

CLASE# 5

Repaso de los temas de Angular

“Sistema de Empleados”

Implementar la consulta de datos de un empleado **por id**, utilizando **ActivatedRoute** para la lectura de parámetros de ruta en la URL y **router.navigate()** correspondiente a la navegación programática dentro del sistema de rutas.

- **Objetivos:**
 - Para realizar la consulta de empleados por id, generar un componente consulta
 - Implementar lógica del componente consulta (ts)
 - Implementar la vista del componente consulta (html).
 - Implementar el uso de la lectura de parámetros en la URL y obtener el id del empleado que viene adjunto en la ruta.
 - Implementar el uso de la navegación programática del sistema de rutas, utilizado para cambiar de componente desde el código Typescript y no desde una plantilla html
- **Tarea personal: En tu proyecto de doctores**
 - Para realizar la consulta de doctores por id, generar un componente consulta
 - Implementar lógica del componente consulta (ts)
 - Implementar la vista del componente consulta (html).
 - Implementar el uso de la lectura de parámetros en la URL y obtener el id del doctor que viene adjunto en la ruta.
 - Implementar el uso de la navegación programática del sistema de rutas, utilizado para cambiar de componente desde el código Typescript y no desde una plantilla html





Al finalizar el avance de hoy esto es lo que vemos en el navegador.

Navbar Listar Agregar Eliminar Actualizar


1794 Search

Lista de Empleados

ID	Nombre	Edad	Sueldo	Antigüedad	Foto
4157	Andrea Márquez	20	\$42,000	3 años	
1794	Lucía Torres	31	\$60,000	6 años	

localhost:4200/consulta/1794

Navbar Listar Agregar Eliminar Actualizar Search



Lucía Torres
ID: 1794
Edad: 31
Sueldo: \$60,000
Antigüedad: 6 años

Lógica de la practica

1 crear componente consulta

```
PS D:\2mispracticAngular\repaso\repasoAngular> ng g c componentes/consulta
CREATE src/app/componentes/consulta/consulta.component.html (24 bytes)
CREATE src/app/componentes/consulta/consulta.component.spec.ts (629 bytes)
CREATE src/app/componentes/consulta/consulta.component.ts (233 bytes)
CREATE src/app/componentes/consulta/consulta.component.css (0 bytes)
PS D:\2mispracticAngular\repaso\repasoAngular>
```



2 en el archivo de rutas, implementar la ruta consulta con un parámetro id

app.routes.ts M X

src > app > TS app.routes.ts > ...

```
1 import { Routes } from '@angular/router';
2 import { ListarEmpleadosComponent } from '../componentes/listar-empleados/listar-empl
3 import { AgregarEmpleadoComponent } from '../componentes/agregar-empleado/agregar-emp
4 import { ConsultaComponent } from '../componentes/consulta/consulta.component';
5
6 export const routes: Routes = [
7   { path: 'listar', component: ListarEmpleadosComponent },
8   { path: 'agregar', component: AgregarEmpleadoComponent },
9   { path: 'consulta/:id', component: ConsultaComponent },
10  { path: '', redirectTo: 'listar', pathMatch: 'full' },
11  { path: '**', redirectTo: 'listar' } // para rutas inválidas
12
13
14 ];
15
```

3 en el ts del componente menú, en el constructor inyectar el uso de la librería Router para tener acceso al método navigate y dirigirnos hacia la vista de otro componente, el método consultar se mandara llamar desde la vista al dar click en el botón search

menu.component.ts M X

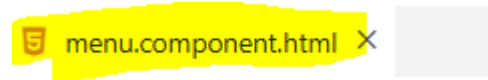
src > app > componentes > menu > menu.component.ts > ...

```
1 import { Component } from '@angular/core';
2 import { Router, RouterLink } from '@angular/router';
3
4 @Component({
5   selector: 'app-menu',
6   imports: [RouterLink],
7   templateUrl: './menu.component.html',
8   styleUrls: ['./menu.component.css']
9 })
10 export class MenuComponent {
11
12   constructor(private router: Router) {}
13
14
15   consultar(id: string) {
16     if (id) {
17       this.router.navigate(['/consulta', id]);
18     }
19   }
20 }
21
```



4 en el html del componente menú debes eliminar la etiqueta form, en su lugar dejar un div, implementar que al dar clic llame al método consulta,

Nota: El input se identifica con #cajaId, #cajaId es importante para obtener con value el id solicitado y limpiar la caja de texto



```
28 <div class="d-flex" role="search">
29   <input #cajaId class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
30   <button class="btn btn-outline-success" (click)="consultar(cajaId.value); cajaId.value=''">Search</button>
31 </div>
32
33 </div>
34 </div>
35 </nav>
```



5 En el ts del componente consulta, en el constructor de la clase, inyectar el uso de la librería ActivatedRoute para obtener el id de la URL e inyectar el uso del servicio para tener acceso al método que consulta por id y obtener los datos del empleado solicitado

Nota:

- En el atributo id guardamos el id que viene como parámetro en la url
- En el atributo empleado, que es un objeto del tipo personal, vamos a guardar los datos del empleados que solicitaste.

```
consulta.component.ts U X
src > app > componentes > consulta > consulta.component.ts > ConsultaComponent
1 import { Component } from '@angular/core';
2 import { Personal } from '../models/personal';
3 import { ActivatedRoute } from '@angular/router';
4 import { EmpleadosService } from '../empleados.service';
5
6 @Component({
7   selector: 'app-consulta',
8   imports: [],
9   templateUrl: './consulta.component.html',
10  styleUrls: ['./consulta.component.css']
11 })
12 export class ConsultaComponent {
13
14
15   id!: number;
16   empleado!: Personal | null;
17
18   constructor(
19     private route: ActivatedRoute,
20     private servicio: EmpleadosService
21   ) {}
22
23   ngOnInit() {
24     this.route.paramMap.subscribe(params => {
25       this.id = Number(params.get('id'));
26       this.empleado = this.servicio.getById(this.id);
27     });
28   }
29 }
30
```

Nota: El ngOnInit() utiliza route que es el objeto del tipo ActivatedRoute, que llama a su propiedad paramMap y este llama al método subscribe para quedar al pendiente de ser notificado de en qué momento se tiene un id en la url, cuando esto se detecta, se extrae el id de la url y se cambia a número porque cuando se extrae el id es una cadena. luego se llama al método getById del servicio mandando el id



6 Incorporar estilos a la etiqueta img de archivo css del componente consulta.

```
consulta.component.css U X
src > app > componentes > consulta > consulta.component.css > ...
1  |  img{
2  |      width: 120px; height: 120px; object-fit: cover;
3  |  }
```

7 En el ts del componente consulta, en el import incluir la librería CommonModule, para que el sueldo lo muestre con comas, para separar los dígitos

```
consulta.component.ts U X
src > app > componentes > consulta > consulta.component.ts > ConsultaComponent
1  |  import { Component } from '@angular/core';
2  |  import { Personal } from '../models/personal';
3  |  import { ActivatedRoute } from '@angular/router';
4  |  import { EmpleadosService } from '../empleados.service';
5  |  import { CommonModule } from '@angular/common';
6  |
7  |  @Component({
8  |      selector: 'app-consulta',
9  |      imports: [CommonModule],
10 |      templateUrl: './consulta.component.html',
11 |      styleUrls: ['./consulta.component.css']
12 |  })
13 |  export class ConsultaComponent {
14 |  }
```



8 En el html, del componente consulta presentar los datos del empleado en un card de Bootstrap

```
consulta.component.html M X
src > app > componentes > consulta > consulta.component.html > div.consulta.d-flex.justify-content-center > div.card.p-3.r
Go to component
1
2 <div class="consulta d-flex justify-content-center">
3
4   @if (empleado !== null) {
5     <div class="card p-3 mt-4 d-flex flex-row " >
6       <img [src]="empleado.foto" alt="Foto del empleado" class="img-thumbnail me-3" >
7       <div>
8         <h5 class="card-title mb-2">{{ empleado.nombre }}</h5>
9         <p class="mb-1"><strong>ID:</strong> {{ empleado.id }}</p>
10        <p class="mb-1"><strong>Edad:</strong> {{ empleado.edad }}</p>
11        <p class="mb-1"><strong>Sueldo:</strong> ${{ empleado.sueldo | number:'1.0-0' }}</p>
12        <p class="mb-0"><strong>Antigüedad:</strong> {{ empleado.antigüedad }} años</p>
13      </div>
14    </div>
15  } @else {
16    <div class="alert alert-warning mt-4 ">
17      No se encontró ningún empleado con el ID {{ id }}.
18    </div>
19  }
20
21 </div>
22
```

9 Levanta servidor y prueba la opción de consultas

