

## 23 - Router : captura de ruta no existente

Como vimos en el concepto anterior en el archivo 'app.routes.ts' configuramos en el arreglo 'routes' todas las rutas que dispone nuestra aplicación, ahora veremos que cambios debemos hacer en el archivo para capturar una ruta inexistente e informar de tal situación al usuario.

### Problema

Confeccionar una página que muestre una barra de navegación con 4 enlaces: 'Inicio', 'Mapa', 'Contacto' y 'Acerca de..'.  
Definir una ruta para cada sección de la página y en el caso de ingresar el usuario una ruta inexistente mostrar tal situación.

- Crearemos primero el proyecto:

```
ng new proyecto018
```

- Creamos 5 componentes, 4 que representan cada una de las secciones del sitio web y una quinta de ruta inexistente (lo que conocemos "404 page not found", tener en cuenta que Angular no hace realmente una petición al servidor, sino que tiene toda la información ya cargada en el navegador)

Creación de las componentes:

```
ng generate component Inicio  
ng generate component Mapa  
ng generate component Contacto  
ng generate component AcercaDe  
ng generate component Pagina404
```

- Modificamos el archivo 'app.routes.ts' especificando las rutas de nuestra aplicación:

```

import { Routes } from '@angular/router';
import { InicioComponent } from '../inicio/inicio.component';
import { MapaComponent } from '../mapa/mapa.component';
import { ContactoComponent } from '../contacto/contacto.component';
import { AcercaDeComponent } from '../acerca-de/acerca-de.component';
import { Pagina404Component } from '../pagina404/pagina404.component';

export const routes: Routes = [{
  path: '',
  component: InicioComponent
},
{
  path: 'inicio',
  component: InicioComponent
},
{
  path: 'mapa',
  component: MapaComponent
},
{
  path: 'contacto',
  component: ContactoComponent
},
{
  path: 'acercade',
  component: AcercaDeComponent
},
{
  path: 'pagina404',
  component: Pagina404Component
},
{
  path: '**',
  redirectTo: 'pagina404'
}
];

```

La última componente del arreglo es la que indicamos en path la cadena '\*\*' y mediante el atributo 'redirectTo' especificamos a que otro path redireccionar:

```

{
  path: '**',
  redirectTo: 'pagina404'
}

```

Es indispensable para el funcionamiento que la captura de ruta inexistente con los '\*\*' sea la última del arreglo.

Si el usuario no carga ninguna ruta, sino directamente el nombre del dominio, luego tenemos que se muestra la componente 'InicioComponent':

```
{
  path: '',
  component: InicioComponent
},
```

- Procedemos a modificar la vista de la componente que crea Angular 'app.component.html' donde creamos la barra de navegación con los enlaces a las distintas rutas de nuestra aplicación:

```
<div class="menu">
  <a routerLink="inicio">Inicio</a>
  <a routerLink="mapa">Mapa</a>
  <a routerLink="contacto">Contacto</a>
  <a routerLink="acercade">Acerca de...</a>
</div>

<div>
  <router-outlet />
</div>
```

Es importante utilizar la propiedad 'routerLink' para indicar cada una de las rutas.

Disponemos la etiqueta 'router-outlet', que sabemos que es el espacio que será sustituido por la componente respectiva definida en el arreglo 'router'.

El archivo 'app.component.ts' queda con el siguiente código (importamos RouterLink):

```
import { Component } from '@angular/core';
import { RouterOutlet, RouterLink } from '@angular/router';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet, RouterLink],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'proyecto018';
}
```

La hoja de estilo para la componente la especificamos en el archivo 'app.component.css' :

```
.menu {
  display: flex;
  justify-content: space-around;
  background-color: #333;
  padding: 10px;
}

.menu a {
  color: #fff;
  text-decoration: none;
  padding: 5px 10px;
}

.menu a:hover {
  background-color: #555;
}
```

- Finalmente agregamos algunos datos a cada una de las 5 componentes que representan las secciones de nuestro sitio web:

inicio.component.html

```
<h1>Contenido de página de inicio.</h1>
```

mapa.component.html

```
<h1>Contenido de página de mapa.</h1>
```

contacto.component.html

```
<h1>Contenido de página de contacto.</h1>
```

acercade.component.html

```
<h1>Contenido de página de acerca de.</h1>
```

pagina404.component.html

```
<h1>Página no encontrada.</h1>
```

Con los pasos dados ya tenemos nuestro sitio que muestra correctamente cada una de las rutas definidas para nuestro sitio web:



Si ingresa una ruta no válida, por ejemplo "http://localhost:4200/noticias", se procede a redireccionar y mostrar la componente que hemos estipulado nosotros:



Es nuestra decisión a que página redireccionar, podemos inclusive no informar nada y redireccionarlo a la página principal del sitio.

Podemos probar esta aplicación en la web [aquí](#).

[Retornar](#)