UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
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: Back-End

Proyecto Final de Carrera

INGENIERÍA INFORMÁTICA

Autor: D.CARLOS ESCRICHE IZQUIERDO

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

25 de julio de 2013

Índice general

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, y a partir de ésta el 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 un aumento de la ocupación de las viviendas, tiempos de publicación y alquiler de los espacios más optimizados, mejoras en la facilidad de uso, y un manejo intuitivo. Otro objetivo es el de conseguir que el mayor número de usuarios utilice la aplicación.

1.4. Trabajo realizado

La aplicación web consta de dos partes diferenciadas: Front-end y Back-end. La parte Front-end se encarga interactuar con el usuario de la aplicación, de recoger los datos de entrada del usuario y de procesarlos de una manera conforme a la especificación que el Back-end pueda usar. La parte Back-end procesa la entrada de datos desde el Front-end, se encarga de la persistencia de datos y envía la información procesada al Front-end.

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 hace referencia a la parte Back-end. Aquí se han desarrollado tareas relacionadas con el servidor web, bases de datos relacionales [50] y lógica de negocio de la aplicación web.

En este proyecto se ha hecho primero una especificación de requisitos y una planificación conjunta del trabajo a realizar. También se ha diseñado la aplicación utilizando una arquitectura desacoplada para mejorar el mantenimiento del software, así como una base de datos relacional para almacenar los datos de manera persistente. Se ha hecho uso de una API de comunicación con redes sociales. Además, se han utilizado herramientas de trabajo en equipo y de control de versiones para la comunicación con la parte Front-End de la aplicación.

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 de presentación, es la que ve o con la que interactúa 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 se centra en la implementación, donde se detallan las tecnologías utilizadas, las herramientas y el entorno de desarrollo. Además, ofrece 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 último capítulo indica las referencias bibliográficas utilizadas para realizar el proyecto y la memoria.

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. [12]

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. **Autentificación:** 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 autentificación.

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 le 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:** Frecuently 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 [32]

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 autentificación.

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. [27]

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 describen 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. [31, 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 ofrece 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 linea 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.
- **Autentificación de usuario:** Para poder realizar determinadas acciones de la aplicación es necesario autentificarse previamente y haberse registrado. El usuario introduce su nombre de usuario[24] y contraseña[6]. Opcionalmente el usuario también puede autentificarse 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 autentificado.
- **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 autentificado.
- **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 autentificado.
- **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 autentificado. 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 autentificación 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

2.3.1. Requisitos de las interfaces externas

Requisitos de usuario

La interfaz de usuario debe ser fácil de utilizar, innovadora, rápida y con una experiencia de usuario enriquecedora. Cualquier usuario sin una formación específica debe de poder usarla.

Las herramientas utilizadas para la creación de la interfaz deben ofrecer las garantías de estabilidad necesarias que la aplicación creada sea robusta. También deben permitir que la interfaz sea rápida para que los usuarios no desesperen en su utilización.

Requisitos de software

Como requisito de software se establece que la aplicación deberá ser compatible con los navegadores google Chrome versión 27, Firefox versión 21, Safari versión 6.1, 6 y 5.1, Internet Explorer 10 y Android 4.

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 usuario 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 al los idiomas iniciales de la aplicación (Inglés, Francés y Español). Una vez seleccionado el administrar 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 *Consulta las Traducciones*. 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 administrar realiza la acción Consultar Traducciones. El sistema entonces muestra un listado con todas las traducciones existentes. El administrar 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 las 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 las 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 autentificarse 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 autentificarse mediante Facebook o mediante la propia aplicación. Si elige Facebook entonces el sistema externo de Facebook se encarga de validar y autentificar al usuario. En caso de que el usuario elija autentificarse 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 autentificado anteriormente puede cerrar sesión. Entonces el sistema redirige al usuario a la página principal. El usuario ya no está entonces autentificado 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 todas 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
Introducció	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 al 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 el número de usuarios conectados simultáneamente habrá que tomar decisiones sobre 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 no debe permitir que estos cambios acaben siendo un problema que acabe con la viabilidad del proyecto.

Para empezar el desarrollo, con un alojamiento web estándar será suficiente, mientras no se tengan datos que evidencien un cambio del sistema o un aumento considerable del número de usuarios.

2.3.4. Restricciones del diseño

Se han seguido, para el desarrollo de esta aplicación web, las recomendaciones marcadas por el W3C [48] en lo relativo al diseño web y aplicaciones. Estas recomendaciones incluyen en los estándares para la construcción de páginas web del dispositivo y otras tecnologías para las aplicaciones web.

2.3.5. Atributos del sistema software

Seguridad

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 autentificación, 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 necesario que el alojamiento del sitio web esté redundado en distintas ubicaciones. Si por cualquier motivo el sistema encargado de almacenar la base de datos o la aplicación web falla, al estar la web redundada en varios sistemas, 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 evitando la presencia de fallos en el servidor y en el código de la aplicación.

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.

Además la aplicación deberá ser fácilmente mantenible por los desarrolladores, por lo que el código creado debe ser muy claro y bien estructurado, así como la estructura general de la aplicación web.

Portabilidad

Al utilizar los principales navegadores web para ejecutar la aplicación, ésta 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 alquiler de viviendas. Los tiempos de respuesta deben de ser lo suficientemente rápidos para que el usuario no tenga que sufrir largas esperas.

El usuario debe tener la funcionalidad completa al terminar de cargar cualquier página. 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 contener los suficientes datos para no estar continuamente realizando peticiones y saturar el servidor web.

2.3.6. Otros requisitos

Será requisito que la aplicación web se pueda utilizar en dispositivos móviles en un futuro. Para ello se utilizarán lenguajes y tecnologías que permitan, de una manera fácil, la adaptación de la aplicación web a estos dispositivos.

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 [37](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 [52] 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 despempeñar varios roles diferentes describe en la sección 2.2.3 donde se hace referencia a los usuarios del sistema especificados.

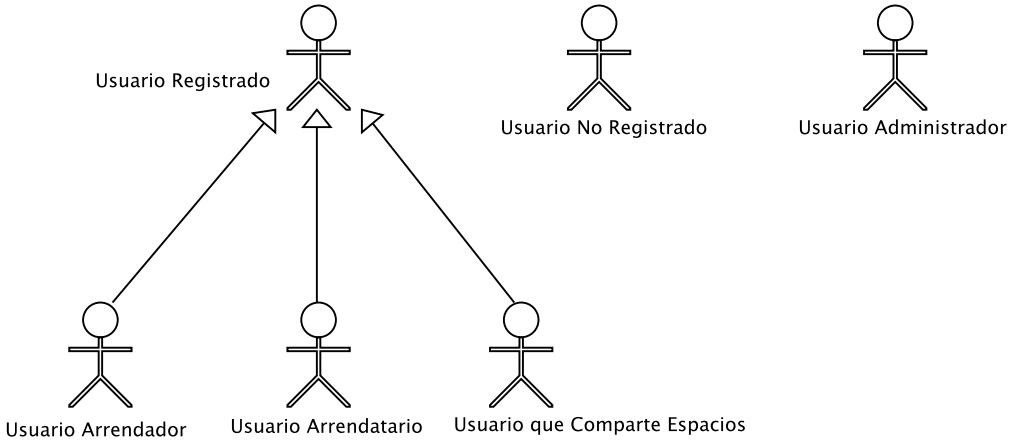

Figura 3.1: Actores.

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*.

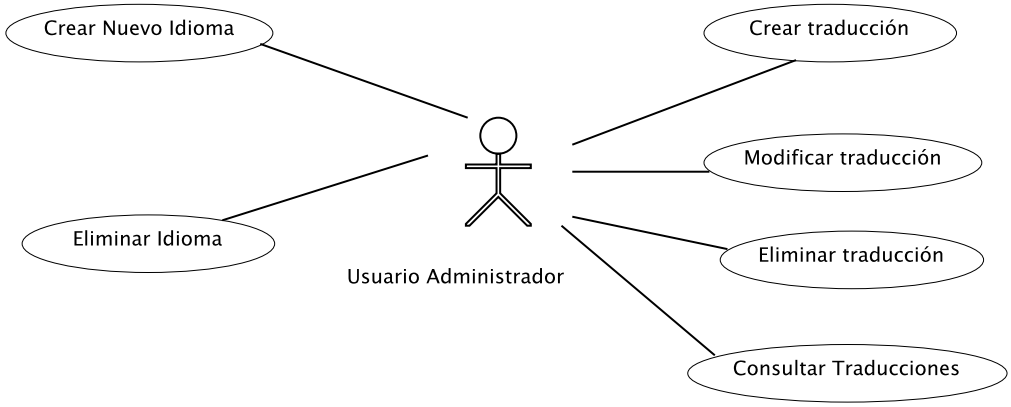

Figura 3.2: Gestión del idioma.

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.

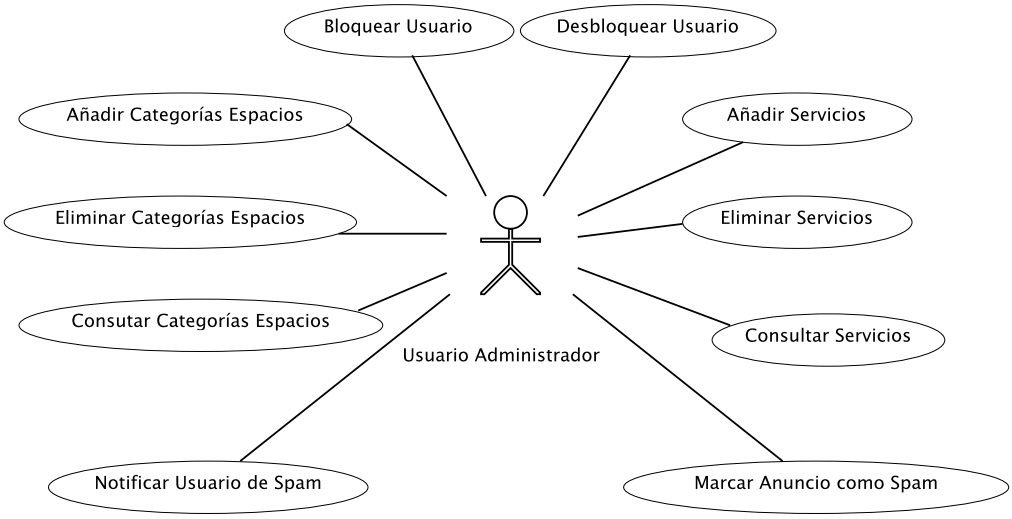

Figura 3.3: Gestión de Espacios y Usuarios.

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.

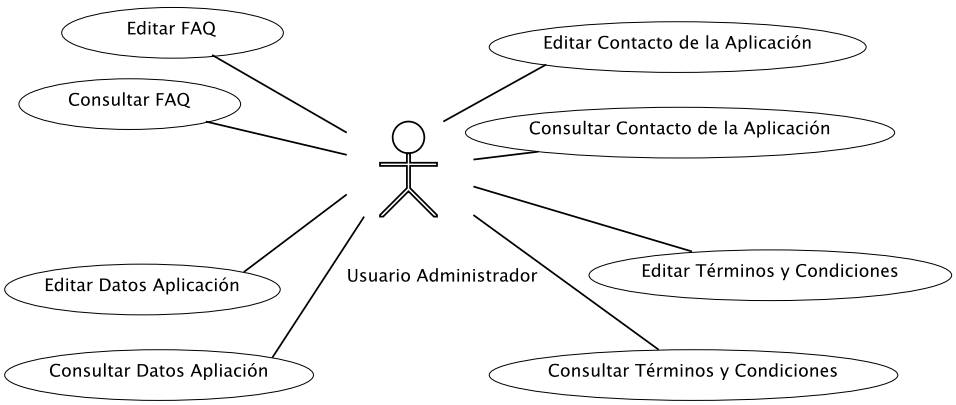

Figura 3.4: Gestión de Datos de la Aplicación.

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.

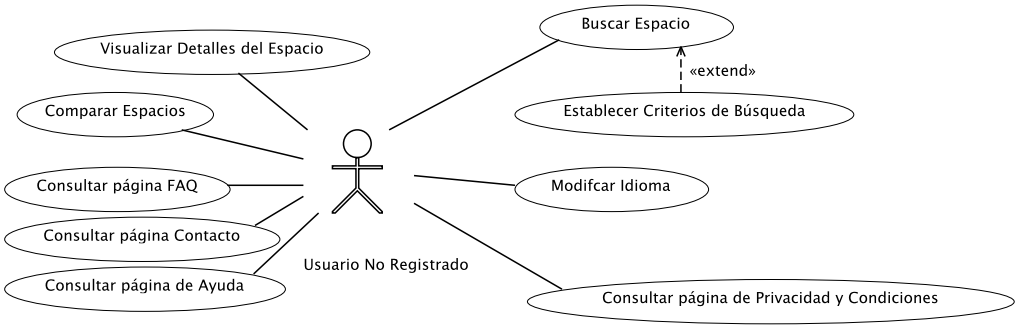

Figura 3.5: Usuario No Registrado.

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 autentificarse, consultar y editar sus datos personales, gestionar sus mensajes dentro de la aplicación y añadir espacios a favoritos.

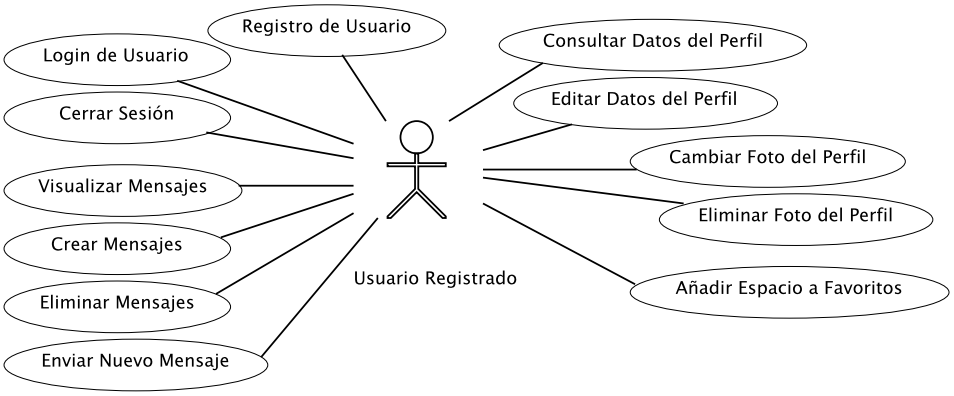

Figura 3.6: Usuario Registrado.

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.

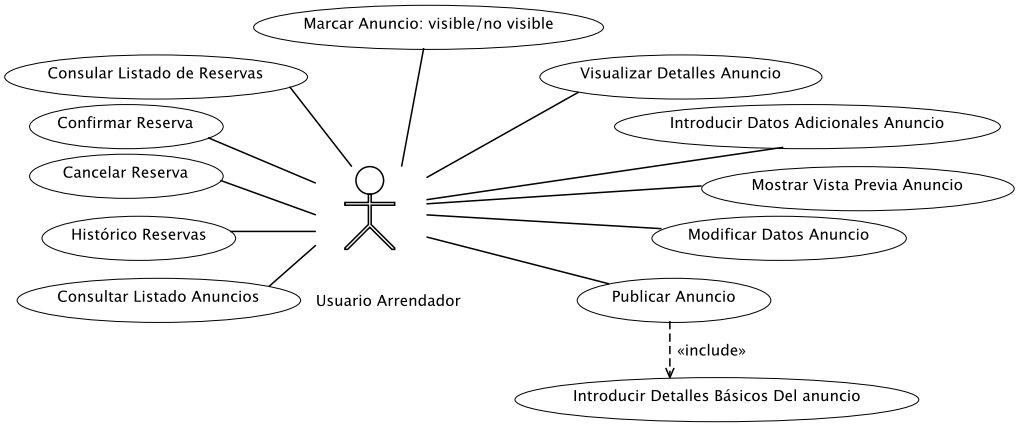

Figura 3.7: Usuario Arrendador.

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.

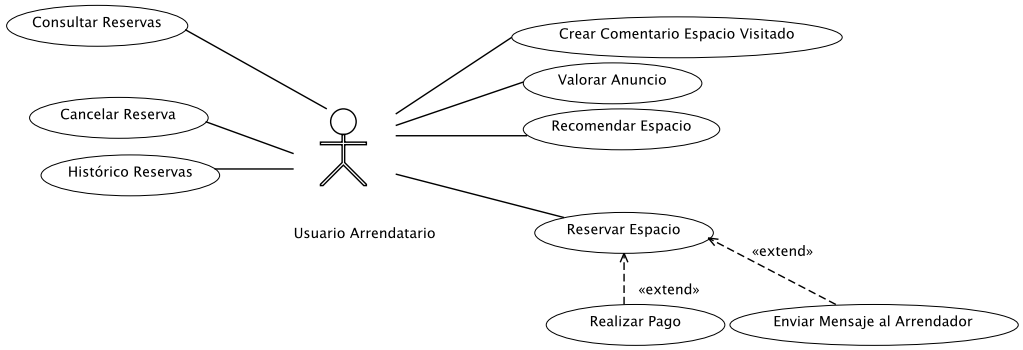

Figura 3.8: Usuario Arrendatario.

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.

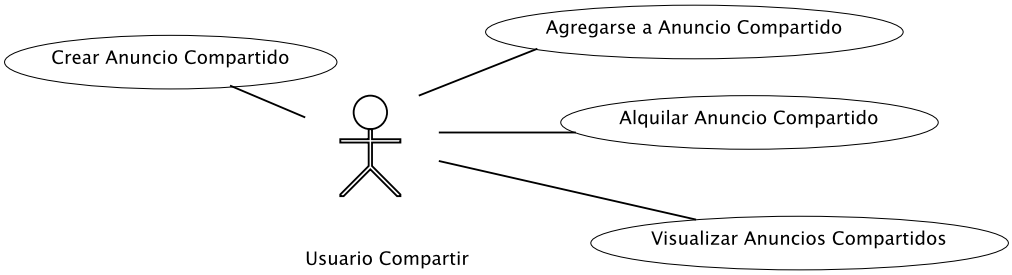

Figura 3.9: Usuario que comparte espacios.

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 divido 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.

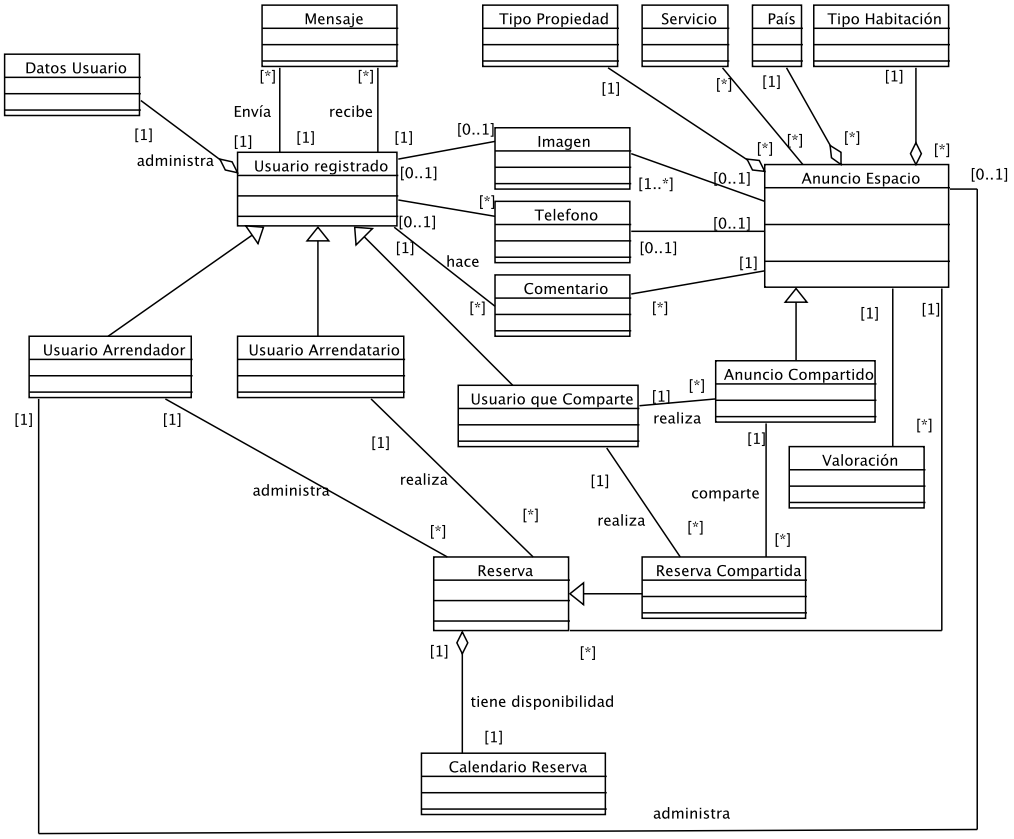

Figura 3.10: Diagrama clases Usuarios.

En este diagrama se puede observar que existe un usuario registrado, especializado después en tres nuevos usuarios: usuario arrendador, usuario arrendatario y usuario que comparte. El usuario registrado posee datos de usuario y puede añadir imágenes a su cuenta, números de teléfono y crear comentarios de espacios que ha visitado. Los usuarios arrendador y arrendatario realizan reservas de espacios, mientras que el usuario que comparte, realiza y comparte reservas compartidas.

Los anuncios de espacios tienen tipo de propiedad, servicios, país y tipo de habitación. Además son administrados por el usuario arrendador. Éste puede subir imágenes y añadir un teléfono de contacto al espacio.

3.3.2. Administrador

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

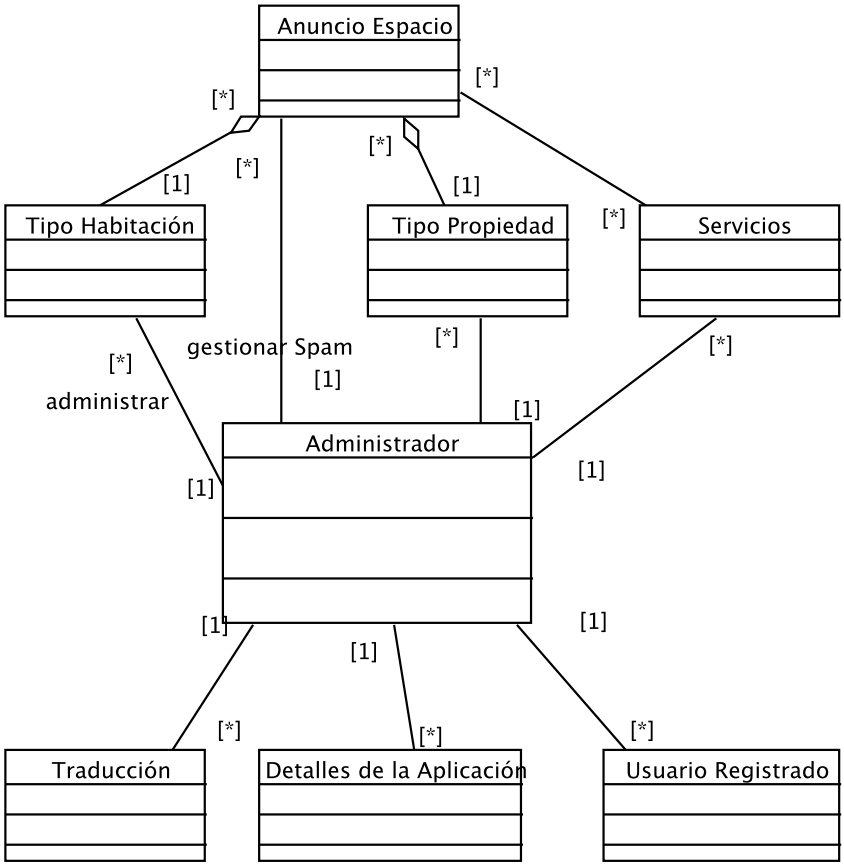

Figura 3.11: Diagrama clases Administrador.

En este diagrama se puede observar que el administrador gestiona los tipos de habitación, tipos de propiedad, servicios, traducciones y detalles de la aplicación existentes en la web.

Además gestiona los anuncios que son Spam, notifica a los usuarios que han creado estos anuncios y puede desactivar sus cuentas.

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. Autentificación Usuario

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

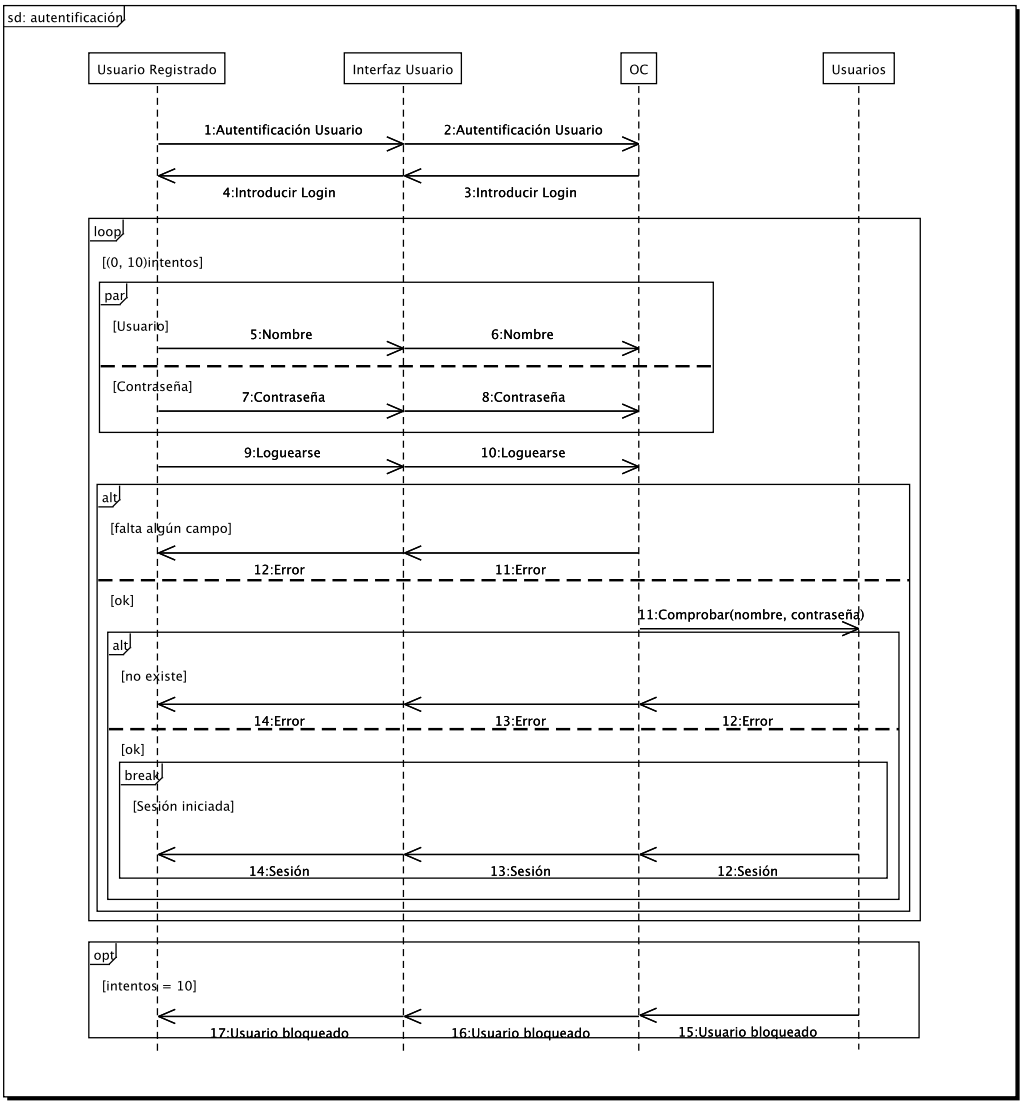

Figura 3.12: Diagrama secuencia Autentificación.

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.

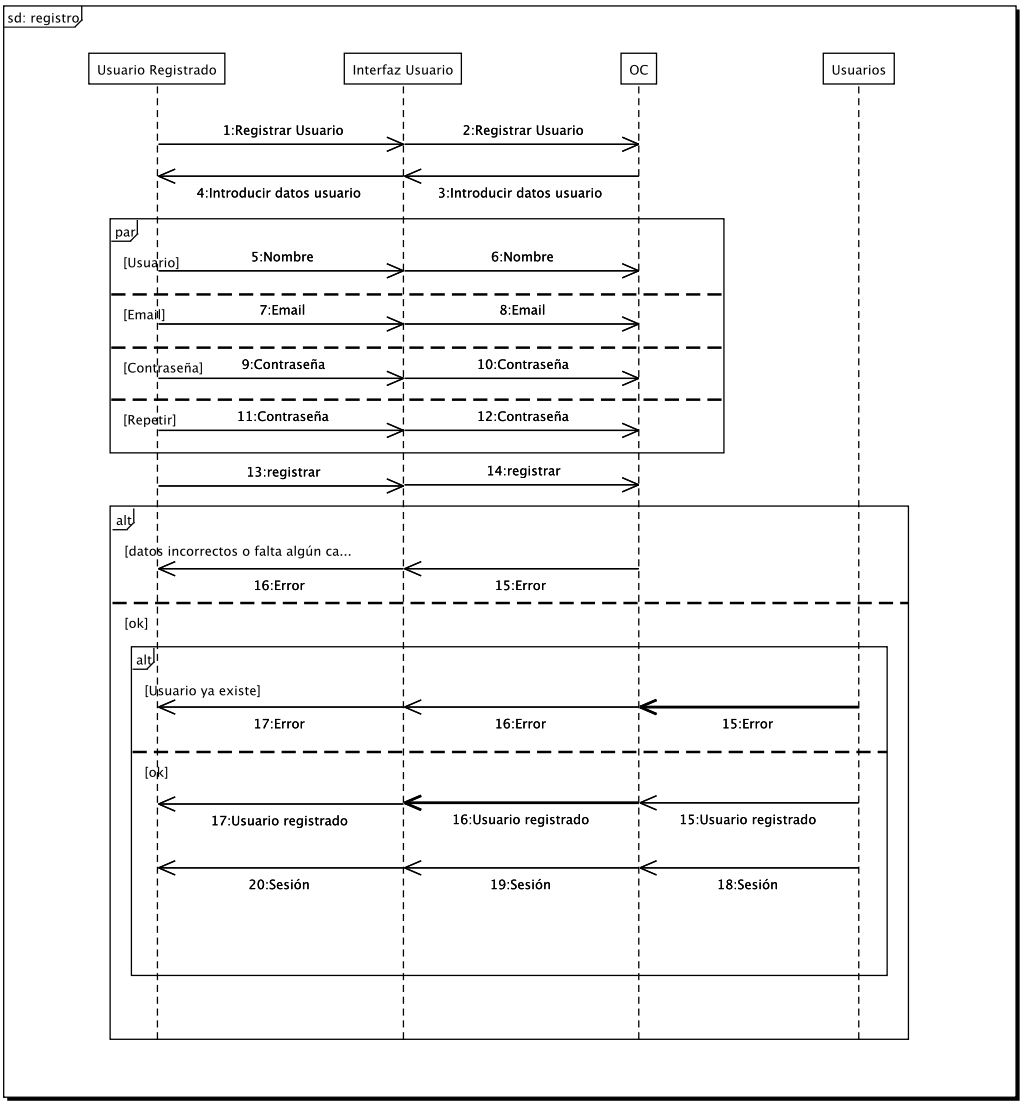

Figura 3.13: Diagrama secuencia Registro Usuario.

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.

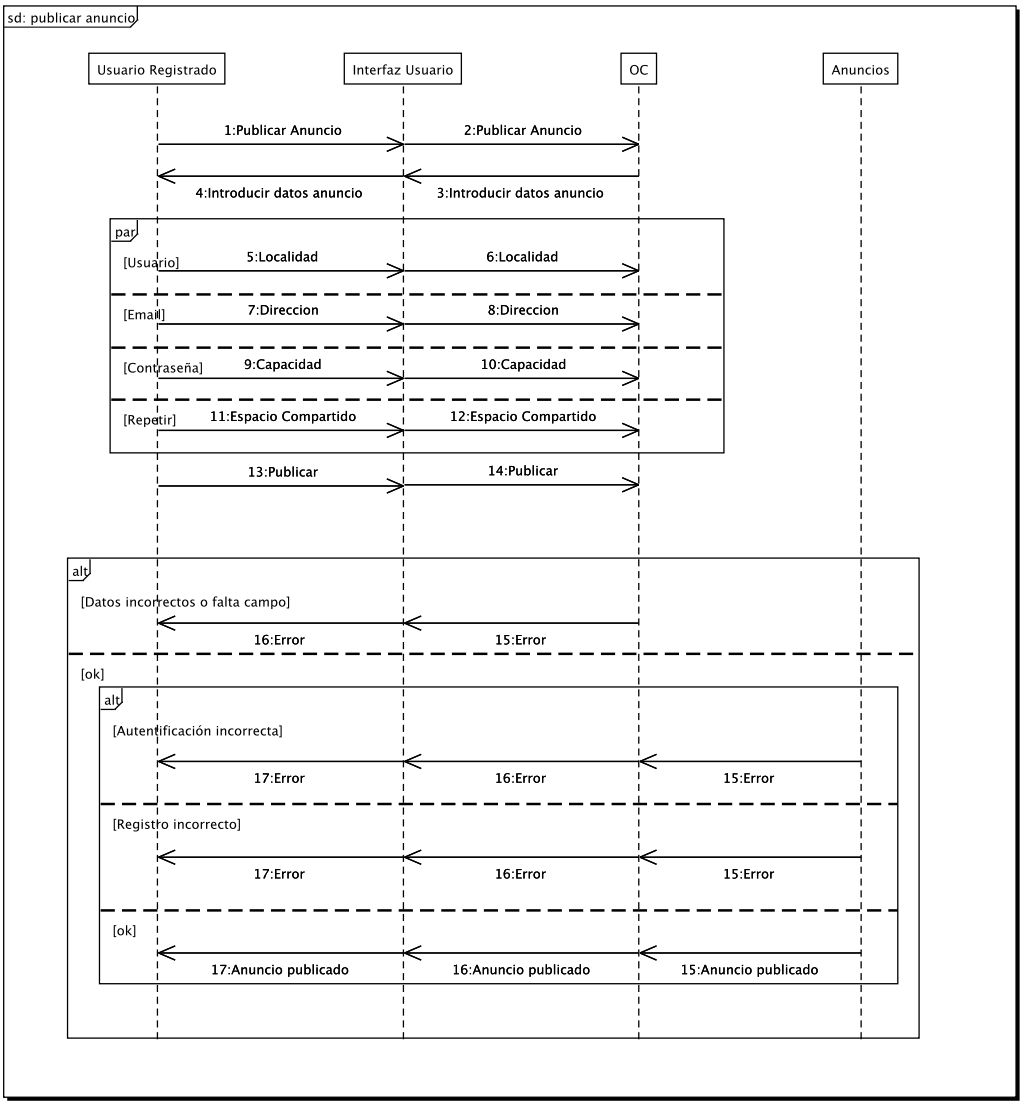

Figura 3.14: Diagrama secuencia Publicar Anuncio.

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.

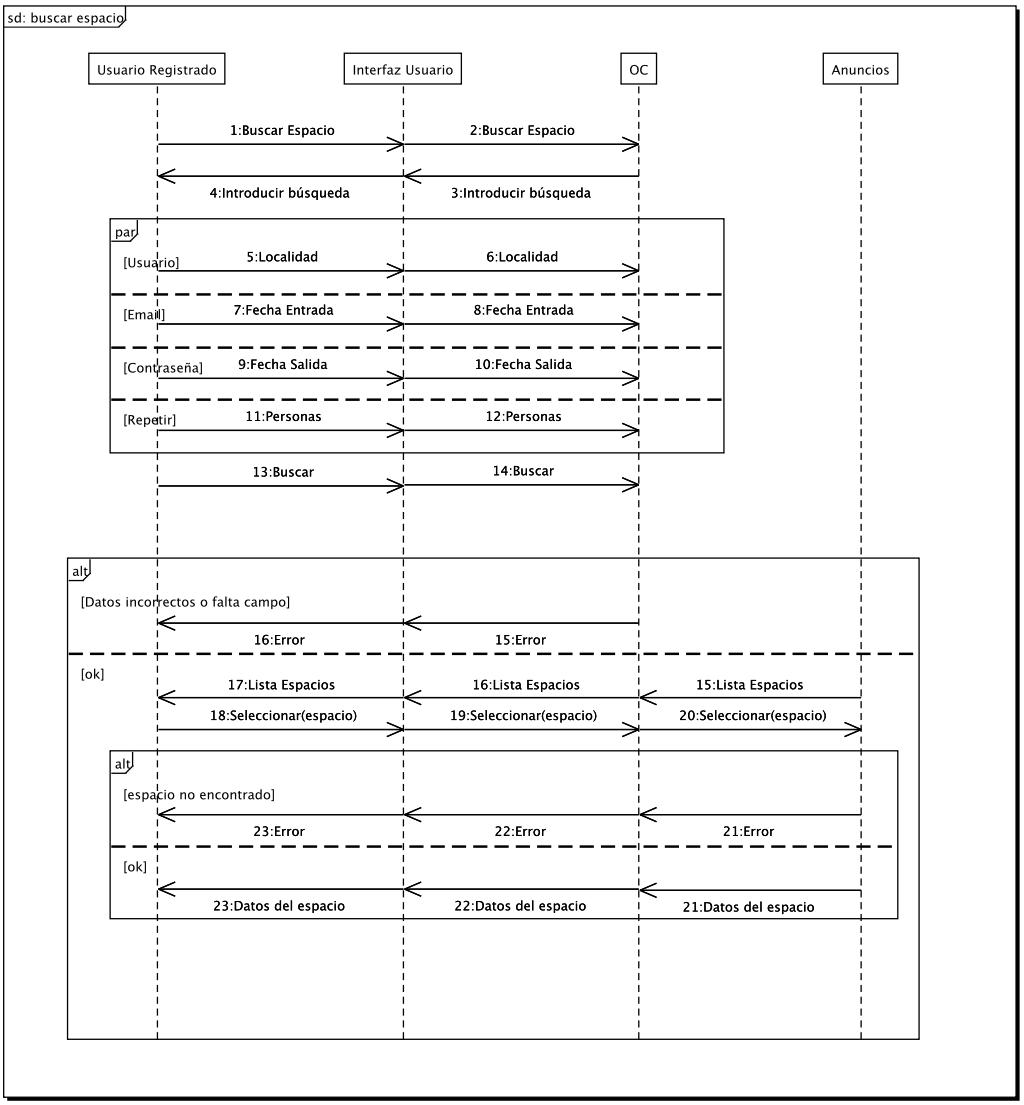

Figura 3.15: Diagrama secuencia Buscar y Seleccionar Espacio.

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.

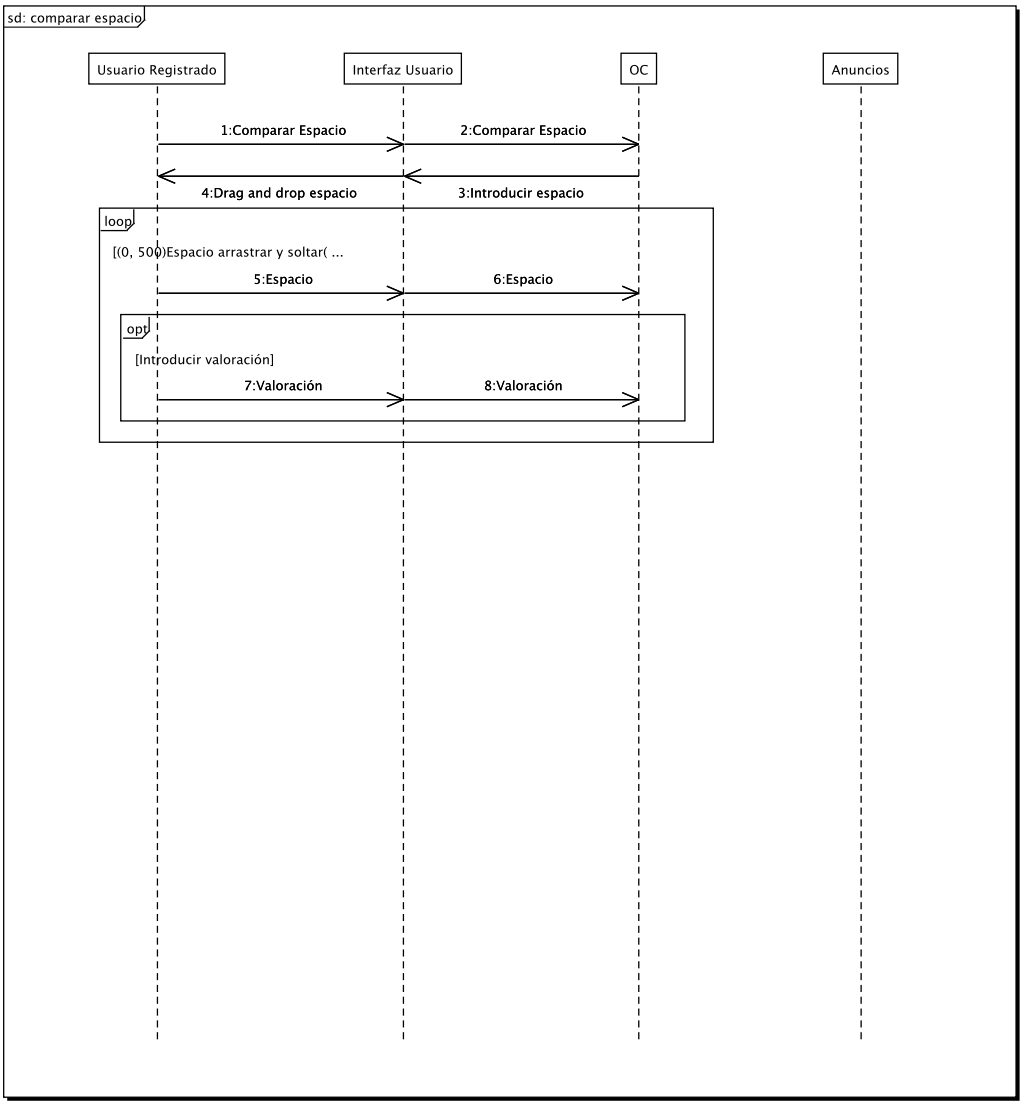

Figura 3.16: Diagrama secuencia Comparar Espacios.

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.

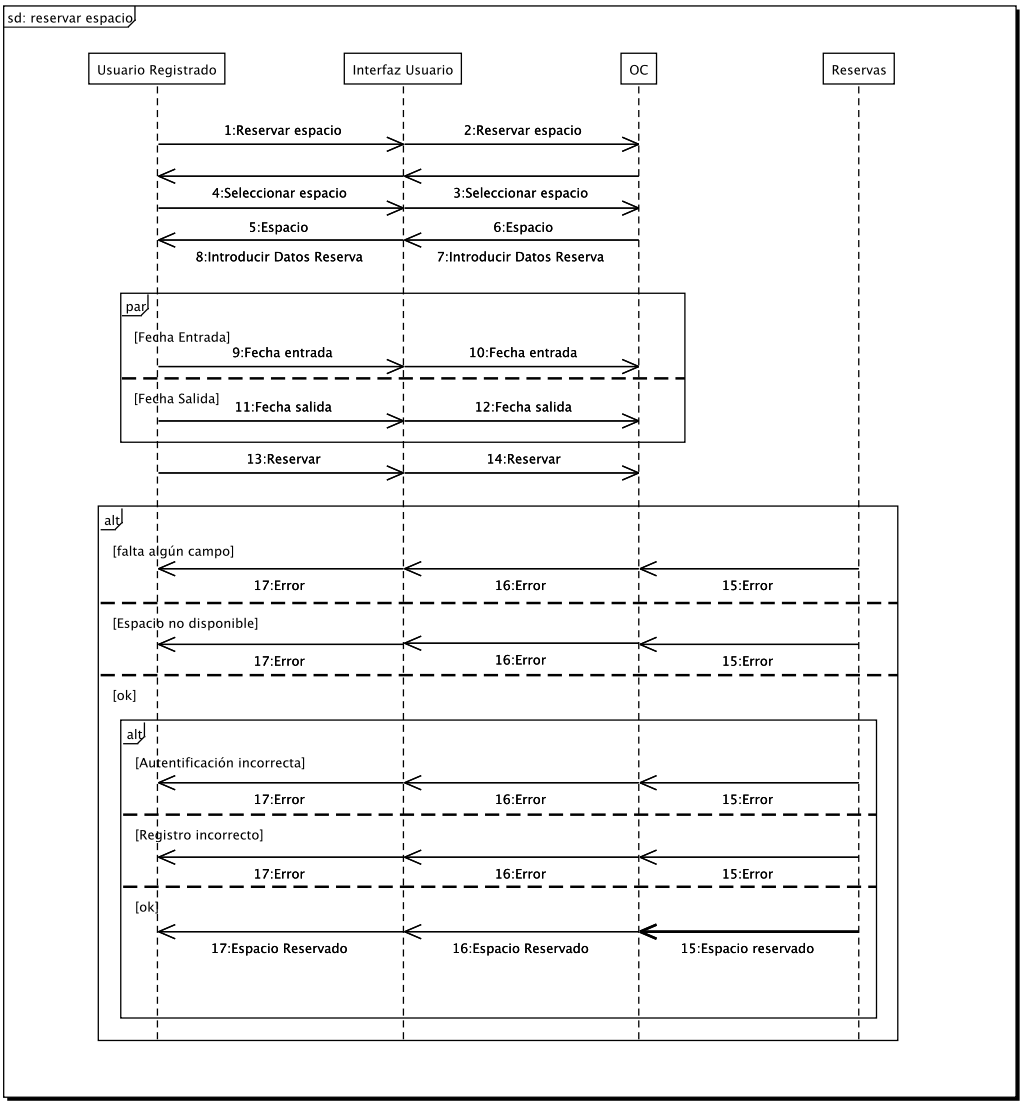

Figura 3.17: Diagrama secuencia Reservar Espacio.

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.

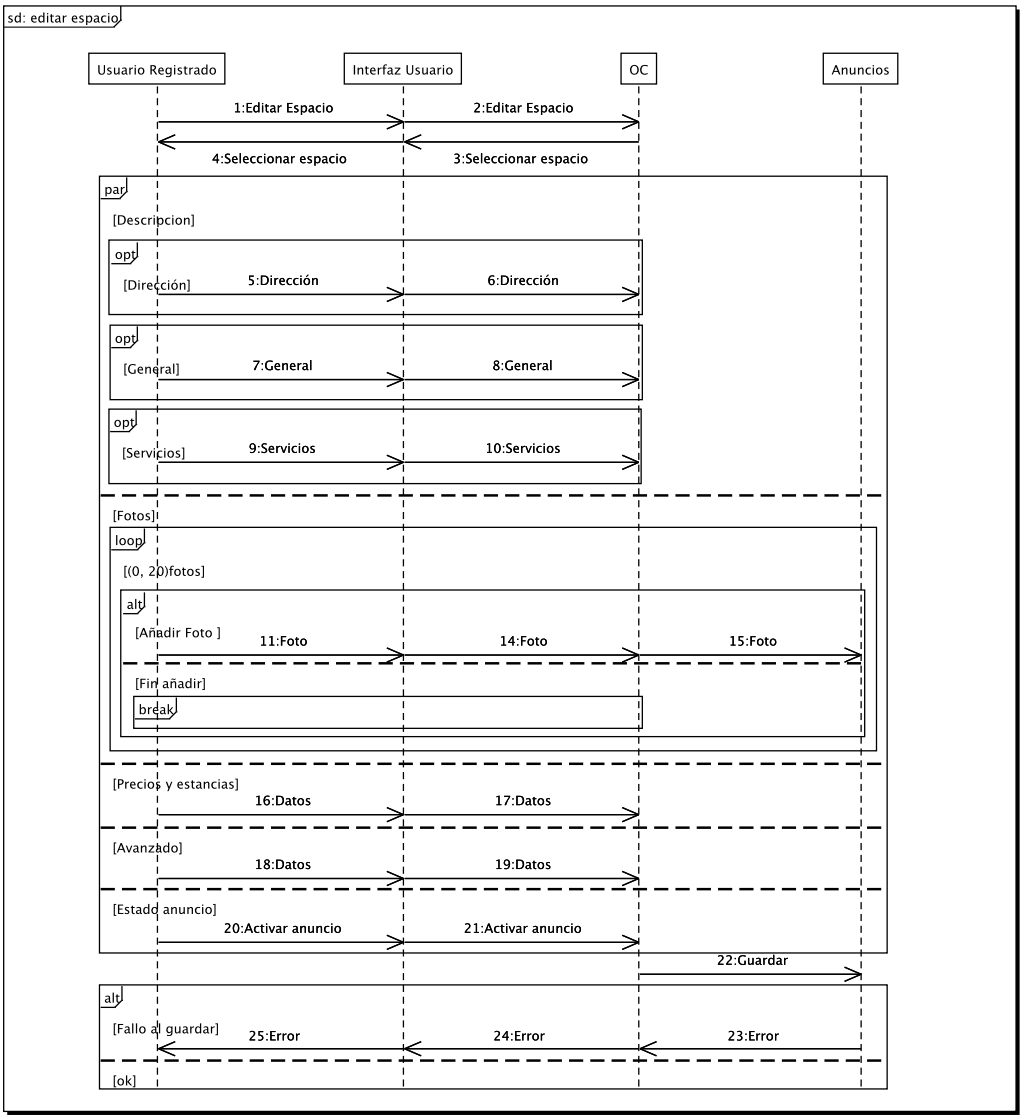

Figura 3.18: Diagrama secuencia Editar Espacio.

Capítulo 4

Diseño

4.1. Introducción

En el diseño de la aplicación web se ha utilizado el Modelo Vista Controlador (MVC)[59], que es un patrón de arquitectura del 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. El patrón MVC define la organización independiente del Modelo (Objetos del negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del flujo de trabajo de la aplicación).

Se ha utilizado este patrón de diseño porque tiene como ideas principales la reutilización de código y la separación de conceptos. Además, ha sido ampliamente utilizado y probado en entornos web.

4.2. Arquitectura del sistema

Como requisito de la aplicación, el servidor web debía ser PHP. Para conseguir esto y que además se utilice una arquitectura MVC, se ha optado por utilizar *Symfony 2*[41].

Symfony es un framework [54] PHP, que se utiliza para optimizar el desarrollo web, y está basado en el patrón Modelo Vista Controlador. Este entorno de desarrollo separa la lógica de negocio, la lógica del servidor y la presentación de la aplicación web, es decir, utiliza el patrón de diseño MVC. Además posee la mayor parte de herramientas necesarias para desarrollar la aplicación, y permite estructurar la aplicación de manera desacoplada. En la figura 4.1 se puede observar el patrón de diseño MVC utilizado en Symfony.

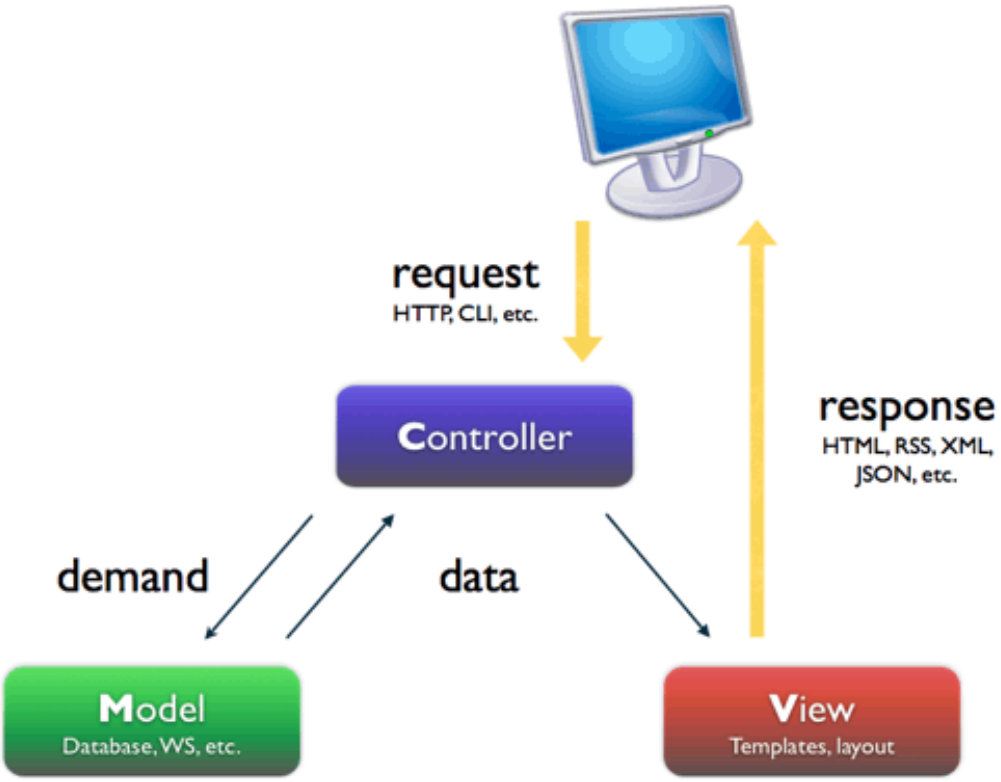

Figura 4.1: Patrón de diseño MVC de Symfony.

4.3. Modelo

El modelo es un objeto que representa datos o actividades. Se encarga de gestionar los accesos a la información y los privilegios de acceso. En Symfony el modelo se comunica con el Controlador. El controlador accede a los datos que gestiona el modelo. En la figura 4.2 se puede ver un ejemplo de un modelo en Symfony utilizado en la aplicación. En Symfony el modelo es denominado Entity.

```php
<?php

namespace Lyh\BuscadorBundle\Entity;

use Doctrine\ORM\Mapping as ORM;

use JMS\Serializer\Annotation\Type;

use JMS\Serializer\Annotation\Groups;
use JMS\Serializer\Annotation\Expose;
use JMS\Serializer\Annotation\VirtualProperty;
use JMS\Serializer\Annotation\SerializedName;

/**
 * Anuncios
 */
class Anuncios
{

    /**
     * @VirtualProperty
     * @Type("string")
     * @SerializedName("pais")
     * @Groups({"manage"})
     *
     * @return string
     */
    public function pais(){
        return $this->pais->setIdpais();
    }

    /**
     * @VirtualProperty
     * @Type("string")
     * @SerializedName("tipohabitacion")
     * @Groups({"manage"})
     */
    public function tipohabitacion(){
```

Figura 4.2: Modelo de la entidad Anuncios.

Para almacenar los datos y gestionarlos se ha utilizado un sistema de gestión de bases de datos [60] MySQL [19]. Este sistema aporta a la aplicación fiabilidad, seguridad, y es software libre. Además, implementa un amplio subconjunto del lenguaje SQL y permite la replicación de bases de datos. En la figura 4.3 se puede observar el esquema de la base de datos MySQL.

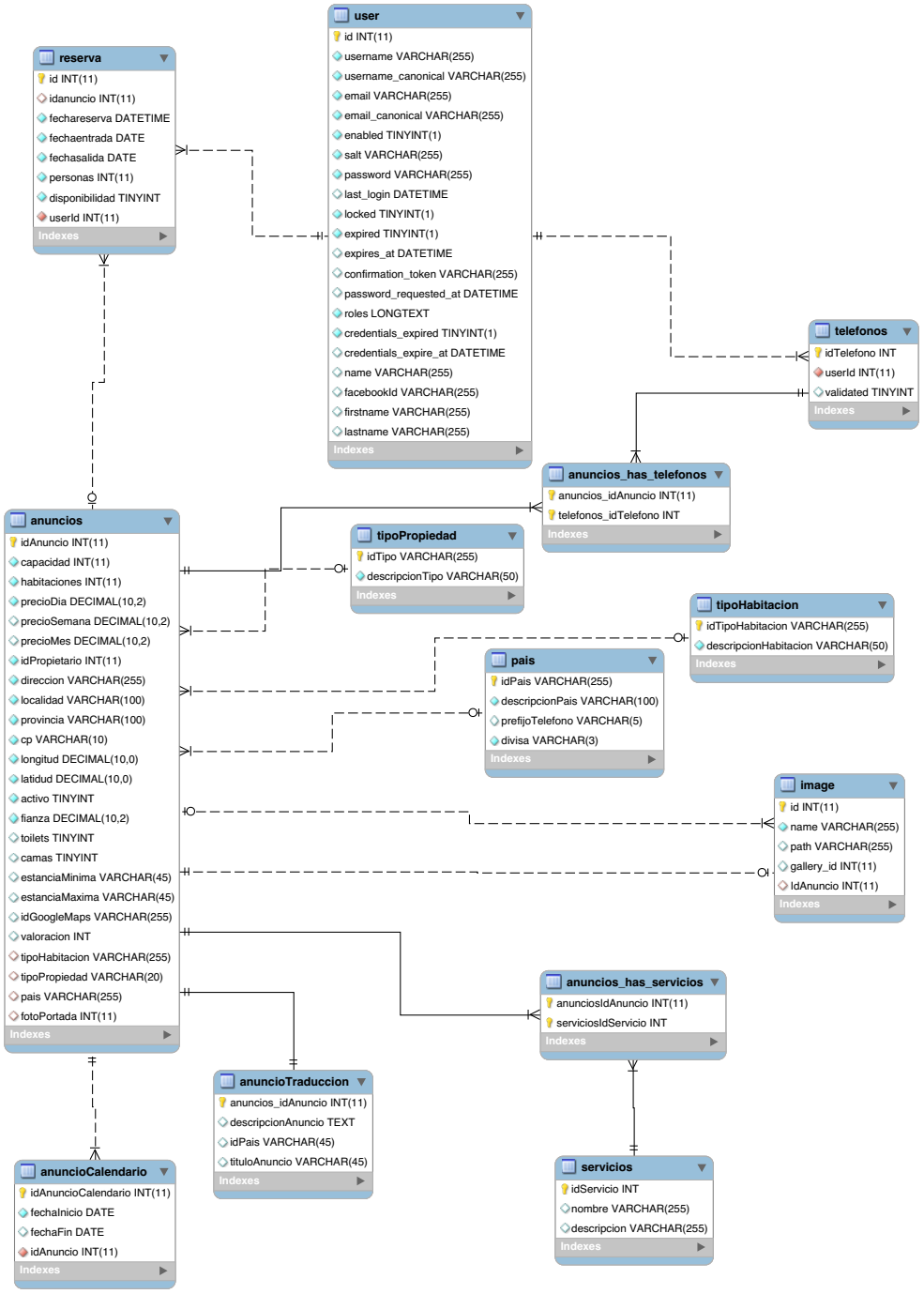

Figura 4.3: Esquema Base de Datos MySQL.

Para facilitar el acceso a los datos desde el framework PHP, se ha integrado una herramienta que permite el mapeo objeto-relacional, Object-Relational mapping (ORM) [58] en inglés, que es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. Se ha elegido en este caso el ORM

Doctrine [45] ya que se integra perfectamente con Symfony.

4.4. Vista

La vista presenta la información y la lógica de negocio mediante una interfaz de usuario. En Symfony la vista se comunica con el controlador. El controlador es el que se encarga de pasar los datos del modelo a la vista. La vista finalmente se envía al usuario para que puede interactuar con ella.

Symfony posee un motor de plantillas [61] para HTML llamado Twig [42]. Este motor de plantillas posee una sintaxis muy flexible y personalizable que permite trabajar muy fácilmente con él.Permiten definir variables, flujo de control, herencia entre plantillas y soporte para varios lenguajes. En la figura 4.4 se puede ver un ejemplo de un plantilla Twig.

```
{% extends app.request.isXmlHttpRequest ? "LyhAppBundle::ajax.html.twig" : "LyhAppBundle::base_main.html.twig" %}

{% block stylesheets %}
<link rel="stylesheet" type="text/css" media="screen" href="{{ asset('bundles/lyhmain/css/main.css') }}" />
<link rel="stylesheet" type="text/css" media="screen" href="{{ asset('bundles/lyhapp/css/mainbar.css') }}" />
<link rel="stylesheet" type="text/css" media="screen" href="{{ asset('bundles/lyhapp/css/calendar/jquery-ui-1.10.0.cust
{% endblock %}

{% block main %}
    <div id="fondo">
        <div id="main_bar">
        <text>
          {% trans %}make.yourself.at.home{% endtrans %}
        </text>
        <form id="main_nav" action="{{ path('lyh_buscador_homepage', {'_locale': locale }) }}">
             <div class="main_localidad"><input id="localidad" class="main_style" type="text" placeholder="{% trans %}wh
             <div class="main_input"><input id="fecha_inicio" class="main_style" type="text" placeholder="{% trans %}ch
             <div class="main_input"><input id="fecha_fin" class="main_style" type="text" placeholder="{% trans %}check-
             <div class="main_input"><select id="personas"  class="main_style" name="personas">
                                        <option value=1>1 {% trans %}person{% endtrans %}</option>
                                        <option value=2>2 {% trans %}persons{% endtrans %}</option>
                                        <option value=3>3 {% trans %}persons{% endtrans %}</option>
                                        <option value=4>4 {% trans %}persons{% endtrans %}</option>
                                        <option value=5>5 {% trans %}persons{% endtrans %}</option>
                                        <option value=6>6 {% trans %}persons{% endtrans %}</option>
                                        <option value=7>7 {% trans %}persons{% endtrans %}</option>
                                        <option value=8>8 {% trans %}persons{% endtrans %}</option>
                                        <option value=9>9 {% trans %}persons{% endtrans %}</option>
                                        <option value=10>10 {% trans %}persons{% endtrans %}</option>
                                    </select>
             </div>
             <div class="main_buscar"><input id="buscar"  type="submit" value="{% trans %}search{% endtrans %}"></div>
             <input id="inmueble" name="inmueble" type="text" value="" hidden>
             <input name="page" type="text" value="1" hidden>
             <input id="opcionmapaview" name="opcionmapaview" type="text" value="1" hidden>
             <input id="opciondetalle" name="opciondetalle" type="text" value="1" hidden>
             <input id="opcionprecio" name="opcionprecio" type="text" value="1" hidden>
```

Figura 4.4: Plantilla Twig base de la aplicación.

4.5. Controlador

El controlador es la parte más importante de la aplicación. En él está incluida la lógica de negocio y relaciona el Modelo con la Vista. El objectivo principal es devolver una respuesta a una petición de usuario, teniendo que realizar distintas acciones para generar esa respuesta.

El controlador de Symfony se encarga de recibir la información que proviene de la vista, de construir la respuesta, y de enviarla de nuevo a la vista. La respuesta puede ser muy variada, desde una página HTML, un documento XML [49], o un vector JSON [44] serializado, hasta una imagen, una redirección o una página de error. En la figura 4.5 se puede ver una ejemplo de un controlador de la aplicación.

```php
<?php

namespace Lyh\MainBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;


class DefaultController extends Controller
{
    public function indexAction()
    {
        $request = $this->getRequest();
        $locale = $request->getLocale();
        return $this->render(
                'LyhMainBundle:Default:index.html.twig',
                array('locale'=> $locale)
        );
    }
}
```

Figura 4.5: Controlador de una página simple de la aplicación.

Capítulo 5

Implementación

5.1. Introducción

En esta capítulo se detalla la implementación de la aplicación web. Primero se describen las distintas tecnologías utilizadas en el proyecto. También se describen las herramientas y el entorno utilizado para el desarrollo de la aplicación, y finalmente se explica el funcionamiento y la estructura de la aplicación web.

5.2. Tecnologías

En esta sección se describen las tecnologías que se han utilizado en el desarrollo de la aplicación. Hay lenguajes de programación, entidades externas, lenguajes y plantillas de marcado, un sistema de gestión de bases de datos [60], un ORM [58], una tecnología de comunicación y varios frameworks [54] de desarrollo web.

5.2.1. Symfony

Symfony [41] es un framework para aplicaciones web que ha sido diseñado para optimizar el desarrollo. Este framework intenta aliviar el exceso de carga asociado a actividades comunes en desarrollos web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo y automatiza las tareas más comunes. Symfony está desarrollado completamente con PHP 5 [34].

Symfony ha ayudado en el proyecto a estructurar correctamente la aplicación mezclando parte del framework y parte de desarrollo propio. Siguiendo su metodología se ha desarrollado toda la aplicación de manera desacoplada. De esta forma si se quiere modificar cualquier funcionalidad o reutilizarla, se puede hacer de manera muy sencilla.

Se han utilizado la mayoría de funcionalidades que ofrece symfony como son:

- **Bundles:** Un bundle es un directorio que almacena todo lo relacionado a una funcionalidad, incluyendo clases PHP, configuración, e incluso hojas de estilo y archivos JavaScript. Toda la aplicación se estructura en una serie de bundles que se detallan en la apartado 5.4.1 de la sección 5.4. Además se han utilizado una serie de bundles de terceros que añaden cierta funcionalidad al framework Symfony.

- **HTTP Cache:** Symfony utiliza un caché [51] que almacena documentos web para reducir el ancho de banda consumido, la carga de los servidores y el retardo en la descarga. Este caché almacena copias de los documentos que pasan por él, de forma que las subsiguientes peticiones pueden ser respondidas por el propio caché, si se cumplen ciertas condiciones. La caché que utiliza Symfony está basada en la especificación HTTP/1.1 [47].

- **Console:** Symfony facilita la creación comandos de línea de comandos [57] de manera sencilla y pudiendo ser probados. Estos comandos pueden ser usados para ejecutar una tarea periódica, automatizar tareas comunes o realizar otras tareas de la consola. A lo largo del proyecto se han creado varios comandos utilizando este componente de Symfony.

- **Controlador:** El controlador de Symfony es un componente del patrón de diseño MVC [59]. Se encuentra detallado en la sección 4.5, y es la parte más importante de la aplicación, ya que es donde se implementa la lógica de negocio. Durante el proyecto ha sido necesario implementar controladores que recibieran una petición HTTP y devolvieran una respuesta HTTP. La respuesta puede ser una página HTML, un documento XML [49], una colección JSON (apartado 5.2.11) serializada, una imagen, una redirección o una página de error.

- **Swiftmailer:** Aunque no forma parte del núcleo de Symfony, Swiftmailer [18] es una biblioteca escrita en PHP 5 para enviar emails que se integra a la perfección con Symfony. Swiftmailer permite enviar emails entre usuarios y desde la aplicación a distintos usuarios. Ha sido de gran utilidad en el proyecto porque abstrae de detalles a bajo nivel y permite centrarse en el desarrollo de la lógica de aplicación.

- **Eventos:** En Symfony hay una funcionalidad que se usa de manera conjunta con el Controlador: los eventos [11]. Con los eventos se pueden efectuar acciones justo antes o justo después de la ejecución de ciertas tareas en el controlador. El sistema de eventos ha sido muy útil en la implementación de las funcionalidades de autentificación de usuario, así como en tareas de persistencia en la base de datos ya que permiten realizar acciones una vez se haya guardado un objeto, evitando que en la base de datos queden datos erróneos.

- **Formularios:** Symfony integra un componente de formularios [5], que hace que tratar con formularios sea una tarea fácil. Permite crear un formulario desde PHP y generar automáticamente el código HTML necesario, la reutilización de formularios en varios bundles diferentes, la herencia entre formularios y la validación de formularios de forma sencilla. Además, es posible integrarlos con

las plantillas HTML y con el componente de la base de datos. También posee algunas algunas características de seguridad como protección contra Cross-site request forgery [53], abrevidado CSRF, que es un método por el cual un usuario malintencionado intenta hacer que los usuarios del sitio envíen datos que no pretenden transmitir.

- **Routing:** El componente Routing [6] de Symfony permite definir URLs visuales que se asignan a las diferentes áreas de la aplicación. A menudo se define una dirección web y se asigna a esa dirección un Controlador que se encarga de realizar la acción que esa dirección requiera.

- **Service Container:** El componente de Symfony, Service Container [8], ayuda a instanciar, organizar y recuperar muchos objectos de la aplicación. Este componente permite estandarizar y centralizar la manera en que los objetos son construidos en la aplicación. Se basa en el patrón de diseño Inyección de Depencencias [24], en inglés Dependency Injection, que es un patrón de diseño orientado a objetos en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree el objeto. Este componente facilita la estructura desacoplada de la aplicación y mejora la eficiencia de la misma.

- **Testing:** Symfony se integra con una librería independiente, llamada PHPUnit [15], para framework de pruebas enriquecido. Symfony permite hacer tanto pruebas unitarias[62], que permiten probar un módulo de código, como pruebas funcionales[55], que permite probar una determinada función de la aplicación siendo necesario, normalmente, probar varios módulos de de código.

- **Seguridad:** Symfony tiene un componente de Seguridad [7] que se encarga de prevenir a un usuario del acceso a un recurso que él no tiene permiso para acceder. Ha sido especialmente útil para la implementación de la funcionalidad de autentificación de usuario donde los protocolos de cifrado y varias tareas más ya estaban implementados por este componente.

- **Traducciones:** El componente de traducciones [9] de Symfony permite realizar el proceso de abstraer de la aplicación, las cadenas de texto y otras piezas específicas de la localidad y ponerlas en una capa donde puedan ser traducidas y convertidas según la configuración regional del usuario, es decir, el idioma y el país. En esta aplicación se ha utilizado para traducirla al francés, inglés y español dependiendo de la configuración del usuario.

- **Configuraciones:** Symfony utiliza unos archivos de configuración que son muy fáciles de manejar y con una sintaxis muy sencilla. Permite utilizar varios lenguajes para la configuración: YAML [21], XML y PHP. El lenguaje con una sintaxis más clara es YAML, el cual está diseñado teniendo muy en cuenta su legibilidad. En la aplicación se ha utilizado YAML para la configuración de todos los módulos, excepto para la base de datos en la que se ha utilizado XML debido a la mayor documentación encontrada de este último para la bases de datos.

- **Validación:** Symfony viene con un componente de validación [10] con el que se pueden validar los datos introducidos en los formularios. Los datos necesitan ser validados antes guardarlos en la base de datos o enviados a un servicio web. Se integra a la perfección con los formularios y también permite validar los datos fuera de ellos. Symfony trae consigo unas restricciones de validación básicas que permiten validar los tipos de datos básicos y más utilizados. Además permite crear restricciones propias mediante extensiones.

5.2.2. PHP 5

PHP [34] es un lenguaje de programación de de uso general de código del lado del servidor diseñado para el desarrollo web. Este lenguaje pude ser desplegado en la mayoría de servidores web y en casi todos los sistemas operativos y plataformas. PHP 5 es la última versión estable del lenguaje PHP.

En el proyecto se ha utilizado para desarrollar toda la parte Back-end alojada en el servidor web. Todos los módulos desarrollados con el framework Symfony han sido escritos en PHP. En el patrón de diseño MVC, PHP se ha utilizado para desarrollar los Controladores.

5.2.3. Twig

Twig [42] es un motor de plantillas para el lenguaje de programación PHP. Es un producto open source y Symfony viene con una extensión que lo utiliza como motor de plantillas por defecto.

Twig forma parte del desarrollo de parte Front-end de la aplicación. En este proyecto se ha utilizado de manera general y para realizar pruebas de los componentes de la aplicación.

5.2.4. HTML 5

HTML [46] es un lenguaje de marcado utilizado el desarrollo de páginas web. Se emplea para describir y traducir la estructura y la información en forma de texto y para añadir al texto objetos como por ejemplo imágenes.

En la aplicación se ha utilizado la versión 5 de html denominada HTML5. Esta versión todavía se encuentra en fase experimental y aún no ha sido reconocido como estándar, aunque muchos navegadores soportan la mayoría de funcionalidades no estandarizadas.

Aunque el trabajo con este lenguaje forma parte del Front-end de la aplicación, en este proyecto Back-end se ha empleado para realizar ciertas pruebas y esqueletos no finales de algunas de las páginas de la aplicación.

5.2.5. MySQL

MySQL [19] es un sistema de gestión de bases de datos [60] relacionales. Es software libre, por lo que permite la utilización y consulta de su código, aunque también existe una versión con licencia privativa de la empresa ORACLE [38]. Es el sistema de gestión de bases de datos más utilizado en aplicaciones web.

En este proyecto se ha utiliza MySQL para almacenar todos los datos de la aplicación que necesitasen persistencia. Se ha utilizado tanto en la parte del servidor web de producción como en el servidor de desarrollo.

5.2.6. Doctrine

Doctrine [45] es un mapeador de objetos-relacional [58], en inglés, Object-Relational mapping (ORM). Está implementado en PHP y proporciona una capa de abstracción al sistema de gestión de bases de datos.

Se ha utilizado integrado con el framework Symfony. Permite no tener que depender de una base de datos concreta, y si en el futuro se decidiera cambiar se podría hacer casi sin modificar los módulos de la aplicación. Se utiliza en el componente Modelo del patrón MVC [59].

5.2.7. JavaScript

JavaScript [36] es un lenguaje de programación interpretado. Se implementó pensando en la parte del cliente de los navegadores web para que pudieran interactuar con el usuario, comunicarse de manera asíncrona, controlar el navegador y alterar el contenido del documento que es mostrado. Es un dialecto del estándar ECMAScript [32].

El lenguaje JavaScript se utiliza en la parte del cliente, en concreto el navegador es quien lo interpreta y ejecuta. Es uno de los lenguajes más utilizados en la actualidad, y existen implementaciones que también lo utilizan en la parte del servidor, como Node.js [20].

Aunque es una tecnología asociada al Front-end, también se ha utilizado en este proyecto para desarrollar la funcionalidad básica de comunicación mediante AJAX, apartado 5.2.8, y para realizar diversas pruebas con la parte del servidor

5.2.8. AJAX

AJAX [28] es una técnica de desarrollo web para crear aplicaciones interactivas. Las aplicaciones con AJAX se ejecutan en el navegador de los usuarios, manteniéndose una comunicación asíncrona con el servidor. Ajax no es una tecnología por sí sola, sino que agrupa a varias tecnologías independientes. Las tecnologías que formas AJAX son:

- **XHTML y CSS** para la creación de una presentación basada en estándares.
- **DOM** para el manejo y modificación dinámica de la presentación.
- **XML y JSON** para representación de los datos en la comunicación.
- **XMLHttpRequest** para la comunicación asíncrona.
- **JavaScript** como lenguaje de programación.

Esta técnica se ha utilizado para mejorar la interacción con el usuario en muchas de las páginas de la aplicación. La parte del cliente se comunica con el servidor mediante JSON, haciendo necesario crear interfaces comunes con el Front-end para que la comunicación entre ambas partes fuese correcta y se pudiera probar.

5.2.9. RequireJS

RequireJS [17] es una librería para cargar módulos JavaScript. Está también escrita en este lenguaje y está optimizada para ser utilizada en navegadores web.

Aunque es un componente de la parte Front-End, se ha utilizado para estructurar y organizar los archivos JavaScript en módulos, debido a que hay una gran cantidad de éstos. Ha ayudado a optimizar también estos archivos, ya que integra una herramienta que permite la compresión, mejorando la respuesta de la aplicación ante peticiones HTTP.

5.2.10. Backbone.js

Backbone.js [13] es una librería JavaScript que da una estructura a la aplicación web proporcionando un patrón MVC [59] en el lado del cliente, es decir, en el navegador. Proporciona Modelos, Colecciones y Vistas, además de muchas funciones de gran utilidad para el desarrollo Front-end.

Aunque es también un componente asociado a la parte Front-end, se ha utilizado en este proyecto para proporcionar una comunicación con el servidor de manera que no haga falta esperar a que las funcionalidades de la parte Front-end estuvieran operativas. También ha permitido adelantar y probar ciertas partes en momentos clave, de forma que se ha podido acelerar el desarrollo de la aplicación.

5.2.11. JSON

JSON [44] es un formato ligero para el intercambio de datos. Es también, un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. Es ampliamente utilizado por la tecnología AJAX, detallada en el apartado 5.2.8, debido a su simplicidad e integración con el lenguaje JavaScript.

Se utiliza para la comunicación con la parte cliente (Front-End) de la aplicación web. Se ha creado una API JSON en el servidor basada en el estilo de arquitectura del software, REST [23], que permite realizar operaciones de consulta, actualización y borrado de objetos en el servidor. Se utiliza una interfaz HTTP con peticiones GET, POST, PUT, PATCH y DELETE [25], donde cada una realiza una acción diferente para el mismo objeto.

5.2.12. API Facebook

La Graph-API [22] de Facebook es una API [12] simple, basada en HTTP, que proporciona acceso al grafo social de Facebook, el cual representa de manera uniforme los objetos en el grafo y las conexiones entre ellos.

Esta API ha sido utilizada tanto en la parte Front-end como en el Back-end para realizar la funcionalidad de registrar y autentificar a un usuario mediante Facebook, y para que los usuarios compartan información con sus amigos en Facebook. Se ha hecho uso de la API para el lenguaje PHP en la parte del servidor, y de la API de JavaScript para la parte del cliente.

5.3. Herramientas y entorno

5.3.1. Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado [56], en inglés integrated development environment (IDE), es una aplicación software que ofrece facilidades a los programadores para el desarrollo de software. En él se integran varias herramientas, que de otra forma habría que utilizar dispersas en varios programas. Y Esto complica a veces el desarrollo cuando el proyecto es demasiado grande, por eso en este proyecto se ha optado por utilizar un IDE.

El IDE elegido ha sido Netbeans [35]. Netbeans es un entorno de desarrollo integrado libre para desarrollar con varios lenguajes entre ellos PHP, HTML5, Twig, YAML, JavaScript y CSS, es decir, soporta la totalidad de los lenguajes de programación utilizados en este proyecto y también los utilizados en la parte Front-end. También es multiplataforma y, en el caso de querer utilizar otro sistema operativo, no habría que aprender nada nuevo.

Antes de utilizarlo se han probado la mayoría de IDEs existentes, pero Netbeans ha sido el más convincente por su facilidad de uso y potencia. Además, posee herramientas de depuración de PHP y de integración con el framework Symfony por lo que es perfecto para este proyecto. En la figura 5.1 se muestra la interfaz gráfica de Netbeans.

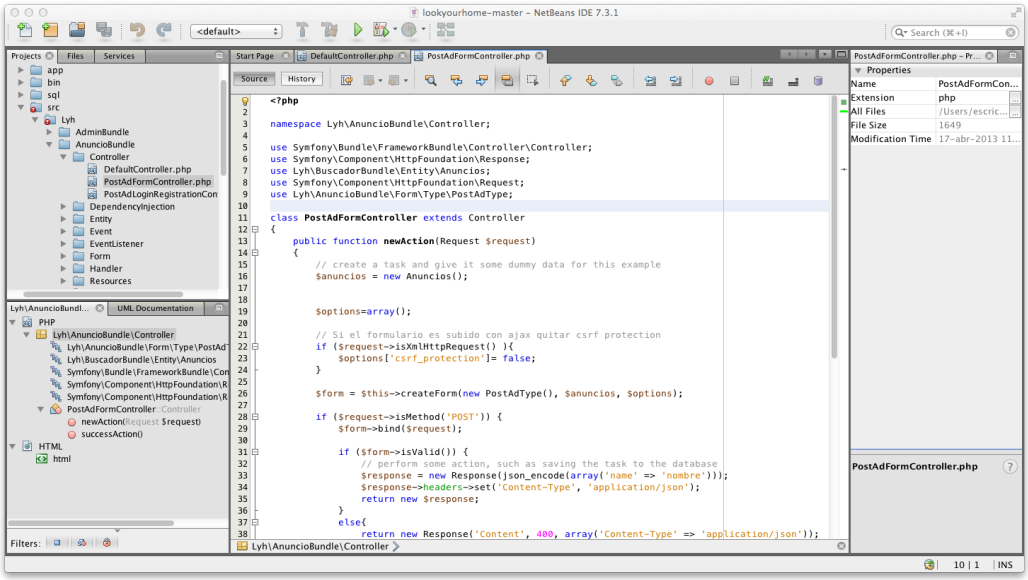

Figura 5.1: Interfaz gráfica de entorno de desarrollo integrado, Netbeans.

5.3.2. Servidor Web

Un servidor web es una aplicación informática que ayuda a distribuir el contendido web, que puede ser accedido a través de Internet. El protocolo de comunicación utilizado es HTTP.

En este proyecto se ha utilizado el servidor HTTP Apache. Apache es un servidor de código abierto que soporta la mayoría de plataformas, su desarrollo es profesional y es utilizado en una gran cantidad de sitios web. Consta de una serie de módulos que aportan funcionalidades adicionales, como por ejemplo el módulo de PHP llamado *mod_php*.

Para hacer uso del servidor Apache en un entorno local de desarrollo se ha utilizado XAMPP [26]. XAMPP es una aplicación que consiste principalmente en una base de datos MySQL, un servidor APACHE, y los intérpretes de PHP y Perl. También existen varios entornos que proporcionan esto, pero la ventaja principal de XAMPP frente a estos es que es multiplataforma. Facilita la instalación de estas herramientas y proporciona una interfaz gráfica de gran utilizad, como se ilustra en la figura 5.2.

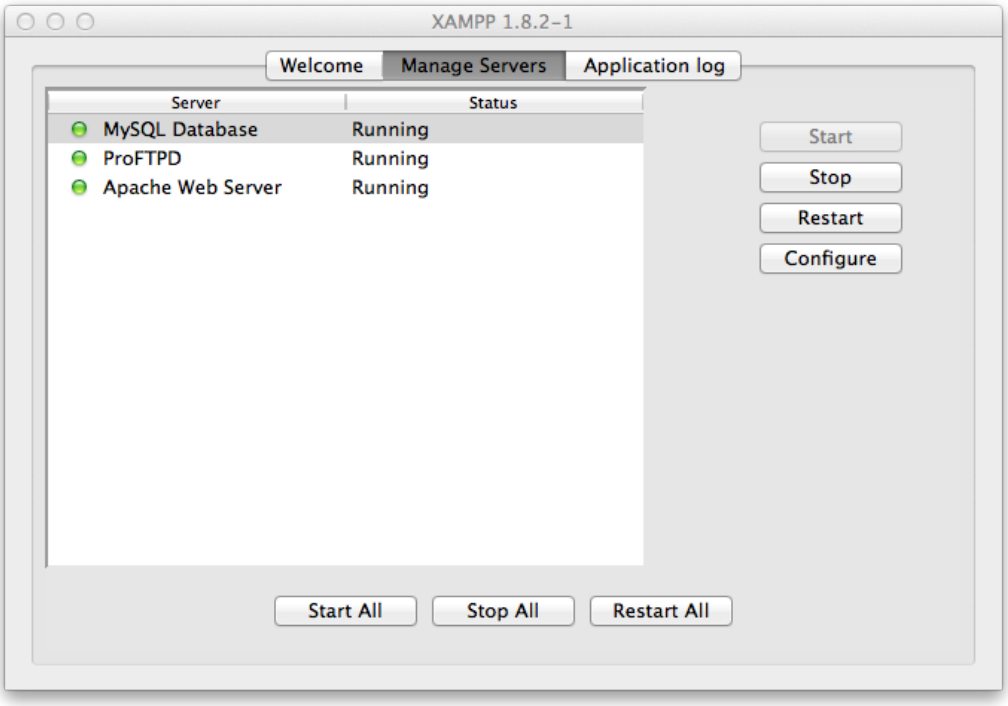

Figura 5.2: Interfaz gráfica de XAMPP

En cuanto al entorno de producción utilizado se ha optado por un hosting 1and1 [4]. Se ha optado por este alojamiento web por resultar ser económico y soportar todas las funcionalidades necesarias para la puesta en producción de la aplicación.

Para llevar la aplicación desde el entorno de desarrollo al entorno de producción de manera automática, se ha utilizado una herramienta de control de versiones, explicada en el siguiente apartado 5.3.3.

5.3.3. Herramientas de Control de Versiones

Una herramienta de control de versiones sirve para gestionar los cambios que se realizan sobre determinadas partes de la aplicación. Estas herramientas permiten automatizar las tareas de integración con distintos desarrolladores, y con ellas se puede llevar un seguimiento de todo lo que se ha desarrollado, así como volver a una versión anterior si fuera necesario.

Para este proyecto se ha utilizado un sistema de control de versiones llamado Git [43]. Este sistema está pensado para que varios desarrolladores trabajen de forma paralela sin tener que preocuparse de cómo se va a integrar el desarrollo. Ha aportado al proyecto velocidad y eliminación de tareas que se han automatizado. Se ha utilizado para la sincronización entre ambas partes del proyecto, Front-end y Back-end, y para llevar el proyecto a producción en el hosting 1and1.

Para la sincronización entre ambas partes del proyecto se ha utilizado GitHub [29], que es una aplicación web para alojar proyectos utilizando Git para el control de versiones. Permite que varios usuarios accedan al mismo repositorio y se puedan sincronizar. También se ha utilizado otra aplicación similar llamada Bitbucket pero después de probar ambas se decidió que GitHub funcionaba de mejor para este proyecto.

Para utilizar Git como control de versiones y sincronizar la parte Front-end con este proyecto se ha hecho uso del software llamado SmartGit. SmartGit es una aplicación que ofrece una interfaz gráfica al sistema de control de versiones Git, faciltando su uso. Aunque no es software libre, permite utilizarlo gratuitamente siempre que se utilice para fines no comerciales. En la figura 5.3 se muestra la interfaz gráfica de este programa.

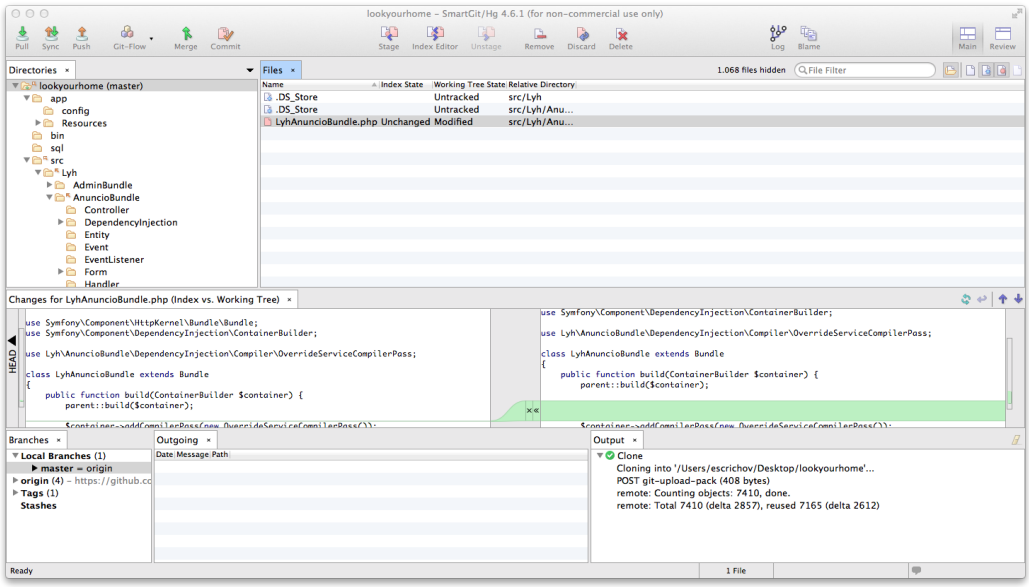

Figura 5.3: Interfaz gráfica de cliente para Git, SmartGit.

5.3.4. MySQL Workbench

MySQL Workbench [39] es un aplicación de diseño de bases de datos. Dispone de interfaz gráfica y con ella se puede administrar y desarrollar bases de datos MySQL.

En este proyecto se ha utilizado para diseñar la base de datos MySQL mediante un esquema de entidad relación que se muestra en la figura 4.3 del capítulo 4. En la figura 5.4 se muestra la interfaz gráfica de MySQL Workbench.

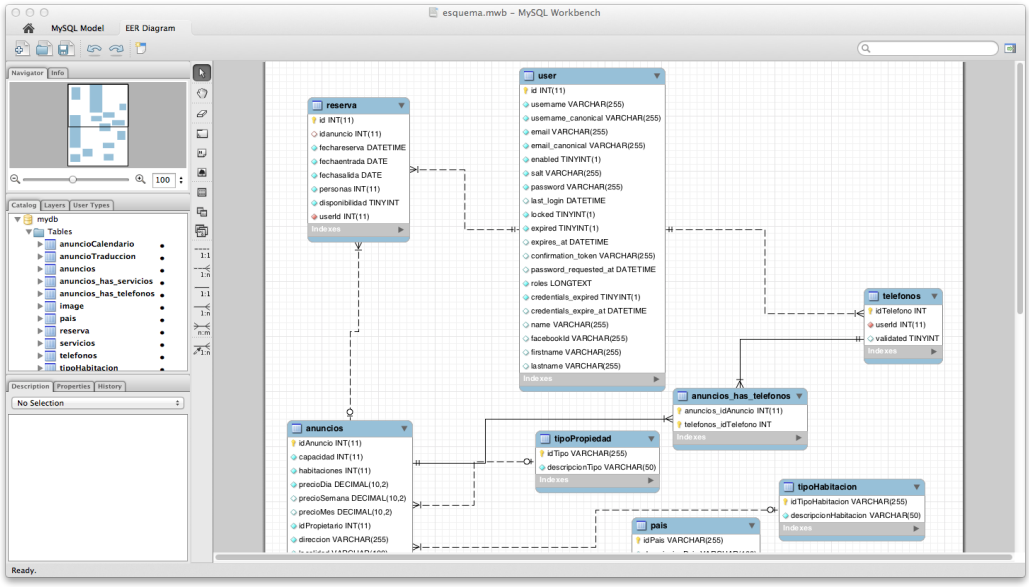

Figura 5.4: Interfaz gráfica de la aplicación MySQL Workbench.

5.3.5. Sistema de Gestión de Bases de Datos

Como sistema de gestión de bases de datos se ha utilizado MySQL, que se detalla en la apartado 5.2.5. Para administrar la base de datos MySQL se ha utilizado phpMyAdmin [40], que es una herramienta web utiliza para la gestión de bases de datos MySQL. Esta aplicación se ha utilizado tanto en el entorno de desarrollo como en el de producción. En la figura 5.5 se muestra la interfaz web de phpMyAdmin.

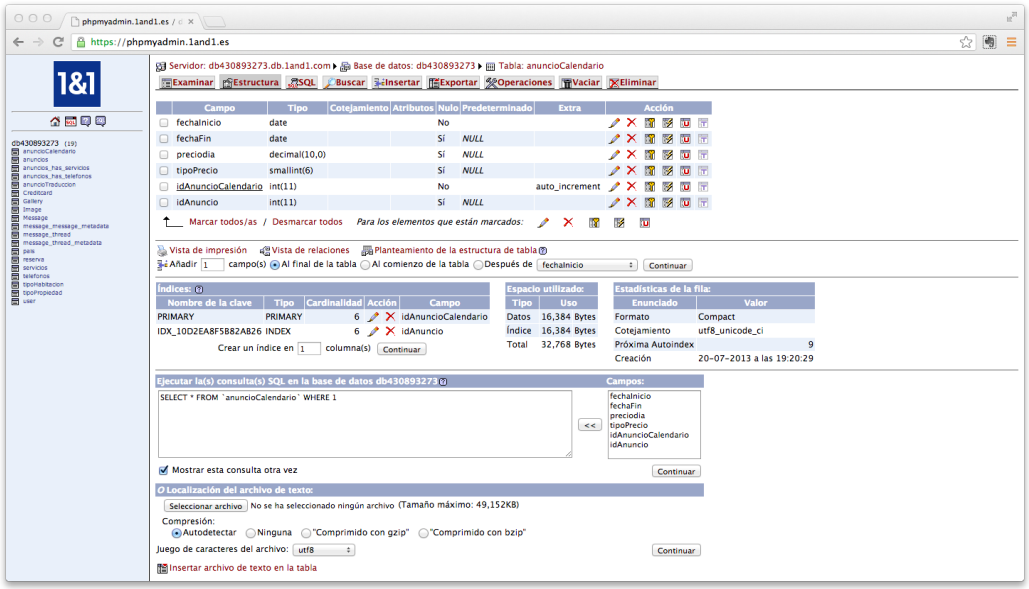

Figura 5.5: Interfaz gráfica de la aplicación web phpMyAdmin.

5.4. Descripción de la implementación

5.4.1. Estructura de la aplicación web

La aplicación web está organizada siguiendo una estructura desacoplada. Cada funcionalidad de la aplicación está contenida en una carpeta llamada Bundle. Un bundle es un directorio, perteneciente al framework Symfony, con una estructura bien definida y que almacena clases, controladores, pruebas, traducciones, comandos, configuración y recursos web relativos a una determinada funcionalidad del sistema.

Bundles de la aplicación

La aplicación web está distribuida en 14 Bundles de Symfony:

- **AdminBundle:** Bundle para administrar la aplicación web.
- **AnuncioBundle:** Bundle para crear un nuevo anuncio.
- **AnuncioManageBundle:** Bundle para gestionar los anuncios creados.
- **AppBundle:** Bundle para aportar los contenidos base que son idénticos en todas las páginas web.
- **BoilerplateBundle:** Bundle para gestionar los contenidos de ayuda de la aplicación web.
- **BuscadorBundle:** Bundle para gestionar la búsqueda y la visualización de toda la información pública de los espacios web.
- **CompartimosBundle:** Bundle para gestionar los anuncios de espacios compartidos.
- **IdiomaBundle:** Bundle para gestionar los idiomas de la web.
- **MainBundle:** Bundle para gestionar la página de inicio de la aplicación web.
- **MessageBundle:** Bundle para gestionar todos los mensajes que se envían a los usuarios.
- **PersonalBundle:** Bundle para gestionar el espacio personal de los usuarios.
- **ReservaBundle:** Bundle para realizar las reservas de los espacios anunciados.
- **UserBundle:** Bundle para administrar las cuentas de los usuarios y los procesos de autentificación y registro.

Bundles de terceros

Además se han utilizada una serie de bundles de terceros que han mejorado las herramientas que Symfony aporta. Todos estos bundles se pueden encontrar en Knp Bundles [33], una página que recoge multitud de bundles de terceros que se integran con Symfony. Los bundles utilizados son:

- **FOSUserBundle :** Bundle que añade soporte para el almacenamiento de cuentas de usuario en una base de datos.
- **FOSRestBundle:** Bundle que proporciona varias herramientas para el desarrolla rápido de API RESTful.
- **FOSFacebookBundle:** Bundle que proporciona autentificación vía Facebook.
- **PunkAveFileUploaderBundle:** Bundle que proporciona la carga de múltiples imágenes en la aplicación.
- **WebProfilerExtraBundle:** Bundle que añade funcionalidades de depuración al framework Symfony.
- **FormBundle:** Bundle que proporciona un campo identificador a los formularios de Symfony.
- **SimpleThingsFormSerializerBundle:** Bundle que ayuda a serializar los formularios de Symfony.
- **chromephp:** Librería que proporciona una extensión de Google Chrome para depurar PHP.
- **JMSSerializerBundle:** Bundle que integra una librería de serialización en Symfony.
- **NelmioApiDocBundle:** Bundle que permite generar información para las APIs.

Estructura de un bundle

Cada bundle está estructurado en las siguientes carpetas, aunque dependiendo de su funcionalidad puede que algunas de ellas no las utilice:

- **Controller:** Contiene los archivos PHP que sirven para gestionar las peticiones de páginas web que realizan los usuarios a través de la URL. El controlador se encuentra detallado en la sección 4.5.
- **DependencyInjection:** Contiene la configuración de los servicios que utiliza el bundle.
- **Entity:** Contiene las entidades, es decir, las clase básicas que contienen datos. En la figura 4.2 sección 4.3 está representada una entidad de la aplicación.

- **Event:** Contiene los eventos relacionados con la funcionalidad que implementa el bundle. En la figura 5.6 se muestra un evento utilizado en la aplicación.

```php
<?php

namespace Lyh\AnuncioBundle\Event;

use Symfony\Component\EventDispatcher\Event;
use Lyh\BuscadorBundle\Entity\Anuncios;

class AnuncioEvent extends Event
{
    //put your code here
    protected $anuncio;

    public function __construct(Anuncios $anuncio) {
        $this->anuncio = $anuncio;
    }

    public function getAnuncio(){
        return $this->anuncio;
    }
}

```

Figura 5.6: Evento relacionado con un Anuncio de un espacio.

- **EventListener:** Contiene los listeners, que son las funciones que se ejecutan cuando se produce un evento, relacionados con el propósito del bundle. Un ejemplo de Listener puede verse en la figura 5.7

```php
<?php

namespace Lyh\AnuncioBundle\EventListener;

use FOS\UserBundle\FOSUserEvents;
use FOS\UserBundle\Event\UserEvent;
use FOS\UserBundle\Event\FilterUserResponseEvent;
use FOS\UserBundle\Security\LoginManagerInterface;
use Symfony\Component\EventDispatcher\EventSubscriberInterface;
use Symfony\Component\Security\Core\Exception\AccountStatusException;

use Symfony\Component\HttpKernel\Exception\NotFoundHttpException;

use JMS\Serializer\SerializerBuilder;

use Lyh\AnuncioBundle\Event\AnuncioEvent;
use Lyh\AnuncioBundle\Event\PostAdEvents;

class AuthenticationListener implements EventSubscriberInterface
{
    private $loginManager;
    private $firewallName;

    public function __construct(LoginManagerInterface $loginManager, $firewallName)
    {
        $this->loginManager = $loginManager;
        $this->firewallName = $firewallName;
    }

    public static function getSubscribedEvents()
    {
        return array(
            FOSUserEvents::REGISTRATION_COMPLETED => 'authenticate',
            FOSUserEvents::REGISTRATION_CONFIRMED => 'authenticate',
            FOSUserEvents::RESETTING_RESET_COMPLETED => 'authenticate',
        );
```

Figura 5.7: Listener utilizado en la autentificación de usuario.

- **Form:** Contiene los formularios utilizados en las páginas web desarrolladas en este bundle. En la figura 5.8 se muestra un formulario implementado en la aplicación.

```php
<?php
namespace Lyh\AnuncioBundle\Form\Type;

use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\FormBuilderInterface;
use Symfony\Component\OptionsResolver\OptionsResolverInterface;

class TipoHabitacionType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder)
    {
        $builder->add('idtipohabitacion', 'choice', array(
            'choices' => array(
                'private_space' => 'privada',
                'shared_space' => 'compartida'
            )
        ));
    }

    public function setDefaultOptions(OptionsResolverInterface $resolver)
    {
        $resolver->setDefaults(array(
            'data_class' => 'Lyh\BuscadorBundle\Entity\TipoHabitacion',
        ));
    }

    public function getName()
    {
        return 'tipoHabitacion';
    }
}
```

Figura 5.8: Formulario del Tipo de Habitación de un espacio.

- **Resources:** Contiene todos los archivos necesarios para el funcionamiento de la parte Front-End. También contiene los archivos de configuración, de enrutado y de traducción escritos en YAML. En la figura 5.9 se puede observar un archivo de configuración que realiza la tarea de enrutar de un bundle.

```yaml
lyh_anuncio_homepage:
    pattern:   /form
    defaults: { _controller: LyhAnuncioBundle:PostAdLoginRegistration:PostAdLoginRegistration }

lyh_anuncio_form_success:
    pattern:   /form_success
    defaults: { _controller: LyhAnuncioBundle:PostAdForm:success }
```

Figura 5.9: Fichero de enrutado que relaciona un ruta de la aplicación con un Controlador.

- **Security:** Contiene archivos que implementan funcionalidades de autentificación y autorización de usuarios. Un archivo de ejemplo se puede ver en la figura 5.10

```php
<?php

namespace Lyh\UserBundle\Security\User\Provider;

use Symfony\Component\Security\Core\Exception\UsernameNotFoundException;
use Symfony\Component\Security\Core\Exception\UnsupportedUserException;
use Symfony\Component\Security\Core\User\UserProviderInterface;
use Symfony\Component\Security\Core\User\UserInterface;
use \BaseFacebook;
use \FacebookApiException;

class FacebookProvider implements UserProviderInterface
{
    /**
     * @var \Facebook
     */
    protected $facebook;
    protected $userManager;
    protected $validator;

    public function __construct(BaseFacebook $facebook, $userManager, $validator)
    {
        $this->facebook = $facebook;
        $this->userManager = $userManager;
        $this->validator = $validator;
    }

    public function supportsClass($class)
    {
        return $this->userManager->supportsClass($class);
    }

    public function findUserByFbId($fbId)
    {
        return $this->userManager->findUserBy(array('facebookId' => $fbId));
    }
```

Figura 5.10: Fichero de que permite la autentificación de usuario mediante la API de Facebook [22].

- **Tests:** Contiene las pruebas unitarias que prueban el código contenido en este bundle. También puede contener las pruebas funcionales que en este caso prueban contenido de uno o más bundles. Las pruebas se encuentran detalladas en la sección 6.2.
- **Validator:** Contiene restricciones de validación creadas extendiendo las básicas de Symfony. En la figura 5.11 se encuentra un ejemplo de restricción.

```php
<?php

namespace Lyh\UserBundle\Validator\Constraints;

use Symfony\Component\Validator\Constraint;
use Symfony\Component\Validator\ConstraintValidator;

class NotEmailValidator extends ConstraintValidator{

    public function validate($value, Constraint $constraint)
    {
        if (null === $value || '' === $value) {
            return;
        }

        if (!is_scalar($value) && !(is_object($value) && method_exists($value, '__toString'))) {
            throw new UnexpectedTypeException($value, 'string');
        }

        $value = (string) $value;
        $valid = filter_var($value, FILTER_VALIDATE_EMAIL);

        if ($valid) {
            $host = substr($value, strpos($value, '@') + 1);

            if (version_compare(PHP_VERSION, '5.3.3', '<') && strpos($host, '.') === false) {
                // Likely not a FQDN, bug in PHP FILTER_VALIDATE_EMAIL prior to PHP 5.3.3
                $valid = false;
            }

            // Check for host DNS resource records
            if ($valid && $constraint->checkMX) {
                $valid = $this->checkMX($host);
            } elseif ($valid && $constraint->checkHost) {
                $valid = $this->checkHost($host);
            }
        }
```

Figura 5.11: Restricción que comprueba que una cadena dada no es un email.

5.4.2. Funcionamiento y servicios de la aplicación

Flujo de la aplicación

El flujo que sigue la aplicación se basa en las acciones que ésta realiza desde que el usuario realiza una petición, hasta que el sistema genera la respuesta correspondiente a esa acción. La aplicación desarrollada en este proyecto sigue el flujo representado en la figura 5.12.

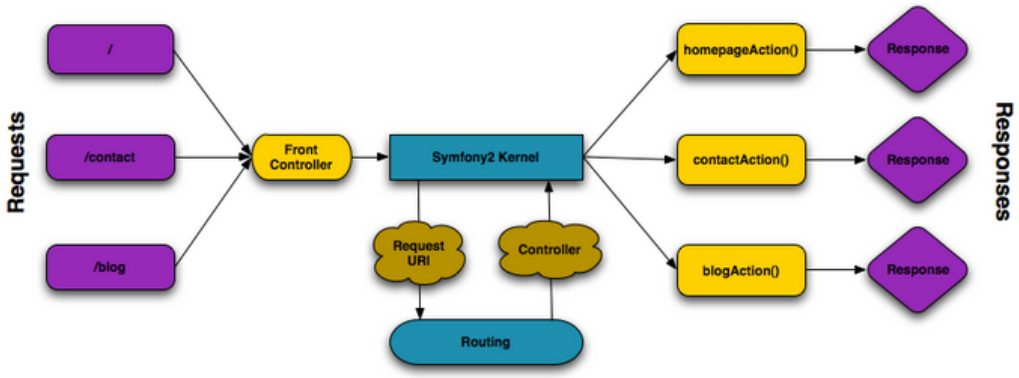

Figura 5.12: Flujo que sigue la aplicación cuando llega un petición de usuario.

Este flujo comienza recibiendo una petición de usuario que llega a través de una petición HTTP. Esta petición está asociada a una ruta de la aplicación web. Después de recibir la petición, el framework Symfony se encarga de relacionar la ruta con un Controlador, donde esta relación ha sido especificada previamente por el desarrollador. Una vez encontrado el Controlador correspondiente, Symfony le pasa al Controlador los datos de la petición y todos los servicios que haya especificado como necesarios. El controlador, que se encuentra explicado con mayor detalle en la sección 4.5, se encarga de efectuar toda la lógica de negocio necesaria, incluyendo el acceso a la base de datos y la gestión de ficheros, y finalmente genera una respuesta HTTP. Esta respuesta puede ser una vista HTML, una colección JSON o XML o una página de error.

Servicios

La aplicación realizada en este proyecto tiene multitud de tareas que se utilizan en múltiples Bundles, detallados en el apartado 5.4.1, y dentro de cada bundle también se utilizan en varios controladores. Para seguir la estructura desacoplada de los bundles y reutilizar la mayor parte de código se utilizan los Servicios [8].

Un servicio es un objeto PHP que desempeña algún tipo de tarea global. Por ejemplo acceder a la base de datos, enviar un email o validar un determinado formulario.

Cada servicio es usado en la aplicación solo cuando se necesita su funcionalidad. Symfony se encarga de instanciarlos cuando es necesario. Según su ámbito, es decir, según el tiempo que es usado el servicio, puede ser de dos tipos: contenedor y prototipo. El servicio del tipo contenedor es instanciado una sola vez a lo largo de la aplicación y se usa cada vez que se necesita, mientras que un servicio del tipo prototipo se crea que cada vez que algún objeto lo necesite.

Capítulo 6

Evaluación

6.1. Introducción

En esta sección se han realizado varias pruebas para evidenciar que el código producido en la implementación es correcto. Además se hace un análisis de la aplicación desde el punto de vista de los usuarios y de cómo se utiliza. No se ha realizado un validación de los estándares HTML, CSS y JavaScript ya que esto corresponde a la parte Front-end del proyecto.

6.2. Pruebas

6.2.1. Pruebas unitarias

En la aplicación se han realizado pruebas unitarias [62] utilizando la librería de PHP, PHPUnit [15]. Una prueba unitaria es una prueba específica que se hace contra una clase PHP. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. En la figura 6.1 se puede observar un ejemplo de prueba unitaria utilizado en la aplicación.

Los beneficios obtenidos en este proyecto al realizar las pruebas unitarias han sido: encontrar problemas de forma temprana en el desarrollo, documentación del código, mejorar el diseño y encontrar errores ocultos en el código.

```php
<?php

namespace Symfony\Bundle\FrameworkBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Tests\TestCase;
use Symfony\Bundle\FrameworkBundle\Controller\ControllerNameParser;
use Symfony\Component\ClassLoader\ClassLoader;

class ControllerNameParserTest extends TestCase
{
    protected $loader;

    protected function setUp()
    {
        $this->loader = new ClassLoader();
        $this->loader->addPrefixes(array(
            'TestBundle'      => __DIR__.'/../Fixtures',
            'TestApplication' => __DIR__.'/../Fixtures',
        ));
        $this->loader->register();
    }

    public function tearDown()
    {
        spl_autoload_unregister(array($this->loader, 'loadClass'));

        $this->loader = null;
    }
```

Figura 6.1: Prueba unitaria del framework Symfony utilizada en la aplicación.

6.2.2. Pruebas funcionales

Además de las pruebas unitarias en este proyecto se han utilizado pruebas que evidencian el correcto comportamiento de las funciones de la aplicación. Estas pruebas son las pruebas funcionales [55].

Las pruebas funcionales comprueban la integración de las diferentes capas de la aplicación, desde las rutas hasta la representación de las vistas. Tienen un flujo especifíco.

El flujo que siguen las pruebas funcionales es el siguiente:

- Primero se hace una petición a la aplicación web.
- Después la prueba que la respuesta obtenida es la correcta.
- Cuando se verifica que la respuesta es la correcta, se prueba a hacer una acción de usuario.
- Una vez realizada la acción de usuario se comprueba que la respuesta obtenida por el sistema es la correcta.
- Finalmente se depuran y corrigen los errores cometidos y se repite el proceso

En la aplicación se han probado las diversas funcionalidades de la especificación de requisitos. En la figura 6.2 se puede ver un ejemplo de prueba funcional utilizado en la aplicación web.

```php
<?php

namespace Symfony\Bundle\FrameworkBundle\Tests\Functional;

/**
 * @group functional
 */
class SessionTest extends WebTestCase
{
    /**
     * Tests session attributes persist.
     *
     * @dataProvider getConfigs
     */
    public function testWelcome($config, $insulate)
    {
        $client = $this->createClient(array('test_case' => 'Session', 'root_config' => $config));
        if ($insulate) {
            $client->insulate();
        }

        // no session
        $crawler = $client->request('GET', '/session');
        $this->assertContains('You are new here and gave no name.', $crawler->text());

        // remember name
        $crawler = $client->request('GET', '/session/drak');
        $this->assertContains('Hello drak, nice to meet you.', $crawler->text());

        // prove remembered name
        $crawler = $client->request('GET', '/session');
        $this->assertContains('Welcome back drak, nice to meet you.', $crawler->text());
```

Figura 6.2: Prueba funcional del framework Symfony utilizada en la aplicación.

6.3. Rendimiento del segmento de usuarios

Para la validación del comportamiento de los usuario, navegadores y datos estadísticos de acceso, se ha utilizado una herramienta llamada Google Analytics [30]. Con este servicio gratuito es posible obtener estadísticas de la aplicación web agrupada por diversas categorías.

Visitas

Primero se han analizado los datos pertenecientes a las visitas de los usuarios a la aplicación. Se han obtenido las siguientes gráficas, figura 6.3, que muestran: el número de visitas a la aplicación, datos demográficos de países dónde se accede a la página, el número de visitantes únicos, las visitas por Navegador, la duración media de las visitas y las páginas medias vistas en cada visita.

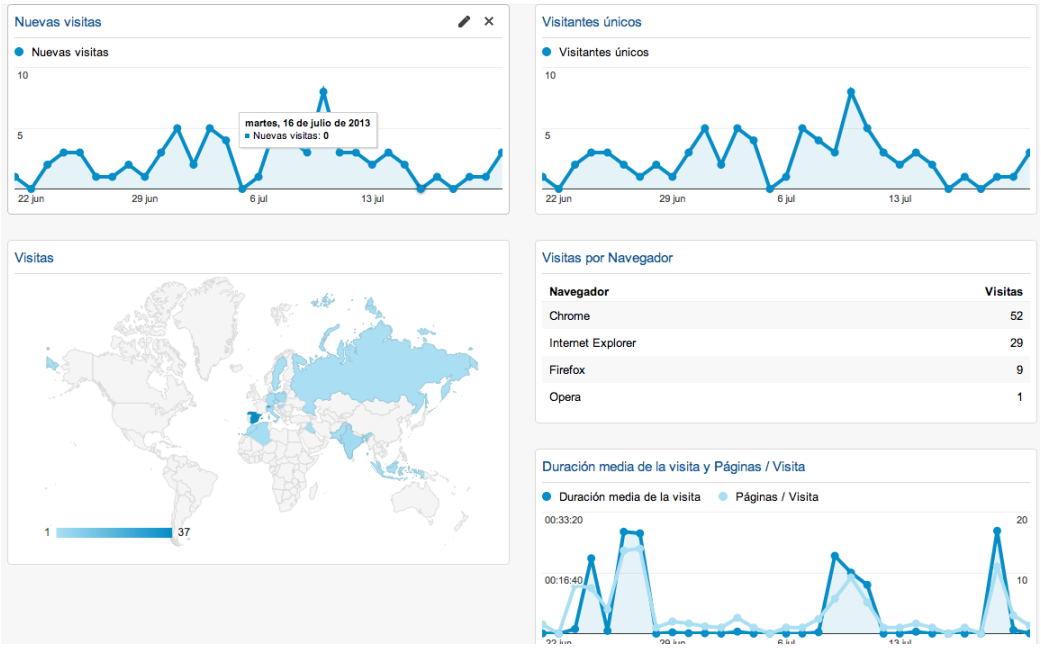

Figura 6.3: Visión general de público de la aplicación.

Comportamiento de los usuarios

Utilizando el mismo programa, Google Analytics, también se ha analizado el flujo de comporatamiento que siguen los usuario en la aplicación. En la siguiente figura 6.4 se pueden ver las interacciones que un usuario hace con la página. Primero se muestra la página por la que accede el usuario a la aplicación y, en las siguientes interaciones, se muestra el comportamiento que ha seguido el usuario en la aplicación.

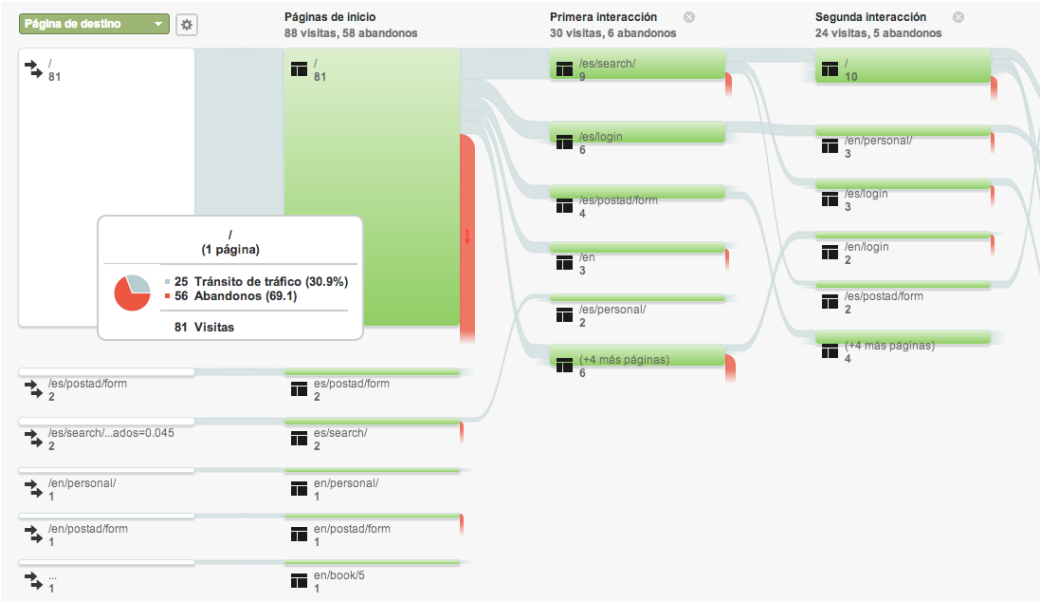

Figura 6.4: Flujo de comportamiento de los usuario en la aplicación.

Capítulo 7

Conclusiones

7.1. Resumen

En este proyecto se ha desarrollado una aplicación web de alquiler de espacios. El proyecto ha comprendido las etapas de especificación de requisitos, análisis, diseño, implementación y evaluación del trabajo realizado.

A lo largo del proyecto se ha logrado cumplir con los objetivos marcados inicialmente. Estos objetivos se han planificado de manera conjunta con la parte Front-end del proyecto. Además, cada cierto tiempo se ha puesto en común el trabajo realizado por la parte Front-end y el de éste proyecto, haciendo necesario a veces modificar ciertos apartados de la aplicación web.

Durante la realización del proyecto se ha tenido que consultar multitud de documentación de las diversas herramientas utilizadas, lo que ha supuesto un ligero retraso en el desarrollo de algunas de las funcionalidades. A medida que ha ido avanzando el proyecto se ha notado cierta soltura con las herramientas utilizadas y mayor rapidez de desarrollo.

Las herramientas utilizadas en el proyecto han sido de gran ayuda al resultar ser maduras y de gran calidad. En múltiples ocasiones han ahorrado mucho tiempo de desarrollo y han permitido estructurar la aplicación web de manera correcta. Esta forma de estructurar la aplicación permite que el mantenimiento sea poco costoso.

Finalmente, después de mucho esfuerzo, se ha podido conseguir una aplicación web utilizando las últimas tecnologías estables de desarrollo y con una interfaz bien definida en la que prima la experiencia de usuario.

7.2. Valoración

Partiendo de los conocimientos previos adquiridos durante la titulación de Ingeniería Informática, se han adquirido una serie de habilidades adicionales durante la realización de este proyecto. Aquí se han podido aplicar parte de los estudios de la carrera, y éstos han servido además para aprender a trabajar con nuevas tecnologías que tienen su base en conceptos previamente estudiados.

En las etapas tempranas del proyecto ha sido necesario buscar mucha información sobre las herramientas y el framework Symfony utilizado en el diseño e implementación de la aplicación. La información encontrada ha sido de muy buena calidad, sobre todo en la página oficial de Symfony [41] y en comunidades de desarrolladores como Stack Overflow [14]. Además, ha habido un inicio complicado del desarrollo y ha hecho falta un periodo de aprendizaje y adaptación ante tantas tecnologías nuevas.

Cabe destacar que ha sido una experiencia muy enriquecedora el solventar los problemas que han ido surgiendo y ver como la aplicación ha ido evolucionando desde algo que parecía sencillo y muy poco profesional hasta obtener una aplicación web con las últimas tecnologías.

7.3. Trabajo futuro

Como trabajo futuro que ha quedado por desarrollar está el desarrollo de una aplicación para dispositivos móviles que se integre con la desarrollada en este proyecto. Ésta aplicación es muy interesante debido al elevado uso de Internet en dispositivos móviles.

Además, hay partes de la aplicación que, aunque son funcionales, necesitan ciertas mejoras. Estas mejoras están relacionadas la experiencia de usuario y la forma en la que el usuario interactúa con la aplicación, siempre respetando la sencillez y facilidad de uso.

Ha quedado también por desarrollar una aplicación que permita el alquiler de espacios por horas. Esta aplicación es menos urgente, y habría que hacer un estudio de viabilidad y coste para desarrollarla.

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] 1and1. Hosting. `http://www.1and1.es/hosting-linux?linkId=hd.mainnav.hosting`, jul 2013. 5.3.2

[5] Symfony 2.3. The book - forms. `http://symfony.com/doc/current/book/forms.html`, jun 2013. 5.2.1

[6] Symfony 2.3. The book - routing. `http://symfony.com/doc/current/book/routing.html`, jun 2013. 5.2.1

[7] Symfony 2.3. The book - security. `http://symfony.com/doc/master/book/security.html`, jun 2013. 5.2.1

[8] Symfony 2.3. The book - service container. `http://symfony.com/doc/current/book/service_container.html`, jun 2013. 5.2.1, 5.4.2

[9] Symfony 2.3. The book - translations. `http://symfony.com/doc/master/book/translation.html`, jun 2013. 5.2.1

[10] Symfony 2.3. The book - validation. `http://symfony.com/doc/current/book/validation.html`, jun 2013. 5.2.1

[11] Symfony 2.3. The event dispatcher component. `http://symfony.com/doc/current/components/event_dispatcher/introduction.html`, jun 2013. 5.2.1

[12] 3Scale. What is an API? Your guide to the Internet Business (R)evolution, 2011. 2.1.3, 5.2.12

[13] Jeremy Ashkenas. Backbone.js. http://backbonejs.org/, oct 2010. 5.2.10

[14] Jeff Attwood. Stack overflow. http://stackoverflow.com/, jul 2013. 7.2

[15] Sebastian Bergmann. Phpunit 3.7. http://phpunit.de/manual/current/en/index.html, jul 2013. 5.2.1, 6.2.1

[16] 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.

[17] James Burke. Requirejs 2.1.8. http://requirejs.org/, jul 2013. 5.2.9

[18] Chris Corbyn. Swiftmailer 5.0.1. http://swiftmailer.org/, jun 2013. 5.2.1

[19] Oracle Corporation. Mysql versión 5.5. http://www.mysql.com, apr 2010. 4.3, 5.2.5

[20] Ryan Dahl. Node.js 0.10.13. http://nodejs.org/, jul 2013. 5.2.7

[21] Clark Evans. Yaml 1.2. http://www.yaml.org/, oct 2009. 5.2.1

[22] Facebook. Graph api. https://developers.facebook.com/docs/reference/api/, jul 2013. 5.2.12, 5.10

[23] Roy Fielding. Architectural styles and the design of network-based software architectures - chapter 5 representational state transfer (rest). http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, 2000. 5.2.11

[24] Martin Fowler. Inversion of control containers and the dependency injection pattern. http://martinfowler.com/articles/injection.html, Ene 2004. 5.2.1

[25] Jose Manuel Lopez Franco. Memoria del proyecto fin de carrera: Integración de tecnologías a través de servidores web - peticiones (request) http, métodos de petición. http://trevinca.ei.uvigo.es/~txapi/espanol/proyecto/superior/memoria/node46.html, oct 2001. 5.2.11

[26] Apache Friends. Xampp. http://www.apachefriends.org/es/xampp.html, jul 2013. 5.3.2

[27] FWikipedia. Spam. http://es.wikipedia.org/wiki/Spam, jun 2013. 2.1.3

[28] Jesse James Garrett. Ajax, asynchronous javascript and xml. http://www.asp.net/ajax, 2005. 5.2.8

[29] Inc GitHub. Github. https://github.com, jun 2013. 5.3.3

[30] Google. Google analytics. `http://www.google.com/analytics/`, jul 2013. 6.3

[31] Ecma International. *IEEE Recommended Practice for Software Requirements Specifications.IEEE Std. 830-1998*. 1998. 2.1.4

[32] Ecma International. *ECMAScript Language Specification. Standard ECMA-262.* 2011. 2.1.3, 5.2.7

[33] Knp Labs. Knp bundles. `http://knpbundles.com/`, jul 2013. 5.4.1

[34] Rasmus Lerdorf. Php 5.5.1. `http://php.net/`, jul 2013. 5.2.1, 5.2.2

[35] Netbeans. Netbeans ide 7.3. `https://netbeans.org`, feb 2013. 5.3.1

[36] Mozilla Foundation Netscape Communications Corp. Javascript. `http://en.wikipedia.org/wiki/JavaScript`, jul 2013. 5.2.7

[37] OMG. Unified Modeling Language. *UML 2.4.1*, 2011. 3.1

[38] Oracle. Mysql enterprise edition. `http://www.mysql.com/products/enterprise/`, abr 2013. 5.2.5

[39] Oracle. Mysql workbench 5.2.47. `http://www.mysql.com/products/workbench/`, jul 2013. 5.3.4

[40] The phpMyAdmin Project. phpmyadmin 4.0.4.1. `http://www.phpmyadmin.net/home_page/index.php`, jun 2013. 5.3.5

[41] Fabien Potencier. Symfony 2.3.0. `http://symfony.com`, may 2013. 4.2, 5.2.1, 7.2

[42] Fabien Potencier. Twig. `http://twig.sensiolabs.org/`, jun 2013. 4.4, 5.2.3

[43] Linus Torvalds. Git 1.8.3.3. `http://git-scm.com/`, jul 2013. 5.3.3

[44] Tesla Ventures. Json, javascript object notation. `http://www.json.org`, apr 2001. 4.5, 5.2.11

[45] Konsta Vesterinen. Doctrine. `http://www.doctrine-project.org/`, ene 2013. 4.3, 5.2.6

[46] W3C. Html, hypertext markup language. `http://www.w3.org/html`, dec 1999. 5.2.4

[47] W3C. Hypertext transfer protocol http/1.1 - caching in http. `http://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html`, sep 2004. 5.2.1

[48] W3C. Word wide web consortium. `http://www.w3.org`, jun 2013. 2.3.4

[49] W3C. Xml extensible markup language. `http://www.xml.org`, 2013. 4.5, 5.2.1

[50] Wikipedia. Base de datos relacional. http://es.wikipedia.org/wiki/Base_de_datos_relacional, may 2013. 1.4

[51] Wikipedia. Caché web. http://es.wikipedia.org/wiki/Caché_web, abr 2013. 5.2.1

[52] Wikipedia. Casos de uso. http://es.wikipedia.org/wiki/Caso_de_uso, jun 2013. 3.2

[53] Wikipedia. Cross-site request forgery. http://en.wikipedia.org/wiki/Cross-site_request_forgery, jul 2013. 5.2.1

[54] Wikipedia. Framework. http://es.wikipedia.org/wiki/Framework, jul 2013. 4.2, 5.2

[55] Wikipedia. Functional testing. http://en.wikipedia.org/wiki/Functional_testing, jul 2013. 5.2.1, 6.2.2

[56] Wikipedia. Integrated development environment. https://en.wikipedia.org/wiki/Integrated_development_environment, jun 2013. 5.3.1

[57] Wikipedia. Línea de comandos. http://es.wikipedia.org/wiki/Línea_de_comandos, may 2013. 5.2.1

[58] Wikipedia. Mapeo objeto-relacional. https://es.wikipedia.org/wiki/Mapeo_objeto-relacional, mar 2013. 4.3, 5.2, 5.2.6

[59] Wikipedia. Modelo vista controlador. http://es.wikipedia.org/wiki/Modelo_Vista_Controlador, jun 2013. 4.1, 5.2.1, 5.2.6, 5.2.10

[60] Wikipedia. Sistema de gestión de bases de datos. https://es.wikipedia.org/wiki/Sistema_de_gestión_de_bases_de_datos, jul 2013. 4.3, 5.2, 5.2.5

[61] Wikipedia. Template engine (web). http://en.wikipedia.org/wiki/Template_engine_(web), jul 2013. 4.4

[62] Wikipedia. Unit testing. http://en.wikipedia.org/wiki/Unit_testing, jun 2013. 5.2.1, 6.2.1