VUE.JS: COMPONENT SYSTEM

IGNACIO ANAYA
IANAYA89.COM

COMPONENTES

» ASAP

» Reutilizables (o no)

» Arbol

» Semantica

COMPONENTES
EJEMPLO DE SEMANTICA

```
<div id="app">
  <top-bar></top-bar>
  <container></container>
  <bottom-bar></bottom-bar>
</div>
```

COMPONENTES

Vue.component()

```
Vue.component('person-item', { // Este objeto tiene la misma estructura que el de app
  template: `<li> {{ person.name }} </li>`,

  props: {
    person: { Type: Object } // Es la manera de pasar información de componente padre a hijo
  }
})

const app = new Vue({
  el: '#app',

  data () {
    return {
      persons: [{ name: 'Ignacio' }, { name: 'Cristian '},{ name: 'Nicolas '}]
    }
  }
})
```

COMPONENTES
IMPLEMENTACIÓN (HTML)

```
<div id="app">
  <ul>
    <person-item v-for="p in persons" :person="p"></person-item>
  </ul>
</div>
```

COMPONENTES
EJERCICIO

1. Llevar el código a CodePen.
2. Crear los componentes del ejemplo de semántica usando Vue.component y agregarlos al HTML.
3. Pruebas con respecto a data y props (inmutabilidad).
4. Probar pasar props sin v-bind.

CICLO DE VIDA

Etapas que atraviesa un componente desde su creación hasta su destrucción.

» Funciones que agregamos al View Modal.

» Nombres predeterminados.

» Flow

CICLO DE VIDA
HOOKS

1. beforeCreate
2. created
3. beforeMount
4. mounted
5. beforeDestroyed
6. destroyed

CICLO DE VIDA
HOOKS CÍCLICOS

1. beforeUpdate

2. updated

CICLO DE VIDA
EJEMPLO

```
const app = new Vue({
    el: '#app',

    created() {
      this.log = 'created...';
    },

    mounted() {
      this.log = 'mounted...';
    }
  })
```

COMUNICACIÓN ENTRE COMPONENTES

» Padres ➡️ Hijos: Propiedades

» Hijos ➡️ Padres: Eventos

COMUNICACIÓN ENTRE COMPONENTES
HIJOS ➡️ PADRES (EVENTOS)

» Componente hijo emite evento.

» Componente padre captura evento con v-on.

COMUNICACIÓN HIJOS - PADRES (EVENTOS)
EJEMPLO (JAVASCRIPT)

```
Vue.component('modal', {
  template: `<div>
    {{ message }}
    <a href="" @click="close">X</a>
  </div>`,

  props: {
    message: { Type: String }
  },

  methods: {
    close () {
      this.$emit('close', {})
    }
  }
})

const app = new Vue({
  el: '#app',

  data () {
    return {
      show: false,
      message: 'Alert!'
    }
  },

  methods: {
    toggle () {
      this.show = !this.show
    }
  }
})
```

COMUNICACIÓN HIJOS - PADRES (EVENTOS)
EJEMPLO (HTML)

```
<div id="app">
  <modal v-show="show" @close="toggle"></modal>
  <!-- Se usa v-on de la misma manera que con un evento tradicional -->
</div>
```

COMUNICACIÓN HIJOS - PADRES (EVENTOS)
EJERCICIOS

1. Llevar el código a CodePen.
2. Cambiar el valor de la propiedad show a true.
3. Mover la lógica para cambiar el valor de show a un método.
4. Agregar el HTML y JS necesario para activar el modal con un botón.