

Módulo 6 – App de Clima (SPA con Vue.js)

1) Propósito

Recrear la App de Clima como una **aplicación de una sola página (SPA)** utilizando **Vue.js**, organizando la interfaz en componentes y utilizando templates, binding y eventos para trabajar de forma reactiva con los datos de clima ya disponibles en el proyecto (mock o API).

En esta iteración el foco está en la **interfaz interactiva y la navegación interna**, no en agregar nuevas llamadas a APIs ni nueva lógica de negocio compleja. Mantén la temática de lugares que elegiste (playas, montañas, planetas, rutas, etc.), pero ahora piensa tu app como un conjunto de **vistas y rutas** dentro de una SPA.

2) Objetivos de aprendizaje

- Describir y usar los **conceptos fundamentales de Vue.js** y componentes.
- Utilizar **templates y directivas de Vue** (`{{ }}`, v-for, v-if/v-show) para desplegar datos.
- Implementar **formularios interactivos** con v-model (two-way binding).
- Manejar **eventos** en Vue (@click, @submit, etc.) para responder a acciones del usuario.
- Implementar **navegación entre vistas** utilizando **Vue Router**.
- Gestionar el proyecto con **Git/GitHub** (commits descriptivos, README actualizado).

3) Alcanc

A partir de lo construido en los módulos anteriores (estructura de lugares, pronósticos y estadísticas):

- La app pasa a ser una **SPA en Vue** con al menos estas vistas:
 - **Home**: listado de lugares con su clima actual.
 - **Detalle de lugar**: información ampliada de un lugar (pronóstico y estadísticas que ya vienes trabajando).
- La navegación entre Home y Detalle debe hacerse **sin recargar la página**, usando Vue Router.
- Se debe incluir al menos una interacción extra en la interfaz, por ejemplo:
 - Búsqueda o filtrado de lugares.
 - Selección de unidad de temperatura (°C / °F).
 - Alguna pequeña opción de configuración.

Los datos pueden seguir viniendo de:

- La API integrada en el Módulo 5, o
- Datos mock definidos en JS,
según lo que se haya trabajado en el proyecto. Lo importante en este módulo es **cómo esos datos se muestran y navegan en Vue**.

4) Requisitos funcionales mínimos

- **Home (ruta inicial):**
 - Mostrar un listado de **≥ 5 lugares** con su información básica de clima.
 - Permitir seleccionar un lugar (click en card, botón o enlace) para ver su detalle.
- **Detalle de lugar (ruta dinámica):**
 - Mostrar información del lugar seleccionado (nombre, clima actual).
 - Mostrar el **pronóstico de varios días**.
 - Mostrar las **estadísticas de la semana** calculadas en módulos anteriores (min, max, promedio, etc.).
- **Interacción:**
 - Incluir al menos una interacción con formularios usando v-model (por ejemplo, buscar un lugar por nombre o cambiar de unidad).
- **Navegación:**
 - La app debe tener rutas diferenciadas para Home y Detalle, manejadas por **Vue Router**.
 - Debe ser posible volver desde el Detalle al Home (link, botón, etc.).
 -

5) Requisitos técnicos

Vue y componentes

- Utilizar **Vue.js** para construir la interfaz.
- Dividir la app al menos en:
 - Un componente raíz (App).
 - Un componente para **Home**.
 - Un componente para **Detalle de lugar**.
- Utilizar data() para el estado local de los componentes.

Templates, binding y eventos

- Utilizar:
 - **Interpolaciones** ({{ }}) para mostrar datos de lugares y clima.
 - **v-for** para renderizar listas (lugares, días del pronóstico).
 - **v-if / v-show** para mostrar/ocultar secciones según el estado (por ejemplo, mensajes de "No se encontró el lugar").
 - **v-model** para enlazar al menos un input de formulario (búsqueda, filtro, opción de configuración).
 - **Eventos** (@click, @submit, etc.) para manejar acciones del usuario.

Vue Router

- Configurar **Vue Router** para manejar al menos:
 - Una ruta para **Home** (/ o similar).
 - Una ruta **dinámica** para el detalle de cada lugar (por ejemplo, /lugar/:id).
- Utilizar enlaces de navegación de Vue (por ejemplo, <router-link>) y/o navegación programática para cambiar entre rutas.

6) Entregables

- Proyecto comprimido en un único archivo **.zip** que contenga:
 - Código fuente de la app en Vue (componentes, configuración de router, etc.).
 - Archivos de estilos (pueden reutilizar lo ya trabajado en módulos anteriores).
 - Cualquier archivo JS adicional que se utilice.
 - Carpeta de recursos (imágenes, íconos, etc.), si corresponde.
- Archivo **README.md** con:
 - Descripción breve de la App de Clima y su temática de lugares.
 - Listado de las **vistas principales** (Home, Detalle, y alguna otra si la hay).
 - Descripción de las **rutas** configuradas en Vue Router.
 - Instrucciones básicas para ejecutar el proyecto (por ejemplo, comandos si se usa Vue CLI).
 - **Enlace al repositorio público de GitHub.**

Rúbrica de evaluación

Criterio	Excelente (3 pts)	Adecuado (2 pts)	Básico (1 pt)	Insuficiente (0 pts)
Uso de Vue y componentes	La app funciona en Vue como SPA; hay componentes claros (App, Home, Detalle u otros) bien separados y usados correctamente.	La app funciona en Vue con componentes básicos, aunque algunos podrían estar mejor separados u organizados.	Se usa Vue de forma parcial; casi todo está en un solo componente o mezclado, pero se nota la intención.	No se usa Vue realmente o la app no funciona en Vue.
Templates, directivas y binding	Usa correctamente {{ }}, v-for, v-if/v-show y v-model; los datos se muestran y actualizan de forma reactiva y clara.	Usa varias directivas (listas, condicionales, binding) con algunos detalles menores o casos no cubiertos.	Uso limitado o incorrecto de directivas; parte importante del contenido sigue "fijo" en el template.	No se usan templates/directivas de Vue de forma significativa.
Vue Router y navegación	Hay rutas definidas para Home y Detalle (ruta dinámica); la navegación entre vistas funciona sin recargar la página.	Hay rutas y la navegación funciona en general, pero con algún detalle menor (por ejemplo, rutas poco claras).	Router configurado de forma muy básica o incompleta; la navegación es confusa o depende de recargas.	No se utiliza Vue Router o las rutas no funcionan.
Interactividad (eventos y formularios)	Se maneja al menos una interacción útil (@click, @submit, etc.) y un formulario con v-model (búsqueda, filtro, configuración) que afecta la vista.	Hay eventos y/o formulario con v-model, pero el impacto en la vista es parcial o no siempre claro.	Se usan eventos de forma mínima o poco relevante; casi no hay interactividad real.	No se manejan eventos ni formularios interactivos.
Git/GitHub y README	≥3 commits descriptivos; repo público; README explica la versión en Vue, las vistas/rutas y cómo ejecutar el proyecto, con enlace al repo.	≥3 commits aceptables; repo público; README básico con descripción y enlace.	Pocos commits o mensajes genéricos; README muy breve o incompleto.	Sin repo público/enlace; sin README o vacío.