

Universidad de Jaén

Escuela Politécnica Superior de Jaén

Departamento de Informática

Don Víctor Manuel Rivas Santos, tutor del Proyecto Fin de Carrera titulado: Análisis del kit de herramientas Flutter y desarrollo de un prototipo de aplicación multiplataforma, que presenta Adson Henrique Moreira da Silva, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, noviembre de 2020

Los tutores:

Victor manuel Rivas Santos

El alumno:

Adson Henrique Moreira da Silva

Índice

[1. INTRODUCIÓN 3](#_Toc61878547)

[2. EL CAMINO HASTA FLUTTER 3](#_Toc61878548)

[2.1. La rápida ascensión de internet y móviles 3](#_Toc61878551)

[2.2. Android y iOS dominan el mercado 6](#_Toc61878552)

[2.3. El mercado de aplicaciones 7](#_Toc61878553)

[2.4. El mercado de desarrollo de software móvil nativo 7](#_Toc61878554)

[2.5. Herramientas para desarrollo móvil multiplataforma 8](#_Toc61878555)

[2.5.1. Xamarin 9](#_Toc61878556)

[2.5.2. PhoneGap 9](#_Toc61878557)

[2.5.3. Titanium 9](#_Toc61878558)

[2.5.4. React Native 9](#_Toc61878559)

[2.6. Herramientas multiplataformas. 9](#_Toc61878560)

[2.6.1. Web Aplicaciones móviles 10](#_Toc61878561)

[2.6.2. Web Aplicaciones hibridas 11](#_Toc61878562)

[2.6.3. Aplicaciones interpretadas 12](#_Toc61878563)

[2.6.4. Aplicaciones generadas por multiplataformas 12](#_Toc61878564)

[2.7. El problema de herramientas multiplataformas 13](#_Toc61878565)

[3. FLUTTER 14](#_Toc61878566)

[Bibliografía 15](#_Toc61878567)

# INTRODUCIÓN

Escribir una introdución

# EL CAMINO HASTA FLUTTER

En este capítulo será abordado una historia concisa de la rápida evolución de internet y de los dispositivos móviles para entender como llegamos a herramientas como Flutter.



## La rápida ascensión de internet y móviles

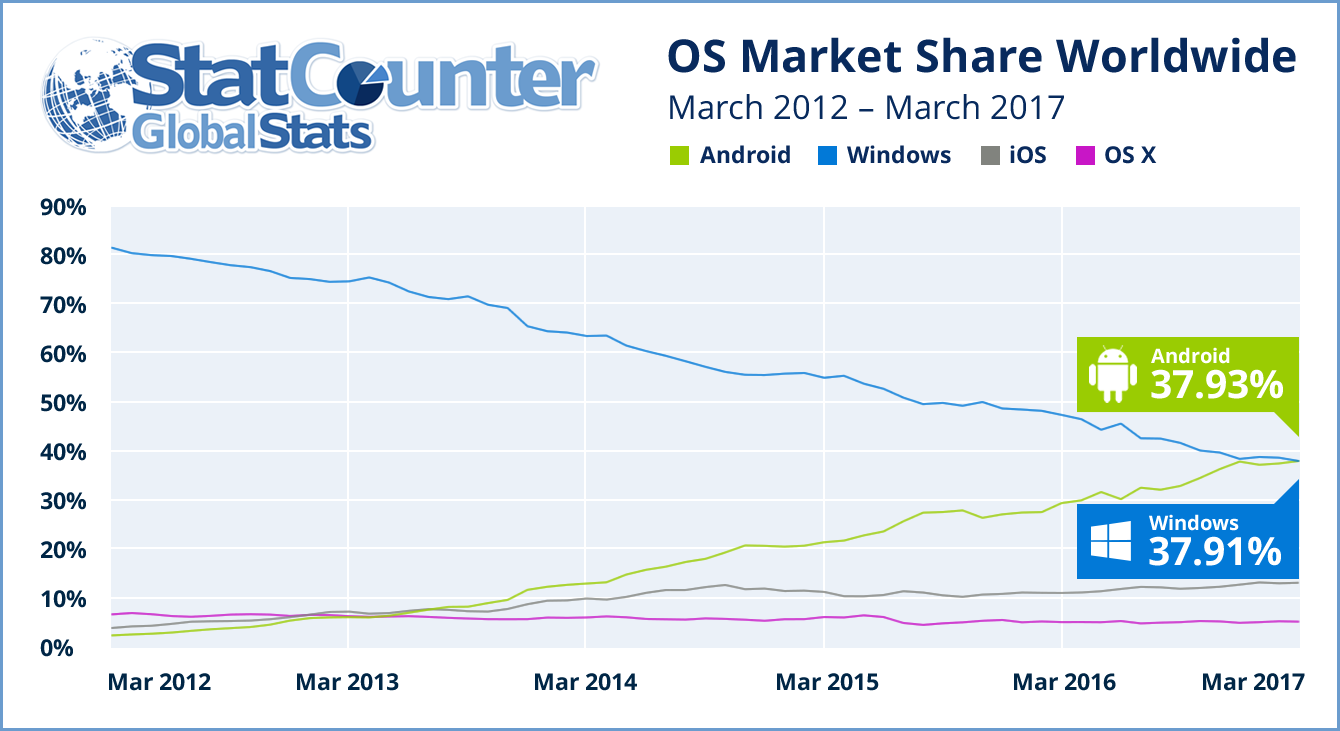
Es importante mirar al pasado y constatar que los móviles, para alcanzar el estado que conocemos hoy, tuvieran una rápida evolución.

Hoy en día, los móviles tienen muchas más funcionalidades si comparado con antiguamente. Hoy no sirven apenas para llamadas, en verdad es que poca gente lo utiliza para eso, sino que sirve para muchas cosas como *internet banking*, juegos, capturar fotos, estar conectado con amigos por redes sociales, mirar noticias en tiempo real, asistir televisión, hacer compras online, escuchar músicas y muchas otras. Y esta rápida evolución se debe casi que integralmente a revolución de la internet, con el *boom* de las puntocom (medio de los años 2000) y con el suceso de las redes sociales. (PONER REFERENCIA).

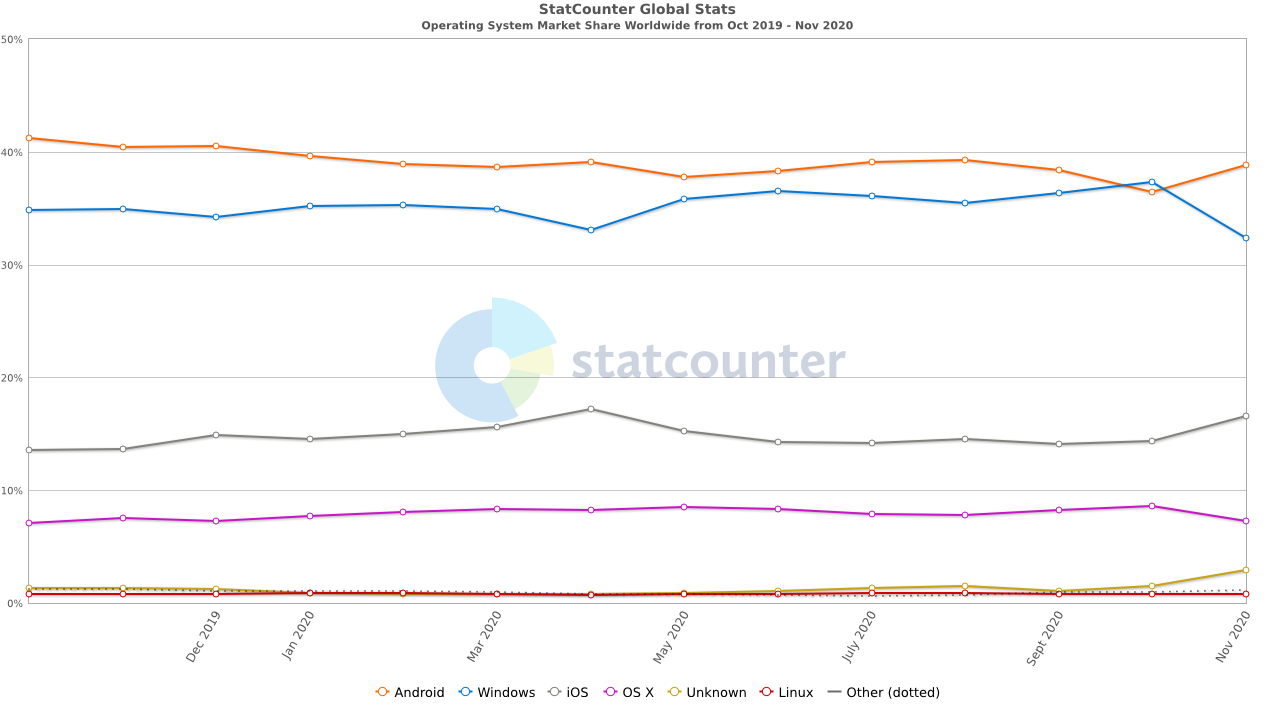
Siguiendo el crecimiento de la internet, muchas innovaciones llegaran para los móviles, en los meados de los años 2000. Fabricantes de móviles empezaran a construir funcionalidades de *MP3 Player,* consola de juegos portátil. El mercado de móviles que antes era dominado por Nokia, Sony Ericsson, LG; ahora a principio de os años de 2007, junto con el aparecimiento del termo *smartphone*, el dominio pertenencia a fabricantes como HTC, BlackBerry y Apple (con el primero iPhone).

Con el suceso de redes sociales como Instagram, Facebook, los móviles adquirirán más funcionalidades. Por ejemplo, Instagram - aplicativo para compartir fotos - lanzado en 2010, trae la preocupación de las fabricantes para producir mejores cameras para sus teléfonos. Cuestiones antiguas como modo de espera, tiempos de llamadas, fueran cambiadas por otras como sistemas operativos, cualidad de las cameras, capacidad de la batería, descargar datos de forma más rápida (PONER REFERENCIA).

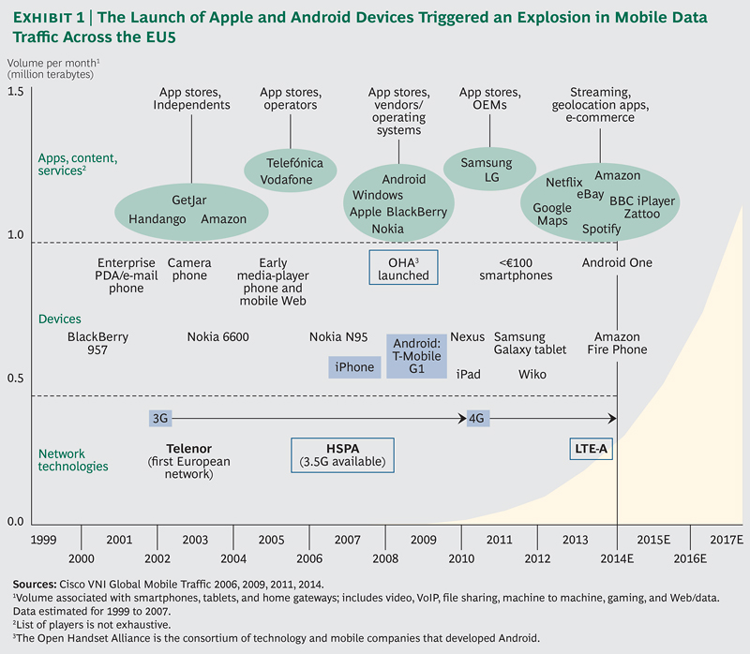
Con el pasar de los años y con el mercado de móviles dominado por Apple con su sistema operativo iOS, Huawei y Samsung optando por usar el sistema operativo Android desarrollado por Google; otras fabricantes como BlackBerry, Microsoft fueran perdiendo el mercado de los móviles y consecuentemente de sistemas operativos móviles (PONER REFERENCIA).

Como si no bastase, en 2007 Android se torna el sistema operativo más utilizado en el mundo, superando a Windows (SO para ordenadores de mesa), conforme muestra la pesquisa, de la empresa de estadísticas web *StatCounter*.

Y aún hoy, sigue siendo el SO más utilizado en el mundo, conforme podemos observar en la imagen a seguir.

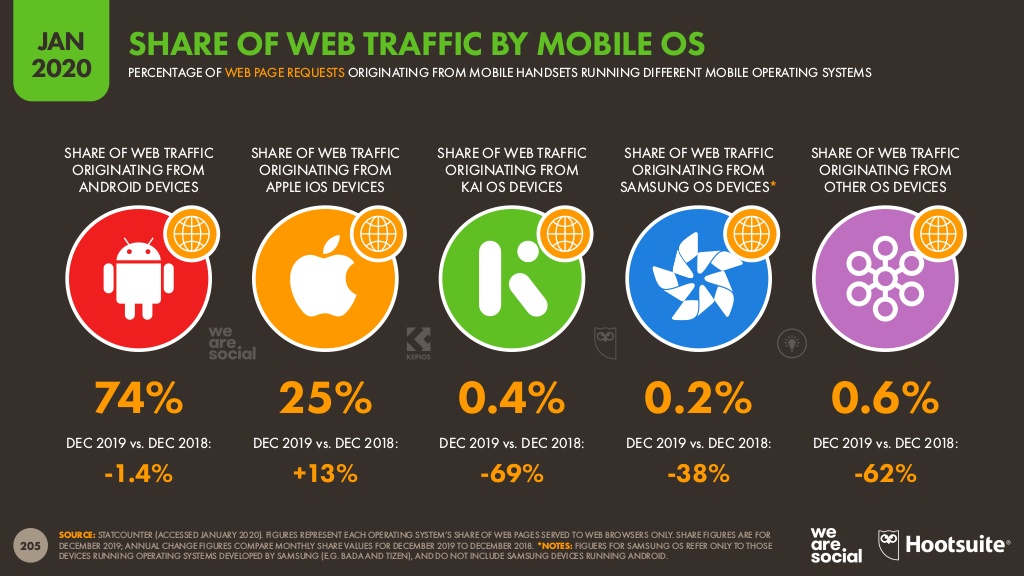
(IMAGEM RETIRADA no dia 02/12/2020)

La imagen en seguida muestra en resumen que lo que ha pasado al largo de los años de la revolución de internet y de los móviles.



## Android y iOS dominan el mercado

Con 5.19 mil millones de usuarios únicos en dispositivos móviles lo que representa 67% de la populación mundial y con 53% de las requisiciones a páginas de internet hechas por móvil, según pesquisa de *We are Social* (PONER REFERENCIA); los *smartphones* están en una ascensión sin precedentes.

Según la misma pesquisa, el sistema operativo Android (Google) es responsable por 74% de las requisiciones a páginas de internet originadas de dispositivos móviles, mientras iOS (Apple) es responsable por 25% y el 1% restante es compartido con otros SO’s.

(RETIRADA NO DIA 14/12/2020)

Lo que vemos entonces es Android y iOS dominando 99% del mercado. De un lado Android, de la gigante Google, sistema elegido por casi todas las fabricantes de móviles, donde debido al código abierto (todos pueden descargar, modificar, redistribuir de forma gratuita) de Android, las fabricantes pueden modificarlo y hacer sus proprias versiones. De otro lado iOS, de Apple, un sistema privado y que solo ella lo utiliza y lo modifica para un de sus principales productos, iPhone.

## El mercado de aplicaciones

Aplicaciones están presente en la vida de todos que tienen un móvil. Sea para usar redes sociales, hacer compras, jugar, entretenimiento, escuchar canciones, mapas. En los días de hoy, es muy difícil que no haya una aplicación, para hacer alguna tarea que utilice internet.

Según el informe anual de *We are Social* (PONER REFERENCIA), pasamos en media 3 horas y 40 minutos por día usando el móvil y ese valor crece 10% a cada año. Y de ese total de tiempo, pasamos 91% de ello en aplicaciones, que descargamos o que vienen en el proprio móvil. Los otros 9% del tiempo pasamos usando los navegadores de internet presentes en el móvil.

Para tener una idea, el número de descargas de aplicaciones en 2019 fue de 204 mil millones, ese número tiende a crecer 6% al año. En el año pasado los usuarios gastaran cerca de 120 mil millones de dólares en aplicaciones móviles y ese valor tiende a continuar creciendo a cada año, con una tasa de 20%.

Sabiendo esto, de todo ese valor agregado a los dispositivos móviles, de todo el capital que ellos están generando en los últimos años y que la tendencia es el crecimiento al largo de los años, la empresa que no tiene su negocio disponible online; sea por medio de aplicaciones, sea por medio de sitios web está perdiendo dinero y probablemente clientes.

## El mercado de desarrollo de software móvil nativo

Hasta el surgimiento de Android y iOS el mercado de desarrollo de software móvil, en general, era pequeño y tímido. Hasta porque los móviles, hasta entonces, no podrían hacer grandes cosas, eran limitados a hacer llamadas, mensajes de texto, juegos y otras cosas más, pero con mucha limitación si comparados a hoy.

Entonces con el surgimiento de iOS y Android y debido a rápida evolución de los *smartphones*, al gran valor agregado que los mismos traen para los usuarios, todo el mercado se cambió. Ahora con eses sistemas operativos, con la popularización accesible de internet y de los móviles no hay justificativa para las empresas no desarrollaren sus proprias aplicaciones y ponerlas a disposición en una de las tiendas de aplicaciones de los sistemas operativos o quizás ponerlas en todas.

Como dicho anteriormente las empresas podrían elegir lanzar sus aplicaciones en una plataforma o en todas. Las empresas que querían lanzar en varias plataformas (sea cualquiera el motivo) tenían que desarrollar la aplicación muchas veces, una para cada plataforma. Pero mirando la diferencia entre las plataformas, los sistemas operativos, eso no era común.

Enfocando más en las plataformas que dominan el mercado y dado la diferencia entre Android, desarrollado en el lenguaje de programación JAVA - inicialmente y iOS con Objective-C, muchas empresas y desarrolladores optaban por lanzar sus aplicaciones solo en una plataforma, generalmente donde tenían la mayor parte de sus clientes. Entonces casos de aplicativos como Instagram, que fue lanzado primeramente en la plataforma iOS y solo dos años después lanzada en Android, era muy común.

Desde entonces el mercado de desarrollo de móvil ha pasado por grandes evoluciones, expansiones e inversiones (PONER REFERENCIA LIBRO). Es difícil encontrar una aplicación que no esté disponible en la mayoría de las plataformas y eso no quiere decir que el desarrollo nativo individual para cada plataforma ha acabado. La verdad es que las herramientas para el desarrollo nativo de aplicaciones (SDK) puestas a disposición por sus mantenedoras mejoraran mucho, posibilitando escribir una aplicación muchas veces para diferentes plataformas, sin tener mucho esfuerzo como antes.

A parte del mejoramiento de los SDK’s nativos, hube también un gran aumento de técnicas y herramientas de desarrollo no nativos de multiplataforma que permitía escribir una aplicación una vez y usarla casi que da la misma forma en todas las plataformas, lo que ha posibilitado entonces desarrollos más rápidos y concisos.

## Herramientas para desarrollo móvil multiplataforma

Con el crecimiento y el desarrollo de herramientas que permitían desarrollar una aplicación una sola vez y usar esa misma única aplicación en todas o casi todas las plataformas, muchas técnicas, *softwares* y *frameworks* surgirán con el propósito de facilitar la vida de los desarrolladores y de las empresas.

### Xamarin

Xamarin es una plataforma de aplicación de código abierto y gratuita de Microsoft para la construcción de aplicaciones iOS y Android modernos y de alto desempeño usando los lenguajes C# y .NET (PONER REFERENCIA).

### PhoneGap

Desarrollado por la equipe de Apache Cordova, PhoneGap es un framework de código abierto que fue descontinuado por la compañía Adobe. Utiliza lenguajes de programación usadas en navegadores de internet (HTML, CSS y Javascript) para desarrollar aplicaciones iOS, Android y otras.

### Titanium

Es un ambiente de desarrollo que te permite crear aplicaciones iOS, Android, mantenida por la empresa AppCelerator. También utiliza lenguajes de programación usadas en la web como PhoneGap con una pequeña diferencia que Titanium añade una camada de abstracción a más.

### React Native

React Native es un framework, desarrollado por Facebook, para construcción de aplicaciones iOS y Android que utiliza el framework React – herramienta creada también por Facebook para creación de interfaces web. React native es gratuito y tiene el código abierto. Actualmente es una de las herramientas más utilizadas.

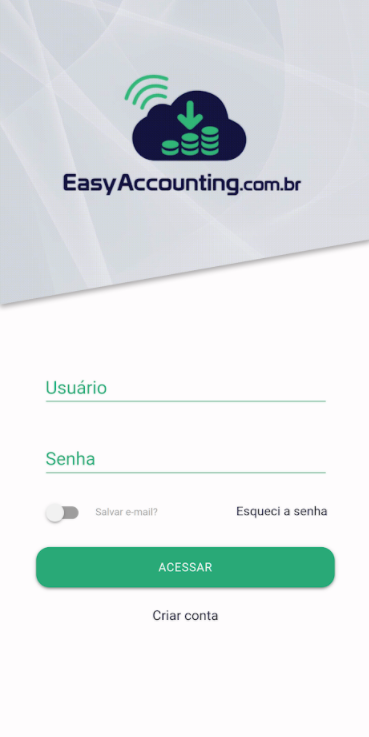
## Herramientas multiplataformas.

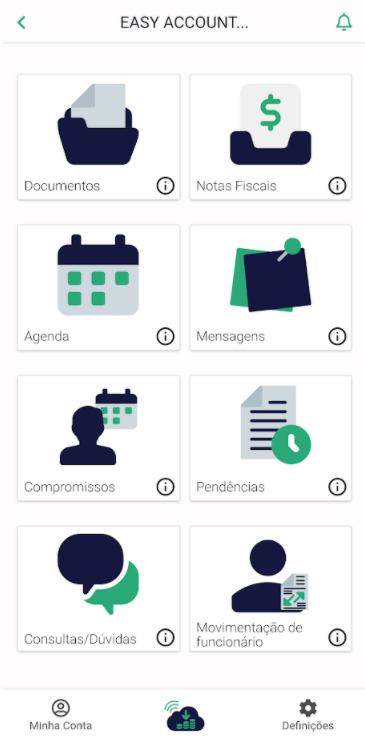
Antes de empezar a entender el problema por detrás de esas herramientas que permiten el desarrollo rápido para que aplicaciones sean lanzadas em muchas plataformas, es importante comprender la clasificación de cada herramienta. Porque el problema surge desde ahí.

Podemos clasificar todas esas herramientas citadas anteriormente y todas las otras que no fueran citadas en cuatro categorías: web aplicaciones móviles, web aplicaciones hibridas, aplicaciones interpretadas y aplicaciones generadas por multiplataformas.

### Web Aplicaciones móviles

Esas aplicaciones son proyectadas para ser ejecutadas en un navegador web (*browsers)* y son independentes de la plataforma de la cual están ejecutando. Es decir, ellas no requieren ninguna adaptación a medida que la plataforma cambia. Por otro lado, el tiempo de respuesta de esas aplicaciones son mayores si comparadas con aplicaciones construidas nativamente (usando SDK fornecidos por los sistemas operativos). Además, esas aplicaciones, por razones de seguridad, tienen permiso limitado para acceder a recursos nativos de un móvil como la cámara o sensores.

PhoneGap es un ejemplo de esa categoría. Esta proyectado con tecnologías web para ser ejecutado en una *webview* (aplicación que ejecuta un ‘navegador web’), conforme el ejemplo en las imágenes a seguir.



Por más que parezca ser aplicativos nativos, no son. Son aplicaciones brasileñas que son desarrolladas utilizando una *webview.*

### Web Aplicaciones hibridas

Web aplicaciones hibridas utilizan el mismo enfoque de la categoría anterior, pero ahora no ejecuta en una *webview.* Ejecutan en un *web container* del dispositivo que tiene mayor acceso a recursos nativos del dispositivo por medio de una interface de programación de aplicación (API).

Aplicaciones hibridas ofrecen grandes ventajas porque te permiten reusar todo el código para varias plataformas. Pero también ofrecen desventajas como ejecución lenta debido al *web container* y la experiencia del usuario al usar la aplicación es comprometida al no utilizar recursos nativos.

La imagen muestra un ejemplo de una aplicación hibrida.

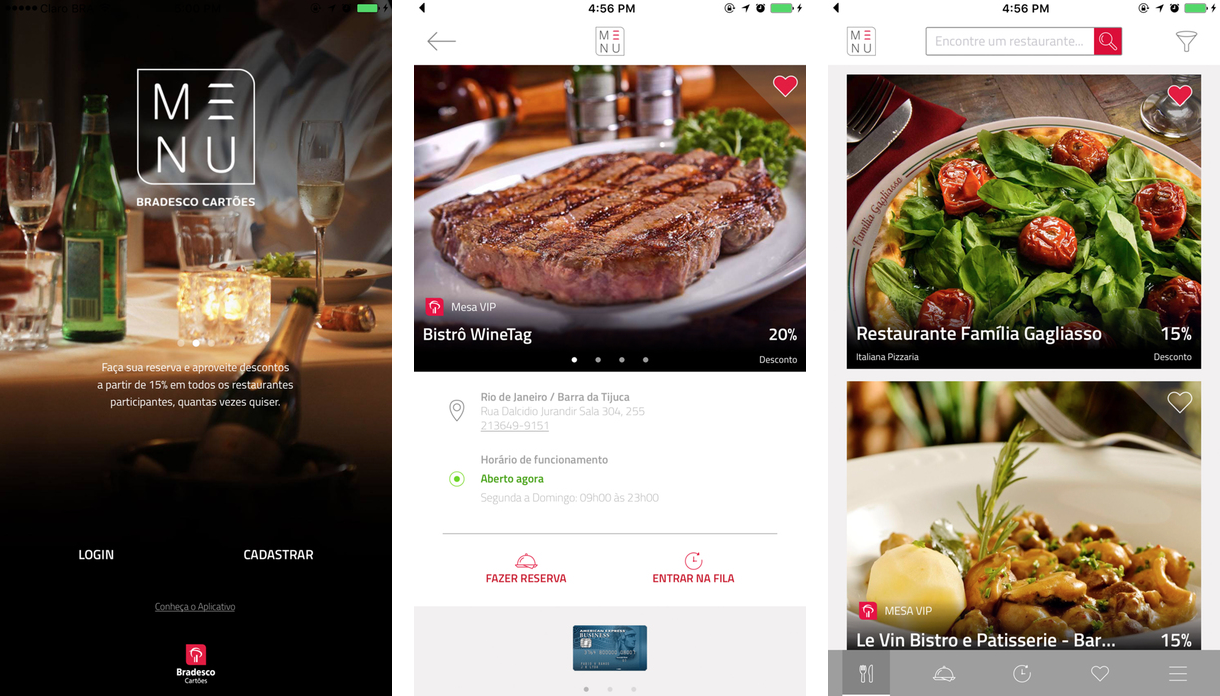


Como se puede observar en la imagen, ella usa el componente nativo del iPhone, Face ID, para autenticar el usuario.

### Aplicaciones interpretadas

Aplicaciones interpretadas son un proyecto que es casi todo traducido para el código nativo. Todo el código fuente de la aplicación es implantada en el dispositivo, donde es interpretada usando algún tipo de mecanismo (generalmente un SDK de la herramienta elegida).

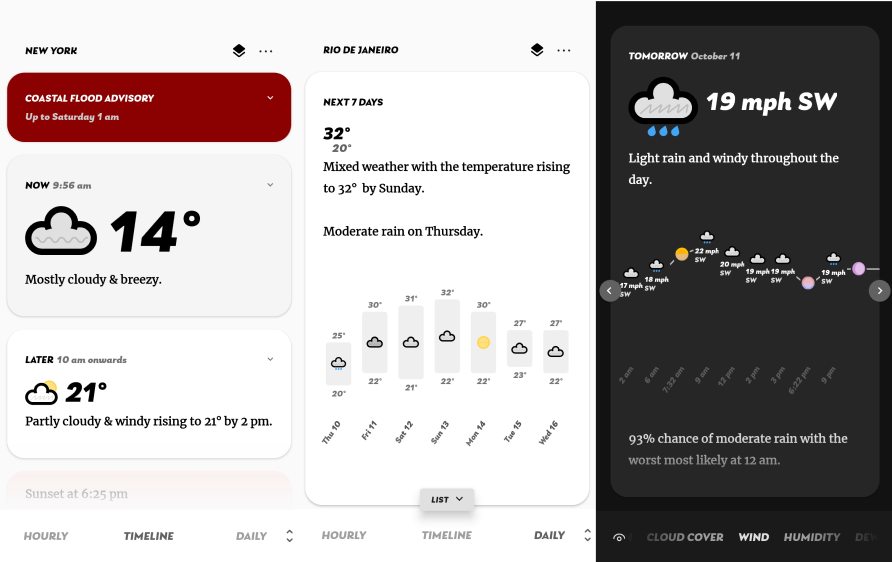
La herramienta Titanium citada anteriormente se encaja en esa categoría, pues cuando la aplicación es instalada en el dispositivo, él es interpretado usando algún tipo de *Javascript Engine*. *Mozilla’s Rhino* es usado en el Android y *JavascriptCore* en iOS.

La ventaja de este tipo de aplicación es que es independiente de la plataforma, consigue llegar muy próximo de una experiencia nativa. Mientras las desventajas son la total dependencia del ambiente de desarrollo.

En la imagen se ve un ejemplo de una aplicación interpretada, casi no si nota la diferencia entre ella y una aplicación nativa justamente por la interpretación.

### Aplicaciones generadas por multiplataformas

Esa categoría es muy parecida con la anterior, la única diferencia es en lugar del código ser interpretado, el código es compilado. Las aplicaciones son compiladas nativamente, creando una versión específica para cada plataforma de destino. Por ejemplo, antes de lanzar una aplicación, la herramienta hace una especie de “traducción” de la aplicación para la plataforma destino. Note que no es el desarrollador que tiene ese trabajo, sino que la herramienta elegida. El desarrollador sigue desarrollando la aplicación una sola vez, pero compilando para las plataformas de destino.

La herramienta Xamarin es un ejemplo de esa categoría y te permite escribir el código una sola vez y compilar para la plataforma que quieres. Las ventajas aquí son las mismas de la anterior y las desventajas aquí son cuanto al tamaño de la aplicación que tienden a ser mayores.

Ejemplo de una aplicación hecha con Xamarin, donde no si nota ninguna diferencia entre ella y una aplicación nativa.

## El problema de herramientas multiplataformas

Hasta ahora vimos que la mayor parte de las desventajas de las herramientas son la perca de la experiencia nativa, la lentitud, el tamaño de las aplicaciones, la dependencia de ambientes de desarrollo.

Pero el principal problema de esas herramientas sin duda es la lentitud y el tamaño de las aplicaciones. Mismo usando herramientas como *React Native*, que es una de las más usadas actualmente, en que la reutilización del código llega hasta 85%; la medida que la aplicación aumenta o va se tornando mas compleja, la herramienta tiende a dejar la aplicación mayor y más lenta.

Y así nasce Flutter, con el intuito de acabar con la mayor parte de esos problemas y continuar siendo un framework multiplataforma, donde se desarrolla “una” vez.

# FLUTTER

Flutter es una creación de la grande compañía Google. Según la propria compañía Flutter es el conjunto de herramientas de interfaz de usuario (UI) para crear hermosas aplicaciones compiladas de forma nativa para dispositivos móviles, web y de escritorio a partir de una única base de código. (PONER REFERENCIA FLUTTER.DEV).

Flutter diferente de Android, usa el lenguaje de desarrollo Dart; que también fue creada por Google en 2011 y tuve su primer lanzamiento estable en noviembre de 2013, cerca de dos años antes del lanzamiento de Flutter.

Flutter ha ganado vida en 2015 con el nombre de *Sky*. Al principio, por si tratar de una herramienta de Google, Flutter solo ejecutaba en dispositivos con el sistema operativo Android, pero después fue portado para iOS y hoy en día sigue cubriendo las dos plataformas que dominan este nicho de mercado.

Después de muchas versiones lanzadas desde el primer lanzamiento inicial en 2015, en diciembre de 2018 acontece el primer lanzamiento estable de Flutter. Y eso significa que la herramienta esta lista para que la comunidad (desarrolladores, compañías, dentro muchos otros) pueda empezar a utilizarla para crear y por en marcha aplicaciones reales.

## ¿Para qué sirve Flutter?

Flutter ha nascido con el mismo intuito de las herramientas a presentadas anteriormente. Una herramienta que agilice el proceso de desarrollo de software (móvil en este caso, es decir, a través de una base de código, ser posible compilar para multiplataformas, sin tener la necesidad de tener una base de código para cada plataforma.

Pero Flutter surge también con la necesidad de acabar con los problemas comunes a estés tipos de herramientas. Y sabiendo de eso, uno de los principales objetivos de Flutter es ser capaz de renderizar interfaces a una constancia de 120 FPS. Lo que posibilita bastante fluidez en las aplicaciones, donde realmente hay una experiencia nativa y la lentitud no es perceptible.

## Partes principales de Flutter

## ¿Porque utilizar Flutter?

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteFlutter diferentemente de la mayor parte de las otras herramientas disponibles, renderiza sus proprios componentes de interfaz. Tampoco usa componentes nativos, como la mayoría de las herramientas hace. Por ejemplo, cuando un desarrollador escribí un código para dibujar un botón en la pantalla, Flutter no hace una llamada para el SO para renderizar el botón, sino que Flutter renderiza el dicho botón por sí solo.

La imagen arriba muestra como React Native hace la renderización de un componente de botón. Debido a esa forma de trabajo, no solo de React Native, la medida que la aplicación crece acaba se tornando más lenta. Luego se nota que Flutter tiene una ventaja sobre esas herramientas, visto que Flutter hace su propria renderización de widgets¹.

Debido a su propria forma de renderizar componentes, Flutter puede proyectar sus proprios *widgets*. En otras palabras, Flutter no está atado al sistema operativo en se tratando de widgets. Eso permite a Flutter ofrecer dos kits de conjuntos de widgets: *Material Design* widgets (Android) y *Cupertino* widgets (iOS).

# Bibliografía

Benkler, Y. (2012). *El Pingüino y el Leviatán: Por qué la cooperación es nuestra arma más valiosa para mejorar el bienestar de la sociedad.*

Velasco, J. G. (2009). *Energías renovables.* Editorial Reverte.