

# **Creación de una web de gestión de álbumes fotográficos utilizando tecnología Java EE**

**Francisco Fernández García**  
ETIS

**José Juan Rodríguez**  
14 de Enero de 2009

# Resumen

El presente trabajo de fin de carrera se plantea en base al análisis, planificación y desarrollo de una aplicación web basada en el modelo Java EE y que permitirá a los usuarios del sistema la creación de colecciones digitales de fotografías.

Java EE es una especificación que permite la creación de aplicaciones multicapa basada en componentes modulares que se ejecutan sobre un servidor de aplicaciones. El uso de este modelo arquitectónico basado en capas nos proporciona independencia y robustez y permite que cada una de las capas se centre en sus objetivos específicos, minimizando las interferencias recibidas por parte del resto de componentes de la aplicación.

Este proyecto se basará en un modelo de tres capas:

- La capa de presentación y control.
- La capa de negocio que controlará la lógica con la que se operan los datos.
- La capa que nos proporcionará el acceso a la información persistente.

Para la capa de presentación utilizaremos Struts2 que es un framework de aplicación open source desarrollado por Apache y que está basado en el patrón MVC (Modelo-Vista-Controlador).

En la capa de negocio usaremos Spring que es un framework también open source que nos proporciona un conjunto de módulos sobre los que desarrollar una aplicación Java EE.

Para la capa que gestionará los accesos a los datos de nuestra aplicación utilizaremos Hibernate que es una herramienta de software libre de mapeo objeto-relacional.

# Indice

<b>1</b>	<b>Introducción</b>	<b>6</b>
1.1	Justificación del TFC y contexto en el que se desarrolla	6
1.2	Objetivos del TFC	6
1.3	Enfoque y método seguido	6
1.4	Planificación del proyecto	7
1.5	Productos obtenidos	8
1.6	Resto de capítulos de la memoria	8
<b>2</b>	<b>Análisis</b>	<b>9</b>
2.1	Especificaciones del proyecto	9
2.2	Identificación de actores	10
2.3	Casos de uso	10
2.4	Diagramas de colaboración	15
2.5	Diagrama de clases	22
<b>3</b>	<b>Diseño</b>	<b>22</b>
3.1	Arquitectura del sistema	22
3.1.1	Java EE	22
3.1.2	Patrón arquitectónico: MVC	24
3.1.3	Frameworks	25
3.1.3.1	Struts2	25
3.1.3.2	Spring	27
3.1.3.3	Hibernate	28
3.2	Diseño de la persistencia	29
3.2.1	Diagrama E-R	29
3.3	Interfaz de usuario	30
<b>4</b>	<b>Implementación</b>	<b>54</b>
4.1	Estructura de la aplicación	55
4.2	Almacenamiento en el servidor	56
4.3	Envío de e-mails	56
<b>5</b>	<b>Requerimientos</b>	<b>57</b>
<b>6</b>	<b>Instalación</b>	<b>57</b>
<b>7</b>	<b>Valoración económica</b>	<b>57</b>
<b>8</b>	<b>Conclusiones</b>	<b>58</b>
<b>9</b>	<b>Bibliografía</b>	<b>58</b>

# Indice de figuras

Figura 1 - Planificación del proyecto .....	8
Figura 2 - Casos de uso.....	10
Figura 3 - D.C. Alta usuario.....	16
Figura 4 - D.C. Conexión al sistema .....	16
Figura 5 - D.C. Modificación perfil.....	17
Figura 6 - D.C. Alta álbum.....	17
Figura 7 - D.C. Lista de usuarios .....	17
Figura 8 - D.C. Borrar usuario .....	18
Figura 9 - D.C. Modificación usuario.....	18
Figura 10 - D.C. Lista de álbumes .....	19
Figura 11 - D.C. Borrar álbum .....	19
Figura 12 - D.C. Enviar invitación .....	20
Figura 13 - D.C. Modificar álbum .....	20
Figura 14 - D.C. Lista de fotos .....	20
Figura 15 - D.C. Borrar foto.....	20
Figura 16 - D.C. Modificar foto .....	21
Figura 17 - D.C. Ver foto.....	21
Figura 18 - D.C. Alta foto.....	21
Figura 19 - D.C. Enviar e-mail .....	22
Figura 20 - Diagrama de clases.....	22
Figura 21 - Arquitectura Java EE.....	24
Figura 22 - Patrón MVC.....	25
Figura 23 - Esquema Struts2 .....	26
Figura 24 - Esquema Struts2-MVC.....	27
Figura 25 - Esquema Spring .....	28
Figura 26 - Esquema Hibernate.....	28
Figura 27 - Diagrama E-R.....	29
Figura 28 - Diseño Base de Datos .....	29

Figura 29 - Pantalla: Home .....	31
Figura 30 - Pantalla: Registro Cliente .....	32
Figura 31 - Pantalla: Registro Cliente OK.....	33
Figura 32 - Pantalla: Conexión .....	34
Figura 33 - Pantalla: Lista de álbumes .....	35
Figura 34 - Pantalla: Modificación usuario .....	36
Figura 35 - Pantalla: Alta de álbum .....	37
Figura 36 - Pantalla: Lista de fotos (propietario) .....	38
Figura 37 - Pantalla: Lista de fotos (invitado).....	39
Figura 38 - Pantalla: Modificación álbum.....	40
Figura 39 - Pantalla: Baja de álbum .....	41
Figura 40 - Pantalla: Invitar a álbum .....	42
Figura 41 - Pantalla: Eliminar invitación .....	43
Figura 42 - Pantalla: Alta de foto .....	44
Figura 43 - Pantalla: Consulta de foto de álbum propio .....	45
Figura 44 - Pantalla: Consulta de foto de álbum invitado .....	46
Figura 45 - Pantalla: Modificación de foto.....	47
Figura 46 - Pantalla: Baja de foto .....	48
Figura 47 - Pantalla: Home de administrador.....	49
Figura 48 - Pantalla: Alta de administrador.....	50
Figura 49 - Pantalla: Lista de usuarios .....	51
Figura 50 - Pantalla: Modificación de usuario (por administrador) .....	52
Figura 51 - Pantalla: Baja de usuario.....	53
Figura 52 - Pantalla: Lista de álbumes (para administrador) .....	54

# 1 Introducción

## 1.1 Justificación del TFC y contexto en el que se desarrolla

Se plantea este TFC como una forma de aplicar los conceptos aprendidos a lo largo de la carrera de ETIS, de manera que se puedan ensamblar los fragmentos de conocimiento proporcionados por las diferentes asignaturas cursadas a la hora de realizar un proyecto.

Al ser el área tratada en el presente TFC la **arquitectura Java EE** el resultado del proyecto deberá ser una aplicación basada en dicha arquitectura, para lo cual pasaremos por las diferentes fases que conforman cualquier proyecto de este tipo: análisis, diseño e implementación del producto final.

El punto de partida del proyecto serán, como se ha comentado antes, los conocimientos que se han obtenido a lo largo de la carrera y que básicamente se podrán aplicar a las etapas de análisis y diseño. Estos tendrán que ser complementados con el estudio de las tecnologías directamente relacionadas con la arquitectura Java EE y que serán los que se deberán aplicar directamente en la etapa de implementación del proyecto.

## 1.2 Objetivos del TFC

El objetivo de este TFC es realizar por mi parte un primer acercamiento a la tecnología Java EE para conseguir familiarizarme, aunque sea de forma mínima, con los recursos y herramientas utilizadas en el desarrollo de este tipo de proyectos.

Esto hace que no sea en absoluto prioritario el acabar consiguiendo una herramienta que pudiera llegar a ser explotada comercialmente por lo que, por ejemplo, no se pondrá especial énfasis en puntos como puede ser la estética final de las pantallas o la seguridad y si que se incidirá más en el diseño de la aplicación y en su implementación intentando ajustarla a los patrones establecidos.

A grandes rasgos la aplicación que vamos a construir tiene que permitir a un usuario registrado el subir sus fotografías a un servidor y organizarlas en colecciones siguiendo los criterios que considere oportunos.

La aplicación también permitirá al propietario de una colección el autorizar a otros usuarios para que éstos puedan visualizar sus colecciones.

En el sistema tendremos la figura del administrador, que tendrá la capacidad de eliminar fotografías o colecciones consideradas inadecuadas y de gestionar los usuarios registrados (baja de usuarios, restauración de passwords,...).

## 1.3 Enfoque y método seguido

Dado el desconocimiento de las tecnologías relacionadas con la arquitectura Java EE que se han utilizado en la etapa de implementación, el enfoque aplicado en el desarrollo del proyecto ha pasado necesariamente por un estudio de dichas tecnologías en paralelo con la realización de las fases de análisis y diseño.

La intención de este planteamiento es reducir al máximo el impacto negativo sobre el proyecto de dicho desconocimiento y así poder llegar a la fase de implementación con una base de conocimientos que permita afrontar esta etapa con alguna garantía de que se podrá conseguir un producto final con un mínimo de calidad.

Dentro de lo que es propiamente el desarrollo del proyecto, tendremos cuatro fases que vendrán marcadas por la realización de las diferentes PAC's:

- Planificación
- Análisis y diseño
- Implementación
- Memoria final

En la etapa de planificación se realiza un primer esbozo de la aplicación a desarrollar y se distribuyen las diferentes tareas de las que constará el proyecto.

En la etapa de análisis se acabarán de cerrar los requerimientos que deberá cumplir la aplicación, especificando todos los casos de uso a los que deberá dar soporte e identificando a los actores que intervendrán. Acabado este punto debemos tener una imagen clara de lo que esperamos obtener una vez finalizado el proyecto y que deberá estar reflejado en el diseño posterior.

En la etapa de diseño se identificarán las entidades que intervendrán y se formalizarán los casos de uso identificados en el paso anterior. También se realizará un diseño de las interfaces finales con el usuario.

En la etapa de implementación se codificarán los componentes necesarios y se realizarán las pruebas que aseguren una mínima calidad del producto final obtenido y comprobando que realmente se cumplen las especificaciones iniciales.

Finalmente, la última parte del proyecto se dedicará a pulir las deficiencias que puedan ir apareciendo para obtener el producto ya definitivo y a redactar la presente memoria del proyecto.

## **1.4 Planificación del proyecto**

Como se ha comentado en el punto anterior la planificación realizada se basa en las fechas de entrega de las diferentes PAC's.

También hay que hacer notar que durante todo el transcurso del proyecto las tareas propias de éste se tendrán que simultanear con la instalación y estudio de las diferentes herramientas que serán necesarias para su implementación final, lo que sin duda a sido lo más costoso del proyecto.

Teniendo en cuenta todo lo anterior la planificación prevista queda reflejada en el siguiente gráfico:

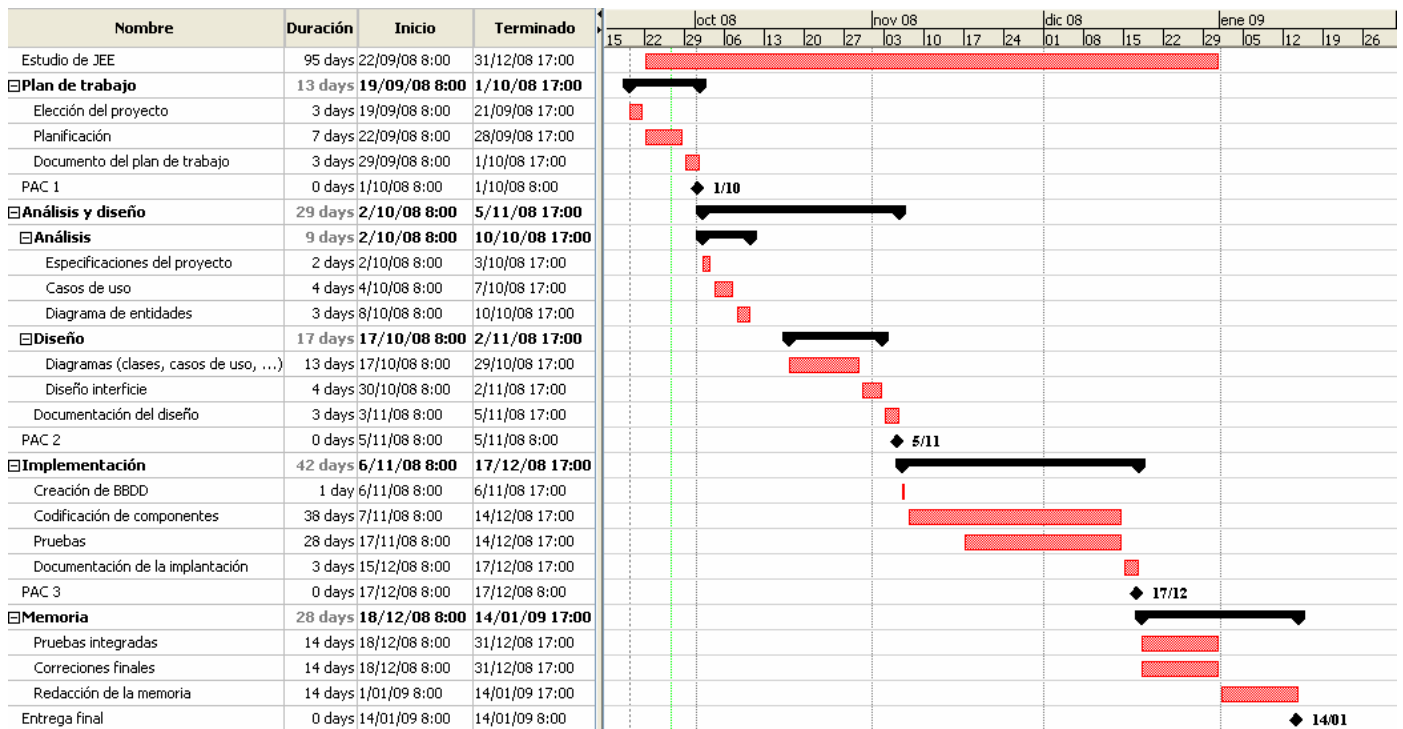


Figura 1 - Planificación del proyecto

## 1.5 Productos obtenidos

Los productos finales obtenidos durante las diferentes fases del proyecto se pueden dividir en dos partes, la documentación generada en las primeras fases del proyecto y el software conseguido en la fase de implementación:

- **Planificación:** Diagrama de Gantt con el desglose y la temporización de las diferentes tareas a realizar en el proyecto (incluido en esta memoria).
- **Análisis:** Identificación de actores y descripción de casos de uso. Diagramas de colaboración y de clases (todo ello incluido en esta memoria).
- **Diseño:** Descripción de la arquitectura de la aplicación y diseño de la persistencia y de la interfaz de usuario (todo ello incluido en esta memoria).
- **Implementación:** Archivo WAR con la aplicación preparada para ser desplegada en un servidor y el correspondiente código fuente. Script SQL para la generación de la Base de Datos utilizada por la aplicación.
- **Memoria:** Memoria del proyecto incluyendo toda la documentación generada y presentación del trabajo realizado.

## 1.6 Resto de capítulos de la memoria

En los siguientes capítulos de la memoria se desarrollan las diferentes fases del proyecto objeto de este TFC

En primer lugar tenemos la fase de análisis en la que, partiendo de las especificaciones iniciales del proyecto, se hace una descripción de los actores que intervendrán, la descripción de los casos de uso identificados en el sistema y finalmente se estudian las colaboraciones



entre los diferentes casos de uso para identificar las clases gestoras, frontera y de entidad que conformarán nuestro sistema.

A continuación se detalla la fase de diseño en la que se hace una descripción de la arquitectura empleada, así como de las herramientas utilizadas como infraestructura básica en la construcción de la aplicación. También se muestra el diseño de la persistencia y la interfaz de usuario a construir.

Por último tenemos los capítulos que describen algunos de los aspectos más destacables de la implementación realizada y los requerimientos mínimos del sistema.

## 2 Análisis

### 2.1 Especificaciones del proyecto

Como ya se ha comentado con anterioridad este TFC consistirá en la creación de una web que permita a los usuarios registrados la creación de colecciones de fotografías almacenadas por el propio usuario en el servidor.

Los usuarios podrán agrupar las fotografías en álbumes en función de sus necesidades o preferencias y el propietario de uno de estos álbumes también tendrá la posibilidad de compartirlo con otros usuarios del sistema mediante el envío de invitaciones.

En el sistema también tendremos la figura del administrador, que tendrá la capacidad de eliminar colecciones de fotografías si, por ejemplo, se consideran inadecuadas y de gestionar los usuarios registrados (baja de usuarios, restauración de passwords,...).

A grandes rasgos la aplicación final debería cumplir con los siguientes **requisitos**:

- Sólo podrán acceder a este servicio usuarios registrados, por lo que será necesario crear un proceso de alta que permita registrarse a los clientes y un proceso de autenticación que verificará el usuario que intenta acceder.
- Los usuarios podrán ser de dos tipos: administradores y clientes y tendrán roles claramente diferentes lo que se verá reflejado en el proceso de alta. El alta de un administrador deberá hacerla necesariamente otro administrador, mientras que la de un cliente la realizará el propio usuario.
- Un administrador no podrá tener colecciones propias de fotografías, pero tendrá privilegios para eliminar del sistema cualquier colección o usuario y para realizar algunas tareas de mantenimiento sobre éstos modificando algunas de sus características (por ejemplo la password de acceso).
- Los clientes podrán subir fotografías al servidor siempre ligadas a alguna de sus colecciones y se le permitirá gestionar dichas colecciones añadiendo o eliminando fotografías o añadiendo comentarios a las fotografías.
- El propietario de una colección podrá compartirla con otros clientes también registrados, los cuales sólo podrán visualizar estas colecciones pero no modificarlas de ninguna manera. No se podrán realizar invitaciones a usuarios que tengan el perfil de administrador por lo que se deberá realizar un filtro de los usuarios susceptibles de ser invitados. Una vez realizada una invitación se le enviará un e-mail de notificación al usuario invitado.



### ❖ Alta usuario

- **Funcionalidad:** Dar de alta un nuevo usuario en el sistema.
- **Actores:** Administrador, Cliente
- **Casos de uso relacionados:** Grabar log
- **Precondición:** El nuevo usuario no existe en el sistema.
- **Postcondición:** Se ha dado de alta el usuario en el sistema. Se ha grabado un registro en el log del sistema.
- **Descripción:** Permite realizar el alta de un nuevo usuario administrador o cliente en el sistema. Un usuario administrador lo tiene que dar de alta otro usuario registrado como administrador.

### ❖ Conexión al sistema

- **Funcionalidad:** Permite que un usuario se conecte al sistema.
- **Actores:** Administrador, Cliente
- **Casos de uso relacionados:** Grabar log
- **Precondición:** El usuario debe estar dado de alta en el sistema, bien como administrador o como cliente.
- **Postcondición:** El usuario se ha conectado en el sistema. Se ha grabado un registro en el log del sistema.
- **Descripción:** Permite que un usuario se conecte al sistema de forma que ya podrá realizar las operaciones asignadas según su rol.

### ❖ Modificación perfil

- **Funcionalidad:** Permite que un usuario modifique las características de su perfil.
- **Actores:** Administrador, Cliente
- **Casos de uso relacionados:** Grabar log
- **Precondición:** El usuario debe estar identificado en el sistema.
- **Postcondición:** El usuario ha modificado los datos de su perfil. Se ha grabado un registro en el log del sistema.
- **Descripción:** Permite que un usuario identificado en el sistema pueda modificar la información relacionada con dicho usuarios, como puede ser el e-mail, la password, etc.

### ❖ Alta de álbum

- **Funcionalidad:** Permite que un cliente cree un nuevo álbum de fotos.
- **Actores:** Cliente
- **Casos de uso relacionados:** Grabar log
- **Precondición:** El usuario debe estar identificado en el sistema y su perfil debe ser de cliente.

- **Postcondición:** El usuario ha creado un nuevo álbum. Se ha grabado un registro en el log del sistema.
- **Descripción:** Permite que un usuario identificado en el sistema cree nuevos álbumes de fotos.

### ❖ Lista de usuarios

- **Funcionalidad:** Muestra una lista de usuarios del sistema.
- **Actores:** Administrador
- **Casos de uso relacionados:** Modificar usuario, Borrar usuario, Lista álbumes
- **Precondición:** El usuario debe estar identificado como administrador en el sistema.
- **Postcondición:** Ninguna
- **Descripción:** Muestra a un administrador la lista de usuarios del sistema para su gestión. A partir de esta lista el administrador podrá realizar las acciones disponibles.

### ❖ Borrar usuario

- **Funcionalidad:** Borra un usuario del sistema.
- **Actores:** Administrador
- **Casos de uso relacionados:** Lista usuarios, Borrar álbum, Enviar e-mail, Grabar log
- **Precondición:** El usuario que realiza la acción debe estar identificado como administrador en el sistema.
- **Postcondición:** El usuario se ha borrado del sistema, junto con los álbumes y las fotos que tuviera en caso de que el usuario borrado fuera cliente. Se ha grabado un registro en el log del sistema.
- **Descripción:** El administrador selecciona uno de los usuarios del sistema y lo borra completamente. En caso de que el usuario sea cliente se borran también los álbumes que pudiera poseer y las fotos que éstos contuviera. Se envía además un e-mail al usuario borrado para avisarle de la baja.

### ❖ Modificación usuario

- **Funcionalidad:** Modifica un usuario del sistema.
- **Actores:** Administrador
- **Casos de uso relacionados:** Lista usuarios, Grabar log
- **Precondición:** El usuario que realiza la acción debe estar identificado como administrador en el sistema.
- **Postcondición:** Se han modificado los datos del usuario. Se ha grabado un registro en el log del sistema.
- **Descripción:** El administrador selecciona uno de los usuarios del sistema y modifica alguna de sus características como pueden ser el e-mail.

### ❖ Lista de álbumes

- **Funcionalidad:** Muestra la lista de los álbumes de un usuario.
- **Actores:** Administrador, Cliente
- **Casos de uso relacionados:** Lista usuarios, Enviar invitación, Borrar álbum, Lista fotos, Modificar álbum
- **Precondición:** El usuario que realiza la acción debe estar identificado en el sistema.
- **Postcondición:** Ninguna.
- **Descripción:** Si el usuario identificado es Cliente se le mostrará la lista de los álbumes que le pertenezcan. En caso de que sea Administrador tendrá que haber elegido previamente un usuario Cliente de la lista de usuarios.

### ❖ Borrar álbum

- **Funcionalidad:** Borra un álbum del sistema.
- **Actores:** Administrador, Cliente
- **Casos de uso relacionados:** Lista álbumes, Borrar usuario, Enviar e-mail, Borrar foto, Grabar log
- **Precondición:** El usuario que realiza la acción debe estar identificado en el sistema.
- **Postcondición:** El álbum se ha borrado del sistema, junto con fotos que tuviera. Se ha grabado un registro en el log del sistema.
- **Descripción:** Se borra del sistema un álbum perteneciente a un cliente, junto con las fotos que colgaran de él.

### ❖ Enviar invitación

- **Funcionalidad:** Permite a otro usuario del sistema el ver un álbum.
- **Actores:** Cliente
- **Casos de uso relacionados:** Lista álbumes, Lista usuarios, Enviar e-mail, Grabar log
- **Precondición:** El usuario que realiza la acción debe estar identificado en el sistema.
- **Postcondición:** Se ha grabado un registro en el log del sistema.
- **Descripción:** Se ha enviado un e-mail al usuario invitado notificándole dicha invitación.

### ❖ Modificar álbum

- **Funcionalidad:** Permite a un usuario del sistema el modificar un álbum de su propiedad.
- **Actores:** Cliente
- **Casos de uso relacionados:** Lista álbumes, Grabar log
- **Precondición:** El usuario que realiza la acción debe estar identificado como Cliente en el sistema.
- **Postcondición:** Se han modificado las características del álbum. Se ha grabado un registro en el log del sistema.
- **Descripción:** Permite a un usuario Cliente el modificar la información de uno de sus álbumes.

### ❖ Lista fotos

- **Funcionalidad:** Muestra la lista de las fotos de un álbum.
- **Actores:** Cliente
- **Casos de uso relacionados:** Lista álbumes, Borrar foto, Ver foto, Alta foto, Modificar foto
- **Precondición:** El usuario que realiza la acción debe estar identificado como Cliente en el sistema.
- **Postcondición:** Ninguna.
- **Descripción:** Se muestra la lista de fotos relacionadas con un álbum del cliente.

### ❖ Borrar foto

- **Funcionalidad:** Permite a un usuario del sistema borrar una foto.
- **Actores:** Administrador, Cliente
- **Casos de uso relacionados:** Lista fotos, Borrar álbum, Grabar log
- **Precondición:** El usuario que realiza la acción debe estar identificado en el sistema.
- **Postcondición:** La foto se ha borrado del sistema. Se ha grabado un registro en el log del sistema.
- **Descripción:** Permite borrar una foto de un álbum del sistema.

### ❖ Modificar foto

- **Funcionalidad:** Permite a un usuario del sistema modificar la información de una foto.
- **Actores:** Cliente
- **Casos de uso relacionados:** Lista fotos, Grabar log
- **Precondición:** El usuario que realiza la acción debe estar identificado como Cliente en el sistema.
- **Postcondición:** Se ha modificado la información de la foto. Se ha grabado un registro en el log del sistema.
- **Descripción:** Permite modificar la información de una foto de un álbum del sistema.

### ❖ Ver foto

- **Funcionalidad:** Permite a un usuario del sistema ver una de las fotos almacenadas.
- **Actores:** Cliente
- **Casos de uso relacionados:** Lista fotos
- **Precondición:** El usuario que realiza la acción debe estar identificado como Cliente en el sistema.
- **Postcondición:** Ninguna.
- **Descripción:** Permite ver una de las fotos del sistema.

### ❖ Alta foto

- **Funcionalidad:** Permite a un usuario del sistema añadir una nueva foto a un álbum.
- **Actores:** Cliente
- **Casos de uso relacionados:** Lista fotos, Grabar log
- **Precondición:** El usuario que realiza la acción debe estar identificado como Cliente en el sistema.
- **Postcondición:** Se ha insertado la foto en el sistema. Se ha grabado un registro en el log del sistema.
- **Descripción:** Permite a un Cliente el añadir una nueva foto a uno de sus álbumes.

### ❖ Enviar e-mail

- **Funcionalidad:** Permite que el sistema envíe un e-mail a uno de sus usuarios.
- **Actores:** Administrador, Cliente
- **Casos de uso relacionados:** Borrar usuario, Borrar álbum, Enviar invitación
- **Precondición:** El usuario que realiza la acción debe estar identificado en el sistema.
- **Postcondición:** Ninguna.
- **Descripción:** Realiza el envío de un e-mail a un usuario avisándole de alguna circunstancia, como puede ser una invitación para compartir un álbum o la baja de uno de sus álbumes.

### ❖ Grabar log

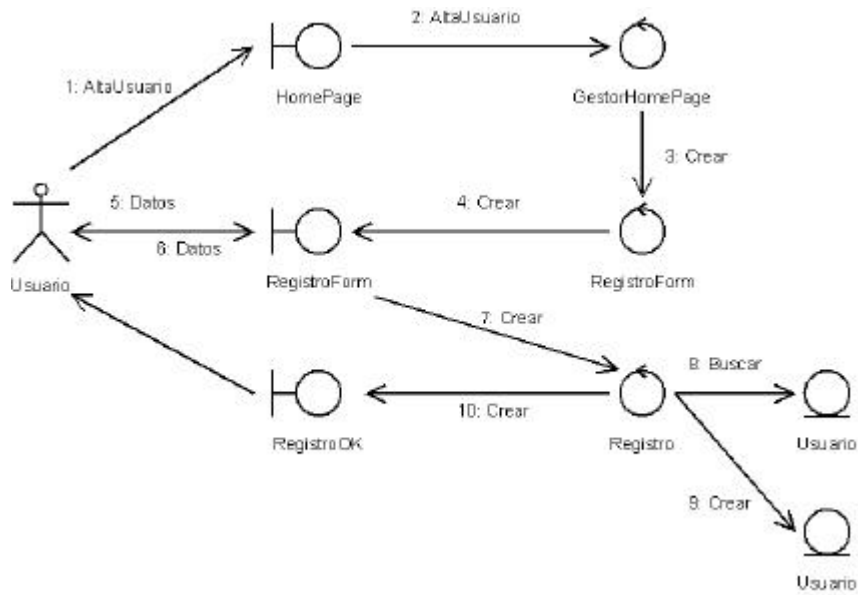
- **Funcionalidad:** Permite grabar un registro cada vez que se realicen determinadas operaciones en el sistema.
- **Actores:** Administrador, Cliente
- **Casos de uso relacionados:** Alta usuario, Conexión al sistema, Modificación perfil, Alta álbum, Borrar usuario, Modificación usuario, Borrar álbum, Enviar invitación, Modificación álbum, Borrar foto, Modificación foto, Alta foto
- **Precondición:** Ninguna.
- **Postcondición:** Ninguna.
- **Descripción:** Graba un registro en el log del sistema dejando constancia de las modificaciones realizadas.

## 2.4 Diagramas de colaboración

En los siguientes diagramas de colaboración se representan las interacciones que se producirán en cada uno de los casos de uso, lo que nos permitirá identificar las clases gestoras, frontera y de entidad.

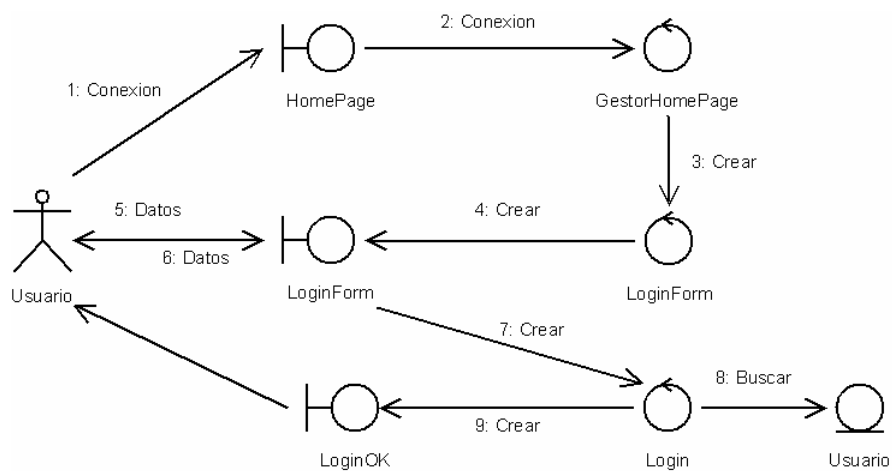
Igual que en el diagrama de casos de uso, en los diagramas de colaboración tampoco se incluye la grabación de registro en el log del sistema cada vez que se realice una operación que necesite dejar constancia.

- **Alta usuario**



**Figura 3- D.C. Alta usuario**

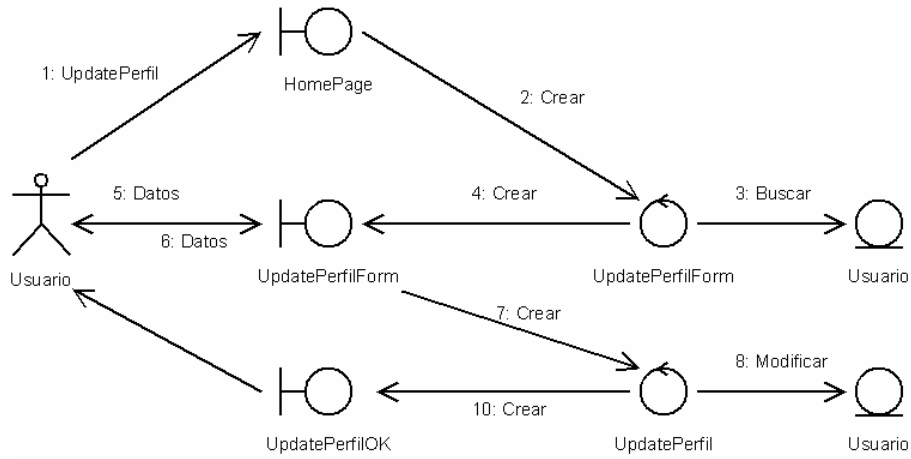
- **Conexión al sistema**



**Figura 4 - D.C. Conexión al sistema**

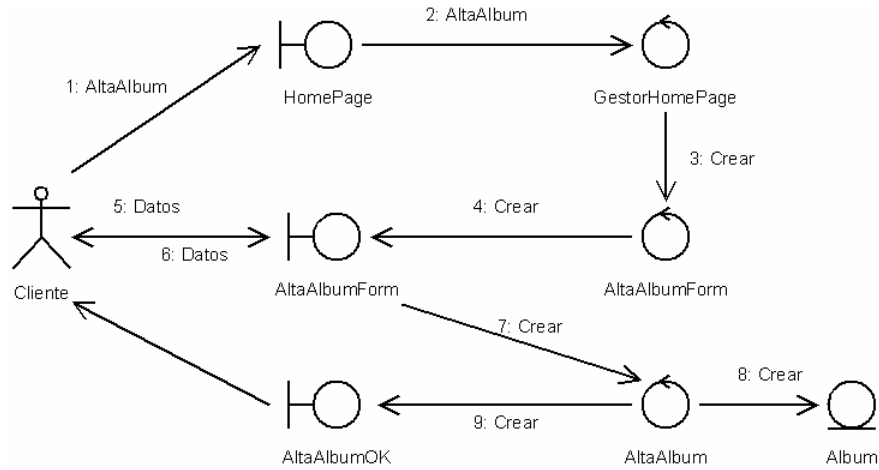


- **Modificación perfil**



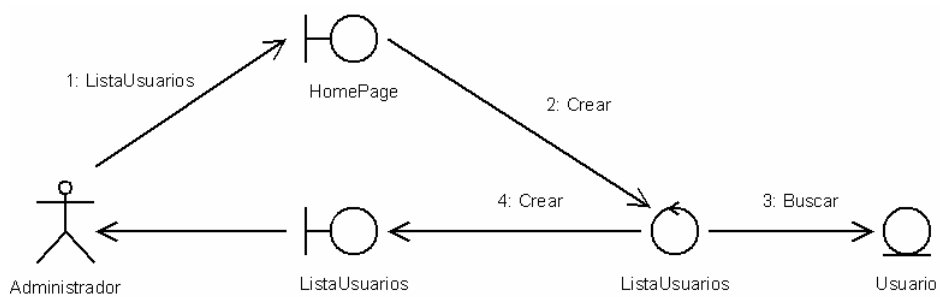
**Figura 5 - D.C. Modificación perfil**

- **Alta de álbum**



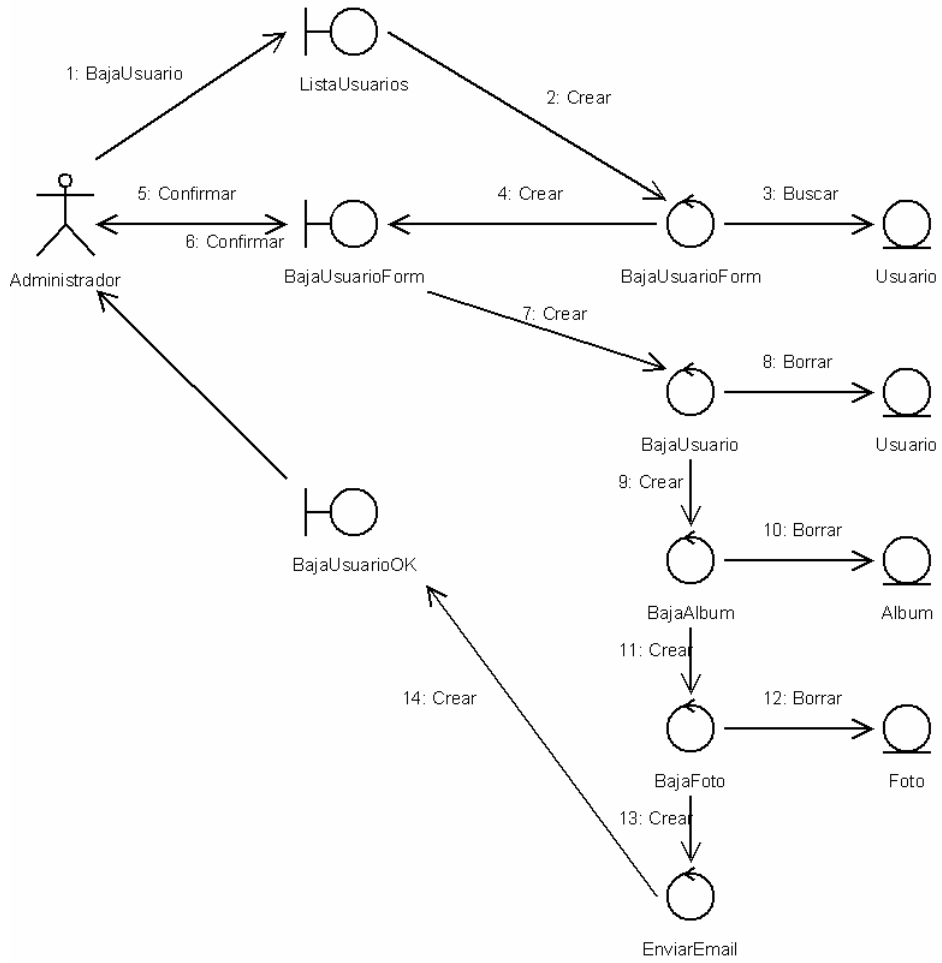
**Figura 6 - D.C. Alta álbum**

- **Lista de usuarios**



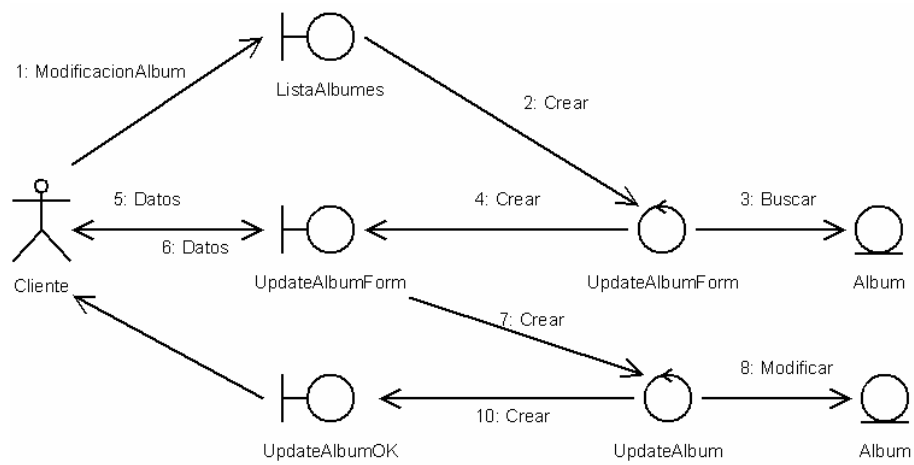
**Figura 7 - D.C. Lista de usuarios**

• **Borrar usuario**



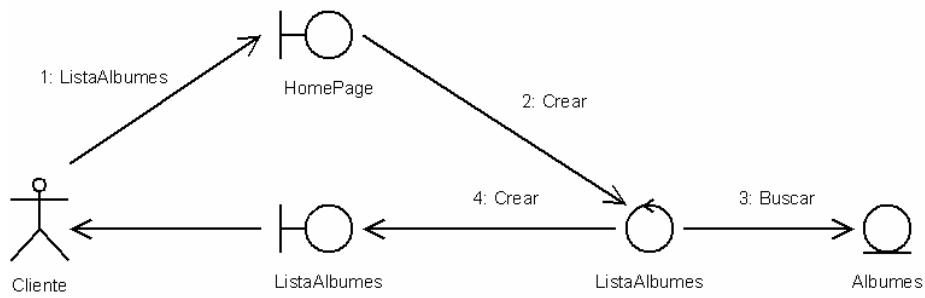
**Figura 8 - D.C. Borrar usuario**

• **Modificación usuario**



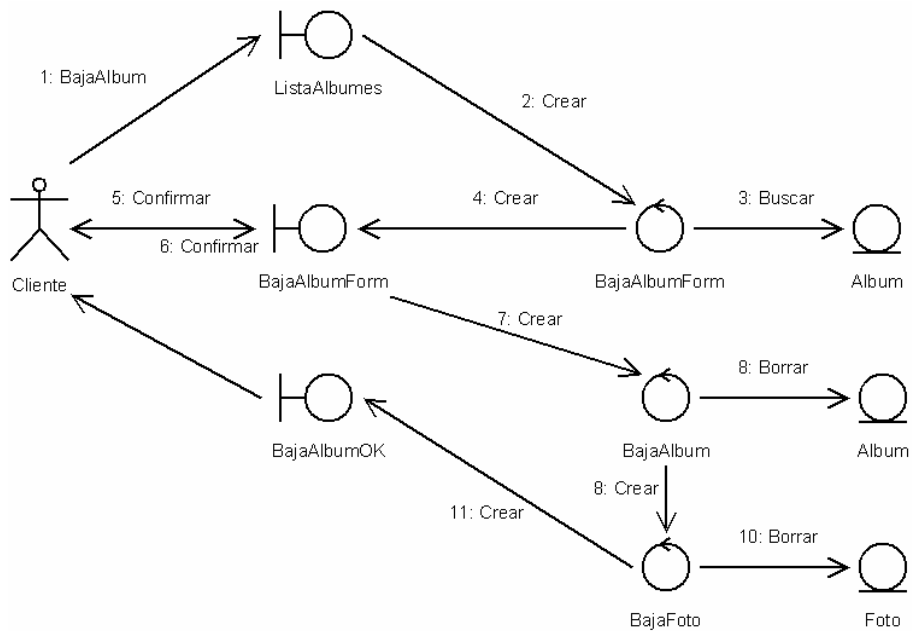
**Figura 9 - D.C. Modificación usuario**

- **Lista de álbumes**



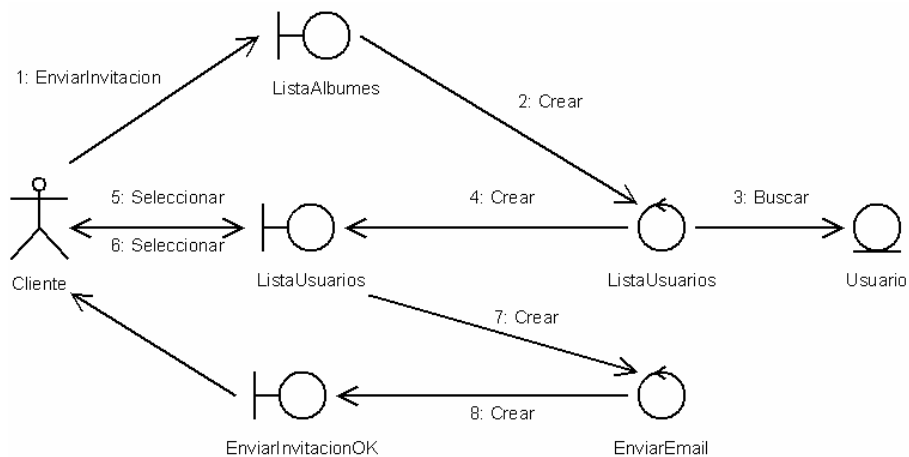
**Figura 10 - D.C. Lista de álbumes**

- **Borrar álbum**



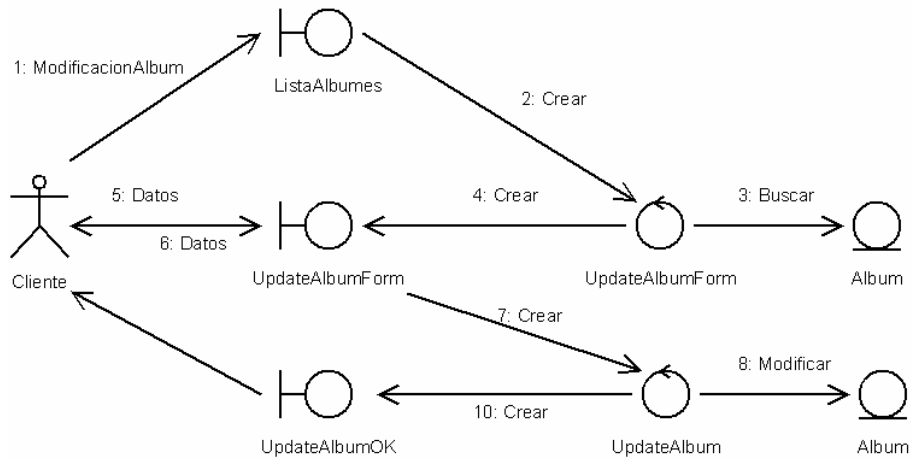
**Figura 11 - D.C. Borrar álbum**

- **Enviar invitación**



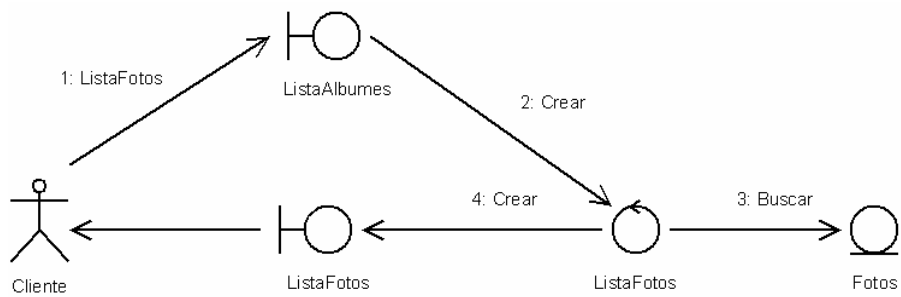
**Figura 12 - D.C. Enviar invitación**

• **Modificar álbum**



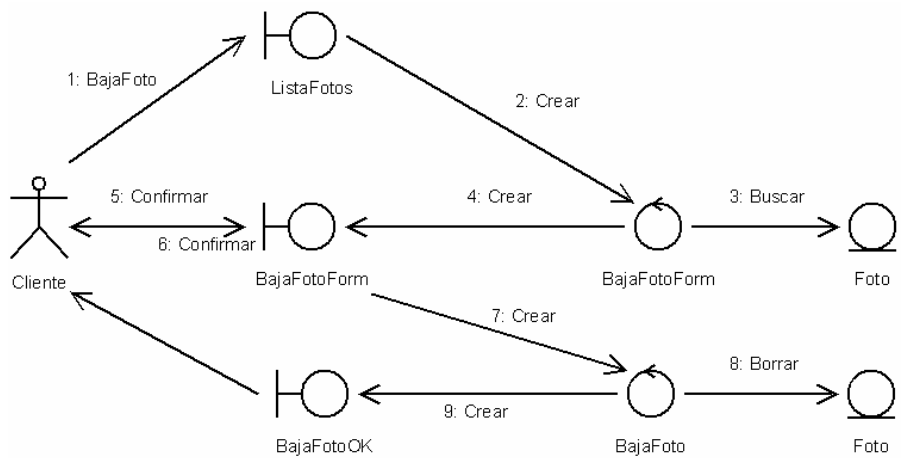
**Figura 13 - D.C. Modificar álbum**

• **Lista de fotos**



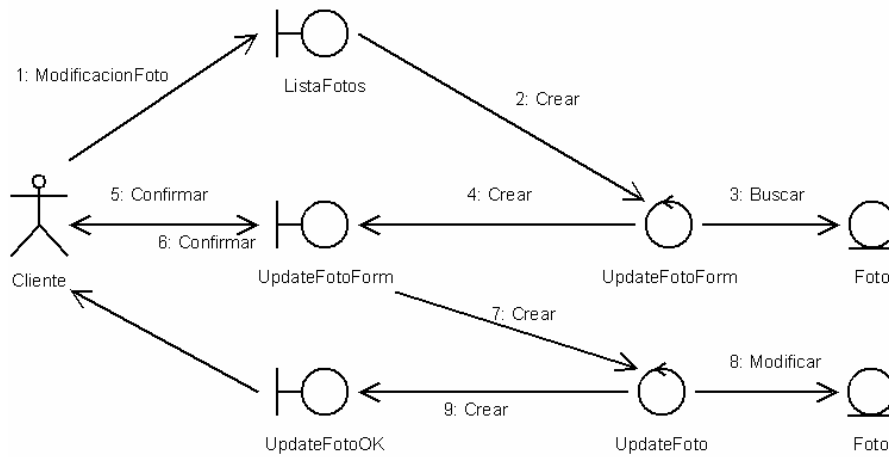
**Figura 14 - D.C. Lista de fotos**

• **Borrar foto**



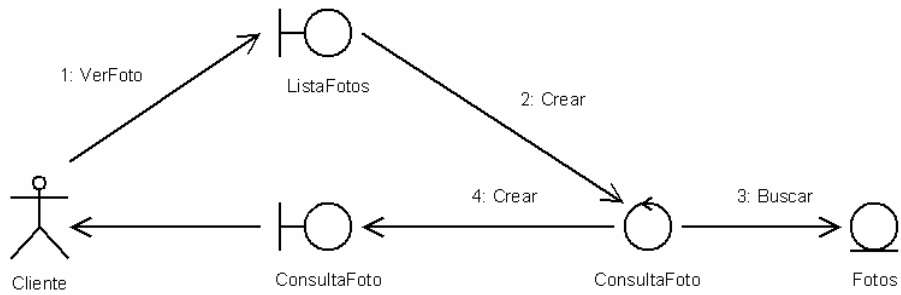
**Figura 15 - D.C. Borrar foto**

- **Modificar foto**



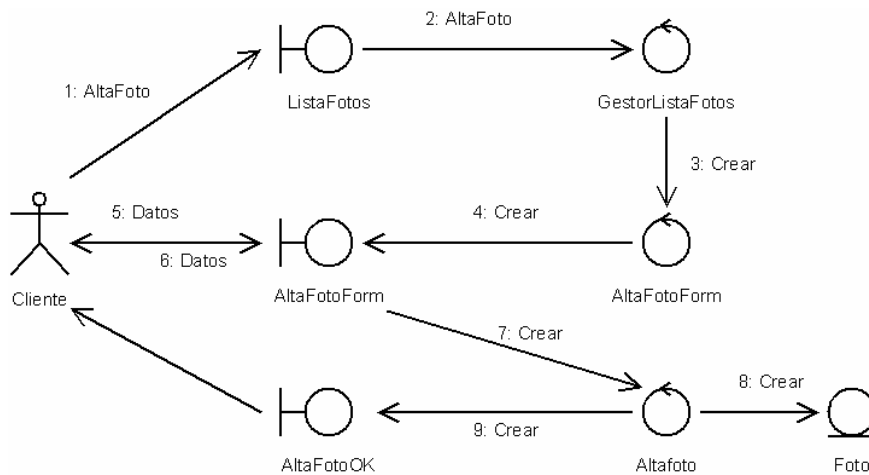
**Figura 16 - D.C. Modificar foto**

- **Ver foto**



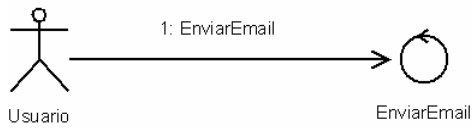
**Figura 17 - D.C. Ver foto**

- **Alta foto**



**Figura 18 - D.C. Alta foto**

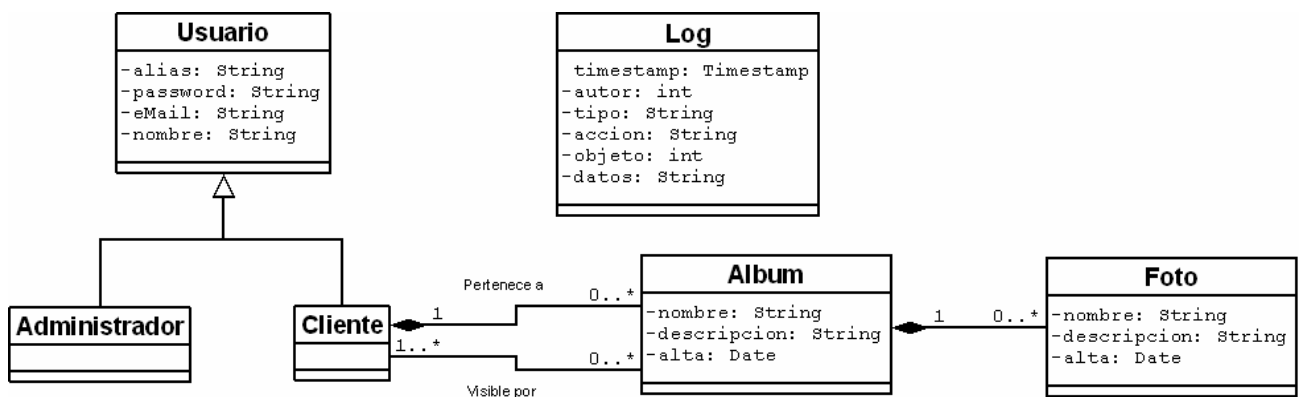
- **Enviar e-mail**



**Figura 19 - D.C. Enviar e-mail**

## 2.5 Diagrama de clases

El diagrama de clases del sistema sería el siguiente:



**Figura 20 - Diagrama de clases**

Las clases Administrador y Cliente representan a los dos actores que tenemos en el sistema. Ambos tienen características comunes por lo que los podemos hacer heredar de la clase Usuario que englobaría dichas características.

En el diagrama se refleja el hecho de que una foto sólo puede pertenecer a un único álbum, que éste sólo puede tener un único propietario y que no podemos tener en el sistema una foto o un álbum que no tengan un propietario.

Entre las clases Cliente y Album hay además otra relación ya que un álbum podrá ser visitado por usuarios diferentes a su propietario, siempre que éstos hayan sido invitados.

## 3 Diseño

### 3.1 Arquitectura del sistema

#### 3.1.1 Java EE

**Java EE** es una plataforma creada por Sun y cuyo objetivo es el desarrollo de aplicaciones distribuidas dirigidas principalmente a la empresa. Entre sus requisitos están la fiabilidad, la facilidad de mantenimiento, la escalabilidad, etc.

Esta plataforma está basado en **Java** que es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystem a principios de los años 90 y cuyos principales objetivos eran:

- Uso de la metodología orientada a objetos.
- Portabilidad del mismo programa en múltiples sistemas operativos.
- Incluir por defecto soporte para trabajo en red.
- Permitir la ejecución de código en sistemas remotos de forma segura.
- Facilidad de uso tomando elementos de otros lenguajes orientados a objetos.

La **portabilidad** entre plataformas que proporciona este lenguaje viene dada por el hecho de que al compilar el código fuente escrito en Java se genera un código conocido como "bytecode" que es ejecutado por lo que se conoce como la máquina virtual de Java. Esta máquina virtual es un programa escrito en código nativo de la plataforma y es el encargado de interpretar y ejecutar el código, es ella la que conoce el hardware sobre el que se ejecuta la aplicación y la que actúa como intermediaria entre ambas.

Esta portabilidad de Java hace que una aplicación Java EE también se pueda instalar fácilmente en cualquier entorno que soporte dicho lenguaje.

La especificación Java EE define un modelo de **capas** mediante el cual la lógica de la aplicación se divide en componentes de acuerdo con su función. Cada uno de estos componentes se puede acabar instalando en una máquina diferente (aunque no necesariamente) dependiendo de la capa a la que pertenezca.

Las capas en las que se divide el sistema son las siguientes:

- **Capa cliente:** Se ejecuta en la máquina cliente. Es la que permite al usuario interactuar con el sistema y pueden ser clientes ligeros, como un navegador web o bien clientes pesados, como una aplicaciones de escritorio.
- **Capa web:** Se ejecuta en un servidor Java EE. Es la encargada de obtener datos del cliente y de solicitar a la capa de negocio las operaciones necesarias.
- **Capa de negocio:** Se ejecuta en un servidor Java EE y forma el núcleo de la aplicación. En esta capa estarán representadas nuestras entidades, relaciones y reglas que implementarán nuestros procesos de negocio.
- **Capa EIS (Enterprise Information System):** Se ejecuta en un servidor de base de datos y es la que contiene los datos del negocio por lo que es la encargada de los accesos a la base de datos y de gestión de transacciones del sistema.

Gráficamente el funcionamiento de una aplicación Java EE sería el siguiente:

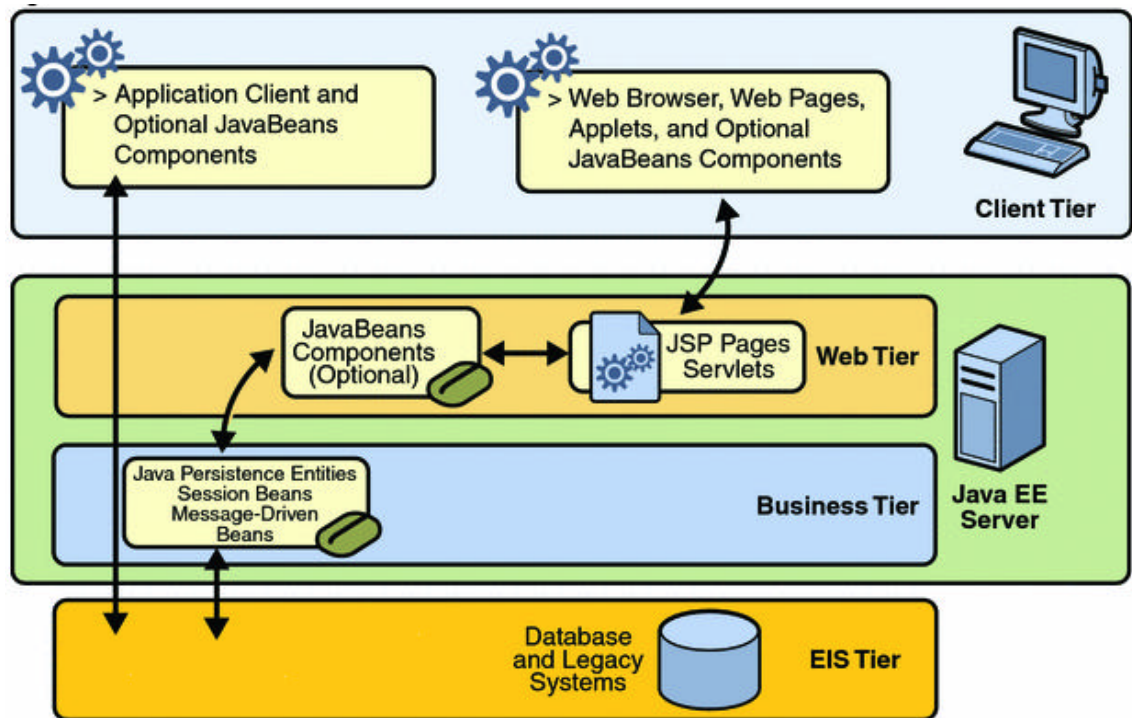


Figura 21 - Arquitectura Java EE

La división en capas facilita que el sistema sea escalable y reduce el acoplamiento entre los diferentes componentes, ya que cada capa tiene sus responsabilidades definidas dentro de la aplicación y nunca tendremos una funcionalidad repartida entre diferentes capas.

### 3.1.2 Patrón arquitectónico: MVC

Los **patrones de diseño (design patterns)** proporcionan soluciones ya probadas a problemas con características similares. Proporcionan catálogos de elementos reusables en el diseño de sistemas y estandarizan la forma en que se realiza el diseño, lo que facilita el aprendizaje.

La utilización de patrones de diseño no es una práctica obligatoria en el diseño de software, pero sí que es altamente recomendable ya que permiten reducir el tiempo de desarrollo de una aplicación al evitar el tener que plantear una solución desde cero.

El patrón **MVC (Modelo-Vista-Controlador)** es el más ampliamente establecido desde el punto de vista arquitectónico y se caracteriza porque divide el sistema en tres partes, de forma que separa los datos de la aplicación de la interfaz de usuario y de la lógica de control.

Con esta división conseguimos, por ejemplo, que la forma de presentar unos datos sea completamente independiente de los datos en sí, con lo que es sencillo el presentar los mismos datos de formas diferentes.

Las responsabilidades de cada una de las partes de este patrón son las siguientes:

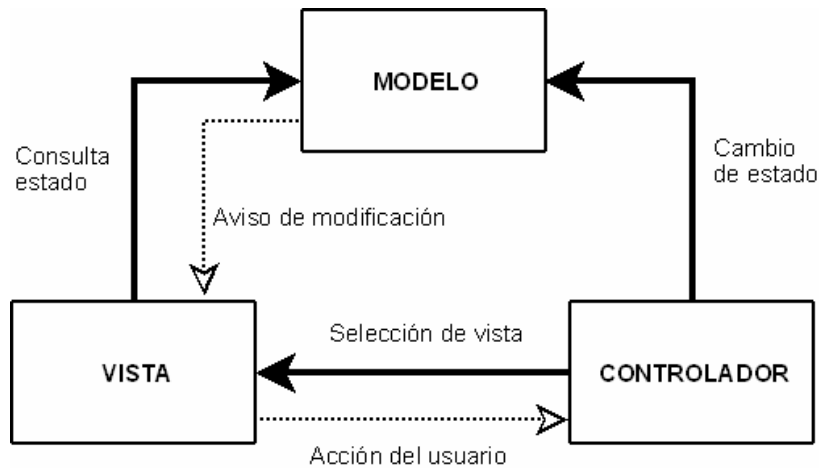
**Modelo:** Contiene los datos con los que opera el sistema. La lógica de datos permite asegurar su integridad y facilita el derivar nuevos datos.

**Vista:** Representa el modelo de datos y las operaciones realizadas en la capa de negocio en un formato adecuado para que el usuario pueda interactuar (mediante la interfaz de usuario).



**Controlador:** Responde a los eventos, normalmente acciones del usuario, modificando el modelo y generalmente también en la vista, por lo que es encargado de la interacción entre los datos y la vista.

El flujo seguido por este patrón entre los diferentes componentes es el siguiente:



**Figura 22 - Patrón MVC**

El usuario interactúa con la vista (interfaz de usuario) realizando algún tipo de acción.

El controlador recibe desde la vista la notificación de la acción solicitada.

El controlador accede al modelo actualizándolo conforme a la acción solicitada por el usuario.

El controlador delega en la vista la tarea de desplegar la interfaz de usuario con la respuesta.

La vista recupera los datos del modelo para poder completar la interfaz con las modificaciones realizadas sobre el modelo.

El modelo no debe tener conocimiento directo de la vista, sin embargo en algunos casos puede notificar a la vista de que se han producido cambios.

### 3.1.3 Frameworks

Un **framework** es una estructura definida a partir de la cual podemos desarrollar un proyecto de software y su intención es establecer una infraestructura que se encargue de realizar las tareas de más bajo nivel necesarias en cualquier proyecto y permitir así a los desarrolladores el poder dedicar más esfuerzo a las tareas de más alto nivel propias del negocio.

#### 3.1.3.1 Struts2

**Struts2** es un framework de aplicación web open source desarrollado por Apache y basado en el patrón MVC (Modelo-Vista-Controlador) que es utilizado ampliamente y considerado de gran solidez.

Se utiliza para construir aplicaciones web basadas en servlets JSP, pudiendo ejecutarse en cualquier contenedor de servlets, incluyendo los servidores de aplicaciones Java EE.

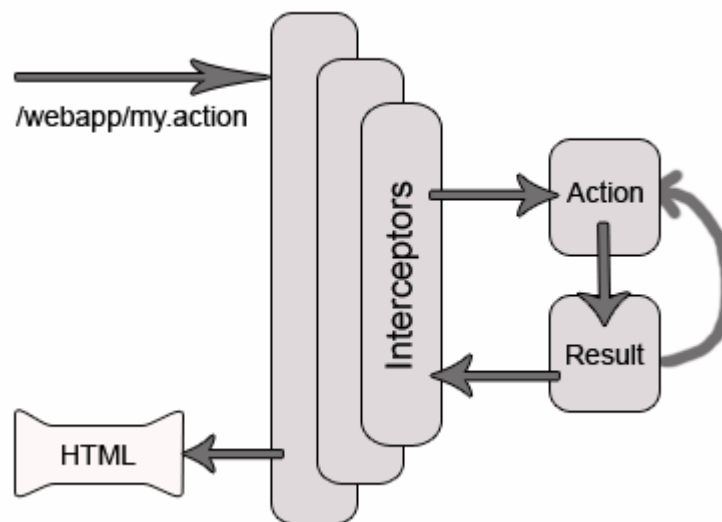
Utiliza internamente una serie de patrones ya definidos (Singleton, Delegate,...) y además proporciona un conjunto de etiquetas JSP personalizadas que facilitan la integración del framework con las páginas JSP. Algunas de estas etiquetas han sido utilizadas en este TFC.

El funcionamiento de Struts2 es el siguiente, cuando el usuario hace una solicitud al servidor el **FilterDispatcher** la captura y determina el **Action** que la debe tratar.

Antes de ejecutar el Action se le aplican a la solicitud los **Interceptors** que hayan sido configurados y que permiten realizar como preproceso una serie de tareas más o menos comunes, como pueden ser validaciones de datos.

A continuación se ejecuta la acción que realiza los accesos para recuperar o almacenar información en la base de datos y se genera la salida mediante un **Result**.

Esta salida vuelve a pasar a través de los Interceptors (en orden inverso al inicial) para realizar posibles operaciones de postproceso y finalmente se devuelve la respuesta al usuario.



*Figura 23 - Esquema Struts2*

Desde el punto de vista del patrón MVC los componentes de Struts2 se distribuirían de la siguiente manera de cara a conseguir la separación de los diferentes componentes de la aplicación:

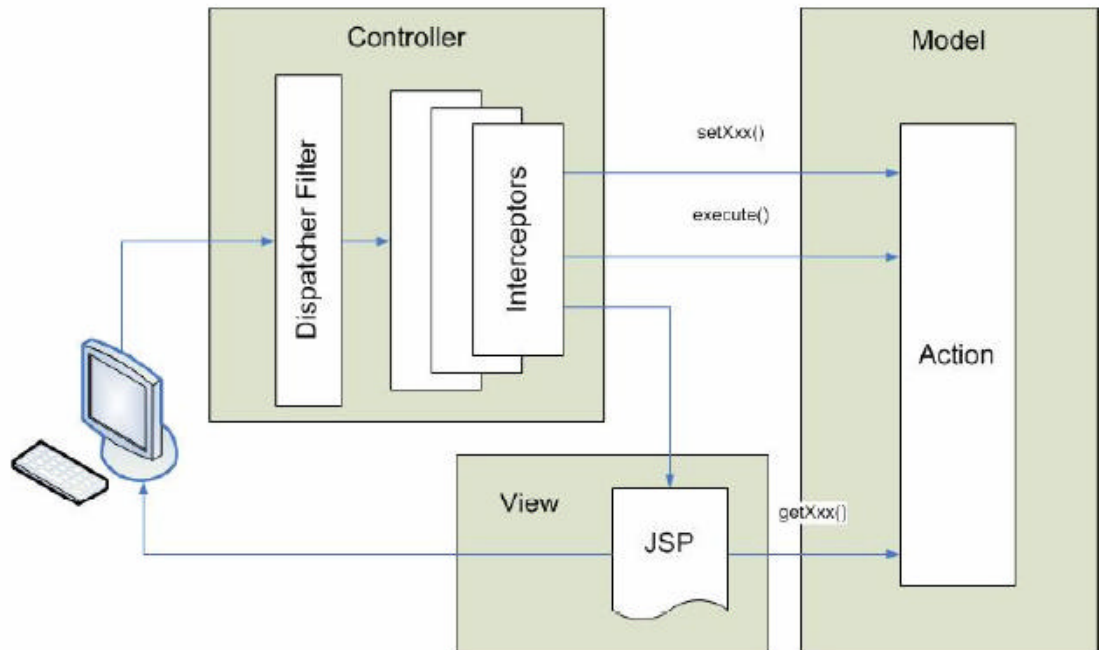


Figura 24 - Esquema Struts2-MVC

### 3.1.3.2 Spring

**Spring** es un framework open source que implementa el patrón Factory y que proporciona un marco de trabajo para el desarrollo de aplicaciones Java EE. Su principal objetivo es ser una alternativa sencilla y fácil de usar frente a Enterprise JavaBean.

Está basado en un conjunto de módulos que proporcionan todo lo necesario para desarrollar una aplicación empresarial aunque no es necesario basar la aplicación al completo en este framework, su modularidad hace que sea posible hacer uso del/los módulos que requiera la aplicación e ignorar el resto.

Su componente principal es un **contenedor de Beans** que nos permite, mediante el uso de la **Inversión de Control** (IoC), acoplar las diferentes partes de un aplicativo utilizando la inyección de dependencias.

Mientras en una arquitectura tradicional (como EJB) un componente debe hacer una llamada al contenedor para solicitar un objeto que necesite para realizar su tarea, con la inyección de dependencias el contenedor ya sabe (mediante su configuración) que el componente necesitará el objeto y se lo proporcionará (inyectará) en tiempo de ejecución.

Con Spring no se crea una instancia de un objeto para cada usuario del sistema, sino que se instancian una única vez al poner en marcha la aplicación, se guardan en la factoría y luego se comparten por todos los usuarios inyectándolos en los objetos que los necesiten mediante el uso de un método 'set' definido en el propio objeto.

La diferencia entre los dos comportamientos queda reflejada en el siguiente ejemplo:

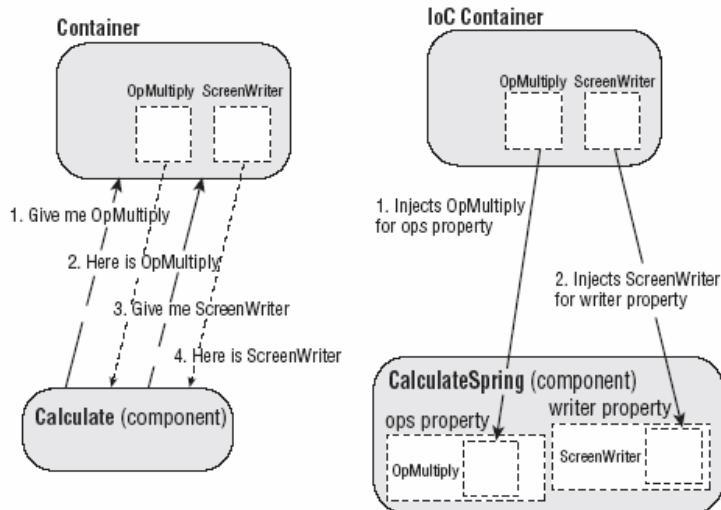


Figura 25 - Esquema Spring

### 3.1.3.3 Hibernate

**Hibernate** es una tecnología que simplifica el acceso a base de datos que se distribuye como una herramienta de software libre distribuida bajo los términos de la licencia GNU LGPL.

Permite establecer una correspondencia entre el modelo de la Base de Datos Relacional y una serie de clases que modelan los objetos de la aplicación (**mapeo objeto-relacional**), es decir, relaciona los dos modelos de datos que conviven en una aplicación, el usado en la memoria del ordenador (orientación a objetos) y el usado en las bases de datos (modelo relacional). Este mapeo hace que en la práctica se cree una base de datos orientada a objetos virtual sobre la base de datos relacional, en la que Hibernate actúa de intermediario con la aplicación que la utiliza.

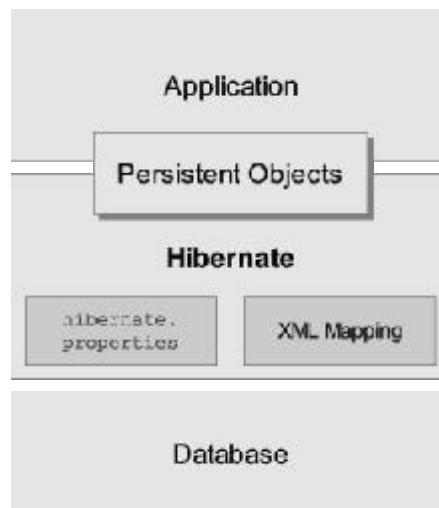


Figura 26 - Esquema Hibernate

Con esta tecnología disponemos de un sistema de acceso a bases de datos relacionales de manera transparente, únicamente se deben escribir sentencias java y de manera independiente, el código escrito con Hibernate funcionará en cualquier motor de datos al que

se dé soporte, ya que Hibernate permite a la aplicación el manipular datos de la base de datos operando sobre objetos.

## 3.2 Diseño de la persistencia

### 3.2.1 Diagrama E-R

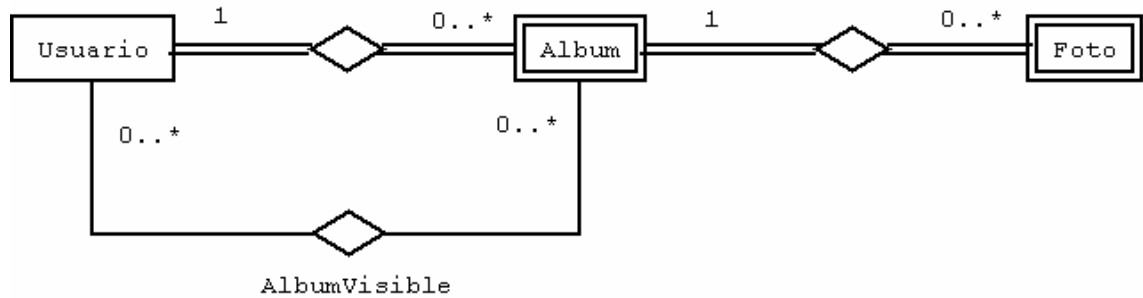


Figura 27 - Diagrama E-R

Las tablas que finalmente obtenemos en nuestro sistema son las siguientes:

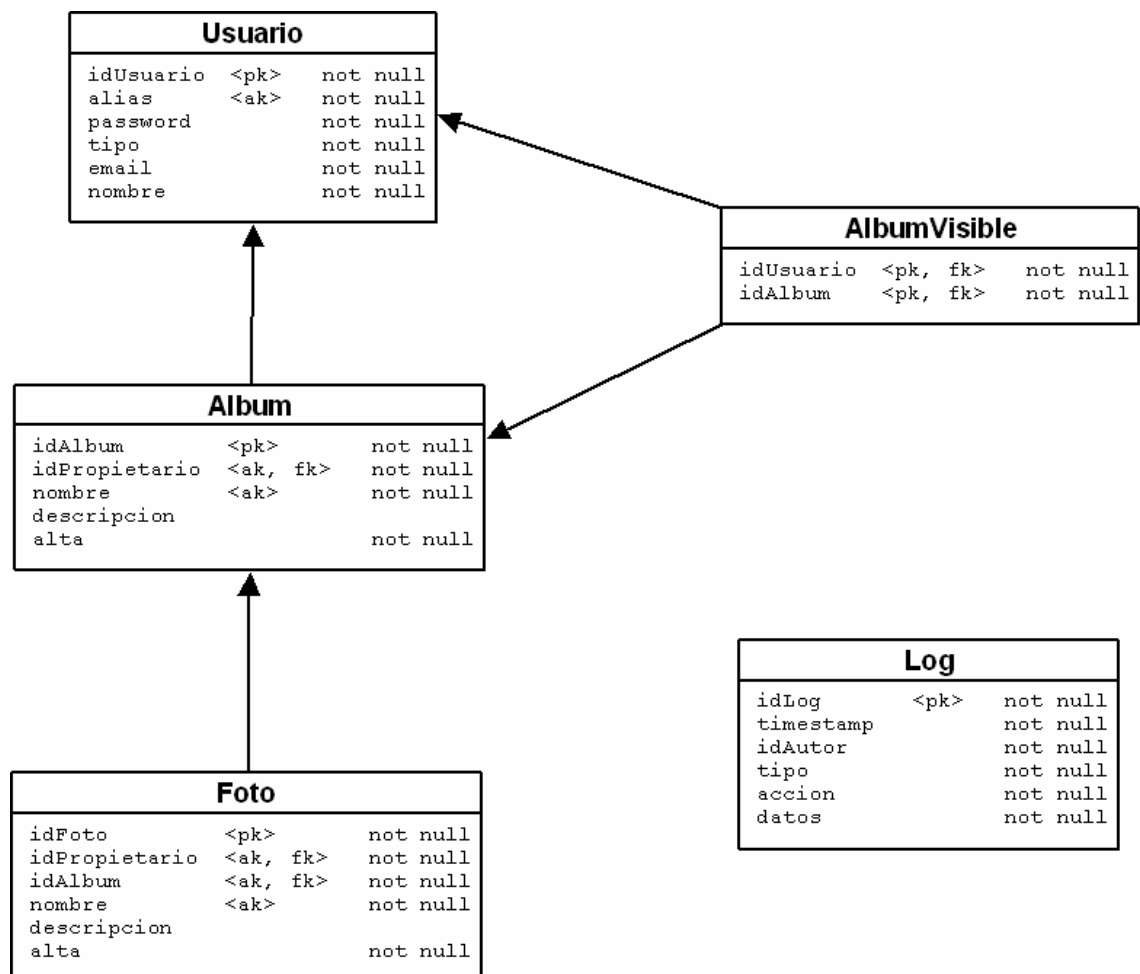


Figura 28 - Diseño Base de Datos

Se ha deshecho la relación de herencia de Cliente y Administrador con Usuario creando una única entidad con un atributo (tipo) que servirá para distinguirlos.

La relación AlbumVisible se ha convertido en una tabla que nos permitirá determinar los álbumes que podrá ver un usuario, sea o no propietario del mismo.

Los archivos con las fotografías no se almacenarán en la BD, sino en directorio del servidor cuyo nombre vendrá definido a partir de la clave primaria de cada fotografía.

En la tabla de Log guardaremos el identificador del usuario que ha realizado la modificación (idAutor), mientras que el objeto que ha sufrido la modificación vendrá identificado por su tipo (usuario, álbum,...) y su identificador correspondiente (su clave primaria).

### **3.3 Interfaz de usuario**

El diseño de las **clases frontera** determinadas a partir de los **Diagramas de Colaboración** ya con el diseño definitivo es el que se muestra a continuación.

La estructura de todas las pantallas de la aplicación es similar:

- Cabecera en la que aparece el nombre y el logo de la web. En caso de que haya un usuario conectado se muestra su alias y si se ha seleccionado algún álbum para trabajar con él también aparece su nombre.
- Menú lateral en la parte izquierda en el que van apareciendo las diferentes opciones disponibles en función de la pantalla en la que nos encontremos. En las pantallas en las que necesariamente tiene que haber un usuario conectado para ser mostradas, siempre aparece como primera opción 'Desconexión' que elimina la sesión del usuario y le envía a la página inicial de la aplicación.
- En la parte superior del cuerpo de las páginas tenemos una barra que nos indica en qué lugar de la web nos encontramos. Pulsando sobre alguna de las páginas previas a la actual nos envía a dicha página sin completar la acción (alta, modificación,...) que se estuviera realizando en ese momento.
- En el cuerpo propiamente dicho de la página nos aparece la información referente a dicha página y que evidentemente irá cambiando en función de la navegación que vaya haciendo el usuario en la aplicación.
- Pie en la que se muestra información de la web

Hay que tener en cuenta que el diseño de las pantallas y la captura de las imágenes se ha realizado usando Firefox. Utilizando otros navegadores el aspecto puede diferir algo respecto al mostrado aquí.

- **Página inicial**

Pantalla inicial de la aplicación con una breve descripción del servicio ofrecido.

Las opciones disponibles son la creación de un nuevo usuario o bien la conexión al sistema utilizando un usuario ya existente en el sistema.



**Figura 29 - Pantalla: Home**

- **Registro de nuevo cliente**

Pantalla de registro para la creación de un nuevo usuario cliente del sistema. Se identifican los campos obligatorios en el formulario y se solicita al usuario que lea y acepte las condiciones de uso de los servicios ofrecidos por la aplicación.

Desde esta pantalla el usuario sólo puede completar el alta del nuevo usuario o bien abandonarla saliendo a la pantalla inicial de la aplicación.

: Album Web :

Salir

**Formulario de registro.**

Una vez leídas las [Condiciones de Uso](#) de nuestra web completa los datos que te solicitamos para poder darte de alta en nuestro sistema y que de esta forma puedas utilizar los servicios que te ofrecemos.

Recuerda que los campos marcados con un asterisco son obligatorios.

\* Usuario:  
[input type="text"]

\* Password:  
[input type="password"]

\* Repita la Password:  
[input type="password"]

\* E-mail:  
[input type="text"]

\* Nombre:  
[input type="text"]

\* He leído y acepto las condiciones de uso.

Registro

pccofg-UOC TFC-JEE (2008/09-1)

**Figura 30 - Pantalla: Registro Cliente**



- **Confirmación de registro OK**

Se muestra un mensaje indicando que el alta del nuevo usuario se ha realizado correctamente. Al completarse el alta del nuevo usuario se realiza automáticamente su conexión al sistema.

El usuario puede desconectarse del sistema o bien ir a su página inicial para empezar a trabajar en la web.



**Figura 31 - Pantalla: Registro Cliente OK**

- **Conexión**

Pantalla de conexión al sistema utilizando un usuario ya registrado previamente. El usuario tiene que proporcionar su identificador y la clave de conexión correspondiente.

El usuario puede completar los datos solicitados para realizar la conexión o bien salir a la página inicial de la aplicación.

La pantalla mostrada después de realizada la conexión dependerá del tipo de usuario que la realiza (cliente o administrador).

A partir del momento en el que se realiza la conexión aparece el alias del usuario en la parte superior derecha de la página.

:. Album Web :.

Salir

**Conexión al sistema.**

Introduce tu usuario y tu clave de acceso para poder entrar en el sistema.

Usuario:

Password:

Enviar

pac@fg-U00 TFC-JEE (2008/09-1)

**Figura 32 - Pantalla: Conexión**

- **Lista de álbumes del usuario conectado**

Es la página que le aparece inicialmente a un cliente cuando realiza su conexión al sistema, en ella aparecen la lista de los álbumes a los que tiene acceso ya sea como propietario o como invitado.

Los álbumes visibles para el usuario pero que no son de su propiedad aparecen en la lista marcados con (\*\*).

Desde aquí el usuario tiene tres opciones, modificar los datos de su usuario, crear un nuevo álbum, o bien acceder a algunos de los álbumes que tiene disponibles pulsando sobre su nombre.

Si se opta por seleccionar uno de los álbumes disponibles a partir de ese momento aparecerá el nombre del álbum elegido en la parte superior derecha de la pantalla.

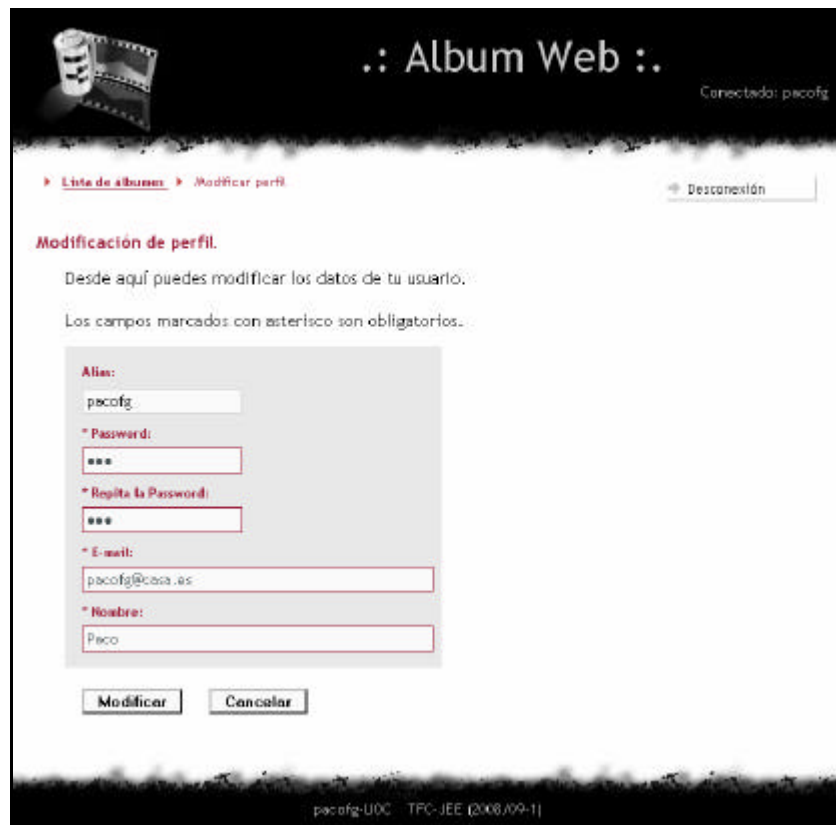


**Figura 33 - Pantalla: Lista de álbumes**

- **Modificación del propio usuario**

La pantalla permite que un usuario modifique sus propios datos. Salvo el propio alias del usuario, el resto de campos son modificables.

Una vez confirmada la modificación se vuelve a mostrar al usuario la pantalla en la que aparecen los álbumes de los que dispone.



The screenshot shows a web interface for 'Album Web'. At the top left is a logo with a film strip and a camera. The title ':. Album Web :.' is centered at the top, with 'Conectado: pacofig' on the right. Below the title is a navigation bar with links for 'Lista de álbumes' and 'Modificar perfil', and a 'Desconexión' button. The main heading is 'Modificación de perfil.' followed by instructions: 'Desde aquí puedes modificar los datos de tu usuario.' and 'Los campos marcados con asterisco son obligatorios...'. The form contains the following fields: 'Alias:' (pacofig), '\* Password:' (masked with \*\*\*), '\* Repite la Password:' (masked with \*\*\*), '\* E-mail:' (pacofig@cas.es), and '\* Nombre:' (Paco). At the bottom of the form are 'Modificar' and 'Cancelar' buttons. The footer text is 'pacofig-UOC TFC-JEE (2008/09-1)'.

**Figura 34 - Pantalla: Modificación usuario**

- **Alta de un nuevo álbum**

Le permite a un usuario realizar el alta de un nuevo álbum de su propiedad para lo cual sólo es obligatorio que le dé un nombre, la descripción es opcional.

Al confirmar el alta se le muestra al usuario la lista de álbumes de los que dispone en la que ya aparece el álbum creado.



The screenshot shows a web application interface for 'Album Web'. At the top left is a logo with a film strip and a camera. The title ':. Album Web :.' is centered at the top, with 'Conectado: pacofg' on the right. Below the title is a navigation bar with links for 'Lista de álbumes' and 'Alta álbum', and a 'Desconexión' button. The main content area is titled 'Alta de un nuevo álbum.' and contains the instruction: 'Debes proporcionarnos los datos necesarios para crear un nuevo álbum en el que podrás incluir las fotografías que desees.' Below this is a form with two fields: '\* Nombre:' with a text input box, and 'Descripción:' with a larger text area. At the bottom of the form are two buttons: 'Crear' and 'Cancelar'. The footer of the page contains the text 'pacofg-UOC TFC-JEE (2008/09-1)'.

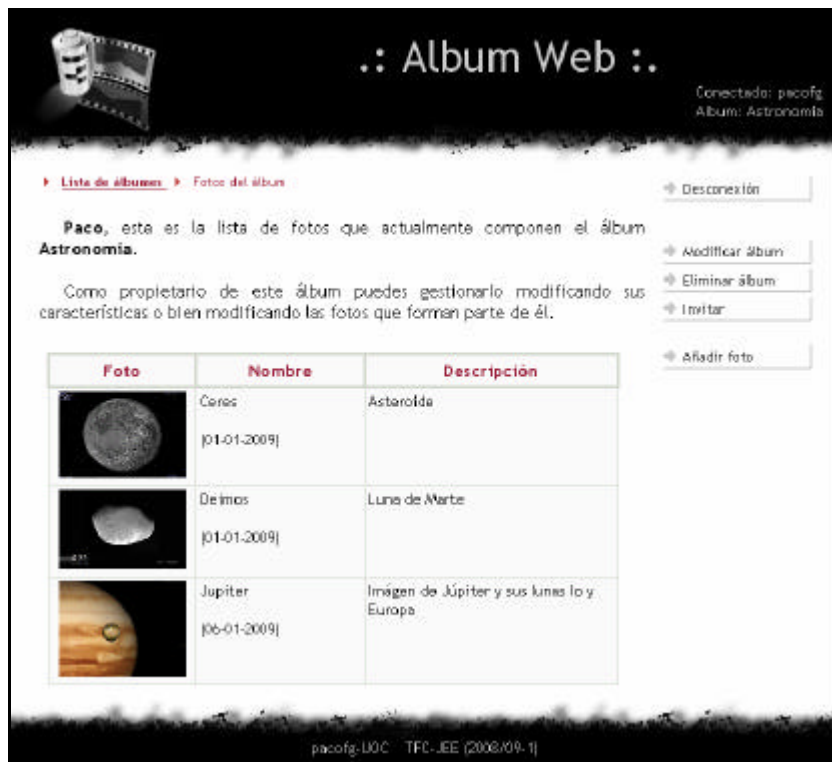
**Figura 35 - Pantalla: Alta de álbum**

- **Lista de fotos de un álbum propio**

Muestra la lista de fotografías relacionadas a un álbum propiedad del usuario conectado.

En esta pantalla el usuario puede gestionar el álbum modificando sus datos o bien dándolo de baja, puede invitar a otro usuario para que quede autorizado a ver el álbum y también puede añadir nuevas fotografías al álbum.

Pulsando sobre la fotografía se enlaza con la pantalla en la que se muestra la fotografía y donde además tenemos las opciones para gestionarla.



**Figura 36 - Pantalla: Lista de fotos (propietario)**

- **Lista de fotos de un álbum para un invitado**

La pantalla es idéntica a la anterior con la salvedad de que al estar en un álbum del que el usuario no es propietario las opciones disponibles son diferentes.

Desaparecen las opciones del menú desde las que se puede realizar la gestión del álbum y aparece una opción desde la que se puede dar de baja la invitación recibida por el usuario, con lo que el álbum al que ha sido invitado desaparecerá de su lista de álbumes disponibles.

Igual que en el caso anterior, pulsando sobre la imagen se accede a la pantalla en la que se muestra la fotografía.

Conectado: pacofg  
Album: Cosmos


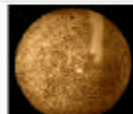
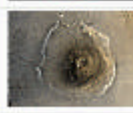
Lista de álbumes Fotos del álbum

Desconexión

Eliminar Invitación

Paco, esta es la lista de fotos que actualmente componen el álbum **Cosmos**.

No eres el propietario de este álbum, por lo que no podrás modificarlo de ninguna manera, aunque si que puedes visualizar las fotografías que lo componen.

Foto	Nombre	Descripción
	Marte [06-01-2009]	Planeta Marte
	Mercurio [07-01-2009]	Planeta Mercurio
	Monte Olimpo [07-01-2009]	Volcan del planeta Marte

pacofg-UDC TFC-JEE (2008/09-1)

**Figura 37 - Pantalla: Lista de fotos (invitado)**

- **Modificación de un álbum**

Muestra la información referente a un álbum dejando los campos desprotegidos para que puedan ser modificados por el usuario. Igual que en el caso del alta el nombre del álbum es obligatorio, pero la descripción asociada no lo es.

Una vez confirmada la modificación se devuelve al usuario a la lista de fotografías vinculadas al álbum con el que está trabajando y si se ha modificado el nombre éste también cambia en la cabecera de la página.



**Figura 38 - Pantalla: Modificación álbum**



- **Baja de un álbum**

Permite realizar la baja de un álbum propiedad del usuario. Esta acción implica el borrado en el servidor del directorio asociado con este álbum y que es donde se guardan las fotografías relacionadas, con lo que éstas se eliminarán de forma permanente del servidor.

Una vez confirmado el borrado se devuelve al usuario a la lista de álbumes que tiene disponibles y de la que ya habrá desaparecido el álbum borrado.



**Figura 39 - Pantalla: Baja de álbum**

- **Invitar a un álbum**

El propietario del álbum puede seleccionar un usuario del desplegable en el que aparecen todos los usuarios del sistema a excepción del propietario del álbum y de los usuarios con rol de administrador.

Una vez confirmada la invitación se devuelve al usuario a la lista de fotografías vinculadas al álbum con el que está trabajando.



**Figura 40 - Pantalla: Invitar a álbum**

- **Baja de invitación**

La pantalla permite a un usuario que ha sido invitado a ver un álbum el eliminar éste de su lista de álbumes disponibles. Esta acción elimina la relación del usuario con el álbum, pero no modifica en ningún aspecto las fotografías almacenadas en el servidor.

Una vez confirmada la baja se devuelve al usuario a la lista de álbumes que tiene disponibles y en la que ya no aparecerá el álbum eliminado.



**Figura 41 - Pantalla: Eliminar invitación**

- **Alta de foto en álbum**

Desde esta pantalla el usuario propietario de un álbum puede añadir nuevas fotografías a dicho álbum, para ello basta con informar el nombre de la fotografía y seleccionar el archivo que contiene la fotografía a subir al servidor. La descripción es un campo opcional.

Una vez confirmada el alta se vuelve a mostrar al usuario la lista de fotografías que tiene asociadas el álbum y en la que ya aparecerá la nueva fotografía incorporada.



The screenshot shows a web interface titled "Album Web". At the top left is a film strip icon. The top right shows the user's connection details: "Conectador: pacofg" and "Album: Astronomia". Below the title is a navigation menu with "Lista de álbumes", "Fotos del álbum", and "Añadir foto". A "Desconectar" button is on the right. The main heading is "Incluir foto en álbum." followed by instructions: "Desde aquí puedes incorporar una nueva foto al album activo. Para hacerlo solo tienes que proporcionarnos los datos que te solicitamos y seleccionar el archivo con la foto que desees incluir en tu álbum." The form contains three fields: "Nombre:" (required), "Descripción:" (optional), and "Archivo foto:" (required) with an "Examinar..." button. At the bottom are "Crear" and "Cancelar" buttons. The footer text is "pacofg-UOC TFC-JEE (2008/09-1)".

**Figura 42 - Pantalla: Alta de foto**

- **Consulta de fotografía de álbum propio**

Se enlaza con esta pantalla al pulsar sobre una de las imágenes que aparecen en la lista de fotografías de un álbum y desde ella se pueden modificar sus datos, o bien darla de baja del álbum.

El usuario también dispone en esta pantalla de unos controles que le permiten navegar por las fotografías del álbum sin necesidad de volver a salir a la lista de fotografías que lo forman.

Estos controles permiten visualizar la fotografía anterior/posterior o bien la primera/última del álbum. En caso de que la fotografía sea la primera o la última aparecen desactivados los controles correspondientes.



**Figura 43 - Pantalla: Consulta de foto de álbum propio**

- **Consulta de fotografía de álbum para invitado**

En el caso de que la fotografía consultada pertenezca a un álbum que no sea propiedad del usuario se muestra la misma pantalla que en el caso anterior pero sin las opciones de menú que permiten enlazar con las operativas de modificación y borrado de la fotografía.



**Figura 44 - Pantalla: Consulta de foto de álbum invitado**

- **Modificación de foto**

Esta pantalla permite modificar el nombre de la fotografía y su descripción, siendo el primer campo obligatorio y el segundo no. No se permite la modificación del archivo relacionado con lo que ya ni siquiera se muestra la opción correspondiente.

Una vez realizada la modificación se muestra al usuario la pantalla con la fotografía con la que está trabajando.



**Figura 45 - Pantalla: Modificación de foto**

- **Baja de fotografía**

En la pantalla se solicita confirmación al usuario para realizar la baja de la fotografía que además será eliminada físicamente del servidor.

Después de confirmada la acción se devuelve al usuario a la lista de fotografías del álbum con el que está trabajando.



**Figura 46 - Pantalla: Baja de foto**



- **Home del administrador**

Pantalla inicial que se muestra a los usuarios con rol de administrador una vez se han identificado correctamente en el sistema.

Las opciones de las que dispone el administrador son la modificación de su usuario (opción exactamente igual a la ya descrita para los usuarios clientes), el alta de un nuevo usuario con rol de administrador y el acceso a la lista de usuarios existentes en el sistema para su gestión.



**Figura 47 - Pantalla: Home de administrador**

- **Registro administrador**

La pantalla mostrada es idéntica a la existente para el registro de usuarios clientes salvo que no es necesario aceptar ningún tipo de condiciones de uso.

Una vez realizado el registro se devuelve al usuario a su pantalla inicial en la que se le confirma que el alta se ha realizado correctamente.

Conectado: admin

Home Alta administrador Desconexión

**Formulario de registro.**

Completa los datos que te solicitamos para realizar en nuestro sistema el alta del nuevo usuario administrador.

Recuerda que los campos marcados con un asterisco son obligatorios.

\* Usuario:

\* Password:

\* Repita la Password:

\* Email:

\* Nombre:

Registro

pacofg-U0C TFC-JEE (2006/09-1)

**Figura 48 - Pantalla: Alta de administrador**

- **Lista de usuarios**

Muestra al administrador la lista con todos los usuarios existentes en el sistema (salvo el propio usuario conectado) con las opciones disponibles según el tipo que sean.

Para todos los usuarios están disponibles las opciones de modificación y baja y sólo para los usuarios clientes está disponible también la opción de gestión de sus álbumes.

**Gestión de usuarios.**

Desde aquí puedes gestionar los usuarios existentes en el sistema. Selecciona la opción a realizar sobre el usuario.

Usuario	e-Mail	Tipo	Opciones
ad	ad@casa.es	Admin.	
ad1	ad1@casa.es	Admin.	
ad2	ad2@casa.es	Admin.	
alnstaln	alnstaln@casa.es	Cliente	
otra	otra@casa.es	Cliente	
pacofg	pacofg@casa.es	Cliente	
prueba	prueba@casa.es	Cliente	
sagan	sagan@casa.es	Cliente	

pacofg-U0C TFC-EE (2006/09-1)

**Figura 49 - Pantalla: Lista de usuarios**

- **Modificación de usuario por administrador**

Esta opción le permite a un administrador el restaurar la password de cualquier usuario por lo que se le muestra toda la información del usuario aunque sólo está desprotegido el campo de password.

Una vez confirmada la modificación se devuelve al usuario a la lista de usuarios del sistema.

:. Album Web :.

Conectado: admin

Home > Lista usuarios > Modificar usuario

Desconexión

**Modificación de usuario.**

Modifique los datos del usuario.

Los campos marcados con asterisco son obligatorios.

Usuario:  
pacofig

\* Password:

\* Repita la Password:

E-mail:  
pacofig@cas.es

Nombre:  
Paco

Modificar Cancelar

pacofig-UOC TFC-JEE (2008/09-1)

**Figura 50 - Pantalla: Modificación de usuario (por administrador)**

- **Baja de usuario**

Se le solicita al administrador confirmación de la baja de usuario solicitada. Si el usuario es de tipo cliente esta baja implica el borrado de todos los directorios existentes en el servidor y que estén relacionados con él y por lo tanto la eliminación física de todas las fotografías que este usuario tenga en el servidor.

Una vez confirmada la baja se devuelve al usuario a la lista de usuarios del sistema de la que ya se habrá eliminado el usuario borrado.



**Figura 51 - Pantalla: Baja de usuario**

- **Lista de álbumes de un usuario para el administrador**

Muestra la lista de álbumes de los que el usuario del sistema es propietario permitiendo la opción de darlo de baja. Esta acción hará que se muestre una pantalla de confirmación exactamente igual a la que aparece cuando la baja la hace el usuario propietario del álbum.

Una vez confirmada la modificación se devuelve al usuario a la lista de usuarios del sistema.



**Figura 52 - Pantalla: Lista de álbumes (para administrador)**

## 4 Implementación

Para facilitar la implementación se ha utilizado **Maven** que es un framework de gestión de proyectos de software que proporciona un modelo estándar de gestión y descripción de proyectos.

Cubre las tareas que van desde la compilación hasta la distribución, despliegue y documentación de los proyectos, aunque en este TFC no se han utilizado todas sus posibilidades.

Utiliza una serie de repositorios de software (que pueden ser locales o remotos), lo que permite de una forma sencilla la reutilización de la lógica común para todos los proyectos que sigan los estándares de Maven.

Para reutilizar al máximo el código JSP se ha utilizado **SiteMesh** que es un framework de construcción de páginas web. Permite construir y decorar páginas web a partir de un patrón diseñado inicialmente lo que permite conseguir que una serie de páginas tengan una estructura y un aspecto semejante, dando así un aspecto más uniforme a las páginas de nuestro sitio web.

## 4.1 Estructura de la aplicación

En el directorio main tenemos las tres carpetas en las que se dividen los componentes de la aplicación

En la **carpeta java** tenemos todo el código java estructurado en cinco paquetes, cada uno de los cuales contiene las clases de un determinado tipo:

- **acciones:** Contiene los componentes **Action** de Struts que son los que proporcionan la lógica necesaria para dar respuesta a la solicitud hecha por el usuario.
- **interceptores:** Contiene los **Interceptors** incorporados a Struts que permiten realizar operaciones de pre/postproceso previas/posteriores a la ejecución de un **Action**. En este caso hay un único interceptor que se encarga de validar si hay un usuario autenticado o no en el sistema.
- **modelo:** Contiene las clases que mapean los objetos persistentes del sistema, en este caso las tablas de MySQL
- **servicios:** Contiene las clases que se encargarán de todos los accesos a las tablas de la aplicación.
- **util:** Contiene las clases no directamente relacionadas con el framework utilizado y que pueden ser utilizadas desde diferentes puntos del sistema. Contiene las clases que permiten realizar el envío de e-mails y la gestión de las fotografías almacenadas en el servidor.

En la **carpeta resources** tenemos los archivos xml que contienen la configuración del sistema, los más importantes de los cuales son:

- **struts.xml:** Contiene la definición de las acciones de Struts y la respuesta que hay que dar para cada una de dichas acciones.
- **applicationContext.xml:** Contiene la configuración de Spring con las clases que debe inyectar y el enlace con la base de datos del sistema, en este caso MySQL.

En la **carpeta webapp** está toda la parte visual de la aplicación, esto es, desde los componentes jsp hasta las imágenes e iconos utilizados.

- **jsp:** Contiene el cuerpo de las páginas jsp utilizadas en la aplicación.
- **menus:** Corresponden a los menús laterales y a la barra superior que aparecen en las páginas, cada uno de los cuales puede ser utilizado en diferentes páginas.
- **imagenes:** Contiene las imágenes e iconos utilizados en la decoración de las páginas. No contiene las imágenes gestionadas por la propia aplicación.
- **styles:** Archivo css con la definición de las características de los diferentes componentes visuales de las pantallas.
- **WEB-INF:** Contiene por un lado el archivo **web.xml** en el que está la definición del núcleo de Struts (el FilterDispatcher) y por otros los archivos propios de Sitemesh que es la herramienta encargada de montar las páginas de forma que el aspecto

final sea el más uniforme posible. En este directorio, dentro de la carpeta **decorators** tenemos el archivo `main.jsp` que es el utilizado como maqueta a la hora de montar las diferentes páginas.

## 4.2 Almacenamiento en el servidor

Las fotografías pertenecientes a los usuarios se almacenan en un directorio del servidor cuyo nombre viene parametrizado en el archivo **applicationContext.xml**. El nombre definido por defecto es '**depositoFotos**' .

Bajo este directorio tenemos un directorio para cada uno de los usuarios de la aplicación, dentro de éstos tenemos un directorio para cada uno de los álbumes del cliente y en éstos ya encontramos las fotografías que lo forman.

La nomenclatura de los directorios y de las fotografías almacenadas está formada por la concatenación de una letra (P para propietario, A para álbum y F para fotografía) y la correspondiente clave primaria del objeto. De esta forma la fotografía con clave 3, del álbum con clave 2, perteneciente al usuario 1 la encontraríamos en una dirección como la siguiente:

```
D:\Tomcat\apache-tomcat-6.0.18\webapps\pacofg_TFC\depositoFotos\P1\A2\F3.JPEG
```

## 4.3 Envío de e-mails

Para el envío de e-mails de aviso a los clientes se ha utilizado el API JavaMail de Sun.

Se ha desestimando la instalación de un servidor SMTP propio, por lo que para poder realizar el envío de correos, la aplicación utilizará una cuenta conseguida de forma gratuita en el servicio proporcionado por Yahoo.

Los parámetros de configuración de dicha cuenta de correo se encuentran en el archivo **applicationContext.xml** y son fácilmente configurables.

El sistema puede enviar estos tres tipos diferentes de e-mails:

- Invitación a álbum.
  - Asunto: Album Web: Invitación
  - Cuerpo: Nos complace comunicarle que el usuario de Album Web {alias usuario propietario} le ha invitado a compartir su álbum {nombre álbum}.
- Baja de usuario.
  - Asunto: Album Web: Baja de usuario
  - Cuerpo: Sentimos comunicarle que su usuario {alias usuario} ha sido dado de baja por el administrador de Album Web.
- Baja de álbum.
  - Asunto: Album Web: Baja de álbum
  - Cuerpo: Sentimos comunicarle que su álbum {nombre álbum} ha sido dado de baja por el administrador de Album Web.



## 5 Requerimientos

Para la realización del proyecto se han utilizado herramientas de software libre y la lista de las empleadas (con su correspondiente versión) es la siguiente:

- **Java 1.5.0.16:** Lenguaje de programación orientado a objetos utilizado en el desarrollo del proyecto.
- **Maven 2.0.9:** Herramienta para la gestión y construcción de proyectos. Permite describir las dependencias del proyecto y se encarga de importarlas desde la red.
- **Struts 2.0.11.2:** Herramienta para el desarrollo de aplicaciones web y que utiliza el patrón MVC para el desarrollo de dichas aplicaciones
- **Spring 2.0.5:** Herramienta que permite la interacción con el sistema de gestión de base de datos y el control transaccional de las operaciones.
- **Hibernate 3.2.1:** Herramienta de mapeo objeto-relacional para java que permite el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación.
- **MySQL 5.0.51b:** Sistema de gestión de base de datos relacional muy rápida en aplicaciones de baja concurrencia como es el caso del presente proyecto.
- **mysql-connector-java-5.1.5:** Driver para la conexión de java con MySQL.
- **Tomcat 6.0.18:** Servidor web con soporte de servlets y JSPs
- **SiteMesh 2.2.1:** Herramienta que permite la estructuración y decoración de páginas web de forma que facilita el que las diferentes páginas que forman una web tengan un aspecto uniforme.
- **JavaMail 1.3.3:** Extensión de Java que permite el envío y recepción de e-mails.

## 6 Instalación

Para poder utilizar la aplicación es necesario disponer de un servidor en el que desplegar la aplicación como puede ser Tomcat y tener instalado el sistema de gestión de base de datos MySQL.

Una vez cumplidos los requisitos anteriores hay que ejecutar en MySQL el script que se adjunta con este documento para crear el usuario y la base de datos empleados por la aplicación.

La configuración actual de la aplicación utiliza la IP **127.0.0.1** y el puerto **3306**, así como el usuario **pacofg** (creado con el script anteriormente mencionado) con password **tfc** para realizar la conexión. Cualquier cambio en alguno de estos parámetros se tendría que hacer en el archivo applicationContext.xml que es donde está definida dicha configuración.

## 7 Valoración económica

Se han cumplido todos los plazos previstos en la planificación inicial, pero a un coste muy superior al esperado.

La causa de este incremento en el coste ha sido la necesidad de simultanear el trabajo propiamente dicho del TFC con la instalación y el estudio del manejo de todas las herramientas utilizadas en su desarrollo, esto hace que cualquier valoración del coste del proyecto quede absolutamente desvirtuada y carezca de cualquier valor, ni siquiera estimativo.

## 8 Conclusiones

La conclusión tras la realización de este TFC es que en general la tecnología relacionada es difícil de asimilar para un recién llegado al mundo de Java EE, pero que una vez comprendida mínimamente es relativamente sencillo el poder avanzar.

La gran cantidad de alternativas disponibles a la hora de decidir las herramientas a utilizar es una ventaja para un experto ya que le proporciona una flexibilidad enorme que no podría tener de otra forma, pero para un principiante puede ser un gran inconveniente.

Esta variedad en las herramientas disponibles, cada una con sus múltiples versiones (parece que no siempre compatibles entre sí) y unido en algunos casos a una documentación un tanto crítica, hacen que el usuario novato se encuentre perdido en medio de la inmensidad del mundo Java EE.

Se ha conseguido cumplir con los requerimientos planteados inicialmente en el proyecto, aunque sería fácil encontrar aspectos en los que la aplicación podría ser mejorada:

- Mejorar el control de errores, por ejemplo en los accesos a BBDD.
- Mejorar la seguridad del sistema (incluso las passwords se guardan en claro en la BBDD).
- Implementar búsquedas de usuarios para facilitar la realización de invitaciones por parte de los clientes o la gestión de usuarios por parte de los administradores.
- Crear un sistema de consultas y de búsquedas en el log de la aplicación que permitiera el control del sistema por parte de los usuarios.
- De cara a mejorar la gestión y el rendimiento de sistema se podrían generar estadísticas que permitieran identificar a los usuarios con mayor consumo de espacio en el servidor, usuarios que no se conectan desde hace tiempo, ...
- Toda la aplicación está desarrollada en castellano, se podría incluir el idioma en el perfil de los usuarios de forma que éstos pudieran elegir el idioma en que se les mostrara la web.

## 9 Bibliografía

### ❖ Java

Página de Sun sobre Java: <http://java.sun.com/javase/>

Java en Castellano: <http://www.programacion.net/java/>

JavaWorld: <http://www.javaworld.com/>

The Server Side: <http://www.theserverside.com/>

## ❖ Java EE

Tutoriales de Adictos al Trabajo: <http://www.adictosaltrabajo.com/tutoriales.php>

Página de Sun sobre Java EE: <http://java.sun.com/javaee/>

The Java EE 5 Tutorial: <http://java.sun.com/javaee/5/docs/tutorial/doc/>

Core J2EE Patterns: <http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>

## ❖ Struts2

Sitio del proyecto Struts2: <http://struts.apache.org/2.x/>

Tutoriales de Struts2: <http://www.roseindia.net/struts/struts2/>

Struts2 in Action. Donald Brown, Chad Michels Davis, Scott Stanlick. Manning - 2008.

Starting with Struts2. Ian Roughley. C4Media Inc - 2006.

Practical Apache Struts2 Web 2.0 Projects. Ian Roughley. Apress - 2007.

## ❖ Spring

Sitio del proyecto Spring: <http://www.springsource.org/>

SpringHispano: <http://www.springhispano.org/>

Beginning Spring Framework 2. Thomas Van de Velde, Bruce Snyder, Christian Dupuis, Sing Li, Anne Horton, Naveen Balani. Wiley Publishing Inc. - 2008.

## ❖ Hibernate

Sitio del proyecto Hibernate: <http://www.hibernate.org/>

Hibernate Annotations: [http://www.hibernate.org/hib\\_docs/annotations/reference/en/html/](http://www.hibernate.org/hib_docs/annotations/reference/en/html/)

Java Persistence with Hibernate. Christian Bauer, Gavin King. Manning - 2008.

## ❖ SiteMesh

Sitio del proyecto SiteMesh: <http://www.opensymphony.com/sitemesh/>

## ❖ Maven

Sitio del proyecto Maven: <http://maven.apache.org/index.html>

Maven Getting Started Guide: <http://maven.apache.org/guides/getting-started/index.html>

## ❖ Tomcat

Sitio del proyecto Tomcat: <http://tomcat.apache.org/>

## ❖ MySQL

Sitio del proyecto MySQL: <http://www.mysql.com/>

Manual Referencia Oficial Ver 5.0: <http://dev.mysql.com/doc/refman/5.0/es/index.html>

## ❖ JavaMail

Sitio de Sun para JavaMail: <http://java.sun.com/products/javamail/>

## ❖ HTML - CSS

Tutorial de HTML: <http://www.w3schools.com/html/default.asp>

Tutorial de CSS: <http://www.w3schools.com/css/default.asp>

## ❖ Información en general

Google: <http://www.google.es/>

Yahoo: <http://es.yahoo.com/>

Wikipedia en castellano: <http://es.wikipedia.org/wiki/Wikipedia:Portada>

Wikipedia en inglés: [http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page)