



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación web de alquiler de espacios y red social para compartir pisos: Front-End

PROYECTO FINAL DE CARRERA
INGENIERÍA INFORMÁTICA

Autor: D.JOSÉ MIGUEL ESTEVE ESTEVE

Director: D.FELIX BUENDÍA GARCÍA
D.CARLOS TAVARES CALAFATE

15 de julio de 2013

Índice general

1. Introducción	4
1.1. Resumen	4
1.2. Contexto	4
1.3. Objetivos	5
1.4. Trabajo realizado	5
1.5. Estructura de la memoria	5
2. Especificación de requisitos	7
2.1. Introducción	7
2.1.1. Propósito	7
2.1.2. Alcance	7
2.1.3. Definiciones, siglas y abreviaciones	8
2.1.4. Visión global	10
2.2. Descripción general	10
2.2.1. Perspectiva producto	10
2.2.2. Funciones del producto	11
2.2.3. Características de los usuarios	12
2.2.4. Restricciones	13
2.2.5. Supuestos y dependencias	14
2.2.6. Requisitos futuros	15
2.3. Requisitos Específicos	15
2.3.1. Requisitos de las interfaces externas	15
2.3.2. Requisitos funcionales	18
2.3.3. Requisitos de rendimiento	35
2.3.4. Restricciones del diseño	35
2.3.5. Atributos del sistema software	35
2.3.6. Otros requisitos	37
3. Análisis	38
3.1. Introducción	38
3.2. Diagramas de casos de uso	38
3.2.1. Actores de la aplicación web	38
3.2.2. Administrador	39
3.2.3. Usuario No Registrado	41
3.2.4. Usuario Registrado	42
3.2.5. Usuario Arrendador	42

3.2.6.	Usuario Arrendatario	43
3.2.7.	Usuario que comparte espacios	43
3.3.	Diagramas de clases	44
3.3.1.	Usuarios de la aplicación	44
3.3.2.	Administrador	45
3.4.	Diagramas de secuencia	46
3.4.1.	Autenticación Usuario	47
3.4.2.	Registro Usuario	48
3.4.3.	Publicar Anuncio	49
3.4.4.	Buscar y seleccionar espacio para reservar	50
3.4.5.	Comparar espacios	51
3.4.6.	Reservar espacio	52
3.4.7.	Editar espacio	53
4.	Diseño	54
4.1.	Introducción	54
4.2.	Arquitectura del sistema	54
4.3.	Modelo	55
4.4.	Vista	56
4.5.	Controlador	57
5.	Implementación	59
5.1.	Introducción	59
5.2.	Tecnologías	59
5.2.1.	SYMFONY	59
5.2.2.	HTML	60
5.2.3.	CSS	61
5.2.4.	JAVASCRIPT	61
5.2.5.	AJAX	61
5.2.6.	XML	61
5.2.7.	JSON	62
5.2.8.	API GoogleMaps	62
5.2.9.	API JQuery User Interface	62
5.3.	Herramientas y entorno	62
5.3.1.	Entorno de desarrollo	62
5.3.2.	Control de versiones	62
5.3.3.	Servidor web	63
5.3.4.	Base de datos	63
5.3.5.	Editor de imágenes	63
5.4.	Descripción de la implementación	63
5.4.1.	Estructura de la aplicación web	63
5.4.2.	Interfaces de usuario	68

6. Evaluación	75
6.1. Introducción	75
6.2. Validación de estándares web	75
6.2.1. Estándar HTML	75
6.2.2. Estándar CSS	77
6.2.3. Estándar Javascript	79
6.3. Análisis de navegación	79
7. Conclusiones	82
7.1. Resumen	82
7.2. Valoración	83
7.3. Trabajo futuro	83
8. Referencias bibliográficas	84

Capítulo 1

Introducción

1.1. Resumen

Este documento contiene la memoria del proyecto final de carrera. El objetivo de este proyecto es la creación de una aplicación web para que cualquier persona pueda publicar, alquilar o compartir espacios.

Para cumplir con esta meta se ha realizado una especificación de requisitos, a partir de ésta un análisis y diseño de la aplicación. A continuación se ha llevado a cabo una primera implementación la cual se ha evaluado mediante una serie de pruebas específicas. En cada una de estas etapas cuando se detecta un error o una posible mejora se revisa la etapa correspondiente, volviendo a repetir el proceso hasta llegar al producto esperado.

1.2. Contexto

El ámbito de esta aplicación se centra en el alquiler de espacios online. Se entiende por espacio a cualquier tipo de vivienda desde casas, pisos, habitaciones, cabañas o camarotes hasta un sofá.

Existen dos perfiles de usuario diferenciados, el arrendador y el arrendatario. El arrendador publica anuncios de espacios. El arrendatario busca espacios para alquilar en un periodo determinado. Existe una interacción entre arrendador y arrendatario para fijar las condiciones de alquiler. Si llega a término el arrendatario disfruta del espacio y el arrendador recibe su correspondiente compensación económica.

El propósito concreto de la aplicación web es alquilar espacios en estancias cortas vacacionales o de negocios. Aunque el arrendador y arrendatario tienen la flexibilidad de establecer la estancia que ellos deseen. Para ello mediante la aplicación web se pone en contacto a arrendatario y arrendador, con el servicio opcional de mediador de pagos que se encarga de efectuar las transacciones de forma segura.

1.3. Objetivos

La finalidad principal de esta aplicación es realizar el máximo número de alquileres de manera satisfactoria. Para conseguir esta meta aparecen como objetivos secundarios, el aumentar la ocupación de las viviendas, optimizar los tiempos de publicación y alquiler de los espacios, mejorar la facilidad de uso, y manejo intuitivo. Otro objetivo es el de conseguir que el mayor número de usuarios utilice la aplicación.

1.4. Trabajo realizado

El proyecto web tiene dos partes Front-End que es la parte de la aplicación que interactúa con el usuario y Back-End que es la que procesa la entrada desde el Front-End.

Como el proyecto web consta de dos partes para poder trabajar en equipo se han utilizado herramientas de control de versiones y un entorno de trabajo común. También se ha planificado el trabajo a realizar para avanzar de manera conjunta.

Desde el inicio se ha utilizado un alojamiento web profesional con el objetivo de depurar los errores en las etapas más tempranas del desarrollo y trabajar en un entorno real.

El trabajo realizado en este proyecto se centra en la parte Front-End. Al ser la parte visible de la aplicación, es decir la parte que ve el usuario final, se ha tenido especial atención al diseño de la aplicación, haciendo una interfaz rápida, sencilla y agradable para el usuario.

En este proyecto se ha diseñado y modelado la aplicación web. Además se ha estructurado la aplicación web en elementos desacoplados para mejorar la mantenibilidad. Se han creado los distintos interfaces con el usuario utilizando un diseño común. Para ello se han definido y diseñado los estilos de las páginas web. También se han incorporado a la aplicación web una de las API (interfaces de programación de aplicaciones) de servidor de mapas y otra de redes sociales.

Finalmente se han creado aplicaciones de comunicación de forma asíncrona entre la parte Front-End y la parte Back-End para intercambiar datos sin tener que recargar la página web.

1.5. Estructura de la memoria

La memoria contiene nueve capítulos.

El primero de ellos es la introducción, en la que se describe el documento de la memoria, el contexto y objetivos de la aplicación, el trabajo realizado en este proyecto y la estructura de la memoria.

El segundo capítulo consiste en la especificación de requisitos siguiendo el estándar IEEE Std.830-1998.[3] Este estándar consiste en la definición de las características que debe cumplir la aplicación web para satisfacer las condiciones y objetivos propuestos de la forma más eficiente posible.

El tercer capítulo trata sobre el análisis de la aplicación. En este análisis se detalla la estructura y el comportamiento de ésta. Para ello se describe el flujo de la aplicación y las funciones que puede realizar cada perfil de usuario. Basándose en el modelo UML se han utilizado casos de uso, diagramas de clases y diagramas de secuencia.

El cuarto capítulo recoge el diseño de la aplicación, describiendo la arquitectura del sistema. Se sigue el patrón de diseño de arquitectura por capas adaptado a la web, diferenciando en tres capas. La primera capa, denominada capa presentación, es la que ve el usuario. La segunda es la capa de negocio, es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Y la tercera es la capa de datos, es donde residen los datos y se encarga de acceder a los mismos.

El quinto capítulo desarrolla la implementación, donde se detallan las tecnologías utilizadas, as herramientas y el entorno de desarrollo. Además de entrar en detalles concretos de la implementación.

El sexto capítulo especifica la evaluación realizada al sistema. Se han utilizado distintos tipos de pruebas que miden el correcto funcionamiento, el rendimiento y la calidad del software. También se hace un análisis de uso y una validación de los estándares web utilizados.

El séptimo capítulo, hace referencia a las conclusiones obtenidas en la realización del proyecto. Se hace una valoración de las mismas y se indican algunos potenciales trabajos futuros.

El capítulo ocho indica las referencias bibliográficas para realizar el proyecto y la memoria.

En el último capítulo se añaden los anexos, es decir, el material adicional utilizado.

Capítulo 2

Especificación de requisitos

2.1. Introducción

2.1.1. Propósito

Esta especificación de requisitos tiene como propósito establecer el conjunto de características y funcionalidades de la aplicación web. Estas funcionalidades deben satisfacer las necesidades de los distintos usuarios del sistema. La especificación de requisitos cumple con el estándar IEEE Std.830-1998. [3]

Este documento sirve para formalizar las funcionalidades de la aplicación para todos los usuarios que intervienen en el sistema como el administrador, el usuario arrendador^[2] y el usuario arrendatario^[3].

2.1.2. Alcance

Este proyecto nace a raíz de la necesidad de poder alquilar un alojamiento en periodos cortos de tiempo de manera fácil y rápida, con el objetivo de que cualquier persona pueda alquilar su espacio sin ningún coste. El arrendatario se beneficia obteniendo mejores precios y periodos de alquiler más flexibles. De esta idea inicial surge otra, que consiste en compartir espacios^[10] entre los usuarios. En primera instancia la idea era de especializarse en alquiler para estudiantes terminando por cubrir todos los segmentos de población.

El enfoque que se pretende obtener con la aplicación es intuitivo, rápido, moderno y orientado a la explotación empresarial. Está basado en las últimas tecnologías web e íntimamente integrado con las redes sociales.

La aplicación debe ser capaz de conseguir que un perfil de usuario no experto pueda publicar o alquilar un anuncio. También un usuario novel tiene que poder compartir su espacio con otras personas. La principal ventaja que tiene la modalidad compartida respecto al tradicional es que los usuarios pueden conocerse antes de compartir el piso.

Para la puesta en funcionamiento de la aplicación será necesario adquirir un dominio y contratar un hosting^[20] para el hospedaje de la web. El nombre de la aplicación a realizar es lookyourhome^[22] por lo que el dominio más conveniente es el de **lookyourhome.com**.

2.1.3. Definiciones, siglas y abreviaciones

En esta subsección se definirán todos los términos, siglas y abreviaturas utilizadas en la especificación de requisitos.

1. **API:** Application programming interface, interfaz de programación de aplicaciones. Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. [4]
2. **Arrendador:** El usuario que publica los anuncios de espacios para alquilar.
3. **Arrendatario** El usuario que utiliza los servicios disponibles en la aplicación para alquilar y compartir espacios.
4. **Autenticación:** Acción mediante la cual un usuario se identifica en el sistema. Para ello el usuario utiliza un nombre de usuario y una contraseña.
5. **Base de datos:** Es conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
6. **Contraseña:** Palabra de paso para acceder al sistema. Se utiliza en la autenticación.
7. **CSS:** Lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas.
8. **CP:** Código postal.
9. **ERS:** Especificación de requisitos.
10. **Espacio:** Es el tipo de alojamiento para alquilar.
11. **Estancia mínima:** Los días que como mínimo se puede alquilar un espacio.
12. **Estancia máxima:** Los días que como máximo se puede alquilar un espacio.
13. **Facebook:** Sitio web de redes sociales. <http://facebook.com>
14. **FAQ:** Frequently Asked Questions, preguntas de uso frecuente.
15. **Fecha entrada:** Fecha a partir de la cual empieza el periodo de alquiler.
16. **Fecha salida:** Fecha a partir de la cual termina el periodo de alquiler.
17. **Fianza:** Cantidad de dinero que el arrendador solicita como depósito como garantía para poder alquilar su espacio.

18. **GoogleMaps:** Servidor de aplicaciones de mapas en la web. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo incluso la ruta entre diferentes ubicaciones.
19. **HTML:** Lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en formato de texto.
20. **Hosting:** Es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes , vídeo o cualquier contenido accesible vía web.
21. **JavaScript:** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript [15]
22. **Lookyourhome:** Nombre de la aplicación web que se va a crear en este proyecto.
23. **Navegador web:** Aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que éstos puedan ser leídos.
24. **Nombre de usuario:** Palabra única en el sistema que identifica a un sólo usuario. Se utiliza en la autenticación.
25. **Paypal:** Es una empresa perteneciente al sector del comercio electrónico por Internet que permite la transferencia de dinero entre usuarios.
26. **PHP:** Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
27. **Publicar anuncio:** Proceso a realizar para publicar un anuncio en la aplicación.
28. **Push:** Describe un estilo de comunicaciones sobre Internet donde la petición de una transacción se origina en el servidor.
29. **Reservar anuncio:** Proceso a realizar para reservar un anuncio en la aplicación.
30. **Spam:** Los mensajes no solicitados, no deseados, o de remitente no conocido, habitualmente de tipo publicitario, generalmente enviados en grandes cantidades que perjudican de alguna o varias maneras al receptor. [9]
31. **SGDB:** Sistema de gestión de bases de datos. Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos^[5], además de proporcionar herramientas para añadir, borrar, modificar y analizar datos.
32. **SQL:** Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.
33. **Tipo habitación:** El espacio puede ser compartido por varios usuarios o no.
34. **Tipo propiedad:** El tipo de espacio puede ser un apartamento, barco, cabina, chalet, casa, cabaña, ático y otros.

2.1.4. Visión global

Este ERS se organiza en dos secciones, descripción general y requisitos específicos.

En la descripción general se detalla la perspectiva y funciones de la aplicación web, las características generales de los distintos tipos de usuario, las restricciones globales del sistema y los factores que pueden afectar a los requisitos.

En la sección de requisitos específicos se describe los requerimientos a un nivel de detalle suficiente para permitir a los diseñadores diseñar un sistema que cumpla estos requisitos, que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. [14, p. 6] Para ello se detallaran las interfaces externas del sistema, los requisitos funcionales, los requisitos relacionados con la carga que tendrá que soportar el sistema, todo aquello que restrinja las decisiones relativas al diseño de la aplicación, los atributos de calidad del sistema y cualquier otro requisito relacionado con el sistema.

2.2. Descripción general

2.2.1. Perspectiva producto

Estudio de mercado

Esta subsección realiza una comparativa entre las aplicaciones web de alquiler de viviendas más representativos del sector.

La más representativa es Airbnb, <http://airbnb.com>, destaca a nivel visual además de ofrecer una red social para que interactúen clientes y proveedor. Aporta el concepto de espacio. Las ventajas que ofrece es la un seguro de daños para el arrendador, sistema de evaluaciones de los espacios por los arrendatarios, atención al usuario 24 horas y el ámbito es mundial con un amplio catálogo de casas. Como desventaja cobra una comisión excesiva. Otras similares a esta son housetrip, <http://housetrip.com>, y Wimdu, <http://wimdu.com>

Otra web interesante es Valenciaflats, <http://valenciaflats.com>, se caracteriza por gestionar una cartera de pisos en propiedad. Como desventaja el usuario no puede publicar anuncios. El ámbito de negocio es local a Valencia.

Web Living valencia, <http://livingvalencia.es>, tiene una línea más tradicional de alquiler por internet. Tiene como peculiaridad que ofrece cualquier tipo de servicio adicional. Entre sus desventajas cabe destacar que cobran una comisión alta y el servicio de atención al cliente no es 24 horas. El ámbito es local a Valencia.

Web es Donde dormir, <http://dondedormir.com>, ventaja tiene una atención personalizada, la web es muy fácil y cómoda de usar. El ámbito es mundial. Aportan una opción de ofrecer colaboraciones para difundir su página. Como desventaja las comisiones son altas.

Por último la web de Homelidays, <http://homelidays.com>, la página es conocida a nivel mundial, comercializan un seguro para cubrir posibles infortunios. El equipo de atención al cliente es amplio. Garantía de reembolso. Interactúa con un sistema de alquiler de coches. Tiene como desventaja que cobra importe fijo por hospedar el anuncio en su web.

Nuestro producto

El producto a desarrollar es una aplicación web de alquiler de espacios y a la vez una red social para compartir pisos.

Esta aplicación pretende recoger las mejores ideas de las páginas web más representativas y aportar varias ideas novedosas que no se encuentran en ninguna otra aplicación del mercado.

Como ideas seleccionadas de otras páginas web se encuentran la inclusión del concepto de espacios, un sistema de evaluaciones de los espacios, periodos de alquiler flexibles y cortos, y ámbito global.

Las ideas aportadas son: destinada a cualquier tipo usuario, totalmente gratuita y se ofrece la posibilidad de mediador de pagos con muy bajas comisiones, y una plataforma para compartir pisos.

2.2.2. Funciones del producto

A continuación se muestra un resumen de las principales funcionalidades de la aplicación web.

- **Búsqueda de espacios:** El usuario puede buscar espacios de una manera intuitiva, estableciendo unos criterios de búsqueda. El sistema de proporcionar un listado de los espacios que se ajusten a las restricciones de la búsqueda realizada.
- **Selección de espacio:** El usuario puede visualizar los espacios buscados previamente y elegir el más óptimo a sus intereses mediante una interfaz natural priorizando la usabilidad de la misma.
- **Comparar espacios:** Se puede hacer una comparativa de los espacios seleccionados anteriormente o mediante nuevas búsquedas. Todos los espacios se van agregando a la interfaz del comparador de espacios. En esa interfaz se visualizan las distintas valoraciones y características de los espacios.

- **Reserva de espacios:** Si el usuario se decide puede proceder a reservar el espacio utilizando la interfaz rápida destinada a este propósito. Esta funcionalidad está relacionada con la interfaz bancaria externa a esta aplicación.
- **Evaluación de espacios:** Una vez el usuario ha disfrutado de la estancia puede realizar una valoración del espacio. Para ello puede realizar un comentario a través de la aplicación. Paralelamente el usuario puede mostrar su opinión en las redes sociales utilizando la API^[1] de Facebook.
- **Registro de usuario:** El usuario puede registrarse en la aplicación introduciendo sus datos en un sólo paso.
- **Autenticación de usuario:** Para poder realizar determinadas acciones de la aplicación es necesario autenticarse previamente y haberse registrado. El usuario introduce su nombre de usuario^[24] y contraseña^[6]. Opcionalmente el usuario también puede autenticarse mediante Facebook.
- **Gestión perfil de usuario:** El usuario puede modificar sus datos personales en cualquier momento y cambiar el estado público o privado de su perfil.
- **Publicar un anuncio:** En la aplicación se puede agregar un anuncio introduciendo los datos básicos del espacio, haciendo uso del API^[1] de GoogleMaps^[18].
- **Gestión de anuncios:** Dentro de la aplicación se pueden modificar los datos específicos de los espacios publicados anteriormente. Además de cambiar el estado entre visible en la web u oculto.
- **Gestión del idioma:** La aplicación por defecto usa el idioma del navegador web^[23]. En todo momento, el usuario puede cambiar el idioma entre el inglés, francés y español.
- **Ayuda al usuario:** Se proporciona una ayuda interactiva dentro de la aplicación que va indicando los pasos a seguir para cada funcionalidad.
- **Gestión de datos de la aplicación:** Se visualizan los datos de la aplicación web, la normativa, FAQ^[14], datos referentes a la aplicación y página de contacto.

2.2.3. Características de los usuarios

En la aplicación web convivirán cinco tipos de usuarios según la función que utilizan o desempeñan en la web:

- **Usuario registrado:** es el usuario que tiene acceso a contenidos privados en la aplicación web mediante un registro previo en el servicio de manera gratuita. Este usuario puede gestionar su perfil. También puede administrar los mensajes que envía y recibe de otros usuarios. El usuario es una persona adulta que sepa utilizar de manera básica un navegador web, sin tener experiencia previa en la aplicación y con un nivel educacional medio. No necesita tener conocimientos técnicos.

- **Usuario no registrado:** es el usuario que busca un espacio sin tener que estar registrado en la aplicación. Este usuario puede buscar, comparar y visualizar espacios mediante unos criterios de selección y unos filtros de búsqueda. El perfil de usuario es el de una persona adulta que sepa utilizar de manera básica un navegador web, sin tener experiencia previa en la aplicación y con un nivel educacional medio. No necesita tener conocimientos técnicos.
- **Usuario arrendador:** es el *usuario registrado* que publica un anuncio de un espacio con el objetivo de alquilarlo. Este usuario publica y administra sus espacios. También administra las reservas que realizan otros usuarios a sus anuncios. Tiene que estar autenticado.
- **Usuario arrendatario:** es el *usuario registrado* que alquila un espacio anunciado en la aplicación web. Este usuario realiza la reserva de espacios y administra sus reservas. Tiene que estar autenticado.
- **Usuario que comparte espacios:** es el *usuario registrado* que tiene la intención de compartir un espacio con otros usuarios. Este usuario puede crear anuncios con el objetivo de compartir un espacio. También puede agregarse a otro anuncio para alquilarlo de manera conjunta con otros usuarios. Tiene que estar autenticado.
- **Usuario administrador:** es el usuario con privilegios especiales que gestiona el mantenimiento de la web. Gestiona los idiomas, control de Spam^[30], la base de datos^[5], la página de contacto, detalles y FAQ^[14] de la aplicación web. Tiene que estar autenticado. El administrador debe tener una experiencia previa en la aplicación, con un nivel educacional alto y con conocimientos técnicos avanzados.

Una persona que utilice la aplicación web puede desempeñar varios de estos roles. Así un usuario arrendatario puede a su vez ser un usuario arrendador y un usuario que comparte espacios.

2.2.4. Restricciones

Las limitaciones de hardware existentes se reducen a equipos que dispongan de un navegador web actualizado con conexión a Internet. Los navegadores web de escritorio compatibles son: Google Chrome, Mozilla Firefox, Safari, Internet Explorer y Opera. Los navegadores web móviles compatibles son: Safari, Chrome, Firefox y Opera.

Como restricción de interfaz con otras aplicaciones existe una limitación de uso de Googlemaps que establece un máximo de consultas al API de su servicio.

La aplicación tiene que ser escalable, pudiendo acceder a ella un número indeterminado de usuarios.

La aplicación tiene que estar disponible en todo momento para cualquier usuario.

Los datos del usuario permanecen ocultos, requiriendo autenticación para mostrarlos. Las contraseñas de los usuarios están encriptadas mediante un protocolo de cifrado seguro.

El lenguaje de la interfaz de usuario debe ser JavaScript^[21] junto con HTML^[19] y hoja de estilos CSS^[7], el lenguaje del servidor debe ser PHP^[26], el utilizado en el SGDB^[31] es SQL^[32]

En cuanto a requisitos de habilidad, la aplicación debe ser intuitiva y de fácil usabilidad.

La aplicación debe cumplir la política de privacidad de protección de datos para el tratamiento de los datos de carácter personal según la normativa vigente española. [1, 2]

Se restringen el método de pago Online, que se usa en la aplicación, a Paypal y pago con tarjeta de crédito. Aunque se da la posibilidad de pagar el inquilino directamente al arrendatario sin usar la aplicación web.

2.2.5. Supuestos y dependencias

Uno de los factores que, si cambia, puede afectar a los requisitos es el marco legal aplicable. El marco legal se ampara en las leyes vigentes que están en continuo cambio por la presión del sector hotelero. Podría darse el caso que si estas leyes cambian, entonces afecten a una funcionalidad de la aplicación como puede ser la regulación del alquiler por días o un cambio sustancial en los impuestos que no permita la viabilidad del negocio, por lo que habría que cambiar el modelo de negocio y los requisitos.

En caso de que los navegadores web dejen de dar soporte a los lenguajes establecidos en las restricciones (apartado 2.2.4), habrá que adaptar los requisitos a los nuevos lenguajes.

Cambios en la normativa de seguridad o privacidad habrá que ajustar los protocolos y tratamiento de datos de carácter personal.

Con el API de Googlemaps si se produce una restricción de los servicios o cambio en su funcionalidad puede afectar a la aplicación y por lo tanto habría que cambiar ciertos requisitos de la interfaz o cambiar a otro servicio externo que se adapte a las necesidades de la aplicación.

Si se produce un cambio en las condiciones de uso de Paypal^[25] o del banco proveedor de los pagos mediante tarjeta de crédito habría que modificar los requisitos relacionados con estas interfaces externas.

2.2.6. Requisitos futuros

Esta subsección esboza futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

La mejora más importante que se puede realizar es la creación de una aplicación adaptada para dispositivos móviles. Esta aplicación tiene como objetivo implementar las funciones básicas del producto pero con una interfaz específica para dispositivos móviles.

Crear un servidor Push^[28] para la aplicación móvil, que permita enviar mensajes al usuario del estado de sus reservas.

Desarrollar una aplicación hermana de alquiler de espacios similar a la especificada en este documento pero enfocada a unos periodos de alquiler por horas.

2.3. Requisitos Específicos

Esta subsección especifica todos los requisitos necesarios para el desarrollo de la aplicación web. Todos los requisitos aquí descritos reflejan alguna necesidad real del sistema, son modificables y se puede realizar su trazabilidad.

2.3.1. Requisitos de las interfaces externas

Requisitos de usuario

La interfaz de usuario se requiere que sea fácil de utilizar, innovadora y muy rápida.

Para que sea fácil de utilizar, deberá ser muy intuitiva, cualquier usuario sin una formación específica debe de poder usarla.

Para alcanzar el requisito de interfaz innovadora se recurrirá a la utilización de los lenguajes y tecnologías de vanguardia, siempre que haya suficiente documentación para poder utilizarlos y ofrezcan las garantías de estabilidad necesarias que la aplicación creada sea robusta.

Y el uso de la interfaz debe ser rápida para que los usuarios no desesperen en su utilización y no decidan abortar cualquier proceso iniciado a causa de las esperas interminables que suelen darse en otros entornos web.

Por todo ello la página de inicio de la aplicación web debe ser muy sencilla, un buscador rápido de espacios que filtre los resultados por localidad. Habrá una barra de navegación superior común a todas las interfaces de usuario. El aspecto final debe ser el de una aplicación web sin desplazamiento horizontal ni vertical, de modo que todo el contenido que se muestre debe ajustarse al marco de la pantalla que utilice el usuario. A priori se establecerán las medidas mínimas del marco en 980 píxeles de ancho y 680 píxeles de alto. En la figura 2.1 se detalla este interfaz de búsqueda inicial.

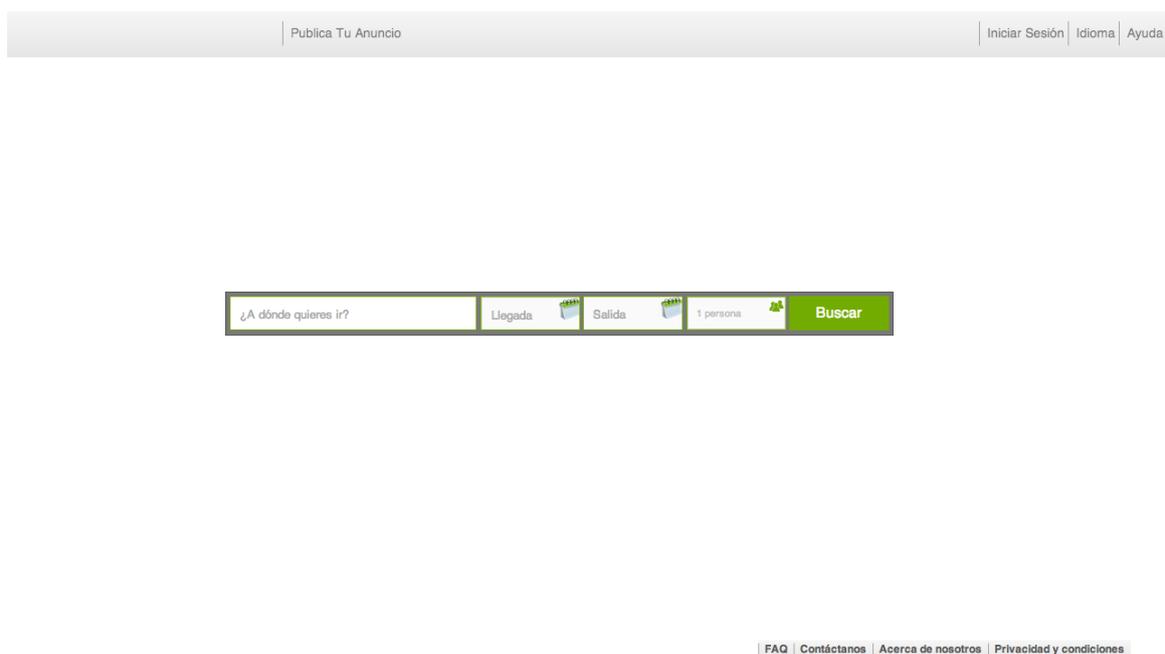


Figura 2.1: Interfaz inicial usuario.

Requisitos de hardware

Al tratarse de una aplicación web no existe un requisito de hardware directo. Los navegadores web en los que se podrá utilizar la aplicación funcionan en los sistemas operativos más populares.

Los navegadores web si tienen restricciones de hardware. Por lo que aplicando una regla transitiva estos mismos requisitos serían aplicables a la aplicación web. Pero en lugar de aplicar estas restricciones de hardware, se aplica la restricción de software de la compatibilidad con determinados navegadores web.

En definitiva no hay requisitos de hardware para esta aplicación web.

Requisitos de software

El único requisito de software que hay es la compatibilidad de la aplicación web con determinados navegadores. Tras analizar estas estadísticas se determina que

la aplicación web deberá ser compatible con las versiones de los navegadores que aparecen en el listado que acompaña al gráfico 2.2. Hay un caso particular que es para el navegador internet explorer que sólo tendrá que ser compatible con la última versión ya que las versiones anteriores no son compatibles con html5.

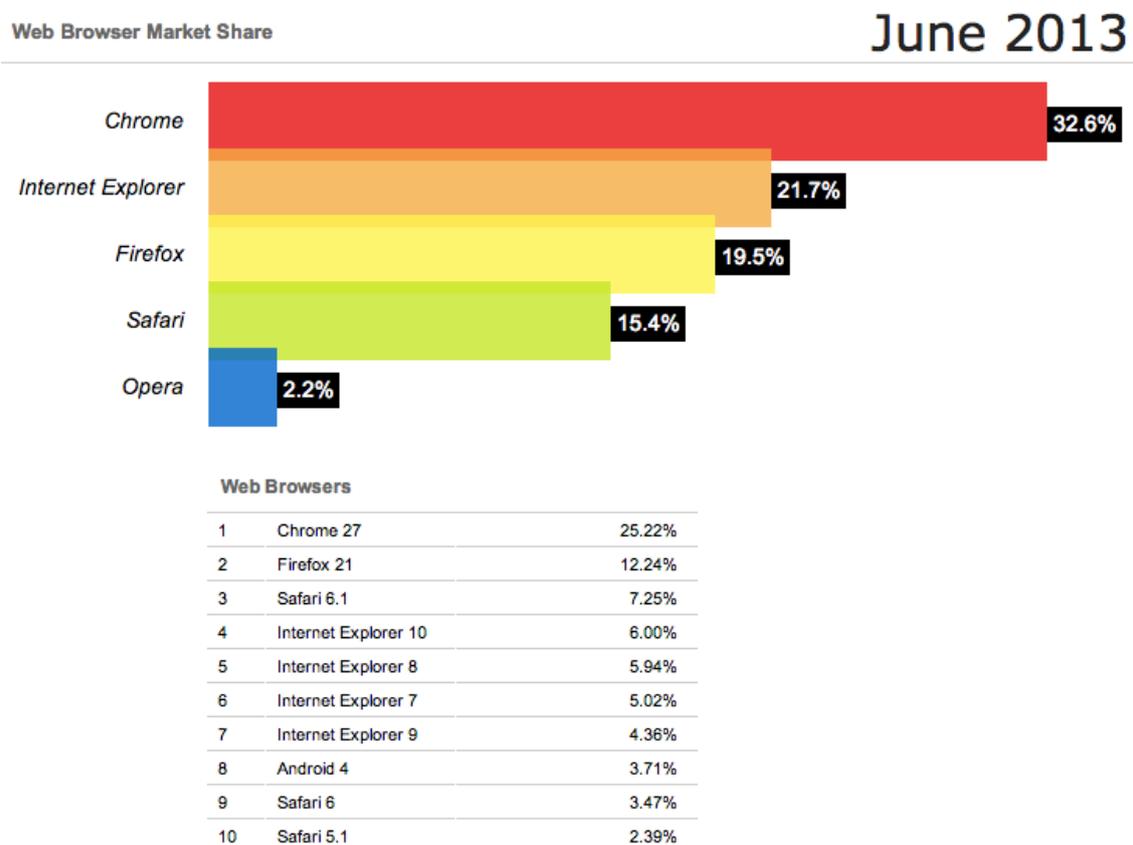


Figura 2.2: Gráfico uso mundial de los navegadores web [31].

Requisitos de comunicaciones

Al tratarse de una aplicación web no existe un requisito de comunicaciones directo.

Los navegadores web si tienen restricciones de comunicaciones. Por lo que aplicando una regla transitiva estos mismos requisitos serían aplicables a la aplicación web. Pero en lugar de aplicar estas restricciones de comunicación, se aplica la restricción de software de la compatibilidad con determinados navegadores web.

En definitiva no hay requisitos de comunicaciones para esta aplicación web.

2.3.2. Requisitos funcionales

Los requisitos funcionales de este proyecto se organizan siguiendo la tipología de usuario. Estos requisitos describen las distintas funcionalidades o casos de uso, que cada uno de los usuarios del sistema puede realizar.

Una funcionalidad es descrita como una introducción, un conjunto de entradas, un proceso y unas salidas. Estas funcionalidades a su vez están representadas con diagramas de casos de uso en la sección 3.2. A continuación se describen cada uno de los requisitos funcionales del sistema:

Administrador

Numero	Crear Nuevo Idioma
Introducción	El administrador puede crear un nuevo idioma en la aplicación que permite a cualquier usuario visualizar la página en ese idioma adicional.
Entradas	Código del idioma, nombre del idioma y descripción del idioma.
Proceso	El sistema muestra primero un formulario para crear el nuevo idioma. Los campos que debe rellenar el administrador son el código del idioma, nombre del idioma y la descripción del idioma. Antes de enviar los datos al servidor, el sistema avisa al administrador si hay algún error en algún campo. Si no hay errores el administrador envía el formulario al servidor. Si los datos son correctos se crea el nuevo idioma en la aplicación. El servidor entonces valida los datos y muestra una pantalla de confirmación o de fallo en caso de error.
Salidas	Aparece una pantalla de confirmación, o una de fallo en caso de error, avisando al administrador que el idioma se ha creado de manera satisfactoria.

Numero	Eliminar Idioma
Introducción	Se puede eliminar un idioma desde la administración de la aplicación. Los idiomas iniciales (Inglés, Francés y Español) no se pueden eliminar.
Entradas	Nombre o código del idioma.
Proceso	El administrador primero consulta el listado de idiomas y selecciona un idioma. Este idioma no puede pertenecer a los idiomas iniciales de la aplicación (Inglés, Francés y Español). Una vez seleccionado el administrador borra el idioma de la aplicación. El sistema se encarga de eliminar el idioma de la aplicación. El sistema muestra un mensaje de confirmación o de fallo en caso de error.
Salidas	Mensaje de eliminación satisfactoria del idioma o de fallo en caso de error.

Numero	Crear y Modificar Traducción
Introducción	El administrador puede crear y modificar traducciones de partes del texto de la aplicación a los distintos idiomas existentes.
Entradas	Idioma de la traducción, código del texto a traducir y texto traducido.
Proceso	El administrador consulta primero las partes de la aplicación que se pueden traducir y selecciona el texto que desea traducir. El sistema muestra entonces un formulario con los campos: idioma de la traducción y texto traducido. Y muestra también el código del texto a traducir y el texto en el idioma que desee el administrador. El administrador entonces introduce o modifica el texto traducido y selecciona el idioma de la traducción. El servidor valida entonces los datos y envía una pantalla de confirmación o de fallo en caso de error. En caso de que los datos sean válidos se crea o modifica la traducción elegida.
Salidas	Mensaje de creación o modificación satisfactoria de la traducción o de fallo en caso de error.

Numero	Eliminar Traducción
Introducción	El administrador puede eliminar una traducción creada anteriormente.
Entradas	Idioma de la traducción y código del texto a traducir.
Proceso	El administrador <i>Consulta las Traducciones</i> . El sistema muestra un listado con todas las traducciones existentes. El administrador elige la traducción y el idioma de ésta que desee eliminar. Entonces la aplicación elimina la traducción y muestra una pantalla de que todo ha ido correcto. En caso de que se produzca un fallo, el sistema avisa al usuario con una pantalla de error.
Salidas	Mensaje de eliminación satisfactoria de la traducción o de fallo en caso de error.

Numero	Consultar Traducciones
Introducción	El administrador puede consultar todas las traducciones existentes.
Entradas	Idioma y código de la traducción.
Proceso	El administrador realiza la acción Consultar Traducciones. El sistema entonces muestra un listado con todas las traducciones existentes. El administrador puede seleccionar cualquiera de ellas y visualizar su texto en el idioma que desee.
Salidas	Texto de la traducción en el idioma que se ha seleccionado.

Numero	Añadir Servicios
Introducción	El administrador puede añadir servicios nuevos para que los arrendadores puedan añadirlos a los espacios.
Entradas	Nombre y descripción del servicio.
Proceso	El sistema muestra un formulario con los campos: nombre del servicio y descripción del servicio. El administrador rellena esos campos y envía el formulario al servidor. El servidor valida entonces los datos y envía una pantalla de confirmación o de fallo en caso de error. En caso de que los datos sean válidos se añade el servicio a la aplicación.
Salidas	Mensaje de creación satisfactoria del servicio o de fallo en caso de error.

Numero	Eliminar Servicios
Introducción	El administrador puede eliminar servicios que ofertan los espacios.
Entradas	Identificador del servicio.
Proceso	El sistema muestra un listado con todos los servicios existentes. El usuario puede seleccionar uno o buscarlo. Cuando lo selecciona puede borrarlo haciendo uso de un botón de borrado. El sistema borra entonces el servicio seleccionado.
Salidas	Mensaje de eliminación satisfactoria del servicio o de fallo en caso de error.

Numero	Consultar Servicios
Introducción	El administrador puede visualizar la descripción de los servicios existentes en la aplicación.
Entradas	Identificador del servicio.
Proceso	El sistema muestra un listado con todos los servicios existentes. El usuario puede consultar cualquiera de ellos seleccionándolo o buscando por su identificador. El sistema muestra los detalles del servicio.
Salidas	Listado con el nombre y descripción de los distintos servicios existentes.

Numero	Notificar Usuario de Spam
Introducción	El administrador puede notificar mediante un mensaje a un usuario que haya infringido los términos de uso.
Entradas	Nombre de usuario a notificar y mensaje.
Proceso	El administrador introduce el nombre de usuario al que quiere enviar la notificación y el mensaje que se enviará. El sistema envía una notificación de Spam al usuario y el mensaje que ha introducido el administrador.
Salidas	Mensaje de notificación satisfactoria o de fallo en caso de error.

Numero	Marcar Anuncio como Spam
Introducción	El administrador puede poner, manualmente, una marca de Spam a un anuncio de un espacio. El anuncio quedará invisible en la aplicación.
Entradas	Identificador del anuncio.
Proceso	El administrador introduce la el identificador del anuncio a marcar y selecciona que lo quiere marcar como Spam. El sistema se encarga de que el anuncio marcado no esté visible en la aplicación.
Salidas	Ventana con un mensaje de que el anuncio ha sido marcado como Spam de forma satisfactoria o con un mensaje de fallo en caso de error.

Numero	Bloquear Usuario
Introducción	El administrador puede bloquear la cuenta de un usuario que haya infringido los términos de uso.
Entradas	Nombre del usuario a bloquear y mensaje.
Proceso	El administrador introduce el nombre de usuario que quiere bloquear y el mensaje que se enviará. El sistema envía una notificación de Bloqueo al usuario y el mensaje que ha introducido el administrador. El usuario queda bloqueado en el sistema y no puede acceder a su cuenta.
Salidas	Ventana con un mensaje de que el usuario ha sido bloqueado de forma satisfactoria o con un mensaje de fallo en caso de error.

Numero	Desbloquear Usuario
Introducción	El administrador puede desbloquear la cuenta de un usuario que previamente ha sido bloqueada si el usuario se compromete a cumplir los términos de uso.
Entradas	Nombre del usuario a desbloquear.
Proceso	El administrador introduce el nombre de usuario que quiere desbloquear. El sistema desbloquea al usuario y envía un mensaje al usuario diciendo que debe cumplir con los términos de la aplicación.
Salidas	Ventana con un mensaje de que el usuario ha sido desbloqueado de forma satisfactoria o con un mensaje de fallo en caso de error.

Numero	Añadir Categorías de Espacios
Introducción	El administrador puede añadir nuevas tipos de habitación y tipos de propiedades de espacios a la aplicación.
Entradas	Nombre, descripción y tipo de la categoría de espacios.
Proceso	El sistema muestra un formulario con los campos: nombre, descripción y tipo de la categoría. El administrador rellena esos campos y envía el formulario al servidor. El servidor valida entonces los datos y envía una pantalla de confirmación o de fallo en caso de error. En caso de que los datos sean válidos se añade la categoría a la aplicación.
Salidas	Mensaje de creación satisfactoria de una nueva categoría de fallo en caso de error.

Numero	Eliminar Categorías de Espacios
Introducción	El administrador puede eliminar ciertos tipos de habitación y de propiedad que no sean usados o ya no sean necesarios.
Entradas	Tipo e identificador de la categoría de espacios.
Proceso	El administrador selecciona un tipo de categoría de espacios: tipo de habitación o tipo de propiedad. Entonces se muestra el listado con las categorías de espacios de ese tipo. El administrador selecciona una de ellas o la busca por su identificador. Entonces puede borrar la categoría haciendo uso de un botón borrar. La sistema se encarga de borrar la categoría de la aplicación.
Salidas	Mensaje de eliminación satisfactoria de una nueva categoría de fallo en caso de error.

Numero	Consultar Categorías de Espacios
Introducción	El administrador puede consultar la descripción de los tipos de habitación y propiedad existentes en la aplicación.
Entrada	Tipo e identificador de la categoría de espacios.
Proceso	El administrador selecciona un tipo de categoría de espacios: tipo de habitación o tipo de propiedad. Entonces se muestra el listado con las categorías de espacios de ese tipo. El administrador selecciona una de ellas o la busca por su identificador. Entonces se muestran los detalles de esa categoría: nombre y descripción.
Salidas	Listado con el nombre y descripción de los Tipos de Habitación y de Propiedad que aparecen en el sistema.

Numero	Consultar y Editar FAQ
Introducción	El administrador puede consultar y editar todas las preguntas frecuentes existentes y la respuesta de cada una de ellas.
Entradas	Identificador de la pregunta frecuente.
Proceso	El sistema muestra el listado con todas las preguntas frecuentes. El administrador selecciona una de ellas o la busca por su identificador. Entonces se muestra la respuesta a esa pregunta y el administrador puede editarla.
Salidas	Listado con las distintas preguntas frecuentes o en caso de la edición, un mensaje de edición satisfactoria de una pregunta frecuente o de fallo en caso de error.

Numero	Consultar y Editar Datos de la aplicación
Introducción	El administrador puede consultar y editar los datos propios de la aplicación como datos que ayudan a los usuario a saber cómo funciona la aplicación y a saber quien hay detrás de ella.
Entradas	Datos de la aplicación a editar.
Proceso	El sistema muestra un formulario con todas los datos propios de la aplicación. El administrador puede editar cada uno de ellos o simplemente visualizarlos.
Salidas	Listado con los distintos datos o en caso de la edición, un mensaje de edición satisfactoria de un dato concreto de la aplicación o de fallo en caso de error.

Numero	Consultar y Editar Contacto de la Aplicación
Introducción	El administrador puede consultar y editar los datos de contacto de la aplicación: dirección, correo electrónico y teléfono.
Entradas	Dirección, correo electrónico y teléfono de la aplicación.
Proceso	El sistema muestra un formulario con los datos de contacto: dirección, correo electrónico y teléfono. El administrador puede editar cada uno de ellos o simplemente visualizarlos.
Salidas	Listado con los distintos datos de contacto de la aplicación o en caso de la edición, un mensaje de edición satisfactoria o de fallo en caso de error.

Numero	Consultar y Editar Términos y Condiciones
Introducción	El administrador puede consultar y editar los términos y condiciones de la aplicación.
Entradas	Texto de los Términos y condiciones.
Proceso	El sistema muestra los términos y condiciones actuales. El administrador puede consultar estos términos actuales y cambiarlos por otros nuevos.
Salidas	Texto con los términos y condiciones de la página web o en caso de la edición, un mensaje de edición satisfactoria de pregunta frecuente o de fallo en caso de error.

Usuario No Registrado

Numero	Visualizar Detalles del Espacio
Introducción	Cualquier usuario no registrado puede ver los detalles de un espacio concreto.
Entradas	Espacio a visualizar.
Proceso	El sistema muestra un listado de los espacios. Cualquier usuario puede seleccionar un anuncio de la lista y visualizar los detalles pertenecientes al espacio. Puede visualizar la vista principal, el calendario, los servicios, los detalles, el mapa y los comentarios pertenecientes al espacio.
Salidas	Listado con los distintos datos del espacio.

Numero	Comparar Espacios
Introducción	Se pueden comparar varios espacios, así como un mismo espacio en distintas fechas.
Entradas	Espacios a comparar.
Proceso	El usuario selecciona varios espacios para compararlos. Se visualiza entonces para cada uno de ellos la fecha de entrada y de salida, los días, la valoración y el precio.
Salidas	Listado de los distintos espacios elegidos para realizar la comparación. De cada espacio se visualiza la puntuación, el precio, las fechas de la reserva y el número de días que se quiere reservar.

Numero	Consultar página de FAQ
Introducción	Un usuario no registrado puede consultar la página de preguntas más frecuentes.
Entradas	Acción de ir a la página de FAQ.
Proceso	El usuario selecciona la página de FAQ desde la página principal de la aplicación. El sistema muestra entonces las distintas preguntas frecuentes y la respuesta para cada una de ellas.
Salidas	Listado de preguntas más frecuentes con su respuesta correspondiente.

Numero	Consultar página de Contacto
Introducción	Un usuario no registrado puede consultar la página contacto de la aplicación.
Entradas	Acción de ir a la página de contacto.
Proceso	El usuario selecciona la página de Contacto desde la página principal de la aplicación. El sistema muestra entonces los datos de contacto y un formulario mediante el cual el usuario puede enviar un correo.
Salidas	Datos de contacto de la aplicación: nombre, email, dirección, teléfonos, horario de atención al cliente y formulario para enviar un correo.

Numero	Consultar página de Ayuda
Introducción	Un usuario no registrado puede consultar la página de ayuda en la que existen datos de cómo funciona la aplicación.
Entradas	Acción de ir a la página de ayuda.
Proceso	El usuario selecciona la página de Ayuda desde la página principal de la aplicación. El sistema muestra entonces consejos y funcionalidades de la aplicación.
Salidas	Listado de consejos y funcionalidades explicadas paso a paso.

Numero	Buscar Espacio
Introducción	Cualquier usuario de la aplicación puede buscar un espacio de manera muy sencilla.
Entradas	Localidad, fecha de llegada, fecha de salida y personas.
Proceso	El sistema muestra una barra de búsqueda para buscar espacios. El usuario introduce la localidad, la fecha de llegada y de salida y el número de personas. El sistema muestra entonces un listado con los espacios que se adaptan a los criterios de búsqueda introducido.
Salidas	Listado de los espacios que cumplen los criterios de búsqueda.

Numero	Modificar Idioma
Introducción	El usuario puede modificar el idioma de la aplicación entre los existentes.
Entradas	Idioma de la aplicación.
Proceso	El usuario selecciona uno de los idiomas entre los existentes de la aplicación. Toda la aplicación entonces se muestra en ese idioma seleccionado.
Salidas	La página web aparece en el idioma seleccionado.

Numero	Consultar Página de Privacidad y Condiciones
Introducción	Un usuario no registrado puede consultar la página de privacidad y condiciones de la aplicación.
Entradas	Acción de ir a la página de privacidad y condiciones
Proceso	El usuario selecciona la página de Privacidad y Condiciones desde la página principal de la aplicación. El sistema muestra los términos y condiciones actuales.
Salidas	Texto con los términos y condiciones de la página web.

Numero	Registro de usuario
Introducción	El usuario se puede registrar en la aplicación introduciendo un correo electrónico, un nombre de usuario y una contraseña.
Entradas	Nombre de usuario, correo electrónico y contraseña.
Proceso	El sistema muestra un formulario de registro de usuario con los campos: nombre de usuario, correo electrónico y contraseña. El usuario introduce los datos que desee para crear su usuario. El sistema muestra si el usuario ya está registrado en la aplicación o si la contraseña es incorrecta. Si los datos son correctos el sistema registra al usuario en la aplicación, envía un correo de confirmación y autentifica al usuario en la aplicación.
Salidas	Mensaje de que el usuario se ha registrado correctamente o de fallo en caso de error.

Usuario Registrado

Numero	Login de Usuario
Introducción	El usuario registrado podrá autenticarse en la aplicación introduciendo su nombre de usuario y contraseña.
Entradas	Nombre de usuario o correo electrónico y contraseña. O acción de conectarse con Facebook en caso de login con Facebook.
Proceso	El sistema muestra un formulario con el login de usuario. El usuario primero elige autenticarse mediante Facebook o mediante la propia aplicación. Si elige Facebook entonces el sistema externo de Facebook se encarga de validar y autenticar al usuario. En caso de que el usuario elija autenticarse mediante la aplicación, debe introducir entonces su nombre de usuario o correo electrónico y la contraseña. El sistema valida los datos y autentifica al usuario si los datos son válidos.
Salidas	Página de usuario si el login de usuario se ha realizado correctamente. En caso de de fallo, mensaje de error y página de login.

Numero	Cerrar Sesión
Introducción	El usuario registrado puede cerrar la sesión de usuario en todo momento.
Entradas	Acción de cerrar sesión.
Proceso	El usuario autenticado anteriormente puede cerrar sesión. Entonces el sistema redirige al usuario a la página principal. El usuario ya no está entonces autenticado en la aplicación.
Salidas	Página principal de la aplicación.

Numero	Visualizar Mensajes Recibidos y Enviados
Introducción	El usuario registrado puede visualizar los mensajes que ha recibido de otros usuarios, así como los que él ha enviado.
Entradas	Acción de visualizar mensajes.
Proceso	El usuario registrado va a la página de mensajes desde su página personal. El sistema muestra un listado con todos los mensajes recibidos y otro listado con todos los mensajes enviados por el usuario.
Salidas	Listado de mensajes enviados y recibidos por el usuario.

Numero	Enviar Mensajes
Introducción	El usuario registrado puede enviar los mensajes a otros usuarios de la aplicación.
Entradas	Nombre de usuario, asunto y texto del mensaje.
Proceso	El sistema muestra un formulario de envío de mensajes con los campos nombre de usuario, asunto y texto del mensaje. El usuario registrado rellena estos campos y envía el mensaje. El sistema se encarga del enviar el mensaje al destinatario correspondiente y el mensaje pasa al listado de mensajes enviados.
Salidas	Mensaje de envío correcto o mensaje de error en caso de fallo.

Numero	Eliminar Mensajes
Introducción	El usuario registrado puede eliminar mensajes recibidos o enviados de su buzón de entrada. Estos mensajes ya no se visualizarán.
Entradas	Mensaje a eliminar.
Proceso	El sistema muestra el listado con todas las preguntas frecuentes. El administrador selecciona una de ellas o la busca por su identificador. Entonces se muestra la respuesta a esa pregunta y el administrador puede editarla.
Salidas	Mensaje de que se ha eliminado correctamente el mensaje o fallo en caso de error.

Numero	Consultar y Editar Datos del Perfil
Introducción	El usuario registrado puede consultar y editar sus datos de perfil.
Entradas	Datos de perfil de usuario.
Proceso	El sistema muestra un formulario con todos los datos de perfil de usuario. El usuario registrado puede editar cualquier campo del formulario.
Salidas	Datos de perfil de usuario o en caso de la edición, un mensaje de edición satisfactoria del campo o de fallo en caso de error.

Numero	Cambiar Foto del Perfil
Introducción	El usuario registrado puede cambiar la foto de perfil visible para otros usuario de la aplicación.
Entradas	Foto de Perfil de Usuario.
Proceso	El sistema muestra un formulario para cambiar la foto de perfil. El usuario registrado puede subir una foto nueva o modificar la anterior. La nueva foto que guardada en la aplicación.
Salidas	Mensaje de que la foto se ha cambiado correctamente o de fallo en caso de error.

Numero	Añadir un Espacio a Favoritos
Introducción	El usuario registrado puede añadir un espacio como favorito.
Entradas	Espacio para añadir a favoritos.
Proceso	El usuario registrado selecciona un espacio y lo marca como favorito. Entonces el sistema lo añade a espacios favoritos del usuario y éste puede visualizarlo posteriormente.
Salidas	Mensaje avisando de que el espacio se ha añadido correctamente a favoritos o mensaje de fallo en caso de error.

Usuario Arrendador

Numero	Consultar Listado de Reservas
Introducción	El usuario arrendador puede consultar las reservas que otros usuarios han hecho a sus espacios.
Entradas	Acción de consultar reservas actuales.
Proceso	El usuario arrendador va a la página de reservas actuales desde su página de usuario. El sistema muestra un listado con todas las reservas vigentes que otros usuarios han hecho a sus espacios.
Salidas	Listado de reservas actuales que otros usuarios han hecho a sus espacios.

Numero	Confirmar Reserva
Introducción	El arrendador puede confirmar reservas que otros usuarios han hecho a sus espacios. Esta confirmación sirve para asegurar la disponibilidad del espacio.
Entradas	Reserva actual no confirmada ni cancelada.
Proceso	El sistema muestra el listado de reservas pendientes de confirmación. El arrendador entonces selecciona aquellas que quiere confirmar y realiza la acción de confirmación. El sistema envía un mensaje al usuario arrendatario que ha realizado la reserva y marca la reserva como confirmada.
Salidas	Mensaje de reserva confirmada correctamente o de fallo en caso de error.

Numero	Cancelar Reserva
Introducción	El arrendador puede cancelar reservas que otros usuarios han hecho a sus espacios. Esta cancelación tiene el significado de que el espacio no está disponible.
Entradas	Reserva actual no confirmada ni cancelada.
Proceso	El sistema muestra el listado de reservas pendientes de confirmación o cancelación. El arrendador entonces selecciona aquellas que quiere cancelar y realiza la acción de cancelación. El sistema envía un mensaje al usuario arrendatario que ha realizado la reserva y marca la reserva como cancelada.
Salidas	Mensaje de reserva cancelada correctamente o de fallo en caso de error.

Numero	Visualizar Histórico de Reservas
Introducción	El usuario arrendador puede visualizar el histórico de reservas realizadas a sus espacios. Puede visualizar las reservas canceladas, confirmadas y pendientes.
Entradas	Acción de consultar histórico de reservas.
Proceso	El usuario arrendador va a la página de reservas antiguas desde su página de usuario. El sistema muestra un listado con todas las reservas antiguas que otros usuarios han hecho a sus espacios.
Salidas	Listado de reservas antiguas que otros usuarios han hecho a sus espacios.

Numero	Consultar Listado de Anuncios
Introducción	El arrendador puede consultar la lista de anuncios publicados de sus espacios.
Entradas	Acción de consultar página de anuncios.
Proceso	El usuario arrendador va a la página de anuncios de espacios desde su página de usuario. El sistema muestra un listado con todos los anuncios de espacios que ha publicado el usuario arrendador.
Salidas	Listado de anuncios que ha creado el arrendador.

Numero	Marcar Anuncio como Visible/No Visible
Introducción	El arrendador tiene la posibilidad de cambiar la visibilidad de un anuncio realizado por él. Puede elegir entre visible en la aplicación y solo visible en su espacio privado.
Entradas	Anuncio para marcar como visible/no visible.
Proceso	El arrendador selecciona un anuncio a marcar y produce la acción de marcar como Visible/No Visible. El sistema se encarga de que el anuncio marcado esté visible/no esté visible en la aplicación.
Salidas	El anuncio no puede ser visualizado por otros usuarios si ha sido marcado como No Visible. Si ha sido marcado como Visible cualquier usuario puede visualizarlo.

Numero	Visualizar y Editar Detalles de un Anuncio
Introducción	El arrendador puede visualizar y editar los detalles de un anuncio publicado anteriormente.
Entradas	Anuncio para visualizar o editar.
Proceso	El sistema muestra el listado con todas los anuncios que tiene el usuario arrendador. El arrendador selecciona uno de ellos. Entonces se muestra los detalles de ese anuncio y el arrendador puede editarlo.
Salidas	Datos de un anuncio o en caso de la edición, un mensaje de edición satisfactoria del campo o de fallo en caso de error.

Numero	Mostrar Vista Previa de un Anuncio
Introducción	El arrendador puede ver una vista previa del anuncio creado para saber como va a quedar finalmente.
Entradas	Anuncio para mostrar vista previa.
Proceso	El sistema muestra el listado con todas los anuncios que tiene el usuario arrendador. El arrendador selecciona uno de ellos y elige la opción de visualizar la vista previa del anuncio.
Salidas	Se visualiza el anuncio como va a quedar en la aplicación.

Numero	Publicar un Anuncio
Introducción	El administrador puede publicar un anuncio en la aplicación en un solo paso.
Entradas	Localidad, dirección, capacidad y tipo de espacio del anuncio.
Proceso	El sistema muestra un formulario en el que el arrendador debe introducir los siguientes datos básicos del espacio: la localidad, dirección, capacidad y tipo de espacio, que puede ser compartido o no compartido. Una vez introducidos estos datos, el arrendador envía el formulario al servidor. El sistema crea el nuevo anuncio, si los datos introducidos son correctos, y muestra al arrendador los detalles del anuncio creado.
Salidas	Mensaje de creación satisfactoria de un nuevo anuncio o de fallo en caso de error.

Usuario Arrendatario

Numero	Consultar Reservas realizadas
Introducción	El usuario arrendatario puede consultar las reservas que ha realizado y el estado de éstas.
Entradas	Acción de consultar reservas realizadas.
Proceso	El usuario arrendatario va a la página de reservas realizadas desde su página de usuario. El sistema muestra un listado con todas las reservas vigentes que él ha realizado.
Salidas	Listado de reservas realizadas por el arrendatario.

Numero	Cancelar Reserva realizada
Introducción	El usuario arrendatario puede cancelar una reserva realizada anteriormente y que aún no haya sido confirmada por el arrendador.
Entradas	Reserva que ha realizado el usuario no confirmada ni cancelada.
Proceso	El sistema muestra el listado de reservas que ha realizado el usuario arrendatario y que están pendientes de confirmación o cancelación por el arrendador. El arrendatario entonces selecciona aquellas que quiere cancelar y realiza la acción de cancelación. El sistema envía un mensaje al usuario arrendador que ha creado el anuncio y marca la reserva como cancelada por el arrendatario.
Salidas	Mensaje de reserva cancelada correctamente o de fallo en caso de error.

Numero	Comentar y Valorar un Espacio Visitado
Introducción	El usuario arrendatario puede hacer un comentario y una valoración de un espacio que ha visitado.
Entradas	Espacio visitado, texto del comentario y valoración.
Proceso	El usuario arrendatario selecciona el anuncio de un espacio que ha visitado. Entonces el sistema muestra un formulario para rellenar con la valoración y comentario del anuncio. El arrendatario lo rellena y lo envía para realizar la valoración del espacio visitado. Estos datos los utiliza el sistema para la valoración global del anuncio.
Salidas	Mensaje de valoración correcta o de fallo en caso de error.

Numero	Recomendar un Espacio
Introducción	El usuario arrendatario puede compartir un espacio en las redes sociales y recomendárselo a otros amigos.
Entradas	Espacio para recomendar.
Proceso	El usuario arrendatario selecciona un espacio y realiza la acción de recomendarlo mediante Facebook. El sistema se encarga de enviar los datos del espacio a la entidad externa Facebook. Este espacio es visualizado por los amigos que tiene el usuario arrendatario en Facebook.
Salidas	Mensaje de que el espacio se ha compartido correctamente o de fallo en caso de error.

Numero	Reservar un Espacio
Introducción	El usuario arrendatario puede hacer una reserva de un espacio.
Entradas	Espacio para reservar, huéspedes, fecha de entrada y fecha de salida.
Proceso	El usuario arrendatario selecciona un espacio para reservar e introduce los huéspedes, la fecha de entrada y la fecha de salida. El sistema envía entonces un mensaje al arrendador que ha publicado el anuncio y crea la reserva.
Salidas	Mensaje de reserva satisfactoria o de fallo en caso de error.

Usuario que comparte espacios

Numero	Crear un Anuncio Compartido
Introducción	El usuario que comparte espacios puede crear un anuncio de un espacio para compartirlo con otros usuarios y alquilarlo conjuntamente.
Entradas	Capacidad, dirección y localidad del espacio para compartir.
Proceso	El sistema muestra un formulario en el que el usuario que comparte espacios debe introducir los siguientes datos básicos del espacio: localidad, dirección, capacidad. Una vez introducidos estos datos, el usuario envía el formulario al servidor. El sistema crea el nuevo anuncio compartido, si los datos introducidos son correctos, y muestra al usuario que comparte espacios los detalles del anuncio creado.
Salidas	Mensaje creación correcta del anuncio o de fallo en caso de error.

Numero	Agregarse a Anuncio Compartido
Introducción	El usuario que comparte espacios puede incorporarse a un anuncio compartido. Este anuncio ha sido antes creado por otro usuario.
Entradas	Anuncio compartido.
Proceso	El usuario que comparte espacio selecciona un anuncio compartido y realiza la acción de agregarse a este anuncio. Entonces el sistema agrega al usuario al anuncio y envía un mensaje a todos los usuarios compartidos que formaran parte del anuncio y los pone en contacto facilitándoles sus datos de contacto.
Salidas	Mensaje de que el usuario ha sido añadido correctamente al anuncio o de fallo en caso de error.

Numero	Alquilar un Espacio Compartido
Introducción	El usuario que comparte espacios puede alquilar de manera conjunta un espacio.
Entradas	Acción de alquilar de manera conjunta un espacio.
Proceso	Cuando los usuarios que comparten un determinado espacio se ponen de acuerdo, realizan la acción de alquilar el espacio compartido. El sistema envía un mensaje al usuario arrendador que ha publicado el espacio y crea una reserva.
Salidas	Mensaje de alquiler satisfactorio del espacio o de fallo en caso de error.

Numero	Visualizar Anuncios Compartidos
Introducción	El usuario que comparte espacios puede visualizar el listado de los anuncios compartidos existentes.
Entradas	Acción de visualizar anuncios compartidos.
Proceso	El usuario que comparte espacios va a la página de visualizar anuncios compartidos desde la página principal de la aplicación. El sistema muestra el listado de anuncios compartidos existentes en la aplicación.
Salidas	Listado de los anuncios compartidos presentes en la aplicación.

2.3.3. Requisitos de rendimiento

En esta subsección se detallan los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Para obtener unos buenos registros de eficiencia la aplicación web deberá ser escalable. Conforme vaya creciendo los usuarios conectados simultáneamente habrá que tomar decisiones como redundar la base de datos o contratar más banda ancha para el servidor dedicado que albergue la web. Por ello la arquitectura empleada en la creación de la aplicación web debe permitir que estos cambios que casi con toda seguridad se irán produciendo en el tiempo no acaben siendo un problema que acabe con la viabilidad del proyecto. Para empezar el desarrollo con un alojamiento web estándar será suficiente, hasta no tener datos que evidencien un cambio del sistema.

Se entiende por alojamiento web estándar al que ofrece un espacio web de 100 GB, un límite de memoria de 60 MB, un tiempo de ejecución de scripts de 30 segundos, un máximo de 15 procesos paralelos y un total de 10 bases de datos con posibilidad de que puedan redundarse.

2.3.4. Restricciones del diseño

Se entiende por restricciones del diseño a todo aquello que restringe las decisiones relativas al diseño de la aplicación web.

Se ha seguido para el desarrollo de esta aplicación web las recomendaciones marcadas por Word Wide Web Consortium abreviado W3C [29] en lo relativo al diseño web y aplicaciones. Estas recomendaciones incluyen los estándares para la construcción de páginas web incluyendo HTML, CSS, SVG, API del dispositivo y otras tecnologías para las aplicaciones web.

2.3.5. Atributos del sistema software

Seguridad

La seguridad es el aspecto más importante de la aplicación web.

La seguridad debe ser un proceso de dos etapas, cuyo objetivo será evitar que un usuario acceda a un recurso al cuál no debería tener acceso. En el primer paso del proceso llamado autenticación, el sistema de seguridad identificará quién es el usuario obligándolo a presentar algún tipo de identificación. En el segundo paso se determinará si el usuario debe tener acceso a un determinado recurso. Esta parte del proceso se llama autorización, y significa que el sistema está comprobando si el usuario tiene suficientes privilegios para realizar una determinada acción.

Fiabilidad

Para garantizar la fiabilidad será requisito que el alojamiento del sitio web esté georedundado. Si por cualquier motivo el elemento del clúster de servidores encargado de almacenar la base de datos o la aplicación web falla, al estar la web redundada en otro clúster el servicio web no se verá interrumpido.

Otro aspecto crítico es la base de datos, que por seguridad habrá que crear un servicio que realice copias de seguridad periódicas y éstas queden almacenadas por un periodo de tiempo suficiente para garantizar los servicios de la aplicación web ante un posible fallo del sistema.

Disponibilidad

La aplicación web estará siempre disponible para que los usuarios puedan utilizarla en cualquier momento. Salvo que se produzca un fallo del sistema.

Mantenibilidad

Para que la aplicación web sea mantenible será necesario reducir la intervención del usuario administrador a lo estrictamente necesario, de modo que la mayor parte de las tareas de mantenimiento sean realizadas por servicios generados para ello o simplemente rediseñar aquellos procesos que requieran mayor intervención del administrador.

Las tareas en las que sea imprescindible la intervención del administrador, deberán ser mantenibles fácilmente mediante un interfaz adecuado para ello, que puede ser suministrado externamente como es el caso de la base de datos. En caso de que no exista interfaz externa, habrá que desarrollarla para poder desempeñar las tareas requeridas.

Portabilidad

La aplicación web debe ser portable a cualquier hosting que admita la tecnología con la que se desarrolle. Al utilizar los principales navegadores web para ejecutar la aplicación, la aplicación será portable a cualquier sistema operativo que soporte la instalación de estos navegadores. En cualquier caso la aplicación debe ser portable a los tres principales sistemas operativos Linux, MacOS y Windows.

Eficiencia

La aplicación web debe ser muy eficiente para poder competir dentro de un sector tan saturado como es el del alquiler de viviendas. Los tiempos de respuesta deben de ser lo suficiente rápidos para que el usuario no tenga sufrir largas esperas. El usuario debe de tener la funcionalidad completa al terminar de cargar cualquier página, antes no debería poder empezar procesos que seguro van a producir algún tipo de error y tendrá que volver a repetir. Para ello habrá que simplificar todos los procesos que se desarrollen, de modo que si un proceso es muy complicado habrá que separarlo en otros procesos más simples. Las respuestas de datos del servidor deberán de contener los suficientes datos para no estar continuamente realizando peticiones, pero a la vez no deberán contener excesivos datos que produzcan tiempos de respuesta no adecuados.

2.3.6. Otros requisitos

El uso de los dispositivos móviles es cada vez más generalizado, por lo que será requisito que la aplicación web se pueda utilizar en ellos en un futuro. Para ello se utilizarán lenguajes y tecnologías que permitan de una manera fácil la adaptación de la aplicación web a éstos.

Capítulo 3

Análisis

3.1. Introducción

Este capítulo corresponde a la fase de análisis del proyecto. Se analiza la aplicación a desarrollar y se describe su estructura y funcionalidad mediante diagramas que permiten establecer el funcionamiento y las características del sistema.

Para llevar a cabo este análisis se ha utilizado UML [19](Unified Modeling Language). UML es el lenguaje de modelado de sistemas software más conocido y utilizado en la actualidad. Se utiliza para especificar o describir métodos o procesos del sistema que se desea modelar. Establece tres tipos de diagrama: diagramas de estructura, de comportamiento y de interacción. En este proyecto se han utilizado un diagrama de cada tipo. Como diagrama de estructura se ha utilizado un diagrama de clases, un diagrama de casos de uso como diagrama de comportamiento y un diagrama de secuencia que hace referencia a los diagramas de interacción.

3.2. Diagramas de casos de uso

En esta sección se analiza la aplicación mediante diagramas de casos de uso que corresponden a los requisitos funcionales especificados en el capítulo de especificación de requisitos sección 2.3.2.

Un diagrama de casos de uso es un diagrama de comportamiento UML. Define una notación gráfica que da una vista general simple de un conjunto de casos de uso. Un caso de uso es una descripción de los pasos o las actividades que deben realizarse para llevar a cabo algún proceso de la aplicación. En el contexto de este proyecto un caso de uso [33] es una secuencia de interacciones que se desarrollan entre el sistema y sus actores en respuesta a un evento que inicia un actor sobre el sistema.

3.2.1. Actores de la aplicación web

Primero se detalla los actores que intervienen en la aplicación web. En la figura 3.1 se detalla el diagrama correspondiente a los actores. Existen cinco actores

que intervienen en el sistema. Los dos primeros son *usuario no registrado* y *administrador*. Además hay una relación de generalización/especialización. Los actores *usuario arrendador*, *usuario arrendatario* y *usuario que comparte espacios* son una especialización del actor más general, *usuario registrado*. Además una persona puede desempeñar varios roles diferentes describe en la sección 2.2.3 donde se hace referencia a los usuarios del sistema especificados.

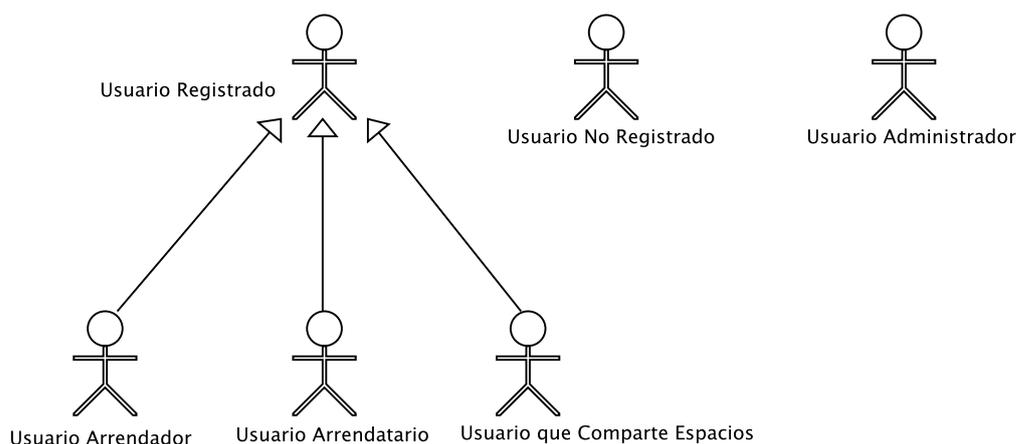


Figura 3.1: Actores.

3.2.2. Administrador

En esta subsección se muestran los diagramas de casos de uso exclusivos del Administrador de la aplicación web. Estos casos de uso están divididos en tres grupos que hacen referencia a características más grandes: *Gestión de los Idiomas*, *Gestión de los Espacios y usuarios* y *Gestión de Datos de la Aplicación*.

Gestión de los Idiomas

El administrador puede gestionar los idiomas de la aplicación. Para poder realizar esta tarea es necesario subdividirla en varios casos de uso más sencillos. Así el administrador puede: *Crear un nuevo idioma*, *Eliminar el idioma*, *Crear una nueva traducción*, *Modificar una traducción*, *Eliminar una traducción* y *Consultar el listado de traducciones existentes*.

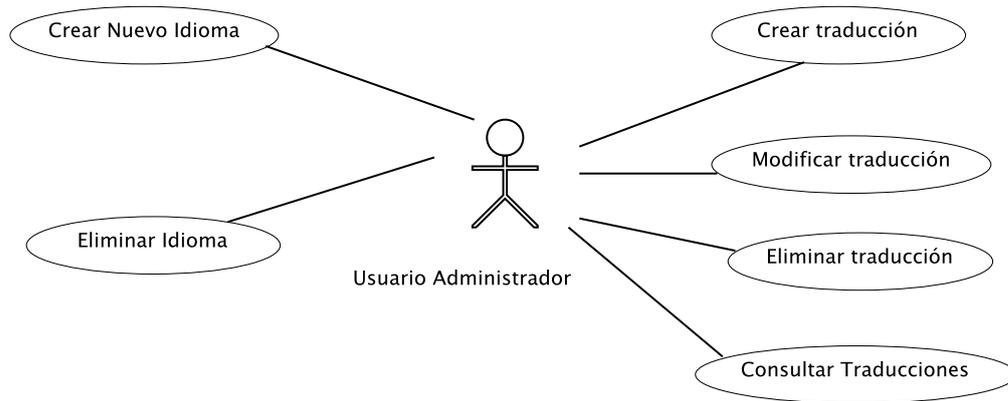


Figura 3.2: Gestión del idioma.

Gestión de los Espacios y usuarios

El siguiente diagrama de casos de uso muestra las funcionalidades relacionadas con la gestión de espacios y usuarios. Se puede apreciar que incluye la gestión de categorías de espacios, la gestión de servicios y la gestión de Spam.

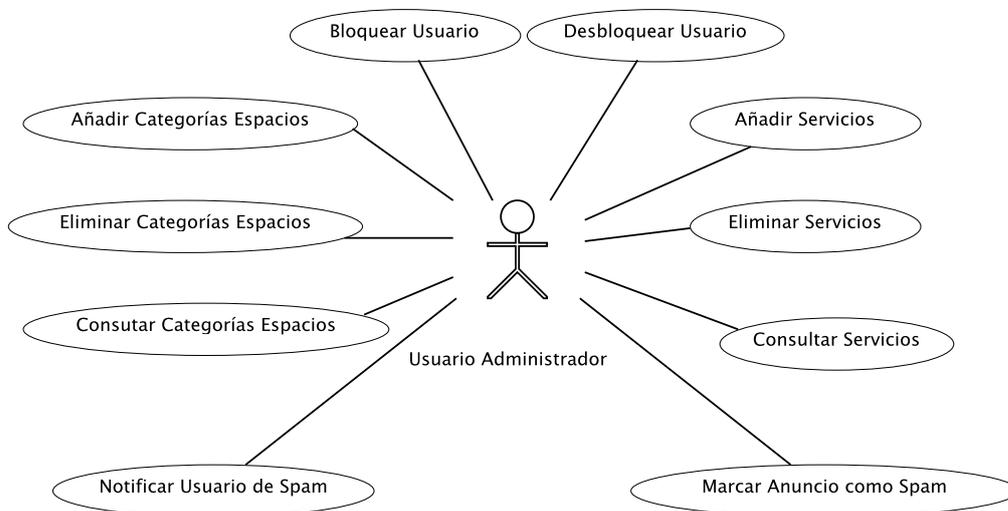


Figura 3.3: Gestión de Espacios y Usuarios.

Gestión de Datos de la Aplicación

A continuación se pueden visualizar las acciones necesarias para que el administrador gestione de una manera correcta los datos propios del sistema. Éste puede

consultar y editar todas las páginas que describen la aplicación, las condiciones de uso y las acciones de ayuda al usuario.

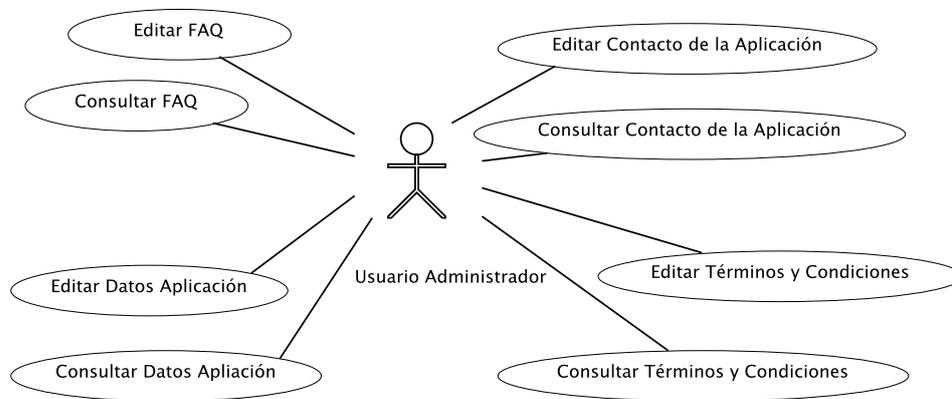


Figura 3.4: Gestión de Datos de la Aplicación.

3.2.3. Usuario No Registrado

En la figura 3.5 se muestran los casos de uso del usuario no registrado, es decir que los puede desempeñar cualquier usuario. Éste puede buscar un espacio, ver los detalles de un espacio, modificar el idioma, consultar las páginas propias de la aplicación, comparar espacios. El caso de uso *Buscar Espacio* se puede observar que tiene relacionado otro mediante una relación *extend*. Esta relación describe que el caso de uso es opcional y se puede utilizar cuando se busca un espacio.

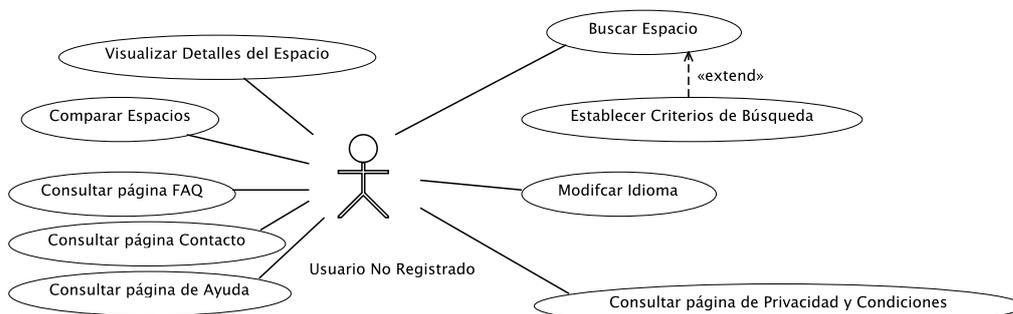


Figura 3.5: Usuario No Registrado.

3.2.4. Usuario Registrado

En el siguiente diagrama se expone la secuencia de interacciones que realiza un usuario registrado en la aplicación. Éste puede autenticarse, consultar y editar sus datos personales, gestionar sus mensajes dentro de la aplicación y añadir espacios a favoritos.

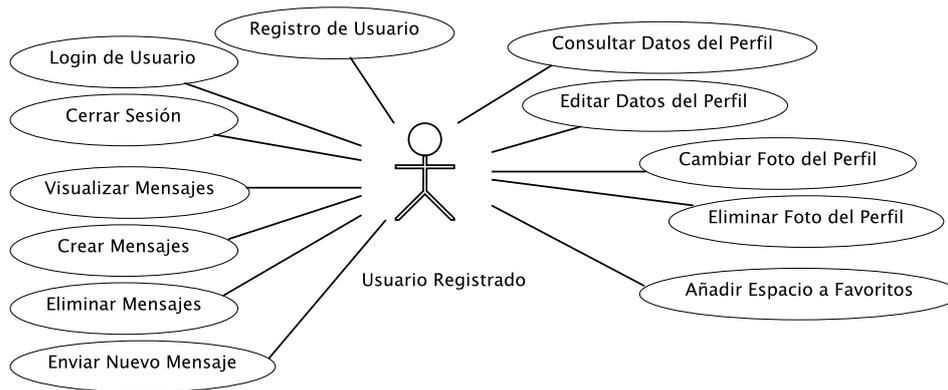


Figura 3.6: Usuario Registrado.

3.2.5. Usuario Arrendador

Los siguientes casos de uso comunican al Usuario Arrendador con el sistema: consultar, confirmar y cancelar Reservas, consultar la lista de espacios que ha publicado, Marcar el anuncio para que se visualice o oculte, ver y modificar los detalles de un anuncio propio y publicar anuncios en la aplicación. Para este último caso de uso primero introduce los datos básicos del anuncio, más tarde rellena los campos adicionales y finalmente publica el espacio.

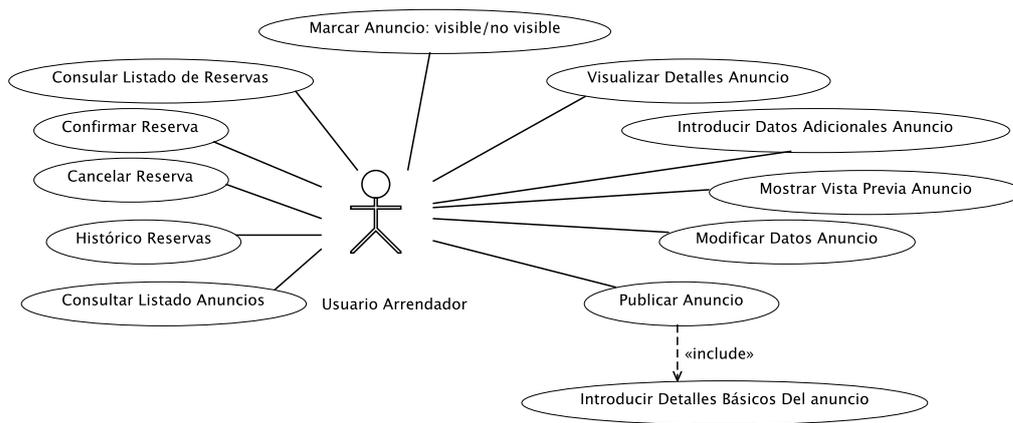


Figura 3.7: Usuario Arrendador.

3.2.6. Usuario Arrendatario

El usuario Arrendatario dispone de las siguientes funcionalidades: consultar reservas y cancelarlas, hacer comentarios y valoración de espacios visitados, recomendar espacios a amigos y reservar espacio. Todas ellas se encuentran representadas en la figura 3.8. Además hay otras dos que se relacionan con el caso de uso *Reservar Espacio: Realizar Pago* y *Enviar Mensaje al Arrendador*. Estos dos casos de uso son opcionales y se realizan en el momento de reservar un espacio. El pago puede no ser realizado desde la aplicación, ya que el arrendador puede elegir otro métodos como la transferencia bancaria.

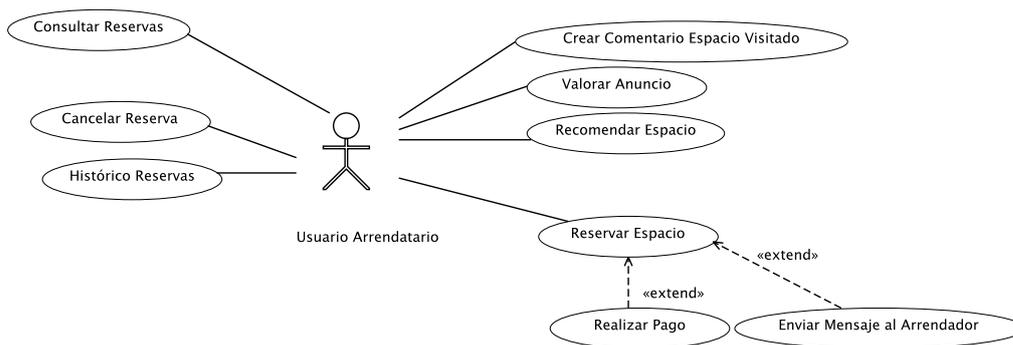


Figura 3.8: Usuario Arrendatario.

3.2.7. Usuario que comparte espacios

Por último el usuario que comparte espacios desempeña las actividades plasmadas en los siguientes casos de uso: Crear un Anuncio Compartido, agregarse a

otros anuncios que han compartido distintos usuarios, alquilar un espacio conjuntamente, y consultar los diferentes espacios compartidos publicados en la aplicación.

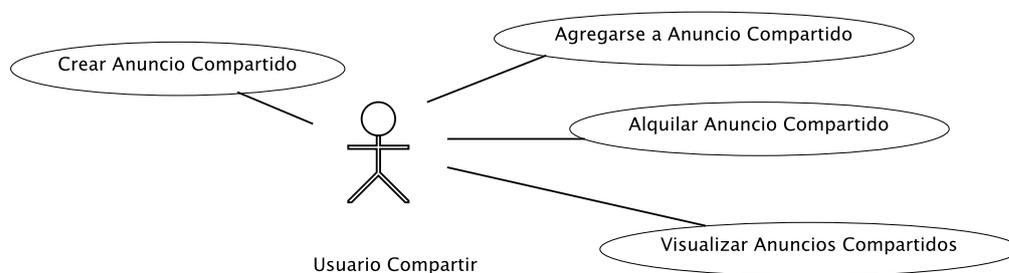


Figura 3.9: Usuario que comparte espacios.

3.3. Diagramas de clases

En esta sección se muestran los diagramas de clases del sistema. Estos diagramas describen la estructura de la aplicación mostrando sus clases y las relaciones entre ellas. Con ellos se crea el diseño conceptual de la información que más tarde se utilizará en el sistema.

Se ha dividido la información que se representa en dos diagramas diferenciados. El primero describe como se estructura la información relacionada con el usuario de la aplicación. Mientras que el segundo está enfocado en el perfil del administrador de la aplicación web.

3.3.1. Usuarios de la aplicación

A continuación se muestra el diagrama de clases (figura 3.10) que representa la información que se crea alrededor de los usuarios, los espacios que publican y las reservas que realizan.

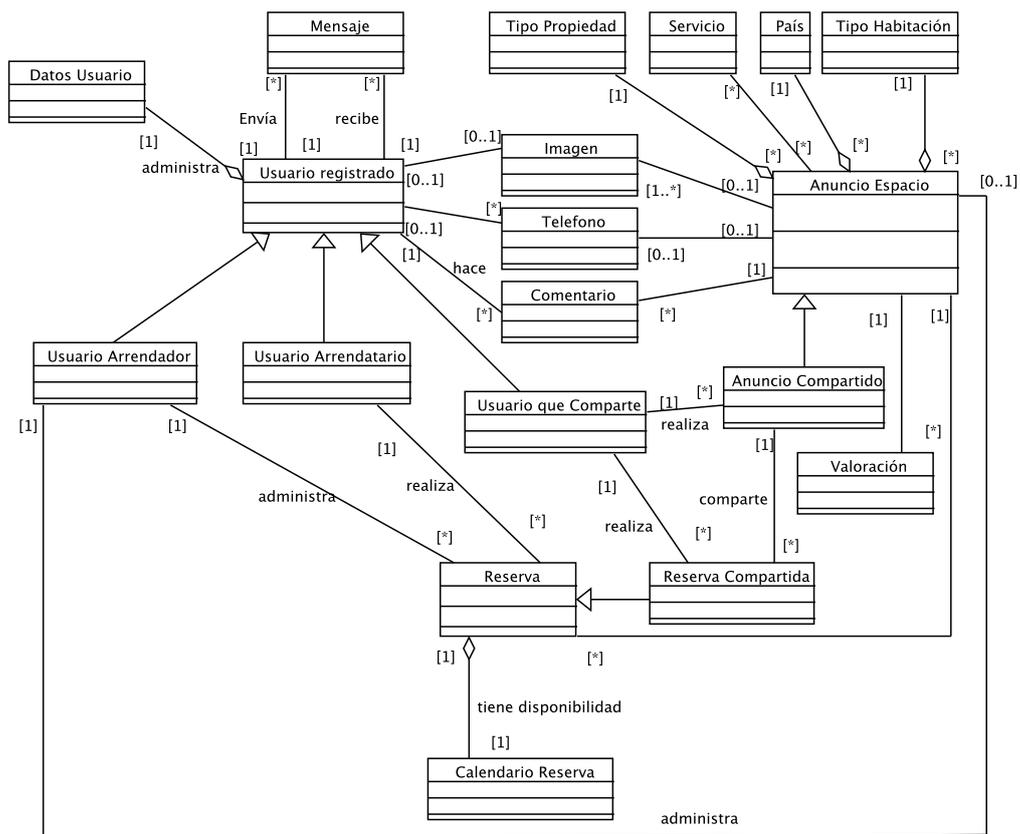


Figura 3.10: Diagrama clases Usuarios.

3.3.2. Administrador

El siguiente diagrama, figura 3.11, describe la estructura de la información relacionada con el Administrador de la aplicación.

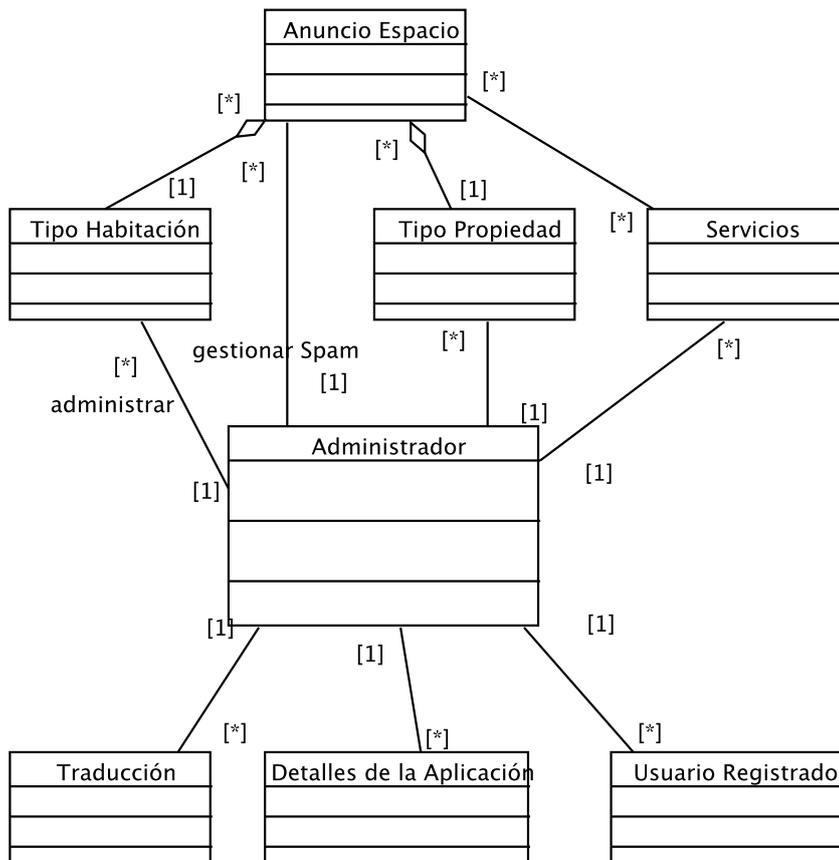


Figura 3.11: Diagrama clases Administrador.

3.4. Diagramas de secuencia

Mediante los diagramas de secuencia se modela la interacción entre los objetos del sistema a través del tiempo. En cada diagrama se han numerado los mensajes para mostrar su orden cronológico. Basándose en el modelo UML se han desarrollado los siguientes diagramas de secuencia.

3.4.1. Autenticación Usuario

En el diagrama de la figura 3.12 se muestra la interacción del usuario con el proceso de autenticación de la aplicación web.

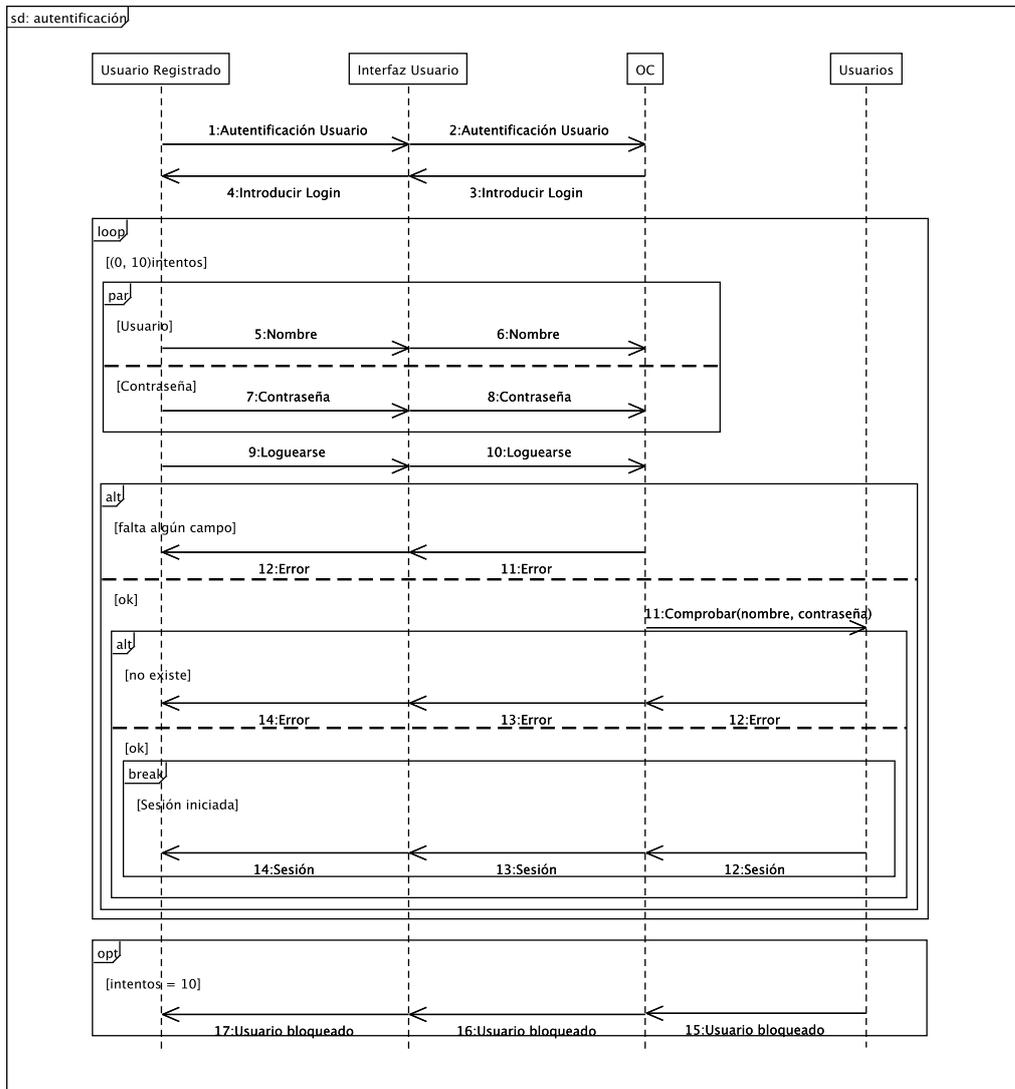


Figura 3.12: Diagrama secuencia Autenticación.

3.4.2. Registro Usuario

En el diagrama de la figura 3.13 se muestra la interacción del usuario con el proceso de registro de la aplicación web.

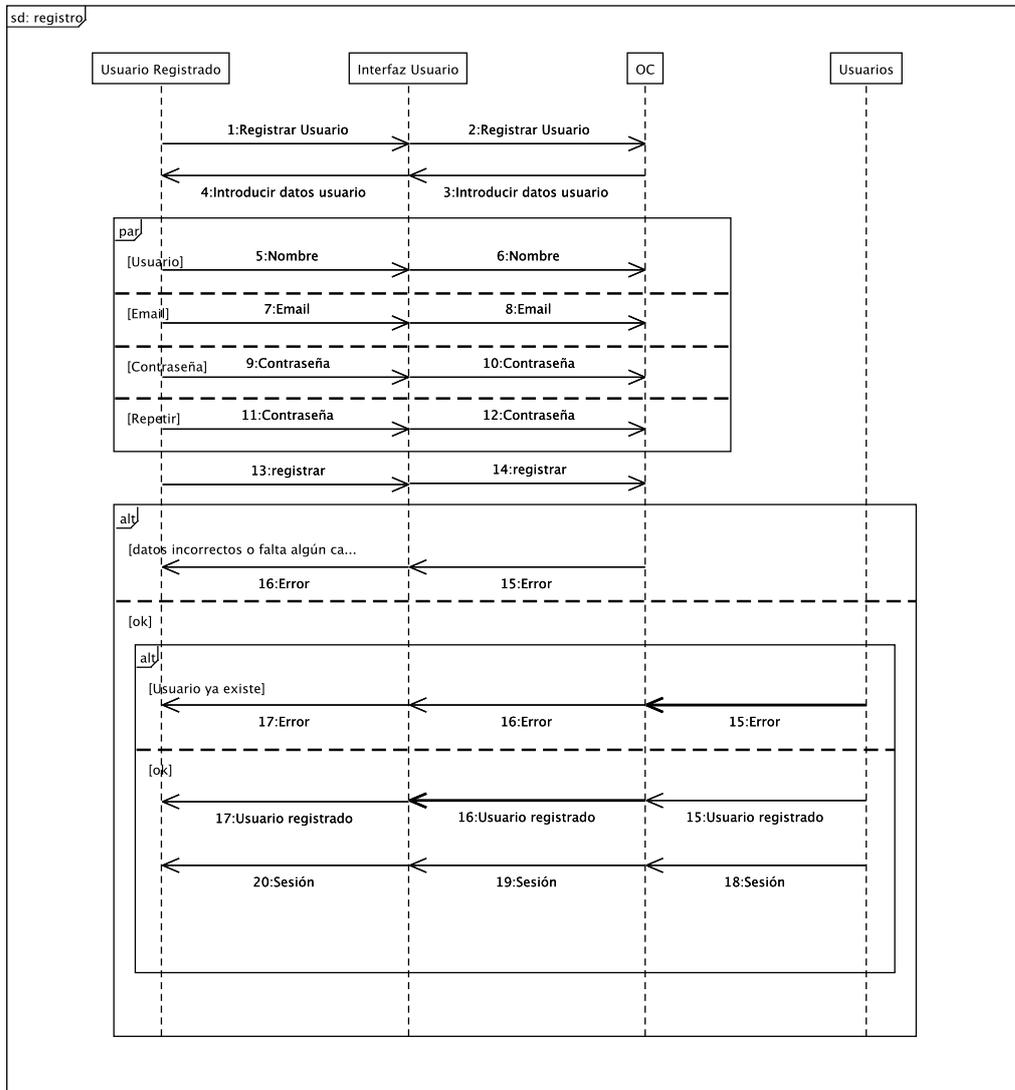


Figura 3.13: Diagrama secuencia Registro Usuario.

3.4.3. Publicar Anuncio

En el diagrama de la figura 3.14 se muestra la interacción del usuario con el proceso de publicar un anuncio de un espacio en la aplicación web.

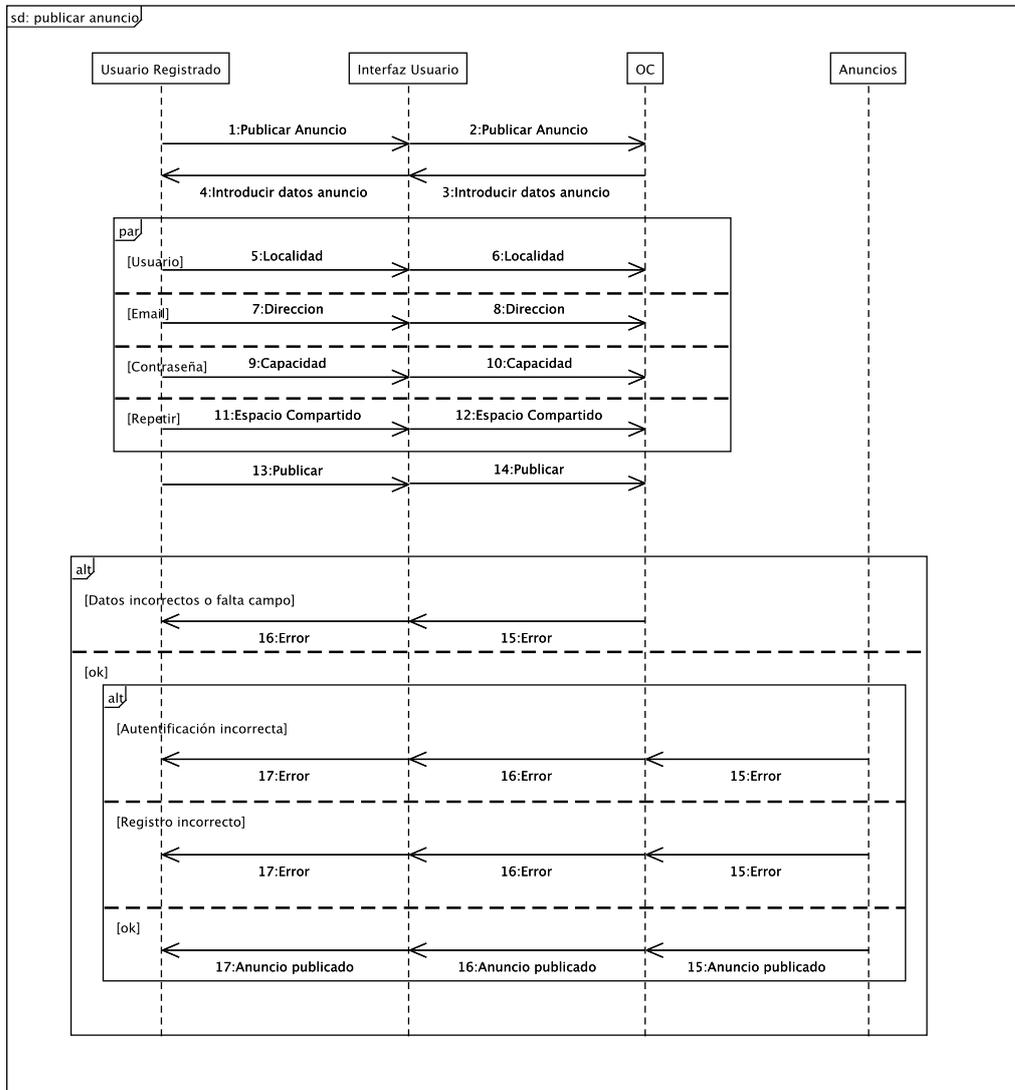


Figura 3.14: Diagrama secuencia Publicar Anuncio.

3.4.4. Buscar y seleccionar espacio para reservar

En el diagrama de la figura 3.15 se muestra la interacción del usuario con el proceso de seleccionar un espacio para posteriormente en otro proceso reservarlo.

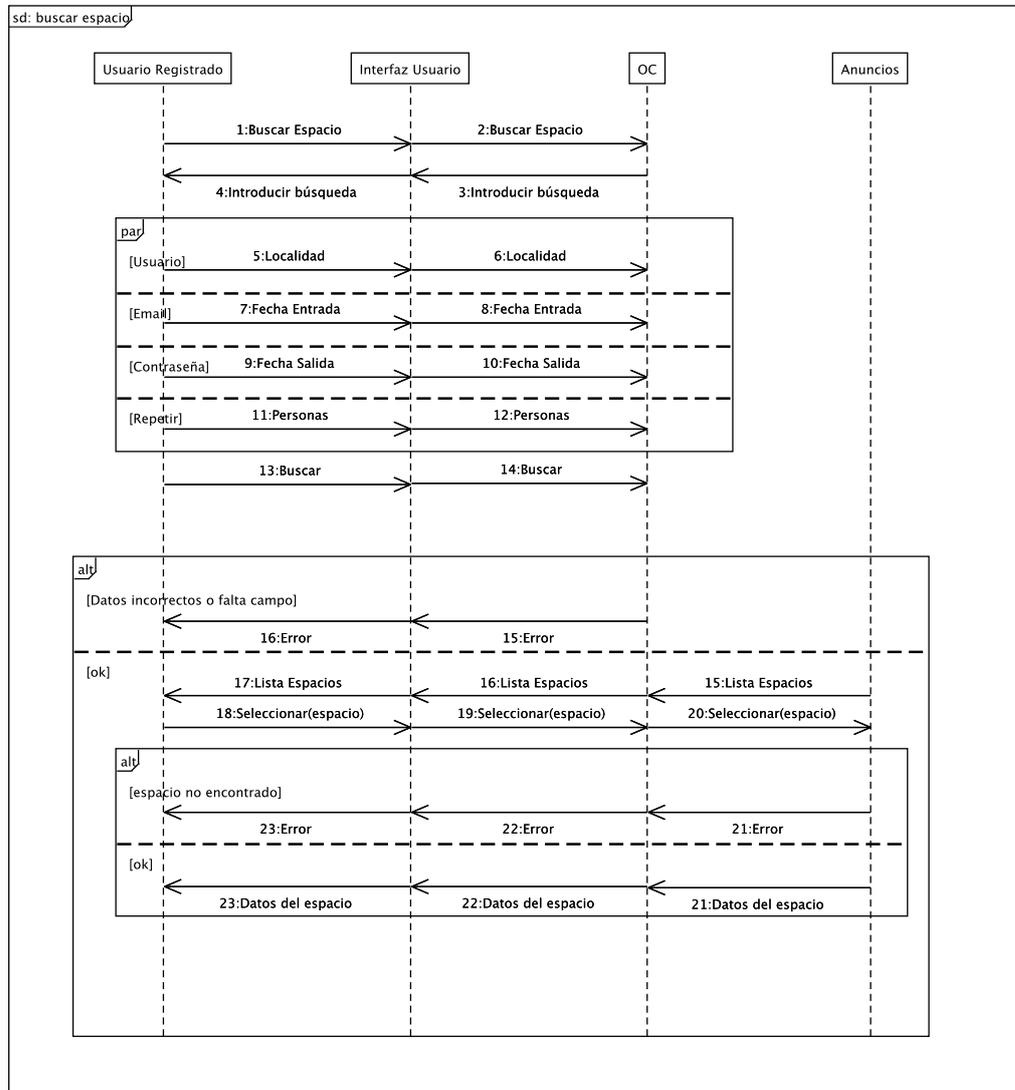


Figura 3.15: Diagrama secuencia Buscar y Seleccionar Espacio.

3.4.5. Comparar espacios

En el diagrama de la figura 3.16 se muestra la interacción del usuario con el proceso de comparar varios espacios.

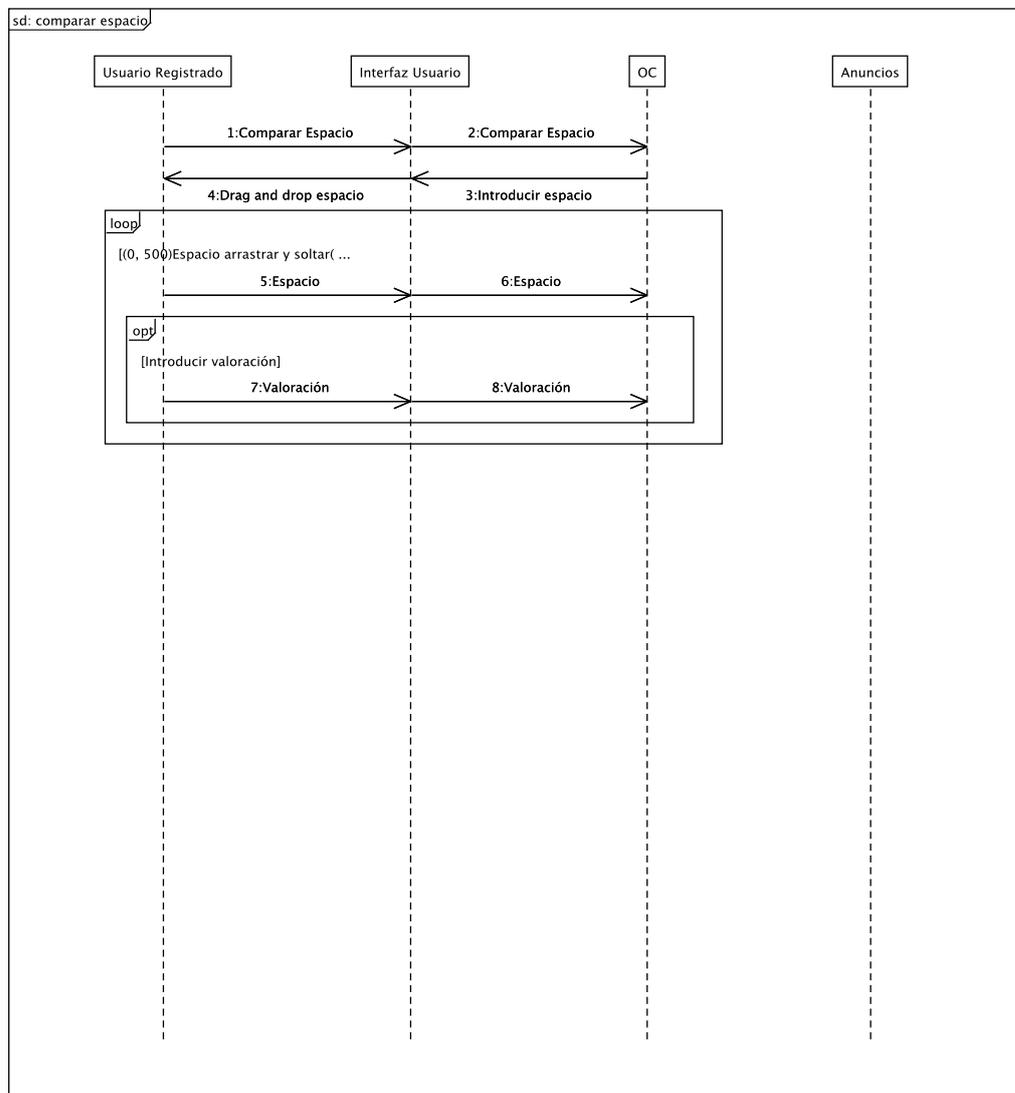


Figura 3.16: Diagrama secuencia Comparar Espacios.

3.4.6. Reservar espacio

En el diagrama de la figura 3.17 se muestra la interacción del usuario con el proceso de reservar un espacio en la aplicación web.

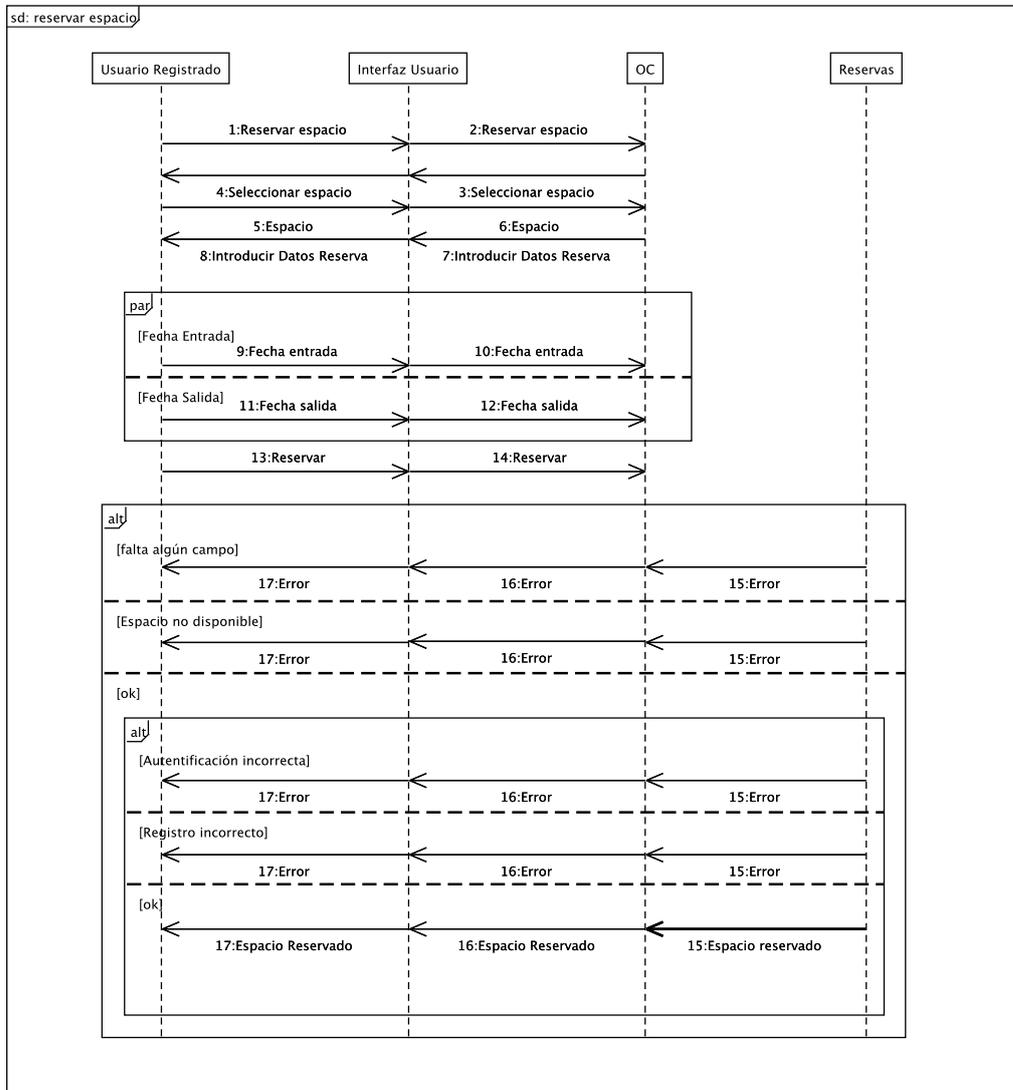


Figura 3.17: Diagrama secuencia Reservar Espacio.

3.4.7. Editar espacio

En el diagrama de la figura 3.18 se muestra la interacción del usuario con el proceso de editar un espacio en la aplicación web.

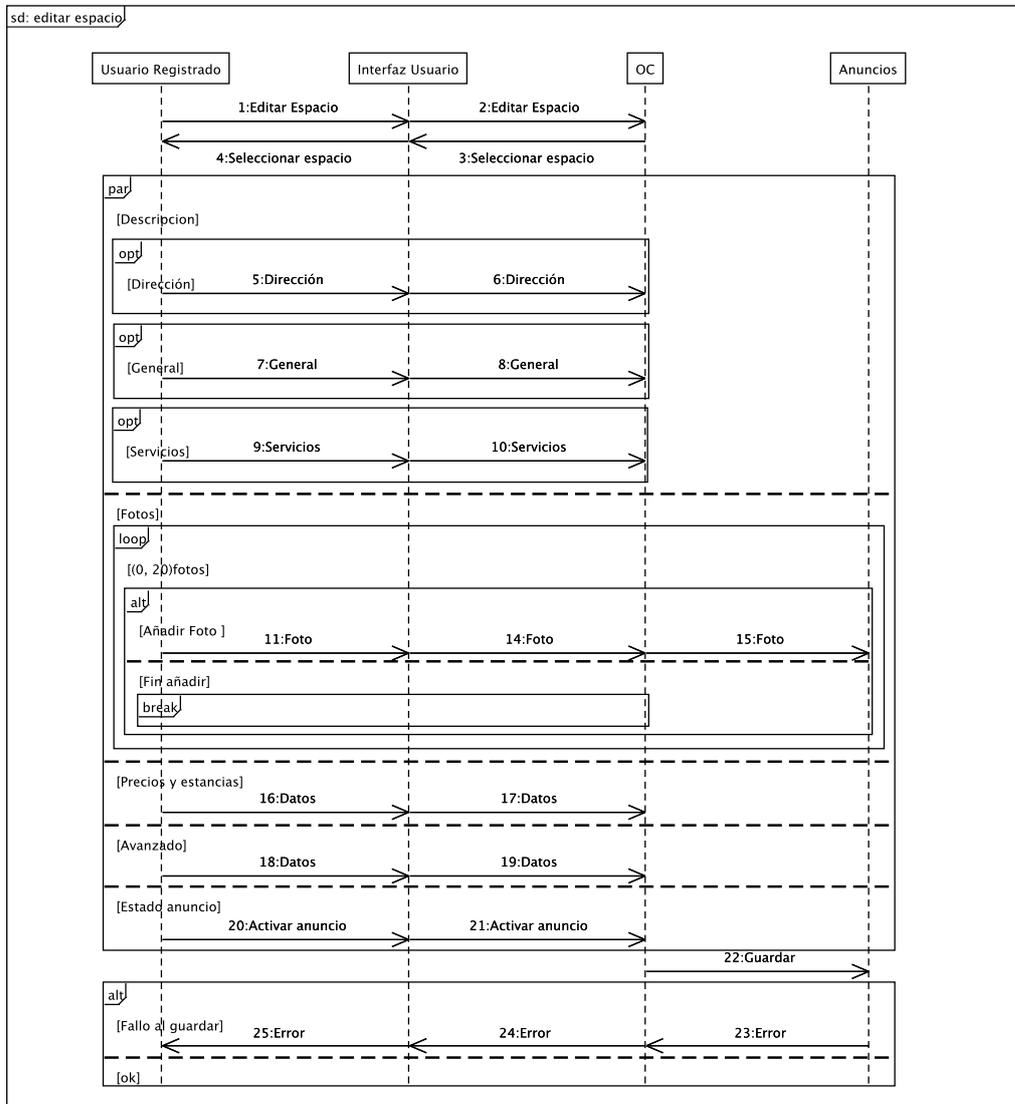


Figura 3.18: Diagrama secuencia Editar Espacio.

Capítulo 4

Diseño

4.1. Introducción

Para el desarrollo de la aplicación web se ha utilizado el Modelo Vista Controlador (MVC) que es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC[34] propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Se ha decidido utilizar este patrón de diseño porque se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

4.2. Arquitectura del sistema

Para conseguir el MVC de la parte Front-End se ha utilizado Backbone [5]. Backbone es una herramienta de desarrollo que se utiliza para desarrollar aplicaciones de una sola página y para mantener las diferentes partes de la aplicación web sincronizadas. En la figura 4.1 se representa la relación entre el modelo, la vista, el controlador y el usuario.

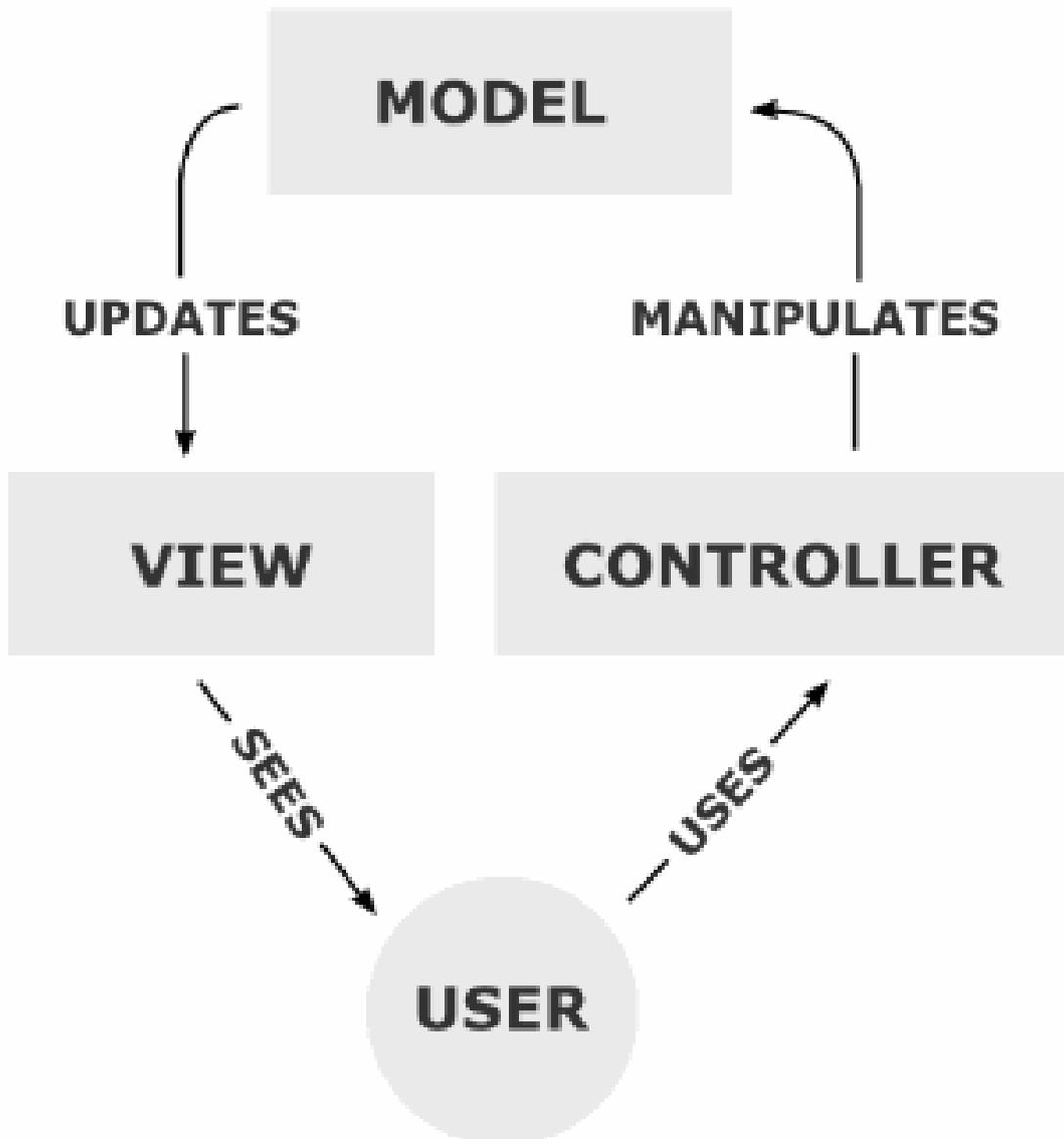


Figura 4.1: MVC.

4.3. Modelo

Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada al usuario. Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

En Backbone los modelos no utilizan métodos de clase, se usan para encapsular un objeto en la capa de negocio. Se entiende como capa de negocio donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Un ejemplo de un modelo lo podemos ver en la figura 4.2.

```

1  define([
2    'Backbone'
3    // Añadir aquí las dependencias de archivos
4  ], function(Backbone) {
5    var ServicioModel = Backbone.Model.extend({
6      name: 'ServicioModel',
7      defaults: {
8        'nombre': '',
9        'descripcion': '',
10       'idservicio': '',
11       'hasanuncio': '',
12       'tiposervicio': ''
13     },
14     initialize: function(){
15     }
16   });
17
18   return ServicioModel;
19
20 });

```

Figura 4.2: Modelo ejemplo.

Una colección es un conjunto de modelos en la parte cliente de la programación por capas de una arquitectura cliente-servidor. Un ejemplo de una colección la podemos ver en la figura 4.3.

```

1  define([
2    'Backbone',
3    'models/ServicioModel'
4  ], function(Backbone, ServicioModel) {
5
6    var ServiciosCollection = Backbone.Collection.extend({
7      model: ServicioModel,
8      url: 'api/servicios/servicios',
9      serviciosTipo: function(tipo){
10
11       var servicios = this.where({'tiposervicio': tipo});
12       //console.log(servicios);
13       return servicios;
14     },
15     ordenar_servicios: function(atributo){
16
17       var servicios_ordenados = this.models.sort(function sortfunction(a,
25     this.models = anuncios_ordenados;
26     },
27     parse : function(resp) {
28       return resp;
29     },
30     initialize: function(){
31       this.reset(jQuery.parseJSON(bootstrappedServicios));
32     }
33   });
34
35   var servicios = new ServiciosCollection();
36
37   return servicios;
38
39 });

```

Figura 4.3: Colección ejemplo.

4.4. Vista

Responde a eventos del usuario e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información. También puede enviar mensajes a su vista

asociada si se solicita un cambio en la forma en que se presenta el modelo, por tanto se podría decir que el controlador hace de intermediario entre la vista y el modelo.

En Backbone las vistas tienen la peculiaridad de no estar necesariamente asociadas a un modelo. Se definen como una pieza lógica reutilizable de la interfaz de usuario. Un ejemplo de una vista lo podemos ver en la figura 4.4.

```
1  define([
2    'Backbone',
3    'geturlparam',
4    'views/index/BuscadorIndexNavView',
5  ], function(Backbone, Url, BuscadorIndexNavView) {
6    var BuscadorIndexAnuncioComparador = Backbone.View.extend({
7      el: $("#div_comparador"),
8      events: {
9        "click #anuncioViewComparador": "anuncioViewComparadorClick"
10     },
11     anuncio_Hide: function(){
12       $('#anuncioHome').hide();
13       $('#anuncioCalendario').hide();
14       $('#anuncioServicio').hide();
15       $('#anuncioDetalle').hide();
16       $('#anuncioMapa').hide();
17       $('#anuncioComentario').hide();
18       $('#anuncioComparador').hide();
19     },
20
21     anuncioViewComparadorClick: function(){
22       Url.setUrlParamHtml5("opciondetalle", 7);
23       this.anuncio_Hide();
24       $("#anuncioComparador").show();
25       BuscadorIndexNavView.view();
26     }
27   });
28   return new BuscadorIndexAnuncioComparador();
29 });
```

Figura 4.4: Vista ejemplo.

4.5. Controlador

Presenta el modelo, es decir, la información y la lógica de negocio en un formato adecuado para interactuar con la interfaz de usuario. También se requiere de dicho modelo la información que debe representar como salida.

En Backbone al controlador se le denomina Router, es un mapa URL a las funciones. Un ejemplo de un controlador lo podemos ver en la figura 4.5.

```
1  define([
2    'Backbone'
3  ], function(Backbone){
4
5    var AppRouter = Backbone.Router.extend({
6      routes: {
7        // Default
8        '*actions' : 'defaultAction'
9      },
10     },
11     defaultAction: function(actions){
12     }
13   });
14
15   var initialize = function(){
16     var app_router = new AppRouter;
17     Backbone.history.start();
18   };
19
20   return {
21     initialize: initialize
22   };
23 });
```

Figura 4.5: Controlador ejemplo.

Capítulo 5

Implementación

5.1. Introducción

En la siguiente sección se describen las tecnologías, herramientas y entorno de trabajo que se han utilizado para realizar la aplicación web en la fase de implementación.

5.2. Tecnologías

5.2.1. SYMFONY

Symfony [20] es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo y automatiza las tareas más comunes. Symfony está desarrollado en PHP.

Las ventajas de la utilización de este framework son de la parte Back-End. En la parte Front-End que es la que ocupa este proyecto, se limita al uso de los interfaces de comunicación con el servidor y la utilización de sus plantillas TWIG.

Las plantillas TWIG son archivos de texto interpretados por PHP que generan HTML. Un ejemplo de una plantilla TWIG la podemos ver en la figura 5.1.

```

1  {% extends app.request.isXmlHttpRequest ? "LyhAppBundle::ajax.html.twig" : "LyhAp
2
3  {% block stylesheets %}
4  <link rel="stylesheet" type="text/css" media="screen" href="{{ asset('bundles/lyh
5  <link rel="stylesheet" type="text/css" media="screen" href="{{ asset('bundles/lyh
6
7  {% endblock %}
8
9  {% block main %}
10 <div id="nav_general">
11   <div class="head_general">
12     <div id="progressbar">
13     </div>
14     <div>
15       <label for="idValoracion">{% trans %}general.valoracion{% endtrans %}
16       <input id="idValoracion" disabled="true" class="inputValoracion" valu
17     </div>
18     <div>
19       <label for="idVisible">{% trans %}general.visible{% endtrans %}</Labe
20     {% if activo == 1 %}
21     <input id="idVisible" type="checkbox" class="inputClick">
22     {% else %}
23     <input id="idVisible" type="checkbox" disabled class="inputClick">
24     {% endif %}
25
26   </div>
27 </div>

```

Figura 5.1: Plantilla Twig ejemplo.

5.2.2. HTML

Para la elaboración de las páginas web que componen la aplicación se ha utilizado HTML[24]. Cada página web se estructura mediante etiquetas DIV que son elementos que sirven para agrupar el documento y permiten aplicar formatos y diseños reutilizando código.

La versión de HTML con la que se ha desarrollado la aplicación web es la 5. HTML5[26] se encuentra en modo experimental pero las últimas versiones de los navegadores más populares soportan la mayoría de las nuevas funcionalidades de este estándar en desarrollo. Se ha decidido utilizar HTML5 por sus avances, mejoras y ventajas respecto a su antecesor.

Algunas de las nuevas funcionalidades que se han utilizado del HTML5 son:

- **Drag and Drop:** Funcionalidad para arrastrar objetos de un DIV a otro o desde fuera de la aplicación.
- **Menu:** Etiquetas para manejar grandes conjuntos de datos.
- **Web semántica:** Etiquetas que permiten describir cual es el significado del contenido, su importancia, su finalidad y las relaciones que existen. Estas etiquetas están orientadas a los buscadores.
- **API Geoposicionamiento:** Permite adaptar los contenidos a la localización del usuario.
- **Canvas:** Etiquetas con codecs para mostrar los contenidos multimedia.
- **Url:** Nuevo tipo de datos para validar el contenido sin Javascript.

5.2.3. CSS

Para dar estilo a las páginas web de la aplicación se ha utilizado CSS[27]. Con el objetivo de separar las reglas de formateo de la estructura básica de la página, se ha creado una hoja de estilo externa, esta es la manera más eficaz de utilizar los CSS.

Los distintos navegadores requieren estilos específicos. Para poder disponer de una funcionalidad similar en todos los navegadores, se ha creado un archivo CSS que incluye las configuraciones del navegador, y que sobrescribe los estilos definidos en una, o varias páginas web.

La versión de CSS que se ha utilizado es la 2.1 que es una versión estable. No se ha utilizado la versión 3 porque todavía no la utilizan los navegadores actuales.

5.2.4. JAVASCRIPT

Para introducir mejoras de funcionalidad en el interfaz de usuario se ha utilizado JavaScript. Javascript está implementado como parte de cualquier navegador web actual. Mediante el DOM[25] JavaScript interactúa con la página web.

Para simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, interacción con AJAX[10] y manejar eventos se ha utilizado una biblioteca de JavaScript llamada JQuery[21].

Como se indica en la sección 4 para implementar el patrón MVC se ha utilizado la librería JavaScript Backbone[5].

Para la programación funcional se ha utilizado la librería de JavaScript Underscore[6] que proporciona una gran cantidad de utilidades sin extender ninguno de los objetos que proporciona JavaScript.

5.2.5. AJAX

AJAX[10] es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el lado del cliente mientras se mantiene la comunicación asíncrona en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones(fuente [32]).

5.2.6. XML

XML[30] es un formato para el intercambio de datos que permite la compatibilidad entre sistemas para compartir información de manera segura.

5.2.7. JSON

JSON[23] es un formato para el intercambio de datos cuyo analizador sintáctico es más sencillos de usar que el de XML.

5.2.8. API GoogleMaps

GoogleMaps[13] es un servidor de aplicaciones de mapas en la web que ofrece imágenes de mapas desplazables entre diferentes ubicaciones. Se ha utilizado la versión 3 del API de GoogleMaps para ubicar y obtener los datos de los espacios publicados.

5.2.9. API JQuery User Interface

Para dotar a la aplicación web de mayor funcionalidad y facilidad se ha utilizado la biblioteca de componentes JQueryUI[22]. Algunos de los componentes que se han utilizado son:

- **Datepicker:** Gestión de calendarios y fechas.
- **Interactions:** Las nuevas funcionalidades de HTML5 de soltar y arrastrar (Drag and Drop) u ordenar items.
- **Widgets:** Funciones de autocompletado de texto, barras de progreso, barras deslizantes (sliders) entre otros.
- **Effects** Efectos como ocultar un DIV o mostrarlo mediante animaciones.

5.3. Herramientas y entorno

5.3.1. Entorno de desarrollo

Los dos únicos entornos que soportan todas las tecnologías usadas para la creación de esta aplicación web son NetBeans y Eclipse. Durante la fase de pruebas que se realizó para determinar que entorno de trabajo era el más adecuado, no fue satisfactorio la utilización de Eclipse. El plugin de Symfony para Eclipse no funciona bien.

Como entorno de desarrollo para todas las tecnologías que se ha utilizado en la creación de esta aplicación web se elige NetBeans[18]. Este entorno de trabajo funciona perfectamente con todas las tecnologías utilizadas en este desarrollo.

5.3.2. Control de versiones

Para trabajar en equipo se ha tenido que utilizar un sistema de control de versiones. El sistema utilizado es el de GitHub[11]. Se probó con otro proveedor de sistemas de control de versiones Bitbucket, pero aunque las pruebas a las que fueron

sometidos ambos servicios fueron satisfactorias, Github tiene mayor funcionalidad cuando se producen conflictos entre las versiones de los usuarios.

5.3.3. Servidor web

Un servidor web HTTP es un programa que procesa una aplicación del lado del servidor realizando conexiones con el cliente generando una respuesta en cualquier aplicación del lado del cliente. Se ha utilizado un servidor web HTTP Apache[17].

5.3.4. Base de datos

La base de datos que se ha utilizado es Mysql[8]. Se ha optado principalmente por esta base de datos porque es la base de datos que ofrecen la mayoría de los proveedores de Hosting. Es una base de datos suficientemente testada para poderla utilizar con total garantía de éxito.

5.3.5. Editor de imágenes

Para la creación y edición de imágenes se ha utilizado Gimp[12]. Se ha seleccionado este manipulador de imágenes porque es gratuito, su manejo es sencillo y los resultados que se obtienen son muy buenos.

5.4. Descripción de la implementación

5.4.1. Estructura de la aplicación web

Los Bundles

Este proyecto realizado en Symfony está estructurado en Bundles. Un Bundle es un conjunto estructurado de archivos que se encuentran en un directorio y que implementan una sola característica. Esto permite que cada funcionalidad esté totalmente desacoplada del resto del proyecto y pueda reutilizarse en otros proyectos. Todo en Symfony es un Bundle.

La aplicación web está compuesta por 14 Bundles:

- **AdminBundle:** Bundle para administrar la aplicación web.
- **AnuncioBundle:** Bundle para crear un nuevo anuncio.
- **AnuncioManageBundle:** Bundle para administrar los anuncios creados.
- **AppBundle:** Bundle que aporta los contenidos base que son idénticos en todas las páginas web, como la barra principal de navegación.
- **AyudaBundle:** Bundle que gestiona los contenidos de ayuda de la aplicación web.

- **BoilerBundle:** Bundle tipo que sirve de plantilla para crear otros bundles
- **BuscadorBundle:** Bundle principal de la aplicación web. Gestiona la búsqueda y la visualización de toda la información pública de los espacios web. Mediante este Bundle se inicia la reserva de los espacios anunciados.
- **CompartimosBundle:** Bundle que gestiona los anuncios de espacios compartidos.
- **IdiomaBundle:** Bundle que gestiona los idiomas de la web.
- **MainBundle:** Bundle que gestiona el buscador de la página de inicio de la aplicación web
- **MessageBundle:** Bundle que gestiona todos los mensajes que se envían a los usuarios.
- **PersonalBundle:** Bundle para administrar los espacios anunciados.
- **ReservaBundle:** Bundle para realizar las reservas de los espacios anunciados.
- **UserBundle:** Bundle para administrar las cuentas de los usuarios.

Estructura de cada Bundle

Cada Bundle está compuesto por cinco partes:

- **Controller:** Contiene los archivos PHP que sirven para gestionar las peticiones de páginas web que realizan los usuarios a través de la URL. Estos archivos están desarrollados en la otra parte del proyecto denominada Back-End.
- **DependencyInjection:** Contiene archivos PHP de configuración de Symfony que están desarrollados en la otra parte del proyecto denominada Back-End.
- **Entity:** Contiene las entidades es decir, las clase básicas que contienen datos. Estos archivos están desarrollados en la otra parte del proyecto denominada Back-End.
- **Resources:** Contiene todos los archivos necesarios para el funcionamiento de la parte Front-End.

Estructura de carpeta Resources

La carpeta Resources de cada Bundle está compuesta por 4 partes:

- **Config:** Contiene los archivos xml generados por la parte del proyecto denominada Back-End para el intercambio de datos entre el interfaz de usuario y el servidor web. También contiene un archivo de configuración (ver figura 5.2) de todas los patrones aceptados para realizar las oportunas peticiones desde el cliente al servidor en formato JSON.

```

5  Lyh_buscador_json:
6    pattern: /json/search.{_format}
7    defaults: { _controller: LyhBuscadorBundle:Json:getBuscador }
8  requirements:
9    _format: json
10   _method: GET
11
12 Lyh_reserva_json:
13   pattern: /json/reserva.{_format}/
14   defaults: { _controller: LyhBuscadorBundle:Json:getReserva }
15 requirements:
16   _format: json
17   _method: GET

```

Figura 5.2: Archivo configuración routing.yml.

- **Translations:** Contiene los diccionarios de palabras y construcciones del Bundle que se usan para las traducciones a los distintos idiomas. Es escalable y aunque de momento sólo está disponible en tres idiomas (inglés, francés y español) en cualquier momento se puede ampliar añadiendo las traducciones correspondientes. En la figura 5.2 se detalla un ejemplo de parte del archivo de traducciones de un Bundle. El formato de los archivos de traducciones contiene dos columnas la primera corresponde al identificador y la segunda al conjunto de palabras al que está asociado el identificador.

```

1  zoom: Ampliar
2  list: Lista
3  shared: Compartidos
4  drag.and.drop: arrastrar y soltar
5  price: Precio
6  distance: Distancia
7  score: Puntuación
8  relevance: Relevancia
9  selected: Seleccionados

```

Figura 5.3: Ejemplo traducción español.

- **Views:** Contiene los archivos TWIG del Bundle.
- **Public:** Contiene todos los archivos CSS, templates HTML, imágenes del Bundle y Javascript.

Estructura de carpeta Public

- **CSS:** Contiene todos los archivos CSS que utiliza el Bundle. Cada archivo CSS está estructurado en niveles, que añaden o eliminan los distintos estilos de cada elemento que conforman cada página web dentro del Bundle. En la figura 5.4 se muestra un ejemplo de parte del CSS del Bundle de reservas.

```
18  □ .ui-widget-content_extend { /* sobrecargada jquery calendar*/
19      border: 0 !important;
20      background: transparent !important;
21      width: 140px !important;
22  }
23  }
24  □ .datepickerPreSeleccionado{
25      background: #73ac00 !important;
26      color:yellow !important;
27      opacity: 0.6;
28  }
29  □ .datepickerPeriodo{
30      background: #8258FA !important;
31      color:yellow !important;
32  }
33  }
34  □ .datepickerPrecios{
35      float:left;
36      position:absolute;
37      margin-top:-15px;
38      margin-left:-25px;
39      color:#73ac00 !important;
40      background: yellow !important;
41      height: 12px !important;
42      width: 25px !important;
43      text-align: center;
44      border-radius: 25px;
45      border: 3px solid #E6E6E6;
46      cursor: pointer !important;
47  }
48  }
```

Figura 5.4: Ejemplo CSS.

- **Templates HTML:** Contiene todos los Templates HTML que son scripts en un archivo de texto que generan HTML, y permiten la reutilización de código. En la figura 5.5 se muestra un ejemplo de Template HTML.

```
1 <script type="text/template" class="template" id="templateDetalle">
2   <div id="anuncioDetalle" class="anuncioMain">
3     <div id="divDetalleTitulo">
4       <input disabled id="detalleTitulo" type="text" value=""/>
5     </div>
6     <div id="divDetalleDescripcion">
7       <textarea disabled rows="20" cols="50" id="detalleDescripcion">
8       </textarea>
9     </div>
10
11     <div id="divDetallesTipo">
12       <div id="divDetalleTipoHabitacion">
13         <label>espacio</label>
14         <input disabled id="detalleTipoHabitacion" type="text" value=""/>
15       </div>
16       <div id="divDetalleTipoPropiedad">
17         <label>Tipo propiedad</label>
18         <input disabled id="detalleTipoPropiedad" type="text" value=""/>
19       </div>
20     </div>
21   </div>
22 </script>
```

Figura 5.5: Ejemplo template HTML.

- **Imágenes del Bundle:** Contiene todas las imágenes utilizadas para el diseño y la apariencia de las distintas vistas del Bundle. El formato estándar de las imágenes utilizado en los Bundles es PNG (Portable Network Graphics).
- **Javascript:** Contiene todos los archivos Javascript del Bundle.

Estructura de carpeta Javascript

La carpeta Javascript de cada Bundle está compuesta por 4 partes:

- **Classes:** Javascript es un lenguaje de programación interpretado orientado a objetos pero que no dispone de clases. Con el objetivo de diseñar bien la aplicación web se ha utilizado Javascript con una estructura que imita las organizaciones en clases. Esta carpeta contiene las clases que se han utilizado en el Bundle. En la figura 5.6 se muestra un ejemplo de parte de una clase del Buscador Bundle.

```

1  define([
2      'Backbone',
3      'collections/ReservaDataCollection',
4      'collections/AnunciosDataCollection',
5      'collections/AnunciocalendarioDataCollection',
6      'geturlparam',
7      'calendar_localization'
8  ], function(Backbone, reserva, anuncios, anunciocalendario, Url, $) {
9      var DatePickerExtend = function() {
10         var datos = {};
11         var reservas = null; //variable global colleccion
12         var anunciocalendarios = null; //variable global colleccion
13         var colorPeriodo = 0;
14         var restoInicial = 0;
15         var restoAtras = 0;
16         var mesActual = 0;
17         var restoFinal = 0;
18         //var dateFormat = "dd/mm/yy";
19         var diasReserva = 0;
20         var crearReservas = function(anuncio, mes) { //funcion global
21             reservas = reserva.anuncio_mes(anuncio, mes);
22         };
23         var crearAnunciocalendarios = function(anuncio, mes) { //funcion global
24             anunciocalendarios = anunciocalendario.anuncio_mes(anuncio, mes);
25         };
26         var marcarAnuncio = function() {
27             var id = parseInt($('.anuncio_reserva').attr('id').match('[0-9]+'));
28             $("#anunMin" + id).addClass("anuncio_reservado");
29             $("#anuncio" + id).addClass("anuncio_reservado");
30             $("#anuncioSelect" + id).addClass("anuncio_reservado");
31         };
32     };

```

Figura 5.6: Ejemplo clase DatePickerExtend.

- **Collections:** Contiene todas las colecciones de los modelos utilizados para la arquitectura MVC del Bundle.(apartado 4.3)
- **Models:** Contiene los modelos utilizados para la arquitectura MVC del Bundle.(apartado 4.3)
- **Views:** Contiene las vistas utilizadas para la arquitectura MVC del Bundle.(apartado 4.4)

5.4.2. Interfaces de usuario

La página de inicio de la aplicación web es muy sencilla, un buscador rápido de espacios que filtra los resultados por localidad. En la figura 5.7 se detalla el interfaz de búsqueda.

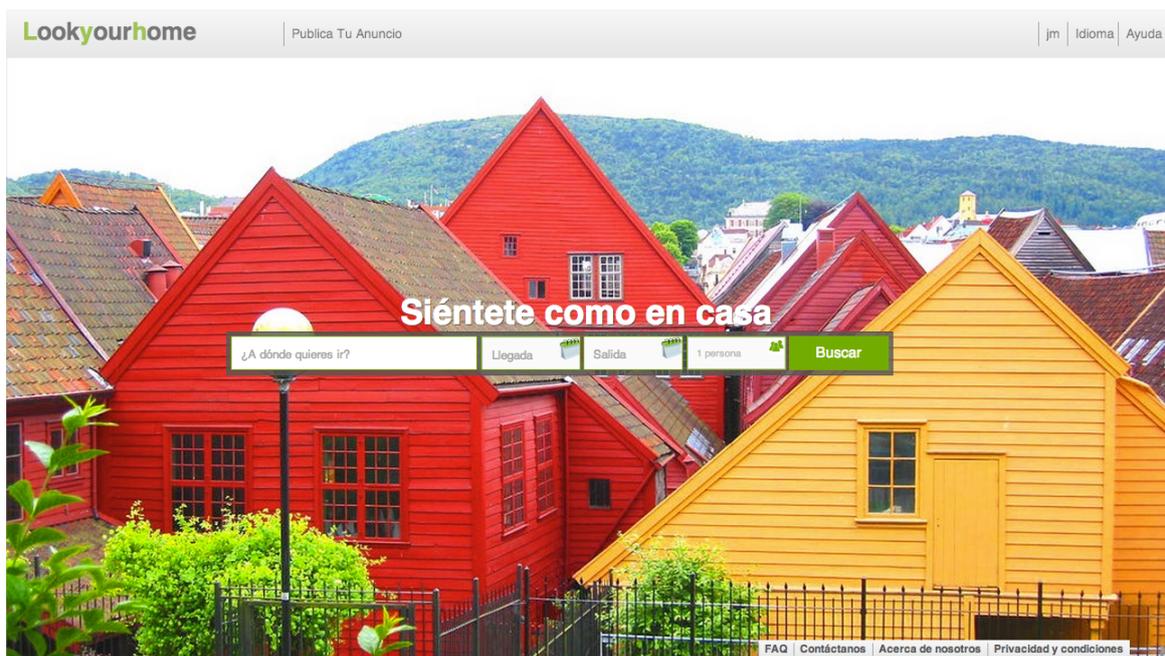


Figura 5.7: Interfaz de usuario de búsqueda de espacios.

Desde esta interfaz inicial se puede acceder a todas las demás interfaces que componen la aplicación.

Al realizar la búsqueda se accede a la interfaz de selección de espacios, dónde se muestran los espacios obtenidos en la búsqueda. En la figura 5.8 se detalla la interfaz de selección.

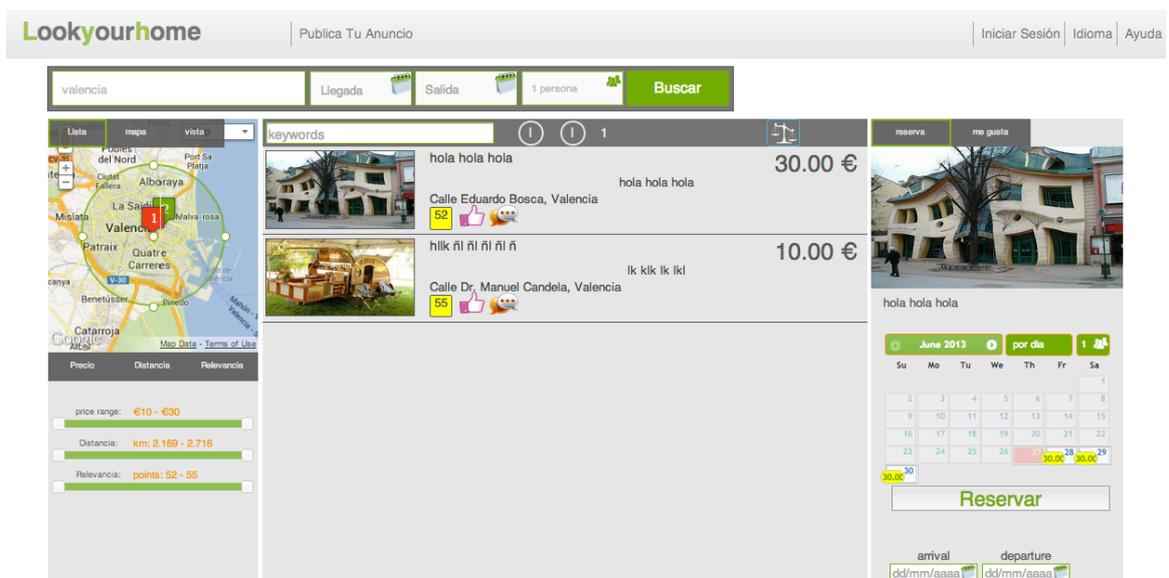


Figura 5.8: Interfaz de usuario de selección de espacios.

Desde la interfaz de selección se puede realizar nuevas búsquedas aplicando una serie de filtros. Además se puede visualizar los detalles de los espacios y realizar la reserva. Desde esta interfaz el usuario visualiza la descripción y los detalles del espacio consultado como se indica en la 5.9. Otra opción de la que el usuario dispone en esta interfaz es poder consultar los distintos servicios que ofrece el espacio seleccionado como se muestra en la figura 5.10. También existe la posibilidad de visualizar las fotos publicadas del espacio como en la figura 5.11.

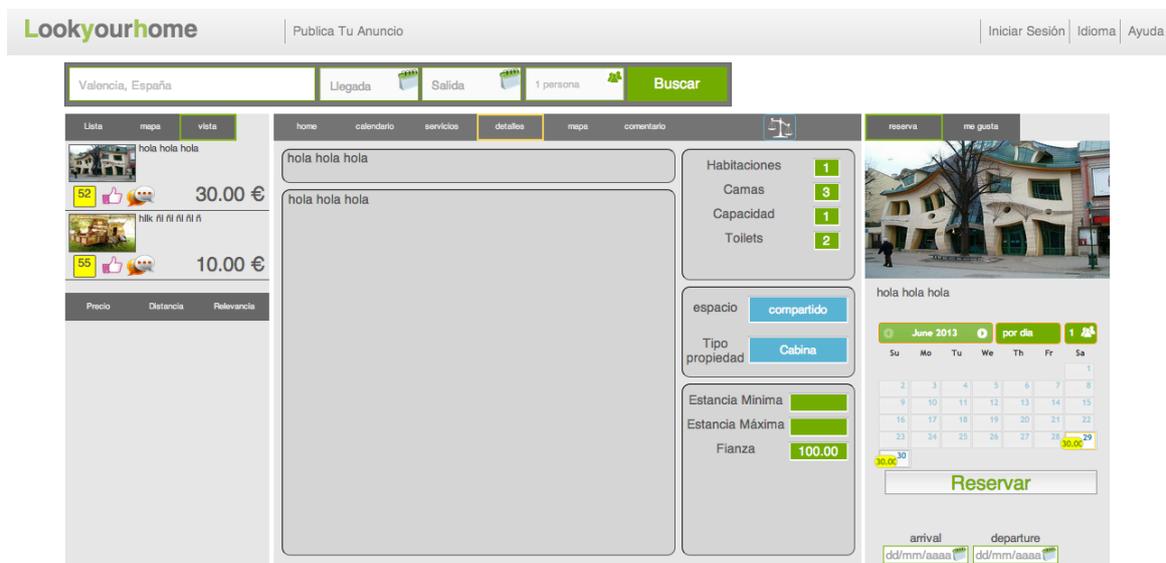


Figura 5.9: Interfaz de usuario para visualizar los detalles del espacio.

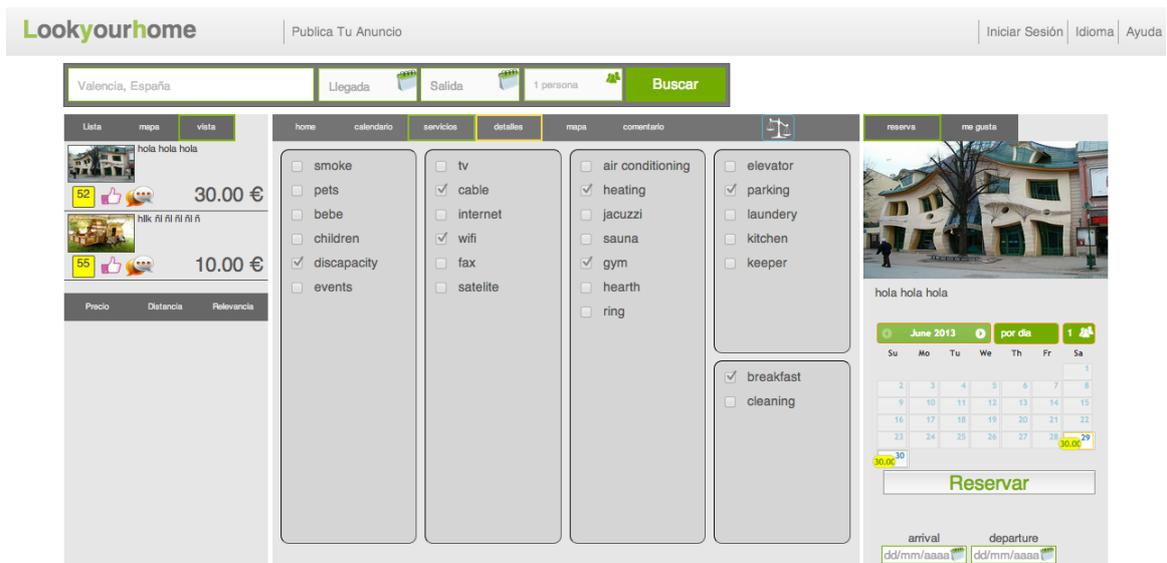


Figura 5.10: Interfaz de usuario para visualizar los servicios del espacio.

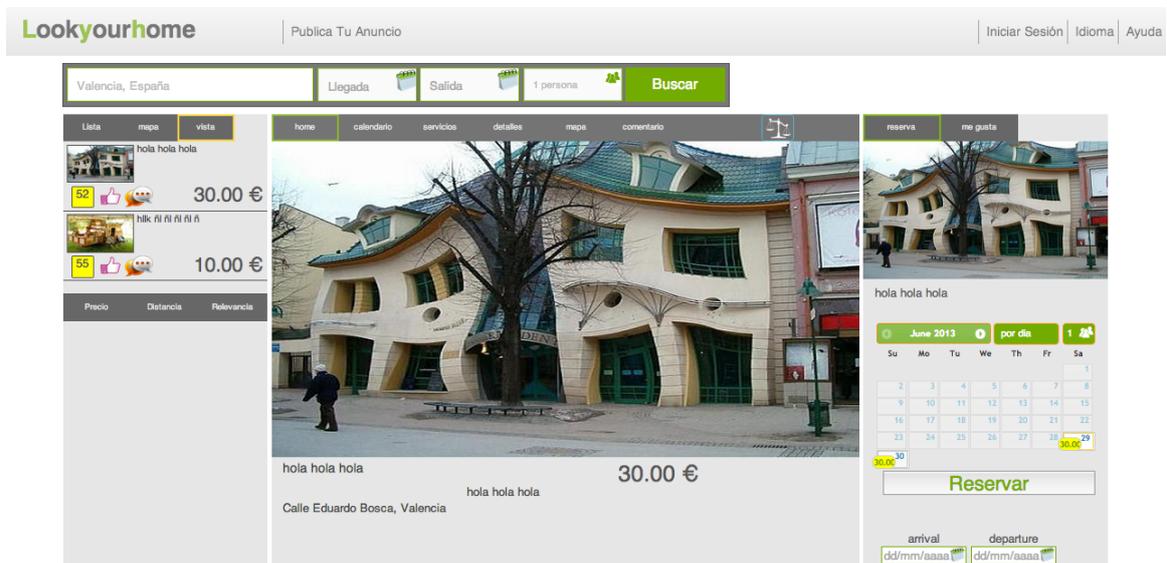


Figura 5.11: Interfaz de usuario para publicar las fotos del espacio.

En la figura 5.12 se muestra la interfaz de usuario para publicar los anuncios de los espacios. Este proceso se realiza en tan sólo dos pasos y de una manera sencilla e intuitiva.

The screenshot shows the 'Publica tu anuncio' interface in two steps. **Paso 1** is titled 'Publica tu anuncio' and includes input fields for 'Localidad' (with a placeholder 'Introduce una ubicación') and 'Dirección'. Below these is a large area with the text 'PINCHA AQUI'. At the bottom, there are dropdown menus for 'Capacidad' (set to 1) and 'Espacio' (set to 'compartido'), and a 'Publicar' button. **Paso 2** is split into 'Register' and 'Login' sections. The 'Register' section has fields for 'Nombre de usuario:', 'Email:', 'Contraseña:', and 'Repita la contraseña:'. The 'Login' section has fields for 'Nombre de usuario:' and 'Contraseña:'. There is a 'Connect with Facebook' button, a 'Registrar' button, a 'Recordar' checkbox, and an 'Entrar' button.

Figura 5.12: Interfaz para publicar el anuncio del espacio.

La siguiente interfaz sirve para administrar los anuncios que se han publicado mediante la interfaz anterior. En la figura 5.13 se muestra dicha interfaz. Desde esta interfaz se pueden administrar la dirección (ver figura 5.14), los precios (ver figura 5.16), las fotos (ver figura 5.15), los servicios, y hacer visible el anuncio (ver figura 5.17).

The screenshot shows the 'Administrar Casas' interface. At the top, there are navigation tabs: 'Administrar Casas', 'Administrar Reservas', 'Perfil', and 'Mensajes'. Below this is a grid of four property listings. Each listing includes a photo, a reference number, the address, location, and a zip code, along with an 'Editar' button.
 - **referencia:1**: Calle Dr. Manuel Candela, Valencia, 46021 Valencia.
 - **referencia:2**: Calle Eduardo Bosca, Valencia, 46023 Valencia.
 - **referencia:3**: Plaza Mayor, Port Saplaya, 46128 Valencia.
 - **referencia:4**: blasco Ibañez 34, Valencia, España, 46021 Valencia.
 Listings 3 and 4 show a camera icon and the text 'Imagen no disponible'.

Figura 5.13: Interfaz para administrar los espacios publicados.

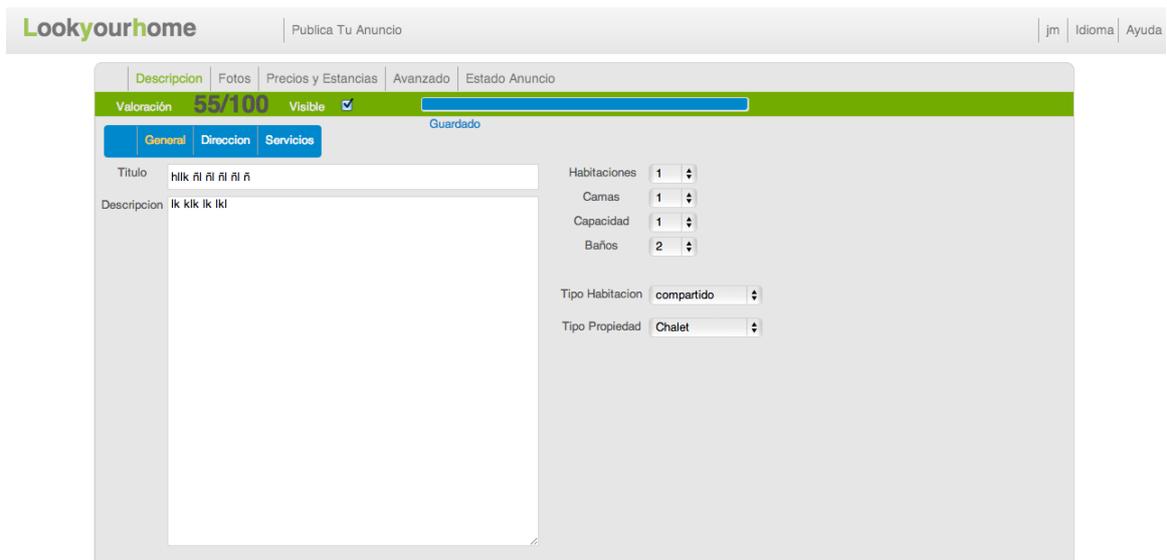


Figura 5.14: Interfaz para administrar la dirección de los espacios.

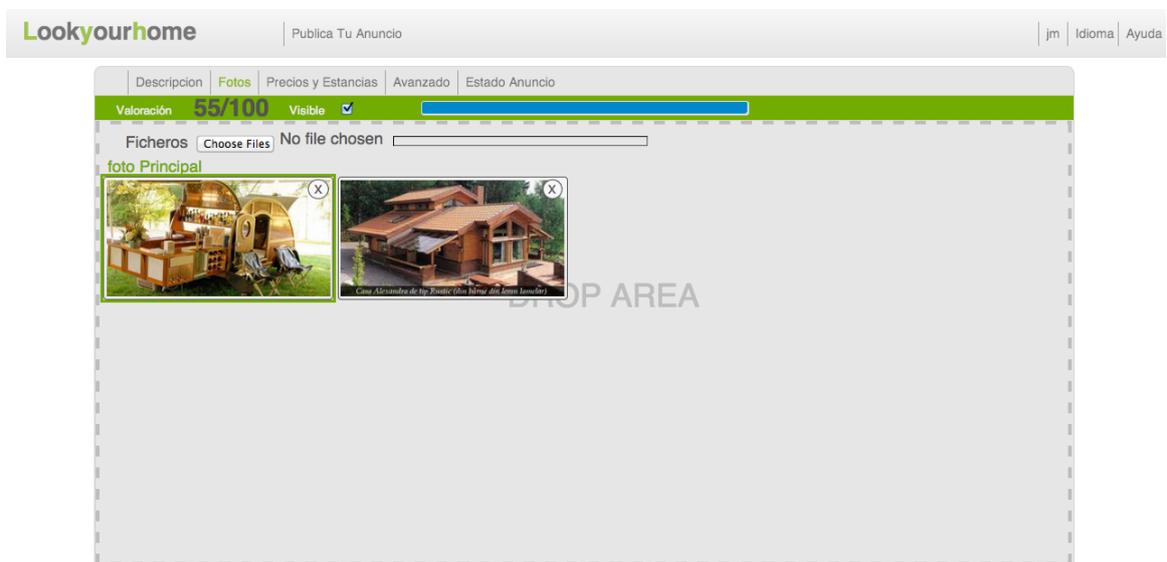


Figura 5.15: Interfaz para administrar las fotos de los espacios.

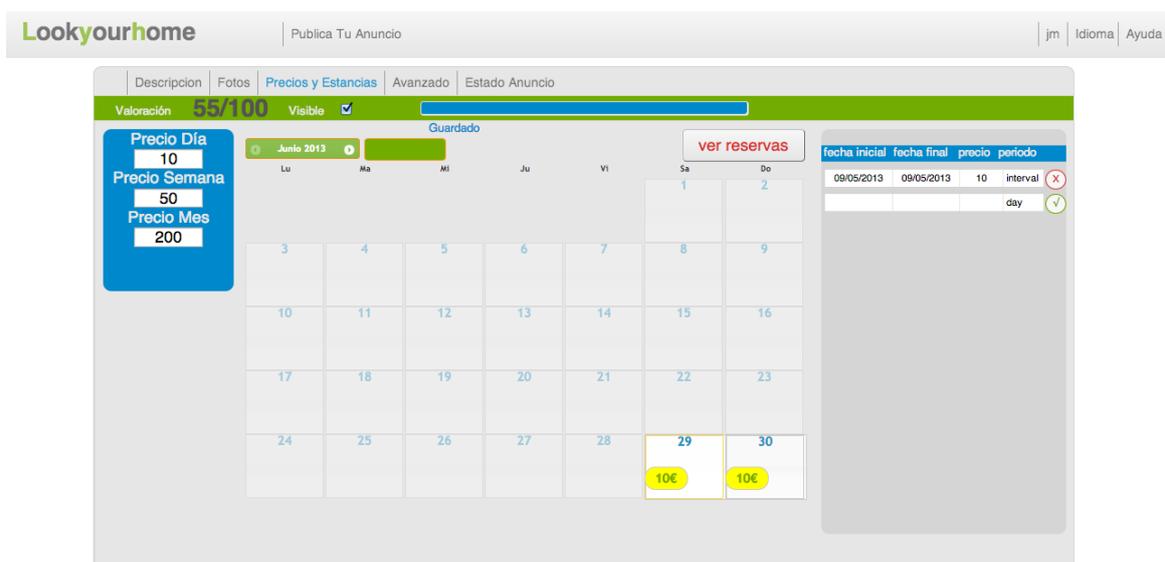


Figura 5.16: Interfaz para administrar los precios de los espacios.

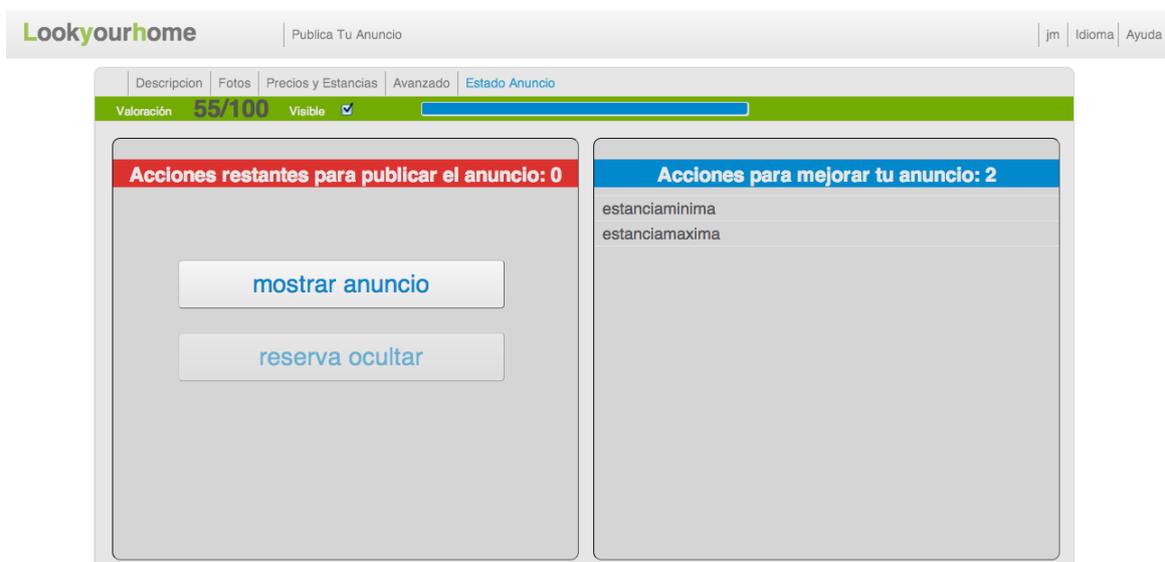


Figura 5.17: Interfaz administrar el estado de los espacios.

Capítulo 6

Evaluación

6.1. Introducción

En esta sección se validan todos los Bundles que componen la aplicación web. Para ello se validan todos los estándares utilizados y se realizado un análisis de los enlaces rotos que puedan existir durante la navegación del sitio web.

6.2. Validación de estándares web

6.2.1. Estándar HTML

Para la validación del estándar HTML se ha utilizado W3C validator service[28]. Se ha validado el HTML de todos los Bundles. Las validaciones obtenidas son las siguientes:

- **AdminBundle:** El resultado obtenido son 0 errores. Al pertenecer el HTML a la parte privada no se ha podido realizar directamente la validación. Se ha validado el código HTML generado introduciéndolo manualmente en la aplicación como se muestra en la figura 6.1.

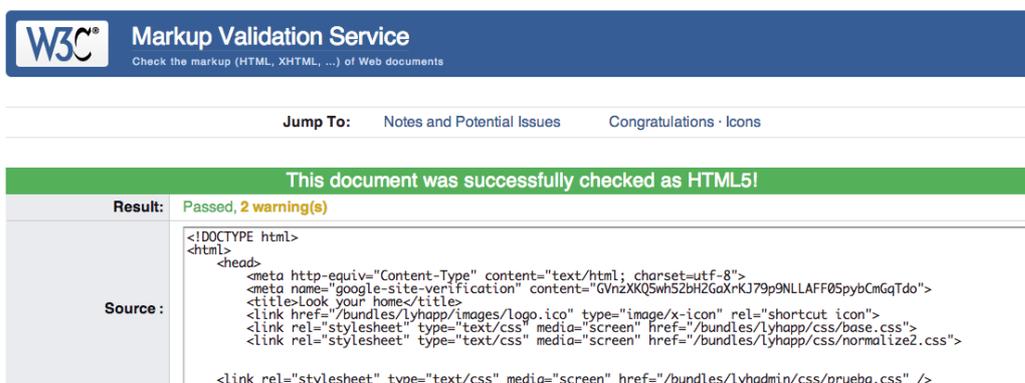


Figura 6.1: Validación del HTML del Bundle admin.

- **AnuncioBundle:** El resultado obtenido son 0 errores.

- **AnuncioManageBundle:** El resultado obtenido son 0 errores.
- **AppBundle:** El resultado obtenido son 0 errores.
- **AyudaBundle:** El resultado obtenido son 0 errores.
- **BuscadorBundle:** El resultado obtenido son 10 errores. Los errores que se han producido se deben a que los input no pueden contener el atributo value vacío. Al tratarse de un input tipo botón no es necesario corregir este error porque funciona correctamente en todos los navegadores y no puede producirse ningún problema de seguridad por no tener valor el atributo.

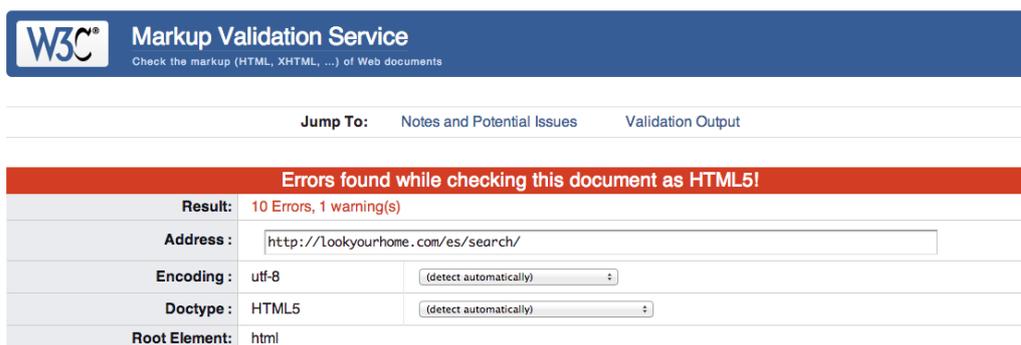


Figura 6.2: Validación del HTML del Bundle buscador.

- **CompartimosBundle:** El resultado obtenido son 0 errores.
- **IdiomaBundle:** El resultado obtenido son 0 errores.
- **MainBundle:** El resultado obtenido son 0 errores. En la figura 6.3 se muestra la validación realizada con resultado satisfactorio.

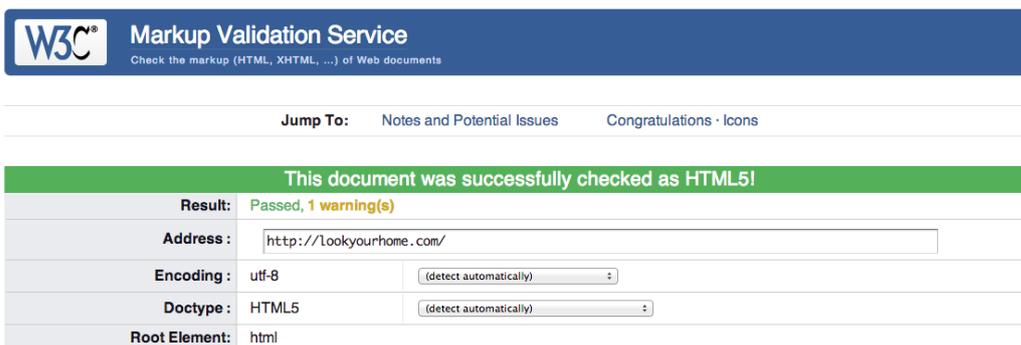


Figura 6.3: Validación del HTML del Bundle main.

- **MessageBundle:** El resultado obtenido son 0 errores.
- **PersonalBundle:** El resultado obtenido son 0 errores.
- **ReservaBundle:** El resultado obtenido son 0 errores.
- **UserBundle:** El resultado obtenido son 0 errores.

6.2.2. Estándar CSS

Para la validación del estándar CSS se ha utilizado W3C validator service[28]. Se ha validado el CSS de todos los Bundles. Las validaciones obtenidas son las siguientes:

- **AdminBundle:** El resultado obtenido son 0 errores. En la figura 6.4 se muestra la validación realizada con resultado satisfactorio.



Figura 6.4: Validación del CSS del Bundle admin.

- **AnuncioBundle:** El resultado obtenido son 9 errores. Estos errores se deben a que la función linear-gradient y appearance no funciona correctamente en todos los navegadores. No es necesario corregir este error porque sí funciona correctamente en todos los navegadores a excepción de Internet Explorer. En la figura 6.5 se muestra la validación realizada.

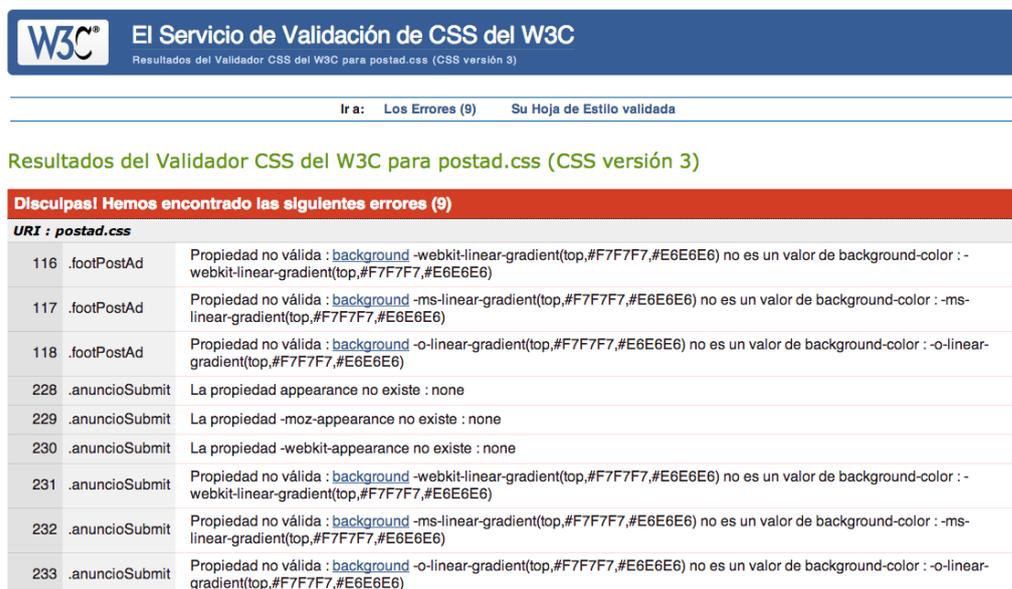


Figura 6.5: Validación del CSS del Bundle anuncio.

- **AnuncioManageBundle:** El resultado obtenido son 36 errores. Estos errores se deben a que la función linear-gradient y appearance no funciona correctamente en todos los navegadores. No es necesario corregir este error porque

sí funciona correctamente en todos los navegadores a excepción de Internet Explorer.

- **AppBundle:** El resultado obtenido son 3 errores. Estos errores se deben a que la función `linear-gradient` no funciona correctamente en todos los navegadores. No es necesario corregir este error porque sí funciona correctamente en todos los navegadores a excepción de Internet Explorer.
- **AyudaBundle:** El resultado obtenido son 0 errores.
- **BuscadorBundle:** El resultado obtenido son 23 errores. Estos errores se deben a que la función `linear-gradient` y `appearance` no funciona correctamente en todos los navegadores. No es necesario corregir este error porque sí funciona correctamente en todos los navegadores a excepción de Internet Explorer.
- **CompartimosBundle:** El resultado obtenido son 0 errores.
- **IdiomaBundle:** El resultado obtenido son 0 errores.
- **MainBundle:** El resultado obtenido son 3 errores. Estos errores se deben a que la función `linear-gradient` no funciona correctamente en todos los navegadores. No es necesario corregir este error porque sí funciona correctamente en todos los navegadores a excepción de Internet Explorer. En la figura 6.6 se muestra la validación realizada.

W3C® El Servicio de Validación de CSS del W3C
Resultados del Validador CSS del W3C para main.css (CSS versión 3)

Ir a: [Los Errores \(3\)](#) [Su Hoja de Estilo validada](#)

Resultados del Validador CSS del W3C para main.css (CSS versión 3)

Disculpas! Hemos encontrado las siguientes errores (3)

URI : main.css

52	ol#footer_nav	Propiedad no válida : <code>background -webkit-linear-gradient(top,#F7F7F7,#E6E6E6)</code> no es un valor de <code>background-color</code> : <code>-webkit-linear-gradient(top,#F7F7F7,#E6E6E6)</code>
53	ol#footer_nav	Propiedad no válida : <code>background -ms-linear-gradient(top,#F7F7F7,#E6E6E6)</code> no es un valor de <code>background-color</code> : <code>-ms-linear-gradient(top,#F7F7F7,#E6E6E6)</code>
54	ol#footer_nav	Propiedad no válida : <code>background -o-linear-gradient(top,#F7F7F7,#E6E6E6)</code> no es un valor de <code>background-color</code> : <code>-o-linear-gradient(top,#F7F7F7,#E6E6E6)</code>

Figura 6.6: Validación del CSS del Bundle main.

- **MessageBundle:** El resultado obtenido son 0 errores.
- **PersonalBundle:** El resultado obtenido son 0 errores.
- **ReservaBundle:** El resultado obtenido son 0 errores.
- **UserBundle:** El resultado obtenido son 0 errores.

6.2.3. Estándar Javascript

Para la validación del estándar Javascript se ha utilizado la consola para depurar errores Javascript de Firefox. Se ha validado el Javascript de todos los Bundles. Las validaciones obtenidas son las siguientes:

- **AdminBundle:** El resultado obtenido son 0 errores.
- **AnuncioBundle:** El resultado obtenido son 0 errores.
- **AnuncioManageBundle:** El resultado obtenido son 0 errores.
- **AppBundle:** El resultado obtenido son 0 errores.
- **AyudaBundle:** El resultado obtenido son 0 errores.
- **BuscadorBundle:** El resultado obtenido son 0 errores.
- **CompartimosBundle:** El resultado obtenido son 0 errores.
- **IdiomaBundle:** El resultado obtenido son 0 errores.
- **MainBundle:** El resultado obtenido son 0 errores. Hay una advertencia que no es necesaria corregirla. En la figura 6.7 se muestra la validación realizada.

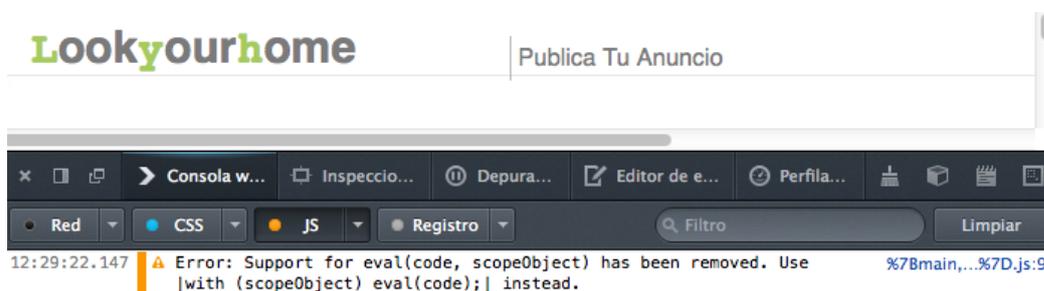


Figura 6.7: Validación del Javascript del Bundle main.

- **MessageBundle:** El resultado obtenido son 0 errores.
- **PersonalBundle:** El resultado obtenido son 0 errores.
- **ReservaBundle:** El resultado obtenido son 0 errores.
- **UserBundle:** El resultado obtenido son 0 errores.

6.3. Análisis de navegación

Los enlaces rotos son uno de los puntos que hay que tener en cuenta a la hora de optimizar un sitio web para el posicionamiento en buscadores. Un enlace roto dificulta a los buscadores la indexación de la página. Los enlaces rotos perjudican el posicionamiento web.

Para comprobar los enlaces rotos se ha utilizado un servicio web llamado Broken Link Checker[16].

- **AdminBundle:** El resultado obtenido son 0 enlaces rotos.
- **AnuncioBundle:** El resultado obtenido son 0 enlaces rotos.
- **AnuncioManageBundle:** El resultado obtenido son 0 enlaces rotos.
- **AppBundle:** El resultado obtenido son 0 enlaces rotos.
- **AyudaBundle:** El resultado obtenido son 0 enlaces rotos.
- **BuscadorBundle:** El resultado obtenido son 0 enlaces rotos. En la figura 6.8 se muestra la validación realizada.

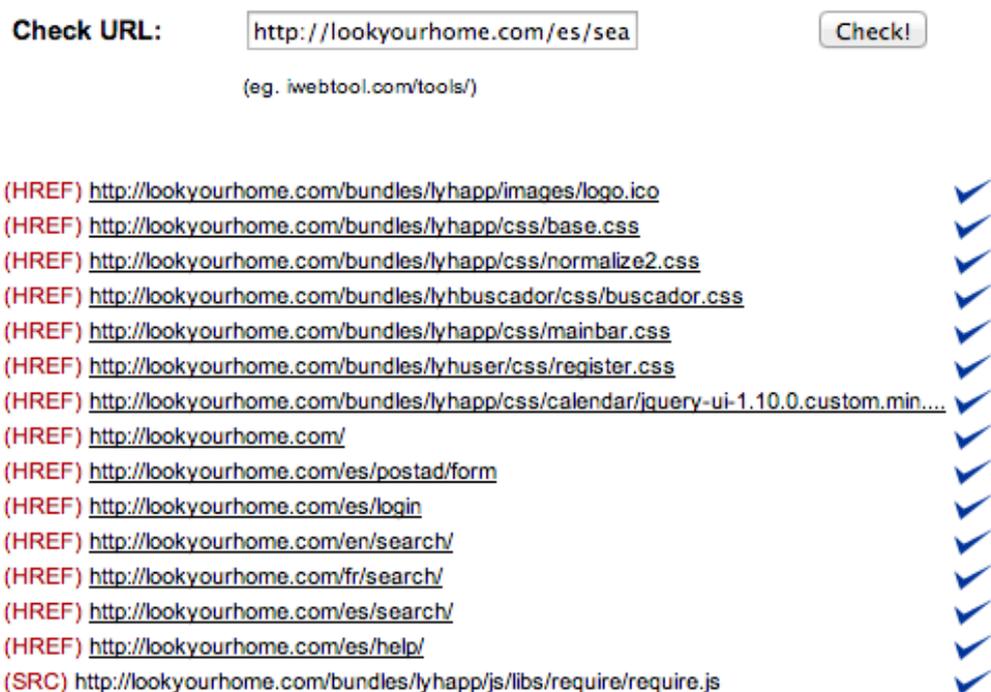


Figura 6.8: Validación de los enlaces del Bundle buscador.

- **CompartimosBundle:** El resultado obtenido son 0 enlaces rotos.
- **IdiomaBundle:** El resultado obtenido son 0 enlaces rotos.
- **MainBundle:** El resultado obtenido son 4 enlaces rotos. Esto se debe a que todavía no se han terminado de desarrollar estas secciones y existen estos enlaces rotos intencionadamente. En la figura 6.9 se muestra la validación realizada.

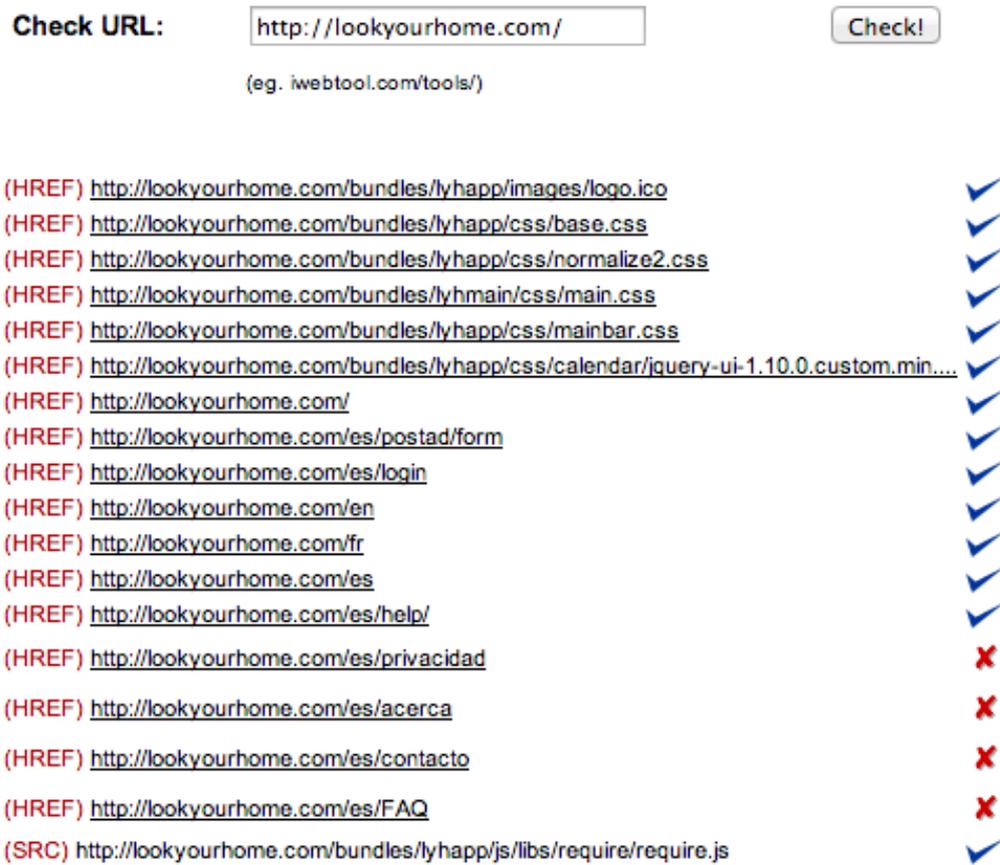


Figura 6.9: Validación de los enlaces del Bundle main.

- **MessageBundle:** El resultado obtenido son 0 enlaces rotos.
- **PersonalBundle:** El resultado obtenido son 0 enlaces rotos.
- **ReservaBundle:** El resultado obtenido son 0 enlaces rotos.
- **UserBundle:** El resultado obtenido son 0 enlaces rotos.

Capítulo 7

Conclusiones

7.1. Resumen

La realización de este proyecto ha comprendido una serie de etapas que han acabado por dar como resultado final la aplicación web de alquiler de espacios look-yourhome.com.

En las etapas iniciales apenas se veía como progresaba el proyecto. Había que decidir y formarse en distintos lenguajes de programación y entornos de trabajo. El proyecto global ha tenido dos partes, la parte Front-End y la parte Back-End, por lo que ha habido que trabajar en equipo para que las dos partes casen adecuadamente al finalizar el proyecto. Para ello se tuvo que utilizar un control de versiones de modo que cualquier imprevisto pudiera ser subsanado. También había que planificar el trabajo porque, al trabajar en paralelo, se tenía que avanzar coordinadamente para que no se produjeran paros improductivos y las dos partes pudieran cumplir con los plazos de entrega pactados.

Aparte de los lenguajes de programación utilizados en esta parte (Front-End) se han tenido que aprender y usar los de la otra parte (Back-End), con el objetivo de dar mayor fluidez al desarrollo y poder ayudarse entre las partes en los momentos más críticos y en los cuellos de botella que se han ido produciendo a lo largo del proyecto.

En más de una ocasión ha habido que desperdiciar el trabajo realizado de una semana porque los resultados no eran satisfactorios. Pero paso a paso ha ido creciendo el proyecto y al final el resultado ha sido el esperado. Al final se han cumplido todas las expectativas que se habían planteado, cubriendo los requisitos iniciales y algunos más que han ido apareciendo conforme se desarrollaba la aplicación web. También han ido naciendo nuevas ideas, pero que por el factor tiempo no se han incluido en el proyecto.

7.2. Valoración

La experiencia de haber trabajado en equipo ha sido muy satisfactoria, y al final se tiene el convencimiento de que se ha aprovechado el tiempo y los conocimientos doblemente. Otro aspecto a tener en cuenta es que, mediante la evolución del proyecto, se puede apreciar como una idea poco a poco va cogiendo forma hasta que finalmente se obtiene la gratificación de poder asistir a su materialización definitiva de ese pensamiento inicial.

En un proyecto como éste se tiene la oportunidad de aprender con los errores que se van cometiendo durante todo el proceso de desarrollo, y uno a uno se les va dando solución.

De todos los lenguajes que he utilizado quizás el que más he disfrutado ha sido con JQuery. Llega un momento en el cual, tras haber escrito miles de líneas de código, has asimilado lo suficiente para que, sin apenas pruebas, no cometes casi ningún error y todo lo que piensas lo llevas a cabo sin ningún problema.

7.3. Trabajo futuro

Al finalizar el proyecto quedan muchas ideas que al final no se han llevado a termino y que se tendrán que valorar para determinar cuales merece la pena desarrollar como continuación de este proyecto, o para que sean la semilla de un nuevo proyecto.

Si que hay dos trabajos futuros que habrá que realizar. Habrá que terminar las secciones incompletas del proyecto, como es el caso de la sección de ayuda, FAQ, acerca de nosotros y privacidad y condiciones. También será necesario la adaptación de la aplicación web para dispositivos móviles.

Capítulo 8

Referencias bibliográficas

- [1] *Ley Orgánica 15/1999, de Protección de Datos de Carácter Personal. Sección I. Disposiciones generales. Departamento Jefatura del Estado. ref. BOE-A-1999-23750. 13 dic 1999. 2.2.4*
- [2] *Real Decreto 1720/2007, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999 de protección de datos de carácter personal. Sección I. Disposiciones generales. Departamento Ministerio de Justicia. ref. BOE-A-2008-979. 21 dic 2007. 2.2.4*
- [3] IEEE Recommended Practice for Software Requirements Specifications. *IEEE Std. 830-1998*, 1998. 1.5, 2.1.1
- [4] 3Scale. What is an API? Your guide to the Internet Business (R)evolution, 2011. 2.1.3
- [5] Jeremy Ashkenas. Backbone.js. <http://backbonejs.org>, oct 2010. 4.2, 5.2.4
- [6] Jeremy Ashkenas. Underscore.js versión 1.5.1. <http://underscorejs.org>, feb 2013. 5.2.4
- [7] Félix Buendía García. *Una guía para la realización y supervisión de proyectos final de carrera (PFC) en el ámbito de la web. 2008-247*. Editorial Universidad Politécnica de Valencia, 2008.
- [8] Oracle Corporation. Mysql versión 5.5. <http://www.mysql.com>, apr 2010. 5.3.4
- [9] FWikipedia. Spam. <http://es.wikipedia.org/wiki/Spam>, jun 2013. 2.1.3
- [10] Jesse James Garrett. Ajax, asynchronous javascript and xml. <http://www.asp.net/ajax>, 2005. 5.2.4, 5.2.5
- [11] Inc GitHub. Github. <https://github.com>, jun 2013. 5.3.2
- [12] GNU. Gimp, gnu image manipulation program. <http://www.gimp.org>, jun 2013. 5.3.5

- [13] Google. Google maps javascript api v3. <https://developers.google.com/maps/documentation/javascript>, jun 2013. 5.2.8
- [14] Ecma International. *IEEE Recommended Practice for Software Requirements Specifications. IEEE Std. 830-1998*. 1998. 2.1.4
- [15] Ecma International. *ECMAScript Language Specification. Standard ECMA-262*. 2011. 2.1.3
- [16] Iwebtool. Broken link checker web tools. http://www.iwebtool.com/broken_link_checker, 2011. 6.3
- [17] Robert McCool. Apache versión 2.4. <http://httpd.apache.org>, feb 2013. 5.3.3
- [18] Netbeans. Netbeans ide 7.3. <https://netbeans.org>, feb 2013. 5.3.1
- [19] OMG. Unified Modeling Language. *UML 2.4.1*, 2011. 3.1
- [20] Fabien Potencier. Symfony 2.3.0. <http://symfony.com>, may 2013. 5.2.1
- [21] John Resig. Jquery 1.9. <http://jquery.com>, ene 2013. 5.2.4
- [22] John Resig. Jquery user interface. <http://jqueryui.com>, ene 2013. 5.2.9
- [23] Tesla Ventures. Json, javascript object notation. <http://www.json.org>, apr 2001. 5.2.7
- [24] W3C. Html, hypertext markup language. <http://www.w3.org/html>, dec 1999. 5.2.2
- [25] W3C. Dom, document object model. <http://www.w3.org/DOM>, jun 2009. 5.2.4
- [26] W3C. Html5, hypertext markup language. <http://www.w3.org/html5>, oct 2009. 5.2.2
- [27] W3C. Css 2.1, cascading style sheets. <http://www.w3.org/Style/CSS>, jun 2011. 5.2.3
- [28] W3C. W3c markup validation service versión 1.3. <http://validator.w3.org>, jun 2013. 6.2.1, 6.2.2
- [29] W3C. Word wide web consortium. <http://www.w3.org>, jun 2013. 2.3.4
- [30] W3C. Xml extensible markup language. <http://www.xml.org>, 2013. 5.2.6
- [31] w3counter. Web browser market share. <http://www.w3counter.com/globalstats.php>, jun 2013. 2.2
- [32] Wikipedia. Ajax. <http://es.wikipedia.org/wiki/AJAX>, jun 2013. 5.2.5

- [33] Wikipedia. Casos de uso. http://es.wikipedia.org/wiki/Caso_de_uso, jun 2013. 3.2
- [34] Wikipedia. Modelo vista controlador. http://es.wikipedia.org/wiki/Modelo_Vista_Controlador, jun 2013. 4.1