

Ejercicio práctico Angular

Ejercicio práctico Angular

Se pide crear un proyecto Angular con los siguientes requisitos:

- El proyecto utilizará bootstrap 5
- El proyecto constará de 3 módulos, shared, users y servers.
- En el módulo app añadiremos el componente home.
- El módulo shared tendrá los componentes navbar y notFound.
- El módulo users tendrá el componentesusers.
- El módulo servers tendrá el componente servers.
- La estructura de la aplicación será la siguiente:

```
<div class="container mt-4">
  <!-- Aquí cargamos el componente navbar-->
  <div class="row my-4">
    <div class="col col-sm-10 col-md-8">
      <!-- Aquí cargamos el componente correspondiente según ruta -->
    </div>
  </div>
</div>
```

localhost:4200

Home Servers Users

NAVBAR

Server Manager App

Manage your Servers and Users

COMPONENTE SEGÚN RUTA
EN LA RUTA VACÍA SE CARGA EL COMPONENTE HOME

- El componente navbar nos permitirá navegar entre rutas y se marcará activa la pestaña de la ruta en la que nos encontremos y que se definen más abajo. Tendrá la siguiente estructura:

```
<div class="row">
  <div class="col col-sm-10 col-md-8">
    <ul class="nav nav-tabs">
      <li class="nav-item"><a class="nav-link" >Home</a></li>
      <li class="nav-item"><a class="nav-link" >Servers</a></li>
      <li class="nav-item"><a class="nav-link" >Users</a></li>
    </ul>
  </div>
</div>
```

- Nuestra aplicación tendrá las siguientes rutas:
 - Ruta vacía: Muestra el componente home
 - users: Muestra el componente users.
 - servers: Muestra el componente servers.
 - Cualquier otra: muestra el component notFound
- El componente notFound tendrá la siguiente estructura:

```
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <div class="error-template">
        <h1>
          Oops!</h1>
        <h2>
          404 Not Found</h2>
        <div class="error-details">
          Sorry, an error has occurred, Requested page not found!
        </div>
        <div class="error-actions">
          <a class="btn btn-primary btn-lg"><span class="glyphicon glyphicon-home"></span>
            Take Me Home </a>
        </div>
      </div>
    </div>
  </div>
</div>
```

Siendo necesario añadirle lo siguientes estilos css, pero solo al componente y en línea:

```
.error-template {padding: 40px 15px;text-align: center;}
.error-actions {margin-top:15px;margin-bottom:15px;}
.error-actions .btn { margin-right:10px; }
```

El botón nos llevará a la página de inicio y se verá:

localhost:4200/adfasdfaf

[Home](#) [Servers](#) [Users](#)

Oops!

404 Not Found

Sorry, an error has occured, Requested page not found!

[Take Me Home](#)

- El componente servers mostrará información sobre los servidores, para ello tendréis que crear un servicio que contenga un array con los servidores:

```
servers = [
  {
    id: 1,
    name: 'Productionserver',
    status: 'online'
  },
  {
    id: 2,
    name: 'Testserver',
    status: 'offline'
  },
  {
    id: 3,
    name: 'Devserver',
    status: 'offline'
  }
]
```

El componente servers debería ser algo similar a esto:

```

<div class="row">
  <div class="col">
    <!-- Por cada servidor crearemos una tarjeta con el nombre y
el
    estado del servidor-->
    <div class="card mt-2">
      <div class="card-header">Nombre del servidor</div>
      <div class="card-body">
        <h5 class="card-title">Server status</h5>
        <p class="card-text">Estado del servidor</p>
      </div>
    </div>
  </div>
</div>

```

Y se vería así:

localhost:4200/servers						
Home Servers Users						
<table> <tr> <th>Productionserver</th></tr> <tr> <td> Server status online </td></tr> </table> <table> <tr> <th>Testserver</th></tr> <tr> <td> Server status offline </td></tr> </table> <table> <tr> <th>Devserver</th></tr> <tr> <td> Server status offline </td></tr> </table>	Productionserver	Server status online	Testserver	Server status offline	Devserver	Server status offline
Productionserver						
Server status online						
Testserver						
Server status offline						
Devserver						
Server status offline						

- El componente users mostrará información sobre los usuarios que estará en un json-server y tendremos que obtener a través de un servicio usuarios.

Json para json-server:

```

{
  "users" : [
    {
      "id": 1,

```

```
    "name": 'Jack',
    "email": 'jack@iesjacaranda.es'
  },
  {
    "id": 2,
    "name": 'Claire',
    "email": 'claire@iesjacaranda.es'
  },
  {
    "id": 3,
    "name": 'Jamie',
    "email": 'jamie@iesjacaranda.es'
  }
]
```

Y el componente users debería ser algo similar a:

```
<div class="row">
  <div class="col">
    <!-- Mostraremos una tarjeta por cada usuario -->
    <div class="card mt-2">
      <div class="card-header">Nombre de usuario</div>
      <div class="card-body">
        <span><strong>User id:</strong> id del usuario</span>
        <br>
        <span><strong>User email: </strong> email del usuario
</span>
      </div>
    </div>
  </div>
</div>
```

Y se vería:

[Home](#) [Servers](#) [Users](#)

Jack

User id: 1

User email: jack@iesjacaranda.es

Claire

User id: 2

User email: claire@iesjacaranda.es

Jamie

User id: 3

User email: jamie@iesjacaranda.es

Extra

Crear un botón en el componente usuario que nos lleve a un nuevo componente que nos permitir añadir nuevos usuarios.