

**Universidad de las Ciencias Informáticas.**

**Facultad 5**



**Título:** Herramienta para almacenar de forma  
Genérica información de los Ciudadanos.

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS.**

**Autoras**

Dayany Díaz Corona  
Yaneisy Villalba Beltrán

**Tutora**

Ing. Dania Souchay Fabrega

**Ciudad de la Habana  
Junio 2007**

## DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos las únicas autoras de este trabajo y autorizamos a la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_ del 2007.

\_\_\_\_\_  
Firma de la Autora

\_\_\_\_\_  
Firma de la Autora

\_\_\_\_\_  
Firma de la Tutora

## DATOS DE CONTACTO

### **Ing. Dania Souchay Fábrega**

Graduada de Ingeniera Informática en la CUJAE, en el año 2004. Actualmente profesora Instructora de la Universidad de las Ciencias Informáticas. Experiencia en el desarrollo de multimedia en el proyecto Guardería año 2004-2005 así como en otros proyectos posteriores. Realizó cursos de superación como diplomados de docencia universitaria y Gráfico por Computadora, así como curso para Tutores ofrecidos por el CREA en la CUJAE y Desarrollo de aplicaciones Web usando tecnología Java. Impartió curso de postgrado relacionado con Lenguaje PHP. Cuenta con la publicación del libro Programación Web destinado a los IPIs.

*"Solo es útil el conocimiento que nos hace mejores."*

*Sócrates*

*A Nuestros Padres.*

## **Agradecimientos**

De **Dayany**:

A mis padres, por la educación que me han brindado, por el cariño, amor y apoyo que me han ofrecido toda la vida, por ser el motivo de inspiración de llevar todo este sueño adelante y poder siempre contar con ellos.

A mi hermana, por ser mi ejemplo a seguir, por su bondad y preocupación, creo que sin ella no hubiese podido lograrlo.

A José, por sus sabios consejos.

A Manzano, por haberme impulsado a estudiar esta carrera.

A Frank, por ayudarme tanto en esta última etapa, por sus consejos, por su amistad, por todo.

A mis compañeros por haber tenido la posibilidad de compartir con ellos.

A mis profesores de toda la vida.

A mi compañera de tesis y amiga, por soportarme todo este tiempo.

A mi tutora, por su confianza, apoyo y comprensión.

A todas aquellas personas que contribuyeron a la realización de este trabajo.

A la Revolución.

De **Yaneisy**:

A todas aquellas personas que han hecho posible que hoy este aquí. A mis padres por su cariño y apoyo. A mi compañera de tesis, a mi tutora por su confianza y comprensión, a mis profes todos, a Olguita, Yunet, gracias por su amistad. A mis compañeros, con los que he transitado durante estos cinco años.

A la Revolución.

El uso de aplicaciones genéricas para dar solución a diferentes problemas y situaciones de la vida cotidiana se ha hecho muy común por estos días, construir aplicaciones capaces de ser adaptadas a diferentes realidades es de vital importancia para el desarrollo de software.

Uno de los proyectos productivos de la Universidad de las Ciencias Informáticas (UCI), de la facultad 5, ha desarrollado el software Evaluador Teórico CISMA, encargado de todo el proceso de realización del examen teórico para obtener la licencia de conducción. Solo permite registrar ciudadanos del país para el cual se realizó la aplicación, lo que trae consigo que el software se encuentre sometido a constantes actualizaciones, siendo esta la problemática a la que se pretende dar solución con esta investigación. Teniéndose por objetivo, desarrollar una herramienta para almacenar la información de los ciudadanos de diferentes nacionalidades que serán evaluados por el Evaluador Teórico CISMA.

La aplicación resultante del producto permitirá almacenar información de esta clase de ciudadanos, ofreciendo además un servicio Web que brinde la funcionalidad de búsqueda a otras aplicaciones, lo cual facilitará que el software Evaluador Teórico CISMA tenga un mejor servicio de almacenamiento y que no se le realicen ante el pedido de diversos clientes, constantes actualizaciones.

**INDICE**

INTRODUCCIÓN .....	1
Capítulo1. Fundamentación Teórica.....	4
1.1 Examen teórico de conducción .....	4
1.2 Evaluador Teórico CISMA .....	4
1.3 Aplicaciones Genéricas.....	6
1.3.1 Modularidad.....	6
1.3.2 Extensibilidad.....	7
1.4 Análisis comparativos con otras soluciones existentes.....	7
1.5 Metodología utilizada .....	8
1.5.1 UML.....	9
1.5.2 Rational Rose .....	9
1.6 El servicio Web para el desarrollo de aplicaciones.....	10
1.6.1 Servicios Web desde PHP con NuSOAP.....	11
1.7 La arquitectura cliente / servidor .....	13
1.8 Apache: Servidor Web.....	13
1.9 Sistemas Gestores de Base de Datos.....	14
1.9.1 SQL.....	15
1.9.2 MySQL.....	15
1.9.3 SQL Server.....	16
1.10 Propuesta elegida: MySQL .....	17
1.11 Estudio sobre los Lenguajes de programación.....	17
1.11.1 HTML .....	17
1.11.2 Java Script.....	18
1.11.3 Hojas de Estilo en Cascada .....	18
1.11.4 AJAX .....	19
1.11.5 PHP .....	20
1.11.6 ASP .....	21
1.12 Lenguaje seleccionado PHP.....	22
1.13 PHP, MySQL y Apache .....	22
1.14 Herramientas utilizadas.....	23
Capítulo 2. Características del sistema .....	25
2.1 Estado actual del Negocio.....	25
2.2 Modelo del dominio .....	26
2.3 Solución Propuesta.....	28
2.4 Requisitos funcionales .....	29
2.5 Requerimientos No Funcionales .....	31
2.6 Modelo de casos de uso del sistema.....	32
2.6.1 Diagrama de Casos de Uso del sistema .....	33
2.6.2 Descripción de los casos de uso del sistema.....	34



2.6.3 Expansión de los Casos de Uso .....	35
Capítulo 3. Desarrollo de la solución propuesta .....	59
3.1 Patrones de diseño.....	59
3.1.1 Patrón Modelo Vista Controlador .....	60
3.2 Diagrama de Clases Web .....	61
3.3 Diseño de la Base de Datos.....	70
3.3.1 Diagrama de Clases Persistentes.....	71
3.4.2 Modelo de Datos. ....	72
3.4 Principios de diseño .....	73
3.4.1 Estándares en la interfaz de la aplicación .....	73
3.4.2 Formato de los reportes.....	74
3.4.3 Tratamiento de excepciones.....	74
3.4.4 Estándares de codificación.....	74
3.5 Modelo de despliegue .....	75
3.6 Descripción preliminar del modelo de pruebas.....	76
3.6.1 Casos de prueba .....	76
CONCLUSIONES GENERALES.....	81
RECOMENDACIONES.....	82
REFERENCIAS BIBLIOGRAFICAS.....	83
BIBLIOGRAFÍA .....	86
ANEXOS .....	87

### INTRODUCCIÓN

Hacer uso de las nuevas tecnologías y la computación en la actualidad es tarea de primer orden. Lograr que nuestro universo quede íntegramente insertado dentro de los ordenadores y así manipularlo empleando la menor fuerza humana posible, sigue siendo una utopía, pues todavía queda mucho por explorar y aprender; sin embargo lo creado hasta ahora demuestra que todo puede suceder cuando de automatización se trata.

La Universidad de las Ciencias Informáticas (UCI), institución educacional que tiene a su cargo la formación de futuros ingenieros informáticos, vinculando la docencia a la producción activa de software, cuya misión es la formación de un personal altamente capacitado y calificado para fomentar la Industria del Software en Cuba, ha entrado en colaboración con la empresa de Simuladores Profesionales (SIMPRO), unidas han desarrollado un software encargado de computarizar el examen teórico que se realiza con el objetivo de obtener la licencia de conducción, recibiendo el nombre de Evaluador Teórico CISMA. El registro de los ciudadanos que realizan esta prueba, se hace a través de una aplicación donde se almacenan sus datos personales y los resultados obtenidos después de la realización de la evaluación teórica.

Este programa funciona a través de una aplicación que muestra una interfaz externa con videos de diferentes situaciones del tránsito con sus correspondientes preguntas, con la que interactúa el cliente una vez efectuado el período evaluativo. Y está integrado por cuatro áreas de trabajo: Área de Administración, Área de Supervisión, Área del Cliente y Área de Registro.

El Evaluador Teórico supramencionado no permite registrar información de residentes de diferentes países, razón por la cual se sometía a frecuentes actualizaciones ante el pedido de esta clase de clientes.

De ahí el surgimiento de este trabajo como necesidad de dar respuesta a las situaciones antes expuestas, por lo que el **problema** a resolver queda formulado de la siguiente forma: **¿Cómo facilitar el proceso**

**de administración de la información de los ciudadanos registrados de diferentes nacionalidades en el software Evaluador Teórico CISMA?**

Para ello el **objeto de estudio** lo constituye la Gestión de la información en el proceso de Evaluación del Examen Teórico de Conducción en el software Evaluador Teórico CISMA. Derivándose que el **campo de acción** que abarca esta investigación es la manipulación de datos asociados a ciudadanos de diferentes nacionalidades en el área del registro para Evaluadores Teórico de Conducción.

Este manejo de datos esta dado por el registro de los residentes, actualización de sus datos o eliminación de los mismos, el proceso de búsqueda de información sobre los que han sido registrados y la obtención de reportes sobre la información almacenada.

Esta indagación tiene como propósito defender la idea de que si se implementa una herramienta que almacene la información de este tipo de ciudadanos, obtenga información de los registrados, permitirá que el software Evaluador Teórico CISMA tenga un mejor servicio de almacenamiento y que ante el pedido de clientes de diversos territorios no sea sometido a reiteradas actualizaciones.

Precisamente se decidió que el **objetivo** del trabajo consistiera en: Desarrollar una herramienta para almacenar de forma genérica la información de los ciudadanos de diferentes nacionalidades que serán evaluados por el Evaluador Teórico CISMA.

Conforme a este planteamiento se derivan las siguientes **tareas**:

- Realizar el estudio del estado del arte en aplicaciones genéricas para almacenar información de ciudadanos.
- Analizar y estudiar cómo se encuentran en el ámbito internacional las tecnologías que se utilizan para llevar a cabo herramientas como la que se pretende desarrollar.
- Seleccionar la Metodología de Análisis y Diseño de Sistemas Informáticos, que facilite la creación y garantice la calidad del sistema.
- Diseñar una base de datos que soporte las funcionalidades del sistema.

- Implementar una aplicación Web que facilite almacenar la información de forma genérica correspondiente a los ciudadanos de distintas nacionalidades que realizaran el examen teórico, que muestre reportes acerca de la información almacenada.
- Modelar un servicio Web que brinde la funcionalidad de búsqueda a otras aplicaciones.
- Iniciar el proceso de desarrollo hasta que el producto se encuentre con una capacidad operativa inicial.

El desarrollo de esta aplicación permitirá su posterior integración con el software Evaluador Teórico CISMA, minimizando las constantes actualizaciones a las que estaba sometido y también podrá ofrecer un servicio Web de búsqueda a cualquier software que lo requiera. Además se extenderá y comercializará la aplicación a otras instituciones nacionales e internacionales.

Estará estructurada la investigación en tres capítulos:

El **Capítulo 1** describe la situación actual, se hace referencia a sistemas automatizados existentes en la realización del examen teórico para obtener la licencia de conducción. Se describen las metodologías, tendencias y tecnologías actuales a considerar en la selección de aquellas que se van a utilizar en la implementación de la aplicación.

El **Capítulo 2** describe el negocio a través de un modelo de Dominio, el cual sirve de base para determinar qué es lo que se va a desarrollar. Descripción de la aplicación que se propone así como de todos los requisitos funcionales y no funcionales que esta debe cumplir.

El **Capítulo 3** describe el desarrollo de la solución propuesta se modelan los diagramas de clases de diseño, los diagramas de interacción, el diagrama de clases persistentes, se plantea el modelo de datos, se especifican los patrones de diseño utilizados, así como los principios para el diseño gráfico y la implementación y se realizan pruebas para los casos de uso más significativos.



## **Capítulo 1. Fundamentación Teórica**

### **Introducción**

En el presente capítulo se brinda una visión general de los aspectos relacionados con las herramientas genéricas para almacenar información, los principales conceptos relacionados con el problema, así como la descripción del objeto de estudio y campo de acción. También se muestran las principales tecnologías, metodologías, herramientas y lenguajes de programación utilizados para el desarrollo de la aplicación. Además de realizarse un análisis comparativo con otras herramientas existentes.

### **1.1 Examen teórico de conducción**

El examen teórico es una evaluación de conocimientos, que se realiza a todos los aspirantes a obtener el permiso, licencia o carné de conducción, teniendo como contenido generalmente dar respuestas a cuestionarios que recogen disímiles preguntas relacionadas con variadas situaciones del tránsito y señalizaciones viales. Esta evaluación presenta la responsabilidad de confirmar si el conductor consta con los conocimientos necesarios para poder conducir, requiriendo una preparación general acerca de los conceptos básicos del manejo de vehículos y el Código del Tránsito vigente en el continente, país, localidad donde se este realizando.

### **1.2 Evaluador Teórico CISMA**

En los últimos años la industria automovilística ha crecido de manera incesante, y junto con ella los accidentes del tránsito, debido a las múltiples irresponsabilidades de los conductores, pues muchos de

ellos no están aptos para conducir y no presentan licencia o carné de conducción, no constituyendo la penalización la mejor solución a la problemática, se hace necesario entonces la realización de los exámenes de aptitud para obtener o renovar una licencia de conducción. El examen se divide en dos tipos de evaluaciones, la parte teórica y la parte práctica.

En muchos países no se cuenta con el recurso humano necesario para la supervisión de este tipo de evaluaciones, o el presupuesto empleado para esta actividad es escaso, lo que trae consigo que se cometan constantemente actos de fraude, otorgando permisos de conducción de forma incorrecta. Ante todo este esquema los resultados son accidentes, mala educación vial y por supuesto muertes.

Como solución a esta situación se ha desarrollado un software encargado de automatizar el examen teórico que realizan los aspirantes a obtener el permiso, carné o licencia de conducción, este programa recibe el nombre de Evaluador Teórico CISMA y es adaptable a la cultura vial de cada país. Consiste en evaluar al ciudadano y si su capacidad no alcanza en materia de conducción y tránsito pues no calificará para obtener la licencia, para esto el Evaluador Teórico CISMA mide técnicamente al solicitante, primeramente se le hace una prueba de la vista donde el interesado debe de tener un promedio visual aceptable, posteriormente el aspirante se someterá a la prueba teórica la cual consta de una serie de preguntas con videos de situaciones del tránsito para medir los conocimientos viales y al final se da la evaluación del examen.

Este software para su funcionamiento está integrado por cuatro áreas de trabajo: Área de Administración, es donde se realiza cualquier actualización requerida en el examen (como cambiar el número de preguntas, el tiempo de realización del examen o la cantidad de videos de situaciones del tránsito), se muestran diferentes reportes de las personas evaluadas y se administran los usuarios con acceso al sistema adicionando o eliminando personas a este. Área de Supervisión: desde esta área se atienden las acciones que realiza cada cliente con la aplicación en dependencia del estado en que se encuentre cada máquina (en espera, disponible, ocupada o terminado) hasta tanto el cliente termina su examen o incluso si se retira antes de culminarlo, caso en el que se debe alistar la máquina para un nuevo cliente. Área del Cliente: es aquí donde se muestra al usuario el examen teórico que muestra videos de diferentes situaciones del tránsito con sus correspondientes preguntas, con las que interactúa el cliente cuando efectúa la prueba para optar por el carné o licencia de conducción. Y por último se encuentra el Área de

Registro: en esta área se solicitan y almacenan los datos personales del ciudadano que realizará el examen y los resultados obtenidos después de su culminación.

### **1.3 Aplicaciones Genéricas**

En la actualidad existe un rápido crecimiento de las tecnologías y la tendencia a realizar aplicaciones genéricas cada vez es mayor debido a las múltiples facilidades que brinda, construir aplicaciones capaces de ser adaptadas a diferentes realidades es de vital importancia para el desarrollo de software. De esta forma, es posible adaptarse rápidamente a los cambios que ocurren en las organizaciones y reutilizar una aplicación en un amplio aspecto de realidades.

La genericidad es un recurso importante de factorización y reusabilidad en la programación, las funciones y los tipos genéricos soportados por la programación funcional son una buena base para introducir conceptos de reutilización de código.

#### **1.3.1 Modularidad**

Un problema complejo no puede solucionarse de una sola vez, por eso una de las técnicas más utilizadas en la programación es la de dividir este problema en subproblemas más sencillos, que den la solución al problema original. Para resolver estos subproblemas se necesita crear y trabajar con subprogramas llamados módulos, de ahí el nombre de programación modular.

Esta última consta de varias secciones divididas de forma que interactúan a través de llamadas a procedimientos, que integran el programa en su totalidad. El programa principal coordina las llamadas a los módulos secundarios y pasa los datos necesarios en forma de parámetros.(ALVAREZ, SARA, 2006b)

La importancia de este tipo de programación, radica precisamente que sus módulos son independientes, es decir pueden ser modificados o reemplazados sin afectar el resto del programa o pueden ser reutilizado en otros programas.

La ventaja de utilizar el diseño modular radica en que sus módulos pueden escribirse y probarse por separados y son más fáciles de mantener y documentar.

### 1.3.2 Extensibilidad

La extensibilidad de un programa o aplicación informática es la capacidad que presentan de adaptarse a los cambios de especificación.

Para programas pequeños realizar cambios de los requisitos, de los algoritmos, de la representación de los datos, de las técnicas de implementación, etc., no es una tarea engorrosa; pero a medida que el software crece comienza a ser cada vez más difícil de adaptar, aunque la extensibilidad se pueden aplicar con pequeños programas, su importancia sólo se refleja con claridad en los grandes proyectos.

De ahí que sus principios esenciales son:

- Simplicidad del diseño: una arquitectura simple siempre será más fácil de adaptar a los cambios que una compleja.
- Descentralización: cuanto más autónomos sean los módulos, más alta es la probabilidad de que un cambio afecte a un solo módulo, o a un número pequeño de módulos, en lugar de provocar una reacción en cadena de cambios en el sistema completo. (PAREDES, 2005)

### 1.4 Análisis comparativos con otras soluciones existentes

El registro de información de los ciudadanos es una actividad cotidiana que se realiza en la tramitación del proceso para obtener el permiso, licencia o carné de conducción. Hacer dicha actividad lo más general posible y así lograr que el Evaluador Teórico CISMA obtenga mejores funcionalidades, es uno de los propósitos de este trabajo.

En la actualidad existen algunos países que realizan el examen teórico de conducción a través de computadoras. Suecia es uno de los países donde el examen teórico para obtener la licencia de conducción se encuentra automatizado, la aplicación que se utiliza para realizar el examen teórico muestra una serie de preguntas y respuestas acompañadas de imágenes, donde el usuario seleccionará la respuesta correcta. El módulo de registro de esta aplicación solamente permite registrar a personas residentes en Suecia.



En Perú también se realiza el examen teórico para obtener la licencia de conducción a través de ordenadores, una vez registrada la información necesaria del ciudadano a realizar el examen, se le muestra un cuestionario en el que deberá seleccionar la respuesta correcta. Esta aplicación solo permite registrar ciudadanos peruanos para obtener la licencia de conducción.

Actualmente en República Dominicana y en Guatemala se aplica la primera versión del sistema Evaluador Teórico CISMA, el cual es nuestro objeto de estudio. Este sistema muestra como algo novedoso la presencia de videos con diferentes situaciones del tránsito e interrogantes que el aspirante a obtener la licencia de conducción deberá responder. El sistema consta de un Área de Registro que al igual que las otras aplicaciones está desarrollado solamente para registrar ciudadanos del país donde se este realizando el examen.

Por ello para lograr un mejor funcionamiento del sistema Evaluador Teórico CISMA, se decidió hacer del Área de Registro una aplicación lo mas general posible, minimizando y facilitando el trabajo en el momento de implementar el Evaluador Teórico CISMA para un país determinado, pues esta área sería un módulo que solamente habría que integrar al sistema.

La herramienta que se propone desarrollar formará parte de la aplicación Evaluador Teórico CISMA, pasando a ser, a su vez, un módulo de esta.

### **1.5 Metodología utilizada**

La tendencia actual en el software lleva a la construcción de sistemas más grandes, rápidos y complejos, ya que se busca adecuarse a las necesidades del usuario.

Entonces se necesita un método común para lograr la productividad del software, que integre las múltiples facetas del desarrollo. Por lo que es necesario definir la metodología de ingeniería de software que guiará el proceso de automatización, en este caso se optó por el Proceso Unificado de Rational (RUP).

Este Proceso Unificado es un marco de desarrollo de software (conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software), dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental y está basado en componentes. Utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software.

RUP no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado para diferentes áreas de aplicación, diferentes organizaciones o proyectos específicos. Esta metodología entre sus múltiples ventajas ayuda a asegurar la producción de software de alta calidad con un costo y tiempo predecible para el usuario.

### 1.5.1 UML

UML (Unified Modeling Language) o Lenguaje Unificado de Modelamiento es un lenguaje gráfico de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos (pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque Orientado a Objetos, utilizándose también en el diseño Web.) de un sistema de software orientado a objetos (OO). (LAFUENTE, 2001)

Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. No es un lenguaje de programación pero las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. Es lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.(ADDISON WESLEY ED. JAMES RUMBAUGH, 2000)

UML es el lenguaje de modelado orientado a objetos estándar predominante ahora y en los próximos años, ya que cuenta con la participación de metodólogos influyentes, de importantes empresas y un estándar del *Object Management Group* (OMG), siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle y Rational.

### 1.5.2 Rational Rose

Para realizar el modelado del desarrollo de los proyectos hay herramientas CASE como el Analise, el Designe y el Rational Rose, siendo esta última la que se utiliza en la modelación de este proyecto. Rational Rose es una herramienta software para el Modelado Visual mediante UML de sistemas software

permitiendo Especificar, Analizar y Diseñar el sistema antes de Codificarlo. Fue desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson) y cubre todo el ciclo de vida de un proyecto.

De manera global las principales características son:

- Mantiene la consistencia de los modelos del sistema software
- Chequeo de la sintaxis UML
- Generación Documentación automáticamente
- Generación de Código a partir de los Modelos
- Ingeniería Inversa (crear modelo a partir código) (BLANCO *et al.*, 2002)

Rational Rose es una herramienta que se puede encuadrar dentro del grupo de herramientas más técnicas debido a que se encarga de llevar a cabo la automatización de los sistemas para la posterior generación de código (esto es, realización de los distintos diagramas y generación del código posterior), tiene una plataforma independiente que ayuda a la comunicación entre los miembros de equipo a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Entre las diferentes ventajas que ofrece el Rose, la más importante es que utiliza la notación estándar en la arquitectura de software (UML), permitiendo esta a los desarrolladores y arquitectos de software utilizar un lenguaje común para poder visualizar el sistema completo. También pueden los diseñadores modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

### **1.6 El servicio Web para el desarrollo de aplicaciones**

El concepto de desarrollo Web ha trascendido el ámbito propio de la red para convertirse en un nuevo paradigma en el desarrollo de software. La mayoría de los sistemas de información y aplicaciones actuales se construyen haciendo amplio uso de tecnologías Web. Estas nuevas aplicaciones son las que albergan la mayor parte de la información que encontramos en Internet y el Servicio Web que ofrece ha dejado de ser una herramienta de publicación de información estática y se ha convertido en una aplicación de tipo Cliente-Servidor.

Los servicios Web son un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web, intercambiando datos entre sí con el objetivo de ofrecer algunos servicios. Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

Ventajas de los servicios Web:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar. (WIKIMEDIA FOUNDATION, 2006d)

Una razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más acusada.

### **1.6.1 Servicios Web desde PHP con NuSOAP**

Los elementos fundamentales que componen a los Servicios Web son:

- **XML (eXtensible Markup Language)** Es un subconjunto simplificado del SGML el cual fue diseñado principalmente para documentos Web. Deja a los diseñadores crear sus propias "etiquetas" o "tags" (Ej: <libro>), habilitando la definición, transmisión, validación, y la interpretación de datos entre aplicaciones y entre organizaciones.
- **SOAP (Simple Object Access Protocol)** Es un protocolo de comunicación, el cual permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje. Esta basado en XML y es la base principal de los Servicios Web. Los mensajes SOAP son documento XML.
- **WSDL (Web Services Description Language)** Es un protocolo basado en XML que describe los accesos al Servicio Web. Podríamos decir que es el manual de operación del Servicio Web, porque nos indica cuales son las interfaces que provee el Servicio Web y los tipos de datos necesarios para la utilización del mismo.
- **UDDI (Universal Discovery Description and Integration)** Es una especificación para mantener directorios estandarizados de información acerca de los Servicios Web, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente. UDDI utiliza WSDL para describir las interfaces de los Servicios Web.

En esta aplicación se va a modelar como desarrollar Servicios Web desde PHP con NuSOAP, un kit de herramientas (ToolKit), esta compuesto por una serie de clases que nos harán mucho más fácil el desarrollo de Servicios Web. Provee soporte para el desarrollo de clientes (aquellos que consumen los Servicios Web) y de servidores (aquellos que los proveen). NuSOAP esta basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1 NuSOAP no es el único soporte para los Servicios Web, existen otros, pero es uno de los que están en una fase de desarrollo mucho más avanzada. PHP a partir de su versión 5 comienza a dar soporte para SOAP, pero aun esta en fase experimental. (*Servicios Web con PHP*, 2007)

Finalmente se llego a la conclusión de utilizarlo ya que esta en una fase madura de desarrollo, no necesita módulos adicionales siendo muy fácil su instalación y uso.

### 1.7 La arquitectura cliente / servidor

En el ámbito de enviar y recibir información la arquitectura Cliente-Servidor es una de las más importantes y utilizadas. Esta arquitectura es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. Es una herramienta potente para guardar los datos en una base de datos como servidor.

Los elementos principales que tiene este modelo son el elemento llamado cliente(es un consumidor de servicios) y el otro elemento llamado servidor(es un proveedor de servicios). Ambos elementos pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga.

El Cliente-Servidor es una relación entre procesos corriendo en máquinas separadas e interactúan por un mecanismo de pasaje de mensajes con un pedido de servicios y una respuesta. Esta arquitectura es el resultado de la integración de dos culturas. Por un lado, la del Mainframe que aporta capacidad de almacenamiento, integridad y acceso a la información y, por el otro, la del computador que aporta facilidad de uso, bajo costo, presentación atractiva y una amplia oferta en productos y aplicaciones. (VALLE, 2005)

### 1.8 Apache: Servidor Web

Apache es el servidor más popular en Internet desde abril de 1996. En Noviembre del 2005, una encuesta sobre servidores Web de Netcraft arrojó que más del 70% de los sitios Web en Internet usaban apache lo que lo hace ser preferido por sobre otros servidores. Entre sus características sobresalen:

- **Fiabilidad:** Alrededor del 90% de los servidores con más alta disponibilidad funcionan con Apache.
- **Gratuidad:** Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.

- **Extensibilidad:** Se pueden añadir módulos para ampliar las amplias capacidades de Apache. Existe una amplia variedad de módulos, que permiten desde generar contenido dinámico con PHP, Java, Perls, entre otros, monitorizar el rendimiento del servidor. Estos módulos pueden ser creados por cualquier persona con conocimientos de programación. (INTERNET, 2006)

El proyecto de Servidor HTTP Apache es un esfuerzo para desarrollar y mantener un servidor HTTP abierto para modernos sistemas operativos como UNIX y Windows NT. El objetivo de este proyecto es ofertar un servidor seguro, eficiente y extensivo que de servicios HTTP en sincronismo con los estándares actuales. (FOUNDATION, 2005)

### 1.9 Sistemas Gestores de Base de Datos

El servidor de Base de Datos es el encargado de garantizar el almacenamiento, integridad, protección y manipulación de la información de sistema.

Una Base de datos es un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una BD puede considerarse una colección de datos variables en el tiempo. (HANSEN y V.HANSEN, 2007)

Un Sistema de Gestión de Base de Datos (SGDB) es un software específico, que permite crear y mantener una (o varias) Base (s) de Datos, asegurando mantener su integridad, confidencialidad y seguridad. Un SGDB esta compuesto por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta.

Los SGDB deben cumplir algunos objetivos específicos como:

- Abstracción de la información.
- Independencia.
- Redundancia mínima.
- Consistencia.

- Seguridad.
- Integridad.
- Respaldo y recuperación.
- Control de la concurrencia.
- Tiempo de respuesta.

Ejemplo de SGDB es el MySQL. (*Sistemas\_Gestores\_de\_Bases\_de\_Datos*, 2006)

### 1.9.1 SQL

El Lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar, actualizar y modificar información de interés de una base de datos, de una forma sencilla.(WIKIMEDIA FOUNDATION, 2006e)

En la actualidad diversos SGDB utilizan el SQL para realizar el tratamiento de los datos almacenado como son MySQL y SQL Server.

### 1.9.2 MySQL

MySQL es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Es la base de datos open source más popular.

MySQL utiliza una arquitectura cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.



La condición de open source de MySQL, hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet. (EDUARDO, 2002)

El servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar. Aunque se encuentra en desarrollo constante, este servidor ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad, y seguridad hacen de él un servidor bastante apropiado para acceder a bases de datos en Internet. MySQL permite una gran interacción con lenguajes de programación como el PHP.

### **1.9.3 SQL Server**

SQL Server es un programa informático de gestión y administración de bases de datos relacionales basada en el lenguaje SQL, que incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

SQL Server presenta la capacidad de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. El servidor de base de datos SQL Server ofrece varias ventajas, entre las que figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Soporta procedimientos almacenados.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accedan a la información.
- Además permite administrar información de otros servidores de datos.(WIKIMEDIA FOUNDATION, 2006c)

Microsoft SQL Server, propietario de Microsoft, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

### **1.10 Propuesta elegida: MySQL**

Tanto el SQL Server como el MySQL son SGDB que operan en una arquitectura cliente/servidor. Después de realizar un análisis minucioso de las características, funcionalidades y facilidades que brindan cada SGDB presentado, se decidió usar el MySQL como SGDB por las siguientes razones:

- El MySQL es multiplataforma.
- El MySQL es gratuito, por lo que se adquiere libremente en el mercado.
- El PHP maneja más fácil al MySQL que al SQL Server, debido a la gran cantidad de funciones que tiene explícitas.

### **1.11 Estudio sobre los Lenguajes de programación**

Al irrumpir en el mundo la World Wide Web y con el amplio crecimiento de la Internet y la Intranet, las aplicaciones y los sitios Web han adquirido gran popularidad a nivel mundial. Esta tecnología requiere de pocos recursos por parte del cliente, basta solamente con la presencia de un navegador de Internet.

Seguidamente se realiza una breve descripción y comparación de algunos de los lenguajes de programación utilizados para la construcción de sitios Web.

#### **1.11.1 HTML**

HTML, acrónimo inglés de HyperText Markup Language, es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. (WIKIMEDIA FOUNDATION, 2006a)

Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

HTML permite crear documentos Web insertando en el texto de los mismos una serie de marcas (tags) que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos. Las marcas o tags que controlan el comportamiento del documento son fragmentos de texto encerrados entre los signos "mayor que" y "menor que" (<marca>). (ARENAS, 2002)

Se puede decir que existen tres utilizaciones básicas de los lenguajes de marcas: los que sirven principalmente para describir su contenido, los que sirven más que nada para definir su formato y los que realizan las dos funciones indistintamente. Las aplicaciones de bases de datos son buenas referencias del primer sistema, los programas de tratamiento de textos son ejemplos típicos del segundo tipo, y aunque no lo parezca, el HTML es la muestra más conocida del tercer modelo.

### **1.11.2 Java Script**

Java Script es un lenguaje de scripts desarrollado por Nestacape, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

Los programas de este tipo van incrustados en los documentos HTML, entre las tags <script> y </script>, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos, etc.

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, orientado a las páginas Web, con una sintaxis semejante a la del lenguaje Java.

No es un lenguaje orientado a objetos propiamente dicho, pues no dispone de Herencia, es más bien orientado a eventos. Actúa en cuanto un evento (un clic en un botón, por ejemplo) sea ejecutado. Aún así javascript implementa una sencilla interfaz de objetos/propiedades/métodos.(WIKIMEDIA FOUNDATION, 2006b)

### **1.11.3 Hojas de Estilo en Cascada**

Hojas de Estilo en Cascada (CSS, acrónimo inglés de Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, ofreciéndole a los desarrolladores Web el control total sobre estilos y formatos de múltiples páginas Web al mismo tiempo. (INTERNET, 2006)

CSS se utiliza para dar estilo a documento HTML y XML, separando el contenido de la presentación. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a:

- Un Web entero, de modo que se puede definir la forma de todo el Web de una sola vez.
- Un documento HTML o página, se puede definir la forma, en un pequeño trozo de código en la cabecera, a toda la página.
- Una porción del documento, aplicando estilos visibles en un trozo de la página.
- Una etiqueta en concreto, llegando incluso a poder unir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación. Podemos definir, por ejemplo, varios tipos de párrafos: en rojo, en azul, con márgenes, sin ellos y otros.

(ALVAREZ, MIGUEL ANGEL, 2006a)

CSS presenta gran importancia ya que:

- Es extremadamente útil como una herramienta para el manejo de documentos.
- Ayuda de gran manera a hacer que las páginas sean más accesibles.
- Los beneficios en rendimiento son fabulosos para el usuario final, ya que un CSS económico resulta que las páginas se carguen más rápido y realicen un renderizado más efectivo.

### **1.11.4 AJAX**

AJAX (acrónimo de " Asynchronous JavaScript And XML"), es una técnica de desarrollo web para crear aplicaciones interactivas mediante la combinación de tres tecnologías ya existentes.

- HTML (o XHTML) y Hojas de Estilo en Cascada (CSS) para presentar la información.

- Document Object Model (DOM) y JavaScript, para interactuar dinámicamente con los datos.
- XML y XSLT, para intercambiar y manipular datos de manera desincronizada con un servidor web.

AJAX funciona en cualquier navegador, es perfectamente compatible con cualquier tipo de servidor estándar y lenguaje de programación Web. PHP, ASP, ASP.Net, Perl, JSP, Cold Fusion.(*Qué es AJAX*, 2007)

Esta nueva técnica, ayuda a la estética de la web, pues se puede combinar la imaginación del desarrollador con la usabilidad de una aplicación web, logrando que las interfaces desarrolladas con Ajax se asemejen en muchos casos a las aplicaciones de escritorio.

### 1.11.5 PHP

PHP (acrónimo de "PHP: Hypertext Preprocessor"), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Es un lenguaje simple y de fácil aprendizaje, que permite una serie de funcionalidades, convirtiéndolo en el favorito de millones de programadores en todo el mundo. Su distribución se realiza de forma gratuita.

Fue creado por Rasmus Lerdorf a finales de 1994 como un complemento para el lenguaje PERL, pero no fue hasta 1995 que se creó la versión 1.0 utilizable para otros usuarios. Esta versión fue conocida como, *Personal Home Page Tools* (Herramientas para Páginas Personales).(VÁZQUEZ, 2003)

En sus inicios, solamente consistía en un analizador sintáctico muy simple que solo entendía unas cuantas macros y una serie de utilidades comunes en las páginas Web. Debido a la gran aceptación que tuvo esta primera versión, el analizador sintáctico fue reescrito a mediados de 1995 y fue nombrado PHP/FI versión 2. En 1997 aparece la Versión 3.0, realizada por Zeev Suraski y Andy Gutsman, donde se combinó con la base de datos MySQL y el servidor Apache. Esta versión fue la más parecida a lo que hoy conocemos como PHP. Desde ese entonces el crecimiento del PHP ha sido exponencial y han surgido versiones nuevas como PHP4 y PHP5.

PHP es Open Source y puede ser utilizado en cualquiera de los principales sistemas operativos existentes en el mercado, incluyendo Linux, diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP,etc.) y

Macs. Además presenta facilidades para ejecutarse bajo Apache, IIS, AOLServer, Roxen y THTTTPD. Otra alternativa es configurarlo como modulo CGI.

PHP posee una sencilla integración con múltiples bases de datos. Aunque MySQL es la base de datos que mejor trabaja con PHP, puede conectarse también a PostgreSQL, Oracle, dbm, filepro, interbasem o cualquier otra base de datos compatible con ODBC(*Open Database Connectivity Standard*).(*Breve historia de PHP, 2006*)

También ofrece la posibilidad de usar una programación orientada a objetos con clases y herencia. Este lenguaje de programación posee variables, sentencias condicionales, bucles, funciones, entre otras. La sintaxis que utiliza, es similar a la de otros lenguajes muy extendidos como C y Perl. El código de PHP está incluido en tags especiales "<?,?>".

Entre sus habilidades (PHP) se incluyen, creación de imágenes, ficheros PDF y películas Flash (usando libswf y Ming). También se pueden presentar otros resultados, como XHTML y ficheros XML. PHP puede auto generar estos ficheros y grabarlos en el sistema de ficheros en vez de presentarlos en la pantalla. Puede usarse también como procesador CGI.

Al tratarse PHP de un *software* libre, puede descargarse y utilizarse en cualquier aplicación, personal o profesional, de manera completamente libre.

### 1.11.6 ASP

ASP es una tecnología desarrollada por Microsoft (MS) para crear páginas Web de contenido dinámico apoyándose en scripts ejecutados en el servidor. El código ASP que se escribe se ejecuta primero en el servidor y luego se envían sus resultados en HTML al usuario que los pidió. (*¿Que es ASP ?*

, 2002)

Las páginas de esta clase se guardan con la extensión .asp y pueden o no contener fragmentos de código HTML El lenguaje que se utiliza por defecto es VBScript (Visual Basic Script), pero se puede utilizar también Jscript. Es un lenguaje totalmente ampliable gracias a su compatibilidad con la tecnología ActiveX.

Las páginas ASP comienzan a ejecutarse cuando un usuario por medio del navegador solicita una página ASP. Esta solicitud llega al servidor el cual tiene alojada la página solicitada. El servidor procesa la página ASP y devuelve el resultado en código HTML al navegador del usuario, donde este lo visualiza.

ASP permite un cómodo acceso a bases de datos, y es muy versátil para la utilización de plantillas (templates) para la construcción de páginas Web dinámicas.

La tecnología supramencionada fue desarrollada por Microsoft, por lo que solo puede ejecutarse en servidores con Windows, hay que usarlo con el IIS (Internet Information Server) o el PWS (Personal Web Server). (MAURO, 2006)

### 1.12 Lenguaje seleccionado PHP

Posterior a la realización de un análisis detallado entre el PHP y el ASP, se decide utilizar el PHP embebido en el código HTML ya que:

- PHP esta soportado en la mayoría de las plataformas de Sistemas Operativos, es multiplataforma, mientras que ASP no lo es por ser propiedad de Microsoft.
- PHP es libre y no tiene costo oculto, lo que significa que cuando se adquiere incluye un sinnúmero de bibliotecas externas, que permiten al desarrollador el manejo de gráficos, archivos PDF, Flash, XML, etc. En caso de que no se tengan las bibliotecas, éstas se pueden encontrar gratis en Internet. En cambio ASP forma parte del *Internet Information Server* que viene integrado en Windows NT-2000 Server con su elevado costo de adquisición.
- Entre el PHP y el ASP existe una gran semejanza en cuanto a la forma de utilización, pero el PHP es más rápido, gratuito y multiplataforma.

### 1.13 PHP, MySQL y Apache

PHP, MySQL y Apache comparten múltiples características, por ello después de realizar un estudio se decidió utilizar esta combinación debido a que los tres son gratuitos, es decir son software libre, de gran popularidad, fácil manejo, potente y multiplataformas pues se pueden instalar sobre Windows o Unix.

Apache es un servidor con gran seguridad y unido a la seguridad implementada por MySQL, satisfacen totalmente los requerimientos de seguridad para el sistema propuesto. Juntos ofrecen un alto nivel de integración, además de ofrecer API (Aplicacion Program Interface) muy potente que permite el desarrollo de cualquier tipo de aplicación de gestión de datos.

### **1.14 Herramientas utilizadas**

Para sustentar la documentación de este trabajo se utilizó el Rational Rose Enterprise Edition 2003 como modelador visual de la notación UML (Unified Modeling Language) para la confección de los diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades.

Para el diseño de las páginas Web fue utilizado el Dreamweaver MX 2004 de la familia Macromedia, para el tratamiento de las imágenes de la aplicación se utilizó el Adobe Photoshop CS2 y para la edición del código PHP el NuSphere PhpED versión 4.6.2, que es un editor de código de páginas Web. Esta herramienta de desarrollo es muy cómoda a la hora de trabajar con html dinámico.

El cliente que se utilizó para manipular la base de datos MySQL Naviscat (MySQL GUI) versión 7.1.13. Facilita la edición de tablas, y presenta diversas funcionalidades para mantener las bases de datos, su uso es relativamente fácil. El servidor web utilizado es Apache en su versión 2.2.



### **Conclusiones**

En este capítulo se realizó una descripción del objeto de estudio y campo de acción, se detallaron los principales conceptos y definiciones que permitirán un mejor entendimiento del problema.

Después de un profundo análisis para respaldar el trabajo a realizar, se arribó a las conclusiones de utilizar:

- Como lenguajes de programación para desarrollar la aplicación Web PHP y Java Script, ambas embebidas en código HTML; para el almacenamiento de los datos se acordó utilizar MySQL por poseer grandes potencialidades con PHP y para los estilos y formatos de la aplicación Web se aprobó utilizar estilos CSS, por ser una tecnología de punta con múltiples ventajas. El servidor Web que se decidió utilizar es el Apache por ser un servidor potente, seguro y compatible con PHP y MySQL.
- Para desarrollar la herramienta se utilizara RUP con UML, pues ofrece mayores facilidades a los desarrolladores.



## **Capítulo 2. Características del sistema**

### **Introducción**

En este capítulo se hace referencia al estado actual del negocio y para poder entender el contexto en que se emplaza el sistema se necesita definir conceptos que se puedan agrupar en un el modelo de dominio como alternativa al modelo de negocio, en sistemas altamente centrados en tecnologías informáticas, debido a la poca estructuración de los procesos que describen el negocio.

Además se detalla la propuesta que trae esta investigación, enumerando los Requisitos Funcionales y No funcionales que debe tener el sistema, lo que permite hacer una concepción general del mismo, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

### **2.1 Estado actual del Negocio**

Para describir los procesos del negocio que se relacionan con el campo de acción de este trabajo, es necesario centrar la atención en el proceso de registro que se realiza en el Evaluador Teórico CISMA, desarrollado en la Universidad en colaboración con la empresa SIMPRO.

Este Evaluador Teórico consta de cuatro áreas de trabajo, una de ellas es el Área de Registro, en esta área es donde se registran los datos necesarios de todas aquellas personas que acuden a la realización del examen teórico para obtener la licencia de conducción. Actualmente en este software solamente se pueden registrar las personas de la nacionalidad para la que fue confeccionado el sistema, representando esto un problema en el momento que el software es solicitado por un cliente de otro país, pues hay que realizar constantes modificaciones en el Área de Registro.

Por otra parte resalta como una de las principales dificultades la imposibilidad de ofrecer un servicio de búsqueda para otras aplicaciones, además no se puede realizar una búsqueda de información de los ciudadanos previamente registrados.

### 2.2 Modelo del dominio

Teniendo en cuenta el bajo nivel de estructuración que presenta el negocio que se está estudiando y que está altamente centrado en tecnologías informáticas, se propone un modelo del dominio, ya que de manera visual permite mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

Para identificar todos los conceptos que se emplearán en el diagrama se utilizará un glosario de términos sobre los nombres.

- Se le denominará **Administrador** a todo aquel usuario que tenga permiso para acceder a todas las opciones que brinda el sistema.
- Se le denominará **Sistema Externo** a cualquier aplicación que solicite el servicio de realizar búsquedas en el sistema o consultar los reportes predefinidos del sistema.
- Se le denominará **Especificación del Ciudadano** a la aplicación donde se muestran un conjunto de características, donde se podrán seleccionar las características específicas para cada ciudadanía o adicionar una nueva característica.
- Se le denominará **Ciudadanía** a la aplicación que permitirá escoger el tipo de ciudadanía o adicionar una nueva ciudadanía.

- Se le denominará **Configuración de Ciudadano** a la aplicación donde se configuraran todos los datos que debe de tener un determinado ciudadano para poder ser registrado para realizar el examen teórico con el objetivo de obtener la licencia de conducción.
- Se le denominará **Ciudadano** a todos los datos recogidos de las personas registradas.
- Se le denominará **Controlador de Configuración de Ciudadanos** a la aplicación que tiene a su cargo administrar todo el proceso de Configuración de Ciudadanos y Ciudadanos.
- Se le denominará **Controlador de Gestión de Ciudadanos** a la aplicación donde se podrán registrar, eliminar y modificar los datos de los ciudadanos.
- Se le denominará **Controlador de Búsqueda** a la aplicación encargada de realizar búsquedas y devolver como resultado Reportes de Búsqueda de los ciudadanos registrados.
- Se le denominará **Solicitudes Externas** a todas aquellas solicitudes que se realicen por otras aplicaciones de acceder al sistema.

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.

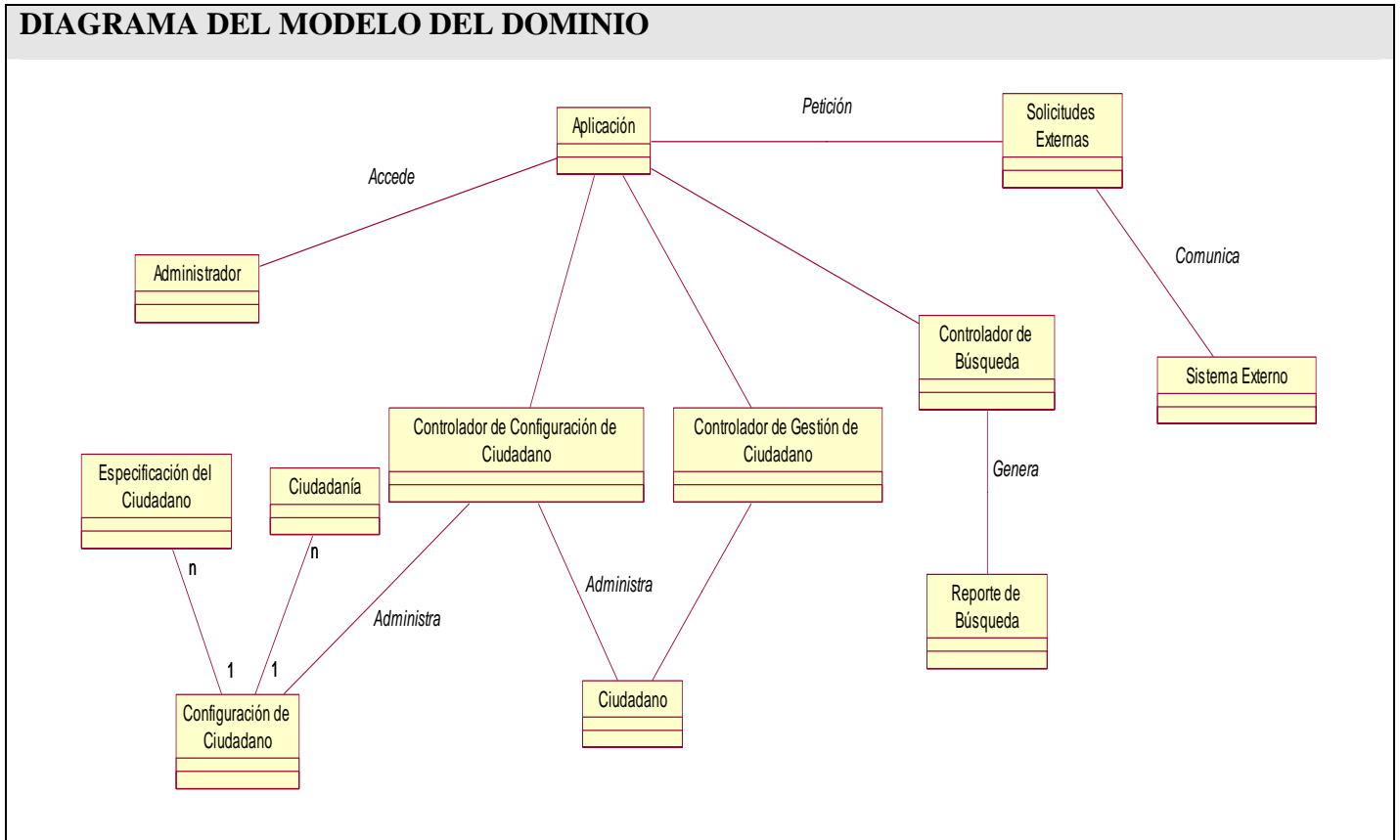


Figura 2.1: Diagrama del modelo del dominio.

### 2.3 Solución Propuesta

Para cumplimentar los objetivos propuestos al inicio de este trabajo la solución que se propone es el desarrollo de una Herramienta para Almacenar de forma Genérica Información de los Ciudadanos, y que forme parte del Evaluador Teórico CISMA. El sistema que se propone brinda las funcionalidades de administrar los usuarios del sistema, gestionar ciudadanía, gestionar los datos de los ciudadanos que incluye registrar, eliminar y modificar los mismos, mostrar reportes, realizar búsquedas y ofrecer servicios externos a otras aplicaciones.

La funcionalidad de administrar usuarios del sistema, permite crear, eliminar y modificar los usuarios que tendrán acceso al mismo.

La funcionalidad gestionar ciudadanía, posibilitará crear o escoger una ciudadanía y configurarla a partir de una serie de características que pueden estar predeterminadas o creadas por él, además podrá modificar los datos de una ciudadanía existente, así como eliminarla.

La funcionalidad gestionar datos de ciudadanos controla el proceso de registro de los ciudadanos que van a realizar el examen teórico, así como las modificaciones que se le pueden realizar a estos datos y la eliminación de ellos en caso que así lo requiera.

La funcionalidad de búsqueda se realizaran de forma simple por el criterio de búsqueda (nombre de la ciudadanía) obteniéndose todos los datos de los ciudadanos registrados. Esta funcionalidad formará parte del servicio que la aplicación brindará a otras que lo necesiten.

La funcionalidad de cambiar contraseña es la encargada de permitir que el administrador pueda cambiar la contraseña de acceso al sistema.

El sistema también mostrará reportes de la cantidad de ciudadanos registrados por ciudadanía, sexo y rangos de edad.

En resumen con este sistema, se puede registrar cualquier ciudadano sin importar su tipo de ciudadanía y ofrece nuevas funcionalidades, logrando que el proceso de registro del Evaluador Teórico CISMA sea más eficiente.

### **2.4 Requisitos funcionales**

Una vez conocidos los conceptos que rodean al objeto de estudio, se puede comenzar a analizar ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo?, para ello enumeraremos a través de requerimientos funcionales las funciones que el sistema deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el sistema. De acuerdo con los objetivos planteados el sistema debe ser capaz de:

**R1. Autenticar Usuario.**

- 1.1. El sistema debe ser capaz de permitir al usuario introducir su nombre de usuario y contraseña para ingresar al sistema.
- 1.2. El sistema debe validar los datos introducidos, comprobando que el usuario y la contraseña sean correctos y que la contraseña no exceda de ocho caracteres.

**R2. Administrar Usuario.**

- 2.1. El sistema debe de permitir al administrador crear nuevos usuarios del sistema.
- 2.2. El sistema debe de permitir al administrador modificar los datos de los usuarios del sistema.
- 2.3. El sistema debe de permitir al administrador eliminar usuarios del sistema.

**R3. Gestionar Ciudadanía.**

- 3.1. El sistema debe de permitir crear una nueva ciudadanía.
- 3.2. El sistema debe permitir seleccionar los datos necesarios de cada ciudadanía para poder ser configurada.
- 3.3. El sistema debe de permitir ingresar nuevos datos para crear una ciudadanía.
- 3.4. El sistema debe permitir modificar los datos ingresados de una ciudadanía.
- 3.5. El sistema debe permitir eliminar una ciudadanía.

**R4. Gestionar datos de Ciudadanos.**

- 4.1. El sistema debe permitir registrar datos de ciudadanos.
- 4.2. El sistema debe de permitir eliminar datos de los ciudadanos registrados.
- 4.3. El sistema debe de permitir realizar modificaciones a los datos de los ciudadanos registrados.

**R5. Realizar búsqueda de Ciudadanos.**

5.1. El sistema debe de permitir realizar búsquedas de ciudadanos sobre la clave de búsqueda: nombre de ciudadanía.

5.2. El sistema debe de permitir mostrar toda la información que se conoce de los ciudadanos obtenidos como resultado de la búsqueda.

**R6. El sistema debe de permitir una integración plena con otras aplicaciones para que utilicen la funcionalidad de búsqueda.**

**R7. El sistema debe de permitir mostrar reportes de la cantidad total de ciudadanos registrados según el tipo de ciudadanía y cantidad de ciudadanos por sexo, así como el promedio de edades registradas.**

**R8. El sistema debe de permitir al administrador poder cambiar su contraseña.**

**R9. El sistema debe de permitir que el usuario registrado cierre su sesión de trabajo desde cualquier lugar del sistema.**

**2.5 Requerimientos No Funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

- **Apariencia o interfaz externa:** La interfaz del sistema debe ajustarse a los estándares establecidos para el desarrollo de un buen diseño. Debe de estar diseñada de modo tal que el usuario pueda ir de un punto a otro dentro de ella con gran facilidad, estando visibles todas las opciones disponibles. Debe tratarse de que la aplicación sea lo más interactiva posible. Conviene realizar un diseño perfectamente encuadrado para resoluciones de 1024x768, pero preparado para verse en otras resoluciones.



- **Usabilidad:** El sistema podrá ser usado por cualquier tipo de personas que posean conocimientos básicos en el manejo de la computadora y el ambiente Web en sentido general.
- **Portabilidad:** El sistema debe de ser multiplataforma para que pueda funcionar en el sistema operativo Windows o Linux.
- **Seguridad:** El sistema debe de identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema. Garantizar que la información sea vista únicamente por quien tiene derecho a verla.
- **Soporte:** El sistema debe de tener una garantía de instalación y prueba del mismo, además de un breve entrenamiento a los futuros usuarios.
- **Software:** El sistema debe de permitir que los requerimientos en el lado del cliente para la utilización de la aplicación solo se limiten a tener disponible un navegador Web. Para la implantación de la herramienta debe de requerirse de un Servidor Web Apache y MySQL 3.27 o superior como servidor de Bases de Datos.
- **Hardware:** La máquina servidora debe tener como mínimo las siguientes características de hardware: Procesador Pentium II 450 MHz o superior, 128 Mb de memoria RAM (incluye la utilizada por el SO) y 2Gb de capacidad en disco duro.

### 2.6 Modelo de casos de uso del sistema

El modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. Los actores representan los usuarios del sistema y otras aplicaciones que interactúan con él, es decir, representan terceros fuera del sistema que se relacionan con él. Estos suelen corresponderse con trabajadores o actores del negocio. Los actores definidos en el sistema son los siguientes:

Actores	Justificación
Administrador	Es un usuario que tiene derecho a realizar todas las operaciones definidas en el sistema.
Sistema Externo	Es una aplicación que interactúa con el sistema.

Tabla 2-1. Actores del Sistema.

### 2.6.1 Diagrama de Casos de Uso del sistema

El diagrama donde se representa la relación existente entre los actores y los casos de uso se representa a continuación:

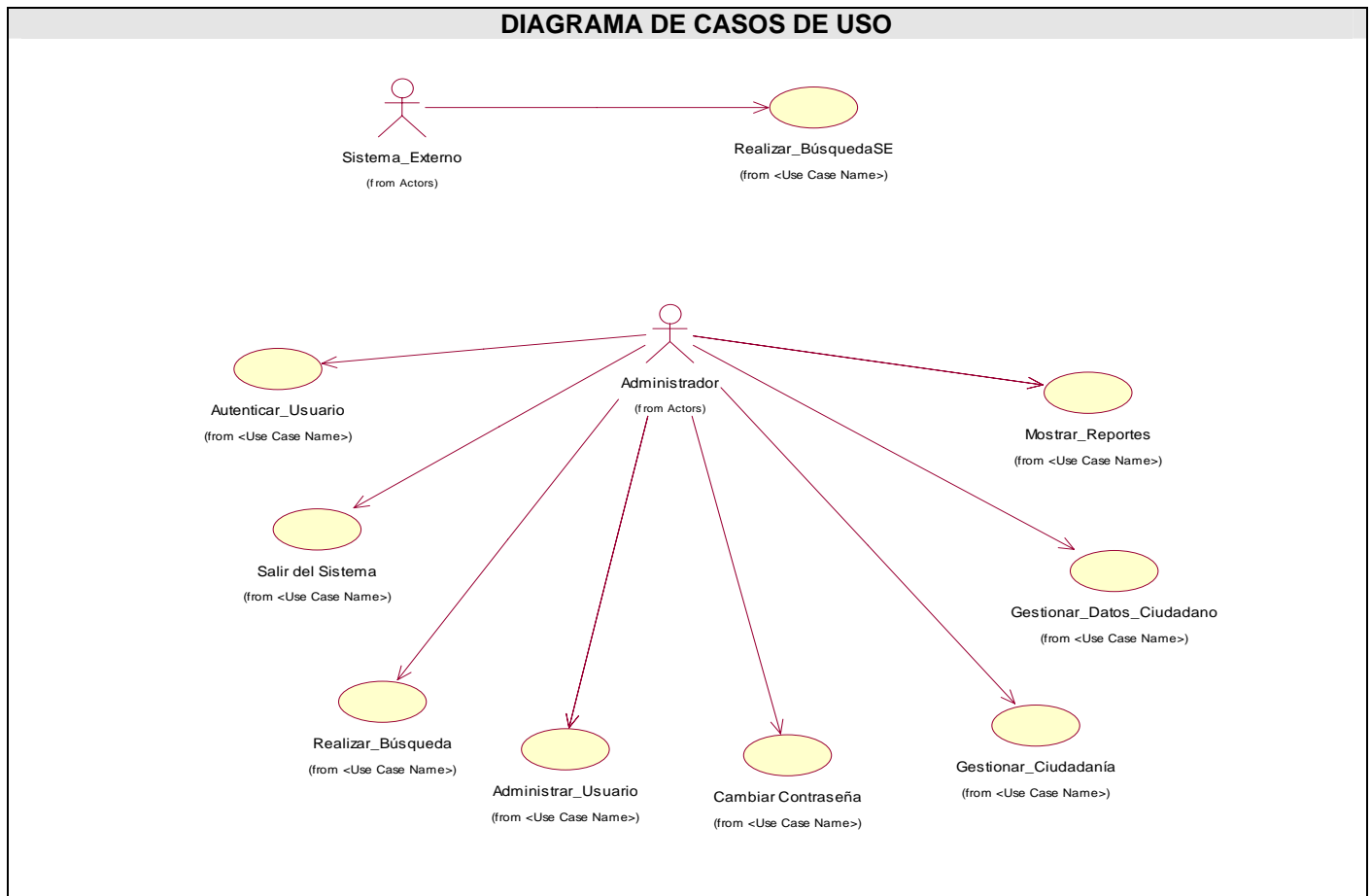


Figura 2.2: Diagrama global de casos de uso del sistema.

### 2.6.2 Descripción de los casos de uso del sistema

A continuación se presentan los casos de uso determinados para satisfacer los requerimientos funcionales del sistema:

<b>CU1</b>	<b>Autenticar Usuario.</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	Brinda la posibilidad de entrar en el sistema, el Administrador proporciona su nombre de usuario y contraseña luego el sistema se encarga de validar si tiene derecho o no de entrar.
<b>Referencia</b>	RF 1

<b>CU2</b>	<b>Realizar Búsqueda.</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	Posibilita que el administrador pueda realizar búsquedas de información de los ciudadanos registrados según la ciudadanía.
<b>Referencia</b>	RF 5

<b>CU3</b>	<b>Mostrar Reportes.</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	El Administrador puede obtener un reporte predefinido con información de la cantidad total de ciudadanos registrados según el tipo de ciudadanía, sexo y promedio de edades registradas.
<b>Referencia</b>	RF 7

<b>CU4</b>	<b>Gestionar Ciudadanía.</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	El Administrador puede crear una ciudadanía y configurarla a partir de una serie de características predeterminadas o creadas por él, además de poder modificarle sus datos o eliminar las ciudadanía existentes.
<b>Referencia</b>	RF 3

<b>CU5</b>	<b>Gestionar Datos Ciudadanos.</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	El Administrador controla el proceso de registro de los ciudadanos que van a realizar el examen teórico, así como las modificaciones que se le pueden realizar a estos datos y la eliminación de ellos en caso que lo requiera.
<b>Referencia</b>	RF 4

<b>CU6</b>	<b>Administrar Usuario.</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	El Administrador puede crear nuevos usuarios del sistema, modificar sus datos y eliminarlos.
<b>Referencia</b>	RF 2

<b>CU7</b>	<b>Cambiar Contraseña.</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	El Administrador puede cambiar su contraseña para tener acceso al sistema.
<b>Referencia</b>	RF 8

<b>CU8</b>	<b>Salir del Sistema.</b>
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador puede salir del sistema cuando estime conveniente desde cualquier lugar de la aplicación.
<b>Referencia</b>	RF 9

<b>CU9</b>	<b>Realizar_BúsquedaSE.</b>
<b>Actor</b>	Sistema Externo
<b>Descripción</b>	El Sistema Externo puede realizar la petición del servicio de búsqueda de información de los ciudadanos registrados de una determinada ciudadanía a la aplicación, enviándole esta la respuesta con los resultados de la petición realizada.
<b>Referencia</b>	RF 6

### 2.6.3 Expansión de los Casos de Uso

Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. A continuación se describen en formato expandido los casos de uso.

<b>Caso de uso</b>	
<b>CU1</b>	<b>Autenticar Usuario.</b>
<b>Propósito</b>	Proporcionar al administrador la entrada al sistema y garantizar que en todo momento se conozca quién está accediendo al mismo.
<b>Actores:</b> Administrador (Inicia)	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador necesita entrar al Sistema para el Registro de Ciudadanos, proporciona sus datos al sistema (nombre de usuario y contraseña). Se le informa si los datos fueron correctos o no y concluye así el caso de uso.	
<b>Referencias</b>	RF 1
<b>Precondiciones</b>	Existencia de Usuario y Clave válidos para el sistema.
<b>Poscondiciones</b>	El administrador es autenticado correctamente.
 <p><b>Pantalla 1</b></p>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Administrador necesita entrar al sistema.	1.1 El sistema muestra la interfaz correspondiente para que el administrador introduzca sus datos (ver Pantalla 1. Usuario (A) y Contraseña (B)).
2. El Administrador Introduce los datos requeridos y presiona el botón Enviar (C) para esperar respuesta del sistema.	2.1 El sistema valida la identidad del administrador y lo envía directamente a las principales opciones a las que tiene acceso según el tipo de privilegio.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2.1 En caso que la autenticación se realice de forma incorrecta se muestra un mensaje de error y retorna a la acción 1.
<b>Puntos de extensión.</b>	
<b>Prioridad</b>	Crítico

<b>Caso de uso</b>	
<b>CU2</b>	<b>Realizar Búsqueda.</b>
<b>Propósito</b>	Brinda la posibilidad al usuario de recuperar información de uno o varios ciudadanos.
<b>Actores:</b> Administrador(Inicia)	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador necesita buscar información de los ciudadanos registrados por ciudadanía en el sistema.	
<b>Referencias</b>	RF 5
<b>Precondiciones</b>	El Usuario tiene que estar autenticado satisfactoriamente por el sistema.



Pantalla 1

Acción del actor	Respuesta del sistema
1. El Administrador desea obtener información de los ciudadanos registrados en el sistema.	1.1. El sistema muestra la interfaz correspondiente al buscador (ver Pantalla 1).

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2. El administrador selecciona la ciudadanía a buscar (A) y presiona el botón Buscar (B) para que se realice la búsqueda.	2.1 El sistema consulta en la base de datos y le presenta la Pantalla 2, con los resultados de la búsqueda mostrando todos los ciudadanos encontrados de la ciudadanía seleccionada.
3. El administrador obtiene la información solicitada y concluye así el caso de uso.	



Pantalla 2

<b>Flujos alternativos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b> 2.1. En caso que el sistema no encuentre ningún resultado muestra un mensaje informando que no se encontraron ciudadanos registrados de esa ciudadanía. Y retorna a la acción 1.
<b>Puntos de Extensión</b>	
<b>Prioridad</b>	Secundario

<b>Caso de uso</b>	
<b>CU3</b>	<b>Mostrar Reporte</b>
<b>Propósito</b>	Brindar al administrador la posibilidad de obtener un reporte predefinido con información de la cantidad total de ciudadanos registrados por sexo, ciudadanía y promedio de edad.
<b>Actores:</b> Administrador (Inicia)	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador necesita obtener el reporte predefinido con información relacionada con la cantidad de personas registradas por ciudadanía, sexo y promedio de edad.	
<b>Referencias</b>	RF 7
<b>Precondiciones</b>	El Administrador tiene que estar autenticado satisfactoriamente por el sistema.



Pantalla 1


<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador desea poder obtener la información que se muestra en el reporte predefinido que presenta el sistema.	1.1 El sistema muestra la interfaz correspondiente al reporte predefinido (ver Pantalla 1), donde se recoge información de la cantidad de personas registradas

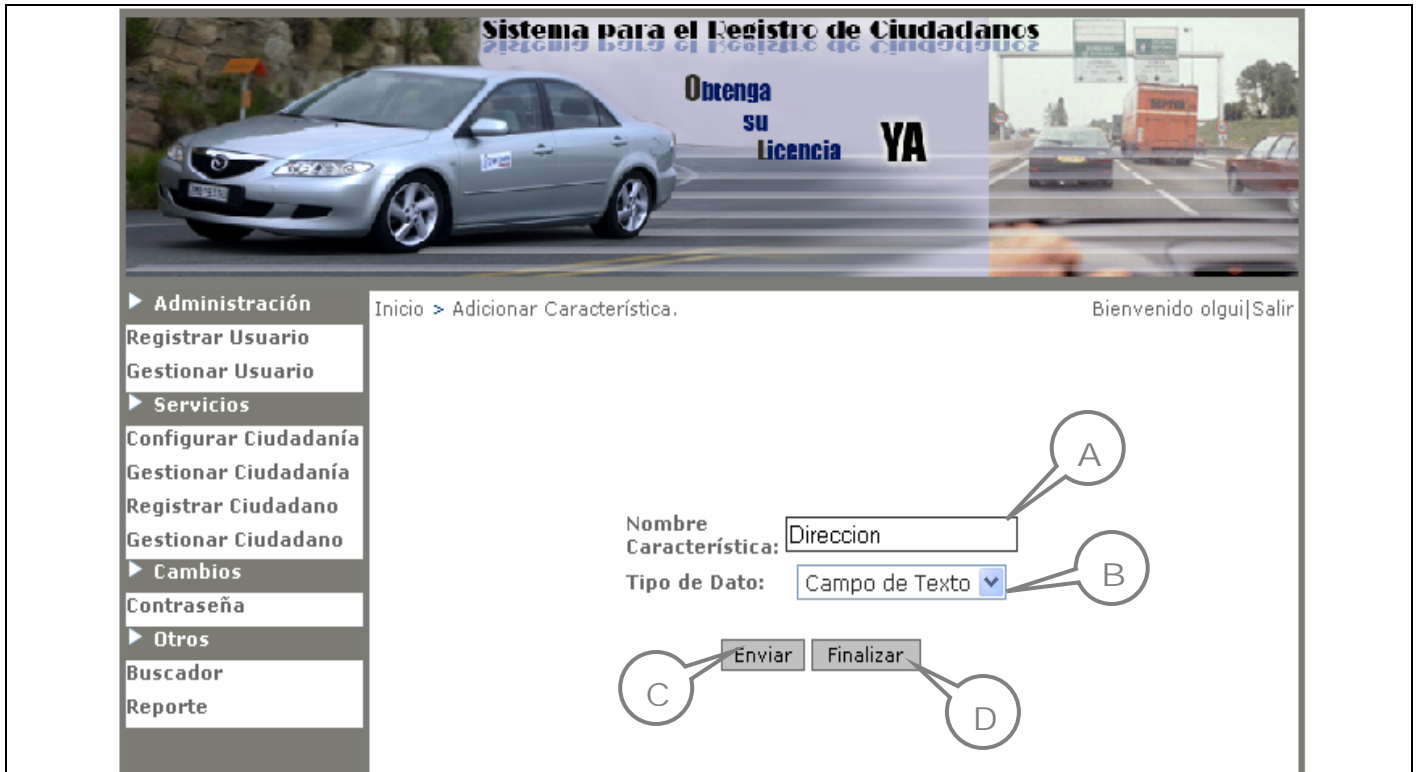


CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

	por sexo, ciudadanía y promedio de edad.
2. El administrador obtiene el reporte solicitado y concluye así el caso de uso.	
<b>Flujo alternativo</b>	
<b>Puntos de Extensión</b>	
<b>Prioridad</b>	Secundario

<b>Caso de uso</b>	
<b>CU4</b>	<b>Gestionar Ciudadanía.</b>
<b>Propósito</b>	Posibilitar al administrador la configuración de una nueva ciudadanía, así como la eliminación o actualización de las ciudadanía existentes.
<b>Actores:</b> Administrador (Inicia)	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador necesita crear y configurar una nueva ciudadanía, o cuando requiere eliminar o actualizar alguna de las ciudadanía ya existentes.	
<b>Referencias</b>	RF 3
<b>Precondiciones</b>	El administrador tiene que estar autenticado satisfactoriamente por el sistema.
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Administrador elige en el menú principal la actividad a realizar.	1.1 El sistema muestra la interfaz correspondiente de las actividades que se pueden realizar. <ul style="list-style-type: none"> <li>a) En caso que el Administrador desee configurar una nueva ciudadanía ir a Sección: <b>Configurar Ciudadanía.</b></li> <li>b) En caso que el Administrador desee adicionar una nueva característica ir a Sección: <b>Adicionar Característica.</b></li> <li>c) En caso que el Administrador desee eliminar una ciudadanía ir a Sección: <b>Eliminar Ciudadanía.</b></li> <li>d) En caso que el Administrador desee Modificar una ciudadanía ir a Sección: <b>Modificar Ciudadanía.</b></li> </ul>
2. El administrador realiza la actividad seleccionada, concluyendo así el caso de uso.	
<b>Sección “Configurar Ciudadanía.”</b>	

	<p style="text-align: center;"><b>Pantalla 1</b></p>
<p>2. El Administrador completa el campo con el nombre de la ciudadanía que desea configurar (A), selecciona las características pertenecientes a esa ciudadanía (B) y pulsa el botón crear (D) para que se registren los datos en la Base de Datos y terminar con el proceso de configuración de la ciudadanía.</p>	<p>1. El Sistema genera una interfaz correspondiente para configurar una nueva ciudadanía, mostrando una serie de características que el administrador debe de seleccionar para concluir con la configuración. (ver Pantalla 1).</p> <p>2.1. El sistema verifica que la ciudadanía no este registrada.</p> <p>2.2 El Sistema almacena los datos correctamente en la Base de Datos.</p>
<p><b>Sección: “Adicionar Característica.”</b></p>	



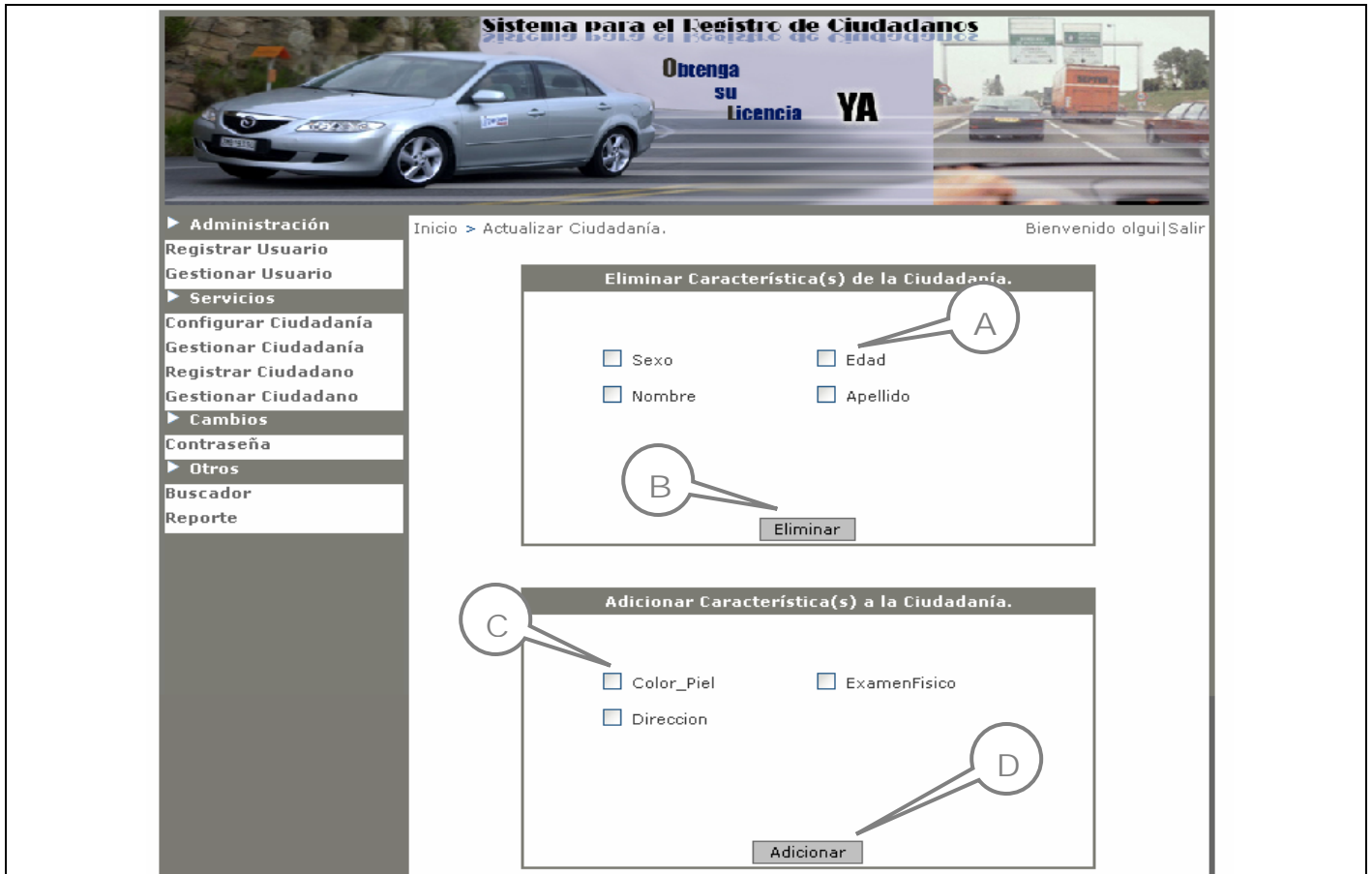
**Pantalla 2**

	<p>1. El Sistema genera una interfaz mostrando los campos que el administrador debe de seleccionar para adicionar una característica. (ver Pantalla 2).</p>
<p>2. El Administrador completa el campo con el nombre de la característica que desea adicionar (A), selecciona el tipo de dato perteneciente a esa característica (B) y pulsa el botón enviar (C) para que se registren los datos en la Base de Datos.</p>	<p>2.1. El sistema verifica que la característica no este registrada.</p> <p>2.2 El Sistema almacena los datos correctamente en la Base de Datos.</p>
<p>3. El Administrador pulsa el botón finalizar (D) y terminar con el proceso adicionar característica.</p>	
<p><b>Sección “Eliminar Ciudadanía.”</b></p>	



**Pantalla 3**

	1. El sistema genera la interfaz correspondiente para eliminar determinadas nacionalidades, mostrando una lista de todas las nacionalidades registradas (ver Pantalla 3).
2. El administrador selecciona la nacionalidad a eliminar (A) y pulsa el botón eliminar (B) para que se realice la operación en la Base de Datos.	2.1 El Sistema elimina los datos correctamente en la Base de Datos.
<b>Sección: “Modificar Nacionalidad.”</b>	
	1. El Sistema genera una interfaz que muestra una lista de las nacionalidades configuradas (ver Pantalla 3).
2. El administrador elige la nacionalidad a la que se le realizarán las modificaciones y pulsa en el vínculo Editar (C).	2.1. El Sistema genera la interfaz correspondiente para modificar la nacionalidad, mostrando sus características, las cuales podrá eliminar o adicionar alguna característica que esta no presente (ver Pantalla 4).



**Pantalla 4**

<p>3. El Administrador completa la operación para actualizar los datos de la ciudadanía, pulsando posteriormente un botón para registrar la información modificada en la Base de Datos.</p>	<p>3.1 El Sistema almacena los datos correctamente en la Base de Datos.</p>
---	---

<p><b>Flujo alternativo</b></p>	
<p><b>Sección “Configurar Ciudadanía.”</b></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
	<p>2.1 Si la ciudadanía ya se encuentra configurada se muestra un mensaje informativo. Y retorna a la acción 1.</p>
<p><b>Sección: “Adicionar Característica.”</b></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA


	2.1 Si la característica ya se encuentra registrada se muestra un mensaje informativo. Y retorna a la acción 1.
<b>Puntos de extensión.</b>	
<b>Prioridad</b>	Crítico

Caso de uso	
<b>CU5</b>	<b>Gestionar datos de Ciudadanos.</b>
<b>Propósito</b>	Posibilitar al administrador el registro de datos necesarios de los ciudadanos que deseen realizar el examen teórico para obtener la licencia de conducción, así como la modificación de estos datos o la eliminación de los mismos.
<b>Actores:</b> Administrador (Inicia)	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador necesita registrar los datos de algún ciudadano que solicite realizar el examen teórico de conducción o cuando necesite realizar modificaciones o eliminar los dato de algún ciudadano registrado.	
<b>Referencias</b>	RF 4
<b>Precondiciones</b>	El administrador tiene que estar autenticado satisfactoriamente por el sistema.
Acción del actor	Respuesta del sistema
1. El Administrador elige en el menú principal la actividad a realizar.	1.1 El sistema muestra la interfaz correspondiente de las actividades que se pueden realizar. <ul style="list-style-type: none"> <li>e) En caso que el Administrador desee registrar un ciudadano ir a Sección: <b>Registrar Ciudadano.</b></li> <li>f) En caso que el Administrador desee modificar los datos de algún ciudadano ir a Sección: <b>Modificar Datos del Ciudadano.</b></li> <li>g) En caso que el Administrador desee eliminar los datos de algún ciudadano registrado ir a Sección: <b>Eliminar Datos del Ciudadano.</b></li> </ul>
2. El administrador realiza la actividad seleccionada, concluyendo así el caso de uso.	
<b>Sección “Registrar Ciudadano.”</b>	

<ul style="list-style-type: none"> <li>▶ Administración</li> <li>Registrar Usuario</li> <li>Gestionar Usuario</li> <li>▶ Servicios</li> <li>Configurar Ciudadanía</li> <li>Gestionar Ciudadanía</li> <li>Registrar Ciudadano</li> <li>Gestionar Ciudadano</li> <li>▶ Cambios</li> <li>Contraseña</li> <li>▶ Otros</li> <li>Buscador</li> <li>Reporte</li> </ul>	<p>Inicio &gt; Escoger Ciudadanía para Registrar Ciudadano. <span style="float: right;">Bienvenido dayi  Salir</span></p> <div style="text-align: center;"> <p><b>Ciudadanías</b></p> <p>---Seleccione---</p> <p>Crear</p> </div> <div style="border: 1px solid gray; padding: 5px; margin: 10px auto; width: 80%;"> <p>Para realizar el registro de un ciudadano, seleccione la ciudadanía correspondiente al mismo y presione el botón crear, para generar el Formulario de Datos Personales.</p> <p>Una vez que el formulario esté generado, complete los campos con la información correspondiente y presione el botón enviar.</p> </div>
<p><b>Pantalla 1</b></p>	
<p>2. El Administrador elige la ciudadanía (A) correspondiente al ciudadano que se le tomaran los datos necesarios para realizar el examen teórico de conducción y presiona el botón Crear (B) para que se genere el formulario de registro de esa ciudadanía seleccionada.</p>	<p>1. El Sistema genera una interfaz que muestra una lista de todas las ciudadanías configuradas en el sistema (ver Pantalla 1).</p> <p>2.1. El sistema genera el formulario correspondiente a esa ciudadanía para registrar un ciudadano, mostrando los campos que deben de ser completados (ver Pantalla 2).</p>

**Formulario de Datos Personales.**

Nombre:	<input type="text" value="Mario"/>
Apellido:	<input type="text" value="Contrera"/>
Edad:	<input type="text" value="26"/>
Dirección:	<input type="text" value="calle 4"/>
Curso_EducaciónVial:	<input type="text" value="Aprobado"/>
Cédula:	<input type="text" value="MaCo345678"/>
Evaluación_MedicaIntegral:	<input type="text" value="Si"/>
Comprobante_PagoDerecho:	<input type="text" value="Si"/>
Sexo:	<input type="text" value="M"/>



**Pantalla 2**

3. El Administrador completa cada campo con la información correspondiente y pulsa el botón enviar (A) para que se registren los datos en la Base de Datos.

3.1 El sistema verifica que el ciudadano no este registrado.

3.2 El Sistema almacena los datos correctamente en la Base de Datos.

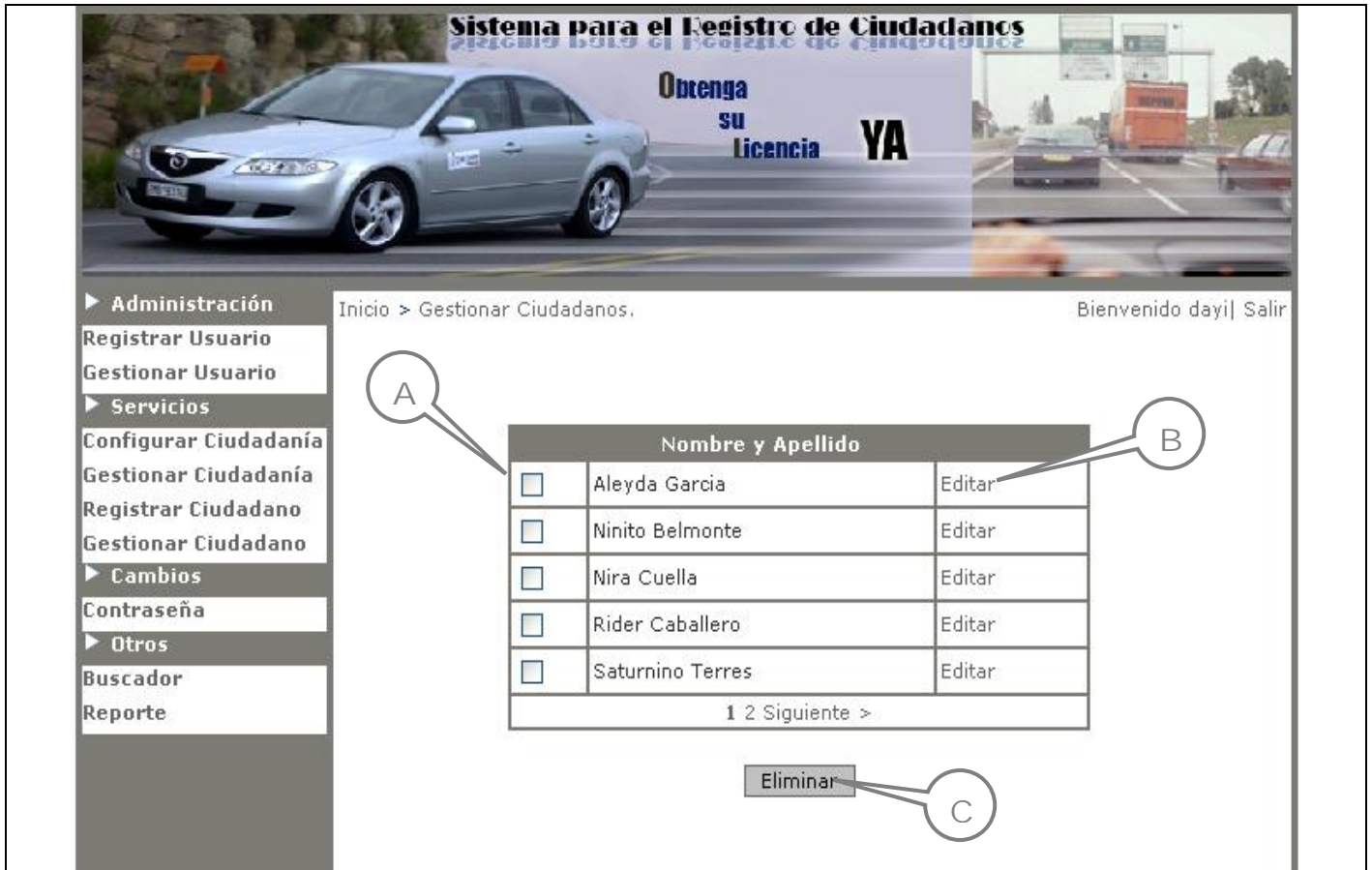
**Sección “Modificar Datos del Ciudadano.”**





**Pantalla 3**

	<p>1. El Sistema genera una interfaz para seleccionar el nombre de la ciudadanía de los ciudadanos que se le desean hacer modificaciones (ver Pantalla 3).</p>
<p>2. El Administrador selecciona el nombre de la ciudadanía (A) y pulsa el botón enviar (B).</p>	<p>2.1 El sistema envía los datos a la Base de Datos y genera una interfaz mostrando la lista de los ciudadanos que han sido registrados (ver Pantalla 4).</p>




**Pantalla 4**

2. El administrador elige el ciudadano al que se le realizarán las modificaciones y pulsa en el vínculo Editar (B).

2.1. El Sistema genera la interfaz correspondiente para modificar los datos, mostrando los campos que pueden ser modificados para actualizar los datos del ciudadano.

### Modificar Datos del Ciudadano.

Nombre:	<input type="text" value="Rider"/>
Apellido:	<input type="text" value="Caballero"/>
Edad:	<input type="text" value="26"/>
Dirección:	<input type="text" value="calle Rosa Vazque"/>
Curso_EducaciónVial:	<input type="text" value="Aprobado"/> ▼
Cédula:	<input type="text" value="RiCa78907"/>
Evaluación_MedicaIntegral:	<input type="text" value="Si"/> ▼
Comprobante_PagoDerecho:	<input type="text" value="Si"/> ▼
Sexo:	<input type="text" value="M"/> ▼

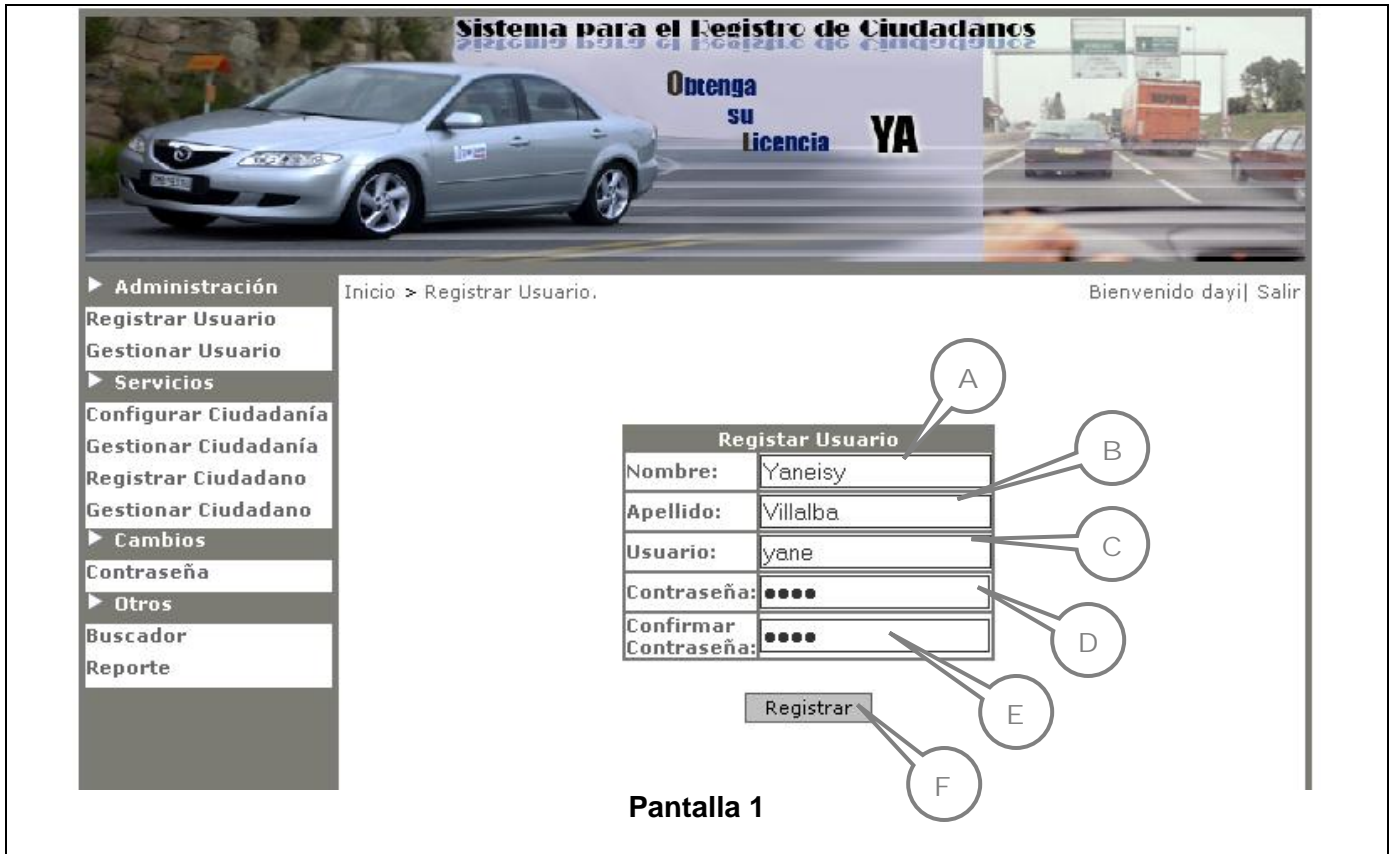


**Pantalla 5**

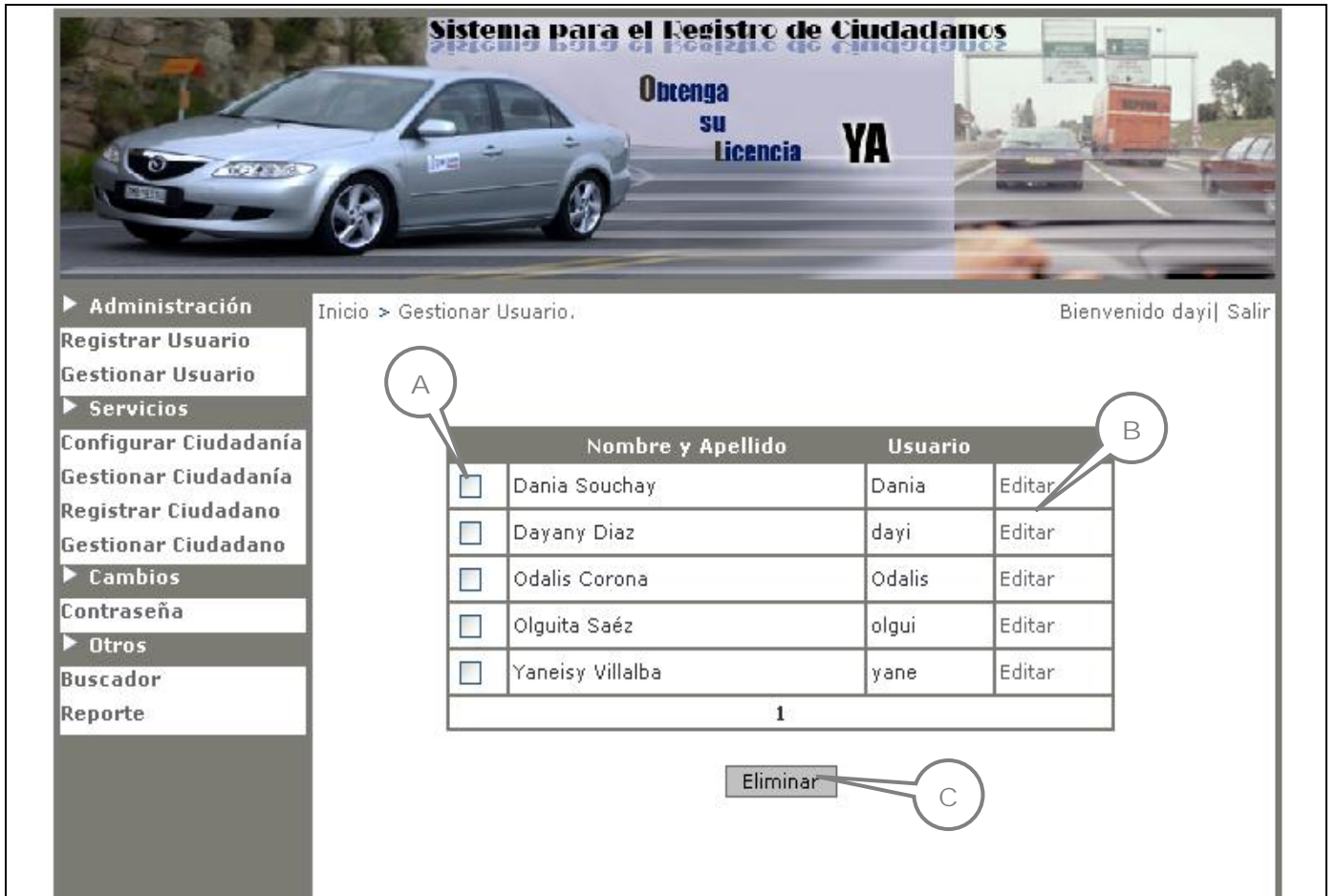
<p>3. El Administrador completa cada campo con la información correspondiente para actualizar los datos del ciudadano, pulsando posteriormente el botón actualizar para registrar la información modificada en la Base de Datos.</p>	<p>3.1. El sistema verifica que los campos obligatorios estén llenados correctamente.</p> <p>3.2 El Sistema almacena los datos correctamente en la Base de Datos.</p>
<b>Sección “Eliminar Datos del Ciudadano.”</b>	
	<p>1. El sistema genera la interfaz correspondiente para eliminar un ciudadano, mostrando una lista de todos los ciudadanos registrados por ciudadanía (ver Pantalla 4).</p>
<p>2. El administrador selecciona el ciudadano a eliminar (A) y pulsa el botón eliminar (C) para que se realice la operación en la Base de Datos.</p>	<p>2.1 El Sistema elimina los datos correctamente en la Base de Datos.</p>
<b>Flujo alternativo</b>	
<b>Sección “Registrar Ciudadano.”</b>	

<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	3.1. Si el ciudadano esta registrado se muestra un mensaje informativo. Y retorna a la acción 2.1.
<b>Puntos de extensión.</b>	
<b>Prioridad</b>	Crítico

<b>Caso de uso</b>	
<b>CU6</b>	<b>Administrar Usuario.</b>
<b>Propósito</b>	Proporcionar al administrador la posibilidad de crear nuevos usuarios en el sistema, así como modificar y eliminar los ya existentes.
<b>Actores:</b> Administrador (Inicia)	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador necesita crear un nuevo usuario, modificar o eliminar algún usuario ya existente.	
<b>Referencias</b>	RF 2
<b>Precondiciones</b>	El administrador tiene que estar autenticado satisfactoriamente por el sistema.
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El Administrador elige en el menú principal la actividad a realizar.	1.1 El sistema muestra la interfaz correspondiente de las actividades que se pueden realizar. a) En caso que el Administrador desee crear un nuevo usuario ir a Sección: <b>Crear Usuario del Sistema.</b> b) En caso que el Administrador desee modificar un usuario ir a Sección: <b>Modificar Usuario del Sistema.</b> c) En caso que el Administrador desee eliminar un usuario ir a Sección: <b>Eliminar Usuario del Sistema.</b>
2. El administrador realiza la actividad seleccionada, concluyendo así el caso de uso.	
<b>Sección “Crear Usuario del Sistema.”</b>	



	<p>1. El Sistema genera la interfaz correspondiente para crear un nuevo usuario, mostrando los campos que deben de ser completados para crear el usuario. (ver Pantalla 1).</p>
<p>2. El Administrador completa cada campo: Nombre(A), Apellido (B), Usuario(C), Contraseña (D) y Confirmar Contraseña (E) con la información correspondiente y pulsa el botón enviar (F) para que se registren los datos en la Base de Datos.</p>	<p>2.1 El sistema verifica que el nombre de usuario no exista.</p> <p>2.2 El Sistema almacena los datos correctamente en la Base de Datos.</p>
<p><b>Sección “Modificar Usuario del Sistema.”</b></p>	



**Pantalla 2**

	<p>1. El Sistema genera una interfaz que muestra una lista con los usuarios del sistema. (ver Pantalla 2)</p>
<p>2. El administrador elige el usuario al que se le realizarán las modificaciones y pulsa en el vínculo <i>Editar</i> (B).</p>	<p>2.1. El Sistema genera la interfaz correspondiente para modificar un usuario del sistema, mostrando los campos que pueden ser modificados para actualizar los datos del usuario (ver Pantalla 3).</p>



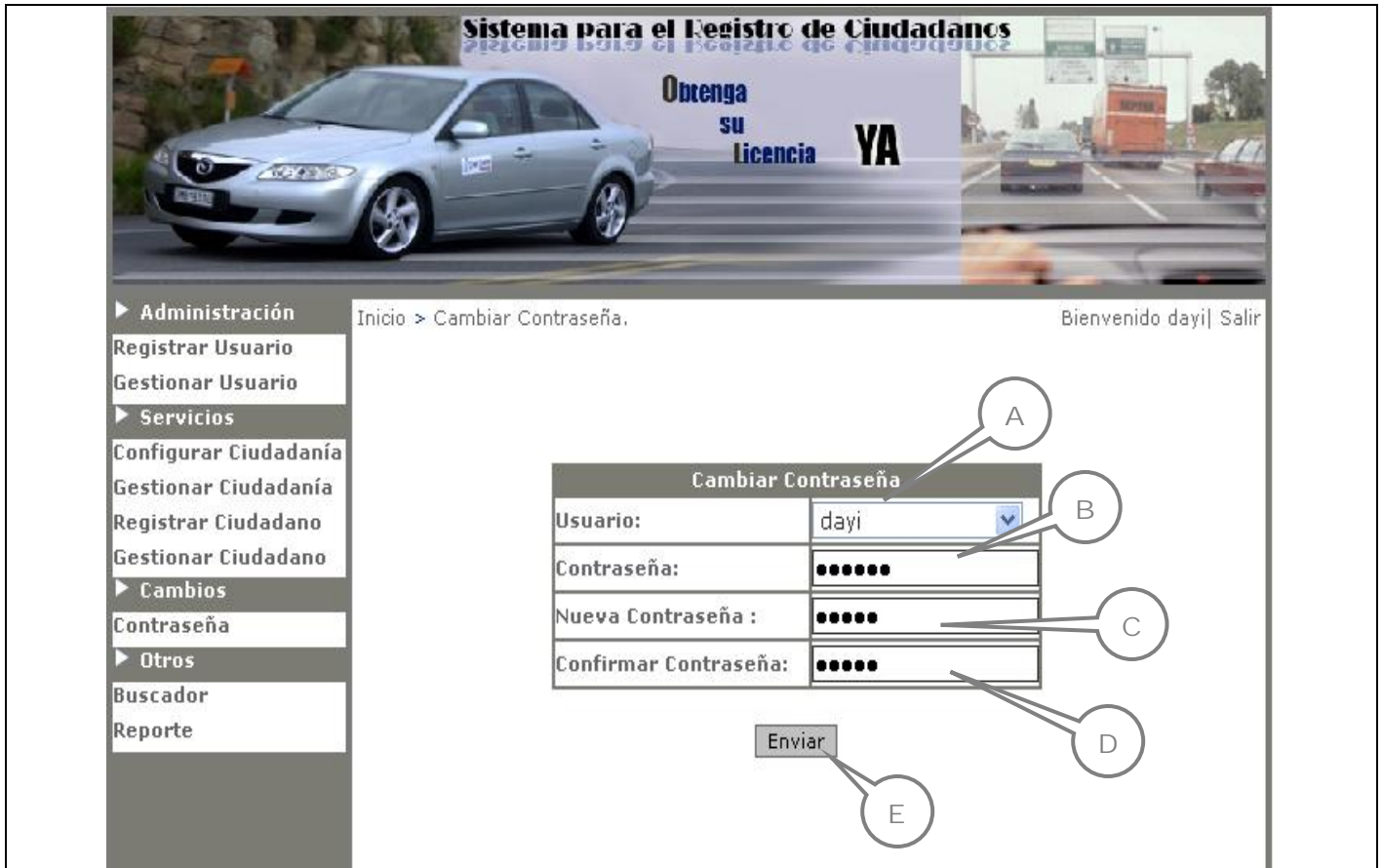
**Pantalla 3**

<p>3. El Administrador completa los campos: nombre(A), apellido (B) ,contraseña(C) y Confirmar Contraseña (D) con la información correspondiente para actualizar los datos del usuario, pulsando posteriormente el botón actualizar(E) para registrar los datos modificados en la Base de Datos.</p>	<p>3.1. El sistema verifica que los campos obligatorios estén llenados correctamente.</p> <p>3.2. El Sistema almacena los datos correctamente en la Base de Datos.</p>
<p><b>Sección “Eliminar Usuario del Sistema.”</b></p>	
	<p>1. El Sistema genera la interfaz correspondiente para eliminar un usuario del sistema.(ver Pantalla 2)</p>
<p>2. El Administrador selecciona el usuario que desea eliminar (A) y pulsa posteriormente el botón eliminar(C) para eliminar los datos en la Base de Datos.</p>	<p>2.1 El Sistema elimina los datos correctamente en la Base de Datos.</p>
<p><b>Flujo alternativo.</b></p>	

<b>Sección “Crear Usuario del Sistema.”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2.1 Si el nombre de usuario existe se muestra un mensaje informativo. Y retorna a la acción 1.
<b>Puntos de extensión.</b>	
<b>Prioridad</b>	Secundario

<b>Caso de uso</b>	
<b>CU7</b>	<b>Cambiar Contraseña</b>
<b>Propósito</b>	Brindar al administrador la posibilidad de cambiar su contraseña para acceder al sistema.
<b>Actores:</b> Administrador (Inicia)	
<b>Resumen:</b> El caso de uso se inicia cuando el Administrador necesita cambiar su contraseña para poder acceder al sistema.	
<b>Referencias</b>	RF 8
<b>Precondiciones</b>	El Administrador tiene que estar autenticado satisfactoriamente por el sistema.





Pantalla 1

Acción del actor	Respuesta del sistema
1. El Administrador desea poder cambiar su contraseña para poder acceder al sistema.	1.1 El sistema muestra la interfaz correspondiente a la opción cambiar contraseña, mostrando los campos que deben de ser llenados para poder efectuar la acción (ver Pantalla 1).
2. El Administrador selecciona el usuario(A) y llena los campos: contraseña (B), nueva contraseña (C), confirmar contraseña (D) con la información correcta y pulsa el botón enviar (E) para actualizar la información en la base de datos. Concluyendo así el caso de uso.	2.1 El sistema verifica que los campos obligatorios estén llenados correctamente.  2.2 Los datos son actualizados de forma satisfactoria en la base de datos.
<b>Flujo alternativo</b>	
<b>Puntos de Extensión</b>	
<b>Prioridad</b>	Secundario

Caso de uso	
<b>CU8</b>	<b>Salir del Sistema.</b>
<b>Propósito</b>	Salir de la interfase de Sistema para el Registro de Ciudadanos.
<b>Actores:</b> Administrador (Inicia)	
<b>Resumen:</b> Un administrador previamente autenticado en la interfase Sistema para el Registro de Ciudadanos selecciona la opción salir del sistema, el sistema cierra su sesión y lo envía hacia la interfase de autenticación.	
<b>Referencias</b>	RF 9
<b>Precondiciones</b>	El administrador debe estar dentro de la interfase Sistema para el Registro de Ciudadanos.
<b>Poscondiciones:</b>	El administrador se encuentra fuera de la interfase Sistema para el Registro de Ciudadanos.
<b>Prioridad</b>	Secundario

Caso de uso	
<b>CU9</b>	<b>Realizar_BúsquedaSE.</b>
<b>Propósito</b>	Proporcionar el servicio de búsqueda a otras aplicaciones.
<b>Actores: Sistema Externo</b> (Inicia)	
<b>Resumen:</b> El caso de uso inicia cuando el Sistema Externo solicita el servicio de búsqueda al Sistema de registro de Ciudadanos para obtener información de los ciudadanos registrados de la ciudadanía pedida. Este le ofrece el servicio concluyendo así el caso de uso.	
<b>Referencias</b>	RF 6
<b>Precondiciones</b>	El Sistema Externo debe enviar una petición.
<b>Poscondiciones:</b>	El Sistema Externo recibió el servicio solicitado.
<b>Prioridad</b>	Secundario.

## **Conclusiones**

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio, un listado con las propiedades que debe cumplir el sistema y las funciones que este debe de realizar, representándolas mediante un Diagrama de Casos de Uso del Sistema, y finalmente describiendo paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Es por ello que ahora se puede comenzar a construir el sistema, tratando de que se cumplan todos los requerimientos y las funciones que se han considerado necesarias en este capítulo.



## Capítulo 3. Desarrollo de la solución propuesta

### Introducción

En este capítulo se modelan los principales artefactos que ayudan a manejar las dificultades que implican la construcción de aplicaciones Web. Para su estudio más detallado se realizarán por cada caso de uso, diagramas de interacción, así como diagramas de clases Web, obteniendo como resultado el diagrama de clases persistentes y su modelo de datos, para construir la base de datos que soportará el trabajo del sistema. También se abordarán brevemente los patrones de diseño y arquitectura que se utilizarán, los principios del diseño y finalizando se presentarán el modelo de implementación mediante el diagrama de despliegue que resultó del diseño realizado de cada uno de los casos de uso del sistema y los casos de pruebas realizado a los casos de uso más significativos.

### 3.1 Patrones de diseño

Los patrones son parejas de problema/solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Asignar correctamente las responsabilidades es muy importante en el diseño orientado a objetos. (LARMAN)

En el diseño de la aplicación se tuvieron en cuenta los patrones GRASP (General Responsibility Assignment Software Patterns) principalmente los patrones Experto y Controlador. El primero plantea que siempre se debe asignar una responsabilidad al experto en información, o sea, la clase con toda la información necesaria para llevarla a cabo. El segundo expresa asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase controladora.

### 3.1.1 Patrón Modelo Vista Controlador

La arquitectura Modelo Vista Controlador (MVC) divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- **Modelo (Model):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- **Vista (View):** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador (Controller):** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("**service requests**") para el modelo o la vista.(RICARDO, 2007)

Se han desarrollado a lo largo de los años, desde la presentación de este patrón a la comunidad científica 3 variantes fundamentales, que se presentan brevemente a continuación.

#### **Variante I:**

Variante en la cual no existe ninguna comunicación entre el Modelo y la Vista y esta última recibe los datos a mostrar a través del Controlador.

#### **Variante II:** (Figura 3.1)

Variante en la cual se desarrolla una comunicación entre el Modelo y la Vista, donde esta última al mostrar los datos los busca directamente en el Modelo, dada una indicación del Controlador, disminuyendo el conjunto de responsabilidades de este último.

#### **Variante III:**

Variante en la cual se diversifica las funcionalidades del Modelo teniendo en cuenta las características de las aplicaciones multimedia, donde tienen un gran peso las medias utilizadas en estas.

Para la realización de la arquitectura del software se empleó la Variante II del patrón MVC.

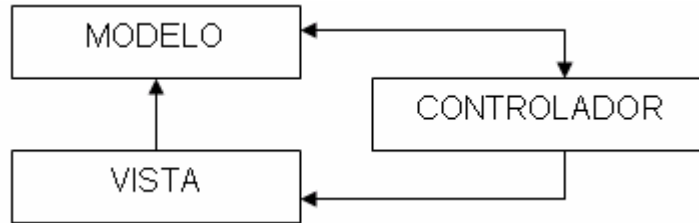


Figura 3.1 Variante Intermedia del Patrón MVC.

### 3.2 Diagrama de Clases Web

Cuando se habla de modelar aplicaciones Web con UML se puede apreciar que difiere un poco del resto de las aplicaciones que estamos acostumbrados a construir, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación.

Modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente sí es muy importante pues estos son los artefactos que se necesitan modelar para que el desarrollador los implemente luego y obtener así un producto final.

De acuerdo a la forma en que se ha organizado el contenido del trabajo, se considera que se debe presentar los modelos organizados por subsistemas (que coinciden con los casos de uso), de forma que se pueda entender mejor la lógica del negocio.

1. **Subsistema 1.** Autenticar Usuario:
2. **Subsistema 2.** Realizar Búsqueda:
3. **Subsistema 3.** Mostrar Reportes:
4. **Subsistema 4.** Gestionar Ciudadanía:

Configurar Ciudadanía.

Adicionar Característica.

Modificar Ciudadanía.

Eliminar Ciudadanía.

5. **Subsistema 5.** Gestionar datos de Ciudadanos:

Registrar Ciudadano.

Modificar Ciudadano.

Eliminar Ciudadano.

6. **Subsistema 6.** Administrar Usuario:

Registrar Usuario.

Modificar Usuario.

Eliminar Usuario.

7. **Subsistema 7.** Cambiar contraseña.

8. **Subsistema 8.** Salir del Sistema.

9. **Subsistema 9.** Realizar Búsqueda por aplicaciones externas.

A continuación se presentan los diagramas de clases del diseño de los subsistemas que representan los procesos y diagramas de interacción para la realización de los casos de usos descritos en el capítulo anterior.

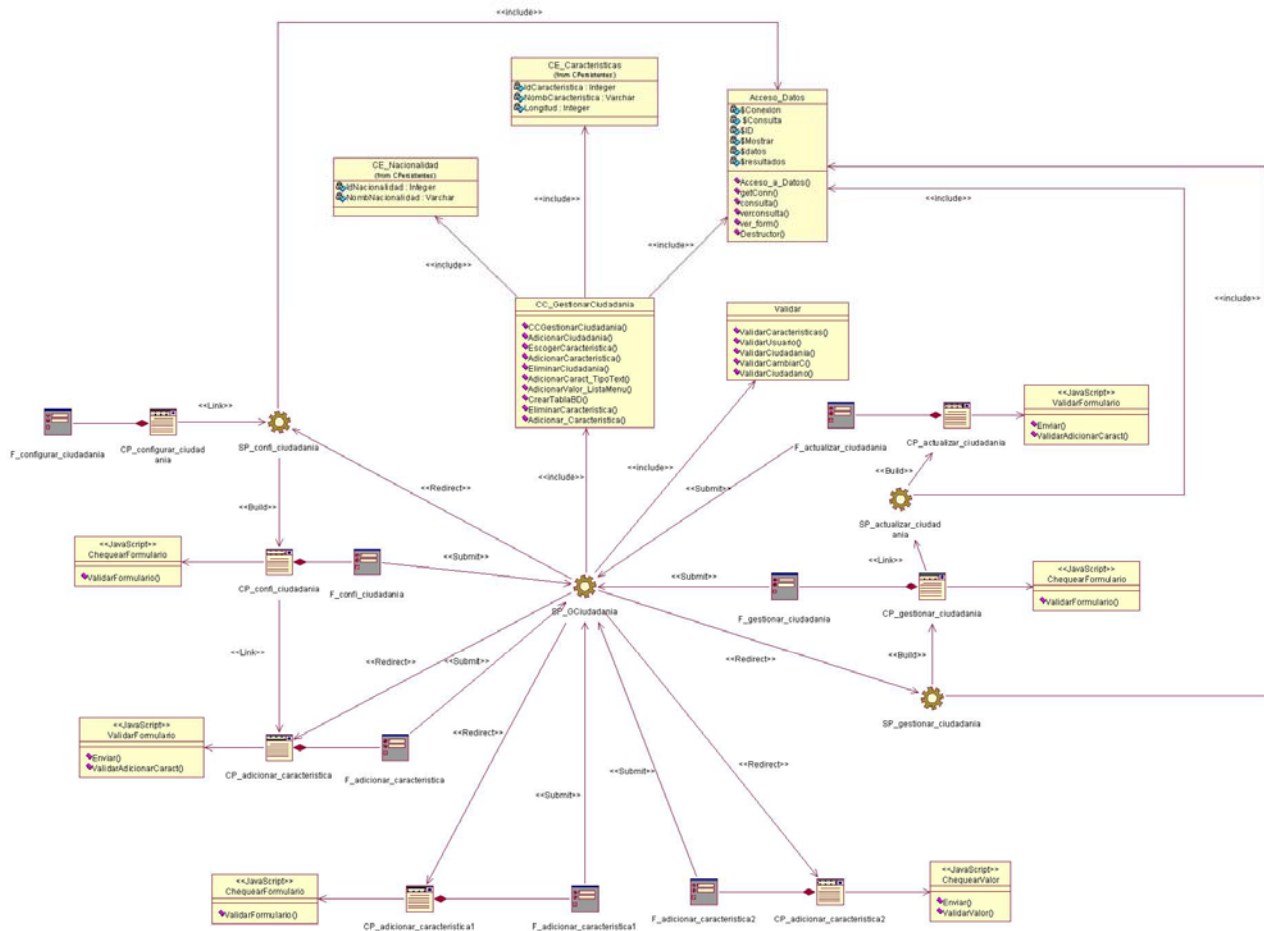


Figura 3.2 Diagrama de Clases del Diseño Gestionar Ciudadanía.

Diagrama de secuencia del caso de uso “Gestionar Ciudadanía”. Sección “Configurar Ciudadanía”. [Ver Anexo 1].

Diagrama de secuencia del caso de uso “Gestionar Ciudadanía”. Sección “Adicionar Característica”. [Ver Anexo 2].

Diagrama de secuencia del caso de uso “Gestionar Ciudadanía”. Sección “Eliminar Ciudadanía”. [Ver Anexo 3].



Diagrama de secuencia del caso de uso “Gestionar Ciudadanía”. Sección “Modificar Ciudadanía”. [Ver Anexo 4].

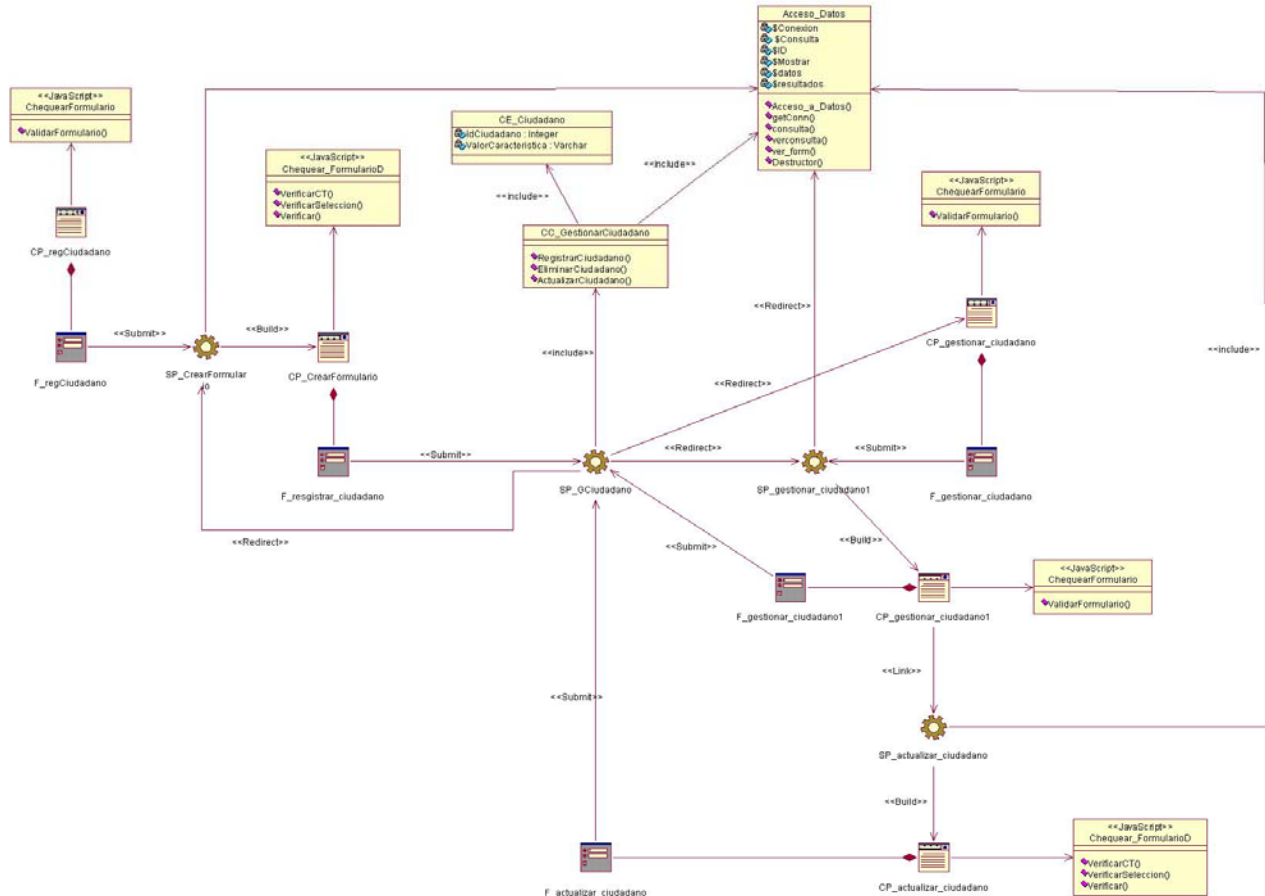


Figura 3.3 Diagrama de Clases del Diseño Gestionar Ciudadano.

Diagrama de secuencia del caso de uso “Gestionar Ciudadano”. Sección “Registrar Ciudadano”. [Ver Anexo 5].

Diagrama de secuencia del caso de uso “Gestionar Ciudadano”. Sección “Eliminar Ciudadano”. [Ver Anexo 6].

Diagrama de secuencia del caso de uso “Gestionar Ciudadano”. Sección “Modificar Ciudadano”. [Ver Anexo 7].

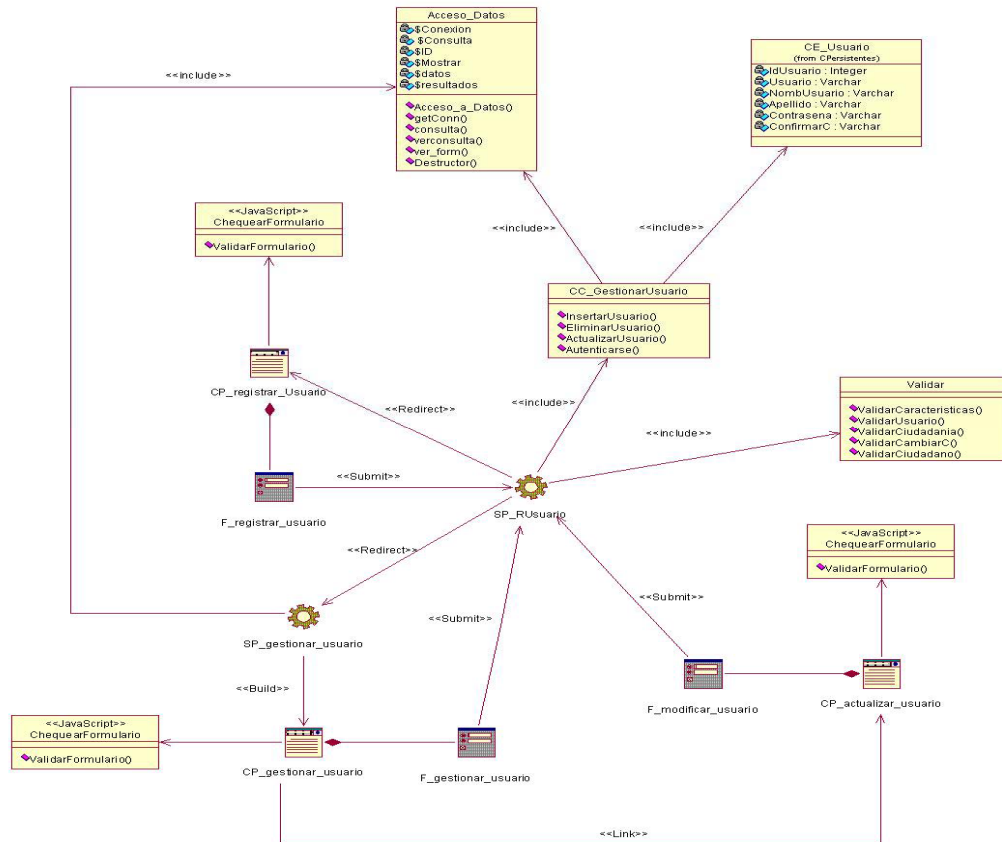


Figura 3.4 Diagrama de Clases del Diseño Administrar Usuario.

Diagrama de secuencia del caso de uso “Administrar Usuario”. Sección “Registrar Usuario”. [Ver Anexo 8].

Diagrama de secuencia del caso de uso “Administrar Usuario”. Sección “Modificar Usuario”. [Ver Anexo 9].

Diagrama de secuencia del caso de uso “Administrar Usuario”. Sección “Eliminar Usuario”. [Ver Anexo 10].

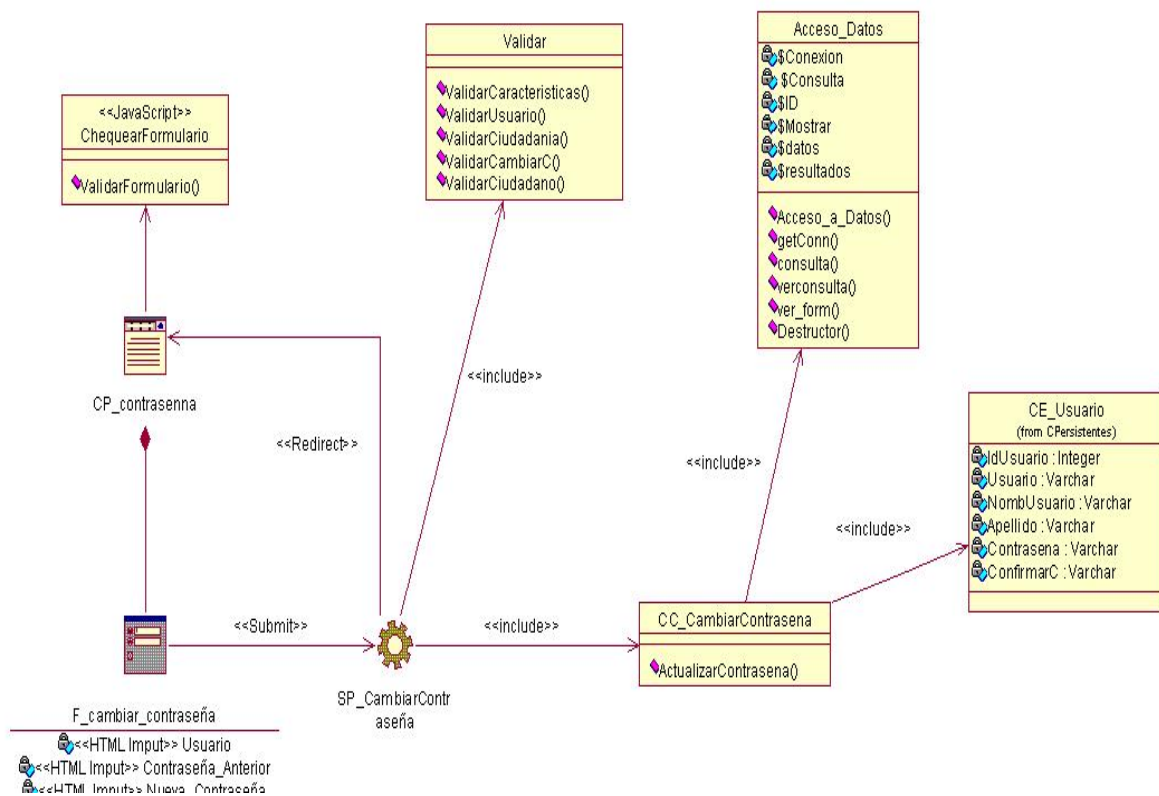


Figura 3.5 Diagrama de Clases del Diseño Cambiar Contraseña.

Diagrama de secuencia del caso de uso “Cambiar Contraseña”. [Ver Anexo11].

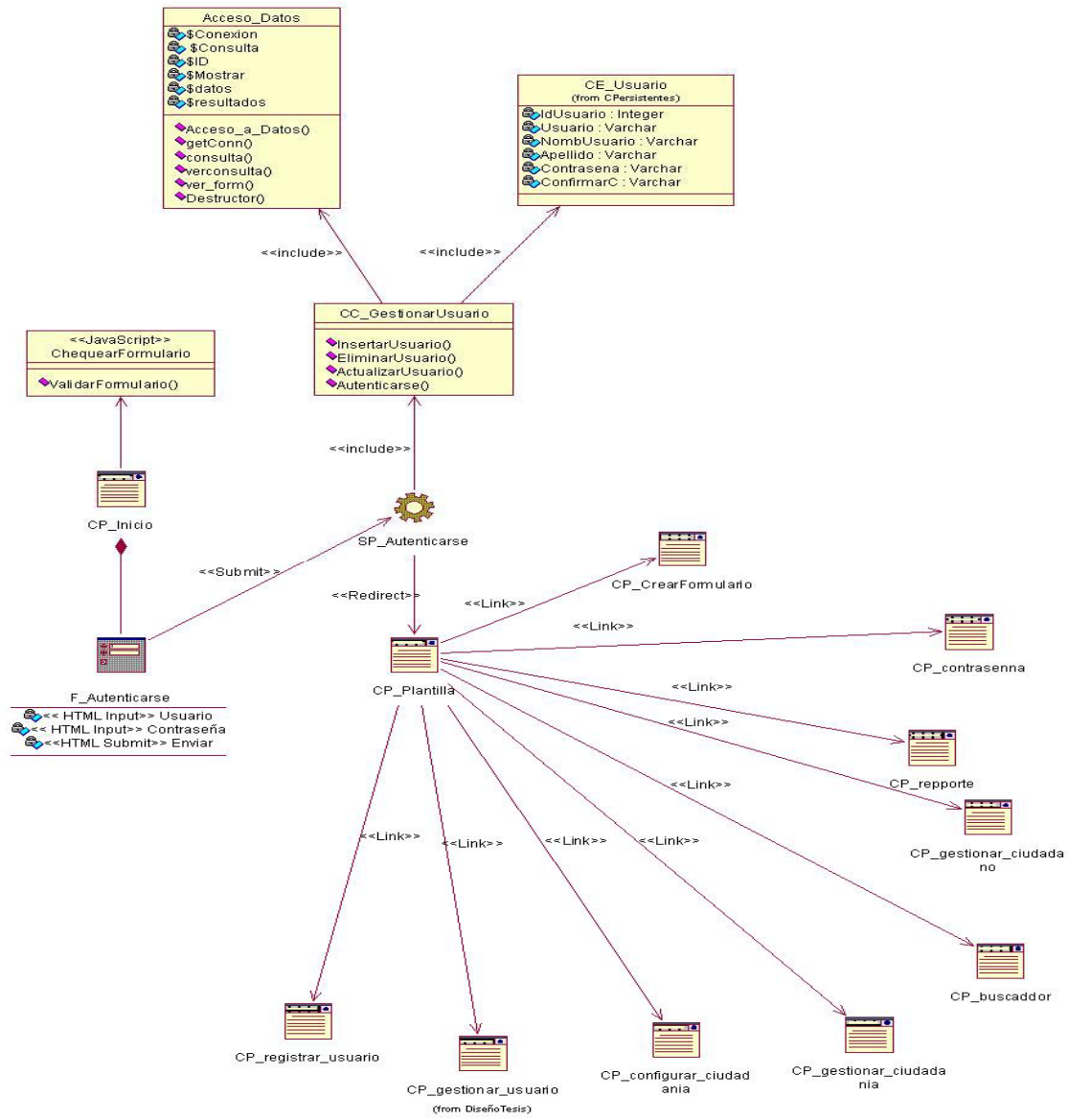


Figura 3.6 Diagrama de Clases del Diseño Autenticarse.

Diagrama de secuencia del caso de uso “Autenticarse”. [Ver Anexo12].

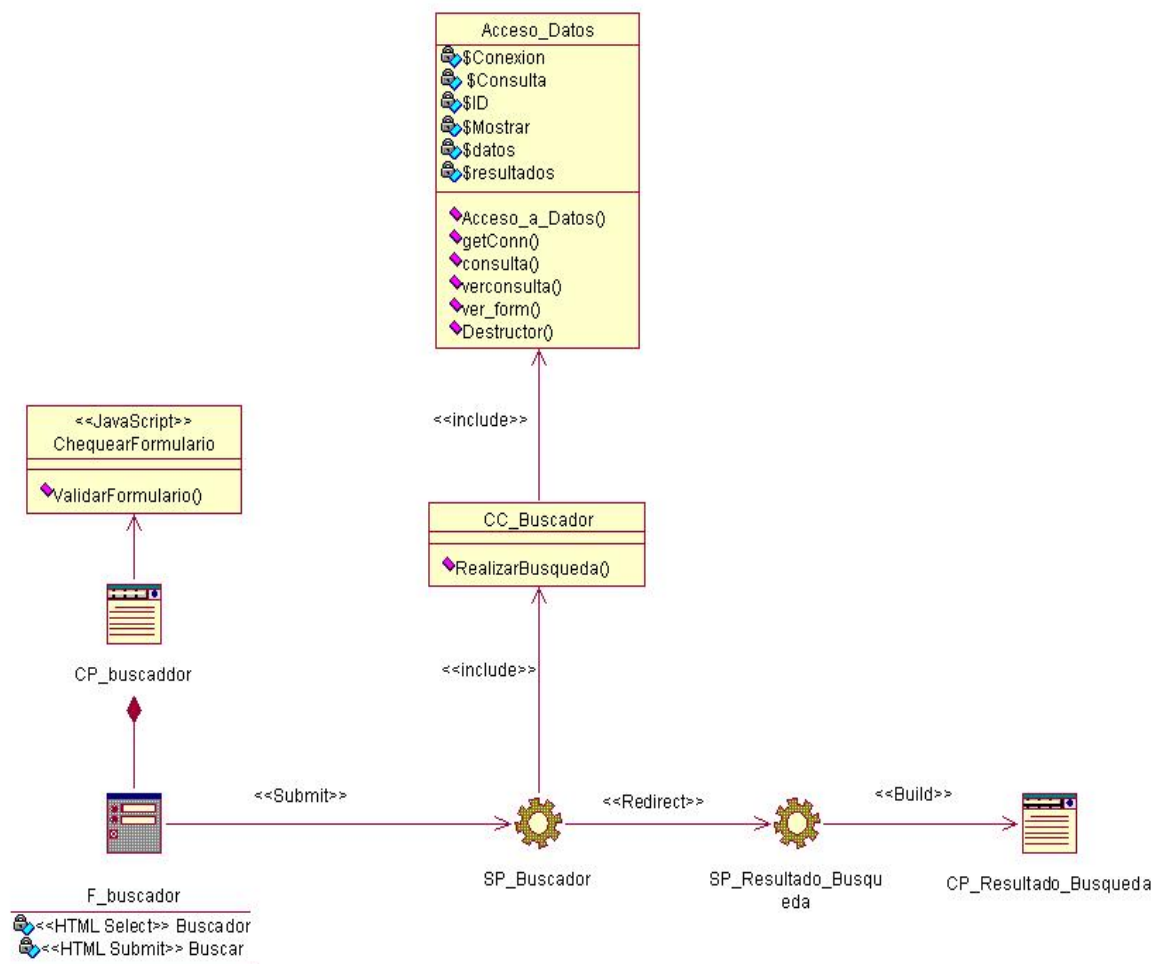


Figura 3.7 Diagrama de Clases del Diseño Realizar Búsqueda.

Diagrama de secuencia del caso de uso “Realizar Búsqueda”. [Ver Anexo13].

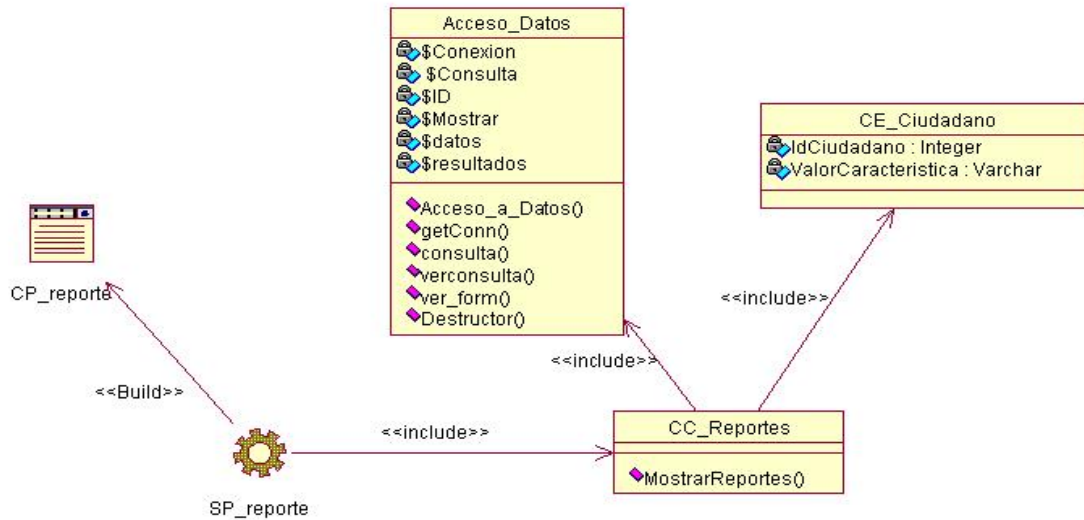


Figura 3.8 Diagrama de Clases del Diseño Mostrar Reporte.

Diagrama de secuencia del caso de uso “Mostrar Reporte”. [Ver Anexo14].

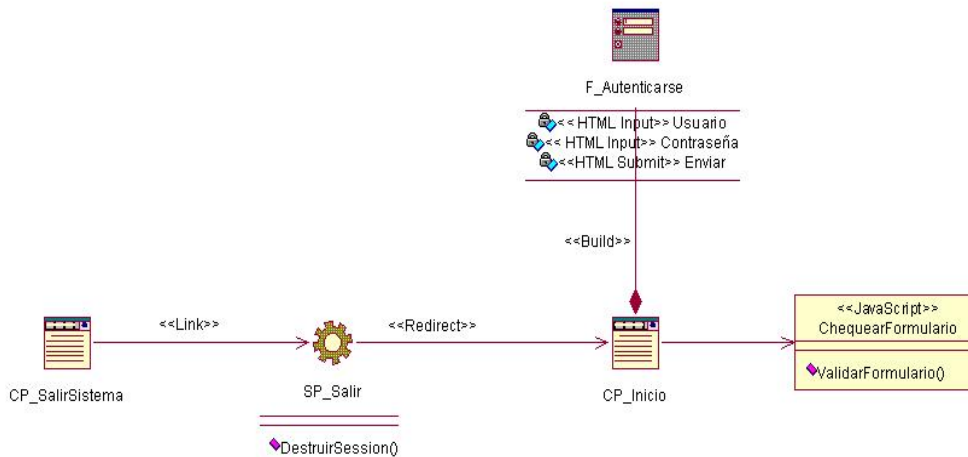


Figura 3.9 Diagrama de Clases del Diseño Salir del Sistema.

Diagrama de secuencia del caso de uso “Salir del Sistema”. [Ver Anexo15].

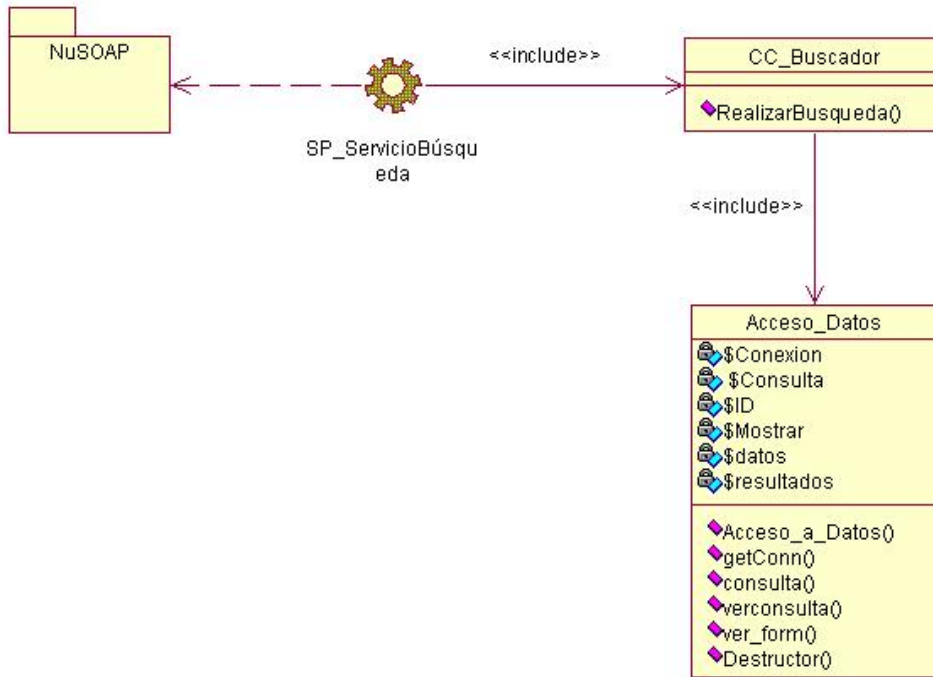


Figura 3.10 Diagrama de Clases del Diseño BuscadorSE.

Diagrama de secuencia del caso de uso “BuscadorSE”. [Ver Anexo16].

### 3.3 Diseño de la Base de Datos

Para el diseño de la base de datos del sistema se utilizó el diagrama de clases persistentes y el modelo de datos, representando las clases que simbolizan los datos que se obtienen y almacenan durante los procesos de la aplicación, siendo estos los que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará el modelamiento de la lógica del negocio.

### 3.3.1 Diagrama de Clases Persistentes.

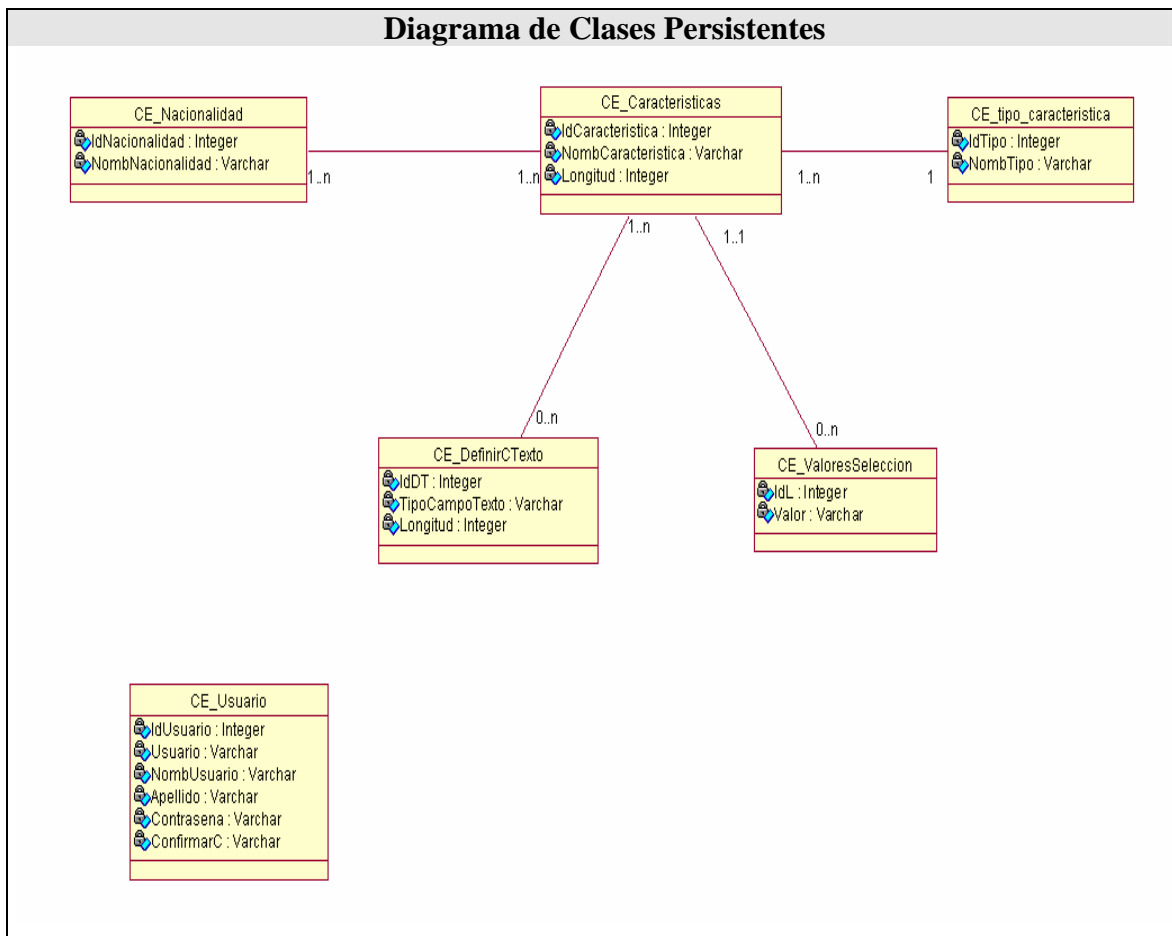


Figura 3.11 Diagrama de Clases Persistentes.



3.4.2 Modelo de Datos.

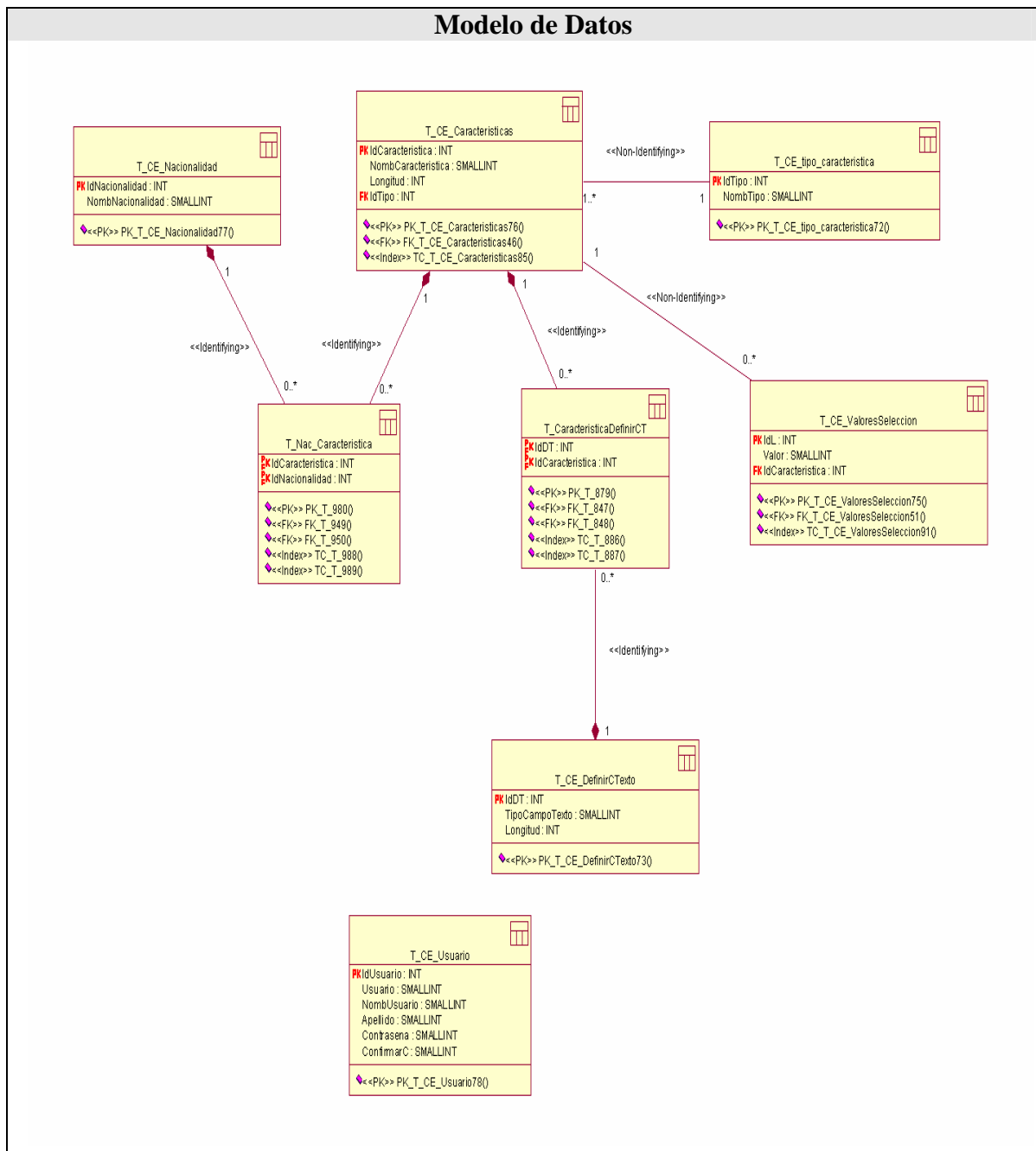


Figura 3.12 Modelo de Datos.

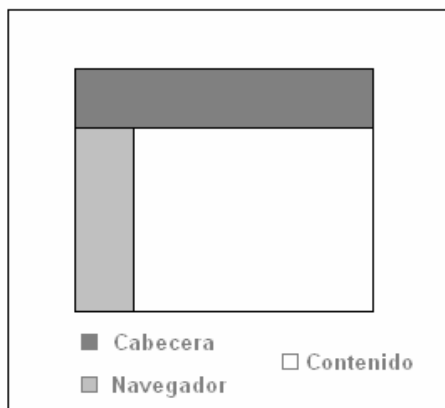
### 3.4 Principios de diseño

El diseño, sea cual sea el objeto del mismo, tiene que basarse en el usuario, pues generalmente muchos de ellos no presentan una preparación en las cuestiones de la informática. Para ello, cada sistema debe utilizar ciertos principios generales que garanticen la usabilidad en los diseños para aplicaciones Web.

#### 3.4.1 Estándares en la interfaz de la aplicación

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

Es por ello que para lograr un diseño consistente de la interfaz de la aplicación se utiliza en todas las páginas el esquema Cabecera-Navegador-Contenido. La cabecera contiene el nombre de la aplicación en el centro superior y muestra una imagen relacionada con el medio para el que se realiza el sistema. En el navegador se incluyen los enlaces a las distintas secciones. En el área del contenido se muestran los formularios de entrada, los reportes, etc.



**Figura 3.13 Esquema de página.**

Se utilizan para el diseño las tablas y plantillas, dado que son 100% compatibles con todos los navegadores, también se emplea una hoja de estilos para guardar la configuración del diseño de todas las

páginas. Esta hoja de estilos establece el tipo, tamaño de fuente de los distintos elementos de cada página, el color de los vínculos, el color de fondo, el formato de los controles de formulario y las tablas, entre otros. En la aplicación se utilizó el color gris porque es un color neutro y pasivo, sugiere seriedad y está muy asociado a las aplicaciones informáticas, ya que la mayoría de las interfaces gráficas son de color gris o lo contienen.

### **3.4.2 Formato de los reportes**

Los reportes se muestran en formato de tabla, con los registros ordenados siguiendo varios criterios, según sea conveniente en cada caso de uso. Dichas tablas muestran solo la información básica de cada elemento del reporte. En el caso de los reportes de los paquetes Gestionar Ciudadanía, Gestionar Ciudadano y Administrar Usuario se brinda la posibilidad de seleccionar un elemento del listado para modificarlo o eliminarlo.

[Ver Anexo 17].

### **3.4.3 Tratamiento de excepciones**

Para reducir la ejecución de errores es utilizado javascript evitando así que el servidor Web procese la página en vano. Este es el caso de los formularios de inserción/actualización, y las eliminaciones. También, se utilizan errores en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar. Y se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

[Ver Anexo 18].

### **3.4.4 Estándares de codificación**

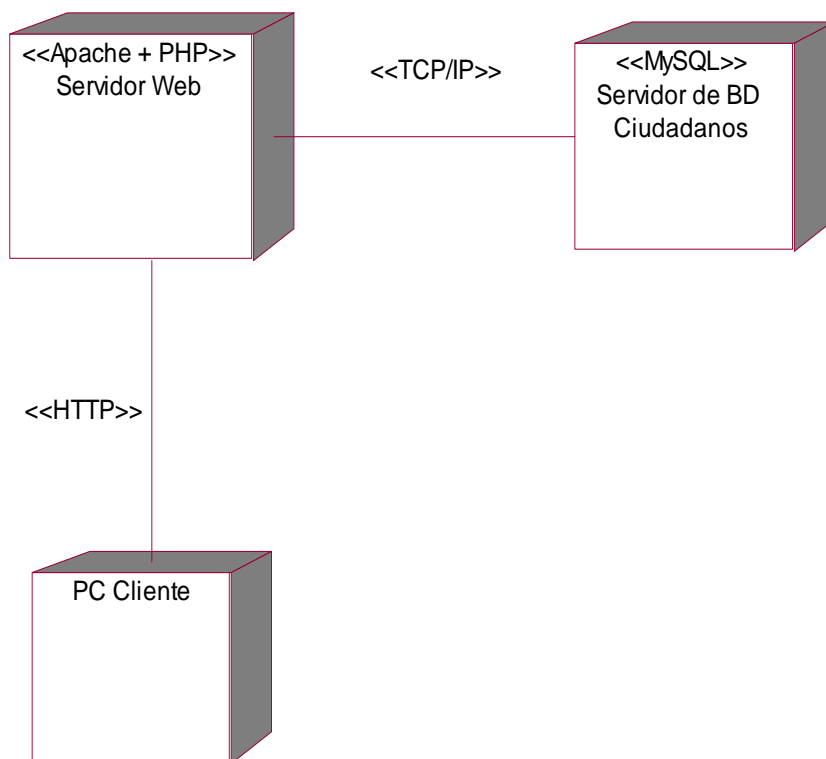
Para lograr un mejor entendimiento del código en la implementación de la aplicación es necesario establecer un estándar de codificación a usar. En la política seguida al respecto todas las variables y nombres de funciones a utilizar se definieron en idioma español. En el caso de los objetos que se utilizan como por ejemplo los *campo de texto* en su nombre incluyen el nombre asociado al valor que va contenido. Los inicios ({} y cierre (}) de ámbito se encuentran alineados debajo de la declaración a la que

pertenecen. Se usa una línea propia para {. Los signos lógicos y de operación se separan por un espacio antes y después de los mismos. Los nombres de las variables utilizadas comienzan en minúscula y son cortos, claros y describen su propósito. Los comentarios se definen comenzando con los caracteres */\** y terminando con *\*/* para los comentarios de varias líneas, y comenzando con los caracteres *//* para los de una sola línea.

### 3.5 Modelo de despliegue

El modelo de despliegue describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware.

A continuación se muestra el diagrama de despliegue de la aplicación.



**Figura 3.17 Diagrama de Despliegue.**

### 3.6 Descripción preliminar del modelo de pruebas

Las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

Con el objetivo de encontrar el mayor número posible de errores, las pruebas deben conducirse por casos de pruebas que deben diseñarse utilizando técnicas definidas. Entre los casos de prueba se puede distinguir dos tipos comúnmente utilizados: las llamadas pruebas de “caja negra” y las de “caja blanca”.

Una prueba de “caja blanca” comprueba la lógica interna del programa, mientras que una prueba de “caja negra” es una prueba del comportamiento observable externamente del sistema.

En el caso del Sistema de Registro para Ciudadanos, se realizan pruebas de “caja negra” para los caso de uso más significativos, para así probar la interacción entre el usuario y el sistema, que se satisfagan las precondiciones y poscondiciones, y que se siga la secuencia de acciones intermedias especificadas por el caso de uso.

#### 3.6.1 Casos de prueba

##### Autenticar Usuario.

<b>Caso de uso: Autenticar Usuario</b>
<b>Caso de prueba:</b> Autenticar usuario, verificando que el usuario y la contraseña sean correctos.
<b>Entrada:</b>
<ul style="list-style-type: none"> <li>El usuario entra los datos de usuario y contraseña.</li> </ul>
<b>Resultado:</b>
<ul style="list-style-type: none"> <li>El sistema permite la entrada del usuario enviándolo a la página principal de la aplicación.</li> </ul>
<b>Condiciones:</b>
<ul style="list-style-type: none"> <li>El usuario tiene acceso al Sistema.</li> </ul>

<b>Caso de uso: Autenticar Usuario</b>
<b>Caso de prueba:</b> Autenticar usuario, verificando que el usuario y la contraseña sean correctos.
<b>Entrada:</b>
<ul style="list-style-type: none"> <li>El usuario entra los datos de usuario y contraseña.</li> </ul>

<b>Resultado:</b> <ul style="list-style-type: none"><li>• El sistema muestra un mensaje donde se especifica que el usuario no se encuentra registrado en la Base de Datos.</li></ul>
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• El Sistema no permite la entrada del usuario.</li></ul>

<b>Caso de uso: Autenticar Usuario</b>
<b>Caso de prueba:</b> Autenticar usuario, verificando que el usuario y la contraseña sean correctos.
<b>Entrada:</b> <ul style="list-style-type: none"><li>• El usuario entra los datos de usuario y contraseña.</li></ul>
<b>Resultado:</b> <ul style="list-style-type: none"><li>• El sistema muestra un mensaje donde se especifica que la clave no es la correcta</li></ul>
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• El usuario no tiene acceso al sistema.</li></ul>

#### Administrar Usuario.

<b>Caso de uso: Administrar Usuario (Registrar).</b>
<b>Caso de prueba:</b> Registrar los datos del usuario que tendrá acceso a la manipulación del sistema.
<b>Entrada:</b> <ul style="list-style-type: none"><li>• El Administrador entra los datos del usuario llenando los campos que tiene la aplicación (Nombre, Apellido, Usuario, Contraseña) y oprime un botón para la entrada de los datos en la Base de Datos.</li></ul>
<b>Resultado:</b> <ul style="list-style-type: none"><li>• Se muestra un mensaje al Administrador que le indica que el usuario fue registrado correctamente.</li></ul>
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• Le permite al Administrador que vuelva a repetir la acción de registrar a otros usuarios que no hayan sido registrados.</li></ul>

<b>Caso de uso: Administrar Usuario (Registrar).</b>
<b>Caso de prueba:</b> Registrar los datos del usuario que tendrá acceso a la manipulación del sistema.
<b>Entrada:</b> <ul style="list-style-type: none"><li>• El Administrador entra los datos del usuario llenando los campos que tiene la aplicación (Nombre, Apellido, Usuario, Contraseña) y oprime un botón para la entrada de los datos en la base de datos.</li></ul>
<b>Resultado:</b> <ul style="list-style-type: none"><li>• El sistema muestra un mensaje al Administrador donde se especifica que el usuario</li></ul>

se encuentra registrado en la Base de Datos.
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• Le permite al Administrador que vuelva a repetir la acción de registrar al usuario en el sistema con otro nombre de usuario.</li></ul>
<b>Caso de uso: Administrar Usuario (Eliminar).</b>
<b>Caso de prueba:</b> El Administrador elimina los datos de los usuarios que sean necesarios en el sistema.
<b>Entrada:</b> <ul style="list-style-type: none"><li>• El Administrador marca una casilla de selección que indica el nombre del usuario que se eliminará, presionando posteriormente el botón que indica la acción de eliminar mostrándose un mensaje de confirmación.</li></ul>
<b>Resultado:</b> <ul style="list-style-type: none"><li>• El sistema elimina correctamente el usuario registrado en la Base de Datos.</li></ul>
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• Le permite al Administrador que vuelva a repetir la acción de eliminar los datos de cualquier otro usuario.</li></ul>

### Gestionar Ciudadanía.

<b>Caso de uso: Gestionar Ciudadanía (Configurar).</b>
<b>Caso de prueba:</b> Configurar los datos de la Ciudadanía que será registrada.
<b>Entrada:</b> <ul style="list-style-type: none"><li>• El Administrador entra el nombre de la Ciudadanía y selecciona las características correspondientes a la misma oprimiendo el botón para la entrada de los datos en la base de datos.</li></ul>
<b>Resultado:</b> <ul style="list-style-type: none"><li>• El sistema registra la Ciudadanía correctamente en la Base de Datos.</li></ul>
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• Le permite al Administrador que vuelva a repetir la acción de configurar otras Ciudadanías.</li></ul>

<b>Caso de uso: Gestionar Ciudadanía (Configurar).</b>
<b>Caso de prueba:</b> Configurar los datos de la Ciudadanía que será registrada.
<b>Entrada:</b> <ul style="list-style-type: none"><li>• El Administrador entra el nombre de la Ciudadanía y selecciona las características correspondientes a la misma oprimiendo el botón para la entrada de los datos en la base de datos.</li></ul>

<b>Resultado:</b> <ul style="list-style-type: none"><li>• El sistema muestra un mensaje al Administrador donde le informa que esa Ciudadanía ya está configurada.</li></ul>
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• Le permite al Administrador que vuelva a repetir la acción de configurar otras Ciudadanías.</li></ul>

<b>Caso de uso: Gestionar Ciudadanía (Adicionar Característica).</b>
<b>Caso de prueba:</b> Adicionar Características para la configuración de la Ciudadanía que será registrada.
<b>Entrada:</b> <ul style="list-style-type: none"><li>•El Administrador entra el nombre de la Característica y selecciona el tipo de dato correspondiente a la misma oprimiendo el botón enviar para registrar los datos en la Base de Datos.</li></ul>
<b>Resultado:</b> <ul style="list-style-type: none"><li>• El sistema registra la Característica correctamente en la Base de Datos.</li></ul>
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• Le permite al Administrador que vuelva a repetir la acción de registrar otras Características.</li></ul>

<b>Caso de uso: Gestionar Ciudadanía (Adicionar Característica).</b>
<b>Caso de prueba:</b> Adicionar Características para la configuración de la Ciudadanía que será registrada.
<b>Entrada:</b> <ul style="list-style-type: none"><li>•El Administrador entra el nombre de la Característica y selecciona el tipo de dato correspondiente a la misma oprimiendo el botón enviar para registrar los datos en la Base de Datos.</li></ul>
<b>Resultado:</b> <ul style="list-style-type: none"><li>• El sistema muestra un mensaje al Administrador donde le informa que esa Característica ya existe.</li></ul>
<b>Condiciones:</b> <ul style="list-style-type: none"><li>• Le permite al Administrador que vuelva a repetir la acción de registrar otras Características.</li></ul>

**Gestionar datos Ciudadanos.**

<b>Caso de uso: Gestionar datos Ciudadanos (Registrar).</b>
<b>Caso de prueba:</b> Registrar los datos de los ciudadanos que realizaran el Examen Teórico de Conducción. .



**Entrada:**

- El Administrador entra los datos personales de los ciudadanos que serán registrados oprimiendo el botón para la entrada de los datos en la base de datos.

**Resultado:**

- El sistema registra al Ciudadano correctamente en la Base de Datos.

**Condiciones:**

- Le permite al Administrador que vuelva a repetir la acción de registrar a otros ciudadanos.

### Conclusiones

En este capítulo se modeló detalladamente la solución propuesta, a través del modelo de diseño y de implementación representado por el diagrama de despliegue. Se seleccionaron los patrones de diseño y arquitectura a utilizar en la construcción de la aplicación, se analizaron los principios de diseño de la interfaz de la misma, los estándares de codificación a emplear durante la fase de construcción y se realizaron los casos de pruebas a los casos de uso más significativos, para comprobar que cumpla los requerimientos que le dieron origen.

### CONCLUSIONES GENERALES

A partir de la investigación realizada para la elaboración de este sistema utilizando el Proceso Unificado de Desarrollo de Software (RUP) como metodología idónea para lograr una mejor comprensión de los requisitos de la aplicación y formalización de los mismos, se arriba a las siguientes conclusiones:

- A través del estudio que se llevó a cabo, se detectó un aspecto que incidía en el registro de ciudadanos para obtener la licencia de conducción mediante el Evaluador Teórico CISMA cuando el producto era solicitado por diferentes nacionalidades.
- Como respuesta al problema originado, se llegó a la propuesta de realizar un sistema para dar soporte al proceso de registro de forma genérica, de la información de los ciudadanos de diferentes nacionalidades que serán evaluados por el Evaluador Teórico CISMA.
- El sistema desarrollado incluye, a parte de las funcionalidades ya existentes, la posibilidad de incorporar ciudadanos de cualquier nacionalidad de forma transparente.
- La utilización de esta aplicación contribuye al fortalecimiento del proceso de registro del Evaluador Teórico CISMA y su puesta en práctica significará una mejora considerable, pues ante el pedido de clientes de diversos territorios no será sometido a reiteradas actualizaciones, lo que maximizará su comercialización

Con la propuesta y el estudio realizado se materializan los objetivos planteados al inicio de esta investigación: Desarrollar una herramienta para almacenar de forma genérica la información de los ciudadanos de diferentes nacionalidades que serán evaluados por el Evaluador Teórico CISMA.

## **RECOMENDACIONES**

Se recomienda:

- Continuar el estudio con el objetivo de encontrar nuevas funcionalidades para refinar e implementar una herramienta más completa y general.
- La implementación del Servicio Web diseñado para la realización de búsqueda por otras aplicaciones.
- La incorporación de otros componentes al formulario dinámico que se genera cuando se va a realizar el registro de los ciudadanos.
- Realizar un estudio de cómo automatizar las vistas que se generan manualmente para facilitar el proceso de búsqueda y reportes del sistema.
- Realizar la implementación de acople de este sistema con el Evaluador Teórico CISMA.

### REFERENCIAS BIBLIOGRAFICAS

1. ADDISON WESLEY ED. JAMES RUMBAUGH, I. J. Y. G. B. El Lenguaje Unificado de Modelado. Manual de Referencia. , Rational Software Corporation., 2000.
2. ALVAREZ, M. A. Características y ventajas de las CSS, 2006a. [2006]. Disponible en: <http://www.desarrolloweb.com/articulos/182.php>
3. ALVAREZ, S. Tipos de Programación, 2006b. [2006]. Disponible en: <http://www.desarrolloweb.com/articulos>
3. ARENAS, M. I. G. Lenguajes de Marcas, 2002. [2006]. Disponible en: <http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml#11>
4. BLANCO, R. G.; S. P. TOBALINA, et al. Introducción a Rational Rose. , 2002. [2006]. Disponible en: <http://ateam.lsi.upc.es/~es-e/web/documents/lab/0304Q2/lessons/lese-2/LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt?PHPSESSID=0ade5aba4d6b566cb794d3b7def7cd08>.
5. EDUARDO. ¿Qué es MySQL? , 2002. [2006]. Disponible en: <http://www.mysql-hispano.org/page.php?id=2>
6. FOUNDATION, T. A. S. The Number One HTTP Server On The Internet 2005. [2006]. Disponible en: <http://httpd.apache.org/>
7. Guía Breve de CSS 2006. [2006]. Disponible en: <http://www.w3c.es/Divulgacion/Guiasbreves/HojasEstilo>
8. INTERNET, A. E. D. Manejo de la plataforma Web: Linux, Apache, MySQL y PHP., 2006. [2006]. Disponible en: [http://www.ciberaula.com/curso/lamp/que\\_es/](http://www.ciberaula.com/curso/lamp/que_es/)

9. LAFUENTE, G. J. UML (Unified Modeling Lenguaje). 2001 [2006]. Disponible en: <http://gidis.ing.unlpam.edu.ar/personas/glafuente/uml/uml.html>
10. MAURO, W. M. D. E. C. Manual basico de ASP (I), 2006. [2006]. Disponible en: [http://www.terra.es/personal6/morenocerro2/asp/manual/asp\\_1.html](http://www.terra.es/personal6/morenocerro2/asp/manual/asp_1.html)
11. PAREDES, A. P. R. Construyendo software de alta calidad 2005. [2006]. Disponible en: [http://www.elguille.info/colabora/NET2005/Percynet\\_ConstruyendoSoftCalidad.htm](http://www.elguille.info/colabora/NET2005/Percynet_ConstruyendoSoftCalidad.htm)
12. ¿Que es ASP ? 2002. [2006]. Disponible en: <http://www.zonagratis.com/microsoft/asp/index.html>
13. SOLDADO, R. M. Herramientas utilizadas en el proceso de Diseño. Hasta donde llega su alcance., 2003. [2006]. Disponible en: <http://lsi.ugr.es/~rosana/investigacion/rational.ps>.
14. VALLE, J. G. Definición arquitectura cliente servidor, 2005. [2006]. Disponible en: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>
15. VÁZQUEZ, J. A. G. Desarrollo Web con PHP y MySQL., Ediciones ANAYA Multimedias (Grupo ANAYA) S.A, 2003.
16. WIKIMEDIA FOUNDATION, I. HTML, 2006a. [2006]. Disponible en: <http://es.wikipedia.org/wiki/HTML>  
---. JavaScript, 2006b. [2006]. Disponible en: <http://es.wikipedia.org/wiki/JavaScript>  
---. Microsoft SQL Server, 2006c. [2006]. Disponible en: [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)  
---. Proceso Unificado, 2006d. [2006]. Disponible en: [http://es.wikipedia.org/wiki/Proceso\\_Unificado](http://es.wikipedia.org/wiki/Proceso_Unificado)  
---. Servicio Web, 2006e. [2006]. Disponible en: [http://es.wikipedia.org/wiki/Servicio\\_Web](http://es.wikipedia.org/wiki/Servicio_Web)  
---. SQL, 2006f. [2006]. Disponible en: <http://es.wikipedia.org/wiki/SQL>
17. Breve historia de PHP [Consultado el: 21/10/2006 Disponible en: <http://www.desarrolloweb.com>.
18. HANSEN, G. W. y V.HANSEN, J. Diseño y Administración de Bases de Datos. 2007, vol. 2,

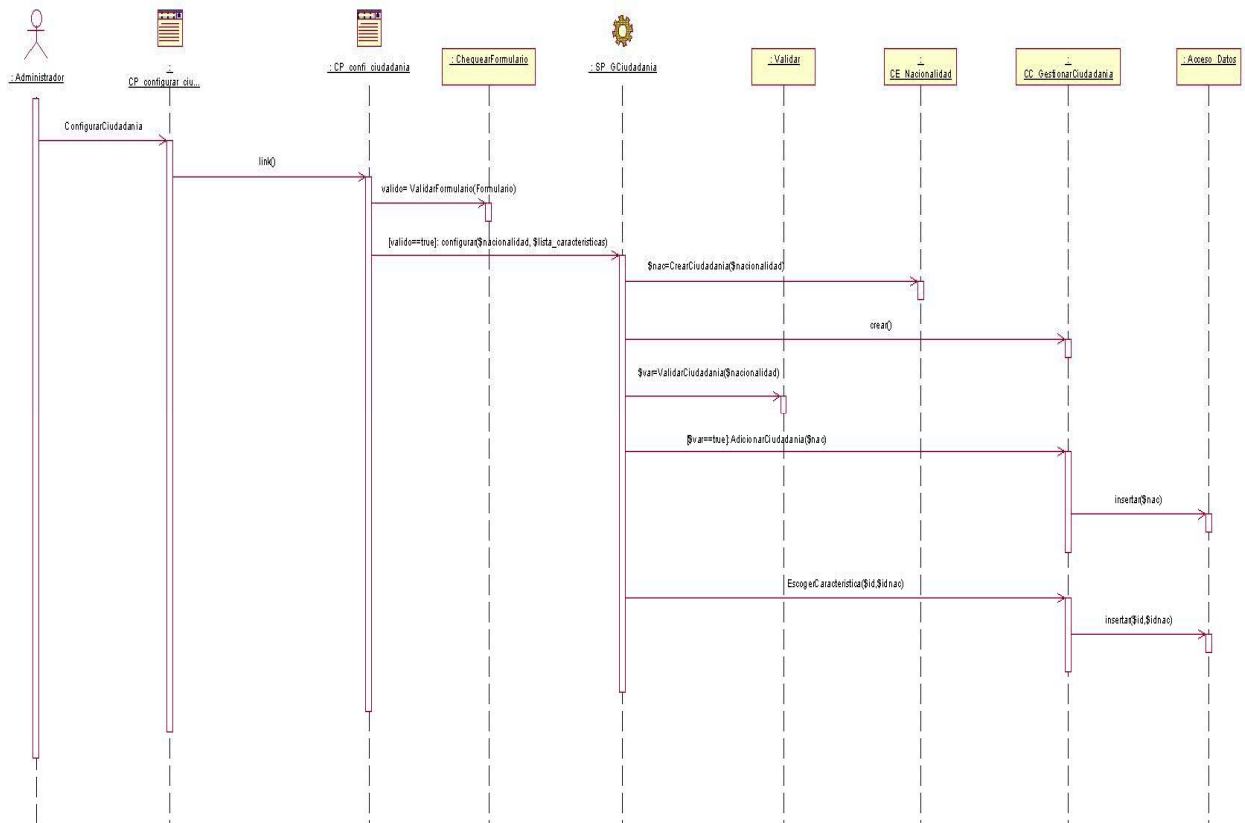
19. LARMAN, C. UML y patrones. vol. Tomo I,
20. Qué es AJAX Disponible en: <http://www.webtaller.com/maletin/articulos/que-es-ajax.php>.
21. RICARDO, F. Á. C. [Consultado el: 30/5/2007 Disponible en:<http://www.monografias.com/trabajos43/patron-modelo-vista>.
22. Servicios Web con PHP Argentina: [Consultado el: 25/1/2007 Disponible en: [www.solar.org.ar](http://www.solar.org.ar).
23. Sistemas\_Gestores\_de\_Bases\_de\_Datos [Consultado el: 21/10/2006 Disponible en: [http://es.wikipedia.org/wiki/Sistemas\\_Gestores\\_de\\_Bases\\_de\\_Datos](http://es.wikipedia.org/wiki/Sistemas_Gestores_de_Bases_de_Datos).

## BIBLIOGRAFÍA

1. *Breve Historia de PHP* Disponible en: <http://www.desarrolloweb.com/articulos>. (21/10/2006)
2. CONVERSE, T.; PARCK, J., *et al. PHP 5 y MySQL*. (30/11/2006)
3. GILFILLAN, I. *La Biblia MySQL*. Disponible en: <http://bibliodoc.uci.cu/pdf/reg02847.pdf>. (30/11/2006)
4. LARMAN, C. *UML y Patrones*. vol. Tomo I, (30/1/2007)
5. MARTÍNEZ, R. *Manual de PHP* Disponible en: <http://brak.unsl.edu.ar/php>. (22/10/2006)
6. RICARDO, F. Á. C.  
  
Utilización del Patrón Modelo – Vista – Controlador (MVC) nº Disponible en:  
<http://www.monografias.com/trabajos43/patron-modelo-vista>. (30/5/2007)
7. WELLING, L. y THOMSON, L. *Desarrollo Web con PHP y MySQL*. Disponible en:  
<http://bibliodoc.uci.cu/pdf/reg02819.pdf>. (5/11/2006)

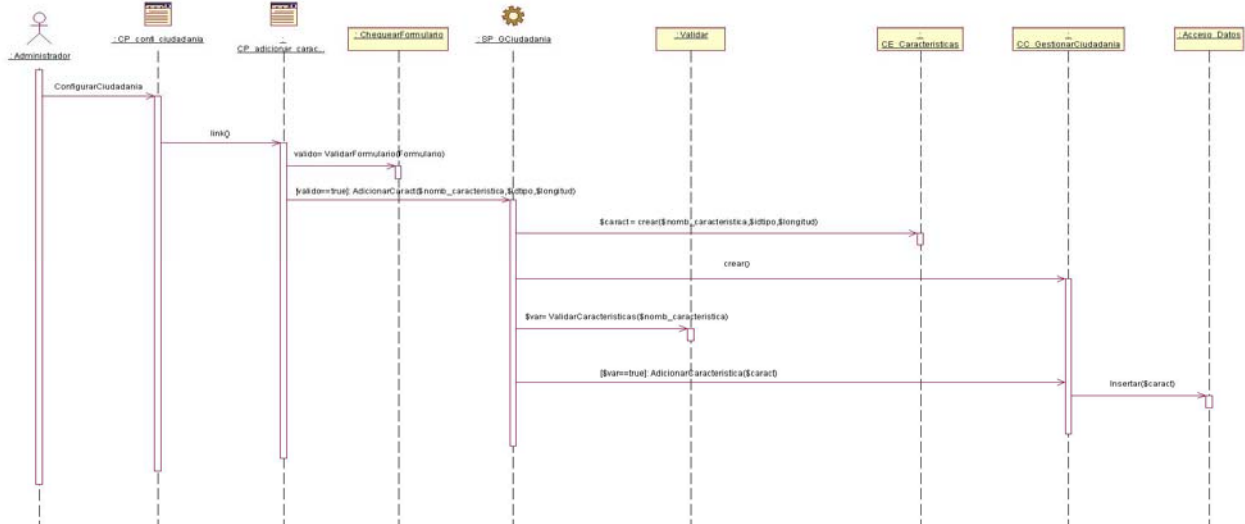
ANEXOS

Anexo 1. Diagrama de secuencia del caso de uso “Gestionar Ciudadanía”. Sección “Configurar Ciudadanía”.

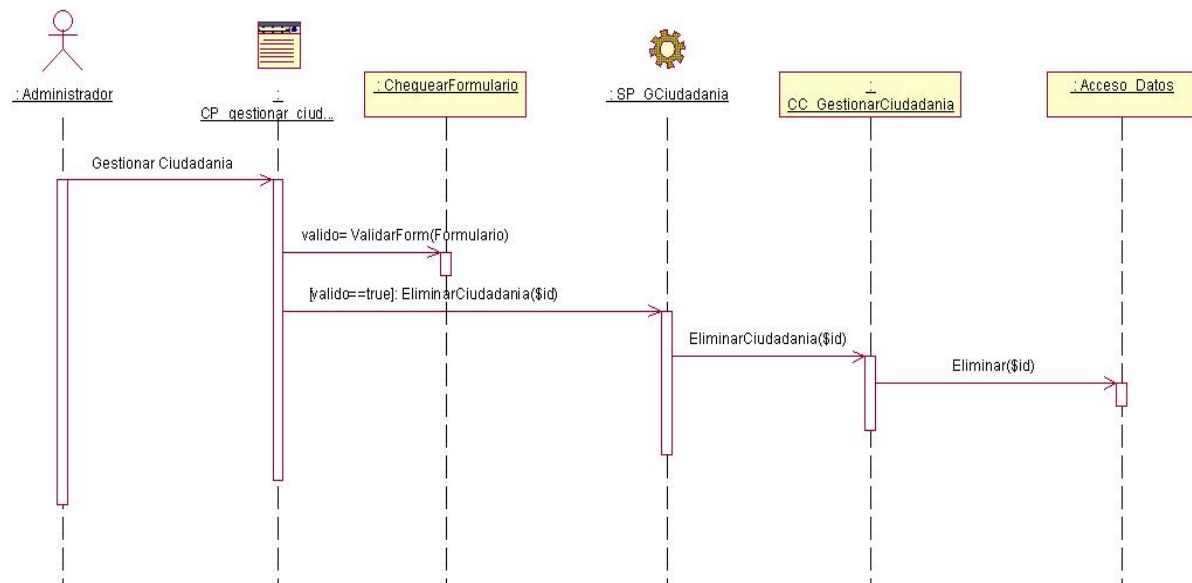




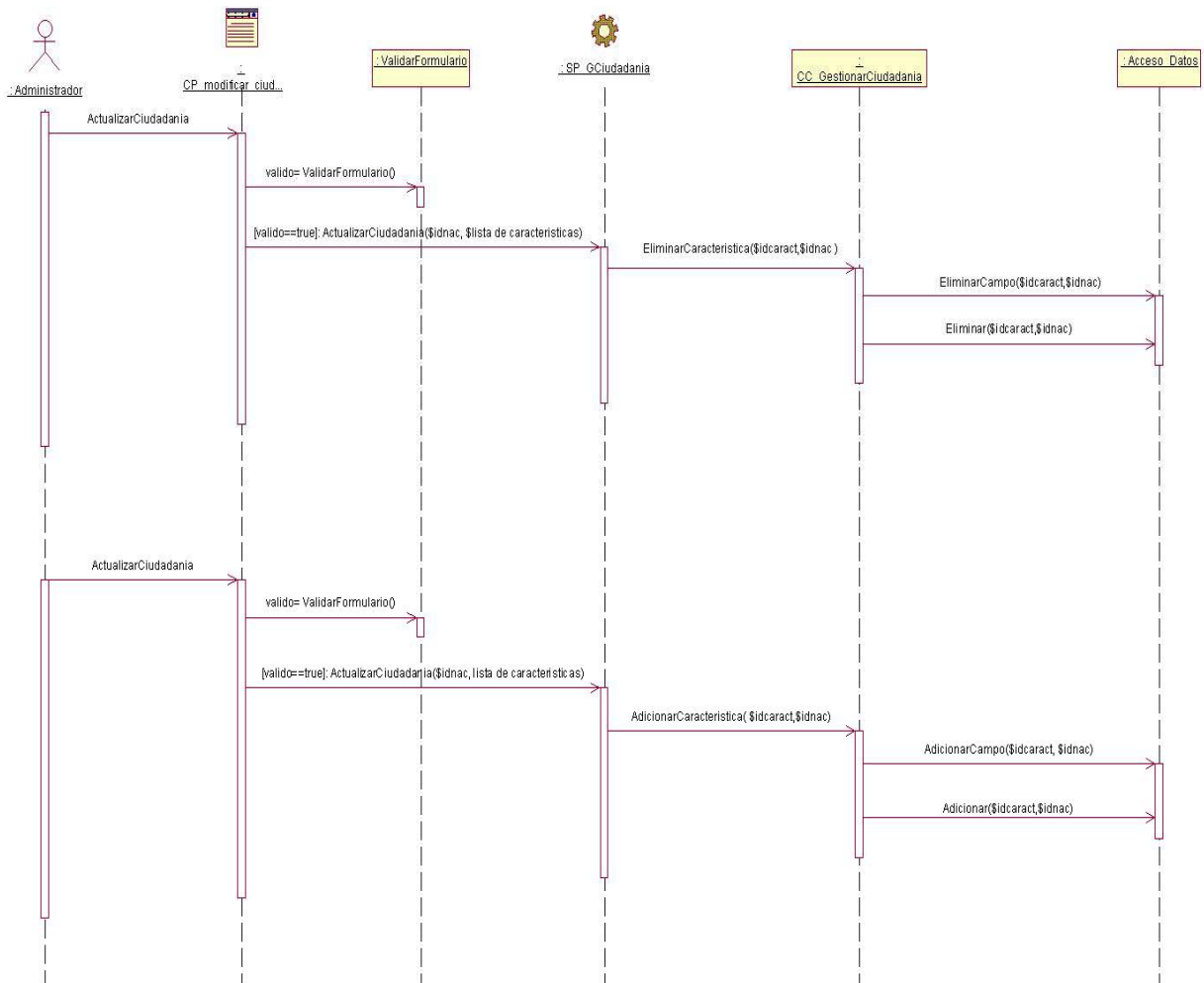
**Anexo 2. Diagrama de secuencia del caso de uso “Gestionar Ciudadanía”. Sección “Adicionar Característica”.**



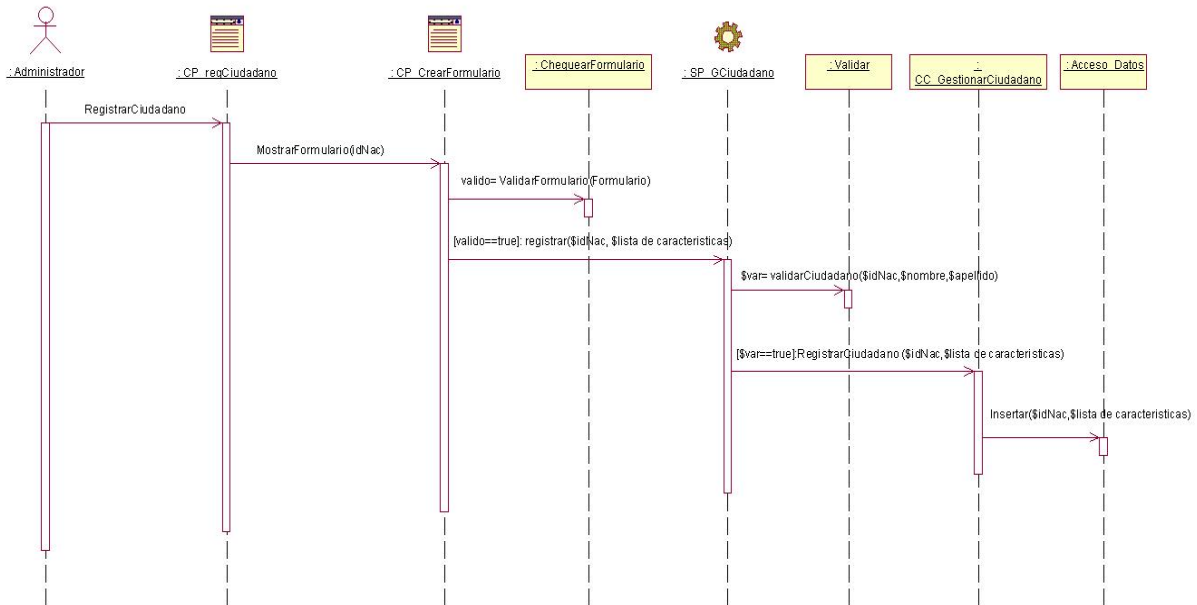
**Anexo 3. Diagrama de secuencia del caso de uso “Gestionar Ciudadanía”. Sección “Eliminar Ciudadanía”.**



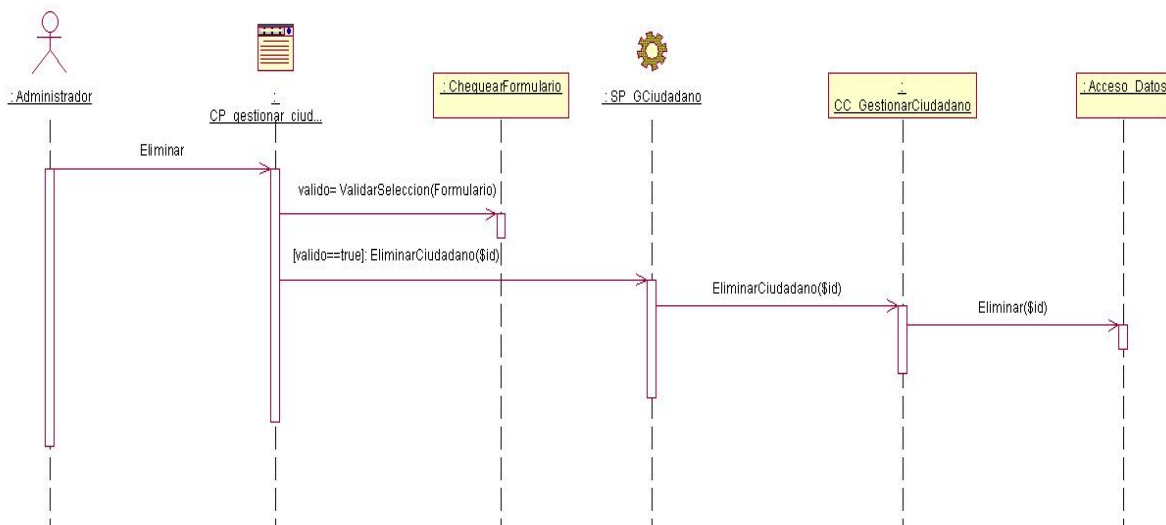
Anexo 4. Diagrama de secuencia del caso de uso “Gestionar Ciudadanía”. Sección “Modificar Ciudadanía”.



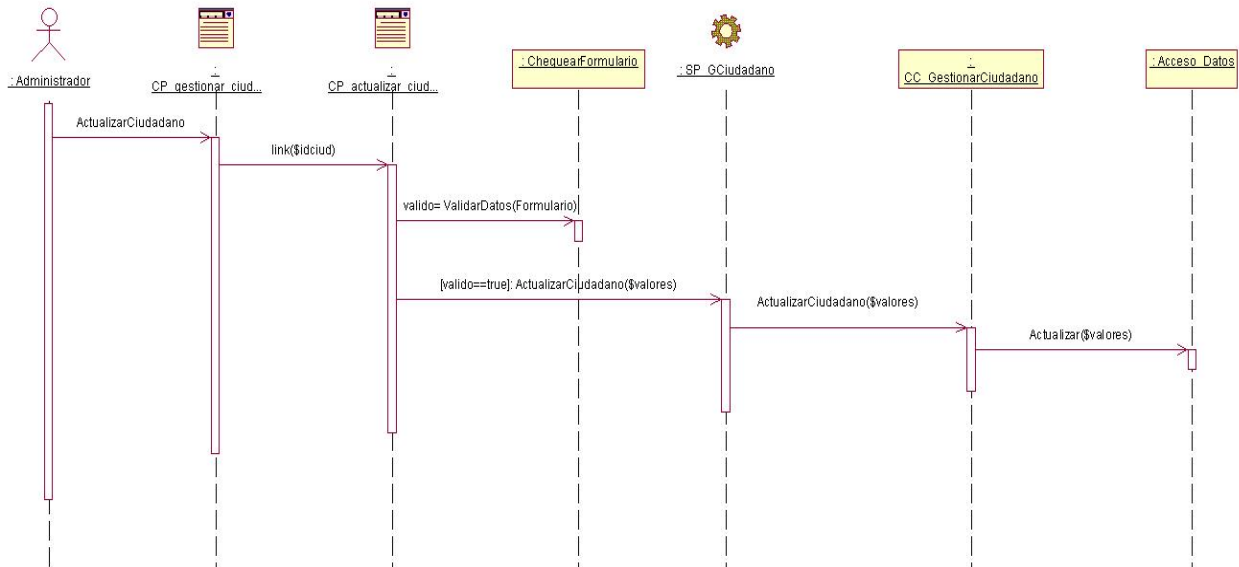
**Anexo 5. Diagrama de secuencia del caso de uso “Gestionar Ciudadano”. Sección “Registrar Ciudadano”.**



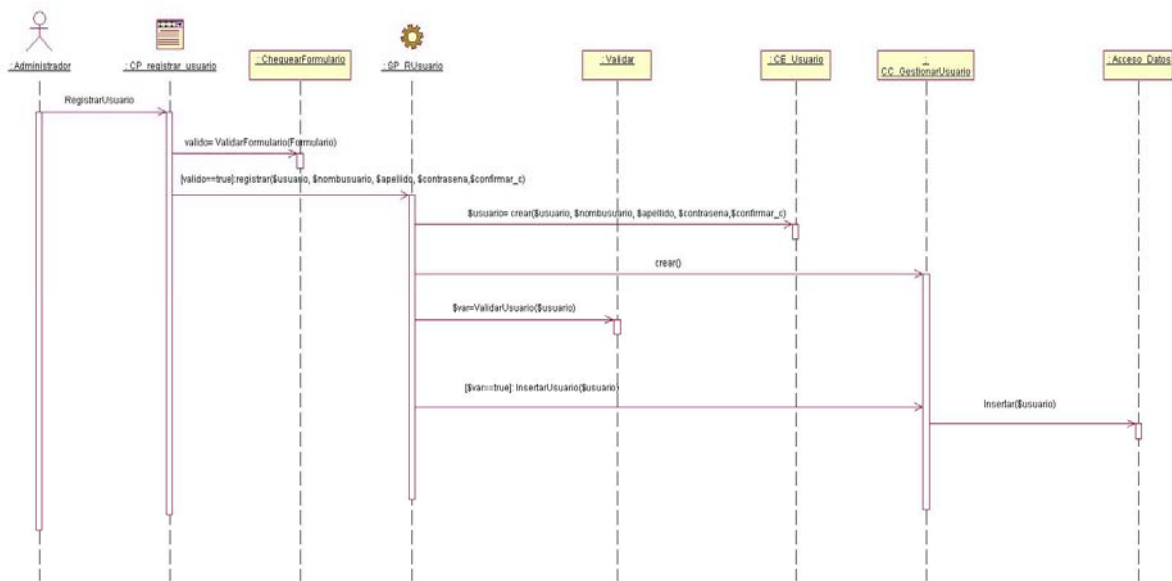
**Anexo 6. Diagrama de secuencia del caso de uso “Gestionar Ciudadano”. Sección “Eliminar Ciudadano”.**



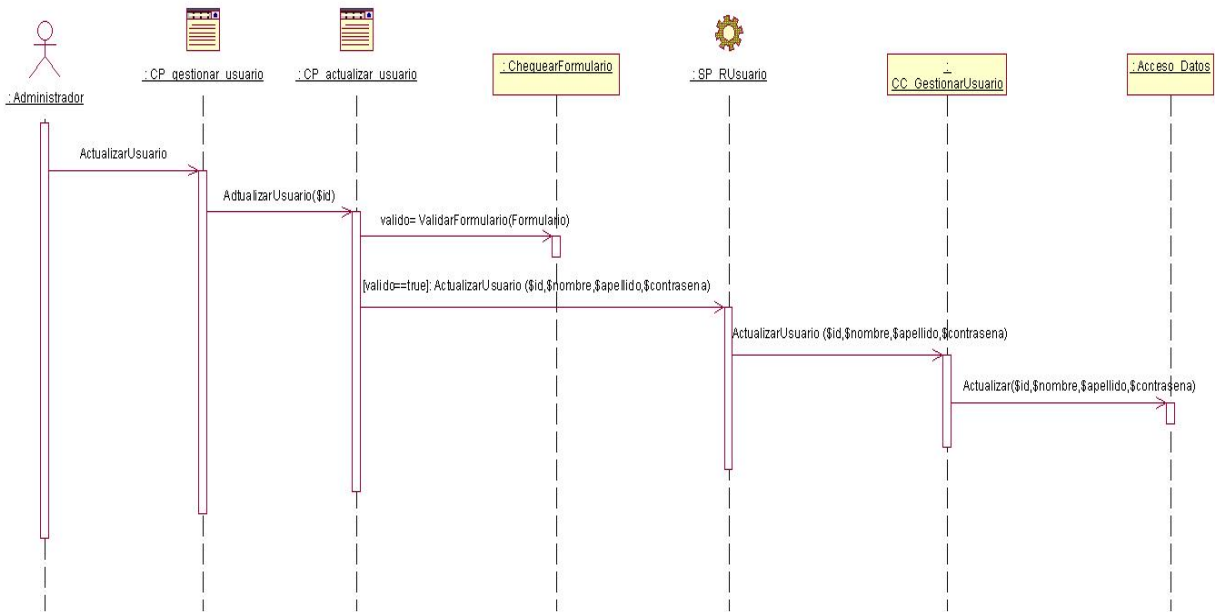
**Anexo 7. Diagrama de secuencia del caso de uso “Gestionar Ciudadano”. Sección “Modificar Ciudadano”.**



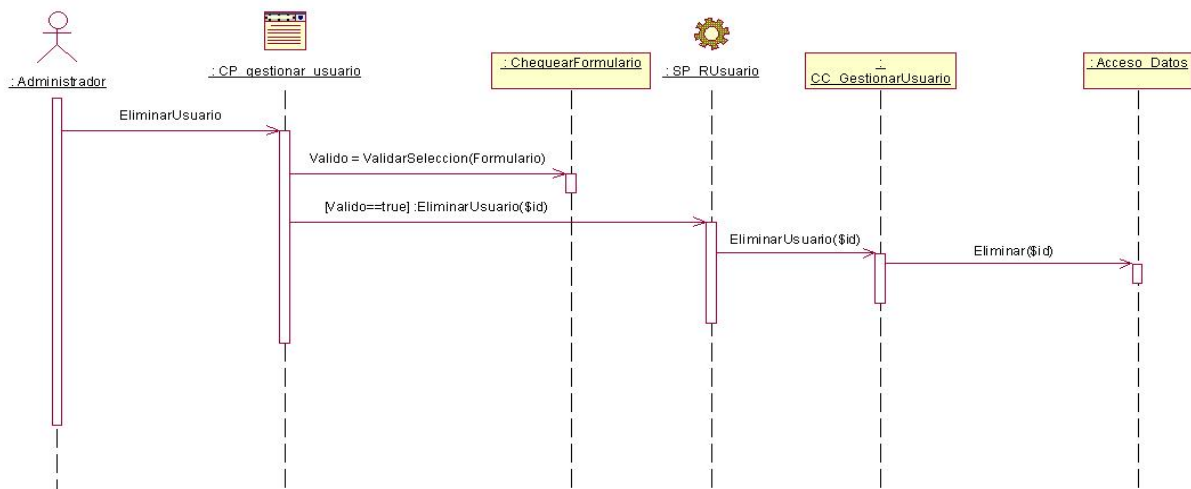
**Anexo 8. Diagrama de secuencia del caso de uso “Administrar Usuario”. Sección “Registrar Usuario”.**



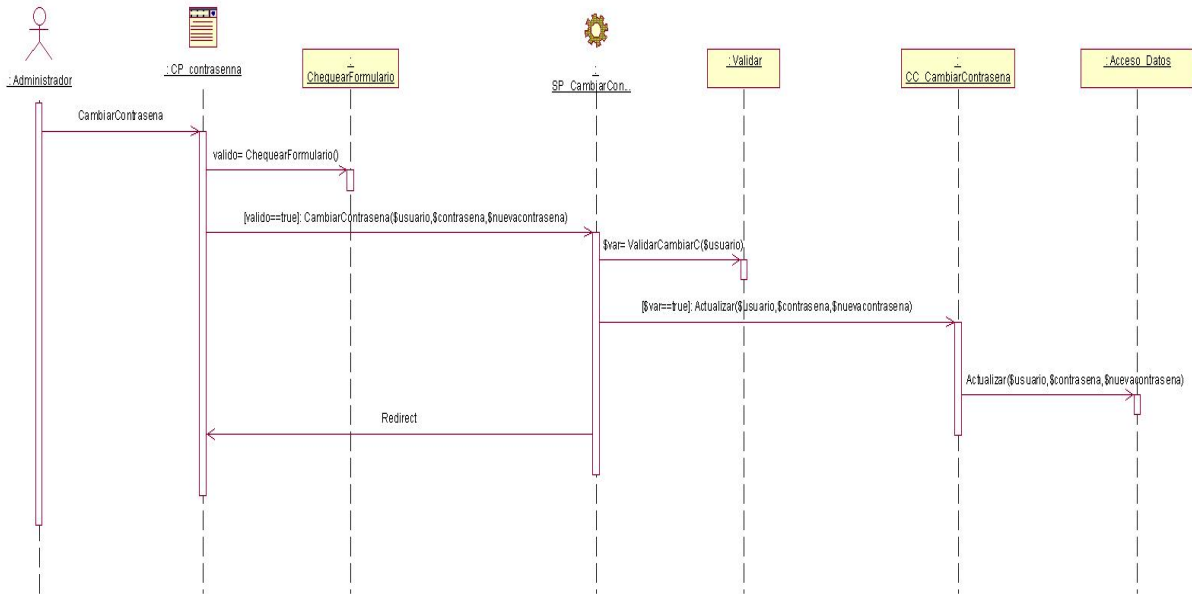
Anexo 9. Diagrama de secuencia del caso de uso “Administrar Usuario”. Sección “Modificar Usuario”.



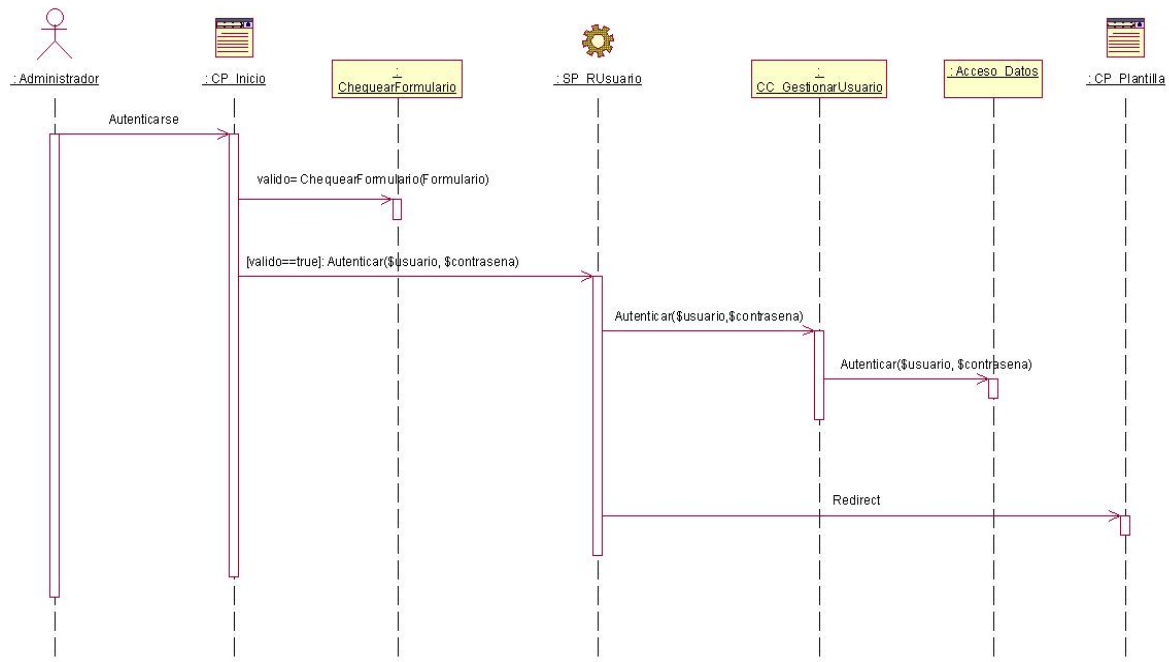
Anexo 10. Diagrama de secuencia del caso de uso “Administrar Usuario”. Sección “Eliminar Usuario”.



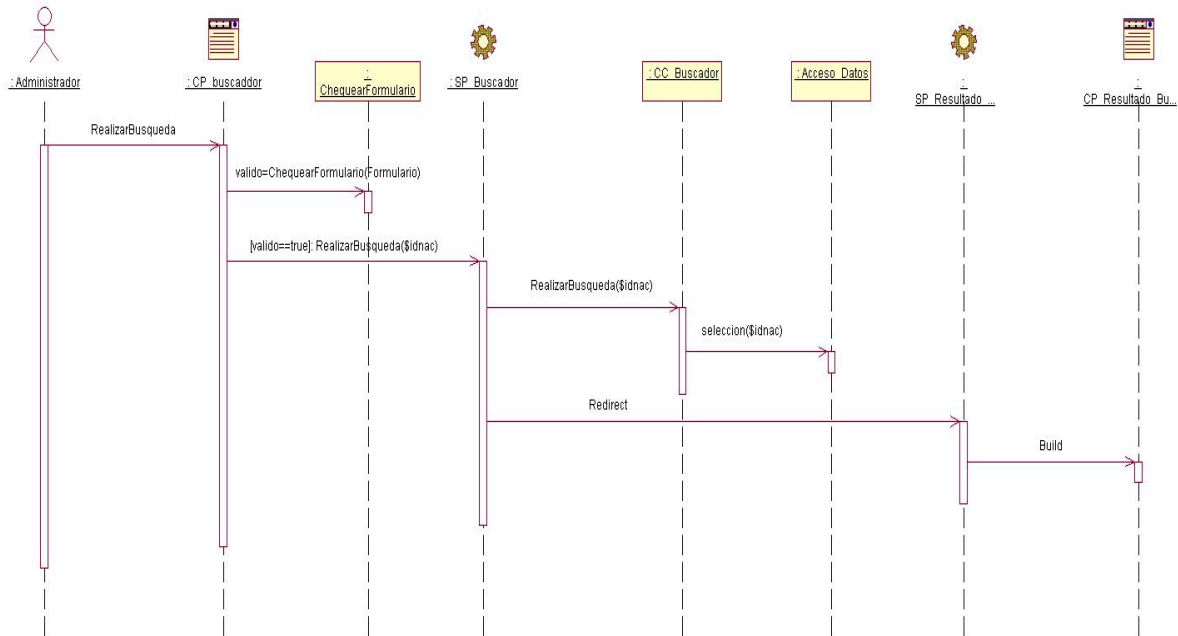
Anexo 11. Diagrama de secuencia del caso de uso “Cambiar Contraseña”.



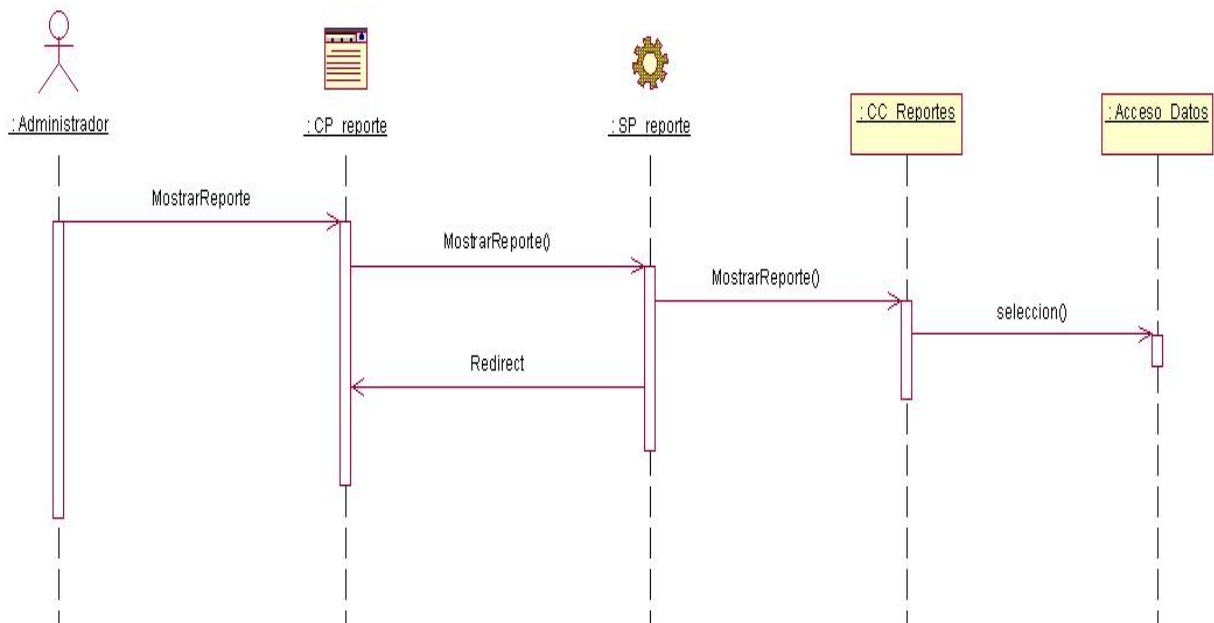
[Anexo12]. Diagrama de secuencia del caso de uso “Autenticarse”.



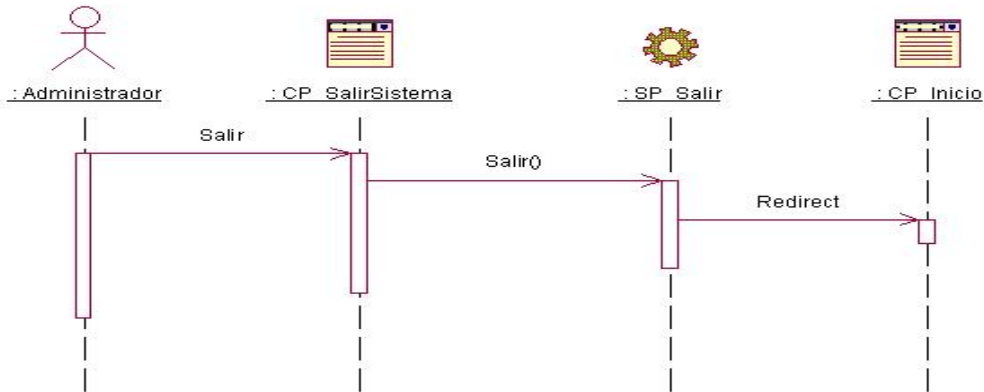
[Anexo13]. Diagrama de secuencia del caso de uso “Realizar Búsqueda”.



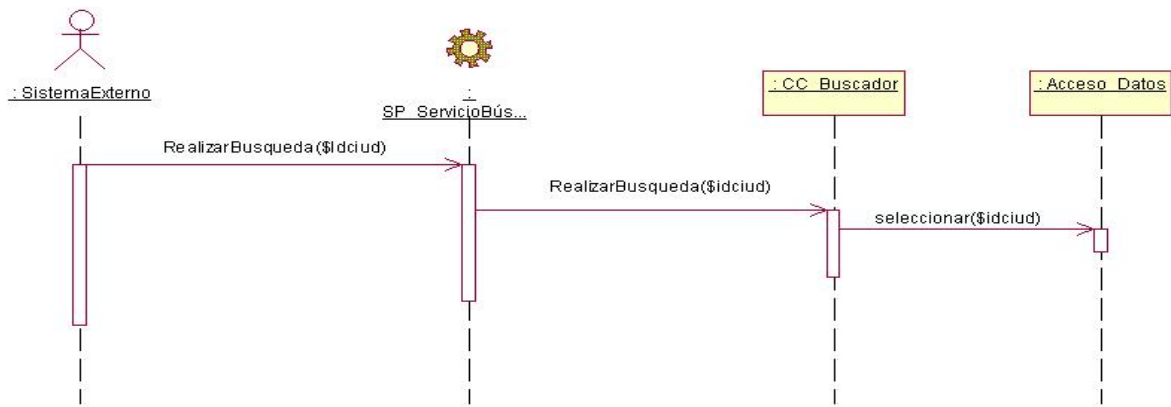
[Anexo14]. Diagrama de secuencia del caso de uso “Mostrar Reporte”.



[Anexo15]. Diagrama de secuencia del caso de uso “Salir del Sistema”.



[Anexo16]. Diagrama de secuencia del caso de uso “BuscadorSE”.





[Anexo17]. **Ejemplo de Reporte.**

	<b>Id</b>	<b>Nombre Ciudadanía</b>	
<input type="checkbox"/>	93	Cubana	Actualizar
<input type="checkbox"/>	91	Española	Actualizar
<input type="checkbox"/>	92	Peruana	Actualizar
<input type="checkbox"/>	79	Venezolana	Actualizar
<b>1</b>			

Eliminar

[Anexo18]. **Tratamiento de excepciones.**

**Figura 3.14 Mensaje de error del cliente.**

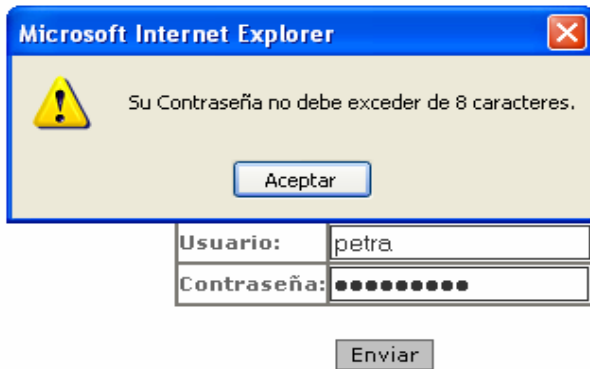


Figura 3.15 Mensaje de confirmación del cliente.

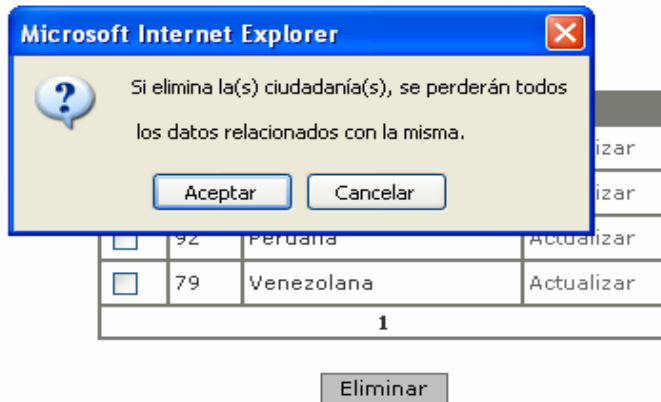


Figura 3.16 Mensaje de error del servidor.

