

# Universidad de las Ciencias Informáticas

## Facultad 1



### **Análisis y Diseño del**

### **Sistema de Gestión de Fotos en la UCI**

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

**Autora:** Yanara Ramil Hernández

**Tutor:** Ing. Julio Cesar Herrera Soria

Ciudad de La Habana, junio de 2008



*"Seamos realistas y hagamos lo imposible"*

*Ernesto Che Guevara*

## **DECLARACIÓN DE AUTORÍA**

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de junio del año 2008

---

Yanara Ramil Hernández

---

Ing. Julio César Herrera Soria

## **DATOS DE CONTACTO**

Graduado de ingeniería en informática en el curso 2006 - 2007 en la Universidad de Holguín, se encuentra en estos momentos en proceso de adiestramiento.

## **OPINIÓN DEL TUTOR**

Como tutor del presente Trabajo de Diploma “Análisis y diseño del sistema de gestión de fotos en la UCI” considero que durante la ejecución del mismo la estudiante mostró un conjunto de cualidades valiosas en su futuro desempeño como profesional en el campo de la Ingeniería Informática. Considero que las cualidades que a continuación se mencionan alcanzaron las siguientes calificaciones: la independencia: media, originalidad: alta, creatividad: alta, laboriosidad: muy alta y la responsabilidad: muy alta. Demostró además habilidades investigativas, capacidades de comunicación con los clientes, habilidades para el trabajo en equipo y ética profesional.

El Trabajo de Diploma presenta muy alta calidad en el aspecto científico técnico, utilizando correctamente la metodología de la investigación, consultas bibliográficas actualizadas, correcto desempeño del proceso de desarrollo de software según RUP y utilización de las técnicas y herramientas actuales para el desempeño de su rol como Analista y diseñadora. Además el trabajo constituye una documentación muy valiosa para los miembros del equipo a la hora de implementar el módulo Gestión de Fotos.

Los resultados obtenidos tendrán un elevado valor para toda la Universidad, el Departamento de Seguridad y Protección y para todo el Polo Científico.

Por todo lo anteriormente expresado considero que la estudiante está apta para ejercer como Ingeniera en Ciencias Informática; y propongo que se le otorgue al Trabajo de Diploma la máxima calificación: 5 puntos. Además considero que los resultados poseen valor para ser publicados.

## AGRADECIMIENTOS

*A mis padres y a mi padrastro “Elena, Chuly, Cruz” que sin su apoyo, su valor, y su amor incondicional nada de esto hubiese sido posible.*

*A la bondad y delicadeza de mis abuelas y al resto de mi familia por su apoyo.*

*A mis camaroncitos duros que me sacan del apuro “Jose A. Cobo y Henry Yordan”.*

*A Yadira y al Juank, “Nadie hizo tanto por mi en tan poco tiempo”.*

*A Yisel y a Julito por la comida de los domingos y por la ayuda brindada.*

*A mis maestros en todo el concepto de la palabra “Yoisel y Herminio”.*

*A esos amigos súper especiales que no los menciono porque son mucho pero ellos saben quienes son pues me han demostrado que “No importa lo que tengas si no a quien tengas a tu lado”.*

*A Marilenys, a Yuly, a Lili, al Adrian, a Cesar, a Ailema, a Yailén y a Julito quienes me han demostrado que el tamaño de la boca, de los ojos, de la bobería o de la frente es proporcional al tamaño del corazón “Tienen un gran corazón”.*

*A Mislady por enseñarme que no importa lo que se coma, si no la sobremesa que se haga.*

*A mis vecinos por enseñarme que es mejor esperar el 1ro si se tiene Sidra.*

*A la familia de Camagüey por su buena “agua de tinajón”.*

*A Yisel, Yadira, Sandra, Daylin, Maidel, Liz, Elsia, y a Lisbet por recogerme cuando no tenía casa.*

*A mi tutor por su ayuda.*

*A la Revolución y a Jesusito.*

*A los santos y a los brujos.*

*En fin a todos los que hicieron de esto un sueño hecho realidad.*

*A todos Gracias!!!*

## DEDICATORIA

*Dedico este trabajo por formar parte esencial de mi vida a:*

*Mi papá, mi padrastro y en especial a mi madre querida.*

*Mis abuelos presentes en especial a Mamá Eloisa y a la memoria del que ya no está físicamente entre nosotros "Papá Chino".*

*Mi familia sanguínea y no consanguínea "tías, tíos, primas, primos y madrastra".*

*La flaqueca de mi hermana "La gordi" y al escuincle de mi hermano "El cagua".*

*Cobo por su amor, ayudarme tanto y ser un ejemplo de paciencia sin límites.*

*Mi vecinito y a Lucecita por su bendición.*

*Mis amigos:*

- *los que crecieron conmigo.*
- *los que rieron conmigo.*
- *los que anduvieron y convivieron conmigo en el pre, en la universidad.*
- *Esos que echaron su suerte junto a mí.*
- *Aquellos que de una forma u otra me aguantaron tanto tiempo.*

*En especial a Marilenys, Yoisel y a mi hermano Yordan.*

*Como Uds. Pocos y como yo ninguna "Por Tenerlos".*

## **RESUMEN**

La incorrecta manipulación que se ha estado llevando a cabo desde el surgimiento de la UCI, de las fotos que identifican a las personas que interactúan en ella, propició que se decidiera por parte de la dirección de Informatización de la Universidad, centralizar todo el proceso en un sistema multiplataforma que cumpliera con las políticas de seguridad y de utilización de software libre de la Universidad. Como consecuencia, surge el módulo de gestión de fotos.

En ese momento se decidió que los principales procesos a informatizar por este módulo iban a ser la gestión de las fotos principales brindadas a todos sistemas de la Universidad, y la gestión de las fotos de seguridad y protección que son de interés para los directivos encargados de la protección de la UCI.

En el presente trabajo se realiza el análisis y diseño del Sistema de Gestión de Fotos en la UCI, a partir de los requisitos definidos con el cliente, y de la arquitectura definida por parte de los arquitectos del proyecto Control de Acceso.

La realización de todos los artefactos pertenecientes a las distintas etapas trabajadas, dará culminación a toda la documentación del módulo, dejándolo listo para una posterior implementación del mismo.

### **Palabras claves**

Imágenes digitales, Metadatos, Negocio, Requerimientos, Análisis, Diseño.

## TABLA DE CONTENIDOS

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1 INTRODUCCIÓN .....	5
1.2 IMÁGENES DIGITALES.....	5
1.2.1 <i>Características de una imagen digital</i> .....	5
1.2.2 <i>Formatos de almacenamiento de imágenes digitales</i> .....	5
1.2.3 <i>¿Qué significa compresión con pérdidas?</i> .....	7
1.2.5 <i>Resolución en imágenes digitales</i> .....	7
1.2.6 <i>Presencia de metadatos en las imágenes</i> .....	8
1.2.7 <i>Metadatos y JPEG como formato de fichero de imagen digital</i> .....	8
1.3 SISTEMAS INFORMÁTICOS Y GESTIÓN DE IMÁGENES DIGITALES PARA LA IDENTIFICACIÓN DE PERSONAS.....	8
1.3.1 <i>Sistemas informáticos de gestión de fotos a nivel internacional</i> .....	9
1.3.2 <i>Sistemas informáticos de gestión de fotos en Cuba</i> .....	11
1.4 METODOLOGÍA, LENGUAJES, TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL DESARROLLO .....	12
1.4.1 <i>Metodología de desarrollo de software</i> .....	13
1.4.2 <i>Lenguaje de modelado</i> .....	15
1.4.3 <i>Herramienta CASE</i> .....	15
1.4.4 <i>Aplicaciones web</i> .....	16
1.4.5 <i>Servicios web</i> .....	16
1.4.6 <i>Arquitectura orientada a servicios</i> .....	19
1.4.7 <i>Lenguajes de programación web</i> .....	20
1.4.8 <i>Framework</i> .....	21
1.4.9 <i>Plugin</i> .....	22
1.5 CONCLUSIONES .....	22
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>23</b>
2.1 INTRODUCCIÓN .....	23
2.2 FLUJO ACTUAL DE LOS PROCESOS.....	23
2.3 ANÁLISIS CRÍTICO DE LA EJECUCIÓN DE LOS PROCESOS.....	23
2.4 PROPUESTA DE SISTEMA .....	24
2.5 MODELO DEL DOMINIO .....	25
2.5.1 <i>Diagrama de clases del modelo de dominio</i> .....	26
2.6 ESPECIFICACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA.....	27
2.6.1 <i>Requerimientos funcionales</i> .....	27
2.6.2 <i>Requerimientos no funcionales</i> .....	28
2.7 DESCRIPCIÓN DEL SISTEMA PROPUESTO.....	30
2.7.1 <i>Descripción de los actores</i> .....	30
2.7.2 <i>Diagrama de casos de usos del sistema</i> .....	31
2.7.3 <i>Descripción de los casos de usos del sistema</i> .....	31
2.8 ESTUDIO DE LA FACTIBILIDAD .....	52
2.9 CONCLUSIONES .....	57
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA</b> .....	<b>59</b>

3.1 INTRODUCCIÓN .....	59
3.2 MODELO DE ANÁLISIS .....	59
3.2.1 DIAGRAMAS DE CLASES DEL ANÁLISIS .....	59
3.3 MODELO DE DISEÑO .....	63
3.4 PATRONES DE DISEÑO .....	64
3.5 PATRONES DE ARQUITECTURA .....	65
3.6 DIAGRAMAS DE CLASES WEB .....	66
3.7 DIAGRAMAS DE INTERACCIÓN .....	67
3.7.1 Diagramas de secuencia .....	67
3.8 DESCRIPCIÓN DE LAS CLASES .....	68
3.9 CONCLUSIONES .....	69
<b>CONCLUSIONES.....</b>	<b>70</b>
<b>RECOMENDACIONES.....</b>	<b>71</b>
<b>BIBLIOGRAFÍA.....</b>	<b>72</b>
<b>ANEXOS.....</b>	<b>75</b>
ANEXO 1: DIAGRAMA DEL DISEÑO DEL CU BUSCAR FOTO PRINCIPAL.....	75
ANEXO 2: DIAGRAMA DEL DISEÑO DEL CU GUARDAR FOTO PRINCIPAL.....	76
ANEXO 3: DIAGRAMA DEL DISEÑO DEL CU GESTIONAR FOTO SEGURIDAD.....	77
ANEXO 4: DIAGRAMA DEL DISEÑO DEL CU ACTUALIZAR CAMINO A IMAGEN .....	78
ANEXO 5: DIAGRAMA DEL DISEÑO DEL CU AUTENTICAR USUARIO. ....	79
ANEXO 6: DIAGRAMA DEL DISEÑO DEL CU GESTIONAR USUARIOS. ....	80
ANEXO 7: DIAGRAMA DEL DISEÑO DEL CU LEER METADATO. ....	81
ANEXO 8: DIAGRAMA DEL DISEÑO DEL CU GESTIONAR METADATO.....	81
ANEXO 9: DIAGRAMA DEL DISEÑO DEL CU ESTANDARIZAR FOTO. ....	82
ANEXO 10: DIAGRAMA DE SECUENCIA DEL CU BUSCAR FOTO PRINCIPAL. ....	82
ANEXO 11: DIAGRAMA DE SECUENCIA DEL CU GUARDAR FOTO PRINCIPAL.....	83
ANEXO 12: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR FOTO SEGURIDAD ESCENARIO GUARDAR FOTO ESTUDIANTE. ....	84
ANEXO 13: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR FOTO SEGURIDAD ESCENARIO GUARDAR FOTO TRABAJADORES.....	85
ANEXO 14: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR FOTO SEGURIDAD ESCENARIO BUSCAR FOTO.....	86
ANEXO 15: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR FOTO SEGURIDAD ESCENARIO ELIMINAR FOTO. ....	87
ANEXO 16: DIAGRAMA DE SECUENCIA DEL CU ACTUALIZAR CAMINO A IMAGEN.....	88
ANEXO 17: DIAGRAMA DE SECUENCIA DEL CU AUTENTICAR USUARIO.....	89
ANEXO 18: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR USUARIO ESCENARIO INSERTAR NUEVO USUARIO.....	90
ANEXO 19: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR USUARIO ESCENARIO ELIMINAR USUARIO.....	91
ANEXO 20: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR USUARIO ESCENARIO INSERTAR NUEVO ROL.....	92
ANEXO 21: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR USUARIO ESCENARIO ELIMINAR ROL.....	93
ANEXO 22: DIAGRAMA DE SECUENCIA DEL CU LEER METADATO.....	94
ANEXO 23: DIAGRAMA DE SECUENCIA DEL CU GESTIONAR METADATOS.....	95
ANEXO 24: DIAGRAMA DE SECUENCIA DEL CU ESTANDARIZAR FOTO.....	95
<b>GLOSARIO.....</b>	<b>96</b>

# ÍNDICE DE TABLAS

## Capítulo 1

TABLA 1. 1: PRINCIPALES CARACTERÍSTICAS DE LOS FORMATOS DE IMÁGENES DIGITALES.....	7
------------------------------------------------------------------------------------	---

## Capítulo 2

TABLA 2. 1: DEFINICIÓN DE LAS ENTIDADES Y CONCEPTOS PRINCIPALES. ....	26
TABLA 2. 2: DEFINICIÓN DE LOS ACTORES DEL SISTEMA. ....	30
TABLA 2. 3: DESCRIPCIÓN DEL CASO DE USO ESTANDARIZAR IMAGEN. ....	32
TABLA 2. 4: DESCRIPCIÓN DEL CASO DE USO GESTIONAR METADATOS. ....	34
TABLA 2. 5: DESCRIPCIÓN DEL CASO DE USO BUSCAR FOTO PRINCIPAL.....	34
TABLA 2. 6: DESCRIPCIÓN DEL CASO DE USO GUARDAR FOTO PRINCIPAL.....	37
TABLA 2. 7: DESCRIPCIÓN DEL CASO DE USO GESTIONAR FOTO SEGURIDAD.....	43
TABLA 2. 8: DESCRIPCIÓN DEL CASO DE USO LEER METADATOS. ....	46
TABLA 2. 9: DESCRIPCIÓN DEL CASO DE USO ACTUALIZAR CAMINO A IMAGEN. ....	47
TABLA 2. 10: DESCRIPCIÓN DEL CASO DE USO AUTENTICAR USUARIO. ....	48
TABLA 2. 11: DESCRIPCIÓN DEL CASO DE USO GESTIONAR USUARIO.....	52
TABLA 2. 12: FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR. ....	53
TABLA 2. 13: FACTOR DE PESO DE LOS CASOS DE USO SIN AJUSTAR. ....	54
TABLA 2. 14: FACTORES DE PESO DE COMPLEJIDAD TÉCNICA.....	55
TABLA 2. 15: FACTORES DE PESO DEL FACTOR DE AMBIENTE.....	56

## Capítulo 3

TABLA 3. 1: DESCRIPCIÓN DE LAS CLASES UTILIZADAS EN EL DISEÑO.....	69
--------------------------------------------------------------------	----

## ÍNDICE DE FIGURAS

FIGURA 1: SISTEMA PHOTO-RDF.....	9
FIGURA 2: VISTA DE LA INTERFAZ DEL PROGRAMA DE INTRODUCCIÓN DE DATOS .....	10
FIGURA 3: FASES E ITERACIONES DE LA METODOLOGÍA RUP .....	14
FIGURA 4: SISTEMA BASADO EN SERVICIOS. ....	17
FIGURA 5: VISIÓN INTERNA DE LOS SERVICIOS. ....	17
FIGURA 6: ARQUITECTURA SOA TRADICIONAL.....	19
FIGURA 7: MODELO DE DOMINIO. ....	26
FIGURA 8: DIAGRAMA DE CASOS DE USOS DEL SISTEMA .....	31
FIGURA 9: DIAGRAMA DE ANÁLISIS DEL CU BUSCAR FOTO PRINCIPAL.....	60
FIGURA 10: DIAGRAMA DE ANÁLISIS DEL CU GUARDAR FOTO PRINCIPAL.....	60
FIGURA 11: DIAGRAMA DE ANÁLISIS DEL CU GESTIONAR METADATOS.....	60
FIGURA 12: DIAGRAMA DE ANÁLISIS DEL CU ESTANDARIZAR FOTO. ....	61
FIGURA 13: DIAGRAMA DE ANÁLISIS DEL CU GESTIONAR FOTO SEGURIDAD. ....	61
FIGURA 14: DIAGRAMA DE ANÁLISIS DEL CU LEER METADATO.....	62
FIGURA 15: DIAGRAMA DE ANÁLISIS DEL CU AUTENTICAR USUARIO. ....	62
FIGURA 16: DIAGRAMA DE ANÁLISIS DEL CU GESTIONAR USUARIO.....	62
FIGURA 17: ÍCONO QUE REPRESENTA LAS PÁGINAS SERVIDORAS.....	66
FIGURA 18: ÍCONO QUE REPRESENTA LAS PÁGINAS CLIENTES.....	67
FIGURA 19: ÍCONO QUE REPRESENTA LOS FORMULARIOS.....	67

### INTRODUCCIÓN

Desde el surgimiento de la Universidad de las Ciencias Informáticas (UCI) entre los primeros retos a enfrentar estaba gestionar la información y la identificación visual de todas las personas (estudiantes y trabajadores) que interactúan en ella. Esta gestión tenía que estar al nivel de una institución donde se fusionan las nuevas tecnologías con los métodos normales de gestión para las instituciones de su tipo.

Para facilitar el proceso de gestión y búsqueda de información de la identidad de estas personas, se tomó la decisión de construir sistemas basados en el conocimiento y las tecnologías, con las se contaba en ese momento y que permitiera vincular el proceso de tomar las fotos con la información de inscripción que se realiza en la Universidad.

De esta forma surgió el sistema Acreditación de Personas que básicamente contaba con dos aplicaciones de escritorio, encargadas de gestionar toda la información asociada a las personas que eran de interés para la UCI.

Es por ello que dentro de los requerimientos de una de estas aplicaciones pertenecientes al sistema Acreditación de Personas se incluye la gestión de las fotos de las personas, ya que permite procesarlas y asociarle a la información de las personas, que pasan a retratarse, en la base de datos la dirección de la foto y almacenar la misma en un directorio.

Por otro lado vale destacar que este proceso es de vital importancia para el correcto funcionamiento de otras aplicaciones que han sido implementadas posteriormente ejemplo de ellas son el Directorio de Personas (<http://directorio.uci.cu>), Control de Acceso a Comedores, Control de Acceso y otras que necesitan acceder a estas fotos para cumplir con sus objetivos.

La aplicación perteneciente al sistema Acreditación de Personas que se encarga de gestionar las imágenes fue el resultado de una solución rápida a un problema existente por lo que no presenta la documentación necesaria para su correcto funcionamiento, ni para el mantenimiento del software ya que no fue especificado el proceso de ingeniería de software realizado. La implementación es funcional solamente para el sistema operativo Windows lo que no se ajusta a la política de la Universidad en cuanto al uso de software privativo. Concebido como una aplicación de escritorio, en la actualidad se requiere que todo el acceso a las fotos sea mediante la web o utilizando servicios web y no por recursos compartidos como hasta ahora. Las fotos gestionadas por el sistema no incluyen información de a quién o a qué pertenecen, así en caso de que ocurra una contingencia que altere el orden de los datos se perdería la confiabilidad en el sistema. No permite un historial de fotos, de las personas a las cuales pertenecen siendo esto en términos de seguridad para la Universidad un riesgo potencial. Los

restantes sistemas mencionados anteriormente, que también necesitan de estas fotos para los servicios que prestan acceden de manera local al directorio de imágenes por lo que se torna un proceso inseguro y poco confiable.

Por otro lado los directivos de seguridad y protección encargados de la seguridad del centro no cuentan con un sistema encargado de gestionar las fotos de identidad que necesitan para tener un mejor desenvolvimiento en las funciones que realizan, tal es el caso de las fotos de incidencia y de perfiles por lo que dependen totalmente de las limitaciones que suponen las funcionalidades brindadas por los sistemas anteriormente mencionados.

Adicionalmente, en la actualidad existen diversos sistemas de gestión de fotos, algunos propietarios y otros de software libre, pero ninguno se ajusta a las necesidades del proceso que se realiza en la UCI.

Por todo lo antes expuesto y a partir de un estudio de estos procesos en la Universidad, se decidió crear por parte del Grupo de informatización de la misma en el proyecto Control de Acceso el Módulo de Gestión de Fotos el cuál será el encargado de realizar un sistema de gestión de fotos para la UCI que satisfaga todas las necesidades propias de su entorno, que centralice los procesos, que sea multiplataforma y que además gestione las fotos de las personas que son del interés del personal de seguridad y protección de la Universidad.

De acuerdo a la situación planteada anteriormente, se establece el problema que da origen y fundamenta la realización del trabajo: ¿Cómo proveer a toda una infraestructura de desarrolladores de software y a los directivos de seguridad y protección, de una herramienta que permita la gestión de las fotos de identificación de personas en la UCI?

Dicho problema se enmarca dentro del objeto de estudio: los procesos de gestión de las fotos de identificación de personas y el campo de acción, los procesos de gestión de las fotos de identificación de personas en la UCI.

El objetivo general del trabajo es realizar la modelación de un sistema multiplataforma de gestión de fotos que genere los artefactos correspondientes para una posterior implementación del mismo y los objetivos específicos son:

- Definir los procesos de gestión de fotos en la Universidad.
- Diseñar los procesos de gestión de fotos en la Universidad.
- Elaborar la propuesta del proceso de gestión de fotos en la Universidad.

La pregunta científica sería: ¿Qué se necesita conocer para realizar el Análisis y Diseño del sistema de Gestión de Fotos?

Se necesita conocer:

- ¿Qué fotos se gestionan, cómo se gestionan y quién las gestiona?
- ¿Cómo trabajar con metadatos?
- ¿Cómo trabajar con servicios Web?
- ¿Cuál es la metodología más acertada para realizar el modelado?
- ¿Es adecuada la herramienta en la que voy a trabajar?

Las tareas científicas a realizar por el autor del trabajo para dar cumplimiento a los objetivos planteados serían:

- Estudio de la bibliografía para elaborar el marco teórico referencial.
- Estudio de la metodología de desarrollo de software a utilizar para la generación de la documentación y el desarrollo del diseño del sistema.
- Estudio de las tecnologías y herramientas a utilizar en el diseño del sistema.
- Elaboración de la fundamentación teórica de todo el proceso.
- Realización de una estimación del esfuerzo requerido para determinar cantidad de personas y tiempo para la culminación del análisis y diseño.
- Descripción de la solución propuesta.

Los métodos de la investigación científica utilizados en este proceso de investigación para darle cumplimiento a las tareas científicas son los siguientes:

- El método de la observación ha provisto a los autores de una vía más, para la verificación del cumplimiento de las características que debe poseer el sistema y garantizar el cumplimiento de los objetivos trazados.
- Entrevista a los agentes de seguridad y protección de la UCI y compañeros del Ministerio del Interior que pueden aportar elementos significativos a la presente investigación sobre su resultado final y su uso.
- Análisis de documentos con el objetivo de determinar los puntos esenciales del proceso.
- Método histórico para determinar si actualmente están desarrollados sistemas informáticos de gestión de fotos.

- Método de la modelación para crear modelos a través de la metodología RUP.

Los resultados esperados de este trabajo son:

- Modelado del negocio.
- Requisitos Funcionales y No Funcionales.
- Modelado del Sistema.
- Modelo de análisis.
- Modelo de diseño.

El presente trabajo consta de introducción, tres capítulos, conclusiones, recomendaciones, bibliografías y anexos.

En el **Capítulo 1** se explican las principales características de las imágenes digitales y de los formatos de almacenamiento, se realiza un estudio del estado del arte de diferentes sistemas de gestión de fotos a nivel internacional y en la UCI. Se hace una descripción de la metodología de desarrollo utilizada, las tecnologías, y las herramientas definidas por el grupo de arquitectura del proyecto Control de Acceso y que son utilizadas para modelar la solución propuesta.

En el **Capítulo 2** mediante el modelado del dominio se aborda el entorno donde se implantará el sistema y se presentan los requerimientos funcionales y no funcionales con los cuales el mismo debe cumplir. Se realiza una propuesta de sistema a través de la definición de los casos de usos del sistema que guiarán todo el proceso de desarrollo. Se analiza si es viable o no llevar a cabo el flujo de trabajo análisis y diseño.

En el **Capítulo 3** se nombran los patrones de diseño y de arquitectura a utilizar y la aplicación de los mismos. Se exponen los diagramas de clases del análisis, diseño, los diagramas de secuencia así como la descripción de forma general de las clases utilizadas.

En los anexos se muestra parte de la documentación generada como resultado del flujo de trabajo de diseño.

### CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

#### 1.1 Introducción

En este capítulo se estudiarán las principales características de las imágenes digitales y otros aspectos fundamentales para la comprensión del sistema que se desea modelar y que son de vital importancia para el desarrollo de la investigación. Se realizará una exposición del estado del arte de los sistemas de gestión de fotos a nivel internacional y en la UCI. Se describirá los lenguajes de programación, la metodología de desarrollo de software y las herramientas a utilizar para el diseño del sistema, todas propuestas por el equipo de arquitectura del proyecto Control de Acceso.

#### 1.2 Imágenes digitales

El hombre siempre ha sentido la necesidad de representar visualmente con precisión las cosas de su cotidianidad, por esta razón desde tiempos remotos utilizó la pintura rupestre y con el paso de los años esta manifestación fue desarrollándose hasta llegar al surgimiento de la fotografía en 1816, pero no fue hasta 1990 que se revolucionó gracias a la aparición de las imágenes digitales y la vinculación de las mismas al sector informático.

##### 1.2.1 Características de una imagen digital

Las características fundamentales que varían entre los diferentes formatos ya abordados son:

- Profundidad de color se trata del número máximo de colores diferentes que puede contener una imagen en un formato
- Compresión si el almacenamiento de la información binaria es tal cual, o previo paso por una etapa de compactación de la información. Las imágenes en formatos comprimidos de un tamaño y profundidad de color dados ocupan menos que las no comprimidas.

##### 1.2.2 Formatos de almacenamiento de imágenes digitales

A la hora de almacenar una imagen digital se puede elegir de entre varios formatos en que la información de tonos, brillos y contrastes va a ser recogida. Unos de ellos son comprimidos: algunos con pérdidas y otros sin ellas; otros en cambio no tienen compresión alguna.

La selección del formato depende de las necesidades en un momento en particular ya que en algunos casos interesa que ocupen poco espacio porque se tenga una memoria limitada, o porque se quiera reducir el tiempo de descarga desde Internet. En otros casos, interesará obtener la máxima calidad posible en la imagen siendo poco relevante el espacio que el fichero pueda ocupar.

En la actualidad los principales formatos de imágenes digitales son:

**TIFF:** Formato de almacenamiento de alta calidad y sin pérdidas que usan muchas cámaras digitales. Utilizado en los programas de retoque de imágenes digitales, admite una profundidad de color de 64 bits, y con el uso de un algoritmo de compresión sin pérdidas consigue reducir su nivel de espacio.

**RAW:** Formato que se utiliza como alternativa a TIFF. Almacena directamente la información que proviene del sensor de la cámara digital. El formato no es estándar entre los fabricantes. Para trabajar en la PC o imprimir imágenes en este formato hay que convertirlas a otro estándar. La conversión a formato TIFF es un proceso lento y requiere mayor espacio de almacenamiento. Sin embargo, el nivel de calidad que tienen las imágenes en formato RAW es semejante al de las imágenes TIFF.

**JPEG:** Es uno de los formatos más populares, siendo uno de los más utilizados en Internet. Permite almacenar y transmitir las imágenes ocupando muy poco espacio, aunque con pérdida de calidad. Afortunadamente se puede decidir el nivel de pérdidas (y por tanto de calidad) que se desea tener. Aún con los niveles de calidad más altos en JPEG el ahorro de espacio es considerable en comparación con un fichero TIFF.

**GIF:** Es un formato popular en Internet. Utiliza un algoritmo de compresión sin pérdidas. Sin embargo, la calidad en las imágenes no llega a ser muy alta por su limitada profundidad de color (sólo 8 bits). Permite transparencias e imágenes rodantes (que reciben el nombre de GIFs animados).

**PNG:** Es un formato de Internet, aunque menos popular que JPEG y GIF. Ha sido concebido como el sustituto de GIF, incrementando su profundidad de color (hasta los 48 bits) y utilizando un mecanismo de compresión sin pérdidas mejorado.

**PSD:** Formato nativo del software de retoque fotográfico *Photoshop*<sup>1</sup>. Admite capas, texto y almacena el estado de edición / manipulación en que puede haber quedado una imagen. Permite almacenar las imágenes con la calidad más alta, consumiendo gran espacio en disco.

Formato	Compresión / Tipo	Profundidad de color	Uso típico
TIFF	Opcional / Sin pérdidas	1 a 64 bits	Imágenes de alta calidad, cámaras digitales, escáneres, impresión.
RAW	Sin pérdidas	48 bits	Cámaras digitales
JPEG	Con pérdidas	8 o 24 bits	Cámaras digitales, Internet, impresión, intercambio de imágenes
GIF	Sin pérdidas	1 a 8 bits	Internet, imágenes de reducido tamaño, logos.

PNG	Sin pérdidas	1 a 48 bits	Internet, gráficos, iconografía software
PSD	Sin pérdidas	1 a 64 bits	Edición y manipulación

**Tabla 1. 1: Principales características de los formatos de imágenes digitales**

### **1.2.3 ¿Qué significa compresión con pérdidas?**

Se trata de un mecanismo de compactación de la información de las imágenes digitales en que se consiguen unos elevados ratios de compresión. Aunque conllevan pérdida en la información y por tanto en la calidad de la imagen. En los formatos de compresión con pérdidas se aplican algoritmos que permiten decidir cual es la información menos relevante para el ojo humano y la desechan. A mayor cantidad desechada mayor compresión, menor espacio, pero también menor calidad. (1)

### **1.2.5 Resolución en imágenes digitales**

La resolución se refiere a la capacidad de una tecnología o un mecanismo para reflejar los detalles de una imagen. La forma de traducir una fotografía en bits para poder manejarla como archivo informático es dividirla según una malla de filas y columnas. A las unidades resultantes se les llama píxeles: son todos del mismo tamaño y representan áreas cuadradas de la imagen original. Si se divide la imagen en pocos píxeles, se podrá codificarla con poca información, pero seguramente se perderá mucho detalle, por lo que se dice que tiene poca resolución. Si se divide en muchas más partes, éstas llegarán a ser tan pequeñas que no se distinguirán. La visión de la imagen será mucho mejor y más detallada, pero también mucho más costosa en bits. Un aspecto importante es que, salvo limitaciones en la tecnología que se utilice, el tamaño y la frecuencia de los píxeles siempre son a voluntad. Los frecuentes equívocos en el uso de la palabra resolución se resuelven distinguiendo en la imagen tres tipos de tamaño: en píxeles, informático y superficial.

#### ➤ Tamaño informático

Se cuenta en unidades de información como bytes, kilobytes o megabytes, y depende directamente de dos cosas: del número de píxeles y de la cantidad de bytes que se gastan para definir cada píxel. La profundidad de bits permite diferenciar y aplicar un número más o menos grande de colores. La mayoría de las cámaras digitales utilizan la profundidad de 24 bits del modo RGB, por lo que cada píxel se anota con 3 bytes. Se calcula rápidamente que cada megapíxel ocupará en memoria 3 megabytes (algo menos, porque la máquina no redondea como las personas). En las tarjetas de memoria suele ocupar mucho menos, porque los datos se guardan comprimidos. (2)

### **1.2.6 Presencia de metadatos en las imágenes**

Cuando es capturada una fotografía digital a esta se le adhiere datos que están un poco ocultos y son imposibles de borrar, varían desde la fecha en que fue tomada, datos de la cámara y en ocasiones puede contener información personal. La mayoría de estas fotografías viajan por la red donde usualmente estos metadatos se pierden al procesar la imagen. En algunos casos esta información llega íntegra a la web donde ya existen servicios web que muestran los datos de las imágenes -ó sus URLs-, lo que conduce a un problema en cuanto a la confidencialidad de la información que viaja con las imágenes.

La presencia de datos ocultos (metadatos) en las imágenes puede acarrear problemas de seguridad de información para su dueño pero a la vez constituye una fuerte herramienta si se sabe trabajar con ella, sobre todo en términos de seguridad y de búsqueda, pues si se tiene una descripción oculta de la imagen, la búsqueda de esta resulta muy fácil cuando se trabaja con grandes cantidades de fotos, además de saber en todo momento a quién o a qué pertenece la imagen con la cual se trabaja.

### **1.2.7 Metadatos y JPEG como formato de fichero de imagen digital**

Según el estándar JPEG, un bloque de comentario puede contener texto arbitrario. No puede asignarse un tipo al texto, JPEG limita cada bloque de comentarios a 64K, pero puede haber tantos bloques como sean necesarios, así que pueden añadirse tantos bloques de texto como se quiera.

### **1.3 Sistemas informáticos y gestión de imágenes digitales para la identificación de personas.**

Las tecnologías de la información y las comunicaciones han alcanzando un desarrollo vertiginoso sobre todo en los países desarrollados, brindando la posibilidad de que varios procesos útiles en el campo de la seguridad, control, e identificación de personas se automaticen continuamente, entre ellos lo de gestión de las imágenes de identificación de personas.

El tratamiento de imágenes implica procedimientos que normalmente se expresan en forma de algoritmos. Así que la mayor parte de las funciones de tratamiento de imagen pueden ser implementadas en software por lo que el hardware específico ha sido poco a poco desplazado por los ordenadores personales de propósito general, los cuales se han adaptado, integrando cada vez más hardware de procesado de imágenes para realizar operaciones a mayor velocidad, y más potencia en las memorias temporales.

Por otro lado, el procesado y la gestión de imágenes se caracterizan por ser soluciones específicas para cada problema. Lo que funciona bien en un caso puede ser totalmente inadecuado en otro. Pero se ha conseguido proporcionar un conjunto de herramientas y un punto de arranque mucho más alejado de lo que estaba hace unos años y por mucho menos coste, además hay una gran

disponibilidad de productos comerciales de procesamiento y gestión que varían desde simples álbumes fotográficos online, hasta complejos sistemas aplicados al reconocimiento de imágenes faciales y a la verificación e identificación de personas mediante soluciones biométricas utilizando imágenes.

En realidad la solución de un problema específico todavía necesita un esfuerzo significativo de investigación, es por ello que para el desarrollo del trabajo se hizo un análisis de las características de los principales sistemas informáticos de gestión y procesamiento de imágenes implantados en el mundo y en la UCI con el fin de identificar si alguno satisface las necesidades del problema planteado anteriormente o tener un punto de arranque para identificar los principales beneficios y limitantes que un sistema como este puede producir. A continuación se abordan algunos de ellos.

### 1.3.1 Sistemas informáticos de gestión de fotos a nivel internacional

#### Sistema Photo-RDF<sup>2</sup>

En este sistema las imágenes se digitalizan y guardan como imágenes JPEG y se describen en *RDF* lo que puede hacer que sean más fáciles de encontrar las fotos que se están buscando. Se escriben los metadatos dentro de las fotos con el programa de introducción de datos (y pueden también ser editadas si son necesarias algunas correcciones) lo que proporciona una descripción no visual de las fotos, con lo cual se contribuye a la accesibilidad de las mismas. Las solicitudes desde la web son servidas por Jigsaw, enviando la foto o los metadatos, dependiendo de la forma de solicitud.

#### Visión general del sistema

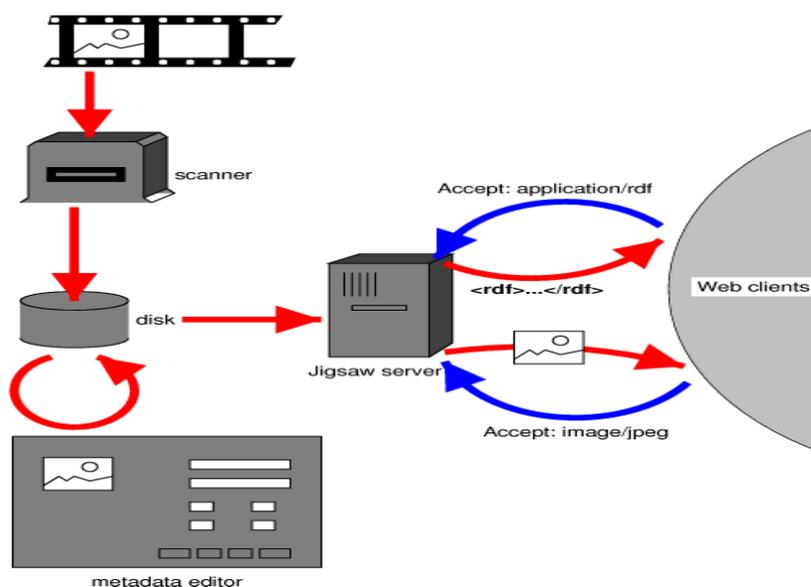


Figura 1: Sistema Photo-RDF

El sistema comprende las siguientes piezas, totalmente independientes:

1. Escanear las fotos y guardarlas en formato JPEG. Para obtener mayor calidad, puede usarse cualquier proceso que produzca JPEG, incluyendo cámaras digitales.
2. Un programa de introducción de datos que permite introducir/editar los metadatos de manera sencilla para cada una de las fotos y almacenar los datos en formato RDF dentro del archivo JPEG.
3. Un módulo para el servidor Jigsaw que puede servir tanto los datos de la imagen JPEG como la descripción RDF que está almacenada en ella, usando negociación de contenido HTTP<sup>3</sup> para determinar cuál de las dos desea el cliente.

El proyecto, consiste en tomar piezas de la tecnología existente (RDF, HTTP y Jigsaw del W3C<sup>4</sup>; JPEG, Java<sup>5</sup>) y proporcionar algo de unión entre ellas para producir una aplicación útil.

### El programa de introducción de datos "rdfpic"

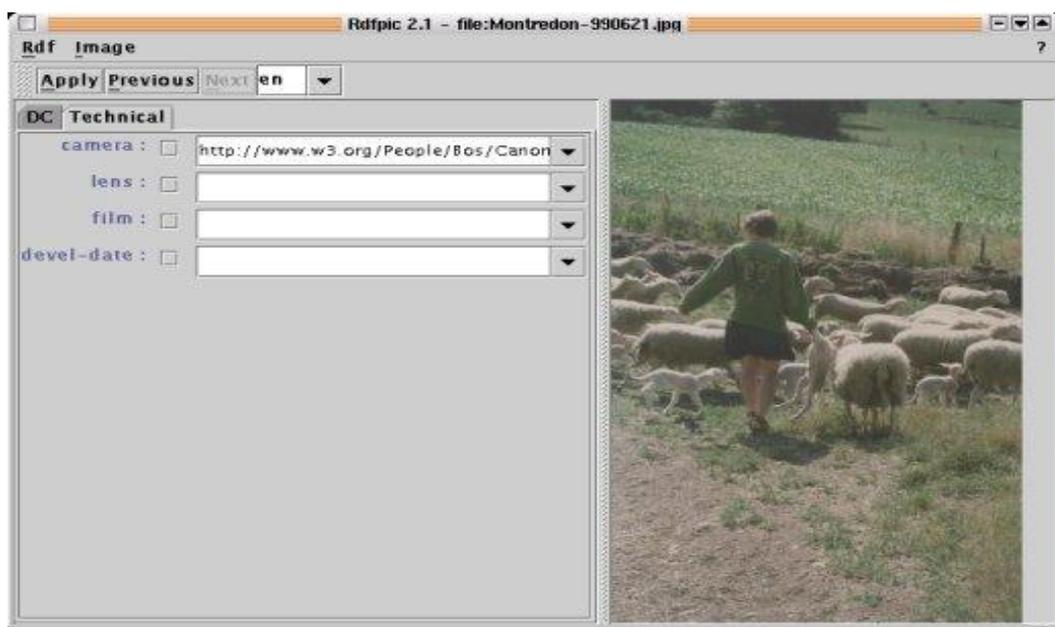


Figura 2: Vista de la interfaz del programa de introducción de datos

El programa de introducción de datos está diseñado para habilitar la introducción rápida de metadatos, asumiendo que las fotos normalmente serán de una o unas pocas series. La mayoría de los campos muestran por defecto el valor que fue introducido para la foto anterior, y ofrece un acceso rápido a los valores introducidos para las últimas fotos. Sólo deberán cambiarse los valores de unos pocos campos de una foto a la siguiente y la cantidad de escritura se reducirá.

El programa está escrito en *Java*, pero la interfaz de usuario se genera mientras el programa está en marcha, directamente desde una versión de los esquemas legible por un ordenador (normalmente no la sintaxis RDF sino una transformación de ella, con información equivalente). Esto significa que el programa no necesita ser cambiado cuando se cambia los esquemas RDF.

### ***La extensión Jigsaw***

Se utilizó la negociación de contenido como mejor manera de servir tanto la versión RDF como la imagen completa, usando los navegadores y herramientas existentes. Por supuesto, esto no excluye el uso de otras técnicas, tales como las extensiones HTTP, para recuperar y almacenar metadatos.

A pesar de que este sistema cumple con la mayoría de las funcionalidades requeridas no es factible ponerlo en explotación en la Universidad pues se necesita que todas estas funcionalidades las realicen aplicaciones web y servicios web que sirvan a otras aplicaciones clientes y estas a su vez le sirvan datos a él, los cuales formarán parte de los datos no visibles de las fotos, además de que este sistema no permite darle el formato ni la resolución que se desee a las fotos y trabaja con pocos volúmenes de imágenes a la vez.

Otras investigaciones realizadas reflejaron la existencia de una nueva tendencia para la gestión y procesamiento de imágenes de identificación de personas, basada en sistemas automatizados de identificación de personas mediante soluciones biométricas que integran hardware y paquetes de software para obtener resultados muy precisos sin margen de error, tal es el caso de los sistemas que identifican a las personas mediante el reconocimiento facial, del iris del ojo o de huellas dactilares a partir de la exposición de las mismas delante de un captador que captura las imágenes que luego son procesadas y comparadas con otras almacenadas en potentes bases de datos. Un ejemplo que ilustra lo anteriormente dicho es AFIS<sup>6</sup>.

Estos sistemas serían ideales ponerlos en explotación pero son completamente propietarios y muy costosos por lo que no es factible implantar alguno en la UCI.

### **1.3.2 Sistemas informáticos de gestión de fotos en Cuba**

La mayor concentración de esfuerzos para el desarrollo de software que existe en Cuba actualmente vinculado a la gestión de imágenes se encuentra en la UCI, aunque se desarrollan productos en centros de alto prestigio. Hasta ahora casi todos han sido desarrollados para la gestión de imágenes médicas vinculadas al polo científico y pocos para el manejo de la identificación visual del personal existente en la Universidad.

### **Alas PACS<sup>7</sup>**

Es un sistema de gestión de fotos creado por el Grupo de Procesamiento de Imágenes de la Universidad de las Ciencias Informáticas con el objetivo de convertir a Cuba en un país a la vanguardia, en la utilización de las más modernas tecnologías de base científico imagenológicas. Provee por su carácter científico, en el tema de procesamiento y gestión digital de imágenes y señales, al sistema nacional de salud y a otros centros e instituciones.

Básicamente este sistema lo que hace es trabajar con algoritmos matemáticos para procesar las imágenes que se obtienen principalmente de equipos médicos de los hospitales, aunque también pueden ser obtenidas de internet o tiradas por el propio equipo con cámaras digitales. Dentro de los algoritmos para procesar las imágenes se encuentran los que le dan formato, resolución, los algoritmos de reconocimiento de patrones, transformaciones morfológicas y lineales.

Las herramientas que utilizaron para lograr el software fueron Matlab, C# como el lenguaje más utilizado, C++ y como IDE<sup>8</sup> Visual Studio 2005.

Este sistema no cumple objetivo ponerlo en práctica porque está desarrollado bajo software de licencia privativa, básicamente fue confeccionado con el objetivo de procesar y gestionar imágenes médicas aunque también procesa imágenes de identificación de personas pero en poca medida, no introduce metadatos en sus fotos de forma tal que no se tienen muchas opciones para acelerar el proceso de la identificación de las imágenes.

### **Identidad**

Es un proyecto que consta de varios módulos tal es el caso de Cedulación, Pasaporte, Extranjería y Emigración que entre las funcionalidades de esto módulos esta la gestión, almacenamiento y procesamiento de imágenes de identificación de personas, estas imágenes pueden ser faciales, firmas o huellas digitales.

Este sistema es un software hecho para las necesidades propias de un cliente extranjero, está desarrollado totalmente bajo software de licencia privativa, no es multiplataforma, el hardware que se utiliza para el control de la captación de imágenes faciales y el captador de huellas digitales es propietarios y muy caro por lo que no es factible ponerlo en explotación en la Universidad.

### **1.4 Metodología, lenguajes, tecnologías y herramientas utilizadas en el desarrollo**

La metodología, lenguajes, herramientas y tecnologías utilizadas en la realización del análisis y diseño del sistema de gestión de fotos fueron definidos por el grupo de arquitectura, los directivos del proyecto

de Control de Acceso y el grupo de Informatización de la Universidad. Es por ello que en este epígrafe solo se realiza una identificación de cada una de ellas.

### 1.4.1 Metodología de desarrollo de software

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

La metodología de desarrollo definida para el desarrollo del sistema Gestión de Fotos es RUP.

#### **Rational Unified Process (RUP)**

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, es un proceso que se caracteriza por ser:

- Dirigido por casos de uso. Los casos de uso describen los requisitos funcionales del sistema desde la perspectiva del usuario y se usan para determinar el alcance de cada iteración y el contenido de trabajo de cada persona del equipo de desarrollo.
- Centrado en la arquitectura. La arquitectura permite ganar control sobre el proyecto para manejar su complejidad y controlar su integridad. Hace posible la reutilización a gran escala y provee una base para la gestión del proyecto.
- Iterativo e incremental. Se divide en 4 fases: Inicio, Elaboración, Construcción y Transición, y cada una de ellas se divide en iteraciones. En cada iteración se trabaja en un número de disciplinas haciendo énfasis en algunas de ellas. Las disciplinas propuestas por RUP son: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, entre otras. Cada iteración añade funcionalidades al producto de software o mejora las existentes. (3)

Divide en 4 fases el desarrollo del software cuyos objetivos son:

- Inicio: determinar la visión del proyecto.
- Elaboración: definir la arquitectura del sistema.
- Construcción: obtener la capacidad operacional inicial.
- Transmisión: el release del producto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

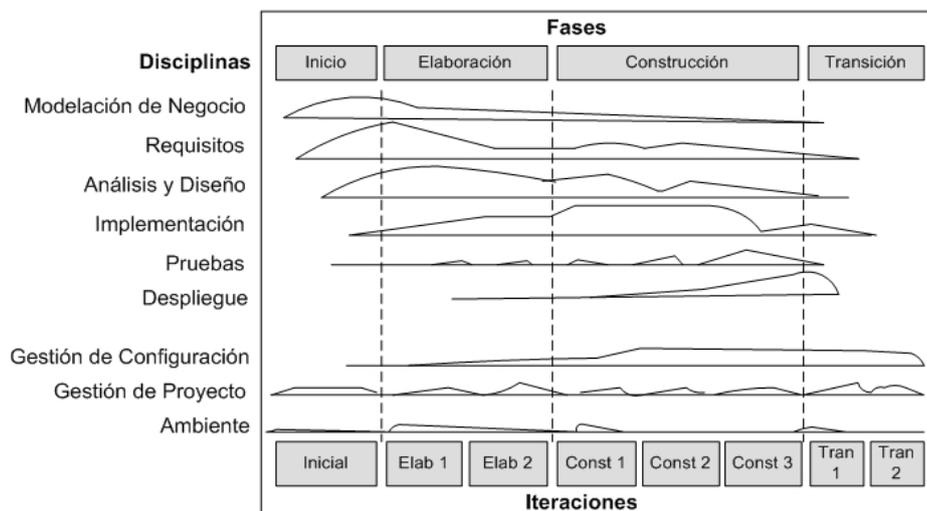
Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos tipos de disciplinas:

**Disciplinas de desarrollo**

- Modelado del Negocio: Entendimiento de las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Crear el software que se ajuste a la arquitectura y que tenga el comportamiento especificado en el diseño.
- Pruebas: Asegurar que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

**Disciplinas de soporte**

- Configuración y administración del cambio: Gestionar todas las versiones del proyecto.
- Administrando el proyecto: Administrar horarios y recursos.
- Ambiente: Definir y administrar el ambiente de desarrollo.
- Despliegue: Realizar acciones relativas al despliegue del sistema como manuales de usuario y puesta a punto del SW.



**Figura 3:** Fases e Iteraciones de la Metodología RUP

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Para la construcción de los artefactos que RUP propone, para cada uno de los flujos de trabajos se utiliza UML<sup>9</sup> como lenguaje de modelado.

### 1.4.2 Lenguaje de modelado

#### UML

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se utiliza para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir ya que capta la información sobre la estructura estática y el comportamiento dinámico de estos.

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. Es un lenguaje de propósito general para el modelado orientado a objetos y el modelado visual que permite una abstracción del sistema y sus componentes.

La importancia de UML radica en que actualmente es considerado un estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

### 1.4.3 Herramienta CASE<sup>10</sup>

Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software (Investigación Preliminar, Análisis, Diseño, Implementación e Instalación).

CASE es también definido como el conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

La herramienta CASE de modelado utilizada para documentar el análisis y diseño realizado como parte de las tareas científicas del trabajo es Visual Paradigm.

### **Visual Paradigm**

Visual Paradigm para UML es una herramienta UML CASE, considerada completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad<sup>11</sup> con otras aplicaciones. Fue diseñada para el ciclo completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Permite generación de código a partir de los modelos, ingeniería inversa y emitir informes. Crea el esquema de clases a partir de una base de datos y la definición de base de datos a partir del esquema de clases. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del paradigma orientado a objetos. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

#### **1.4.4 Aplicaciones web**

Las aplicaciones web se ejecutan en el servidor Web, encargándose de controlar el estado de un negocio o una problemática dada mediante el uso de uno o varios lenguajes de programación. De esta forma se generan páginas dinámicas cuyo contenido la mayor parte de las veces está vinculado con los datos almacenados en bases de datos relacionales. Actualmente las aplicaciones web cuentan con gran popularidad debido al auge que tuvo Internet en la década pasada, y dado que brindan grandes posibilidades de despliegue en oposición a las aplicaciones de escritorio por el simple hecho que los usuarios solo necesitan un navegador para hacer uso de ellas. Además, estas aplicaciones permiten el aprovechamiento de todas las características de Internet, estando disponibles desde cualquier parte del mundo donde se tenga acceso a la red de redes. Son fáciles de usar, ya que no requieren conocimientos avanzados por parte de los usuarios. Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información lo cual permite que el usuario acceda a ella de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo diligenciamiento de formularios, participar en juegos diversos y acceder a manejador de base de datos de todo tipo.

#### **1.4.5 Servicios web**

Los servicios web son componentes de software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, utilizando Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas o invocadas a través de protocolos Web estándar, como XML<sup>12</sup>, WSDL, SOAP<sup>13</sup> o UDDI. Los Servicios Web realizan funciones que pueden ser de todo tipo: desde simples peticiones hasta complejos procesos de negocio. Una vez puesto en marcha, otras aplicaciones pueden localizarlo e invocar el servicio. Los objetivos principales que persiguen los servicios web son la interoperabilidad y la integración. Esta

última permite obtener la información solicitada en tiempo real, agilizando el proceso de toma de decisiones.

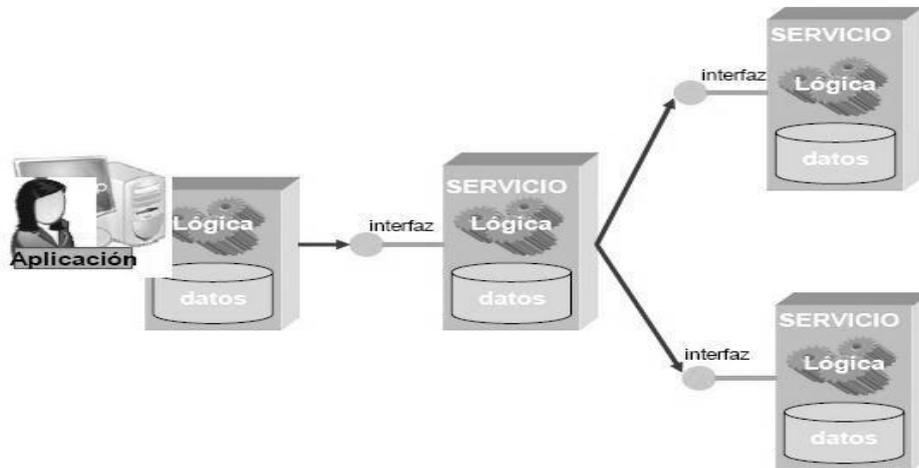


Figura 4: Sistema basado en servicios.

En la figura 1.5 se representa un sistema sencillo basado en servicios. Una aplicación además de implementar sus propios componentes de negocio y datos, también puede reutilizar la funcionalidad de servicios existentes en la red empresarial. Un servicio debe ser una aplicación completamente autónoma e independiente. A pesar de esto, expone una interfaz de llamado basada en mensajes, capaz de ser accedida a través de la red. Generalmente los servicios incluyen tanto la lógica de negocio como los datos relevantes a la solución del problema para el cual fueron diseñados. La comunicación hacia y desde el servicio, se realiza utilizando mensajes que deben contener o referenciar toda la información necesaria para entenderlo. (4)

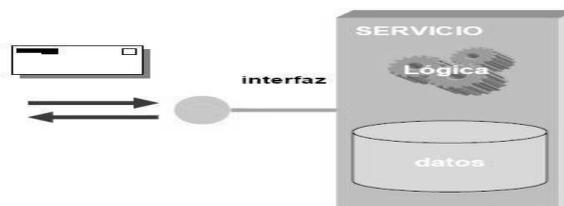


Figura 5: Visión interna de los servicios.

### Protocolos que utilizan los servicios web XML

XML no es sólo un lenguaje, es un conjunto de reglas que sirven para definir etiquetas semánticas para organizar un documento. Todo lenguaje que se exprese de una forma determinada puede ser XML. No solo se utiliza en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de

texto y hojas de cálculo. Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

### **SOAP**

Es un protocolo basado en XML para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP. Existen varios modelos de mensajes SOAP, pero el más común es el RPC, en donde un nodo de la red (el cliente) envía un mensaje de solicitud a otro nodo (el servidor) y este último responde al mensaje del cliente. Estos mensajes son independientes del sistema operativo y pueden transportarse en protocolos como SMTP<sup>14</sup>, MIME<sup>15</sup> y HTTP.

SOAP ha recibido gran apoyo siendo el primer protocolo de su tipo que ha sido aceptado prácticamente por todas las grandes compañías de software del mundo. Entre las compañías que soportan SOAP son Microsoft, IBM, SUN Microsystems, SAP y Ariba.

### **WSDL**

El lenguaje de descripción de servicios Web (WSDL, Web Service Description Language) es un dialecto basado en XML sobre el esquema que describe un servicio Web. Un documento WSDL proporciona la información necesaria al cliente para interactuar con el servicio Web. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red, incluyendo SOAP sobre HTTP e incluso protocolos que no se basan en XML como DCOM<sup>16</sup> sobre UDP<sup>17</sup>. Los documentos WSDL definen los servicios como colecciones de puntos finales de red o puertos. En WSDL, la definición abstracta de puntos finales y de mensajes se separa de la instalación concreta de red o de los enlaces del formato de datos. Esto permite la reutilización de definiciones abstractas: mensajes, que son descripciones abstractas de los datos que se están intercambiando y tipos de puertos, que son colecciones abstractas de operaciones. Describe cuatro piezas principales:

- Información sobre la interfaz exportada.
- Tipo de datos de los mensajes intercambiados (peticiones, respuestas, errores).
- Información de brindar la definición de cómo transportar los mensajes en la comunicación.
- Dirección para localizar los servidores.

## UDDI

UDDI (Descripción Universal, Descubrimiento e Integración) es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que estas ofrecen. A través de UDDI, se puede publicar y descubrir información de una empresa y de sus servicios. Se pueden utilizar sistemas taxonómicos estándar para clasificar estos datos y poder encontrarlos posteriormente en función de la categorización. Lo más importante es que UDDI contiene información sobre las interfaces técnicas de los servicios de una empresa. A través de un conjunto de llamadas a basadas en SOAP, se puede interactuar con UDDI tanto en tiempo de diseño como de ejecución para descubrir datos técnicos de los servicios que permitan invocarlos y utilizarlos.

### 1.4.6 Arquitectura orientada a servicios

La arquitectura orientada a servicios (SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario. Al contrario de las arquitecturas orientadas a objetos, las SOAs están formadas por servicios de aplicación débilmente acoplados y altamente interoperables. Los servicios web se han convertido en el estandarte de SOA, ya que esta tecnología posee un conjunto de características que permiten cubrir todos los principios de la orientación a servicios. La definición de la interfaz encapsula las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo. Con esta arquitectura, se pretende que los componentes software desarrollados sean reusables, ya que la interfaz se define siguiendo un estándar. En la figura, se presenta la estructura básica de funcionamiento de un SOA tradicional.

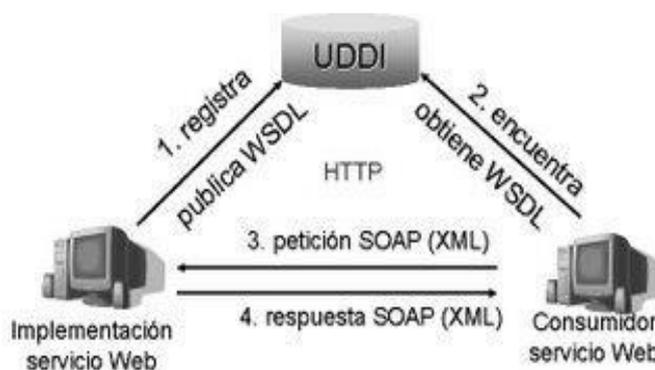


Figura 6: Arquitectura SOA tradicional.

Se puede observar la existencia de tres roles claramente diferenciados:

- Cliente del servicio: Es el que solicita la ejecución del servicio web, y por lo tanto el que lo consume.

- Proveedor del servicio: Es el encargado de implementar el servicio web y ofrecerlo a los clientes.
- Registro del servicio: Es un repositorio donde se almacenan las descripciones de los servicios, para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades.

### 1.4.7 Lenguajes de programación web

#### Lenguajes del lado del servidor

##### PHP<sup>18</sup>

Es un lenguaje script incrustado en las páginas HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características especiales del lenguaje. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. (5) Una de sus características más potentes es su soporte para gran cantidad de bases de datos entre las cuales se encuentran InterBase, MSQL, MySQL, Oracle, Informix y PosgreSQL.

PHP no soporta directamente punteros por lo que evita los problemas de depuración provocados por estos y permite optimizar la cantidad de líneas de código. Es un lenguaje multiplataforma, funciona en todas las plataformas que soporten Apache. Está publicado bajo licencia GPL.

Sus características más relevantes son:

- Velocidad: Tiene gran velocidad de ejecución y evita demoras en la máquina. No requiere muchos recursos del sistema.
- Estabilidad: PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- Seguridad: El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini.
- Simplicidad: Les permite a los programadores generar código productivamente en el menor tiempo posible. (6)

#### Lenguajes del lado del cliente

##### JavaScript

Es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Con JavaScript se puede crear efectos especiales en las páginas y definir interacciones con el usuario. El navegador del cliente es el encargado de

interpretar las instrucciones JavaScript y ejecutarlas, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. JavaScript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras complejas de datos.

### 1.4.8 Framework

Un framework es una mini arquitectura reusable que provee una estructura y comportamiento genéricos para una familia de abstracciones de software]. (7) Es un conjunto de componentes con interfaces bien definidas que interactúan entre sí para cumplir una tarea.

*GoF*<sup>19</sup> define un framework como “un conjunto de clases que constituyen un diseño reutilizable para un tipo específico de aplicaciones”.

El framework que soportará el desarrollo del sistema gestión de fotos es Symfony.

#### **Symfony**

Es un completo framework desarrollado con PHP. Diseñado para optimizar el desarrollo de las aplicaciones Web, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Automatiza las tareas más comunes permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Compatible con la mayoría de los gestores de base de datos (MySQL, PostgreSQL, Oracle y SQL Server de Microsoft) y con la mayoría de las plataformas. Es fácil de instalar, sigue la mayoría de las mejores prácticas y patrones de diseño para la Web, es sencillo de utilizar y suficientemente flexible para adaptarse a los casos más complejos. Permite su integración con librerías desarrolladas por terceros, con código es fácil de leer.

Utilizado fundamentalmente para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

#### **Propel**

Es un Framework programado en PHP5 de persistencia de Datos y consulta, lo que significa que brinda un mecanismo para almacenar objetos PHP en una base de datos y un sistema para búsqueda y restauración de objetos PHP desde una base de datos, todo esto a partir de un mecanismo de abstracción de los datos y utilizando ORM<sup>20</sup>. Propel permite realizar consultas complejas y manipulación de datos sin escribir una sola cláusula SQL. Hace más fácil la escritura de aplicaciones,

el despliegue y mucho más migrar la aplicación a otro gestor de Base de Datos si alguna vez la situación lo amerita.

Symfony utiliza a Propel como ORM y Propel utiliza Creole como capa de abstracción de bases de datos.

### 1.4.9 Plugin

Los plugins son librerías de archivos que cumplen un propósito específico. Pueden ser considerados componentes que se le agregan a las aplicaciones para aumentar su rendimiento y extender sus funcionalidades. Además de que brinda la posibilidad de reutilizar una porción de código en determinadas partes de la aplicación o entre distintas aplicaciones, sobre todo cuando este código comprende algo más que una clase.

#### **CkWebServicePlugin**

Es un plugin para Symfony que permite exponer los módulos y acciones como un servicio web, el plugin está basado en el módulo para la utilización estándar SOAP de PHP, Ofrece generación automática de archivos .wsdl con parámetros de entrada y tipos de retornos, a partir del código fuente utilizando un generador WSDL. El plugin permite, la configuración de los parámetros pasados por la llamada SOAP, para poder usarlos como parámetros de la respuesta. También permite configurar en cual acción se guardará el resultado, y si no se especifica es asumida.

### 1.5 Conclusiones

En este capítulo se enmarca al lector en los conceptos básicos de la fotografía digital, sus características, tipos de formatos, e importancia de usar los metadatos como identificador dentro de las imágenes.

Se analizan sistemas de gestión de fotos a nivel internacional y en la UCI, por lo que se concluyó que ninguno es factible ponerlo en explotación pues no cumplen con todas las funcionalidades que se requieren en la Universidad para un sistema de este tipo.

Se exponen la metodología, las herramientas y tecnologías utilizadas las cuales cumplen con las políticas de utilización de software libre de la Universidad, por lo que trae como consecuencia positiva una reducción notable del costo del sistema por concepto de licencias de software y soporte, además de que cuentan a nivel internacional con una gran comunidad de apoyo y desarrollo tal es el caso de PHP, Symfony y los servicios web.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### 2.1 Introducción

En este capítulo se realizará el análisis de los objetivos propuestos en la UCI para la gestión de las fotos de identificación de personas que en ella se manejan. Se analizarán los procesos del negocio que soportan a dichos objetivos. Se estudiará el entorno donde se implantará el futuro sistema mediante el modelado del dominio. Se explicará el comportamiento del sistema a través de los requisitos funcionales y no funcionales, casos de uso y su relación con los actores del sistema, a través del diagrama de casos de usos. Para la documentación de los requisitos y el negocio se generarán los artefactos propuestos por la metodología de desarrollo RUP para los flujos de trabajo Modelado del Negocio y Requerimientos. Por último se describe un estudio realizado de la factibilidad del software.

### 2.2 Flujo actual de los procesos

Actualmente, los procesos de negocios en la Universidad para la gestión de las fotos de identificación de personas, suceden de la siguiente manera:

Cuando a una persona se le da alta en el departamento de Capital Humano sus datos de identificación se almacenan en una base de datos central, donde se guarda toda la información referente al personal de la Universidad ya sea estudiante, o trabajador. Seguidamente esta persona pasa al departamento de acreditación donde se le toma una foto facial que lo identificará dentro de la UCI, esta foto es procesada y posteriormente almacenada por una de las aplicaciones pertenecientes al sistema Acreditación de Personas en el directorio de imágenes, la cual se encarga también de asociarle a los datos de la persona en cuestión, la dirección física de donde se encuentra la imagen dentro del directorio.

Por otro lado casi todas las aplicaciones de la Universidad acceden a estas fotos y se la muestran a sus respectivos usuarios.

### 2.3 Análisis crítico de la ejecución de los procesos

Estas aplicaciones que trabajan con las fotos de identificación de personas tanto para guardarlas como para buscarlas lo hacen de forma autónoma y por recursos compartidos, contribuyendo a que ocurra una sobre carga de operaciones con la base de datos que guarda la dirección física de la foto como del directorio donde se encuentran las imágenes, puesto que se tienen varias aplicaciones solicitando los mismos servicios de forma independiente.

Por otro lado los directivos de seguridad y protección solo cuentan con estas fotos y con las limitaciones de los sistemas que las gestionan actualmente para cumplir con las tareas de protección del centro.

### **2.4 Propuesta de sistema**

Teniendo en cuenta la importancia estratégica para la Universidad de lo mencionado anteriormente y con el propósito de dar cumplimiento a los objetivos propuestos, el trabajo se centrará en el análisis y diseño de una propuesta de sistema basado en:

- Servicios web: serán los encargados de gestionar todos los procesos vinculados a las imágenes faciales de las personas y a los cuales podrán acceder el resto de las aplicaciones de la UCI que necesiten de ellas para brindar alguna funcionalidad. Los servicios brindados básicamente serán guardar, procesar y buscar las imágenes correspondientes a una persona. Es necesario destacar que el servicio que brinda la foto podrá ser consumido por cualquier sistema y el que guarda las imágenes solo por los sistemas autorizados a hacerlo en este caso el sistema Acreditación de Personas.
- Aplicación web: será la encargada de permitir que se manejen los procesos de administración de todo el sistema por parte de los administradores quienes serán los únicos autorizados a acceder a estas funcionalidades y la gestión de las fotos que sean del interés para los directivos de seguridad y protección de la Universidad. Esta gestión consistirá en estandarizar las imágenes, guardarlas, eliminarlas, brindarlas, permitir un historial de fotos por personas y el trabajo con los metadatos que se insertan dentro de las mismas para ayudar a la identificación de las personas y de las imágenes en consecuencia. A estas funcionalidades los únicos que tienen acceso son los directivos de seguridad y protección de la Universidad.

Como se decía anteriormente los procesos de objeto de automatización serán el procesado y la gestión de las imágenes de las personas con las que se trabaja en la Universidad.

Esta serie de funcionalidades que se estarán modelando responde a la necesidad de encontrar una fórmula que sirva de herramienta de apoyo para lograr un mejor desenvolvimiento en los procesos identificación y de seguridad y protección de la Universidad y que permita a diferentes sistemas informáticos obtener, sin importar la plataforma en la que han sido desarrollados, las fotos que son de su interés de manera centralizada con altos niveles de disponibilidad e inmediatez, pues como se decía antes los sistemas que existen actualmente trabajan localmente y se emplea mucho tiempo y esfuerzo.

### 2.5 Modelo del dominio

Teniendo en cuenta la descripción que se brinda en los epígrafes anteriores se puede inferir que no existe un proceso de negocio puesto que no están bien definidos los flujos de los eventos, así como las funciones que se desarrollan en el ambiente o entorno que se definen, en el cual está enmarcado el problema. Esto sugiere utilizar un modelo de dominio que permite la representación visual de los conceptos u objetos significativos del entorno o área del problema. Para ello se presenta una tabla con los conceptos y descripciones que ayudarán a una mejor comprensión del problema.

Concepto	Descripción
Universidad	Institución donde se desarrolla el dominio del trabajo, la cual tiene una aplicación para gestionar las fotos, y otras que buscan la foto y la muestran a los usuarios que interactúan con ella.
Usuarios	Es el rol que representan el conjunto de personas de la Universidad que utilizan los sistemas para buscar la foto. Las cuales pueden ser de dos tipos: estudiantes y trabajadores.
Estudiantes	Estudiantes que utilizan los sistemas para buscar las fotos.
Trabajadores	Trabajadores que utilizan los sistemas para buscar las fotos. Que pueden ser de dos tipos: usuarios que utilizan los sistemas para buscar la foto o Especialista de acreditación.
Especialista de Acreditación	Es un tipo de trabajador que básicamente representa dos roles dentro del dominio del problema, el primero es como un usuario que utiliza los sistemas de la Universidad para buscar la foto de una determinada persona y el segundo es como el administrador del sistema Acreditación de Personas que gestiona las fotos de las personas en la Universidad.
Sistema	Es el conjunto de aplicaciones que acceden a las fotos. Actualmente existen varias en la Universidad en explotación pero solo se mencionarán las principales que son de dominio público.
Akados	Aplicación que busca y muestra la foto de los estudiantes de la Universidad
Sistema de control de acceso a comedores	Aplicación que busca y muestra la foto de todo el personal existente en la Universidad.
Directorio	Aplicación que busca y muestra la foto de todo el personal existente en la Universidad.
Control de acceso	Aplicación que busca y muestra la foto de todo el personal existente en la Universidad.

Sistema Acreditación de Personas	Aplicación que gestiona las fotos de todo el personal existente en la Universidad.
Base de datos	Es el conjunto de las bases de datos de la Universidad que los sistemas antes mencionados utilizan para buscar o guardar la dirección de las fotos, correspondientes a una persona, y que después se guardan en un directorio.
Akados	Se guarda la dirección de las fotos correspondientes a los estudiantes de la Universidad.
Asset	Se guarda la dirección de las fotos correspondientes a los Profesores de la Universidad.
Terceros	Se guarda la dirección de las fotos correspondientes a los trabajadores de la Universidad.
Repositorio	Es donde se guardan las fotos del personal que se maneja en la Universidad.
Foto	Representación visual del personal que se maneja en la Universidad.

Tabla 2. 1: Definición de las entidades y conceptos principales.

### 2.5.1 Diagrama de clases del modelo de dominio

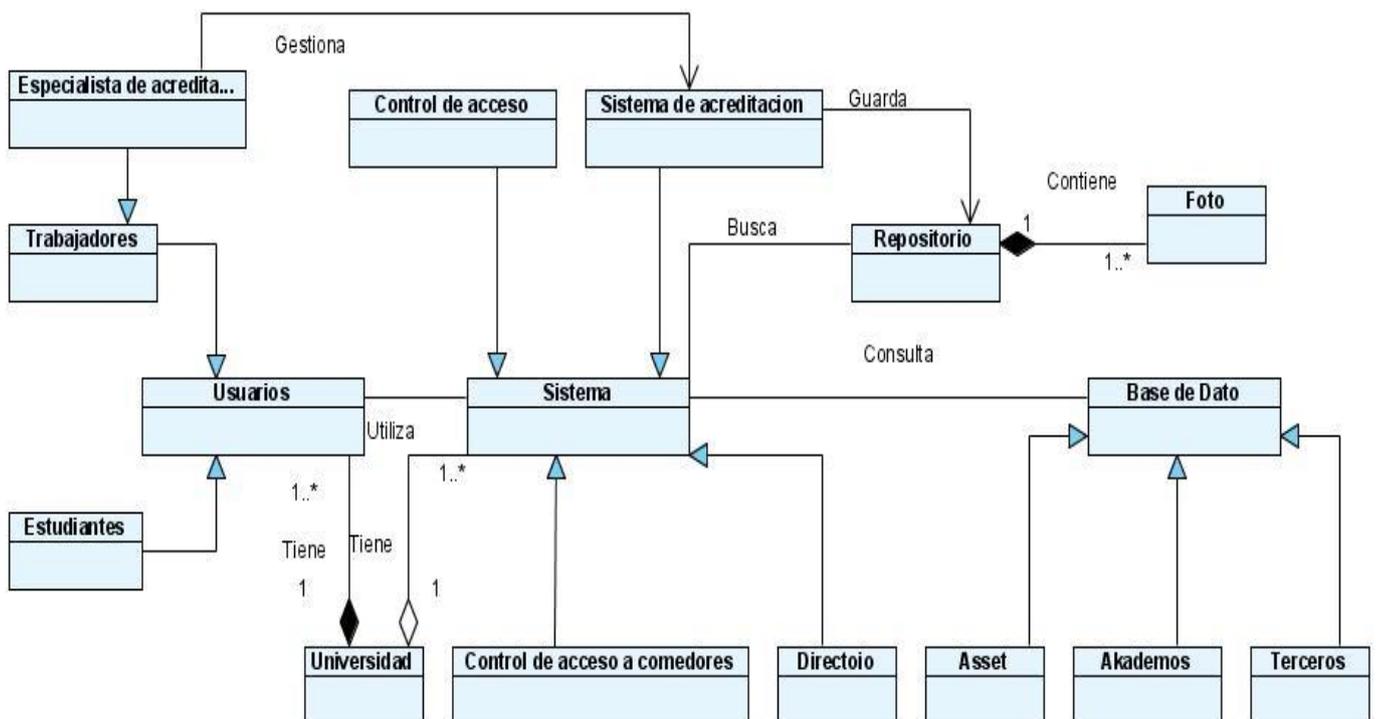


Figura 7: Modelo de dominio.

### 2.6 Especificación de los requerimientos del sistema

Cuando se conocen los conceptos relacionados con el objeto de estudio se puede comenzar a analizar las capacidades o condiciones con las que debe cumplir el sistema propuesto y las propiedades o cualidades que debe tener y se enumeran a través de requerimientos funcionales y no funcionales para que cumpla con los objetivos planteados.

#### Dependencias y relaciones con otro software

El sistema propuesto se relacionará con los servicio de suministro de fotos y datos que proporcionará el nuevo sistema de Acreditación de Personas actualmente en desarrollo, y con los servicios web de la uci, específicamente con los de Registro de Identidad UCI y con los restantes sistemas de la Universidad que brindan la posibilidad a los usuarios de mostrar la foto de una persona en particular.

#### 2.6.1 Requerimientos funcionales

**R1:** Autenticar el usuario, que solicite cualquier servicio, con su usuario y contraseña del dominio UCI

**R2:** Verificar exactitud de los datos y fotos recibidas.

**R3:** Estandarizar imagen

3.1: Dar formato a la imagen.

3.2: Dar resolución a la imagen.

3.3: Dar tamaño a la imagen.

**R4:** Gestionar los metadatos.

4.1: Borrar datos EXIF de las fotos.

4.2: Introducir datos no visibles en las fotos.

4.3 Leer los datos no visibles.

**R5:** Crear directorio para almacenamiento de la imágenes.

5.1: Crear directorio principal.

5.2: Crear directorio seguridad.

**R6:** Brindar los servicios destinados a gestionar la foto facial.

6.1 Guardar foto en el directorio principal.

6.2 Buscar foto en el directorio principal.

**R7:** Permitir historial de fotos.

**R8:** Gestionar las fotos de seguridad.

8.1: Guardar foto en el directorio de seguridad.

8.2 Buscar la foto en el directorio de seguridad.

8.9 Eliminar la foto en el directorio de seguridad.

**R9:** Devolver datos no visibles de las fotos.

**R10:** Gestionar usuarios y sus roles.

10.1 Agregar un nuevo usuario.

10.2 Eliminar un usuario existente.

10.3 Asignar roles a un usuario.

10.4 Eliminarle roles a un usuario.

**R11:** Actualizar las dirección física de la imagen.

### 2.6.2 Requerimientos no funcionales

Las propiedades o cualidades que el sistema propuesto anteriormente debe tener son:

#### Usabilidad

- El tiempo de entrenamiento requerido para que usuarios normales sean productivos operando el sistema es de dos días y el de los avanzados es de un día.
- El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre la computación o hayan trabajado con la web.
- El sistema debe tener una opción de ayuda sobre las principales operaciones que se realizan y sus íconos respectivos para lograr un menor tiempo de aprendizaje.

#### Rendimiento

- El sistema debe ser lo más eficiente posible para poder lograr un tiempo de respuesta adecuado. Este tiempo de respuesta estará dado por la cantidad de fotos a procesar.
- Al igual que el tiempo de respuesta, la velocidad de procesamiento de la foto, la actualización y la recuperación dependerán de la cantidad de fotos que tenga que procesar la aplicación.

#### Seguridad

- Protección contra acciones no autorizadas o que puedan afectar la integridad de la información, es decir, la corrección e integridad de los datos.
- Permitir que un Servicio Web sea utilizado solo por las aplicaciones autorizadas.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.
- Verificación sobre acciones irreversibles (eliminaciones).

### **Confiabilidad**

- Debe mantenerse la consistencia de los datos en correspondencia con la realidad.
- La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores.

### **Disponibilidad**

- El sistema debe estar disponible las 24 horas del día todos los días de la semana.

### **Software**

#### Servidor

Para una correcta ejecución de la aplicación se necesita:

- Sistema operativo servidor: Linux Ubuntu 7.10 o superior.
- Servidor Web: Apache 2.0, con módulo PHP 5.2 disponible.

#### Cliente

- Una computadora personal con plataforma de los sistemas operativos Linux o Windows.
- Navegador: Mozilla Firefox 1.5 o superior, Internet Explorer 6 o superior.

### **Hardware**

#### Servidor

- Velocidad del Procesador: Intel Pentium, 2.4 GHz.
- Mínimo de Memoria RAM para servidor de aplicación: se recomiendan 512 MB.
- Mínimo de Disco duro: 15 GB.

#### Cliente

- Velocidad del Procesador: Intel Pentium, 333 MHZ.

- Mínimo de Memoria RAM: 128 MB.
- Mínimo de Disco duro: 5 GB.
- Tarjeta de red de 100 Mbps.

## **2.7 Descripción del sistema propuesto**

### **2.7.1 Descripción de los actores**

Actor	Descripción
Usuario.	Es el rol que representan el conjunto de aplicaciones de la Universidad que se van a conectar para consumir el servicio.
Sistema Acreditación.	Es el sistema que se conecta para consumir el servicio, ya sea para guardar la foto como para buscarla es una especialización del actor usuario ya que puede realizar las tareas que este realiza.
UASP(usuario de administración, seguridad y protección)	Es el rol que representan el conjunto de personas autorizadas a usar la aplicación web. Las cuales pueden ser el administrador del sistema o el directivo de seguridad y protección.
Administrador de sistema	Es el rol que representa la persona encargada de administrar todas las actividades que se realizan en la aplicación.
Directivo SP	Es el rol que representa la persona encargada de gestionar las fotos de seguridad en la aplicación web
Registro Identidad UCI	Es un sistema externo que se encarga del control de la información de las personas de la UCI y con el cual va a interactuar el sistema para el intercambio de información mediante servicios web.

**Tabla 2. 2: Definición de los actores del sistema.**

2.7.2 Diagrama de casos de usos del sistema

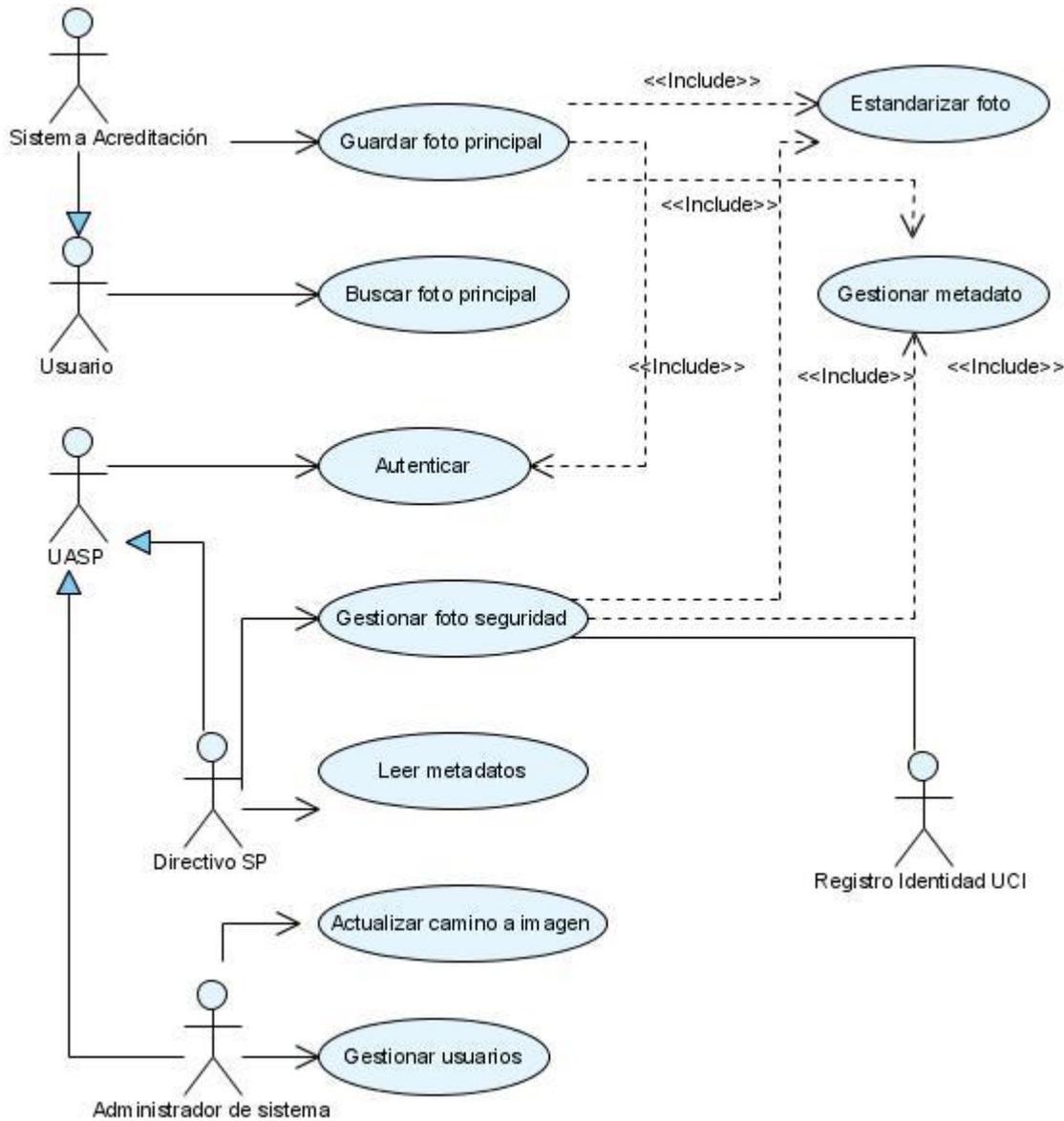


Figura 8: Diagrama de casos de usos del sistema

2.7.3 Descripción de los casos de usos del sistema

<b>Caso de Uso:</b>	<b>Estandarizar imagen</b>
<b>Actores:</b>	
<b>Resumen:</b>	El caso de uso se inicia cuando se solicita guardar una imagen principal o de seguridad, y cualquiera de estos casos de usos lo ejecutan. Se le da resolución, formato, tamaño a la imagen y se envía una respuesta en correspondencia.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<b>Precondiciones:</b>	Haberse solicitado guardar una foto.
<b>Referencias</b>	R3
<b>Prioridad</b>	Secundario.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1. Recibe la imagen para: <ul style="list-style-type: none"> <li>▪ Dar resolución a la imagen.</li> <li>• Dar formato a la imagen.</li> <li>• Dar tamaño y ancho a la imagen.</li> </ul>
	1. Se validan los datos. En caso contrario ver flujo alternativo 1.
	2. Se realiza la petición.
	3. Se envía respuesta.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Flujo alternativo 1	
	Envía una notificación de error al sistema y termina el CU.
<b>Poscondiciones</b>	La imagen se estandariza.

**Tabla 2. 3: Descripción del caso de uso Estandarizar imagen.**

<b>Caso de Uso:</b>	<b>Gestionar metadatos.</b>
<b>Actores:</b>	
<b>Resumen:</b>	El caso de uso se inicia cuando se solicita guardar una imagen principal o de seguridad y estos casos de usos lo ejecutan. Se envía los datos necesarios, se borran los datos EXIF a la imagen, se le introducen los datos enviados y se brinda una respuesta en correspondencia.
<b>Precondiciones:</b>	Haberse solicitado guardar una foto.
<b>Referencias</b>	R4.
<b>Prioridad</b>	Secundario.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<p>1. Recibe la foto y todos los datos. Los cuales pueden ser:</p> <ul style="list-style-type: none"> <li>• Ocupación.</li> <li>• Área (Opcional en dependencia de la ocupación).</li> <li>• Facultad (Opcional en dependencia de la ocupación).</li> <li>• Grupo (Opcional en dependencia de la ocupación).</li> <li>• CI.</li> <li>• Nombre.</li> <li>• Primer apellido.</li> <li>• Segundo apellido.</li> <li>• Usuario</li> <li>• Solapín.</li> </ul>
	<p>2. Verifica la exactitud de los datos. En caso contrario ver flujo alternativo 1.</p>
	<p>3. Verifica que la foto tiene datos EXIF. En caso contrario ver flujo alternativo 2.</p>
	<p>4. Elimina los datos EXIF de la foto.</p>
	<p>5. Clasifica jerarquía de datos establecidos.</p>
	<p>6. Inserta jerarquía de datos dentro del fichero de la foto y envía respuesta.</p>
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Flujo alternativo 1	
	Si no recibe todos los datos envía una notificación.
Flujo alternativo 2	
	Si la foto no posee datos EXIF, se pasa a la línea 5 y se sigue ejecutando el CU.
<b>Poscondiciones</b>	Se gestionan los metadatos en las fotos.

**Tabla 2. 4: Descripción del caso de uso Gestionar Metadatos.**

<b>Caso de Uso:</b>	<b>Buscar foto principal</b>	
<b>Actores:</b>	Usuario(inicia)	
<b>Resumen:</b>	El caso de uso se inicia cuando se solicita buscar una imagen. Se envía el solapín de la persona para realizar la búsqueda. El servicio solicitado es ejecutado y se devuelve la foto.	
<b>Precondiciones:</b>	La foto tiene que estar guardada.	
<b>Referencias</b>	R6.	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema.</b>	
1. El actor envía una petición de búsqueda de foto por solapín.	2. Recibe el criterio de búsqueda.	
	3. Valida que el dato recibido sea correcto. En caso contrario ver flujo alternativo 1.	
	4. Busca la foto más actual. En caso contrario ver flujo alternativo 2.	
	5. Envía respuesta.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
Flujo alternativo 1		
	Envía una notificación de error al usuario.	
Flujo alternativo 2		
	Si no encontró la foto envía una notificación de no encontrada al usuario.	
<b>Poscondiciones</b>	La foto es buscada.	

**Tabla 2. 5: Descripción del caso de uso Buscar foto principal.**

<b>Caso de Uso:</b>	<b>Guardar foto principal</b>
<b>Actores:</b>	Sistema Acreditación (inicia)

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<b>Resumen:</b>	El caso de uso se inicia cuando se solicita guardar una imagen principal. Se envían los datos necesarios para el procesamiento de los servicios a través de la aplicación y para poder solicitar el servicio. Si los datos son correctos, el servicio solicitado es ejecutado y se brinda una respuesta en correspondencia.
<b>Precondiciones:</b>	El sistema puede solicitar ese servicio.
<b>Referencias</b>	R5, R6
<b>Prioridad</b>	Crítico

### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. Envía la solicitud de guardar la foto, envía los datos de autenticación.	2. Recibe la foto y todos los datos.
	3. Valida el dato ocupación. <ul style="list-style-type: none"> <li>a. Si la ocupación es de tipo estudiantes. Véase sección “Validar Estudiantes”.</li> <li>b. Si la ocupación es de tipo trabajadores. Véase sección “Validar Trabajadores”.</li> </ul>

### Flujo Normal de Eventos

#### Sección “Validar Estudiantes”

Acción del Actor	Respuesta del Sistema
	1. Si la ocupación es de tipo Estudiante, el servicio aparte de la foto debe recibir los siguientes datos en ese orden. <ul style="list-style-type: none"> <li>• Usuario.</li> <li>• Contraseña.</li> <li>• Facultad.</li> <li>• Grupo.</li> <li>• CI.</li> <li>• Nombre.</li> <li>• Primer apellido.</li> <li>• Segundo apellido.</li> <li>• Provincia.</li> <li>• Municipio.</li> <li>• Usuario.</li> </ul>
	2. Verifica la exactitud de los datos. En caso contrario

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	ver Flujo alternativo 1.
	3. Ejecuta CU <sup>21</sup> Autenticar usuario.
	4. Ejecuta CU Gestionar metadatos.
	5. Ejecuta CU Estandarizar imagen.
	6. Crea la dirección de la foto atendiendo a los dos primeros datos recibidos.
	7. Guarda los datos, la dirección de la foto y la fecha actual.
	8. Verifica existencia de directorio atendiendo a los dos primeros datos recibidos. En caso contrario ver flujo alternativo 2.
	9. Realiza la petición del sistema y envía respuesta.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Flujo alternativo 1.	
	Si no recibe todos los datos envía una notificación de error al sistema.
Flujo alternativo 2.	
	Si no está creado el directorio, se crea y se guarda la foto.
<b>Flujo Normal de Eventos</b>	
<b>Sección "Validar Trabajadores"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>1. Si la ocupación es de tipo Trabajadores, el servicio aparte de la foto debe recibir los siguientes datos en ese orden.</p> <ul style="list-style-type: none"> <li>• Área.</li> <li>• Cl.</li> <li>• Nombre.</li> </ul>

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<ul style="list-style-type: none"> <li>• Primer apellido.</li> <li>• Segundo apellido.</li> <li>• Provincia.</li> <li>• Municipio.</li> <li>• Usuario.</li> <li>• Solapín</li> </ul>
	2. Verifica la exactitud de los datos. En caso contrario ver flujo alternativo 1.
	3. Ejecuta CU Autenticar usuario.
	4. Ejecuta CU Gestionar metadatos.
	5. Ejecuta CU Estandarizar imagen.
	6. Crea la dirección de la foto atendiendo al primer dato recibido.
	7. Guarda los datos, la dirección de la foto y la fecha actual en la base de datos.
	8. Verifica existencia de directorio atendiendo al primer dato recibido. En caso contrario ver flujo alternativo 2.
	9. Realiza la petición del sistema y envía respuesta.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Flujo alternativo 1.	
	Si no recibe todos los datos envía una notificación de error al sistema.
Flujo alternativo 2.	
	Si no está creado, el directorio se crea y se guarda la foto.
<b>Poscondiciones</b>	La foto es guardada.

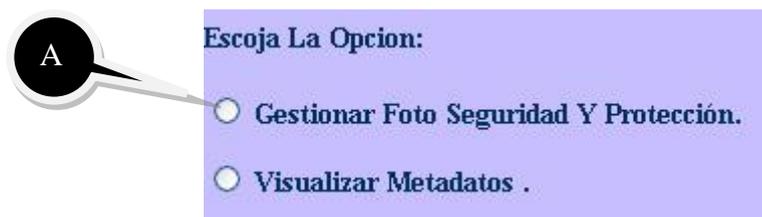
**Tabla 2. 6: Descripción del caso de uso Guardar Foto principal.**

<b>Caso de Uso:</b>	<b>Gestionar foto seguridad</b>
<b>Actores:</b>	Directivo SP (inicia).
<b>Resumen:</b>	El caso de uso se inicia cuando el directivo de seguridad y protección solicita la opción de guardar, eliminar o buscar las fotos de una persona. Envía los datos necesarios para el procesamiento de la aplicación, y se brinda una respuesta en correspondencia.
<b>Precondiciones:</b>	El usuario debe estar autenticado previamente.
<b>Referencias</b>	R5, R7, R8.
<b>Prioridad</b>	Crítico

**Flujo Normal de Eventos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción de gestionar las fotos de seguridad y protección (A).	2. Muestra una interfaz para que el administrador escoja la acción que desea realizar
1. Selecciona. <ul style="list-style-type: none"> <li>• Guardar foto.</li> <li>• Eliminar foto.</li> <li>• Buscar foto.</li> </ul>	2. Si el administrador selecciona: <ul style="list-style-type: none"> <li>a. Guardar foto. Véase sección “Guardar foto”.</li> <li>b. Eliminar foto. Véase sección “Eliminar foto”.</li> <li>c. Buscar foto. Véase sección “Buscar Foto”.</li> </ul>

**Prototipo de Interfaz**



**Flujo Normal de Eventos**

**Sección “Guardar Foto”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario escoge la opción de guardar la foto del estudiante (C). En caso contrario véase flujo alternativo 1.	2. El sistema muestra una interfaz para guardar la foto del estudiante
3. El actor deberá escoger el tipo de foto que desea guardar (D), el nombre (E), el 1er apellido (F), el 2do apellido (G), el CI (H), el	

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

usuario (I), buscar por provincia (J), municipio (K), Fecha (L), facultad (LL), grupo (M), hacer una descripción de la foto(X), y subir la foto a guardar (W)	
4. Envía los datos (Y). En caso contrario véase flujo alterno 2.	5. Valida la entrada de los datos. En caso contrario véase flujo alterno 3.
	6. Ejecuta CU Estandarizar imagen.
	7. Ejecuta CU Gestionar metadatos.
	8. Crea la dirección de la foto atendiendo a los dos primeros datos recibidos.
	9. Guarda los datos y la dirección de la foto.
	10. Verifica existencia de directorio atendiendo a los dos primeros datos recibidos. En caso contrario ver flujo alterno 4.
	11. Guarda la foto y envía respuesta.

**Prototipo de Interfaz**

**Que Desea Guardar**

Estudiante

**Escoger Tipo De Foto** [v]

**Nombre** [ ]

**1er Apellido** [ ]

**2do Apellido** [ ]

**CI** [ ]

**Usuario** [ ]

**Provincia** [ ]

**Municipio** [ ]

**Fecha** [ ]

**Facultad** [ ]

**Grupo** [ ]

**Subir La Foto**

[ (x) ]

[ Examinar ]

**Descripcion De La Foto**

[ ]

[ Enviar ] [ cancelar ]

Callouts: C, D, E, F, G, H, I, J, K, L, M, W, X, Y, Z

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Flujo alternativo 1	
1. El usuario escoge la opción de guardar la foto del trabajador (N).	2. El sistema muestra una interfaz para guardar la foto del trabajador
3. El actor deberá escoger el tipo de foto que desea guardar (Ñ), el nombre (O), el 1er apellido (P), el 2do apellido (Q), el CI (R), el usuario (S), buscar por provincia (T), municipio (U), área (V), fecha (A), hacer una descripción de la foto(X), y subir la foto a guardar (W). Va a la línea 4 y el caso de uso termina en la línea 12	
Flujo alternativo 2.	
El actor escoge la opción de cancelar(Z), y se termina el caso de uso	
Flujo alternativo 3	
	Envía una nota de error y se termina el caso de uso

Flujo alternativo 4	
	El sistema crea el directorio.
<b>Sección “Eliminar Foto”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. Invoca CU Buscar Foto.
	2. Muestra una interfaz y visualiza el resultado del CU Buscar Foto.
3. El usuario escoge la opción de eliminar (1A).	4. Verifica que exista directorio de expedidos. En caso contrario ver flujo alternativo 1
	5. Guarda la o las fotos seleccionadas en el directorio de expedidos.
	6. El sistema actualiza los datos
	7. El sistema elimina la o las fotos del directorio y envía notificación.
<b>Prototipo de Interfaz</b>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Flujo alternativo 1	
	Crea el directorio y vuelve a la línea 5
<b>Sección “Buscar Foto”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El actor deberá escoger el tipo de foto que desea buscar (B), y el nombre (C), o el 1er apellido (D), o el 2do apellido (E), o el CI (F), o el usuario (G), o buscar por provincia (H), municipio (I), fecha (J), facultad (K), grupo (L) o área (M)	

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2. Envía los datos (N). En caso contrario ver flujo alternativo 1	3. Valida la entrada de los datos. En caso contrario ver flujo alternativo 2.
	4. Pide información al Registro Identidad UCI.
	5. Busca la foto y la visualiza.

### Prototipo de Interfaz

### Flujos Alternos

Acción del Actor	Respuesta del Sistema
Flujo alternativo 1.	
El actor escoge la opción de cancelar(O), y se termina el caso de uso	
Flujo alternativo 2.	
	No se envían los datos y termina el caso de uso(X).

**Tabla 2. 7: Descripción del caso de uso Gestionar foto seguridad**

<b>Caso de Uso:</b>	<b>Leer metadatos</b>
<b>Actores:</b>	Directivo SP (inicia).

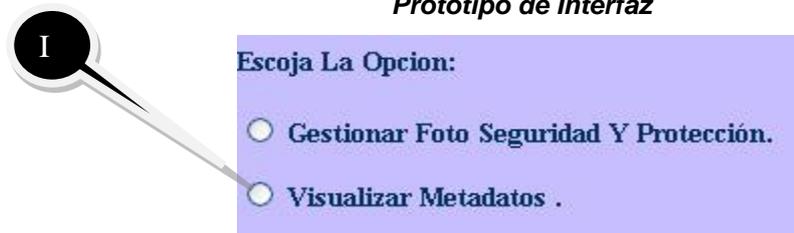
## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<b>Resumen:</b>	El CU comienza cuando el directivo de seguridad y protección solicita el servicio que devuelve los metadatos de una foto. Para que el servicio devuelva resultados satisfactorios deberá enviar el CI para realizar la búsqueda de la foto en el directorio o enviar la foto.
<b>Precondiciones:</b>	El usuario debe estar autenticado previamente.
<b>Referencias</b>	R4
<b>Prioridad</b>	Crítico.

### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El actor solicita el servicio.	2. El sistema para realizar la acción solicita datos obligatorios que pueden ser: <ul style="list-style-type: none"> <li>• CI.</li> <li>• Foto.</li> </ul>
3. El actor envía los datos. <ol style="list-style-type: none"> <li>a. Envía CI. Véase sección "Envía CI".</li> <li>b. Envía Foto. Véase sección "Envía Foto".</li> </ol>	

### Prototipo de Interfaz



**Sección "Envía CI"**

Acción del Actor	Respuesta del Sistema
1. El actor entra el CI (C) y escoge la opción de enviar los datos (E). En caso contrario ver flujos alternos 1.	2. Valida los datos. En caso contrario ver flujos alternos 2.
	3. Busca la dirección de la foto correspondiente al CI. En caso contrario ver flujo alternativo 3.
	4. Realiza la petición del sistema y envía respuesta.
	5. Visualiza la información.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Flujo alternativo 1	
El actor escoge la opción de cancelar (F) y se termina el caso de uso.	
Flujo alternativo 2	

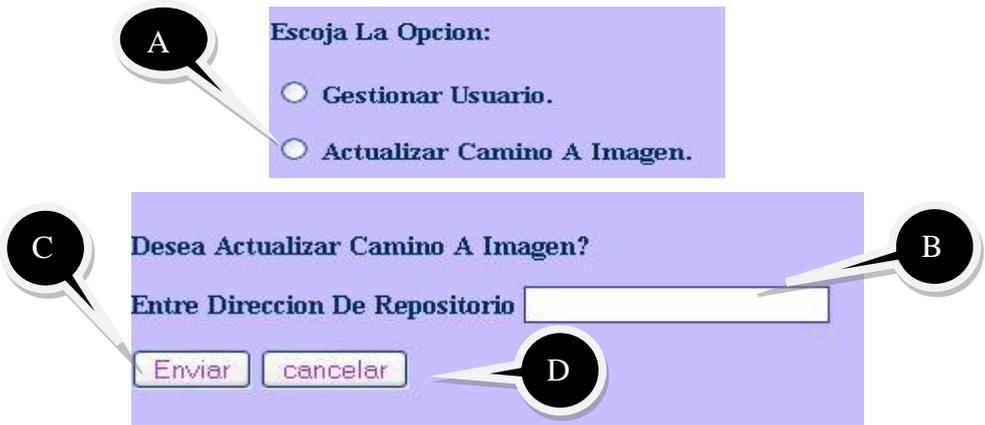
## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	Si no recibe los datos exactos envía una notificación de error al actor y se termina el caso de uso.
Flujo alternativo 3	
	Envía una notificación de no encontrado y se termina el caso de uso.
<b>Sección "Envía Foto"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Entra la foto (D) y escoge la opción de enviar los datos (E). En caso contrario ver Flujos Alternos1.	2. Valida que la foto sea de tipo JPGE. En caso contrario ver flujo alternativo1.
	3. Verifica si puede leer datos no visibles en la foto. En caso contrario ver flujo alternativo 2.
	4. Realiza la petición del sistema y envía respuesta.
	5. Visualiza la información.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Flujo alternativo 1	
El actor escoge la opción de cancelar (F) y se termina el caso de uso.	
Flujo alternativo 2.	
	Si la foto no tiene formato JPGE envía una notificación de error al actor.
Flujo alternativo 2	
	Si no puede leer los datos no visibles en las fotos envía una notificación de error.
<b>Poscondiciones</b>	Se leen los datos no visibles a las fotos.

**Tabla 2. 8: Descripción del caso de uso Leer metadatos.**

<b>Caso de Uso:</b>	<b>Actualizar camino a imagen.</b>
<b>Actores:</b>	Administrador de sistema (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando el actor solicita la opción de actualizar el camino a la imagen. El sistema lee uno por uno los metadatos de cada foto y va actualizando las direcciones de las fotos.
<b>Precondiciones:</b>	El usuario debe estar autenticado previamente.
<b>Referencias</b>	R11.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El actor escoge la opción de actualizar el camino de la imagen (A).	2. Muestra una interfaz para que el usuario confirme su decisión e inserte la dirección del directorio (B).	
3. El usuario envía los datos. En caso contrario ver flujo alternativo 1.	4. Busca en el directorio cada foto y lee los metadatos.	
	5. Con los datos no visibles que lee crea una nueva dirección de la foto.	
	6. Actualiza el camino hacia la imagen	
<b>Prototipo de Interfaz</b>		
		
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
Flujo alternativo 1		
El usuario escoge la opción de cancelar (D) y termina el caso de uso		
<b>Poscondiciones</b>	Se actualizan las direcciones físicas de las fotos.	

**Tabla 2. 9: Descripción del caso de uso Actualizar camino a imagen.**

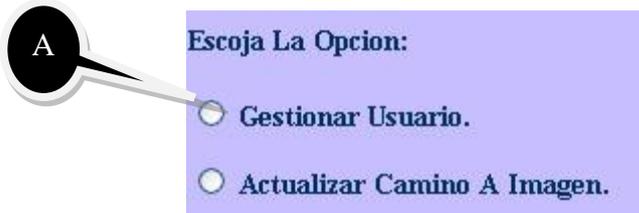
<b>Caso de Uso:</b>	<b>Autenticar usuario</b>
<b>Actores:</b>	UASP(inicia)
<b>Resumen:</b>	El caso de uso comienza cuando un usuario necesita autenticarse para realizar cualquier función que le brinda el sistema.
<b>Precondiciones:</b>	Se realice la solicitud de cualquier funcionalidad
<b>Referencias</b>	R1

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El actor solicita algún servicio del sistema.	2. Pide al actor que se autentique. Pide datos obligatorios. <ul style="list-style-type: none"> <li>• Usuario(A)</li> <li>• Contraseña(B)</li> </ul>	
3. El actor envía los datos(C). En caso contrario ver flujo alterno 1.	4. Verifica la validez de los datos para autenticar al usuario. En caso contrario ver flujo alterno 2.	
	5. Busca que privilegios tiene es usuario. En caso contrario ver flujo alterno 3.	
	6. Autentica el usuario.	
	7. Muestra la interfaz a la que tiene acceso.	
<b>Prototipo de Interfaz</b>		
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>	
Flujo alterno 1		
El actor escoge la opción de cancelar (D) y se termina el caso de uso.		
Flujo alterno 2		
	Si los datos enviados no son validos se muestra un mensaje de error.	
Flujo alterno 3		
	Si el actor está tratando de ejecutar algún servicio al que no tiene acceso se muestra un mensaje de error y se termina el caso de uso.	
<b>Poscondiciones</b>	Se autentica el actor y se muestra la interfaz a la que tiene acceso.	

**Tabla 2. 10: Descripción del caso de uso Autenticar usuario.**

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<b>Caso de Uso:</b>	<b>Gestionar usuario</b>	
<b>Actores:</b>	Administrador de sistema (inicia)	
<b>Resumen:</b>	El caso de uso comienza cuando el administrador de sistema necesita, insertar un nuevo usuario, eliminar un usuario, asignarle roles a un usuario o eliminarle un rol. El administrador pasa los datos necesario y se ejecuta la acción solicitada.	
<b>Precondiciones:</b>	El usuario tiene que haberse autenticado	
<b>Referencias</b>	R10.	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona la opción de gestionar usuario(A).	2. Muestra una interfaz para que el administrador escoja la acción que desea realizar	
3. Selecciona. <ul style="list-style-type: none"> <li>• Eliminar usuario.</li> <li>• Insertar usuario.</li> <li>• Asignar roles.</li> <li>• Eliminar roles.</li> </ul>	4. Si el administrador selecciona: <ol style="list-style-type: none"> <li>a. Eliminar usuario. Véase sección “Eliminar usuario IP”.</li> <li>b. Insertar usuario. Véase sección “Insertar usuario”.</li> <li>c. Asignar roles. Véase sección “Asignar roles”.</li> <li>d. Eliminar roles. Véase sección “Eliminar Roles”.</li> </ol>	
<b>Prototipo de Interfaz</b>		
		
<b>Sección “Eliminar Usuario”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El actor entra el usuario (A) y envía los datos (B). En caso contrario ver flujo alternativo 1.	2. Busca usuario. En caso contrario ver flujo alternativo 2.	
	3. Pregunta si está seguro de eliminar el usuario.	
4. El actor confirma su decisión. En caso contrario ver flujo alternativo 3.	5. Elimina el usuario.	

6. Envía notificación.	
<b>Prototipo de Interfaz</b>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Flujo alternativo 1	
El actor escoge la opción de cancelar(D) y se termina el caso de uso	
Flujo alternativo 2	
	Si el sistema no encuentra al usuario envía una notificación de no encontrado y se termina el CU.
Flujo alternativo 3.	
	Si el actor cancela su decisión se termina el caso de uso.
<b>Sección "Insertar Usuario"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El actor entra el rol (A) y el usuario (B) y envía los datos (D).	2. Valida la entrada de los datos. En caso contrario ver flujo alternativo1.
	3. Busca si no existe el usuario. En caso contrario ver flujo alternativo2.
	4. Inserta el usuario y el rol que tendrá.
	5. Envía notificación.
<b>Prototipo de Interfaz</b>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Flujo alternativo 1	
El actor escoge la opción de cancelar(D) y se termina el caso de uso	

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Flujo alternativo 2	
	Si los datos no son correctos no se envían los datos y se envía una notificación de error
Flujo alternativo 2	
	Si el usuario ya existe se le informa al actor y se termina el CU
Sección "Asignar roles"	
Acción del Actor	Respuesta del Sistema
1. El actor entra el nuevo rol (A) y el usuario (B) y escoge la opción enviar(C). En caso contrario ver flujo alternativo 1.	2. El sistema valida la entrada de los datos. En caso contrario ver flujo alternativo 2.
	3. El sistema busca el usuario. En caso contrario ver flujo alternativo 3.
	4. El sistema inserta el nuevo rol.
	5. El sistema envía notificación.
<p><b>Prototipo de Interfaz</b></p>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Flujo alternativo 1	
El actor escoge la opción de cancelar(D) y se termina el caso de uso	
Flujo alternativo 2	
	Si los datos no son correctos no se envían los datos y se envía una notificación de error
Flujo alternativo 3	
	Si no encuentra al usuario envía una notificación de no encontrado y se termina el CU.
Sección "Eliminar Roles"	
Acción del Actor	Respuesta del Sistema

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

1. El actor entra el rol (A) y el usuario (B) que desea eliminar y escoge la opción de enviar (C). En caso contrario ver flujo alternativo 1.	2. El sistema valida la entrada de los datos. En caso contrario ver flujo alternativo2.
	3. Verifica si no existe el usuario. En caso contrario ver flujo alternativo 3.
	4. Busca si el usuario posee el rol. En caso contrario ver flujo alternativo 4.
	5. Elimina el rol.
	6. Envía notificación.
<p><b>Prototipo de Interfaz</b></p>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Flujo alternativo 1	
El actor escoge la opción de cancelar(D) y se termina el caso de uso	
Flujo alternativo 2	
	Si los datos no son correctos no se envían los datos y se envía una notificación de error
Flujo alternativo 3	
	Si el usuario no existe se le informa al actor y se termina el CU
Flujo alternativo 4	
	Si el rol no existe se le informa al actor y se termina el CU
Prototipo de Interfaz	
<b>Poscondiciones</b>	Se gestionan los usuarios y los roles que tendrán dentro de la aplicación.

**Tabla 2. 11: Descripción del caso de uso Gestionar Usuario.**

### 2.8 Estudio de la factibilidad

Antes de comenzar a realizar un software lo primero que se debe tener en cuenta es ver si es viable llevar a cabo su realización o no y la estimación de costes, tiempos y recursos requeridos, pues la estimación establece planes razonables para desarrollar la Ingeniería de Software y manejar los

cambios de los proyectos de Software. Para realizar estas estimaciones la Ingeniería de Software ofrece un grupo de herramientas y técnicas entre las que se incluye COCOMO<sup>22</sup>, y la estimación mediante el análisis de Puntos de Casos de Uso. Para realizar el estudio de la factibilidad se utilizará este último método.

**Método de estimación por puntos de caso de uso**

La estimación mediante el análisis de puntos de casos de uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

A continuación, se detallan los pasos a seguir para la aplicación de éste método.

- El primer paso consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor, se calcula a partir de la siguiente ecuación:

**UUCP = UAW + UUCW** donde,

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

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

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

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

Tipo de actor	Descripción	Factor de Peso	Actores	Total
Simple	Otro sistema que interactúa con la aplicación a desarrollar mediante una interfaz de programación.	1	2	2
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	3	9

$$UAW = \sum CantidadActores_i * Peso_i$$

**Tabla 2. 12: Factor de Peso de los Actores sin ajustar.**

**UAW = 11**

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

Tipo de CU	Descripción	Factor de Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	7	35
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	2	20
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
$UUCW = \sum CantidadCU_i * Peso_i$				

**Tabla 2. 13: Factor de Peso de los Casos de Uso sin ajustar.**

**UUCW = 55** Finalmente los Puntos de Casos de Uso sin ajustar resultan,

**UUCP = 55 + 11 = 66.**

- El segundo paso consiste en el cálculo de Puntos de Casos de Uso ajustados. Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

**UCP = UUCP x TCF x EF donde,**

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

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

**TCF:** Factor de complejidad técnica.

**EF: Factor de ambiente.**

Este cálculo se realiza mediante el cómputo de un conjunto de valores de complejidad asignada a los diferentes factores con una escala de 0 a 5.

**Significado de los valores**

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.

3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

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

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Factor	Descripción	Peso	Valor Asignado	Total
T1	Sistema distribuido.	2	4	8
T2	Tiempo de respuesta.	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Funcionamiento Interno complejo	1	2	2
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	5	5
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	3	3
	Se requieren facilidades especiales de entrenamientos a usuarios.	1	1	1
Sumatoria				51
$TCF = 0.6 + 0.01 * \sum (Peso_i * ValorAsignado_i)$				

**Tabla 2. 14: Factores de Peso de Complejidad Técnica.**

$$TCF = 0.6 + 0.01 * 51 = 1.11$$

### Factor Ambiente (EF)

Factor	Descripción	Peso	Valor Asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	2	3
E2	Experiencia en la aplicación	0.5	2	1
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder.	0.5	3	1,5
E5	Motivación	1	4	4
E6	Estabilidad de los requerimientos.	2	5	10

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

E7	Personal Part-Time.	-1	5	-5
E8	Dificultad del lenguaje de programación	-1	2	-2
Sumatoria				16.5
$EF = 1.4 - 0.03 * \sum (Peso_i * ValorAsignado_i)$				

**Tabla 2. 15: Factores de Peso del Factor de Ambiente.**

$$EF = 1.4 - 0.03 * 16.5 = 1.4 - 0.495 = 0.905$$

Finalmente, los puntos de casos de usos ajustados resultan:

$$UCP = 66 * 1.11 * 0.905 = 66.3003$$

- El tercer paso consiste en calcular el esfuerzo del flujo de trabajo implementación mediante la siguiente ecuación:

$$E = UCP * CF \text{ donde,}$$

**E:** Esfuerzo estimado en horas-hombre.

**UCP:** Puntos de casos de usos ajustados.

Para calcular **CF**:

**CF = 20 horas-hombre** (si Total  $EF \leq 2$ ).

**CF = 28 horas-hombre** (si Total  $EF = 3$  ó Total  $EF = 4$ ).

**CF = abandonar o cambiar proyecto** (si Total  $EF \geq 5$ ).

**Total  $EF = \text{Cant } EF < 3 \text{ (entre E1 -E6)} + \text{Cant } EF > 3 \text{ (entre E7- E8)}$**  luego,

**CF = 28 horas-hombre**

Finalmente el esfuerzo del flujo de trabajo implementación resulta:

$$E = 66.3003 * 28 = 1856,4084.$$

Actividad	Porcentaje	Horas/Hombre
Análisis	10.00%	464.1021
Diseño	20.00%	928.2042
Implementación	40.00%	1856.4084
Pruebas	15.00%	696,15315

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Sobrecarga (otras actividades)	15.00%	696,15315
Total	100%	4641.021

Debido a que el desarrollo de la investigación comprende las fases de análisis y diseño, el esfuerzo en estas etapas sería:

**ET = 1392.3063 horas-hombre** entonces,

Estimando que:

Cada mes tiene 4 semanas, cada semana 5 días laborables, y cada día 10 horas laborables nos quedaría como promedio 200 horas laborables entonces,

**ET = 6.9 Mes/Hombre** luego,

Si, **Tiempo = ET / CH** donde,

**CH:** cantidad de hombres,

**Tiempo = 6.9 / 1**

**Tiempo = 7**

Esto quiere decir que con 1 hombre trabajando en el proyecto el mismo se desarrollará en aproximadamente 7 meses.

Costo del Proyecto.

Se asume como salario promedio mensual \$100.00

Tiempo: Tiempo total del proyecto.

**CHM = CH \* SalarioPromedio**

**CHM = 100 \$/mes** luego,

**Costo = CHM \* ET / CH** entonces,

**Costo = \$690**

De los resultados obtenidos se interpreta que con un hombre trabajando en el proyecto el mismo se desarrollará en 7 meses y su costo total se estima que sea \$690.

### 2.9 Conclusiones

Como se ha visto, en este capítulo, se comienza a profundizar en el desarrollo de la propuesta de solución. Primeramente la elaboración del modelo del dominio o modelo conceptual permite un mejor estudio del entorno en el que se desarrollará el sistema y una mejor comprensión del problema que el software ya terminado tiene que resolver. Luego en el flujo de captura de requerimientos se obtiene

una lista de los requisitos con los que debe cumplir el sistema. El modelado del sistema muestra gráfica y detalladamente las funcionalidades que brindará el mismo.

Se describe un estudio realizado de la factibilidad del software, basados en la evaluación por punto de caso de uso, teniéndose en cuenta los factores que influyen en el desarrollo del Software. Se ofrecen datos de sumo interés como los costos del proyecto y tiempo de producción especificado por hombres.

### CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

#### 3.1 Introducción

Con el presente capítulo se obtendrán el Modelo de Análisis y Diseño del sistema que se está modelando. Se traducirán los requisitos a una especificación que describe cómo implementar el sistema. Pasando del análisis que consiste en obtener una visión del sistema ya que solo se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales, al diseño que es un refinamiento del análisis, que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos. El diseño será suficiente para que el sistema pueda ser implementado sin ambigüedades. Toda la documentación generada como parte del proceso de diseño está anexada y referenciada en el cuerpo del documento.

#### 3.2 Modelo de análisis

El Modelo de Análisis servirá para lograr un mayor entendimiento del dominio del problema descrito en el capítulo anterior por un Modelo Conceptual que se elaboró. También será indispensable a su vez como punto de partida para realizar el diseño del sistema. En el modelo de análisis no se toman en cuenta el lenguaje de programación que se va a utilizar en la construcción ni otros aspectos como la plataforma o los componentes reutilizables de otros sistemas, pues su objetivo principal es comprender, preparar, modificar y en general mantener exactamente los requisitos del software y no precisar cómo se realizará su implementación. Además existe una trazabilidad directa entre esas distintas estructuras, la cual se define entre casos de uso del modelo de casos de uso y realizaciones del caso de uso en el modelo de análisis. Se considera que este modelo es la primera aproximación al Modelo de Diseño y se describe en un lenguaje comprensible para los desarrolladores; se compone además por clases del análisis y sus objetos organizados en paquetes que colaboran entre sí.

##### 3.2.1 Diagramas de clases del análisis

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos del dominio de un problema y que representa las cosas del mundo real, no de la implementación automatizada. Está compuesto por clases del análisis y sus relaciones. Las clases del análisis están centradas en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos importantes y relaciones del dominio. Estas clases tienen atributos y existen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos entre ellas. Estas clases se clasifican en:

- Entidad: Modelan información que posee larga vida y que es a menudo persistente.

- Interfaz: Modelan la interacción entre el sistema y sus actores.
- Control: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

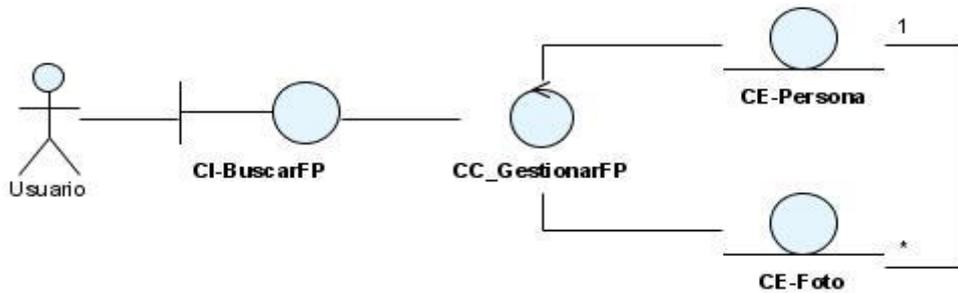


Figura 9: Diagrama de Análisis del CU Buscar foto principal

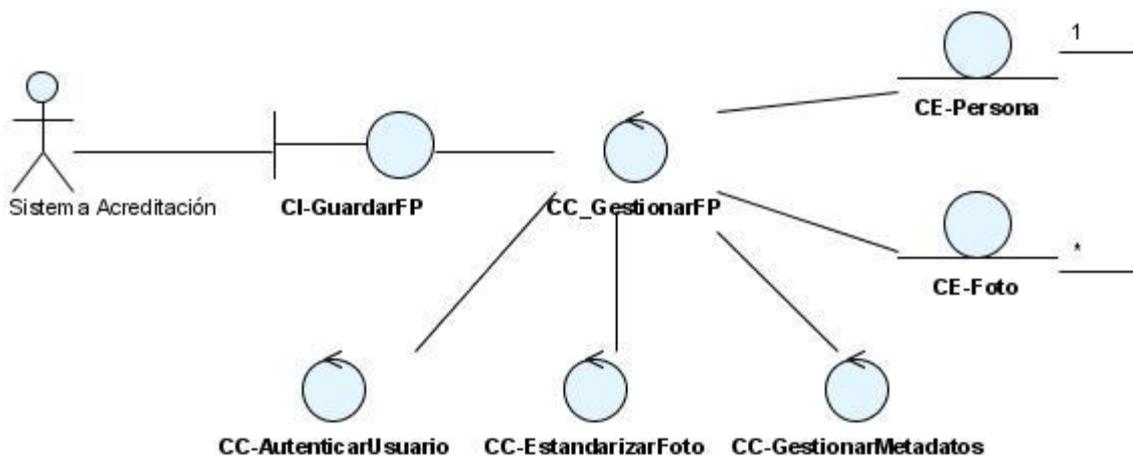


Figura 10: Diagrama de Análisis del CU Guardar foto principal

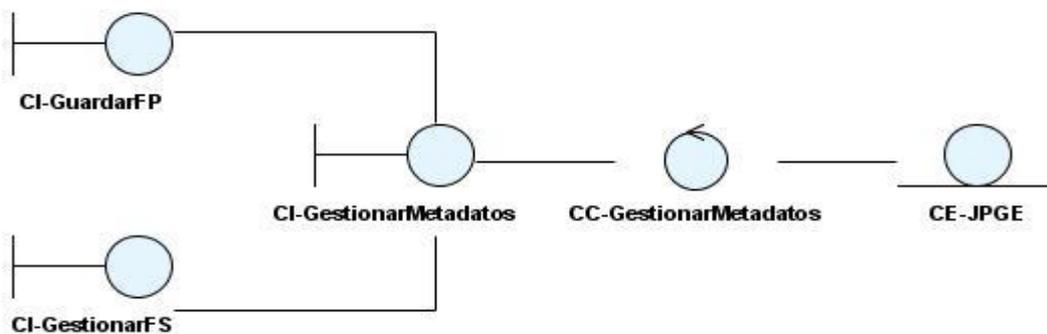


Figura 11: Diagrama de Análisis del CU Gestionar metadatos.

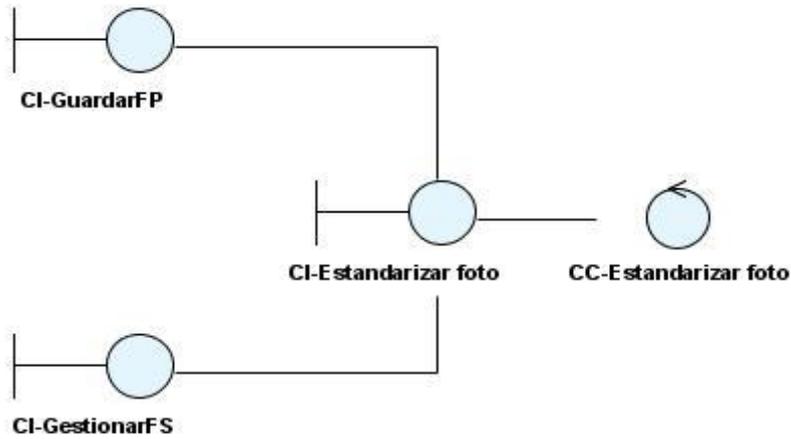


Figura 12: Diagrama de Análisis del CU Estandarizar Foto.

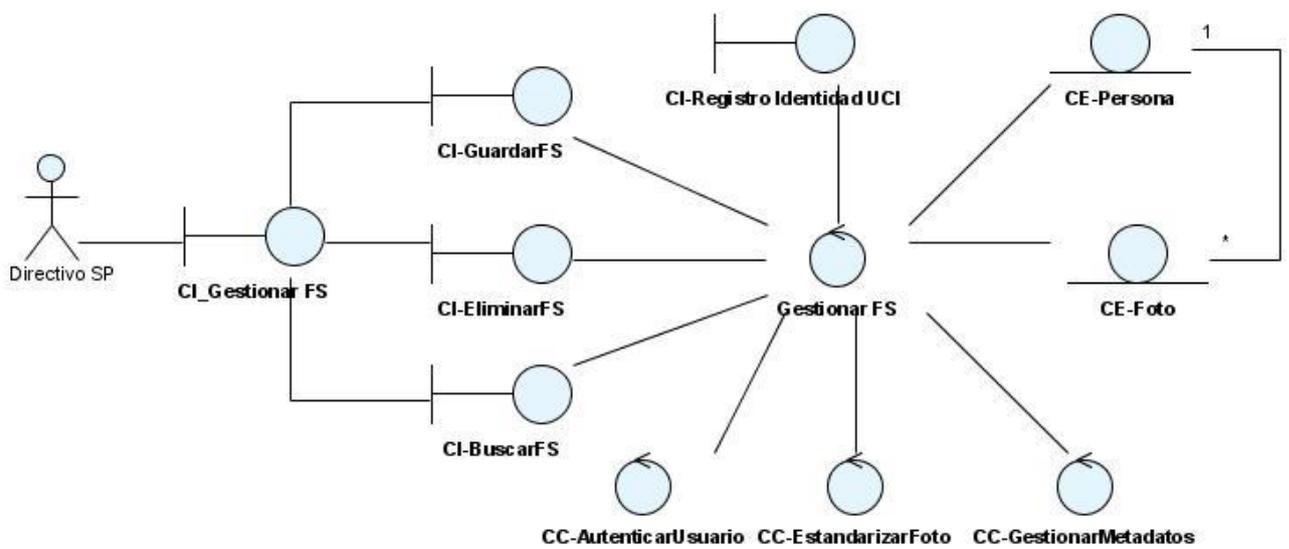


Figura 13: Diagrama de Análisis del CU Gestionar foto seguridad.

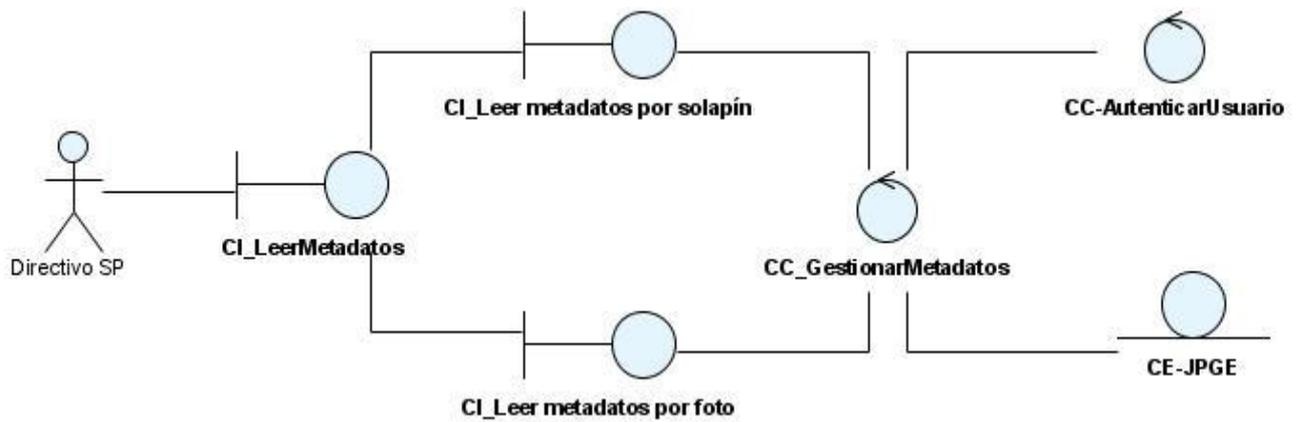


Figura 14: Diagrama de Análisis del CU Leer metadato.

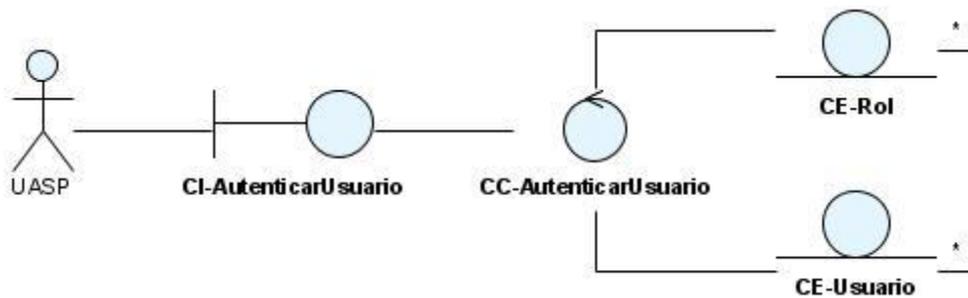


Figura 15: Diagrama de Análisis del CU Autenticar usuario.

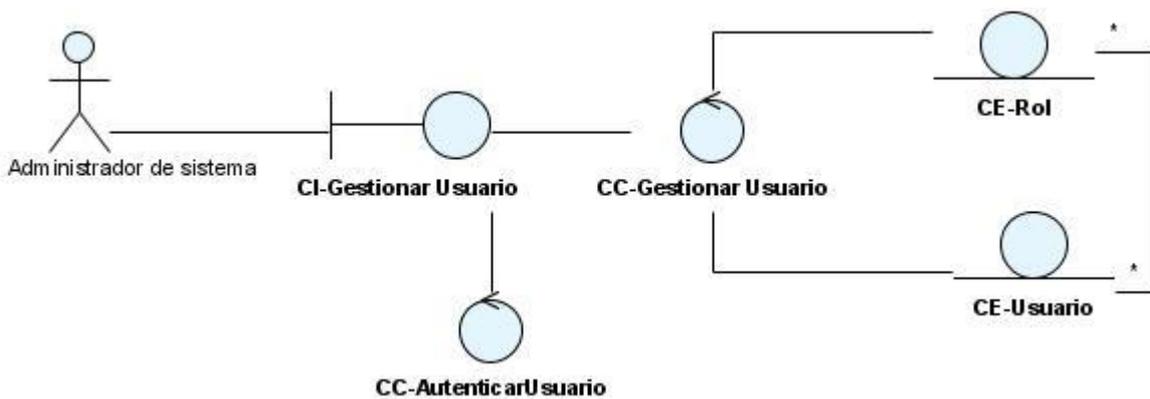


Figura 16: Diagrama de Análisis del CU Gestionar usuario.

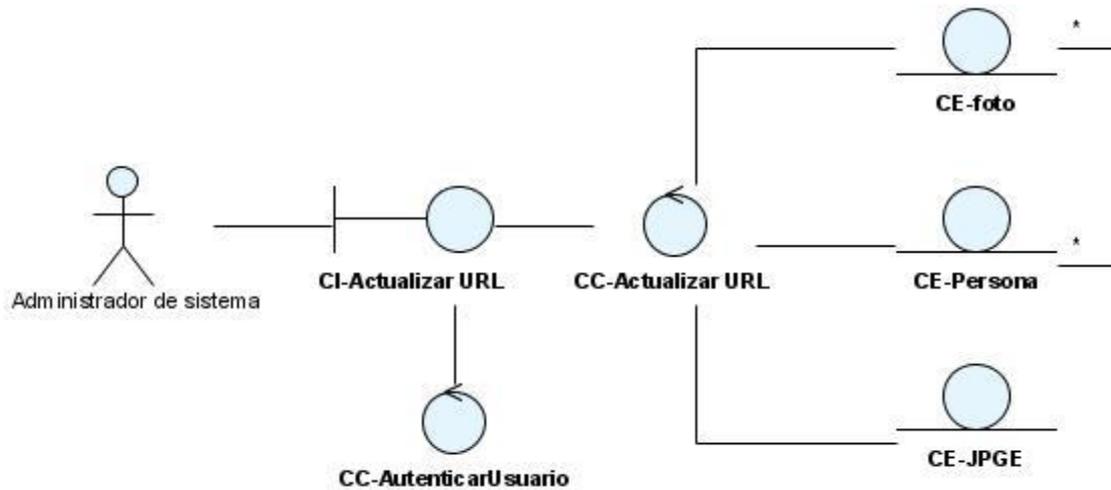


Figura 3.9: Diagrama de Análisis del CU Actualizar camino a imagen.

### 3.3 Modelo de diseño

Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis que proporciona una comprensión detallada de los requisitos. En el diseño se modela el sistema y se encuentra su forma, incluyendo la arquitectura, para que soporte todos los requisitos funcionales, no funcionales y las restricciones que se le suponen. El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de usos. Es específico para una implementación y depende del lenguaje de programación. La esencia del Diseño es la elaboración de los diagramas de interacción, que describen cómo los objetos se comunican entre ellos para dar cumplimiento a los requerimientos que fueron definidos y permiten la realización de los diagramas de clases del diseño, que contienen las clases que se pueden implementar en un software.

Para la realización del trabajo, se creó un diseño flexible basado en patrones de diseño para promover la flexibilidad y posibilitar la facilidad de extensión y en patrones de arquitectura para separar la aplicación en capas lógicas, facilitando el desarrollo del software de manera modulada y paralela.

Es válido destacar que los patrones tanto de diseño como de arquitectura utilizados en la realización del diseño del trabajo fueron definidos por los arquitectos del proyecto Control de Acceso, al igual que la decisión de soportar toda la aplicación sobre el marco de trabajo arquitectónico Symfony, por lo que el diseño realizado se adapta a las particularidades de este framework.

### 3.4 Patrones de diseño

Los patrones de diseño son búsquedas de soluciones a problemas comunes en el desarrollo de software. Un patrón es una solución efectiva de un problema, que lo ha resuelto en otras ocasiones y que puede ser utilizado de nuevo en otras circunstancias. Para la realización del diseño del sistema se utilizaron los patrones GRASP.

#### Patrones GRASP

Es el acrónimo de General Responsibility Assignment Software Patterns. Aunque más que patrones se les considera prácticas recomendables a la hora de diseñar el software.

#### Bajo Acoplamiento

El acoplamiento hace referencia a las relaciones que tienen objetos entre si dentro de un sistema. Cuando una serie de objetos tienen relación con varios objetos, si se cambian estos últimos, los objetos relacionados han de verse afectados. Al utilizar bajo acoplamiento se beneficia al programador pues cada objeto tiene la mínima dependencia con el resto del sistema, de esta forma se pueden hacer modificaciones en partes del programa sin necesidad de cambiar el resto del sistema

#### Alta cohesión

Cuando se dice que una clase tiene alta cohesión se quiere decir que el objeto tiene bien delimitadas sus responsabilidades. La cohesión de un objeto significa cuan relacionadas y enfocadas están las acciones del objeto. Cada elemento de nuestro diseño realiza una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable.

#### Experto

La responsabilidad de realizar una labor, es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Hay que tener en cuenta que esto es aplicable mientras se esté considerando los mismos aspectos del sistema:

- Lógica de negocio.
- Persistencia a la base de datos.
- Interfaz de usuario.

No tiene sentido considerar que una clase se debe escribir a sí misma en base de datos o formatearse para presentarse en una página HTML por el hecho de poseer los datos. Estos son elementos estructuralmente distintos y deben considerarse desde una perspectiva distinta.

### **Creador**

Se asigna la responsabilidad de que una clase B cree un objeto de la clase A solamente cuando:

- B contiene a A.
- B es una agregación (o composición) de A.
- B almacena a A.
- B tiene los datos de inicialización de A (datos que requiere su constructor).
- B usa a A.

Brinda soporte a un bajo acoplamiento y mejores oportunidades de reutilización.

### **Controlador**

Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, y otras). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema. En UP (proceso unificado), al construir el modelo de análisis, existen estereotipos predefinidos que favorecen la separación de entidades, mecanismos de interfaz y mecanismos de control. En aplicaciones Web, se tiende a la separación de la lógica de presentación y de la lógica de negocio. Patrones bien conocidos como MVC o Fachada, son de amplia utilización.

En el diseño de clases de negocio, cuando no se tiene claro a qué clase pertenece la responsabilidad de realizar una determinada tarea, se debe crear una clase controladora que se llame igual a la tarea a desempeñar.

### **3.5 Patrones de arquitectura**

Expresan un paradigma fundamental para estructurar u organizar un sistema software. Proporcionan un conjunto de subsistemas o módulos predefinidos, con reglas y guías para organizar las relaciones entre ellos. Para la solución del modelado del sistema propuesto de utilizó el patrón MVC.

#### **MVC (Modelo-Vista-Controlador)**

La arquitectura del patrón Modelo-Vista-Controlador es un paradigma de programación bien conocido para el desarrollo de aplicaciones con interfaz gráfica. El principal objetivo de este patrón es aislar tanto los datos de la aplicación como el estado (modelo) de la misma, del mecanismo utilizado para representar (vista) dicho estado, así como para modularizar esta vista y modelar la transición entre estados del modelo (controlador).

Las aplicaciones MVC se dividen en tres grandes áreas funcionales:

- Vista: La presentación de los datos.
- Controlador: El que atenderá las peticiones y componentes para la toma de decisiones de la aplicación.
- Modelo: La lógica del negocio o servicio y los datos asociados con la aplicación.

Es una arquitectura preparada para los cambios, que desacopla datos y lógica de negocio de la lógica de presentación, permitiendo la actualización y desarrollo independiente de cada uno de los citados componentes.

### 3.6 Diagramas de clases Web

A diferencia del resto de los diagramas de clases, en el que se construye aplicaciones web, son más importantes la modelación de la lógica y estado del negocio, que los detalles de presentación, además cuentan con un grupo de estereotipos adicionales que permiten modelar estas aplicaciones.

En este caso los principales estereotipos utilizados fueron:

- Páginas Servidoras: Representa la página web que tiene código que se ejecuta en el servidor y el ícono usado fue:



**Figura 17: Ícono que representa las páginas servidoras.**

- Páginas Cliente: Representa una página web, con formato HTML. Son interpretadas por el navegador y el ícono usado fue:



**Figura 18:** Ícono que representa las páginas clientes.

- Formularios: Colección de elementos de entrada que son parte de una página cliente. Sus atributos son los elementos de entrada del formulario y el ícono usado fue:



**Figura 19:** Ícono que representa los formularios.

Además para una mejor representación gráfica se tomó la decisión de representar las clases del diseño por caso de uso, puesto que estas representarían de mejor forma las funcionalidades del sistema. Estos diagramas de clases se pueden encontrar desde [Anexos 1](#) hasta [Anexos 9](#).

### 3.7 Diagramas de Interacción

Los diagramas de interacción muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos. Modelan el comportamiento dinámico del sistema; el flujo de control en una operación. Describen la interacción entre objetos; los objetos interactúan a través de mensajes para cumplir ciertas tareas. Las interacciones proveen un “comportamiento” y típicamente implementan un caso de uso. Existen dos tipos de diagramas de interacción en UML:

- Diagramas de Secuencia (*dimensión temporal*)
- Diagramas de Colaboración (*dimensión estructural*)

#### 3.7.1 Diagramas de secuencia

Los diagramas de secuencia destacan el orden temporal de los mensajes, o sea, la forma en que las operaciones se producen en el tiempo. Se crean durante la elaboración, cuando es útil identificar los detalles de los eventos del sistema para poner en claro cuáles son las operaciones que se deben diseñar que gestione el sistema, escribir los contratos de las operaciones del sistema y posiblemente, dar soporte a la estimación.

Para modelar los aspectos dinámicos del sistema se utilizaron los diagramas de secuencia por cada escenario de caso de uso los cuales se pueden encontrar desde [Anexo 10](#) hasta [Anexo 24](#).

### 3.8 Descripción de las clases

Clases	Descripción
miAccionSuccess.php	Muestran el resultado de una acción determinada. Por ejemplo si se tiene una acción llamada miAcción el archivo de plantilla que se crea se llamará miAcciónSuccess.php
Layout	Es el template de la aplicación o sea es global para toda la aplicación o al menos para un grupo de páginas. Constituye la parte de afuera de las páginas y contiene a las plantillas dentro de él.
Actions y Action	Son las clases que tienen el sufijo Actions o Action las cuales representan las acciones de forma tal que puedan acceder directamente (por herencia) a las funcionalidades del núcleo de Symfony, y utilizar así las funcionalidades, ya que estas son las que van a representar las principales acciones que va a desarrollar la aplicación. Las acciones contienen la lógica de la aplicación. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación.
Clases del modelo	Estas clases representan de forma Orientada a Objetos el Modelo Entidad Relación de la base de datos, de la aplicación, se generan automáticamente, en función de la estructura de datos de la aplicación. Implementan la lógica del negocio porque representan la información con la que trabaja la aplicación, ejemplo de ellas son las clases: Persona, PersonaPeer, Foto, FotoPeer, Usuario, UsuarioPeer, Rol, RolPeer, UsuarioRol, UsuarioRolPeer.
WS	La clase que tiene este prefijo, es la clase del servicio que implementa estas funcionalidades del lado del servidor. Mediante el plugin ckWebServicePlugin se publican las acciones como servicios web. Mediante el WSDL se brinda al usuario la información necesaria para acceder a las mismas.
SoapClient	Es la clase que se encarga de instanciar un nuevo cliente para comunicarse con un servicio web externo.
Librerías externas.	Son las librerías en las que se apoyan las acciones para realizar tareas específicas tal es el caso de la clase gestionarMetadatos, la clase estandarizarFoto y LDAP.

PHP_JPEG_Metadata_toolkit	Es una librería para PHP que se encarga de gestionar los metadatos en las imágenes con formato de tipo JPEG.
---------------------------	--------------------------------------------------------------------------------------------------------------

**Tabla 3. 1: Descripción de las clases utilizadas en el diseño.**

### **3.9 Conclusiones**

En este capítulo se modela todo el flujo de trabajo análisis y diseño, basado en la propuesta de sistema. Se hace una breve descripción de los patrones de diseño y de arquitectura a utilizar, definidos por el grupo de arquitecto del proyecto de Control de Acceso. Se modela utilizando estereotipos web, lo cual ha dado una mejor perspectiva y entendimiento de los diagramas antes presentados y por último se presentan algunos artefactos de los obtenidos en este flujo: diagramas de clases del análisis y del diseño, diagramas de secuencia y se hace una descripción de las clases utilizadas en estos.

### CONCLUSIONES

En el presente trabajo a partir de la verificación de si era viable llevar a cabo el modelado del sistema propuesto mediante un estudio de la factibilidad del software basado en la evaluación por puntos de casos de usos, se obtuvo como resultados el cumplimiento de los objetivos propuestos pues en el desempeño del autor en los roles de analista y diseñador se generó toda la documentación y los artefactos de los flujos de trabajo modelación de negocio, requisitos, análisis y diseño del sistema de gestión de fotos, dejándolo listo para una posterior implementación del mismo.

La utilización de la metodología RUP avaló que el desarrollo de software quedase estructurado por fases, con la calidad requerida y disminuyendo el tiempo de desarrollo, además constituye una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

## **RECOMENDACIONES**

Tomando como base la experiencia acumulada durante la realización del trabajo se recomienda:

- Extender el uso de las aplicaciones basadas en los servicios Web dentro del proyecto Control de Acceso y de la Universidad en general para lograr la interoperabilidad entre los diversos sistemas, independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Realizar un estudio sobre tecnologías y técnicas para servicios web, por parte de los investigadores del proyecto Control de Acceso, que protejan los mensajes desde el principio hasta el destino final e integrarlo en una futura versión del sistema.
- Realizar la implementación del sistema y la ejecución de las pruebas pertinentes, por parte de los equipo de implementación y de calidad del proyecto Control de Acceso, así como generar la documentación necesaria para estas fases, por las ventajas que esto trae consigo para los usuarios del sistema.
- Implementar un manual de usuario por parte de los documentadores del proyecto Control de Acceso para facilitar el trabajo a los usuarios finales.

## BIBLIOGRAFÍA

### Bibliografía citada

- 1. Formatos de almacenamiento de imágenes digitales. Citado: enero de 2008 [En línea]  
<http://www.dzoom.org.es/cont-38-formato-fichero-imagen-fotografia.html>.
- 2. Resolución de la media de la imagen digital. Citado: enero de 2008 [En línea]  
<http://www.guesabesde.com/camdig/articulos.asp?articulo=96>.
- 3. IVAR JACOBSON, GRADY BOOCH y otros. *El Proceso Unificado de Desarrollo de Software*. 2004.
- 4. JORRÍN, Sergio Alejandro. *Arquitectura SOA para la integración entre software libre y software propietario en entornos mixtos*.
- 5. Henst, Christian Van Der. ¿Qué es el PHP? Citado: febrero de 2008 [En línea]  
<http://www.maestrosdelweb.com/editorial/phpintro>.
- 6. Alfaro, Ricardo. ¿Por qué elegir PHP? Citado: febrero de 2008 [En línea]  
<http://www.mmug.cl/articulos.php?id=283>.
- 7. Kaisler, Stephen H. *Software Paradigms*. 2005.

### Bibliografía consultada

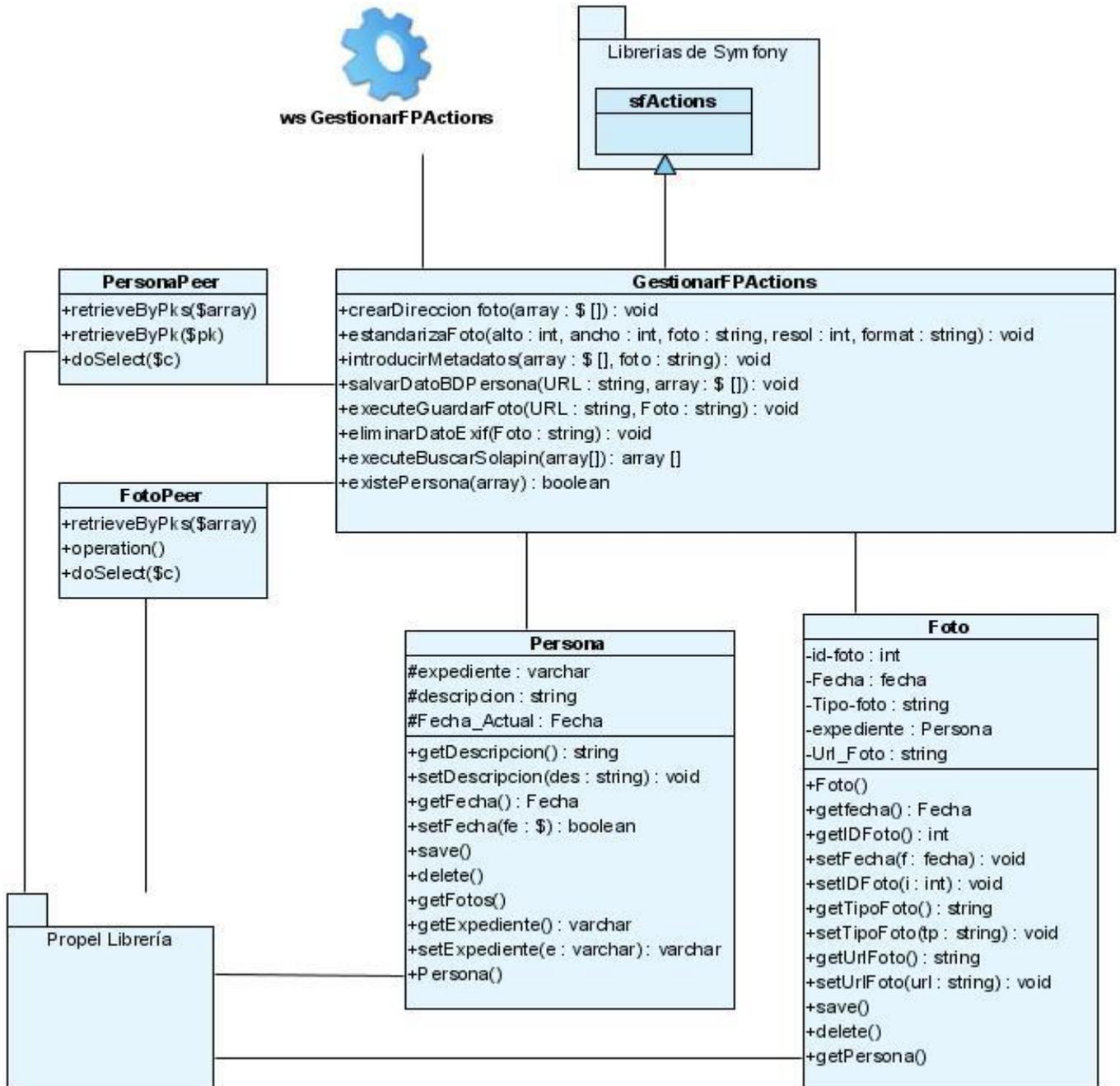
- Asignatura: Ingeniería de Software I Curso: 2007-2008
- Booch, G.; Rumbaugh, J. y Jacobson, I.; "El Lenguaje Unificado de Modelado". 2000. Addison-Wesley.
- Boehm "SW Cost Estimation with COCOMO II". Prentice Hall. 2000.
- Comprendiendo los metadatos. Consultada: diciembre 2007 [En línea]  
<http://www.fotonatura.org/revista/articulos/293/1>
- Diez razones para escoger Visual Paradigm.
- Hernández, Rolando Alfredo; Coello, Sayda. El paradigma cuantitativo de la investigación científica
- Entorno Virtual de aprendizaje.

- Fabien Potencier, François Zaninotto “*Symfony la guía definitiva* “. Consultada: febrero de 2008 [En línea] <http://www.librosweb.es>
- GOF. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
- Herrera J., Lizka Johany. “*Ingeniería de Requerimientos – Ingeniería de Software*”. Consultada: diciembre de 2008 [En línea] <http://www.monografias.com/trabajos6/resof>.
- Introducción a los Web Services en PHP. desarrolloweb.com. Consultada: diciembre de 2007 [En línea] <http://www.desarrolloweb.com/articulos/1852.php>.
- Introducción al ckWebServicePlugin. Consultada: marzo de 2008 [En línea] <http://www.symfony-project.org>
- Introducción a php. Consultada: febrero de 2008 [En línea] <http://www.ciberteca.net/webmaster/php>.
- J. G. Daugman. “*High Confidence Visual Recognition of Persons by a Test of Statistical Independence*”. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, nº 11. Noviembre 1993.
- Jim Conallen, “*Building Web Applications with UML*”, Object Technology Series by Addison Wesley Longman, 1999
- Jim Conallen, “*UML Extension for Web Applications* “.
- Jim Conallen, “*Modeling Web Application Architectures with UML*”, Rational Software, June 1999
- A. K. Jain. “*Fundamentals of Digital Image Processing*”. Prentice Hall, 1989.
- Jain, A., Bolle, R. & S. Pankanti. Biometrics: “*Personal Identification in a Networked Society*”, Eds. Kluwer, 1999.
- JACOBSON, I.” *Modeling with use cases- Formalizing use-case modelling Journal of Object-Oriented Programming*”, 1995.
- Johnson, R. E. and Helm, R. 1995. “*Design Patterns: Elements of Reusable Object-Oriented Software*”. Addison-Wesley Pub Co.
- LARMAN, C. “*UML y PATRONES. Introducción al análisis y diseño orientado a objetos*”. México, Prentice Hall, 1999. 536 p.

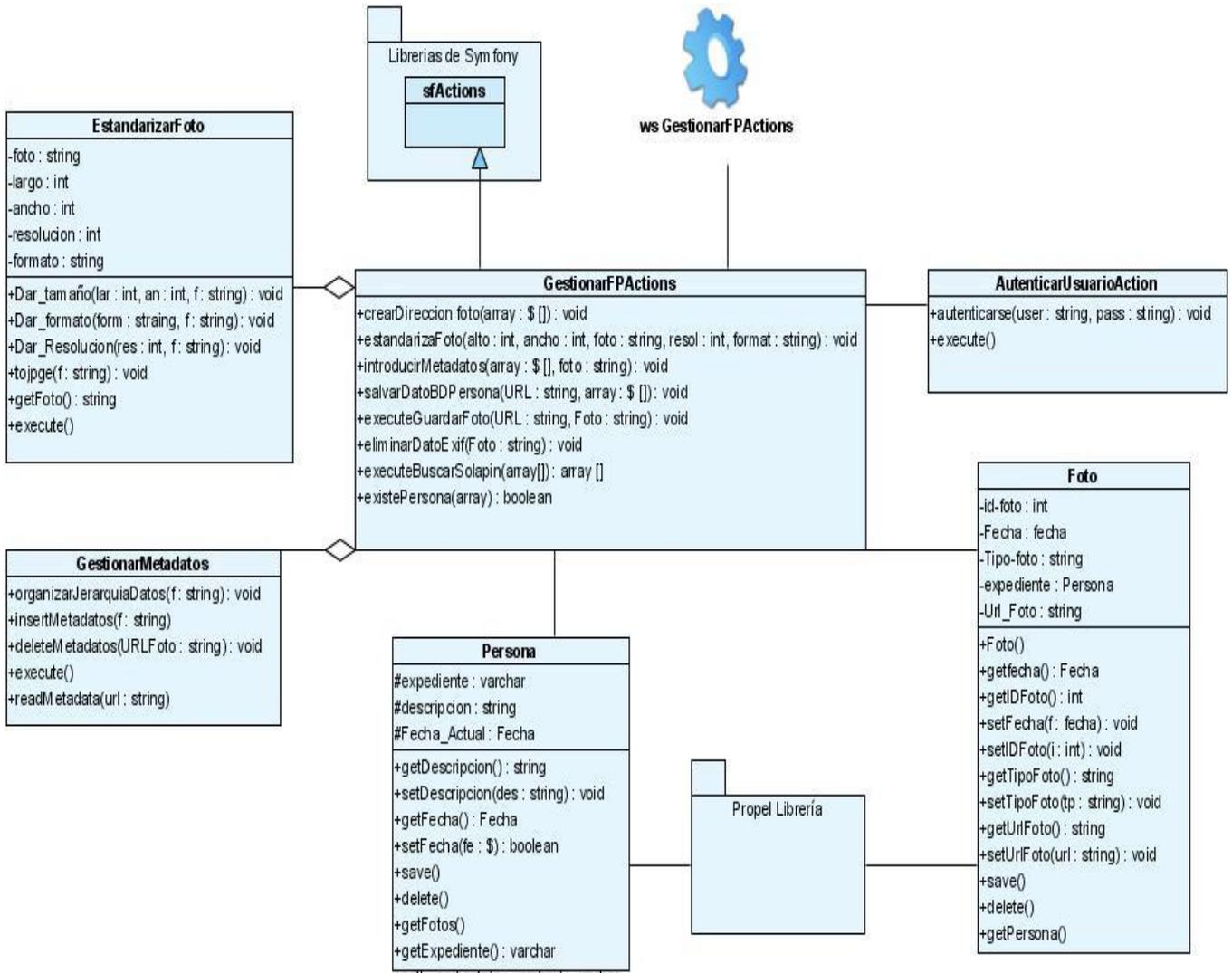
- Lellelid, Hans. 2004. "*Propel - Guía de usuario*".
- Manual de JavaScript Consultada: enero de 2008 [En línea]  
[http://www.redestb.es/soporte/aula/jScript.\(17/01/2002\)](http://www.redestb.es/soporte/aula/jScript.(17/01/2002))
- Pressman, Rogers." *Ingeniería del software*". Parte 1. La Habana, 2005. Editorial Félix Varela.
- Pressman, Rogers. "*Ingeniería del software*". Parte 2. La Habana, 2005. Editorial Félix Varela.
- Rational Unified Process, Rational Software Corporation, "Rational Unified Process", Version 2001A.04.00, Copyright 1987-2001.

## ANEXOS

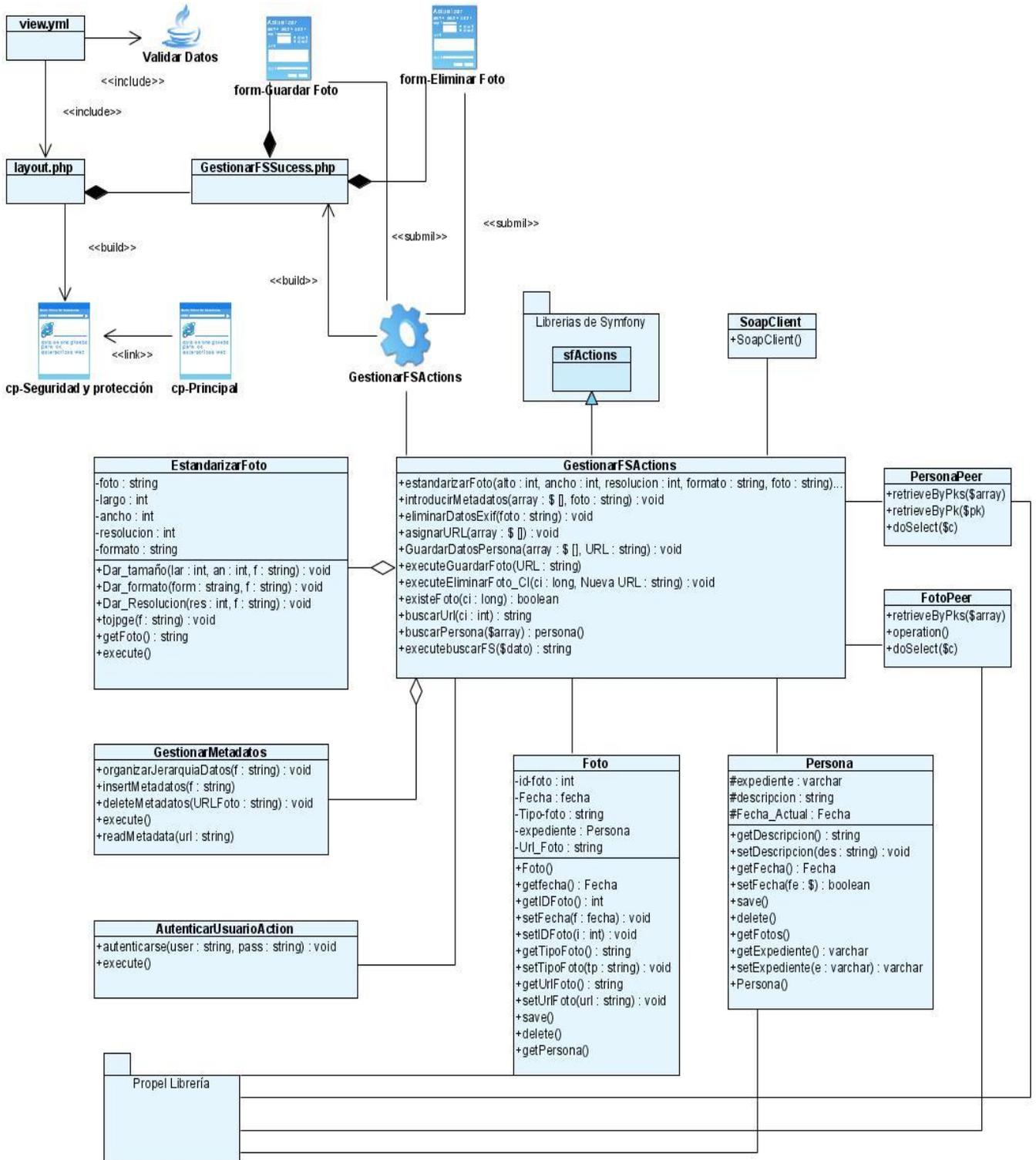
## Anexo 1: Diagrama del diseño del CU Buscar foto principal.



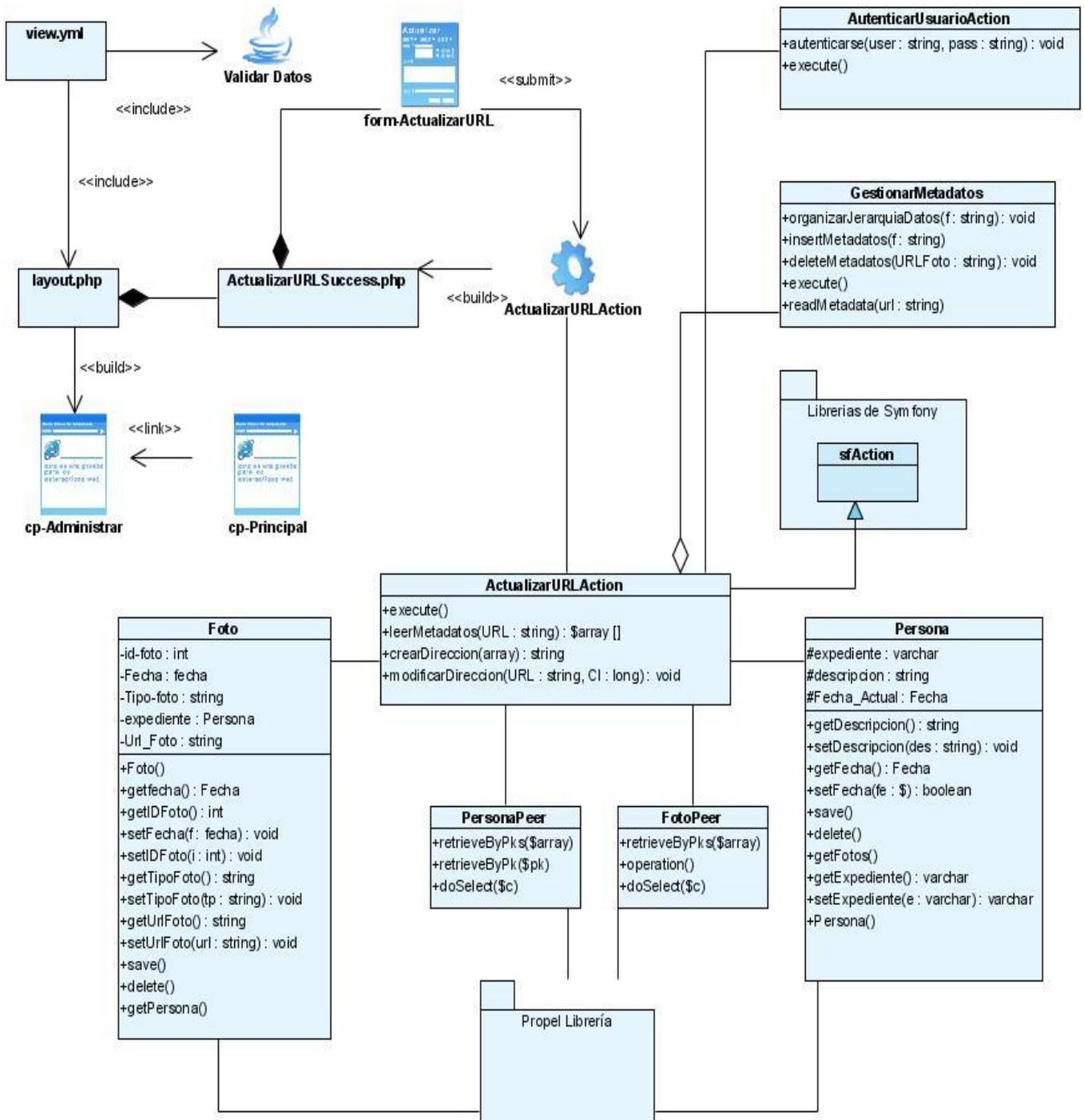
Anexo 2: Diagrama del diseño del CU Guardar foto principal.



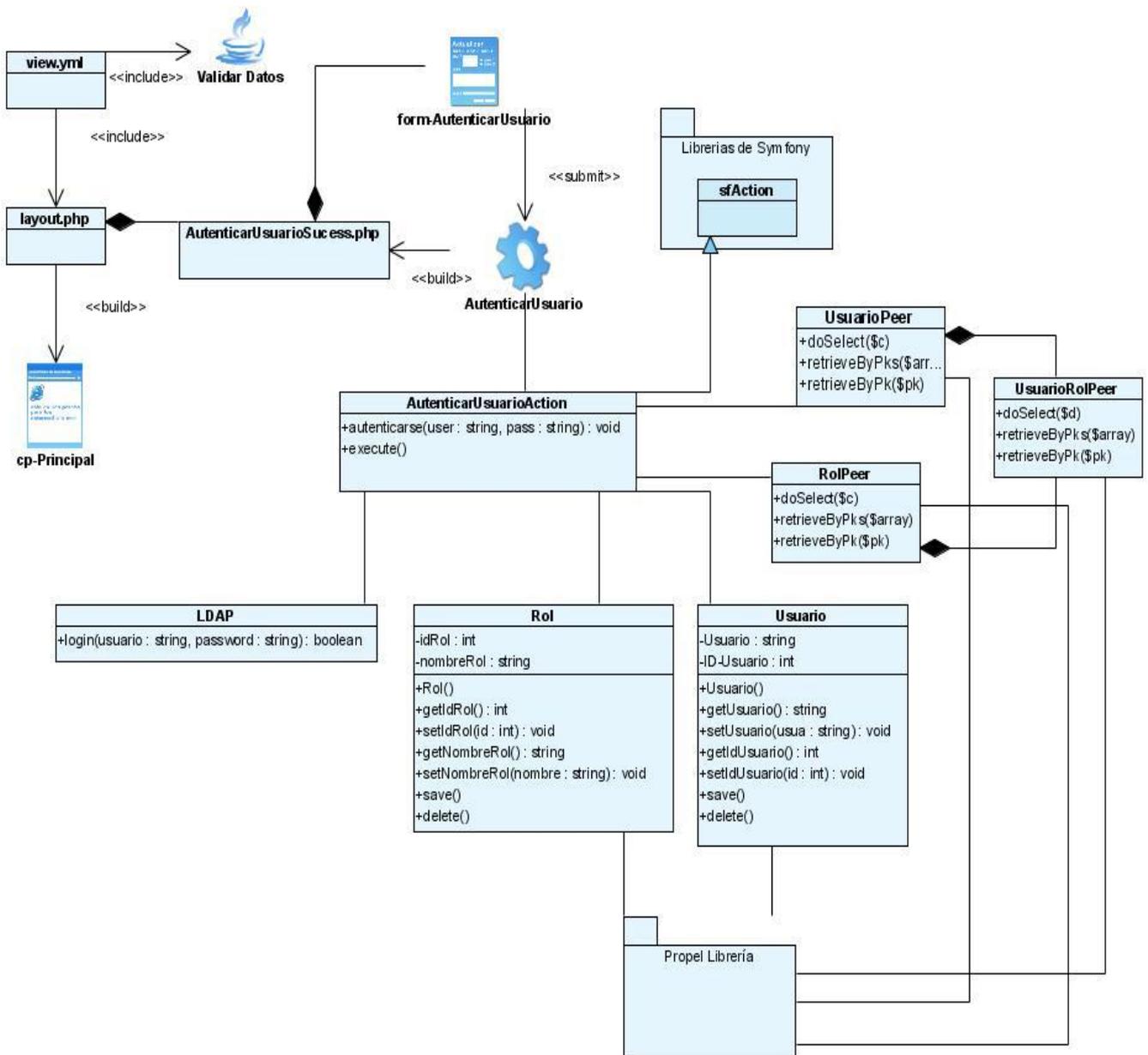
Anexo 3: Diagrama del diseño del CU Gestionar foto seguridad.



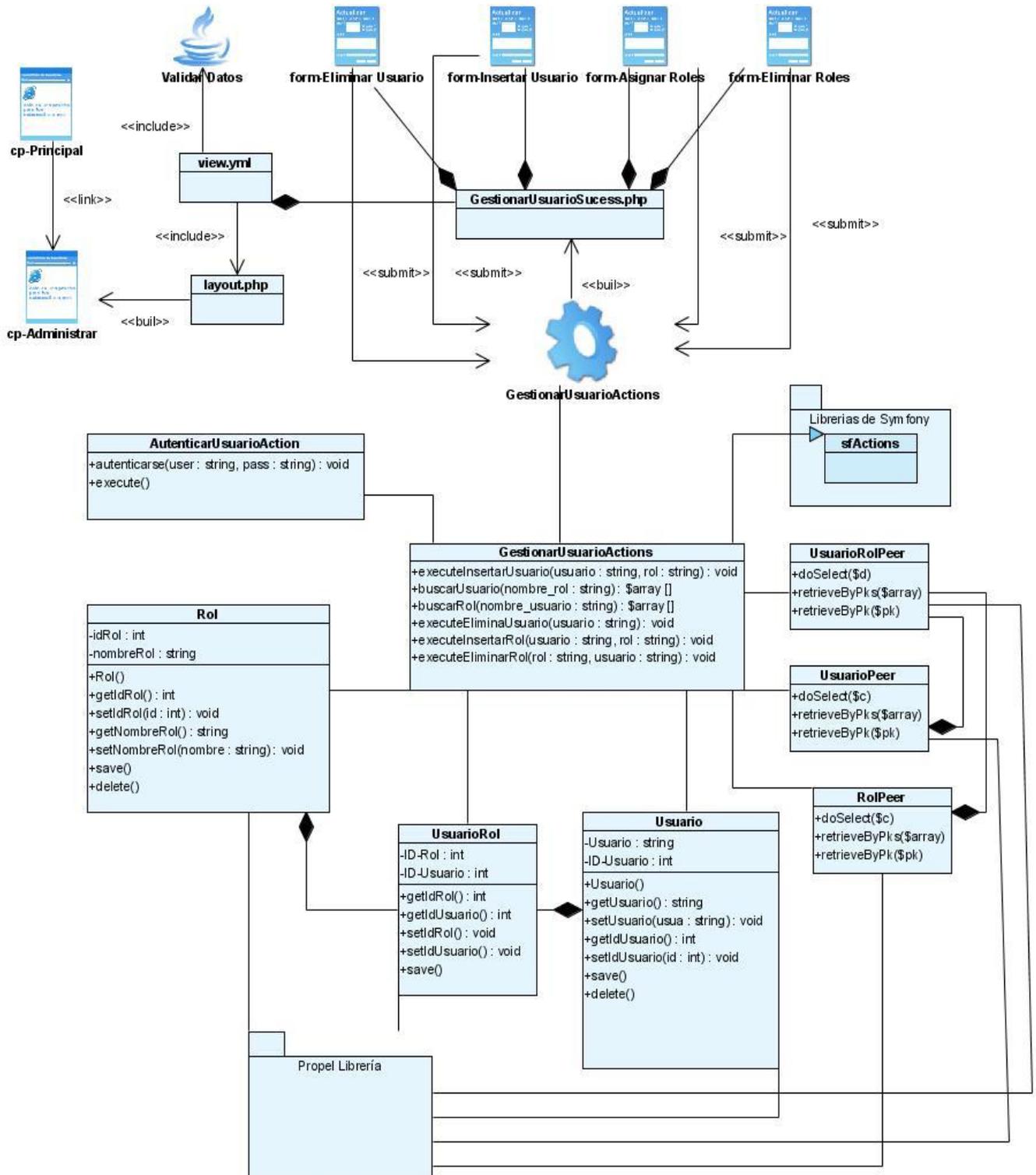
Anexo 4: Diagrama del diseño del CU Actualizar camino a imagen



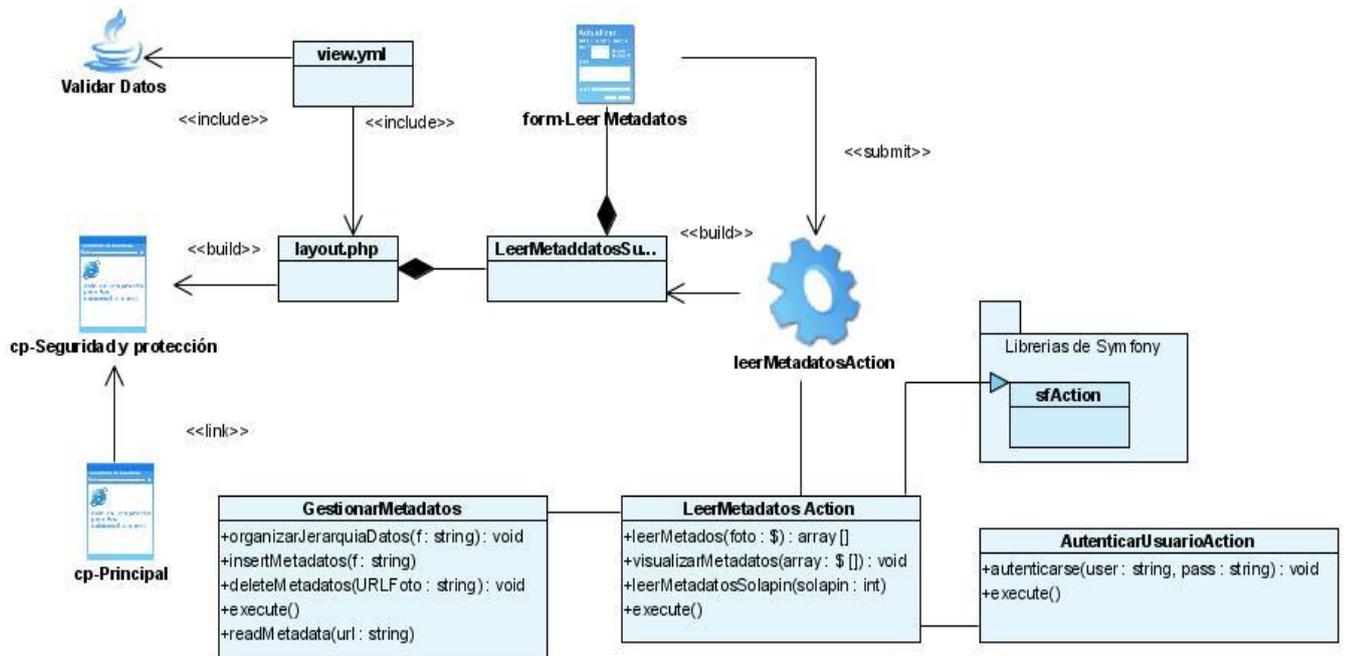
Anexo 5: Diagrama del diseño del CU Autenticar usuario.



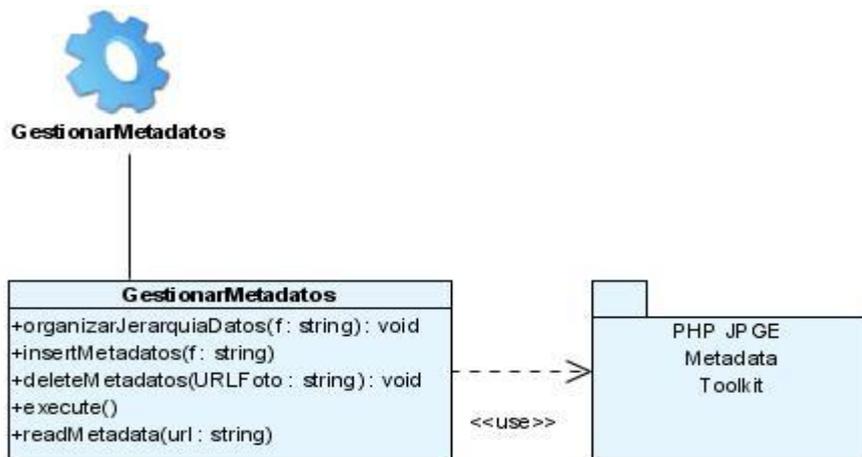
Anexo 6: Diagrama del diseño del CU Gestionar usuarios.



Anexo 7: Diagrama del diseño del CU Leer metadato.



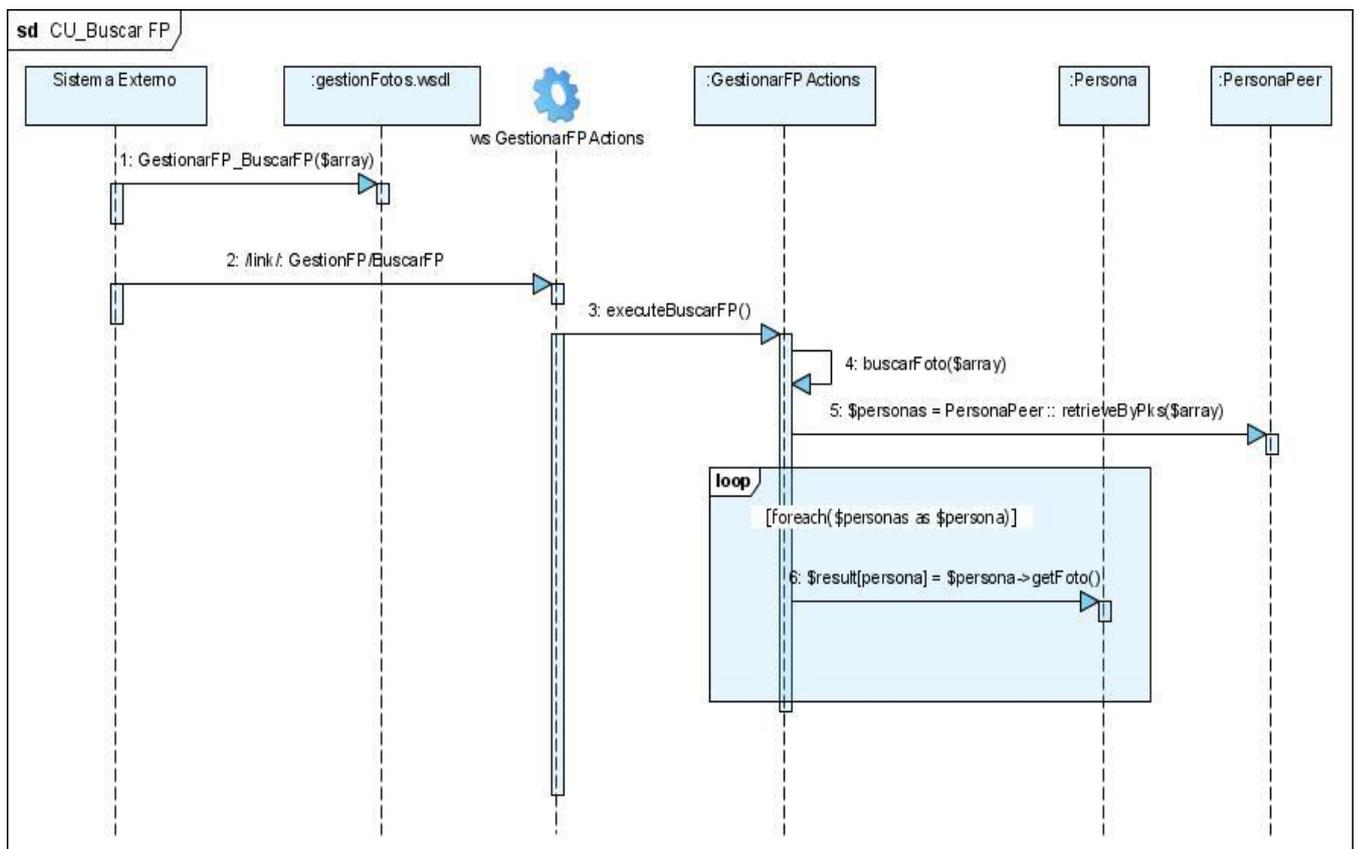
Anexo 8: Diagrama del diseño del CU Gestionar metadato.



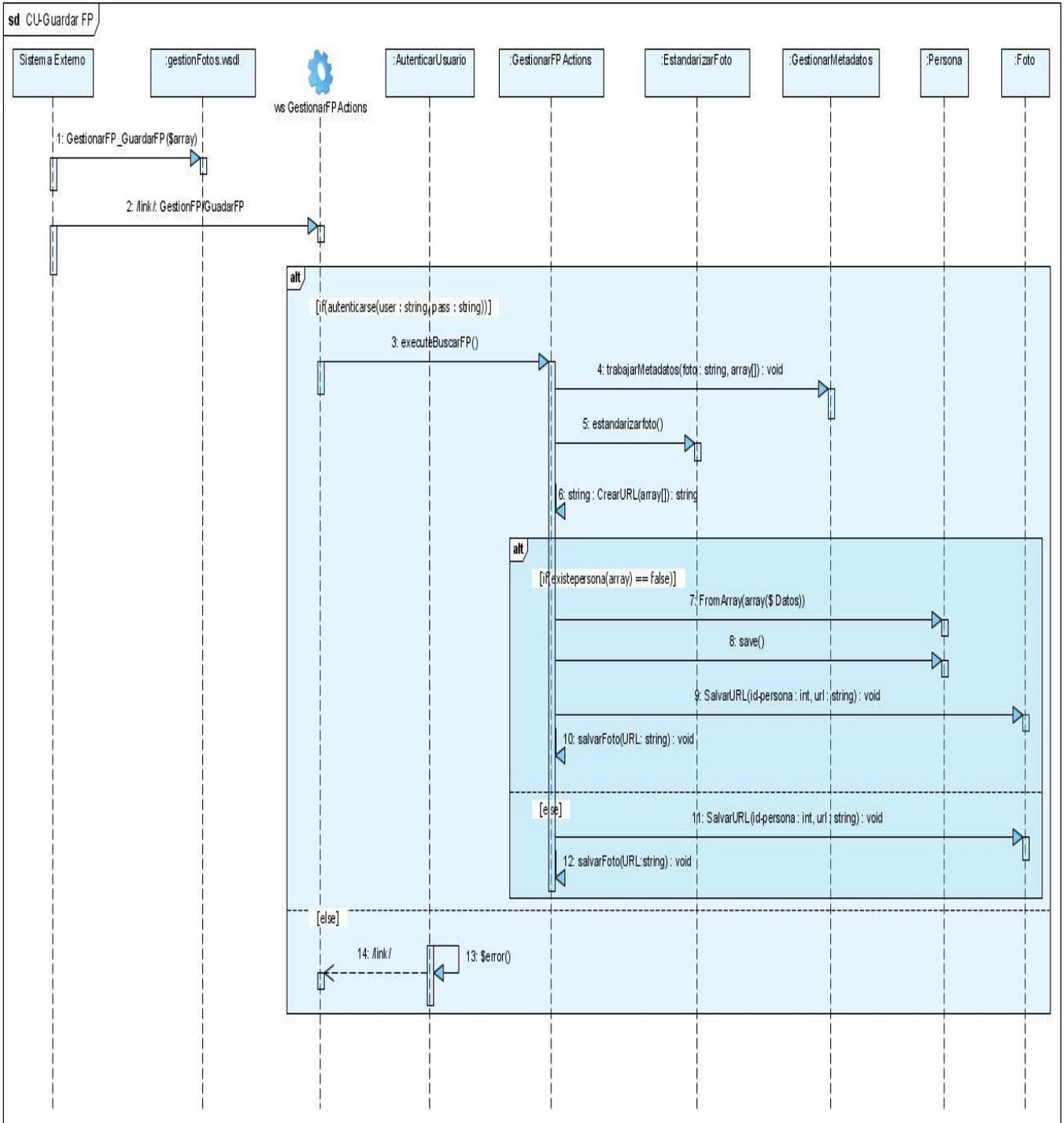
Anexo 9: Diagrama del diseño del CU Estandarizar foto.



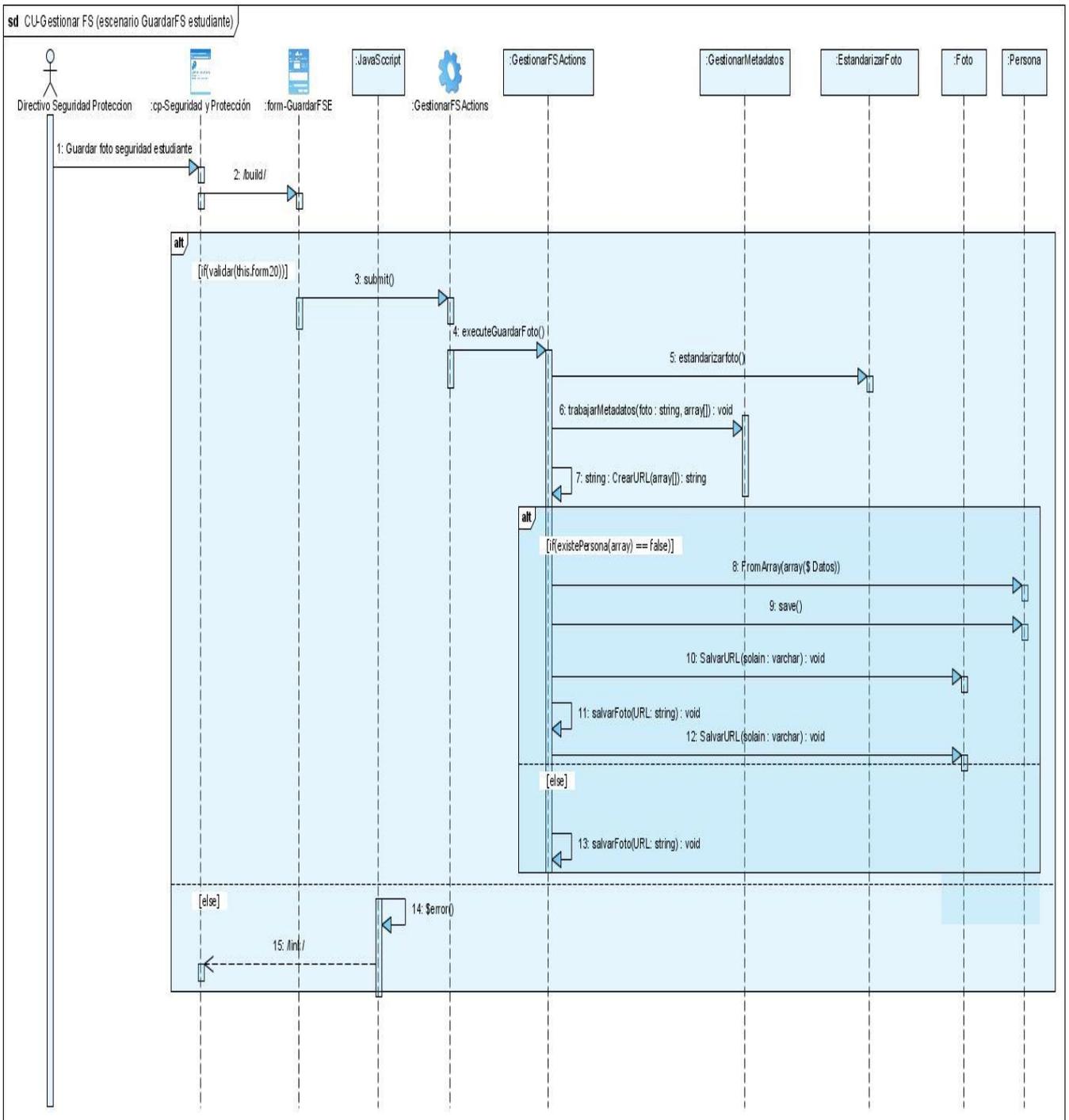
Anexo 10: Diagrama de secuencia del CU Buscar foto principal.



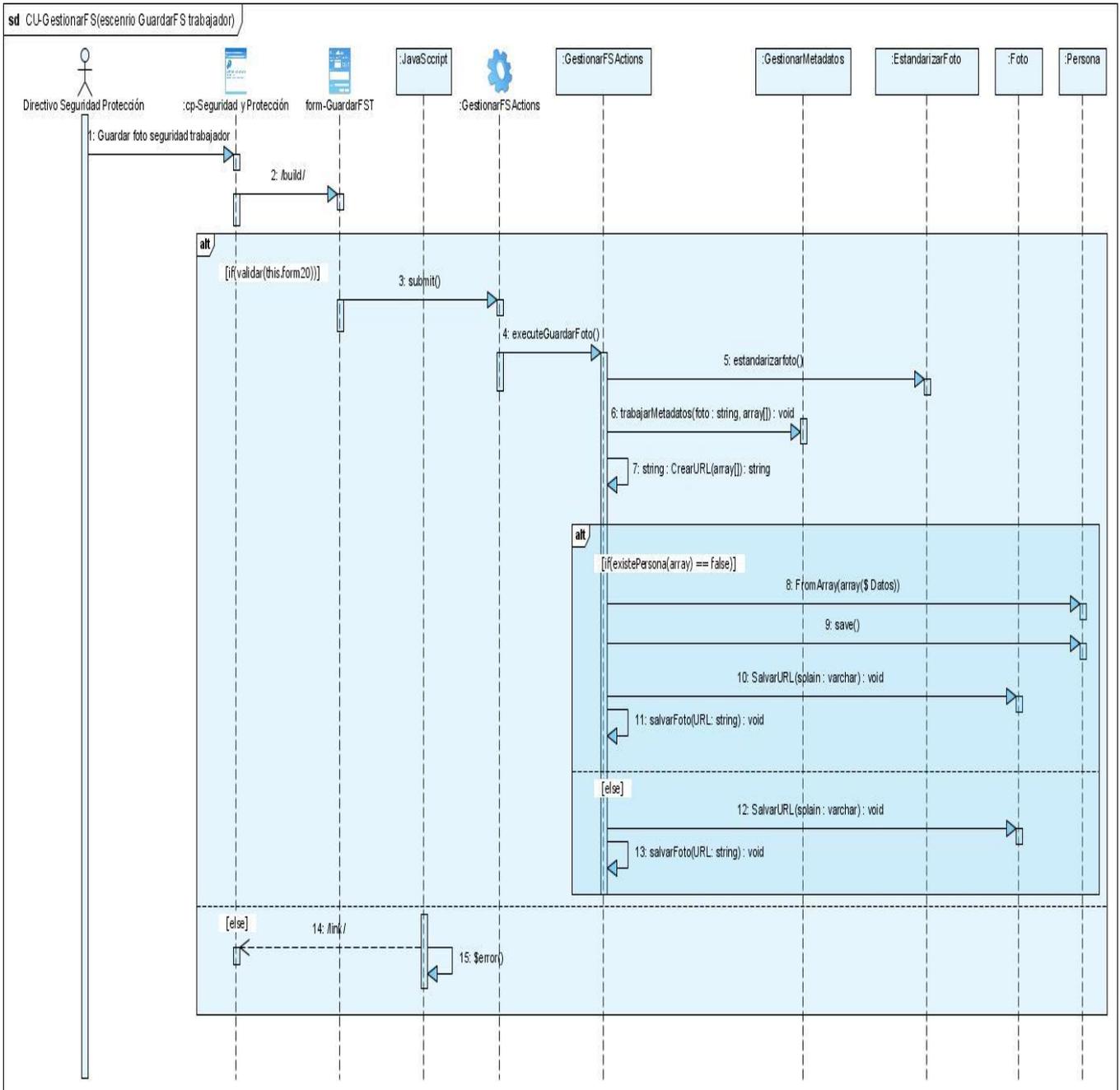
Anexo 11: Diagrama de secuencia del CU Guardar foto principal.



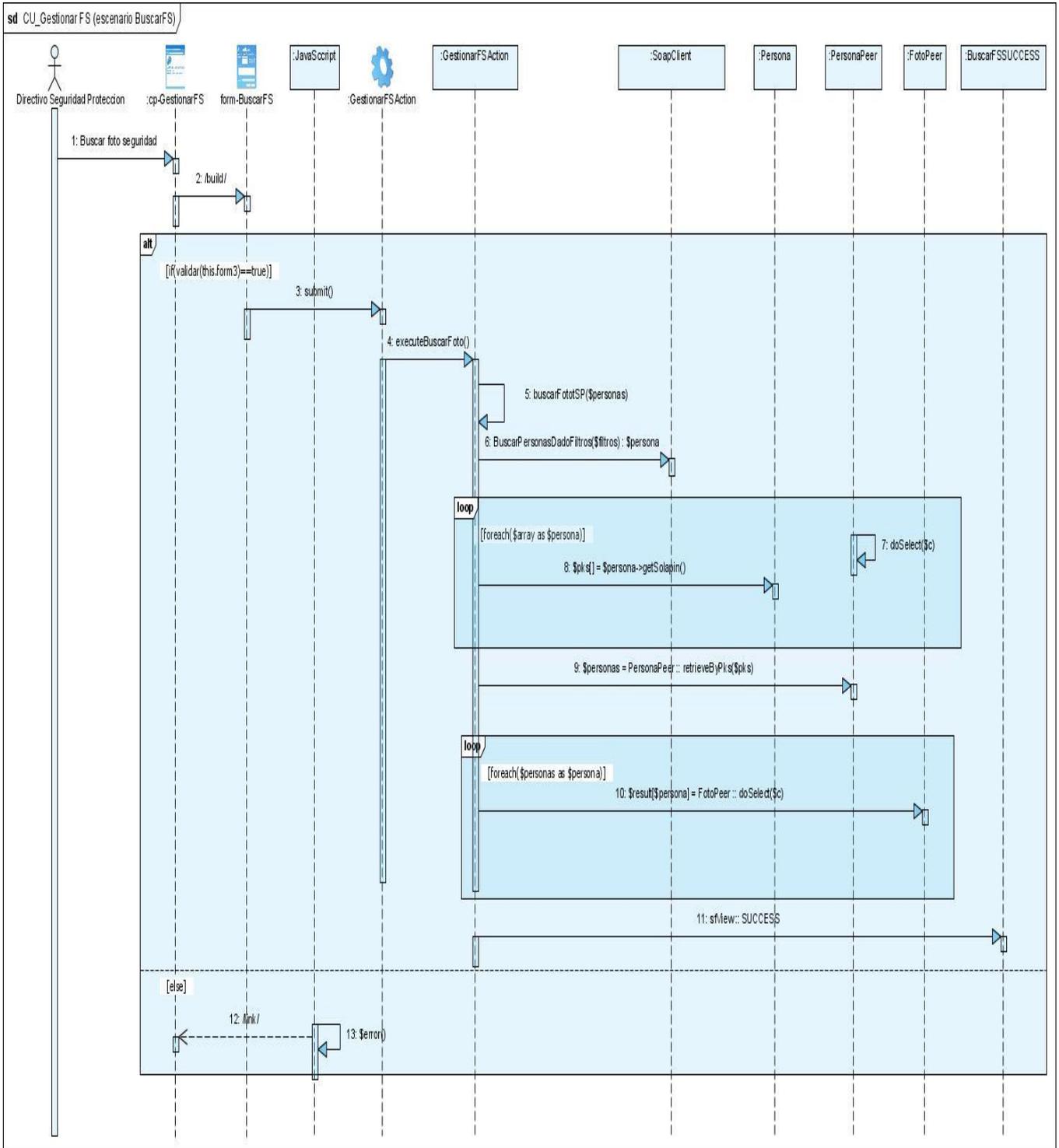
**Anexo 12: Diagrama de secuencia del CU Gestionar foto seguridad escenario guardar foto estudiante.**



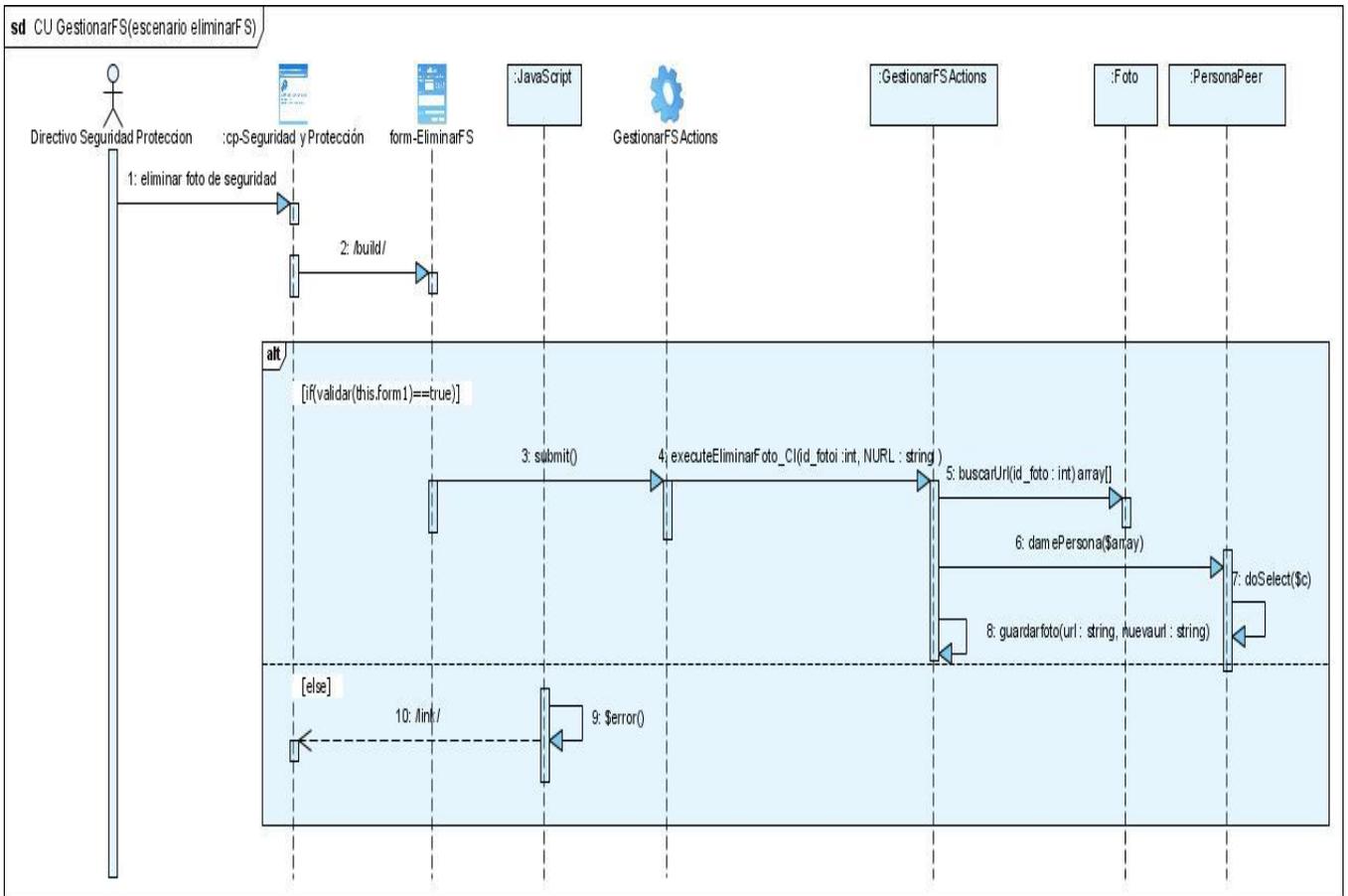
**Anexo 13: Diagrama de secuencia del CU Gestionar foto seguridad escenario guardar foto trabajadores.**



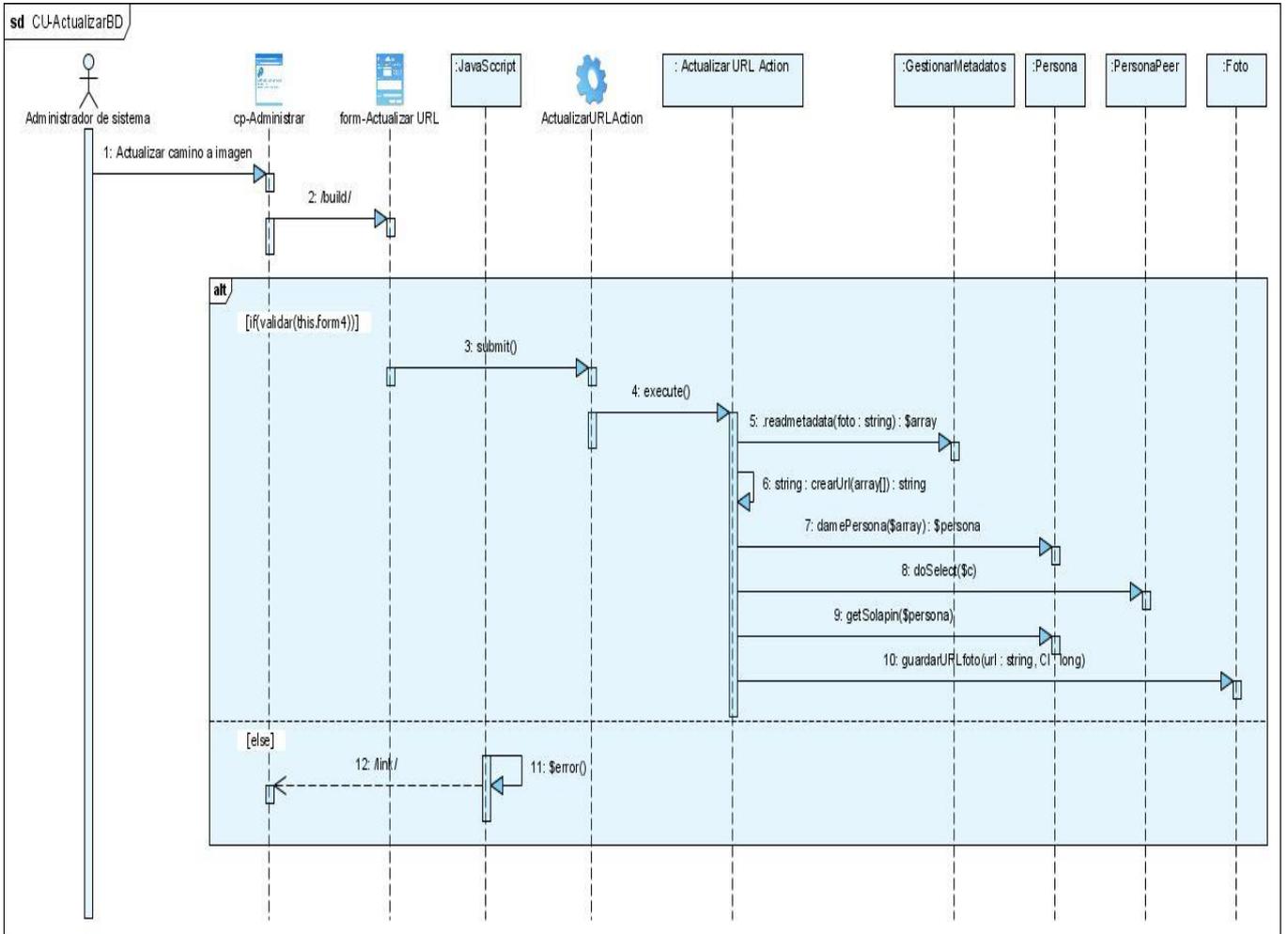
Anexo 14: Diagrama de secuencia del CU Gestionar foto seguridad escenario buscar foto.



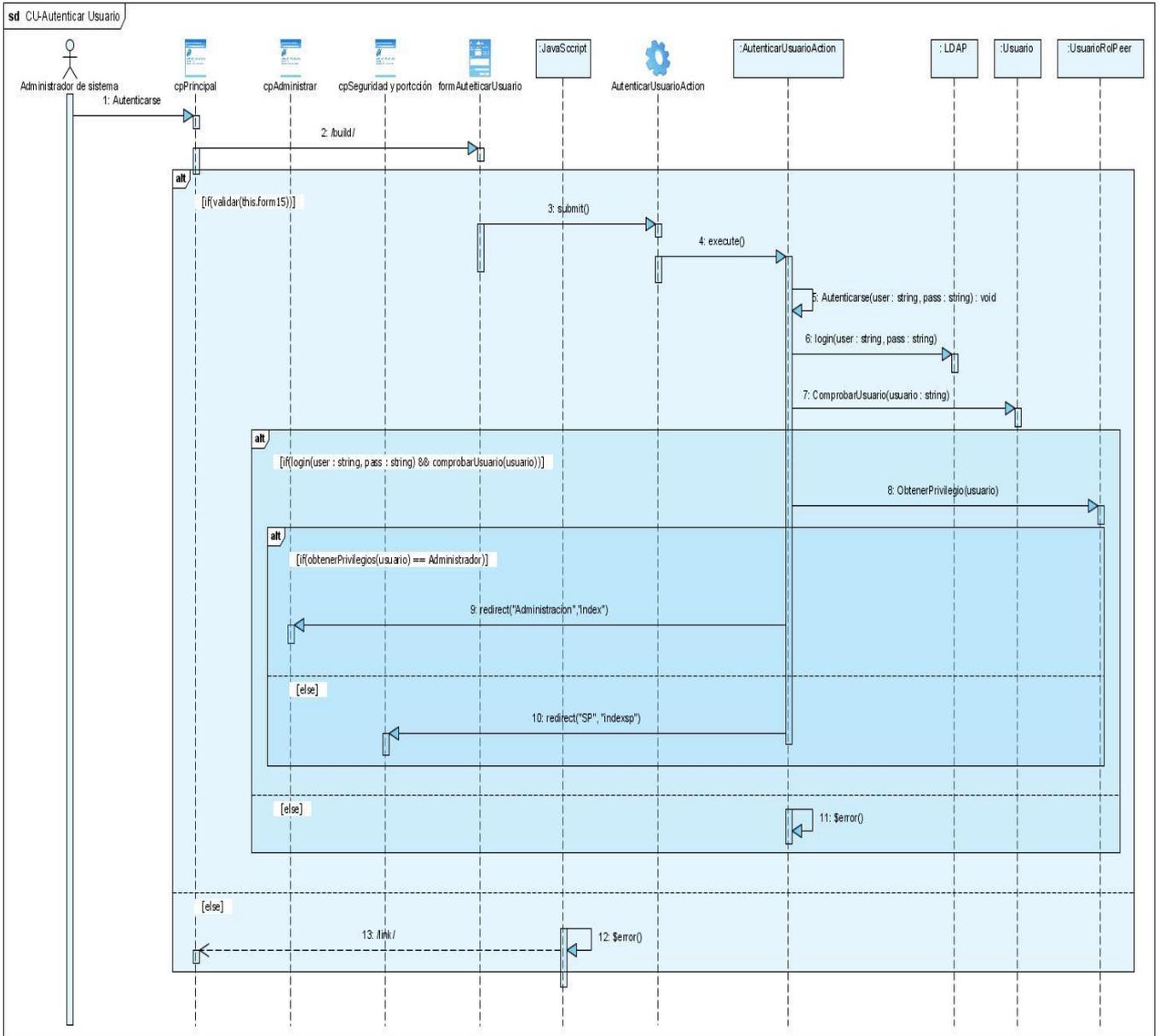
**Anexo 15: Diagrama de secuencia del CU Gestionar foto seguridad escenario eliminar foto.**



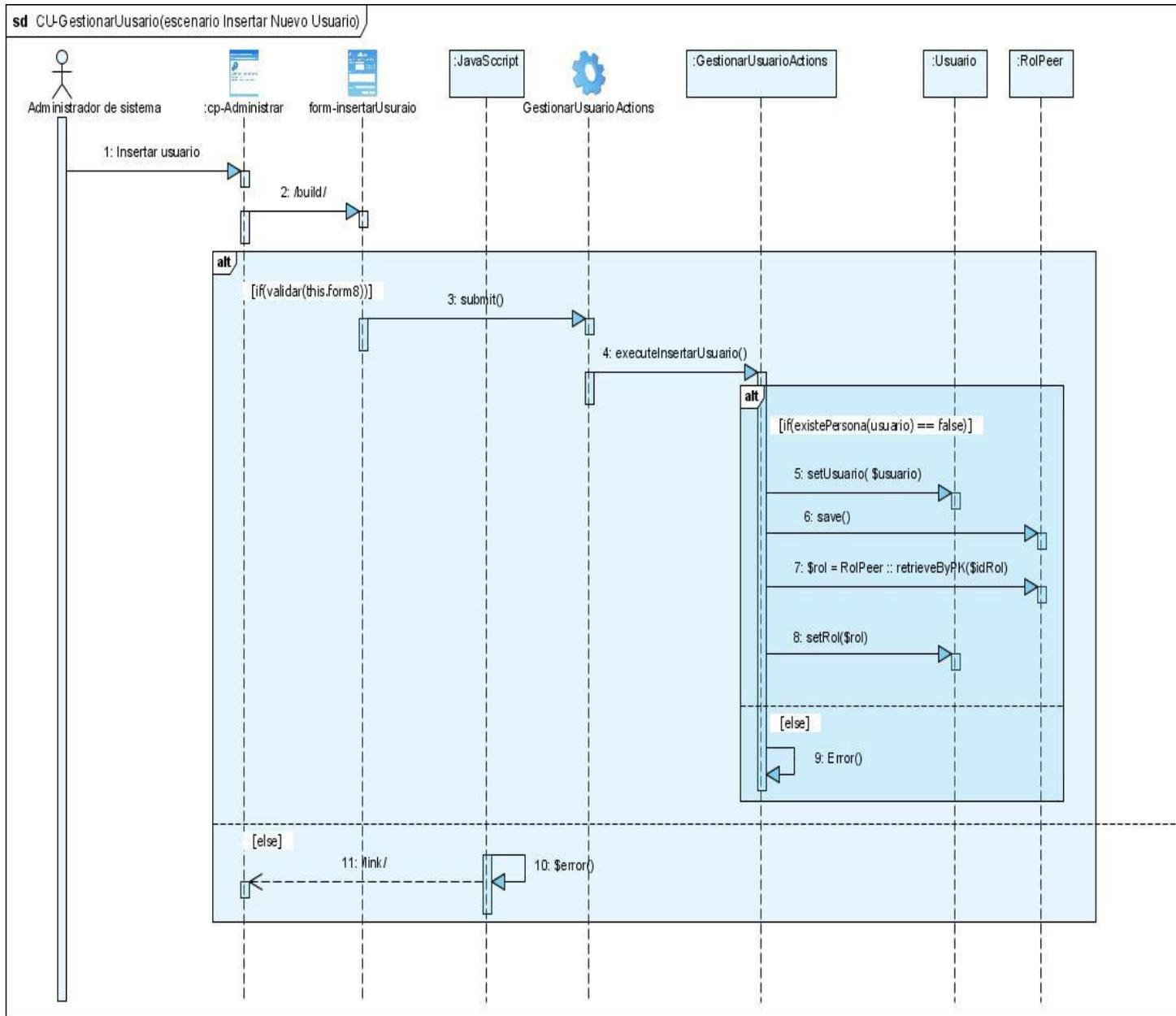
Anexo 16: Diagrama de secuencia del CU Actualizar camino a imagen.



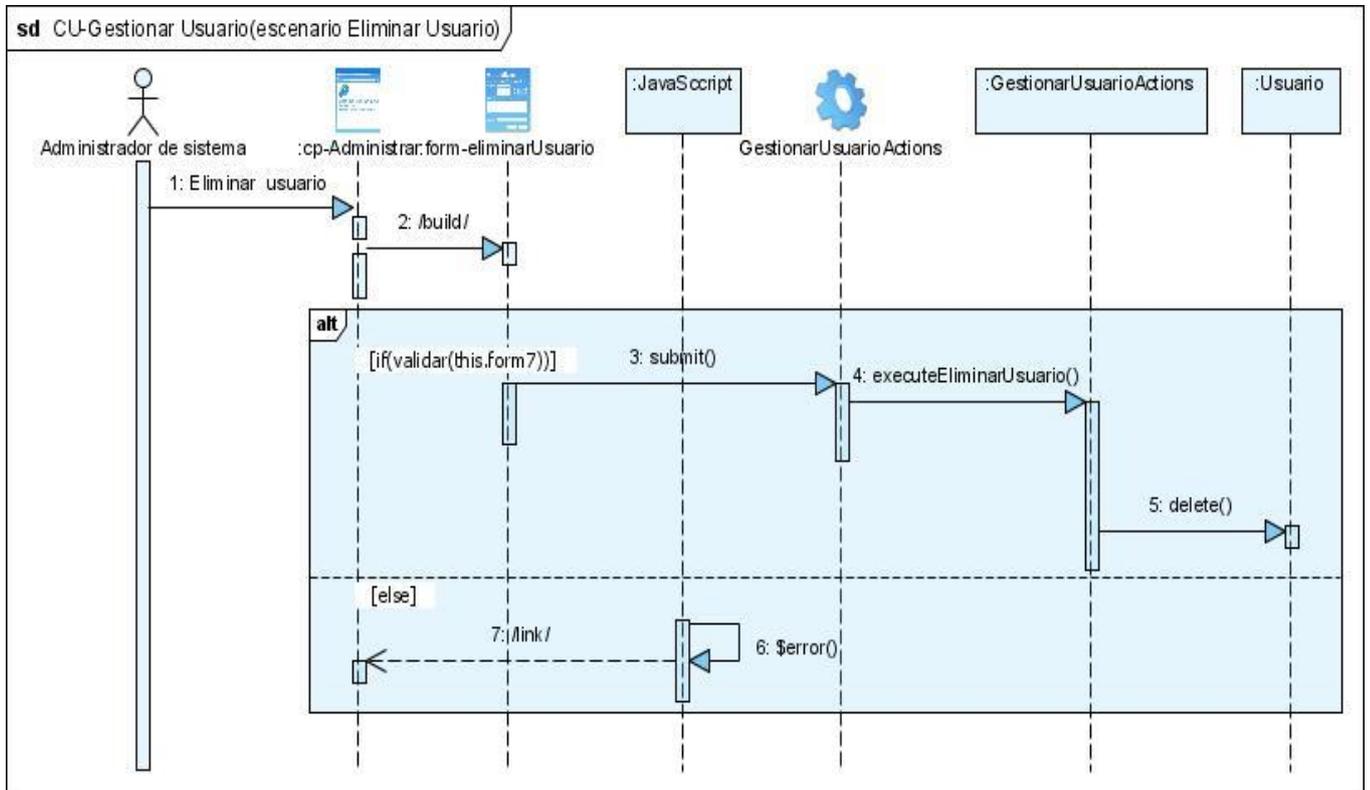
Anexo 17: Diagrama de secuencia del CU Autenticar usuario.



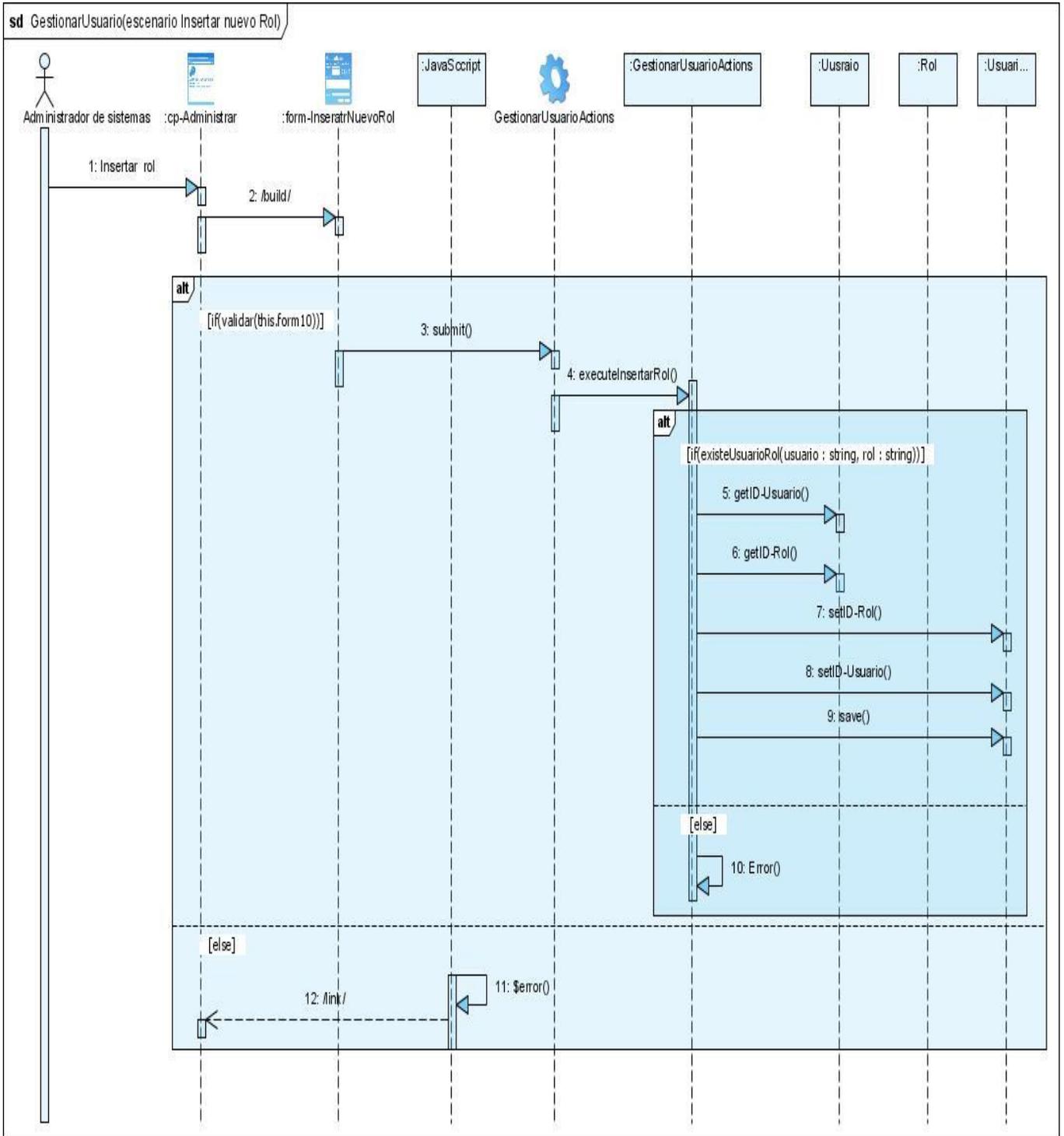
Anexo 18: Diagrama de secuencia del CU Gestionar usuario escenario insertar nuevo usuario.



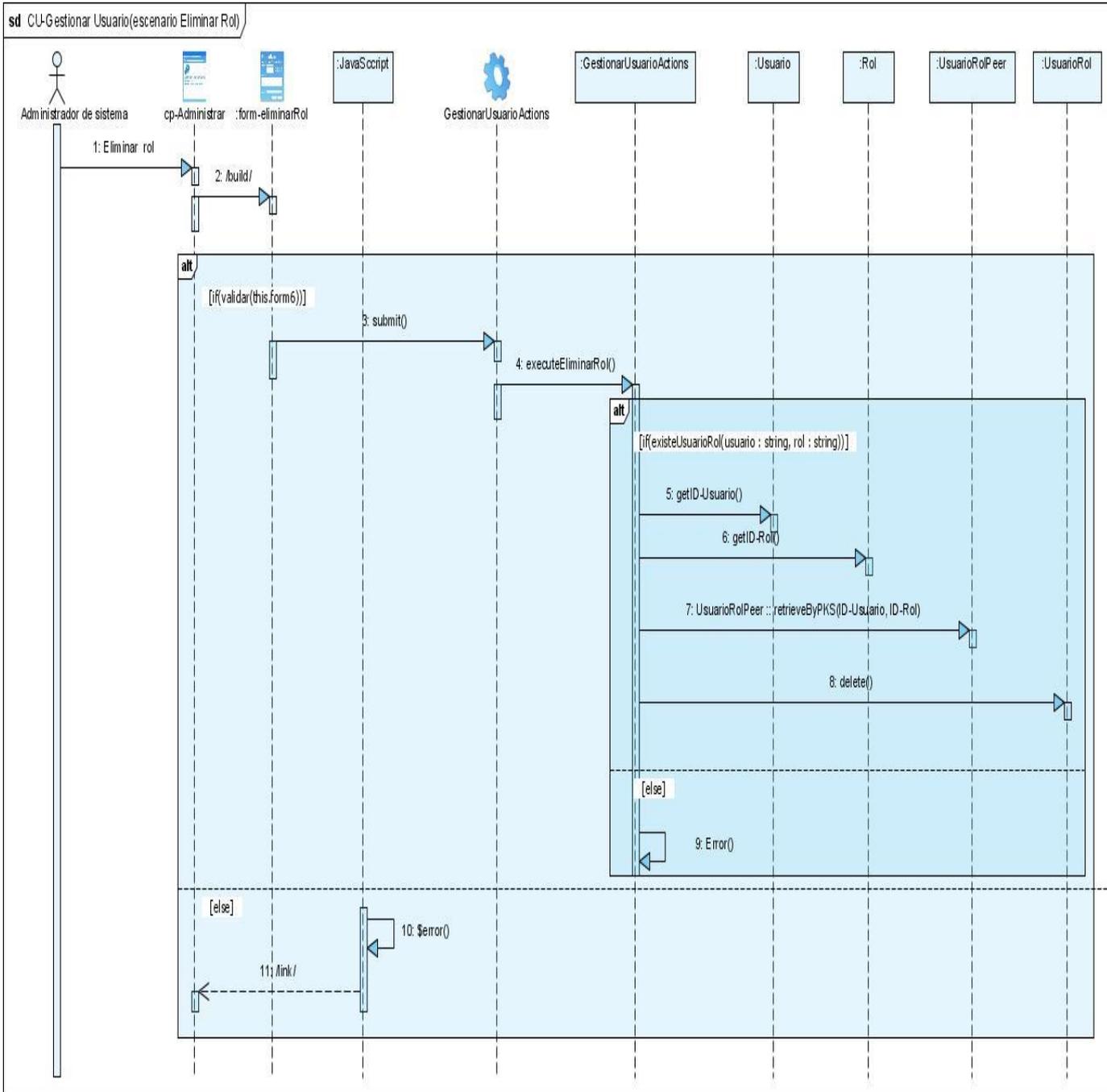
**Anexo 19: Diagrama de secuencia del CU Gestionar usuario escenario eliminar usuario.**



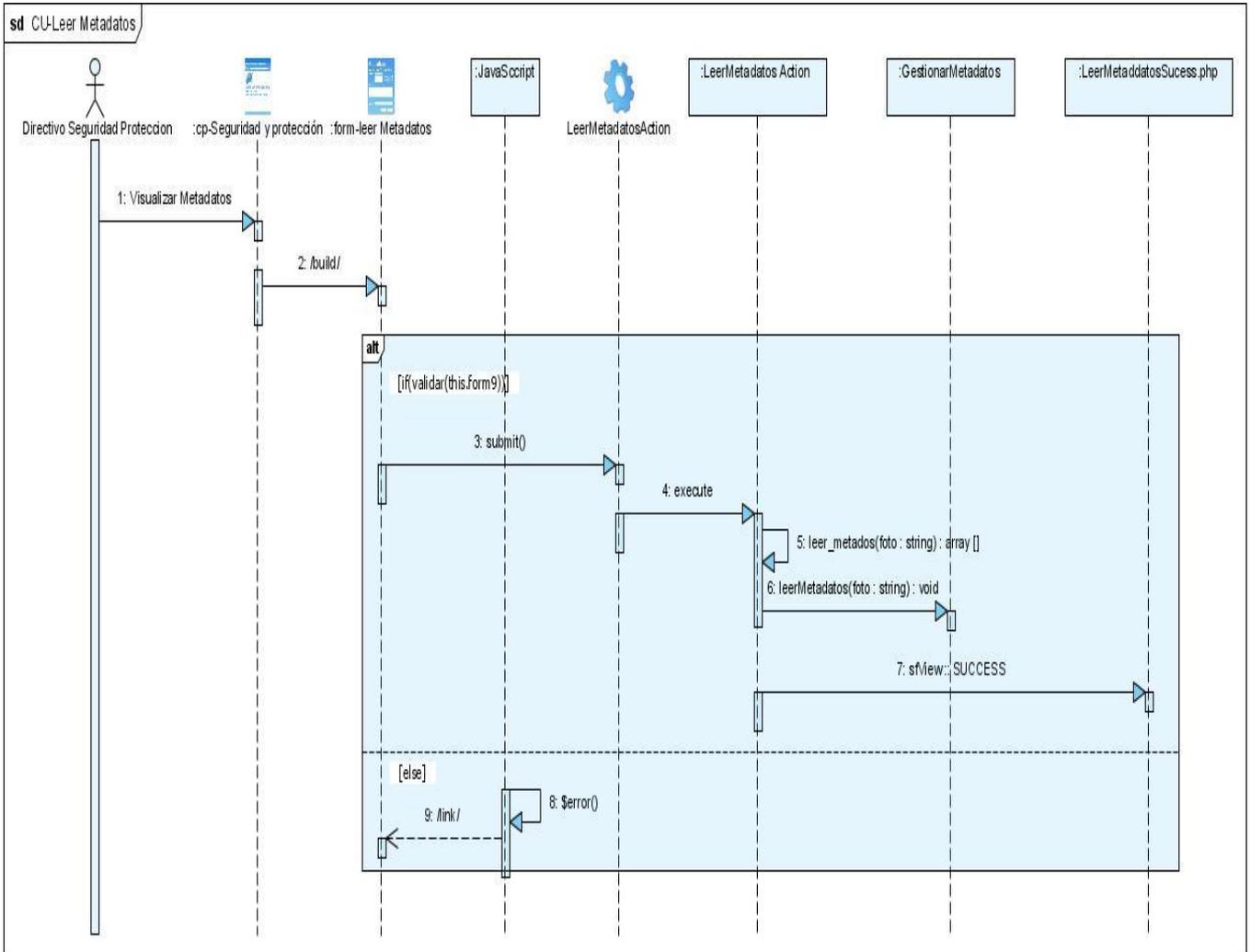
Anexo 20: Diagrama de secuencia del CU Gestionar usuario escenario insertar nuevo rol.



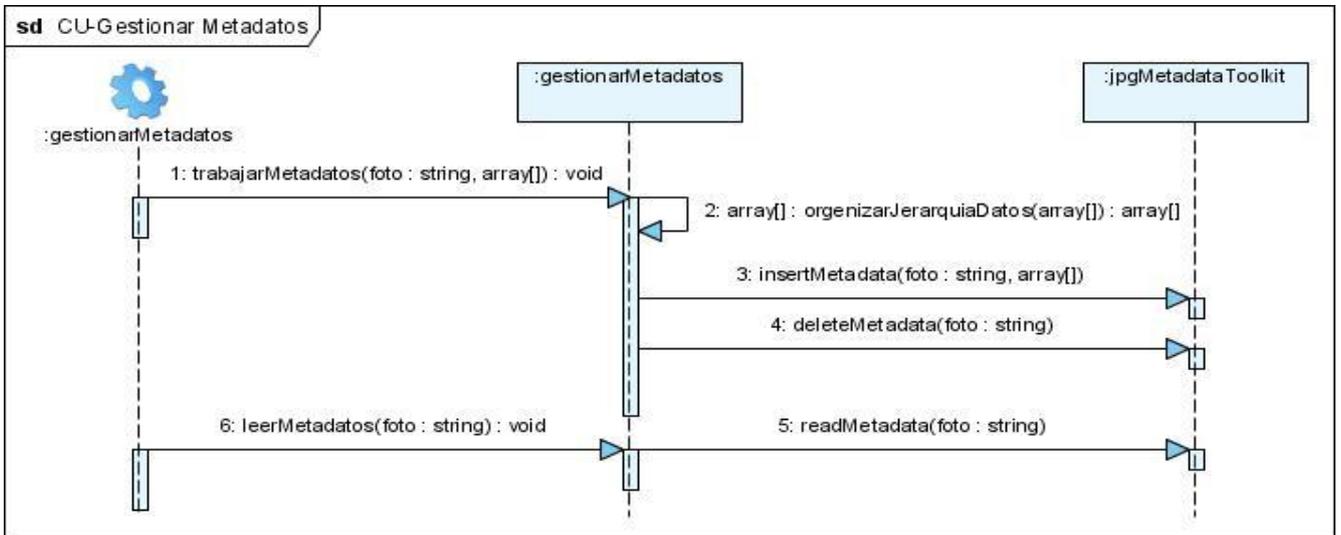
Anexo 21: Diagrama de secuencia del CU Gestionar usuario escenario eliminar rol.



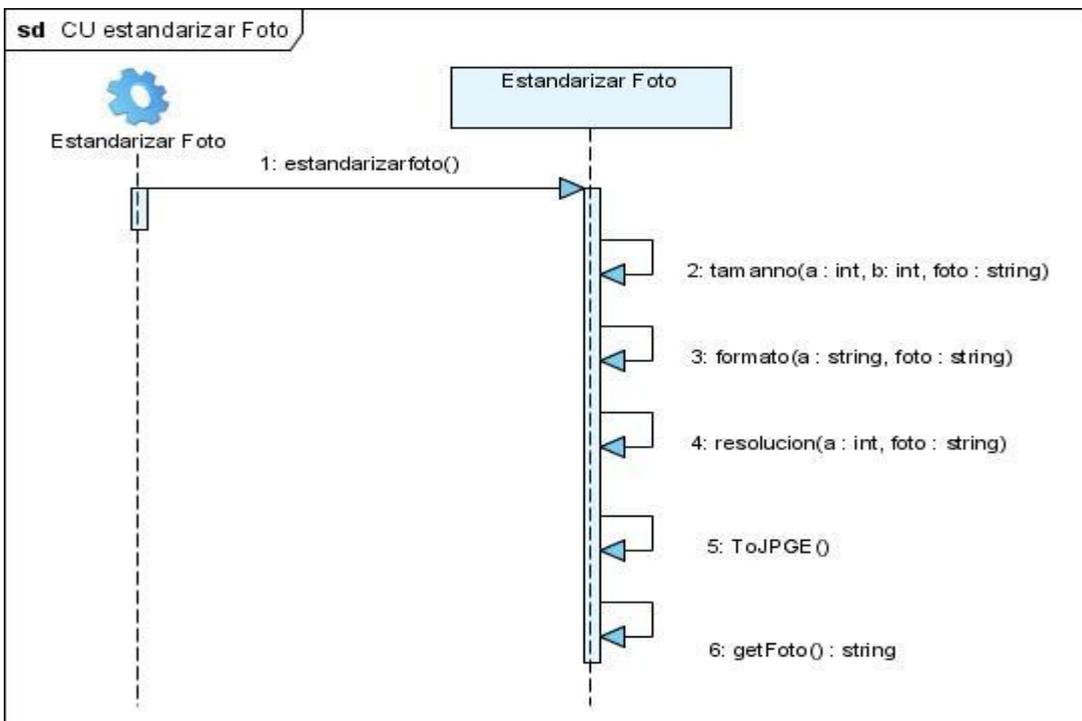
Anexo 22: Diagrama de secuencia del CU Leer metadato.



**Anexo 23: Diagrama de secuencia del CU Gestionar metadatos.**



**Anexo 24: Diagrama de secuencia del CU Estandarizar foto.**



## GLOSARIO

---

<sup>1</sup> Photoshop : Adobe Photoshop® (Ps) es una aplicación en forma de taller de pintura y fotografía que trabaja sobre un "*lienzo*" y que está destinado para la edición, retoque fotográfico y pintura a base de imágenes bitmap, jpeg, gif, etc, elaborada por la compañía de software Adobe Systems

<sup>2</sup> RDF, del acrónimo en inglés *Resource Description Framework*: Es un framework para metadatos en la World Wide Web (WWW), desarrollado por el World Wide Web Consortium (W3C).

<sup>3</sup> HTTP del acrónimo en inglés *HyperText Transfer Protocol*. El protocolo de transferencia de hipertexto es el protocolo usado en cada transacción de la web (WWW).

<sup>4</sup> W3C del acrónimo en inglés: *World Wide Web Consortium*. Es un consorcio internacional que produce estándares para la World Wide Web.

<sup>5</sup> Java: Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++ pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

<sup>6</sup> AFIS del acrónimo en inglés *Automated Fingerprint Identification System*: *Sistema automatizado de identificación de huellas dactilares*

<sup>7</sup> PACS del acrónimo en inglés *Picture Archiving and communication system*: sistema de archivo y transmisión de imágenes. Se trata de un sistema computarizado para el archivo digital de imágenes médica y para la transmisión de estas a estaciones de visualización dedicadas a través de una red informática.

<sup>8</sup> IDE, del acrónimo en inglés *Integrated Development Enviroment*: Entorno integrado para el desarrollo de aplicaciones.

<sup>9</sup> UML del acrónimo en inglés *Unified Modeling Language*: Lenguaje Unificado de Modelado.

<sup>10</sup> CASE del acrónimo en inglés *Computer Aided Software Engineering*: Ingeniería de Software Asistida por Ordenador.

<sup>11</sup> Interoperabilidad: Condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos.

<sup>12</sup> XML: del acrónimo en inglés *Extensible Markup Language*: Lenguaje de marcado extensible.

<sup>13</sup> Soap: del acrónimo en inglés *Simple Object Access Protocol*.

<sup>14</sup> SMTP: del acrónimo en inglés *Simple Mail Transfer Protocol*. El protocolo simple de transferencia de correo. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos (computador de mano, teléfonos móviles, etc.).

<sup>15</sup> MIME: del acrónimo en inglés *Multipurpose Internet Mail Extensions*. Extensiones de Correo Internet Multipropósito), son una serie de convenciones o especificaciones dirigidas a que se puedan intercambiar a través de Internet todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario.

<sup>16</sup> DCOM: del acrónimo en inglés *Distributed Component Object Model*. En español Modelo de Objetos de Componentes Distribuidos, es una tecnología propietaria de Microsoft para desarrollar componentes software distribuidos sobre varios ordenadores y que se comunican entre sí.

<sup>17</sup> UDP del acrónimo en inglés *User Datagram Protocol*. Es un protocolo del nivel de transporte basado en el intercambio de datagramas.

<sup>18</sup> PHP del acrónimo en inglés *Personal Home Page*: Es un lenguaje interpretado de alto nivel impregnado en páginas HTML y ejecutado en el servidor.

<sup>19</sup> GoF, del acrónimo en inglés *Gang of Four*. Es el nombre con el que se conoce a los autores del libro *Design Patterns*, referencia en el campo del diseño orientado a objetos. La 'Banda de los cuatro' se compone de los autores: Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.

<sup>20</sup> ORM, del acrónimo en inglés "*object-relational mappin*". *Herramienta* para el mapeo de objetos a bases de datos.

<sup>21</sup> CU: Caso(s) de uso(s).

<sup>22</sup> COCOMO del acrónimo en inglés *Constructive Cost Model*. Modelo constructivo de costes. Es un modelo de estimación de costes de software.