**Taller temas sobre Angular**

**Clase de Taller de Aplicaciones Móviles - Grupo 1**

**Estudiante:** Diana Sofia Benavides Monroy.  **Código:** 55823004

1. **Componentes básicos**
   * **Qué es un componente y su estructura (decorador @Component, selector, template, style).**

Un componente en Angular es un bloque de construcción principal que encapsula una parte de la interfaz de usuario, combinando lógica (TypeScript), presentación (HTML) y estilos (CSS) para crear elementos interactivos y reutilizables en una aplicación. Aunque puedes crear manualmente cada uno de los archivos necesarios de un componente, lo más recomendable es utilizar el CLI (Command Line Interface) de Angular en la terminal:

ng generate component nombre-del-componente

El componente Angular está inspirado en los patrones MVC (Modelo-Vista-Controlador).

“La estructura básica es:

*// app/navbar/navbar.component.ts*

import { Component } from "@angular/core";

@Component({

selector: "app-navbar",

templateUrl: "./navbar.component.html",

styleUrls: ["./navbar.component.css"],

})

export class NavbarComponent {

constructor() {}

}

Lo primero que se hace es importar Component de **@angular/core**. Esto sirve para poder invocar justo debajo **@Component**, una especie de método que recibe un objeto con la configuración del componente. Dentro de esa configuración hay tres propiedades:

* **selector:** El selector es el nombre que va a tener la etiqueta HTML que sirve para poder usar este componente, para este ejemplo del navbar será <app-navbar></app-navbar>, es decir, desde el HTML de cualquier otro componente poniendo esa etiqueta se pintará el navbar.
* **templateUrl:** La ruta al fichero .html de ese componente para crear la vista.
* **styleUrl:** La ruta al fichero .html de ese componente para crear la vista.

Lo normal es que estás rutas empiecen por ./ (carpeta actual) ya que normalmente los ficheros html y css se encuentran en la misma carpeta que el fichero .ts.

Por último hacemos export class con el nombre del componente (sin guiones aquí). Dentro del export se crea el método constructor (vacío por el momento).” (Angular - Cómo Crear Componentes y Su Estructura Principal, 2020)

* + **Diferencia entre un componente de página (vista completa) y uno reutilizable (ej. ProductCard).**

“Los Web Components se fundamentan en tres tecnologías principales que permiten la creación de componentes robustos, encapsulados y reutilizables”

**2. Standalone Components (Angular 15+)**

* + **Qué significa que no hay módulos (NgModule).**
  + **Uso de la propiedad standalone: true.**

“La propiedad **standalone: true** en Angular marca un componente, directiva o pipe como autónomo, eliminando la necesidad de declararlo en un NgModule.”(*Google Gemini*, n.d.)

* + **Importación de otros componentes/módulos dentro de imports.**

La sentencia **import** se usa para importar funciones que han sido exportadas desde un módulo externo.

Algunos ejemplos de como realizar este proceso:

“

| import { nombre } from "./file.js" | Importa el elemento nombre de file.js. |
| --- | --- |
| import { nombre as newName } from "./file.js" | Importa el elemento nombre de file.js como newName. |
| import { n1, n2... } from "./file.js" | Importa los elementos indicados desde file.js. |
| import nombre from "./file.js" | Importa el elemento **por defecto** de file.js como nombre. |
| import \* as name from "./file.js" | Importa todos los elementos de file.js en el objeto name. |
| import "./file.js" | Ejecuta el código de file.js. No importa ningún elemento. |
| import { name } from "https://web.com/file.js" |  |

”(*Importar Módulos - Javascript En Español*, n.d.)

**3. Data Binding**

* + **Interpolación** ({{ }})

Es uno de los tipos de Data Binding que existen, este utiliza doble llave ({{ }}) ara enlazar datos del componente a la vista.

* + **Property Binding** ([prop]="value")

Usa corchetes ([ ]) para enlazar propiedades del componente a propiedades del DOM.

* + **Event Binding** ((event)="handler()")

Usa paréntesis (( )) para enlazar eventos del DOM a métodos del componente.

**Referencias**

Angular - Cómo crear componentes y su estructura principal. (2020, September 9). <https://codingpotions.com/angular-componentes/>

*Angular — Interpolación y Binding (Property & Event) | by Fernando Andrade | Medium*. (n.d.). Retrieved September 15, 2025, from <https://medium.com/@ingenieroandrade/angular-interpolaci%C3%B3n-y-binding-property-event-fff0fd7723dc>

Bosarreyes, R. (2023, June 25). *¿Qué son los componentes y cómo se crean? | Rodrigo Bosarreyes | Medium*. <https://medium.com/@bosarreyesrodrigo/componentes-en-angular-i-63db68231067>

*Google Gemini*. (n.d.). Retrieved September 15, 2025

*Importar módulos - Javascript en español*. (n.d.). Retrieved September 15, 2025, from <https://lenguajejs.com/javascript/modulos/import/>

Simões, C. (2024, May 6). *Componentes reutilizables con Web Components*. <https://www.itdo.com/blog/componentes-reutilizables-con-web-components/>