

Angular  
Escalable

# Desarrollo sostenible

Reutilización de código



# Descomponer para reutilizar



## **Repartir responsabilidad**

Muchos pequeños actores  
especializados



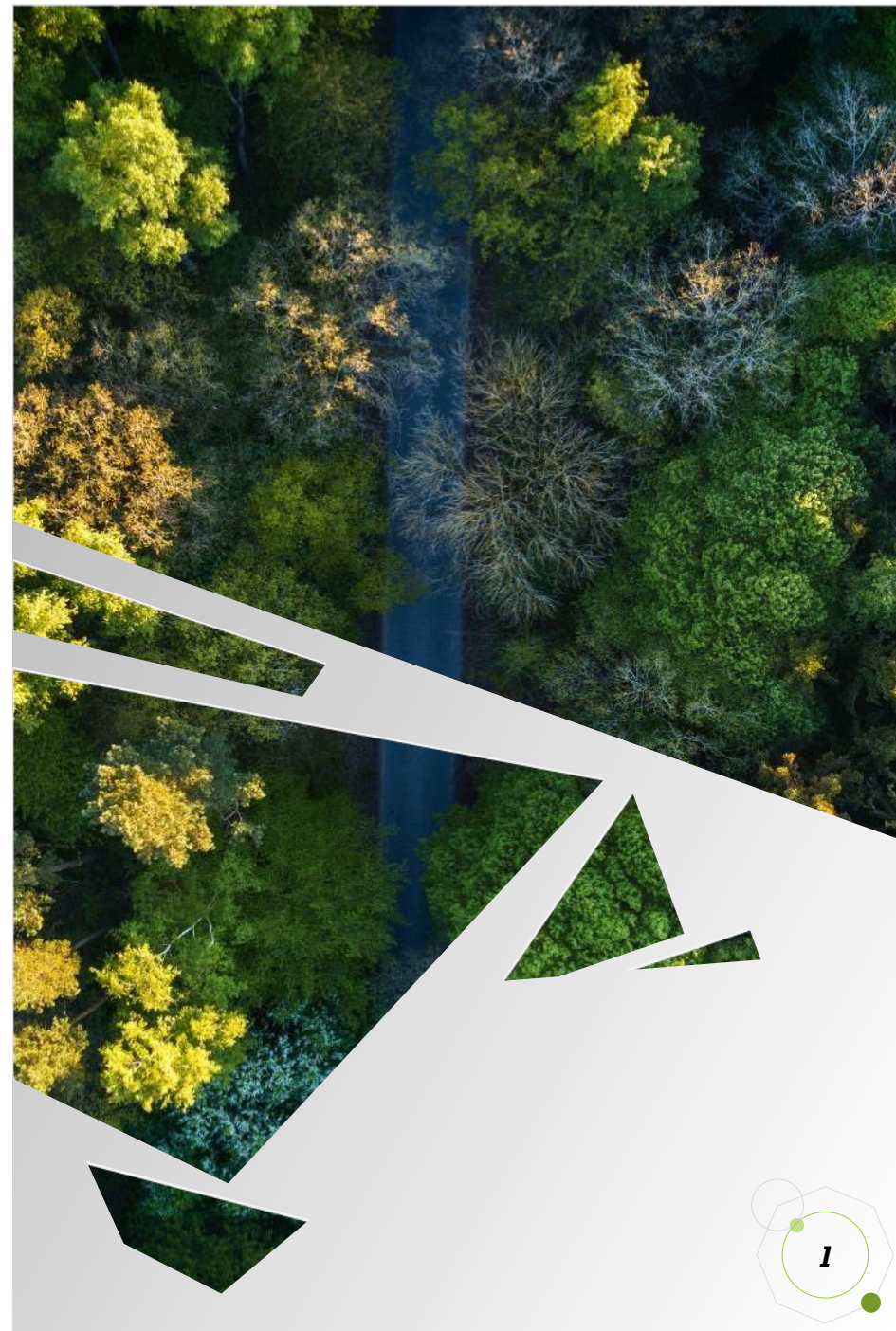
## **Visual e infraestructura**

No sólo presentación,  
también lógica y gestión  
de datos.



## **D.R.Y.**

El objetivo es escribir en  
un solo sitio cada regla de  
negocio.



# Plantillas

Incrustar HTML  
en los *slots*  
de otro componente.

`<ng-content select="">`

## ➤ Otros

- ng-container
- ng-template





# Directivas



## **Custom attribute**

Agrega capacidades a elementos HTML estándar y a componentes Angular



## **ElementRef**

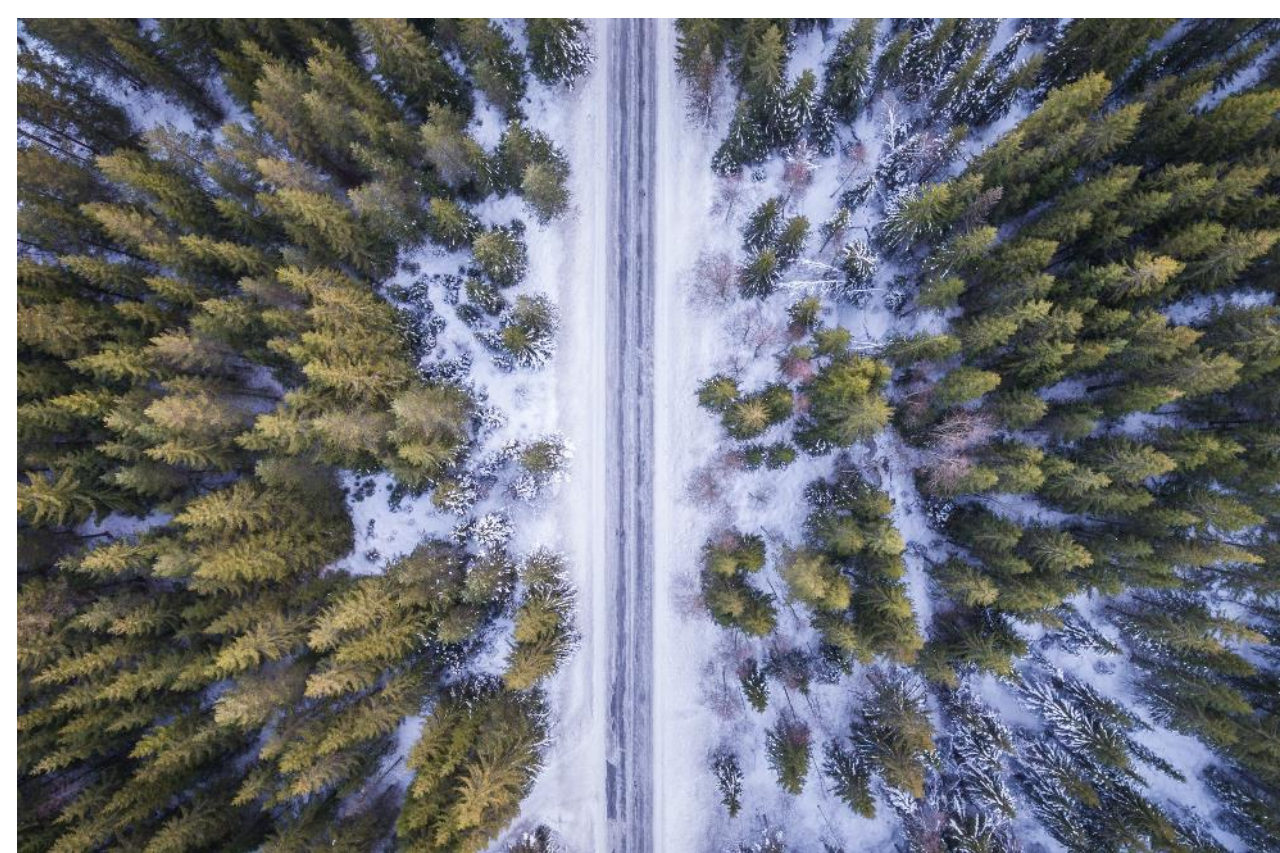
Dependencia con referencia directa al elemento HTML nativo



## **@HostListener**

Suscripción a eventos HTML estándar





# Tuberías

Limpian la vista, limpian el controlador  
y reducen el coste de recálculo

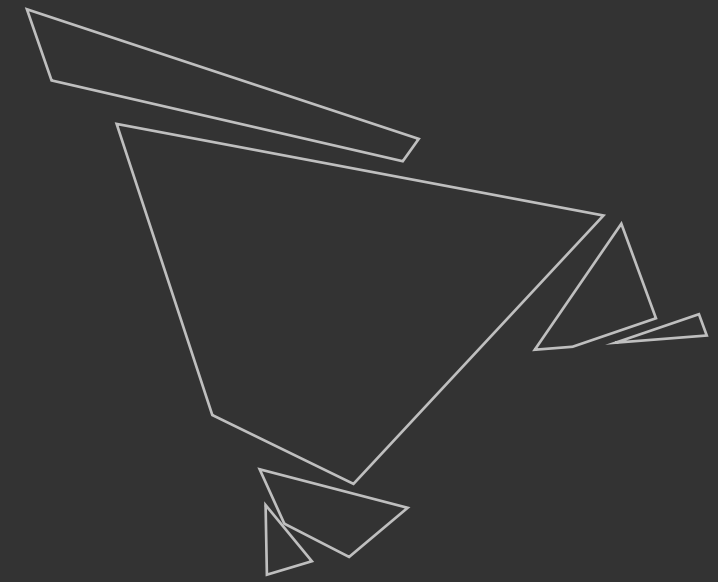


## Cambiar la representación sin cambiar el origen

Busca antes de crear.

Los casos más comunes ya están desarrollados.





# Interceptores

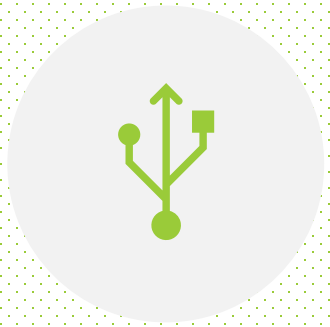
Aplicar lógica común a tus **comunicaciones**

Mediante el uso de **patrones**:

Inyección de dependencias

Inversión del control

# Usos comunes que ahorran código y centralizan lógica



## **Adaptador**

Cambios en los datos que enviamos o recibimos



## **Cache**

Evitar llamadas innecesarias con datos locales



## **Seguridad**

Envío de credenciales y control de respuestas



## **Reintento**

Recuperación ante caídas transitorias



## **Errores**

Gestión unificada para reporte a desarrollo e información a usuario





# Angular sostenible

Evita la duplicidad  
sin caer en abstracciones complejas

- Descompón interfaces
- Reparte responsabilidades
- Centraliza decisiones