7- Modal II

En el post anterior intentamos dejar el código sin errores pero también quedo sin funcionalidad.

También a lo mejor hemos agregado cosas de Vue sin explicar del todo. Repasemos el ultimo codigo puesto

```
let elemento = new Vue({
    el: '.app',
    data: {
        menu: 0,
        modalGeneral:0,
        titleModal:'',
        messageModal:'',
        modalDeparture:0,
        titleDeparture:'',
        errorTitleDeparture:0
    },
    methods: {
        closeModal(){
        },
        createDeparture(){
        }
    },
})
```

La parte de **data** de vue ya lo habíamos visto. Es donde **definimos variables**. Lo que aparece como nuevo es **methods**, es aqui donde **definimos las funciones disponibles**. La manera que estoy definiendo las funciones es sencilla pero tiene un problema de incompatibilidad con **Internet Explorer 11**. Si se desea mantener esa compatibilidad también se tiene que tomar en cuenta que **IE11**no tiene una buena relación con **ES6**. Por lo que habría que buscar otros métodos de usar este código. Si es de interés podríamos hacer un post de este tema.

Ahora dejaremos este código un poco quieto por ahora y modificaremos el HTML para crear el botón de agregar departamento.

Teníamos este código en home

```
<div class="columns">
  <div class="column text-center">
     <h3>Departamentos</h3>
  </div>
</div>
```

ahora lo cambiaremos de la siguiente manera

```
<div class="columns">
     <div class="column text-center">
           <h3>Departamentos</h3>
     </div>
     <div class="column">
          <a class="button is-success" @click="openModal('departure','create')">Agregar Departamento</a>
     </div>
</div>
```

Lo que hemos hecho es agregar un botón. En este botón usaremos una nueva función **openModal**. Esta función sera la misma para todos los eventos del CRUD de todos los elementos por lo que le indicaremos 3 parámetros.

- Elemento en que se trabaja (**Departure**, **Position** o **Employee**)
- Acción a realizar (**Create**, **Update**, **Delete**)
- En caso de update o delete se incluirá el elemento a modificar.

En este caso al ser create solo usaremos los dos primeros parámetros.

Debemos incluir esta función en **methods** de vue de la siguiente manera:

```
openModal(type, action, data = []) {
    switch (type) {
        case "departure":
            {
                switch (action) {
                    case 'create':
                        {

                            break;
                        }
                    case 'update':
                        {
                            break;
                        }
                    case 'delete':
                        {
                            break;
                        }

                }
                break;
            }
        case "position":
            {
                switch (action) {
                    case 'create':
                        {

                            break;
                        }
                    case 'update':
                        {
                            break;
                        }
                    case 'delete':
                        {
                            break;
                        }

                }
                break;
            }
        case "employee":
            {
                switch (action) {
                    case 'create':
                        {

                            break;
                        }
                    case 'update':
                        {
                            break;
                        }
                    case 'delete':
                        {
                            break;
                        }

                }
                break;
            }
    }

},
```

¿Que hemos hecho aquí? Hemos creado el esqueleto de todas las posibilidades de **openModal**. Primero se hace un **switch case** a **type**que es el primer parámetro y ya se explico arriba y luego se valora **action** que es el segundo parámetro. Dependiendo del **type + action** realizaremos una u otra tarea.

En esta primera etapa solo nos interesa **departure / create**.

En esa función pondremos el siguiente código

```
case 'create':
    {
        this.modalGeneral = 1;
        this.titleModal = 'Creación de Departamento';
        this.messageModal = 'Ingrese el titulo del departamento';
        this.modalDeparture = 1;
        this.titleDeparture = '';
        this.errorTitleDeparture = 0;
        break;
    }
```

Antes que nada una observación importante. Las variables se utilizan a través del **this**. Por ejemplo la primer variable **modalGeneral**que ya hemos visto anteriormente que dependiendo de su valor el modal tendrá la clase **is-active** o no aquí se usa con **this.modalGeneral.** Y así con todas las variables de Vue dentro de las funciones.

Miremos una a una.

- **modalGeneral** -> al tener un valor diferente de 0 el modal tendrá la clase **is-active** lo que significa que se mostrara
- **titleModal** -> La usamos para poner el titulo del Modal

- **messageModal** -> La usamos para poner un mensaje en el modal
- **modalDeparture** -> Cumple doble función. Si el valor no es 0 mostrara la zona de **Departure** pero además con el valor 1 mostrara los botones relacionados con **create** como vimos al ver el código del modal en el post anterior. Esto lo iremos repasando varias veces mas así que si aun hay dudas del concepto no hay que preocuparse
- **titleDeparture** -> Este es el **v-model** que usamos en el input. Al ser un create lo logico es que este vacío.
- **errorTitleDeparture** -> Esta variable la usamos para mostrar el error si es diferente de 0. Cambien lo hemos visto en las variables del modal. En principio es lógico su valor a 0

Esas son las variables iniciadas al abrir el modal. Ahora al hacer click en el nuevo botón el modal se abre correctamente pero tenemos un problema aun no se cierra. Eso es porque **closeModal** aun esta vacía.

Cambiaremos esa función así:

```
closeModal() {
    this.modalGeneral = 0;
    this.titleModal = '';
    this.messageModal = '';
},
```

Sencillo verdad? Simplemente reiniciamos las variables del modal.

- **modalGeneral** -> a cero cierra el modal
- **titleModal** -> la dejamos null
- **messageModal** -> la dejamos null

En realidad la única indispensable es modalGeneral a cero. Las demás se ponen a **null** por buenas practicas.

Ahora ya tenemos el modal, sus botones, el campo de titulo de departamento y el modal se abre y se cierra correctamente.

Código: Github

En el siguiente post: Primera creación