



Antes de empezar...

- Asegúrate de tener listo tu editor (Visual Studio Code, por ejemplo) y de manejar la consola
- Asegúrate de haber bajado el proyecto de inicio desde el github de Taws.
https://github.com/Taws-Espol/front_vue
- Asegúrate de haber bajado los complementos de Vue para tu editor, y el navegador web (opcional)



Vue.js

Taller de Vue.js

Presentador: Piero Ulloa



Frameworks de Javascript

- + reactivo
- Escribir apps web + rápido.
- Mejor estructura del código



Logo del V8, uno de los motores de JS más utilizados en el mundo.

Introducción



+



¿Cómo funciona Vue.js?

- ¡Componentes!
- Reutilizables
- Reciben propiedades a través de la plantilla
- Un componente puede usar componentes internamente!
- Emiten eventos para cambiar estado
- Cada componente posee su propio estilo



Pequeña aclaración

```
class Point {  
    private double x;  
    private double y;  
  
    Point(double x, double y){  
        this.x = x;  
        this.y = y;  
    }  
}
```

Clase de Java

```
var point = {  
    x = 10,  
    y = 12  
}
```

Objeto de Javascript

Componentes

¿Cómo escribir tu primer componente?

- Vamos a utilizar componentes en archivos individuales (archivos .vue)
- Empezaremos por la estructura del archivo





Partes del componente (.vue)

- `<template>` (plantilla)
- `<script>` (código)
- `<style>`(CSS)

<template> (Plantilla)

- Utiliza html, más las directivas (v-if, v-model, v-else, etc...=)
- Hace interpolación de texto ({{text}})
- Puede instanciar otros componentes (usando tags como <comp-one></comp-one>)

components > ▼ MyComponent.vue > {} "MyComponent.vue" > 📦 template > 📦 div

You, 2 minutes ago | 1 author (You)

<template>

```
<div>
  <!-- Interpolación de texto -->
  <h2>{{myText}}</h2>
  <!-- v-model 2 way data binding -->
  <div>
    <label for="texto">Texto a mostrar: </label>
    <input id="texto" v-model="myText"/>
  </div>
  <div>
    <button v-on:click="shouldShow = !shouldShow">
      Cambiar visibilidad</button>
    <!-- Ejemplos de v-if y v-else -->
    <h2 v-if="shouldShow">
      Ahora me ves
    </h2>
    <h2 v-else>
      Ahora no me ves
    </h2>
  </div>
</div>
<div class="btn-holder">
```



<script> (código)

- data: Contiene el estado del componente
- methods: funciones llamables desde la plantillas
- Existen otros campos que se pueden usar, pero hablaremos de ellos más tarde.

```
<script>
export default {
  data: function ()
  {
    return {
      myText: "Edita el cuadro de abajo para cambiar el texto!",
      shouldShow: false,
      listItems: [
        1, 2, 3, 4
      ]
    }
  },
  methods: {
    addItem: function() {
      this.listItems.push(this.listItems.length + 1)
    },
    removeItem: function() {
      this.listItems.pop()
    }
  }
}
</script>
```



<style> (CSS)

- Encapsula los estilos del componente
- <style scoped> para estilos que aplican solo en este componente.

```
<script>
export default {
  data: function () {
    {
      return {
        myText: "Edita el cuadro de abajo para cambiar el texto!",
        shouldShow: false,
        listItems: [
          1, 2, 3, 4
        ]
      }
    },
    methods: {
      addItem: function() {
        this.listItems.push(this.listItems.length + 1)
      },
      removeItem: function() {
        this.listItems.pop()
      }
    }
  }
}
</script>
```

Directivas en Vue.js



v-if

- Muestra el contenido de la rama if, si la condición es true
- Muestra la rama que tiene la directiva v-else si la condición es falsa
- No muestra nada si no hay otro nodo v-else contiguo.



v-for

- Básicamente te permite iterar una lista y crear nodos HTML según los contenidos.
- Útil para mostrar listas dentro del HTML.
- Puedes combinar otras directivas dentro de v-for



v-show

- Parecida a v-if, solo que en este caso vue en vez de quitar y poner elementos (cuando se usa v-if, los elementos se van), en este caso les pone CSS para desaparecerlos
- Más rápido que v-if si la variable cambia constantemente.



v-model

- Hace la conexión doble vía entre los datos y la vista
- Te permite modificar en vivo los datos y mostrarlos en otra parte



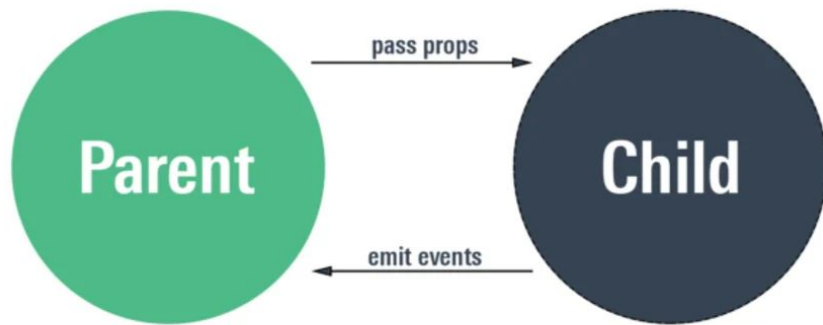
v-bind

- Esto permite asignar una propiedad del nodo html (o de un componente), como un valor de JS.
- Esto nos permite pasarle valores a los componentes hijos

Props

¿Qué son los props?

Una manera de comunicarse un componente padre hacia un componente pequeño.





¿Cómo los utilizo?

- Decláralos en tu componente, dentro del campo props,
- Úsalos directo en las plantillas, o con this en funciones.

Eventos



¿Para qué son los eventos?

- Recibir información desde otros componentes
- Manejar cambios en la página web.
- Pasar información de formularios, etc...



v-on

- Directiva para establecer "listeners" para algún evento.
- v-on:click para cuando se hace click.
- .prevent si se quiere evitar que se ejecute un handler por defecto.
- Ej: v-on:click.prevent="<JS a realizar o handler>"