

[1. Que es vue.js 3](#_heading=h.30j0zll)

[2. Instalación 3](#_heading=h.1fob9te)

[3. Interpolación y directivas 6](#_heading=h.3znysh7)

[3.1. Directivas básicas 7](#_heading=h.2et92p0)

[3.1.1 v-text 7](#_heading=h.2s8eyo1)

[3.1.2 v-html 8](#_heading=h.17dp8vu)

[3.1.3 v-model 9](#_heading=h.3rdcrjn)

[3.2. Directivas condicionales 11](#_heading=h.44sinio)

[3.2.1 v-show 11](#_heading=h.2jxsxqh)

[3.2.2 v-if v-else v-else-if 13](#_heading=h.z337ya)

[3.3. Directivas de bucle 16](#_heading=h.3j2qqm3)

[3.3.1 v-for 16](#_heading=h.1y810tw)

[3.4. Directivas avanzadas 19](#_heading=h.4i7ojhp)

[3.4.1 v-bind 19](#_heading=h.1pxezwc)

[3.4.1.1 Dinamico 21](#_heading=h.49x2ik5)

[3.4.1.2 Objetos 22](#_heading=h.jm9ggzod5sdh)

[3.4.1.3 Class 23](#_heading=h.kfvexy91gzka)

[3.4.1.4 Style 25](#_heading=h.55jb3ee8skyn)

[5. Software Reviews 27](#_heading=h.2p2csry)

# 1. Que es vue.js

Vue.js es un framework de JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página. es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas **Single-Page Applications** cuando se utiliza en combinación con herramientas modernas y librerías de apoyo.

Consultar más en el siguiente video: <https://youtu.be/AqesL138vMA>

# 2. Instalación

Para este proceso es recomendable tener las bases de javaScript y el tema de módulos claros, realizaremos la instalación por medio la URL de CDN que nos ofrece la documentación oficial de vue.js, creamos la siguiente configuración básica para ello



En el archivo index.html, realizaremos la estructura básica de html y vinculamos el archivo app.js, en el body creamos un div









# 3. Interpolación y directivas



Cuando se trata de mostrar los datos del modelo a la vista, hablamos de interpolación. La interpolación más básica en Vue.js es usar "bigotes", es decir, pares de llaves con una expresión JavaScript para el contenido textual en su DOM.

Las directivas de Vue son atributos especiales que se colocan en las etiquetas HTML y están prefijados por v-, como por ejemplo, v-for, v-bind o v-on, entre muchas otras. Estas directivas permiten realizar acciones dinámicas potentes (bucles, condicionales, etc...) que no se pueden realizar en HTML por sí solo,

Dichas directivas están formadas por varias partes:

* **Directiva:** El nombre de la directiva, que a veces, es posible abreviarlo con un carácter.
* **Argumento:** En ciertas directivas se indica un parámetro.
* **Modificador:** En ciertas directivas se puede modificar el comportamiento.
* **Valor:** En ciertas directivas, se requiere establecer un valor. Se escribe como el valor de un atributo HTML.

Existen varias directivas en Vue, divididos en varios grupos:

| **Tipos de directivas** | **Ejemplos** | **Definición** |
| --- | --- | --- |
| Directivas básicas | * v-text * v-html * v-model | Permite realizar tareas simples. |
| Directivas condicionales | * v-show * v-if * v-else * v-else-if | Permiten realizar acciones según condiciones. |
| Directivas de bucles | * v-for | Permiten realizar operaciones varias veces |
| Directivas avanzadas | * v-bind * v-on * v-slot | Permiten realizar tareas más específicas. |

## 3.1. Directivas básicas

Existen varias directivas en Vue, las he dividido en varios grupos e iremos profundizando en ellas si deseas conocer más directiva consulte el siguiente enlace de la documentación oficial de vue.js [Directivas integradas](https://vuejs.org/api/built-in-directives.html)

### 3.1.1 v-text

Vue nos permite hacer referencia a las variables (o métodos, propiedades computadas, etc...) simplemente escribiendo entre {{ dobles llaves }}, lo que se conoce como sintaxis de plantillas o formato mustache, en referencia a uno de los primeros sistemas de plantillas de Javascript: [mustache](https://mustache.github.io/).







### 3.1.2 v-html

Si quisiéramos incluir código HTML en una de esas variables, se mostrarían las etiquetas literalmente. La directiva v-html funciona exactamente igual que v-text, solo que en lugar de mostrar las etiquetas, las procesa y renderiza.







### 3.1.3 v-model

Una de las directivas básicas más interesantes es v-model. Esta directiva permite crear un modelo de datos bidireccional entre un elemento HTML concreto y una variable de Vue. ¿Qué significa esto? Significa que podemos sincronizar el contenido de una variable con el contenido que tenga un elemento HTML <input> (por ejemplo) en su atributo value.









## 3.2. Directivas condicionales

Condiciones directamente en nuestro etiquetado HTML, sin tener que recurrir a lógica Javascript o realizar código más complejo. Esto nos permite crear múltiples posibilidades directamente en nuestra parte del template, haciendo el código mucho más sencillo.

### 3.2.1 v-show

La directiva v-show se define como una directiva que muestra u oculta algo, dependiendo del valor que tenga: si es true lo muestra, si es false lo oculta con CSS













### 3.2.2 v-if v-else v-else-if

La directiva v-if hace algo muy parecido a la directiva v-show pero tiene una diferencia principal: mientras que v-show mantiene el código en el HTML y lo oculta o muestra con CSS dependiendo del valor indicado, v-if evaluará la condición y si es falsa no la colocará en el HTML. Además, con v-if podemos usar v-else y/o v-else-if, por lo que es algo más potente.



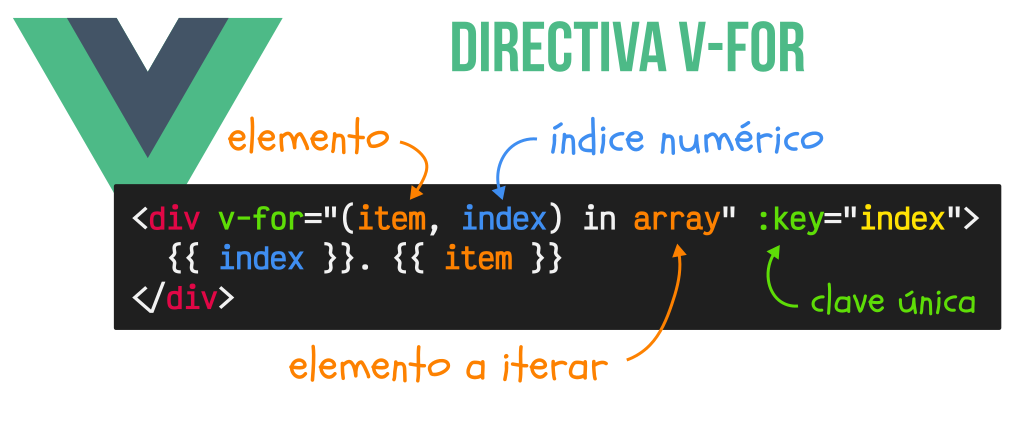








## 3.3. Directivas de bucle



La estructura de un v-for es muy sencilla y se basa en la posibilidad de crear un bucle for desde las templates de código HTML de Vue. Teniendo en cuenta su sintaxis, puedes crear código complejo en muy pocas líneas, basado en el bucle forEach de Javascript

### 3.3.1 v-for

| **Ejemplo** | **Configuración** | **Definición** |
| --- | --- | --- |
| v-for="(**(Configuración**) in array" :key="id" | (val, id) | Iteramos por cada uno de los elementos del array. |
| v-for="(**(Configuración)** in object" :key="id" | (val, name, id) | Iteramos por cada una de las propiedades del objeto. |
| v-for="(**(Configuración)** in number" :key="id" | (i, id) | Iteramos en un bucle de 1 al número number. |
| v-for="(**(Configuración)** in string" :key="id" | (char, id) | Iteramos por cada carácter del string. |

****

****

****

****



## 3.4. Directivas avanzadas

### 3.4.1 v-bind

La directiva v-bind (dos puntos) es una de las directivas más utilizadas y populares de Vue. Esta directiva permite enlazar (bindear) una variable de Vue con un atributo específico de una etiqueta HTML. De esta forma, podemos colocar como valor de un atributo HTML el contenido que tengamos almacenado en una variable de la lógica de Javascript.









#### 3.4.1.1 Dinamico

En algunos casos que nos interesen, podríamos utilizar los corchetes para indicar que recoja el atributo de una variable, pudiendo hacer este proceso aún más dinámico







#### 3.4.1.2 Objetos

Enlazar completamente una etiqueta HTML con un objeto que tenemos en las variables de Vue. De esta forma actualizamos la etiqueta completa simplemente actualizando el desde Javascript.







#### 3.4.1.3 Class

Ahora que sabemos utilizar la directiva v-bind o : con atributos genéricos de una etiqueta, vamos a ver casos especiales con el atributo class de una etiqueta, utilizada para añadir clases CSS a un elemento HTML.











#### 3.4.1.4 Style

Si lo que nos interesa es aplicar propiedades específicas en lugar de clases CSS, podemos hacer uso del atributo style, de la misma forma que hicimos con :class. De esta forma, añadimos las propiedades CSS que existan en la variable a la que hacemos referencia









### 3.4.2 v-on

### 

### La directiva v-on (abreviada como @) es una directiva utilizada para gestionar los eventos del DOM desde los templates de Vue, haciendo más cómodo y práctico su utilización, permitiendo escribir mucho menos código. Así pues, nos ahorramos realizar .addEventListener(), y en su lugar definimos las acciones de forma mucho más directa.

### 

### 

# 

# 

# 5. Software Reviews

Instalación de git videos

<https://www.youtube.com/shorts/xffjZCsu3N0>