



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

## Desarrollo de una red social para perros

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Jorge Marchán Gutiérrez

**Tutor:** Sergio Saez Barona

2014/2015



# Resumen

---

Se ha desarrollado una aplicación web la cual es una red social para perros, esta red social ha sido bautizada con el nombre de *Companimal*, en este sitio, los usuarios son los propios animales y los dueños de los mismos realizan una función de gestión de perfiles. Es decir, el dueño inicia sesión en la aplicación y desde ahí toma la decisión de acceder al perfil de la mascota que desee, a partir de ese momento, se toma la identidad de la mascota seleccionada.

Además, en esta red social, los animales tendrán la posibilidad de establecer relaciones de amistad con otros animales, subir sus propias fotografías a su galería personal, enviarse mensajes con otros perros que sean sus amigos o incluso crear en el sistema aquellos lugares que hayan visitado y dar su opinión sobre el mismo con la finalidad de que otros usuarios puedan ir o conocer su opinión. Toda esta funcionalidad será posible para los usuarios con el único requisito de tener conexión a internet y un navegador web.

**Palabras clave:** php, red social, yii, framework, web

# Abstract

---

We have developed a web application, which is a social network for dogs. This network has been named *Companimal*. On this site, users are the animals themselves and their owners manage their profiles. The owner logs in to the application and from there makes the decision to access the profile of the pet that he/she wants, from that moment the identity of the selected pet is taken.

Furthermore, in this network, the animals will be able to make friends with other animals, upload their photos to their personal galleries, send messages to other dogs that are friends or even create in the system those places that they have visited and give their opinion in order to allow other users to know what they think about these venues. All this functionality is possible for users with the only requirement of having Internet access and a web browser.

**Keywords :** php, social network, yii, framework, web



# Tabla de contenidos

---

1.	Introducción.....	8
1.1.	Motivación y justificación.....	8
1.2.	Objetivos.....	8
1.3.	Estado del arte.....	8
1.4.	Límites del proyecto.....	9
1.5.	Estructura de la memoria.....	9
2.	Requisitos software.....	11
2.1.	Propósito.....	11
2.1.1.	Alcance.....	11
2.1.2.	Personal involucrado.....	11
2.1.3.	Referencias.....	12
2.2.	Descripción general.....	12
2.2.1.	Perspectiva del producto.....	12
2.2.2.	Funcionalidad del producto.....	12
2.2.3.	Características de los usuarios.....	12
2.2.4.	Restricciones.....	13
2.2.5.	Suposiciones y dependencias.....	13
2.2.6.	Evolución previsible del sistema.....	13
2.3.	Requisitos funcionales.....	14
2.3.1.	Requisitos comunes de los interfaces.....	18
2.4.	Requisitos no funcionales.....	18
2.4.1.	Requisitos de rendimiento.....	18
2.4.2.	Requisitos de seguridad.....	18
2.4.3.	Requisitos de fiabilidad.....	18
2.4.4.	Requisitos de disponibilidad.....	18
2.4.5.	Requisitos de mantenibilidad.....	18
2.4.6.	Requisitos de portabilidad.....	18
3.	Fase de análisis.....	19
3.1.	Diagrama entidad-relación.....	19
3.2.	Casos de uso.....	20
3.2.1.	Usuario anónimo.....	20
3.2.2.	Usuario autenticado.....	22

3.2.3.	Mascota.....	25
4.	Fase de diseño .....	32
4.1.	Vistas.....	32
4.2.	Capa de persistencia .....	47
4.3.	Lógica de la aplicación.....	49
5.	Detalles de implementación.....	51
5.1.	Tecnologías empleadas .....	51
5.1.1.	Máquina servidor .....	51
5.1.2.	Sistema operativo de la máquina servidor .....	51
5.1.3.	Servidor Web .....	51
5.1.4.	Persistencia.....	51
5.1.5.	Lógica.....	52
5.1.6.	Framework utilizado .....	52
5.2.	Estructura de ficheros y dependencias .....	52
5.2.1.	Modelo-Vista-Controlador (MVC) .....	53
5.2.2.	Extensiones.....	53
5.3.	Software empleado en el desarrollo.....	54
6.	Pruebas.....	58
6.1.	Conformidad con el estándar w3c .....	58
6.2.	Visualización en navegadores .....	58
6.2.1.	Visualización en Chrome.....	59
6.2.2.	Visualización en Internet Explorer.....	60
6.2.3.	Visualización en Firefox .....	61
6.3.	Visualización a distintas resoluciones .....	62
6.4.	Requisitos funcionales .....	63
7.	Conclusiones .....	73
7.1.	Técnicas .....	73
7.2.	Personales .....	73
7.3.	Posibles ampliaciones del proyecto .....	74
8.	Bibliografía.....	75
8.1.	Libros .....	75
8.2.	Documentación online.....	75

# Índice de figuras

Figura 1: Diagrama entidad-relacion .....	19
Figura 2: Diagrama de casos de uso usuario anónimo.....	20
Figura 3: Diagrama de casos de uso usuario autenticado .....	22
Figura 4: Diagrama de casos de uso de mascota .....	25
Figura 5: Wireframe portada.....	32
Figura 6: Wireframe formulario de login .....	33
Figura 7: Wireframe formulario recordar contraseña.....	34
Figura 8: Wireframe confirmación de cambio de contraseña.....	34
Figura 9: Wireframe formulario de registro.....	35
Figura 10: Wireframe gestor de perfiles .....	36
Figura 11: Wireframe popup de confirmación.....	37
Figura 12: Wireframe formulario edición de cuenta .....	37
Figura 13: Wireframe formulario registro de nueva mascota .....	38
Figura 14: Wireframe popup recortar imagen .....	39
Figura 15: Wireframe perfil de mascota .....	39
Figura 16: Wireframe pestaña de amigos.....	40
Figura 17: Wireframe pestaña de lugares .....	40
Figura 18: Wireframe pestaña de mensajes .....	41
Figura 19: Wireframe popup envío de mensajes .....	41
Figura 20: Wireframe popup notificaciones de amistad.....	42
Figura 21: Wireframe popup añadir imagen .....	42
Figura 22: Wireframe búsqueda de animales .....	43
Figura 23: Wireframe administración de imagenes.....	43
Figura 24: Wireframe edición de imagen.....	44
Figura 25: Wireframe creación de un lugar .....	45
Figura 26: Wireframe vista detallada de un lugar.....	46
Figura 27: Esquema de base de datos relacional.....	47
Figura 28: Diagrama de flujo de la web .....	50
Figura 29: Estructura de ficheros y directorios Yii Framework.....	52
Figura 30: Captura de pantalla IDE NetBeans.....	54
Figura 31: Captura de pantalla Balsamiq Mockups.....	55
Figura 32: Captura de pantalla StarUML.....	55
Figura 33: Captura de pantalla Gliffy .....	56
Figura 34: Captura de pantalla phpmyadmin .....	56
Figura 35: Captura de pantalla cliente FTP Filezilla .....	57
Figura 36: Captura de pantalla MobaXTerm .....	57
Figura 37: Errores detectados en primer lugar por w3c markup validator .....	58
Figura 38: Diagrama navegadores más empleados .....	59
Figura 39: Vista portada en Google Chrome .....	59
Figura 40: Vista perfil en Google Chrome.....	60
Figura 41: Vista portada en Internet Explorer .....	60
Figura 42: Vista perfil en Internet Explorer.....	61
Figura 43: Vista portada en Firefox.....	61
Figura 44: Vista perfil en Firefox .....	62
Figura 45: Vista perfil en Firefox.....	62

Figura 46: Vista perfil en Google Chrome (1024x768) .....	63
Figura 47: Vista caso de uso login (I) .....	64
Figura 48: Vista caso de uso login (II) .....	64
Figura 49: Vista caso de uso registro cuenta .....	65
Figura 50: Vista caso de uso registro mascota (I) .....	65
Figura 51: Vista caso de uso registro mascota (II) .....	66
Figura 52: Vista caso de uso registro mascota (III).....	66
Figura 53: Vista caso de uso registro mascota (IV) .....	67
Figura 54: Vista caso de uso eliminar mascota .....	67
Figura 55: Vista caso de uso añadir imagen (I) .....	68
Figura 56: Vista caso de uso añadir imagen (II) .....	68
Figura 57: Vista caso de uso buscar animales (I) .....	69
Figura 58: Vista caso de uso buscar animales (II) .....	69
Figura 59: Vista caso de uso añadir amistad .....	70
Figura 60: Vista caso de uso enviar mensaje (I) .....	70
Figura 61: Vista caso de uso enviar mensaje (II).....	71
Figura 62: Vista caso de uso enviar mensaje (III).....	71
Figura 63: Vista caso de uso crear lugar (I).....	72
Figura 64: Vista caso de uso ver lugar (I).....	72

# 1. Introducción

## 1.1. Motivación y justificación

Tras finalizar los estudios de Grado en Ingeniería Informática me di cuenta de que el desarrollo web no se toca con demasiada profundidad y es una parte que personalmente me interesa mucho.

Por esto, he considerado el trabajo final de grado una oportunidad única para ser autodidacta en un campo que considero imprescindible hoy en día para cualquier experto en TI y de esta forma ampliar mis conocimientos.

Debido a los requisitos planteados por el proyecto he considerado necesario la utilización del *framework Yii* de PHP el cual se encuentra actualmente en una situación de alta demanda en el mundo laboral, debido a que se trata de un *framework* relativamente nuevo y con mucho potencial.

## 1.2. Objetivos

El objetivo principal del proyecto es desarrollar una aplicación que permita a los usuarios tomar la identidad de sus mascotas y actuar como si fuesen ellas dentro de un entorno social en el cual podrán interactuar con otros usuarios.

La aplicación se encargará de gestionar los usuarios, así como todo lo que incluyan: amistades, galerías, conversaciones, perfiles, etc. También entra dentro de los objetivos de este proyecto implementar un motor de búsqueda, tanto básica como avanzada, de otros usuarios del sistema basándose en diversas características de los perfiles (nombre, raza, sexo, color, etc.).

En cuanto a objetivos a nivel personal, al terminar el proyecto se espera dominar el *framework* utilizado, ya que hoy en día es indispensable saber desarrollar en algún sistema de características similares, dominar los distintos lenguajes de programación que voy a utilizar y por último ser capaz de enfrentarme a otros proyectos web similares en el futuro, más complejos o de mayor envergadura.

## 1.3. Estado del arte

Una aplicación web tradicional consiste en ficheros HTML, CSS y en algunas ocasiones un poco de Javascript. El HTML le dice al navegador el contenido que tiene que mostrar, el CSS le dice al navegador como debe mostrar ese contenido y por último Javascript se encarga de hacer pequeñas cosas, como mostrar un *popup* o algún mensaje. Este tipo de sitios web son difíciles de mantener, ya que cambiar el contenido conlleva cambiar el código fuente y volver a subirlo al entorno de producción, mediante un cliente FTP, por suerte, existen lenguajes como PHP que permiten generar páginas “*al vuelo*” esto quiere decir, que si se conectan con una base de datos, pueden generar contenido dinámico basándose en esta.

Usar PHP sin ningún *framework* en el lado del servidor es una tarea difícil si el proyecto tiene cierta envergadura y queremos generar código escalable y fácil de mantener.



Actualmente, con la incursión de las aplicaciones móviles, el modelo cliente-servidor está quedándose atrás, debido a que cada vez más, existen entornos en los que se requiere una conexión abierta continua entre el cliente y el servidor, sin tener que ser necesaria una petición para generar una respuesta, es decir, cada vez se requiere más que la comunicación sea bidireccional, lo cual es interesante para aplicaciones que requieren conexiones en tiempo real, este nuevo modelo se basa en el lenguaje NodeJS el cual es Javascript en la parte servidor.

Sin embargo, no he considerado oportuno emplearlo en este proyecto debido principalmente a que la estabilidad es bastante menor que un servidor Apache y la robustez del código Javascript es bastante menor en comparación con PHP, el cual ha venido demostrando desde años atrás ser un lenguaje sólido y seguro. Por otra parte he considerado más adecuado un modelo relacional de base de datos en lugar de un modelo no relacional, por la naturaleza del proyecto, estos motivos han sido los que han hecho que termine decantándome por esta arquitectura.

## 1.4. Límites del proyecto

Este proyecto abarca el diseño y la implementación de la parte básica de la red social y sus diferentes fases, análisis, diseño e implementación.

## 1.5. Estructura de la memoria

En esta memoria empezaré enseñando los requisitos software de la aplicación según la normativa del [IEEE 830-1998](#), a continuación expondré la fase de análisis donde se verá el estudio según los casos de uso de todos los actores que intervienen en la aplicación de igual forma que se hará con el diagrama de clases.

Seguido a este apartado, encontraremos la fase de diseño, la cual se divide en tres fases: diseño de la presentación, diseño del modelo de persistencia y diseño de la lógica de la aplicación.

- **Diseño de la presentación:** Esta sección de la fase de diseño será expuesta a partir de *wireframes* o *mockups*.
- **Modelado de datos:** En esta sección expondré el esquema de la base de datos y las relaciones entre las tablas que forman parte de la misma.
- **Diseño de la lógica de aplicación:** En este apartado veremos un mapa web del sitio, para hacernos una idea de las posibilidades que tendrán nuestros usuarios para moverse dentro del sistema.

Tras finalizar la explicación de las distintas fases de diseño de la aplicación pasaremos a los detalles de implementación, en esta sección veremos y analizaremos la estructura de ficheros que nos genera el *framework* que estamos empleando, seguidamente pasaremos a hablar de las tecnologías empleadas durante el desarrollo del proyecto y el software que se ha utilizado.

En el siguiente punto comprobaremos cómo reacciona nuestro sistema al someterlo a diversas pruebas, estas pruebas incluyen algunas de cumplimiento de estándares, como someter nuestra web al *w3c Markup Validation Service*, el cual es un servicio web que únicamente analiza el código HTML buscando faltas de conformidad con el estándar *w3c*. También ofreceremos pruebas para observar cómo se comporta nuestra aplicación ante diversas resoluciones de pantalla.

Seguidamente nos encontraremos con la fase de pruebas, en esta fase haremos pruebas de uso mediante las cuales analizaremos y pondremos a prueba toda la funcionalidad que ofrece la aplicación explicándola mediante la ayuda de capturas de pantalla de la aplicación.

Para finalizar, llegamos a las conclusiones, divididas en dos tipos, conclusiones a nivel personal y a nivel técnico, después de las conclusiones dejaremos paso a la bibliografía utilizada y al glosario de términos empleados, ordenado de forma alfabética.

## 2. Requisitos software

### 2.1. Propósito

La presente sección tiene como propósito definir las especificaciones funcionales y no funcionales para el desarrollo de un sistema web que permitirá a distintos usuarios interactuar entre sí y con el sistema, una red social. Este será utilizado por personas de diferente perfil y situación social con un único punto en común, todos tendrán perros como mascotas.

A partir de esta especificación, se podrá establecer un diseño que se ajuste a los requisitos aquí expuestos.

#### 2.1.1. Alcance

Esta especificación de requisitos está dirigida al usuario del sistema *Companimal*. El cual tiene como objetivo ser una red social para animales de compañía (perros). El objetivo es lograr que el sistema sea un punto de encuentro para todas aquellas personas que tengan como mascota un perro.

#### 2.1.2. Personal involucrado

A continuación, las personas que estarán involucradas en el desarrollo de este proyecto, en primer lugar, el alumno y en segundo lugar el profesor que ha tutelado y supervisado el desarrollo del mismo.

Nombre	Jorge Marchán Gutiérrez
Rol	Desarrollador software
Categoría profesional	Estudiante de Grado en Ingeniería Informática
Responsabilidades	Análisis de información, diseño y programación del sistema
Información de contacto	jorma16@inf.upv.es

Nombre	Sergio Sáez Barona
Rol	Director del proyecto
Categoría profesional	Doctor en Informática y profesor de la Universitat Politècnica de Valencia
Responsabilidades	Supervisar y asesorar en el desarrollo del proyecto al desarrollador
Información de contacto	ssaez@disca.upv.es

### 2.1.3. Referencias

Referencia	Título	Ruta	Fecha	Autor
1	Desarrollo de una red social para perros	Este documento	25/08/2015	Jorge Marchán Gutiérrez
2	Estandar IEEE 830/1998	<a href="#">IEEE 830-1998</a>	22/10/2015	Universidad Complutense Informática

## 2.2. Descripción general

### 2.2.1. Perspectiva del producto

Este proyecto trata sobre el desarrollo de una red social para animales domésticos (perros). Se trata de un producto que estará orientado, diseñado y desarrollado para trabajar en entornos web y de escritorio, lo cual permitirá su utilización de forma rápida y eficaz, sin embargo su utilización en entornos móviles estará fuera del alcance del proyecto, quedando como líneas futuras de desarrollo.

El proyecto no tendrá dependencias externas de ningún tipo.

### 2.2.2. Funcionalidad del producto

Este sistema pretende ser un punto en común para todas aquellas personas que posean un perro como animal de compañía, permitiéndoles interactuar en ella a través de sus mascotas, subir fotos y comentar las de sus conocidos, así como establecer relaciones con los mismos y comunicarse los unos con los otros. Todo, gestionando un perfil por mascota, es decir, realmente son los animales los que interactúan entre sí (virtualmente hablando).

La aplicación gestionará los usuarios, su registro y su identificación entre otras cosas, así como toda la funcionalidad asociada a establecer relaciones entre los distintos usuarios, gestión de la cuenta, edición de los distintos perfiles. También se desarrollará un apartado de búsqueda basada en diversas características de las mascotas.

### 2.2.3. Características de los usuarios

Tipo de usuario	Usuario no autenticado
Formación	Diversos perfiles contemplados
Actividades	Ver la portada, ver los términos y condiciones, contactar a través del formulario, registrarse

Tipo de usuario	Usuario autenticado
Formación	Diversos perfiles contemplados
Actividades	Crear mascotas, editar su perfil, editar el perfil de la mascota, eliminar la mascota, eliminar la cuenta

Tipo de usuario	Usuario Mascota
Formación	Diversos perfiles contemplados
Actividades	Subir imágenes, buscar otras mascotas, editar imágenes, crear lugares, añadir lugares, opinar sobre lugares, establecer relaciones con otras mascotas, editar mascota

#### 2.2.4. Restricciones

- Interfaz para ser usada con internet.
- Lenguajes y tecnologías que se usarán: HTML5, Javascript, PHP, MySQL, Ajax, CSS3.
- El servidor deberá ser capaz de atender peticiones de forma concurrente.
- El sistema se diseñará siguiendo la arquitectura cliente-servidor clásica, es decir, petición-respuesta.

#### 2.2.5. Suposiciones y dependencias

Se asume que los requisitos aquí descritos son estables y que los equipos en los que se vaya a ejecutar el sistema deberán tener una conexión a internet de banda ancha y cualquier navegador web del mercado instalado en el equipo, además de la potencia suficiente para obtener los resultados en un tiempo razonable.

La web estará implementada sobre un servidor Apache con una base de datos montada en MySQL.

La aplicación no está pensada para funcionar en dispositivos móviles, sin embargo, se podrá acceder desde los mismos, pero la funcionalidad no estará asegurada.

#### 2.2.6. Evolución previsible del sistema

Se trata de un proyecto el cual se va a desarrollar en principio como un trabajo final de grado, por lo que el desarrollo finalizará en el momento de su defensa. Sin embargo se podría continuar con el desarrollo si se optase por publicarlo en la web para que pasase a una fase de producción.



## 2.3. Requisitos funcionales

En este apartado vamos a hablar sobre las distintas funciones del sistema software y de sus componentes. Estos requisitos funcionales establecen el comportamiento del sistema para cada uno de ellos.

Identificación del requisito	RF01
Nombre del requisito	Autenticación de Usuario
Características	Los usuarios deberán identificarse para acceder a cualquier parte del sistema
Descripción del requisito	El sistema podrá ser consultado por cualquier usuario del mismo, pudiendo acceder a unos módulos u otros en función de su nivel de accesibilidad
Prioridad del requisito	Alta/Esencial

Identificación del requisito	RF02
Nombre del requisito	Registrar usuarios
Características	Los usuarios deberán registrarse en el sistema para poder autenticarse y acceder a cualquier parte del mismo
Descripción del requisito	El sistema permitirá al usuario registrarse. El usuario debe suministrar datos como: email y contraseña
Prioridad del requisito	Alta/Esencial

Identificación del requisito	RF03
Nombre del requisito	Visitar perfiles
Características	El sistema ofrecerá a los distintos usuarios información general acerca de los demás perfiles accesibles en el sistema y de su contenido.
Descripción del requisito	Visitar perfiles: Muestra el perfil de otro usuario así como su galería de foto, datos del usuario y últimos comentarios
Prioridad del requisito	Alta

Identificación del requisito	RF04
Nombre del requisito	Búsqueda de usuarios
Características	Los usuarios podrán buscar otros usuarios del sistema.
Descripción del requisito	Se facilitará a los usuarios un sistema de búsqueda para promover la interacción entre ellos, de esta forma, los usuarios podrán conocer de un vistazo todos los perfiles que son de su interés.
Prioridad del requisito	Alta

Identificación del requisito	RF05
Nombre del requisito	Añadir amigos
Características	Los usuarios podrán crear relaciones de amistad entre sí
Descripción del requisito	En la vista de cada perfil, se mostrará información acerca de la relación que se tiene con ese usuario, se podrá agregar al usuario como amigo en caso de no tenerlo ya agregado
Prioridad del requisito	Alta

Identificación del requisito	RF06
Nombre del requisito	Edición del perfil
Características	Los usuarios podrán editar los perfiles de sus mascotas
Descripción del requisito	Los usuarios podrán, una vez creados los perfiles de sus mascotas, modificarlos mediante una interfaz de usuario pensada para tal fin.
Prioridad del requisito	Alta

Identificación del requisito	RF07
Nombre del requisito	Eliminar perfil
Características	Los usuarios podrán borrar los perfiles de sus mascotas
Descripción del requisito	Como administradores de todas sus mascotas, los usuarios podrán eliminar aquellos perfiles que ya no utilicen o no quieran tener en el sistema.
Prioridad del requisito	Alta

Identificación del requisito	RF08
Nombre del requisito	Crear perfil
Características	Los usuarios podrán crear perfiles de sus mascotas
Descripción del requisito	Desde la interfaz de gestión de perfiles los usuarios tendrán acceso a un formulario de creación de mascotas, en este formulario rellenarán una serie de campos describiendo a su mascota y posteriormente tendrán un nuevo perfil en la red social.
Prioridad del requisito	Alta

Identificación del requisito	RF09
Nombre del requisito	Eliminar cuenta
Características	Los usuarios podrán darse de baja del sistema
Descripción del requisito	Desde la sección de editar la cuenta, los usuarios tendrán la posibilidad de borrar de forma definitiva la cuenta, con todos los perfiles de mascotas que ésta posea.
Prioridad del requisito	Alta

Identificación del requisito	RF10
Nombre del requisito	Recuperar contraseña
Características	Los usuarios tendrán la posibilidad de recuperar su contraseña
Descripción del requisito	Indicando la dirección de correo electrónico los usuarios podrán generar una contraseña nueva y será enviada a su correo electrónico, a partir de ese momento deberán acceder a la cuenta con esa contraseña.
Prioridad del requisito	Alta

Identificación del requisito	RF11
Nombre del requisito	Subir imagen
Características	Los usuarios podrán subir fotos nuevas a las galerías de sus diferentes perfiles
Descripción del requisito	Mediante un formulario los usuarios podrán subir fotos a sus diferentes perfiles, así como poner un título y una pequeña descripción a cada una de sus fotografías
Prioridad del requisito	Alta

Identificación del requisito	RF12
Nombre del requisito	Edición de imagen
Características	Los usuarios podrán editar cualquier imagen que tengan en las galerías de sus mascotas
Descripción del requisito	Podrán editar el título o la descripción de todas las imágenes que estén disponibles en las galerías de sus mascotas
Prioridad del requisito	Alta

Identificación del requisito	RF13
Nombre del requisito	Eliminar imagen
Características	Eliminar del sistema una imagen de un perfil concreto
Descripción del requisito	Los usuarios podrán borrar cualquier imagen de un perfil del cual sean propietarios
Prioridad del requisito	Alta

Identificación del requisito	RF14
Nombre del requisito	Búsqueda de imágenes
Características	Buscar una imagen concreta
Descripción del requisito	Los usuarios podrán, mediante la fecha de subida o alguna palabra clave, buscar imágenes concretas de las cuales ellos sean propietarios
Prioridad del requisito	Alta



Identificación del requisito	RF15
Nombre del requisito	Ver lugares
Características	Ver donde está un lugar y toda la información relacionada
Descripción del requisito	Los usuarios podrán ver el mapa con la localización del lugar señalada, los comentarios de los demás usuarios y una breve descripción del lugar.
Prioridad del requisito	Alta

Identificación del requisito	RF16
Nombre del requisito	Crear lugares
Características	Crear lugares que has visitado con tu mascota
Descripción del requisito	Los usuarios podrán señalar y buscar en un mapa los lugares donde han estado con sus mascotas, para crearlos e incluirlos en el sistema, de forma que otros usuarios puedan añadirlos en un futuro.
Prioridad del requisito	Alta

Identificación del requisito	RF17
Nombre del requisito	Opinar sobre lugares
Características	Escribir una opinión acerca de un lugar
Descripción del requisito	Los usuarios podrán dar su opinión sobre aquellos lugares que han visitado, quedando a la vista de los demás usuarios en la vista detallada del lugar
Prioridad del requisito	Alta

Identificación del requisito	RF18
Nombre del requisito	Añadir lugar
Características	Añadir lugar a tu lista de visitados
Descripción del requisito	Los usuarios podrán añadir lugares ya existentes a su lista de lugares visitados.
Prioridad del requisito	Alta

Identificación del requisito	RF19
Nombre del requisito	Borrar lugar
Características	Borrar lugar de lista de visitados
Descripción del requisito	Los usuarios podrán borrar lugares de su lista de visitados, sin embargo, no podrán eliminar lugares del sistema
Prioridad del requisito	Alta

Identificación del requisito	RF20
Nombre del requisito	Buscar lugar
Características	Buscar lugar por nombre
Descripción del requisito	Los usuarios podrán buscar lugares dados de alta en el sistema basándose en el nombre del mismo o en palabras existentes en su descripción
Prioridad del requisito	Alta

### **2.3.1. Requisitos comunes de los interfaces**

- Las interfaces de usuario (GUI) deberán ser sencillas para que los usuarios naveguen por ellas de forma cómoda e intuitiva, no resultar engorrosas para el usuario ni agotadoras visualmente.
- Las interfaces hardware están constituidas por una máquina servidora y las máquinas cliente, las cuales acceden mediante el protocolo HTTP a la aplicación.
- Las interfaces de comunicación que se utilizan en este proyecto son principalmente el protocolo HTTP para hacer llegar las distintas vistas de la aplicación al cliente y que este pueda visualizarlo, por debajo del protocolo HTTP encontramos el protocolo TCP/IP, el cual es el encargado de que los datos lleguen sin pérdidas y de una forma rápida y ordenada.

## **2.4. Requisitos no funcionales**

A continuación se hablará de los requisitos no funcionales del sistema, estos son aquellos que no describen información a guardar o formas de interactuar con el sistema.

### **2.4.1. Requisitos de rendimiento**

El sistema tiene que ser capaz de reaccionar con un tiempo de respuesta inferior a 3 segundos.

### **2.4.2. Requisitos de seguridad**

La seguridad de las cuentas de usuario no debe estar comprometida en ningún momento.

### **2.4.3. Requisitos de fiabilidad**

Las operaciones permitidas deben desarrollarse sin incidentes.

### **2.4.4. Requisitos de disponibilidad**

La disponibilidad del sistema debe ser máxima y en caso de fallo, la reactivación del servicio deberá ser en el menor tiempo posible.

### **2.4.5. Requisitos de mantenibilidad**

Las tareas de administración del sistema deben ser lo más sencillas posibles, haciendo que el encargado de realizarlas tenga la menor cantidad de trabajo posible.

El sistema tiene que poder ampliarse en caso de necesitar más funcionalidad en un futuro, sin que esto afecte a lo que ya se tiene desarrollado.

### **2.4.6. Requisitos de portabilidad**

La portabilidad y migración de todo el sistema debe de poder llevarse a cabo en la menor cantidad de tiempo posible, siendo lo más transparente posible para el usuario final.

## 3. Fase de análisis

### 3.1. Diagrama entidad-relación

A continuación voy a representar las distintas partes de mi aplicación mediante un diagrama entidad-relación con el objetivo de hacernos una idea de las diferentes partes que de las que va a constar el sistema y de las relaciones entre las mismas. En la parte de diseño entraremos más en detalle en cada una de las entidades.

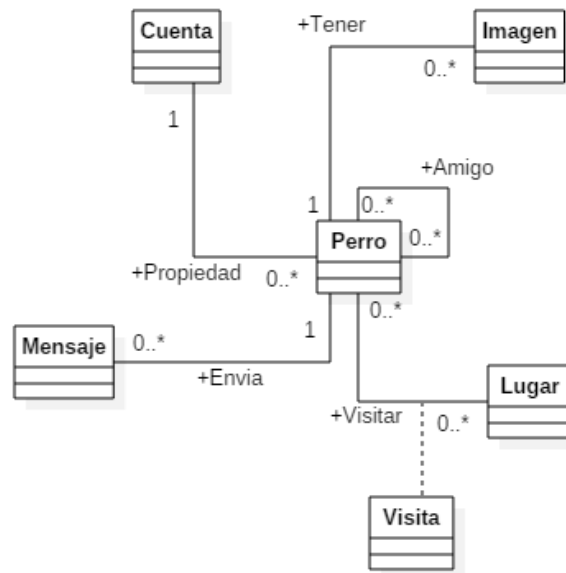


Figura 1: Diagrama entidad-relacion

En la **Figura 1** como ya se ha mencionado, se puede observar las diferentes entidades presentes en el sistema así como las relaciones y multiplicidades entre ellas.

En primer lugar se observa que van a existir **cuentas** en el sistema, cada usuario tendrá una única cuenta y esta cuenta estará vinculada a ningún **perro** o a muchos perros, se entiende por perros, todas las mascotas que posea una cuenta, es decir, los distintos perfiles a los que los usuarios de la aplicación tienen acceso.

A continuación se puede ver que un perro podrá no tener ninguna **imagen** o tener muchas, sin embargo una imagen únicamente puede ser de un perro. También se puede observar como un perro puede no tener amigos o tener muchos.

Si nos fijamos en la entidad **Lugar** se observa que tiene una relación con Perro la cual indica que un lugar puede no haber sido visitado nunca por ningún perro o haberlo sido por muchos, también observamos cómo cada una de las relaciones establecidas, tiene una entidad relación con datos específicos de la visita.

Para finalizar, la comunicación entre los distintos usuarios de la aplicación se representará mediante una última entidad llamada **mensaje**, que como podemos observar un mensaje se le envía a un solo perro, sin embargo un perro puede enviar muchos mensajes.

## 3.2. Casos de uso

A continuación el diagrama de casos de uso de la aplicación, nuestro diagrama de casos de uso consta de dos actores. Usuario autenticado y usuario sin autenticar. En primer lugar vamos a ver y comentar el diagrama del usuario sin autenticar, que es el más limitado de los dos y a continuación el usuario autenticado en la plataforma.

### 3.2.1. Usuario anónimo

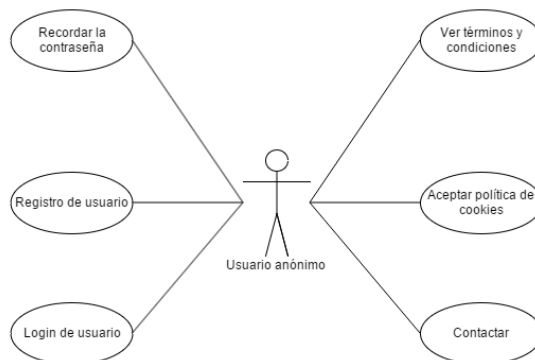


Figura 2: Diagrama de casos de uso usuario anónimo

A continuación veremos las fichas de los casos de uso para el usuario anónimo en nuestra web:

<b>Nombre</b>	Recordar la contraseña	
<b>Descripción</b>	El usuario desea recordar la contraseña con la que se registró en Companimal	
<b>Precondición</b>	Ninguna	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario accede a la web
	2	El usuario hace click en “Recordar contraseña”
	3	El usuario introduce su dirección de correo electrónico y lo envía
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	Solicitud de cambio de contraseña enviada	
<b>Notas</b>	-	

<b>Nombre</b>	Registro de usuario	
<b>Descripción</b>	El usuario toma la decisión de registrarse en la plataforma	
<b>Precondición</b>	Ninguna	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario accede a la web
	2	El usuario hace click en “Regístrate”
	3	El usuario cumplimenta el formulario de registro y lo envía
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	Usuario registrado en el sistema	
<b>Notas</b>	-	

<b>Nombre</b>	Login de usuario	
<b>Descripción</b>	El usuario inicia sesión en la plataforma	
<b>Precondición</b>	Ninguna	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario accede a la web
	2	El usuario introduce sus credenciales de acceso
	3	El usuario accede a la plataforma
<b>Errores / Alternativas</b>	<b>Error / Alternativa</b>	<b>Acción</b>
	El usuario introduce unas credenciales inválidas	Se le indica que son inválidas y se le ofrece la oportunidad de volverlas a introducir
<b>Postcondición</b>	Usuario autenticado en el sistema	
<b>Notas</b>	-	

<b>Nombre</b>	Ver términos y condiciones	
<b>Descripción</b>	El usuario quiere ver los términos y condiciones de la plataforma	
<b>Precondición</b>	Ninguna	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario accede a la web
	2	El usuario hace click en “Términos y condiciones”
	3	El usuario puede leer los términos y condiciones
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Aceptar política de cookies	
<b>Descripción</b>	El usuario acepta la política de cookies	
<b>Precondición</b>	Ninguna	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario accede a la web
	2	El usuario hace click en “Aceptar” en la barra de cookies
	3	La barra desaparece
<b>Errores / Alternativas</b>	<b>Error / Alternativa</b>	<b>Acción</b>
	El usuario hace click en “¿Qué son las cookies?”	Se le redirecciona a la página de la Wikipedia donde explica que son las cookies
<b>Postcondición</b>	El usuario ha aceptado la política de cookies	
<b>Notas</b>	-	

<b>Nombre</b>	Contactar	
<b>Descripción</b>	El usuario quiere contactar con la plataforma	
<b>Precondición</b>	Ninguna	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario accede a la web
	2	El usuario hace click en “Contacta”
	3	El usuario rellena el formulario de contacto
<b>Errores / Alternativas</b>	<b>Error / Alternativa</b>	<b>Acción</b>
	El usuario no introduce todos los campos obligatorios del formulario o los introduce de manera incorrecta.	Se le indica que campos son erróneos u obligatorios y se le da la oportunidad de cambiarlos
<b>Postcondición</b>	Email enviado el sistema	
<b>Notas</b>	-	

### 3.2.2. Usuario autenticado

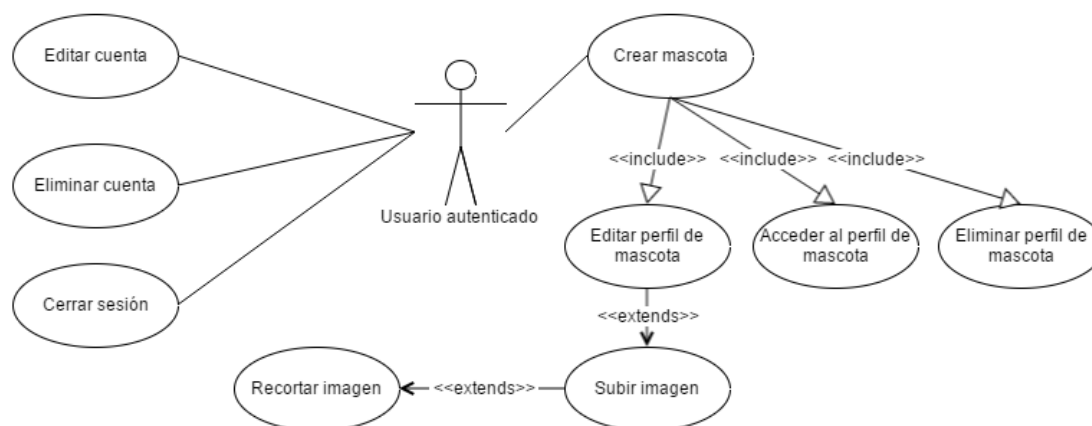


Figura 3: Diagrama de casos de uso usuario autenticado

<b>Nombre</b>	Editar cuenta	
<b>Descripción</b>	El usuario puede editar su información personal	
<b>Precondición</b>	Estar autenticado en la plataforma	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click en “Editar cuenta”
	2	El usuario edita los campos que quiere de su cuenta
	3	El usuario guarda los cambios
<b>Errores / Alternativas</b>	<b>Error / Alternativa</b>	<b>Acción</b>
	El usuario no introduce algún campo que es obligatorio o lo introduce mal	Se le indica que campos son erróneos u obligatorios y se le da la oportunidad de cambiarlos
<b>Postcondición</b>	Campos actualizados en el sistema	
<b>Notas</b>	-	

<b>Nombre</b>	Eliminar cuenta	
<b>Descripción</b>	El usuario quiere borrar su cuenta	
<b>Precondición</b>	Estar autenticado en la plataforma	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click en “Eliminar cuenta”
	2	El usuario confirma la eliminación de la cuenta
	3	La cuenta se elimina
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	Cuenta borrada	
<b>Notas</b>	Si un usuario elimina una cuenta se eliminan todos aquellos objetos que tengan relación con ella.	

<b>Nombre</b>	Cerrar sesión	
<b>Descripción</b>	El usuario quiere salir del sitio	
<b>Precondición</b>	Estar autenticado en la plataforma	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click en logout
	2	Se cierra su sesión
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	Usuario anónimo	
<b>Notas</b>	-	

<b>Nombre</b>	Crear mascota	
<b>Descripción</b>	El usuario quiere crear un perfil de mascota	
<b>Precondición</b>	Estar autenticado en la plataforma	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click en “Crear nueva mascota”
	2	El usuario rellena el formulario
	3	Se crea el perfil de la mascota
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	Mascota añadida al sistema y con funcionalidad total	
<b>Notas</b>	-	

<b>Nombre</b>	Editar perfil de mascota	
<b>Descripción</b>	El usuario quiere cambiar la información de alguna de sus mascotas	
<b>Precondición</b>	Estar autenticado en la plataforma	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click sobre el botón “Editar” de la mascota correspondiente
	2	El usuario rellena el formulario
	3	Se actualizan los datos
<b>Errores / Alternativas</b>	<b>Errores / Alternativas</b>	<b>Acción</b>
	El usuario introduce los datos de manera equivocada	Se indica que datos son erróneos y se da la posibilidad de cambiarlos
<b>Postcondición</b>	Mascota actualizada	
<b>Notas</b>	-	

<b>Nombre</b>	Acceder al perfil de mascota	
<b>Descripción</b>	El usuario desea acceder a la plataforma como una de sus mascotas	
<b>Precondición</b>	Estar autenticado en la plataforma	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click sobre el botón “Acceder” de la mascota correspondiente
	2	El usuario toma la identidad de la mascota
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	A partir de ese momento pasa de ser usuario autenticado a ser una mascota	
<b>Notas</b>	-	

<b>Nombre</b>	Eliminar perfil de mascota	
<b>Descripción</b>	El usuario quiere eliminar una de sus mascotas	
<b>Precondición</b>	Estar autenticado en la plataforma	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click sobre el botón “Eliminar” de la mascota correspondiente
	2	El usuario confirma la eliminación de la cuenta
	3	La cuenta se elimina
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	Mascota eliminada	
<b>Notas</b>	Si un usuario decide borrar una de sus mascotas se borrarán también todos los objetos relacionados con ella del sistema.	

<b>Nombre</b>	Subir imagen	
<b>Descripción</b>	El usuario quiere subir una foto para el perfil de su mascota	
<b>Precondición</b>	Estar autenticado en la plataforma y crear o editar una mascota	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click sobre el botón “Subir foto”
	2	El usuario elige la fotografía
	3	La foto de perfil de la mascota se actualiza
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	La información de la mascota se actualiza	
<b>Notas</b>	-	



<b>Nombre</b>	Recortar imagen	
<b>Descripción</b>	El usuario quiere recortar una foto que acaba de subir	
<b>Precondición</b>	Estar autenticado en la plataforma y crear o editar una mascota y subir una foto	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona arrastrando y soltando la región de la imagen en la cual está interesado
	2	El usuario hace click en “Recortar”
	3	La foto se recorta y se actualiza el perfil de la mascota
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	La información de la mascota se actualiza	
<b>Notas</b>	-	

### 3.2.3. Mascota



Figura 4: Diagrama de casos de uso de mascota

<b>Nombre</b>	Enviar mensaje	
<b>Descripción</b>	El usuario quiere enviar un mensaje interno a uno de sus amigos	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click en la pestaña “Amigos”
	2	El usuario hace click en “Enviar mensaje”
	3	El usuario escribe el mensaje y lo envía
<b>Errores / Alternativas</b>	No	
<b>Postcondición</b>	Se envía el mensaje al amigo correspondiente	
<b>Notas</b>	-	

<b>Nombre</b>	Ver amigos	
<b>Descripción</b>	El usuario puede ver una lista de aquellos perros que son amigos suyos	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click en la pestaña “Amigos”
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Ver mensajes	
<b>Descripción</b>	El usuario puede ver su buzón de correo, tanto los mensajes enviados como los recibidos	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click en la pestaña “Mensajes”
<b>Errores / Alternativas</b>	<b>Errores / Alternativa</b>	<b>Acción</b>
	Leer los recibidos	Hace click en “recibidos”
	Leer los enviados	Hace click en “enviados”
<b>Postcondición</b>	Los mensajes se marcan como leídos	
<b>Notas</b>	-	

<b>Nombre</b>	Buscar imagen	
<b>Descripción</b>	El usuario puede buscar las imágenes de su galería que más le interesen en ese momento	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas y estar administrando su galería	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Rellena el formulario de búsqueda
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Filtrar búsqueda de imágenes	
<b>Descripción</b>	El usuario puede filtrar las imágenes que busca	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas y estar administrando su galería	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Rellena el formulario de búsqueda y elegir el filtro correspondiente
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Administrar imágenes	
<b>Descripción</b>	El usuario puede acceder a una sección de la plataforma donde podrá gestionar todas las imágenes que haya subido a ese perfil.	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Hacer click en “Administrar imágenes”
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Eliminar imagen	
<b>Descripción</b>	El usuario puede eliminar cualquier imagen que haya subido el	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas y estar administrando su galería	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Darle al botón de eliminar cuando la imagen correspondiente aparezca en pantalla
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	Imagen eliminada del sistema	
<b>Notas</b>	-	

<b>Nombre</b>	Añadir imagen	
<b>Descripción</b>	El usuario puede subir cualquier imagen de su equipo a la plataforma para compartirla con los demás	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Darle al botón de añadir imagen
	2	Elegir la imagen correspondiente
	3	Ponerle un título
	4	Ponerle una descripción
<b>Errores / Alternativas</b>	<b>Errores / Alternativas</b>	<b>Acción</b>
	No se rellenan todos los campos obligatorios o se rellenan mal	Se indica al usuario aquellos campos incorrectos y se le da la oportunidad de corregirlos
<b>Postcondición</b>	Imagen insertada en el sistema	
<b>Notas</b>	-	

<b>Nombre</b>	Ver notificaciones de amistad	
<b>Descripción</b>	El usuario puede ver las notificaciones de amistad que tiene de otros usuarios	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Darle a la pestaña de notificaciones de amistad
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Aceptar notificación de amistad	
<b>Descripción</b>	De las notificaciones de amistad que le hayan enviado el usuario puede aceptar las que le convengan y rechazar las demás	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Darle a la pestaña de notificaciones de amistad
	2	Aceptar notificación
<b>Errores / Alternativas</b>	<b>Errores / Alternativas</b>	<b>Acción</b>
	Rechazar notificación	La notificación desaparece del sistema
<b>Postcondición</b>	Los usuarios son amigos	
<b>Notas</b>	-	

<b>Nombre</b>	Editar información	
<b>Descripción</b>	Actualizar el perfil de la mascota	
<b>Precondición</b>	Estar autenticado en la plataforma	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario hace click en "Editar información"
	2	El usuario rellena el formulario
	3	Se actualizan los datos
<b>Errores / Alternativas</b>	<b>Errores / Alternativas</b>	<b>Acción</b>
	El usuario introduce los datos de manera equivocada	Se indica que datos son erróneos y se da la posibilidad de cambiarlos
<b>Postcondición</b>	Mascota actualizada	
<b>Notas</b>	-	

<b>Nombre</b>	Crear lugar	
<b>Descripción</b>	El usuario puede crear cualquier lugar que haya visitado con esa mascota en la plataforma	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Hacer click en “Crear lugar”
	2	Ponerle nombre
	3	Posicionar en google maps el lugar
	4	Ponerle una descripción
<b>Errores / Alternativas</b>	<b>Errores / Alternativas</b>	<b>Acción</b>
	No se rellenan todos los campos obligatorios o se rellenan mal	Se indica al usuario aquellos campos incorrectos y se le da la oportunidad de corregirlos
<b>Postcondición</b>	Lugar creado en el sistema	
<b>Notas</b>	-	

<b>Nombre</b>	Opinar sobre lugar	
<b>Descripción</b>	El usuario puede opinar sobre cualquier lugar que haya visitado con su mascota	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Hacer click en la pestaña “Lugares”
	2	Seleccionar la fila correspondiente y hacer click en “Opinar”
	3	Escribir la opinión
	4	Enviar el formulario
<b>Errores / Alternativas</b>	<b>Errores / Alternativas</b>	<b>Acción</b>
	No se rellenan todos los campos obligatorios o se rellenan mal	Se indica al usuario aquellos campos incorrectos y se le da la oportunidad de corregirlos
<b>Postcondición</b>	Opinión registrada en el sistema	
<b>Notas</b>	-	

<b>Nombre</b>	Añadir lugar	
<b>Descripción</b>	El usuario puede añadir cualquier lugar a su lista de lugares visitados de los que ya existen en el sistema	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Hacer click en “Añadir lugar”
	2	Seleccionar el lugar
	3	Opinar sobre el
	4	Enviar el formulario
<b>Errores / Alternativas</b>	<b>Errores / Alternativas</b>	<b>Acción</b>
	No se rellenan todos los campos obligatorios o se rellenan mal	Se indica al usuario aquellos campos incorrectos y se le da la oportunidad de corregirlos
<b>Postcondición</b>	Lugar añadido a tu perfil	
<b>Notas</b>	-	

<b>Nombre</b>	Buscar perros	
<b>Descripción</b>	El usuario puede buscar los perros que le interesen	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Hacer click en “Buscar perros”
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Filtrar búsqueda de perros	
<b>Descripción</b>	El usuario puede acotar la búsqueda de animales con el fin de encontrar lo que busca	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas y en la pantalla de búsqueda	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Elegir los filtros deseados
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Ver perfil	
<b>Descripción</b>	El usuario puede ver su perfil o el de cualquier otro usuario de la plataforma	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Hacer click sobre una mascota o en “Perfil”
<b>Errores / Alternativas</b>	<b>Errores / Alternativas</b>	<b>Acción</b>
	No es tu perfil	No podrás editar información
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Ver galería	
<b>Descripción</b>	El usuario puede ver la galería de cualquier mascota desde su perfil	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas y estar en el perfil de alguna mascota	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Hacer click sobre la pestaña “Galería”
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

<b>Nombre</b>	Ver lugar	
<b>Descripción</b>	El usuario puede ver la galería de cualquier mascota desde su perfil	
<b>Precondición</b>	Estar autenticado en la plataforma e interactuando como una de las mascotas y estar en el perfil de alguna mascota	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	Hacer click sobre la pestaña “Lugares” para ver los lugares que ha visitado y la opinión
	2	Hacer click sobre el lugar deseado para ir a la vista detallada
<b>Errores / Alternativas</b>	-	
<b>Postcondición</b>	-	
<b>Notas</b>	-	

## 4. Fase de diseño

A continuación vamos a comentar la fase de diseño del proyecto, la he segmentado en tres partes. En primer lugar hablaré del diseño gráfico de la aplicación, es decir, las distintas vistas de la aplicación, para el diseño inicial y la arquitectura de la información he optado por una herramienta software que me facilitaba la tarea, *Balsamiq mockups*.

Seguidamente pasaré a hablar del diseño de la capa de persistencia de la aplicación, en esta parte mostraré el diagrama de clases de la misma, donde explicaré un diagrama similar al visto en la fase de análisis (entidad-relación) pero entrando más al detalle de la información que queremos guardar de cada entidad y la forma de representación de las diferentes relaciones presentes en nuestro esquema.

Para finalizar la fase de diseño, mostraré un diagrama de flujo donde podremos observar las distintas acciones que podrá realizar un usuario desde los distintos puntos de la aplicación.

### 4.1. Vistas

Voy a empezar mostrando los distintos *layouts* que conforman la aplicación, en ellos veremos la distribución de la información en cada una de las vistas. Al finalizar esta sección sabremos donde debe estar la información en cada una de las vistas del sistema.

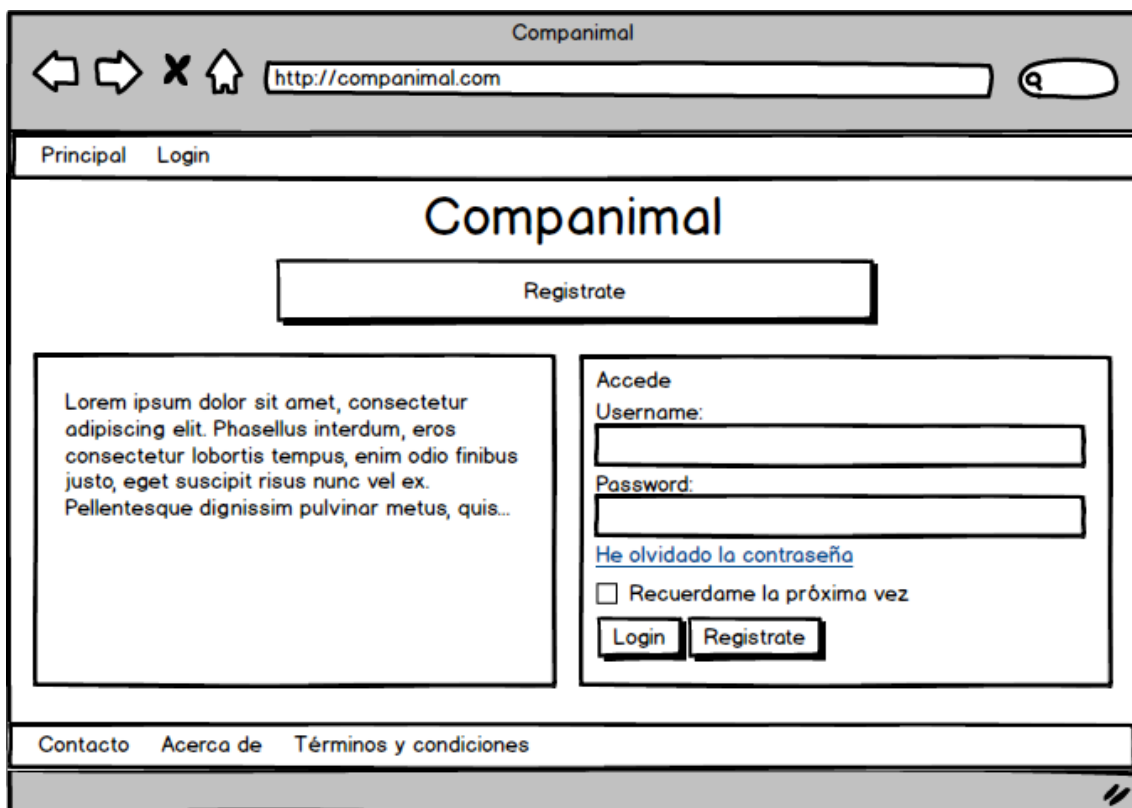


Figura 5: Wireframe portada

En la **Figura 5** podemos observar lo que verá el usuario no identificado que acceda a Comanimal, como podemos observar, las acciones que tiene disponibles son muy

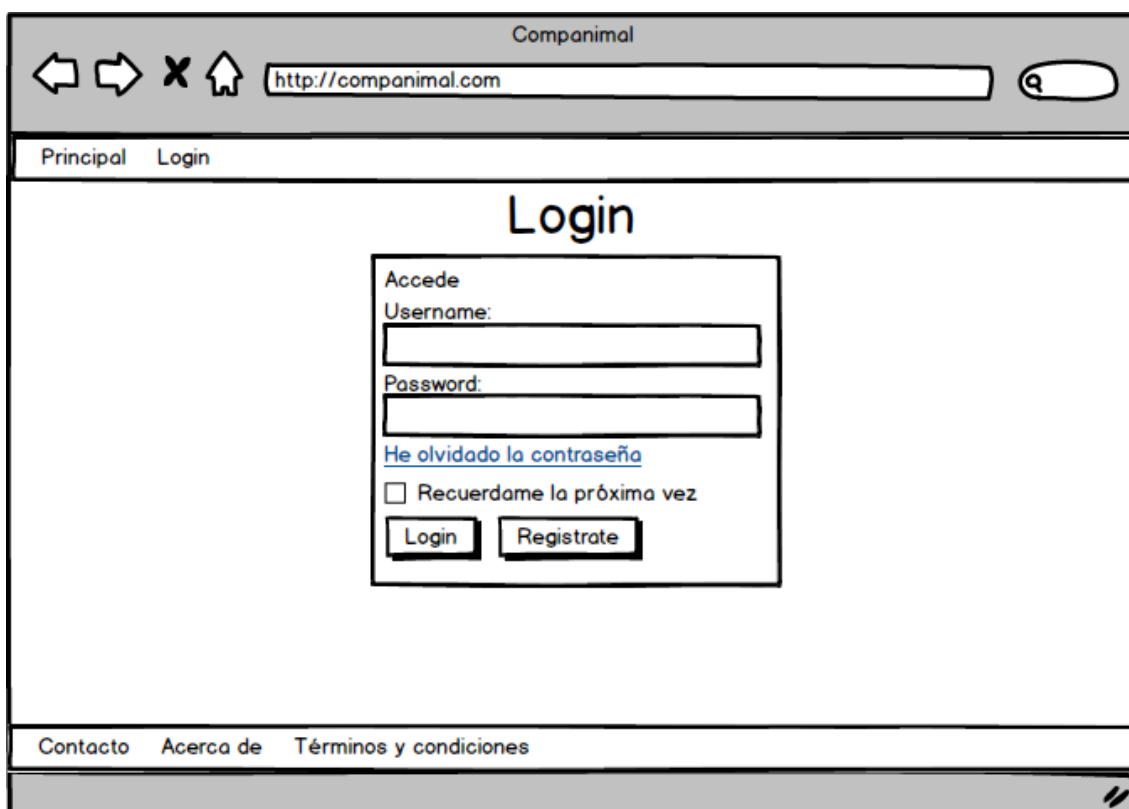


limitadas, debido a que se trata de una red social y para acceder a ella es necesario estar identificado en el sistema.

El usuario anónimo podrá identificarse, bien desde el formulario que se muestra en esta misma página o mediante el botón de *login* que tiene en la barra superior, tiene opciones para recordar contraseña y para que el sistema recuerde que ha iniciado sesión y no tenga que hacerlo la próxima vez, así como se le da la posibilidad en caso de no tener cuenta en el sistema, de crearla, mediante los botones de registro.

En la parte izquierda podemos observar un texto que será texto descriptivo de la funcionalidad de la aplicación, cuyo objetivo será llamar la atención del usuario y explicarle de que trata la web, de una forma rápida.

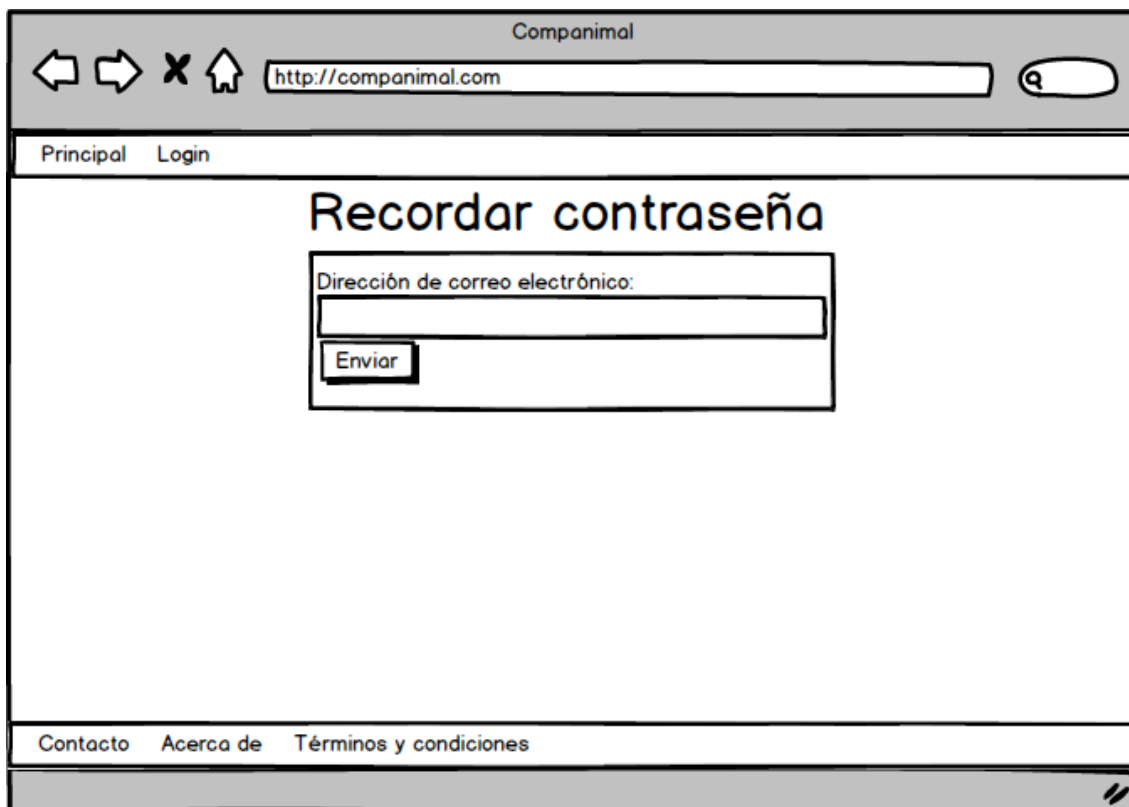
En el *footer* tendrá las secciones de contacto, acerca de y los términos y condiciones, por si son de interés o quiere contactar con nosotros si tiene problemas de registro, inicio de sesión o sugerencias de cualquier tipo.



The image shows a wireframe of a web browser window titled 'Companimal'. The address bar contains 'http://companimal.com'. The page has a navigation bar with 'Principal' and 'Login' links. The main content area is titled 'Login' and contains a form with the following elements: 'Accede', 'Username:' with a text input field, 'Password:' with a text input field, a blue link 'He olvidado la contraseña', a checkbox labeled 'Recuérdame la próxima vez', and two buttons: 'Login' and 'Registrate'. The footer contains links for 'Contacto', 'Acerca de', and 'Términos y condiciones'.

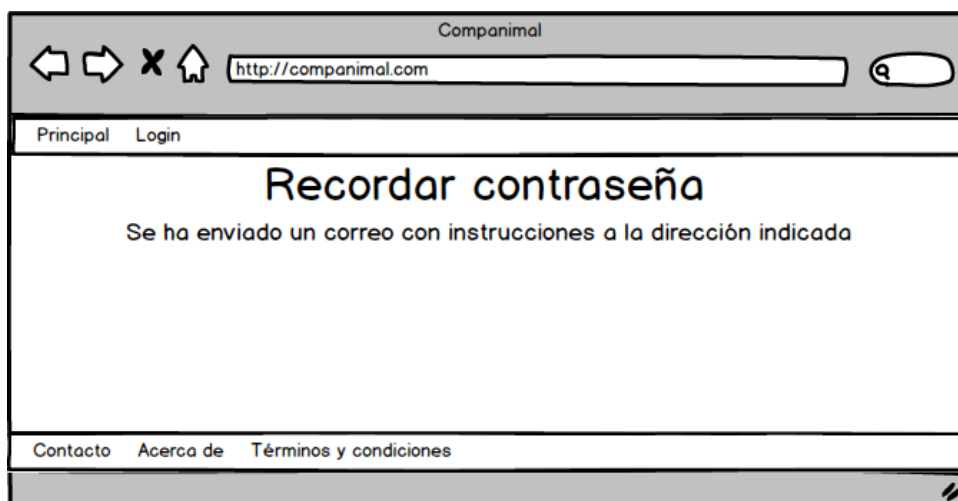
**Figura 6: Wireframe formulario de login**

Cuando el usuario sin identificar haga click sobre el botón de la barra de navegación “*Login*” se le mostrará una vista similar a la **Figura 6** en la que tendrá el mismo formulario que en la página principal mostrada en la **Figura 5** pero sin tanta distracción.



**Figura 7: Wireframe formulario recordar contraseña**

Si el usuario hace click en la opción de “Recordar contraseña” mostrada tanto en la **Figura 5** como en la **Figura 6**, se le mostrará una vista como la de la **Figura 7** en la cual introduciendo su dirección de correo electrónico podrá generar una nueva contraseña y recibirlo en él en el caso de estar registrada en el sistema.



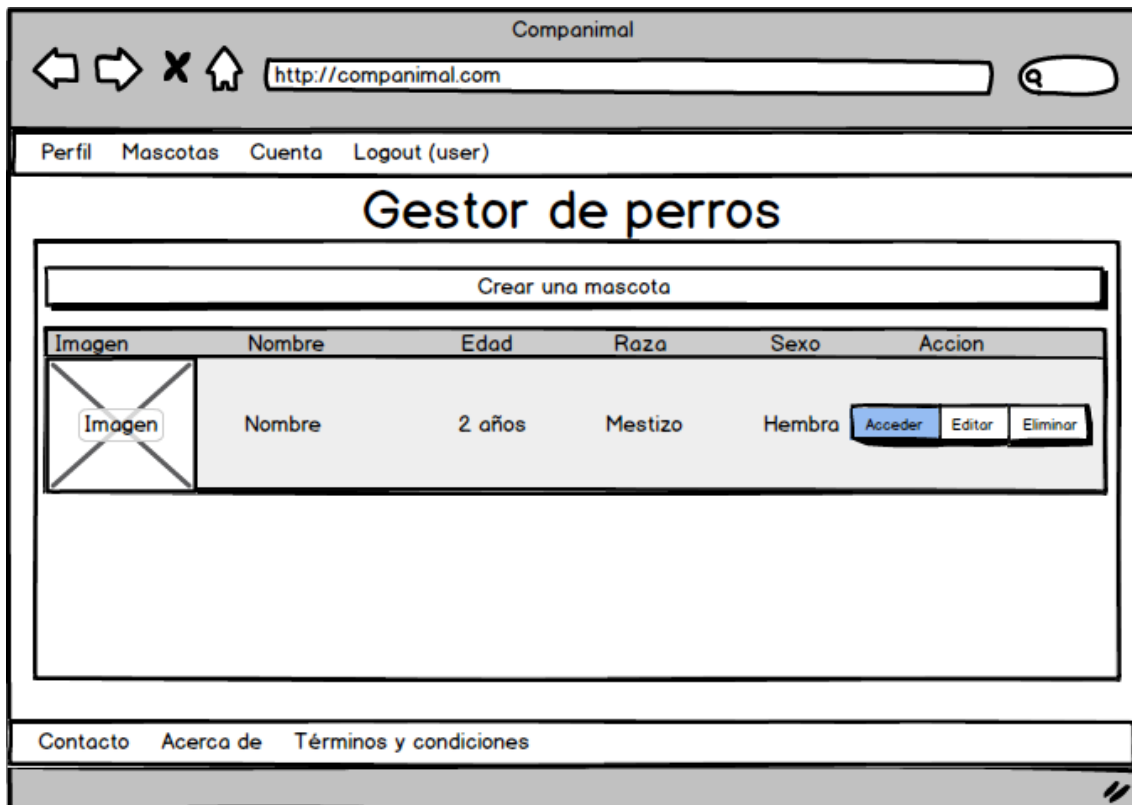
**Figura 8: Wireframe confirmación de cambio de contraseña**

En el caso de solicitar el cambio de contraseña, la ventana de confirmación es la **Figura 8**.

The image shows a wireframe of a web browser window. The browser's title bar reads 'Companimal'. The address bar contains 'http://companimal.com'. Below the browser window, there is a navigation bar with links for 'Principal' and 'Login'. The main content area features a large heading 'Registrar una nueva cuenta'. Below this heading is a registration form with the following fields: 'Nombre de usuario:', 'Email:', 'Password:', and 'Repite password:'. Each field is represented by a horizontal input box. At the bottom of the form is a button labeled 'Crear'. Below the main content area, there is a footer with links for 'Contacto', 'Acerca de', and 'Términos y condiciones'.

**Figura 9: Wireframe formulario de registro**

Si el usuario, desde la **Figura 5** o la **Figura 6** hace click en uno de los botones de “Regístrate” se le mostrará una ventana similar a la **Figura 9** en la que mediante un formulario, el usuario podrá crear una cuenta en el sistema para acceder al mismo a partir de ese momento. En este formulario simplemente se le solicita un nombre de usuario para distinguir la cuenta del resto de usuarios, una dirección de correo electrónico con la que comunicarnos y la contraseña, con la correspondiente verificación de la misma.



**Figura 10:** Wireframe gestor de perfiles

Una vez el usuario se ha identificado en el sistema, lo primero que verá en pantalla será la **Figura 10**, lo que se ha denominado “Gestor de perfiles” o “Gestor de perros”, como se puede observar, al estar identificado en el sistema, se ha cambiado las acciones disponibles de la barra de navegación, ahora puede acceder al perfil de la mascota actualmente en uso mediante la opción “Perfil”, también puede volver en cualquier momento al gestor de la **Figura 10** mediante la opción “Mascotas” así como acceder a la pantalla de edición de la cuenta de usuario mediante la opción “Cuenta”, por último el usuario podrá salir de la aplicación mediante la opción de “Logout” de la barra.

En la parte central observamos un botón que ocupa todo el ancho del panel mediante el cual podremos dar de alta a nuestras mascotas en el sistema, debajo del mismo observamos como en una tabla se nos muestran las mascotas que ya tenemos registradas y las acciones asociadas a las mismas, que en este punto son “Acceder” mediante la cual entraríamos en la plataforma como la mascota, “Editar”, mediante la cual editaríamos la información referente a la mascota y “Eliminar” mediante la cual se borraría la mascota del sistema para siempre.

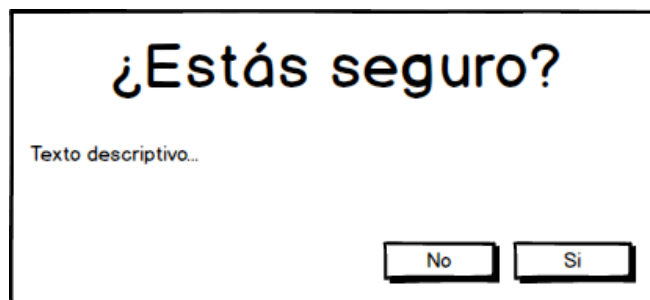


Figura 11: Wireframe popup de confirmación

La operación de borrado siempre es delicada así que le mostramos al usuario un dialogo para que confirme la operación que estará a punto de realizar en caso de que le dé al botón “Borrar” para evitar borrados accidentales. El dialogo es el mostrado en la **Figura 11**.

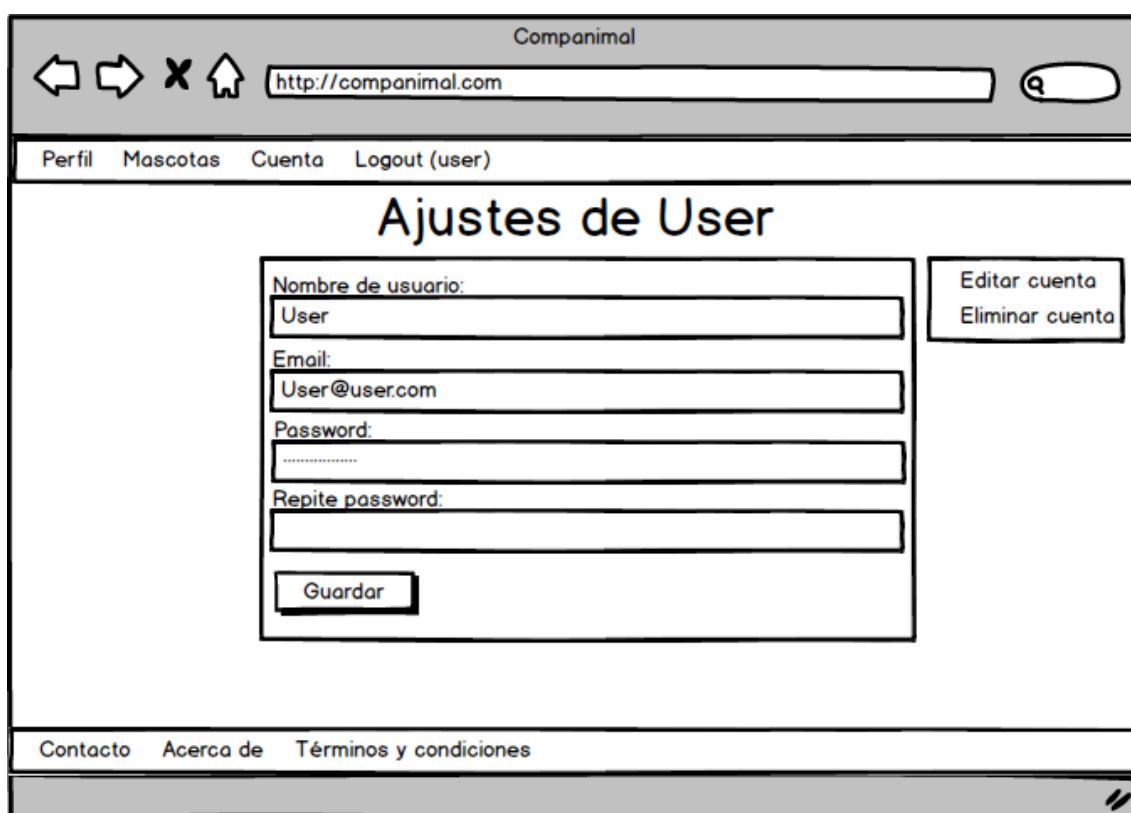


Figura 12: Wireframe formulario edición de cuenta

En la **Figura 12** se muestra la vista que aparecerá al usuario cuando haga click sobre la opción de la barra de navegación de “Cuenta” una vez se ha identificado en el sistema. Como se puede observar, se trata de un formulario similar al de registro mostrado en la **Figura 9**, con la particularidad de que ya está relleno con la información del usuario. Desde aquí podrá cambiar sus datos, incluso su contraseña, guardará los cambios mediante un click en el botón de “Guardar”.

En la parte lateral derecha veremos acciones que podemos realizar sobre la cuenta, la opción de eliminar tiene un dialogo de confirmación similar al mostrado en la **Figura 11**.

The image shows a wireframe of a web browser window for 'Companimal'. The browser's address bar shows 'http://companimal.com'. The navigation menu includes 'Perfil', 'Mascotas', 'Cuenta', and 'Logout (user)'. The main heading is 'Registrar una nueva mascota'. The form contains the following fields and options:

- Nombre: [Text input field]
- Raza: [Dropdown menu with 'Mestizo' selected]
- Sexo:
  - Macho
  - Hembra
- Color: [Text input field]
- Constitución: [Text input field]
- Edad: [Text input field]
- Provincia: [Dropdown menu with 'Valencia' selected]
- Adoptado:
  - Si
  - No
- Imagen de perfil: [Image upload button]
- Crear: [Submit button]

The footer contains links for 'Contacto', 'Acerca de', and 'Términos y condiciones'.

**Figura 13: Wireframe formulario registro de nueva mascota**

Cuando desde la **Figura 10** se hace click en “Crear mascota” aparecerá una vista similar a la de la **Figura 13**, en ella veremos un formulario que deberemos completar si queremos registrar una nueva mascota en el sistema, en el caso de seleccionar una imagen de perfil para ella, nos aparecerá un *popup* como el mostrado en la **Figura 14** en el cual podremos recortar la imagen mediante pinchar y arrastrar y subirla a la aplicación.

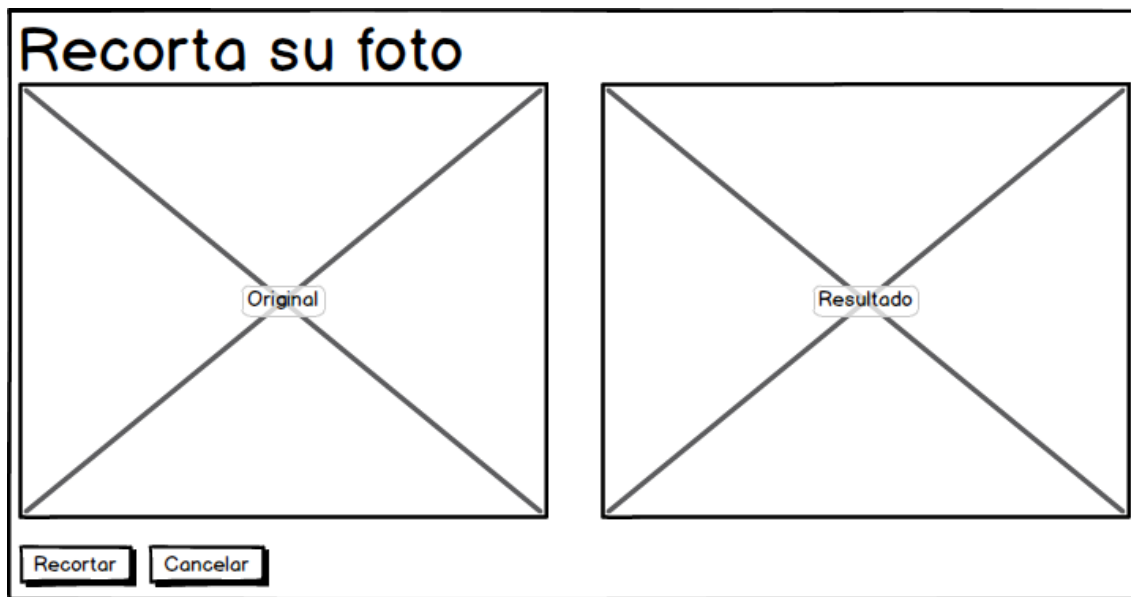


Figura 14: Wireframe popup recortar imagen

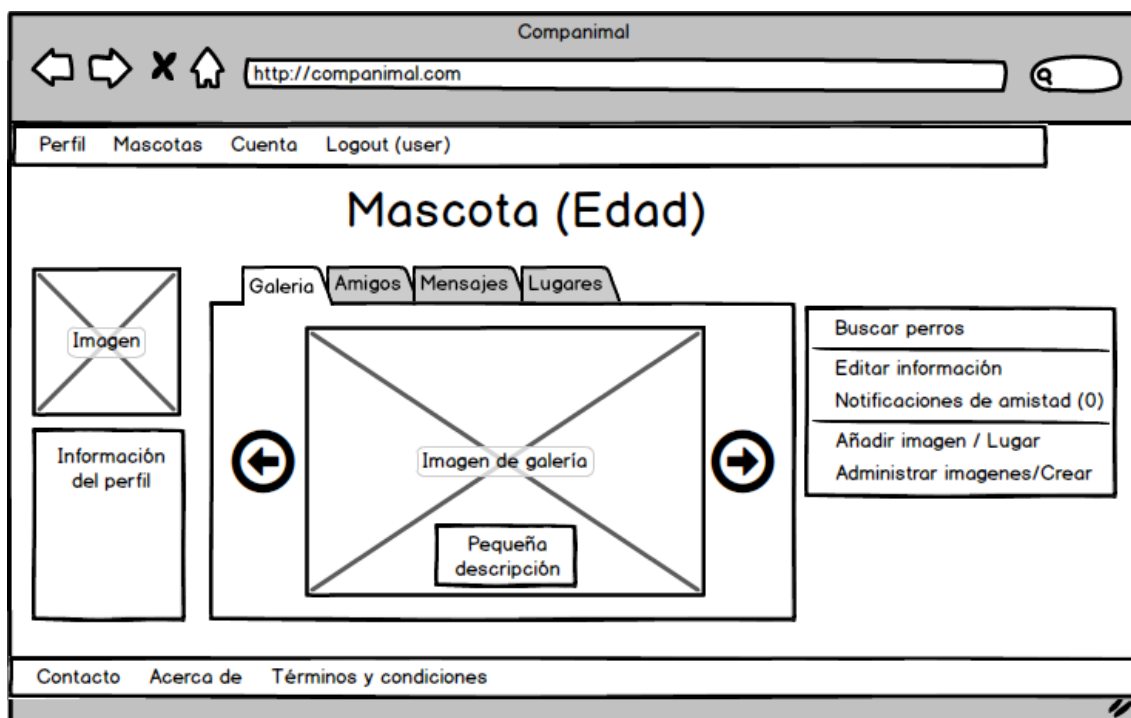
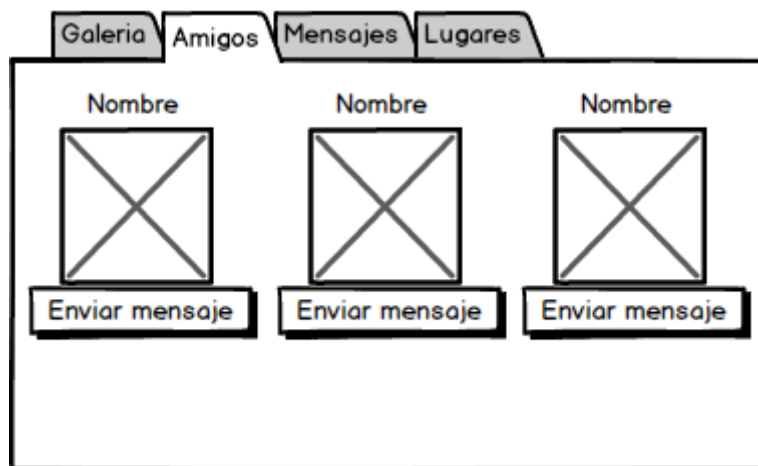


Figura 15: Wireframe perfil de mascota

En la **Figura 15** se puede ver cómo sería la página de perfil de las mascotas de la red, la información mostrada en el panel de la derecha variará si la mascota es de nuestra propiedad o no, siéndonos imposible de realizar todas las acciones menos la de buscar perros si la mascota no es nuestra, en función de la pestaña que tengamos seleccionada en el panel central las dos últimas opciones de este menú lateral variarán, adecuándose a las que correspondan en cada momento. En cuanto al panel de pestañas que encontramos en la parte central, tampoco será posible ver los mensajes del animal si no se trata de un animal nuestro. Si la mascota que estamos visualizando no es de nuestra propiedad podremos añadirla como amigo con una opción que aparecerá en el menú derecho.

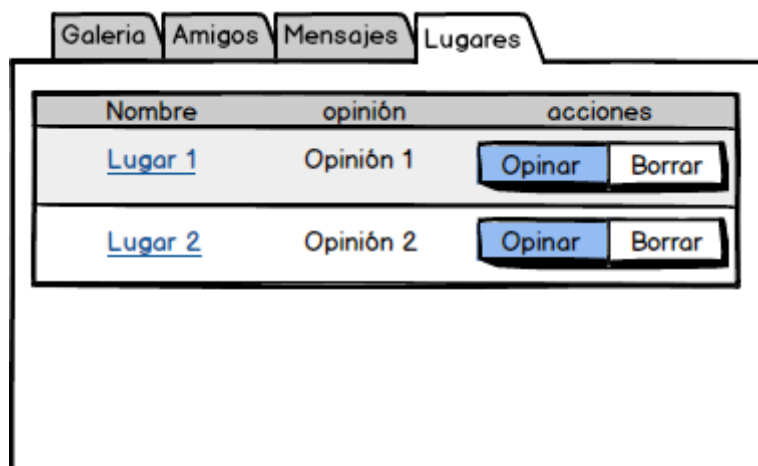
En las pestañas, la primera corresponde a la galería de imágenes de la mascota en cuestión, mediante las flechas que aparecen podremos cambiar entre las fotografías y ver una pequeña descripción de las mismas en cada una de ellas, las otras pestañas corresponden a los amigos de la mascota y a los lugares que ha visitado, el contenido de las mismas se muestra en las **Figura 16** y **Figura 17** respectivamente, mientras que el contenido de la pestaña privada “Mensajes” se muestra en la **Figura 18**.

En la columna izquierda de esta vista observamos la foto de perfil de la mascota en cuestión y sus características más destacables. Como titular de la página se muestra el nombre de la mascota acompañado de su edad.



**Figura 16: Wireframe pestaña de amigos**

En la pestaña de amigos mostrada en la **Figura 16** se verá una lista de amigos, con el nombre, la foto de perfil y un botón mediante el cual al hacer click se nos abrirá un *popup* como el de la **Figura 19** para enviarles un mensaje.



**Figura 17: Wireframe pestaña de lugares**

En la **Figura 17** se puede ver cómo sería la pestaña correspondiente a los lugares visitados, veríamos una vista tipo tabla en la que se mostraría el nombre del lugar en forma de enlace, que nos llevaría a la vista detallada del mismo, la opinión acerca del sitio de la mascota y las acciones que se pueden realizar sobre el lugar, en caso de que exista una opinión no se mostraría la opción de opinar.



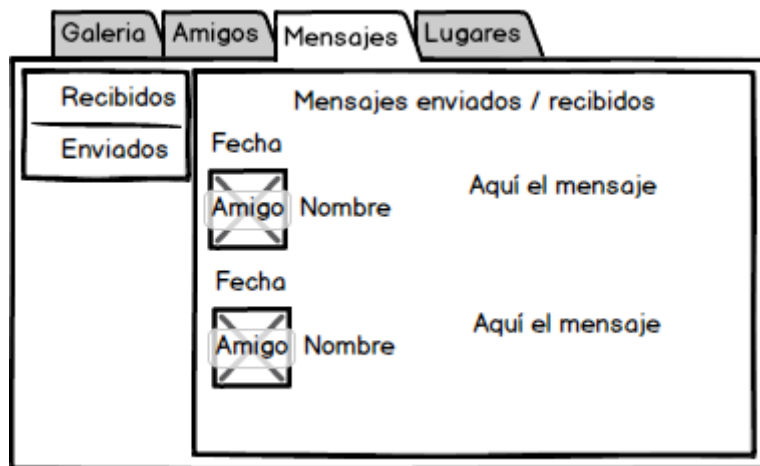


Figura 18: Wireframe pestaña de mensajes

En la **Figura 18** se observa cómo queda la pestaña de los mensajes, en esta pestaña tenemos la opción de elegir ver los mensajes recibidos o los enviados, en función de lo que seleccionemos, nos cambiará la información del panel derecho, sin embargo, la forma de visualizar la información de este panel es siempre la misma, la fecha del mensaje acompañada de la foto de perfil del remitente y el nombre y en una segunda columna el texto del mensaje.

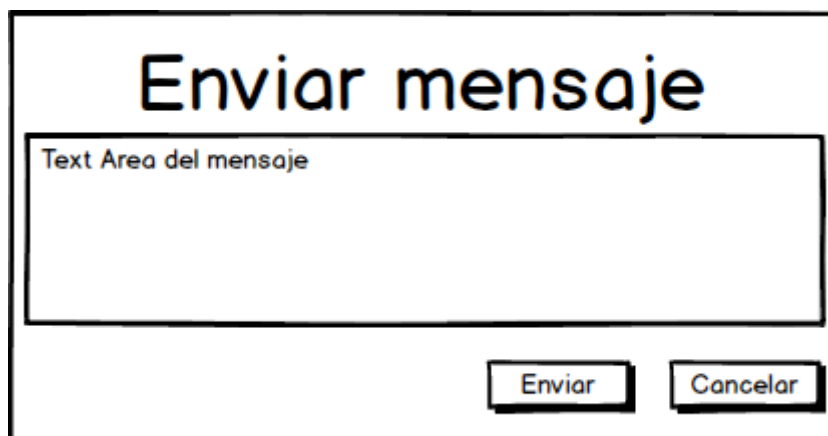


Figura 19: Wireframe popup envío de mensajes

En la **Figura 19** lo que se ve es el *popup* que se muestra al usuario cuando quiere enviar un mensaje.

Imagen	Nombre	Dueño	Fecha	Acción
Imagen	Nombre 1	Dueño 1	Fecha 1	Aceptar Rechazar
Imagen	Nombre 2	Dueño 2	Fecha 2	Aceptar Rechazar

Figura 20: Wireframe popup notificaciones de amistad

En la **Figura 20** se observa el *popup* que se muestra al usuario cuando desde la vista de la **Figura 15** hacen click en la opción de la barra lateral derecha “Notificaciones de amistad” desde este *popup* podrán aceptar o rechazar todas y cada una de las notificaciones de amistad que tengan.

**Añadir nueva imagen**

Título:

Imagen

Descripción:

Guardar Cancelar

Figura 21: Wireframe popup añadir imagen

En la **Figura 21** se puede ver el *popup* que nos saldría si desde cualquier parte de la aplicación queremos subir una imagen para la galería de imágenes de una mascota, se trata de un formulario donde deberemos introducir el título de la imagen que vamos a subir, al hacer click sobre el botón de “Subir imagen” nos aparecerá otro *popup* como el de la **Figura 14** para que añadamos la foto y cuando la estemos subiendo al sistema, la barra de progreso de la parte inferior del *popup* se irá rellenando. Por último solicitamos al usuario una breve descripción o comentario acerca de la imagen.

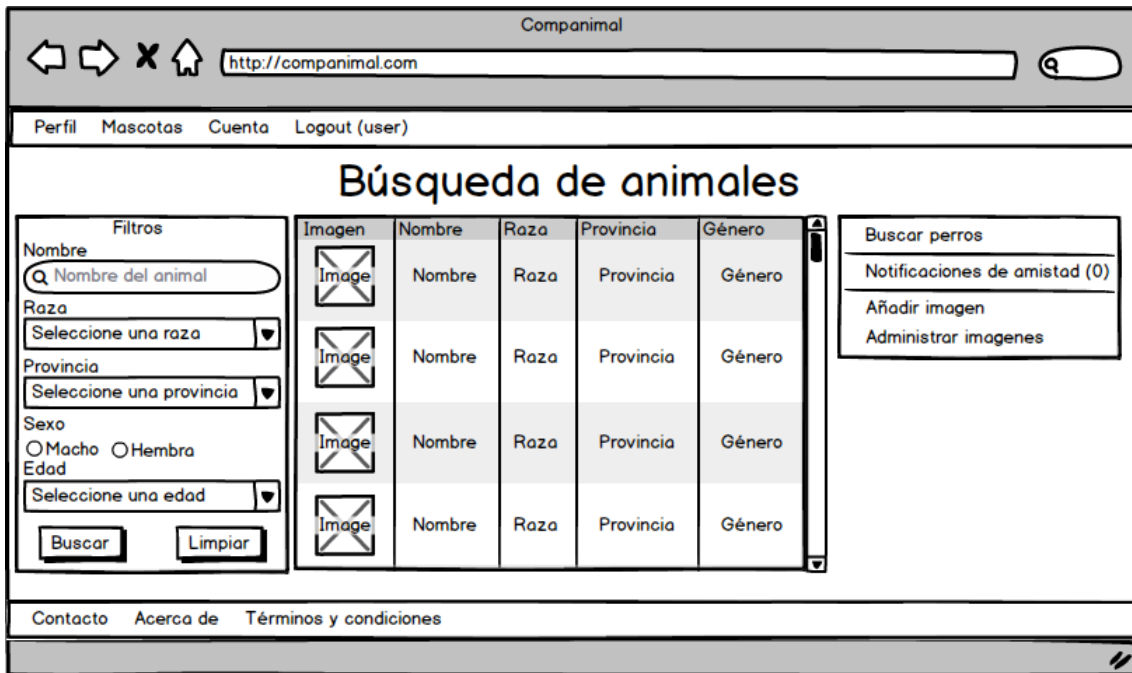


Figura 22: Wireframe búsqueda de animales

La **Figura 22** corresponde con la búsqueda de animales en el sistema, como se puede observar, los filtros los encontramos en la parte izquierda de la ventana, en función de los filtros que seleccionemos, cuando le demos al botón de “Buscar” nos cambiara el contenido de la tabla central y mediante un click en la foto del animal seleccionado, nos iremos a su página de perfil donde podremos añadirle como amigo.

Acerca de la barra lateral derecha no hay mucho que decir, es la que está presente en casi todas las vistas de la web, lo único que cambia entre unas vistas y otras son las opciones presentes.

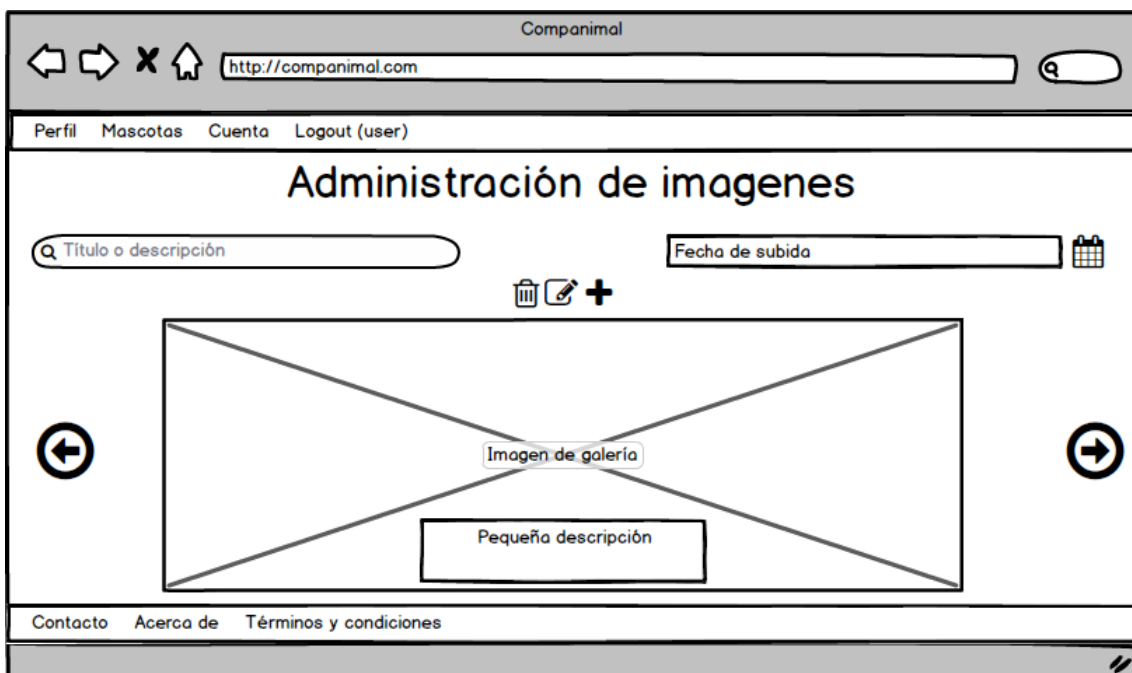


Figura 23: Wireframe administración de imagenes

En la **Figura 23** observamos la interfaz de administración de imágenes, desde esta vista podremos buscar imágenes por su nombre o descripción y por la fecha en la que fueron subidas al sistema. Además, podremos verlas y realizar acciones sobre ellas, como editarlas o eliminarlas.

Al añadir una imagen nueva se nos mostrará el *popup* correspondiente a la **Figura 21**

El wireframe muestra una interfaz de usuario en un navegador web. El navegador tiene el título 'Companimal' y la URL 'http://companimal.com'. La barra de navegación superior contiene los enlaces 'Perfil', 'Mascotas', 'Cuenta' y 'Logout (user)'. El contenido principal está dividido en dos secciones: a la izquierda, un cuadro con una 'X' diagonal y el texto 'Imagen' en el centro; a la derecha, un formulario con los campos 'Título' y 'Descripción', y un botón 'Guardar' debajo. La barra de pie inferior contiene los enlaces 'Contacto', 'Acerca de' y 'Términos y condiciones'.

**Figura 24: Wireframe edición de imagen**

En la **Figura 24** podemos observar la vista de edición de imágenes, en esta vista únicamente podemos cambiar el título y la descripción de la imagen, además de las opciones ya mencionadas de la barra de navegación y el *footer*.



**Figura 25: Wireframe creación de un lugar**

En la **Figura 25** se puede ver cómo queda la vista de creación de lugares, se trata de un formulario en el que introduciremos el nombre con el que se identificará de manera interna el lugar y lo buscaremos también en un mapa de google para poder localizarlo y que esté al alcance de todos los usuarios de la plataforma, además de añadir una pequeña descripción del sitio para orientar a futuras personas que quieran acercarse.

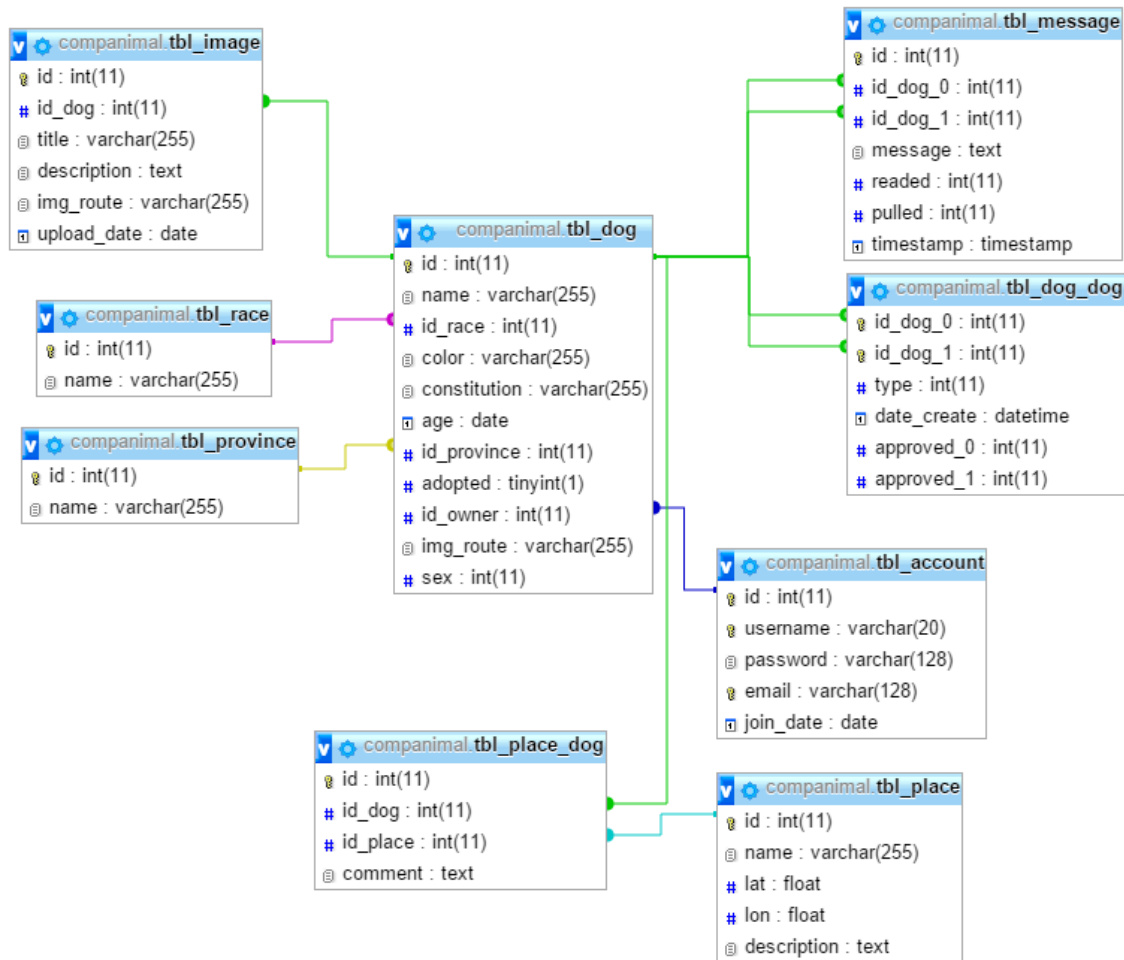


**Figura 26:** Wireframe vista detallada de un lugar

Cuando entramos en la vista detallada de algún lugar existente veremos algo similar a la **Figura 26** donde podremos saber el nombre del lugar que estamos visualizando en pantalla, un mapa con la localización del mismo señalada y en la parte derecha podremos ver las opiniones de los diferentes usuarios que lo han visitado y han dado su opinión en la plataforma.

## 4.2. Capa de persistencia

Tras analizar las necesidades del sistema el esquema relacional que representa mi base de datos es el mostrado a continuación en la **Figura 27**:



**Figura 27: Esquema de base de datos relacional**

Aquí podemos observar como cualquier tabla presente en la base de datos tiene relación con la tabla de los perros, denominada *tbl\_dog*, debido a que todo en nuestro sistema gira en torno a ellos.

De los animales almacenamos un nombre, su raza, el color, la constitución, la edad, de donde són, si son adoptados, quien es el propietario, la ruta donde está su imagen de perfil y el sexo. Ciertos campos queremos que estén presentes en la búsqueda por lo tanto para que las búsquedas sean satisfactorias ciertos campos no podemos dejar que el usuario los introduzca de forma liberal, para ello se crean tablas donde iremos almacenando los términos por los cuales más tarde querremos buscar, estas tablas son *tbl\_province* y *tbl\_race*. En *tbl\_dog* se guarda la id correspondiente a la fila de la otra tabla, se establece una relación de clave ajena con *id\_race* y *id\_province* respectivamente.

La cuenta de usuario viene representada en la tabla *tbl\_account*, en la cual nos guardamos el nombre de usuario, la contraseña cifrada, el email y la fecha en la que se registró en la plataforma. Podemos saber quién es el propietario de cada mascota estableciendo una relación de clave ajena, almacenamos en *tbl\_dog* el id del propietario en el atributo *id\_owner*.

Como ya se ha mencionado, además de la foto de perfil, que al solo poder tener una a la vez se almacena directamente en *tbl\_dog*, las fotos de la galería necesitan de una tabla aparte en la que nos guardamos la id del perro al que hacen referencia (clave ajena), el título que se le ha puesto a la imagen, una breve descripción, la ruta donde está almacenada y la fecha de subida.

Las relaciones entre los animales, al tratarse de una relación muchos a muchos ha sido necesario una tabla intermedia *tbl\_dog\_dog* donde almacenamos los identificadores de los animales relacionados (claves ajenas), el tipo de relación pensando que en un futuro puedan establecerse tanto relaciones de amistad como de parentesco o cualquier otro tipo, la fecha de creación y si los dos extremos de la relación han aceptado la petición.

En cuanto a la comunicación entre animales se representa en la base de datos de una forma similar a las relaciones entre perros, guardamos ambos identificadores de los animales que están teniendo un intercambio de mensajes como claves ajenas y el texto del mensaje, además, como variables de control de la conversación almacenamos si el mensaje ha sido leído, cargado y el tiempo UNIX en el cual fue enviado.

En cuanto a los lugares, para representarlos únicamente hemos tenido que crear una tabla *tbl\_place* donde almacenamos el identificador del lugar, el nombre, la latitud y la longitud que nos las solicita Google para montar el mapa y una breve descripción del lugar.



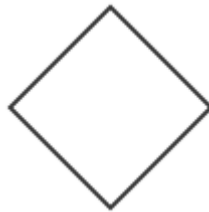

La relación entre los perros y los lugares es de muchos a muchos, por lo tanto nos hemos tenido que ayudar de una tabla intermedia *tbl\_place\_dog* en la cual además guardamos el atributo “*comment*” correspondiente a la relación, nos guardamos los identificadores del perro y del lugar, para poder relacionarlos.

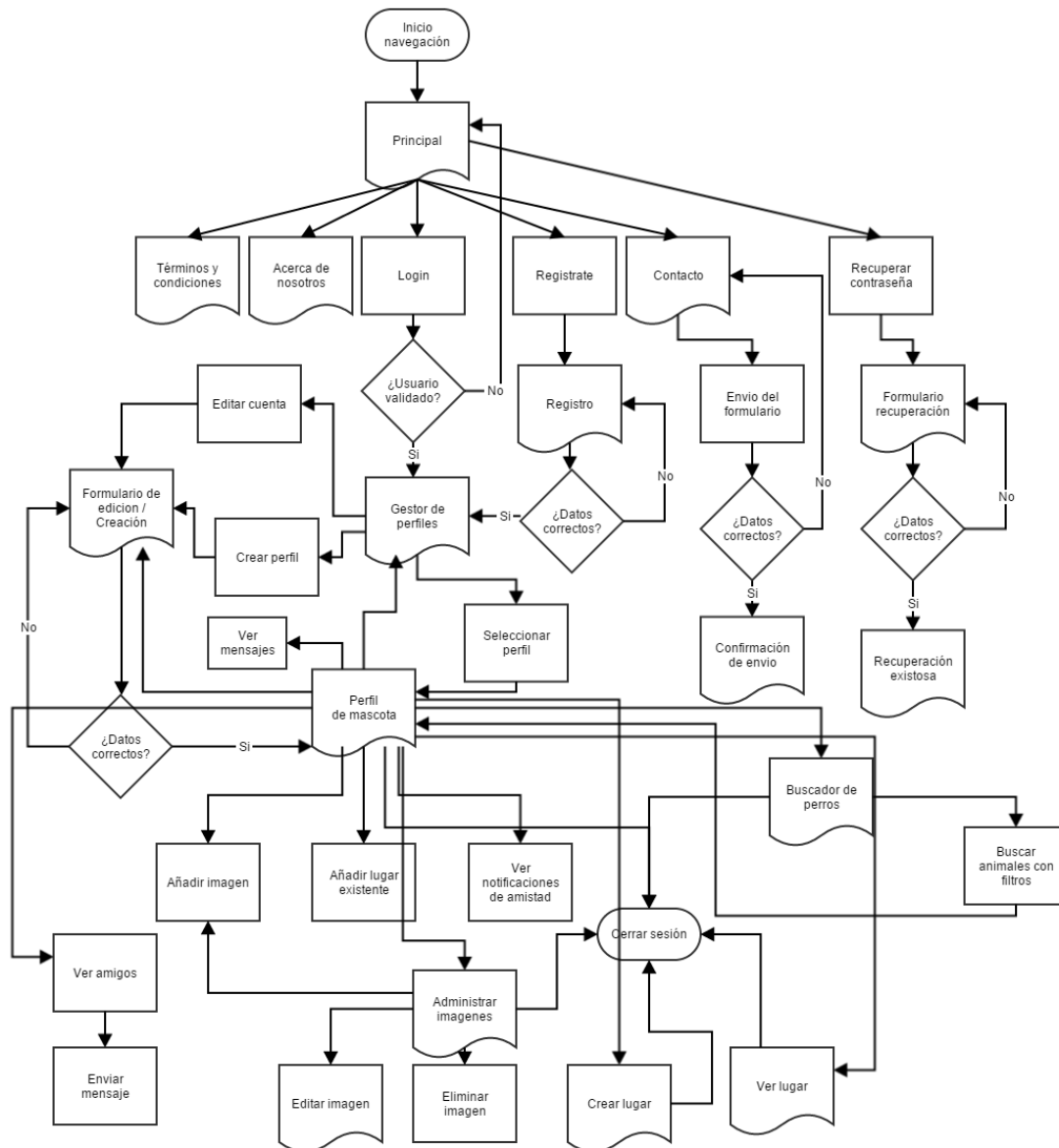
Esta arquitectura de base de datos es suficiente para el alcance del proyecto cubriéndose todos los aspectos y objetivos que se proponen al inicio.



### 4.3. Lógica de la aplicación

En esta sección se explicará mediante un diagrama de flujo el comportamiento de la página y las acciones que se pueden realizar desde la misma. Antes de mostrar el diagrama explicaremos cada una de las figuras que se van a utilizar.

	Con este icono se determinarán los diferentes documentos que se sirven a los usuarios, es decir, las distintas vistas o páginas de la aplicación.
	Mediante este icono se representarán las acciones que se pueden realizar en la página.
	A través de este símbolo representaremos aquellas situaciones en las que el usuario de nuestra página debe tomar una decisión, en este punto la lógica de la aplicación determinará el camino a seguir en base a la decisión tomada.
	Inicio/Fin, se representa el inicio de la navegación del usuario en nuestra página con este símbolo, así como también representa el fin de la navegación.



**Figura 28: Diagrama de flujo de la web**

En el diagrama de flujo podemos observar como todo gira en torno al perfil de la mascota, siéndonos posible desde ese punto llegar a cualquier parte de la web sin demasiado esfuerzo, también podemos ver que las posibilidades de cerrar sesión están presentes desde cualquier punto de la aplicación, siempre que tengamos la sesión iniciada.

## 5. Detalles de implementación

A continuación se explicarán aquellos aspectos más destacables en lo referente a la implementación del proyecto, se hablará tanto de las tecnologías utilizadas como de la estructura de ficheros y directorios que se ha empleado.

### 5.1. Tecnologías empleadas

En este apartado se hablará de que tecnologías se han utilizado en el desarrollo del proyecto y el motivo por el cual se han utilizado. Desde las características de la máquina utilizada hasta el software utilizado para maquetar los bocetos o hacer los diagramas.

#### 5.1.1. Máquina servidor

El hardware que se ha alquilado a OVH para la realización de este proyecto no ha sido muy potente debido a que el entorno en el que se iba a trabajar es un entorno de desarrollo, sin tener en mente en ningún momento pasar a producción. El servidor alquilado tiene 1GB de memoria RAM un procesador de 1 núcleo y un disco duro de 10GB.

#### 5.1.2. Sistema operativo de la máquina servidor

El sistema operativo por el cual me he decantado ha sido Centos6, en su versión de 32 bits, el motivo por el cual he elegido este sistema operativo es porque la distribución de Linux de Centos está dirigida más a servidores que otras distribuciones como puede ser Ubuntu y al tratarse de un proyecto web he considerado esta distribución la mejor alternativa entre las disponibles en el mercado.

#### 5.1.3. Servidor Web

Como servidor Web se ha optado por Apache. Apache es un servidor HTTP, está disponible para múltiples plataformas y es de código abierto. Actualmente es el servidor predominante en Internet debido a que sus características son muy configurables, haciendo que este servidor pueda adaptarse a prácticamente cualquier entorno.

Actualmente y durante los últimos años Apache está siendo uno de los servidores web más utilizados, uno de los motivos es el soporte de la comunidad de usuarios que lo utilizan y tratan de mejorarlo, también es ampliamente configurable y muy robusto. Este es el motivo principal por el que me he decantado por Apache frente a otro tipo de software como puede ser NGINX.

#### 5.1.4. Persistencia

Para la capa de persistencia del proyecto, se ha optado por MySQL, se trata de un sistema de gestión de bases de datos relacionales, dada la naturaleza del proyecto he pensado que un sistema relacional era mejor en base a mis objetivos que uno no relacional.

Además la elección de un esquema relacional también se ha debido a que son más rápidos a la hora de leer información que los no relacionales. Un proyecto web es un entorno en el que la concurrencia en la actualización de los datos no es muy elevada, sin embargo, la lectura de la información debe ser rápida para que el usuario final espere el menor tiempo posible, por lo tanto un esquema relacional es ideal.



### 5.1.5. Lógica

Para la capa de lógica he decidido que la mejor alternativa era usar PHP, éste es un lenguaje de programación del lado servidor.

La elección de PHP frente a otras opciones más actuales como por ejemplo NodeJS fue por la madurez que tiene PHP, ha demostrado en múltiples ocasiones que es un lenguaje robusto y que posee una enorme comunidad de programadores tan experimentados como generosos a la hora de compartir experiencias.

En cuanto a escalabilidad y rendimiento, he elegido PHP frente a NodeJS porque PHP por cada conexión reserva una pequeña porción de la memoria del sistema, lo cual quiere decir, que cada usuario es independiente del resto. Al contrario que en NodeJS, en PHP no tenemos que preocuparnos de que las acciones de un usuario perjudiquen al resto porque tendrían que haber muchas peticiones concurrentes al mismo tiempo para que esto sucediera.

### 5.1.6. Framework utilizado

Con todo lo anterior establecido, es el momento de elegir un *Framework*, ya que agiliza muchas tareas de programación y ayuda con la estructura de ficheros y directorios, se ha optado por *Yii Framework* debido a que tiene una curva de aprendizaje asumible y una potencia más que aceptable.

Yii te fuerza a programar siguiendo una arquitectura MVC (*Model-View-Controller*) la cual hoy en día puede considerarse un estándar de programación, lo cual es bueno, pues Yii está completamente orientado a esta arquitectura.

Está compuesto de una serie de librerías y herramientas que hacen que la programación sea más sencilla, especial mención al mapeo objeto-relacional que hace de manera automática si optamos por programar mediante sus AR (*Active Records*) que son autogenerados a partir de nuestro esquema de base de datos.

## 5.2. Estructura de ficheros y dependencias

He creado en el servidor un subdominio llamado *Companimal* dentro del cual hay una carpeta que es la raíz del proyecto, cuyo nombre es “*companimal*”.

En nuestro caso, la estructura de ficheros y directorios es automáticamente generada por el *framework* empleado siguiendo el patrón de arquitectura MVC.

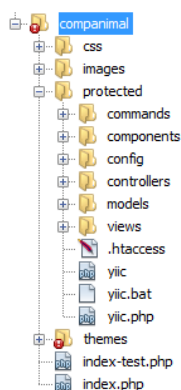


Figura 29: Estructura de ficheros y directorios Yii Framework

### 5.2.1. Modelo-Vista-Controlador (MVC)

MVC es un patrón de diseño que separa los tres componentes básicos de una aplicación para estructurar la implementación y facilitar su mantenimiento. A lo largo del proyecto me he basado en estos distintos componentes a la hora de analizar y diseñar la aplicación, pensando en facilitar una futura implementación.

- **El modelo:** Este componente es el encargado de representar la información contenida en la base de datos. Yii posee una herramienta que autogenera los modelos y además, facilita su acceso y su manipulación mediante lo que él denomina *ActiveRecords (AR)*, se trata de una técnica de mapeo objeto-relacional en la que cada AR representa una tabla de la base de datos cuyos atributos son representados como propiedades del AR y una instancia de AR representa una fila de la tabla. Las operaciones CRUD (*Create Read Update Delete*) relacionadas con cada una de las tablas están representadas como funciones dentro de los AR.
- **La vista:** Hace referencia a la interfaz de usuario, a como mostrar la información al usuario, en Yii las vistas están dentro de la carpeta *views* y organizadas en función del controlador al que hacen referencia.
- **El controlador:** El controlador se situaría entre las vistas y los modelos, procesando las solicitudes, llamando a los modelos y ejecutando las funciones necesarias en cada caso, y montando las vistas necesarias para mandarle al usuario y que pueda visualizar la página de manera adecuada.

Los ficheros que conforman estas tres capas que he mencionado están almacenados dentro del servidor dentro del directorio “*protected*” el cuál posee en su interior un fichero *.htaccess* el cual tiene como objetivo que ninguno de estos ficheros sea accesible desde el exterior, porque todo el procesamiento se va a llevar a cabo desde el controlador. Esto es una medida de seguridad estupenda ya que bajo ninguna circunstancia se van a permitir conexiones de ningún tipo a ficheros de configuración, modelos de datos, o controladores.

Los controladores están dentro de la carpeta *protected*, entonces, un usuario final no puede acceder a ellos, el usuario final accede a los controladores necesarios a través de lo que Yii denomina el controlador frontal (*Front Controller*). La función de este controlador es la de recibir peticiones del usuario y transmitir las a los controladores adecuados que se encargarán de procesarlas.

### 5.2.2. Extensiones

Yii es un *framework* que tiene muchas extensiones que facilitan las labores de desarrollo, sin embargo únicamente se ha empleado la extensión denominada “*gettext*” que nos facilita la traducción de la web mediante ficheros de extensión *.po* y *.mo*. Se ha hecho únicamente a modo de demostrar lo sencillo que es traducir una web con esta tecnología y por lo tanto sólo se ha traducido la portada, la portada cambia de idioma según el idioma del navegador que está visualizándola, actualmente sólo hay dos posibilidades, español e inglés pero puede ampliarse a todas las que se quiera y aplicar cada uno en la ocasión que se quiera.



### 5.3. Software empleado en el desarrollo

En la máquina servidor además del servidor web también ha sido necesario instalar Virtualmin para facilitar la gestión del servidor, la creación de subdominios, cuentas de correo electrónico, servidor FTP, etc.

En cuanto al IDE empleado para desarrollar el producto se ha empleado NetBeans en su versión de PHP y HTML5, he decidido utilizar NetBeans en su versión 8.0.2 debido a que se trata de software gratuito y que tiene la posibilidad de trabajar contra un servidor de FTP, haciendo mucho más ágil el trabajo al no tener que estar continuamente preocupándose de subir los cambios, los cambios se suben de manera automática cada vez que se guarda.

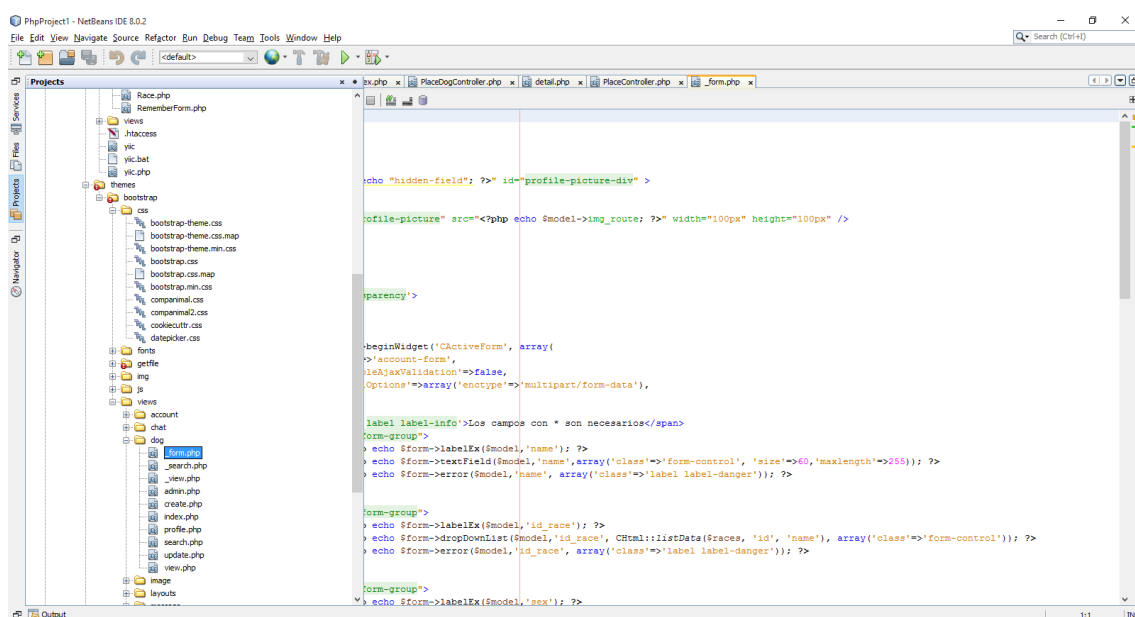
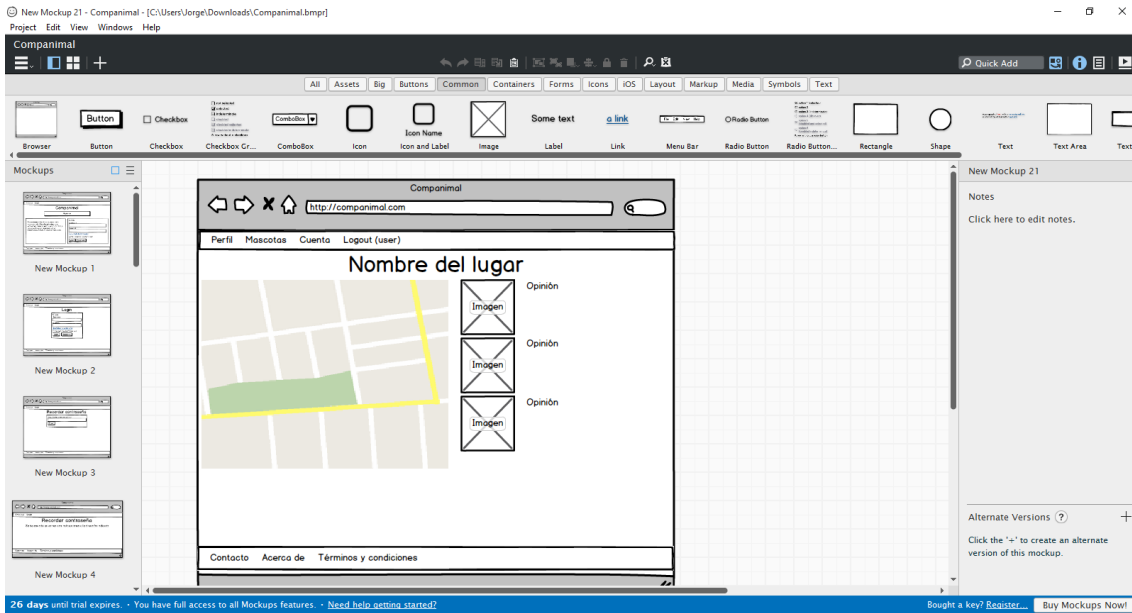


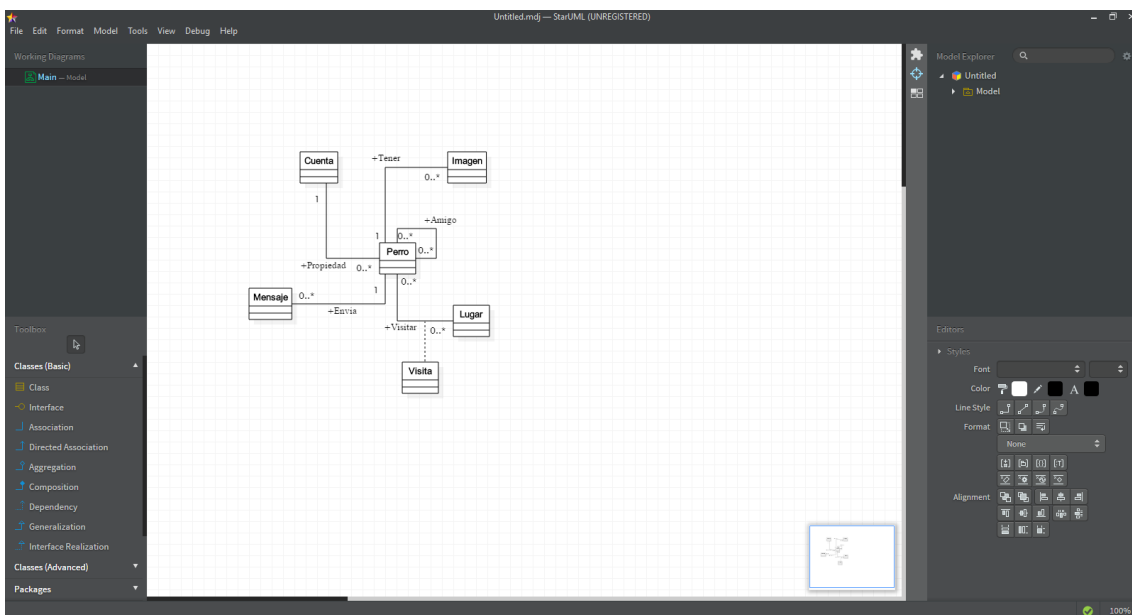
Figura 30: Captura de pantalla IDE NetBeans

Para los mockups he empleado un software en su versión de prueba, denominado “Balsamiq Mockups” el cual te permite definir los bocetos de cualquier tipo de aplicación de una manera fácil y rápida, de esta forma la toma de decisiones referentes al diseño de la aplicación se llevan a cabo de una forma rápida y sin demasiado esfuerzo.



**Figura 31: Captura de pantalla Balsamiq Mockups**

Para el diagrama entidad-relación que aparece en el presente documento se ha utilizado un software en su versión de prueba llamado StarUML el cual permite dibujar este tipo de diagramas sin mucho esfuerzo y de manera rápida.



**Figura 32: Captura de pantalla StarUML**

Para los diagramas de casos de uso y el diagrama de flujo se ha empleado una herramienta distinta, en este caso se ha empleado una herramienta online denominada Gliffy la cual es muy potente y tiene una versión gratuita con funcionalidad limitada pero suficiente para las tareas que se han desarrollado con ella.



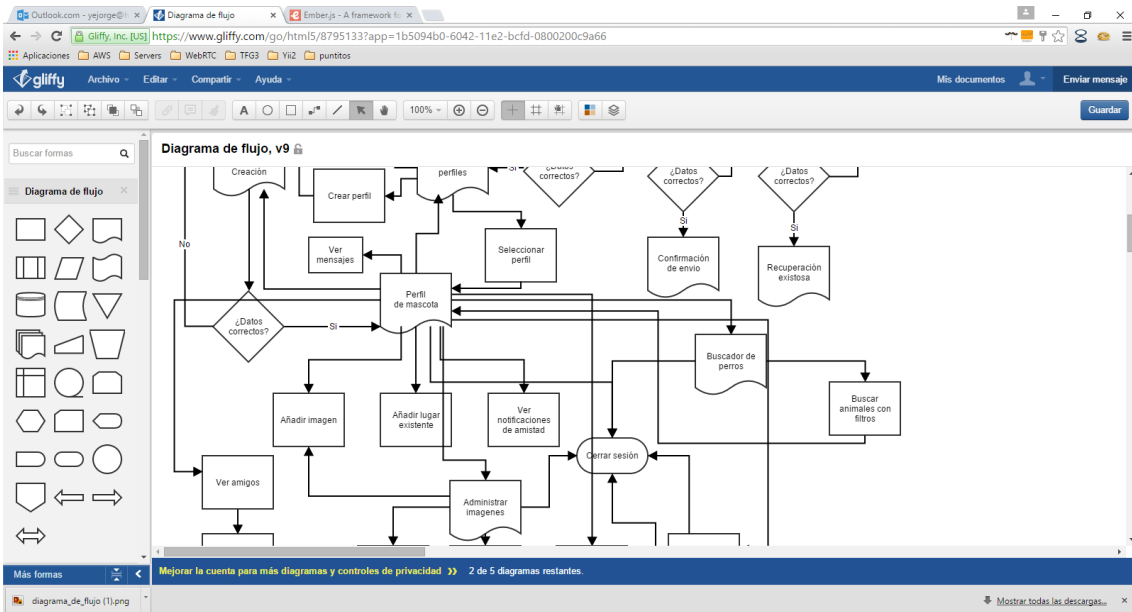


Figura 33: Captura de pantalla Gliffy

Para el diseño de la capa de persistencia se ha empleado la herramienta web “phpmyadmin” y la figura mostrada esta extraída de manera automática de este software.

The screenshot shows the phpMyAdmin interface. The top navigation bar includes 'Structure', 'SQL', 'Search', 'Query', 'Export', 'Import', 'Operations', 'Routines', 'Events', and 'Triggers'. The main content area displays a table structure for the 'companimal' database. The table list is as follows:

Table	Action	Rows	Type	Collation	Size	Overhead
tbl_account	Browse Structure Search Insert Empty Drop	-5	InnoDB	latin1_swedish_ci	48 K1B	-
tbl_chat	Browse Structure Search Insert Empty Drop	-8	InnoDB	latin1_swedish_ci	48 K1B	-
tbl_dog	Browse Structure Search Insert Empty Drop	-4	InnoDB	latin1_swedish_ci	64 K1B	-
tbl_dog_dog	Browse Structure Search Insert Empty Drop	-3	InnoDB	latin1_swedish_ci	32 K1B	-
tbl_image	Browse Structure Search Insert Empty Drop	-5	InnoDB	latin1_swedish_ci	32 K1B	-
tbl_message	Browse Structure Search Insert Empty Drop	-2	InnoDB	latin1_swedish_ci	48 K1B	-
tbl_place	Browse Structure Search Insert Empty Drop	-3	InnoDB	latin1_swedish_ci	16 K1B	-
tbl_place_dog	Browse Structure Search Insert Empty Drop	-2	InnoDB	latin1_swedish_ci	48 K1B	-
tbl_province	Browse Structure Search Insert Empty Drop	-2	InnoDB	latin1_swedish_ci	16 K1B	-
tbl_race	Browse Structure Search Insert Empty Drop	-3	InnoDB	latin1_swedish_ci	16 K1B	-
10 tables	Sum		MyISAM	latin1_swedish_ci	368 K1B	0 B

Below the table list, there is a 'Create table' form with fields for 'Name' and 'Number of columns', and a 'Go' button.

Figura 34: Captura de pantalla phpmyadmin



Para cambios puntuales y en caliente se ha utilizado el cliente FTP filezilla.

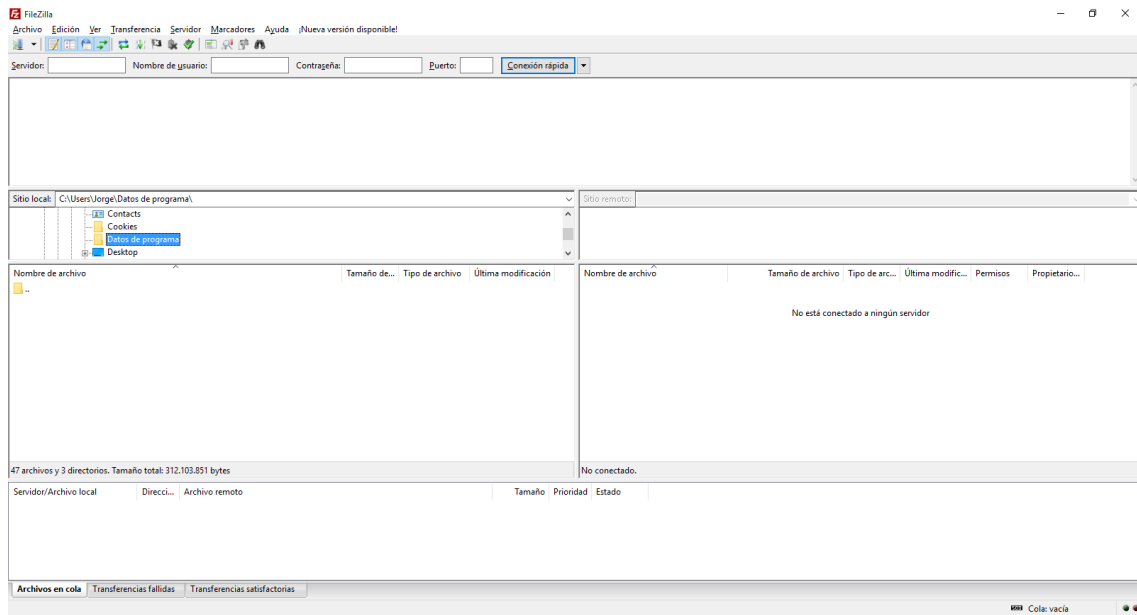


Figura 35: Captura de pantalla cliente FTP Filezilla

Para accesos al servidor y configuración del mismo se ha utilizado una conexión SSH, para establecer esta conexión se ha utilizado el software MobaXTerm.

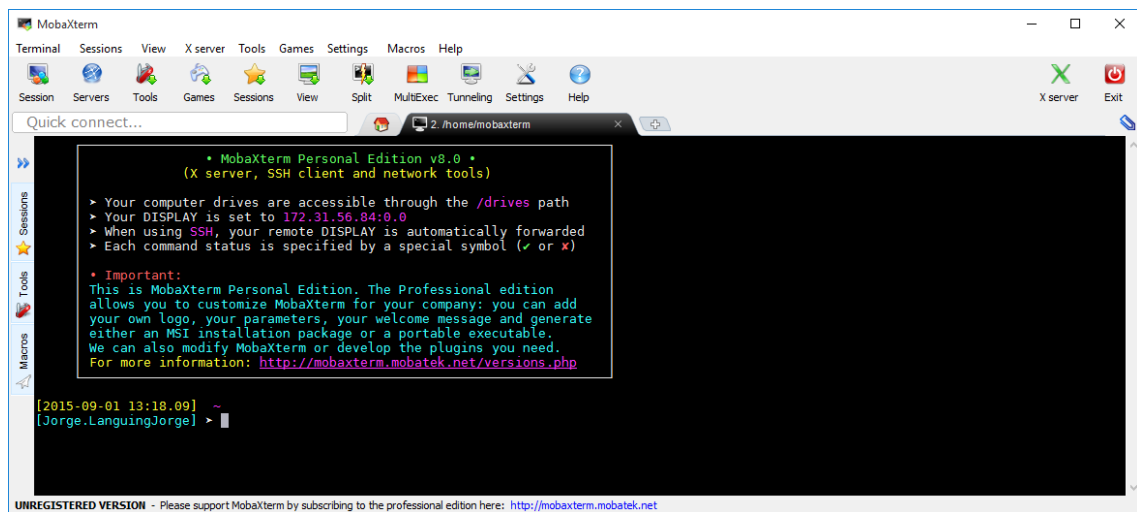


Figura 36: Captura de pantalla MobaXTerm



## 6. Pruebas

En un primer momento se establecieron los requisitos funcionales de la plataforma, en esta sección vamos a probar que todos y cada uno de ellos se cumplen, en cuanto a los requisitos no funcionales, también serán analizados, así como pruebas de conformidad de cumplimiento de estándares.

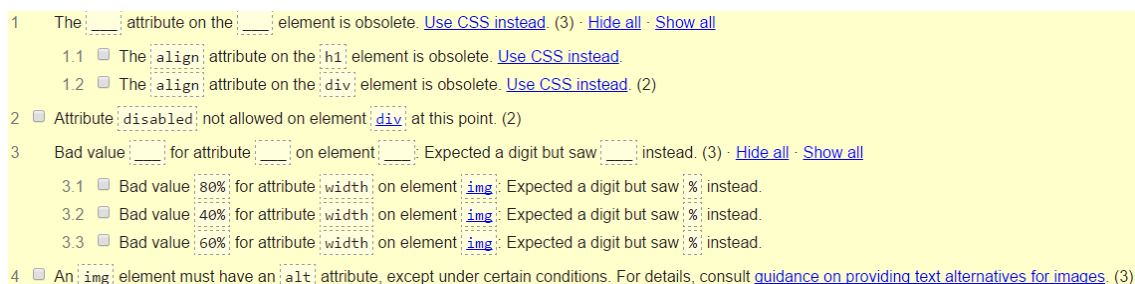
Los casos de uso que se van a probar corresponden a los principales de la aplicación, que son los siguientes:

- Autenticación del usuario
- Registro de usuario
- Eliminar cuenta
- Añadir imagen
- Búsqueda de perfil
- Añadir amistad
- Enviar mensaje
- Crear lugar
- Ver lugar

### 6.1. Conformidad con el estándar w3c

En primer lugar se ha comprobado que la página cumple con el estándar w3c, el incumplimiento del estándar puede desembocar en una incorrecta visualización del sitio en los distintos navegadores del mercado, lo cual no queremos que ocurra, para comprobar que se cumple el estándar existen herramientas web como puede ser la herramienta oficial que está disponible en <https://validator.w3.org> donde podremos poner la URL de nuestra web y ver si cumple con el estándar de un vistazo.

Como podemos observar en la **Figura 30** nuestra web tiene una serie de errores pero el propio validador te dice como corregirlos, una vez localizados, se corrigen como sugiere el estándar.



**Figura 37: Errores detectados en primer lugar por w3c markup validator**

### 6.2. Visualización en navegadores

En esta sección se va a analizar la visualización de las pantallas principales de la web, es decir, la pantalla principal y la del perfil de la mascota en los navegadores más importantes del mercado.

Según <http://netmarketshare.com> la utilización de los navegadores de escritorio en el intervalo de tiempo del último año ha sido la representada en la **Figura 31**. Se puede

observar como en sistemas de escritorio el navegador más empleado es Internet Explorer, seguido por Chrome y Firefox, el navegador Safari y Opera poseen una cuota de mercado bastante baja por lo que no se van a realizar las pruebas sobre esos navegadores.

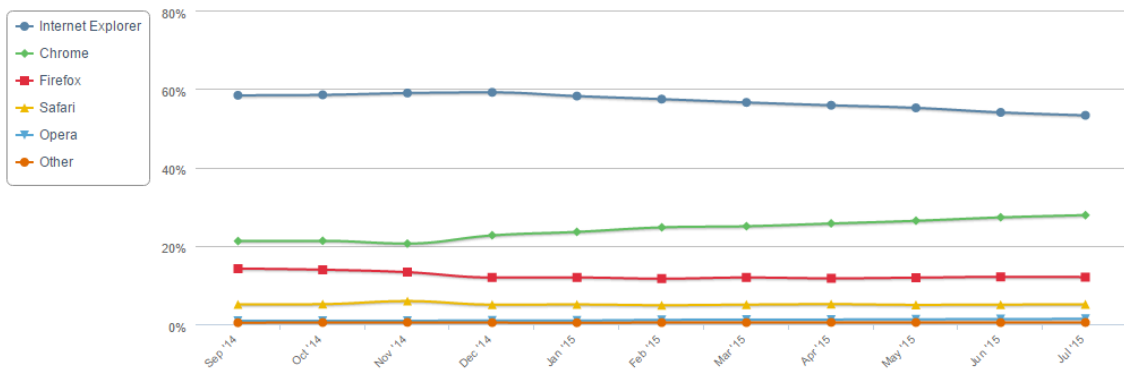


Figura 38: Diagrama navegadores más empleados

### 6.2.1. Visualización en Chrome

La ventana principal se visualiza correctamente en el navegador Chrome como puede observarse en la **Figura 32** y la ventana del perfil mostrada en la **Figura 33** también.

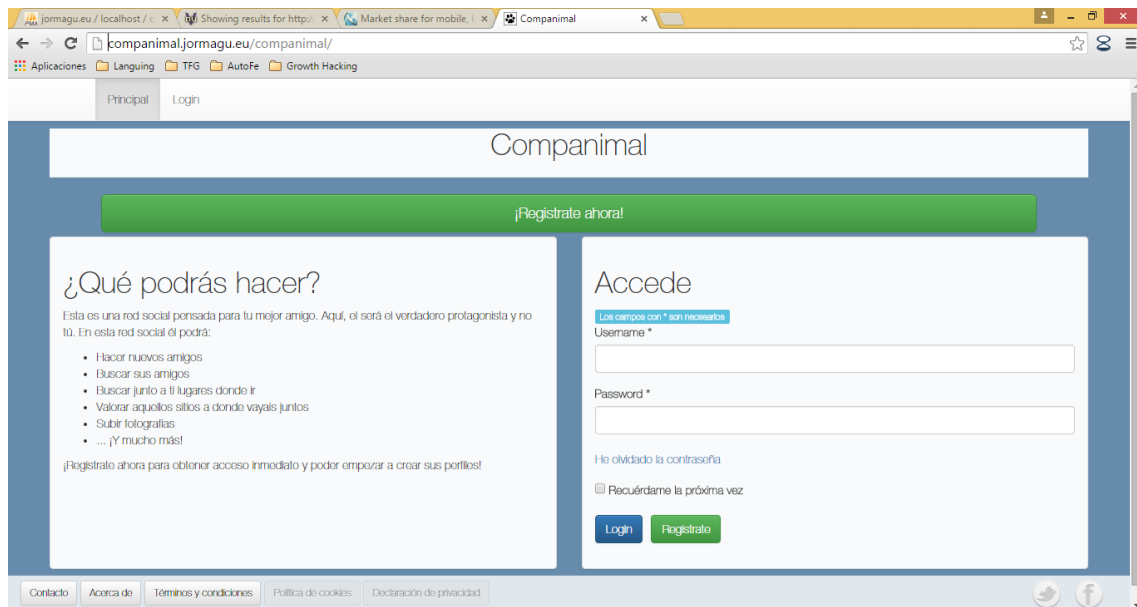


Figura 39: Vista portada en Google Chrome



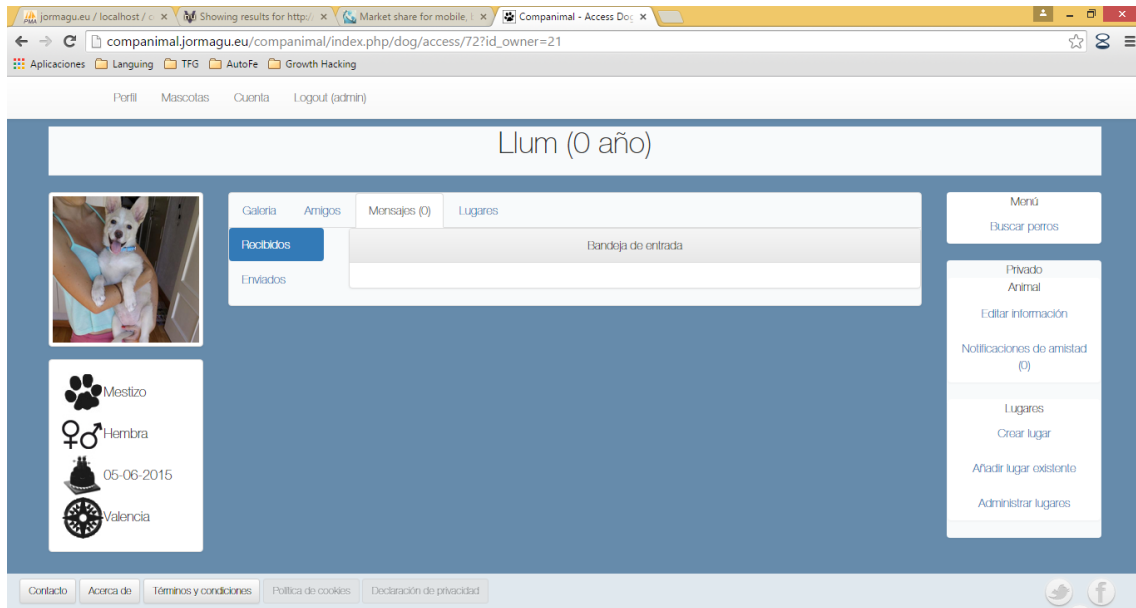


Figura 40: Vista perfil en Google Chrome

### 6.2.2. Visualización en Internet Explorer

Como se puede observar, en Internet Explorer también se visualiza correctamente la web aunque la tipografía es algo diferente a la de Chrome.

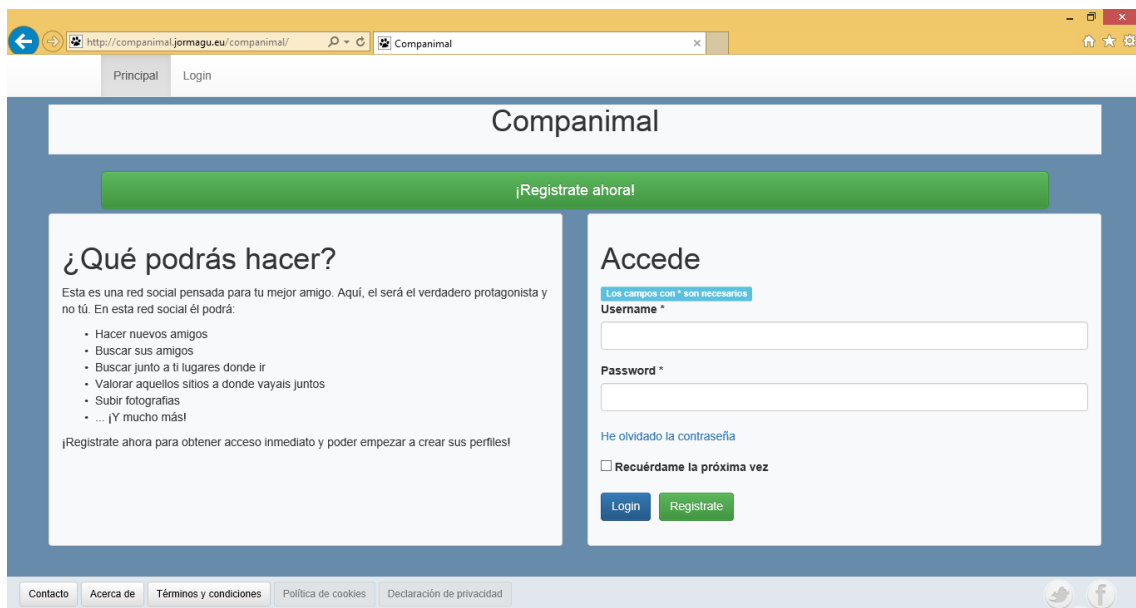


Figura 41: Vista portada en Internet Explorer

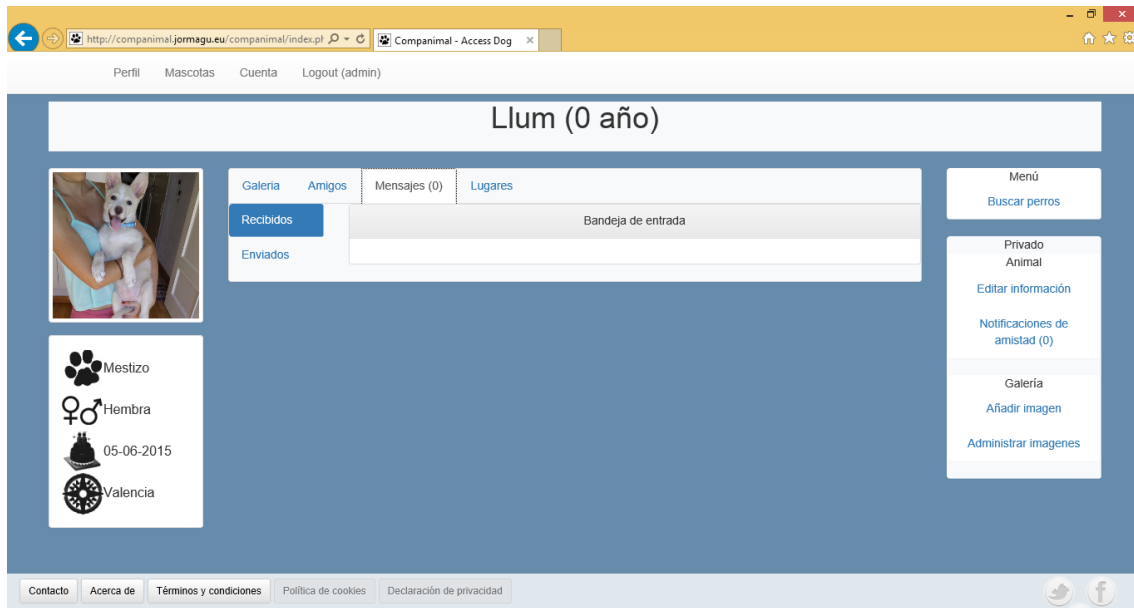


Figura 42: Vista perfil en Internet Explorer

### 6.2.3. Visualización en Firefox

En Firefox podemos observar como también se visualizan correctamente las secciones más importantes de la web.

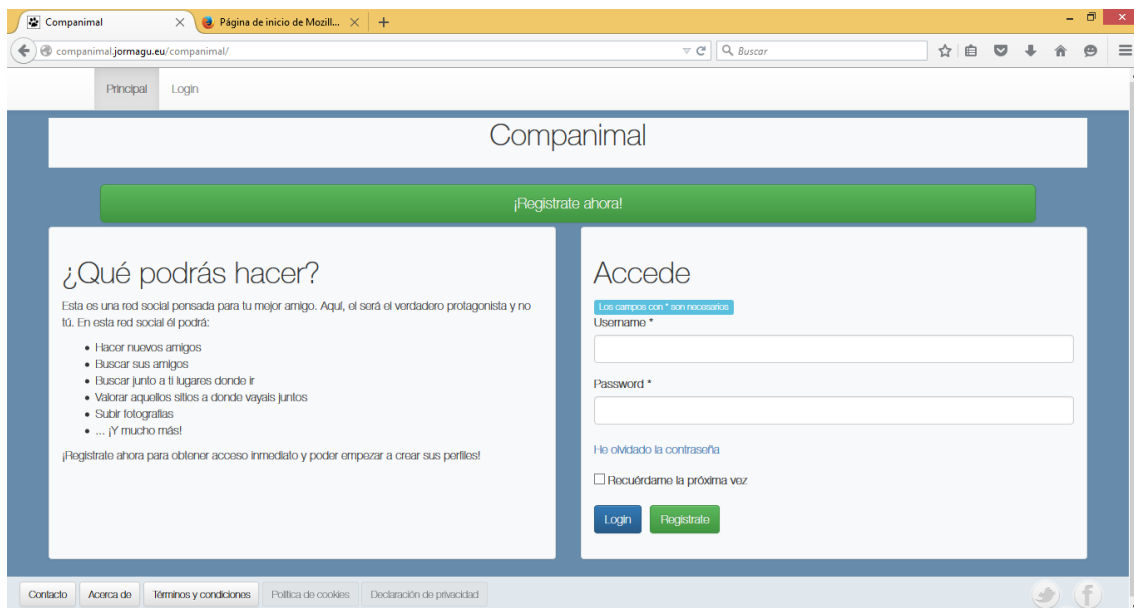


Figura 43: Vista portada en Firefox

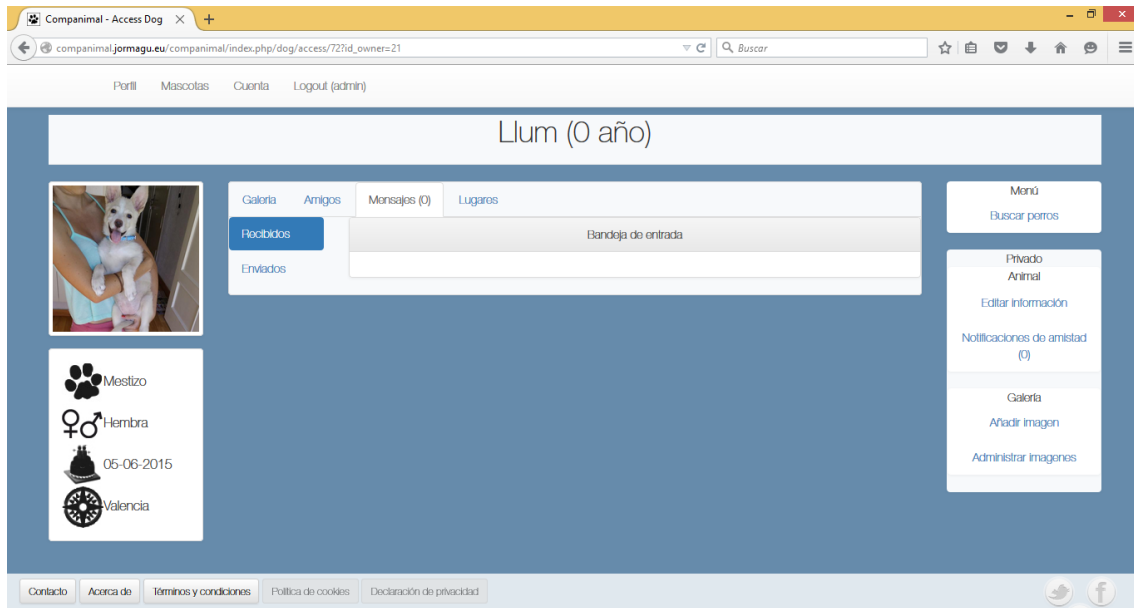


Figura 44: Vista perfil en Firefox

### 6.3. Visualización a distintas resoluciones

A continuación vamos a mostrar la página web a distintas resoluciones, no voy a poder mostrar la resolución 1920x1080 debido a limitaciones técnicas pero sí que voy a poder representar otras resoluciones presentes en el mercado.

En la **Figura 38** se muestra como se visualiza la aplicación en la resolución de pantalla 1366x768

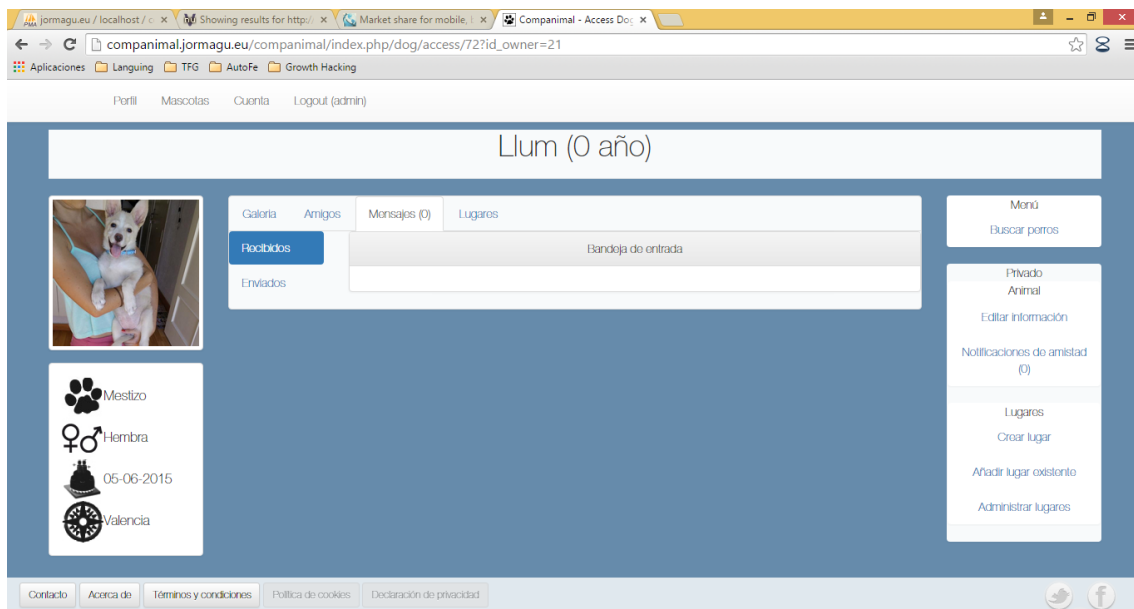
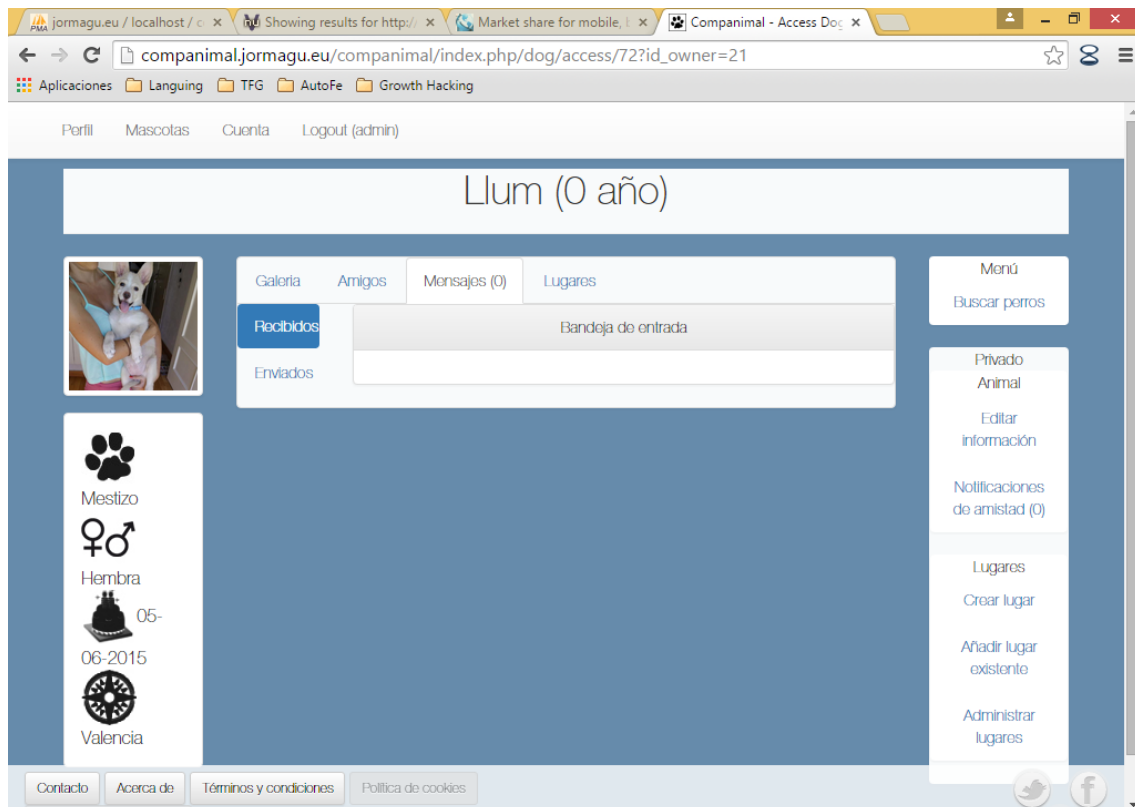


Figura 45: Vista perfil en Firefox

En la **Figura 39** podemos observar la resolución 1024x768.



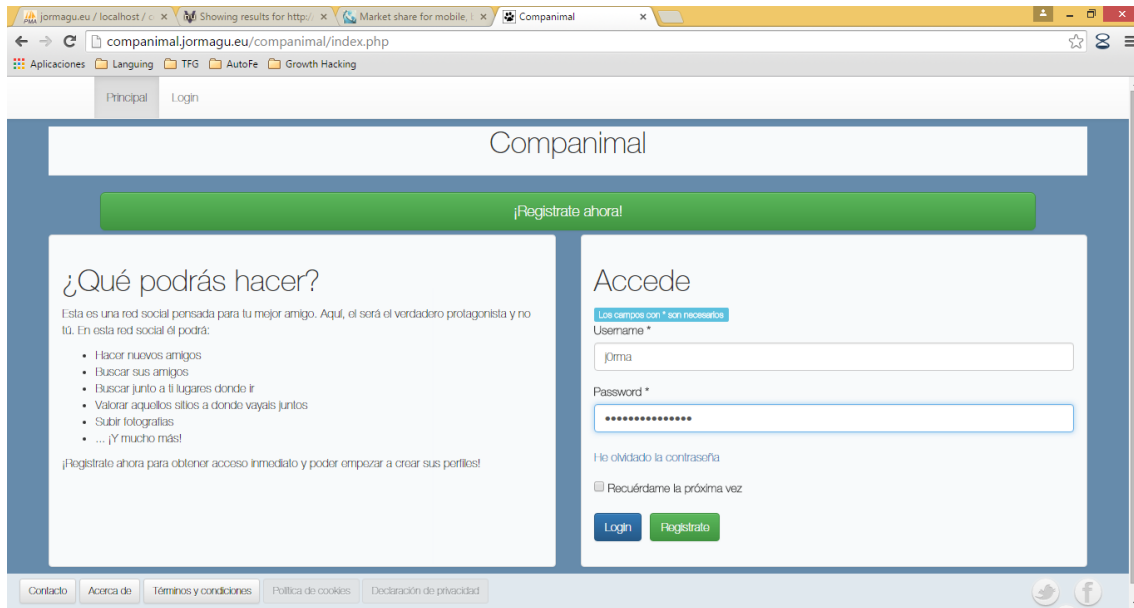
**Figura 46: Vista perfil en Google Chrome (1024x768)**

Como se puede observar la hoja de estilos CSS utilizada adapta la pantalla a la resolución deseada

## 6.4. Requisitos funcionales

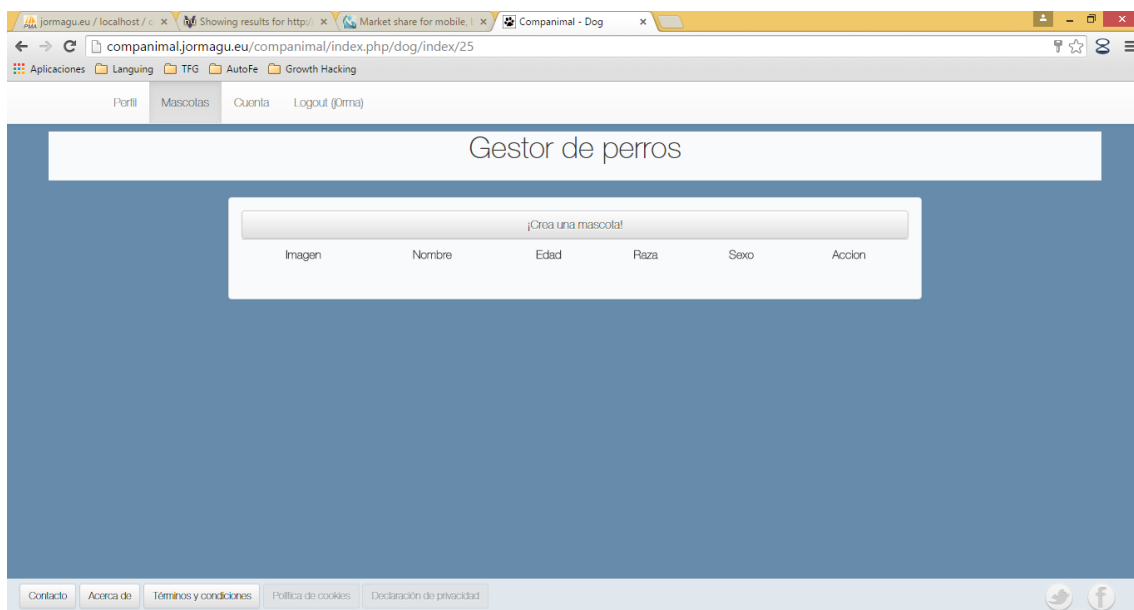
En esta sección se comprobará que se cumplen los requisitos funcionales principales.

En primer lugar la secuencia de las figuras **40** y **41** hacen referencia a la autenticación del usuario.



**Figura 47: Vista caso de uso login (I)**

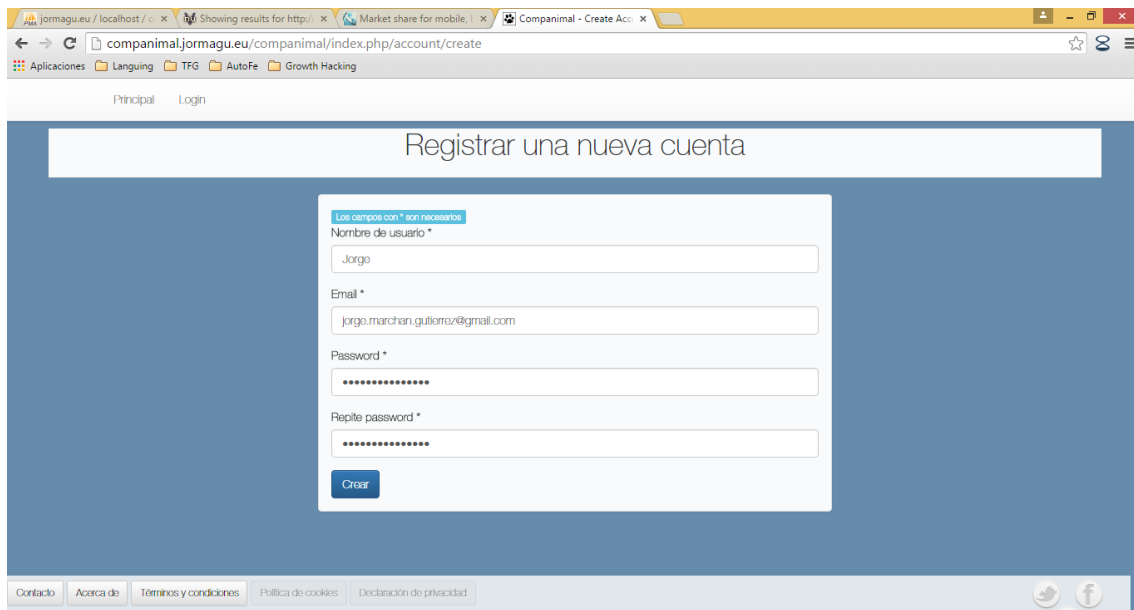
Tras hacer click en *login* se valida al usuario.



**Figura 48: Vista caso de uso login (II)**

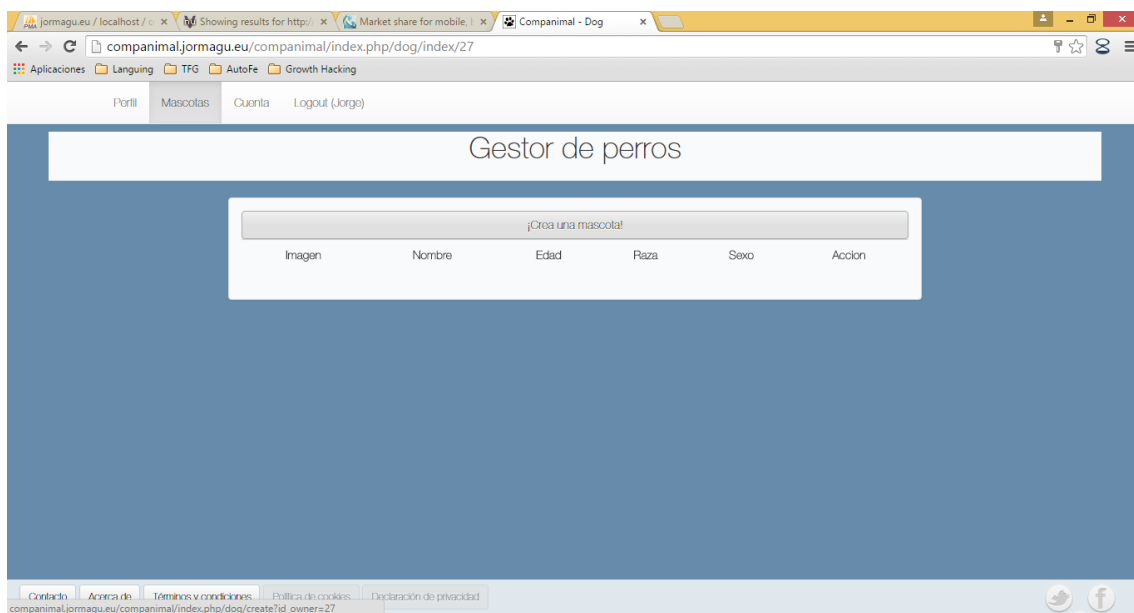
Si la validación es exitosa pasaremos a esta pantalla que es el gestor de los perfiles, lo siguiente es el registro de usuarios, viene representado por las **Figura 42**.





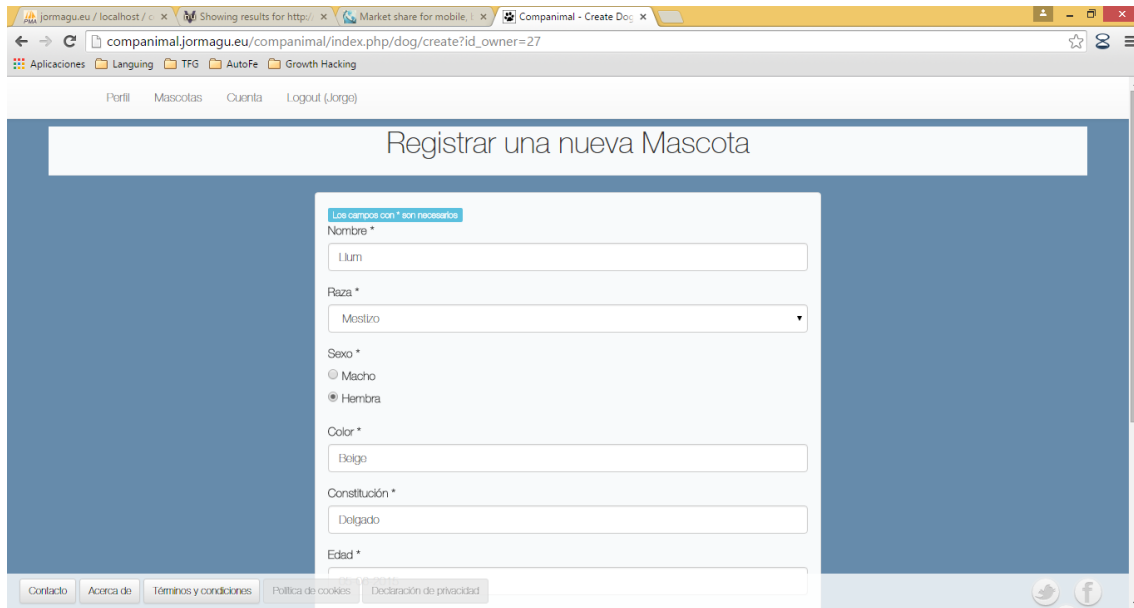
**Figura 49: Vista caso de uso registro cuenta**

A continuación se muestra el proceso de creación de un perfil de mascota, corresponde con las figuras 43, 44, 45 y 46.



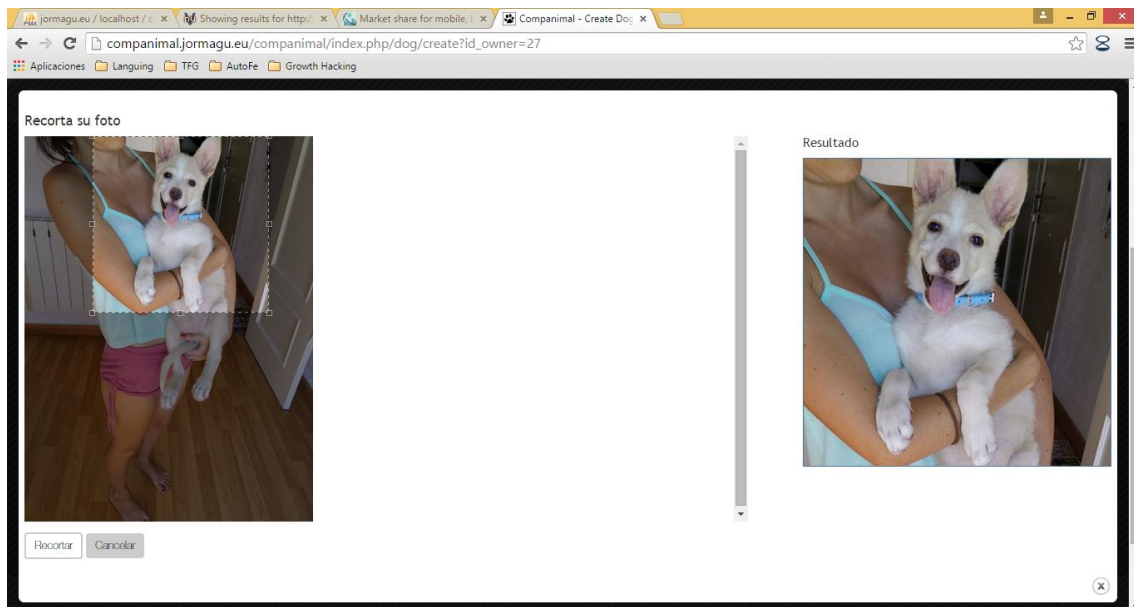
**Figura 50: Vista caso de uso registro mascota (I)**

Al inicio, como estamos creando un perfil desde cero para las pruebas, no existe ningún perfil creado, así que haciendo click en el botón de “Crear mascota” nos vamos a la siguiente vista.

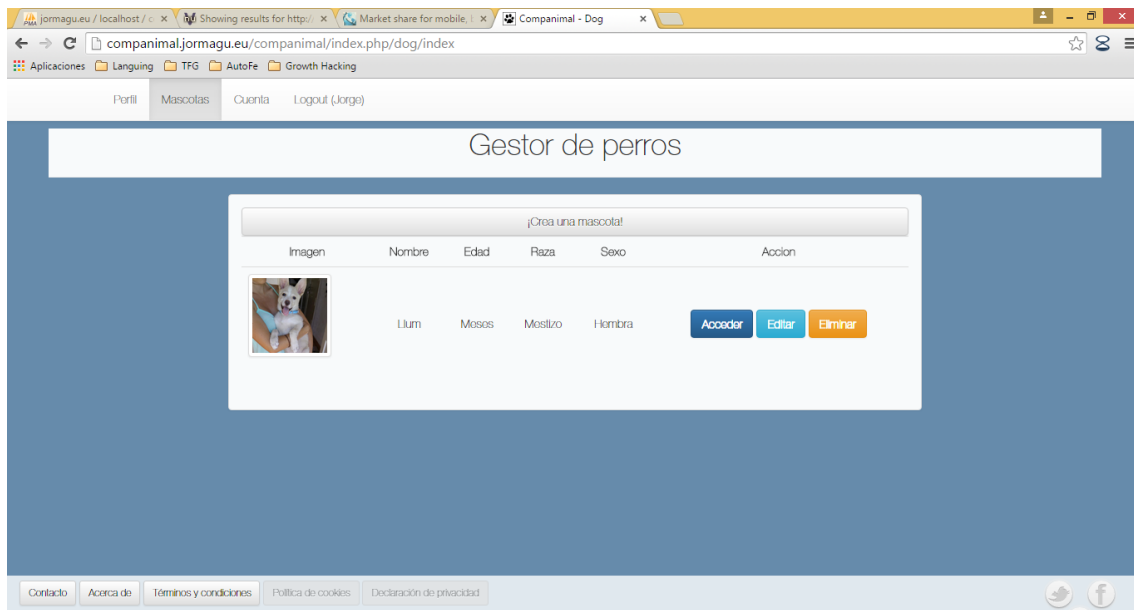


**Figura 51: Vista caso de uso registro mascota (II)**

Este es el formulario que hay que rellenar para crear un nuevo perfil de mascota, a continuación el recorte de la imagen seleccionada como foto de perfil.

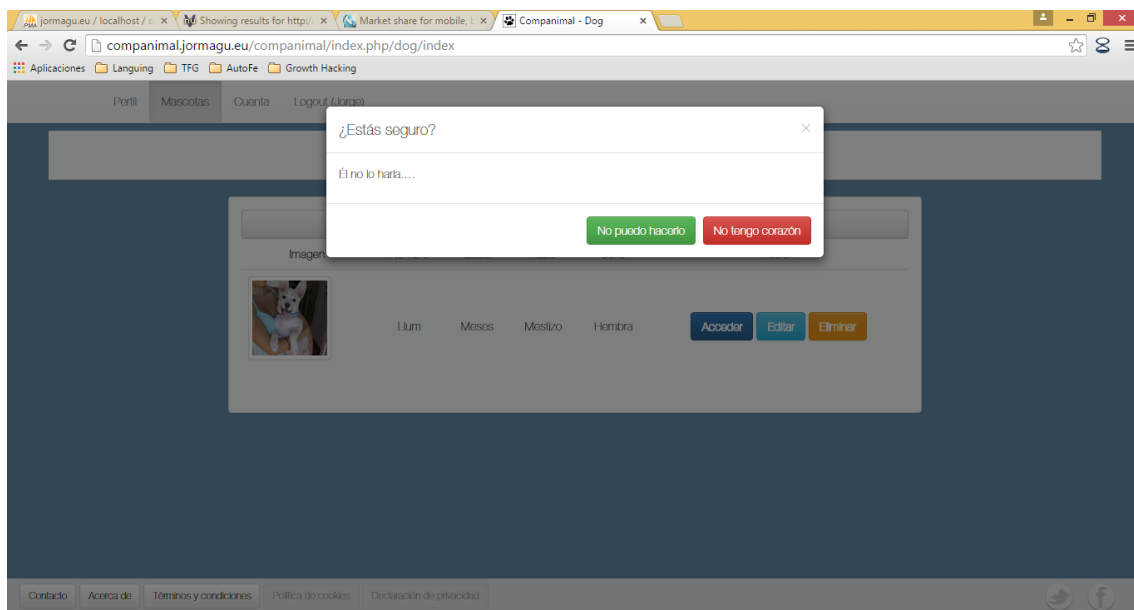


**Figura 52: Vista caso de uso registro mascota (III)**



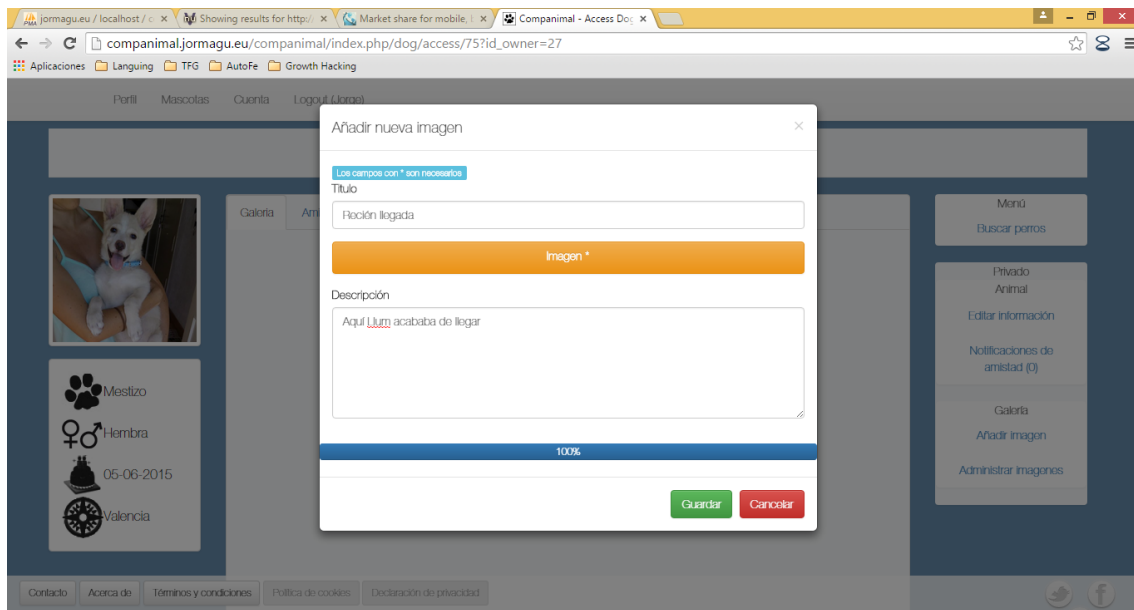
**Figura 53: Vista caso de uso registro mascota (IV)**

Una vez finalizada la creación del perfil vemos como este nos aparece añadido en la ventana del gestor. Para evitar el borrado accidental se incluye un *popup* sobre el botón eliminar como se muestra en la **Figura 47**.

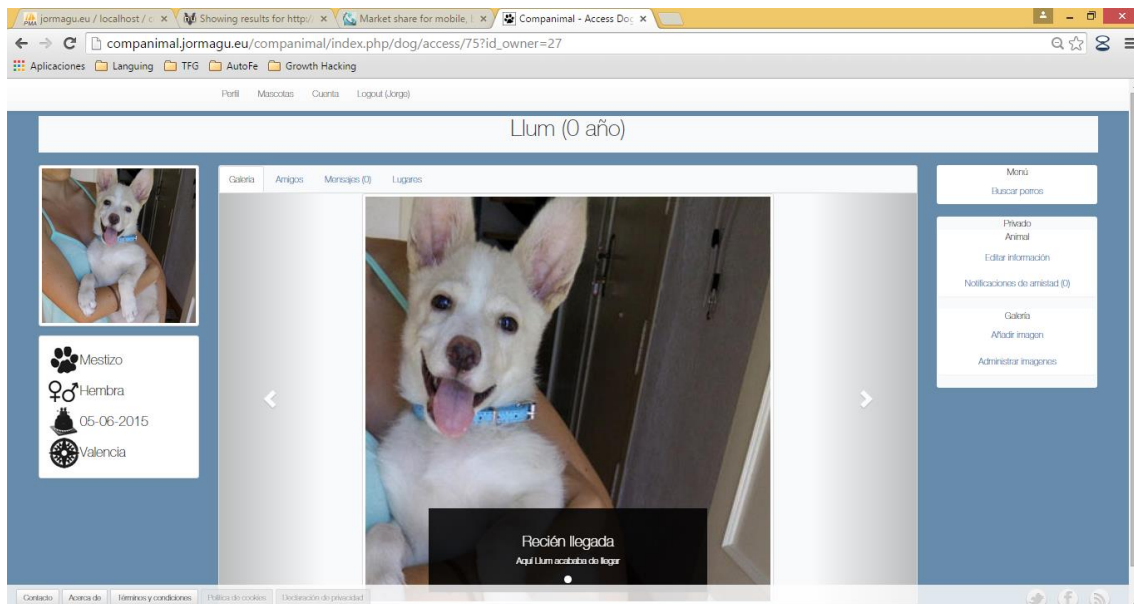


**Figura 54: Vista caso de uso eliminar mascota**

El proceso de añadir fotografías se realiza como puede observarse en las figuras **48** y **49**.



**Figura 55: Vista caso de uso añadir imagen (I)**



**Figura 56: Vista caso de uso añadir imagen (II)**

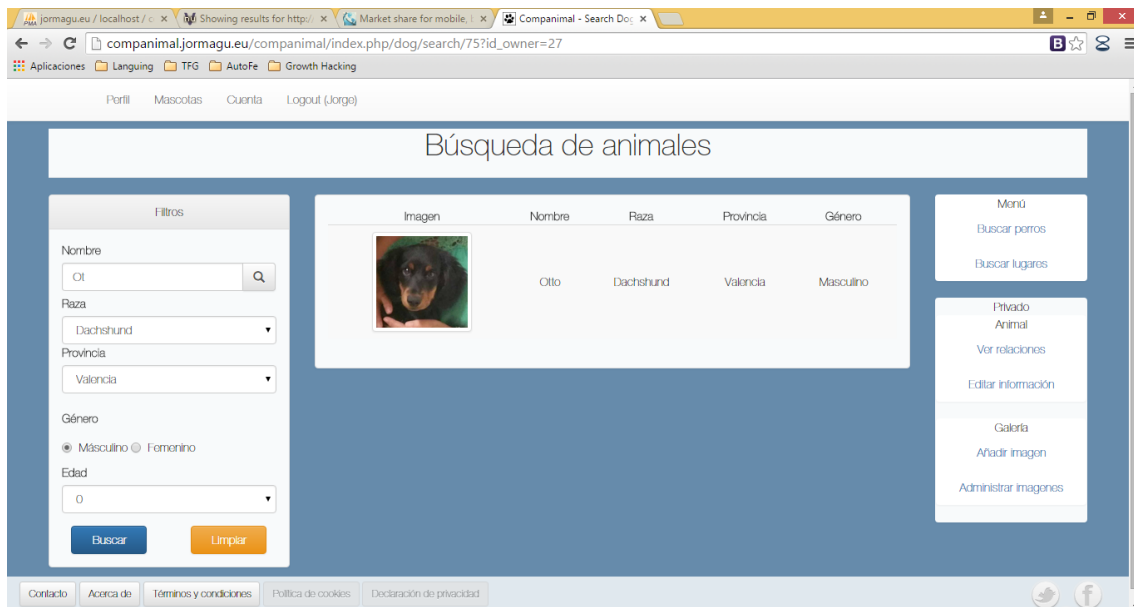
Tras cargar y recortar la imagen vemos como la barra de progreso llega al 100% y posteriormente se visualiza correctamente en el perfil.

A continuación se muestra la búsqueda de otros animales en el sistema.



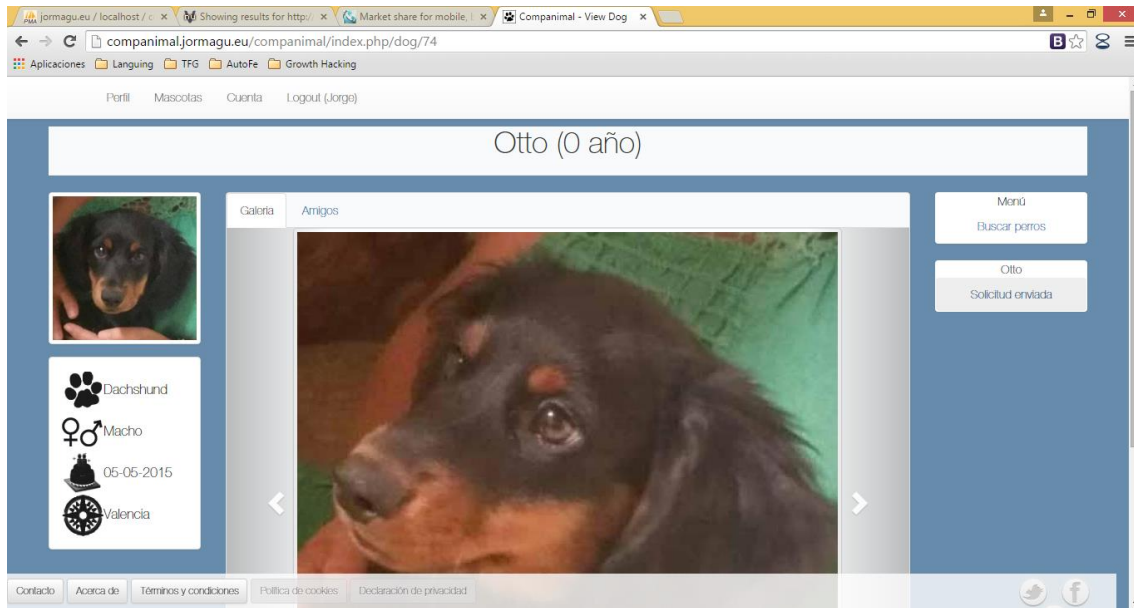
**Figura 57: Vista caso de uso buscar animales (I)**

La **Figura 50** representa la ventana principal de búsqueda de animales, tras rellenar los campos que nos interesan vemos los resultados.



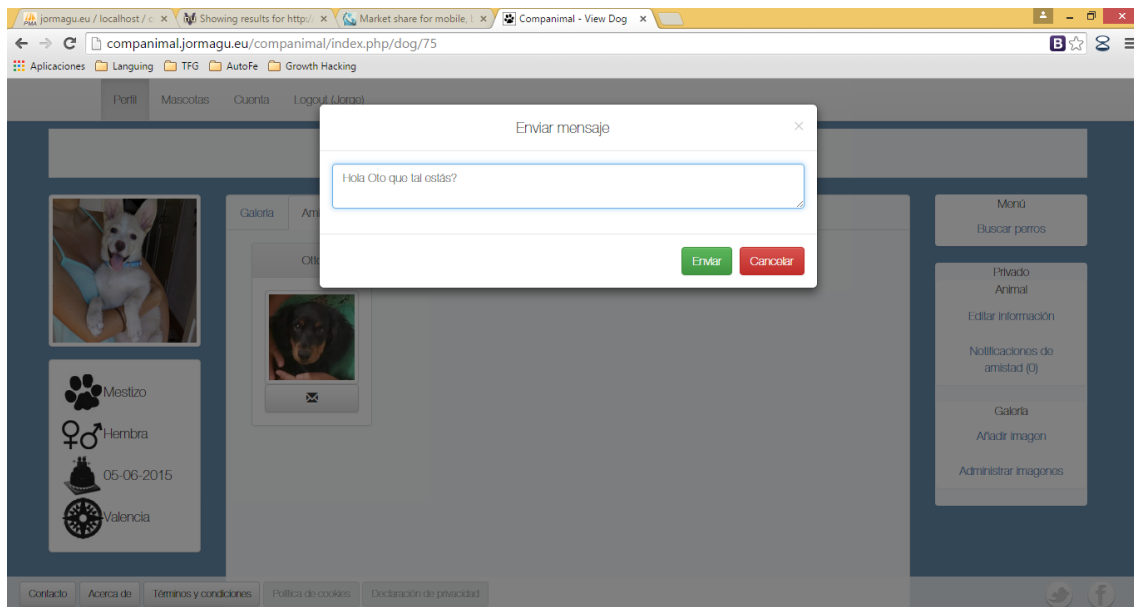
**Figura 58: Vista caso de uso buscar animales (II)**

En la **Figura 51** se observa que la búsqueda funciona, haciendo click en la imagen de perfil accedemos al perfil y podemos añadirlo como amigo.

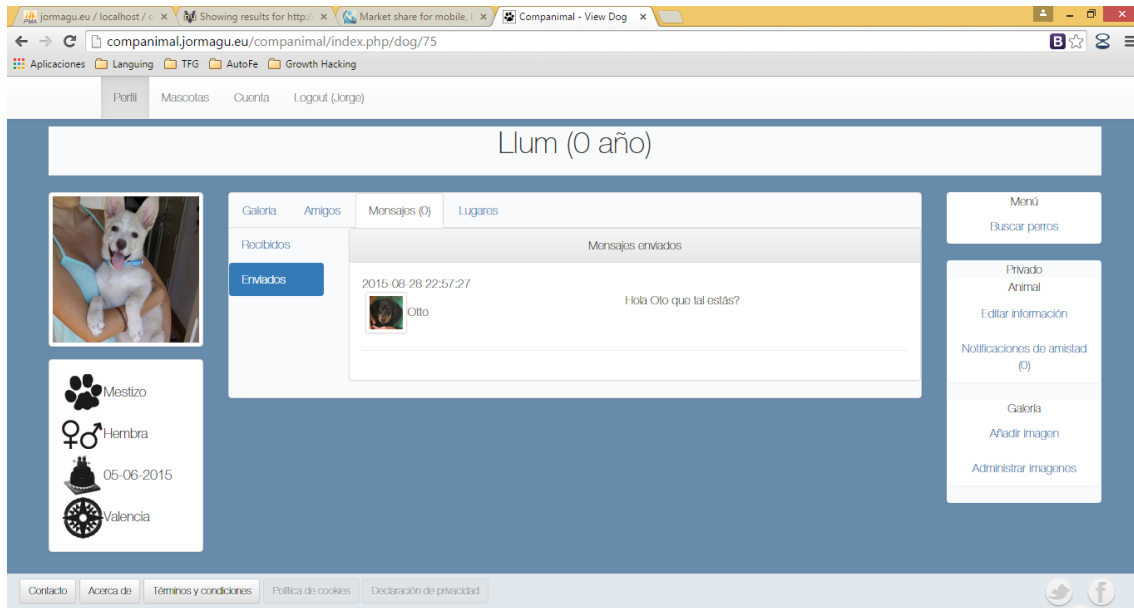


**Figura 59: Vista caso de uso añadir amistad**

Tras aceptar nuestra petición de amistad observamos como aparece en la lista de amigos y podemos enviarle un mensaje.

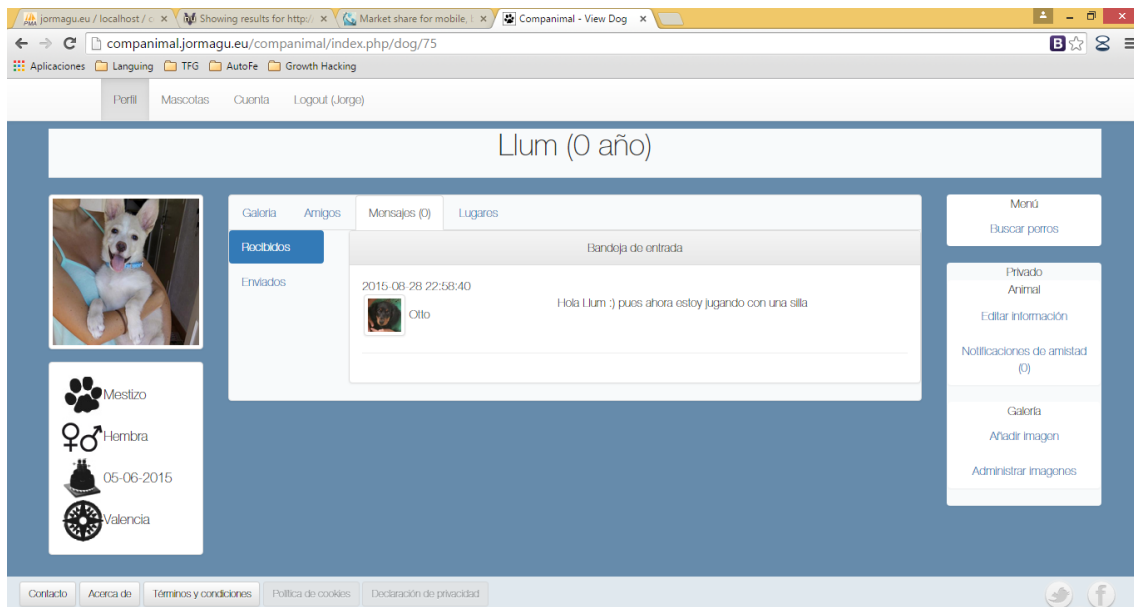


**Figura 60: Vista caso de uso enviar mensaje (I)**



**Figura 61: Vista caso de uso enviar mensaje (II)**

Cuando nos contesta el otro usuario nos aparece como mensaje sin leer y al leerlo desaparece.



**Figura 62: Vista caso de uso enviar mensaje (III)**

A continuación la vista de la creación de lugares donde podremos dar de alta lugares en el sistema y marcarlos sobre un mapa.

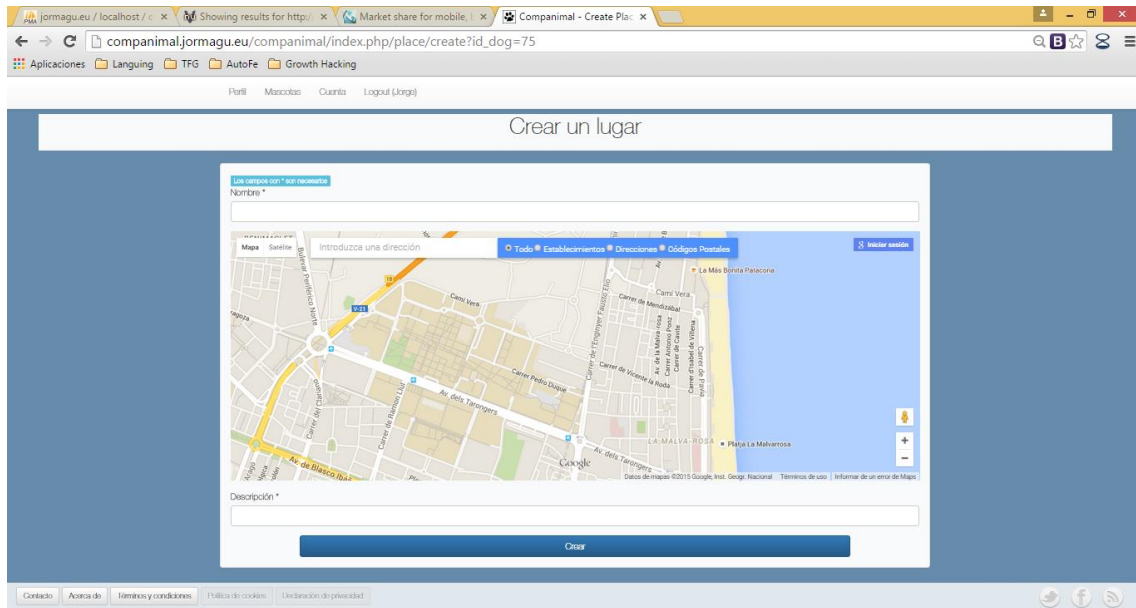


Figura 63: Vista caso de uso crear lugar (I)

Al rellenar el formulario pasaríamos a la vista detallada del lugar como se muestra en la siguiente figura.



Figura 64: Vista caso de uso ver lugar (I)

En cuanto a la seguridad de las contraseñas de los usuarios, se han cifrado todas mediante SHA-256, uno de los algoritmos de encriptación más potentes conocidos hasta la fecha y más costoso de descifrar, por lo que las contraseñas de los usuarios son complicadas de obtener en caso de ataque.



## 7. Conclusiones

Para finalizar la memoria de mi proyecto, las conclusiones extraídas de la realización del mismo, divididas en dos partes, en primer lugar conclusiones técnicas y en segundo lugar conclusiones personales, para continuar con posibles ampliaciones del proyecto.

### 7.1. Técnicas

En este proyecto he aprendido a utilizar un *framework* de PHP, cosa que antes de empezar a realizarlo ni siquiera sabía que existían. Hoy en día, el mercado laboral precisa de programadores de PHP con conocimientos de algún *framework*, es impensable un proyecto de cierta envergadura sin emplear uno de ellos ya que resuelven muchos problemas referentes a calidad del producto y seguridad.

También me ha servido como inmersión en el mundo de los servidores, de los dominios web, subdominios, seguridad de servidores, firewalls, conexiones SSH y demás debido a que aunque a lo largo de la carrera en ciertas asignaturas se nos ha explicado el funcionamiento y se ha puesto en práctica, actualmente me he visto en un entorno real, en el que a lo largo de su desarrollo mi servidor a estado expuesto a ataques reales que han causado su continua caída y he tenido que solucionar esos problemas de seguridad, lo cual me ha parecido apasionante.

También destacar que he aprendido a utilizar herramientas que antes desconocía como PHPMyAdmin que es una herramienta muy sencilla que facilita mucho la tarea de gestionar una base de datos relacional en MySQL. El proceso de instalación y puesta a punto del servidor también me ha servido para romper mano con servidores web ya que he tenido problema con la autogeneración de código por parte de Yii y los permisos de UNIX sobre el fichero afectado.

### 7.2. Personales

En lo personal, este proyecto me ha dado las bases para poder afrontar otros proyectos más grandes y complejos en el futuro. He completado una parte de mi perfil profesional de una forma completamente autodidacta y que era de mucho interés para mí, como es el desarrollo web y mediante uno de los sistemas más buscados en el mundo laboral, el uso de *frameworks* para desarrollo de aplicaciones web.

En lo personal, también he observado cómo realizar una web como la de mi proyecto es un proceso arduo, complicado y que en ocasiones puede parecer no tener fin y en realidad, no tiene fin, porque la forma de trabajo empleada hace que siempre haya cosas nuevas que hacer y expandir la web, implantarle nueva funcionalidad o mejorar aquella que ya tenemos.



### 7.3. Posibles ampliaciones del proyecto

En el momento de la presentación el proyecto no se encuentra traducido en su totalidad, sin embargo la página principal sí que está traducida al inglés, si se accede desde un navegador cuyo idioma predeterminado es el inglés encontraremos la página traducida. Una posible ampliación sería traducir el resto de la página.

Por otra parte, se quedó en el tintero la incorporación de un chat basado en web sockets. Con más tiempo y pudiéndole dedicar más tiempo al día podría haberse implementado.

El diseño de la aplicación se puede mejorar, sin embargo dado el interés académico del proyecto no he considerado relevante centrarme en ese aspecto de la web y he preferido centrarme en aspectos más técnicos, no descarto en un futuro rediseñar la web para hacerla más atractiva para el usuario.

En el caso de haber muchos usuarios sería necesario un panel de administración para controlar los lugares que se van creando y la información que van introduciendo en el sistema, por lo que habría que gestionar los roles de los usuarios en la plataforma y diseñar esta interfaz de administración.

En el caso de pasar a un entorno de producción, sería necesario empezar a medir las cosas integrando herramientas como Google Analytics o Mixpanel según lo que necesitésemos en cada caso, el uso de estas herramientas queda fuera del alcance del proyecto sin embargo, son herramientas muy útiles e interesantes.

Otra posibilidad sería la de añadir comentarios sobre las fotografías de los diferentes usuarios de la web, que no se ha añadido al proyecto debido a que tenemos un escenario similar añadiendo opiniones sobre los lugares que se han visitado y era un escenario redundante, se ha considerado de mayor interés desarrollar otro tipo de funcionalidad, como la mensajería interna.

Se trata de un proyecto muy escalable y modular, en cualquier momento podría surgir una idea de ampliación y podría ser llevada a cabo sin demasiada dependencia debido a la modularidad y escalabilidad que te ofrece un *framework* de desarrollo como el utilizado.

## 8. Bibliografía

### 8.1. Libros

Winesset, Jeffrey (2da. Ed.) (Noviembre 2012). *Web Application Development with Yii and PHP*. Birmingham: Packt Publishing Ltd.

Makarov, Alexander (1ra. Ed) (Agosto 2011). *Yii 1.1 Application Development Cookbook*. Birmingham: Packt Publishing Ltd.

### 8.2. Documentación online

Varios autores. (2008-2015). La guía definitiva para Yii 1.1, consultado de enero a septiembre del 2015, en <http://www.yiiframework.com/doc/guide/1.1/es/index>

Varios autores. (2008-2015). *Class Reference*, consultado de enero a septiembre del 2015, en <http://www.yiiframework.com/doc/api/>

Varios autores. (Agosto 2015). Manual de PHP, consultado de enero a septiembre del 2015, en <http://php.net/manual/es>

Varios autores. (Septiembre 2015). Portal Web Stackoverflow, consultado durante todo el desarrollo del proyecto, en <http://stackoverflow.com/>

Varios autores. (Septiembre 2015). Manual de referencia MySQL 5.7, consultado durante todo el desarrollo del proyecto, en <http://dev.mysql.com/doc/refman/5.7/en/>

Varios autores. (Septiembre 2015). Documentación de jQuery, consultado durante todo el desarrollo del proyecto, en <http://api.jquery.com/>

Varios autores. (Septiembre 2015). Documentación de Twitter Bootstrap, consultado durante todo el desarrollo del proyecto, en <http://getbootstrap.com/>