

Clase 7 Angular Material

Paso 1: Crear un nuevo proyecto Ionic

Crea un nuevo proyecto utilizando la plantilla en blanco. Esto generará la estructura básica de un proyecto **ionic** con **Angular** como framework:

```
ionic start miAppSimple blank --type=angular
```

- `miAppSimple` : Es el nombre de tu proyecto.
- `blank` : Es una plantilla básica sin elementos adicionales.
- `-type=angular` : Especifica que el proyecto usará Angular.

Paso 2: Navegar al directorio del proyecto

Después de crear el proyecto, muévete al directorio del proyecto para comenzar a trabajar:

```
cd miAppSimple
```

Paso 3: Instalar Angular Material

Instala **Angular Material**, **CDK**, y **Animations**. Angular Material te ofrece una serie de componentes preconstruidos que siguen las guías de diseño de Material Design.

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

Sigue las indicaciones, selecciona un tema (te recomiendo **indigo-pink**) y habilita las animaciones.

Paso 4: Instalar módulos adicionales de Angular Material

Si ves que faltan dependencias, instálalas manualmente:

```
npm install @angular/material @angular/cdk @angular/animations
```

Paso 5: Configurar el archivo `global.scss`

Abre el archivo `src/global.scss` y añade la importación del tema de Angular Material. Esto garantiza que se apliquen los estilos de **Material Design** en la aplicación.

```
@import '~@angular/material/prebuilt-themes/indigo-pink.css';
```

Paso 6: Configurar el archivo `index.html`

Para que los estilos de fuente y los iconos de **Material Design** funcionen correctamente, agrega estas líneas en el archivo `src/index.html` dentro de la etiqueta `<head>`:

```
<!-- Fuente Roboto, utilizada por Angular Material -->
<link href="<https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap>" rel="stylesheet">
<!-- Iconos de Material Design -->
<link href="<https://fonts.googleapis.com/icon?family=Material+Icons>" rel="stylesheet">
```

Paso 7: Configurar `app.module.ts`

Abre el archivo `src/app/app.module.ts` y asegúrate de importar los módulos necesarios de **Angular Material** y **BrowserAnimationsModule**, que es obligatorio para que las animaciones de Angular Material funcionen correctamente.

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { RouteReuseStrategy } from '@angular/router';
```

```

import { AppComponent } from './app.component';
import { AppRoutingModuleModule } from './app-routing.module';

// Importar módulos de Angular Material
import { MatButtonModule } from '@angular/material/button';
// Módulo de botones de Material
import { MatCardModule } from '@angular/material/card';
// Módulo de tarjetas (card) de Material
import { BrowserAnimationsModule } from '@angular/platform-browser/animations'; // Animaciones necesarias

@NgModule({
  declarations: [AppComponent], // Declaramos el componente principal de la app
  imports: [
    BrowserModule, // Navegador
    IonicModule.forRoot(), // Ionic
    AppRoutingModuleModule, // Manejo de rutas
    MatButtonModule, // Importamos el módulo de botones de Material
    MatCardModule, // Importamos el módulo de tarjetas de Material
    BrowserAnimationsModule // Importamos el módulo de animaciones de Angular
  ],
  providers: [{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy }], // Configuramos el enrutamiento
  bootstrap: [AppComponent], // Inicializamos la aplicación con AppComponent
})
export class AppModule {}

```

Paso 8: Crear el contenido de la aplicación

Edita el archivo `src/app/app.component.html` para construir una interfaz simple que incluya un botón y una tarjeta de **Angular Material**. Estos elementos se mostrarán

dentro de la interfaz de Ionic.

```
<ion-app>
  <!-- ion-app es el contenedor principal de la aplicación -->
  </ion-app>

  <ion-header>
    <!-- ion-header es el encabezado de la aplicación -->
    <ion-toolbar>
      <!-- ion-toolbar proporciona un marco flexible para poner títulos, botones, etc. -->
      <ion-title>
        Aplicación Simple con Angular Material
      </ion-title>
    </ion-toolbar>
  </ion-header>

  <ion-content class="ion-padding">
    <!-- ion-content es el contenedor principal para el contenido desplazable de la app -->

    <!-- Botón de Angular Material -->
    <button mat-raised-button color="primary" (click)="mostrarMensaje()">
      <!-- mat-raised-button es un botón elevado con sombra -->
      Presionar
    </button>

    <!-- Tarjeta de Angular Material -->
    <mat-card class="ion-margin-top">
      <!-- mat-card es un contenedor de tarjeta estilizado con Material Design -->
      <mat-card-header>
        <!-- mat-card-header es el encabezado de la tarjeta -->
```

```

        <mat-card-title>Tarjeta de Información</mat-card-titl
e>
    </mat-card-header>
    <mat-card-content>
        <!-- mat-card-content contiene el contenido de la tar
jeta -->
        <p>{{ mensaje }}</p> <!-- Aquí mostramos el mensaje q
ue cambia con el botón -->
    </mat-card-content>
</mat-card>
</ion-content>
</ion-app>

```

Paso 9: Lógica del botón en `app.component.ts`

Ahora edita el archivo `src/app/app.component.ts` para manejar la lógica del botón. El botón actualizará el valor de la variable `mensaje`, que se mostrará en la tarjeta.

```

import { Component } from '@angular/core';

@Component({
  selector: 'app-root', // Nombre del selector del component
  e raíz
  templateUrl: 'app.component.html', // El archivo de plantil
  la que contiene el HTML
  styleUrls: ['app.component.scss'], // Archivo de estilos
})
export class AppComponent {
  // Variable que contiene el mensaje a mostrar
  mensaje: string = 'Presiona el botón para ver un mensaje';

  // Método que cambia el mensaje al presionar el botón
  mostrarMensaje() {
    this.mensaje = '¡Hola! Este es un mensaje desde el botó
n.';
    // Al hacer clic en el botón, se actualiza el valor de "m

```

```
    ensaje"  
  }  
}
```

Paso 10: Ejecutar la aplicación

```
ionic serve
```