Clase 15

VUE.js Parte 1

Fuente: https://es.vuejs.org/

VUE.js

¿Qué es Vue.JS?

Vue.JS es un framework JavaScript progresivo de código abierto que se utiliza para desarrollar interfaces web interactivas. Es uno de los marcos famosos que se utilizan para simplificar el desarrollo web. Vue.JS se centra en la capa de vista. Se puede integrar fácilmente en grandes proyectos para el desarrollo front-end sin ningún problema.

La instalación de Vue.JS es muy fácil. Cualquier desarrollador puede comprender y crear interfaces web interactivas fácilmente en cuestión de tiempo. Vue.JS fue creado por Evan You, un ex empleado de Google. La primera versión de Vue.JS se lanzó en febrero de 2014.

VUE.js

Que es Vue?

Es un framework de JavaScript.

Vue (pronunciado /vjuː/, como view) es un **framework** progresivo para construir interfaces de usuario. A diferencia de otros frameworks, 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.

VUE.js

La forma más fácil de comenzar a usar Vue.js es crear un archivo index.html e incluir Vue con:

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

La página de instalación https://es.vuejs.org/v2/guide/installation.html proporciona más opciones de instalación de Vue. Nota: No recomendamos que los principiantes comiencen con vue-cli.

DOM Virtual

Vue.js utiliza DOM virtual, que también es utilizado por otros frameworks como React, Ember, etc.

Los cambios no se realizan en el DOM, sino que se crea una réplica del DOM que está presente en forma de estructuras de datos JavaScript. Siempre que se deben realizar cambios, se realizan en las estructuras de datos de JavaScript y esta última se compara con la estructura de datos original.

Luego, los cambios finales se actualizan al DOM real, que el usuario verá cambiar. Esto es bueno en términos de optimización, es menos costoso y los cambios se pueden realizar a un ritmo más rápido.

La instancia Vue

Creando una instancia de Vue

Cada aplicación de Vue se comienza creando una nueva **Instancia de Vue** con la función `Vue`:

```js
var vm = new Vue({
  // opciones
})
```

Aunque no estrictamente asociado con el patrón MVVM, el diseño de Vue fue en parte inspirado por él.
Como una convención, solemos usar la variable `vm` (abreviación de ViewModel) para hacer referencia a nuestra instancia de Vue.

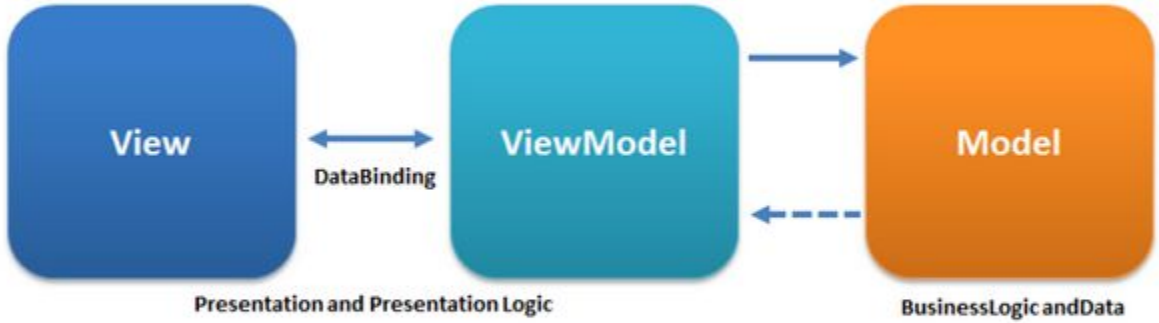
- https://es.vuejs.org/v2/guide/instance.html

Patrón MVVM

Model–view–viewmodel

From Wikipedia, the free encyclopedia

Model–view–viewmodel (**MVVM**) is a software architectural pattern that facilitates the separation of the development of the graphical user interface (the *view*) – be it via a markup language or GUI code – from the development of the business logic or back-end logic (the *model*) so that the view is not dependent on any specific model platform. The *view model* of MVVM is a value converter,[1] meaning the view model is responsible for exposing (converting) the data objects from the model in such a way that objects are easily managed and presented. In this respect, the view model is more model than view, and handles most if not all of the view's display logic.[1] The view model may implement a mediator pattern, organizing access to the back-end logic around the set of use cases supported by the view.



- https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel

Modelo de enlazado de datos

- Modelo que permite enlazar (comunicar) el documento HTML con un JS.

Data Binding

La función de data binding ayuda a manipular o asignar valores a atributos HTML, cambiar el estilo, asignar clases con la ayuda de la directiva de enlace llamada **v-bind** disponible en Vue.JS.

Renderización Declarativa (Interpolación)

Sintaxis de Template
Note que VUE utiliza las las **llaves dobles** para encerrar el dato que quiere mostrar {{ }}, similar a Template String de JS, que lo hace con ${ }

{{ message }}

Los **datos** y el **DOM** ahora **están vinculados**, y todo ahora es reactivo. Si cambio el valor de `app.message` a un valor diferente. Debería ver que el ejemplo se ha renderizado con el nuevo valor que acaba de ingresar.

Renderización Declarativa

En el archivo **HTML**

```
<body>
    <div id="app">
          {{ message }}
     </div>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
    <script src="index.js"></script>
</body>
```

Y en **JavaScript**

```
var app = new Vue({
  el: '#app',
  data: {
    message: 'Bienvenido a  Codo a Codo !!!'
  }
})
```

En el navegador vería: Bienvenido a Codo a Codo !!!

En el archivo index.js se crea un nuevo objeto Vue, con new Vue() y dentro de los paréntesis, un objeto .:

La propiedad el:'#app' vincula el nuevo objeto Vue al elemento HTML con id = "app" .

Directivas

Vue utiliza **directivas** para aplicar un comportamiento especial al DOM.

Las directivas tienen el prefijo `v-` para indicar que son atributos especiales proporcionados por Vue.

v-text: https://es.vuejs.org/v2/api/#v-text
v-bind: https://es.vuejs.org/v2/api/#v-bind
v-if, v-else, v-elseif: https://es.vuejs.org/v2/api/#v-if
v-for: https://es.vuejs.org/v2/api/#v-for
v-show: https://es.vuejs.org/v2/api/#v-show
v-model: https://es.vuejs.org/v2/api/#v-model

Más directivas: https://es.vuejs.org/v2/api/#Directivas

Atributos v-bind

Los *mustaches* (llaves doble) no se pueden utilizar dentro de los atributos HTML.

En su lugar, use una directiva v-bind:

Ejemplos:

```
<div v-bind:id="dynamicId"></div>
```

```
<img class="img-fluid" v-bind:src="img/"{{playerDetail.pic}} alt="">
```

Renderización Condicional

En Vue, usamos la directiva `v-if` para mostrar una etiqueta o no mostrarla

```
<h1 v-if="ok">Sí</h1>
```

También es posible agregar un “bloque *else*“ con `v-else`:

```
<h1 v-if="ok">Sí</h1>
<h1 v-else>No</h1>
```

El `v-else-if`, como su nombre lo indica, sirve como “bloque *else if*“ para `v-if`.

```
<div v-if="type === 'A'">
 A
</div>
<div v-else-if="type === 'B'">
 B
</div>
<div v-else>
 Si no es A, B o C
</div>
```

Renderización de una lista

Vamos Buenos Aires

Podemos usar la directiva `v-for` para representar una lista de elementos basada en una Array. La directiva `v-for` requiere una sintaxis especial en forma de `item in items`, donde los `items` son el array de datos de origen y el `item` es un alias para el elemento del Array que se está iterando:

```
<ul id="example-1">
 <li v-for="fruta in frutas">
    {{ fruta.nombre }}
 </li>
</ul>
```

```
var example1 = new Vue({
 el: '#example-1',
 data: {
   frutas: [
     {nombre:'naranja'},
     {nombre:'banana' }
   ]
 }
})
```

Enlace bidireccional

La directiva `v-model` vincula el valor de los elementos HTML a los datos de la aplicación.

Esto se denomina enlace bidireccional:

```
<div id="app">

  <p>{{ message }}</p>

  <p><input v-model="message"></p>          Aquí en el input, lo que ingrese el usuario, modifica el message y el

</div>                                      message modifica el 1er párrafo

<script>

var myObject = new Vue({

    el: '#app',

    data: {message: 'Hello Vue!'}

})

</script>
```

Manejo de eventos

Escuchar eventos

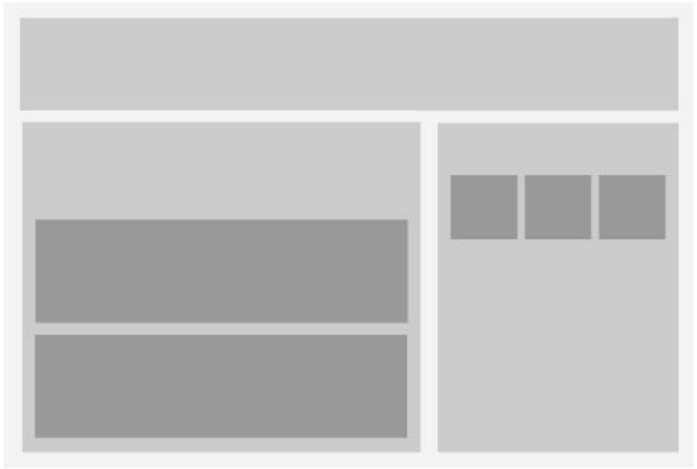
Podemos usar la directiva `v-on` para escuchar eventos DOM y ejecutar algunas instrucciones de JavaScript cuando se activan.

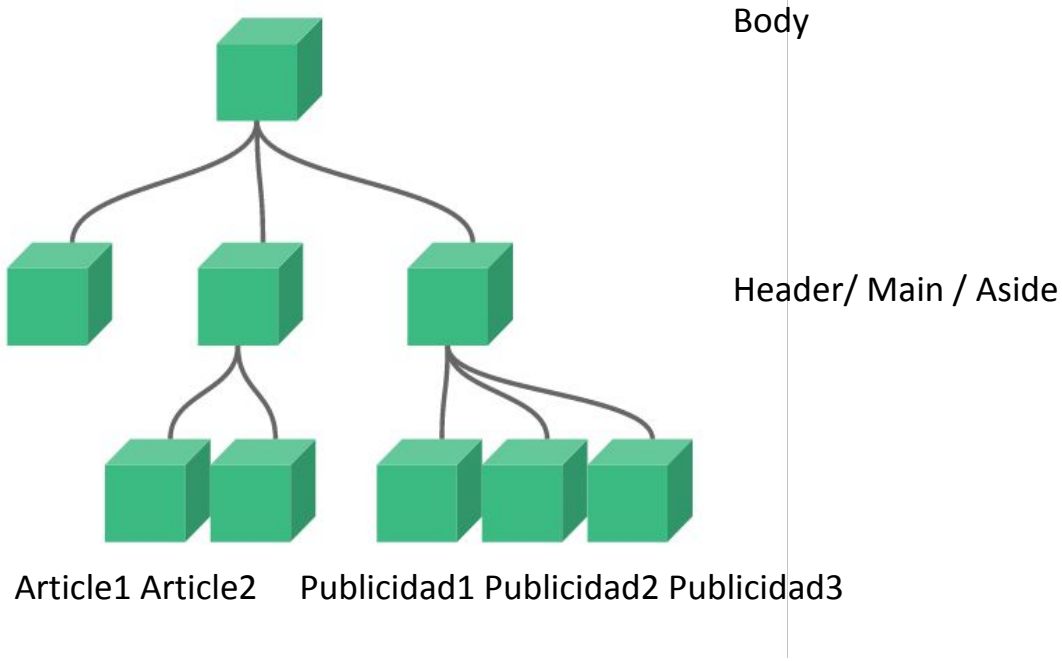
```
<div id="example-1">
 <button v-on:click="counter += 1">Add 1</button>
 <p>Se ha hecho clic en el botón de arriba {{ counter }} veces.</p>
</div>
var example1 = new Vue({
 el: '#example-1',
 data: {
   counter: 0
 }
})
```

Concepto Básico de Componente

Organización de Componentes

Es común que una aplicación se organice en un árbol de componentes anidados:





Concepto Básico de Componente

Vue te permite tomar una pagina web y dividirla en componentes cada uno con su HTML, CSS y JS necesario para generar esa parte de la página

Guía Vue.js

- https://es.vuejs.org/v2/guide/index.html#

Ejemplos Vue

- Ejemplos Vue: https://vuejsexamples.com/