

Laboratorio #4

Jeisson Andrés Vergara Vargas
Arquitectura de Aplicaciones Web
2021

El laboratorio debe ser desarrollado en parejas.

Actividades

i. Requisitos

Laboratorio #3 finalizado en su totalidad.

ii. Vue.js

a. Instalación

1. Instalar **Node.js** y **NPM**:

- [Windows/MacOS](#).
- [Linux \(Ubuntu\)](#).

2. Instalar **Vue CLI**. Usar el comando:

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

b. Creación del Proyecto

1. Ejecutar el comando:

```
vue init webpack supermarket_wa
```

2. Ingresar las siguientes entradas:

```
Project Name: supermarket_wa
Project Description: Supermarket Web Application
Author: «nombre»
Vue Build: Runtime + Compiler
Install vue-router? (Y/n): n
Use ESLint to lint your code? (Y/n): n
Set up unit test (Y/n): n
Setup e2e tests with Nightwatch? (Y/n): n
Should we run...: Yes, use NPM
```

3. Ubicarse en la raíz del proyecto y ejecutar el comando:

```
npm run start
```

4. En un navegador web, ingresar a: <http://localhost:8080>.

c. Estructura del Proyecto

1. Dentro de `../src/`, tener en cuenta:

- **assets/**: se ubican los archivos planos.
- **components/**: se ubican los componentes de la aplicación.
- **App.vue**: componente inicial de la aplicación.
- **main.js**: archivo principal de la aplicación.

2. Eliminar el archivo `/components/HelloWorld.vue`.

3. Editar el archivo **App.vue**:

```
<template>
  <div id="app">
  </div>
</template>

<script>
  export default {
    name: 'App'
  }
</script>

<style>
</style>
```

iii. Componente Web

a. Creación del Componente

1. Editar el archivo **main.js**:

```
import Vue from 'vue'
import App from './App'
import vueRouter from 'vue-router'
import router from './router'

Vue.use(vueRouter)

Vue.config.productionTip = false

new Vue({
  router,
  el: '#app',
  components: { App },
  template: '<App/>'
})
```

2. Editar el archivo **App.vue**:

```
<template>
  <div id="app">
    <div class="header">

      <h1>Supermarket</h1>
      <nav>
        <button v-on:click="init" v-if="is_auth">Inicio</button>
        <button v-on:click="categories" v-if="is_auth">Categorías</button>
        <button v-if="is_auth">Productos</button>
        <button v-if="is_auth">Cerrar Sesión</button>
      </nav>
    </div>
    <div class="main-component">
      <router-view></router-view>
    </div>
    <div class="footer">
      <h2>Arquitectura de Aplicaciones Web</h2>
    </div>
  </div>
</template>

<script>
  export default {
    name: 'App',
```

```

components: {},

data: function(){
  return {
    is_auth: localStorage.getItem('isAuth') || false
  }
},

methods: {

  init: function(){
    if(this.$route.name !== "user"){
      let username = localStorage.getItem("current_username")
      this.$router.push({name: "user", params:{ username: username }})
    }
  },

  categories: function(){
    if(this.$route.name !== "categories"){
      let username = localStorage.getItem("current_username")
      this.$router.push({name: "categories", params:{ username: username }})
    }
  },

  beforeCreate: function(){
    localStorage.setItem('current_username', 'Jeisson')
    localStorage.setItem('isAuth', true)

    this.$router.push({name: "user", params:{ username: 'Jeisson'}})
  }
}
</script>

<style>
body{
  margin: 0 0 0 0;
}
.header{
  margin: 0%;
  padding: 0;
  width: 100%;
  height: 10vh;
  min-height: 100px;

  background-color: #283747 ;
  color:#E5E7E9 ;

  display: flex;

```

```
    justify-content: space-between;
    align-items: center;
}
.header h1{
    width: 20%;
    text-align: center;
}
.header nav {
    height: 100%;
    width: 45%;

    display: flex;
    justify-content: space-around;
    align-items: center;

    font-size: 20px;
}
.header nav button{
    color: #E5E7E9;
    background: #283747;
    border: 1px solid #E5E7E9;
    border-radius: 5px;
    padding: 10px 20px;
}
.header nav button:hover{
    color: #283747;
    background: #E5E7E9;
    border: 1px solid #E5E7E9;
}
.main-component{
    height: 75vh;
    margin: 0%;
    padding: 0%;
    background: #FDFEFE ;
}
.footer{
    margin: 0;
    padding: 0;
    width: 100%;
    height: 10vh;
    min-height: 100px;
    background-color: #283747;
    color: #E5E7E9;
}
.footer h2{
    width: 100%;
    height: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
```

```
}  
</style>
```

3. En **src/**, crear un archivo llamado **router.js**:

```
import vueRouter from 'vue-router'  
import App from './App'  
import User from './components/User'  
import Categories from './components/Categories'  
  
const router = new vueRouter({  
  mode: 'history',  
  base: __dirname,  
  routes: [  
    {  
      path: '/',  
      name: "root",  
      component: App  
    },  
    {  
      path: '/user/:username',  
      name: "user",  
      component: User  
    },  
    {  
      path: '/user/:username/categories/',  
      name: "categories",  
      component: Categories  
    }  
  ]  
})  
  
export default router
```

4. En **src/components/**, crear un archivo llamado **User.vue**:

```
<template>  
  <div id="User">  
    <h2>Hola, <span>{{username}}</span>. ¡Bienvenido!</h2>  
  </div>  
</template>  
  
<script>  
  export default {  
    name: "User",  
    data: function(){  
      return {
```

```

        username: "none"
      }
    },
    created: function() {
      this.username = this.$route.params.username
    }
  }
}
</script>

<style>
  #User{
    width: 100%;
    height: 100%;

    display: flex;
    justify-content: center;
    align-items: center;
  }
  #User h2{
    font-size: 50px;
    color: #283747;
  }
  #User span{
    color: crimson;
    font-weight: bold;
  }
</style>

```

5. En **src/components/**, crear un archivo llamado **Categories.vue**:

```

<template>
  <div id="Categories">
    <h2>Usuario autenticado: <span>{{username}}</span></h2>
    <button v-on:click="getCategory">Obtener categoría</button>
    <h2>La categoría con id = <span>{{category_id}}</span>, tiene el nombre <span>{{n:
  </div>
</template>

<script>

  import axios from 'axios';

  export default {

    name: 'Categories',

    data: function (){
      return {
        category_id: "",

```

```

        name: "",
        description: ""
    }
},

created: function() {

    this.username = this.$route.params.username
    let self = this

},

methods: {

    getCategory: function() {

        this.username = this.$route.params.username
        let self = this

        axios.get("http://localhost:4000/categories/" + 1)
            .then((result) => {
                self.category_id = result.data.id,
                self.name = result.data.name,
                self.description = result.data.description
            })
            .catch((error) => {
                alert("ERROR de Servidor");
            });
    }

}

}
</script>

<style>
    #Categories{
        width: 100%;
        height: 100%;
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
    }
    #Categories h2{
        font-size: 25 px;
        color: #283747;
    }
    #Categories span{
        color: crimson;
        font-weight: bold;
    }
</style>

```


b. Ejecución del Componente

1. Detener la ejecución del componente, realizada al inicio del laboratorio.
2. Ubicarse en la raíz del proyecto y ejecutar los siguientes comandos:

```
npm install --save axios  
  
npm install --save vue-router  
  
npm run start
```

iv. Microservicio (CORS)

(supermarket_ms)

1. En el archivo **requirements.txt**, añadir la siguiente línea:

django-cors-headers

2. En el archivo **project/settings.py**, añadir:

- En **INSTALLED_APPS = [..]**

'corsheaders'

- En **MIDDLEWARE = [..]**

*'corsheaders.middleware.CorsMiddleware',
'django.middleware.common.CommonMiddleware'*

- Al final del archivo (opción 1):

CORS_ORIGIN_ALLOW_ALL = True

- Al final del archivo (opción 2):

CORS_ORIGIN_ALLOW_ALL = False

CORS_ORIGIN_WHITELIST = ('http://localhost:8080',)