

1. ¿Qué es Vue.js?

Es un framework para javascript que sirve principalmente para el desarrollo de la capa de visualización en diseño web basado en componentes. Este facilita el desarrollo frontend al ser flexible y progresivo, además de que es fácil de integrar con otras librerías y proyectos existentes (Vue.js., s.f.).

2. ¿Qué es un componente en Vue?

Cada componente es un objeto o instancia, los cuales pueden tener métodos y la lógica de como sirve cada aplicación que se llegue a crear, básicamente tienen cierta parte específica de la interfaz y de las funciones que realiza la página. Están compuestos de una sección template que tiene HTML principalmente, script que incluye los elementos de javascript y style que trabaja la parte de css. Los componentes se pueden utilizar como etiquetas HTML personalizadas, lo cual permite una fácil implementación (Gauchard, s.f.).

3. ¿Cuál es la estructura de un proyecto en Vue.js?

La estructura de un proyecto en Vue puede variar dependiendo de la implementación que se llegue a dar, pero hay ciertos elementos que suelen prevalecer en la mayoría de las ocasiones, los cuales serían (Roman, s.f.):

- Carpeta de raíz: es el directorio general del proyecto, donde se agregan todos los demás archivos y carpetas necesarias.
- Node_modules: carpeta donde se agregan librerías y demás dependencias del proyecto.
- Public: carpeta donde se suele agregar el favicon.ico de la página y el index.html del proyecto.
- Src: cuenta con dos subcarpetas principales, assets (que incluye los elementos que se referencian dentro del código, como imágenes principalmente o archivos multimedia) y components (donde se guardan todos los componentes que utilice el programador, son archivos .vue).
- Archivos .config.js: usualmente son frameworks de otras librerías que se trabajan, también se categorizan como “ficheros de configuración”.
- Views: guarda componentes de la página que se generan a partir de las rutas definidas dentro del proyecto, son “vistas” principales, que no suelen ser reutilizables como los componentes usuales de la carpeta components.
- Archivos README y .gitignore: archivos de uso general donde se describen los pasos de uso en el caso del README y en el .gitignore se agregan los archivos que se deben de ignorar a nivel de GitHub.

4. ¿Cual es la función del archivo App.vue?

Es un SFC (Single File Component) de Vue.js, el cual es el componente principal de la aplicación web. A partir de este es que se cargan los demás componentes que utilice el programador (Roman, s.f.).

5. ¿Qué significa la directiva v-if y v-for en Vue.js?

- V-if: se trabaja para la renderización condicional en el código y tiene la siguiente estructura (Vue.js., s.f.):
`<h1 v-if="booleano">Sí</h1>`
Se trabaja igual que un if de cualquier otro lenguaje, si “booleano” o cualquier otra condición que se indique dentro de las comillas se cumple, como, por ejemplo:

```
<div v-if="Math.random() > 0.5">
```

Si es mayor a 0.5 el rand este texto se despliega en la página

```
</div>
```

Lo que esté dentro de dicha etiqueta se desplegará, si no se cumple, no se despliega.

- V-for: se utiliza para iterar elementos dentro de etiquetas y generarlas automáticamente, un ejemplo sería el siguiente (Vue.js., s.f.):

```
<ul id="example-1">
```

```
<li v-for="item in items">
```

```
  {{ item.mensaje }}
```

```
</li>
```

```
</ul>
```

```
var example1 = new Vue({  
  el: '#example-1',  
  data: {  
    items: [  
      { mensaje: 'Primero' },  
      { mensaje: 'Segundo' }  
    ]  
  }  
})
```

En el ejemplo anterior, lo que pasaría es que a partir del for se crearían etiquetas de tipo “li” con el contenido de cada item de la lista de elementos en “items”, es decir, una etiqueta li con el texto “Primero” y una segunda etiqueta

del mismo estilo con “Segundo”. También, es posible llamar a index, el cual indica la iteración por la que va el for: v-for="(item, index) in items", para referenciar dicho número en algún método o para ingresarlo en el texto visible de la página.

6. ¿Qué es un componente global en Vue.js?

Los componentes globales se pueden usar en cualquier instancia del Vue raíz o en cualquier otro subcomponente del árbol de componentes de la instancia de Vue (Vue.js., s.f.).

7. ¿Qué es un componente local en Vue.js?

Los componentes locales son componentes que solo se pueden utilizar dentro del componente en el que se crean, ni siquiera en los subcomponentes que se deriven. Se registran explícitamente dentro del componente donde serán utilizados (Vue.js., s.f.).

Bibliografía

Gauchard, Y. (s. f.). *Vue.js—Utilizar los componentes*. Recuperado 9 de septiembre de 2024, de <https://www.ediciones-eni.com/libro/vue-js-desarrolle-aplicaciones-web-modernas-en-javascript-con-un-framework-progresivo-9782409034701/utilizar-los-componentes#:~:text=Definici%C3%B3n&text=Dividir%20la%20aplicaci%C3%B3n%20en%20varios,template%20de%20un%20componente%20padre>.

Roman, J. (s. f.). *Estructura de carpetas de VueJS - Javascript en español*. Recuperado 9 de septiembre de 2024, de <https://lenguajejs.com/vuejs/introduccion/estructura-carpetas/>

Vue.js. (s. f.-a). *Introducción—Vue.js*. Recuperado 9 de septiembre de 2024, de <https://es.vuejs.org/v2/guide/>

Vue.js. (s. f.-b). *Registro de componente—Vue. Js*. Recuperado 9 de septiembre de 2024, de <https://es.vuejs.org/v2/guide/components-registration>

Vue.js. (s. f.-c). *Renderización Condicional—Vue.js*. Recuperado 9 de septiembre de 2024, de <https://es.vuejs.org/v2/guide/conditional>

Vue.js. (s. f.-d). *Renderizado de lista—Vue.js*. Recuperado 9 de septiembre de 2024, de <https://es.vuejs.org/v2/guide/list#v-for-con-un-Objeto>