

## Guía de laboratorio – Proyecto final: Parte II

### PASO 1: Creación del proyecto

Ejecutar el siguiente comando en la terminal

```
ng new proyectofinalng
```

Y crear el proyecto con Angular Routing

### PASO 2: Integración de librerías

- Integrar Angular Material con el siguiente comando

```
ng add @angular/material
```

- Integrar Angular Flex-Layout

```
npm i -s @angular/flex-layout @angular/cdk
```

- Integrar Angular infinite Scroll

```
npm i ngx-infinite-scroll
```

### PASO 3: Archivos base

- Copiar el archivo banner.png a la carpeta assets del proyecto.
- Copiar el archivo styles.css a la carpeta raíz del proyecto.

### PASO 4: Variables de entorno

Configurar las variables de entorno para modo desarrollo agregar dos propiedades al archivo environment.ts de la carpeta environments:

```
apiBase: 'http://localhost:8080/api'
```

```
loginUrl: 'http://localhost:8081/login'
```

### PASO 5: Actualizar el index.html

- Cambiar el idioma del archivo index.html.
- Agregar el siguiente fragment HTML dentro de la etiqueta head del archivo index.html.  
<link rel="preconnect" href="https://fonts.gstatic.com">  
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=swap" rel="stylesheet">  
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
- Agregar la clase mat-typography a la etiqueta body del archivo index.html.

### PASO 6:

Importar los módulos BrowserModule y HttpClientModule en el archivo app.module.ts

### PASO 7: Implementación del panel administrativo

- Crear el módulo admin e importar los módulos: CommonModule, MaterialModule, ReactiveFormsModule, FlexLayoutModule.
- Crear un componente llamado 'layout' que va servir como la plantilla de todas las páginas del panel administrativo.
- Crear un archivo 'admin-routing.module.ts' para las mapear las rutas del panel administrativo. [TIP] Usar como plantilla el archivo app-routing.module.ts. Luego importarlo en el módulo admin.
- Agregar la ruta 'admin' y usar lazy loading para cargar el módulo admin en app-routing.module.ts
- Crear la interface interfaces/libro usando quicktype.io
- Crear los servicios admin/services/asset y admin/services/libro
- Crear los componentes admin/pages/lista-libros, admin/pages/nuevo-libro y admin/pages/editar-libro.