

HINTS

PREVINIENDO ERRORES CON COMPUTED PROPERTIES

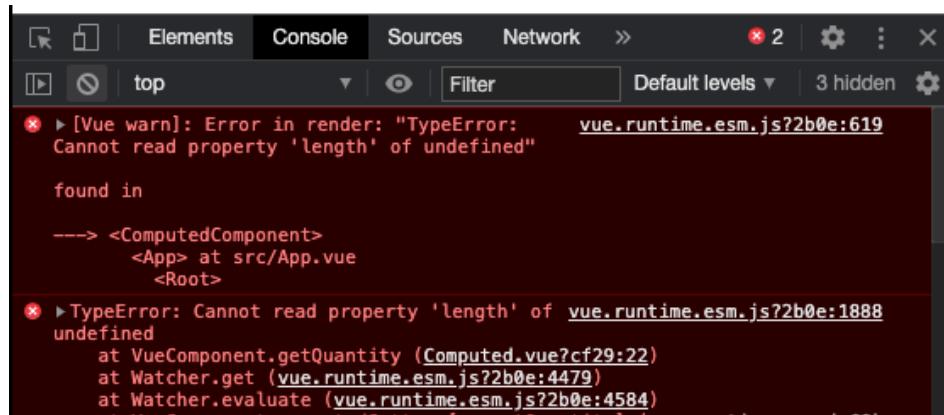
Al trabajar con propiedades computadas, siempre debemos definir los datos que usaremos dentro del objeto data; ya que éstos se ejecutan cuando el componente se renderiza, buscando un valor definido para poder hacer los cálculos.

```
1 data:function() {  
2   return{  
3     productos:[  
4       {id:1, name:"CocaCola", price:1000},  
5       {id:2, name:"PepsiCola", price:1000},  
6       {id:3, name:"Sprite", price:1000},  
7       {id:4, name:"Fanta", price:1100}  
8     ]  
9   }  
10 },
```

```
1 computed:{  
2   getQuantity(){  
3     return this.productos.length;  
4   },  
5   getTotal(){  
6     let total = this.productos.reduce((total,prod)=>{  
7       return total + prod.price  
8     },0)  
9     return total  
10  }  
11 },
```

Si nos fijamos, las dos propiedades están utilizando funciones de Array, para retornar datos como length y reduce. Si por ejemplo, declaramos el dato "productos" como undefined, provocaríamos un error.

```
1 data:function() {  
2   return{  
3     productos:undefined,  
4   }  
5 },
```



Si el dato que necesitamos utilizar viene de una llamada a una base de datos, deberíamos declarar al dato producto, como un Array vacío.

```
1 data: function () {  
2  
3   return {  
4     productos: [],  
5   }  
6 },
```

Productos 0

Total: 0

RECOMENDACIONES

Cuando trabajamos con métodos, lo ideal es separar las funcionalidades en partes que solo tengan una responsabilidad. Como lo revisamos en el ejemplo de la validación de formulario, había 3 métodos que hacían validaciones por separados, y un método que los llamaba para saber si la validación en su totalidad es correcta.

```
1 validate() {
2     if (this.titleRules() && this.emailRules() &&
3 this.commentRules()) {
4         return true;
5     } else {
6         return false;
7     }
8 },
9 titleRules() {
10    if (this.title == "") {
11        alert("Titulo es requerido");
12        this.$refs.title.focus();
13        return false;
14    } else {
15        return true;
16    }
17 },
18 emailRules() {
19    let regex = /^\\w+@[a-zA-Z_]+?\\. [a-zA-Z]{2,3}$/;
20    if (regex.test(this.email)) {
21        return true;
22    } else {
23        alert("Correo no valido");
24        this.$refs.email.focus();
25        return false;
26    }
27 },
28 commentRules() {
29    if (this.body == "") {
30        alert("Comentario es requerido");
31        this.$refs.body.focus();
32        return false;
33    } else {
34        return true;
35    }
36 },
```