

# Desarrollando un CRUD en ANGULAR

---

Parte 2  
Routing

Mct. Esteban Calabria



# Enunciado

---

En el laboratorio anterior logramos mostrar una lista de contactos en una tabla y además entendimos la estructura básica de un proyecto de angular como así también los elementos fundamentales de este framework como binding de atributos.

Ahora vamos a implementar la siguiente funcionalidad:

- Agregar un contacto nuevo
- Separar el agregar en otra url mediante el uso de angular routing
- Implementar la funcionalidad de eliminar
- Implementar la funcionalidad de editar



# Crear Componentes Necesarios

## Paso 1

- Armaremos la estructura de componentes para probar la navegación
- Ya tenemos el siguiente componente
  - ListadoContacto
- Crear componentes necesarios
  - CrearContacto
  - EditarContacto
  - DetalleContacto (Opcional)
- Modificar el ListadoContacto para agregar la navegación



Mct Esteban Calabria

# Crear Componentes Necesarios

---

Primero armaremos la estructura de componentes para probar la navegación para poder luego desarrollar la funcionalidad correspondiente

```
C:\Cursos\Angular\Angular-PWA\Material\crud-demo-parte-2>ng generate component /components/CrearContacto --skip-tests
CREATE src/app/components/crear-contacto/crear-contacto.component.html (30 bytes)
CREATE src/app/components/crear-contacto/crear-contacto.component.ts (277 bytes)
CREATE src/app/components/crear-contacto/crear-contacto.component.css (0 bytes)
```

```
C:\Cursos\Angular\Angular-PWA\Material\crud-demo-parte-2>ng generate component /components/EditarContacto --skip-tests
CREATE src/app/components/editar-contacto/editar-contacto.component.html (31 bytes)
CREATE src/app/components/editar-contacto/editar-contacto.component.ts (281 bytes)
CREATE src/app/components/editar-contacto/editar-contacto.component.css (0 bytes)
```

```
C:\Cursos\Angular\Angular-PWA\Material\crud-demo-parte-2>ng generate component /components/DetalleContacto --skip-tests
CREATE src/app/components/detalle-contacto/detalle-contacto.component.html (32 bytes)
CREATE src/app/components/detalle-contacto/detalle-contacto.component.ts (285 bytes)
CREATE src/app/components/detalle-contacto/detalle-contacto.component.css (0 bytes)
```

# Implementar Angular Routing

## Paso 2

- Definir rutas
- Asociar Ruta a Componente



Mct Esteban Calabria

# Antes de continuar debe saber lo siguiente



## ¿Que es una SPA?

- <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

## Angular Routing

- <https://docs.angular.lat/tutorial/toh-pt5>
- <https://medium.com/@jsmuster/enrutamiento-de-angular-en-5-minutos-spanish-dfbadc2c1cb7>
- <https://www.youtube.com/watch?v=r5DEBMuStPw>

# Generar las siguientes rutas

---

```
import { Routes } from '@angular/router';
import { ListadoContactosComponent } from '../components/listado-contactos/listado-contactos.component';
import { CrearContactoComponent } from '../components/crear-contacto/crear-contacto.component';
import { DetalleContactoComponent } from '../components/detalle-contacto/detalle-contacto.component';
import { EditarContactoComponent } from '../components/editar-contacto/editar-contacto.component';

export const routes: Routes = [
  { path: '', component: ListadoContactosComponent },
  { path: 'contactos', component: ListadoContactosComponent },
  { path: 'contactos/nuevo', component: CrearContactoComponent },
  { path: 'contactos/:id', component: DetalleContactoComponent },
  { path: 'contactos/editar/:id', component: EditarContactoComponent }
];
```

Archivo app.routes.ts

# Preguntas

---

¿Que es una SPA?

¿Se refresca la pagina?

¿Que es el atributo routerLink?

¿Que es el componente router-outlet?





# Implementar Angular Routing

## Paso 3

- Navegar al agregar
  - Probar el agregar
  - router-link



Mct Esteban Calabria

# Vamos a navegar al agregar contacto desde el listado-contacto.component



UPS!! Se me muestra siempre el link de agregar contacto.

¿Cómo lo solucionarían?

# Navegar al agregar contacto desde el app.component

---

```
<h1>{{title}}</h1>

<a routerLink="contactos/nuevo">
  Agregar Contacto Nuevo
</a>

<router-outlet></router-outlet>

<!--
  Reemplazamos por el Router Outlet para que se
  muestre el componente que corresponde segun la url

<app-listado-contactos></app-listado-contactos>
-->
```

Archivo app.component.html

```
import { Component } from '@angular/core';
import { ListadoContactosComponent } from
  './components/listado-contactos/listado-contactos.component'
;

import { RouterOutlet, RouterModule } from
  '@angular/router';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [ListadoContactosComponent, RouterOutlet,
    RouterModule],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'crud-demo';
}
```

Archivo app.component.ts

# routerLink

Directiva en Angular que permite la navegación entre componentes.

Utiliza la configuración de rutas definida en el módulo de enrutamiento de Angular.

## Características:

- **Enlace de navegación:**
  - Genera enlaces que permiten navegar a rutas especificadas.
  - Reemplaza el atributo href de HTML en aplicaciones Angular.
- **Sintaxis:**
  - Uso básico: `[routerLink]="['/ruta']"`.
  - Puede aceptar parámetros de ruta: `[routerLink]="['/ruta', parametro]"`.
- **Interpolación:**
  - Soporta la interpolación de datos en la URL.
  - Ejemplo: `[routerLink]="['/ruta', { id: 1 }]"`.
- **Integración con eventos:**
  - Compatible con eventos como click.
  - Puede utilizarse con `routerLinkActive` para resaltar el enlace activo.

—

# router-outlet

Directiva en Angular que actúa como un marcador de posición para cargar componentes basados en la ruta actual.

Define dónde se deben mostrar los componentes mapeados en las rutas.

## Características:

- **Contenedor de vistas:**
  - Espacio en la plantilla donde se insertarán los componentes en función de la ruta activa.
- **Componentización:**
  - Permite la carga de diferentes componentes sin cambiar la estructura principal de la aplicación.
- **Configuración en el enrutador:**
  - Las rutas se configuran en el módulo de enrutamiento utilizando RouterModule.

# Implementar Angular Routing

Paso 4

- Navegar al Editor
- Navegar programáticamente



Mct Esteban Calabria

# Probar el editar

Vamos a agregar al listado de contactos un botón para editar el contacto actual.

```
<div>
  <h2>Lista de Contactos</h2>
  <table>
    <thead>
      ...
    </thead>
    <tbody>
      <tr *ngFor="let contacto of contactos">
        <td>{{ contacto.documento }}</td>
        <td>{{ contacto.nombreUsuario }}</td>
        ..
        <td>
          <a [routerLink]="'/contacts/editar/'
+ contacto.id">Editar</a></td>

```

listado-contacto.component.html

```
@Component({
  selector: 'app-listado-contactos',
  standalone: true,

  imports: [CommonModule, RouterLink],

  templateUrl:
    './listado-contactos.component.html',
  styleUrls: ['./listado-contactos.component.css']
})
export class ListadoContactosComponent {

```

listado-contacto.component.ts

# Mostrar el id del contacto en el Editar-Contacto

---

```
<p>Editando el contact : {{ contactoId }}</p>
```

editar-contacto.component.html

```
export class EditarContactoComponent {  
  
  contactoId: number = 0; // Declara una variable para  
  almacenar el id del contacto  
  
  constructor(private activatedRoute: ActivatedRoute)  
  {  
    // Accede al parámetro 'id' de la URL usando el  
    servicio ActivatedRoute  
    this.activatedRoute.params.subscribe(  
      {  
        next: (params: any) => {  
          // Almacena el id del contacto en la variable  
          this.contactoId = params['id'];  
        }  
      }  
    );  
  }  
}
```

editar-contacto.component.ts



# ActivatedRoute

Servicio en Angular que proporciona información sobre la ruta actual en una aplicación.

Contiene detalles sobre la ruta, parámetros, datos de configuración y fragmentos.

## Características:

- **Información sobre la ruta:**
  - Proporciona información detallada sobre la ruta activa en ese momento.
  - Permite acceder a parámetros de ruta, datos estáticos y de resolución, entre otros.
- **Reactivo:**
  - Utiliza observables para proporcionar cambios en los parámetros de ruta y otros datos relacionados.
- **Integración:**
  - Se inyecta en componentes que necesitan acceder a la información de la ruta actual.

## Propiedades Principales:

- **params:**
  - Observable que emite los parámetros de la ruta.
  - Ejemplo: `this.activatedRoute.params.subscribe(params => { console.log(params['id']); });`

# Implementar Angular Routing

## Paso 5

- Implementar el volver
  - Volver desde cualquier componente al home de nuestro sitio



Mct Esteban Calabria

# Ejemplo Volver a la página principal

---

```
<p>crear-contacto works!</p>
```

```
<button (click)="volver()">Volver</button>
```

crear-contacto.component.html

```
export class CrearContactoComponent {  
  constructor(private router:Router){  
  }  
  
  volver(){  
    this.router.navigate([""])  
  }  
}
```

crear-contacto.component.ts

# Router

- Servicio en Angular responsable de gestionar la navegación y la configuración de rutas en una aplicación.
- Facilita la navegación programática, el acceso a la información de la ruta y la configuración de rutas.

## Características:

- **Navegación Programática:**
  - Permite cambiar de ruta mediante código.
  - Útil para redireccionamientos basados en lógica interna.
- **Configuración de Rutas:**
  - Define y gestiona la configuración de rutas en la aplicación.
  - Soporta rutas estáticas y dinámicas, así como rutas con parámetros.
- **Información sobre la Ruta:**
  - Proporciona acceso a la información sobre la ruta activa y el historial de navegación.
- **Eventos de Navegación:**
  - Emite eventos durante el ciclo de vida de la navegación, como inicio, finalización, cancelación y errores.

# Usted debe conocer...

---

- Cómo definir rutas
- Asociar rutas a componentes
- Atributo routerLink
- ActivatedRoute
- Router
- Componente router-outlet



Asegúrese de manejar esos conceptos antes de continuar



Mct Esteban Calabria

Sigamos Trabajando...