code URJC

Vue

Tema 3 Aplicaciones Multipágina - Router

Micael Gallego
@micael_gallego

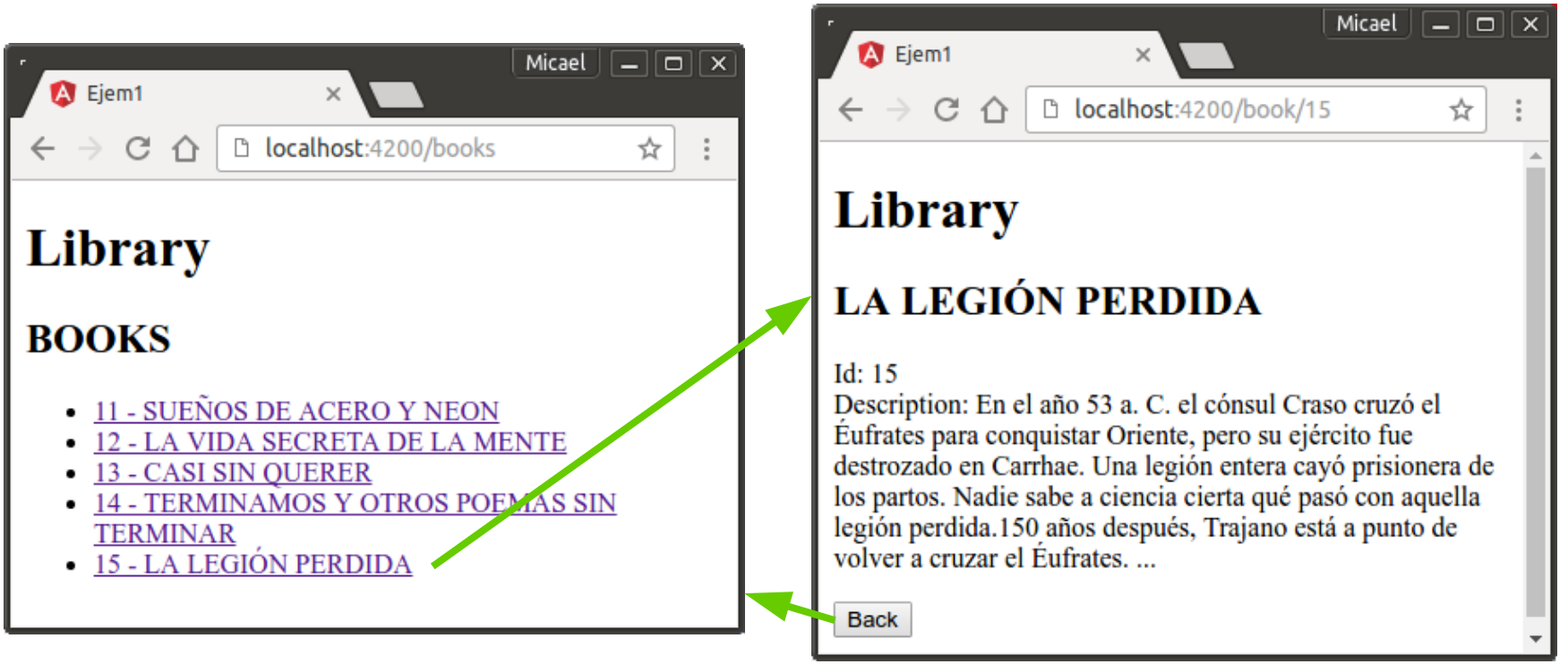
Universidad Rey Juan Carlos

Aplicaciones Multipágina

Router

Aplicaciones multipágina: Router

- Las **webs SPA** (*single page application*) pueden tener **varias pantallas** simulando la **navegación** por diferentes páginas



https://router.vuejs.org/

Aplicaciones multipágina: Router

code URJC

- En Vue, el router no viene incluido en el proyecto creado por vue-cli por defecto
- Generaremos otro proyecto seleccionando del router y dejando el resto de opciones por defecto

```
Vue CLI v3.11.0
? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select, <a> to toggle all, <i> to
 invert selection)
❯◉ Babel
 ○ TypeScript
 ○ Progressive Web App (PWA) Support
 ○ Router
 ○ Vuex
 ○ CSS Pre-processors
 ◉ Linter / Formatter
 ○ Unit Testing
 ○ E2E Testing
```

Aplicaciones multipágina: Router

- El **componente principal** de la aplicación (**app**) tiene una parte fija (**cabecera, footer**) y una parte cuyo **contenido** depende de la **URL** (**`<router-view/>`**)
- En **`router.js`** se define qué **componente** se muestra en lugar del **`<router-view/>`** para cada **URL**
- Existen **links especiales** para navegar dentro de la aplicación web (**`<router-link>`**)
- Desde el código se puede navegar de forma automática (**`this.$router.push('/path')`**)

https://router.vuejs.org/

Aplicaciones multipágina: Router

code URJC

ejem18

Componente principal

App.vue

```
<template>
  <div id="app">
    <h1 class="title">Library</h1>
    <router-view/>
  </div>
</template>
```

Aplicaciones multipágina: Router

code URJC

ejem18

Componente principal

App.vue

```
<template>
  <div id="app">
    <h1 class="title">Library</h1>
    <router-view/>
  </div>
</template>
```

Zona que cambia en función de la URL

Aplicaciones multipágina: Router

code URJC

ejem18

Configuración de las rutas

router.js

```
import Vue from 'vue'
import Router from 'vue-router'

import BookList from './components/BookList.vue'
import BookDetail from './components/BookDetail.vue'

Vue.use(Router)

export default new Router({
  mode: 'history',
  base: process.env.BASE_URL,
  routes: [
    { path: '/', redirect: '/books' },
    { path: '/books', name: 'bookslist', component: BookList },
    { path: '/book/:id', name: 'bookDetail', component: BookDetail }
  ]
})
```

Aplicaciones multipágina: Router

code URJC

ejem18

Configuración de las rutas

router.js

```
import Vue from 'vue'
import Router from 'vue-router'

import BookList fr
import BookDetail

Vue.use(Router)

export default new Router({
  mode: 'history',
  base: process.env.BASE_URL,
  routes: [
    { path: '/', redirect: '/books' },
    { path: '/books', name: 'bookslist', component: BookList },
    { path: '/book/:id', name: 'bookDetail', component: BookDetail }
  ]
})
```

Para cada URL se indica el **componente** que será visualizado en el **router-view**

ejem18

Configuración de las rutas

main.js

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

Vue.config.productionTip = false

new Vue({
  router,
  render: h => h(App)
}).$mount('#app')
```

Aplicaciones multipágina: Router

code URJC

ejem18

Configuración de las rutas

main.js

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

Vue.config.productionTip = false

new Vue({
  router,
  render: h => h(App)
}).$mount('#app')
```

Las rutas se consideran un módulo que debe importarse en la aplicación

Aplicaciones multipágina: Router

code URJC

BookList

ejem18

BookList.vue

```
<template>
  <div>
    <h2>BOOKS</h2>
    <ul>
      <li v-for="book in books" :key="book.id">
        <router-link :to="'/book/'+book.id">{{book.id}}-{{book.title}}</router-link>
      </li>
    </ul>
  </div>
</template>

<script>
import books from './books.js';

export default {
  computed:{
    books(){
      return books.getBooks();
    }
  }
}
</script>
```

Aplicaciones multipágina: Router

code URJC

BookList

ejem18

BookList.vue

```
<template>
  <div>
    <h2>BOOKS</h2>
    <ul>
      <li v-for="book in books" :key="book.id">
        <router-link :to="'/book/'+book.id">{{book.id}}-{{book.title}}</router-link>
      </li>
    </ul>
  </div>
</template>

<script>
import books from './books.js';

export default {
  computed:{
    books(){
      return books.getBooks();
    }
  }
}
</script>
```

En vez de **<a>**, los links se especifican con **<router-link>**. La URL se puede indicar como un string (**completa**) o como un **objeto** con el nombre del componente (**name**) y sus parámetros (**params**)

https://router.vuejs.org/api/#router-link

Aplicaciones multipágina: Router

code URJC

BookDetail

ejem18

BookDetail.js

```
<template>
  <div>
    <h2>{{book.title}}</h2>
    <div><label>Id:</label>{{book.id}}</div>
    <div><label>Description:</label>{{book.description}}</div>
    <p><button @click="gotoBooks()">Back</button></p>
  </div>
</template>


<script>
import books from './books.js';

export default {
  computed: {
    book(){
      let id = this.$route.params.id;
      return books.getBook(id);
    }
  },
  methods: {
    gotoBooks() {
      this.$router.push('/books');
    }
  }
}
</script>
```

Aplicaciones multipágina: Router

BookDetail

ejem18

BookDetail.js

```
<template>
  <div>
    <h2>{{book.title}}</h2>
    <div><label>Id:</label>{{book.id}}</div>
    <div><label>Description:</label>{{book.description}}</div>
    <p><button @click="gotoBooks()">Back</button></p>
  </div>
</template>

<script>
import books from './books.js';

export default {
  computed: {
    book(){
      let id = this.$route.params.id;
      return books.getBook(id);
    }
  },
  methods: {
    gotoBooks() {
      this.$router.push('/books');
    }
  }
}
</script>
```

Accedemos al parámetro **:id** con el objeto **this.$route.params**

Aplicaciones multipágina: Router

BookDetail

ejem18

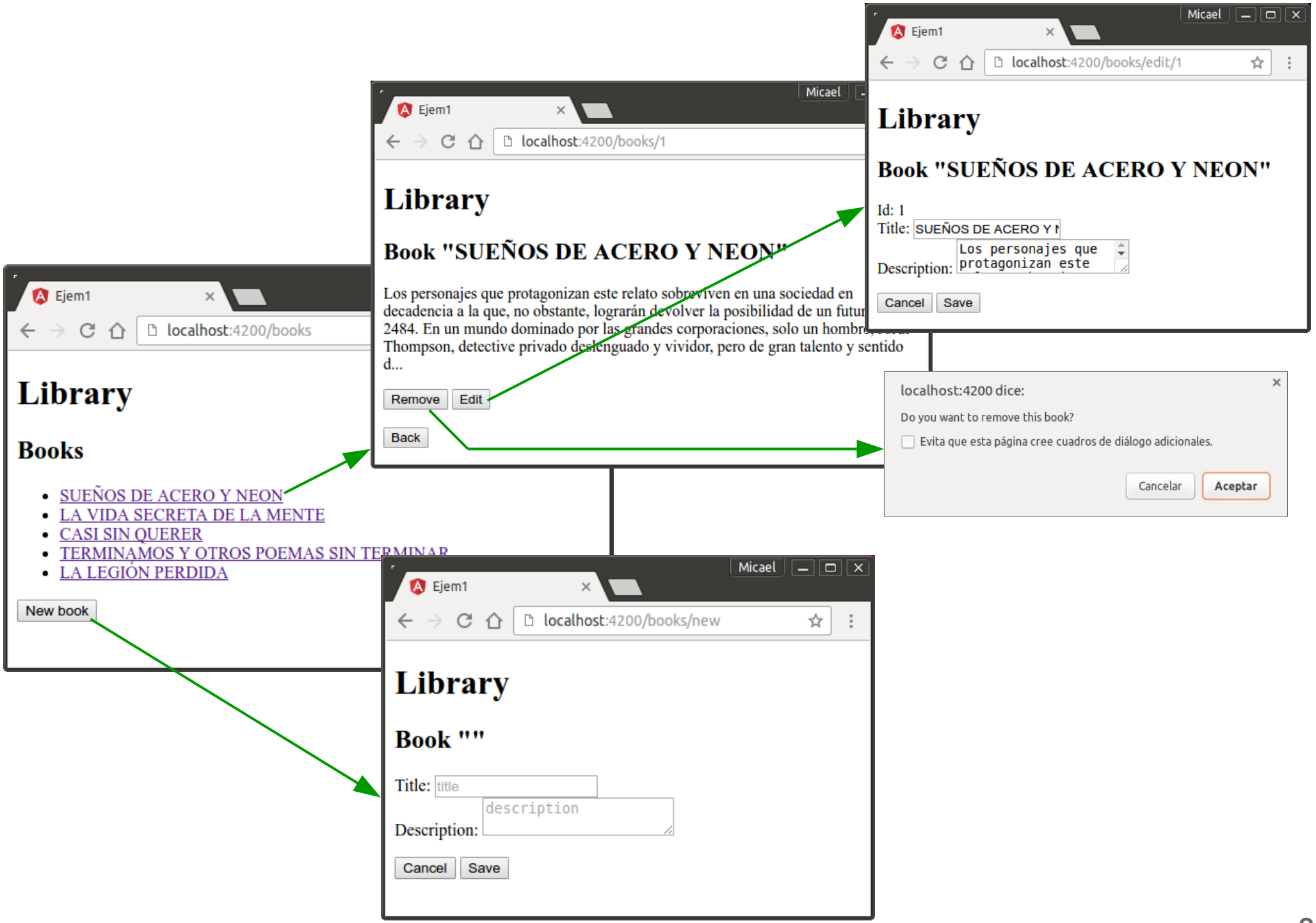
BookDetail.js

```
<template>
  <div>
    <h2>{{book.title}}</h2>
    <div><label>Id:</label>{{book.id}}</div>
    <div><label>Description:</label>{{book.description}}</div>
    <p><button @click="gotoBooks()">Back</button></p>
  </div>
</template>


<script>
import books from './books.js';

export default {
  computed: {
    book(){
      let id = this.$route.params.id;
      return books.getBook(id);
    }
  },
  methods: {
    gotoBooks() {
      this.$router.push('/books');
    }
  }
}
</script>
```

Para navegar desde código usamos el **this.$router.push(...)**

Ejercicio 4

- Implementa una **aplicación CRUD** de gestión de **libros**
- Funcionalidades (**en varias pantallas**)
  - Listado de todos los libros (títulos)
  - Formulario de nuevo libro
  - Vista de detalle de un libro
  - Modificación de libro
  - Borrado de un libro
- Cada libro tiene las propiedades: id, title, description


Ejem1
localhost:4200/books
Library
Books
SUEÑOS DE ACERO Y NEON
LA VIDA SECRETA DE LA MENTE
CASI SIN QUERER
TERMINAMOS Y OTROS POEMAS SIN TERMINAR
LA LEGIÓN PERDIDA
New book
Micael
Ejem1
localhost:4200/books/1
Library
Book "SUEÑOS DE ACERO Y NEON"
Los personajes que protagonizan este relato sobreviven en una sociedad en
decadencia a la que, no obstante, lograrán devolver la posibilidad de un futur
2484. En un mundo dominado por las grandes corporaciones, solo un hombr
Thompson, detective privado desenguado y vividor, pero de gran talento y sentido
d...
Remove
Edit
Back
Micael
Ejem1
localhost:4200/books/edit/1
Library
Book "SUEÑOS DE ACERO Y NEON"
Id: 1
Title: SUEÑOS DE ACERO Y N
Description: Los personajes que protagonizan este
Cancel
Save
localhost:4200 dice:
Do you want to remove this book?
Evita que esta página cree cuadros de diálogo adicionales.
Cancelar
Aceptar
Micael
Ejem1
localhost:4200/books/new
Library
Book ""
Title: title
Description: description
Cancel
Save

Aplicaciones multipágina: Router

- **Navegar al mismo componente con otros datos**
  - En algunas apps se navega al mismo componente pero con otros datos en la URL
  - En ese caso, se reutiliza la misma instancia del componente, no se crea una nueva
  - Para que funcione correctamente, el acceso a los parámetros siempre tiene que hacerse en una **computed property**

Aplicaciones multipágina: Router

code URJC

ejem19

- **Navegar al mismo componente con otros datos**

```
computed: {
  book(){
    let id = this.$route.params.id;
    return books.getBook(id);
  }
}
```

Aplicaciones multipágina: Router

code URJC

- **Funcionalidades avanzadas**
  - Rutas hijas (child routes) que se aplican a un componente concreto (en vez de las aplicadas a toda la app)
  - Ejecutar código al salir de una pantalla
    - Si el usuario navega a otra página "sin guardar" se le puede preguntar si realmente desea descartar los cambios o abortar la navegación
  - Verificar si se puede ir a una nueva pantalla
    - Generalmente se comprueba si el usuario tiene permisos para hacerlo
  - Las pantallas sólo se cargan al acceder a ellas (*lazy loading*)
  - Animaciones en las transiciones entre pantallas