

TEMA III



PRIMEROS PASOS EN EL DESARROLLO DE APPS EN IONIC

3.3 Navegación entre páginas

Existen diversas formas de movernos entre páginas mediante el routing de Angular dentro de ionic:

1.-RouterLink.

Para este método solo basta con agregar la propiedad routerLink="" al botón para que cumpla la función de redirección, este es el método mas simple para realizar un ruteo.

```
<ion-header>
  <ion-toolbar>
    <ion-title>
      Ionic Blank
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div class="ion-padding">
    <ion-button routerLink="/pagina2" routerDirection="forward">Ir
a la página 2</ion-button>
  </div>
</ion-content>
```

Aunque no es obligatorio podemos utilizar como en el ejemplo el parámetro `routerDirection` para indicar la dirección de la animación. Los valores posibles son tres: `forward`, `back` y `root`.

2.- Router de Angular.

En el constructor de la clase, inyectamos una variable de tipo `Router`. Esta clase de angular tiene métodos que nos permiten navegar a otras páginas, por ejemplo, `navigate("página")`

```
<ion-button expand="full" color="success" (click)="routingViaAngular()">
  navegacion con Router
</ion-button>
```

```
constructor(private router: Router){
}
routingViaAngular(){
  this.router.navigate(['/pagina2']);
}
```

3.NavController.

El siguiente método es el routing utilizando el `NavController` incorporado por ionic 3. De forma análoga al caso anterior inyectamos en el constructor una variable de ese tipo.

```
constructor(private navCtrl: NavController){
}
```

Esta clase de ionic incorpora métodos para la navegación y el paso de parámetros. Tenemos tres métodos que podemos utilizar para navegar según la dirección. Estos tres métodos son: `navigateForward`, `navigateBack` y `navigateRoot`.

```
<ion-button expand="full" color="primary" (click)="routingViaNavParams()" >  
  navegacion con navCtrl  
</ion-button>
```

```
constructor(private router: Router, private navCtrl: NavController){  
}  
routingViaAngular(){  
  this.router.navigate(['/pagina2']);  
}  
routingViaNavParams(){  
  this.navCtrl.navigateForward('/pagina2');  
}  
}
```

Pasando parámetros

Muchas veces nos interesa pasar datos al llamar a una página. Por ejemplo, podemos tener un listado de productos y al pulsar en un producto queremos llamar a una página donde se muestre el detalle de ese producto. En las tres posibilidades que hemos visto arriba, los parámetros se pasan vía URL lo que significa que tendríamos que pasar toda la información del producto en la propia url convertido en json. Evidentemente no podemos hacerlo así. Lo más usual es pasar un dato que identifique la información que queremos pasar (por ejemplo el id del producto) y crear un servicio que nos proporcione los datos correspondientes.

1.-RouterLink.

Valor es el dato o la variable que quieres enviar a la pagina2

```
<ion-button [routerLink]="['/pagina2', valor]
```

Valor no va entre {{}}

2. Router angular

Para el router de Angular se necesita enviar el dato dentro del array que pasamos como parámetro en la 2da posición

```
valor= "nombre";  
constructor(private router: Router, private navCtrl: NavController)  
{  
  routingViaAngular(){  
    this.router.navigate(['/pagina2', this.valor]);  
  }  
}
```

3. NavController

Como vamos a cambiar la url puesto que añadimos una información al path, tenemos que cambiar dicho path en el routing.module

```
{  
  path: 'pagina2/:id',  
  loadChildren: () => import('./paginas/pagina2/pagina2.module').then( m => m.Pagina2PageModule)  
},
```

Definimos la nueva URL indicando que pasamos un valor. Para ello en la clase

```
mostrarDetalles(pelicula){  
  this.r.navigateForward(['/pagina2', pelicula.id]);  
}
```

Por último, veamos cómo obtenemos el dato en Pagina2. En las tres opciones necesitamos importar `ActivatedRoute` para poder obtener los parámetros recibidos en la url. Inyectamos en el constructor una variable de este tipo y utilizamos

```
This.r.snapshot.params.get('nombre parámetro')
```