



# Renderizado condicional y de listas

Inmaculada Rodríguez Santiago

Jordi Barea Colomer

Factors Humans i Computació, 22-23

# v-if

<https://es.vuejs.org/v2/guide/conditional.html>

- La directiva v-if permite renderizar un elemento HTML si es true el resultado.

```
<h1 v-if="ok">Sí</h1>
```

```
data () {  
  return {  
    ok: true  
  };  
}
```

- Si *ok* es true en la propiedad data de la aplicación Vue, el <h1> se renderiza
- Si *ok* es false, el <h1> no se renderiza

# v-if

<https://v3.vuejs.org/guide/conditional.html#conditional-groups-with-v-if-on-template>

```
<template v-if="ok">
  <h1>Title</h1>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</template>
```

Nota: v-if se aplica a un solo elemento, para aplicar a más elementos, se deben incluir dentro de un `<template>`

HTML

El **elemento HTML** `<template>` sirve para indicar contenido [HTML](#) del lado del cliente que no se renderiza cuando se carga una página, pero que posteriormente puede ser instanciado durante el tiempo de ejecución empleando JavaScript.

# v-if v-else

<https://es.vuejs.org/v2/guide/conditional.html>

- La directiva **v-if** permite renderizar un elemento HTML si es true el resultado de su evaluación, en caso contrario, se renderiza el elemento del **v-else**

```
<div v-if="Math.random() > 0.5">  
  Ahora me ves  
</div>  
<div v-else>  
  Ahora no  
</div>
```

Un elemento v-else debe seguir inmediatamente a un elemento v-if o v-else-if.

# v-if ... v-else

<https://es.vuejs.org/v2/guide/conditional.html>

- Ejemplo

```
<div id="no-key-example" class="demo">
  <div>
    <template v-if="loginType === 'username'">
      <label>Nombre de Usuario</label>
      <input placeholder="Ingrese su nombre de usuario">
    </template>
    <template v-else="">
      <label>Email</label>
      <input placeholder="Ingrese su email">
    </template>
  </div>
  <button @click="toggleLoginType">Alternar tipo de Login</button>
</div>
```

Nombre de Usuario

Alternar tipo de Login

```
const app = Vue.createApp({
  data: {
    loginType: 'username'
  },
  methods: {
    toggleLoginType: function () {
      return this.loginType = this.loginType
        === 'username' ? 'email' : 'username';
    }
  }
});
```

# v-else-if

<https://es.vuejs.org/v2/guide/conditional.html>

```
<div v-if="type === 'A'">
  A
</div>
<div v-else-if="type === 'B'">
  B
</div>
<div v-else-if="type === 'C'">
  C
</div>
<div v-else>
  Si no es A, B o C
</div>
```

En el ejemplo, se renderiza el contenido del primer `<div>` en el cual el resultado de evaluar `v-if` es true

`type` es un dato de la aplicación Vue:

```
data () {
  return{
    type: "C"
  };
}
```

# v-show

<https://v3.vuejs.org/guide/conditional.html#v-show>

- v-show funciona de forma similar a v-if. Solo que NO puede haber un v-else y NO funciona con <template>

```
<h1 v-show="ok">Hello!</h1>
```

# Renderizado de listas con v-for

<https://v3.vuejs.org/guide/list.html#v-for-with-a-range>

- Sirve para representar una lista de elementos basada en un **array**.

Sintaxis de v-for: “item in items”, donde items es la matriz de datos de origen y item es un alias para el element del array por el que se está iterando

```
<ul id="example-1">
  <li v-for="item in items">
    {{ item.mensaje }}
  </li>
</ul>
```

- Foo
- Bar

```
const app = new Vue.createApp({
  data() {
    return {
      items: [
        { mensaje: 'Foo' },
        { mensaje: 'Bar' }
      ]
    }
  }
});
app.mount("#example-1");
```

Otro ejemplo en: <https://v3.vuejs.org/guide/list.html#v-for-with-a-range>



# Renderizado de listas con v-for

<https://es.vuejs.org/v2/guide/list.html>

- Con un **objeto**  
(se utiliza value para referirse a cada valor del objeto)

```
<ul id="v-for-object" class="demo">
  <li v-for="value in object">
    {{ value }}
  </li>
</ul>
```

HTML

```
new Vue({
  el: '#v-for-object',
  data: {
    object: {
      primerNombre: 'John',
      apellido: 'Doe',
      edad: 30
    }
  }
})
```

JS

- John
- Doe
- 30

# v-for en un template

<https://v3.vuejs.org/guide/list.html#v-for-on-a-template>

- Se puede usar la etiqueta <template> para mostrar un bloque de varios elementos

```
<ul>
  <template v-for="item in items" :key="item.msg">
    <li>{{ item.msg }}</li>
    <li class="divider" role="presentation"></li>
  </template>
</ul>
```

# Renderizado de listas con v-for

<https://v3.vuejs.org/guide/list.html#maintaining-state>

- Recomendación: usar un atributo *key* con v-for porque ayuda al virtual DOM de Vue a saber si hay cambios desde la última vez que se recorrió la lista. <https://vueschool.io/articles/vuejs-tutorials/tips-and-gotchas-for-using-key-with-v-for-in-vue-js-3/>
- El key debe ser un string o un número, no un objeto.

```
<div v-for="item in items" :key="item.id">  
  <!-- content -->  
</div>
```

# v-for con v-if

<https://v3.vuejs.org/guide/list.html> - v-for-with-v-if

- Recomendación: NO usar v-for con v-if

```
<li v-for="todo in todos" v-if="!todo.isComplete">
  {{ todo.name }}
</li>
```

Lanzará un error, porque *v-if* se evaluará primero (es más prioritaria que v-for) , por lo que el dato **todo** no existe en este momento.

- Solución:

```
<template v-for="todo in todos" :key="todo.name">
  <li v-if="!todo.isComplete">
    {{ todo.name }}
  </li>
</template>
```

# Ejercicio 1

Bajar código base del CV EjercicioListaTareasRenderizadoCondicionalCodigoBase

Inicio:

**Lista de tareas**

Añadir tarea

sss

Resultado al añadir botones “Mostrar tareas/Ocultar tareas”  
y “Eliminar tareas”

**Lista de tareas**

Añadir tarea

Ocultar tareas

Eliminar tareas

ssss

aaa

**Lista de tareas**

Añadir tarea

Mostrar tareas

Eliminar tareas

# Ejercicio 2

Bajar código base del CV [EjercicioFiltroBusquedaCodiBase.zip](#)

Resultado final añadiendo la funcionalidad de filtrado usando el método [filter](#) de js:

Inicio: (sin filtrado)

## Ejemplo de filtrado con Vue

camis	
Producto	Precio
Camiseta corta	10
Camiseta larga	25
Camisola	20
Chaqueta	200
Chubasquero	100

## Ejemplo de filtrado con Vue

camis	
Producto	Precio
Camiseta corta	10
Camiseta larga	25
Camisola	20

The JS [filter\(\)](#) method of creates an array filled with all array elements that pass a test (provided as a function)