



Universidad
Rey Juan Carlos

ESCUELA TÉCNICA SUPERIOR
DE INGENIERÍA DE TELECOMUNICACIÓN

INGENIERÍA DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

DISEÑO E IMPLEMENTACIÓN DE UNA
APLICACIÓN WEB PARA LA GESTIÓN Y
DESARROLLO DE ASOCIACIONES DE
ADOPCIÓN DE ANIMALES

Autor: Rocío Quiñones Grande
Tutor: Gregorio Robles Martínez

CURSO ACADÉMICO 2013/2014

Proyecto Fin de Carrera
DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y
DESARROLLO DE ASOCIACIONES DE ADOPCIÓN DE ANIMALES

Autor
ROCÍO QUIÑONES GRANDE

Tutor
GREGORIO ROBLES MARTÍNEZ

La defensa del presente Proyecto Fin de Carrera se realizó el día de
de 2014, siendo evaluada por el siguiente tribunal:

PRESIDENTE:

VOCAL:

SECRETARIO:

y habiendo obtenido la siguiente CALIFICACIÓN:

FUENLABRADA, A DE DE 2014

Copyright ©2014 Rocío Quiñones Grande

Este documento se publica bajo la licencia

Creative Commons Reconocimiento-CompartirIgual 3.0 España

<http://creativecommons.org/licenses/by-sa/3.0/es>

(Ver Apéndice C.)

*Dedicado a
mi familia*

Agradecimientos

Este proyecto supone la culminación de mi etapa universitaria y debido a ello son muchas las personas a las que me gustaría dar las gracias por su apoyo en todo este tiempo.

En el ámbito académico son varios los profesores a los que tendría que agradecer su formación y entrega. Gracias a todos ellos y más concretamente a mi tutor de proyecto Gregorio.

Agradecer a mi familia su paciencia y dedicación. Sin ellos no habría sido posible.

Este largo proceso me ha permitido conocer a parte de las personas más importantes de mi vida, sin duda es lo que me llevo de tantos años de duro trabajo. Con David y Cristina viví de la mano los primeros años de Universidad y quizás los más bonitos de esta etapa. David ese instante de *A single man* es nuestra esencia, el fiel retrato de nuestra leal amistad. Cristina con la que compartí cada segundo de esta travesía de la forma más dulce que conozco. Os quiero. Los últimos años me permitieron conocer a mis chungas a las que adoro. Cristina cuya transparencia me calo desde el instante cero y de la que decidí no me alejaría jamás. A su lado he vivido etapas difíciles que solo han reforzado una amistad eterna. Gracias a ella conocí a Aurora, toda una sorpresa para mi. Ella es la palabra GENEROSIDAD, así, en mayúsculas. Por que es delgadita, nuestro pajarito, pero con un alma gigante. Las dos han sido mi pilar fundamental estos años y lo serán el resto de nuestras vidas. He conocido a Beatriz de la que no solo admiro su constancia y su capacidad de entrega sino que además me llevo una de las personas más buenas que he conocido. Te adoro mi pequeño panda. También tengo que agradecer estos años, porque gracias a esta etapa he conocido a una persona generosa, trabajadora y dedicada a sus amigos, Cintia. Gracias infinitas por todo, te quiero.

En este camino son muchas las personas que me han acompañado desde siempre. Mari Luz, las aventuras de piratas, los nervios y un primer amor resumen más de veinte años a tu lado. Mi princesa Inma con la que es difícil no sentir la rotación de la tierra, una de las mujeres más especiales y gigantes que he conocido. Te admiro. Y mi mejor amigo Alberto, quizás la persona más importante en mi realización en esta etapa, puesto que confío en mi cuando nadie lo hacía. Junto a él lloré cuando creía que no podría más con todo esto y fue quién mantuvo charlas eternas a mi lado hasta que logró darme la seguridad que a veces tanto necesitaba. A pesar de que los lazos que nos unían cambiaran, en ningún momento dejo de preocuparse por mí y de regalarme sus consejos y riñas.

Sin vosotros nada habría sido posible. Os adoro.

Contenido

Índice de figuras	XVI
Índice de tablas	XVII
Abstract	XIX
Resumen	XXI
Abreviaciones	XXIII
1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Objetivos	2
1.4. Estructura de la memoria.	3
2. Estado del arte	5
2.1. Introducción	5
2.2. Aplicación Cliente-Servidor	5
2.3. Aplicación web	6
2.3.1. Arquitecturas de las Aplicaciones web	8
2.3.2. Ventajas e inconvenientes de las aplicaciones web	9
2.3.3. Historia de las aplicaciones web	10
2.3.4. Futuro de la aplicación web	11
3. Tecnologías utilizadas	13
3.1. HTML5	13
3.1.1. Historia de HTML5	13
3.1.2. Características de HTML5	14
3.1.2.1. Diferencias con HTML4	15
3.1.3. Ventajas e inconvenientes de HTML5	15
3.2. CSS3	16
3.2.1. Ventajas es inconvenientes de CSS3	17
3.2.2. ¿Por qué HTML5 y CSS3?	18
3.2.3. Sitios que utilizan HTML5 y CSS3	18
3.3. Node.js	18

3.3.1.	Comparativa Node.js con Apache	19
3.3.2.	Uso de Node	20
3.4.	MySQL	22
3.4.1.	Características de MySQL	22
3.5.	XAMPP	22
3.5.1.	Panel de control de XAMPP	23
4.	Diseño de la aplicación	25
4.1.	Diseño del Servidor	26
4.1.1.	Creación del modelo de datos.	26
4.1.1.1.	Tablas de datos	30
4.1.1.2.	Problemas con XAMPP y la base de datos	31
4.1.2.	Servicio Web	32
4.1.3.	Características del Servidor	33
4.1.3.1.	Uploads	34
4.1.3.2.	Vídeo	36
4.2.	Diseño del Cliente	38
4.2.1.	Características del Cliente	38
4.2.2.	Desarrollo e implementación características	45
4.2.2.1.	Crear y leer códigos QR	45
4.2.2.2.	Crear Fotos con Canvas	45
4.2.2.3.	Realización de vídeos	46
4.2.2.4.	Geolocalización	47
5.	Resultados	49
5.1.	Origen de la Aplicación	49
5.2.	Pruebas realizadas	50
5.2.1.	Prueba de foto	50
5.2.2.	Prueba de geolocalización	50
5.2.3.	Prueba de vídeo	50
5.2.4.	Creación y lectura de códigos QR	51
5.2.5.	Subir archivos	51
5.3.	Opiniones finales	51
6.	Conclusiones	55
6.1.	Análisis del trabajo realizado	55
6.2.	Aprendizaje	56
6.3.	Líneas futuras	57
6.4.	Valoración final	58
	Apéndices	59
	A. Publicación del código	61

B. Planificación del Proyecto	63
B.1. Elección del proyecto	63
B.2. Estudio de los objetivos	63
B.3. Selección de las herramientas, así como el estudio de las mismas	63
B.4. Desarrollo de la aplicación	64
B.5. Desarrollo de la memoria	64
C. Licencia Creative Commons	65
C.1. Definiciones	65
C.2. Límites de los derechos	68
C.3. Concesión de licencia	68
C.4. Restricciones	69
C.5. Exoneración de responsabilidad	70
C.6. Limitación de responsabilidad	71
C.7. Finalización de la licencia	71
C.8. Miscelánea	71
C.9. Aviso de Creative Commons	72
Glosario	I
Bibliografía	VII

Índice de figuras

2.1. Separación Cliente Servidor [1].	6
2.2. Arquitectura Web, todo en el servidor [1].	8
2.3. Arquitectura Web con la base de datos a parte [1].	8
2.4. Arquitectura Web con todo en el servidor incluido servidor aplicaciones [1].	9
3.1. Soporte para la API de geolocalización de los distintos navegadores [2].	14
3.2. Ejemplo visual CSS sprites [3].	17
3.3. Vimeo [4].	18
3.4. Gráfico conexiones Apache y Node.js [5].	19
3.5. Gráfico de empresas que trabajan con Node.js [5].	21
3.6. Vision de PHPMyadmin[6].	23
3.7. Ventana control XAMPP[6].	24
4.1. Esquema de la Arquitectura	25
4.2. Esquema de Funcionalidad	26
4.3. Servidores web [20]	27
4.4. Panel de control XAMPP.	28
4.5. Panel de control sin iniciar módulos.	28
4.6. Panel de control iniciado.	29
4.7. Página de configuración XAMPP.	29
4.8. Esquema Base de datos.	31
4.9. Menu.	38
4.10. Perfil.	38
4.11. Figura de los donativos y la función adopciones	39
4.12. Creación de foto y vídeo	40
4.13. Galeria vídeos.	40
4.14. Requisitos adopción.	41
4.15. Información socio.	41
4.16. Información simpatizante.	42
4.17. Información Voluntario.	42
4.18. Información Voluntario 1.	43
4.19. Información Voluntario 2.	43
4.20. Información Voluntario 3.	44
4.21. Al rescate.	44
4.22. Figura de la imagen	45

4.23. Mostrar ruta.	47
5.1. Tabla usuarios.	52
5.2. Gráfico circular y de barras	52

Índice de tablas

3.1. Diferencias entre Node.js y Apache	20
4.1. Métodos HTTP	33
5.1. Opiniones	53

RESUMEN

El desarrollo de las tecnologías, y más concretamente de las aplicaciones web, han favorecido el avance del ocio y entretenimiento en nuestra sociedad a través de redes sociales o juegos. Sin embargo es posible aprovechar el potencial que aportan los terminales móviles, dándoles un enfoque más altruista al uso de estas aplicaciones, mediante el acceso a la información, su uso y la implicación en la sociedad.

En esta línea nace este proyecto fin de carrera con el fin de ofrecer una aplicación web con un enfoque más solidario y que pudiese servir de ayuda en la sociedad. Conforme a esta idea se produce el diseño, implementación y prueba de una aplicación que favorece a la adopción de animales así como la gestión de asociaciones relacionadas con este tema, suponiendo una mejora de páginas web de este tipo, como por ejemplo la de la asociación *El Refugio*.

Debido a ello el planteamiento inicial surgió como una conversión de las páginas de este tipo de asociaciones, las cuales, resultaban bastante arcaicas, a una aplicación web con las funcionalidades principales, pero con un diseño mucho más visual y moderno.

Tras el estudio de varios lenguajes de programación a lo largo de la carrera pude analizar y elegir los que ofrecían más funcionalidades, que podrían cubrir ciertas necesidades de dichas asociaciones, y que, a pesar de su complejidad, resultaban bastante interesantes. Las habilidades tecnológicas y de análisis que he ido adquiriendo me facilitaron el desarrollo de nuevas características con las que complementar la aplicación. Entre estas se encontraba la de poder hacer fotos o vídeos desde el navegador. Esto unido a la geolocalización permitía a las asociaciones, por ejemplo, poner al servicio de cualquier usuario la ayuda inmediata de un animal que pudiese estar en peligro.

A lo largo de la presente memoria se irán analizando todas las etapas necesarias hasta llegar a una solución. Consiguiendo una aplicación con un aspecto muy agradable e intuitivo, y que además, mediante las nuevas funcionalidades consigue cubrir parte de las necesidades de estas asociaciones, permitiéndome a nivel personal poder llevar a cabo muchos de los conocimientos adquiridos durante la carrera, así como complementarlos con todo lo que he aprendido al realizar la aplicación.

ABREVIACIONES

API	A pplication P rogramming I nterface – Interfaz de programación de aplicaciones
ARPANET	A dvanced R esearch P rojects textbfAgency N etwork
AWS	A mazon W eb S ervices
BD	B ase de D atos
DNS	D omain N ame S ystem
GPL	G eneral P ublic L icense
HTML	H yper T ext M arkup L anguage
HTTP	H yper T ext T ransfer P rotocol
I/O	I nput O utput
ODBC	O pen D ata B ase C onnectivity
PHP	H ypertext P reprocessor
REST	R epresentational S tate T ransfer
SEO	S earch E ngine O ptimization
TCP/IP	T ransmission C ontrol P rotocol I nternet P rotocol
WHATWG	H yper T ext A pplication T echnology W orking G roup
W3C	W orld W ide W eb C onsortium – Consorcio World Wide Web

CAPÍTULO 1

INTRODUCCIÓN

La finalidad de este proyecto fin de carrera es construir una *Aplicación Web*, la cual nace de una necesidad por parte de las asociaciones. La necesidad de lograr un mecanismo más ágil que permita fomentar la adopción de mascotas. La necesidad de cubrir una serie de objetivos en lo que se refiere a las facilidad del rescate de animales, ganando en sencillez y consiguiendo una cercanía con el usuario que facilita indirectamente la defensa y protección de los animales.

1.1. Contexto

Las aplicaciones web en la actualidad son la tecnología en auge y se han incorporado en nuestras vidas formando parte de ellas de una manera habitual. Disponemos de aplicaciones que favorecen el entrenamiento personal de cuerpo y mente. Cada día recibimos correos electrónicos que forman parte de nuestro día a día, además invertimos tiempo de ocio en juegos y disponemos de aplicaciones que nos permiten realizar compras sin ningún tipo de esfuerzo. Las aplicaciones web nos ponen en contacto con seres queridos situados a cientos de kilómetros a través de redes sociales. Debido a todo ello sería injusto no darle la relevancia, a nivel social, que en muy pocos años han logrado, introduciéndose en nuestra vida y facilitándola enormemente. A pesar de la gran cantidad de utilidades vigentes, el desarrollo de aplicaciones en otros ámbitos relacionados con fines solidarios no han sido encauzadas de forma que permitan dar servicio a las distintas organizaciones humanitarias de una forma más útil.

Sin embargo parece que este enfoque inicial está cambiando, surgiendo aplicaciones cuyo objetivo final está vinculado a una labor social más que de divertimento. Por ejemplo en *Android* existen algunas aplicaciones de este tipo bastante interesantes, como *FoodShareFilter* creada entre *Instagram* y Manos Unidas. Se trata de un *plugin* de la aplicación de *Instagram* con la que, al hacer fotos de un plato de comida y compartirlo mediante *FoodShareFilter*, estarás colaborando con una pequeña donación con la que paliar el hambre en el mundo.

1.2. Motivación

A pesar de que existen herramientas de carácter altruista, éstas suelen estar bastante descompensadas con el resto de aplicaciones dedicadas más al entretenimiento. Además las que existen suelen tener pocas funcionalidades para el potencial que aporta esta tecnología en la actualidad.

Todo ello, unido a que en la actualidad no conozco ninguna aplicación dedicada a la adopción, protección y rescate de los animales motivaron la creación de este proyecto fin de carrera.

Por tanto, esta *Aplicación Web* es el resultado de una idea propia, motivada por el contexto actual y las necesidades de las asociaciones de acercarse a las personas, darse a conocer, evacuar stock, encontrar mascotas desaparecidas, obtener donaciones o proteger animales en peligro. Estas carencias pude conocerlas debido a que, desde hace algún tiempo, mi relación con personas que invierten mucho tiempo en ellas ha sido cada vez más relevante. Este contacto directo me ha facilitado conocer muchas de sus necesidades directas y de qué manera podría ayudar a solventarlas.

Pero no solo eso, a nivel personal mi idea era la de crear un proyecto fin de carrera que satisficiera mi necesidad de crear algo con finalidades solidarias. Una aplicación que me permitiera enfocar las telecomunicaciones con una necesidad social y que a su vez resultase novedoso. Todo ello ha originado un proyecto fin de carrera realizado con muchísima ilusión de principio a fin.

1.3. Objetivos

La idea original del proyecto era la de crear una aplicación web que resultase una mejora evidente a otro tipo de páginas como por ejemplo la de la asociación *El Refugio*. Esta asociación, y por tanto su página web, se dedica a denunciar el maltrato y el abandono a los animales, promover la defensa de los mismos y gestionar su centro de adopción. Siguiendo las características de la página web anterior se crea un entorno mucho más dinámico y sencillo, que pudiese ser utilizado desde cualquier dispositivo ayudando a este tipo de asociaciones.

De esta manera podemos establecer unos objetivos finales:

- Obtener una aplicación web mucho más sencilla que la página original de las asociaciones como *El Refugio*.
- Estudiar las necesidades primarias de las asociaciones y buscar la manera de enfocarlo en la aplicación resultante.
- Seleccionar las herramientas adecuadas para el desarrollo de la aplicación.
- Lograr aumentar el número de adopciones de mascotas.
- Facilitar el rescate de animales que estén en peligro o se hayan perdido, ya sea mediante geolocalización o códigos QR.

1.4. Estructura de la memoria.

La memoria del siguiente proyecto fin de carrera está estructurada en seis capítulos principales. El primero de ellos es el dedicado a la *Introducción*, donde se han mostrado los objetivos principales del proyecto.

Posteriormente nos encontramos con el capítulo dos, *Estado del arte*, en el que se hace un resumen de la tecnología en la que está basada el proyecto fin de carrera y un resumen del mismo. El tercer capítulo, *Tecnologías utilizadas*, se exponen cada una de las herramientas utilizadas y el porqué de su elección así como las características principales de cada una de ellas. Por tanto a su vez este capítulo se divide en: *aplicaciones web*, HTML5, CSS3, Node.js, MySQL y XAMPP.

El siguiente capítulo, *Diseño de la aplicación*, está dedicado al diseño principal de la aplicación; por ello consta de una breve introducción, una sección dedicada al diseño del Servidor, donde se explica el desarrollo del mismo, así como de la base de datos, las tablas que han tenido que ser creadas, etc. Posteriormente se realiza un diseño del Cliente donde se explican la implementación y características del cliente, así como los problemas principales que se han dado en su realización.

El capítulo cinco, *Resultados*, es una recopilación de los orígenes de la aplicación y de cada una de las pruebas realizadas.

Finalmente tenemos el capítulo seis, *Conclusiones*, dedicado a las conclusiones que han nacido al realizar el proyecto, así como una serie de posibles mejoras futuras.

Por último, los tres apéndices se corresponden con la publicación del código (ver apéndice A), con la planificación realizada (ver apéndice B) y con la licencia de la memoria del proyecto (ver apéndice C).

CAPÍTULO 2

ESTADO DEL ARTE

En este capítulo se hará un resumen de los conceptos más importantes de este proyecto fin de carrera. En primer lugar se explicará qué es una aplicación cliente-servidor para posteriormente profundizar en el concepto de las aplicaciones web, dando una visión más global de la idea principal de este proyecto fin de carrera, sus ventajas e inconvenientes, así como de la historia que ha influido en el desarrollo de las aplicaciones web.

2.1. Introducción

El desarrollo de este Proyecto fin de carrera se trata por tanto de un tipo de aplicación web basada en la arquitectura cliente-servidor (sección 2.2). En este caso la Arquitectura web dispone de una base de datos alojado de manera externa al servidor de la aplicación, de manera que tendremos una aplicación web de tres niveles (subsección 2.3.1).

El tipo de arquitectura elegida favorece a la idea inicial del proyecto, ya que una de las finalidades es que el uso de la aplicación sea independiente al de la plataforma, y por tanto, no ponga impedimentos a los usuarios para poder acceder a ella sin problemas.

Además en el desarrollo de este proyecto habían dos claras necesidades que había que cubrir, por un lado favorecer la *Escalabilidad* y por otro lado la *Concurrencia*. Para obtener ambas era importante el uso de las aplicaciones web (sección 2.3).

2.2. Aplicación Cliente-Servidor

Una aplicación cliente-servidor es una arquitectura de red en la que cada ordenador o proceso de red es o cliente o servidor. Existen también otro tipo de arquitecturas como *peer to peer* en la que cada máquina de la red posee responsabilidades equivalentes. La arquitectura cliente-servidor implica una relación entre aquellos procesos que solicitan un servicio (cliente) y el que los ofrece (servidor). Estos procesos se pueden ejecutar en el mismo ordenador o en distintos. Además permite la separación de funciones en tres niveles, cuya ventaja principal es que al tratarse de un desarrollo en diferentes capas, si hay que realizar algún cambio o modificación, se puede atacar solo al nivel requerido [7]. De esta manera a cada uno de estos niveles tiene una determinada misión facilitando

el diseño de arquitecturas escalables. Las tres capas se presentan a continuación tal y como se encuentra en la figura 2.1.

Lógica de presentación

Es la capa que está formada por un conjunto de componentes que implementan la interfaz de nuestra aplicación, mostrando información y manejando las interacciones del usuario. También es denominado interfaz gráfica y debe ser sencilla. Esta capa solo se comunica con la lógica de aplicación.

Lógica de aplicación

En esta capa están alojados los programas que se ejecutan, además recibe las peticiones del usuario y se encarga de enviar las respuestas. Es la capa en la que se establecen las reglas que deben cumplirse. Al tratarse de la capa intermedia se comunica tanto con la capa de presentación como con la lógica de datos.

Lógica de datos

Esta capa debe ser independiente para que pueda ser distribuida según las necesidades de la empresa en cada momento, además en ella se encuentran los datos y por tanto se accede a los mismos.

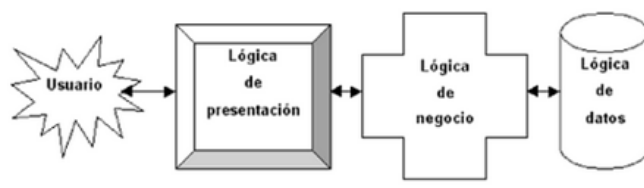


Figura 2.1. Separación Cliente Servidor [1].

La principal ventaja de esta arquitectura es que facilita la separación de los servicios según su función. Además las redes de ordenadores permiten que partes diferentes de una misma aplicación puedan darse en múltiples procesadores logrando *Concurrencia*. Se pueden cambiar aplicaciones de un procesador a otro. También nos permiten obtener *Escalabilidad* horizontal o vertical de la aplicación.

La escalabilidad horizontal se refiere a la capacidad de poder añadir o quitar clientes de la aplicación. La escalabilidad vertical esta relacionada con la capacidad de migrar el servicio hacia servidores de mayor capacidad o velocidad. Finalmente permite el acceso a los datos independientemente de donde se encuentre el cliente.

2.3. Aplicación web

Una aplicación web es un tipo especial de aplicación cliente-servidor, donde el cliente, el servidor y el protocolo mediante el que se comunican están estandarizados. Estas aplicaciones son populares debido a lo práctico del navegador web y a la independencia del sistema operativo, así como la facilidad para gestionar la aplicación web sin tener que instalar un *Software* [1].

Al principio de la computación cliente-servidor, cada aplicación disponía de su propio programa cliente, que servía como interfaz de usuario y tenía que ser instalado en cada ordenador. El cliente realizaba las peticiones oportunas al servidor, que era quien le daba la respuesta. Cualquier cambio o mejora en el servidor suponía tener que realizar un cambio en cada uno de los clientes instalados en cada ordenador, ocasionando un coste en soporte técnico y una pérdida de productividad.

Las aplicaciones web generan de forma dinámica una serie de páginas en un formato estándar que es el que soportan los navegadores comunes, como HTML (**H**yper **T**ext **M**arkup **L**anguage) o XHTML (**eX**tensible **H**yper **T**ext **M**arkup **L**anguage). Las páginas web se envían al servidor como documentos estáticos, pero la secuencia de las mismas ofrecen al usuario una experiencia interactiva.

El proceso completo desde que el cliente solicita un recurso al servidor, hasta que lo recibe, comienza en el momento en el que el usuario especifica en el cliente la dirección de la página que desea consultar y para ello el usuario escribe en el navegador la dirección de la página que desea visitar. De esta forma el cliente crea una conexión con el servidor, para posteriormente solicitar la página al servidor, que se encargará de enviar al cliente la página solicitada. Posteriormente se cierra la conexión entre el cliente y el servidor, esto se hace así para evitar desperdiciar ancho de banda de la red mientras el usuario lee la página. Cuando el usuario pincha otro enlace o busca otra página se establece una nueva conexión, por eso parece que siempre estamos conectados.

Las aplicaciones web se utilizan en tres entornos distintos: Internet, *Extranet* e *Intranet*. Una Intranet está formada por una red de ordenadores de una organización basada en el protocolo *TCP/IP* (**T**ransmission **C**ontrol **P**rotocol **I**nternet **P**rotocol) y que es accesible sólo por los miembros de la organización. Los *Cortafuegos* evitan el acceso a la red de usuarios no autorizados. En el caso de la aplicación web desarrollada en el proyecto el entorno utilizado será únicamente el de Internet.

2.3.1. Arquitecturas de las Aplicaciones web

La arquitectura de una aplicación web está basada en la aplicación cliente-servidor: por un lado el cliente (el navegador o explorador) y por otro lado el servidor web. Existen diferentes variantes de la arquitectura. Las arquitecturas más comunes son:

El cliente en el navegador, en el servidor incluida la base de datos (figura 2.2).

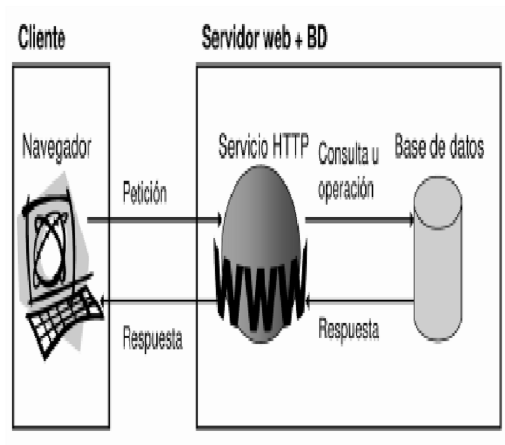


Figura 2.2. Arquitectura Web, todo en el servidor [1].

La base de datos alojada en otro servidor externo al de la aplicación (figura 2.3).

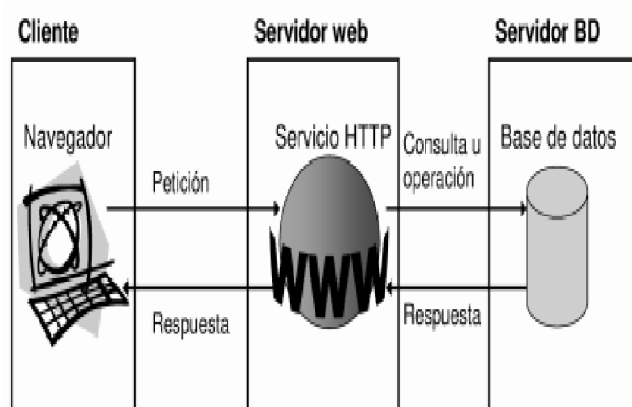


Figura 2.3. Arquitectura Web con la base de datos a parte [1].

Todo en el servidor unido también a un servidor de aplicaciones (figura 2.4).

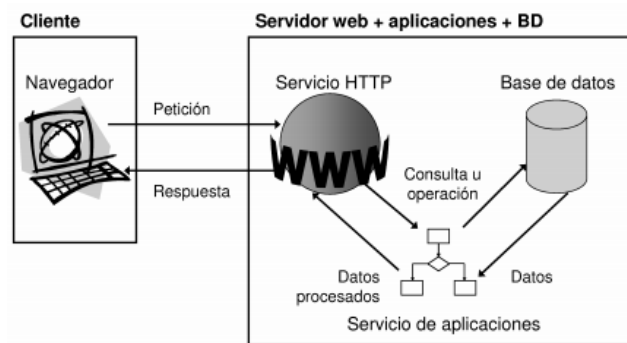


Figura 2.4. Arquitectura Web con todo en el servidor incluido servidor aplicaciones [1].

El objetivo de separar cada servicio en un servidor independiente es ofrecer mayor escalabilidad para obtener un mayor rendimiento

2.3.2. Ventajas e inconvenientes de las aplicaciones web

Para que quede aún más claro este concepto, se presentan a continuación tanto las ventajas como los inconvenientes que ofrece el uso de esta tecnología.

Una de las principales ventajas es que permite realizar tareas sin complicación y sin la necesidad de tener que descargar ningún programa, lo que se traduce en un ahorro de tiempo. Además se reduce el problema de gestionar el código en el cliente. Suponiendo que existe un navegador o explorador estándar en cada cliente. Todos los cambios, tanto de interfaz como de funcionalidad se realizan cambiando el código que está en el servidor web. Esto provoca un ahorro de tiempo ya que solo debemos actualizar en una sola máquina.

Otra ventaja bastante significativa es que evita la gestión de versiones, sin tener problemas de inconsistencia en las actualizaciones, porque no existen clientes con diferentes versiones de la misma aplicación, ya que el *Software* lo gestiona el propio desarrollador.

El uso de las aplicaciones web no presenta problemas de compatibilidad ni ocupa espacio en nuestro disco duro. Debido a que la mayor parte de la aplicación no se encuentra en el navegador, el uso de la aplicación no consume los recursos de la máquina.

Además las aplicaciones web son independientes de la plataforma. Para que se puedan ejecutar en diferentes plataformas, sólo necesitamos un navegador para cada plataforma, sin tener que adaptar el código a ninguna de ellas. Además, se ofrece una interfaz gráfica igual para todos los usuarios.

Pero también presenta una serie de inconvenientes. El uso de las aplicaciones web ofrece menos funcionalidad que una aplicación de escritorio. Una aplicación de escritorio es un programa informático diseñado para ayudar al usuario a realizar singulares o múltiples tareas específicas, relacionadas como por ejemplo *OpenOffice*, *Excel*, *Buscaminas*... Además se depende de un tercero; por lo que si, por ejemplo, el servidor cae no podremos acceder a la aplicación web.

La programación en la web no es tan versátil o potente como la tradicional, aunque esta desventaja este desapareciendo rápidamente. Al principio las aplicaciones web eran de "solo lectura": ya que se permitía una interacción con el usuario casi nula, pero con la aparición de nuevas herramientas como Java, Javascript esta limitación tiende a desaparecer.

2.3.3. Historia de las aplicaciones web

El desarrollo de las aplicaciones web ha revolucionado el uso de Internet ya que permite al usuario acceder de un modo más interactivo al contenido de las páginas. Además de haber aumentado la cantidad y calidad de los contenidos, ya que hace diez años, por ejemplo, no se podían chatear y ahora está a la orden del día. Al igual que realizar compras por Internet de forma segura.

Hay varios acontecimientos que provocaron el despegue de las aplicaciones web, y que, a pesar de darse en un periodo de tiempo bastante largo están relacionados entre sí [8].

Uno de los primeros lenguajes de programación, diseñado por Larry Wall en 1987, y creado específicamente para el desarrollo de aplicaciones web fue Perl. Sin embargo este fue creado antes de que Internet pasase a ser utilizado por el público en general; es por ello que realmente es el nacimiento de PHP(**H**ypertext **P**reprocessor), por parte de Rasmus Lerdorf, lo que provoca el auge de las aplicaciones web. PHP es un lenguaje adecuado para el desarrollo web, que puede ser incrustado en HTML y de código abierto. El auge de este lenguaje esta relacionado con lo sencillo que resulta para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

Posteriormente nace *JavaScript*, un lenguaje de programación interpretado que no requiere de compilación ya que funciona en el lado del cliente y que ofrece muchas posibilidades utilizadas para crear pequeños programas y ser insertados en una página web, lo que permite a los programadores cambiar de forma dinámica el contenido de las paginas web; permitiendo a los usuarios un acceso mucho más interactivo

En 1996 Sabeer Bhatia y Jack Smith lanzaron *Hotmail*, un servicio de correo electrónico que permitía al público acceder a su correo en cualquier momento y desde cualquier lugar, incluso lejos de su ordenador.

En 1997 apareció la plataforma *Flash*, un software originalmente diseñado para

crear animaciones que se pueden usar en páginas web, permitiendo el uso de contenido multimedia.

En 1998 dos jóvenes estudiantes (Larry Page y Sergey Brin) crean el buscador *Google* que aunque no fue el primero, sí que superó en poco tiempo al resto gracias a que su interfaz simple y amigable, y los resultados muy relevantes facilitaban enormemente la búsqueda de información.

En 2001 nace *Wikipedia*, una enciclopedia en línea, que además permite a los usuarios agregar contenido, consiguiendo en el primer año una contribución desorbitada por parte de los usuarios (20000 páginas en 18 idiomas)

En 2003 nace *MySpace*; convirtiéndose en una plataforma de ejemplo para otras aplicaciones web como *Youtube*.

Pero es en el 2004 cuando ocurren los tres acontecimientos más relevantes. En primer lugar en una conferencia de la Web 2.0 es mencionado por primera el concepto de web como plataforma. En segundo lugar nace *Digg*, un sitio web sobre información de ciencia y tecnología y que propuso una idea innovadora de acceso a noticias y poder compartirlas. Por último nace *Facebook* convirtiéndose en la actualidad en la red social más importante, no sólo por contar con unos 900 millones de usuarios. En 2005 nace *Youtube* de manera oficial, permitiendo a los usuarios compartir vídeos. Por otra parte *Twitter* nace en el 2006, sufriendo un crecimiento enorme desde el 2007 al 2012. En el 2007 nace *iPhone*, que se convierte en directo responsable de las plataformas móviles y las aplicaciones web.

2.3.4. Futuro de la aplicación web

Con el paso de los años las aplicaciones web se están consolidando consiguiendo una funcionalidad mucho más completa y sencilla. El impacto de estas aplicaciones está afectando en la forma en la que los usuarios interactúan con Internet y con su vida en general; tanto a nivel de negocio como de relaciones sociales. Es posible trabajar desde tu propio hogar, hacer la compra o mantener conversaciones con personas que se encuentran a miles de kilómetros. Este tipo de aplicaciones han creado un impacto social enorme, afectando en la vida de las personas y en la sociedad considerablemente. En resumen, los desarrolladores actuales están dando forma a lo que será el futuro digital de las aplicaciones web.

CAPÍTULO 3

TECNOLOGÍAS UTILIZADAS

En este capítulo se hará una recopilación de cada una de las tecnologías utilizadas en la realización del proyecto, así como una descripción de cada una de ellas.

3.1. HTML5

HTML es un lenguaje basado en etiquetas, con el que se definen las páginas web. Las etiquetas (entre otras muchas cosas) le dicen al navegador en qué juego de caracteres está la página, de qué tipo es cada uno de los fragmentos de texto, si están centrados o alineados hacia un lado, el tipo letra en el que está el texto (cursiva, negrita, etc) o su tamaño, si hay tablas, de qué anchura son etc. Por tanto, las etiquetas le dan al navegador las indicaciones para poder presentar la página en pantalla de la forma adecuada. HTML5 es un lenguaje de referenciación; es decir, para añadir elementos externos a la página, como podría ser imágenes, es necesario hacer referencia a la ubicación de dicho elemento mediante texto.

3.1.1. Historia de HTML5

El Consorcio World Wide Web [9] es un consorcio internacional donde las organizaciones miembro, personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web. Liderado por el inventor de la Web, Tim Berners-Lee, y el Director Ejecutivo, Jeffrey Jaffe, la misión del W3C es guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web. El W3C trata de alcanzar su objetivo principalmente a través de la creación de Estándares Web y Pautas.

Es por ello que el W3C es el encargado de establecer las distintas definiciones de HTML hasta llegar a la versión actual, HTML5. Este proceso es el resultado de varios años de evolución y para poder entender mejor las ventajas e inconvenientes de este lenguaje es necesario realizar primero un repaso de su historia, con el fin comprender el desarrollo desde sus inicios hasta HTML5 [10].

Es en 1989 cuando nace la primera versión de HTML con el objetivo de crear estructuras básicas de páginas web y así gestionar su contenido y facilitar el acceso; mejorando de esta forma el uso deficiente de acceso a Internet, ya que desde su creación

encontrar una búsqueda alrededor de Internet resultaba una tarea difícil. El contenido de la información se trataba como documentos individuales y la única forma de poder acceder a ellos era conociendo las direcciones de los mismos. Para ponerle solución a este prosaico sistema Tim Berners-Lee crea dos tecnologías: HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) y HTML.

Desde 1989 HTML sufrió varias revisiones y en 2004 se crea el Web WHATWG (**H**yper**T**ext **A**pplication **T**echnology **W**orking **G**roup): una comunidad de personas interesadas en la evolución de HTML y las tecnologías conexas, nacido al margen del W3C con el propósito de crear una nueva versión del estándar no tan académico y mucho más práctico. En 2007 el W3C decide utilizar como base el trabajo del WHATWG para proponer una nueva actualización del estándar, naciendo en 2008 la primera versión de HTML5.

3.1.2. Características de HTML5

HTML5 aporta nuevas características al lenguaje HTML, las nuevas funcionalidades se indican a continuación [11].

Semántica Una de las innovaciones de HTML5 son las relacionadas con la Web Semántica, es decir, la inclusión de *metadatos* semánticos a la web para que sea más fácil de comprender por navegadores o motores de búsqueda; enriqueciendo el intercambio y flujo de datos, y favoreciendo al desarrollo de las tecnologías y consiguiendo así que los usuarios encuentren en los buscadores la información que desean obtener de forma más exacta, obteniendo de esta forma mejores resultados.

Geolocalización Otra de las novedades de HTML5 es la incorporación de nuevas APIs de Javascript aumentando la potencia del lenguaje. Una de ellas es la *API* (**A**pplication **P**rogramming **I**nterface) de geolocalización que nos permite obtener la ubicación del usuario, siempre que el navegador la tenga implementada y que el usuario dé su permiso. A continuación se muestra la versión de los navegadores que soporta la geolocalización de HTML5 3.1. La geolocalización es una de las funcionalidades nuevas incorporadas en el diseño e implementación de la aplicación web de este proyecto fin de carrera.



Figura 3.1. Soporte para la API de geolocalización de los distintos navegadores [2].

Offline Otra de las nuevas características de HTML5 es que permite que nuestras aplicaciones puedan funcionar de forma offline, es decir, desconectada. En la actualidad podemos guardar la información de forma local mediante localStorage, o IndexedDB (almacenamiento de la información de forma local, es decir, en el lado del cliente). Pero estas funcionalidades tan interesantes pueden resultar insuficientes si la página web no puede ser cargada por falta de conectividad; es por ello que HTML5 tiene mecanismos para detallar aquellas páginas que deben estar siempre disponibles, sabiendo en cada momento en que forma de trabajo estamos en ese momento.

3.1.2.1. Diferencias con HTML4

La última versión de HTML presenta, por tanto, bastantes diferencias con respecto a su antecesor HTML4.

- HTML5 respeta la compatibilidad hacia atrás para poder seguir procesando de manera correcta las páginas web creadas con HTML4.
- HTML5 seguirá siendo un lenguaje de etiquetado pero con la posibilidad de combinarlo con la síntesis XML.
- HTML5 nos simplifica y facilita la programación hasta en las cosas más sencillas como la declaración del DOCTYPE. En la imagen que se presenta a continuación se ve claramente la diferencia.

3.1.3. Ventajas e inconvenientes de HTML5

HTML5 tiene nuevas características sintácticas, incluyendo tags como <vídeo>, <audio>y <canvas>lo que facilita la integración multimedia sin utilizar flash. Además de nuevas etiquetas como <section>, <article>,<header>y <nav>enriqueciendo el valor semántico del documento [12].

Con HTML5 no es necesario utilizar *plugin*, ni programas de terceros para poder integrar audio vídeo ya que esta funcionalidad esta integrada en el navegador. Los elementos de HTML5, aumentan el valor semántico de la página web, con etiquetas que permiten clasificar y ordenar en diferentes capas y estructuras el contenido. Además, añade *metadatos* de manera más formal, favoreciendo el posicionamiento SEO (**S**earch **E**ngine **O**ptimization), es decir las siglas en inglés de lo que se conoce como el posicionamiento en buscadores, que consiste en la optimización de los mismos con el fin de mejorar la visibilidad de los diferentes sitios web en los resultados de los buscadores.

HTML5 facilita realizar diseños que son adaptables a diferentes dispositivos (web, tablets, móviles). El código es más sencillo realizando páginas más fluidas que se cargan más rápidamente, favoreciendo la usabilidad y la indexación en buscadores. Además ofrece una compatibilidad mayor con los navegadores de dispositivos móviles.

Otra de las ventajas que ofrece es la Caché de aplicación, que permite a los desarrolladores acceder a determinados recursos de la web sin necesidad de conexión. Esto además mejorará notablemente la velocidad, puesto que al meter determinados recursos locales

en la caché, provocamos un acceso a los mismos de un modo mucho más rápido. Además el navegador solo descargará del servidor aquellos recursos que han cambiado. HTML5 incorpora una base de datos en el lado del cliente (IndexedDB), en la que cada registro está compuesto de una clave y un objeto. Además de la geolocalización, permitiendo conocer la ubicación geográfica del usuario, siempre y cuando el navegador lo permita.

Sin embargo no todo en HTML5 es idílico. Como cualquier lenguaje de programación también presenta algunos inconvenientes. El principal problema de la aceptación de HTML5 es que sólo los navegadores modernos soportan. Además, aunque partes de la lengua son muy estables, el lenguaje en sí mismo es considerado como un trabajo en progreso, por lo que técnicamente, cualquiera de los elementos podría cambiar en cualquier momento.

3.2. CSS3

Al igual que HTML5 nos permite definir la estructura de nuestra página web, CSS3 es el encargado de establecer el diseño y estilo de la página en los diferentes dispositivos. Por tanto, las hojas de estilo nos permiten definir la representación de la página web de una manera eficiente, logrando separar el contenido de la forma.

La primera de las versiones CSS nace en 1996 y fue recopilando adeptos, ganando en popularidad y consiguientemente llegar hasta la versión 2.1, estándar que ya ofrecía una considerable compatibilidad con casi todos los navegadores. Es en el año 2005 cuando comienza a definirse la versión CSS3, y con la llegada de esta última versión entran en juego multitud de nuevas características [13]:

- Módulo de color de CSS3: La última versión ofrece nuevos colores.
- Bordes degradados y con imágenes lo que se logra con la propiedad `border-image`. Además de esquinas y bordes redondeados: Con CSS3 se puede diseñar de forma más sencilla bordes redondeados con la propiedad `border-radius`.
- Cajas con sombras: CSS3 permite obtener esta funcionalidad mediante la propiedad `box-shadow`.
- Múltiples columnas: CSS3 incorpora nuevos atributos para que el navegador se encargue de producir texto multicolumna, es decir, que maquete directamente el texto en varias columnas sin tener que hacer nosotros nada.
- Efectos de texto: CSS3 permite añadir nuevas propiedades que crean diseños de texto más atractivos, entre ellos destacan `text-shadow`, que permite añadir efectos de sombras en el texto.
- Animaciones: CSS3 ofrece la posibilidad de crear resultados parecidos a las animaciones con *Flash*. Se trata de la regla *Keyframe*, dentro de ella se encuentra el estilo CSS que define la animación. Además es necesario asignar un nombre que identifique al keyframes, de este modo, se puede aplicar el efecto desde el objeto que queramos.

3.2.1. Ventajas es inconvenientes de CSS3

El uso de CSS3 permite conseguir efectos visuales que antes solo eran posibles con el uso de tecnologías adicionales. Además ahorramos tiempo y trabajo al poder seguir varias técnicas (bordes redondeados, sombra en el texto, sombra en las cajas...) sin necesidad de usar un editor gráfico.

Otra de las ventajas que ofrece es que la carga de la página descenderá gracias a la técnica CSS Sprites [3] que consiste en un conjunto de imágenes agrupadas en una única imagen, a cada una de ellas se accede mediante un desplazamiento respecto del borde superior izquierdo, cota (0,0), y sus dimensiones absolutas. Es, por tanto, un método de agrupación de imágenes que, posteriormente, pueden ser tratadas de forma independiente. La carga disminuirá porque si al utilizar múltiples imágenes, se realizarán varias llamadas al servidor y ocasionará una sobrecarga directamente proporcional al acceso a la página, por el contrario, al utilizar CSS Sprites se reducirá sustancialmente la carga, debido a que se descargará múltiples imágenes con una sola llamada, un solo archivo.

Fijémonos en el ejemplo que se ve en la figura (3.2), las tres primeras figuras de una supuesta barra de herramientas, se han agrupado en una única en formato sprite, que utilizaremos en nuestros ejemplos; ahora en la carga de la página ahorraremos dos accesos al servidor al tener que referenciar una única imagen.

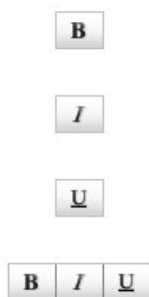


Figura 3.2. Ejemplo visual CSS sprites [3].

Pero también presenta inconvenientes como el uso de tablas también permite diseñar estilos complejos de manera más sencilla que mediante CSS, sin embargo CSS3 está intentando facilitar este trabajo.

3.2.2. ¿Por qué HTML5 y CSS3?

Las características principales que sitúan a HTML5 y CSS3 [14] a la vanguardia frente a competidores son varias. Obtiene sitios web de alto impacto visual debido al uso de CSS3 que favorece a la presentación de la página web, permitiendo lograr estilos y efectos visuales que antes no era posible conseguir, más llamativos y modernos, además permite el desarrollo de aplicaciones web compatibles con *iPhone*, *Android* y otros dispositivos. Al no utilizar *plugin* evita las desventajas que puede traer en compatibilidad, rendimiento y seguridad. Como ya se menciona anteriormente, mejora la semántica de los sitios web favoreciendo la búsqueda.

3.2.3. Sitios que utilizan HTML5 y CSS3

Son varias las aplicaciones reales que utilizan HTML5 y CSS3 [4]. El uso de HTML5, que permite a algunos vídeos reproducirse de forma nativa en el navegador web, sin añadidos de software adicional, supone que algunos contenidos de *Vimeo* (figura de la página 3.3) se cargarán en menos de un segundo, según relata Brad Dougherty, un ingeniero de aplicaciones en el blog de la compañía.



Figura 3.3. Vimeo [4].

Gmail utiliza también Web Storage para guardar los correos más recientes en el disco del usuario, así como Drag and Drop para arrastrar y soltar archivos. Para esto último puedes probar en *Gmail*, crea un email e intenta arrastrar un archivo del explorador de archivos al correo y verás que es posible adjuntarlo con sólo arrastrarlo. El gesto de arrastrar y soltar ahora es posible gracias a HTML5.

3.3. Node.js

La web se ha vuelto más interactiva, los usuarios de todo el mundo quieren comunicarse entre sí y en tiempo real, en actividades como, chatear, jugar online, intervenir y actualizar en redes sociales... Esto implica una comunicación entre cliente y servidor a través de la red y de manera instantánea. Pero existen además otra característica casi más importante; esta comunicación debe no solo darse en tiempo real, sino además a gran escala, soportando cientos, miles y millones de usuarios.

Para que esto sea posible necesitamos una comunicación en tiempo real y escalable entre cliente y servidor. Es aquí donde entra Node.js [15], que es capaz de solucionar estos problemas con bastante eficacia, podría decirse de hecho, que ambas características son las principales de Node.js, tratándose por tanto de un *Framework* de programación en el que no solo podemos crear pequeños script, sino que también nos permite desarrollar completas aplicaciones web, rápidas y fiables. Es muy idóneo para la comunicación a gran escala y en tiempo real.

Utiliza un modelo I/O (entrada/salida) orientado a eventos, es decir, una propuesta tecnológica que es adoptada por una comunidad de programadores y en la que el flujo del programa está determinado por eventos o mensajes desde otros programas o hilos de ejecución. Además es no bloqueante que permite poder realizar otras tareas en vez de esperar a que los datos estén disponibles dotándole de más eficiencia y ligereza, muy apropiado para aquellas aplicaciones en tiempo real que hagan uso de datos intensivos y que se ejecuten mediante dispositivos distribuidos.

3.3.1. Comparativa Node.js con Apache

En esta sección se realizará una comparativa entre los Servidores Node.js con otro tipo de servidores como Apache, y los lenguajes utilizados en cada uno de ellos [5]. En la tabla 3.1 se muestra un resumen de las mismas además de un gráfico comparativo de las conexiones de ambos servidores (figura 3.4).

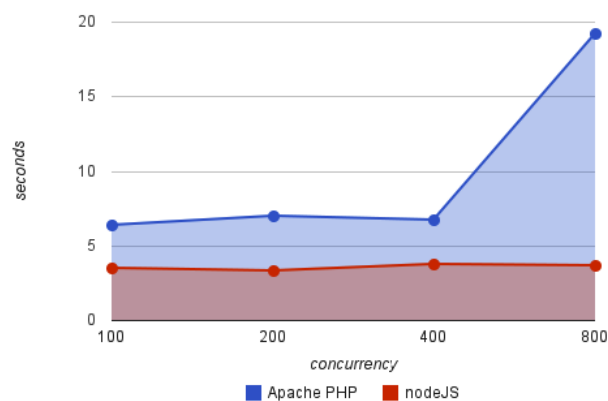


Figura 3.4. Gráfico conexiones Apache y Node.js [5].

Característica principal	Node.js	Apache
<i>Concurrencia</i>	Atiende varias peticiones a la vez con un solo hilo	Crea un nuevo hilo por cada conexión cliente-servidor, lo cual funciona bastante bien cuando hay pocas conexiones; sin embargo crear nuevos hilos es costoso, lo que puede apreciarse bien en la figura inferior, ya que a partir de 400 conexiones simultáneas aumenta el número de segundos necesarios para atender a cada una de las peticiones
<i>Inicialización y carga del código</i>	Tiene todo cargado en memoria que lo ha cargado una sola vez antes de gestionar la petición	Necesita ejecutar la inicialización y carga en cada petición
<i>Comunidad</i>	Aunque esté creciendo todavía no es muy grande	Bastante más grande comparado con Node.js

Tabla 3.1. Diferencias entre Node.js y Apache

3.3.2. Uso de Node

A la hora de decidir si utilizar o no Node.js tenemos que tener en cuenta que su uso sería apropiado en aquellas aplicaciones web que necesiten conexión permanente [16]. Algunos ejemplos en los que Node.js encaja perfectamente serían: Juegos online, correo online que nos permitiría, siempre que el navegador estuviese abierto, ver notificaciones en tiempo real de nuevos correos recibidos, herramientas de traducción en tiempo real. También es muy útil en redes sociales ya que nos permitiría actualizar las novedades de manera instantánea, siempre que quisiésemos compartir alguna actualización.

Como cualquier herramienta Node.js puede no ajustarse a lo que necesitamos en el desarrollo de una aplicación web. Por ello, a la hora de decidir si utilizar o no Node.js como servidor deberíamos responder, en función de aquello que deseamos, a las siguientes preguntas. Si contestamos en todas que sí, entonces Node.js se ajustaría perfectamente a nuestra aplicación, pero si por el contrario, en alguna de las preguntas contestamos que no, entonces puede que Node.js no sea la mejor solución y por tanto, deberíamos buscar otro servidor que se ajuste mejor a nuestras necesidades.

- ¿Es importante la concurrencia y los tiempos bajos de respuesta?
- ¿Será alto el número de usuarios activos?
- ¿Se trata de un proyecto pequeño?

Node.js forma parte de un gran ecosistema, con una gran comunidad a su alrededor. Existen diferentes ejemplos de empresas que utilizan Node.js. Por ejemplo, *LinkedIn* utiliza Node.js en combinación con HTML5 en su aplicación móvil para ofrecer la máxima rapidez posible. En la página oficial de Node.js aparecen todas las empresas que trabajan con esta herramienta y una breve explicación de su uso. A continuación se muestran algunos de los ejemplos más conocidos con la explicación que aparece en la web oficial [17] del porqué de su utilización (figura 3.5).



Figura 3.5. Gráfico de empresas que trabajan con Node.js [5].

3.4. MySQL

MySQL es un sistema de gestión de bases de datos, es decir una serie de programas que permiten el almacenamiento, modificación y extracción de la información, así como añadir, borrar modificar y analizar datos en una base de datos. Una de las características principales de MySQL es que es relacional, esto es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común.

3.4.1. Características de MySQL

MySQL se trata de un sistema gestor de bases de datos muy conocido y bastante utilizado por su simplicidad y notable rendimiento. Resulta una opción atractiva para aplicaciones comerciales, como por ejemplo para el entretenimiento debido a su facilidad de uso y tiempo reducido de reacción. Además la distribución libre en Internet mediante licencia *GPL* (**G**eneral **P**ublic **L**icense) le aporta estabilidad y facilidad de desarrollo. Las principales características de MySQL [18] son:

- Se distribuye en ejecutables para plataformas diferentes.
- La *API* se encuentra disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl. Además está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta y se puede utilizar como cliente-servidor o incrustado en aplicaciones, además soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferente para poder optimizar el sistema de gestión de bases de datos a cada caso concreto.
- Su administración se basa en usuarios y privilegios y las opciones de conectividad abarcan TCP/IP, sockets, *Unix* y sockets NT, además de soportar completamente *ODBC* (**O**pen **D**ata**B**ase **C**onnectivity).
- Altamente confiable en cuanto a estabilidad se refiere.

3.5. XAMPP

XAMPP es un servidor web independiente, es decir un sistema de archivos agrupados y relacionados para cumplir un objetivo común que es hacer funcionar un Sistema Web: ya sea desde una página simple hasta algo más complejo, desde el propio ordenador. XAMPP esta conformado por la base de datos MySQL, el servidor web Apache, además de PHP y Perl y actúa como software libre, capaz de interpretar páginas dinámicas y de uso visual y sencillo.

XAMPP fue diseñado por Baiker y Anthony Cooperation; pretendían que su uso fuese únicamente de desarrollo, ya que querían facilitar a los diseñadores de sitios webs probar sus trabajos con sus propios ordenadores y sin acceso a Internet, sin embargo, en realidad XAMPP es utilizado como servidor de sitios Web ya que tiene la seguridad suficiente para poder serlo, incluyendo una herramienta especial que protege las partes más importantes. Por tanto lo que incluye XAMPP [19] es lo siguiente:

APACHE: Apache es la aplicación de servidor web que procesa y entrega el contenido web a un ordenador.

MySQL: Cada aplicación web, ya sea simple o compleja, requiere una base de datos para almacenar los datos que recoge. MySQL, que es de código abierto, es el sistema gestor de base de datos más popular.

PHP: Es un lenguaje de script del lado del servidor que hace funcionar algunas de las páginas web más famosas del mundo, incluyendo WordPress y Facebook

Perl: Perl es un lenguaje de programación dinámico y de alto nivel que se usa ampliamente en la programación de redes, la administración de sistemas, etc.

La herramienta principal es phpMyAdmin, escrita en PHP, que permite la gestión de la base de datos MySQL (crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL en general) a través de páginas web consecutivas, proporcionando una interfaz de usuario muy orientativa (figura 3.6).

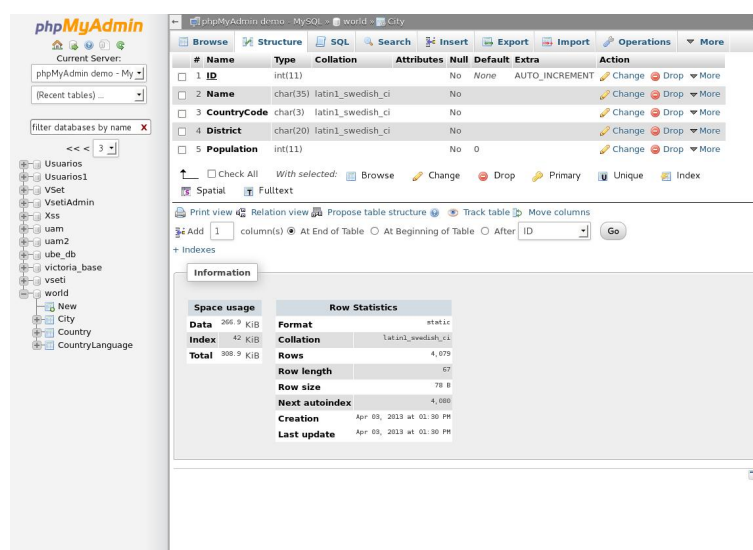


Figura 3.6. Vision de PHPMyadmin[6].

3.5.1. Panel de control de XAMPP

XAMPP dispone de un panel de control bastante útil a la hora de visualizar que aplicación se esta ejecutando. En el que se distinguen las siguientes partes [6], tal y como se muestra en la figura 3.7:

- La zona de módulos, en la que se muestra para cada uno de los módulos de XAMPP: si está instalado como servicio, su nombre, el identificador de proceso, el puerto utilizado e incluye unos botones para iniciar y detener los procesos, administrarlos, editar los archivos de configuración y abrir los archivos de registro de actividad.

- La zona de notificación, en la que XAMPP informa del éxito o fracaso de las acciones realizadas
- La zona de utilidades, para acceder rápidamente

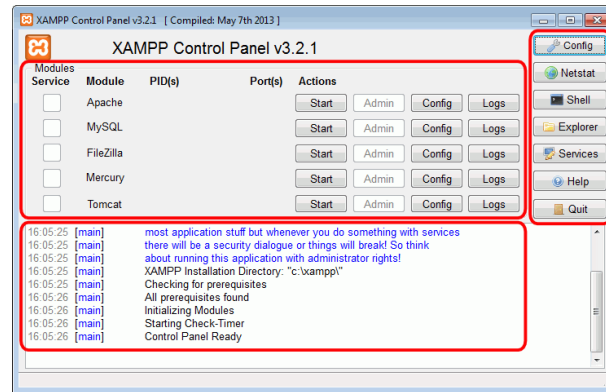


Figura 3.7. Ventana control XAMPP[6].

CAPÍTULO 4

DISEÑO DE LA APLICACIÓN

Este capítulo es el eje principal de la memoria, en el que analizaremos el uso principal de la aplicación web, explicando cada una de sus funcionalidades de la manera más cercana al lector posible. El desarrollo de cada uno de estos objetivos y los problemas que ha ocasionado, para terminar por dar una posible visión futura de la que podría ser el desarrollo de la aplicación. A lo largo de este capítulo se explicará cada uno de los módulos que constituyen el proyecto en su totalidad.

Esta constituido por tres partes claramente diferenciadas: El cliente, el servidor y la base de datos (figura 4.1). La aplicación web se constituye básicamente de:

- El cliente que se ha realizado con el lenguaje de programación de HTML5 y que por lo tanto constituye la primera de las capas de nuestra aplicación web.
- El servidor que se encargará de atender las peticiones del cliente y que esta basado en Node.js.
- Así mismo, este servidor utilizará la base de datos del servidor propio de la aplicación XAMPP, el cual trabaja sobre Apache.
- El último de los módulos realizará las búsquedas y actualizaciones en la base de datos.

Tras analizar de una manera muy superficial cada uno de los módulos, se analizarán en posteriores secciones cada parte de una manera más profunda y concisa.

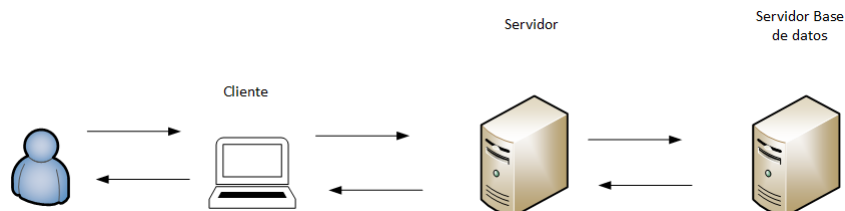


Figura 4.1. Esquema de la Arquitectura

En la figura 4.2 se puede observar un gráfico con las nuevas funcionalidades incorporadas con respecto a la página web oficial de la asociación como por ejemplo la del *El Refugio*.

En él se observa las características prioritarias de la aplicación y las funciones que ofrece a los usuarios que accedan a la aplicación.

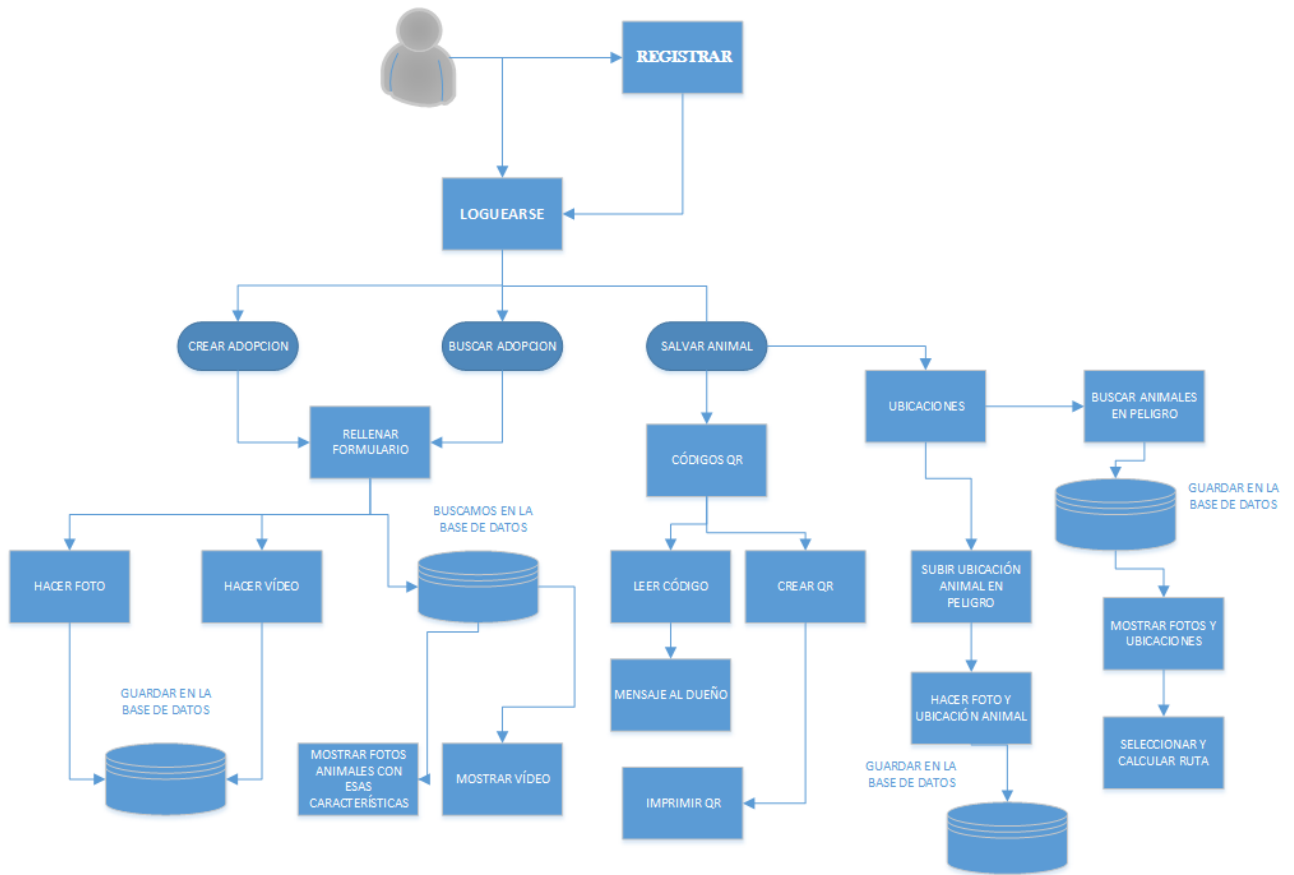


Figura 4.2. Esquema de Funcionalidad

4.1. Diseño del Servidor

En este apartado del proyecto se explicará tanto las características propias del servidor como de la base de datos utilizada en el desarrollo. La aplicación web diseñada en este proyecto fin de carrera presenta la arquitectura web en la que la base de datos está alojada en un servidor externo al de la aplicación. Comenzaré explicando el modelo de datos para después ver de una manera más clara las características del servidor.

4.1.1. Creación del modelo de datos.

Como servidor web [20] adicional para la puesta en marcha de la base de datos me encontré con las siguientes posibilidades: XAMPP, LAMP, MAMP Y WAMP.

En la figura (4.3) se muestran los cuatro servidores principales.

Pero mi elección se debatió entre los dos que me permiten trabajar en Windows (XAMPP



Nombre	Descripción	Logo
WAMP SERVER	Wamp Server utiliza las siguientes herramientas: Windows como Sistema Operativo, Apache, como servidor web, MySQL, como gestor de bases de datos y PHP como intérprete del lenguaje de programación PHP.	
LAMP SERVER	Lamp Server es un sistema análogo pero que funciona bajo el Sistema Operativo Linux.	
MAMP SERVER	Mamp Server es un sistema análogo pero que funciona bajo el Sistema Operativo Macintosh.	
XAMPP SERVER	Este paquete tiene la característica principal que se puede instalar para cualquiera de los diferentes sistemas operativos, es decir, existen versiones tanto para Windows como para Linux, Macintosh. Siendo, también, un sistema análogo al Wamp Server.	

Figura 4.3. Servidores web [20]

y WAMP). Las características de ambos son bastante similares, además los servicios que ofrecen son los mismos: PHP, Apache y MySQL. Esto complicó bastante la elección definitiva. Una de las razones por la que me decanté por XAMPP, a pesar de que mi proyecto se haya desarrollado en Windows, y por tanto, pudiese utilizar cualquiera de los dos servidores es porque si tenía que mover mi proyecto a otro Sistema Operativo XAMPP me lo permitiría al ser multiplataforma; a diferencia de WAMP que solo trabaja sobre Windows. Si me familiarizaba con XAMPP y por cualquier motivo tuviese que mudar de sistema operativo no tendría que conocer una nueva herramienta. Pero la razón principal es que al iniciar XAMPP, aparece la ventana XAMPP Control Panel, en la que están reflejadas las aplicaciones que están corriendo en ese momento. Estas aplicaciones son: Apache, MySQL, PHP y Perl. Con WAMP es necesario ir navegando por el menú de su icono de la barra de tareas para saber qué servicios están funcionando, lo cual es muy incómodo. Por tanto la búsqueda de una herramienta más visual y sencilla fue primordial (figura 4.4).

Para poder utilizar el servicio MySQL ofrecido por el servidor web elegido, antes hubo que instalar XAMPP. Para ello en primer lugar descargamos la aplicación en la página web correspondiente <https://www.apachefriends.org/download.html>.

Una vez instalada y siempre que queramos usarla es necesario abrir el panel de control y hacer clic en el botón Start correspondiente a Apache. Como Apache abre puertos en el ordenador (al menos la primera vez), el Cortafuegos de Windows pide al usuario confirmación. Para poder utilizarlo habrá que autorizar el acceso en las redes privadas.



Figura 4.4. Panel de control XAMPP.

Si el arranque del servidor Apache tiene éxito, el panel de control mostrará el nombre del módulo con el fondo en verde, su identificador de proceso y los puertos abiertos (HTTP y HTTPS). El botón Start de cada uno de los módulos se convertirá en el botón Stop y en la zona de notificación se verá el resultado de las operaciones realizadas.

En la primera de las figuras 4.5 se muestra el panel de control sin haber iniciado los módulos

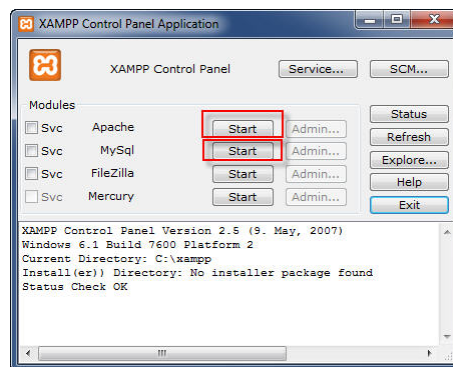


Figura 4.5. Panel de control sin iniciar módulos.

En la figura 4.6 se muestra el panel de control iniciado.

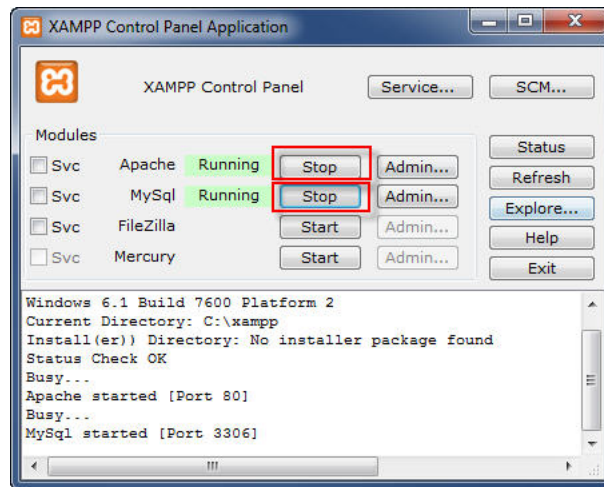


Figura 4.6. Panel de control iniciado.

Para comprobar que todo funciona correctamente, hay que escribir en el navegador la dirección `http://localhost`. Al abrir la página por primera vez, XAMPP pedirá seleccionar el idioma. Una vez elegido el idioma, se mostrará la página de configuración (figura 4.7).

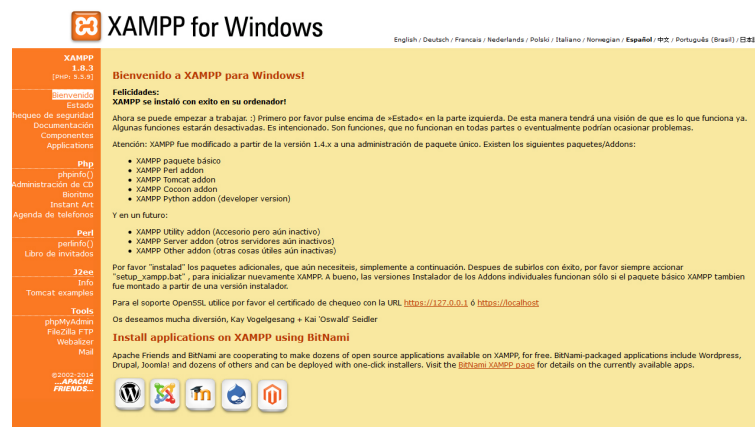


Figura 4.7. Página de configuración XAMPP.

Una vez hecho esto accedemos a la herramienta phpMyAdmin por si deseamos visualizar, modificar o borrar las bases de datos.

Además para poder trabajar con Node.js y la base de datos fue necesario descargar el paquete node-mysql con el fin de utilizar Javascript y a la vez MySQL. Una vez establecida la conexión creamos las bases de datos correspondientes para que luego puedan ser usadas sin problema. La interacción a cualquiera de las tablas se hará con el protocolo propio de MySQL.

4.1.1.1. Tablas de datos

Para poder llevar a cabo el proyecto ha sido necesario implementar una serie de tablas que forman parte de una misma base de datos denominada Usuarios. Las tablas creadas son las que se especifican a continuación. Además se muestra un gráfico a modo de resumen a cerca de como están estructuradas la base de datos (figura 4.8).

- **User:** Esta tabla creará un campo nuevo siempre que un usuario decida registrarse en la aplicación, cuando no exista ya un usuario con el mismo nombre de sesión, ya que en ese caso tendrá que utilizar otro nombre distinto. Los campos correspondientes a esta tabla serán (id, usuario y password).
- **Mensaje:** Esta tabla es la encargada de gestionar los mensajes enviados a los usuarios registrados en la aplicación. Dispondrá de una serie de campos: usuarioenvia en cuyo campo se registrará el nombre del usuario que escribe el mensaje, usuariorecibe campo que gestiona el receptor del correo, mensaje el encargado de guardar el mensaje correspondiente, el cual tiene un número de caracteres limitado.
- **Fotos:** La tabla fotos se encargará de guardar cada una de las fotos que el usuario haya hecho a la mascota. Los campos utilizados para poder gestionarla han sido los siguientes: usuario ya que necesitamos hacer corresponder cada tabla con el usuario que la ha subido y añadimos este campo a la tabla; de esta manera guardaremos el nombre del creador de la foto, Img donde guardaremos la imagen correspondiente. Los campos que siguen a continuación fueron creados para establecer las características principales del animal en adopción y así a posteriori facilitar la búsqueda de un animal: El nombre del animal en adopción, la historia del animal en adopción, el sexo de la mascota, el tipo (distinguiendo entre perros y gatos ya que son los animales que se ofrecen en la web original) y la edad.
- **Fotos subidas:** Al igual que en la tabla anterior, deberá de tener los mismos campos que esta. La diferencia principal reside en que esta tabla gestiona las fotos subidas, no las realizadas directamente desde la aplicación, por ello no dispondrá del resto de campos de búsqueda de animales.
- **QR:** Esta será la tabla encargada de guardar los códigos creados por el usuario. Tendrá dos campos principales: usuario y QR.
- **Vídeo:** Esta tabla es la que gestiona los vídeos creados en la aplicación y para poder organizarlo tendrá los campos: usuario que al igual que en los casos anteriores es la forma que tenemos de organizar la gestión de los vídeos, por tanto, también tendrá un campo con ellos.
Los campos que siguen a continuación fueron creados para establecer las características principales del animal en adopción y así a posteriori facilitar la búsqueda de un animal: nombre, texto, sexo, tipo y edad.
- **Localización:** En esta tabla se gestionará la ubicación de los animales perdidos: usuario, foto donde guardaremos la imagen correspondiente del animal en peligro que haya sido encontrado, latitud y longitud con lo que podremos encontrar al animal perdido.

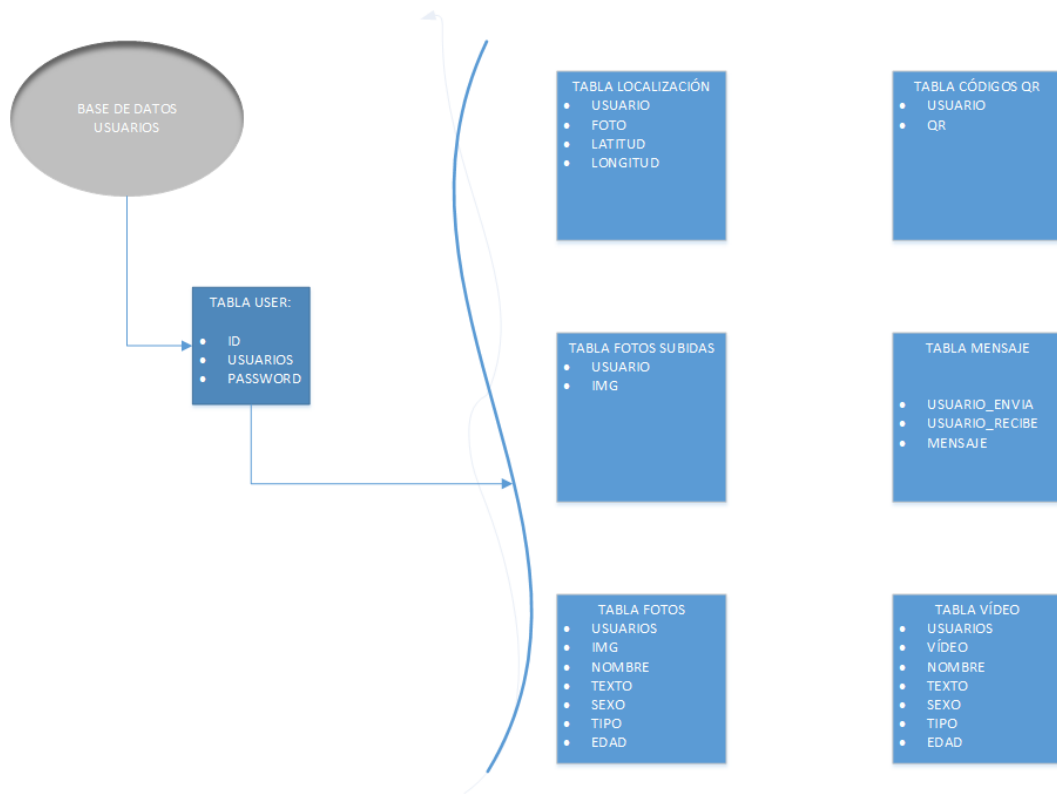


Figura 4.8. Esquema Base de datos.

4.1.1.2. Problemas con XAMPP y la base de datos

Durante el desarrollo del proyecto me encontré con varios problemas en lo que se refiere a la base de datos y el servidor web utilizado.

En primer lugar tuve problemas con la herramienta XAMPP en Windows, ya que en un principio lo instalé en un directorio y no en la raíz, esto provoca problemas a la hora de utilizarlo, pero tras buscar en Internet averigüé que bastaba con ejecutar `setupxampp.bat`; este script actualiza las rutas de los componentes de XAMPP a la ruta actual.

El segundo problema estaba relacionado con el uso de los puertos que usan los diferentes módulos de XAMPP. Por ejemplo el servidor Apache que trae XAMPP está configurado para utilizar puertos estándar, es decir el puerto 80 para HTTP y el 443 HTTPS, que puede ser que estén en uso, pero esto pasa también con el resto de módulos. Esto se ve en el panel de control al iniciarlo. Con el botón `netstat` se pueden ver todos los puertos que están en uso y el programa que los están usando. De esta forma me di cuenta, que ya estaba utilizando el puerto 80 con otro programa. Para poder arreglarlo existen tres posibles soluciones: detener el otro programa, cambiar su puerto, o cambiar el puerto en XAMPP.

Los dos archivos principales de configuración son los archivos `httpd.conf` y `httpd-`

ssl.conf (Apache). Para editarlos se puede utilizar el panel de control de XAMPP, que los abre directamente en el bloc de notas. En estos ficheros habrá que cambiar las líneas que tengan el puerto que queremos cambiar y modificar el número por el puerto actual. Pero si cambias los puertos de Apache para poder acceder a la pagina XAMPP, tendremos que poner también el puerto que hemos cambiado, por ejemplo localhost:8080.

También surgieron los problemas típicos del uso de cualquier base de datos, y es que al ir avanzando en el desarrollo de la aplicación se tuvo que ir modificando la estructura principal de las tablas y la gestión de las mismas, teniendo que modificar varias veces el código en función de los campos. Debido a esto era necesario entrar en el servidor de la aplicación XAMPP para borrar las tablas creadas, puesto que sino daba error al haber sido modificadas.

Por ejemplo esto me paso con la tabla Fotos, ya que tuve que realizar varios campos a lo largo del desarrollo de la aplicación. Al principio la idea era guardar tanto las fotos realizadas desde la aplicación como las fotos subidas en la misma tabla, pero posteriormente, me encontré con el problema de que las fotos subidas tenían que ser guardadas en una carpeta que llamarías uploads y lo que antes era el campo img ahora se trataría de una referencia a la carpeta lo cual originaba problemas a la hora de mostrarlas posteriormente. Tras varios cambios decidí que debía crear una tabla independiente de la de Fotos, puesto que a pesar de parecer bastante iguales, guardaban muchas diferencias en cuanto a la gestión de las mismas.

4.1.2. Servicio Web

Como servidor de la aplicación web se ha utilizado Node.js. Para poder seleccionar el servidor apropiado respondí a las preguntas que se encontraban en el capítulo *Uso de las Tecnologías* (la subsección 3.3.2).

- Al tratarse de una aplicación web en tiempo real será importante la concurrencia y los tiempos de respuesta.
- Como cualquier aplicación web tendrá una gran cantidad de usuarios activos.
- En este caso, según el uso que le vamos a dar se trata de un proyecto pequeño.

Como se explico en la sección dedicada a Node.js en el capítulo 2, este servidor ofrece altos niveles de *Concurrencia* y *Escalabilidad*; características apropiadas para el uso de esta aplicación web.

Node.js ofrece para Windows un instalador .msi que lo hará todo el mismo. Lo podemos encontrar en la sección de descargas del proyecto Node.js disponible tanto para 32 como para 64 bits. Además como *Framework* se ha utilizado Express, esto con el único objetivo de simplificar el proceso de desarrollo de la aplicación y así extender las posibilidades. Para poder utilizar Express fue necesario instalar el módulo correspondiente.

Una de las herramientas que provee Express es el uso de rutas, lo que nos permite vincular una ruta a un bloque de código mediante una función que lo recibe y un

Callback que se ejecuta cuando se recibe la petición.

Esta función sigue la arquitectura *REST* (**R**epresentational **S**tate **T**ransfer) ya que es más sencillo, es multiformato y más rápido. Los métodos HTTP son los que aparecen en la tabla 4.1:

Express además habilita la función de las sesiones. En el caso de este proyecto

Métodos HTTP	Características
GET	Pide una representación del recurso especificado
POST	Envía los datos en el cuerpo de la petición y así puedan ser procesados por el recurso identificado. Esto puede resultar en la creación de un nuevo recurso o de las actualizaciones de los recursos existentes o ambas cosas.
PUT	Sube, carga o realiza un upload de un recurso especificado (archivo), es el camino más eficiente para subir archivos a un servidor.
DELETE	Borra el recurso especificado

Tabla 4.1. Métodos HTTP

resultado de gran utilidad puesto que era necesario poder controlar cada una de las sesiones con el navegador, distinguirlos para no poder acceder desde un mismo dispositivo a dos cuentas diferentes. Esta función Express permitió resolver este problema creando una sesión con el nombre del usuario con el que se había logueado. Esta sesión quedará guardada, hasta que el usuario se desconecte.

4.1.3. Características del Servidor

El servidor tiene una serie de funcionalidades que es necesario explicar para comprender mejor el comportamiento del mismo dentro del proyecto, en primer lugar cada una de las comunicaciones con el cliente es realizada mediante *Sockets* que establecen la conexión y realizan la tarea específica con el servidor. Estas tareas se encargan desde guardar las fotos realizadas por el cliente hasta enviar todos los mensajes recibidos al usuario correspondiente. Además en el servidor tenemos todas las rutas encargadas de redirigir al cliente a la página correspondiente en función de la petición realizada.

En el servidor se llevarán a cabo por tanto, registrar al usuario correspondiente a través de un método POST recibido de parte del cliente; una vez obtenidos los parámetros (nombre de usuario, contraseña) se buscará en la base de datos ese usuario por si ya existiese un cliente con ese nombre, si es así se enviará el mensaje apropiado

para que el navegador pueda identificarlo, sino se registrará al usuario y se comunicará al cliente que ya puede loguearse.

También habrá que almacenar las fotos en la base de datos para poder mostrarlas siempre que el usuario quiera ver la galería de animales que necesitan ser adoptados. Estas fotos se guardarán con una serie de características extras como son el sexo del animal, el tipo, la edad y una breve historia. De forma que también se pueda seleccionar de la base de datos los animales con los requisitos que se desean para la adopción y enviarlo al cliente para que solo vea los animales que puede adoptar y que suplen las características que desea en función del formulario rellenado previamente. Será necesario registrar los mensajes enviados a un usuario con el nombre del que envía y a quien va dirigido para poder mostrarlo al cliente posteriormente. Además habrá que guardar las coordenadas latitud y longitud con la foto del animal que es preciso rescatar en la base de datos. Cuando sea necesario se mostrará al cliente todos los animales que están en apuros y que necesitan ser salvados, además de enviar al cliente las coordenadas para que éste, mediante geolocalización, muestre la ubicación exacta. Finalmente se podrá guardar e imprimir códigos QR en la base de datos.

A parte de estas funcionalidades hay dos que requieren una explicación mucho más concisa debido a la dificultad que provocaron a lo largo del proyecto y a que siguen una línea más dispar al resto de características. Estas son la de guardar los archivos subidos y los vídeos.

4.1.3.1. Uploads

Una de las características que tenía que disponer el proyecto era la de poder subir archivos desde el cliente. Para poder hacerlo es necesario hacer un formulario especial del lado de cliente que tenga en el cuerpo un simple formulario que lleva como atributo `action` la ruta `fileUpload`, como valor del atributo `method` `post` y el atributo `enctype=multipartform-data` necesario para enviar el archivo al servidor.

En el servidor habrá que definir el método `app.POST`, `app` es la variable que utilizará el modulo `Express` que ya explique anteriormente y que básicamente sirve para facilitar el desarrollo de cualquier página web. Una vez configurada la aplicación se hará lo mismo con las rutas; asociando cada una de ellas con una función encargada de controlar la acción a ejecutar. Esto es lo que hace `app.POST`, recoge el valor del formulario de la ruta específica y por tanto las imágenes subidas gracias a la función `multipart form-data`.


```

app.post('/fileUpload', login, function(req, res){
    var uploadedFile = req.files.uploadingFile;
    var tmpPath = uploadedFile.path;
    var targetPath = './views/uploads/' + uploadedFile.name;
    var b=tmpPath.split("\\");
    var l=b.length;
    var path= './uploads/' + uploadedFile.originalFilename;
    fs.rename(tmpPath, targetPath, function(err) {
        if (err) throw err;
        fs.unlink(tmpPath, function() {
            if (err) throw err;
            connection.query('INSERT INTO Fotos_Subidas
            (usuario, img) values(' + "'" + usuario
            + "'" + ',' + "'" + path + "'" + ' ');');
            res.send('File Uploaded to ' + targetPath
            + ' - ' + uploadedFile.size + ' bytes');
        });
    });
});

```

Como se muestra en el código, parseamos los ficheros que se han subido desde el navegador y lo definimos como variables para guardarlo posteriormente en una carpeta dentro de nuestro proyecto. Esta carpeta se llamará Uploads y es la encargada de guardar todas las imágenes subidas al servidor de esta manera.

La diferencia es que las fotos creadas con la nueva funcionalidad de HTML5 nos creaba imágenes que pasábamos de manera completa mediante un socket al servidor y, por tanto, esto era lo que guardábamos en la base de datos. Ahora al subir archivos cambia bastante el método, porque lo que se recibe es un objeto completo: esto es un problema a la hora de guardarlo en una base de datos, por la sencilla razón de que no se pueden guardar. Pensé en *Serializar* el objeto y así meterlo en la base de datos puesto que esto permitiría de una forma sencilla distinguir los archivos subidos en función del usuario que lo ha hecho. Por ello comencé a serializar el objeto recibido, pero cuando intentaba enviar al cliente la imagen después de haber deserializado el objeto observé que no funcionaba. Al mirar lo que recibía me dí cuenta de que el objeto había sido modificado, ya que todos los métodos desaparecían. Esto complicaba las cosas así que me decanté por guardar las imágenes en la carpeta y utilizar una base de datos en la que añadiría el path de esa imagen, relacionándola con el usuario dueño de esa imagen y de esta manera distinguir las fotos de cada una de ellas.

4.1.3.2. Vídeo

La realización del vídeo fue un problema que ya se explicará en las características del lado del cliente; pero por parte del servidor también hubo complicaciones. En un principio lo que guardaba era una *url blob* y al enviar el vídeo al cliente pensaba que guardaba en la base de datos el vídeo entero, pero como se menciona en la definición de este tipo de url, esta se refiere a los datos que tiene el navegador actualmente, por tanto cuando enviaba los vídeos al cliente para que pudiese verlos ninguno de estos vídeos podía visualizarse. Esto era extraño porque sin embargo ponía el mismo enlace en la url del navegador y si que podía verlo siempre y cuando no hubiese salido de la ventana donde creaba el vídeo (debido a lo explicado anteriormente). De esto me dí cuenta después por eso esto me supuso muchos problemas, porque no sabía si guardaba el vídeo incorrectamente o si era porque los creaba mal. No entendía por qué no podría verlos en mi propia aplicación pero si en una pestaña a parte. Posteriormente me dí cuenta de que si que podía ver el último vídeo creado, entonces tras buscar información entendí que los vídeos blob se guardan en un principio en una caché temporal que desaparece al momento que salía de la página donde creaba el vídeo. Esa era la razón por la que podía ver con esa url solo aquellos vídeos creados en ese momento. Estaba guardando mal los vídeos. En realidad directamente no los estaba guardando.

El método que estaba utilizando no funcionaba así que opté por cambiar de opción; incluso al crear el vídeo.

Los vídeos creados del lado del cliente son objetos y al intentar guardarlos en el servidor me pasaba lo mismo que con los archivos subidos; no podía guardar el objeto. Fue entonces cuando decidí que podría guardarlo también en una carpeta; como había hecho con upload.

```

function writeToDisk(dataURL, fileName) {
    var fileExtension = fileName.split('.').pop(),
        fileRootNameWithBase = './views/uploads/' + fileName,
        filePath = fileRootNameWithBase,
        fileID = 2,
        fileBuffer;
    while (fs.existsSync(filePath)) {
        filePath = fileRootNameWithBase + '(' + fileID +
            + fileExtension;
        fileID += 1;
    }
    dataURL = dataURL.split(',')[0];
    fileBuffer = new Buffer(dataURL, 'base64');
    fs.writeFileSync(filePath, fileBuffer);
}

function merge(socket, audioName, videoName){
    var isWin = !!process.platform.match(/win/);
    if (isWin){
        ifWin(socket, audioName, videoName);}else{
        ifMac(socket, audioName, videoName);}
}

function ifWin(socket, audioName, videoName) {
    var merger = __dirname + '\\views\\merger.bat';
    var audioFile = __dirname + '\\views\\uploads\\' + audioName;
    var videoFile = __dirname + '\\views\\uploads\\' + videoName;
    var mergedFile = __dirname + '\\views\\uploads\\' +
        audioName.split('.')[0] + '-merged.webm';
    var command = "C:\\ffmpeg\\bin\\ffmpeg.exe" + " " +
        "-i " + videoFile + " " + "-i " + audioFile + " " +
        + mergedFile + '';
    console.log(command);
    var cmd = exec(command, function(error){
        if (error){
            console.log('Error code: ' + error.code);
            console.log('Signal received: ' + error.signal);
        }else{
            socket.emit('merged', audioName.split('.')[0] +
                + '-merged.webm');
        }
    });
}

```

El método para conseguirlo queda descrito en el código superior. Primero escribíamos en el disco el vídeo y el audio; ya que se grababan por separado en el cliente. Guardamos ambas partes en nuestra carpeta y Node.js invoca ffmpeg para combinar wav / WebM en un solo archivo "WebM". Entonces le enviamos al cliente la dirección del archivo combinado.

4.2. Diseño del Cliente

Tras el diseño e implementación del servidor haremos lo mismo con el cliente de nuestra aplicación web. El cliente está diseñado en HTML5 y CSS3 para permitir un diseño sencillo y moderno, además tiene código en Javascript para la generación de pequeños script.

4.2.1. Características del Cliente

El usuario que decida utilizar la aplicación web tendrá a su disposición diferentes funcionalidades que siguen la línea de la página de El Refugio. En esta idea y teniendo en cuenta las facilidades y estética que aporta HTML5 y CSS3 se presentan las funcionalidades.

Primeramente, el usuario deberá identificarse. Una vez logueado el usuario se redirige a una página de menú; en ella se especifican cada una de las actividades ofrecidas por el cliente (figura 4.9). Como puede observarse en la imagen, presenta un aspecto visual sencillo pero bastante atractivo.



Figura 4.9. Menu.

Además el menú presenta las siguientes características.

PERFIL: El perfil de usuario (figura 4.10) tiene tres botones. Uno dedicado a la comunicación que permite enviar mensajes a los usuarios que están registrados en la aplicación, el segundo botón nos permite ver los mensajes que tengamos en el buzón de mensajes y finalmente podemos ver las fotos que ha subido dicho usuario.

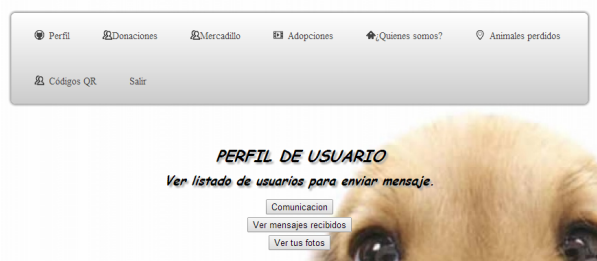
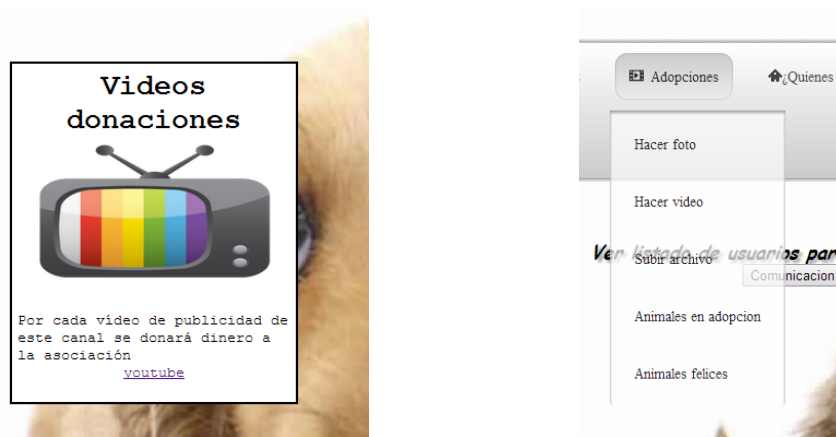


Figura 4.10. Perfil.



a) Donaciones

b) Función adopciones

Figura 4.11. Figura de los donativos y la función adopciones

DONACIONES: Se han añadido algunos detalles heredados de la página real. En este de donaciones tenemos un enlace a los anuncios de Youtube; con los que se les ayuda a la asociación a recaudar fondos de una manera sencilla mediante la visualización de los vídeos, con los que se le aportaría a la asociación unos pequeños donativos (figura a) de la imagen 4.11).

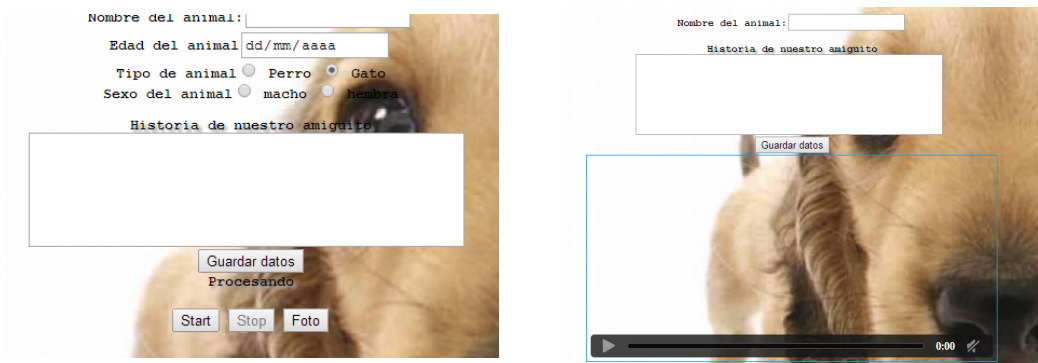
ADOPCIONES: En esta función tenemos diferentes pestañas: hacer foto, hacer vídeo, subir archivo, animales en adopción y animales felices (figura b) de la imagen 4.11).

SUBIR ARCHIVO: Esta opción nos permite subir archivos al servidor, ya sean fotos o vídeos que estén guardados en cualquier dispositivos desde el que se acceda a la aplicación.

ANIMALES EN ADOPCIÓN: Galería de los animales que pueden ser adoptados, con la posibilidad de mandar un mensaje al usuario que ha subido la foto para ponerse en contacto con él.

HACER FOTO: En esta opción se hará la foto del animal que se desea ofrecer en adopción. Además se rellenará un formulario para que luego la búsqueda del animal sea más sencilla y rápida (figura a) de la imagen 4.12).

HACER VÍDEO: Al igual que en el caso anterior de nuevo existe un formulario y la posibilidad de hacer un vídeo de aquellos animales que ya hayan sido adoptados (figura b) de la imagen 4.12).



a) Función hacer fotos

b) Función hacer vídeo

Figura 4.12. Creación de foto y vídeo

ANIMALES FELICES: Galería de vídeos de los animales que ya han sido adoptados, con la posibilidad de enviar mensajes para preguntar cualquier duda al usuario que ha adoptado (figura 4.13).

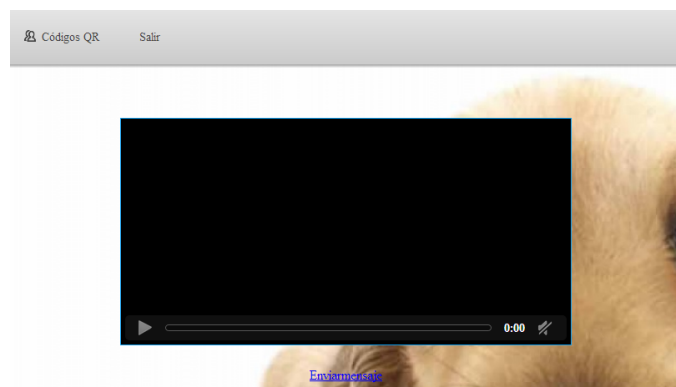


Figura 4.13. Galería vídeos.

¿QUIENES SOMOS?: Esta pestaña presenta información sobre la asociación: cómo adoptar, búsqueda y ¿Quieres ayudar?.

BUSCAR: Esta opción tiene un formulario a rellenar por la persona que desee realizar una nueva adopción. De esta manera, tras buscar en la base de datos en función de los requisitos, se ofrecerá al usuario las imágenes de los animales que presentan esas cualidades.

CÓMO ADOPTAR: En esta opción se presentan los requisitos necesarios para adoptar un nuevo animal. Se ha logrado una imagen mucho más moderna y sencilla utilizando herramientas de CSS3 y HTML5 (figura 4.14).



Figura 4.14. Requisitos adopción.

¿QUIERES AYUDAR?: Esta pestaña ofrece a su vez tres nuevas opciones: socio, simpatizante o voluntario.

SOCIO: Presenta la información conveniente para hacerse socio de la asociación (figura 4.15).

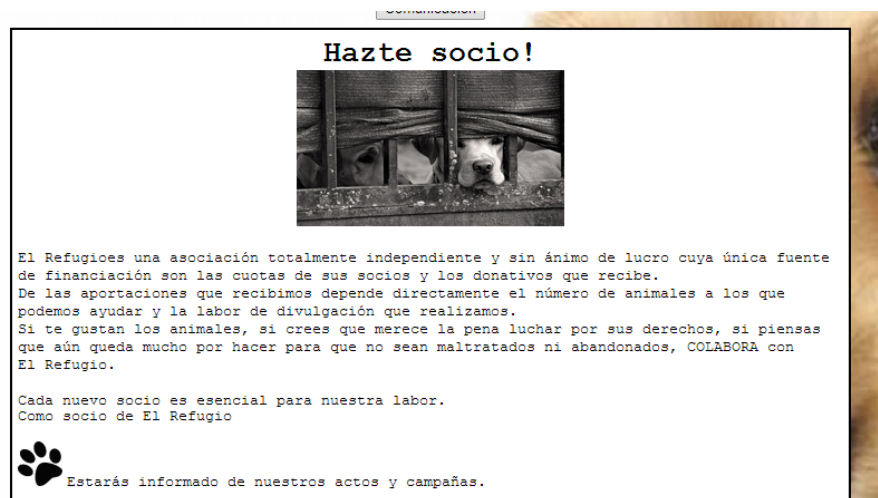


Figura 4.15. Información socio.

SIMPATIZANTE: Presenta la información conveniente para hacerse simpatizante de la asociación (figura 4.16).



Figura 4.16. Información simpatizante.

VOLUNTARIO: Presenta la información conveniente para hacerse voluntario de la asociación (figura 4.17).

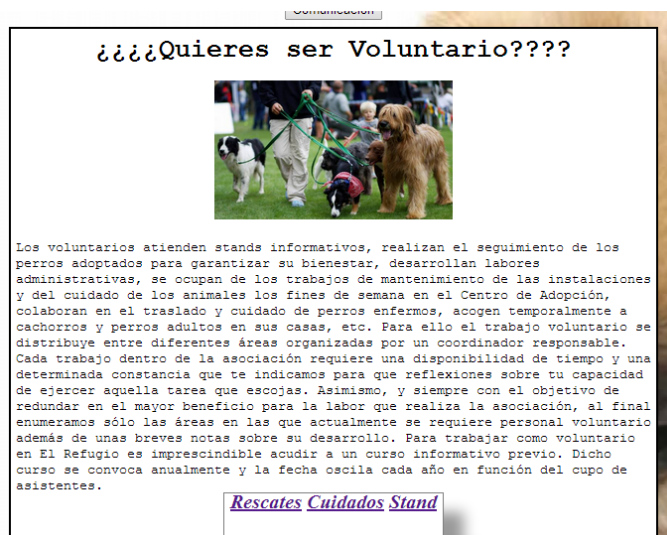


Figura 4.17. Información Voluntario.

Ademas tenemos tres tipos de ayudas como voluntario que son las que se muestran en las siguientes imágenes.

En la figura 4.18 se da instrucciones a cerca de ayudas para el rescate de animales.

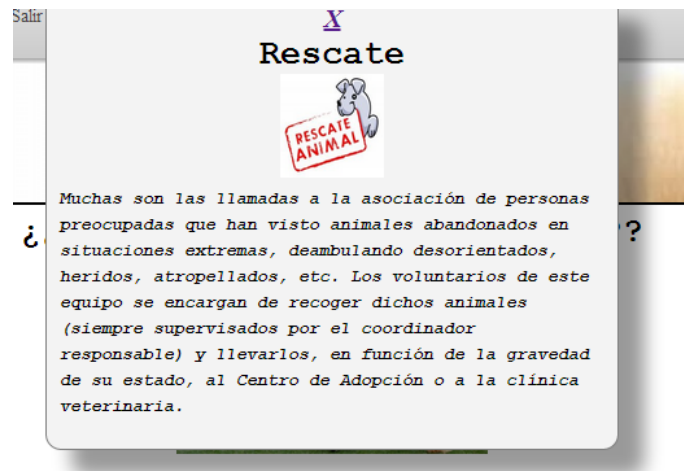


Figura 4.18. Información Voluntario 1.

En la figura 4.19 se ofrece la posibilidad de ayudar con los cuidados diarios que requieren

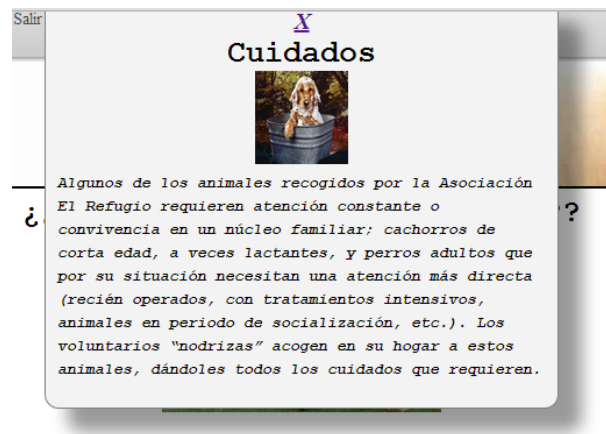


Figura 4.19. Información Voluntario 2.

Finalmente la figura 4.20 permite la posibilidad de ayudar divulgando información.

ANIMALES PERDIDOS: Esta opción tiene: Foto con localización y el rescate.

FOTO CON LOCALIZACIÓN: Nos permitirá subir una foto de algún animal que necesite ayuda: o bien que haya sido abandonado o que este herido. Junto con la foto se guardará la localización; de esta manera la gente que este cerca o voluntarios podrán ir a ayudar a ese lugar.

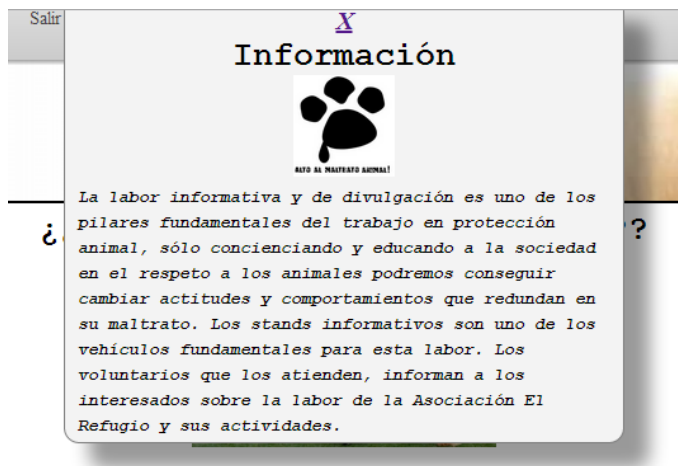


Figura 4.20. Información Voluntario 3.

AL RESCATE!: Galería con los animales en apuros, así como la localización (figura 4.21).

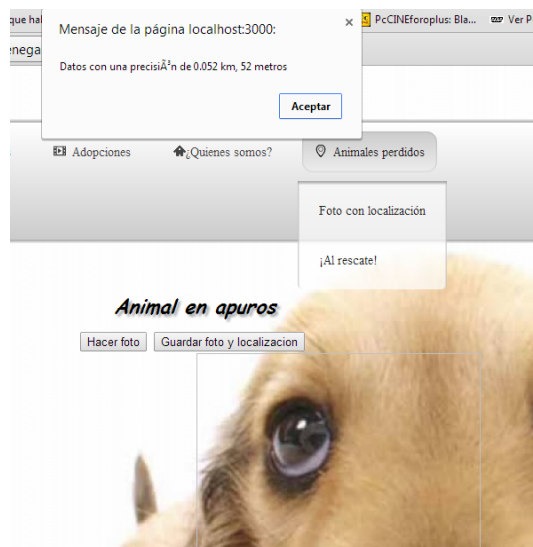


Figura 4.21. Al rescate.

CÓDIGOS QR: Se compone de un Lector QR y un Crear QR.

LECTOR QR: Esta opción presenta una cámara que captará en todo momento las imágenes hasta detectar un código QR, mostrando en pantalla la información del mismo (imagen b de la figura 4.22).

CREAR QR: Esta opción tiene un formulario con el que se guarda el nombre de un usuario o una página web; creando el código QR correspondiente. De esta forma en el caso de que un usuario haya perdido un animal podrá poner carteles informativos con el código QR correspondiente para que accedan a la web y se puedan comunicar

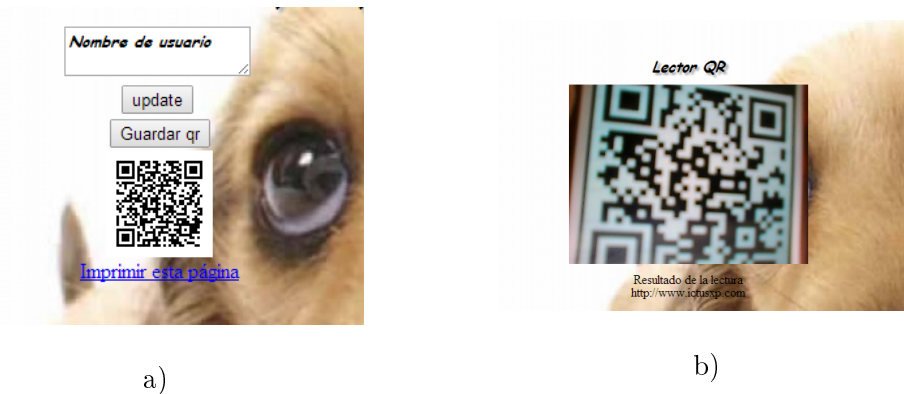


Figura 4.22. Figura de la imagen

con el dueño en caso de tener información sobre el animal. Además presenta la opción de poder imprimirlo una vez haya sido generado (imagen a de la figura 4.22).

4.2.2. Desarrollo e implementación características

En este punto de la memoria se explicarán el desarrollo de las funciones más características del cliente, las librerías utilizadas y las complicaciones que generaron durante su implementación. Las actividades han sido resumidas en el punto anterior.

4.2.2.1. Crear y leer códigos QR

Para añadir una serie de mejoras al proyecto se pensó el incorporar la creación y lectura de códigos QR. Se trataría de que aquellos usuarios que hayan perdido uno de sus animales pudiesen crear códigos QR con el fin de poder añadirlos en fotos de búsqueda. De manera que si un usuario tiene información a cerca del animal pueda leer el código QR y acceder al perfil del usuario para poder ponerse en contacto con él.

Para la creación de códigos QR se ha utilizado la librería qrcode.js y jquery.qrcode.js. Para la lectura del código se utilizo la librería html5-qrcode.min.js. Para poder generar el código a partir de los datos obtenidos del formulario se utiliza la función createqrcode, donde le pasamos el texto con el que se creará el qr correspondiente a esa información a partir de las diferentes librerías utilizadas. Finalmente para guardarlo se utiliza la función updateqr, pero además se ha añadido la opción de imprimir ese código.

4.2.2.2. Crear Fotos con Canvas

Para poder hacer fotos desde el navegador resulta necesario empezar por usar navigator.getUserMedia(), lo que nos permite conectarnos a la entrada del micrófono y de la cámara web sin necesidad de ningún complemento. Ahora se puede acceder a la cámara con solo realizar una llamada, sin tener que instalar nada. Los datos se procesan y se envían directamente al navegador. La detección de funciones es simplemente una comprobación de la existencia de navigator.getUserMedia.

Para utilizar una cámara web o un micrófono, necesitamos solicitar permiso. El

primer parámetro de `getUserMedia` es para especificar el tipo de medio al que se quiere acceder. Por ejemplo, si quieres solicitar la cámara web, el primer parámetro debe ser vídeo. Para usar tanto el micrófono como la cámara, se utiliza vídeo, audio. Mediante esta función utilizamos la cámara desde el navegador, siempre que antes se haya aceptado el acceso.

El elemento `canvas` se define como un entorno para crear imágenes dinámicas. El propio elemento es bastante simple. Lo único que hay que especificar al usarlo son sus dimensiones. Luego dispone de métodos realizados mediante javascript que permiten realizar cosas como crear fotos mediante el vídeo `getUserMedia`; esto se ha conseguido a través del método `drawImage(vídeo, 0, 0)` del API de `<canvas>`, lo que hace que resulte muy fácil dibujar fotogramas de `<video>` en `<canvas>`. Por supuesto, tras obtener la entrada de vídeo a través de `getUserMedia()` que se explico anteriormente. De esta forma siempre que pulsemos la opción *hacer foto* recogeremos los datos de la cámara y mediante `canvas` y el método `drawImage` podremos obtener la imagen correspondiente.

El mayor problema que me encontré con el uso de `canvas` fue a la hora de dimensionar la imagen y el vídeo puesto que no conseguía cuadrar bien las fotos, además fue necesario buscar mucha información para poder realizar esta funcionalidad puesto que no conocía nada de esta nueva característica que implementa HTML5.

4.2.2.3. Realización de vídeos

Probablemente la realización de vídeos haya sido una de las funcionalidades que más problemas me ha generado. Esto es debido a que a pesar de que HTML5 lleve ya algún tiempo, todavía surgen cierto problemas a la hora de crear audio y vídeo junto; sobre todo, para poder guardarlo en el servidor. En un principio comencé por generar vídeos sin voz puesto que no podía hacerlo con la librería que había seleccionado en ese momento y tras haber buscado mucho era esta la mejor de las opciones. Esta librería creaba vídeos con una url blob y lo guardaba de manera local en el navegador. Yo pensaba que estaba guardando el vídeo en el servidor pero en realidad el vídeo se perdía una vez cargada la página para realizar vídeos de nuevo; tras darle muchas vueltas e intentar en vano solucionar el problema decidí cambiar a otra librería que me permitían además crear vídeos con audio. Esto fue bastante complicado porque HTML5 fallaba bastante con el uso de otras librerías. Hasta que después de buscar muchísima información encontré lo más apropiado para mi aplicación.

La librería utilizada `RecordRTC.js` y `recorder.js` permitía grabar audio y vídeo por separado como wav/webm. Posteriormente enviaba los ficheros al servidor a través de `socket.io`, de manera que el servidor recibía los ficheros. Otro de los mayores problemas era el guardarlo en el servidor; así que opté por hacerlo de la misma forma que hacía con la subida de archivos; en una carpeta a parte, de manera que en la base de datos solo hubiese que relacionar el nombre del archivo creado en la carpeta con el del usuario que lo había creado para poder mostrarlo posteriormente.

En el navegador se creaban los vídeos con audio, pero en el servidor era necesario guardarlos a parte y ejecutar desde el propio código una llamada a ffmpeg para poder unir en un solo elemento el audio y el vídeo; convirtiendo ambos ficheros en uno solo webm. Finalmente, mediante socket.io le volvemos a enviar al cliente la url del fichero creado para que muestre el resultado final en ese momento, además como ya expliqué, se guarda en la base de datos la ubicación exacta del vídeo con el fin de poder mostrarlo siempre que quieran los usuarios.

4.2.2.4. Geolocalización

En el desarrollo de la aplicación era necesario incorporar la geolocalización, puesto que una de las funcionalidades de la web iba a ser la de poder incorporar fotos con la localización exacta de un animal perdido. Ambas cosas serían registradas en una base de datos, de esta manera las asociaciones o personas próximas al lugar podrían ir a ayudar al animal en apuros; incluyendo además la posibilidad de dibujar una ruta desde el destino de la persona encargada de ayudar al animal hasta el mismo.

Para poder utilizar la geolocalización fue necesario añadir las librerías `http://j.maxmind.com/app/geoip.js` y `http://maps.google.com/maps/api/js?sensor=false`. Al igual que ocurre con GetUserMedia para utilizar geolocalización tendremos que solicitar permiso al navegador.

Una vez aceptada el acceso a geolocalización se recogen las coordenadas y se muestra el mapa con la localización exacta.

Además el proyecto incorpora la funcionalidad de trazar una ruta desde tu posición actual hasta la posición en la que se encuentra el animal (figura 4.23). Esto se realiza guardando primeramente las coordenadas latitud y longitud y calculando posteriormente la ruta mediante el método DirectionsService.

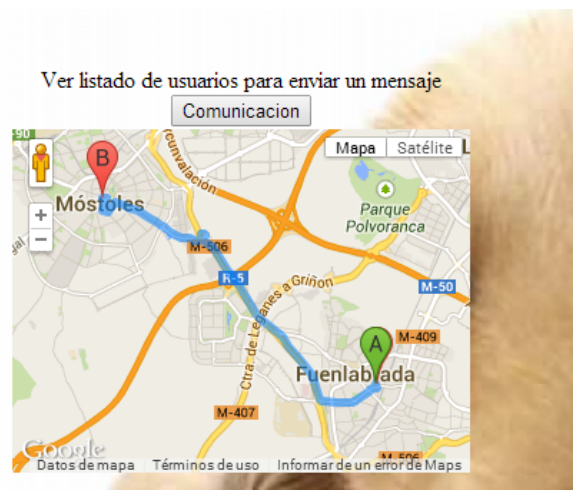


Figura 4.23. Mostrar ruta.

CAPÍTULO 5

RESULTADOS

En el siguiente capítulo se hará un seguimiento del origen de la aplicación así como de una serie de pruebas que se han llevado a cabo entre diferentes usuarios, tanto cercanos a las asociaciones para poder comprobar la utilidad del mismo, como de usuarios externos para ser conscientes de la utilidad en lo que a las adopciones se refiere.

5.1. Origen de la Aplicación

En la siguiente sección se explicará brevemente el contacto que tuve con las asociaciones para poder llevar en práctica la idea principal del proyecto.

En un principio mi idea estaba más enfocada en desarrollar una aplicación que sirviera de ayuda a las perreras. Por ello comencé por visitar varias como la de Móstoles. Hablé con algunas de las personas que trabajaban allí, pero en un principio, nadie supo darme opciones ni exponerme sus necesidades, más bien me daban la idea de ir a preguntar a Asociaciones que ya disponían de página web con la que podría orientarme. Fue por ello que me acordé de la asociación **PROA** en la que conozco voluntarios que trabajan allí. También conocía la Asociación de **El Refugio**, ya que disponen de una página web en la que está basada la aplicación, y que me ha servido como referencia a lo largo del proyecto.

Mantuve varias charlas con voluntarios de ambos; en ellas me explicaron las mayores necesidades de las que carecen estas asociaciones y como podría ayudar en su implementación en forma de aplicación, así como posibilidad de orientarme con la página oficial del **Refugio**.

Posteriormente le expliqué esto a Gregorio, que tras dar la confirmación me permitió comenzar el proyecto. Una vez dado el visto bueno comencé el desarrollo de la aplicación para posteriormente realizar las pruebas que se mostrarán a continuación.

5.2. Pruebas realizadas

A medida que se iban realizando algunas de las funcionalidades más importantes, fueron realizándose pruebas. Para ello me reunía con gente de las asociaciones, de este modo comencé por realizar una prueba general sobre las siguientes funcionalidades, las cuales no estaban en la página web original.

5.2.1. Prueba de foto

Comenzamos probando esta característica de la aplicación, ya que resultaba muy importante para poder llevar a cabo las adopciones de los animales, puesto que de las galerías de fotos se seleccionarían las mascotas.

En la primera de las pruebas realizadas, algunos voluntarios me propusieron incluir un formulario con algunas de las características básicas del animal, ya que únicamente por la foto no sería suficiente para poder buscar en función de las necesidades del adoptante, además al mostrar la galería únicamente se podrían visualizar las fotos de los animales más recientes, lo que no facilitaría a la asociación la adopción de forma adecuada.

Conforme a ello decidí establecer un formulario que habría que rellenar al realizar la foto. De esta manera también aquellos usuarios que quisieran adoptar tendrían que rellenar un formulario inicial que serviría de buscador en la base de datos.

5.2.2. Prueba de geolocalización

En un principio esta funcionalidad solo mostraba la ubicación de algunos de los animales que se encontraban en peligro, sin embargo no ofrecía la posibilidad de relacionar esta posición con una foto realizada al animal, lo cual podría ayudar a aquellos usuarios que pudiesen ir a auxiliarlos. Este cometido fue propuesto también por varios de los usuarios.

Además, en relación con esta característica, se ofreció el poder obtener la ubicación del usuario dispuesto a cooperar, de esta manera, una vez obtenido el emplazamiento, y seleccionado el animal que resultase más fácil de salvar, se podría además calcular una ruta desde el origen del usuario al destino del animal.

5.2.3. Prueba de vídeo

La siguiente prueba consistía en probar la función de los vídeos, los cuales no solo valdrían para conocer animales en posible adopción, sino que también serviría para observar vídeos de animales que ya han sido adoptados. Como se mencionó en el capítulo 4 la inclusión de los mismos dio bastantes problemas, y por ello en un principio no se conseguía tener un vídeo con audio. Sin embargo, para que la página resultase más atractiva, y a petición de los usuarios fue necesario buscar una solución. Así que a pesar de las dificultades este objetivo se convirtió en primordial.

5.2.4. Creación y lectura de códigos QR

La opción de los códigos QR gustó bastante. Sobre todo la funcionalidad que se le había dado, es decir, la de lograr que si un animal ha desaparecido, y se tienen noticias a cerca del mismo, a partir de la lectura del código QR sea mucho más sencillo ponerse en contacto con el dueño de la mascota.

Junto a Gregorio llegamos a la conclusión de que sería apropiado poner la opción de imprimir el código QR. De esta manera podrían ponerse carteles con el código QR impreso.

5.2.5. Subir archivos

Tras realizar varias pruebas, y a partir de comentarios, pensé que sería bueno dar esta opción, con la que se podrían subir archivos desde el dispositivo.

5.3. Opiniones finales

Una vez incluidas todas las aportaciones por parte de los usuarios y voluntarios, y tras haberse realizado pruebas con diferentes dispositivos se decidió probar la aplicación de nuevo con aquellos usuarios que lo habían hecho inicialmente. Se realizaron una serie de estadísticas en función de la utilidad, la sencillez y las necesidades cumplidas.

A partir de la encuesta se creó esta tabla 5.1 en la que aparecen una serie de notas a cada una de las características de la herramienta. Estas puntuaciones se corresponden con las de diez usuarios que utilizaron la aplicación y que además ofrecerían más tarde un breve comentario a cerca de la misma. Estos usuarios van desde voluntarios, hasta adoptantes y usuarios.

De los datos obtenidos a partir de las preguntas se crearon dos gráficos para mostrar los resultados de una forma más sencilla y dinámica (gráficos a y b de la figura 5.2). Como podrá observarse la característica más relevante ha sido la de la utilidad y la menos valorada ha sido la de las necesidades cubiertas.

Usuarios	Utilidad	Sencillez	Necesidad Cubierta
Blanca (adoptante)	9	8	8
Fernando(usuario)	8	8	7
David(usuario)	9	8	6
Leticia(voluntaria)	9	9	7
Pedro(usuario)	8	7	8
Patricia(voluntario)	10	7	7
Alberto(adoptante)	8	7	6
Martín(voluntario)	10	8	7
Clara(voluntario)	10	8	8
Almudena(adoptante)	9	7	10

Figura 5.1. Tabla usuarios.

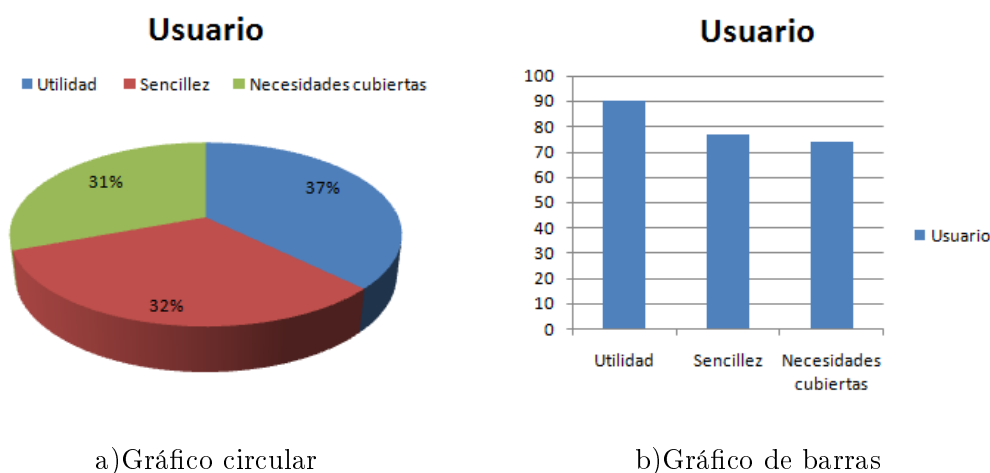


Figura 5.2. Gráfico circular y de barras

En la figura superior aprecia que se ha logrado la sencillez, que era uno de los requisitos finales. Lo menos puntuado han sido las necesidades cubiertas, todos los usuarios coincidían en que el resultado final era muy útil pero que a largo plazo podrían añadirse detalles que cubriesen aún más las necesidades de las asociaciones, que sin embargo si que han sido satisfechas primeramente con las funcionalidades incluidas.

En la tabla siguiente 5.1, se muestra las opiniones concretas de cada usuario a cerca de la aplicación desarrollada, como puede observarse tras ver las opiniones de cada una de los usuarios, están bastante contentos con la aplicación y con las nuevas funcionalidades introducidas.

Usuario	Opinión
Blanca adoptante	Me hubiese gustado que existiese una aplicación de este tipo cuando me decidí por adoptar una mascota. Probablemente hubiese conocido antes a Turbo. Muy sencilla además.
Fernando	Me parece una aplicación muy dinámica y creo que podría cubrir parte de las necesidades de cualquier asociación.
David	Aplicación muy útil para aquellos que quieran buscar una mascota
Leticia voluntaria	A pesar de mi breve experiencia como voluntaria, creo que es muy útil porque es una manera sencilla de acceder a mascotas que necesitan un hogar. Además a pesar de que no puede cubrir todas las necesidades de las asociaciones, si que resulta muy útil.
Pedro usuario	Nunca me he decidido por adoptar pero creo que puede dar muchas facilidades.
Patricia voluntaria	A parte de la adopción la opción de geolocalización puede sernos muy útil además me ha resultado super sencilla.
Alberto adoptante	Podría ser muy útil para adoptar, me gusta el formulario.
Martín voluntario	Nos han gustado mucho las nuevas funcionalidades, pero además, sí que consideramos que se le podrían añadir más características.
Clara voluntario	A mi me ha parecido que podría facilitar mucho las adopciones, además es muy fácil de usar.
Almudena adoptante	Me hubiese gustado tenerla cuando quería buscar a mi gato Lino.

Tabla 5.1. Opiniones

CAPÍTULO 6

CONCLUSIONES

Este capítulo es la culminación final del proyecto y como consecuencia de ello en él se valorará la realización de cada uno de los hitos expuestos al principio de la memoria así como algunas líneas futuras, constituyendo un resumen de todo el trabajo realizado. Finalmente se explicará las conclusiones obtenidas a partir del trabajo realizado.

6.1. Análisis del trabajo realizado

Al inicio de la memoria del proyecto se explicaban cada uno de los hitos que constituían el proyecto y el orden que se establecía para poder realizarlo (sección 1.3), por tanto los objetivos cumplidos son los siguientes:

- Se ha logrado una aplicación sencilla e intuitiva que favorece el contacto del usuario con la aplicación y por tanto favorece el trabajo de las asociaciones.
- Se han cumplido las necesidades y expectativas de las organizaciones.
- Acorde a las características y necesidades de la aplicación se eligió como servidor Node.js y con respecto al servidor de la base de datos XAMPP.
- Las funcionalidades añadidas y el entorno agradable de la aplicación facilitan, y por consiguiente aumentan el número de adopción de mascotas.
- La utilización de geolocalización y el cálculo de rutas favorecen el rescate de animales en peligro, así como el uso de códigos QR que ayuda a poner en contacto con el dueño del animal desaparecido a los usuarios que tengan noticias.

Puesto que la idea del proyecto no era ninguna ofrecida por el profesor y por tanto no se trataba de un problema real el primero de los objetivos fue el de darle forma al proyecto; para ello decidí hablar con muchas personas que estaban relacionadas de forma directa con asociaciones de animales para informarme de las necesidades principales que pudiesen tener y cómo podría relacionarlas con el proyecto (en la sección 5.1 se encuentra información a cerca de ello). Posteriormente le aporte las ideas a Gregorio, para que diese por válido, o no, cada uno de los objetivos: puesto que no solo la idea debía ser de ayuda a las asociaciones, sino que además, debía incluir elementos

tecnológicos novedosos. De esta manera se llegó a un cúmulo de ideas donde se pudiese lograr cierta reciprocidad entre las necesidades primarias de las asociaciones y el uso de nuevas herramientas tecnológicas.

Tras darle forma al proyecto era necesario elegir de manera correcta las herramientas: el lado del cliente no resultó problemático, puesto que se decidió rápidamente el utilizar HTML5 y CSS3, sin embargo la elección del servidor me costó un poco más hasta que finalmente me decanté por Node.js; herramienta que no había utilizado nunca y que tuve que aprender a usar, así como aprender más en profundidad el lenguaje de Javascript. Además a medida que avanzaba con el proyecto tuve que tomar decisiones a la hora de elegir una herramienta adecuada para el uso de la base de datos: esto también me llevó su tiempo, pero finalmente me decanté por XAMPP y como en el caso anterior tuve que leer tutoriales a cerca de su uso.

Una vez elegidas las herramientas me estructuré un orden para la realización de cada uno de los objetivos del proyecto. Este orden al final tuvo que verse alterado porque hubo hitos del proyecto que me dieron problemas y en los que decidí no quedarme estancada, para ir avanzando. Tal y como se explica en los objetivos mencionado arriba se fue estudiando cada una de las metas y su realización, a medida que iba aprendiendo el lenguaje.

En un principio todo el proyecto fue realizado con el ordenador al igual que cada una de las pruebas, pero quise darle más versatilidad y para ello decidí probar el uso de todo lo realizado en otros dispositivos como móvil y tablet. Tuve que realizar algún cambio, pero no resultaron muy problemáticos porque la aplicación funcionaba en la gran mayoría de los objetivos en cada uno de los dispositivos.

Además para asegurarme del uso en otros dispositivos y de la impresión que pudiese causar en otras personas, le hice probar la aplicación a bastante personas de mi entorno, incluidas algunas de las personas que conozco relacionadas con Asociaciones de animales. En ninguno de los casos dio problemas el uso de la aplicación, además todos coincidieron en que resultaba una herramienta muy sencilla e intuitiva y que podría servir de utilidad.

6.2. Aprendizaje

La realización de este proyecto, *Diseño e implementación de una aplicación web para la gestión y desarrollo de asociaciones de adopción de animales*, ha supuesto una culminación en mi aprendizaje a lo largo de la carrera. Con la realización de este proyecto he podido aplicar parte de los conocimientos inculcados durante mis estudios universitarios.

La realización de la aplicación me ha permitido aprender mucho más a fondo a cerca de lenguajes como HTML que ya había utilizado en otras asignaturas de la carrera; sin embargo, en este caso tuvo que aprender sobre la última versión de este lenguaje, HTML5, y sus nuevas funcionalidades como el uso de canvas o el poder

hacer fotos y vídeos desde el propio navegador. Mis conocimientos sobre MySQL eran buenos porque los había obtenido a lo largo de la carrera, sin embargo, al igual que me ocurrió con CSS hacía un tiempo que no los utilizaba y estaban bastante oxidados, por lo que tuve que recordar muchas de las características de ambos. En concreto, en el caso de CSS, este recordatorio me sirvió como referencia para poder profundizar mucho más en la nueva versión del lenguaje, CSS3, que sería el que utilizaría en la aplicación.

Para poder llevar a cabo el proyecto fue necesario la elección de un servidor, en este caso Node.js. Esto me supuso tener que aprender a fondo el lenguaje Javascript, del que sólo tenía ligeras nociones que había aprendido al tener que usarlas de una forma muy básica, agregando interactividad a páginas que tuve que realizar en asignaturas a lo largo de la carrera. Además tuve que conocer a fondo todos los módulos necesarios en el desarrollo de una aplicación con el servidor Node.js.

Además utilicé un servidor externo para la base de datos, XAMPP. Esto también provocó que me tuviera que documentar a cerca del uso del mismo, de su instalación y de como podía revisar los cambios provocados en la base de datos, así como poder acceder a cada una de las tablas creadas para ir revisando los cambios.

Por tanto la realización del proyecto me ha permitido ampliar mis conocimientos aprendidos en la carrera, así como formarme en nuevas tecnologías y lenguajes.

6.3. Líneas futuras

En esta sección se incorporan posibles líneas futuras que permitan un mayor desarrollo del proyecto consiguiendo mejoras en el mismo. Una de las posibles funciones futuras podría ser la incorporación de un chat que permitiese a los usuarios conectados mantener conversaciones de manera instantánea y no solo a través de los mensajes que ya tiene incorporado el proyecto.

Además en un futuro la aplicación pudiese extenderse para otro tipo de animales en extinción o salvajes; de forma que la pudiesen tener todos los trabajadores de parques naturales para mantener el control de los animales y poder captar a otros que estén heridos.

Para conseguir que la aplicación pueda ser mucho más provechosa podría realizarse una aplicación nativa para *Android* o *iPhone*; de esta manera todas las notificaciones las podrías recibir de manera instantánea en un tu móvil, lo que facilitaría el trabajo de las asociaciones. Ya que, por ejemplo, podrías recibir en tiempo real las notificaciones de los animales nuevos disponibles en adopción.

En esta línea se podría además configurar una especie de aviso de manera que si estas cerca de un animal en peligro te salte una alarma en el móvil, con la ubicación exacta y la foto del animal, así se podrá ir en su ayuda sin necesidad de tener que buscar tu los animales que puedan estar en peligro.

Convertir la aplicación web a nativa podría hacerse con Phonegap que permite empaquetar tu código como aplicación para distintas plataformas: *iOS*, *Android*, *Blackberry*, *Windows Phone*...

6.4. Valoración final

Tras haber terminado el proyecto y la realización de la memoria; se detallará una valoración final del trabajo realizado. Una vez terminado el proyecto y tras realizar las pruebas del mismo y enseñárselo a personas que podrían estar interesadas en su uso he llegado a la conclusión de que he llegado a los objetivos primordiales del proyecto aunque no me hubiese importado añadir más funcionalidades al proyecto. Las personas con las que pude hablar tras probar la aplicación me hablaron de que les había parecido intuitiva y sencilla; logrando por tanto uno de los objetivos primarios. Además todo funciona correctamente. Por tanto puede decirse que a nivel de trabajo este ha sido realizado con éxito.

A nivel personal la aplicación ha logrado ser de utilidad para las asociaciones, ayudando a una gestión de las mismas así como a la adopción de animales y a la protección de los mismos; consiguiendo lograrlo mediante nuevas tecnologías. Los usuarios que la han utilizado han estado de acuerdo en la sencillez y la utilidad de la aplicación.

Ha sido un trabajo complicado y largo; en el que además he aprendido muchísimo a cerca de herramientas y lenguajes que no conocía. Esto unido al objetivo final relacionado con la ayuda de los animales ha provocado que a pesar del trabajo y esfuerzo haya sido una tarea muy satisfactoria a nivel personal.

APÉNDICES

APÉNDICE **A**

PUBLICACIÓN DEL CÓDIGO

El código desarrollado para la creación y lectura de códigos QR se encuentra en el repositorio de GitHub:

<http://dwa012.github.io/html5-qrcode/>

El código desarrollado para la creación de vídeos se encuentra en el repositorio de GitHub:

<https://github.com/muaz-khan/WebRTC-Experiment/tree/master/RecordRTC/RecordRTC-over-Socketio/uploads/>

Las APIs utilizada para la geolocalización:

<http://j.maxmind.com/app/geoip.js>

<http://maps.google.com/maps/api/js?sensor=false>

Esta memoria se encuentra en:

<http://gsyc.urjc.es/~grex/pfcs/2014-rocio-quinones>

APÉNDICE B

PLANIFICACIÓN DEL PROYECTO

En esta sección de la memoria se detallará el tiempo realizado en el proyecto; de esta manera quedará descrita el tiempo que ha sido necesario para poder realizar la aplicación y por tanto el volumen de trabajo que ha sido necesario emplear para poder llevarlo a cabo. Es necesario explicar además que a medida que realizaba el proyecto tuve que terminar las tres asignaturas que me quedaban, así como trabajar en una beca. Todo ello ha supuesto un enorme trabajo y dedicación que sin embargo también han provocado mayor satisfacción al haber logrado terminar en el tiempo estimado; aunque para ello fuera necesario emplear un gran esfuerzo y dedicación.

A continuación se expondrá el tiempo estimado para cada uno de los objetivos:

B.1. Elección del proyecto

Fue en junio de 2013, una vez terminado el curso, cuando hablé con Gregorio sobre la idea de realizar el proyecto. En un principio se decidió realizar uno de los proyectos que él ofrecía y la idea era dedicarme a ello durante el verano. Le dí varias vueltas pero en realidad no era una idea que me llenase del todo. Además a la hora de utilizar la maquina virtual de la Universidad me dio muchos problemas, todo ello me retraso bastante y me desmotivó mucho. Fue por ello que comencé a cavilar la idea de cambiar de proyecto a algo que me motivase mucho más y que mejor que relacionar mi trabajo fin de carrera con una de mis aficiones. Le di bastantes vueltas a ello y hablé con mucha gente hasta que por fin le expuse mi idea a Gregorio a finales de octubre.

B.2. Estudio de los objetivos

Desde mi primera reunión con Gregorio hasta finales de noviembre estuve leyendo muchos libros sobre HTML5 y CSS3 con el fin de obtener ideas novedosas que pudiese relacionar con la idea principal de añadir mejoras a una web de asociación de animales.

B.3. Selección de las herramientas, así como el estudio de las mismas

En diciembre le comenté a Gregorio los objetivos del proyecto y tras dar el visto bueno comencé a buscar información para la elección del servidor. Fue a mediados de diciembre cuando ya tenía elegidas las herramientas y empecé a programar la aplicación. A partir de

aquí empleé unas 6 horas diarias al desarrollo de la aplicación entre semana, aumentando a 8 horas los fines de semana.

B.4. Desarrollo de la aplicación

Desde diciembre a abril me dediqué al desarrollo de cada uno de los objetivos, hasta que le enseñé el resultado final a Gregorio y decidimos que ya podía empezar por escribir la memoria final del proyecto.

B.5. Desarrollo de la memoria

La memoria la tuve que hacer mientras a su vez estudiaba la última asignatura que me queda: Procesamiento de Señal y Sistemas Móviles, por ello el desarrollo de la misma se ha alargado casi dos meses.

APÉNDICE C

LICENCIA CREATIVE COMMONS

CREATIVE COMMONS CORPORATION NO ES UN DESPACHO DE ABOGADOS Y NO PROPORCIONA SERVICIOS JURÍDICOS. LA DISTRIBUCIÓN DE ESTA LICENCIA NO CREA UNA RELACIÓN ABOGADO-CLIENTE. CREATIVE COMMONS PROPORCIONA ESTA INFORMACIÓN TAL CUAL (ON AN AS-IS BASIS). CREATIVE COMMONS NO OFRECE GARANTÍA ALGUNA RESPECTO DE LA INFORMACIÓN PROPORCIONADA, NI ASUME RESPONSABILIDAD ALGUNA POR DAÑOS PRODUCIDOS A CONSECUENCIA DE SU USO.

Licencia

LA OBRA O LA PRESTACIÓN (SEGÚN SE DEFINEN MÁS ADELANTE) SE PROPORCIONA BAJO LOS TÉRMINOS DE ESTA LICENCIA PÚBLICA DE CREATIVE COMMONS (CCPL O LICENCIA). LA OBRA O LA PRESTACIÓN SE ENCUENTRA PROTEGIDA POR LA LEY ESPAÑOLA DE PROPIEDAD INTELECTUAL Y/O CUALESQUIERA OTRAS NORMAS QUE RESULTEN DE APLICACIÓN. QUEDA PROHIBIDO CUALQUIER USO DE LA OBRA O PRESTACIÓN DIFERENTE A LO AUTORIZADO BAJO ESTA LICENCIA O LO DISPUESTO EN LA LEY DE PROPIEDAD INTELECTUAL.

MEDIANTE EL EJERCICIO DE CUALQUIER DERECHO SOBRE LA OBRA O LA PRESTACIÓN, USTED ACEPTA Y CONSIENTE LAS LIMITACIONES Y OBLIGACIONES DE ESTA LICENCIA, SIN PERJUICIO DE LA NECESIDAD DE CONSENTIMIENTO EXPRESO EN CASO DE VIOLACIÓN PREVIA DE LOS TÉRMINOS DE LA MISMA. EL LICENCIADOR LE CONCEDE LOS DERECHOS CONTENIDOS EN ESTA LICENCIA, SIEMPRE QUE USTED ACEPTE LOS PRESENTES TÉRMINOS Y CONDICIONES.

C.1. Definiciones

- a. La *obra* es la creación literaria, artística o científica ofrecida bajo los términos de esta licencia.
- b. En esta licencia se considera una *prestación* cualquier interpretación, ejecución,

fonograma, grabación audiovisual, emisión o transmisión, mera fotografía u otros objetos protegidos por la legislación de propiedad intelectual vigente aplicable.

- c. La aplicación de esta licencia a una **colección** (definida más adelante) afectará únicamente a su estructura en cuanto forma de expresión de la selección o disposición de sus contenidos, no siendo extensiva a éstos. En este caso la colección tendrá la consideración de obra a efectos de esta licencia.
- d. El **titular originario** es:
- I. En el caso de una obra literaria, artística o científica, la persona natural o grupo de personas que creó la obra.
 - II. En el caso de una obra colectiva, la persona que la edite y divulgue bajo su nombre, salvo pacto contrario.
 - III. En el caso de una interpretación o ejecución, el actor, cantante, músico, o cualquier otra persona que represente, cante, lea, recite, interprete o ejecute en cualquier forma una obra.
 - IV. En el caso de un fonograma, el productor fonográfico, es decir, la persona natural o jurídica bajo cuya iniciativa y responsabilidad se realiza por primera vez una fijación exclusivamente sonora de la ejecución de una obra o de otros sonidos.
 - V. En el caso de una grabación audiovisual, el productor de la grabación, es decir, la persona natural o jurídica que tenga la iniciativa y asuma la responsabilidad de las fijaciones de un plano o secuencia de imágenes, con o sin sonido.
 - VI. En el caso de una emisión o una transmisión, la entidad de radiodifusión.
 - VII. En el caso de una mera fotografía, aquella persona que la haya realizado.
 - VIII. En el caso de otros objetos protegidos por la legislación de propiedad intelectual vigente, la persona que ésta señale.
- e. Se considerarán **obras derivadas** aquellas obras creadas a partir de la licenciada, como por ejemplo: las traducciones y adaptaciones; las revisiones, actualizaciones y anotaciones; los compendios, resúmenes y extractos; los arreglos musicales y, en general, cualesquiera transformaciones de una obra literaria, artística o científica. Para evitar la duda, si la obra consiste en una composición musical o grabación de sonidos, la sincronización temporal de la obra con una imagen en movimiento (synching) será considerada como una obra derivada a efectos de esta licencia.
- f. Tendrán la consideración de **colecciones** la recopilación de obras ajenas, de datos o de otros elementos independientes como las antologías y las bases de datos que por la selección o disposición de sus contenidos constituyan creaciones intelectuales. La mera incorporación de una obra en una colección no dará lugar a una derivada a efectos de esta licencia.
- g. El **licenciador** es la persona o la entidad que ofrece la obra o prestación bajo los términos de esta licencia y le concede los derechos de explotación de la misma conforme a lo dispuesto en ella.

- h. **Usted** es la persona o la entidad que ejercita los derechos concedidos mediante esta licencia y que no ha violado previamente los términos de la misma con respecto a la obra o la prestación, o que ha recibido el permiso expreso del licenciador de ejercitar los derechos concedidos mediante esta licencia a pesar de una violación anterior.
- i. La **transformación** de una obra comprende su traducción, adaptación y cualquier otra modificación en su forma de la que se derive una obra diferente. La creación resultante de la transformación de una obra tendrá la consideración de obra derivada.
- j. Se entiende por **reproducción** la fijación directa o indirecta, provisional o permanente, por cualquier medio y en cualquier forma, de toda la obra o la prestación o de parte de ella, que permita su comunicación o la obtención de copias.
- k. Se entiende por **distribución** la puesta a disposición del público del original o de las copias de la obra o la prestación, en un soporte tangible, mediante su venta, alquiler, préstamo o de cualquier otra forma.
- l. Se entiende por **comunicación pública** todo acto por el cual una pluralidad de personas, que no pertenezcan al ámbito doméstico de quien la lleva a cabo, pueda tener acceso a la obra o la prestación sin previa distribución de ejemplares a cada una de ellas. Se considera comunicación pública la puesta a disposición del público de obras o prestaciones por procedimientos alámbricos o inalámbricos, de tal forma que cualquier persona pueda acceder a ellas desde el lugar y en el momento que elija.
- m. La **explotación** de la obra o la prestación comprende la reproducción, la distribución, la comunicación pública y, en su caso, la transformación.
- n. Los **elementos de la licencia** son las características principales de la licencia según la selección efectuada por el licenciador e indicadas en el título de esta licencia: Reconocimiento, CompartirIgual.
- ñ. Una **licencia equivalente** es:
- I. Una versión posterior de esta licencia de Creative Commons con los mismos elementos de licencia.
 - II. La misma versión o una versión posterior de esta licencia de cualquier otra jurisdicción reconocida por Creative Commons con los mismos elementos de la licencia (ejemplo: Reconocimiento-CompartirIgual 3.0 Japón).
 - III. La misma versión o una versión posterior de la licencia de Creative Commons no adaptada a ninguna jurisdicción (*Unported*) con los mismos elementos de la licencia.
 - IV. Una de las licencias compatibles que aparece en <http://creativecommons.org/compatiblelicenses> y que ha sido aprobada por Creative Commons como esencialmente equivalente a esta licencia porque, como mínimo:
 - a. Contiene términos con el mismo propósito, el mismo significado y el mismo efecto que los elementos de esta licencia.

- b. Permite explícitamente que las obras derivadas de obras sujetas a ella puedan ser distribuidas mediante esta licencia, la licencia de Creative Commons no adaptada a ninguna jurisdicción (*Unported*) o una licencia de cualquier otra jurisdicción reconocida por Creative Commons, con sus mismos elementos de licencia.

C.2. Límites de los derechos

Nada en esta licencia pretende reducir o restringir cualesquiera límites legales de los derechos exclusivos del titular de los derechos de propiedad intelectual de acuerdo con la Ley de propiedad intelectual o cualesquiera otras leyes aplicables, ya sean derivados de usos legítimos, tales como la copia privada o la cita, u otras limitaciones como la resultante de la primera venta de ejemplares (agotamiento).

C.3. Concesión de licencia

Conforme a los términos y a las condiciones de esta licencia, el licenciador concede, por el plazo de protección de los derechos de propiedad intelectual y a título gratuito, una licencia de ámbito mundial no exclusiva que incluye los derechos siguientes:

- a. Derecho de reproducción, distribución y comunicación pública de la obra o la prestación.
- b. Derecho a incorporar la obra o la prestación en una o más colecciones.
- c. Derecho de reproducción, distribución y comunicación pública de la obra o la prestación lícitamente incorporada en una colección.
- d. Derecho de transformación de la obra para crear una obra derivada siempre y cuando se incluya en ésta una indicación de la transformación o modificación efectuada.
- e. Derecho de reproducción, distribución y comunicación pública de obras derivadas creadas a partir de la obra licenciada.
- f. Derecho a extraer y reutilizar la obra o la prestación de una base de datos.
- g. Para evitar cualquier duda, el titular originario:
 - I. Conserva el derecho a percibir las remuneraciones o compensaciones previstas por actos de explotación de la obra o prestación, calificadas por la ley como irrenunciables e inalienables y sujetas a gestión colectiva obligatoria.
 - II. Renuncia al derecho exclusivo a percibir, tanto individualmente como mediante una entidad de gestión colectiva de derechos, cualquier remuneración derivada de actos de explotación de la obra o prestación que usted realice.

Estos derechos se pueden ejercitar en todos los medios y formatos, tangibles o intangibles, conocidos en el momento de la concesión de esta licencia. Los derechos mencionados incluyen el derecho a efectuar las modificaciones que sean precisas técnicamente para el ejercicio de los derechos en otros medios y formatos. Todos los derechos no concedidos

expresamente por el licenciador quedan reservados, incluyendo, a título enunciativo pero no limitativo, los derechos morales irrenunciables reconocidos por la ley aplicable. En la medida en que el licenciador ostente derechos exclusivos previstos por la ley nacional vigente que implementa la directiva europea en materia de derecho sui generis sobre bases de datos, renuncia expresamente a dichos derechos exclusivos.

C.4. Restricciones

La concesión de derechos que supone esta licencia se encuentra sujeta y limitada a las restricciones siguientes:

- a. Usted puede reproducir, distribuir o comunicar públicamente la obra o prestación solamente bajo los términos de esta licencia y debe incluir una copia de la misma, o su Identificador Uniforme de Recurso (URI). Usted no puede ofrecer o imponer ninguna condición sobre la obra o prestación que altere o restrinja los términos de esta licencia o el ejercicio de sus derechos por parte de los concesionarios de la misma. Usted no puede sublicenciar la obra o prestación. Usted debe mantener intactos todos los avisos que se refieran a esta licencia y a la ausencia de garantías. Usted no puede reproducir, distribuir o comunicar públicamente la obra o prestación con medidas tecnológicas que controlen el acceso o el uso de una manera contraria a los términos de esta licencia. Esta sección 4.a también afecta a la obra o prestación incorporada en una colección, pero ello no implica que ésta en su conjunto quede automáticamente o deba quedar sujeta a los términos de la misma. En el caso que le sea requerido, previa comunicación del licenciador, si usted incorpora la obra en una colección y/o crea una obra derivada, deberá quitar cualquier crédito requerido en el apartado 4.c, en la medida de lo posible.
- b. Usted puede distribuir o comunicar públicamente una obra derivada en el sentido de esta licencia solamente bajo los términos de la misma u otra licencia equivalente. Si usted utiliza esta misma licencia debe incluir una copia o bien su URI, con cada obra derivada que usted distribuya o comunique públicamente. Usted no puede ofrecer o imponer ningún término respecto a la obra derivada que altere o restrinja los términos de esta licencia o el ejercicio de sus derechos por parte de los concesionarios de la misma. Usted debe mantener intactos todos los avisos que se refieran a esta licencia y a la ausencia de garantías cuando distribuya o comunique públicamente la obra derivada. Usted no puede ofrecer o imponer ningún término respecto de las obras derivadas o sus transformaciones que alteren o restrinjan los términos de esta licencia o el ejercicio de sus derechos por parte de los concesionarios de la misma. Usted no puede reproducir, distribuir o comunicar públicamente la obra derivada con medidas tecnológicas que controlen el acceso o uso de la obra de una manera contraria a los términos de esta licencia. Si utiliza una licencia equivalente debe cumplir con los requisitos que ésta establezca cuando distribuya o comunique públicamente la obra derivada. Todas estas condiciones se aplican a una obra derivada en tanto que incorporada a una colección, pero no implica que ésta tenga que estar sujeta a los términos de esta licencia.

- c. Si usted reproduce, distribuye o comunica públicamente la obra o la prestación, una colección que la incorpore o cualquier obra derivada, debe mantener intactos todos los avisos sobre la propiedad intelectual e indicar, de manera razonable conforme al medio o a los medios que usted esté utilizando:
- I. El nombre del autor original, o el seudónimo si es el caso, así como el del titular originario, si le es facilitado.
 - II. El nombre de aquellas partes (por ejemplo: institución, publicación, revista) que el titular originario y/o el licenciador designen para ser reconocidos en el aviso legal, las condiciones de uso, o de cualquier otra manera razonable.
 - III. El título de la obra o la prestación si le es facilitado.
 - IV. El URI, si existe, que el licenciador especifique para ser vinculado a la obra o la prestación, a menos que tal URI no se refiera al aviso legal o a la información sobre la licencia de la obra o la prestación.
 - V. En el caso de una obra derivada, un aviso que identifique la transformación de la obra en la obra derivada (p. ej., "traducción castellana de la obra de Autor Original," "guión basado en obra original de Autor Original").

Este reconocimiento debe hacerse de manera razonable. En el caso de una obra derivada o incorporación en una colección estos créditos deberán aparecer como mínimo en el mismo lugar donde se hallen los correspondientes a otros autores o titulares y de forma comparable a los mismos. Para evitar la duda, los créditos requeridos en esta sección sólo serán utilizados a efectos de atribución de la obra o la prestación en la manera especificada anteriormente. Sin un permiso previo por escrito, usted no puede afirmar ni dar a entender implícitamente ni explícitamente ninguna conexión, patrocinio o aprobación por parte del titular originario, el licenciador y/o las partes reconocidas hacia usted o hacia el uso que hace de la obra o la prestación.

- d. Para evitar cualquier duda, debe hacerse notar que las restricciones anteriores (párrafos 4.a, 4.b y 4.c) no son de aplicación a aquellas partes de la obra o la prestación objeto de esta licencia que únicamente puedan ser protegidas mediante el derecho sui generis sobre bases de datos recogido por la ley nacional vigente implementando la directiva europea de bases de datos

C.5. Exoneración de responsabilidad

A MENOS QUE SE ACUERDE MUTUAMENTE ENTRE LAS PARTES, EL LICENCIADOR OFRECE LA OBRA O LA PRESTACIÓN TAL CUAL (ON AN AS-IS BASIS) Y NO CONFIERE NINGUNA GARANTÍA DE CUALQUIER TIPO RESPECTO DE LA OBRA O LA PRESTACIÓN O DE LA PRESENCIA O AUSENCIA DE ERRORES QUE PUEDAN O NO SER DESCUBIERTOS. ALGUNAS JURISDICCIONES NO PERMITEN LA EXCLUSIÓN DE TALES GARANTÍAS, POR LO QUE TAL EXCLUSIÓN PUEDE NO SER DE APLICACIÓN A USTED.

C.6. Limitación de responsabilidad

SALVO QUE LO DISPONGA EXPRESA E IMPERATIVAMENTE LA LEY APLICABLE, EN NINGÚN CASO EL LICENCIADOR SERÁ RESPONSABLE ANTE USTED POR CUALESQUIERA DAÑOS RESULTANTES, GENERALES O ESPECIALES (INCLUIDO EL DAÑO EMERGENTE Y EL LUCRO CESANTE), FORTUITOS O CAUSALES, DIRECTOS O INDIRECTOS, PRODUCIDOS EN CONEXIÓN CON ESTA LICENCIA O EL USO DE LA OBRA O LA PRESTACIÓN, INCLUSO SI EL LICENCIADOR HUBIERA SIDO INFORMADO DE LA POSIBILIDAD DE TALES DAÑOS.

C.7. Finalización de la licencia

- a. Esta licencia y la concesión de los derechos que contiene terminarán automáticamente en caso de cualquier incumplimiento de los términos de la misma. Las personas o entidades que hayan recibido de usted obras derivadas o colecciones bajo esta licencia, sin embargo, no verán sus licencias finalizadas, siempre que tales personas o entidades se mantengan en el cumplimiento íntegro de esta licencia. Las secciones 1, 2, 5, 6, 7 y 8 permanecerán vigentes pese a cualquier finalización de esta licencia.
- b. Conforme a las condiciones y términos anteriores, la concesión de derechos de esta licencia es vigente por todo el plazo de protección de los derechos de propiedad intelectual según la ley aplicable. A pesar de lo anterior, el licenciador se reserva el derecho a divulgar o publicar la obra o la prestación en condiciones distintas a las presentes, o de retirar la obra o la prestación en cualquier momento. No obstante, ello no supondrá dar por concluida esta licencia (o cualquier otra licencia que haya sido concedida, o sea necesario ser concedida, bajo los términos de esta licencia), que continuará vigente y con efectos completos a no ser que haya finalizado conforme a lo establecido anteriormente, sin perjuicio del derecho moral de arrepentimiento en los términos reconocidos por la ley de propiedad intelectual aplicable.

C.8. Miscelánea

- a. Cada vez que usted realice cualquier tipo de explotación de la obra o la prestación, o de una colección que la incorpore, el licenciador ofrece a los terceros y sucesivos licenciatarios la concesión de derechos sobre la obra o la prestación en las mismas condiciones y términos que la licencia concedida a usted.
- b. Cada vez que usted realice cualquier tipo de explotación de una obra derivada, el licenciador ofrece a los terceros y sucesivos licenciatarios la concesión de derechos sobre la obra objeto de esta licencia en las mismas condiciones y términos que la licencia concedida a usted.
- c. Si alguna disposición de esta licencia resulta inválida o inaplicable según la Ley vigente, ello no afectará la validez o aplicabilidad del resto de los términos de esta licencia y, sin ninguna acción adicional por cualquiera de las partes de este acuerdo, tal disposición se entenderá reformada en lo estrictamente necesario para hacer que tal disposición sea válida y ejecutiva.

- d. No se entenderá que existe renuncia respecto de algún término o disposición de esta licencia, ni que se consiente violación alguna de la misma, a menos que tal renuncia o consentimiento figure por escrito y lleve la firma de la parte que renuncie o consienta.
- e. Esta licencia constituye el acuerdo pleno entre las partes con respecto a la obra o la prestación objeto de la licencia. No caben interpretaciones, acuerdos o condiciones con respecto a la obra o la prestación que no se encuentren expresamente especificados en la presente licencia. El licenciador no estará obligado por ninguna disposición complementaria que pueda aparecer en cualquier comunicación que le haga llegar usted. Esta licencia no se puede modificar sin el mutuo acuerdo por escrito entre el licenciador y usted.

C.9. Aviso de Creative Commons

Creative Commons no es parte de esta licencia, y no ofrece ninguna garantía en relación con la obra o la prestación. Creative Commons no será responsable frente a usted o a cualquier parte, por cualesquiera daños resultantes, incluyendo, pero no limitado, daños generales o especiales (incluido el daño emergente y el lucro cesante), fortuitos o causales, en conexión con esta licencia. A pesar de las dos (2) oraciones anteriores, si Creative Commons se ha identificado expresamente como el licenciador, tendrá todos los derechos y obligaciones del licenciador.

Salvo para el propósito limitado de indicar al público que la obra o la prestación está licenciada bajo la CCPL, ninguna parte utilizará la marca registrada "Creative Commons" o cualquier marca registrada o insignia relacionada con "Creative Commons" sin su consentimiento por escrito. Cualquier uso permitido se hará de conformidad con las pautas vigentes en cada momento sobre el uso de la marca registrada por "Creative Commons", en tanto que sean publicadas su sitio web (website) o sean proporcionadas a petición previa. Para evitar cualquier duda, estas restricciones en el uso de la marca no forman parte de esta licencia.

Puede contactar con Creative Commons en: <http://creativecommons.org/>.

Glosario

A

API Conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

Aplicación Web Se trata de una herramienta que permite a los usuarios interactuar con con una serie de datos con el fin de lograr una tarea específica en la web; es decir, las herramientas que pueden utilizar los usuarios para acceder a un servidor web a través de Internet.

C

Callback Es una función A que se usa como argumento de otra función B.

Concurrencia Es la simultaneidad en la ejecución de múltiples tareas interactivas. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa. Las tareas se pueden ejecutar en una sola unidad central de proceso, en varios procesadores o en una red de computadores distribuidos. La programación concurrente está relacionada con la programación paralela, pero enfatiza más la interacción entre tareas. Así, la correcta secuencia de interacciones o comunicaciones entre los procesos y el acceso coordinado de recursos que se comparten por todos los procesos o tareas son las claves de esta disciplina.

Cortafuegos Es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.

E

Escalabilidad Es la capacidad de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejando el crecimiento continuo de trabajo de manera fluida, o bien estando preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

Extranet Se puede decir en otras palabras que una Extranet es parte de la Intranet de una organización que se extiende a usuarios fuera de ella.

F

Framework Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

G

GPL Es la licencia más ampliamente usada¹ en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.

I

Intranet Es una red informática que utiliza la tecnología del Protocolo de Internet para compartir información, sistemas operativos o servicios de computación dentro de una organización.

K

Keyframe Son fotogramas con un contenido específico, se crean, por tanto, para insertar en ellos un nuevo contenido no existente en la película.

M

metadatos Datos que describen otros datos, por ejemplo en una biblioteca las fichas que sirven para especificar libros, autores...

Middleware Es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones que son necesarias en los sistemas distribuidos. De esta forma se provee una solución que mejora la calidad de servicio, seguridad, envío de mensajes, directorio de servicio, etc...

Motor Javascript Es un intérprete informático que interpreta el código fuente en lenguaje JavaScript y ejecuta un script acorde a las instrucciones dadas.

O

ODBC Es un estándar de acceso a las bases de datos desarrollado por SQL Access Group en 1992. El objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos almacene los datos.

P

peer to peer Es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados.

plugin Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

R

REST REST es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP. REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP.

S

Serializar Consiste en un proceso de codificación de un objeto en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato humanamente más legible como XML o JSON, entre otros. La serie de bytes o el formato pueden ser usados para crear un nuevo objeto que es idéntico en todo al original, incluido su estado interno (por tanto, el nuevo objeto es un clon del original). La serialización es un mecanismo ampliamente usado para transportar objetos a través de una red, para hacer persistente un objeto en un archivo o base de datos, o para distribuir objetos idénticos a varias aplicaciones o localizaciones.

Sockets Constituyen el mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos apropiados. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.

Software Se trata del equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.

T

TCP/IP El modelo TCP/IP, describe un conjunto de guías generales de diseño e implementación de protocolos de red específicos para permitir que un equipo pueda comunicarse en una red. TCP/IP provee conectividad de extremo a extremo especificando cómo los datos deberían ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario. Existen protocolos para los diferentes tipos de servicios de comunicación entre equipos.

U

Unix *Unix* es un sistema operativo portable, multitarea y multiusuario desarrollado por los laboratorios BELL en 1969. Existen muchas familias dentro de *Unix* que han evolucionado de forma independiente, como es el sistema operativo *Linux*.

url blob La url no se refiere a los datos que existen en el servidor, se refiere a los datos que el navegador tiene actualmente en la memoria, de la página actual. Por lo que no va a estar disponible en otras páginas, ni en otros navegadores, tampoco desde otros equipos.

W

Widget Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual. Aunque no es condición indispensable, los widgets suelen ser utilizados para ser insertados en otra página web, copiando el código que el mismo widget pone a disposición del usuario.

Bibliografía

- [1] S. L. Mora, “Programación en Internet. Clientes web,” Tech. Rep. [Online]. Available: http://rua.ua.es/dspace/bitstream/10045/16995/1/sergio_lujan-programacion_de_aplicaciones_web.pdf
- [2] (Mayo, 2014) Soporte geolocalización navegadores. [Online]. Available: <http://blog.video2brain.com/es/geolocalizacion-en-html5.html>
- [3] (Mayo, 2014) Sprites y css3. [Online]. Available: <http://www.desarrolloweb.com/articulos/sprites-css.html>
- [4] (Mayo, 2014) Ejemplos reales html5 y css3. [Online]. Available: <http://www.pcworld.es/redes-sociales/vimeo-construye-un-nuevo-reproductor-html5-desde-cero-para-acelerar-la-carga-de-videos>
- [5] (Mayo, 2014) Comparativa entre node.js y apache. [Online]. Available: <http://zgadzaj.com/benchmarking-nodejs-basic-performance-tests-against-apache-php>
- [6] (Mayo, 2014) Configuración xampp. [Online]. Available: http://www.mclubre.org/consultar/php/otros/in_php_configuracion.html
- [7] (Mayo, 2014) Arquitectura cliente servidor. [Online]. Available: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>
- [8] (Mayo, 2014) Historia de las aplicaciones web. [Online]. Available: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>
- [9] (Mayo, 2014) Consorcio Worl Wild Web. [Online]. Available: <http://www.w3c.es/Consorcio/about-w3c.html>
- [10] (Mayo, 2014) Historia html5. [Online]. Available: <http://www.uv.es/jac/guia/html5.html>
- [11] (Mayo, 2014) Nuevas características html5. [Online]. Available: <https://developer.mozilla.org/es/docs/HTML/HTML5>
- [12] (Mayo, 2014) Ventajas de html5. [Online]. Available: <http://www.go-movil.es/index.php/12-ventajas-del-html5/>

- [13] (Mayo, 2014) Características de css3. [Online]. Available: <http://pamarke.com/css3-sus-nuevas-propiedades/>
- [14] (Mayo, 2014) Principales características html5 y css3. [Online]. Available: <http://www.newemage.com.mx/blog/paginas-web/ventas-de-html5-y-css3>
- [15] (Mayo, 2014) Conceptos de node.js. [Online]. Available: <http://unadocenade.com/una-docena-de-conceptos-que-deberias-conocer-node-js/>
- [16] (Mayo, 2014) Uso de node.js. [Online]. Available: <http://venturebeat.com/2011/08/16/linkedin-node/>
- [17] (Mayo, 2014) Ejemplos reales de utilización de node.js. [Online]. Available: <http://nodejs.org/industry/>
- [18] (Mayo, 2014) La base de datos mysql. [Online]. Available: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf
- [19] (Mayo, 2014) Tutorial xampp. [Online]. Available: <https://www.udemy.com/blog/tutorial-de-xampp-como-usar-xampp-para-ejecutar-su-propio-servidor-web/>
- [20] (Mayo, 2014) Servidores web. [Online]. Available: <http://achetemele.wordpress.com/2013/04/10/que-tipo-de-servidor-utilizar-wamp-mamp-xampp-lamp/>
- [21] (Mayo, 2014) Aplicaciones web. [Online]. Available: http://es.wikipedia.org/wiki/Aplicación_web
- [22] (Mayo, 2014) Fases de internet. [Online]. Available: <http://www.rocasalvatella.com/es/blog/2012/11/llega-la-internet-de-las-cosas-llega-la-internet-de-los-servicios>
- [23] (Mayo, 2014) Información html5. [Online]. Available: <http://www.w3.org/TR/html5/>
- [24] (Mayo, 2014) Nuevos tipos html5. [Online]. Available: <http://www.anerbarrena.com/nuevos-input-html5-3739/>
- [25] (Mayo, 2014) Nuevas etiquetas html5. [Online]. Available: <http://tutomalaga.blogspot.com.es/2012/12/breve-introduccion-html-5.html>
- [26] (Mayo, 2014) Tipos html5. [Online]. Available: <http://olgacarreras.blogspot.com.es/2014/03/html5-y-accesibilidad-nuevos-tipos-de.html>
- [27] (Mayo, 2014) Compatibilidad entre navegadores html5. [Online]. Available: <http://desarrolloweb.dlsi.ua.es/cursos/2012/nuevos-estandares-desarrollo-sitios-web/compatibilidad-navegadores-html5>
- [28] Página oficial node.js.
- [29] (Mayo, 2014) Node.js y el bucle de eventos. [Online]. Available: <http://mocionsoft.com/blog/porque-usamos-node-js/>

- [30] (Mayo, 2014) Cifras sobre el crecimiento de internet. [Online]. Available: <http://nachocarnes.wordpress.com/2014/01/28/cifras-de-internet-en-2013/>
- [31] (Mayo, 2014) Semántica html5 y html4. [Online]. Available: <http://www.aloud.es/html5-web-presentacion-etiquetas-semanticas-mas-populares>

