

Curso de **Vue.js** - Clase 3

Vuenos días! 🖐️



FICTIZIA



Repaso clases 1 y 2

- Interpolación de texto con `{{ }}`
- Bind de attributes con `v-bind` o `:`
- Bind de eventos con `v-on` o `@`
- `v-model` en input text, checkbox, select...
- Renderizado condicional: `v-if`, `v-else-if`, `v-else`
- Renderizado de colecciones: `v-for`, `key`
- Lifecycle hook: `created()`, `mounted()`, `destroyed()`...
- Computed properties: `computed: {}`
- Filters: `filters: {}`
- Watchers: `watch: {}`

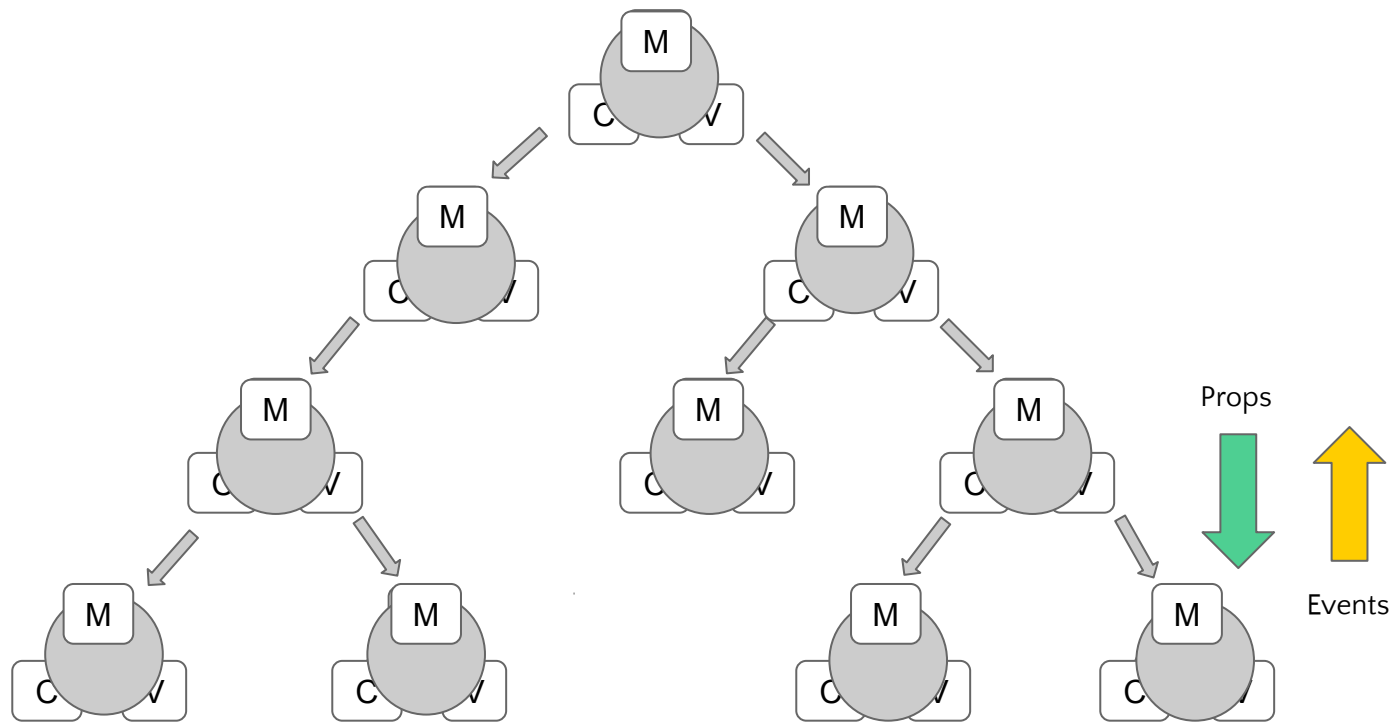


Component System





Componetización





Vue-cli (+ui)

- Vue

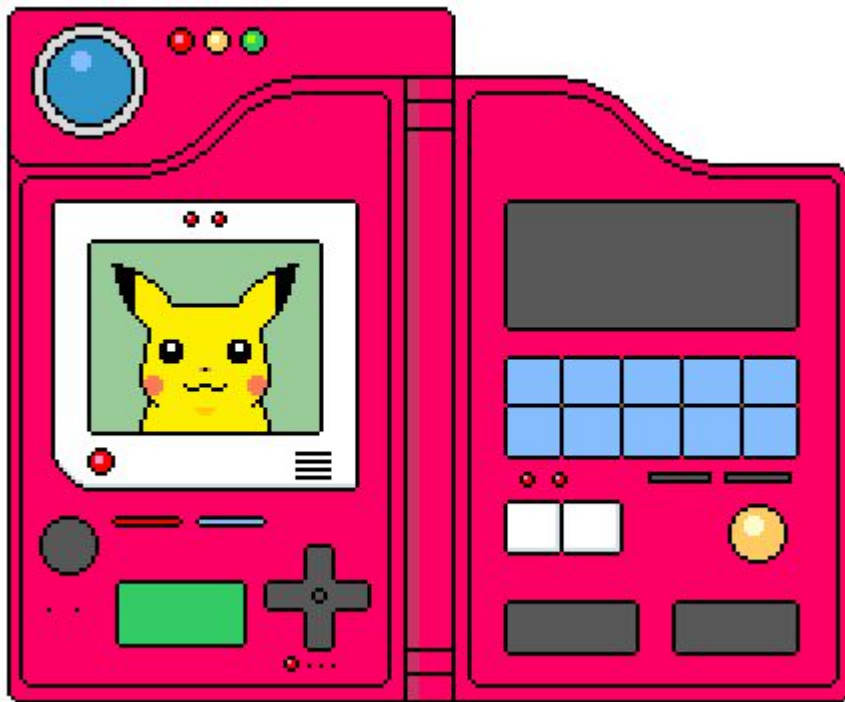
-

-

-

- Vue

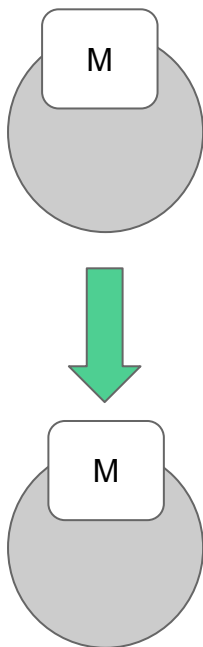
- Ver



[etur](#)



Props



Definición:

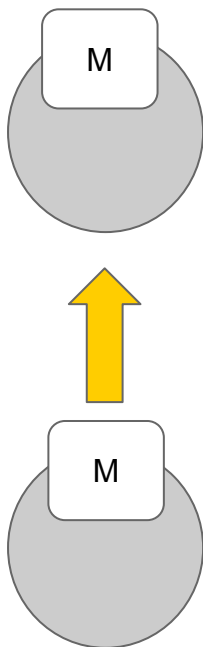
- Array
- Object
 - type, default, required, validator
- Props as events, callbacks

Herencia de atributos

- inheritAttrs
- \$attrs



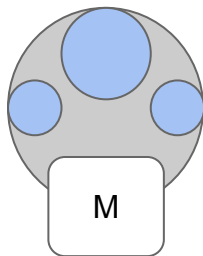
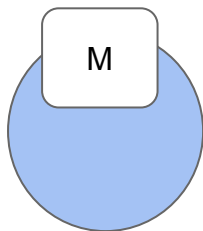
Events



- `this.$emit`
- parámetros en eventos
- v-model
 - `:value`
 - `@input`
- `.sync` modifier
 - `this.$emit('update:attr, newAttr')`



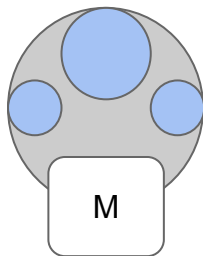
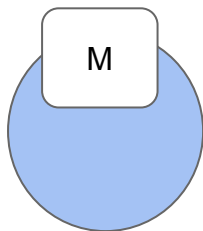
Slots



- `<slot>`
- Fallback content
- Named slots
 - `<slot name="footer">`
 - `v-slot:footer`, `#footer`



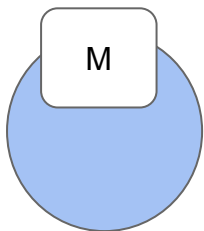
Scoped Slots



- Scoped slots
 - `<slot :user="user">`
 - `#footer="slotProps"`
- Destructuring slot props

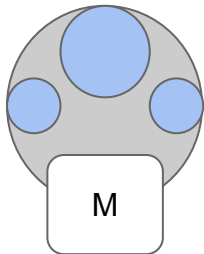


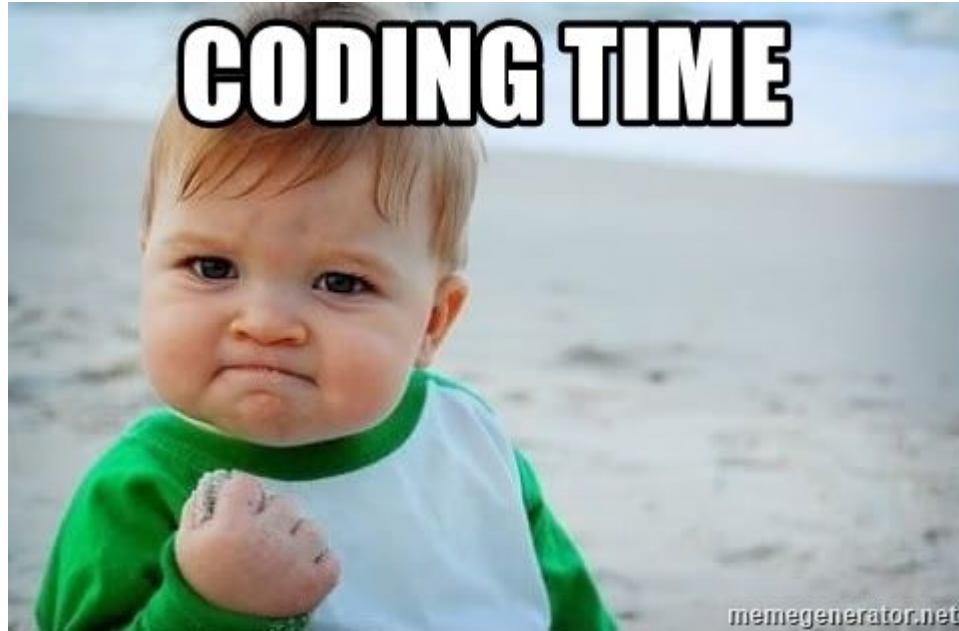
Slots & Scoped Slots



v-slot / #default

#default="{ user }"





Users list: <https://pastebin.com/RLUh8wJT>

**VUE
ROUTER**



HTML

```
<script src="https://unpkg.com/vue/dist/vue.js"></script>
<script src="https://unpkg.com/vue-router/dist/vue-router.js"></script>

<div id="app">
  <h1>Hello App!</h1>
  <p>
    <!-- use router-link component for navigation. -->
    <!-- specify the link by passing the `to` prop. -->
    <!-- `<router-link>` will be rendered as an `<a>` tag by default -->
    <router-link to="/foo">Go to Foo</router-link>
    <router-link to="/bar">Go to Bar</router-link>
  </p>
  <!-- route outlet -->
  <!-- component matched by the route will render here -->
  <router-view></router-view>
</div>
```

¿Por qué un router?

- Navegación adelante/atrás
- Refresh
- Compartir urls con vistas

JavaScript

```
// 0. If using a module system (e.g. via vue-cli), import Vue and VueRouter
// and then call `Vue.use(VueRouter)`.

// 1. Define route components.
// These can be imported from other files
const Foo = { template: '<div>foo</div>' }
const Bar = { template: '<div>bar</div>' }

// 2. Define some routes
// Each route should map to a component. The "component" can
// either be an actual component constructor created via
// `Vue.extend()`, or just a component options object.
// We'll talk about nested routes later.
const routes = [
  { path: '/foo', component: Foo },
  { path: '/bar', component: Bar }
]

// 3. Create the router instance and pass the `routes` option
// You can pass in additional options here, but let's
// keep it simple for now.
const router = new VueRouter({
  routes // short for `routes: routes`
})

// 4. Create and mount the root instance.
// Make sure to inject the router with the router option to make the
// whole app router-aware.
const app = new Vue({
  router
}).$mount('#app')

// Now the app has started!
```

Nos **Vu**emos! 🖐️



FICTIZIA