



# airbnb

## Proyecto Final

**DEV.F.**  
DESARROLLAMOS(PERSONAS);

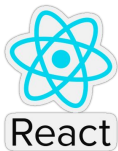
Elaborado por: Jesua Luján



[Jesua Hadai Luján](#)



dev



# Proyecto Final: Mini Clone Airbnb



*Objetivo:*



Construir una Single Page App (SPA) que simula una **plataforma estilo Airbnb** donde los usuarios puedan **registrarse, iniciar sesión y explorar propiedades ficticias.** 😈

Destino  
 Buscar destinos

Llegada  
 Agregar fec...

Salida  
 Agregar fec...

Huéspedes  
 ¿Cuántos?


## Alojamientos populares en Oaxaca >

Favorito entre huéspedes

Casa de huéspedes en Oaxaca  
 \$582 MXN por 2 noches ★ 4.86

Favorito entre huéspedes

Habitación en Oaxaca  
 \$699 MXN por 2 noches ★ 4.87

Favorito entre huéspedes

Habitación en Oaxaca  
 \$1,513 MXN por 2 noches ★ 4.92

Favorito entre huéspedes

Habitación en Oaxaca  
 \$608 MXN por 2 noches ★ 4.98

Favorito entre huéspedes

Loft en Oaxaca  
 \$1,241 MXN por 2 noches ★ 5.0

Favorito entre huéspedes

Loft en Oaxaca  
 \$2,305 MXN por 2 noches ★ 4.76

Favorito entre huéspedes

Habitación en Oaxaca  
 \$1,506 MXN por 2 noches ★ 4.94

## Disponibles el mes que viene en Ciudad de México >

Favorito entre huéspedes

Favorito entre huéspedes

Favorito entre huéspedes

Favorito entre huéspedes

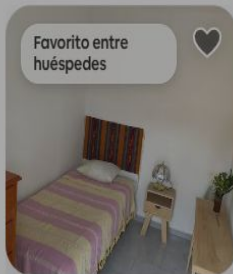
Favorito entre huéspedes

Favorito entre huéspedes

Favorito entre huéspedes

Destino  
Buscar destinos

## Alojamientos populares en Oaxaca >



Favorito entre huéspedes

Casa de huéspedes  
en Oaxaca

\$582 MXN por 2 noches ★ 4.86

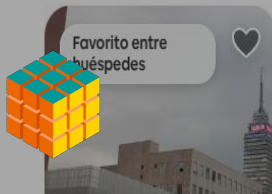


Favorito entre huéspedes

Habitación en Oaxaca

\$699 MXN por 2 noches ★ 4.87

## Disponibles el mes que viene en Ciudad de México



Favorito entre huéspedes



Favorito entre huéspedes



Inicia sesión o regístrate

¡Te damos la bienvenida a Airbnb!

Pais/Región

México (+52)



Número telefónico

Te vamos a confirmar el número por teléfono o mensaje de texto. Sujeto a tarifas estándar para mensajes y datos. [Política de privacidad](#)

Continuar

o



Continuar con Google



Continuar con Apple



Continuar con un correo electrónico



Continuar con Facebook



Favorito entre huéspedes

Loft en Oaxaca

\$2,305 MXN por 2 noches ★ 4.76



Favorito entre huéspedes

Habitación en Oaxaca

\$1,506 MXN por 2 noches ★ 4.94



Favorito entre huéspedes

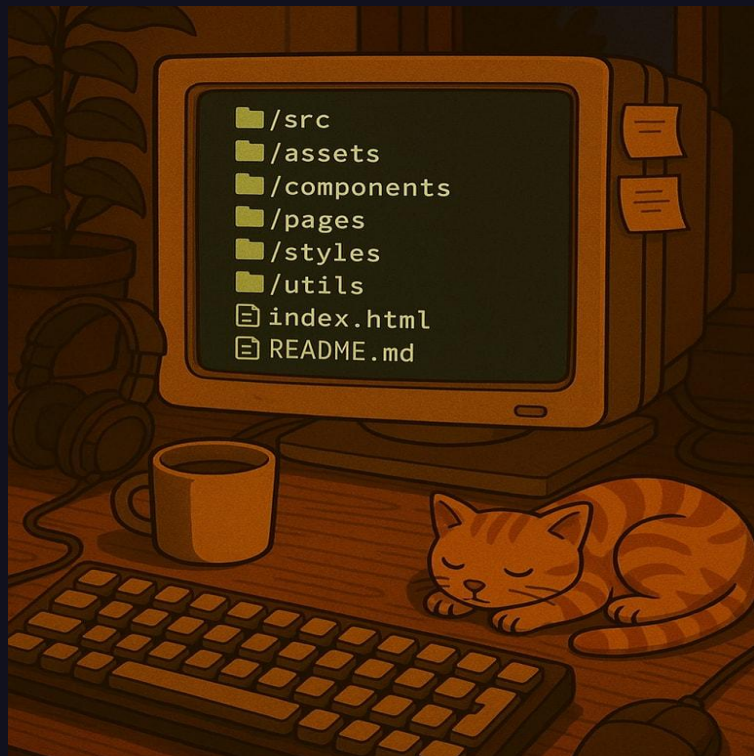


Favorito entre huéspedes





# | Arquitectura



DEV.F:

## 📁 Estructura sugerida del proyecto:



1. src/
2. | — assets/ # Imágenes o íconos
3. | — **components/**
4. | | — Navbar.jsx
5. | | — Card.jsx
6. | — **pages/**
7. | | — Home.jsx
8. | | — Login.jsx
9. | | — Signup.jsx
10. | | — Alojamientos.jsx
11. | | — Detalle.jsx
12. | — **routes/**
13. | | — AppRoutes.jsx
14. | — **utils/**
15. | | — auth.js
16. | — App.jsx
17. | — main.jsx





React + Vite with TailwindCSS



## 🧠 Tecnologías esperadas

1. React + Vite
2. react-router-dom
3. useState, useEffect, useNavigate
4. TailwindCSS
5. API simulada (puede ser JSON local con fetch, o algo como DummyJSON para adaptar propiedades)



## 🔧 Funcionalidades mínimas obligatorias

### ✓ 1. Inicio

- Ruta: "/"
- Bienvenida con call to action ("Explora alojamientos")
- Navbar con login/signup (si no hay sesión) o logout/perfil (si hay sesión)



## 🔧 Funcionalidades mínimas obligatorias

- ✓ 2. Simulación de Login/Signup
  - Formulario profesional usando Tailwind
  - Validaciones básicas
  - Guardar sesión en localStorage
  - Login falso: solo email y contraseña simulada





# 🔧 Funcionalidades mínimas obligatorias

## ✅ 3. Listado de Alojamientos

- Ruta: “/alojamientos”
- Consumir una API real o fake (**puede ser una simulación con fetch local o JSON externo**)

- Mostrar cards con:

Imagen

- **Título, breve descripción, imagen de la habitación**
- **Precio por noche**
- **Ubicación**

- Spinner mientras carga



## 🔧 Funcionalidades mínimas obligatorias

### ✓ 4. Detalle

- Ruta: “/alojamientos/:id”
- Mostrar **detalle completo** de la propiedad seleccionada

### ✓ 5. Rutas protegidas

- **Si no estás logueado** y visitas “/alojamientos” o “/alojamientos/:id”, **redirecciona a “/login”**

### ✓ 6. Navbar dinámica

- Si hay sesión: **"Mis Reservas", "Cerrar sesión"**
- Si no hay sesión: **"Iniciar sesión", "Registrarse"**



| 📺 Bonus (opcional pero recomendado) 🔥

DEV.F.



- Filtro por ciudad o precio
- Dark mode
- Animaciones con **Framer Motion** o **Animate.css**
- Scroll infinito o paginación
- Página de perfil



La fecha para entregar el proyecto y presentarlo en clases es la siguiente.

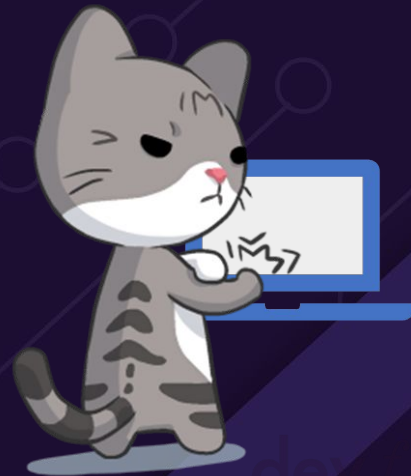
Fecha de entrega: **martes 1° de Julio.**





# Implementación

**DEV.F**  
DESARROLLAMOS(PERSONAS);



## | 🧠 1. utils/auth.js – Simulación de autenticación

Este archivo puede manejar las funciones para:

- Guardar sesión
- Obtener **usuario autenticado**
- Eliminar sesión
- (Opcional) **validar login simulado**

```
// Guarda el usuario en localStorage (simulando sesión)
Windsurf: Refactor | Explain | ✕
export const login = (userData) => {
  localStorage.setItem('user', JSON.stringify(userData));
};

// Elimina la sesión del usuario
Windsurf: Refactor | Explain | ✕
export const logout = () => {
  localStorage.removeItem('user');
};

| Ctrl+L to chat, Ctrl+K to generate
// Obtiene el usuario si existe en localStorage
Windsurf: Refactor | Explain | ✕
export const getUser = () => {
  const data = localStorage.getItem('user');
  return data ? JSON.parse(data) : null;
};

// Verifica si hay sesión activa
Windsurf: Refactor | Explain | ✕
export const isAuthenticated = () => {
  return !!localStorage.getItem('user');
};
```





## | 🧪 2. API simulada con JSON local + fetch

Vamos a usar fetch para cargar un JSON que imite una API como **DummyJSON** o AirBnb.

📁 **public/data/alojamientos.json**

**DEV.E.:**

```
[
  {
    "id": 1,
    "titulo": "Departamento moderno en CDMX",
    "ubicacion": "Roma Norte, Ciudad de México",
    "precio": 750,
    "imagen": "/assets/roma.jpg",
    "descripcion": "Moderno, céntrico y con balcón privado."
  },
  {
    "id": 2,
    "titulo": "Cabaña en el bosque",
    "ubicacion": "Valle de Bravo",
    "precio": 1200,
    "imagen": "/assets/cabana.jpg",
    "descripcion": "Relájate en una cabaña rústica entre los árboles."
  },
  {
    "id": 3,
    "titulo": "Casa colonial",
    "ubicacion": "San Miguel de Allende",
    "precio": 950,
    "imagen": "/assets/colonial.jpg",
    "descripcion": "Casa amplia con decoración tradicional y terraza."
  }
]
```



*nota: ustedes deben crear la data*

## ✓ Cómo usarlo con fetch

En tu componente

Alojamientos.jsx:

### ✓ Ventajas de este enfoque

- No requiere backend ni configurar json-server
- Simula fetch real
- Permite que los alumnos concentren esfuerzo en **frontend y lógica**
- Escalable: puedes agregar **/usuarios.json, /reservas.json, etc.**

```
import { useEffect, useState } from 'react';
Windsurf: Refactor | Explain | Generate JSDoc | X
const Alojamientos = () => {
  const [alojamientos, setAlojamientos] = useState([]);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    // Fetch al archivo local dentro de /public
    // -> si lo deseas puedes usar try{}catch({})
    fetch('/data/alojamientos.json')
      .then(res => res.json())
      .then(data => {
        setAlojamientos(data);
        setLoading(false);
      })
      .catch(err => {
        console.error('Error cargando alojamientos:', err);
        setLoading(false);
      });
  }, []);

  if (loading) return <p>Cargando alojamientos...</p>;

  return (
    <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6 p-6">
      {alojamientos.map(aloj => (
        <div key={aloj.id} className="border rounded shadow p-4 bg-white">
          <img src={aloj.imagen} alt={aloj.titulo} className="w-full h-48 object-cover mb-4 rounded" />
          <h3 className="text-xl font-bold">{aloj.titulo}</h3>
          <p className="text-gray-600">{aloj.ubicacion}</p>
          <p className="text-blue-600 font-semibold">${aloj.precio} / noche</p>
        </div>
      ))}
    </div>
  );
};

export default Alojamientos;
```



## I Paso a paso para comenzar

1. Crear tu proyecto usando el comando:

```
npm create vite@latest nombre_de_tu_proyecto -- --template react
```

2. Si usas tailwindcss (v.3.4)+ postcss:

```
npm install -D tailwindcss@3 postcss autoprefixer
```

```
npx tailwindcss init -p
```

