

# Universidad de las Ciencias Informáticas

## Facultad 2



## Título: Fachada de Servicios Telemáticos

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autoras** Rita Elena López Díaz  
Lilian Saúco Altuna

**Tutora** Ing. Lianet Liben Martínez

**Co-tutores** Ing. Yunier Saborit Ramírez  
Ing. Manuel Alejandro Gil Martín

**Asesores** Ing. Yudiel Alfredo Tamayo Agramonte  
Ing. Ronny Zamora Aguilar

Ciudad de La Habana, Junio del 2008

**UN MODO DE VENCER ES, DESDE YA,  
NO DEJAR DE INTENTARLO...**

## DECLARACIÓN DE AUTORÍA

Declaramos ser autoras de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

---

Rita Elena López Díaz  
Autora

---

Lilian Saúco Altuna  
Autora

---

Ing. Lianet Liben Martínez  
Tutora

## AGRADECIMIENTOS

*A mi abuela Elena, por ser amiga, por su apoyo, su constancia y comprensión, en todos los momentos de esta carrera. Por brindarme el impulso cuando no tuve fuerzas, secando mis lágrimas, riendo conmigo, por su confianza en mí.*

*A mi mamá, por todos sus consejos, su fe en mí, por no dejarme sola en ningún instante. Por su educación, por transmitirme toda su fuerza, valor y ternura.*

*A mi padre, por guiar mis pasos, por ser mi orgullo, guía, amigo, porque a pesar de haber estado lejos, siempre sentí su mano cerca, dándome fuerzas para seguir adelante.*

*A mi hermana, por existir, por las fuerzas que me da y por su inmenso cariño.*

*A toda mi familia que me quiere, mis tíos, mis abuelos, por todo aquello que me brindaron para que pudiera seguir adelante, sin ellos tampoco este sueño pudiera haberse realizado.*

*A mis tres hermanas, Ivette, Yeilín y Daiamna, por haber estado conmigo en los momentos difíciles, por su apoyo, su ayuda, gracias por hacerme estos 5 años más alegres.*

*A todos mis amigos, aunque algunos ya no estén, les agradezco haber compartido conmigo momentos alegres y tristes durante esta carrera.*

*A Yamílka, Midelis, Marialís, Dilma, Halena, Sure, Yaser, Yuniel, Yady y a todos los demás que me ayudaron este último año y sobre todo con este trabajo, que igual son muy especiales e importantes para mí.*

*A mi tutora Lianet, Chony, Ronny, Alberto, Dionísio, Yudiel, Yunier, por brindarme su ayuda transmitiéndome todos los conocimientos posibles.*

*Agradezco a la Revolución y a Fidel por esta universidad y por permitir que me formara profesionalmente en ella.*

*Por último, a aquellas personas que no creyeron que podía llegar hasta aquí, de esa manera, me ayudaron a encontrar fuerzas para seguir adelante y demostrarle lo contrario.*

*Rita Elena*

*Agradezco...*

*A mis padres, porque al buscar las palabras exactas para representarlos aquí, sencillamente me percaté de que son TODO para mí... mis verdaderos guías, mis mayores orgullos, mis ejemplos a seguir...*

*A mi hermana por confiar siempre en mí y porque aunque no lo crea, profesionalmente la admiro mucho y desde hace un tiempo intento seguir sus pasos...*

*A mis abuelos y tíos por su cariño y por estar siempre muy pendientes de cada uno de mis pasos... A mis primos, en especial a Aidita y a Enríquito por sus revisiones, sugerencias y estar tan cerca y unidos a mí durante estos 5 maravillosos años de universidad...*

*A Martica y a Maylín, mis otras dos "hermanas", porque el simple hecho de saber que están ahí duplica mis alegrías, minimiza mis angustias y me da fuerzas para seguir adelante...*

*A Deby y a Yunia por su incondicional amistad y apoyo en todo, solo lamento no haberlas conocido antes...*

*A mis amigos del pre, en especial a Darián que no me verá graduarme, pero sé que en pensamiento estará conmigo para decirme que todo estará bien...*

*A mis amigos de la universidad, en especial de mis primeros grupos 2108 y 2204, y de "La Cumbiamba"...*

*A mi gran amigo Rafael porque su compañía ha significado siempre seguridad y confianza para mí...*

*A mi compañera de tesis, porque cuando dije que hacíamos un buen equipo de trabajo lo dije siempre muy en serio... "tú me empujas y yo te freno"... esa fue la clave del éxito...*

*A Lianet, Yudiel, Ronny, Alberto, Dionisio, Chony y Yunier... porque sin la guía y ayuda de ellos no habría sido posible la realización de este trabajo.*

*A Zenayda, Dayamí, Juan Carlos y Yolexi, por la estupenda familia que formamos en uno de los periodos más bellos y difíciles que me ha tocado vivir...*

*A la Universidad de las Ciencias Informáticas (UCI) que tanto quiero y desafiando, por demostrarme que siempre hay tiempo para todo, y que es importante dar lo que se puede y un poquito más... y por enseñarme, entre otras cosas, a no dormir... A la Revolución y a Fidel por brindarme la posibilidad de ver realizado este maravilloso sueño...*

*A todos aquellos que, con obstáculos, me han hecho crecer...*

*Lilían*

DEDICATORIA

*A mis inigualables padres, a mi abuelita Flena y mi hermanita, por ser la razón de mi vida.*

*Rita Flena*

*A mis queridos padres, hermana y abuelos... porque más que mio, este también es su sueño...*

*Lilian*

## RESUMEN

Comunicarnos con una red, acceder a archivos ubicados en otra PC, dialogar en tiempo real con personas que pueden estar muy distantes o acceder a un sitio Web, son tareas que realizamos casi a diario. Estas acciones que tienen que ver con las temáticas de acceso y procesamiento de información entre computadoras son denominadas en términos informáticos Servicios Telemáticos.

Para apoyar y soportar la automatización de estos servicios en la Universidad de las Ciencias Informáticas (UCI), cuya misión es crear una micro-sociedad completamente informatizada que pudiera constituir el modelo de sociedad cubana del futuro, se hace necesario llevar un control estricto y real de todos sus usuarios.

El objetivo central del presente trabajo es realizar el análisis, diseño e implementación de una herramienta que permita sincronizar la información real de todo el personal UCI registrado en los diferentes sistemas primarios de información de las personas que existen hoy en dicha institución, con la posibilidad de gestionar estos datos y llevar un control más estricto de las modificaciones que se le realizan. Este proceso se ejecuta hoy de forma manual, con la consecuente sobrecarga del Directorio Activo del centro, que constantemente se consulta y encuesta para llevar a cabo dichas funciones. Debe desarrollarse además una Fachada Telemática donde tengamos servicios que brinden la información registrada en la aplicación anterior.

Una vez desarrolladas estas soluciones se dispondrá de un medio eficiente para facilitar el trabajo del personal encargado de realizar la gestión de los servicios básicos de las personas con relación a los servicios telemáticos en la UCI.

## PALABRAS CLAVE

Servicios Telemáticos, Directorio Activo, Personal UCI, Fachada Telemática.

## **ABSTRACT**

To communicate with a network, to have access to files located on another computer, to talk in real time with other people who may be very distant or to access a Web site, are almost daily tasks. These actions related to the topics of access and processing information between computers are the so called, in computer terms, Telematic Services.

To assist and support the automatization of these services at the University of Computer Sciences (UCI), whose mission is to create a complete computerized micro-society that would be the Cuban society model, it is necessary to keep strict and real control of all its users.

The main objective of this paper is to provide the analysis, design and implementation of a tool that allows real synchronize information of the UCI registered staff in the primary information systems of the current personnel; to process the data and control the changes made.

This process runs manually, with the resulting overloading of the Center Active Directory, which is constantly consulted and surveyed to carry out these functions. It must also be developed a telematic facade where we provide the information recorded in the previous application.

With these solutions, once developed, we'll count on an efficient means to facilitate the work of the staff in charge of carrying out the management of basic services of the telematic services in the UCI.

## **KEYWORDS**

Telematic Services, Active Directory, UCI Staff, Telematic facade



## TABLA DE CONTENIDOS

<b>AGRADECIMIENTOS</b> .....	<b>I</b>
<b>DEDICATORIA</b> .....	<b>III</b>
<b>RESUMEN</b> .....	<b>IV</b>
<b>ABSTRACT</b> .....	<b>V</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
<b>1.1 Introducción</b> .....	<b>5</b>
<b>1.2 Estudio del Estado del Arte</b> .....	<b>5</b>
1.2.1 Sistemas de Gestión de Datos Telemáticos.....	6
1.2.2 Vista Única.....	6
1.2.3 Fachada de Servicios.....	8
<b>1.3 Metodología a utilizar</b> .....	<b>8</b>
1.3.1 Metodología de desarrollo de software: RUP.....	8
<b>1.4 Lenguajes a utilizar</b> .....	<b>11</b>
1.4.1 Lenguaje de modelado: UML .....	11
1.4.2 Lenguaje de programación: PHP .....	11
<b>1.5 Herramientas y Tecnologías a utilizar</b> .....	<b>12</b>
1.5.1 Herramienta CASE: Visual Paradigm.....	12
1.5.2 Herramienta de diseño de páginas Web: Dreamweaver .....	13
1.5.3 Herramienta de desarrollo: Zend Studio .....	14
1.5.4 Servidor Web: Apache .....	14
1.5.5 Framework: Code Igniter .....	15
1.5.6 Sistema Gestor de Base de Datos (SGBD): MySQL .....	16
1.5.7 Servicios Web .....	17
1.5.7.1 XML (Extensible Markup Language).....	17
1.5.7.2 SOAP (Simple Object Access Protocol).....	17
1.5.7.3 WSDL (Servicios Web Definition Language).....	18

1.5.7.4	UDDI (Universal Description, Discovery and Integration)	18
1.5.8	Arquitectura de Software: SOA	18
<b>1.6</b>	<b>Patrones aplicados</b>	<b>19</b>
1.6.1	Patrones de Casos de Uso	19
1.6.2	Patrones de Diseño	19
1.6.3	Patrón de Arquitectura de Software	20
<b>1.7</b>	<b>Conclusiones del Capítulo</b>	<b>21</b>
 <b>CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA</b>		 <b>22</b>
<b>2.1</b>	<b>Introducción</b>	<b>22</b>
<b>2.2</b>	<b>Objeto de estudio</b>	<b>22</b>
2.2.1	Problema y situación problemática	22
2.2.2	Objeto de automatización	23
<b>2.3</b>	<b>Solución Propuesta</b>	<b>23</b>
<b>2.4</b>	<b>Modelamiento del Negocio o Ambiente del Sistema</b>	<b>24</b>
2.4.1	Modelo de Dominio o Modelo Conceptual	25
2.4.2	Glosario de Términos para el Modelo de Objetos del Dominio	25
2.4.3	Modelo de Objetos del Dominio	26
2.4.4	Reglas del Negocio a considerar	27
<b>2.5</b>	<b>Modelamiento del Sistema</b>	<b>29</b>
2.5.1	Requerimientos	29
1.3.1.1	Requisitos funcionales	30
1.3.1.2	Requisitos no funcionales	30
2.5.2	Actores del Sistema	32
2.5.3	Casos de Uso del Sistema	32
2.5.4	Diagrama de Casos de Uso del Sistema	33
2.5.5	Especificación de los Casos de Uso del Sistema	34
2.5.5.1	CU1_Gestionar Datos de Usuario	34
2.5.5.2	CU2_Buscar Usuarios	37
2.5.5.3	CU3_Mostrar Datos de Usuario	38

2.5.5.4	CU4_Generar Reporte.....	39
2.5.5.5	CU5_Brindar Datos de Usuario .....	42
2.5.5.6	CU6_Sincronizar/Actualizar Datos de Usuario .....	48
2.5.5.7	CU7_Autenticar Usuario.....	49
<b>2.6</b>	<b>Conclusiones del Capítulo .....</b>	<b>50</b>
 <b>CAPÍTULO III: ANÁLISIS Y DISEÑO .....</b>		<b>52</b>
<b>3.1</b>	<b>Introducción.....</b>	<b>52</b>
<b>3.2</b>	<b>Diagrama de Clases.....</b>	<b>52</b>
<b>3.3</b>	<b>Análisis del Sistema.....</b>	<b>52</b>
3.3.1	Definición del Modelo de Análisis .....	52
3.3.2	Clases del Análisis .....	52
3.3.3	Diagramas de Clases del Análisis .....	53
3.3.3.1	DCA del CU: Gestionar Datos de Usuario .....	54
3.3.3.2	DCA del CU: Mostrar Datos de Usuario .....	54
3.3.3.3	DCA del CU: Generar Reporte .....	55
3.3.3.4	DCA del CU: Brindar Datos de Usuario.....	55
3.3.3.5	DCA del CU: Sincronizar/Actualizar Datos de Usuario.....	56
3.3.3.6	DCA del CU: Autenticar Usuario .....	56
<b>3.4</b>	<b>Diseño del Sistema.....</b>	<b>57</b>
3.4.1	Definición de Modelo de Diseño .....	57
3.4.2	Diagramas de Clases del Diseño.....	57
3.4.2.1	DCD del CU: Gestionar Datos de Usuario .....	57
3.4.2.2	DCD del CU: Mostrar Datos de Usuario .....	58
3.4.2.3	DCD del CU: Generar Reporte .....	59
3.4.2.4	DCD del CU: Brindar Datos de Usuario.....	60
3.4.2.5	DCD del CU: Sincronizar/Actualizar Datos de Usuario.....	60
3.4.2.6	DCD del CU: Autenticar Usuario .....	61
3.4.3	Diagramas de Interacción.....	61
3.4.3.1	Diagrama de Secuencia.....	61

3.4.3.2	Diagrama de Colaboración.....	62
<b>3.5</b>	<b>Diagrama de Clases Persistentes.....</b>	<b>62</b>
<b>3.6</b>	<b>Conclusiones del Capítulo .....</b>	<b>63</b>
<b>CAPÍTULO IV: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....</b>		<b>64</b>
<b>4.1</b>	<b>Introducción.....</b>	<b>64</b>
<b>4.2</b>	<b>Diagrama de Despliegue .....</b>	<b>64</b>
<b>4.3</b>	<b>Diagrama de Componentes.....</b>	<b>65</b>
4.3.1	Diagrama de Componentes: Base de Datos .....	65
4.3.2	Diagrama de Componentes: Código Fuente .....	66
4.3.3	Diagrama de Componentes: Componentes Web o Código Ejecutable.....	67
<b>4.4</b>	<b>Conclusiones del Capítulo .....</b>	<b>67</b>
<b>CAPÍTULO V: ESTUDIO DE LA FACTIBILIDAD .....</b>		<b>68</b>
<b>5.1</b>	<b>Introducción.....</b>	<b>68</b>
<b>5.2</b>	<b>Planificación basada en casos de uso .....</b>	<b>68</b>
<b>5.3</b>	<b>Beneficios Tangibles e Intangibles .....</b>	<b>73</b>
<b>5.4</b>	<b>Análisis de Costos y Beneficios.....</b>	<b>74</b>
<b>5.5</b>	<b>Conclusiones del Capítulo .....</b>	<b>74</b>
<b>CONCLUSIONES .....</b>		<b>75</b>
<b>RECOMENDACIONES .....</b>		<b>76</b>
<b>BIBLIOGRAFÍA.....</b>		<b>77</b>
<b>ANEXOS.....</b>		<b>80</b>
<b>GLOSARIO .....</b>		<b>86</b>

## INTRODUCCIÓN

La Universidad de las Ciencias Informáticas (UCI), como institución universitaria, cuenta con un gran número de personas afiliadas a sus diferentes áreas de trabajo y estudio. Su inmensa labor por la digitalización se hace cada vez mayor dada la importancia de que todas las informaciones y servicios se automaticen, tarea esta que se hace cada día más necesaria.

Actualmente nuestra universidad cuenta con varios sistemas que registran la información de todo el personal que existe en ella. Bases de datos como Akademos (registra información de los estudiantes), Assets (registra información de los trabajadores UCI) y Registro de Personal (registra información de los que no son estudiantes ni trabajadores UCI, pero que prestan servicios a la universidad o viven en ella), se encargan de registrar esta valiosa información, ya que los servicios telemáticos deben nutrirse de dichos datos.

Todo el personal registrado en estas bases de datos está relacionado con los servicios informáticos que se brindan en la universidad. Para hacer uso de los mismos se necesita mayormente verificar ciertos datos de condición única para cada persona, de ahí la importancia de tener actualizados dichos sistemas de información de las personas.

Uno de los recursos más importantes con que cuenta la UCI son los servicios telemáticos, entre ellos el correo electrónico, servicio de mensajería instantánea, navegación básica en Internet, acceso pleno a Internet, navegación interna, cuota de almacenamiento para buzones de correo, cuota de tráfico para el acceso a Internet, entre otros. Estos servicios tienen una alta demanda de acceso por parte de los usuarios de la UCI, tanto en forma de consulta como de autenticación.

El Directorio Activo de la UCI es el punto único de integración de todos los servicios telemáticos, sin dejar de mencionar que se consultan a través de él informaciones de estado de cuentas, login, etc. Actualmente se leen los datos telemáticos por una vista de ciudadano, pero que en realidad se registran los datos que vienen en una planilla. Es importante señalar que la información que está en el directorio activo no hay manera de importarla hoy hacia los demás sistemas.

Debido a las condiciones anteriormente citadas, se observa que el directorio se muestra sobrecargado dada su relación con las múltiples aplicaciones que constantemente lo encuestan, por lo que se hace necesaria la creación de un sistema que almacene las informaciones de los usuarios, a las cuales se

accedería en forma de consulta para de esta forma dejar al directorio sólo con las tareas de autenticación.

Dada la situación problemática detallada se plantea entonces el siguiente **problema científico**, centrando interés en las investigaciones que deben realizarse para darle una solución efectiva:

¿Cómo sincronizar de manera eficiente la información de los usuarios de la UCI y los servicios telemáticos?

Seguidamente se esclarece el objetivo general y los específicos que servirán de guía para el desarrollo de la solución.

**Objetivo general**: Desarrollar una fachada de servicios para el Directorio Activo de la UCI.

**Objetivos específicos**:

- Definir y analizar los procesos que se van a automatizar.
- Desarrollar un sistema de réplica orientada a servicios.
- Desarrollar una aplicación Web para actualizar y gestionar los datos de los usuarios significativos para los servicios telemáticos.
- Desarrollar una orquestación de los servicios básicos de información de las personas.

Las investigaciones se centran en la parte objetiva donde se debe actuar desde el punto de vista teórico y práctico para darle solución al problema, para ello se define como **objeto de estudio**:

Los datos telemáticos.

Dentro de estos se define como **campo de acción** a investigar:

Los procesos de gestión de datos telemáticos.

Concretados ya el problema científico, objeto de estudio y campo de acción, se puntualiza la **idea a defender**:

Con el desarrollo de un sistema que presente una Fachada Telemática para el Directorio Activo de la UCI se contará con un proceso de actualización automática de la información en la Base de Datos del Nodo.

Las **tareas de investigación** que servirán de apoyo para el cumplimiento de los objetivos específicos quedan definidas a continuación:

- Estudio de las necesidades que conllevan a la creación del sistema.
- Realización de las entrevistas necesarias para el levantamiento de requisitos.
- Selección de las herramientas para el desarrollo de la aplicación.
- Estudio de los servicios telemáticos que se brindan actualmente.
- Estudio y análisis del proceso de sincronización actual entre sistemas y aplicaciones con el directorio.
- Estudio de los servicios básicos de las personas con relación a los servicios telemáticos.

### **Estructuración del contenido:**

**Capítulo 1:** *“Fundamentación Teórica”*. En este capítulo se hace un análisis del estado del arte del objeto de estudio, se investiga sobre las soluciones informáticas vinculadas al campo de acción y se fundamentan las tecnologías, herramientas, metodología y lenguajes utilizados para el desarrollo de la aplicación informática.

**Capítulo 2:** *“Características del Sistema”*. Este capítulo describe el objeto de estudio, la solución propuesta, el modelamiento del negocio y del sistema. Se definen además las reglas del negocio, así como las características y funcionalidades del sistema. Para ello se realiza una modelación detallada del sistema a través de descripciones y diagramas de casos de uso para comprender mejor el funcionamiento de la aplicación que se diseñará.

**Capítulo 3:** *“Análisis y Diseño”*. Incluye la definición y descripción de los diagramas de clases de aplicaciones web del análisis y del diseño, así como los diagramas de interacción del mismo para cada realización de los casos de uso. Muestra también la descripción de cada una de las clases.

**Capítulo 4:** *“Construcción de la Solución Propuesta”*: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes, estructurando así la aplicación.

**Capítulo 5:** *“Estudio de Factibilidad”*: Se describe la aplicación de la técnica de estimación del esfuerzo.



## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En el presente capítulo se ofrece una visión general de los aspectos teóricos relacionados con el desarrollo de la Fachada Telemática. Son tratados los temas y conceptos precisos para entender la solución propuesta y que permiten facilitar el entendimiento del trabajo que se presenta. Son descritas las tecnologías, herramientas, lenguajes y metodologías utilizadas.

### 1.2 Estudio del Estado del Arte

La Telemática<sup>1</sup> es una disciplina científica y tecnológica que surge de la evolución de la telecomunicación y de la informática. Aunque usualmente se usa la concepción simplista de la Telemática como la mera aplicación de las técnicas de la telecomunicación y de la informática sobre la transmisión a larga distancia de información computarizada, la Telemática incluye en realidad un campo más amplio y engloba el estudio, diseño, gestión y aplicación de las redes y servicios de comunicaciones, para el transporte, almacenamiento y procesado de cualquier tipo de información (datos, voz, vídeo, etc.).

Cada vez con mayor frecuencia, se encuentra en los medios de comunicación noticias y reportajes que hacen referencia a la importancia que han adquirido para la actual sociedad interconectar los Servicios Telemáticos en todos los centros informatizados, lo cual ha provocado que en el mundo hayan surgido numerosos sistemas automatizados dirigidos al control eficiente y veloz de dichos servicios.

Lo anteriormente descrito plantea la necesidad de llevar un estudio acerca de las nuevas tendencias a garantizar eficientes procesos de gestión de datos telemáticos. Se asocian a estas tareas los términos: sistemas de gestión de datos telemáticos, vista única, fachada de servicios, entre otros; que tributan en todos los casos al aseguramiento de servicios telemáticos de calidad.

A continuación se hace un estudio crítico sobre cómo y hacia dónde se van enfocando las tecnologías de la información y las comunicaciones (TIC) en función de las mencionadas soluciones; argumentando la necesidad, definiciones, características y aplicaciones importantes de las mismas.

---

<sup>1</sup> "Ciencia y Tecnología". Disponible en: [http://cybermicrologic.com/cml\\_web/esp/html/3\\_tecnologia.html](http://cybermicrologic.com/cml_web/esp/html/3_tecnologia.html) (22/05/2008)

### 1.2.1 Sistemas de Gestión de Datos Telemáticos

El acceso a los servicios telemáticos por parte de los usuarios es algo que no sigue un esquema general que haya sido planteado a nivel internacional, ni siquiera de cada nación en particular. Cada empresa, organismo o institución se encarga de establecer las políticas de acceso de sus usuarios a los servicios que brinda, garantizando para ello la disponibilidad y control de los datos primarios e informaciones relevantes de cada uno de los usuarios con acceso a estos servicios.<sup>2</sup>

Se nota, por tanto, que el aumento y desarrollo de las políticas de control de acceso a este valioso recurso es de vital importancia, dado el nivel de confidencialidad e integridad que debe tener la información generada, que se encuentra disponible y accesible por medio de las redes de computadoras actuales. La estrategia que mayormente se sigue es el establecimiento de cuentas de usuarios con permisos para acceder a los distintos servicios de intercambio de información, de esta forma se evita que el personal no autorizado tenga acceso a los mismos.

Son precisamente los Sistemas de Gestión de Datos Telemáticos los encargados de avalar que la información que se brinda de los usuarios mantiene una única versión de la realidad del ciudadano en toda la administración: “la información adecuada en el momento preciso”. De este modo el manejo de los servicios telemáticos se podrá llevar a cabo de manera racional y controlada, y cada usuario obtendrá el acceso a los servicios en dependencia de sus permisos.

### 1.2.2 Vista Única

En la actualidad, la información vital de un ciudadano es distribuida por los organismos de la administración mediante una serie de sistemas diferentes. Esta situación puede dar lugar a errores y procesos ineficientes que, a su vez, derivan en un menoscabo del servicio al ciudadano.

La integración de los sistemas y la información que estos contienen (con el objetivo de mejorar la calidad de los procesos) ha sido una batalla durante décadas y los intentos de resolver este reto han variado a lo largo de la historia: desde la integración punto-a-punto entre sistemas hasta invertir el tiempo de los propios empleados sincronizando la información entre sistemas. La salida no es otra que la implantación de soluciones como la Vista Única.

---

<sup>2</sup> Iztiair G. Alonso González, Gustavo Rodríguez Rodríguez, Domingo Marrero Marrero. “*Servicios Telemáticos*”. Disponible en: <http://absys01.ulpgc.es/cgi-bin/abnetopac/O7081/IDed04735a/NT15> (15/11/2007)

Gracias a la Vista Única, los distintos organismos esperan poder identificar a un ciudadano y obtener de modo rápido la visibilidad completa de su relación con toda la información y los servicios de los colectivos a los que pertenece, y ofrecerle una vista consistente mediante todos los canales de acceso posibles.

La Plataforma de Aplicaciones Compuestas de Sun (Sun Java CAPS), utilizando una Vista Unificada ofrece una plataforma de integración completa e integrada para construir y gestionar Aplicaciones Compuestas que mantengan la inversión en las aplicaciones existentes, con un incremento de la flexibilidad y la agregación de sus datos: Sun SeeBeyond eView.

### **Sun SeeBeyond eView:**<sup>3</sup>

- Proporciona un marco de trabajo flexible para diseñar y configurar un Índice Maestro del Ciudadano a escala Administración, que crea una vista única de la información del ciudadano.
- Mantiene la información más actualizada de las personas que lo integran a lo largo de los distintos organismos y conecta información de diferentes lugares y sistemas.
- Emplea una estandarización inteligente de los datos y algoritmos de emparejamiento (matching) para reconocer la información relativa al ciudadano que se encuentra distribuida en dichos sistemas a lo largo de toda la arquitectura, enlazando las identidades (IDs) de los silos de información en una identidad universal del ciudadano y creando un índice maestro del ciudadano.
- Garantiza datos precisos de los miembros, identificando potenciales registros duplicados y ofreciendo la posibilidad de fusionar o resolver la duplicidad de registros.
- Toda la información de los miembros se centraliza en un índice compartido.
- Mantiene una base de datos centralizada para múltiples sistemas, por lo que permite integrar datos, mientras los sistemas locales continúan operando de forma independiente sin verse afectados en su operación.

La Vista Única puede utilizarse para mejorar la calidad de los datos de origen, para identificar y construir nuevas aplicaciones de diferentes colectivos de ciudadanos. Puede ser también la base de una nueva generación de servicios orientados al ciudadano.

---

<sup>3</sup> RUBIO, Manuel. “Vista Única: la base para una nueva generación de servicios orientados al ciudadano”. Disponible en: [http://blogs.sun.com/manuelrubio/entry/vista\\_%C3%BAnica\\_la\\_base\\_para](http://blogs.sun.com/manuelrubio/entry/vista_%C3%BAnica_la_base_para) (06/06/2008)

En este sentido las Arquitecturas Orientadas a Servicios nos brindan la posibilidad de ofrecer mejores servicios al ciudadano. El empleo de estándares, la interoperabilidad y la capacidad de publicar y compartir servicios entre los organismos de la Administración hacen posible que el ciudadano disfrute de una vista unificada y consolidada de su información y pueda acceder a sus servicios de un modo seguro y desde cualquier lugar, siendo transparente para este la dispersión geográfica de sus datos y de los sistemas de información que lo soportan.

### 1.2.3 Fachada de Servicios

Una fachada de servicios o legacy, es un conjunto de servicios web que brindan una determinada funcionalidad de un sistema existente que no estaba hecho o preparado para la comunicación con otros sistemas. Desarrollarle una fachada de servicios a un sistema no es más que implementar servicios web que accedan a ese sistema o a sus bases de datos, para exponer información referente a ellos hacia otros sistemas de una intranet.

## 1.3 Metodología a utilizar

### 1.3.1 Metodología de desarrollo de software: RUP

RUP<sup>4</sup> (Proceso Unificado de Desarrollo de Software) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

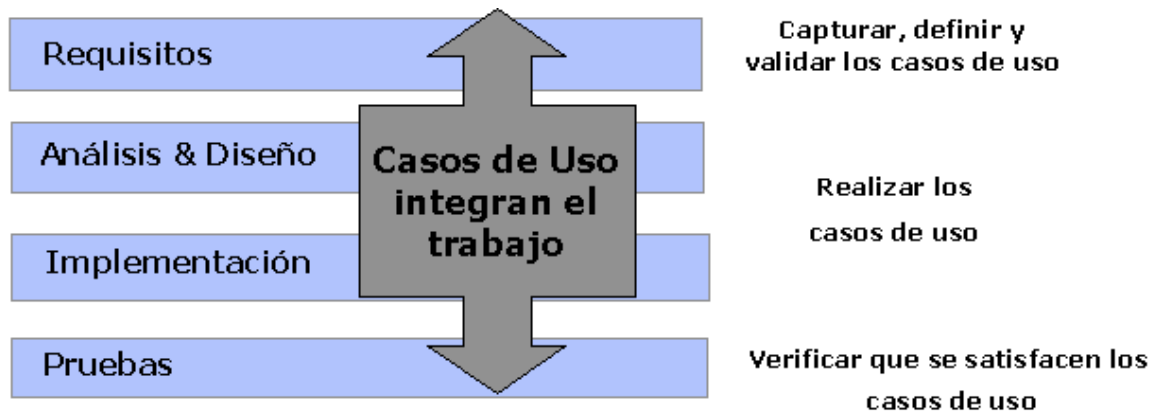
El ciclo de vida de RUP se caracteriza por:<sup>5</sup>

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. En RUP los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, también guían su diseño, implementación y prueba. Constituyen un elemento integrador y una guía del trabajo, como se muestra en la *Figura 1.1*.

---

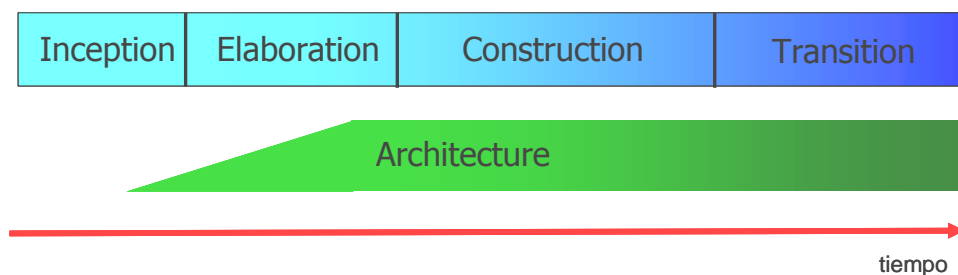
<sup>4</sup> Booch, G., Rumbaugh, J., Jacobson, I. “*El Proceso Unificado de Desarrollo de Software*”. Addison-Wesley Longman, 2000.

<sup>5</sup> Dpto. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. “*Introducción a RUP*”, 2006. Disponible en <https://pid.dsic.upv.es/C1/Material/default.aspx> ( 23/11/2007)



**Figura 1.1:** Los Casos de Uso integran el trabajo

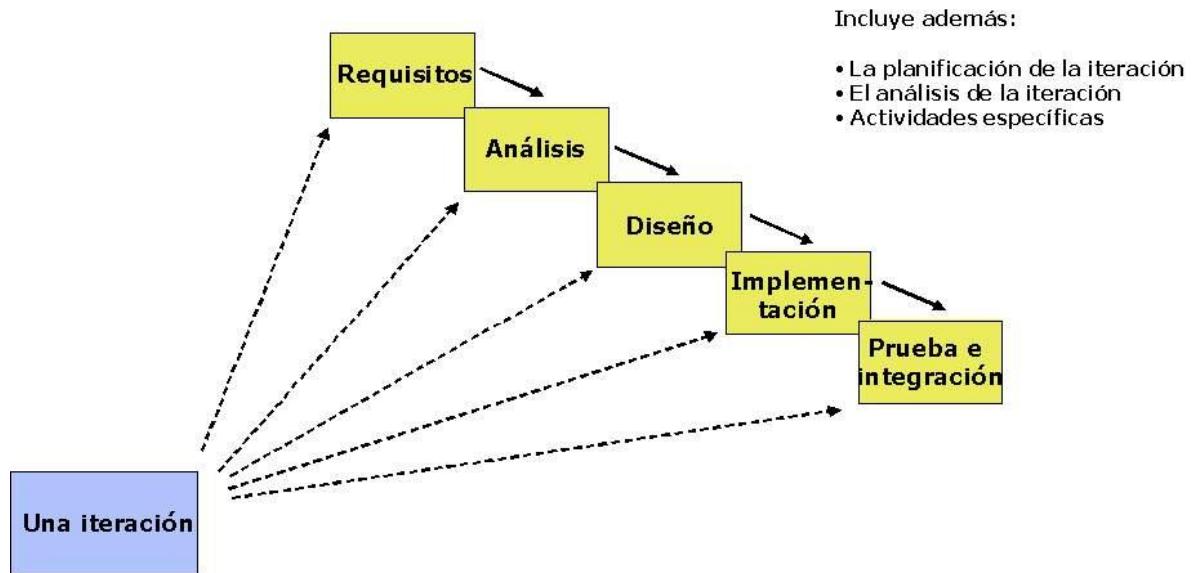
- Centrado en la arquitectura: La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. En el caso de RUP, además de utilizar los casos de uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Los casos de uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y en el futuro. En la *Figura 1.2* se ilustra la evolución de la arquitectura durante las fases de RUP, observando una arquitectura más robusta en las fases finales del proyecto.



**Figura 1.2:** Evolución de la arquitectura del sistema

- Iterativo e incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración, como se refleja en la *Figura 1.3*, involucra actividades de todos los flujos de trabajo. El proceso iterativo e incremental consta de una secuencia de iteraciones, donde cada una de ellas aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes. Aunque

cada iteración tiene que proponerse un incremento en el proceso de desarrollo que produzca un crecimiento en el producto, todas deben aportar al principal resultado de la fase en la que se desarrolla.



**Figura 1.3:** Una iteración RUP

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Es una metodología muy organizativa y eficiente.<sup>6</sup>

El desarrollo de este trabajo, dado sus características, necesita una documentación profunda, que sirva de base para futuros cambios, una mitigación de riesgos a tiempo y una buena organización. La metodología RUP ofrece cómo resolver las necesidades anteriormente descritas, a través de las características de su ciclo de desarrollo, agrupando las actividades en grupos lógicos que se definen por 9 flujos de trabajo (negocio, requerimientos, análisis y diseño, implementación, prueba, instalación, administración de proyectos, administración de configuración y cambios, y ambiente) y 4 fases de desarrollo (inicio, elaboración, construcción y transición).

Además, RUP describe los requisitos del software desde el punto de vista del usuario y define los requisitos técnicos sin incluir detalles de implementación. Cuenta con diferentes elementos de

<sup>6</sup> Mendoza Sánchez, María A. “Metodologías de desarrollo de software”. Disponible en: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html) (23/11/2007).

planificación (plan de desarrollo, plan de iteración, plan de calidad, etc.) que permiten controlar el desarrollo del software. A través de un predefinido esquema de escalabilidad y gestión de riesgos, se pueden reconocer previamente problemas y fallos de forma temprana y prevenirlos o corregirlos. Define en cada momento del ciclo de vida del proyecto, qué artefactos, con qué nivel de detalle y por qué rol, se deben crear.

### 1.4 Lenguajes a utilizar

#### 1.4.1 Lenguaje de modelado: UML

**UML**<sup>7</sup> es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, desde una perspectiva OO.

UML viene muy ligado a la metodología que se escoge para el desarrollo de este software. Este sistema requiere de características y facilidades que brinda este lenguaje. Se puede destacar entre ellas que detrás de cada símbolo en la notación UML hay una semántica bien definida, de manera que un desarrollador puede escribir un modelo en UML, y otro desarrollador, o incluso otra herramienta, puede interpretar ese modelo sin ambigüedad.

Permite ingeniería directa e inversa y además es un lenguaje que cubre toda la documentación de la arquitectura de un sistema y todos sus detalles. Proporciona un lenguaje para expresar requisitos y pruebas del software. Provee un lenguaje para modelar las actividades de planificación de proyectos y gestión de versiones. Facilita la gestión de los modelos, pues ayuda a mantener la consistencia entre los elementos del sistema y ayuda a mejorar la habilidad de un equipo de desarrollo para manejar la complejidad del software. Además de verificar la calidad del software.

#### 1.4.2 Lenguaje de programación: PHP

**PHP** es un lenguaje de programación rápido y con una gran librería de funciones. Se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web con los cuales se puede programar las páginas HTML y los códigos fuente.<sup>8</sup>

---

<sup>7</sup> Booch, G., Rumbaugh, J., Jacobson, I. “*El Lenguaje Unificado de Modelado*”, Addison-Wesley, 2000

<sup>8</sup> “*PHP*”. Disponible en: [http://www.ideocentro.com/recursos.php?id=14&ideocentro\\_diseno\\_web\\_interactive=PHP--PHP-Hypertext-Pre-processor-](http://www.ideocentro.com/recursos.php?id=14&ideocentro_diseno_web_interactive=PHP--PHP-Hypertext-Pre-processor-) (22/05/2008)

Su uso para la programación de este sistema es muy importante, ya que es un lenguaje multiplataforma, que lee y manipula datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Permite las técnicas de Programación Orientada a Objetos y crear los formularios para la web; además de poseer una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Brinda muchas facilidades a la hora de su uso para la programación por disponer de una biblioteca nativa de funciones sumamente amplia e incluida.

### 1.5 Herramientas y Tecnologías a utilizar

La selección de las herramientas y tecnologías a utilizar constituye un paso de vital importancia durante el proceso de desarrollo de cualquier sistema informático, pues debe garantizar un óptimo desempeño del sistema si se realiza de manera correcta.

#### 1.5.1 Herramienta CASE: Visual Paradigm

Las Herramientas CASE (Computer Aided Software Engineerinh o Ingeniería de Software Asistida por Ordenador) son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.<sup>9</sup> Estas permiten incrementar la productividad y el control de calidad en cualquier proceso de elaboración de software, ya que transforman la actividad de desarrollar software en un proceso automatizado.

La herramienta CASE escogida para el modelado de la aplicación fue el Visual Paradigm, debido fundamentalmente a que es multiplataforma.

**Visual Paradigm**<sup>10</sup> es una suite completa de herramientas CASE que da soporte al modelado visual con UML 2.0, ofreciendo distintas perspectivas del sistema. Es independiente de la plataforma y está dotada de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, así como garantizar la calidad del producto final.

Posee entre sus principales características las siguientes:

---

<sup>9</sup> Instituto Nacional de Estadística e Informática, Dirección Técnica de Desarrollo Informático. *Herramientas CASE*. Disponible en: <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>. (22/01/2008)

<sup>10</sup> Sitio oficial del Visual Paradigm. Disponible en: <http://www.visual-paradigm.com/product/vpuml>. (21/01/2008)



- Es profesional: brinda la posibilidad de crear un conjunto bastante amplio de artefactos utilizados con mucha frecuencia durante la confección de un Software. Todos estos, cumpliendo con el Standard UML 2.0.
- Es amigable: puede ser utilizado en varios idiomas, sus componentes se encuentran relacionados, por lo que se hace muy fácil la creación de cualquier tipo de diagrama.
- Brinda un número considerable de estereotipos: lo que permite un mayor entendimiento de los diagramas.
- Facilidades para redactar especificaciones de casos de uso: es posible crear plantillas para las especificaciones de casos de uso y describirlos.
- Generación de código e ingeniería inversa: brinda la posibilidad de generar código a partir de los diagramas, para plataformas como PHP, que es el escogido para el desarrollo de este software, así como obtener diagramas a partir del código.
- Interoperabilidad con otras aplicaciones: brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose. Además permite importar y exportar XML y XMI.
- Generación de código ORM: (Object–Relational Map) permite generar a partir de un Diagrama de Entidad Relación una Base de Datos Relacional y el código necesario para acceder a esta base de datos utilizando PHP entre otros lenguajes.
- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.(Paradigm).

### 1.5.2 Herramienta de diseño de páginas Web: Dreamweaver

Dreamweaver es la herramienta de diseño de páginas web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrará en este programa razones para utilizarlo, sobretodo en lo que a productividad se refiere. Esta herramienta cumple perfectamente el objetivo de diseñar páginas con aspecto profesional y soporta gran cantidad de tecnologías, además muy fáciles de usar:<sup>11</sup>

---

<sup>11</sup> “Probablemente el mejor editor de páginas web para diseñadores que busquen resultados profesionales”. Disponible en: <http://www.desarrolloweb.com/articulos/332.php> (13/03/2008)

Su selección se basa fundamentalmente en que proporciona una combinación potente de herramientas visuales, características de desarrollo de aplicaciones y soporte para la edición de código. Destaca funciones para cargar y descargar archivos, ahora funcionan en el background sin interrumpir la productividad en el programa. Se pueden diseñar páginas con una interfaz completamente gráfica mientras se observa simultáneamente el código generado. Además, como el lenguaje escogido fue PHP, esta herramienta colorea el código y te permite testear las páginas con un servidor configurado.

### 1.5.3 Herramienta de desarrollo: Zend Studio

**Zend Studio** es uno de los ambientes de desarrollo integrado o IDE (Integrated Development Environment) por sus siglas en inglés, destinado a desarrolladores profesionales que agrupa todos los componentes necesarios para un ciclo de desarrollo de aplicaciones PHP.

Es un IDE propietario, compatible con las plataformas Linux, MAC y Windows. Incluye todos los componentes necesarios durante el ciclo de vida de una aplicación en PHP. Incluye editor, análisis, depuración, optimizadores de código y herramientas de base de datos. Zend Studio nos permite agilizar el desarrollo web y permite simplificar proyectos complejos.<sup>12</sup>

Entre sus principales características destaca excelente completamiento de código, coloreado en la sintaxis del código, administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documenter, manual de PHP, integración con subversión, los navegadores, integración avanzada con FTP, soporte para Web Services, PHP4, PHP5 y SQL.

Agiliza nuestro trabajo, cuenta con un buen Depurador, infinitas opciones que permiten un desarrollo profesional de nuestras aplicaciones.

### 1.5.4 Servidor Web: Apache

Un servidor Web es un programa que implementa el protocolo HTTP (Hypertext Transfer Protocol) y se encarga de atender las peticiones HTTP llevadas a cabo por un cliente y responder con el contenido que el mismo solicita. Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML.

---

<sup>12</sup> PÉREZ VALDÉS, Damián. *“Editores Web que facilitan tu trabajo”*. Disponible en: <http://www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo> (06/02/2008)

El **Servidor Apache** se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation y es hoy el servidor Web más utilizado del mundo, ofreciendo una perfecta combinación entre estabilidad y sencillez, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Por supuesto, se distribuye prácticamente con todas las implementaciones de Linux.<sup>13</sup>

Las principales características que se tuvieron en cuenta para su elección fueron:

- Funcionalidad en múltiples plataformas.
- Elaborado índice de directorios.
- Sencilla administración basada en la configuración de un único archivo.
- Mensajes de error altamente configurables.
- Bases de datos de autenticación y negociado de contenido.

Además, Apache está diseñado para ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos, a través de su diseño modular. Es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web, permitiendo que una misma característica o funcionalidad sea implementada de diferentes maneras para obtener una mayor eficiencia.

### 1.5.5 Framework: Code Igniter

Un framework en el contexto de la programación es un set de funciones o código genérico que realiza tareas comunes y frecuentes en todo tipo de aplicaciones (creación de objetos, conexión a base de datos, limpieza de strings, etc.). Esto brinda una base sólida sobre la cual desarrollar aplicaciones concretas y permite obviar los componentes más triviales y genéricos del desarrollo.<sup>14</sup>

El empleo de frameworks permite además lograr un código ordenado, limpio, fácil de actualizar, robusto y mucho más eficiente.

---

<sup>13</sup> Sitio: The Apache Software Foundation. Disponible en: <http://www.apache.org/> (06/02/2008)

<sup>14</sup> CELIS Ismael. “*El ataque de los Frameworks*”. Disponible en: <http://www.estadobeta.com/2005/11/20/el-ataque-de-los-frameworks/> (15/03/2008)

**Code Igniter** es un poderoso marco de desarrollo para PHP, con una muy pequeña huella, construido para los codificadores de PHP que necesitan un sencillo y elegante conjunto de herramientas para crear todo lo que ofrecen las aplicaciones web.<sup>15</sup>

Se selecciona este framework pues está pensado para ofrecer un alto rendimiento, ser ligero y fácilmente instalable. Viene con varias librerías para gestionar el acceso a datos, sesiones de usuarios, formularios, la seguridad, etc. Es compatible tanto para PHP4 como para PHP5, ofrece magnífica documentación, enorme comunidad de desarrolladores y no requiere más de media hora de estudio para empezar a hacer cosas interesantes.

Utiliza el patrón de diseño Modelo-Vista-Controlador (MVC) como paradigma de arquitectura de desarrollo, la cual separa en 3 capas distintas: la representación de datos, la interfaz de usuario y el controlador de eventos respectivamente.

### 1.5.6 Sistema Gestor de Base de Datos (SGBD): MySQL

Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.<sup>16</sup>

**MySQL**<sup>17</sup> es el sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

MySQL es un sistema de administración de base de datos y a la vez es un sistema de administración *relacional de bases de datos*<sup>18</sup>, permitiendo velocidad y flexibilidad. Es un software de fuente abierta que presenta múltiples motores de almacenamiento, agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo, disponibilidad en gran cantidad de plataformas y sistemas, conectividad segura, replicación, etc. Su relación con php es muy eficiente.

---

<sup>15</sup> Web oficial de Code Igniter. Disponible en: <http://codeigniter.com> (30/04/2008)

<sup>16</sup> "Sistemas Gestores de Base de Datos". Disponible en: [http://www.error500.net/garbagecollector/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos.html](http://www.error500.net/garbagecollector/bases_de_datos/sistema_gestor_de_base_de_datos.html) (24/11/2007)

<sup>17</sup> "Plan de Cases: Curso de Instalación y configuración de CMS". Disponible en: <http://dspace.idict.cu/bitstream/123456789/134/1/Plan+de+Cases.doc> (23/05/2008)

<sup>18</sup> Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo.

### 1.5.7 Servicios Web

Un **servicio web** (en inglés Web Service) es un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías no presentan una interfaz gráfica con la que las personas puedan interaccionar, sino que intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.

Los servicios Web son muy prácticos ya que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio, permitiendo que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más acusada.<sup>19</sup>

Hay un juego de estándares de los que se habla ligados a los servicios web. Incluyen los siguientes:<sup>20</sup>

#### 1.5.7.1 XML (Extensible Markup Language)

**XML** (Lenguaje Extensible de Etiquetas): Lenguaje de definición de documentos que permite una forma estándar de representar datos para su fácil intercambio. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible.

#### 1.5.7.2 SOAP (Simple Object Access Protocol)

**SOAP** (Protocolo de Acceso Simple a Objetos): Es un protocolo de comunicación, el cual permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma y del lenguaje. Está basado en XML y es la base principal de los SW, los mensajes SOAP son documento XML propiamente dicho. La especificación de SOAP está bien definida y es

---

<sup>19</sup> Oficina española del w3c (World Wide Web). Guía Breve de Servicios Web. Disponible en: <http://www.w3c.es/Divulgacion/Guiasbreves/ServiciosWeb#uso#uso> (21/01/2008)

<sup>20</sup> BANKHACKER.COM. Taller: Computación Distribuida: Servicios Web de Traducción. Disponible en: <http://web-services.bankhacker.com> (21/01/2008)

sumamente simple. Independiente de plataformas y lenguajes. El uso de XML y HTTP hace a SOAP completamente independiente de plataformas, sistemas operativos o lenguajes de programación. Permitiendo el uso de protocolos de transporte como FTP, SMTP, además de HTTP y sus variantes (HTTPS). Los sistemas distribuidos basados en SOAP tienen un bajo acoplamiento, por lo cual pueden ser fácilmente mantenidos dado que pueden ser modificados independientemente de otros sistemas.

### 1.5.7.3 WSDL (Servicios Web Definition Language)

**WSDL** (Lenguaje de Descripción de Servicios Web): Es un protocolo basado en XML que describe los accesos al SW. Se puede decir que es el manual de operación del SW, porque indica cuáles son las interfaces que provee el SW y los tipos de datos necesarios para la utilización del mismo, la forma de accederlos e interactuar con ellos.

### 1.5.7.4 UDDI (Universal Description, Discovery and Integration)

**UDDI** (Integración, Descubrimiento y Descripción Universal): Es un modelo de directorios para SW. Es una especificación para mantener directorios estandarizados de información acerca de los SW, sus capacidades, ubicación y requerimientos, en un formato reconocido universalmente. Utiliza WSDL para describir las interfaces de los SW.

## 1.5.8 Arquitectura de Software: SOA

La Arquitectura Orientada a Servicios (en inglés Service-Oriented Architecture o SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario.

**SOA** es una arquitectura de software que permite la creación y/o cambios de los procesos de negocio desde la perspectiva de TI de forma ágil, a través de la composición de nuevos procesos utilizando las funcionalidades de negocio que están contenidas en la infraestructura de aplicaciones actuales o futuras (expuestas bajo la forma de web services).

SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación. En un ambiente SOA, los nodos de la red hacen disponibles sus recursos a otros participantes en la red como servicios independientes a los que tienen acceso de un modo estandarizado.

## 1.6 Patrones aplicados

### 1.6.1 Patrones de Casos de Uso

- Patrón CRUD (Creating, Reading, Updating and Deleting): este patrón posee un caso de uso, llamado “Información CRUD” o “Gestionar Información”, que modela todas las operaciones que se pueden realizar sobre una parte de información de un tipo determinado; ya sea crearla, leerla, actualizarla y eliminarla. Es importante destacar que este patrón se usa cuando todos los flujos contribuyen al mismo valor de negocio y son cortos y simples.
- Actores múltiples: Rol común (Multiple Actors: Common Role): dos actores interpretan el mismo papel en determinado caso de uso. Este rol es representado por otro actor, que contiene de forma hereditaria los actores que comparten este rol. Este patrón debe aplicarse cuando, desde el punto de vista de un caso de uso hay solo una entidad externa interactuando con cada instancia del caso de uso.<sup>21</sup>

### 1.6.2 Patrones de Diseño

Los Patrones de Diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces, introduciendo el trabajo con patrones de diseño para asignar responsabilidades (GRASP).

- Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos. La intención básica del patrón es encontrar un creador que necesite conectarse a este objeto en alguna situación, ante la problemática de quién debería ser el responsable de la creación de una nueva instancia de alguna clase.
- Controlador: consiste en asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.
- Experto: consiste en asignar la responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir dicha responsabilidad.
- Bajo acoplamiento: el acoplamiento mide qué tan fuerte está una clase conectada con otras (es decir, cuántas clases conoce y necesita). Una clase con bajo acoplamiento no depende de

---

<sup>21</sup> LARMAN, Craig. “UML y Patrones. Introducción al análisis y diseño orientado a objetos”. Primera Edición por Prentice Hall, Hispanoamericana S.A. 1999. Tomo I, Capítulo 18.

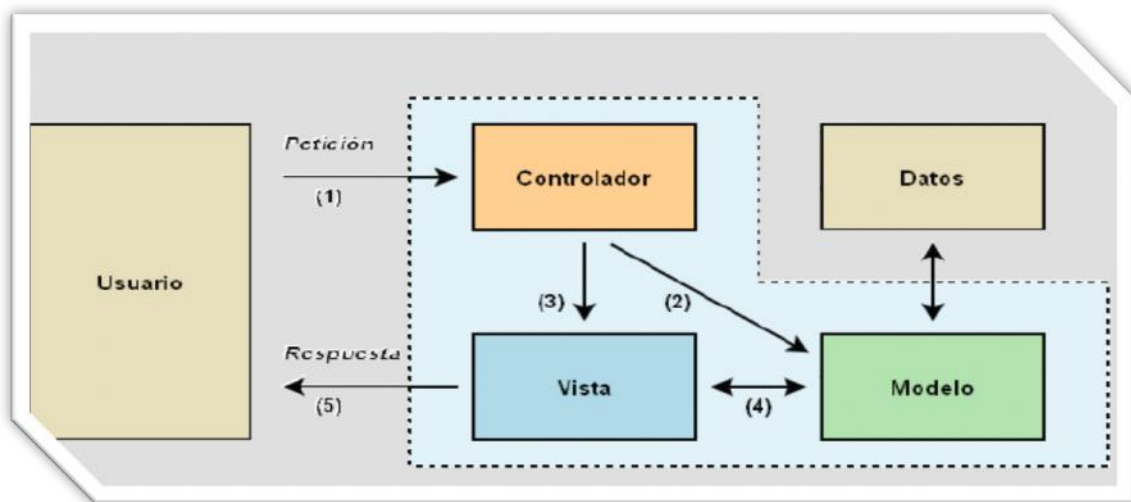
"muchas otras" clases. Una clase con alto acoplamiento recurre a muchas otras clases. Este tipo de clase no es conveniente, pues: cambios en las clases relacionadas ocasionan cambios en la clase local; son más difíciles de entender; son más difíciles de reutilizar.

- Alta cohesión: surge ante la problemática de cómo mantener la complejidad dentro de límites manejables. En la perspectiva del diseño orientado a objetos, la cohesión (o, más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

### 1.6.3 Patrón de Arquitectura de Software

El patrón **MVC** es el patrón de arquitectura de software que introduce el framework CodeIgniter. El mismo separa los datos<sup>22</sup> de una aplicación, la interfaz de usuario<sup>23</sup> y la lógica de control<sup>24</sup> en tres componentes distintos.

La *Figura 1.4* muestra la interacción entre el usuario y el sistema MVC.



**Figura 1.4:** Diagrama de clases de la interacción entre usuario y el Sistema MVC

<sup>22</sup> El dato (del latín *datum*), es una representación simbólica (numérica, alfabética, etc.), atributo o característica de una entidad. El dato no tiene valor semántico (sentido) en sí mismo, pero convenientemente tratado (procesado) se puede utilizar en la realización de cálculos o toma de decisiones. En programación un dato es la expresión general que describe las características de las entidades sobre las cuales opera un algoritmo.

<sup>23</sup> La interfaz de usuario es la forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

<sup>24</sup> La lógica de control Es el conjunto de operaciones lógicas y estructuras de control que determinan el orden de ejecución de las instrucciones de un programa.



- *Modelo*: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- *Vista*: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- *Controlador*: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Este patrón se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

### 1.7 Conclusiones del Capítulo

En este capítulo se realizó un estudio y análisis de los conceptos necesarios para el entendimiento de la creación del sistema desarrollado, centrandó especial interés en las nuevas tendencias y tecnologías vinculadas a la solución propuesta.

El objetivo primordial en esta sección ha sido lograr que el lector se ubique, de forma general, en los aspectos teóricos a tener en cuenta para el desarrollo del sistema, con la intención de sumergirlo en el mundo de los servicios telemáticos y las diversas formas existentes de gestionarlos.

Se fundamentó la metodología de desarrollo del software, el lenguaje de programación y el sistema gestor de base de datos a utilizar, así como el lenguaje de modelado y la herramienta CASE seleccionada para la modelación tanto de los procesos del negocio como del diseño de la aplicación; llegando a la conclusión de utilizar las mismas durante el ciclo de desarrollo del sistema.

## CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

### 2.1 Introducción

En el presente capítulo se presenta la solución propuesta para implementar el sistema, a partir de la realización de la metodología RUP. Se describe el modelo de dominio para representar el negocio actual, así como los casos de uso del sistema involucrados en la solución del problema. Se detallan los requerimientos funcionales y los no funcionales que tendrá el producto informático, permitiendo representar una concepción general del sistema e identificar mediante un Diagrama de Casos de Uso las relaciones de los actores que interactúan con el sistema y las secuencias de acciones que se realizan.

### 2.2 Objeto de estudio

#### 2.2.1 Problema y situación problemática

Se utilizan<sup>25</sup> en nuestra universidad tres sistemas que registran<sup>26</sup> información del personal de la UCI: Akademos, Asset y Registro de personal; abarcando informaciones de los estudiantes, trabajadores y otras personas (viven, visitan o prestan servicios a la universidad), respectivamente.

Todo este personal, en dependencia de la labor que desempeña, necesita acceder a los servicios que ofrece<sup>27</sup> el centro universitario, entre ellos los servicios telemáticos, interactuando<sup>28</sup> con los mismos lo que los convierte en lo que llamamos usuarios de la UCI.

La universidad utiliza también un Directorio Activo que controla<sup>29</sup> datos únicos para cada uno de sus usuarios y este directorio es el que gestiona<sup>30</sup> los servicios telemáticos; es decir que para acceder a estos servicios obligatoriamente hay que comunicarse con el Directorio Activo, en forma de consulta y/o autenticación, labor que ocurre de forma transparente al usuario pero que constantemente se está llevando a cabo debido a la gran cantidad de usuarios existentes en la universidad.

---

<sup>25</sup> **Utilizar:** servirse de una cosa, aprovecharse de su uso.

<sup>26</sup> **Registrar:** Poner algo de manifiesto para su examen y anotación.

<sup>27</sup> **Ofrecer:** Prometer. Presentar y dar una cosa para que alguien las aproveche.

<sup>28</sup> **Interactuar:** Ejercer entre personas o cosas actos propios de su naturaleza.

<sup>29</sup> **Controlar:** Comprobar, examinar, inspeccionar, revisar. Dirigir, gobernar, dominar.

<sup>30</sup> **Gestionar:** Hacer diligencias para el logro de algún objeto.

La información que se almacena en el Directorio Activo no hay manera de importarla hoy hacia los diferentes sistemas, haciendo difícil la sincronización de los servicios telemáticos y la información primaria de las personas.

### 2.2.2 Objeto de automatización

Entre las actividades en las que está inmerso el sistema propuesto, se propone automatizar la sincronización y actualización de los datos (primarios y telemáticos) de los usuarios de la UCI en una BD central del Nodo, los cuales se encuentran dispersos geográficamente en los diferentes sistemas de almacenamiento de información de usuarios de dicho centro.

Para ello se desarrollan cuatro acciones fundamentales:

- ✓ Identificar usuarios que hayan sido alta a la universidad y añadirlos al sistema.
- ✓ Identificar usuarios que hayan sido baja de la universidad y eliminarlos del sistema.
- ✓ Identificar modificaciones a los datos primarios de los usuarios existentes y actualizarlos en el sistema.
- ✓ Identificar variaciones en los datos telemáticos de los usuarios existentes y actualizarlos en el sistema.

Las informaciones sobre los datos primarios de las personas se obtienen de las bases de datos Akademos, Assets y Registro de Personal. Y las referentes a los datos telemáticos se suministra del Directorio Activo.

## 2.3 Solución Propuesta

La solución está compuesta por servicios Web que brindan una Fachada Telemática para el Directorio Activo de la UCI (por ejemplo, visualizar la información de la cuota de los usuarios) y un software de gestión que realiza la actualización automática de la información en la Base de Datos del Nodo y que debe ser administrado por el nodo de redes de la UCI.

Para ello se hace necesaria la sincronización de toda la información almacenada en las bases de datos Akademos, Assets y Registro de Personal, en el nuevo sistema; y este sería entonces quien interactuaría con el Directorio, de manera que se mantenga actualizado y realice las acciones

necesarias para modificar la información contenida en él, para llevar los datos reales existentes en cada uno de los gestores de datos anteriormente mencionados.

En otras palabras, el nuevo sistema se convierte en el punto de comunicación entre los sistemas primarios de información de las personas y los servicios telemáticos.

La *Figura 2.1* describe la solución propuesta anteriormente detallada.

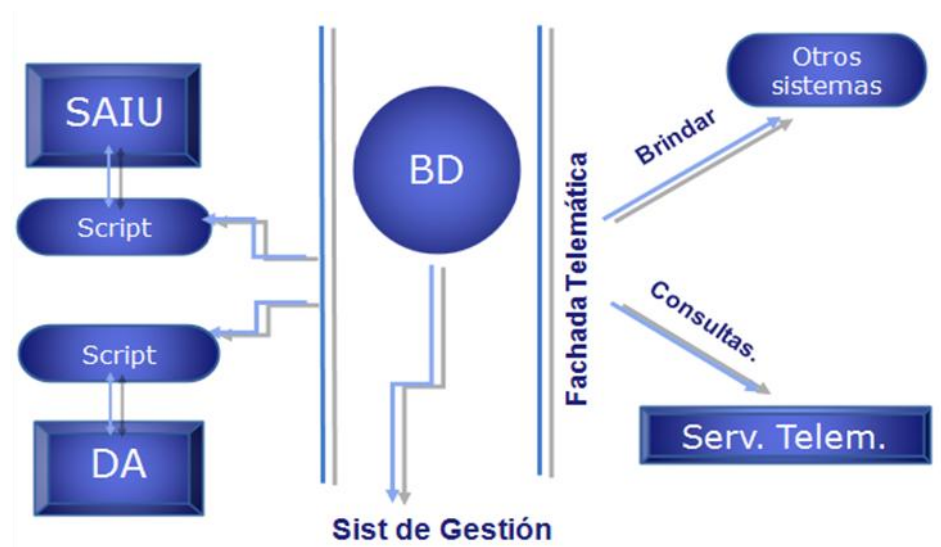
Términos:

**SAIU:** Sistemas de Almacenamiento de Información de los Usuarios de la UCI.

**DA:** Directorio Activo de la UCI.

**Serv. Telem.:** Servicios Telemáticos.

**BD:** Base de Datos.



*Figura 2.1: Solución Propuesta*

## 2.4 Modelamiento del Negocio o Ambiente del Sistema

La evaluación del estado del negocio o descripción del negocio consiste básicamente en evaluar el estado actual de la organización en la cual el sistema será explotado. Esto permite identificar correctamente los procesos del negocio y definir cómo modelarlos.

### 2.4.1 Modelo de Dominio o Modelo Conceptual

Dependiendo de la situación o escenario que presente el negocio, RUP propone dos alternativas para desarrollar este proceso: modelo de negocio y modelo de dominio.

En este caso se determinó que no era necesario un modelo completo del negocio debido a que los procesos que se estudian no se encuentran bien definidos; por lo que se muestra el funcionamiento de la organización mediante el modelado de objetos del dominio o modelamiento del dominio, capturando los tipos más importantes de objetos que se presentan en el contexto descrito que permitirán mostrar los principales conceptos manejados en el dominio del sistema en desarrollo.

Se considera que un escenario apropiado para esta alternativa es aquel donde el objetivo principal es la gestión y presentación de información. El modelo de dominio se considera en RUP un subconjunto del llamado modelo de objetos del negocio.

### 2.4.2 Glosario de Términos para el Modelo de Objetos del Dominio

Para mayor comprensión de los términos o conceptos representados en el modelo de objetos del dominio a continuación se definen y especifican cada uno de ellos. Esto permite establecer un lenguaje común entre desarrolladores, clientes, usuarios finales del sistema y otros interesados.

**UCI:** Universidad de las Ciencias Informáticas, centro que representa el dominio donde se establecerá el nuevo sistema y donde radica el personal que utilizará el mismo.

**SAIU:** Sistema de Almacenamiento de Información de Usuarios, aplicación cuyo objetivo central es registrar y brindar informaciones relevantes de las personas de una institución, en este caso de la UCI.

Se utilizan en la UCI tres SAIU, abarcando entre ellos a todo su personal:

**Akademos:** SAIU que registra las informaciones de los estudiantes de la universidad, con el objetivo central de garantizar la sistematización de las actividades docentes del centro.

**Assets:** SAIU que registra las informaciones de los trabajadores que pertenecen a dicha institución universitaria, con el objetivo de llevar un adecuado control de los recursos humanos.

**Registro de Personal:** SAIU que registra las informaciones de aquellas personas que prestan servicios a la universidad, familiares de algunos trabajadores que residen en el centro y

personal que nos visite por algún período de tiempo que necesite tener acceso a los servicios que presta la universidad.

**Servicio Telemático:** Prestación que tiene que ver con las temáticas de acceso y procesamiento de información entre computadoras de una red informática o área determinada de la misma, en este caso los servicios de este tipo que ofrece la UCI.

**Directorio Activo:** Componente central de la plataforma Windows que proporciona los medios para gestionar las identidades y relaciones que organizan los entornos de red de una institución, en este caso de la UCI.

**UsuarioUCI:** Persona que tiene acceso a todos o algunos de los Servicios Telemáticos que ofrece la UCI.

Los usuarios de la UCI se clasifican, para su almacenamiento y control, en:

**Estudiante:** Usuario que cursa estudios en el centro.

**Trabajador:** Usuario que labora en el centro.

**Otro:** Usuario que presta servicios al centro (terceros<sup>31</sup> y eventuales<sup>32</sup>), es familiar de algún trabajador que reside en el centro o se encuentra de visita por algún período de tiempo en la universidad.

### 2.4.3 Modelo de Objetos del Dominio

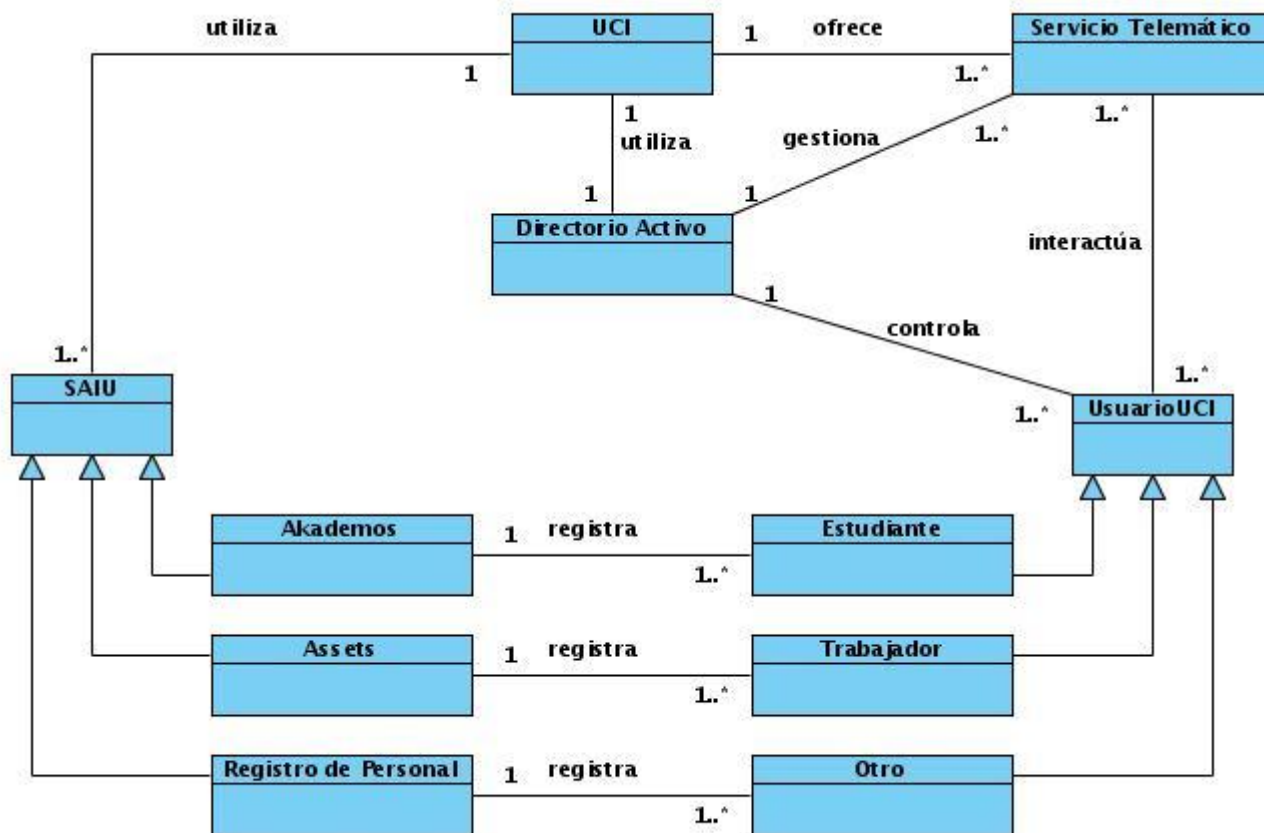
Un modelo conceptual (Modelo del Dominio) explica los conceptos significativos en el dominio del problema, constituyendo el artefacto más importante durante el análisis orientado a objetos, ya que descompone el problema en conceptos u objetos individuales.

Una vez definidos los objetos del dominio, que representan las “cosas” que existen o los eventos que suceden en el entorno en que trabaja el sistema, no resultará difícil comprender las relaciones que se establecen entre los mismos, representados en el siguiente Modelo de Objetos del Dominio.

---

<sup>31</sup> Persona que presta servicios a la universidad en ramas como por ejemplo Salud, Comercio y Gastronomía.

<sup>32</sup> Persona que pertenece a otro centro de trabajo y presta servicios a la universidad por ejemplo como Profesor Adjunto.



**Figura 2.2:** Modelo de Objetos de Dominio

#### 2.4.4 Reglas del Negocio a considerar

Las reglas del negocio son una colección de políticas, restricciones de una organización que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. Se deben expresar de manera que pueda ser validada su exactitud por el personal conector del negocio así como poder verificar recíprocamente su coherencia.<sup>33</sup>

Los procesos que se llevan a cabo en cualquier centro cumplen determinadas reglas que han sido implantadas con anterioridad. Las mismas deben tenerse en cuenta a la hora de automatizar dichos procesos pues de lo contrario el software no cumpliría las restricciones del lugar.

<sup>33</sup> The Business Rules Group. “Manifiesto de Reglas de Negocio. Los Principios de la Independencia de las Reglas”. Disponible en: <http://www.businessrulesgroup.org/brmanifesto/BRManifiesto%5Bv1%5B1%5D.0%5D.pdf>

### 1. Reglas de estructura:

- ✓ Término:<sup>34</sup>
  - Persona: Estudiantes, Trabajadores, Terceros, Eventuales, Administrador y Usuario.
  - Se considera como Servicio Telemático a las acciones que tienen que ver con las temáticas de acceso y procesamiento de información entre computadoras.
  - Se considera como Usuario a las personas que utilizan servicios telemáticos.
  - Se consideran como datos de las personas relevantes para los servicios telemáticos los siguientes: ID del expediente (número de solapín), ID de la Foto, Nombre(s) del usuario, Apellidos del usuario, Usuario (login), CI, Tipo de persona, Cuota de Navegación, Cuota utilizada, Nivel de navegación, Capacidad del inbox, Type.
  - Se considera como Reporte al listado de usuarios que cumplen determinada condición.
- ✓ Modelo de datos:<sup>35</sup>
  - La cantidad de estudiantes, trabajadores, terceros y eventuales es mayor que cero en cada caso.
  - La unidad de medida del volumen asociado a la cuota de navegación, cuota utilizada y capacidad del inbox de cada usuario debe ser expresada siempre en MegaBytes (MB).
- ✓ Relación:<sup>36</sup>
  - El administrador controla el sistema para conocer los datos relevantes para los servicios telemáticos de los estudiantes, trabajadores, terceros y eventuales, a través de reportes que solicita.

### 2. Reglas de derivación:

- ✓ Inferencia:<sup>37</sup>
  - Un reporte debe contener al menos un usuario en dicha lista.
- ✓ Cálculo:<sup>38</sup>

---

<sup>34</sup> Conceptos en el contexto del negocio.

<sup>35</sup> Controla que la información básica almacenada para cada atributo o propiedad de un concepto es válida.

<sup>36</sup> Controla las relaciones entre los datos.

<sup>37</sup> Especifican que un hecho es cierto por inferencia.

<sup>38</sup> Controla la obtención de información que se puede calcular a partir de la ya existente.



- La Cuota Disponible (de navegación) de un usuario es la diferencia entre la cuota de navegación y la cuota utilizada.

### 3. Reglas de acción:

- ✓ Flujo:<sup>39</sup>
  - Toda persona debe identificarse antes de solicitar ver o editar datos de alguna persona.
  - El administrador es el encargado de gestionar los datos de las personas y de generar los reportes asociados a estos datos.
  - Un usuario podrá ver detalles de sus propios datos telemáticos.
- ✓ Restricciones de operaciones:<sup>40</sup>
  - Ninguna persona puede ver datos de otra persona, a no ser que tenga permisos especiales para ello.
- ✓ Estímulo y respuesta:<sup>41</sup>
  - Si la persona no pertenece al dominio UCI, se exige indicar a qué dominio pertenece.

Este proceso de especificación de reglas del negocio implica que hay que “identificarlas” dentro del negocio, “evaluar” si son relevantes dentro del campo de acción que se está modelando e “implementarlas” en la propuesta de solución.

## 2.5 Modelamiento del Sistema

### 2.5.1 Requerimientos

A continuación se exponen los requisitos funcionales y los no funcionales a tener en cuenta para el desarrollo del sistema.

---

<sup>39</sup> Determinan y limitan cómo fluye la información a través de un sistema.

<sup>40</sup> Especifican condiciones que deben ser ciertas para asegurarse que una operación se ejecute correctamente.

<sup>41</sup> Restringen el comportamiento especificando cuándo y qué condiciones deben cumplirse para que una operación de respuesta sea inmediatamente ejecutada.

### 1.3.1.1 Requisitos funcionales

Los requerimientos funcionales son las condiciones o capacidades que el sistema debe cumplir. Son el punto de partida para identificar qué debe hacer el sistema, ya que definen el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica.

De acuerdo con los objetivos el sistema debe ser capaz de:

**RF1** Mostrar datos reales de los usuarios del dominio por parte del administrador.

**RF2** Eliminar usuarios del dominio por parte del administrador.

**RF3** Modificar datos de los usuarios del dominio por parte del administrador.

**RF4** Actualizar datos de los usuarios en el Directorio Activo.

**RF5** Leer datos de los usuarios a través de servicios WEB.

**RF6** Mostrar datos de los usuarios a través de servicios WEB.

**RF7** Autenticar usuarios.

**RF8** Realizar búsquedas de usuarios.

**RF9** Mostrar a un usuario sus datos.

**RF10** Generar reporte de los usuarios habilitados en la fachada de autenticación.

**RF11** Generar reporte de los usuarios que tienen la cuota deshabilitada.

**RF12** Generar reporte de los usuarios que no tienen login.

### 1.3.1.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener.

Un requerimiento no funcional es, en la ingeniería de sistemas y la ingeniería de software, una restricción que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requerimientos funcionales.

- ✓ Requerimientos de apariencia o interfaz externa: El software presenta una interfaz fácil de manejar por el usuario. Es muy legible y discreto para los usuarios, no está cargado de tantos gráficos ni de tanta animación. El usuario puede acceder a las diferentes opciones sin tener que navegar por muchos puntos intermedios.
- ✓ Requerimientos de Usabilidad: El producto puede ser usado por personas con un mínimo de conocimientos sobre los entornos Web y el manejo del ordenador.
- ✓ Requerimientos de Rendimiento: El tiempo de respuestas a solicitudes en el sistema, debe ser en el menor tiempo posible. El producto debe tener gran velocidad de procesamiento ya que el sistema utiliza grandes volúmenes de datos e información. Además el sistema debe estar disponible en todo momento para todos los usuarios y el SGBD deberá llevar a cabo las actualizaciones diarias.
- ✓ Requerimientos de Soporte: El producto requiere de constantes mantenimientos debido a futuras actualizaciones y cambios, después de instalado.
- ✓ Requerimientos de Portabilidad: El sistema es compatible con todos los sistemas operativos sin necesidad de cambios significativos.
- ✓ Requerimientos de Seguridad: La información que se almacenará en el sistema esta protegida de accesos y cambios no autorizados. Se mantiene la integridad de todos los datos, ya que los usuarios que accedan al software deberán autenticarse.
- ✓ Requerimientos Legales: El producto esta regido por las normativas que se establecen en los diferentes reglamentos de la Universidad de las Ciencias Informáticas (UCI).
- ✓ Requerimientos de Confiabilidad: El producto garantiza la integridad de todos los datos y su seguridad. En cada mantenimiento se deberán reportar posibles fallos y cómo serán monitoreados en caso de que ocurran. La herramienta de implementación tiene soporte para recuperación ante fallos y errores.
- ✓ Requerimientos de Software: Se deberá contar con: Navegador de Internet (Microsoft Internet Explorer, Netscape, Mozilla Firefox). Servidor Web (Apache para ambas. Para el funcionamiento del sistema en las terminales cliente es necesario el S.O. Windows 95 o superior, Linux o Unix.
- ✓ Restricciones en el diseño y la implementación: Para el análisis y el diseño del sistema debe ser utilizada la metodología RUP, usando el lenguaje de modelación UML y como herramienta para llevarlo a cabo el Visual Paradigm. Implementación del sistema en el lenguaje PHP. PhpMyAdmin, aplicación cliente utilizada para la gestión de la información en la base de datos. Zend Studio como herramienta de desarrollo.

### 2.5.2 Actores del Sistema

Un actor no es parte del sistema, es un rol o función que asume un usuario que puede intercambiar información o puede ser un recipiente pasivo de información. Representa a un ser humano, grupo, entidad, organización, software, máquina o sistema de información que interactúa con el sistema.

En este caso interactúan con el sistema los actores que se definen a continuación:

Actor	Justificación
Usuario	Es una generalización que representa el rol de los usuarios para autenticarse en la aplicación, y en dependencia de esto podrá desempeñarse como: "Usuario Dominio" o "Administrador".
Usuario Dominio	Representa el rol del individuo del dominio de la UCI o de alguna de las facultades regionales, que podrá acceder a la aplicación para realizar búsquedas de sus respectivos datos (registrados en el sistema) y visualizarlos.
Administrador	Representa el rol del individuo que utiliza y se beneficia de todas las funcionalidades de la aplicación: generar reportes y realizar búsquedas de datos de cualquier usuario, tanto para visualizarlos como para manipularlos (modificar/eliminar).
Sistema UCI	Es una generalización del rol de los sistemas externos que solicitan información de los usuarios del Dominio a la aplicación, a través de los servicios Web que brinda la misma.
Reloj	Representa el rol del actor del sistema que se encarga de actualizar una vez al mes los datos primarios de las personas y cada 24 horas los datos telemáticos de los usuarios en la base de datos central de la aplicación.

**Tabla 2.1** Descripción de los Actores del Sistema

### 2.5.3 Casos de Uso del Sistema

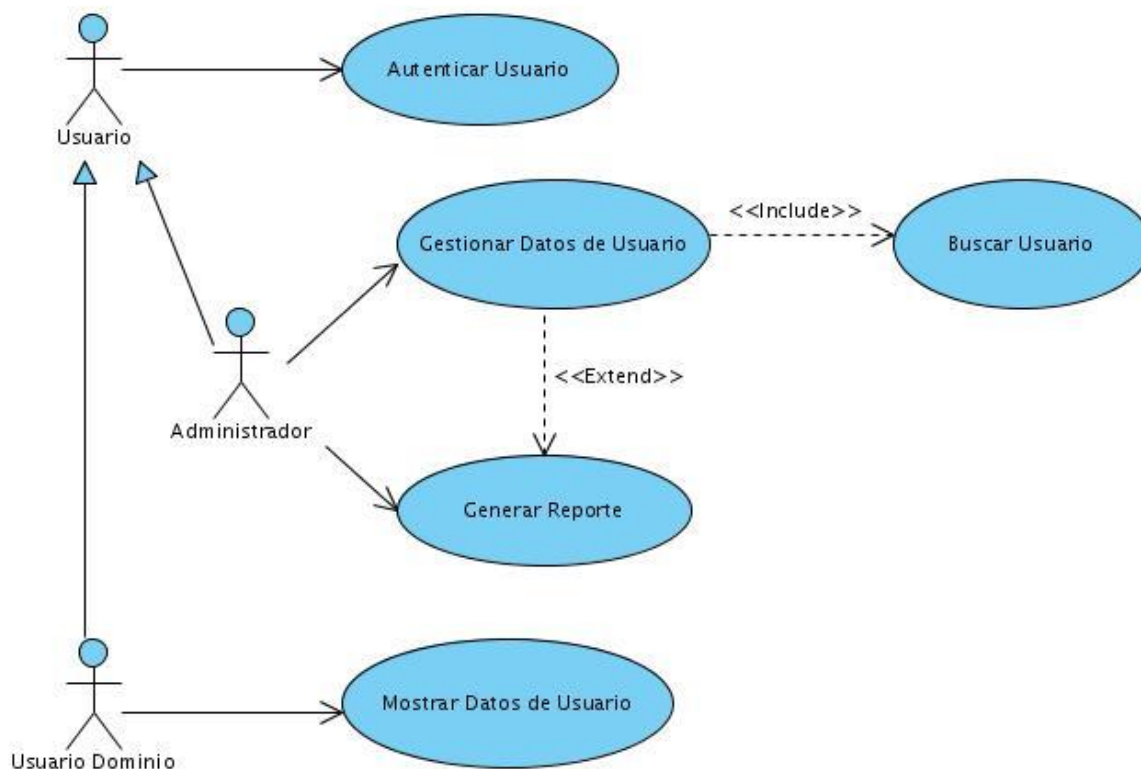
Los Casos de Uso describen las especificaciones de un sistema, qué hace, no cómo lo hace. Para el sistema que se propone se definen los siguientes casos de uso:

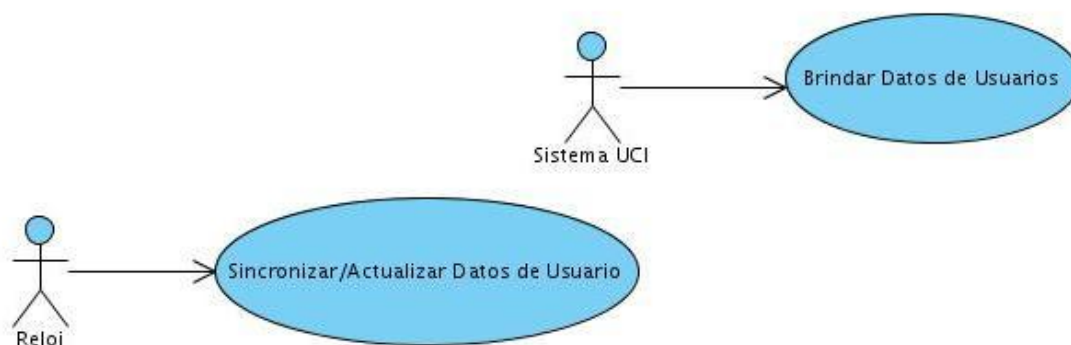
- 1 Gestionar Datos de Usuario.

- 1.1 Eliminar Usuario.
- 1.2 Ver Detalles.
- 1.3 Modificar Datos de Usuario.
- 2 Buscar Usuarios.
- 3 Mostrar Datos de Usuario.
- 4 Generar Reporte.
- 5 Brindar Datos de Usuario.
- 6 Sincronizar/Actualizar Datos de Usuario.
- 7 Autenticar Usuario

#### 2.5.4 Diagrama de Casos de Uso del Sistema

Un Diagrama de Casos de Uso del sistema, es un modelo que contiene actores, casos de uso y sus relaciones. A continuación se muestra el diagrama de casos de uso para el sistema propuesto:





**Figura 2.3:** Diagrama de Casos de Uso del Sistema

### 2.5.5 Especificación de los Casos de Uso del Sistema

Las especificaciones de los Casos de Uso son documentos narrativos que incluyen la secuencia de los eventos de un actor (agente externo) que utiliza el sistema para completar un proceso. Tienen como objetivo detallar su flujo de sucesos, incluyendo como comienza, termina e interactúa con los actores.

#### 2.5.5.1 CU1\_Gestionar Datos de Usuario

Caso de Uso:	CU1 Gestionar Datos de Usuario.
Actores:	Administrador
Propósito:	Llevar a cabo acciones sobre los datos de los usuarios para mantener la información actualizada.
Resumen:	El caso de uso se inicia cuando el Administrador indica llevar a cabo la acción de gestionar algún dato de un usuario. El caso de uso incluye 3 escenarios: Eliminar Usuario, Ver Detalles y Modificar Datos de Usuario. El sistema muestra las interfaces correspondientes a cada una de estas opciones. El caso de uso finaliza con las acciones necesarias que realiza el administrador para que se elimine un usuario, se muestren los datos o se modifiquen los mismos.
Referencia:	RF1, RF2, RF3
Precondiciones:	El usuario autenticado tiene que ser administrador del sistema. Deben existir usuarios insertados siempre que se desee eliminar usuarios o modificar datos de alguno de ellos del sistema.

	Debe estar seleccionado algún usuario para gestionar sus datos.
<b>Poscondiciones:</b>	Los datos de un usuario son eliminados o modificados.
<b>Flujo normal de los eventos:</b>	
Acciones del Actor	Respuesta del Sistema
1 El administrador accede al vínculo: "Gestionar Datos".	1.1 El sistema le muestra un formulario donde le pide que introduzca el nombre, el número de solapín o el login del usuario del cual desea gestionar sus datos.
2 El administrador introduce el nombre, el número de solapín o el login del usuario deseado y da clic en el botón <b>Buscar</b> .	2.1 El sistema verifica que los campos obligatorios (al menos uno de los 3) estén llenos.
	2.2 El sistema invoca el caso de uso incluido Buscar Usuarios y muestra los usuarios encontrados de acuerdo al criterio de búsqueda especificado.
	2.3 El sistema da la posibilidad de seleccionar uno de los usuarios listados y presenta al administrador 3 acciones a realizar sobre el mismo: <ul style="list-style-type: none"> <li><b>a) Eliminar Usuario.</b></li> <li><b>b) Ver Detalles.</b></li> <li><b>c) Modificar Datos de Usuario.</b></li> </ul>
3 El administrador selecciona el usuario al cual desea gestionar datos y: <ul style="list-style-type: none"> <li>a) Si desea que el mismo sea eliminado de la BD elige el vínculo <b>Eliminar Usuario</b> para pasar a la sección "Eliminar Usuario".</li> <li>b) Si solo necesita mostrar los datos del mismo elige la opción <b>Ver Detalles</b> del</li> </ul>	

<p>usuario para ir a la sección “Ver Detalles”.</p> <p>c) Si lo que desea es actualizar datos del mismo, elige la opción <b>Modificar Datos de Usuario</b>, para ir a la sección de “Modificar Datos de Usuario”.</p>	
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 a) El sistema muestra al administrador un mensaje de aviso: “Teclee un criterio de búsqueda”.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.2 a) Si el caso de uso no devuelve ningún usuario dado el(los) criterio(s) de búsqueda indicado(s) el sistema muestra un mensaje de aviso indicando: “No se encontraron usuarios dado(s) el(los) criterio(s) de búsqueda indicado(s)”.
	2.2 b) El sistema da la posibilidad de realizar otra búsqueda.
<b>Sección “Eliminar Usuario”</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 El sistema pide comprobar la acción a realizar por parte del administrador mediante un mensaje: “¿Está seguro de querer eliminar el usuario?”.
4 El administrador acepta la acción mostrada en el mensaje.	4.1 El sistema elimina el usuario de la BD, muestra al administrador un mensaje: “El usuario ha sido eliminado correctamente”.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
4 a) El administrador cancela la acción mostrada en el mensaje.	4.1 El sistema da la posibilidad de realizar otra búsqueda.
<b>Sección “Ver Detalles”</b>	



Acciones del Actor	Respuesta del Sistema
	3.1 El sistema invoca al caso de uso <i>Mostrar Datos de Usuario</i> para el usuario seleccionado.
<b>Sección “Modificar Datos de Usuario”</b>	
Acciones del Actor	Respuesta del Sistema
	3.1 El sistema invoca al caso de uso <i>Mostrar Datos de Usuario</i> para el usuario seleccionado y muestra los datos al administrador, dando la posibilidad de editarlos (eliminarlos o modificarlos).
4 El administrador realiza las ediciones deseadas y pulsa el botón “ <b>Actualizar</b> ”.	4.1 El sistema pide comprobar la acción a realizar por parte del administrador mediante un mensaje: “¿Está seguro de querer modificar el (los) dato(s)?”.
5 El administrador acepta la acción mostrada en el mensaje.	5.1 El sistema actualiza los datos del usuario en la BD, muestra al administrador un mensaje: “Los datos del usuario han sido modificados correctamente” y finaliza el caso de uso.
<b>Curso Alterno de los eventos:</b>	
Acciones del Actor	Respuesta del Sistema
5 a) El administrador cancela la acción mostrada en el mensaje.	5.1 El sistema da la posibilidad de realizar otra búsqueda.
<b>Prioridad:</b>	<b>Crítico</b>
<b>Prototipo</b>	

2.5.5.2 CU2\_Buscar Usuarios

Caso de Uso:	CU2 Buscar Usuarios
<b>Actores:</b>	Administrador
<b>Propósito:</b>	Llevar a cabo la búsqueda de usuarios por criterio de selección.
<b>Resumen:</b>	El caso de uso es un caso de uso incluido, se inicializa cuando el caso de uso Gestionar Datos de Usuario necesita responder las peticiones de búsqueda de usuarios dado un criterio de búsqueda determinado. El caso de uso termina

	cuando se devuelven en un listado todos los usuarios registrados en el sistema que cumplen los requisitos de búsqueda especificados.
<b>Referencia:</b>	RF8
<b>Precondiciones:</b>	Debe recibir un criterio de búsqueda.
<b>Poscondiciones:</b>	
<b>Flujo normal de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1.1 El sistema recoge el(los) criterio(s) de búsqueda devuelto(s) por el caso de uso que lo invoca.
	1.2 El sistema realiza la búsqueda de usuarios correspondiente.
	1.3 El sistema devuelve un listado de todos los usuarios registrados en el sistema que cumplen los requisitos de búsqueda especificados y finaliza el caso de uso.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1.3 a) El sistema no encuentra ningún usuario dado el(los) criterio(s) de búsqueda especificado(s) y devuelve un mensaje de error: "No se encontraron usuarios con estas características".
	1.4 El sistema da la posibilidad de realizar otra búsqueda.
<b>Prioridad:</b>	<b>Crítico</b>
<b>Prototipo</b>	

2.5.5.3 CU3\_Mostrar Datos de Usuario

<b>Caso de Uso:</b>	<b>CU3 Mostrar Datos de Usuario</b>
<b>Actores:</b>	Usuario Dominio

<b>Propósito:</b>	Llevar a cabo la acción de ver detalles de su propia información
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario del dominio solicita visualizar sus datos. El caso de uso finaliza cuando son mostrados a dicho usuario sus datos.
<b>Referencia:</b>	RF5, RF9
<b>Precondiciones:</b>	El usuario debe estar autenticado
<b>Poscondiciones:</b>	
<b>Flujo normal de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1 El usuario del dominio pulsa el vínculo " <b>Ver Detalles</b> ".	1.1 El sistema busca los datos del usuario dado el número de solapín del usuario autenticado y son mostrados: <ul style="list-style-type: none"> <li>• ID expediente (número de solapín).</li> <li>• Foto.</li> <li>• Nombre(s) del usuario.</li> <li>• Apellidos del usuario.</li> <li>• Usuario (login).</li> <li>• CI.</li> <li>• Tipo de persona.</li> <li>• Cuota de navegación.</li> <li>• Cuota usada.</li> <li>• Nivel de navegación.</li> <li>• Capacidad del inbox.</li> </ul>
<b>Prioridad:</b>	<b>Crítico</b>
<b>Prototipo</b>	

2.5.5.4 CU4\_Generar Reporte

<b>Caso de Uso:</b>	<b>CU4 Generar Reporte</b>
<b>Actores:</b>	Administrador

<b>Propósito:</b>	Generar reportes de las personas.	
<b>Resumen:</b>	El caso de uso comienza cuando el administrador necesita conocer reportes de los diferentes usuarios, pudiendo ser: Personas sin login, Personas con la cuota deshabilitada, Personas habilitadas en la fachada de autenticación. El caso de uso finaliza cuando es mostrado un reporte.	
<b>Referencia:</b>	RF10, RF11, RF12	
<b>Precondiciones:</b>	El usuario autenticado tiene que ser administrador del sistema.	
<b>Poscondiciones:</b>		
<b>Flujo normal de los eventos:</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1 El administrador accede al vínculo " <b>Generar Reportes</b> ".	1.1 El sistema muestra las diferentes opciones de reportes y permite seleccionar alguno de ellos: <b>a) Usuarios sin login.</b> <b>b) Usuarios con la cuota deshabilitada.</b> <b>c) Usuarios habilitados en la fachada de autenticación.</b>	
2 Si el administrador desea ver: a) Un reporte de los usuarios sin login, selecciona <b>Usuarios sin login</b> como reporte a generar, para pasar a la sección "Usuarios sin login". b) Un reporte de los usuarios que poseen la cuota deshabilitada, selecciona <b>Usuarios con la cuota deshabilitada</b> como reporte a generar, para pasar a la sección "Usuarios con la cuota deshabilitada". c) Un reporte de los usuarios que han sido baja de la		

<p>universidad selecciona <b>Usuarios habilitados en la fachada de autenticación</b> como reporte a generar, para pasar a la sección “Usuarios habilitados en la fachada de autenticación”.</p>	
<b>Sección “Usuarios sin login”</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 El sistema muestra un listado de las personas que no tienen login.
	4.2 El sistema da la posibilidad de solicitar un nuevo reporte (volviendo a la vista anterior) o gestionar datos de alguno de los usuarios listados (invocando el caso de uso Gestionar Datos de Usuario).
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 El sistema muestra el mensaje: “No se encontraron personas sin login.”
<b>Sección “Usuarios con la cuota deshabilitada.”</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 El sistema muestra un listado de las personas que tienen la cuota deshabilitada.
	4.2 El sistema da la posibilidad de solicitar un nuevo reporte (volviendo a la vista anterior) o gestionar datos de alguno de los usuarios listados (invocando el caso de uso Gestionar Datos de Usuario).
<b>Curso Alternativo de los eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 El sistema muestra el mensaje: “No se encontraron personas con la cuota deshabilitada”.
<b>Sección “Usuarios habilitados en la fachada de autenticación”</b>	

Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra un listado de las personas autenticadas en la fachada de autenticación.
	4.2 El sistema da la posibilidad de solicitar un nuevo reporte (volviendo a la vista anterior) o gestionar datos de alguno de los usuarios listados (invocando el caso de uso Gestionar Datos de Usuario); y finaliza el caso de uso.
<b>Curso Alterno de los eventos:</b>	
Acciones del Actor	Respuesta del Sistema
	4.1 El sistema muestra el mensaje: “No se encontraron personas autenticadas en la fachada de autenticación”.
<b>Prioridad:</b>	<b>Crítico</b>
<b>Prototipo</b>	

2.5.5.5 CU5\_Brindar Datos de Usuario

Caso de Uso:	CU5 Brindar Datos de Usuario
<b>Actores:</b>	Sistema UCI
<b>Propósito:</b>	Obtener información sobre los datos de los usuarios del dominio, registrados en el sistema.
<b>Resumen:</b>	El caso de uso inicia cuando el Sistema UCI solicita algún servicio de los brindados por el sistema. El sistema obtiene la información que es devuelta al sistema externo, finalizando así el caso de uso.
<b>Referencia:</b>	RF6
<b>Precondiciones:</b>	El perfil del Sistema UCI autenticado debe tener permiso para utilizar el servicio de información solicitado.
<b>Poscondiciones:</b>	
<b>Flujo normal de los eventos:</b>	
Acciones del Actor	Respuesta del Sistema

<p>1 El Sistema UCI solicita:</p> <ul style="list-style-type: none"> <li>a) Si solicita ver todos los usuarios que coincidan con una cadena de búsqueda específica ver sección “Obtener usuarios dada cadena de búsqueda”.</li> <li>b) Si solicita ver todos los usuarios habilitados en la fachada de autenticación ver sección “Obtener usuarios”.</li> <li>c) Si solicita ver la información de los usuarios dado un filtro ver sección “Buscar usuarios dado filtro”.</li> <li>d) Si solicita ver la información de los usuarios dado una lista de idExpedientes (número de solapín) ver la sección “Obtener usuarios dada lista”.</li> <li>e) Si solicita ver el usuario dado IdExpediente ver sección “Obtener usuario dado IdExpediente”.</li> <li>f) Si solicita ver el usuario dado el nombre de usuario ver sección “Obtener usuario dado nombre de usuario”.</li> <li>g) Si solicita ver los dominios registrados ver sección “Obtener dominios”.</li> <li>h) Si solicita verificar si existe un dominio ver sección “Existe</li> </ul>	
--	--

<p>dominio”.</p> <p>i) Si solicita obtener dominio local ver sección “Obtener dominio local”.</p> <p>j) Si solicita ver datos de un dominio dado un nombre de dominio ver sección “Obtener dominio dado nombre dominio”.</p> <p>k) Si solicita verificar si es posible o no la autenticación de un usuario dado dominio, nombre de usuario y contraseña ver sección “Autenticar usuario LDAP”.</p> <p>l) Si solicita autenticar un usuario dado dominio, nombre de usuario y contraseña ver sección “Autenticar usuario”.</p>	
<b>Sección “Obtener usuarios dada cadena de búsqueda”</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2 El sistema UCI especifica la cadena de búsqueda; ya sea nombre de usuario, IdExpediente y/o CorreoE.	2.1 El sistema verifica que la cadena de búsqueda sea válida.
	2.2 El sistema confecciona el listado de todos los usuarios correspondientes al criterio de búsqueda especificado.
	2.3 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Curso Alterno de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 El sistema informa al sistema UCI que la cadena de búsqueda entrada no es válida.
<b>Sección “Obtener usuarios”</b>	



Acciones del Actor	Respuesta del Sistema
	2.1 El sistema confecciona el listado de todos los Usuarios habilitados en la fachada de autenticación.
	2.2 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Sección “Buscar usuarios dado filtro”</b>	
Acciones del Actor	Respuesta del Sistema
2 El sistema UCI especifica el filtro de selección; ya sea por nombre, por valor u ordenado.	2.1 El sistema confecciona el listado de todos los usuarios registrados en la fachada de autenticación dado el filtro especificado.
	2.2 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Sección “Obtener usuarios dada lista”</b>	
Acciones del Actor	Respuesta del Sistema
2 El sistema UCI especifica una lista de idExpedientes.	2.1 El sistema verifica que la lista de idExpedientes sea válida y tenga el formato requerido.
	2.2 El sistema confecciona el listado de todos los usuarios registrados en la fachada de autenticación dada la lista de idExpedientes especificada.
	2.3 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Curso Alternativo de los eventos:</b>	
Acciones del Actor	Respuesta del Sistema
	2.1 El sistema informa al sistema UCI que la lista de idExpedientes entrada no es válida.
<b>Sección “Obtener Usuario Dado IdExpediente”</b>	
Acciones del Actor	Respuesta del Sistema
2 El sistema UCI especifica un IdExpediente.	2.1 El sistema verifica que el idExpediente sea válido.
	2.2 El sistema obtiene la información solicitada.
	2.3 El sistema devuelve la respuesta al sistema externo,

	finalizando el caso de uso.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 El sistema informa al sistema UCI que el idExpediente entrado no es válido.
<b>Sección "Obtener usuario dado nombre de usuario"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2 El sistema UCI especifica un nombre de usuario.	2.1 El sistema verifica que el nombre de usuario sea válido.
	2.2 El sistema obtiene la información solicitada.
	2.3 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 El sistema informa al sistema UCI que el nombre de usuario entrado no es válido.
<b>Sección "Obtener dominios"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 El sistema confecciona un listado de todos los dominios registrados en la fachada de autenticación
	2.2 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Sección "Existe dominio"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2 El sistema UCI especifica un dominio.	2.1 El sistema verifica si el dominio especificado es válido.
	2.2 El sistema obtiene la información solicitada.
	2.3 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

	2.1 El sistema informa al sistema UCI que el dominio entrado no es válido.
<b>Sección "Obtener dominio local"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 El sistema obtiene la información solicitada.
	2.2 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Sección "Obtener dominio dado nombre dominio"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2 El sistema UCI especifica un nombre de dominio.	2.1 El sistema verifica si el dominio especificado es válido.
	2.2 El sistema obtiene la información solicitada.
	2.3 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 El sistema informa al sistema UCI que el dominio entrado no es válido.
<b>Sección "Autenticar usuario LDAP"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
2 El sistema UCI especifica dominio, nombre de usuario y contraseña.	2.1 El sistema verifica que el dominio, el nombre de usuario y la contraseña sean válidos.
	2.2 El sistema obtiene la información solicitada.
	2.3 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 El sistema informa al sistema UCI que los datos entrados no son válidos.
<b>Sección "Autenticar usuario"</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

2 El sistema UCI especifica dominio, nombre de usuario y contraseña.	2.1 El sistema verifica que el dominio, el nombre de usuario y la contraseña sean válidos.
	2.2 El sistema obtiene la información solicitada.
	2.3 El sistema devuelve la respuesta al sistema externo, finalizando el caso de uso.
<b>Curso Alternativo de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.1 El sistema informa al sistema UCI que los datos entrados no son válidos.
<b>Prioridad:</b>	<b>Crítico</b>
<b>Prototipo</b>	

2.5.5.6 CU6\_ Sincronizar/Actualizar Datos de Usuario

<b>Caso de Uso:</b>	<b>CU6 Sincronizar/Actualizar Datos de Usuario</b>	
<b>Actores:</b>	Reloj	
<b>Propósito:</b>	Sincronizar los datos primarios de las personas registrados en los SAIU (Sistemas de Almacenamiento de Información de Usuarios) y los datos telemáticos registrados en el DA (Directorio Activo), y actualizarlos en el sistema.	
<b>Resumen:</b>	El caso de uso es inicializado automáticamente por el sistema cada 24 horas. El sistema lleva a cabo la sincronización/actualización de los datos de las personas.	
<b>Referencia:</b>	RF4	
<b>Precondiciones:</b>	El sistema debe tener permiso para modificar datos sobre su BD.	
<b>Poscondiciones:</b>	Se actualizan los datos reales de las personas.	
<b>Flujo normal de los eventos:</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1 El Reloj inicia el caso de uso automáticamente cada 24 horas.	1.1 El sistema consulta y obtiene los datos reales de todas las personas registrados en los SAIU. 1.2 El sistema compara la información obtenida de esta consulta con la almacenada hasta el momento:	

	<ul style="list-style-type: none"> <li>a) Si encuentra alguna persona que no tenga registrada, la adiciona al sistema.</li> <li>b) Si no encuentra alguna persona de las que tiene registrada, la elimina del sistema.</li> <li>c) Actualiza en el sistema los datos primarios que hayan sido modificados de las personas que tiene registradas.</li> <li>d) Actualiza en el sistema los datos telemáticos que hayan sido modificados de las personas que tiene registradas.</li> </ul>
<b>Prioridad:</b>	<b>Crítico</b>
<b>Prototipo</b>	

2.5.5.7 CU7\_Autenticar Usuario

<b>Caso de Uso:</b>	<b>CU7 Autenticar Usuario</b>	
<b>Actores:</b>	Usuario	
<b>Propósito:</b>	Llevar a cabo la acción de validar datos de acceso para mantener la integridad de los datos.	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario entra sus datos para acceder al sistema a realizar las acciones que le son permitidas según su nivel de acceso. El sistema verifica los datos dejando abierto el acceso a las funcionalidades según corresponda con el usuario autenticado, finalizando el caso de uso.	
<b>Referencia:</b>	RF7	
<b>Precondiciones:</b>		
<b>Poscondiciones:</b>	Queda autenticado el usuario como Usuario Dominio o Administrador para realizar las acciones correspondientes a cada uno, que son habilitadas por el sistema.	
<b>Flujo normal de los eventos:</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	

1 El usuario intenta acceder al sistema.	1.1 El sistema le muestra un formulario donde le pide entrar usuario, contraseña y dominio.
2 El usuario inserta los datos para la autenticación (usuario, contraseña y dominio (opcional)), y da clic en el botón <b>Entrar</b> .	2.1 El sistema comprueba que los datos estén correctos y estén llenos los campos obligatorios (usuario y contraseña).
	2.2 Si los datos están correctos y completos, el sistema verifica tipo de usuario.
	2.3 Si el tipo de usuario es "Administrador" el sistema muestra una interfaz para acceder a todas las funcionalidades del sistema, finalizando el caso de uso.
<b>Curso Alterno de los eventos:</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2.2 a) Si el sistema reconoce error en los datos, muestra el mensaje de error: "Introduzca los datos correctos".
	2.2 b) Si el sistema reconoce que faltan datos, muestra el mensaje de error "Teclee usuario y contraseña".
	2.3 a) Si el tipo de usuario es "Usuario Dominio" el sistema muestra una interfaz para acceder a las funcionalidades del sistema correspondientes a este tipo de usuario, finalizando el caso de uso.
<b>Prioridad:</b>	<b>Crítico</b>
<b>Prototipo</b>	

## 2.6 Conclusiones del Capítulo

En este capítulo se presentaron las características fundamentales del sistema. En tal sentido y a partir del análisis de la situación del negocio actual, se capturaron inicialmente los tipos de objetos más importantes en el contexto del sistema a través del Modelo de Objetos del Dominio, definiendo previamente cada uno de los conceptos relacionados en el mismo.

Se realizó también un levantamiento de requisitos que permitió identificar cuidadosamente los requerimientos funcionales y no funcionales que debe presentar la aplicación que se propone una vez desarrollada. Se determinaron y describieron los actores que participan en el sistema a desarrollar. Se agruparon las funcionalidades del sistema por Casos de Uso, desarrollando las correspondientes descripciones para representar las especificaciones del sistema y representando estas funciones (en relación con los actores) mediante un conjunto de Diagramas de Casos de Uso.

## CAPÍTULO III: ANÁLISIS Y DISEÑO

### 3.1 Introducción

En el presente capítulo se detalla todo lo referente al tema de análisis y diseño de la solución propuesta. Se describen las clases necesarias para dar cumplimiento a los requerimientos del sistema, así como los diagramas necesarios para una mayor comprensión de la aplicación. Se obtiene a través del análisis una visión del sistema que se preocupa de ver qué hace el mismo, y a través del diseño se muestra cómo cumple el sistema sus objetivos.

### 3.2 Diagrama de Clases

En esta etapa de Análisis y Diseño el Diagrama de Clases es el diagrama principal para el sistema.

Un diagrama de clases es una colección de elementos de un modelo estático declarativo, tales como clases, interfaces y sus relaciones; conectados como un grafo entre sí y con sus contenidos. En él se especifica la estructura de clases del sistema, con relaciones entre clases y estructuras de herencia.

Durante el análisis, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama y se modifica para satisfacer los detalles de las implementaciones.

### 3.3 Análisis del Sistema

#### 3.3.1 Definición del Modelo de Análisis

El modelo de análisis es la primera representación técnica de un sistema. Debe lograr tres objetivos primarios:

- (1) Describir lo que requiere el cliente.
- (2) Establecer una base para la creación de un diseño de software.
- (3) Definir un conjunto de requisitos que se pueda validar una vez que se construye el software.

#### 3.3.2 Clases del Análisis

Existen tres estereotipos de clases estandarizados en UML y se utilizan para ayudar a los desarrolladores a distinguir el ámbito de las diferentes clases. Estas clases son:



- Interfaz: Modelan la interacción entre el sistema y sus actores.
- Control: Coordinan la realización de un caso de uso, controlando las actividades de los objetos que implementan su funcionalidad.
- Entidad: Modelan información que posee larga vida y que por lo general es persistente.

A continuación se presentan las clases del análisis identificadas, así como el tipo de cada una de ellas dentro del sistema:

Nombre de la Clase	Tipo de Clase
CI_Buscar Usuario	Interfaz
CI_Mostrar Datos Usuario	Interfaz
CI_Modificar Datos Usuario	Interfaz
CI_Generar Reporte	Interfaz
CI_Autenticarse	Interfaz
CI_Administrador	Interfaz
CI_UsuarioDominio	Interfaz
CI_Fachada de Servicios	Interfaz
CC_Gestionar Datos Usuario	Control
CC_Generar Reporte	Control
CC_Autenticar Usuario	Control
CC_Brindar Datos de Usuario	Control
CC_Sincronizar/Actualizar BD	Control
CE_Dominio	Entidad
CE_Usuario	Entidad
CE_Administrador	Entidad

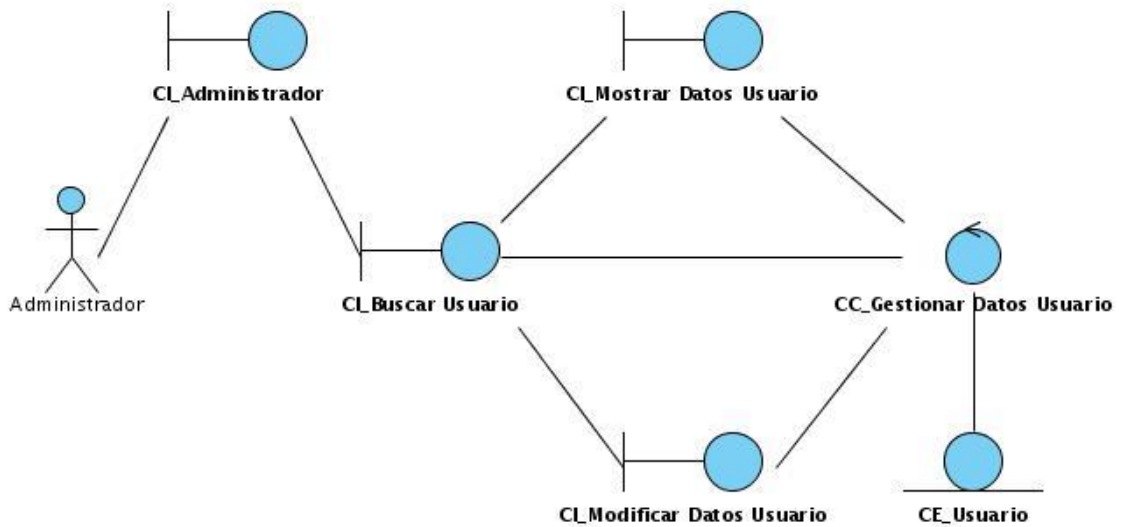
**Tabla 3.1** Descripción de las Clases del Análisis

### 3.3.3 Diagramas de Clases del Análisis

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. En este tipo de diagrama se representan las clases y sus relaciones. Ellos representan una vista estática del sistema.

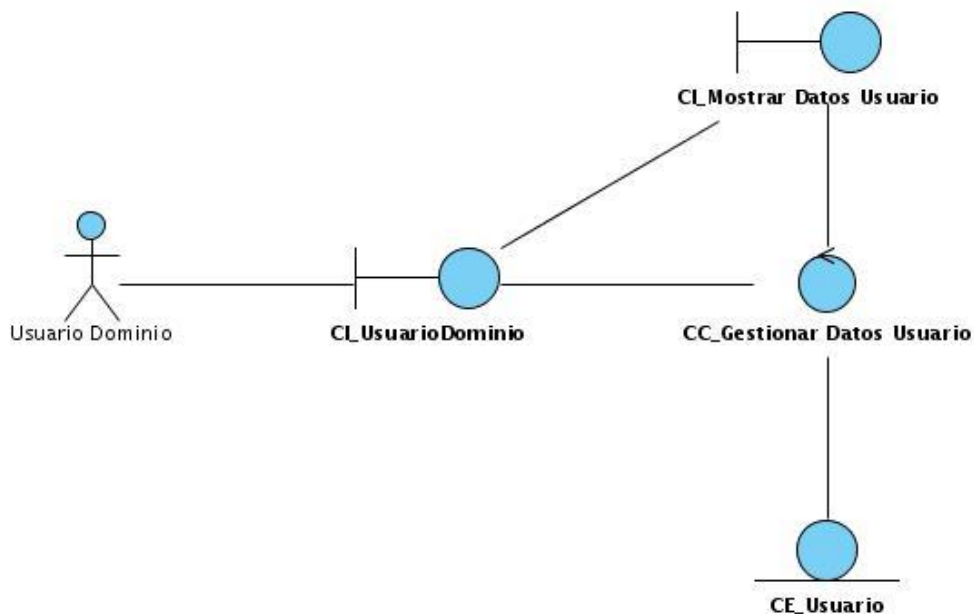
A continuación se presentan los Diagramas de Clases del Análisis (DCA) por cada caso de uso (CU):

3.3.3.1 DCA del CU: Gestionar Datos de Usuario



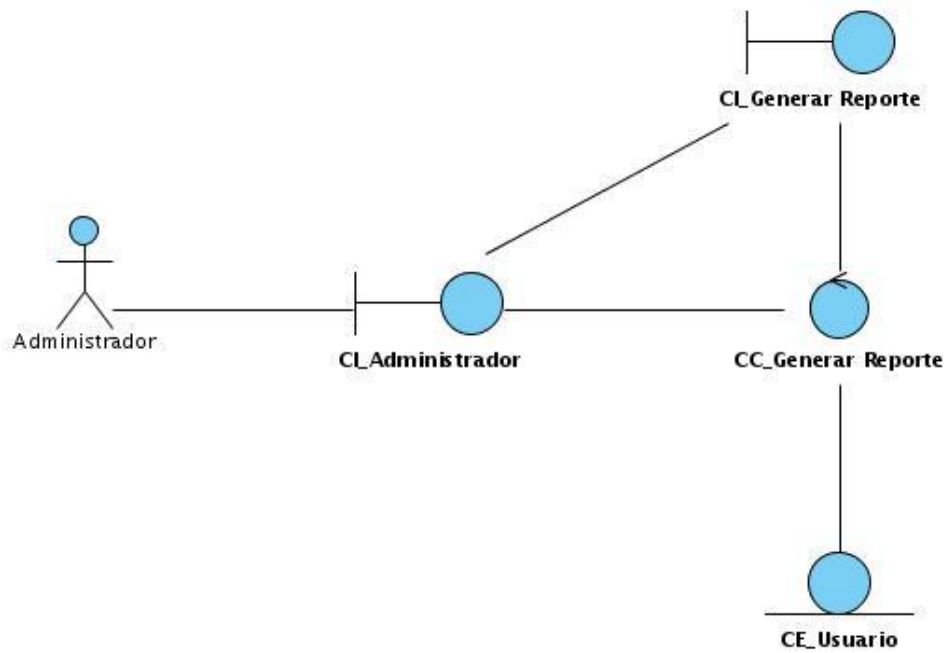
**Figura 3.1:** Diagrama de clases del análisis “CU: Gestionar Datos de Usuario”

3.3.3.2 DCA del CU: Mostrar Datos de Usuario



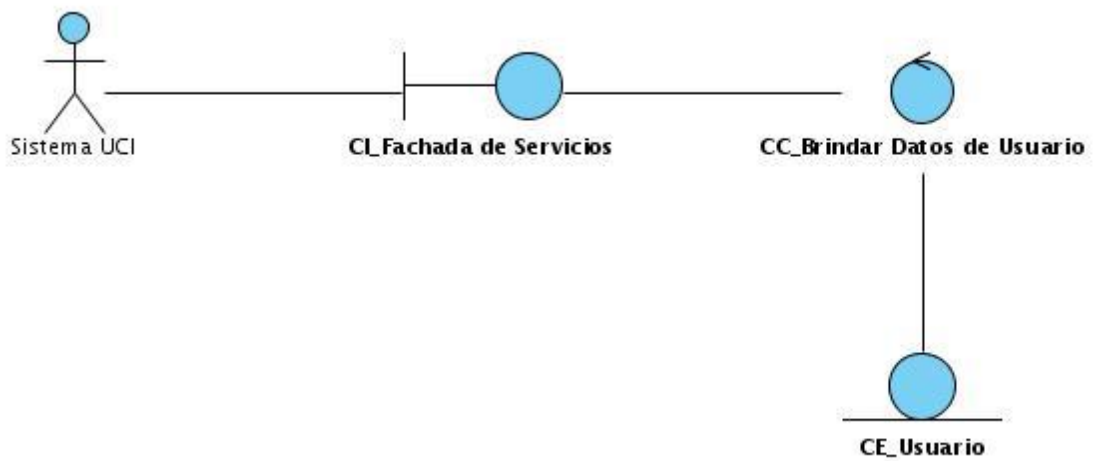
**Figura 3.2:** Diagrama de clases del análisis “CU: Mostrar Datos de Usuario”

3.3.3.3 DCA del CU: Generar Reporte



**Figura 3.3:** Diagrama de clases del análisis “CU: Generar Reporte”

3.3.3.4 DCA del CU: Brindar Datos de Usuario



**Figura 3.4:** Diagrama de clases del análisis “CU: Brindar Datos de Usuario”

3.3.3.5 DCA del CU: Sincronizar/Actualizar Datos de Usuario

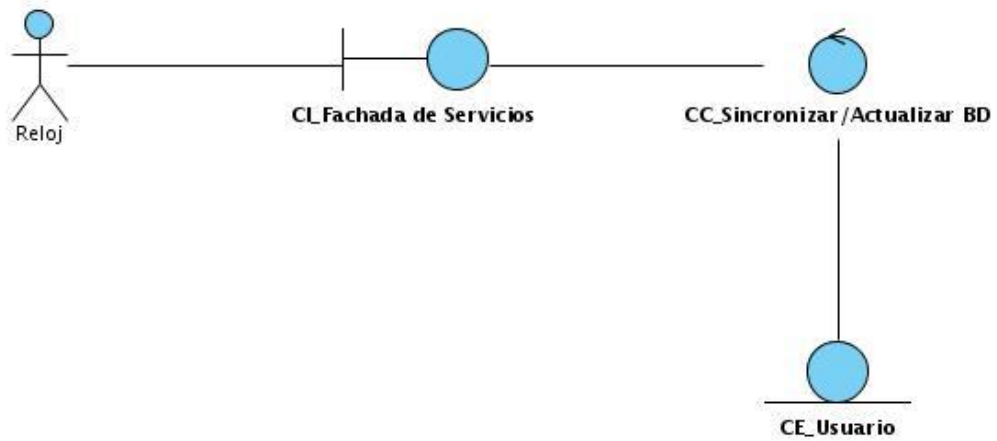


Figura 3.5: Diagrama de clases del análisis “CU: Sincronizar/Actualizar Datos de Usuario”

3.3.3.6 DCA del CU: Autenticar Usuario

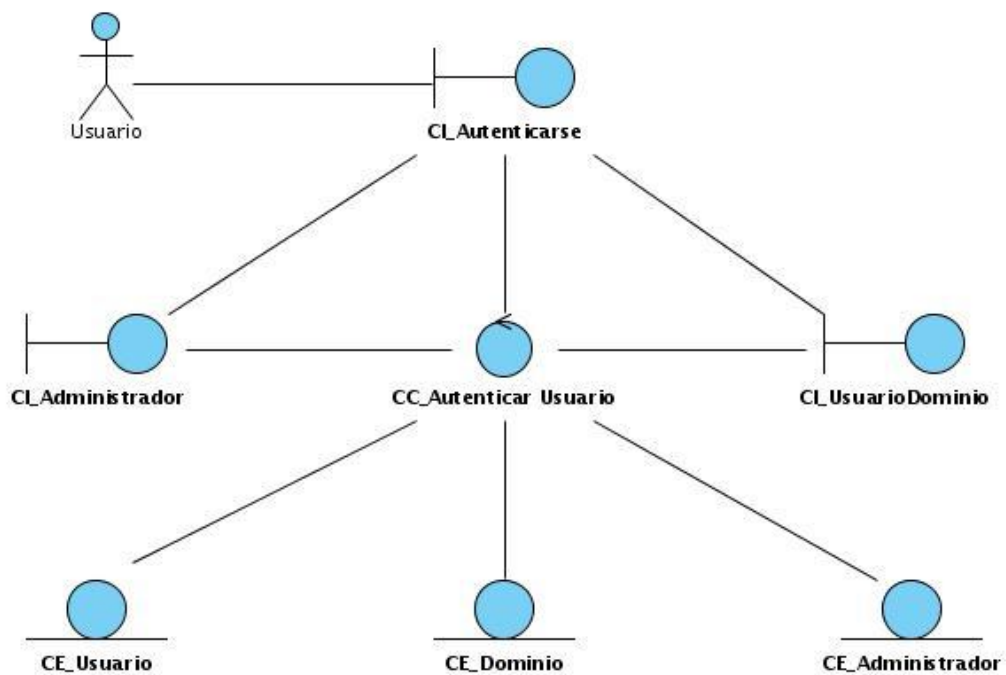


Figura 3.6: Diagrama de clases del análisis “CU: Autenticar Usuario”

### 3.4 Diseño del Sistema

#### 3.4.1 Definición de Modelo de Diseño

El modelo de diseño entrega un enfoque global e integrador de la gestión estratégica y ejecución de negocios. Se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física. Transforma elementos estructurales de la arquitectura del programa. Dentro del diseño es donde se fomenta la calidad del Proyecto, es la única manera de materializar con precisión los requerimientos del cliente.

#### 3.4.2 Diagramas de Clases del Diseño

A continuación se presentan los Diagramas de Clases del Diseño (DCD) por cada caso de uso (CU):

##### 3.4.2.1 DCD del CU: Gestionar Datos de Usuario

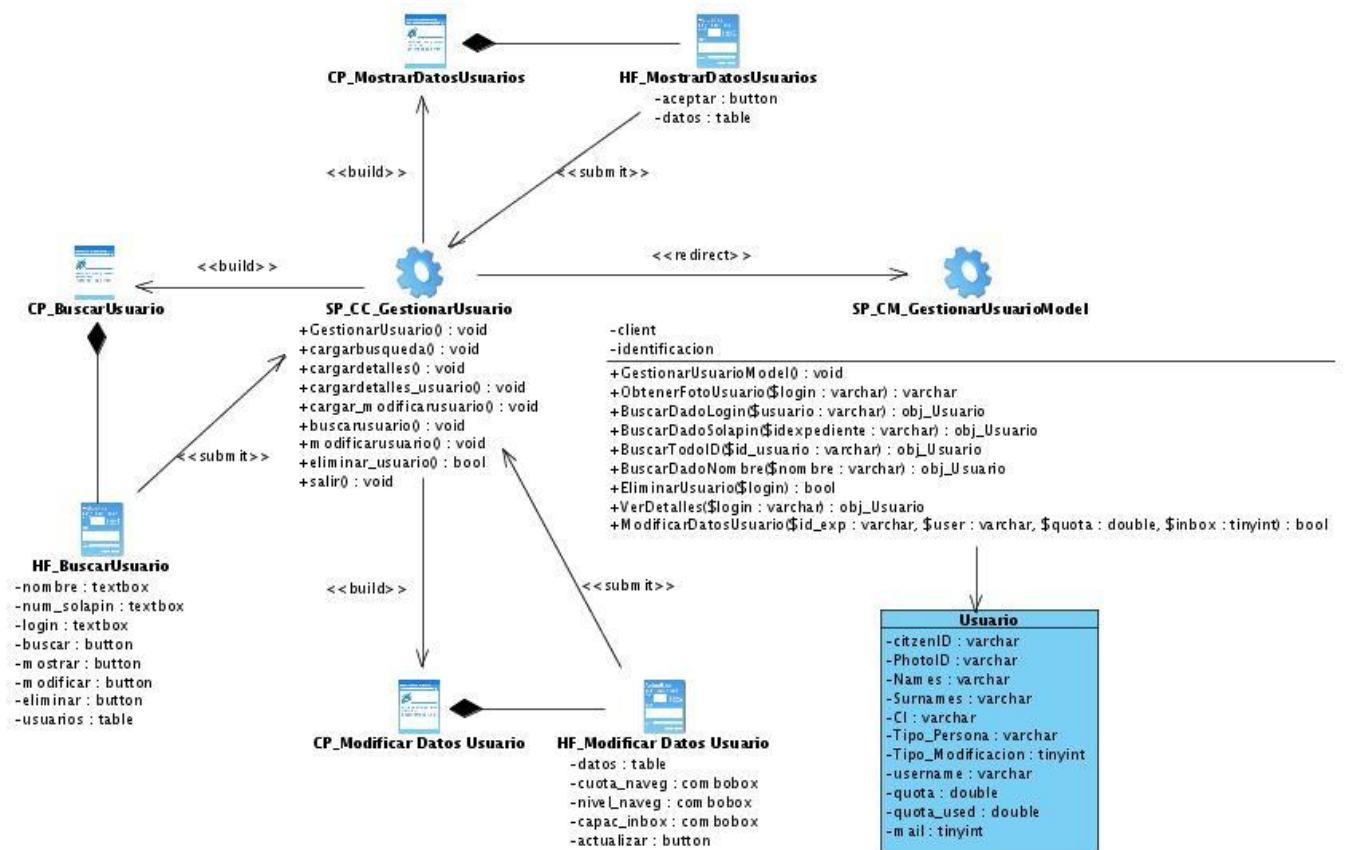


Figura 3.7: Diagrama de clases del diseño “CU: Gestionar Datos de Usuario”

3.4.2.2 DCD del CU: Mostrar Datos de Usuario

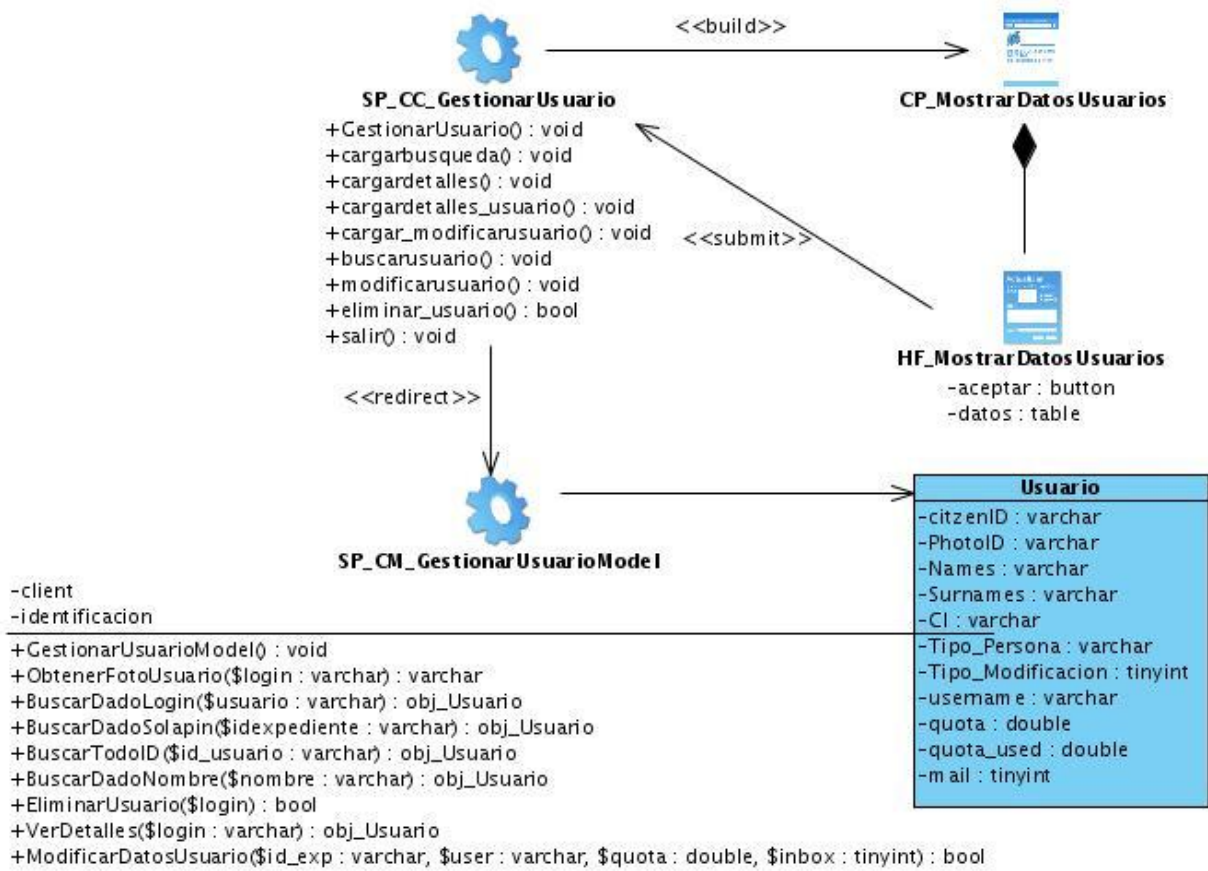


Figura 3.8: Diagrama de clases del diseño “CU: Mostrar Datos de Usuario”

3.4.2.3 DCD del CU: Generar Reporte

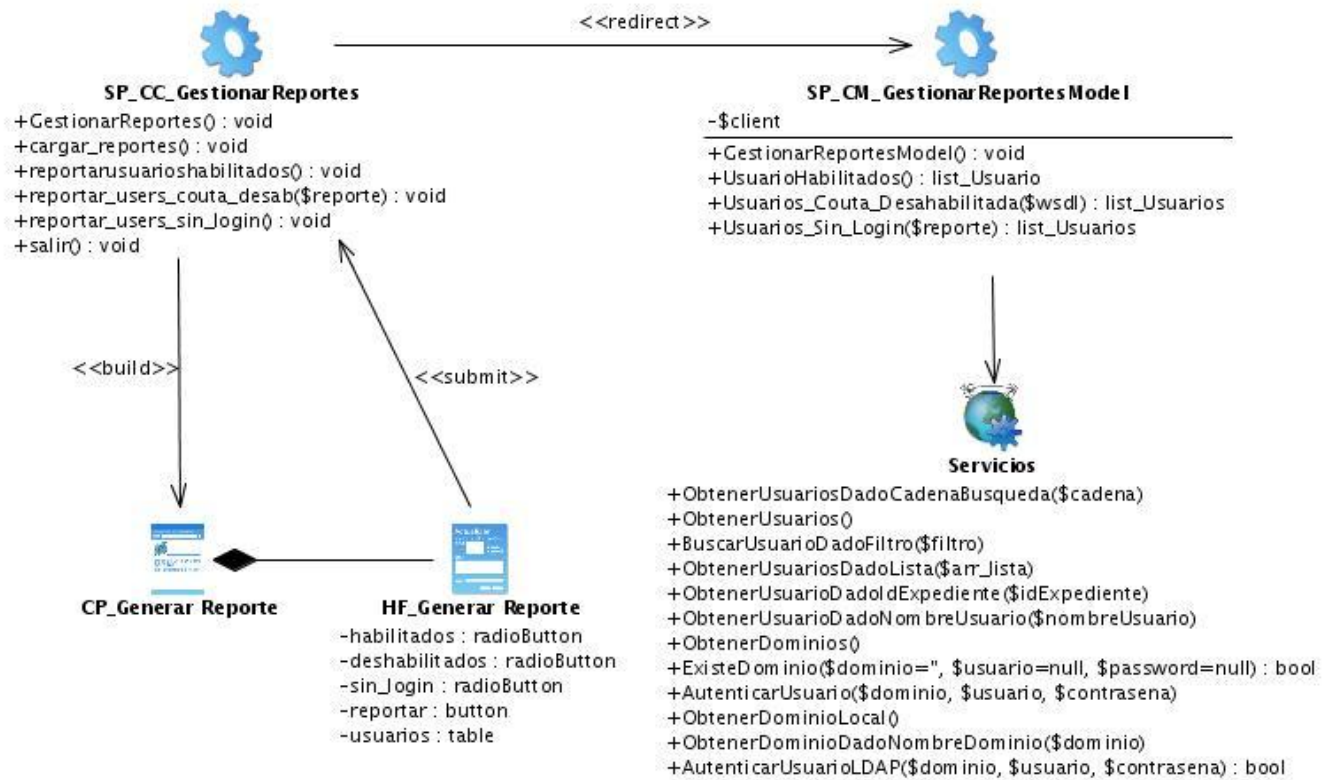


Figura 3.9: Diagrama de clases del diseño "CU: Generar Reporte"

3.4.2.4 DCD del CU: Brindar Datos de Usuario

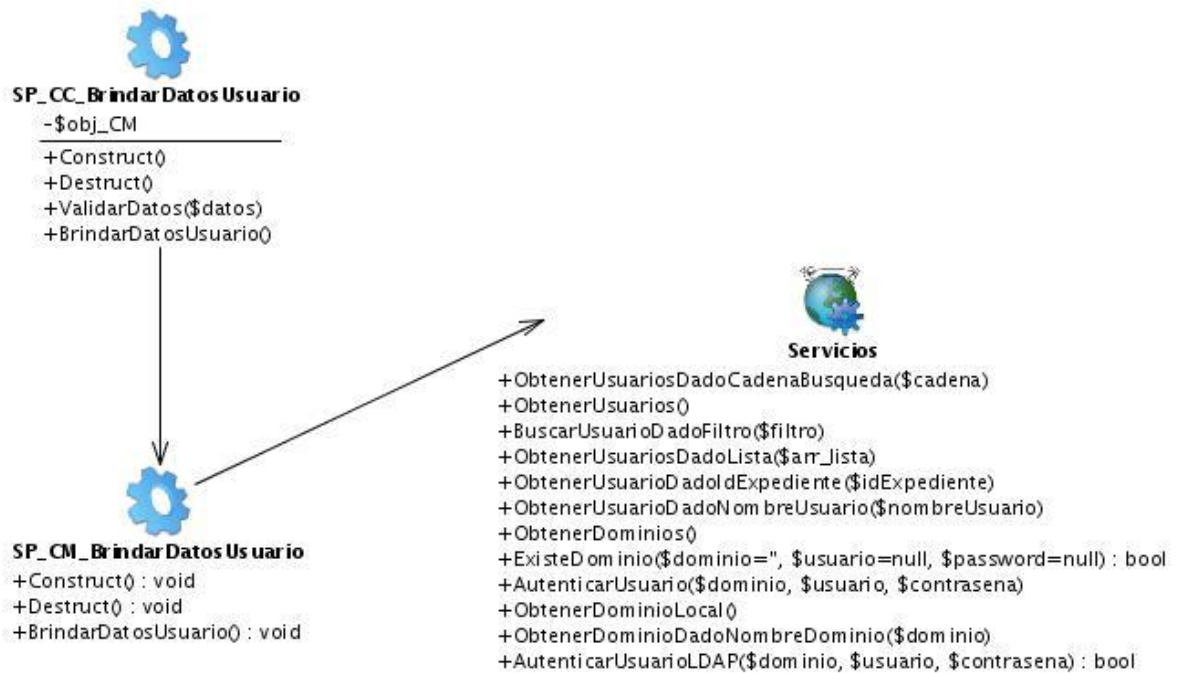


Figura 3.10: Diagrama de clases del diseño “CU: Brindar Datos de Usuario”

3.4.2.5 DCD del CU: Sincronizar/Actualizar Datos de Usuario

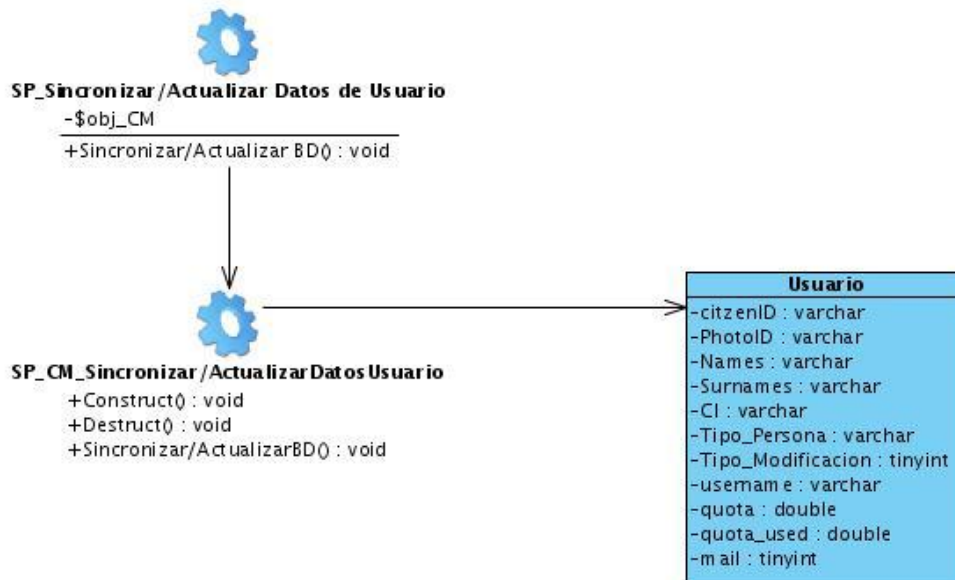


Figura 3.11: Diagrama de clases del diseño “CU: Sincronizar/Actualizar Datos de Usuario”



3.4.2.6 DCD del CU: Autenticar Usuario

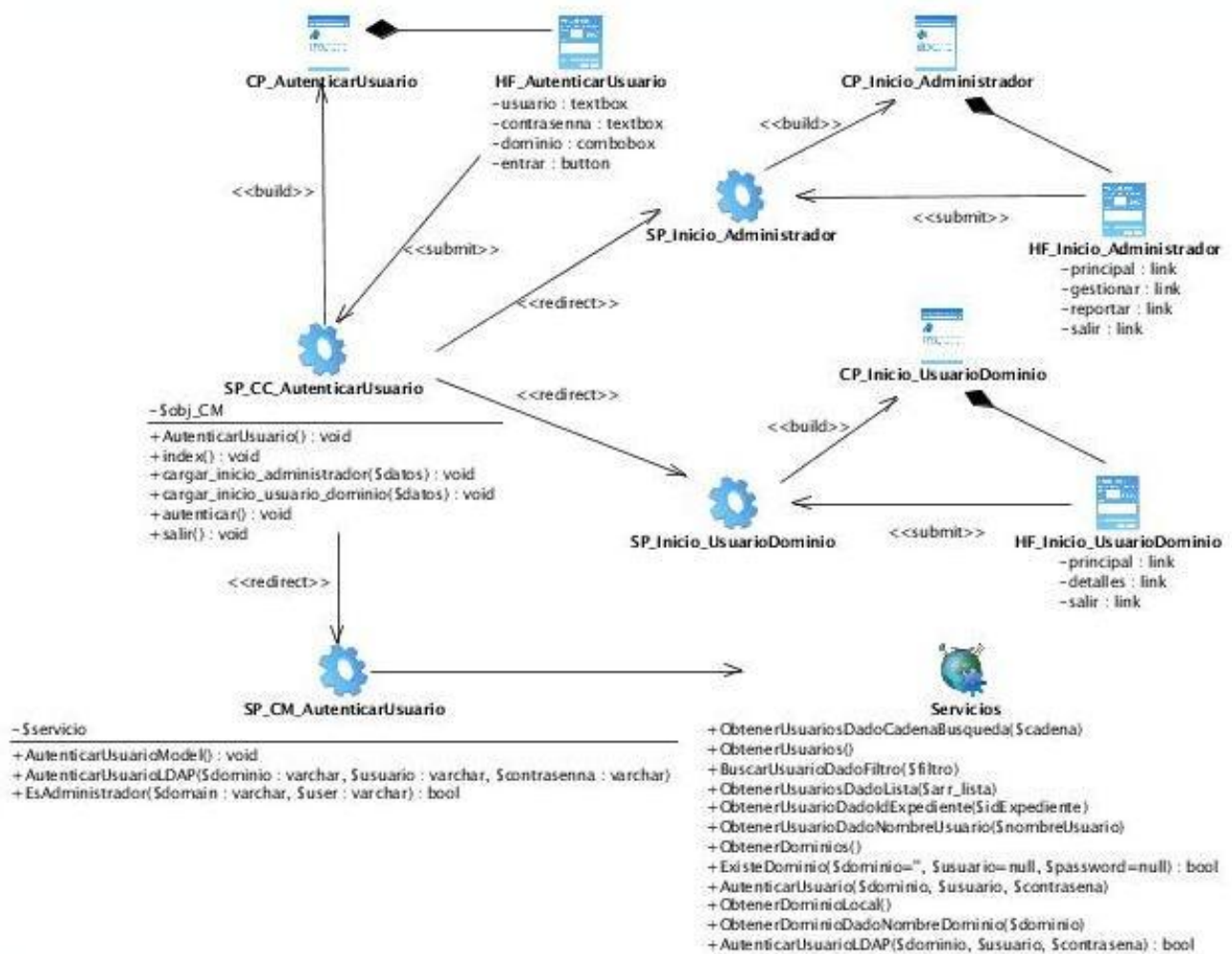


Figura 3.12: Diagrama de clases del diseño "CU: Autenticar Usuario"

3.4.3 Diagramas de Interacción

3.4.3.1 Diagrama de Secuencia

El *Diagrama de Secuencia* es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia se modela para cada caso de uso, este contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos. Cada objeto viene mostrado por una barra vertical que es la que da idea de tiempo, la línea vertical representa el paso del tiempo (de arriba hacia abajo).

Los diagramas de secuencia de los CU más significativos del sistema se muestran en los anexos. VER ANEXOS (1-6)

#### 3.4.3.2 Diagrama de Colaboración

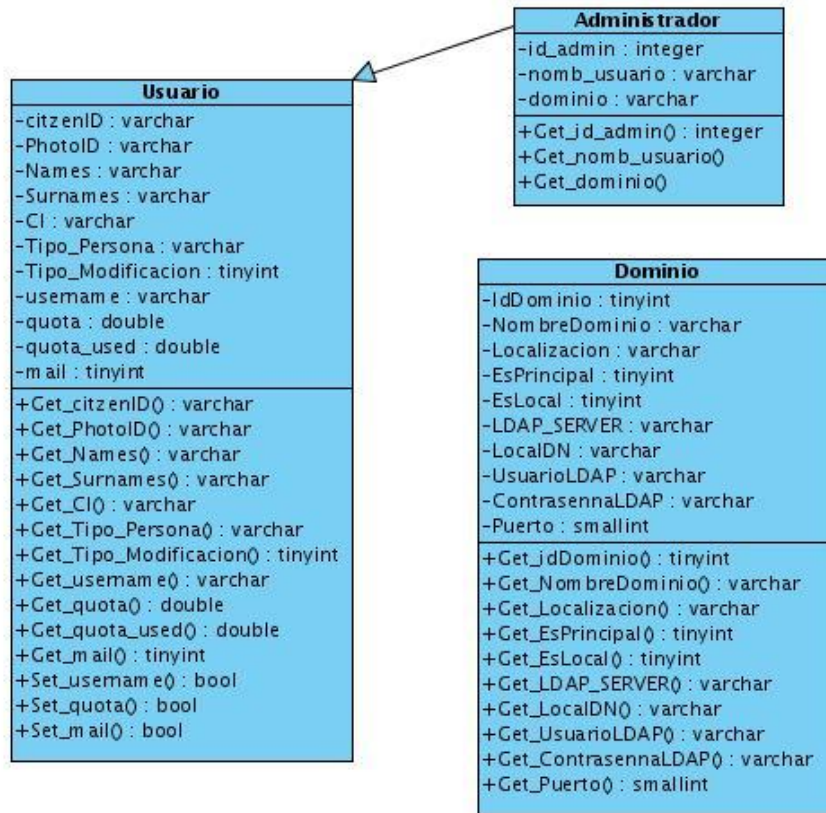
El *Diagrama de Colaboración* presenta una alternativa al diagrama de secuencia para modelar interacciones entre objetos en el sistema. Mientras que el diagrama de secuencia se centra en la secuencia cronológica del escenario que estamos modelando, el diagrama de colaboración se centra en estudiar todos los efectos de un objeto dado durante un escenario.

Los objetos se conectan por medio de enlaces, cada enlace representa una instancia de una asociación entre las clases implicadas. El enlace muestra los mensajes enviados entre los objetos, el tipo de mensaje y la visibilidad de un objeto con respecto a los otros. Dan una visión clara del flujo de control en el contexto en el que se desarrollan. Son útiles en la fase exploratoria para identificar objetos. La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto a los demás. La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces.

### 3.5 Diagrama de Clases Persistentes

No todas las clases identificadas en el dominio del análisis son persistentes, sino aquellas cuyos objetos deben ser almacenados en algún repositorio como una base de datos relacional. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo.

El diseñador tiene la responsabilidad de determinar cuáles son las clases que son persistentes, ya que cada una de ellas equivale a una tabla en la base de datos, y un registro o línea de esta tabla es un objeto persistente de esta clase.



**Figura 3.13:** Diagrama de Clases Persistentes

### 3.6 Conclusiones del Capítulo

En este capítulo se definieron y describieron las clases que contendrá el diseño de la aplicación web, necesarias para dar cumplimiento a los requerimientos del sistema, al igual que los diagramas necesarios para una mejor comprensión de la aplicación; siguiendo en todos los casos el patrón arquitectónico, Modelo-Vista-Controlador. En este sentido se modelaron los diagramas de clases (del análisis y del diseño) para cada uno de los casos de uso identificados, así como los diagramas de interacción correspondientes a los escenarios existentes más significativos, para conocer mejor los mensajes que intercambian los objetos al llevar a cabo las funcionalidades del sistema. Se definieron además las clases persistentes.

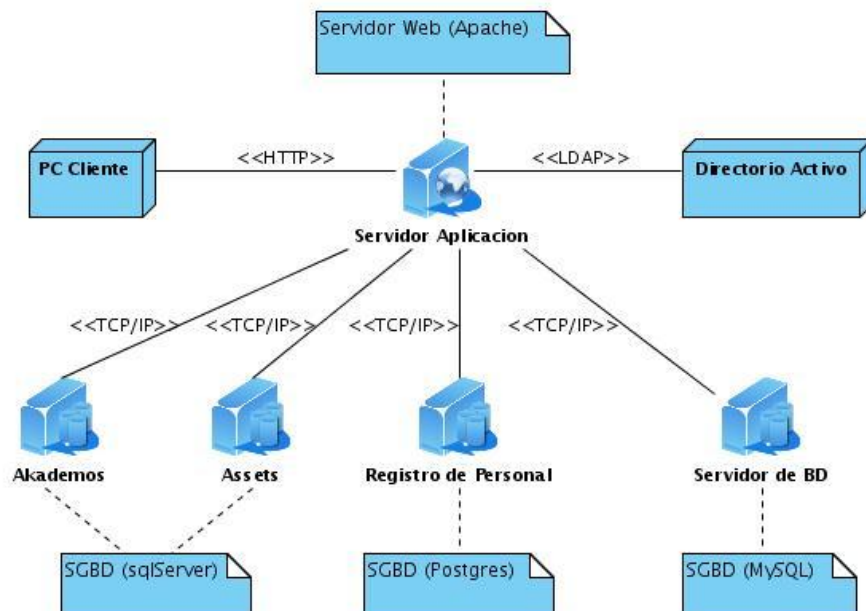
## CAPÍTULO IV: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### 4.1 Introducción

En este capítulo se llevan a cabo las descripciones de los artefactos que constituyen las clases de la aplicación y demás elementos necesarios para la implementación. Se definen además elementos significativos de la estructuración física de la aplicación, mediante los modelos de despliegue y de componentes.

### 4.2 Diagrama de Despliegue

Muestran nodos, conexiones, componentes y objetos. Los nodos representan objetos físicos con recursos computacionales como procesadores y periféricos; pueden mostrarse como una clase (ej. una familia de procesadores) o una instancia, por lo que su nombre sigue la misma sintaxis establecida para clases y objetos. Las conexiones son asociaciones de comunicación entre los nodos, y se etiquetan con un estereotipo que identifica el protocolo de comunicación o la red utilizada. Los componentes son archivos de código ejecutable, que residen y se ejecutan dentro de un nodo; se pueden representar relaciones de dependencia entre los componentes que, de manera similar a las dependencias entre paquetes, corresponden al uso de servicios.



**Figura 4.1:** Diagrama de Despliegue

A continuación se especifican detalles de cada uno de los nodos del modelo:

PC Cliente: se refiere a las estaciones de trabajo que utilizará el usuario para acceder a la aplicación Web de Gestión de Datos Telemáticos, así como a los sistemas externos que accedan al sistema para consumir los servicios brindados.

Servidor Aplicación: se refiere al Servidor Web que publica tanto la aplicación Web de Gestión de Datos Telemáticos como la Fachada de Servicios Telemáticos, que posibilitan que los diferentes usuarios que accedan a ella dispongan de sus respectivas funcionalidades.

Servidor de BD: se refiere al servidor que radica en el nodo central de la UCI donde van a estar centralizados los datos telemáticos.

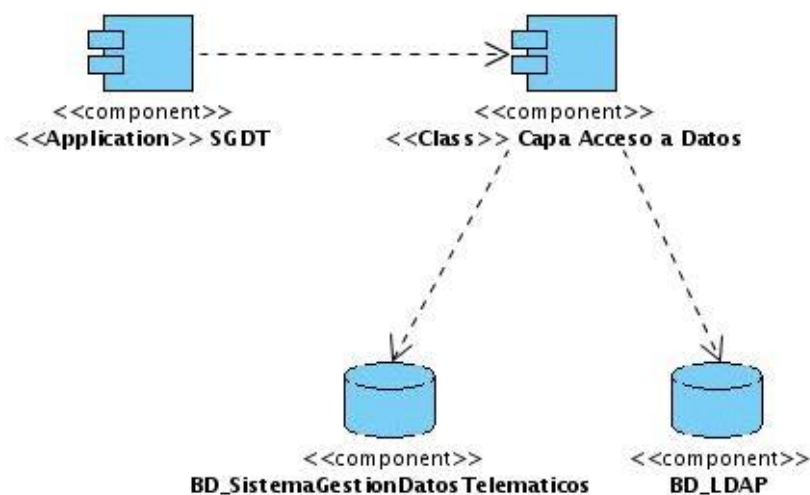
Directorio Activo: Directorio Activo de la UCI.

### 4.3 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización, incluyendo código fuente, binario y ejecutable.

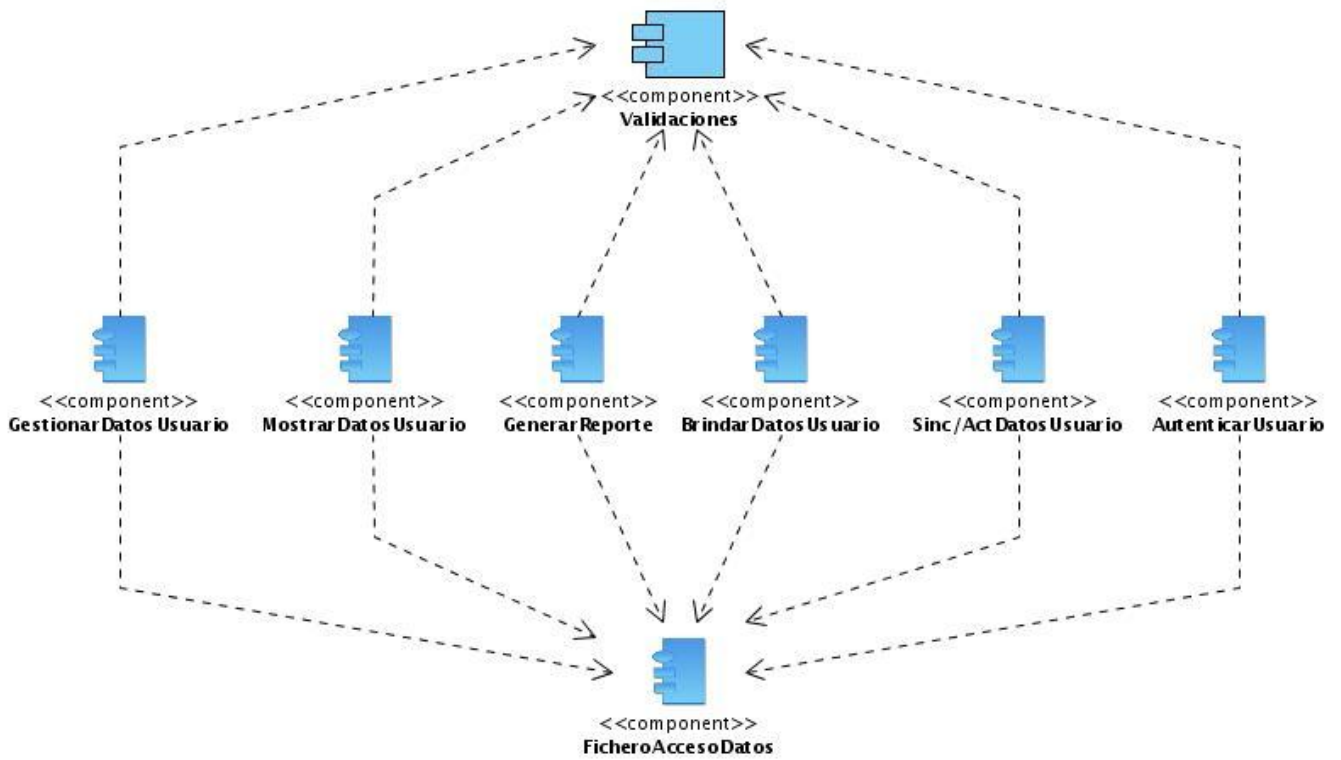
Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc.

#### 4.3.1 Diagrama de Componentes: Base de Datos



**Figura 4.2:** Diagrama de Componentes: Base de Datos

4.3.2 Diagrama de Componentes: Código Fuente



*Figura 4.3: Diagrama de Componentes: Código Fuente*

4.3.3 Diagrama de Componentes: Componentes Web o Código Ejecutable

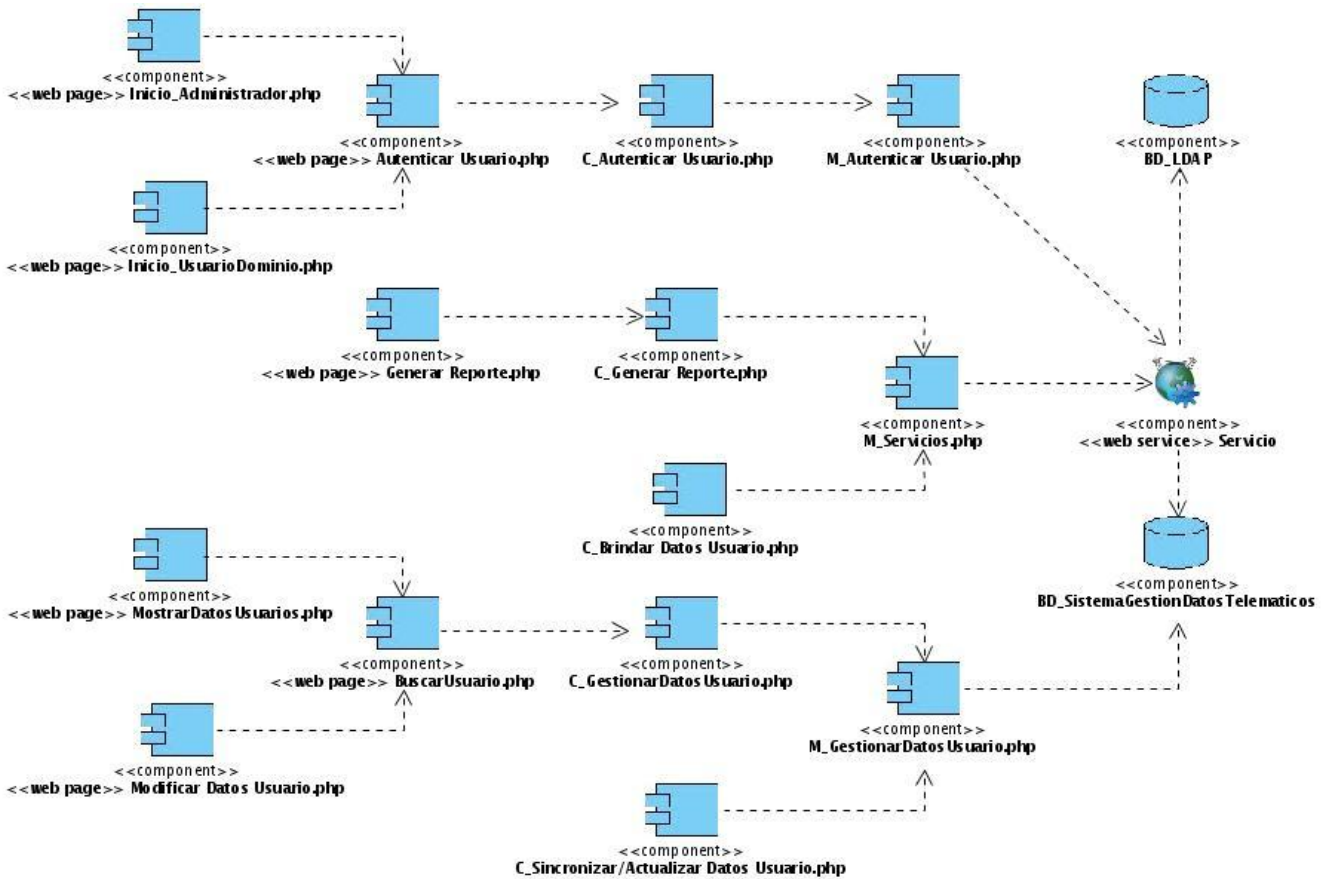


Figura 4.4: Diagrama de Componentes: Componentes Web o Código Ejecutable

4.4 Conclusiones del Capítulo

En este capítulo, con la realización de los diagramas de despliegue y componentes, quedó conformado el modelo de implementación de la aplicación, reforzados con la representación de los diagramas de componente de base de datos, código fuente y código ejecutable.

## CAPÍTULO V: ESTUDIO DE LA FACTIBILIDAD

### 5.1 Introducción

Antes de elaborar un proyecto, es de suma importancia conocer si es conveniente seguir o no con su desarrollo. El estudio de la factibilidad nos ayuda en gran medida a realizar un análisis profundo; comparando costos, esfuerzos y beneficios.

En este capítulo se muestra el estudio realizado para determinar la factibilidad del software a desarrollar, utilizando el método “Análisis de Puntos de Casos de Uso” para llevar a cabo las estimaciones del esfuerzo y la complejidad de las actividades identificadas en el proyecto. Como resultado de este análisis se obtienen: el tiempo de desarrollo en meses, el costo y la cantidad de personas necesarias para desarrollar el proyecto.

### 5.2 Planificación basada en casos de uso

La estimación por Puntos de Casos de Uso resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de los primeros Casos de Uso de un sistema, si se sigue una aproximación iterativa como el Proceso Unificado de Rational. En este tipo de aproximación, los primeros Casos de Uso a desarrollar son los que ejercitan la mayor parte de la arquitectura del software y los que a su vez ayudan a mitigar los riesgos más significativos (iteraciones de Elaboración en el Proceso Unificado).<sup>42</sup>

Definición de los pasos para realizar este proceso:

- 1- Cálculo de Puntos de Casos de Uso sin ajustar.

$$\mathbf{UUCP = UAW + UUCW}$$

Donde:

**UUCP:** Puntos de Casos de Uso sin ajustar.

**UAW:** Factor de Peso de los Actores sin ajustar.

**UUCW:** Factor de Peso de los Casos de Uso sin ajustar.

- 1.1 Factor de Peso de los Actores sin ajustar (**UAW**)

---

<sup>42</sup> PERALTA, Mario. “Estimación del esfuerzo basada en casos de uso” [pdf]. Disponible en: <http://www.itba.edu.ar/capis/rtis/rtis-6-1/estimacion-del-esfuerzo-basada-en-casos-de-usos.pdf>



Tipo de Actor	Descripción	Factor de peso	Actores	Total = (cant actores * peso)
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1	1	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	1	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3	3	9
<b>Total <math>\Sigma</math></b>				<b>12</b>

1.2 Factor de Peso de los Casos de Uso sin ajustar (**UUCW**).

Tipo de CU	Descripción	Peso	Cantidad de CU	Total = cant CU * peso
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	10	50
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	1	15
<b>Total <math>\Sigma</math></b>				<b>65</b>

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

$$\text{UUCP} = 12 + 65$$

$$\text{UUCP} = 77$$

2- Cálculo de Puntos de Casos de Uso ajustados.

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF}$$

Donde:

**UCP:** Puntos de Casos de Uso ajustados.

**UUCP:** Puntos de Casos de Uso sin ajustar.

**TCF:** Factor de complejidad técnica.

**EF:** Factor de ambiente.

2.1 Factor de complejidad técnica (**TCF**).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante, esto está reflejado en la siguiente tabla:

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>	<b>Valor asignado</b>	<b>Total = peso * valor asignado</b>
T1	Sistema distribuido	2	0	<b>0</b>
T2	Tiempo de respuesta	1	4	<b>4</b>
T3	Eficiencia del usuario final	1	1	<b>1</b>
T4	Procesamiento interno complejo	1	3	<b>3</b>
T5	El código debe ser reutilizable	1	5	<b>5</b>
T6	Facilidad de instalación	0.5	5	<b>2.5</b>
T7	Facilidad de uso	0.5	5	<b>2.5</b>
T8	Portabilidad	2	5	<b>10</b>
T9	Facilidad de cambio	1	3	<b>3</b>
T10	Concurrencia	1	4	<b>4</b>
T11	Incluye objetivos especiales de seguridad	1	3	<b>3</b>
T12	Provee acceso directo a terceras partes	1	5	<b>5</b>
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	<b>1</b>
<b>Total = 0.6 + 0.01 * <math>\Sigma</math></b>				<b>1.04</b>

2.2 Factor de ambiente (**EF**).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente.

Factor	Descripción	Peso	Valor asignado	Total = peso * valor asignado
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	<b>4.5</b>
E2	Experiencia en la aplicación	0.5	3	<b>1.5</b>
E3	Experiencia en la orientación a objetivos	1	5	<b>5</b>
E4	Capacidad del analista líder	0.5	5	<b>2.5</b>
E5	Motivación	1	5	<b>5</b>
E6	Estabilidad de requerimientos	2	3	<b>6</b>
E7	Personal Part-Time.	-1	3	<b>-3</b>
E8	Dificultad del lenguaje de programación	-1	0	<b>0</b>
<b>Total = 1.4 - 0.03 * <math>\Sigma</math></b>				<b>0.755</b>

- ✓ Para los factores del E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).
- ✓ Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- ✓ Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.
- ✓ Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part-time (nadie es full-time).
- ✓ Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF}$$

$$\text{UCP} = 77 \times 1.04 \times 0.755$$

$$\text{UCP} = 60.4604$$

2.3 De los puntos de casos de uso a la estimación.

- ✓ Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores del E1 a E6.
- ✓ Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

- ✓ Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- ✓ Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- ✓ Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Factor de Conversión (CF): 20 horas/hombres

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde:

**E:** esfuerzo estimado en horas-hombre

**UCP:** Puntos de Casos de Uso ajustados

**CF:** factor de conversión

$$E = UCP \times CF$$

$$E = 60.4604 \times 20$$

$$E = 1209.208$$

Para una estimación más completa de la duración total del proyecto, hay que agregar a la obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Actividad	Porcentaje %	Horas-Hombre
Análisis	10	302.302
Diseño	20	604.604
Implementación	40	1209.208
Pruebas	15	453.453
Sobrecarga (otras actividades)	15	453.453
<b>Total</b>	<b>100</b>	<b>3023.02</b>

Esfuerzo Total (Horas--Hombre)	ET1	3023.02
--------------------------------	-----	---------

Esfuerzo Total (Mes--Hombre)	ET2	12.5959167
Salario Promedio	SM	100
Cantidad de Hombres	CH	2
Costo Hombre--Mes	CHM	200
Costo Total	Costo	2519.18333

En resumen:

$$\text{Tiempo} = \text{ET} / \text{CH}$$

$$\text{Tiempo} = 12.5959167/2$$

$$\text{Tiempo} = \mathbf{6.29795835 \sim 6.30 \text{ meses}}$$

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto, el mismo se desarrolla en 6.30 meses y su costo total se estima sea de \$2519.04.

### 5.3 Beneficios Tangibles e Intangibles

La Fachada de Servicios Telemáticos no es un producto con fines comerciales, el principal objetivo de esta es sincronizar en una vista única de información los datos de las personas necesarios para brindar dichos servicios.

Entre los beneficios se contará con una aplicación Web dinámica, flexible y sencilla para actualizar y gestionar los datos de los usuarios necesarios para los servicios telemáticos.

Los beneficios inmediatos son por lo general intangibles, entre ellos se encuentran:

- Se contará con un proceso de actualización automática de la información en la Base de Datos del Nodo.
- Brinda un punto de comunicación entre los sistemas primarios de información de las personas y los servicios telemáticos.
- Enriquece los servicios del portal de la Intranet de la UCI.
- Permite que el Directorio Activo repose de las responsabilidades de consulta de estos datos, brindando una vista única de información de datos telemáticos.
- Aporta facilidad y rapidez a la hora de obtener reportes de la información de los usuarios.

- Establece una sincronización entre los datos primarios de los usuarios de la UCI, registrados en los sistemas de almacenamiento de información de usuarios, con los datos correspondientes a estos mismos usuarios en el Directorio Activo del centro.

### **5.4 Análisis de Costos y Beneficios**

Para llevar a cabo el desarrollo de un sistema informático es necesario analizar los costos que incurrirán en su proceso. Analizando el costo del proyecto y los numerosos beneficios que reporta, detallados con anterioridad, se puede dar por sentado que el desarrollo de este producto es realmente factible.

### **5.5 Conclusiones del Capítulo**

En este capítulo se efectuó y describió el estudio de factibilidad correspondiente al desarrollo del sistema propuesto, teniendo en cuenta los aspectos que de una forma u otra influyen en este proceso. El tiempo y costo estimados para su realización, así como los beneficios que reportará la terminación del producto en cuestión indican que es factible la implementación y utilización del mismo.

## CONCLUSIONES

Tras haber estudiado con profundidad la problemática existente en la Universidad de las Ciencias Informáticas en cuanto a la gestión de datos telemáticos, se determinó llevar a cabo la automatización de los procesos de sincronización y actualización de los datos referentes a los servicios telemáticos y la información primaria de cada usuario.

En este informe se recoge de forma clara y detallada cómo darle solución a la misma; describiendo las herramientas necesarias para llevar a cabo este proceso, además de presentar un estudio de las tendencias actuales para realizar tareas similares.

Se cumplió satisfactoriamente el objetivo general propuesto, contando con la creación de la Fachada Telemática. La misma implementa servicios web que permiten exponer, hacia otros sistemas de la universidad, información de sus usuarios relevante para los servicios telemáticos.

El sistema desarrollado es de vital importancia para el resto de los sistemas de la UCI, ya que es el centro de la autenticación para sistemas como Intranet, Akademos, Assets y demás. Esto es aún más complicado al sumarse las facultades regionales al asunto, puesto que hay servicios que necesitan comunicación con los servicios telemáticos de la sede central.

Se obtuvieron además los siguientes resultados:

- 1- Documentación detallada del análisis, diseño e implementación de la aplicación.
- 2- Quedaron demostradas la validez y eficacia de las herramientas seleccionadas, así como las tendencias actuales estudiadas.
- 3- Se creó una base de datos con la información primaria y los respectivos datos telemáticos de cada usuario de la universidad.
- 4- La solución propuesta es probada y aceptada en la universidad y en las facultades regionales de Ciego de Ávila y Manzanillo.

Por lo anteriormente descrito queda reflejado el cumplimiento de los objetivos específicos.

## **RECOMENDACIONES**

Buscando posibles mejoras al sistema, se recomienda:

- Agregar al Sistema de Gestión de Datos Telemáticos un módulo de gestión de administradores.
- Agregar al Sistema de Gestión de Datos Telemáticos un módulo de gestión de dominios.



## BIBLIOGRAFÍA

"Ciencia y Tecnología". Disponible en: [http://cybermicrologic.com/cml\\_web/esp/html/3\\_tecnologia.html](http://cybermicrologic.com/cml_web/esp/html/3_tecnologia.html) (22/05/2008).

Iztiair G. Alonso González, Gustavo Rodríguez Rodríguez, Domingo Marrero Marrero. "Servicios Telemáticos". Disponible en: <http://absys01.ulpgc.es/cgi-bin/abnetopac/O7081/IDed04735a/NT15> (15/11/2007).

RUBIO, Manuel. "Vista Única: la base para una nueva generación de servicios orientados al ciudadano". Disponible en: [http://blogs.sun.com/manuelrubio/entry/vista\\_%C3%BAnica\\_la\\_base\\_para](http://blogs.sun.com/manuelrubio/entry/vista_%C3%BAnica_la_base_para) (06/06/2008).

Booch, G., Rumbaugh, J., Jacobson, I. "El Proceso Unificado de Desarrollo de Software". Addison-Wesley Longman, 2000.

Dpto. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. "Introducción a RUP", 2006. Disponible en <https://pid.dsic.upv.es/C1/Material/default.aspx> ( 23/11/2007).

Mendoza Sánchez, María A. "Metodologías de desarrollo de software". Disponible en: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html) (23/11/2007).

Booch, G., Rumbaugh, J., Jacobson, I. "El Lenguaje Unificado de Modelado", Addison-Wesley, 2000.

"PHP". Disponible en: [http://www.ideocentro.com/recursos.php?id=14&ideocentro\\_diseno\\_web\\_interactive=PHP--PHP-Hypertext-Pre-processor-](http://www.ideocentro.com/recursos.php?id=14&ideocentro_diseno_web_interactive=PHP--PHP-Hypertext-Pre-processor-) (22/05/2008).

Instituto Nacional de Estadística e Informática, Dirección Técnica de Desarrollo Informático. *Herramientas CASE*. Disponible en: <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>. (22/01/2008).

Sitio oficial del Visual Paradigm. Disponible en: <http://www.visual-paradigm.com/product/vpuml>. (21/01/2008).

“Probablemente el mejor editor de páginas web para diseñadores que busquen resultados profesionales”. Disponible en: <http://www.desarrolloweb.com/articulos/332.php> (13/03/2008).

PÉREZ VALDÉS, Damián. *“Editores Web que facilitan tu trabajo”*. Disponible en: <http://www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo> (06/02/2008).

Sitio: The Apache Software Foundation. Disponible en: <http://www.apache.org/> (06/02/2008).

CELIS Ismael. *“El ataque de los Frameworks”*. Disponible en: <http://www.estadobeta.com/2005/11/20/el-ataque-de-los-frameworks/> (15/03/2008).

Web oficial de Code Igniter. Disponible en: <http://codeigniter.com> (30/04/2008).

*“Sistemas Gestores de Base de Datos”*. Disponible en: [http://www.error500.net/garbagecollector/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos.html](http://www.error500.net/garbagecollector/bases_de_datos/sistema_gestor_de_base_de_datos.html) (24/11/2007).

*“Plan de Cases: Curso de Instalación y configuración de CMS”*. Disponible en: <http://dspace.idict.cu/bitstream/123456789/134/1/Plan+de+Cases.doc> (23/05/2008).

Oficina española del w3c (World Wide Web). Guía Breve de Servicios Web. Disponible en: <http://www.w3c.es/Divulgacion/Guiasbreves/ServiciosWeb#uso#uso> (21/01/2008).

BANKHACKER.COM. *Taller: Computación Distribuida: Servicios Web de Traducción*. Disponible en: <http://web-services.bankhacker.com> (21/01/2008).

LARMAN, Craig. *“UML y Patrones. Introducción al análisis y diseño orientado a objetos”*. Primera Edición por Prentice Hall, Hispanoamericana S.A. 1999. Tomo I, Capítulo 18.

The Business Rules Group. *“Manifiesto de Reglas de Negocio. Los Principios de la Independencia de las Reglas”*. Disponible en: <http://www.businessrulesgroup.org/brmanifiesto/BRManifiesto%5Bv1%5B1%5D.0%5D.pdf> .

PERALTA, Mario. *“Estimación del esfuerzo basada en casos de uso”* [pdf]. Disponible en: <http://www.itba.edu.ar/capis/rtis/rtis-6-1/estimacion-del-esfuerzo-basada-en-casos-de-usos.pdf> .

Nora, S. y Minc, A. "La informatización de la sociedad". Traducción de Paloma García de Pruneda, F.C.E. de España, Madrid, 1983.

Consultoría Biomundi / IDICT, Lourdes Vilaragut Llanes. "Estado Actual y Tendencias de la Informatización de los servicios. Comportamiento en Cuba". (Noviembre/2007).

Biblioteca Nacional José Martí, de Cuba. Disponible en: <http://www.bnjm.cu/librinsula> (15/11/2007).

"Políticas de los Servicios Telemáticos en la Universidad", 2004 (vigente). Disponible en: <http://telematicos.uci.cu/documents/ts-policy.doc> (24/11/2007).

Dirección de Información Universidad de las Ciencias Informáticas. Norma ISO 690 (1 y 2).ppt (24/11/2007).

Kruchten, P. "The Rational Unified Process: An Introduction", Addison Wesley, 2000.

Booch, G., Rumbaugh, J., Jacobson, I. "The Unified Software Development Process". Addison-Wesley Longman, Inc. 1999.

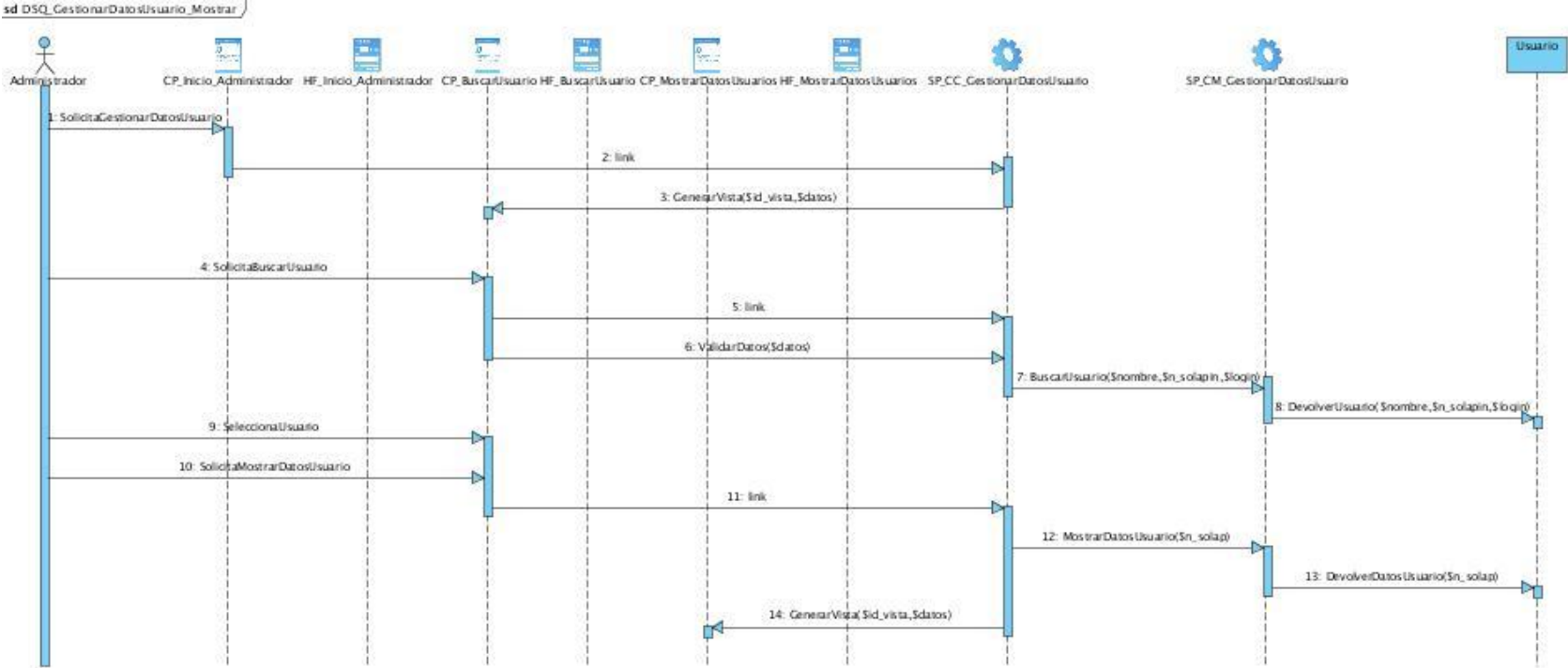
LLANO, Oscar. "Desarrollo Web Hosting Software". Disponible en: <http://marquetingnet.com/net/disenio-web/disenio-dinamico.html> (22/05/2008).

Arquitectura del Servidor Apache. Disponible en: <http://www.desarrolloweb.com/articulos/1112.php> (06/02/2008).

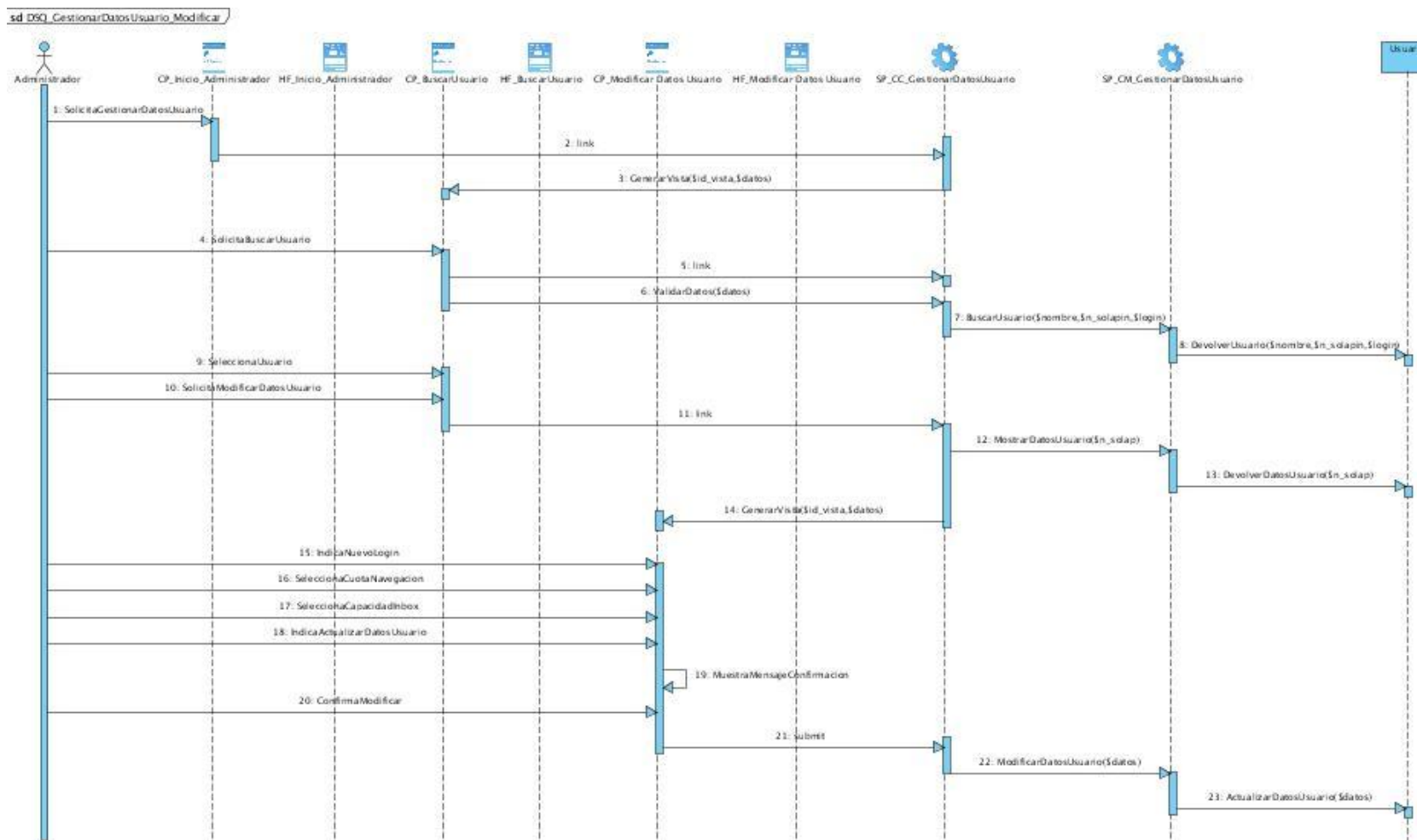
DUEÑAS, José. "Code Igniter, un gran framework para PHP". Disponible en: <http://www.joseduenas.com/archivos/codeigniter-un-gran-framework-para-php> (30/04/2008).

# ANEXOS

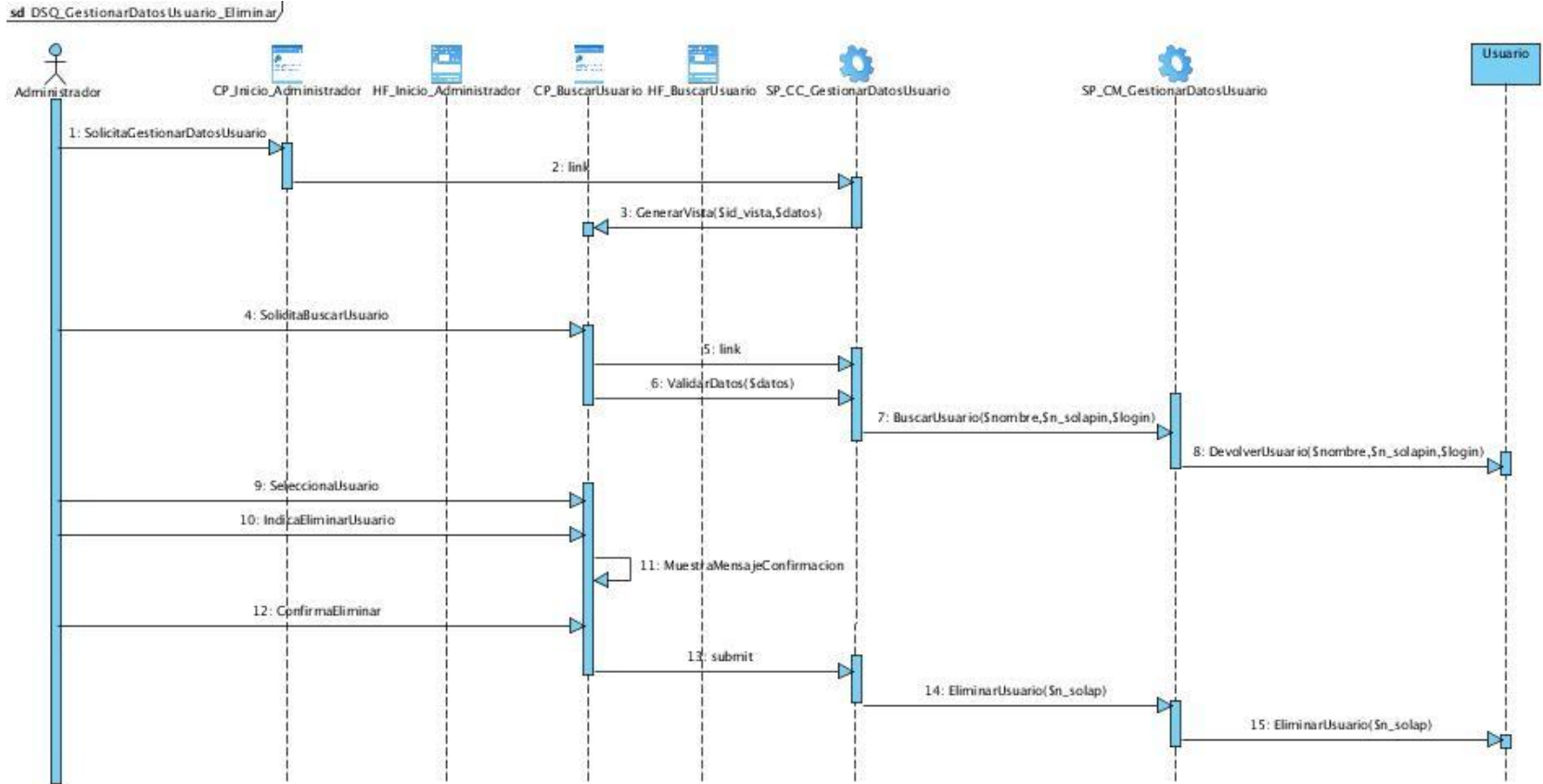
Anexo 1. Diagrama de secuencia: CU\_GestionarDatosUsuario\_Mostrar



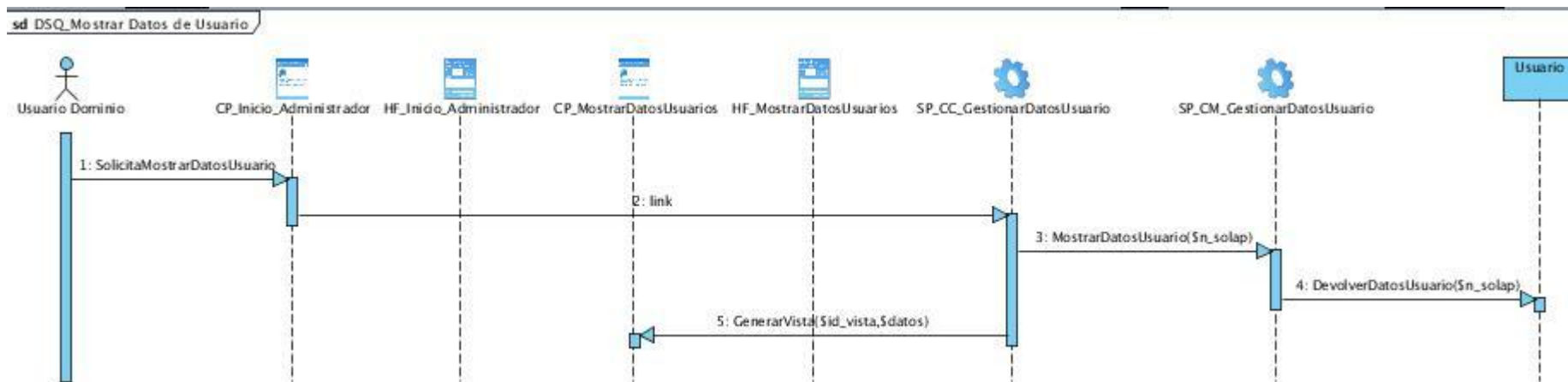
## Anexo 2. Diagrama de secuencia: CU\_GestionarDatosUsuario\_Modificar



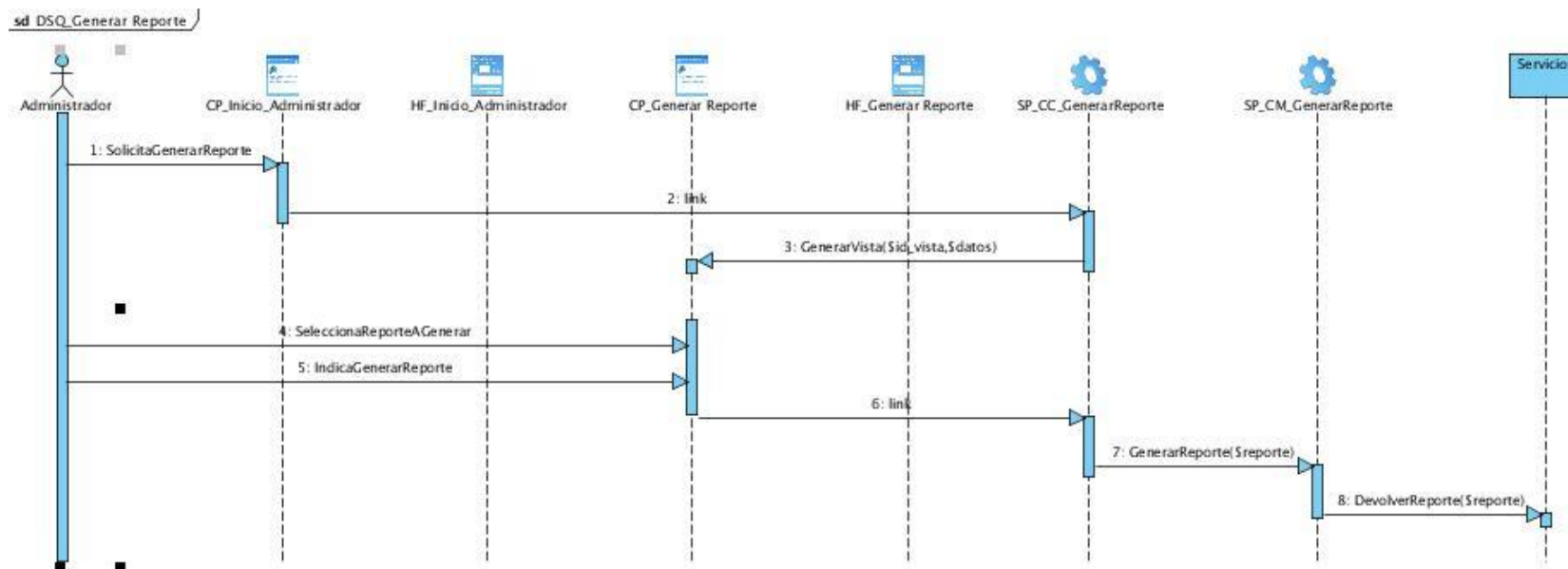
**Anexo 3. Diagrama de secuencia: CU\_GestionarDatosUsuario\_Eliminar**



Anexo 4. Diagrama de secuencia: CU\_Mostrar Datos de Usuario

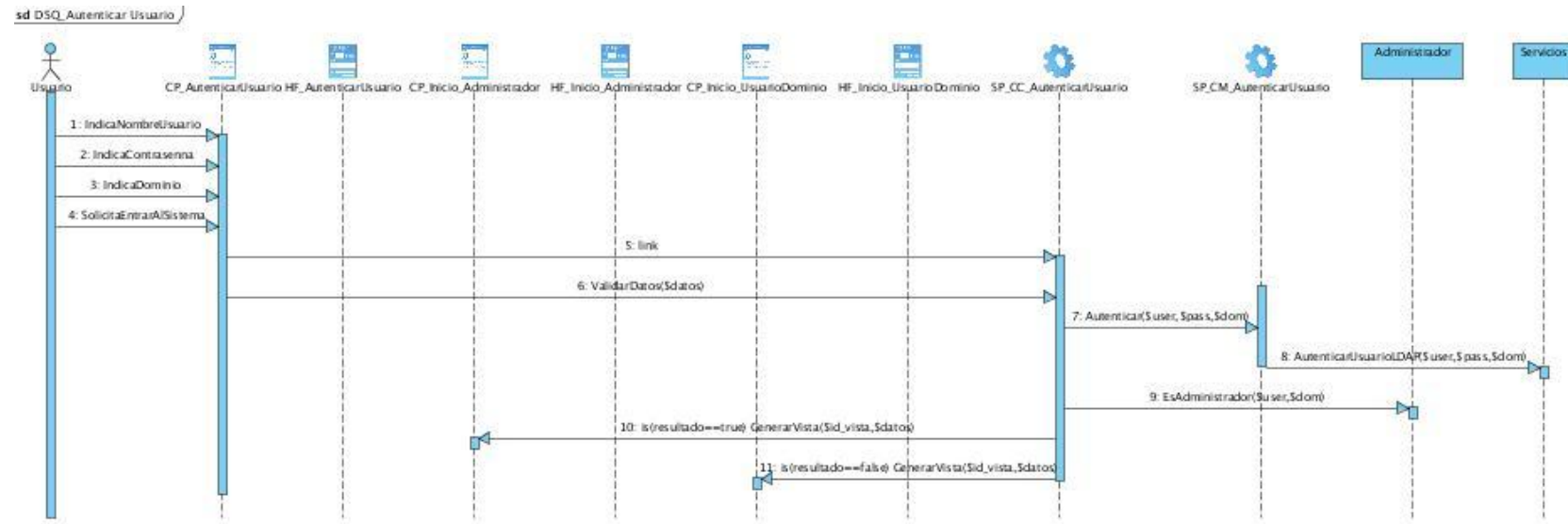


### Anexo 5. Diagrama de secuencia: CU\_Generar Reporte





## Anexo 6. Diagrama de secuencia: CU\_AutenticarUsuario



## **GLOSARIO**

**API** (Application Programming Interface): Es un conjunto de funciones que facilitan el intercambio de mensajes o datos entre dos aplicaciones.

**CASE** (Computer Aided Software Engineering/ Ingeniería de Software Asistida por Ordenado): Son aplicaciones informáticas para aumentar la productividad en el desarrollo de software reduciendo costos de tiempo y de dinero.

**FTP** (Protocolo de transferencia de archivos): Ofrecido por la capa de Aplicación del modelo de capas de red TCP/IP.

**HTML** (Lenguaje de Marcas de Hipertexto): Es usado para describir la estructura y contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**HTTP** (HyperText Transfer Protocol): Protocolo usado en cada transacción de la Web.

**HTTPS** (HyperText Transfer Protocol Secure): Es la versión segura de HTTP.

**IDE** (Integrated Development Environment): Es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

**LDAP** (Lightweight Directory Access Protocol): Permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

**Login**: Nombre de usuario para ser identificado en la red, es de condición única.

**OO** (Orientado a Objeto): Se refiere al paradigma orientado a objetos.

**SQL** (Structured Query Language): Lenguaje declarativo de acceso a bases de datos relacionales.

**SW**: Software

**USB** (Universal Serial Bus): Puerto que sirve para conectar periféricos a una computadora.