



Instituto Superior Politécnico “José Antonio Echeverría”
Facultad de Ingeniería Industrial
Centro de Estudios de Ingeniería y Sistema
Ingeniería Informática

BASE ÚNICA DEL CIUDADANO-UCI.
SERVICIOS BÁSICOS

Trabajo para optar por el título de
Ingeniería Informática

***Autor:** Raisa Perdomo Miranda*

***Tutor:** Ing. Maidely Calderón Montero*

Ciudad de La Habana

Junio de 2005

DECLARACIÓN DE AUTORÍA

Yo, *Raisa Perdomo Miranda*, declaro que soy la única autora de este trabajo y autorizo al Centro de Estudios de Ingeniería de Sistemas (CEIS) del Instituto Superior Politécnico José Antonio Echeverría (CUJAE) y a la Dirección de Informatización de la Universidad de las Ciencias Informáticas (UCI) a que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 15 días del mes de julio de 2005.

Firma del Autor

Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado Base Única del Ciudadano-UCI. Servicios Básicos, fue realizado en La Universidad de las Ciencias Informáticas (UCI). Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reporta un efecto económico que asciende a _____

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____

Representante de la entidad Cargo

Firma

Cuño

“Crear es la palabra de paso de esta generación”

José Martí

Agradecimientos

*A mis padres,
A Roly por su trabajo y esfuerzo,
A mi tutora Maidely,
A mis profesores de la Universidad de Camagüey,
A los profesores de la UCI que me brindaron su ayuda,
A todos mis compañeros, a Gea, y a Yuny y Melquiades por la ayuda, apoyo y
compañía en estos últimos meses de trabajo.*

Dedicatoria

*En especial a mami y a papi,
que han deseado este momento más que yo,
que se han preocupado tanto como yo,
que me han apoyado y ayudado en estos años de estudio,
que me han formado.*

En estos momentos el país está llevando a cabo la importante tarea de informatización de todos los procesos de la sociedad, donde se están aplicando nuevas tecnologías que contribuyen al desarrollo en todas las esferas de la economía y al aumento de la calidad de vida de la población. Para lograr este objetivo se está diseñando la Base Única del Ciudadano, que servirá de fundamento a todo el proceso.

La Universidad de Ciencias Informáticas (UCI) además de jugar un papel importante en este proceso de informatización, se encuentra enfrascada en un proceso similar: automatizar todas las actividades que en la Universidad se llevan a cabo. Para esto se necesita también una base de datos única a partir de la cual todas las aplicaciones se integren como un todo. Dada la importancia de esta base de datos, como eje central de este proyecto es necesario realizar un diseño adecuado garantizando que se almacene y controle la información distintiva de las personas que forman o han formado parte de la UCI, evitando que la información esté repetida en cada área donde la misma sea necesitada.

Para cumplimentar con el problema anteriormente expresado, con este trabajo queda diseñada la Base Única del Ciudadano-UCI, e implementada la librería de servicios Web. Para esto fue necesario tener en cuenta la migración hacia el software libre como parte de la política a seguir por la UCI, por lo que se consideró necesaria la elección de un gestor de base de datos y un lenguaje de programación de libres.

Esta base de datos podría ser la antesala del proceso migratorio que deberá permitir su integración con el resto de las aplicaciones existentes y con otras que puedan surgir en el proceso de automatización de la universidad.

Introducción.....	1
Capítulo 1.Fundamentación del tema.....	5
1.1. Introducción	5
1.2. Sistemas de bases datos.....	5
1.2.1. Evolución de las bases de datos.....	5
1.2.2. Definiciones.....	6
1.2.3. Características y objetivos de las bases de datos.....	6
1.3. Tendencias del software libre	7
1.3.1. Definiciones.....	8
1.3.2. Ventajas del software libre	9
1.3.3. Uso del software libre.....	9
1.3.3.1. Comportamiento en el mundo.....	9
1.3.3.2. Comportamiento en Cuba.....	10
1.3.3.3. Comportamiento en la Universidad de las Ciencias Informáticas.....	11
1.4. Sistemas automatizados existentes vinculados al campo de acción	12
1.4.1. Base de datos <i>Ciudadano</i>	12
1.5. Análisis comparativo de otras soluciones existentes con la propuesta.....	13
1.5.1. Base Única del Ciudadano.....	13
1.5.2. Sistema de la CUJAE.....	14
1.6. Propuesta de solución	14
1.7. Conclusiones	15
Capítulo 2. Tendencias y tecnologías actuales a considerar.....	16
2.1. Introducción	16
2.2. Fundamentación de la metodología utilizada.....	16
2.2.1. Proceso Unificado de Modelado (RUP).....	16
2.2.2. Lenguaje Unificado de Modelado (UML).....	17
2.2.3. Rational Rose.....	18
2.3. Fundamentación de los lenguajes utilizados	18
2.3.1 Gestor de base de datos PostgreSQL.....	18

2.3.2. Plataformas.Net y Mono	21
2.3.2.1. Plataforma.NET	21
2.3.2.2 Mono: La plataforma.NET Libre	22
2.4. Servicios Web	24
2.5. Conclusiones	25
Capítulo 3. Descripción de la solución propuesta.....	26
3.1. Introducción	26
3.2. Descripción de los procesos del negocio.....	26
3.3. Modelo del dominio.....	27
3.4. Requerimientos.....	28
3.4.1. Requerimientos funcionales	28
3.4.2. Requerimientos no funcionales	32
3.5. Descripción del sistema propuesto	34
3.5.1. Concepción general del sistema	34
3.5.2. Modelo de casos de uso del sistema.	35
3.5.2.1. Actores del sistema	35
3.5.2.2. Casos de uso del sistema.....	36
3.5.2.3. Diagramas de casos de uso del sistema	36
3.5.2.4. Descripción de los casos de uso	37
3.6. Conclusiones	46
Capítulo 4. Construcción de la solución propuesta	47
4.1. Introducción	47
4.2. Diagrama de clases	47
4.3. Diseño de la base de datos	56
4.4. Principios de diseño.....	59
4.4.1. Estándares en la interfaz de la aplicación	59
4.4.2. Concepción general de la ayuda.....	59
4.4.3. Tratamiento de excepciones	59
4.5. Estándares de codificación	59
4.6. Modelo de despliegue.....	60
4.7. Modelo de implementación	60

4.8. Conclusiones	63
Capítulo 5. Estudio de factibilidad	65
5.1. Introducción	65
5.2. Planificación.....	65
5.3. Costos.....	70
5.4. Beneficios tangibles e intangibles.	72
5.5. Análisis de costos y beneficios.	72
5.6. Conclusiones.	73
Conclusiones	74
Recomendaciones.....	75
Referencias bibliográficas	76
Bibliografía	78
Glosario de términos	81

Introducción

Hoy en día las bases de datos se han convertido en un recurso de vital importancia para las organizaciones, proporcionando la infraestructura requerida para gestionar la información de estas, cuando el volumen o la organización de los datos es imposible hacerla a mano.

Las bases de datos están jugando hoy un papel importante en el proceso de informatización de la sociedad cubana actual, proceso que se está llevando a cabo en nuestro país, y en el que están involucrados varios organismos, encabezados por el Ministerio de la Informática y las Comunicaciones.

En el comienzo de este amplio proyecto se está trabajando en el diseño de la Base Única del Ciudadano que servirá de base o centro de información al resto de las aplicaciones que gestionan la información de cada una de las esferas de la sociedad, como son vivienda, salud y educación, entre otras.

La UCI sin estar aislada de estas transformaciones, está realizando un proyecto similar, con el que se pretende la automatización de todos o casi todos los procesos que en la comunidad universitaria se realizan.

Para lograr este proyecto se necesita de una estructura integrada que tenga como centro de datos una base donde se almacenen la información distintiva de cada una de las personas que forman parte de la universidad. Permitiendo que la información no se repita en los diferentes departamentos o áreas del centro donde esta es usada. Con esta estructura se disminuyen los errores porque no existe redundancia de la información en bases de datos distintas, lo que además provocaría inconsistencia de la información.

En estos momentos existe en la UCI esta base de datos centralizada, llamada *Ciudadano*, a través de la cual las aplicaciones manipulan la información de las personas para hacer cualquier tipo de transacción.

A pesar de cumplir su objetivo como núcleo central de la mayoría de las aplicaciones de la UCI, la base de datos *Ciudadano*, presenta problemas en el almacenamiento de la información de las personas, tanto de las activas como de las

que causan baja del centro, puede existir información repetida en ambos casos lo que provoca que la base de datos aumente más su tamaño por la existencia de duplicidad y exista inconsistencia en la información. Además su gestor de base de datos y los softwares utilizados en la confección de los servicios Web asociados a la base de datos, no son los más adecuados en cuanto a la política que sigue el país y la universidad, referida a generalizar la utilización del software libre.

Este trabajo surge precisamente por la necesidad que tiene la UCI de dar solución a la **situación problémica** expuesta anteriormente; por lo que el **problema** a solucionar sería el diseño e implementación de la Base Única del Ciudadano-UCI teniendo en cuenta la política en la universidad referente a la utilización del software libre.

El **Objeto de estudio** de este trabajo serán las bases de datos y el software libre.

Mientras que el **Campo de acción** será el diseño de la Base Única del Ciudadano-UCI y el gestor de base de datos de libre distribución PostgreSQL junto con la plataforma libre del.NET, Mono.

En correspondencia con la idea central de esta tesis el **objetivo general** de este trabajo consiste en: diseñar la Base Única del Ciudadano UCI, de manera que ello permita evitar la duplicidad e inconsistencia de la información y el empleo de las herramientas de software libre cumpliendo con las políticas de migración.

Para cumplir este objetivo general se definen los siguientes **objetivos específicos**:

- Crear un nuevo diseño de la base de datos *Ciudadano*.
- Elaborar la capa de acceso a los datos a través de servicios Web, manteniendo la librería con estos servicios para que el resto de las aplicaciones hagan uso de los mismos.
- Selección de los softwares a utilizar acorde a la política del software libre, para la confección de la base de datos y la implementación de la librería de servios Web.

Las **tareas** trazadas para la realización de este trabajo son:

- Estudiar la información que se desea de las personas y como almacenarla, teniendo en cuenta como se hace en la base de datos ciudadano y las deficiencias que esta presenta.
- Analizar los sistemas de almacenamiento de personas que existen y que tienen similares características al sistema que necesita la UCI, con el fin de tomar lo mejor de estos.
- Confeccionar el análisis y el diseño del sistema a través de la metodología establecida *Rational Unified Process* (RUP)
- Identificar el gestor de base de datos a utilizar en correspondencia con el volumen de información a manejar y el lenguaje de programación adecuado para la implementación de los servicios Web.
- Confeccionar la base de datos en el gestor seleccionado e implementar la librería de servicios Web, teniendo en cuenta la seguridad de estos últimos.

El trabajo está estructurado en 5 capítulos:

- Capítulo 1. Fundamentación del tema

Se explican los conceptos fundamentales para el desarrollo del proyecto, de manera que ello posibilite poder comprender su funcionalidad e importancia. Además se explican varios sistemas utilizados en nuestro país a los cuales se tuvo acceso y que presentan características similares al proyecto que se pretende desarrollar.

- Capítulo 2. Tendencias y tecnologías actuales a considerar.

Se identifican las principales herramientas empleadas para la confección de la aplicación y se estudian distintas bases de datos con características similares a la que se pretende diseñar.

- Capítulo 3. Descripción de la solución propuesta.

Este capítulo se dedica fundamentalmente al análisis del sistema a desarrollar, se tratan los conceptos de modelo de dominio, y todo lo relacionado con la

funcionalidad de la aplicación (obtención de requerimientos, actores y casos de usos, y la relación de estos).

- Capítulo 4. Construcción de la solución propuesta.

Se enfoca la construcción de la solución mediante diagramas de clases, de datos, y se plantean los principios para el diseño y la implementación. Se construyen las funcionalidades que se definieron en el capítulo anterior.

- Capítulo 5. Estudio de factibilidad.

En este último capítulo se hace un estudio para obtener una estimación de algunos parámetros necesarios en el diseño y construcción de la aplicación, como son costo, tiempo y cantidad de personas para la realización del proyecto. Se hace además un análisis de los beneficios tangibles e intangibles.

Capítulo 1. Fundamentación del tema

1.1. Introducción

En este capítulo se da una visión de los conceptos de bases de datos y otros relacionados con el software libre y su utilización en el mundo, en Cuba y en particular en la UCI, para tener una idea más clara de los temas tratados en el trabajo. Además se realiza un estudio de los sistemas de bases de datos existentes en el país, a los cuales se ha tenido acceso y que presentan similares características al que se quiere desarrollar en la UCI.

1.2. Sistemas de bases de datos

1.2.1. Evolución de las bases de datos

Originalmente las aplicaciones que se desarrollaban para las organizaciones tenían como objetivo cubrir las necesidades de una tarea específica, centrándose solo en ella. De esta manera cada parte de la organización manejaba su información de manera independiente, en muchos casos usaban los mismos datos en diferentes aplicaciones lo que provocaba la inconsistencia y redundancia de la información.

Para evitar estos problemas se comienza a formular la lógica de los datos de la organización como un todo, se tienden a integrar los procesos, evitando aplicaciones aisladas. Se diferencian las estructuras lógicas y físicas, de manera que las aplicaciones solo se vinculen con las estructuras lógicas; es precisamente en este marco donde las bases de datos tienen gran importancia como una unidad tanto teórica como conceptual y física. Las bases de datos surgen precisamente como una herramienta que permite definir y almacenar las relaciones entre los datos de una organización.

1.2.2. Definiciones

Existen varios conceptos que definen las bases de datos entre ellos tenemos uno muy sencillo:

“Serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.”

[1]

Otra definición más completa sería:

“... un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.” [2]

Para lograr la recuperación y mantenimiento de la información en las bases de datos, se necesita de los Sistemas de Gestión de Bases de Datos (SGBD) que sería:

“... un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad.” [3]

1.2.3. Características y objetivos de las bases de datos

- Integridad de los datos y minimización de la redundancia.

Debe garantizar que en cualquier momento los datos almacenados son correctos, para lograr esto es necesario que se garantice además que exista el mínimo de redundancia en la información.

- Independencia de los datos y los programas de aplicación.

Diferentes aplicaciones necesitarán de los mismos datos. Los cambios en la estructura de almacenamiento de la base de datos deben ser invisibles a las aplicaciones.

- Integración y sincronización de las bases de datos.

Integración: Garantizar una respuesta a los requerimientos de diferentes aspectos de los mismos datos por diferentes usuarios. Entregar a cada aplicación lo que ella necesite sin importar la forma en que este estructurada la base de datos.

Sincronización: Garantizar el acceso múltiple y simultáneo a la base de datos, puede ocurrir en un momento dado que algunos datos estén siendo usados por más de un usuario a la vez.

- Facilidad en la manipulación de la información.

Debe tener la capacidad de hacer búsquedas rápidas por diferentes criterios

- Seguridad y protección de los datos

Seguridad: Proteger los datos del acceso no autorizado a los mismos.

Protección: Disponer de métodos para restaurar la base de datos ante algún fallo técnico, interrupción de la energía eléctrica, etc.

- Control centralizado

Se debe garantizar el control centralizado de la información de manera sistemática. Para ello deben de existir una o varias personas responsables de administrar los datos operacionales de la base de datos, que puedan decidir su contenido informativo, la estrategia de acceso de los usuarios, etc.

1.3. Tendencias del software libre

Alrededor de los años 60 y 70 el software no era considerado un producto, que se vendía, era un añadido que los vendedores de las computadoras daban a sus clientes para que pudieran usarlos, y eran compartidos por los programadores y desarrolladores. Fue a finales de los años 70 cuando se comenzó la comercialización del software, imponiendo restricciones a los usuarios, con el uso de las licencias.

En 1984 Richard Stallman comenzó a trabajar en el proyecto GNU, y en el 1985 se creó la Fundación del Software Libre (Free Software Foundation) que constituyó el

inicio del movimiento del software libre, tratando de cambiar la situación existente. A este proyecto se vincularon programadores de todo el mundo, y en la actualidad varias de las empresas dedicadas a la industria de software se han ido acogiendo a los principios de esta fundación.

1.3.1. Definiciones

Software Libre

El Software libre (free software) se refiere a aquellos softwares que no están controlados por su(s) fabricante(s), son completamente libres de usar por cualquier persona u organización. Una vez obtenidos puede ser usado, copiado, estudiado, modificado y distribuido libremente. Este concepto no se refiere a libertad de precios, sino a la posibilidad de ser usado por cualquiera haciendo uso de sus cuatro libertades:

- Libertad 0: Libertad de ser utilizado el programa por cualquiera con cualquier propósito.
- Libertad 1: Libertad de estudiar como funciona el programa y adaptarlo a sus necesidades (para esto es indispensable contar con el código fuente)
- Libertad 2: Libertad de ser copiado y distribuido libremente.
- Libertad 3: Libertad de modificarlo. Realizarle mejoras y hacerlas publicas (para esto es indispensable contar con el código fuente). [4]

Para evitar confusiones entre estas libertades y la libertad de precios, en algunas bibliografías se utiliza además el término *software gratis* referido al software que en ocasiones incluye el código fuente y que por supuesto no se paga.

Open source

Existe otro movimiento relacionado al tema, es el de software de código abierto (open source), aparecido en el año 1998 cuando Netscape anunció revelar el código fuente de su navegador de Internet, Netscape Navigator. El surgimiento de esa nueva expresión dio origen dos grupos de seguidores: los que adoptaron el nuevo término y los que creían que no era lo suficientemente exacto.

El Open source se diferencia del software libre en que plantea la posibilidad de realizarle modificaciones al código fuente, como puede hacerse con el software libre, pero puede existir en el software propietario.

“La diferencia fundamental entre los dos movimientos está en sus valores, en su visión del mundo. Para el movimiento open source, la cuestión de si el software debe ser de fuente abierta es una cuestión práctica, no ética. (...) Para el movimiento open source, el software no libre es una solución ineficiente. Para el movimiento de software libre, el software no libre es un problema social y el software libre es la solución.” [5]

1.3.2. Ventajas del software libre

- Permite que colaboren unos usuarios con otros.
- Ahorros en la adquisición de licencias. En los países pobres esto puede ayudar a reducir la brecha digital, evitando la dependencia tecnológica a las grandes empresas del software, permitiendo la generación de riquezas y recursos nacionales.
- Permite la reutilización del conocimiento: los programadores pueden utilizar lo que otros han hecho.
- Cada programador puede adaptar el software a sus necesidades.
- Combate efectivo a la copia ilícita de software. [6]

Por todas estas razones el software se puede catalogar como un bien público, en lugar de un bien privado.

1.3.3. Uso del software libre

1.3.3.1. Comportamiento en el mundo

“Hasta hace poco tiempo era imposible usar una computadora moderna sin la instalación de un sistema operativo propietario, provisto por el fabricante mediante licencias con altas restricciones para su uso. Nadie poseía permiso para compartir

programas libremente con otros usuarios y difícilmente alguien tenía el permiso de modificar los programas para adecuarlos a sus necesidades.” [7]

En medio de un mercado de software que tiene tendencias monopólicas, muchas empresas en la actualidad usan el sistema operativo GNU/Linux, en servidores y computadoras personales, y cada día son más las organizaciones que se suman a la Fundación del Software Libre. Por el ahorro que representa para los países, muchos de estos han adoptado políticas para el establecimiento del software libre principalmente en la administración pública. Entre esos países se encuentra China, México, Brasil, Colombia, Perú, Argentina, Uruguay, España, Alemania y Francia entre otros.

1.3.3.2. Comportamiento en Cuba

En nuestro país el uso del software propietario, principalmente Microsoft Office y Windows, está difundido en todos los sectores de la economía y en los Órganos y Organismos del Estado y el Gobierno. En todos los niveles de nuestra enseñanza ocurre lo mismo desde el nivel primario hasta el universitario.

A pesar de existir uniformidad en el uso de la plataforma basada en Windows, Cuba no puede comprar la licencia de estos softwares, solo algunas empresas lo ha hecho a través de terceros países. Esto ocurre por ser nuestro país víctima del bloqueo impuesto por el gobierno de Estados Unidos, el que prohíbe la adquisición de muchos productos. Las pocas licencias de software propietario que han entrado al país se han copiado libremente entre las instituciones, por lo que la implementación de estos softwares al país no le ha costado mucho.

Ahora Cuba se encuentra trabajando en el desarrollo de la Industria Cubana del Software, para suplir las necesidades nacionales y se convierta en una importante fuente de ingresos de divisas al país, pero como no podemos adquirir las licencias de los nuevos softwares y servicios corremos el riesgo de quedarnos con una tecnología atrasada, además de no poder exportar los productos cubanos basados en softwares propietarios.

Por tal motivo Cuba está organizando la migración progresiva al software libre en los organismos de la Administración Central del Estado, sobre la base del sistema

operativo GNU/Linux, eliminando así la presencia casi exclusiva del Windows en las máquinas. La Oficina para la Informatización de la Sociedad ha diseñado una estrategia que comprende acciones de organización, técnicas, diseño de un marco legal, así como la capacitación y el cambio paulatino de los sistemas de Windows a Linux. [8]

“La política será rectorada por un Grupo Nacional, que integran entre otros la propia Oficina, los ministerios de Justicia, del Interior, de Informática y las Comunicaciones, la Red Telemática de Salud (Infomed), la CUJAE, los Joven Club y la Universidad de las Ciencias Informáticas. Esta última, con más de 6 000 estudiantes, ya ha destinado una de sus facultades para el desarrollo de programas sobre Linux.” [8]

1.3.3.3. Comportamiento en la Universidad de las Ciencias Informáticas

La UCI como parte del proceso de informatización de la sociedad pretende la migración paulatina de todas las aplicaciones que se utilizan en la universidad. Para ello se está trabajando en distintos proyectos en los que se tiene en cuenta la utilización del software libre, se pretende además incluir en algunas asignaturas herramientas de este tipo y dar postgrados relacionados con el tema. Otro proyecto importante que ya está funcionando, es el portal de software libre en la UCI: <http://softwarelibre.uci.cu> donde se promueve la utilización del mismo, se realizan foros, debates y se tiene a disposición información actualizada y descargas de programas de este tipo. En general se preparan distintas actividades para motivar la utilización del software libre como fue el “Festival de Instalación”. Se está trabajando en el desarrollo de una distribución de Linux basada en el sistema Gentoo, la cual cuenta con una amplia gama de herramientas de desarrollo.

La Facultad 10 y el grupo PULSAR de la dirección de informatización pretenden la masificación de las técnicas y herramientas del software libre.

En la Facultad 10 se han planificado 8 cursos optativos a ser acreditados que incluyen elementos básicos y avanzados del Sistema Operativo Linux, programación básica y avanzada de aplicaciones Web y de escritorio y gestores de Bases de Datos.

Con la migración de la base de datos *Ciudadano* hacia un gestor libre, se están sentando las bases para que en un plazo no muy largo de tiempo puedan migrar las aplicaciones que dependen de ella.

1.4. Sistemas automatizados existentes vinculados al campo de acción

1.4.1. Base de datos *Ciudadano*

Actualmente en la UCI existe una base de datos para el almacenamiento de las personas que forman parte de la misma. La cual cumple con el objetivo principal de ser eje central de casi todas las aplicaciones de la universidad, permitiendo la integración y el intercambio de información con el resto de las aplicaciones a través de una librería de servicios Web.

Esta base de datos tiene problemas en el almacenamiento de la información al insertar una nueva persona o al darle de baja y pasarla al historial, en ambos casos no se verifica que la persona exista provocando que en la base de datos haya información repetida y en muchos casos desactualizada, este cúmulo de información podría afectar el buen funcionamiento del sistema, en cuanto a velocidad de respuesta y veracidad de la información.

La base de datos *Ciudadano* está montada sobre un servidor SQL Server 2000, y su librería de servicios implementada en C# sobre la plataforma.NET, todos softwares propietarios, cuando la universidad al igual que en el resto del país aspiran a migrar todas sus aplicaciones al software libre. Se necesita migrar primero esta base de datos por servir de centro de información al resto de las aplicaciones que dependen de ella y que posteriormente realizarán la migración.

1.5. Análisis comparativo de otras soluciones existentes con la propuesta

1.5.1. Base Única del Ciudadano

La informatización de la Sociedad Cubana “es la utilización ordenada y masiva de las tecnologías de la información y las comunicaciones en todas las esferas de la sociedad cubana, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos. La Informatización se basa en los principios y valores fundamentales de nuestro sistema social, con una marcada tendencia a defender las conquistas que hemos logrado.” [9]

Para este proceso se han trazado varios programas principales:

- IS-ITH: Infraestructura, Tecnologías y Herramientas (“Infratech”).
- IS-CIUD: Sistemas y Servicios Integrales para los ciudadanos.
- IS-GOB: Informatización del Gobierno, la Administración y la Economía.
- IS-MUN: Informatización Territorial.
- IS-CULT: Fomento de la Cultura digital.
- IS-JCLUB: Fortalecimiento del papel de los Joven Club.
- IS-ICSW: Fomento de la Industria Nacional de las Tecnologías de la Información y las Comunicaciones.
- IS-IDA: Investigación, desarrollo y asimilación tecnológica. [9]

Para que estos programas y muchos otros que surjan en el proceso de informatización puedan llevarse a cabo es necesario tener una base de datos donde se almacene la información de todos los ciudadanos del país, la misma servirá de fuente de información para que se integren el resto de las aplicaciones relacionadas a estos programas.

Esta base de datos es precisamente La Base Única del Ciudadano que está siendo diseñada en estos momentos y tiene gran similitud con la base de datos *Ciudadano* que está implantada en la UCI la cual se pretende mejorar con el presente trabajo.

Los diseñadores de esta base de datos están estudiando el problema que existe al tener el número del carné de identidad como llave primaria, en las bases de datos que se están usando actualmente en cada provincia. Se plantea entonces crear una llave a partir de los datos registro civil, tomo, folio y año de asiento del carné de identidad, pero en el momento de la realización de este documento no se había llegado a ninguna conclusión. Con respecto a los otros datos que son de interés almacenar, en su mayoría son los mismos que se tienen en *Ciudadano*. Estas bases de datos difieren en el almacenamiento del historial de personas, pues para la Base Única del Ciudadano el historial estaría compuesto por las personas fallecidas y en el caso de la UCI, por personas dadas de baja del centro las cuales en un futuro podrían regresar a formar parte de la universidad.

1.5.2. Sistema de la CUJAE

En el Instituto Superior Politécnico José Antonio Echeverría, existen varias bases de datos para el almacenamiento de la información referente a la identificación de las personas que forman parte de la institución, se encuentran separados los datos de estudiantes y profesores en una base y los de trabajadores en otra.

En la base de datos donde se encuentran los profesores y estudiantes no solo se tiene la información significativa, sino la relacionada con los cursos, matriculas, facultades, áreas, y otra.

La llave de identificación de las personas es el carné de identidad, lo que representa un error por el problema que existe con la duplicidad de este número.

Las personas dadas de baja en la institución pasan a un historial donde estarán repetidas la cantidad de veces que hayan causado baja del centro.

1.6. Propuesta de solución

Después de realizado el análisis de varias bases de datos con similares características a la base de datos que necesita la UCI para el almacenamiento de la información distintiva de las persona, se determina realizar un diseño similar al de la

base de datos *Ciudadano* existente en la universidad, a la que se le incluirán algunos aspectos que solucionen las deficiencias que esta presenta.

Se analizó para la llave de la persona, la propuesta que se tiene para la Base Única del Ciudadano del país, pero como no se ha comprobado que esta nunca vaya a repetirse, se decidió mantener la llave existente en la base *Ciudadano* de la UCI.

Para el almacenamiento de las personas que han causado baja, se propone un diseño que permita el control de estas, teniendo en la base de datos una sola vez la información de la persona, pero controlando todos los movimientos que esta ha efectuado.

1.7. Conclusiones

En este capítulo se trataron algunos aspectos de importancia relacionados con la utilización del software libre, se presentan definiciones asociadas al tema, ventajas y comportamientos a nivel mundial, nacional y particularmente en la UCI. Se habló de los conceptos fundamentales de las bases de datos para poder comprender la forma de almacenamiento de la información y sus características.

Además se analiza la base de datos *Ciudadano*, y otras dos existentes en el país, una de ellas es la base de datos de la CUJAE y la otra, la Base Única del Ciudadano, que aunque todavía no rebasa la etapa de diseño, es la que más puntos en común presenta con la base de datos *Ciudadano*.

Capítulo 2. Tendencias y tecnologías actuales a considerar

2.1. Introducción

Con el avance de las tecnologías y la informática cada día aparecen nuevas herramientas de trabajo para la gestión de la información en las organizaciones. Lo que trae consigo que los desarrolladores deban tomar más tiempo, tanto para decidir cuales son las herramientas idóneas según la aplicación que se quiera realizar, como para seleccionar la metodología de diseño adecuada.

Por este motivo en el presente capítulo se hace un estudio de la metodología a utilizar para el diseño y análisis, y de las herramientas utilizadas para la construcción de la aplicación. Para esto se abordan aspectos importantes de cada una de ellas y se realizan comparaciones con otras herramientas del mismo tipo.

2.2. Fundamentación de la metodología utilizada

Las computadoras son cada día más potentes por el avance de las tecnologías y el mercado es exigente. Por tanto necesitamos que los softwares que utilizamos para el manejo de las computadoras y el intercambio de información se adapten mejor a nuestras necesidades, es imprescindible contar con una nueva forma de trabajo, un método común, un proceso o metodología que sirva de guía para ordenar las actividades de un equipo en el proceso de desarrollo de un software.

2.2.1. Proceso Unificado de Modelado (RUP)

Definición:

“El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación,

diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.” [10]

RUP utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para describir un sistema. Este proceso se resume en tres fases claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Esto es lo que hace único al Proceso Unificado. [10]

El Proceso Unificado ha establecido un marco de trabajo en el que se integran todas las fases de un proceso polifacético, que tenga en cuenta ciclos, fases de trabajo, gestión de riesgo, control de calidad, gestión de proyecto y control de la configuración.

2.2.2. Lenguaje Unificado de Modelado (UML)

Este lenguaje es el resultado de la unificación de los métodos de modelado orientados a objetos de Booch, Rumbaugh (OMT: Object Modeling Technique) y Jacobson (OOSE: Object-Oriented Software Engineering).

UML es un lenguaje basado en notación gráfica la cual permite: especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objeto. Define reglas sintácticas que especifican como combinar elementos del lenguaje.

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Hay que tener en cuenta que el estándar UML no es un proceso, no es una metodología de desarrollo, sino una notación, un lenguaje. [10]

Mediante el uso de UML como lenguaje estándar, el mercado experimenta un soporte por herramientas mucho mejor que el que nunca haya tenido ningún lenguaje modelado. UML es hoy un estándar industrial ampliamente utilizado. [10]

2.2.3. Rational Rose

Es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información, pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

2.3. Fundamentación de los lenguajes utilizados

Para el estudio y selección de las herramientas a utilizar se tuvieron en cuenta varios criterios, los cuales no fueron en ningún caso factor decisivo en la selección:

- Experiencia de los programadores en el trabajo con cada herramienta.
- Estructura que necesita adoptar la UCI.
- Necesidad de la universidad y el país de utilizar el software libre.
- Características, ventajas y desventajas de cada herramienta según las características del proyecto a desarrollar.

2.3.1 Gestor de base de datos PostgreSQL

PostgreSQL es uno de los tantos gestores de bases de datos de libre distribución, desarrollado por voluntarios y que tienen gran calidad y superioridad con respecto a los softwares propietarios.

El PostgreSQL está basado en el modelo relacional e incorpora conceptos del modelo orientados a objetos, como es la herencia. Es uno de los productos libres y de código abierto más avanzados entre los gestores de bases de datos. Soporta casi todas las sintaxis SQL (incluyendo subconsultas, disparadores (trigger), reglas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto

de enlaces con lenguajes de programación. Además incorpora definición de tipos de datos y funciones, además de herencias.

Puede ser integrado al ambiente Windows, permitiendo la migración desde distintos gestores de bases de datos y ser usado como servidor de base de datos en aplicaciones realizadas sobre Windows en lenguajes como Visual Basic, FoxPro, Visual FoxPro, Visual C/C++, Delphi,.NET, etc.

Límites de la base de datos:

Elemento	Capacidad Máxima
Tamaño Máximo para Base de Datos	Ilimitado
Tamaño Máximo para Tabla	64 Tera Bytes (en todos los sistemas operativos)
Tamaño Máximo para Registro (fila)	Ilimitado a partir de la versión 7.1
Tamaño Máximo para Campo	1 Giga Byte a partir de la versión 7.1
Número Máximo de Registros por Tabla	Ilimitado
Número Máximo de Columnas por Tabla	1600
Número Máximo de Índices por Tabla	Ilimitado

Comparando con MySQL.

MySQL es uno de los gestores de base de datos más importantes y estandarizados del mundo, desarrollado bajo la filosofía de código abierto. Por lo que se debe de considerar para hacer una buena elección de un gestor de base de datos libre.

	MySQL	PostgreSQL
--	--------------	-------------------

Plataformas	Linux, Solaris,HP-UX, MacOS, AIX, SCO, IRIX, FreeBSD, NetBSD, OpenBSD, Windows 95/98/NT/2000, BSDI, DEC, OS/2, Compaq Tru64	Linux, Solaris,HP-UX, AIX, IRIX, FreeBSD, OpenBSD, NetBSD, MacOS, SCO OpenServer, SCO Unixware, BeOS, BSDI, Compaq Tru64, QNX, WinNT/2000.
Cumplimiento con el SQL estándar	Media	Alta
Velocidad:	Media / Alta	Media (2 a 3 veces más lento que MySQL)
Estabilidad:	Alta / Muy Alta	Alta
Integridad de datos	NO	Si
Seguridad	Alta	Alta
Soporte de Vistas	No	Si
Soporte Subconsultas	NO	Si
Replicación	Si	Si
Procedimientos almacenados	Si (a partir de 5.0)	Si
Soporte Disparadores	Si (un poco rudimentarios desde 5.0)	Si
Integridad referencial	NO	Si
Interfaces de programación	ODBC, JDBC, C/C++, OLEDB, Delphi, Perl, Python, PHP	ODBC, JDBC, C/C++, SQL embebido (en C), Tcl/Tk, Perl, Python, PHP
Transacciones	Si	Si
Claves foráneas	Si	Si
Consumo de recursos	Consume pocos	Consume bastante mas recursos y

	recursos tanto de CPU como de memoria	carga mas el sistema
--	--	----------------------

Se selecciona el PostgreSQL como gestor de base de datos a utilizar para la construcción de la Base Única del Ciudadano-UCI, por presentar las características que necesita la base de datos Ciudadano. PostgreSQL es ideal para bases de datos donde se requiera recibir gran cantidad de peticiones, por lo que es factible su uso en grandes bases de datos. Además de presentar soporte de vistas, subconsultas, tener integridad referencial y buena escalabilidad. En MySQL ocurre todo lo contrario, por lo no es la solución para la Base Única del Ciudadano-UCI, donde debe manejarse gran volumen de información y es necesario el acceso continuo, dando respuesta a las peticiones en un corto plazo de tiempo.

2.3.2. Plataformas.Net y Mono

2.3.2.1. Plataforma.NET

“.NET ofrece la capacidad de desarrollar, implementar, administrar y utilizar soluciones conectadas a través de Servicios Web XML, de manera rápida, económica y segura. Estas soluciones permiten una integración más rápida y ágil entre las empresas y el acceso a información a cualquier hora, en cualquier lugar y a través de cualquier dispositivo.” [11]

“.NET permite que los sistemas y aplicaciones, ya sea nuevos o existentes, conecten sus datos y transacciones independientemente del sistema operativo, tipo de computadora o dispositivo móvil que se utilice, o del lenguaje de programación empleados para crearlo.” [11]

Características:

- Independencia del lenguaje que se programe: Tiene un Common Intermediate Language (CIL) el cual proporciona todo lo necesario para utilizar la mayoría de los lenguajes actuales. El CIL incluye:
 - Common Type System (CTS)
 - Common Language Specification (CLS)
 - Virtual Execution System (VES)
 - Metadata Definitions and Semantics
- Contiene una librería de clases común para todos los lenguajes que soporta:
- La Framework Class Library (FCL) Esta Integrada por más de 4000 clases, objetos y métodos. En ella se incluye el soporte de ficheros XML, servicios web, programación gráfica, criptografía, bases de datos, manejo de texto, funciones matemáticas, etc.
- Es independiente del sistema operativo:
Compilación del código fuente a un código intermedio (CIL), y a través del Common Language Runtime (CLR) traduce del CLI al código de máquina.

2.3.2.2 Mono: La plataforma.NET Libre

Es una plataforma para correr y desarrollar aplicaciones modernas, basadas en el estándar ECMA/ISO. Este proyecto implementa una plataforma de desarrollo libre, basado en Linux y compatible con Microsoft.NET, por lo que en Mono pueden correr programas existentes en esta plataforma.

”Cualquier aplicación compilada con mono/.NET en cualquier arquitectura se puede ejecutar directamente en mono/.NET en cualquier otra arquitectura (que soporte mono) siempre y cuando las librerías que se utilicen sean las librerías definidas en el estándar ECMA.” [12]

La plataforma Mono implementa las siguientes partes de la tecnología.NET:

- Common Language Runtime (CLR)

- Compilador/Desensamblador IL
- Compilador C#
- Compilador Visual Basic.NET
- Librería de clases (FCL)
- Otras librerías de funcionalidades. [11]

Características de la plataforma mono

- Independencia de lenguaje: puedes usar clases escritas en cualquier lenguaje soportado por Mono.
- Multiplataforma.
- Extensa librería de clases para el acceso a bases de datos: PostgreSQL, MySQL, DB2, TDS, Sybase, Oracle, ODBC, Gnome-GDA, MS SQL, OLE DB, etc.
- Velocidad: el lenguaje intermedio se compila en cada plataforma con unos compiladores muy rápidos (JIT) lo que lo hace mucho más rápido que lenguajes interpretados como PHP ó Python y más rápido en la compilación (JIT) que Java. Únicamente un poco más lento que C.
- Gestión automática de memoria.
- Seguridad.
- Para aplicaciones Web se puede usar cualquier lenguaje soportado por Mono.
- Contiene una implementación de la librería de clases de.NET.

”Mono proporciona las funciones necesarias para crear servicios Web, esto incluye las tecnologías XML, SOAP, ASP.NET, y Remoting así como el acceso a bases de datos, a través ADO.NET, de tipo Oracle, MySQL, DB2, SQL Server, o ProgreSQL. Con estas librerías podemos portar servicios desarrollados en Windows o escribir los nuestros propios pudiendo utilizar Apache como servidor Web.” [13]

“Mono es una plataforma eficaz y sencilla para el desarrollo de aplicaciones, para portar soluciones ya existentes para Windows al mundo Linux, y en general como

una excelente herramienta para podernos focalizar en el desarrollo de programas multiplataforma y servicios Web. Es sin duda una plataforma que será crucial para que las empresas puedan migrar sus aplicaciones al mundo Linux y para que nuevas aplicaciones puedan ser desarrolladas con el menor coste posible” [13]

Mono Develop es el completo entorno de desarrollo integrado con que consta la plataforma Mono. Herramienta sobre la cual se desarrollan las aplicaciones.

2.4. Servicios Web

“Un Web Service es un componente de software que se comunica con otras aplicaciones codificando los mensajes en XML y enviando estos mensaje a través de protocolos estándares de Internet tales como el Hypertext Transfer Protocol (HTTP).” [14]

Un Web Service no tiene interfaz de usuario, brinda servicios a distintas aplicaciones, recibiendo las solicitudes de estas a través de un mensaje formateado en XML, a el cual le realiza una acción y luego devuelve un mensaje de respuesta también formateado en XML. Esto es posible sin importar el sistema operativo o la plataforma en que se ejecutan. Además las rutinas de los Servicios Web se actualizan de forma transparente para el programador y para el encargado de mantenimiento de la aplicación.

Un Web Service usa estándares basados en XML como: ebXML, RosettaNET, SOAP, UDDI, WSDL, y muchos otros protocolos para permitir la comunicación entre las aplicaciones.

Con el concepto de Web Service, una computadora ya no se considerara como un núcleo de cómputo sino como un repositorio de servicios de n aplicaciones distribuidas por Internet.

Existen dos funciones fundamentales en el modelo de programación de servicios Web XML:

- Creación de un servicio Web XML. Cuando se crea un servicio Web XML, se crea una aplicación que expone funcionalidad a clientes de servicios Web XML.

- Acceso a un servicio Web XML. Cuando se tiene acceso a un servicio Web XML, la aplicación de cliente busca, utiliza y hace referencia a la funcionalidad que contiene el servicio Web XML. El cliente de un servicio Web XML puede ser una aplicación basada en un explorador, un componente o incluso otro servicio Web XML. [15]

2.5. Conclusiones

En este capítulo quedan definidas las herramientas a utilizar para el desarrollo del proyecto. Para lo que se realizó el análisis de cada una y la fundamentación de su elección. Se decide utilizar como gestor de base de datos el PostgreSQL y la plataforma Mono para la implementación de los servicios Web, por las facilidades que esta brinda de compilar los códigos hechos en la plataforma.NET y por la experiencia de los programadores en la misma.

Capítulo 3. Descripción de la solución propuesta

3.1. Introducción

En este capítulo se hace la descripción de la solución propuesta para la implementación del sistema, se describen los procesos del negocio y se decide la utilización de un Modelo de Dominio, donde se describe el entorno del problema. Se explica el comportamiento del sistema a través de los requisitos funcionales y no funcionales, casos de uso y su relación con los actores del sistema, a través del diagrama de casos de usos.

3.2. Descripción de los procesos del negocio

La mayoría de las aplicaciones de la UCI necesitan manipular los datos relativos a la identificación de las personas, para poder cada una, cumplir con sus objetivos específicos. Esta información no puede existir físicamente en cada una de las aplicaciones, porque la misma siempre estaría desactualizada en las distintas bases de datos, provocando el mal funcionamiento de las aplicaciones.

Por tanto se tiene una base de datos centralizada donde se almacena esa información distintiva y con la cual pueden interactuar las aplicaciones, haciendo un uso correcto de ella, al evitar en alguna medida, la redundancia e inconsistencia de la información.

Esta base de datos incluye una librería de servicios Web a través de los cuales las aplicaciones autorizadas pueden manipular la información que necesitan.

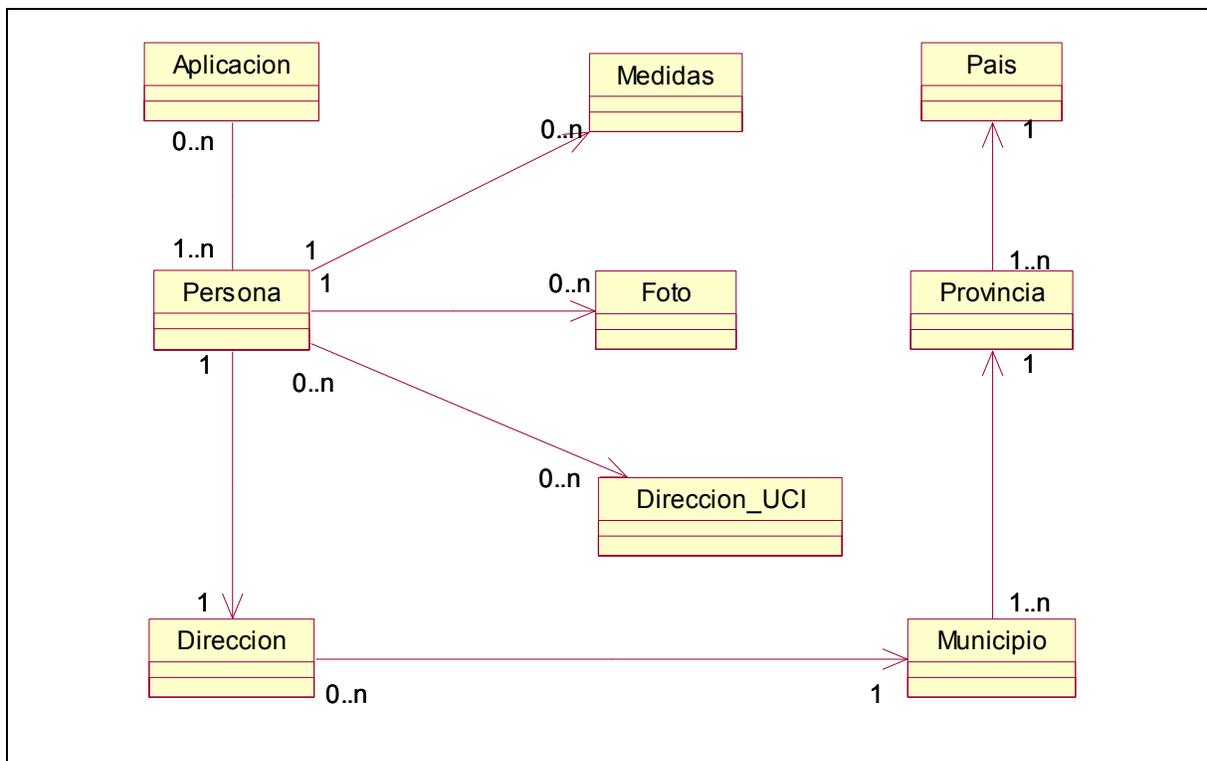
3.3. Modelo del dominio

Por la descripción de los procesos del negocio expuesto anteriormente, se puede definir que la aplicación que se encarga de almacenar la información de las personas de la UCI, no representa un proceso desarrollado en la Universidad, ni ninguna actividad en específico; su objetivo es simplemente la gestión y presentación de la información de forma centralizada, para que sea usada por otros procesos.

Por tal motivo, se utilizará un modelo del dominio que nos permite la representación visual de los conceptos u objetos significativos del dominio o área del problema.

Para comprender mejor este modelo se explican a continuación los dos conceptos fundamentales que se manejan: *Persona* y *Aplicación*. Se llamará *Persona* a cualquiera que forme parte de la UCI de la cual se almacene su información. Se considera *Aplicación* a cualquier sistema de la UCI que esté autorizado a obtener información de las personas o a modificarla.

El resto de los conceptos representados en el modelo están asociados a la información que se necesita almacenar de las personas.



3.4. Requerimientos

3.4.1. Requerimientos funcionales

1. Insertar una nueva persona en la base de datos.
2. Dar de baja a una persona de la base de datos.
3. Realizar búsquedas en la base de datos por diferentes criterios.
 - 3.1. Dado CI, Id_Persona, Id_UCI, o usuario de dominio devuelva todos los datos de la persona.
 - 3.2. Dado el CI o Id_UCI de una persona devuelva la fecha en que causó baja de la UCI y la causa que motivó la misma.
 - 3.3. Dado un tipo de persona devuelva la cantidad de personas de ese tipo.
 - 3.4. Dado un tipo de persona devuelva lista de personas de ese tipo (Nombre completo, Id_UCI).
 - 3.5. Devolver un listado con el identificador de las personas becadas de la UCI.

- 3.6.** Dado el identificador de dos personas devolver el tipo de relación de parentesco que existe entre ellas.
- 3.7.** Dado el usuario de dominio de una persona devuelva el Id_UCI de la misma.
- 3.8.** Dado el nombre de una provincia devuelva CI de las personas que pertenecen a la misma.
- 3.9.** Dado el nombre de una provincia devuelva Id_UCI de las personas que pertenecen a la misma.
- 3.10.** Dado el CI o el Id_UCI de una persona devuelva el historial de las direcciones UCI en las que ha vivido.
- 3.11.** Dado el nombre de una provincia devuelva un listado con los municipios que pertenecen a la misma.
- 3.12.** Dado el Id_UCI, Id_Persona o usuario del dominio de una persona devuelva todos los datos de la persona.
- 3.13.** Dado el CI o el Id_Persona de una persona devuelva las direcciones UCI de la misma.
- 3.14.** Dado el CI o el Id_Persona devuelva los datos de la dirección permanente de esa persona.
- 3.15.** Dado el CI o el Id_Persona de una persona devuelva la ciudadanía.
- 3.16.** Dado el CI o el Id_Persona devuelva las medidas de esa persona.
- 3.17.** Dado el Id_UCI, CI o Id_Persona de una persona devuelva otros datos del CI (No Serie CI, Tomo, Folio, Fecha Registro, Observaciones).
- 3.18.** Dado el Id_UCI o CI de una persona devuelva Id_Persona.
- 3.19.** Dado un Id_Persona devuelva su nombre de usuario.
- 3.20.** Dado un Id_Persona devuelva todas sus relaciones de parentesco con personas activas.
- 3.21.** Dado un Id_Persona devuelva si es becado o no.

- 3.22.** Dado un Id_Persona devuelva sus fotos.
- 3.23.** Dado un Id_Persona devuelva su última dirección en la UCI.
- 3.24.** Realizar búsquedas fonéticas.
 - 3.24.1.** Por primer nombre.
 - 3.24.2.** Por segundo nombre.
 - 3.24.3.** Por primer apellido.
 - 3.24.4.** Por segundo apellido.
- 4.** Realizar actualizaciones en los datos de las personas.
 - 4.1.** Actualizar datos identificativos de la persona (1er Nombre, 1er Apellido, 2do Apellido, Id_UCI, Becado, Estado civil, Tipo persona, Alias).
 - 4.2.** Actualizar la ciudadanía de la persona.
 - 4.3.** Actualizar las relaciones de parentesco.
 - 4.4.** Actualizar otros datos de la persona (# serie CI, tomo, folio, registro civil, observaciones).
 - 4.5.** Actualizar dirección permanente (Residencia permanente, No, Entre calles, Localidad).
- 5.** Manipular medidas de la persona (talla, peso, fecha de registro).
 - 5.1.** Insertar medidas de una persona.
 - 5.2.** Eliminar medidas de una persona.
 - 5.3.** Actualizar medidas de una persona.
- 6.** Manipular usuario del dominio de una persona.
 - 6.1.** Insertar el usuario del dominio persona.
 - 6.2.** Eliminar el usuario del dominio persona.
 - 6.3.** Actualizar el usuario del dominio persona.

7. Manipular las fotos de la persona (Directorio, Chiquita (1x1), CI (3x3), Pasaporte (5x4)).
 - 7.1. Insertar foto de un tipo determinado.
 - 7.2. Eliminar foto de un tipo determinado.
 - 7.3. Actualizar foto de un tipo determinado.
8. Manipular dirección UCI de la persona (Edificio, Apto, Calle, Localidad, Fecha de registro).
 - 8.1. Insertar Direccion_UCI de una persona.
 - 8.2. Eliminar Direccion_UCI de una persona.
 - 8.3. Actualizar Direccion_UCI de una persona.
9. Manipular número de pasaporte. (Id_Persona, Num_Pasaporte)
 - 9.1. Insertar número de pasaporte de una persona.
 - 9.2. Eliminar número de pasaporte de una persona.
 - 9.3. Actualizar número de pasaporte de una persona.
10. Manipular aplicaciones. (Id_Ap, Nombre_Ap, Password)
 - 10.1. Insertar aplicaciones de la base de datos.
 - 10.2. Eliminar aplicaciones de la base de datos.
 - 10.3. Actualizar aplicaciones en la base de datos.
11. Manipular servicios Web. (Id_WS, Nombre_WS)
 - 11.1. Insertar servicios Web en la base de datos.
 - 11.2. Eliminar servicios Web en la base de datos.
12. Manipular asignación a las aplicaciones de los servicios Web que podrán usar. (Id_WS, Id_Ap)
 - 12.1. Insertar asignación de un servicio Web a una aplicación.
 - 12.2. Eliminar asignación de un servicio Web a una aplicación.

13. Verificar permisos de usuario para usar los servicios Web.

14. Verificar permisos de administración.

3.4.2. Requerimientos no funcionales

Apariencia o Interfaz externa:

Diseño sencillo para el proceso de seguridad de los servicios Web por parte del administrador del sistema.

Usabilidad:

Se proporciona una librería, compuesta por servicios Web, los que permiten que las aplicaciones autorizadas, puedan acceder a la información almacenada en la base de datos de forma rápida y eficiente. La interfaz Web para el administrador es muy sencilla y fácil de usar.

Rendimiento:

El sistema debe dar respuesta lo más rápido posible a las solicitudes con un mínimo margen de error, debe permitir la conexión de varios usuarios a la base de datos y poder responder a todos sus pedidos en un corto plazo de tiempo, a pesar del volumen de información que se maneja y la demanda de esta.

Soporte:

Debe permitir darle mantenimiento a la base de datos y actualizar la librería de servicios Web, para brindar un mejor servicio.

Portabilidad:

El uso de servicios Web permite que sean usados desde distintos tipos de plataformas de programación.

Seguridad:

Se debe dar la posibilidad de aplicarle las políticas de seguridad existentes en el centro.

- **Confidencialidad:** Garantizar que solo aplicaciones autorizadas tengan acceso a la base de datos y a la librería de servicios.

- **Integridad:** Protección de la información contra la corrupción y estados inconsistentes.
- **Disponibilidad:** Los usuarios autorizados tendrán acceso a la información, sin que esta se retrase por los mecanismos o dispositivos de seguridad.

Legales:

La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL que es totalmente gratis.

Confiabilidad:

Debe tener servidores de respaldo de la base de datos para casos de problemas en el servidor principal, como puede ser la pérdida de la información en caso de fallos en el momento de hacer operaciones sobre los datos (insertar, eliminar o modificar), permitiendo su recuperación.

Ayuda y documentación en línea:

Debe brindarles a los usuarios la documentación en línea necesaria para facilitarles el manejo de los servicios Web. Se presenta información como: descripción, parámetros de entrada, parámetros de salida, forma de conexión y ejemplos de su uso.

Requerimientos de Software:

Los servicios Web pueden ser ejecutados sobre cualquier plataforma. Debe disponerse de un servidor de PostgreSQL para la base de datos y plataforma Mono o.NET.

Requerimientos de Hardware:

Deben disponerse de una red que permita el tránsito de la información que se genera al hacer uso de los servicios Web, soportar grandes volúmenes de datos y tener alta velocidad de procesamiento. Debe estar provista de una máquina con 1GB de RAM para ejecutar el servidor de PostgreSQL.

3.5. Descripción del sistema propuesto

3.5.1. Concepción general del sistema

Teniendo en cuenta los requisitos planteados en el epígrafe anterior, se propone diseñar una base de datos que permita el almacenamiento y manipulación de la información de las persona, una librería de servicios Web y un módulo administrativo para el control de los servicios Web que utiliza cada aplicación, para este último se propone la implementación de una aplicación Web sencilla.

Las aplicaciones AKADEMOS y Capital Humano son las principales aplicaciones que van a interactuar con el sistema, porque son las encargadas de la manipulación de la información más importante que se almacena en la base de datos, ellas son las únicas aplicaciones encargadas de insertar o dar de baja a una persona.

Otras aplicaciones que también pueden manipular la informaciones pero más específicas en su radio de acción, son: Aplicación de Salud (medidas), Aplicación de Solapines (fotos), Servicios telemáticos (usuarios del dominio), Sistema de Cooperación Internacional (números de pasaporte) y Aplicación de beca (direcciones de la UCI).

Existe otro grupo (Usuario General) que solo de información para realizar sus procesos, para esto existe un grupo de servicios encargados de realizar búsqueda de información por diferentes criterios, tratando de satisfacer las necesidades de las aplicaciones. En este grupo se pueden incluirse además los dos anteriores.

En el módulo administrativo se debe controlar que las aplicaciones que usan los servicios Web sean las autorizadas, para esto el administrador es el encargado de manipular esta información en la base de datos. Se almacena cada aplicación con su clave y el grupo de servicios que puede usar. El administrador puede insertar, eliminar y modificar servicios Web y aplicaciones. Cuando el programador de una aplicación solicita usar un servicio es el administrador el encargado de autorizarlo.

En resumen con este sistema se podrá almacenar la información significativa de cada una de las personas que forman parte de la UCI, se podrá acceder a ella a través de una librería de servicios Web, los que serán usados solo por las aplicaciones autorizadas que serán almacenadas y controladas en la base de datos.

3.5.2. Modelo de casos de uso del sistema.

3.5.2.1. Actores del sistema

Nombre del Actor	Descripción
AKADEMOS	Aplicación para la gestión de matrícula y control de los estudiantes UCI. Requisitos asociados: 1, 2, 3, 4, 13
Capital Humano	Aplicación que controla los datos de todos los trabajadores, tanto docentes, como no docentes de la UCI. Requisitos asociados: 1, 2, 3, 4, 13
Usuario Especial	Actor abstracto dentro del sistema, está formado por la unión de los actores AKADEMOS y Capital Humano, para representar la acción que ambos realizan sobre casos de uso comunes.
Usuario General	Conjunto de todas las aplicaciones que interactúan con la base de datos para realizar búsquedas de información. Requisitos asociados: 3, 13
Administrador del sistema	Persona encargada de la administración de la base de datos, máximo responsable del movimiento de información que se produzca en la misma. Requisitos asociados: 10, 11, 12, 13, 14
Servicios Telemáticos	Sistema encargado de los servicios telemáticos, responsable de la asignación y control de los usuarios del dominio de las personas de la UCI. Requisitos asociados: 6, 13
Aplicación de Solapines	Aplicación que controla las fotos de las personas, en la base de datos. Requisitos asociados: 7, 13
Sistema de	Aplicación para la gestión del número de pasaporte de

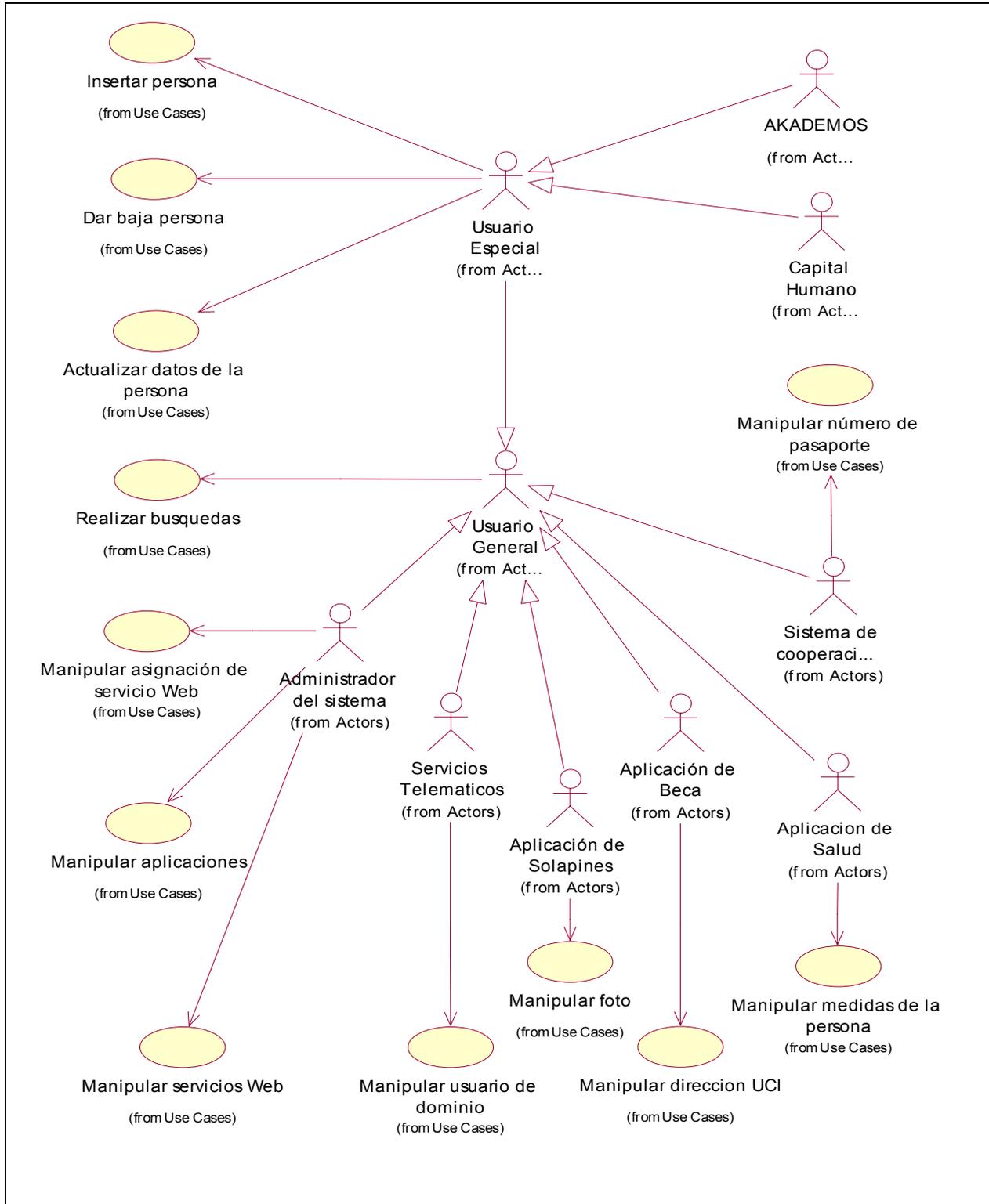
cooperación internacional	las personas. Requisitos asociados: 9, 13
Aplicación de Beca	Aplicación encargada de gestionar la dirección de las personas que viven dentro de la UCI. Requisitos asociados: 8, 13
Aplicación de Salud	Aplicación que controla las medidas de las personas, tales como la talla, el peso. Requisitos asociados: 5, 13

3.5.2.2. Casos de uso del sistema

- Caso de Uso #01. Insertar persona.
- Caso de Uso #02. Dar baja persona.
- Caso de Uso #03. Realizar búsquedas.
- Caso de Uso #04. Actualizar datos de la persona.
- Caso de Uso #05. Manipular medidas de la persona.
- Caso de Uso #06. Manipular usuario del dominio.
- Caso de Uso #07. Manipular fotos.
- Caso de Uso #08. Manipular dirección UCI.
- Caso de Uso #09. Manipular número de pasaporte.
- Caso de Uso #10. Manipular aplicaciones.
- Caso de Uso #11. Manipular servicios Web.
- Caso de Uso #12. Manipular asignación de servicio Web.

3.5.2.3. Diagramas de casos de uso del sistema

Diagrama de casos de uso



3.5.2.4. Descripción de los casos de uso

Caso de Uso #01.	Insertar una persona.
Actores:	Usuario Especial (AKADEMOS, Capital humano)
Propósito:	Permitir a las aplicaciones AKADEMOS y Capital humano, insertar una nueva persona en la base de datos.
Descripción:	
<p>El caso de uso se inicia cuando AKADEMOS o Capital Humano desean insertar una persona en la base de datos a través de un servicio Web.</p> <p>Primeramente se verifica que la aplicación esté autorizada a usar dicho servicio. Luego se busca a la persona que se desea insertar en la base de datos, si existe en el historial se copian sus datos y se actualizan, si existe como una persona activa se muestra un mensaje de error y no se realiza la operación. Se realiza la operación de inserción en caso contrario a los antes planteados.</p>	
Referencias:	R1, R13.
Precondiciones:	
Poscondiciones:	Se crea una nueva persona en la base de datos, la información queda actualizada para hacer cualquier operación sobre la misma.
Requerimientos especiales	-
Prototipos	-

Caso de Uso #02.	Dar baja persona.
Actores:	Usuario Especial (AKADEMOS, Capital humano)
Propósito:	Permitir a las aplicaciones AKADEMOS y Capital humano, dar de baja a una persona existente en la base de datos.
Descripción:	
<p>El caso de uso se inicia cuando AKADEMOS o Capital Humano desean dar de baja a una persona de la base de datos a través de un servicio Web.</p> <p>Primeramente se verifica que el usuario este autorizado a usar dicho servicio.</p>	

Luego se chequea que la persona que se desea dar de baja se encuentre en la base de datos como una persona activa y en el historial. Si la persona se encuentra en el historial solo se actualizan los datos con la nueva información, en caso contrario pasan todos sus datos al historial.	
Referencias:	R2, R13.
Precondiciones:	
Poscondiciones:	La información queda actualizada en la base de datos para futuras operaciones.
Requerimientos especiales	-
Prototipo	-

Caso de Uso #03.	Realizar búsquedas.
Actores:	Usuario General.
Propósito:	Permitir al Usuario General realizar búsquedas de información en la base de datos.
Descripción:	
<p>El caso de uso se inicia cuando el Usuario General desea hacer una búsqueda de la información de las personas en la base de datos por algunos de los criterios establecidos.</p> <p>Las búsquedas se realizarán a través varios servicios Web (uno por cada criterio) que serán los encargados de realizar la recopilación de la información para ser brindada al usuario. Se debe verificar primeramente que el usuario esté autorizado a utilizar el servicio Web solicitado.</p>	
Referencias:	R3, R13.
Precondiciones:	
Poscondiciones:	-
Requerimientos especiales	-
Prototipo	-

Caso de Uso #04.	Actualizar datos persona.
Actores:	AKADEMOS, Capital Humano
Propósito:	Permitir a las aplicaciones AKADEMOS y Capital humano, actualizar la información de las personas en la base de datos.
Descripción:	
<p>El caso de uso se inicia cuando AKADEMOS o Capital Humano desean actualizar la información que ellos manejan de alguna de las persona de la UCI de acuerdo a varios criterios.</p> <p>Las actualizaciones se realizarán a través varios servicios Web (uno por cada criterio). Se debe verificar primeramente que el usuario esté autorizado a utilizar el servicio Web solicitado, luego se chequea que la persona a la que se desea actualizar la información se encuentre en la base de datos como una persona activa.</p>	
Referencias:	R4, R13.
Precondiciones:	
Poscondiciones:	La información de la persona queda actualizada para hacer cualquier otra operación sobre la misma.
Requerimientos especiales	-
Prototipo	-

Caso de Uso #05.	Manipular medidas de la persona.
Actores:	Aplicación de Salud.
Propósito:	Permitir a la Aplicación de salud, insertar, eliminar o modificar las medidas de una persona.
Descripción:	

<p>El caso de uso se inicia cuando la Aplicación de Salud desea insertar o eliminar las medidas de una persona a través de un servicio Web. Primero se verifica la autorización de la aplicación para usar ese servicio Web y luego para cualquier tipo de operación se chequea que la persona se encuentre en la base de datos.</p>	
Referencias:	R5, R13.
Precondiciones:	
Poscondiciones:	La información de la persona queda actualizada para hacer cualquier otra operación sobre la misma.
Requerimientos especiales	-
Prototipos	-

Caso de Uso #06.	Manipular usuario de dominio.
Actores:	Servicios Telemáticos.
Propósito:	Permitir Servicio Telemáticos, insertar, eliminar o modificar el usuario de dominio de una persona.
Descripción:	
<p>El caso de uso se inicia cuando Servicios Telemáticos desea actualizar, insertar o eliminar el usuario del dominio de una persona, lo cual se realizará a través de un servicio Web (uno para cada tipo de operación). Para ello se realiza la búsqueda de la persona a la cual se le realizará la actualización o inserción para verificar su existencia en la base de datos. Se verifica además que el usuario este autorizado a usar el servicio Web.</p>	
Referencias:	R6, R13.
Precondiciones:	
Poscondiciones:	La información de la persona queda actualizada para hacer cualquier otra operación sobre la misma.
Requerimientos especiales	-

Prototipo	-
------------------	---

Caso de Uso #07.	Manipular fotos.	
Actores:	Aplicación de Solapines.	
Propósito:	Permitir a la Aplicación de Solapines, insertar, eliminar o modificar las fotos de una persona.	
Descripción:	<p>El caso de uso se inicia cuando la aplicación de solapines decide insertar, eliminar o actualizar las fotos de alguna persona, chequeando primero la existencia de la persona en la base de datos. Este proceso se hará a través de un servicio Web (para cada tipo de operación), para el cual debe de estar autorizado el usuario. Debe verificarse además que la aplicación esté autorizada a usar el servicio.</p>	
Referencias:	R7, R13.	
Precondiciones:		
Poscondiciones:	La información de la persona queda actualizada para hacer cualquier otra operación sobre la misma.	
Requerimientos especiales	-	
Prototipo	-	

Caso de Uso #08.	Manipular dirección UCI.	
Actores:	Aplicación de Beca	
Propósito:	Permitir a la Aplicación de Beca, insertar, eliminar o modificar la dirección UCI de una persona.	
Descripción:	<p>El caso de uso se inicia cuando la Aplicación de beca desea insertar o eliminar la dirección UCI de alguna persona a través de un servicio Web (uno para cada tipo de</p>	

operación). El proceso comienza verificando que el usuario este autorizado para usar el servicio y luego que la persona exista en la base de datos y sea becada. Debe tenerse en cuenta que se debe mantener un historial de las direcciones por las que ha pasado la persona.	
Referencias:	R8, R13.
Precondiciones:	
Poscondiciones:	La información de la persona queda actualizada para hacer cualquier otra operación sobre la misma.
Requerimientos especiales	-
Prototipo	

Caso de Uso #09.	Manipular número de pasaporte.
Actores:	Sistema de Cooperación Internacional
Propósito:	Permitir al Sistema de Cooperación Internacional, insertar, eliminar o modificar el número de pasaporte de una persona.
Descripción:	
<p>El caso de uso se inicia cuando el Sistema de Cooperación Internacional desea insertar, eliminar o actualizar el número de pasaporte de alguna persona. Se hará a través de un servicio Web que será el encargado de buscar a la persona a la cual se le actualizará o insertará el número de pasaporte, verificando que esta se encuentre en la base de datos. También se verifica el permiso la aplicación para hacer uso del Servicio Web.</p>	
Referencias:	R9, R13.
Precondiciones:	
Poscondiciones:	La información de la persona queda actualizada para hacer cualquier otra operación sobre la misma.
Requerimientos especiales	-

Prototipo	-
------------------	---

Caso de Uso #10.	Manipular aplicaciones.	
Actores:	Administrador del sistema	
Propósito:	Permitir al administrador, insertar, eliminar o actualizar la información de las aplicaciones.	
Descripción:	<p>El caso de uso se inicia cuando el Administrador desea insertar, eliminar o actualizar la información de las aplicaciones. Se debe verificar la existencia de la aplicación en la base de datos para cualquiera de las operaciones.</p>	
Referencias:	R10.	
Precondiciones:	El administrador debe estar previamente autenticado.	
Poscondiciones:	La información de la aplicación queda actualizada para hacer cualquier otra operación sobre la misma.	
Requerimientos especiales	-	
Prototipo	Si	

Caso de Uso #11.	Manipular servicios Web.	
Actores:	Administrador del Sistema.	
Propósito:	Permitir al Administrador del sistema, insertar, eliminar o actualizar la información de los servicios Web.	
Descripción:	<p>El caso de uso se inicia cuando el Administrador del Sistema desea insertar o eliminar los servicios Web de la base de datos. Se debe verificar la existencia del</p>	

servicio Web en la base de datos para cualquiera de las operaciones.	
Referencias:	R11.
Precondiciones:	El administrador debe estar previamente autenticado.
Poscondiciones:	La información de los servicios Web queda actualizada para hacer cualquier otra operación sobre la misma.
Requerimientos especiales	-
Prototipo	Si

Caso de Uso #12.	Asignar servicio Web.
Actores:	Administrador del Sistema.
Propósito:	Permitir al administrador del sistema, realizar, inserciones o eliminaciones en la asignación de los servicios Web a las aplicaciones.
Descripción:	
<p>El caso de uso se inicia cuando el Administrador del Sistema a pedido del jefe de una aplicación, inserta o elimina una asignación de un servicio Web, para ser usado por esta. Se debe verificar la existencia de la aplicación y el servicio Web en la base de datos para cualquiera de las operaciones.</p>	
Referencias:	R12.
Precondiciones:	El administrador debe estar previamente autenticado.
Poscondiciones:	La información de la asignación de los servicios Web, a las aplicaciones queda actualizada.
Requerimientos especiales	-
Prototipo	Si

3.6. Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución, se describieron los procesos del negocio, se definieron los requisitos funcionales y no funcionales, se representó a través de un diagrama de casos de uso las funciones del sistema y se describieron todas las acciones de los actores con los casos de uso con los que interactúa. En este momento podemos comenzar a describir el sistema cumpliendo con las funciones y requisitos definidos en este capítulo.

Capítulo 4. Construcción de la solución propuesta

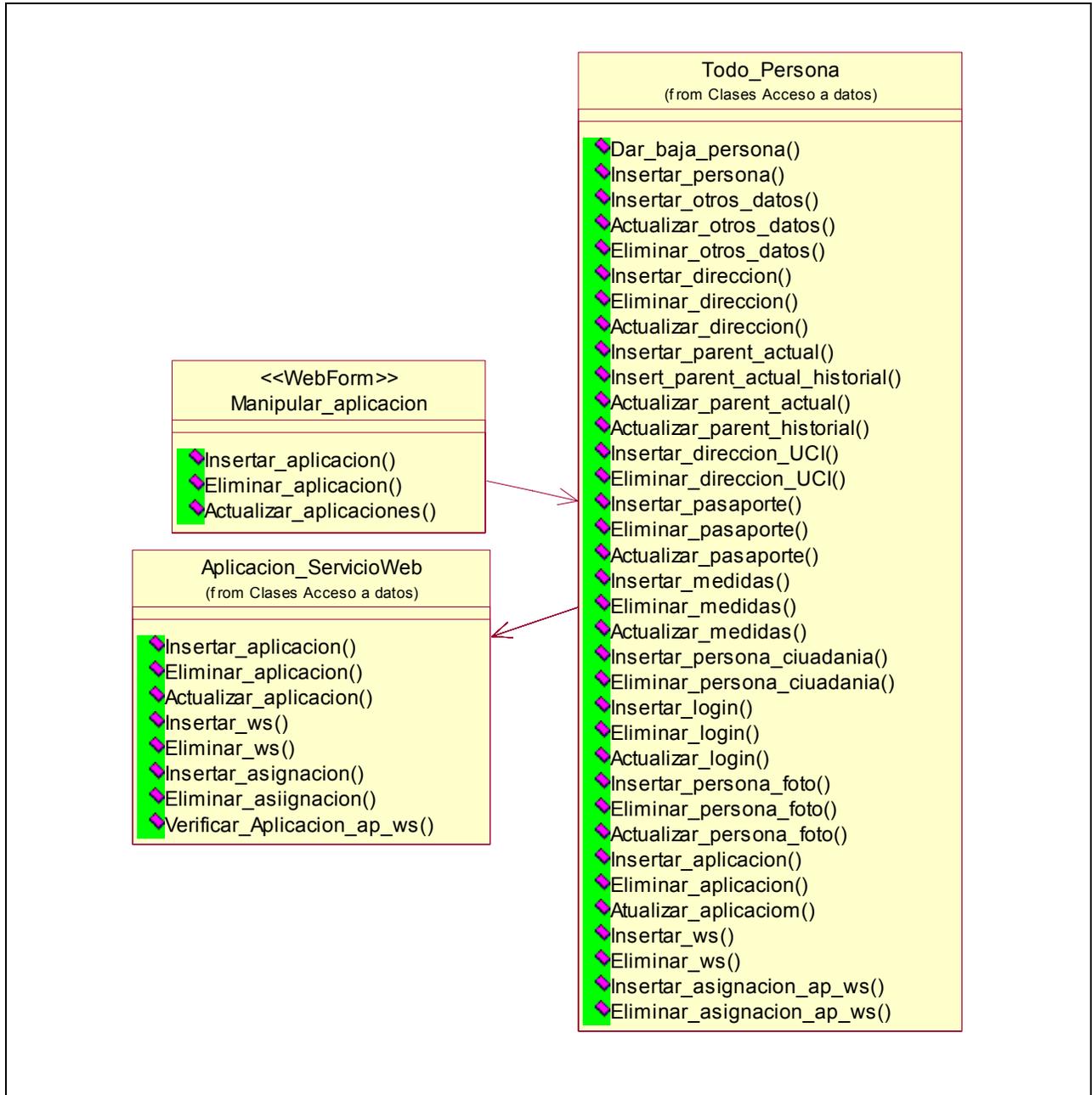
4.1. Introducción

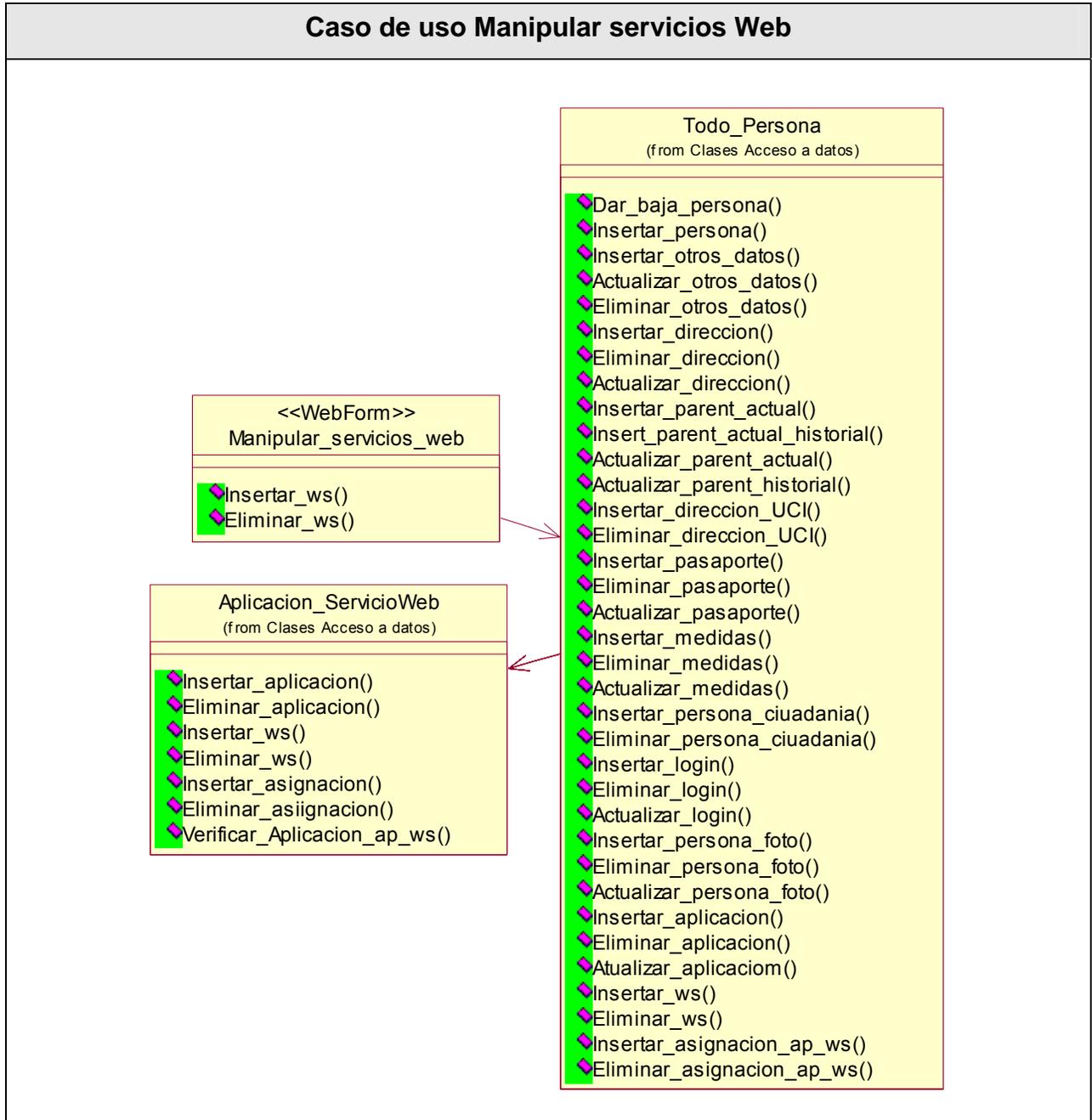
En este capítulo los componentes de la aplicación se tratan como clases y se representan las relaciones entre ellas. Se presenta el diagrama de clases persistentes y el modelo de datos, que es la base para construir finalmente la base de datos del sistema. Se tratan algunos principios de diseño, estándares de codificación a seguir y por último se presentan los diagramas de despliegue e implementación, donde se definen los componentes que forman parte de la aplicación y los nodos que forman la topología de hardware, que soportará al sistema.

4.2. Diagrama de clases

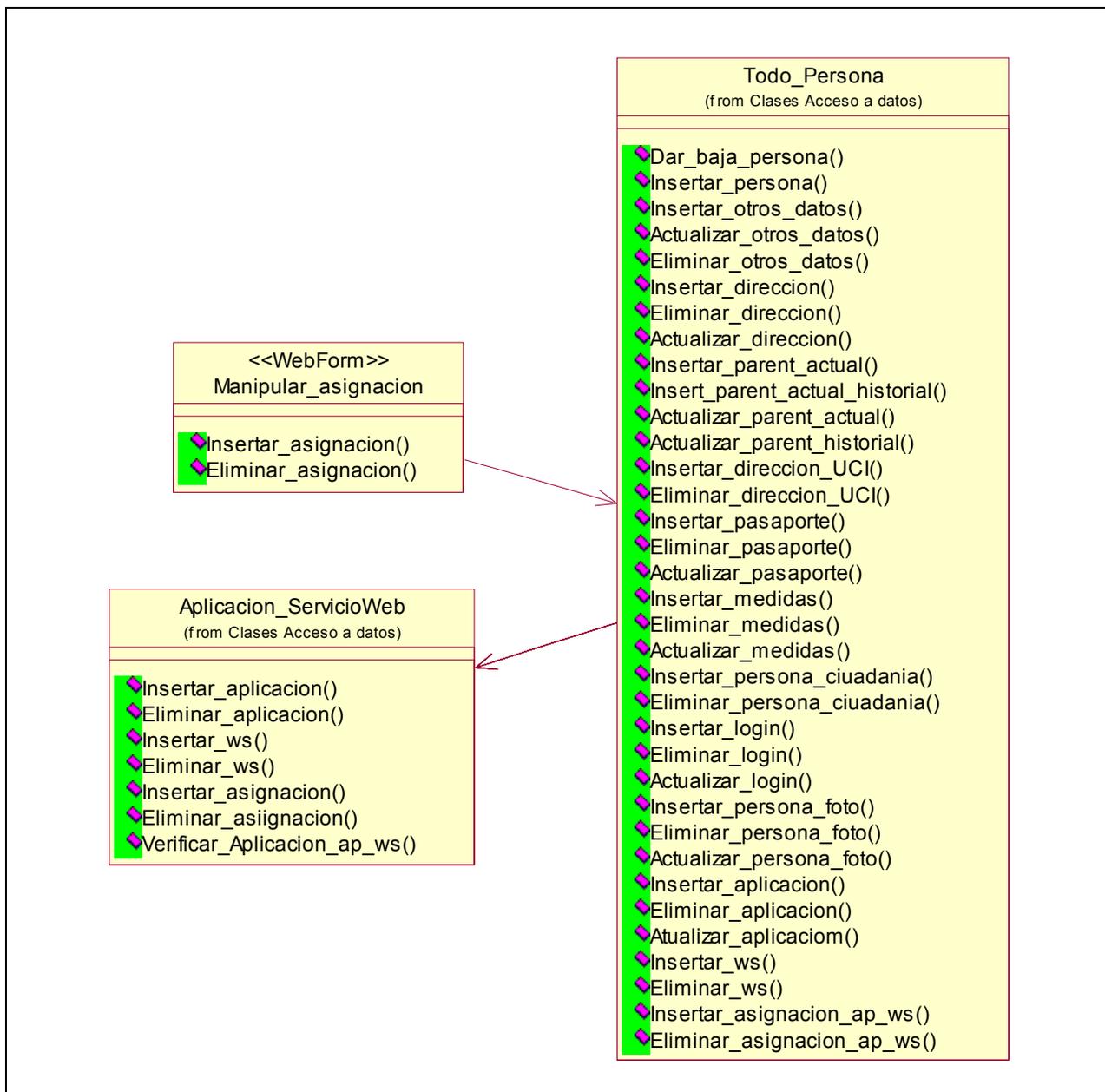
Los diagramas de clases se han confeccionado por casos de uso, para una mejor comprensión de los mismos se han separado los relacionados con las actividades del administrador y los relacionados con los servicios Web.

A continuación se presentan los diagramas de clases relacionados con la administración para:





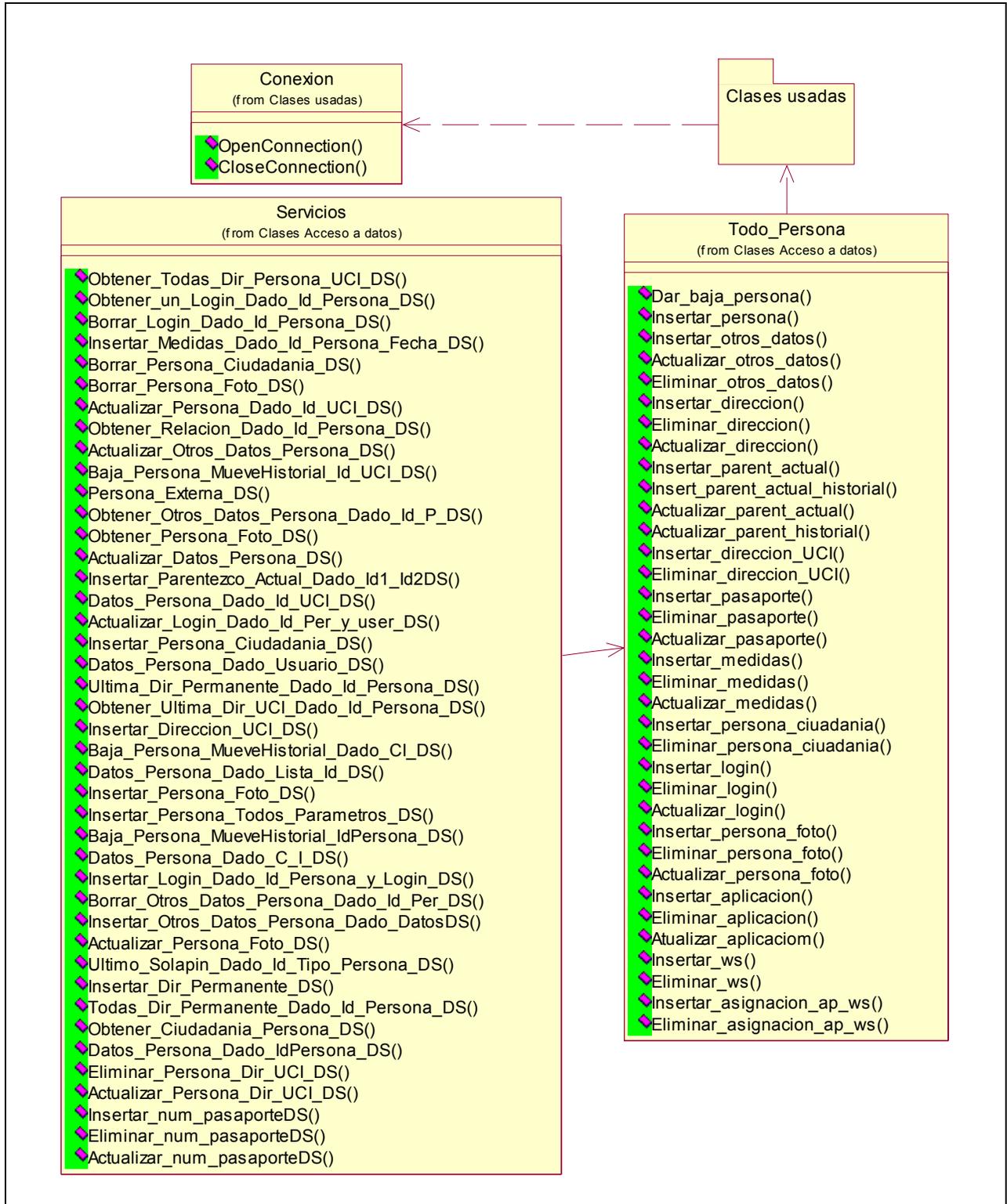
Caso de uso Manipular asignación



Diagramas de clases para casos de uso relacionados con los servicios Web

Se representa un diagrama general, que consta de un paquete que contiene las clases del que serán usadas para cada caso de uso.

Diagrama general

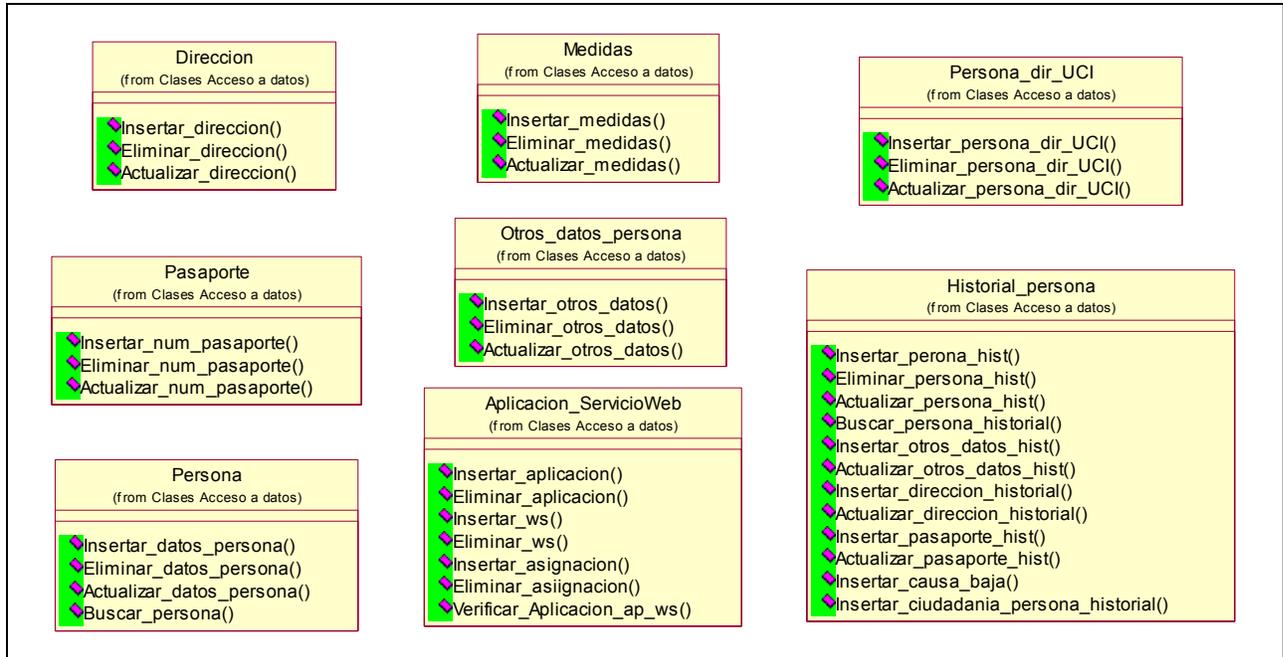


Paquete *Clase usadas* para:

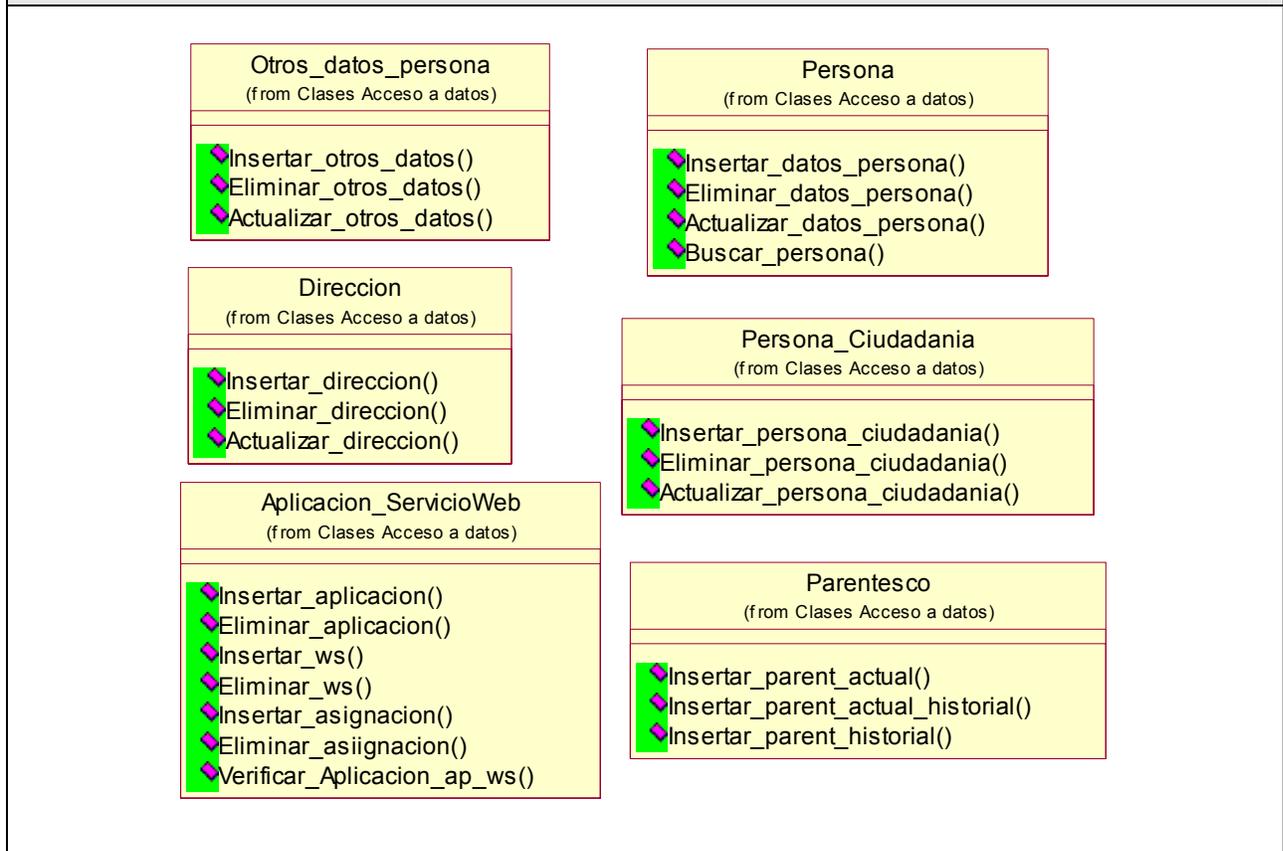
Caso de uso Insertar persona

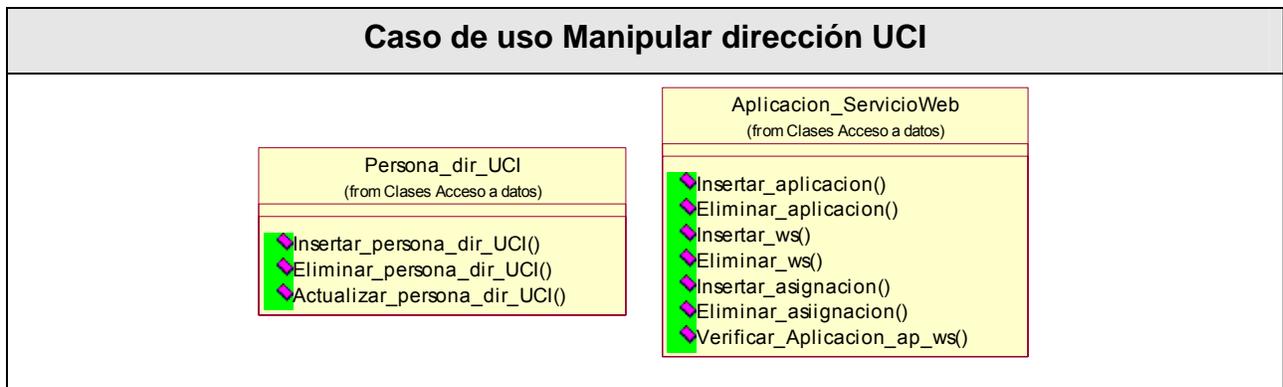
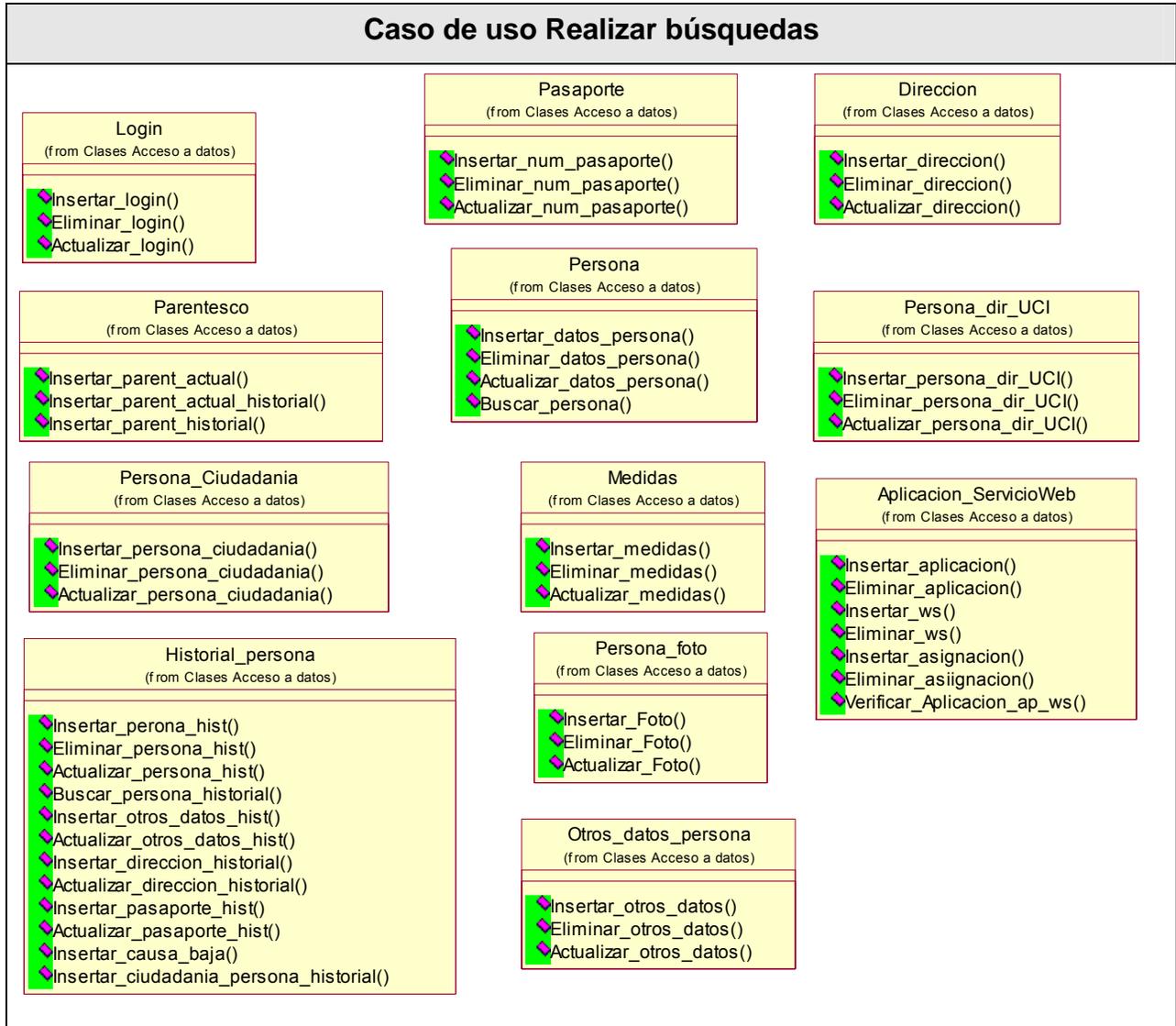


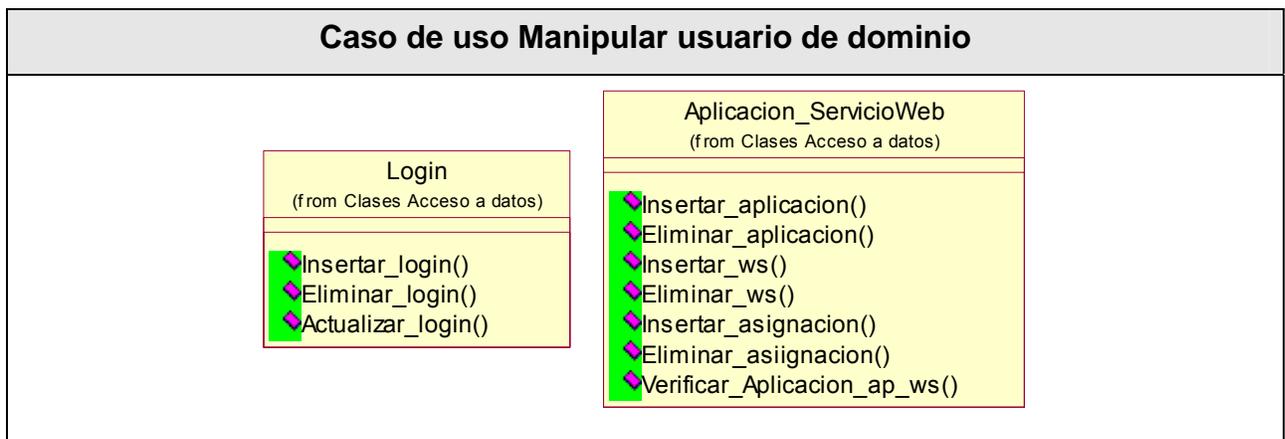
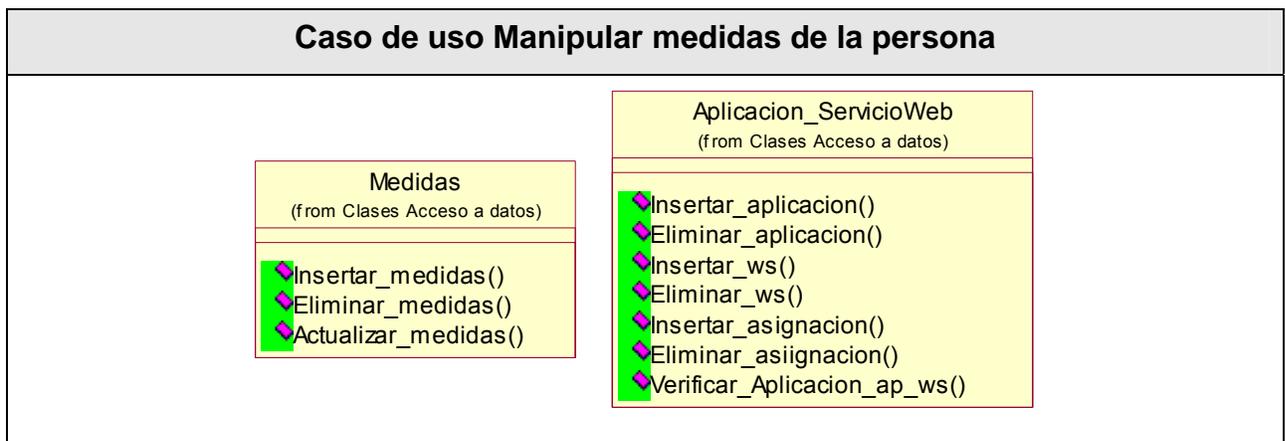
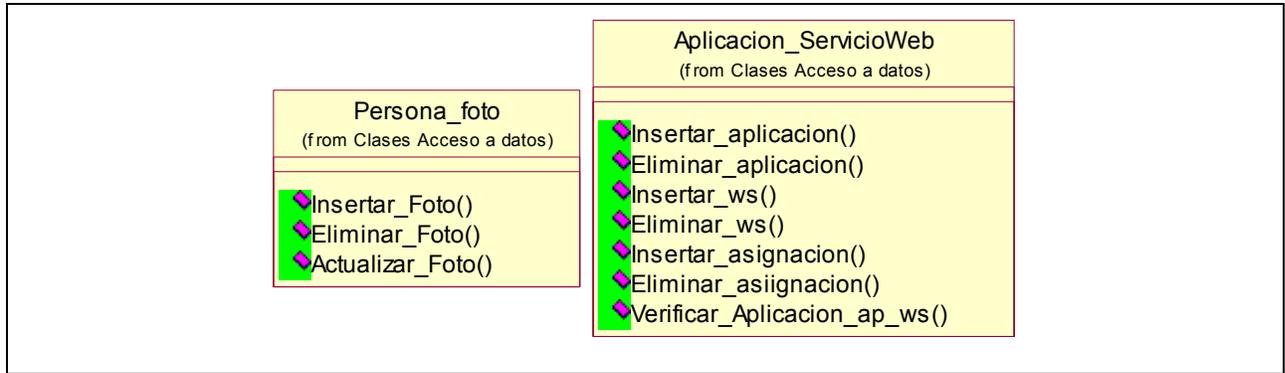
Caso de uso Dar baja persona

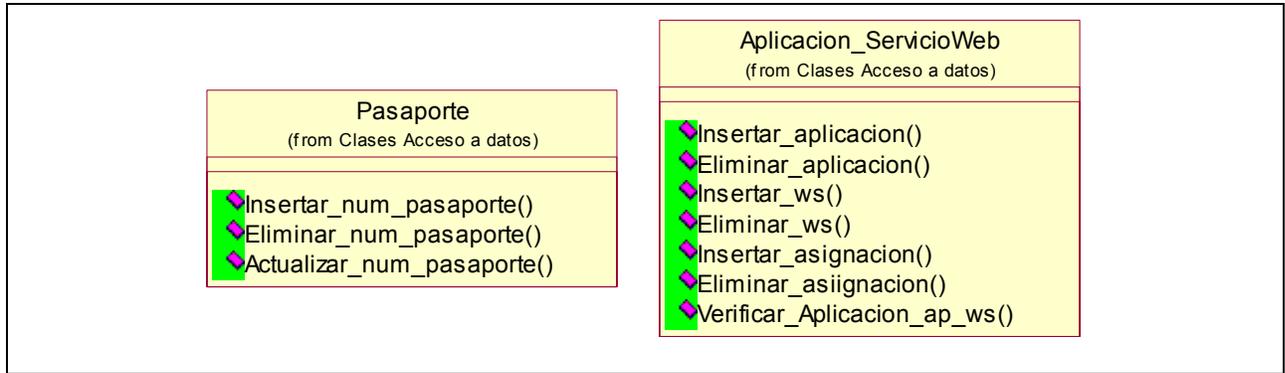


Caso de uso Actualizar datos de la persona





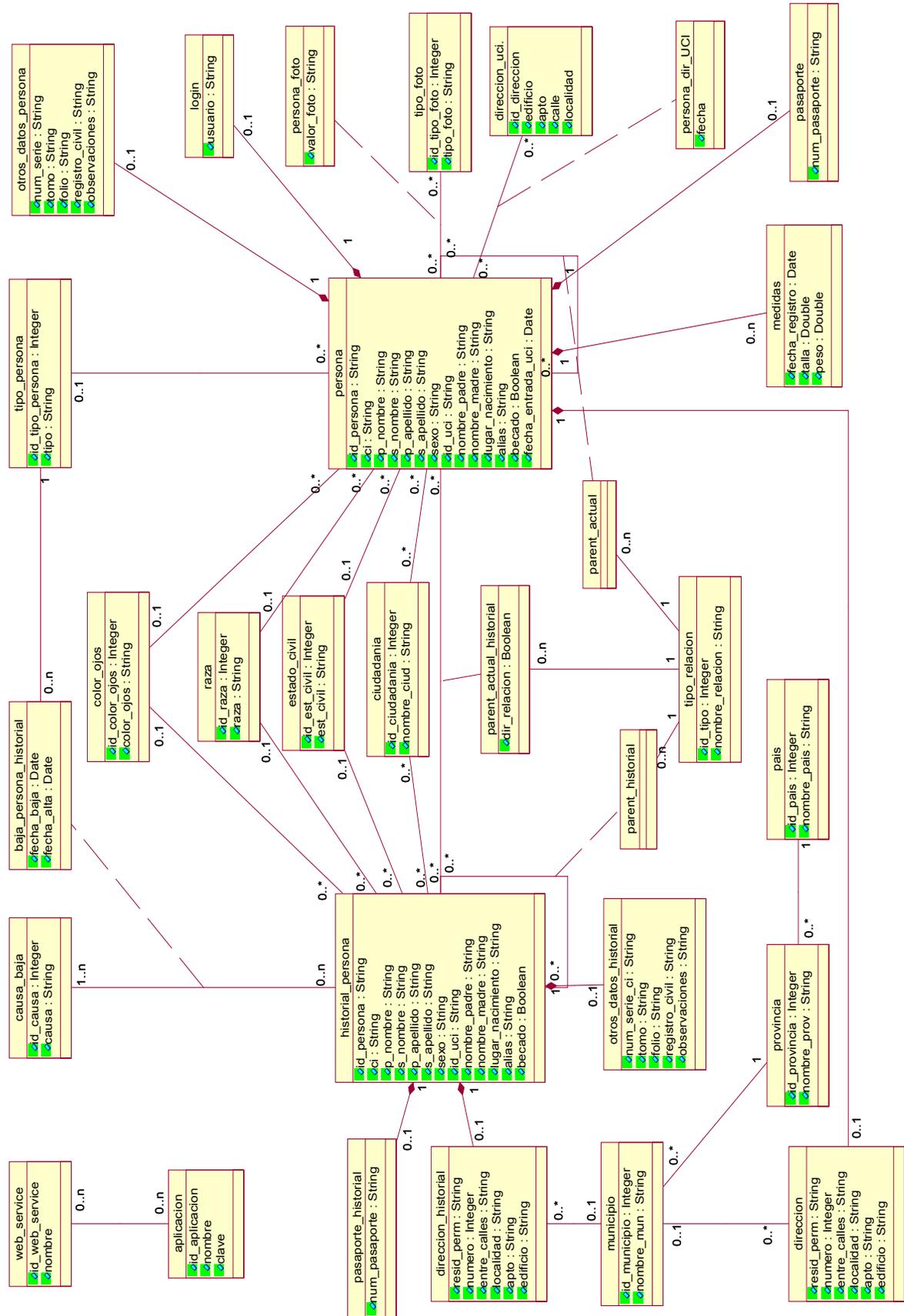




4.3. Diseño de la base de datos

Para la construcción de la base de datos se utiliza el diagrama de clases persistentes, a partir de cual se obtiene el modelo de datos, que corresponde con la representación física de la base de datos del sistema.

Diagrama de clases persistentes



4.4. Principios de diseño

4.4.1. Estándares en la interfaz de la aplicación

Las aplicaciones solo interactúan con el sistema a través de servicios Web. El administrador es al único que se le presenta una interfaz muy sencilla para el control de las aplicaciones y servicios Web que acceden a la aplicación. Se deben utilizar colores que permitan la visibilidad de los textos, y que sean agradables a la vista.

4.4.2. Concepción general de la ayuda

Consta de una documentación en línea necesaria para facilitarle a los desarrolladores el manejo de cada servicio Web, para que puedan sin dificultad utilizarlos en sus aplicaciones obteniendo el mejor provecho de los mismos. Se presenta información como: descripción, parámetros de entrada, parámetros de salida, forma de conexión y ejemplos de su uso.

4.4.3. Tratamiento de excepciones

Se chequea el éxito de cada operación de inserción, eliminación o actualización realizada sobre los datos.

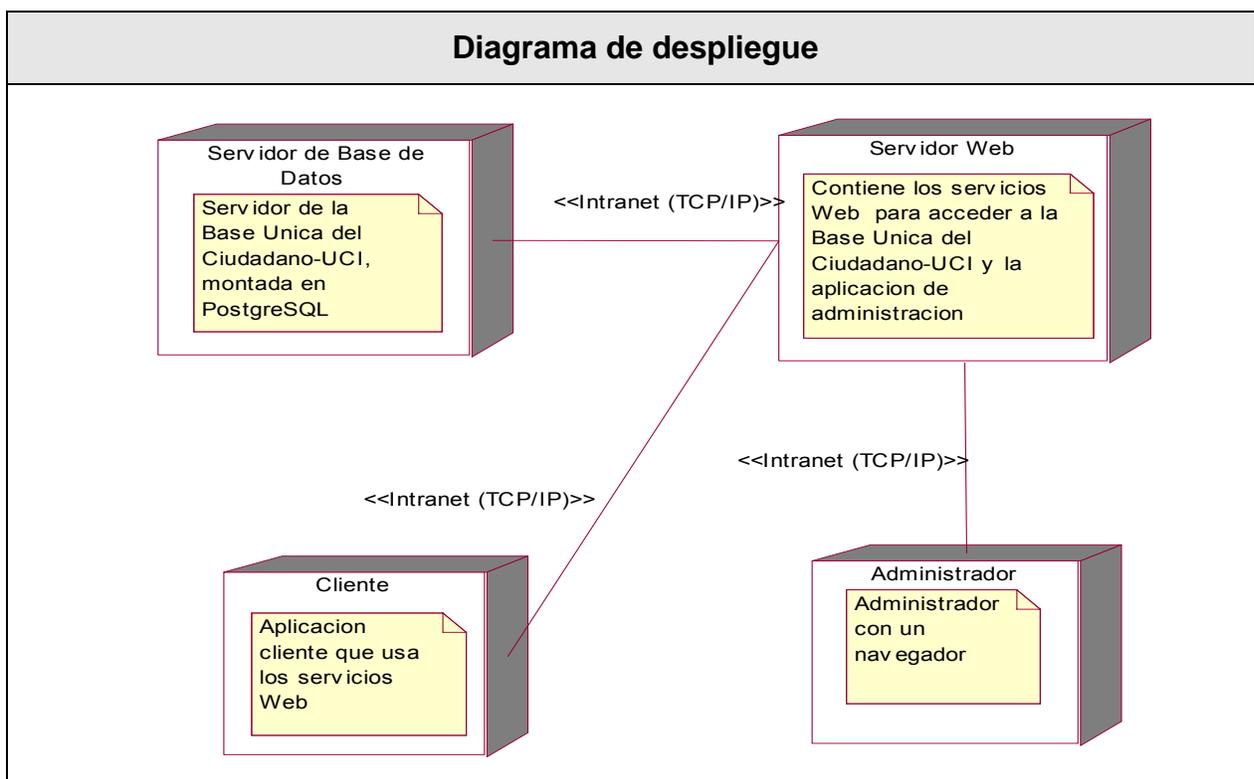
4.5. Estándares de codificación

En la construcción de la base de datos, todas las tablas comienzan con el prefijo `tbl_` y las funciones con `sp_` manteniendo los mismos nombres (tanto de funciones y tablas como de atributos) que en la anterior implementación en SQL Server. Estos nombres fueron cambiados todos a minúsculas, para evitar confusiones en el PostgreSQL. Los nombres de los servicios Web también se mantuvieron invariables para que las aplicaciones no tengan que hacer muchos cambios al reemplazar los servicios existentes por los nuevos.

Mantener una codificación estándar es necesario para reducir posibles errores y obtener un código comprensible, lo que facilita la comunicación entre los programadores y en un futuro el mantenimiento del software.

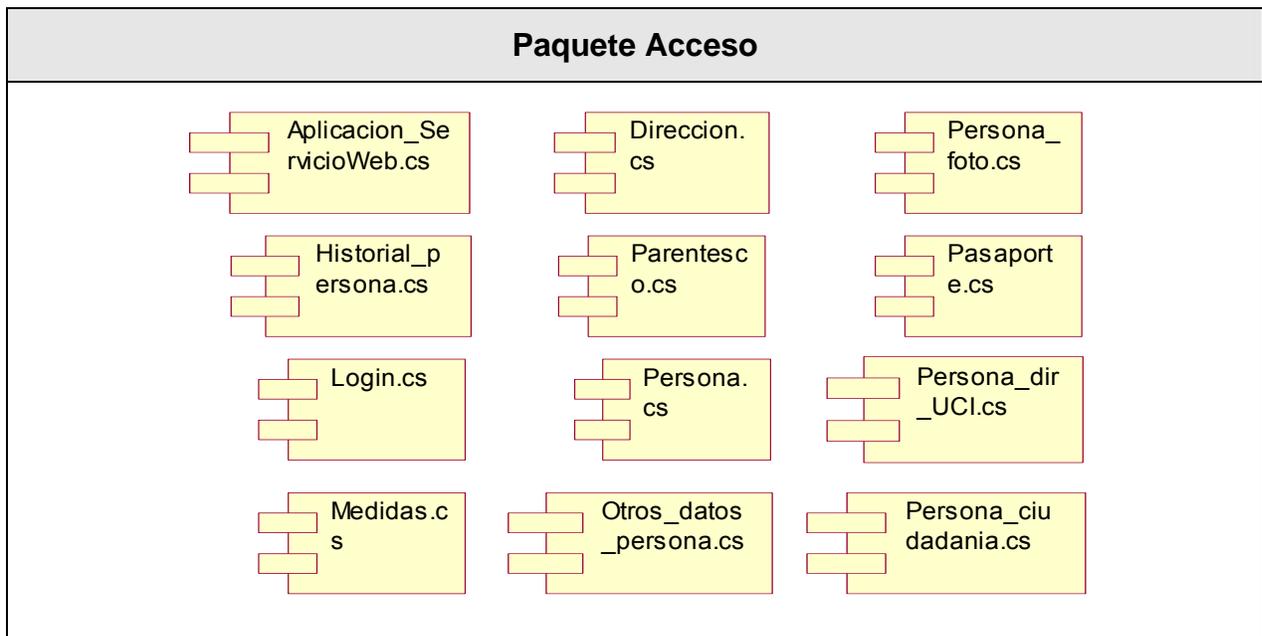
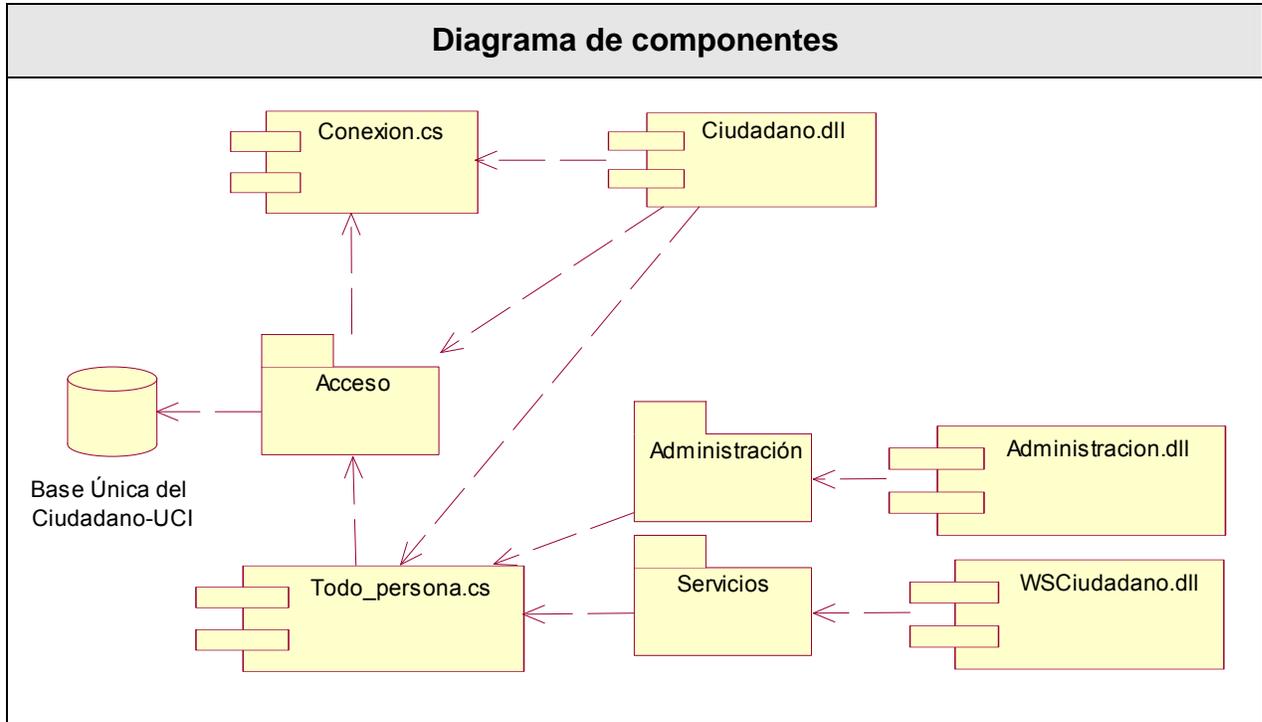
4.6. Modelo de despliegue

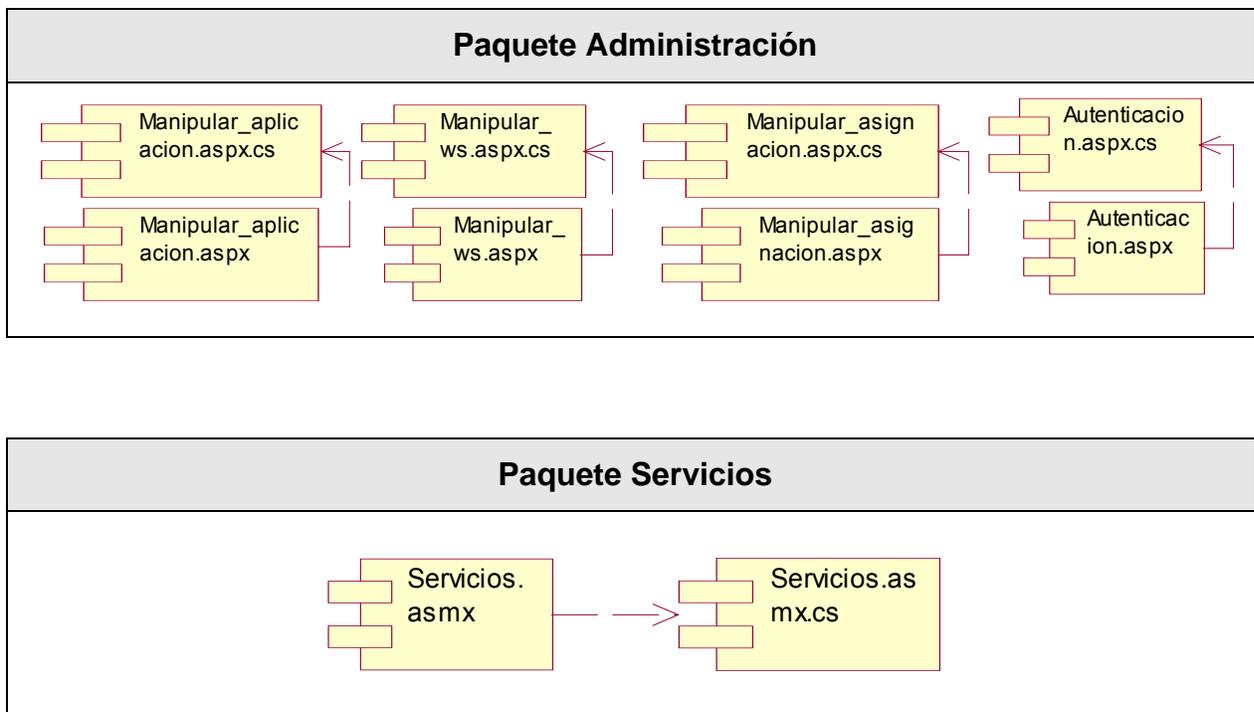
El diagrama de despliegue muestra la configuración de los nodos, representando como se encuentran relacionados físicamente los componentes de la aplicación. Está formado por un servidor donde se encuentra la base de datos en PostgreSQL 8.0 y otro donde se encuentran los servicios Web, el cual debe tener instalada la plataforma Mono.



4.7. Modelo de implementación

El modelo de implementación se construye en la fase de implementación y en ellos se representan los componentes, su organización y dependencia entre ellos.





Explicación de los componentes

En el diagrama de componentes están representados todos los ficheros del sistema, pero para una mejor comprensión, se han agrupado en diferentes paquetes, ficheros que presentan comportamientos similares.

- Paquete Acceso

Formado por los componentes donde se encuentran las implementaciones de las clases que se ocupan de la persistencia de los datos. Estas clases (una por cada fichero) tienen el mismo nombre que los componentes donde se encuentran.

- Paquete Servicios

Formado por los componentes donde se encuentra la implementación de la clase que contiene los servicios Web.

- Paquete Administración

Formado por los componentes donde se encuentran las implementaciones de las clases relacionadas con las actividades del administrador del sistema

(autenticación y asignación de los servicios Web a las aplicaciones). Se representan los componentes aspx que tienen asociado otro con el mismo nombre y de extensión cs donde se implementan las clases.

- Ciudadano.dll

Librería necesaria para las clases de acceso a datos (Conexion, Aplicación_ServicioWeb, Direccion, Persona_Foto, Historial_persona, Login, Medidas, Parentesco, Persona, Otros_datos_persona, Pasaporte, Persona_dir_UCI, Persona_ciudadania) y lógica del negocio (Todo_persona).

- Administracion.dll

Librería necesaria para las clases relacionadas con las actividades de administración (Manipular_ws, Manipular_aplicacion, Manipular_asignacion y Autenticacion)

- WSCiudadano.dll

Librería necesaria para la clase que implementa los servicios Web (Servicios).

- Conexion.cs

Contiene la clase Conexión con los métodos necesarios para que sea utilizada por el resto de las clases de la capa de acceso a datos (incluidas en el paquete Acceso) para conectarse a la base de datos.

- Todo_persona

Componente donde se encuentra la implementación de la clase Todo_persona que sirve para modelar el negocio.

4.8. Conclusiones

En este capítulo se tienen todos los elementos necesarios para la implementación del software. Se describen las clases y sus relaciones mediante los diagramas de clase, se define el diagrama de clases persistentes y el modelo de datos, para la construcción de la base de datos y por último se determinan los nodos y

componentes mediante los diagramas de despliegue e implementación para poder mostrar como queda estructurada la aplicación físicamente.

5.1. Introducción

Antes de comenzar un proyecto de software y que este resulte lo esperado es necesario evaluar si es factible o no su confección. Aunque estas estimaciones no son exactas, nos dan una idea del esfuerzo a emplear, tiempo de trabajo, recursos requeridos, y cantidad de líneas de código. Esto evita gastos de recursos innecesarios en caso de que no pueda dar continuidad a un proyecto en determinado momento, por falta de recursos, de tiempo, entre otras.

Para realizar estas estimaciones la Ingeniería de Software incluye un grupo de herramientas y técnicas entre las que se incluye COCOMO (*CO*nstructive *CO*st *MO*del), utilizada en este capítulo para realizar el estudio de factibilidad del sistema.

Este capítulo incluye un análisis de costos y de beneficios tangibles e intangibles del proyecto.

5.2. Planificación

Entradas Externas

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Insertar persona	4	35	Compleja
Dar baja persona	18	59	Compleja
Actualizar datos identificativos	1	7	Simple
Actualizar ciudadanía	1	1	Simple
Actualizar relaciones de parentesco	1	2	Simple
Actualizar datos del CI	1	2	Simple
Actualizar dirección permanente	1	7	Simple
Insertar medidas de la persona	1	3	Simple
Actualizar medidas de las	1	3	Simple

personas			
Eliminar medidas de la persona	1	3	Simple
Insertar usuario de dominio	1	1	Simple
Actualizar usuario de dominio	1	1	Simple
Eliminar usuario de dominio	1	1	Simple
Insertar foto	1	1	Simple
Actualizar foto	1	1	Simple
Eliminar foto	1	1	Simple
Insertar dirección UCI	2	7	Media
Actualizar dirección UCI	2	7	Media
Eliminar dirección UCI	1	3	Simple
Insertar número de pasaporte	1	1	Simple
Actualizar número de pasaporte	1	1	Simple
Eliminar número de pasaporte	1	1	Simple
Insertar aplicaciones	1	3	Simple
Eliminar aplicaciones	1	3	Simple
Actualizar aplicaciones	1	3	Simple
Insertar servicios Web	1	2	Simple
Eliminar servicios Web	1	2	Simple
Insertar asignación de un servicio Web a una aplicación	3	4	Simple
Eliminar asignación de un servicio Web a una aplicación	3	2	Media

Salidas Externas

Nombre de la salida externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Listado de las personas becadas	1	4	Simple

Peticiones

Nombre de la petición	Cantidad de	Cantidad de	Clasificación
-----------------------	-------------	-------------	---------------

	ficheros	Elementos de datos	(Simple, Media y compleja)
Mostrar listado de todos los datos de una persona	18	40	Compleja
Mostrar listado de las personas de un mismo tipo	1	2	Simple
Mostrar relación de parentesco que existe entre dos personas	3	5	Simple
Mostrar usuario del dominio de una persona	1	2	Simple
Mostrar direcciones en la UCI donde ha vivido una persona	2	7	Media
Mostrar listado de los datos del CI de una persona	1	6	Simple
Mostrar dirección permanente de una persona	1	8	Simple
Mostrar ciudadanía de una persona	2	4	Simple
Mostrar medidas de la persona	1	4	Simple
Mostrar Id_UCI de una persona	1	2	Simple
Mostrar todas las relaciones de parentesco de una persona	3	7	Media
Mostrar fotos de una persona	1	3	Simple
Mostrar resultado de una búsqueda fonética del nombre de una persona	1	4	Simple

Ficheros internos

Nombre del fichero interno	Cantid	Cantidad	Clasificación
-----------------------------------	---------------	-----------------	----------------------

	ad de record s	de Elementos de datos	(Simple, Media y compleja)
tbl_aplicacion_ws	1	2	Simple
tbl_aplicaciones	1	3	Simple
tbl_baja_persona_historial	1	5	Simple
tbl_causa_baja	1	2	Simple
tbl_ciudadania	1	2	Simple
tbl_color_ojos	1	2	Simple
tbl_direccion	1	8	Simple
tbl_direccion_historial	1	8	Simple
tbl_direccion_uci	1	5	Simple
tbl_estado_civil	1	2	Simple
tbl_historial_persona	1	16	Simple
tbl_login	1	2	Simple
tbl_medidas	1	4	Simple
tbl_municipio	1	3	Simple
tbl_otros_datos_historial	1	6	Simple
tbl_otros_datos_persona	1	6	Simple
tbl_pais	1	2	Simple
tbl_parent_actual	1	3	Simple
tbl_parent_actual_historial	1	4	Simple
tbl_parent_historial	1	3	Simple
tbl_pasaporte	1	2	Simple
tbl_pasaporte_historial	1	2	Simple
tbl_persona	1	18	Simple
tbl_persona_ciudadania	1	2	Simple
tbl_persona_ciudadania_historial	1	2	Simple
tbl_persona_direccion_uci	1	3	Simple
tbl_persona_foto	1	3	Simple
tbl_provincia	1	3	Simple
tbl_raza	1	2	Simple

tbl_sonidos	1	2	Simple
tbl_tipo_foto	1	2	Simple
tbl_tipo_persona	1	2	Simple
tbl_tipo_relacion	1	2	Simple
tbl_web_service	1	2	Simple

Puntos de Función desajustados

Elementos	Simple		Medio		Complejo		Subtotal
	No	Peso	No	Peso	No	Peso	
Ficheros lógicos internos	34	7	0	10	0	15	238
Ficheros de interfaces externas							
Entradas externas	24	3	3	4	2	6	100
Salida externa	1	4	0	5	0	7	4
Peticiones	10	3	2	4	1	6	44
Total							386

Cálculo de la cantidad de instrucciones

Ratio según lenguajes [16]

59 para C# (85%)

39 para SQL (15%)

Se elige el lenguaje SQL, porque para la implementación de las funciones en PostgreSQL se utilizan el SQL y el PLPGSQL, este último basado en SQL. Se elije C# porque se está utilizando la plataforma Mono que es la versión libre del.NET.

SLOC = UFP * ratio.

Para C#

$$\text{SLOC} = 328.1 * 59$$

$$\text{SLOC} \approx 19357.9$$

$$\text{KSLOC} \approx 19.3579$$

Para SQL

SLOC = $57.9 * 39$
 SLOC ≈ 2258.1
 KSLOC ≈ 2.2581
 Total
 SLOC ≈ 21616
 KSLOC ≈ 21.616

Características	Valor
Puntos de función desajustados	386
Lenguaje	39 sql (15%) 59 C# (85%)
Instrucciones fuentes por puntos de función (miles de instrucciones)	2.2581 19.357921.616
Instrucciones fuentes (miles de instrucciones)	21.616

5.3. Costos.

Cálculo de:	Valor	Justificación
Esfuerzo	PM = 70.5	<p>A= 2.94, B = 0.91, Size = KSLOC ≈ 21.6</p> <p>SFj : Factores de Escala</p> <p>PREC: Precedencia. = 4.96</p> <p>FLEX: Flexibilidad. = 3.04</p> <p>RESL: Riesgos. = 2.83</p> <p>TEAM: Cohesión del Equipo. = 2.19</p> <p>PMAT: Madurez de las Capacidades. = 7.80</p> <p>$\Sigma SFj = 20.82$</p> <p>$E = 0.91 + 0.01 * 20.82 = 1,1182$</p> <p>RCPX = 1.00</p>

		<p>RUSE = 1.07</p> <p>PDIF = 1.00</p> <p>PERS = 0.83</p> <p>PREX = 0.87</p> <p>FCIL = 1.00</p> <p>SCED = 1.00</p> <p>PEM = 0.772647</p> <p>PM = 2.94 * 21.6^{1.1182} * 0.772647 = 70.54</p>
Tiempo de desarrollo	TEDV = 14.43	<p>B = 0.91, C = 3.67, D = 0.28</p> <p>TDEV = 3.67 (70.5)^{0.32164} = 14.43</p> <p>F = 0.28 + 0.2 * 0.01 * 20.82 = 0,32164</p>
Cantidad de personas	5	<p>PM = 70.5</p> <p>TEDV ≈ 14.43</p> <p>PM/TEDV = 4.88</p>
Costo	\$ 81075	<p>Salario medio de una persona = 230.</p> <p>CHM = 5 * 230 = 1150</p> <p>C = CHM * PM = 1150 * 70.5 = 81075</p>
Salario medio	\$ 230	
RCPX	1.0	<p>RELY = NOMINAL</p> <p>DOCU = NOMINAL</p> <p>CPLX = NOMINAL</p> <p>DATA = Moderado</p>
RUSE	1.07	La aplicación da la posibilidad de ser reutilizada
PDIF	1.0	<p>TIME = 50%</p> <p>STOR = 50%</p> <p>PVOL = estable</p>
PREX	0.87	APEX = ALTO

		PLEX= ALTO LTEX = ALTO
FCIL	1.0	TOOL = NOMINAL SITE = NOMINAL
SCED	1.0	SCED = 100 %
PERS	0.83	ACAP = ALTO PCAP = ALTO PCON = ALTO

5.4. Beneficios tangibles e intangibles.

El almacenamiento de la información de las personas de la UCI, tiene un carácter centralizado y a través de una librería de servicios Web acceden a la misma las aplicaciones que la necesitan para realizar sus funciones. Esto trae muchos beneficios entre los cuales se encuentran:

- Fácil acceso a información actualizada.
- Gestión de la información de manera centralizada.
- Se evita en un alto por ciento la duplicidad e inconsistencia.
- Pueden acceder al mismo tiempo varias aplicaciones a la base de datos para obtener la misma información.
- Los datos que maneja son de interés para la mayoría de las aplicaciones de la UCI.

5.5. Análisis de costos y beneficios.

El desarrollo de este sistema es factible ya que no conlleva grandes gastos de recursos, pero si de tiempo, y tener que confeccionar una base de datos en un gestor que comienza a abrirse paso en la universidad, donde no existen muchos programadores con un dominio del mismo, como existe por ejemplo con el SQL. Es un sistema portable por lo que un cambio de plataforma para la implementación del mismo

no traería muchos inconvenientes, por la estructuración de capas con que ha sido diseñado.

5.6. Conclusiones.

En este capítulo se realizó el estudio de factibilidad correspondiente al desarrollo del proyecto. Se obtuvo un costo de \$81075 para 14 meses de trabajo, con un equipo de 5 personas. Es factible implementarlo, dada su importancia. En el desarrollo del proyecto intervienen dos personas lo que puede influir en el tiempo estimado para la implementación del mismo.

Conclusiones

En la Universidad de las Ciencias Informáticas se está llevando a cabo la política de migración de todas las aplicaciones al software libre. La Base Única del Ciudadano-UCI como parte de esta política y con la realización de este trabajo, quedó construida sobre el gestor de base de datos libre, PostgreSQL, donde están implementadas las funciones que se encargan de insertar, eliminar, actualizar y buscar la información. Para acceder a los datos se tiene una librería de servicios que se está siendo implementada sobre la plataforma libre de.NET, Mono. Con la utilización de estas herramientas se está teniendo en cuenta la política del país y de la UCI en lo referido al empleo del software libre.

Unido a la migración a la base *Ciudadano*, se le realizó un nuevo diseño el con el cual se soluciona en gran medida el problema de la duplicidad e inconsistencia de la información, lo que es de vital importancia para lograr la rapidez que se necesita para acceder a los datos.

La nueva Base Única del Ciudadano-UCI y su librería de servicios, no se encuentran en funcionamiento, porque además de que esta última no está concluida, se necesita que las aplicaciones que usan los servicios incluyan en la llamada a los mismos su nombre y contraseña para poder acceder a ellos.

Considerando la política migratoria hacia el software libre, conlleva a pensar en este proceso de manera paulatina extenderlo a toda la universidad, por lo que se aspira que a partir de este momento, comience la migración del resto de las aplicaciones, para que muy pronto esté la mayor parte de la UCI utilizando el software libre.

En el país se está llevando a cabo el proceso de informatización de la sociedad cubana, el cual incluye la construcción de una base de datos centralizada que contará con la información distintiva de todos los ciudadanos del país. Esta Base Única del Ciudadano contará con un “Servicio de Certificación de Identidades” para que cada empresa u organismo del país pueda acceder a la misma y obtener la información necesaria para el control de su capital humano.

En este proceso la Base Única del Ciudadano-UCI debe cambiar en algunos aspectos para integrarse al grupo de las aplicaciones que harán uso del Servicio de Certificación de Identidades. Uno de estos aspectos es la llave única del ciudadano, que según lo planificado por las entidades encargadas del diseño de la misma, diferirá con la que se tiene en la UCI. Por tanto se recomienda para cuando la Base Única del Ciudadano quede construida, estudiar y rediseñar la Base Única del Ciudadano-UCI, para no entrar en contradicción con la misma y poder aprovechar al máximo los servicios que pueda brindar.

Para un futuro inmediato se recomienda continuar con la implementación de los servicios Web que han quedado pendientes, e incorporar otros que puedan serle útiles a las aplicaciones en sus funciones. Además, terminar la implementación del acceso del administrador a la base de datos.

Otra importante recomendación es continuar con el proceso de migración del resto de las aplicaciones de la universidad al software libre, aprovechando las bondades que este brinda.

Referencias bibliográficas

- [1] Las bases de datos.
<http://www.monografias.com/trabajos14/basededatos/basededatos.shtml#def>
(12/805/2005)
- [2] Trejo, Janhil. Base de datos.
<http://www.monografias.com/trabajos11/basda/basda.shtml> (2/05/2005)
- [3] Ramírez, Mercedes. Bases de datos.
<http://www.inf.udec.cl/~basedato/apunte/capitulo1/capitulo1.html> (12/052005)
- [4] Software libre
<http://www.gnu.org/home.es.html> (22/05/2005)
- [5] Stallman, Richard. Software libre para una sociedad libre
<http://www.sindominio.net/biblioweb/pensamiento/softlibre/index.html>
(20/05/2005)
- [6] Manifiesto del Software Libre
<http://manifiesto.cofradia.org> (21/05/2005)
- [7] El software libre y la administración pública
<http://curso-sobre.berlios.de/curso/trab/palacios/ProgramasLibresPolitica.doc>
(20/05/2005)
- [8] del Valle, Amauri. Impulsará nuestro país sistema operativo Linux
<http://www.jrebelde.cubaweb.cu/2005/abril-junio/mayo-17/impulsara.html> (10/06/2005)
- [9] Informatización de la Sociedad
<http://www.mic.gov.cu/hinfosoc.asp> (30/05/2005)
- [10] Jacobson, Booch, Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. Madrid, España 2000
- [11] Que es.NET
<http://www.microsoft.com/latam/net/introduccion/quees.asp> (16/05/2005)
- [12] Mono.plataforma libre del.NET
<http://www.monohispano.org> (15/05/2005)
- [13] Mas, Jordi. Mono: mucho mas que una implementacion libre de.net
<http://www.softcatala.org/~jmas/articles/articulo-mono.pdf> (10/05/2005)

- [14] Web services con.NET
<http://www.desarrolloweb.com> (10/05/2005)
- [15] Calderón, Maidely. *Servicios Básicos del Ciudadano*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2004.
- [16] Instalación de un servidor PostgreSQL bajo Linux
<http://www.webexperto.com> (15/05/2005)

1. <http://www.hispalinux.es/informes/wheeler/> (10/005/2005)
2. Coletti, Daniel. ¿Por qué Software Libre?
http://www.aat-ar.org/Revista_art.asp?iid=186 (19/05/2005)
3. UML. Análisis, diseño e implementación de aplicaciones
<http://www.vico.org> (15/05/2005)
4. Que es MySQL
<http://www.software-shop.com/Productos/MySQL/mysql.html> (19/05/2005)
5. Software Linux: Programas y Aplicaciones de Linux
<http://linux.bankhacker.com/software/MySQL/> (21/05/2005)
6. MySQL
<http://es.wikipedia.org/wiki/MySQL> (10/05/2005)
7. Blanco, Sergio. Mono la plataforma libre de.NET
<http://www.marblestation.com/publicaciones/paper-mono.pdf> (13/05/2005)
8. Programación:C sharp NET
[http://es.wikibooks.org/wiki/Programaci%C3%B3n:C_sharp_NET / Cap%C3%A1tulo_1](http://es.wikibooks.org/wiki/Programaci%C3%B3n:C_sharp_NET/_Cap%C3%A1tulo_1) (10/05/2005)
9. Web Services. XML-RPC, SOAP, sobre PHP, Perl, y otros conceptos
<http://web-services.bankhacker.com/> (14/05/2005)
10. Lamadrid, Victor Creando Servicios Windows en.NET - (Parte 2/2)
<http://www.informatizate.net> (14/05/2005)
11. Que es, formas y tipos de licenciamiento de software.
<http://www.wilkinsonpc.com.co/free/articulos/softwarelibre.html> (22/05/2005)
12. ¿Qué es GNU/Linux?
<http://microlug.linux.net.uy/gnu/gnu.htm> (21/05/2005)

13. Stallman, Richard. Why "Free Software" is better than "Open Source"
<http://www.gnu.org/philosophy/=open-source-or-free.html> (22/05/2005)
14. Categorías de Software Libre y No Libre
<http://www.gnu.org/philosophy/categories.es.html#TOCOpenSource> (24/05/2005)
15. Algorta, Diego. OpenGenerator
<http://opengenerator.codigolibre.net/docs/articulo/html/x56.html> (18/05/2005)
16. El software libre y la administración pública
<http://curso-sobre.berlios.de/curso/trab/palacios/ProgramasLibresPolitica.doc>
(20/05/2005).
17. Software libre
http://es.wikipedia.org/wiki/Software_libre (18/05/2005)
18. Por qué usar programas abiertos? ¡Atención a las cifras!
<http://www.hispalinux.es> (19/05/2005)
19. Stallman, Richard. Software libre para una sociedad libre. Diciembre 2004,
Versión 1.0
<http://www.sindominio.net/biblioweb/pensamiento/softlibre/index.html>
(14/05/2005)
20. ¿Qué es GNU/Linux?
<http://www.molinux.info> (10/05/2005)
21. Zugaldia, Antonio. Plataforma Mono
<http://www.monohispano.com> (14/05/2005)
22. Serrano, Jorge..NET Framework, uno para todos y todos para uno
<http://www.msdn.com> (10/05/2005)
23. Trejo Martínez, Janhil Aurora. "Base de datos".
<http://www.monografias.com/trabajos11/basda/basda.shtml#bas> (2/05/2005)

24. Domingo, Alejandro. Common Language Runtime

<http://www.elguille.info> (21/05/2005)

Glosario de términos

EbXML (E-Business Extensible Markup Language): es un estándar basado en XML exclusivamente para los negocios en la red.

Licencia GNU/GPL La GNU General Public License (Licencia General Pública de GNU) es la norma por la que se rigen muchas aplicaciones de libre distribución, entre las que se encuentran la mayoría de los trabajos en torno al sistema GNU/Linux.

RosettaNET: Conjunto de estándares para Internet que contiene diccionarios de datos, un framework y especificaciones de proceso para la estandarización de e-Business.

SOAP (Simple Object Access Protocol): Protocolo basado en XML para intercambio de información en ambiente distribuido.

TCP/IP Se refieren a dos protocolos de red: Transmission Control Protocol (Protocolo de Control de Transmisión) e Internet Protocol (Protocolo de Internet). Los diferentes protocolos de la suite TCP/IP trabajan conjuntamente para proporcionar el transporte de datos dentro de Internet (o Intranet). En otras palabras, hacen posible que accedamos a los distintos servicios de la Red.

UDDI (Universal, Description, Discovery and Integration): es un estándar basado en XML que permite que las aplicaciones sepan lo que un Web Service ofrece.

WSDL (Web Service Description Language): Este protocolo basado en XML, es un lenguaje en formato XML que define las operaciones que proporciona un servicio.

XML (Extensible Markup Language): Es un metalenguaje de marcas que permite definir cómo es la información que se transmite. Esto permite una comunicación de datos entre distintos sistemas.