

16- Empleado. Modificando el Modal

Hasta ahora hemos trabajado solo con un campo titulo por cada modelo y ocasionalmente con su id que se genera automáticamente en la db. Realmente el trabajo hasta ahora ha sido bastante sencillo. Sin embargo ahora entraremos en algunas complicaciones.

La primera complicación es la cantidad de variables nuevas. Ahora serán 5 campos + el id. En esta serie las manejaremos de forma individual. Se podría crear un objeto employee y controlarlo ahí o incluso un array con los valores.

Una vez mas la idea es la transparencia de código para entender el funcionamiento. Pero hay muchos caminos para mejorarlo. Empecemos.

Cambiamos el html para agregar el botón de agregar empleado.

```
<div class="columns">
  <div class="column text-center">
    <h3>Empleado</h3>
  </div>
  <div class="column" v-if="positions.length">
    <a class="button is-success" @click="openModal('employee','create')">Agregar Empleado</a>
  </div>
  <div class="column" v-else>
    <span class="text-danger">Debe existir un cargo por lo menos</span>
  </div>
</div>
```

Agregamos al modal

```
<p class="control" v-if="modalEmployee">
  <input class="input" :readonly="modalEmployee==3" placeholder="Nombre" v-model="nameEmployee">
  <input class="input" :readonly="modalEmployee==3" placeholder="Apellido" v-model="lastnameEmployee">
  <input class="input" :readonly="modalEmployee==3" placeholder="Correo" v-model="emailEmployee">
  <input class="input" :readonly="modalEmployee==3" placeholder="Nacimiento" v-model="birthdayEmployee">
  <label>Departamento: </label>
  <select class="select" :disabled="modalEmployee==3" v-model="idFilterDeparture">
    <option v-for="departure in filterDeparture" :value="departure.id">@{{ departure.title }}
  </option>
  </select>
  <label>Cargo: </label>
  <select class="select" :disabled="modalEmployee==3" v-model="idFilterPosition">
    <option v-for="position in filterPosition" :value="position.id">@{{ position.title }}
  </option>
  </select>
</p>
<div v-show="errorEmployee" class="columns text-center">
  <div class="column text-center text-danger">
    @{{ errorMessageEmployee }}
  </div>
</div>
```

Los 3 botones de aceptar debajo de los otros

```
<a class="button is-success" @click="createEmployee()" v-if="modalEmployee==1">Aceptar</a>
<a class="button is-success" @click="updateEmployee()" v-if="modalEmployee==2">Aceptar</a>
<a class="button is-success" @click="destroyEmployee()" v-if="modalEmployee==3">Aceptar</a>
```

Las variables en data:

```
/****** Employee *****/
idEmployee: 0,
employee: [],
modalEmployee: 0,
nameEmployee: '',
lastnameEmployee: '',
emailEmployee: '',
birthdayEmployee: '',
idFilterDeparture: 0,
filterDeparture: [],
idFilterPosition: 0,
filterPosition: [],
errorEmployee: 0,
errorMessageEmployee: ''
```

Agregamos las 3 funciones vacías en **methods**:

```
createEmployee() {},
updateEmployee() {},
destroyEmployee() {},
```

agregamos en **closeModal**

```
this.modalEmployee = 0;
```

y por ultimo en **openModal employee / create**

```
case 'create':  
  {  
    this.modalGeneral = 1;  
    this.titleModal = 'Creación de Empleado';  
    this.messageModal = 'Ingrese los datos del Empleado';  
    this.modalEmployee = 1;  
    this.nameEmployee = '';  
    this.lastnameEmployee = '';  
    this.emailEmployee = '';  
    this.birthdayEmployee = '';  
    this.idFilterDeparture = 0;  
    this.filterDeparture = [];  
    this.idFilterPosition = 0;  
    this.filterPosition = [];  
    break;  
  }  
}
```

Bien. Teniendo esto podemos abrir el modal y ver los cambios que iremos aplicando.

Los primeros campos del Modal son sencillos (Nombre y Apellido. Son simplemente campos string como hemos hecho anteriormente con titulo en los otros dos modelos.)

El Tercero es el correo electrónico. Este campo tendra dos validaciones. Una de front (El campo no puede estar vacío) y otra de back (el formato debe ser de un correo electrónico).

El cuarto (que sera el que resolveremos primero) es un problema solo de formato. **HTML 5** nos ofrece el **<input type="date">** que es de los mejores picker de fechas que me he encontrado pero no todos los navegadores son compatibles por lo que nos encontramos con el mismo problema que cuando no existia.

Como estamos usando **Vue** usaremos un picker de fecha compatible.

Los select no usan los arrays **departures** y **positions** porque ambos tienen que filtrarse. Eso lo veremos mas adelante.

Hecho el repaso superficial pasemos al siguiente Post

Lamentablemente por un error de una linea del git aparecen 4 subidas de este post. Dejo aqui el enlace valido. Mis disculpas.

Código: [Github](#)

Siguiente Post: [Usando componentes Vue.](#)