

Introducción al Enrutamiento en Angular

¿Qué son las Rutas en Angular?

Las rutas en Angular son patrones que se utilizan para identificar componentes en una aplicación. Cada ruta tiene un path, que es la parte de la URL que coincide con la ruta, y un component, que es el componente que se muestra cuando se coincide con la ruta.

Las rutas se utilizan para crear URLs que son significativas para los usuarios. Por ejemplo, una aplicación de comercio electrónico podría tener una ruta para la página de inicio, una ruta para la página de productos, y una ruta para la página del carrito de compras.

En Angular funciona mediante el uso de un módulo llamado RouterModule. Este módulo proporciona una directiva llamada RouterLink, que se utiliza para crear enlaces a rutas específicas. Cuando un usuario hace clic en un enlace RouterLink, el enrutador cambia la URL y muestra el componente correspondiente.

Paso 1: Instala Angular CLI (si aún no lo has hecho)

Para corroborar si tienes instalado angular en una terminal ejecuta el siguiente comando:

```
ng --version
```

Si no tienes Angular CLI instalado, abre tu terminal y ejecuta el siguiente comando:

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

Paso 2: Crea un nuevo proyecto Angular

Ejecuta el siguiente comando para crear un nuevo proyecto Angular. Llamaremos a nuestro proyecto "clase-enrutamiento-en-angular"(puedes elegir el nombre que prefieras):

```
ng new clase-enrutamiento-en-angular
```

Sigue las instrucciones en la terminal para configurar las opciones de tu proyecto.

"Recuerda elegir la opción de routing"

Paso 3: Navega al directorio del proyecto

Ingresa al directorio del proyecto que acabas de crear:

```
cd clase-enrutamiento-en-angular
```

Paso 4: Agrega enrutamiento a tu proyecto (Solo si no lo has elegido en la configuración inicial, cuando se instala el proyecto - Paso 2)

Angular CLI facilita la adición de enrutamiento a tu proyecto con un solo comando:

```
ng generate module app-routing --flat --module=app
```

Este comando generará un módulo de enrutamiento llamado `app-routing.module.ts` y lo importará automáticamente en el módulo principal de la aplicación (`app.module.ts`).

Paso 5: Crea componentes

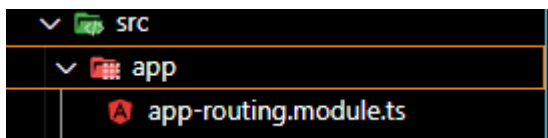
Crea los componentes para las páginas que quieras mostrar. Por ejemplo:

```
ng generate component inicio
```

```
ng generate component contacto
```

Paso 6: Configura las rutas en `app-routing.module.ts`

Abre el archivo **`app-routing.module.ts`** en tu editor de código:



Y configura las rutas. Por ejemplo, para una aplicación con una página de inicio y una página de contacto, puedes definir las rutas de la siguiente manera:

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

// Importar los componentes
import { InicioComponent } from '../inicio/inicio.component';
import { ContactoComponent } from '../contacto/contacto.component';

const routes: Routes = [
  {path: 'inicio', component: InicioComponent},
  {path: 'contacto', component: ContactoComponent},
  {path: '', redirectTo: '/inicio', pathMatch: 'full'},
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Explicación del Código:

Las siguientes dos líneas importan los componentes InicioComponent y ContactoComponent. Estos componentes son las vistas que se mostrarán cuando el usuario navegue a las rutas /inicio y /contacto, respectivamente.

La declaración const routes: Routes = [...] define un array de rutas. Cada ruta tiene una ruta y un componente asociado. Cuando el usuario navega a una ruta, el componente correspondiente se renderizará.

La primera ruta del array tiene una ruta de inicio y un componente de InicioComponent. Esto significa que cuando el usuario navegue a la ruta /inicio, se renderizará el InicioComponent.

La segunda ruta del array tiene una ruta de contacto y un componente de ContactoComponent. Esto significa que cuando el usuario navegue a la ruta /contacto, se renderizará el ContactoComponent.

La tercera ruta del array tiene una ruta de `` y una ruta de redireccionamiento de /inicio. Esto significa que cuando el usuario navegue a la URL raíz (es decir, localhost:4200), se redireccionará a la ruta /inicio.

Paso 7: Agrega enlaces de navegación en tu aplicación

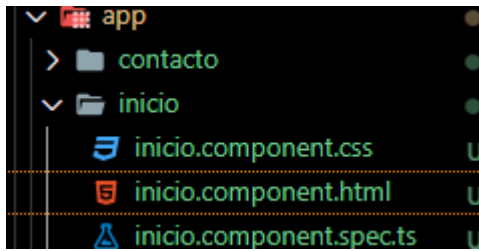
Abre el archivo app.component.html borramos todo su contenido menos la etiqueta (`<router-outlet></router-outlet>`) y agrega enlaces de navegación utilizando la directiva routerLink:

```
<nav>
  <ul>
    <li><a routerLink="/inicio">Inicio</a></li>
    <li><a routerLink="/contacto">Contacto</a></li>
  </ul>
</nav>

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

Paso 8: Crea un Front básico para inicio y contacto

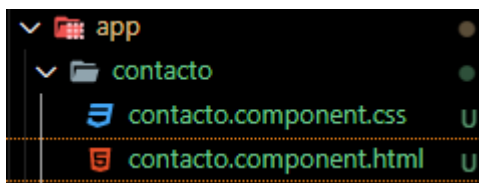
Nos iremos a inicio.component.html:



y crearemos su Front:

```
<h1>Inicio</h1>
<p>Esta es la pagina de inicio</p>
```

Lo mismo hacemos con contacto.component.html:



```
<h1>Contacto</h1>
<form action="" method="POST">
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre" required>
  <br><br>
  <label for="email">Correo electrónico:</label>
  <input type="email" id="email" name="email" required>
  <br> <br>
  <label for="mensaje">Mensaje:</label>
  <textarea id="mensaje" name="mensaje" rows="4" required></textarea>
  <br><br>
  <input type="submit" value="Enviar">
</form>
```

Paso 9: Ejecuta tu aplicación

Puedes iniciar tu aplicación con el siguiente comando:

```
ng s -o
```

La aplicación mostrará el contenido correspondiente de los componentes InicioComponent y ContactoComponent.

Trabajo Práctico: Desarrollo de una Aplicación Web con Angular y Bootstrap

Objetivo del Trabajo Práctico:

El objetivo principal de este trabajo práctico es que adquieran experiencia en el desarrollo de una aplicación web simple utilizando Angular y Bootstrap (pueden instalarlo en la aplicación que es lo más recomendable o usar los CDN). También se espera que comprendan cómo funcionan las rutas en Angular y cómo estructurar una aplicación de una manera organizada.

“Todo lo que se pide está en el cuadernillo.”

Requisitos Generales:

- Utilizar Angular para crear la aplicación web.
- Diseñar un navbar y contenga enlaces a las 4 rutas mencionadas.
- Cada ruta debe tener su propio contenido y diseño relacionado con su tema.
- Utilizar componentes de Bootstrap para el diseño y diseño responsivo de la aplicación.
- Los títulos de cada página (componente) se tiene que hacer con la interpolación.
- Opcionalmente, usar interpolación para todos los string que usen en la aplicación.
- Opcionalmente, subir el proyecto a un repositorio público en GitHub.
- Opcionalmente, agregar un footer con información de los alumnos en todas las páginas.

Rutas (los components que se tiene que crear en el proyecto):

Inicio:

En la página de inicio, mostrar los siguientes datos de los alumnos:

1. Nombre de los alumnos.
2. Carrera que están cursando.
3. Año de cursado.
4. Nombre de la materia.
5. Página de GitHub (puede ser un enlace).
6. Opcional: Enlace a los perfiles de LinkedIn de los alumnos.

Utilizar Bootstrap para organizar y diseñar la información de manera atractiva.

Angular:

1. Utilizar tarjetas (cards) de Bootstrap para presentar el contenido.
2. Proporcionar un tutorial breve sobre los siguientes temas relacionados con Angular:
 - a. Cómo instalar Angular.
 - b. Cómo crear componentes en Angular.
 - c. Comandos básicos que se utilizan en Angular.
 - d. Explicación de qué es Angular.
 - e. Opcional: Cómo utilizar el enrutamiento (router) en Angular.

Git:

1. Elegir un componente de Bootstrap (por ejemplo, una lista desplegable) para organizar los comandos de Git.
2. Proporcionar una lista de comandos de Git y explicar para qué se utilizan.

Terminales:

1. Utilizar un diseño limpio y ordenado con Bootstrap para esta página.
2. Proporcionar información sobre el uso de terminales (puede ser sobre la terminal de comandos de un sistema operativo).
3. Incluir comandos de terminal y explicar su funcionalidad.