

## **Parte 1**

En este ejemplo vamos a estudiar cómo usar el módulo de enrutamiento (**RouterModule**), y cómo usar la librería de componentes **Angular Material**.

### **Creamos nuestra aplicación**

- El primer paso es, usando el CLI, crear nuestra nueva aplicación .  
Ejecutamos:

```
ng new holaMundoRoute --no-standalone
```

- Al preguntar SSR y SSG, pulsamos **y** (yes), lo demás por defecto.

- Cambiamos al directorio de la aplicación: `cd holaMundoRoute`

### **Creamos un componente menú que va a contener un (toolBar) y tres botones.**

Añadimos los paquetes de Angular Material, nos facilitan la creación de componentes..

```
ng add @angular/material
```

 , aceptamos el tema por defecto y respondemos **'yes'** a las siguientes preguntas sobre instalación de otros paquetes (animaciones, estilos ...)

Generamos un primer componente principal desde la librería **Angular Material** para generar un panel de botones que apunten a las distintas páginas.Primeramente:

```
ng g component menu
```

El componente creado se habrá añadido automáticamente a nuestro módulo raíz. Nos vamos a <https://material.angular.io/components/toolbar/overview> (versión 15-19) y vemos en la pestaña API, que es necesario importar el módulo del componente a nuestro módulo raíz y añadirlo al array **imports** para su uso.

```
import {MatToolbarModule} from '@angular/material/toolbar';
```

Ahora vamos a añadir cada código (TS, HTML, CSS) en sus archivos correspondientes (hay casos en los que no es necesario ni usar, ni añadir nada al TS, ya que el componente nuevo está creado con su nombre de atributo <<**selector**>> establecido, en este caso **app-menu** , y tampoco se tiene que acceder a ningún otro atributo o propiedad).

Nos vamos a ejemplos o en la pestaña principal del componente de **Angular Material** y elegimos el código HTML que debemos copiar y pegar (< >) en el nuevo componente **app-menu** (Basic Toolbar) .

Modificamos ese HTML copiado y añadimos 3 botones, (Home, Productos y Contacto) dentro del **toolBar** haciendo uso del componente **button** de Angular Material, y teniendo en cuenta la importación del módulo que lo contiene.

El archivo **menu.component.html** debe ser algo como :

```
<mat-toolbar>

  <button mat-flat-button>Home</button>

  <button mat-flat-button>Productos</button>

  <button mat-flat-button>Contacto</button>

</mat-toolbar>
```

y en el archivo **menu.component.css** ,

```
.mat-mdc-button-base {

  margin: 8px 8px 8px 0;

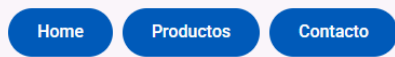
}
```

Seguidamente reemplazamos todo el contenido de **app.component.html** por :

```
<app-menu></app-menu>
```

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

Probamos: `ng serve -o`, debe quedar algo como :



**Generamos 4 componentes que irán asociados a nuestras páginas a enrutar.**

```
ng generate component home
```

```
ng g c productos
```

```
ng g c contacto
```

```
ng g c pagenotfound
```

De momento los dejamos con el contenido HTML por defecto.

Para crear las rutas a las distintas páginas vamos a asociar los componentes generados a cada una de ellas. Para ello **modificaremos** el módulo **app-routing.module.ts** que se creó automáticamente al generar la aplicación. Este módulo facilita la creación de rutas y enlaces a cada página.

Antes de nada, y teniendo que hacer referencia a cada componente (página) en dicho módulo, y perteneciendo éstos a otro módulo distinto (al principal), es necesario importarlos :

```
import { HomeComponent } from './home/home.component';  
import { ProductosComponent } from './productos/productos.component';
```

```
import { ContactoComponent } from './contacto/contacto.component';  
import { PagenotfoundComponent } from  
'./pagenotfound/pagenotfound.component';
```

A continuación creamos las rutas y las asociamos a los componentes:

```
const routes: Routes = [  
  { path: 'home', component: HomeComponent },  
  { path: 'productos', component: ProductosComponent },  
  { path: 'contacto', component: ContactoComponent },  
  { path: '', redirectTo: '/home', pathMatch: 'full' },  
  { path: '**', component: PagenotfoundComponent }  
];
```

Los dos últimos **path**, redireccionan al **home** cuando la ruta está vacía, y con **\*\***, cualquier otra página que no exista se deriva al componente de **página no encontrada**, respectivamente.

Seguidamente, modificamos **menu.component.html** para crear los enlaces (**routerlink**) en los botones y que apunten a cada ruta:

```
<mat-toolbar>  
  <button mat-flat-button routerLink="/home">Home</button>  
  <button mat-flat-button routerLink="/productos">Productos</button>  
  <button mat-flat-button routerLink="/contacto">Contacto</button>  
</mat-toolbar>
```

Probamos que podemos navegar por las páginas:



home works!

### Rutas de anidamiento

Cuando la aplicación se hace más compleja, es posible que sea necesario enrutar a distintas páginas en una determinada vista/componente, para ello se usan las rutas secundarias (hijas), relativas a un componente. Primero se añade una nueva etiqueta **<router-outlet><router-outlet>** en la plantilla del componente padre, y luego se añaden las rutas como cualquier otra ruta con la diferencia de que se establecen hijas con la propiedad **children**.

Para implementarlo, creamos dos componentes (que serán hijos de **home**)

```
ng g c childA
```

```
ng g c childB
```

Sustituimos el contenido de la plantilla de **home** por :

```
<h2>Home</h2>

<nav>

  <ul>

    <li><a routerLink="/home/child-a">Child A</a></li>

    <li><a routerLink="/home/child-b">Child B</a></li>

  </ul>

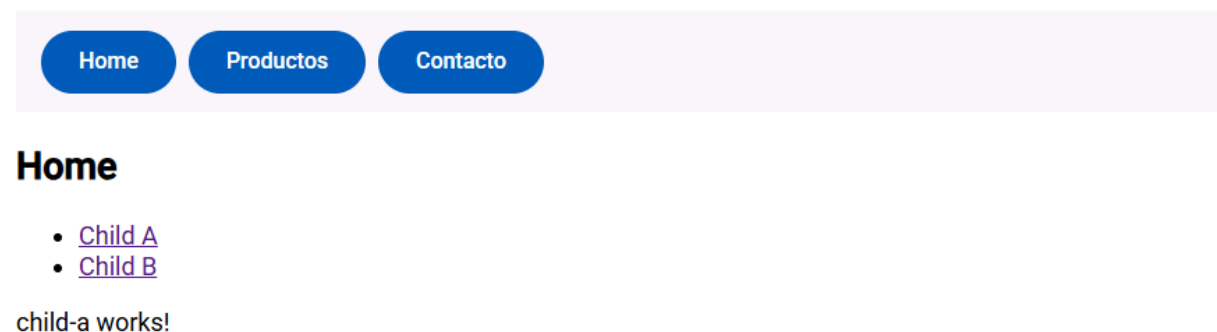
</nav>
```

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

En **app-routing.module.ts** sustituimos el **path** de **home** por : (importando los componentes al módulo)

```
{ path: 'home',  
  component: HomeComponent,  
  children: [  
    {  
      path: 'child-a', // child route path  
      component: ChildAComponent,  
    },  
    {  
      path: 'child-b',  
      component: ChildBComponent,  
    },  
  ],  
},
```

Si todo ha ido bien, debería de verse algo como :



Compilamos y desplegamos. **IMPORTANTE:** comprobamos que los enrutamiento funcionan correctamente una vez desplegados , haciendo pruebas de refresco de cada ruta o enlace.

## Parte 2

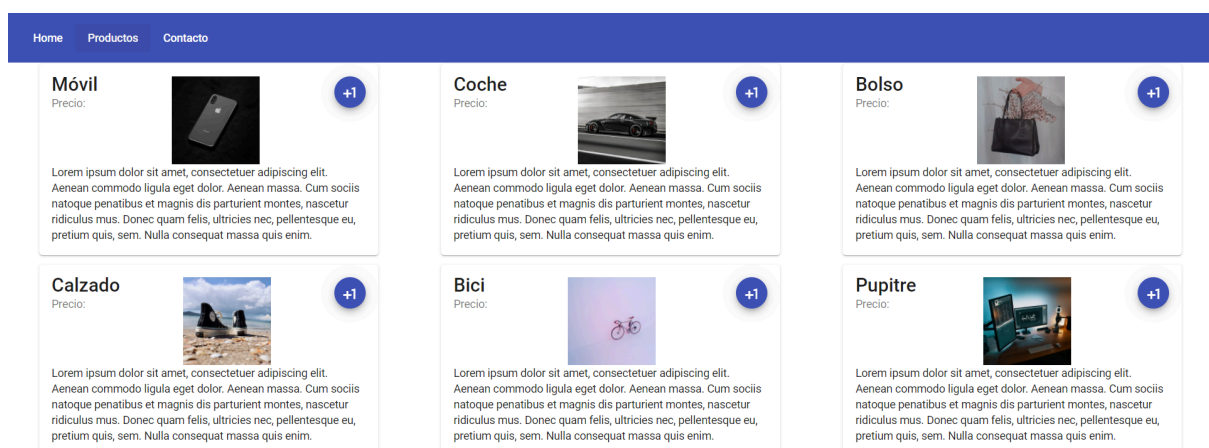
Creamos el mismo proyecto pero sin activar SSR/SSG, copiaremos todos los archivos necesarios creados en la **parte 1**, desplegamos (**dist**) y probamos de nuevo los enrutamientos.

## Parte 3

Ahora para completar la aplicación y enriquecer las páginas, vais a usar (libremente) , componentes de **Angular Material** . En el **Home** pondremos componentes que muestren una descripción de una tienda de productos a modo de presentación, en la página **Productos** , usad componentes **grid** y **card** (por ejemplo) para crear fichas de productos , y en **contacto** , vamos a añadir un formulario de contacto. Finalmente compilamos y desplegamos.

Aclarar, que si vamos a hacer uso de imágenes, iconos , etc ... deben estar alojados en la carpeta **./public**

Ejemplo para **productos** y **contacto**:



### [Tarea 3.Tema 14. Ejemplo guiado sobre: Routing y Angular Material](#)

Home			Productos			Contacto		
Company								
Empresa S.A.								
First name			Long Last Name That Will Be Trun...					
Address								
1600 Amphitheatre Pkwy								
Address 2								
City			State			Postal Code		
						94043		

5 / 5

**NOTA:** Solo entregar la **parte 3** como tarea.