**1. Conceptos básicos**

**1.1 Definición de componente**

Un **componente** en Angular es una de las piezas fundamentales de la arquitectura de una aplicación Angular. Cada componente es una clase que se asocia a un template HTML y se encarga de manejar una parte específica de la interfaz de usuario. Un componente está compuesto por:

* **Clase**: Define la lógica del componente y se encuentra en un archivo TypeScript (por ejemplo, nombre-componente.component.ts).
* **Template**: Es el HTML que define la vista del componente, especificado mediante la propiedad templateUrl o template en el decorador @Component.
* **Estilos**: Se pueden definir estilos específicos para el componente mediante la propiedad styleUrls o styles en el decorador @Component.
* **Metadatos**: Proporcionados por el decorador @Component, que indican cómo se debe procesar, instanciar y utilizar el componente.

No todos los archivos son necesarios para el funcionamiento de un componente, pero generalmente se recomienda tener al menos la clase y el template.

**1.2 Función de la sentencia import**

La sentencia import al inicio de cada archivo de componente en Angular se utiliza para incluir otras clases, módulos o funciones que el componente necesita para funcionar. Por ejemplo, se pueden importar módulos de Angular, como Component de @angular/core, que es esencial para definir un componente. Esto permite que el componente tenga acceso a las funcionalidades y características que se encuentran en otros archivos o bibliotecas.

**2. Integración de componentes**

La integración de diferentes componentes en un proyecto de Angular se logra a través de los siguientes pasos:

1. **Crear los componentes**: Utilizar los comandos de Angular CLI para generar nuevos componentes.
2. **Declarar los componentes**: Asegurarse de que los nuevos componentes estén declarados en el módulo correspondiente (generalmente en app.module.ts), dentro de la sección declarations.
3. **Usar los selectores**: Cada componente tiene un selector que se utiliza como una etiqueta HTML en el template de otros componentes. Por ejemplo, si el selector de un componente es app-mi-componente, se puede usar en el template de otro componente como <app-mi-componente></app-mi-componente>.
4. **Comunicación entre componentes**: Si es necesario, establecer la comunicación entre componentes mediante @Input() y @Output() para pasar datos y emitir eventos.

**3. Comandos para crear componentes**

Para crear nuevos componentes en Angular, se utilizan los siguientes comandos en la terminal:

ng generate component <nombreComponente> o su abreviación ng g c <nombreComponente>. Por ejemplo, para crear un componente llamado PrimerComponente, se usaría:

ng generate component PrimerComponente

o

ng g c PrimerComponente

Es importante que cada componente tenga un nombre descriptivo que refleje su propósito en la aplicación.