



# Decorador @Directive

Lo que el decorador @Directive se utiliza para personalizar. Las directivas son instrucciones que se pueden usar en los elementos del DOM dentro de tu aplicación y pueden modificar su estructura, apariencia o comportamiento. A través del decorador @Directive, puedes crear tu propia directiva o usarla en los elementos del DOM dentro de tu aplicación.

Por ejemplo, puedes crear una directiva que cambie el fondo de un elemento en función del tiempo actual del día. Esto significa que el fondo del elemento cambie de color automáticamente.

En el archivo donde codificamos:

```
import { Directive, ElementRef, Renderer2, OnInit } from '@angular/core';

@Directive({
  selector: '[appCambiarFondoSegunHora]'
})
export class CambiarFondoSegunHoraDirective implements OnInit {
  constructor(private el: ElementRef, private renderer: Renderer2) { }

  ngOnInit() {
    this.cambiarFondoSegunHora();
  }

  private cambiarFondoSegunHora() {
    const horaActual = new Date().getHours();
    let colorFondo = '';

    if (horaActual >= 6 && horaActual < 12) {
      colorFondo = 'yellow'; // Mañana
    } else if (horaActual >= 12 && horaActual < 18) {
      colorFondo = 'lightgray'; // Tarde
    } else {
      colorFondo = 'gray'; // Noche
    }

    this.renderer.setStyle(this.el.nativeElement, 'background-color', colorFondo);
  }
}
```

```
@Directive({
  selector: '[appCambiarFondoSegunHora]'
})
```

@Directive es un decorador que se utiliza para definir una directiva personalizada en Angular.

```
selector: '[appCambiarFondoSegunHora]',
```

Indica que esta directiva se aplicará a elementos del DOM con el atributo appCambiarFondoSegunHora.

```
constructor(private el: ElementRef, private renderer: Renderer2) { }
```

En el constructor, se inyectan las instancias de ElementRef y Renderer2. ElementRef proporciona acceso al elemento del DOM al que se aplica la directiva, y Renderer2 permite manipular el DOM de manera segura.

```
ngOnInit() {
  this.cambiarFondoSegunHora();
}
```

El método ngOnInit() se llama automáticamente después de que Angular haya inicializado la directiva. En este

caso, se utiliza para llamar al método `cambiarFondoSegunHora()`.

```
private cambiarFondoSegunHora() {
  const horaActual = new Date().getHours();
  let colorFondo = '';

  if (horaActual >= 6 && horaActual < 12) {
    colorFondo = 'yellow'; // Mañana
  } else if (horaActual >= 12 && horaActual < 18) {
    colorFondo = 'lightgray'; // Tarde
  } else {
    colorFondo = 'gray'; // Noche
  }
  this.renderer.setStyle(this.el.nativeElement, 'background-color', colorFondo);
}
```

El método privado `cambiarFondoSegunHora()` determina la hora actual y asigna un color de fondo según la hora del día. Luego, utiliza `Renderer2` para establecer el estilo del fondo del elemento.

- Ahora en la clase principal de la aplicación hacemos uso de esta directiva.

`app.component.ts`

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { CambiarFondoSegunHoraDirective } from './cambiar-fondo-segun-hora.directive';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet, CambiarFondoSegunHoraDirective],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
}
```

Y en la plantilla disponemos la directiva que hemos creado:

`app.component.html`

```
<div appCambiarFondoSegunHora>
  Este fondo cambia según la hora del día.
</div>

<router-outlet />
```

Podemos probar esta aplicación también en la web [aquí](#).

## Retornar