

**Universidad de las Ciencias Informáticas**  
**Facultad 10**



**Título: Análisis y diseño de una propuesta de Plataforma para la  
Gestión de la Información de Usuarios**

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

**Autor:** Jorge Luis del Valle Barthelemy.

**Tutor:** Ing. Alexander Hernández Chapman.

Ciudad de La Habana, junio de 2008.

## DECLARACIÓN DE AUTORÍA.

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

---

Nombre del Autor

Nombre del Tutor

---

Firma del Autor

---

Firma del Tutor

## Dedicatoria

*El presente trabajo está dedicado a toda la gente linda que ha tenido que ver con los resultados alcanzados con la realización del mismo.*

*A mis Padres que sin su apoyo durante todos estos años no hubiera sido posible que llegara a formarme como profesional. A mi Tío, a mis Abuelos (los cuatro), a mi Novia que me dio todo el apoyo necesario en los momentos que me hizo falta. A mis amigos de la escuela, a los de la zona, para Alfred, Magdiel, Adrian, Yoan, Indy y todos los demás que sin ellos esto no hubiera sido posible. A los profes que me ayudaron en todo, a todos mis compañeros de aula y de apartamento, a los que quedaron en el camino por diferentes causas, a Aramis, al Charlys, y otros, que sé que conseguirán su sueño donde quiera que se encuentren más pronto que tarde...*

## **RESUMEN**

El objetivo de este trabajo es presentar una Propuesta de Plataforma para la Gestión de la Información de Usuarios de cualquier institución que requiera este tipo de aplicación para el control y administración de la información referente a sus usuarios. Para lograr este objetivo se parte de la definición de Plataforma para la Gestión de la Información de Usuarios, su importancia, actualidad y necesidad. Se realizará un estudio de sistemas similares, así como de las tecnologías que más se usan para llevar a cabo este tipo de aplicaciones.

También se llevará a cabo un amplio estudio sobre el desarrollo y evolución de este tipo de Plataformas en la actualidad, para de esta manera poder realizar una propuesta que sea fiable y cumpla con los objetivos de este trabajo para su posterior implementación.

Palabras clave: servicio web, plataforma, gestión de usuarios.

## ÍNDICE GENERAL.

<b>INTRODUCCIÓN.</b> .....	1
<b>Estructuración del contenido.</b> .....	3
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.</b> .....	5
<b>1.1 Plataforma para la Gestión de la Información de Usuarios. Definiciones.</b> .....	5
<b>1.2 Desarrollo y evolución de los Sistemas de Gestión de Usuarios.</b> .....	6
<b>1.3 Otros sistemas relacionados: SSO.</b> .....	12
<b>1.4 Necesidad de una Plataforma para la Gestión de la Información de Usuarios.</b> .....	15
<b>1.5 Ventajas de contar con una Plataforma para la Gestión de la Información de Usuarios.</b> .....	16
<b>1.6 Metodologías, Tecnologías y Software utilizados.</b> .....	17
1.6.1 RUP.....	17
1.6.2 UML.....	17
1.6.3 Visual Paradigm.....	18
<b>1.7 Tecnologías que se proponen para el futuro desarrollo e implementación de la Plataforma para la Gestión de la Información de Usuarios.</b> .....	18
1.7.1 Servicio Web. ....	18
1.7.2 XML.....	19
1.7.3 SOAP.....	19
1.7.4 WSDL.....	20
1.7.5 Servidor Web Apache.....	21
1.7.6 MySQL.....	21
1.7.7 Lenguaje de programación PHP.....	21
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	23

<b>2.1</b>	<b>Problema y situación problemática.....</b>	<b>23</b>
2.1.1	Objetivos estratégicos de la organización y proceso de negocio que los soportan.....	23
2.1.2	Flujo actual de los procesos involucrados en el campo de acción.....	23
2.1.3	Análisis crítico de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problemática y las consecuencias. ....	23
<b>2.2</b>	<b>Objeto de Automatización. ....</b>	<b>24</b>
2.2.1	Descripción de los procesos que serán objeto de automatización.....	24
2.2.2	Descripción de los sistemas automatizados que están vinculados con el campo de acción..	24
<b>2.3</b>	<b>Información que se maneja. ....</b>	<b>24</b>
2.3.1	Documentos específicos que se procesen, detalles de la información que se manipule.....	24
<b>2.4</b>	<b>Propuesta de Sistema.....</b>	<b>25</b>
2.4.1	Descripción general de la propuesta de sistema, cómo debe funcionar. ....	25
2.4.2	Análisis comparativo de otras soluciones existentes con la propuesta, llegando a conclusiones sobre los aspectos en los que esta propuesta se diferencia de las otras y las supera. ...	26
<b>2.5</b>	<b>Modelo de dominio.....</b>	<b>26</b>
<b>2.6</b>	<b>Especificación de los requisitos de software. ....</b>	<b>27</b>
2.6.1	Requerimientos Funcionales. ....	27
2.6.2	Requerimientos no Funcionales. ....	28
<b>2.7</b>	<b>Definición de los actores y casos de uso del sistema. ....</b>	<b>29</b>
2.7.1	Actores del sistema.....	29
2.7.2	Casos de uso del sistema. ....	29
2.7.3	Breve descripción de los Casos de Uso del Sistema. ....	30
2.7.4	Descripción expandida de los Casos de Uso del Sistema.....	32

<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.</b>	41
<b>3.1 Análisis.</b>	41
3.1.1 Definición del modelo de análisis. Modelo de clases de análisis.	41
3.1.2 Diagramas de clases de análisis.	41
3.1.3 Diagramas de interacción.	43
<b>3.2 Diseño.</b>	45
3.2.1 Diagramas de clases de diseño.	46
3.2.2 Descripción de las clases.	49
3.2.3 Diagramas de interacción.	53
3.2.4 Diseño de la Base de Datos.	55
3.2.5 Descripción de las Tablas de la Base de Datos.	56
<b>CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD.</b>	58
<b>4.1 Características del proyecto.</b>	58
4.1.1 Entradas externas.	58
4.1.2 Consultas externas.	58
4.1.3 Archivos lógicos internos.	59
4.1.4 Archivos de interfaz externos.	59
4.1.5 Puntos de función sin ajustar.	59
<b>4.2 Cálculo de esfuerzo e instrucciones fuentes.</b>	59
<b>CONCLUSIONES.</b>	64
<b>RECOMENDACIONES.</b>	65
<b>REFERENCIAS BIBLIOGRÁFICAS.</b>	66
<b>GLOSARIO DE TÉRMINOS.</b>	68

## ÍNDICE DE FIGURAS.

Fig. 1 PayRoll RH .....	6
Fig. 2 PayRoll RH .....	8
Fig. 3 Simple HCM.....	9
Fig. 4 GREHU.....	11
Fig. 5 GREHU.....	11
Fig. 6 GREHU.....	12
Fig. 7 GREHU.....	12
Fig. 8 SAML.....	14
Fig. 9 Cams .....	15
Fig. 10 Modelo de Dominio .....	27
Fig. 11 Diagrama de Casos de Uso del Sistema.....	30
Fig. 12 Diagrama de clases de análisis del Caso de Uso “Autenticar Usuario” .....	41
Fig. 13 Diagrama de clases de análisis del Caso de Uso “Gestionar Usuario” .....	42
Fig. 14 Diagrama de clases de análisis del Caso de Uso “Obtener Ficha de Usuario” .....	42
Fig. 15 Diagrama de clases de análisis del Caso de Uso “Buscar Información de Usuario” .....	42
Fig. 16 Diagrama de colaboración del Caso de Uso “Autenticar Usuario” .....	43
Fig. 17 Diagrama de colaboración del Caso de Uso “Gestionar Usuario”.....	44
Fig. 18 Diagrama de colaboración del Caso de Uso “Obtener Ficha de Usuario” .....	45
Fig. 19 Diagrama de colaboración del Caso de Uso “Buscar Información de Usuario” .....	45
Fig. 20 Diagrama de clases de diseño del Caso de Uso “Autenticar Usuario” .....	46
Fig. 21 Diagrama de clases de diseño del Caso de Uso “Gestionar Usuario” .....	47
Fig. 22 Diagrama de clases de diseño del Caso de Uso “Obtener Ficha de Usuario” .....	48



Fig. 23 Diagrama de clases de diseño del Caso de Uso “Buscar información de Usuario” .....	49
Fig. 24 Diagrama de secuencia del caso de uso “Autenticar Usuario” .....	53
Fig. 25 Diagrama de secuencia del caso de uso “Gestionar Usuario”. Escenario “Adicionar” .....	54
Fig. 26 Diagrama de secuencia del caso de uso “Gestionar Usuario”. Escenario “Modificar” .....	54
Fig. 27 Diagrama de secuencia del caso de uso “Gestionar Usuario”. Escenario “Eliminar” .....	54
Fig. 28 Diagrama de secuencia del caso de uso “Obtener Ficha de Usuario” .....	55
Fig. 29 Diagrama de secuencia del caso de uso “Buscar Información de Usuario” .....	55
Fig. 30 Diagrama de Clases Persistentes .....	56
Fig. 31 Modelo de Datos .....	56

## **INTRODUCCIÓN.**

El incremento de los volúmenes de información, así como el desarrollo de las empresas son la causa de que en la actualidad sea necesario guardar toda esa información de manera persistente y por tiempo indefinido, a la vez que también reviste una gran importancia el poder acceder a ella de manera fácil y segura pues no sabemos en qué momento será necesario el uso de la misma. En este trabajo nos enfocaremos directamente en la información referida a los usuarios de dichas empresas o instituciones, ya que la misma se hace necesaria a la hora de tomar cualquier tipo de decisión, sea esta de tipo administrativo o no.

Por las razones anteriormente expuestas, contar con toda esa información en el momento que se desee es de vital importancia, y a su vez, esta importancia recae en la necesidad de contar con un sistema que sea capaz de gestionar toda esa información y brindarla en el momento preciso, ya que todos los datos referentes a los usuarios de una empresa o institución son vitales a la hora de asignar alguna tarea o responsabilidad a uno de ellos.

En la actualidad existen diferentes sistemas de gestión de usuarios, pero la mayoría de ellos sólo son aplicables a determinadas instituciones, por lo que se hace necesaria la creación de una Plataforma para la Gestión de la Información de Usuarios, es decir, un sistema que sea adaptable a cualquier tipo de institución o empresa y que cumpla a cabalidad con todas las funcionalidades necesarias y requeridas por la institución para el manejo de toda la información referente a sus usuarios, lo cual constituye la **Situación Problemática** de este trabajo, y que a su vez origina el siguiente **Problema Científico**:

¿Cómo diseñar una Plataforma para la Gestión de la Información de Usuarios?

El **Objeto de la Investigación** son las Plataformas para la Gestión de la Información de Usuarios y el **Campo de Acción** es la gestión de la información de usuarios.

Para esto partiremos de la siguiente **Hipótesis**:

“Con la implementación de una Plataforma para la Gestión de la Información de Usuarios se lograrían capturar la información exacta sobre todos los atributos que distinguen a los usuarios de cualquier institución, para el control, gestión y administración de dichos usuarios, independientemente del tipo de esta.”

Para esta respuesta anticipada al Problema Científico, se pueden identificar las **variables** siguientes:

1. Plataforma para la Gestión de la Información de Usuarios (variable independiente).
2. Gestión de la información referente a los usuarios, independientemente de la institución a la que estos pertenezcan (variable dependiente).

El **Objetivo General** de este trabajo es:

Realizar una propuesta de Plataforma para la Gestión de la Información de Usuarios que tenga aceptación en cualquier tipo de institución o empresa, de tal forma que reúna las características y expectativas para todas las ramas institucionales.

Los **Objetivos Específicos** son:

- Investigar acerca del desarrollo de las Plataformas para la Gestión de la Información de Usuarios en nuestro país y en el resto del mundo.
- Investigar sobre las diferentes tendencias y tecnologías más usadas y actuales para el desarrollo de una Plataforma para la Gestión de la Información de Usuarios.
- Realizar el análisis y diseño de una Plataforma para la Gestión de la Información de Usuarios que pueda integrarse con el resto de los módulos del Centro Rector del claustro de Universidad Para Todos.

Para poder llevar a feliz término los objetivos propuestos, se hace necesaria la realización de las siguientes **Tareas**:

- Realizar un estudio acerca del desarrollo de las Plataformas para la Gestión de la Información de Usuarios y hacer una descripción de ellas a nivel nacional y mundial
- Estudiar las tecnologías más usadas para la creación de Plataformas para la Gestión de la Información de Usuarios.
- Realizar el análisis y diseño de una Plataforma para la Gestión de la Información de Usuarios.

La realización de estas tareas y el consecuente logro de los objetivos propuestos darán como resultado un Servicio Web de gran usabilidad que facilitará la gestión de los usuarios de cualquier institución o empresa.

Para llevar a cabo las tareas propuestas anteriormente se usarán los siguientes **Métodos Científicos de la Investigación:**

## **Método(s) Teórico(s):**

- ❖ Analítico – Sintético.

Este método fue utilizado para analizar toda la información que pueda estar relacionada con el presente trabajo, dígame teoría, documentos, y cualquier otro material con el fin de extraer los elementos fundamentales que se relacionen con el objeto de estudio de esta investigación.

## **Método(s) Empírico(s):**

- ❖ Observación.

Se ha realizado una observación de la situación actual del tema de la investigación con el fin de acercarse al objetivo final que se persigue.

- ❖ Entrevista.

Fueron entrevistadas varias personas con el objetivo de que aportaran sus experiencias en el desarrollo de este tipo de aplicación y nos brindaran ayuda para llevar a cabo de manera correcta la presente investigación.

Todos los Métodos expuestos anteriormente han sido de gran ayuda para el desarrollo de este Trabajo.

## **Estructuración del contenido.**

El desarrollo de este trabajo consta de tres capítulos, de los cuales se hará una breve descripción a continuación:

Capítulo 1: en este capítulo se realizará una investigación acerca de los Sistemas de Gestión de Usuarios, su definición, tipología, y las ventajas de contar con una Plataforma para la Gestión de la Información de Usuarios para cualquier tipo de institución. Además se analizarán algunos sistemas de gestión de usuarios existentes en el mundo y en nuestro país, así como las tecnologías y tendencias más usadas para la creación de este tipo de sistemas.

Capítulo 2: este capítulo contará con una explicación detallada de cómo se realiza el proceso de negocio actualmente en diferentes instituciones y empresas, y se realizará el modelado de este a través de una herramienta CASE. Se realizará además la representación y descripción de los casos de uso del sistema y la propuesta del mismo llevando a cabo la captura de los requerimientos funcionales y no funcionales para poder, posteriormente, comenzar con el diseño del sistema propuesto.

Capítulo 3: en este capítulo se llevarán a cabo el análisis y diseño del sistema propuesto, utilizando para esto la metodología RUP. Se mostrarán los diagramas de clases del análisis y el diseño, así como los diagramas de interacción del análisis y las descripciones de las clases del diseño y de las tablas de la base de datos.

Capítulo 4: en este capítulo se realizará un estudio de factibilidad del proyecto para determinar si es conveniente llevarlo a cabo o no.

Este trabajo contará además con un Glosario de Términos para facilitar la comprensión de las palabras que puedan dar dificultades a quien les este documento.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.**

El vertiginoso desarrollo de las Tecnologías de la Información y las Comunicaciones ha hecho posible un avance paralelo de los medios de gestión de la información. Como consecuencia de esto y con la necesidad de un resultado positivo y eficaz a la hora de generalizar los procesos de gestión de la información de usuarios, se hace imprescindible el desarrollo de una plataforma que sea capaz de manipular toda la información referente a los usuarios de cualquier institución o empresa, es decir, que se adapte a las necesidades de información de los usuarios requerida por la empresa en la cual se aplique este sistema.

Este capítulo abordará acerca de las diferentes Plataformas de gestión de usuarios que existen en Cuba y el resto del mundo, sus características y las ventajas que estas proporcionan. Se realizará un estudio acerca de las distintas tecnologías que se usan en la actualidad para el desarrollo de este tipo de software, así como las metodologías usadas para este fin, con el objetivo de hacer una correcta selección de las mismas y de esta manera poder dar inicio a nuestro trabajo para poder terminarlo con la calidad requerida.

### **1.1 Plataforma para la Gestión de la Información de Usuarios. Definiciones.**

La Plataforma para la Gestión de la Información de Usuarios que se propone en este trabajo es un sistema informático basado en la tecnología de los Servicios Web, que es capaz de adaptarse a los requerimientos de información de usuarios de cualquier institución. Dicha plataforma cuenta con la capacidad de comunicarse con la(s) base(s) de datos existente(s) en la institución de manera tal que a la hora de procesar la información referente a los usuarios, esta información sea tomada de manera eficaz y no exista redundancia de información en la institución.

En informática, una plataforma es precisamente el principio, ya sea de hardware o software, sobre el cual un programa puede ejecutarse. Ejemplos típicos incluyen: arquitectura de hardware, sistema operativo, lenguajes de programación y sus librerías de tiempo de ejecución.

La gestión de la información de usuarios es el proceso mediante el cual toda la información de los usuarios registrados de una entidad es recopilada para luego ser procesada y de esta manera ayudar a los directivos de dicha entidad en la toma de decisiones, ya sean de tipo administrativa o de otro tipo.

## 1.2 Desarrollo y evolución de los Sistemas de Gestión de Usuarios.

Luego de haber hecho una búsqueda profunda respecto a los sistemas de gestión de usuarios se encontraron algunas aplicaciones que están dedicadas a la Gestión de Recursos Humanos, pero todas están ancladas a las empresas en las que se desempeñan, en las cuales gestionan la información, los reportes, aspectos organizativos, de selección y desempeño en los negocios, resaltando así su proyección administrativa.

- ✓ **Gestores relacionados en el mundo.**

### PayRoll RH



**Fig. 1 PayRoll RH**

Las características principales de este sistema son la administración de una base de datos completa con información de la empresa y del empleado, la disponibilidad de amplia información para realizar reportes de gestión gerencial y operativa, con múltiples salidas, la implantación de la Solución Integral de Gestión de Recursos Humanos de acuerdo a las necesidades del cliente, la migración de los datos históricos como parte de la implantación, la flexibilidad y escalabilidad modular, los campos libres definibles para

conceptos propios de la empresa y la administración de su personal, el apoyo informático para la negociación empresarial y de recursos humanos y tiene la mejor relación costo/beneficio. Teniendo en cuenta sus módulos a la hora de manejar los datos de los empleados, se obtienen las siguientes funcionalidades:

- Administración de Personal, Liquidación de Haberes, Capacitación y Planificación Orgánica.
- Control de Tiempo y Asistencia.
- Descripción de Puestos.
- Selección de Personal y Carga de CV vía web.
- Evaluación de Desempeño.
- Diagnóstico de Clima Organizacional.
- PayRoll Net / RRHH on line.
- Planeamiento de Carrera (en desarrollo).





Fig. 2 PayRoll RH

Simple HCM.

Las características principales de este sistema son que posee un poderoso y flexible generador de reportes, presenta notificador y manejo de cambios, presenta documentos en línea, es configurable, controla el acceso de directores y empleados, presenta un directorio telefónico, presenta un sistema de eventos por departamentos y realiza un seguimiento a información personal, de trabajo, compensación, educación, aptitudes y hoja de vida de los empleados. Teniendo en cuenta sus módulos a la hora de manejar los datos de los empleados, se obtienen los siguientes:

- Manejo de Beneficios.
- Evaluaciones 360°.
- Planeación de Salarios.

- Entrenamiento de Personal.
- Extractos de Compensación.
- Soporte de Pagos en línea.
- Selección de Personal y Seguimiento.

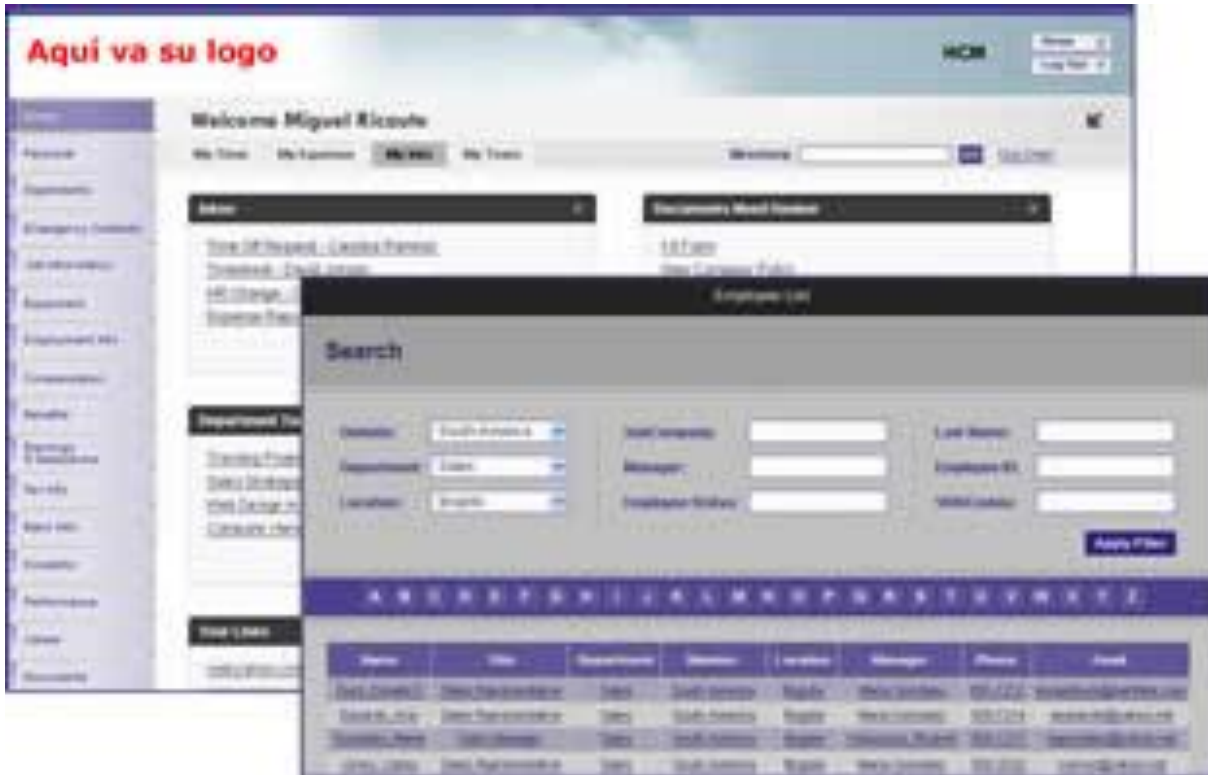


Fig. 3 Simple HCM

✓ **Gestores relacionados en Cuba.**

GREHU.

Las características principales de este sistema son que presenta una adaptabilidad a las particularidades propias de la instalación, comunicación con otros sistemas de la entidad a partir del desarrollo de interfaces, posibilidad de prever las posibles promociones, necesidades de formación y capacitación del personal, los reclutamientos futuros, etc., a partir del conjunto de salidas estadísticas, gráficas y

nominativas, control automático de asistencia por medio de relojes electrónicos, propone procedimientos y pautas a seguir para un mejor control y manipulación de los datos contenidos en el modelo principal del expediente laboral de los trabajadores, las actas de sanciones y medidas disciplinarias entre otros, relacionados con la nueva concepción del Inventario de Personal. Permite realizar preselección tanto de candidatos externos como de los trabajadores existentes según criterios preestablecidos para realizar promociones o nuevos contratos. Teniendo en cuenta sus módulos a la hora de manejar los datos de los empleados, se tiene que la aplicación es capaz de realiza las siguientes operaciones:

- Inventario de Personal
- Selección y Contratación
- Evaluación del desempeño
- Puesto y Case
- Dirigentes y Cuadros
- Capacitación del Personal
- Control de Asistencia
- Prenómina
- Nómina
- Propina Colectiva
- Estimulación en Divisa
- Protección e Higiene

**Especificación Generales de la Nómina**

*Última Nómina Procesada*

Desde: 01/12/2005    Hasta: 31/12/2005    Número de la Última Nómina: 32

**Impuestos**

- Contribución Especial (5/7 %)
  - No Aplicar a los Subsidios
- Aporte Seguridad Social (14%): 14,00
- Impuesto Utiliz. Fza Trabajo (25%): 25,00

**Vacaciones de**

Contratados	Capacitación
<input checked="" type="radio"/> Se Acumulan	<input checked="" type="radio"/> Se Acumulan
<input type="radio"/> Se Pagan	<input type="radio"/> No se Acumulan

Fig. 4 GREHU

**Especificaciones del Pago por Resultados**

*Tipo de Pago a Considerar*

Regular     Irregular

Aplicar CPL     Excluir por Ausencias  
 Ajustar CPL     Considerar Evaluación  
 Ajustar Importe     Incluir Bajas

Num. Última Nómina: 19     Liquidar Vacaciones

*Considerar*

Salario Escala     Total Devengado  
 Vacaciones     Licencia

*Incluir Además del Salario Escala*

Componentes de Salario	Pagos Realizados Por
<input type="checkbox"/> Condiciones Laborales	<input type="checkbox"/> Nocturnidad
<input type="checkbox"/> Horario Irregular	<input type="checkbox"/> Días Feriados
<input type="checkbox"/> Plus Salarial	<input type="checkbox"/> Horas Extras
<input type="checkbox"/> Antigüedad	<input type="checkbox"/> Doble Turno
<input type="checkbox"/> Idoneidad	<input type="checkbox"/> Otros Pagos
<input type="checkbox"/> Otros Pagos (Ctto)	

*Forma de Pago*

Sueldistas  
 Jornaleros  
 Vinculados

*Tipo de Trabajador*

Directos  
 Indirectos  
 Apoyos

Fig. 5 GREHU

Fig. 6 GREHU

Fig. 7 GREHU

### 1.3 Otros sistemas relacionados: SSO.

Otros sistemas relacionados con la Plataforma que se propone en este trabajo son los sistemas que utilizan SSO (Single Sign-On o Sistemas de Autenticación Reducida). SSO es un procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación y este es uno de los problemas que se pretende resolver con la Plataforma que se propone en este trabajo. (RAJEEV ANGAL, y otros, 2005)

A continuación se muestran algunos sistemas que utilizan este procedimiento:

### SAML Single Sign-On (SSO) Service para las Aplicaciones de Google.

Las Aplicaciones de Google ofrecen un servicio SAML (Security Assertion Markup Language) SSO a sus compañías asociadas y de esta manera les ofrece un control total sobre la autorización y la autenticación de los usuarios registrados, que de esta manera pueden acceder a los servicios web como Gmail o el Calendario de Google. Usando el modelo SAML Google actúa solamente como el prestatario de los servicios y los socios de Google como los abastecedores de identidad y controlan los nombres de usuario, las contraseñas y toda otra información que pueda ser usada para identificar a los usuarios para que puedan acceder a los servicios que oferta Google.

La siguiente figura muestra como funciona este sistema:

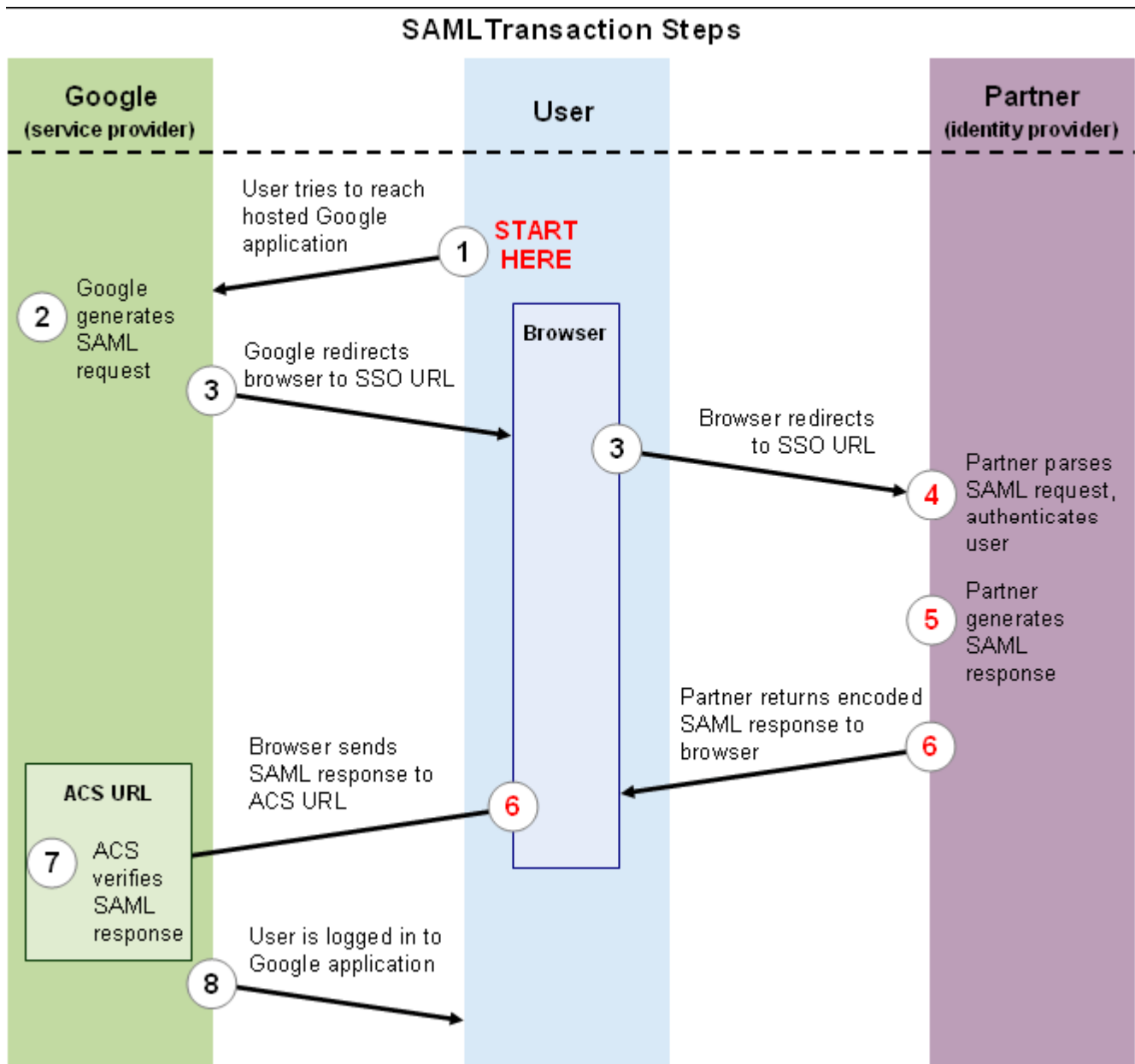


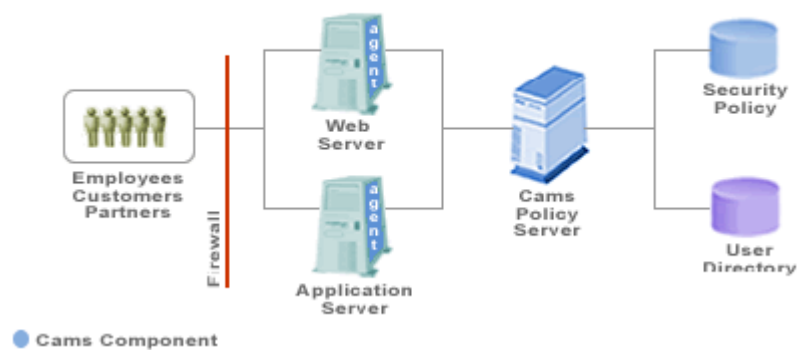
Fig. 8 SAML

Cams:

Cams es un software seguro, flexible y asequible basado en Web-SSO que controla de manera centralizada el acceso a servidores de aplicaciones entre otras cosas. Cams mejora la seguridad y la manejabilidad de los sitios web ya que centraliza la autenticación, el control de acceso y la administración sin necesidad de instalar una plataforma de seguridad. Además elimina la necesidad de que los desarrolladores hagan módulos de autenticación y control de acceso en ausencia del código de las páginas web. Este enfoque centralizado de la seguridad permite reducir la complejidad de la administración así como los costos.

A continuación se muestra la arquitectura de este sistema:

**Cams Architecture**



**Fig. 9 Cams**

Esta aplicación, aunque es libre, no es factible con el propósito de este trabajo ya que se encuentra programada en JAVA y no está basada en la tecnología de los Servicios Web.

#### **1.4 Necesidad de una Plataforma para la Gestión de la Información de Usuarios.**

La gestión de información acerca de los usuarios se convierte en una necesidad social tanto en el mundo como en nuestro país a partir del momento en que la información constituye el recurso más fuerte e influyente en los asuntos de negocios y desarrollo de las empresas e instituciones. Considerando el avance producido constantemente en el mundo de las tecnologías para la gestión de usuarios, y el efecto que podría producir un sistema práctico sobre las necesidades de obtener información de manera rápida,



organizada, precisa y correcta nos hace comprender la necesidad que presentan las empresas de dicho sistema a la hora de alcanzar sus metas.

Los sistemas de gestión de la información de usuarios han seguido un proceso ininterrumpido de desarrollo que se inició en edades muy tempranas, continuando más adelante en los distintos niveles de exigencias y manteniéndose hasta los días de hoy brindando la posibilidad de poder manejar, controlar y organizar grandes cantidades de información, de satisfacer las peticiones de los usuarios, tanto como almacenar sus datos, que en la mayoría de las ocasiones son volúmenes de información muy extensos.

Teniendo en cuenta lo antes planteado nos percatamos de que se hacen completamente imposibles y poco útiles los procesos utilizados en los inicios de la existencia de estos sistemas, pues la gestión adecuada de la información o las peticiones de los usuarios manualmente no serían rápidas y eficientes.

Mirando la realidad del mundo, las empresas e instituciones están obligadas a utilizar sistemas de gestión de la información de usuarios a disposición de los mismos usuarios y directivos, que les permitan el acceso a las fuentes de información y sistemas actuales, facilitando la información confiable y precisa.

La necesidad de contar con una Plataforma para la Gestión de la Información de Usuarios está dada por la gran usabilidad que esta posee, pues el sistema que se propone en este trabajo es un sistema capaz de adaptarse a los requerimientos de cualquier institución o empresa y a partir de ahí comenzar el procesamiento de los datos de los usuarios. La propuesta de esta Plataforma implica, además, una nueva manera de organización de los datos de los usuarios y del trabajo, puesto que se eliminaría la redundancia de información y las búsquedas de datos serían más rápidas y eficaces lo cual daría al traste con un sustancial ahorro de tiempo.

### **1.5 Ventajas de contar con una Plataforma para la Gestión de la Información de Usuarios.**

Contar con una Plataforma para la Gestión de la Información de Usuarios reportaría grandes ventajas tanto para las instituciones que las utilicen como para los administradores y desarrolladores de Sistemas de Gestión de la Información de Usuarios. A continuación se mencionan las más importantes:

- Se llevaría al máximo exponente la capacidad de las instituciones y empresas de gestionar los datos de sus usuarios y trabajadores.

- Se lograría una mayor organización, integridad y centralización de la información, lo cual facilitaría la manipulación de la misma, y esto, a su vez posibilitaría un mejor y más detallado análisis de los datos de los usuarios.
- Se lograría un sustancial ahorro de tiempo y trabajo a la hora de agregar, modificar o suprimir algún tipo de dato de los que se recogen de los usuarios, ya que al contar con una Plataforma solamente se tendrían que introducir los cambios deseados y esta automáticamente actualizaría la base de datos del sistema.
- Las búsquedas de datos serían más eficaces y rápidas ya que la Plataforma almacenaría las búsquedas recientes en una caché y de esta manera no tendría que realizarla cada vez que se invoque una búsqueda realizada anteriormente.

### **1.6 Metodologías, Tecnologías y Software utilizados.**

#### 1.6.1 RUP.

La metodología utilizada para el desarrollo de esta aplicación fue RUP (Rational Unified Process), pues la misma es una metodología escalable al estar formada por fases y estas a su vez por flujos de trabajo, lo cual hace que el proceso de desarrollo del software sea iterativo e incremental, y junto con UML constituye la metodología estándar más utilizada en el mundo para el análisis, implementación y documentación de sistemas orientados a objetos. RUP describe como utilizar de forma efectiva reglas de negocio y procedimientos probados en el desarrollo de software para equipos de desarrollo de software, de forma disciplinada y asignando tareas y responsabilidades con la finalidad de tener una interacción transparente con los clientes para lograr su satisfacción una vez obtenida la Plataforma para la Gestión de la Información de Usuarios.

#### 1.6.2 UML.

UML (Lenguaje Unificado de Modelado). Para la modelación de la Plataforma para la Gestión de la Información de Usuarios fue utilizado este lenguaje por su sencillez y porque en la actualidad es el lenguaje de modelado más conocido y utilizado en el mundo, y cuenta con el respaldo de la OMG (Object Management Group). El mismo es un lenguaje gráfico que se utiliza para visualizar, especificar, construir y documentar sistemas de software. Ofrece también un estándar para describir el plano o

modelo del software que se va a implementar e incluye aspectos conceptuales, tales como procesos de negocios y funciones del sistema. Además contempla aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software que puedan ser reutilizables. (ESPERANZA MARCOS M., 2004)

### 1.6.3 Visual Paradigm.

Como herramienta CASE (Computer-Aided Software Engineering) para el modelado de la Plataforma para la Gestión de la Información de Usuarios se usó el Visual Paradigm ya que esta es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. La misma ayuda a una construcción de aplicaciones de más calidad y a un menor coste. También permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación, además de ser una herramienta libre y funcionar tanto en Linux como en Windows. (2007)

## **1.7 Tecnologías que se proponen para el futuro desarrollo e implementación de la Plataforma para la Gestión de la Información de Usuarios.**

### 1.7.1 Servicio Web.

Se propone la tecnología de Servicio Web debido a que el mismo está formado por un conjunto de aplicaciones o tecnologías con capacidad para interoperar en la Web, además de proporcionar mecanismos de comunicación estándar entre varias aplicaciones que interactúan entre sí para presentar información de forma dinámica al usuario, proporcionando interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas. Las funcionalidades de estos se han incrementado aún más debido a la utilización de estándares como XML para el intercambio de información estructurada entre diferentes plataformas, independiente del tipo de sistema operativo o plataforma en que se ejecuten. Estos juegan un papel importante en la actualidad, debido a las necesidades presentadas por las empresas, tanto de brindar sus servicios como obtener información a través del inmenso flujo que circula por internet mediante aplicaciones empresariales, las cuales son invadidas por un nivel de exigencias en espiral impuestos por las grandes y poderosas corporaciones existentes a nivel mundial. La gran importancia y utilidad

que presentan los Servicios Web es que son entidades que se programan proporcionando elementos de funcionalidades determinados, como lógica de aplicación, donde podemos acceder a él desde distintos sistemas, sin importar cuán diferentes y potenciales sean los mismos, mediante estándares de internet muy extendidos y conocidos como XML y HTTP, y sin importar cuál sea el lenguaje de programación, plataforma de implementación o sistema operativo, siendo contenedores de cápsulas con funciones específicas, todo esto dejando como ventaja que los todas sus funciones puedan ser utilizadas en cualquier servidor.

### 1.7.2 XML.

Se propone el uso de XML ya que este es un lenguaje de metamarcado definido por la W3C (World Wide Web Consortium) que ofrece un formato para la descripción de datos estructurados lo cual facilita y hace más precisas las declaraciones de contenido y hace más significativos los resultados de búsquedas en varias plataformas, a la vez que garantiza que los datos estructurados sean uniformes e independientes de aplicaciones o fabricantes. XML es muy valioso para internet, así como para los entornos de intranets corporativas de gran tamaño, pues proporciona interoperabilidad mediante un formato basado en estándares flexible y abierto, con nuevas formas de acceso a las bases de datos existentes y de entregar datos a clientes de Web. Las aplicaciones se pueden generar más rápidamente, su mantenimiento es más sencillo y pueden ofrecer fácilmente varias vistas de los datos estructurados.

### 1.7.3 SOAP.

Se propone el uso de SOAP, el cual es un protocolo estándar creado por Microsoft, IBM y otros, y está actualmente bajo el auspicio de la W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Actualmente SOAP es el núcleo de los Servicios Web y es el primer protocolo de su tipo que ha sido aceptado prácticamente por todas las grandes compañías de software del mundo debido a las múltiples ventajas que presenta, algunas de las cuales se mencionan a continuación:

- No está asociado con ningún lenguaje: Los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista pero los desarrolladores responsables de mantener antiguas aflicciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo

que la implementación de la API se deja al lenguaje de programación, como en Java, y la plataforma como Microsoft .Net.

- No se encuentra fuertemente asociado a ningún protocolo de transporte: La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- No está atado a ninguna infraestructura de objeto distribuido: La mayoría de los sistemas de objetos distribuidos se pueden extender, y ya lo están algunos de ellos para que admitan SOAP.
- Aprovecha los estándares existentes en la industria: Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.
- Permite la interoperabilidad entre múltiples entornos: SOAP se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dichos estándares pueden comunicarse mediante mensajes SOAP con aplicaciones que se ejecuten en otras plataformas. Por ejemplo, una aplicación de escritorio que se ejecute en una PC puede comunicarse con una aplicación del back-end ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP.

#### 1.7.4 WSDL.

Se propone el uso de WSDL (Web Services Description Language), un formato XML propuesto por la W3C para la descripción de Servicios Web. Podríamos decir que WSDL es el manual de operación del Servicio Web porque indica cuales son las interfaces que provee el Servicio Web y los tipos de datos necesarios para la utilización del mismo. WSDL es usado a menudo en combinación con SOAP y XML Schema. De esta manera, un cliente que se conecta a un Servicio Web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Una de las ventajas WSDL es que permite

separar la descripción abstracta de la funcionalidad ofrecida por un Servicio, de los detalles concretos del mismo, como el enlace a un protocolo de red o formato de mensaje concreto que puede ser SOAP, HTTP u otro.

### 1.7.5 Servidor Web Apache.

Como servidor web se propone el uso de Apache ya que este es uno de los más destacados dentro de la gran cantidad de servidores web que existen en el planeta, y esto se debe a que es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache presenta, entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido; aunque fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Este servidor web posee una amplia aceptación en la red: en 2005, Apache fue el servidor HTTP más usado en el mundo, siendo empleado en el 48% de los sitios web que se encuentran en el planeta. Entre sus principales ventajas destacan que es un sistema modular, open source, multi-plataforma, extensible, gratuito y que es muy fácil conseguir ayuda o soporte debido a su gran popularidad.

### 1.7.6 MySQL.

Como Sistema Gestor de Bases de Datos se propone el uso de MySQL puesto que este sistema es muy utilizado en aplicaciones web y su popularidad como aplicación web está muy ligada a PHP como lenguaje a utilizar, gracias a su escalabilidad es un gestor muy rápido en la lectura y manipulación de grandes base de datos, funciona sobre múltiples plataformas, sistemas operativos posibilitando y mostrando su gran conectividad entre diferentes máquinas, incluso siendo sus sistemas operativos diferentes y ha sido probado sobre muchos compiladores diferentes. Está escrito en los lenguajes C y C++, presenta un sistema de privilegios y contraseñas con mucha flexibilidad y seguridad. En el mundo existe una gran cantidad de información referente a este gestor, lo cual sirve de apoyo a los que lo usan.

### 1.7.7 Lenguaje de programación PHP.

Como lenguaje de programación se propone el uso de PHP por las ventajas y funcionalidades que presenta para la creación de los servicios web, pues es soportado en una gran cantidad de servidores,

plataformas y gestores de bases de datos, posibilitando y asegurando una migración futura por parte de los usuarios, teniendo como punto de partida la posibilidad de utilizar la capa de abstracción de datos *ADODB*, y su programación del lado del servidor, además de profundizar en las magnificas ventajas de su integración con el Apache como servidor a utilizar y el MySQL como manejador de base de datos. Tiene un gran parecido con los lenguajes estructurados como C y Perl, es compatible con aplicaciones de contenido dinámico sin tener que llegar a aprenderse nuevas funciones. Es de libre acceso a todos, contando con una gran cantidad de información relacionada y de códigos de ejemplos.

Llegado a este punto, se puede concluir que con el diseño de la Plataforma para la Gestión de la Información de Usuarios se pretende dar una gran mejoría al proceso de gestión de usuarios, puesto que la Plataforma corregiría muchos problemas existentes en la actualidad en aplicaciones similares tanto en el ámbito nacional como el internacional, tales como la redundancia y dispersión de información, agilidad y eficacia en los procesos de búsqueda, y otros. Otra de las ventajas de contar con la Plataforma es que esta no estaría diseñada para gestionar la información de usuarios de una institución en específico, sino que podría ser utilizada indistintamente en cualquier lugar que requiera de este servicio.

Se puede concluir además que la Plataforma que se propone contaría con gran usabilidad ya que podría ser utilizada sobre cualquier sistema operativo, a su vez que está diseñada para ser desarrollada usando herramientas libres.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### 2.1 Problema y situación problemática.

#### 2.1.1 Objetivos estratégicos de la organización y proceso de negocio que los soportan.

- Mejorar la forma en la que se realiza la gestión de usuarios y eliminar la redundancia de información existente.
- Guardar la información de manera organizada, para que a la hora de realizar una búsqueda sea fácil encontrar los resultados que se desean.

Actualmente el proceso de gestión de la información de los usuarios se encuentra muy desorganizado, es por eso que los procesos se hacen tan lentos y en algunos casos hasta se cometen errores a la hora de realizar cualquier operación en la que estén involucrados los datos de los usuarios, y de esta manera es que se encuentran soportados los objetivos estratégicos.

#### 2.1.2 Flujo actual de los procesos involucrados en el campo de acción.

El objetivo de la gestión de la información de usuarios, en cualquier institución o empresa, es recoger y organizar la información que se tiene de los mismos para que de esta manera la toma de decisiones por parte de los directivos sea más eficaz y cuente con bases sólidas.

Este proceso se realiza de la siguiente forma:

**CU Gestionar Usuario:** se inicia cuando por alguna razón se necesita información referente a alguno de los trabajadores (usuarios) de la institución, el funcionario(a) obtiene los datos buscados, que puede ser cualquiera que esté en la(s) base(s) de datos de dicha institución, obtenida la información se acaba el proceso, si el dato incorporado al sistema por el funcionario(a) está incorrecto la Plataforma no mostrará nada, por lo que se debe incorporar de forma correcta.

#### 2.1.3 Análisis crítico de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problemática y las consecuencias.

El proceso de gestión de la información de usuarios se encuentra en la actualidad muy desorganizado, ya que podemos encontrar la misma información repetida varias veces en distintos



lugares de la institución, lo cual provoca que los mismos no se encuentren bien actualizados y puedan existir diferencias entre ellos. De igual manera, los sistemas de gestión de usuarios existentes solo satisfacen las necesidades de las empresas para las que fueron creados, lo cual se traduce en que son sistemas con una portabilidad escasa o nula prácticamente, siendo estas algunas de las causas que originan la situación problemática que da lugar a esta investigación.

### **2.2 Objeto de Automatización.**

#### 2.2.1 Descripción de los procesos que serán objeto de automatización.

Los procesos que serán objeto de automatización en este trabajo son básicamente los procesos de gestión de la información de usuarios, dichos procesos incluyen actividades tales como:

- Autenticar usuario.
- Añadir un usuario al sistema.
- Eliminar un usuario del sistema.
- Modificar los datos de un usuario.
- Realizar búsquedas de usuarios mediante el uso de parámetros.
- Obtener la ficha de un usuario.

#### 2.2.2 Descripción de los sistemas automatizados que están vinculados con el campo de acción.

La Plataforma que se propone interactuará con diferentes sistemas y bases de datos que existan en la institución en la cual esté funcionando. Estas aplicaciones ofrecerán a los usuarios servicios diversos, pero para acceder a ellos el usuario deberá estar registrado y autenticado en el sistema, operación que se realizará a través de la Plataforma, que es la encargada de la gestión de los usuarios, y de acuerdo con sus privilegios será el nivel de acceso que se le conceda.

### **2.3 Información que se maneja.**

#### 2.3.1 Documentos específicos que se procesen, detalles de la información que se manipule.

Para realizar el proceso de gestión de la información relativa a los usuarios se manejan datos personales como Carné de Identidad, nombre completo, dirección particular, número de teléfono, correo electrónico, y otros datos profesionales como la ocupación, el nivel escolar vencido, salario básico, etc.; los cuales se encuentran almacenados en bases de datos existentes en la institución.

### **2.4 Propuesta de Sistema.**

#### 2.4.1 Descripción general de la propuesta de sistema, cómo debe funcionar.

El sistema que se propone es una Plataforma para la Gestión de la Información de Usuarios. Dicha plataforma debe ser capaz de funcionar en cualquier tipo de institución o empresa que necesite de un sistema de este tipo. La tecnología que se propone para su implementación es la de los Servicios Web, puesto que hoy en día los clientes son muy inestables en cuanto a lo que desean como tecnología de desarrollo y esta es la tecnología más aceptada pues en caso de hacer una migración, el sistema no sufriría cambios estructurales, sino que solo se cambiaría la interfaz y todos los demás componentes y paquetes de la Plataforma quedarían intactos.

Esta Plataforma propone un nuevo método de trabajo a la hora de gestionar los datos de los usuarios, puesto que contaría con una pequeña base de datos en la cual almacenaría solamente los datos esenciales de los usuarios (dígase el identificador del usuario, el nombre de usuario y la contraseña), y los otros datos referentes a los usuarios de la institución o empresa los tomaría de la(s) base(s) de datos que exista(n) en la aplicación o aplicaciones que existan en dicha institución o empresa, con la(s) cual(es) interactuaría.

El funcionamiento de la Plataforma que se propone es bien básico, puesto que ella se encargaría solamente de autenticar a los usuarios de la institución y una vez realizada esta operación, les concedería el acceso a los servicios que brinden las aplicaciones que existan en la institución en la cual se encuentre funcionando la plataforma. Además, una vez autenticado el usuario, la Plataforma debe recuperar la información restante de ese usuario y de esta manera permitir realizar búsquedas de los datos de todos los usuarios de la institución. Estos datos ajenos a la Plataforma, se manejarían en forma de estructuras e irían de la aplicación a la plataforma en formato XML.

### 2.4.2 Análisis comparativo de otras soluciones existentes con la propuesta, llegando a conclusiones sobre los aspectos en los que esta propuesta se diferencia de las otras y las supera.

De acuerdo con los ejemplos vistos en el capítulo anterior y después de haberlos analizado en profundidad se decidió descartar esos softwares porque ninguno de ellos está implementado usando la tecnología de los Servicios Web y se encuentran muy ligados a las empresas en las que funcionan, además, en su mayoría son software propietario lo cual va en contra de la filosofía que se busca con la implementación de este sistema.

También se analizó el sistema de gestión de usuarios de la UCI y este análisis arrojó como resultado que ese sistema se encuentra muy desorganizado, ya que existe mucha información repetida en distintas bases de datos de la institución, lo cual trae consigo que algunos de los mismos tengan diferentes niveles de actualización y no coincidan, incluso siendo de la misma persona. Esto además conlleva a que las búsquedas sean lentas y en ocasiones no proporcionen resultados.

## **2.5 Modelo de dominio.**

Dado que el negocio que se está estudiando presenta un bajo nivel de estructuración y que se encuentra altamente centrado en tecnologías informáticas se propone un modelo de dominio, puesto que este permite mostrar al usuario de manera visual los principales conceptos que se manejan en el dominio del sistema que se propone. Esto ayuda a los usuarios, clientes, desarrolladores y personal interesado a utilizar un vocabulario común para poder entender el contexto en el cual se encuentra emplazado el sistema que se propone. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

A continuación se realiza la descripción del modelo de dominio mediante un diagrama de clases UML en el cual se especifican las principales clases conceptuales que pueden intervenir en el sistema, estas representarán los objetos que existen o eventos que suceden en el entorno en el que se desarrollará el mismo. Se procederá a explicar los principales conceptos que se identifican en este modelo.

**Plataforma:** esta clase se refiere como tal a la Plataforma que se propone en este trabajo y es quien se va a encargar de autenticar a los usuarios que intenten acceder a los servicios de la aplicación externa, recuperar los datos de los usuarios y realizar búsquedas de los datos de usuario.

**Aplicación:** esta clase se refiere a las aplicaciones que se encuentran en la institución en la cual se monte la Plataforma, dicha clase cuenta con una o varias bases de datos en las cuales se encuentran almacenados los datos adicionales de los usuarios, y va a ser accedida por los usuarios que se autentican en la Plataforma.

**Trabajador:** esta clase representa a todas las personas que pertenecen a la institución, dichas personas pueden o no interactuar directamente con el sistema.

**Usuario:** esta clase agrupa a todas las personas (trabajadores) que van a interactuar de una forma u otra con la Aplicación y que se van a autenticar en la Plataforma.

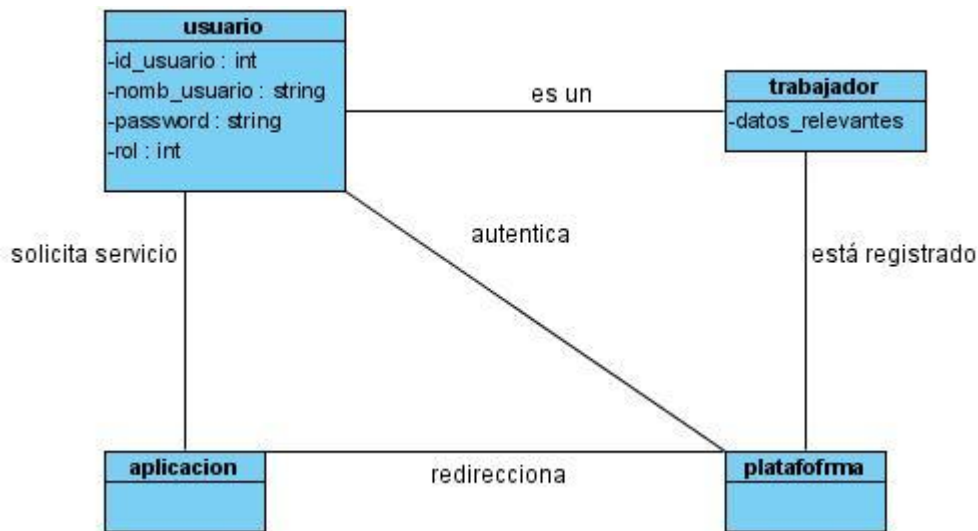


Fig. 10 Modelo de Dominio

## 2.6 Especificación de los requisitos de software.

### 2.6.1 Requerimientos Funcionales.

RF1 – Autenticar usuario.

RF2 – Gestionar usuario.

RF2.1 – Adicionar un usuario al sistema.

RF2.2 – Modificar datos de un usuario.

RF2.3 – Eliminar un usuario del sistema.

RF3 – Obtener ficha de usuario.

RF4 – Buscar información de usuarios y guardar en caché.

## 2.6.2 Requerimientos no Funcionales.

### 2.6.2.1 Software.

- El Servicio Web requiere tener instalado el lenguaje de programación PHP 5.0.
- El Servicio Web utilizará como sistema gestor de bases de datos MySQL.
- Se utilizará como servidor web Apache 2.0.

### 2.6.2.2 Seguridad y privacidad.

- Se establecerá una llave privada entre el cliente y el proveedor del Servicio Web, para evitar acceso no autorizado a los servicios.

### 2.6.2.3 Portabilidad.

- Se podrá acceder a los servicios desde cualquier protocolo de transporte que sea capaz de transportar texto (Ejemplo: HTTP, HTTPS, SMTP, etc.).
- El Servicio Web estará diseñado para soportar su funcionalidad en sistemas operativos tanto libres como propietarios.

### 2.6.2.4 Hardware.

- El sistema requerirá como mínimo 512 MBytes de RAM.
- El sistema requerirá un espacio mínimo de 5 GBytes en disco duro para almacenar la Base de Datos.

### 2.6.2.5 Restricciones en el diseño y la implementación.

- Lenguaje de programación PHP.
- Librería NuSoap.
- Librería ADOdb.

2.6.2.6 Usabilidad.

- El Servicio Web podrá ser accedido desde diferentes plataformas de desarrollo pues su protocolo de comunicación está basado en XML, que es compatible con todas las plataformas.

2.6.2.7 Legales.

- El Servicio Web y toda la documentación generada pertenecen al proyecto del Centro Rector del Claustro de Universidad para Todos y la Universidad de las Ciencias Informáticas.

2.6.2.8 Confiabilidad.

- El Servicio Web debe estar disponible las 24 horas de forma tal que se pueda acceder a todas sus funcionalidades.

**2.7 Definición de los actores y casos de uso del sistema.**

Utilizando las facilidades que nos brinda el lenguaje de modelado UML, a continuación se representarán los requerimientos funcionales del sistema mediante diagramas de casos de uso. Para ello hay que definir primeramente, de acuerdo a lo planteado en los epígrafes anteriores, cuáles serían los actores que van a interactuar con el sistema y los casos de uso que van a representar las funcionalidades.

2.7.1 Actores del sistema.

Actor del sistema	Justificación
Usuario	Es quien invoca los servicios que ofrecen la Plataforma y la aplicación, se autentica en la Plataforma y en dependencia de su rol tendrá privilegios para la administración de la misma, es decir, que podrá adicionar, eliminar o modificar los datos de los usuarios.

2.7.2 Casos de uso del sistema.

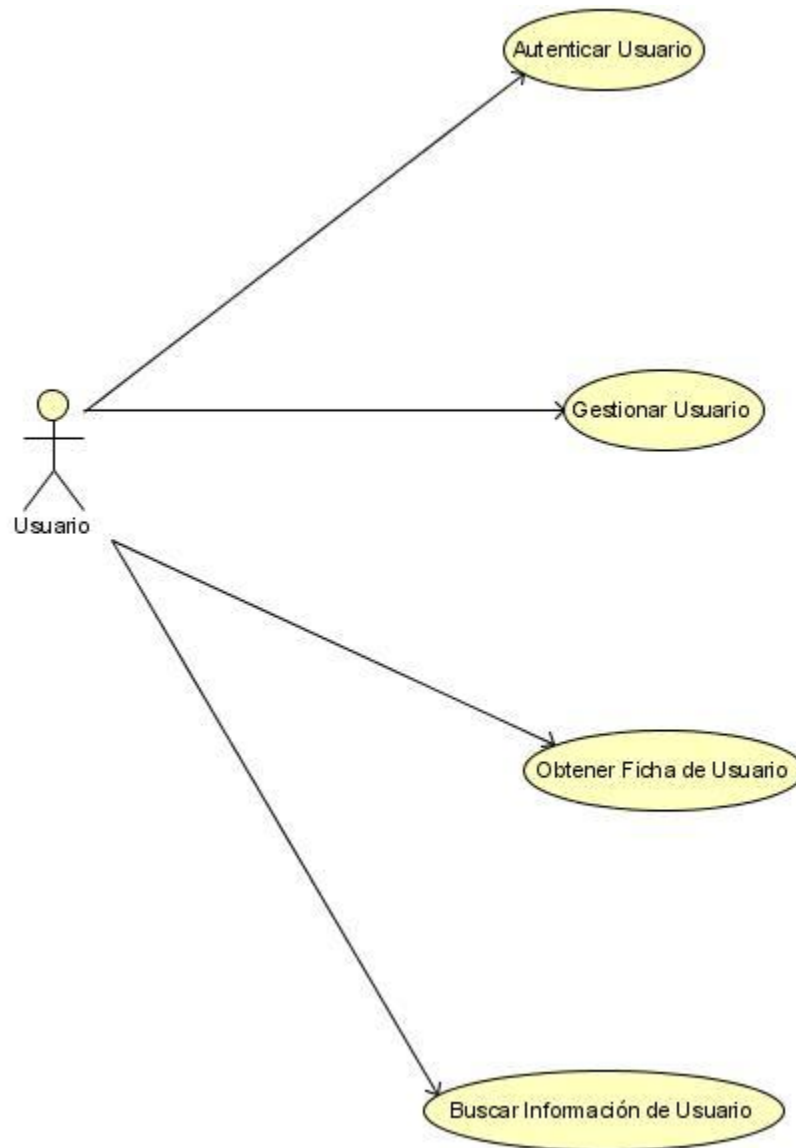


Fig. 11 Diagrama de Casos de Uso del Sistema

2.7.3 Breve descripción de los Casos de Uso del Sistema.

CU-1	Autenticar Usuario
Actor(es)	Usuario

## CARACTERÍSTICAS DEL SISTEMA

Descripción	Este Caso de Uso se inicia cuando el usuario va a acceder a uno de los servicios de la aplicación y es redireccionado a la Plataforma para que introduzca su nombre de usuario y contraseña.
Referencia	RF1

CU-2	Gestionar Usuario
Actor(es)	Usuario
Descripción	Este Caso de Uso se inicia cuando el Usuario con rol de Administrador decide adicionar un nuevo usuario, modificar los datos de alguno existente, o eliminar alguno existente en el sistema. Si decide adicionar uno se invoca el servicio Adicionar Usuario, Usuario Administrador envía los datos del nuevo usuario y se le envía un mensaje informando que la operación se realizó correctamente; si lo que se desea es modificar los datos o eliminar algún usuario existente, el Usuario Administrador debe enviar el identificador del usuario y solicitar el servicio que se requiere y luego la Plataforma le envía un mensaje diciendo que la operación fue realizada correctamente.
Referencia	RF2

CU-3	Obtener Ficha de Usuario
Actor(es)	Aplicación
Descripción	Este Caso de Uso se inicia cuando el Usuario solicita el servicio, el cual permite mostrar la ficha por la cual fue registrado un usuario cualquiera, finalizando el Caso de Uso de esta manera.
Referencia	RF3



CU-4	Buscar Información de Usuario
Actor(es)	Aplicación
Descripción	Este caso se inicia cuando el Usuario invoca el servicio que permite obtener la información de los usuarios. Estas búsquedas son avanzadas y se realizan filtrando los parámetros enviados a la Plataforma, la cual envía un listado de datos con el resultado de la búsqueda y de esta manera finaliza el Caso de Uso.
Referencia	RF4

2.7.4 Descripción expandida de los Casos de Uso del Sistema.

CU-1	Autenticar Usuario	
Propósito	Dar al usuario los permisos que le tocan una vez este se haya autenticado.	
Actor(es)	Usuario	
Resumen	Este Caso de Uso se inicia cuando el usuario accede a uno de los servicios de la aplicación y es redireccionado a la Plataforma para que introduzca su nombre de usuario y contraseña, y una vez realizada esta operación, en dependencia de la veracidad de los datos le será concedido el acceso o no al servicio solicitado.	
Referencias	RF1	
	Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> <li>1. El actor intenta acceder a la aplicación.</li> <li>3. El actor introduce los datos (nombre de usuario y contraseña).</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema le muestra un formulario solicitando los datos de autenticación.</li> <li>4. El sistema verifica la autenticidad de los datos.</li> <li>5. Muestra la página de inicio del usuario</li> </ol>

## CARACTERÍSTICAS DEL SISTEMA

	autenticado en dependencia de sus privilegios.
4.* Error en los datos (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje indicando que existen errores en los datos.</li> <li>2. El sistema envía al actor de vuelta al paso 2 del Flujo Normal de Eventos.</li> </ol>
Prioridad	Crítico

CU-2	Gestionar Usuario
Propósito	Adicionar un usuario al sistema, modificar los datos de alguno existente, o eliminar usuarios del sistema.
Actor(es)	Usuario
Resumen	Este Caso de Uso se inicia cuando el Usuario con rol de Administrador decide adicionar un nuevo usuario, modificar los datos de alguno existente, o eliminar alguno existente en el sistema. Si decide adicionar uno se invoca el servicio Adicionar Usuario, Usuario Administrador envía los datos del nuevo usuario y se le envía un mensaje informando que la operación se realizó correctamente; si lo que se desea es modificar los datos o eliminar algún usuario existente, el Usuario Administrador debe enviar el identificador del usuario y solicitar el servicio que se requiere y para terminar, la Plataforma le envía un mensaje diciendo que la operación fue o no realizada correctamente.
Referencias	RF4

## CARACTERÍSTICAS DEL SISTEMA

Acción del Actor	Respuesta del Sistema
<p>1. El Usuario Administrador selecciona uno de los servicios disponibles para la gestión de los usuarios.</p> <p>a) Si elige adicionar un usuario, ir al escenario “Adicionar Usuario”.</p> <p>b) Si elige modificar los datos de un usuario, ir al escenario “Modificar Datos de Usuario”.</p> <p>c) Si elige eliminar un usuario, ir al escenario “Eliminar Usuario”.</p>	
Escenario “Adicionar Usuario”	
Acción del Actor	Respuesta del Sistema
<p>1. a) El Usuario Administrador llama el servicio “Adicionar Usuario”.</p>	<p>2. El sistema muestra un formulario para que sean introducidos los datos de los usuarios.</p> <p>3. El sistema verifica que los datos obligatorios hayan sido enviados.</p> <p>4. El sistema verifica que los datos enviados sean correctos.</p> <p>5. El sistema verifica que el número de carnet enviado sea único.</p> <p>6. El sistema almacena los datos del usuario en el sistema.</p> <p>7. El sistema retorna un mensaje informando que el usuario fue adicionado con éxito.</p>

## CARACTERÍSTICAS DEL SISTEMA

	Finaliza el Caso de Uso.
3.* Hay campos obligatorios que no han sido llenados (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> <li>1. El sistema envía un mensaje indicando que hay algún campo obligatorio que no ha sido llenado.</li> <li>2. El sistema envía al actor de vuelta al paso 2 del Flujo Normal de Eventos.</li> </ol>
4.* Error en los datos (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> <li>1. El sistema envía un mensaje indicando que hay datos incorrectos.</li> <li>2. El sistema envía al actor de vuelta al paso 2 del Flujo Normal de Eventos.</li> </ol>
5.* El número de carnet ya existe en el sistema (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> <li>1. El sistema envía un mensaje indicando que ya el usuario existe en el sistema. Finaliza el Caso de Uso.</li> <li>2. El sistema envía al actor de vuelta al paso 2 del Flujo Normal de Eventos.</li> </ol>
Escenario "Modificar Datos de Usuario"	
Acción del Actor	Respuesta del Sistema
1. b) El Usuario Administrador llama el servicio "Modificar Datos de Usuario", el	2. El sistema muestra un formulario para introducir el nombre de usuario del usuario

## CARACTERÍSTICAS DEL SISTEMA

<p>cual permite modificar todos los datos de los usuarios del sistema.</p> <p>3. El Usuario Administrador introduce el nombre de usuario del usuario que se va a modificar.</p> <p>5. El Usuario Administrador introduce los nuevos datos.</p>	<p>que se desea modificar.</p> <p>4. El sistema busca el usuario y muestra un formulario para que sean introducidos los nuevos datos.</p> <p>6. El sistema verifica que los campos obligatorios hayan sido llenados.</p> <p>7. El sistema verifica que los datos enviados estén correctos.</p> <p>8. El sistema actualiza los cambios en la base de datos. Finaliza el Caso de Uso.</p>
4.* Error en el nombre de usuario (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del sistema
	<p>1. El sistema muestra un mensaje indicando que el nombre de usuario es incorrecto o no existe en el sistema y envía al actor al paso 4 del Flujo Normal de Eventos.</p>
6.* Hay campos obligatorios vacíos (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del sistema
	<p>1. El sistema muestra un mensaje indicando que hay campos obligatorios que no han sido llenados y envía al actor de vuelta al paso 4 del Flujo Normal de Eventos.</p>
7.* Error en los datos (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del sistema
	<p>1. El sistema muestra un mensaje indicando</p>

## CARACTERÍSTICAS DEL SISTEMA

	que los datos son incorrectos y envía al actor de vuelta la paso 4 del Flujo Normal de Eventos.
<b>Escenario “Eliminar Usuario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. c) El Usuario Administrador llama el servicio “Eliminar Usuario” y selecciona el usuario que se va a eliminar.	2. El sistema elimina el usuario solicitado. 3. El sistema actualiza los cambios en la base de datos. 4. El sistema retorna un mensaje informando que el usuario fue eliminado correctamente.
<b>Flujo Alternativo de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Prioridad</b>	<b>Crítico</b>

<b>CU-3</b>	Obtener Ficha de Usuario
<b>Propósito</b>	Mostrar todos los datos de un usuario.
<b>Actor(es)</b>	Aplicación
<b>Resumen</b>	Este Caso de Uso se inicia cuando el usuario solicita el servicio, el cual permite mostrar la ficha por la cual fue registrado un usuario, finalizando el Caso de Uso de esta manera.
<b>Referencias</b>	RF3
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

## CARACTERÍSTICAS DEL SISTEMA

<p>1. El usuario invoca el servicio “Obtener Ficha de Usuario”, y envía el nombre de usuario del usuario del cual quiere obtener la ficha.</p>	<p>2. El sistema verifica el nombre de usuario en el sistema.</p> <p>3. El sistema busca todos los datos pertenecientes al usuario con dicho nombre de usuario.</p> <p>4. El sistema retorna la ficha del usuario y la guarda en el sistema en forma de estructura. Finaliza el caso de uso.</p>
<p>2.* Error en el nombre de usuario (Flujo Alternativo de Eventos)</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>1. El sistema envía un mensaje indicando que el nombre de usuario no existe o está incorrecto.</p>
<p>Prioridad</p>	<p>Crítico</p>

<p>CU-4</p>	<p>Buscar Información de Usuario</p>
<p>Propósito</p>	<p>Brindar las opciones que permitan buscar información de cualquier usuario del sistema.</p>
<p>Actor(es)</p>	<p>Aplicación</p>
<p>Resumen</p>	<p>Este caso se inicia cuando el usuario invoca el servicio que permite obtener la información de los usuarios. Estas búsquedas son avanzadas y se realizan filtrando los parámetros enviados a la Plataforma, la cual envía un listado de datos con el resultado de la búsqueda y de esta manera finaliza el Caso de Uso.</p>
<p>Referencias</p>	<p>RF4</p>

## CARACTERÍSTICAS DEL SISTEMA

Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> <li>1. El usuario invoca el servicio.</li> <li>3. El usuario envía los parámetros por los cuales desea buscar información.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema muestra un formulario par que se introduzcan los parámetros de la búsqueda.</li> <li>4. El sistema verifica que al menos uno de los parámetros de la búsqueda haya sido enviado.</li> <li>5. El sistema verifica que los datos estén correctos.</li> <li>6. El sistema realiza una búsqueda por filtrado de los usuarios existentes en la(s) base(s) de datos que contengan todos o algunos de los parámetros.</li> <li>7. El sistema muestra un listado de los usuarios con los datos por los cuales se realizó la búsqueda, y muestra además algún otro dato de importancia. Finaliza el Caso de Uso.</li> </ol>
4.* No hay parámetros para la búsqueda (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje que indica que al menos debe haber un parámetro de búsqueda y envía al actor al paso 2.</li> </ol>
5.* Error en los datos (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> <li>1. El sistema devuelve un mensaje indicando</li> </ol>



## CARACTERÍSTICAS DEL SISTEMA

	que los parámetros son incorrectos y envía al actor de vuelta al paso 2.
7.* La búsqueda no produjo resultados (Flujo Alternativo de Eventos)	
Acción del Actor	Respuesta del Sistema
	1. El sistema retorna un mensaje indicando que no existen usuarios que posean tales criterios de búsqueda. Finaliza el Caso de Uso.
Prioridad	Crítico

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.

### 3.1 Análisis.

#### 3.1.1 Definición del modelo de análisis. Modelo de clases de análisis.

El proceso de análisis consiste en obtener una visión del sistema que permita entender qué hay que hacer, por lo que sólo se interesa por los requisitos funcionales. Se identifican las clases que describen la realización de los Casos de Uso, los atributos y las relaciones entre ellas, necesarias para la construcción del modelo de análisis.

Se elaborará un diagrama de clase de análisis por cada Caso de Uso. Un diagrama de clases de análisis es un artefacto en el que se representan los conceptos en un dominio del problema y representa las cosas del mundo real, no de la implementación automatizada.

Para facilitar el análisis, las clases se clasifican de la siguiente forma:

#### 3.1.2 Diagramas de clases de análisis.

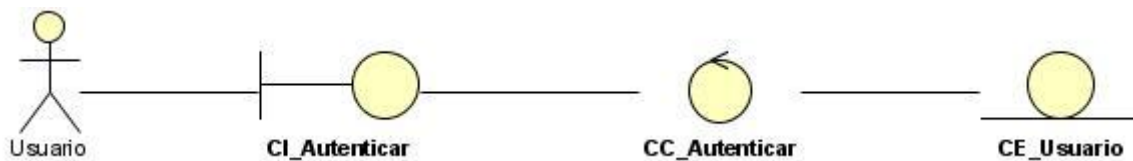


Fig. 12 Diagrama de clases de análisis del Caso de Uso "Autenticar Usuario"

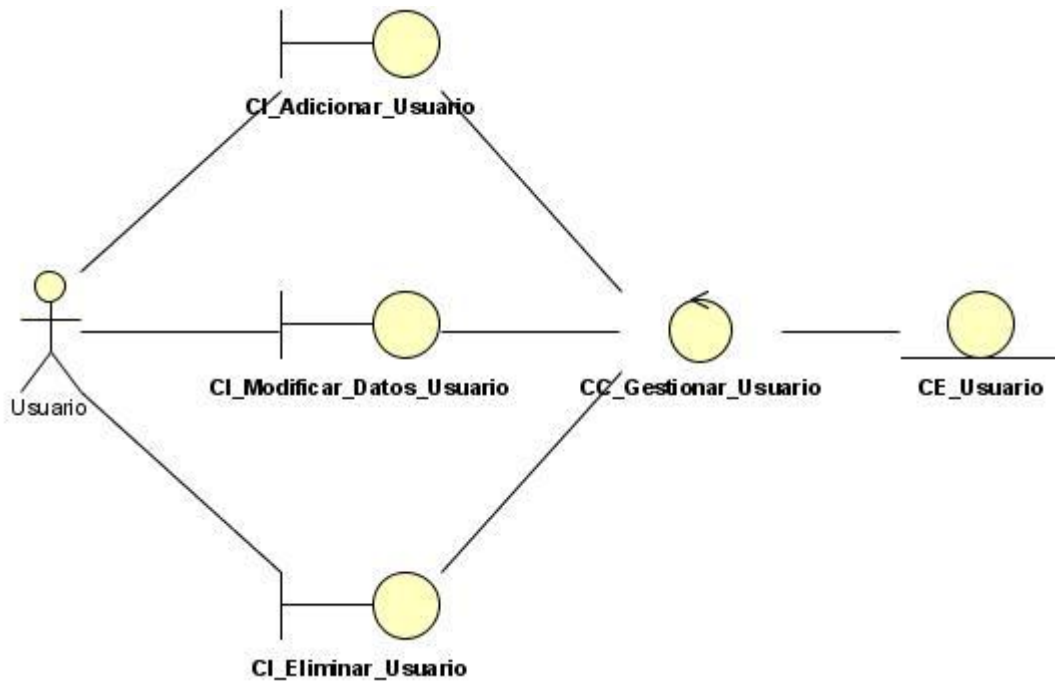


Fig. 13 Diagrama de clases de análisis del Caso de Uso "Gestionar Usuario"

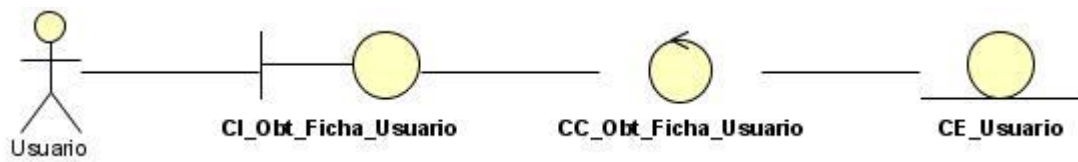


Fig. 14 Diagrama de clases de análisis del Caso de Uso "Obtener Ficha de Usuario"

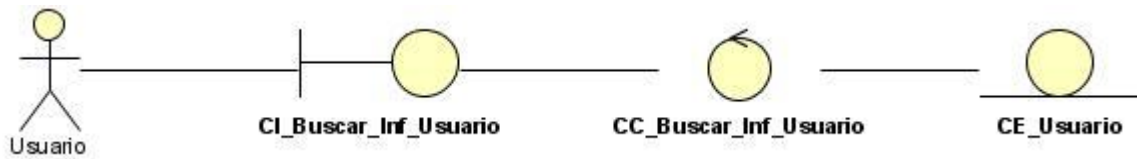


Fig. 15 Diagrama de clases de análisis del Caso de Uso "Buscar Información de Usuario"

## 3.1.3 Diagramas de interacción.

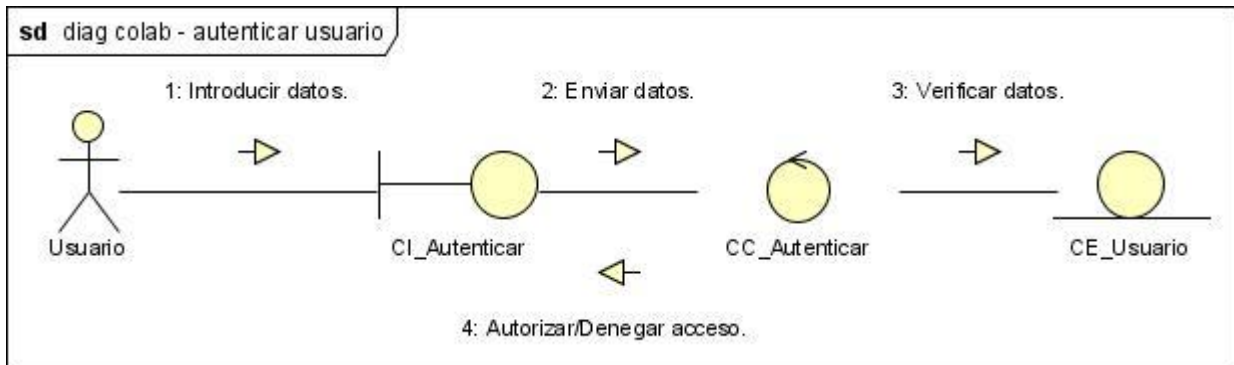


Fig. 16 Diagrama de colaboración del Caso de Uso "Autenticar Usuario"

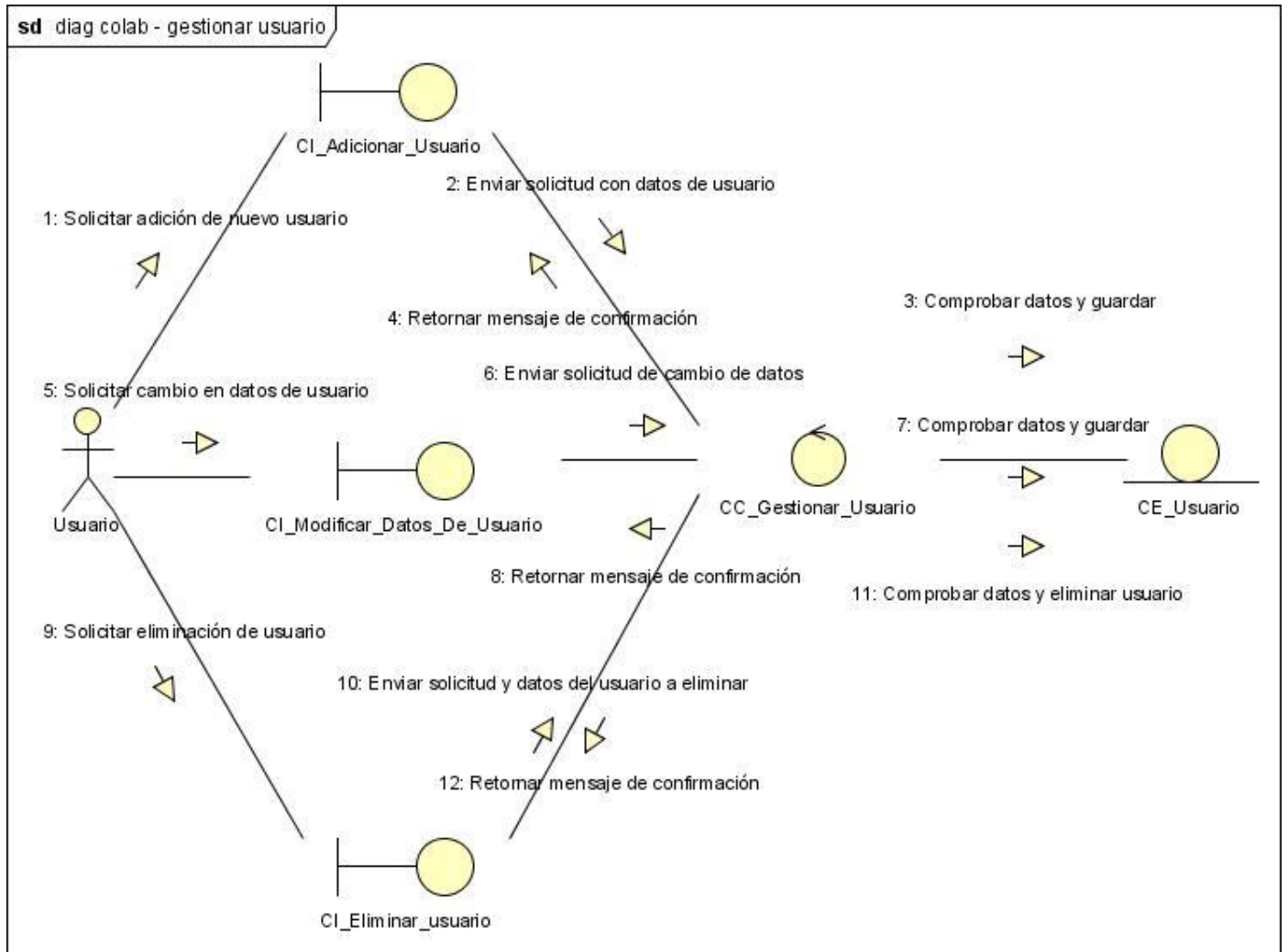


Fig. 17 Diagrama de colaboración del Caso de Uso “Gestionar Usuario”

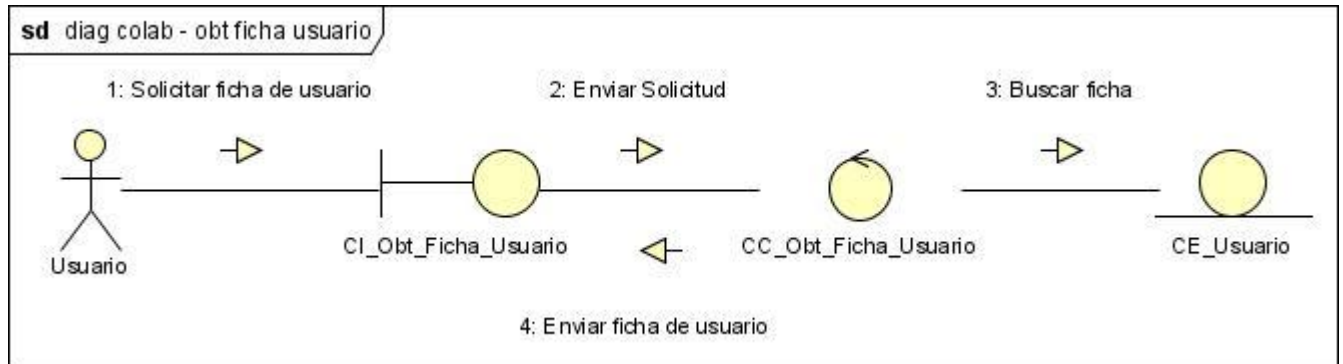


Fig. 18 Diagrama de colaboración del Caso de Uso "Obtener Ficha de Usuario"

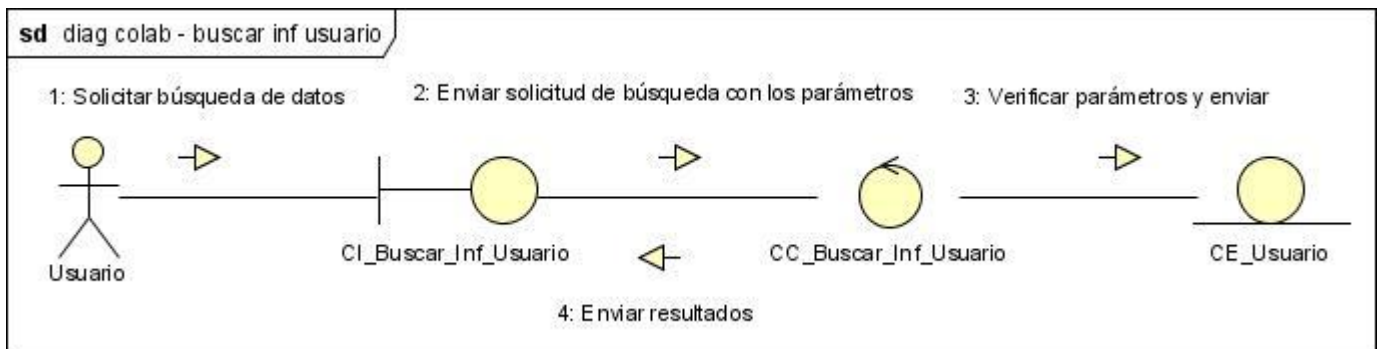


Fig. 19 Diagrama de colaboración del Caso de Uso "Buscar Información de Usuario"

### 3.2 Diseño.

A través del flujo de trabajo de Diseño, RUP permite utilizar las ventajas de la Programación Orientada a Objetos (POO). La POO es un paradigma de la programación que define los programas en términos de "clases de objetos", estos objetos son entidades que combinan estado (datos), comportamiento (procedimientos o métodos) e identidad (propiedades de un objeto que lo diferencia de los otros). La POO expresa un programa como un conjunto de estos objetos que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

En esta sección se presentan los resultados obtenidos luego de haber realizado, a partir del Análisis, el diseño del sistema propuesto. Estos resultados se refieren específicamente a los diagramas de interacción, los diagramas de clases del diseño, la descripción de las clases y el modelo de datos.

### 3.2.1 Diagramas de clases de diseño.

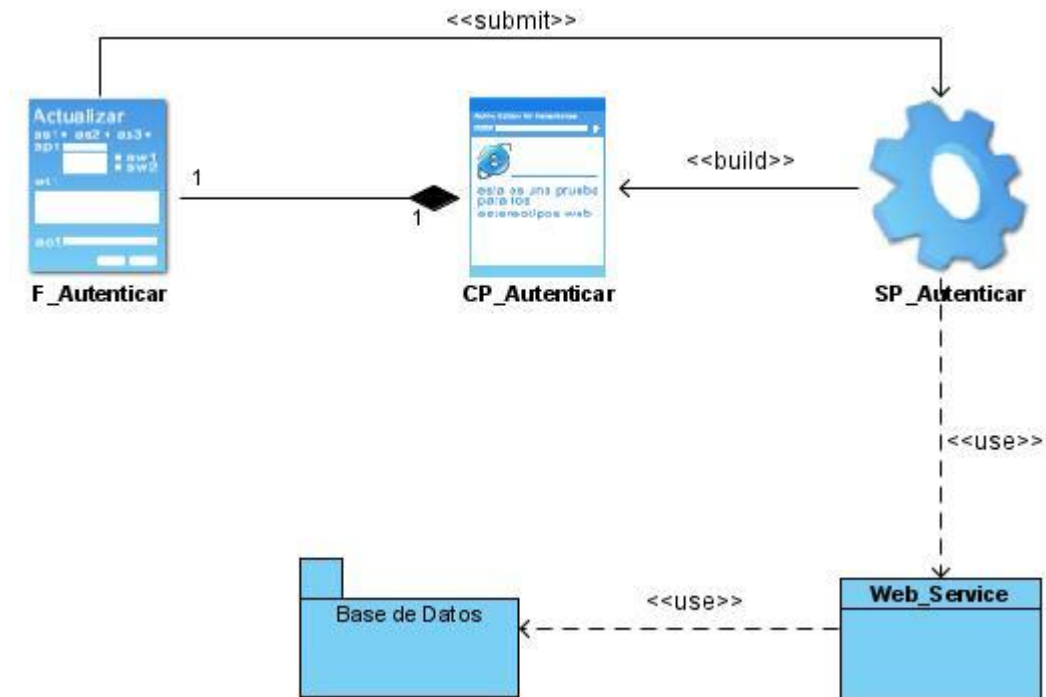


Fig. 20 Diagrama de clases de diseño del Caso de Uso "Autenticar Usuario"

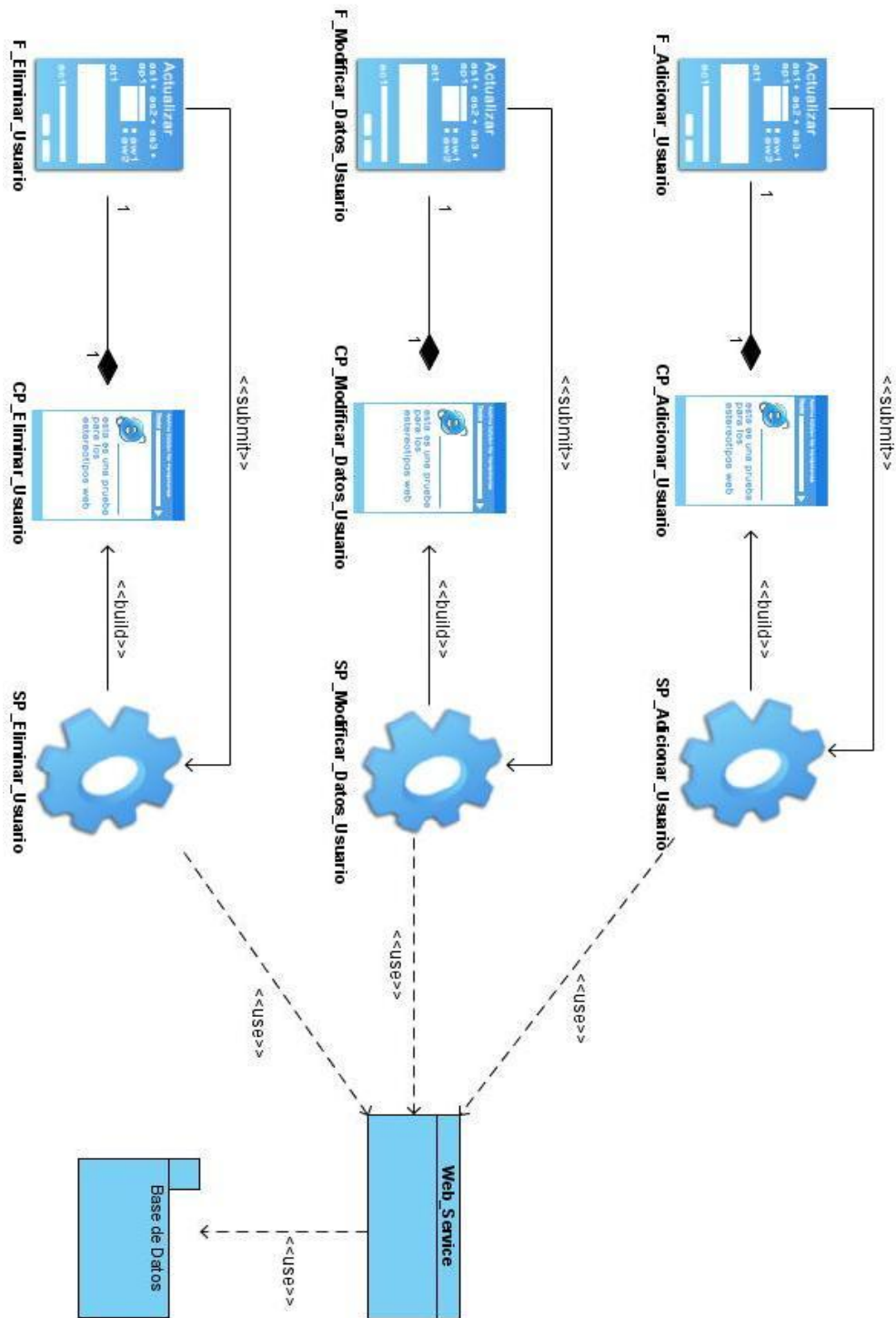


Fig. 21 Diagrama de clases de diseño del Caso de Uso "Gestionar Usuario"



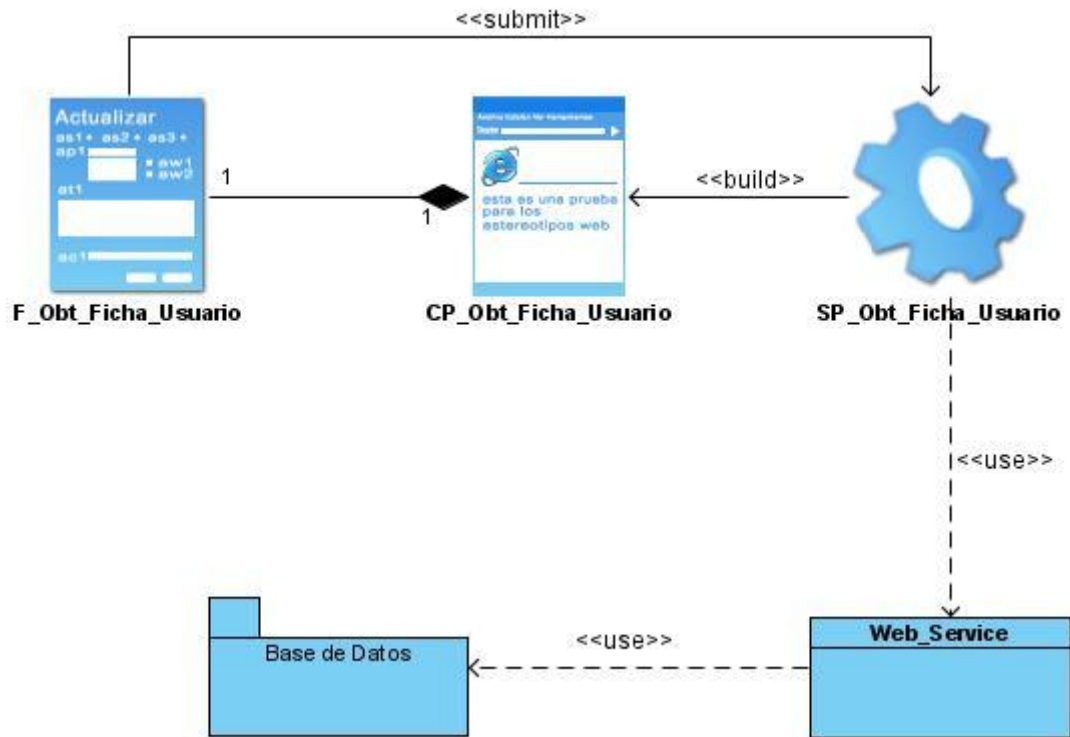


Fig. 22 Diagrama de clases de diseño del Caso de Uso "Obtener Ficha de Usuario"

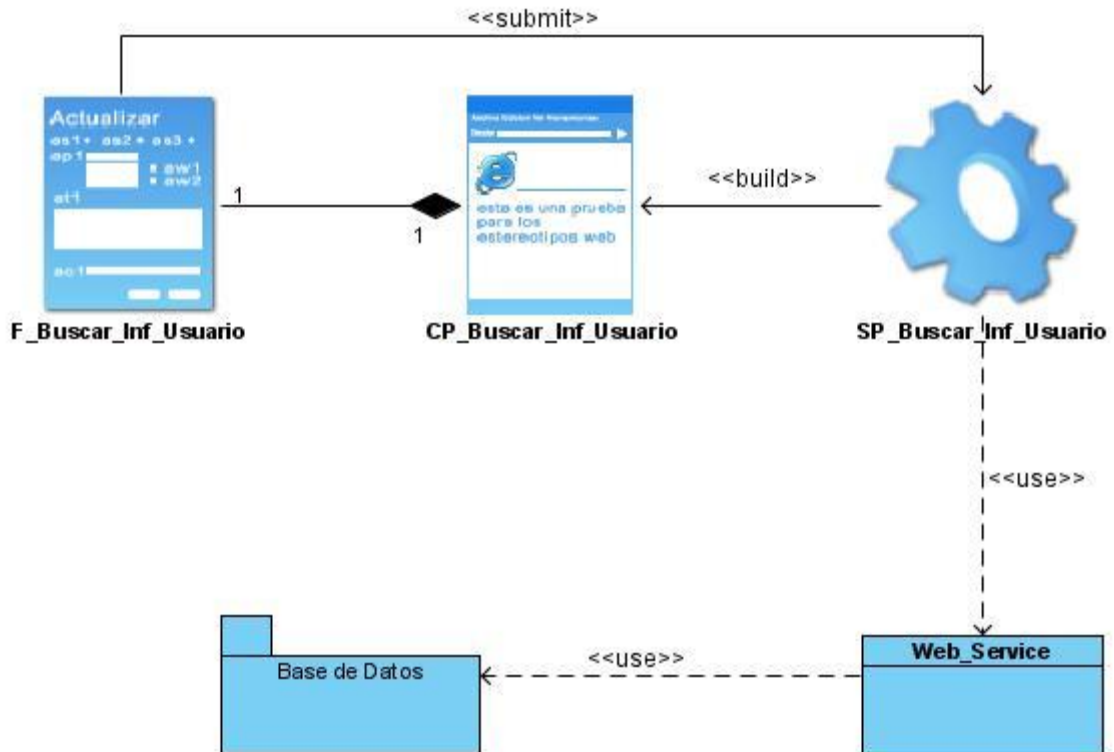


Fig. 23 Diagrama de clases de diseño del Caso de Uso “Buscar información de Usuario”

3.2.2 Descripción de las clases.

Nombre.	SP_Autenticar	
Tipo de clase.	Controladora	
Atributo.	Tipo.	
Usuario	Varchar	
Contraseña	Varchar (encriptado MD5)	
Para cada responsabilidad:		
Nombre.	Crear_Sesion()	
Responsabilidad.	Este método crea una sesión en el navegador.	

Nombre.	Verificar_Usuario()
Responsabilidad.	Verifica que el usuario que acceda al sistema exista y que sus datos de autenticación estén correctos.

Nombre.	SP_Adicionar_Usuario	
Tipo de clase.	Controladora	
Atributo.	Tipo.	
Usuario	Varchar	
Contraseña	Varchar (encriptado MD5)	
Repetir contraseña	Varchar (encriptado MD5)	
Rol	Int	
Para cada responsabilidad:		
Nombre.	Adicionar_Usuario(usuario)	
Responsabilidad.	Adiciona un usuario al sistema.	
Nombre.	Verificar_Datos_Usuario()	
Responsabilidad.	Verifica que los datos de los usuarios estén correctos.	

Nombre.	SP_Modificar_Datos_Usuario	
Tipo de clase.	Controladora	
Atributo.	Tipo.	
Usuario	Varchar	
Nuevos Datos		

Para cada responsabilidad:	
Nombre.	Modificar_Datos_Usuario()
Responsabilidad.	Este método modifica los datos que hayan sido seleccionados de los usuarios del sistema.
Nombre.	Verificar_Datos_Usuario()
Responsabilidad.	Verifica que los nuevos datos de los usuarios hayan sido introducidos de manera correcta.

Nombre.	SP_Eliminar_Usuario	
Tipo de clase.	Controladora	
Atributo.	Tipo.	
Usuario	Varchar	
Para cada responsabilidad:		
Nombre.	Eliminar_Usuario(nomb_usuario)	
Responsabilidad.	Este método elimina permanentemente un usuario del sistema.	
Nombre.	Verificar_Usuario()	
Responsabilidad.	Verifica que el usuario exista en la base de datos del sistema para después ser eliminado.	

Nombre.	SP_Obt_Ficha_Usuario	
Tipo de clase.	Controladora	
Atributo.	Tipo.	
Usuario	Varchar	

Para cada responsabilidad:	
Nombre.	Obt_Ficha_Usuario(nomb_usuario)
Responsabilidad.	Este método devuelve todos los datos de un usuario registrado.
Nombre.	Verificar_Usuario()
Responsabilidad.	Verifica que el usuario exista en el sistema.

Nombre.	SP_Buscar_Inf_Usuario	
Tipo de clase.	Controladora	
Atributo.	Tipo.	
Usuario	Varchar	
Para cada responsabilidad:		
Nombre.	Buscar_Inf_Usuario()	
Responsabilidad.	Este método devuelve un listado con los usuarios que tengan datos coincidentes con los que se pasen por parámetro.	

Nombre.	Web_Service	
Tipo de clase.	Controladora	
Atributo.	Tipo.	
Nombre.	Actualizar_Datos_Usuario(id_usuario)	
Descripción.	Este método actualiza todos los datos de un usuario.	
Nombre.	Buscar_Usuario()	

Descripción.	Este método busca y devuelve un usuario en el sistema.
Nombre.	Eliminar_Usuario(id_usuario)
Descripción.	Este método elimina un usuario del sistema.
Nombre.	Devolver_Arreglo_Usuarios()
Descripción.	Esta función devuelve un arreglo con los usuarios que poseen cierto criterio de búsqueda.
Nombre.	Vaciar_Bd_Usuario()
Descripción.	Esta función elimina todos los registros de la base de datos Usuario.

### 3.2.3 Diagramas de interacción.

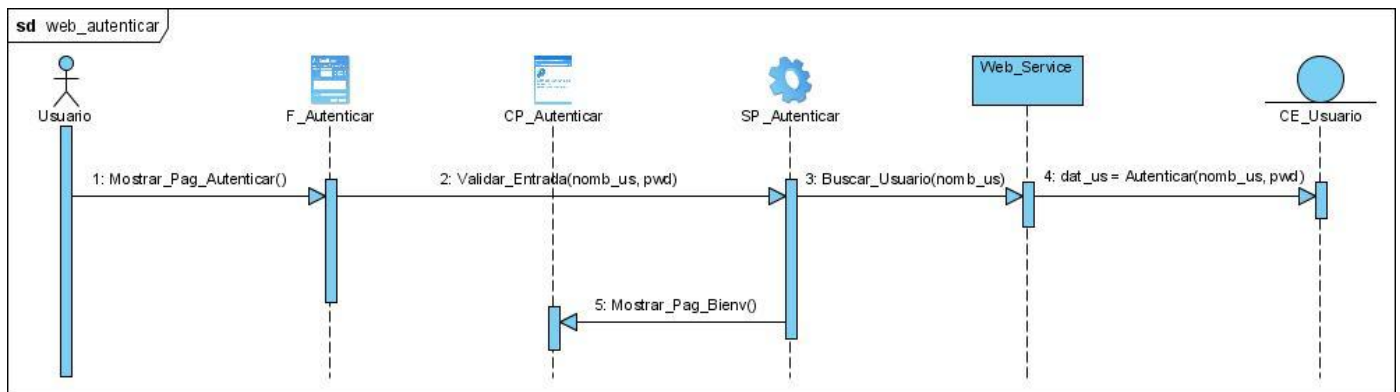


Fig. 24 Diagrama de secuencia del caso de uso "Autenticar Usuario"

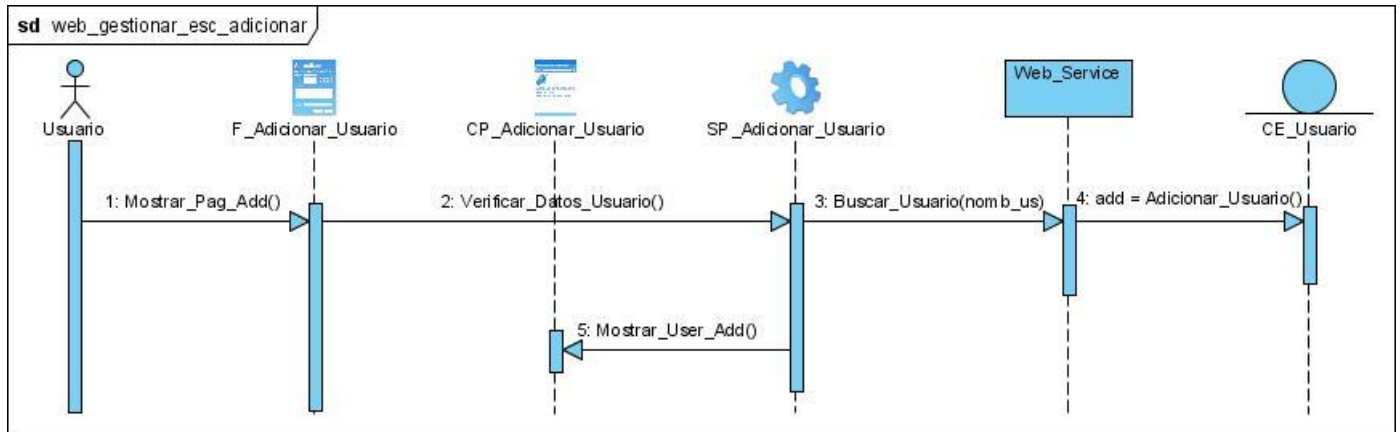


Fig. 25 Diagrama de secuencia del caso de uso "Gestionar Usuario". Escenario "Adicionar".

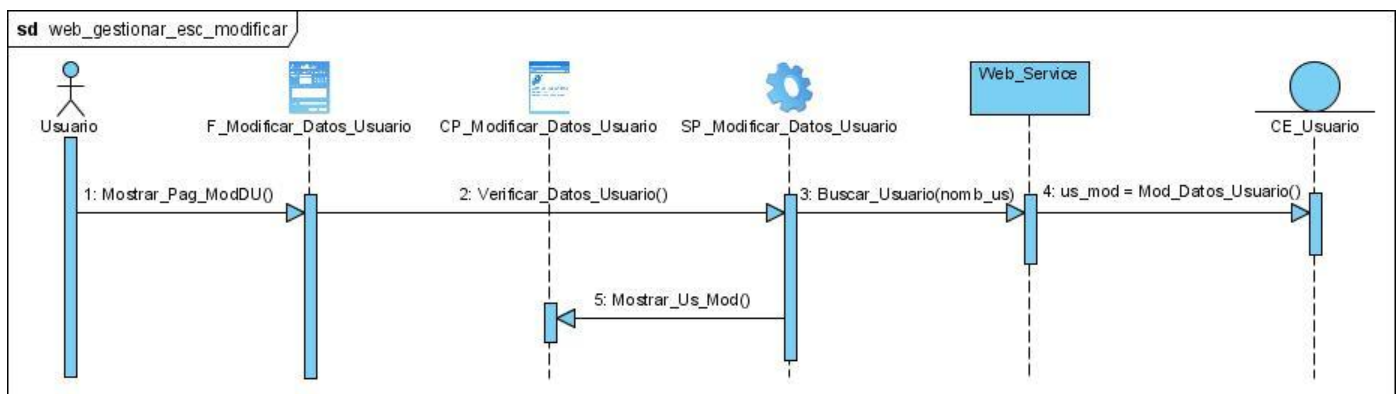


Fig. 26 Diagrama de secuencia del caso de uso "Gestionar Usuario". Escenario "Modificar".

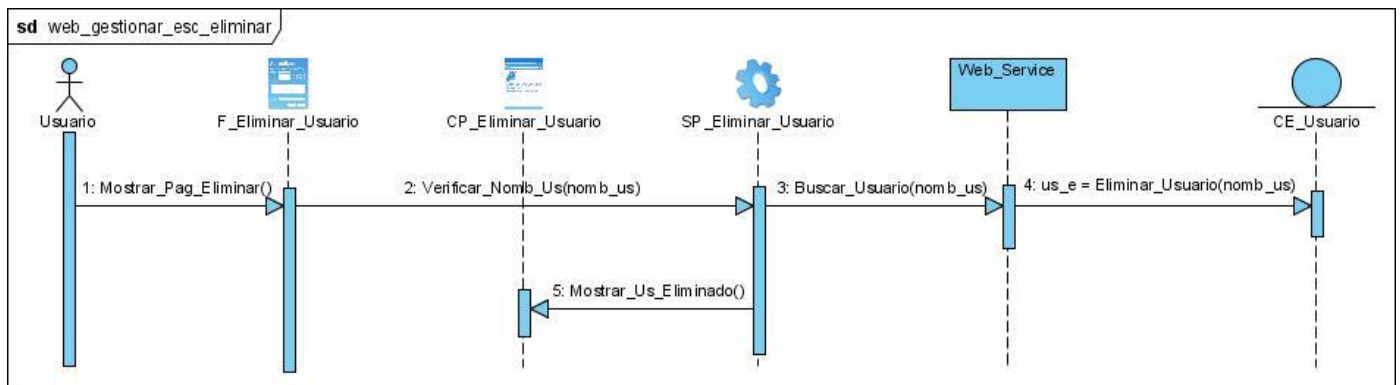


Fig. 27 Diagrama de secuencia del caso de uso "Gestionar Usuario". Escenario "Eliminar".

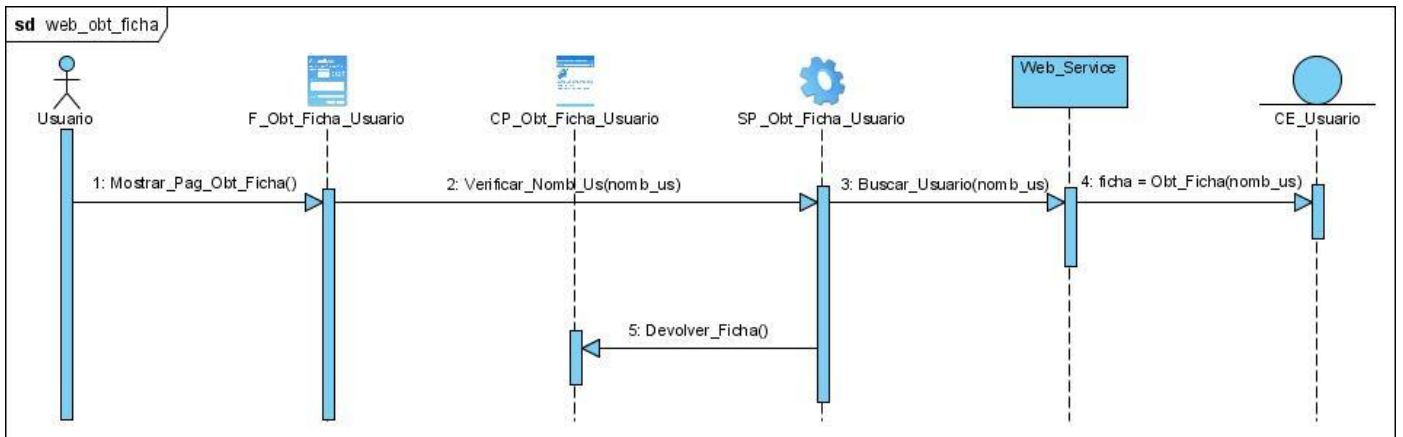


Fig. 28 Diagrama de secuencia del caso de uso "Obtener Ficha de Usuario"

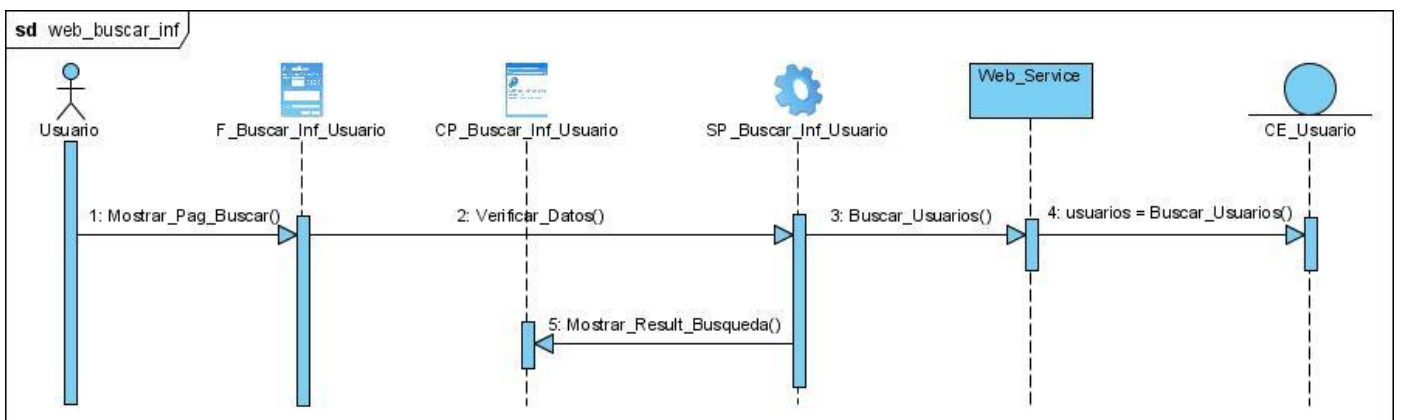


Fig. 29 Diagrama de secuencia del caso de uso "Buscar Información de Usuario"

### 3.2.4 Diseño de la Base de Datos.

A continuación se presenta el diseño de la base de datos, modelo que es representado a partir del diagrama de clases persistentes, una vez que estas hayan sido seleccionadas. Estas clases son las que representan todos los datos que se obtienen y almacenan durante todo el ciclo de vida del producto que se propone, y las relaciones que existen entre ellos.

A partir del diagrama de clases persistentes se presenta el modelo de datos que corresponde, así como la descripción de la(s) tabla(s) de este modelo.



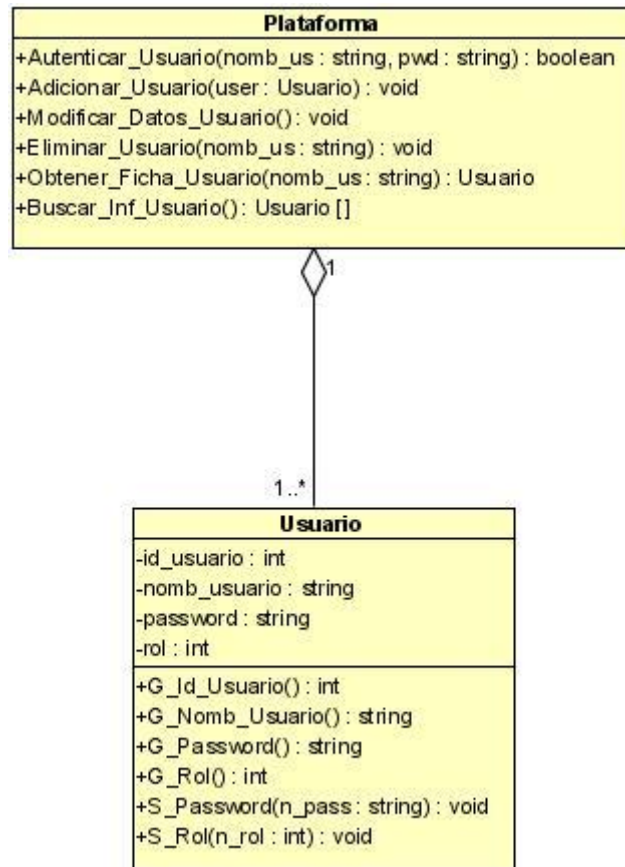


Fig. 30 Diagrama de Clases Persistentes

Usuario		
<b>+id_usuario</b>	<b>integer(10)</b>	<b>Nullable = false</b>
nomb_usuario	varchar(255)	Nullable = false
password	varchar(255)	Nullable = false
rol	integer(1)	Nullable = false

Fig. 31 Modelo de Datos

### 3.2.5 Descripción de las Tablas de la Base de Datos.

Nombre.	Usuario
---------	---------

Descripción.	Esta clase contiene todos los datos de los usuarios.	
Atributo	Tipo	Descripción
id_usuario	integer	Identificador del usuario.
nomb_usuario	varchar	Nombre de usuario.
password	varchar	Contraseña de acceso al sistema.
rol	integer	Rol que juega el usuario en el sistema.

**CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD.**

Una de las tareas de mayor importancia en la planificación de proyectos es la estimación, la cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para llevar a cabo el proyecto. En el presente capítulo se realiza un estudio de factibilidad del sistema que se propone, haciendo una estimación del esfuerzo necesario y el costo desarrollar el mismo.

**4.1 Características del proyecto.**

## 4.1.1 Entradas externas.

Nombre de la entrada externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Autenticar usuario	1	2	Simple
Adicionar usuario	1	-	Media
Modificar datos de usuario	1	-	Media
Eliminar usuario	1	1	Simple

## 4.1.2 Consultas externas.

Nombre de la consulta externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Buscar información de usuarios	1	-	Media
Obtener ficha de usuario	1	-	Media

4.1.3 Archivos lógicos internos.

Nombre del archivo lógico interno	Tipos de registro	Elementos de datos	Clasificación (simple, media, compleja)
Usuario	1	1	Simple

4.1.4 Archivos de interfaz externos.

Nombre del archivo de interfaz externo	Tipos de registro	Elementos de datos	Clasificación (simple, media, compleja)
Base de datos externa	-	-	Media

4.1.5 Puntos de función sin ajustar.

	Complejidad						Aporte
	Simple		Media		Compleja		
	cantidad	peso	cantidad	peso	cantidad	peso	
Entradas externas	4	3					12
Consultas externas			2	4			8
Archivos lógicos internos	1	7					7
Archivos de interfaz externos			1	7			7
	Total						34

4.2 Cálculo de esfuerzo e instrucciones fuentes.

El modelo COCOMO II utiliza Puntos de Función y/o Líneas de Código Fuente (SLOC) como base para medir el tamaño de un proyecto en los modelos de estimación de Diseño Preliminar y Post-Arquitectura y considera solamente los Puntos de Función Sin Ajustar (UFP). Los Puntos de Función procuran cuantificar la funcionalidad de un sistema de software y su objetivo es obtener un número que caracterice completamente al sistema, estos son buenos estimadores, ya que están basados en información que está disponible en las etapas tempranas del ciclo de vida del desarrollo del software.

Para realizar el cálculo de esfuerzo en un proyecto se parte de la siguiente fórmula:

$$PM_{nominal} = A * (Size)^B$$

En la misma se tiene que:

**PM<sub>nominal</sub>**: es el esfuerzo nominal requerido en meses/hombre.

**Size**: es el tamaño estimado del software en miles de líneas de código (KSLOC) o en Puntos de Función sin Ajustar (convertibles a KSLOC mediante un factor de conversión que depende del lenguaje y la tecnología).

**A**: es una constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del software. El modelo la calibra inicialmente con un valor de **2,94**.

**B**: es una constante denominada Factor Escalar, la cual tiene un impacto exponencial en el esfuerzo y su valor está dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto. Para el cálculo de esta constante se tiene que: **B = 0,91 + 0,01 \* Σ(W<sub>i</sub>)**.

**W<sub>i</sub>**: son una serie de Factores Escalares a los cuales se les asigna un valor en dependencia de su ponderación. A continuación se muestra una tabla con los Factores Escalares, el valor que se le asigna y su justificación.

Factor Escalar	Valor	Justificación
PREC	3,72	Existen algunas aplicaciones en el mundo que tienen algo en común con la que se propone, pero no hay ninguna que realice su función completamente.
FLEX	3,04	La flexibilidad durante el desarrollo del software no es excesiva.

RESL	1,41	La arquitectura de desarrollo es bastante fuerte por lo cual no se identifican riesgos.
TEAM	1,10	El equipo de trabajo presenta una alta cooperación.
PMAT	3,12	Existe una experiencia previa en el desarrollo de Servicios Web.

Luego de esto se tiene que  $\Sigma(W_i) = 12,39$ , y por lo tanto  $B = 1,0339$ . El valor de **Size** se calcula como el producto de los Puntos de Función sin Ajustar (UFP) por un factor de conversión que depende del Lenguaje de Programación a utilizar en el desarrollo del sistema, que en este caso es PHP y su factor de conversión es 53 SLOC/UFP, por lo cual se tiene que **Size = 34 \* 53 = 1802 SLOC = 1,802 KSLOC**.

Llegados a este punto, ya se puede determinar que:

$$PM_{nominal} = 2,94 * (1,802)^{1,0339} = 5,4047 \text{ meses/hombre}$$

Para completar la estimación hay que ajustar el Esfuerzo Nominal de acuerdo a las características del proyecto. Dicho ajuste se realiza de la siguiente forma:

$$PM_{ajustado} = PM_{nominal} * \Pi(ME_i)$$

Donde:

$PM_{ajustado}$  es el esfuerzo ajustado dado en meses/hombre y  $ME_i$  son Multiplicadores de Esfuerzo que varían en dependencia del modelo de estimación que se seleccione, en el sistema que se propone se aplica el modelo Post-Arquitectura, y a continuación se muestra una tabla con los multiplicadores de esfuerzo, su valor y la justificación.

Multiplicador de Esfuerzo	Valor	Justificación
RELY	1,10	Una falla en el sistema ocasionaría pérdidas con un alcance moderado, a pesar d que los datos se recuperarían de manera fácil.

DATA	0,90	La base de datos que se maneja no es de gran tamaño.
DOCU	1,00	La documentación es acorde a las necesidades del proyecto.
CPLX	1,00	El software presenta una complejidad media.
RUSE	1,15	La reusabilidad del código es muy alta.
TIME	1,00	El sistema necesitará una disponibilidad de recursos normal.
STOR	1,00	El sistema necesitará poco espacio en el disco duro para funcionar correctamente.
PVOL	1,00	La velocidad de cambio de hardware se realiza de manera normal.
ACAP	0,85	La capacidad de los analistas es alta.
PCAP	0,88	La capacidad de los programadores es alta.
AEXP	1,22	No hay experiencia en el trabajo con aplicaciones similares.
PEXP	1,00	Los programadores poseen más de un año de experiencia con la plataforma de desarrollo a utilizar.
LTEX	0,91	Se tiene una gran experiencia con el lenguaje de programación y las herramientas de desarrollo.
PCON	1,00	El porcentaje de rotación del personal es normal.
TOOL	0,90	Se usan herramientas potentes durante todo el ciclo de vida del software.
SITE	0,99	El soporte de comunicación cubre todas las ubicaciones.
SCED	1,00	Los requerimientos del calendario de desarrollo son normales.

Una vez determinados los valores de todos los Multiplicadores de Esfuerzo se obtiene que

$$\Pi(ME_i) = 0,8424$$

Y de esta manera se tiene finalmente que  **$PM_{ajustado} = 5,4047 \text{ meses/hombre} * 0,8424 = 4,5529 \text{ meses/hombre}$** .

Si se expresa este mismo valor en horas/hombre, y teniendo en cuenta que un mes tiene aproximadamente 160 horas, esfuerzo resulta:

**$PM_{ajustado} = 4,5529 * 160 = 728,464 \text{ horas/hombre}$** .

Una vez realizada la estimación del proyecto se puede concluir que es factible acometer su desarrollo puesto que el valor obtenido del esfuerzo es aceptable. Además, el costo de este sistema es relativamente bajo por el uso de tecnologías y herramientas libres y los beneficios que puede aportar a la institución que lo utilice son inmensos, puesto que se organiza de gran manera el trabajo con los datos de los usuarios.



**CONCLUSIONES.**

Con la realización de este trabajo quedó demostrada la necesidad de contar con una Plataforma para la Gestión de la Información de Usuarios, un sistema que es capaz de funcionar en cualquier tipo de institución o empresa y que elimina la redundancia de información existente en muchos de estos organismos, organizando y manejando de manera precisa y eficiente toda la información referente a los usuarios de las instituciones.

Para realizar la propuesta se realizó un profundo análisis de sistemas similares que existen en el resto del mundo y en nuestro país. Se utilizó RUP como metodología de desarrollo y para el modelado el lenguaje UML, todo esto se aplicó utilizando Visual Paradigm como herramienta CASE.

Se realizó la propuesta de utilizar para el futuro de desarrollo del sistema propuesto la tecnología de Servicios Web, puesto que es la mejor solución para clientes que no están realmente seguros de lo que quieren, como lenguaje de programación se propuso el uso de PHP, como sistema gestor de bases de datos MySQL, y además se propuso el uso de las librerías NuSOAP y ADOdb.

Se realizó la modelación del negocio y a partir de esto y del levantamiento de requisitos funcionales y no funcionales se realizó el modelo del sistema propuesto y la modelación y descripción de los Casos de Uso necesarios para llevar a cabo la realización de la Plataforma. Luego de esto se realizó el análisis del sistema, en el cual se expusieron los diagramas de clases y de colaboración, para a partir de esto realizar el diseño del sistema, en el cual se expusieron los diagramas de clases de diseño con sus respectivos diagramas de interacción, el diagrama de clases persistentes del sistema y el modelo de datos.

Se llevó a cabo la estimación del esfuerzo necesario para llevar a cabo el desarrollo del sistema y esta arrojó como resultado que es factible acometer la implementación de la Plataforma para la Gestión de la Información de Usuarios.

Una vez realizado todo esto se puede concluir que la Plataforma para la Gestión de la Información de Usuarios que se propuso con este trabajo es un sistema sumamente útil para todo tipo de instituciones y su implementación es necesaria para lograr la organización que se requiere en el manejo de los datos de los usuarios.

### **RECOMENDACIONES.**

Sobre la presente investigación se recomienda:

- Implementar el Servicio Web que se propone en este trabajo.
- Continuar investigando sobre las tendencias más utilizadas en el mundo para el desarrollo de este tipo de software.

**REFERENCIAS BIBLIOGRÁFICAS.**

1. Axxon Consulting realiza nueva alianza con Payroll RH. Disponible en: [http://www.axxonconsulting.com/newsletter/Vol23\\_0806.htm](http://www.axxonconsulting.com/newsletter/Vol23_0806.htm)
2. Cams Overview. Disponible en: <http://www.cafesoft.com/products/cams/camsOverview.html>
3. Gestión de usuarios. Disponible en: <http://www.ibermatica.com/ibermatica/sit/situsuarios>
4. GREHU, Sistema de gestion de recursos humanos. Disponible en: <http://grehu.cujae.edu.cu/General/Caracteristicas.asp>
5. Introduction to Single Sign-On. Disponible en: [http://www.opengroup.org/security/sso/sso\\_intro.htm](http://www.opengroup.org/security/sso/sso_intro.htm)
6. New SQL/XML functions provide a standards-based bridge between relational data and XML., 2003. [Disponible en: [http://www.oracle.com/technology/oramag/oracle/03-may/o33\\_xml.html?\\_template=/ocom/print](http://www.oracle.com/technology/oramag/oracle/03-may/o33_xml.html?_template=/ocom/print)
7. PayRoll Recursos Humanos. Disponible en: <http://www.payroll.com.ar/>
8. PayRoll RH. Disponible en: <http://www.pct.org.ar/3-AcercaParque/35-Clusters/1-pinfo/PAYROLL.html>
9. Por qué XML. Disponible en: [http://www.gamarod.com.ar/articulos/por\\_que\\_xml.asp](http://www.gamarod.com.ar/articulos/por_que_xml.asp)
10. ¿Qué es SOAP? Disponible en: <http://www.desarrolloweb.com/articulos/1557.php>
11. SAML Single Sign-On (SSO) Service for Google Apps. Disponible en: [http://code.google.com/apis/apps/sso/saml\\_reference\\_implementation.html](http://code.google.com/apis/apps/sso/saml_reference_implementation.html)
12. Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) 6.0., 2007. [Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)
13. Web single sign-on (SSO) for Apache, Microsoft IIS, Sun ONE, BEA WebLogic, IBM WebSphere, JBoss, Oracle 9iAS/10g, Tomcat and J2EE web and application servers. Disponible en: <http://www.cafesoft.com/?referer=AdWords-WebSSO&gclid=CPn6jbigiJMCFQv4gAodr2AqTw>

14. ESPERANZA MARCOS M., M. V. D. C., BELÉN VELA S. Una extensión de UML para el modelado de Servicios Web., 2004. [Disponible en: <http://redalyc.uaemex.mx/redalyc/pdf/730/73000207.pdf>]
15. GROUP., T. O. X/Open Single Sign-on Service (XSSO) — Pluggable Authentication Modules.
16. RAJEEV ANGAL, S. M. C. K., MICROSOFT; HUBERT LE VAN GONG, SUN MICROSYSTEMS; EVE MALER, SUN MICROSYSTEMS; ARI MEDVINSKY, MICROSOFT; JOHN SHEWCHUK, MICROSOFT. Web Single Sign-On Interoperability Profile, 2005.

### GLOSARIO DE TÉRMINOS.

- **ADOdb:** Paquete de librería de abstracción de base de datos de una manera portable.
- **CASE:** Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador), aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.
- **HTML:** Lenguaje de Marcación de Hipertexto. Es un lenguaje de marcas diseñado para estructurar textos y representarlos en forma de hipertexto, que es el formato estándar de las páginas web.
- **MySQL:** Servidor de bases de datos recomendable para desarrollos que necesiten manejar numerosos registros y sesiones simultáneas.
- **NuSOAP:** Es un grupo de las clases escritas en PHP que permiten a los desarrolladores crear y consumir servicios web a través del protocolo SOAP.
- **PHP:** Lenguaje de programación interpretado.
- **Plataforma:** En informática, una plataforma es precisamente el principio, ya sea de hardware o software, sobre el cual un programa puede ejecutarse. Ejemplos típicos incluyen: arquitectura de hardware, sistema operativo, lenguajes de programación y sus librerías de tiempo de ejecución.
- **RUP:** El Proceso Unificado de Racional es un proceso de desarrollo de software.
- **SAML:** SAML es un derivado de XML que está diseñado para el intercambio de datos de autenticación y autorización. El propósito de SAML es permitir a Single Sign-On para aplicaciones Web. SAML utiliza TLS para garantizar la confidencialidad de los datos de autenticación y autorización durante el tránsito.
- **SOAP:** Protocolo Simple de Acceso a Objeto. Protocolo para el intercambio de mensajes XML entre redes, comúnmente utilizado sobre el protocolo HTTP.

- **UML:** Lenguaje Unificado de Modelado, lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.
- **Usuario:** Un usuario es la persona que utiliza o trabaja con algún objeto o que es destinataria de algún servicio público o privado, empresarial o profesional. En el contexto de la informática, un usuario es aquel que utiliza un sistema de informático. En el caso de la Plataforma que se propone, usuario es toda aquella persona que interactúe directamente con el sistema.
- **W3C:** World Wide Web Consortium. Desarrolla tecnologías íter-operables (especificaciones, guías, software y herramientas) para maximizar el potencial de la web.
- **XML:** Extensible Markup Language (lenguaje de marcas extensible) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).