

Clase 8 Animaciones con Ionic

Paso 1: Creación del Proyecto y Configuración Inicial

1.1. Instalar Ionic CLI

Si aún no tienes Ionic instalado, abre tu terminal y ejecuta el siguiente comando:

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

1.2. Crear un Nuevo Proyecto Ionic

Ahora vamos a crear el proyecto con el siguiente comando:

```
ionic start miAppMovilConAnimaciones plantilla --type=angular
```

- `miAppMovilConAnimaciones`: Es el nombre del proyecto.
- `plantilla`: Se refiere a una plantilla básica.
- `--type=angular`: Indica que queremos usar Angular como framework.

Después de crear el proyecto, navega a su directorio con:

```
cd miAppMovilConAnimaciones
```

Paso 2: Configuración del Módulo Principal

2.1. Configuración del `app.module.ts`

1. **Abre** el archivo `src/app/app.module.ts`.
2. **Agrega** el siguiente código:

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

```

import { IonicModule, IonicRouteStrategy, AnimationController
} from '@ionic/angular'; //////////////// // ### Importamos el con
trolador de animaciones
import { AppComponent } from './app.component';
import { AppRoutingModuleModule } from './app-routing.module';

@NgModule({
  declarations: [AppComponent], // ### Componente principal
  imports: [
    BrowserModule, // ### Módulo para funcion
amiento en el navegador
    IonicModule.forRoot(), // ### Configuración de I
onic con Angular
    AppRoutingModuleModule // ### Módulo de enrutamie
nto
  ],
  providers: [
    { provide: RouteReuseStrategy, useClass: IonicRouteStrate
gy }, // ### Estrategia de reutilización de rutas
    AnimationController // ### Proveedor del controlador de
animaciones////////////////////
  ],
  bootstrap: [AppComponent] // ### Punto de arranque de la a
plicación
})
export class AppModule { }

```

Paso 3: Implementación de las Animaciones

3.1. Lógica de Animaciones en `app.component.ts`

1. **Abre** el archivo `src/app/app.component.ts`.
2. **Añade** la lógica para las cuatro animaciones:

```

import { Component, OnInit } from '@angular/core'; //////////
, OnInit
import { AnimationController } from '@ionic/angular';////////

@Component({
  selector: 'app-root',
  templateUrl: 'app.component.html',
  styleUrls: ['app.component.scss'],
})
export class AppComponent implements OnInit {    /////impleme
nts OnInit

  constructor(private animacionCtrl: AnimationController)
  {}////////

  ngOnInit() {////////////////////////////////////
    this.animarIcono(); // ### Inicia la rotación del ícono a
l cargar la página
  }

  // ### Método para animar el botón
  animarBoton() {
    const boton = document.querySelector('.boton-animado');
    // ### Selecciona el botón por su clase

    if (boton) {
      const animacion = this.animacionCtrl.create()
        .addElement(boton) // ### Aplica la animación al bot
ón
        .duration(500) // ### Duración de la animación (500
ms)
        .fromTo('transform', 'scale(1)', 'scale(1.2)') // ##
# Aumenta el tamaño del botón
        .fromTo('background-color', '#3880ff', '#ff5722');
    // ### Cambia el color de azul a naranja

```

```

        animacion.play(); // ### Ejecuta la animación
    } else {
        console.error('No se pudo encontrar el botón con la clase .boton-animado');
    }
}

// ### Método para rotar el ícono
animarIcono() {
    const icono = document.querySelector('.icono-rotativo');
    // ### Selecciona el ícono por su clase

    if (icono) {
        const animacion = this.animacionCtrl.create()
            .addElement(icono) // ### Aplica la animación al ícono
            .duration(2000) // ### Duración de la rotación (2 segundos)
            .iterations(Infinity) // ### Repite indefinidamente
            .fromTo('transform', 'rotate(0deg)', 'rotate(360deg)'); // ### Rota el ícono 360 grados

        animacion.play(); // ### Ejecuta la animación
    } else {
        console.error('No se pudo encontrar el ícono con la clase .icono-rotativo');
    }
}

// ### Método para deslizar la tarjeta
animarTarjeta() {
    const tarjeta = document.querySelector('.tarjeta-animada'); // ### Selecciona la tarjeta por su clase

    if (tarjeta) {

```

```

        const animacion = this.animacionCtrl.create()
            .addElement(tarjeta) // ### Aplica la animación a la
tarjeta
            .duration(1000) // ### Duración de la animación (1 s
egundo)
            .fromTo('transform', 'translateX(-100%)', 'translateX
(0%)') // ### Desliza la tarjeta desde la izquierda
            .fromTo('opacity', '0', '1'); // ### Cambia la opaci
dad de 0 a 1

        animacion.play(); // ### Ejecuta la animación
    } else {
        console.error('No se pudo encontrar la tarjeta con la c
lase .tarjeta-animada');
    }
}

}

```

Paso 4: Estructura Visual del Componente

4.1. Modificar `app.component.html`

1. **Abre** el archivo `src/app/app.component.html`.
2. **Agrega** el siguiente código:

```

<ion-app>
  <ion-header>
    <ion-toolbar>
      <ion-title>
        Animaciones en Aplicaciones Móviles
      </ion-title>
    </ion-toolbar>
  </ion-header>

```

```

<ion-content class="ion-padding">

  <!-- ### Ícono que rota indefinidamente -->
  <ion-icon name="sync" class="icono-rotativo"></ion-icon>
  ###

  <!-- Botón que será animado al presionarlo -->
  <ion-button class="boton-animado" (click)="animarBoton
  ()">Presióname</ion-button>

  <!-- Botón para mostrar la tarjeta animada -->
  <ion-button color="secondary" (click)="animarTarjeta()">M
  ostrar Tarjeta</ion-button>

  <!-- ### Tarjeta que se desliza al presionar el botón -->
  <ion-card class="tarjeta-animada">
    <ion-card-header>
      <ion-card-title>
        Tarjeta Animada
      </ion-card-title>
    </ion-card-header>
    <ion-card-content>
      Esta tarjeta se desliza desde la izquierda cuando pre
      sionas el botón "Mostrar Tarjeta".
    </ion-card-content>
  </ion-card>

</ion-content>
</ion-app>

```

Paso 5: Estilos de los Elementos Animados

5.1. Modificar `app.component.scss`

1. **Abre** el archivo `src/app/app.component.scss`.
2. **Agrega** los siguientes estilos:

```
/* ### Estilos para el botón animado */
.boton-animado {
  width: 150px; /* Ancho del botón */
  height: 50px; /* Altura del botón */
  background-color: #3880ff; /* Color inicial de fondo (azul) */
  display: block; /* El botón es un bloque */
  margin: 20px auto; /* Centramos el botón con margen superior e inferior */
  text-align: center; /* Centramos el texto dentro del botón */
  transition: background-color 0.3s ease; /* Transición suave para el cambio de color */
}

/* ### Estilos para el ícono rotativo */
.ícono-rotativo {
  font-size: 50px; /* Tamaño del ícono */
  color: #ff5722; /* Color del ícono (naranja) */
  display: block; /* El ícono es un bloque */
  margin: 20px auto; /* Centramos el ícono */
}

/* ### Estilos para la tarjeta animada */
.tarjeta-animada {
  width: 90%; /* Ancho de la tarjeta */
  max-width: 400px; /* Ancho máximo */
  margin: 20px auto; /* Centramos la tarjeta */
  opacity: 0; /* Inicialmente invisible */
}
```

```
transform: translateX(-100%); /* Posición inicial fuera de
la pantalla */
}
```

Paso 6: Ejecutar la Aplicación

Para probar la aplicación y ver las animaciones en acción:

1. **Ejecuta** el servidor de desarrollo:

```
ionic serve
```

1. Abre el navegador en `http://localhost:8100` y podrás interactuar con las animaciones.

Resumen de las Animaciones Implementadas

1. Animación de Botón:

- **Descripción:** El botón crece en tamaño y cambia de color al presionarlo.
- **Función:** `animarBoton()`.

2. Animación de Ícono:

- **Descripción:** Un ícono gira continuamente para simular una acción en progreso.
- **Función:** `animarIcono()`.

3. Animación de Tarjeta:

- **Descripción:** Una tarjeta se desliza desde la izquierda cuando se presiona un botón.
- **Función:** `animarTarjeta()`.