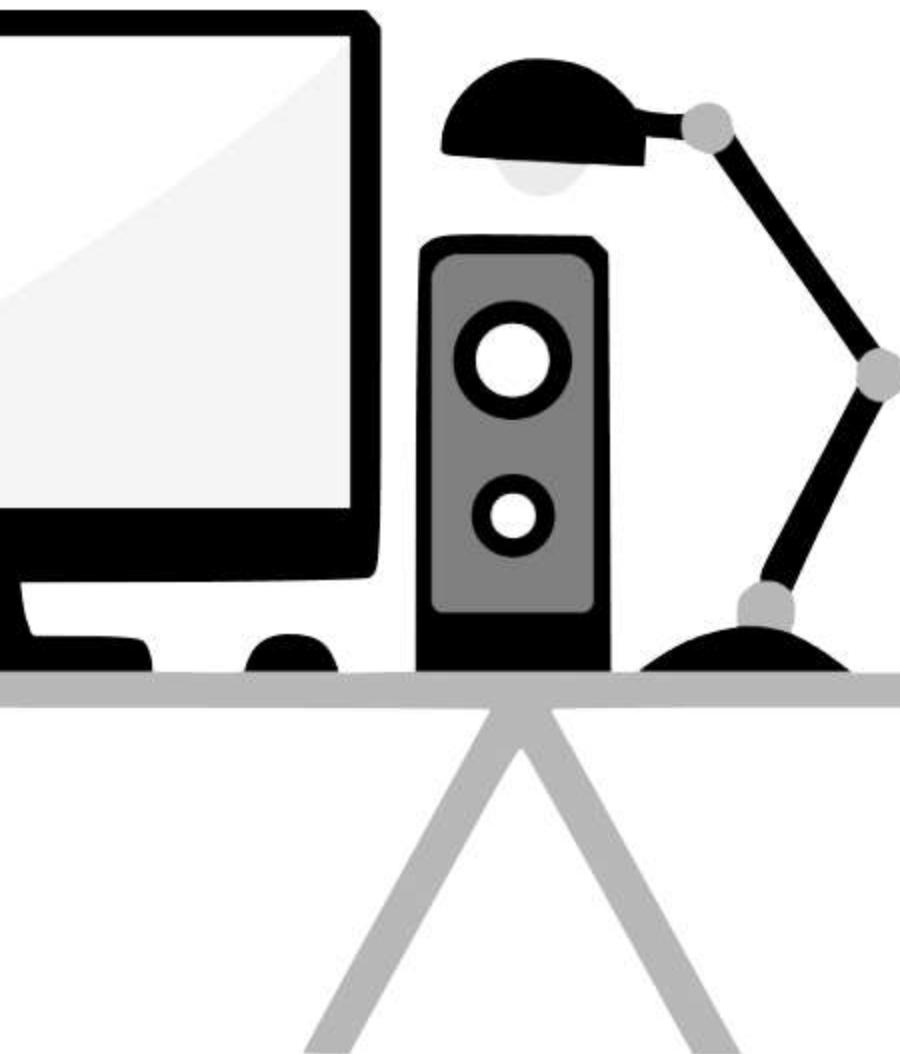




*sustantiva*  
C O N • S E N T I D O

**Lección 1:**  
**Introducción al Framework VueJs**

# Contenidos



## Introducción al Framework VueJs

- Qué es un Framework Progresivo
- Beneficios de utilizar un Framework
- Qué es una SPA
- Qué es un componente web
- Qué es reactividad
- Otras librerías o frameworks orientadas a componentes
- Comparación entre React, Angular y Vue
- Qué es Vue Js, por qué usarlo, ¿quiénes lo utilizan?
- Alternativas a Vue JS
- El patrón de diseño MVVM
- Formas de instalar Vue en una aplicación web
- Setup de una aplicación con Vue CLI y sus características agregadas.
- Herramientas de desarrollo: Webpack, Vue.js Devtools
- El archivo package.json
- Estructura básica de un componente Vue
- Morfología de un componente Vue: HTML, JavaScript y CSS en un mismo archivo
- Montar componente en elemento HTML
- Conociendo el objeto data.
- Diferencias entre One way y Two way binding
- Formularios y directiva model

# ¿Qué es un framework?

Un Framework es una herramienta que proporciona componentes listos para usar o soluciones que se personalizan para acelerar el desarrollo.

Un framework puede incluir programas de soporte, compiladores, bibliotecas de códigos, conjuntos de herramientas y API para desarrollar software y crear sistemas.

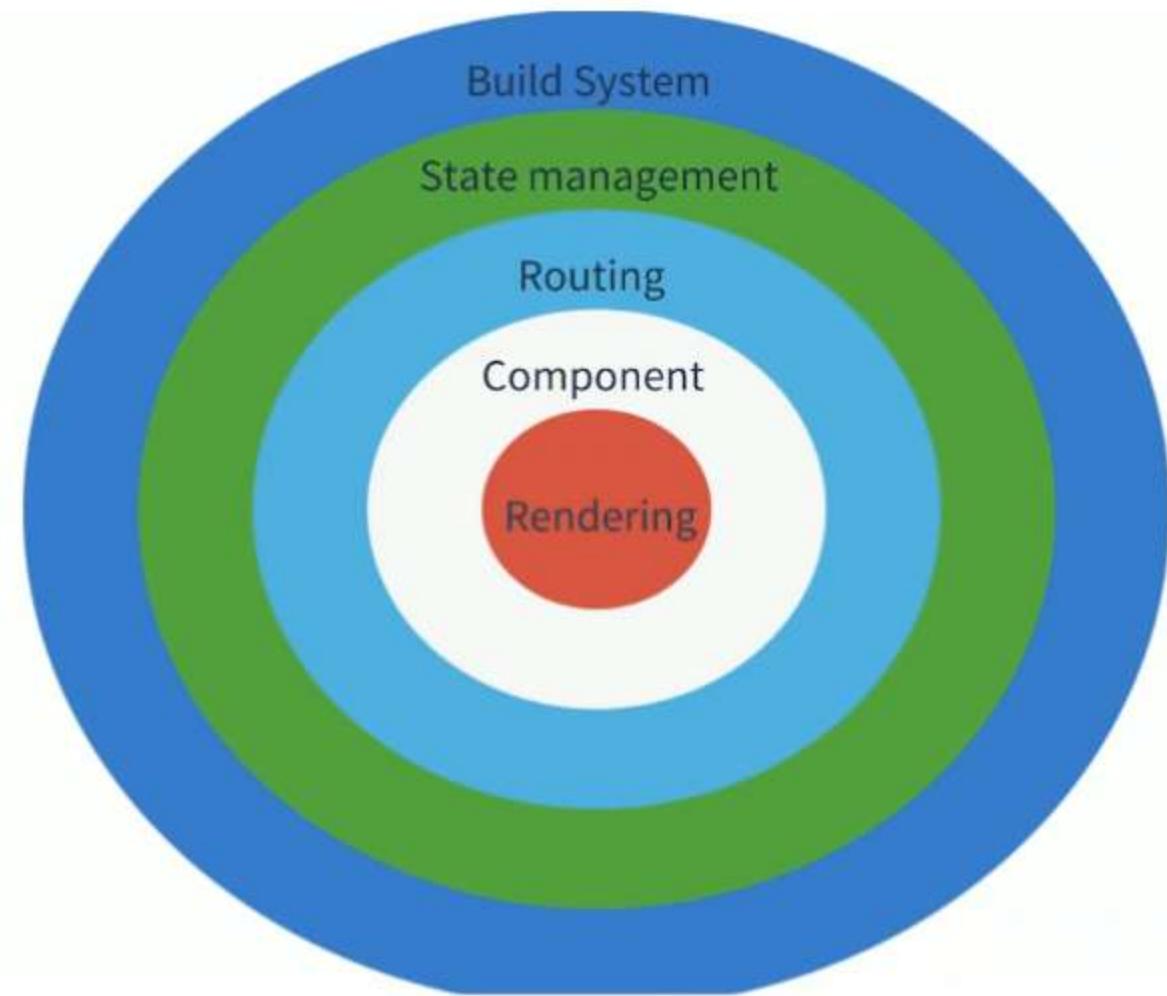
El uso de un framework te permite concentrarte en la funcionalidad de alto nivel de la aplicación ya que aquél ocupa de cualquier funcionalidad de bajo nivel.



# ¿Qué es un framework progresivo?

La palabra "progresivo" significa que el framework se implementa como un marcado adicional a HTML. Es básicamente un modelo de plantilla vinculado a un modelo de datos. Si se actualiza el modelo, se actualiza el HTML del navegador, es decir, reacciona a las actualizaciones del modelo.

En un framework progresivo puedes usar el framework y agregarle progresivamente más partes, es decir, comienzas con algo pequeño y gradualmente construyes algo más grande. Por ejemplo, puedes comenzar simplemente usando la funcionalidad de vista, luego agregarle el enrutador, luego agregar el estado, y así sucesivamente.



# Beneficios de los frameworks

Como ya se mencionó, los frameworks implican los siguientes beneficios:

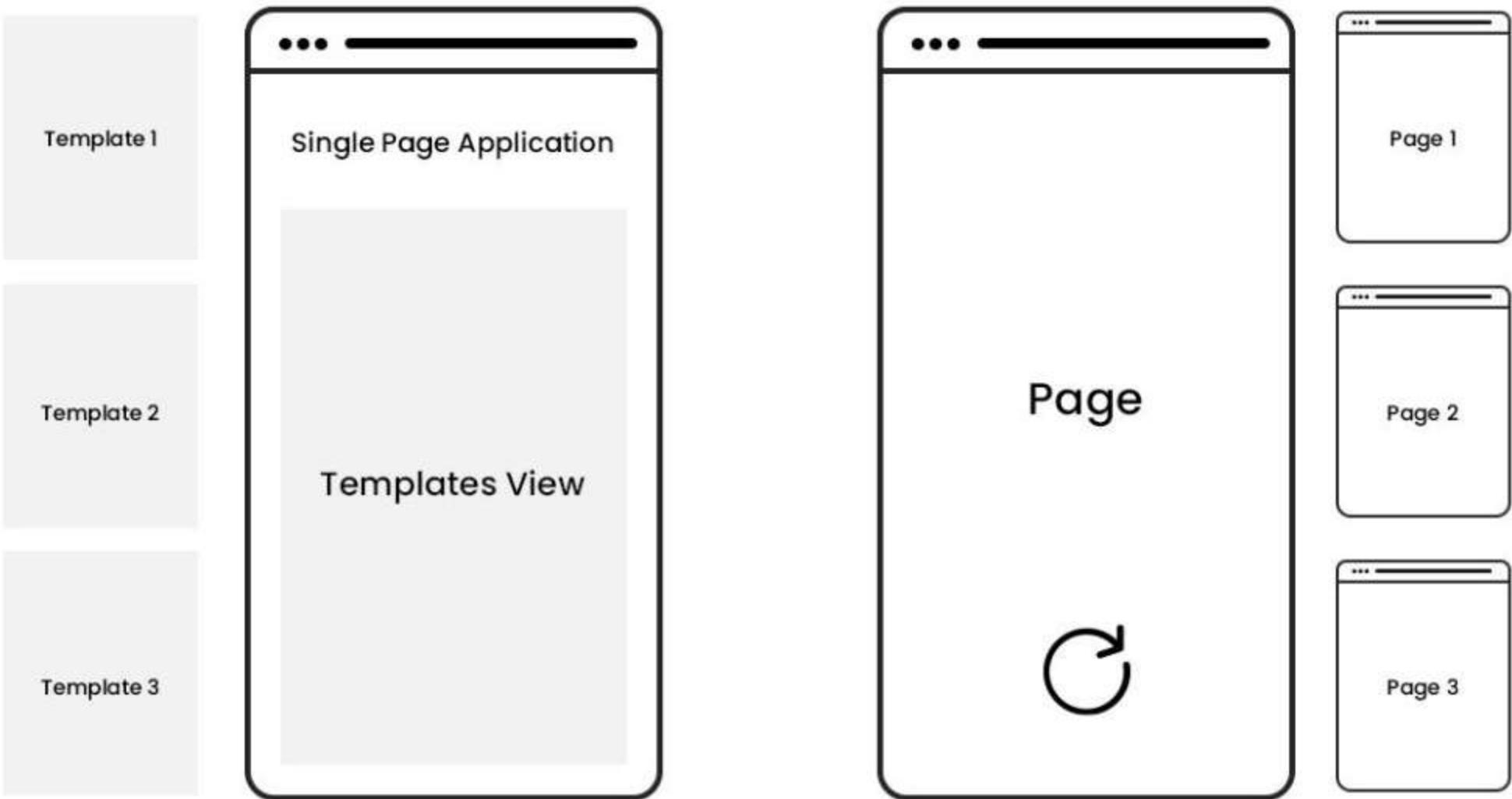
- Facilita el proceso de desarrollo.
- Facilita el Debugging y la mantención de la aplicación.
- Reduce la extensión del código.
- Mejora las características de las bases de datos.
- Refuerzan la seguridad.



# ¿Qué es una SPA?

Una SPA (Single-page application) es una aplicación web que carga solo un documento web y luego actualiza el contenido del cuerpo de ese documento a través de las API de JavaScript, como XMLHttpRequest y Fetch, cuando se muestra contenido diferente.

Por lo tanto, esto permite a los usuarios usar sitios web sin cargar páginas completamente nuevas desde el servidor, lo que puede generar una experiencia más dinámica ya que el archivo permanece constante mientras que la lógica del JavaScript del lado del cliente cambia solo para actualizar la vista. Esto se debe a que las solicitudes de datos se efectúan más rápido que la carga de los archivos por el servidor, sumado al tiempo que tarda para luego mostrarse en el navegador.



# ¿Qué es un componente web?

Los componentes web son un conjunto de API de plataforma web que te permiten crear nuevas etiquetas HTML encapsuladas, reutilizables y personalizadas para usar en páginas web y aplicaciones web.

Por ejemplo, podemos crear nuestro nuevo componente llamado `<my-web-component>`, con su estilo y funcionalidad únicos, y usarlo en cualquier framework o biblioteca de JavaScript.



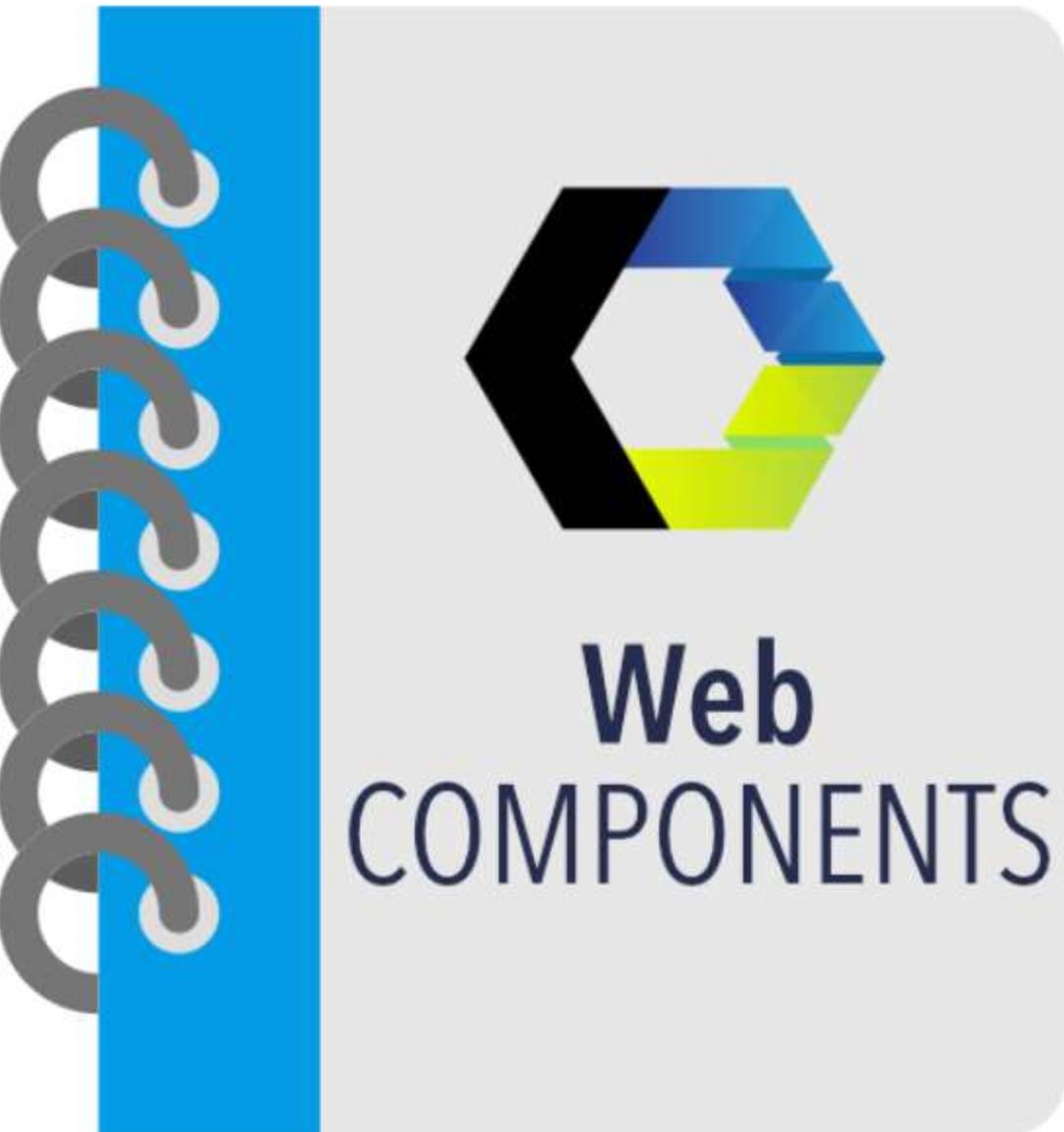
# Ventaja de un componente Web

Los componentes web constan de tres tecnologías principales, que se pueden usar juntas para crear elementos personalizados versátiles con funcionalidad encapsulada y que se pueden reutilizar donde quieras sin temor a colisiones de código.

**Elementos personalizados:** un conjunto de API de JavaScript que te permiten definir elementos personalizados y su comportamiento, que luego se pueden usar como se deseé en tu interfaz desarrollada.

**Shadow DOM:** se representa por separado del DOM del documento principal y controla diferentes funcionalidades asociadas. De esta manera, se pueden escribir y diseñar sin temor a colisionar con otras partes del documento.

**Plantillas HTML:** los elementos `<template>` y `<slot>` te permiten escribir plantillas que no se muestran en la página representada y se pueden reutilizar varias veces.



# Ventaja de un componente Web

Los componentes web te permiten crear componentes reutilizables basados en HTML, JavaScript y CSS, separando su funcionalidad y diseño del resto del código, por lo que es una buena herramienta de encapsulación. Entre sus ventajas, tenemos:

- **No necesitas agregar librerías para utilizar los componentes web.**
- **Son fáciles de reutilizar y compartir.**
- **Muy útiles para realizar tareas específicas.**
- **No necesitas de dependencias complejas.**
- **Código muy legible.**

Sin embargo, los componentes web muy pesados no son muy amigables con los motores de búsquedas (SEO) debido a una deficiente indexación ineficiente. Este último es el proceso por el cual Google, por ejemplo, anexa una página web a su índice para mostrarla en los resultados de una búsqueda.



# ¿Qué es reactividad?

La programación reactiva es un enfoque de diseño que utiliza lógica de programación asincrónica para manejar ajustes en tiempo real a información típicamente estática.

El concepto fundamental de la programación reactiva son las secuencias ordenadas en el tiempo de notificaciones de eventos conectados. Por lo general, una transmisión comenzará con un observador, ya sea una sección de código dentro de una aplicación que monitorea las condiciones relacionadas con la aplicación o un dispositivo. Una secuencia a veces se representa desde el observador hasta uno o más controladores hasta que se procesa, culminando en un estado incorrecto o se bifurca en secuencias derivadas. La programación reactiva se trata de construir los observadores y controladores y enhebrar la secuencia según sea necesario.

Cool! What does  
that mean exactly?



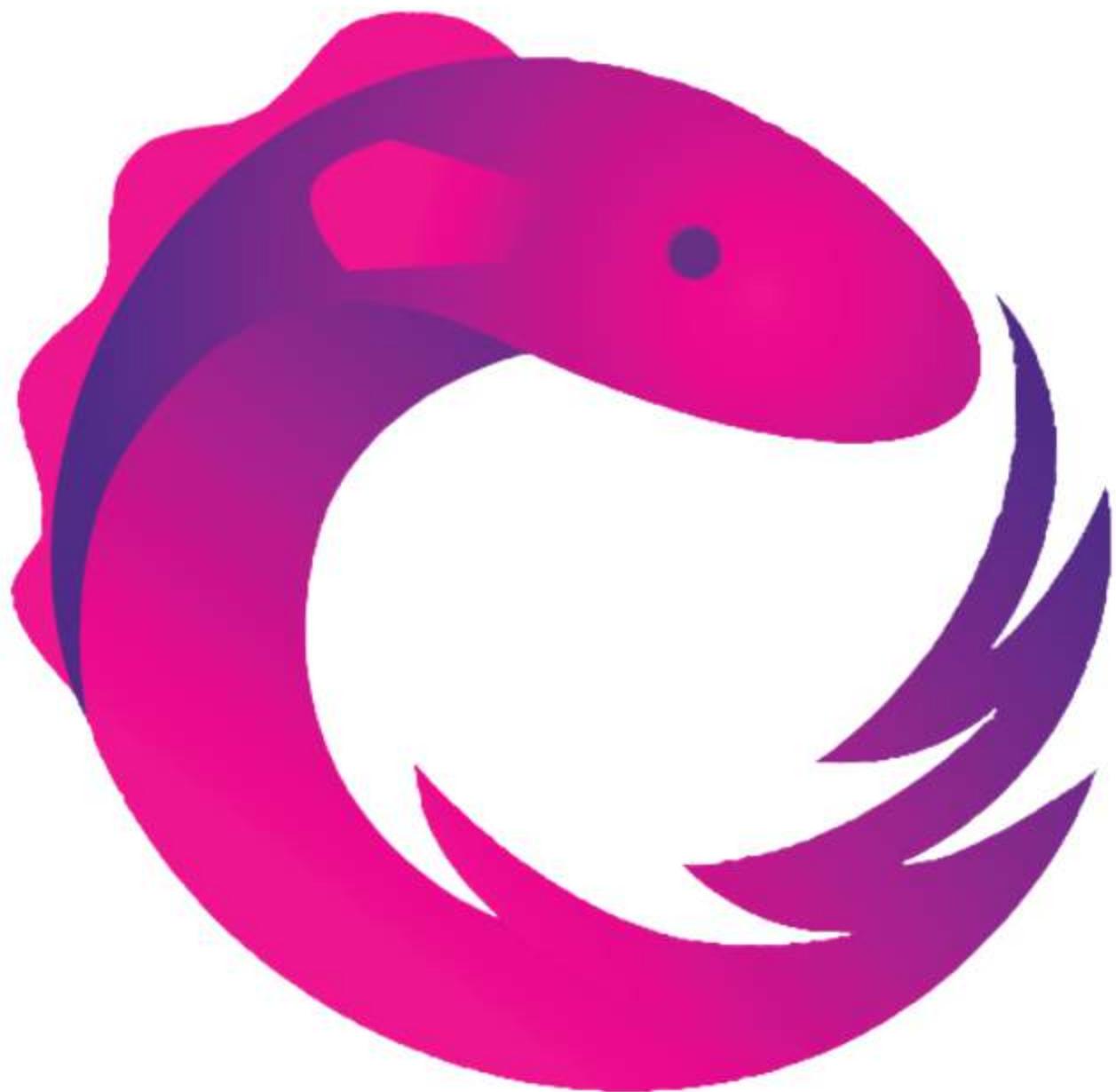
# ¿Qué es reactividad?

En otras palabras, es una programación con flujos de datos asincrónicos que transmite datos a un consumidor a medida que están disponibles, lo que permite a los desarrolladores diseñar un código que responda de forma rápida y asincrónica.

Las librerías reactivas te permiten:

- Expresar fácilmente los flujos de datos.
- Evaluar automáticamente los flujos de datos y propagar los cambios de datos.

RxJS es una biblioteca para la programación reactiva que usa “Observables”, para facilitar la composición de código asíncrono o basado en callbacks.



# Frameworks orientados a componentes

Angular

React

Vue



# Comparación entre Angular, React y Vue

Tecnología	Angular	React	Vue
Desarrollador	Google	Facebook	OpenSource
Tipo	Framework MVC escrito en JS	Librería JavaScript (Vista MVC, requiere Flux en su arquitectura)	Librería JavaScript
Concepto	Incorpora JS a Html Manipula el DOM real Renderización al lado cliente	Incorpora JS a Html Trabaja con un DOM virtual Renderización al lado servidor	Emplea HTML basado en plantillas facilitando la migración entre aplicaciones
Enlace de datos	Bidireccional	Unidireccional	Uni (por defecto) y bidireccional
Dependencias	Administra dependencias automáticamente	Requiere herramientas adicionales para administrar dependencias	Administra dependencias automáticamente
Lenguaje	JS + HTML	JS + JSX	JS+HTML basado en plantillas
Adecuado para...	SPA que actualiza una vista a la vez	SPA que actualizan varias vistas a la vez	SPA no tan pesadas que requieren desarrollo rápido

# Framework Vue

Vue (pronunciado /vju:/, como view) es un framework de JavaScript para construir interfaces de usuario. Se basa en HTML, CSS y JavaScript estándar y proporciona un modelo de programación declarativo y basado en componentes que lo ayuda a desarrollar interfaces de usuario de manera eficiente, ya sean simples o complejas.

**Representación declarativa:** Vue amplía el HTML estándar con una sintaxis de plantilla que nos permite describir de forma declarativa la salida HTML en función del estado de JavaScript.

**Reactividad:** Vue rastrea automáticamente los cambios de estado de JavaScript y actualiza de manera eficiente el DOM cuando ocurren cambios.



# Ventajas de Vue

Entre las ventajas de Vue, se puede mencionar:

- **Simplicidad:** Puedes escribir tu aplicación Vue dentro de 5-6 líneas de código.
- **Arquitectura basada en componentes:** Esto significa que todo el código de la aplicación frontend se puede dividir en componentes independientes.
- **Código Reutilizable:** El enfoque basado en componentes de Vue permite la creación de componentes de archivo reutilizables. Dentro de un componente, la plantilla, la lógica y los estilos están inherentemente acoplados.
- **Alto rendimiento:** Vue js es extremadamente liviano y ocupa alrededor de 20 KB. Con características tan mínimas, Vue js es muy rápido de instalar y puedes comenzar a trabajar en cuestión de minutos. Además, el bajo tiempo de carga de la página también afecta positivamente la clasificación SEO de sus páginas web.
- **Documentación muy detallada.**
- Vue js usa una **representación virtual del DOM real** de una página web, la cual es creada usando objetos Javascript. Por lo tanto, los objetos DOM se pueden representar fácilmente sin modificar y actualizar todo el árbol cada vez.
- **Enlace bidireccional:** Un desarrollador puede realizar cambios en la capa Modelo o en la capa Vista, y Vue.js realizará los cambios en las otras capas, manteniendo el código sincronizados en toda la aplicación.
- **Representación declarativa** y puedes escribir plantillas utilizando **JavaScript, HTML o JSX**, una extensión de sintaxis de JavaScript basada en XML.

# Algunas desventajas de Vue...

## Falta de complementos

Vue es desarrollado de forma independiente por su propietario Evan You y algunos otros entusiastas de Vue sin el respaldo de gigantes tecnológicos como Facebook (React) y Google (Angular). La velocidad de lanzamiento de nuevos complementos es sustancialmente menor y hay una falta de complementos de uso común.

## Flexibilidad de código excesiva

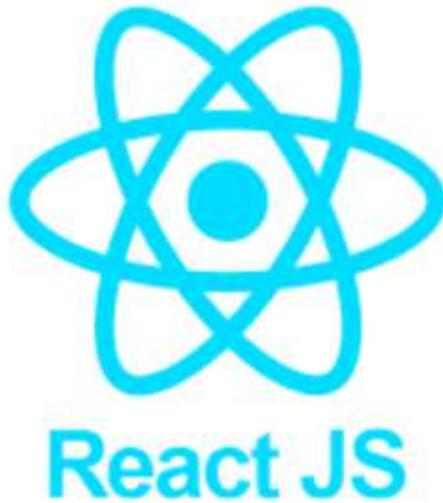
Si bien la flexibilidad del código es algo bueno, también tiene sus desventajas, principalmente en lo que respecta a la irregularidad y los errores del código.



# Alternativas a Vue

## React

React de Facebook sobresale en las deficiencias de Vue, es decir, proyectos web a gran escala. Si bien se diseñó originalmente para crear interfaces de usuario de aplicaciones, ahora se puede usar para diseñar y mantener ecosistemas JavaScript frontend con facilidad.



## jQuery

Podría decirse que jQuery es una de las bibliotecas JS más implementadas y se usa en más del 70% de los 10 millones de sitios web más populares en Internet. jQuery envuelve tareas que requieren múltiples líneas de código como manejo de eventos, animaciones CSS y Ajax en una sola línea de código.



## Mithril

Mithril es un pequeño framework de JavaScript que se utiliza para crear aplicaciones web de una sola página del lado del cliente. Incluye paquetes más pequeños que Vue (<10 kb), pero ofrece un rendimiento similar junto con XHR y capacidades de enrutamiento listas para usar.



# Alternativas a Vue

## Svelte

A diferencia de React y Vue, que completan la mayoría de las tareas en el navegador, Svelte no funciona en el navegador; en cambio, cambia el trabajo a un paso de compilación y actualiza el DOM al instante, ofreciendo una mejor velocidad, reactividad y escalabilidad.



SVELTE



Inferno

## Infiero JS

Infiero JS es una biblioteca de JavaScript de pila completa similar a React que se utiliza para desarrollar interfaces de usuario efectivas tanto en el cliente como en el servidor. En lugar de seguir los patrones MVC o MVVM, Inferno también usa el enfoque basado en componentes como Vue para hacer que el código sea predecible y reutilizable.

## Angular

Angular de Google es una plataforma y un framework del lado del cliente que se utiliza para crear aplicaciones web, de escritorio y móviles de una sola página utilizando TypeScript, un superconjunto escrito de JavaScript y HTML.



ANGULARJS  
by Google

# Alternativas a Vue

## Nest Js

NestJS es posiblemente uno de los frameworks NodeJS de más rápido crecimiento entre las alternativas de Vue. Escrito en TypeScript, NestJS es de código abierto, progresivo, versátil y extensible en el entorno NodeJS que se utiliza para crear sistemas de back-end atractivos y exigentes.

Está fuertemente influenciado por React, Vue y Angular y ofrece varias características como la inyección de dependencia lista para usar, lo que permite a los desarrolladores escribir fácilmente aplicaciones escalables y débilmente acopladas.



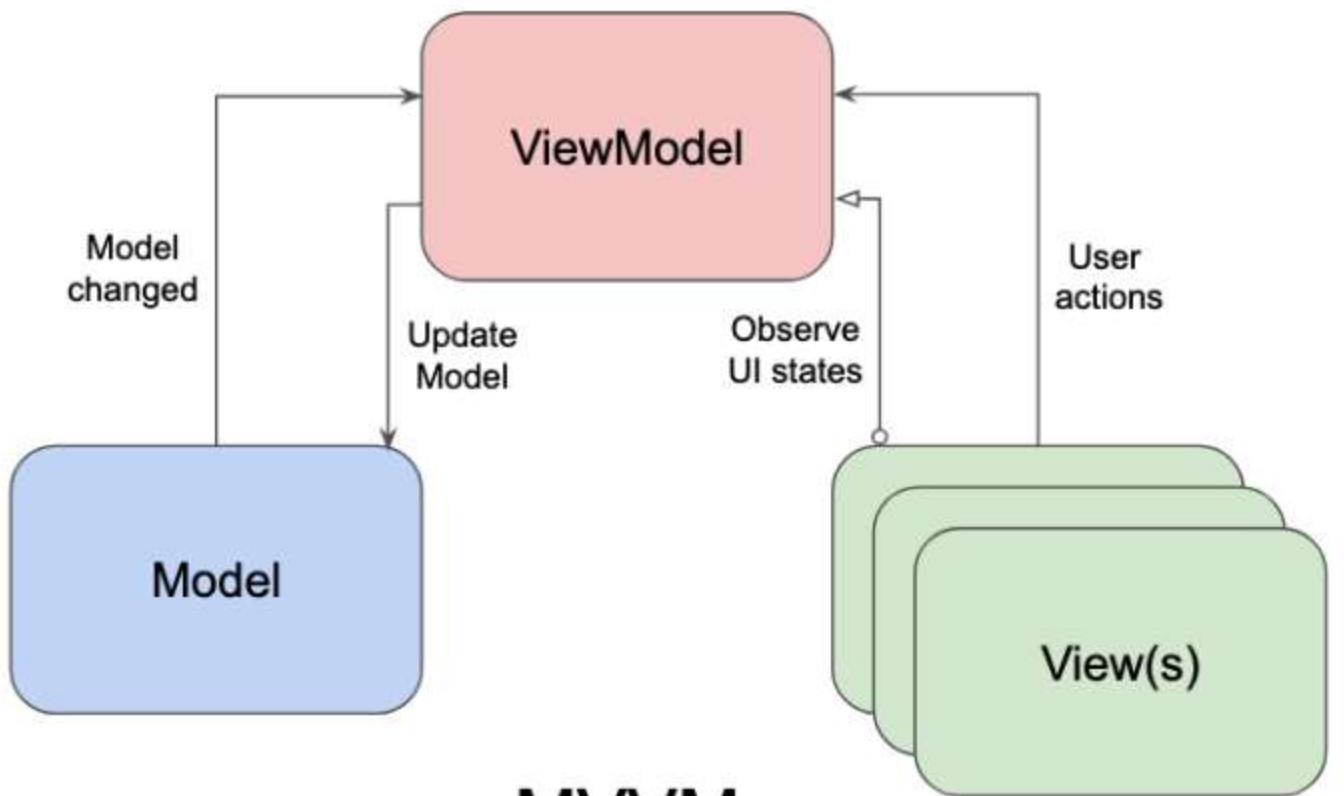
# Patrón de diseño MVVM

**Model — View — ViewModel (MVVM)** es el patrón de arquitectura de software reconocido en la industria que supera todos los inconvenientes de los patrones de diseño MVP y MVC. **MVVM** sugiere separar la lógica de presentación de datos (vistas o interfaz de usuario) de la parte lógica empresarial central de la aplicación.

**Modelo:** esta capa es responsable de la abstracción de las fuentes de datos. Model y ViewModel trabajan juntos para obtener y guardar los datos.

**Vista:** El propósito de esta capa es informar al ViewModel sobre la acción del usuario. Esta capa observa el ViewModel y no contiene ningún tipo de lógica de aplicación.

**ViewModel:** Expone esos flujos de datos que son relevantes para la Vista. Además, sirve como enlace entre el Modelo y la Vista.



# Instalar Vue: CDN

Puedes incluir el script de Vue desde una **CDN** en un archivo HTML, copia y pega el siguiente script en la sección **HEAD** de la plantilla de tu proyecto:

```
<script src="https://unpkg.com/vue@3.0.3/dist/vue.global.prod.js"></script>
```

Para incluir la versión de desarrollo debes eliminar el sufijo .prod de la extensión del archivo mencionada anteriormente.

```
<script src="https://unpkg.com/vue@3.0.3/dist/vue.global.js"></script>
```

# Instalar Vue: NPM/YARN

Yarn (Yet Another Resource Negotiator) y NPM (Node Package Manager) son administradores de paquetes utilizados para la codificación de JavaScript. Trabajan con Node.js, que sirve para ayudar a los usuarios a desarrollar y ejecutar código JavaScript fuera de un navegador web. Primero debes instalar Node.js descargando la versión apropiada para tu equipo desde el siguiente link:

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

Luego, para crear un proyecto con Vue y npm o yarn, tendrás que instalar la herramienta de línea de comandos VUE CLI, con la que podrás compilar el código Vue en código JavaScript. Para instalar Vue en tu sistema, abre una ventana del terminal e instálalo con uno de estos dos comandos, dependiendo del gestor de paquetes que utilices:



```
# Instalación de Vue CLI con NPM
npm i -g @vue/cli @vue/cli-service-global
# Instalación de Vue CLI con Yarn
yarn global add @vue/cli @vue/cli-service-global
```

# Setup de aplicación

Una vez hayas instalado Vue CLI, ya podrás crear un proyecto usando el comando **vue create**. Por ejemplo, para crear el proyecto  **proyecto-nuevo**, tendrás que desplazarte hasta el directorio en el que quieras crearlo y usar el siguiente comando:

```
vue create proyecto-nuevo
```

Se te preguntará si prefieres usar **Vue 2** con las opciones por defecto, **Vue 3** con las opciones por defecto o si por el contrario prefieres escoger los componentes a instalar. Selecciona la opción que prefieras, que seguramente será Vue 3, ya que instalará los componentes por defecto de la última versión de Vue.

# Características Agregadas de Vue

**Babel** es una cadena de herramientas que se utiliza principalmente para convertir el código ECMAScript 2015+ en una versión retrocompatible de JavaScript en navegadores o entornos actuales y antiguos. Estas son las principales cosas que Babel puede hacer:

- Sintaxis de transformación
- Funciones de Polyfill que faltan en su entorno de destino (a través de un Polyfill de terceros como core-js). Un polyfill es un fragmento de código que se utiliza para proporcionar una funcionalidad moderna en navegadores más antiguos que no lo admiten de forma nativa.
- Transformaciones de código fuente (codemods)



# Características Agregadas de Vue

Las **PWA (Progressive Web App)** son aplicaciones web desarrolladas utilizando una serie de tecnologías específicas que les permiten aprovechar las funciones de tanto las aplicaciones webs y nativas. Por ejemplo, es mucho más fácil y rápido visitar un sitio web que instalar una aplicación, y también se pueden compartir aplicaciones web enviando un enlace. Por otro lado, las aplicaciones nativas se integran mejor con el sistema operativo y, por lo tanto, ofrecen una experiencia más fluida para los usuarios. Puedes instalar una aplicación nativa para que funcione sin conexión.

Muchos PWA se crean utilizando frameworks de JavaScript modernos (como **Vue**). Además, **Vue CLI** ofrece un complemento conveniente para convertir una app en PWA, que si tuviera que hacerse manualmente, sería una tarea tediosa.

PWA

# Características Agregadas de Vue

El enrutamiento en el lado del servidor significa que el servidor envía una respuesta basada en la ruta URL que está visitando el usuario. Cuando hacemos clic en un enlace en una aplicación web tradicional renderizada por un servidor, el navegador recibe una respuesta HTML del servidor y vuelve a cargar toda la página con el nuevo HTML.

**Vue Router** es el enrutador oficial de Vue.js. Se integra profundamente con el núcleo de Vue.js para facilitar la creación de aplicaciones de una sola página con Vue.js. Sus características incluyen:

- Mapeo de rutas anidadas
- Enrutamiento dinámico
- Configuración de enrutador modular basada en componentes
- Parámetros de ruta, consulta.
- Control de navegación detallado
- Enlaces con clases CSS activas automáticas
- Modo historial HTML5 o modo hash
- Comportamiento de desplazamiento personalizable
- Codificación adecuada para URL



**Vue Router**

# Características Agregadas de Vue

**Un preprocesador de CSS** es una herramienta que se utiliza para ampliar la funcionalidad básica del CSS. Nos ayuda a usar una sintaxis lógica compleja como: variables, funciones, combinaciones, anidamiento de código y herencia, por nombrar algunos, potenciando el CSS estándar.

Sin embargo, los navegadores solo pueden entender el código CSS nativo y no podrán interpretar la sintaxis del preprocesador CSS. Por lo tanto, la sintaxis del preprocesador debe compilarse primero en la sintaxis CSS nativa que luego pueden interpretar los navegadores. Si bien los diferentes preprocesadores tienen sus propias sintaxis únicas, finalmente todos ellos se compilan en el mismo código CSS nativo.

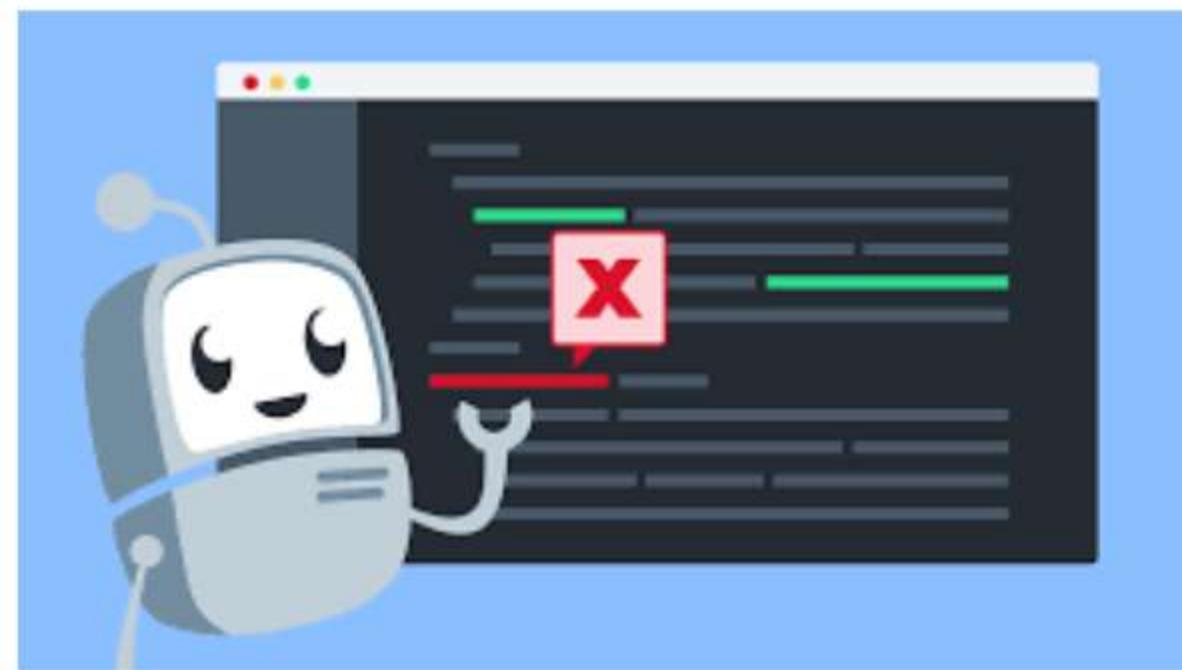
Los 3 preprocesadores CSS más populares que actualmente utilizan los desarrolladores de todo el mundo son Sass, LESS y Stylus. **Los proyectos de Vue CLI vienen con soporte para PostCSS, módulos CSS y preprocesadores, incluidos Sass, Less y Stylus.** Puede seleccionar preprocesadores (Sass/Less/Stylus) al crear el proyecto.



# Características Agregadas de Vue

**Linting** es el proceso automatizado de análisis estático del código en busca de posibles errores. Por lo general, linting puede hacer cosas como detectar errores de sintaxis, como una coma faltante, o advertirle que un archivo tiene una variable definida que nunca se usa. Linting no salvará el código de ningún defecto de lógica comercial, pero se asegurará de que sea sintácticamente preciso y siga algunas de las mejores prácticas.

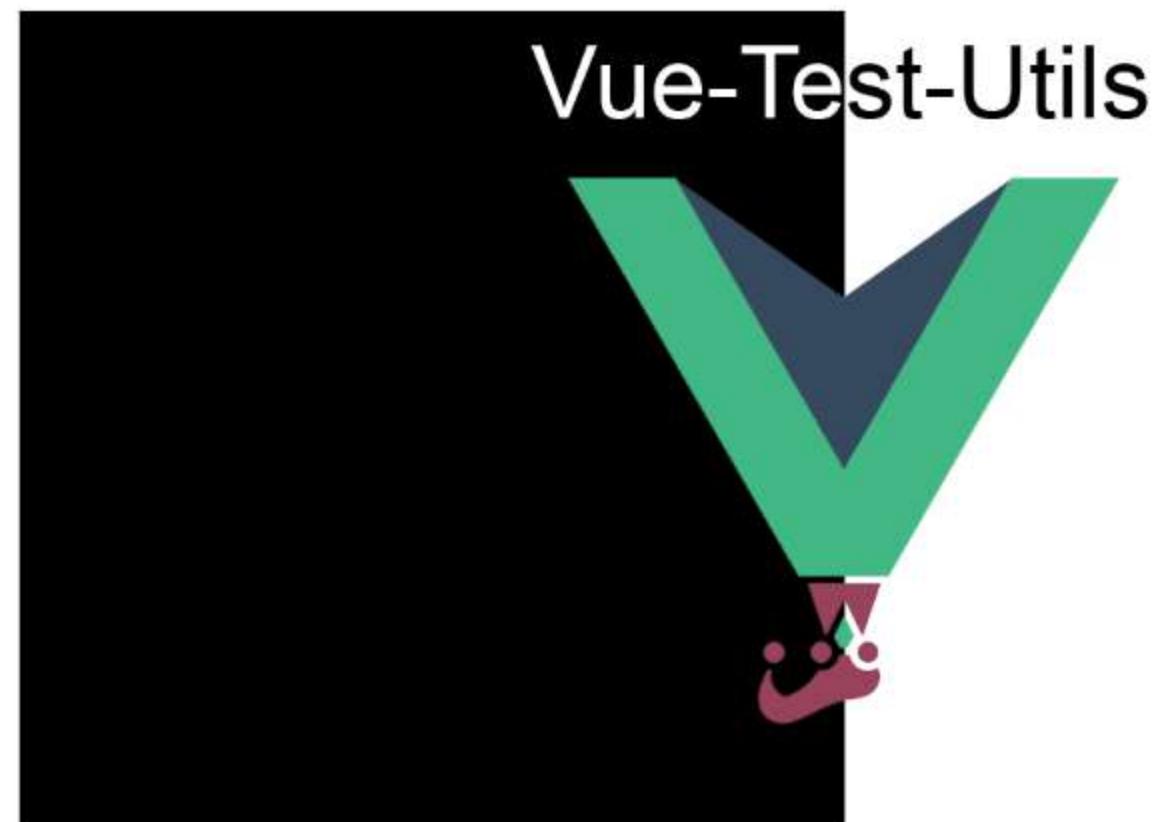
**ESLint de Vue.js** puede detectar errores en el código, e incluso puede corregirlos automáticamente. Puede ejecutarse como una herramienta de línea de comandos y también está integrado en los IDE más comunes. Por lo general, esto consiste en una línea roja ondulada debajo del código infractor, muy parecido al corrector ortográfico.



# Características Agregadas de Vue

Las pruebas o testing son útiles al crear aplicaciones Vue complejas de forma rápida y segura para evitar regresiones, detectando los problemas y solucionarlos antes de su lanzamiento.

Vue CLI tiene opciones integradas para testing unitario con Jest o Mocha que funcionan de manera inmediata. También tenemos el paquete oficial Vue Test Utils, que proporciona una guía más detallada para configuraciones personalizadas.



# Herramientas de desarrollo en Vue

**Vue** cuenta con las siguientes herramientas para desarrolladores:

**Vue Devtools:** Es una extensión de navegador para Chrome y Firefox para el debugging de aplicaciones Vue. Vue Devtools 6 es compatible con Vue 3 y, en este momento, está en versión beta y disponible para Chrome y Firefox.

**Webpack** es un paquete de módulos que toma los componentes Vue y archivos Javascript para combinarlos en paquetes. Esto es diferente de simplemente concatenar y minimizar archivos JavaScript. En pocas palabras, con los paquetes de webpack solo se cargan los módulos requeridos en la página actual. Cuando webpack procesa la aplicación, crea internamente un gráfico de dependencia a partir de uno o más puntos de entrada y luego combina todos los módulos que necesita el proyecto en uno o más paquetes.



**webpack**

El archivo package.json es un repositorio central para la configuración de herramientas que existe en la raíz del proyecto de JavaScript y se usa para administrar las dependencias del proyecto, los scripts, las versiones, el nombre del proyecto, su descripción y mucho más.

```
ackage.json > ...
{
  "name": "sustain_be",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

# Morfología de un componente Vue

A continuación, puedes observar cómo es la morfología o estructura típica de un componente en Vue (en formato .vue):

```
<template>
//Aquí va el contenido HTML

</template>

<script>
//Aquí va el contenido JavaScript
</script>

<style scoped>
//Aquí va el contenido CSS. Por otro lado, se agrega scope para señalar su alcance solo a este componente
</style>
```

# Injectando componente en HTML

Una vez instalada la aplicación Vue CLI, ya se puede usar el comando **vue**. Para crear una nueva aplicación, desplázate al directorio en el que quieras crear tu aplicación y ejecuta el siguiente comando:

```
vue create primer-proyecto
```

Siguiendo los pasos para crear la aplicación con Vue 3, debes desplazarte al directorio de tu nueva aplicación para iniciar el servidor de desarrollo, que por defecto estará en el puerto 8080:

```
npm run serve
```

# Injectando componente en HTML

Crea el componente `hola.vue` dentro de la carpeta `components`. En `script` puedes indicar que se trata de un módulo exportable, añadiéndole un nombre (“`hola-mundo`”).

```
<template>
  <p>Hola Mundo</p>
</template>

<script>
export default {
  name: 'hola-mundo',
}
</script>

<style scoped>
p{
  color:green;
}
</style>
```

# Injectando componente en HTML

Importas el método para crear la App y también el componente al interior del archivo **main.js**. Con **mount**, montas el componente en el selector indicado:

```
import { createApp } from 'vue'  
import App from './components/hola.vue'  
  
createApp(App).mount('#hola')
```

# Injectando componente en HTML

Dicho selector se encuentra en el **index.html** o archivo html que se carga en el navegador.

```
<body>
  <noscript>
    <strong>We're sorry but <%= htmlWebpackPlugin.options.title %>
doesn't work properly without JavaScript enabled. Please enable it to
continue.</strong>
  </noscript>
  <div id="hola"></div>
  <!-- built files will be auto injected -->
</body>
```

# Injectando componente en HTML

Tendrás por resultado el componente injectado, pero ahora visible en tu navegador al implementar get al puerto en el cual el servidor está instalado por defecto (8080).



# Objeto data en Vue

La opción `data()` para un componente es una función. Vue llama a esta función como parte de la creación de una nueva instancia del componente. `Data()` devuelve un objeto, que luego Vue envolverá en su sistema de reactividad y lo almacenará en la instancia del componente como `$data`.

Las propiedades de la instancia se agregan solo cuando la instancia se crea, por lo que puedes usar valores nulos, indefinidos o algún otro marcador de posición para las propiedades cuando el valor deseado aún no está disponible.

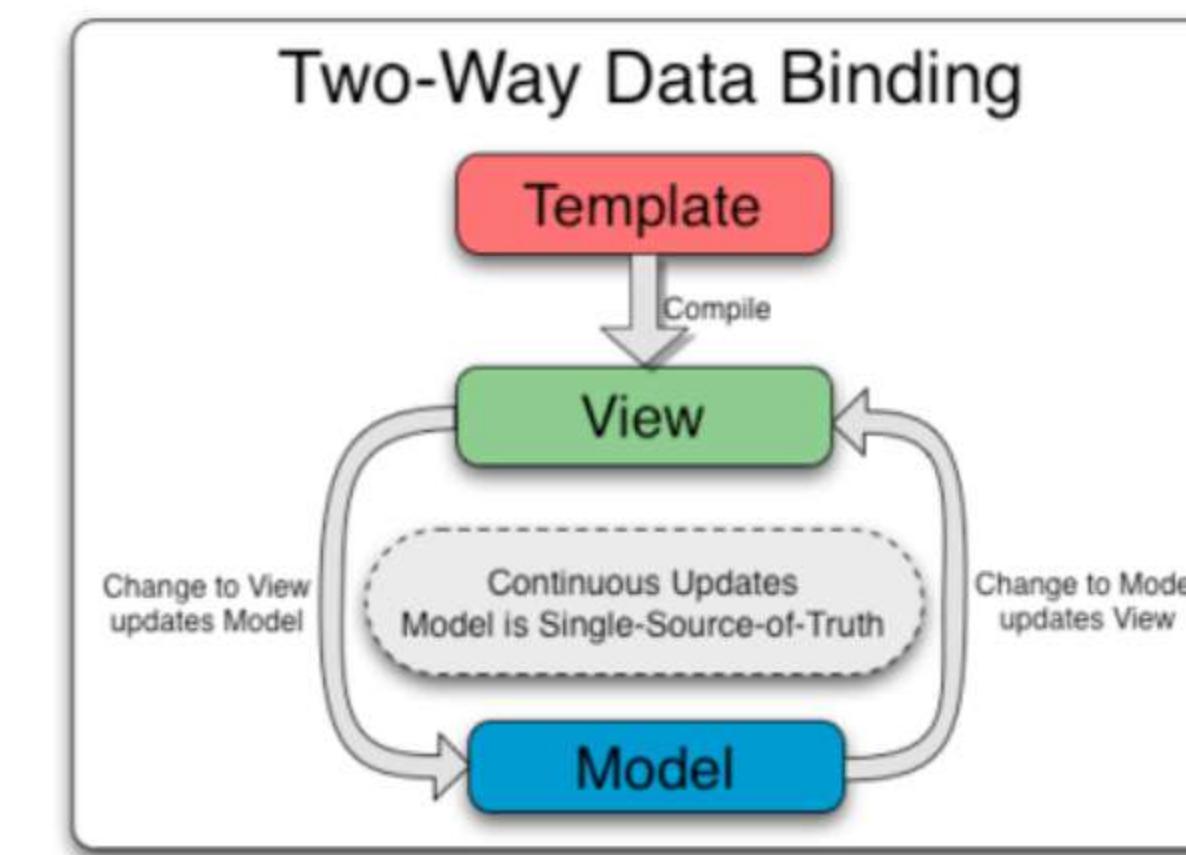
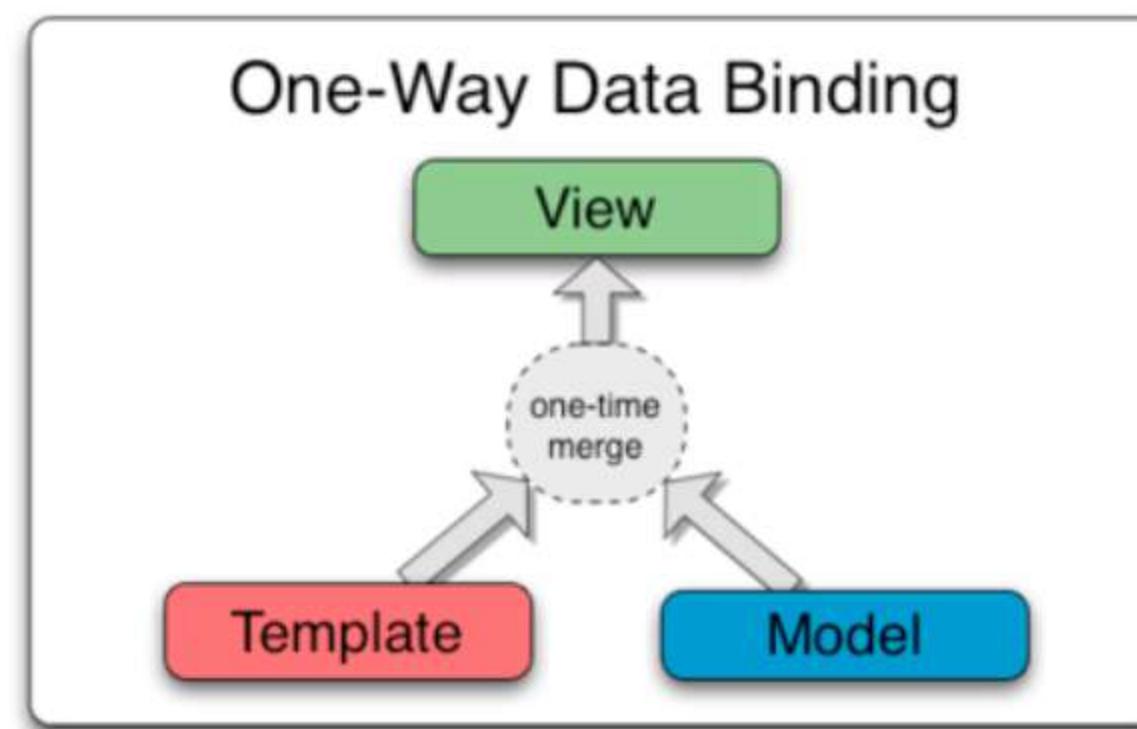
Los componentes de Vue son estáticos, por la cual cada componente debe tener un objeto de datos. De lo contrario, todos los componentes podrían acceder a los mismos datos. Y en caso de cambiar los datos, cambiaría para cada componente.

```
const app = Vue.createApp({  
  data() {  
    return { numero: 4 }  
  }  
})  
  
const vm = app.mount('#app')  
  
console.log(vm.$data.numero) // => 4  
console.log(vm.numero)      // => 4
```

# One-way y Two-way binding

El enlace de datos (data binding) es una de las características más elegantes de Vue.js porque proporciona un enlace de datos reactivo/bidireccional. En Vue.js, no tenemos que escribir muchas líneas para tener un enlace de datos bidireccional, a diferencia de otros frameworks.

El enlace de datos unidireccional significa que la variable solo está vinculada al DOM. Por otro lado, bidireccional significa que la variable también está vinculada desde el DOM. Cuando se cambia DOM, la variable también se cambia.



# One-way data binding

Si queremos vincular cualquier variable, simplemente podemos usar la sintaxis de llaves dobles de Vue.js para vincular cualquier variable de la instancia del componente relativo.

```
<p> {{ Texto }} </p>
```

O, si queremos vincular cualquier variable dentro de un atributo HTML, podemos usar la directiva v-bind.

```
<div v-bind:class="container"></div>
```

Vue.js también proporciona la abreviatura para vincular variables en un atributo HTML. En lugar de escribir v-bind:nombre-atributo, solo podemos usar dos puntos ":" y el nombre del atributo.

```
<div :class="container"></div>
```

# Two-way data binding

Para utilizar el enlace de datos reactivo o two-way, podemos usar la directiva **v-model** en un campo de formulario de entrada. Emitirá internamente un evento y cambiará la variable, la que podemos enlazar en otro lugar de la plantilla usando llaves dobles.

```
<input v-model="texto" placeholder="Escribe algo" />  
  
<p>Estas escribiendo: {{ texto }}</p></td>
```

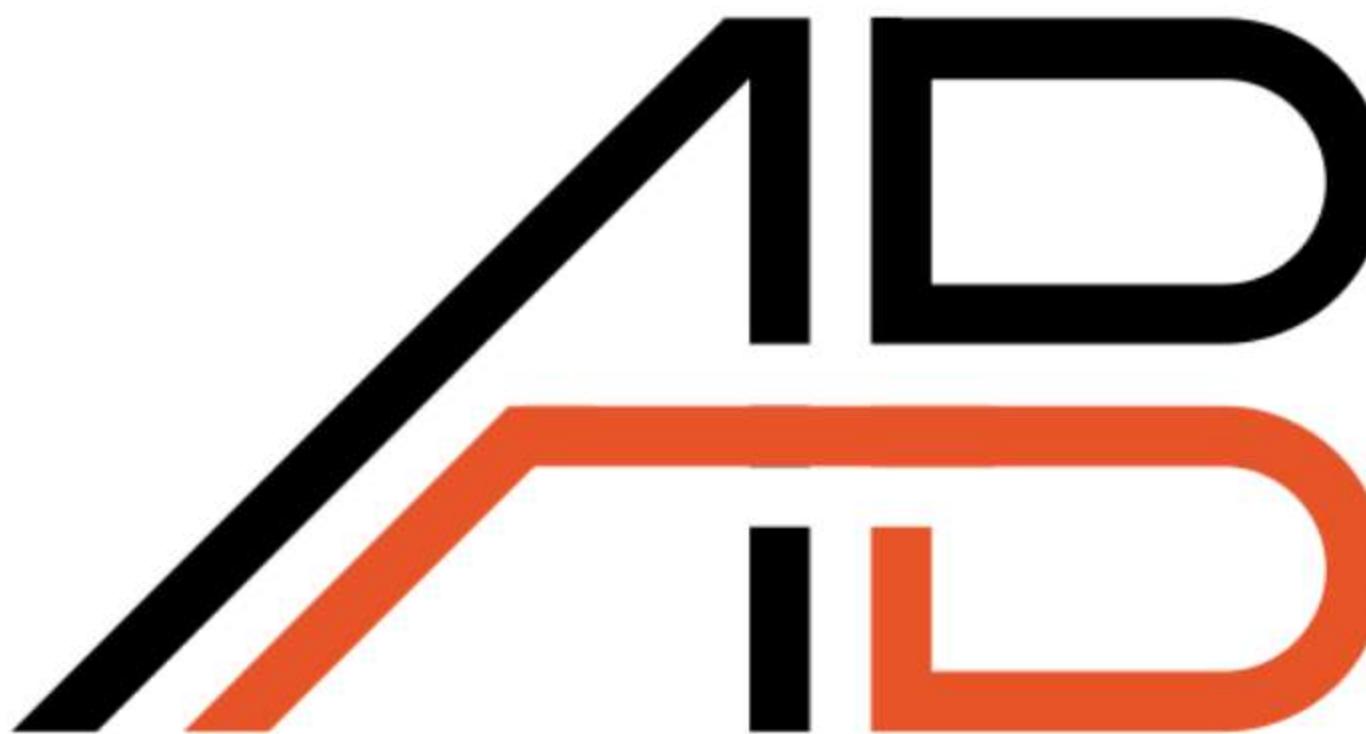
Vue v-model es una directiva que crea un enlace de datos bidireccional entre un valor en la plantilla y un valor en nuestras propiedades de datos. Un caso de uso común para usar v-model es cuando se diseñan **formularios y entradas**. Podemos usarlo para que nuestros elementos de entrada DOM puedan modificar los datos en nuestra instancia de Vue.

```
var V = new Vue({  
  el: '#vue-instancia',  
  data:{  
    nombre:'Diego'  
  }  
})
```

```
<div id="vue-instancia">  
  <div>One way binding<input type="text" :value="nombre"></div>  
  <div>Two way binding : <input type="text"  
  v-model="nombre"></div>  
</div>
```

**Salida.** Notar que si se modifica desde el DOM el segundo input (two-way), se cambia automáticamente el primero. Por el contrario, no se modifica el segundo input al cambiar el primero desde el DOM.

One Way binding   
Two way binding :



Bootcamp  
**Academy**  
*sustantiva*