



Aprender **VueJS**

con 100 ejercicios prácticos

Ángel Vázquez Vázquez y Ramón Serrano Valero



Aprender

VueJS

con 100 ejercicios prácticos

Acceda a www.marcombo.info
para descargar gratis
el contenido adicional
complemento imprescindible de este libro

Código:

VUEJS1

Aprender

VueJS

con 100 ejercicios prácticos



Aprender VueJS con 100 ejercicios prácticos

© 2020 Ángel Vazquez Vazquez y Ramón Serrano Valero

© 2020 MARCOMBO, S.L.

www.marcombo.com

Diseño de la cubierta: Giancarlo Salinas

Director de colección: Pablo Martínez Izurzu

Directora de producción: M.^a Rosa Castillo Hidalgo

Maquetación: María Paz Mora Encinas

Correctora: Laura Seoane

«Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra sólo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necesita fotocopiar o escanear algún fragmento de esta obra.»

ISBN: 978-84-267-2802-9

Producción del ebook: booqlab.com

Presentación

APRENDER VUEJS CON 100 EJERCICIOS PRÁCTICOS

Los 100 capítulos que contiene este libro realizan un recorrido por los principios de desarrollo utilizando el framework de Vue. Es recomendable realizar los ejercicios en orden, sobre todo a partir del ejercicio 40, donde empieza la aplicación SPA mediante el CLI. Puede apoyarse en los ejercicios, ya implementados subidos en la plataforma Marcombo, plenamente funcionales.

Una vez finalizado este libro, el lector habrá adquirido el conocimiento para crear su propia aplicación SPA, habilitándole para afrontar proyectos en esta tecnología. Tendrá capacidad de crear aplicaciones rápidamente, mejorar su pensamiento crítico a la hora de abordar un nuevo desarrollo, así como la habilidad para tomar como base la arquitectura propuesta y poder mejorarla conforme nuevos conocimientos sean incorporados.

A QUIÉN VA DIRIGIDO

A todo aquel que busque iniciarse en el desarrollo web, la programación o adentrarse en el mundo de este nuevo framework Vue.

También está dirigido a aquellos que ya hayan realizado sus primeros pasos con Vue, pues en este libro encontrarán las respuestas a las dudas que les surjan y aquellas que no hayan podido resolver hasta ahora.

LA FORMA DE APRENDER

Nuestra experiencia en el ámbito de la enseñanza nos ha llevado a diseñar este tipo de manual, en el que cada una de las funciones se ejercita mediante la realización de un ejercicio práctico. Dicho ejercicio se halla explicado paso a paso y pulsación a pulsación, a fin de no dejar ninguna duda en su proceso de ejecución. Además, lo hemos ilustrado con imágenes descriptivas de los pasos más importantes o de los resultados que deberían obtenerse y con recuadros **IMPORTANTE** que ofrecen información complementaria sobre los temas tratados en los ejercicios.

LOS ARCHIVOS NECESARIOS

En la parte inferior de la primera página del libro encontrará el código de acceso que le permitirá descargar de forma gratuita los contenidos adicionales del libro en *www.marcombo.info*.

Los recuadros Importante incluyen acciones que deben hacerse para asegurarse de que realiza el ejercicio correctamente. También contienen información interesante a aprender, porque le facilitará su trabajo con el programa.

Los ejercicios se han escrito sistemáticamente, paso a paso, para que nunca se pierda durante su realización.

Ramón: *Ha sido duro compaginar el mundo laboral con el Máster en Big Data y la escritura de este libro, sin embargo, gracias al apoyo de mi mujer Sara de la V., he podido sacar fuerzas para poder finalizar este libro, ¡eres mi fuerza, siempre sacas lo mejor de mí! Agradezco a mis padres por inculcarme el espíritu de esfuerzo. A Xavi y Martín, mis sobrinos, que, algún día, cuando comprendan este libro, aprendan este framework y los nuevos que aparezcan. A mi amigo Ángel V., una mente siempre activa, preparado para cualquier reto, un placer escribir junto a ti. A Gabriel G. por años compartiendo conocimientos que nos permitieron innovar. Así como a mis compañeros Miguel N. y Alonso V. por habernos aventurado juntos en este mundo Vuetizado.*

Ángel: *Este libro es para las personas que me rodean: mis padres que siempre me apoyan, mis amigos y compañeros, mis gatas que tanta compañía me hicieron a la hora de trabajar en este libro, ¡y para tí, Lucía!*

Mención especial para mi compañero Ramón con el que colaboro ya por tercera vez en este tipo de aventuras. ¡Gracias por hacer tan fácil trabajar contigo, nano!

No pudisteis estar presentes Juanto G. y Pablo F. en este libro, pero formáis parte de este equipo, esperamos abordar nuevos retos juntos de nuevo.

Índice

- ¡Hola, Vue!
- Configurar el entorno
- Instancia Vue
- Instancias Vue
- Ciclo de vida
- Data binding
- Watch
- Computed properties
- Métodos
- Componentes de la instancia
- Eventos
- Renderizar HTML
- Vincular propiedades
- Condicionales
- Listas
- Enumeración de propiedades
- Filtros
- Renderizar solo una vez
- Mouse move
- Directivas personalizadas
- Directivas personalizadas II

Directivas personalizadas III
Directivas personalizadas IV
Componente
Componente parametrizables
Componente Tipo Propiedades
Cambiar valor de props
Reactividad Data
Reactividad NextTick
Directiva v-show vs v-if
Bucle v-for y key
Bucle atributos y rango
Peticiones Fetch
Contexto This
Promesas
Conjunto de Promesas
Inline templates
Slots
Transiciones/Animaciones I
Transiciones/Animaciones II
Instalación NodeJs
Nuevo proyecto con Vue CLI 3
Estructura del proyecto
Compilar en desarrollo y producción
Variables de entorno y ejecución
CLI Service scripts y servidor http

Componente – App, Librería y WC
Despliegue en Servidor de Aplicaciones
Instalación Vue Router y Modos
Carga óptima de Ruta y navegación
Rutas dinámicas
Rutas anidadas
Navegación desde código
Protección global de rutas
Protección en ruta y componente
Vuetify Diseño menú lateral
V-Toolbar y rutas de navegación
Crear Tablas
Selección de filas en tabla
Comunicación Hijo a Padre y Snackbar
Diseñar y visualizar ficha
Vuex, primeros pasos
Vuex – Getters
Vuex – Mutaciones
Vuex – Acciones
Vuex – Módulos
Vuex Plugin Logger
Vuex – Persistencia estados al refrescar
Vuex – Persistencia en Cookies
Vuex – Subscriptores
Axios – Comunicación con Servidor

Axios – Cancelar solicitud
Axios – Interceptor de solicitudes
Axios – Interceptor Respuestas
Axios – Post
Axios – PUT
Axios – Delete
Comunicación – Padre – Hijo
Comunicación – Hijo – Padre
Comunicación – Hijo – Padre por Callback
Comunicación – Entre hermanos
Comunicación – Bus de eventos
Comunicación – Mensajería Global
Comunicación bidireccional
Formulario – Campo Texto
Formularios – Email con validación
Formularios – Fechas moment
Formularios – Opciones Checkbox
Formularios – Opción única Radiobutton
Formularios – Combobox
Formularios – Campo Personalizado
Formularios – Validación Vuelidate
Formularios – Validación personalizada
Pruebas Unitarias – Jest
Pruebas Unitarias – Creación
Pruebas Unitarias – Props

Pruebas Unitarias – Aislar componentes

Pruebas Unitarias – Mock de Vuex

Pruebas Unitarias – Router

Pruebas desde UI

¡Hola, Vue!

LOS COMIENZOS

¡Bienvenidos a Vue, queridos lectores! En este ejercicio veremos cómo empezar a trabajar con Vue con el mínimo esfuerzo, pero en primer lugar daremos algunas pinceladas sobre Vue.

¿Qué es Vue? Vue es un framework progresivo para construir interfaces de usuario ❶.

¿Qué significa progresivo? Simplemente que está modularizado de manera que su librería principal está enfocada solo a la parte visual y, usándola en conjunto con otras librerías o proyectos, puede construir fantásticas aplicaciones web SPA. Esto significa que Vue no es un todo o nada sino que puede escoger las partes que más le interesen e incluso usarlas conjuntamente con otros frameworks o librerías web.

Para ilustrar el concepto de progresivo vamos a importar lo mínimo necesario para crear nuestra primera aplicación.

En primer lugar, vamos a utilizar un editor online como JSFiddle (<https://jsfiddle.net/>) y en la ventana de HTML añadiremos el siguiente contenido ❷.

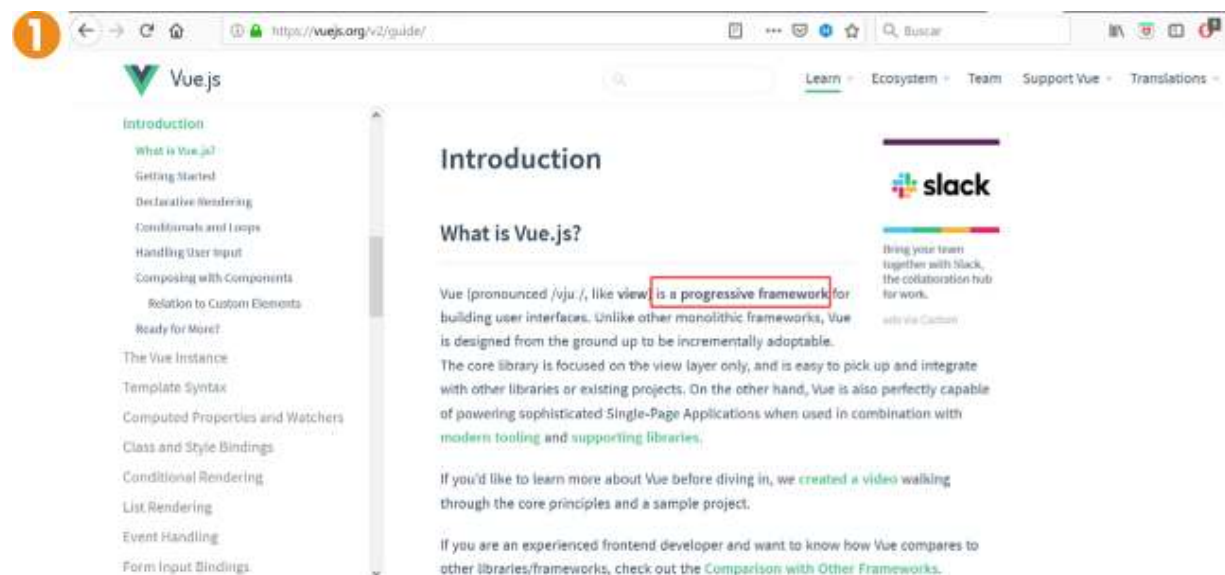
Tenemos nuestro “Hola Vue” que renderiza una página de forma estática. Para trabajar de forma dinámica añadiremos los scripts necesarios para importar la librería de Vue ❸.

En la ventana del editor para el código Javascript añadimos la instancia de Vue, la ligamos al elemento con id #app y añadimos

una variable en el bloque **data** denominada message⁴.

Volviendo al HTML asociamos a uno de los elementos el id que hemos declarado en la parte de Javascript ⁵. ¡Dentro de este elemento añadimos un título (h1) con la variable message y ya tenemos nuestro “¡Hola Vue!”

¡Así de fácil ha sido crear nuestra primera aplicación y de forma online! En siguientes ejercicios configuraremos nuestro entorno en local para trabajar cómodamente y con la ayuda de múltiples herramientas para el desarrollo web.



2



Run

Save

Tidy

Collaborate

Update

New and updated JS/CSS linters

Fiddle meta

Untitled fiddle

No description

Add title to make the fiddle visible
on your profile page

Resources URL cdnjs

Async requests

Other (links, license)

HTML

```
1 <html>
2
3 <head>
4   <meta charset="utf-8">
5   <title>My Vue</title>
6 </head>
7
8 <body>
9   <div>
10    <h1>Hola Vue!</h1>
11  </div>
12 </body>
13 </html>
```

CSS

1

Hola Vue!

3

Untitled fiddle

No description

Add title to make the fiddle visible
on your profile page

Resources URL cdnjs

Async requests

Other (links, license)

HTML

```
1 <html>
2
3 <head>
4   <meta charset="utf-8">
5   <title>My Vue</title>
6 </head>
7
8 <body>
9   <div id="app">
10    <h1>{{message}}</h1>
11  </div>
12  <script src="https://unpkg.com/vue"></script>
13 </body>
14 </html>
```

1

Hello Vue!

JavaScript + No-Library (pure JS)

```
1 const app = new Vue({
2   el: '#app',
3   data: {
4     message: 'Hello Vue!'
5   }
6 });
```

4

resources

to make the fiddle visible
profile page

Resources URL cdnjs

requests

links, license)

```
1 <meta charset="utf-8">
2 <title>My Vue</title>
3 </head>
4
5 <body>
6   <div id="app">
7     <h1>{{message}}</h1>
8   </div>
9   <script src="https://unpkg.com/vue"></script>
10 </body>
11 </html>
```

JavaScript + No-Library (pure JS)

```
1 const app = new Vue({
2   el: '#app',
3   data: {
4     message: 'Hello Vue!'
5   }
6 });
```

5

resources

to make the fiddle visible
profile page

Resources URL cdnjs

requests

links, license)

Fiddle meta

Untitled fiddle

No description

Add title to make the fiddle visible
on your profile page

Resources URL cdnjs

Async requests

Other (links, license)

HTML

```
1 <html>
2
3 <head>
4   <meta charset="utf-8">
5   <title>My Vue</title>
6 </head>
7
8 <body>
9   <div id="app">
10    <h1>{{message}}</h1>
11  </div>
12  <script src="https://unpkg.com/vue">
13 </script>
14 </body>
15 </html>
```


Configurar el entorno

PRIMEROS PASOS CON ATOM

Comenzaremos este viaje de aprendizaje configurando un entorno que nos facilite el desarrollo de nuestras aplicaciones. Para este libro como editor hemos elegido Atom pero podría utilizar cualquier otro con el que se sienta cómodo.

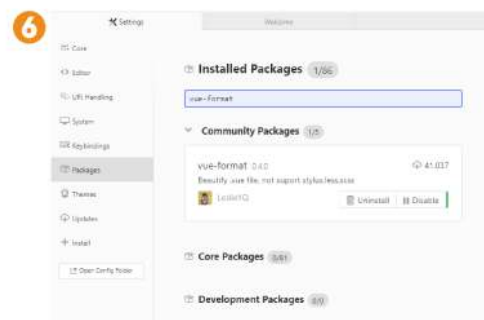
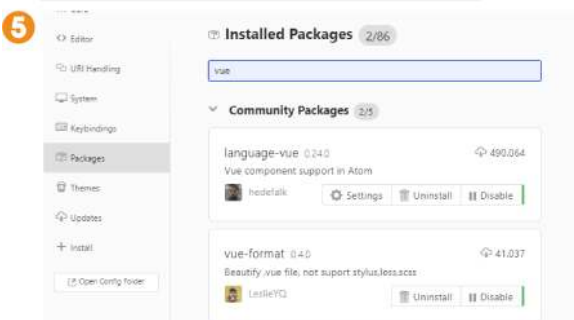
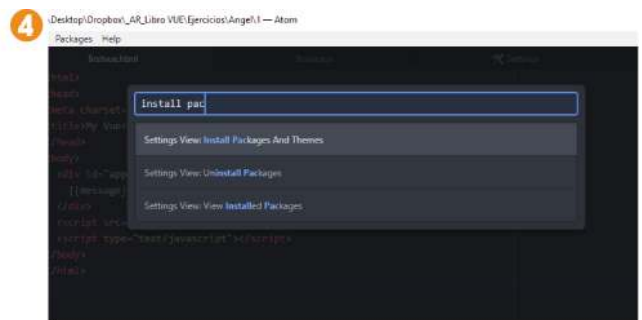
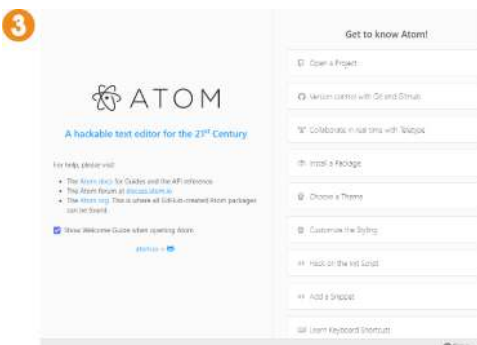
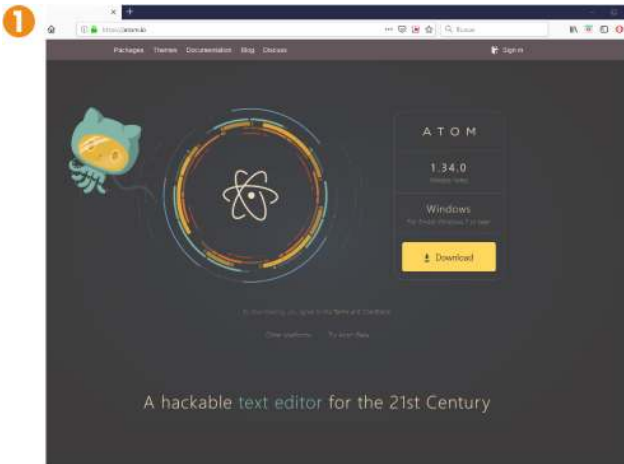
En primer lugar, vamos a la página de Atom (<https://atom.io/>) ¹ y una vez descargado instalamos ² y lo iniciamos ³ y a partir de este momento ya tenemos un editor para nuestro código.

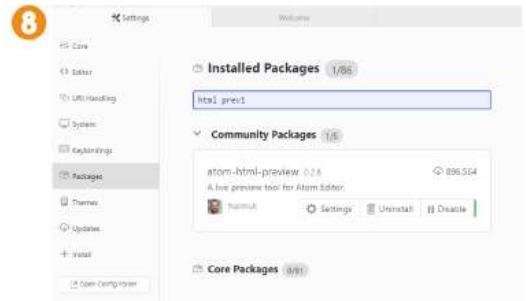
Esto no se acaba aquí, queremos tener también a nuestro alcance las herramientas que nos hagan desarrollar nuestras aplicaciones más cómoda y rápidamente, con lo que vamos a añadir algunos plugins.

Para ello iremos a la vista de ajustes en la que podremos buscar y seleccionar plugins para su instalación ⁴. Entre la ingente cantidad de herramientas que podremos añadir a nuestro editor nosotros hemos elegido algunas que pueden ser útiles. Sin embargo, utilice tantas como quiera y necesite para sentirse cómodo ya que esta es una elección muy personal.

Entre las que hemos seleccionado están language-vue ⁵ que resalta la sintaxis de vue, vue-format ⁶, atom-beautify ⁷ que dan formato a nuestro código y atom-html-preview ⁸ que integra dentro de nuestro editor una ventana para renderizar el contenido automático de código en vivo.

De esta forma, si retomamos los archivos de nuestra primera aplicación, podremos trabajar cómodamente en nuestro editor local con todos los plugins para la ayuda al desarrollo y el renderizado HTML integrado 9.





9

```

1. <html>
2.
3. <head>
4.   <meta charset="utf-8">
5.   <title>My Vue</title>
6. </head>
7.
8. <body>
9.   <div id="app">
10.    <h1>{{message}}</h1>
11.  </div>
12.  <script src="https://unpkg.com/vue"></script>
13.  <script src="firstvue.js"></script>
14. </body>
15.
16. </html>

```

Hello Vue!

Instancia Vue

MANEJO DE VARIABLES

Vamos a construir un pequeño blog de viajes y, para ello, como hemos visto en anteriores ejercicios, creamos la instancia Vue que manejará el contexto del elemento con id #app asociado ❶.

Dentro de este elemento crearemos un título general y una serie de artículos con un pequeño titular y contenido ❷. De esta forma definimos una plantilla para nuestro blog y para la que nos faltaría añadir contenido. Este contenido variará, con lo que como primera aproximación crearemos unas cuantas variables dentro de la instancia Vue que representarán los textos que hayamos escrito sobre nuestros últimos viajes.

Tenemos nuestro pequeño blog de viajes listo para añadir contenido, este contenido podría cargarse en las variables desde alguna fuente externa tanto para imágenes como para texto, pero como primer paso simplemente representamos el valor contenido dentro de las variables de la instancia Vue ❸.

Esto Vue lo realiza así de fácil, podemos considerar el data como el lugar donde definimos las variables del modelo con las que trabaja nuestra instancia de Vue. Hasta ahora la definición de estos valores se realiza de forma estática, ya veremos más adelante cómo podemos ir modificando estos valores dentro de la instancia, donde su valor se reflejaría automáticamente en la plantilla, al haber sufrido un cambio, esto es propio de la reactividad de Vue. ¡Genial!

1

```

firstvue.html
8 <body>
9 <h1>My Vue Blog</h1>
10 <div class="flex-container">
11   <div id="app" class="flex-item flex-item-content">
12     <h2>Content</h2>
13
14     <h3>{{title1}}</h3>
15     {{post1}}
16
17     <h3>{{title2}}</h3>
18     {{post2}}
19
20     <h3>{{title3}}</h3>
21     {{post3}}
22   </div>
23 </div>
24 <script src="https://unpkg.com/vue"></script>
25 <script src="firstvue.js"></script>
26 </body>

```

```

firstvue.js
1 const app = new Vue({
2   el: '#app',
3   data: {
4     title1: 'Trip 1',
5     title2: 'Trip 2',
6     title3: 'Trip 3',
7     post1: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eu

```

firstvue.html Preview

My Vue Blog

Content

Trip 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 3

2

```

firstvue.html
8 <body>
9 <h1>My Vue Blog</h1>
10 <div class="flex-container">
11   <div id="app" class="flex-item flex-item-content">
12     <h2>Content</h2>
13
14     <h3>{{title1}}</h3>
15     {{post1}}
16
17     <h3>{{title2}}</h3>
18     {{post2}}
19
20     <h3>{{title3}}</h3>
21     {{post3}}
22   </div>
23 </div>
24 <script src="https://unpkg.com/vue"></script>
25 <script src="firstvue.js"></script>
26 </body>

```

```

firstvue.js
1 const app = new Vue({
2   el: '#app',
3   data: {
4     title1: 'Trip 1',
5     title2: 'Trip 2',
6     title3: 'Trip 3',
7     post1: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eu
8     post2: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eu

```

firstvue.html Preview

My Vue Blog

Content

Trip 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed

3

```
firstvue.js
1 const app = new Vue({
2   el: '#app',
3   data: {
4     title1: 'Trip 1',
5     title2: 'Trip 2',
6     title3: 'Trip 3',
7     post1: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.',
8     post2: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.',
9     post3: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.'
10  }
11 });
12
```

firstvue.html Preview

My Vue Blog

Content

Trip 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit, s

Instancias Vue

GESTIONANDO MÁS DE UNA INSTANCIA

En nuestro blog de viajes además de una colección de artículos queremos añadir un pequeño índice con información sobre los títulos y en qué fecha se publicaron. Con este fin podemos añadir una nueva instancia de Vue que controle este componente índice.

Partimos de nuestro ejemplo anterior ¹ creando una nueva columna en la parte derecha con nuestro nuevo componente. Este nuevo elemento contendrá una lista no ordenada de títulos con fecha de publicación, y para ello crearemos el HTML adecuado y lo identificamos con el id app2.

La segunda parte del ejercicio será crear una nueva instancia Vue en el código Javascript con la que manejar nuestro nuevo componente. Hasta este momento tenemos declarada una lista con elementos en HTML con una variable por ítem de la lista y ahora declararemos y daremos valores a todas y cada una de ellas ² ³.

Hemos creado nuestro blog de artículos de viaje y lo hemos completado con la creación de un pequeño índice resumen con títulos y fechas. Comprobamos así que pueden convivir varias instancias Vue, una para cada componente, sin ningún problema.

1

firstvue.html

```

8 <body>
9   <h1>My Vue Blog</h1>
10  <div class="flex-container">
11    <div id="app" class="flex-item flex-item-content">
12      <h2>Content</h2>
13
14      <h3>{{title1}}</h3>
15      {{post1}}
16
17      <h3>{{title2}}</h3>
18      {{post2}}
19
20      <h3>{{title3}}</h3>
21      {{post3}}
22    </div>
23  </div>
24  <script src="https://unpkg.com/vue"></script>
25  <script src="firstvue.js"></script>
26 </body>

```

firstvue.js

```

1 const app = new Vue({
2   el: '#app',
3   data: {
4     title1: 'Trip 1',
5     title2: 'Trip 2',
6     title3: 'Trip 3',
7     post1: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
8       tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.'
9   }
10 })

```

firstvue.html Preview

My Vue Blog

Content

Trip 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 3

```

1  firstvue.html
2
3  19 <h3>{{title2}}</h3>
4    20 {{post2}}
5    21
6    22 <h3>{{title3}}</h3>
7    23 {{post3}}
8    24 </div>
9
10 25
11 26
12 27 <div id="app2" class="flex-item flex-item-summary">
13 28 <h2>{{message}}</h2>
14 29 <ul class="summary">
15 30 <li class="item">{{title1}}</li>
16 31 <li class="item">{{title2}}</li>
17 32 <li class="item">{{title3}}</li>
18 33 </ul>
19 34 </div>
20 35
21 36
22 37 <script src="https://unpkg.com/vue"></script>
23 38 <script src="firstvue.js"></script>
24 39 </body>
25 40
26 41 </html>
27 42
28
29 firstvue.js
30 5 title2:'Trip 2',
31 6 title3:'Trip 3',
32 7 post1: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt
33
34
35
36
37
38
39
40
41
42

```


3

firstvue.html

firstvue.js

firstvue.html Preview

```
1 const app = new Vue({
2   el: '#app',
3   data: {
4     title1: 'Trip 1',
5     title2: 'Trip 2',
6     title3: 'Trip 3',
7     post1: 'Lorem ipsum dolor sit amet, consectetur adipisicing',
8     post2: 'Lorem ipsum dolor sit amet, consectetur adipisicing',
9     post3: 'Lorem ipsum dolor sit amet, consectetur adipisicing'
10  }
11 });
12
13 const app2 = new Vue({
14   el: '#app2',
15   data: {
16     message: 'Summary',
17     title1: '16/06/2019 Trip 1',
18     title2: '05/12/2018 Trip 2',
19     title3: '04/10/2018 Trip 3'
20  }
21 });
22
```

My Vue Blog

Content

Summary

Trip 1

16/06/2019 Trip 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

05/12/2018 Trip 2

04/10/2018 Trip 3

Trip 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Trip 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Ciclo de vida

FASES DEL CICLO DE VIDA

Se denomina ciclo de vida de una instancia Vue a una serie de estados por los que va pasando el componente. Estos estados son básicamente cuatro: created, mounted, updated y destroyed.

Vue nos permitirá definir acciones anteriores y posteriores a la transición desde o hacia cada estado interno del componente. Los métodos en cuestión para implementar estas acciones son:

- **beforeCreate:** evento lanzado antes de tener el componente cargado, implica no poder acceder al componente a nivel de Dom.
- **created:** evento donde se verifica si el componente tiene plantilla, entonces se compila y se observan las propiedades computadas, data, métodos y eventos. Pero no podemos acceder al \$el.
- **beforeMount:** evento que ocurre antes de representar el componente.
- **mounted:** evento que implica que el componente está cargado por completo, se añade al DOM, quedando el componente accesible a través de \$el.
- **beforeUpdate:** evento que se ejecuta cuando el valor del data del componente cambia.
- **updated:** evento invocado tras finalizar la modificación de valor del data.

- `beforeDestroy`: evento que elimina eventos que estuvieran activos en el componente, antes de eliminar la instancia.
- `destroyed`: evento lanzado tras desacoplar el componente.

En estos métodos podremos reservar o liberar recursos, realizar cálculos, hacer llamadas http, etc.

Para ilustrar mejor cómo funciona el ciclo de vida implementaremos cada uno de estos métodos escribiendo en una consola del navegador un mensaje por cada vez que entre en los métodos mencionados ❶.

Una vez que carga la página vemos que la instancia es creada, cargada en el DOM y está disponible cuando pasa al estado `mount` ❷.

Para comprobar qué pasa cuando actualizamos el componente podemos editar el texto "Hola Vue!" ❸ y añadir alguna modificación para ver que pasa por los métodos `beforeUpdate` y `updated` ❹.

En el caso de que eliminemos el componente podremos forzar su destrucción a través del botón `Destroy` comprobar que la instancia pasa por los métodos `beforeDestroy` y `destroyed` ❺.

1

```

firstvue.js
1  const app = new Vue({
2    el: '#app',
3    data: {
4      message: 'Hola Vue!',
5    },
6    beforeCreate: function() {
7      console.log("beforeCreate");
8    },
9    created: function() {
10     console.log("created");
11   },
12   beforeMount: function() {
13     console.log("beforeMount");
14   },
15   mounted: function() {
16     console.log("mounted");
17   },
18   beforeUpdate: function() {
19     console.log("beforeUpdate");
20   },
21   updated: function() {
22     console.log("updated");
23   },
24   beforeDestroy: function() {
25     console.log("beforeDestroy");
26   },
27   destroyed: function() {
28     console.log("destroyed");
29   },
30   methods: {
31     handleClick: function(){
32       app.$destroy();
33     }
34   }
35 });
36

```

2

```

firstvue.html
5  <title>My Vue</title>
6  <style src="firstvue.css"></style>
7  <link rel="stylesheet" href="firstvue.css">
8  </head>
9
10 <body>
11   <div id="app">
12     <h1>{{ message }}</h1>
13
14     <input type="text" v-model="message"/>
15     <button type="button" v-on:click="handleClick">Destroy</button>
16   </div>
17
firstvue.js
1  const app = new Vue({
2    el: '#app',
3    data: {

```

firstvue.html Preview

Hola Vue!

1. beforeCreate
2. created
3. beforeMount
4. mounted
5. Download the Vue Devtools extension for a better development experience:
<https://github.com/vuejs/vue-devtools>
6. You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at <https://vuejs.org/guide/deployment.html>

3

```

firstvue.html
3 <head>
4   <meta charset="utf-8">
5   <title>My Vue</title>
6   <style src="firstvue.css"></style>
7   <link rel="stylesheet" href="firstvue.css">
8 </head>
9
10 <body>
11   <div id="app">
12     <h1>{{ message }}</h1>
13
14     <input type="text" v-model="message"/>
15     <button type="button" v-on:click="handleClick">Destroy</button>
16   </div>
17
18   <ul id="myULContainer" class="console"></ul>
19
20   <script src="https://unpkg.com/vue"></script>
21   <script type="application/javascript" src="https://cdn.rawgit.com/Alorel/co
22

```

firstvue.html Preview

Hola Vue!

1. beforeCreate
2. created
3. beforeMount
4. mounted
5. Download the Vue Devtools extension for a better development experience: <https://github.com/vuejs/vue-devtools>
6. You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at <https://vuejs.org/guide/deployment.html>

4

firstvue.html Preview

Hola Vue!!!

1. beforeCreate
2. created
3. beforeMount
4. mounted
5. Download the Vue Devtools extension for a better development experience: <https://github.com/vuejs/vue-devtools>
6. You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at <https://vuejs.org/guide/deployment.html>
7. beforeUpdate
8. updated
9. beforeUpdate
10. updated

5

1. beforeCreate
2. created
3. beforeMount
4. mounted
5. Download the Vue Devtools extension for a better development experience: <https://github.com/vuejs/vue-devtools>
6. You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at <https://vuejs.org/guide/deployment.html>
7. beforeUpdate
8. updated
9. beforeUpdate
10. updated
11. beforeDestroy
12. destroyed

Data binding

VALORES Y VARIABLES

Volvemos de nuevo con nuestro blog de viajes, en esta ocasión además de listar los artículos escritos y el índice queremos añadir un pequeño apartado para editar nuevos artículos.

En primer término, crearemos un input para escribir un título para nuestro artículo y una caja de texto más amplia para poder escribir todo nuestro artículo en la misma ❶.

Además de crear el marcado en lenguaje HTML añadiremos a cada input un nuevo atributo con el siguiente formato `v-model="nombreVariable"` para enlazar la variable con el contenido que escribamos. Del mismo modo tendremos que declarar esas variables con los nombres que habremos elegido en el código Javascript de la instancia Vue ❷.

Posteriormente añadiremos el título y el contenido en la página para que se renderice a medida que se escribe y podamos ir viendo cómo queda ❸. Esta es la “magia” del enlazado de variables con los inputs ya que somos capaces de darle un valor a una variable dentro de la instancia e ir mostrando lo que guarda esa variable dentro del contexto de nuestra instancia. ❹

Simplemente con estos pasos tendremos una forma fácil de añadir un nuevo artículo con su título y contenido y poder ver el renderizado en la página inmediatamente a medida que vamos generando contenido ❺.