



Parte 2 - Pasos en la creación de una aplicación básica en Vue

Inmaculada Rodríguez Santiago
Factors Humans i Computació, 22-23

Pasos en la creación de una aplicación básica en Vue

- Paso 1: settings y crear app
 - Incluir .js de Vue y nuestro app.js en <head>
 - Crear la aplicación Vue (app.js) diciendo qué parte de HTML controlará.
- Paso 2:
 - Configurar datos (data) y lógica (methods) en la app Vue
- Paso 3:
 - Enlazar el HTML con datos en Vue a través de directivas v-
- Paso 4:
 - Llamar a métodos Vue desde HTML

Ejercicio 1

(En 4 pasos)

Bajar código base del CV:
HelloWorldConVue-
ConDatosCodiBase.zip

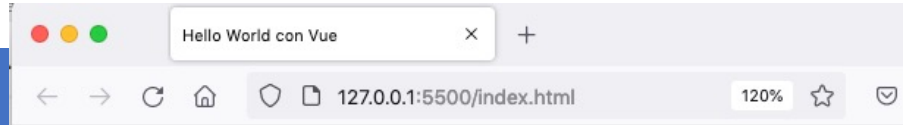
index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Hello World con Vue</title>
7     <link rel="stylesheet" href="styles.css" />
8     <!--añadir links de Vue y tu .js -->
9   </head>
10  <body>
11    <div id="app">
12      <h1></h1>
13      <p><a href=""></a></p>
14    </div>
15  </body>
16 </html>
17
```

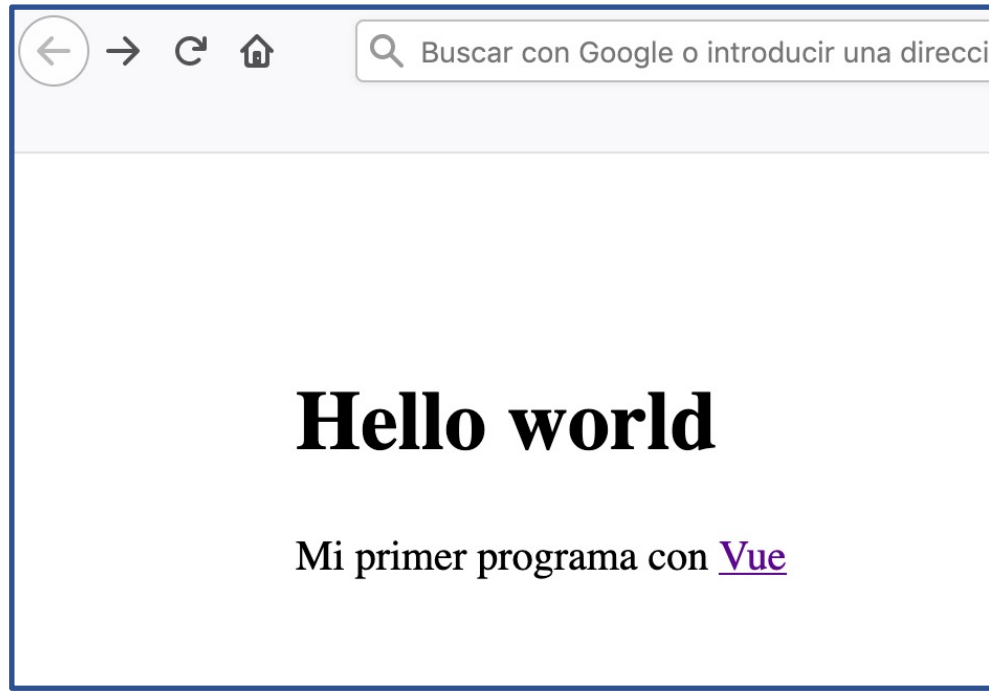
app.js

```
1
2
3
4
5
```

Al inicio (código base ejercicio1)



Al final



Paso 1/4 – Settings y crear la aplicación Vue I

En el <head> del documento HTML añadir scripts:

- Código de Vue:

```
<script src="https://unpkg.com/vue@next" defer></script>
```

- Tu fichero javascript, app.js

```
<script src="./app.js" defer></script>
```

Paso 1/4 – Settings y crear la aplicación Vue II

Crear la aplicación Vue que “quiere” controlar nuestra pagina HTML (*createApp*) y decir a Vue qué parte de la pagina HTML será controlada (*mount*).

index.html

```
...  
<div id="app">  
  <h1>..</h1>  
  <p>...</p>  
</div>  
...
```

app.js

```
const app =  
Vue.createApp( {  
  ...  
} );  
  
app.mount( "#app" );
```

Paso 2/4 – Configurar datos y lógica en Vue

Configurar los **datos** (propiedad ***data***) y **lógica** de la aplicación Vue (propiedad ***methods***) a través del objeto que le pasamos a *createApp*.

index.html

```
...  
<div id="app">  
  <h1>..</h1>  
  <p>...</p>  
</div>  
...
```

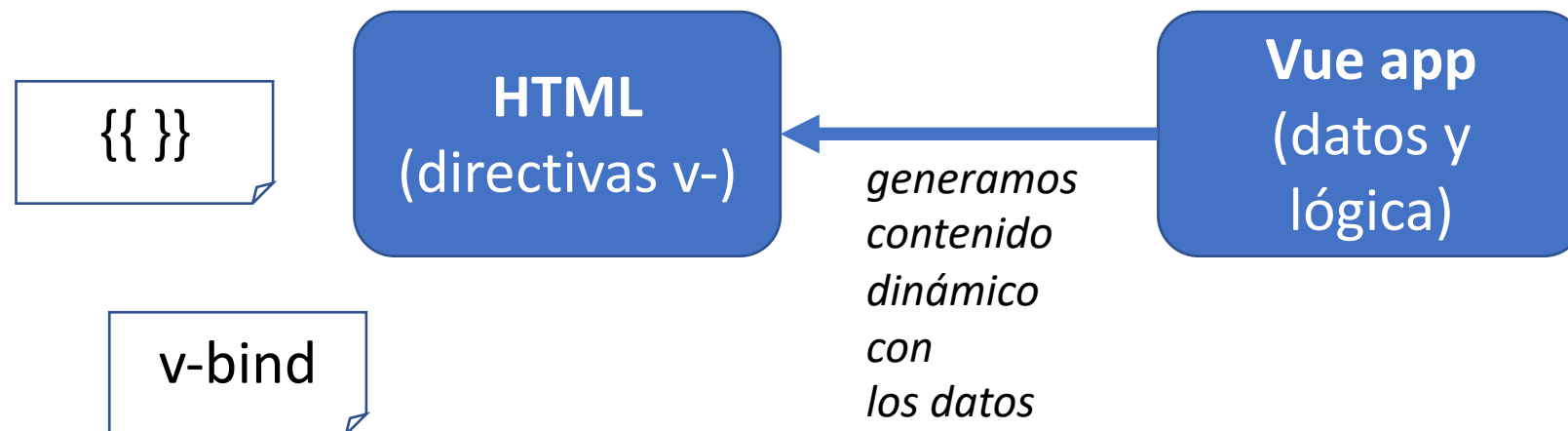
app.js

```
const app = Vue.createApp({  
  data: function () {  
    return {  
      mensaje: 'Hello world'  
    };  
  },  
  methods: {  
  
  }  
});  
app.mount("#app");
```

Paso 3/4 - Generar contenido y vincular atributos I

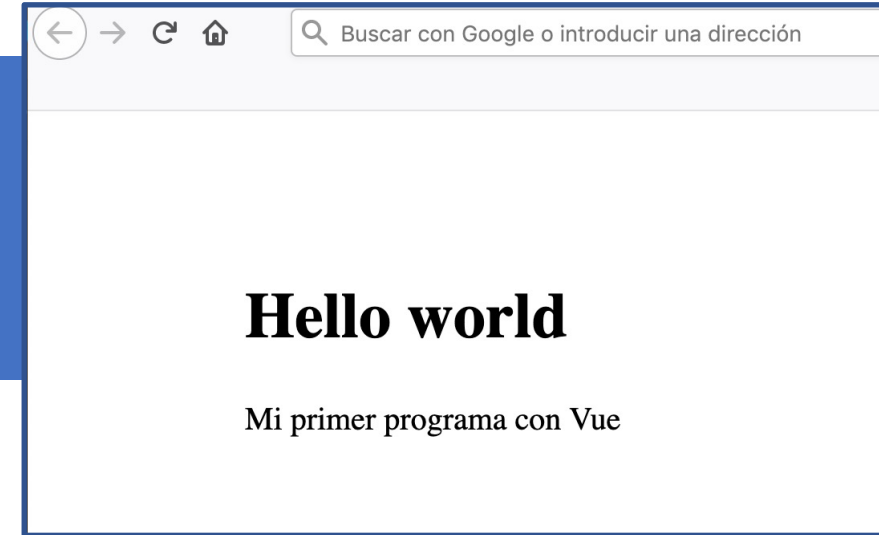
Dentro del código HTML se usan directivas Vue (v-) para **conectar el HTML a los datos y lógica** administrados desde la aplicación Vue.

Por ejemplo, para generar texto dinámicamente en la plantilla (HTML)



- Paso 3/4 - Generar contenido y vincular atributos II

- Enlazamos los **datos de Vue** con los **elementos HTML** mediante el operador *Interpolation* **{{ }}**



index.html

```
...  
<div id="app">  
  <h1>{{ mensaje1 }}</h1>  
  <p>{{ mensaje2 }}</p>  
</div>  
...
```

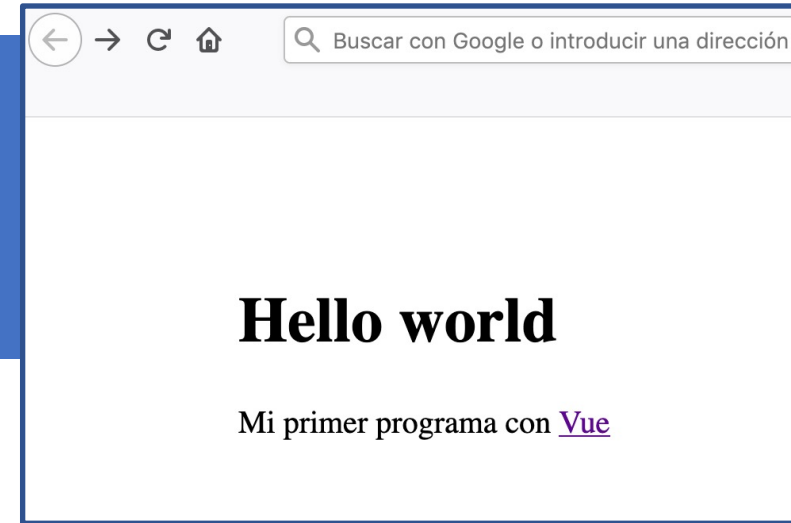
generamos el
contenido
almacenado
en los datos

app.js

```
const app = Vue.createApp({  
  data: function () {  
    return {  
      mensaje1: 'Hello world',  
      mensaje2: 'Mi primer programa con Vue'  
    };  
  },  
  methods: {  
  }  
});  
app.mount("#app");
```


Paso 3/4 - Generar contenido y vincular atributos III

- Pasamos valores dinamicos a **atributos de elementos HTML** con la directiva de **data binding (v-bind)**.



index.html

```
...  
<div id="app">  
  <h1>{{mensaje1}}</h1>  
  <p>{{mensaje2}} con  
    <a v-bind:href="url">{{jsF}}</a>  
  </p>  
</div>  
...
```

la forma abreviada de v-bind:href es :href

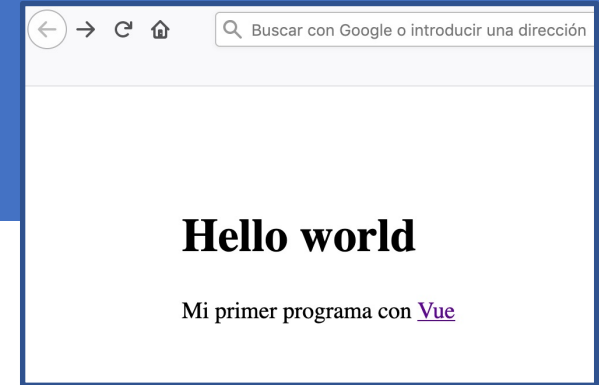
generamos datos

app.js

```
const app = Vue.createApp({  
  data: function () {  
    return {  
      mensaje1: 'Hello world',  
      mensaje2: 'Mi primer programa con',  
      jsF: 'Vue',  
      url: 'https://vuejs.org'  
    };  
  },  
  methods: {}  
});  
app.mount("#app");
```

Paso 4/4 – Definir y llamar a métodos

(this para acceder a los datos)



app.js

generamos datos

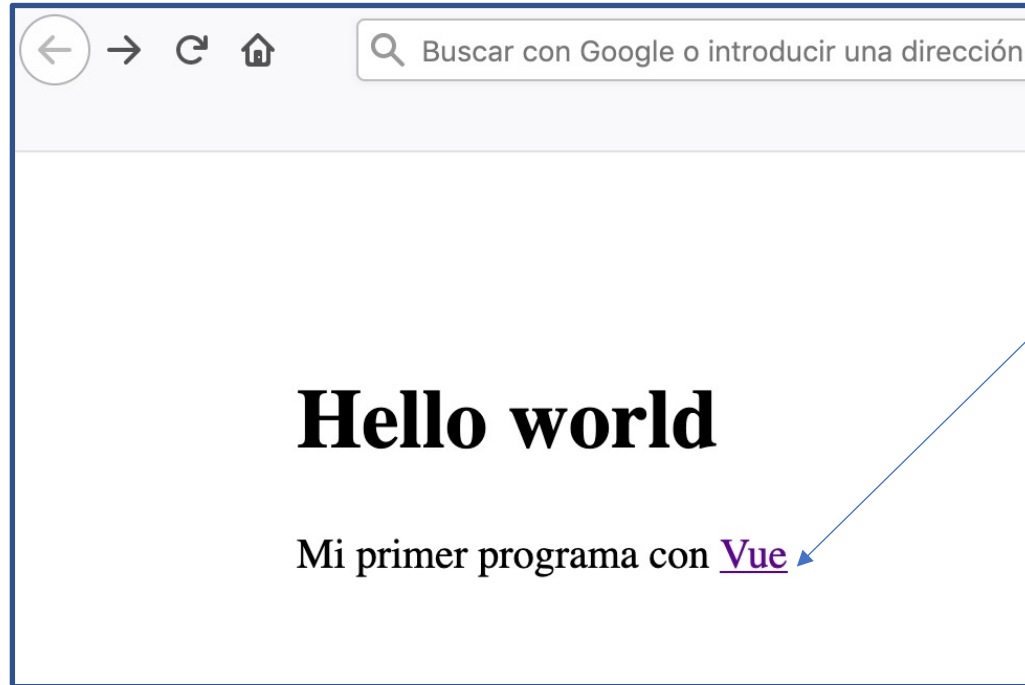
index.html

```
...  
<div id="app">  
  <h1>{{mensaje1}}</h1>  
  <p>{{mensaje2}} con  
    <a v-  
bind:href="mostrarUrl()">{{jsF}}</a>  
  </p>  
</div>  
...
```

```
const app = Vue.createApp({  
  data: function () {  
    return {  
      mensaje1: 'Hello world',  
      mensaje2: 'Mi primer programa con',  
      jsF: 'Vue', url: 'https://vuejs.org'  
    };  
  },  
  methods: {  
    mostrarUrl() { return this.url; }  
  }  
});  
app.mount("#app");
```

console.log() para
mostrar mensajes en consola

Ejercicio 2



Modifica el ejercicio 1 para que en lugar de “Vue” aparezca **de forma aleatoria Vue, React o Angular**, con su correspondiente link a la web de cada framework:

- <https://vuejs.org>
- <https://es.reactjs.org/>
- <https://angular.io>

Puedes usar arrays en js, y para generar un numero aleatorio entre 0 i 2:

- `Math.floor(Math.random() * 3)`