



Escola Universitària  
Politécnica de Mataró

**Ingeniería Técnica en Informática de Gestión**

**RED SOCIAL DE INTERCAMBIO DE ALOJAMIENTOS**

Autor: **ANTONIO RUIZ BUENO**  
Ponente: **JOAN JOU I MAJÓ**

**PRIMAVERA 2009**

## **Resumen**

A nivel europeo se está extendiendo una práctica relativamente reciente que consiste en el intercambio de alojamientos. Las personas ofrecen sus alojamientos a otras personas para que durante sus vacaciones tengan un sitio donde alojarse sin costes de estancia. También es frecuente que la persona que va a alojarse ofrezca también su alojamiento como moneda de cambio.

Este proyecto quiere facilitar este proceso mediante una red social online en la que se pueda publicar y consultar alojamientos de todo el mundo, permitiendo un sistema de reservas e intercambios, usando, además, el servicio de Google Maps para mostrar los alojamientos geográficamente.

## **Resum**

A nivell europeu s'està estenent una pràctica relativament recent que consisteix en l'intercanvi d'allotjaments. Les persones ofereixen els seus allotjaments a altres persones per a que durant les seves vacances tinguin un lloc on allotjar-se sense costos d'estada. També és freqüent que la persona que va a allotjar-se ofereixi també el seu allotjament com a moneda de canvi.

Aquest projecte vol facilitar aquest procés mitjançant una xarxa social online en la qual es pugui publicar i consultar allotjaments de tot el món, permetent un sistema de reserves i intercanvis, fent servir el servei de Google Maps per a mostrar els allotjaments geogràficament.

## **Abstract**

# ÍNDICE

	<b>Pág.</b>
1. Introducción.....	1
2. Objetivos.....	2
3. Estudio de Mercado.....	3
4. Metodología y Tecnología.....	7
4.1 Metodología.....	7
4.2 Herramientas de Desarrollo.....	8
5. Planificación.....	9
6. Análisis.....	11
6.1 Requisitos. Casos de Uso.....	11
6.2 Patrones.....	21
6.3 Modelo del Negocio.....	24
6.4 Reglas de Negocio.....	25
7. Diseño.....	27
7.1 GUI.....	30
7.2 Dominio.....	40
7.3 Persistencia.....	48
8. Desarrollo e Implementación.....	49
8.1 Presentación.....	50
8.2 Controlador y Dominio.....	67
8.3 Persistencia.....	69
9. Pruebas.....	77
10. Conclusiones.....	83
Anexo.....	85
11. Bibliografía.....	87

## 1. Introducción

### 1.1 Descripción del proyecto

El proyecto consiste en desarrollar una aplicación web que albergue una red social donde las personas puedan ofrecer y buscar alojamientos a nivel nacional e internacional, contemplando también la posibilidad de realizar intercambios directos.

La aplicación debe facilitar la publicación, gestión, reserva, intercambio y búsqueda de alojamientos, aprovechando todas las ventajas que ofrece el servicio de Google Maps, que permite saber la situación geográfica de los alojamientos en un mapa interactivo. Así se crea una comunidad de intercambio de alojamientos que facilite la estancia cuando se viaja, sin necesidad de depender de los hoteles y los costes que suponen.

### 1.2 Motivación

Uno de los principales problemas a la hora de viajar son los costes que supone la estancia en un hotel o cualquier otro tipo de servicio de alojamiento. En algunas fechas recurrir a un hotel puede resultar realmente caro e incluso se hace complicado encontrar sitio.

Entorno a esta base, es frecuente que personas ofrezcan sus propias casas para alojar a otras personas que quieran viajar a otros países e incluso, últimamente, se está haciendo más común el intercambio simultáneo de casas, anulando prácticamente cualquier coste de estancia. La idea es que una persona le deje su casa a otra y viceversa durante un periodo de tiempo, como moneda de cambio.

Esto que en principio puede parecer utópico, ya está realizándose a nivel mundial, sólo es cuestión de hacerlo con personas de confianza (también se pueden hacer contratos), tener ganas de conocer gente y saber dónde encontrar la información. Y ahí es donde entra este proyecto.

## 2. Objetivos

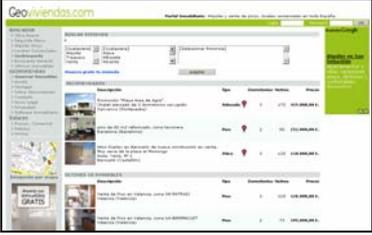
El objetivo es realizar una aplicación web que permita crear un banco de alojamientos que se comparten entre usuarios a nivel mundial.

La aplicación deberá ofrecer las siguientes características:

- Gestión de alojamientos: publicar un nuevo alojamiento, editarlo, añadir o quitar fotografías, borrarlo y consultar las reservas solicitadas por parte de otros usuarios, así como confirmar o rechazar las peticiones de reserva.
- Realizar búsquedas personalizadas de todos los alojamientos que hay publicados.
- Permitir comentar y valorar los alojamientos por parte de los inquilinos para ayudar a otros usuarios a conocer mejor el alojamiento.
- Permitir realizar reservas en las fechas deseadas para un alojamiento.
- Consultar las reservas que se ha solicitado y ver su estado.
- Consultar la información de un usuario, así como sus alojamientos y sus estancias.
- Permitir el intercambio de alojamientos entre dos usuarios.
- Uso de Google Maps tanto en la búsqueda como en el alta de alojamientos para saber en todo momento la ubicación geográfica de los alojamientos, ofreciendo un interfaz visual más atractiva y útil.
- Permitir valorar a un inquilino por parte del usuario que le ha ofrecido su alojamiento. Valoración de usuarios.

### 3. Estudio de Mercado

A continuación algunas aplicaciones web de referencia en el mercado para analizar sus ventajas e inconvenientes.

Competencia	Ventajas	Inconvenientes
<p>Geoviviendas <a href="http://www.geoviviendas.com">www.geoviviendas.com</a></p> 	<ul style="list-style-type: none"> <li>• Uso de Google Maps</li> <li>• Buscador siempre presente</li> <li>• Buscador personalizable</li> <li>• Dos opciones de búsqueda: con o sin Google Maps</li> <li>• Información del alojamiento en el propio mapa</li> <li>• Permite fotos</li> <li>• Sistema de ordenación</li> </ul>	<ul style="list-style-type: none"> <li>• No permite vídeos</li> <li>• No permite guardar alojamientos en una lista para los usuarios que busquen ofertas (cesta de la compra)</li> <li>• Registro sólo para propietarios</li> <li>• Nivel nacional</li> </ul>
<p>Servihabitat <a href="http://www.servihabitat.com">www.servihabitat.com</a></p> 	<ul style="list-style-type: none"> <li>• Aviso de ofertas y demandas</li> <li>• Lista de alojamientos favoritos</li> <li>• Servicio de tasación</li> <li>• Servicio de gestión de venta y publicidad</li> <li>• Registro para propietarios e inquilinos</li> <li>• Buscador personalizable</li> </ul>	<ul style="list-style-type: none"> <li>• Sólo permite 1 foto por alojamiento</li> <li>• Carece de Google Maps</li> <li>• Carece de un sistema de ordenación</li> <li>• Nivel nacional</li> </ul>

<p>HomeForHome www.homeforhome.com</p> 	<ul style="list-style-type: none"> <li>• Lista de favoritos sin necesidad de registro</li> <li>• Nivel internacional</li> <li>• Buscador completo</li> <li>• Sistema de ordenación</li> <li>• Registro para propietarios</li> </ul>	<ul style="list-style-type: none"> <li>• Carece de Google Maps</li> </ul>
<p>Intervacacionesclub www.intervacacionesclub.com</p> 	<ul style="list-style-type: none"> <li>• Situación geográfica Google Maps en la descripción del alojamiento</li> </ul>	<ul style="list-style-type: none"> <li>• De pago</li> <li>• Carece de Google Maps a nivel de búsqueda</li> </ul>

Fig 1. Tabla comparativa de algunos sitios web de referencia en alquiler e intercambio de alojamientos.

Las características deseadas para nuestro producto son las siguientes:

- Uso de Google Maps tanto en la búsqueda como en el alta de alojamientos.
- Buscador personalizable y siempre a la vista
- Inserción ilimitada de imágenes relacionadas con el alojamiento (¿vídeos?)
- Sistema de valoración de alojamientos (sólo para inquilinos)
- Registro de usuario requerido para intercambiar alojamientos.
- Sistema de reserva de alojamientos.
- Servicio gratuito
- Ubicación geográfica en Google Maps.
- Sistema de intercambios no simultáneos.

## 4. Metodología y Tecnología

### 4.1 Metodología

La aplicación se desarrollará con una arquitectura por capas (Capa Presentación, Capa Controlador, Capa Dominio, Capa Persistencia).

La aplicación será multiusuario y se ejecutará online, cuya interfaz estará representada como una página web alojada en un servidor.

La comunicación con el servidor se realizará basándose en la arquitectura Modelo-Vista-Controlador y se implementará mediante **Struts** [20], [12], [1].

La base de datos será referencial y estará alojada en un servidor **Oracle** [19].

Para la realización de este proyecto se ha seguido las pautas del **Proceso Unificado**, que consiste en el desarrollo software iterativo e incremental, aprovechando el uso de patrones y permitiendo depurar los requisitos del sistema en cada etapa del desarrollo del proyecto [21].

Se utilizarán términos y artefactos propios de este proceso de desarrollo de software para realizar el proyecto.

Para identificar los requisitos funcionales se utilizarán los **Casos de Uso**, que serán implementados cada uno por un *Action* siguiendo la arquitectura basada en Struts. Por lo tanto tendremos tantos *ActionForm* y *Action* como Casos de Uso hayamos identificado.

El software para desarrollar la aplicación es **NetBeans 6.01**. Se aplicarán los patrones básicos de diseño software para la implementación de la aplicación como son los patrones **GRASP** y **GoF** [21].

El lenguaje de programación será Java, apoyado de JSP, JSTL y Javascript para la implementación de la interfaz. La base de datos se implementará con SQL y Oracle.

## 4.2 Herramientas de Desarrollo

<b>Herramientas de Entorno</b>	<ul style="list-style-type: none"> <li>• PC: PIV 3000/120GB/512MB</li> <li>• Microsoft Windows 2000 Professional SP4</li> <li>• Microsoft Office Professional</li> <li>• Firefox 3</li> </ul>
<b>Herramientas de Diseño</b>	<ul style="list-style-type: none"> <li>• Corel Paint Shop Pro X</li> <li>• Corel DRAW 12</li> <li>• Macromedia Dreamweaver</li> <li>• Gliffy</li> <li>• Microsoft Project 2003</li> </ul>
<b>Herramientas de Programación</b>	<ul style="list-style-type: none"> <li>• NetBeans IDE 6.0.1</li> <li>• API Google Maps</li> </ul>
<b>Herramientas de Base de Datos</b>	<ul style="list-style-type: none"> <li>• SQL Developer</li> <li>• Power Designer 12</li> </ul>

Fig 2. Herramientas para desarrollar el proyecto.



10 5. Planificación

i	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
ii	Análisis de Requisitos Funcionales	7 días?	lun 02/03/09	mar 10/03/09		Jefe de Proyecto
	Escritura de los Casos de Uso	15 días?	mié 11/03/09	mar 31/03/09	1	Jefe de Proyecto
	Análisis de Requisitos no Funcionales	5 días?	mié 11/03/09	mar 17/03/09	1	Jefe de Proyecto
	Diseño Arquitectura Software	3 días?	mié 18/03/09	vie 20/03/09	3	Jefe de Proyecto
	Diseño de la GUI	9 días?	lun 23/03/09	jue 02/04/09	4	Jefe de Proyecto
	Diseño del Domino	8 días?	vie 03/04/09	mar 14/04/09	5	Jefe de Proyecto
	Diseño de la Base de Datos	5 días?	mié 15/04/09	mar 21/04/09	6	Jefe de Proyecto
	Implementación de la GUI	10 días	mié 22/04/09	mar 05/05/09	7	Jefe de Proyecto
	Implementación del Dominio	15 días?	mié 06/05/09	mar 26/05/09	8	Jefe de Proyecto
	Implementación de la Base de Datos	10 días?	mié 27/05/09	mar 09/06/09	9	Jefe de Proyecto
	Testeo	5 días?	mié 10/06/09	mié 17/06/09	10	Jefe de Proyecto

**Fig B** – Listado de tareas del proyecto

## 6. Análisis

### 6.1 Requisitos. Casos de Uso.

A continuación se redactan los Casos de Uso, que describen las funcionalidades (o requisitos funcionales) de la aplicación.

Aquellos casos de uso que precisan de una descripción más detallada debido a su complejidad o peculiaridad están redactados en un formato más extenso que explica paso a paso el escenario y la interacción del Sistema con los actores.

Para esta aplicación distinguiremos dos perfiles

- **Visitante:** No está registrado en la web. Limitado a unos casos de uso.
- **Usuario:** Está registrado en la web. Puede acceder a todos los casos de uso.

El objetivo de nuestro proyecto es crear una red social, es decir, incentivar el concepto de comunidad, por lo tanto la aplicación estará enfocada a dar privilegios y ventajas a los usuarios registrados.

## CASOS DE USO

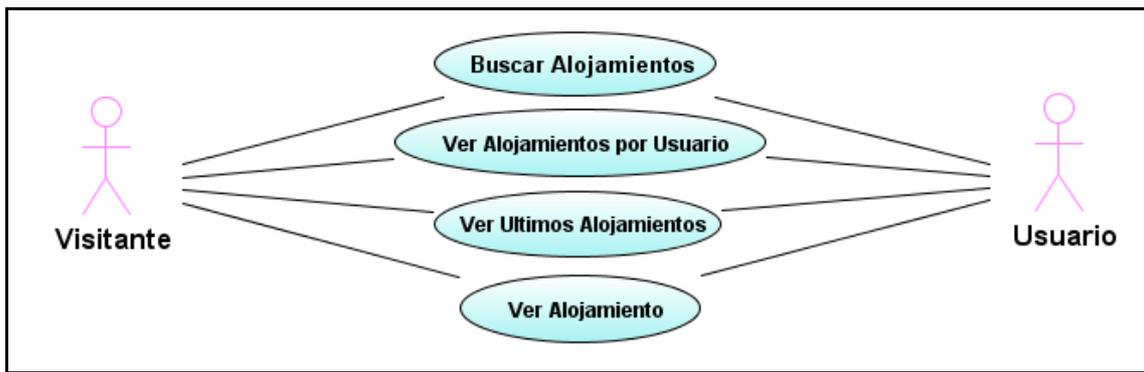


Fig. 3 - Diagrama de los casos de uso relacionados con la Consulta de Alojamiento.

### Caso de Uso: Buscar alojamiento

Un usuario introduce información sobre el alojamiento que busca. El sistema busca los alojamientos que se corresponden con los requisitos de búsqueda y se los presenta al usuario.

### Caso de Uso: Ver alojamientos por Usuario

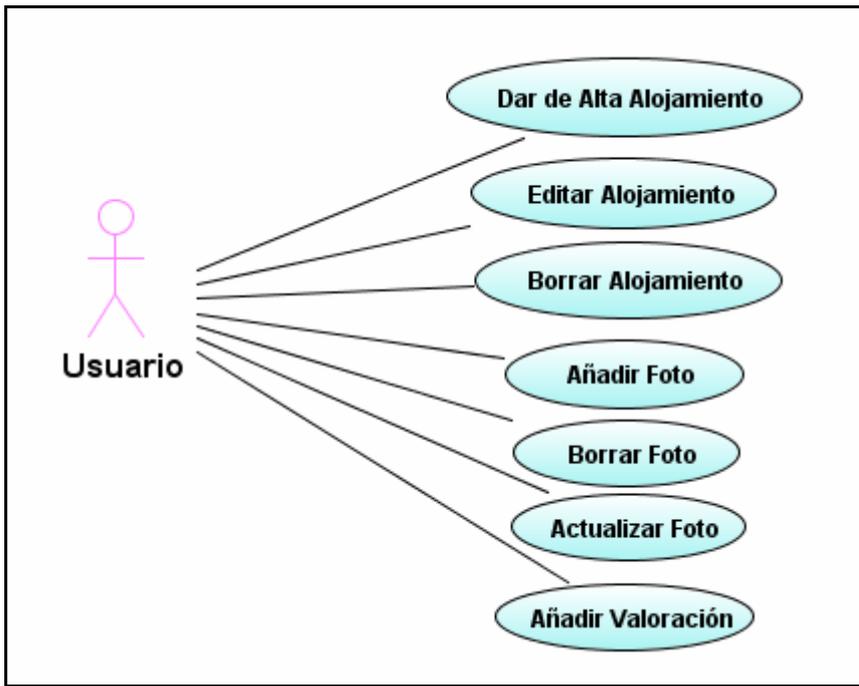
Un Usuario quiere consultar todos los alojamientos que tiene dados de alta en la red. El Sistema muestra al Usuario todos sus alojamientos., incluyendo las reservas que se han realizado sobre éstos (con su estado, nombre del solicitante, alojamiento intercambiado en caso de ser un intercambio y las fechas solicitadas.)

### Caso de Uso: Ver Últimos Alojamiento

Un Usuario quiere una lista de los últimos alojamientos que se han dado de alta. El Sistema muestra los últimos alojamientos dados de alta (10 días).

### Caso de Uso: Ver alojamiento

Un Usuario quiere ver toda la información relacionada con un alojamiento, como el tipo de alojamiento, país, provincia, disponibilidad, características, ubicación geográfica, imágenes, valoraciones, comentarios, etc. El Sistema muestra toda la información de un alojamiento.



**Fig. 4** - Diagrama de los casos de uso relacionados con la Manipulación de Alojamiento.

**Caso de Uso:** Dar de Alta alojamiento

Un Usuario introduce los datos de un alojamiento que quiere dar de alta en la red. El Sistema registra un nuevo alojamiento y lo asocia al usuario.

**Caso de Uso:** Editar alojamiento

Un Usuario quiere editar/actualizar la información de un alojamiento que tiene publicado. El Sistema muestra la información de alojamiento. El Usuario modifica los datos. El Sistema aplica los cambios que el Usuario ha indicado y los registra.

**Caso de Uso:** Borrar alojamiento

Un Usuario quiere dar de baja un alojamiento que tiene publicado en la red. El Sistema elimina el alojamiento publicado por un Usuario concreto.

**Caso de Uso:** Añadir foto

Un Usuario quiere añadir una foto a uno de sus alojamientos. El Usuario notifica al Sistema la nueva foto que quiere añadir. El Sistema añade la foto al alojamiento de ese usuario y muestra la nueva foto.

**Caso de Uso:** Actualizar foto

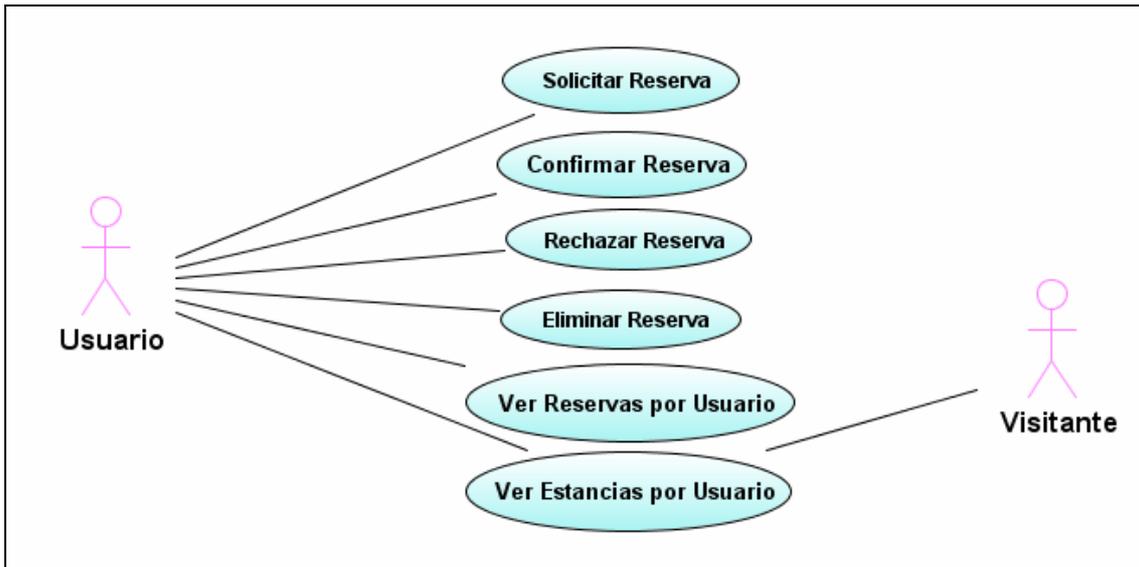
Un Usuario quiere actualizar una de las fotos que muestra de su alojamiento. El usuario notifica al Sistema la nueva foto. El Sistema actualiza la foto correspondiente y muestra al usuario la modificación realizada.

**Caso de Uso:** Borrar foto

Un usuario quiere borrar una foto de uno de sus alojamientos. El Usuario indica al Sistema cuál es la foto que quiere borrar. El Sistema borra la foto asociada al alojamiento de ese usuario y notifica los cambios.

**Caso de Uso:** Añadir valoración

Un usuario quiere valorar un alojamiento. El usuario puntúa y comenta un alojamiento. El Sistema comprueba que el Usuario ha estado alojado en ese alojamiento y añade la valoración al alojamiento. El Sistema muestra el alojamiento actualizado con la valoración añadida.



**Fig. 5** - Diagrama de los casos de uso relacionados con las Reservas y Estancias.

**Caso de Uso:** Rechazar reserva

Un usuario quiere rechazar una Reserva Solicitada para uno de sus alojamientos por parte de otro usuario. El usuario rechaza la reserva. El Sistema registra la reserva como Reserva Rechaza. El Sistema muestra al usuario la reserva con su estado actualizado.

**Caso de Uso:** Eliminar reserva

Un usuario quiere eliminar una reserva de uno de sus alojamientos. El usuario elimina la reserva. El Sistema elimina la reserva de ese alojamiento y muestra al usuario el alojamiento actualizado, sin la reserva eliminada.

**Caso de Uso:** Ver reservas por Usuario

Un usuario quiere ver las reservas que ha realizado. El Sistema muestra al usuario todas sus reservas realizadas en alojamientos de otros usuarios, con su estado, las fechas solicitadas, el país y el alojamiento relacionado.

**Caso de Uso:** Solicitar reserva

Un usuario quiere solicitar una reserva para un alojamiento. El usuario escoge una fecha de entrada y de salida. El Sistema registra una nueva reserva solicitada para ese alojamiento.

**Precondiciones:** El Usuario conoce las fechas disponibles para ese alojamiento porque lo está visualizando.

**Postcondiciones:** La lista de reservas del usuario se actualiza con una reserva nueva solicitada.

**Escenarios**

*Flujo Básico*

1. El Usuario quiere solicitar una reserva para un alojamiento.
2. El Usuario introduce las fechas en las que desea hacer la reserva (fecha de entrada y fecha de salida).
3. El Sistema comprueba que las fechas solicitadas están dentro de la fecha de disponibilidad y no están ocupadas por reservas confirmadas de otros usuarios.
4. El Sistema crea y añade una nueva Reserva Solicitada al alojamiento, con el usuario solicitante, las fechas solicitadas y el alojamiento propuesto en caso de intercambio.
5. El Sistema registra la reserva y muestra al usuario la información de contacto del propietario.

### *Flujos Alternativos*

2a. El Usuario quiere proponer en la reserva un alojamiento para un intercambio (**Precondición:** El propietario del alojamiento está interesado en intercambios y el Sistema muestra la opción de proponer intercambio).

1. El Usuario solicita al Sistema que quiere proponer un intercambio.
2. El Sistema muestra al Usuario sus alojamientos.
3. El Usuario selecciona uno de sus alojamientos para proponerlo como intercambio y lo notifica al Sistema.
4. El Sistema memoriza el alojamiento propuesto.
5. El Sistema vuelve a mostrar el alojamiento que el usuario quería reservar, informando del alojamiento escogido para el intercambio.

*Continúa en el paso 3 del Flujo Básico.*

3a. Las fechas solicitadas no están disponibles.

1. El Sistema notifica al Usuario que las fechas solicitadas no están disponibles.

### **Caso de Uso:** Ver estancias por Usuario

Un usuario quiere ver las estancias de un usuario concreto. El Sistema muestra la lista de las estancias del usuario especificado, con el alojamiento asociado, el alojamiento intercambiado en caso de ser intercambio y las fechas de estancia.

**Caso de Uso:** Confirmar reserva

Un usuario quiere confirmar una Reserva Solicitada para uno de sus alojamientos por parte de otro usuario. El Sistema actualiza la reserva y la registra como Reserva Confirmada. El Sistema muestra al usuario la reserva con su estado actualizado.

**Precondiciones:** El usuario tiene una reserva Solicitada para uno de sus alojamientos.

**Postcondiciones:** La reserva queda confirmada. En caso de intercambio, se crea una reserva para el alojamiento de intercambio y se confirma. Se obtienen dos reservas confirmadas, ambas con alojamiento de intercambio, que representa el intercambio de alojamientos entre dos usuarios.

**Escenarios**

*Flujo Básico*

1. El Usuario notifica al Sistema que quiere confirmar uno las reservas que han solicitado para uno de sus alojamientos (Alojamiento A)
2. El Sistema actualiza el estado de la reserva.
3. El Sistema muestra al usuario la reserva actualizada.

*Flujos Alternativos*

- 2a. La reserva tiene un alojamiento propuesto para intercambio (Alojamiento B)
  1. El Sistema memoriza el Alojamiento A como alojamiento propuesto para intercambio.
  2. El Sistema muestra la información y formulario de reserva del Alojamiento B para que el usuario pueda realizar una reserva.
  3. El Usuario selecciona las fechas deseadas para la reserva del Alojamiento B.
  4. El Sistema crea y añade una nueva Reserva Solicitada al Alojamiento B, con el usuario solicitante, las fechas solicitadas y el Alojamiento A como alojamiento de intercambio.
  5. El Sistema confirma la reserva del alojamiento A y la reserva del alojamiento B.

*Continúa en el paso 3 del Flujo Básico*

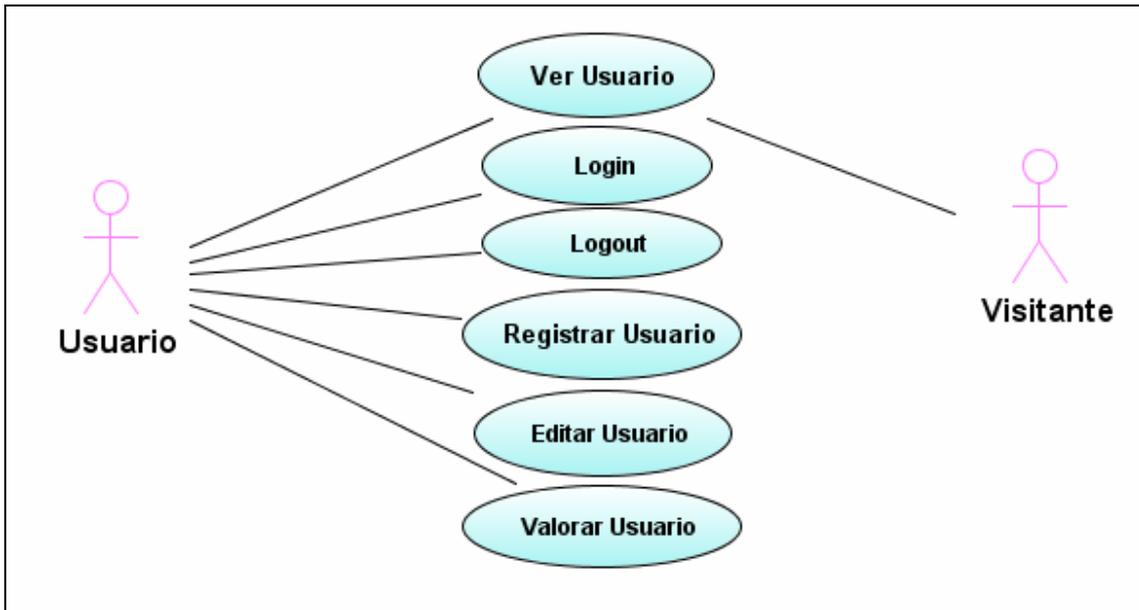


Fig. 6 - Diagrama de los casos de uso relacionados con el Usuario.

**Caso de Uso:** Ver usuario

Un usuario quiere ver información de otro usuario. El Sistema muestra la información del usuario solicitado.

**Caso de Uso:** Login

Un usuario quiere identificarse en el Sistema. El usuario introduce su nombre de usuario y contraseña. El Sistema valida el nombre de usuario y la contraseña. El Sistema identifica al usuario y lo guarda en la sesión. El Sistema muestra las funciones disponibles para usuarios registrados.

**Caso de Uso:** Logout

Un usuario quiere salir de la sesión. El usuario al Sistema que quiere desloguearse. El Sistema elimina la sesión del usuario.

**Caso de Uso:** Registrar Usuario

Un Usuario quiere registrarse en el Sistema. El Usuario introduce los datos de registro (nombre, apellidos, teléfono, e-mail, nombre de usuario, contraseña y país preferente para intercambio si es el caso). El Sistema valida los datos introducidos por el Usuario. El Sistema crea y registra un nuevo usuario con los datos suministrados.

**Caso de Uso:** Editar Usuario

El usuario quiere modificar sus datos. El Sistema muestra al usuario sus datos para que los modifique. El usuario modifica sus datos. El Sistema registra los cambios.

**Caso de Uso:** Valorar Usuario

Un usuario quiere valorar a otro usuario. El usuario indica una puntuación. El Sistema comprueba que el usuario que quiere valorar es propietario de un alojamiento en el que ha estado alojado el usuario a valorar. El Sistema recalcula la media de la valoración del usuario. El Sistema vuelve a mostrar la información del usuario actualizada.

## 6.2 Patrones

### Patrón Capas

Utilizaremos este patrón para separar la interfaz, el dominio, la lógica de control y la persistencia.

### Patrón Modelo-Vista-Controlador

Es un patrón integrado con la filosofía de Struts, en el que se separa la lógica del negocio, la interfaz y la lógica de control.

En esta aplicación la Vista es la interfaz web, el Controlador está formado por todas las clases ActionForm y Action, que se encargan de controlar y distribuir las acciones al dominio y a la persistencia.

El Modelo es la Capa de Dominio, donde están todos los objetos del negocio.

(Más información: [20],[17]).

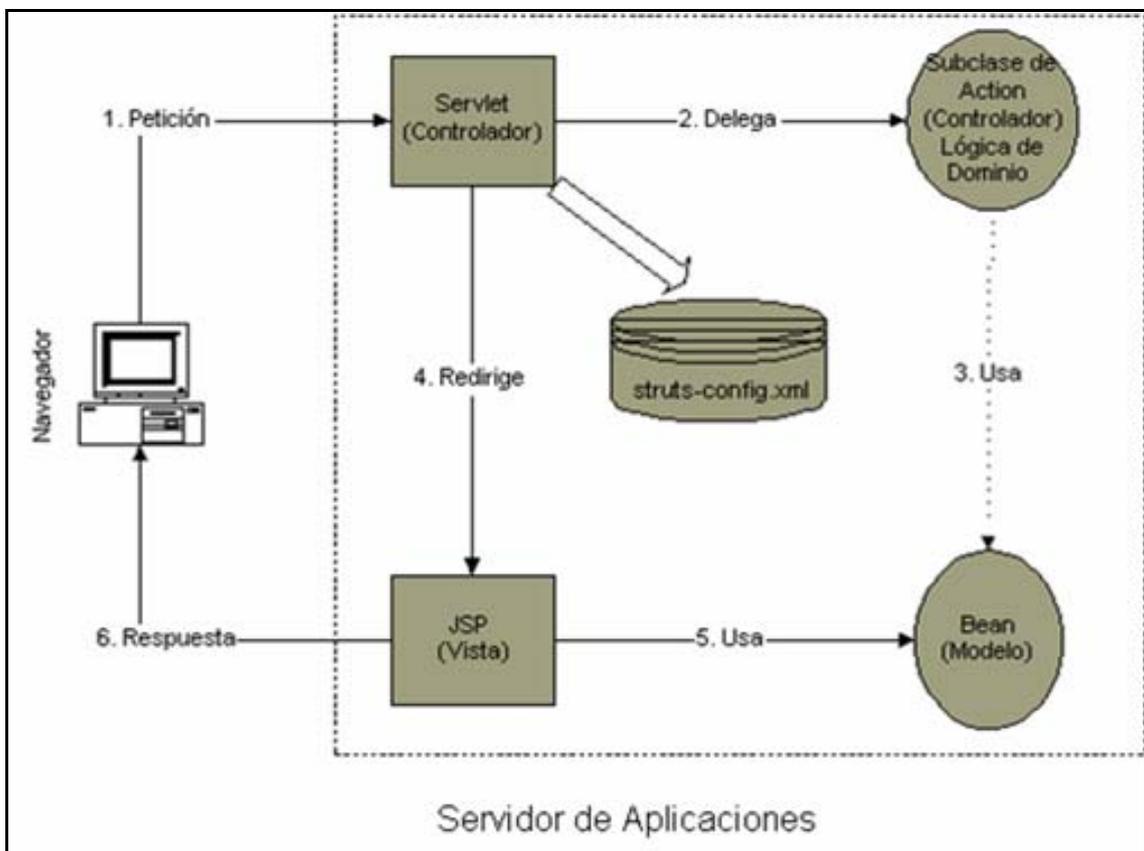


Fig. 7 - Struts y el patrón Modelo-Vista-Controlador.

### Patrón Estado

Lo utilizaremos para controlar los diferentes estados en los que se puede encontrar una Reserva.

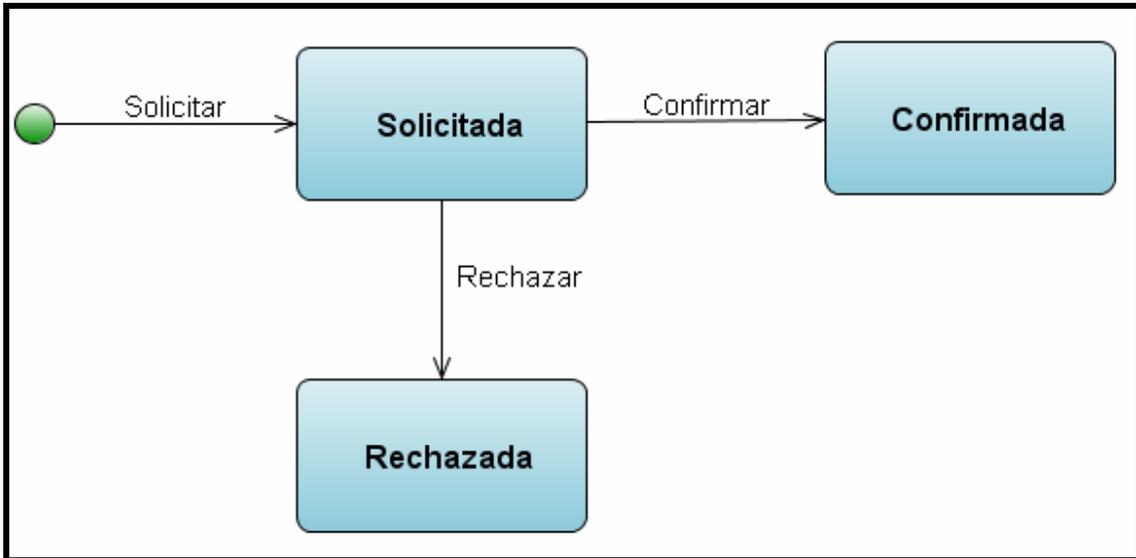


Fig. 8 - Diagrama de Estados para una Reserva.

Como podemos ver en la **Figura 8**, una Reserva se crea como Solicitada, y una vez ha sido solicitada puede ser Confirmada o Rechazada. Se quedará en uno de esos dos estados hasta que sea eliminada o se convierta en una Estancia.

### Patrón Singleton

Lo utilizaremos para todas aquellas clases de las cuales sólo necesitemos una sola instancia. Lo aplicaremos en diferentes situaciones, como a la hora de usar el patrón Factoría, las clases de la Capa de Persistencia y en el patrón Estado.

### Patrón Factoría

Lo vamos a utilizar para crear instancias de las clases de la Capa de Persistencia que se encargan de comunicarse con la base de datos. Usando este patrón tenemos una clase, que se encarga de crear los objetos de las clases de persistencia sin necesidad de que la clase desde la cual se solicitan servicios de persistencia tenga la responsabilidad de conocer esas clases y deba solamente conocer las interfaces correspondientes.

### **Patrón Transacción**

Este patrón se utiliza para controlar las transacciones y la concurrencia en la base de datos, evitando la posibilidad de que la base de datos quede en un estado inconsistente cuando más de un usuario acceden a los mismos datos al mismo tiempo.

Además de estos patrones, se han utilizado implícitamente los 5 patrones básicos GRASP (Experto en Información, Creador, Alta Cohesión, Bajo Acoplamiento y Controlador) para asignar responsabilidades a los objetos y diseñar una buena arquitectura software.

*(Más información sobre Patrones: [21]).*

### 6.3 Modelo del Negocio

El Modelo del Negocio lo utilizaremos para conocer la lógica del negocio que queremos trasladar a la aplicación.

Como podemos ver en la **Figura 9**, un Usuario posee uno o varios alojamientos. Un Alojamiento tiene como mínimo una Foto (Imagen) y puede tener ninguna o varias Valoraciones, que realizan los Usuarios.

A su vez, un Alojamiento puede tener Reservas realizadas por Usuarios.

Las Estancias las realiza un Usuario sobre un Alojamiento y sirven como historial de las reservas que ya se han cumplido.

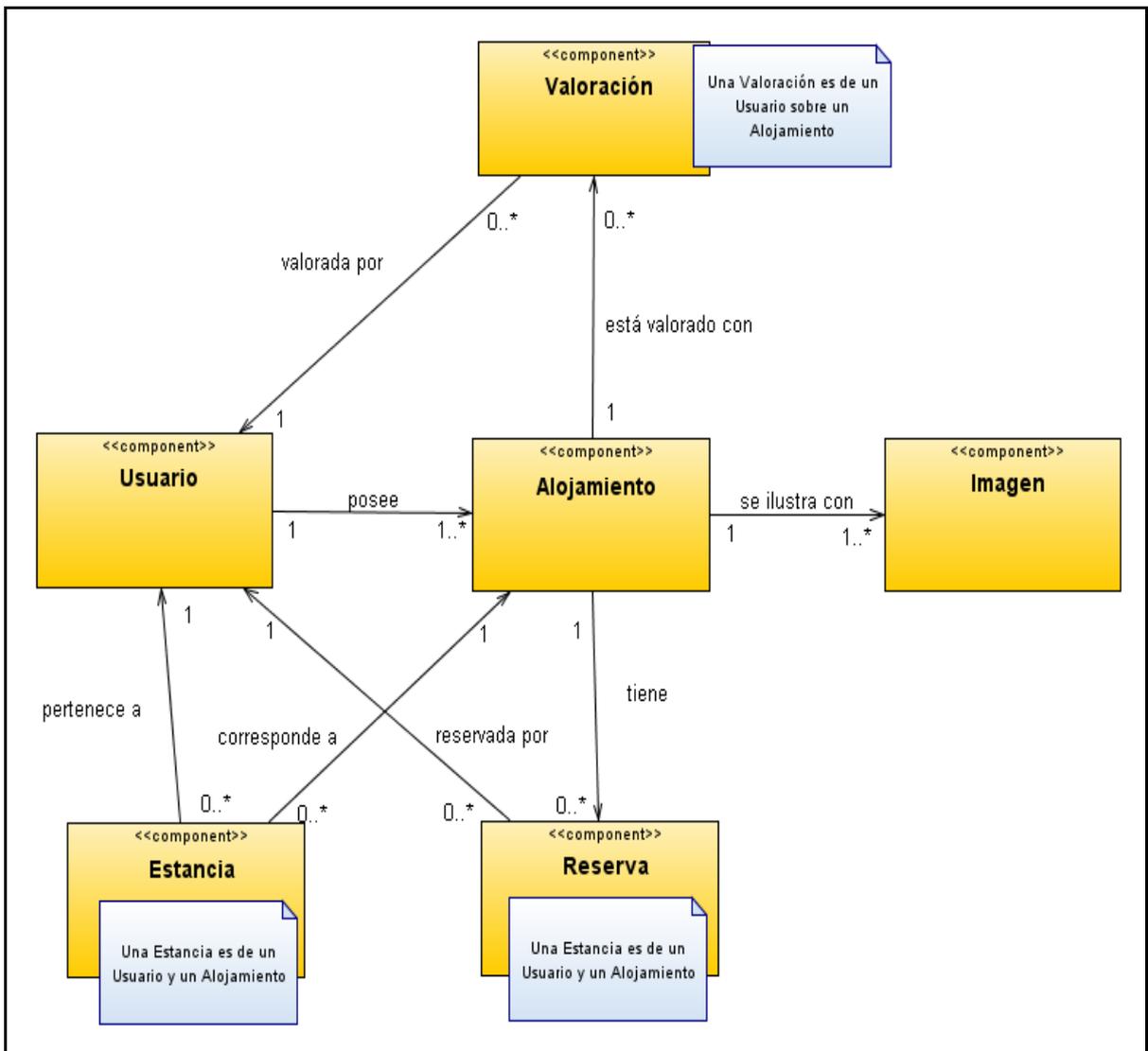


Fig. 9 - Modelo del Negocio de la aplicación a desarrollar.

## 6.4 Reglas de Negocio

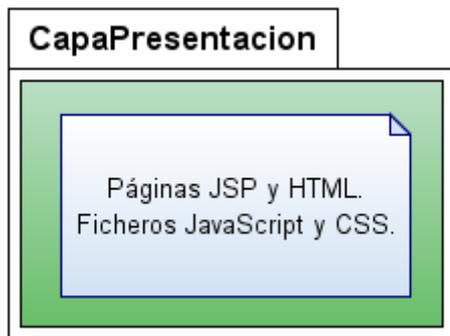
En este apartado se describen las reglas y restricciones propias del negocio que la aplicación deberá respetar.

REGLA 1	<p>Un usuario puede solicitar reservas tantas veces como haya ofrecido alojamiento. Es decir, si el número de veces que ha aceptado alojamiento es menor o igual que el número de veces que ha sido aceptado.</p> <p>Esta limitación sirve para evitar el abuso y que haya un verdadero intercambio de alojamientos entre usuarios.</p>
REGLA 2	Un alojamiento debe tener vinculada como mínimo una foto.
REGLA 3	Sólo se pueden solicitar reservas con intercambio a alojamientos cuyo propietario esté interesado en intercambios.
REGLA 4	Las fechas solicitadas para una reserva no deben estar ocupadas por otras reservas y deben estar dentro de las fechas de disponibilidad del alojamiento.
REGLA 5	Sólo los usuarios que han estado alojados en un alojamiento pueden añadir una valoración sobre el mismo.
REGLA 6	Un usuario no puede solicitar una reserva para un alojamiento si ya tiene una solicitada pendiente de ser confirmada o rechazada para el mismo alojamiento.
REGLA 7	La fecha de entrada no puede ser posterior a la fecha de salida.
REGLA 8	Un usuario sólo puede ser valorado por el propietario del alojamiento donde éste ha estado alojado.
REGLA 9	No se pueden modificar las fechas de disponibilidad de un alojamiento si tiene reservas solicitadas.

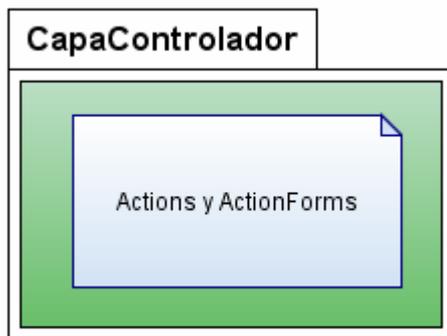
**Fig.10** - Tabla de reglas de negocio.

## 7. Diseño

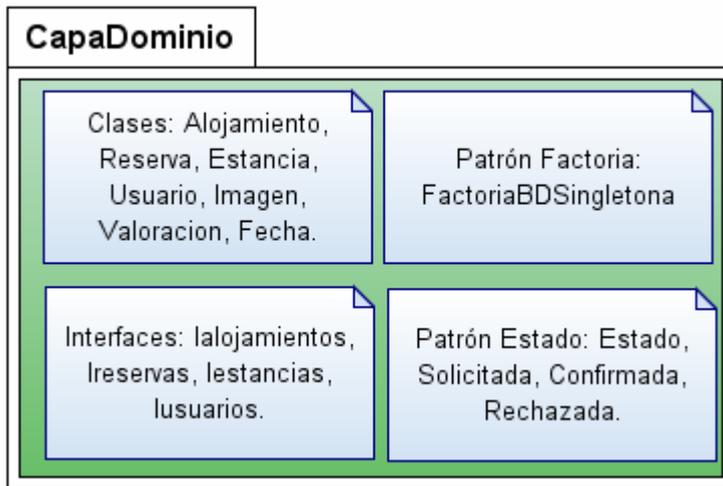
A continuación se describe en contenido de cada una de las capas de las que se compone la arquitectura software de la aplicación



Contiene los archivos HTML, JSP, JavaScript y CSS que conforman la página web.

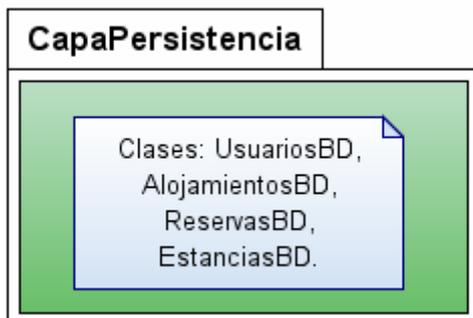


Contiene las clases *ActionForm* encargadas de recoger las peticiones del usuario desde Presentación y las clases *Action* encargadas de comunicarse con los objetos del Dominio y la Persistencia para efectuar las acciones solicitadas por el usuario.



Contiene las clases de la lógica del negocio.

También es donde se almacena la factoría de instancias de clases de Persistencia y las interfaces para acceder a éstas.



Contiene las clases que se encargan de realizar las operaciones sobre la base de datos.

La arquitectura software diseñada funciona de la siguiente manera:

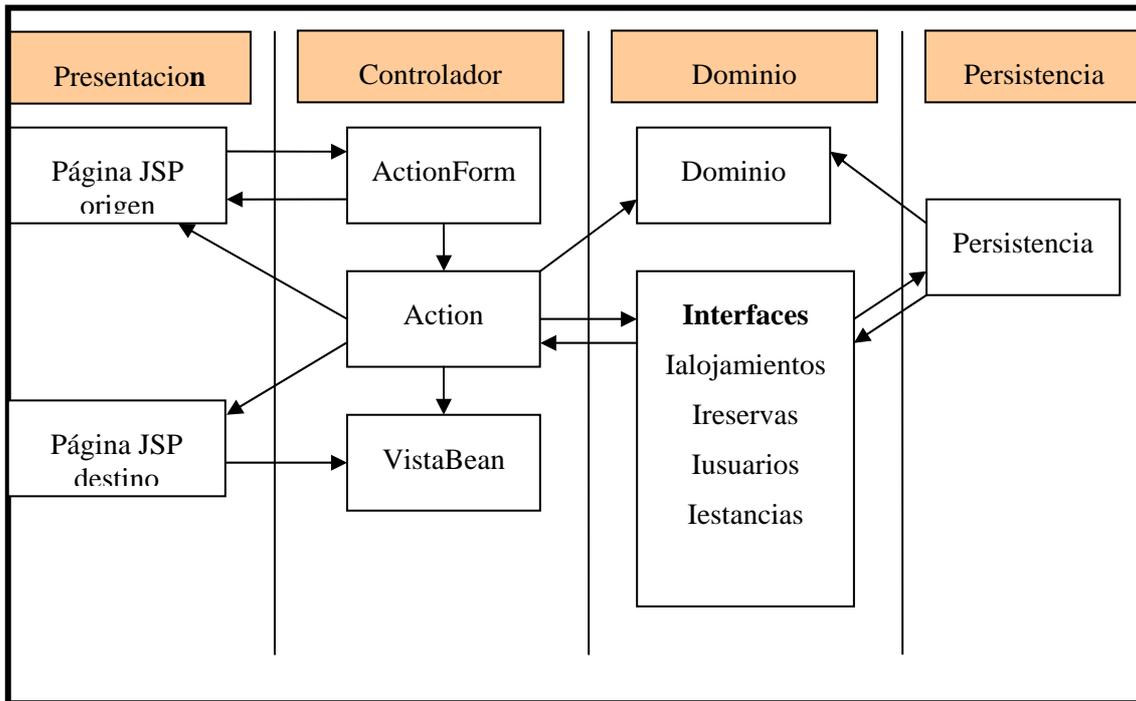


Fig. A – Funcionamiento y Comunicación entre Capas.

La página JSP mediante un formulario u otro mecanismo realiza una petición a la clase ActionForm correspondiente, que almacena los datos de la petición.

La clase Action se encarga de recoger esos datos y ejecutar la acción correspondiente utilizando los objetos del Dominio y de la Persistencia (mediante las interfaces predefinidas para ello: *Ialojamientos*, *Ireservas*, *Iusuarios*, *Iestancias*).

La Persistencia utiliza los objetos del Dominio y devuelve los resultados (si es el caso) mediante las mismas interfaces.

La clase Action recibe los datos y envía la página JSP que debe visualizarse. Si necesita almacenar la información en la sesión utiliza la clase VistaBean, un *bean* alojado en la sesión al que acceden las páginas JSP para mostrar los resultados que ha procesado la clase Action.

## 7.1 GUI

En este apartado se especifica el diseño de la Interfaz Gráfica de Usuario para la aplicación.

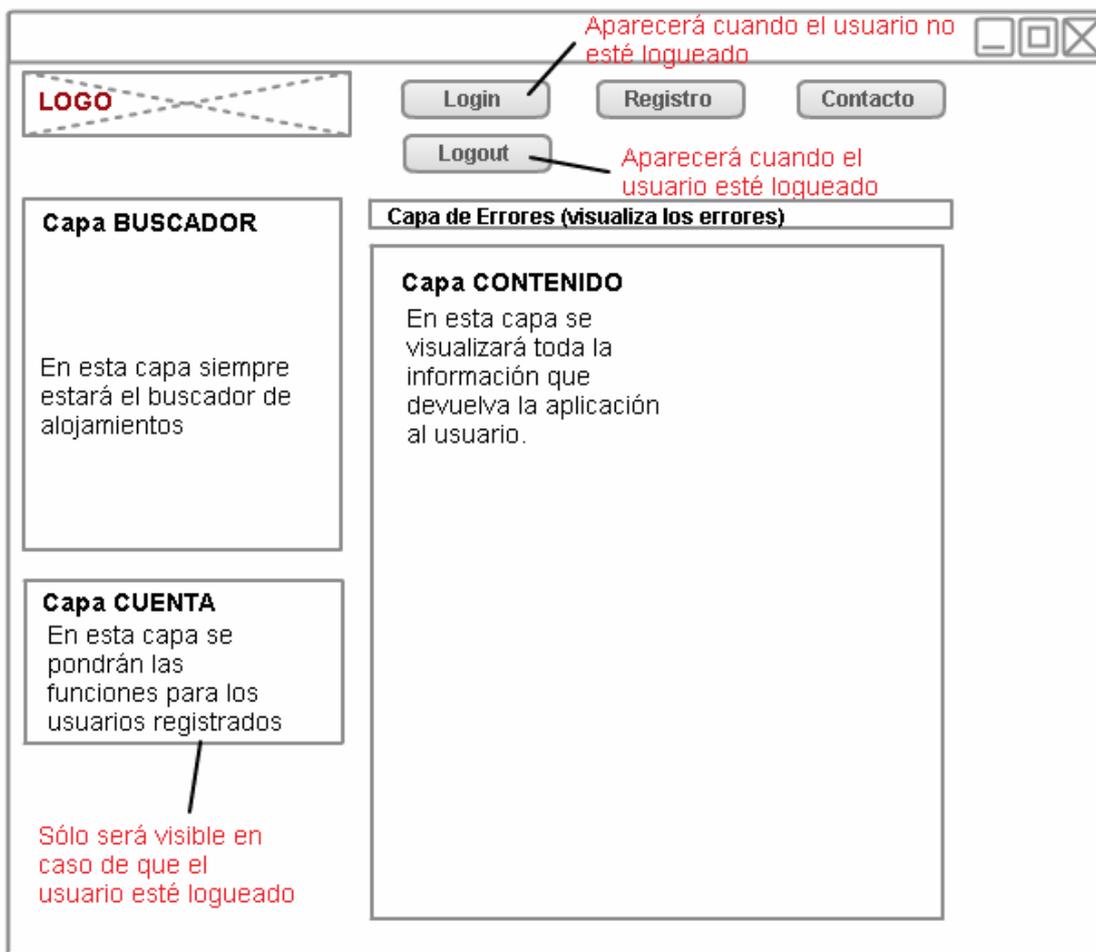
La web está optimizada para una resolución a partir de 800x600 y bajo el explorador de internet Firefox.

**Nota:** Los comentarios en color rojo son explicaciones sobre las funcionalidades o características de algunos elementos. Los hipervínculos (enlaces) se representan subrayados y en color azul.

Los siguientes diseños son prototipos de la interfaz gráfica, no reflejan necesariamente el aspecto final de la interfaz.

### Estructura base

Esta es la estructura base de la web, estructurada en capas y tablas.



**Fig. 11** - Diseño de la estructura base de la web.

Está formada por 5 capas.

Una capa para la **Cabecera**, que siempre estará visible y en la que pondremos el logo de la web en la esquina superior izquierda, junto a dos opciones orientadas al usuario: identificación en la aplicación, registro de nuevo usuario e información de contacto sobre la web. La capa de **Errores** visualizará los errores que se produzcan.

La otra capa es la del **Buscador**, donde tendremos un formulario de búsqueda que también estará visible siempre en la parte izquierda de la pantalla. El motivo por el que el buscador ocupa ese lugar se debe a que comúnmente los menús de los sitios web se encuentran en ese lugar y los usuarios están acostumbrados a ello. Dada la importancia que se le quiere dar a la búsqueda de alojamientos, colocándolo en esa posición al usuario le será intuitivo y cómodo encontrarlo.

El formulario de búsqueda, titulado 'BUSCADOR', contiene los siguientes campos:

- país: campo de selección con una flecha hacia abajo.
- provincia: campo de selección con una flecha hacia abajo.
- fecha entrada: campo de texto con un icono de calendario.
- fecha salida: campo de texto con un icono de calendario.
- nº habitaciones: campo de selección con una flecha hacia abajo y el texto '1 - 5' a su derecha.
- compartido: campo de selección con una flecha hacia abajo y el texto 'Sí / No' a su derecha.
- país de intercambio: campo de selección con una flecha hacia abajo.

Debajo de los campos se encuentra un botón rectangular con el texto 'Buscar'.

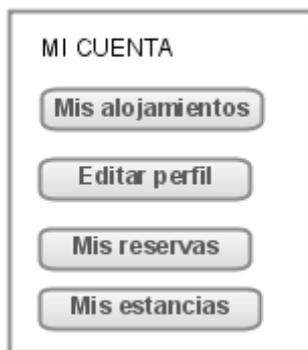
**Fig. 12** - Contenido de la Capa Buscador. Formulario de búsqueda de alojamientos.

El formulario sólo requiere que como mínimo se especifique el País, el resto de parámetros se pueden omitir o combinar como se prefiera, ajustándose a las necesidades de búsqueda de cada usuario.

El *País de Intercambio* permite buscar alojamientos cuyo propietario esté interesado en el país que se haya indicado, de forma que se facilite la búsqueda para aquellos usuarios interesados en intercambios.

La siguiente capa es la capa **Cuenta**, que sólo se visualiza a usuarios registrados y una vez se hayan identificado en la web. Estará posicionado también a la izquierda, justo debajo del buscador porque se considera el sitio más cómodo e intuitivo.

Siempre estará visible (salvo si el usuario se *desloguea*) y permite a un usuario consultar sus alojamientos publicados, editar su información (perfil) y consultar sus reservas o estancias.



**Fig. 13** - Contenido de la Capa Cuenta. Menú para usuarios registrados

Por último tenemos la capa **Contenido**, en la parte central de la web, que es la que se encarga de visualizar todos los resultados y el resto de funcionalidades de la aplicación.

## Pantalla “Ver alojamiento”

Esta pantalla visualizará toda la información de un alojamiento, incluyendo fotos y comentarios de inquilinos.

Visualiza además las funciones necesarias para realizar reservas con o sin intercambio, seleccionando fechas de entrada y de salida deseadas, así como la posibilidad de comentar el alojamiento y añadirle una puntuación (sólo para inquilinos).

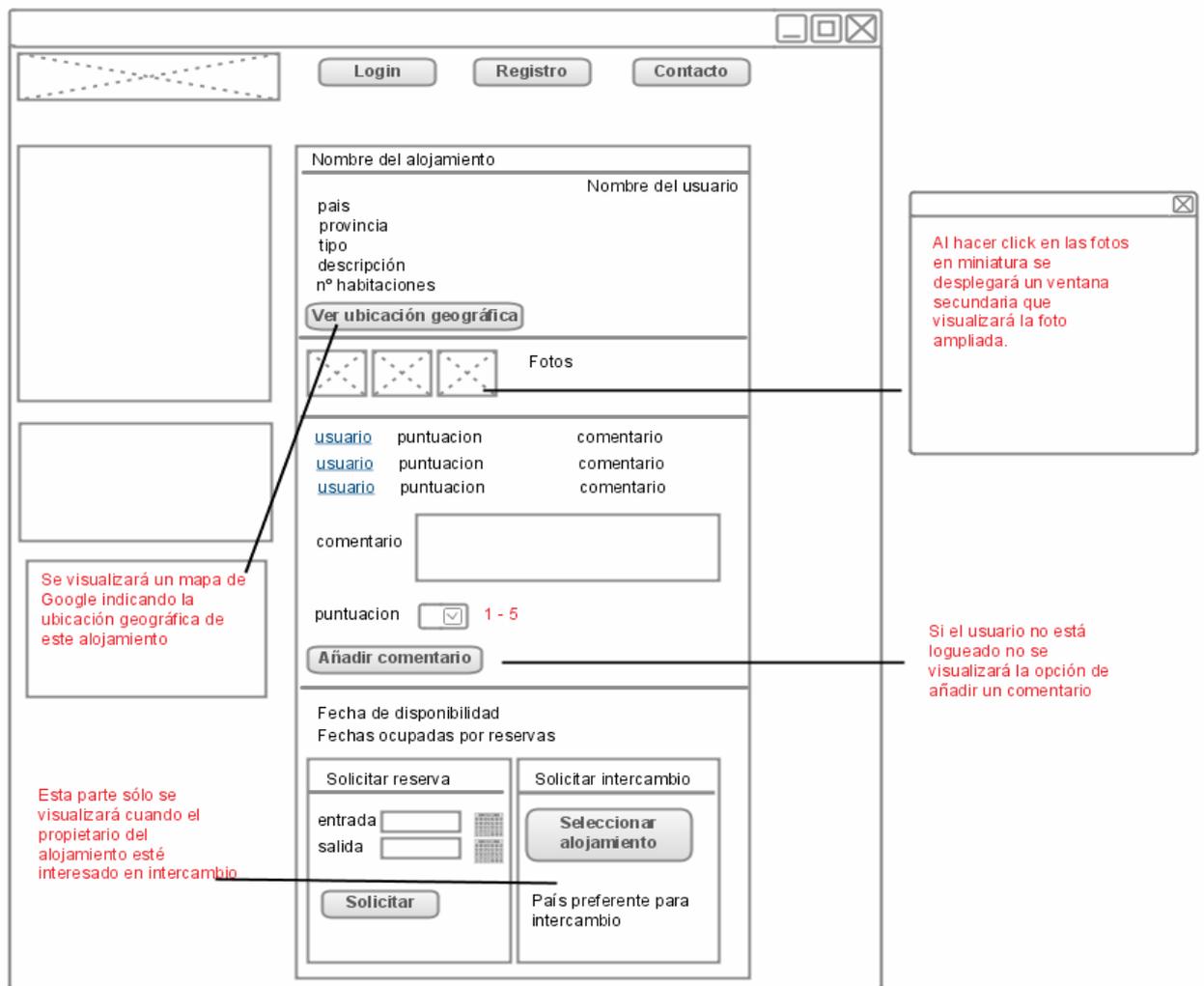


Fig. 14 - Diseño de la pantalla que muestra la información de un alojamiento.

### Pantalla “Mis alojamientos”

Esta pantalla permite gestionar los alojamientos publicados por un usuario. Permite consultar, editar y borrar un alojamiento. Además por cada alojamiento también se visualizan las reservas, permitiendo consultarlas, confirmarlas, rechazarlas o borrarlas.

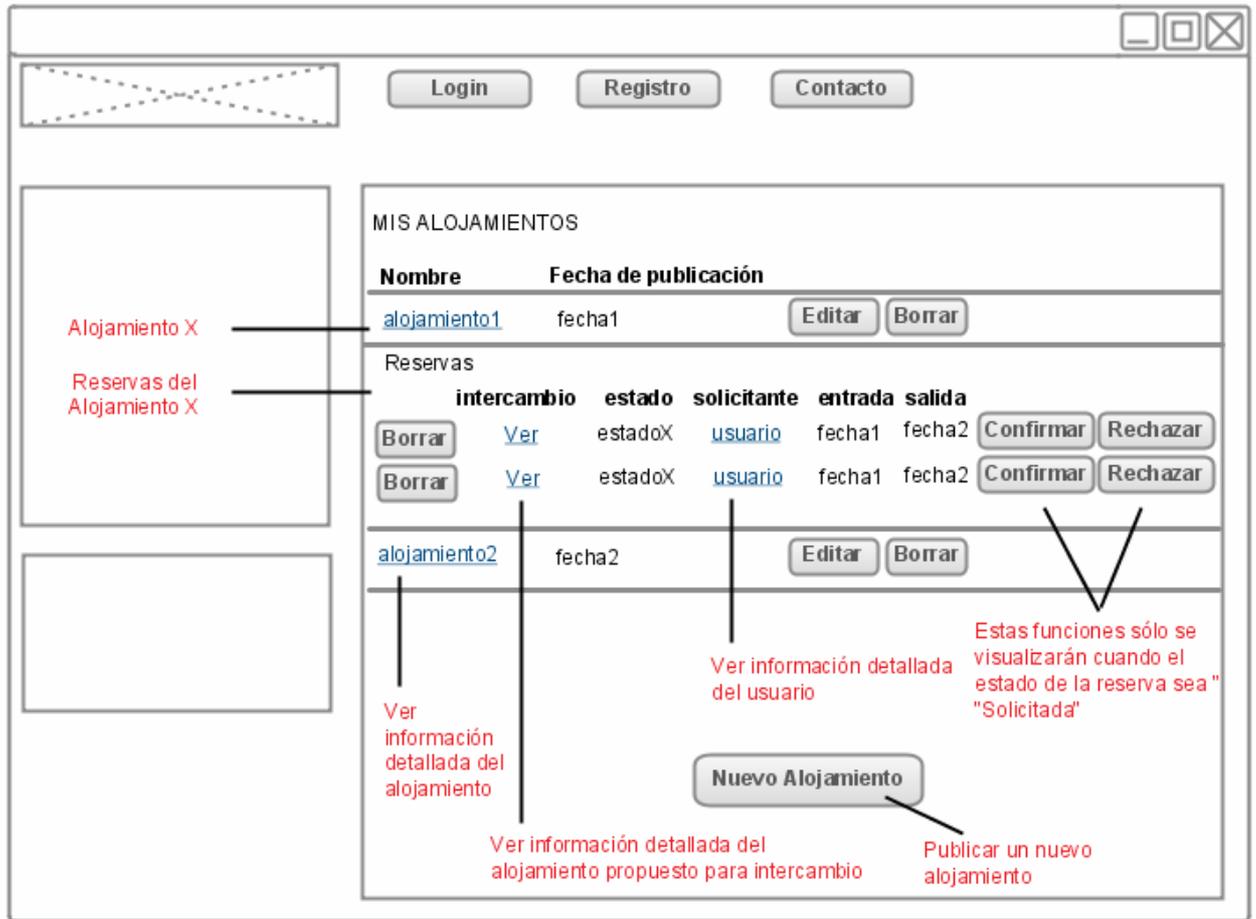


Fig. 15 - Diseño de la pantalla para gestionar los alojamientos.

## Pantalla “Resultados”

En esta pantalla se visualizan los resultados de la búsqueda. Se muestran los alojamientos directamente en un mapa de Google y debajo la lista de los alojamientos encontrados para poder consultarlos.

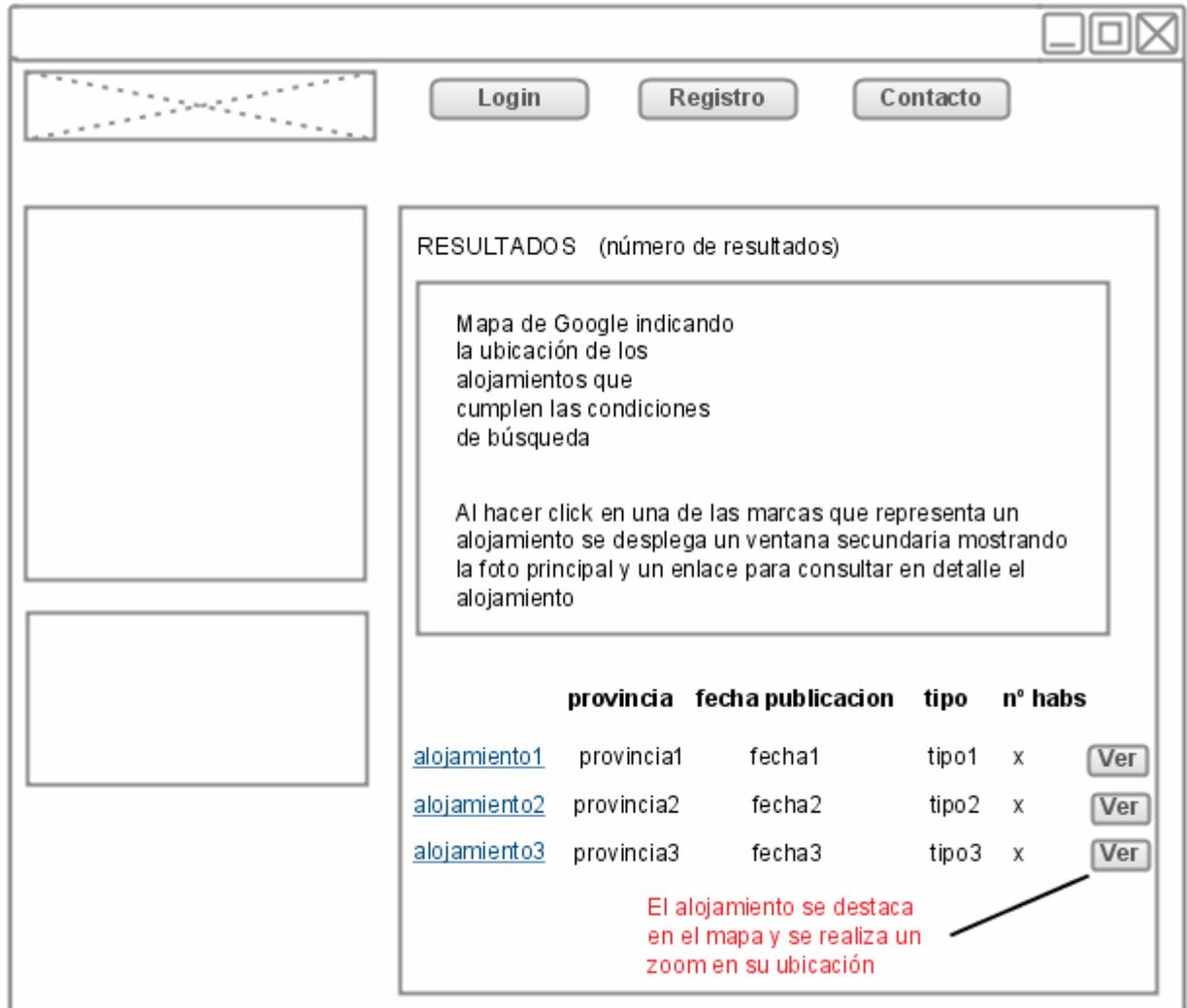
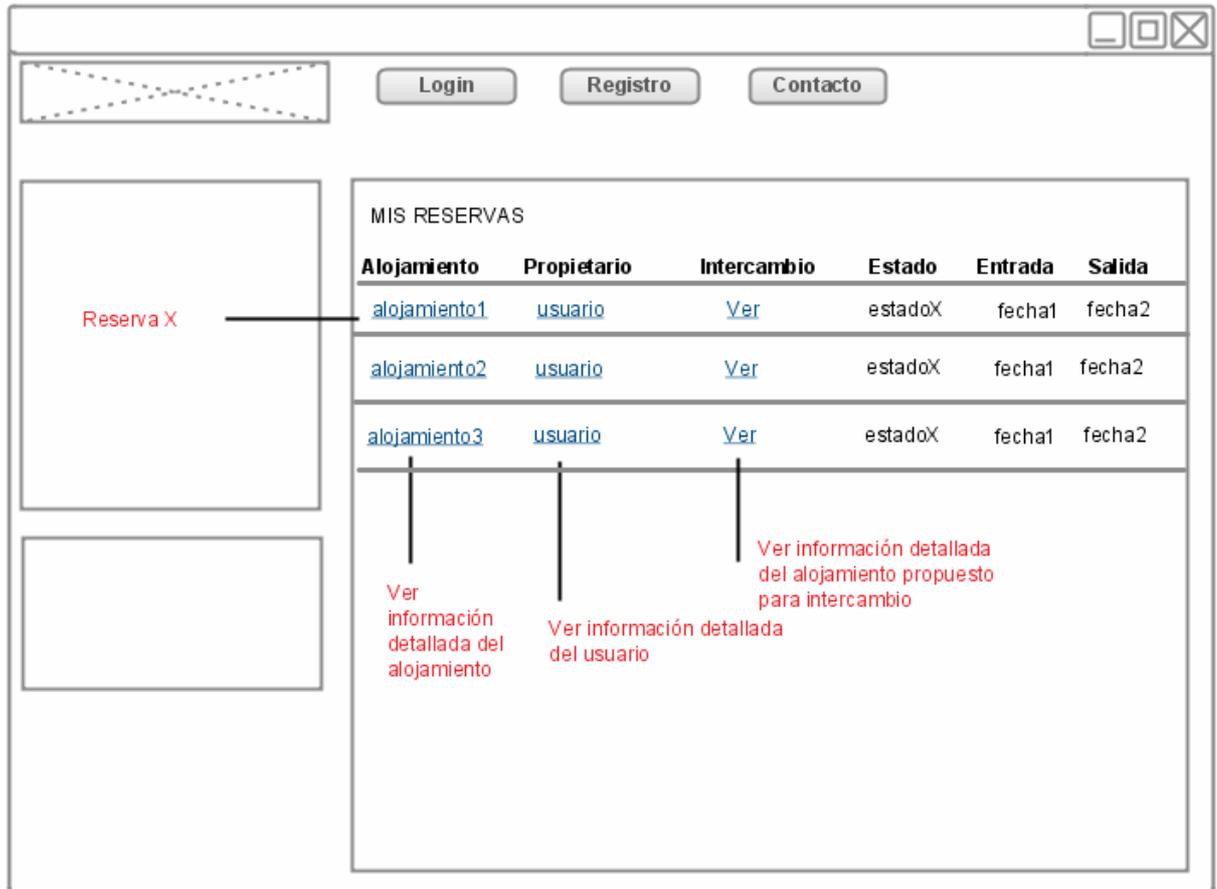


Fig. 16 - Diseño de la pantalla de resultados.

### Pantalla “Mis reservas”

En esta pantalla se visualizan todas las reservas que ha realizado el usuario, con su estado, el alojamiento relacionado, el alojamiento propuesto para intercambio si es el caso, y fechas solicitadas.



**Fig. 17** - Diseño de la pantalla que visualiza las reservas realizadas por un usuario.

## Pantalla “Alta alojamiento”

En esta pantalla se muestra el formulario para dar de alta un nuevo alojamiento.

Logout Registro Contacto

**NUEVO ALOJAMIENTO**

Nombre

País

Provincia

Fecha entrada

Fecha salida

Descripción

nº habitaciones  1 - 5

Tipo

Compartido

Foto principal

Dirección

**Geolocalizar** — A partir de la Dirección introducida se muestra un mapa de Google con la ubicación

Mapa de Google con una marca que indica la ubicación. El usuario puede reposicionar la marca para más precisión.

Enviar

Fig. 18 - Diseño de la pantalla “Alta alojamiento”

### Pantalla “Editar alojamiento”

Visualiza el formulario para modificar la información un alojamiento.

Logout Registro Contacto

**NUEVO ALOJAMIENTO**

Nombre

País  ▼

Provincia  ▼

Fecha entrada

Fecha salida

Descripción

n° habitaciones  ▼

Tipo  ▼

Compartido  ▼

Foto principal

Dirección

**Geolocalizar** A partir de la Dirección introducida se muestra un mapa de Google con la ubicación

Mapa de Google con una marca que indica la ubicación.

**Fotos**

Fig. 19 - Diseño de la pantalla “Editar alojamiento”

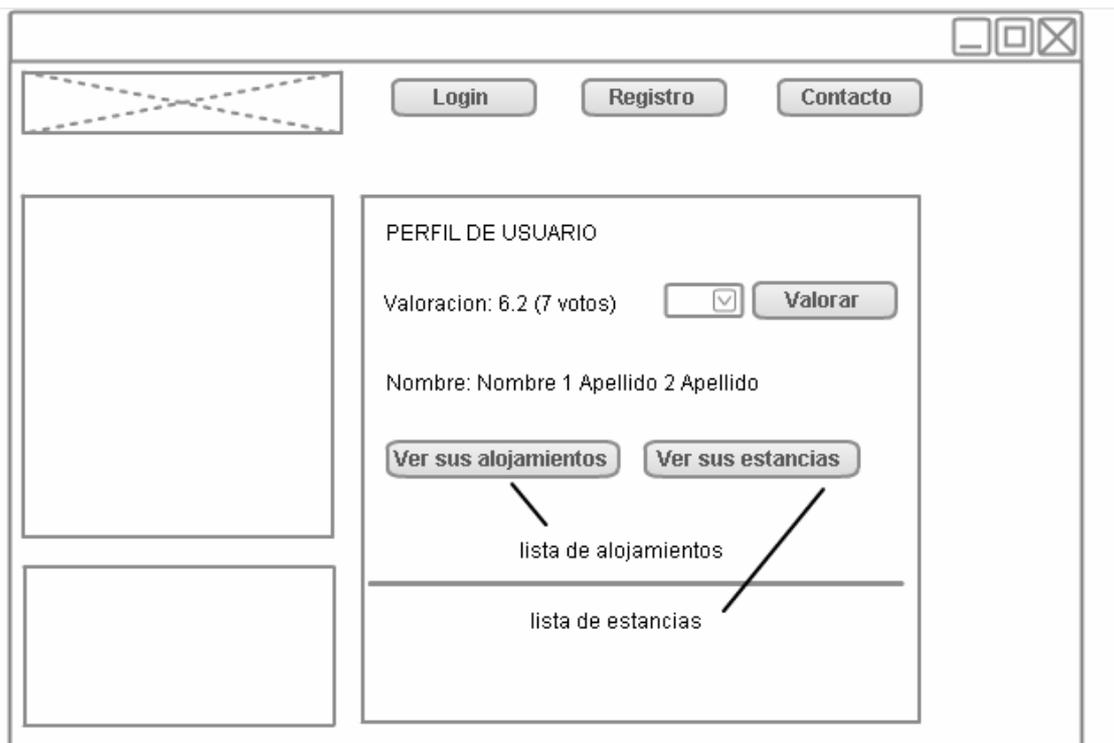
## Pantalla “Ver Usuario”

Esta pantalla muestra la información de un usuario. Visualiza también la valoración que tiene por parte de otros usuarios y el número de votos, junto a un pequeño formulario que permite valorar al usuario.

La información de contacto del usuario sólo se suministra a los usuarios que solicitan reservas en alguno de sus alojamientos por lo que en esta pantalla no se mostrará.

Un poco más abajo tenemos dos botones/enlaces, para mostrar la lista de alojamientos y de estancias de ese usuario respectivamente.

Las listas se visualizarán en esta misma pantalla.



**Fig. 19A** – Diseño de la pantalla “Ver Usuario”

## 7.2 Dominio

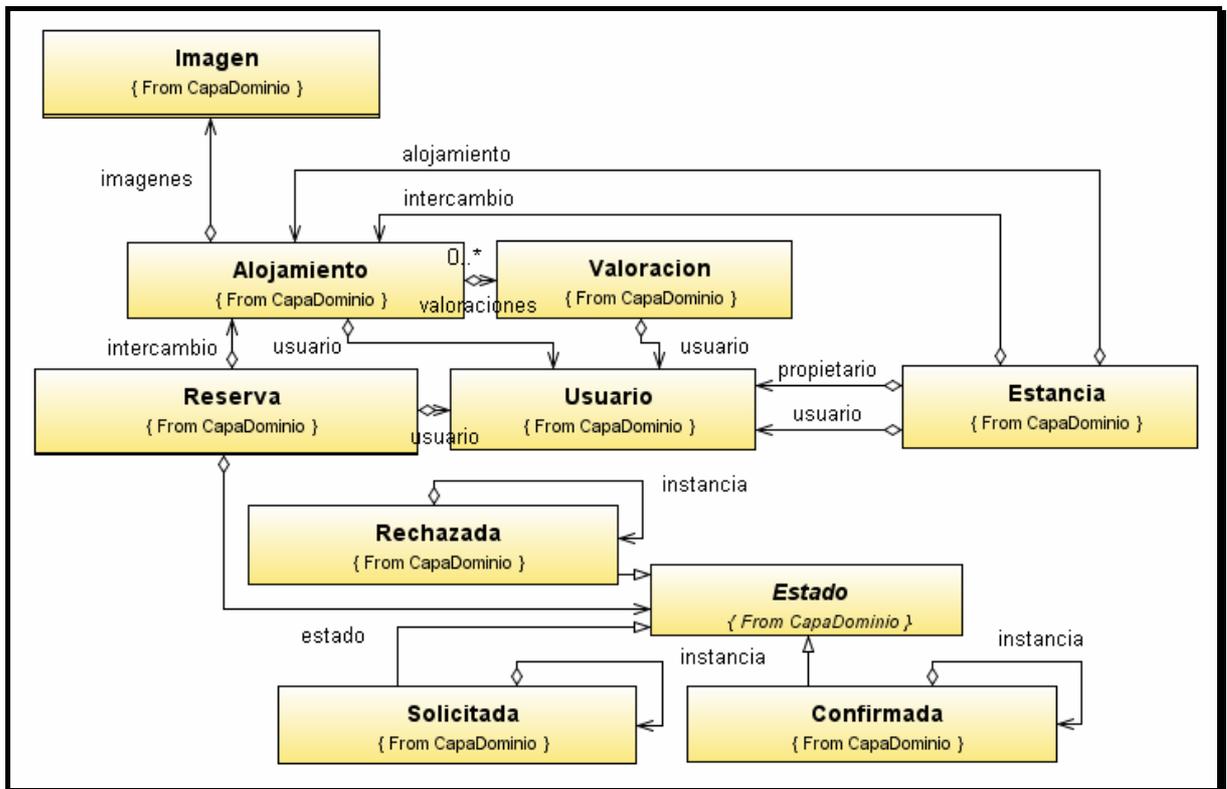


Fig. 20 – Diagrama de Clases. Dominio.

Para ver en detalle el Diagrama de Clases del Dominio se recomienda consultar el ANEXO 3.

## CLASES

### Alojamiento

Esta es la clase principal de la aplicación y representa un Alojamiento.



- **idAlojamiento:** es el identificador del alojamiento
- **titulo:** es un nombre descriptivo del alojamiento.
- **descripcion:** es una descripción sobre el alojamiento, contiene información sobre las características de éste.
- **tipo:** indica el tipo de vivienda (adosado, apartamento, ático, bungalow, casa, chalet, dúplex, loft, piso).
- **longitud:** indica la ubicación del alojamiento en el eje de las X dentro de un mapa.
- **latitud:** indica la ubicación del alojamiento en el eje de las Y dentro de un mapa.
- **habitaciones:** indica el número de habitaciones del alojamiento.
- **compartido:** indica si la estancia en el alojamiento se compartirá o no con el propietario (Sí / No).

## Usuario

Representa un usuario registrado en el Sistema, es la persona que posee los alojamientos.



- **usuario:** es el nombre de usuario, se utiliza para identificarse en el Sistema. Es único para cada usuario.
- **password:** es la contraseña utilizada para identificarse en el Sistema.
- **nombre:** es el nombre del usuario.
- **apellidos:** es el primer y segundo apellido del usuario.
- **paisIntercambio:** es el país en el que el usuario está interesado alojarse por medio de un intercambio.
- **valoracion:** es la media de todas las valoraciones que se han efectuado sobre el usuario por parte de otros usuarios.
- **votos:** número de votos que ha recibido el usuario

## Reserva

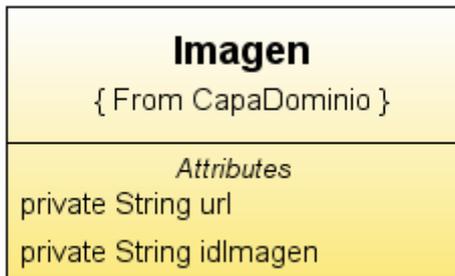
Representa una estancia pendiente de realizar en un alojamiento. Se utiliza para reservar la estancia en un alojamiento en unas fechas determinadas.

<b>Reserva</b>
<i>Attributes</i>
private String idReserva
private Fecha fechaEntrada
private Fecha fechaSalida
private Usuario usuario
private Estado estado
private Alojamiento intercambio

- **idReserva:** es el identificador de la Reserva.
- **fechaEntrada:** es la fecha de entrada de la estancia en el alojamiento.
- **fechaSalida:** es la fecha de salida de la estancia en el alojamiento.
- **usuario:** es el usuario que solicita la reserva / estancia.
- **estado:** es el estado en el que se encuentra la reserva (Solicitada, Confirmada o Rechazada).
- **intercambio:** es el alojamiento propuesto para un intercambio.

## Imagen

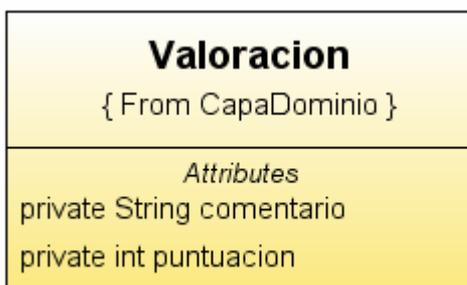
Es la foto de un alojamiento.



- **idImagen:** es el identificador de la imagen / foto.
- **url:** es la dirección de Internet donde está ubicada la foto.

## Valoracion

Representa la valoración u opinión de un usuario (inquilino) sobre un alojamiento en el que ha estado alojado.



- **comentario:** es un comentario escrito que describe la opinión de un usuario sobre el alojamiento.
- **puntuacion:** es la puntuación (un número del 1 al 5) que le da un usuario al alojamiento.

## Estancia

Representa un periodo de tiempo comprendido entre dos fechas en el que un usuario ha estado alojado en un alojamiento.

Es una reserva que ya ha sido cumplida y que contiene toda la información incluso después de que el alojamiento relacionado haya sido eliminado.

Las reservas están vinculadas a un alojamiento, si éste es eliminado por el usuario, desaparecen también las reservas. El objetivo es que el usuario pueda consultar sus reservas que ya han vencido, es decir, sus estancias, aunque el alojamiento ya no se pueda consultar.

Por eso se ha diseñado la clase Estancia, para que el usuario pueda tener un historial de sus estancias.

<b>Estancia</b>
<i>Attributes</i>
private String idEstancia
private Alojamiento alojamiento
private Usuario propietario
private Usuario usuario
private Alojamiento intercambio
private Fecha entrada
private Fecha salida

- **idEstancia:** identificador de la estancia
- **alojamiento:** alojamiento sobre el que se ha realizado la estancia
- **propietario:** propietario del alojamiento sobre el que se ha realizado la estancia.
- **usuario:** usuario que ha realizado la estancia (inquilino).
- **intercambio:** alojamiento con el que se ha realizado el intercambio. Es propiedad del usuario (inquilino).
- **entrada:** fecha de entrada de la estancia.
- **salida:** fecha de salida de la estancia.

## Fecha

Representa una fecha.

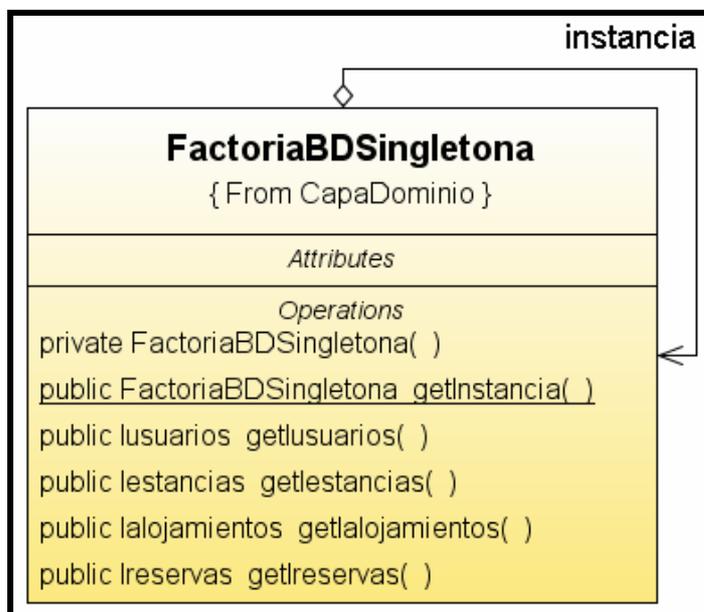
Esta clase ha sido creada para trabajar con fechas de forma personalizada, para adecuar la creación, el mantenimiento y la visualización de fechas a las necesidades y características de la aplicación.



- **date:** instancia de la clase *java.util.Date*.

## FactoriaBDSingletona

Es la clase encargada de crear objetos de las clases de Persistencia.



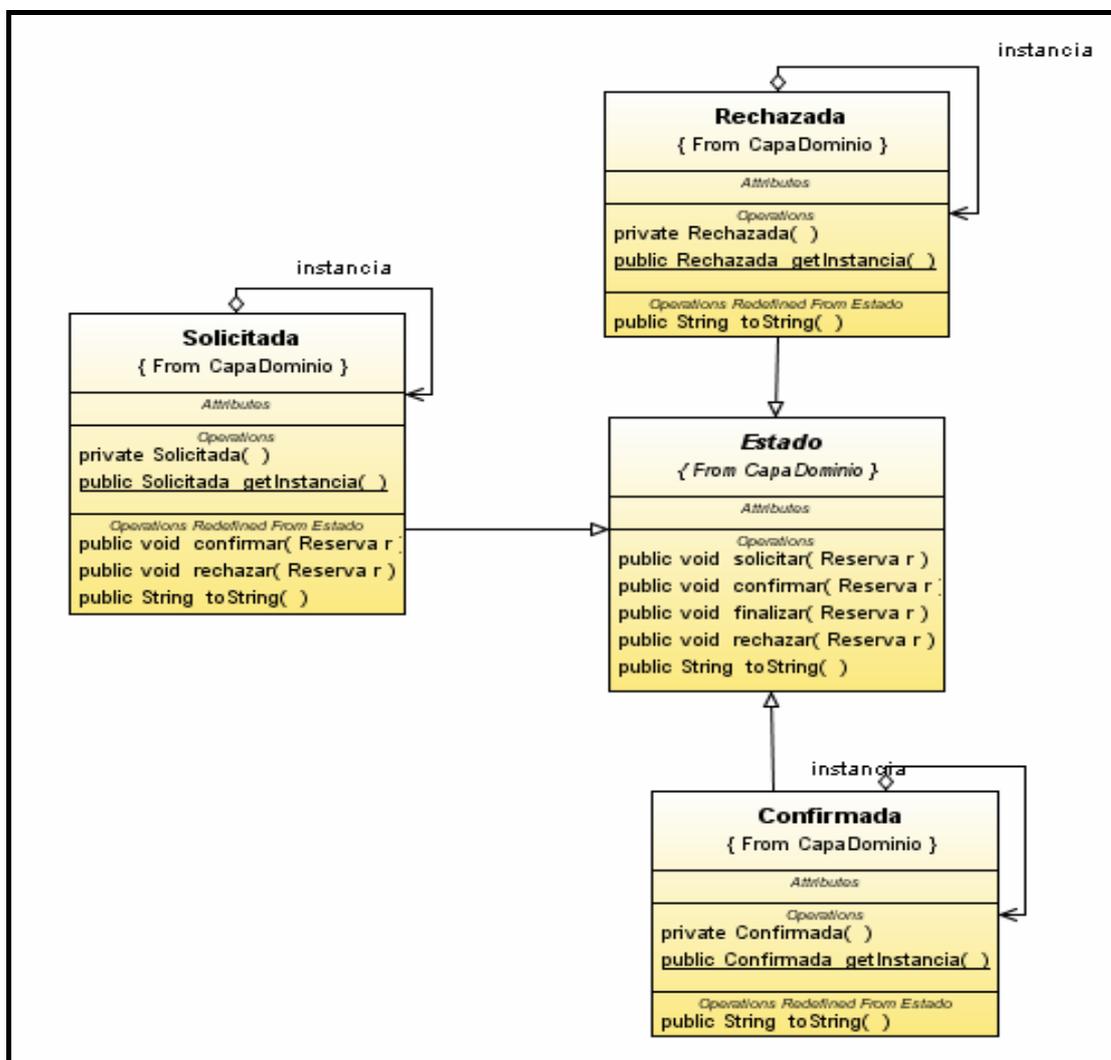
**Fig 20** - Clase FactoriaBDSingletona para crear instancias de Persistencia. Patrón Factoría.

**Estado (Solicitada, Confirmada, Rechazada). Patrón Estado.**

Es la implementación del patrón Estado requerido para la Reserva.

Un objeto de la clase *Reserva* tiene un atributo *estado* que es una instancia de la clase *Estado*. Esta instancia puede ser de la clase *Solicitada*, *Confirmada* o *Rechazada*, ya que éstas heredan de *Estado*.

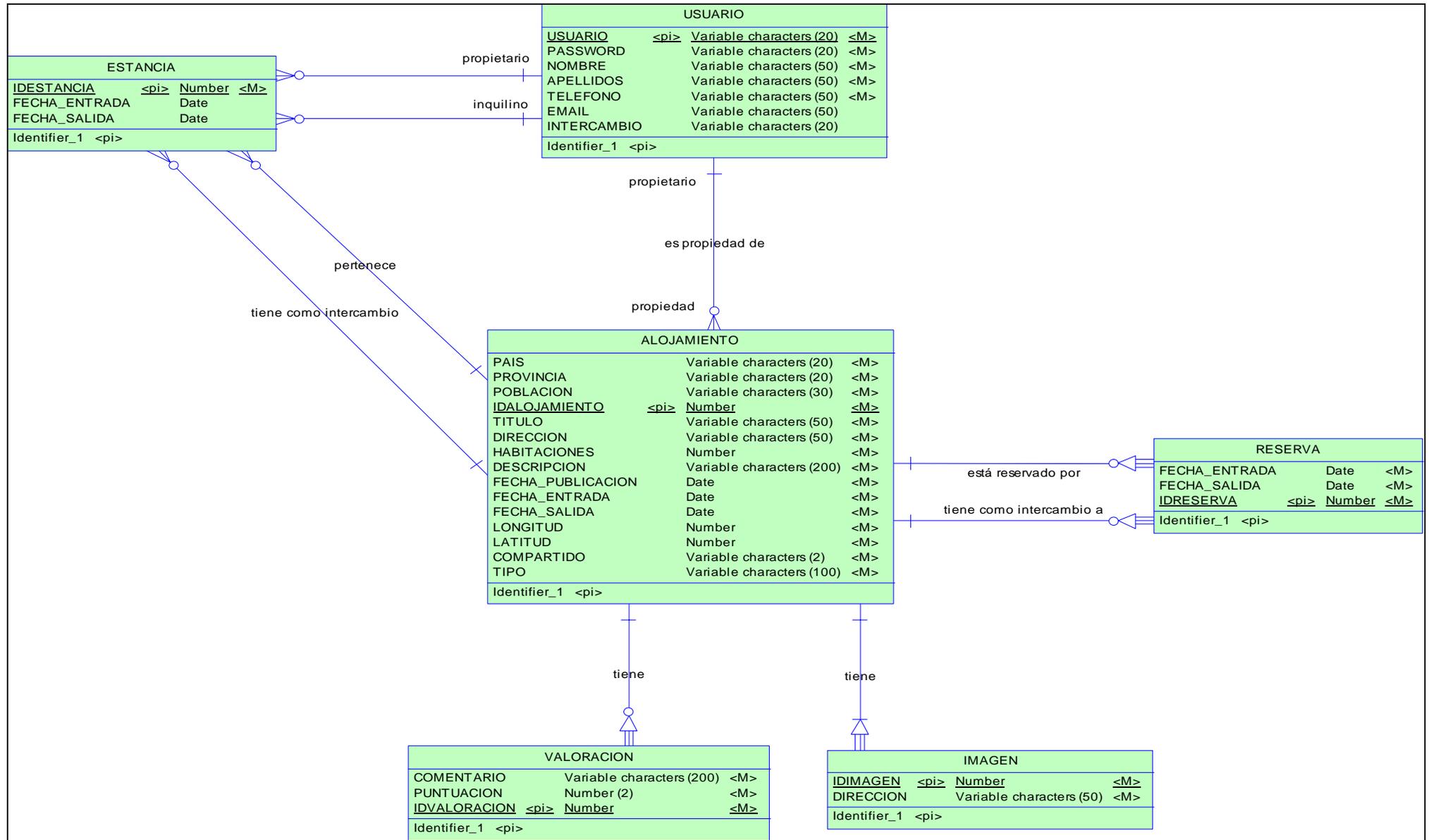
- **Solicitada:** significa que está pendiente de confirmación por parte del propietario.
- **Confirmada:** significa que ha sido confirmada por el propietario pero que la estancia en el alojamiento aún no se ha efectuado.
- **Rechazada:** significa que ha sido rechazada por el propietario.



**Fig. 21** – Implementación del patrón Estado para controlar los estados de un Reserva.

### 7.3 Persistencia

## Modelo Conceptual



## 8. Desarrollo e Implementación

La explicación en detalle de la implementación se puede encontrar en el propio código de la aplicación a modo de comentarios. En este apartado se describen sólo algunas decisiones de desarrollo e implementación que puedan suponer algún problema a la hora de entenderlas o aquellas que son relevantes.

Lenguajes utilizados para el desarrollo de la aplicación:

- UML
- SQL, SQL Oracle
- Java
- HTML, JSP y JSTL
- JavaScript

## 8.1 Presentación

### Desarrollo final de la interfaz gráfica

The screenshot shows the 'punto de encuentro' website interface. At the top, there is a logo and navigation buttons for LOGOUT, REGISTRAR, and CONTACTO. Below the logo, the text 'Red social de intercambio de alojamientos' is displayed. The main content area is divided into two sections. On the left, there is a search filter section titled 'Buscador' with fields for País, Provincia, Fecha entrada, Fecha salida, Habitaciones, Compartido, and Ofrezco intercambio en. A 'BUSCAR' button is located at the bottom of this section. On the right, there is a table titled 'Alojamientos añadidos los últimos 10 días' with columns for Nombre, País, Provincia, and Fecha publicación. The table contains one entry: 'Apartamento en Mataró' in España, Barcelona, published on 15/06/2009. Below the search filter, there is a user navigation section titled 'Bienvenido Toni' with links for 'Mis Alojamientos', 'Editar Perfil', 'Mis Reservas', and 'Mis Estancias'.

home.jsp

Esta es la pantalla principal de la web, donde se muestran los alojamientos publicados los últimos 10 días, indicando la fecha de publicación.

En este ejemplo el usuario está identificado en la sesión, por eso se muestra a la izquierda el panel con las opciones de usuarios registrados.

**punto de encuentro**  
Red social de intercambio de alojamientos

LOGOUT REGISTRAR CONTACTO

**Buscador**

País: [Selecciona País] ▾  
 Provincia: ▾  
 Fecha entrada:   
 Fecha salida:   
 Habitaciones: 1 (mínimo)  
 Compartido: Indiferente ▾  
 Ofrezco intercambio en: [Selecciona País] ▾  
 BUSCAR

**MIS ALOJAMIENTOS**  
 Gestiona tus Alojamientos

Nombre	Fecha publicación	
<a href="#">Apartamento de 1 dormitorio con terraza</a>	05/05/2009	Editar  Borrar

**Reservas**

Intercambio	Estado	Solicitante	Fecha entrada	Fecha salida
	<input checked="" type="checkbox"/> Confirmada	<a href="#">Toni</a>	05/07/2009	10/07/2009
	<input checked="" type="checkbox"/> Confirmada	<a href="#">Pedro</a>	11/07/2009	19/07/2009

**Bienvenido Jose**

- [Mis Alojamientos](#)
- [Editar Perfil](#)
- [Mis Reservas](#)
- [Mis Estancias](#)

Nuevo Alojamiento

mis\_alojamientos.jsp

Esta pantalla sirve para gestionar todos los alojamientos de un usuario. Permite crearlos, borrarlos, editarlos, ver sus reservas y gestionar éstas.

En el ejemplo hay dos reservas ya confirmadas para el alojamiento. Si hubiera alguna solicitada aparecerían a la derecha de cada reserva unos enlaces para confirmarla o rechazarla.

**punto de encuentro** LOGOUT REGISTRAR CONTACTO

Red social de intercambio de alojamientos

**Buscador**

País: [Selecciona País] | Provincia: [ ] | Fecha entrada: [ ] | Fecha salida: [ ] | Habitaciones: 1 (mínimo) | Compartido: Indiferente | Ofrezco intercambio en: [Selecciona País] | **BUSCAR**

**Bienvenido Toni**

- Mis Alojamientos
- Editar Perfil
- Mis Reservas
- Mis Estancias

**Editar Alojamiento**

Título: Apartamento de 1 dormitorio con terraza

País: España | Provincia: Pontevedra | Fecha entrada: 01/07/2009 | Fecha salida: 20/07/2009

[Añadir fechas](#)

Descripción: Ascensor, Trastero, Calefacción, Exterior, Garaje. Obra Nueva.

Nº Habitaciones: 1 | Tipo de vivienda: Adosado | Compartido: No

Dirección: sanxenxo pontevedra | **Geolocalizar**

Mapa de Google: Sanxenxo, Pontevedra, Ria de Pontevedra, Meaño, Poio, Marin, Vilaboa, E-1

POWERED BY Google | Datos de mapa ©2009 Tele Atlas

**Fotos**

<http://imagenes.globaliza.com/rcs/imagenes/2008/10/05/11/p15>  
 Actualizar  Eliminar

[http://www.habitat24.com/ftp/1728/1728\\_008639\\_Foto\\_1\\_9332686](http://www.habitat24.com/ftp/1728/1728_008639_Foto_1_9332686)  
 Actualizar  Eliminar

**AGREGAR**

**PUBLICAR**

editar\_alojamiento.jsp

Esta es la pantalla de edición de un alojamiento. Muestra un formulario similar al del formulario de alta pero en un formato editable.

Las fotos tienen un formulario a parte que permite gestionarlas individualmente: actualizarlas, eliminarlas o añadir nuevas. Todos los cambios se verán reflejados al momento.



LOGOUT
REGISTRAR
CONTACTO

Red social de intercambio de alojamientos

**Buscador**

País

Provincia

Fecha entrada

Fecha salida

Habitaciones  (mínimo)

Compartido

Ofrezco intercambio en:

BUSCAR

**Piso en Zamora**

Publicado por [Sonia](#)

**País:** España

**Provincia:** Zamora

**Tipo:** Adosado

**Descripción:** Piso en buen estado se vende con todos los muebles y cocina equipada 2 Dormitorios, 1 Baño(s), 50 Metros Útiles.

**Nº Habitaciones** 2

Ver Ubicación Geográfica

**Fotos**



**Comentarios**

[Toni](#) Es muy acogedor. 4/5

**Bienvenido Toni**

- [Mis Alojamientos](#)
- [Editar Perfil](#)
- [Mis Reservas](#)
- [Mis Estancias](#)

Comentario

Puntuación

Valorar este alojamiento

<b>Fecha disponibilidad:</b>	01/07/2009 hasta 20/07/2009
<b>Reservas</b>	
<a href="#">Juan</a>	01/07/2009 hasta 10/07/2009
<p><b>Solicitar reserva</b></p> <p><i>Selecciona las fechas en las que deseas solicitar tu estancia en este alojamiento</i></p> <p>Fecha entrada <input type="text"/></p> <p>Fecha salida <input type="text"/></p> <p style="text-align: center;"><span style="background-color: orange; color: white; padding: 2px 5px;">SOLICITAR</span></p>	
<p style="color: blue; text-align: center;"><a href="#">Proponer intercambio</a></p> <p style="text-align: center;"><i>El propietario de este alojamiento está dispuesto a un intercambio en Reino Unido</i></p>	

ver\_alojamiento.jsp

Esta es una de las principales pantallas de la web, pues muestra toda la información al detalle de un alojamiento, al mismo tiempo que ofrece las funcionalidades para realizar las reservas o intercambios.

Tenemos un apartado donde se muestra la información textual del alojamiento. A continuación está disponible un botón para ver la ubicación del alojamiento en un mapa (el mapa aparecerá a la izquierda, debajo del panel de usuario).

Seguidamente vemos la lista de fotos del alojamiento (ampliables con un click) y a continuación el formulario de valoraciones, donde se mostrarán todas las valoraciones de ese alojamiento por parte de usuarios que han estado alojados.

Se mostrará el nombre del usuario, un comentario y la puntuación sobre 5.

Por último tenemos el apartado para efectuar reservas. Se muestra la fecha de disponibilidad del alojamiento y las fechas ocupadas por reservas ya confirmadas de otros usuarios. En función de eso, el usuario que desee solicitar una reserva deberá indicar las fechas deseadas.

A la derecha tenemos la opción para proponer intercambio en caso de que el propietario del alojamiento esté interesado en intercambio. En este ejemplo, la usuario Sonia está interesada en intercambios en Reino Unido, por lo tanto, se da la opción de intercambio.

Buscador		NUEVO ALOJAMIENTO	
País	[Selecciona País] ▼	Título	<input type="text"/>
Provincia	▼	País	[Selecciona País] ▼
Fecha entrada	<input type="text"/>	Provincia	- ▼
Fecha salida	<input type="text"/>	Fecha entrada	<input type="text"/>
Habitaciones	1 (mínimo)	Fecha salida	<input type="text"/>
Compartido	Indiferente ▼	<a href="#">Añadir</a>	
Ofrezco intercambio en:	[Selecciona País] ▼	Descripción	<input type="text"/>
 <input type="button" value="BUSCAR"/>		Nº Habitaciones	1 ▼
		Tipo de vivienda	Adosado ▼
		Compartido	No ▼
		Foto principal	<input type="text"/>
		Dirección	<input type="text"/>
		<input type="button" value="Geolocalizar"/>	
		<i>Introduce la dirección y comprueba que la ubicación es correcta. Puedes reubicar el punto manualmente para obtener más precisión.</i>	

Bienvenido <b>Toni</b>	
	<a href="#">Mis Alojamientos</a>
	<a href="#">Editar Perfil</a>
	<a href="#">Mis Reservas</a>
	<a href="#">Mis Estancias</a>

alta\_alojamiento.jsp

Esta pantalla está destinada a dar de alta un alojamiento en la web. Consta de un formulario estándar para la inserción de los datos del alojamiento y un sistema para insertar la ubicación geográfica del alojamiento mediante Google Maps.

Consiste en introducir la dirección y mediante el botón “Geolocalizar” se mostrará en un mapa la ubicación en función de la dirección, el país y la provincia introducida.

La marca que se mostrará en el mapa se podrá reubicar manualmente para más precisión.

**punto de encuentro** LOGOUT REGISTRAR CONTACTO

Red social de intercambio de alojamientos

**Buscador**

País: [Selecciona País] ▾  
 Provincia: ▾  
 Fecha entrada:    
 Fecha salida:    
 Habitaciones: 1 ▾ (mínimo)  
 Compartido: Indiferente ▾  
 Ofrezco intercambio en: [Selecciona País] ▾

BUSCAR

**EDITAR Perfil**

Uusario\* Toni  
 Contraseña\* toni  
 Nombre\*   
 Apellidos\*   
 Telefono\*   
 Email

Estoy interesado en Intercambios de

Actualizar

**Bienvenido Toni**

[Mis Alojamientos](#)

\* Campos obligatorios

editarUsuario.jsp

La pantalla para editar el perfil de un usuario mediante un formulario similar al formulario de registro.

The screenshot shows the 'punto de encuentro' website interface. At the top, there is a logo and navigation buttons for 'LOGIN', 'REGISTRAR', and 'CONTACTO'. Below the logo, the text 'Red social de intercambio de alojamientos' is displayed. On the left side, there is a search bar titled 'Buscador' with various filters: País, Provincia, Fecha entrada, Fecha salida, Habitaciones (set to 1), Compartido (set to Indiferente), and Ofrezco intercambio en. A 'BUSCAR' button is at the bottom of the search bar. The main content area displays the profile for 'PERFIL DE Toni'. It includes a rating of 3.3333333 (3 votes) with a 'Valorar' button. The name 'Nombre: Toni Ruiz' is shown. A note states: 'La información de CONTACTO de este usuario se suministra a los usuarios que han solicitado una reserva en alguno de sus alojamientos.' Below this, there are two tables. The first table, 'Estancias', has columns for Alojamiento, Propietario, Intercambio, Entrada, and Salida, with one entry: 'Piso en Zamora' by 'Sonia' with dates 05/08/2008 and 14/08/2008. The second table, 'Alojamientos', has columns for Nombre, País, Provincia, and Fecha publicación, with two entries: 'Apartamento de 1 dormitorio con terraza' in España, Pontevedra (05/05/2009) and 'Apartamento en Mataró' in España, Barcelona (15/06/2009). At the bottom, there are links for 'Ver sus Alojamientos' and 'Ver sus Estancias'.

verUsuario.jsp

Esta es la pantalla para mostrar la información de un usuario. Como se puede ver, la información de contacto sólo se suministra a usuarios que han solicitado reservas en alguno de sus alojamientos.

Se habilitan dos enlaces para ver los alojamientos y las estancias de ese usuario a fin de conocer un poco más el recorrido de ese usuario en la comunidad.

También permite valorar al usuario (sólo para propietarios de alojamientos donde ha estado alojado el usuario a valorar). El objetivo es saber los usuarios que son buenos inquilinos y los que no.



mis\_reservas.jsp

Esta pantalla muestra las reservas que ha efectuado un usuario, para saber el estado en el que se encuentran y cuando pueden alojarse.



ver\_estancias.jsp

Es una pantalla similar a la de las reservas, pero mostrando las estancias que el usuario ya ha realizado en los diferentes alojamientos.

**punto de encuentro**  
Red social de intercambio de alojamientos

LOGOUT REGISTRAR CONTACTO

**Buscador**

Se han encontrado 7 alojamientos en **España**

Mapa Satélite Híbrido

POWERED BY Google

BIENVENIDO **Toni**

- Mis Alojamientos
- Editar Perfil
- Mis Reservas
- Mis Estancias

	Provincia	Fecha publicación	Tipo	nºHab.	
<a href="#">Apartamento de 1 dormitorio con terraza</a>	Pontevedra	05/05/2009	Adosado	1	
<a href="#">Apartamento de 1 dormitorio con terraza</a>	Pontevedra	11/05/2009	Adosado	1	
<a href="#">Apartamento de 1 dormitorio con terraza</a>	Pontevedra	05/05/2009	Adosado	1	

resultados.jsp

Esta es otra de las pantallas principales de la web. Muestra los resultados del buscador de alojamientos mediante un mapa de Google Maps, mostrando mediante marcas la ubicación de los alojamientos resultantes y una lista con los detalles.

Al lado de cada alojamiento tenemos un icono en forma de casa que sirve para iluminar la marca del alojamiento seleccionado y verlo ampliado en el propio mapa.

Al hacer click en cada marca se visualizará un pequeña foto del alojamiento y un enlace para ver más detalles.

### Uso del API Google Maps en la interfaz

Se ha utilizado el API de Google Maps (*Ref, [16]*) para poder visualizar un mapa de Google en el que marcar la ubicación geográfica de los alojamientos.

Las páginas JSP en las que se ha utilizado han sido las siguientes:

- **resultados.jsp:** un mapa para mostrar la ubicación de los alojamientos que coinciden con las condiciones de búsqueda.
- **alta\_alojamiento.jsp:** un mapa para mostrar la ubicación geográfica del alojamiento según la dirección. Posibilidad de reposicionar la marca manualmente.
- **ver\_alojamiento.jsp:** un mapa para mostrar una marca con la ubicación geográfica del alojamiento.

Esta API está implementada con Javascript y es el lenguaje de programación que hay que usar para trabajar con ella.

Las clases y métodos utilizadas para esta aplicación han sido las relacionadas con la creación y visualización de mapas, añadido de controles, posicionamiento y reposicionamiento de marcas, personalización de marcas, inserción y extracción de coordenadas y obtención de coordenadas a partir de direcciones. (*Ref: [2],[3],[4]*)

Para la visualización correcta de la ubicación geográfica de los alojamientos en el mapa, hemos almacenado las coordenadas de cada alojamiento en la base de datos, de forma que podamos cargarlas y visualizarlas en el mapa por medio de marcas.

El almacenamiento de estas coordenadas se realiza al dar de alta un alojamiento.

País	<input type="text" value="España"/>
Provincia	<input type="text" value="Barcelona"/>

Dirección	<input type="text" value="avenida diagonal 605"/>
	<input type="button" value="Geolocalizar"/>
	<i>Introduce la dirección y comprueba que la ubicación es correcta. Puedes reubicar el punto manualmente para obtener más precisión.</i>

A partir del país, la provincia y la dirección que proporciona el usuario obtenemos las coordenadas y visualizamos la ubicación por medio del botón *Geolocalizar*, que llama a una función almacenada en el fichero Javascript *geolocalizar.js* que se encarga de darnos las coordenadas a partir de esos datos (Ref. [6],[7],[8])



**Fig. 22** - Ejemplo de geolocalización por dirección.

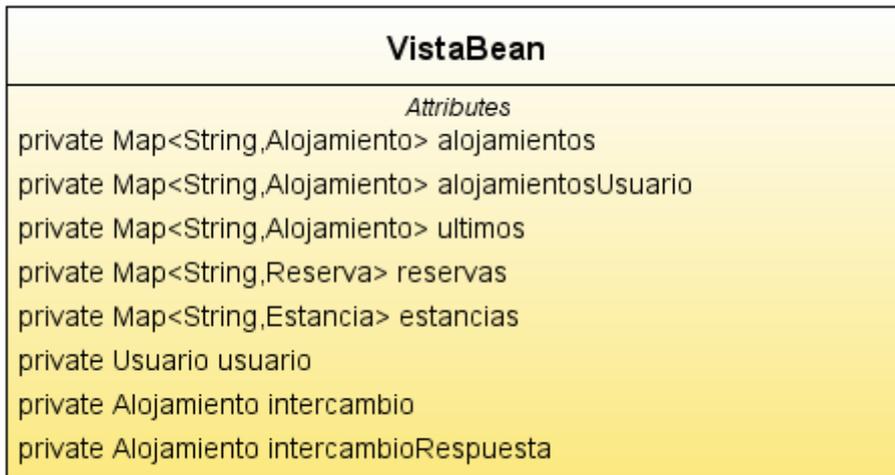
La marca indica la ubicación del alojamiento. El usuario puede interactuar con el mapa y reubicar la marca manualmente para obtener más precisión.

Las coordenadas se guardan en el formulario como inputs *hidden* y se actualizan automáticamente cada vez que el usuario reubicación la marca.

Al enviar el formulario de alta, se envían las coordenadas de la marca.

**Visualización de los resultados en la web: VISTA (clase VistaBean.java)**

Esta clase se almacena en la sesión del usuario y contiene todos los resultados que devuelve la aplicación. Las páginas JSP acceden a este JavaBean para visualizar los resultados en la web.



Clase VistaBean.java

- **alojamientos:** almacena la lista Alojamiento cargados en la última consulta.
- **reservas:** almacena la lista de Reservas de un usuario mediante el caso de uso “Ver mis reservas”
- **alojamiento:** almacena el último Alojamiento cargado mediante el caso de uso “Ver Alojamiento”
- **usuario:** almacena el último Usuario cargado mediante el caso de uso “Ver Usuario”
- **estancias:** almacena la lista de Estancias cargadas en la última consulta.
- **intercambio:** almacena el Alojamiento que un usuario ha propuesto para intercambio.
- **intercambioRespuesta:** almacena el Alojamiento que un usuario propone para intercambio cuando acepta una reserva con intercambio.
- **alojamientosUsuario:** almacena la lista de Alojamiento de un Usuario concreto.
- **ultimos:** almacena la lista de los últimos alojamientos publicados.

Aprovechamos también este JavaBean para evitar accesos innecesarios a la base de datos. Por ejemplo si un usuario quiere consultar información sobre un alojamiento, el Sistema intentará cargarlo de la lista de alojamientos que tiene el usuario cargados en la sesión (en VistaBean) usando la propiedad *alojamientos*. En caso de no encontrar el alojamiento, el Sistema irá a buscarlo a la base de datos.

Hay que tener en cuenta entonces, que el Sistema deberá realizar una búsqueda en memoria, pero siempre será más rápido que realizar un acceso a persistencia.

Esta filosofía de funcionamiento se verá aplicada durante todo el desarrollo de la aplicación para garantizar un mínimo de rendimiento.

## Visualización de Excepciones

Para poder recoger las excepciones y errores que puedan suceder durante el uso de la aplicación y mostrárselo al usuario se ha utilizado un *JavaBean* que estará alojado en la sesión y al cual accederán las páginas JSP.

La clase se llama **ErrorsBean** y tiene una propiedad llamada **error**, donde el *Action* correspondiente almacena el mensaje de error pertinente al capturar una excepción.

```
public class ErrorsBean {

    private String error;

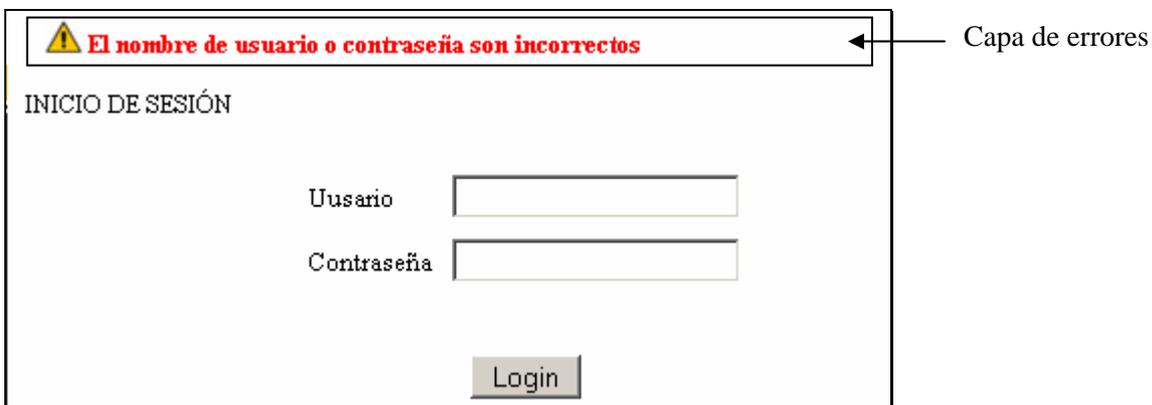
    public String getError() {
        return error;
    }

    public void setError(String error) {
        this.error = error;
    }
}
```

Fig. 23 – Clase ErrorsBean.

Entre la capa *Cabecera* y la capa *Contenido* de la interfaz gráfica se ha insertado otra capa exclusivamente para mostrar los errores.

Cuando la aplicación genera un error, lo añade a **ErrorsBean** y devuelve la página de origen, que visualizará el error añadido.



The image shows a web form for user login. At the top, there is a red error message box with a yellow warning triangle icon and the text "El nombre de usuario o contraseña son incorrectos". Below this, the form is titled "INICIO DE SESIÓN" and contains two input fields: "Usuario" and "Contraseña". A "Login" button is positioned at the bottom center of the form. An arrow on the right side of the image points to the error message box, with the label "Capa de errores".

Fig. 24 – Identificación de usuario errónea.

El código JSP/JSTL que se añade en la capa de errores de cada página es el siguiente:

```
<div id="errores">
  <c:if test="${!empty sessionScope.errorsBean.error}">
    
    <span class="error"> ${errorsBean.error}</span>
    <c:set target="${errorsBean}" property="error" value=""/>
  </c:if>
</div>
```

La página JSP accede a **ErrorsBean** y si hay un error añadido lo muestra y posteriormente lo elimina.

## Otros ficheros

- **capas.css:** Define la estructura visual de la web en CSS.
- **cambiaProvinciaAlta.js:** función Javascript que cambia las provincias en función del país seleccionado. Para el formulario de alta. **cambiaProvinciaBusca.js** para el formulario de Búsqueda. **cambiaProvinciaEditar.js** para el formulario de edición de alojamiento.
- **geolocalizar.js:** función Javascript que devuelve las coordenadas de un punto en Google Maps en función de país, provincia y dirección.
- **validaBuscador.js:** función Javascript que valida los datos del formulario de Búsqueda.
- **países:** fichero Javascript que almacena todos los países y provincias.

## 8.2 Controlador y Dominio

A continuación se explican algunas decisiones de implementación relacionadas con la Capa Controlador y la Capa Dominio, como el uso y aprovechamiento de los *Action* y *ActionForm* y su relación con los casos de uso descritos anteriormente.

### Correspondencia entre Casos de Uso y Actions

La siguiente tabla muestra los *Action* y *ActionForm* que se han utilizado para implementar cada Caso de Uso.

Caso de Uso	ActionForm / Action
Buscar alojamientos	buscarAlojamientos
Dar de alta alojamiento	altaAlojamiento
Ver alojamiento	verAlojamiento
Ver últimos alojamientos	verUltimos
Borrar alojamiento	borrarAlojamiento
Editar alojamiento	verAlojamiento (para mostrar la información del alojamiento en formato editable) editarAlojamiento (para efectuar los cambios realizados)
Ver Alojamientos por Usuario	verAlojamientosUsuario
Actualizar foto	actualizarImagen
Añadir foto	addImagen
Borrar foto	borrarImagen
Añadir valoración	addValoracion
Solicitar reserva	solicitarReserva proponerIntercambio (sólo intercambio)

Confirmar reserva	confirmarReserva aceptarIntercambio(sólo para intercambio) solicitarReserva (sólo para intercambio)
Rechazar reserva	rechazarReserva
Eliminar reserva	quitarReserva
Ver usuario	verUsuario
Logout	logout
Login	login
Ver Estancias por Usuario	verEstanciasUsuario
Ver Reservas por Usuario	verReservasUsuario
Editar usuario	verUsuario (para mostrar la información del usuario en formato editable) editarUsuario (para efectuar los cambios realizados)
Valorar usuario	valorarUsuario
Registrar usuario	registrarUsuario

**Fig. 25** – Correspondencia entre Casos de Uso y Actions.

### **Diagrama de Secuencia del caso de uso *Solicitar Reserva***

*Ver ANEXO 6 (Proyecto NetBeans)*

## 8.3 Persistencia

### Tablas de la BBDD Oracle

#### ALOJAMIENTO

Column Name	Data Type	Nullable
IDALOJAMIENTO	NUMBER	No
PAIS	VARCHAR2(50 BYTE)	No
PROVINCIA	VARCHAR2(50 BYTE)	No
USUARIO	VARCHAR2(50 BYTE)	No
TITULO	VARCHAR2(200 BYTE)	No
DIRECCION	VARCHAR2(200 BYTE)	No
HABITACIONES	NUMBER	No
DESCRIPCION	VARCHAR2(2000 BYTE)	No
FECHA_PUBLICACION	DATE	No
FECHA_ENTRADA	DATE	No
FECHA_SALIDA	DATE	No
LONGITUD	FLOAT	No
LATITUD	FLOAT	No
COMPARTIDO	VARCHAR2(2 BYTE)	No
TIPO	VARCHAR2(100 BYTE)	No

#### Restricciones

- COMPARTIDO: Sólo puede contener la cadena “SÍ” o la cadena “No”.
- TIPO: Sólo puede contener alguna de las siguientes cadenas: “Adosado”, “Apartamento”, “Ático”, “Casa”, “Chalet”, “Dúplex”, “Loft”, “Piso”.

#### USUARIO

Column Name	Data Type	Nullable
USUARIO	VARCHAR2(50 BYTE)	No
PASSWORD	VARCHAR2(50 BYTE)	No
NOMBRE	VARCHAR2(50 BYTE)	No
APELLIDOS	VARCHAR2(50 BYTE)	No
TELEFONO	VARCHAR2(50 BYTE)	No
EMAIL	VARCHAR2(50 BYTE)	Yes
INTERCAMBIO	VARCHAR2(20 BYTE)	Yes
VALORACION	FLOAT	No

## RESERVA

Column Name	Data Type	Nullable
IDRESERVA	NUMBER	No
FECHA_ENTRADA	DATE	No
FECHA_SALIDA	DATE	No
USUARIO	VARCHAR2(50 BYTE)	No
IDALOJAMIENTO	NUMBER	No
INTERCAMBIO	NUMBER	Yes
ESTADO	VARCHAR2(4000 BYTE)	No

## Restricciones

- ESTADO: Sólo puede contener los siguientes valores:  
 “CapaDominio.Solicitada”,  
 “CapaDominio.Confirmada”,  
 “CapaDominio.Rechazada”.

## ESTANCIA

Column Name	Data Type	Nullable
IDESTANCIA	NUMBER	No
FECHA_ENTRADA	DATE	No
FECHA_SALIDA	DATE	No
PROPIETARIO	VARCHAR2(50 BYTE)	No
USUARIO	VARCHAR2(50 BYTE)	No
ALOJAMIENTO	NUMBER	No
INTERCAMBIO	NUMBER	Yes

## IMAGEN

Column Name	Data Type	Nullable
IDIMAGEN	NUMBER	No
IDALOJAMIENTO	NUMBER	No
URL	VARCHAR2(1000 BYTE)	No

## VALORACION

Column Name	Data Type	Nullable
IDVALORACION	NUMBER	No
COMENTARIO	VARCHAR2(200 BYTE)	No
PUNTUACION	NUMBER(2,0)	No
IDALOJAMIENTO	NUMBER	No
USUARIO	VARCHAR2(50 BYTE)	No

Además de estas tablas, en la base de datos podemos encontrar una función PL/SQL que se ha utilizado para resolver un caso de uso. Se explica a continuación.

## Funciones PL/SQL

Para el caso de uso “Ver Ultimos Alojamientos” se ha implementado una versión ligeramente modificada y personalizada de una función PL/SQL sacada de:

[http://www.techonthenet.com/oracle/questions/add\\_days.php](http://www.techonthenet.com/oracle/questions/add_days.php)

Esta función quita un determinado número de días a una fecha determinada. La utilizaremos para listar los alojamientos publicados en los últimos X días (*Ref. [15]*).

```

create or replace Function quitar_dias (start_date_in date, days_in number)
  return date
IS
  v_counter number;
  v_new_date date;
  v_day_number number;
BEGIN
  v_counter := 1;
  v_new_date := start_date_in;

  while v_counter <= days_in
  loop
    v_new_date := v_new_date - 1;
    v_day_number := to_char(v_new_date, 'd');

    if v_day_number >= 2 and v_day_number <= 6 then
      v_counter := v_counter + 1;
    end if;
  end loop;

  RETURN v_new_date;

EXCEPTION
WHEN OTHERS THEN
  raise_application_error(-20001,'An error was encountered -
  ' || SQLCODE || '-ERROR- ' || SQLERRM);
END;
```

A nivel de programación Java, dentro del proyecto NetBeans y en la capa Persistencia encontramos las clases descritas a continuación.

### Capa Persistencia

Para la implementación de la Capa de Persistencia se han utilizado las siguientes clases que se encargan de hacer consultas a la base de datos:

<b>ReservasBD</b>
<i>Attributes</i>
private Connection con
<i>Operations</i>
public ReservasBD( )
private getEstado( String estado )

Persistencia relacionada con las Reservas. Implementa la interfaz ***Ireservas***.

<b>AlojamientosBD</b>
<i>Attributes</i>
private Connection con
<i>Operations</i>
public AlojamientosBD( )
private ResultSet sqlDinamica( Map parametros )
private void addEstancia( r, a, String usuario )

Persistencia relacionada con los Alojamientos. Implementa la interfaz ***Ialojamientos***

<b>UsuariosBD</b>
<i>Attributes</i>
private PreparedStatement pstmt private Connection con private Statement st private ResultSet rs
<i>Operations</i>
public UsuariosBD( )

Persistencia relacionada con los Usuarios. Implementa la interfaz ***Iusuarios***.

<b>EstanciasBD</b>
<i>Attributes</i>
private Connection con
<i>Operations</i>
public EstanciasBD( )

Persistencia relacionada con las Estancias. Implementa la interfaz ***Iestancias***.

<b>Conexion</b>	instancia
<i>Attributes</i>	
private String DRIVER = "oracle.jdbc.driver.OracleDriver" private String DB = "jdbc:oracle:thin:@rama.eupmt.es:1521:logo10" private String USER = "student20" private String PASS = "student" private Connection con	
<i>Operations</i>	
private Conexion( ) public Conexion getInstance( ) protected Connection getConexion( ) public void cerrarConexion( )	

Es la clase que se encarga de entablar conexión con la base de datos Oracle. Singletona (Ref. [11]).

En la capa de Persistencia también se ha implementado el patrón Transacción para garantizar un estado consistente de la base de datos en todo momento.

### Implementación del Patrón Transacción

```
public void iniciarTransaccion() throws Exception {
    con.setAutoCommit(false);
}

public void cerrarTransaccion(boolean exito) throws Exception {
    if (exito) con.commit();
    else con.rollback();
    con.setAutoCommit(true);
}
```

Fig. 26 -Implementación del patrón Transacción

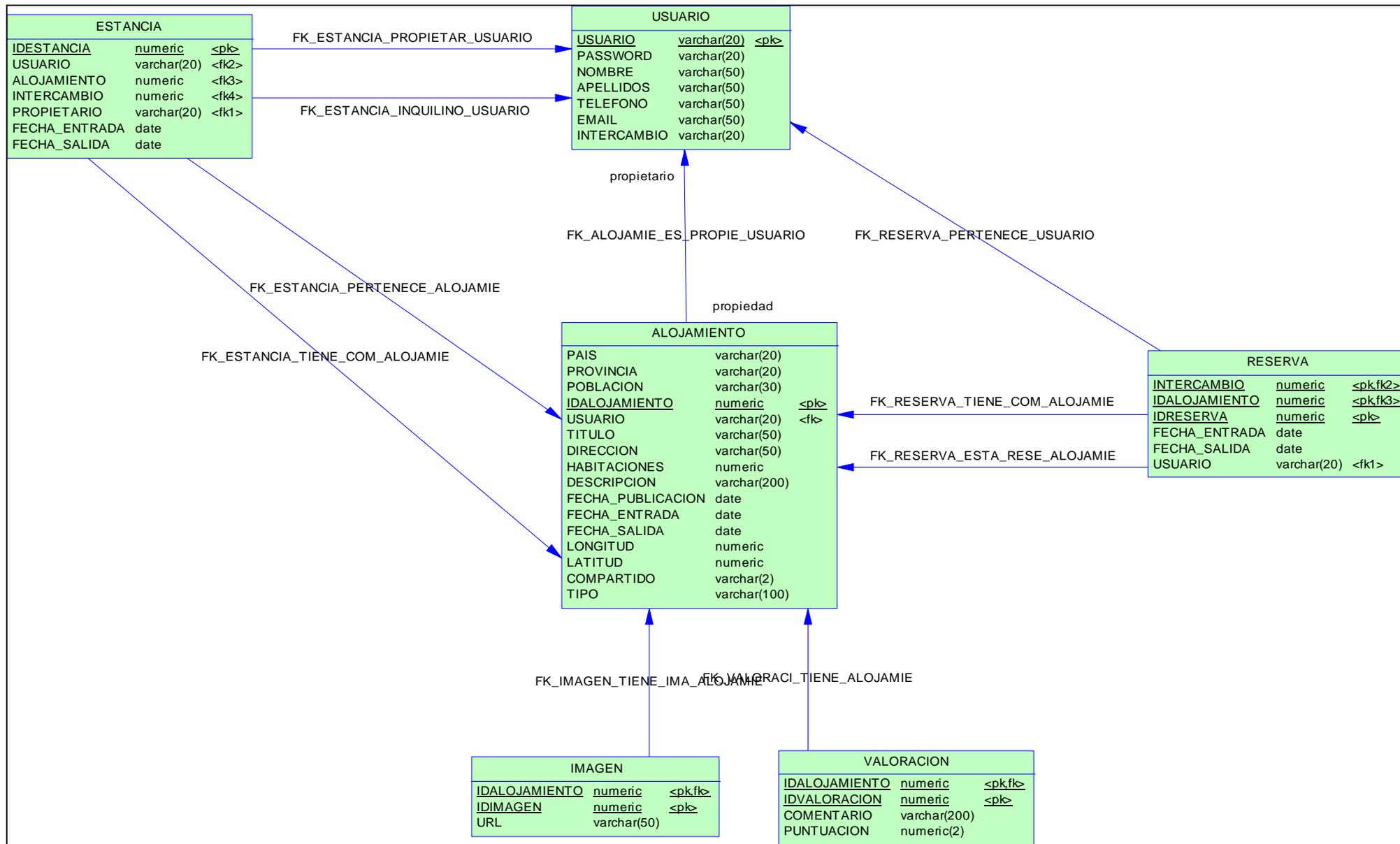
La implementación consiste en controlar el *autocommit* (por defecto está a *true*) de la base de datos para construir bloques de instrucciones.

Por defecto, cada transacción que hacemos en la base de datos se confirma individualmente, pero si queremos ejecutar varias transacciones en conjunto, que no tienen sentido individualmente, iniciamos y cerramos la transacción manualmente utilizando este patrón.

Por ejemplo, para dar de alta un Alojamiento en la base de datos el Sistema modifica 2 tablas: Alojamiento e Imagen. Por lo tanto se hacen dos accesos. Si uno de ellos falla, por ejemplo la inserción en la tabla Imagen, el Alojamiento se habría almacenado pero la Imagen no, y tendríamos la base de datos inconsistente.

Para evitar esto se utiliza el patrón. Cuando vamos a ejecutar un conjunto de pequeñas transacciones, ponemos el *autocommit* a *false*, para poder hacer *rollback* en caso de error. Y cuando terminamos, si no ha habido errores (comprobación mediante la variable *exito*), hacemos *commit* y cerramos la transacción

### Modelo Relacional / Físico



## 9. Pruebas

A continuación se describen pruebas de los casos de uso más complicados de la aplicación.

### Prueba de un intercambio de alojamientos entre dos usuarios

Marco es un usuario que tiene un alojamiento publicado en Argentina y está interesado en un intercambio en España.

Sergio es un usuario que tiene un alojamiento en España y está interesado en un intercambio en Argentina.

Buscador	Piso en Santa Rosa
País <input type="text" value="[Selecciona País]"/>	Publicado por <a href="#">Marco</a>
Provincia <input type="text"/>	<b>País:</b> Argentina
Fecha entrada <input type="text"/>	<b>Provincia:</b> La Pampa
Fecha salida <input type="text"/>	<b>Tipo:</b> Piso
Habitaciones <input type="text" value="1"/> (mínimo)	<b>Descripción:</b> Piso de 3 habitaciones, una de ellas tipo "suite", muy grandes (20, 14, 13 m <sup>2</sup> ) todas con armarios empotrados totalmente revestidos y con amplias cajoneras. Dos baños grandes (7 y 8 m <sup>2</sup> ), uno con bañera de hidromasaje y el otro con cabina de hidromasaje. Salón de 25 m <sup>2</sup> , cocina de 10 m <sup>2</sup> tipo office con amplia terraza. Aire acondicionado con bomba de calor en hab. principal. Dobles ventanas "climalit" y mosquiteras en toda la casa. Plaza de garaje amplia (14m <sup>2</sup> ). Sol todo el día en todas las habitaciones.
Compartido <input type="text" value="Indiferente"/>	<b>Nº Habitaciones</b> 3
Ofrezco intercambio en: <input type="text" value="[Selecciona País]"/>	<input type="button" value="Ver Ubicación Geográfica"/>
<input type="button" value="BUSCAR"/>	
<b>Bienvenido Sergio</b>	

Fig. 27 - Alojamiento de Marco, "Piso en Santa Rosa", en Argentina.

Sergio quiere reservar este alojamiento en Argentina que es propiedad de Marco.

<b>Fecha disponibilidad:</b>	20/12/2009 hasta 25/12/2009
<b>Solicitar reserva</b> <i>Selecciona las fechas en las que deseas solicitar tu estancia en este alojamiento</i> Fecha entrada <input type="text"/> Fecha salida <input type="text"/> <input type="button" value="SOLICITAR"/>	<a href="#">Proponer intercambio</a> <i>El propietario de este alojamiento está dispuesto a un intercambio en España</i>

Fig. 28 - Formulario de Reserva del alojamiento de Marco, "Piso en Santa Rosa".

Como Marco está interesado en un intercambio en España, el Sistema le da la opción a Sergio de proponer un intercambio.

INTERCAMBIO				
Selecciona uno de tus alojamientos para proponer un intercambio				
	<b>País</b>	<b>Provincia</b>	<b>Fecha publicación</b>	
<a href="#">Apartamento Alcossebre</a>	España	Castellón	11/05/2009	<input checked="" type="radio"/>
<input type="button" value="Proponer alojamiento"/>				

Fig. 29 - Alojamientos publicados por Sergio.

Cuando Sergio hace click en "Proponer intercambio" el Sistema muestra sus alojamientos publicados para que elija uno para proponer a Marco un intercambio.

<b>Fecha disponibilidad:</b>	20/12/2009 hasta 25/12/2009
<b>Solicitar reserva</b> <i>Selecciona las fechas en las que deseas solicitar tu estancia en este alojamiento</i>	
Fecha entrada	<input type="text"/>
Fecha salida	<input type="text"/>
<input type="button" value="SOLICITAR"/>	
<b>Solicitar intercambio</b> Alojamiento propuesto: <a href="#">Apartamento Alcossebre</a> <i>Selecciona las fechas en las que deseas solicitar tu estancia en este alojamiento</i>	
Fecha entrada	<input type="text" value="23/12/2009"/>
Fecha salida	<input type="text" value="25/12/2009"/>
<input type="button" value="SOLICITAR"/>	

Fig. 30 - Formulario de Reserva del alojamiento “Piso en Santa Rosa”, con alojamiento propuesto.

Sergio selecciona uno de sus alojamientos y acepta. El Sistema muestra a Sergio el alojamiento que quiere proponer para el intercambio y le da la opción de seleccionar las fechas deseadas para solicitar la reserva.

<b>Información de Contacto de Marco</b>	
<b>Nombre:</b>	Marco Gutierrez Tarego
<b>Telefono:</b>	689 56 47 33
<b>Email:</b>	marco@gmail.com
Reserva solicitada	

Fig. 31 - Notificación de Reserva Solicitada.

Una vez que Sergio ha seleccionado las fechas deseadas, se registra la reserva solicitada y el Sistema le muestra la información de contacto de Marco para que pueda contactar con él y conocerse mejor.

MIS RESERVAS						
Consulta tus Reservas						
Alojamiento	Propietario	Intercambio	Estado	Fecha entrada	Fecha salida	
<a href="#">Piso en Santa Rosa</a>	<a href="#">Marco</a>	<a href="#">Ver</a>	 Solicitada	23/12/2009	25/12/2009	

Fig. 32 - Reservas de Sergio.

Si Sergio consulta sus reservas, verá como se ha añadido una nueva reserva solicitada pendiente de ser confirmada para el alojamiento de Marco.

Marco, por su parte, si consulta sus alojamientos verá que tiene una nueva reserva solicitada con propuesta de intercambio para su alojamiento llamado “Piso en Santa Rosa”.

También podemos ver que se ha propuesto un intercambio.

MIS ALOJAMIENTOS						
Gestiona tus Alojamientos						
Nombre	Fecha publicación					
<a href="#">Piso en Santa Rosa</a>	10/05/2009 <a href="#">Editar</a> 					
Reservas						
Intercambio	Estado	Solicitante	Fecha entrada	Fecha salida		
	<a href="#">Ver</a>	 Solicitada	<a href="#">Sergio</a>	23/12/2009	25/12/2009	<a href="#">Intercambiar</a> 

Fig. 33 - Alojamientos de Marco.

Marco puede confirmar esta reserva haciendo click en “Intercambiar”. De esta forma Marco aceptará la reserva solicitada con intercambio de Sergio.

Buscador	Apartamento Alcossebre
País <input type="text" value="[Selecciona País]"/> Provincia <input type="text" value=""/> Fecha entrada <input type="text" value=""/> Fecha salida <input type="text" value=""/> Habitaciones <input type="text" value="1"/> (mínimo) Compartido <input type="text" value="Indiferente"/> Ofrezco intercambio en: <input type="text" value="[Selecciona País]"/> <input type="button" value="BUSCAR"/>	Publicado por <a href="#">Sergio</a> <b>País:</b> España <b>Provincia:</b> Castellón <b>Tipo:</b> Apartamento <b>Descripción:</b> Maravilloso apartamento de lujo en primerísima línea de playa, ubicado en la mejor zona de Alcossebre, con vista al mar y a la piscina (niños y adultos). 16 m <sup>2</sup> de terrazas. Gran plaza de garaje y trastero incluidos. 2 Dormitorios, 2 Baño(s), 75 Metros Útiles. Costa del Azahar <b>Nº Habitaciones</b> 2 <input type="button" value="Ver Ubicación Geográfica"/> <div style="text-align: right;"><b>Fotos</b></div> 
Bienvenido Marco	
<input type="button" value="Home"/> <a href="#">Mis Alojamientos</a>	

Fig. 34 - Alojamiento de Sergio, “Apartamento Alcossebre”.

Al hacer click en “Intercambiar”, el Sistema mostrará el alojamiento que Sergio ha propuesto como intercambio, para que Marco pueda consultarlo y seleccionar las fechas deseadas para la reserva.

<b>Fecha disponibilidad:</b>	01/09/2009 hasta 18/09/2009
<b>Reservas</b>	
<b>Solicitar reserva</b> <i>Selecciona las fechas en las que deseas solicitar tu estancia en este alojamiento</i> Fecha entrada <input type="text" value=""/> Fecha salida <input type="text" value=""/> <input type="button" value="SOLICITAR"/>	<b>Solicitar intercambio</b> Alojamiento propuesto: <a href="#">Piso en Santa Rosa</a> <i>Selecciona las fechas en las que deseas solicitar tu estancia en este alojamiento</i> Fecha entrada <input type="text" value=""/> Fecha salida <input type="text" value=""/> <input type="button" value="SOLICITAR"/>

Fig. 35 - Formulario de Reserva del alojamiento de Sergio, “Apartamento Alcossebre”.

Vemos como el Sistema automáticamente indica el alojamiento de Marco como alojamiento propuesto para un intercambio.

MIS ALOJAMIENTOS					
Gestiona tus Alojamientos					
Nombre		Fecha publicación			
<a href="#">Piso en Santa Rosa</a>		10/05/2009		<a href="#">Editar</a>	
Reservas					
Intercambio	Estado	Solicitante	Fecha entrada	Fecha salida	
	<a href="#">Ver</a>	<input checked="" type="checkbox"/> Confirmada	<a href="#">Sergio</a>	23/12/2009	25/12/2009

Fig. 36-Alojamientos de Marco. Reserva de Sergio confirmada para el Alojamiento “Piso en Santa Rosa”.

Cuando Marco ha seleccionado las fechas deseadas y ha pedido la reserva, automáticamente se ha confirmado la reserva solicitada por Sergio.

Por otra parte, si vamos a ver los alojamientos de Sergio, veremos que se ha añadido una nueva reserva para el alojamiento que él había propuesto para intercambio en la reserva del alojamiento de Marco, y es una reserva automáticamente confirmada.

Es decir, las reservas con intercambio se han realizado con éxito, ahora hay un intercambio de alojamientos entre Sergio y Marco.

MIS ALOJAMIENTOS					
Gestiona tus Alojamientos					
Nombre		Fecha publicación			
<a href="#">Apartamento Alcossebre</a>		11/05/2009		<a href="#">Editar</a>	
Reservas					
Intercambio	Estado	Solicitante	Fecha entrada	Fecha salida	
	Solicitada	<a href="#">Ana</a>	01/09/2009	10/09/2009	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
	<a href="#">Ver</a>	<input checked="" type="checkbox"/> Confirmada	<a href="#">Marco</a>	06/09/2009	17/09/2009

Fig. 37 - Alojamientos de Sergio. Reserva de Marco confirmada para el alojamiento “Apartamento Alcossebre”.

## 10. Conclusiones y Ampliaciones

La aplicación cumple los objetivos básicos que nos habíamos marcado al comenzar el proyecto.

Permite registrarse en la comunidad como usuario para poder disponer de las ventajas de los usuarios registrados, así como editar el perfil.

Permite dar de alta y publicar un alojamiento, indicando la ubicación en un mapa interactivo, especificar fotos, editarlo y borrarlo.

Permite que un usuario registrado gestione sus alojamientos y las reservas de éstos, así como consultar el estado las reservas que ha efectuado y sus estancias.

Permite acceder a toda la información de un alojamiento, añadir comentarios y realizar reservas y/o intercambios.

Permite ver información de otros usuarios y valorarlos.

Permite realizar búsqueda personalizadas de alojamientos, con una visualización de resultados basada en Google Maps, indicando la ubicación exacta de los alojamientos en un mapa interactivo.

En definitiva, la aplicación está preparada para albergar una red social donde los usuarios pueden ofrecer y encontrar alojamientos a nivel mundial (la lista de países está pendiente de ampliar), de forma rápida y sencilla, contando con la atractiva visualización de la ubicación de todos los alojamientos en un Google Map, ventajas para usuarios registrados y un sistema de reservas e intercambios.

A nivel personal existen algunas cosas no del todo satisfactorias, como un diseño visual de la web que peca de ser demasiado simple, la ausencia de inserción de vídeos para los alojamientos o la no posibilidad de que la aplicación almacene las fotos en el servidor y deban guardarse direcciones URL.

### **Proyectos futuros**

Como proyecto futuro entorno a la base propuesta por esta aplicación se propone incluir una serie de características:

Inserción de vídeos e imágenes de forma ilimitada en el propio servidor para ilustrar y poder ver los alojamientos con más detalle, dándole a la web un punto de vista más visual si cabe.

Para evitar la creación múltiple de cuentas de usuario e impedir burlar la limitación de solicitud de reservas del sistema, se deberá implementar un sistema de seguridad que identifique a los usuarios de forma más precisa, por medio de IP u otro tipo de mecanismo.

Sistema de valoración de usuarios más complejo, con una valoración por cada estancia y permitiendo saber el usuario que ha realizado la valoración.

Integración total con Google Maps, incluyendo cálculo de rutas y el uso de *Street View*, explotando todas las posibilidades del servicio de Google.

Inserción de un foro y un sistema de envío de mensajes entre usuarios para dar más fuera al concepto de comunidad.

Inserción de chat.

Para solucionar otro de los inconvenientes de viajar, se dará la posibilidad de asociar a cada alojamiento un medio de transporte para no sólo ofrecer alojamiento, sino también una forma de trasladarse una vez se esté alojado.

## 11. Bibliografía

[1] [http://www.programacion.com/java/articulo/tips\\_struts/](http://www.programacion.com/java/articulo/tips_struts/)

JavaWorld. *Evitar los obstáculos del desarrollo con Struts.*

[2] <http://code.google.com/intl/es/apis/maps/documentation/index.html>

*Conceptos sobre el API de Google Maps*

[3] <http://code.google.com/intl/es/apis/maps/articles/yourfirstmap.html>

*Tutorial: Creating Your First Map*

[4] <http://code.google.com/intl/es/apis/maps/documentation/reference.html>

*Referencias para el API de Google Maps*

[5] <http://itouchmap.com/latlong.html>

*Longitud y Latitud de un Punto*

[6] <http://www.floob.es/2008/08/25/coordenadas-de-una-direccion-para-googlemaps/>

*Coordenadas de una dirección para Google Maps.* 25 de Agosto de 2008.

[7] <http://www.a2sistemas.com/blog/2009/03/20/obtener-la-latitud-y-longitud-de-una-direccion-con-google-maps-y-c/>

*Obtener la latitud y longitud de una dirección con Google Maps y C#.* 20 de Marzo de 2009.

[8] <http://www.programacionweb.net/articulos/articulo/?num=551>

*Mostrar ciudad en un mapa – G2Map.* 20/10/07

[9] [http://www.w3schools.com/jsref/jsref\\_onClick.asp](http://www.w3schools.com/jsref/jsref_onClick.asp)

*Javascript onclick Event*

[10] <http://www.lsi.us.es/docencia/get.php?id=2089>

Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla.  
*Diseño de la capa de presentación.*

[11] <http://www.proactiva-calidad.com/java/jdbc/conexion.htm>

*JDBC, Conexión a la base de datos.*

[12] <http://www.slideshare.net/siis/struts-en-java>

Ana María Pacheco Aspilcueta. *Struts en Java.*

[13] <http://www.webtaller.com/construccion/lenguajes/javascript/lecciones/elementos-formulario-select-asociados-javascript.php>

Mundomanuales.com. *Elementos de formulario select asociados con Javascript.*

[14] <http://www.anieto2k.com/2006/11/21/crear-una-tabla-de-contenidos/>

aNieto2k. *Crear una tabla de contenidos mediante Javascript.*

[15] [http://www.devjoker.com/asp/indice\\_contenido.aspx?co\\_grupo=PLSQ](http://www.devjoker.com/asp/indice_contenido.aspx?co_grupo=PLSQ)

Pedro Herrarte Sánchez. *Tutorial PL/SQL.* 26/07/2006

[16] <http://www.desarrolloweb.com/manuales/desarrollo-con-api-de-google-maps.html>

Miguel Angel Alvarez. *Manual del API de Google Maps.* Desde Julio de 2007.

[17]

[http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=Struts+Best+Practices&PHPSESSID=2ec2da95fb9bc745e0c62f5292896b5f#Manejo\\_Efectivo\\_de\\_excepciones](http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=Struts+Best+Practices&PHPSESSID=2ec2da95fb9bc745e0c62f5292896b5f#Manejo_Efectivo_de_excepciones)

Gerardo A. Oliveto. *Mejores prácticas de Struts*. 29/06/2005.

[18] <http://www.desarrolloweb.com/articulos/1489.php>

Eugenia Bahit. *Los objetos select y option*. 07/05/2004.

[19] Alfons Palacios i González, Montse Rabassa i Jou. *Sistemes gestors de bases de dades. Volum I. Volum II*. Febrer 2008. Escola Universitària Politècnica de Mataró.

[20] Jose Roure Alcobé, Joan Jou Majó. *Laboratori d'Informàtica 2*. Setembre 2007. Escola Universitària Politècnica de Mataró.

[21] Craig Larman. *UML y Patrones*. 2ª Edición. 2002. Ed. Prentice Hall

[22] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=validacionStruts>

Alberto Carrasco Montenegro. *Extender la validación en Struts*. 2006-02-17

## Anexo

ANEXO1: Contenido del CD-ROM

ANEXO2: Artículo

ANEXO3: Diagrama de Clases en formato PNG (alta resolución)

ANEXO 4: Modelo Conceptual en formato PNG

ANEXO 5: Modelo Relacional en formato PNG

ANEXO 6: Diagrama de Secuencia del caso de uso “Solicitar Reserva” incorporado en el proyecto NetBeans.