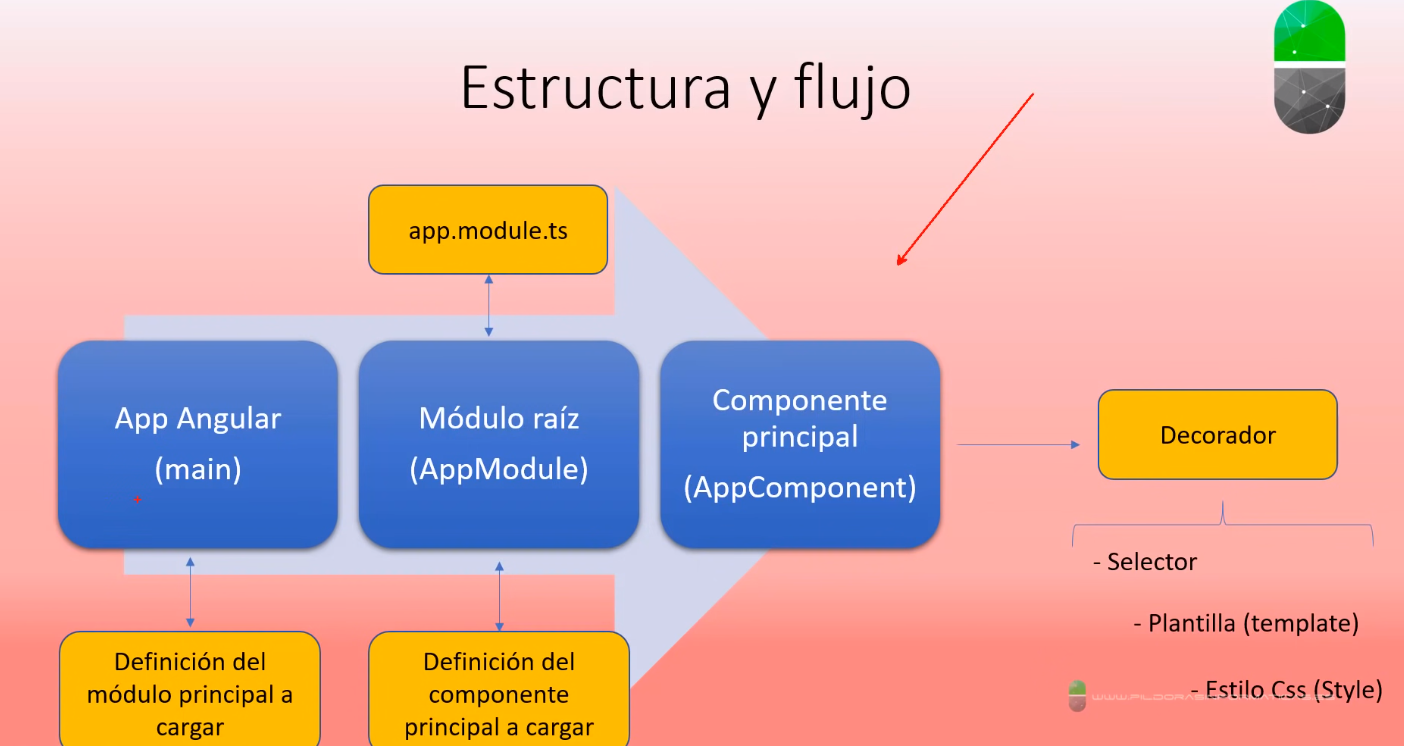
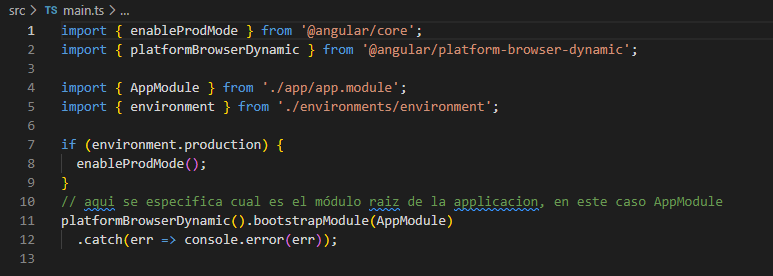
Video 4:

Estructura y flujo de una página de Angular

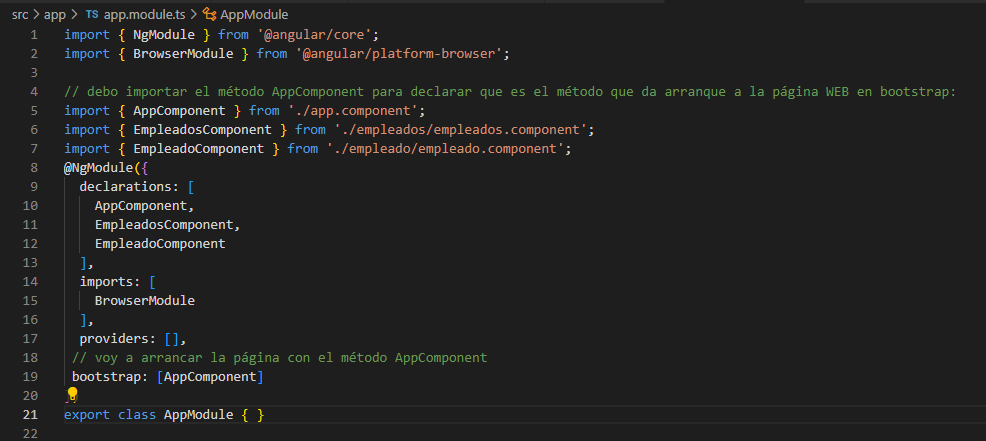


1. El punto de inicio es main. En main está la definición del modulo principal que se va a cargar, este modulo es el módulo raíz.
2. El módulo raíz está definido en el módulo app.module.ts.
3. En el módulo raíz se encuentra la definición del componente principal a cargar.
4. Por definición cuando instalamos Angular el componente principal se llama AppComponent.
5. AppComponent tiene la definición de una clase y de un decorador.

En main.ts:

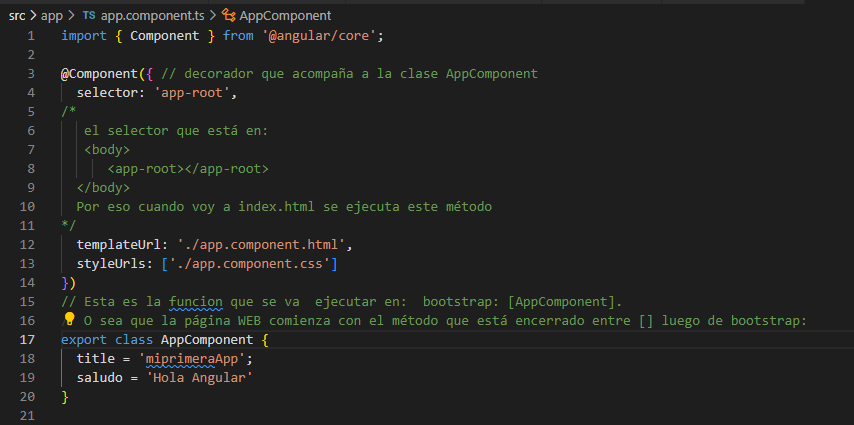


Vamos a cargar AppModule como modulo raíz:



La página va a arrancar con AppComponent porque es el componente que esta entre [ ] luego de la palabra Bootstrap:

En AppComponent vemos que tenemos el selector app-root



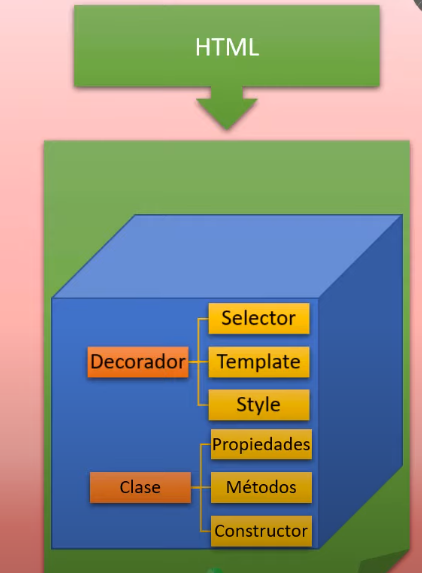
Recapitulando:

De main.ts 🡪 app.module.ts 🡪 app.componente.ts

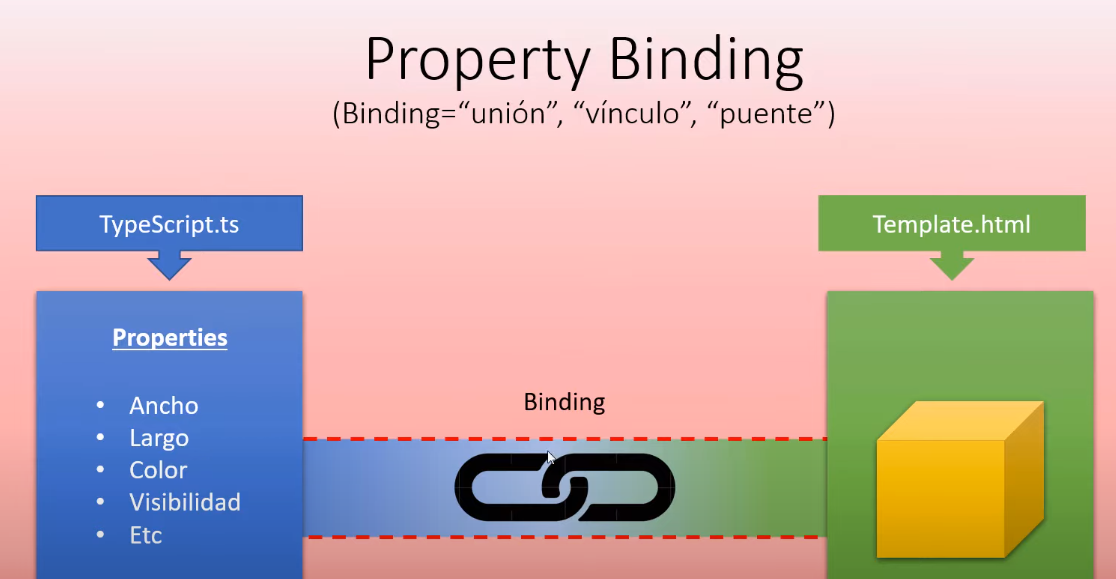
En app.component..ts especificamos:

* El selector que lo carga: ‘app-root’
* La plantilla donde se carga: ‘app.component.html’
* El style que le da formato: ‘app.component.css’

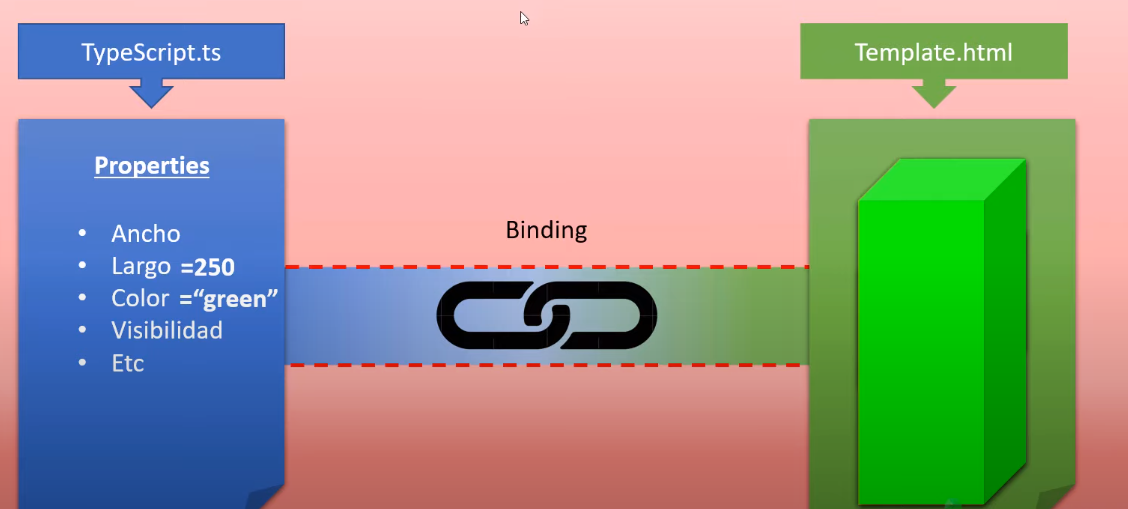
Agregar componentes



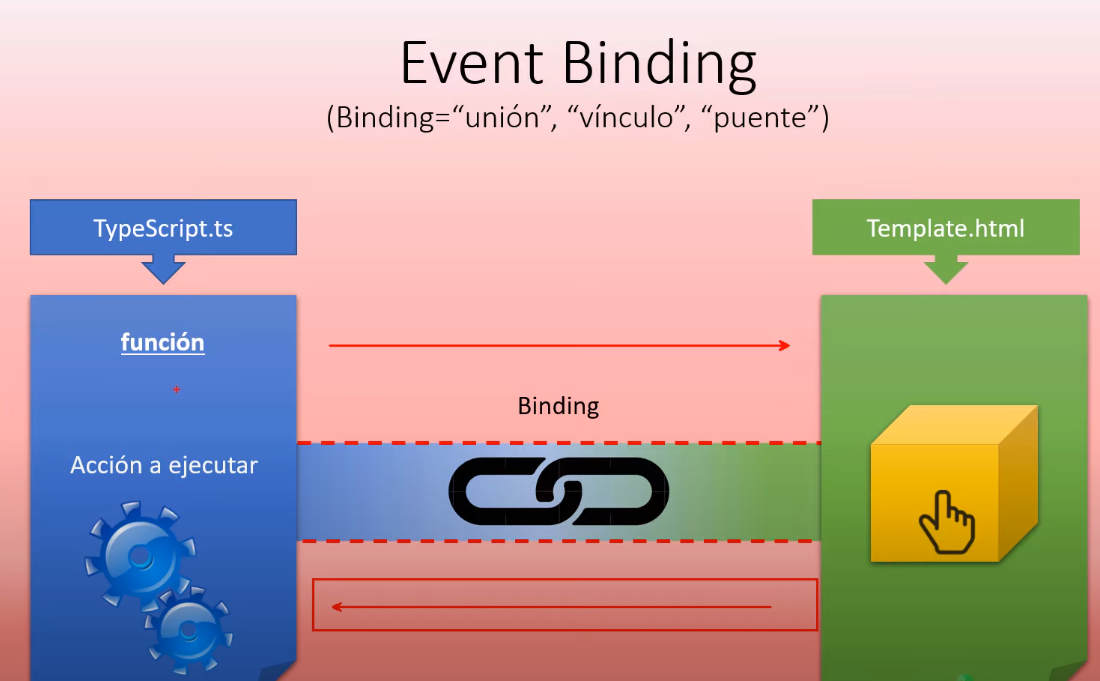
Property binding: Cambiamos las características de un objeto de forma dinámica



Cambio las propiedades

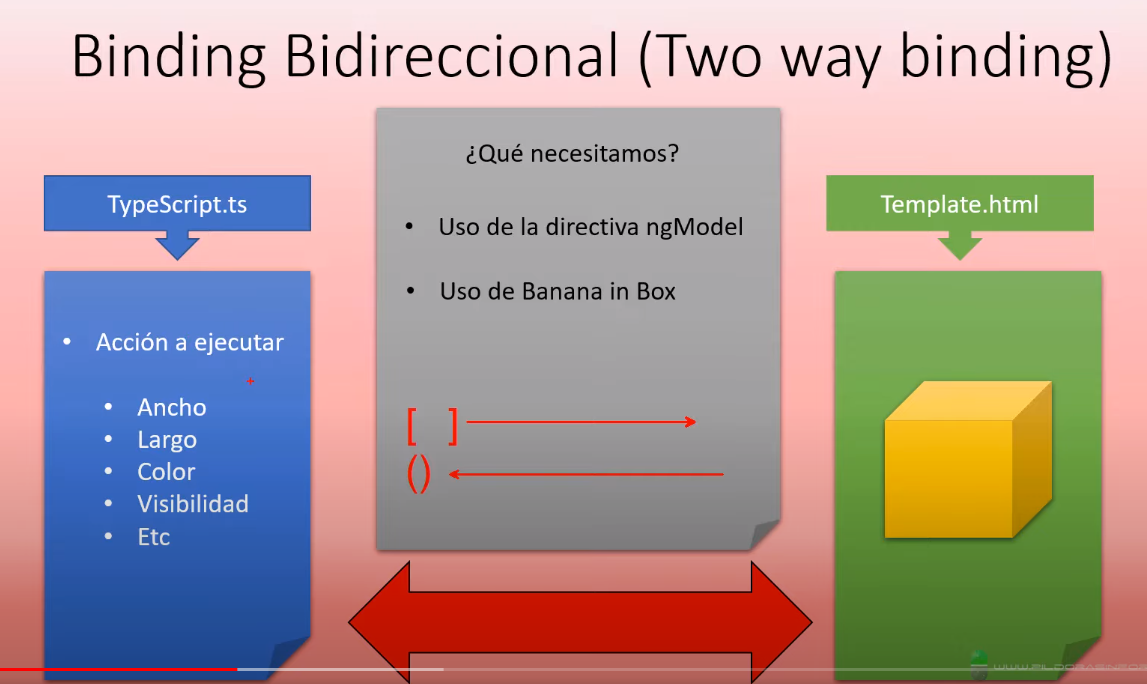


Event binding: Cuando hago una acción sobre un objeto se desencadena una función

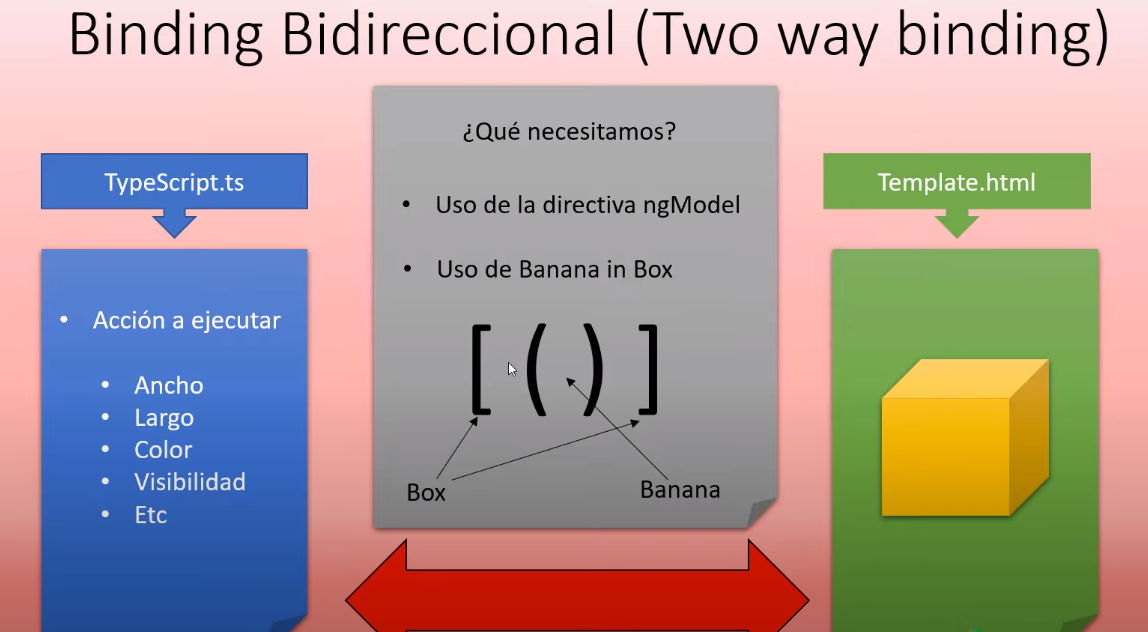


Puede ser bidireccional pero inicialmente vemos de objeto a función.

Binding bidireccional:



Property binding [ ] Event binding ( )



Two way binding: [( )] : Banana in box

No es imprescindible usar [( )] y ngModel pero facilita las cosas

**Para instalar Bootstrap:**

En la terminal escribir: npm install bootstrap –save

En el archivo angular.json agregar:

 "styles": [

              "src/styles.css",

              "node\_modules/bootstrap/dist/css/bootstrap.min.css"

            ],

            "scripts": ["node\_modules/bootstrap/dist/js/bootstrap.min.js" ]

Ya puedo usar todas las clases de Bootstrap

**Directivas:**

