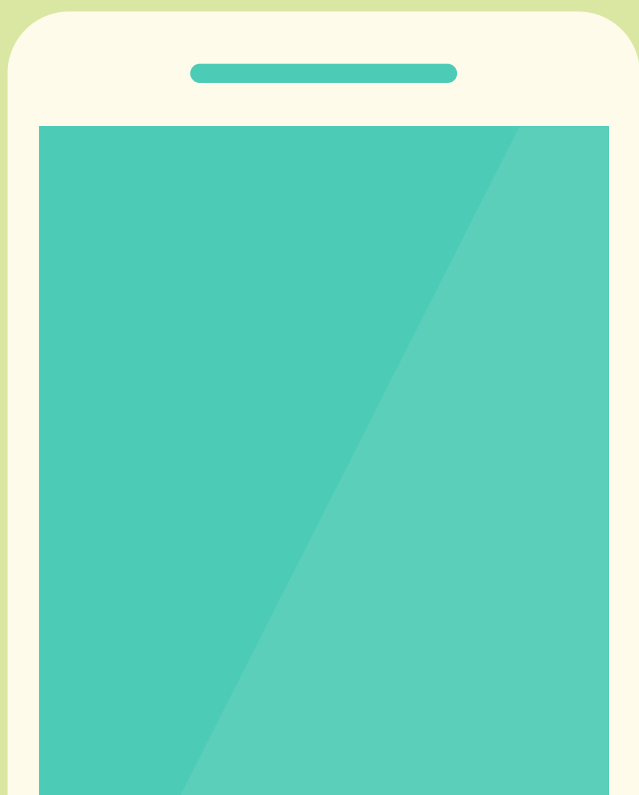


LENGUAJES PARA DESARROLLAR

ANDROID



POR LUCAS SOLARES - 2018

INTRODUCCIÓN

Este documento tiene como finalidad, explicar que lenguajes y frameworks son los más populares para desarrollar en Android, pero además de esos también se comunicaran las diferencias entre una aplicación nativa y una híbrida, clasificando los lenguajes y frameworks según estos aspectos.

También se planea opinar desde mi punto de vista las ventajas y desventajas que estos poseen, comparar los más modernos y sus características, con los mas antiguos, también se incluirán algunos links que redireccionaran tanto a la parte de la documentación del lenguaje o framework, como a algún curso gratuito del mismo.

Este documento también planea dar a conocer la filosofía que "No existe ningún lenguaje o framework mejor que otros, solo algunos que se adaptan a algunos problemas y otros a otros".

ÍNDICE

1	-----	Caratula
2	-----	Introducción
3	-----	Índice
4 <--> 5	-----	Nativas vs Híbridas
6 <--> 7	-----	Frameworks Híbridos
8 <--> 12	-----	Leguajes y Frameworks Nativos
13	-----	Conclusión
14	-----	E-grafía

NATIVAS VS HIBRIDAS

En la época moderna de la tecnología se ha buscado optimizar el desarrollo de aplicaciones, pero ¿En qué aspecto?, bueno si bien las aplicaciones solucionan problemas, surgió un problema mayor en lo que viene siendo los dispositivos que corren las mismas, recordemos que existen dos grandes IOS y Android, si bien ambos tienen como propósito ser soluciones móviles, para desarrollar apps se utilizan frameworks y lenguajes distintos, esto provoca que haya que contratar un equipo de desarrollo para Android y para IOS, pero lo peor de todo es que ambos proyectos por lo general no van a estar actualizados es decir mientras se está implementando algo en IOS puede ser que Android no, y así surgían esos problemas, ¿Cuál fue la solución?, simple crear un framework o lenguaje que se compile tanto a lenguaje Android como IOS, surgiendo los que son aplicaciones **Cross Platform**, básicamente aplicaciones con una sola base de código que corren en cualquier dispositivo, pero como es de esperar en la programación un problema solucionado genera otro problema, y este es, bajo rendimiento porque ahora no se desarrollaba de manera nativa sino que se hacían aplicaciones híbridas.

HIBRIDAS

Este tipo de aplicaciones básicamente se definen de la siguiente manera, la aplicación emula un navegador en el celular que es capaz de interactuar con el hardware del mismo, esto permitía que corriera en cualquier dispositivo incluso en un navegador, un ejemplo claro de esto es **IONIC**, pero no todo es color de rosa, a cambio de esta adaptación tan sencilla surgieron problemas de rendimiento y estabilidad, rápidamente bajando la popularidad de los frameworks.

NATIVAS

Las aplicaciones nativas son las que están escritas meramente en el lenguajes del sistema al que se van a desplegar, sin embargo esto no quiere decir que el lenguaje en el que se escriban sea el mismo, a que me refiero pues si fuera este el caso tendríamos nuestros proyecto en **Swift o Kotlin**, pero no este el caso ¿Por qué?, bueno hay frameworks que se encargan de si bien devolver código nativo, no escribirse en ese ejemplo de esto es **React Native y Flutter**.

IONIC

Este framework fue de los pioneros en la solución a múltiples proyectos, creado y mantenido por Google, permite la creación de aplicaciones móviles utilizando frameworks o librerías de UI conocidas como es el caso de **React y Angular**, es bastante robusto porque comenzó utilizando únicamente Angular que es un framework bastante robusto y una de sus principales ventajas es que todos los componentes para mobile como Navigation Bars, App Bars, etc., ya vienen implementados, sin embargo como desventajas podemos tomar el hecho de que al ser híbrido, posee un rendimiento que deja mucho que desear además de poseer la inestabilidad del mismo.



XAMARIN

Este lenguaje fue muy bien adoptado por las empresas, además que en países que no poseen una actualización constante en cuanto a tecnología respecta, se mantienen básicamente en los que es **Java**, **.NET** y **PHP**, por este motivo las empresas se dieron cuenta que si pueden crear su páginas en .NET y además también sus apps, aumentarían su rendimiento y este es el caso de Xamarin, un framework mobile basado en .NET mantenido y creado por Microsoft que te permite hacer aplicaciones **Cross-Platform**.



JAVA

El lenguaje por excelencia de Android, todos los frameworks o lenguajes que se pueden programar en Android su compilación final siempre es Java, este lenguaje fue inicialmente creado por Sun Microsystems para luego se comprado por Oracle, actualmente se encuentra en la versión 10, y aunque es software libre, las últimas versiones tienen limitaciones por derechos de autor.

Java es un lenguaje principalmente pensado para ser lenguajes orientado a objetos, y la mitad de los dispositivos a nivel mundial corren en este lenguaje, como por ejemplo Android.



KOTLIN

Cuando se comenzó a utilizar Java en el desarrollo mobile, los desarrolladores buscaron la manera de solucionar su vida, por lo que crearon un lenguaje que es tan poderoso para programar en Android y tan sencillo y explicito para programar, por este motivo Kotlin aumento muchas popularidad en este siglo, Kotlin además que este lenguaje no necesita compilarse a Java directamente ya que corre sobre la JVM del mismo.



C#

Aunque este lenguaje sirve perfectamente para programar aplicaciones en Android, no es por lo que tomo popularidad, sino por los videojuegos, ya que frameworks como Unity lo utilizan en su core, por tener la capacidad de comunicarse de manera sencillas con el hardware y dar toda la potencia de Java, siendo este muy parecido al mismo, pero con la potencia de lenguajes de bajo nivel como C, es desarrollado y mantenido por Microsoft, y es el lenguaje principal también en .NET.



REACT Y SVELTE NATIVE

Así como IONIC dio una solución muy efectiva en cuanto a mantener un solo proyecto mobile se refiere, nos dio el problema de las aplicaciones híbridas, pero este problema se soluciona en frameworks como React o Svelte en su versión nativa ya que por detrás esto se compila ya sea a Java o a Kotlin directamente, dando como resultado aplicaciones 100% nativas, pero con la facilidad de escribir React o Svelte.



FLUTTER

Este framework en mi opinion es el mejor para desarrollar en nativo multiplataforma no solo te incluye diseño de material design, sino que también al tener su propio renderizado de graficos como los SKIA, incluso corre en rendimiento mejor que una aplicación nativa. Flutter es una propuesta de Google para ser el único lenguaje realmente multiplataforma, tanto para desktop, mobile, web, etc.



CONCLUSIÓN

Desde mi punto de vista los desarrolladores deberían dejar de usar frameworks híbridos y moverse a cualquiera de los frameworks nativos, ¿A cuál? en realidad en ese caso no importa, en el que se sientan mas cómodos, da igual manera tendrán el mismo rendimiento en todos.

El mundo del desarrollo esta evolucionando a tal punto donde tener múltiples proyectos será ineficiente e innecesario, un claro ejemplo de esto es Flutter.

E-GRAFIA

<https://svelte-native.technology/>

https://es.wikipedia.org/wiki/C_Sharp

<https://reactnative.dev/>

<https://flutter.dev/>

<https://dotnet.microsoft.com/apps/xamarin>