

Creación Componente Producto Listado



Creación del Componente ProductoLista en Angular

En esta guía, te explicaré paso a paso cómo crear el componente `producto-lista` en **Angular**, por qué realizamos cada acción y cómo asegurarnos de que todo funcione correctamente.

1 Generar el Componente ProductoLista

Ejecuta el siguiente comando en la terminal dentro de tu proyecto de Angular:

```
ng g c producto-lista --skip-tests
```

◆ Explicación del comando

- `ng g c producto-lista`: Genera un nuevo **componente** llamado `ProductoListaComponent`.
- `--skip-tests`: Evita la creación del archivo de pruebas (`producto-lista.component.spec.ts`), ya que en este caso no lo necesitamos.

Después de ejecutar este comando, Angular generará una carpeta llamada `producto-lista` dentro de `src/app/`, que incluirá los siguientes archivos:

```
/src/app/producto-lista/
  producto-lista.component.ts
  producto-lista.component.html
  producto-lista.component.css (lo eliminaremos)
  producto-lista.component.spec.ts (no se generó por --skip-tests)
```

2 Editar el Archivo `producto-lista.component.html`

Reemplazamos su contenido con:

```
<div class="container">
  <h3>Sistema de Inventarios</h3>
</div>
```

◆ Explicación del Código

- `<div class="container">`: Creamos un contenedor para el contenido del componente.
- `<h3>Sistema de Inventarios</h3>`: Agregamos un título para mostrar en la interfaz.

Este código establece la **estructura visual básica** del componente `ProductoLista`.

3 Revisar `producto-lista.component.ts`

El archivo `producto-lista.component.ts` quedó así:

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-producto-lista',
  imports: [],
  templateUrl: './producto-lista.component.html'
})
export class ProductoListaComponent {
```

◆ Explicación del Código

1. `@Component({ ... })`: Decorador que define los metadatos del componente.
2. `selector: 'app-producto-lista'`: Permite usar el componente con `<app-producto-lista>`.
3. `imports: []`: No importamos módulos adicionales porque este componente aún no los necesita.
4. `templateUrl: './producto-lista.component.html'`: Define el archivo HTML que se usará para la plantilla del componente.
5. `export class ProductoListaComponent {}`: La clase del componente, que por ahora está vacía, pero en el futuro contendrá la lógica del inventario.

Nota: Al eliminar el archivo CSS (ver siguiente punto), no necesitamos el atributo `styleUrls`.

4 Eliminamos el Archivo `producto-lista.component.css`

El archivo `producto-lista.component.css` se eliminó porque **no estamos utilizando estilos específicos en este componente**.

◆ ¿Por qué se eliminó?

- Si el componente **no tiene estilos específicos**, podemos eliminar este archivo para mantener el código más limpio.
- Si en el futuro necesitamos estilos personalizados, podemos agregarlos en:
 - **Un archivo global** (`styles.css` en `src/`).
 - **Dentro del componente con `styleUrls`** (si lo volvemos a incluir).
 - **Directamente en el HTML con `style`** (menos recomendado).

Nota: Si Angular detecta que no hay archivo CSS, no intentará cargarlo, evitando errores.

5 Modificar el Componente en `app.component.html`

Abrimos `src/app/app.component.html` y agregamos:

```
<app-producto-lista></app-producto-lista>
```

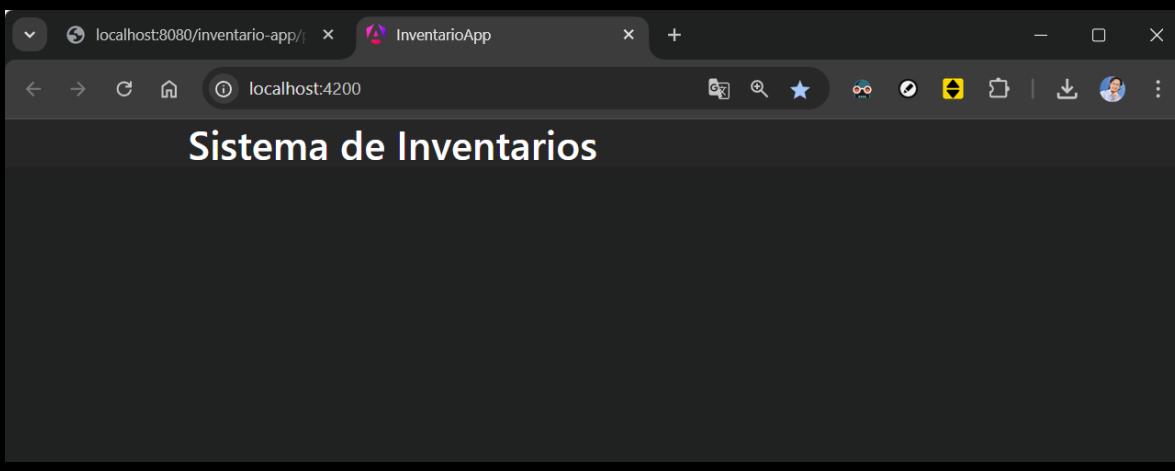
◆ ¿Por qué hacemos esto?

- <app-producto-lista> es el **selector** definido en `producto-lista.component.ts`.
- Al agregarlo en `app.component.html`, estamos **mostrando el componente `ProductoLista` dentro del componente principal de la aplicación (`AppComponent`)**.
- Esto nos permite reutilizar el componente sin necesidad de reescribir su código en `app.component.html`.

Nota: Angular usa esta estrategia para **modularizar la aplicación**, permitiendo dividir la lógica en **componentes reutilizables**.

Ejecutamos la aplicación

- Ejecutamos la aplicación con el comando: `ng serve -o`
- Entramos a la URL: <http://localhost:4200/>



Conclusión

- Generamos un componente `ProductoLista` en Angular.
- Lo agregamos en `app.component.html` para mostrarlo en la app.
- Definimos su HTML básico con una estructura simple.
- Eliminamos el archivo CSS porque no lo necesitábamos.
- Optimizamos el código del componente en `producto-lista.component.ts`.
 - ◆ Ahora tienes un **componente modular y reutilizable**, listo para expandirse con más funcionalidad .

Saludos!

Ing. Ubaldo Acosta

Fundador de GlobalMentoring.com.mx