

03 Front-End con Vue.js

Intalación Vue UI

<https://cli.vuejs.org/guide/prototyping.html>

```
npm install -g @vue/cli-service-global
```

La configuración de un nuevo proyecto ya está explicado en el curso, por lo tanto en general ejecutar:

```
vue ui
```

Realizar la instalación de un nuevo proyecto utilizando Vue Router y Vuex

#Bootstrap

Una vez dentro de tu proyecto instalar <https://bootstrap-vue.js.org/docs/>:

```
npm install vue bootstrap-vue bootstrap
```

Abrir archivo `main.js`

```
import BootstrapVue from 'bootstrap-vue'

Vue.use(BootstrapVue)

import 'bootstrap/dist/css/bootstrap.css'
import 'bootstrap-vue/dist/bootstrap-vue.css'
```

Vue-axios

```
https://www.npmjs.com/package/vue-axios
```

```
npm install --save axios vue-axios
import axios from 'axios'
import VueAxios from 'vue-axios'

Vue.use(VueAxios, axios)

// Agregamos la URL base de nuestra API
axios.defaults.baseURL = 'http://localhost:3000/api';
```

#router.js

Configurar rutas: Agregaremos la siguiente ruta con una vista:

```
{
  path: '/notas',
  name: 'notas',
  component: () => import(/* webpackChunkName: "about" */
'./views/Notas.vue')
}
```

#Rutas Backend

GET (Todas las notas)

```
http://localhost:3000/api/nota
```

GET (Nota individual)

```
http://localhost:3000/api/nota/idNota
```

POST (Agregar nota)

```
http://localhost:3000/api/nueva-nota
```

DELETE (Eliminar nota)

```
http://localhost:3000/api/nota/idNota
```

PUT (Actualizar nota)

```
http://localhost:3000/api/nota/idNota
```

#View GET

Agregar una vista llamada `Notas.vue`, donde agregaremos un nuestro listado de notas:

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">Nombre</th>
      <th scope="col">Descripción</th>
      <th scope="col">Fecha</th>
      <th scope="col">Acciones</th>
    </tr>
  </thead>
  <tbody>
    <tr v-for="(item, index) in notas" :key="index">
      <th scope="row">{{item._id}}</th>
      <td>{{item.nombre}}</td>
      <td>{{item.descripcion}}</td>
      <td>{{item.date}}</td>
      <td>
        <b-button class="btn-warning btn-sm mx-2"
@click="activarEdicion(item._id)">Actualizar</b-button>
        <b-button class="btn-danger btn-sm mx-2"
@click="eliminarNota(item._id)">Eliminar</b-button>
      </td>
    </tr>
  </tbody>
</table>
export default {
  data() {
    return {
      notas: [],
    };
  },
  created(){
    this.listarNotas();
  },
  methods:{
    listarNotas(){
      this.axios.get('notas')
        .then((response) => {
          // console.log(response.data)
          this.notas = response.data;
        })
        .catch((e)=>{
          console.log('error' + e);
        })
    }
  }
}
```

```
    })  
  }  
}  
};
```

#Alertas Bootstrap

<https://bootstrap-vue.js.org/docs/components/alert#auto-dismissing-alerts>

```
<b-alert  
  :show="dismissCountDown"  
  dismissible  
  :variant="mensaje.color"  
  @dismissed="dismissCountDown=0"  
  @dismiss-count-down="countDownChanged"  
>  
  {{mensaje.texto}}  
</b-alert>  
data() {  
  return {  
    mensaje: {color: 'success', texto: ''},  
    dismissSecs: 5,  
    dismissCountDown: 0  
  };  
},  
methods:{  
  countDownChanged(dismissCountDown) {  
    this.dismissCountDown = dismissCountDown  
  },  
  showAlert() {  
    this.dismissCountDown = this.dismissSecs  
  }  
}  
}
```

View POST

```
<form @submit.prevent="agregarNota(nota)" v-if="agregar">  
  <h3 class="text-center">Agregar nueva Nota</h3>  
  <input type="text" placeholder="Ingrese un Nombre" class="form-control my-2" v-model="nota.nombre">
```

```

    <input type="text" placeholder="Ingrese una descripcion"
class="form-control my-2" v-model="nota.descripcion">
    <b-button class="btn-sm btn-block btn-success"
type="submit">Agregar</b-button>
</form>
export default {
  data() {
    return {
      notas: [],
      nota: {},
      agregar: true,
    };
  },
  methods:{
    agregarNota(item){
      this.axios.post('nueva-nota', item)
        .then(res => {
          // Agrega al inicio de nuestro array notas
          this.notas.unshift(res.data);

          // Alerta de mensaje
          this.showAlert();
          this.mensaje.texto = 'Notas Agregada!'
          this.mensaje.color = 'success';
        })
        .catch( e => {

console.log(e.response.data.error.errors.nombre.message);

          // Alerta de mensaje
          this.showAlert();
          this.mensaje.color = 'danger';
          this.mensaje.texto =
e.response.data.error.errors.nombre.message;
        })
        this.notas = {}
      },
    }
  }
};

```

View DELETE

```

<!-- Dentro del ciclo for (Tabla) -->
<b-button class="btn-danger btn-sm mx-2"
@click="eliminarNota(item._id)">Eliminar</b-button>
eliminarNota(id){

```

```

    this.axios.delete(`nota/${id}`)
      .then(res => {
        let index = this.notas.findIndex( item => item._id ===
res.data._id )
        this.notas.splice(index, 1);

        this.showAlert();
        this.mensaje.texto = 'Notas Eliminada!'
        this.mensaje.color = 'danger'
      })
      .catch( e => {
        console.log(e.response);
      })
  },

```

View PUT

```

<form @submit.prevent="editarNota(notaEditar)" v-else>
  <h3 class="text-center">Editar Nota</h3>
  <input type="text" placeholder="Ingrese un Nombre" class="form-
control my-2" v-model="notaEditar.nombre">
  <input type="text" placeholder="Ingrese una descripcion"

```

```

    class="form-control my-2" v-model="notaEditar.descripcion">
    <b-button class="btn-sm btn-block mb-1 btn-warning"
type="submit">Editar</b-button>
    <b-button class="btn-sm btn-block" @click="agregar =
true">Cancelar</b-button>
</form>

<!-- Dentro del ciclo for (Tabla) -->
<b-button class="btn-warning btn-sm mx-2"
@click="activarEdicion(item._id)">Actualizar</b-button>
data() {
  return {
    agregar: true,
    notaEditar: {},
  };
},
method:{
  activarEdicion(id){
    this.agregar = false;
    this.axios.get(`buscar/?_id=${id}`)
      .then(res => {
        this.notaEditar = res.data;
      })
      .catch(e => {
        console.log(e.response);
      })
  },
  editarNota(item){
    this.axios.put(`nota/${item._id}`, item)
      .then(res => {
        let index = this.notas.findIndex( itemNota => itemNota._id
=== this.notaEditar._id);
        this.notas[index].nombre = this.notaEditar.nombre;
        this.notas[index].descripcion =
this.notaEditar.descripcion;
        this.notaEditar = {}

        this.showAlert();
        this.mensaje.texto = 'Nota Actualizada'
        this.mensaje.color = 'success'
      })
      .catch(e => {
        console.log(e);
      })
    this.agregar = true;
  },

```

```
}
```

Producción

Ejecutar:

```
npm run build
```