

Aplicación de Gestión Cártel Coches en VUE.js 3

1. Contexto del proyecto

El proyecto se basa en el caso real del **Cártel de Coches en España (2006–2013)**, donde fabricantes intercambiaron información para fijar precios, afectando a millones de usuarios.

La app pertenece a un permitirá **registrar clientes afectados, sus vehículos y gestionar el estado de sus reclamaciones**

2. Tipo de aplicación

- SPA en Vue.js 3
 - Desarrollada por parejas
 - Usando:
 - Componentes reutilizables
 - Vite + Vue 3
 - Git como control de versiones
 - Accesible y usable (web accesible)
-


3. Módulos principales de la aplicación

La aplicación debe tener **3 grandes bloques funcionales**:

3.1. Registro de Clientes Afectados

Formulario para dar de alta clientes con:

- DNI (único, validado con regex)
- Nombre y apellidos
- Teléfono
- Email
- Fecha de registro
- Estado de la resolución (pendiente, en trámite, resuelto, rechazado)
- Relación con uno o varios vehículos

 Cada cliente puede tener **uno o más coches asociados**.

3.2. Registro de Vehículos

Formulario para registrar coches afectados:

- **Matrícula** (única, validada con regex)
- **Marca** (select)
- **Modelo** (dependiente de la marca)
- **Año de matriculación**
- Otros datos:
 - Color
 - N° de puertas
 - Observaciones

📌 Cada vehículo debe estar vinculado a un cliente.

3.3. Panel de Control de Incidencias (Administrador)

- Existe **un solo administrador**
- Puede:
 - Ver todos los clientes
 - Ver sus vehículos
 - Modificar el **estado de resolución**
 - Gestionar incidencias

📌 El panel es un **dashboard** con:

- Listado de clientes
 - Filtros por estado
 - Edición del estado
-

4. Requisitos técnicos

4.1. Vue.js

- Uso de:
 - Componentes (`ClienteForm.vue`, `VehiculoForm.vue`, `Dashboard.vue`, etc.)
 - Props y emits
 - Reactividad (`ref`, `reactive`, `computed`)

- Vue Router para navegación
-

4.2. Reutilización de componentes

Ejemplos:

- InputText.vue
- SelectField.vue
- ButtonPrimary.vue
- ModalConfirm.vue



La app debe estar **bien estructurada por componentes reutilizables**.

4.3. Base de datos (simulada)



IDEA DEL PROYECTO:



Usar **Google Docs / Google Sheets como BBDD** (simulada) para almacenar datos

Alternativamente:

- JSON local
 - LocalStorage
 - API simulada
-

4.4. Control de versiones

- Repositorio en **GitHub**
 - Trabajo en pareja
 - Uso de:
 - Git Lens (VSCode)
 - Vue DevTools (navegador)
-

5. Accesibilidad y UX

La web debe ser:

- Navegable por teclado
- Con etiquetas `<label>` bien asociadas
- Contrastes adecuados
- Textos claros

6. Estructura recomendada del proyecto

```
src/
├── components/
│   ├── ClienteForm.vue
│   ├── VehiculoForm.vue
│   ├── ClienteList.vue
│   ├── Dashboard.vue
│   └── BaseInput.vue
├── views/
│   ├── Home.vue
│   ├── RegistroCliente.vue
│   ├── RegistroVehiculo.vue
│   └── AdminPanel.vue
├── router/
│   └── index.js
├── App.vue
└── main.js
```

7. Objetivo pedagógico

Con este proyecto el alumno aprende a:

- Crear una SPA realista en Vue 3
- Trabajar con formularios complejos
- Gestionar relaciones (cliente → vehículos)
- Reutilizar componentes
- Validar datos
- Trabajar en equipo con Git