**TRABAJO PRÁCTICO N° 10**

**INTRODUCCIÓN A COMPONENTES EN ANGULAR**

1. **Conceptos básicos** 
   1. **Definición de componente**

a. Un componente es una clase a la que se le añade un decorador y que permite crear nuevas etiquetas HTML mediante las cuales podemos añadir funcionalidad a nuestras páginas controlando determinadas zonas de pantalla.

¿COMO ESTA COMPUESTO? Se compone de tres archivos principales:

* **HTML:** Define la estructura visual del componente (botones, listas, formularios, etc.)
* **TypeScript:** Contiene la lógica del componente. Aquí se definen las propiedades, los métodos y las interacciones con otros componentes.
* **CSS:** Define el aspecto visual del componente. Aquí se aplican los estilos (colores, fuentes, tamaños, etc.)

¿Son todos los archivos necesarios para el funcionamiento de un componente?

Sí, los tres archivos son esenciales para un componente completo:

* Sin el HTML, no tendríamos la parte visual del componente.
* Sin la clase, no tendríamos la lógica que controla el comportamiento del componente.
* Sin el estilo, el componente carecería de un aspecto visual atractivo.

b. La instrucción `import` en un archivo de componente Angular sirve para “incorporar módulos, componentes, servicios u otras entidades” que serán utilizadas dentro de ese componente. En esencia, actúa como una forma de "traer" código de otros lugares de tu proyecto para que puedas utilizarlo en tu componente actual.

* 1. **Estructura del componente**
* ¿Qué es un selector?

Un selector en un componente de Angular es como un identificador único que le dice a Angular dónde insertar una instancia de ese componente en el template HTML. Selecciona componentes para insertarlos en la estructura de la aplicación.

¿Qué especificamos en él?

En el selector, generalmente se especifica un nombre que sea descriptivo y único para el componente. Este nombre puede ser cualquier cadena de texto válida, pero se recomienda utilizar una convención de nomenclatura consistente para mejorar la legibilidad del código.

* ¿Qué significa la propiedad templateUrl en un componente?

La propiedad `templateUrl` en un componente de Angular especifica la ubicación del archivo HTML que contiene la plantilla del componente. Esta plantilla define la estructura visual del componente y cómo se mostrará en la aplicación.

¿Por qué está entre corchetes y qué tipo de valores se colocan allí separados por comas?

**Corchetes:** Indican que `templateUrl` es un atributo de un decorador en TypeScript.

**Valores:**

**\*** Tipo de valor: `templateUrl` espera una cadena de texto que representa la ruta relativa al archivo TypeScript del componente donde se encuentra la plantilla HTML.

**\*** Separados por comas: No se colocan valores separados por comas en `templateUrl`. Es un solo valor de cadena que indica la ubicación del archivo de la plantilla.

1. **Integración de componentes**

La integración de componentes en Angular es un proceso fundamental para construir aplicaciones complejas y modulares. Pasos clave para lograr una correcta integración:

1. Crear componentes: Generar los archivos necesarios.

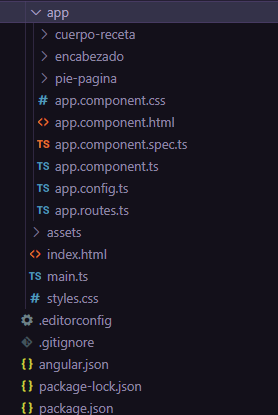
2. Declarar en el módulo: Agregar los componentes al arreglo `declarations`.

3. Utilizar en plantillas: Insertar el componente utilizando su selector.

4. Establecer relaciones: Definir jerarquías padre-hijo y utilizar `@Input` y `@Output`.

5. Configurar rutas: Si es necesario, utilizar el Angular Router.

6. Compartir datos: Utilizar servicios para compartir datos entre componentes.

1. **Comandos para crear componentes**