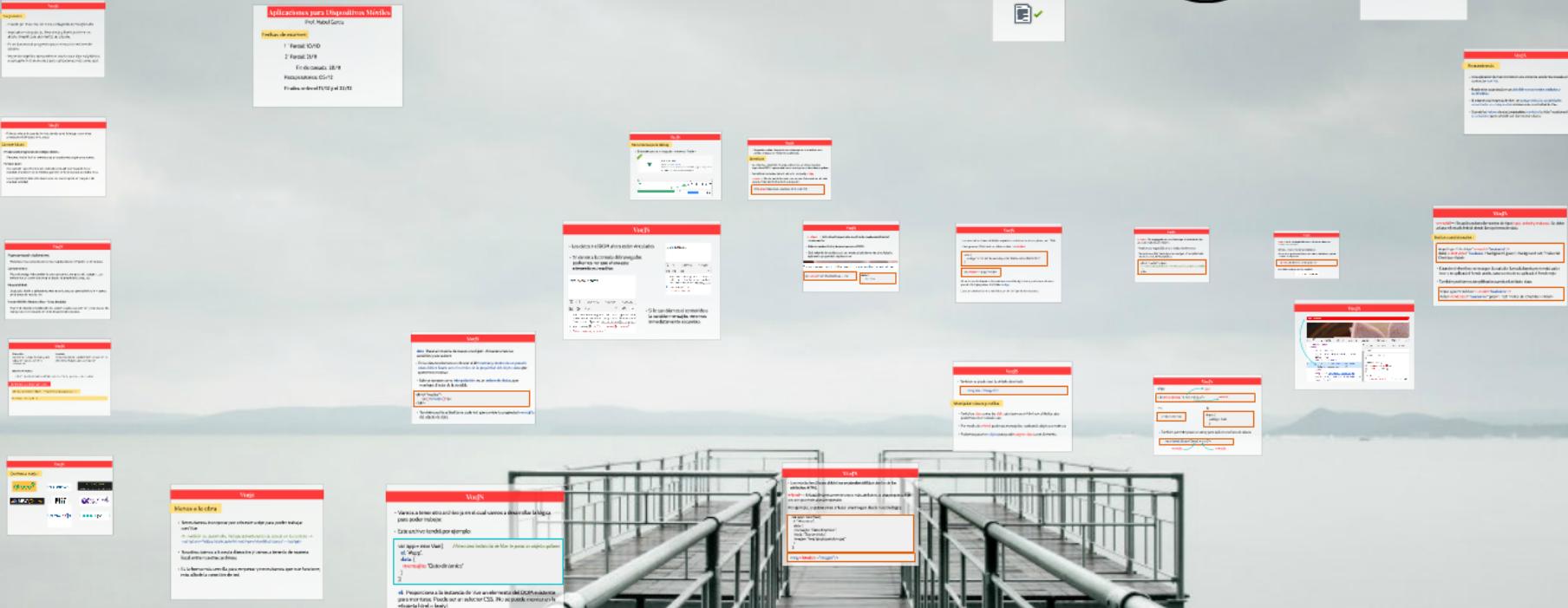


ADM



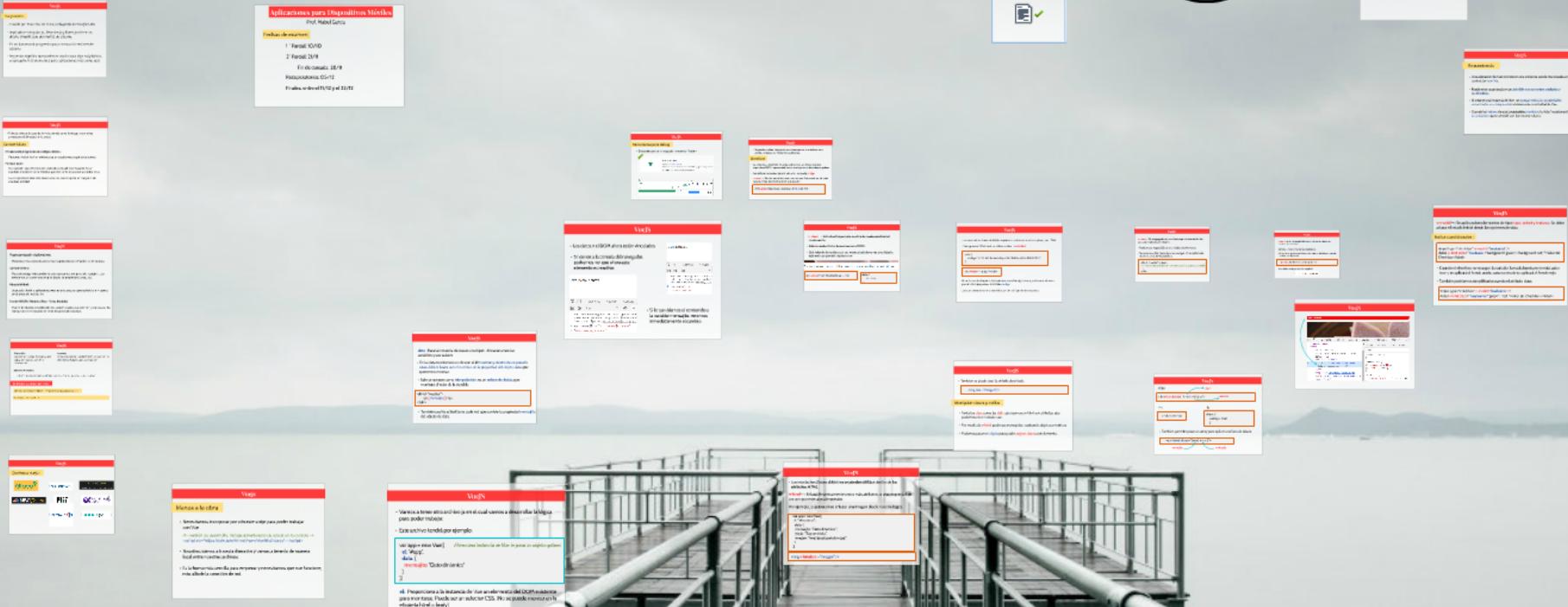
Dudas?



ADM



Dudas'



Aplicaciones para Dispositivos Móviles

Prof. Mabel García

Fechas de examen:

1º Parcial: 10/10

2º Parcial: 21/11

Fin de cursada: 28/11

Recuperatorios: 05/12

Finales: entre el 11/12 y el 22/12

Surgimiento

- Creado por Evan You en 2014, trabajando en Google Labs
- inspirado en Angular.js , Reactive.js y Rivets.js ofrece un diseño simplificado de interfaz de usuario.
- Es un framework progresivo para construir interfaces de usuario
- Pogresivo significa que podemos usarlo para algo muy básico, o agregarle más elementos para aplicaciones más complejas.

VueJS

- Trabaja solo en la capa de la vista, siendo sencillo integrar con otros proyectos o bibliotecas existentes.

Características:

- **Framework progresivo de código abierto :**

Podemos incluir las herramientas que necesitemos según precisemos.

- **Virtual Dom:**

Es cualquier tipo de representación de un DOM real. Trata de evitar cambios innecesarios en el DOM, que son caros en cuanto a rendimiento.

La re-representación solo ocurre una vez que se aplica un conjunto de cambios al DOM.

VueJS

Representación declarativa:

Podemos crear variables en nuestras aplicaciones e imprimirlas en la vista.

Componentes :

Pieza de código independiente que representa una parte de la página. Los componentes tienen sus propios datos, su propio JavaScript, etc.

Adaptabilidad:

Se puede añadir a aplicaciones web ya existentes y aprovecharlo sin mayores problemas de instalación.

Patrón MVVM (Modelo Vista - Vista Modelo)

Patrón de diseño o modelo de abstracción usados para definir y estructurar los componentes necesarios en el desarrollo de software.

VueJS

El modelo

Representa la capa de datos y/o la lógica de negocio, contiene información.

La vista

Debe representar la información a través de los elementos visuales que la componen.

Sistema Reactivo:

- Mantiene interacción constante con su entorno. por medio de eventos

Referencias a tener presente

URL Documentación Oficial : <https://es.vuejs.org/v2/guide/>

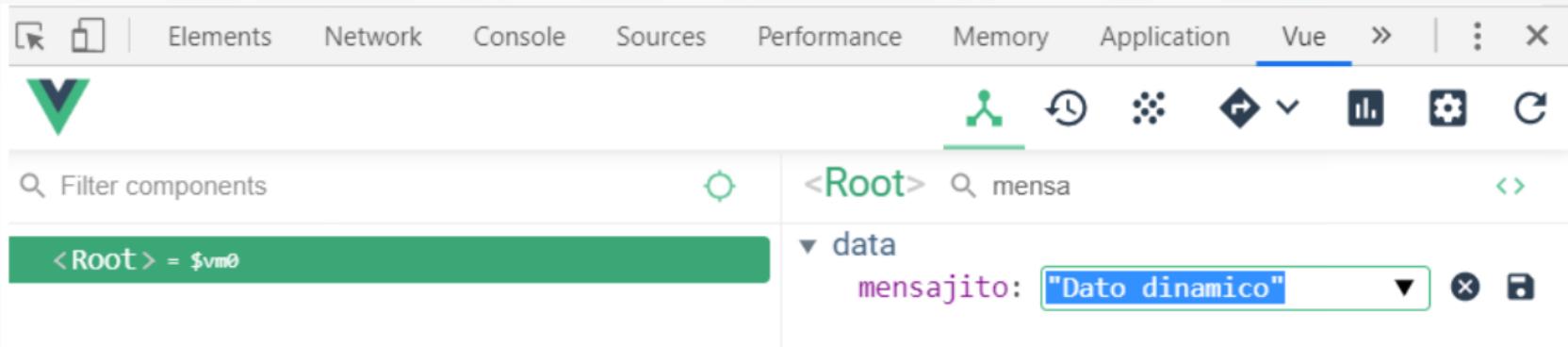
Versión a usar : v2.6.11



VueJS

Herramienta para debug

- Extensión para el navegador (chrome y Firefox)



VueJS

Quién usa vuejs?



euronews.

PLAYCODE

Code Sandbox. Online Code Editor.



Carrefour

LARAC-➔STS

Manos a la obra

- Necesitamos incorporar por cdn este script para poder trabajar con Vue

```
<!-- versión de desarrollo, incluye advertencias de ayuda en la consola -->
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

- Nosotros vamos a ir a esta dirección y vamos a tenerlo de manera local entre nuestros archivos.
- Es la forma más sencilla para empezar y necesitamos que vue funcione, más allá de la conexión de red.

VueJS

- Vamos a tener otro archivo js en el cual vamos a desarrollar la lógica para poder trabajar.
- Este archivo tendrá por ejemplo:

```
var app = new Vue({      //crea una instancia de Vue, le pasa un objeto options
  el: '#app',
  data: {
    mensajito: 'Dato dinámico'
  }
})
```

el: Proporciona a la instancia de Vue un elemento del DOM existente para montarse. Puede ser un selector CSS. (No se puede montar en la etiqueta html o body)

VueJS

data : Para la instancia de vue es un object. Almacenamos las variables y sus valores

- En la vista tendremos un div con el id **mostrar** y dentro de un párrafo unas dobles llaves con el nombre de la propiedad del objeto data que queremos mostrar.
- Esto se conoce como **interpolación**, es un **enlace de datos**, que mostrará el valor de la variable.

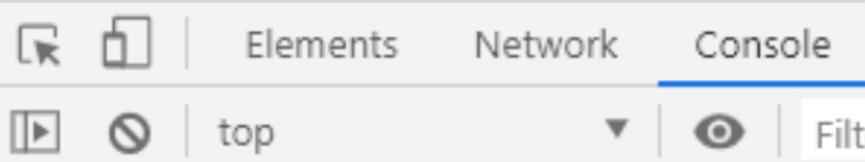
```
<div id="mostrar">  
  <p>{{mensajito}}</p>  
</div>
```

- También podría actualizarse cada vez que cambie la propiedad **mensajito** del objeto de data.

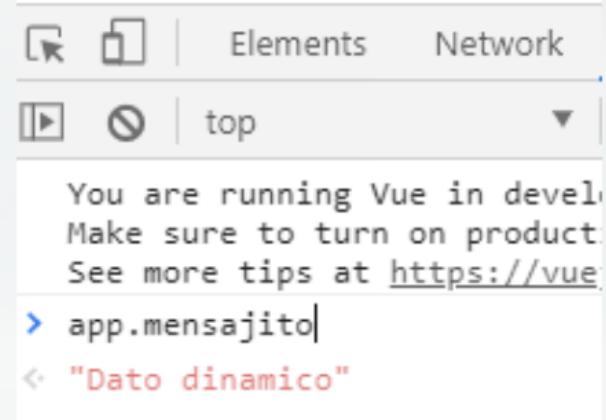
VueJS

- Los datos y el DOM ahora están vinculados
- Si vamos a la consola del navegador, podremos ver que ahora este elemento es reactivo.

Otro mensaje nuevo



Dato dinamico



- Si le cambiamos el contenido a la variable mensajito, veremos inmediatamente el cambio.

- Se pueden realizar interpolaciones únicas que no se actualizan en el cambio de datos, por medio de una directiva.

Directivas

- Las directivas ofrecen distintos tipos de comportamientos reactivos especiales al DOM representado sobre las etiquetas en las cuales se aplican.
- Son atributos especiales para un elemento con prefijo **v-algo**
- **v-once** --> Renderizará el elemento solo una vez. Esto puede ser utilizado para optimizar la performance de actualización.

```
<h1 v-once>Esto no va a cambiar: {{ titulo }}</h1>
```

VueJS

v-show --> Esta directiva permite mostrar de modo condicional un elemento.

- Este se renderizará y lo veremos en el DOM.
- Si el valor de la variable es true, mostrará el elemento, si es false le aplicará la propiedad display:none

Este texto se mostrará en tanto la variable ver tenga valor true, caso contrario se le aplicará display none

```
<p v-show="ver">Este texto se ... </p>
```

```
data: {  
    ver: true  
}
```

VueJS

- Los mustaches (llaves doble) interpretan los datos como texto plano, no HTML.
- Para generar HTML real, se utiliza la directiva **v-html**:

```
data:{  
  codigo: "<h1> Escrito con etiquetas html se usa v-html </h1>"  
}
```

```
div v-html="codigo"></div>
```

- Si dentro de la etiqueta div tuvieramos escrito algún texto, sería reemplazado por el valor que posee la variable **codigo**
- Después veremos como crear bloques de código de otra manera.

VueJS

v-cloak--> Se le agrega al elemento hasta que la instancia de Vue asociada finalice la compilación.

- Requiere usar reglas CSS como [v-cloak] { display: none }
- Se usa para ocultar interpolaciones {{codigo}} sin compilar hasta que la instancia de Vue esté lista.

```
<div id="mostrar" v-cloak>
  <!-- todo lo que esté dentro no se mostrará hasta que se compile
  -->
</div>
```

VueJS

v-pre--> Omite la compilación para el elemento y todos sus elementos secundarios

- Sirve para mostrar mustaches sin procesar.
- Si se omiten grandes cantidades de nodos sin directivas se puede acelerar la compilación

```
<p v-pre>Directiva v-pre: {{mensajito}}</p>
```

- La variable existe, pero no la compilará

Directiva v-pre: {{mensajito}}

VueJS

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

v-bind--> Enlaza dinámicamente uno o más atributos, o una propiedad de un componente a una expresión.

Por ejemplo, si quisieramos enlazar una imagen desde nuestra lógica.

```
var app= new Vue({  
el: "#mostrar",  
data: {  
mensajito: "Dato dinamico",  
titulo: "Soy un titulo",  
imagen: "img/graptopetalum.jpg"  
}  
})
```

```
<img v-bind:src ="imagen"/>
```

VueJS

- También se puede usar la sintaxis abreviada:

```
<img :src ="imagen"/>
```

Manipular clases y estilos

- Tanto los **class** como los **style** que usamos en html son atributos que podemos afectar desde vue.
- Por medio de **v-bind** podemos manejarlos, evaluando objetos o matrices
- Podemos pasar un **objeto** para poder **asignar clases** a un elemento.

VueJS

HTML

```
<div v-bind:class="{ red: esRojo }">
```

clase

variable

CSS

```
.red{color:red;}
```

js

```
data: {  
    esRojo: true  
}
```

- También permite pasar un array para aplicar una lista de clases:

```
<p v-bind:class="[mal, error]">
```

variable

variable

VueJS

Dato dinámico



Elements Network Console Sources Performance Memory App

> <head>...</head>

▼ <body>

 ▼ <div id="mostrar">

 <h1>Esto no va a cambiar: Soy un título</h1>

 ▼ <div class="red">

 <p class="malClass errorClass"> Dato dinámico</p> == \$0

 </div>

Styles Computed Event Listeners

Filter

element.style { }

.errorClass {

background: ▢ red;

color: □ white;

}

.malClass {

 text-decoration: ▢ line-through;

VueJS

v-model--> Se aplica sobre elementos de tipo **input**, **select** y **textarea**. Se debe aclarar el estado inicial desde las opciones de data

Estilos condicionales :

```
<input type="checkbox" v-model="booleano" />
<label v-bind:style=" booleano ? 'background: green;' : 'background: red;' ">Valor del
Checkbox</label>
```

- Cuando el checkbox se marque, la variable llamada booleano tendrá valor true y se aplicará el fondo verde, caso contrario se aplicará el fondo rojo.
- También podríamos simplificarlo usando el atributo class:

```
<input type="checkbox" v-model="booleano" />
<label v-bind:class=" booleano ? 'green' : 'red' ">Valor del Checkbox </label>
```

Resumiendo

- Una aplicación de Vue consiste en una instancia raíz de Vue creada con el constructor `new Vue`.
- Puede estar organizada en un **árbol de componentes anidados y reutilizables**
- Al crearse una instancia de Vue, se **agregan todas las propiedades encontradas en el objeto data** al sistema de reactividad de Vue .
- Cuando los **valores** de esas propiedades **cambian**, la vista "reaccionará" y **se actualizará** para coincidir con los nuevos valores.

Resumiendo

- La vista y el modelo se vinculan por medio de **interpolaciones**
- Permiten que **los valores** de las **variables** declaradas en el modelo, se puedan **ver en la vista**.
- Las directivas otorgan nuevos comportamientos a los elementos del html.
- Podemos tener más de una directiva afectando un mismo elemento.



Dudas?

ADM



Dudas?

