

## HINTS

### DESACOPLE DE COMPONENTE

- Como vimos en los ejercicios, cuando creamos un componente vue, y éste usa la propiedad `$route.params` para obtener los datos dinámicos, provoca una dependencia de la ruta, ya que si ésta no se encuentra presente, el componente no podrá funcionar.
- Para hacer a nuestro componente más versátil, utilizaremos props; esto lo que permite es utilizar el componente con una ruta, o usarlo dentro de cualquier otro componente, sin hacerlo dependiente de una url en el navegador.
- En el siguiente ejemplo, utilizaremos el componente `Pokemon.vue`, creado en el Exercise 1.
- Si nos damos cuenta, el componente ahora depende de una prop como entrada, lo que hace posible que éste pueda ser utilizado en múltiples partes del proyecto.

```

1 {
2   path: '/pokemon/:id',
3   name: 'Pokemon',
4   component: Pokemon,
5   props: true,
6 },

```

Si nos fijamos en la sección de script, la props "id" ahora es pasada al método `fetchPokemon`, con esto dejamos de depender de `$route.params`

```

1 <script>
2
3 import axios from 'axios'
4 export default {
5   name: "Pokemon-component",
6   props: ['id'],
7   data: function() {
8     return {
9       name: "",
10      image: "",
11    }
12  },
13  methods: {
14    fetchPokemon() {
15      axios.get(`https://pokeapi.co/api/v2/pokemon/${this.id}`)
16    }
17  },
18  .then(resp=>{
19    console.log(resp)
20    this.setData(resp.data)
21  })
22 }

```

```
20     })
21     .catch(error=>{
22         console.log(error);
23     })
24 },
25 setData(data) {
26     this.name = data.name;
27     this.image = data.sprites.front_default;
28 }
29 },
30 created() {
31     this.fetchPokemon()
32 }
33 }
34 </script>
```

## RECOMENDACIONES

### Nombra rutas dinámicas:

Cuando trabajamos con rutas dinámicas, es importante dar un nombre expresivo a esa variable, para así poder identificarla de manera sencilla.

```
1 {
2     path: '/pokemon/:id',
3     name: 'Pokemon',
4     component: Pokemon,
5     props: true,
6 },
```