

# PROYECTO DE SOFTWARE

# DIRECTIVAS VUE

# DIRECTIVAS VUE

- Son atributos específicos de Vue que comienzan con v-.
- **v-text**, **v-once**, **v-html**.
- **v-bind**, **v-model**.
- Condicionales: **v-if**, **v-else**, **v-else-if**.
- Bucles: **v-for**.
- Eventos: **v-on**.
- **v-show**.

## DIRECTIVA V-BIND:

- La **intepolación** `{{}}` no funciona para atributos, se utiliza **v-bind** para relacionar atributos con datos de Vue.
- Veamos **v-bind**.

```
<div id="app">
  <span v-bind:title="message">
    Hover your mouse over me !!
  </span>
</div>
<script>
  const App = {
    data() {
      return {
        message: 'Fecha ' + new Date().toLocaleString()
      }
    }
  }
  vue_app = Vue.createApp(App).mount('#app')
</script>
```

# DIRECTIVA CONDICIONAL **V-IF**:

- Veamos v-if.

```
<div id="app">
  <p>
    <span v-if="seen">Now you see me</span>
    <span v-else>Oh no 😞</span>
  </p>
</div>
<script>
  const App = {
    data() {
      return {
        seen: true
      }
    }
  }
  vue_app = Vue.createApp(App).mount('#app')
</script>
```

# BUCLES V-FOR:

- Veamos v-for.

```
<div id="app">
  <ul>
    <li v-for="product in products">
      {{ product }}
    </li>
  </ul>
</div>

<script>
  const App = {
    data() {
      return {
        products: [
          'Harina',
          'Arroz',
          'Yerba',
        ]
      }
    }
  }

  vue_app = Vue.createApp(App).mount('#app')
</script>
```

- Modifiquemos `vue_app.products`, agregando:  
`vue_app.products.push("Manteca")` y eliminando: `vue_app.products.pop()`.

# MÉTODOS Y EVENTOS **V-ON**:

- La directiva **v-on** nos permite actuar cuando se produzca algún **evento DOM**.
- Dentro de la sección **methods** ponemos el método que se va a disparar cuando el evento se produzca.
- Veamos **v-on**.

```
<div id="app">
  <p>{{ message }}</p>
  <button v-on:click="reverseMessage">Reverse Message</button>
</div>

<script>
  const App = {
    data() {
      return {
        message: 'Bienvenidos a Proyecto de Desarrollo!!'
      }
    },
    methods: {
      reverseMessage: function () {
        this.message = this.message.split('').reverse().join('')
      }
    }
  }
}
```



```
vue_app = Vue.createApp(App).mount('#app')  
</script>
```

## EVENTOS V-ON:

- Incluso es posible "colgarse" de múltiples eventos:

```
<div v-on="
  click      : onClick,
  keyup      : onKeyUp,
  keydown    : onKeyDown
">
</div>
```

- Notar que se modifica el estado de nuestra aplicación sin tocar el DOM, todo eso lo hace Vue.
- El código queda simplificado y enfocado en la lógica de lo que hay que resolver.

## DIRECTIVA **V-MODEL**:

- Hace la relación bidireccional entre un input y los datos de la aplicación Vue.
- Veamos v-model y v-models.

```
<div id="two-way-binding">
  <p>{{ message }}</p>
  <input v-model="message">
</div>

<script>
  const App = {
    data() {
      return {
        message: 'Bienvenidos a Proyecto!!'
      }
    }
  }
  vue_app = Vue.createApp(App).mount( '#two-way-binding' )
</script>
```

# PROPIEDADES COMPUTADAS:

- Veamos propiedades-computadas.
- Nos evita poner demasiada lógica en la visualización.

```
<div id="app">
  <ul>
    <li v-for="product in products">
      {{ product }}
    </li>
  </ul>
  Cantidad de elementos: {{ countProducts }}
</div>
<script>
const App = {
  data() {
    return {
      products: [
        'Harina',
        'Arroz',
        'Yerba'
      ]
    }
  },
  computed: {
    // a computed getter
    countProducts: function () {
      return this.products.length
    }
  }
}
```

```
vue_app = Vue.createApp(App).mount('#app')  
</script>
```

# WATCHERS:

- Veamos watcher.
- Para reaccionar cuando un dato cambia.

```
<div id="app">
  <ul>
    <li v-for="product in products">
      {{ product }}
    </li>
  </ul>
  Cantidad de elementos: {{ countProducts }}
</div>
<script>
const App = {
  data() {
    return {
      products: [
        'Harina',
        'Arroz',
        'Yerba',
      ],
      countProducts: 3
    }
  },
  watch: {
    products: {
      handler () {
        this.countProducts=this.products.length
      },
      deep: true
    }
  }
}
```

```
    }  
  }  
}  
vue_app = Vue.createApp(App).mount( '#app' )  
</script>
```

# CONSUMIENDO UNA API CON VUEJS



## EJEMPLO BÁSICO CONSULTANDO UNA API CON **FETCH**

- Veamos lista-api
- En este caso utilizamos el hook **created** dentro del ciclo de vida de la instancia Vue.

```
<div id="app">
  Datos de: https://jsonplaceholder.typicode.com/
  <ul>
    <li v-for="post in posts">
      {{ post.id }} - <b>{{ post.title }}:</b> {{ post.body }}
    </li>
  </ul>
</div>

<script>
  const App = {
    data() {
      return {
        posts: []
      }
    },
    created() {
      fetch('https://jsonplaceholder.typicode.com/posts')
        .then(response => response.json())
        .then(json => {
          this.posts = json
        })
    }
  }
}
```

```
vue_app = Vue.createApp(App).mount('#app')  
</script>
```

## UTILIZANDO EL CLIENTE HTTP **AXIOS**

- Realiza los **XMLHttpRequests** del cliente (browser).
- Realiza las peticiones HTTP en node.js (servidor).
- Soporta API de **Promesas de JS**.
- Intercepta y transforma los datos de requerimientos y respuestas.
- Transforma automáticamente a JSON.
- Soporte del lado del cliente para protección contra **CSRF**.

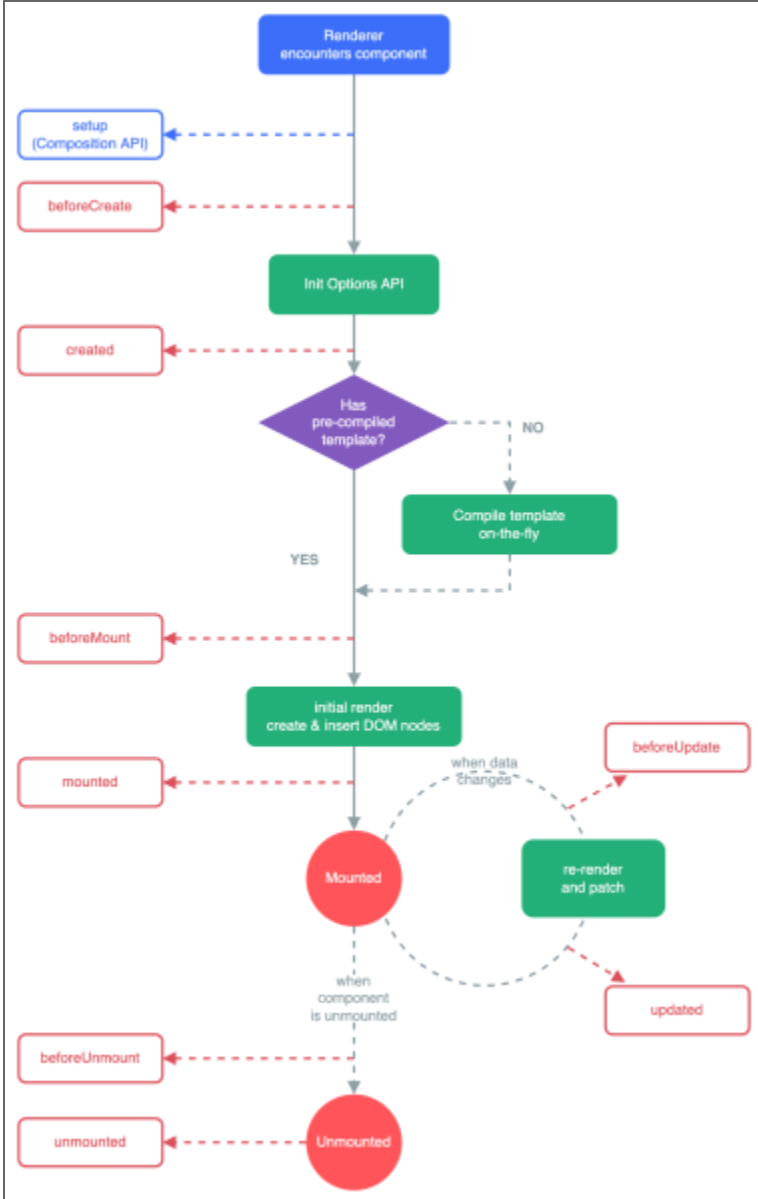
# EJEMPLO BÁSICO CONSULTANDO UNA API CON **AXIOS**

- Veamos lista-api-axios.
- Ahora, abramos directamente el archivo local en el navegador.

```
<!DOCTYPE html>
<html>
<head>
  <title>Provincias Argentinas</title>
  <meta charset="UTF-8" />
  <script src="https://unpkg.com/vue@3/dist/vue.global.js">
</script>
  <script src="https://unpkg.com/axios/dist/axios.min.js">
</script>
</head>
<body>
  <div id="app">
    <div v-if="provincias && provincias.length">
      <h1>Provincias Argentinas:</h1>
      <ul>
        <li v-for="p in provincias">
          {{ p.id }} - {{ p.nombre }}
        </li>
      </ul>
    </div>
    <div v-if="errors && errors.length">
      <h1>Errores:</h1>
      <ul>
        <li v-for="error of errors">
          {{error.message}}
        </li>
      </ul>
    </div>
  </div>
</body>
</html>
```

```
</ul>
</div>
Fuente: <a href="https://datos.gob.ar/"
target="blank">https://datos.gob.ar/</a>
</div>
<script>
  const App = {
    data() {
      return {
        provincias: [],
        errors: []
      }
    },
    created() {
      axios.get('https://apis.datos.gob.ar/georef/api/provincias')
        .then(response => {
          // JSON responses are automatically parsed.
          this.provincias = response.data.provincias
        })
        .catch(e => {
          this.errors.push(e)
        })
    }
  }
  vue_app = Vue.createApp(App).mount('#app')
</script>
</body>
</html>
```

## **CICLO DE VIDA DE UNA INSTANCIA VUE:**



## REFERENCIAS VUE

- Guia Oficial Vue: <https://vuejs.org/guide/introduction.html>
- Vue Mastery's Intro to Vue 3: <https://www.vuemastery.com/courses/intro-to-vue-3/intro-to-vue3/>
- Ejemplos Vue: <https://vuejs.org/examples/>
- Online **Playground**:



**¿DUDAS?**

**SEGUIMOS CON VUEJS LA PRÓXIMA SEMANA...**

Speaker notes