












Angular 17

Tema 3. Arquitectura de Angular

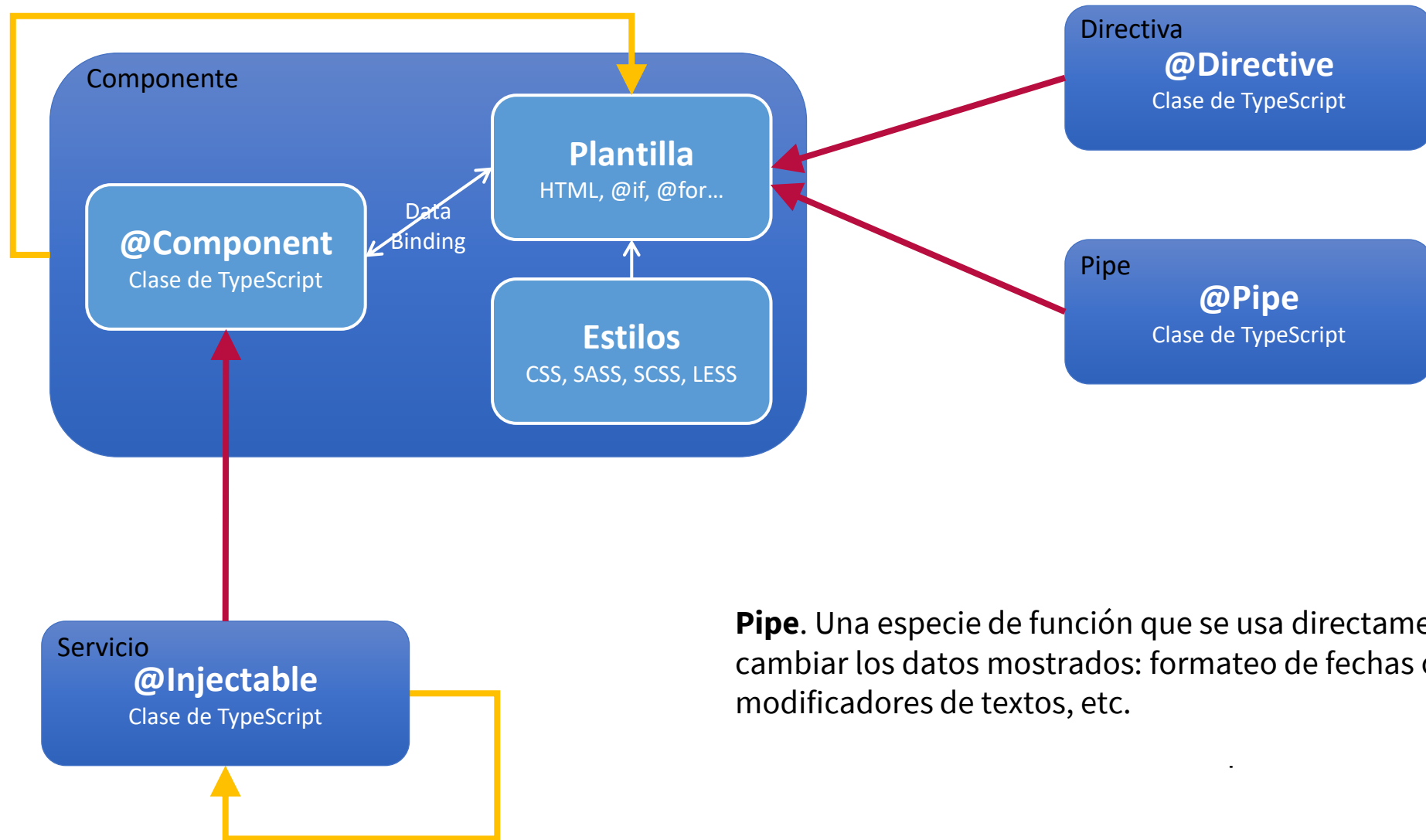
Objetivos

-  Funcionamiento de Angular
-  Conocer todos sus elementos
-  Saber diseñar correctamente una aplicación

Contenidos

-  Arranque de Angular
-  Componentes: TypeScript, Plantillas y CSS. Data binding
-  Directivas de atributo
-  Servicios
-  Pipes
-  Módulos (arquitectura tradicional)

Elementos de Angular



Pipe. Una especie de función que se usa directamente en la plantilla para cambiar los datos mostrados: formateo de fechas o números, modificadores de textos, etc.

Ejercicio 03 A (1/2)

Crea un proyecto llamado «ejercicio03A» con las opciones habituales y añádele tres componentes: triángulo, círculo y cuadrado.

Dibuja esos componentes repetidos varias veces en el componente principal de tal modo que el resultado sea similar al de la imagen. Lógicamente no me importa el aspecto exacto, pero sí que se dibujen de izquierda a derecha en tres filas separadas (yo lo he hecho con tres «div» distintas y un par de estilos)

He dibujado los componentes «geométricos» aplicando estas hojas de estilos (triángulo, círculo y cuadrado):

```
div {
  width:0px;
  height:0px;

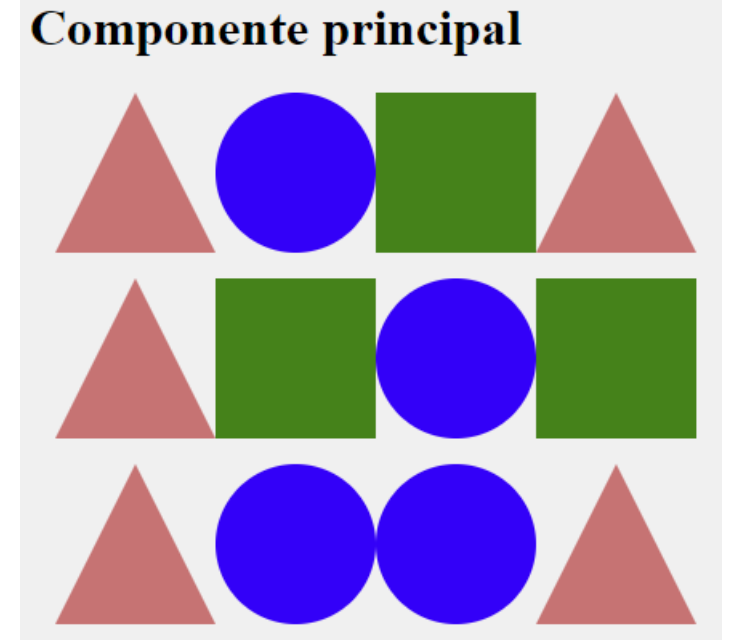
  border-style:solid;
  border-color:transparent transparent #e07070 transparent;

  border-top-width:0px;
  border-left-width:50px; /* base de 100 */
  border-right-width:50px;

  border-bottom-width:100px; /* altura */
}
```

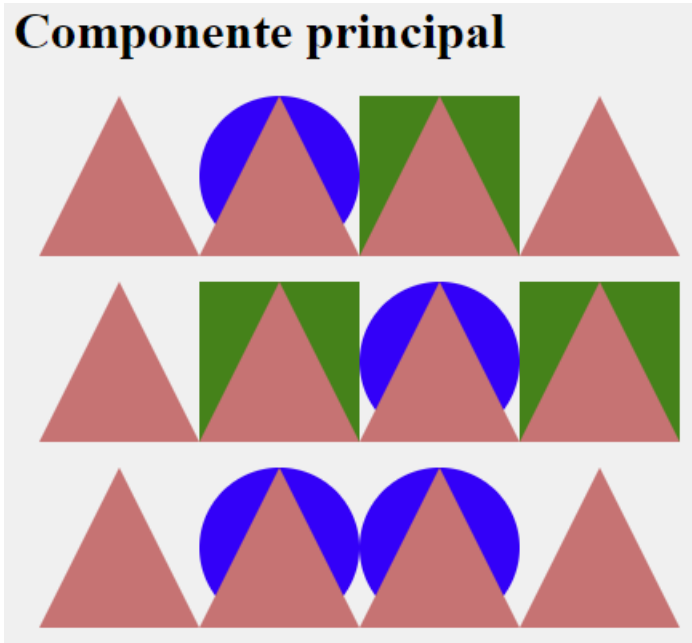
```
div {
  background-color: blue;
  width: 100px;
  height: 100px;
  border-radius: 50px;
}
```

```
div {
  background-color: green;
  width: 100px;
  height: 100px;
}
```



Ejercicio 03 A (2/2)

Cuando acabes, modifica el proyecto para que los cuadrados y los círculos dibujen dentro de ellos triángulos. Haciéndolo en mi ejemplo, el resultado es el siguiente:



Modifica el color de fondo de la página para que sea gris.

Ejercicio 03 B

Modifica el ejemplo que acabamos de ver y añade dos botones más a cada elemento de la lista, «disminuir» y «restaurar»:

Listado de Productos

- **100 Sed vitae.** 23 €
Nunc sed tellus egestas malesuada vel at erat

Aumentar Disminuir Restaurar
- **200 Donec ornare.** 23.32 €
Mi egestas mollis aliquam, metus odio facilisis arcu ut rhoncus sapien nisi id mauris

Aumentar Disminuir Restaurar
- **320 Nulla id iaculis.** 119 €
Tortor Donec id condimentum purus. Curabitur sit amet convallis felis

Aumentar Disminuir Restaurar

El primer botón disminuye un 5% el precio del producto y el segundo lo deja tal y como estaba al principio, independientemente de los cambios que se le hayan podido hacer.

Directivas predefinidas

Hay muchas, y nosotros definiremos alguna adicional. Pero las más habituales son éstas:

ngClass Añade o elimina clases.

```
[ngClass]="{'activado':true}"
```

ngStyle Permite aplicar estilos CSS a una etiqueta.

```
[ngStyle]="{'background-color':'red', 'font-size':'20pt'}"
```

ngModel Diseñada para gestionar controles de formulario. Se suele emplear «two way binding».

```
<input type="text" [(ngModel)]="campoDelComponente">
```

routerLink Enlace virtual para acceder a un componente cuando usamos rutas

```
<a routerLink="/productos/lista">Listado de productos</a>
```

***ngIf, *ngFor** Abreviaturas tradicionales (la sintaxis real no se solía usar) de las directivas «ngIf», «ngFor».

```
<li *ngFor="let valor of lista">{{valor}}</li>
```


Pipes predefinidos

Al igual que con las directivas, disponemos de decenas de funciones , y podemos crear las nuestras. El formato es:
«valor_typeScript | nombrePipe : parámetroUno : parámetroDos»

number Formatea números con comas y puntos. Permite establecer el «Locale» deseado.

```
<p> {{cantidad | number:'1.2-2'}} </p>
```

date Formatea números a fechas. Admite docenas de máscaras distintas.

```
<p> {{fecha | date:'dd-MM-YYYY h:mm:ss'}} </p>
```

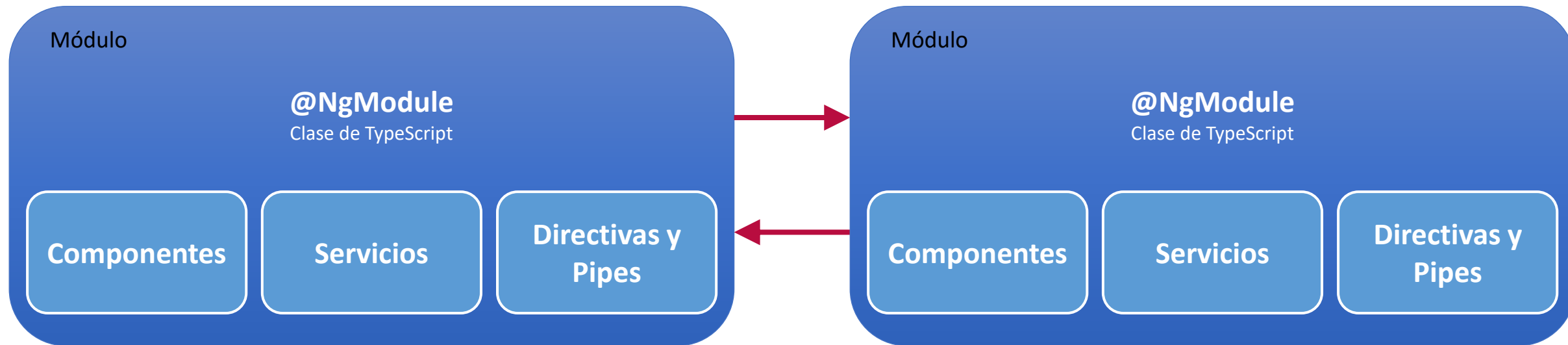
uppercase, lowercase Convierte un texto a mayúsculas o minúsculas.

```
<p> {{mensaje | lowercase}} </p>
```

json Muestra el contenido de un JSON. Muy útil durante el desarrollo.

```
<pre> {{unObjeto | json}} </pre>
```

Elementos de Angular. Módulos



Módulos. Clases de TypeScript que actúan como contenedores del resto de elementos.

En el **Angular tradicional**, la pieza fundamental de organización del código. La aplicación se compone de módulos reutilizables que definen específicamente qué contienen, qué comparten y qué necesitan de otros módulos.

Elementos de Angular. Definiciones

Componente. Un elemento «reutilizable» de la interfaz de usuario. Un elemento gráfico que a partir de ciertos datos puede trabajar de forma independiente: una lista de productos o el detalle de un usuario.

Servicio. Código de TypeScript puro que encapsula una parte de la lógica de negocio, un controlador por lo general compartido por varios componentes o por otros servicios: la conexión con el servidor, bibliotecas genéricas, etc.

Directiva. Permite modificar el comportamiento de una etiqueta de HTML simulando ser un atributo de la misma. Por ejemplo, puede cambiar su aspecto o contenido en función del valor del «atributo». Los atributos estándares también se pueden convertir en directivas.

Pipe. Una especie de función que se usa directamente en la plantilla para cambiar los datos mostrados: formateo de fechas o números, modificadores de textos, etc.

Módulos. Clases de TypeScript que actúan como contenedores del resto de elementos. En el **Angular tradicional**, la pieza fundamental de organización del código. La aplicación se compone de módulos reutilizables que definen específicamente qué contienen, qué comparten y qué necesitan de otros módulos.

¿Qué hemos aprendido?

- ▶ Elementos de Angular: componentes, servicios, directivas y pipes
- ▶ Elementos tradicionales: módulos
- ▶ Construcción y arranque de una aplicación

Resumen de comandos

npm

npm install [-g] paquete
npm install
npm uninstall paquete
npm -version

ng (1)

ng version
ng help
ng serve [-o]
ng build
ng new nombre_proyecto
 --routing false
 --skip-tests
 --skip-git
 --no-standalone

ng (2)

ng generate component nombre_componente / ng g c
ng generate service nombre_servicio / ng g s
ng generate module nombre_módulo / ng g m

Ejercicio 03 C

Se trata de otra modificación del ejercicio de listado de productos. Quiero que la lista aparezca paginada, y que puedas ir de una «página» a otra con los clásicos botones de «anterior» y «siguiente»:

Listado de Productos

[anterior](#) [siguiente](#)

- **K72SF / Mauris.** 94.5 €
05-11-2019
Interdum nec nunc et mollis
- **L23FG / Orci varius natoque.** 12.0 €
01-09-2007
Penatibus et magnis dis parturient montes, nascetur ridiculus mus.
- **M87GD / Suspendisse.** 42.0 €
29-11-2001
Dignissim orci est at vulputate nisl condimentum non. Vestibulum et enim sapien

Los botones sólo cambiarán los datos si existen, por ejemplo «anterior» no hará nada cuando estés al comienzo de la lista.

Añade una «fecha» al producto, y que tanto ésta como el precio aparezcan correctamente formateados con pipes. Una «ngClass» para remarcar los precios baratos y caros (escoge los valores que desees) tal como ves en la imagen; tendrás que escribir un par de comparaciones en la plantilla.