

Universidad de las Ciencias Informáticas

Facultad 7



Cugnot Explorer: Explorador del Repositorio de Datos

Trabajo de Diploma para optar por el título

Ingeniero en Ciencias Informáticas

Autora: Daily González Chery

Tutores: Ing. Jesmar Ernesto Fajardo Martin

Ing. Rafael Leodán Cardero Álvarez

Ciudad de La Habana, Junio de 2009

“Año del 50 aniversario del triunfo de la Revolución”

PENSAMIENTO

Hagamos el propósito de redoblar nuestros esfuerzos y juremos ante nosotros mismos que si un día nuestro trabajo nos pareciera bueno, debemos luchar por hacerlo mejor, y si fuera mejor, debemos luchar por hacerlo perfecto, conociendo de antemano que para un comunista nada será nunca suficiente bueno, y ninguna obra humana será jamás suficientemente perfecta.

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 12 días del mes de junio del año 2009.

Firma del Autor

Firma del Tutor

Firma del Tutor

Datos de Contacto

Tutores:

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.

Rafael L. Cardero Álvarez (email rlcardero@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.

AGRADECIMIENTOS

A mis padres, hermanos, tíos, primos y abuelos porque sin quererlo han sido autores de esta tesis; pues han sido mi faro y mi guía para poder realizarla.

A mis tutores Ing. Rafael Leodán Cardero Álvarez y Ing. Jesmar Ernesto Fajardo Martín pues sin ellos no hubiera sido posible la realización de esta tesis.

A mis amistades y compañeros de grupos principalmente Augusto y Yordanis por toda la ayuda que me han brindado.

A todos los que de una forma u otra aportaron su granito de arena en la realización de este trabajo y que no fueron mencionados.

A todos, Muchas Gracias.

DEDICATORIA

A mis padres que han sido mi mayor inspiración en todos estos años de estudio y esfuerzo.

A mis hermanos, primos, tíos y abuelos por estar siempre en todos los momentos de mi vida.

A la familia de mi novio por el apoyo incondicional que me han sabido dar todo este tiempo.

A Reinier Hernández Rodríguez por ser mi amigo, mi compañero, mi pareja; por haber estado ahí todo este año, por haber soportado mis cambios de humor y por todo su apoyo.

RESUMEN

En Cuba el control de vehículos se realiza generalmente de forma manual. Esto trae como consecuencia la pérdida de los registros de datos y conlleva un gran esfuerzo por parte del personal que se encarga de esta tarea. Además no existe un sistema eficiente que permita controlar de forma general toda la información que se genera producto al control vehicular, lo que trae como consecuencia que se dificulte la búsqueda de la misma.

El Grupo de Procesamiento de Imágenes (GPI) de la facultad 7 de la Universidad de Ciencias Informáticas está desarrollando una solución que permite llevar a cabo el control vehicular de forma automatizada. Cugnot Explorer forma parte de esta solución y tiene como objetivo facilitar la visualización de los datos de la **suite Cugnot**.

Para el desarrollo de la aplicación se utiliza como tecnología .Net ASP.net, como gestor de base de datos PostgreSQL y para lograr una solución dinámica e interactiva se utiliza la librería Yahoo User Interface.

Se espera que Cugnot Explorer brinde al usuario las herramientas que necesita para tener acceso a la información de la Base de Datos de tráfico de la suite. De forma general se espera que la solución **Cugnot** brinde un gran aporte a la sociedad ante la necesidad que tiene el país de contar con un sistema automatizado de control vehicular; pues este sistema resulta difícil de obtener por otras vías debido al alto costo que tiene mundialmente.

PALABRAS CLAVES

Control vehicular, Cugnot Explorer, Visualización de Datos, **suite Cugnot**

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	4
1.1 Estado del Arte.....	4
1.1.1 Sistemas Internacionales	4
1.1.2 Sistemas Nacionales.....	8
1.2 Principales herramientas y tecnologías utilizadas.....	8
1.2.1 Tecnología .Net.....	8
1.2.2 Librería de Interfaz de Usuario	9
1.2.3 Entorno de Desarrollo Integrado	9
1.2.4 Gestor de Base de Datos.....	10
1.2.5 Controlador de Versiones.....	12
1.2.6 Herramienta Case	13
1.2.7 Lenguaje de Modelación	13
1.2.8 Metodología de Desarrollo	14
1.3 Justificación de la elección de las herramientas y tecnologías utilizadas.....	14
CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA	15
2.1 Objeto de automatización	15
2.2 Propuesta de Sistema	15
2.3 Descripción del Negocio	16
2.4 Modelo de Negocio.....	16
2.4.1 Actores del Negocio.....	16

2.4.2 Trabajadores del Negocio	16
2.4.3 Diagrama de CU del Negocio.....	17
2.4.4 Diagrama de Clases del Modelo de Objetos	17
2.5 Especificación de los requisitos del software	20
2.5.1 Requerimientos Funcionales.....	20
2.5.2 Requerimientos no Funcionales.....	20
2.6 Definición de los Casos de Uso del Sistema.....	22
2.6.1 Definición de los actores	22
2.6.2 Listado de Casos de Uso	23
2.6.3 Diagrama de Casos de Uso del Sistema.....	25
2.6.4 Casos de Uso por Ciclo de desarrollo.	25
2.6.5 Expansión de los Casos de Uso del Sistema	27
CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA	28
3.1 Patrones de Arquitectura	28
3.1.1 Arquitecturas en Capas.....	28
3.2 Modelo de Análisis	31
3.2.1 Diagrama de Clases del Análisis.....	32
3.3 Modelo de Diseño.....	32
3.3.1 Diagramas de Clases del Diseño	33
3.3.2 Diagramas de Interacción del Diseño.....	36
3.4 Descripción de las Clases	37
3.5 Modelo de Datos	37
CAPÍTULO IV: IMPLEMENTACIÓN.....	39

4.1 Diagrama de Despliegue	40
4.2 Diagrama de Componentes	40
CONCLUSIONES	42
RECOMENDACIONES.....	43
REFERENCIAS	44
BIBLIOGRAFÍA.....	46
ANEXOS	49
Anexo 1	49
Anexo 2	54
Anexo 3	56
Anexo 4	58
Anexo 5	60

INTRODUCCIÓN

Con el crecimiento de las ciudades y la utilización masiva de vehículos fue necesario utilizar métodos para el control vehicular por parte de las autoridades. Inicialmente este control era realizado de forma manual por los agentes del orden y la información era almacenada por escrito, lo que traía como consecuencia la pérdida de la información por deterioro de los registros. Además resultaba difícil para el personal que se encargaba de esta tarea acumular tanta información, lo que además dificulta la búsqueda y almacenamiento de la misma.

Ante esta situación y el avance de las tecnologías, específicamente de la informática, surgieron los **sistemas de reconocimiento automático de matrículas** (*Automatic number plate recognition, ANPR* en inglés), reduciendo así esfuerzo humano, mejorando la seguridad y mejorando la eficacia operacional.

El ANPR¹ es un método de vigilancia en masa que utiliza reconocimiento óptico de caracteres en imágenes para leer las matrículas de los vehículos. Se utiliza para almacenar las imágenes de las matrículas de los vehículos capturadas por cámaras, y algunas se pueden configurar para almacenar una fotografía del conductor. Estos sistemas a menudo utilizan iluminación infrarroja para hacer posible que la cámara pueda tomar fotografías en cualquier momento del día. La tecnología ANPR tiende a ser específica para una región, debido a la variación entre matrículas de un lugar a otro.

El software del sistema para el reconocimiento automático de matrículas se ejecuta sobre una microcomputadora estándar y puede ser enlazado con otras aplicaciones o bases de datos. Primero utiliza una serie de técnicas de manipulación de la imagen para detectar, normalizar y realzar la imagen del número de la matrícula, y finalmente reconocimiento óptico de caracteres para extraer los alfanuméricos de la matrícula.

Los propietarios de vehículos han utilizado una variedad de técnicas para intentar evadir los sistemas ANPR. Un método consiste en incrementar las propiedades de reflexión de las letras y aumentar así la probabilidad de que el sistema no sea capaz de localizar la matrícula o de producir suficiente nivel de contraste para lograr leerla.

¹ ANPR: Reconocimiento automático del número de la matrícula

Cugnot Explorer: Explorador del Repositorio de Datos

Introducción

El reconocimiento automático de los vehículos permitiría al país un mejor control para el acceso de los vehículos a instalaciones. Así como facilitar la localización de autos robados, la detección de violaciones de las leyes del tránsito, el control de peajes y el monitoreo de autopistas.

En el país existe un sistema de reconocimiento automático de vehículo de origen ruso, pero el mismo cumple sus funciones solamente en algunos puntos de Ciudad de la Habana pues constituye un gasto económico muy grande debido a que es un software propietario.

El proceso de identificación de vehículos en Cuba se realiza generalmente de forma manual, lo que trae como consecuencia la pérdida de los registros de datos. No se cuenta con un sistema integral que permita monitorear el tráfico en las autopistas y gestionar la información generada.

En el Polo productivo “Imágenes” de la Facultad 7, de la Universidad de las Ciencias Informáticas, se está desarrollando la solución “**Cugnot. Suite** para la identificación de los vehículos”. La solución permite a partir de la imagen de video de un vehículo extraer el número de su matrícula, almacenando la misma en un repositorio de imágenes y guardando en una base de datos la información referente al vehículo.

Ante la necesidad que tiene el país de contar con un software de este tipo se hace necesario encontrar una vía fácil y dinámica que permita al usuario final del sistema acceder a todos los datos que se generan luego de procesadas las imágenes de acuerdo a criterios previamente establecidos; de ahí que el análisis de la problemática expuesta origine el planteamiento del siguiente **problema científico**: ¿cómo facilitar la visualización de los datos de la **Suite Cugnot**?

Queda definido como **objeto de estudio** el proceso de visualización de los datos de la Suite **Cugnot**; el **campo de acción** se enmarca en los procesos de visualización de información en sistemas de gestión de información. Como **objetivo general** se propone implementar un componente que facilite la visualización de los datos de la Suite **Cugnot**.

Para ello se trazaron las siguientes **tareas de investigación**:

- Determinar los lenguajes y tecnologías de desarrollo.
- Analizar las características básicas del lenguaje JavaScript.
- Realizar una investigación sobre Yahoo User Interface (YUI).
- Realizar un estudio de las características de sitios WEB que utilicen intercambio asincrónico de datos (AJAX).

Cugnot Explorer: Explorador del Repositorio de Datos

Introducción

- Identificar los aspectos teóricos conceptuales referentes a la implementación del sistema.
- Realizar el análisis de la aplicación.
- Realizar el diseño de la aplicación.
- Realizar la implementación utilizando los patrones de diseño establecidos en el análisis.
- Analizar los gestores de Base de Datos existentes con el objetivo de utilizar el más factible.
- Implementar el sitio que manipule los datos de la Base de Datos de tráfico de la Suite Cugnot.
- Generar la documentación asociada al desarrollo del producto Cugnot Explorer.

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

En el Capítulo 1: Fundamentación Teórica. Constituye el respaldo teórico de la investigación. Se realiza un análisis de las herramientas y tecnologías utilizadas.

El Capítulo 2: Características del Sistema. Se describen los procesos estudiados, así como la propuesta del sistema. También incluye los requerimientos, tanto funcionales como no funcionales, los modelos del negocio y del sistema.

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

El Capítulo 4: Implementación. Representación de los componentes finales resultantes del sistema 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 I: FUNDAMENTACIÓN TEÓRICA

Este capítulo está dedicado a realizar un análisis del estado del arte de los distintos sistemas de reconocimiento automático de matrículas, así como las distintas herramientas, metodologías y lenguajes de programación.

1.1 Estado del Arte

Las tecnologías de la informática y las comunicaciones han evolucionado aceleradamente en los últimos años. Además, en el presente, los vehículos son utilizados con gran auge en la mayoría de los procesos de la sociedad. Estos factores han favorecido la creación de compañías con la misión de desarrollar soluciones para la identificación vehicular.

A continuación se detallarán productos ofrecidos por algunas compañías líderes en este campo.

1.1.1 Sistemas Internacionales

Computer Recognition Systems Ltd

La compañía Computer Recognition Systems Ltd, CRS (1), 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. CRS desarrolla los productos más avanzados para el procesamiento de imágenes basado en cámaras para la detección, clasificación e identificación de vehículos sin importar las condiciones del tiempo.

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.

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 1

Algunos de sus logros son el desarrollo del primer lector de matrículas (en el año 1979), el desarrollo del primer lector de matrículas capaz de operar sin detectores de presencia (en el año 1983) y constituir la primera compañía del sector en introducir las cámaras digitales HDTV en sus soluciones (en el año 1997).

Anpr International Ltd

La compañía Anpr International Ltd (2) 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™, eyeGUARDIAN™ y streetSWEEPER™.

El sistema eyeTRAFFIC™ es una solución integral para monitorear el tráfico de los vehículos, pudiendo ser utilizada en varios entornos (ejemplo control de acceso). 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 DVLA². El sistema eyeGUARDIAN™ es un producto para gestionar parqueos mientras que streetSWEEPER™ (ver figura 1) constituye una unidad móvil que permite monitorear el tráfico de los vehículos.

Hi-tech solutions

La compañía Hi-tech solutions (3) es un líder mundial cuya sede central radica en Rishon, Le-Zion, Tel-Aviv, Israel. Los sectores cubiertos por sus productos son el control de acceso a las instalaciones, la gestión de parqueos y el monitoreo del tráfico en las autopistas. La misión de esta compañía es desarrollar aplicaciones “al aire libre” que utilicen técnicas de reconocimiento óptico de caracteres (OCR). La línea de productos (relacionados con la identificación vehicular) más reconocidos de esta compañía es SeeCar.

La línea de productos SeeCar está basada en los sistemas de Reconocimiento de Matrículas que detectan y leen la matrícula de los vehículos para el control de acceso de los mismos en parqueos, para

² DLVA: Agencia que mantiene el registro de los vehículos y los conductores de Inglaterra

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 1

el tráfico vehicular y aplicaciones de seguridad. Como parte de esta línea de productos se destacan SeeLane, SeeWay y SeeRoad.

Como se muestra en la figura 1 el producto SeeLane sirve como un sistema de control de acceso, controlando la entrada y salida por determinada puerta, almacenando información referente a los eventos que se producen en la misma, así como la imagen del vehículo y del conductor.



Fig.1 Utilización del sistema SeeLane

SeeWay es un sofisticado producto basado en Reconocimiento de Matrículas que identifica y captura el número de la matrícula de los vehículos que viajan desde bajas a altas velocidades. El sistema puede integrar múltiples carriles en las autopistas y múltiples cámaras por carril en un simple sistema de PC estándar.

Este sistema SeeRoad puede ser usado para monitorear el tráfico de los vehículos de un lado de la carretera o carril de una autopista, con el objetivo de alertar para la detección de vehículos buscados, u otros propósitos. Se puede usar incluso para leer el número de la matrícula en una cinta de video para un posterior análisis.

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 1

Cell2solutions

La compañía Cell2solutions (4) 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. Entre los productos que oferta esta compañía relacionados con la identificación de la matrícula se encuentran MaxVision VP y MaxVision CA.

MaxVision VP (ver figura 2) es una solución para la lectura de matrículas desde cualquier vehículo, que funciona con cámaras motorizadas ocultas o visibles. Fiable en vehículos que circulen a alta velocidad y en todo tipo de condiciones atmosféricas.



Fig. 2 Utilización del sistema MaxVision VP

MaxVision CA es una solución de reconocimiento automático de matrículas para el control de accesos a aparcamientos. MaxVision CA es la solución perfecta para organismos públicos que buscan una herramienta simple y eficaz.

Adaptive Recognition Hungary Inc

La compañía Adaptive Recognition Hungary Inc (5) fue fundada en el año 1991. Su sede central radica en Budapest, Hungría. El principal producto de la compañía es CARMEN FreeFlow. Este producto está orientado hacia la recolección automática de peaje, el control de acceso y el monitoreo del tráfico en las autopistas.

1.1.2 Sistemas Nacionales

La información referente al reconocimiento automático de matrículas es prácticamente nula ante el poco desarrollo que tiene el país en este tema pues estos sistemas a nivel mundial son muy costosos y no tiene los recursos para su adquisición. Aunque en Cuba existe un sistema de este tipo de origen ruso en algunos puntos de Ciudad de la Habana.

1.2 Principales herramientas y tecnologías utilizadas.

Constituye un objetivo fundamental de los diseñadores de software alcanzar y mantener un nivel técnico acorde con el desarrollo actual en la automatización de la información para la gestión de cualquier proceso a desarrollar, para lo cual es necesario hacer un estudio detallado de las tecnologías a utilizar y a las posibilidades de desarrollo que estas brindan, así como los conceptos ligados a estas. A continuación en este epígrafe se describe los principales conceptos, tecnologías y herramientas propuestas para el desarrollo de la solución tratada en el trabajo.

1.2.1 Tecnología .Net

1.2.1.1 ASP.Net

ASP.NET es una tecnología para aplicaciones web desarrollada y comercializada por Microsoft. Es usada por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 1

Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

ASP.NET es una de las piezas centrales de Microsoft .NET Framework y proporciona la infraestructura para aplicaciones Web .NET dinámicas desarrolladas fácilmente. ASP.NET no sólo sucede a Microsoft Active Server Pages (ASP), es una plataforma de desarrollo Web unificada que proporciona los servicios necesarios a los desarrolladores para generar aplicaciones Web de empresa. ASP.NET proporciona grandes mejoras con respecto a ASP e incluye muchas características nuevas. (6)

La principal desventaja de utilizar el lenguaje de programación ASP.net, de la plataforma .NET, radica en la naturaleza privativa de la plataforma. Para contrarrestar este problema, se adoptó una estrategia de migración hacia la plataforma libre Mono.

1.2.2 Librería de Interfaz de Usuario

La Librería de Interfaz de Usuario (YUI) ³ (7) es un paquete de utilidades y controles, escritos en JavaScript, que facilitan la construcción de aplicaciones interactivas (RIA), usando técnicas DOM, DHTML, y AJAX.

La finalidad de esta librería es facilitar el desarrollo de aplicaciones ricas del lado del cliente (usuario), logrando elementos visuales e interactivos que incluyen CSS.

Todos los componentes YUI han sido desarrollados en Open Source bajo licencia BSD (licencia que permite la utilización del código fuente, incluso para aplicaciones comerciales), y está disponible para todo tipo de uso.

1.2.3 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o Integrated Development Environment (IDE), es un programa compuesto por un conjunto de herramientas para un programador. Un IDE⁴ es un entorno de

³ YUI: Yahoo User Interface

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 1

programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz grafica GUI. (8)

1.2.3.1 Visual Studio 2008

Visual Studio 2008 provee una serie de herramientas para desarrollo, así como características de debugging, funcionalidad en base de datos y características innovadoras para la creación de aplicaciones en una variedad de plataformas. (9)

Visual Studio 2008 también provee la habilidad de poder escoger entre múltiples versiones del Framework con el mismo entorno de desarrollo, así se puede desarrollar cualquiera de las versiones ya sea en .NET Framework 2.0, 3.0 o 3.5, así que soporta un gran variedad de proyectos en la versión X en el mismo entorno de desarrollo.

Visual Studio 2008 ofrece a desarrolladores nuevas herramientas para la fácil creación de aplicaciones conectadas en las ultimas plataformas incluyendo web, Windows Vista, Office 2007, SQL Server 2008 y Windows Server 2008. Para la web, ASP.NET, AJAX y otras tecnologías como Silverlight, WPF, etc. que dará la posibilidad de crear aplicaciones con rica interfaz de usuario, para poder dar una experiencia de usuario sin precedentes. (10)

1.2.4 Gestor de Base de Datos

Un sistema gestor de base de datos es una aplicación capaz de manejar un conjunto de datos de manera eficiente y cómoda, asegurando su integridad, confidencialidad y seguridad.

1.2.4.1 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 todos los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Es completamente conforme al estándar de Atomicidad,

⁴ IDE: entorno de desarrollo integrado

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 1

Consistencia, Aislamiento y Durabilidad (ACID)⁵. 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 incluyendo enteros, numéricos, booleanos, character, varchar, fecha, intervalo, y timestamp. También apoya el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o vídeo. Posee interfaces de programación nativas para C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otras y muy buena documentación.

PostgreSQL cuenta con sofisticadas características tales como la versión multi-control de concurrencia (MVCC)⁶, 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: (11)

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

⁵ Atomicity, Consistency, Isolation and Durability

⁶ Multi-version Concurrency Control

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 1

PostgreSQL es sin duda uno de los gestores de base de datos libres más empleados debido a las ventajas y la seguridad de almacenamiento de información que este proporciona.

1.2.5 Controlador de Versiones

1.2.5.1 Microsoft Visual SourceSafe 2005

Microsoft Visual SourceSafe es un sistema de control de versiones en el nivel de archivos, que permite a muchos tipos de organizaciones trabajar en distintas versiones de un proyecto al mismo tiempo. Esta funcionalidad es especialmente ventajosa en un entorno de desarrollo de software, donde se usa para mantener versiones de código paralelas. Sin embargo, el producto también se puede utilizar para mantener archivos en cualquier otro tipo de equipo.

Ventajas

Para las personas que desarrollan programas bajo el sistema operativo Windows, resulta una herramienta útil ya que se integra fuertemente con el entorno de desarrollo integrado o IDE de Visual Studio permitiendo un manejo relativamente simple de versiones sobre una computadora individual y en equipos de trabajo relativamente pequeños.

Desventajas

El SourceSafe es configurable, permitiendo que un solo programador modifique el código fuente (recomendado) o que lo hagan varios. Las herramientas de gestión de diferencias para reunificar el código fuente modificado por varios programadores no son demasiado buenas comparadas con las de otros gestores de código fuente. Es inestable cuando se suben ficheros binarios de gran tamaño, ya que espera solo ficheros de texto. Así que no vale para almacenar documentación, sólo código fuente.

Visual SourceSafe incluye, como mínimo, las siguientes funciones:

- Ayuda al equipo a evitar la pérdida accidental de archivos.
- Permite realizar un seguimiento de las versiones anteriores de un archivo.

- Admite la bifurcación, el uso compartido, la combinación y la administración de versiones de archivos.
- Realiza el seguimiento de las versiones de proyectos completos.
- Realiza el seguimiento del código modular (un archivo que se reutiliza, o se comparte, en varios proyectos). (12)

1.2.6 Herramienta Case

1.2.6.1 Enterprise Architect 7.0

Enterprise Architect 7.0 es una herramienta gráfica multi-usuario, basado en Windows, diseñada para ayudar a construir software robusto y confiable. Cuenta con una gran flexibilidad y documentación de alta calidad. Enterprise Architect combina el poder de la última especificación UML 2.1 con un alto rendimiento, interfaz intuitiva, usa perfiles UML para extender el dominio de modelado, mientras que la validación de modelos asegura integridad. (13)

1.2.7 Lenguaje de Modelación

1.2.7.1 Lenguaje Unificado de Modelado

UML (por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. (14)

1.2.8 Metodología de Desarrollo

1.2.8.1 RUP

En la investigación, la generación de los artefactos estará guiada por los principios del Proceso Unificado de Desarrollo (RUP, por sus siglas en idioma Inglés). Dicha metodología está basada en componentes y utiliza el Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema software. (15)

RUP⁷ 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. Dirigido 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, y es iterativo e incremental, donde divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada. (16)

Todos los aspectos, mencionados anteriormente, permitirán obtener artefactos de calidad óptima, realizando un esfuerzo mínimo.

1.3 Justificación de la elección de las herramientas y tecnologías utilizadas

Por las características de las herramientas y metodologías antes expuestas se consideró la utilización de las mismas en la elaboración de este proyecto. La potencialidad, prestaciones y fácil manejo al interactuar con ellas son una de las principales características que sobresalen en estas herramientas. Cada una de ellas está entre las más usadas a nivel mundial, se han ganado un sitio de honor dentro de los programadores por la serie de ventajas y facilidades que ofrece a pesar de que algunas cuentan con varias limitaciones, como es el tema de las licencias para algunas. Además se decidió utilizar estas herramientas en el desarrollo del sistema como cumplimiento de los lineamientos tecnológicos establecidos en el Grupo de Procesamiento de Imágenes (GPI).

En este capítulo se han analizado los principales sistemas de reconocimiento automático de matrículas, así como las distintas herramientas, metodologías y lenguajes de programación que se utilizan en el desarrollo del sistema.

⁷ Rational Unified Process

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

La realización de la aplicación conlleva un proceso exhaustivo de ingeniería del software. En este capítulo se describen los procesos estudiados, así como la propuesta del sistema. Además los requerimientos, tanto funcionales como no funcionales, los modelos del negocio y del sistema como parte de la fase de inicio.

2.1 Objeto de automatización

El control vehicular en Cuba se realiza principalmente al acceder a determinadas instalaciones. Existe personal de seguridad que se encarga de registrar el acceso de los vehículos a las mismas. Por mucho que el personal se esfuerce es inevitable la pérdida de los registros de datos, pues resulta difícil almacenar información que se acumula con el tiempo. Además no existe un proceso de gestión de información que permita llevar a cabo esta tarea sin mayores contratiempos. Eso sin contar el esfuerzo humano que requiere.

A simple vista el no almacenamiento de información producto al acceso de los vehículos en las instalaciones no acarrea grandes problemas, pero es de considerar la gran utilidad de la misma ante la ocurrencia de actos delictivos.

Si ocurriese un hecho de este tipo, resultaría necesario buscar en los registros de datos toda información que permita o ayude a aclarar el hecho. En las condiciones actuales, como se puede ver, este sistema de control de acceso resulta ineficiente. Con la realización del sistema **Cugnot Explorer** se pretende automatizar el proceso de búsqueda de información, agilizando el acceso a los datos requeridos.

2.2 Propuesta de Sistema

El sistema propuesto consiste en una aplicación web que permita la visualización de los datos de la Base de Datos de tráfico de la **Suite Cugnot**. Debe permitir la realización de búsquedas empleando la combinación de los criterios matrícula, punto de control y fecha y hora de un período. El usuario puede conocer además los puntos de control registrados en la base de datos, los servidores de imágenes

existentes y la cantidad de entradas asociadas a los mismos. Debe permitir además generar reportes asociados a las búsquedas, así como la visualización de las imágenes de los vehículos. No es necesario autenticarse.

2.3 Descripción del Negocio

Si un vehículo arriba a un punto de control de una instalación el custodio, que es el que se encarga de la seguridad en los puntos de control, consulta el Registro de Vehículos y verifica que el mismo esté autorizado a pasar por ese punto. Si la matrícula es encontrada en el registro, el custodio permite el acceso al vehículo y actualiza el Registro de Entrada indicando la hora en que pasó. Si la matrícula no es encontrada el custodio niega el acceso y el vehículo debe abandonar el punto de control.

2.4 Modelo de Negocio

2.4.1 Actores del Negocio

Un actor representa un conjunto coherente de roles que los usuarios de los casos de uso juegan al interactuar con éstos. Normalmente, un actor representa un rol que es jugado por una persona, un dispositivo hardware o incluso otro sistema. (17)

Actores del negocio	Justificación
Vehículo	Representa al que interactúa con las actividades del negocio.

2.4.2 Trabajadores del Negocio

Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Trabajadores del negocio	Justificación
Custodio	Responsable de realizar las actividades del negocio, en

	este caso, de permitir el acceso por un punto de control.
--	---

2.4.3 Diagrama de CU del Negocio

Un diagrama de CU del Negocio representa la relación entre los actores del negocio y los procesos del negocio que constituyen las principales actividades que se realizan.

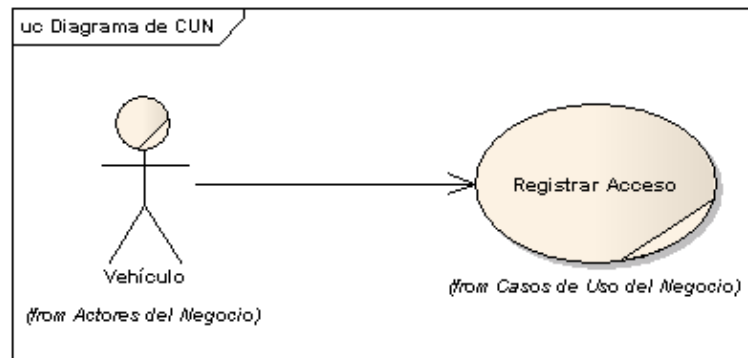


Fig. 3 Diagrama de CU del Negocio

2.4.4 Diagrama de Clases del Modelo de Objetos

El Modelo de Objetos es un diagrama de clases que representa la relación entre los trabajadores del negocio y las entidades. Estas representan contenedores de información de larga duración. A continuación se representa del Modelo de Objetos del negocio en cuestión.

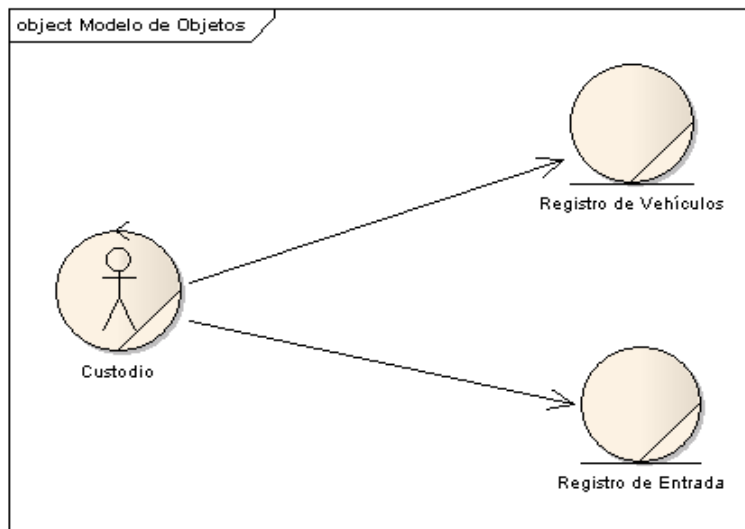


Fig. 4 Modelo de Objetos

2.4.5 Diagrama de Actividades <Registrar Acceso>

El diagrama de actividades representa la secuencia de actividades que se ejecutan durante la realización del flujo de trabajo. A continuación se representa el Diagrama de Actividades enfatizando en la actividad a automatizar.

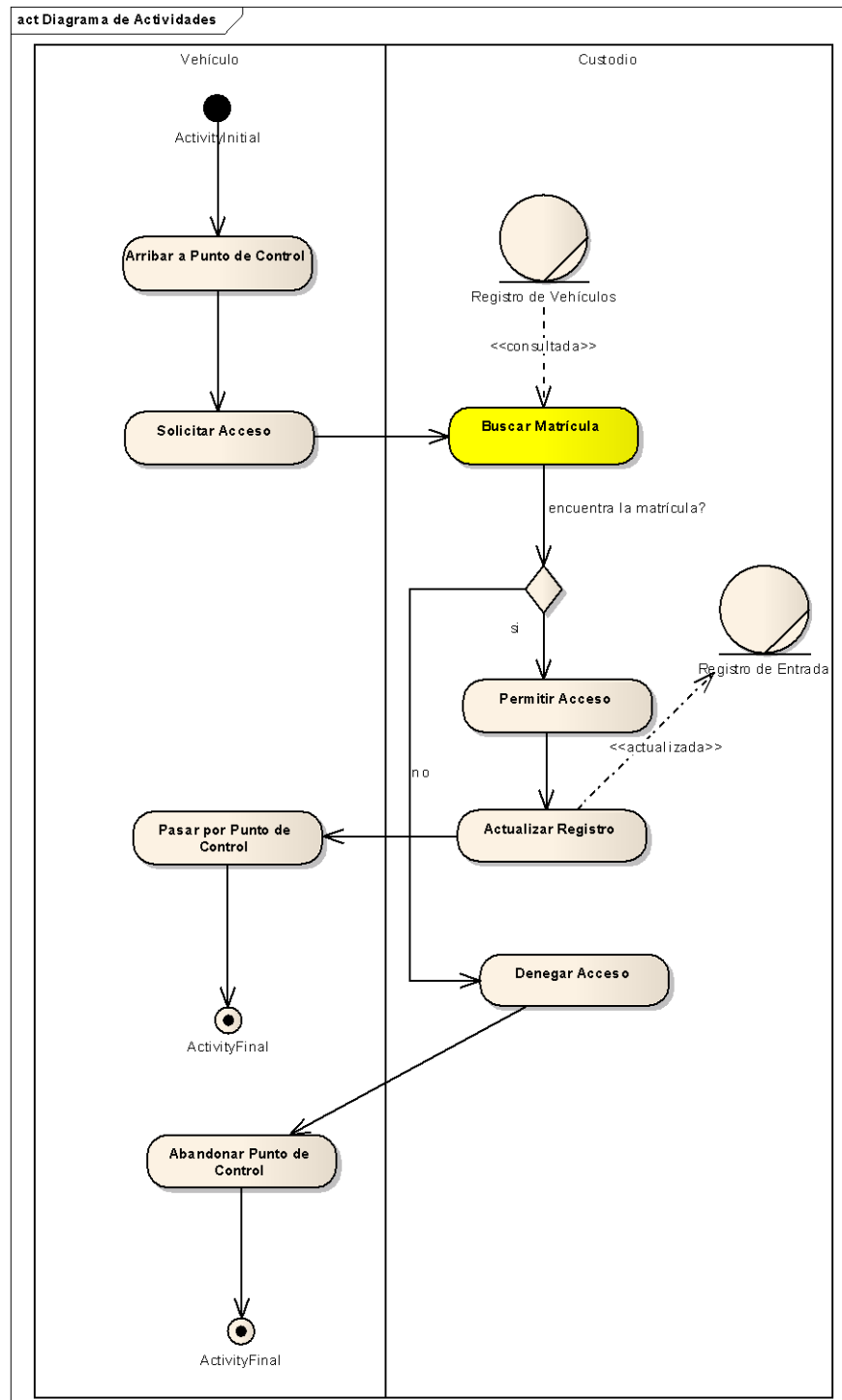


Fig. 5 Diagrama de Actividades

2.5 Especificación de los requisitos del software

Los requerimientos son una descripción de las necesidades o deseos de un producto. La meta principal en esta etapa es identificar y documentar lo que en realidad se necesita, en una forma en que pueda fácilmente ser transmitido al cliente y al equipo de desarrollo. (18)

2.5.1 Requerimientos Funcionales

Los requerimientos funcionales identifican las capacidades o condiciones que el sistema debe cumplir. A continuación se listan los requerimientos que debe cumplir el sistema.

Identificador	Descripción
RF1 ⁸	El sistema debe permitir la realización de búsquedas empleando la combinación de las matrículas, los puntos de control, la fecha de inicio y terminación de un período y la hora de inicio y fin del período.
RF2	Debe mostrar los puntos de control registrados en la base de datos y la cantidad de entradas asociadas a ellos.
RF3	Debe permitir mostrar los servidores de imágenes existentes en la base de datos y la cantidad de entradas asociadas a ellos.
RF4	Debe permitir la generación de reportes a partir de las búsquedas.
RF5	Debe permitir la visualización de la imagen de un vehículo.
RF6	Debe permitir la aplicación del efecto lupa a la imagen de un vehículo.

2.5.2 Requerimientos no Funcionales

Los requerimientos no funcionales identifican propiedades o cualidades que el producto debe tener.

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

⁸ Requerimiento Funcional

Rendimiento

Identificador	Descripción
RNF1 ⁹	El sistema debe permitir hacer búsquedas de manera rápida y eficiente y debe ser posible buscar por múltiples criterios.
RNF2	Las tareas de búsqueda y obtención deben realizarse de forma asincrónica.

Portabilidad

Identificador	Descripción
RNF3	EL sistema debe ser compatible con la plataforma libre MONO.

Legales

Identificador	Descripción
RNF4	La aplicación, junto con toda la información generada durante el desarrollo de la misma pertenece a la Universidad de la Ciencias Informáticas.

Interfaz

Identificador	Descripción
RNF5	La interfaz de la aplicación debe ser amigable y lo más sencilla posible.

Ayuda y documentación en línea

⁹ Requerimiento no Funcional

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 2

Identificador	Descripción
RNF6	La aplicación debe ofrecer una ayuda.

Software

En el Servidor

Identificador	Descripción
RNF7	NET Framework v2.0.
RNF8	Internet Information Service (IIS) v5 o superior.
RNF9	PostgreSQL v8.2.

2.6 Definición de los Casos de Uso del Sistema

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. Proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema.

Además ayudan a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo. Por lo general el nombre de un caso de uso comienza con un verbo en infinitivo. Un caso de uso describe un proceso de principio a fin, es decir, una secuencia de eventos, las acciones y las transacciones que se requieren para realizarlo. (19)

2.6.1 Definición de los actores

Actores	Justificación
---------	---------------

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 2

Usuario	Es el que interactúa con el sitio explorando de esta forma la base de datos.
---------	--

2.6.2 Listado de Casos de Uso

CU-1	Realizar Búsquedas
Actor	Usuario(inicia)
Descripción	El caso de uso se inicia cuando el actor entra los parámetros necesarios para realizar las búsquedas. Seguidamente el sistema muestra en un listado el resultado de la búsqueda.
Referencia	RF1

CU-2	Mostrar Propiedades
Actor	Usuario(inicia)
Descripción	El caso de uso se inicia cuando el actor selecciona la opción para mostrar las propiedades. El sistema muestra las propiedades.
Referencia	RF2, RF3

CU-3	Generar Reportes
Actor	Usuario(inicia)
Descripción	El caso de uso se inicia cuando el actor selecciona la opción generar reportes. El sistema permite la generación del reporte.
Referencia	RF4

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 2

CU-4	Visualizar Imagen
Actor	Usuario(inicia)
Descripción	El caso de uso se inicia cuando el actor selecciona la opción para visualizar una imagen. El sistema muestra la imagen.
Referencia	RF5

CU-5	Aplicar efecto lupa.
Actor	Usuario(inicia)
Descripción	El caso de uso se inicia cuando el actor aplica el efecto sobre la imagen. El sistema muestra la sección que el usuario escoge para el efecto más grande.
Referencia	RF6

2.6.3 Diagrama de Casos de Uso del Sistema

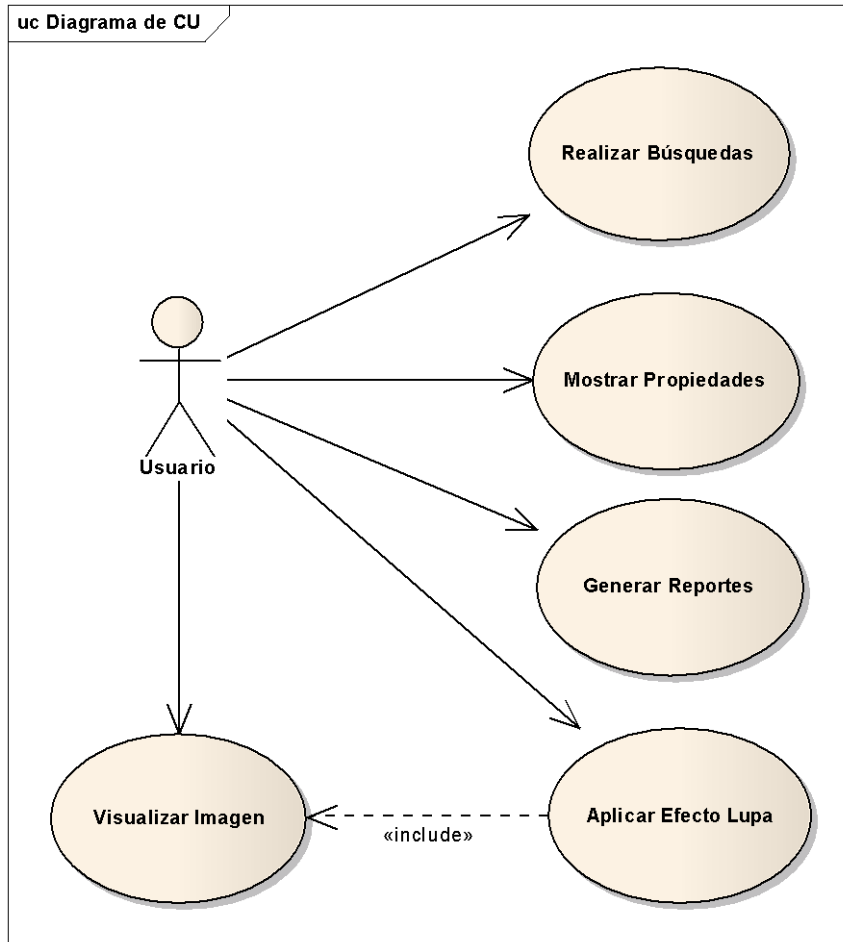


Fig. 6 Diagrama de Casos de Uso

2.6.4 Casos de Uso por Ciclo de desarrollo.

Ciclo1

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Realizar Búsquedas	Mostrar Información	Es el caso de uso más significativo pues permite a partir de la entrada de datos por parte del usuario obtener las principales

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 2

			características de determinado vehículo.
--	--	--	--

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-2	Mostrar Propiedades	Mostrar Información	Muestra al usuario las propiedades referentes a la cantidad de entradas existentes en la base de datos, los puntos de control registrados y las imágenes existentes.

Ciclo 2

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-3	Generar Reportes	Generar Reportes	Permite al usuario generar reportes.

Ciclo 3

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-4	Visualizar Imagen	Imagen	Permite al usuario visualizar la imagen de una matrícula.

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-5	Aplicar Efecto Lupa	Imagen	Permite al usuario aplicar efecto lupa sobre una imagen.

2.6.5 Expansión de los Casos de Uso del Sistema

Los casos de uso expandidos son muy útiles para alcanzar un conocimiento más profundo de los procesos y de los requerimientos. Constituyen un documento narrativo que describe la secuencia de eventos de un actor que utiliza el sistema para completar un proceso. (20)

Ver casos de uso expandidos en [Anexo 1](#).

En el capítulo, a partir de los criterios del cliente sobre el sistema y de como debe quedar gráficamente se realizó el levantamiento de requisitos del sistema. Por lo que se está en condiciones de terminar con la fase de inicio de la ingeniería de software. Se han descrito los procesos del negocio que interviene en la realización del sistema. Se detallaron los actores participantes del mismo, sus funciones y su interacción con la aplicación. Se brindó una descripción de los requisitos, tanto funcionales como no funcionales. Además se describieron los casos de uso que participan en la elaboración del software tanto del negocio como del sistema.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

Al finalizar la fase de inicio, se hace necesario diseñar el sistema propuesto pues es la clave para una óptima implementación del mismo.

El flujo de trabajo de Análisis y Diseño del sistema tiene como objetivo transformar los requisitos en el diseño del futuro sistema, creando un plano del modelo de implementación. (21)

3.1 Patrones de Arquitectura

Un estilo arquitectónico o variante arquitectónica define a una familia de sistemas informáticos en términos de su organización estructural. Un estilo arquitectónico describe 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

3.1.1 Arquitecturas en Capas

Este patrón define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque 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, 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.

3.1.1.2 Arquitectura en Tres Capas.

La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido (ver figura 7).

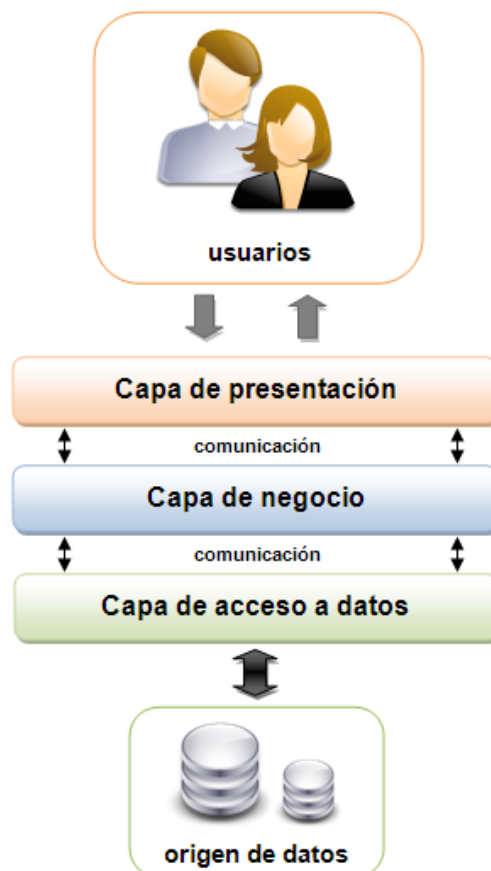


Fig. 7 Arquitectura de tres capas.

Cugnot Explorer: Explorador del Repositorio de Datos

Capítulo 3

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.

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.

La solución Cugnot Explorer presenta una arquitectura en tres capas. A continuación se brinda una panorámica de cómo están distribuidas las páginas, clases e interfaces del sistema en cuestión (ver figura 8).

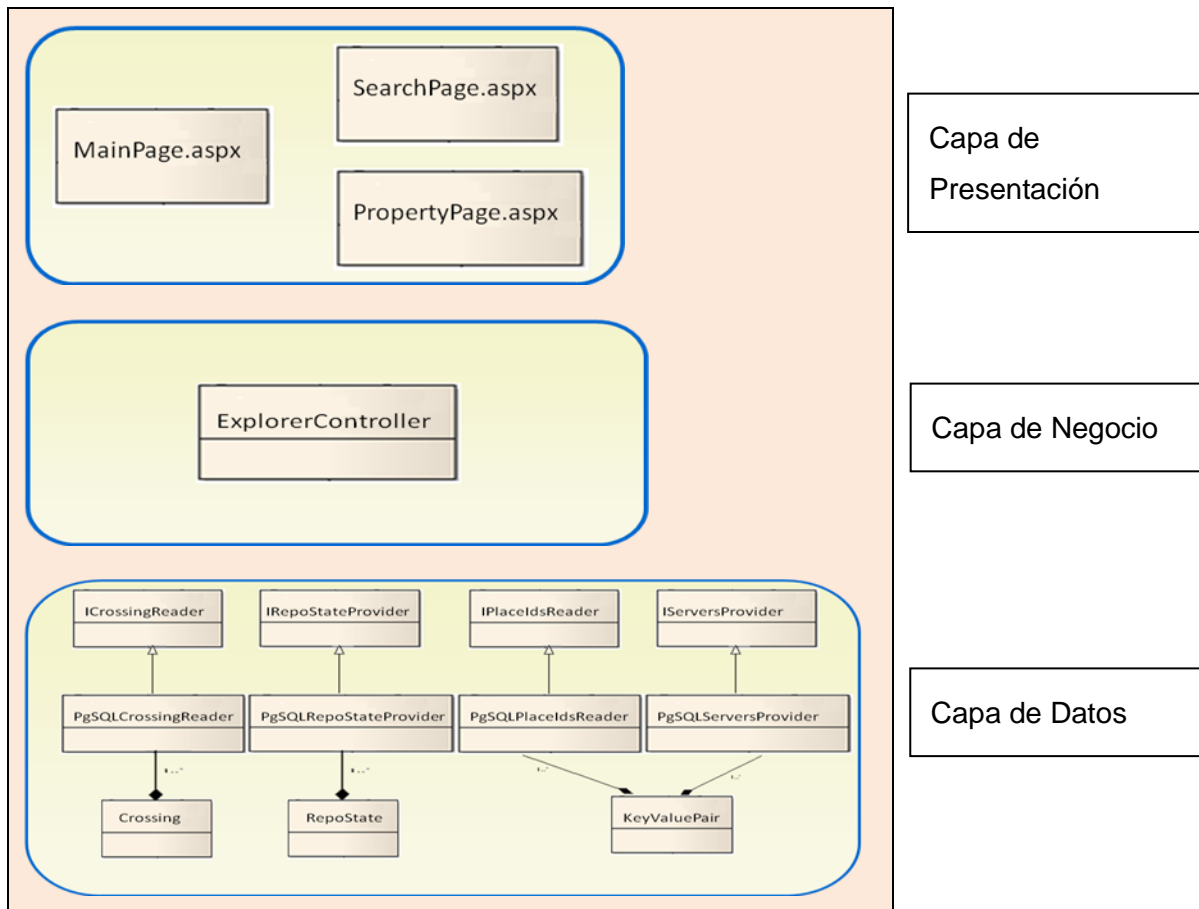


Fig. 8 Arquitectura de tres capas.

3.2 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 que captura la funcionalidad del sistema, el Modelo de Análisis da forma a la arquitectura para soportar las funcionalidades que en el anterior modelo se expresa. Es netamente Ingenieril. (22)

3.2.1 Diagrama de Clases del Análisis

En el Diagrama de Clases de Análisis se representan las clases participantes en la realización de los casos de usos así como sus relaciones entre ellas y con los actores. (23)

Seguidamente se muestran los Diagramas de Clases del Análisis de los casos de uso más significativos Realizar Búsquedas y Mostrar Propiedades. El resto de estos diagramas se pueden ver en el [Anexo 2](#).

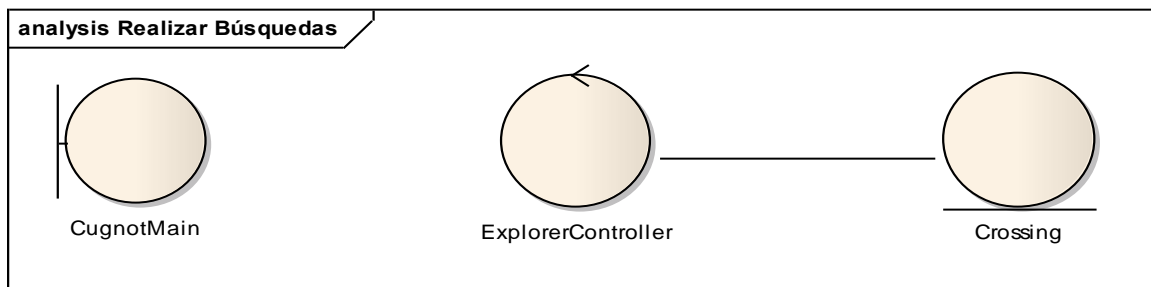


Fig. 9 Diagrama de Clases "Realizar Búsquedas"

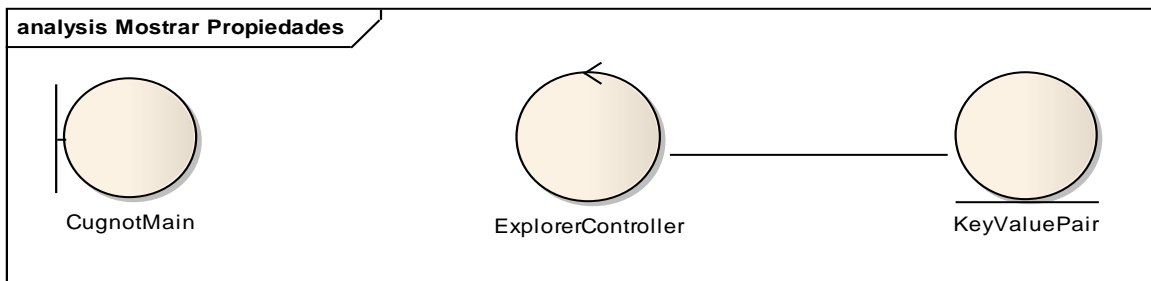


Fig. 10 Diagrama de Clases "Mostrar Propiedades"

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. Esto contribuye a una arquitectura estable y sólida, y crear un plano del modelo de

implementación. Durante la fase de construcción, cuando la arquitectura es 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 se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos. Además, impone una estructura del sistema que debe ser conservado lo más fielmente posible cuando se conforme al sistema. (24)

3.3.1 Diagramas de Clases del Diseño

El Diagrama de Clases del Diseño ofrece una vista estática de las definiciones de las clases participantes en el software.

A continuación se muestran los Diagramas de Clases del Diseño de los casos de usos más significativos Realizar Búsquedas y Mostrar Propiedades. El resto de los diagramas se pueden ver en el [Anexo 3](#).

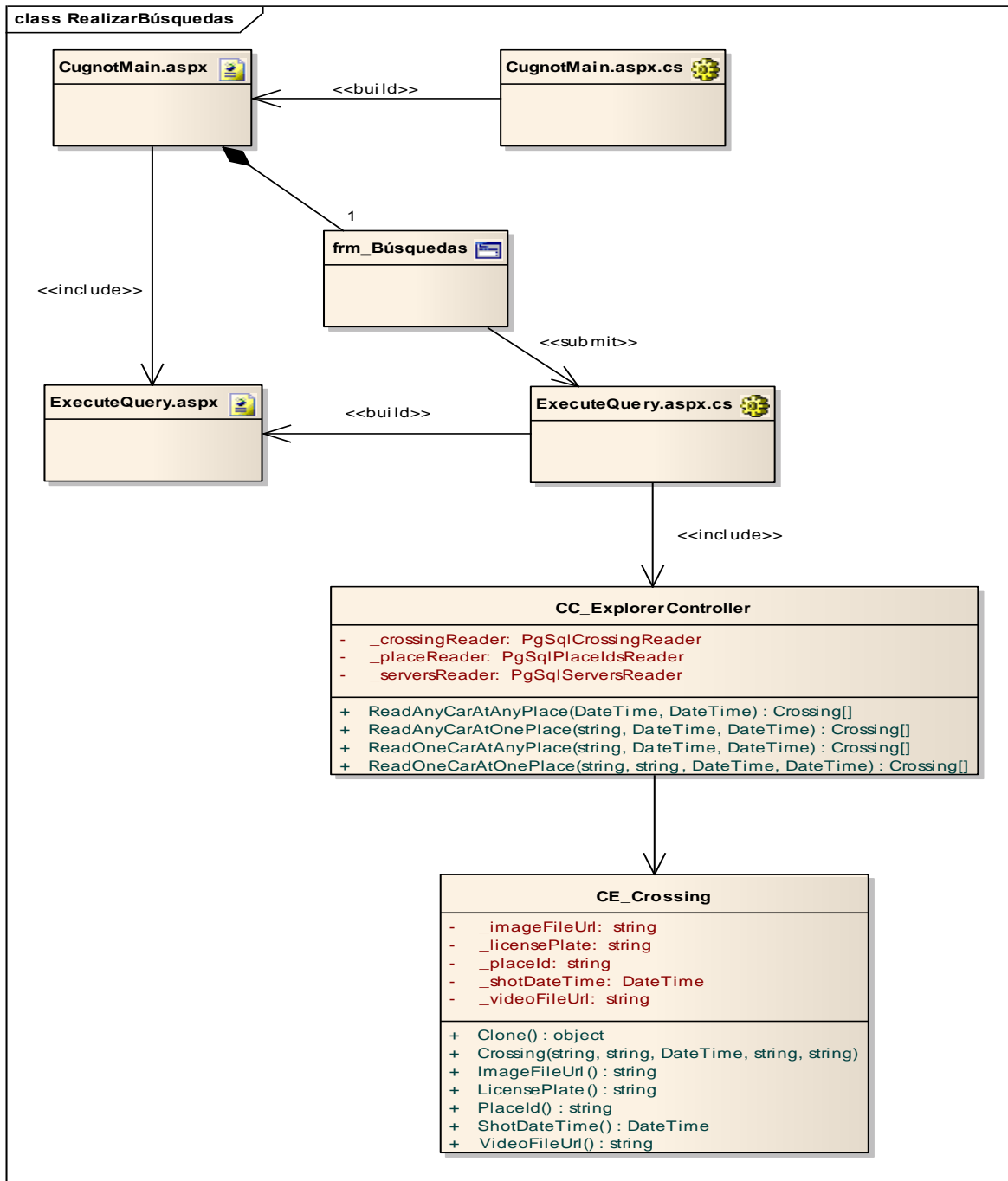


Fig. 11 Diagrama de Clases "Realizar Búsquedas"

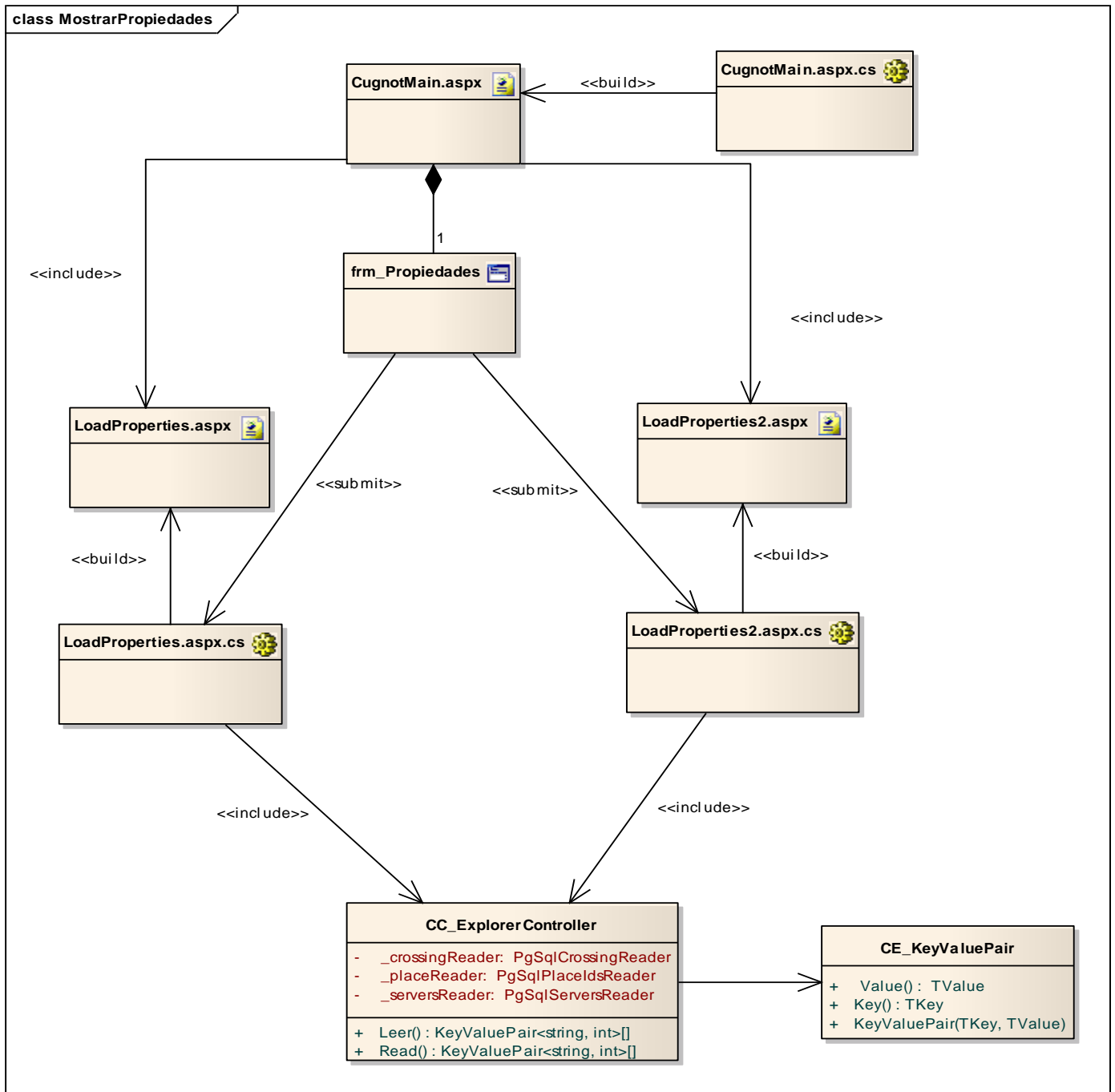


Fig. 12 Diagrama de Clases "Mostrar Propiedades"

3.3.2 Diagramas de Interacción del Diseño

Mediante el Diagrama de Interacción se describe la secuencia de intercambios de mensajes entre los objetos para realizar las tareas de una manera gráfica.

En las figuras 14 y 15 se muestran los Diagramas de Interacción de los casos de uso Realizar Búsquedas y Mostrar Propiedades. El resto de los diagramas se pueden apreciar en el [Anexo 4](#).

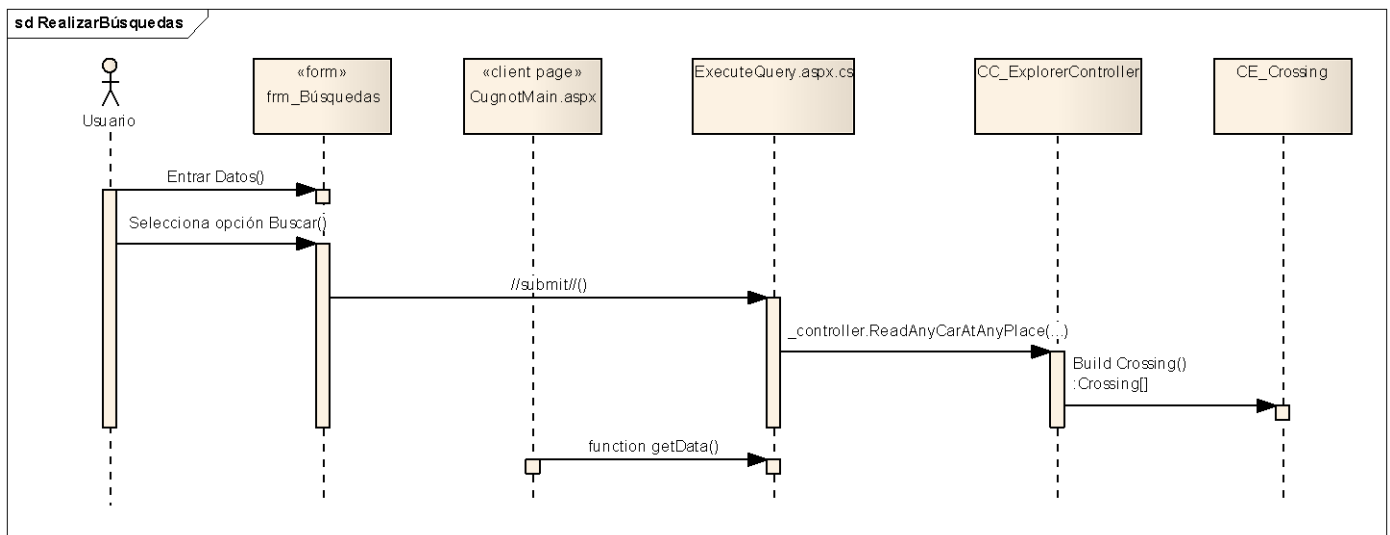


Fig. 13 Diagrama de Interacción "Realizar Búsquedas"

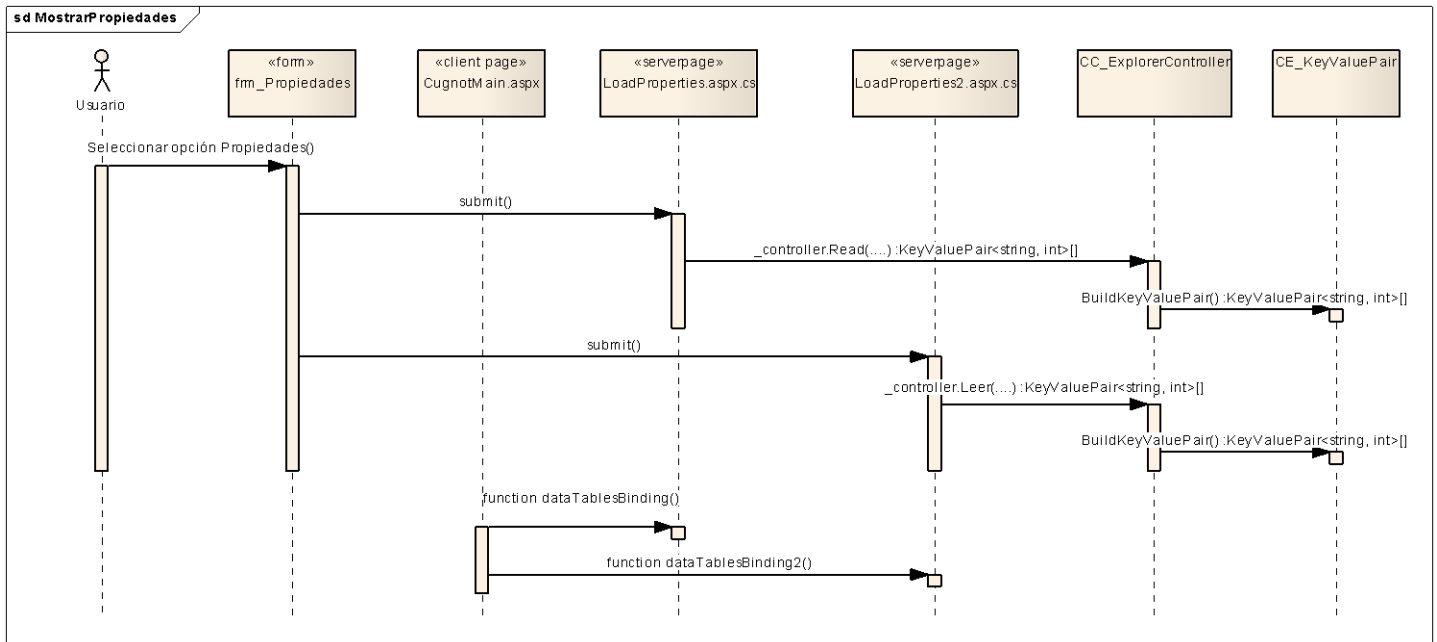


Fig. 14 Diagrama de Interacción "Mostrar Propiedades"

3.4 Descripción de las Clases

Con la descripción de las clases se obtiene una información más detallada sobre las clases que están implicadas, pues se conoce sus atributos, así como las operaciones que las hacen únicas de acuerdo a su responsabilidad dentro del sistema.

En el [Anexo 5](#) se pueden apreciar las descripciones de las clases utilizadas, así como las páginas principales del sitio.

3.5 Modelo de Datos

El Modelo Entidad Relación de la aplicación se muestra a continuación (ver figura 16).

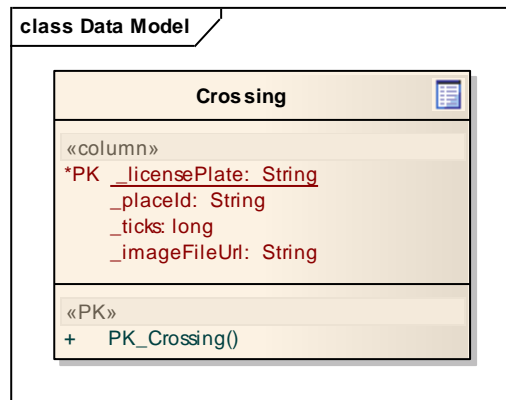


Fig. 15 Tabla Crossing.

En este capítulo se desarrolló uno de los flujos más importantes durante el proceso de ingeniería del software, el Análisis y Diseño donde se obtuvo un plano del modelo de implementación. Se logró establecer la arquitectura candidata basada en la realización de casos de uso y centrada en los requerimientos funcionales y no funcionales.

CAPÍTULO IV: 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. (25)

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. (26)

4.1 Diagrama de Despliegue

Los diagramas de despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

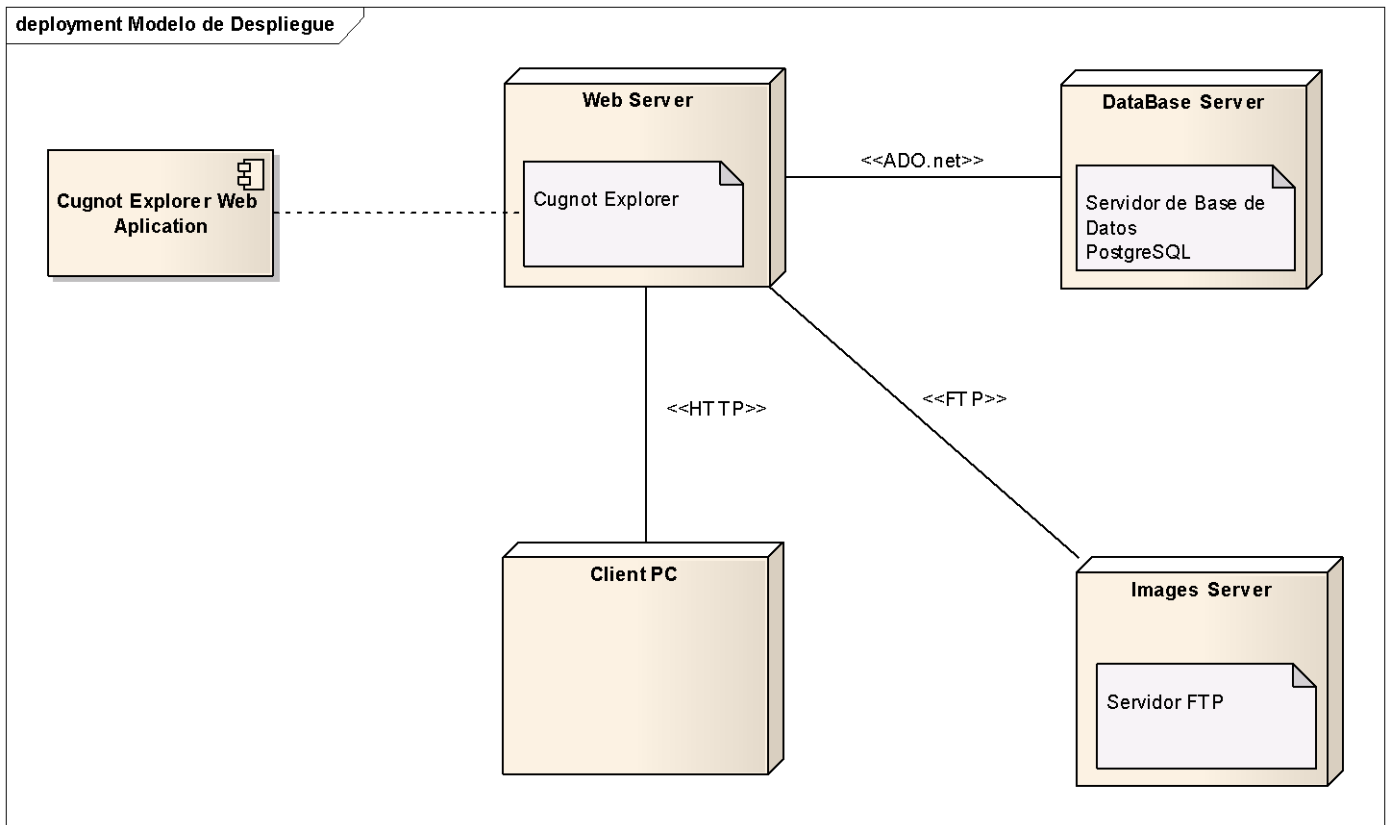


Fig. 16 Diagrama de Despliegue

4.2 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de

dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

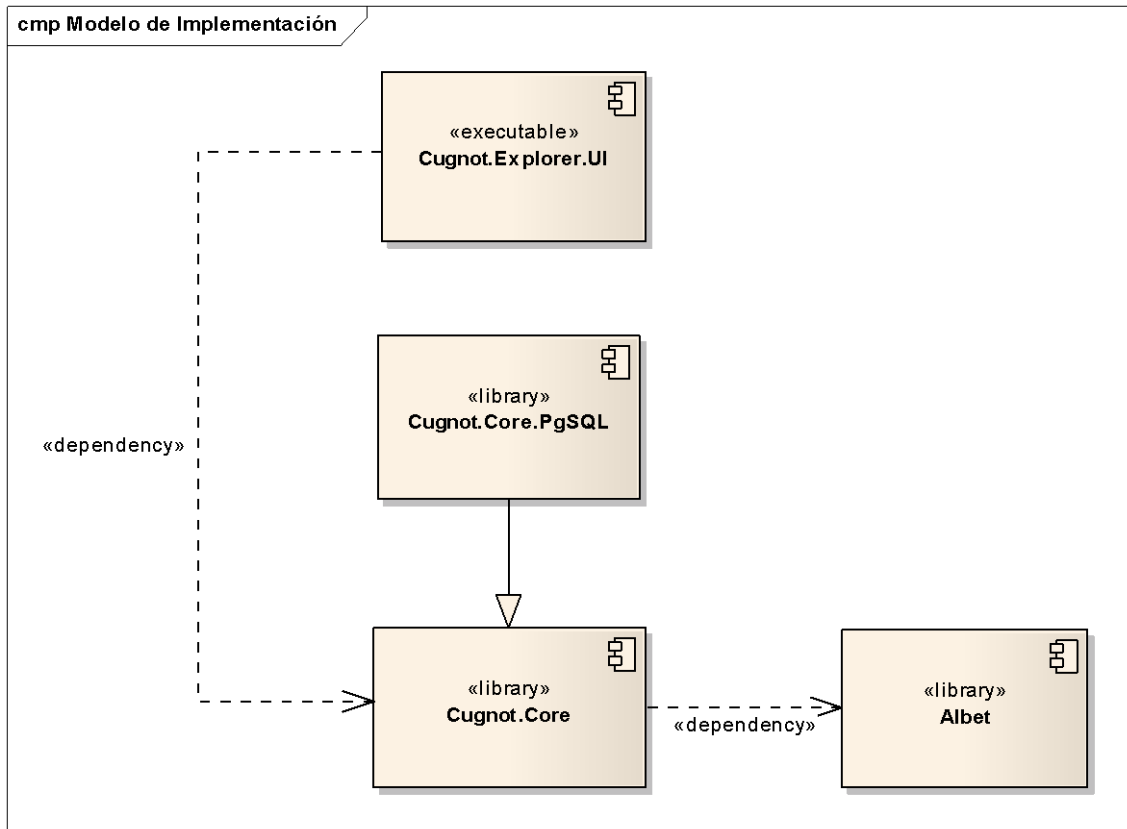


Fig. 17 Diagrama de Componentes

En este capítulo se mostró el diagrama de despliegue, donde se ve la relación que hay entre los componentes del sistema. También se vio el diagrama de componente donde se mostró la relación que hay entre los elementos que conforman el software.

CONCLUSIONES

Se implementó la aplicación web Cugnot Explorer, que es el componente que facilita la visualización de los datos de la **suite Cugnot**.

La realización de este trabajo hizo posible la automatización del proceso de visualización de la información de la Base de Datos de tráfico, agilizando el acceso a la información requerida.

Concretamente, Cugnot Explorer permite:

- Realizar búsquedas atendiendo a los criterios de matrícula, punto de control e inicio y fin de un período.
- Mostrar los servidores de imágenes existentes y la cantidad de entradas asociadas a ellos.
- Mostrar los puntos de control existentes y la cantidad de entradas asociadas a ellos.
- Generar reportes a partir de las búsquedas.
- Visualizar las imágenes de vehículos existentes en el servidor.
- Aplicar efecto lupa a las imágenes.

Cugnot Explorer como parte de la **suite Cugnot** contribuye a mejorar el control vehicular existente actualmente en el país con el mínimo de costo posible.

RECOMENDACIONES

Se recomienda seguir trabajando en otras versiones de Cugnot Explorer con el objetivo de hacer un sistema más potente y robusto con nuevas funcionalidades haciendo énfasis en la seguridad del sitio.

Se recomienda además continuar el estudio de la librería YUI¹⁰ pues la misma constituye una rica fuente de funcionalidades que permiten realizar un sitio más interactivo y dinámico.

¹⁰ Yahoo User Interface

REFERENCIAS

1. Computer Recognition Systems Ltd. *Computer Recognition Systems Ltd.* [Online] <http://www.crs-traffic.co.uk/index.html>.
2. Anpr International Ltd. *Anpr International Ltd.* [Online] <http://www.anpr-international.com/anpr-technology-c3.html>.
3. Hi-tech solutions. *Hi-tech solutions.* [Online] <http://www.htsol.com/Products/SeeCar.html> .
4. Cell2solutions. *Cell2solutions.* [Online] <http://www.cell2solutions.com/en/anpr.php> .
5. Adaptive Recognition Hungary Inc . *Adaptive Recognition Hungary Inc .* [Online] <http://www.arhungary.hu/> .
6. DevelopersDotNet.com. *DevelopersDotNet.com.* [Online] diciembre 2008. <http://developersdotnet.com/wikis/aspnet/191-qu-233-es-asp-net.aspx>.
7. Yahoo!Developer Network. *Yahoo!Developer Network.* [Online] diciembre 2008. <http://developer.yahoo.com/yui/>.
8. Entorno de desarrollo integrado. *Wikipedia, la enciclopedia libre.* [Online] http://es.wikipedia.org/wiki/Ambiente_integrado_de_desarrollo.
9. Visual Studio 2008 Development System. *msdn.* [Online] <http://msdn.microsoft.com/es-es/vstudio/products/bb931331.aspx>.
10. Características de Visual Studio. *Tecnologias Microsoft.* [Online] <http://mredison.wordpress.com/2007/12/02/caractersticas-de-visual-studio-2008/>.
11. **Durañona, Yanoksy Yero and Rodriguez, Lazaro Gonzalez.** PostgreSQL. *Cassandra Server.* Ciudad de la Habana : s.n., 2007.
12. Introducing Visual Sourcesafe. *msdn.* [Online] [http://msdn.microsoft.com/en-us/library/3h0544kx\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/3h0544kx(VS.80).aspx).

Cugnot Explorer: Explorador del Repositorio de datos

Referencias Bibliográficas

13. Enterprise Architect. *Enterprise Architect*. [Online]
<http://www.sparxsystems.com/products/ea/index.html>.
14. Lenguaje Unificado de Modelado. *Wikipedia, la enciclopedia libre*. [Online]
http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado.
15. **Jacobson, Ivar, Grady Booch, y James Rumbaugh**. *El Proceso Unificado de Desarrollo*. s.l. : Addison Wesley Logman, 2000.
16. **Galves, Jorge**. *Fundamentos de la metodología RUP, Rational Unified Process*. [Online] Septiembre 16, 2007. [Cited: Noviembre 20, 2008.] <http://www.scribd.com/doc/297224/RUP>.
17. **Booch, Grady and RUMBAUGH, J**. *UML: El lenguaje Unificado de modelado*. s.l. : Addison 2000.
18. Requerimientos Funcionales y no Funcionales. *Tecnologico*. [Online]
<http://www.mitecnologico.com/Main/RequerimientosFuncionalesYNoFuncionales>.
19. **Naranjo, Alejandro Arias**. Definición de los Casos de Uso. *CassandraWeb*. Ciudad de la Habana : s.n., 2007.
20. **Larman, C**. *UML y Patrones 2*. s.l. : Addison Wesley Professional, 2002.
21. **Andricain, Augusto Meilan and Denis, Yordanis Marquez**. Análisis y Diseño del Sistema. *Cugnot Server*. Ciudad de La Habana : s.n., 2009.
22. **Jacobson, Ivar, Booch, Gardy and Rumbaugh, James**. Análisis. *El prodeso Unificado de desarrollo de Software*. Madrid : s.n., 2000.
23. **Naranjo, Alejandro Arias**. Diagrama de Clases del Análisis. *CassandraWeb*. Ciudad de la Habana : s.n., 2007.
24. **Andricain, Augusto Meilan and Denis, Yordanis Marquez**. Modelo de Diseño. *Cugnot Server*. Ciudad de la Habana : s.n., 2009.
25. **IBM Maximo Asset Management, IBM Tivoli Asset Management**. *Workflow Implementation Guide*. 2008.
26. **Informáticas, Universidad de Ciencias**. *Flujo de Implementación*. UCI, s.l. : s.n.

BIBLIOGRAFÍA

1. Computer Recognition Systems Ltd. *Computer Recognition Systems Ltd.* [Online] <http://www.crs-traffic.co.uk/index.html>.
2. Anpr International Ltd. *Anpr International Ltd.* [Online] <http://www.anpr-international.com/anpr-technology-c3.html>.
3. Hi-tech solutions. *Hi-tech solutions.* [Online] <http://www.htsol.com/Products/SeeCar.html> .
4. Cell2solutions. *Cell2solutions.* [Online] <http://www.cell2solutions.com/en/anpr.php> .
5. Adaptive Recognition Hungary Inc . *Adaptive Recognition Hungary Inc .* [Online] <http://www.arhungary.hu/> .
6. DevelopersDotNet.com. *DevelopersDotNet.com.* [Online] diciembre 2008. <http://developersdotnet.com/wikis/aspnet/191-qu-233-es-asp-net.aspx>.
7. Yahoo!Developer Network. *Yahoo!Developer Network.* [Online] diciembre 2008. <http://developer.yahoo.com/yui/>.
8. Entorno de desarrollo integrado. *Wikipedia, la enciclopedia libre.* [Online] http://es.wikipedia.org/wiki/Ambiente_integrado_de_desarrollo.
9. Visual Studio 2008 Development System. *msdn.* [Online]
10. <http://msdn.microsoft.com/es-es/vstudio/products/bb931331.aspx>.
11. Características de Visual Studio. *Tecnologias Microsoft.* [Online] <http://mredison.wordpress.com/2007/12/02/caractersticas-de-visual-studio-2008/>.
12. **Durañona, Yanoksy Yero and Rodriguez, Lazaro Gonzalez.** PostgreSQL. *Cassandra Server.* Ciudad de la Habana : s.n., 2007.
13. Introducing Visual Sourcesafe. *msdn.* [Online] [http://msdn.microsoft.com/en-us/library/3h0544kx\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/3h0544kx(VS.80).aspx).

Cugnot Explorer: Explorador del Repositorio de datos

Bibliografía

14. Enterprise Architect. *Enterprise Architect*. [Online]
<http://www.sparxsystems.com/products/ea/index.html>.
15. Lenguaje Unificado de Modelado. *Wikipedia, la enciclopedia libre*. [Online]
http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado.
16. **Jacobson, Ivar, Grady Booch, y James Rumbaugh**. *El Proceso Unificado de Desarrollo*. s.l. : Addison Wesley Logman, 2000.
17. **Galves, Jorge**. *Fundamentos de la metodología RUP, Rational Unified Process*. [Online]
Septiembre 16, 2007. [Cited: Noviembre 20, 2008.] <http://www.scribd.com/doc/297224/RUP>.
18. **Booch, Grady and RUMBAUGH, J**. *UML: El lenguaje Unificado de modelado*. s.l. : Addison 2000.
19. Requerimientos Funcionales y no Funcionales. *Tecnologico*. [Online]
<http://www.mitecnologico.com/Main/RequerimientosFuncionalesYNoFuncionales>.
20. **Naranjo, Alejandro Arias**. Definición de los Casos de Uso. *CassandraWeb*. Ciudad de la Habana : s.n., 2007.
21. **Larman, C**. *UML y Patrones 2*. s.l. : Addison Wesley Professional, 2002.
22. **Andricain, Augusto Meilan and Denis, Yordanis Marquez**. Análisis y Diseño del Sistema. *Cugnot Server*. Ciudad de La Habana : s.n., 2009.
23. **Jacobson, Ivar, Booch, Gardy and Rumbaugh, James**. Análisis. *El prodeso Unificado de desarrollo de Software*. Madrid : s.n., 2000.
24. **Naranjo, Alejandro Arias**. Diagrama de Clases del Análisis. *CassandraWeb*. Ciudad de la Habana : s.n., 2007.
25. **Andricain, Augusto Meilan and Denis, Yordanis Marquez**. Modelo de Diseño. *Cugnot Server*. Ciudad de la Habana : s.n., 2009.
26. **IBM Maximo Asset Management, IBM Tivoli Asset Management**. *Workflow Implementation Guide*. 2008.
27. **Informáticas, Universidad de Ciencias**. *Flujo de Implementación*. UCI, s.l. : s.n.
28. **Barajas, Saulo**. Saulo.net. *Saulo.net*. [En línea] 8 de Diciembre de 2001.

Cugnot Explorer: Explorador del Repositorio de datos

Bibliografía

29. **Braude, J.** *Ingeniería de software: Una perspectiva orientada a objetos.* . s.l. : Ra-ma,, 2003.
30. **Boogs, Wendy, y Michael Boogs.** *Mastering UML with Rational Rose 2002.* California : s.n., 2002.
31. **Chonoles, Michael Jesse, y James A. Schardt.** *UML 2 for dummies.* New York: : Wiley, 2003.
32. **Cooper, James W.** *Introduction to Design Patterns in C#.* 2002.
33. *Fase de elaboración. Flujo de análisis y diseño. Modelo de análisis Conferencia de Ingeniería de Software I. UCI.* La Habana : s.n., 2008.
34. **Ferguson, Jeff, Brian Patterson, y Jason Beres.** *La biblia de C#.* Madrid: : Anaya Multimedia, 2003.
35. **Hamilton, Kim, y Russell Miles.** *Learning UML 2.0.* 2006.
36. **Larman, Graig.** Capas arquitectónicas. [aut. libro] Pablo Eduardo Roig Vlkquez. *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.
37. **Pressman, Roger S.** *Ingeniería de software: Un enfoque práctico.* s.l. : Addison-Wesley, 2002.
38. **Wesley, Addison.** *El Proceso Unificado de Desarrollo de Software.* 1999.

ANEXOS

Anexo 1

Caso de uso	
CU-1	Realizar Búsquedas
Propósito	Mostrar las características de determinado vehículo a partir de los datos entrados por el usuario.
Actores	Usuario.
Resumen: El usuario entra la matrícula del vehículo, el punto de control, la fecha y hora del inicio y fin del período, con la posibilidad de dejar vacío cualquiera de los campos. El sistema muestra los datos solicitados.	
Referencias	RF1
Acción del actor	Respuesta del sistema
1- Entra los datos.	2- El sistema verifica que existan vehículos o puntos de control acorde a los datos entrados. 3-Si los datos están correctos el sistema muestra la información referente.
Flujo Alternativo	
Acción del actor	Respuesta del sistema
	3- Si los datos no están correctos el sistema lanza una alerta especificando que no existen vehículos o puntos de control con esas características.

Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Puntos de extensión.

Caso de uso	
CU-2	Mostrar Propiedades.
Propósito	Mostrar puntos de control existentes y los servidores de imágenes, así como las entradas asociadas a ellos.
Actores	Usuario.
Resumen: El usuario solicita las propiedades y el sistema se las muestra.	
Referencias	RF2, RF3
Acción del actor	Respuesta del sistema
1- Solicita propiedades.	2- El sistema muestra las propiedades.
Puntos de extensión.	

Caso de uso	
CU-3	Generar Reportes.

Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Propósito	Generar un reporte a partir de la búsqueda que realizó el usuario.
Actores	Usuario.
Resumen: El usuario selecciona la opción de generar reporte y el sistema permite la generación del mismo.	
Referencias	RF4
Precondiciones	Para la realización de esta funcionalidad es necesario que se hayan realizado búsquedas primero.
Acción del actor	Respuesta del sistema
1- Solicita generación de reporte.	2- El sistema genera el reporte en formato rtf.
Puntos de extensión.	

Caso de uso	
CU-4	Visualizar Imagen
Propósito	Visualizar la imagen de la matrícula de un vehículo a partir de las búsquedas realizadas.
Actores	Usuario.
Resumen: El usuario solicita la visualización de la imagen y el sistema la muestra.	
Referencias	RF5

Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Precondiciones	Para visualizar una imagen es necesario haber realizado búsquedas.
Acción del actor	Respuesta del sistema
1- Solicita imagen.	2- El sistema muestra la imagen.
Puntos de extensión.	

Caso de uso	
CU-5	Aplicar Efecto Lupa
Propósito	Aplicar efecto lupa sobre la imagen de un vehículo a después de ser visualizada la imagen.
Actores	Usuario.
Resumen: El usuario aplica efecto lupa y el sistema muestra la sección elegida por el usuario más grande.	
Referencias	RF6
Precondiciones	Tiene que haberse visualizado una imagen primero.
Acción del actor	Respuesta del sistema
1- Aplica efecto lupa.	2- El sistema muestra la sección que el usuario señala con el mouse más grande.

Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Puntos de extensión.

Anexo 2

Diagrama de Clases del Análisis: CU GenerarReportes

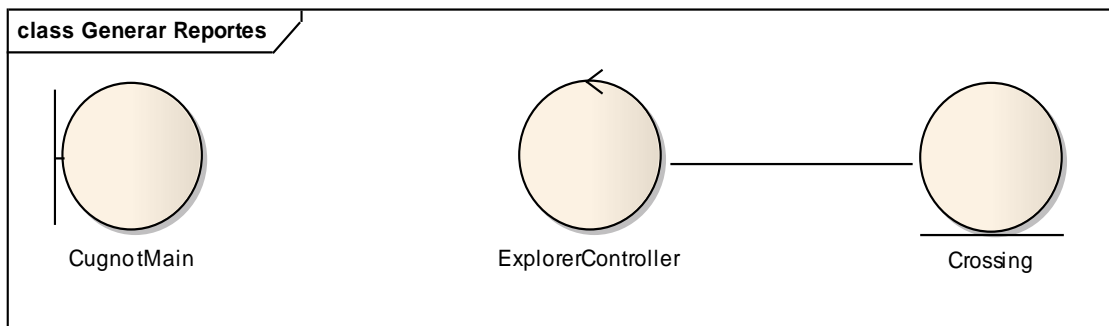
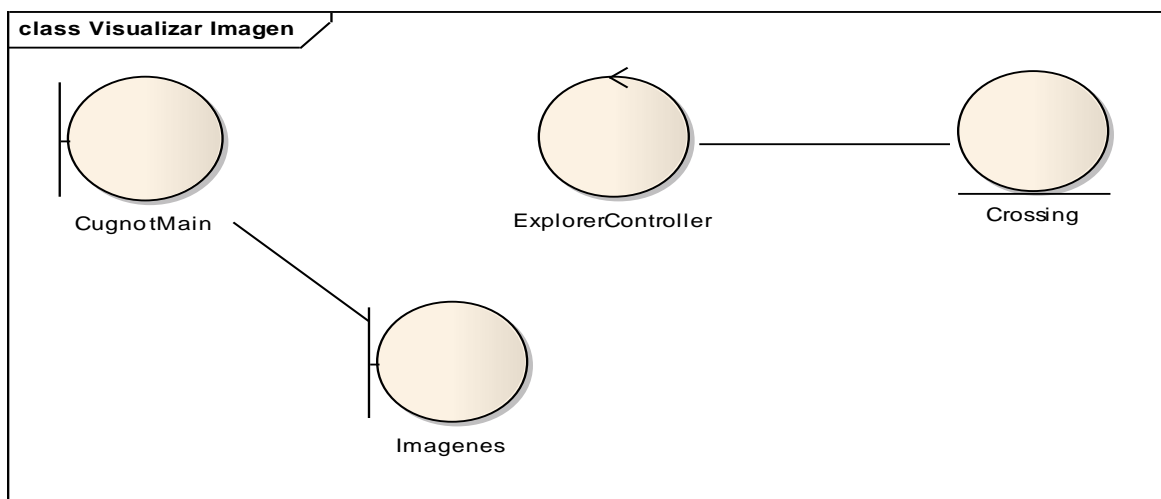


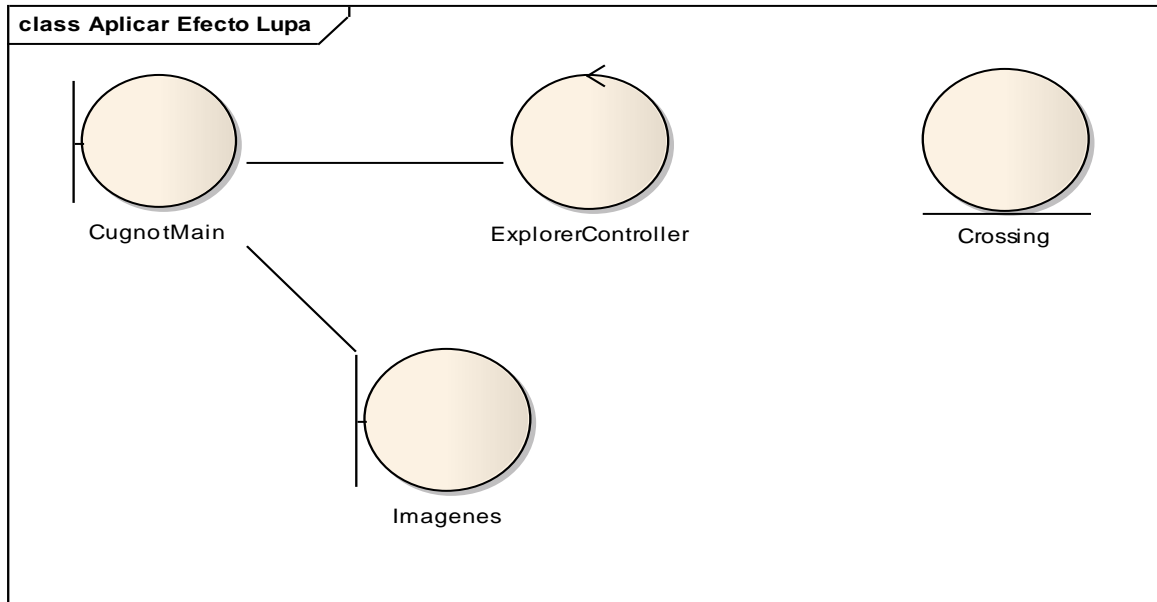
Diagrama de Clases del Análisis: CU VisualizarImagen



Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Diagrama de Clases del Análisis: CU Aplicar Efecto Lupa



Anexo 3

Diagrama de Clases del Diseño: CU GenerarReportes

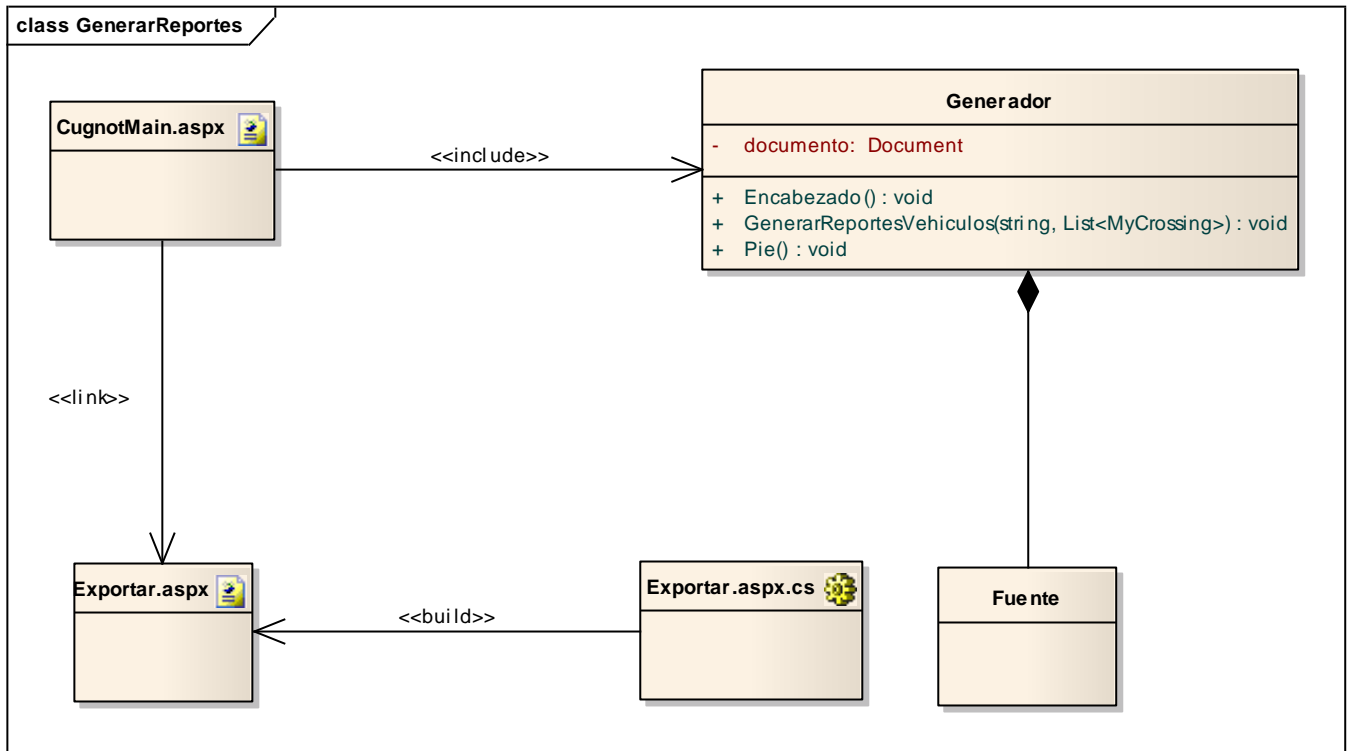


Diagrama de Clases del Diseño: CU VisualizarImagen

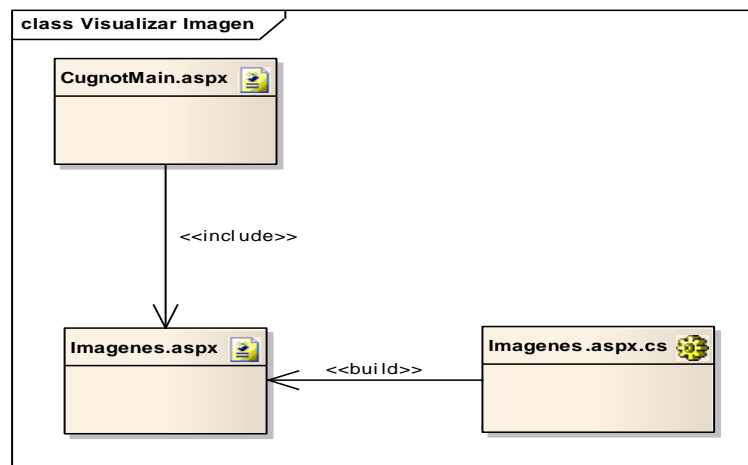
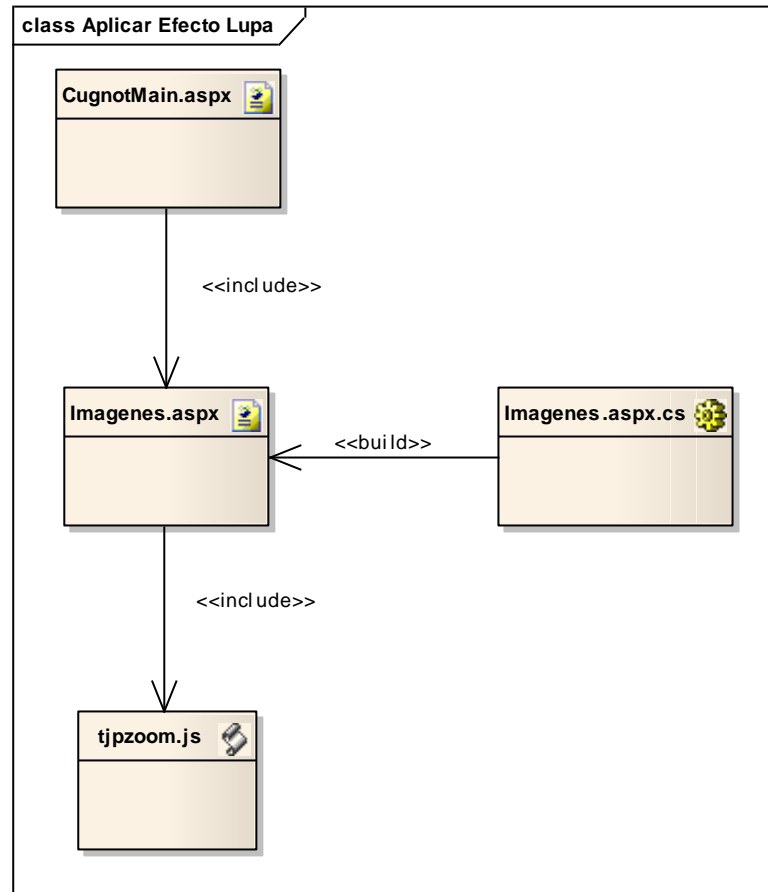


Diagrama de Clases del Diseño: CU Aplicar Efecto Lupa



Anexo 4

Diagrama de Secuencia del Diseño: CU GenerarReportes

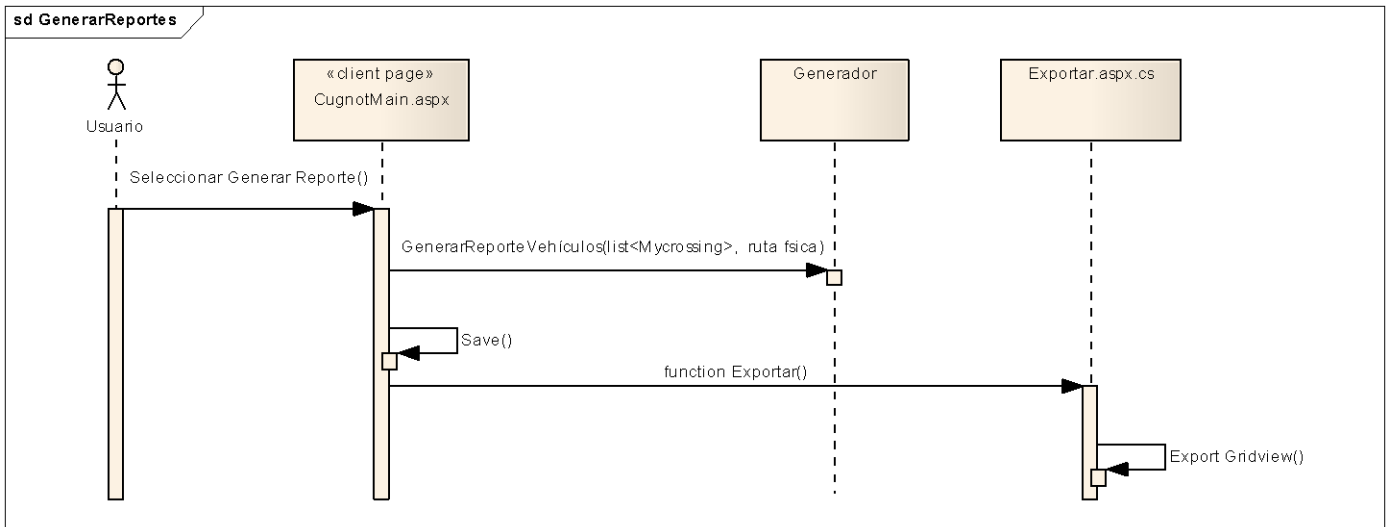


Diagrama de Secuencia del Diseño: CU VisualizarImagen

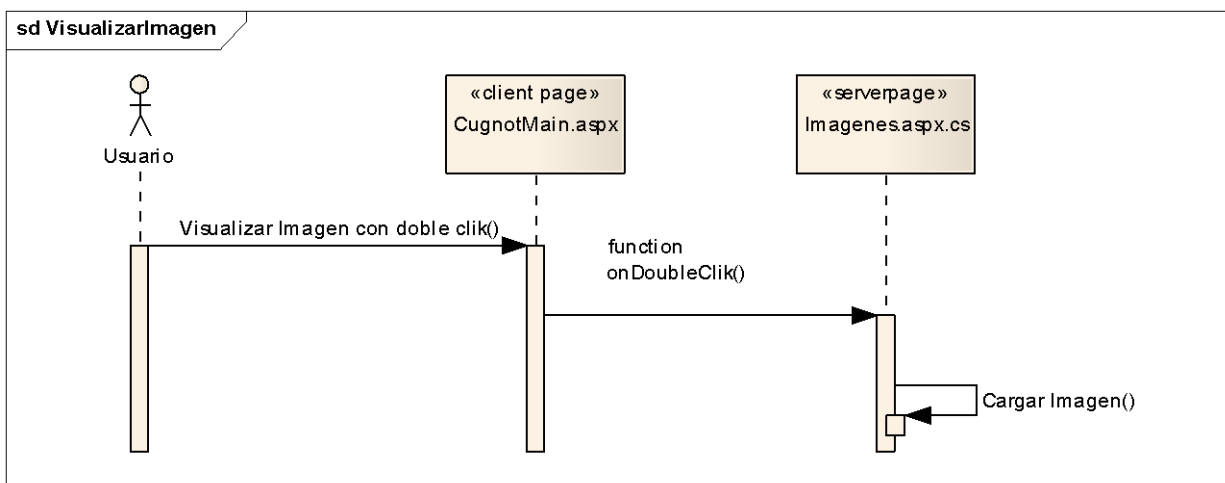
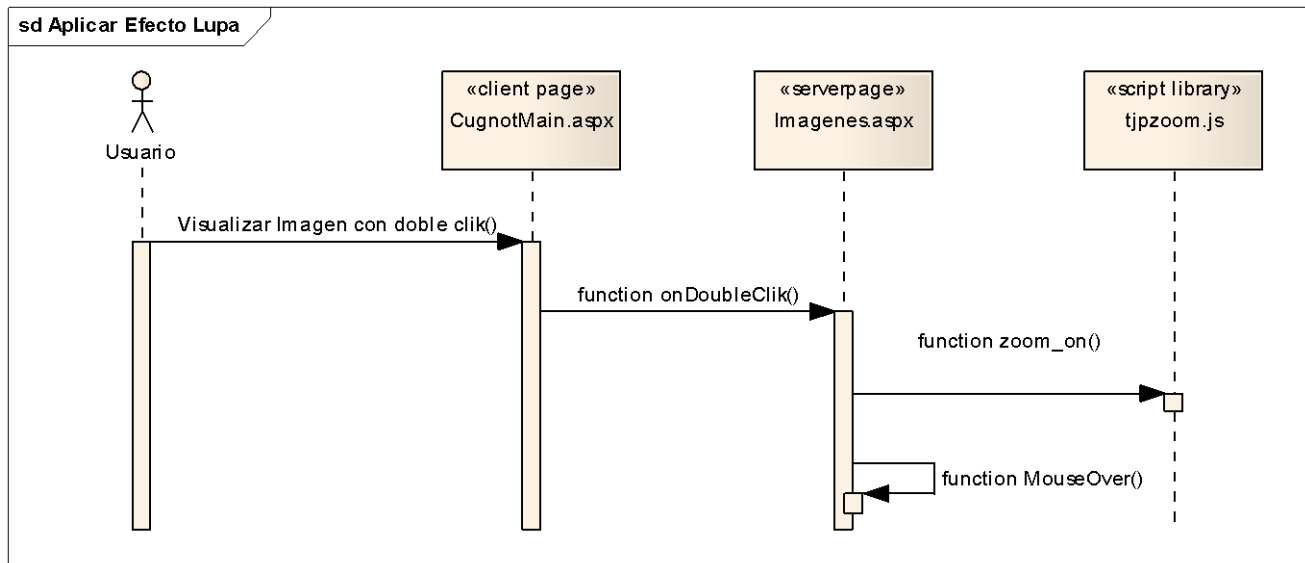


Diagrama de Secuencia del Diseño: CU Aplicar Efecto Lupa



Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Anexo 5

Clases:

Nombre: ExplorerController	
Tipo de clase (controladora)	
Atributo	Tipo
_crossingReader	PgSqlCrossingReader
_placeReader	PgSqlPlaceIdsReader
_serversReader	PgSqlServersReader
Para cada responsabilidad:	
Nombre:	Crossing[] ReadOneCarAtOnePlace(string licencePlate, string placeld, DateTime start, DateTime end, string groupByField)
Descripción:	Busca los cruces que ha realizado un vehículo por un punto de control, en período determinado.
Nombre:	Crossing[] ReadOneCarAtAnyPlace(string licencePlate, string placeld, DateTime start, DateTime end, string groupByField)
Descripción:	Busca los cruces que ha realizado un vehículo por cualquier punto de control, en un período determinado.
Nombre:	Crossing[] ReadAnyCarAtOnePlace(string licencePlate, string placeld, DateTime start, DateTime end, string groupByField)
Descripción:	Busca los cruces que ha realizado cualquier vehículo por un punto de control, en un período determinado.
Nombre:	Crossing[] ReadAnyCarAtAnyPlace(string licencePlate, string placeld, DateTime start, DateTime end, string groupByField)

Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Descripción:	Busca los cruces que ha realizado cualquier vehículo por cualquier punto de control, en un período determinado.
Nombre:	KeyValuePair<string, int>[] Read();
Descripción:	Recupera el identificador de los puntos de control asociados a las entradas del repositorio central.
Nombre:	KeyValuePair<string, int>[] Leer();
Descripción:	Recupera la dirección de los servidores de imágenes asociados a las entradas del repositorio central.

Nombre: Crossing	
Tipo de clase (entidad)	
Atributo	Tipo
_licensePlate	string
_placeld	string
_shotDateTime	DateTime
_imageFileUrl	string
_videoFileUrl	string
Para cada responsabilidad:	
Nombre:	Crossing(string licensePlate, string placeld, DateTime shotDateTime, string imageFileUrl)

Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Descripción:	Inicializa una nueva instancia de este tipo.
Nombre:	string LicensePlate
Descripción:	Retorna o modifica la matrícula del vehículo que cruza el punto de control.
Nombre:	string Placeld
Descripción:	Retorna o modifica el identificador del punto de control.
Nombre:	DateTime ShotDateTime
Descripción:	Retorna o modifica la fecha y hora en que se produce el cruce.
Nombre:	string ImageFileUrl
Descripción:	Retorna o modifica la dirección de la imagen asociada al cruce.
Nombre:	bool Equals(Crossing other)
Descripción:	Indica si la instancia actual es igual a otra instancia del mismo tipo.
Nombre:	override bool Equals(object obj)
Descripción:	Indica si un objeto es igual a esta instancia.
Nombre:	override int GetHashCode()
Descripción:	Genera un código hash a partir del estado de la instancia.
Nombre:	object Clone()
Descripción:	Retorna un nuevo objeto que es una copia exacta de esta instancia.

Nombre: KeyValuePair
Tipo de clase (entidad)
Para cada responsabilidad:

Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Nombre:	KeyValuePair(TKey key, TValue value)
Descripción:	Inicializa una nueva instancia de este tipo.
Nombre:	TKey Key { get ; }
Descripción:	Retorna la llave en el par key/value.
Nombre:	TValue Value { get ; }
Descripción:	Retorna el valor en el par key/value.
Nombre:	override string ToString()
Descripción:	Retorna una representación string de System.Collections.Generic.KeyValuePair<TKey,TValue>, usando la representación string de la llave y el valor.

Páginas:

Nombre: CugnotMain.aspx
Tipo de página: Cliente
Descripción: Esta página es la encargada de presentar toda la información del sitio. Es con la que interactúa directamente el cliente.

Nombre: ExecuteQuery.aspx.cs
Tipo de página: Servidora
Descripción: Esta página es la encargada de enviar y obtener la información relacionada con las búsquedas; así como crear el reporte y guardarlo.

Cugnot Explorer: Explorador del Repositorio de datos

Anexos

Nombre: LoadProperties.aspx.cs

Tipo de página: Servidora

Descripción: Esta página es la encargada de obtener la información relacionada con los puntos de control y las entradas asociadas a los mismos.

Nombre: LoadProperties2.aspx.cs

Tipo de página: Servidora

Descripción: Esta página es la encargada de obtener la información relacionada con los servidores de imágenes y las entradas asociadas a los mismos.

Nombre: Exportar.aspx.cs

Tipo de página: Servidora

Descripción: Esta página es la encargada de abrir el reporte.

Nombre: Imagenes.aspx

Tipo de página: Cliente

Descripción: Esta página es la encargada de mostrar la imagen.

Nombre: Imagenes.aspx.cs

Tipo de página: Servidora

Descripción: Esta página es la encargada de obtener la imagen para mostrarla en la cliente.