



IPARTEK

APPS HÍBRIDAS CON IONIC

NATIVO VS WEB VS HÍBRIDAS

NATIVA

ALTO RENDIMIENTO

MAYOR ADAPTACIÓN AL LOOK & FEEL

ACCESO COMPLETO A LAS FUNCIONES DEL DISPOSITIVO

NO PORTABLE

APRENDIZAJE DE VARIOS LENGUAJES

WEB

DESARROLLO ÚNICO

COSTE

HTML5-CSS3-JS (CURVA DE APRENDIZAJE BAJA)

MENOR RENDIMIENTO

POCA ADAPTACIÓN AL LOOK & FEEL

ACCESO MUY LIMITADO A LAS FUNCIONES DEL DISPOSITIVO

HÍBRIDA

DESARROLLO ÚNICO

ADAPTACIÓN AL LOOK & FEEL

COSTE

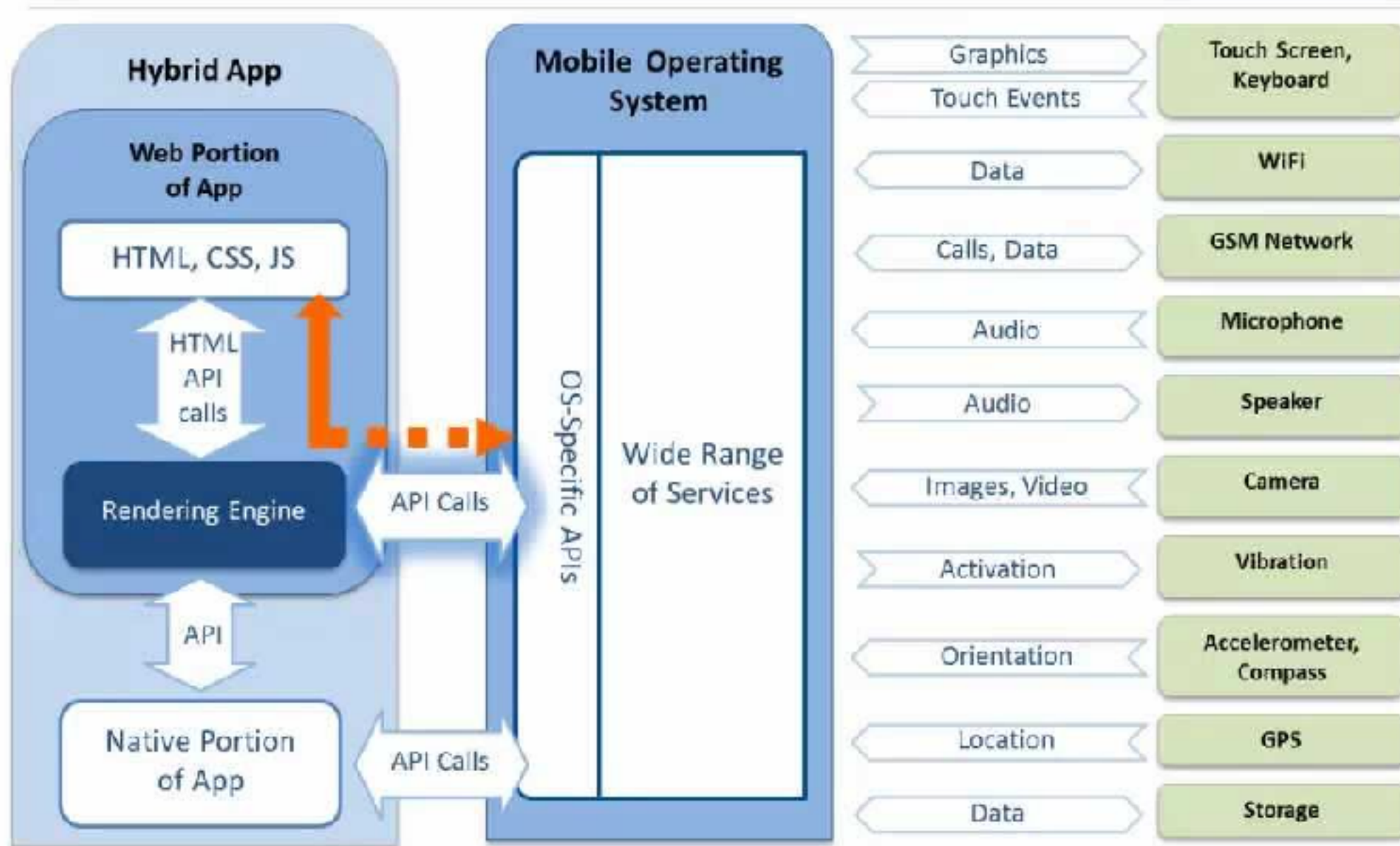
CURVA DE APRENDIZAJE MODERADA (HTML5-CSS3-JS + IONIC-ANGULAR)

MENOR RENDIMIENTO

ADAPTACIÓN RELATIVA A LA TECNOLOGÍA

ACCESO A LAS FUNCIONES DEL DISPOSITIVO A VECES DEPENDIENTES DE FRAMEWORKS

RENDIMIENTO



TECNOLOGÍAS

ionic@2



ANGULAR

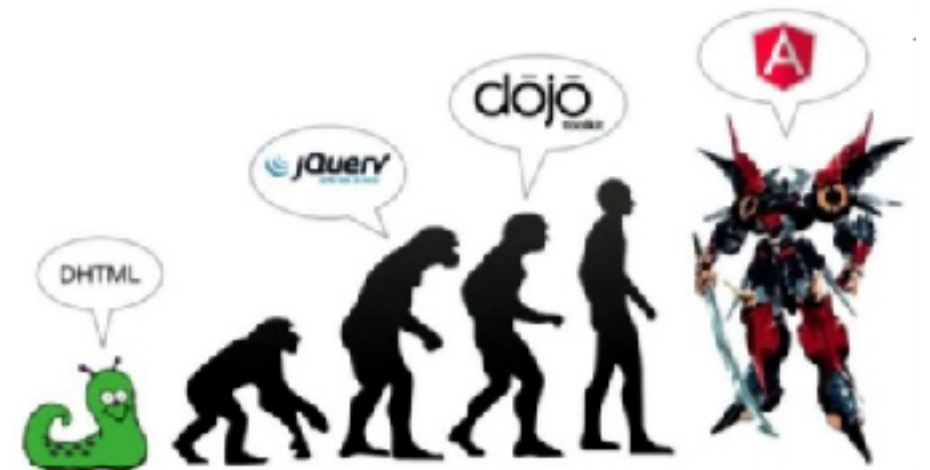
VIENE A SOLUCIONAR EL PROBLEMA DE LA LÓGICA DE NEGOCIO EN EL FRONT

PROPORCIONABA TAMBIÉN UN PATRÓN MVC, LLAMADO MVVM

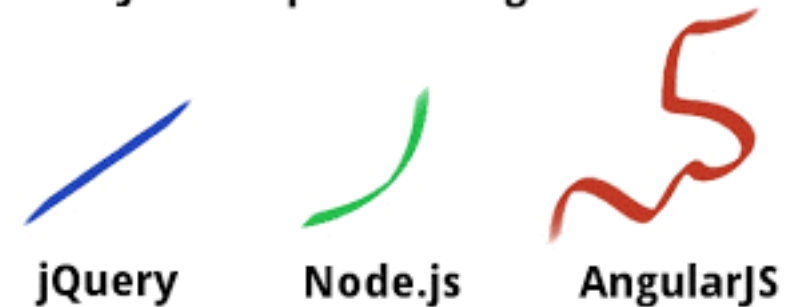
CREAR APPS CON ANGULAR:

- CREAMOS TEMPLATES CON EL MARKUP DE ANGULAR
- COMPONENTES PARA GESTIONAR ESAS PLANTILLAS
- DIRECTIVAS QUE AFECTAN AL COMPORTAMIENTO DE LOS COMPONENTES.
- ENCAPSULAMOS LA LÓGICA DE LA APLICACIÓN EN SERVICIOS
- DEFINIMOS UN MÓDULO PRINCIPAL

Evolution of Web Apps



JavaScript Learning Curves



ANGULARJS – ANGULAR2 – ANGULAR4

ANGULARJS != ANGULAR 2

A MEDIDA QUE LAS APPS CON ANGULAR HA IDO CRECIENDO, HAN IDO SURGIENDO PROBLEMAS.

ANGULAR 2 RESUELVE LOS SIGUIENTES PROBLEMAS QUE TENIA ANGULARJS:

- UTILIZACIÓN DE TYPESCRIPT
- LAZY SPA (TREE SHAKING)
- RENDERIZADO UNIVERSAL
- DATA BINDING FLOW
- COMPONENTES

NOVEDADES ANGULAR 4:

- AÑADIDO ELSE EN EL NGTEMPLATE
- MÓDULO DE ANIMACIÓN SEPARADO
- USO DE STRICTNULLCHECKS DE TYPESCRIPT
- MEJORAS DE RENDIMIENTO GRACIAS A FESM

ANGULAR CLI (COMMAND LINE INTERFACE)

NOS OFRECE DE MANERA SENCILLA UNA INTERFAZ PARA:

- CREAR EL ESQUELETO DE UNA APLICACIÓN ANGULAR 2 (NG NEW)
- UN SERVIDOR PARA SERVIR EL PROYECTO POR HTTP (NG SERVE)
- UN SISTEMA DE LIVE-RELOAD, PARA QUE CUANDO CAMBIAMOS ARCHIVOS DE LA APLICACIÓN SE REFRESQUE EL NAVEGADOR
- HERRAMIENTAS PARA TESTING
- HERRAMIENTAS PARA DESPLIEGUE DEL PROYECTO

DISPOSICIÓN DE CARPETAS

PRUEBA

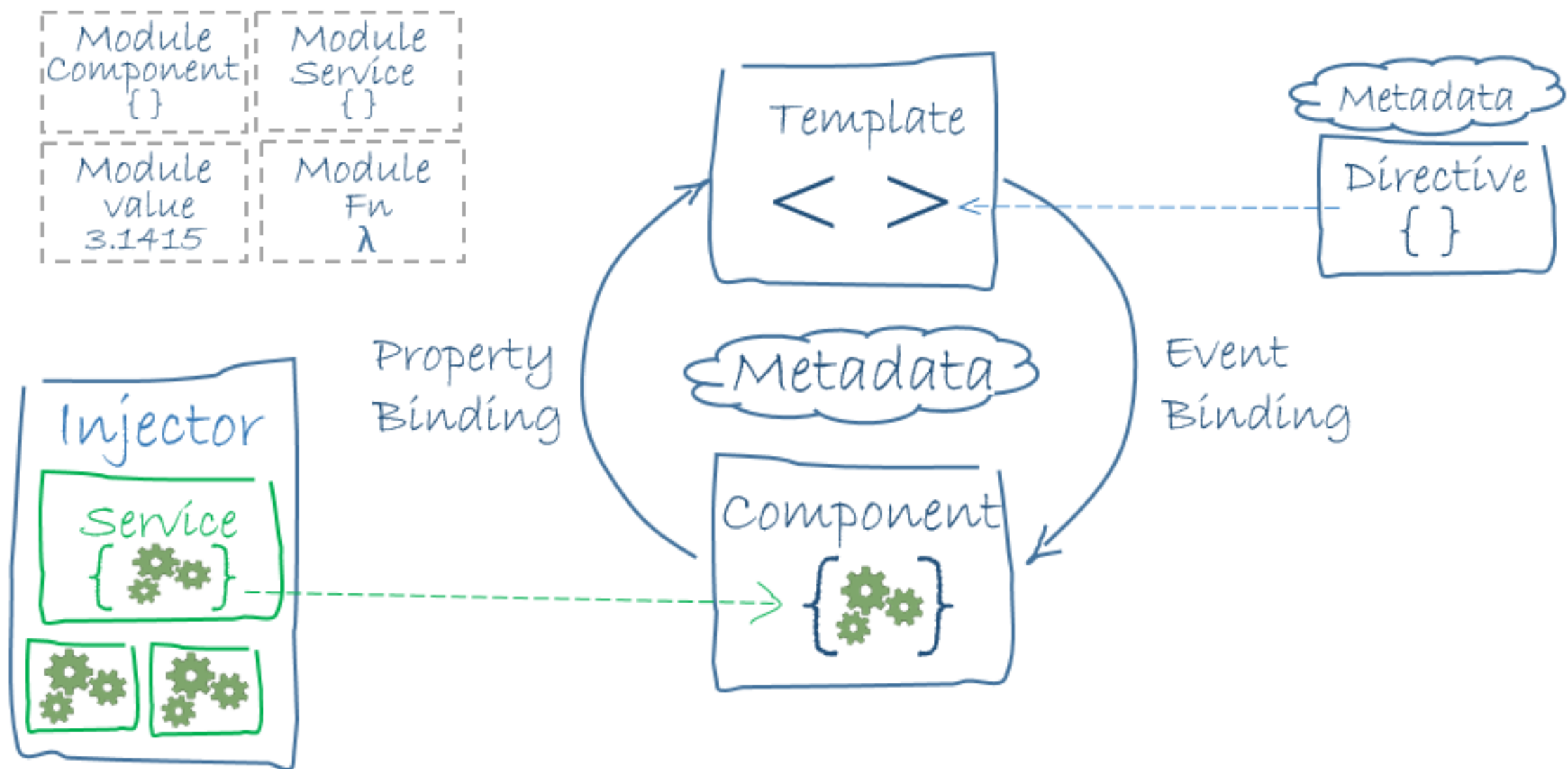
- e2e
- node_modules
- src
- { } .angular-cli.json
- ⚙ .editorconfig
- 🔄 .gitignore
- 🔑 karma.conf.js
- { } package-lock.json
- { } package.json
- JS protractor.conf.js
- 📖 README.md
- { } tsconfig.json
- { } tslint.json

ES PARA EL DESARROLLO DE LAS PRUEBAS. VIENE DE "END TO END" TESTING.

SON LOS ARCHIVOS DE LAS DEPENDENCIAS QUE MANTENEMOS VÍA NPM. POR TANTO,

LUGAR DONDE COLOCARÁS EL CÓDIGO FUENTE DE TU PROYECTO.

ARQUITECTURA ANGULAR



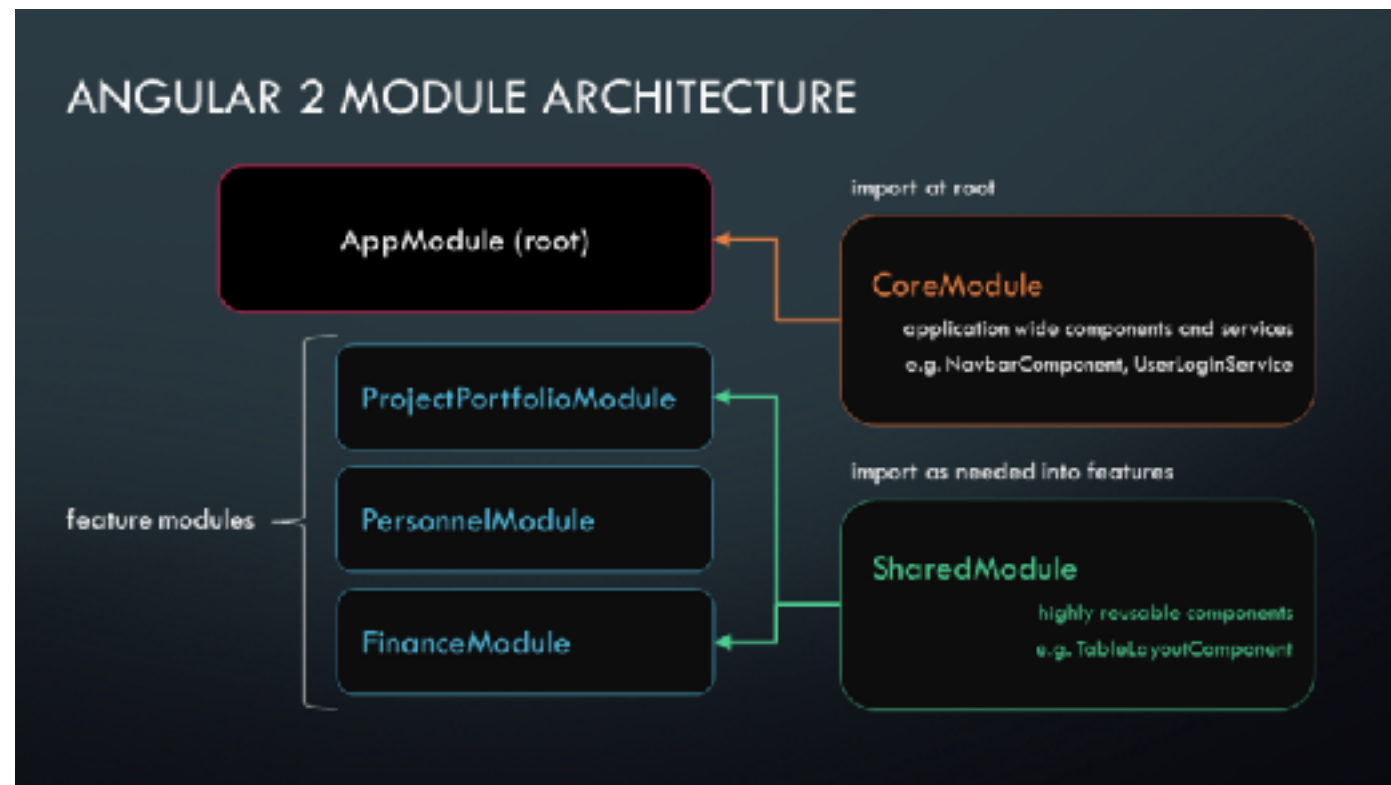
MODULOS

LAS APLICACIONES EN ANGULAR SON MODULARES.

EL MODULO PRINCIPAL ES APP

DENTRO DE CADA MODULO ES HABITUAL TENER MAS DE UN COMPONENTE

ENTENDEREMOS LOS MÓDULOS COMO PIEZAS DE LEGO PARA MONTAR NUESTRA APLICACIÓN



COMPONENTES

LOS COMPONENTES SON ETIQUETAS HTML QUE VAMOS A CREAR NOSOTROS DONDE SE MOSTRARA AQUELLO QUE NOS INTERESE

UN COMPONENTE ESTA COMPUESTO POR:

- UN DECORADOR (@COMPONENT) QUE DECLARA UN SELECTOR, UN HTML Y UN CSS
- LA CLASE XXXCOMPONENT

LA CLASE COMPONENTE SE EXPORTA PARA QUE PUEDA SER IMPORTABLE

EN ESTA CLASE COLOCAREMOS TODOS LOS ATRIBUTOS Y MÉTODOS QUE VAMOS A UTILIZAR EN LA VISTA

¿COMPONENTES O DIRECTIVAS?

EN ANGULAR 2 PERDURA EL CONCEPTO DE DIRECTIVA.

LOS COMPONENTES SON PIEZAS DE NEGOCIO,

LAS DIRECTIVAS SE SUELEN USAR PARA PRESENTACIÓN.

SE PUEDE PENSAR EN UN COMPONENTE COMO UN LISTADO DE DATOS, UN FORMULARIO, ETC.

EN LA DOCUMENTACIÓN DE ANGULAR 2 INDICA QUE UN COMPONENTE ES UN TIPO DE DIRECTIVA.

EXISTEN TRES TIPOS DE DIRECTIVAS:

- COMPONENTES: UN COMPONENTE ES UNA DIRECTIVA CON UN TEMPLATE. RESUELVEN NECESIDADES DEL NEGOCIO.
- DIRECTIVAS DE ATRIBUTOS: CAMBIAN LA APARIENCIA O COMPORTAMIENTO DE UN ELEMENTO. (NGCLASS)
- DIRECTIVAS ESTRUCTURALES: SON LAS QUE REALIZAN CAMBIOS EN EL DOM DEL DOCUMENTO. (NGFOR)

DECORADOR DE COMPONENT

ANGULAR 2 USA LOS DECORADORES PARA REGISTRAR UN COMPONENTE.

LA FORMA DE UN DECORADOR ES LA SIGUIENTE:

```
@COMPONENT({  
  MODULEID: MODULE.ID,  
  SELECTOR: 'ETIQUETA',  
  TEMPLATEURL: 'HTML',  
  STYLEURLS: ['CSS']  
})
```

AL DECORADOR LE ENTREGAMOS SON UNOS "METADATOS" PARA DESCRIBIR AL COMPONENTE QUE SE ESTÁ CREANDO.

LOS METADATOS SON:

- MODULEID: SIRVE PARA PODER RESOLVER URLS RELATIVAS. PERO NO TENEMOS QUE PONERLA YA QUE ES PARA TODOS LOS COMPONENT LA MISMA.
- SELECTOR: ESTE ES EL NOMBRE DE LA ETIQUETA HTML QUE USAREMOS PARA MOSTRAR NUESTRO COMPONENTE.
- TEMPLATEURL: ES EL NOMBRE DEL ARCHIVO .HTML CON EL CONTENIDO DEL COMPONENTE.
- STYLEURLS: ES UN ARRAY CON TODAS LAS HOJAS DE ESTILOS CSS QUE DEBEN PROCESARSE COMO ESTILO LOCAL PARA ESTE COMPONENTE.

VISTAS

LAS VISTAS FORMAN PARTE DE LOS COMPONENTES.

EN ANGULAR DENTRO DE LAS VISTAS PODEMOS UTILIZAR:

PROPIEDAD: CUALQUIER VALOR QUE PODEMOS ASIGNAR POR MEDIO DE UN ATRIBUTO DEL HTML. PUEDE SER UN ATRIBUTO HTML, PROPIO DE ANGULAR 2 O UN ATRIBUTO PERSONALIZADO, CREADO PARA UN COMPONENTE EN ESPECÍFICO.

EXPRESIÓN: ES UN VOLCADO DE CUALQUIER INFORMACIÓN EN EL TEXTO DE LA PÁGINA.

BINDING: ES UN ENLACE ENTRE EL MODELO Y LA VISTA.

EVENTO: ES UN SUCESO QUE OCURRE Y PARA EL CUAL SE PUEDEN DEFINIR MANEJADORES.

