

# Aprender **VueJS**

con 100 ejercicios prácticos

Ángel Vázquez Vázquez y Ramón Serrano Valero



Aprender

VueJS

con 100 ejercicios prácticos

Acceda a [www.marcombo.info](http://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](http://www.marcombo.com)

Diseño de la cubierta: Giancarlo Salinas

Director de colección: Pablo Martínez Izurzu

Directora de producción: M.<sup>a</sup> 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](http://www.cedro.org)) si necesita fotocopiar o escanear algún fragmento de esta obra.»

ISBN: 978-84-267-2802-9

Producción del ebook: [booqlab.com](http://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](http://www.marcombo.info).

# Cómo leer los libros “Aprender...”

El título de cada ejercicio expresa, sin lugar a dudas, en qué consiste este.

De esta forma, si le interesa, puede acceder directamente a la acción que desea aprender o refrescar.

Puede seguir el ejercicio de forma gráfica y paso a paso. Los números colocados en las fotos le remiten a entradas en el cuerpo de texto.

El número a la derecha de la página le indica claramente en qué ejercicio se encuentra en todo momento.

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.

**Importante**

Como veremos más tarde en el componente, el **prop** es un atributo que se aplica a los componentes, así como a los estados de la propiedad, para establecer el valor de la propiedad, en caso contrario será el valor en planteo.

**Tipo de Parámetro**

Como visto la importancia de los **prop** en los componentes, así la primera línea de entrada de texto y que tanto deben ser el nombre de la propiedad y el tipo de dato que se va a almacenar en la misma.

Como es típico para trabajar con los tipos de valor que tiene el atributo **prop**, en el futuro “futuro”, el componente “**prop-type**”, donde en la secuencia prop, definimos el tipo de dato que se va a almacenar en la propiedad “**prop**”, es decir, el valor que se va a introducir en el atributo “**value**”, es decir, el valor que se va a validar, para verificar su trabajo correcto.

Si por ejemplo, se define el tipo de dato “**checkbox**” en el atributo “**value**”, en este caso, podemos trae el valor para saber si el valor que se introduce mediante la propiedad componente “**componente**”, propType tiene el valor “checkbox”, texto o fecha, al realizar un **onchange** de tipo “**checkbox**”.

**Tipo Propiedades**

Componente que nos permite crear un tipo de dato para que se pueda utilizar en el resto de componentes. Es decir, que se crea un tipo de dato que se va a utilizar en el resto de componentes.

**Propiedades**

Este es el panel de propiedades de la componente “**Tipo Propiedades**”, que nos muestra las propiedades que se definen en el componente.

**Validación**

Este es el panel de validación de la componente “**Tipo Propiedades**”, que nos muestra las validaciones que se definen en el componente.

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 ①.

¿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 ②.

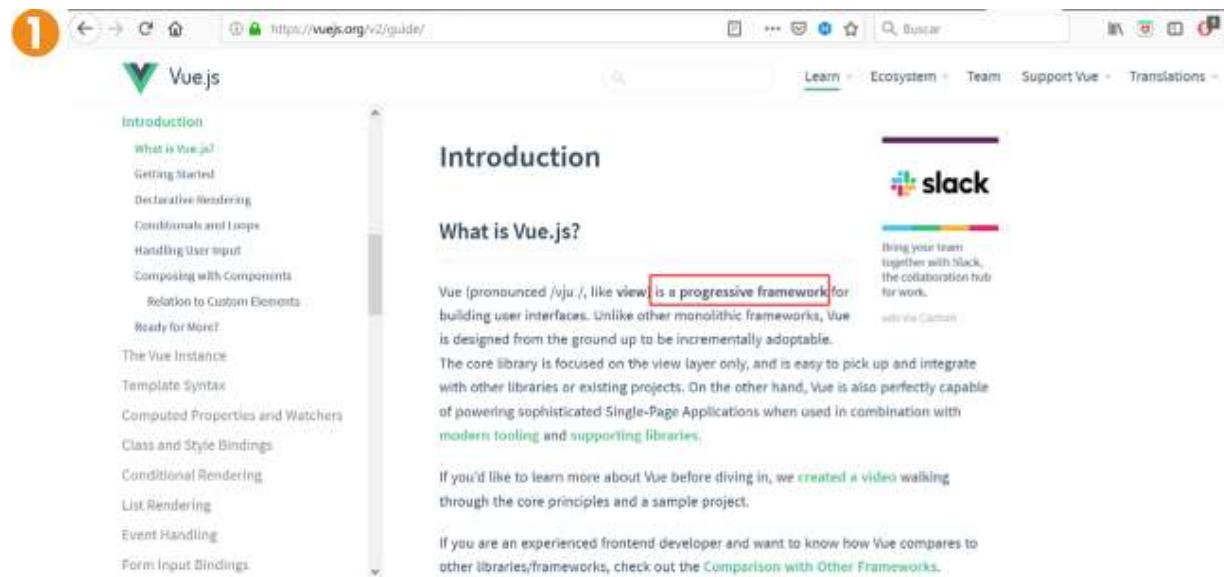
Tenemos nuestro “Hola Vue” que renderiza una página de forma estática. Para trabajar de forma dinámica añadiremos los scripts necesarios para importar la librería de Vue ③.

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 **message**<sup>4</sup>.

Volviendo al HTML asociamos a uno de los elementos el id que hemos declarado en la parte de Javascript <sup>5</sup>. ¡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.



The screenshot shows a web browser displaying the Vue.js v2 guide at <https://vuejs.org/v2/guide/>. The page is titled 'Introduction'. On the left, there is a sidebar with a table of contents for the introduction section, including links to 'What is Vue.js?', 'Getting Started', 'Directive Rendering', 'Conditionals and Loops', 'Handling User Input', 'Composing with Components', 'Relation to Custom Elements', 'Ready for More?', 'The Vue Instance', 'Template Syntax', 'Computed Properties and Watchers', 'Class and Style Bindings', 'Conditional Rendering', 'List Rendering', 'Event Handling', and 'Form Input Bindings'. The main content area starts with the heading 'Introduction' and the sub-section 'What is Vue.js?'. A callout box highlights the text: 'Vue (pronounced /vju/, like view) is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable.' To the right of this section, there is a 'slack' logo with the text 'Bring your team together with slack, the collaboration hub for work.' and a 'Join via GitHub' button. Below the slack section, there is a note: 'If you'd like to learn more about Vue before diving in, we created a video walking through the core principles and a sample project.' At the bottom, there is another note: 'If you are an experienced frontend developer and want to know how Vue compares to other libraries/frameworks, check out the Comparison with Other Frameworks.'

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 ▾

```
<html>
  <head>
    <meta charset="utf-8">
    <title>My Vue</title>
  </head>
  <body>
    <div>
      <h1>Hola Vue!</h1>
    </div>
  </body>
</html>
```

CSS ▾

Hola Vue!

3 Untitled fiddle

No description

Add title to make the fiddle visible on your profile page

Resources URL cdns

Async requests

Other (links, license)

```
<html>
  <head>
    <meta charset="utf-8">
    <title>My Vue</title>
  </head>
  <body>
    <div id="app">
      <h1>{{message}}</h1>
    </div>
    <script src="https://unpkg.com/vue"></script>
  </body>
</html>
```

JavaScript + No-Library (pure JS) ▾

```
const app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
});
```

# Hello Vue!

4  [Run](#) [Save](#) [Tidy](#) [Collaborate](#) [Update](#) [New and up...](#)

to make the fiddle visible  
profile page

uses URL cdns  
requests  
links, license)

```
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"><scr
13 </body>
14
15 </html>
16
```

JavaScript + No-Library (pure JS) ▾

```
const app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
});
```

5  [Run](#) [Save](#) [Tidy](#) [Collaborate](#) [Update](#) [New and up...](#)

title meta

intitled fiddle

description

Id title to make the fiddle visible  
your profile page

SOURCES URL cdns  
sync requests  
ther (links, license)

```
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>
```

```
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/>)<sup>1</sup> y una vez descargado instalamos <sup>2</sup> y lo iniciamos <sup>3</sup> y a partir de este momento ya tenemos un editor para nuestro código.

Esto no se acaba aquí, querremos 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 <sup>4</sup>. 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 <sup>5</sup> que resalta la sintaxis de vue, vue-format <sup>6</sup>, atom-beautify <sup>7</sup> que dan formato a nuestro código y atom-html-preview <sup>8</sup> 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.

1. The Atom website download page. A hackable text editor for the 21st Century.

2. The Atom splash screen with a rocket launch illustration. Atom is being installed. It will launch once it is done.

3. The Atom welcome screen with a 'Get to know Atom!' list.

4. The Atom settings menu with the 'Install pac' option highlighted.

5. The Atom package manager showing installed packages for 'Vue'.

6. The Atom package manager showing installed packages for 'vue-format'.

7

8

9

![Screenshot of a code editor showing a Vue.js template. The code is as follows:
1. <html>
2. 
3. <head>
4.   <meta charset=](firstvue.js)

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

# Instancia Vue

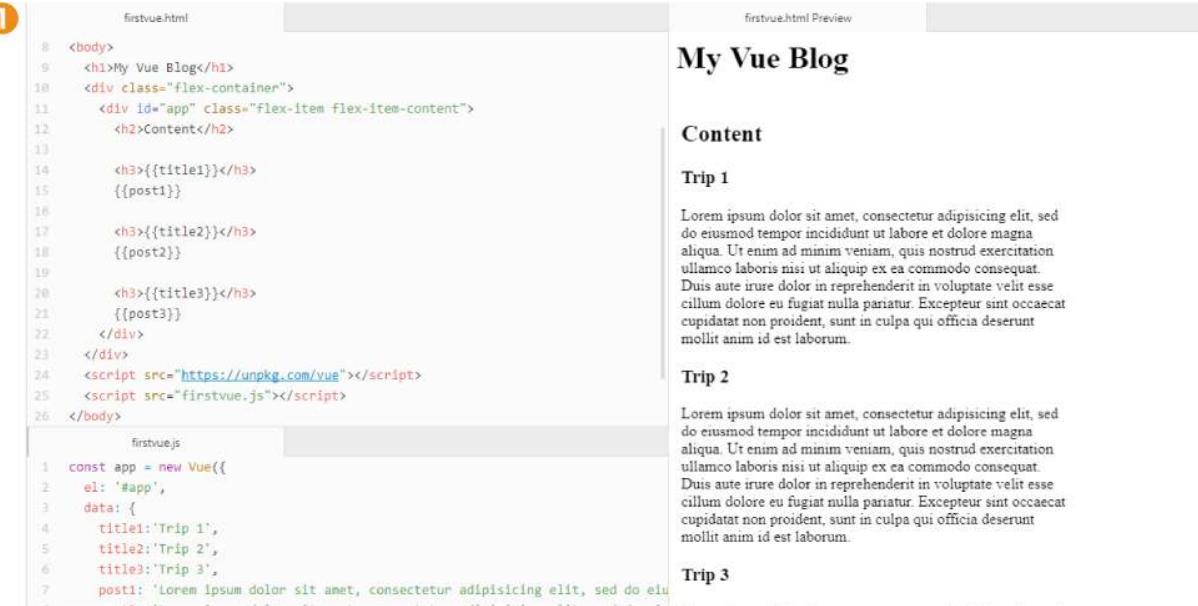
## MANEJO DE VARIABLES

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 ①.

Dentro de este elemento crearemos un título general y una serie de artículos con un pequeño titular y contenido ②. 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 ③.

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 

```

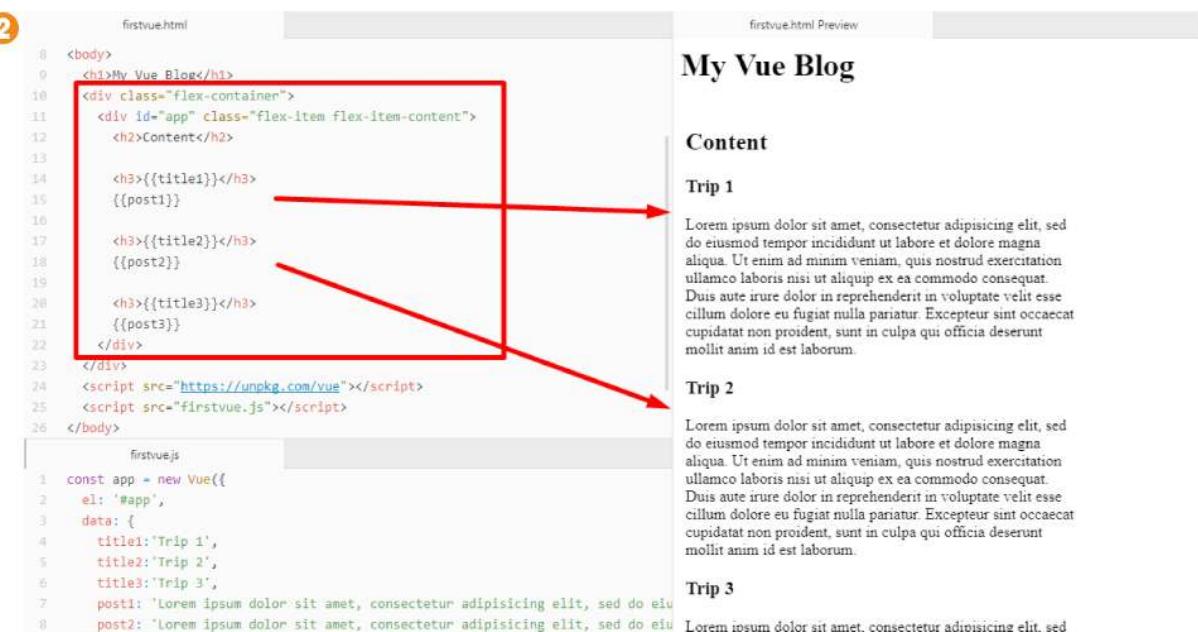
firstvue.html
8 <body>
9   <h1>My Vue Blog</h1>
10  <div class="flex-container">
11    <div id="app" class="flex-item flex-item-content">
12      <h2>Content</h2>
13
14      <h3>{{title1}}</h3>
15      {{post1}}
16
17      <h3>{{title2}}</h3>
18      {{post2}}
19
20      <h3>{{title3}}</h3>
21      {{post3}}
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 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   }
9 })

```

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 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   }
10 })

```

The image shows a development environment with two main panes. On the left is a code editor with a tab labeled 'firstvue.js' containing the following Vue.js code:

```
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 
```

On the right is a browser preview window titled 'firstvue.html Preview' showing the rendered content of the application. The title 'My Vue Blog' is displayed, followed by three sections labeled 'Content' with titles 'Trip 1', 'Trip 2', and 'Trip 3', each containing a large block of placeholder text (Lorem ipsum).

# 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 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   }
9 })

```

firstvue.html Preview 

**My Vue Blog**

**Content**

**Trip 1**

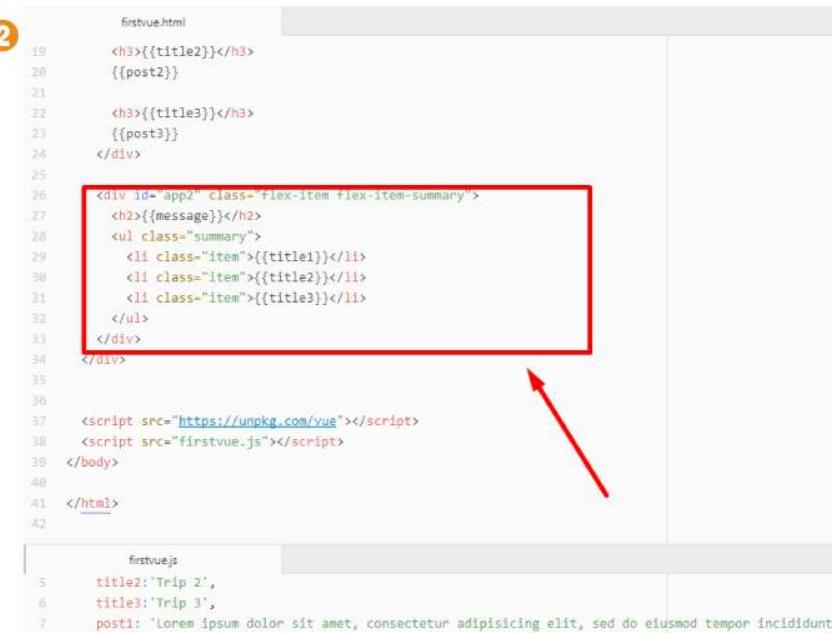
*... (post1 content) ...*

**Trip 2**

*... (post2 content) ...*

**Trip 3**

*... (post3 content) ...*

2  firstvue.html

```

19 <h3>{{title2}}</h3>
20   {{post2}}
21
22   <h3>{{title3}}</h3>
23   {{post3}}
24 </div>
25
26 <div id="app2" class="flex-item flex-item-summary">
27   <h2>{{message}}</h2>
28   <ul class="summary">
29     <li class="item">{{title1}}</li>
30     <li class="item">{{title2}}</li>
31     <li class="item">{{title3}}</li>
32   </ul>
33 </div>
34
35
36
37  <script src="https://unpkg.com/vue"></script>
38  <script src="firstvue.js"></script>
39 </body>
40
41 </html>
42

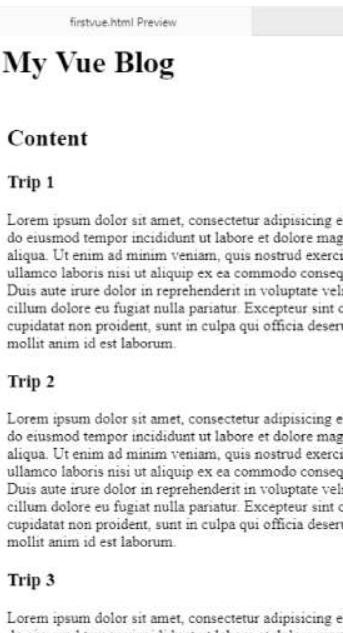
```

firstvue.js

```

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

```

firstvue.html Preview 

**My Vue Blog**

**Content**

**Trip 1**

*... (post1 content) ...*

**Trip 2**

*... (post2 content) ...*

**Trip 3**

*... (post3 content) ...*

3

firstvue.html
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
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

```

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.

#### Summary

16/06/2019 Trip 1  
 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. 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.

# 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 ①.

Una vez que carga la página vemos que la instancia es creada, cargada en el DOM y está disponible cuando pasa al estado `mount` ②.

Para comprobar qué pasa cuando actualizamos el componente podemos editar el texto "Hola Vue!" ③ y añadir alguna modificación para ver que pasa por los métodos `beforeUpdate` y `updated` ④.

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` ⑤.

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
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
firstvue.js
1 const app = new Vue({
2   el: '#app',
3   data: {
```

firstvue.html Preview

Hola Vue!

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/Alorem/co
22

```

firstvue.html Preview

Hola Vue!

Hola Vue! Destroy

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!!!**

Hola Vue!!! Destroy

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>

5 **firstvue.html Preview**

Hola Vue!!! Destroy

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 querremos 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 **1**.

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 **2**.

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 **3**. 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. **4**

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**5**.