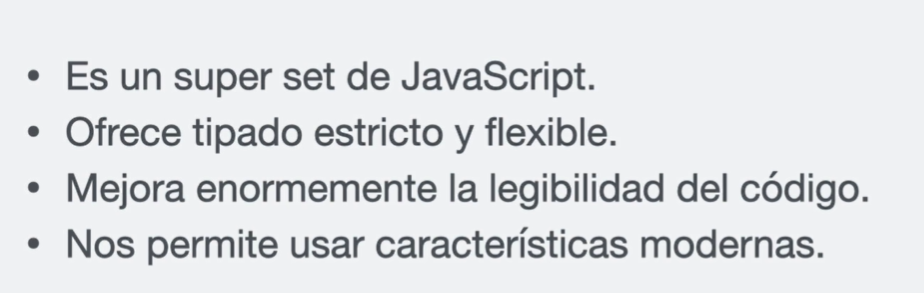
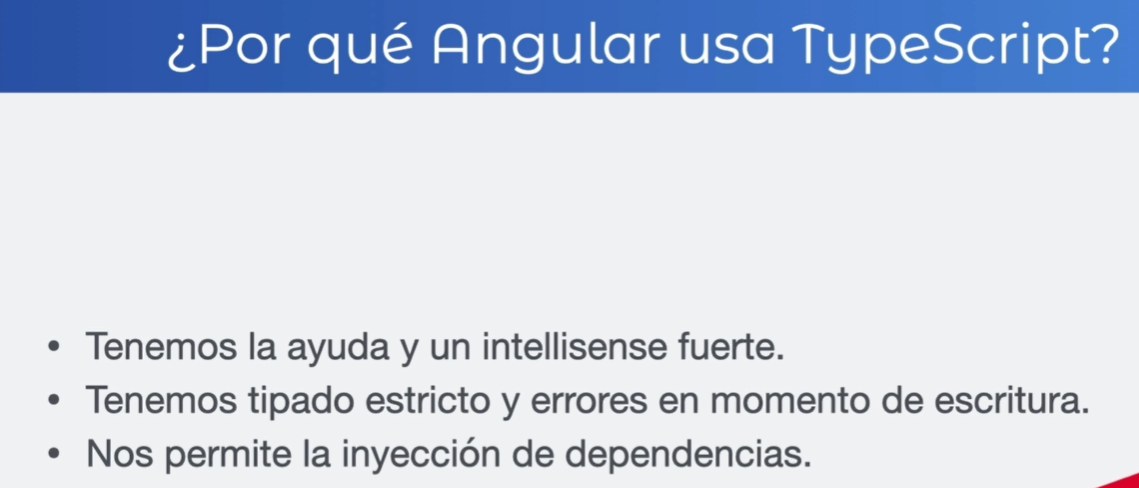
**Curso Angular 2021**

**Angular CLI:** Nos permite lanzar comandos para optimizar tiempo, ya se encarga de crear componentes, modulos,directivas, pipes,etc.

**TypeScript:** Es una extendida de javascript, mejorándolo y orientándolo a POO.





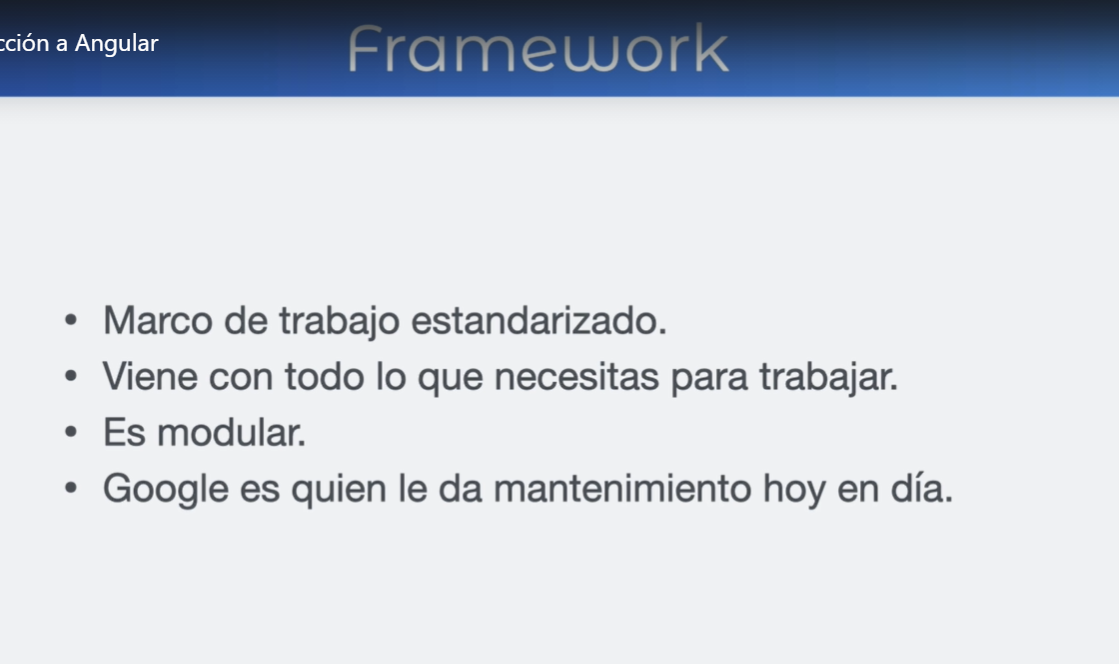
Decoradores: con los decoradores en angular, sirven para diferencias si la clase es un componente, o es un pipe, o una directiva, etc.

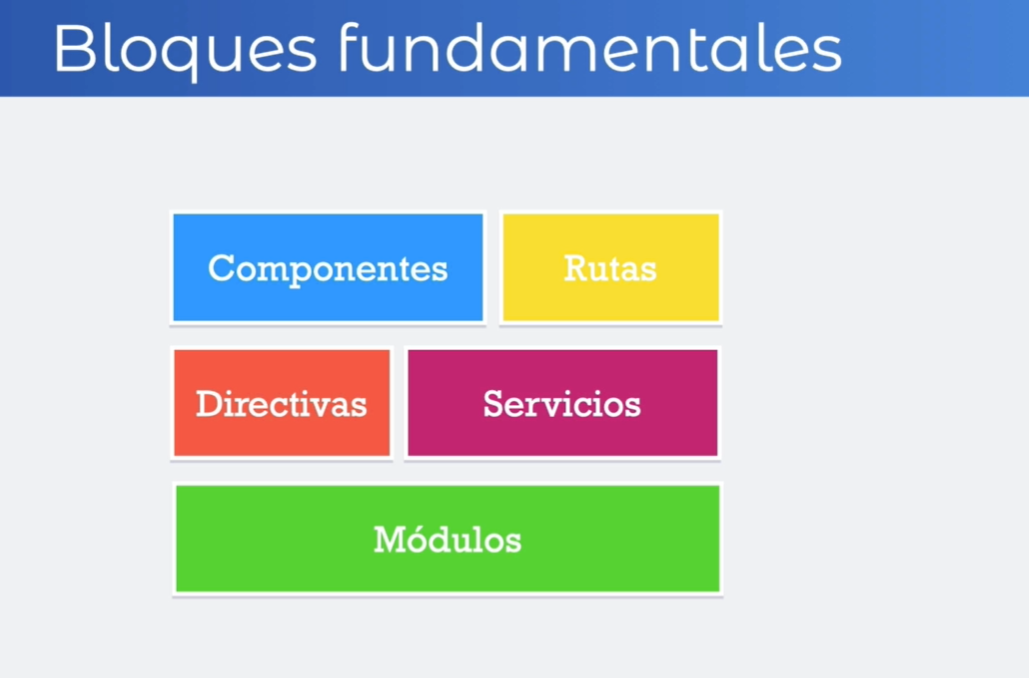
Un decorador es una función que expande las funcionalidades de la clase a la cual se le añade.

**Introducción a Angular**

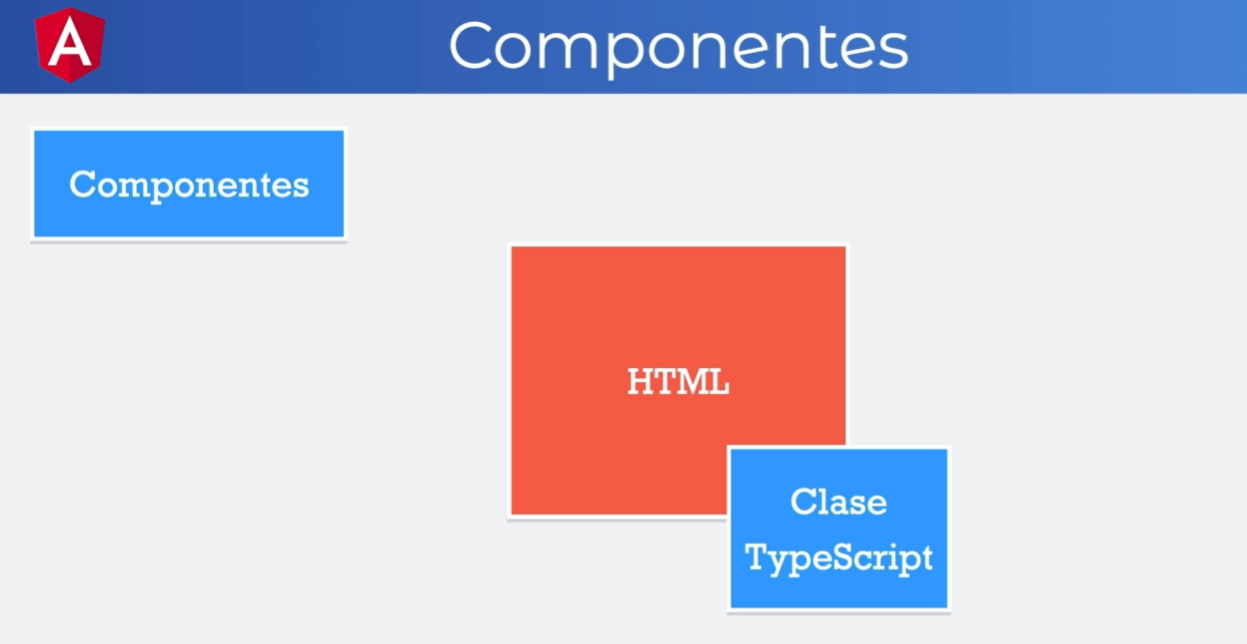
Con angular se puede aplicar móvil, web.

Angular es un framework Javascript





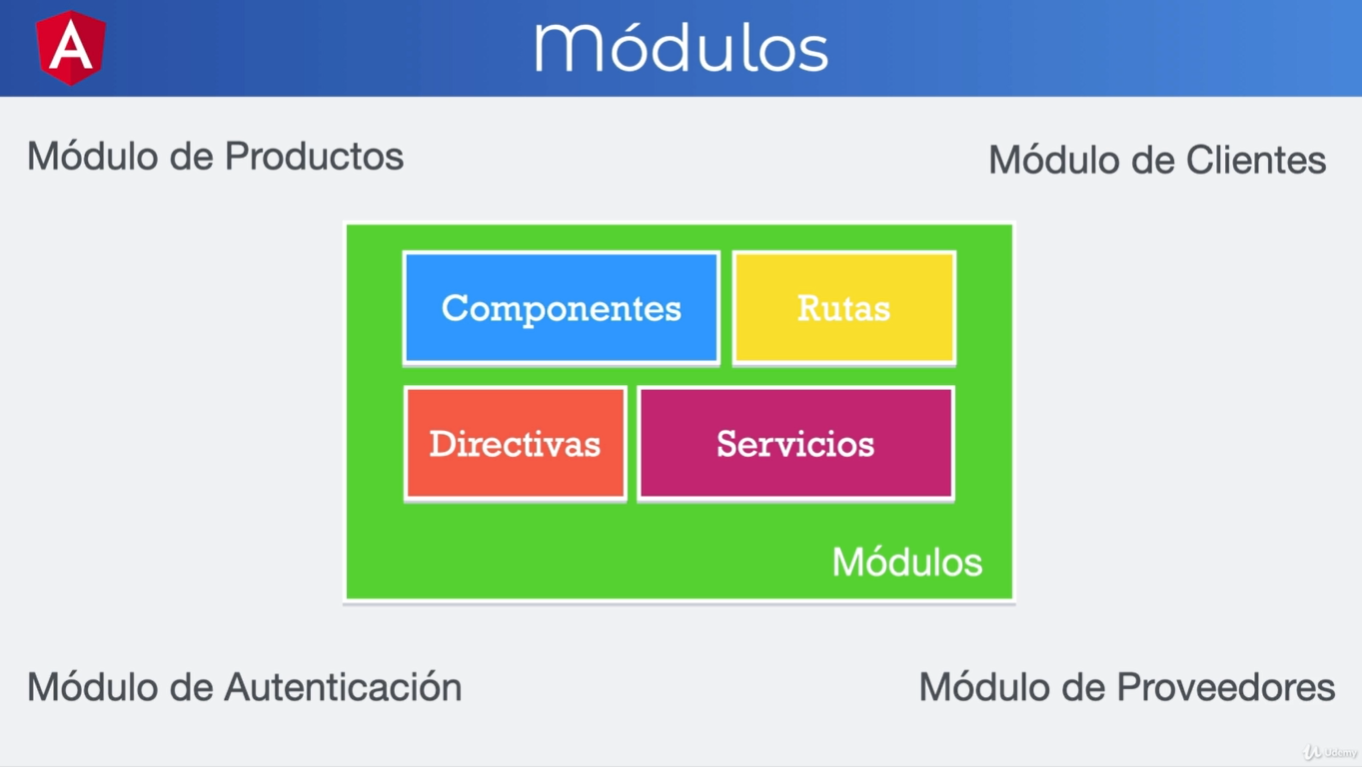
**Componentes**: Son bloques de código que tiene html y una clase typescript



**Servicios:** Son lugares centralizados de información.







**Tslint.json:** Son reglas de código limpio.

**Tsconfig.json:** Dice como se quiere transformar typescript a javascript

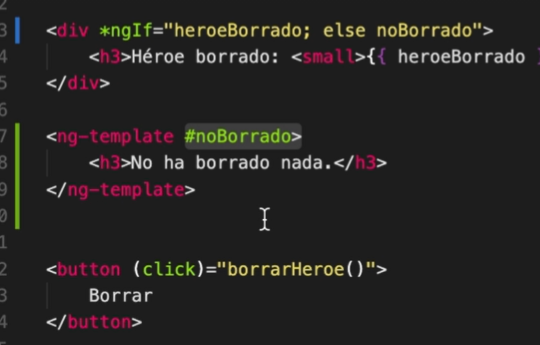
**Tsconfig.spec.json**: Es una extendida de tsconfig y funciona para las pruebas.

**Karma.confi.js:** Configuración de las pruebas manejas por karma.

**Angular.json:** Configuraciones del proyecto, se agrega donde esta los estilos, los scripts y lo inyecta a nivel del proyecto.

**One way data bindings:** La vinculación de datos unidireccional vinculará los datos del componente a la vista (DOM) o de la vista al componente. El enlace de datos unidireccional es unidireccional. Solo puede vincular los datos del componente a la vista o de la vista al componente.

Las referencias locales son del lado del template (#), el **ng-template** se utiliza en una condición if else, donde en el **ng template** se ejecuta la sentencia else.



**Modulos:** Nos ayudan a agrupar componentes que están relacionados entre sí.

-Encapsula los elementos y ayuda con la carga perozosa

Siempre en angular, se maneja clases y se implementa un decorado,

Para el caso del modulo se hace lo siguiente:



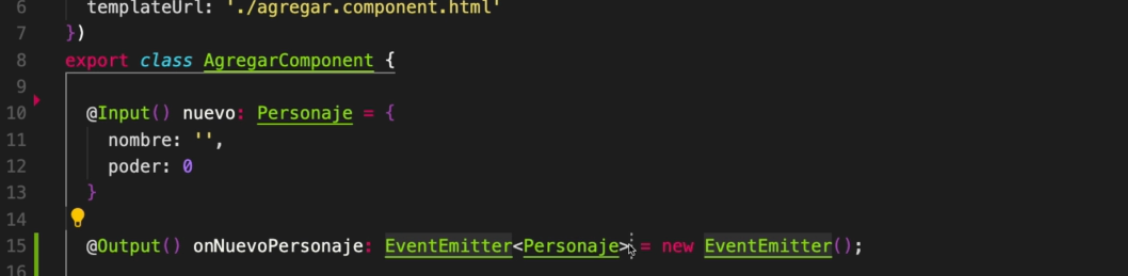
Los **declarations** son los componentes que se encuentran dentro del modulo.

Los **exports** son los componentes que queremos que otros modulos tengan acceso a ese componente.

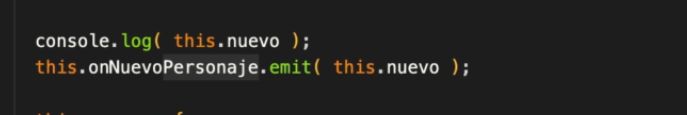
Los **imports (SOLO SE IMPORTAN MODULOS)** sirve para utilizar modulos externos.

**Two way data bindings**: nos permite modificar el valor de una propiedad de forma bidireccional, de forma que cuando nosotros modificamos el valor de una propiedad en la vista el valor también se actualiza en el modelo de datos. Resumiendo, cambiamos el valor de una propiedad con una caja de texto en la vista, ese valor se cambia en la vista y también en el componente.

**El input** sirve para llevarle valores a un componente hijo y un **output** Es para devolver información al componente padre.

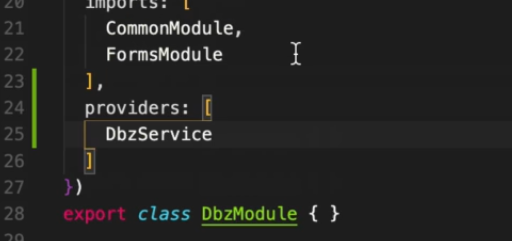


El **@Output()** se maneja con evento que emite el valor



**Servicios:** Se encargan de centralizar la información, estos funcionan como un singleton que consiste en crear una sola instancia de la clase

En el módulo, **Providers** se agregan los servicios.



Facilitan el manejo de la información.