

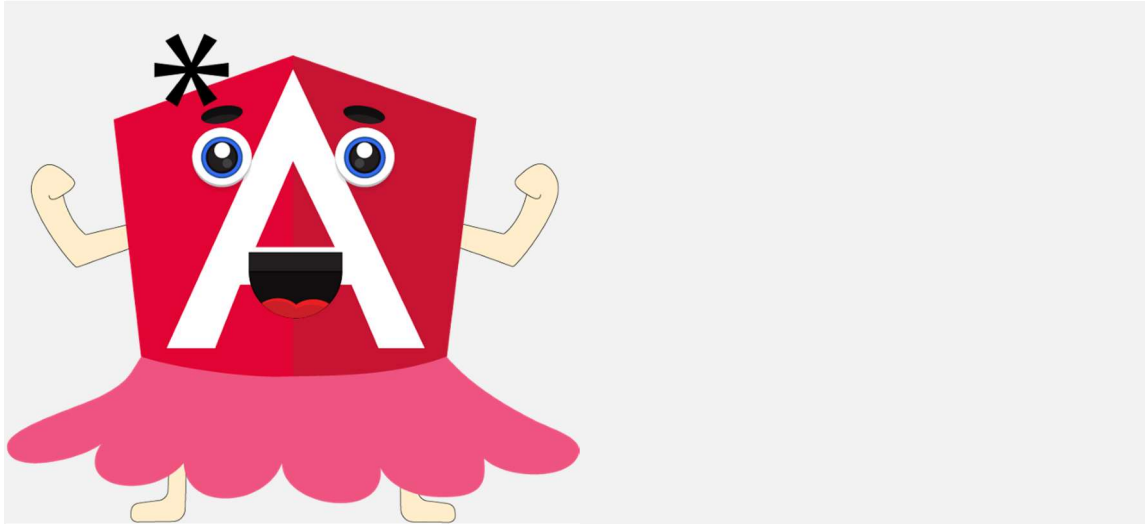
Directivas en Angular

Las Directivas son una parte súper importante de Angular, ellas nos permiten extender nuestro HTML y darle un toque único que veremos a continuación.



¿Qué es una Directiva?

Una directiva se puede considerar como una parte muy importante del núcleo de Angular.



Directiva

Las Directivas extienden la funcionalidad del HTML usando para ello una nueva sintaxis. Con ella podemos usar lógica que será ejecutada en el DOM (Document Object Model).

Cada Directiva que usamos tiene un nombre, y determina donde puede ser usada, sea en un elemento, atributo, componente o clase.

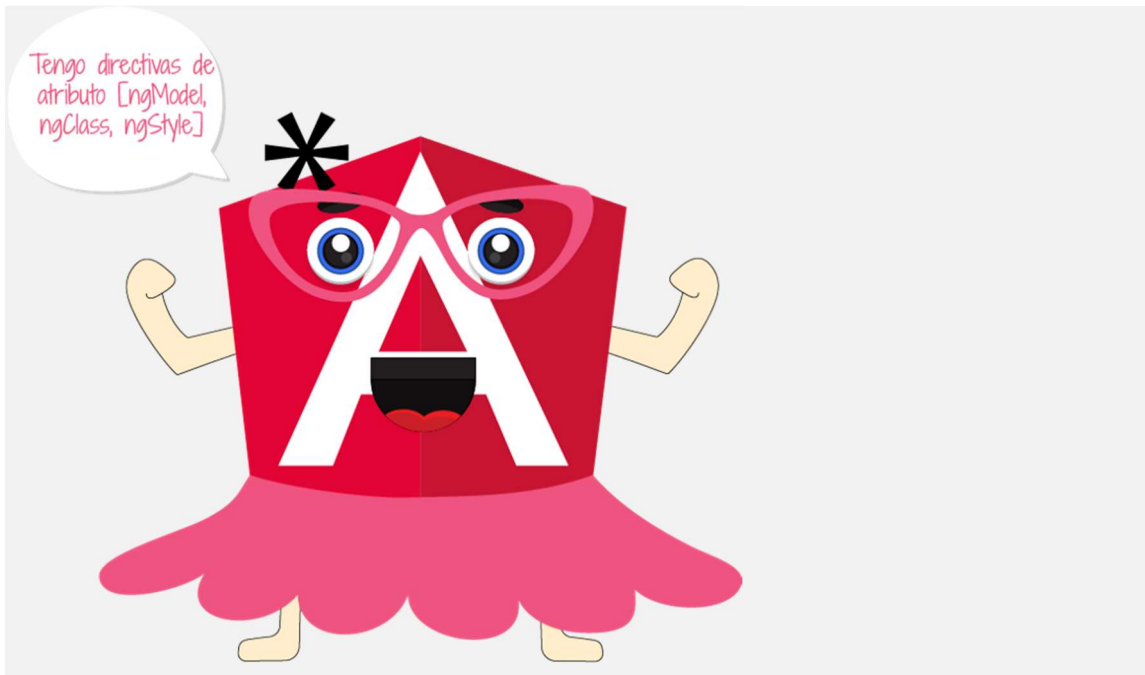
Se dividen en tres tipos diferentes:

- Directivas de Atributo
- Directivas de estructurales
- Componentes

Tipos de Directivas

Directivas de Atributo

Alteran la apariencia o comportamiento de un elemento del DOM y son usados como atributos de los elementos.



Entre la directivas de atributo, encontramos:

- **ngModel:** Implementa binding
- **ngClass:** permite añadir/eliminar varias clases
- **ngStyle:** permite asignar estilos inline

Directivas Estructurales

Alteran la estructura del DOM, agregando, eliminando y manipulando los elementos host a los que están unidos.



Algunas directivas estructurales tienen un asterisco (*), que precede al nombre del atributo de la directiva.

En las directivas estructurales podemos encontrar las siguientes:

- ***ngIf**: Nos permite incluir condicionales de lógica en nuestro código, como por ejemplo evaluar sentencias, hacer comparaciones, mostrar u ocultar secciones de código, y entre las muchas condiciones que deseemos crear, para que se renderice nuestro HTML, cumpliendo la sentencia a evaluar. Con el *ngIf, podemos evaluar sentencias con un simple If, podemos evaluar el **else**, para que no cumpliéndose la primera condición que se evalúa nuestro código ejecute otra acción en el caso contrario y podemos además incluir el **then**, para que cumpliéndose la condición afirmativa (if), podamos añadir más flexibilidad a nuestro código incluyéndole un camino afirmativo adicional.

```
<section class="img-section" *ngIf="selectedHero.name">
  <h2>{{selectedHero.name}}</h2>
  <img [src]="selectedHero.img" [alt]="selectedHero.name"/>
</section>
```

Ejemplo *ngIf

- ***ngFor**: Permite ejecutar bucles, los bucles son los que conocemos en lógica de programación como: for, while, foreach, etc. Con esta directiva estructural podemos evaluar de acuerdo a nuestra condición n veces.

```

<ul class="heroes">
  <li *ngFor="let hero of superHeroes" (click)="onSelect(hero)">
    <span class="badge">{{hero.id}}</span> {{hero.name}}
  </li>
</ul>

```

Ejemplo de *ngFor

- **ngSwitch**: esta directiva es similar al *ngIf, y es como el switch en lógica de programación. En esta directiva se pueden crear los diferentes casos que deseamos evaluar y cuando se cumple la condición esperada, oculta/muestra el HTML. Nos permite mantener nuestro código más limpio, si necesitamos evaluar varias sentencias.

```

1 <section>
2   <h1>Type of Super Heroes</h1>
3   <ul *ngFor="let hero of superHeroes" [ngSwitch]="hero.type">
4     <li *ngSwitchCase="'Marvel'" class="text-opt1">{{ hero.name }} {{ hero.type }}
5   </li>
6     <li *ngSwitchCase="'DC'" class="text-op2">{{ hero.name }} {{ hero.type }}
7   </li>
8     <li *ngSwitchDefault class="text-op3">{{ hero.name }} {{ hero.type }}
9   </li>
10  </ul>
11 </section>

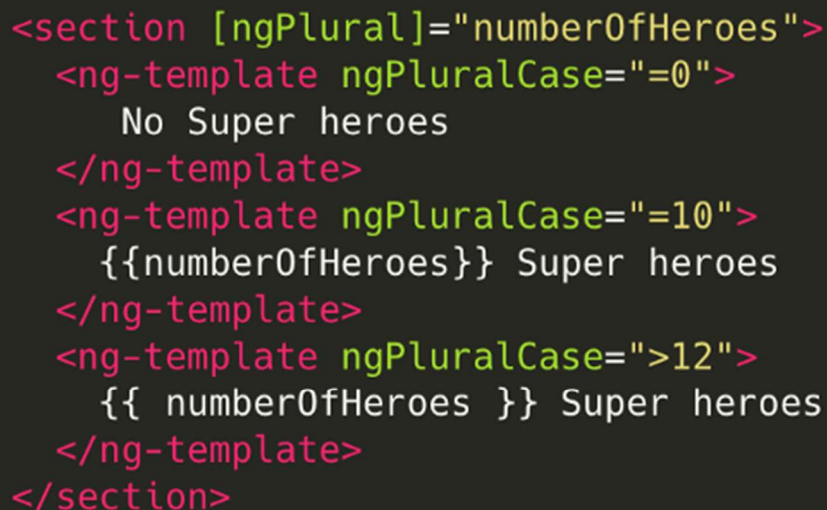
```

Ejemplo ngSwitch

- **ngPlural**: es una directiva que permite agregar o remover elementos del DOM, basado en un valor numérico. Para usar esta directiva, se debe proporcionar un elemento contenedor

que establezca el atributo [ngPlural] en una expresión de cambio. Los elementos internos con un [ngPluralCase] se mostrarán en función de su expresión. Si [ngPluralCase] se establece en una expresión (que comience con = o '>' o '<' etc.), el elemento se mostrará, si el valor es igual a la expresión.

Para mostrar valores por defecto se puede usar el string "other".



```
<section [ngPlural]="numberOfHeroes">
  <ng-template ngPluralCase="=0">
    No Super heroes
  </ng-template>
  <ng-template ngPluralCase="=10">
    {{numberOfHeroes}} Super heroes
  </ng-template>
  <ng-template ngPluralCase=">12">
    {{ numberOfHeroes }} Super heroes
  </ng-template>
</section>
```

Ejemplo ngPlural

- **ngTemplate:** esta directiva como su nombre lo indica es un template en Angular. El contenido de esta etiqueta puede reutilizarse en otros templates. Dentro de la etiqueta ng-template tenemos acceso a las mismas variables de contexto que son visibles en el template exterior, como por ejemplo la

variable 'noSuperHeroes'. Esto se debe a que las instancias de ng-template tienen acceso al mismo contexto en el cual están integradas. Además cada template también puede definir su set de variables.

```
<ng-template #noSuperHeroes>
  <div>There is not superHeroes to show</div>
</ng-template>

<ng-container *ngTemplateOutlet="noSuperHeroes"></ng-container>
```

Ejemplo ngTemplate

- **ngComponentOutlet**: nos permite crear componentes dinámicos.

```
<button type="button" (click)="renderComponent()">Render dynamic component</button>
<ng-container *ngComponentOutlet="currentComponent"></ng-container>
```

Ejemplo ngComponentOutlet

Directivas de Componente

Las Directivas de Componente son directivas con un Template. Los componentes tienen decoradores “**@Component**”, el componente es un decorador **@Directive** que es extendido con características propias de los *templates*.

¿Cómo generar una directiva?

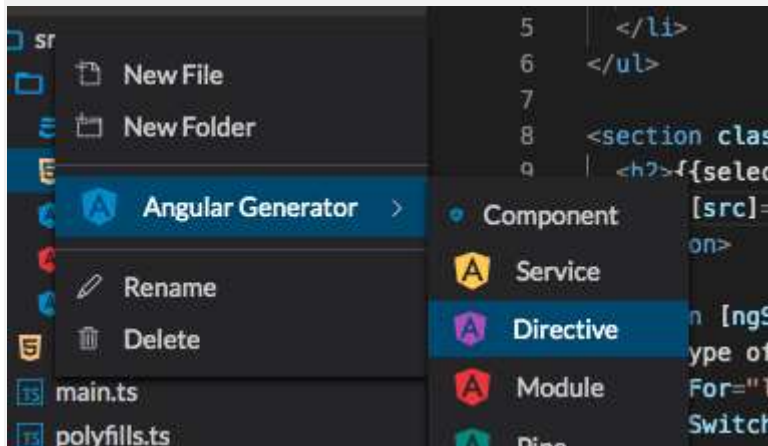
Desde el Angular CLI podemos generar una directiva usando el siguiente comando:

ng generate **directive** <name> [options]

En su forma abreviada sería:

ng g d [name]

Si usamos un IDE online como **Stackblitz** con solo dar clic derecho sobre la carpeta **src**, nos aparecerá una opción llamada Angular Generator, si colocamos el mouse sobre esa opción, mostrará varias opciones, entre ellas Directive. Al seleccionarla y colocar un nombre, se genera la directiva.



Creación de directiva usando Stackblitz

A screenshot of a code editor window in Stackblitz showing the content of a file named 'test.directive.ts'. The code is as follows:

```
1 import { Directive } from '@angular/core';
2
3 @Directive({
4   selector: '[appTest]'
5 })
6 export class TestDirective {
7
8   constructor() { }
9
10 }
```

Ejemplo de la directiva generada en Stackblitz

<https://stackblitz.com/>