

A promotional image for Vue.js featuring three Harry Potter characters: Cedric Diggory on the left, Hermione Granger in the center, and Harry Potter on the right. They are all holding wands and looking forward. Overlaid on the image is the Vue.js logo, which consists of a large green 'V' with a blue chevron inside it. The text 'VUE.JS' is written in white with a red outline above the logo. Below the logo, the text 'EL FRAMEWORK JAVASCRIPT PARA MUGGLES' is written in white.

VUE.JS

**EL FRAMEWORK
JAVASCRIPT
PARA
MUGGLES**

DESARROLLO WEB 101

- HTML *Estructura y Contenidos*
- CSS *Presentación*
- Javascript **Interacción**

PROGRAMAR JAVASCRIPT SIN LIBRERÍAS NI FRAMEWORKS



SERIOUSLY?

EVEN IM NOT THAT CRAZY

JAVASCRIPT CON JQUERY

- Simple
- Potente
- Ligero
- Está muy bien ... para cosas sencillas

EJEMPLO

Pleno del Parlamento 4/4/2017



Hora de inicio: 12:06:41

Hora de finalización: 14:45:44

Comentarios

La señora presidenta	12:07:17
La señora presidenta	12:08:07
La señora presidenta	12:14:05
La señora presidenta	12:17:05
1.1.- 9L/PO/P-1273.- centros de enseñanza públicos que no prestan servicio de comedor al alumnado, dirigida al Sr. Presidente del Gobierno.	12:17:14
La señora presidenta	12:17:31
Noemí Santana Perera	12:17:43
La señora presidenta	12:17:56
Fernando Clavijo Batlle	12:17:59

COMO DEBE FUNCIONAR

- A la izquierda hay un listado de comentarios
- A medida que el vídeo avance, se debe resaltar el comentario pertinente
- Si pulsamos en el comentario, reposiciona el vídeo

Pleno del Parlamento 4/4/2017



Hora de inicio: 12:06:41
Hora de finalización: 14:45:44



Comentarios

La señora presidenta	12:07:57
La señora presidenta	12:08:07
La señora presidenta	12:14:05
La señora presidenta	12:17:05
1.1.- 9L/POI-1273.- centros de enseñanza públicos que no prestan servicio de comedor al alumnado, dirigida al Sr. Presidente del Gobierno.	12:17:14
La señora presidenta	12:17:31
Noemí Santana Perera	12:17:43
La señora presidenta	12:17:56
Fernando Clavijo Batlle	12:17:59

HASTA AQUÍ, BIEN

- Es factible hacerlo con jQuery
- De hecho, estaba así
- Pero llegan nuevas especificaciones
 - Contador de tiempo real (Hora de inicio + timestamp)
 - Controles play/pause adicionales
 - Al hacer pause, queremos cambiar la url

JQUERY EMPIEZA A QUEDARSE PEQUEÑO

- Código confuso, frágil y difícil de modificar
- Todas las funciones tienen que saber de todo
- Por ejemplo, el botón de pause debe:
 - Poner el vídeo en pausa
 - Cambiar la URL
 - Cambiar su propia imagen (de  a )
 - Parar el contador de tiempo real



ENTORNOS REACTIVOS

Para resolver estos problemas se desarrollan frameworks reactivos, que se basan en:

- El patron MVVM (Modelo Vista VistaModelo)
- **data-binding y two way data-binding**

EL PATRON MVVM (MODELO VISTA VISTAMODELO)

Es una versión del patron MVC. En este caso tenemos un modelo y una vista (o vistas). Pero la interacción entre ellos es **automática**, no se implementa con un controlador sino que **se declara**.

DATA-BINDING

Al cambiar un dato en el modelo, este cambio se refleja o propaga de inmediato a la vista o vistas que dependan de él

HOLA, MUNDO EN VUE.JS

Lab 1 - Data binding

```
<div id="app">  
  <p>hola, {{ nombre }}.</p>  
</div>
```

```
<script>  
data = {  
  'nombre': 'Zaratustra'  
};
```

```
new Vue({  
  el: '#app',  
  data: data  
});  
</script>
```

OBSÉRVESE QUE...

- Al iniciar Vue especificamos el elemento en el que puede actuar, en este caso, `#app`. Vue no interactuará ni afectará nada que no esté dentro de ese elemento
- Usamos los delimitador `{{`, `}}` para realizar un enlace o *binding* entre la página y nuestro modelo.
- La página es **reactiva**

VUE TAGS: V-HTML

Podemos vincular con el contenido, respetando las marcas html con la directiva `v-html`

Lab 2: La directiva v-html

```
<div id="app">  
  <div v-html="msg"> </div>  
</div>
```

```
<script>  
data = {  
  'msg': '<p class="warning">' +  
    'Esto es un <b>aviso</b>' +  
    '$$</p>'  
};
```

```
var vm = new Vue({  
  el: '#app',  
  data: data  
});  
</script>$
```

OBSÉRVESE QUE...

- Podemos almacenar el resultado de la llamada a Vue en una variable (**vm** en este caso)
- Todas las directivas de vue empieza con **V -**

VINCULAR UN ATRIBUTO HTML

Podemos vincular un dato de nuestro modelo con un atributo; para ello usamos la directiva **v-bind**

Lab 3: La directiva v-bind

```
<style>
.redbox {
  border: 2px solid red;
  background-color: maroon; color:white
}
.bluebox {
  border: 2px solid blue;
  background-color: DarkBlue color:white
}
</style>

<div id="app">
  <div v-bind:align="position" v-bind:class="klass">
    Hola, mundo
  </div>
</div>

<script>
var data = {
  position: 'left',
  klass: 'redbox'
};

var vm = new Vue({
  el: '#app',
  data: data
});
</script>
```

OBSÉRVESE QUE...

- Lo más habitual es para cambiar la clase, aunque se puede modificar cualquier atributo
- Se puede abreviar de la siguiente manera:

`v-bind:href` \equiv `:href`

TWO-WAY DATA BINDING

- Podemos realizar una vinculación de doble sentido
- Los cambios en el modelo se reflejarán en la vista, y los cambios en la vista se reproducirán en el modelo
- para ello usamos la directiva **v-model**

Lab 4: La directiva v-model

```
<div id="app">
  <form>
    <input type="text" name="txt" v-model="nombre">
    <p>Hola, {{ nombre }}</p>
  </form>
</div>

<script>
var data = {
  nombre: 'Federico Mendelson Bartholdy'
};

var vm = new Vue({
  el: '#app',
  data: data
});
</script>
```


LAS DIRECTIVAS V-IF Y V-SHOW

- Las dos permite controlar si un elemento aparece o no en la página
- La diferencia es que con **v-if** el elemento realmente estará o no en el árbol DOM, dependiendo del predicado, pero con **v-show** el elementos siempre estará, y será visible o no en función del predicado

Lab 5: La directiva v-if

```
<div id="app">  
  <p v-if="flag">Cara</p>  
  <p v-if="!flag">Cruz</p>  
</div>
```

```
<script>  
var data = {  
  flag: true  
};  
  
var vm = new Vue({  
  el: '#app',  
  data: data  
});  
</script>
```

OBSÉRVESE QUE...

- Dentro de la condición podemos incluir expresiones en javascript (solo una línea)
- Si queremos que la directiva afecte a varias etiquetas, podemos agruparlas `<template v-if="...">` y `</template>`, estas desaparecen en el resultado final
- Existe la directiva `v-else`

LA DIRECTIVA V-FOR

- Podemos usar la directiva `v-for` para recorrer una lista de datos
- La sintaxis usa la forma `v-for="item in items"`

Lab 6: iterar con la directiva v-for

```
<div id="app">
  <ul>
    <li v-for="t in tareas">
      {{ t.texto }}:{{ t.prioridad }}
    </li>
  </ul>
</div>
```

```
<script>
var data = {
  tareas: [
    {texto: 'Aprender Vue.js', prioridad: 7},
    {texto: 'Dominar el mundo', prioridad: 9}
  ]
};
```

```
var vm = new Vue({
  el: '#app',
  data: data
});
</script>
```

OBSERVACIONES

- Podemos obtener el índice con la sintaxis `(item, index) in items`
- También se puede iterar por las propiedades de un objeto
- Ojo con los métodos que no modifican un array, sino que devuelven uno nuevo
- Cuidado: Cambios por el índice no son detectados por Vue, usar siempre `shift`, `unshift`, `pop`, etc...

MÉTODOS CON V-ON

- Podemos vincular acciones con métodos usando la directiva **v-on:evento**
- Tenemos que definir nuestros métodos en la entrada methods de Vue.

Lab 7: Métodos y la directiva v-on

```
<script>
var data = {
  nuevo_texto: '',
  nueva_prioridad: '',
  tareas: [
    {texto: 'Aprender Vue.js', prioridad: 7},
    {texto: 'Dominar el mundo', prioridad: 9}
  ]
};
...
```



```
...  
var vm = new Vue({  
  el: '#app',  
  data: data,  
  methods: {  
    addTarea: function() {  
      this.tareas.push({  
        texto: this.nuevo_texto,  
        prioridad: this.nueva_prioridad,  
      });  
      this.nuevo_texto = '';  
      this.nueva_prioridad = '';  
    }  
  }  
});  
</script>
```

```
<div id="app">
  <ul>
    <li v-for="t in tareas">
      {{ t.texto }} - {{ t.prioridad }}
    </li>
  </ul>

  <p>
    Tarea:
    <input type="text" name="t" v-model="nuevo_texto">
    Prioridad:
    <input type="text" name="p" v-model="nueva_prioridad">
  </p>
  <button v-on:click="addTarea">+</button>
</div>
```

OBSERVACIONES

- El contenido de v-on es javascript, pero lo normal es que sea una llamada a un método
- Existen modificadores del evento, en la forma `v-on.event.modifier`, por ejemplo `v-on:keyup.enter` solo se activa si la tecla pulsada es enter
- Podemos abreviar `v.bind:evento=...` como `@evento=...`

FILTROS

- Vue.js permite definir filtros para presentación y formateo de resultados
- Se usan igual que los filtros de Django, después del caracter pipe "|"
- Podemos encadenar varios filtros
- Pueden tener argumentos adicionales (Y más de uno, no como Django)
- Se declaran en la entrada **filters**

COMPONENTES

- Podemos hacer componentes, que son como instancias pequeñas de vue con sus propias plantillas, métodos, modelos, etc...
- Muy interesantes y muy potentes
- Los componentes pueden usar otros componentes, lo que permite estructurar la aplicación a modo de bloques

COSAS QUE NO PODREMOS VER HOY

Pero resultan interesantes

Sobre todo si tienes javascript en el servidor

- Componentes en un solo fichero .vue
- Includes dinámicos de código javascript
- Vuex para gestionar el estado de la aplicación
- Plugins
- Testing
- One-page apps
- *A lot of stuff...*

YO VINE AQUÍ A HABLAR DE MI LIBRO (1/2)



Gobernación, Justicia, Igualdad y Diversidad

1 de Junio de 2017 10:05

Hora de inicio: 10:05:14

Hora de finalización: 11:53:50

Vídeos anteriores

Compartir este momento 10:20:34

Comentarios ▲

10:05:24 La señora presidenta de la comisión

10:05:47 1.1.- 9L/C-0718.- -Consejero de Presidencia, Justicia e Igualdad-, sobre medidas para mejorar la situación de las mujeres víctimas de violencias machistas.

10:05:52 María Del Río Sánchez

10:10:57 La señora presidenta de la comisión

10:11:02 José Miguel Barragán Cabrera

10:20:20 La señora presidenta de la comisión

10:20:24 Melodie Mendoza Rodríguez

10:23:02 La señora presidenta de la comisión

YO VINE AQUÍ A HABLAR DE MI LIBRO (2/2)



Gobernación, Justicia, Igualdad y Diversidad

1 de Junio de 2017 10:05

Hora de inicio: 10:05:14

Hora de finalización: 11:53:50

Vídeos anteriores

Compartir este momento 11:28:47



11:15:34 La señora presidenta de la comisión

11:16:49 3.1.- 9L/PNL-0412.- la ubicación del nuevo edificio judicial de Santa Cruz de Tenerife.

11:17:00 Gustavo Matos Expósito

11:28:38 La señora presidenta de la comisión

11:28:46 Melodie Mendoza Rodríguez

11:31:53 La señora presidenta de la comisión

11:31:57 María Esther González González

11:36:12 La señora presidenta de la comisión

11:36:17 Asunción Delgado Luzardo

11:42:53 La señora presidenta de la comisión

MEJORAS OBTENIDAS

- De jquery a Vue.js, con la misma funcionalidad inicial, produjo un código \cong 30% más pequeño
- Con la nueva funcionalidad, \cong 90% código que en la version anterior con jQuery (Awesome!)
- Mucho más fácil de comprender, reparar y/o extender
- Puedes seguir usando jQuery, no son incompatibles

TRUCO 1: USAR EN DJANGO

- Las plantillas de Django usan el mismo sistema que Vue; los delimitadores {{ y }}
- Para usar Vue.js desde Django con comodidad, hay un parámetro en la configuración, **delimiters** que permite usar como delimitadores los que tú especifiques, yo cambio {{ }} por [[]]

delimiters: ['[[', ']]'],

TRUCO 2: HAZ QUE DESAPAREZCAN LOS BIGOTES {{ }}

- En la carga de la página, se pueden ver, las marcas de plantilla hasta que Vue haya terminado de cargarse e inicializarse. Feo.
- Para resolverlo, Vue aplica inicialmente la directiva `v-cloak` a todo lo que tenga bajo su control, y cuando ya esté completamente operativo la quita.
- En resumen, esto en tu hoja de estilos debería resolver el problema:

```
[v-cloak] { display:none; }
```

FRAMEWORKS SIMILARES A VUE.JS

- Angular.js
- React
- Knockout.js
- Ember, Polymer, Backbone... *You named it*

¿POR QUÉ VUE.JS? (1/2)

WARNING: ABSOLUTELY SUBJECTIVE OPINIONS AHEAD!

- Pequeño tamaño
- Los modelos son simples objetos javascript, no hay que heredar, registrar, llamarlos de determinada manera, etc...
- Se puede empezar a usar simplemente añadiendo la librería vue.js (Luego si quieres te puedes liar la manta a la cabeza, pero no es obligatorio instalar 20 dependencias para el hola mundo)

¿POR QUÉ VUE.JS? (2/2)

WARNING: ABSOLUTELY SUBJECTIVE OPINIONS AHEAD!

- No es de facebook
- Para componentes usa los tres lenguajes que ya conocemos: Html, css y javascript para contenidos, presentacion y lógica
- Si quieres, puedes entender lo que pasa internamente, no hay "magia"

Por eso esta presentación se llama...

VUE.JS - EL FRAMEWORK JAVASCRIPT PARA MUGGLES



Juan Ignacio Rodriguez de León

Licenciado en Frameworks Mágicos por la escuela de Magia de Hogwarts

euribates at gmail.com · @jjleon en twitter

Esta presentación se puede descargar de <https://github.com/euribates/vuelab/>