

# Desarrollo de aplicaciones móviles con Ionic

# Agenda

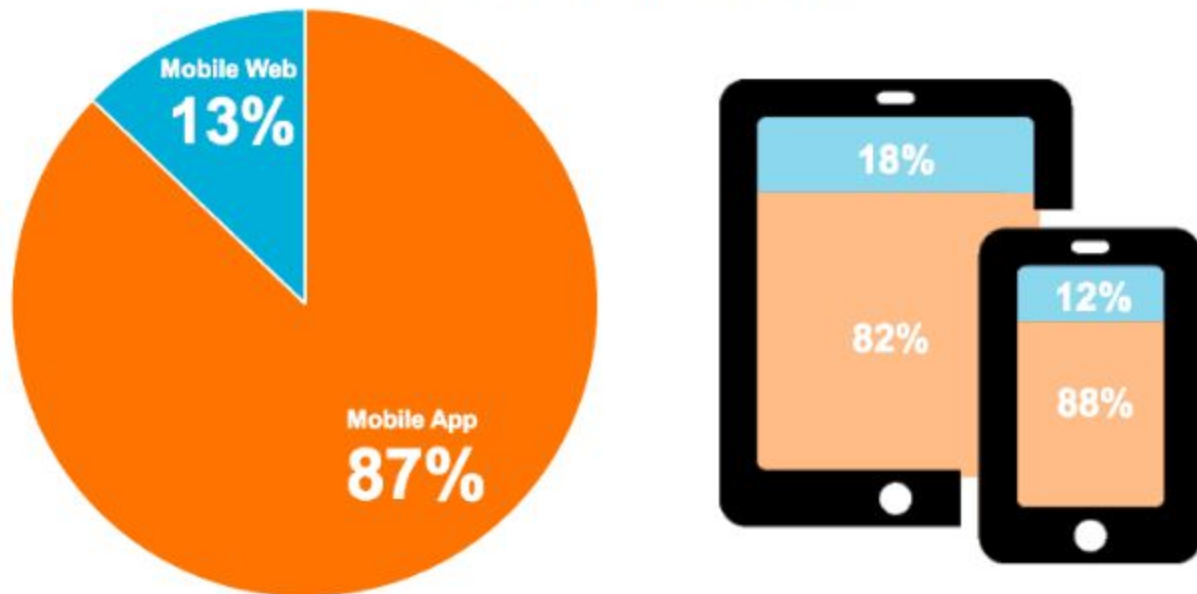
- Introducción
- Ionic
- Typescript
- Angular
- RX
- PWA
- Plugins
- Seguridad

Time Spent per Adult User per Day with Digital Media, USA,  
2008 – 2016



### Share of Time Spent on Mobile: App vs. Web

Source: comScore Mobile Metrix, U.S., Age 18+, June 2017



- En 2019 había 3,986 millones de usuarios únicos de internet móvil. El número total de usuarios activos de Internet es de 4.388 millones.
- Desde octubre de 2016, el tráfico móvil y de tabletas ha superado el tráfico de escritorio, según Statcounter.
- El 33.5% de los ingresos del Black Friday en 2018 provino de pedidos móviles
- En 2017, el 95% de los usuarios de Facebook accedieron a su cuenta a través de un dispositivo móvil.
- Más de la mitad de toda la transmisión de video proviene de un dispositivo móvil.

# Desarrollo de apps móviles

Nativo vs Multiplataforma

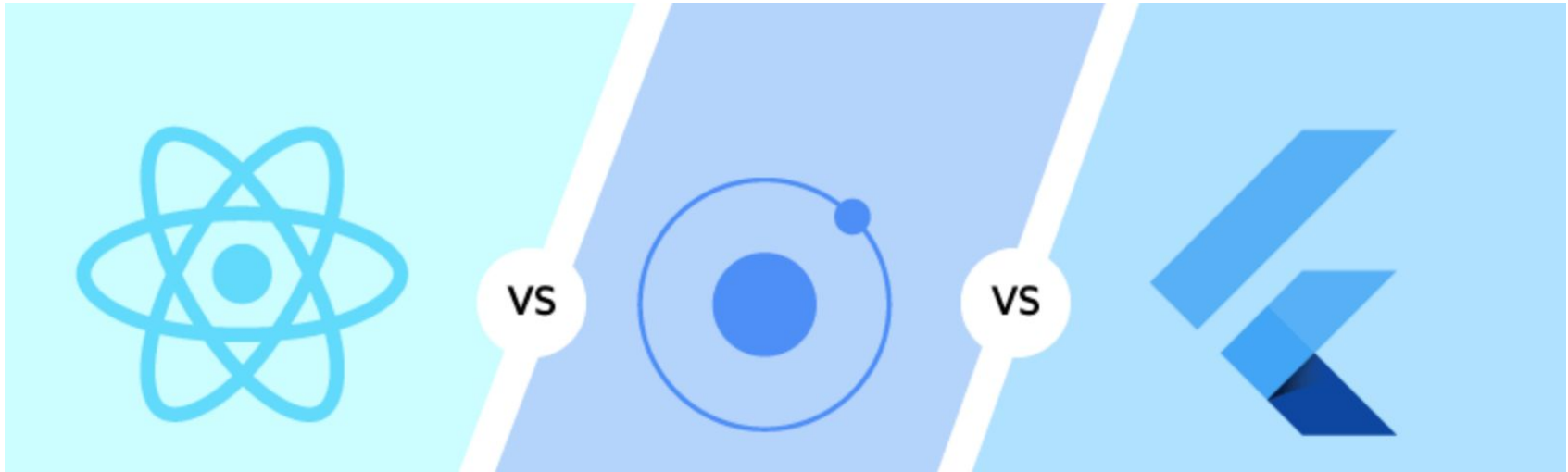
# IOS

Swift  
Objective-C

# ANDROID

Kotlin  
Java

# MULTIPLATAFORMA



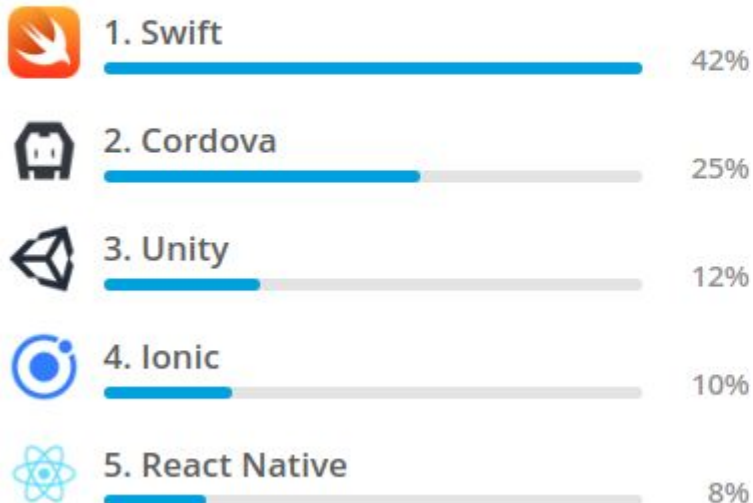


IONIC

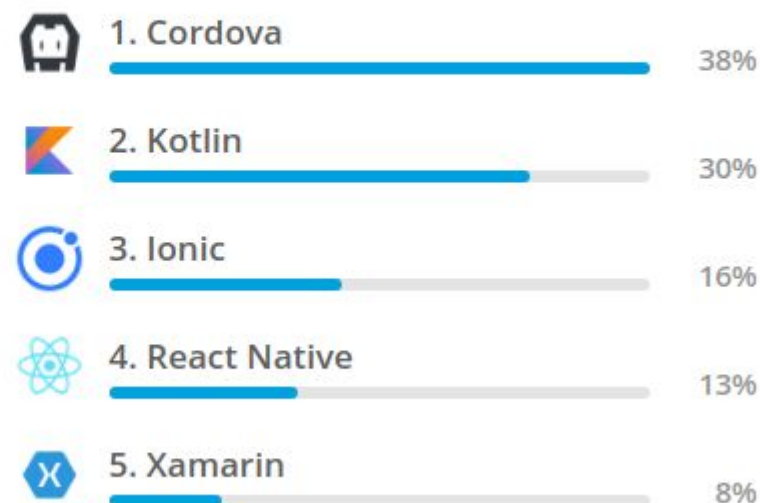


<https://ionicframework.com/customers>

## App Store



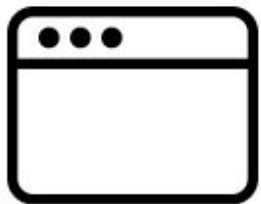
## Google Play



# Ionic framework



html, csss, javascript



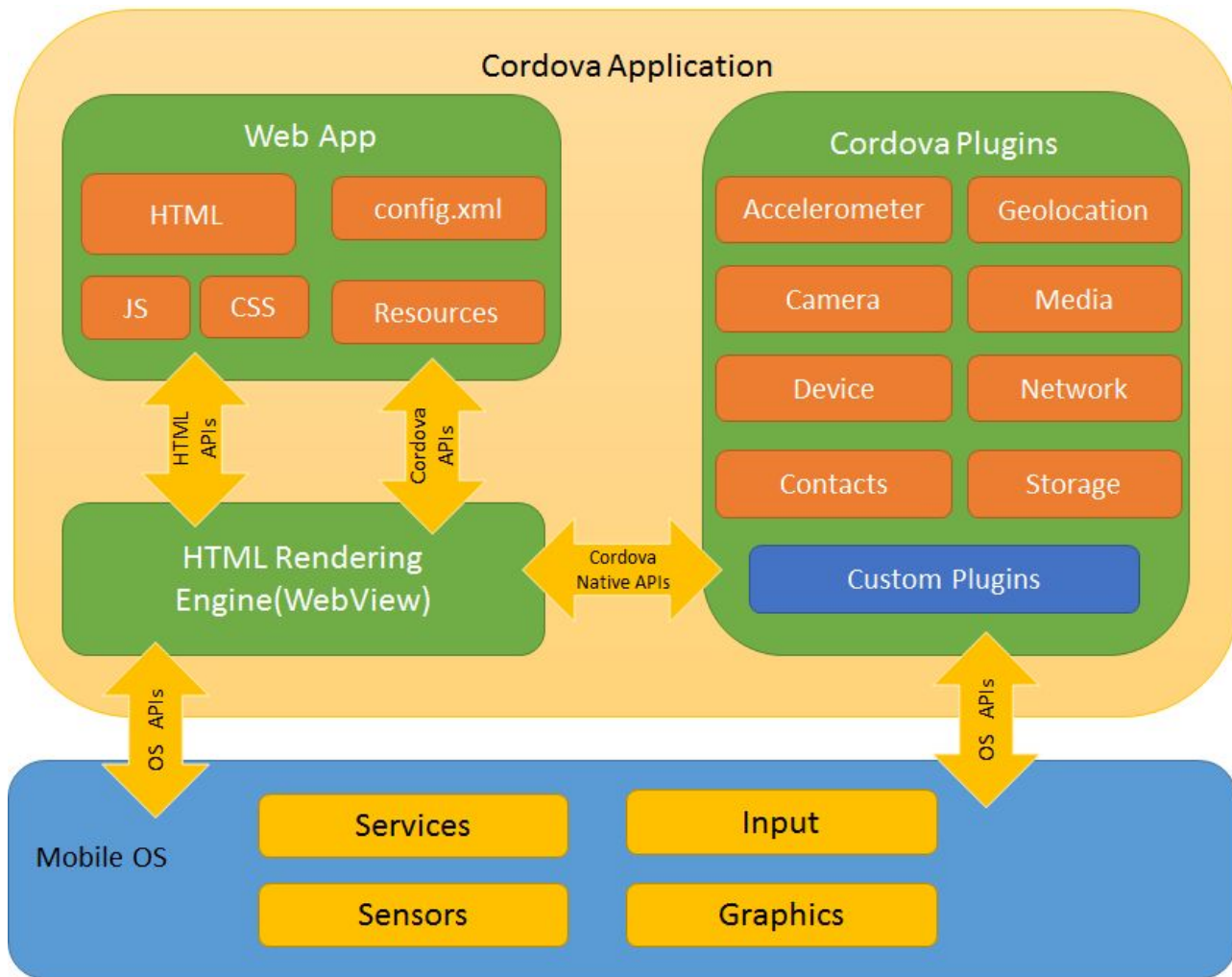
WebView (componente nativo de android, ios o navegador web)

PhoneGap

Cordova

Capacitor







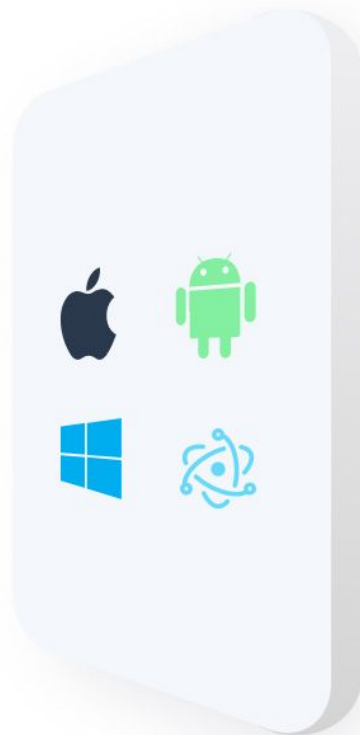
YOUR APP (ANGULAR)



UI CONTROLS (IONIC)



NATIVE ACCESS (CAPACITOR)

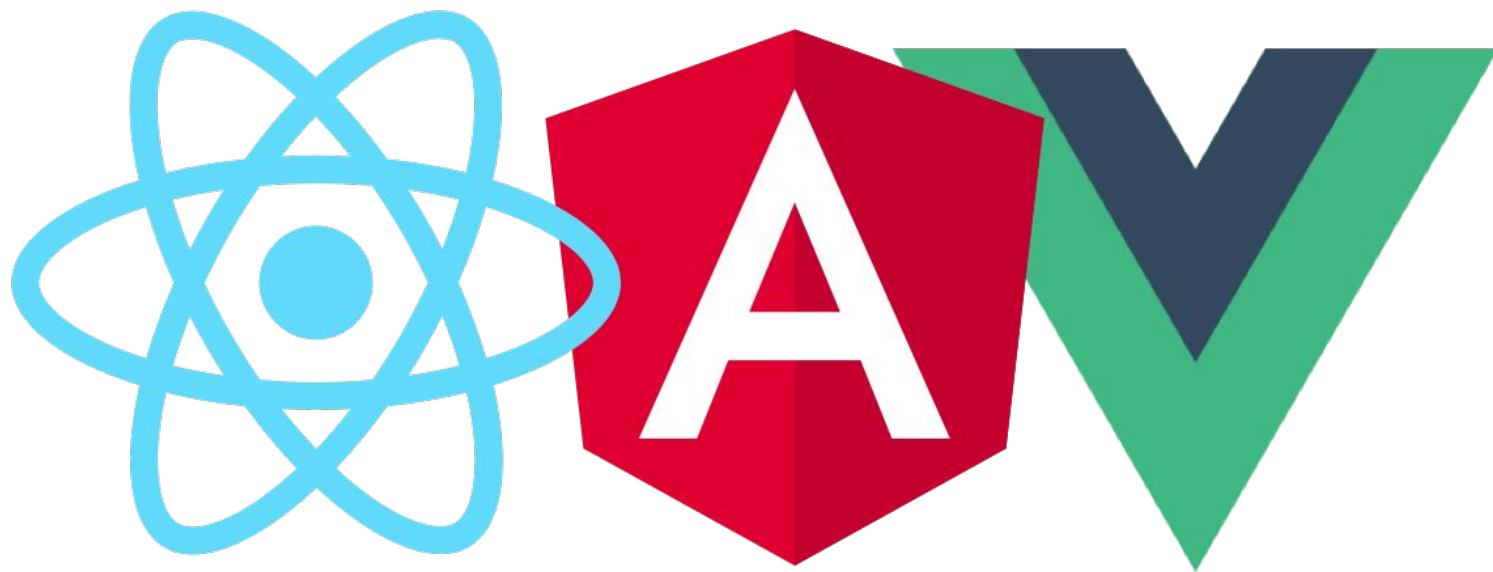


DISTRIBUTION PLATFORMS

# Capacitor

Sucesor de cordova creado con inspiración en otros proyectos cómo react native

- Cross-platform runtime
- Compatibilidad con cordova
- #UseThePlatform





# Ionic

- IonicFramework es un framework open source
- Ofrece un conjunto de herramientas que facilitan el desarrollo y las pruebas
- Se puede desarrollar para múltiples plataformas con el mismo código
- Adapta su diseño dependiendo a la plataforma en la que está corriendo
- Vamos a utilizar ionic con Angular. Angular es un framework javascript que será el encargado de de realizar la lógica, las ruta entre las pantallas y la manipulación del HTML

# Instalación del ambiente

# Node en Ubuntu

```
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -  
sudo apt-get install -y nodejs
```

```
# To compile and install native addons from npm you may also need to install build tools:  
apt-get install -y build-essential
```

Es necesario tener instalado [Node](#)

```
$ npm install -g @ionic/cli cordova
```

# Otras herramientas

- Visual Studio Code
- Google Chrome
- Postman
- Android Studio (para compilar en Android)
- Xcode (para compilar en ios)

# Extensiones para Visual Studio Code

- [AB HTML Formatter](#)
- [Angular 8 Snippets](#)
- [Angular Language Service](#)
- [Auto Rename Tag](#)
- [Color Picker](#)
- [Cordova Tools](#)
- [Debugger for Chrome](#)
- [Paste JSON as Code](#)
- [TypeScript Importer](#)

# 01-creacion-proyecto

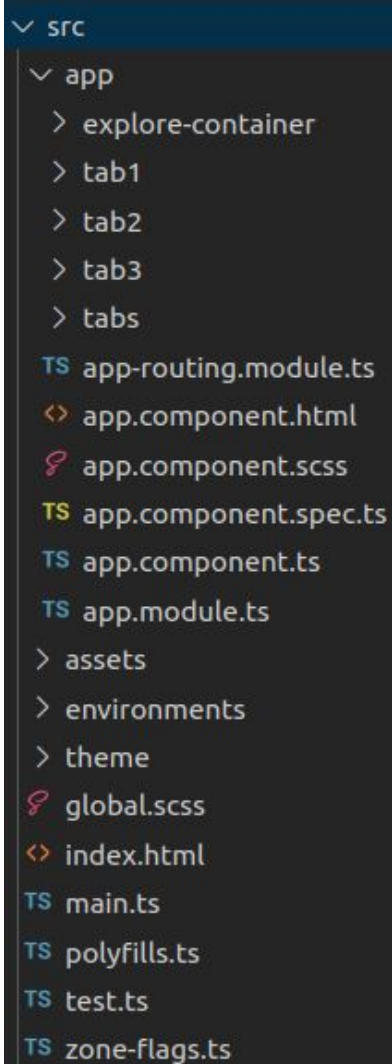
<https://github.com/terminuslabsok/curso-ionic>

```
> e2e
> node_modules
  ✓ src
    > app
      > assets
      > environments
      > theme
      global.scss
      index.html
      TS main.ts
      TS polyfills.ts
      TS test.ts
      TS zone-flags.ts
    > typings
    .gitignore
    {} angular.json
    ≡ browserslist
    ionic.config.json
    karma.conf.js
    {} package-lock.json
    {} package.json
    {} tsconfig.app.json
    tsconfig.json
    {} tsconfig.spec.json
    {} tslint.json
```

# Estructuras del proyecto

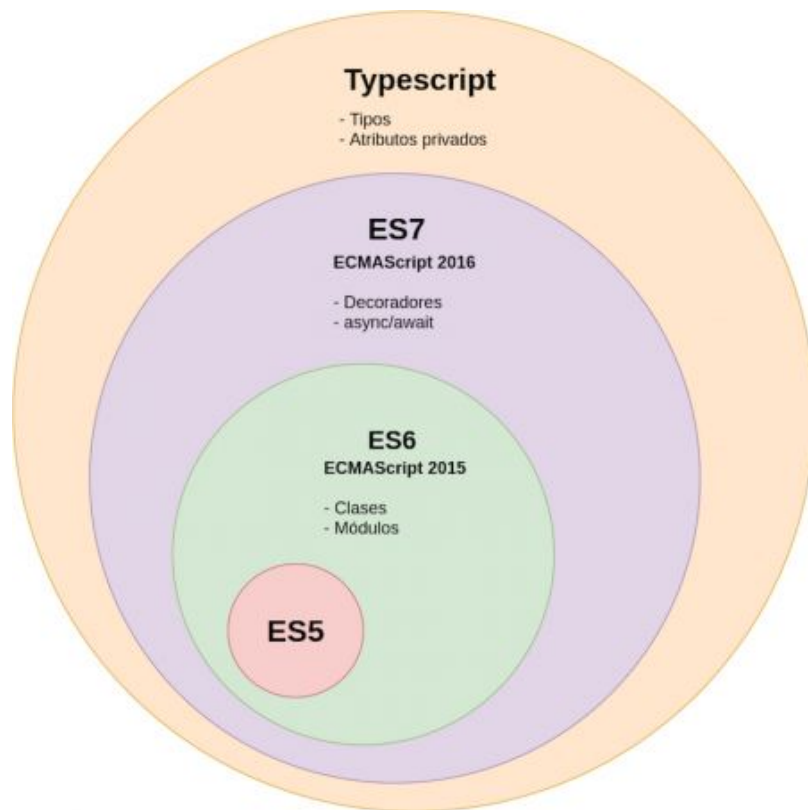
- e2e: sirve para compilar y levantar una app Angular y luego ejecuta pruebas de extremo a extremo utilizando Protractor.
- **node\_modules**: dependencias del proyecto.
- platforms: plataformas de córdova donde va a correr la app
- plugins: plugins de córdova
- www: aplicación compilada pronta para desplegar
- resources: recursos según plataforma
- angular.json: configuración del proyecto angular
- browserlist: Define los navegadores destino dónde se ejecutará
- ionic.config.json: integración con herramientas de angular
- **package.json**: Configura las dependencias de los paquetes npm del proyecto.
- tsconfig.json: archivo de configuración de typescript
- tslint.json: las reglas de codificación del proyecto





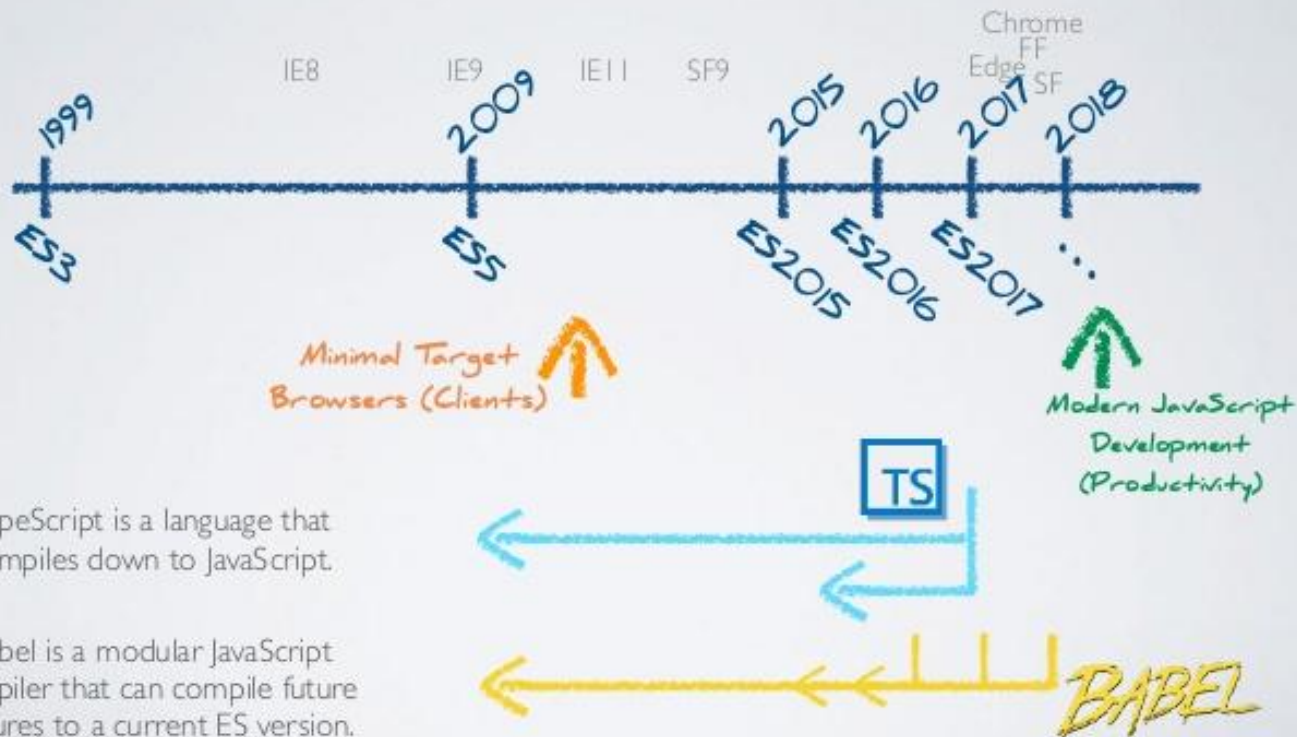
- **src/app**: carpeta de la aplicación angular, con todos los componentes, directivas, servicios, etc.
- **src/app/app-routing.module.ts**: configuración de las rutas
- **src/app/app.component.\*** : componente principal
- **src/app/app.module.ts**: módulo global
- **src/assets**: recursos estáticos, imágenes, etc
- **src/environments**: Configuración de variables según ambientes
- **src/theme**: tema de estilos de ionic
- **src/global.scss**: estilos globales
- **src/index.html**: archivo que carga nuestra app SPA
- **src/polyfills.ts**: compatibilidad con los navegadores antiguos

# TypeScript



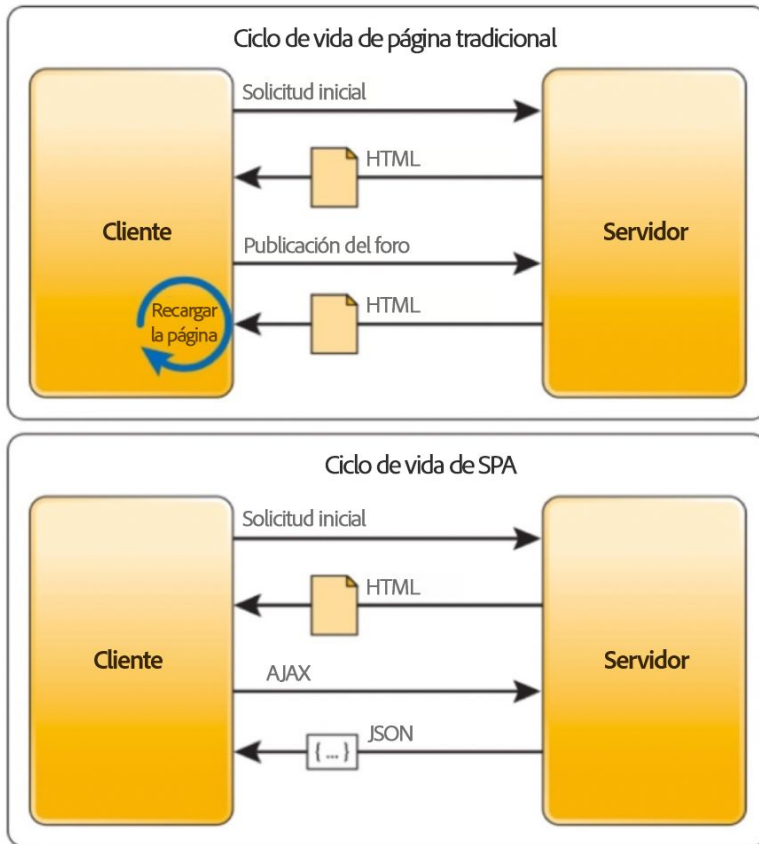
# Addressing the Feature Gap

JavaScript has evolved rapidly in the past few years.

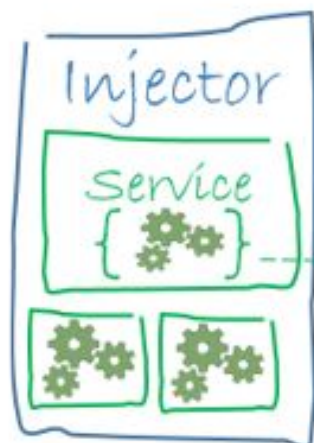


DEMO

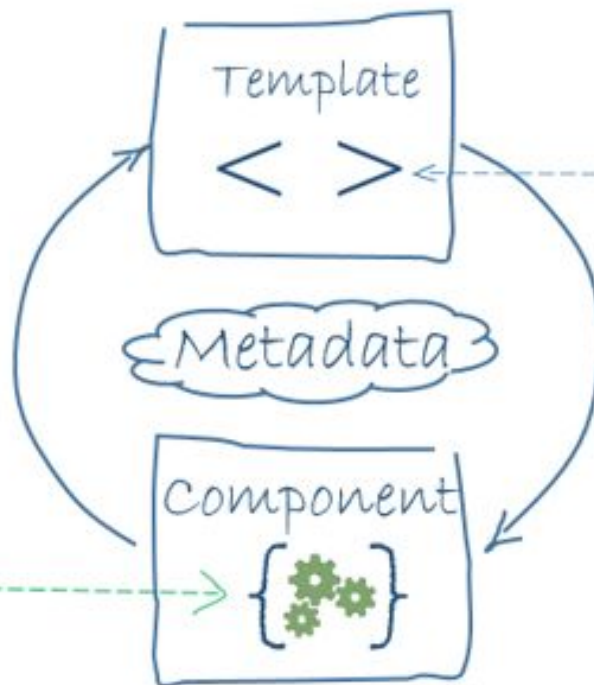
# Angular



Module Component { }	Module Service { }
Module value 3.1415	Module Fn $\lambda$



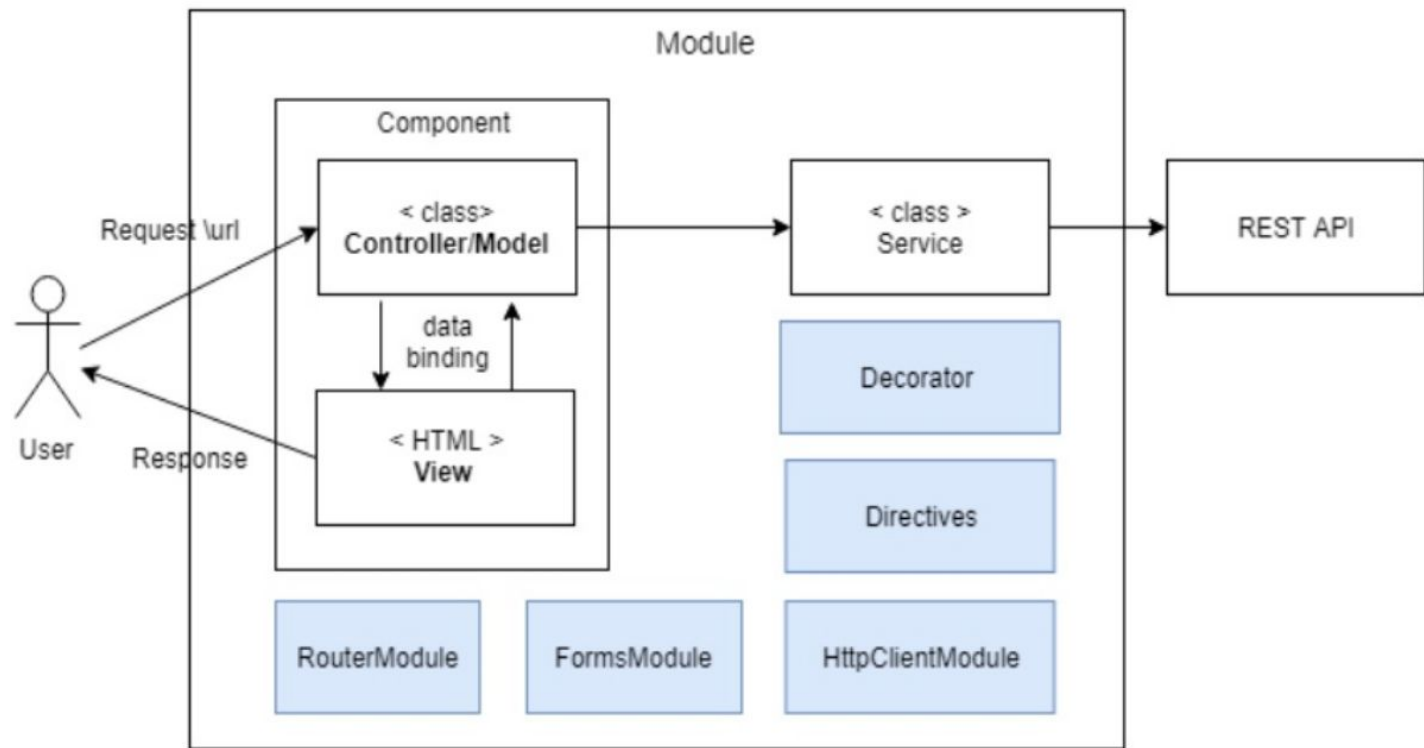
Property  
Binding



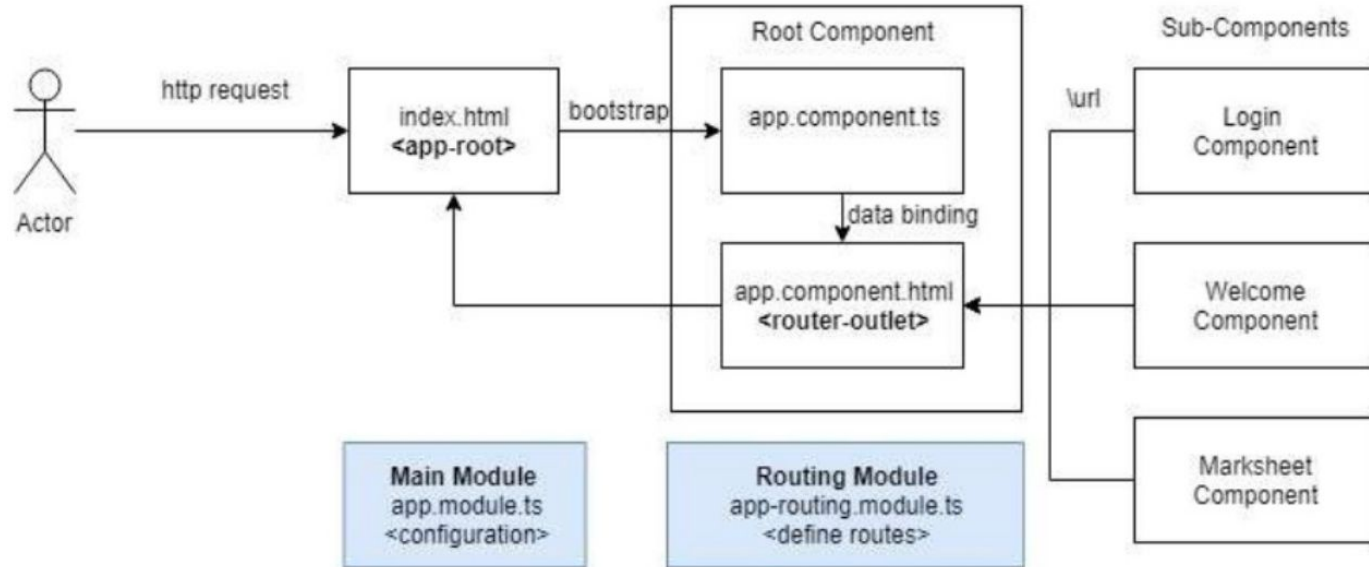
Event  
Binding



# MVC



# Project flow





# Modules

```
import { FormsModule } from '@angular/forms';
import { AppRoutingModule } from '../app-routing.module';
@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,
    WelcomeComponent
  ],
  imports: [
    AppRoutingModule,
    FormsModule,
  ],
  providers: [
    UserService,
    MarksheetService,
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

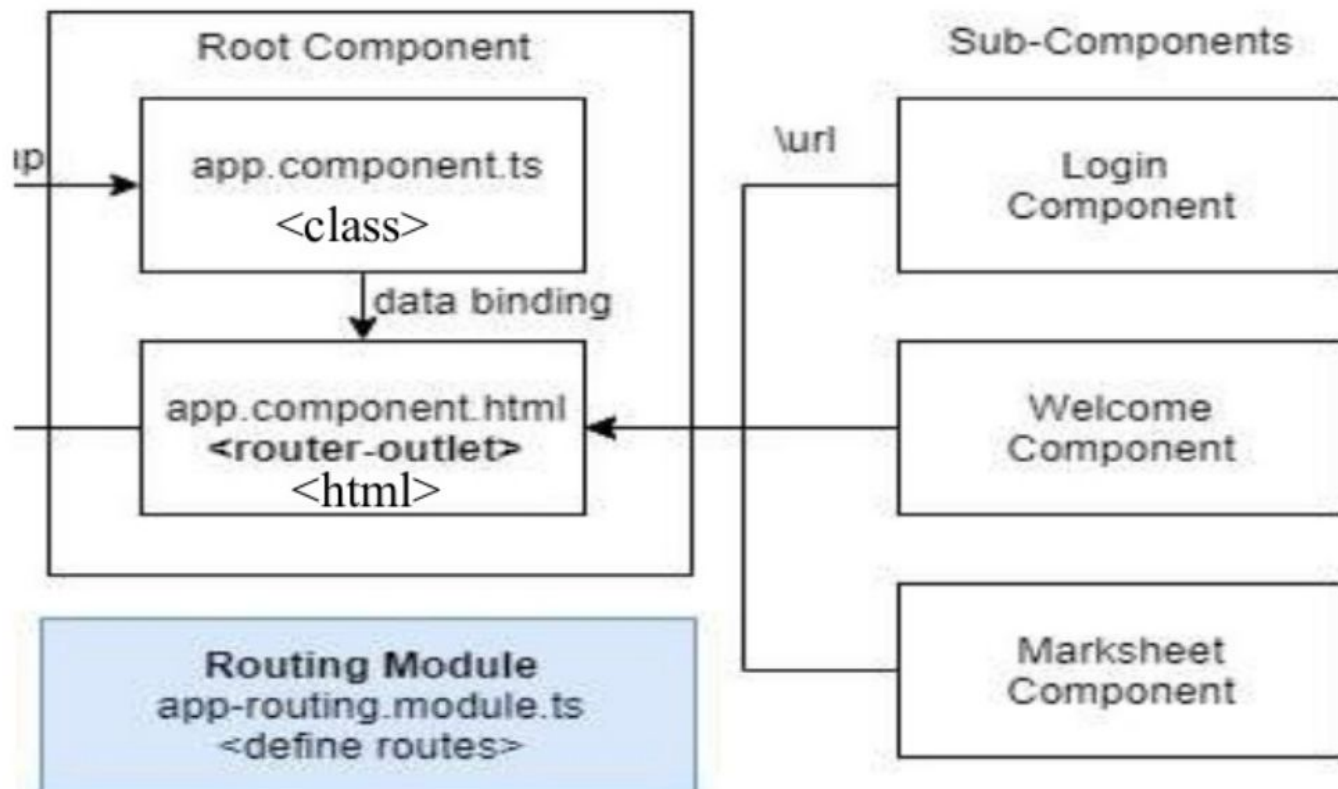
The diagram illustrates the components of an Angular module definition. Yellow callout boxes point to specific parts of the code:

- Component**: Points to the `declarations` array, which lists the components managed by this module.
- Modules**: Points to the `imports` array, which lists the modules on which this module depends.
- Services**: Points to the `providers` array, which lists the services provided by this module.
- Root Component**: Points to the `bootstrap` array, which lists the components that can bootstrap the application.
- Module Class**: Points to the `export class AppModule { }` line, which defines the module class.

# Componentes

- Un componente es creado por página
- Se pueden crear componente usando el comando : `ionic g component nombre-componente`
- 1 componente contiene 4 archivos
  - Controlador .ts
  - Vista .html
  - Estilos .scss
  - Test Unitarios
- Los componentes se configuran dentro de los modulos

## @Component



# Directivas

- son usadas para cambiar la estructura del DOM de la pagina
- angular tiene muchas directivas predefinidas, cómo \*ngFor o \*ngIf
- Podemos crear nuestras propias directivas
- Hay 4 tipos de directiva
  - Component directives
  - Structural directives
  - Attribute directives
  - Custom Directives

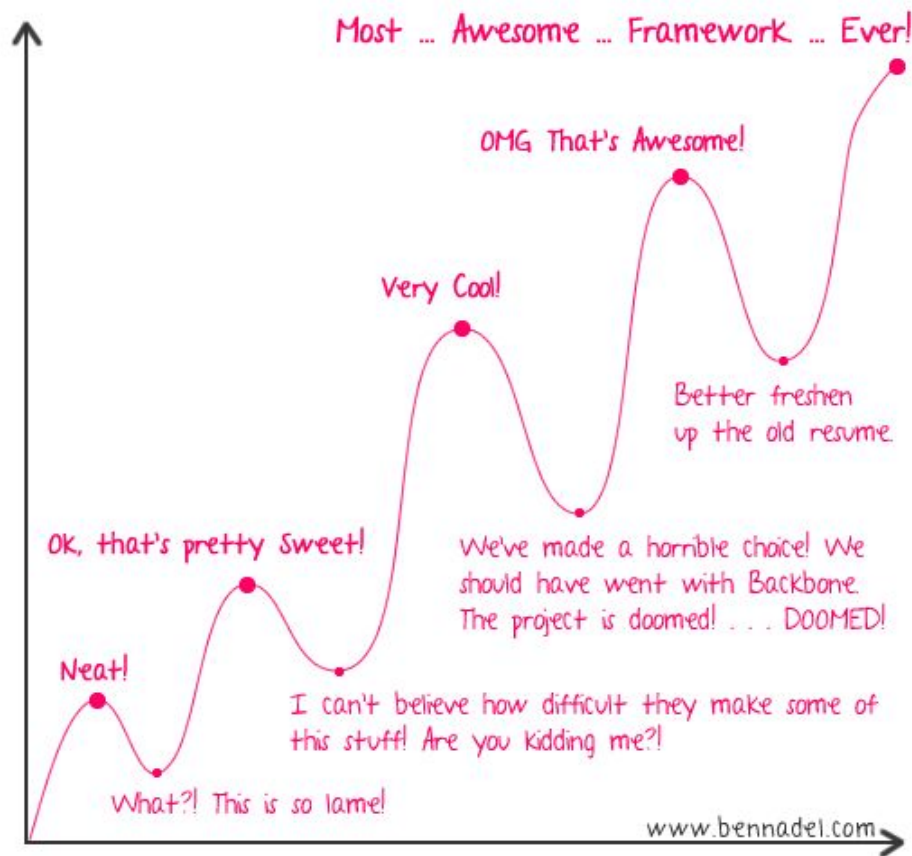
# Pipes

- son usados para dar formato a los datos
- El caracter pipe | es usado para aplicar el pipe a un atributo
- Ejemplo
  - {{ name | uppercase }}
  - {{ name | lowercase }}

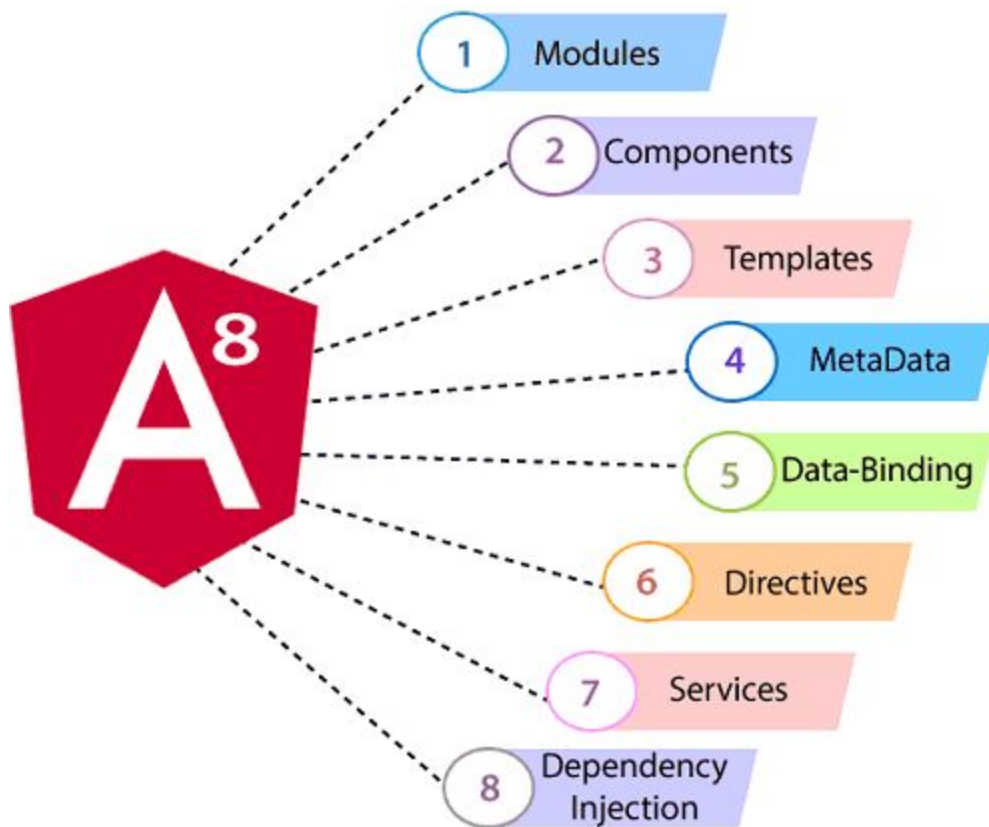
# Services

- Los services contienen logica de negocio, que es común a múltiples componentes
- En general los services contienen llamas REST con operaciones CRUD
- Los servicios son inyectados en los controladores a través de la inyección de dependencia

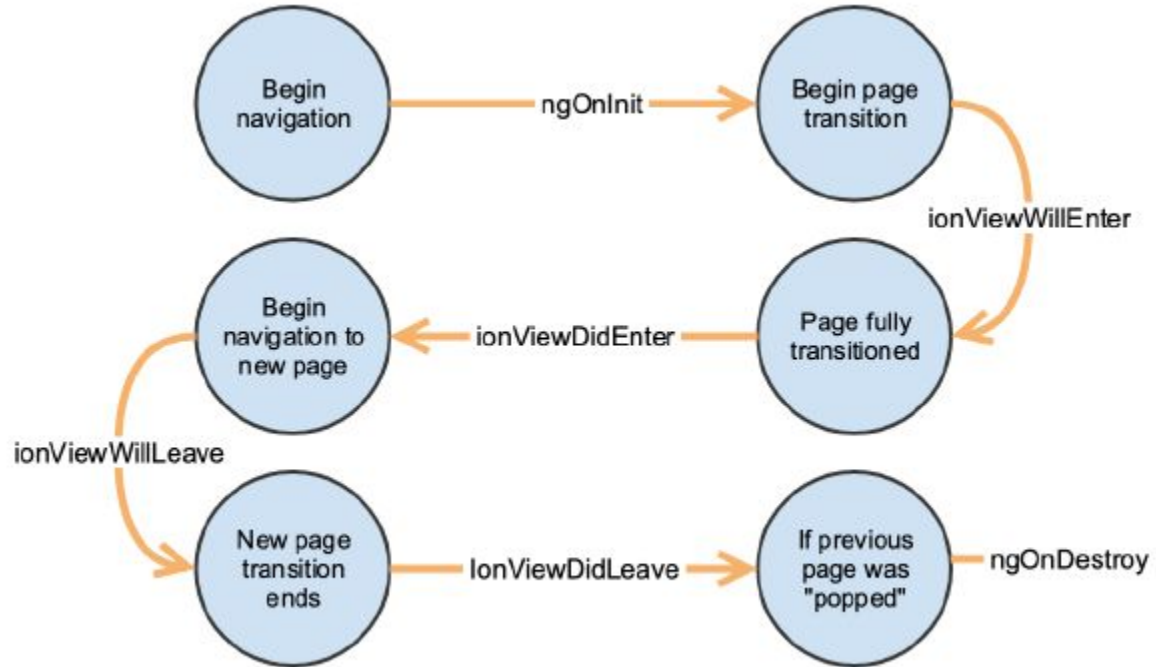
Angular







# Ciclo de vida



page 1

Go to page 2



Console

What's New



top

Page 1 ngOnInit

Page 1 ionViewWillEnter

Page 1 ionViewDidEnter



02-angular

debugger;