

LECTURE 9

PLATFORM BASED DEVELOPMENT

ACKNOWLEDGEMENTS

- A bunch of Websites where I took all the pictures without asking. Apologies :)
- Stanford CS 47 course.
- U. South Carolina CSCE 546 course.
- Statcounter (<https://gs.statcounter.com/os-market-share/mobile/worldwide>)
- Businessofapps (<https://www.businessofapps.com/data/app-revenues/>)
- Toptal blog (<https://www.toptal.com/javascript/choosing-best-front-end-framework>)

TODAY'S LECTURE

@FRANCISCOVILCHEZV

- Cross-platform development
 - Native Apps
 - Mobile Web
 - Hybrid Apps
- Which one is better?
- Ionic
 - Components
 - Plugins
 - Building Process
 - CLI

CROSS-PLATFORM DEVELOPMENT

CROSS-PLATFORM DEVELOPMENT

INTRODUCTION

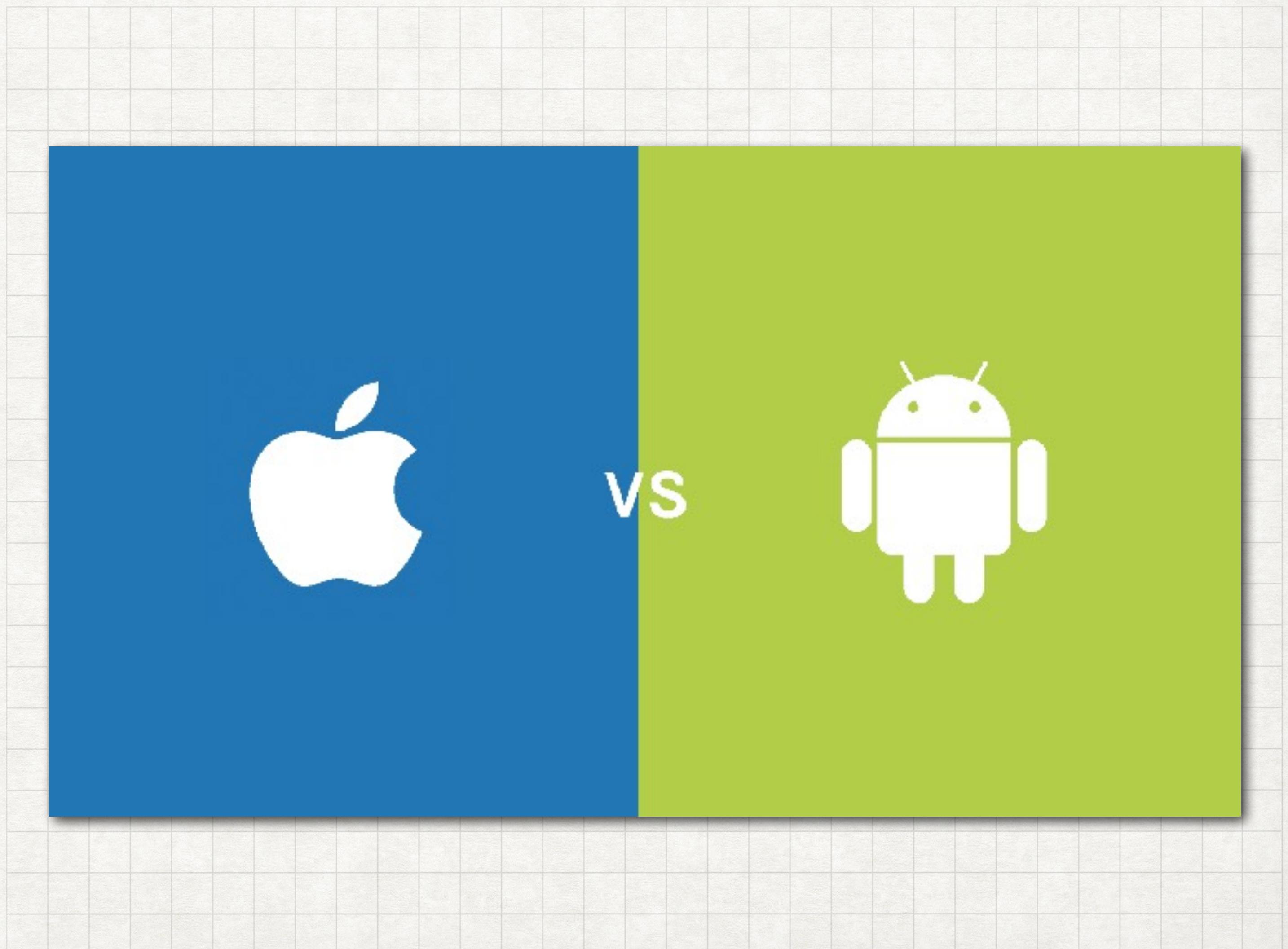
- Actualmente, el mercado móvil está por dispositivos:
 - Android (72.44%)
 - iOS (26.75%)
- Interesantemente, las ventas de aplicaciones en cada una de sus tiendas es inversa:
 - App Store (\$72.3b)
 - Google Store (\$38.6b)



CROSS-PLATFORM DEVELOPMENT

NATIVE APPS

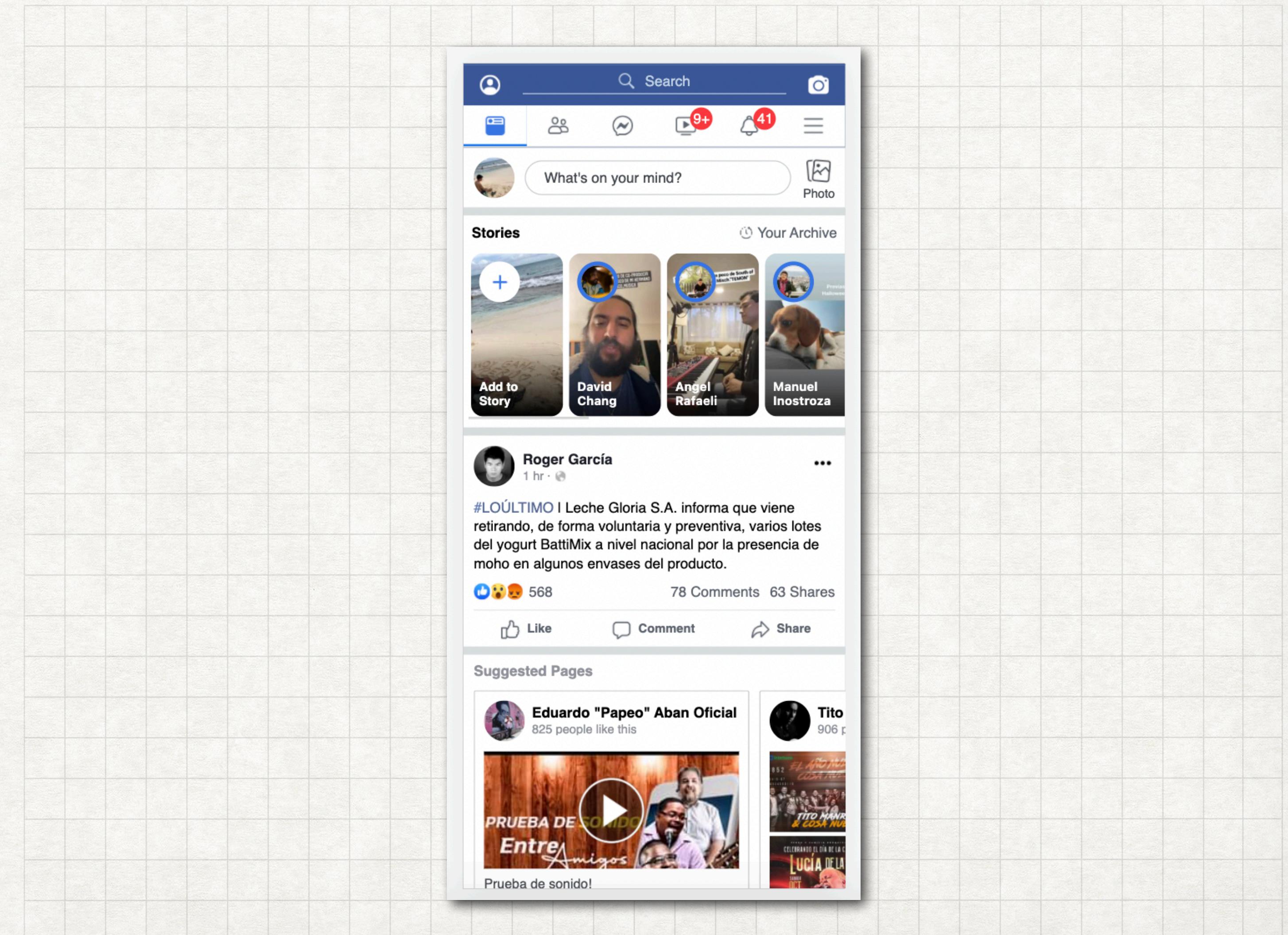
- Cada uno de dichos sistemas permite a los programadores utilizar los distintos dispositivos y funcionalidades del teléfono mediante el uso de librerías, expuestas en lo que generalmente se conoce como el SDK
 - Android SDK: Librerías en Java
 - iOS SDK: Librerías en Swift y Objective-C
- Debido a ello, se necesita el uso de distintos lenguajes de programación para desarrollar aplicaciones para cada sistema operativo.
- Algunas de las funcionalidades que exponen son:
 - Acceso a la cámara del teléfono
 - Uso del GPS
 - Envío de Push Notifications
 - Etc...



CROSS-PLATFORM DEVELOPMENT

MOBILE WEB

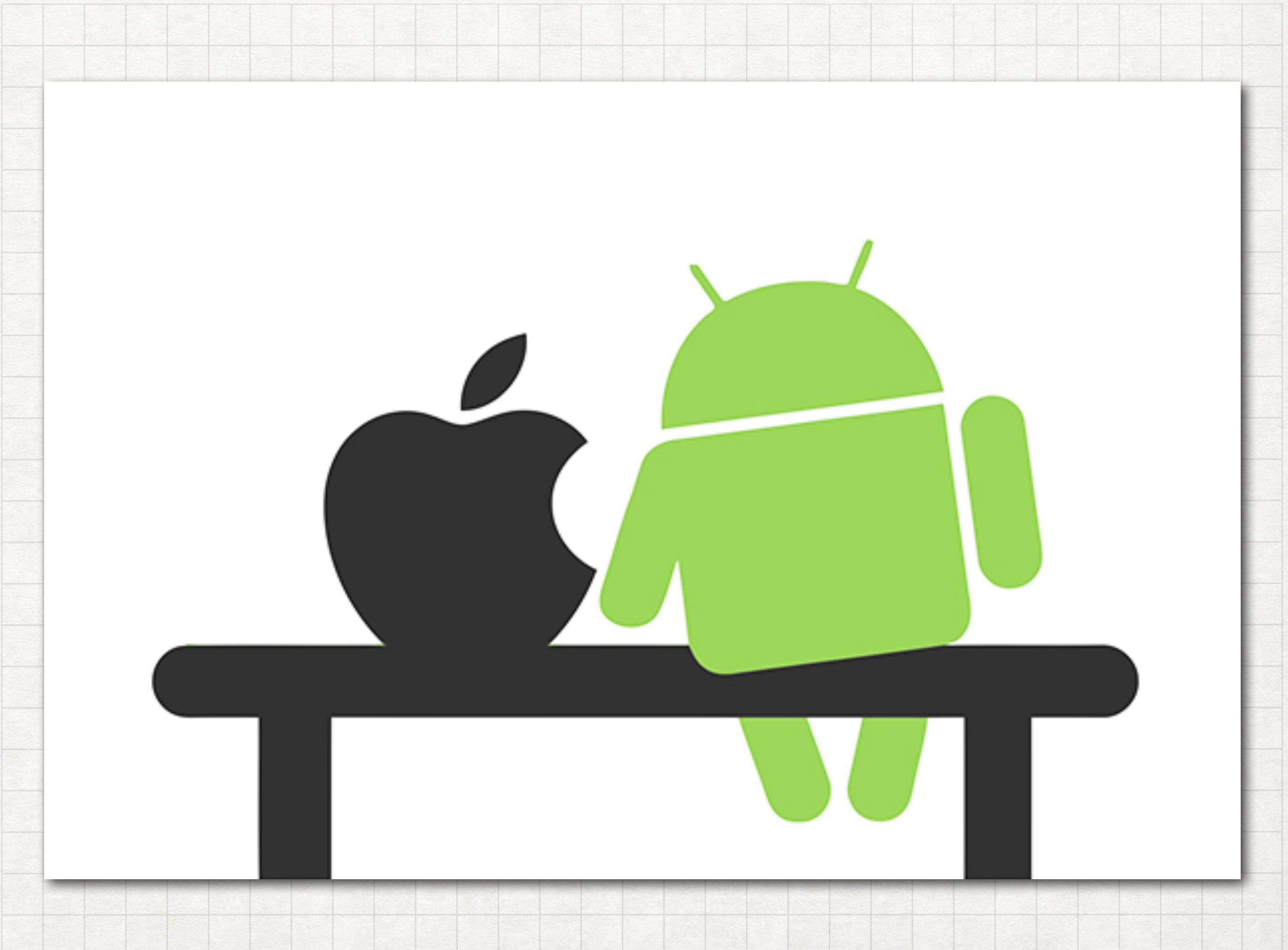
- Debido a que es necesario distintos programadores para el desarrollo de una aplicación para cada plataforma, algunos programadores optan por el desarrollo de una aplicación web para ahorrarse el trabajo de desarrollar múltiples proyectos.
- Por ejemplo, Facebook tiene una versión web especial para dispositivos mobile (mobile web): m.facebook.com
- Sin embargo, dicha opción no permite aprovechar algunas capacidades del dispositivo, como push notifications, acceso a data de otras aplicaciones como el calendario, etc.



CROSS-PLATFORM DEVELOPMENT

HYBRID APP

- Distintos frameworks han aparecido para permitir el desarrollo de aplicaciones para múltiples dispositivos con el uso de "un sólo lenguaje de programación".
 - React Native: Javascript, HTML, CSS + React
 - Ionic: Javascript, HTML, CSS + Angular/React
 - Flutter: Dart
 - Xamarin: C## (.Net)
- Alguno de ellos permiten que incluso el mismo código genere aplicaciones para Windows, MacOS, TvOS, etc...



**WHICH ONE IS
BETTER?**

WHICH ONE IS BETTER?

DON'T BE LOYAL TO A TECHNOLOGY

- En el mundo de la tecnología, es muy importante mantener una mente abierta.
- Cada tecnología es una herramienta para nosotros, y la usamos de acuerdo a los requerimientos del proyecto. Si no nos permite cumplir la misión, usamos otra.
- Algunos criterios utilizados para elegir una tecnología:
 - Disponibilidad de recursos educativos
 - Popularidad
 - Funcionalidades específicas
- Caso airbnb: <https://medium.com/airbnb-engineering/sunsetting-react-native-1868ba28e30a>

				
[redacted]	✓		✓	✓
[redacted]	✓	✓	✓	
[redacted]		✓		✓
[redacted]		✓	✓	✓
[redacted]	✓		✓	✓

WHICH ONE IS BETTER?

COMPARISON

NATIVE vs. WEB vs. HYBRID: 7 FACTORS OF COMPARISON

KEY CON PRO NEUTRAL

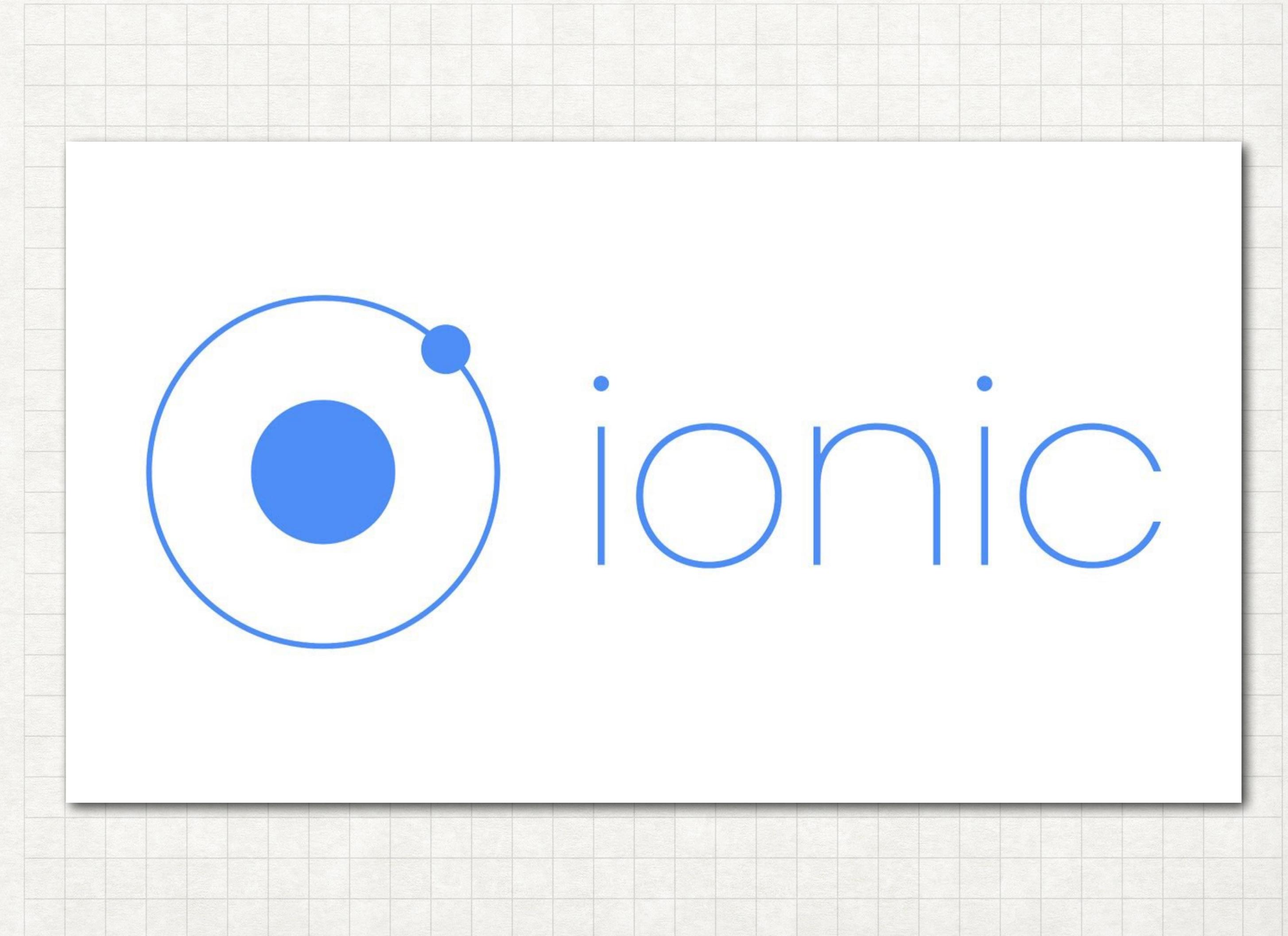
	NATIVE	HYBRID	WEB
COST	Commonly the highest of the three choices if developing for multiple platforms	Similar to pure web costs, but extra skills are required for hybrid tools	Lowest cost due to single codebase and common skillset
CODE REUSABILITY/PORTABILITY	Code for one platform only works for that platform	Most hybrid tools will enable portability of a single codebase to the major mobile platforms	Browser compatibility and performance are the only concerns
DEVICE ACCESS	Platform SDK enables access to all device APIs	Many device APIs closed to web apps can be accessed, depending on the tool	Only a few device APIs like geolocation can be accessed, but the number is growing
UI CONSISTENCY	Platform comes with familiar, original UI components	UI frameworks can achieve a fairly native look	UI frameworks can achieve a fairly native look
DISTRIBUTION	App stores provide marketing benefits, but also have requirements and restrictions	App stores provide marketing benefits, but also have requirements and restrictions	No restrictions to launch, but there are no app store benefits
PERFORMANCE	Native code has direct access to platform functionality, resulting in better performance	For complex apps, the abstraction layers often prevent native-like performance	Performance is based on browser and network connection
MONETIZATION	More monetization opportunities, but stores take a percentage	More monetization opportunities, but stores take a percentage	No store commissions or setup costs, but there are few monetization methods

IONIC

IONIC

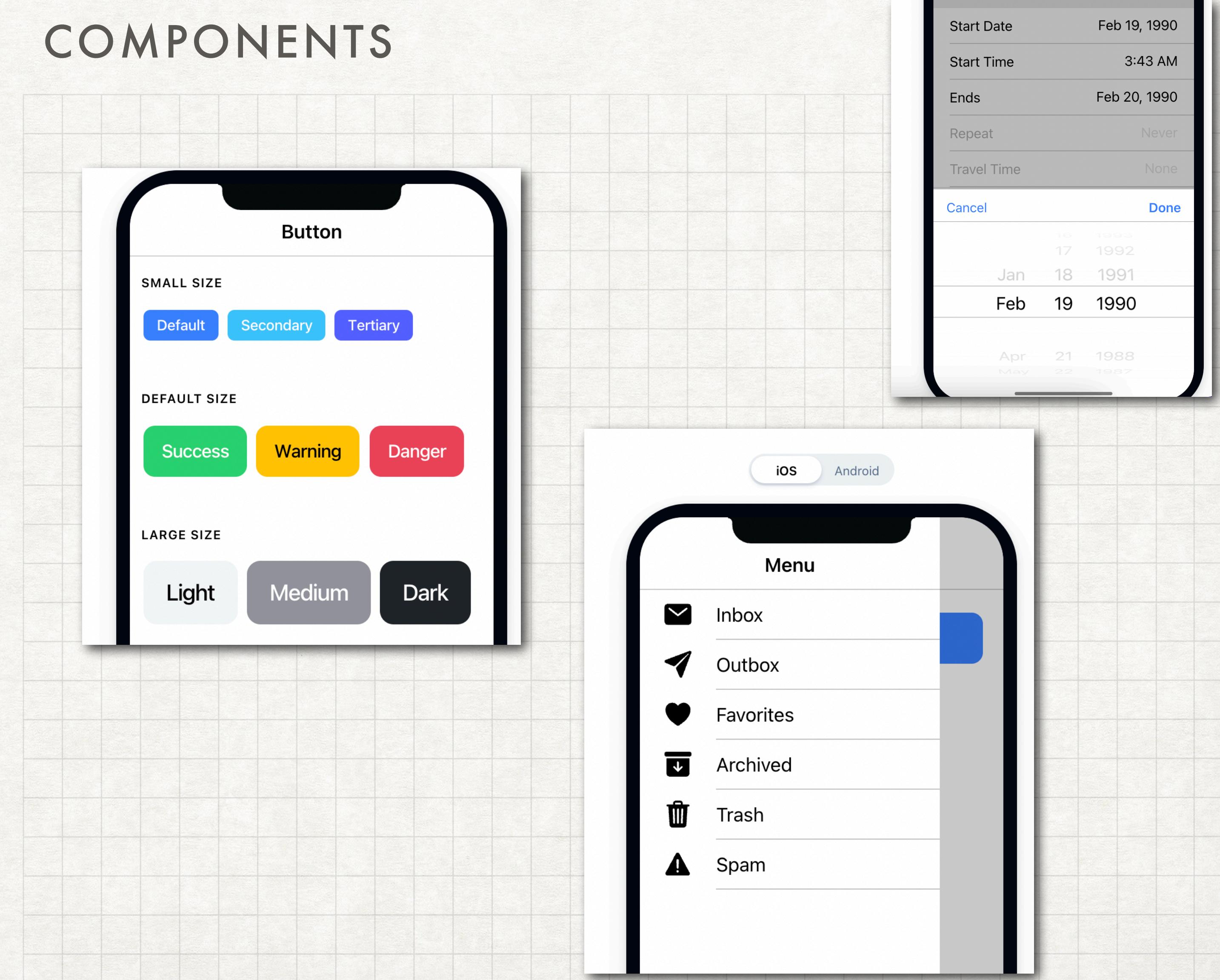
INTRODUCTION

- Ionic es un framework que permite el desarrollo de aplicaciones híbridas.
- Utiliza HTML, CSS y Javascript para el desarrollo de la aplicación.
- Permite utilizarlo en conjunto con Angular, React, Vue, etc...
- Te permite generar los respectivos .apk o .app para ser instalados en el dispositivo.



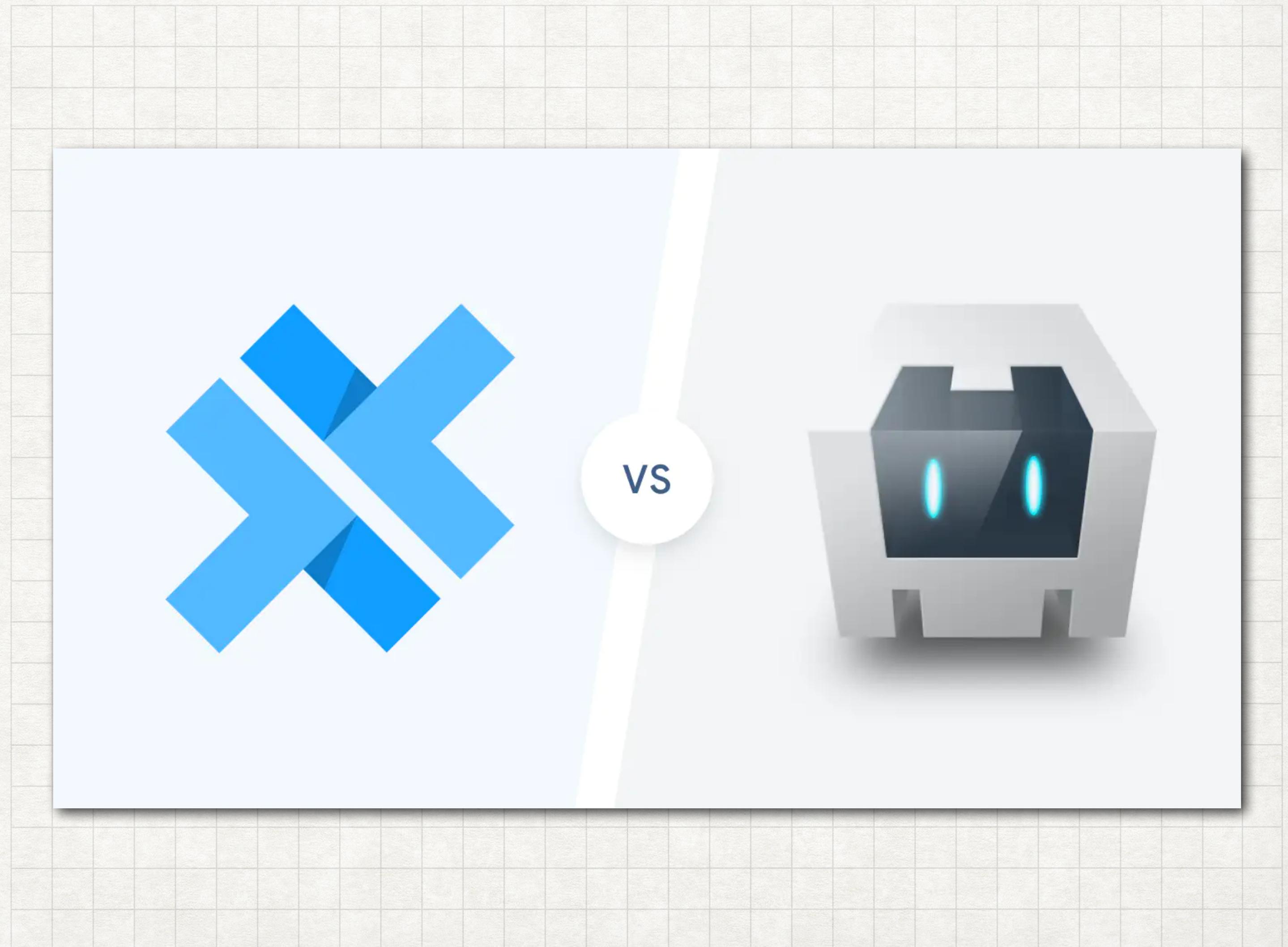
IONIC COMPONENTS

- Ionic te brinda un conjunto de componentes ya creados para que uses en tu aplicación (tags HTML ya definidos por el framework Ionic)
 - Botones
 - Listas
 - Alertas
 - Menu bars
 - Por lo tanto, no es necesario Bootstrap u otro frameworks CSS.



IONIC PLUGINS

- Ionic utiliza internamente librerías de **Capacitor** o **Córdova** para el uso de las funcionalidades del teléfono, como acceso a la cámara, GPS, push notifications, etc...
- Capacitor ha sido creado por el mismo team de Ionic, y permite ser usado no solo para desarrollo móvil, sino para Electron o Web Progressive apps.
- Por esta razón, usaremos los plugins de Capacitor en el desarrollo del curso.



IONIC CAPACITOR

- Capacitor ofrece acceso a distintas funcionalidades del teléfono.
- Para usar dichos dispositivos, se importa la librería respectiva y te da acceso a funciones para utilizarlo dicho dispositivo de acuerdo a tus necesidades.
- En las imágenes podemos ver ejemplos de la documentación de Capacitor

Example

```
import { Camera, CameraResultType } from '@capacitor/camera';

const takePicture = async () => {
  const image = await Camera.getPhoto({
    quality: 90,
    allowEditing: true,
    resultType: CameraResultType.Uri
  });

  // image.webPath will contain a path that can be set as an image src.
  // You can access the original file using image.path, which can be
  // passed to the Filesystem API to read the raw data of the image,
  // if desired (or pass resultType: CameraResultType.Base64 to getPhoto)
  var imageUrl = image.webPath;

  // Can be set to the src of an image now
  imageElement.src = imageUrl;
};
```

```
import { Motion } from '@capacitor/motion';

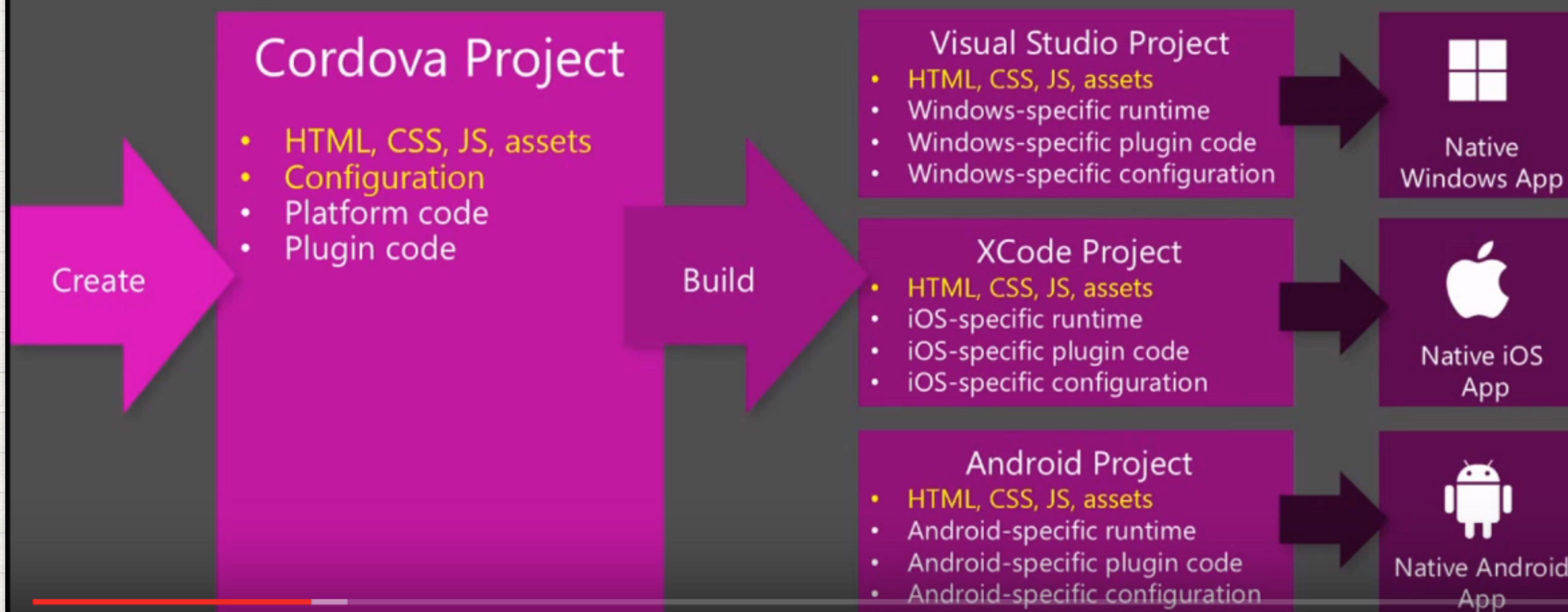
myButton.addEventListener('click', async () => {
  try {
    await DeviceMotionEvent.requestPermission();
  } catch (e) {
    // Handle error
    return;
  }

  // Once the user approves, can start listening:
  Motion.addListener('accel', event => {
    console.log('Device motion event:', event);
  });
});
```

Action Sheet
App
App Launcher
Browser
Camera
Clipboard
Device
Dialog
Filesystem
Geolocation
Haptics
Keyboard
Local Notifications
Motion
Network
Push Notifications
Screen Reader
Share

IONIC + CAPACITOR/CORDOVA BUILDING PROCESS

How does Cordova work?



IONIC CLI

- Ionic cuenta con su propia cli que usaremos para:
 - Crear components
 - Services
 - Guards
 - Build the project
 - Sincronizar dependencias
 - Etc...

THANKS