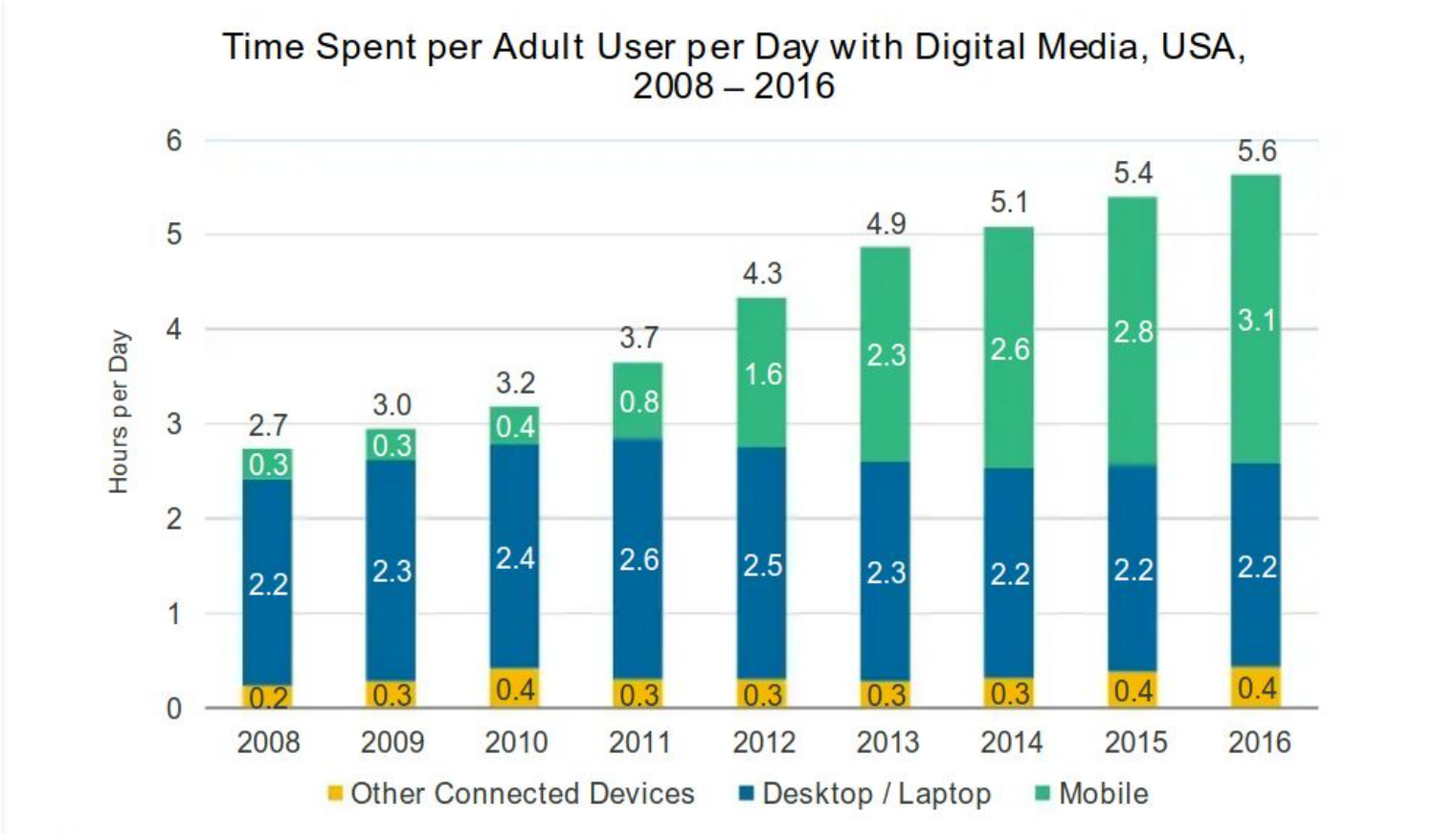
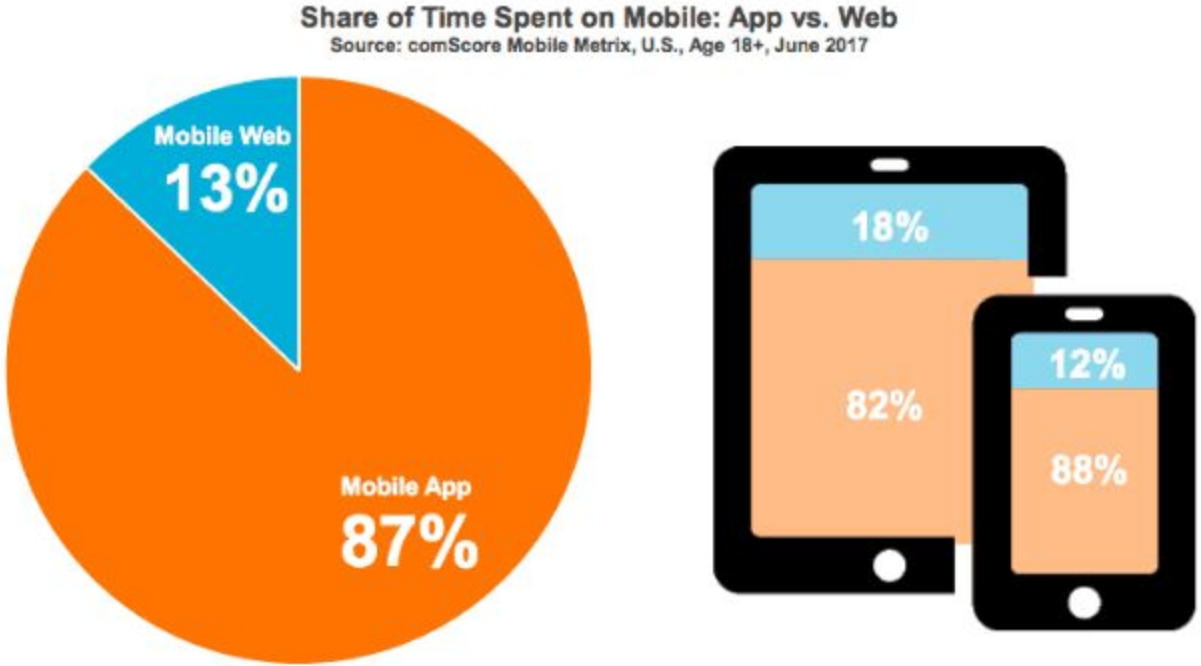
Desarrollo de aplicaciones móviles con Ionic

Agenda

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



Fuente: [Mary Meeker]



Fuente: ComScore

- 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

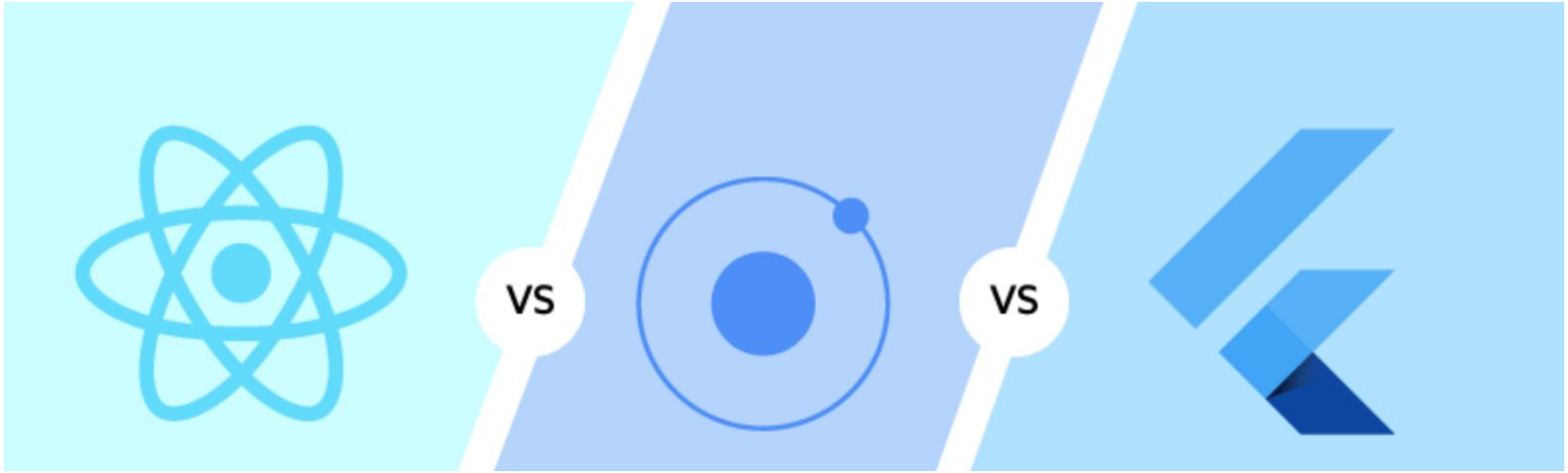
IOS

Swift
Objective-C

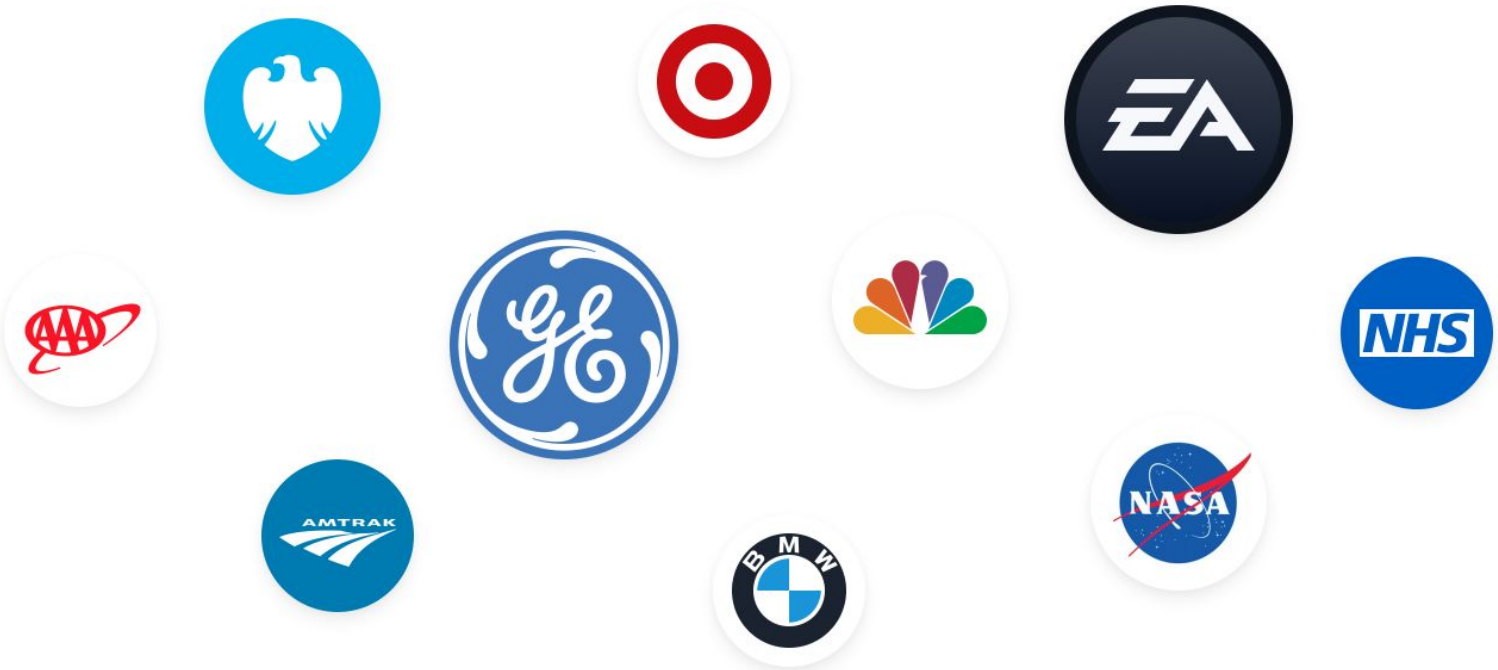
ANDROID

Kotlin
Java

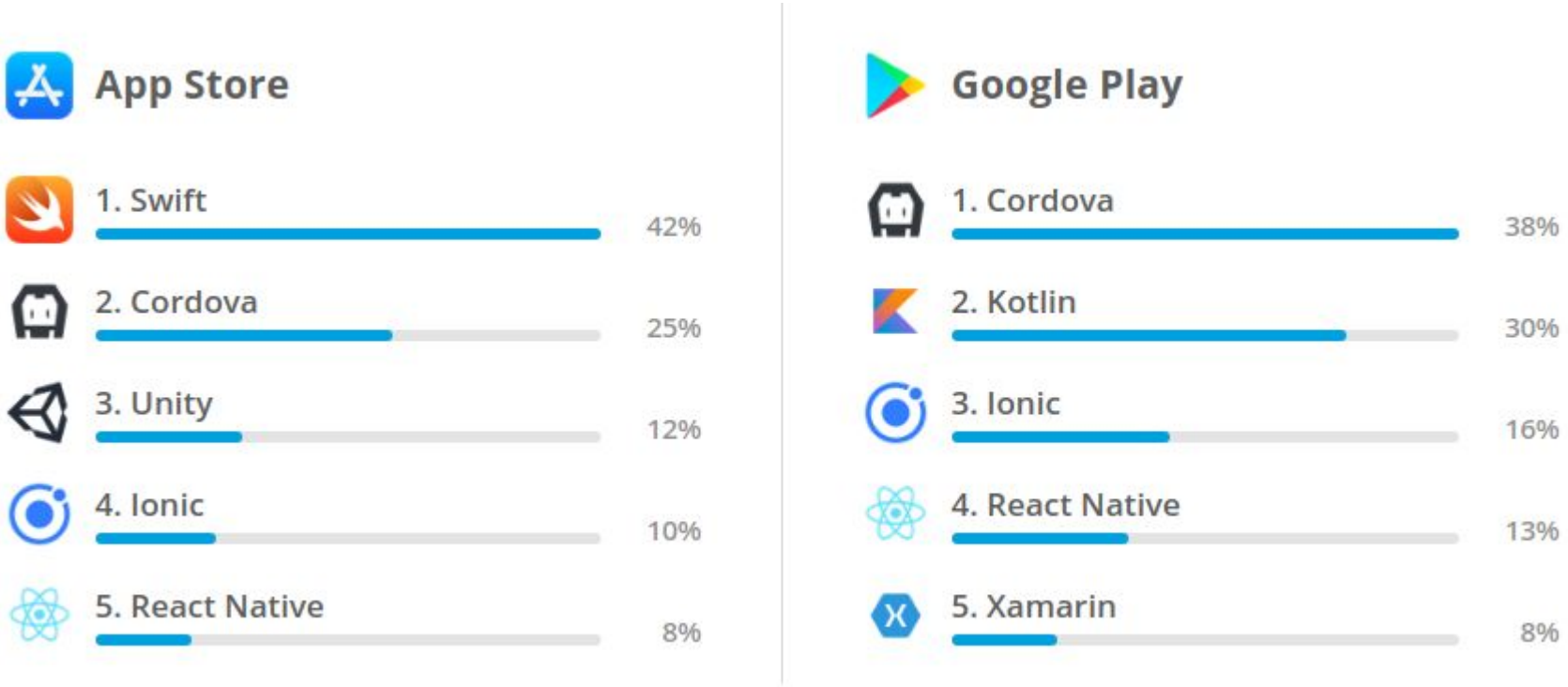
MULTIPLATAFORMA



IONIC



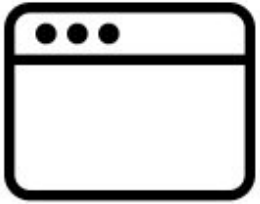
https://ionicframework.com/customers



https://appfigures.com/top-sdks/development/all

Ionic framework

html, csss, javascript

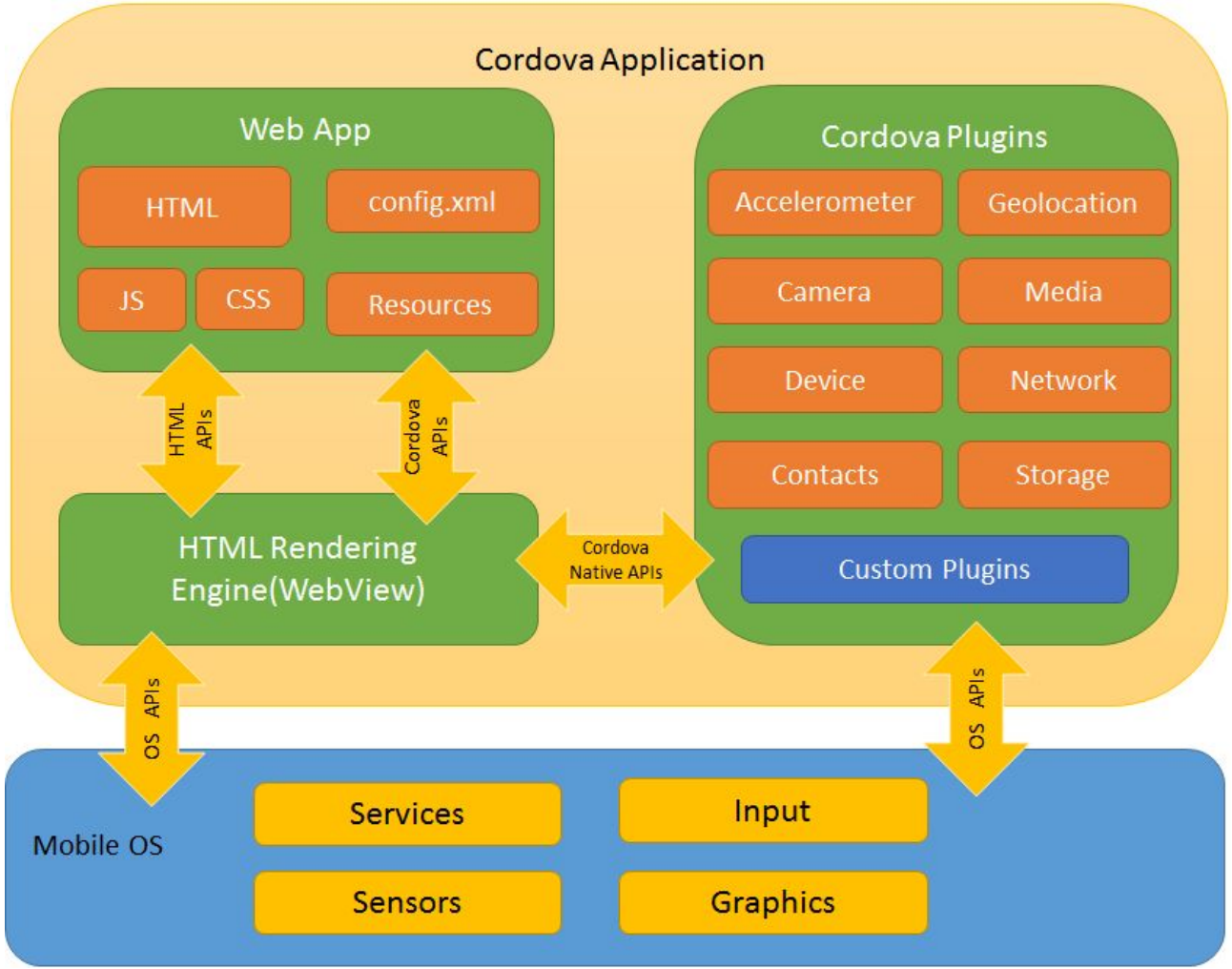


WebView (componente nativo de android, ios o navegador web)

PhoneGap

Cordova

Capacitor

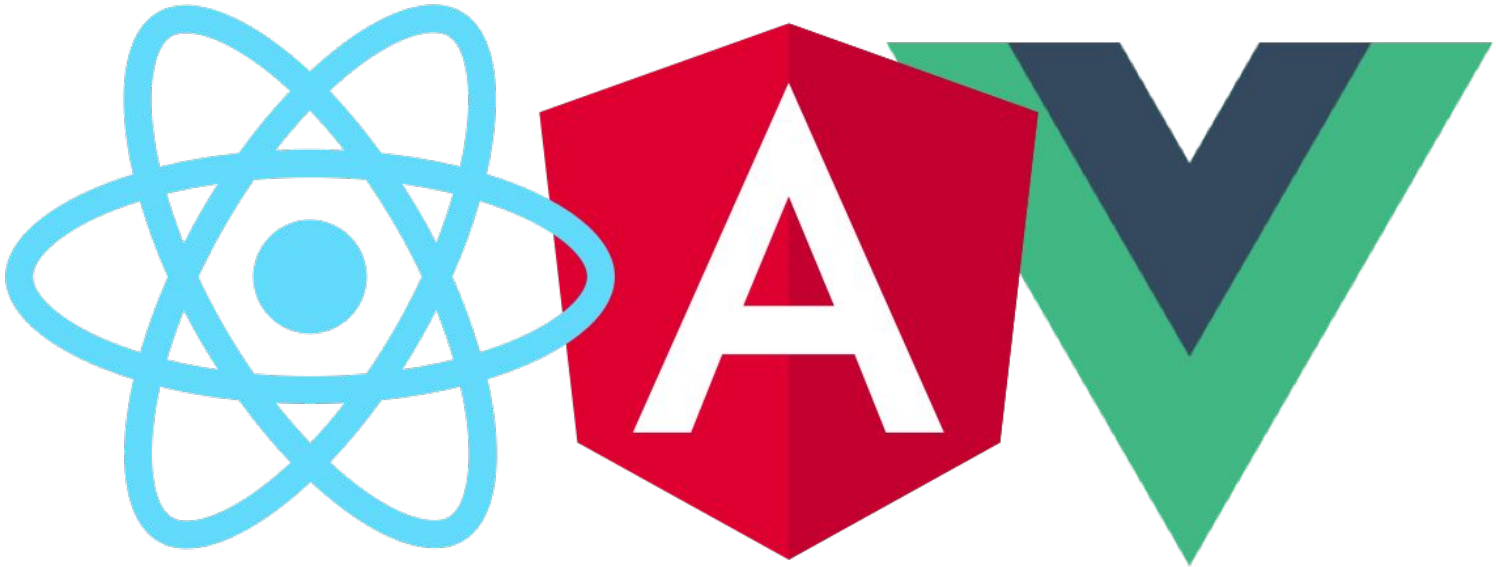

Cordova Application
Web App
HTML
config.xml
JS
CSS
Resources
Cordova Plugins
Accelerometer
Geolocation
Camera
Media
Device
Network
Contacts
Storage
Custom Plugins
HTML APIs
Cordova APIs
HTML Rendering Engine(WebView)
Cordova Native APIs
OS APIs
OS APIs
Mobile OS
Services
Input
Sensors
Graphics


HTML
CSS
JS
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
- #UseThePlatfom



Ionic

- IonifFramework 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

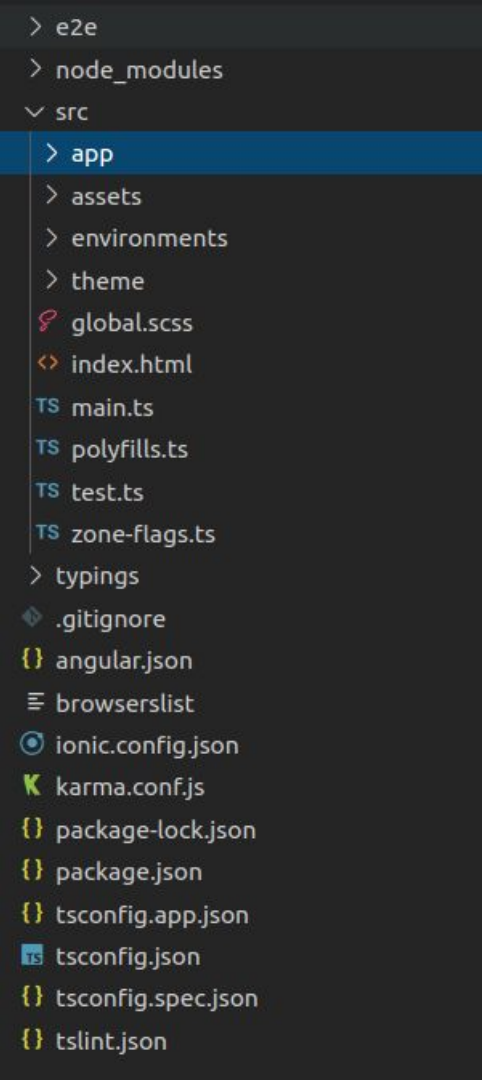
- Visual Studio Code
- Google Chrome
- Postman
- Android Studio (para compilar en Android)
- Xcode (para compilar en ios)

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

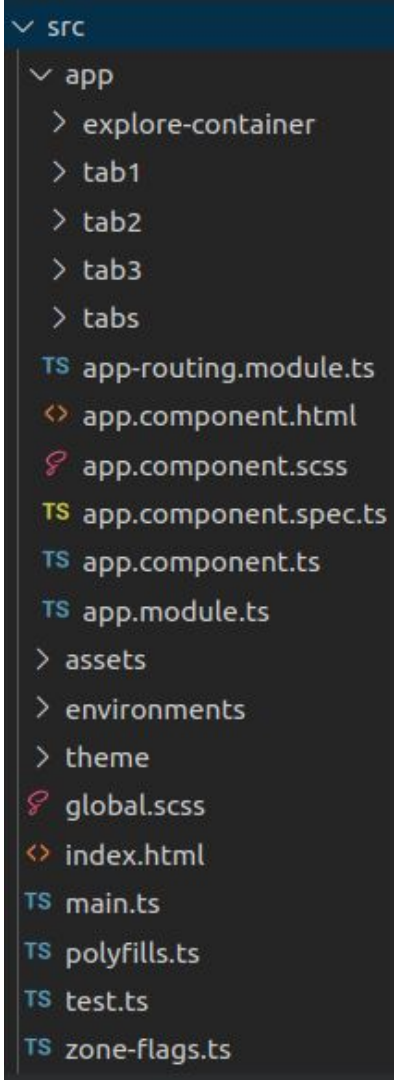
01-creacion-proyecto

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



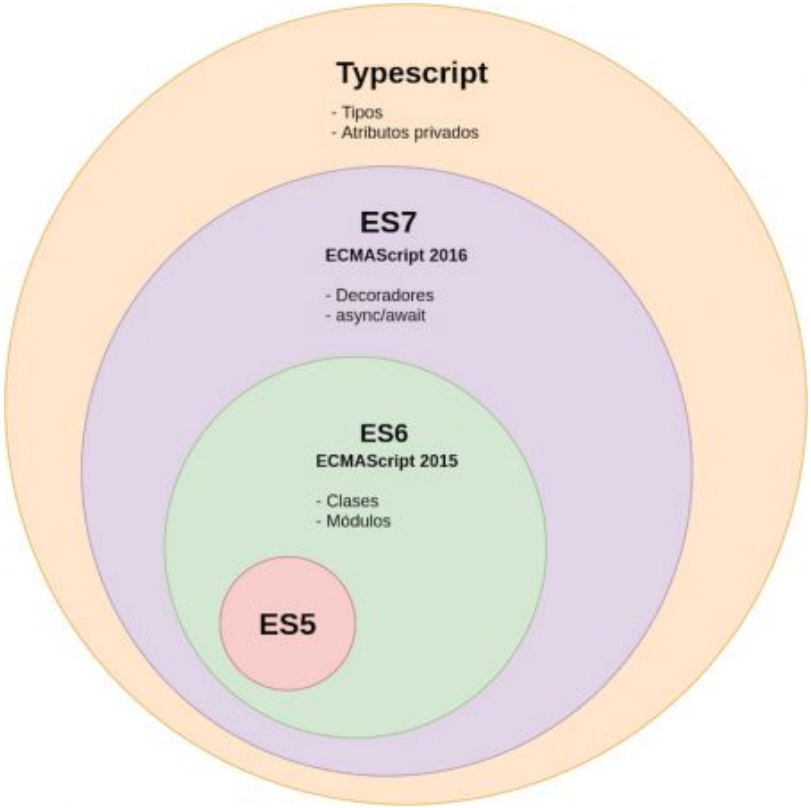
Estructuras del proyecto

- 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 esporadores destino dónde se ejecutará
- ionic.config.json: integracón con herramientas de angular
- **package.json**: Configura las dependencias del 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


Typescript
- Tipos
- Atributos privados
ES7
ECMAScript 2016
- Decoradores
- async/await
ES6
ECMAScript 2015
- Clases
- Módulos
ES5

Addressing the Feature Gap

JavaScript has evolved rapidly in the past few years.

Minimal Target Browsers (Clients)

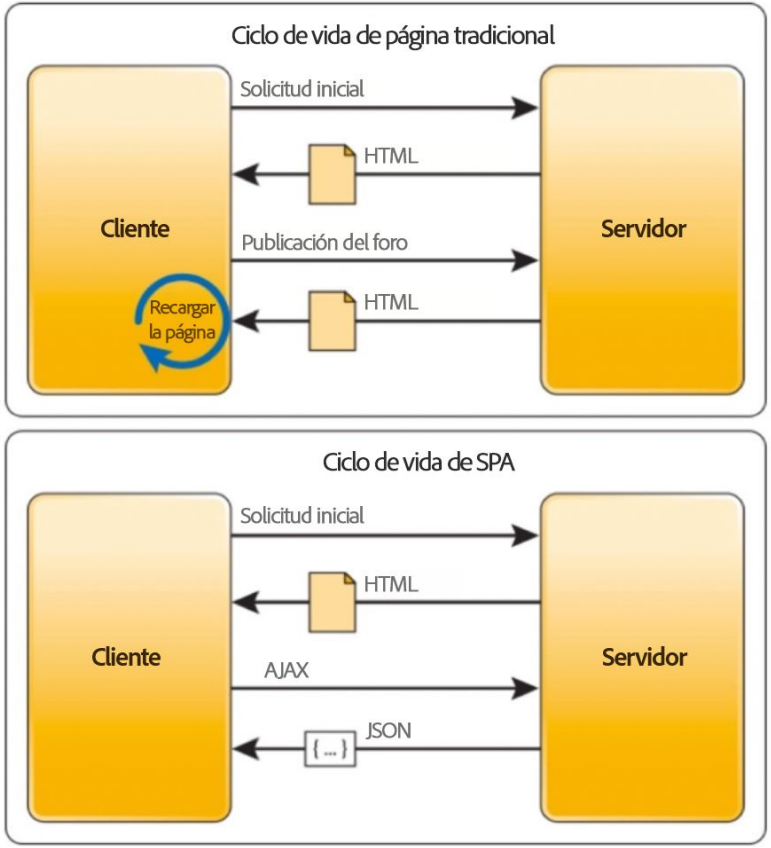
Modern JavaScript Development (Productivity)

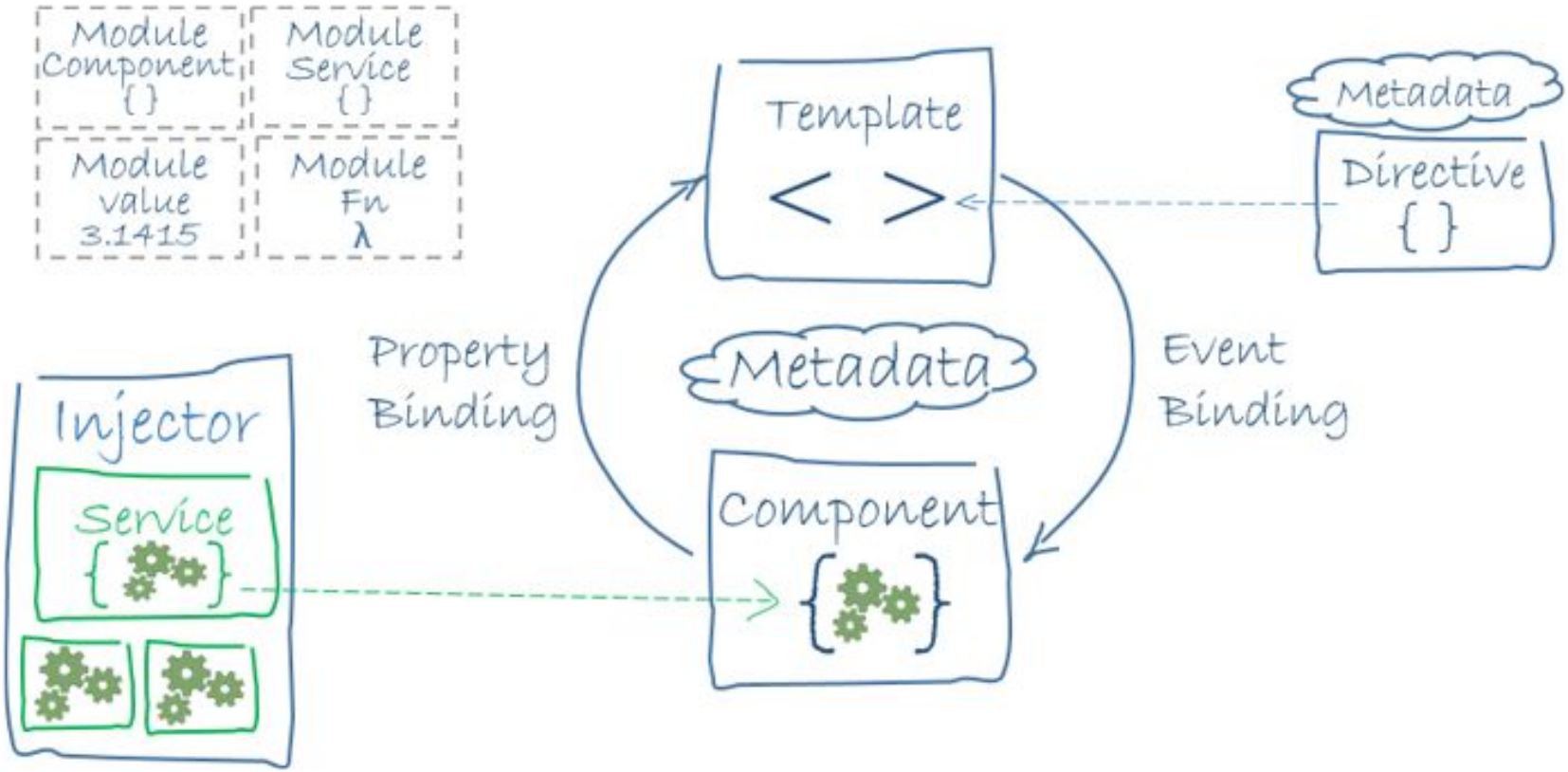
TypeScript is a language that compiles down to JavaScript.

Babel is a modular JavaScript compiler that can compile future features to a current ES version.

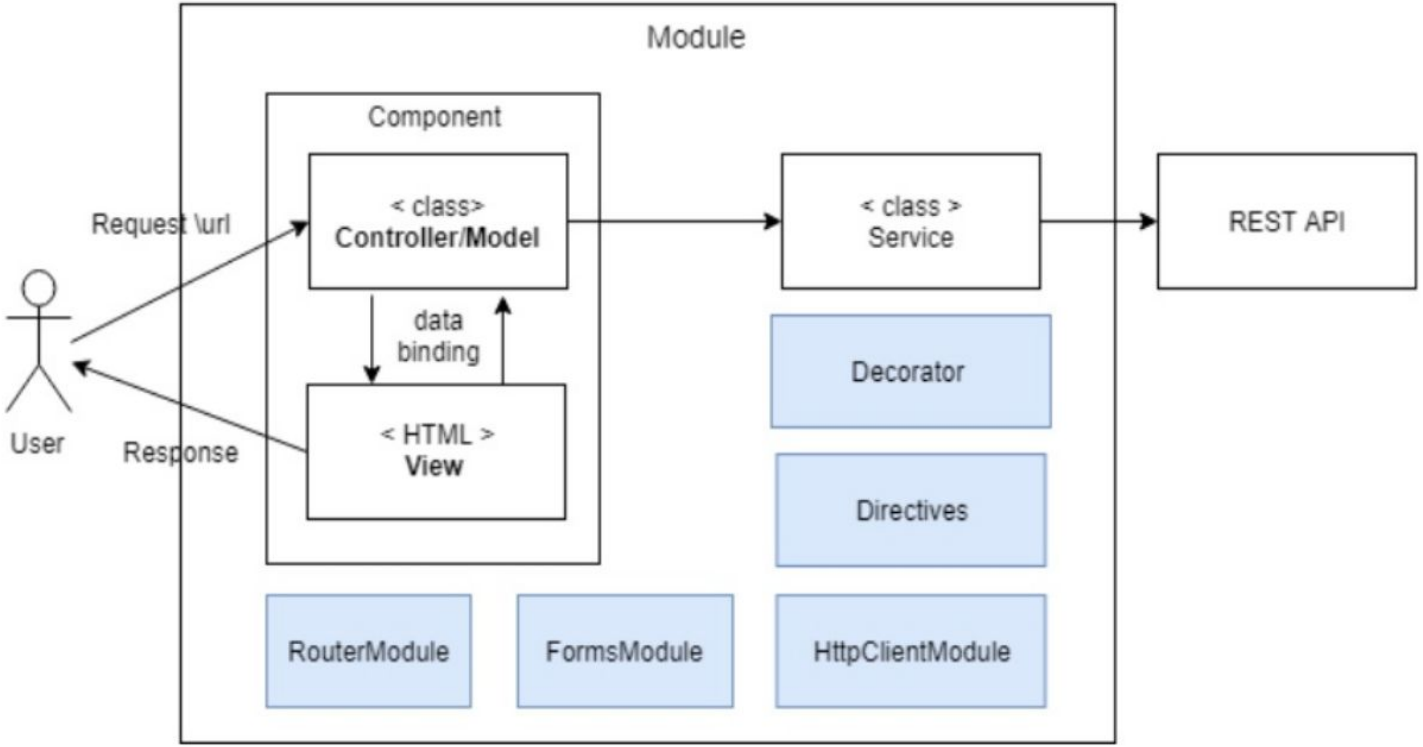
DEMO

Angular

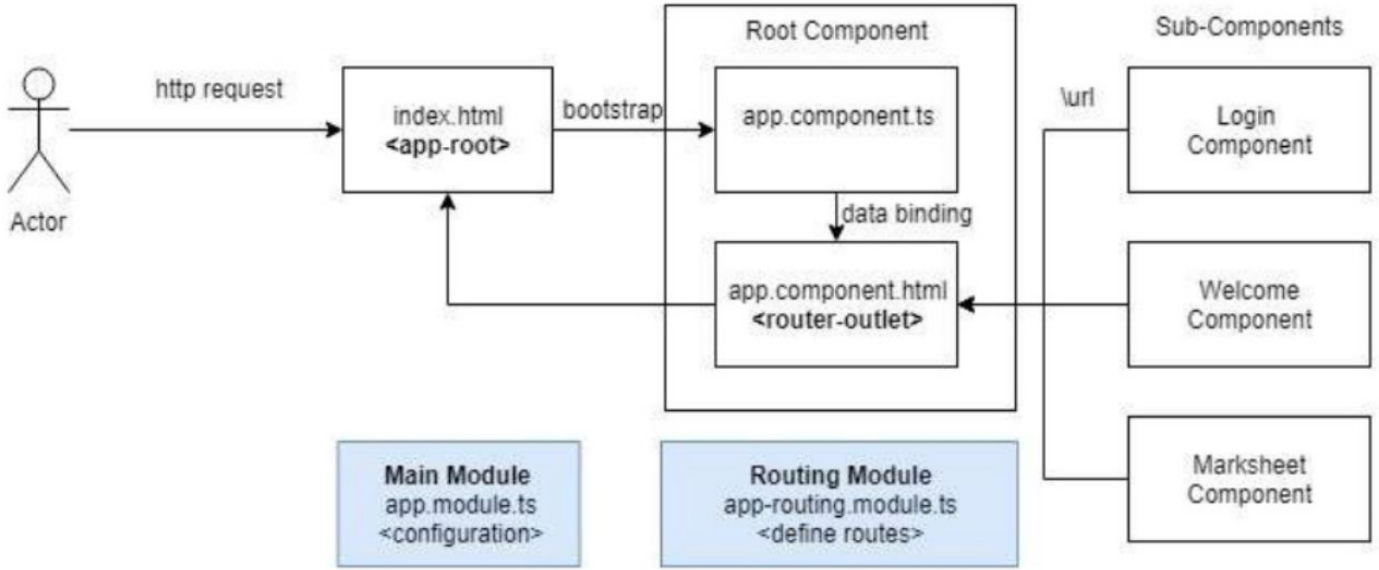

Ciclo de vida de página tradicional
Cliente
Servidor
Solicitud inicial
HTML
Publicación del foro
HTML
Recargar la página
Ciclo de vida de SPA
Cliente
Servidor
Solicitud inicial
HTML
AJAX
JSON
{ ... }


Module
Component
{}
Module
Service
{}
Module
value
3.1415
Module
Fn
λ
Template
< >
Metadata
Directive
{}
Property
Binding
Metadata
Event
Binding
Injector
Service
Component

MVC


Module
Component
< class>
Controller/Model
Request \url
data
binding
< HTML >
View
User
Response
< class >
Service
REST API
Decorator
Directives
RouterModule
FormsModule
HttpClientModule

Project flow



Modulos

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

Component

Modules

Services

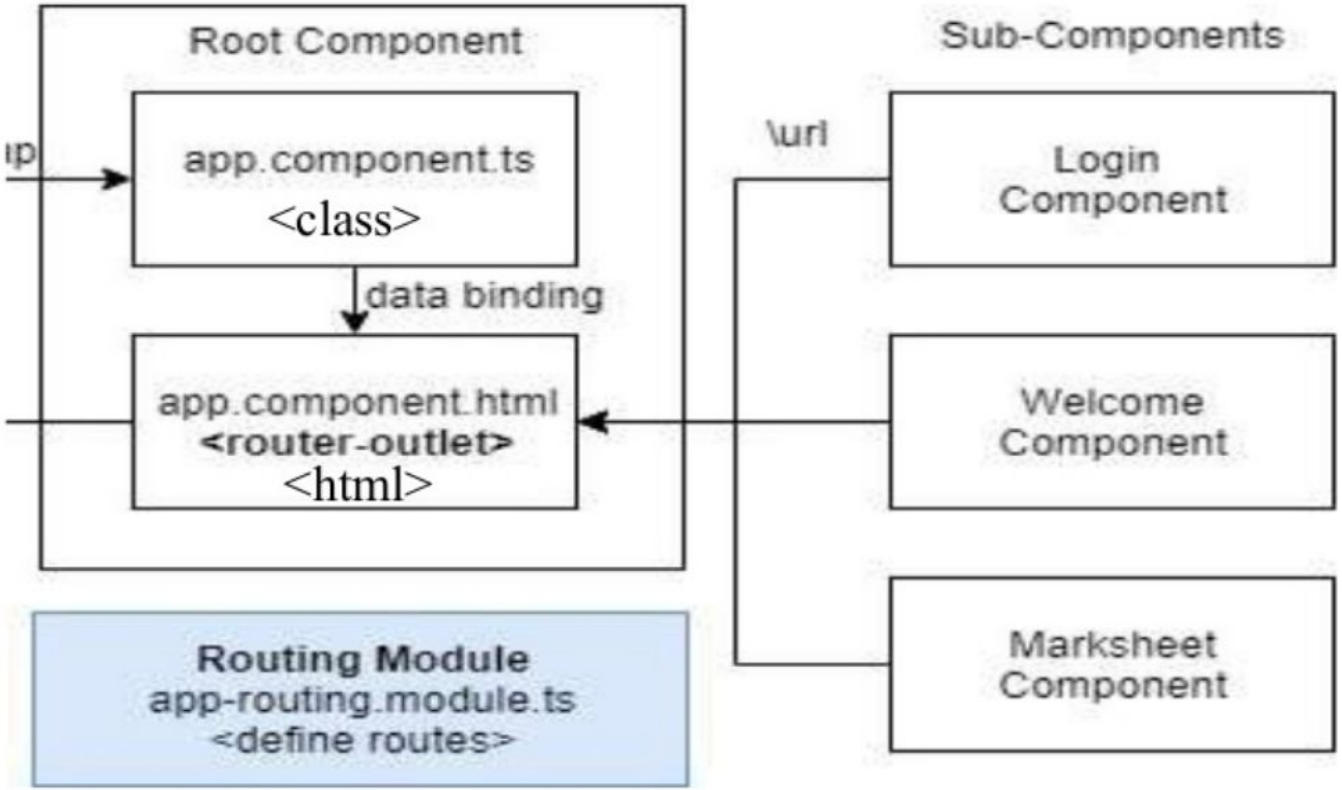
Root Component

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 Unitatios
- Los componentes se configuran dentro de los modulos

@Component


Root Component
Sub-Components
ip
app.component.ts
<class>
\url
Login
Component
data binding
app.component.html
<router-outlet>
<html>
Welcome
Component
Marksheet
Component
Routing Module
app-routing.module.ts
<define routes>

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
  - Componetns directives
  - Structural directives
  - Attribute directives
  - Custom Directives

Pipes

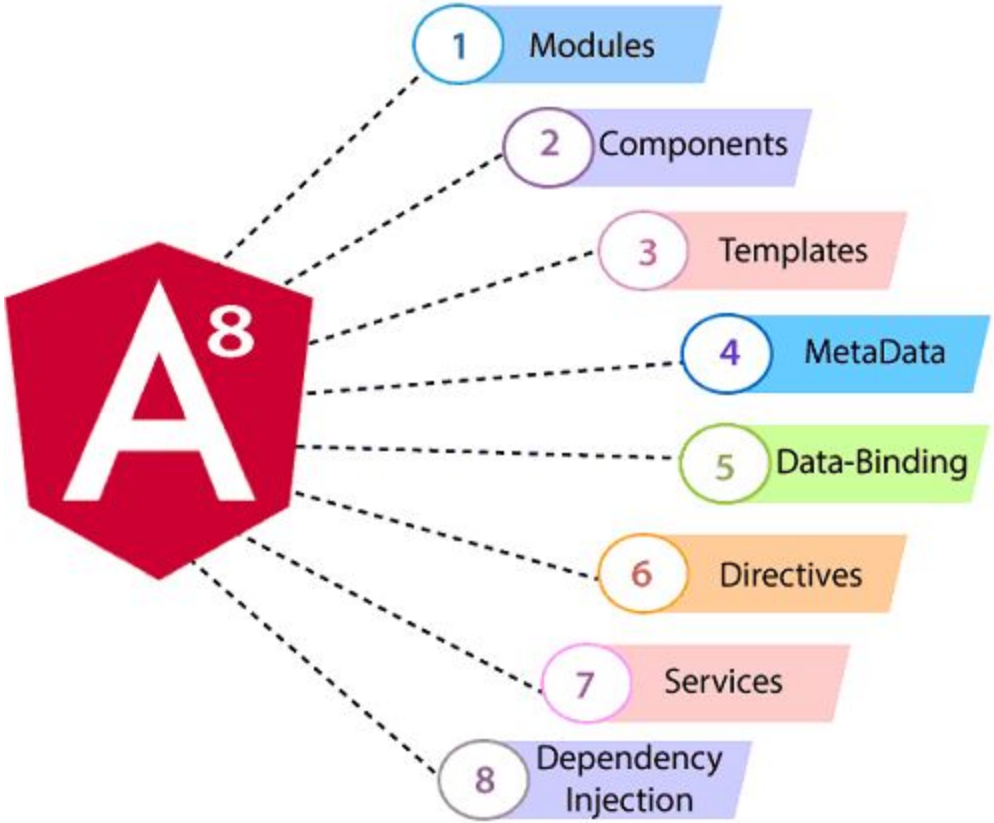
- son usados para dar formato a los datos
- El caracter pipe | es usado para aplicar el pipe a un atributo
- Ejemplo
  - {{ name | uppercase }}
  - {{ name | lowercase }}

Services

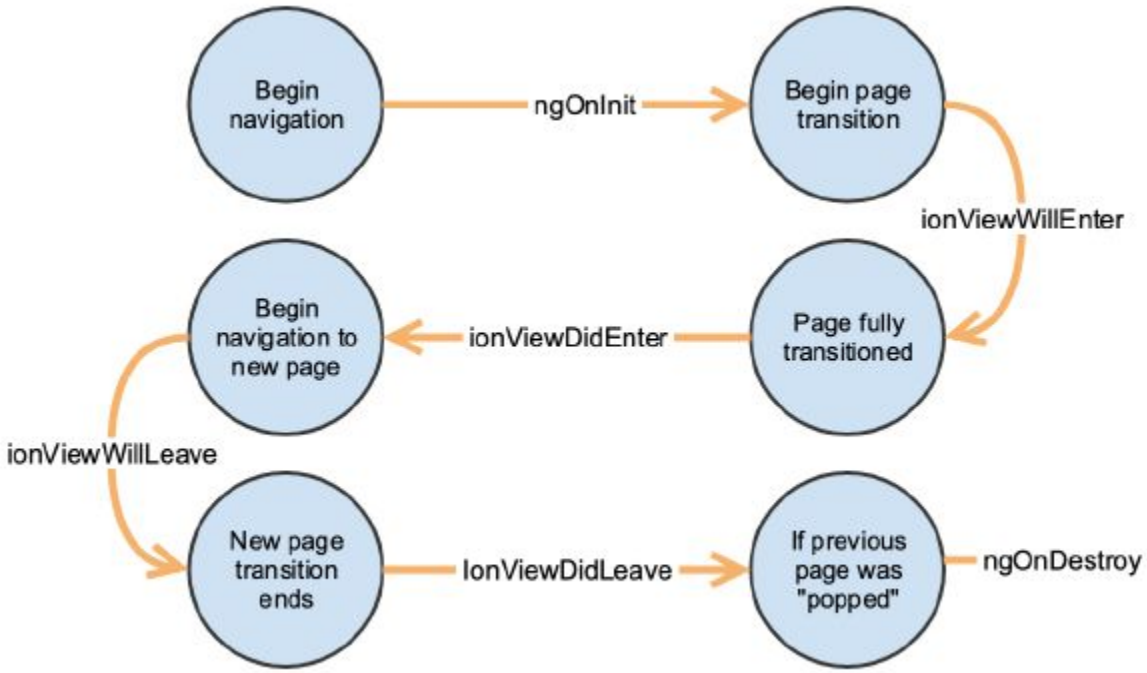
- Los services contienen logica de negocio, que es común a múltiples componentes
- En general los services contienen llamas REST con operaciones CRUD
- Los servicios son inyectados en los controladores a través de la inyección de dependencia

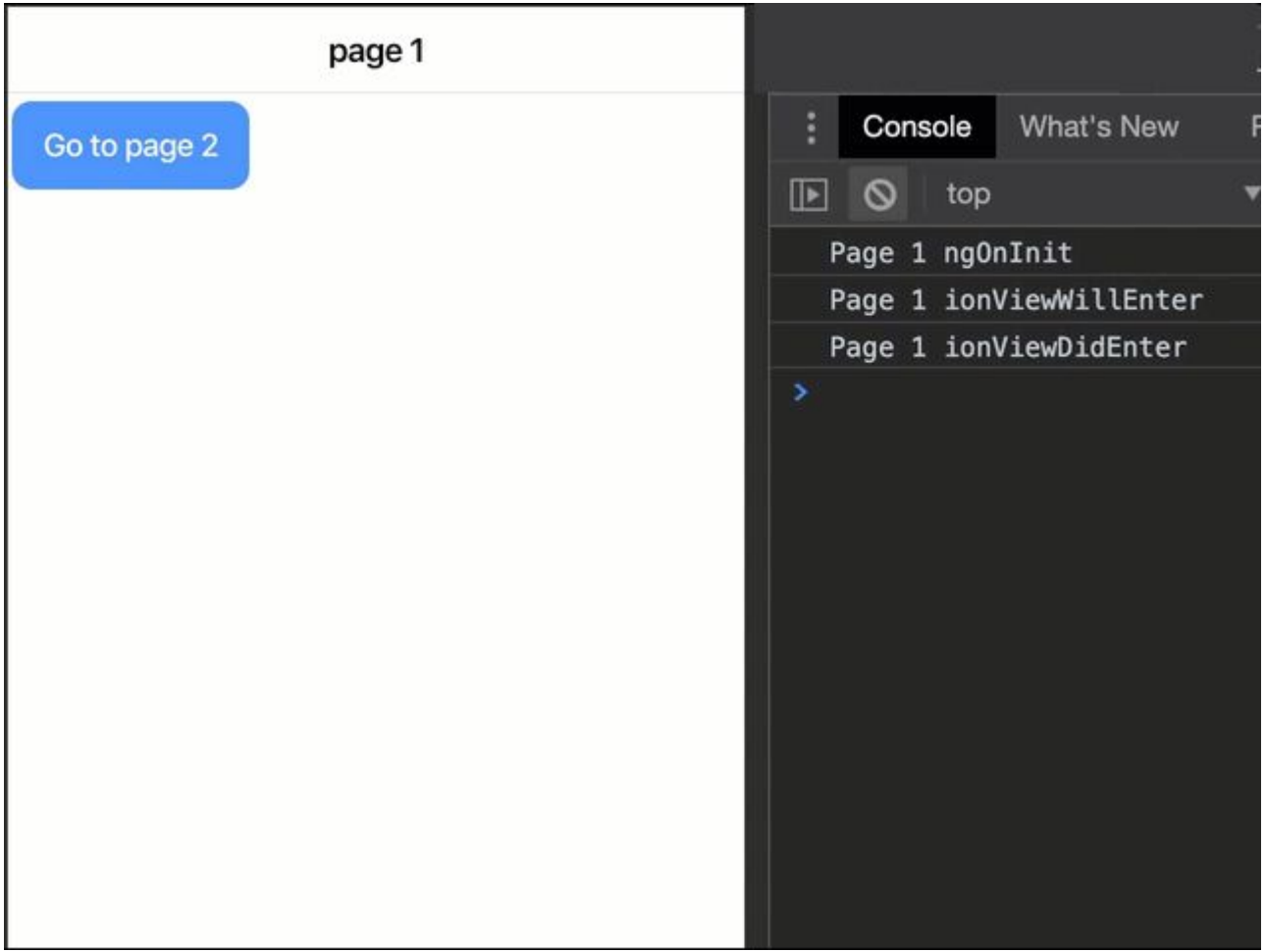
Angular


Most ... Awesome ... Framework ... Ever!
OMG That's Awesome!
Very Cool!
Better freshen up the old resume
OK, that's pretty Sweet!
We've made a horrible choice! We should have went with Backbone. The project is doomed! . . . DOOMED!
Neat!
I can't believe how difficult they make some of this stuff! Are you kidding me?!
What?! This is so lame!
www.bennadel.com


A8
1 Modules
2 Components
3 Templates
4 MetaData
5 Data-Binding
6 Directives
7 Services
8 Dependency Injection

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;
```

Correr aplicación en dispositivo móvil

Android:

- Instalar Android Studio
- Agregar la variable ANDROID_SDK_ROOT al path

iOS:

- Instalar Xcode

```
$ionic cordova platform add android
$ionic run android
```

Usando capacitor

```
ionic start --capacitor
```

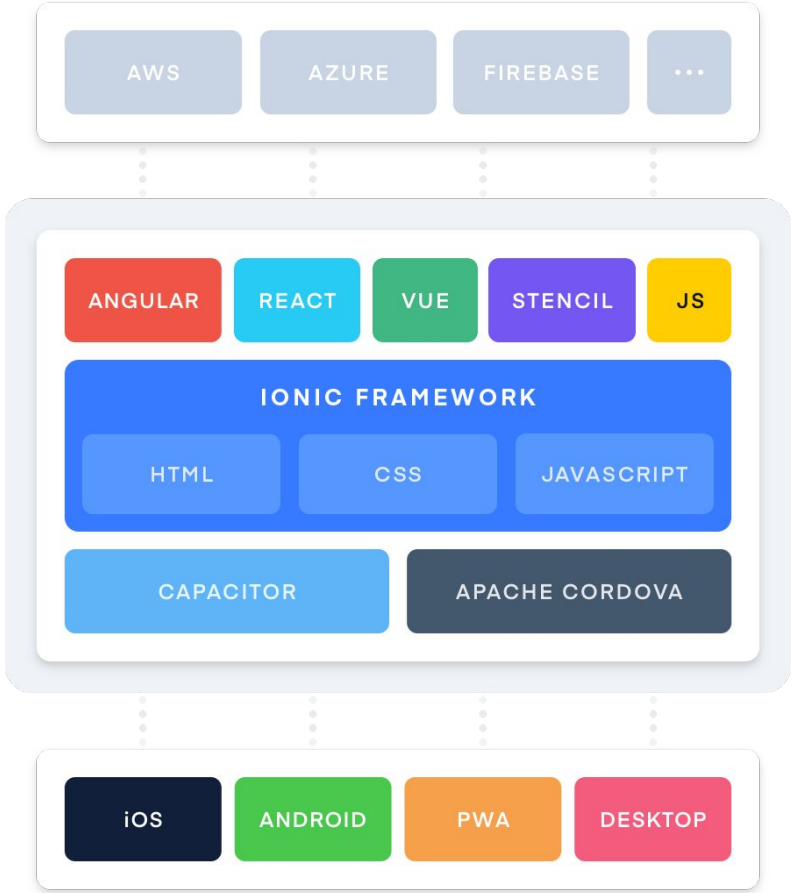
```
ionic start
ionic integrations enable capacitor
```

```
npx cap add android
```

```
ionic build --prod

npx cap copy

npx cap open android

npx cap open ios
```

03-ejecutar-en-móvil

REST


AWS
AZURE
FIREBASE
...
ANGULAR
REACT
VUE
STENCIL
JS
IONIC FRAMEWORK
HTML
CSS
JAVASCRIPT
CAPACITOR
APACHE CORDOVA
iOS
ANDROID
PWA
DESKTOP

REST (representational state transfer) un estilo de arquitectura para desarrollar servicios

- Cliente / Servidor
- Sin estado
- Cache
- Servicios uniformes
- Arquitectura en capas

REST siempre se diferenció por su sencillez ya que utiliza todas las características que puede de HTTP en vez de reinventarse lo que ya tiene HTTP:

- Las operaciones a realizar
- La Estructura de la URI
- Tratamiento de errores
- El formato de los datos
- El estado de la aplicación
- Cache
- ....

https://ionic-curso-preguntas-service.herokuapp.com/swagger-ui.html

RX-JS

“The Reactive Extensions (Rx) is a library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators.”

Según Microsoft

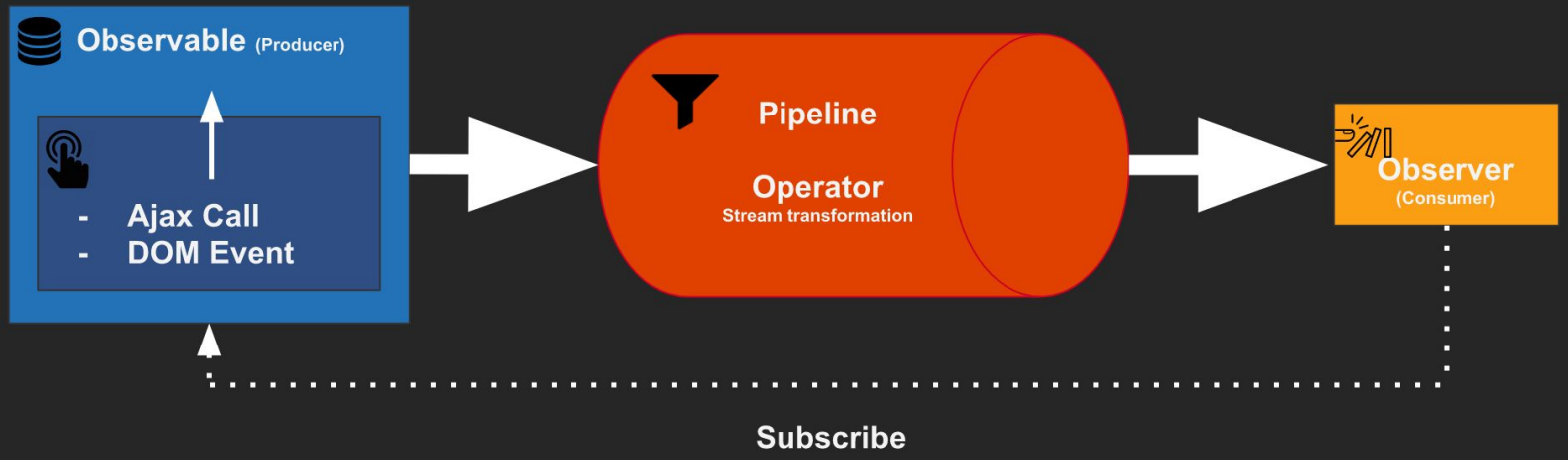
```
async_func1(function(err, data){
   if (!err) {
      async_func2(function(err, data){
          if (!err) {
             async_func3(function(err, data){
                 if (!err) {
                     async_func4(function(err, data){
                      });
                 } else {
                     log(err);
                 }
             });
          } else {
             log(err);
          }
      });
  } else {
     log(err);
  }
});
```

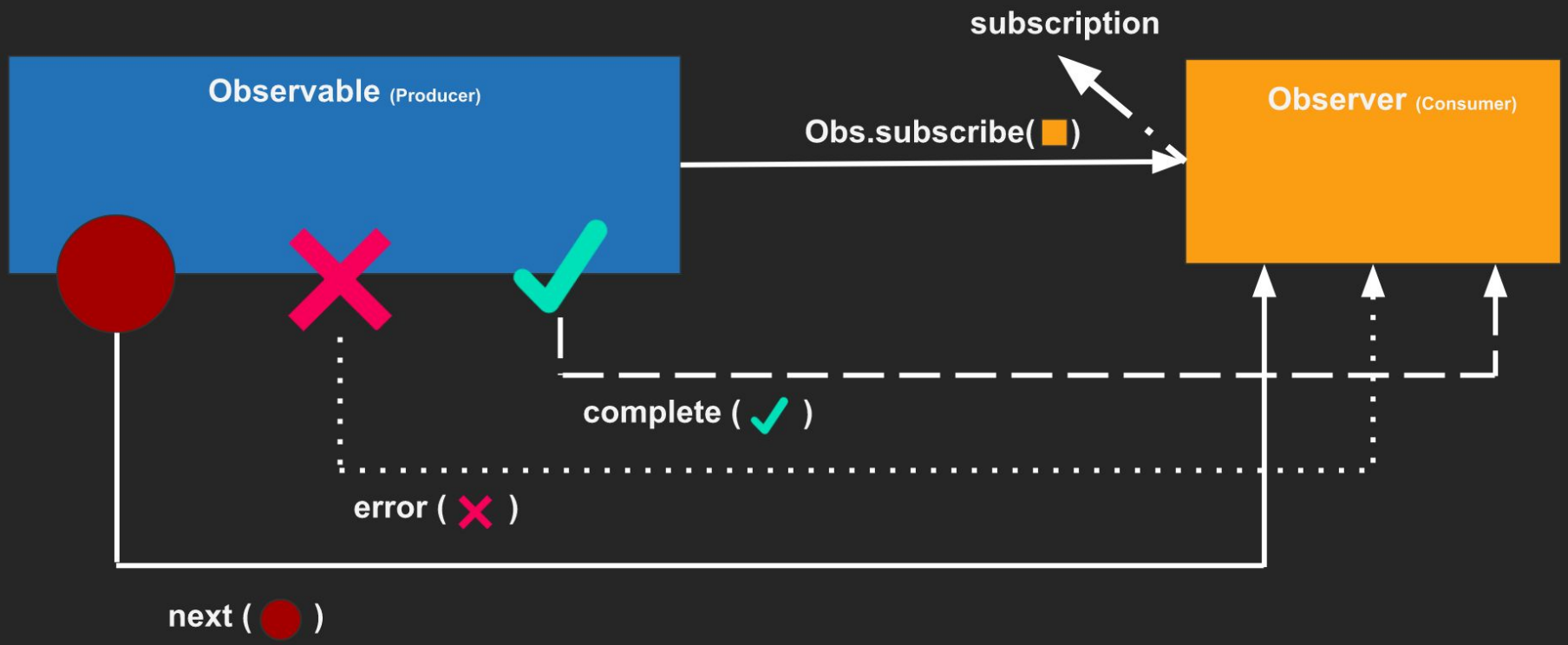
Observable

- En RxJava, Observable es una clase que emite un flujo de datos o eventos.
- Los observables son las fuentes de los datos. Por lo general, comienzan a proporcionar datos una vez que un suscriptor comienza a escuchar.
- Un observable puede emitir cualquier número de elementos (incluido cero elementos).
- Puede terminar con éxito o con un error. algunos nunca pueden terminar, por ejemplo, un observable para un click de botón puede producir potencialmente una secuencia infinita de eventos.

Suscriptores

- Subscriber es una clase que actúa sobre los elementos emitidos.
- Un observable puede tener cualquier número de suscriptores.
- Si el observable emite un nuevo elemento, onNext() se llama al método en cada suscriptor.
- Si el observable finaliza con éxito su flujo de datos, onComplete()se llama al método en cada suscriptor.
- De manera similar, si el observable termina su flujo de datos con un error, onError()se llama al método en cada suscriptor.


Observable (Producer)
- Ajax Call
- DOM Event
Pipeline
Operator
Stream transformation
Observer
(Consumer)
Subscribe


subscription
Observable (Producer)
Obs.subscribe( )
Observer (Consumer)
complete ( )
error ( )
next ( )

Algunos observables

- `of('Hola mundo')`
- `from([1, 2, 3, 4])`
- `interval(1000)`
- `ajax('http://example.com')`
- `webSocket('ws://echo.websocket.com')`
- `fromEvent(button, 'cliack')`
- `...`

```
myObservable.subscribe(
  value => console.log ('next', value),
  err => console.error('error', err),
  () => console.log('complete')
);
```

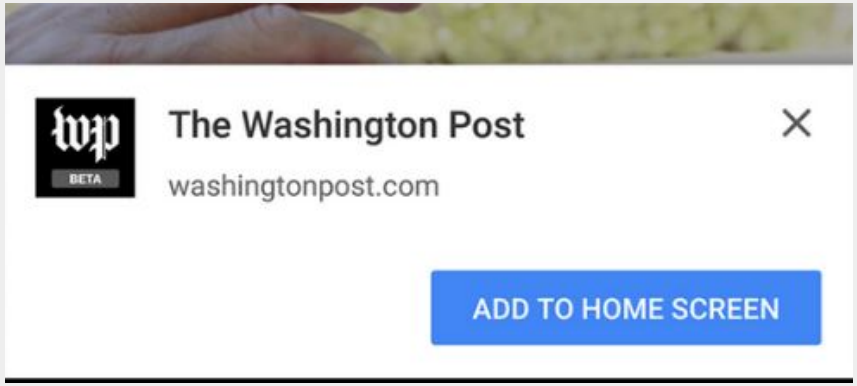
zip switchMap map filter merge reduce

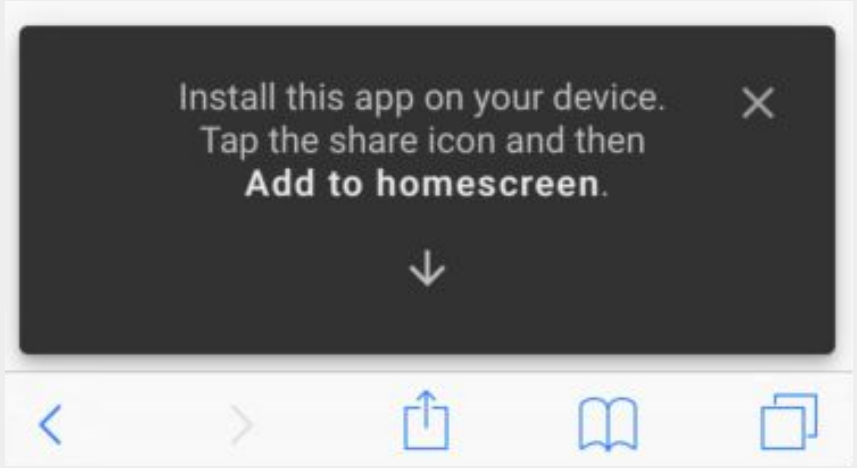
...

04-rx

PWA

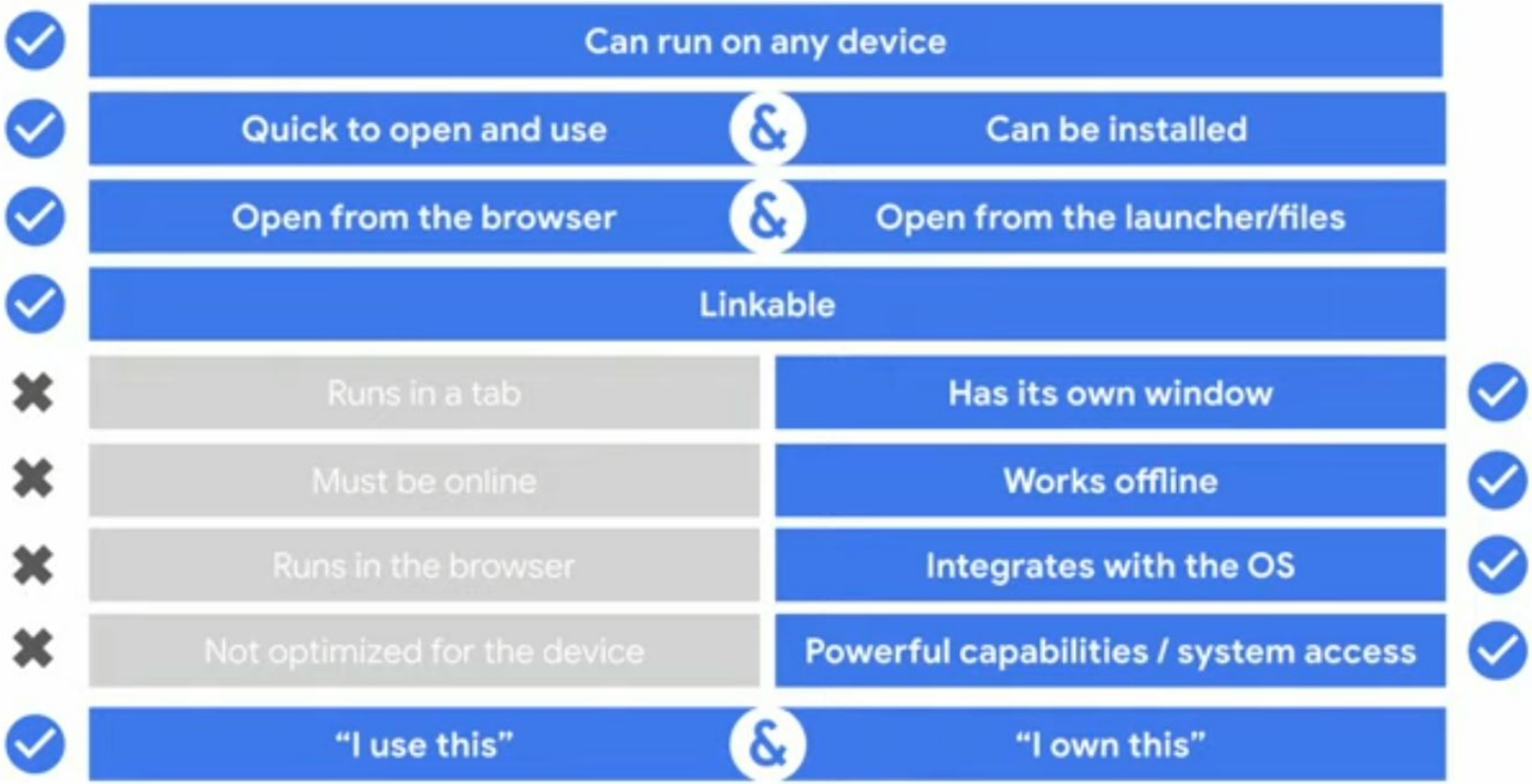
progressive web application





Web	Apps
Can run on any device	Usually ecosystem-specific
Quick to open and use	Download & Install
Open from the browser	Open from the launcher/files
Always runs in the browser	[Feels like it] runs on its own
Always runs in a tab	Has its own window
Probably won't work offline	Usually works fine offline
Not optimized for the device	Powerful capabilities / system access
Linkable	Not linkable
"I use this"	"I own this"

PWAs: toward a universal application platform



Cualquier aplicación es una PWA?

- manifiesto de la aplicacion: define el nombre, icono, url, oritentación de la pantalla, ….
- serviceworkers: son js que funcionan en segundo plano para generar la experiencia offline, por ejemplo el acceso a las páginas, las notificaciones push, etc.

05-pwa

Ionic

- Una aplicación ionic puede estar ejecutándose, en un ordenador, cómo en un dispositivo móvil.
- Puede estar ejecutándose cómo una aplicación web, una pwa, o una aplicación nativa.
- Mucho cuidado con el acceso a funciones nativas y la compatibilidad de los plugins, aunque estemos en un entorno que esté ejecutando cordova a veces no hay soporte para el dispositivo.
- Generalmente lo nativo es más eficiente, pero tiene menos compatibilidad

06-dispositivo

Oauth

- Estándar para la autorización de recursos
- Permite compartir información entre sitios sin compartir la identidad
- Implementa diferentes flujos de autenticación: authorization code flow, resource owner password credential flow, implicit flow…
- Utilizados por grandes compañias: Google, Facebook, Microsoft, Twitter, Github ….

¿Por qué OAuth?

- Solucionar el intercambio continuo de credenciales entre cliente y servidor
- Integración con aplicaciones de terceros
- El usuario es quien delega la capacidad de realizar ciertas acciones en su nombre
- Al desarrollar una aplicación, no tenemos necesidad de almacenar usuario/password

Participantes

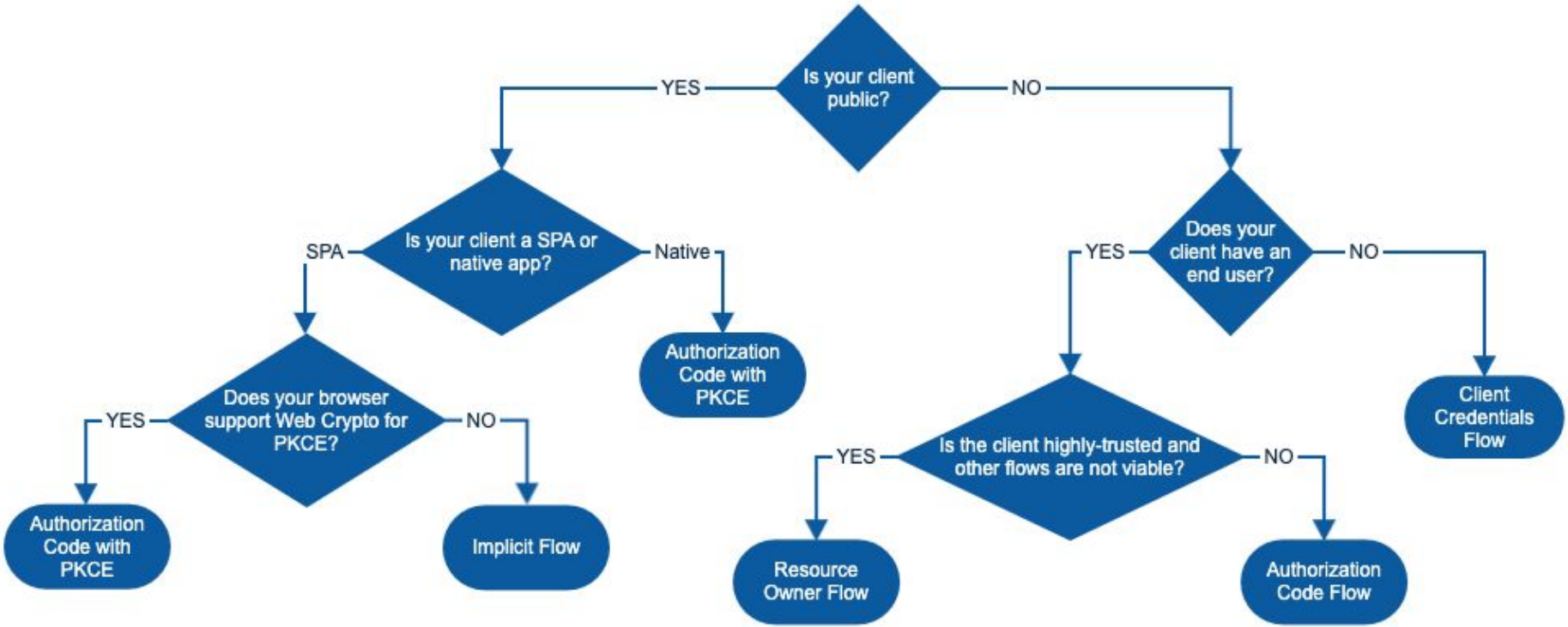
- Dueño del recurso (owner)
- Cliente (Client)
- Servidor de recursos protegidos (Resource server)
- Servidor de autorización (Authorization Server)

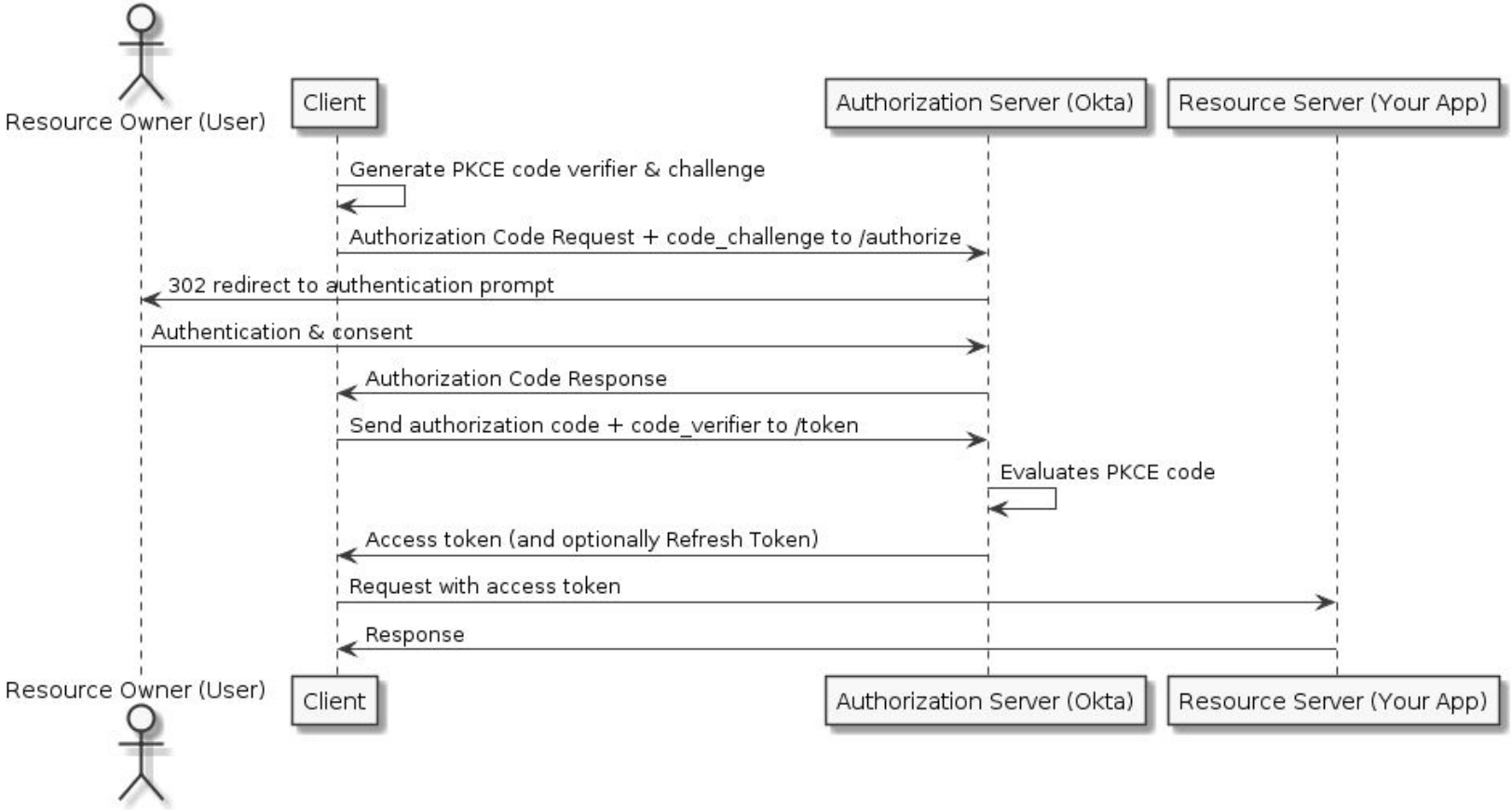
Tipos de clientes

- Clientes confidenciales: son aquellos capaces de guardar una contraseña sin que esta sea expuesta
- Clientes públicos: son aquellos que son capaces de guardar una contraseña y manterla a salvo
- En función del cliente necesimaos implemetar un flujo de oauth o otro.

OpenID Connect

- Agrega una capa a oauth
- Permite a los clientes verificar la identidad del usuario final en función de la autenticación realizada por un servidor de autorización
- Obtiene información de perfil básica sobre el usuario final
- Es necesario, porque oauth proporciona autorización, pero no proporcina autenticación


Is your client public?
YES
NO
Is your client a SPA or native app?
SPA
Native
Does your client have an end user?
YES
NO
Does your browser support Web Crypto for PKCE?
YES
NO
Is the client highly-trusted and other flows are not viable?
YES
NO
Authorization Code with PKCE
Implicit Flow
Authorization Code with PKCE
Client Credentials Flow
Resource Owner Flow
Authorization Code Flow


Resource Owner (User)
Client
Authorization Server (Okta)
Resource Server (Your App)
Generate PKCE code verifier & challenge
Authorization Code Request + code_challenge to /authorize
302 redirect to authentication prompt
Authentication & consent
Authorization Code Response
Send authorization code + code_verifier to /token
Evaluates PKCE code
Access token (and optionally Refresh Token)
Request with access token
Response
Resource Owner (User)
Client
Authorization Server (Okta)
Resource Server (Your App)

demo