

Parte A — Nuevas directivas (v-for, v-if, :class, :key)

Ejercicio 2 — Listado de productos con búsqueda

Objetivo: practicar `v-for`, `:key`, filtros y estados vacíos.

Requisitos:

- Lista local con `id`, `nombre`, `precio`, `stock`.
- Input de búsqueda (case-insensitive) que filtra por nombre.
- Estado vacío: mostrar mensaje si no hay resultados.
- Estilo/flag visual para `stock === 0`.
- Botón “Agregar al carrito” que emite un evento con el `id`.

Ejercicio 3 — Carrito mínimo con totales

Objetivo: eventos entre componentes y `computed`.

Requisitos:

- Escuchar el evento “add-to-cart(id)” y agregar ítems.
- Listar ítems del carrito con `cantidad`, `precio`, `subtotal`.
- Cálculo de total general con propiedad computada.
- Controles `+` / `-` para cantidad; eliminar cuando llegue a 0.
- Mensaje si el carrito está vacío.

Ejercicio 4 — Validación en el registro de clientes

Objetivo: `v-if/v-else` y clases dinámicas.

Requisitos:

- Si el nombre tiene `< 3` caracteres: mostrar error y deshabilitar botón.
- Al registrar: mostrar mensaje “Bienvenido, {{ nombre }}”.
- Borde/estilo del input cambia según validez.

Parte B — Integración de Vue Router

Objetivo

Separar la app en vistas: **Inicio**, **Productos**, **Detalle de producto**, **Clientes**.

Estructura mínima

- **Rutas:** / (inicio), /productos, /productos/:id, /clientes.
- **Ruta dinámica:** :id muestra el detalle del producto seleccionado.
- **Navegación:** enlaces de menú y navegación programática hacia el detalle.
- **Comportamiento:** volver desde el detalle a la lista; restaurar el scroll al navegar.

Criterios de aceptación del ruteo

- Todas las rutas existen y renderizan sus vistas correspondientes.
- La ruta /productos/:id recibe el id y muestra datos del producto.
- Enlaces activos destacados en la navegación.
- No hay errores de claves en listas (:key estable en todos los v-for).
- Estados vacíos/errores controlados (por ejemplo, "Producto no encontrado").