

11- Modificar Departamento

Este post es muy parecido al [anterior](#). Solo hay unos cambios lógicos porque la operación es diferente.

Paso a paso lo que haremos es:

En la tabla en la cabecera agregaremos

```
<th>Editar</th>
```

Y en la misma tabla en el Tbody agregamos

```
<td @click="openModal('departure','update',departure)">
  <i class="fa fa-pencil" aria-hidden="true"></i>
</td>
```

Modificaremos el **modal**

Donde tenemos:

```
<input class="input" placeholder="Departamento" v-model="titleDeparture" v-if="modalDeparture==1">
```

Lo modificaremos así:

```
<input class="input" placeholder="Departamento" v-model="titleDeparture" v-if="modalDeparture==1 ||
modalDeparture==2">
```

Agregando un **or (||)** al **v-if**. De esta manera usaremos el mismo **input** para crear y modificar.

Luego veremos como podemos usar el mismo **input** también para eliminar. Pero eso sera al final de este post. Continuemos:

Los botones aceptar que habíamos creado antes ahora debería quedar así:

```
<a class="button is-success" @click="createDeparture()" v-if="modalDeparture==1">Aceptar</a>
<a class="button is-success" @click="updateDeparture()" v-if="modalDeparture==2">Aceptar</a>
<a class="button is-success" @click="destroyDeparture()" v-if="modalDeparture==3">Aceptar</a>
```

El orden da igual pero si que estén las 3 opciones.

Modificaremos el **openModal** en **departure / update**

```
case 'update':
{
  this.modalGeneral = 1;
  this.titleModal = 'Modificación de Departamento';
  this.messageModal = 'Modifique el titulo del departamento';
  this.modalDeparture = 2;
  this.titleDeparture = data['title'];
  this.errorTitleDeparture = 0;
  this.idDeparture = data['id'];
  break;
}
```

El único cambio con el post anterior es el valor de **modalDeparture** que esta vez le asignamos el valor 2.

En **methods** agregamos la función **updateDeparture**

```
updateDeparture() {
  if (this.titleDeparture == '') {
    this.errorTitleDeparture = 1;
    return;
  }
  let me = this;
  axios.put('{{route('departureupdate')}}', {
    'title': this.titleDeparture,
    'id': this.idDeparture
  })
  .then(function (response) {
    me.titleDeparture = '';
    me.idDeparture = 0;
    me.errorTitleDeparture = 0;
    me.modalDeparture = 0;
    me.closeModal();
  })
  .catch(function (error) {
    console.log('error: ' + error);
  });
},
```

Agregamos la ruta

```
Route::put('/departure/update', 'DepartureController@update')->name('departureupdate');
```

y la función del controlador:

```
public function update(Request $request)
{
  $departure = Departure::find($request->id);
  $departure->title = $request->title;
  $departure->save();
}
```

Como se puede observar casi todo lo que hacemos ya lo hemos visto. Lo que cambia mas es la función del controlador que busca el departamento y cambia el titulo.

A esta altura hablaremos del código. Tanto el código del front como del back tiene otras posibilidades de codificar. Tanto Laravel como Vue ofrecen múltiples opciones y depende muchas veces de las maneras que cada uno utilice.

Mas adelante repasaremos posibilidades sobre el código existente y como mejorarlo. La idea por ahora es ser sumamente transparente en el código y agradezco la paciencia de los puristas. El único objetivo es funcional.

Ahora veamos algo que tenemos pendiente. Dejar solamente un **input** para las 3 operaciones. Si hemos seguido el tutorial tenemos que tener dos **inputs** así:

```
<input class="input" placeholder="Departamento" v-model="titleDeparture" v-if="modalDeparture==1 ||
modalDeparture==2">
<input class="input" placeholder="Departamento" v-model="titleDeparture" readonly v-if="modalDeparture==3">
```

La única diferencia real entre un **input** y otro es el atributo **readonly** pero una vez mas Vue nos da una solución:

```
<input class="input" placeholder="Departamento" v-model="titleDeparture" :readonly="modalDeparture==3">
```

Ahora este **input** se mostrara para las tres operaciones pero cuando **modalDeparture** tenga el valor 3 al **input** se le asignara el atributo **readonly**.

Sencillo y efectivo.

También modificaremos el elemento **<p>** que envuelve el input que luego de la modificación anterior quedo así:

```
<p class="control" v-if="modalDeparture!=0">
  <input class="input" placeholder="Departamento" v-model="titleDeparture" :readonly="modalDeparture==3">
</p>
```

El **v-if** que esta en **<p>** es transparente pero es lo mismo que poner **v-if=»modalDeparture»** que queda mejor.

Para finalizar el **Post** hagamos un repaso de los protocolos **http** usados aquí para las operaciones **ajax** y las acciones «recomendadas». Cada protocolo tiene sus reglas y sus recomendaciones de uso y es bueno seguir esta aproximación:

- **POST** -> Crear
- **GET** -> Leer
- **PUT** -> modificar
- **DELETE** -> borrar

Esto explica los diferentes formatos que hemos usado en **Axios** hasta ahora.

Código: [Github](#)

Siguiente post: [Comienzo del nuevo modelo](#)