Clase 3 Navegación 2

Paso 1: Instalar Ionic CLI

Si aún no tienes Ionic CLI instalado, debes instalarlo globalmente en tu sistema. Puedes hacerlo utilizando npm:

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

Este comando instala Ionic CLI, que es la herramienta de línea de comandos que utilizarás para crear y administrar proyectos Ionic.

Paso 2: Crear un nuevo proyecto Ionic

Ahora, crea un nuevo proyecto Ionic utilizando la plantilla de pestañas (tabs), que ya tiene enrutamiento básico configurado.

```
ionic start MyApp tabs --type=angular
```

Explicación:

- `MyApp`: Es el nombre del proyecto.
- `tabs`: Es la plantilla que viene con un diseño básico de pestañas.
- `-type=angular`: Especifica que queremos utilizar Angular como framework.

Paso 3: Navegar al directorio del proyecto

Después de crear el proyecto, navega al directorio del proyecto recién creado:

```
cd MyApp
```

Paso 4: Crear la página de Login

Utiliza Ionic CLI para generar una nueva página de Login. Esto creará automáticamente los archivos necesarios y agregará la ruta correspondiente en el archivo de enrutamiento.

```
ionic generate page login
```

Explicación:

- Este comando crea una nueva carpeta `login` dentro de `src/app/` con los siguientes archivos:
  - `login.module.ts`: Módulo para la página de Login.
  - `login.page.ts`: Lógica del componente de la página de Login.
  - `login.page.html`: Estructura HTML de la página de Login.
  - `login.page.scss`: Estilos específicos de la página de Login.

Paso 5: Crear la página de Detalle

Del mismo modo, crea una página de `Detail` (Detalle) para poder navegar hacia ella desde la página de Login.

```
ionic generate page detail
```

Este comando creará una nueva carpeta `detail` dentro de `src/app/`, con los mismos tipos de archivos mencionados anteriormente.

Paso 6: Configurar el enrutamiento

Ahora, necesitamos configurar las rutas en la aplicación para poder navegar entre la página de Login y la página de Detalle. Además, configuraremos la aplicación para que redirija automáticamente a `login` cuando se inicie.

Ruta y Archivo: `src/app/app-routing.module.ts`

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

const routes: Routes = [
  {
    path: '',
```

```
    redirectTo: 'login',  // #### MODIFICACIÓN: Redirige auto
máticamente a la página de Login al iniciar la aplicación
    pathMatch: 'full'
  },
  {
    path: 'login',  // Ruta para la página de Login
    loadChildren: () => import('./login/login.module').then(m
=> m.LoginPageModule)
  },
  {
    path: 'detail',  // Ruta para la página de Detail
    loadChildren: () => import('./detail/detail.module').then
(m => m.DetailPageModule)
  }
];

@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: Preloa
dAllModules })  // Estrategia de pre-carga para mejorar el re
ndimiento
  ],
  exports: [RouterModule]
})
export class AppRoutingModule {}
```

Explicación:

- **Redirección a `login`:** La línea `{ path: '', redirectTo: 'login', pathMatch: 'full' }` asegura que cuando el servidor se ejecute y la aplicación se cargue en el navegador, el usuario sea redirigido automáticamente a la página de `Login`.

Paso 7: Diseñar la página de Login

Vamos a crear un diseño básico para la página de Login, que incluirá un botón para navegar a la página de Detalle.

Ruta y Archivo: `src/app/login/login.pa'´ge.html`

```
<ion-header>
  <ion-toolbar>
    <ion-title>Login</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content class="ion-padding">
  <ion-button expand="full" (click)="navigateToDetail()">Go t
o Detail</ion-button>
</ion-content>
```

Explicación:

- `ion-button`: Este botón permite la navegación a la página de Detalle cuando se hace clic en él.

Paso 8: Implementar la navegación desde la página de Login

En el archivo TypeScript de la página de Login, implementaremos la lógica para navegar a la página de Detalle cuando se haga clic en el botón.

Ruta y Archivo: `src/app/login/login.page.ts`

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-login',
  templateUrl: './login.page.html',
  styleUrls: ['./login.page.scss'],
})
export class LoginPage {

  constructor(private router: Router) {}
```

```
  navigateToDetail() {
    this.router.navigate(['/detail']);  // Navegamos a la pág
ina de detalle
  }
}
```

Explicación:

- `this.router.navigate(['/detail']);`: Este método permite la navegación programática a la ruta `/detail`, que muestra la página de Detalle.

Paso 9: Diseñar la página de Detalle con un botón de retorno

Ahora vamos a crear un diseño básico para la página de Detalle, que incluirá un botón para regresar a la página de Login.

Ruta y Archivo: `src/app/detail/detail.page.html`

```
<ion-header>
  <ion-toolbar>
    <ion-title>Detail</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content class="ion-padding">
  <h2>Welcome to the Detail Page</h2>
  <!-################-->
  <ion-button expand="full" (click)="navigateToLogin()">Go ba
ck to Login</ion-button>  <!-- Botón para regresar a Login --
>
</ion-content>
```

Explicación:

- `ion-button`: Este botón permite la navegación de regreso a la página de Login cuando se hace clic en él.

Paso 10: Implementar la navegación de retorno en la página de Detalle

En el archivo TypeScript de la página de Detalle, implementaremos la lógica para regresar a la página de Login cuando se haga clic en el botón.

Ruta y Archivo: `src/app/detail/detail.page.ts`

```
import { Component } from '@angular/core';
import { Router } from '@angular/router'; //######

@Component({
  selector: 'app-detail',
  templateUrl: './detail.page.html',
  styleUrls: ['./detail.page.scss'],
})
export class DetailPage {

  constructor(private router: Router) {}

  navigateToLogin() {
    this.router.navigate(['/login']);  // Navegamos de regres
o a la página de Login
  }
}
```

Explicación:

- `this.router.navigate(['/login']);`: Este método permite la navegación programática de regreso a la ruta `/login`, que muestra la página de Login.

Paso 11: Ejecutar la aplicación

Finalmente, ejecuta la aplicación para ver el resultado.

```
ionic serve
```