

FIND ME

CICLO FORMATIVO DE GRADO SUPERIOR

DESAROLLO DE APLICACIONES WEB

Autor

VICTOR JIMENEZ ANDREU

TUTORES

JESUS VIVES CESPEDES

JOSE ANTONIO NAVARRETE SANCHEZ

COORDINADOR

JOSE ANTONIO NAVARRETE SANCHEZ

ÍNDICE

1. INTRODUCCIÓN	4
2. ENGLISH INTRODUCTION	5
3. OBJETIVOS	6
3.1 Objetivos de la fase actual	6
3.2 Objetivos en fases futuras	7
4. PLANIFICACIÓN	9
4.1 Tabla de hitos	10
4.2 Diagrama de Gantt	11
5. ANALISIS	12
5.1 Análisis del estado del arte	12
5.2 Funcionalidades	13
6. DISEÑO	15
6.1 Requisitos técnicos	15
6.2 Arquitectura web	16
6.3 Diseño back-end	18
6.3.1 Modelo de datos	19
6.3.2 Servicios REST	21
6.3.3 Paquetes adicionales	21
6.4 Diseño front-end	22
6.4.1 Mock-ups	22
6.4.2 Guía de estilos	23
6.4.3 Paquetes adicionales	24
6.4.4 Capturas de la aplicación	25

7.	IMPLEMENTACIÓN.....	33
7.1	Servidor	33
7.2	Cliente.....	45
8.	DESPLIEGUE	46
8.1	Modelo de despliegue utilizado.....	46
8.2	Datos iniciales y configuración.....	47
8.3	Pasos para el despliegue.....	47
8.4	Proveedores y servicios utilizados	48
9.	HERRAMIENTAS UTILIZADAS.....	49
10.	VIABILIDAD DEL PROYECTO.....	51
10.1	Viabilidad técnica	51
10.2	Viabilidad funcional	52
10.3	Posible uso real	52
10.4	Propuestas de mejora.....	53
11.	PREVENCIÓN DE RIESGOS LABORALES	54
11.1	Riesgos asociados al desarrollo	54
11.2	Medidas de prevención adoptadas	55
11.3	Buenas prácticas implementadas	55
ANEXO 1.	Bibliografía	57

1. INTRODUCCIÓN

El presente documento recoge el desarrollo completo del proyecto FindMe, realizado en el marco del módulo de Proyecto del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web.

FindMe es una aplicación web orientada a facilitar la búsqueda y localización de animales perdidos o encontrados por ciudadanos. Surge como respuesta a la necesidad real de disponer de una plataforma centralizada donde los usuarios puedan publicar, visualizar, compartir y gestionar este tipo de situaciones de forma colaborativa, ágil y segura.

La aplicación ofrece funcionalidades clave como la creación de publicaciones, comentarios, valoraciones, un sistema de mensajería privada entre usuarios y un módulo de denuncias tanto de contenido como de usuarios. También incluye herramientas de administración para la moderación de contenido, aplicación de sanciones (baneos) y visualización de chats denunciados, aportando un entorno más controlado y confiable.

El desarrollo ha sido llevado a cabo con tecnologías estándar del entorno web, utilizando PHP, MySQL, HTML5, JavaScript y CSS, todo ello estructurado con buenas prácticas de organización de código, separación de responsabilidades y validaciones tanto del lado cliente como servidor.

FindMe está diseñado para ser fácilmente desplegado en un servidor de hosting externo, siendo adaptable, escalable y con posibilidades claras de evolución hacia una versión con funcionalidades más avanzadas.

Este documento detalla todo el proceso del proyecto: desde su planteamiento inicial, objetivos y planificación, hasta su diseño, implementación y despliegue final, incluyendo además aspectos transversales como la viabilidad del proyecto y la prevención de riesgos laborales en el desarrollo.

2. ENGLISH INTRODUCTION

This document outlines the full development of the FindMe project, carried out within the framework of the Project module of the Advanced Vocational Training Program in Web Application Development.

FindMe is a web application designed to facilitate the search and identification of lost or found animals by citizens. It emerges as a response to the real need for a centralized platform where users can publish, view, share, and manage such cases in a collaborative, agile, and secure manner.

The application offers essential features such as post creation, comments, ratings, a private messaging system between users, and a reporting module for both content and users. It also includes administrative tools for content moderation, enforcement of sanctions (bans), and the review of reported conversations, contributing to a safer and more controlled environment.

The project has been developed using standard web technologies, including PHP, MySQL, HTML5, JavaScript, and CSS, following good practices in code organization, separation of concerns, and client-server validation.

FindMe is designed to be easily deployed on an external hosting server, with a flexible and scalable architecture, and clear potential for future versions with more advanced features.

This document covers the entire project process: from the initial planning, objectives, and scheduling, to the design, implementation, and final deployment, including transversal aspects such as project viability and risk prevention in software development.

3. OBJETIVOS

El objetivo principal del proyecto es desarrollar una herramienta web que permita favorecer la recuperación de animales perdidos mediante la creación de un espacio centralizado de comunicación entre personas que han extraviado o encontrado un animal.

Este proyecto persigue dar respuesta a una problemática real, mediante el diseño y desarrollo de una herramienta tecnológica que conecte a ciudadanos de forma ágil, accesible y organizada, promoviendo la colaboración comunitaria en casos de extravío animal.

El planteamiento parte de un enfoque práctico y social, orientado a ofrecer una respuesta digital eficaz a una necesidad cotidiana, asegurando su viabilidad técnica y su utilidad real en contextos urbanos o rurales.

3.1 Objetivos de la fase actual

Durante esta fase del proyecto, se ha establecido como objetivo principal el desarrollo completo de una versión funcional y estable de la aplicación FindMe, que permita su uso real sin necesidad de futuras ampliaciones inmediatas.

Entre los objetivos ya cumplidos para este proyecto están:

- a. Sistema funcional de registro y autenticación de usuarios.
- b. Sistema de anuncios y notificaciones sobre publicaciones de interés.
- c. Publicaciones personales con intención de encontrar animales perdidos.
- d. Implementación de mapas para facilitar la última ubicación del animal.
- e. Filtros para mostrar las publicaciones de interés y filtros para paginas exclusivas de administradores.

- f. Sistema de mensajería directa entre usuarios tanto pública como privada.
- g. Sistema de denuncias funcional donde los usuarios puedan señalar el mal uso de la aplicación por otro usuario.
- h. Sistema de castigo a usuarios no colaborativos mediante el uso de sanciones personalizadas.

Estos objetivos completados garantizan que la aplicación cumple con su cometido principal de crear un ambiente colaborativo y moderado con el fin principal de reunir a los animales con sus dueños.

3.2 Objetivos en fases futuras

De cara a una futura evolución del proyecto, se han identificado diversas mejoras y ampliaciones que permitirían enriquecer la plataforma FindMe, dotándola de nuevas funcionalidades y optimizando la experiencia del usuario.

Los principales objetivos para fases futuras son:

- a. Sistema de generación de rutas posibles que haya podido tomar el animal de la publicación dependiendo el tipo de animal.
- b. Sistema de recompensas para usuarios que incentiven el correcto uso de la aplicación mediante insignias visibles para el resto de los usuarios.
- c. Incorporación de opciones de difusión mediante redes sociales.
- d. Desarrollo de una sección que facilite la adopción de animales que no sean reclamados mediante la colaboración de protectoras de animales.

Estas mejoras a futuro consolidarían a este proyecto como una herramienta mucho más completa y versátil. Para el desarrollo de la mayoría de estas funciones se necesitaría mejorar el sistema general del proyecto. Para la mejora de facilitación de adopción a parte se necesitaría la ayuda externa de protectoras de animales consolidadas y dispuestas a colaborar.

4. PLANIFICACIÓN

Para garantizar el desarrollo ordenado y coherente del proyecto FindMe, se ha llevado a cabo una planificación previa que define las tareas principales, su duración estimada y la secuencia de ejecución. Esta planificación ha sido esencial para distribuir correctamente el tiempo disponible dentro del módulo de Proyecto y evitar retrasos durante el desarrollo.

Los primeros pasos para asegurar el correcto desarrollo de la aplicación fueron analizar los objetivos principales y diseñar tanto la base de datos como un backend inicial con una estructura adecuada.

El siguiente paso fue crear un frontend básico conectado con el backend para poder comprobar las primeras funciones de la aplicación, además de permitir visualizar posibles errores a futuro.

El siguiente paso fue ir terminando objetivos principales de la página completando el backend por completo. Una vez terminado el backend, continué completando el frontend en su totalidad y por último realizar las pruebas finales para asegurarme de que la aplicación cumple todas las funciones requeridas.

Por último, crearía la documentación completa del proyecto una vez terminado el programa y terminaría haciendo las pruebas finales del proyecto.

4.1 Tabla de hitos

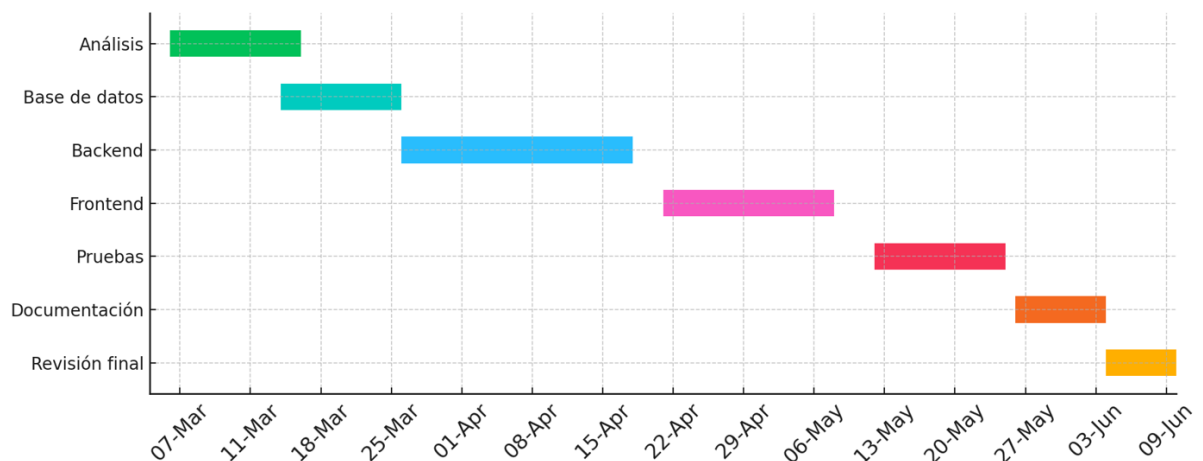
A continuación, se muestra una tabla donde se especifica mejor los tiempos de desarrollo del proyecto incluyendo la duración estimada de cada tarea. Cabe aclarar que señalo la duración en días no refiriéndome a 24 horas por día. Me refiero a días donde me haya centrado en ese apartado principalmente.

Fase	Tarea	Inicio	Duración (días)
Análisis	Definición de objetivos, estructura y funcionalidad principal.	07/03/2025	8
Base de datos	Diseño y creación completa de la base de datos. (Incluyendo actualizaciones)	12/03/2025	7
Backend	Lógica completa de la aplicación incluyendo funciones y despliegue	15/03/2025	22
Frontend	Creación completa de la interfaz del proyecto y conexión con el backend	21/03/2025	16
Pruebas	Pruebas de funciones desarrolladas y arreglos de errores encontrados	01/04/2025	12
Documentación	Redacción y preparación para la memoria completa del proyecto.	03/06/2025	3
Revisión final	Revisión general del proyecto	07/06/2025	2

4.2 Diagrama de Gantt

Para complementar la tabla de hitos presentada en el apartado anterior, he elaborado un diagrama de Gantt que permite visualizar gráficamente la planificación temporal del proyecto, mostrando la duración de las tareas.

Con este diagrama se podrá ver como avanza el proyecto de forma visual. Tengo que aclarar que el diagrama no representa de forma exacta el desarrollo del proyecto, aunque si muestra fielmente el orden de sucesos y estima la duración del desarrollo.



5. ANALISIS

El análisis previo ha sido esencial para entender el contexto en el que se sitúa el proyecto FindMe, definir sus funcionalidades clave y asegurar que su diseño y desarrollo respondan a una necesidad real.

Este apartado recoge tanto el estudio de otras soluciones existentes como la definición detallada de las funcionalidades implementadas. Todo ello ha permitido construir una base sólida sobre la que se ha desarrollado la aplicación.

5.1 Análisis del estado del arte

Existen paginas web y aplicaciones que tratan el mismo tema que FindMe. Páginas web como Rescota o Mascota Perdida 112 cumplen con un cometido similar al que cumple FindMe.

Aunque existan este tipo de páginas realmente no me he fijado en ninguna para tomar inspiración para el proyecto. Las funciones que se han desarrollado para FindMe no están inspiradas en ninguna otra aplicación existente, aunque coincidan funciones como el uso de mapas para localizar la zona donde se vio por última vez al animal.

Aunque mi proyecto no se inspire en otros en cuanto a lógica y funciones, aun así voy a señalar funciones que considero más correctas que los dos ejemplos señalados anteriormente.

Rescota y Mascota Perdida 112 tienen como función principal el mostrar un cartel de mascota perdida a usuarios de Facebook que estén en tu área designada. La función se parece a la de avisar a los usuarios que hay una publicación en su zona de FindMe.

Esta es una buena forma de difundir la publicación de mascota perdida, pero estas páginas únicamente funcionan con Facebook y en el caso de Rescota solo avisan a 5000 usuarios y no a todos los posibles habitantes del área designada.

FindMe tiene el potencial de alcanzar a muchas más personas por que no limita la cantidad de usuarios a los que llega la notificación, a demás de que cualquier usuario puede entrar a la publicación e interesarse. Además, FindMe también trata casos no reportados por el dueño del animal. Cualquier usuario puede reportar que ha encontrado un animal perdido y señalar su ubicación.

5.2 Funcionalidades

A continuación, listaré las principales funciones de FindMe a detalle que actualmente están totalmente desarrolladas e implementadas.

- a) Gestión de usuarios
 - a. Registro de nuevos usuarios.
 - b. Inicio y cierre de sesión
 - c. Recuperación de cuenta mediante verificación
 - d. Edición parcial del perfil
- b) Publicaciones
 - a. Creación de publicaciones
 - b. Inclusión de imágenes en la publicación
 - c. Compartir información de forma orgánica
 - d. Posibilidad de cerrar la publicación
 - e. Posibilidad de añadir la ubicación en un mapa
- c) Filtrado y búsqueda
 - a. Filtro por tipo de animal
 - b. Filtro por localidad

d) Comunicación

- a. Sistema de comentarios en la publicación
- b. Sistema de mensajes privados entre usuarios

e) Moderación

- a. Sistema de denuncias para usuarios, publicaciones e imágenes.
- b. Sección de gestión de denuncias
- c. Funciones para sancionar progresivamente a usuarios
- d. Mensajes de moderadores

f) Notificaciones

- a. Generación de notificaciones de diferente tipo
- b. Aviso por publicación en tu localidad
- c. Aviso por nuevos comentarios en tu publicación
- d. Aviso por mensaje privado

6. DISEÑO

El diseño del proyecto FindMe se ha llevado a cabo teniendo en cuenta tanto los requisitos funcionales definidos previamente como los aspectos técnicos necesarios para su implementación efectiva.

Se ha seguido una estructura modular, separando claramente el backend (lógica y conexión a base de datos) del frontend (interfaz de usuario), lo que facilita la organización del código, su mantenimiento y su futura escalabilidad.

A continuación, se detallan los elementos que componen el diseño técnico de la aplicación, desde los requisitos hasta la arquitectura, bases de datos, diseño visual y capturas reales de funcionamiento.

6.1 Requisitos técnicos

Partiendo de las funcionalidades definidas para el proyecto se determinan los requisitos técnicos necesarios para el correcto desarrollo. A continuación, hay una lista de los requisitos técnicos.

- a) Gestión de usuarios
 - a. PHP y MySQL para la validación de campos obligatorios
 - b. Sesiones con PHP
 - c. Validaciones usando JS y PHP

b) Publicaciones

- a. Formulario en HTML con envío POST, almacenamiento en MySQL
- b. Manejo de PHP y previsualización con JS (FileReader)
- c. Sistema de validación mediante PHP
- d. Integración con Leaflet.js y eventos para poder marcar el mapa

c) Filtrado y búsqueda

- a. Consultas SQL y menú desplegable para tipo de animal y localidad

d) Comunicación

- a. Comentarios almacenados en tablas relacionales usando JS y PHP
- b. Chat privado usando MySQL y PHP

e) Moderación

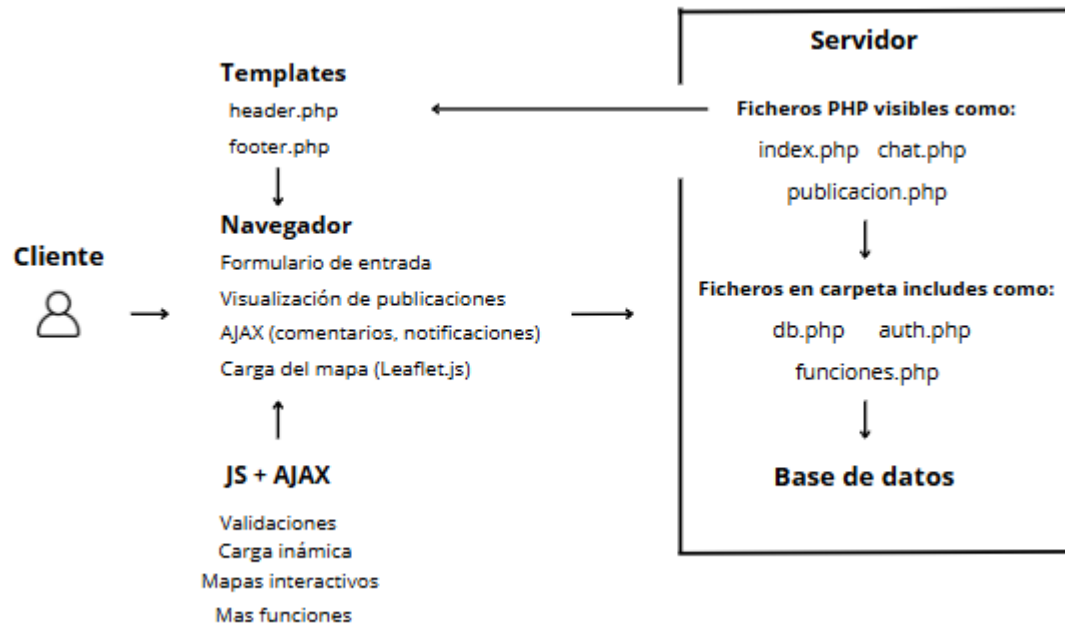
- a. Sistema de denuncias usando MySQL y PHP
- b. Lógica PHP junto con MySQL que calcule sanciones

f) Notificaciones

- a. Uso de PHP y consultas para controlar y avisar de eventos

6.2 Arquitectura web

A continuación, muestro un mapa con la estructura general del proyecto.



La solución implementada en el proyecto FindMe se basa en una arquitectura modular que se aproxima al modelo MVC (Modelo–Vista–Controlador), aunque sin seguir una estructura estricta.

Los ficheros PHP visibles actúan como controladores, ya que gestionan la lógica de cada página.

Los archivos de la carpeta includes/ contienen funciones reutilizables y consultas, actuando como parte del modelo.

Los templates (header.php, footer.php) y el código HTML representan la vista.

No se utiliza ningún framework ni sistema formal de enrutamiento, pero se respeta la separación entre lógica, datos y presentación.

Tampoco se utilizan servicios REST ni tecnologías basadas en contenedores.

6.3 Diseño back-end

El proyecto FindMe ha sido desarrollado en PHP estructurado sin el uso de ningún framework. Se ha optado por una arquitectura modular organizada por carpetas que separa claramente la lógica del servidor, los controladores visibles y los componentes reutilizables.

Organización general:

- Ficheros raíz (.php): actúan como puntos de entrada del sistema (login, publicaciones, perfil, etc.).
- Carpeta includes/: contiene la lógica del sistema, con scripts como db.php, funciones.php, auth.php o procesar_denuncia.php.
- Carpeta templates/: contiene elementos comunes de interfaz como header.php, footer.php, formularios preconfigurados, etc.
- Carpeta uploads/: almacena las imágenes subidas por los usuarios.

Configuración relevante:

- Conexión a base de datos definida en includes/db.php, centralizada para facilitar la reutilización.
- Uso intensivo de require e include para ensamblar páginas dinámicamente con fragmentos reutilizables.
- Control de acceso a páginas protegido mediante comprobaciones de sesión en cada archivo que lo requiera.
- Validaciones tanto en el cliente como en el servidor, para garantizar seguridad y consistencia de datos.

Este diseño busca simplicidad, eficiencia y claridad para facilitar el mantenimiento del sistema y permitir futuras ampliaciones sin depender de librerías externas ni frameworks complejos.

6.3.1 Modelo de datos

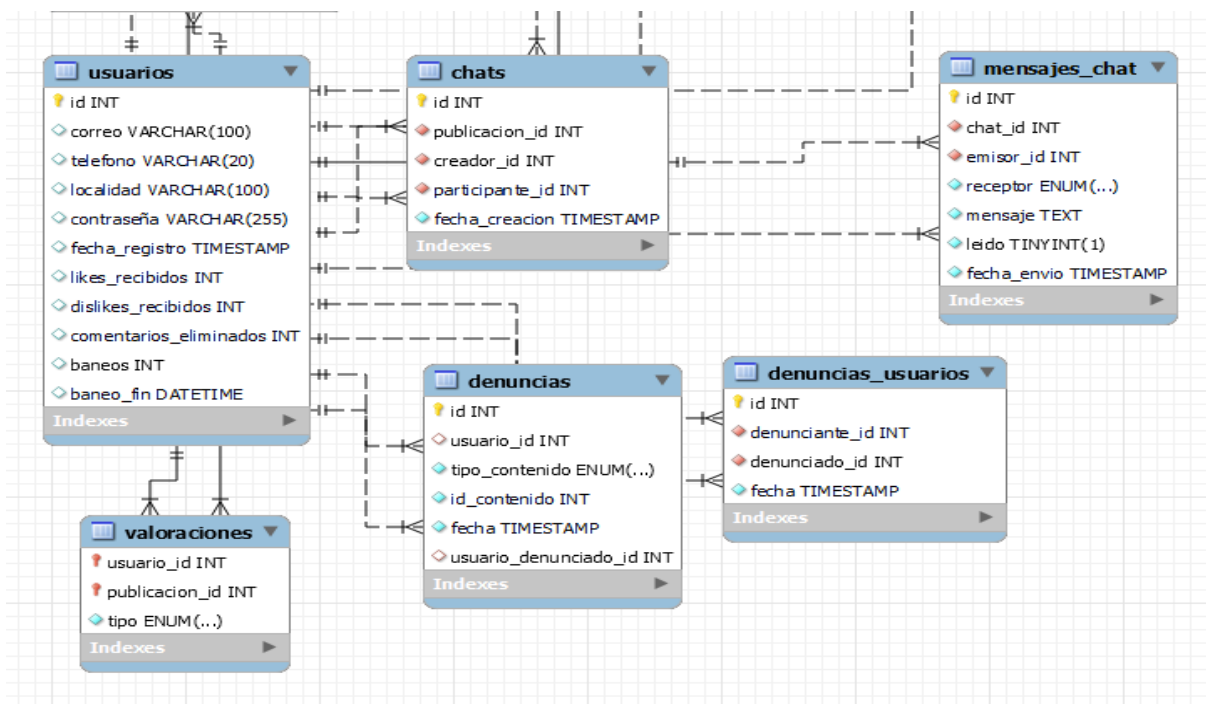
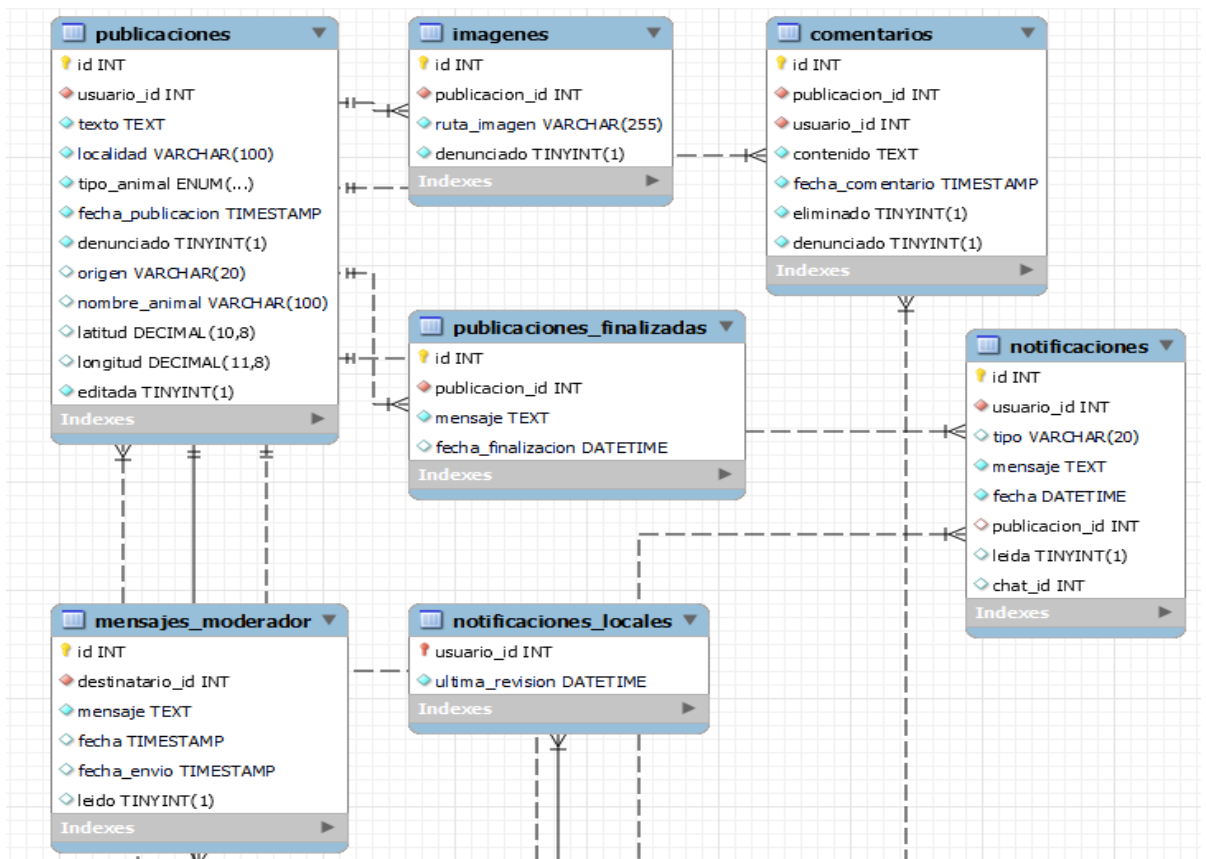
El modelo de datos de FindMe se basa en una estructura relacional implementada en MySQL. Está compuesto por diversas entidades interrelacionadas que permiten gestionar usuarios, publicaciones, comentarios, mensajes privados, notificaciones y denuncias.

A continuación, se describen las tablas principales y sus relaciones:

- usuarios: almacena los datos de registro, perfil y rol (usuario o administrador).
- publicaciones: contiene los anuncios de animales perdidos o encontrados creados por los usuarios.
- comentarios: registra los comentarios realizados en publicaciones.
- mensajes: gestiona los mensajes privados entre usuarios.
- denuncias: recoge denuncias de publicaciones o de usuarios.
- notificaciones: guarda los eventos importantes que deben ser notificados al usuario.
- baneos: mantiene el historial de sanciones aplicadas a los usuarios.

El diseño incluye claves foráneas para mantener la integridad referencial entre las tablas (por ejemplo, cada comentario está vinculado a una publicación y a un autor).

En la siguiente página se podrá ver el diagrama de la base de datos generada en MySQL Workbench mediante la función Reverse Engine.



6.3.2 Servicios REST

El proyecto FindMe no implementa servicios REST en su arquitectura actual.

La lógica del sistema está basada en un modelo tradicional de desarrollo web con PHP estructurado, donde la interacción entre cliente y servidor se realiza mediante formularios HTML estándar y peticiones asíncronas puntuales mediante AJAX.

No obstante, la arquitectura actual permitiría, en una fase futura, extender el sistema con una API REST para ofrecer funcionalidades como consumo externo de publicaciones, integración con apps móviles o automatización de tareas.

6.3.3 Paquetes adicionales

Durante el desarrollo del proyecto FindMe, se han utilizado algunos paquetes y recursos externos que han facilitado la implementación de funcionalidades concretas. A continuación, se enumeran los más relevantes:

He utilizado Leaflet.js para poder implementar mapas al proyecto. Junto con un JS adecuado se podrá editar y señalar activamente en el mapa una ubicación.

También he implementado jQuery para poder cargar comentarios y poder simplificar algunas llamadas AJAX.

Otro implemento es AJAX para enviar y recibir datos sin recargar la página.

Por último utilizo recursos menores como iconos para zonas puntuales de la página.

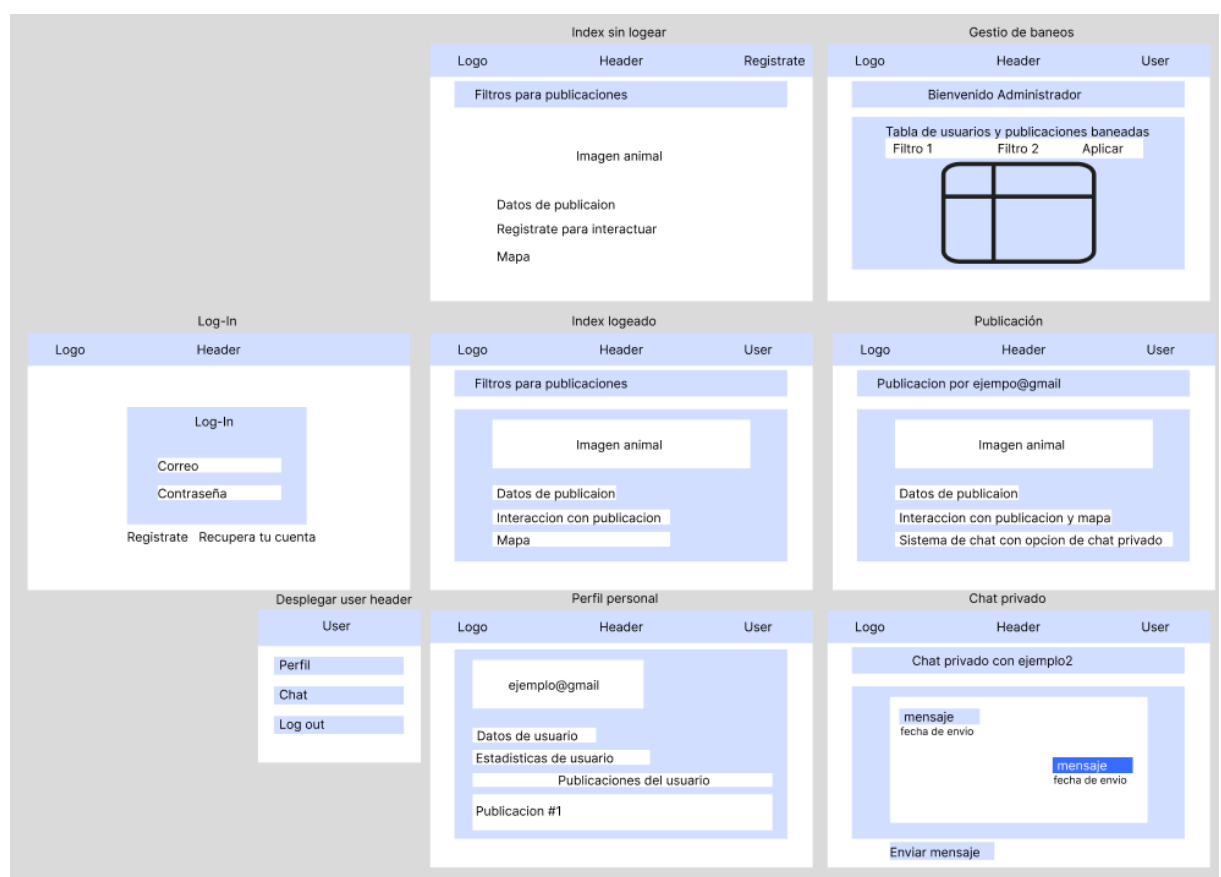
6.4 Diseño front-end

La interfaz de usuario de FindMe ha sido diseñada con el objetivo de proporcionar una experiencia clara, intuitiva y funcional tanto en ordenadores como en dispositivos móviles. Se ha utilizado HTML5, CSS3 y JavaScript para crear una estructura coherente y adaptable, sin depender de frameworks externos como Bootstrap.

La interfaz está organizada en bloques reutilizables, integrados mediante fragmentos PHP (header.php, footer.php, etc.), lo que permite mantener una apariencia uniforme y facilita el mantenimiento del código.

6.4.1 Mock-ups

En este apartado voy a mostrar cómo fue la primera idea de la página. Aunque básica y falto de detalles de este diseño surgió la aplicación final.



6.4.2 Guía de estilos

El diseño visual de este proyecto está construido directamente en cada fichero. Ficheros que se reutilizan como header.php también tiene esta estructura, así al usar el fichero también se usan sus estilos automáticamente.

En cuanto a los colores principales de la página he decidido no saturar de color el proyecto. He preferido usar color en puntos específicos para poder ayudar al usuario a distinguir bien sus opciones e incentivar el correcto uso del programa.

Los colores principales son el azul (en diferentes tonos) y blanco para los fondos. Aunque también he utilizado otros colores como el rojo para botones de denunciar, pero estos son la excepción.

Esta selección de colores le dan seriedad a la pagina y demuestran que lo mas importante es el objetivo de la pagina y no pasar el rato.

La tipografía se comparte en todas las paginas y se utilizan textos en negrita para resaltar algunos puntos.

La familia utilizada en los estilos es Arial, sans-serif, familia elegida por su forma fácilmente legible y seria.

La pagina completa está pensada para poder verse correctamente en cualquier dispositivo. Esto incluye desde ordenadores como en móviles.

El sistema utilizado para lograr que la pagina sea responsive es el uso de mediaquerys. Con su correcto uso he podido lograr que la página sea totalmente responsive.

La forma general de mostrar la información en la página se basa en tarjetas. Todas las tarjetas tienen la misma estructura y están correctamente señalados con class e id para poder implementar fácilmente el css.

Las tarjetas consisten en un contenedor padre con fondo blanco donde se pondrá otro contenedor dentro donde estructurar la información. Estos contenedores interiores también se separan en diferentes contenedores mas pequeños con el fin de darle un estilo y tamaño personalizado a cada uno.

6.4.3 Paquetes adicionales

Leaflet.js

- Función: visualización de mapas interactivos en el navegador.
- Uso: permite al usuario marcar la ubicación de una publicación en un mapa, así como ver dónde se ha perdido o encontrado un animal.
- Carga: mediante CDN externo (<https://unpkg.com/leaflet@1.9.3/dist/leaflet.js>).

jQuery

- Función: simplificar llamadas AJAX y manipulación del DOM.
- Uso: carga dinámica de comentarios, notificaciones, y algunas validaciones ligeras.
- Carga: a través de CDN (<https://code.jquery.com/jquery-3.6.0.min.js>).

AJAX

- Función: enviar/recibir datos sin recargar la página.
- Uso: actualización en tiempo real de comentarios, notificaciones, mensajes y validaciones.

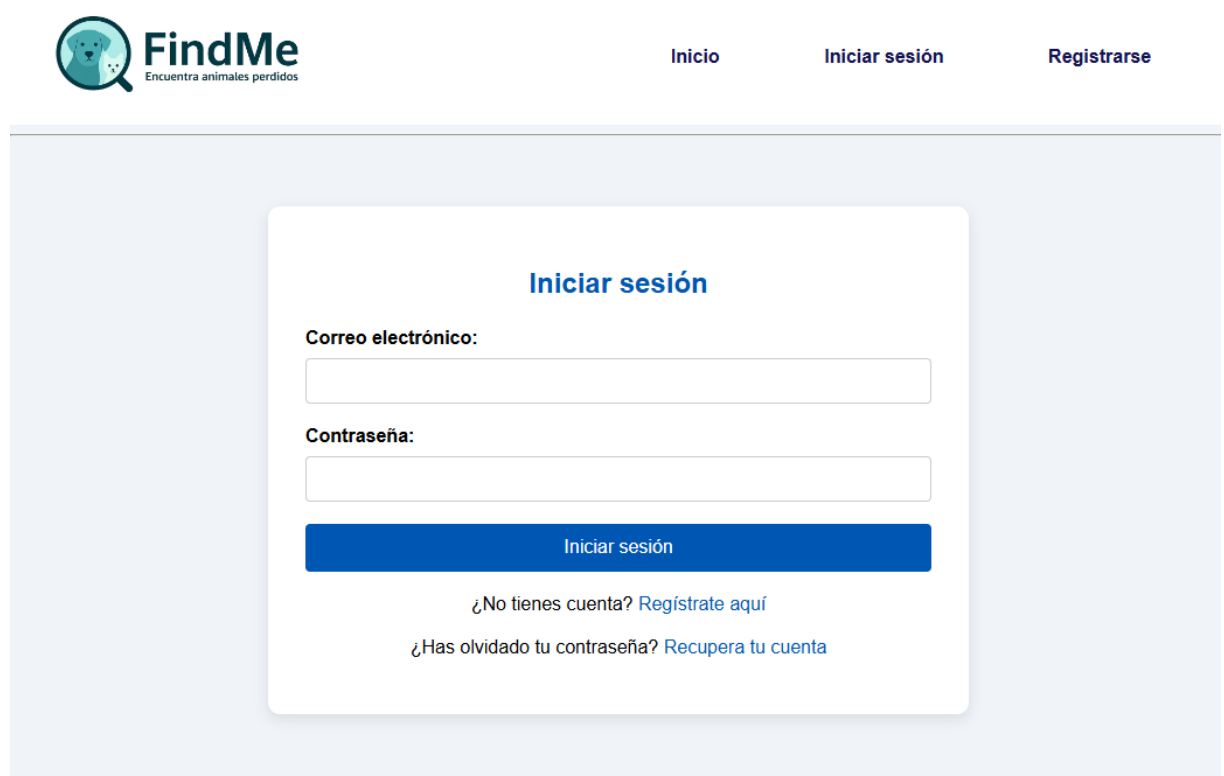
Otros recursos menores

- Íconos e imágenes SVG gratuitos para botones o secciones del sitio.
- CSS personalizado para mejorar accesibilidad y diseño sin frameworks.

6.4.4 Capturas de la aplicación

Para terminar esta sección del documento voy a mostrar ejemplos del diseño final del proyecto. Nombraré el fichero y el tamaño de pantalla de cada captura.

[login.php en PC](#)



FindMe
Encuentra animales perdidos

[Inicio](#) [Iniciar sesión](#) [Registrarse](#)

Iniciar sesión

Correo electrónico:

Contraseña:

[Iniciar sesión](#)

[¿No tienes cuenta? Regístrate aquí](#)

[¿Has olvidado tu contraseña? Recupera tu cuenta](#)

[login.php en movil](#)



Inicio

Iniciar sesión

Registrarse

Iniciar sesión

Correo electrónico:

Contraseña:

Iniciar sesión

¿No tienes cuenta? [Regístrate aquí](#)

¿Has olvidado tu contraseña? [Recupera tu cuenta](#)

**FindMe**
Encuentra animales perdidos[Inicio](#)[Nueva publicación](#)[Notificaciones](#)[root2@gmail.com](#) ▼[Ver publicaciones finalizadas](#)

Últimas publicaciones

Tipo de animal: -- Todos -- ▼Localidad: -- Todas -- ▼[Aplicar filtro](#)[Quitar filtros](#)**Tipo de animal:** perro**Texto:** Me he encontrado a este perro.**Relación con el animal:** Lo ha encontrado**Localidad:** ELCHE**Publicado por:** [root2@gmail.com](#)

👍 Up Vote: 0 | 👎 Down Vote: 0

**Tipo de animal:** gato**Nombre del animal:** Titan**Texto:** He perdido a mi gato. Se ll...**Relación con el animal:** Es su mascota**Localidad:** HONDON DE LAS NIEVES**Publicado por:** [root3@gmail.com](#)

👍 Up Vote: 0 | 👎 Down Vote: 0

**Tipo de animal:** perro**Texto:** Me he encontrado este perro...**Relación con el animal:** Lo ha encontrado**Localidad:** ELCHE**Publicado por:** [root@gmail.com](#)

👍 Up Vote: 0 | 👎 Down Vote: 0

[Index.php en Tablet](#)

[Inicio](#)[Nueva publicación](#)[Notificaciones](#)

R

[root2@gmail.com ▼](#)[Ver publicaciones finalizadas](#)

Últimas publicaciones

Tipo de animal: Localidad: [Aplicar filtro](#)[Quitar filtros](#)**Tipo de animal:** perro**Texto:** Me he encontrado a este perro.**Relación con el animal:** Lo ha encontrado**Localidad:** ELCHE**Tipo de animal:** gato**Nombre del animal:** Titan**Texto:** He perdido a mi gato. Se ll...**Relación con el animal:** Es su mascota**Localidad:** HONDON DE LAS NIEVES

[Index.php en Movil \(separado en 2 imagenes\)](#)

[Inicio](#)[Nueva publicación](#)[Notificaciones](#)[root2@gmail.com ▼](#)[Ver publicaciones finalizadas](#)

Últimas publicaciones

Tipo de animal:

-- Todos --



Localidad:

-- Todas --



Aplicar filtro

Quitar filtros



Tipo de animal: perro

Texto: Me he encontrado a este perro.

Relación con el animal: Lo ha encontrado

Localidad: ELCHE

Publicado por: root2@gmail.com


👍 Up Vote: 0 | 👎 Down Vote: 0

Fecha: 08/06/2025 12:25



[Ver detalles](#)

chat.php en PC

**FindMe**
Encuentra animales perdidos

[Inicio](#)

[Nueva publicación](#)

[Notificaciones](#)

R

root2@gmail.com ▼

Chat privado con root@gmail.com

[← Volver a mis chats](#)[⚠ Denunciar usuario](#)

Mensaje
07/06/2025 17:06

Mensaje
07/06/2025 17:06

Ejemplo de mensaje un poco mas largo largo largo largo largo largo largo
07/06/2025 17:55

Tú:
Ejemplo de respuesta.
07/06/2025 18:06

Tú:
Ejemplo de mensaje un poco mas largo largo largo largo largo largo largo
07/06/2025 18:09

Escribe tu mensaje aquí...

Enviar

[chat.php en Movil](#)[Inicio](#)[Nueva publicación](#)[Notificaciones](#)

R

[root2@gmail.com](#) ▼

Chat privado con root@gmail.com

[← Volver a mis chats](#)[⚠ Denunciar usuario](#)

07/06/2025 17:06

Ejemplo de mensaje un poco mas
largo largo largo largo largo largo
largo

07/06/2025 17:55

Tú:

Ejemplo de respuesta.

07/06/2025 18:06

Tú:

Ejemplo de mensaje un poco mas
largo largo largo largo largo largo
largo

07/06/2025 18:09

7. IMPLEMENTACIÓN

A continuación, voy a explicar como he implementado la parte del servidor y la parte de cliente del proyecto empezando por servidor. En estas explicaciones voy a mostrar partes del código relevantes para el funcionamiento.

También explicaré el orden de desarrollo y la intención de ciertas partes del código.

7.1 Servidor

La parte del servidor del proyecto FindMe se ha desarrollado utilizando PHP como lenguaje principal y MySQL como sistema de gestión de bases de datos. No se ha utilizado ningún framework.

El orden de desarrollo de la parte servidor es el siguiente:

- a. Base de datos
- b. Sistema de usuarios
- c. Sistema de publicaciones
- d. Sistema de comentarios
- e. Mensajes privados
- f. Denuncias y sanciones
- g. Notificaciones
- h. Panel de administración

El proyecto comenzó con la **base de datos**. La base de la aplicación es el correcto uso de la base de datos para poder utilizar cualquier página. El correcto uso de la base de datos también me ha servido a futuro para poder gestionar denuncias y sanciones, pero ahora voy a explicar las tablas más importantes.

Aquí está el SQL para crear una de las tablas más importantes, la tabla usuarios

```
CREATE TABLE usuarios (  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    correo VARCHAR(100),  
    telefono VARCHAR(20),  
    localidad VARCHAR(100),  
    contraseña VARCHAR(255),  
    fecha_registro TIMESTAMP DEFAULT current_timestamp,  
    likes_recibidos INT(11) DEFAULT 0,  
    dislikes_recibidos INT(11) DEFAULT 0,  
    comentarios_eliminados INT(11) DEFAULT 0,  
    baños INT(11) DEFAULT 0,  
    baño_fin DATETIME DEFAULT NULL,  
    PRIMARY KEY (id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Sin duda el principal y mas importante atributo de esta tabla y de toda la aplicación es el id de la tabla usuarios. Cada usuario tiene un id propio que en la base de datos es auto incrementable.

Este id servirá tanto para mostrar paginas privadas como para controlar que información posee y puede ver el usuario. En conclusión, sin este id el proyecto no podría funcionar por su vital importancia.

también quiero tratar la tabla publicaciones. Al igual que usuarios voy a mostrar como crear la tabla en la base de datos, y por los alter table añadidos se puede ver que durante el proceso de desarrollo he ido implementado datos a la tabla.

```
CREATE TABLE publicaciones (  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    usuario_id INT(11) NOT NULL,  
    texto TEXT NOT NULL,  
    localidad VARCHAR(100) NOT NULL,  
    tipo_animal ENUM('perro', 'gato', 'pajaro', 'reptil', 'otro') NOT NULL,  
    fecha_publicacion TIMESTAMP NOT NULL DEFAULT current_timestamp(),  
    denunciado TINYINT(1) NOT NULL DEFAULT 0,  
    PRIMARY KEY (id),  
    KEY usuario_id (usuario_id),  
    CONSTRAINT fk_publicaciones_usuario FOREIGN KEY (usuario_id)  
REFERENCES usuarios(id) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
ALTER TABLE publicaciones ADD COLUMN origen VARCHAR(20);
```

```
ALTER TABLE publicaciones ADD COLUMN nombre_animal VARCHAR(100);
```

```
ALTER TABLE publicaciones ADD COLUMN latitud DECIMAL(10, 8) DEFAULT  
NULL;
```

```
ALTER TABLE publicaciones ADD COLUMN longitud DECIMAL(11, 8) DEFAULT  
NULL;
```

```
ALTER TABLE publicaciones ADD COLUMN editada BOOLEAN NOT NULL  
DEFAULT 0;
```

Esta tabla es la segunda mas importante por que almacena toda la información de las publicaciones, además de asociar las publicaciones a los usuarios que la crean. Los campos latitud y longitud se han implementado para guardar correctamente el mapa de la publicación.

Estas dos tablas son las mas importantes para el proyecto. El resto de las tablas siguen una estructura centrada en relacionarse con las publicaciones y usuarios.

El **sistema de usuarios** aun sin ser demasiado complejo es de vital importancia como comenté antes. Un usuario consta de un email, un número de teléfono, una localidad y una contraseña.

Mediante el uso de estos datos y el id que se asigna automáticamente el usuario es capaz de visualizar correctamente todo el programa

El siguiente segmento de código muestra la importancia de el correcto uso de la base de datos. En concreto este fragmento está sacado de perfil.php, pero se puede ver bloques como este en todo el programa seleccionando la información necesaria para la pagina en particular.

```
$stmt = $pdo->prepare("

    SELECT correo, localidad, fecha_registro, likes_recibidos, dislikes_recibidos,
    comentarios_eliminados, baneos, baneo_fin

    FROM usuarios WHERE id = ?

");

$stmt->execute([$usuario_id]);

$usuario = $stmt->fetch();
```

Para concluir con el usuario tengo que mencionar que para controlar las acciones propias del administrador es necesario conectarse como un usuario concreto, y mediante condicionales permite acceder o no a ciertas páginas.

```
$es_root = false;

if (isset($_SESSION['usuario_id'])) {

    $stmt = $pdo->prepare("SELECT correo, localidad FROM usuarios WHERE id =
    ?");

    $stmt->execute([$_SESSION['usuario_id']]);

    $usuario_data = $stmt->fetch();
```

```
if ($usuario_data) {  
  
    $correo = $usuario_data['correo'];  
  
    $usuario_localidad = $usuario_data['localidad'];  
  
    if ($correo === 'root@gmail.com') {  
  
        $es_root = true;  
  
    }  
  
}  
  
}
```

El **sistema de publicaciones** primero depende de los usuarios, una vez que el usuario se identifica se le permite crear publicaciones, denunciar, comentar, editar y terminar una publicación si es dueño de ella.

Esto se controla mediante la id que tenga asociada la publicación, si la publicación y el usuario logeado coinciden en el id entonces el programa reconoce como dueño a ese usuario y le ofrece todas las opciones. En el caso contrario las opciones privadas del dueño se ocultarán y no podrán ser usadas por otro usuario.

Para poder mostrar la información de la publicación, tanto la creada da por el dueño como los comentarios, se utilizan consultas y se muestran los datos si no están denunciados.

El sistema actual de denuncia oculta la información denunciada con una sola denuncia, pero este apartado se ha mantenido por motivos de prueba y para poder mostrar el funcionamiento de la pagina sin tener que denunciar múltiples veces.

El **sistema de comentarios** funciona a partir de las publicaciones. Los comentarios son mensajes que otros usuarios pueden dejar en tu publicación.

Tu como dueño de una publicación tienes la capacidad de borrar comentarios (sumándole en 1 un contador que tiene cada usuario de mensajes borrados), iniciar chat privado y denunciar comentario (esta función está pensada para que cualquier persona pueda denunciar, la diferencia está en que denunciar algo no borra el contenido, solo lo oculta).

Aquí están los ejemplos de como se muestra la opción de iniciar chat:

```
<?php if (
    isset($_SESSION['usuario_id']) &&
    $_SESSION['usuario_id'] == $publicacion['usuario_id'] &&
    $_SESSION['usuario_id'] != $com['usuario_id']
): ?>

    <form method="POST" action="includes/crear_chat.php" style="display:inline;">

        <input type="hidden" name="publicacion_id" value="<?=
$publicacion['publicacion_id'] ?>">

        <input type="hidden" name="destinatario_id" value="<?= $com['usuario_id']
?>">

        <button type="submit" class="btn-chat"> Iniciar chat</button>

    </form>

<?php endif; ?>
```

En cuanto a los **mensajes privados** tienen una forma de uso particular. Solo puedes iniciar una conversación privada con alguien que haya comentado en tu publicación, además que tiene que iniciar la conversación privada el dueño del post comentado. Esto previene spam de usuarios que solo quieran husmear en la aplicación.

Una vez que un usuario comienza una conversación privada se crea un chat privado y se almacenará en la base de datos con un id personal de esta conversación y se almacena el id de ambos usuarios para gestionar casos de denuncia. Aquí está la tabla de los chats privados.

```
CREATE TABLE chats (  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    publicacion_id INT(11) NOT NULL,  
    creador_id INT(11) NOT NULL,  
    participante_id INT(11) NOT NULL,  
    fecha_creacion TIMESTAMP NOT NULL DEFAULT current_timestamp(),  
    PRIMARY KEY (id),  
    KEY fk_chat_publicacion (publicacion_id),  
    KEY fk_chat_creador (creador_id),  
    KEY fk_chat_participante (participante_id),  
    CONSTRAINT fk_chat_publicacion FOREIGN KEY (publicacion_id)  
REFERENCES publicaciones(id) ON DELETE CASCADE,  
    CONSTRAINT fk_chat_creador FOREIGN KEY (creador_id) REFERENCES  
usuarios(id) ON DELETE CASCADE,  
    CONSTRAINT fk_chat_participante FOREIGN KEY (participante_id)  
REFERENCES usuarios(id) ON DELETE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```


El **sistema de denuncias** funciona de la siguiente forma:

Un usuario denuncia un contenido o usuario. Cuando se denuncia cualquier contenido y se sobrepasa el límite de denuncias (actualmente 1).

En la carpeta includes, junto a otros ficheros que gestionan eventos, está el fichero procesar_denuncia.php. Este fichero gestiona las denuncias cuando alguien las confirma. El fichero diferencia el tipo de contenido denunciado y limita las denuncias de ese contenido para poder ocultarlo. La condición es la siguiente:

```
if ($total_denuncias >= 1) {

    $stmt = $pdo->prepare("UPDATE {$tabla} SET denunciado = 1 WHERE id = ?");

    $stmt->execute([$id_contenido]);

}
```

Por parte de **denunciar a usuarios**, solo se puede denunciar a un usuario si estás en un chat privado. Una vez denunciado al usuario el moderador tendrá la posibilidad de ver la conversación y los datos de ambos usuarios.

Una denuncia a un usuario no lo baneará, impedirá que ambos usuarios puedan mandarse mensajes privados y se mostrará el suceso en administrar denuncias.

En caht.php existe una condición (<?php if (!\$bloqueado): ?>) que controla los usuarios estan denunciados

```
<?php if (!$bloqueado): ?>

<form method="POST" action="includes/enviar_mensaje.php">

    <input type="hidden" name="chat_id" value="<?= $chat_id ?>">

    <input type="hidden" name="receptor" value="<?= $soy_creador ?
'participante' : 'creador' ?>">
```

```
<textarea name="mensaje" rows="3" required placeholder="Escribe tu
mensaje aquí..."></textarea>
```

```
<button type="submit">Enviar</button>
```

```
</form>
```

```
<?php endif; ?>
```

Esto consulta en la base de datos el estado de denuncia, si es 0 entonces se podrán enviar mensajes. En el momento que se denuncia se activa este código en el mismo fichero que activa la denuncia entre esos dos usuarios.

```
$stmt = $pdo->prepare("SELECT 1 FROM denuncias WHERE tipo_contenido =
'usuario' AND ((usuario_id = ? AND id_contenido = ?) OR (usuario_id = ? AND
id_contenido = ?))");
```

```
$stmt->execute([$usuario_id, $otro_usuario_id, $otro_usuario_id, $usuario_id]);
```

```
$bloqueado = $stmt->fetchColumn();
```

El sistema de **notificaciones** es simple. Cada publicación de cualquier tipo tiene un contador de visualizado, si es 0 avisa al usuario si es contenido relevante para él. El contenido relevante se caracteriza por estar situado en su área, por ser respuestas a su post o si son mensajes privados. Si el usuario entra en la publicación se marca como leída. Este es un ejemplo del código

```
if (isset($_GET['notificacion']) && isset($_SESSION['usuario_id'])) {
```

```
    $notif_id = $_GET['notificacion'];
```

```
    $stmt = $pdo->prepare("UPDATE notificaciones SET leida = 1 WHERE id = ?
```

```
AND usuario_id = ?");
```

```
$stmt->execute([$notif_id, $_SESSION['usuario_id']]);
```

```
}
```

Y por ultimo el **panel de administración**. Es una pagina donde solo puede acceder el administrador.

(archivo admin_denuncias.php)

```
if ($correo !== 'CORREO DEL ADMINISTRADOR') {
```

```
    die("Acceso denegado. No tienes permisos para ver esta página.");
```

```
}
```

(archivo index.php)

```
if (isset($_SESSION['usuario_id'])) {
```

```
    $stmt = $pdo->prepare("SELECT correo, localidad FROM usuarios WHERE id = ?");
```

```
    $stmt->execute($_SESSION['usuario_id']);
```

```
    $usuario_data = $stmt->fetch();
```

```
    if ($usuario_data) {
```

```
        $correo = $usuario_data['correo'];
```

```
        $usuario_localidad = $usuario_data['localidad'];
```

```
        if ($correo === 'CORREO DEL ADMINISTRADOR') {
```

```
            $es_root = true;
```

```
        }
```

```
    }
```

En la pagina de admin_denuncias el administrador podrá ver las denuncias activas donde podrá verlas y gestionar si eliminarlas o restaurarlas. A demás de ser capaz de ver las denuncias entre usuarios y poder acceder a una pagina anexa que es listar usuarios.

Listar usuarios funciona como un admin_denuncias para usuarios. Aquí puede ver y filtrar todos los usuarios y sancionar a los usuarios. Podrá poner una sanción automática que se va incrementando o podrá darle un tiempo personalizado. A parte de poder sancionar permanentemente a un usuario y quitarle el baneo.

Un usuario sancionado no podrá iniciar sesión con su cuenta y deberá esperar el tiempo indicado. Entonces ese usuario solo podrá ver la pagina sin iniciar sesión, lo que solo le permite ver el contenido de la pagina sin interactuar. Aquí está un ejemplo del código que muestra como un usuario no puede comentar nada.

```
<?php if (isset($_SESSION['usuario_id'])): ?>

    <hr>

    <h4>Dejar un comentario</h4>

    <form method="POST" action="includes/comentar.php">

        <textarea name="contenido" rows="3" cols="50" required></textarea><br><br>

        <input type="hidden" name="publicacion_id" value="<?=$publicacion['publicacion_id'] ?>">

        <button type="submit">Comentar</button>

    </form>

<?php else: ?>

    <p><a href="login.php">Inicia sesión</a> para comentar.</p>

<?php endif; ?>

</div>

</div>
```

7.2 Cliente

El apartado de cliente está desarrollado mediante el uso de tecnologías web estándar como: HTML5, CSS y JS.

La estructura de las paginas se desarrollan con el uso de HTML5 y con algunos fragmentos php que ayudarán a mostrar correctamente los datos.

En cuanto al apartado de estilos solo se ha utilizado CSS alojado en los propios ficheros. El diseño general del proyecto utiliza flexbox, grid y mediaquerys para mostrar correctamente todas las paginas para todas las pantallas.

Utilizo Ajax y jQuery para las funciones interactivas como mensajes, comentarios y notificaciones.

Una integración muy importante para el proyecto es la integración de mapas interactivos con Leaflet.js desde CDN. Estos mapas son funcionales y marcan la ubicación actual del usuario. Esta función junto con un JS básico añadido permite que el usuario pueda seleccionar la ubicación manualmente.

Esto permite que la aplicación no solo sirva para marcar la ubicación donde se encuentra el animal, sino que también puedes marcar la zona donde se ha perdido originalmente.

8. DESPLIEGUE

El proyecto FindMe ha sido desplegado en un entorno real utilizando los servicios de Hostinger, un proveedor de alojamiento web que permite ejecutar aplicaciones PHP con base de datos MySQL.

8.1 Modelo de despliegue utilizado

El modelo utilizado se trata de un modelo clásico basado en la subida de archivos vía FTP, la configuración manual de base de datos MySQL y ajustes de conexión en archivos de configuración PHP

Las características del despliegue son las siguientes:

- Servidor con soporte para PHP 7+
- MySQL gestionado mediante phpMyAdmin
- Almacenamiento compartido con espacio para archivos subidos por usuarios (carpeta uploads/)
- Acceso al hosting desde navegador o FTP

8.2 Datos iniciales y configuración

Realmente no es necesario precargar casi ningún dato ya que lo único que los únicos requisitos para que funcione la aplicación es que el código esté completo. Pero lo que si es necesario es crear un usuario administrador como se señala en el readme donde se señala como debe crearse tanto en la aplicación iniciada o desde la base de datos.

Realmente lo único necesario que exista en el proyecto es el usuario administrador, una imagen default para las publicaciones y una carpeta vacía que sirve para almacenar las imágenes subidas. Esta carpeta se llama uploads y debe estar dentro de la carpeta raíz del proyecto.

Para que a la hora de importar el proyecto no desaparezca la carpeta le he puesto un .txt avisando que no se debe eliminar la carpeta.

8.3 Pasos para el despliegue

- a. Importar todas las carpetas y ficheros del proyecto
- b. Exportar la base de datos proporcionada en el fichero sql en el proyecto.
- c. Cambiar el nombre de la base de datos o el usuario y contraseña si se necesita en el fichero /includes/db.php
- d. Verificar que las rutas sean validas y correctas. (En mi caso debía cambiar algunas rutas por no usar mayusculas).

8.4 Proveedores y servicios utilizados

- **Proveedor de hosting:** Hostinger
- **Dominio:** Actualmente no he pagado un dominio
- **Servicio de base de datos:** MySQL remoto (incluido en Hostinger)
- **Gestión de base de datos:** phpMyAdmin desde el panel del proveedor
- **Subida de archivos:** mediante **FTP** o gestor web
- **Mapa interactivo:** servicio externo **Leaflet.js** (CDN)
- **Scripts auxiliares:** jQuery (CDN)

No se han contratado servicios de cloud computing (como Amazon EC2, S3) ni certificados SSL de terceros.

9. HERRAMIENTAS UTILIZADAS

Durante el desarrollo del proyecto FindMe, se han empleado diversas herramientas que han facilitado las distintas fases del trabajo: desde el análisis y diseño hasta la programación, pruebas y despliegue final.

Análisis y documentación

- **Microsoft Word**

Para la redacción estructurada de la memoria del proyecto, incluyendo formato oficial, numeración y anexos.

- **Draw.io / Canva**

Para la elaboración de diagramas de arquitectura, modelo de datos y flujo de funcionamiento de la aplicación.

Desarrollo

- **Visual Studio Code**

Editor de código principal para trabajar con PHP, HTML, CSS y JavaScript. Permite organización por carpetas, resaltado de sintaxis, extensiones útiles para desarrollo web y conexión con FTP.

- **XAMPP**

Entorno de desarrollo local que incluye Apache, MySQL y phpMyAdmin. Se ha utilizado para probar toda la lógica del servidor antes del despliegue.

- **phpMyAdmin (local y remoto)**

Interfaz gráfica para la gestión de bases de datos. Se ha usado para crear, modificar y consultar las tablas, así como para exportar e importar datos.

Frontend

- **Leaflet.js**

Librería JavaScript para integrar mapas interactivos en la aplicación. Permite marcar ubicaciones en publicaciones.

- **jQuery**

Utilizado para simplificar llamadas AJAX y mejorar la manipulación del DOM en acciones como cargar comentarios, mensajes o notificaciones sin recargar la página.

Despliegue

- **Hostinger**

Plataforma de alojamiento web donde se ha desplegado la aplicación. Ofrece soporte para PHP y MySQL, así como acceso a FTP y gestor de archivos.

- **FileZilla**

Cliente FTP utilizado para subir archivos del proyecto al servidor de Hostinger de forma rápida y segura.

- **Navegadores (Chrome, Firefox)**

Usados para realizar pruebas visuales, comprobar diseño responsive y depurar errores mediante herramientas de desarrollador.

10. VIABILIDAD DEL PROYECTO

En esta sección voy a tratar la viabilidad del proyecto. Comentaré si el proyecto tiene potencial real y si es viable mantenerlo y continuar dándole soporte.

10.1 Viabilidad técnica

El proyecto FindMe es técnicamente viable, ya que se basa en tecnologías ampliamente utilizadas, bien documentadas y soportadas por la mayoría de los servidores web:

- Lenguaje del servidor: PHP
- Base de datos: MySQL
- Lado cliente: HTML5, CSS3, JavaScript
- Servicios externos: Leaflet.js, jQuery (CDN)
- Despliegue: Hostinger, con PHP y MySQL preinstalados

No requiere recursos especiales ni licencias, y puede ser instalado en cualquier entorno estándar de alojamiento compartido.

10.2 Viabilidad funcional

La aplicación cumple plenamente con los requisitos funcionales planteados al inicio:

- Gestión completa de usuarios
- Publicaciones con imágenes y ubicación
- Comentarios, filtros, denuncias y notificaciones
- Moderación con sanciones personalizadas
- Interfaz responsive y adaptativa

Todo el flujo funcional está implementado, probado y desplegado en entorno real, además de poder mejorar a futuro las funciones actuales y desarrollar nuevas funciones.

10.3 Posible uso real

FindMe podría ser utilizado como una plataforma real para localizar animales perdidos en comunidades locales:

- Fácilmente adaptable a asociaciones protectoras o ayuntamientos
- Interfaz accesible para usuarios no técnicos
- Posibilidad de integrarlo en redes vecinales o proyectos escolares

Su funcionamiento no depende de terceros, por lo que puede ser alojado de forma independiente por cualquier organización.

10.4 Propuestas de mejora

Algunas funcionalidades podrían ampliarse o perfeccionarse en el futuro:

- Sistema de recompensas para usuarios
- Estadísticas y paneles con datos avanzados
- Versión móvil como app
- Integración con redes sociales para mayor difusión

11. PREVENCIÓN DE RIESGOS LABORALES

Como ultimo punto a tratar en la documentación voy a hablar sobre los posibles riesgos laborales relacionados al proyecto y al sector laboral y como prevenirlos.

11.1 Riesgos asociados al desarrollo

El desarrollo del proyecto FindMe ha implicado principalmente trabajo frente al ordenador, por lo que los riesgos laborales se han centrado en los siguientes aspectos:

- **Riesgos ergonómicos:** posturas prolongadas, fatiga visual y carga muscular en cuello y espalda.
- **Riesgos psicosociales:** fatiga mental, presión por fechas de entrega o acumulación de tareas.
- **Riesgos eléctricos:** por el uso continuado de equipos conectados a red eléctrica.
- **Riesgos informáticos:** errores de seguridad en el código que podrían comprometer datos personales si el proyecto se expone públicamente.

11.2 Medidas de prevención adoptadas

Durante el desarrollo del proyecto, se han adoptado las siguientes medidas para minimizar los riesgos:

- **Pausas activas regulares** (cada 60–90 minutos) para reducir la tensión postural y ocular.
- **Uso de mobiliario adecuado**, con silla ergonómica y altura de pantalla ajustada.
- **Gestión del tiempo mediante planificación por fases**, para evitar sobrecargas en la recta final.
- **Respaldos de seguridad regulares** del código y base de datos para prevenir pérdida de trabajo.
- **Control de acceso a archivos sensibles** (como db.php) y buenas prácticas en el tratamiento de datos.

11.3 Buenas prácticas implementadas

Durante el desarrollo del proyecto FindMe, se han aplicado diversas buenas prácticas orientadas a reducir los riesgos laborales propios del trabajo prolongado frente al ordenador:

- **Ergonomía en el puesto de trabajo:**
 - Silla con respaldo ajustable y soporte lumbar.
 - Pantalla a la altura de los ojos y con iluminación ambiental adecuada.
 - Teclado y ratón posicionados para evitar tensión muscular.
- **Higiene postural y visual:**

- Descansos visuales cada 20 minutos (regla 20-20-20: mirar 20 segundos a 6 metros).
- Estiramientos breves al finalizar cada fase de trabajo o bloque de código.
- **Gestión del tiempo y del estrés:**
 - División del trabajo por objetivos semanales.
 - Revisión de tareas con antelación para evitar acumulaciones.
 - Desconexión activa durante pausas (evitando multitarea constante).
- **Ambiente de trabajo saludable:**
 - Ruido mínimo, espacio ordenado, ventilación adecuada y agua a disposición.
 - Uso de luz natural en la medida de lo posible.

Estas prácticas han favorecido tanto la calidad del trabajo como la salud física y mental durante todo el desarrollo del proyecto.

ANEXO 1. Bibliografía

- Mozilla Developer Network (s.f.). *Documentación HTML, CSS y JavaScript*. <https://developer.mozilla.org/es/>
- OpenStreetMap (s.f.). *Leaflet.js - Mapas interactivos*. <https://leafletjs.com/>
- PHP.net (s.f.). *Documentación oficial de PHP*. <https://www.php.net/manual/es/>
- W3Schools (s.f.). *Guías y referencias HTML/CSS/JS*. <https://www.w3schools.com/>
- jQuery Foundation (s.f.). *Documentación de jQuery*. <https://api.jquery.com/>
- Hostinger (s.f.). *Guía de despliegue en hosting compartido*. <https://www.hostinger.es/tutoriales/>
- XAMPP (s.f.). *Documentación oficial de XAMPP*. <https://www.apachefriends.org/es/index.html>