

Universidad de las Ciencias Informáticas

Facultad 7



Cugnot Server:
Servidor de imágenes de Vehículos

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

Autores: Augusto Meilán Andricaín
Yordanis Márquez Denis

Tutores: Ing. Rafael Leodan Cardero Álvarez
Ing. Jesmar Ernesto Fajardo Martín

Ciudad de La Habana, Marzo 2009
“Año del 50 aniversario del triunfo de la Revolución”

Declaración de Autoría

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

Autores:

Augusto Meilán Andricaín

Yordanis Márquez Denis

Tutores:

Ing. Rafael Leodan Cardero Álvarez

Ing. Jesmar Ernesto Fajardo Martín

Datos de Contacto

Tutores:

Rafael L. Cardero Álvarez (email rlcordero@uci.cu).

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas, en el año 2008. Su área de investigación está relacionada con el procesamiento digital de imágenes, las técnicas de reconocimiento automático del número de la matrícula y el desarrollo de componentes reutilizables. Ha presentado artículos en el III Taller de Procesamiento de Imágenes y Señales Aplicadas (UCIENCIA 2006), en el V Congreso Nacional (Cuba) de Reconocimiento de Patrones (REC PAT' 2007) y en el III Taller sobre nuevas tecnologías de base de datos y programación avanzada (UCIENCIA 2007). Se desempeña como Coordinador del Área Temática Procesamiento Digital de Imágenes, del Polo Imágenes de la Facultad 7 y es Visionario del Proyecto Cugnot.

Jesmar Ernesto Fajardo Martin (email jefajardo@uci.cu).

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas, en el año 2008. Participó en actividades científicas estudiantiles siendo miembro del Grupo de Procesamiento de Imágenes de la facultad 7. Se desempeña actualmente como Jefe de Proyecto en dicho grupo.

– De Augusto

A mis padres por formarme tal como soy hoy, por todo su apoyo y confianza.

A mi hermana que tanto hubiera querido estar aquí.

*A Juan Andrés por ser el hermano que nunca tuve, a Elier por ser el gran amigo que todos quieren tener,
a Edier por todo lo que ha sido capaz de ayudarme en mi formación.*

A Damaris por su amor y apoyo en la realización de esta tesis.

A Rafael Leodan por su entrega y apoyo.

*A mis amigos en forma general, a mis compañeros de proyecto y a todos los que de una manera u otra han
contribuido en la elaboración de este trabajo.*

A todos muchas gracias.

– De Yordanis

A mi madre, por darme su apoyo incondicional y ser siempre mi guía.

A mis abuelos, más que eso han sido mis padres; a mi hermana.

A mis compañeros de proyecto, a mi tutor,

A Denys y a todos mis amigos.

Muchas Gracias.

Augusto y Yordanis

A la Revolución, a la Universidad de Ciencias Informáticas por habernos dado la oportunidad de cursar la carrera de ciencias informáticas. A GPI por habernos ayudado a mejorar la calidad en la formación como ingenieros. A los participantes en la realización de la Suite Cugnot.

Resumen

Los países subdesarrollados como Cuba, realizan el control de vehículos mayoritariamente de forma manual. Esto genera grandes problemas a las autoridades a la hora de monitorear el tráfico en las ciudades. Para mejorar este proceso, una alternativa viable es desarrollar de una solución informática nacional.

El Grupo de Procesamiento de Imágenes de la facultad 7 de la UCI trabaja en un sistema de control de vehículos del que forma parte el presente trabajo de diploma. Cuyo objetivo es desarrollar una solución informática para el monitoreo del repositorio de imágenes en el sistema de control de vehículos Cugnot Suite.

Para el desarrollo de la solución propuesta se utilizan la plataforma .Net, de esta Net Remoting para la comunicación entre procesos; como lenguaje de programación C#. Como gestor de base datos PostgreSQL.

Como resultado se espera que el repositorio de imágenes juegue un papel importante al convertirse en una manera confiable de probar que un auto estuvo en un lugar dado a una hora determinada si ocurriera algún hecho delictivo. El aporte social del presente trabajo está en la creación de un producto que permitirá monitorear y manejar eficientemente las imágenes de vehículos mediante un sistema informático desarrollado en el país.

Palabras claves

Control de vehículos, monitoreo de imágenes

Tabla de Contenidos

INTRODUCCIÓN.....	- 9 -
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	- 14 -
1.1. PROTOCOLOS Y MECANISMOS PARA LA TRANSMISIÓN DE ARCHIVOS.....	- 14 -
1.2. PROTOCOLOS TCP/IP.....	- 18 -
1.3. PROTOCOLO FTP	- 20 -
1.4. NET Y LA TRANSFERENCIA DE ARCHIVOS.....	- 25 -
1.5. SERVICIOS DE WINDOWS.....	- 27 -
1.6. ACTUALIDAD	- 27 -
1.7. CONSUMO DE RECURSOS DE LA COMPUTADORA	- 30 -
1.8. ARQUITECTURA DE SOFTWARE	- 31 -
1.9. PLUG-IN.....	- 35 -
1.10. HERRAMIENTAS Y METODOLOGÍAS UTILIZADAS.....	- 36 -
1.11. JUSTIFICACIÓN DE LAS HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.....	- 40 -
1.12. IMPORTANCIA DE UN SERVIDOR DE IMÁGENES EN UN SISTEMA DE CONTROL DE VEHÍCULOS.....	- 41 -
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	- 42 -
2.1. DESCRIPCIÓN DEL NEGOCIO	- 42 -
2.2. MODELO DEL NEGOCIO	- 43 -
2.3. PROPUESTA DE SISTEMA	- 46 -
2.4. REQUISITOS DEL SISTEMA.....	- 47 -
2.5. CASOS DE USO DEL SISTEMA	- 49 -
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	- 57 -
3.1. MODELO DE ANÁLISIS.....	- 57 -
3.2. DIAGRAMAS DE CLASES DEL ANÁLISIS.....	- 58 -
3.3. MODELO DE DISEÑO	- 59 -
3.4. DIAGRAMAS DE INTERACCIÓN DEL DISEÑO	- 61 -
3.5. DIAGRAMA CLASES DEL DISEÑO	- 64 -
3.6. DESCRIPCIÓN DE LAS CLASES	- 67 -
CAPÍTULO 4: IMPLEMENTACIÓN	- 68 -
4.1. MODELO DE DATOS.....	- 69 -
4.2. DIAGRAMA DE DESPLIEGUE	- 69 -
4.3. DIAGRAMA DE COMPONENTES.....	- 70 -

4.4. TRATAMIENTO DE ERRORES	- 70 -
4.5. ESTRATEGIAS DE SEGURIDAD	- 71 -
4.6. ESTRATEGIAS DE CODIFICACIÓN.....	- 71 -
CONCLUSIONES.....	- 72 -
RECOMENDACIONES	- 73 -
REFERENCIAS BIBLIOGRÁFICAS.....	- 74 -
BIBLIOGRAFÍA.....	- 76 -
ANEXOS.....	- 79 -
ANEXOS 1: DESCRIPCIÓN DE LAS CLASES INTERFACES.....	- 79 -
ANEXOS 2: DESCRIPCIÓN DE LA CLASE CONTROLADORA	- 91 -
ANEXOS 3: DESCRIPCIÓN DE LAS CLASES ENTIDAD.....	- 96 -

INTRODUCCIÓN

Debido al propio desarrollo de la humanidad, el uso de los medios de transporte se ha convertido en un servicio imprescindible para la vida cotidiana. Se estima que existen actualmente más de quinientos millones de automóviles en las carreteras de todo el mundo. El aumento en la circulación de autos, conjuntamente con otros factores, ha incrementado los accidentes del tránsito. Así como, las acciones delictivas donde se ven involucrados, de una manera u otra, algún tipo de equipo automotor. Por ello, la identificación de vehículos es una necesidad para los sectores encargados del control y la seguridad social.

Actualmente para diferenciar los equipos automotores se les asigna un número de matrícula como identificador primario. Este implica un permiso legal para participar en el tránsito público. Todos los automóviles deben tener su número de licencia en una placa montada sobre su cuerpo (por lo menos en el lado trasero); ningún vehículo debe circular por las carreteras sin la placa legible, correctamente montada o bien visible.

A nivel internacional los procesos asociados al análisis, clasificación y generación de reportes dinámicos de datos se realizan de manera automatizada mediante el uso de las tecnologías de la información y las comunicaciones. Si estos datos ya están digitalizados, su procesamiento es mucho más simple de realizar en cualquier sistema informático. Cuando se trata del desarrollo de una aplicación informática que maneje información asociada a vehículos, no basta con el número de la licencia, aunque este sea su identificador principal.

Si el responsable de la seguridad de una compañía quisiera tener un sistema que informe en tiempo real en el lugar donde están los autos de la compañía: en el garaje o en la carretera. Registrando cada entrada y cada salida del garaje, el sistema podría informar siempre la ubicación de cada auto. La cuestión clave es que el registro del movimiento de los vehículos debe ser realizado automáticamente por el sistema. De otra manera, requeriría personal para el procesamiento manual de toda esta información, la cual tiende a aumentar progresivamente.

Los sistemas de reconocimiento automático de matrículas (ANPR¹, por sus siglas en inglés) efectúan el reconocimiento de la placa para automatizar la entrada de datos. En sentido estricto, estos son dispositivos integrados de hardware + software que leen la placa de la licencia de los vehículos.

¹ Sistema de reconocimiento de matriculas de automóviles.

Introducción

Para la adquisición de imágenes los dispositivos más utilizados son las cámaras de contexto y las cámaras infrarrojas. Las primeras, capturan la imagen de la escena real en escala de grises o en un modelo de color (generalmente RGB²); se utilizan cuando se requiere visualizar las imágenes de los vehículos a observadores humanos. Por su parte, las cámaras infrarrojas operan en el rango infrarrojo del espectro electromagnético y se utilizan mayoritariamente para garantizar la visión nocturna. (1)

La cámara envía la toma fotográfica del automóvil hacia las estaciones de reconocimiento. Allí le son aplicados algoritmos para detectar el lugar de la matrícula, extraer los caracteres de la imagen, validar la sintaxis y formular la posible matrícula del vehículo. Finalmente, la respuesta depende de la exactitud de trabajo del sistema.

Los sistemas ANPR se utilizan de dos modos: uno, permite que el proceso sea realizado en su totalidad en el lugar de la toma en tiempo real. Mientras que el otro, transmite las imágenes de varias cámaras a una computadora remota donde se realiza posteriormente el proceso de Reconocimiento Óptico de Caracteres (por sus siglas en inglés OCR³). En el primer modo, la información capturada de la matrícula alfanumérica, fecha y hora, identificación del lugar y cualquier otra debe ser completada en unos 250 milisegundos. La misma es convertida en pequeños paquetes de datos y puede ser transmitida fácilmente a algún servidor remoto para un posterior procesamiento, o almacenada en el lugar para su posterior recuperación. En el otro modo debe existir una gran cantidad de computadoras montadas en una red de servidores pues se manejan altas cargas de trabajo.

Los servidores utilizados por el motor ANPR además de permitir guardar las imágenes generadas pueden servir como constancias de la ubicación de un vehículo en cierto lugar y hora. Actualmente si existe un acto delictivo que involucre un automóvil sólo se puede verificar de forma manual y existen inconvenientes como la oscuridad, poca visibilidad o equivocación del personal que dificultan poder esclarecer si el auto estaba en el lugar. El servidor de imágenes contribuye a erradicar estos problemas pues con una imagen del vehículo se puede probar su estancia en el lugar.

En ocasiones la utilización de un servidor exige la necesidad de remitir las imágenes a otro servidor remoto. Este es el encargado de monitorear las imágenes que se encuentren almacenadas en la computadora utilizada como repositorio de imágenes.

² Red, Green and Blue.

³ Optical Character Recognition.

Introducción

Los servidores de imágenes de vehículos presentan algunas de las dificultades como son: requerir en un momento dado, un gran ancho de banda para la transmisión; ser muy dependientes del hardware. Si no se cuenta con un hardware suficientemente poderoso como para cubrir estas necesidades puede ocurrir un funcionamiento no esperado. Las exigencias de hardware provocan que el precio de adquisición e instalación de un servidor sea muy alto ya que para poner en marcha un servidor se debe adquirir además de este el software. Todo esto conlleva a que la mayoría de los países en vía de desarrollo no puedan adquirir este tipo de tecnologías.

En el caso de Cuba, la situación económica y financiera tampoco permite comprar soluciones de control de vehículos debido a sus altos precios. Aunque se han invertido cuantiosos recursos en cámaras de seguridad que se han instalado en las esquinas de las calles más importantes, lugares de interés económico y puntos de control. Estas se utilizan principalmente para la vigilancia y el seguimiento de autos en tiempo real. Por tanto, el control de vehículos es mantenido fundamentalmente por los guardias de seguridad mediante el chequeo de monitores. Ello tiene como principal desventaja que si algún auto no es detectado por el guardia correspondiente en el intervalo de tiempo que visualiza la imagen, este circula por la ciudad con plena libertad.

Todo esto, permite plantear que existe en Cuba la necesidad de desarrollar una solución propia para dar respuesta a los problemas de control de vehículos utilizando la transmisión y el monitoreo de las imágenes generadas. Pues, al consultarlas posteriormente será posible detectar infracciones no vistas por los guardias de seguridad, así como dar una posible ubicación de un vehículo infractor buscado. Esta solución tendrá en cuenta que el mayor peso en el procesamiento de las imágenes debe recaer en el software.

De lo anterior se identificó el siguiente **problema:** ¿Cómo facilitar el monitoreo del repositorio de imágenes en un sistema de control de vehículos?

Para dar respuesta al problema planteado, queda definido como ***objeto de estudio*** el proceso de monitoreo de imágenes; a raíz de lo cual el ***campo de acción*** se enfoca en la automatización de este proceso en la solución de control de vehículos (Cugnot Suite). Para lo cual es fundamental el uso las nuevas tecnologías informáticas en aras de potenciar la confiabilidad, la estabilidad, la autonomía y las facilidades de configuración en el sistema.

El **objetivo general de la investigación:** desarrollar una solución informática para el monitoreo del repositorio de imágenes en el sistema de control de vehículos Cugnot Suite.

Introducción

Para dar cumplimiento a los objetivos propuestos se identificaron las siguientes **tareas**:

- Identificar mecanismos y protocolos que permitan transmitir archivos.
- Valorar los principales sistemas para el control de vehículos existentes a nivel mundial.
- Gestionar las funcionalidades que ofrece la plataforma .NET para el desarrollo de aplicaciones de monitoreo y transmisión de archivos.
- Identificar las principales características del gestor de Base de Datos PostgreSQL.
- Identificar las prestaciones de plataforma .NET para la consulta del consumo de recursos.
- Realizar el análisis y el diseño de la aplicación.
- Implementar el diseño propuesto.

El aporte social del presente trabajo de diploma está dado por la creación de un producto, cuya aplicación permitirá monitorear y manejar eficientemente las imágenes de vehículos mediante un sistema informático desarrollado en el país.

En la investigación se utilizaron métodos teóricos como: el histórico lógico para caracterizar los procesos de desarrollo de software, las tecnologías y herramientas utilizadas en su desarrollo. Además del sistémico para una mejor comprensión orientada a los diferentes componentes y sus relaciones.

El documento está conformado por cuatro capítulos, a través de los cuales se ofrece una información más pormenorizada del objeto de la investigación:

El **Capítulo 1** Fundamentación Teórica, constituye el respaldo teórico de la investigación. Se realiza un análisis de las herramientas y tecnologías utilizadas en los procesos de control de vehículos.

El **Capítulo 2** Características del Sistema, se describen los procesos de monitoreo de imágenes, así como la propuesta del sistema. También incluye los requerimientos funcionales y no funcionales; los modelos del negocio y del sistema.

En el **Capítulo 3** Análisis y diseño del sistema, se modela en detalle el sistema mediante las clases del diseño, describiendo también las interacciones entre estas y se especifica el flujo de eventos entre las mismas a través de los diagramas de secuencia.

Introducción

En el **Capítulo 4** Implementación, se realiza la representación de los componentes finales resultantes y la interacción entre estos; así como el modelo de despliegue, el cual define la forma en que debe realizarse la implantación del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se realiza un análisis crítico de las principales herramientas y tecnologías utilizadas en los procesos de control de vehículos y se fundamenta la necesidad de desarrollar una solución informática capaz de monitorear las imágenes del sistema de control de vehículos.

A continuación se analizan los protocolos de transmisión de archivos más utilizados en la actualidad para el movimiento de datos. Estos brindan facilidades para el manejo de información entre distintos tipos de ordenadores que los hacen altamente demandados. Entre los distintos protocolos que existen uno de los más utilizados es el protocolo FTP (File Transfer Protocol).

1.1. Protocolos y Mecanismos para la Transmisión de Archivos

La familia de protocolos de Internet es un conjunto de protocolos de red en la que esta se basa y permiten la transmisión de datos entre redes de computadoras. En ocasiones se le denomina *conjunto de protocolos TCP/IP*⁴, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

Entre los protocolos a nombrar de esta familia se encuentran *HTTP* (HyperText Transfer Protocol), utilizado para acceder a sitios Web, el *ARP* (Address Resolution Protocol) para la resolución de direcciones, el *SMTP* (Simple Mail Transfer Protocol), *POP* (Post Office Protocol) para correo electrónico y el *FTP* para transferencia de archivos.

TCP/IP puede describirse por analogía como parte del modelo OSI⁵

El Modelo de Referencia de Interconexión de Sistemas Abiertos, conocido mundialmente como Modelo OSI, fue creado por la ISO⁶ y en él pueden modelarse o referenciarse diversos dispositivos que reglamenta la ITU⁷, con el fin de poner orden entre todos los sistemas y componentes requeridos en la transmisión de datos, además de simplificar la interrelación entre fabricantes. Así, todo dispositivo de

⁴ Transmission Control Protocol & Internet Protocol.

⁵ Open System Interconnection.

⁶ Organización Estándar Internacional.

⁷ Unión de Telecomunicación Internacional.

Capítulo 1

cómputo y telecomunicaciones podrá ser referenciado al modelo y por ende concebido como parte de un sistema interdependiente con características muy precisas en cada nivel. (2)

El objetivo perseguido por OSI establece una estructura que presenta las siguientes particularidades: (3)

Estructura multinivel: Se diseñó una estructura multinivel con la idea de que cada nivel se dedique a resolver una parte del problema de comunicación. Esto es, cada nivel ejecuta funciones específicas.

El nivel superior utiliza los servicios de los niveles inferiores: Cada nivel se comunica con su similar en otras computadoras, pero debe hacerlo enviando un mensaje a través de los niveles inferiores en la misma computadora. La comunicación entre los diferentes niveles está bien definida. El nivel N utiliza los servicios del nivel N-1 y proporciona servicios al nivel N+1.

Puntos de acceso: Entre los diferentes niveles existen interfaces llamadas "puntos de acceso" a los servicios.

Dependencias de Niveles: Cada nivel es dependiente del nivel inferior y también del superior.

Encabezados: En cada nivel, se incorpora al mensaje un formato de control. Este elemento de control permite que un nivel en la computadora receptora se entere de que su similar en la computadora emisora esta enviándole información. Cualquier nivel dado, puede incorporar un encabezado al mensaje. Por esta razón, se considera que un mensaje esta constituido de dos partes: Encabezado e Información. Entonces, la incorporación de encabezados es necesaria aunque representa un lote extra de información, lo que implica que un mensaje corto pueda ser voluminoso. Sin embargo, como la computadora destino retira los encabezados en orden inverso a como fueron incorporados en la computadora origen, finalmente el usuario sólo recibe el mensaje original.

El Modelo OSI cuenta con 7 capas o niveles (4) vea la **Figura 1.1:**

Capítulo 1



Figura 1.1 Modelo OSI.

El nivel Físico describe las características físicas de la comunicación, como las convenciones sobre la naturaleza del medio usado para la comunicación (comunicaciones por cable, fibra óptica o radio), y todo lo relativo a los detalles como los conectores, código de canales y modulación, potencias de señal, longitudes de onda, sincronización, temporización y distancias máximas.

El nivel de Enlace especifica cómo son transportados los paquetes sobre el nivel físico, incluyendo los delimitadores (patrones de bits concretos que marcan el comienzo y el fin de cada trama). Ethernet⁸, por ejemplo, incluye campos en la cabecera de la trama que especifican que máquina o máquinas de la red son las destinatarias de la trama.

⁸ International standard networking technology for wired implementations.

Capítulo 1

El nivel de red está basado en el intercambio de datos entre una red origen y una red destino. Incluye un enrutamiento de paquetes a través de una red de redes, conocida como Internet. Realiza las tareas básicas para conseguir transportar datos desde un origen a un destino. Puede pasar los datos a una serie de protocolos superiores; cada uno de esos protocolos es identificado con un único "Número de protocolo IP". ICMP⁹ e IGMP¹⁰ constituyen los protocolos 1 y 2, respectivamente.

El nivel de Transporte utiliza los servicios del nivel de red para proveer un servicio eficiente y confiable a sus clientes, que normalmente son los procesos en el nivel de aplicación. El hardware y software dentro de este nivel se llaman la *entidad de transporte*. Por lo tanto ¿por qué no solamente el nivel de red? La razón es los usuarios no tienen ningún control sobre él. El nivel de transporte permite que los usuarios puedan mejorar el servicio del nivel de red.

El nivel de Sesión se encarga de llevar a cabo un control de las sesiones que se establecen sobre los servicios ofrecidos en la red y emular una comunicación dúplex. Lleva a cabo funciones de sincronismo, que permiten restablecer una comunicación en caso de que se produzca algún corte. También tiene la posibilidad de establecer prioridades a las distintas sesiones establecidas.

El nivel de presentación define el formato de los datos que se van a intercambiar entre las aplicaciones y ofrece a los programas de usuario un conjunto de servicios relacionados con la transformación de datos: compresión de datos y su cifrado. En resumen, se encarga de adaptar los datos a la forma en la que serán tratados por la máquina.

El nivel de aplicación es el nivel que los programas utilizan para comunicarse a través de una red con otros programas. Los procesos que acontecen en este nivel son aplicaciones específicas que pasan los datos al nivel de aplicación en el formato que internamente use el programa y es codificado de acuerdo con un protocolo estándar. Una vez que los datos de la aplicación han sido codificados en un protocolo estándar del nivel de aplicación son pasados *hacia abajo* al siguiente nivel de la pila de protocolos TCP/IP. En el nivel de transporte, las aplicaciones normalmente hacen uso de TCP y UDP¹¹ y son habitualmente asociados a un número de puerto bien conocido (*well-known port*). (5)

⁹ Internet Control Message Protocol.

¹⁰ Internet Group Management Protocol.

¹¹ User Datagram Protocol.

Capítulo 1

1.2. Protocolos TCP/IP

Internet no es un nuevo tipo de red física, sino un conjunto de tecnologías que permiten interconectar redes muy distintas entre sí; no es dependiente de la máquina ni del sistema operativo utilizado. De esta manera, se puede transmitir información entre un servidor Unix y un ordenador que utilice Windows 98. O entre plataformas completamente distintas como Macintosh, Alpha o Intel. Entre una máquina y otra generalmente existen distintos tipos de redes: redes Ethernet, redes Token Ring e incluso enlaces vía satélite. No se puede utilizar ningún protocolo que dependa de una arquitectura en particular. Lo que se necesita es un método de interconexión general que sea válido para cualquier plataforma, sistema operativo y tipo de red.

La familia de protocolos que se eligieron para permitir que Internet sea una red de redes es TCP/IP. Es una familia de protocolos, pues son muchos los que la integran, aunque en ocasiones para simplificar, se utilice sencillamente el término protocolo TCP/IP. Vea **Figura 1.2**.

Los protocolos TCP/IP tienen que estar a un nivel superior del tipo de red empleado, funcionar de forma transparente en cualquiera de esta y a un nivel inferior de los programas de aplicación (páginas WEB, correo electrónico) particulares de cada sistema operativo.

TCP/IP tiene el siguiente modelo de referencia (6)

Capa de aplicación (HTTP, SMTP, FTP, TELNET...)
Capa de transporte (UDP, TCP)
Capa de red (IP)
Capa de acceso a la red (Ethernet, Token Ring...)
Capa física (cable coaxial, par trenzado...)

Figura 1.2 Estructura de TCP/IP.

El nivel más bajo es la capa física. Este es el medio físico por el cual se transmite la información. Generalmente es un cable aunque no se descarta cualquier otro medio de transmisión como ondas o enlaces vía satélite.

La capa de acceso a la red determina la manera en que las estaciones (ordenadores) envían y reciben la información a través del soporte físico proporcionado por la capa anterior.

Capítulo 1

Las dos capas anteriores quedan a un nivel inferior del protocolo TCP/IP, es decir, no forman parte de este protocolo. La capa de red define la forma en que un mensaje se transmite a través de distintos tipos de redes hasta llegar a su destino. El principal protocolo de esta capa es el IP aunque también se encuentran a este nivel el protocolo ARP, ICMP e IGMP. Esta capa proporciona el direccionamiento IP y determina la ruta óptima a través de los *routers* que debe seguir un paquete desde el origen al destino.

La capa de transporte (protocolos TCP y UDP) ya no se preocupa de la ruta que siguen los mensajes hasta llegar a su destino. Sencillamente, considera que la comunicación extremo a extremo está establecida y la utiliza. Además añade la noción de puertos, como se detalla más adelante.

La capa de aplicación proporciona los distintos servicios de Internet: correo electrónico, páginas Web, FTP, TELNET, entre otros.

Principales protocolos de TCP/IP (7)

- SMTP *Simple Mail Transport Protocol* o Protocolo Simple de Transferencia de correo proporciona servicios de correo electrónico en las redes Internet e IP.
- TCP *Transport Control Protocol* o Protocolo de Control de Transporte es un protocolo de transporte orientado a la conexión. TCP gestiona la conexión entre las computadoras emisora y receptora de forma parecida al desarrollo de las llamadas telefónicas.
- UDP *User Datagram Protocol* o Protocolo de Datagrama de Usuario es un protocolo de transporte sin conexión que proporciona servicios en colaboración con TCP.
- IP *Internet Protocol* o Protocolo Internet es la base para todo el direccionamiento que se produce en las redes TCP/IP y proporciona un protocolo orientado a la capa de red sin conexión. Funciona de forma semejante a una carta con remite echada al buzón y después entregada a su destinatario.

Capítulo 1

- *ARP Address Resolution Protocol* o Protocolo de Resolución de Direcciones hace corresponder las direcciones IP con las direcciones MAC¹² de hardware.

1.3. Protocolo FTP

El protocolo FTP¹³ promueve el intercambio de archivos. Entre sus principales opciones está fomentar directa o indirectamente el uso de equipos remotos, proteger a un usuario de las variaciones en los sistemas de almacenamiento de archivos entre hosts y transferir datos fiable y eficientemente. Aunque directamente utilizable por un usuario en un terminal, está diseñado principalmente para su uso por programas. El intento en este pliego de condiciones para satisfacer las diversas necesidades de los usuarios. (8)

Se implementa mediante el sistema cliente-servidor. En el equipo remoto se ejecuta el programa servidor FTP, en el equipo local se ejecuta un programa cliente. Este hace pedidos o solicita tareas de transmisión o manejo de archivos las que el servidor atiende.

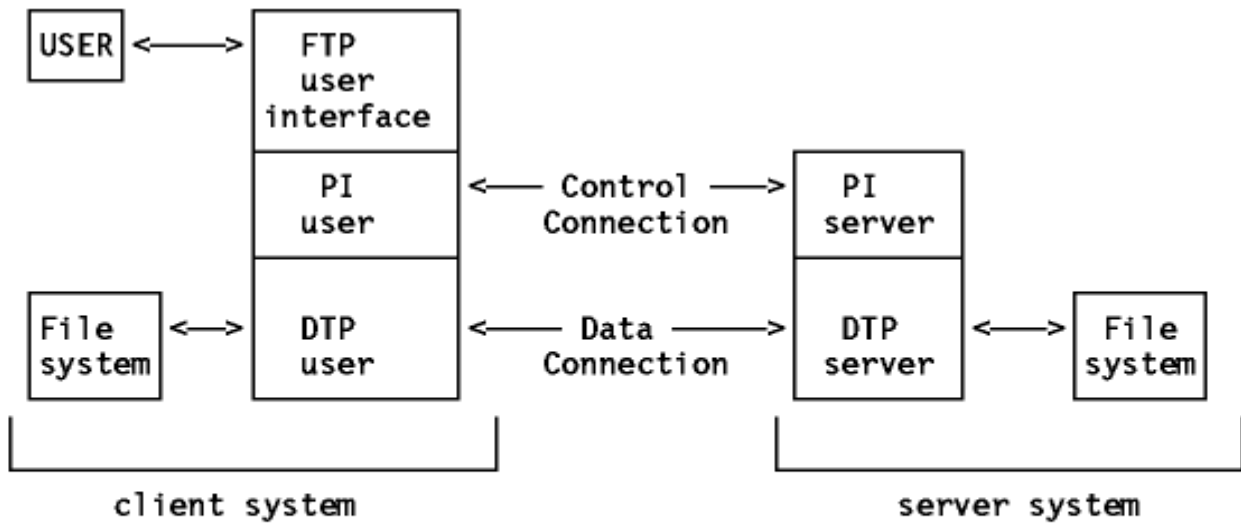
El protocolo FTP trabaja en la capa 7 del modelo OSI, utilizando normalmente los puertos 20 y 21. La mayor ventaja de su uso es la rapidez con que se realiza la transferencia de archivos aunque las transferencias se realizan en dato claro. Como solución a este problema, los autores consideran adoptar una nueva versión de este protocolo llamado *File Transfer Protocol Secure (FTPS)*, que permite la transferencia de manera cifrada por medio de SSL¹⁴. Para entender mejor el flujo de FTP vea la **Figura 1.3**.

¹² Media Access Control.

¹³ File Transfer Protocol.

¹⁴ Secure Sockets Layer.

Capítulo 1



PI : protocol interpreter
DTP: data transfer process

Figura 1.3 Flujo de FTP.

1.3.1. Servidor FTP

Es un programa especial que se ejecuta en un equipo servidor. Su función es permitir la transferencia de archivos entre equipos. Las aplicaciones más comunes de los servidores FTP suelen ser el alojamiento web, en el que sus clientes utilizan el servicio para subir sus páginas web y sus archivos correspondientes o como servidor de respaldo.

FTP tiene dos modalidades de uso

FTP Anónimo: Supone un servidor FTP configurado para permitir el acceso público, el sistema se ajusta a una clave de acceso público para el acceso anónimo a todos los archivos que se han compartido.

FTP Privado: Se basa en la autenticación a partir de la base de datos de usuarios locales, sólo pueden iniciar sesión los usuarios que hayan sido dados de alta en dicho sistema.

Capítulo 1

1.3.2. Algunos servidores FTP

Gene6 FTP Server: (9)

- **Plataforma:** Windows 98/Me/NT/2000/XP, Serv2003.
- **Última actualización:** 06/13/2006
- **Versión:** v3.8.0.34.
- **Tamaño:** 3,94 MB
- **Licencia:** Shareware

Sitio de descarga <http://www.softbull.com/download.php?id=5372>

Principales características

- Encriptación SSL
- Administración remota segura
- Completa vista de dominios FTP y lista de usuarios.
- Funciones de seguridad
- Soporte de estándares FTP
- Interfaz intuitiva

Serv-U FTP Server: (10)

- **Plataforma:** Windows95, 98, 98SE, NT, 2000, ME, XP, 2003.
- **Última actualización:** 03/05/2004
- **Versión:** 5.0.0.4.
- **Tamaño:** 3,50 MB
- **Licencia:** Shareware (30 días)
- **Sitio de descarga** <http://www.softbull.com/download.php?id=345>

Principales características

- Gestión de usuarios
- Bloqueo de acceso

Capítulo 1

- Control de usuarios online
- Estadísticas de acceso
- Configuración de la velocidad de descarga

FileZilla Server: (11)

- **Plataforma:** Windows98, 98SE, Me, NT, 2000, XP, Serv2003.
- **Última actualización:** 03/04/2004
- **Version:** 2.2.4e.
- **Tamaño:** 2,034 MB
- **Licencia:** Freeware
- **Sitio de descarga** <http://filezilla-server.softonic.com/descargar>

Principales características:

- Soporte de idioma español durante la ejecución del programa, no así el instalador, donde no se ofrece esa posibilidad.
- Capacidad para continuar descargas interrumpidas.
- Herramienta de administrador de sitios FTP.
- Capacidad para mantener viva la conexión con el servidor FTP.
- Soporte para trabajar junto con firewalls.
- Soporte SOCKS4/5 y HTTP1.1.
- Conexiones seguras SSL y SFTP.
- Cola de cargas y descargas.
- Soporte para Drag & Drop (Arrastrar y soltar).
- Posibilidad de trabajo con múltiples conexiones.

Capítulo 1

WS_FTP Server: (12)

- **Plataforma:** Windows98, 98SE, Me, NT, 2000, XP, Serv2003.
- **Última actualización:** 03/04/2004
- **Version:** 5.0.
- **Tamaño:** 6,86 MB
- **Licencia:** Shareware
- **Sitio de descarga** <http://ws-ftp-server.programas-gratis.net>

Principales características:

- Analizador de registros.
- Asistentes.
Cifrado de 128 bits con el estándar Capa de conexión segura (SSL).
- Autenticación de usuarios, para NT, basada en un controlador independiente de dominios.
- Controles del acceso de los usuarios.
- Carpetas virtuales.
- Ocultar la existencia de las carpetas de otros usuarios.
- Administración remota.
- Integración con ODBC.
- Comandos de sitio personalizados.
- Opciones de seguridad configurables.
- Recuperación automática.
- Bases de datos de usuarios.
- Utilidad de línea de comandos para agregar usuarios.

Capítulo 1

1.3.3. Cliente FTP

Es un programa que emplea el protocolo FTP para conectarse a un servidor y transferir archivos. Para hacer uso de este se necesita conocer: el nombre del archivo, la dirección en que reside el servidor, la dirección a que se quiere transferir el archivo y la carpeta en que se encuentra. La conexión para la transferencia se puede hacer de forma pasiva o activa. En la forma activa el cliente le indica al servidor que puerto tiene que abrir para la conexión, no así en la forma pasiva donde el servidor indica a los clientes los puertos por donde realizar la conexión. De esta última forma, se garantiza una mejor seguridad e integridad de los datos transferidos y del servidor. (13)

Tipos de transferencia de archivos en FTP

*Tipo ASCII*¹⁵: Adecuado para transferir archivos que sólo contengan caracteres imprimibles por ejemplo páginas HTML sin imágenes.

Tipo binario: Usado para transferir archivos comprimidos, ejecutables, imágenes, archivos de audio, etc. (14) (15)

1.4. Net y la Transferencia de Archivos

Para la transferencia de archivos la plataforma .Net cuenta con la clase *System.Net.WebRequest* que es la clase base abstracta del modelo de solicitud/respuesta para obtener acceso a datos desde Internet. Una aplicación puede solicitar datos de Internet de una manera que sea válida para los protocolos, donde la aplicación trabaja con instancias de la clase *WebRequest* mientras que las clases descendientes específicas del protocolo se encargan de los detalles de la solicitud. (16)

Las solicitudes se envían desde una aplicación a un identificador URL¹⁶ concreto, como una página Web de un servidor. El identificador URI determina la clase descendiente adecuada que se debe crear a partir de una lista de descendientes de *WebRequest* registrados para la aplicación. Los descendientes suelen registrarse para controlar un protocolo específico, como HTTP¹⁷

¹⁵ American Standard Code for Information Interchange

¹⁶ Uniform Resource Locator

¹⁷ Hypertext Transfer Protocol

Capítulo 1

(*HttpWebRequest*) o FTP (*FtpWebRequest*). También se pueden registrar para controlar una solicitud a un servidor específico o una ruta de acceso.

Esta clase produce *WebException* cuando se producen errores en el acceso a un recurso de Internet. La propiedad *Status* es uno de los valores de *WebExceptionStatus* que indica el origen del error. Cuando *Status* es *WebExceptionStatus.ProtocolError*, la propiedad *Response* contiene el valor de *WebResponse* recibido del recurso de Internet.

Dado que *WebRequest* es una clase abstracta, el comportamiento real de las instancias de *WebRequest* en tiempo de ejecución viene determinado por la clase descendiente devuelta por el método *Create*.

1.4.1. Clase *FtpWebRequest* (17)

Para obtener una instancia de ***FtpWebRequest***, se utiliza el método *Create*, para cargar y descargar información de un servidor FTP. Cuando se especifica un recurso de red que utiliza el esquema FTP, la clase ***FtpWebRequest*** proporciona la capacidad para interactuar mediante programación con servidores de este tipo.

El identificador URL puede ser absoluto o relativo. Si este tiene la forma "ftp://Algo.com/%2fpath" (% 2f es el carácter de escape de '/'), el URL es absoluto y el directorio actual es /path. Si la forma es ftp://algo.com/path, .NET Framework inicia la sesión en el servidor FTP mediante el nombre de usuario y contraseña establecidos por la propiedad *Credentials*. A continuación, el directorio actual se establece en <UserLoginDirectory>/path.

Se debe tener un nombre de usuario y contraseña válidos para el servidor o bien el servidor debe permitir el inicio de sesión anónimo. Para especificar las credenciales que se utilizan para conectarse al servidor, se establece mediante la propiedad *Credentials* o incluyéndolas en la parte *User/Info* del identificador URL pasado al método *Create*.

Capítulo 1

1.5. Servicios de Windows

En los sistemas operativos Microsoft Windows, un servicio es un ejecutable de larga duración que lleva a cabo funciones específicas y que está diseñado para no requerir la intervención del usuario. Estos pueden ser configurados para iniciarse cuando el sistema operativo se inicia y se ejecutan en segundo plano, o puede ser iniciado manualmente cuando sea necesario. Son similares en concepto a un demonio de Unix.

Muchos aparecen en la lista de procesos en el Administrador de tareas de Windows, la mayoría de las veces con un nombre de usuario de sistema, servicio local o de red. No obstante, no todos los procesos con el nombre de usuario son servicios. El resto de servicios se ejecutan a través de svchost.exe como dlls cargadas en memoria.

1.6. Actualidad

En la actualidad, existe gran competencia para lograr el liderazgo a nivel mundial en el desarrollo de soluciones informáticas para el control de vehículos. Algunas compañías europeas van a la cabeza de esta, entre las principales empresas se encuentran:

1.6.1. Computer Recognition Systems Ltd (18)

La compañía Computer Recognition Systems Ltd (CRS) es una transnacional cuya sede central radica en Wokingham, Inglaterra. Los sectores cubiertos por sus productos son la gestión de parqueos, la recolección automática de peaje y el monitoreo del tráfico en las autopistas. Su producto estrella es el sistema NRS. Este sistema es capaz de operar con cámaras digitales, cámaras analógicas, iluminación visible e iluminación infrarroja. Además, soporta los medios de transmisión de datos GSM inalámbrico, GPRS inalámbrico, LAN, PSTN, ISDN y ADSL.

Es imprescindible destacar que esta compañía ha sido pionera en el desarrollo de soluciones para la identificación vehicular. Algunos de sus logros son el desarrollo del primer lector de matrículas (año 1979), el desarrollo del primer lector de matrículas capaz de operar sin detectores de presencia (año 1983) y constituir la primera compañía del sector en introducir las cámaras digitales HDTV en sus soluciones (año 1997).

Capítulo 1

1.6.2. Anpr International Ltd (19)

La compañía Anpr International Ltd es un líder mundial cuya sede radica en Sheffield, Inglaterra. Los sectores cubiertos por sus productos son la gestión de parqueos, el control de acceso a las instalaciones y el monitoreo del tráfico en las autopistas. Sus principales productos son eyeTRAFFIC™ y streetSWEEPER™.

El sistema eyeTRAFFIC™ es una solución integral para monitorear el tráfico de los vehículos. Este sistema gestiona la disponibilidad de las cámaras, soporta varios medios para transmitir datos y es capaz de obtener los detalles de cada vehículo desde la base de datos DVLA (Driver and Vehicle Licensing Agency) o cualquier otra base de datos nacional de vehículos. Sus cámaras pueden trabajar cuando la red no está disponible (de hecho, puede capturar los datos de matrícula de hasta 14 días).

El sistema streetSWEEPER™ constituye una unidad móvil que permite monitorear el tráfico de los vehículos. Está diseñado para:

- Capturar automáticamente datos de la chapa vehículo.
- Analizar los datos en DVLA para buscar números licencia y / o infractores.
- Activar una señal de alarma cuando se detecta a un infractor.
- Edición de una multa al infractor.

1.6.3. Security Design Services (20)

Security Design Services con su producto *SDS ANPR SYSTEM*, es una de las principales empresas del sector. La exactitud de captura de imágenes vehiculares de este sistema oscila entre el 98% y el 99%. Registra imágenes de vehículos a velocidades de hasta 120 MPH (millas por hora), e incluye además una base datos donde es almacenada la información capturada.

Este producto utiliza varios tipos de cámaras, cada una con características específicas. Las cámaras a color capturan las imágenes del vehículo y el conductor, especialmente en puertas y entradas. En la noche sólo se capturan las imágenes si la zona esta bien iluminada.

Otro tipo de dispositivo de video es la cámara mono. Esta tiene como único requisito la captura del número de placa, las imágenes de los vehículos no son importantes. Por último, la cámara dual ofrece las ventajas de la cámara de color y la mono. Son proporcionadas por el día las imágenes a color y durante la noche sólo se captura el número de la placa a partir de imágenes monocromática.

Capítulo 1

1.6.4. Cell2Solution (21)

La compañía Cell2solutions fue fundada en el año 2005. Su sede central radica en Polígono de Silvota, Llanera, Asturias, España. La misión de esta compañía es comercializar y distribuir componentes electrónicos en el mercado Iberoamericano. Algunos de estos componentes son los paneles luminosos y las cámaras ANPR (cámaras que se utilizan para reconocer la matrícula de los vehículos).

El catálogo de productos ofrecido por Cell2solutions es impresionante. La compañía ofrece soluciones para controlar el acceso a las instalaciones, clasificar los vehículos según su modelo, estimar el tiempo de viaje entre 2 puntos, gestionar parqueos y calcular la velocidad media de un vehículo. Además, incluye una unidad móvil para monitorear el tráfico de los vehículos en las autopistas. Todos estos productos son comercializados bajo la marca MaxVision.

1.6.5. Carmen (22)

La compañía Adaptive Recognition Hungary Inc fue fundada en el año 1991. Su sede central radica en Budapest, Hungría. El principal producto de la compañía es CARMEN FreeFlow Este es capaz de realizar la lectura digital de placas en imágenes desde cualquier tipo de fuentes. Tanto si se trata de un mega píxel de tamaño, (12bits/píxel de alto rango dinámico), de alta resolución de imagen digital o un tamaño de 384x288 (colores de vídeo de circuito cerrado de televisión). Para proporcionar las imágenes utiliza cámaras del tipo FXCAM, una avanzada tecnología dedicada a la captura del número de matrículas. Tiene un alcance efectivo de 15 metros, pero puede variar dependiendo de las condiciones ambientales y su exactitud es de más del 98,5% las 24 horas del día.

1.6.6. En Cuba

En Cuba no se cuenta con una solución informática para el control de vehículos, aunque ya se dan los primeros pasos para su desarrollo desde varias instituciones. Entre ellas: el CENATAV (23) (Centro de Aplicaciones de Tecnologías de Avanzada), trabaja en investigaciones teóricas y aplicadas en el área del Reconocimiento de Patrones y la Minería de Datos. Este centro desarrolla una serie de algoritmos para lograr el reconocimiento automáticos de chapas de vehículos. Además en la Universidad de las Ciencias Informáticas desde el 2007 se desarrolla *Cugnot Suite*, por parte del polo productivo de

Capítulo 1

Imágenes y Señales de la facultad 7. Esta es una solución de carácter nacional que brindará las mismas funcionalidades que las soluciones antes expuestas.

1.7. Consumo de Recursos de la Computadora

1.7.1. Espacio en Disco

Las computadoras poseen en su estructura muchos dispositivos de hardware. Uno de los más importantes son los discos duros. En ellos se guarda toda la información de la máquina desde el sistema operativo hasta la información más sensible para cualquier empresa o usuario. De cada unidad o partición se pueden crear otras. De todas ellas es necesario conocer información como: sus propiedades físicas: cuanto puede almacenar, cuanto tiene de espacio libre, entre otros datos de interés.

La plataforma .NET posee muchas librerías que realizan varias funciones. Para la lectura de la información sobre una unidad de disco, específicamente, se utiliza la clase *System.IO.DriveInfo*. Esta clase modela una unidad, y proporciona métodos y propiedades para consultar información sobre la unidad. *DriveInfo* se utiliza para determinar qué unidades están disponibles y de qué tipo son. Así como determinar la capacidad de la unidad y su espacio libre disponible.

1.7.2. RAM en Uso

La RAM¹⁸ es uno de los dispositivos fundamentales en el funcionamiento de una computadora. Por ello, en toda máquina es necesario conocer su estado. Su uso racional hace que en el equipo tanto el sistema operativo como todos los programas que se ejecuten, trabajen sin poner “la máquina lenta”. Por eso se necesita que el sistema operativo consulte el estado de la RAM.

Con la ayuda de la plataforma .NET es posible implementar un algoritmo que permita conocer el estado de la memoria. Usando las Api de Windows se puede acceder a informaciones como son: memoria física total (dado en KB, MB o TB según corresponda), memoria en uso, memoria sin usar, entre otras.

¹⁸ Random Access Memory

1.8. Arquitectura de Software

La arquitectura de software es el conjunto de decisiones significativas sobre la organización de un sistema. Esta permite la selección de los elementos estructurales e interfaces del sistema y su comportamiento en el mismo. Además describe los cimientos que son necesarios como base para comprender, desarrollar y producir económicamente el sistema. La misma se representa en 4+1 vistas arquitectónicas. Vea Figura 1.4.

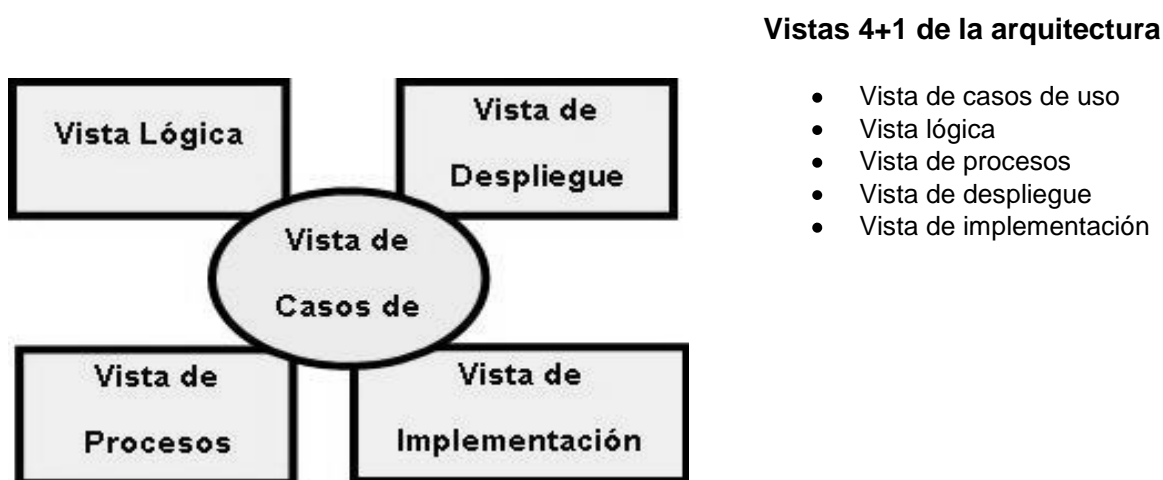


Figura 1.4 4+1 vistas arquitectónicas.

Vista de casos de uso

Esta vista representa un subconjunto del artefacto Modelo de casos de uso y lista los casos de usos o escenarios del modelo de casos de uso más significativos, con las funcionalidades centrales del sistema. Si el sistema se hace extenso entonces se debería organizarse en paquetes, lo cual facilitaría la comprensión de la vista de casos de uso.

Aquí se representa el diagrama de casos de uso de los mismos con una breve descripción de cada uno.

Capítulo 1

Vista lógica

Esta vista representa un subconjunto del artefacto Modelo de diseño, la cual representas los elementos de diseño más importantes para la arquitectura del sistema. Este describe las clases más importantes, su organización en paquetes y subsistemas, y estos a su vez en capas. También describe las realizaciones de casos de uso más importantes como por ejemplo las que describen aspectos dinámicos del sistema.

Vista de procesos

Esta vista suministra una base para la comprensión de la organización de los procesos de un sistema, ilustrados en el mapeo de las clases y subsistemas en procesos e hilos. Sólo suele usarse cuando el sistema presenta procesos concurrentes o hilos.

Vista de despliegue

Esta vista suministra una base para la comprensión de la distribución física de un sistema a través de nodos. Suele utilizarse cuando el sistema está distribuido. Esto incluye la asignación de tareas provenientes de la vista de procesos en los nodos.

Vista de implementación

Esta vista describe la descomposición del software en capas y subsistemas de implementación. También provee una vista de la trazabilidad de los elementos de diseño de la vista lógica ahora para la implementación.

Esta contiene:

- Una enumeración de los subsistemas.
- Diagramas de componentes que ilustran la organización en capas y jerarquías de los subsistemas.
- Dependencia entre subsistemas.

Capítulo 1

1.8.1. Patrones de Arquitectura

Un estilo o variante arquitectónica define a una familia de sistemas informáticos en términos de su organización estructural. Describe sus componentes y las relaciones entre ellos; con las restricciones de su aplicación, la composición asociada y el diseño para su construcción.

Los sistemas empresariales distribuidos pueden agrupar los siguientes estilos arquitectónicos:

- Modelo-Vista-Controlador (MVC)
- Arquitecturas en Capas
- Arquitecturas Orientadas a Objetos
- Arquitecturas Basadas en Componentes
- Arquitecturas Orientadas a Servicios

1.8.2. Arquitecturas en Capas

Este patrón define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, donde los componentes de una capa sólo pueden hacer referencia a otros en capas inmediatamente inferiores. Este simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, ayuda a identificar las partes a reutilizar y proporciona una estructura que ayuda a tomar decisiones sobre qué partes comprar y qué partes construir.

Principales estilos de arquitecturas estratificadas de las aplicaciones distribuidas contemporáneas:

- Arquitecturas de dos niveles.
- Arquitecturas de tres niveles.
- Arquitecturas de n niveles.

Capítulo 1

1.8.3. Arquitectura en Tres Capas

La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido. Vea Figura 1.4.

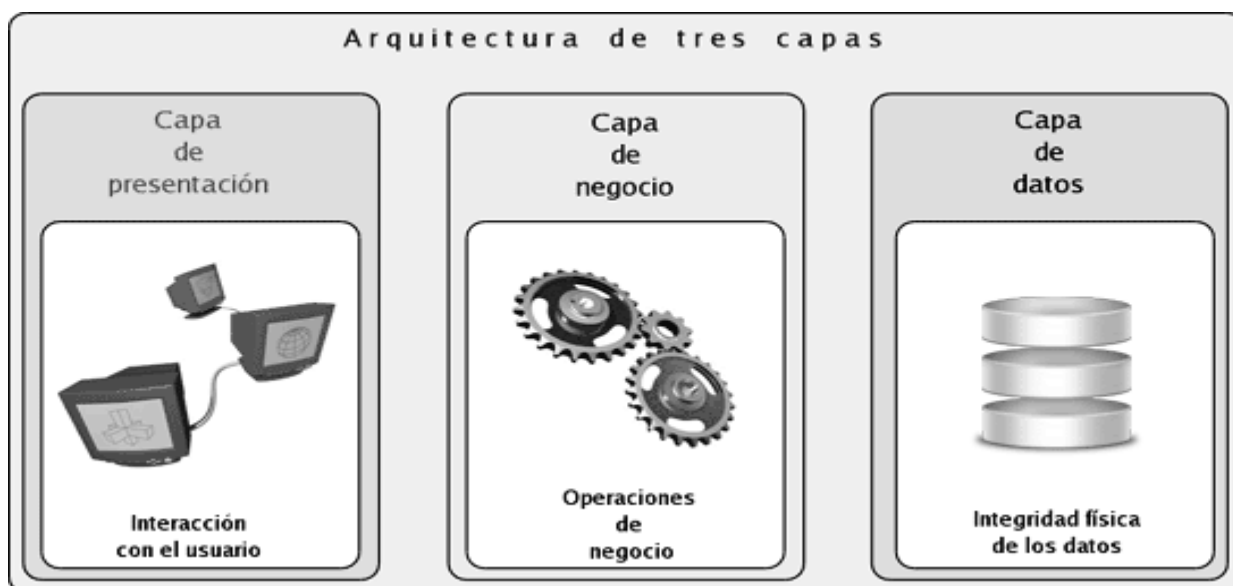


Figura 1.5. Arquitectura de tres capas.

La capa de presentación o interfaz de usuario

Está formada por los formularios y los controles que se encuentran en los formularios y es la capa con la que interactúa el usuario. Es la responsable de presentar la información del sistema al usuario. Reúne todos los aspectos para interactuar y construir las interfaces de usuarios, generalmente involucran el trabajo con ventanas, reportes, gráficos, menús y elementos de multimedia. Con la única capa que puede relacionarse es con la capa de negocio a la cuál le envía los datos que son recogidos en los formularios.

Capítulo 1

La capa de negocio

Se le conoce como capa de lógica de negocio, donde se establece la comunicación con la capa de datos y la capa de presentación. Recibe y responde cada petición que los usuarios realizan. Es la parte del sistema donde se establecen las reglas de negocio que deben cumplirse. Al recibir una petición del cliente, sus programas son los encargados de comunicarse con la capa de datos para almacenar, actualizar o recuperar información, emitiendo una respuesta.

La capa de acceso a datos

Contiene clases que interactúan con la base de datos, y permiten, utilizando los procedimientos almacenados, generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio. Ella recibe las peticiones de la capa de Negocio.

1.9. Plug-in

La aplicación ofrece servicios de acogida que el plug-in puede utilizar, incluyendo una forma a fin de registrarse en la aplicación host y un protocolo para el intercambio de datos. Los plug-in dependen de los servicios prestados por la aplicación host y no suelen funcionar por sí mismos. Por el contrario, la aplicación host es independiente de estos, lo que lo hace posible para los usuarios finales añadir y actualizarlos dinámicamente sin necesidad de hacer cambios en la aplicación host.

Las interfaces de programación de aplicaciones (API) proporcionan una interfaz estándar, lo que permite a terceros crearlos en interacción con la aplicación host. Las API de terceros plug-in permite la estabilidad para seguir funcionando como la versión original y los cambios para ampliar el ciclo de vida de las aplicaciones obsoletas. Para comprender mejor el mecanismo de Plug-in vea la Figura 1.6.

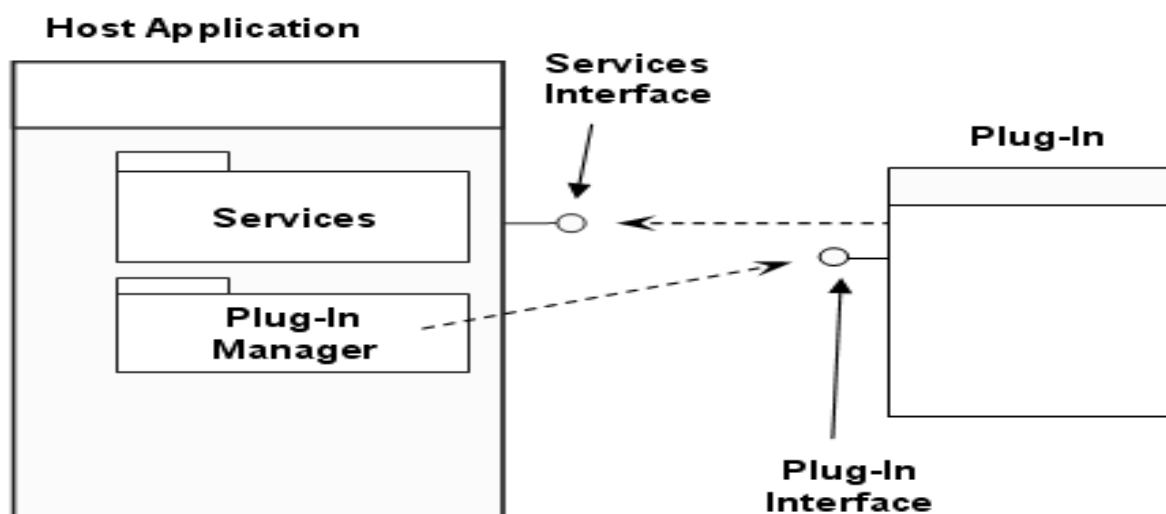


Figura 1.6. Mecanismo de plug-in.

1.10. Herramientas y Metodologías Utilizadas

1.10.1. PostgreSQL 8.2

PostgreSQL 8.2 es un poderoso gestor de base de datos. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una fuerte reputación de fiabilidad, la integridad de los datos, y la corrección.

Funciona en los principales sistemas operativos: incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Es compatible al estándar de Atomicidad, Consistencia, Aislamiento y Durabilidad (ACID)¹⁹. Posee soporte pleno de llaves foráneas, joins, vistas, triggers y funciones en varios lenguajes. Incluye la mayoría de los tipos de datos de SQL92 y SQL99. Así como enteros, numéricos, booleanos, character, varchar, fecha, intervalo, y timestamp. También apoya el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos y vídeo. Posee interfaces de programación nativas para C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otras y muy buena documentación.

¹⁹ Atomicity, Consistency, Isolation and Durability

Capítulo 1

PostgreSQL cuenta con sofisticadas características tales como la versión multi-control de concurrencia (MVCC)²⁰, en el punto en el tiempo de recuperación, de tablas, replicación asincrónica, transacciones anidadas, copias de seguridad, un sofisticado plan de consulta / optimizador, y escribir por delante de registro para la tolerancia de fallas. Soporta juegos de caracteres internacionales, de múltiples codificaciones de caracteres, unicode entre otras. Es altamente escalable, tanto en la cantidad de datos que puede manipular como la cantidad de usuarios concurrentes que soporta. Algunos límites generales se pueden apreciar en la siguiente tabla: (24)

Límite	Valor
Tamaño Máximo de la Base de Datos	Ilimitado
Tamaño Máximo de las Tablas	32 TB
Tamaño Máximo de las Filas	1.6 TB
Tamaño Máximo de un campo (atributo)	1 GB

Tabla 1.3 Límites del gestor de base datos PostgreSQL.

1.10.2. RUP (25)

RUP²¹ es un proceso para el desarrollo de un proyecto de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Se dirige por los Casos de Uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere. Está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden. Es iterativo e incremental, divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

²⁰ Multi-version Concurrency Control

²¹ Rational Unified Process

Capítulo 1

Como filosofía RUP maneja 6 principios claves:

1. Adaptación del proceso.

El proceso deberá adaptarse a las características propias de la organización. El tamaño del mismo, así como las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

2. Balancear prioridades.

Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.

3. Colaboración entre equipos.

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes y resultados.

4. Demostrar valor iterativamente.

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

5. Elevar el nivel de abstracción.

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o esquemas (frameworks) por nombrar algunos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML²².

6. Enfocarse en la calidad.

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

²² Unified Modeling Language

Capítulo 1

1.10.3. Microsoft Visual Studio Team System 2008

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Su última versión, la 2008 fue publicada el 17 de Noviembre de 2007 en inglés, mientras que la versión en español se dio a conocer el 2 de Febrero de 2008.

El nuevo frameworks (.Net 3.5) está diseñado para aprovechar las ventajas que ofrece sistema operativo *Windows Vista* a través de sus subsistemas (WCF)²³ y (WPF)²⁴. Estos tienen como objetivo la construcción de aplicaciones orientadas a servicios y la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento respectivamente.

1.10.4. Microsoft Visual Source Safe 2005

El Sistema de Control de Versiones es sencillo y fácil de usar. Está caracterizado por la integración con Visual Studio 2005 y 2008; simplifica en gran medida el uso de este tipo de herramientas. Resulta invaluable para el desarrollo en equipo, porque registra un historial de los cambios en cada fichero. Además, brinda la posibilidad de codificar de forma concurrente el mismo proyecto e incluso el mismo fichero. También posibilita la administración de los ficheros de código fuente. Adicionalmente soporta archivos XML, acceso remoto sobre HTTP, compatibilidad con versiones anteriores, y mayor capacidad llegando hasta 4Gb. (26)

²³ Windows Communication Foundation

²⁴ Windows Presentation Foundation

Capítulo 1

1.10.5. Enterprise Architect

Enterprise Architect 7.0 es una herramienta gráfica multi-usuario, basada en Windows y orientada a objetos. Es competitiva para el desarrollo de software, administración de proyecto, administración de requerimientos y análisis de negocio. Resulta muy usada para el diseño robusto y confiable de software guiado por computadora. Es portadora de buena flexibilidad y documentación de alta calidad.

Combina el poder de la última especificación UML 2.1 con un alto rendimiento e interfaz intuitiva. Se integra perfectamente a la plataforma .NET y permite hacer uso de técnicas de ingeniería inversa; así como de generación automática de documentación, para extender el dominio de modelado mientras que la validación de modelos asegura buena integridad.

Otras características importantes son: usa perfiles UML, capacidad de Importación-Exportación XML 2.1, motores de Reporte HTML, transformaciones MDA, rastreo de recursos y mantenimiento. (27)

1.10.6. El lenguaje C#

C# es un lenguaje de programación que combina las potencialidades de C, C++ y Java con la agilidad y la velocidad para el desarrollo propios de Visual Basic. Sin dudas, es el lenguaje más importante de la plataforma .NET. Fue diseñado para crear sistemas en el mundo actual, que puedan comunicarse con otras aplicaciones y servicios mediante una red de información, donde conviven bases de datos de distintos fabricantes, los sistemas son construidos a partir de componentes locales o remotos, y por tanto requieren soluciones integrales para diversos tipos de plataformas. Con C# se pueden crear sistemas de una gran diversidad: simples o complejos; aplicaciones de consola, de escritorio o para la Web; programas para computadoras personales o para dispositivos móviles.

1.11. Justificación de las herramientas y tecnologías utilizadas

Las herramientas y metodologías que han sido caracterizadas, están entre las más usadas a nivel mundial. Sus potencialidades, prestaciones y fácil manejo son sus principales características. Por esto, tienen gran aceptación entre los desarrolladores de software. En consecuencia, los autores del presente trabajo consideran una necesidad utilizarlas en la elaboración de la aplicación.

1.12. Importancia de un Servidor de Imágenes en un sistema de control de vehículos

El uso de un sistema de monitoreo de imágenes en una solución de control de vehículos contribuye con la organización de las imágenes, la distribución, el aprovechamiento del espacio en disco y la correcta actualización de la base de datos. Las imágenes se organizan de forma que al tener que ser borradas o copiadas a otro servidor se comienza por las que más tiempo lleven en el disco duro. Da una mayor robustez al sistema en caso de que el otro servidor al que fueron copiadas las imágenes no se encuentre disponible. Ocurre lo mismo en caso de elegir la opción de borrar las imágenes.

Al subir o bajar imágenes existe la posibilidad de usar gran cantidad de recursos de la computadora y como consecuencia a esto provocar que esta no responda con la rapidez necesaria. Utilizando un servidor se puede verificar a simple vista el uso de los recursos de la computadora. Se puede prever si aún esta podría seguir respondiendo acorde a las necesidades en caso de ejecutar una acción que involucre grandes volúmenes de imágenes. Desde el punto de vista de su despliegue, significa tener una constancia de que un vehículo pasó por un punto de control ya que es la prueba más confiable que se tiene de probar, de existir algún hecho delictivo, que un determinado auto paso por cierto lugar a cierta hora.

En este capítulo se valoraron las principales tendencias en cuanto a los protocolos de transmisión de archivos; entre ellos el protocolo FTP y sus posibles servidores a utilizar. Así como, las principales compañías desarrolladoras de software para la identificación y control de vehículos, y sus productos representativos. Se analizaron las posibles arquitecturas de software a utilizar, optando por la arquitectura en 3 capas. También, fueron determinadas las herramientas, metodologías y tecnologías para la realización del sistema propuesto.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El desarrollo de Cugnot Server impone un reto tecnológico. Los procesos principales de la aplicación deben ser ejecutados de forma automática. Esto implica un trabajo complejo desde el punto de vista de la programación. Los registros de estado y eventos constituyen también una tarea importante; principalmente para que el usuario comprenda y valore el funcionamiento del servidor. Además, garantizar que sólo los administradores del sistema tengan privilegios de configuración así como de otras tareas que afecten directamente el funcionamiento del sistema.

Para la realización de esta solución se realiza un proceso exhaustivo de ingeniería de software (28). Durante el desarrollo del presente capítulo y los restantes se verá todo el proceso al que se ha sometido el sistema planteado. La ingeniería de software comienza con la fase de inicio y dentro de ella el flujo de modelado del negocio y levantamiento de requisitos. Ambos flujos de trabajo serán detallados en este capítulo. El capítulo define además los principales requisitos y características que debe cumplir el sistema y una breve descripción de cada una, así como se muestran los elementos que intervienen en la solución del problema y la relación entre ellos. Además de definir los actores y los casos de uso; se justifican los actores como futuros usuarios y se da una descripción detallada de los casos de uso.

2.1. Descripción del negocio

La identificación de vehículos en Cuba generalmente se realiza de forma manual. Cuando un vehículo arriba a un punto de control, el custodio (responsable de seguridad) verifica que se le ha concedido permiso para proseguir la marcha. Para ello, el custodio localiza la matrícula en el registro de vehículos autorizados. Si los caracteres de la matrícula son encontrados, el custodio autoriza el paso del vehículo y actualiza el registro. Si la matrícula no es encontrada, el custodio deniega el acceso y el vehículo debe abandonar el punto de control.

Capítulo 2.

2.2. Modelo del Negocio

2.2.1. Actores del Negocio

Los actores del negocio son presentados y justificados en la tabla:

Actores del negocio	Justificación
Vehículo	Representa el inicio de todas las actividades del negocio

Tabla2.1 Actores del Negocio.

2.2.2. Trabajadores del negocio

Trabajadores del negocio	Justificación
Custodio	Responsable de darle paso al vehículo

Tabla 2.2 Trabajadores del Negocio.

Diagrama de casos de uso del negocio

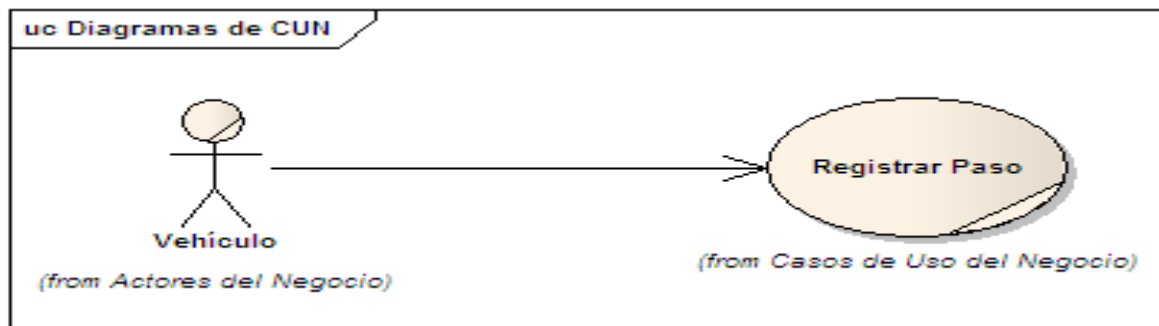


Figura 2.1 Diagrama de casos de uso del negocio.

Capítulo 2.

2.2.3. Diagrama de actividades: CUN Registrar Paso:

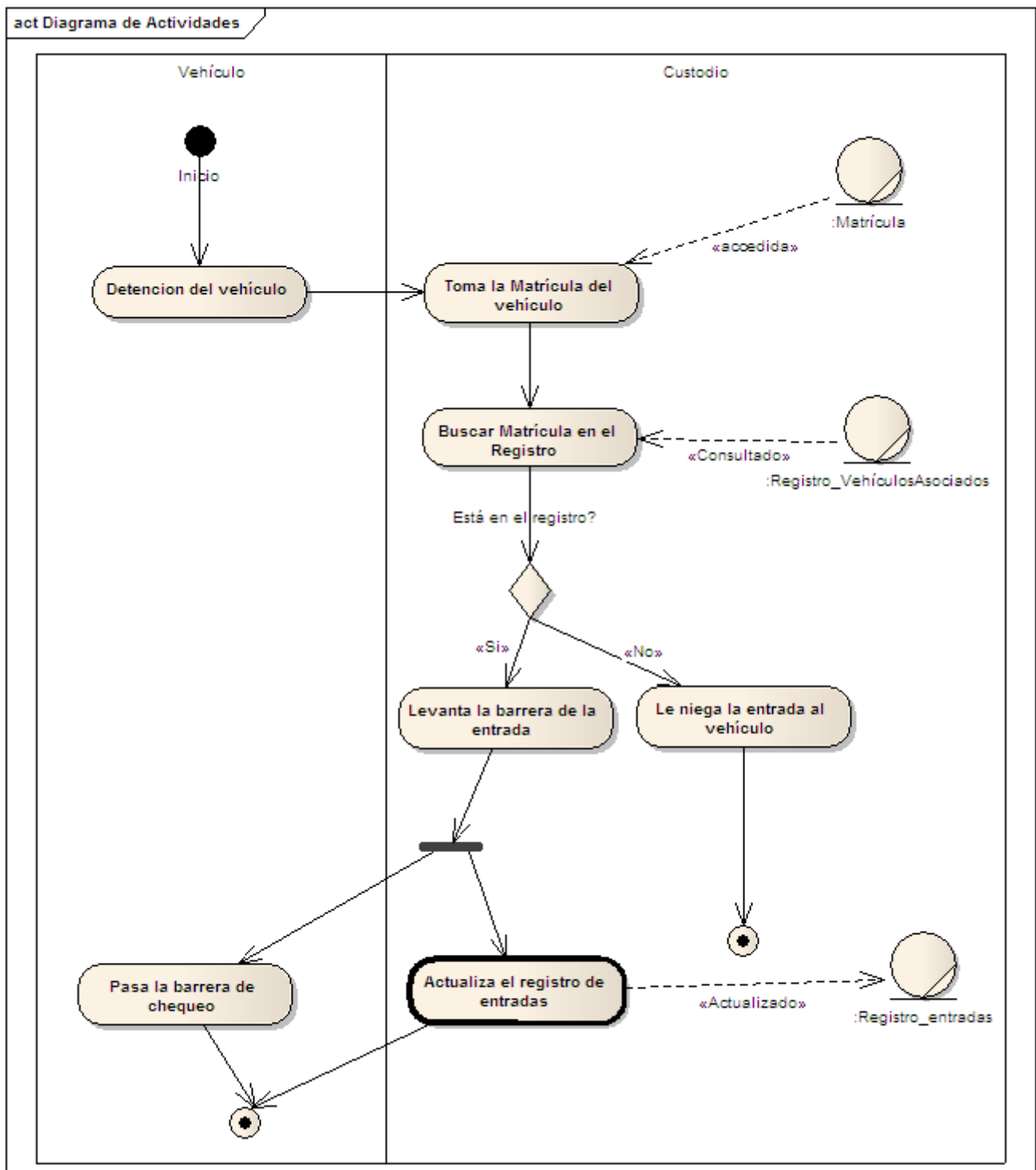


Figura 2.2 Diagrama de actividades CUN Registrar Paso.

Capítulo 2.

2.2.4. Modelo de Objetos del Negocio

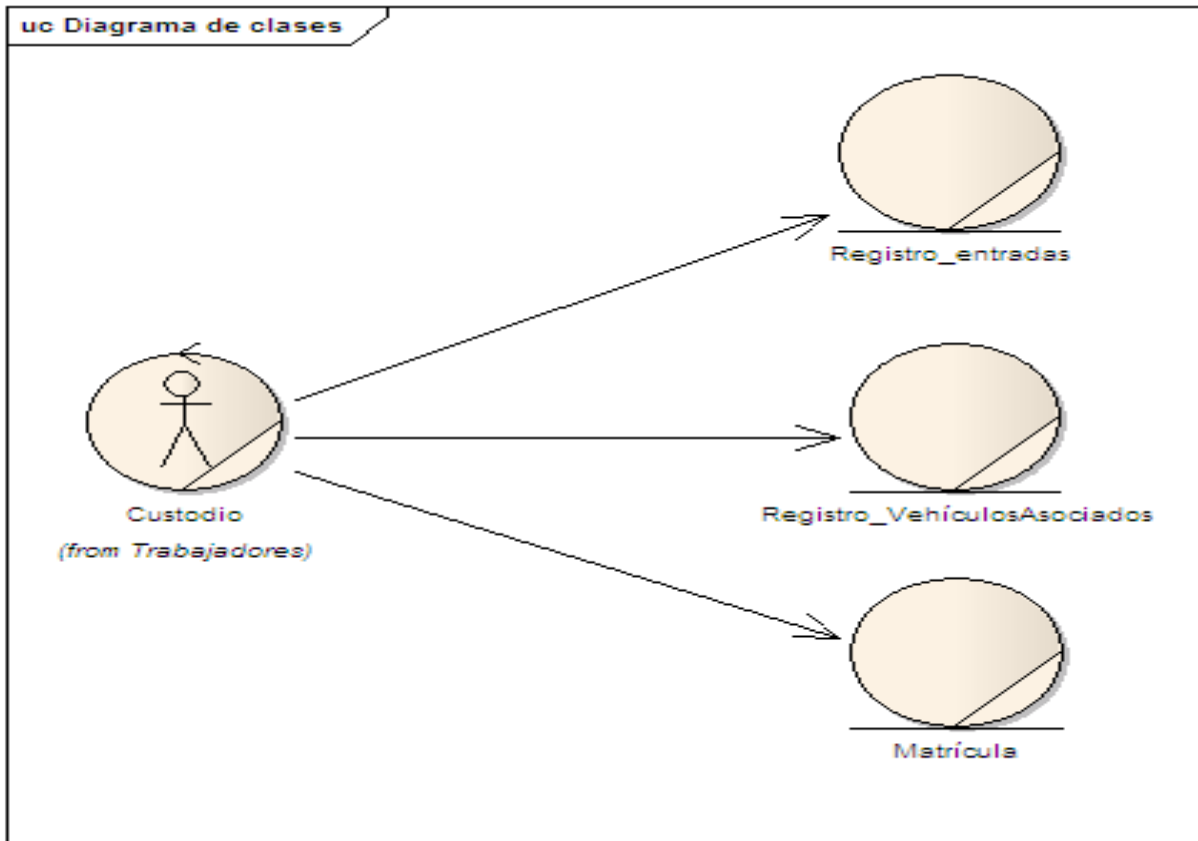


Figura 2.3 Modelo de Objetos del Negocio.

2.3. Propuesta de sistema

El presente trabajo de diploma propone desarrollar la solución informática (Cugnot Server) con el objetivo monitorear las particiones de la computadora. Este sistema sustituirá la forma tradicional de guardar la prueba de que un automóvil cruzó por un punto de control por una más efectiva, con menos posibilidad de errores y más utilidades.

Cugnot Server puede ser configurado, para mover imágenes hacia otro servidor o borrarlas, en caso de crisis de espacio en la partición de la computadora. Además, el sistema concibe que el usuario no tenga que autenticarse en la computadora para que el sistema pueda funcionar una vez que este haya sido configurado. Para esto, la aplicación usa un servicio que se inicia junto al sistema operativo y que no es posible cerrar sin apagar la computadora.

Junto al sistema funcionará un servidor FTP. Cuya función será atender las solicitudes de los usuarios y las demás aplicaciones que envían las imágenes hacia él. A su vez aprovecha las prestaciones de estos servidores para subir, listar, bajar y borrar imágenes. La propuesta desarrollada combina la aplicación de Cugnot Server con un servidor FTP para lograr dar respuesta efectiva al problema planteado.

Como todas las actividades quedan registradas en un log, en caso de interrupción a causa de un fallo eléctrico o de red, estas se pueden reanudar al normalizarse las condiciones. Así mismo pasa con la actualización de la base de datos central. En caso de algún problema, se mantiene una copia de la base de datos central a nivel local. Al ser restablecidas las condiciones, todo puede retornar a la normalidad con una mínima afectación.

Capítulo 2.

2.4. Requisitos del sistema

2.4.1. Requisitos Funcionales

Identificador	Descripción
RF1 ²⁵ .	El sistema debe permitir configurar un tipo de reacción predeterminada ante una crisis de espacio libre en la partición.
RF2	El sistema debe ejecutar automáticamente la reacción configurada ante la crisis de espacio libre en la partición.
RF3	El sistema debe reportar hacia el registro de sucesos los cambios en su funcionamiento.
RF4	El sistema debe mostrar el estado del servidor.

Tabla 2.3 Requisitos Funcionales.

2.4.2. Requisitos no Funcionales

A continuación se listan los requerimientos no funcionales por sus correspondientes categorías:

Requerimientos de implementación.

Identificador	Descripción
RNF1 ²⁶	El sistema debe ejecutarse sin necesidad de que exista algún usuario autenticado en el equipo.

Tabla 2.4 Requerimientos de Implementación.

²⁵ Requisitos Funcionales.

²⁶ Requisitos no Funcionales.

Capítulo 2.

Requerimientos de portabilidad.

Identificador	Descripción
RNF2	EL sistema debe ser compatible con la plataforma libre MONO.

Tabla 2.5 Requerimientos de Portabilidad.

Requerimientos de Software.

Identificador	Descripción
RNF3	NET Framework 2.0 o superior.

Tabla 2.6 Requerimientos de Software.

Requerimientos de Usabilidad.

Identificador	Descripción
RNF4	Facilidad de configuración del servidor.

Tabla 2.7 Requerimientos de Usabilidad.

Requerimientos de Hardware.

Identificador	Descripción
RNF5	512mb de memoria RAM (mínimo).
RNF6	CPU Pentium IV 2.8GHz (recomendado).

Tabla 2.8 Requerimientos de Hardware.

Capítulo 2.

2.5. Casos de Uso del sistema

2.5.1. Actores del sistema.

Actores	Justificación
Timer	Encargado de invocar el caso de uso consultar estado de la partición disco duro. Es no configurable desde el usuario externo.
Administrador	Es el encargado de configurar al sistema, puede ser el jefe de vigilancia o alguien directamente responsable de la seguridad.

Tabla 2.9 Actores del sistema.

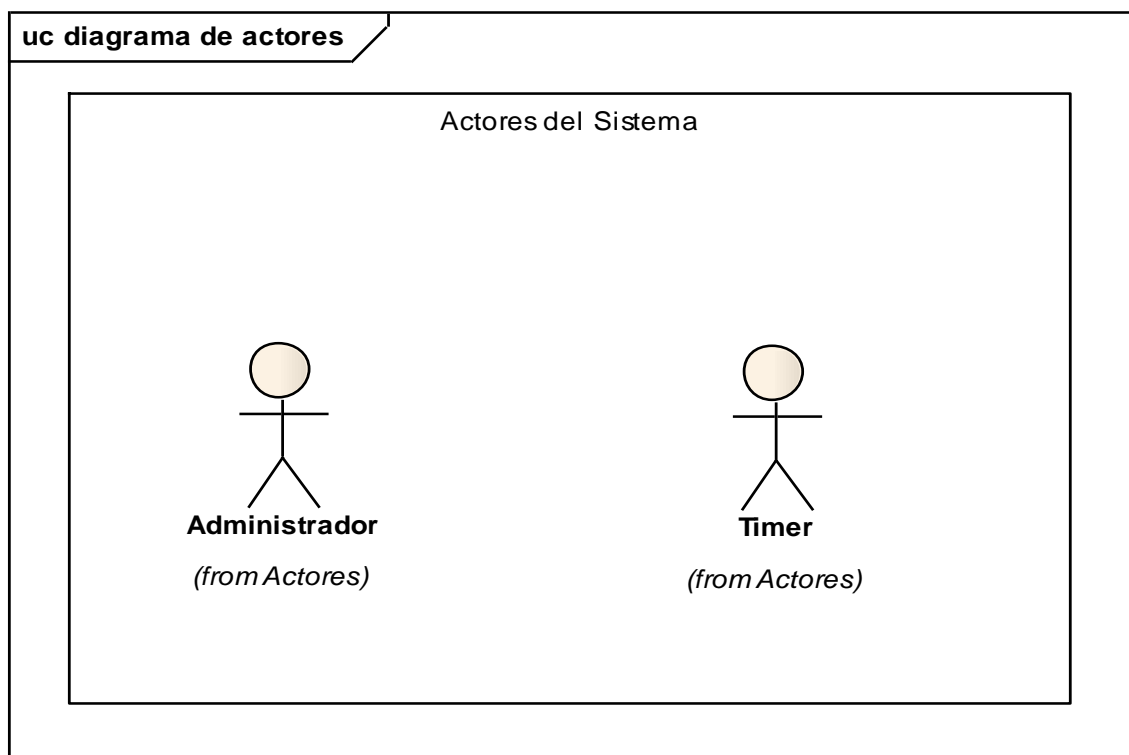


Figura 2.4 Actores del sistema.

2.5.2. Casos de Uso del Sistema

CU-1	Configurar Servidor.
Actor	Administrador.
Descripción	Es configurado el servidor para caso de reacción ante crisis de espacio en disco.
Referencia	RF1; RNF4.

Tabla 2.1 Casos de Uso del Sistema (Configurar Servidor).

CU-2	Consultar Estado Partición.
Actor	Timer
Descripción	Este caso de uso es invocado cada cierto tiempo con el objetivo de obtener datos del volumen donde se encuentra el repositorio de imágenes.
Referencia	RF2; RF4; RNF1.

Tabla 2.2 Casos de Uso del Sistema (Consultar Estado Partición).

CU-3	Liberar Espacio.
Actor	Consultar Estado Partición.
Descripción	Comienza a borrar imágenes o a moverlas hacia otro lugar en dependencia de la configuración que haya elegido el usuario.
Referencia	RF2; RF3; RF4; RNF1; RNF5; RNF6; RNF7.

Tabla 3.3 Casos de Uso del Sistema (Liberar Espacio).

2.5.3. Diagrama de Casos de Uso

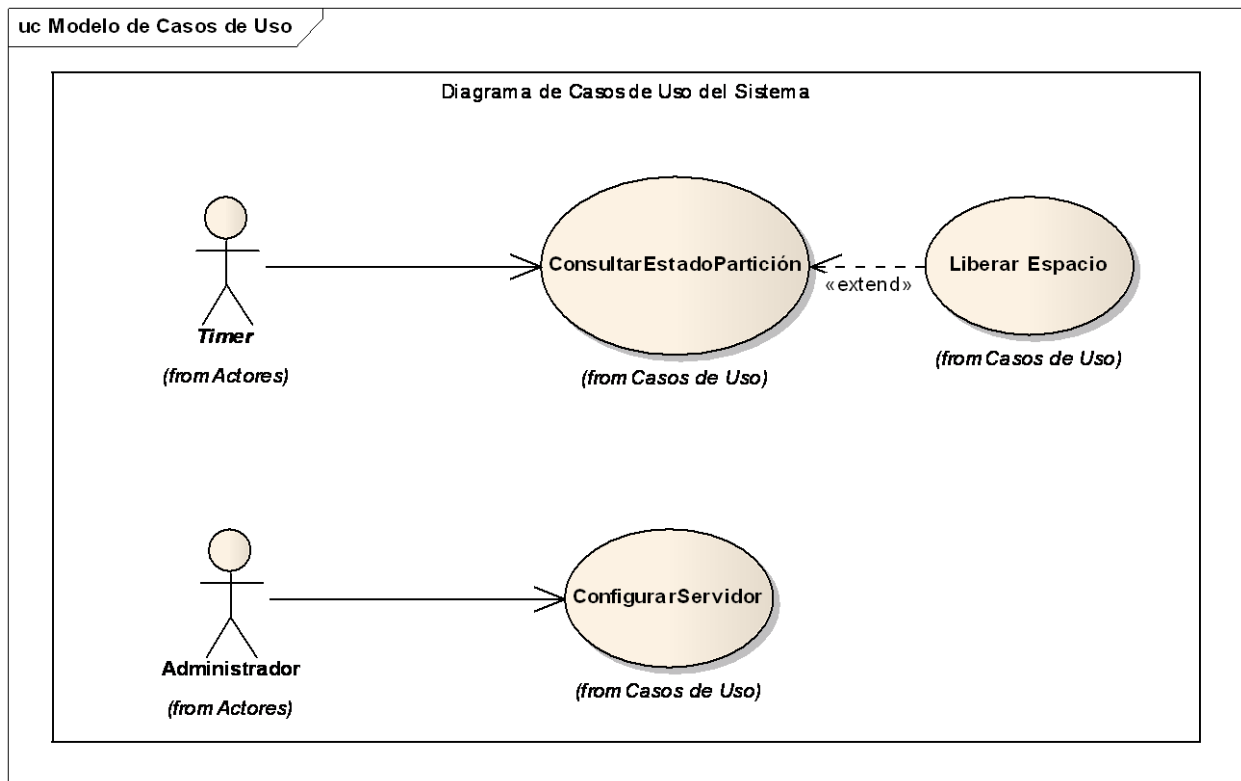


Figura 2.5 Diagramas de Casos de Uso.

Capítulo 2.

2.5.4. Casos de Uso por Ciclos de desarrollo

Ciclo1:

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-2	Consultar Estado Partición.	Monitoreo de Imágenes.	Caso de uso necesario y base del funcionamiento del servidor.
CU-3	Liberar Espacio.	Monitoreo de Imágenes.	Caso de uso que asegura la efectividad del servidor. Hito principal de la aplicación.

Tabla 3.1 Casos de Uso por Ciclos de desarrollo (Ciclo 1).

Ciclo 2:

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Configurar Servidor.	Configuración.	Caso de uso necesario para que la aplicación tenga un desempeño deseado por el usuario. Importante para el CU-3 el flujo que este debe seguir.

Tabla 3.2 Casos de Uso por Ciclos de desarrollo (Ciclo 2).

Capítulo 2.

2.5.5. Expansión de los casos de uso del sistema

Caso de uso	
CU-1	Configurar Servidor.
Propósito	Es necesario configurar el servidor para que este tenga un buen funcionamiento. El objetivo principal es configurar el tipo de reacción ante crisis de espacio en disco.
Actores	Administrador (Inicia).
Resumen: El administrador configura el servidor para que este reaccione según parámetros definidos por él.	
Referencias	RF1; RF3.
Acción del actor	Respuesta del sistema
1 - El administrador elije la opción configurar servidor. 3- El administrador llena los datos necesarios para realizar las pruebas de funcionamiento, presiona el botón actualizar configuración. 8- Cierra la aplicación.	2- Muestra la pantalla de configuración. 4- Valida los datos. 5- Hace test de funcionamiento. 6- Emite mensaje de validez de los datos. 7- Salva los datos de configuración en un fichero (XML).

Tabla 3.1 Expansión de los casos de uso del sistema (**Caso de uso:** Configurar Servidor).

Capítulo 2.

Caso de uso	
CU-2	Consultar Estado Partición.
Propósito	Consultar cada cierto período de tiempo el estado de la partición donde se encuentra el repositorio de imágenes.
Actores	Timer (Inicia).
Resumen:	El timer invoca el CU para obtener datos del repositorio de imágenes. En cada invocación actualiza los datos de estado y en caso de crisis de espacio invoca al CU-3.
Referencias	RF4; RNF1 Extiende el CU-3.
Acción del actor	Respuesta del sistema
1- El timer invoca el método que inicia la búsqueda de datos referentes a la partición donde está el repositorio de imágenes.	2- Actualiza los valores anteriores con los valores actuales. 3- Escribe en el fichero de estado actual los nuevos valores. 4- Si los valores actuales son mayores que los límites permitidos en la configuración del usuario. Invoca el CU-3
Puntos de extensión.	
Línea 5 ver CU-3.	

Tabla 3.2 Expansión de los casos de uso del sistema (**Caso de uso:** Consultar Estado Partición).

Capítulo 2.

Caso de uso	
CU-3	Liberar Espacio.
Propósito	Garantizar una correcta organización y distribución de las imágenes.
Actores	CU-2 (Inicia).
Resumen: Cuando la partición se encuentra ante crisis de espacio se invoca este caso de uso para borrar o copiar hacia otro servidor imágenes.	
Referencias	Es extendido del CU-2.
Acción del actor	Respuesta del sistema
1- Desde el CU-2 el sistema detecta una alteración en el por ciento de espacio libre en la partición.	2- El sistema busca las imágenes que más tiempo lleven. 3- El sistema borra las imágenes hasta que los valores reales sean iguales que los configurados por el usuario, actualiza la base de datos.
Flujo alternativo.	Mover hacia otro servidor
Acción del actor	Respuesta del sistema
1- Desde el CU-2 el sistema detecta una alteración en el por ciento de espacio libre en la partición.	2- El sistema busca todas las imágenes que lleven más tiempo. 3- Comienza a mover las imágenes hacia otro servidor hasta que los valores reales sean iguales que los configurados por el usuario, actualiza la Base de datos.
Puntos de extensión.	
En la línea 1 Ver CU-2.	

Tabla 3.3 Expansión de los casos de uso del sistema (**Caso de uso:** Liberar Espacio).

Capítulo 2.

Al concluir este capítulo se han determinado las características y cualidades que debe tener el sistema, lo que fue posible al realizar el levantamiento de requisitos. Fueron descritos los procesos del negocio y los actores que intervienen en la realización del sistema. Así como las funciones e interacción de estos con la aplicación. Finalmente, se describen los casos de uso que participan en la elaboración del software tanto del negocio como del sistema.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Una vez concluida la fase de inicio es posible diseñar el sistema propuesto, comenzando la segunda etapa: Fase de Elaboración y dentro de esta el flujo de Análisis y Diseño. Este flujo de trabajo es el centro de atención de la fase de elaboración e inicio. Su objetivo es transformar los requisitos en el diseño del futuro sistema y contribuir a una arquitectura estable y sólida. Así como, a lograr un diseño robusto de la base de datos y un plano del modelo de implementación.

3.1. Modelo de Análisis

El modelo de Análisis es usado para representar la estructura global del sistema, describe la realización de casos de uso, sirve como una abstracción del Modelo de Diseño y se centra en los requerimientos no funcionales. Puede contener: las clases y paquetes de análisis, las realizaciones de los casos de uso, las relaciones y los diagramas. Es opcional detallar las realizaciones de los casos de uso pues estas pueden estar en el modelo de diseño donde se recomienda que se encuentre. A diferencia del Modelo de Casos de Uso, da forma a la arquitectura para soportar las funcionalidades. Este modelo netamente ingenieril. (29)

En el análisis se debe:

- Conseguir una comprensión más precisa de los requisitos, refinarlos y estructurarlos.
- Utilizar el lenguaje de los desarrolladores para analizar con profundidad los requisitos funcionales.
- Proporcionar una visión general del sistema.

El modelo de análisis de Cugnot Server es un primer intento de definir los conceptos claves que describen el sistema; no es un diagrama final de todos los posibles conceptos y sus relaciones. Su utilidad radica en que permite una apreciación global conceptual del sistema.

3.2. Diagramas de Clases del Análisis

3.2.1. Diagrama de Clases del Análisis: CU (Monitorear Partición)

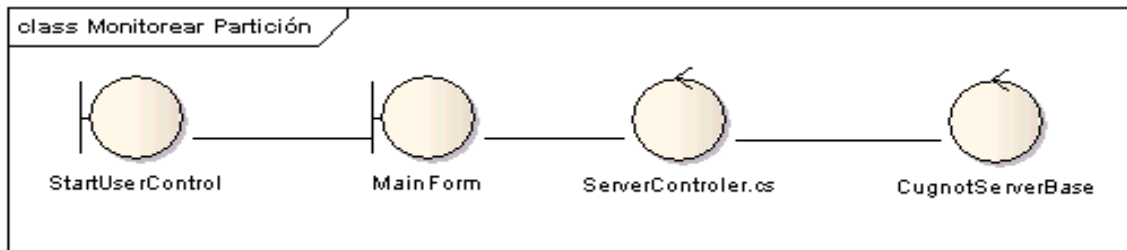


Figura 3.1 Diagrama de Clases del Análisis: CU (Monitorear Partición).

3.2.2. Diagrama de Clases del Análisis: CU (Configurar Servidor)

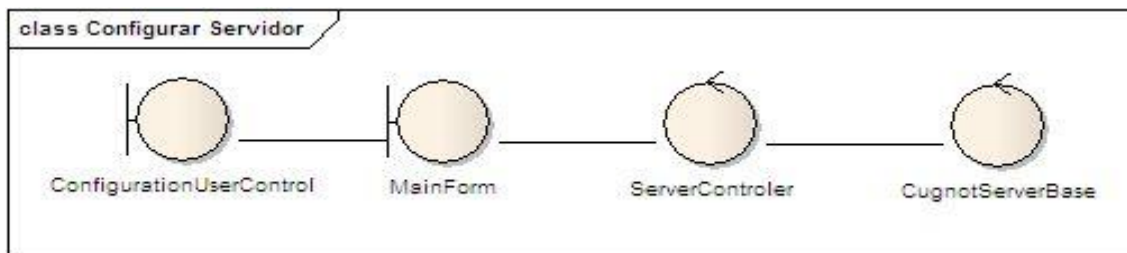


Figura 3.2 Diagrama de Clases del Análisis: CU (Configurar Servidor).

3.2.3. Diagrama de Clases del Análisis: CU (Liberar Espacio)

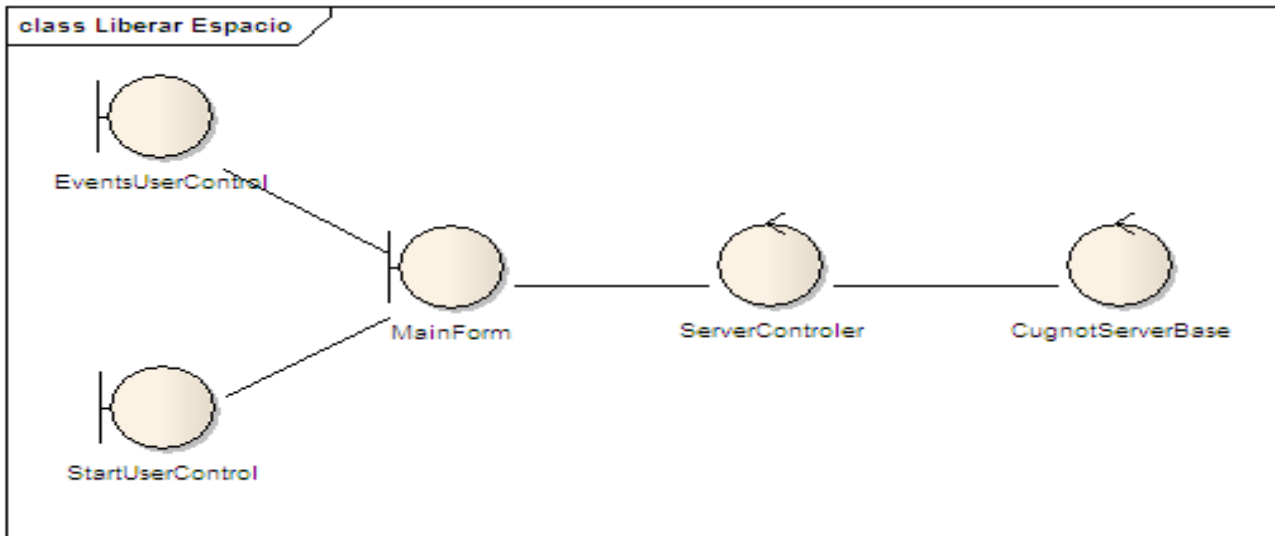


Figura 3.3 Diagrama de Clases del Análisis: CU (Liberar Espacio).

3.3. Modelo de diseño

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Este contribuye a una arquitectura estable y sólida; y a crear un plano del modelo de implementación. Durante la fase de construcción, cuando se ha logrado una arquitectura estable y los requisitos están bien entendidos el centro de atención se desplaza a la implementación.

En el diseño se modela el sistema y se encuentra la forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que suponen. Además, impone una estructura del sistema que debe ser conservada lo más fielmente posible. Una entrada esencial en el diseño es el resultado del análisis.

3.3.1. Arquitectura Utilizada

La solución Cugnot Server presenta una arquitectura en tres capas: en la **(Figura 3.7 Clases del Diseño)** se muestra un diagrama de clases organizado por capas. Esta, está definida con el objetivo de poder añadir nuevas funcionalidades sin necesidad de ser cambiada. Para ello, define que todas las relaciones entre clases deben ser mediante interfaces; lo cual posibilita que los cambios en la implementación de las funcionalidades no afecten la concepción inicial del sistema.

3.3.2. Formas de extensión

Cugnot Server ha sido diseñado como un núcleo de características básicas con un mecanismo de extensión que permite extender ese núcleo basado en el uso de plug-in.

La aplicación puede ser extendida en varios aspectos dando lugar a tres categorías de extensiones:

- 1- Relacionada con la forma de obtención de espacio del disco duro.
- 2- Relacionada con la forma de actualización de la base da datos.
- 3- Relacionadas con la alternativa a seguir para liberar espacio.

Los mecanismos utilizados para monitoreo de imágenes, liberar espacio en la partición y actualización de la base de datos; permiten identificar algunas operaciones que podrían modificar el comportamiento de la aplicación. Si se decidiera cambiar de sistema operativo es posible añadir una nueva forma de obtención de espacio en el disco. Así como, ante un cambio de gestor de base de datos es posible añadir la forma de actualización compatible con el nuevo gestor utilizado.

Por su parte, la alternativa para liberar espacio puede ser conveniente para algunos lugares y para otros no, dependiendo de la importancia que pudiera revestir el tener almacenadas o no las imágenes. Por eso, aunque Cugnot Server define hoy dos formas para liberar espacio, cuando su uso no sea conveniente, le puede ser añadida otra(s) forma(s) de liberar espacio.

3.4. Diagramas de Interacción del diseño

3.4.1. Diagrama de Secuencia del diseño: CU (Monitorear Partición)

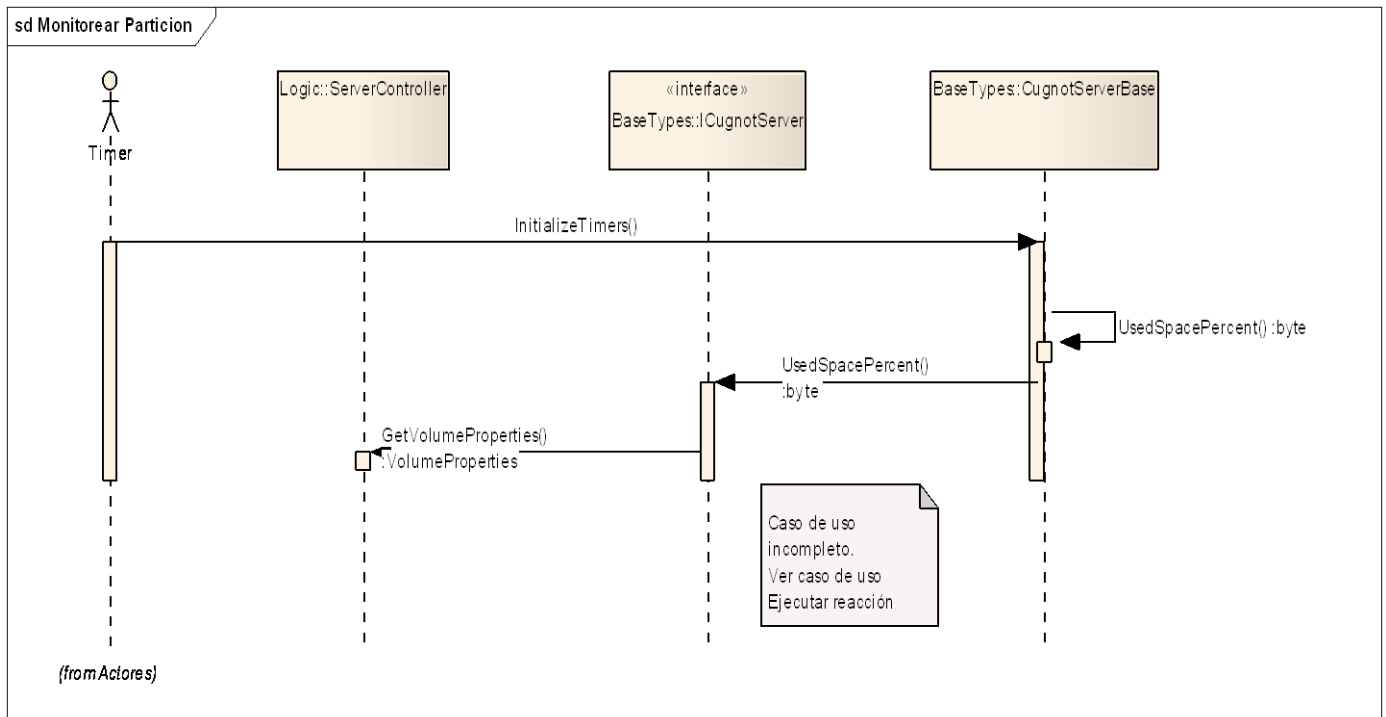


Figura 3.4 Diagrama de Secuencia del diseño: CU (Monitorear Partición).

3.4.2. Diagrama de Secuencia del diseño: CU (Configurar Servidor)

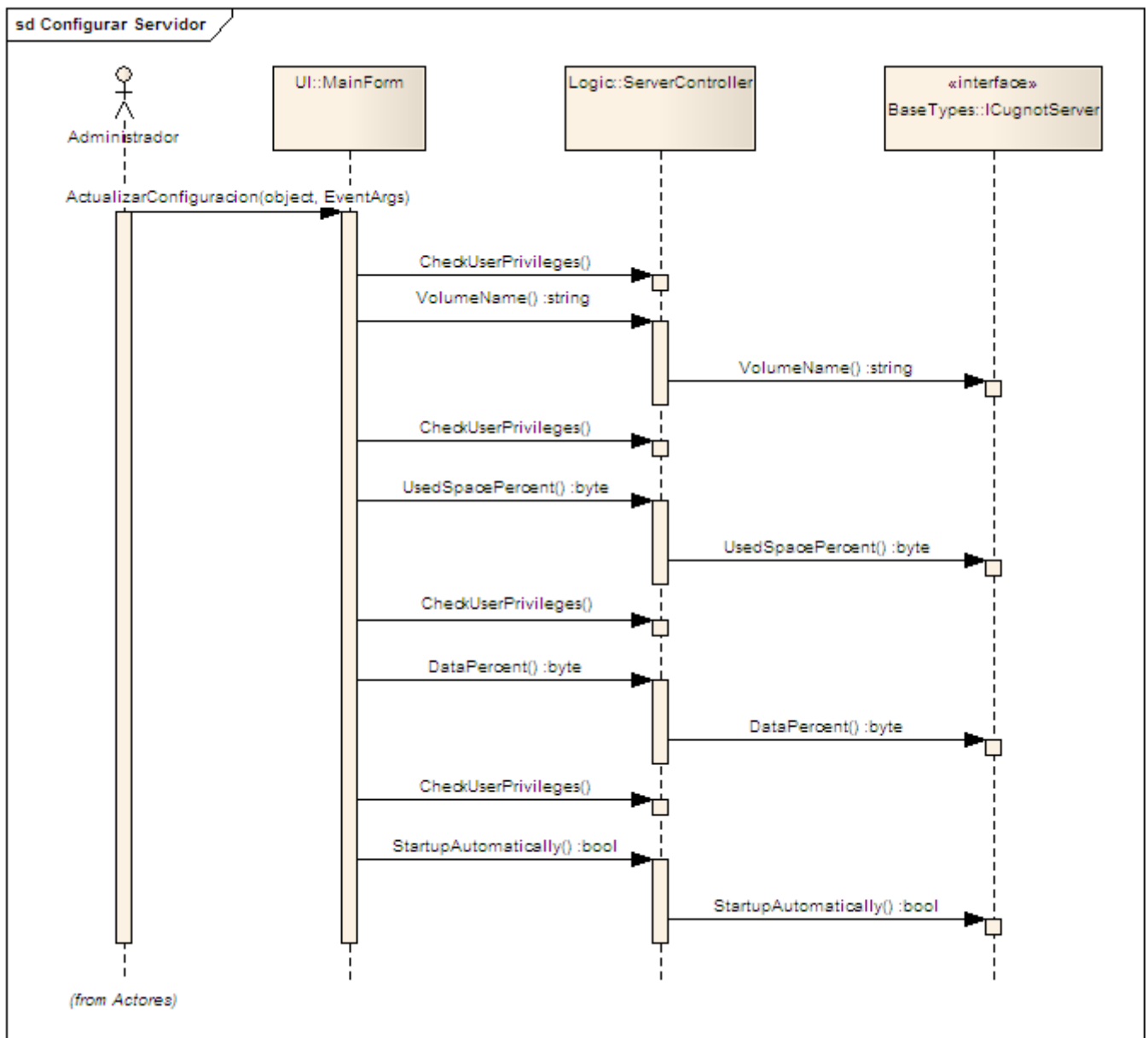


Figura 3.5 Diagrama de Secuencia del diseño: CU (Configurar Servidor).

3.4.3. Diagrama de Secuencia del diseño: CU (Liberar Espacio)

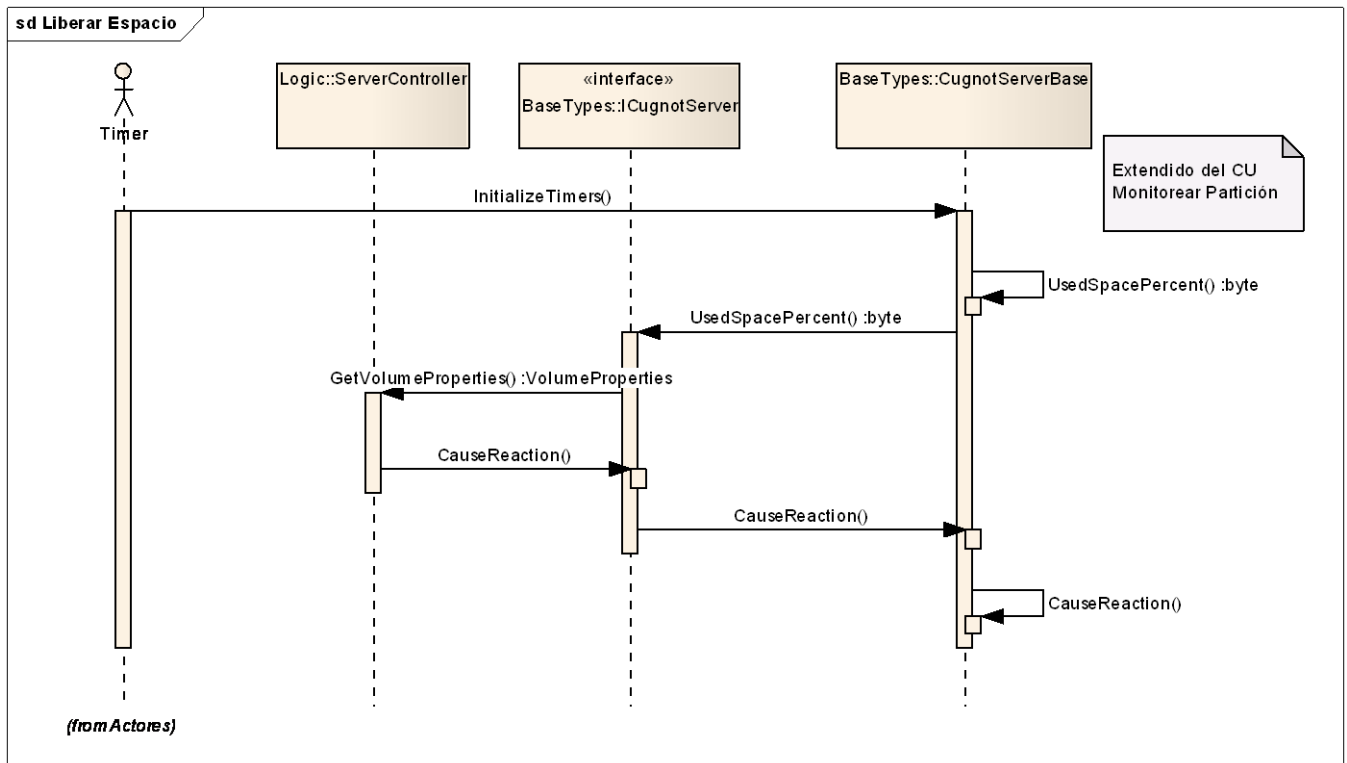


Figura 3.6 Diagrama de Secuencia del diseño: CU (Liberar Espacio).

3.5. Diagrama Clases del Diseño

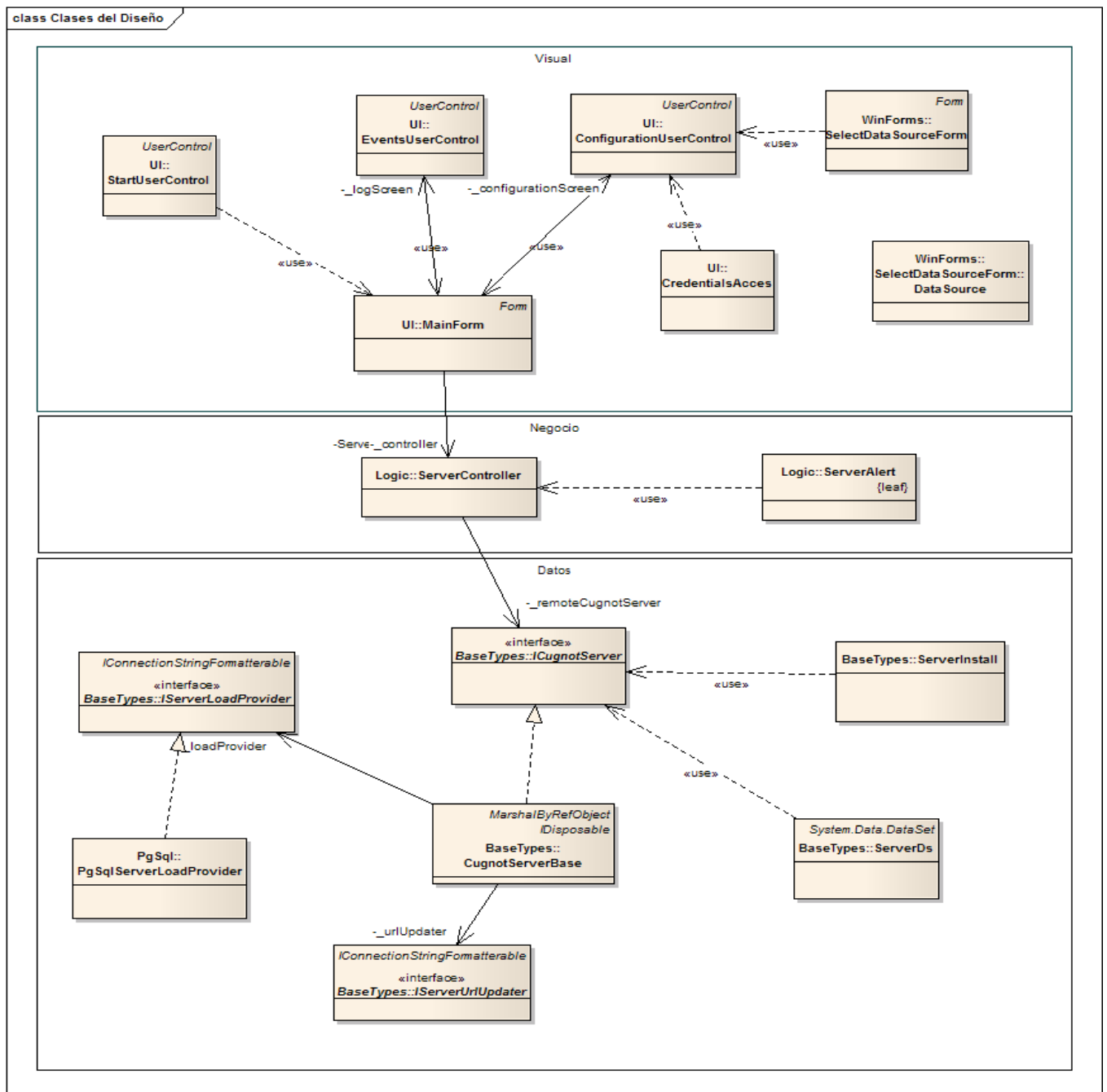


Figura 3.7 Clases del Diseño.

Cugnot Server: Servidor de Imágenes de vehículos

Capítulo 3.

3.5.1. Diagramas de Clases del diseño: CU Monitorear Partición

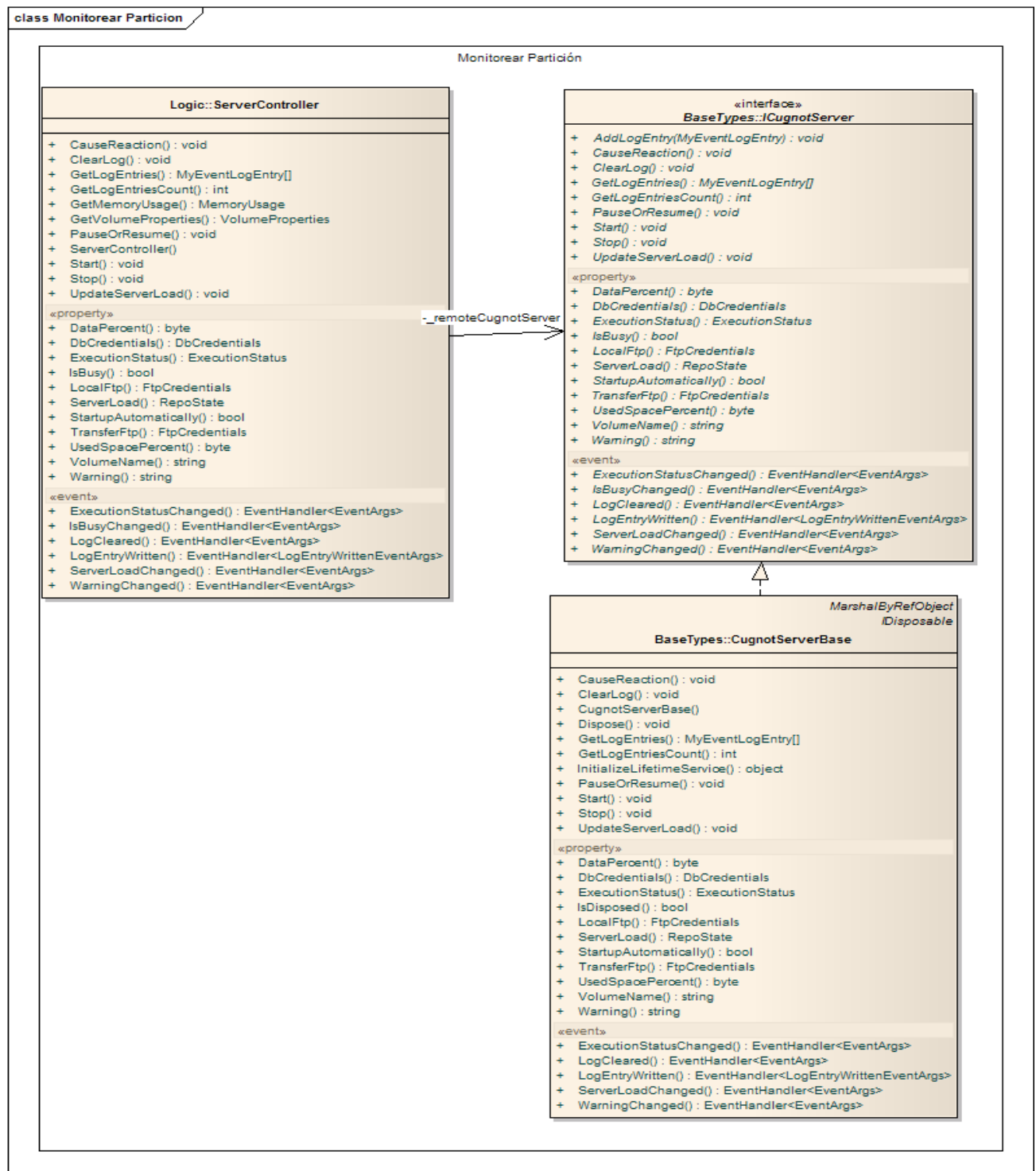


Figura 3.8 Diagrama de Clases: CU (Monitorear Partición).

Cugnot Server: Servidor de Imágenes de vehículos

Capítulo 3.

3.5.2. Diagrama de Clases: CU (Configurar Servidor)



Figura 3.9 Diagrama de Clases: CU (Configurar Servidor).

3.5.3. Diagrama de Clases CU (Liberar Espacio)

El diagrama de clases del diseño de este CU es el mismo que el perteneciente al CU Monitorear Partición. Debido a que el CU **Liberar Espacio** es extendido del caso de uso subrayado y su realización sólo implica la invocación de más funcionalidades en las mismas clases.

3.6. Descripción de las clases

Para entender mejor los detalles de las clases interfaces se pueden consultar las mismas con más profundidad. Ver ANEXO 1: DESCRIPCIÓN DE LAS CLASES INTERFACES, en las tablas:

A.1.1 Descripción de la clase [MainForm](#).

A.1.2 Descripción de la clase [ConfigurationUserControl](#).

A.1.3 Descripción de la clase [EstablishCredentialsForm](#).

A.1.4 Descripción de la clase [EventsUserControl](#).

A.1.5 Descripción de la clase [ProductAboutUserControl](#).

A.1.6 Descripción de la clase [StartUserControl](#).

Para consultar los detalles de las clases controladoras Ver ANEXO 2: DESCRIPCIÓN DE LAS CLASES CONTROLADORAS, en las tablas:

A.2.1 Descripción de la clase [ServerController](#).

En el capítulo se desarrolló el flujo de Análisis y Diseño, uno de los más importantes durante el ciclo de vida de desarrollo de un software. Este necesita la comunicación entre el cliente y el grupo de trabajo de programadores. Como resultado se obtuvo una arquitectura candidata basada en la realización de casos de uso y centrada en los requerimientos funcionales y no funcionales. Con su desarrollo se crearon las clases y relaciones que posteriormente serán implementadas.

CAPÍTULO 4: IMPLEMENTACIÓN

Antes de implementar cualquier método o clase, el software pasa por fases importantes durante su ciclo de desarrollo. Etapas como la fase de inicio, donde se obtiene el modelo de negocio y el levantamiento de requisitos; y en el análisis y diseño donde se proporciona un alto nivel de refinamiento del producto. Gracias a todo esto el software queda arquitectónicamente listo para pasar al siguiente nivel: la fase de construcción (30)

El propósito principal de la implementación es desarrollar la arquitectura y el sistema como un todo. También se necesita cumplir otros propósitos como son:

- Planificar las integraciones de sistema necesarias en cada iteración. Se sigue para ello un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue. Esto se basa fundamentalmente en las clases activas encontradas durante el diseño.
- Implementar las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementan como componentes de fichero que contienen código fuente.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema. (31)

Capítulo 4.

4.1. Modelo de Datos

Para profundizar en los detalles de las clases entidades ver ANEXO 3: DESCRIPCIÓN DE LAS CLASES ENTIDADES, en las tablas:

A.3.1 Descripción de la clase [PgSqlServerLoadProvider](#).

A.3.2 Descripción de la clase [ICugnotServer](#).

A.3.3 Descripción de la clase [CugnotServerBase](#).

4.2. Diagrama de despliegue

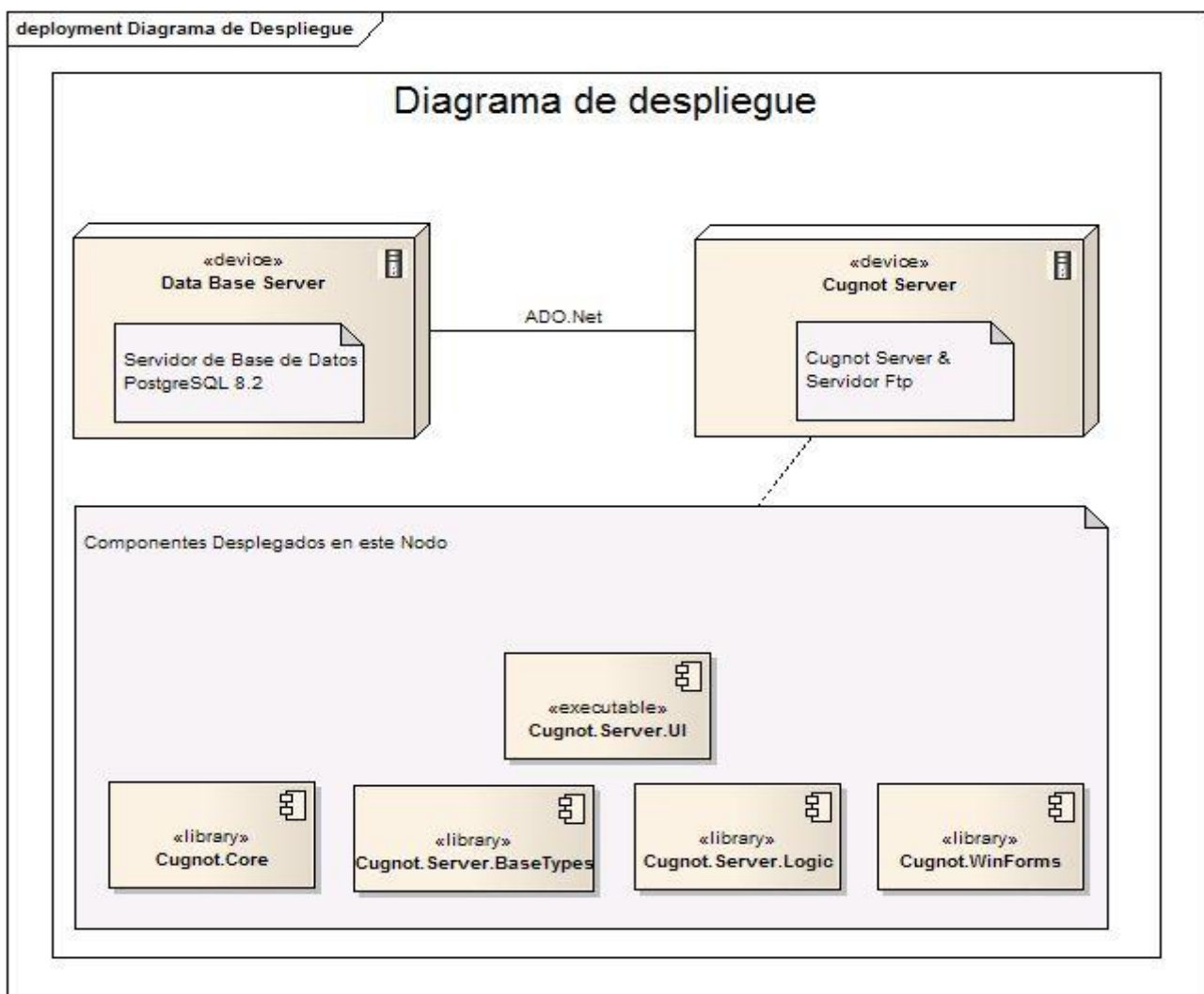


Figura 4.1 Diagrama de Despliegue.

4.3. Diagrama de componentes

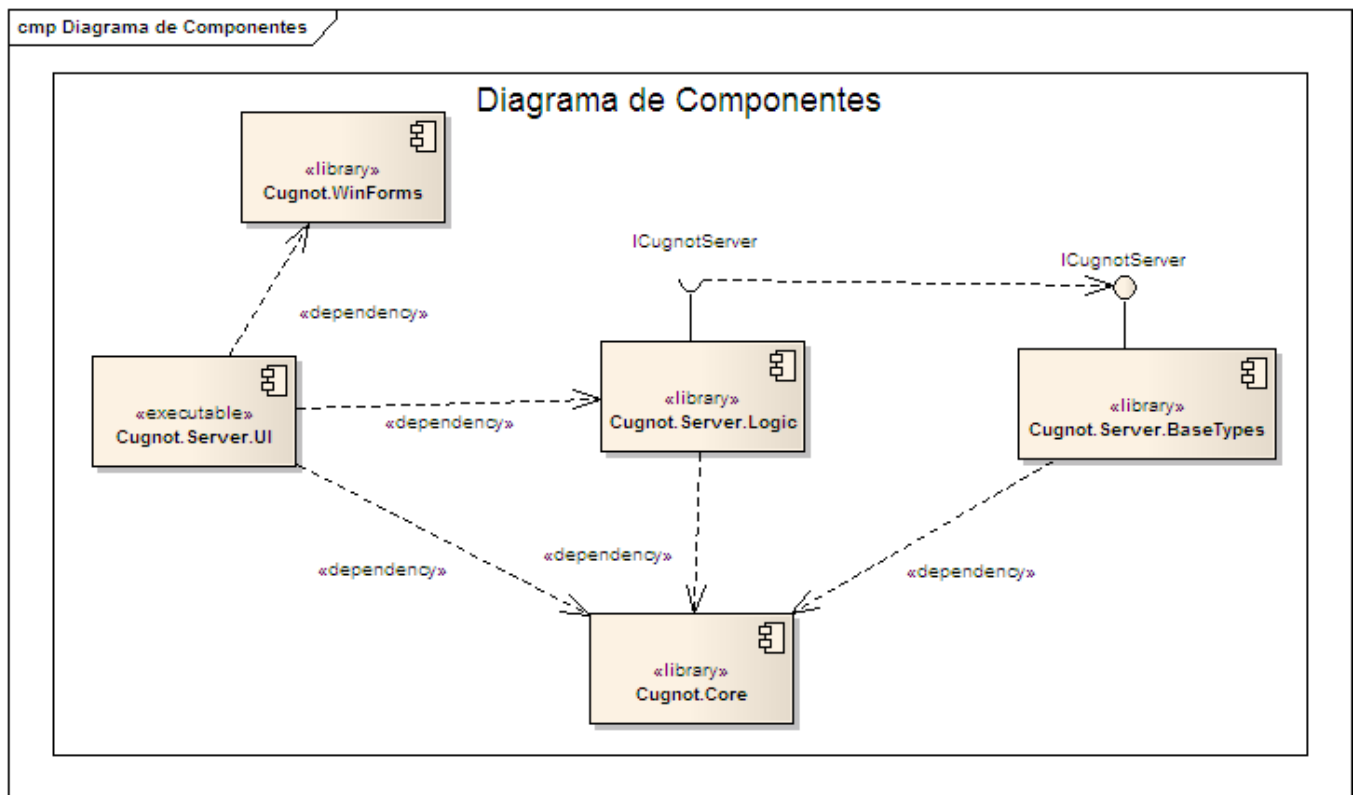


Figura 4.2 Diagrama de Componentes.

4.4. Tratamiento de errores

El tratamiento de errores se realiza estrictamente durante todo el proceso de implementación, basado en los principios de la plataforma .Net. Durante la codificación se realizaron validaciones tempranas con el objetivo de capturar las posibles excepciones o errores. En la suite Cugnot se lanzan excepciones sólo cuando estas pueden ser resueltas por el usuario.

4.5. Estrategias de seguridad

La solución Cugnot utiliza los roles de autenticación del sistema operativo. Toda aquella persona que desee utilizar la aplicación debe tener acceso a la máquina en la cual este corriendo la misma. Si algún usuario desea cambiar la configuración debe estar autenticado con una cuenta de administración de la máquina. Además, se utilizan los roles del servidor FTP. Si un usuario es administrador de la computadora pero no es administrador del servidor FTP no podrá realizar ninguna acción en este.

4.6. Estrategias de codificación

La estrategia de codificación utilizada son las guías de diseño de la plataforma .Net. Estas son convenciones de nomenclatura plasmadas en la plataforma. Para la declaración de las clases se comienza con mayúscula y si el nombre es compuesto las demás palabras comenzarían también con mayúsculas. Para la declaración de atributos se comienza con guión bajo seguido del nombre con letra minúscula. Las propiedades se escriben con mayúsculas y si son compuestas las demás palabras van con mayúsculas también. Los métodos igualmente, comienzan con mayúscula y las variables auxiliares con minúsculas.

El uso de estrategias de codificación garantiza un mejor entendimiento a la hora de leer el código. Asegura la reutilización del código y la eficacia en el trabajo, pues si se incorpora un nuevo integrante al equipo no le costaría entender el código existente.

En el capítulo se ha concluido la implementación del software. Fueron obtenidos los diagramas de despliegue y componentes, así como el tratamiento de errores, las estrategias de codificación y el aseguramiento del sistema frente a posibles ataques.

CONCLUSIONES

Al concluir este trabajo de diploma se han cumplido las tareas y el objetivo propuesto:

Se analizaron las tendencias actuales en la implementación de soluciones de control de vehículos. Para ello fueron identificados los principales mecanismos y protocolos de transmisión de archivos.

Se profundizó en las prestaciones de la plataforma .NET para el desarrollo de aplicaciones de monitoreo de imágenes y se fundamentó su utilización como entorno de desarrollo. Se demostró la necesidad e importancia de un servidor de imágenes en una solución de control de vehículos.

Todo lo cual permitió desarrollar la solución informática Cugnot Server para el monitoreo del repositorio de imágenes en el sistema de control de vehículos Cugnot Suite.

RECOMENDACIONES

Como resultado de la experiencia adquirida durante el desarrollo del presente trabajo de diploma los autores recomiendan:

- Dar continuidad a este trabajo en función de mejorar los sistemas de monitoreo de imágenes.
- Añadir a la solución Cugnot Server los servicios de un servidor FTP.
- Mantener actualizado el sistema, mediante la investigación continua de las tendencias más novedosas en el monitoreo de imágenes de los sistemas de control de vehículos.

REFERENCIAS BIBLIOGRÁFICAS

1. **Álvarez, Rafael Leodan Cordero.** Papel de Cugnot Suite. Octubre de 2008, pág. 3.
2. Universidad Francisco Marroquin. *IEspaña*. [En línea] 19 de Mayo de 2002.
http://elsitiodetelecomunicaciones.iespana.es/modelo_osi.htm.
3. Monografia.com. *Monografia.com*. [En línea] s/a.
<http://www.monografias.com/trabajos13/modosi/modosi.shtml>.
4. **wordpress.** f34k.wordpress.com/. *f34k.wordpress.com/*. [En línea]
<http://f34k.wordpress.com/2007/12/17/niveles-del-modelo-osi/>.
5. **Behrouz, Forouzan A.** *Transmisión de datos y redes de comunicaciones, Segunda edición*. Universidad Carlos III de Madrid : Mic Grao Hill, 2005.
6. **Barajas, Saulo.** Saulo.net. *Saulo.net*. [En línea] 8 de Diciembre de 2001.
<http://www.saulo.net/pub/tcpip/a.htm>.
7. Ver la referencia 5.
8. **Portel, J y Reinols, J.** *Request For Comments 959*. 2007.
9. www.softbull.com. *www.softbull.com*. [En línea] <http://www.softbull.com/>.
10. Ver la referencia 9.
11. Softonic. *Softonic*. [En línea] <http://filezilla-server.softonic.com/>.
12. Programas-gratis.net. *Programas-gratis.net*. [En línea] <http://ws-ftp-server.programas-gratis.net/>.
13. Ver la referencia 8.
14. *Request For Comments 114*. 1971.
15. Sencilla explicación y otros detalles del mismo protocolo. *servidorftp*. [En línea] 2005. [Citado el: 11 de November de 2008.] <http://www.servidorftp.es>.
16. WebRequest Class. *Msdn en español*. [En línea] Microsoft, 2007. [Citado el: 16 de November de 2008.] [http://msdn.microsoft.com/es-es/library/kwdt6w2k\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/kwdt6w2k(VS.80).aspx).
17. **Microsoft.** FtpWebRequest (Clase). *Msdn Online*. [En línea] Microsoft, 2007. [Citado el: 19 de November de 2008.] <http://msdn.microsoft.com/en-us/library/system.net.ftpwebrequest.aspx>.

Referencias Bibliográficas.

18. Computer Recognition Systems Ltd . *Computer Recognition Systems Ltd* . [En línea] <http://www.crs-traffic.co.uk/index.html>.
19. Anpr International Ltd. *Anpr International Ltd.* [En línea] <http://www.anpr-international.com/anpr-technology-c3.html>.
20. Security Design Services. [En línea] [Citado el: 18 de 11 de 2008.] <http://www.securitydesignservices.co.uk/>.
21. Cell2solutions. *Cell2solutions*. [En línea] <http://www.cell2solutions.com/en/anpr.php>.
22. Adaptive Recognition Hungary Inc. *Adaptive Recognition Hungary Inc.* [En línea] <http://www.arhungary.hu/> .
23. Centro de Aplicaciones de Tecnologías de Avanzada. [En línea] [Citado el: 18 de 11 de 2008.] <http://www.cenatav.co.cu/es/>.
24. **Yero, Yanoksy Durañona y Rodriguez, Lazaro Gonzalez.** PostgreSQL. *Cassandra Server*. Ciudad de la Habana : s.n., 2007.
25. **Galves, Jorge.** Fundamentos de la metodologia RUP, Rational Unified Process. [En línea] 16 de Septiembre de 2007. [Citado el: 20 de Noviembre de 2008.] <http://www.scribd.com/doc/297224/RUP>.
26. **Microsoft.** Msdn en español. [aut. libro] Microsoft Team. *Msdn*. Ciudad de la Habana : s.n., 2008.
27. Enterprise Architect. *Enterprise Architect*. [En línea] <http://www.sparxsystems.com/products/ea/index.html>.
28. **Jacobson, Ivar, Booch, Grandy y Bumbaugh, James.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación. S.A, 2000.
29. **Jacobson, Ivar, Booch, Gardy y Rumbaugh, James.** Análisis. *El prodeso Unificado de desarrollo de Software*. Madrid : s.n., 2000.
30. **IBM Maximo Asset Management, IBM Tivoli Asset Management.** *Workflow Implementation Guide*. 2008.
31. **UCI.** *Flujo de Implementación*. UCI, s.l. : s.n.

BIBLIOGRAFÍA

1. Adaptive Recognition Hungary Inc. *Adaptive Recognition Hungary Inc.* [En línea] <http://www.arhungary.hu/>.
2. **Álvarez, Rafael Leodan Cordero.** Papel de Cugnot Suite. Octubre de 2008, pág. 3.
3. Anpr International Ltd. *Anpr International Ltd.* [En línea] <http://www.anpr-international.com/anpr-technology-c3.html>.
4. **Archer, Tom.** *C# a fondo.* Washington : Mc Graw Hill, 2001.
5. **Barajas, Saulo.** Saulo.net. *Saulo.net.* [En línea] 8 de Diciembre de 2001.
6. Batanga, Autos. ¿Cuántos vehículos hay en todo el mundo? Batanga autos. . *Batanga autos.* . [En línea] http://autos.batanga.com/Cuantos_vehiculos_hay_en_todo_el_mundo?/44680588-6045-4C2D-B656-3A78BCF7A2CC.htm (último acceso: 9 de 11 de 2007).
7. **Behrouz, Forouzan A.** *Transmisión de datos y redes de comunicaciones, Segunda edición.* Universidad Carlos III de Madrid : Mic Grao Hill, 2005.
8. **Braude, J.** *Ingeniería de software: Una perspectiva orientada a objetos.* . s.l. : Ra-ma,, 2003.
9. **Boogs, Wendy, y Michael Boogs.** *Mastering UML with Rational Rose 2002.* California : s.n., 2002.
10. **Butow, Eric, y Tommy Ryan.** *C#: Your visual blueprint for building .NET applications.* New York : Hungry Minds, Inc, 2002.
11. Cell2solutions. *Cell2solutions.* [En línea] <http://www.cell2solutions.com/en/anpr.php>.
12. Centro de Aplicaciones de Tecnologías de Avanzada. [En línea] [Citado el: 18 de 11 de 2008.]
13. **Chonoles, Michael Jesse, y James A. Schardt.** *UML 2 for dummies.* New York: : Wiley, 2003.
14. <http://www.cenatav.co.cu/es/>.
15. **Cooper, James W.** *Introduction to Design Patterns in C#.* 2002.
16. CRS. Computer Recognition Systems. *CRS. Computer Recognition Systems.* [En línea] CRS. Computer Recognition Systems. <http://www.crs-traffic.co.uk/index.html> (último acceso: 2 de 12 de 2007).
17. Enterprise Architect. *Enterprise Architect.* [En línea] <http://www.sparxsystems.com/products/ea/index.html>.
18. *Fase de elaboración. Flujo de análisis y diseño. Modelo de análisis Conferencia de Ingeniería de Software I.* **UCI.** La Habana : s.n., 2008.
19. **Ferguson, Jeff, Brian Patterson, y Jason Beres.** *La biblia de C#.* Madrid: : Anaya Multimedia, 2003.

Bibliografía.

20. **Galves, Jorge.** Fundamentos de la metodología RUP, Rational Unified Process. [En línea] 16 de Septiembre de 2007. [Citado el: 20 de Noviembre de 2008.]
21. **Hamilton, Kim, y Russell Miles.** *Learning UML 2.0.* 2006.
22. **Harvey, Deitel M, Paul J Deitel, Jeffrey A Listfield, Tem R Nieto, Cheryl H Yaeger, y Marina Zlaktina.** *C# How to program.* s.l. : Prentice Hall, 2001.
23. <http://f34k.wordpress.com/2007/12/17/niveles-del-modelo-osi/>.
24. <http://www.saulo.net/pub/tcpip/a.htm>.
25. <http://www.securitydesignservices.co.uk/>.
26. **IBM Maximo Asset Management, IBM Tivoli Asset Management.** *Workflow Implementation Guide.* 2008.
27. **Jacobson, Ivar, Booch, Gardy y Rumbaugh, James.** *Análisis. El prodeso Unificado de desarrollo de Software.* Madrid : s.n., 2000.
28. **Larman, Graig.** *Capas arquitectbnicas.* [aut. libro] Pablo Eduardo Roig Viquez. *UML y Patrones Introducción al análisis y diseño orientado a objetos.* México : Version en español de la primera edición de la obra titulada: Applying UML and Pattern. An Introduction to Object-Oriented Analysis, 1999.
29. **Larman, Craic.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : Prentice Hall, 1999.
30. *Me Rinde.* [En línea] <http://merinde.rinde.gob.ve/index.php>.
31. **Microsoft.** *C# language specification.* s.l. : Redmond, 2006.
32. **Microsoft.** *FtpWebRequest (Clase).* *Msdn Online.* [En línea] Microsoft, 2007. [Citado el: 19 de November de 2008.] <http://msdn.microsoft.com/en-us/library/system.net.ftpwebrequest.aspx>.
33. **Microsoft.** *Msdn en español.* [aut. libro] Microsoft Team. *Msdn.* Ciudad de la Habana : s.n., 2008.
34. *Monografia.com.* *Monografia.com.* [En línea] s/a.
<http://www.monografias.com/trabajos13/modosi/modosi.shtml>.
35. **Pressman, Roger S.** *Ingeniería de software: Un enfoque práctico.* s.l. : Addison-Wesley, 2002.
36. **Portel, J y Reinols, J.** *Request For Comments 959.* 2007.
37. *Programas-gratis.net.* *Programas-gratis.net.* [En línea] <http://ws-ftp-server.programas-gratis.net/>.
38. *Request For Comments 114.* 1971.
39. *Security Design Services.* [En línea] [Citado el: 18 de 11 de 2008.]

Bibliografía.

40. Sencilla explicación y otros detalles del mismo protocolo. *servidorftp*. [En línea] 2005. [Citado el: 11 de November de 2008.] <http://www.servidorftp.es>.
41. Softonic. *Softonic*. [En línea] <http://filezilla-server.softonic.com/>.
42. **UCI**. *Flujo de Implementación*. UCI, s.l. : s.n.
43. WebRequest Class. *Msdn en español*. [En línea] Microsoft, 2007. [Citado el: 16 de November de 2008.] [http://msdn.microsoft.com/es-es/library/kwdt6w2k\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/kwdt6w2k(VS.80).aspx).
44. **Wesley, Addison**. *El Proceso Unificado de Desarrollo de Software*. 1999.
45. **Wordpress**. f34k.wordpress.com/. *f34k.wordpress.com/*. [En línea]
46. www.softbull.com. *www.softbull.com*. [En línea] <http://www.softbull.com/>.
47. **Yero, Yanoksy Durañona y Rodriguez, Lazaro Gonzalez**. *PostgreSQL. Cassandra Server*. Ciudad de la Habana : s.n., 2007.

ANEXOS

Anexos 1: Descripción de las Clases Interfaces

A.1.1 Descripción de la clase [MainForm](#).

Nombre: MainForm	
Tipo de clase: Interfaz	
Atributo	Tipo
_controller	ServerController
_startScreen	StartUserControl
_configurationScreen	ConfigurationUserControl
_logScreen	EventsUserControl
_legalScreen	ProductAboutUserControl
Para cada responsabilidad:	
Nombre:	InitializeU()
Descripción:	Inicializa los elementos de la interfaz de usuario de acuerdo a la cultura actual.
Nombre:	InitHelpProviders()
Descripción:	Inicializa el mecanismo que muestra la sugerencia asociada a cada control del formulario.
Nombre:	ShowScreen(AppScreen screen)
Descripción:	Muestra una pantalla en la interfaz de usuario del producto.
Nombre:	InkHelp_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el label que indica la ayuda.
Nombre:	MainForm_KeyUp(object sender, KeyEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se presiona la tecla F1.

Anexos.

Nombre:	InkStart_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el label link que indica que se debe mostrar la pantalla de inicio.
Nombre:	InkConfig_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el label link que indica que se debe mostrar la pantalla de configuración.
Nombre:	InkEventLog_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el label link que indica que se debe mostrar la pantalla de sucesos.
Nombre:	InkProductAbout_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el label link que indica que se debe mostrar la pantalla de server.

Tabla A.1.1 Descripción de MainForm.

A.1.2 Descripción de la clase [ConfigurationUserController](#)

Nombre: ConfigurationUserController	
Tipo de clase: Interfaz	
Atributo	Tipo
_controller	ServerController
_localFtpCredentials	FtpCredentials
_transferFtpCredentials	FtpCredentials
_dataSourceCredentials	DbCredentials
_volumeCapacity	long
_localFtpCredentialsForm	EstablishCredentialsForm
_selectDataSourceForm	SelectDataSourceForm
_testDataSourceCallback	TestDataSourceCallback
_formatter	IStringFormatter < TextValuePair < string >>
_showStateCallback	ShowStateCallback
Para cada responsabilidad:	
Nombre:	ShowLocalFtpCredentialsForm()
Descripción:	Muestra el formulario que permite establecer las credenciales de acceso al servidor FTP local.
Nombre:	ShowDataSourceForms()
Descripción:	Muestra el formulario que permite establecer las credenciales de acceso al origen de datos.
Nombre:	UpdateConfiguration()
Descripción:	Actualiza la configuración de Cugnot Server.
Nombre:	InitializeIU()
Descripción:	Inicializa los elementos de la interfaz de usuario de acuerdo a la cultura actual.

Anexos.

Nombre:	InitHelpProviders()
Descripción:	Inicializa el mecanismo que muestra la sugerencia asociada a cada control del formulario.
Nombre:	FillLogicalDrives()
Descripción:	LLena la lista desplegable que muestra el nombre las unidades lógicas existentes en el equipo local.
Nombre:	TestConnection()
Descripción:	Hace un test al servidor Ftp de transferencia con las credenciales que se encuentran en los controles.
Nombre:	ShowState()
Descripción:	Muestra el estado actual del controlador.
Nombre:	TestDataSource(DataSource target)
Descripción:	Comprueba la disponibilidad del origen de datos.
Nombre:	_localFtpCredentialsForm_CredentialsChanged(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando cambian las credenciales de acceso al servidor FTP local.
Nombre:	_selectDataSourceForm_DataSourceChanged(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando cambian las credenciales de acceso al origen de datos.
Nombre:	checkBoxAnonymousFtp_CheckedChanged(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en la casilla de verificación que indica que el servidor FTP de respaldo soporta el modo "FTP anónimo".
Nombre:	rBtnDelete_CheckedChanged(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el botón de selección que indica que las imágenes deben eliminarse.
Nombre:	buttonEstablishingCredentials_Click(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el botón que permite establecer las credenciales de acceso al servidor FTP local.

Anexos.

Nombre:	btnSetDataSource_Click(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando cuando se hace clic en el botón que permite establecer las credenciales del acceso al origen de datos.
Nombre:	InkTestConnection_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el botón que permite comprobar la disponibilidad del servidor FTP de respaldo, en caso de ser necesario.
Nombre:	trackBarUsedSpace_Scroll(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando cambia el valor de la barra de desplazamiento que indica el por ciento de utilización en el disco que provoca una reacción de Cugnot Server.
Nombre:	btnUpdateConfig_Click(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el botón que permite actualizar la configuración de Cugnot Server.
Nombre:	trackBarData_Scroll(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando cambia el valor de la barra de desplazamiento que indica el volumen de datos a procesar durante una reacción de Cugnot Server.

Tabla A.1.2 Descripción de ConfigurationUserControl.

Anexos.

A.1.3 Descripción de la clase [EstablishCredentialsForm](#)

Nombre: EstablishCredentialsForm	
Tipo de clase: Interfaz	
Atributo	Tipo
_credentials	FtpCredentials
CredentialsChanged	EventHandler<EventArgs>
Para cada responsabilidad:	
Nombre:	Credentials
Descripción:	Retorna las credenciales de acceso al servidor FTP local.
Nombre:	OnCredentialsChanged(EventArgs e)
Descripción:	Invoca el evento <see cref="CredentialsChanged"/> .
Nombre:	ShowCredentials(FtpCredentials credentials)
Descripción:	Muestra el estado asociado a una credencial en los controles de la interfaz de usuario.
Nombre:	FtpCredentials GetCredentials()
Descripción:	Retorna la credencial equivalente al estado que se muestra en los controles de la interfaz de usuario.
Nombre:	AcceptChanges()
Descripción:	Acepta los cambios efectuados en los controles de la interfaz de usuario.
Nombre:	CancelChanges()
Descripción:	Cancela los cambios efectuados en los controles de la interfaz de usuario.
Nombre:	TestConnection()
Descripción:	Hace un test al servidor FTP con las credenciales que se encuentran en los controles.
Nombre:	InitializeIU()
Descripción:	Inicializa los elementos de la interfaz de usuario de acuerdo a la cultura actual.

Anexos.

Nombre:	InitHelpProviders()
Descripción:	Inicializa el mecanismo que muestra la sugerencia asociada a cada control del formulario.
Nombre:	buttonCancel_Click(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se cancelan los cambios efectuados en los controles de la interfaz de usuario.
Nombre:	checkBoxAnonymousFtp_CheckedChanged(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando cambia el estado de la casilla de verificación que indica si el servidor FTP soporta el modo "Anónimo".
Nombre:	buttonAccept_Click(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se aceptan los cambios efectuados en los controles de la interfaz de usuario.
Nombre:	InkTestConnection_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el botón que permite comprobar la disponibilidad de un servidor FTP.

Tabla A.1.3 Descripción de EstablishCredentialsForm.

A.1.4 Descripción de la clase [EventsUserController](#)

Nombre: EventsUserController	
Tipo de clase: Interfaz	
Atributo	Tipo
_controller	ServerController
_showStateCallback	ShowStateCallback
Para cada responsabilidad:	
Nombre:	ClearEventLog()
Descripción:	Elimina todas las entradas existentes en le Registro de sucesos de Cugnot Server.
Nombre:	ShowEntriesOnIU()
Descripción:	Muestra en los controles de la interfaz de usuario todas las entradas existentes en el registro de sucesos de Cugnot Server.
Nombre:	ShowEntryOnIU(MyEventLogEntry entry)
Descripción:	Muestra en los controles de la interfaz de usuario el estado asociado a una entrada del registro de sucesos de Cugnot Server.
Nombre:	ShowState()
Descripción:	Muestra el estado actual del controlador
Nombre:	InitialzeIU()
Descripción:	Inicializa los elementos de la interfaz de usuario de acuerdo a la cultura actual.
Nombre:	InitHelpProviders()
Descripción:	Inicializa el mecanismo que muestra la sugerencia asociada a cada control del formulario.
Nombre:	InitHelpProviders()
Descripción:	Establece las opciones de formato de la tabla que muestra las entradas existentes en el registro de sucesos de Cugnot Server.
Nombre:	IsOk()

Anexos.

Descripción:	Indica si la instancia del controlador puede ser manejada.
Nombre:	_controller_LogCleared(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se eliminan todas las entradas existentes en el registro de sucesos de Cugnot Server.
Nombre:	_controller_LogEntryWritten(object sender, LogEntryWrittenEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se adiciona una nueva entrada al Registro de sucesos de Cugnot Server.
Nombre:	IViewLogEntries_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando cambia la selección en la tabla que muestra las entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	btnClear_MouseEnter(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando entra el mouse al Botón btnClear.
Nombre:	btnClear_MouseClick(object sender, MouseEventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando se hace clic en el botón que permite eliminar todas las entradas existentes en el Registro de sucesos de Cugnot Server.

Tabla A.1.4 Descripción de EventsUserControl.

A.1.5 Descripción de la clase [ProductAboutUserControl](#)

Nombre: ProductAboutUserControl	
Tipo de clase Interfaz	
Atributo	Tipo
_controller	ServerController
_showStateCallback	ShowStateCallback
Para cada responsabilidad:	
Nombre:	LaunchEula()
Descripción:	Muestra el contenido del contrato de usuario de licencia final asociado el producto Cugnot Server.
Nombre:	FormatTable()
Descripción:	Establece las opciones de formato de la tabla que muestra la información legal asociada al producto Cugnot Server.
Nombre:	InitializeUI()
Descripción:	Inicializa los elementos de la interfaz de usuario de acuerdo a la cultura actual.
Nombre:	ShowState()
Descripción:	Muestra el estado actual del controlador
Nombre:	InitHelpProviders()
Descripción:	Inicializa el mecanismo que muestra la sugerencia asociada a cada control del formulario.

Tabla A.1.5 Descripción de ProductAboutUserControl.

A.1.6 Descripción de la clase [StartUserController](#)

Nombre: StartUserController	
Tipo de clase: Interfaz	
Atributo	Tipo
_controller	ServerController
_drawTimer	System.Threading. Timer
_partitionTimer	System.Threading. Timer
_showStateCallback	ShowStateCallback
Para cada responsabilidad:	
Nombre:	UpdateControl()
Descripción:	Actualiza la información que se muestra en la pantalla.
Nombre:	ShowState()
Descripción:	Muestra el estado actual del controlador.
Nombre:	InitializeIU()
Descripción:	Inicializa los elementos de la interfaz de usuario de acuerdo a la cultura actual.
Nombre:	InitHelpProviders()
Descripción:	Inicializa el mecanismo que muestra la sugerencia asociada a cada control del formulario.
Nombre:	ShowPartitionState()
Descripción:	Muestra el estado del disco.
Nombre:	DrawRamUsage()
Descripción:	Dibuja el historial de la utilización de la memoria RAM.
Nombre:	ShowInformation()
Descripción:	Muestra datos estadísticos sobre las imágenes almacenadas en el servidor.
Nombre:	InitTimers()

Anexos.

Descripción:	Inicializa los timers que se utilizan en esta pantalla.
Nombre:	HandleDrawTimerTick(object state)
Descripción:	Controla las acciones a ejecutar cuando el mecanismo que se utiliza para actualizar los gráficos relacionados con el estado de la memoria RAM hace tick.
Nombre:	HandlePartitionTimerTick(object state)
Descripción:	Controla las acciones a ejecutar cuando el mecanismo que se utiliza para actualizar el gráfico relacionado con el estado de la partición hace Tick.
Nombre:	_controller_ServerLoadChanged(object sender, EventArgs e)
Descripción:	Especifica las acciones a ejecutar cuando la carga laboral del servidor cambia.

Tabla A.1.6 Descripción de StartUserControl.

Anexos 2: Descripción de la Clase Controladora

A.2.1 Descripción de la clase [ServerController](#)

Nombre: ServerController	
Tipo de clase: Controladora	
Atributo	Tipo
_remoteCugnotServer	ICugnotServer
_warningChangedWrapper	WarningChangedEventWrapper
_executionStatusChangedWrapper	ExecutionStatusChangedEventWrapper
_logEntryWrittenWrapper	LogEntryWrittenEventWrapper
_logClearedWrapper	LogClearedEventWrapper
_serverLoadChangedWrapper	RepoStateChangedEventWrapper
_logClearedWrapper	LogClearedEventWrapper
_serverLoadChangedWrapper	RepoStateChangedEventWrapper
Para cada responsabilidad:	
Nombre:	Warning
Descripción:	Retorna o o modifica el aviso principal de Cugnot Server.
Nombre:	VolumeName
Descripción:	Retorna o modifica el identificador del volumen que Cugnot Server debe monitorear.
Nombre:	UsedSpacePercent
Descripción:	Retorna o modifica el por ciento máximo de utilización permitido en el volumen que Cugnot Server debe monitorear.
Nombre:	DataPercent
Descripción:	Retorna o modifica el por ciento de imágenes a procesar durante una reacción de Cugnot Server.

Cugnot Server: Servidor de Imágenes de vehículos.

Anexos.

Nombre:	StartupAutomatically
Descripción:	Indica si Cugnot Server debe iniciarse junto al Sistema Operativo.
Nombre:	ExecutionStatus
Descripción:	Retorna el estado de ejecución de Cugnot Server.
Nombre:	TransferFtp
Descripción:	Retorna o modifica las credenciales del servidor FTP de transferencia.
Nombre:	LocalFtp
Descripción:	Retorna o modifica las credenciales del servidor FTP local.
Nombre:	DbCredentials
Descripción:	Retorna o modifica las credenciales de acceso al repositorio central.
Nombre:	ServerLoad
Descripción:	Retorna el objeto que representa la carga de imágenes en Cugnot Server.
Nombre:	Start()
Descripción:	Inicia la ejecución de Cugnot Server.
Nombre:	PauseOrResume()
Descripción:	Pausa o reanuda la ejecución de Cugnot Server.
Nombre:	Stop()
Descripción:	Detiene la ejecución de Cugnot Server.
Nombre:	GetLogEntries()
Descripción:	Retorna las entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	GetLogEntriesCount()
Descripción:	Retorna la cantidad de entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	ClearLog()

Anexos.

Descripción:	Elimina todas las entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	UpdateServerLoad()
Descripción:	Actualiza el objeto que representa la carga de imágenes en Cugnot Server.
Nombre:	CauseReaction()
Descripción:	Ejecuta una reacción de Cugnot Server.
Nombre:	GetMemoryUsage()
Descripción:	Retorna la cantidad memoria en uso, en el equipo local.
Nombre:	GetVolumeProperties()
Descripción:	Retorna la propiedades del volumen que Cugnot Server debe monitorear.
Nombre:	ExportLogEntries(string path)
Descripción:	Genera un reporte a partir de las entradas existentes en el Registro de sucesos.
Nombre:	OnWarningChanged(EventArgs e)
Descripción:	Invoca el evento <see cref="WarningChanged"/> .
Nombre:	OnExecutionStatusChanged(EventArgs e)
Descripción:	Invoca el evento <see cref="ExecutionStatusChanged"/> .
Nombre:	OnLogEntryWritten(LogEntryWrittenEventArgs e)
Descripción:	Invoca el evento <see cref="LogEntryWritten"/> .
Nombre:	OnLogCleared(EventArgs e)
Descripción:	Invoca el evento <see cref="LogCleared"/> .
Nombre:	OnServerLoadChanged(EventArgs e)
Descripción:	Invoca el evento <see cref="ServerLoadChanged"/> .
Nombre:	LegalProductKeys

Anexos.

Descripción:	Retorna los posibles nombres legales del producto que se legaliza.
Nombre:	ReleaseManagedResources()
Descripción:	Libera los recursos manejados asignados a esta instancia.
Nombre:	ReleaseUnmanagedResources()
Descripción:	Libera los recursos no manejados asignados a esta instancia.
Nombre:	ConnectServerObjects()
Descripción:	Conecta los objetos remotos asociados a esta instancia.
Nombre:	DisconnectServerObjects()
Descripción:	Desconecta los objetos remotos asociados a esta instancia.
Nombre:	GetRemoteObjectUrl()
Descripción:	Retorna la URL del objeto remoto que expone las funcionalidades del producto Cugnot Server.
Nombre:	InitializeRemoteEvents()
Descripción:	Inicializa el mecanismo que interactúa con los eventos remotos del producto Cugnot Repository.
Nombre:	ClearRemoteEvents()
Descripción:	Establece la configuración predeterminada al mecanismo que interactúa con los eventos remotos del producto Cugnot Server.
Nombre:	_serverLoadChangedWrapper_RepoStateChanged(object sender, EventArgs e)
Descripción:	Invoca el evento <see cref="RepoStateChanged"/>
Nombre:	_logClearedWrapper_LogCleared(object sender, EventArgs e)
Descripción:	Invoca el evento <see cref="LogCleared"/>
Nombre:	_logEntryWrittenWrapper_LogEntryWritten(object sender, LogEntryWrittenEventArgs e)
Descripción:	Invoca el evento <see cref="LogEntryWritten"/>

Anexos.

Nombre:	_executionStatusChangedWrapper_ExecutionStatusChanged(object sender, EventArgs e)
Descripción:	Invoca el evento <see cref="ExecutionStatusChanged"/>
Nombre:	_warningChangedWrapper_WarningChanged(object sender, EventArgs e)
Descripción:	Invoca el evento <see cref="WarningChanged"/>

Tabla A.1.6 Descripción de ServerController.

Anexos 3: Descripción de las Clases entidad

A.3.1 Descripción de la clase [PgSqlServerLoadProvider](#)

Nombre: PgSqlServerLoadProvider	
Tipo de clase: Entidad.	
Atributo	Tipo
_formatter	IConnectionStringFormatter
Para cada responsabilidad:	
Nombre:	Formatter
Descripción:	Retorna el objeto de tipo <code><see cref="Albet.Formatters.IConnectionStringFormatter"/></code> que genera la cadena de conexión a una base de datos específica.
Nombre:	GetLoadInfo(string url, string connectionString)
Descripción:	Recupera información resumida sobre las entradas del Repositorio central que están asociadas a un servidor de imágenes específico.
Nombre:	GetLoadItems(string url, OrderingMode mode, PageLimits limits, string connectionString)
Descripción:	Recupera información detallada sobre las entradas del Repositorio central que están asociadas a un servidor de imágenes específico.

Tabla A.3.1 Descripción de [PgSqlServerLoadProvider](#).

A.3.2 Descripción de la clase [ICugnotServer](#)

Nombre: ICugnotServer	
Tipo de clase: Entidad.	
Atributo	Tipo
WarningChanged	EventHandler<EventArgs>
ExecutionStatusChanged	EventHandler<EventArgs>
LogEntryWritten	EventHandler<LogEntryWrittenEventArgs>
LogCleared	EventHandler<EventArgs>
ServerLoadChanged	EventHandler<EventArgs>
Para cada responsabilidad:	
Nombre:	VolumeName
Descripción:	Retorna o modifica el identificador currentDelegate volumen que Cugnot Server debe monitorear.
Nombre:	UsedSpacePercent
Descripción:	Retorna o modifica el porcentaje máximo de utilización permitido en el volumen que Cugnot Server debe monitorear.
Nombre:	DataPercent
Descripción:	Retorna o modifica el porcentaje de imágenes a procesar durante una reacción de Cugnot Server.
Nombre:	StartupAutomatically
Descripción:	Indica si Cugnot Server debe iniciarse junto al Sistema Operativo.
Nombre:	ExecutionStatus
Descripción:	Retorna el estado de ejecución de Cugnot Server.
Nombre:	LocalFtp
Descripción:	Retorna o modifica las credenciales currentDelegate servidor FTP local.

Anexos.

Nombre:	DbCredentials
Descripción:	Retorna o modifica las credenciales de acceso al repositorio central.
Nombre:	ServerLoad
Descripción:	Retorna el objeto que representa la carga de imágenes en Cugnot Server.
Nombre:	Start()
Descripción:	Inicia la ejecución de Cugnot Server.
Nombre:	PauseOrResume()
Descripción:	Pausa o reanuda la ejecución de Cugnot Server.
Nombre:	Stop()
Descripción:	Detiene la ejecución de Cugnot Server.
Nombre:	GetLogEntries()
Descripción:	Retorna las entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	GetLogEntriesCount()
Descripción:	Retorna la cantidad de entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	ClearLog()
Descripción:	Elimina todas las entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	UpdateServerLoad()
Descripción:	Actualiza el objeto que representa la carga de imágenes en Cugnot Server.
Nombre:	CauseReaction()
Descripción:	Ejecuta una reacción de Cugnot Server.

Tabla A.3.2 Descripción de ICugnotServer.

A.3.3 Descripción de la clase [CugnotServerBase](#)

Nombre: CugnotServerBase .	
Tipo de clase: Entidad	
Atributo	Tipo
_warning	string
_volumeName	string
_usedSpacePercent	byte
_dataPercent	byte
_startupAutomatically	bool
_isDisposed	bool
_executionStatus	ExecutionStatus
_serverLoad	RepoState
_usedSpaceTimer	Timer
_serverLoadUpdateTimer	Timer
_localFtp	FtpCredentials
_dbCredentials	DbCredentials
_urlUpdater	IServerUrlUpdater
_loadProvider	IServerLoadProvider
_crossingUpdater	ICrossingUpdater
_log	EventLog

Anexos.

WarningChanged	EventHandler<EventArgs>
ExecutionStatusChanged	EventHandler<EventArgs>
LogEntryWritten	EventHandler<LogEntryWrittenEventArgs>
LogCleared	EventHandler<EventArgs>
ServerLoadChanged	EventHandler<EventArgs>
Para cada responsabilidad:	
Nombre:	InitializeLifetimeService()
Descripción:	Controla el tiempo de vida de esta instancia.
Nombre:	Warning
Descripción:	Retorna o modifica el aviso principal de Cugnot Server.
Nombre:	VolumeName
Descripción:	Retorna o modifica el identificador del volumen que Cugnot Server debe monitorear.
Nombre:	UsedSpacePercent
Descripción:	Retorna o modifica el por ciento máximo de utilización permitido en el volumen que Cugnot Server debe monitorear.
Nombre:	DataPercent
Descripción:	Retorna o modifica el por ciento de imágenes a procesar durante una reacción de Cugnot Server.
Nombre:	StartupAutomatically

Cugnot Server: Servidor de Imágenes de vehículos.

Anexos.

Descripción:	Indica si Cugnot Server debe iniciarse junto al Sistema Operativo.
Nombre:	ExecutionStatus
Descripción:	Retorna el estado de ejecución de Cugnot Server.
Nombre:	TransferFtp
Descripción:	Retorna o modifica las credenciales del servidor FTP de transferencia.
Nombre:	LocalFtp
Descripción:	Retorna o modifica las credenciales del servidor FTP local.
Nombre:	DbCredentials
Descripción:	Retorna o modifica las credenciales de acceso al repositorio central.
Nombre:	ServerLoad
Descripción:	Retorna el objeto que representa la carga de imágenes en Cugnot Server.
Nombre:	IsDisposed
Descripción:	Indica si los recursos asignados a esta instancia han sido liberados.
Nombre:	Start()
Descripción:	Inicia la ejecución de Cugnot Server.
Nombre:	PauseOrResume()
Descripción:	Pausa o reanuda la ejecución de Cugnot Server.
Nombre:	Stop()
Descripción:	Detiene la ejecución de Cugnot Server.

Anexos.

Nombre:	GetLogEntries()
Descripción:	Retorna las entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	GetLogEntriesCount()
Descripción:	Retorna la cantidad de entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	ClearLog()
Descripción:	Elimina todas las entradas existentes en el Registro de sucesos de Cugnot Server.
Nombre:	UpdateServerLoad()
Descripción:	Actualiza el objeto que representa la carga de imágenes en Cugnot Server.
Nombre:	CauseReaction()
Descripción:	Ejecuta una reacción de Cugnot Server.
Nombre:	Dispose()
Descripción:	Libera los recursos asignados a la instancia.
Nombre:	~MyCugnotServer()
Descripción:	Libera los recursos de esta instancia.
Nombre:	OnWarningChanged(EventArgs e)
Descripción:	Invoca el evento <see cref="WarningChanged"/>.
Nombre:	OnExecutionStatusChanged(EventArgs e)

Anexos.

Descripción:	Invoca el evento <see cref="ExecutionStatusChanged"/>.
Nombre:	OnLogEntryWritten(LogEntryWrittenEventArgs e)
Descripción:	Invoca el evento <see cref="LogEntryWritten"/>.
Nombre:	OnLogCleared(EventArgs e)
Descripción:	Invoca el evento <see cref="LogCleared"/>.
Nombre:	OnServerLoadChanged(EventArgs e)
Descripción:	Invoca el evento <see cref="ServerLoadChanged"/>.
Nombre:	ReleaseManagedResources()
Descripción:	Libera los recursos manejados asignados a esta instancia.
Nombre:	ReleaseUnmanagedResources()
Descripción:	Libera los recursos no manejados asignados a esta instancia.
Nombre:	UrlUpdater
Descripción:	Retorna el mecanismo que se utiliza para sincronizar en las entradas del Repositorio central los cambios que se producen en el URL del servidor.
Nombre:	LoadProvider
Descripción:	Retorna el mecanismo que se utiliza para recuperar información sobre las entradas del Repositorio central que están asociadas al servidor.
Nombre:	CrossingUpdater
Descripción:	Retorna el mecanismo que se utiliza para actualizar los datos almacenados en el Repositorio central.

Anexos.

Nombre:	Dispose(bool disposing)
Descripción:	Libera los recursos asociados a la instancia.
Nombre:	RaiseExceptionIfNecessary()
Descripción:	Lanza una excepción si los recursos asociados a la instancia ya han sido liberados.
Nombre:	HandleUsedSpaceTick(object state)
Descripción:	Controla las acciones a ejecutar cuando el Timer asociado al mecanismo que monitorea el espacio ocupado en el Servidor de imágenes de vehículos hace Tick.
Nombre:	HandleServerLoadUpdateTick(object state)
Descripción:	Controla las acciones a ejecutar cuando el Timer asociado al mecanismo de actualización de la carga de imágenes en el Servidor hace Tick.
Nombre:	LoadDependenciesIfNecessary()
Descripción:	Carga las dependencias.
Nombre:	LoadDefaultConfig()
Descripción:	Carga la configuración por defecto de Cugnot Server.
Nombre:	ReadConfigIfNecessary()
Descripción:	Lee la última configuración de la instancia.
Nombre:	WriteConfig(
Descripción:	Salva la última configuración de la instancia.

Anexos.

Nombre:	ReadStatelfNecessary()
Descripción:	Lee el último estado de la instancia.
Nombre:	WriteState()
Descripción:	Salva el último estado de la instancia.
Nombre:	ReadUnsyncIfNecessary()
Descripción:	Lee los últimos datos pendientes por actualización en el Repositorio central.
Nombre:	WriteUnsync()
Descripción:	Salva los últimos datos pendientes por actualización hacia el Repositorio central.
Nombre:	AddUnsync(Crossing current, Crossing value)
Descripción:	Adiciona una nueva entrada a los datos pendientes por actualización en el Repositorio central.
Nombre:	CheckUnsyncs()
Descripción:	Activa el mecanismo que maneja las entradas pendientes por actualización en el Repositorio central.
Nombre:	RegisterStatusChange(ExecutionStatus current, ExecutionStatus value)
Descripción:	Registra en el Visor de sucesos los cambios en el estado de ejecución de Cugnot Server.
Nombre:	InitializeTimers()
Descripción:	Inicializa los Timers que activan los mecanismos involucrados en la instancia.

Anexos.

Nombre:	InitializeLog()
Descripción:	Inicializa el objeto asociado al Registro de sucesos del componente Cugnot Server.
Nombre:	AddLogEntry(MyEventLogEntry entry)
Descripción:	Adiciona una nueva entrada al Registro de sucesos.
Nombre:	CheckVolumeName(string value
Descripción:	Comprueba si una cadena coincide con el identificador de algún volumen lógico del equipo local.
Nombre:	RequestInstanceOwnership()
Descripción:	Solicita el derecho de posesión de la instancia única de la clase.
Nombre:	RequestInstanceOwnership()
Descripción:	Solicita el derecho de posesión de la instancia única de la clase.
Nombre:	ReleaseInstanceOwnership()
Descripción:	Libera el derecho de posesión de la instancia única de la clase.
Nombre:	GetUsedSpacePercent()
Descripción:	Retorna el porcentaje de utilización en el volumen que Cugnot Server debe monitorear.

Tabla A.3.3 Descripción de CugnotServerBase.