Universidad de las Ciencias Informáticas

Facultad 6

Aplicación para descubrir Cámaras IP en una Red de Área Local

Trabajo de Diploma para optar por el Título de

Ingeniero en Ciencias Informáticas

Autor: Elizabeth Mora Saavedra

Tutor: Ing. Reynier Pupo Gómez

Ciudad de La Habana, 14 de Junio de 2011

"Año 53 de la Revolución"

"Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad."

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autor:

_____

Elizabeth Mora Saavedra

Tutor:

_____

Ing. Reynier Pupo Gómez

DATOS DE CONTACTO

Tutor: Ing. Reynier Pupo Gómez:

Correo electrónico: rgomez@uci.cu

Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2010.

Instructor Recién Graduado

A mi mamá, por siempre hacer hasta lo imposible para ayudarme a alcanzar mis sueños.

AGRADECIMIENTOS

Este trabajo de diploma marca el final de una larga etapa de estudios. Quisiera agradecer a todos aquellos que han tenido que ver en mi formación como profesional y como persona todos estos años.

A mis profesores, por ayudarme a comprender un mundo que en principio me era totalmente desconocido.

A la UCI, porque estando en ella he crecido y aprendido a ver la vida de una forma diferente.

A la Revolución y a Fidel, por darme la oportunidad de convertirme en una profesional.

A mi tutor y a su hermano por aclarar mis no pocas dudas.

A mis amigos, por compartir mis tristezas y alegrías estos 5 años, en especial a Anett porque, aunque la dinámica de esta universidad nos ha mantenido alejadas, seguimos siendo tan amigas como cuando nos conocimos en décimo grado.

A mi tío Minino, por ser como un padre para mí.

A mi novio, por el maravilloso tiempo que me ha regalado a su lado, por escucharme y brindarme su apoyo y confianza en los momentos en que creí que no podía seguir adelante.

Sobre todas las cosas, le agradezco a mi mamá, por ser padre y madre desde que nací, por hacerlo todo por mí desinteresadamente, por todas las noches sin dormir, por enseñarme a ser una buena persona, por impulsarme siempre a superarme, por los consejos y la confianza que ha depositado en mí, por todo su amor y porque, a pesar de las dificultades, ha sabido ser la mejor mamá del mundo.

RESUMEN

La evolución tecnológica ha permitido el surgimiento de métodos novedosos para garantizar la seguridad física como es el caso de la vídeo vigilancia IP. Con el objetivo de satisfacer las necesidades de las empresas cubanas en este campo, en la Universidad de las Ciencias Informáticas se encuentra en proceso de desarrollo el Sistema de Vídeo Vigilancia Suria. En la actualidad el sistema no cuenta con la funcionalidad de descubrir automáticamente las cámaras IP conectadas en la red.

La presente investigación tiene como objetivo la creación de una aplicación basada en componentes, encargada del descubrimiento automático de cámaras IP, que se integre al Sistema de Vídeo Vigilancia Suria. Para la construcción de esta solución han sido utilizados RUP como metodología de desarrollo, UML como lenguaje de modelado, como lenguaje de Programación C++, así como el marco de trabajo Qt.

El resultado de esta investigación es una aplicación que permite el descubrimiento de cámaras IP en una Red de Área Local. Posibilita, además, configurar las cámaras y añadirlas al sistema Suria.

Palabras claves:

cámaras IP, red de área local, vídeo vigilancia

ABSTRACT

Technological development has allowed the emergence of innovative methods to ensure physical security. One of these methods is IP video surveillance. In order to meet the needs of Cuban enterprises, the University of Information Sciences is in the process of developing the SURIA Video Surveillance System. Currently the system lacks of the functionality to automatically discover IP cameras on the network. This research aims to implement a component-based application, responsible for the automatic discovery of IP cameras, which can be integrated into SURIA Video Surveillance System.

This solution has been developed using Rational Unified Process as the methodology to structure, plan, and control the process, C++ as the programming language and Qt as the framework used to build the system. The result of this research is an application that allows the discovery of IP cameras on a Local Area Network. It also allows to configure the cameras and to add these to SURIA System.

Keywords:

discovery, local area network, IP cameras, video surveillance.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

TABLA DE CONTENIDOS

INTRODUCCIÓN

La seguridad física constituye el primer y más visible aspecto de protección. Está compuesta por un conjunto de elementos que resultan vitales para, en caso de amenaza, lograr un tiempo adecuado de respuesta y evitar, en el momento oportuno, cualquier peligro. Desde tiempos remotos el hombre ha empleado diferentes medios para salvaguardar su vida y sus posesiones, desde puertas y cerraduras hasta guardias de seguridad. La evolución tecnológica ha permitido el surgimiento de métodos novedosos y más eficientes para lograr la protección de medios y vidas en hogares, oficinas, bancos, aeropuertos y otras instituciones. Entre esos métodos se encuentra la vídeo vigilancia, la cual no es más que el empleo de vídeo en el monitoreo de una o varias localizaciones desde un sitio remoto.

Históricamente, las aplicaciones de monitorización y de vigilancia han sido soportadas por la tecnología analógica de Circuito Cerrado de Televisión (CCTV), que ha mostrado ser débil debido a la falta de flexibilidad, escalabilidad y tolerancia a fallas, además de su elevado costo. Gracias al acelerado desarrollo de las redes y el vídeo digital, los nuevos sistemas de vídeo vigilancia IP se han posicionado como una alternativa muy fuerte en el mercado pues permiten adaptarse a medios de transmisión más eficientes y flexibles.

Estos modernos sistemas ofrecen toda una serie de ventajas y funcionalidades avanzadas que no puede proporcionar un sistema de vídeo vigilancia analógica. Entre las ventajas se incluyen la accesibilidad remota, la alta calidad de la imagen, la gestión de eventos, así como las posibilidades de una integración sencilla con otras funciones.

Numerosas empresas en el mundo se dedican a la producción de software de vídeo vigilancia, como es el caso de Milestone Systems, Vídeo Insight y NCH Software, las cuales se encuentran entre las más relevantes.

Cuba, a pesar de ser un país subdesarrollado y de pocos recursos, ha ido avanzando en los últimos años en la Informatización de la Sociedad, proceso que busca lograr más eficacia y eficiencia, que permitan una mayor generación de riquezas y hagan sustentable el aumento sistemático de la calidad de vida de los ciudadanos.

La Industria Cubana del Software está llamada a convertirse en una significativa fuente de ingresos nacional, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano

disponible. La Universidad de las Ciencias Informáticas (UCI) y el sistema de empresas cubanas vinculadas a este trabajo juegan un papel importante en el desarrollo de esta industria, y en la materialización de los proyectos asociados al programa cubano de informatización.

La UCI como universidad de nuevo tipo, posee un novedoso modelo de formación que combina el estudio con la producción y la investigación. Está estructurada por 10 facultades, 7 en su sede central y 3 en diferentes regiones que incluyen el occidente, centro y oriente del territorio nacional.

Actualmente en la Facultad 6 de la sede central de la UCI se encuentra en desarrollo el Sistema de Vídeo Vigilancia Suria, con el objetivo de satisfacer las necesidades de las empresas cubanas. En las condiciones actuales, para que el sistema sea capaz de controlar las cámaras IP conectadas en la red es necesario que uno de los administradores inserte manualmente cierta información como la marca, el modelo y su dirección IP (Protocolo de Internet), debido a esto el proceso es engorroso, pues cuando el número de cámaras conectadas es grande, resulta prácticamente imposible, lo que provocaría que en el momento del despliegue la tarea se realice con lentitud.

Todo lo expresado anteriormente permite identificar como **problema de investigación**: La presencia en la red de una gran cantidad de cámaras de distintos fabricantes provoca que el proceso de adición de estas al Sistema de Vídeo Vigilancia Suria resulte engorroso. Para dar solución al problema detectado se propone como **objetivo general**: Desarrollar una herramienta capaz de encontrar cámaras IP de distintos fabricantes y modelos en una red, teniendo como **objeto de Estudio** el descubrimiento de dispositivos en la red.

Se ha definido como **campo de acción** el descubrimiento de cámaras IP en una Red de Área Local (LAN). Inicialmente se parte de la **hipótesis:** Con un componente capaz de encontrar las cámaras IP en una LAN se acelerará el proceso de puesta en marcha del Sistema de Vídeo Vigilancia Suria en los lugares donde se despliegue.

Para alcanzar el objetivo planteado anteriormente se han establecido las tareas de investigación que se muestran a continuación:

1. Caracterizar el proceso de descubrimiento de dispositivos en una red.
2. Describir el estado del arte de las aplicaciones para el descubrimiento de cámaras IP en una red.
3. Caracterizar las tecnologías y herramientas a utilizar para el desarrollo de la aplicación.

4. Analizar y diseñar la aplicación para el descubrimiento de cámaras IP en una LAN.
5. Implementar el núcleo de la aplicación así como los componentes para los principales fabricantes.
6. Realizar pruebas a la aplicación.

"El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones." (1)Los métodos teóricos permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis. (1). Los siguientes métodos teóricos se han utilizado en la realización de esta investigación:

Inducción y deducción: Utilizado para conocer las características fundamentales de los sistemas que realizan descubrimiento automático de dispositivos en una red, así como los métodos empleados con este fin.

Análisis y síntesis: Permite concretar los elementos más importantes relacionados con el objeto de estudio. Se realizó el análisis de un gran volumen de documentación, que permitió sintetizar el contenido que da soporte a la propuesta del trabajo a desarrollar.

Modelación: Empleado para estudiar las cualidades, las relaciones y la interacción entre los elementos del análisis y el diseño de la solución propuesta.

La investigación está estructurada de la siguiente manera:

En el Capítulo 1 "Fundamentación teórica", se hace una breve revisión de algunas aplicaciones que se dedican al descubrimiento de cámaras IP y se exponen los conceptos fundamentales asociados al dominio del problema.

En el Capítulo 2 "Tendencias y tecnologías actuales", se describen las metodologías de desarrollo, lenguajes de programación y tecnologías actuales para el desarrollo de la solución presentada, justificando además la selección y uso de estas.

En el Capítulo 3 “Características del sistema”, se realiza el modelado del dominio con el objetivo de comprender el contexto de la aplicación, se describe cómo debe funcionar y se destacan sus características distintivas; se especifican sus Requisitos Funcionales y No Funcionales y se presenta el Modelo de Sistema.

En el Capítulo 4 “Análisis y Diseño del Sistema”, se muestran los resultados obtenidos en el desarrollo de los procesos de análisis y diseño del sistema, así como los diagramas que fueron necesarios para obtener una mayor claridad a la hora de elaborar la solución que se propone.

En el Capítulo 5 “Implementación y validación de la solución”, se plasman los artefactos resultantes de la implementación así como los resultados de las pruebas.

CAPÍTULO 1. **FUNDAMENTACIÓN TEÓRICA**

1.1. Introducción

El presente capítulo muestra una exposición de los conceptos asociados al dominio del problema, que posibilitarán una mejor comprensión del tema tratado. Además, se presentan los principales aspectos relacionados con el descubrimiento de dispositivos en una red y se hace un análisis de las principales aplicaciones que descubren cámaras IP.

1.2. Conceptos asociados al dominio del problema.

Red de Área Local (LAN): Las redes de área local, generalmente llamadas LAN, son redes localizadas dentro de un solo edificio o campus de hasta unos pocos kilómetros de tamaño. Son ampliamente utilizadas para conectar las computadoras personales, estaciones de trabajo y otros dispositivos con el objetivo de compartir recursos e intercambiar información. Las LAN se distinguen de otros tipos de redes por tres características: su tamaño, su tecnología de transmisión y su topología. (2)

Protocolo: Un protocolo es un conjunto de reglas predefinidas que rigen la forma en que dos o más procesos se comunican e interactúan para intercambiar datos. Los procesos pueden estar en la misma máquina o en máquinas diferentes. Por ejemplo, un programa de la capa de transporte[1] en una máquina utiliza un protocolo para comunicarse con su contraparte en otra máquina. Los protocolos están generalmente asociados a determinados servicios o tareas, tales como el empaquetamiento de datos o el enrutamiento de paquetes. Un protocolo especifica las reglas para la creación, realización, y terminación de una conexión de comunicaciones, y también especifica el formato que los paquetes de información deben tener cuando se viaja a través de esta conexión. (3)

IP (Protocolo de Internet): Es el protocolo de capa de red con más amplio apoyo para la Internet. Es uno de los protocolos en la suite TCP/IP. Este protocolo define y enruta datagramas a través de Internet y ofrece un servicio de transporte no orientado a conexión. El protocolo IP utiliza la conmutación de paquetes y hace el mejor esfuerzo para entregar sus paquetes. (3)

[1] La capa de transporte del Modelo OSI es la encargada de controlar el flujo de datos entre los nodos que establecen una comunicación.

Dirección IP: Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP. (4)

Cámara IP: Una cámara de red, también llamada *cámara IP*, puede describirse como una cámara y un ordenador combinados para formar una única unidad. Como un ordenador, la cámara de red dispone de su propia dirección IP y está directamente conectada a la red. Una cámara IP puede proporcionar servicios de servidor web, FTP (Protocolo de transferencia de archivos) y funciones de correo electrónico, SMS, etc.

1.3. Descubrimiento de dispositivos en una red.

1.3.1. Auto-descubrimiento

El auto-descubrimiento es la capacidad de descubrir automáticamente los componentes de una red (5). Por lo general, se realiza en dos niveles:

Nivel 3 o Capa de Red[2]: Este nivel incluye el descubrimiento de los dispositivos relacionados con la capa 3 del modelo OSI[3]. Aquí se incluyen todos los dispositivos que tienen al menos una dirección IP. (6)

Nivel 2 o Capa de Enlace de Datos[4]: Este nivel incluye el descubrimiento de los dispositivos relacionados con la capa 2 del modelo OSI (como es el caso de los conmutadores). (6)

1.3.2. Alcance del auto-descubrimiento

El alcance del auto-descubrimiento se relaciona con el tipo o la estructura de la red que la herramienta es capaz de cubrir. El proceso de auto-descubrimiento puede extenderse a través de redes LAN y WAN:

A través de redes LAN: El proceso de auto-descubrimiento posee la capacidad de descubrir los nodos del dominio de difusión local, es decir, hasta la puerta de enlace local.

---

[2] La capa de red del Modelo OSI es la encargada de encaminar los paquetes además de ocuparse de entregarlos.
[3]El modelo de referencia de Interconexión de Sistemas Abiertos (OSI, Open Systems Interconnection) es el modelo de red descriptivo creado por la Organización Internacional para la Estandarización (ISO) lanzado en 1984. Es un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.
[4]La capa de enlace de datos se encarga de desplazar los datos por el enlace físico de comunicación hasta el nodo receptor.

A través de redes WAN: El proceso de auto-descubrimiento posee la capacidad de descubrir los nodos más allá del dominio de difusión local, incluyendo los enrutadores locales y remotos. Las herramientas de descubrimiento ofrecen opciones de configuración para definir hasta dónde se quiere descubrir, ya que si no se especifica un límite, la herramienta puede intentar detectar automáticamente los nodos en toda la Internet. Por ejemplo, este límite se puede definir proveyendo explícitamente los rangos de la red de interés.

1.3.3. Técnicas de auto-descubrimiento

Las dos técnicas principales para llevar a cabo el descubrimiento automático son: Técnica de Descubrimiento Activo: El proceso de descubrimiento envía tráfico a los dispositivos de red para estimular una respuesta. (7)

Técnica de Descubrimiento Pasivo: El proceso de descubrimiento no introduce ningún tráfico en la red, se dedica estrictamente a escuchar. (8)

Técnicas Activas

Las técnicas activas generan ciertos paquetes conocidos como “sondas” que estimulan la respuesta de los nodos de la red. Los datos de las respuestas y el comportamiento resultante se utilizan para obtener información de descubrimiento. Estos paquetes se generan a través de protocolos de red, entre los que se incluyen: SNMP, ARP, ICMP, TCP, DNS, NetBIOS, etc.

De los protocolos mencionados anteriormente, SNMP[5] tiene la capacidad para proporcionar la mayor cantidad de información para el proceso de descubrimiento y, como tal, es el más utilizado por las herramientas que realizan el auto-descubrimiento de forma activa. Las “sondas” generan una vista instantánea de la información, por tanto, para mantener una visión actualizada de la red, estos mensajes deben ser generados de forma periódica. Las técnicas activas descubren rápidamente la información a expensas de los recursos de ancho de banda de la red. También ofrecen la posibilidad de descubrir redes

[5]SNMP (Simple Network Management Protocol): El Protocolo Simple de Administración de Red es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, así como planear su crecimiento.

remotas, pero pueden, sin embargo, estar limitadas por los dispositivos de filtrado tales como los cortafuegos. (9)

Técnicas pasivas

A diferencia de las técnicas activas, las técnicas pasivas emplean aparatos de escucha llamados "sniffers" para obtener la información de descubrimiento. Los sniffers logran su objetivo mediante el análisis de las Unidades de Datos de Protocolo (PDU), así como de la observación de los comportamientos de ciertos protocolos (por ejemplo, TCP). No generan ningún paquete de datos en la red con el fin del descubrimiento, de esta manera no son limitados por los dispositivos de filtrado como los cortafuegos. Sin embargo, pueden generar tráfico de datos para comunicar sus hallazgos a una entidad central cuando se han implementado de forma distribuida. (10)

Debido a que un sniffer solo puede "escuchar" el tráfico de datos en su segmento de la red, las técnicas de detección pasiva a menudo distribuyen una serie de sniffers de manera que se logre cubrir la red de destino. Las técnicas pasivas presentan una visión en tiempo real de la red, sin embargo, los nodos que no transmiten o reciben datos no serán descubiertos por ellas.

Estas técnicas pueden detectar todos los servicios activos, no sólo los de los puertos conocidos. Debido a la forma en que operan, los métodos pasivos en general, acumularán la información en un período de tiempo más largo que el empleado por las técnicas activas. (8)

Los protocolos comúnmente analizados son: ARP, IP, RIP[6], OSPF[7], EGP[8], ICMP, TCP y UDP.

1.3.4. Categorías en que se pueden clasificar las herramientas de auto descubrimiento.

Herramientas basadas en SNMP

Esta categoría incluye las herramientas que se apoyan fuertemente en SNMP para realizar el descubrimiento y recoger la información. Estas herramientas requieren que la generalidad de los dispositivos de red tenga habilitado el protocolo SNMP. Cuando no sucede así, la mayoría será capaz de

[6]RIP (Routing Information Protocol): El Protocolo de Encaminamiento de Información es un protocolo utilizado por los enrutadores, aunque también puede actuar en equipos, para intercambiar información acerca de las redes.
[7]OSPF (Open Shortest Path First): Es un protocolo de enrutamiento jerárquico.
[8] EGP (Exterior Gateway Protocol): El Protocolo de Pasarela Exterior es un protocolo estándar usado para intercambiar información de enrutamiento entre sistemas autónomos.

descubrir poca información de interés, por lo general, se limitará a descubrir la dirección IP y el estado del nodo.

Herramientas Activas Híbridas

Esta categoría incluye las herramientas que realizan el descubrimiento, haciendo uso de múltiples técnicas activas o mediante la combinación de alguna técnica pasiva con una activa. Una herramienta puede incluir SNMP como parte de sus protocolos de descubrimiento activos, pero no se basa principalmente en SNMP para su funcionamiento. De esta forma, la herramienta es capaz de proporcionar suficiente información interesante, incluso si SNMP no es compatible con los dispositivos de red. (11)

Herramientas Pasivas

Las técnicas pasivas dependen de sensores para monitorizar los paquetes que fluyen por la red e inspeccionar el contenido de estos (cabeceras o datos encapsulados). En contraste con un enfoque activo, el proceso de recopilación pasiva de información no tiene ningún impacto en el ancho de banda o en los activos controlados. Por lo tanto la vigilancia pasiva se puede utilizar en todo momento sin riesgo de causar interrupciones en el servicio. (8)

1.3.5. Protocolos utilizados en el auto-descubrimiento

SNMP

El Protocolo Simple de Gestión de Red (SNMP) es parte de la suite de protocolos TCP / IP y fue diseñado para facilitar el intercambio de información de administración entre dispositivos de red. Es un protocolo de la capa de aplicación que fue diseñado para operar a través de UDP.

ICMP

El Protocolo de Control de Mensajes de Internet (ICMP) maneja varios tipos de mensajes que asisten a las comunicaciones TCP / IP. Por ejemplo, se puede utilizar para determinar si una máquina está activa en la red. Una solicitud de eco ICMP se envía a una dirección IP (por ejemplo, a través de ping), y si hay una máquina en esa dirección, enviará de vuelta una respuesta de eco ICMP.

TCP

El Protocolo de Control de la Transmisión (TCP) fue diseñado para proporcionar una comunicación fiable y orientada a la conexión sobre la capa IP. Implementa mensajes de control y señalización para establecer,

mantener y terminar una conexión entre dos nodos. Las técnicas activas utilizan los mensajes de señalización de TCP para descubrir información. Además, se pueden realizar pruebas a los puertos para descubrir cuáles son los servicios de red que responden. Esto revela qué servicios se están ejecutando y qué puertos están abiertos o cerrados.

ARP

El Protocolo de Resolución de Direcciones (ARP) es usado para mapear las direcciones IP y MAC[9]. (12) La dirección MAC es requerida por cada paquete que se transmite. Cuando un host no conoce la dirección MAC del siguiente salto, difunde una solicitud ARP preguntando la dirección MAC asociada con la dirección IP de su interés. La máquina con la dirección IP de interés da respuesta a la petición ARP y proporciona su dirección MAC. Por lo tanto, el protocolo ARP es utilizado por las técnicas activas para obtener las direcciones MAC de los nodos IP descubiertos. (13)

DNS

El Sistema de Nombres de Dominio (DNS), establece una correspondencia entre direcciones IP numéricas y nombres de host, a través de una base de datos distribuida. Las técnicas activas utilizan los servicios de DNS para encontrar el nombre de un host a partir de la dirección IP o viceversa, poniéndose en contacto con el servidor de nombres. Esto permite proporcionar las etiquetas de nombre de host para los nodos descubiertos.

NetBIOS

El Sistema de Entrada y Salida Básica de Red (NetBIOS) fue diseñado para soportar las comunicaciones y la transferencia de datos entre aplicaciones. NetBIOS puede ser utilizado para descubrir los recursos ofrecidos por un servidor de red.

1.4. Situación Problemática

En el centro de desarrollo de GeoInformática y Señales Digitales (GEySED) de la facultad 6 de la Universidad de las Ciencias Informáticas se encuentra en proceso de desarrollo el Sistema de Vídeo

[9] Dirección MAC: Es un identificador hexadecimal de 48 bits que se corresponde de forma única con una tarjeta o interfaz de red.

Vigilancia Suria. En la actualidad cuando se configura una cámara por primera vez, se anotan en un papel o en un documento digital la dirección IP, el modelo de la cámara y alguna otra información necesaria. Luego, estos datos son introducidos en el sistema manualmente, lo cual no representa ningún problema cuando el número de cámaras es pequeño. Pero, ¿qué sucede cuando el sistema tiene que controlar una gran cantidad de cámaras? El proceso de incorporación de estas resulta entonces tedioso, y se da lugar a la aparición de errores, pues el administrador o la persona encargada de la tarea puede confundir una dirección o algún otro dato, imposibilitando así que el sistema sea capaz de controlar las cámaras.

1.5. Análisis de las soluciones existentes.

En el mundo existen varias compañías dedicadas a la producción de software de gestión de vídeo vigilancia, algunos de los sistemas comercializados por estas compañías vienen acompañados de una utilidad para encontrar las cámaras IP en la red. Los fabricantes de cámaras también brindan la posibilidad de encontrar las mismas en la red. A continuación se hace un análisis de estas soluciones para evaluar si es factible la utilización de una de ellas en el Sistema Suria.

Axis Camera Station:

AXIS Camera Station es un software de gestión de vídeo especialmente diseñado para las cámaras IP y los servidores de vídeo de Axis. Entre sus funcionalidades cuenta con la búsqueda de cámaras y servidores de vídeo presentes en la red. Lista las cámaras que se encuentran en la subred local, así como aquellas conectadas a un enrutador que soporte tráfico multidifusión. De las cámaras o servidores de vídeo muestra la siguiente información. (14)

- ✓ Nombre.
- ✓ Dirección IP.
- ✓ Modelo.
- ✓ Puerto de Vídeo.

IVTVision VMS Software

Inaxsys, con sede en Saint-Léonard, Québec, es una compañía desarrolladora de Sistemas de Gestión de Vídeo para aplicaciones de seguridad y vigilancia. Entre sus productos se encuentra IVTVision VMS Software. (15)

IVTVision VMS Software es una solución de vídeo vigilancia que soporta cámaras analógicas, así como cámaras IP de múltiples fabricantes. Se basa en una arquitectura cliente-servidor. Mediante la función Inaxsys IP Camera Finder, brinda la posibilidad de encontrar automáticamente cámaras de los fabricantes Axis, IQinVision, Sony, Panasonic, ACTi, Vivotek, Arecont, ioimage, Stardot, Ganz, Basler, IPX. Para que las cámaras sean detectadas deben estar ubicadas en la misma subred que el Servidor IVTVision. Después de realizado el descubrimiento, de cada cámara encontrada se muestra la siguiente información:

- ✓ Dirección IP.
- ✓ Fabricante.
- ✓ Modelo.
- ✓ Dirección MAC.
- ✓ Versión del Firmware.

AutoIP™

GVI Security Inc. es un proveedor de soluciones de seguridad de vídeo vigilancia que cuenta con un producto llamado AutoIP™. AutoIP™ es un sistema de gestión de vídeo de plataforma abierta que entre sus funciones realiza la de detectar y configurar automáticamente las cámaras en la misma subred. Los fabricantes soportados por esta aplicación son Samsung, Axis y Sony.

Aplicación “Find Local Cameras” para cámaras Lumenera

Lumenera Corporation es un desarrollador y fabricante de cámaras digitales de alto rendimiento utilizadas en todo el mundo para aplicaciones de vigilancia. (16)

Lumenera ofrece una aplicación llamada " Find Local Cameras "(en Español, Encontrar cámaras locales) que envía un mensaje de multidifusión DNS identificando las cámaras IP. La respuesta contiene las direcciones IP y MAC así como la descripción de la cámara. Esta aplicación no encuentra las cámaras que están fuera de la subred del host donde se ejecuta la aplicación.

XProtect

Milestone Systems es la compañía líder en el desarrollo de software de plataforma abierta para la gestión de la vídeo vigilancia IP. (17)

Las soluciones Xprotect, de esta compañía, soportan una gran cantidad de hardware de fabricantes como ACTi, Arecont, Axis, Bosch, D-Link, FLIR, Infinova, IQinVision, JVC, Lumenera, Mobotix, Panasonic, Pelco, Pixord, Sony, Samsung, Sanyo, Toshiba, VCS, Verint, Vivotek. Realiza descubrimiento automático de cámaras mediante la utilización de Universal Plug-and-Play, difusión y escaneo de rango de IP.

Honeywell IP Utility

Herramienta que descubre automáticamente cámaras de red Honeywell. Encuentra todas las cámaras de este fabricante, incluyendo aquellas que están en otra subred. Para que esto suceda el conmutador o enrutador debe permitir tráfico multidifusión. Después del descubrimiento inicial, la aplicación detecta si alguna nueva cámara es instalada o si alguna de las existentes es retirada.

1.5.1. Conclusiones del análisis de las soluciones existentes.

No se encontró ninguna solución que solo realice descubrimiento de cámaras, para poder integrarla con el sistema que está actualmente en proceso de desarrollo. Las soluciones de las grandes compañías analizadas aquí son funcionalidades ofrecidas por aplicaciones propietarias mucho más grandes y, por lo tanto, son altamente dependientes de estas, haciendo imposible su uso en el sistema Suria. Las aplicaciones provistas por los fabricantes como Axis, Lumenera y Honeywell encuentran únicamente sus propias cámaras, lo cual no es factible debido a la gran variedad de marcas y modelos que debe soportar el sistema. Se necesita una herramienta que brinde soporte para una gran cantidad de modelos y que sea extensible con la finalidad de añadir soporte para nuevos modelos de diferentes fabricantes cuando sea necesario. Es importante, además, que la herramienta descubra las cámaras en cualquier subred de una LAN, solo especificando un rango de direcciones IP, sin importar qué equipamiento existe en la red.

1.6. Conclusiones

Se logró una mayor comprensión del problema mediante el estudio de los conceptos asociados al dominio del mismo, la caracterización del objeto de estudio y el análisis de otras cuestiones teóricas de importancia. Además, después del análisis de algunas aplicaciones que fueron evaluadas como posibles soluciones, se pudo determinar que se necesita desarrollar una aplicación propia que posea las características principales que permitan dar solución al problema que existe actualmente en el Sistema Suria.

CAPÍTULO 2. **TENDENCIAS Y TECNOLOGÍAS ACTUALES.**

2.1. Introducción

El progreso de la informática ha ido en aumento y con él el perfeccionamiento de las herramientas y tecnologías para la construcción de software, que constituyen hoy instrumentos fundamentales para garantizar la celeridad y calidad de los procesos de desarrollo.

En este capítulo se exponen las características de la metodología de desarrollo, herramienta CASE, lenguaje de programación, marco de trabajo y Entorno de Desarrollo Integrado seleccionados para el desarrollo del presente trabajo.

2.2. Metodología de Desarrollo

Una metodología es “un conjunto de procedimientos, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software” (18)

Si se quiere construir un software de alta calidad, desarrollado en el tiempo planificado y con los costes establecidos, pero que además satisfaga la necesidad de ser elaborado de una forma más acelerada y logrando una reducción del costo del producto, es necesario enfocarse en trabajar de forma organizada, de manera tal que se controle y documente todo lo relacionado con el proyecto en cuestión y se puedan eliminar los riesgos que podrían presentarse durante el desarrollo del mismo, lo cual no sería posible sin el empleo de una metodología eficaz que se adapte a las características propias del software que se esté desarrollando. (19)

2.2.1. Proceso Unificado de Desarrollo

El Proceso Unificado es un proceso de ingeniería del software que proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo. Su propósito es asegurar la producción de software de alta calidad que se ajuste a las necesidades de sus usuarios finales con costos y calendario predecibles.

Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyectos. (20)

El Proceso Unificado de Desarrollo es:

Basado en Componentes: El sistema software en construcción está basado en componentes software interconectados a través de interfaces bien definidas. (20)

Dirigido por casos de uso: El proceso de desarrollo avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan y los casos de uso finales son la fuente a partir de la cual se construyen los casos de prueba.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. (20)

Iterativo e incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. (20)

Se utilizará RUP como metodología de desarrollo pues esta es muy detallada, lo que posibilita documentar todo el proceso de forma minuciosa. Se ajusta tanto a proyectos grandes como a pequeños, siendo así muy conveniente su elección y utilización. Aunque genera un gran volumen de información, esto permite un mayor entendimiento entre los miembros del equipo de desarrollo. La gran cantidad de artefactos que se generan contribuyen a una mejor comprensión del problema y la solución.

2.3. Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.

Se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Da apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (21)

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos. (21)

2.4. Herramienta CASE.

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste del mismo en términos de tiempo y dinero. (22)

Las herramientas CASE proporcionan al ingeniero la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Al igual que las herramientas de ingeniería y de diseño asistidos por computadora que utilizan los ingenieros de otras disciplinas, las herramientas CASE ayudan a garantizar que la calidad se diseñe antes de llegar a construir el producto. (23)

El uso de una herramienta CASE brinda una serie de ventajas:

- ✓ Facilita la verificación y mantenimiento de la consistencia de la información del proyecto.
- ✓ Facilita el establecimiento de estándares en el proceso de desarrollo.
- ✓ Facilita el mantenimiento del sistema y las actualizaciones de la documentación.
- ✓ Facilita la aplicación de las técnicas de una metodología.
- ✓ Disponibilidad de funciones automatizadas tales como: obtención de prototipos, generación de código, generación de pantallas e informes, generación de diseños físicos de bases de datos, verificadores automáticos de consistencia.
- ✓ Facilita la aplicación de técnicas de reutilización y reingeniería.
- ✓ Facilita la planificación y gestión del proyecto informático.

2.4.1. Enterprise Architect

Para el modelado de los artefactos generados a lo largo del ciclo de vida del proyecto se decidió emplear Enterprise Architect 7.0.

Esta es una herramienta de uso muy sencillo, que aborda el diseño y análisis UML y cubre el desarrollo de software desde la captura de requerimientos a lo largo de las etapas de análisis, diseño, pruebas y

mantenimiento. Es una herramienta multiusuario, diseñada para ayudar a construir software robusto y fácil de mantener. Además, permite generar documentación e informes flexibles y de alta calidad.

Enterprise Architect proporciona trazabilidad completa desde el análisis de requerimientos y los artefactos de diseño, hasta la implementación y el despliegue.

Soporta la generación e ingeniería inversa de código fuente para muchos lenguajes, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript y PHP. Con un editor de código fuente con "marcador de sintaxis" incorporado, Enterprise Architect permite navegar y explorar el modelo de código fuente en un mismo entorno completamente integrado. Las plantillas de generación de código permiten personalizar el código fuente generado de acuerdo a las necesidades de los desarrolladores.

Soporta los 13 diagramas de UML 2.1. Los diagramas de comportamiento incluyen: Casos de Uso, Actividades, Estado, Secuencia y Comunicación. Los diagramas estructurales incluyen: Paquetes, Clases, Objetos, Composición, Componentes y Despliegue.

Presenta una interfaz de usuario intuitiva con un amplio rango de barras de herramientas, ventanas acoplables, y estilos visuales. Guarda y restaura disposiciones de ventanas personalizadas. Desplaza las ventanas acopladas para maximizar el espacio de las ventanas y mejorar la eficiencia del trabajo.

2.5. Lenguaje de programación

Un lenguaje de programación es un *"conjunto de instrucciones, órdenes, comandos y reglas que permite la creación de programas"*. (24)

2.5.1. C++

Para llevar a cabo la implementación de la aplicación que se necesita se utilizará como lenguaje de programación C++, el cual es un lenguaje de programación muy difundido y popular que puede ser usado para resolver diferentes tipos de problemas, desde los más simples hasta los más complejos, y cuyo código puede ser compilado en diversas plataformas.

Entre las características que lo hacen un lenguaje de programación atractivo se encuentran:

Programación orientada a objetos: La posibilidad de orientar la programación a objetos permite al programador diseñar aplicaciones como una comunicación entre objetos en lugar de en una secuencia

estructurada de código. Además, permite una mayor reutilización de código en una forma más lógica y productiva.

Portabilidad: Prácticamente se puede compilar el mismo código C++ en casi cualquier tipo de computadora y sistema operativo sin realizar ningún cambio.

Brevedad: El código escrito en C+ + es muy corto en comparación con otros lenguajes de programación, ya que el uso de caracteres especiales se prefiere a las palabras clave.

Programación Modular: El cuerpo de una aplicación en C+ + puede estar formado por varios archivos de código fuente que se compilan por separado y luego se unen. Esta característica permite vincular código C + + con el código producido en otros lenguajes, como Ensamblador o C.
Velocidad: El código resultante de una compilación de C+ + es muy eficiente, gracias a su capacidad de actuar como lenguaje de de alto y bajo nivel y a la reducida medida del lenguaje.

2.6. Marco de trabajo.

Como marco de trabajo para la implementación de la aplicación se seleccionó Qt, el cual es una intuitiva biblioteca de C++ que garantiza la portabilidad entre sistemas operativos embebidos y de escritorio, que presenta herramientas de desarrollo integrado con IDE multiplataforma y un alto desempeño en dispositivos embebidos.

Este marco de trabajo permite la utilización de hilos independientemente del sistema operativo, contiene un módulo para el trabajo con protocolos de red, posee soporte para aplicaciones orientadas a componentes, trabajo con los gestores de bases de datos más conocidos además de poder implementar soporte para otros. Estas características hacen de Qt un marco de trabajo altamente aplicable a cualquier aplicación de escritorio sin tener que utilizar bibliotecas externas.

2.7. Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado (IDE) es una herramienta informática que ofrece servicios integrales a los programadores para el desarrollo de software. Un IDE normalmente cuenta con un editor de código fuente, un depurador, un compilador y / o intérprete y un constructor de Interfaz grafica de usuario. (25)

2.7.1. Qt Creator

El Entorno de Desarrollo Integrado que se empleará será Qt Creator.

Qt Creator es un completo Entorno de Desarrollo Integrado para la creación de aplicaciones con el marco de trabajo Qt.

Una de las principales ventajas de Qt Creator es que le permite a un equipo de desarrolladores trabajar con un proyecto en diferentes plataformas (Microsoft Windows, MacOS X, y Linux), con una herramienta común.

Las características principales de Qt Creator permiten a los desarrolladores realizar las tareas siguientes:

- ✓ Desarrollar aplicaciones Qt rápida y fácilmente con asistentes de proyectos, así como acceder rápidamente a los últimos proyectos y sesiones.
- ✓ Diseñar interfaces de usuario con el editor integrado Qt Designer.
- ✓ Desarrollar aplicaciones con el avanzado editor de código para C++ que proporciona características de gran alcance para completar fragmentos y refactorización de código.
- ✓ Generar, ejecutar e implementar proyectos de Qt dirigidos a plataformas móviles y de escritorio, tales como Microsoft Windows, Mac OS X, Linux, Symbian, Android y Maemo.
- ✓ Depurar el código mediante los depuradores GDB[10] y CDB[11] usando una interfaz gráfica que permite un mayor entendimiento de la estructura de clases de Qt.
- ✓ Acceder fácilmente a la documentación mediante el uso de la ayuda contextual integrada de Qt.

2.8. Conclusiones.

En este capítulo se hizo un estudio de las herramientas y tecnologías actuales. Se definió el Proceso Unificado de Desarrollo como metodología a utilizar, Enterprise Architect como herramienta CASE, como lenguaje de programación C++, como marco de trabajo se escogió Qt, y como Entorno de Desarrollo Integrado se eligió QtCreator. Mediante la utilización de estas herramientas, será posible la creación de un software robusto que se ajuste a las necesidades del proyecto.

[10] GNU Symbolic Debugger.
[11] Microsoft Console Debugger.

CAPÍTULO 3. **CARACTERÍSTICAS DEL SISTEMA.**

3.1. Introducción.

En este capítulo se presenta la propuesta de solución. Se muestra el Modelo de Dominio, en el cual se expone un marco conceptual y las relaciones entre las definiciones, para ayudar al entendimiento del entorno donde se desarrolla la aplicación. Se definen los requerimientos funcionales y no funcionales. Se presenta el diagrama de casos de uso del sistema, así como las descripciones de los casos de uso críticos para el mismo.

3.2. Modelo de Dominio.

Por no existir un negocio real, y al no poder precisar la estructura de los procesos de negocio, se emplea un modelo de dominio o modelo conceptual. *"Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema."* (20) El modelo de dominio es una forma de mostrar al usuario los principales conceptos que se tratan en el entorno, logrando un mejor entendimiento del sistema.

3.2.1. Diagrama de Clases del Modelo de Dominio.

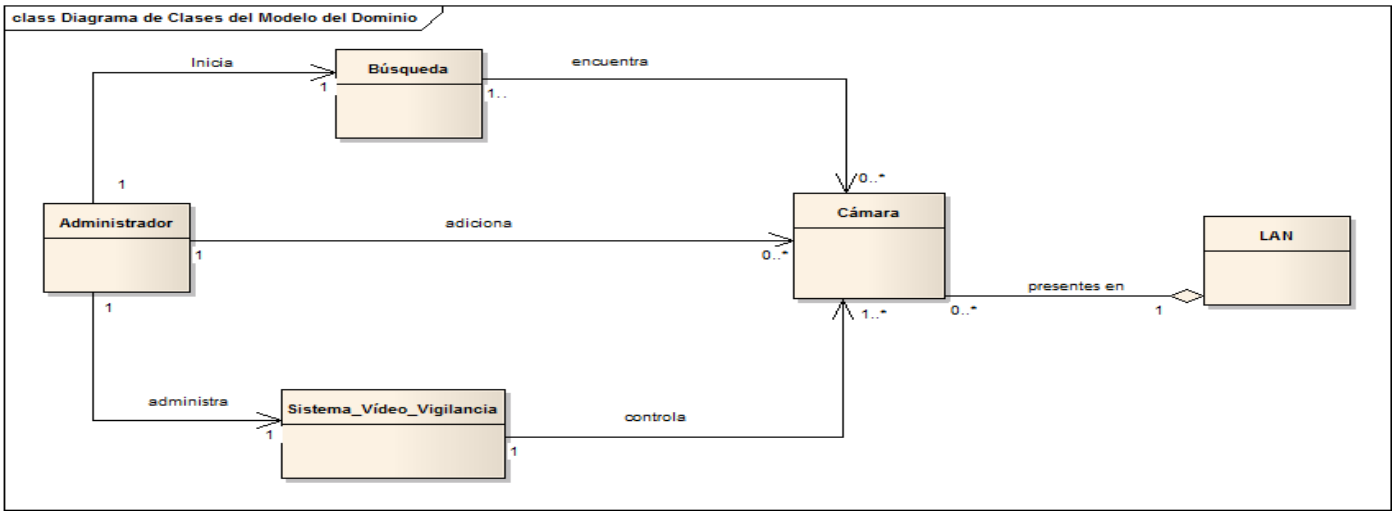


Figura 1. Diagrama de clases del dominio

Con el objetivo de que el sistema Suria sea capaz de controlar las cámaras IP conectadas en una Red de Área Local, el administrador inicia una búsqueda mediante la cual encuentra estas cámaras. Luego, puede adicionarlas al sistema, permitiendo así que este pueda realizar el resto de las operaciones con ellas.

3.2.2. Descripción de las clases.

- Administrador: Es el encargado de administrar el sistema de Vídeo Vigilancia. Es quien realiza las búsquedas para adicionar las cámaras al sistema.
- Búsqueda: Proceso de búsqueda de cámaras que se realiza en una Red de Área Local.
- Cámara: Dispositivo de captura de imagen y vídeo.
- LAN: Es la Red de Área Local a la que están conectadas las cámaras.
- Sistema_Vídeo_Vigilancia: Representa al sistema de Vídeo Vigilancia Suria.

3.3. Requerimientos del sistema.

Los requisitos, de manera general, son una descripción de lo que se necesita o desea de un producto. La meta primaria de la fase de requisitos es identificar y documentar lo que en realidad se necesita, en una forma clara y entendible tanto para los clientes como para los miembros del equipo de desarrollo.

3.3.1. Requerimientos funcionales.

Los Requerimientos Funcionales son las capacidades o condiciones que el sistema debe cumplir para darle respuesta a las necesidades de los clientes.

R.F.1 Descubrir cámaras IP.

El sistema debe permitir encontrar las cámaras en un rango de direcciones IP definido por el administrador.

R.F.2 Configurar Rangos de IP.

El sistema debe permitir que el administrador adicione, modifique, elimine y seleccione los rangos de IP en los que se va a realizar el descubrimiento.

R.F.2.1 Adicionar Rango de IP.

R.F.2.2 Eliminar Rango de IP.

R.F.2.3. Modificar Rango de IP.

R.F.2.4 Seleccionar Rango de IP.

R.F.3 Configurar Rangos de Puertos.

El sistema debe permitir que el administrador adicione, modifique, elimine y seleccione los rangos de puertos que se probarán inicialmente para cada dirección IP.

R.F.3.1 Adicionar Rango de Puertos.

R.F.3.2 Eliminar Rango de Puertos.

R.F.3.3. Modificar Rango de Puertos.

R.F.3.4 Seleccionar Rango de Puertos.

R.F.4. Gestionar Credenciales

El sistema debe permitir que los administradores añadan, modifiquen y eliminen los conjuntos usuario-contraseña que servirán para descubrir las cámaras.

R.F. 4.1. Adicionar Credencial.

R.F. 4.2. Modificar Credencial.

R.F. 4.3. Eliminar Credencial.

R.F.5. Modificar Cámara.

El sistema debe permitir que el administrador modifique los datos de una cámara, los cambios se efectuarán en el mismo dispositivo.

R.F.6. Adicionar cámara al Sistema

El sistema debe permitir que el administrador adicione la información de una cámara al Sistema Suria.

R.F.7. Autenticar Usuario

El sistema permitirá que los usuarios se autentiquen de forma segura para acceder al sistema, solo permitirá el acceso a los administradores del Sistema Suria.

3.3.2. Requerimientos no funcionales.

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.

A continuación se presentan los requisitos no funcionales de la solución.

Usabilidad

La aplicación debe tener una interfaz amigable, de manera que sea atractiva al usuario y que brinde además un fácil acceso a todas las operaciones del sistema.

El sistema deberá validar que la información introducida sea correcta. Además debe validarse que el tipo de dato se corresponda con lo previsto para el campo.

Eficiencia

Se debe lograr un óptimo aprovechamiento de los recursos de hardware.

Hardware

El hardware de la PC donde se ejecute la aplicación debe tener una capacidad de 1GB de memoria RAM; además de un CPU de varios núcleos a 2.4GHz.

Seguridad

- **Confidencialidad:** la información manejada por el sistema está protegida de acceso no autorizado y divulgación.
- **Integridad:** la información manejada por el sistema es objeto de cuidadosa protección contra la corrupción y estados inconsistentes.
- **Disponibilidad:** los usuarios autorizados (autenticados) se les garantizará el acceso a la información.

Restricciones de diseño

Se debe desarrollar la aplicación en el lenguaje C++, utilizando el marco de trabajo Qt en su versión 4.7.

3.4. Descripción de la solución propuesta.

La solución será una aplicación de escritorio cuya principal funcionalidad será descubrir las cámaras en una Red de Área Local. Además, brindará la posibilidad de modificar una cámara y añadirla al sistema. Los administradores del sistema de Vídeo Vigilancia serán los únicos con acceso a la aplicación, la cual

será basada en componentes, permitiendo, de esta manera, añadir soporte para diferentes marcas y modelos en la medida que se necesite.

Para la realización de algunas de las funcionalidades de la aplicación se necesita el uso del Gestor del Sistema Suria, debido a que este módulo es el único con acceso a la base de datos del sistema de modo tal que cualquier intento de cambio debe pasar por él.

3.4.1. Descripción de los actores del sistema.

Actor	Descripción
Administrador	Rol responsable de velar por el funcionamiento apropiado de la aplicación, es el único que puede realizar todas las funcionalidades que permite la misma.

Tabla 1. Descripción de los actores.

3.4.2. Diagrama de casos de uso del sistema.

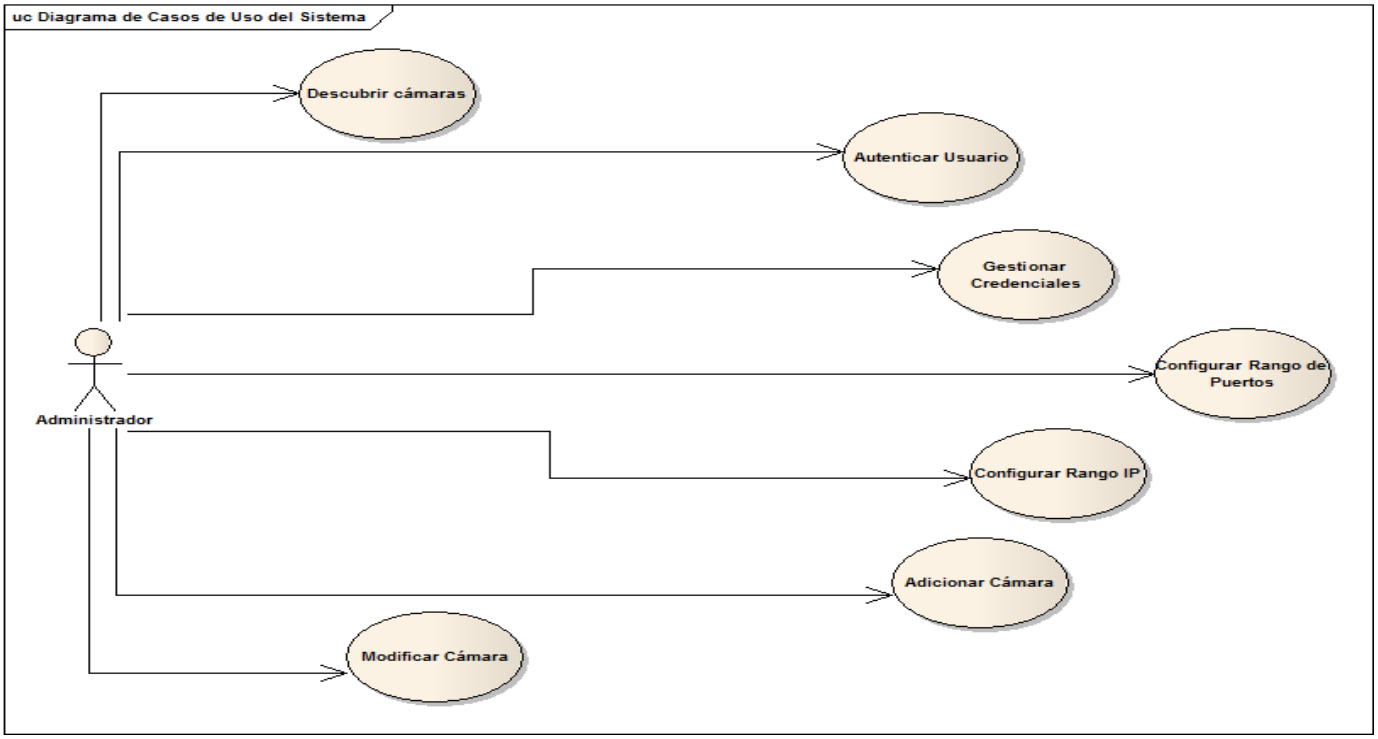


Figura 2. Diagrama de casos de uso del sistema

3.4.3. Descripción detallada de los casos de uso del sistema.

A continuación se describen los casos de uso críticos para el sistema. Las descripciones del resto de los casos de uso se pueden consultar en el Anexo 1: Descripción de los casos de uso no críticos del sistema.

3.4.3.1. Descripción detallada del caso de uso Descubrir Cámaras.

<table>
<tr><td>Caso de Uso:</td><td colspan="2">Descubrir Cámaras</td></tr>
<tr><td>Actores:</td><td colspan="2">Administrador</td></tr>
<tr><td>Resumen:</td><td colspan="2">El caso de uso se inicia cuando el Administrador selecciona la opción “Descubrir cámaras”.</td></tr>
<tr><td>Precondiciones:</td><td colspan="2">El actor debe haber seleccionado un rango de direcciones IP.</td></tr>
<tr><td>Referencias</td><td colspan="2">R.F.1</td></tr>
<tr><td>Prioridad</td><td colspan="2">Crítico</td></tr>
<tr><td colspan="3">Flujo Normal de Eventos</td></tr>
<tr><td>Acción del Actor</td><td colspan="2">Respuesta del Sistema</td></tr>
<tr><td>1. El actor selecciona la opción “Descubrir cámaras”.</td><td colspan="2">2. El sistema escanea el rango de direcciones IP.
2.1. El sistema examina cada dirección IP de los rangos seleccionados.
2.2. Si en una dirección IP hay una cámara, obtiene los datos de esta, y los añade al listado de cámaras.
2.3. El sistema actualiza la interfaz mostrando el listado de cámaras modificado.</td></tr>
<tr><td colspan="3">Prototipo de Interfaz</td></tr>
</table>

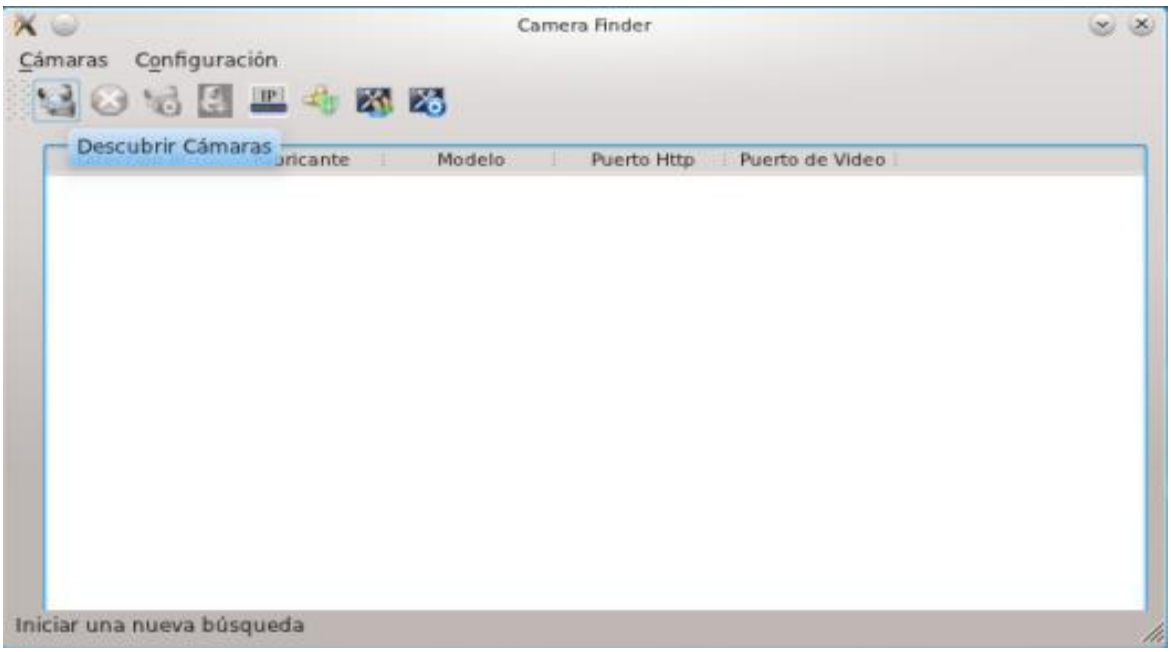
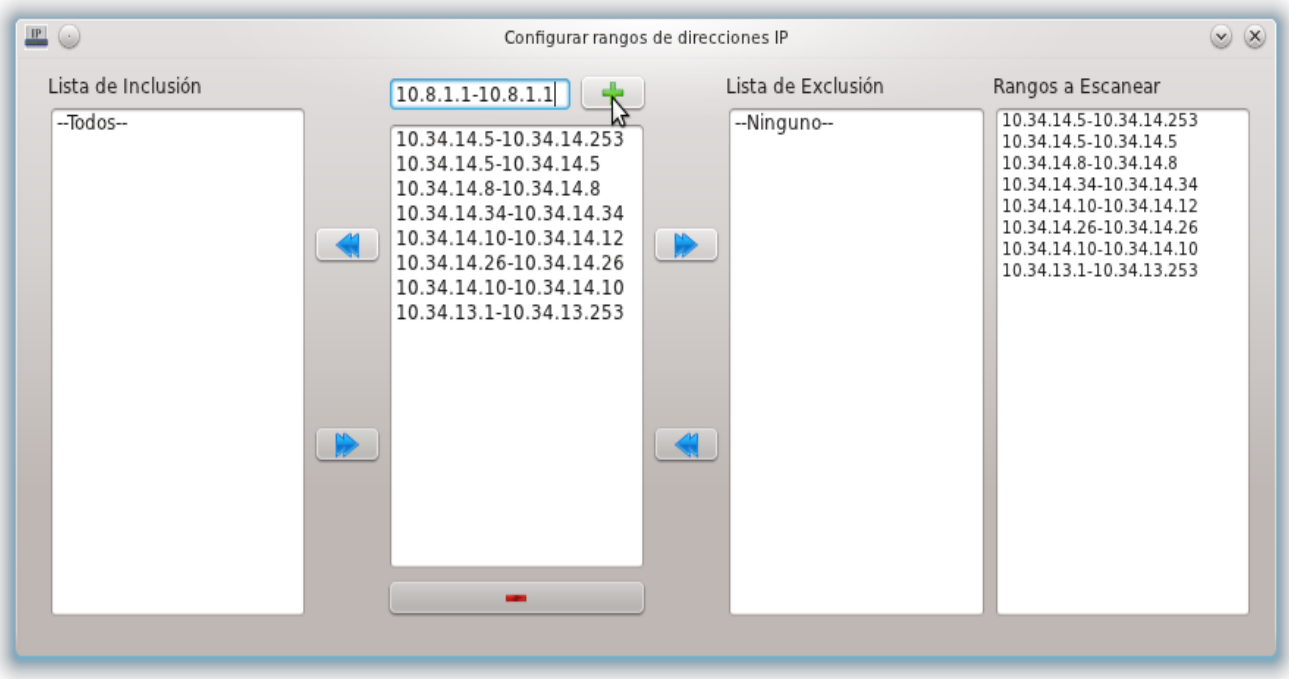
<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td colspan="3">Flujos Alternos</td></tr>
<tr><td colspan="2">Acción del Actor</td><td>Respuesta del Sistema</td></tr>
<tr><td colspan="2"></td><td>2.3. El sistema muestra un mensaje informando que ninguna cámara fue encontrada.</td></tr>
<tr><td>Poscondiciones</td><td colspan="2">Se muestra un listado con las cámaras encontradas, sobre las cuales se pueden realizar otras operaciones.</td></tr>
</table>

Tabla 2. Descripción textual del caso de uso Descubrir Cámaras

3.4.3.2. Descripción detallada del caso de uso Configurar Rango de IP.

Caso de Uso:	Configurar rangos de IP
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el Administrador selecciona la opción “Configurar rangos de IP”.
Precondiciones:	

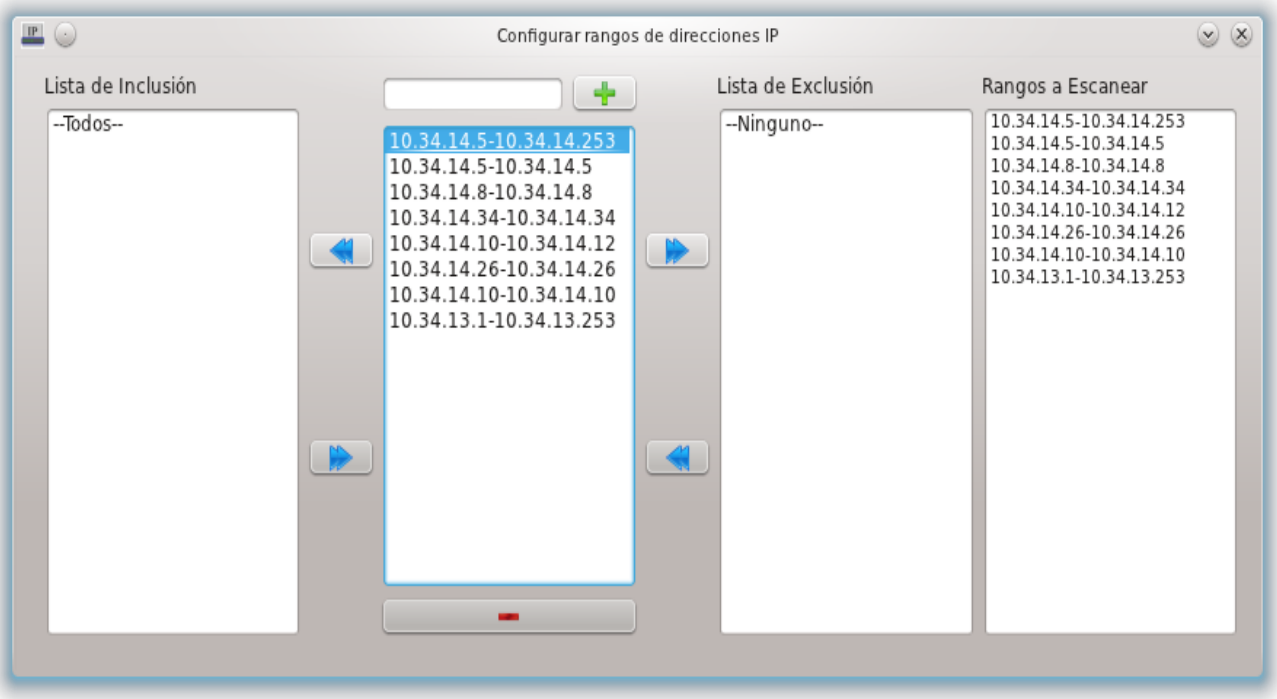
Referencias	R.F.2,R.F.2.1,R.F.2.2,R.F.2.3,R.F.2.4
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “Adicionar Rango de IP”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción “Adicionar rango de IP”.	2. El sistema muestra un cuadro de texto donde el usuario debe introducir el IP inicio y el IP fin, separados por un guión.
3. El actor introduce los datos y presiona el botón Adicionar.	4. El sistema comprueba que no existen campos vacíos. 5. El sistema comprueba que el rango introducido es válido. 6. El sistema comprueba que el rango no exista. 7. El sistema actualiza el fichero de los rangos con el rango introducido por el usuario.
Prototipo de Interfaz	



Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4. El sistema muestra un mensaje de error informando que hay campos vacíos. 5. El sistema muestra un mensaje de error informando que el rango introducido no es válido. 6. El sistema muestra un mensaje de error informando que el rango ya existe.
Sección “Eliminar Rango de IP”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona un rango de IP y	2. El sistema elimina el rango de IP.

marca la opción “Eliminar”.	

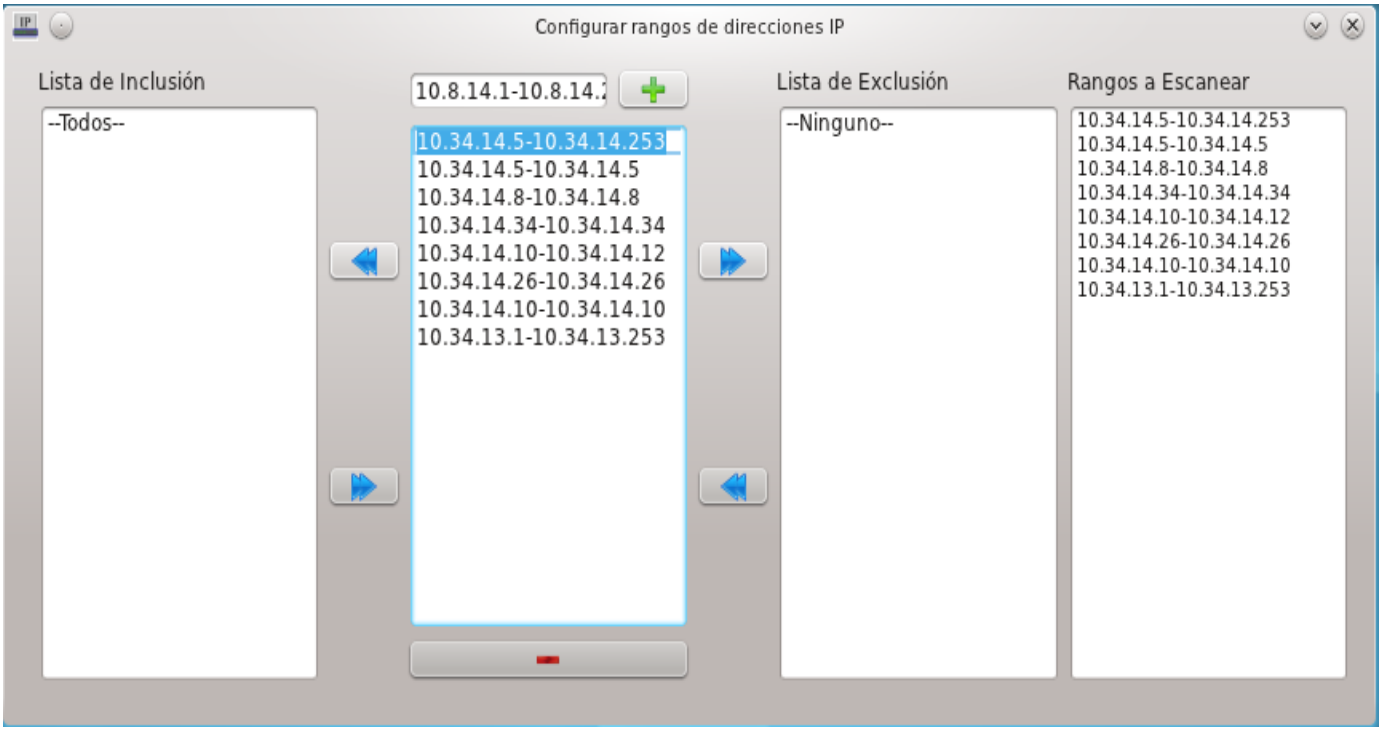
Prototipo de Interfaz



Sección “Modificar Rango de IP”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona un rango de IP y marca la opción “Modificar rango de IP”.	2. El sistema muestra un cuadro de texto con el rango seleccionado.
3. El actor introduce los datos y da doble click.	4. El sistema comprueba que no existen campos vacíos. 5. El sistema comprueba que el rango introducido es válido.

	6. El sistema comprueba que el rango no exista. 7. El sistema actualiza el fichero de los rangos con el rango modificado por el usuario.

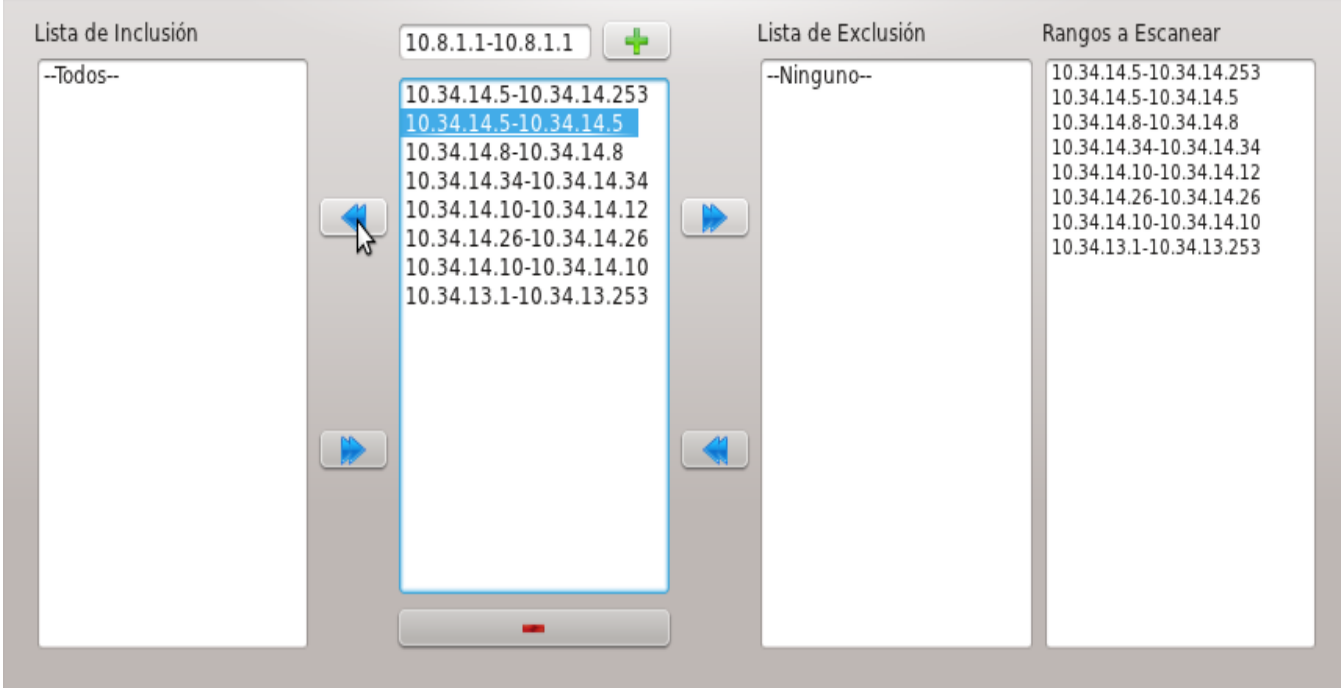
Prototipo de Interfaz



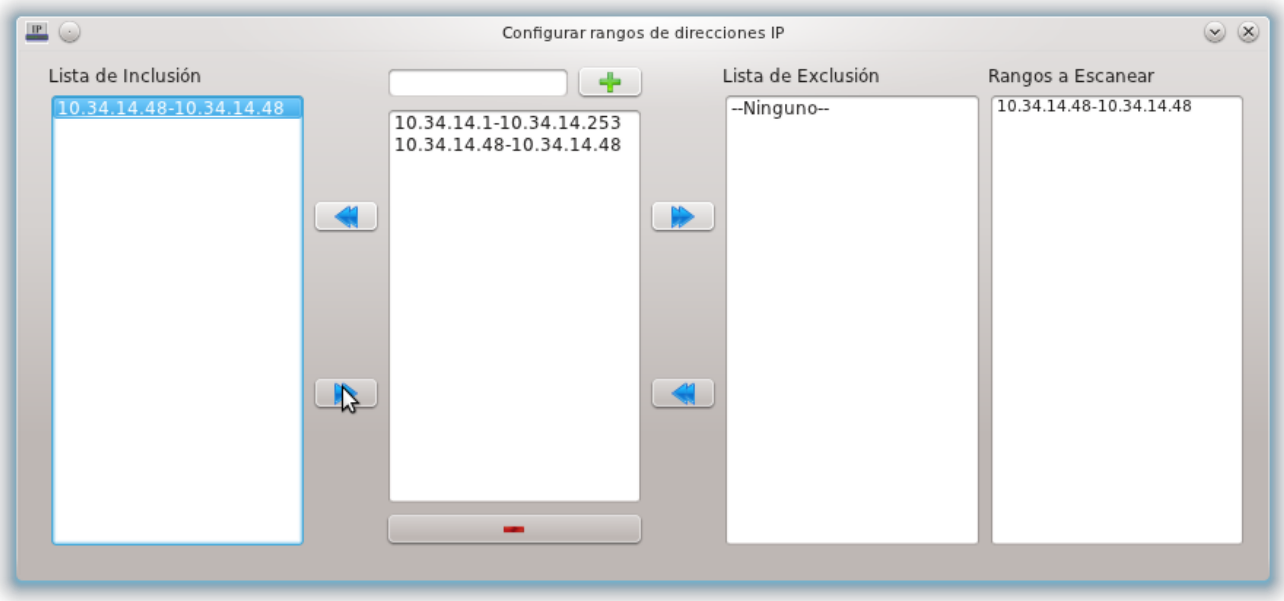
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4. El sistema muestra un mensaje de error informando que hay campos vacíos. 5. El sistema muestra un mensaje de error informando que el rango introducido no es válido. 6. El sistema muestra un mensaje de error informando que el rango ya existe.

Sección “Añadir rango a la lista de inclusión”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona un rango de direcciones IP que desea escanear y presiona el botón << para añadirlos a la lista de inclusión.	2. Si el rango no está en la lista de inclusión, el sistema lo añade a la misma. 3. El sistema calcula los rangos que se escanearán y actualiza los mismos para que el usuario los pueda visualizar.

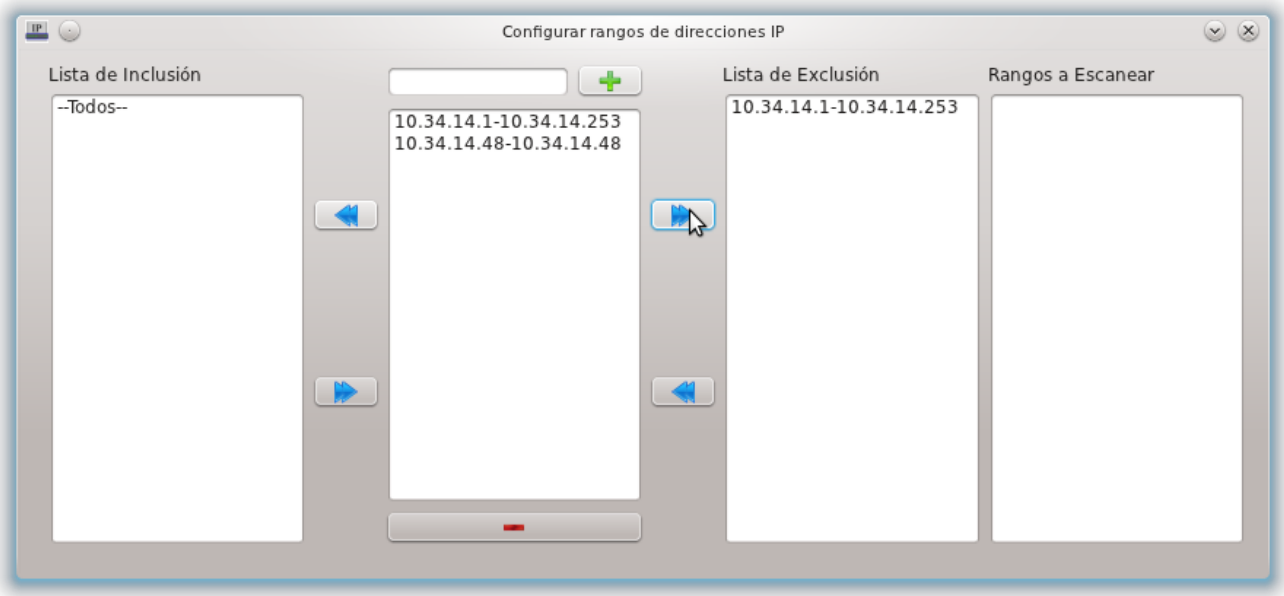
Prototipo de Interfaz



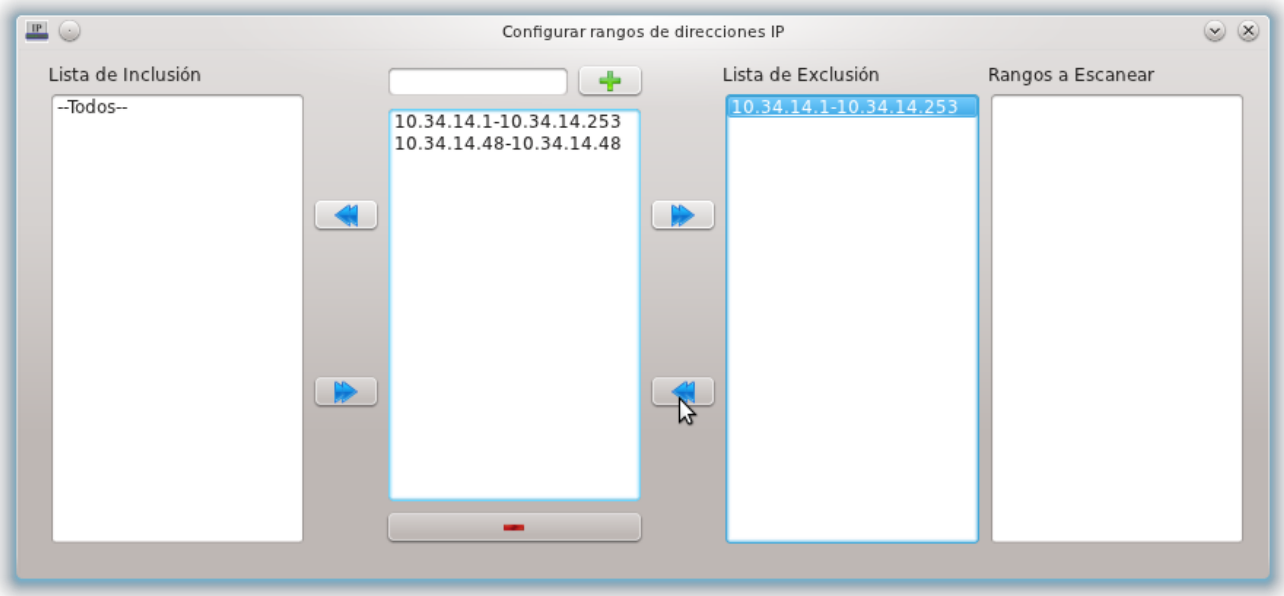
Sección “Eliminar rango de la lista de inclusión”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona un rango de la lista de inclusión y presiona el botón >> para eliminarlos de la misma.	2. El sistema elimina el rango de la lista de inclusión, calcula los rangos que se escanearán y muestra la lista de estos al usuario.

Prototipo de Interfaz


Sección “Añadir rango a la lista de exclusión ”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona un rango de direcciones IP que no desea escanear y presiona el botón >> para añadirlos a la lista de exclusión.	2. Si el rango no está en la lista de exclusión, el sistema lo añade a la misma. 3. El sistema calcula los rangos que se escanearán y actualiza los mismos para que el usuario los pueda visualizar.
Prototipo de Interfaz	



Sección “Eliminar rango de la lista de exclusión”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona un rango de la lista de exclusión y presiona el botón << para eliminarlos de la misma.	2. El sistema elimina el rango de la lista de exclusión, calcula los rangos que se escanearán y muestra la lista de estos al usuario.
Prototipo de Interfaz	



Poscondiciones	Se actualiza el fichero de configuración de rangos de direcciones IP y quedan seleccionados los rangos en los que se efectuará el descubrimiento.

Tabla 3. Descripción textual del caso de uso Configurar rangos de IP

3.4.3.3. Descripción detallada del caso de uso Adicionar Cámara.

Caso de Uso:	Adicionar Cámara
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador selecciona una cámara de la lista de cámaras y presiona el botón “Adicionar cámara al sistema”.
Precondiciones:	Se debe haber invocado el caso de uso “Descubrir cámaras” previamente.
Referencias	R.F.5
Prioridad	Crítico
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1. El actor selecciona una cámara y presiona el botón “Adicionar Cámara al Sistema”.	2. El sistema muestra una ventana con los datos de la cámara, con los botones “Aceptar” y “Cancelar”.
3. El usuario presiona el botón “Aceptar”.	4. El sistema envía un mensaje al gestor con los datos de la cámara. 5. El gestor guarda los datos de la cámara, notifica a la aplicación, y esta confirma la operación al usuario.

Prototipo de Interfaz

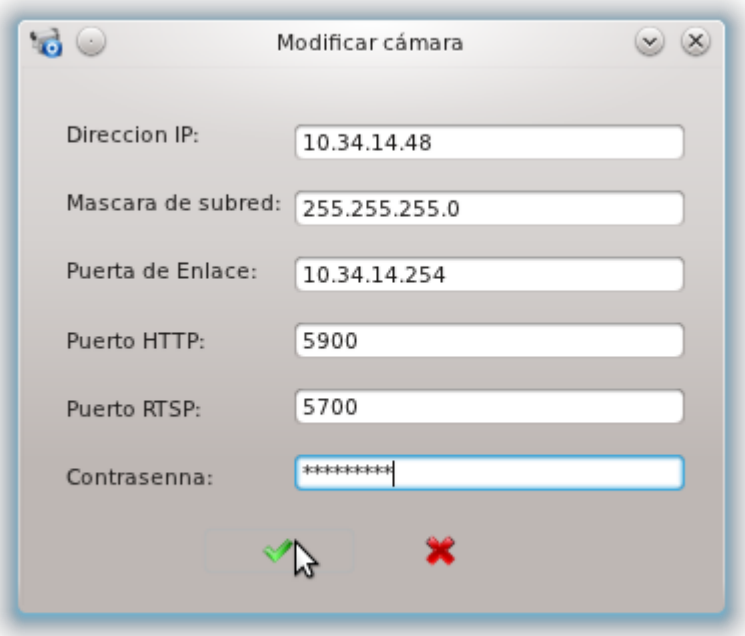
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El usuario presiona el botón “Cancelar”.	4. El Sistema muestra la pantalla principal.

	4. El sistema muestra un mensaje de error informando que la cámara no pudo ser añadida al Sistema Suria.
Poscondiciones	La cámara seleccionada es añadida al sistema Suria

Tabla 4. Descripción textual del caso de uso Adicionar Cámara

3.4.3.4. Descripción detallada del caso de uso Modificar Cámaras.

Caso de Uso:	Modificar Cámara
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador selecciona una cámara de la lista de cámaras y presiona el botón “Modificar Cámara”.
Precondiciones:	Se debe haber invocado el caso de uso “Descubrir cámaras” previamente.
Referencias	R.F.6
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona una cámara y presiona el botón “Modificar Cámara”.	2. El sistema muestra una ventana con los campos llenos con los datos de la cámara.
3. El usuario modifica los datos y presiona el botón “Modificar”.	4. El sistema comprueba que no existen campos vacíos. 5. El sistema comprueba que los datos introducidos son válidos. 6. El sistema modifica los datos de la cámara.
Prototipo de Interfaz	



Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El usuario presiona el botón “Cancelar”.	4. El Sistema muestra la pantalla principal.
	4. El sistema muestra un mensaje de error informando que no pueden existir campos vacíos. 5. El sistema muestra un mensaje de error informando que algunos de los datos introducidos no son válidos.
Poscondiciones	Quedan modificados los datos de una cámara.

Tabla 5. Descripción textual del caso de uso Modificar Cámara

3.4.3.5. Descripción detallada del caso de uso Autenticar Usuario.

Caso de Uso:	Autenticar Usuario

Actores:	Administrador
Resumen:	Se inicia cuando el usuario inicializa la aplicación.
Precondiciones:	
Referencias	R.F.7
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor inicia la aplicación.	2. El sistema muestra una pantalla con los campos Usuario y Contraseña.
3. El actor llena los campos y presiona el botón “Entrar”.	4. El sistema comprueba que el usuario existe en la Base de Datos del Sistema Suria, que la contraseña coincide con la almacenada en esta y que tiene los permisos necesarios para acceder a la aplicación.

Prototipo de Interfaz

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	4. El sistema muestra un mensaje de error informando que el usuario o la contraseña son incorrectos, o que el usuario no tiene permisos para acceder a la aplicación.
Poscondiciones	El usuario tiene acceso a la aplicación.

Tabla 6. Descripción textual del caso de uso Autenticar Usuario

3.5. Conclusiones.

Se logró un mayor entendimiento del entorno donde se desarrolla el sistema mediante la realización del modelo de dominio, se definieron las capacidades y condiciones que el sistema debe cumplir. Se ha modelado el sistema propuesto contando con el diagrama de casos de uso y la descripción detallada de los mismos. Todo esto ofrece una visión general de la estructura y función de la solución que se propone.

CAPÍTULO 4. **ANÁLISIS Y DISEÑO DEL SISTEMA**

4.1. Introducción

En este capítulo se abordan temas fundamentales para la construcción del sistema propuesto. Se incluyen los modelos de análisis y diseño, los cuáles dejarán sentadas las bases para comenzar con la implementación de la solución. Esto se logrará a través de los diferentes diagramas de clases que incluyen dichos modelos. Además se explican los patrones de la arquitectura y el diseño empleados para garantizar la viabilidad y calidad requeridas por el sistema a construir.

4.2. Modelo de Análisis

El objetivo del análisis es lograr una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura. El modelo de análisis constituye una entrada fundamental cuando se da forma al sistema en el diseño y en la implementación y proporciona una visión general del sistema que puede ser más difícil de entender mediante el estudio de los resultados del diseño y la implementación, debido a que estos contienen demasiados detalles.

4.2.1. Diagramas de clases del análisis.

A continuación se presentan los diagramas de clases del análisis de los casos de uso Descubrir Cámaras, Autenticar Usuario y Configurar Rangos de IP, el resto de los diagramas de clases del análisis pueden ser consultados en los Anexos.

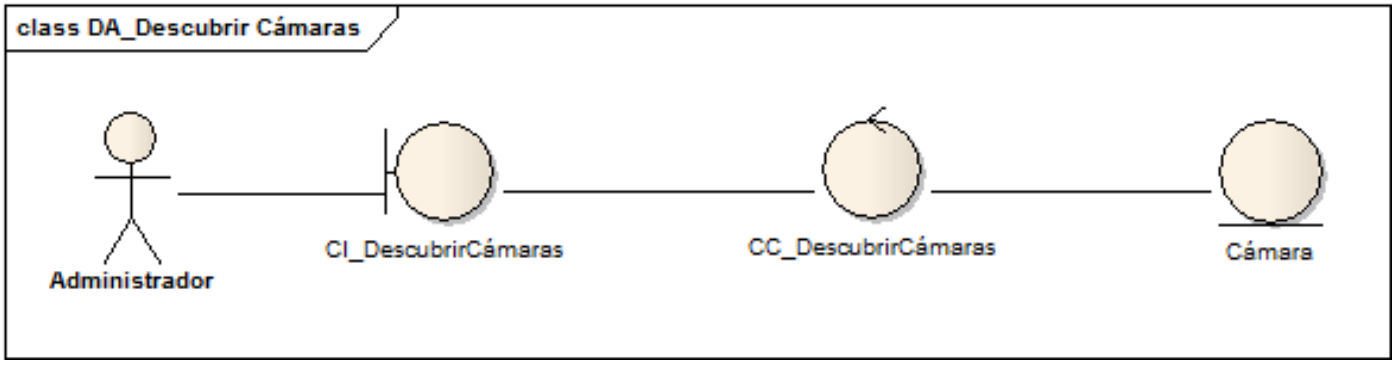


Figura 3. Diagrama de clases del análisis del caso de uso Descubrir Cámaras

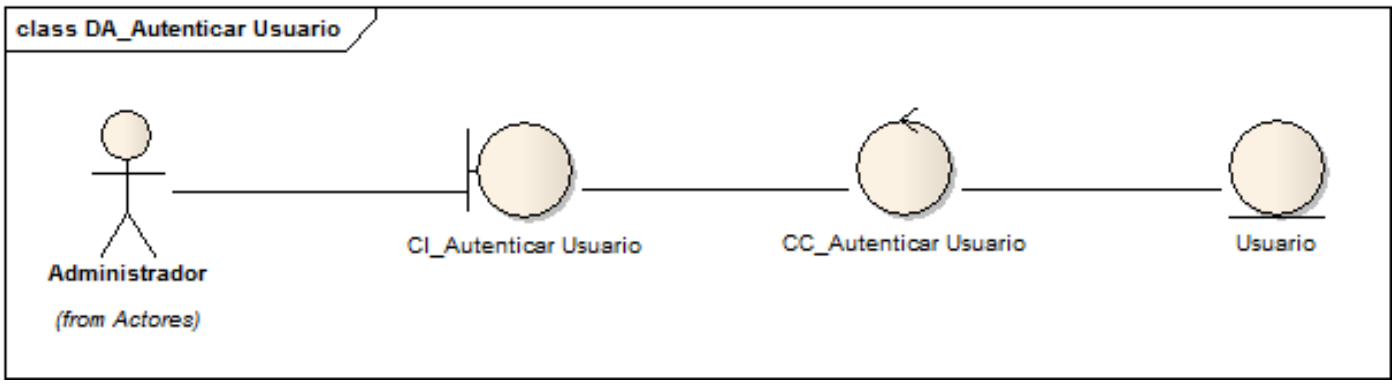


Figura 4. Diagrama de clases del análisis del caso de uso Autenticar Usuario

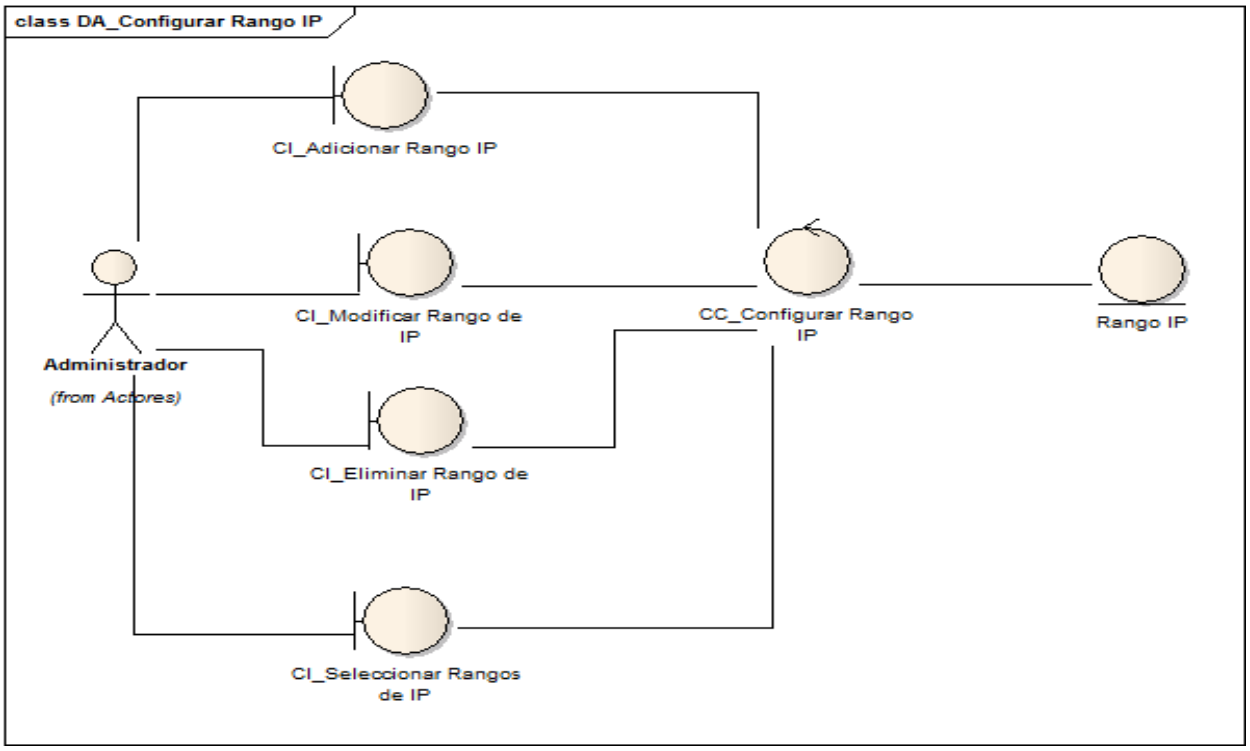


Figura 5. Diagrama de clases del análisis del caso de uso Configurar Rangos de IP

4.3. Diseño de la solución.

EL propósito del diseño es adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación. Es usado como entrada esencial en las actividades relacionadas con la implementación.

4.3.1. Arquitectura del Sistema Suria.

El sistema Suria está diseñado siguiendo una Arquitectura base en forma de pizarra, en su variante de tablero de control. Ésta desacopla el sistema en componentes denominados agentes autónomos, los cuales son independientes en la realización atómica de su funcionalidad pero dependen de una entrada de información externa, que es provista por otros agentes, y a su vez, producen un resultado que puede ser entrada de otros agentes.

Cada agente se rige por interfaces estándares que permiten cambios en el ambiente externo al agente sin que éste sufra cambios en su funcionamiento interno. Todo el funcionamiento de los agentes autónomos, está coordinado por un elemento central, denominado Repositorio Activo, el cual entrega y recibe información de los agentes y coordina su funcionamiento. En el caso del sistema de vídeo vigilancia Suria, el **Gestor** cumple con la función de Repositorio activo al que se pueden conectar diferentes agentes autónomos.

4.3.2. Arquitectura de la solución.

Para el desarrollo de la arquitectura de la presente solución se tendrán en cuenta los siguientes patrones arquitectónicos:

- ✓ Patrón Modelo - Vista - Controlador

  El patrón Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

  - Modelo: El modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado y responde a instrucciones de cambiar el mismo.
  - Vista: Maneja la visualización de la información.
  - Controlador: Controla el flujo entre la vista y el modelo.

  Tanto la vista como el controlador dependen del modelo, que no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.

- ✓ Arquitectura basada en componentes:

Debido a que el sistema debe soportar una gran variedad de cámaras, es necesario un mecanismo que permita dar soporte a nuevos modelos sin tener que recompilar el código de la aplicación cada vez que esto sea necesario. Es por eso que la arquitectura basada en componentes es ideal para dar solución a este problema. Para cumplir con este patrón se ha diseñado un mecanismo de plugins, agregados externos a la aplicación, los cuales deben dedicarse al descubrimiento de un tipo en específico de cámara.

4.3.3. Patrones de diseño.

Para elaborar el diseño de la solución se ha tenido en cuenta la utilización de varios patrones de diseño. Un patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. (26)

Patrones GRASP.

Los patrones GRASP (patrones de los principios generales para asignar responsabilidades) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (26)

En la construcción del diagrama de clases para la solución se emplearon los siguientes patrones GRASP.

- ✓ Experto: Asigna una responsabilidad a la clase que cuenta con la información necesaria para cumplirla.
- ✓ Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos.
- ✓ Bajo Acoplamiento: Asigna una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases.
- ✓ Alta Cohesión: Asigna una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase.

Patrones 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.

Patrones de creación: Los patrones creacionales abstraen el proceso de creación de instancias. Ayudan a construir un sistema independiente de cómo sus objetos se crean, componen, y representan.

El patrón Singleton asegura que una clase tenga una sola instancia y provee un punto de acceso global a la misma. Este patrón asegura que se brinde desde cualquier clase un solo punto de acceso a las clases que se encargan de guardar y cargar los rangos de direcciones IP, los rangos de puertos y las credenciales que serán utilizadas para acceder a las cámaras, con el objetivo de evitar inconsistencias en los datos.

Patrones estructurales: Los patrones estructurales tienen que ver con cómo las clases y los objetos se componen para formar estructuras más grandes.

El patrón Fachada proporciona una interfaz unificada para un conjunto de interfaces en un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar. Qt provee un mecanismo de plugins, que pueden fomentar el crecimiento de funcionalidades, estos plugins no son más que objetos que implementan una determinada interfaz, lo que posibilita que la aplicación que los use pueda tener conocimiento de su estructura.

Patrones de Comportamiento: Los patrones de comportamiento tienen que ver con los algoritmos y la asignación de responsabilidades entre los objetos.

El patrón Observador garantiza que cuando un objeto cambia, todos los objetos que dependen del mismo sean notificados y actualizados. Este patrón se evidencia en el uso del mecanismo de señales y slots de Qt. También se puede ver en la comunicación entre los plugins para descubrir las cámaras y la aplicación, pues esta última se convierte en observadora de los plugins y es notificada en el momento en que uno de ellos culmina el análisis.

4.3.4. Diagramas de secuencia.

Los diagramas de interacción describen cómo un grupo de objetos colaboran para lograr algún fin. Se presentan en diferentes formas, basadas todas ellas en una misma información subyacente pero resaltando cada una un punto de vista de la misma.

Los diagramas de secuencia muestran los objetos que participan en una interacción mediante sus líneas de vida y los mensajes que intercambian, organizados en forma de una secuencia temporal.

A continuación se muestran los diagramas de secuencia del diseño para los casos de uso Descubrir Cámara, Modificar Cámara y para el escenario Adicionar Rango de IP del caso de uso Configurar Rangos

de IP. Debido a que el resto de los diagramas de secuencia son muy similares a los presentados en el capítulo, pueden ser consultados en los anexos.

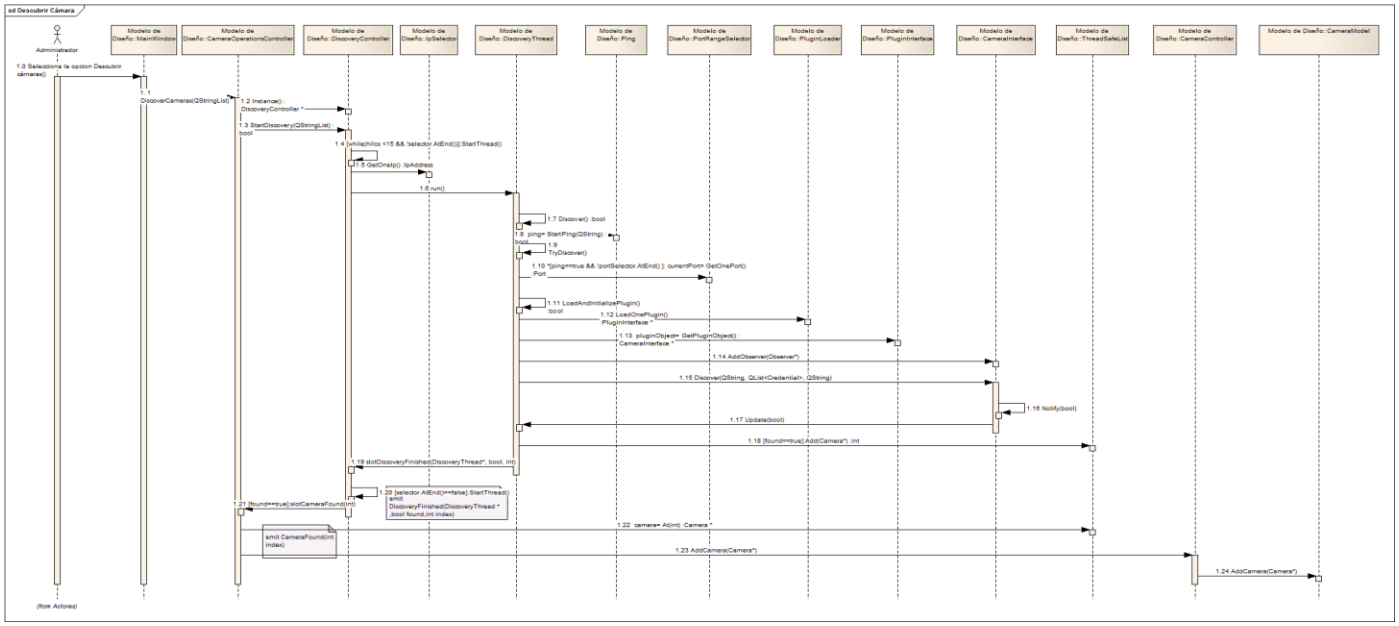


Figura 6. Diagrama de secuencia del caso de uso Descubrir Cámara

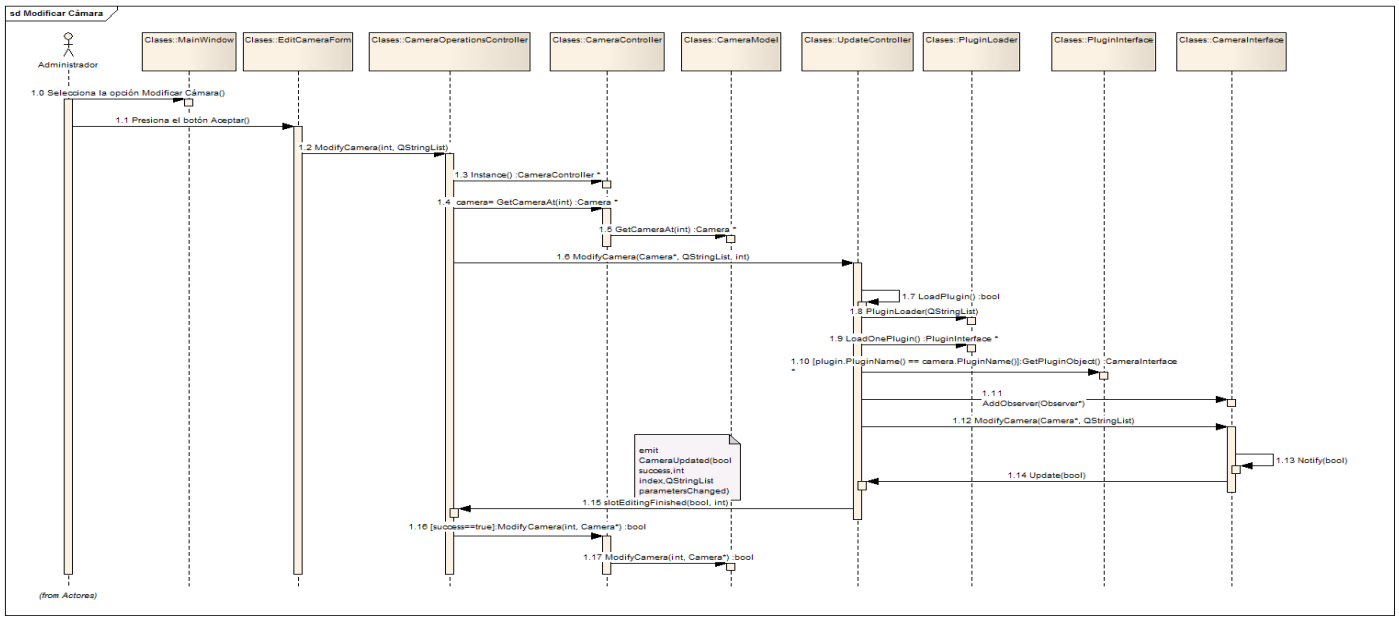


Figura 7. Diagrama de Secuencia del caso de uso Modificar Cámara

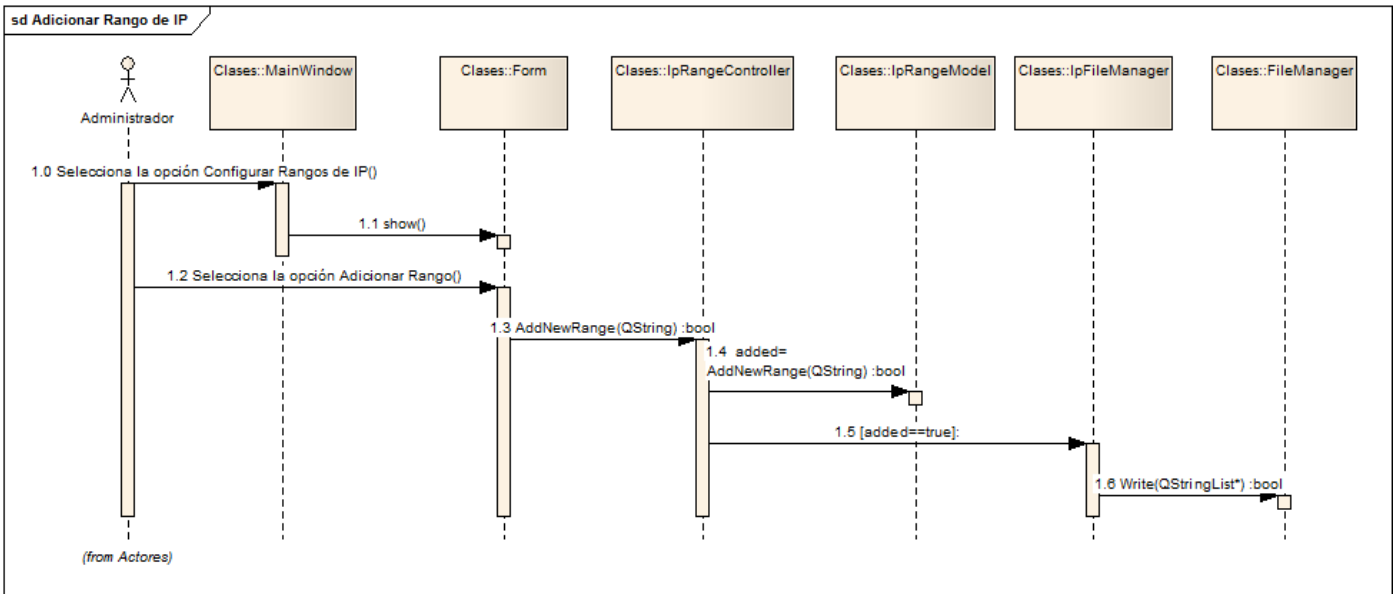


Figura 8. Diagrama de secuencia del escenario Adicionar Rango de IP del caso de uso Configurar Rangos de IP

4.3.5. Diagrama de clases del diseño.

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Seguidamente se presenta el diagrama de clases del diseño, particularmente la interacción entre los paquetes. El diagrama de clases para cada uno de los paquetes, con los atributos y operaciones detallados, puede ser consultado en los Anexos.

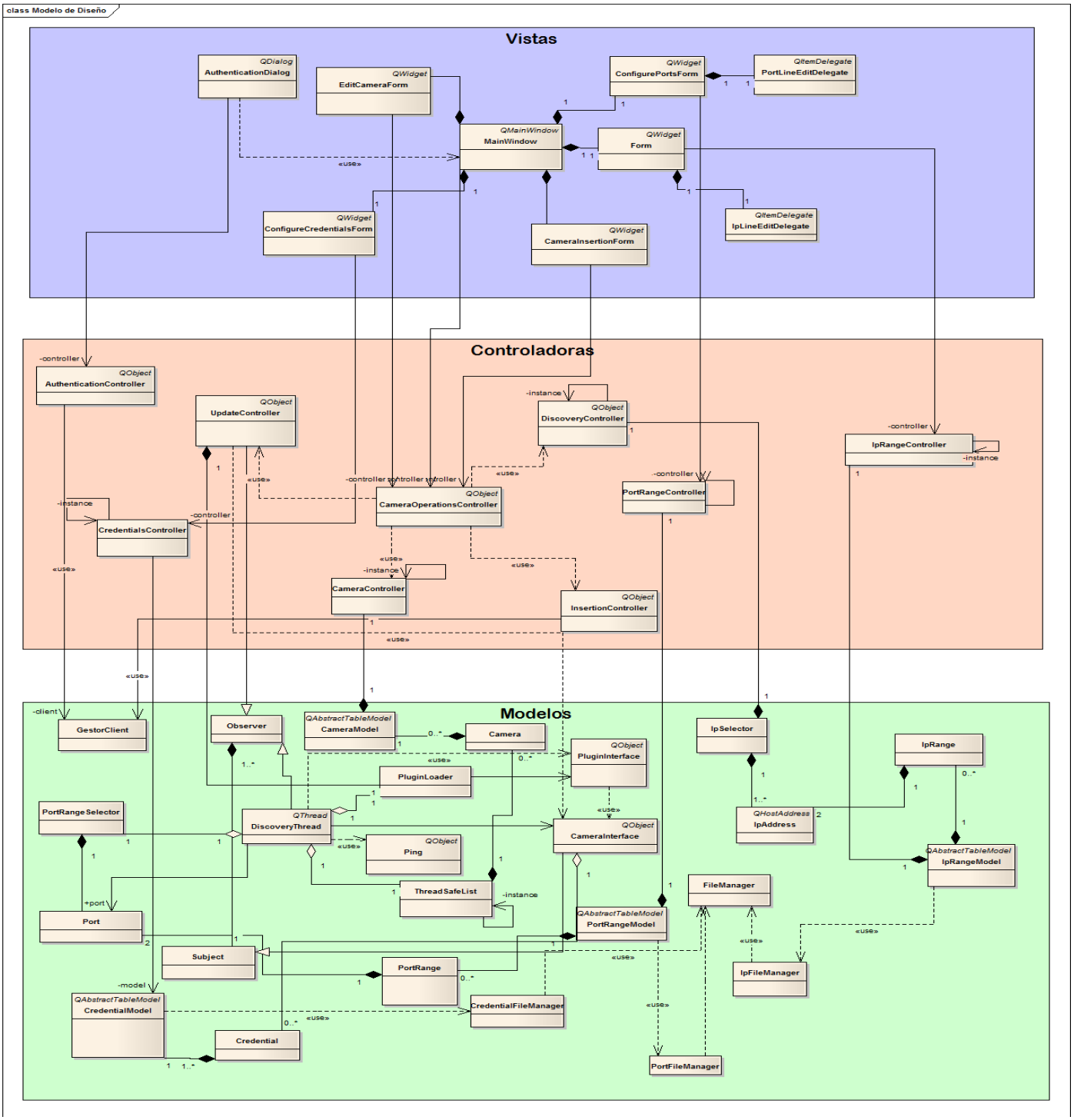


Figura 9. Diagrama de clases del diseño. Interacción entre paquetes.

A continuación se muestra un diagrama de clases que representa la estructura que tendrán los plugins que han de crearse para que la aplicación sea capaz de brindar soporte a los diferentes modelos.

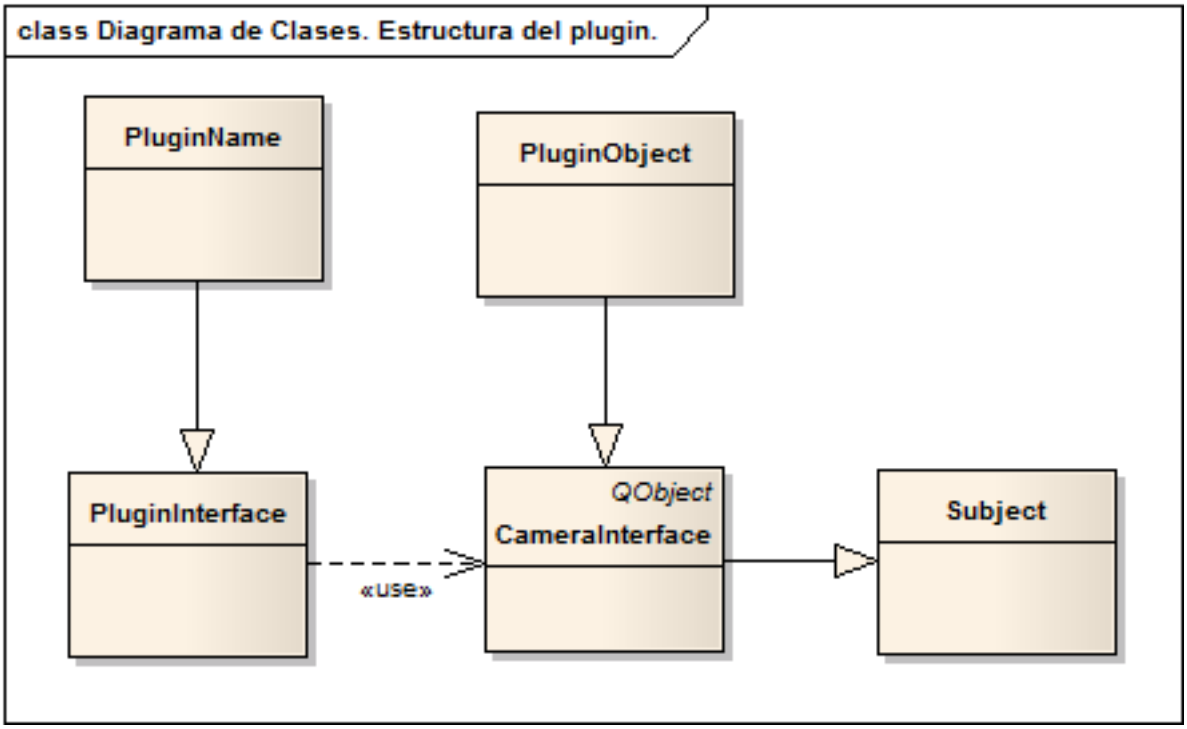


Figura 10. Estructura de los plugins.

4.4. Conclusiones.

Una vez concluido el capítulo se puede afirmar que con la realización del diagrama de clases del análisis se logró un mayor entendimiento de los requisitos funcionales del sistema. Luego de haber definido la arquitectura de la solución y mediante la utilización de los patrones de diseño escogidos se garantiza la reutilización y el mantenimiento del código y con el diagrama de clases del diseño se ha logrado una visión detallada de las clases necesarias para completar la implementación del sistema.

CAPITULO 5. **IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN.**

5.1. Introducción.

En este capítulo se tendrán en cuenta los aspectos del diseño para llevar a cabo la implementación del sistema, se muestra el diagrama de despliegue y el de componentes. Además, se realizarán las pruebas que validarán la correcta implementación del sistema.

5.2. Implementación de la solución.

El propósito fundamental de la implementación es desarrollar la arquitectura y el sistema como un todo. (20) Los objetivos de esta disciplina son: definir la organización del código en términos de los subsistemas de implementación organizados en capas, implementar los elementos de diseño en términos de los elementos de implementación, así como probar y desarrollar componentes como unidades.

5.2.1. Modelo de Despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad en los nodos de cómputo. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento, y las conexiones entre ellos en el sistema. El modelo de despliegue consta de uno o más nodos (elementos de procesamiento con, al menos, un procesador, memoria, y posiblemente otros dispositivos), dispositivos (nodos estereotipados sin capacidad de procesamiento), y conectores entre los nodos, y entre nodos y dispositivos. (27)

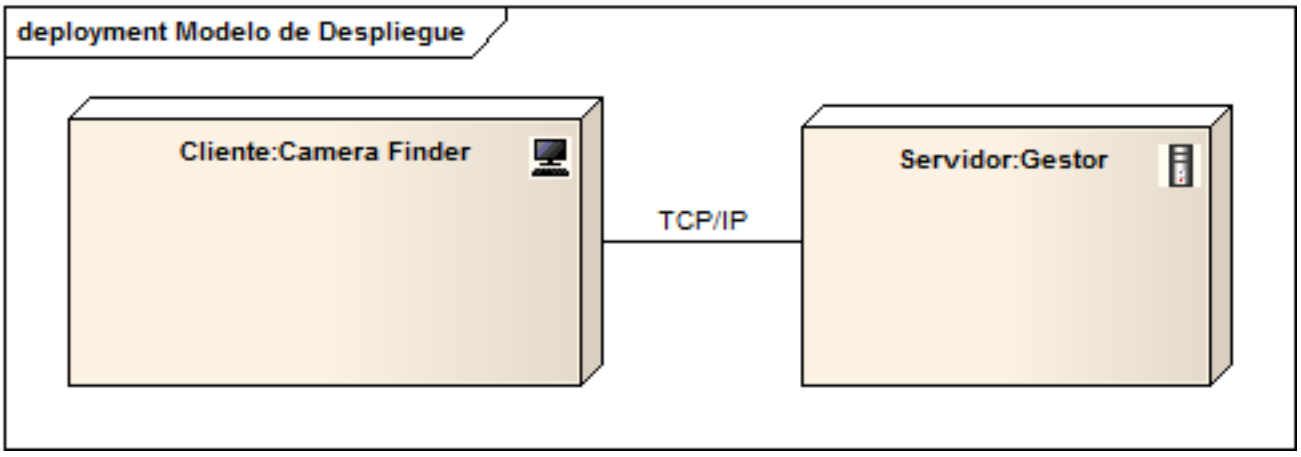


Figura 11. Modelo de Despliegue

Descripción de los nodos:

Cliente: Camera Finder. Es el nodo donde correrá la aplicación para descubrir las cámaras.

Servidor: Gestor. En este nodo reside el Gestor del Sistema de Vídeo Vigilancia Suria. Con este se comunica el nodo Cliente mediante el protocolo TCP/IP con el propósito de insertar las cámaras y autenticar los usuarios.

5.2.2. Diagrama de componentes.

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes. Además, describe cómo se organizan los componentes de acuerdo a los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizados, y cómo dependen los componentes unos de otros.

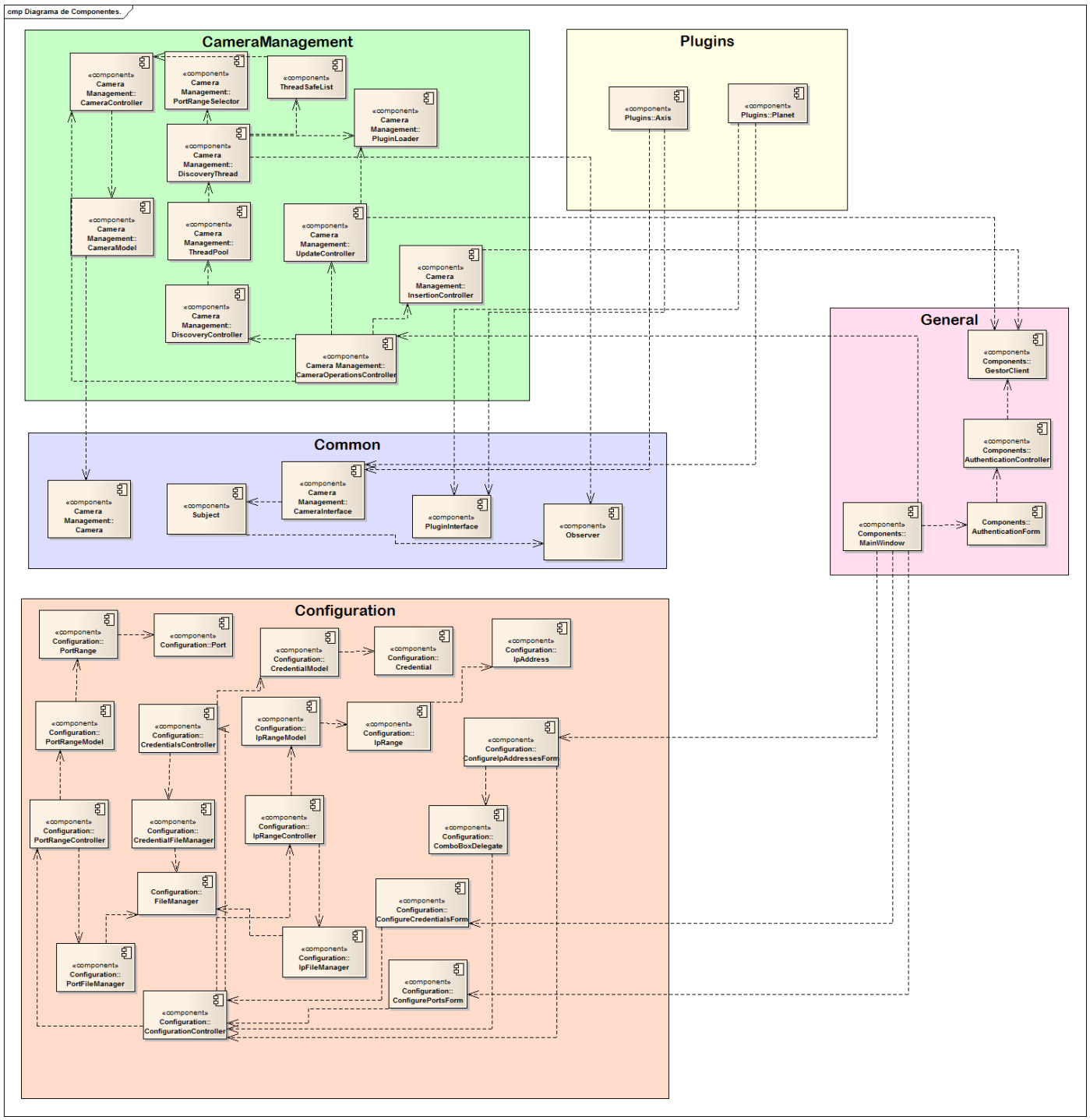


Figura 12. Diagrama de componentes.

5.3. Validación de la solución.

El desarrollo del software implica una serie de actividades de producción en las que existe la posibilidad de que aparezcan errores humanos. Estos errores pueden comenzar a darse desde el primer momento del proceso en el que los requerimientos pueden estar especificados incorrectamente o pueden surgir en posteriores momentos del diseño y desarrollo. Por esta causa el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad del mismo.

Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

5.3.1. Método de prueba utilizado.

Para la realización de las pruebas se tendrá en cuenta el método de caja negra, el cual se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del sistema sin tener mucho en cuenta la estructura interna del software.

5.3.2. Casos de prueba.

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados, desarrollados para cumplir un objetivo en particular o una función esperada. A continuación se muestra el caso de prueba para el caso de uso Gestionar Credenciales, el resto puede ser consultado en los Anexos, al igual que los resultados de las mismas.

Nombre de la sección	**Descripción de la funcionalidad**	**Escenarios de la sección**	**Flujo Central**

Adicionar Credencial	El objetivo de esta funcionalidad es adicionar una credencial al diccionario de usuarios y contraseñas que se utilizará para descubrir las cámaras.	EC 1.1: "Adición exitosa".	El usuario introduce un usuario y una contraseña y presiona el botón Adicionar. El sistema adiciona la credencial a la lista de credenciales.
		EC 1.2: "Faltan datos"	El usuario deja el campo usuario vacío y presiona el botón Adicionar. El sistema muestra un mensaje informando que debe introducir al menos el nombre de usuario para realizar esta operación.
		EC 1.3: "La credencial ya existe"	El usuario introduce una credencial que ya se encuentra en la lista y selecciona la opción Adicionar. El sistema muestra un mensaje informando que la credencial existe.
Eliminar credencial	El propósito de esta funcionalidad es eliminar una credencial de la lista de credenciales.	EC 2.1: "Eliminación exitosa"	El usuario selecciona una credencial y presiona el botón Eliminar. El sistema elimina la credencial de la lista.
		EC 2.2: "No hay una credencial seleccionada"	El usuario presiona el botón Eliminar sin seleccionar una credencial. El sistema no realiza la operación.

Tabla 7. Caso de prueba "Gestionar credenciales".

5.3.3. Resultados.

Fueron probados siete casos de uso, los que representan el cien por ciento del total de funcionalidad. Para cada caso de uso se han tenido en cuenta varios escenarios que implican cada una de las posibles interacciones del usuario o combinaciones de situaciones posibles con el fin de evaluar el comportamiento del sistema ante cada funcionalidad. El resultado de las pruebas coincide con la especificación de cada uno de los casos de uso, lo que demuestra la correcta implementación de los mismos.

5.4. Conclusiones.

En este capítulo se partió del diseño realizado en el capítulo anterior para llevar a cabo la implementación del sistema. Con la realización del diagrama de despliegue se obtuvo una idea de cuál sería la distribución física del sistema. Con el diagrama de componentes, se determinó cómo serían implementadas las clases del diseño en términos de componentes software, de acuerdo con los mecanismos de estructuración en el entorno de implementación y en el lenguaje de programación utilizados. Todo esto posibilitó la implementación del sistema, el cual fue probado posteriormente demostrándose que posee las validaciones necesarias y devuelve los resultados esperados de cada funcionalidad.

CONCLUSIONES GENERALES

Con el desarrollo del presente trabajo de diploma se dio cumplimiento a las tareas y objetivos trazados, lo que permitió arribar a las siguientes conclusiones:

- ✓ Fueron estudiadas las principales aplicaciones que realizan descubrimiento de cámaras, llegando a la conclusión de que ninguna satisfacía las necesidades del Sistema de Vídeo Vigilancia Suria.
- ✓ Se definieron las herramientas y tecnologías, las cuales contribuyeron a un correcto proceso de modelación e implementación.
- ✓ Fue diseñada una arquitectura que logró la implementación de una aplicación escalable y a la medida, con todas las funcionalidades necesarias, y que además brinda la posibilidad de añadir soporte para el descubrimiento de nuevos modelos de cámaras.
- ✓ Todas las funcionalidades de la aplicación están implementadas correctamente, lo cual queda respaldado por los resultados obtenidos en las pruebas.

Los objetivos alcanzados a través del desarrollo del presente trabajo, así como la utilización de la aplicación lograda serán de gran utilidad para los usuarios del Sistema de Vídeo Vigilancia Suria.

RECOMENDACIONES

Al concluir el desarrollo de este trabajo de diploma se recomienda:

- ✓ Continuar con la implementación de plugins con el fin de que el sistema sea capaz de soportar más modelos de cámara.
- ✓ Crear una nueva versión que utilice protocolos como Universal Plug And Play y Bonjour, que agilizan el proceso de descubrimiento de dispositivos.

REFERENCIAS

1. **MSc Pérez Martinto, Pedro Carlos.** *El diseño metodológico de la investigación científica.Teoría de Muestreo: población y muestra. Diseño experimental y métodos.* Ciudad de la Habana, Cuba : Universidad de las Ciencias Informáticas, 2010.

2. **Tanenbaum, Andrew S.** *Computer Networks.* Nueva Jersey : Prentice Hall PTR , 2003. ISBN 0-13-066102-3 .

3. **Feibel, Werner.** *The Encyclopedia of Networking, Second Edition.* Alameda,CA : Sybex, 1996. ISBN: 0-7821-1829-1.

4. **Information Technology Systems, UNCW.** *NETWORKING AT UNCW.* s.l. : Information Technology Systems, 2010.

5. **Schwarts, Susana.** *What is on your network?* s.l. : ARKIPELAGO, 2009.

6. **EMC Corporation.** *EMC Smarts IP Availability Manager.* s.l. : EMC Corporation, 2005.

7. **Whyte, David.** *Network scanning detection etrategies for enterprise networks.* Ottawa,Canada : School of Computer Science at Carleton University, 2008.

8. **De Montigny-Leboeuf, Annie y Massicotte, Frédéric.** *Passive Network Discovery for Real Time Situation Awareness.* Ottawa, Canadá : Communication Research Centre Canada, 2004.

9. **Arkin, Ofir.** *Deficiencies of Active Network Discovery Systems.* s.l. : Insightix Ltd., 2005.

10. **Treurniet, J.** *An Overview of Passive Information Gathering Techniques for Network Security.* Ottawa,Canada : Defence R&D Canada – Ottawa, 2004.

11. **Avallone, S., y otros.** *A Topology Discovery Module based on a Hybrid Methodology.* s.l. : University of Napoli “Federico II”.

12. **Beasley, Jeffrey S.** *Networking, Second Edition.* s.l. : Prentice Hall, 2009. ISBN-13: 978-0-13-135838-6.

13. **Torres Faría, Daniela A.** *Protocolos de Internet(ARP,RARP,TCP/IP).* s.l. : Universidad Simón Bolívar, 2009.

14. **Axis Communications.** Axis Camera Station. *Axis Communications.* [En línea] [Citado el: 4 de Diciembre de 2010.] http://www.axis.com/files/manuals/um_acs_40909_en_1011.pdf.

15. **Inaxsys ICT Securitity Systems Inc.** About Inaxsys IVT. *Inaxsys Vídeo technologies.* [En línea] Inaxsys ICT Securitity Systems Inc, 2009. [Citado el: 02 de Diciembre de 2010.] http://www.inaxsys.com/en/about/inaxsys-ivt.html.

16. **Lumenera Corporation.** About Lumenera. *Lumenera Corporation.* [En línea] Lumenera Corporation, 2010. [Citado el: 11 de Noviembre de 2010.] http://www.lumenera.com/corporate/about-lumenera.php.

17. **Milestone Systems.** Company Profile. *Milestones Company.* [En línea] 2010. [Citado el: 15 de Noviembre de 2010.] http://www.milestonesys.com/company/company_overview/company_profile.

18. **Velthuis, Piattini.** *Análisis y Diseño detallado de aplicaciones Informáticas de Gestión.* s.l. : Librería y Editorial Microinformática, 1996. ISBN: 8478972331.

19. **Kruchten, Philippe.** *The Rational Unified Process An Introduction.* s.l. : Addison Wesley, 2001.

20. **Jacobson, I., Booch, G. y Rumbaugh, J.** *"El Proceso Unificado de Desarrollo de software".* s.l. : Adisson-Wesley, 2000.

21. **Rumabugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado.Manual de Referencia.* s.l. : Adisson-Wesley, 1998.

22. **Lazalde Miranda, Cristina y Miranda Valles, Mayra Yanin.** EcuRed. [En línea] [Citado el: 22 de Febrero de 2010.] http://www.ecured.cu/index.php/Herramienta_CASE.

23. **Pressman, Roger S.** *Ingeniería del Software.Un enfoque práctico.* s.l. : Mc Graw Hill, 2005. ISBN: 9701054733.

24. Glosario.NET. [En línea] HispaNetwork Publicidad y Servicios, S.L., 16 de Marzo de 2007. [Citado el: 23 de Febrero de 2011.] http://tecnologia.glosario.net/terminos-viricos/lenguaje-de-programaci%F3n-9768.html.

25. 9 of the Best Free Linux Integrated Development Environments (IDEs). [En línea] 18 de Enero de 2010. [Citado el: 23 de Febrero de 2011.] http://www.linuxlinks.com/article/20090620114618990/IDE.html.

26. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* s.l. : Prentice Hall, 1999. ISBN 0-13-748880-7.

27. **Rational Software Corporation.** Artifact: Deployment Model. *Rational Software Corporation.* [En línea] Junio de 2003. [Citado el: 6 de Abril de 2011.] http://rup.hops-fp6.org/process/artifact/ar_dplmdl.htm.

28. **MacMahon, Richard A.** *Introducción a las Redes.*

BIBLIOGRAFÍA

- ✓ **MSc Pérez Martinto, Pedro Carlos.** *El diseño metodológico de la investigación científica.Teoría de Muestreo: población y muestra. Diseño experimental y métodos.* Ciudad de la Habana, Cuba : Universidad de las Ciencias Informáticas, 2010.
- ✓ **Tanenbaum, Andrew S.** *Computer Networks.* Nueva Jersey : Prentice Hall PTR , 2003. ISBN 0-13-066102-3 .
- ✓ **Feibel, Werner.** *The Encyclopedia of Networking, Second Edition.* Alameda,CA : Sybex, 1996. ISBN: 0-7821-1829-1.
- ✓ **Information Technology Systems, UNCW.** *NETWORKING AT UNCW.* s.l. : Information Technology Systems, 2010.
- ✓ **Schwarts, Susana.** *What is on your network?* s.l. : ARKIPELAGO, 2009.
- ✓ **EMC Corporation.** *EMC Smarts IP Availability Manager.* s.l. : EMC Corporation, 2005.
- ✓ **Whyte, David.** *Network scanning detection etrategies for enterprise networks.* Ottawa,Canada : School of Computer Science at Carleton University, 2008.
- ✓ **De Montigny-Leboeuf, Annie y Massicotte, Frédéric.** *Passive Network Discovery for Real Time Situation Awareness.* Ottawa, Canadá : Communication Research Centre Canada, 2004.
- ✓ **Arkin, Ofir.** *Deficiencies of Active Network Discovery Systems.* s.l. : Insightix Ltd., 2005.
- ✓ **Treurniet, J.** *An Overview of Passive Information Gathering Techniques for Network Security.* Ottawa,Canada : Defence R&D Canada – Ottawa, 2004.
- ✓ **Avallone, S., y otros.** *A Topology Discovery Module based on a Hybrid Methodology.* s.l. : University of Napoli “Federico II”.
- ✓ **Beasley, Jeffrey S.** *Networking, Second Edition.* s.l. : Prentice Hall, 2009. ISBN-13: 978-0-13-135838-6.
- ✓ 13. **Torres Faría, Daniela A.** *Protocolos de Internet(ARP,RARP,TCP/IP).* s.l. : Universidad Simón Bolívar, 2009.

- ✓ **Axis Communications.** Axis Camera Station. *Axis Communications.* [En línea] [Citado el: 4 de Diciembre de 2010.] http://www.axis.com/files/manuals/um_acs_40909_en_1011.pdf.
- ✓ **Inaxsys ICT Securitity Systems Inc.** About Inaxsys IVT. *Inaxsys Vídeo technologies.* [En línea] Inaxsys ICT Securitity Systems Inc, 2009. [Citado el: 02 de Diciembre de 2010.] http://www.inaxsys.com/en/about/inaxsys-ivt.html.
- ✓ **Lumenera Corporation.** About Lumenera. *Lumenera Corporation.* [En línea] Lumenera Corporation, 2010. [Citado el: 11 de Noviembre de 2010.] http://www.lumenera.com/corporate/about-lumenera.php.
- ✓ **Milestone Systems.** Company Profile. *Milestones Company.* [En línea] 2010. [Citado el: 15 de Noviembre de 2010.] http://www.milestonesys.com/company/company_overview/company_profile.
- ✓ **Velthuis, Piattini.** *Análisis y Diseño detallado de aplicaciones Informáticas de Gestión.* s.l. : Librería y Editorial Microinformática, 1996. ISBN: 8478972331.
- ✓ **Kruchten, Philippe.** *The Rational Unified Process An Introduction.* s.l. : Addison Wesley, 2001.
- ✓ **Jacobson, I., Booch, G. y Rumbaugh, J.** *"El Proceso Unificado de Desarrollo de software".* s.l. : Adisson-Wesley, 2000.
- ✓ **Rumabugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado.Manual de Referencia.* s.l. : Adisson-Wesley, 1998.
- ✓ **Lazalde Miranda, Cristina y Miranda Valles, Mayra Yanin.** EcuRed. [En línea] [Citado el: 22 de Febrero de 2010.] http://www.ecured.cu/index.php/Herramienta_CASE.
- ✓ **Pressman, Roger S.** *Ingeniería del Software.Un enfoque práctico.* s.l. : Mc Graw Hill, 2005. ISBN: 9701054733.
- ✓ Glosario.NET. [En línea] HispaNetwork Publicidad y Servicios, S.L., 16 de Marzo de 2007. [Citado el: 23 de Febrero de 2011.] http://tecnologia.glosario.net/terminos-viricos/lenguaje-de-programaci%F3n-9768.html.
- ✓ 9 of the Best Free Linux Integrated Development Environments (IDEs). [En línea] 18 de Enero de 2010. [Citado el: 23 de Febrero de 2011.] http://www.linuxlinks.com/article/20090620114618990/IDE.html.

- ✓ **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* s.l. : Prentice Hall, 1999. ISBN 0-13-748880-7.
- ✓ **Rational Software Corporation.** Artifact: Deployment Model. *Rational Software Corporation.* [En línea] Junio de 2003. [Citado el: 6 de Abril de 2011.] http://rup.hops-fp6.org/process/artifact/ar_dplmdl.htm.
- ✓ **MacMahon, Richard A.** *Introducción a las Redes.*
- ✓ **Groth, David y Skandier, Toby.** *Network+ Study Guide.* s.l. : Sybex, 2005. ISBN-10: 0782144063 .
- ✓ **Clark, Martin P.** *Data networks, IP, and the Internet : networks, protocols, design, and operation.* West Sussex : John Wiley & Sons , 2003. ISBN 0-470-84856-1.
- ✓ **Poratti , Gustavo Gabriel.** *Redes: La guía de referencia actual y definitiva.* s.l. : MP Ediciones S.A.
- ✓ **Alcántara Pérez, Javier de Jesús y Vilchis Serrano, Arturo César.** *Tesina para obtenerel Título de Licenciado en Ciencias Informáticas: Desarrollo de tienda virtual consumidor a consumidor para la empresa Tasi Software S.A de C.V.* Mexico D.F : Instituto Politécnico Nacional, 2010.
- ✓ **Axis Communications.** Axis Communications. *¿Qué es una cámara de red?* [En línea] 2010. [Citado el: 10 de Noviembre de 2010.] http://www.axis.com/es/products/vídeo/camera/about_cameras/overview.htm.
- ✓ **Network Instruments.** *SNMP Monitoring: One Critical Component to Network Management.* s.l. : Network Instruments, 2005.
- ✓ **Orebaugh, Angela y Pinkard, Becky.** *Nmap in the Enterprise: Your Guide to Network Scanning.* s.l. : Syngress Publishing, Inc., 2008. ISBN 13: 978-1-59749-241-6.
- ✓ **Lumenera Corporation.** IP Discovery and Connectivity App Note . *Lumenera Corporation.* [En línea] Marzo de 2005. [Citado el: 10 de Noviembre de 2010.] http://www.lumenera.com/support/pdf/LA-2113-IPDiscoveryAndConnectivity.pdf.
- ✓ **GVI Security.** *AutoIP Sales Flier.* s.l. : GVI Security, 2010.
- ✓ El modelo OSI y los protocolos de red. [En línea] [Citado el: 2 de 12 de 2010.] blyx.com/public/docs/pila_OSI.pdf .

- ✓ **Creator, Qt.** Qt Creator. [En línea] http://developer.qt.nokia.com/wiki/Category:Tools::QtCreator_Spanish.
- ✓ **Gómez, Reynier Pupo.** *Gestor de ficheros para el entorno de escritorio Quantico.* Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2010.
- ✓ **Soulie, Juan.** cplusplus.com. [En línea] 29 de Septiembre de 2009. [Citado el: 23 de Febrero de 2011.] http://www.cplusplus.com/info/description/.
- ✓ **Colectivo de Autores.** Departamento de Informática y Sistemas.Universidad de Murcia. [En línea] [Citado el: 23 de Febrero de 2011.] http://dis.um.es/~ginesgm/files/doc/pav/guion1.pdf.
- ✓ —. *Conferencia 1.Introducción a la Ingeniería de Software.* s.l. : Universidad de las Ciencias Informáticas, 2007.
- ✓ **Hernández Orallo, E.** *El Lenguaje Unificado de Modelado (UML).*
- ✓ **Corporation, Nokia.** Qt Developer Network. *Qt Creator Whitepaper.* [En línea] 2010. [Citado el: 02 de 23 de 2010.] http://developer.qt.nokia.com/wiki/QtCreatorWhitepaper.
- ✓ **IEEE Computer Society .** *1471-2000. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.* 2000. E-ISBN 0-7381-2519-9.
- ✓ **Gamma, Erich, y otros.** *Design Patterns. Elements of Reuseable Object-Oriented Software.*
- ✓ **Buschmann, Frank, y otros.** *Pattern-Oriented Software Architecture. A system of patterns.* s.l. : JOHN WILEY & SONS, 1996.
- ✓ **Scott, Col l.** *A Deeper Look at Signals and Slots.* 2005.

ANEXOS

Anexo 4. Diagramas de clases del diseño.

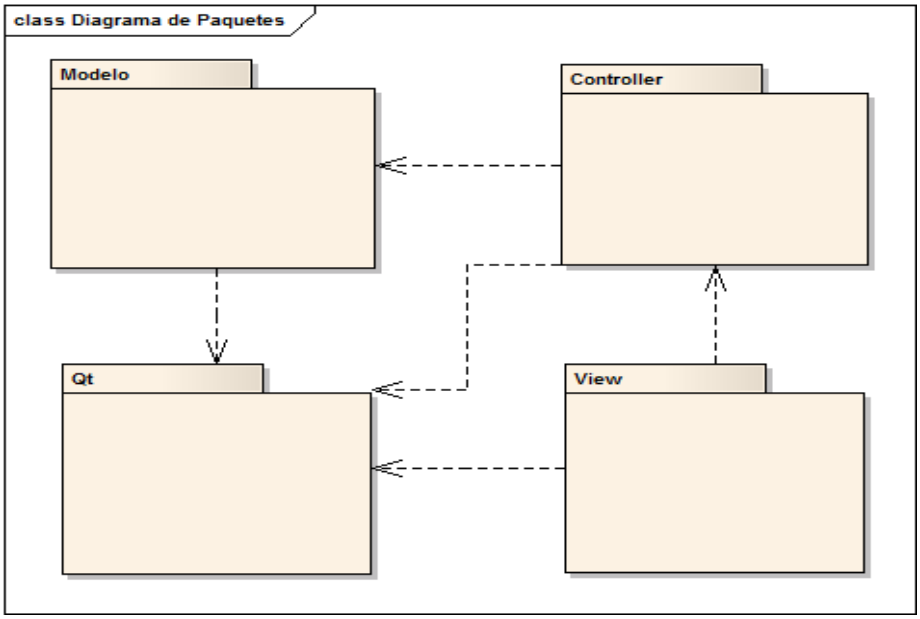


Figura 13. Vista de paquetes del diagrama de clases del diseño.

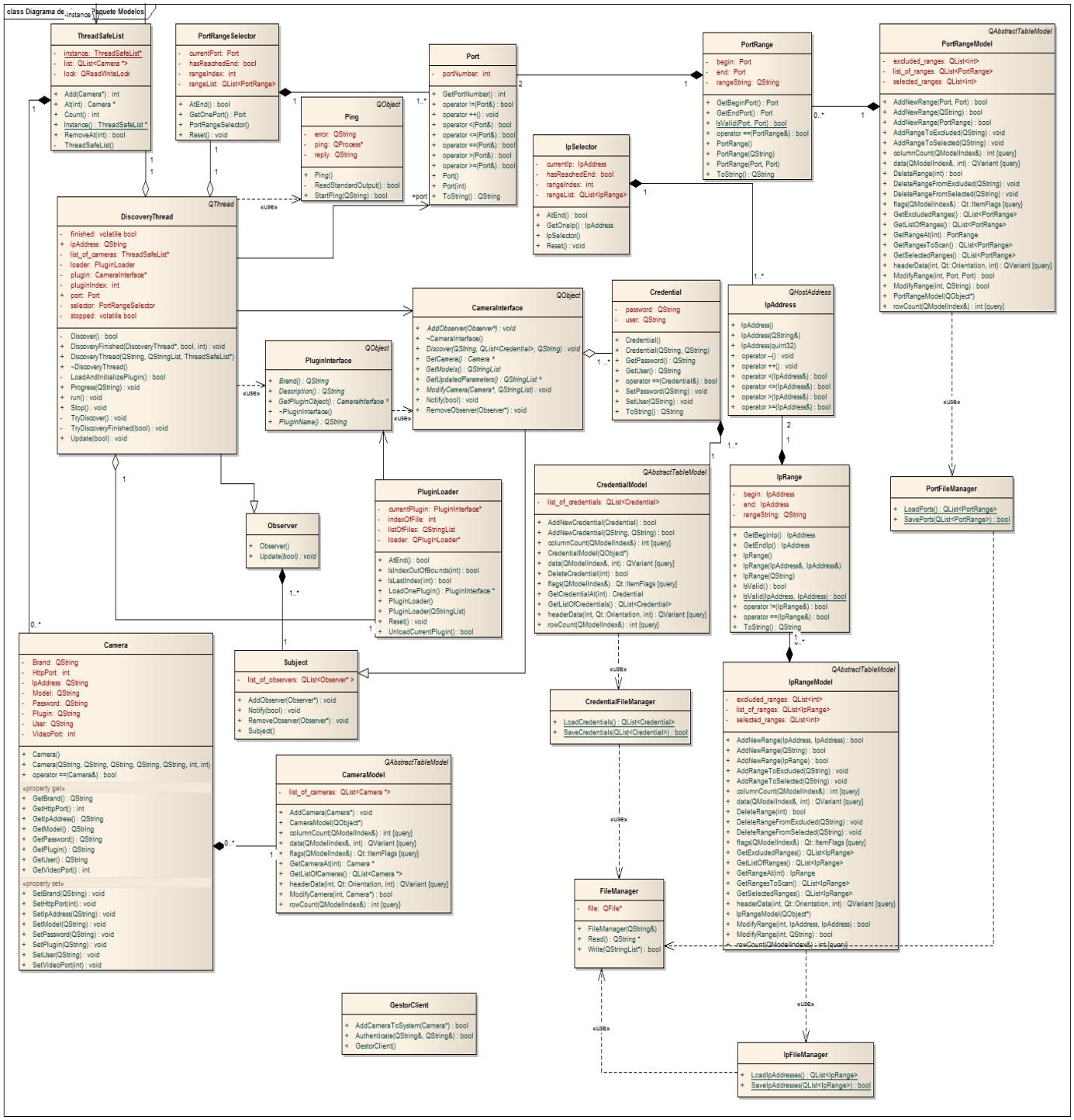


Figura 14. Diagrama de clases. Paquete Modelos.

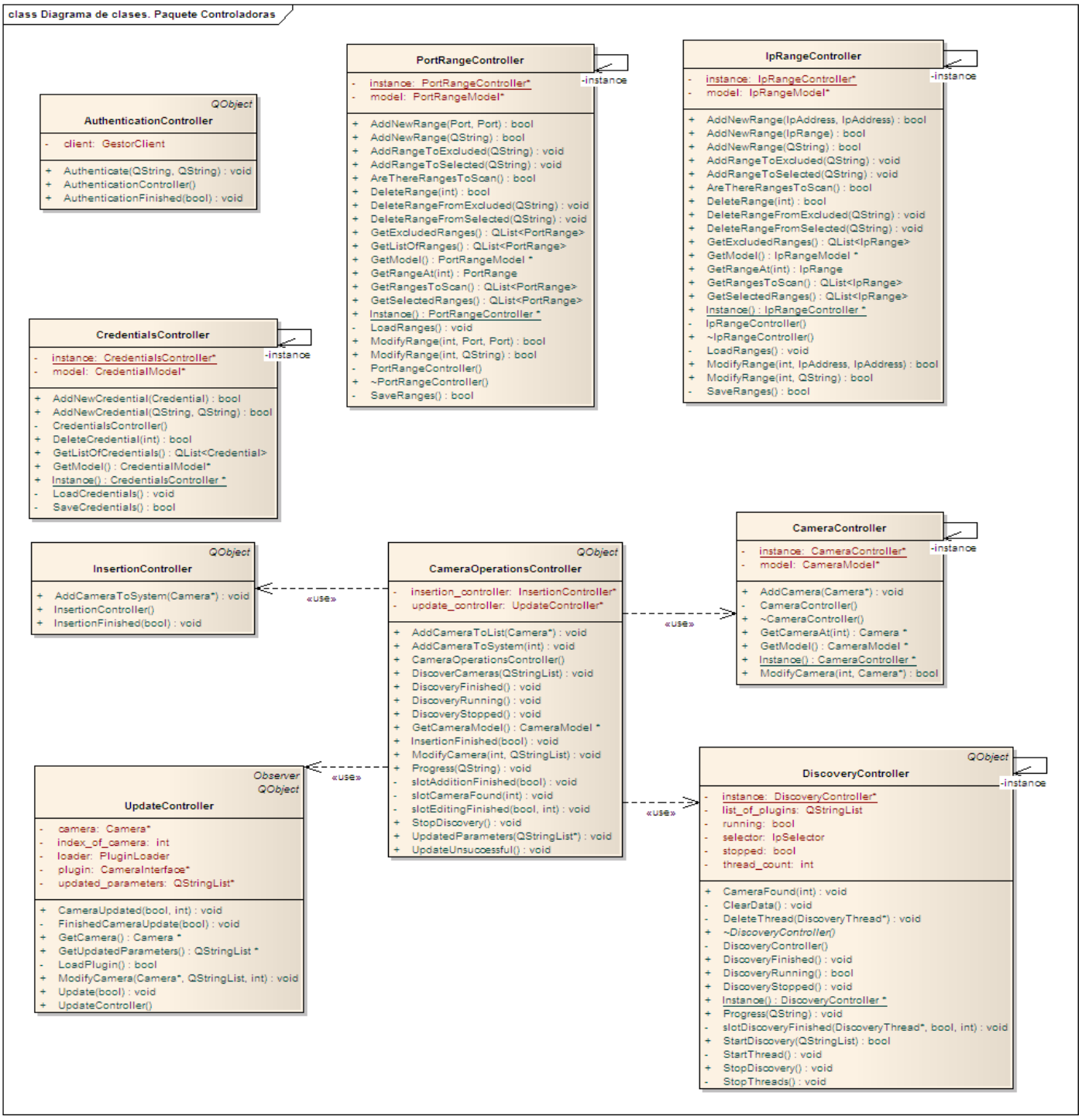


Figura 15. Diagrama de clases. Paquete Controladoras.

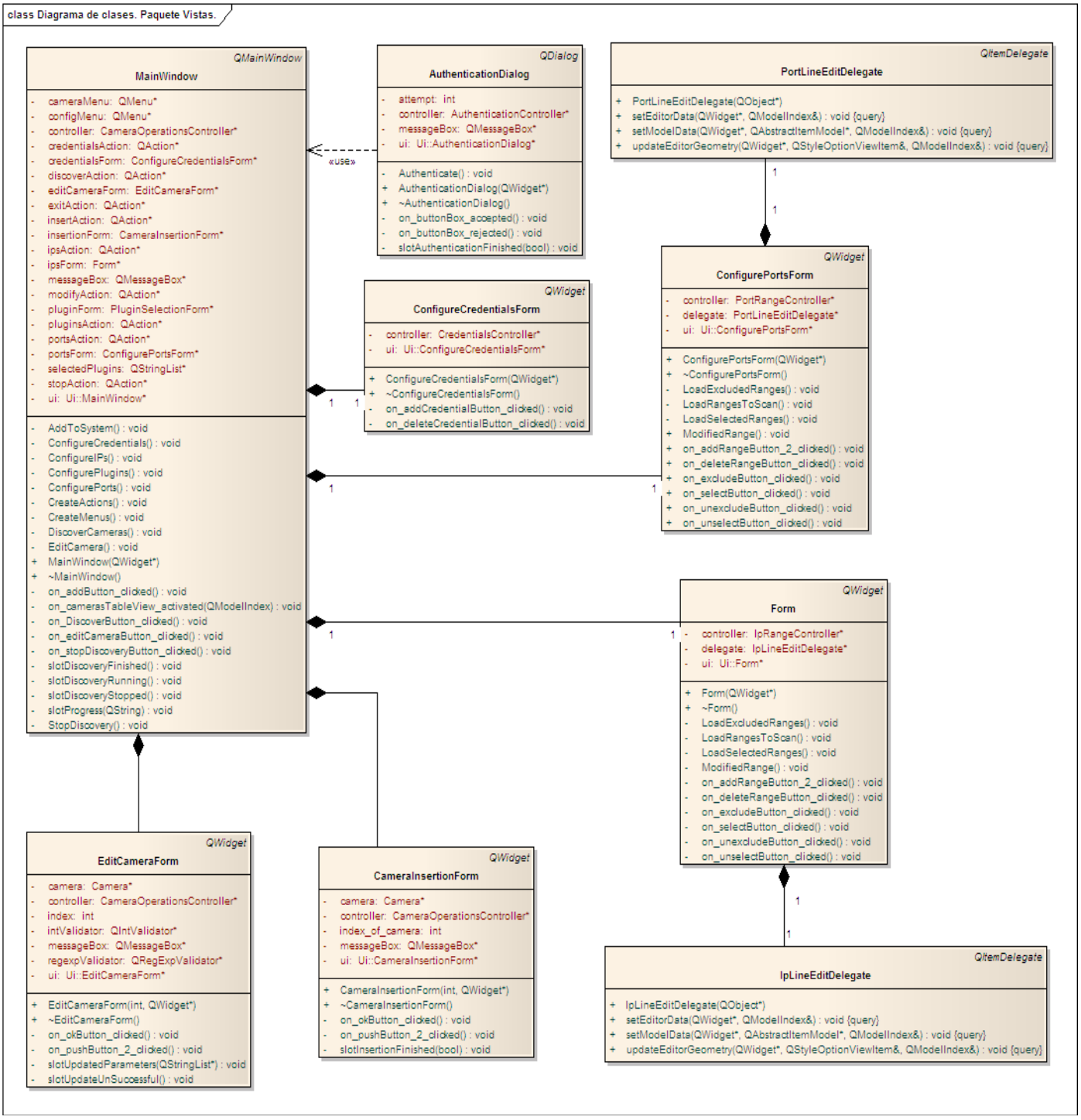


Figura 16. Diagrama de clases. Paquete Vistas.

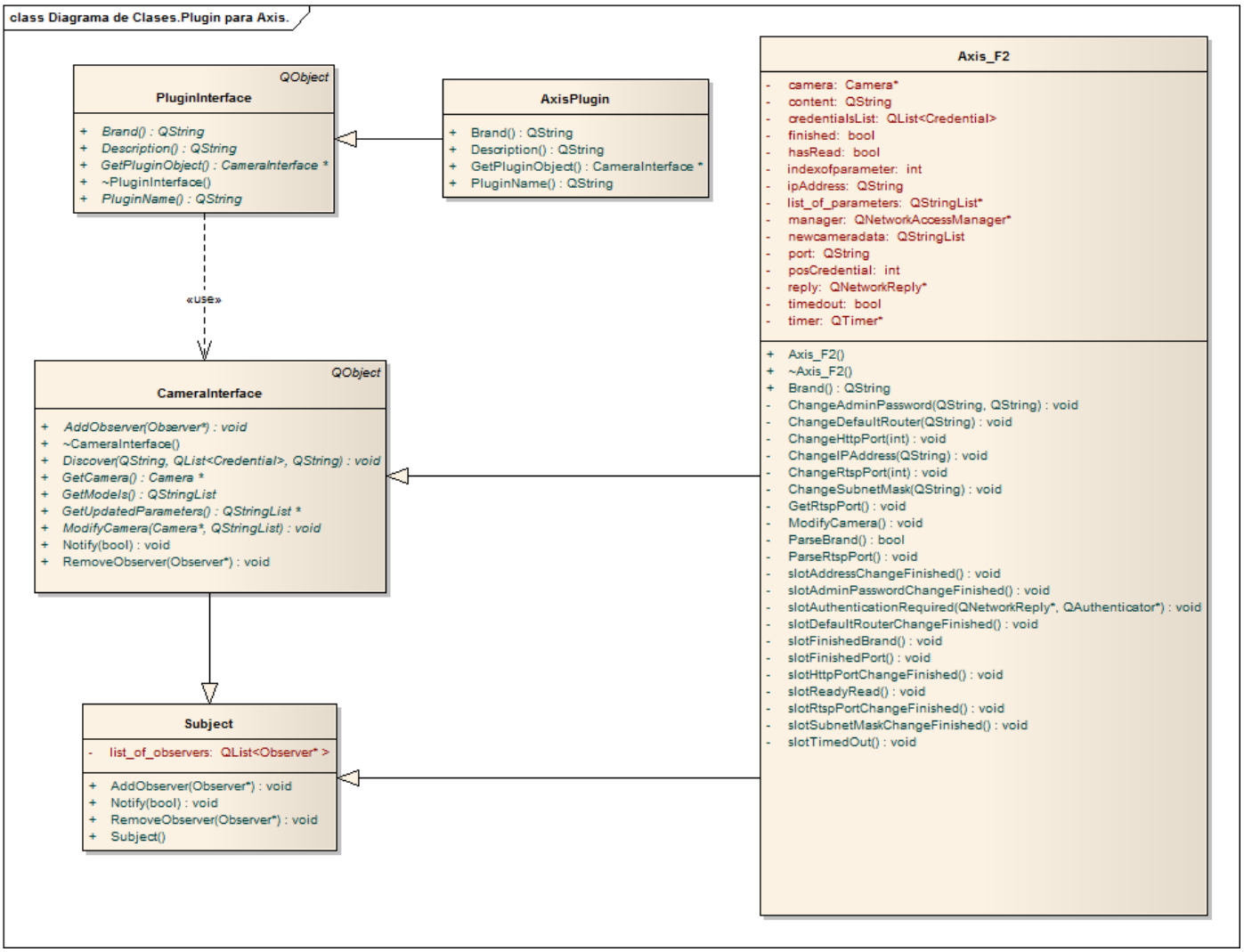


Figura 17. Diagrama de Clases. Plugin Axis.

GLOSARIO DE TÉRMINOS.

Bonjour: Bonjour es un método para descubrir servicios en una red de área local. Esta tecnología permite a los usuarios establecer una red sin ningún tipo de configuración.

CASE: Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste del mismo en términos de tiempo y de dinero. (22)

SNMP: El Protocolo Simple de Administración de Red es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento.

TCP/IP: Es un paquete integrado de protocolos (más de un protocolo trabajando en conjunto para conseguir una tarea concreta) que se ha convertido en el lenguaje de Internet y cuyo uso es un prerrequisito para acceder a ella. El protocolo TCP/IP combinado es el protocolo más usado en el mundo porque actúa como protocolo de las capas de transporte y red usado globalmente en Internet (28)

Universal Plug And Play: Es un conjunto de protocolos de red promulgados por el Foro UPnP, cuyo objetivo es permitir a los dispositivos de red que se conecten entre sí y simplificar la implementación de las redes informáticas en el hogar y los entornos corporativos.