

## PRÁCTICAS DE LABORATORIO

No. De Control	Nombre	Fecha de realización

CARRERA	PLAN DE ESTUDIO	CLAVE DE ASIGNATURA
Ingeniería en Sistemas Computacionales		TMD-1902
NOMBRE DE LA ASIGNATURA	PRACTICA No.	NOMBRE DE LA PRÁCTICA
Desarrollo de Aplicaciones Web	1	Primeros pasos en Angular

### 2 | OBJETIVO (COMPETENCIA)

Proporcionar una introducción práctica a Angular, donde los estudiantes aprenderán a configurar un proyecto básico utilizando Angular/CLI, crear y gestionar componentes, manipular datos dinámicos, y aplicar controles de flujo. Además, los estudiantes se familiarizarán con la comunicación entre componentes y la implementación de rutas en una aplicación de una sola página (SPA), desarrollando una base sólida para la construcción de aplicaciones dinámicas y modulares con Angular.

### 3 | INTRODUCCIÓN:

Angular es un **framework de desarrollo web** de código abierto mantenido por Google, diseñado para construir **aplicaciones de una sola página (SPA)** con una estructura modular y eficiente. Está escrito en **TypeScript**, un superset de JavaScript que añade tipado estático y otras características útiles para grandes proyectos.

#### Principales características de Angular:

- Componentes:** Angular se basa en una arquitectura de componentes. Cada parte de la interfaz de usuario se puede dividir en componentes reutilizables, que encapsulan la lógica y el diseño.
- Binding de datos bidireccional:** Permite que los datos del modelo se sincronicen automáticamente con la vista y viceversa, lo que simplifica la gestión del estado de la aplicación.
- Inyección de dependencias:** Angular facilita la gestión de servicios y dependencias a través de su sistema de inyección, lo que mejora la modularidad y testabilidad.
- Ruteo:** Angular permite crear aplicaciones SPA, donde el ruteo permite navegar entre diferentes vistas o componentes sin recargar la página.
- Directivas:** Angular proporciona directivas que te permiten manipular el DOM, como \*ngIf para control condicional y \*ngFor para loops.
- Formularios reactivos:** Angular ofrece herramientas avanzadas para la gestión de formularios, con validación y manipulación sencilla.
- HTTP Client:** Facilita la comunicación con APIs externas para enviar o recibir datos.

#### Usos comunes:

Angular es ideal para crear aplicaciones web dinámicas, como:

- **Sistemas de gestión (CRM, ERP)**
- **Dashboards interactivos**

- Aplicaciones móviles híbridas (con Angular Ionic)

## 4 MATERIALES Y EQUIPO

### Requerimientos de hardware:

1. **Procesador:** Intel Core i3 o equivalente.
2. **Memoria RAM:** 4 GB mínimo (recomendado 8 GB para un rendimiento más fluido).
3. **Almacenamiento:** Al menos 10 GB de espacio libre en disco.
4. **Resolución de pantalla:** 1366x768 píxeles (recomendado 1920x1080 para una mejor experiencia de desarrollo).
5. **Conexión a internet:** Para instalar dependencias y paquetes desde npm.

### Requerimientos de software:

1. **Sistema Operativo:**
  - Windows 10 o superior (64 bits).
  - macOS 10.12 o superior.
  - Distribuciones Linux modernas (Ubuntu, Fedora, etc.).
2. **Node.js:** Angular requiere Node.js para instalar dependencias y ejecutar comandos de Angular CLI.
  - Versión recomendada: Node.js 14.x o superior.
3. **npm (Node Package Manager):** Se instala automáticamente con Node.js y se usa para manejar paquetes y dependencias.
  - Versión recomendada: npm 6.x o superior.
4. **Angular CLI:** Para generar, desarrollar y servir proyectos Angular.
  - Se puede instalar ejecutando el siguiente comando:
    - npm install -g @angular/cli
5. **Editor de código:**
  - **Visual Studio Code** (recomendado) o cualquier otro editor de texto compatible con JavaScript/TypeScript.
  - Se sugiere instalar la extensión oficial de Angular para VS Code para facilitar el desarrollo.
6. **Navegador web:**
  - Google Chrome, Mozilla Firefox, Microsoft Edge o cualquier navegador moderno compatible con herramientas de desarrollo para visualizar la aplicación.

## 5 PROCEDIMIENTOS (DESCRIPCIÓN)

1. Crea un proyecto angular en disco, nómbralo descriptivamente considerando que estarás construyendo un FrontEnd
2. Edita la plantilla por default del proyecto Angular a manera de entender su funcionamiento, identifica los archivos que integran el proyecto y describelos.
3. Describe la anatomía de un componente angular del ejemplo que acabas de crear.
4. Implementa de acuerdo a lo visto en clase lo siguiente:
  - a. Nuevos componentes
  - b. Enlace de datos a nivel de propiedad, atributo, evento y doble enlace
  - c. Añade controles de flujo (if, else, for)
  - d. Establece métodos de comunicación entre componentes
  - e. Define rutas de navegación entre componentes.

**6 | SUGERENCIAS DIDÁCTICAS**

1. <https://angular.dev/>
2. <https://nodejs.org/en>

**7 | DOCUMENTACIÓN DEL PROCEDIMIENTOS**

“DOCUMENTA AQUÍ TUS RESULTADOS, UTILIZA IMÁGENES ACOMPAÑADAS DE LA DESCRIPCIÓN DE LOS PASOS QUE REALIZASTE”