

Capítulo 3

Entender los fundamentos de Vue.js

1. Instalación

1.1 Una versión por entorno

En este capítulo veremos que es posible instalar Vue.js de diferentes formas, dependiendo de las herramientas de las que disponga un proyecto existente o si inicia un nuevo proyecto de aplicación SPA.

En cualquiera de los casos hay dos versiones de la librería:

- una versión de desarrollo: incluye un modo de depuración que le permite mostrar advertencias en la consola e interactuar con varias herramientas de ayuda al desarrollo (extensiones del navegador);
- una versión de producción: es una versión minificada que no cuenta con el modo de depuración ni con el código de las herramientas de ayuda al desarrollo. Se utiliza para que cuando su sitio web está en línea se cargue lo más rápido posible en el navegador.

■ Observación

Vue.js no es compatible con IE 8 ni sus versiones anteriores. Solo se puede utilizar en navegadores compatibles con ES 5 y versiones posteriores de JavaScript.

1.2 Mediante descarga manual

- Para comenzar a usar Vue.js, descargue la última versión de la librería desde el sitio oficial: <https://vuejs.org/v2/guide/installation.html>



- Descargue la versión de desarrollo. Una vez que se haya descargado el archivo `vue.js`, colóquelo en el directorio de su sitio web y cárguelo en su página HTML con la etiqueta `<script>`.

■ `<script src="vue.js"></script>`

1.3 Mediante la inclusión de un CDN (lo más sencillo)

- Si no desea aumentar el peso de su proyecto, puede cargar este archivo desde un servidor remoto usando el siguiente CDN (*Content Delivery Network*):

■ `<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>`

Este es el método de instalación más sencillo para comenzar a usar las funciones de `Vue.js`.

■ Observación

Como la librería está alojada en un servidor remoto, no podrá trabajar en su proyecto sin conexión.

- Para una versión de producción, use el siguiente CDN para disponer de la última versión estable (v2.6.11):

■ `<script src="https://cdn.jsdelivr.net/npm/vue@2.6.11"></script>`

Si está utilizando módulos nativos de JavaScript ES, también hay una compilación compatible a partir de la versión 2.6 de `Vue.js`:

```
<script type="module">
  import Vue from 'https://cdn.jsdelivr.net/npm/vue@2.6.0/dist/
vue.esm.browser.js'
</script>
```

1.4 Mediante npm o yarn para proyectos grandes

1.4.1 Descarga del paquete vue

npm (*Node Package Manager*) es el registro de librerías JavaScript más grande del mundo (más de 800 000). Cada librería de este registro se carga como un módulo CommonJS o ES, que sigue la sintaxis del módulo en el que se desarrolló.

También es un administrador de dependencias que se instala en su máquina con el framework Node.js.

Para instalar de manera optimizada varias librerías de JavaScript que vayan a utilizarse en proyectos de Vue.js de tamaño mediano, se recomienda usar un administrador de dependencias como npm o yarn (supercapa de npm). En este caso, para usarlas es necesario utilizar un empaquetador de módulos como Webpack o Browserify.

■ Puede descargar Node.js desde la siguiente dirección:

<https://nodejs.org/es/download/>

■ Una vez instalado, abra el terminal de su máquina, escriba el siguiente comando y siga las instrucciones para crear un nuevo proyecto:

```
■ npm init
```

A continuación, se crea un archivo `package.json`, que servirá como un registro para que npm conozca los diferentes paquetes que utiliza su proyecto.

■ Escriba el siguiente comando en la raíz de su proyecto web para descargar el paquete `vue`:

```
■ npm install vue
```

Este comando crea un directorio `vue` en `node_modules` dentro de su proyecto y añade la referencia a ese módulo en el archivo `package.json`.

Si más adelante desea compartir su proyecto, puede compartir solo el código fuente sin el directorio `node_modules`.

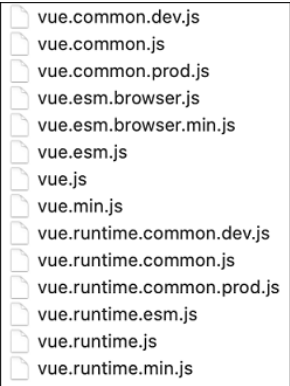
Con el comando `npm install`, npm vuelve a instalar de manera automática todas las librerías que su proyecto necesita, leyendo el archivo `package.json`.

1.4.2 Explicación de las dierentes builds

Para entender las siguientes explicaciones, debe saber que la codificación de módulos JavaScript ha evolucionado mucho en los últimos años. Además, no es posible utilizarlos directamente en el navegador (excepto los módulos nativos de ES). Por este motivo es necesario utilizar un empaquetador capaz de analizar su sintaxis, para cargar estos archivos en un navegador (consulte el capítulo Nociones esenciales de JavaScript, sección Utilización de los módulos JavaScript).

Encontrará las diferentes compilaciones de la librería en el directorio

/node_modules/vue/dist:



Aquí se resumen sus diferencias:

	UMD	CommonJS Module	ES Module (para empaquetadores)	ES Module (para navegadores)
Full	vue.js	vue.common.js	vue.esm.js	vue.esm.brow- ser.js
Runtime- only	vue.runtime.js	vue.runtime .common.js	vue.runtime.esm.js	-
Full (production)	vue.min.js	-	-	vue.esm.brow- ser.min.js
Runtime- only (production)	vue.run- time.min.js	-	-	-

A continuación se listan las definiciones de los diferentes términos.

- **Full:** este término designa las builds que contienen la parte Compilador y la parte Runtime.
- **Compiler:** se trata del código que se utiliza para compilar las cadenas de caracteres de la propiedad `template` de un componente Vue.js, dentro de una función de renderizado. Después de la compilación, el siguiente código:

```
// esto necesita un compilador
new Vue({
  template: '<div>{{ hello }}</div>'
})
```

se convertirá en:

```
// esto no lo necesita
new Vue({
  render(h) {
    return h('div', this.hello)
  }
})
```

- **Runtime:** este es el código que se encarga de crear la instancia y los componentes de Vue, manipular el DOM virtual y renderizar el DOM final, que representa a todo lo demás. Si los componentes Vue.js de la aplicación se escriben en un único archivo con la extensión `.vue`, la parte Compiler no es útil en la build. Esto hace necesario instalar en su proyecto el módulo `vue-loader` con `npm` o `yarn`, si está usando una compilación escrita con la sintaxis del módulo ES, o el módulo `vueify` si está usando una build escrita con la sintaxis CommonJS.
- **UMD (Universal Module Definition):** es una sintaxis de módulo compatible con los módulos CommonJS y AMD. Por lo tanto, estos archivos no requieren un empaquetador y el navegador puede leerlos cuando se cargan con una etiqueta `<script>`. De forma predeterminada, la versión que se puede descargar manualmente desde el sitio oficial de Vue.js y a través de los CDN incluye las partes Compiler y Runtime.
- **CommonJS Module:** esta es la sintaxis antigua que se utiliza para escribir módulos en JavaScript, antes de la aparición de los módulos nativos de ES con la versión ES 6 de JavaScript. Estos archivos se utilizan si su proyecto usa el empaquetador de módulos Browserify o la primera versión de Webpack.
- **ES Module:** desde la versión ES 6, esta es la nueva sintaxis que se utiliza para escribir módulos de forma nativa en el lenguaje JavaScript.
 - Existe una versión para empaquetadores modernos como Webpack o Rollup. Esta versión está optimizada para que el empaquetador la pueda analizar más rápidamente, de modo que solo conserve las partes de la librería que se utilizan en su aplicación.

- La versión "browser" permite que los archivos se carguen directamente en el navegador usando la etiqueta `<script>`, con el atributo `type` con el valor `module`.

■ Observación

La versión ES Module Browser le permite cargar la librería directamente en su página. Sin embargo, para evitar en su navegador problemas de compartición de recursos de origen cruzado (CORS-Cross-Origin Resource Sharing), es necesario utilizar un servidor local en su máquina. No podrá utilizar la librería Vue.js en una página `index.html` cargada en el navegador desde su explorador de archivos (URL en `file:///`).

Sin la parte Compiler, la librería es un 30 % más ligera. Por lo tanto, para proyectos grandes se recomienda usar componentes de un solo archivo con la extensión `.vue` y un empaquetador em moderno como Webpack, Rollup o Parcel, con el módulo `vue-loader`.

Si no está usando `vue-loader` o `vueify` (utiliza el empaquetador `Browserify` más antiguo) y desea usar una compilación con la parte Compiler, será necesario crear un alias para la build adecuada en la configuración de su empaquetador.

1.5 Mediante Vue-CLI

Si no desea pasar uno o dos días configurando un empaquetador para su proyecto de aplicación SPA, Vue.js proporciona un CLI (*Command Line Interface*) oficial, para iniciar un proyecto de desarrollo de aplicación SPA de manera rápida y sin configuración.

Vue-CLI es un paquete npm construido sobre el empaquetador Webpack y proporciona un comando `vue` en su terminal.

Vue-CLI es un sistema completo para desarrollar sus aplicaciones Vue rápidamente y en particular ofrece:

- una interfaz gráfica para crear y administrar un proyecto Vue.js;
- la recarga en caliente (*hot reloading*), lo que le permite probar su código sin esperar la compilación de una build por Webpack, durante el desarrollo;
- una colección de complementos oficiales que integran las mejores herramientas frontend: transpiler, linter, preprocesador CSS, herramientas de prueba unitaria, etc.

Para obtener más información, puede consultar el capítulo Crear y desplegar una aplicación con Vue CLI.

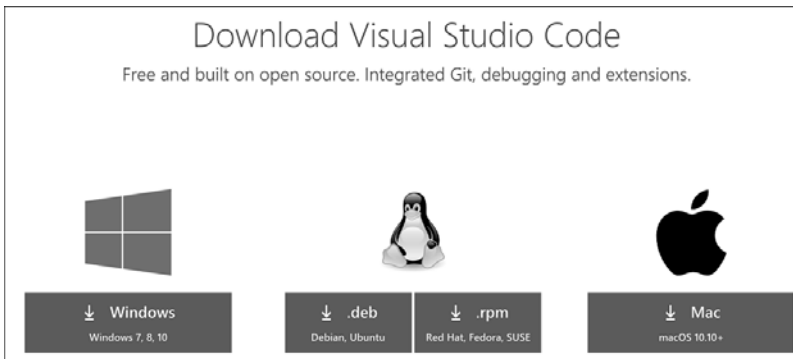
2. Herramientas de desarrollo

2.1 VS Code y sus plugins

2.1.1 Instalar y configurar VS Code

Visual Studio Code es un editor de código gratuito, publicado por Microsoft y uno de los más utilizados en la actualidad por los desarrolladores de Vue.js. Es muy ligero y potente al mismo tiempo. También es muy extensible, gracias a sus miles de plugins.

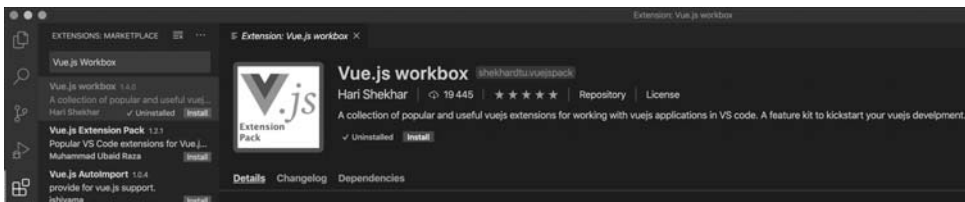
VS Code está disponible para descargar en Windows, Mac y Linux en:
<https://code.visualstudio.com/Download>



► Una vez instalado, ábralo.

► Pulse el botón  **Extensions**, a la izquierda de la ventana.

► Escriba **"Vue.js Workbox"**:



► Ahora pulse el botón verde **Install**.