

Ejemplo Vue:

Crea una aplicación web utilizando **Vue.js** que permita a los usuarios gestionar una lista dinámica.

La aplicación debe cumplir los siguientes requisitos:

1. Mostrar una lista inicial de usuarios con nombres y un identificador único (NIF).
2. Permitir al usuario agregar un nuevo nombre a la lista a través de un campo de texto y un botón.
3. Asegurarse de que cada nuevo usuario agregado tenga un identificador único generado automáticamente.
4. Validar que el campo de entrada no esté vacío antes de agregar un nuevo usuario.
5. Mostrar la lista actualizada en tiempo real.

listaUsuarios.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Definimos la codificación de caracteres para el archivo HTML -->
  <meta charset="UTF-8">

  <!-- Hace que la página sea completamente responsiva (se adapta a diferentes
  tamaños de pantalla) -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <!-- Título que aparecerá en la pestaña del navegador -->
  <title>Lista de Usuarios con Vue</title>

  <!-- Incluimos Vue.js desde un CDN para facilitar la implementación -->
  <script src="https://unpkg.com/vue@3"></script>

  <!-- Estilos internos para la página -->
  <style>
    /* Estilos básicos para la lista de usuarios */

    /* Eliminamos los puntos de lista y el relleno de los <ul> */
    ul {
      list-style-type: none;
      padding: 0;
    }

    /* Estilos para cada elemento <li>, como el color de fondo y los márgenes */
    li {
      background: #f0f0f0;
```

```

    margin: 5px 0;
    padding: 10px;
    border-radius: 5px;
  }
</style>
</head>
<body>
  <!-- Este es el div donde Vue manejará el contenido dinámico -->
  <div id="app">
    <h1>Lista de Usuarios</h1>
    <!-- Campo para ingresar un nuevo usuario -->
    <input v-model="nuevoNombre" placeholder="Ingresa el nombre" />
    <button @click="agregarUsuario">Agregar Usuario</button>

    <!-- Lista dinámica -->
    <ul>
      <li v-for="usuario in usuarios" :key="usuario.nif">
        {{ usuario.nombre }} (NIF: {{ usuario.nif }})
      </li>
    </ul>
  </div>

  <!-- Script para Vue -->
  <script>
    const app = Vue.createApp({
      data() {
        return {
          usuarios: [ // Lista inicial de usuarios
            { nombre: "Laura Pérez", nif: "12345A" },
            { nombre: "Carlos García", nif: "67890B" }
          ],
          nuevoNombre: "" // Para capturar el nuevo nombre ingresado
        };
      },
      methods: {
        agregarUsuario() {
          if (this.nuevoNombre.trim() !== "") { // Validación del campo
            const nuevoUsuario = {
              nombre: this.nuevoNombre,
              nif: Math.random().toString(36).substring(2, 10) // Generar un NIF único
            };
            this.usuarios.push(nuevoUsuario); // Agregar a la lista
            this.nuevoNombre = ""; // Limpiar el campo
          } else {
            alert("El nombre no puede estar vacío."); // Mensaje de validación
          }
        }
      }
    });
  </script>

```

```

    }
  });

  app.mount('#app'); // Montar Vue en el div con id="app"
</script>
</body>
</html>

```

Vue.js

```

<script>
const app = Vue.createApp({
  data() {
    return {
      usuarios: [ // Lista inicial
        { nombre: "Laura Pérez", nif: "12345A" },
        { nombre: "Carlos García", nif: "67890B" }
      ],
      nuevoNombre: "" // Para capturar el nuevo nombre ingresado
    };
  },
  methods: {
    agregarUsuario() {
      if (this.nuevoNombre.trim() !== "") { // Validación del campo
        const nuevoUsuario = {
          nombre: this.nuevoNombre,
          nif: Math.random().toString(36).substring(2, 10) // Generar un NIF único
        };
        this.usuarios.push(nuevoUsuario); // Agregar a la lista
        this.nuevoNombre = ""; // Limpiar el campo
      } else {
        alert("El nombre no puede estar vacío."); // Mensaje de validación
      }
    }
  }
});

app.mount('#app'); // Montar Vue en el div con id="app"
</script>

```

¿Qué hace este código?

1. **Vue.js** está manejando el estado de los usuarios (en el array `usuarios`), que inicialmente contiene dos usuarios.
2. **v-model** se usa en el campo de entrada para vincular el valor de `nuevoNombre` al campo de texto.
3. **@click="agregarUsuario"** llama al método `agregarUsuario` cuando se hace clic en el botón, lo que añade un nuevo usuario con un nombre generado y un NIF único.
4. La lista de usuarios se muestra dinámicamente en una lista ``, usando **v-for** para iterar sobre los usuarios en el array y mostrar su nombre y NIF.