



BIENVENIDOS  
AL CURSO:

## **Angular 9: Proyecto de Integración**

SESIÓN 01





01

Espacio de trabajo (workspace)

---

02

Estructura del proyecto (scaffolding)

---



En angular existe la posibilidad de crear espacios de trabajo, lo cual facilita el desarrollo de múltiples aplicaciones y es muy usado en el mundo laboral.

```
ng new my-workspace --createApplication="false"
```

## ■ Espacio de trabajo (workspace)



Además no solo podemos generar nuestros propios proyectos, sino también librerías, de esta manera podemos distribuir mejor nuestro código según sea la necesidad.

```
ng generate application my-first-app
```

```
ng generate library my-first-library
```

## ■ Espacio de trabajo (workspace)



Angular nos da la libertad de ordenar nuestro código en una estructura de carpetas personalizada, pero esto puede beneficiarnos o perjudicarnos de acuerdo al tamaño del proyecto, si no tenemos una lógica al crear el árbol de carpetas.

#### Estructura del Espacio de trabajo

- Apps: carpeta de aplicaciones
- Libs: carpeta de librerías propias

```
<workspace>
— apps/
  — <app-1>/
  — <app-2>/
— libs/
  — <app-1>-commons/
  — <app-2>-commons/
```

## ■ Estructura del proyecto (scaffolding)



Estructura de una aplicación:

- Features: carpeta de módulos que representan las características del negocio.

```
— <app-1>/  
  — src/  
    — app/  
      — features/  
        — feature-1/  
        — feature-2/  
      — app-routing.module.ts  
      — app.component.html  
      — app.component.scss  
      — app.component.ts  
      — app.module.ts  
    — assets/  
    — environments/  
    — main.ts
```



Estructura de una característica (feature):

- Views: carpeta de componentes contenedores
- Models: Modelos
- Commons: Carpeta que agrupa los componentes, directivas, pipes, etc.

```
<feature-1>/  
— views/  
— models/  
— commons/  
  — components/  
  — directives/  
  — pipes/  
  — services/  
  — <feature-1>-commons.module.ts  
— <feature-1>-routing.module.ts  
— <feature-1>.component.html  
— <feature-1>.component.scss  
— <feature-1>.component.ts  
— <feature-1>.module.ts
```



Estructura de una librería:

- lib: Carpeta que agrupa los componentes, directivas, pipes, etc.
- Public-api.ts: archivo que exporta todo el código que será publico

```
<app-1>-commons/  
— src/  
  — lib/  
    — components/  
    — directives/  
    — pipes/  
    — services/  
    — libs.module.ts  
  — public-api.ts
```

## ■ Estructura del proyecto (scaffolding)





# GRACIAS

**POR SU PREFERENCIA**