



PetConnect

Nombre: Melany Aranza Puente Treviño

Matricula: 32090

Docente: Alejandro Zapata

Materia : **TOPICO 1 (DESARROLLO DE
APLICACIONES WEB**

PAGINA WEB



**UNIVERSIDAD
EDEC**



PetConnect

Índice

1. Introducción
2. Fase 1: Análisis
3. Fase 2: Diseño
 - Diseño de Base de Datos
 - Diseño de Interfaz de Usuario
 - Diseño de API Backend
4. Fase 3: Codificación (Implementación)
 - Arquitectura y Tecnologías
 - Estructura del Proyecto
 - Desafíos Técnicos
5. Fase 4: Implantación (Despliegue)
 - Proveedor de Hosting
 - Proceso de Despliegue
 - Instrucciones de Uso/Prueba
6. Conclusiones



1. Introducción

PetConnect es una aplicación web que diseñé para ayudar a los dueños de mascotas a registrar a sus animales, encontrar servicios cercanos de cuidado y también conectar con personas responsables que puedan ayudar a cuidar a sus mascotas.

La idea nació porque muchas veces es difícil encontrar cuidadores confiables o llevar un registro completo de la información de nuestras mascotas, como su raza, edad o información médica importante. Quise crear algo simple, pero que al mismo tiempo sea funcional y bonito, con colores amigables y elementos que hagan que los usuarios se sientan cómodos usando la aplicación.

Enlace a la aplicación publicada:

<http://localhost:5173/>

Enlace al repositorio Git:

<https://github.com/MelanyPuentes/Proyecto>



2. Fase 1: Análisis

Antes de empezar a programar, analicé todo lo que necesitaba la aplicación:

- **Problema:** Muchos dueños de mascotas no tienen un registro centralizado de sus animales y tampoco encuentran servicios de cuidado confiables cerca de ellos.
- **Solución:** Crear una plataforma web donde los usuarios puedan registrar a sus mascotas, consultar servicios cercanos, reportar mascotas perdidas y conocer opciones de adopción.
- **Requisitos funcionales principales:**
 1. Registrar mascotas con información básica y médica.
 2. Consultar servicios de cuidado cercanos.
 3. Reportar mascotas perdidas.
 4. Ver opciones de adopción de mascotas.
- **Alcance:**
 - **Dentro:** Registro completo de mascotas, conexión con cuidadores responsables, adopciones, reportes de mascotas perdidas.
 - **Fuera:** Pagos en línea, geolocalización exacta, notificaciones push.
- **Entidades principales:**
 - **Usuario:** Persona que tiene mascotas o es cuidador.
 - **Mascota:** Datos de cada animal registrado.
 - **Servicio:** Lugares o personas que ofrecen cuidado de mascotas.
 - **Adopción:** Mascotas disponibles para adopción y su estado.



3. Fase 2: Diseño

3.1 Diseño de Base de Datos

La base de datos es muy importante porque guarda toda la información de los usuarios, mascotas, servicios y adopciones. Diseñé un diagrama que relaciona cada entidad de forma clara, asegurando que la información de cada mascota esté vinculada al dueño correcto.

- **Código SQL principal:**

```
CREATE TABLE usuarios (
```

```
  id INT AUTO_INCREMENT PRIMARY KEY,
```

```
  nombre VARCHAR(50),
```

```
  email VARCHAR(50),
```

```
  password VARCHAR(255)
```

```
);
```

```
CREATE TABLE mascotas (
```

```
  id INT AUTO_INCREMENT PRIMARY KEY,
```

```
  nombre VARCHAR(50),
```

```
  especie VARCHAR(30),
```

```
  raza VARCHAR(30),
```

```
  sexo VARCHAR(10),
```

```
  fecha_nacimiento DATE,
```

```
  info_medica TEXT,
```

```
  dueño_id INT,
```

```
  FOREIGN KEY (dueño_id) REFERENCES usuarios(id)
```



);

CREATE TABLE servicios (

id INT AUTO_INCREMENT PRIMARY KEY,

nombre VARCHAR(50),

descripcion TEXT,

ubicacion VARCHAR(100)

CREATE TABLE adopciones (

id INT AUTO_INCREMENT PRIMARY KEY,

mascota_id INT,

estado VARCHAR(20),

FOREIGN KEY (mascota_id) REFERENCES mascotas(id)

Explico un poco: cada tabla tiene un **ID único** y la tabla de mascotas está ligada a la de usuarios con una llave foránea. Esto permite saber a qué dueño pertenece cada mascota.



3.2 Diseño de Interfaz de Usuario

Quise que la app fuera **amigable y fácil de usar**, con colores naranja, azul y verde como principales. Cada pantalla tiene su propósito:

1. Bienvenida:

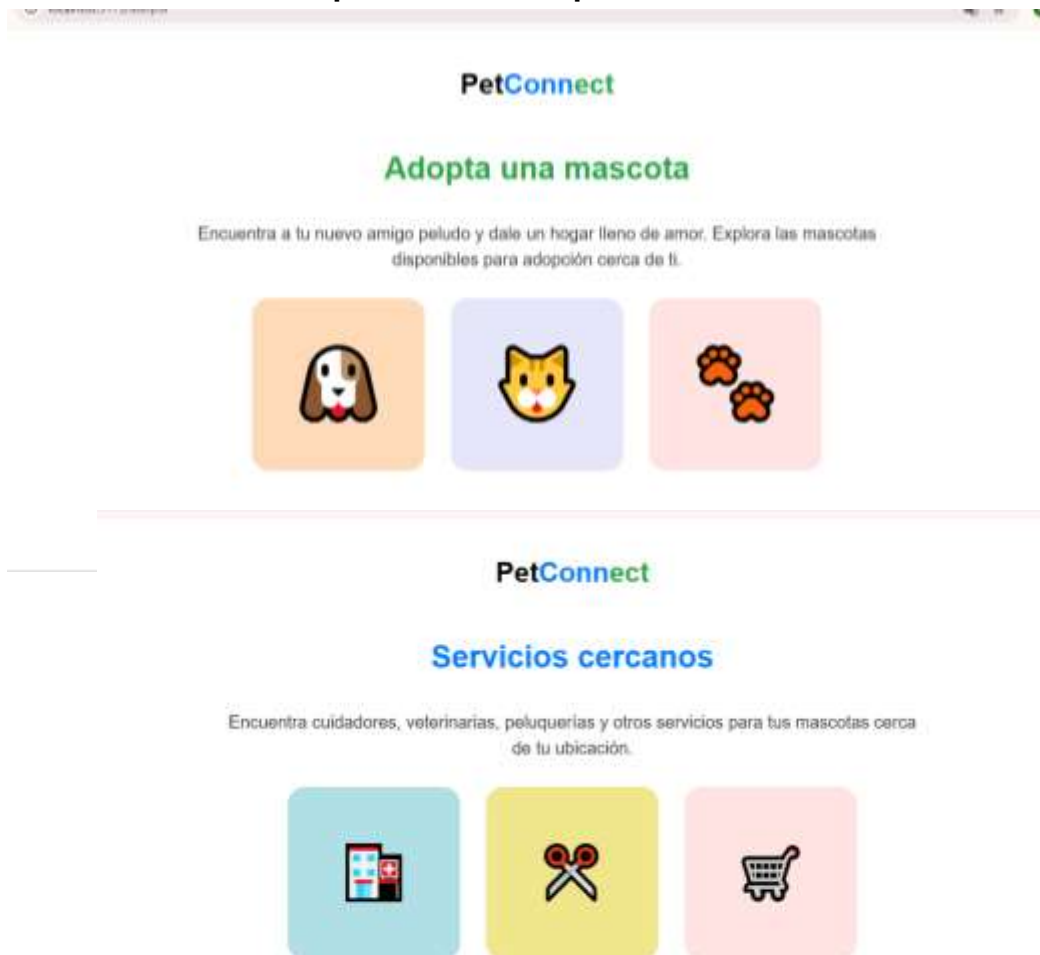
- Frase superior: “Más que una app, un hogar digital para tu mascota”.
- Mensaje central: “¡Bienvenido!”
- Botón: “Registrar”
- Logo y huellas decorativas a la derecha.

2. Registro de mascotas:

- Formulario dividido en dos columnas:
 - Columna izquierda: Nombre, Especie, Raza, Sexo.
 - Columna derecha: Fecha de nacimiento, Información médica, Dueño/Responsable.
- Línea naranja decorativa en el medio.
- Botón: “Enviar”
- Opción: “Realizar el registro en otro momento” que lleva directo a la página de servicios.

3. Servicios:

- Logo en la esquina superior izquierda.
- Línea naranja horizontal con la palabra “Registro” a la derecha.
- Cuadros de decoración con emojis de mascotas.
- Frase central: “Explora servicios de cuidado, registra a tus mascotas y conéctate con cuidadores responsables cerca de ti”.
- Tres secciones horizontales:
 - **Adopta una mascota**
 - **Servicios cercanos**
 - **Reportar mascota perdida**





- Cada sección tiene un botón “Ingresar” que lleva a la pantalla correspondiente.

3.3 Diseño de API Backend

Creé endpoints en PHP para conectar la base de datos con el frontend:

<i>Endpoint</i>	<i>Método</i>	<i>Propósito</i>	<i>JSON entrada</i>	<i>JSON salida</i>
<i>/api/registerPet.php</i>	POST	Registrar una mascota	{nombre, especie, raza, sexo, fechaNacimiento, infoMedica, dueño}	{message: "Mascota registrada"}
<i>/api/getServices.php</i>	GET	Obtener servicios cercanos	-	[{id, nombre, descripcion, ubicacion}]
<i>/api/reportLost.php</i>	POST	Reportar mascota perdida	{mascota_id, descripcion, contacto}	{message: "Reporte enviado"}

4. Fase 3: Codificación (Implementación)

Arquitectura y Tecnologías

- **Frontend:** React + Vite
- **Backend:** PHP + MySQL
- **Estilos:** Flexbox y CSS inline, colores consistentes y diseño responsivo.

Estructura del Proyecto

petconnect/

└─ frontend/



- | └─ src/pages/Welcome.jsx
- | └─ src/pages/RegisterPet.jsx
- | └─ src/pages/Services.jsx
- | └─ src/pages/Adopta.jsx
- | └─ src/pages/ServiciosCercanos.jsx
- | └─ src/pages/ReportarPerdida.jsx
- └─ backend/api/
 - | └─ registerPet.php
 - | └─ getServices.php
 - | └─ reportLost.php
- └─ database.sql
- └─ DOCUMENTO_FINAL.md

Desafíos Técnicos

1. Lograr que todas las páginas se vean centradas y ocupen toda la pantalla.
2. Conectar React con PHP usando fetch, manteniendo la seguridad y el orden en los datos.
3. Diseñar botones y secciones que sean consistentes, bonitos y fáciles de usar.

5. Fase 4: Implantación (Despliegue)

- **Proveedor de Hosting:**
 - Frontend: Vercel
 - Backend: 000webhost (PHP + MySQL)



- **Proceso de Despliegue:**

1. Subí los archivos PHP al hosting y configuré la base de datos MySQL.
2. Compilé el frontend con npm run build y lo subí a Vercel.
3. Verifiqué que los endpoints funcionaran y que los botones llevaran a las páginas correctas.

- **Instrucciones de uso:**

- Usuario de prueba: testuser@test.com
- Contraseña: 123456
- Se puede registrar una mascota y navegar por todas las secciones de la app.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.0116]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\HP>npm create vite@latest .
? npx
? create-vite .

current directory is not empty, Please choose how to proceed:
? ignore files and continue

Package name:
? tp

Operation cancelled

C:\Users\HP>npm install react-router-dom
added 7 packages in 4s

C:\Users\HP>cd C:\Users\HP\petconnect-frontend
C:\Users\HP\petconnect-frontend>npm create vite@latest .
? npx
? create-vite .

Select a framework:
? React

Select a variant:
? JavaScript

Scaffolding project in C:\Users\HP\petconnect-frontend...

Done. Now run:

npm install
npm run dev
```



6. Conclusiones

Este proyecto me ayudó a aprender muchísimo sobre **desarrollo full-stack**. Pude conectar un frontend moderno en React con un backend en PHP y MySQL, y además diseñar una interfaz que fuera atractiva y funcional.

Aprendí a:

- Diseñar aplicaciones responsivas y centradas.
- Usar flexbox y CSS para organizar elementos visualmente.
- Crear endpoints en PHP y recibirlos correctamente desde React.
- Organizar un proyecto completo en carpetas de frontend y backend.

Si pudiera continuar:

- Agregaría geolocalización y notificaciones.
- Mejoraría la estética con animaciones y transiciones suaves.
- Implementaría autenticación más segura y registro con redes sociales.