

Mercado De Aplicaciones Híbridas: Ionic Vs React Vs Flutter

Víctor Barrios Álvarez
Arquitecto De Software



vbarrios1512



vbalvarez15

Contenido

01

Aplicaciones Híbridas y su marco teórico

Conceptos Generales

02

Ionic vs React Vs Flutter

Ventajas , desventajas y Oportunidades

03

¿Cómo está el mercado?

¿Cual es mejor? ¿Hay un favorito?

01

Aplicaciones Híbridas y su marco teórico

Conceptos Generales

Time To Market

La principal ventaja de la app híbrida es que con el mismo código podemos desplegar en diferentes dispositivos y sistemas operativos: iOS, Android, web...esta característica las convierte en una opción más económica frente a una app nativa.

¿Cómo Elijo Invertir Entre Híbrido y Nativo?

Aspecto

A Evaluar

Descripción

4 Características



Público

Objetivo de la App
(Importancia
Alta)

El portafolio nativo es más completo

Una aplicación nativa es rica en componentes, interfaces de usuario, usabilidad, servicios de terceros, librerías y formas de personalización. Una aplicación híbrida solo te provee hasta donde el entorno de desarrollo (PhoneGap, Cordova, Capacitor) te permita.



Core

Del Negocio
(Importancia
Alta)

No todas las organizaciones van bajo las mismas necesidades

Todos los negocios y sus soluciones van por lo general, orientados hacia un foco, el cual puede hacer participe a este tipo de aplicativos o no.

¿Cuándo es ideal una App Híbrida?

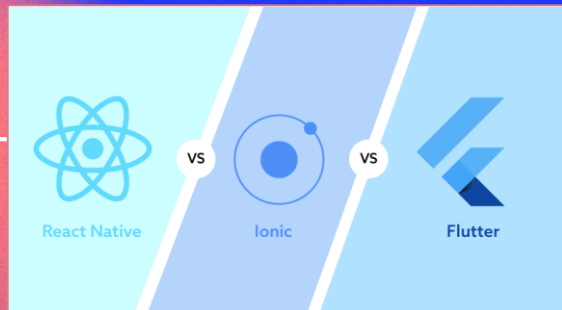
¿Quieres saber cuando es conveniente la realización de una app Híbrida?

Veamos los siguientes aspectos:

Poca Madurez e inversión de tu proyecto	MVP (Producto Mínimo Viable): Aquí no haya restricciones técnicas, con menos inversión económica podremos lanzar al mercado una aplicación que sirva para diferentes dispositivos y validar qué tal acogida tiene cuanto antes.
Uso básico o mediano de las características del dispositivo	En el caso de que quieras desarrollar un juego o alguna funcionalidad que utilice a fondo las características del dispositivo, probablemente te interesará nativa.
La rapidez no es el objetivo primario	La principal desventaja de elegir una app híbrida es que usa frameworks intermedios al sistema operativo ralentizando la ejecución.

02

Ionic vs React Vs Flutter



Ventajas , desventajas y Oportunidades

Estructura Interna



HTML, CSS, JavaScript
(Vanilla, Angular, React o Vue)

Creado en 2013
Drifty Co

Implementan canal de comunicación nativa mediante Cordova, Ionic Native y Capacitor a través de WebView.



JavaScript
(React)

Creado en 2015
Meta Platforms, Inc.

Uso de JSX y "VirtualDOM" para generación de componentes nativos. No hay WebView.

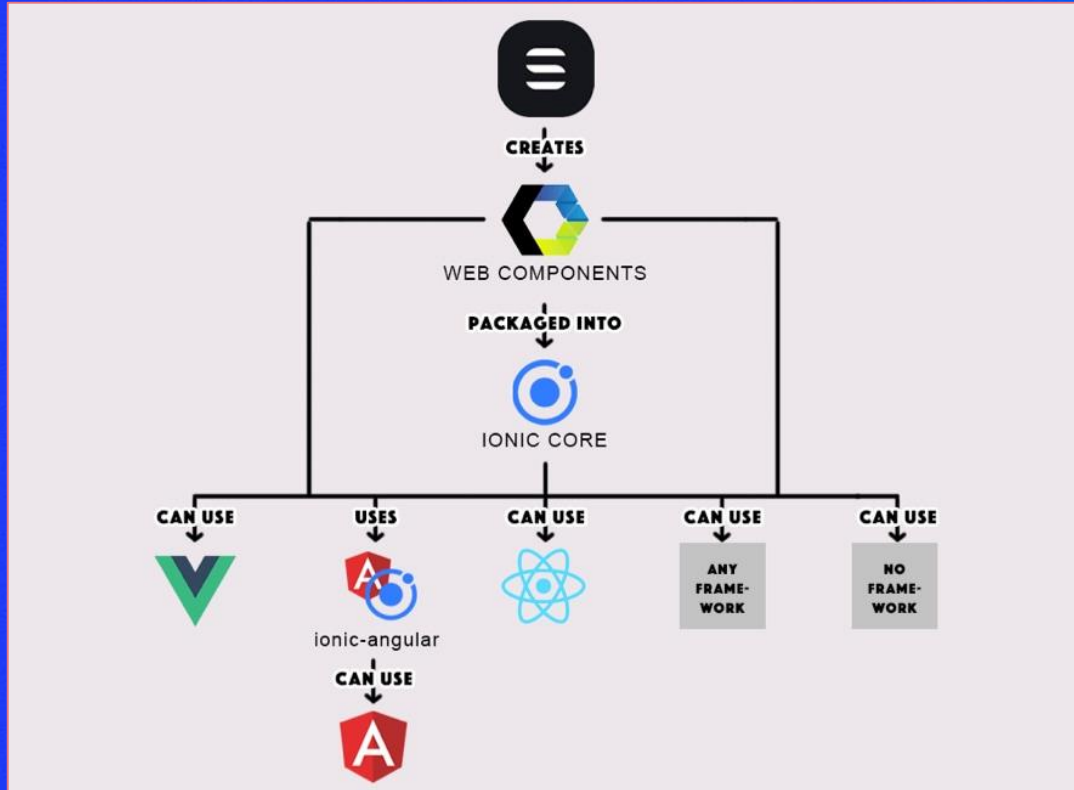


Dart
(Alternativa Moderna a JavaScript)

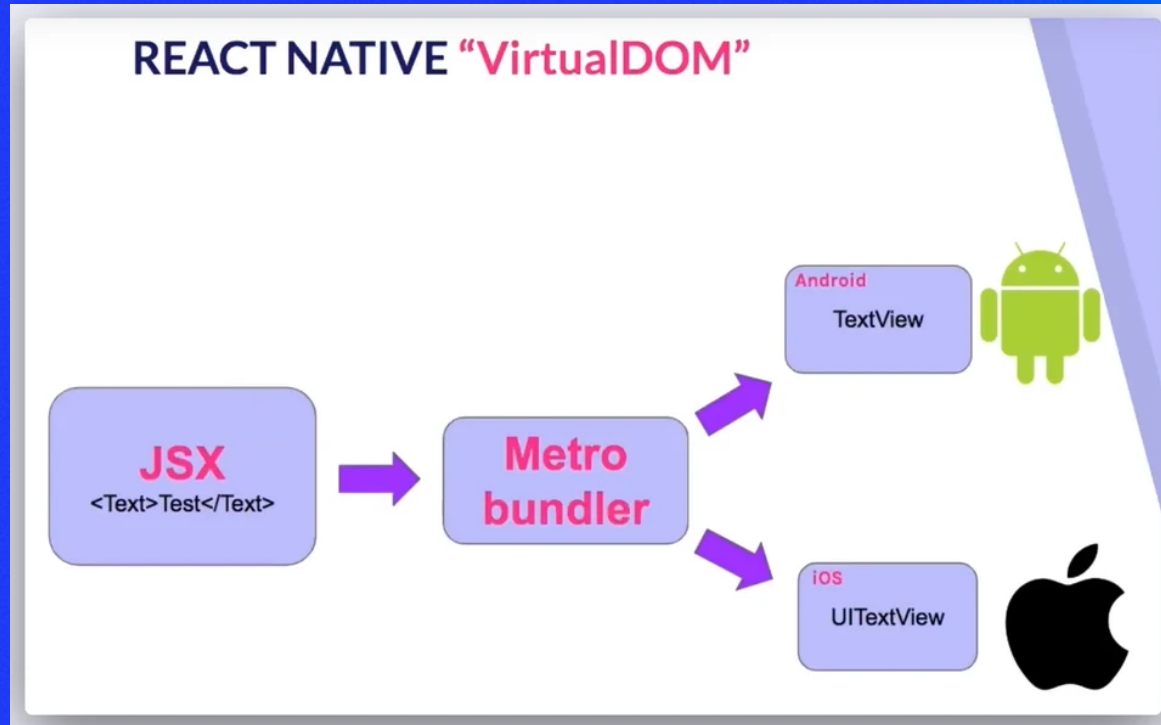
Creado en 2017
Google

Conversión de código al 100% Nativo Mediante Flutter Engine (C++).

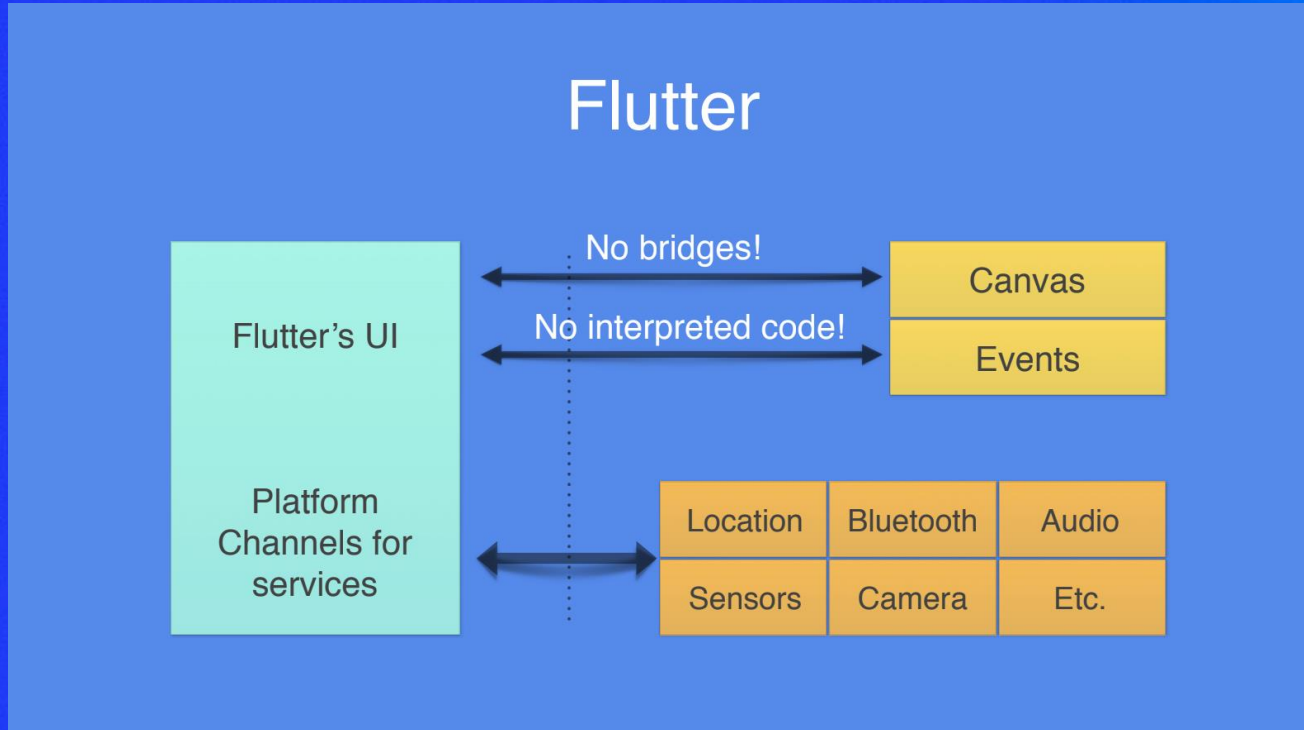
Logística De Ionic



Logística De React Native



Logística De Flutter



03

¿Cómo está el mercado?

¿Cual es mejor? ¿Hay un favorito?

En la vida no hay favoritos. Hay
quienes cubren más necesidades que
otros.

— **Frase Personal**

Atributos Para Elegir al “Favorito”



Lenguajes

Mi conocimiento



Reutilización

¡No repitas!



Biblioteca De Componentes

El portafolio de la experiencia de usuario



Performance

¿Rendimiento o no?



Usabilidad

Interfaces de usuario

Atributos Para Elegir al “Favorito”



Ecosistema

Servicios de Terceros



Mercado y comunidad

¿Quiénes me apoyan? ¿A quiénes voy a unirme?



Nivel Nativo

¿Qué tanto usaré de las características del dispositivo?



Plataformas Compatibles

Mi foco de trabajo hacia el cliente final



Aplicaciones Populares

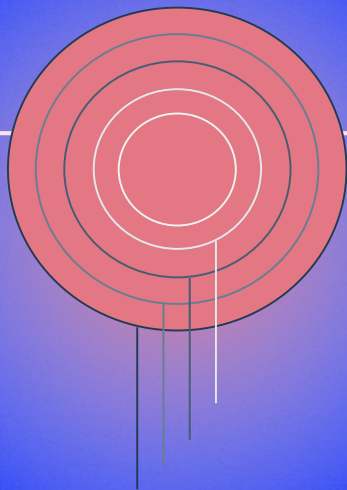
La experiencia y el recorrido



Precios

Mi necesidad es lo que poseo en el momento.

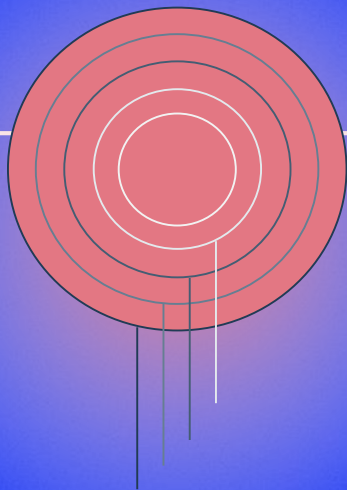
Lenguajes



Mi conocimiento

- React Native: Usa JavaScript y lo combina con React JS. Es respaldado por Facebook. Permite modificar hasta cierto punto, componentes nativos en Swift, Objective C o Java.
- Ionic: Utiliza HTML5, CSS y JavaScript o TypeScript, en complemento con el envoltorio de Apache Cordova para acceder a los controladores nativos. Compatible con Angular, React JS y Vue JS.
- Flutter: Utiliza DART, que está basado en C/C++ y Java. Se proyecta como un lenguaje Futurista.

Reutilización



¡No repitas!

- React Native: Solo proporciona un conjunto básico de componentes para comenzar. Tienes que diseñar la mayoría de ellos por tu cuenta, por lo tanto, se requiere más trabajo para lograr estilos apropiados en ambas plataformas. Sin embargo, el código se puede reutilizar.
- Ionic: Al ser un marco embebido en un WebView, los componentes son fáciles de reutilizar.
- Flutter: Al ser de Google, la mayoría de los estilos corresponden a Material Design. Por lo tanto, tienes que diseñar gran parte de ellos por tu cuenta, sobre todo para iOS.

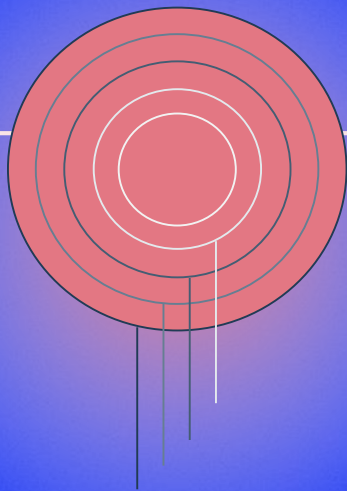
Biblioteca De Componentes

El portafolio de la experiencia de usuario

- React Native: Proporciona un conjunto base de componentes integrados, pero usted debe diseñar muchos de ellos. Y no son adaptables, en cambio, al igual que con Flutter, obtienes alternativas para ambos sistemas operativos. Eso requiere ajustes en su código, donde tiene que elegir, usar y diseñar widgets condicionalmente.
- Ionic: Tiene un gran conjunto de componentes precontruidos.. Los componentes se adaptan automáticamente a la plataforma en la que se ejecuta la aplicación y, por lo tanto, hace que la creación de aplicaciones de aspecto nativo sea muy sencilla.
- Flutter: Viene con un conjunto completo de widgets integrados. Estos utilizan principalmente Material Design, también existen algunos widgets de estilo Cupertino (estilo iOS). Sin embargo si necesita diferentes looks para diferentes plataformas, puede requerir un poco de esfuerzo ya que los widgets no se adaptan automáticamente.

Performance

¿Rendimiento o no?



- React Native: El rendimiento que proporciona es muy similar al de las aplicaciones nativas. React también permite a los desarrolladores usar módulos nativos escritos en idiomas nativos para escribir código para operaciones complicadas. Sin embargo, no se pueden reutilizar en dos plataformas
- Ionic: No es muy rápido, dado que es por modalidad WebView. No tiene componentes nativos, solo apariencia nativa.
- Flutter: En cuanto al rendimiento, se lleva la corona sobre sus competidores. Debido a que tiene las ventajas de Dart y no hay un puente de JavaScript para iniciar interacciones con los componentes nativos del dispositivo, la velocidad que ofrece es sorprendente.

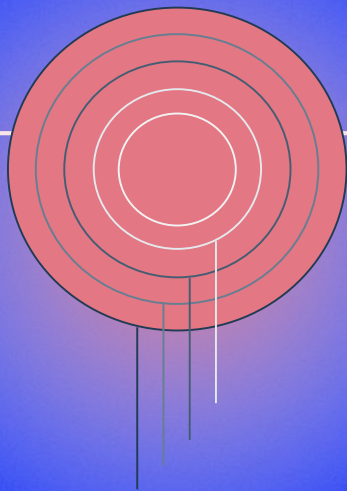
Usabilidad

Interfaces de usuario

- React Native: Los módulos React Native asociados con los controladores de interfaz de usuario nativos brindan una experiencia de usuario increíble que está muy cerca de las aplicaciones nativas. Además, utiliza la biblioteca ReactJS con amplios elementos de interfaz de usuario y agiliza el desarrollo de la interfaz de usuario.
- Ionic: La interfaz de usuario de Ionic no utiliza elementos nativos en absoluto y presenta todo en HTML y CSS. Luego aplica Cordova para brindar una experiencia móvil nativa.
- Flutter: Suministra las mejores interfaces de usuario, al ser una comunicación nativa en doble vía.

Ecosistema

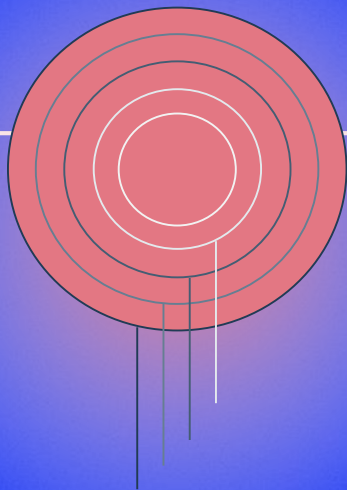
Servicios de terceros



- React Native: Los servicios de terceros quedan en desuso constantemente, ya que React Native tiene un muy buen ritmo de trabajo en sus actualizaciones de versión.
- Ionic: Posee una comunidad extensa en NPM y Angular.
- Flutter: Los servicios de terceros están en crecimiento y cada día lanzando nuevos servicios. La comunidad aun no es muy extensa pero tiene una gran proyección.

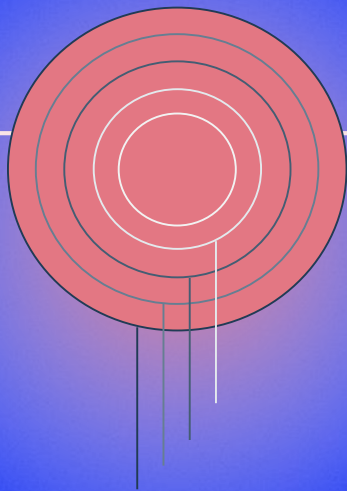
Mercado y comunidad

¿Quienes me apoyan? ¿A quienes voy a unirme?



- React Native: Además de ser reconocida por Facebook, tiene un gran mercado de desarrolladores experimentados. Esto, la convierte en una plataforma Sólida.
- Ionic: Viene arrastrando el mercado y comunidad de PhoneGap y Apache Cordova. Tiene una gran población de desarrolladores de apoyo continuo. Asi como servicios de templates gratuitos.
- Flutter: Al ser una tecnología reciente, aun no es muy popular. Pero Google lo anuncia intensamente, lo que indica que será un buen modelo para el mundo móvil.

Nivel Nativo

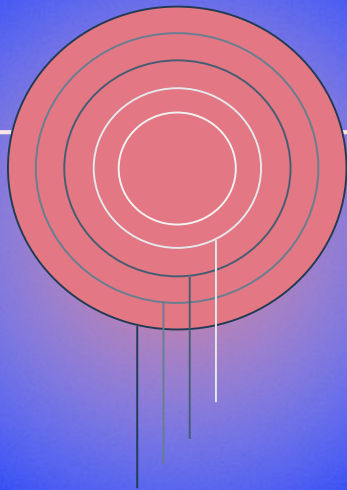


¿Qué tanto usaré de las características del dispositivo?

- React Native: Posee una gran cantidad de paquetes de terceros muy útiles para integrar con funciones nativas. Sin embargo, depende de la comunidad que cree dichos paquetes o los actualice.
- Ionic: Utiliza comunicación nativa mediante Apache Cordova o Capacitor. Sin embargo, depende de la comunidad que cree dichos plugins o los mantenga actualizados.
- Flutter: Hay un ecosistema muy vibrante y, por lo tanto, encontrará un paquete para prácticamente cualquier función nativa a la que desee acceder.

Plataformas Compatibles

Mi foco de trabajo hacia el cliente final



- React Native: Android 4.1+, iOS 8+
- Ionic (V6): Android 5.0+, 13.0+
- Flutter: Android 4.1+, iOS 8+

Aplicaciones Populares

La experiencia y el recorrido

- React Native: Facebook, Instagram, UberEats, Airbnb (Versiones iniciales).



- Ionic (V6): JustWatch, Sanvello, Nationwide

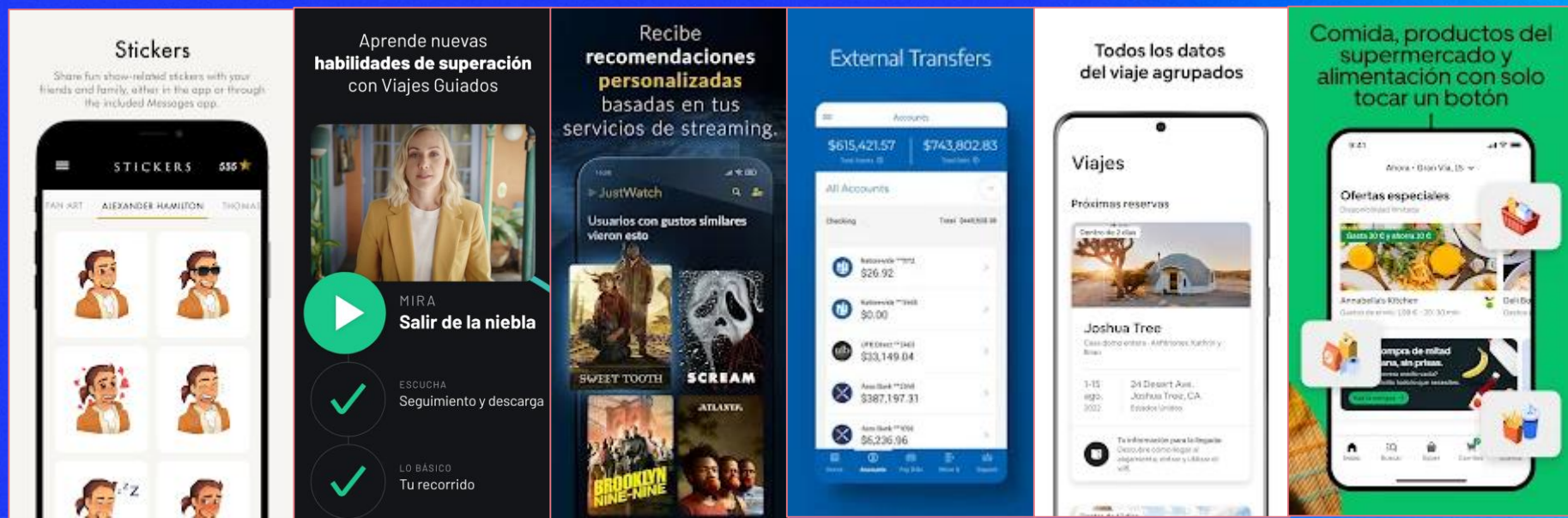


- Flutter: Hamilton

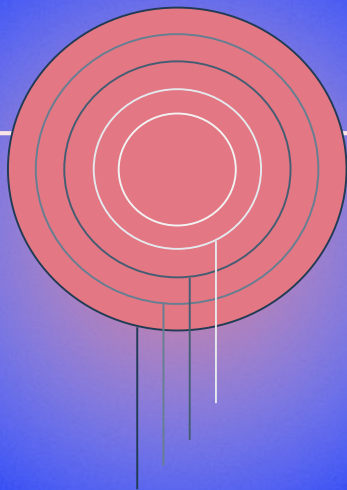


Aplicaciones Populares

La experiencia y el recorrido



Precios



Mi necesidad es lo que poseo en el momento.

- React Native: Marco de código abierto. Componentes, librerías y servicios están de forma gratuita.
- Ionic: Ofrece un entorno de código abierto, pero con ciertas limitaciones. Para un entorno de desarrollo más profesional, se requiere pagar.
- Flutter: Marco de código abierto. Componentes, librerías y servicios están de forma gratuita.

Gracias

La inteligencia consiste no sólo en el
conocimiento, sino también en la destreza
de aplicar los conocimientos en la práctica.

Aristóteles (384 AC-322 AC) Filósofo griego.

