

Tipo : Guía de laboratorio
Capítulo : Angular 8
Duración : 60 minutos

I. OBJETIVO

Usar los conceptos vertidos en el presente capítulo, agregando rutas a la aplicación para mostrar distintas páginas.

II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Editor Visual Studio Code
- NodeJS

III. EJECUCIÓN DEL LABORATORIO

- Ejercicio Nº 2.4: Agregar rutas y configurar Lazy Loading

Agregue rutas y configurar Lazy loading

1. Cree los componentes **SignIn**, **ProductEdit** y **ProductNew**.

```
# --skip-tests=true: No crear el archivo .spec
ng g c login/sign-in --skip-tests=true -m login
ng g c admin/product-edit --skip-tests=true -m admin
ng g c admin/product-new --skip-tests=true -m admin
```

2. Cree un archivo routing por cada módulo y registre las rutas correspondientes:

[src/app/admin/admin.module.ts](#)

```
imports: [
  CommonModule,
  AdminRoutingModule
]
```

src/app/login/login.module.ts

```
imports: [  
  CommonModule,  
  LoginRoutingModule  
]
```

3. Registre las siguientes rutas en el módulo principal:

src/app/app.module.ts

```
const routes: Routes = [  
  {  
    path: 'login',  
    loadChildren: () => import('./login/login.module').then(m  
=> m.LoginModule)  
  },  
  {  
    path: 'admin',  
    loadChildren: () => import('./admin/admin.module').then(m  
=> m.AdminModule )  
  }  
]
```

IV. EVALUACIÓN

1. ¿Para qué sirven las rutas?

Las rutas permiten crear un **SPA** (Single Page Application) completo, en donde se puede tener distintos componentes que tendrán el rol de página.

2. ¿Qué beneficios nos da implementar Lazy Loading en el proyecto?

Nos permite cargar los módulos a demanda, una vez que naveguemos en las rutas correspondientes a un módulo.