

## Informe del Proyecto Final

**Integrantes:** *Ruben Stiven Perdomo Carvajal*

**Curso:** *Lenguaje de programación 2*

**Fecha de entrega:** *6/10/2025*

### Aplicación Web React – “Petcare Vet”

**Tema:** Veterinaria (Catálogo de razas de perros con favoritos)

#### 1. Resumen ejecutivo

Se desarrolló una aplicación web educativa con **React + Vite** que permite **explorar razas de perros**, filtrar por nombre/temperamento y **marcar favoritas** con persistencia en localStorage. La app consume una **API pública (TheDogAPI)**, utiliza **React Router** para navegación entre páginas y aplica una **UI responsiva** con CSS. El objetivo fue practicar conceptos fundamentales de frontend: componentes funcionales, estado, efectos, fetch de datos y enrutamiento.

#### 2. Objetivo y alcance

- **Objetivo general:** Construir una app web sencilla y bien estructurada que demuestre fundamentos de desarrollo frontend.
- **Alcance:**
  - Listado de razas (catálogo) con imágenes y datos clave.
  - Búsqueda/filtrado en memoria.
  - Gestión de favoritos (persistente en el navegador).
  - Páginas de Inicio, Catálogo, Favoritos y Acerca.
  - Documentación y guía de ejecución.

#### 3. Requisitos y trazabilidad

Requisito	Cómo se cumplió
Interfaz interactiva y responsiva	Layout con <b>CSS Grid/Flex</b> , estilos modernos, tarjetas (cards), barra de búsqueda y feedback de carga/errores.
Componentes funcionales en React	NavBar, Card, Loader, ErrorBox y páginas (Home, Catalog, Favorites, About).
Consumo de API pública (opcional)	<b>Sí.</b> services/api.js usa <b>Axios</b> para <a href="https://api.thedogapi.com/v1/breeds">https://api.thedogapi.com/v1/breeds</a> y normaliza datos.
React Router (opcional)	<b>Sí.</b> Rutas: /, /catalogo, /favoritos, /acerca.
Estilos con CSS	src/styles.css con enfoque mobile-first y buen contraste.
Código bien estructurado	Estructura modular: components/, pages/, services/, hooks/.

#### 4. Arquitectura y tecnologías

- **Stack:** React 18, React Router 6, Axios, Vite.
- **Patrones y principios:** componentes presentacionales/reutilizables, separación de responsabilidades (**UI, servicios, estado**), manejo de efectos con `useEffect` y memoización con `useMemo`.
- **Estructura de carpetas:**

src/

components/

Card.jsx

ErrorBox.jsx

Loader.jsx

NavBar.jsx

hooks/

useLocalStorage.js

pages/

About.jsx

Catalog.jsx

Favorites.jsx

Home.jsx

services/

api.js

App.jsx

main.jsx

styles.css

#### 5. Diseño de UI/UX

- **Navegación clara:** encabezado con marca y enlaces activos.
- **Catálogo visual:** tarjetas con imagen, nombre y datos (vida, peso, altura), con botones de acción (favorito y “Más info”).
- **Accesibilidad básica:** roles ARIA en Loader/Error, etiquetas en inputs, contraste de colores.
- **Responsividad:** grillas adaptativas (auto-fill + minmax), barra superior sticky.

## 6. Funcionalidades implementadas

1. **Listado de razas:** consulta a TheDogAPI; normalización de campos (imagen, peso, altura, temperamento).
2. **Búsqueda/filtrado:** por nombre o temperamento en tiempo real (estado local + useMemo).
3. **Favoritos persistentes:** useLocalStorage (hook personalizado) para guardar IDs y rehidratarlos.
4. **Enrutamiento:** navegación SPA entre Inicio, Catálogo, Favoritos y Acerca.
5. **Manejo de estados:** loading, error y datos listos con feedback visual.
6. **Buenas prácticas:** componentes pequeños y reutilizables, manejo de errores, UI consistente.

## 7. Detalles técnicos clave

- **Consumo de API (Axios):** services/api.js centraliza la comunicación y **normaliza** los objetos.
- **Gestión de estado:** React Hooks (useState, useEffect) para ciclo de vida y carga; useMemo para optimizar filtrado.
- **Persistencia ligera:** localStorage para favoritos (sin backend).
- **Rendimiento:** filtrado en memoria; imágenes con object-fit: cover; uso de dependencias mínimas.

## 8. Ejecución y despliegue

### Ejecución local

npm install

npm run dev

Abrir la URL que muestra Vite (p. ej. <http://localhost:5173/>).

### Build de producción

npm run build

npm run preview

### Despliegue (opcional)

- **Vercel/Netlify:** build npm run build, directorio dist/.
- **GitHub Pages:** publicar carpeta dist (ajustar base en vite.config.js si es subruta).

## 9. Control de versiones (Git)

- Repositorio con ramas: main (estable) y ramas de funcionalidad (por ejemplo, feat/catalogo, feat/favoritos).
- Commits atómicos y descriptivos: feat(catalogo): agregar filtrado por temperamento.
- .gitignore para excluir node\_modules/ y artefactos de build.

## 10. Pruebas y calidad

- Pruebas manuales de flujo principal: carga del catálogo, búsqueda, marcado/quita de favoritos, navegación entre rutas.
- Validaciones visuales de responsividad (móvil, tablet, desktop).
- Manejo de errores de red simulado (desconectar/limitar red) para verificar ErrorBox.

## 11. Resultados

- La aplicación **cumple todos los requisitos** establecidos.
- Se logra una **experiencia fluida** con tiempos de respuesta inmediatos en filtrado y navegación.
- La **estructura modular** facilita el mantenimiento y la extensión futura.

## 12. Conclusiones y trabajo futuro

- **Conclusión:** El proyecto consolidó conceptos clave de React y frontend moderno, integrando fetch de API, router, estado y estilos en una base limpia y didáctica.
- **Mejoras futuras:**
  - Detalle de raza con routing dinámico (/catalogo/:id).
  - Tests automatizados (Jest + React Testing Library).
  - Modo oscuro/sistema de temas.
  - Paginación o carga diferida (lazy load) de imágenes.
  - Internacionalización (i18n) y accesibilidad ampliada.

## 13. Anexos (evidencias)

- Capturas sugeridas: Inicio, Catálogo (con búsqueda), Favoritos, About.

# Bienvenido a InfoPetVet

Recorre un catálogo amplio de razas de perros con fichas detalladas: origen, tamaño, temperamento, nivel de energía, compatibilidad con niños, cuidados, salud y consejos del día a día. Guarda tus favoritas y compáralas para encontrar la compañera perfecta.

Ver catálogo

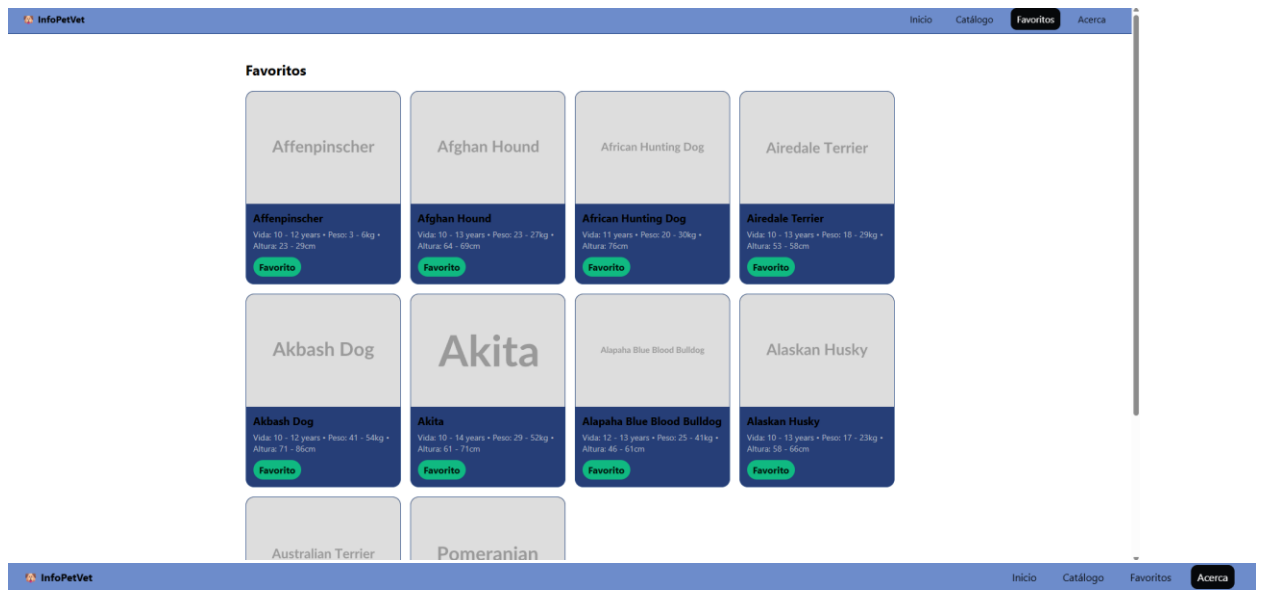
¿Cómo se hizo?

## Catálogo de Razas

Seleccionar colores y temperamento

172 resultados

<div>Affenpinscher</div> <div>Affenpinscher</div> <div>Vida: 10 - 12 years • Peso: 3 - 6kg • Altura: 23 - 29cm</div> <div><div>★ Quitar</div><div>Más info</div></div>	<div>Afghan Hound</div> <div>Afghan Hound</div> <div>Vida: 10 - 13 years • Peso: 23 - 27kg • Altura: 64 - 69cm</div> <div><div>★ Quitar</div><div>Más info</div></div>	<div>African Hunting Dog</div> <div>African Hunting Dog</div> <div>Vida: 11 years • Peso: 20 - 30kg • Altura: 75cm</div> <div><div>★ Quitar</div><div>Más info</div></div>	<div>Airedale Terrier</div> <div>Airedale Terrier</div> <div>Vida: 10 - 13 years • Peso: 10 - 29kg • Altura: 53 - 58cm</div> <div><div>★ Quitar</div><div>Más info</div></div>
<div>Akbash Dog</div> <div>Akbash Dog</div> <div>Vida: 10 - 12 years • Peso: 41 - 54kg • Altura: 71 - 85cm</div> <div><div>★ Quitar</div><div>Más info</div></div>	<div>Akita</div> <div>Akita</div> <div>Vida: 10 - 14 years • Peso: 29 - 52kg • Altura: 61 - 71cm</div> <div><div>★ Quitar</div><div>Más info</div></div>	<div>Alapaha Blue Blood Bulldog</div> <div>Alapaha Blue Blood Bulldog</div> <div>Vida: 12 - 13 years • Peso: 25 - 41kg • Altura: 46 - 61cm</div> <div><div>★ Quitar</div><div>Más info</div></div>	<div>Alaskan Husky</div> <div>Alaskan Husky</div> <div>Vida: 10 - 13 years • Peso: 17 - 23kg • Altura: 58 - 66cm</div> <div><div>★ Quitar</div><div>Más info</div></div>



### Acerca de la aplicacion web

- Interfaz interactiva y responsiva
- Componentes funcionales en React
- Consumo de API pública (TheDogAPI)
- React Router para navegación
- Estilos con CSS
- Código modular y estructurado

### Componentes de la pagina

```
src/
├── components/
│   ├── Card.jsx
│   ├── ErrorBox.jsx
│   ├── Loader.jsx
│   ├── NavBar.jsx
│   └── hooks/
│       └── useLocalStorage.js
├── pages/
│   ├── About.jsx
│   ├── Catalog.jsx
│   ├── Favorites.jsx
│   ├── Home.jsx
│   └── services/
│       ├── api.js
│       ├── App.jsx
│       ├── main.jsx
│       └── styles.css
```

Lenguaje de programacion 2.

- Enlace al repositorio GitHub.

[https://github.com/Ruberemix/Petcare\\_Vet.git](https://github.com/Ruberemix/Petcare_Vet.git)

Rubermix

Public

Pin

Watch 0

Fork 0

Star 0

main

1 Branch

0 Tags

Go to file

Add file

Code

Update README.md

d3ac15e · now

3 Commits

README.md

Update README.md

now

index.html

Add files via upload

1 minute ago

package-lock.json

Add files via upload

1 minute ago

package.json

Add files via upload

1 minute ago

vite.config.js

Add files via upload

1 minute ago

README

Aplicación web sencilla construida con React + Vite.

Tema y Objetivo

Tema: Veterinaria — Catálogo de razas de perros (con favoritos).  
Objetivo: Crear una interfaz interactiva y responsiva que consuma una API pública, gestione estado, use router para navegación y aplique buenas prácticas de estructura y estilos.

Requisitos Funcionales

1. Interfaz de usuario interactiva y responsiva

- Diseño adaptativo con CSS y grid.

2. Componentes funcionales en React

- Componentes: `Navbar`, `Card`, `Loader`, `ErrorBox`.

3. Consumo de API pública

- `TheDogAPI (/v1/breeds )` usando Axios en `services/api.js`.

4. React Router para navegación

Se desarrolló una aplicación web educativa con React + Vite que permite explorar razas de perros, filtrar por nombre/temperamento y marcar favoritas con persistencia en localStorage. La app consume una API pública (TheDogAPI), utiliza React Router para navegación entre páginas y aplica una UI responsiva con CSS.

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

HTML 69.1%

JavaScript 30.9%

Suggested workflows

Based on your tech stack