

PENSAMIENTO.VUE

```
<script>
export default {
  props:['pensamiento'],//levanto el pensamiento del padre

  Una propiedad es un atributo personalizado para pasar
  información desde componentes padres. Un componente hijo
  necesita declarar explícitamente las propiedades que
  espera recibir utilizando la opción props:

  data(){
    return{
      editMode: false//variable aux
    }
  },//data

  methods:{

    onClickDelete: function(){
      axios.delete('/pensamientos/'+this.pensamiento.id)
        .then(() => {
          this.$emit('delete');
        });
    },

    onClickEdit: function(){
      this.editMode=true;
    },

    onClickUpdate: function(){
      const params={
        descripcion:this.pensamiento.descripcion
      };
      axios.put('/pensamientos/${this.pensamiento.id}',params)
        .then(response=>{
          this.editMode=false;
          const pensamiento = response.data;
          this.$emit('update',pensamiento);
        });
    },

  },//metodos

} //export
</script>
```

TODOS LOS PENSAMIENTOS.VUE

componente general

```
<template>

  //se detecta el emit,
  //y se envia al metodo
  <form-pensamiento
    @new="addPensamiento"
  >
  </form-pensamiento>

  <pensamiento
    v-for="(pensamiento, index) in pensamientos"
    //se le agragar un indice
    :key="pensamiento.id"// si indica la key no necesario
    :pensamiento="pensamiento"
    //pasamos a los componentes hijos el objeto pensamiento
    @delete="deletePensamiento(index)"//@ evento
    @update="updatePensamiento(index, ...arguments)"
  >
  </pensamiento>

</template>

<script>
export default {
  data(){
    return {
      pensamientos:[]
    }
  },//data
  mounted(){
    axios.get('/pensamientos').then(response=>{
      this.pensamientos=response.data;
    });
  },//mounted
  methods:{
    addPensamiento: function(pensamiento){
      this.pensamientos.push(pensamiento);
    },
    deletePensamiento: function(index){
      this.pensamientos.splice(index,1); //la posicion en el arreglo, y
      //la cantidad
    },
    updatePensamiento: function(index, pensamiento){
      this.pensamientos[index]=pensamiento;
    },
  },
  //metodos
} //export
</script>
```

FORMULARIO.VUE

```
<script>
export default {
  data(){
    return{
      descripcion:""
    }
  },//data

  methods:{

    newPensamiento: function(){//toma el texto del input del
      //formulario el formulario tiene el metodo new, y el input
      //pensamiento

      const params={
        descripcion: this.descripcion
      };
      axios.post('pensamientos', params).then((response)=>{
        const pensamiento = response.data;
        this.$emit('new', pensamiento);//(evento, info)
        this.descripcion="";
      });
    },

  },

  //metodos
} //export
</script>
```