# Aproximandonos a Flutter

# Android Estudio V3.0.17/Flutter

Fuente: [Fuente principal](https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514)

## Un poco de arquitectura

### El tiempo 0

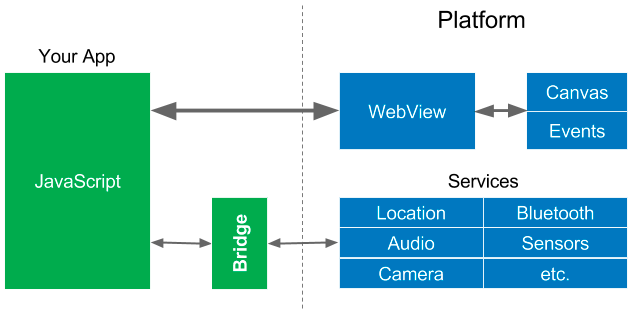
Fig:1

Ios se lanzo el 2008🡪objetive c

Y android el 2009🡪Java

Esta es una arquitectura simple, . prácticamente tiene que crear aplicaciones separadas para cada plataforma porque los widgets son diferentes, sin mencionar los idiomas nativos.

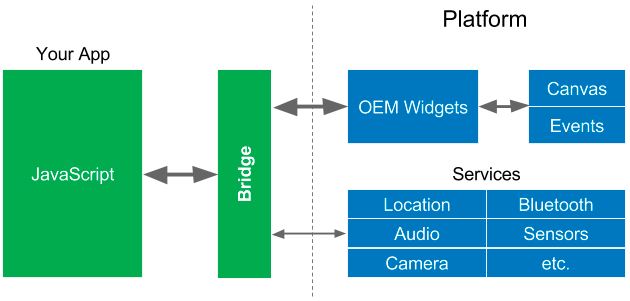
### marcos multiplataforma

Fig:2

Estaban básados sobre javaScript e incluian una familia de framework relacionadas:PhoneGap, Cordova y otras soluciones

Es dificil para javascript hablar directamente con el código nativo (como los servicios), por lo que pasan por un puente(“Bridge”), como los servicios de plataforma no se llaman con tanta frecuencia , no se causa problemas de rendimiento

### Marco Web reactiva

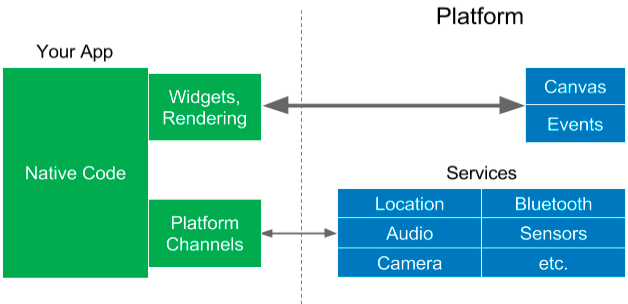


simplifican la creación de vistas web a través del uso de patrones de programación tomados de la programación reactiva. En 2015, React Native se creó para brindar los muchos beneficios de las vistas de estilo reactivo a las aplicaciones móviles.

La programación reactiva es un paradigma de programación que está orientado a los flujos de datos y más específicamente a la propagación de los cambios de los datos en dicho flujo.Es una evolución del patrón de diseño “Observador”.

Con animaciones, transiciones o pasando el dedo por encima se pueden tener problemas de rendimiento

## Flutter



Flutter adopta un enfoque diferente para evitar los problemas de rendimiento causados por la necesidad de un puente de JavaScript mediante el uso de un lenguaje de programación compilado, a saber, Dart

Dart se compila "antes de tiempo" (AOT) en código nativo para múltiples plataformas. Esto permite que Flutter se comunique con la plataforma sin pasar por un puente de JavaScript que hace un cambio de contexto. La compilación con código nativo también mejora los tiempos de inicio de la aplicación

### Los Widget

Los widgets son los elementos que afectan y controlan la vista y la interfaz con una aplicación. No es una exageración decir que los widgets son una de las partes más importantes de una aplicación móvil. De hecho, los widgets solo pueden hacer o deshacer una aplicación.

Flutter no utiliza los widgets OEM (o DOM WebViews), sino que proporciona sus propios widgets.

Todo lo que Flutter requiere de la plataforma es un canvas(“lienzo”) en el que mostrar los widgets para que puedan aparecer en la pantalla del dispositivo y acceder a eventos (toques, temporizadores, etc.) y servicios (ubicación, cámara, etc.)….pueden llegara ha ser ordenes de magnitud más r+apidos que javaScript

El tamaño mínimo de una app es de 6.7 mb comparable a las otras herramientas.

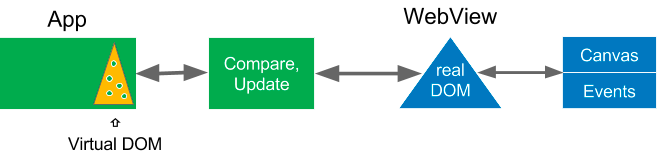
### Layout

Layout determina el tamaño y la posición de los widgets según un conjunto de reglas (también llamadas restricciones). Las reglas implementan múltiples métodos de diseño.

Debido a que los widgets ahora son parte de la aplicación, se pueden agregar nuevos widgets y los widgets existentes se pueden personalizar para darles una apariencia diferente, o para que coincida con la marca de una compañía.

Flutter viene con conjuntos de widgets ricos y personalizables para Android, iOS y Material Design (de hecho, nos han dicho que Flutter tiene una de las implementaciones de mayor fidelidad de Material Design). Usamos la personalización de Flutter para construir estos conjuntos de widgets, para que coincida con la apariencia de los widgets nativos en múltiples plataformas. Los desarrolladores de aplicaciones pueden usar la misma personalización para ajustar aún más los widgets a sus deseos y necesidades.

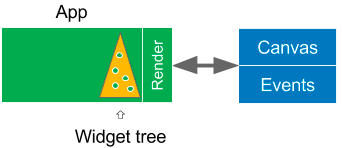
### Web reactivas



El DOM(Document Object Model) es una api usada por JavaScript para manipular HTML, representando un árbol de elementos.En las web reactiva,el DOM virtual es inmutable.

EL virtual DOM es comparado al DOM real para generar un conjunto de cambios mínimos, los cuales son ejecutado para actualizar el DOM real.Finalmente la plataforma punta le DOM real sobre un canvas.

React Native hace una cosa similar,pero para apps moviles, en lugar del DOM , este manipula los widget natiivos de la plataforma movil, en vez del DOM virtual .este construye un árbol virtual de widget y compara este con los widget nativos y solamente actualiza estos que han cambiado.Flutter va un paso más adelante



Recordemos que en Flutter, los widgets y el renderizador se han subido de la plataforma hacia la aplicación del usuario. No hay widgets OEM originales para manipular, entonces, ¿qué era un árbol virtual de widgets? Flutter ahora pinta sobre un canvas el árbol de widget,internamente usa varios árboles para pintar solamente lo necesario

El lenguaje de programación Dart

Flutter usa objetos que pueden durar un frame(un sexagesimo de segundo), por lo cual se va creando mucha basura, dart usa un [generational garbage collection](https://en.wikipedia.org/wiki/Tracing_garbage_collection#Generational_GC_.28ephemeral_GC.29), lo cual es barato para este tipo de sistemas, ya que no requieren de bloqueo

Dat tambien usa un compilador [tree shaking](https://medium.com/@Rich_Harris/tree-shaking-versus-dead-code-elimination-d3765df85c80) ,el cual usa solamente el código que necesitas en la app, por lo cual se pueden usar chorrocientas librerias sin problemas.Puede ser interesante leer [Why Flutter uses Dart](https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf)

## Hot reload

Se pueden hacer cambios a la app míentras esta corriendo, este recarga el código cambiado y prosigue

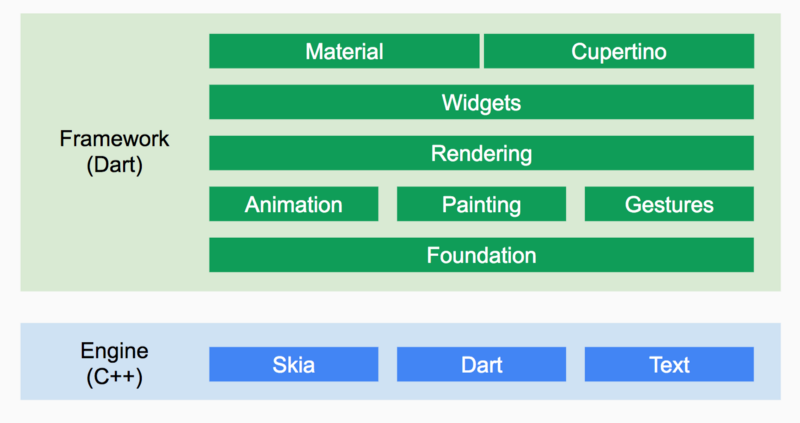
## Compatibilidad

Sus aplicaciones no solo funcionarán, sino que funcionarán igual en versiones recientes del sistema operativo: Android Jelly Bean y versiones más recientes e iOS 8.0 y posteriores. Esto reduce significativamente la necesidad de probar aplicaciones en versiones anteriores del sistema operativo. Además, es probable que sus aplicaciones funcionen en futuras versiones del sistema operativo.

## Otras gracias

Tiene [repository of software packages](https://pub.dartlang.org/flutter/packages/) para extender las capacidades de tu Software, hay paquetes que hace más fácil entrar [access Firebase](https://github.com/flutter/plugins/blob/master/FlutterFire.md) a por ejemplo

Cada cosa en verde puede ser customizada

.