra el mapa con la modificación visual requerida. Termina el CU.

**Flujos alternos. No hay capas seleccionadas.**

	Actor	Sistema
3.1	Hace click fuera del Mapa.	
4.1		No muestra ningún cambio.

<b>Relaciones</b>	<b>CU Incluidos</b>	No procede.
	<b>CU Extendidos</b>	No procede.

Tabla. 8: Descripción Textual del Caso de Uso Mostar la distribución de cámaras por zonas.

<b>CU6</b>	Mostar la distribución de cámaras por zonas	
<b>Objetivo</b>	Mostrar la organización de la cámaras en forma de árbol	
<b>Actores</b>	Operador RG	
<b>Resumen</b>	El Sistema debe permitirle al usuario conocer la organización de la cámaras en forma de árbol, permitiendo ver a la zona que pertenecen	
<b>Referencias</b>	RF11	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítica.	
<b>Precondiciones</b>	Usuario y contraseña verificados.	
<b>Postcondiciones</b>	Muestra la distribución de cámara por zona.	
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Conexión disponible.		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un Panel con una estructura de árbol, describiendo la distribución de cámaras por zona. Termina el CU
<b>Flujos alternos.</b> Base de Datos vacía.		
	<b>Actor</b>	<b>Sistema</b>
1.1		Muestra un Panel vacío. Termina el CU
<b>Relaciones</b>	<b>CU Incluidos</b>	CU 2
	<b>CU Extendidos</b>	No procede.

Tabla. 9: Descripción Textual del Caso de Uso Gestionar Cámaras.

<b>CU8</b>	Gestionar Cámaras	
<b>Objetivo</b>	Realizar operaciones con las cámaras	
<b>Actores</b>	Operador RG	
<b>Resumen</b>	El Sistema debe permitirle al usuario Añadir, Eliminar, Modificar y Visualizar las cámaras en el mapa.	
<b>Referencias</b>	RF13, RF14, RF15, RF16	
<b>Complejidad</b>	Alta.	
<b>Prioridad</b>	Crítica.	
<b>Precondiciones</b>	<p>Para Modificar y Eliminar: Capa de Cámara seleccionada, Herramienta Conmutar edición seleccionada, Cámara seleccionada.</p> <p>Para Visualizar: Exista alguna cámara añadida.</p> <p>Para Añadir: Exista alguna Capa de Cámara seleccionada, Herramienta Conmutar edición seleccionada.</p>	
<b>Postcondiciones</b>		
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Sección 1: Añadir Cámara		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click sobre una de las cámaras del Panel de Cámaras y la arrastra hacia la zona de visualización.	
2.		Le pide al usuario de un clic hacia el lugar donde está mirando la cámara.
3.	Hace click	
4.		Muestra la representación de la nueva cámara en el mapa.
<b>Flujos alternos.</b> Click en área incorrecta		

## CAPÍTULO 2. ANÁLISIS Y DISEÑO

4.1		Muestra un mensaje de error: "No es un área valida". Cancela la inserción y termina el CU.
<b>Flujos alternos.</b> No hay capa de cámara seleccionada.		
2.1		Muestra un mensaje de error: "No hay ninguna capa de cámara seleccionada". Termina el CU.
<b>Flujos alternos.</b> La edición de capa no está activada.		
2.1		Muestra un mensaje de error: "No está activada la edición de capas". Termina el CU.
<b>Flujo básico:</b> Sección 2: Modificar posición de la cámara		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Capa". Luego en "Conmutar edición"	
2.	Hace click en el menú "Edición". Luego en "Mover objetos espaciales"	
3.	Hace click sobre la cámara seleccionada y la arrastra hacia la nueva posición	
4.		Muestra el mapa con la posición de la cámara modificada. Termina el CU.
2.1	Hace click en una de las cámaras del Panel de cámaras y la arrastra hacia el mapa	
3.1		Muestra el mapa con la posición de la cámara modificada. Termina el CU.
<b>Flujo básico:</b> Sección 3: Eliminar Capa		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Capa". Luego en "Conmutar edición"	

2.	Hace click en el menú "Edición". Luego en "Borrar lo seleccionado"	
3.		Muestra el mapa con la representación de la cámara eliminada. Termina el CU.
<b>Flujo básico:</b> Sección 4: Visualizar Cámara		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el "Panel de Capas" sobre una Capa de Cámara, para visualizarla.	
2.		Muestra la representación de las cámaras en el mapa. Termina el CU.
<b>Flujos alternos.</b> No existen cámaras en esa capa seleccionada		
	<b>Actor</b>	<b>Sistema</b>
2.1		No muestra ningún cambio en el mapa.
<b>Relaciones</b>	<b>CU Incluidos</b>	No procede.
	<b>CU Extendidos</b>	No procede.

Tabla. 10: Descripción Textual del Caso de Uso Manipular atributos de la cámara.

<b>CU9</b>	Manipular atributos de la cámara
<b>Objetivo</b>	Obtener los atributos de una cámara y modificarlos.
<b>Actores</b>	Operador RG
<b>Resumen</b>	El Sistema debe permitirle al usuario visualizar y modificar los atributos de una cámara.
<b>Referencias</b>	RF17, RF18
<b>Complejidad</b>	Media.
<b>Prioridad</b>	Crítica.

<b>Precondiciones</b>	Para Modificar Atributos: Capa de Cámara seleccionada. Para Visualizar: Exista alguna cámara añadida, Capa de Cámara seleccionada.	
<b>Postcondiciones</b>		
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Sección 1: Visualizar atributos		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Ver". Luego en "Identificar objetos espaciales"	
2.		Muestra el ícono de " Identificar objetos espaciales" marcado, indicando que está listo para usarse.
3.	Hace click sobre la representación de la cámara mapa.	
4.		Muestra una ventana con los datos del objeto seleccionado, un botón "Aceptar" y otro de "Cancelar". Termina el CU.
<b>Flujos alternos.</b> La capa de cámara no está seleccionada.		
	<b>Actor</b>	<b>Sistema</b>
4.1		No muestra ningún cambio.
<b>Flujo básico:</b> Sección 2: Modificar atributos		
1.	Hace los pasos de Sección 1	
2.	Modifica los datos mostrados con los nuevos valores y hace clic sobre el botón "Aceptar"	
3.		Cambia los valores. Termina el CU.
2.1	Hace click en "Cancelar".	
3.1		No hace ningún cambio de los valores. Termina el CU.
<b>Relaciones</b>	<b>CU Incluidos</b>	No procede.

	<b>CU Extendidos</b>	No procede.
--	----------------------	-------------

Tabla. 11: Descripción Textual del Caso de Uso Mostrar flujo de video captado por la cámara.

<b>CU10</b>	Mostrar flujo de video captado por la cámara	
<b>Objetivo</b>	Obtener la perspectiva de la cámara.	
<b>Actores</b>	Operador RG	
<b>Resumen</b>	El Sistema debe permitirle al usuario obtener en un reproductor VLC el contenido de video captado por la cámara.	
<b>Referencias</b>	RF19	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítica.	
<b>Precondiciones</b>	Capa de Cámara seleccionada y Cámara seleccionada.	
<b>Postcondiciones</b>	Muestra reproductor VLC.	
<b>Flujo de eventos</b>		
<b>Flujo básico:</b> Selección de la cámara con todas la precondiciones.		
	<b>Actor</b>	<b>Sistema</b>
1.	Hace click en el menú "Ver". Luego en "Mostrar Video "	
2.		Muestra una nueva ventana que contiene un reproductor VLC con el contenido del video captado por la cámara seleccionada.
<b>Relaciones</b>	<b>CU Incluidos</b>	No procede.
	<b>CU Extendidos</b>	CU8, CU9, CU10, CU11, CU12, CU13

Tabla. 12: Descripción Textual del Caso de Uso Mostrar el área de influencia.



<b>CU12</b>	Mostrar el área de influencia
<b>Objetivo</b>	Iniciar la conexión con el Módulo Gestor del Sistema Silema Suria
<b>Actores</b>	Operador RG
<b>Resumen</b>	El Sistema debe permitirle al usuario ingresar los datos para establecer la conexión con el módulo Gestor del SXS
<b>Referencias</b>	RF1
<b>Complejidad</b>	Media.
<b>Prioridad</b>	Crítica.
<b>Precondiciones</b>	Capa de Cámaras seleccionada.
<b>Postcondiciones</b>	El sistema se conecta al Módulo Gestor del Sistema Silema Suria permitiendo usar sus funcionalidades.

### Flujo de eventos

#### Flujo básico:

	Actor	Sistema
1.	Hace click en el menú "Ver". Luego en "Área de Influencia de las Cámaras"	
2.		Muestra una ventana para seleccionar las capas bloqueantes. Ver Sección: Seleccionar Capas Bloqueantes.
7.		Muestra un cuadro de dialogo para escoger el nombre de la nueva capa.
8.	Introduce el dato referente al nombre de la capa de determinación de Área de Influencia y hace clic en "Aceptar"	
9.		El sistema muestra una nueva capa con el Área de Influencia de las cámaras.

**Flujo Básico.** Sección: Seleccionar Capas Bloqueantes.

	Actor	Sistema
3.	El usuario selecciona las capas que pueden bloquear el área de influencia.	
4.		Marca las capas seleccionadas por el usuario.
5.	El usuario da click en la opción "Aceptar".	
		Guarda los cambios y termina el CU. Ir al paso 7 del Flujo de Eventos.
<b>Flujos alternos. Cancelar</b>		
	Actor	Sistema
5.1	El usuario da click en la opción "Cancelar".	
6.1		No guarda cambios y termina el CU.
<b>Flujos Alternos. Capa de determinación existente</b>		
1.1	Hace click en el menú "Ver". Luego en "Área de Influencia de las Cámaras"	
2.1		Muestra un mensaje de error: "Ya existe una capa con el área de visualización de las cámaras".
3.1	El usuario da click en la opción "OK".	
<b>Relaciones</b>	<b>CU Incluidos</b>	No procede.
	<b>CU Extendidos</b>	No procede.

### 2.7. Patrones

Un patrón es un modelo a seguir para realizar una actividad en específico. Estos surgen con la acumulación de experiencias de las actividades desarrolladas por los seres humanos en aras de alcanzar ciertos objetivos referentes a las esferas de la vida social. Por tanto se puede decir que estos constituyen una abstracción de “problema – solución”, ocupándose de resolver problemas repetidos. De esta manera se proporciona un vocabulario y entendimiento común a todo aquel que haga uso de ellos. En los siguientes acápite se harán alusión a los patrones de diseño y arquitectura utilizados en el sistema.

#### 2.7.1. Patrón Arquitectónico.

La Arquitectura de Software se refiere a “La organización fundamental de un sistema encarnada en sus componentes, las relaciones de los componentes con cada uno de los otros y con el entorno, y con los principios que orientan su diseño y evolución” (Bass, 2013). Para el diseño del MIGREC-XS se propuso una Arquitectura en Capas. La arquitectura en capas definida según Garlan y Shaw es una organización jerárquica tal que cada capa proporciona servicios a la capa inmediata superior y se sirve de las prestaciones que le brinda la inmediata inferior (Shaw, 2006). Las ventajas del estilo en capas son muchas, primero que nada, el estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los programadores la participación de un problema complejo en una secuencia de pasos incrementales. Por otra parte admite optimizaciones y refinamientos, propiciando además una amplia reutilización, lo que se convierte en uno de los fuertes de esta arquitectura (Shaw, 2006). Los factores mencionados sumados a la posibilidad de tener bien acopladas las clases de MIGREC-XS y conocer el flujo de datos generado por las mismas hace que la arquitectura seleccionada se diferencie de otras como la de Tuberías y Filtros, Basadas en Componentes y Orientadas a Objetos que también fueron evaluadas pero que se desecharon por sus dificultades para lograr la comunicación correcta con el SXS.

En la Figura 5 se muestra las capas que se definieron para distribuir a MIGREC-XS y se muestra a través de que capa se realizará la comunicación con el SXS y con el Servidor Streaming de cada una de las cámaras. De esta forma se evidencia la integración de QGis con SXS.

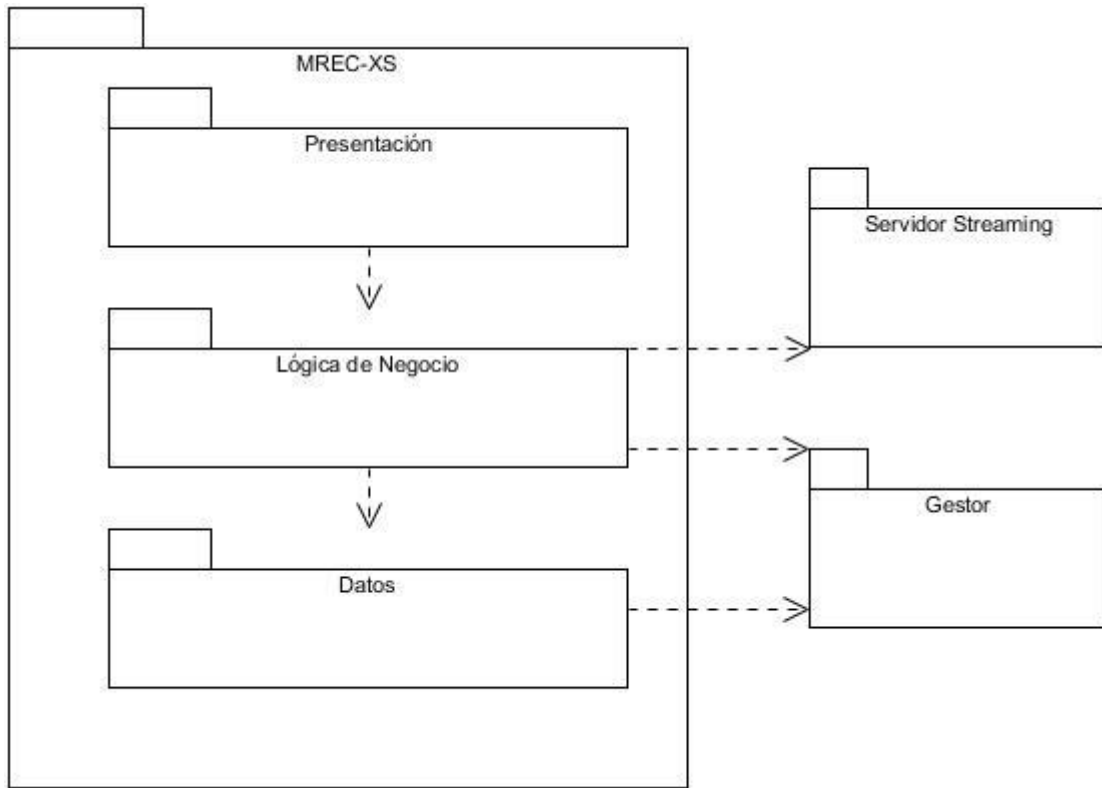


Fig. 5: Arquitectura del Sistema

### 2.7.2. Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ambientes referentes al diseño de interacción o interfaces, siendo aplicables a diferentes problemas de diseño en múltiples circunstancias. En este caso fue necesario la utilización de diversos patrones tanto GRASP como GoF, evidenciados de la siguiente forma:

#### GRASP:

Patrones GRASP, acrónimo de *General Responsibility Assignment Software Patterns* en español Patrones Generales de Software para Asignación de Responsabilidades, según (Gamma, y otros, 2004 ) estos describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.

En el acápite anterior donde se presenta la Figura 5 referente a la Arquitectura en Capas se evidenciaba que la capa de Lógica de Negocio es la encargada de recibir las peticiones del usuario y enviar las respuestas tras el proceso. En esta capa se tiene la clase controladora de procesos llamada `qgisapp.cpp`,

la cual controla el flujo de información del sistema, facilitando la centralización de las funcionalidades, y por tanto respondiendo al patrón de diseño: **Controlador**.

Como otras de las ventajas propiciadas por la arquitectura se evidencia que el mantenimiento del sistema es fácil debido al **Bajo Acoplamiento** y la **Alta Cohesión** entre las capas y la posibilidad de cambiar su implementación sin cambiar las interfaces. Por ejemplo, para el caso de MIGREC-XS permite cambiar la forma de conexión con el servidor streaming de cada cámara por cualquier otro servidor sin necesidad de reestructurar el sistema y cambiar las clases ya existentes. Además proporciona amplia reutilización por el bajo acoplamiento entre clases.

También se evidencia dentro de esta capa el patrón **Creador**, el cual plantea la necesidad de asignarle a una clase la responsabilidad de crear una instancia (Larman, 2003), ejemplo de esto refiere a la clase Treemodel la cual crea y contiene objetos del tipo Treeltem.

Para la asignación de responsabilidades se tuvo en cuenta el patrón **Experto**, donde a cada clase se le asignaron solo las responsabilidades que le correspondían según con la información que contaban en cada caso con la información necesaria para cumplir la responsabilidad.

### **Gof:**

Los patrones GoF (*Gang of Four*, en español Pandilla de los Cuatro), se clasifican en 3 categorías basadas en su propósito: creacionales, estructurales y de comportamiento. Los estructurales tratan la combinación de clases, su relación y la formación de estructuras de alta complejidad, los creacionales tratan la creación de instancias y los de comportamientos tratan la interacción y cooperación entre clases. (Gamma, y otros, 2004 )

Se utilizó el patrón **Fachada** el cual proporciona una interfaz unificada para un conjunto de interfaces o subsistemas, por lo tanto se puede afirmar que este patrón define una interfaz de alto nivel que hace más fácil el uso de los subsistemas.

El patrón **Observador** que fue otro de los utilizados, define una dependencia del tipo *uno a muchos* entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes. Este patrón es una de las características centrales en Qt utilizando *Signals* y *Slot* para la comunicación entre objetos, por lo que se evidencia su uso en la aplicación (Gamma, y otros, 2004 )

## 2.8. Diagramas de Clase del Diseño

Los diagramas de clases muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos se utilizan para modelar la vista de diseño estática de un sistema mostrando de esta forma las dependencias e interacciones.

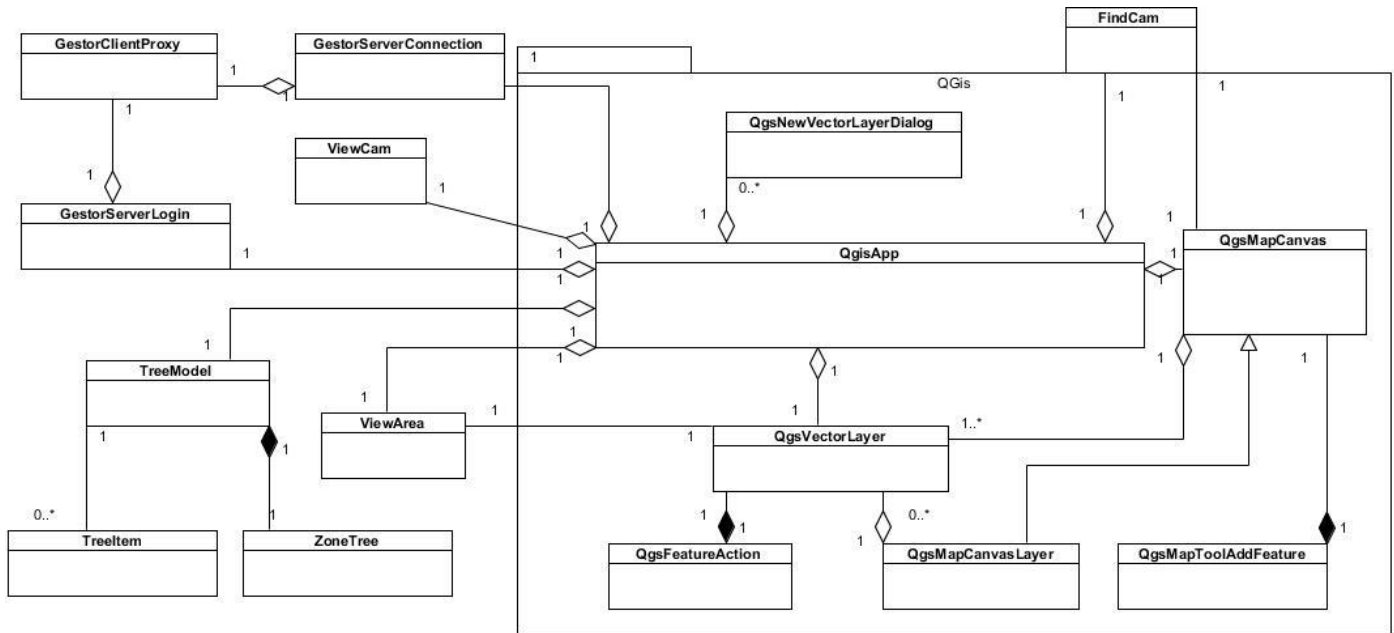


Fig. 6: Diagrama de Clases del Diseño

## 2.9. Utilización del Modelo de Diseño

Con la finalidad de situar el dominio del problema dentro de la perspectiva de los desarrolladores, RUP propone realizar el análisis y diseño del sistema. (Pressman, 2002) Afirma que el análisis tiene la misión de estudiar los requisitos, refinarlos y estructurarlos, con el objetivo de conseguir una comprensión más precisa de los mismos. Por su parte, el diseño es una representación abstracta de lo que se va a construir, contribuye a formar una arquitectura sólida y ayuda a crear un plano para la implementación. Mediante él se modela el sistema para que soporte todos los requisitos incluyendo los no funcionales. El propósito y objetivo del análisis debe alcanzarse de algún modo en todo proyecto. Pero la manera exacta de ver y de emplear el análisis puede diferir de un proyecto a otro y una de las variantes que se pueden emplear es no utilizar en absoluto el modelo de análisis para describir los resultados del análisis.

En cambio el proyecto analiza los requisitos como parte integrada de la captura de requisitos o en el diseño (Jacobson, 2000). (IBM, 2007) Hace referencia a los seis principios que define RUP. Uno de ellos plantea que la metodología se debe ajustar al tamaño del proceso y a las necesidades del proyecto. Considerando este principio que afirma que la metodología es altamente configurable, y que por tanto hay artefactos que

pueden obviarse durante el desarrollo, se decide hacer una transición directa del flujo de trabajo de requisitos al diseño, sin necesidad de realizar el análisis. Se llega a esta conclusión pues los requisitos son bien conocidos y se cuenta con toda la comprensión de ellos.

### **2.10. Conclusiones Parciales**

En el presente capítulo se analizó el modelo de dominio del problema para tomarlo como punto de partida para la investigación. Posteriormente se determinaron los requisitos funcionales y no funcionales dando paso a la reagrupación de estos en los CU. Además se realizó la descripción textual de los CU, con la intención de lograr un mejor entendimiento de las funcionalidades de la herramienta a desarrollar. Mediante el estudio de los patrones se logró detallar la arquitectura en n capas, en este caso 3 capas. Por otra parte se definieron los principales lo patrones GOF y GRASP, logrando con esto aplicar conceptos de reutilización y extensibilidad en las clases definidas en el proyecto. También dentro de los acápite de este capítulo se realizaron los diagramas de clases de diseño, secuencia y de colaboración para cada uno de los casos de usos del sistema.

## Capítulo 3. IMPLEMENTACIÓN Y PRUEBAS

### 3.1. Introducción

En este capítulo se presentan todos los elementos relacionados al flujo de trabajo de implementación, describiéndose detalladamente las técnicas utilizadas en el desarrollo de la aplicación. Se realizarán los diagramas de despliegue y de componentes de implementación resultantes de esta etapa de construcción. Se describen las pruebas de caja negra e integración realizadas a la aplicación, con el objetivo de validar su correcto funcionamiento y se exponen los resultados arrojados por estas.

### 3.2. Modelo de Implementación

En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, ficheros de código fuente .cpp y .h debido a su implementación en el lenguaje C++.

#### 3.2.1. Diagramas de Despliegue

El modelo de despliegue, es un importante artífice describe la arquitectura física del sistema durante su ejecución, en términos de procesadores, dispositivos y componentes de software. Describe, además, la topología del sistema, es decir, la estructura de los elementos de hardware y software que ejecuta cada uno de ellos.

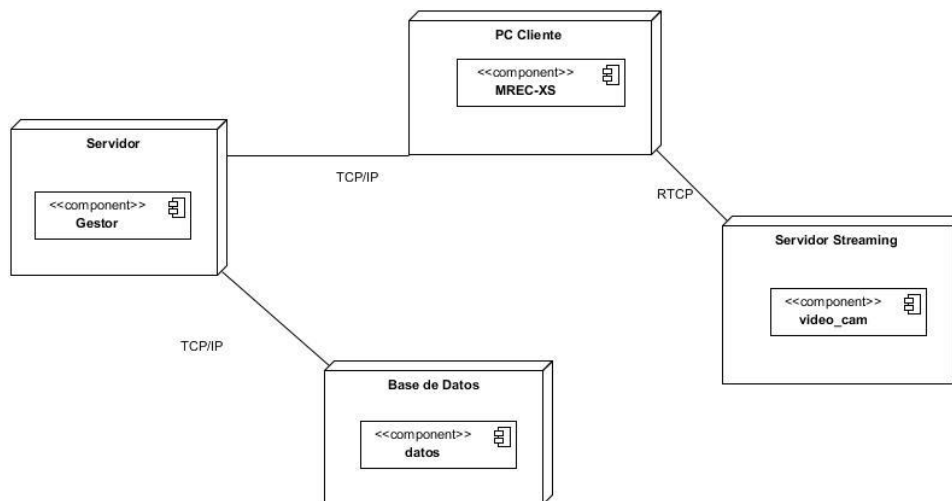
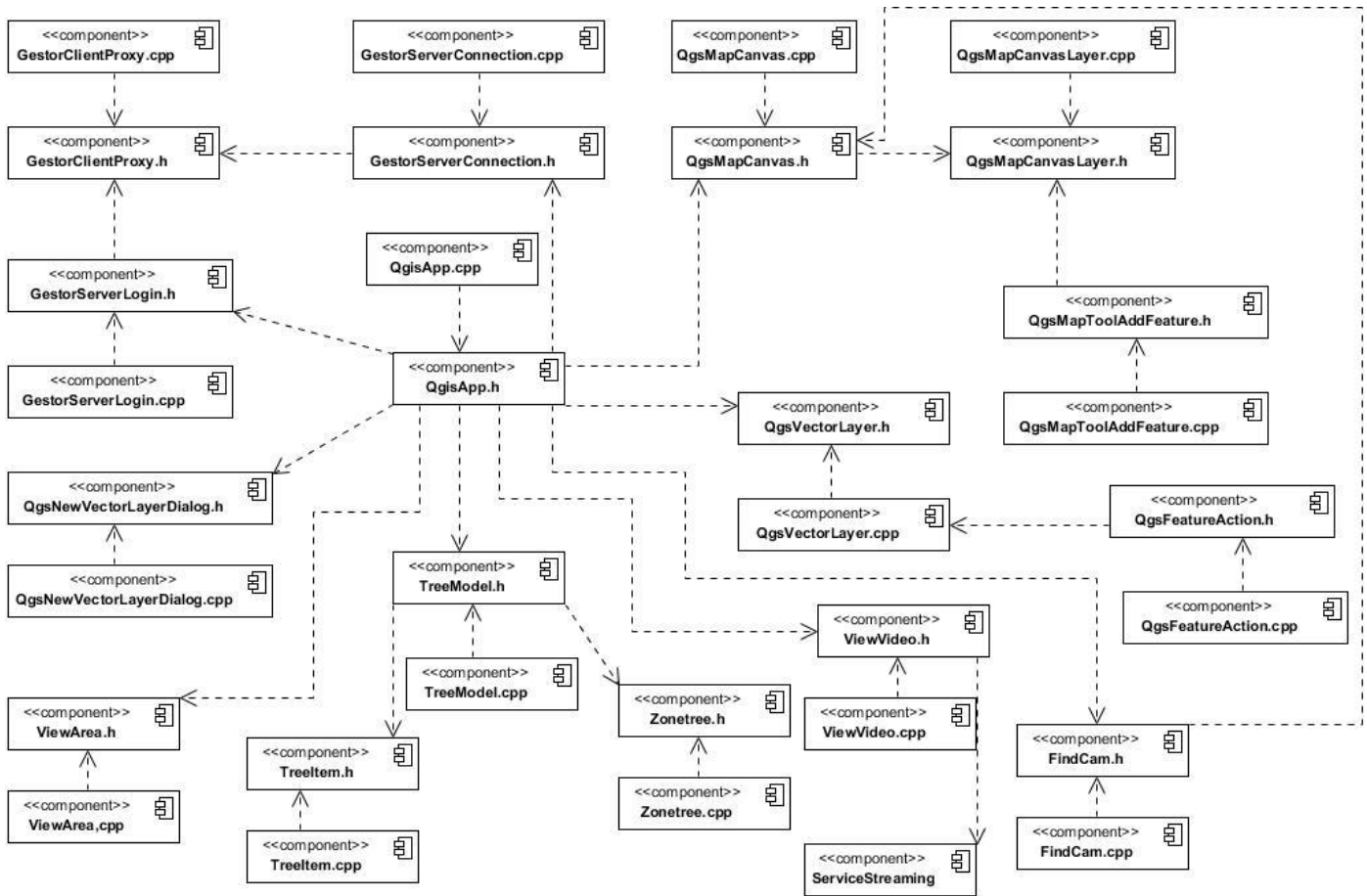


Fig. 7: Diagrama de Despliegue de MIGREC-XS



## 3.2.2. Diagramas de Componentes del Código Fuente



## 3.3. Pruebas

El objetivo principal de las pruebas como su nombre lo indica es revelar errores o incorrectos funcionamientos dentro una actividad en específico. Esto es muy aplicable dentro del tema de la Ingeniería de Software (IS), ya que con ello podemos garantizar entregar un producto de más calidad y eficacia. Dentro de las pruebas aplicadas al MIGREC-XS se encuentran la de caja negra, esta se llevó a cabo sobre la toda la interfaz del software y la de integración, para comprobar la funcionalidad del nuevo módulo con el Módulo Gestor de Xilema Suria.

### 3.3.1. Diseño de Prueba de Caja Negra sobre MIGREC-XS

Las pruebas de caja negra centran principalmente su atención en las características del software independientemente de lo materializado en el código. Para llevar a cabo estas se hace necesario tener conocimiento sobre los escenarios que en este se desarrollan, que no son más que las diferentes formas en que debe funcionar la aplicación en dependencia de la entrada o no de datos, los cuales pueden ser datos válidos y no válidos.

## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS

A continuación se hace un resumen de las pruebas aplicadas en aras de demostrar su correcto funcionamiento, en esta sección solo se muestra las correspondientes CU Gestionar Cámaras, para consultar las restantes ver **Anexo 1**.

Tabla. 13: Caso de Prueba de CU. Gestionar cámara (Insertar)

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Insertar cámara.	Inserta la representación de la cámara en el mapa.	Muestra la representación de la nueva cámara en el mapa	1. Click sobre una de las cámaras del Panel de Cámaras. 2. Arrastrar la cámara seleccionada hacia la zona de visualización. 3. Click para indicar a donde está mirando la cámara.
EC 1.2 Clic fuera del área de visualización.	No representan la cámara en el mapa y cancela la operación.	Muestra un mensaje de error: "No es un área valida". Cancela la inserción	1. Click sobre una de las cámaras del Panel de Cámaras. 2. Arrastrar la cámara seleccionada hacia la zona de visualización. 3. Click para indicar a donde está mirando la cámara.
EC 1.3 No hay capa de cámara seleccionada.	Verifica que existe una cámara seleccionada.	Muestra un mensaje de error: "No hay ninguna capa de cámara seleccionada"	1. Click sobre una de las cámaras del Panel de Cámaras. 2. Arrastrar la cámara seleccionada hacia la zona de visualización.
EC 1.4 La edición de capa no está activada.	Verifica que la edición de capa esté activa	Muestra un mensaje de error: "No está activada la edición de capas".	1. Click sobre una de las cámaras del Panel de Cámaras. 2. Arrastrar la cámara seleccionada hacia la zona de visualización.

Tabla. 14: Caso de Prueba de CU. Gestionar cámara. (Modificar)

Escenario	Descripción	Respuesta del sistema	Flujo central

## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS

EC 2.1 Modifica la posición de la cámara.	Modifica la posición de la cámara y la muestra en la nueva posición.	Muestra el mapa con la posición de la cámara modificada.	1. Selecciona el menú "Capa" y luego en "Conmutar edición" 2. Selecciona el menú "Edición". Y luego en "Mover objetos espaciales" 3. Click sobre la cámara seleccionada y la arrastrar hacia la nueva posición.
EC 2.2 Cámara existente.	Modifica la posición de la cámara y la muestra en la nueva posición.	Modifica la posición de la cámara existente.	1. Selecciona el menú "Capa" y luego en "Conmutar edición" 2. Selecciona el menú "Edición". Y luego en "Mover objetos espaciales" 3. Click sobre la cámara seleccionada y la arrastrar hacia la nueva posición. 4. Hace click en una de las cámaras del Panel de cámaras arrastrar hacia el mapa

Tabla. 15: Caso de Prueba de CU. Gestionar cámara. (Eliminar)

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 3.1 Modifica la representación de la cámara en el mapa.	Elimina la representación de la cámara del mapa.	Muestra el mapa con la representación de la cámara eliminada.	1. Selecciona el menú "Capa" y luego en "Conmutar edición" 2. Hace click en el menú "Edición". Luego en "Borrar lo seleccionado"
EC 3.2 No Hay elementos seleccionados.	No muestra ningún cambio en el mapa	No activa la función "Borrar lo seleccionado".	1. Selecciona el menú "Capa" y luego en "Conmutar edición" 2.

# CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS

Tabla. 17: Caso de Prueba de CU. Gestionar cámara. (Visualizar)

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 4.1 Modifica la representación de la cámara del mapa	Elimina la representación de la cámara del mapa.	Muestra la representación de las cámaras en el mapa.	1. Click en el "Panel de Capas" sobre una Capa de Cámara, para visualizar.
EC 4.2 No hay elementos insertados.	No muestra ningún cambio en el mapa con respecto a las cámaras	Muestra el mapa sin la ocurrencia de ningún cambio.	1. Click en el "Panel de Capas" sobre una Capa de Cámara, para visualizar.

### 3.3.2. Resultados Obtenidos

Esta sección se enfoca en los resultados arrojados en la fase de pruebas realizadas al MIGREC-XS.



Fig. 8: Gráfico de pruebas realizadas al MIGREC-XS

El gráfico refleja el resultado obtenido en la primera y segunda fase de prueba. Los errores reflejados en él, fueron detectados en los escenarios de los casos de pruebas de los CU Gestionar cámaras, Manipular atributos de la cámara, Mostrar flujo de video captado por la cámara. A continuación se definen los problemas detectados, y el número de iteración en que fueron hallados.

- ✚ Se podía insertar una cámara en cualquier capa de tipo punto. 1
- ✚ Luego de cerrar el proyecto y volverlo a cargar se podía repetir la representación de una cámara en la misma capa.
- ✚ Los atributos de la cámara no podían modificarse, luego de cerrar y volver a cargar el proyecto. 1

- ✚ Luego de realizada una visualización de un flujo de video quedaba activada la opción de ver video aunque no hubiese capa de cámara seleccionada. 2
- ✚ Las cámaras no guardaban las coordenadas de la dirección de la visión correctamente. 2

### **3.3.3. Pruebas de Integración**

Estas pruebas fueron ejecutadas para asegurar que los componentes en el modelo de implementación operaran correctamente al ser combinados para ejecutar un CU, ejemplo de esto se evidencia en la integración del Módulo Gestor de Xilema Suria con MIGREC-XS. Para ello fue necesario ir comprobando paquete a paquete y funcionalidad a funcionalidad en aras de descubrir errores o incompletitud. Durante el proceso se utilizó una integración incremental donde se fueron construyendo las pruebas y probando en pequeños segmentos de manera que los errores fueran más fáciles de aislar y corregir, consiguiendo con esto poder probar completamente las interfaces. Como alternativa para lograr un mejor desempeño de las pruebas se usó una estrategia de integración incremental ascendente para comprobar los niveles subordinados, es decir y probando desde lo más pequeño hasta lo general.

Al aplicar estas pruebas se detectaron diversos problemas en las tecnologías y librerías utilizadas por los diferentes módulos. Ejemplos de esto se evidenció en la en las versiones de la librería QCA y zeromq, las cuales fueron evaluadas y acopladas a las características de los dos sistemas.

### **3.4. Conclusiones Parciales.**

En este acápite de la investigación se logró desarrollar todas las funcionalidades presentadas en la descripción de requisitos funcionales. Esto propició obtener una herramienta denominada MIGREC-XS, la que se integró satisfactoriamente con el Sistema de Video Vigilancia Xilema Suria, representando un aporte notable a las funcionalidades del sistema existente. Se elaboraron y aplicaron diversas pruebas de caja negra e integración a la aplicación, donde se comprobó que sus funcionalidades son operativas y utilizables. Al analizar los resultados obtenidos por las pruebas y las encuestas desarrolladas a los especialistas, se puede decir que el MIGREC-XS cumple con la calidad requerida, además cuenta con una interfaz intuitiva y amigable para cualquier usuario que la utilice.

## CONCLUSIONES GENERALES

En este trabajo se presentó el análisis, diseño e implementación de una herramienta que permite la georreferenciación de las cámaras del SXS, la que se denomina MIGREC-XS. Este proyecto le añade un alto valor funcional al actual sistema de Video Vigilancia desarrollado en el centro, posibilitando al usuario (Operador del Representador Geográfico), tener una perspectiva mucho más amplia y detallada sobre el ámbito donde están ubicados los dispositivos de seguridad.

La solución alcanzada combina las fortalezas de los SIG con las ventajas que brinda la tecnología en aras de una mejor percepción del ambiente en que se establezca. Para la realización del módulo se hizo necesario la utilización de servicios y herramientas ya creadas, de las cuales se nutrió para mejorar las prestaciones. Ejemplo de esto se evidencia en el Módulo Gestor del SXS y los servidores Streaming correspondientes a cada una de las cámaras, lo que promueve el trabajo colaborativo y en equipo, artífice fundamental para la formación de profesionales del país.

Para el avance y perfeccionamiento de esta aplicación se incorporaron diversas funcionalidades que surgieron a partir de entrevistas realizadas al personal inmiscuido tanto en el desarrollo como en la utilización del tema. También fue necesario el uso de fundamentos de la trigonometría y la geometría computacional para mejorar las prestaciones que brindaba el SIG utilizado en correspondencia con la representación de territorios en forma circular. La presente investigación constituirá una variante muy efectiva para mejorar y actualizar el sistema de protección de seguridad de la UCI.

Las herramientas seleccionadas para el desarrollo de la aplicación facilitaron el trabajo en gran medida, pero a su vez exigió una búsqueda exhaustiva de donde colocar cada uno de los métodos a incorporar debido a la complejidad de la arquitectura del SIG utilizado.

El uso de tecnologías libres propicia la alternativa necesaria para el país, debido a la reducción de gastos y la sustitución de importaciones.

Se comprobó el correcto funcionamiento del módulo a partir de las pruebas de caja negra, realizadas mediante los casos de prueba, propiciando con ello la verificación de los requisitos definidos.

Se implementó un procedimiento para personalizar una capa del SIG de tal manera que agrupe todas los metadatos de una cámara. El mismo ha sido útil y extensible a otras problemáticas dentro del proyecto que requieren de lo mismo aunque con otras primitivas.

## CONCLUSIONES GENERALES

---

Se logró una representación espacial en forma de abanico basada en fundamentos de trigonometría, esta aporta un nuevo modelo de representación en el SIG; que hasta ahora entre sus primitivas solo se encontraban punto, línea y polígono. La forma de representación implementada propone una tematización real del área de visualización de una cámara.

## RECOMENDACIONES

Se recomienda:

- ✚ Implementar un algoritmo para determinar automáticamente el mejor posicionamiento de n cámaras en un área determinada por el usuario.
- ✚ Implementar un algoritmo para la representación del área de visualización de las cámaras teniendo en cuenta los obstáculos.
- ✚ Hacer uso de la solución propuesta en el Sistema de Video Vigilancia instaurado en la UCI.



## REFERENCIAS BIBLIOGRÁFICAS

**FAO. 2012.** Los sistemas de información geográfica y la telepercepción en la pesca y la agricultura. [En línea] 2012. [Citado el: 12 de 3 de 2015.] <http://www.fao.org/docrep/003/T0446S/T0446S08.htm>.

**Accesor Applications ans Services. 2014.** Accesor. [En línea] 2014. [Citado el: 26 de 3 de 2016.] [http://www.accesor.com/esp/art2\\_query.php?fam=5](http://www.accesor.com/esp/art2_query.php?fam=5).

**Álvarez, Claudio. 2014.** Educación Visual y Plástica. [En línea] 2014. [Citado el: 1 de junio de 2016.] <https://profecaevp.wordpress.com/2015/04/05/sistemas-de-representacion-espacial/>.

**ARC Gis Resorces. 2014.** Esri. [En línea] 2014. [Citado el: 12 de mayo de 2016.] <http://resources.arcgis.com/es/help/getting-started/articles/026n0000000s000000.htm>.

**Axis Communications. 2014.** Axis. [En línea] 2014. [Citado el: 22 de 6 de 2016.] <http://www.axis.com/global/es/glossary/network-video>.

**Bass, Len, Clements, Paul y Kazman, Rick. 2013.** *Software Architecture in Practice 3rd Edition*. New York : Addison Wesley, 2013.

**Bravo, J. 200.** *Breve Introducción a la Cartografía y a los Sistemas*. 200.

**Carrero, Angel. 2010.** QtCreator, Un completo entorno de desarrollo. [En línea] 10 de junio de 2010. [Citado el: 27 de enero de 2016.] [http://www.programacion.com/noticia/qt\\_creator-un\\_completo\\_entorno\\_de\\_desarrollo\\_1723](http://www.programacion.com/noticia/qt_creator-un_completo_entorno_de_desarrollo_1723).

**Chagoya, Ena Ramos. 2008.** *Licenciatura en ciencias de la educación. Maestría en docencia media superior y superior*. . Minatitlan, Mexico. : s.n., 2008.

**DATYS. 2005.** DATYS. [En línea] 2005. [Citado el: 16 de 12 de 2015.] <http://www.datys.cu/spa/site/index>.

**FAO. 2015.** Los sistemas de información geográfica y la telepercepción en la pesca y la agricultura. [En línea] 2015. [Citado el: 4 de 10 de 2015.] <http://www.fao.org/docrep/003/T0446S/T0446S08.htm>.

**Gabriel Ortiz. 2003.** GabrielOrtiz. [En línea] 2003. [Citado el: 4 de 10 de 2015.] <http://gabrielortiz.com/>.

**Gamma, Erich, y otros. 2004** . *Design Patterns* . Zúrich : s.n., 2004 .

**Geo Vision. 2009.** Geo Vision. [En línea] 2009. [Citado el: 3 de 10 de 2015.] [http://www.geovision.com.tw/SP/Prod\\_GVGIS.asp](http://www.geovision.com.tw/SP/Prod_GVGIS.asp).

## REFERENCIAS BIBLIOGRÁFICAS

---

- Hernandez, Rodrigo Jorge. 2013.** Video-ip. [En línea] 17 de 07 de 2013. [Citado el: 1 de 10 de 2015.] <http://video-ip.net/2013/07/17/4-sistemas-de-gestion-gratuitos-para-videovigilancia-ip/>.
- HotMath. 2013.** HotMath.com. [En línea] 2013. [Citado el: 29 de abril de 2016.] [http://hotmath.com/hotmath\\_help/spanish/topics/lines-segments-rays.html](http://hotmath.com/hotmath_help/spanish/topics/lines-segments-rays.html).
- Hudiel, Sergio Navarro. 2008.** *Manual de Topografía* . 2008.
- IBM. 2007.** *Ayuda de Rational Unified Process*. 2007.
- Jacobson. 2000.** *El proceso unificado de desarrollo software*. Madrid : Series Editors, 2000.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000. 84-7829-036-2..
- Larman, Craig. 2003.** *Modelo del Domino*. 2003.
- . **1999.** *UML y Patrones Introducción al análisis y diseño orientado a objetos*. México : PRENTICE HALL, 1999. 970-17-0261-1..
- MathsisFun. 2011.** Disfruta las Matemáticas. [En línea] 2011. [Citado el: 25 de mayo de 2016.] <http://www.disfrutalasmatematicas.com/geometria/poligonos.html>.
- Milord, Andrés Reynaldo. 2012.** *GeoVV: Una herramienta para la gestión de video* . La Habana : s.n., 2012.
- Netkrom. 2010.** Seguridad Publica y Video Vigilancia. [En línea] 2010. [Citado el: 3 de 10 de 2015.] <http://netkromsolutions.com/seguridad-publica-y-videovigilancia>.
- Novenca Security Systems. 2013.** Novenca. [En línea] 2013. [Citado el: 6 de 4 de 2016.] [http://www.novenca.com/site/index.php?option=com\\_content&view=article&id=98&Itemid=88](http://www.novenca.com/site/index.php?option=com_content&view=article&id=98&Itemid=88).
- Pressman. 2002.** *Ingeniería del Software. Un enfoque práctico. Quinta Edición*. 2002.
- Quintanilla, Yunier Leyva. 2013.** *Componente para la representación espacial de*. 2013.
- Rivera, Miguel Félix Mata. 2012.** *Recuperando y Analizando Datos*. Mexico : s.n., 2012.
- Salomón, Leticia. 2004.** *EL DESEMPEÑO POLICIAL Y LA SATISFACCIÓN DE LA CIUDADANÍA*. Tegucigalpa, Honduras : s.n., 2004.

## REFERENCIAS BIBLIOGRÁFICAS

---

**Shaw, David Garlan y Mary. 2006.** *An introduction to software architecture*. s.l. : CMU Software Engineering Institute Technical Report, 2006. CMU/SEI-94-TR-21 ESC-TR-94-21.

**Sosa, Angel Gabriel Oliveira. 2010.** *Ingeniería en Sistemas Computacionales*. República de Campeche : s.n., 2010.

**Tecnun. 2004.** *Aprenda C++ Básico como si estuviera en primero*. Navarra : s.n., 2004.

**Vázquez, María Inés. 2007.** *La metodología de Casos*. s.l. : Gráfica Don Bosco, 2007. ISBN 978-9974-7704-8-5.

### BIBLIOGRAFÍAS CONSULTAS:

**FAO. 2012.** Los sistemas de información geográfica y la telepercepción en la pesca y la agricultura. [En línea] 2012. [Citado el: 12 de 3 de 2015.] <http://www.fao.org/docrep/003/T0446S/T0446S08.htm>.

**Accesor Applications ans Services. 2014.** Accesor. [En línea] 2014. [Citado el: 26 de 3 de 2016.] [http://www.accesor.com/esp/art2\\_query.php?fam=5](http://www.accesor.com/esp/art2_query.php?fam=5).

**Álvarez, Claudio. 2014.** Educación Visual y Plástica. [En línea] 2014. [Citado el: 1 de junio de 2016.] <https://profecaevp.wordpress.com/2015/04/05/sistemas-de-representacion-espacial/>.

**ARC Gis Resorces. 2014.** Esri. [En línea] 2014. [Citado el: 12 de mayo de 2016.] <http://resources.arcgis.com/es/help/getting-started/articles/026n0000000s000000.htm>.

**Axis Communications. 2014.** Axis. [En línea] 2014. [Citado el: 22 de 6 de 2016.] <http://www.axis.com/global/es/glossary/network-video>.

**Bass, Len, Clements, Paul y Kazman, Rick. 2013.** *Software Architecture in Practice 3rd Edition*. New York : Addison Wesley, 2013.

**Bravo, J. 200.** *Breve Introducción a la Cartografía y a los Sistemas*. 200.

**Carrero, Angel. 2010.** QtCreator, Un completo entorno de desarrollo. [En línea] 10 de junio de 2010. [Citado el: 27 de enero de 2016.] [http://www.programacion.com/noticia/qt\\_creator-un\\_completo\\_entorno\\_de\\_desarrollo\\_1723](http://www.programacion.com/noticia/qt_creator-un_completo_entorno_de_desarrollo_1723).

**Chagoya, Ena Ramos. 2008.** *Licenciatura en ciencias de la educación. Maestría en docencia media superior y superior*. . Minatitlan, Mexico. : s.n., 2008.

**Conferencia 7. Ingeniería Software. 2006-2007.** *Fase de Inicio. Flujo de Análisis y Diseño. Diagramas de Interacción*. 2006-2007.

**DATYS. 2005.** DATYS. [En línea] 2005. [Citado el: 16 de 12 de 2015.] <http://www.datys.cu/spa/site/index>.

**FAO. 2015.** Los sistemas de información geográfica y la telepercepción en la pesca y la agricultura. [En línea] 2015. [Citado el: 4 de 10 de 2015.] <http://www.fao.org/docrep/003/T0446S/T0446S08.htm..>

**Gabriel Ortiz. 2003.** GabrielOrtiz. [En línea] 2003. [Citado el: 4 de 10 de 2015.] <http://gabrielortiz.com/>.

## BIBLIOGRAFÍAS CONSULTADAS

---

**Gamma, Erich, y otros. 2004 .** *Design Patterns* . Zúrich : s.n., 2004 .

**Geo Vision. 2009.** Geo Vision. [En línea] 2009. [Citado el: 3 de 10 de 2015.]  
[http://www.geovision.com.tw/SP/Prod\\_GVGIS.asp](http://www.geovision.com.tw/SP/Prod_GVGIS.asp).

**Gunnar ÖVERGAARD, Karin PALMKVIST. 2004.** *Use Cases: Patterns and Blueprints*” . s.l. : Addison, 2004.

**Hernandez, Rodrigo Jorge. 2013.** Video-ip. [En línea] 17 de 07 de 2013. [Citado el: 1 de 10 de 2015.]  
<http://video-ip.net/2013/07/17/4-sistemas-de-gestion-gratuitos-para-videovigilancia-ip/>.

**HotMath. 2013.** HotMath.com. [En línea] 2013. [Citado el: 29 de abril de 2016.]  
[http://hotmath.com/hotmath\\_help/spanish/topics/lines-segments-rays.html](http://hotmath.com/hotmath_help/spanish/topics/lines-segments-rays.html).

**Hudiel, Sergio Navarro. 2008.** *Manual de Topografía* . 2008.

**IBM. 2007.** *Ayuda de Rational Unified Process*. 2007.

**Ivar Jacobson, Grady Booch, James Rumbaugh. 2000.** *El Proceso Unificado de Desarrollo de*. s.l. : Addison Wesley, 2000.

**Jacobson. 2000.** *El proceso unificado de desarrollo software*. Madrid : Series Editors, 2000.

**Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000. 84-7829-036-2..

**Larman, Craig. 2003.** *Modelo del Domino*. 2003.

—. **1999.** *UML y Patrones Introducción al análisis y diseño orientado a objetos*. México : PRENTICE HALL, 1999. 970-17-0261-1..

**Len Bass, Paul Clements y Rick Kazman. 2013.** *Software Architecture in Practice 3rd Edition*. New York : Addison Wesley, 2013.

**MathsIsFun. 2011.** Disfruta las Matemáticas. [En línea] 2011. [Citado el: 25 de mayo de 2016.]  
<http://www.disfrutalasmaticas.com/geometria/poligonos.html>.

**Milord, Andrés Reynaldo. 2012.** *GeoVV: Una herramienta para la gestión de video* . La Habana : s.n., 2012.

## BIBLIOGRAFÍAS CONSULTADAS

---

**Netkrom. 2010.** Seguridad Publica y Video Vigilancia. [En línea] 2010. [Citado el: 3 de 10 de 2015.] <http://netkromsolutions.com/seguridad-publica-y-videovigilancia>.

**Novenca Security Systems. 2013.** Novenca. [En línea] 2013. [Citado el: 6 de 4 de 2016.] [http://www.novenca.com/site/index.php?option=com\\_content&view=article&id=98&Itemid=88](http://www.novenca.com/site/index.php?option=com_content&view=article&id=98&Itemid=88).

**Pressman. 2002.** *Ingeniería del Software. Un enfoque práctico. Quinta Edición.* 2002.

**Quintanilla, Yunier Leyva. 2013.** *Componente para la representación espacial de.* 2013.

**Rivera, Miguel Félix Mata. 2012.** *Recuperando y Analizando Datos.* Mexico : s.n., 2012.

**Salomón, Leticia. 2004.** *EL DESEMPEÑO POLICIAL Y LA SATISFACCIÓN DE LA CIUDADANÍA.* Tegucigalpa, Honduras : s.n., 2004.

**Shaw, David Garlan y Mary. 2006.** *An introduction to software architecture.* s.l. : CMU Software Engineering Institute Technical Report, 2006. CMU/SEI-94-TR-21 ESC-TR-94-21.

**Sosa, Angel Gabriel Oliveira. 2010.** *Ingeniería en Sistemas Computacionales.* República de Campeche : s.n., 2010.

*Técnicas para facilitar las especificaciones de la aplicación.*

**Tecnun. 2004.** *Aprenda C++ Básico como si estuviera en primero.* Navarra : s.n., 2004.

**Vázquez, María Inés. 2007.** *La metodología de Casos.* s.l. : Gráfica Don Bosco, 2007. ISBN 978-9974-7704-8-5

**Abdolazimian, Farid y Mansori, Saeedeh. 2008.** *Business Process Reengineering by Rational Unified Process (RUP) Methodology.* s.l. : World Applied Sciences Journal, 2008.

**Álvarez de Zayas, Carlos. 1995.** *Metodología de la investigación científica.* Santiago de Cuba : Centro de Estudios de Educación Superior" Manuel F. Gran". Universidad de Oriente, 1995.

**Bass, Len, Clements, Paul y Kazman, Rick. 2013.** *Software Architecture in Practice 3rd Edition.* New York : Addison Wesley, 2013.

**Bertino, E A y Martino, L A. 1995.** *Sistemas de bases de datos orientadas a objetos.* 1995.

**Daniel, Rivero. 2013.** *Metodología de la investigación.* 2013.

**Day, Lewis F. 2013.** *Pattern design.* s.l. : Courier Corporation, 2013.

## BIBLIOGRAFÍAS CONSULTADAS

---

**Fernández, Eduardo. 2013.** *Security patterns in practice: designing secure architectures using software patterns.* . s.l. : John Wiley & Sons, 2013.

**Gamma, Erich, y otros. 2004.** *Design Patterns.* Zúrich : s.n., 2004.

**Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Perason Educacion, 2000.

**Jiménez, Ileana Vargas. 2012.** *LA ENTREVISTA EN LA INVESTIGACIÓN CUALITATIVA: NUEVAS TENDENCIAS Y RETOS.* 2012. pág. 21.

**Josutis, Nicolai M. 2012.** *The C++ standard library: a tutorial and reference.* s.l. : Addison-Wesley, 2012.

**Juristo, Natalia, Moreno, Ana M. y Vegas, Sira. 2006.** *Técnicas de Evaluación de Software.* Madrid : s.n., 2006.

**León, Rolando Alfredo Hernández y González, Sayda Coello. 2011.** *El proceso de investigación científica.* . s.l. : Editorial Universitaria, 2011.

**Lyon, David y Ball, Kirstie. 2012.** *Routledge handbook of surveillance studies.* s.l. : Routledge, 2012.

**Malmenbrat, Johan y Brooks, David J. 2015.** *CCTV surveillance: The differing aims and functions of CCTV.* 2015.

**Meurant, Gerard. 2013.** *CCTV Surveillance: Video Practices and Technology.* s.l. : Elsevier, 2013.

**OXFORD University Press. 2016.** *oxforddictionaries. oxforddictionaries.* [En línea] 21 de marzo de 2016. <http://www.oxforddictionaries.com/es>