Universidad Autónoma de Madrid

Departamento de Informática

Proyecto de Sistemas Informáticos
Práctica - 2.3

Roberto Marabini
Alejandro Bellogín

Registro de Cambios

Versión[1]	**Fecha**	**Autor**	**Descripción**
1.0	10.10.2022	RM	Primera versión.
1.1	17.11.2022	RM	Reformateo del código.
1.2	5.12.2022	RM	Renumerar práctica 3 ->2
1.3	12.12.2022	AB	Traducción al inglés
1.4	28.1.2023	RM	Revisar antes de subir a *Moodle*

[1]La asignación de versiones se realizan mediante 2 números $X.Y$. Cambios en Y indican aclaraciones, descripciones más detalladas de algún punto o traducciones. Cambios en X indican modificaciones más profundas que o bien varían el material suministrado o el contenido de la práctica.

Índice

1. Objetivo **3**

2. Creando el Proyecto **3**

3. Analizando los Ficheros Creados **3**

4. Creando los Componentes **5**

1. Objetivo

Cuando creamos aplicaciones *Vue.js*, casi siempre tenemos varias páginas que queremos que visite nuestro usuario. Para lograr la navegación sin tener que conectarse al back-end (servidor), usamos un "router". En *Vue.js*, vue-router es la biblioteca de "enrrutamiento". Describimos su uso en un caso sencillo a continuación.

2. Creando el Proyecto

Crea un nuevo proyecto *Vue.js* llamado `twopages`, esta vez selecciona la opción router. Esto es:

```
npm init vue@3.2 twopages
# answer as follows the questions

Add TypeScript? No** / Yes
Add JSX Support? No** / Yes
Add \vuejs Router for Single Page Application development? No /
    ↪ Yes** <<<<<<
Add Pinia for state management? No** / Yes
Add Vitest for Unit Testing? No** / Yes
Add Cypress for End-to-End testing? No** / Yes
Add ESLint for code quality? No** / Yes
Add Prettier for code formatting? No** / Yes
```

No te olvides de inicializar el proyecto como describimos (`cd twopages && npm install`)

3. Analizando los Ficheros Creados

Vamos a comentar las principales diferencias entre los ficheros creados por defecto cuando se añade y cuando no se añade el router.

main.js

Si recordáis las aplicaciones *Vue.js* se instancian en el fichero `main.js`. Ahora podéis ver que tras instanciarse la aplicación se añade el módulo router (`app.use(router)`).

Listing 1: Contenido del fichero `main.js`, las líneas relacionadas con el router se marcan con una cadena de #.

```
//src/main.js
import { createApp } from 'vue'
import App from './App.vue'
import router from './router' ################
import './assets/main.css'
const app = createApp(App)
app.use(router) ################
app.mount('#app')
```

index.js

Este fichero es similar al `urls.py` de *Django* y contiene un "mapeo" entre URLs y ficheros `vuejs`. El que viene por defecto es bastante complejo, borrarlo y copiar el código siguiente:

```
//src/router/index.js
import { createRouter, createWebHistory } from 'vue-router'
import HomeView from '../views/HomeView.vue'

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes: [
    {
      path: '/',
      name: 'home',
      component: HomeView
```

```
    },
    {
      path: '/page-one',
      name: 'pageone',
      component: () => import('../components/PageOne.vue')
    },
    {
      path: '/page-two',
      name: 'pagetwo',
      component: () => import('../components/PageTwo.vue')
    },
  ]
})

export default router
```

donde se crean tres mapeos para los URLs `/`, `/page-one` y `/page-two`.

4. Creando los Componentes

Todavía no hemos creado los componentes `page-one` y `page-two`. Crea dos ficheros llamados `PageOne.vue` y `PageTwo.vue` en el directorio `components` con el contenido:

Listing 2: Template para crear los ficheros `PageOne.vue` y `PageTwo.vue`. Cambia XXX por ONE o TWO para distinguir ambas páginas.

```
<template>
    <!-- cambia XXX por ONE or TWO -->
    <h1>Hi! I am page XXX! </h1>
</template>

<script>
    export default {}
```

```
</script>
```

igualmente vamos a simplificar el fichero que se carga por defecto (`App.vue`) y la aplicación que se carga por defecto `HomeView.vue` para que se vean más claramente el efecto de nuestros cambios en el fichero de configuración del router.

Listing 3: Contenido del fichero App.vue. Los componentes asociados a cada URL se introducirá entre los tags `<router-view></router-view>`.

```
<script setup>
import { RouterLink, RouterView } from 'vue-router'
</script>

<template>
  <div>
      <h1>Ejemplo de uso de Router</h1>
      <router-view></router-view>
  </div>
</template>

<script>
  export default {}
</script>
```

y finalmente simplifiquemos la vista que se carga por defecto `HomeView.vue`.

Listing 4: Contenido del fichero Homeview.vue.

```
<template>
  <main>
    <h1>Home Page</h1>
  </main>
</template>
```

ya puedes arrancar el servidor y conectarte a los URLs: `http://localhost:3000/`, `http://localhost:3000/page-one` y `http://localhost:3000/page-two` con lo que

obtendrás los siguientes resultados:

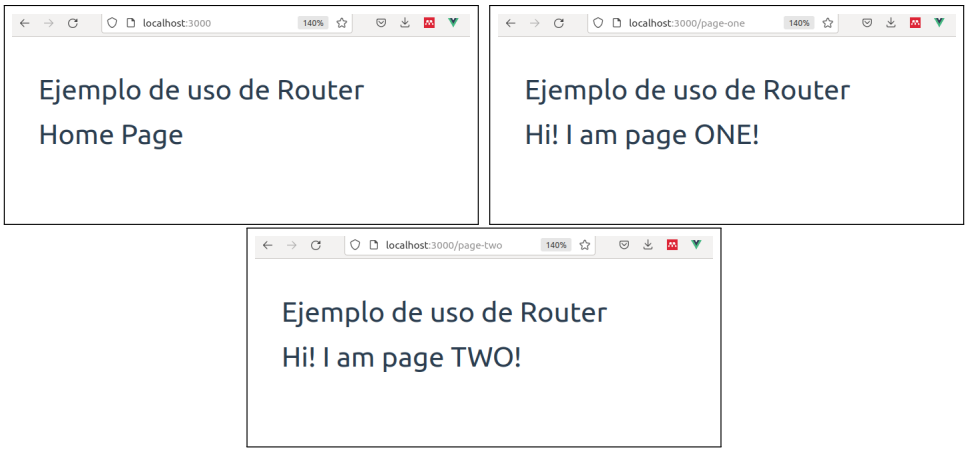

Figura 1: Páginas correspondientes a los URLs `http://localhost:3000/`, `http://localhost:3000/page-one` y `http://localhost:3000/page-two`.

Creando Links

Finalmente te puede hacer falta crear links para saltar de una página a otra. Si usas la syntaxis tradicional `<a href=‘‘...’’>link </a>` el navegador se conectará al servidor que no es lo que deseas puesto que lo que se busca es que el navegador gestione el paso de una página a otra sin que intervenga el servidor. En particular, el navegador va a estar siempre en la misma página pero va a ir inyectado diferentes componentes en cada parte. Añade las siguientes dos líneas de código al fichero `App.vue`:

Listing 5: Añadiendo links a una página *Vue.js*.

```
<a class="button" ><router-link to="/page-one">Click for Page 1</
    ↪ router-link></a> 
<a class="button" ><router-link to="/page-two">Click for Page 2</
    ↪ router-link></a>
```

ahora apareceran sendos links en la página principal que te permitirán acceder a los componentes `OnePage` y `TwoPage` tal y como se aprecia en la siguiente imagen.

Figura 2: Páginas correspondientes al URLs `http://localhost:3000/page-two` mostrando los links a los URLs , `http://localhost:3000/page-one` y `http://localhost:3000/page-two`.