

DOI: [https://doi.org/10.37811/cl\\_rcm.v6i4.2950](https://doi.org/10.37811/cl_rcm.v6i4.2950)

## Analisis comparativo de entornos de desarrollo móvil

Luis René Quisaguano

[luis.quisaguano1@utc.edu.ec](mailto:luis.quisaguano1@utc.edu.ec)

<https://orcid.org/0000-0003-1345-0898>

Tannia Guissela Camalle

[tannia.camalle3@utc.edu.ec](mailto:tannia.camalle3@utc.edu.ec)

<https://orcid.org/0000-0002-8187-8047>

José Wladimir Toca

[jose.toca9367@utc.edu.ec](mailto:jose.toca9367@utc.edu.ec)

<https://orcid.org/0000-0002-9711-0614>

Universidad Técnica De Cotopaxi

Latacunga-Ecuador

Correspondencia: [luis.quisaguano1@utc.edu.ec](mailto:luis.quisaguano1@utc.edu.ec)

Artículo recibido: 15 julio 2022. Aceptado para publicación: 20 agosto 2022.

Conflictos de Interés: Ninguna que declarar

Todo el contenido de **Ciencia Latina Revista Científica Multidisciplinar**, publicados en este sitio están disponibles bajo Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/).

Como citar: René Quisaguano, L., Camalle, T. G., & Toca, J. W. (2022). Análisis comparativo de entornos de desarrollo móvil. *Ciencia Latina Revista Científica Multidisciplinar*, 6(4), 4478-4498.

[https://doi.org/10.37811/cl\\_rcm.v6i4.2950](https://doi.org/10.37811/cl_rcm.v6i4.2950)

### RESUMEN

El desarrollo de aplicaciones móviles mantuvo un año de crecimiento significativo después de años, aunque para las empresas relativamente pequeñas y de desarrollo de software, es difícil abordar el desarrollo móvil de forma nativa, con desarrolladores especializados en ambas plataformas o con diferentes equipos de desarrollo paralelo. Por este motivo, se han popularizado los entornos de desarrollo que permiten realizar el desarrollo y el mantenimiento utilizando código común, pero generando código separado para plataformas diferentes. Sin embargo, cuando una empresa decide utilizar un entorno de desarrollo multiplataforma, a menos que decida volver a desarrollar una aplicación existente desde cero, es tentador poder migrar al menos parte de la aplicación. mencionado nuevo entorno de desarrollo. Android es la plataforma móvil con el mayor número de aplicaciones desarrolladas de forma nativa. Por otro lado, Flutter es un marco de desarrollo multiplataforma lanzado recientemente por Google, además de tener un soporte incluido.

**Palabras clave:** Android; Xamari; Flutter; Migración de aplicaciones; Desarrollo multiplataforma.

## Comparative analysis of mobile development environments

### ABSTRACT

Mobile app development retained significant growth year after year, although for relatively small and software development companies, it is difficult to approach mobile development natively, with developers specializing in both platforms, or with different parallel development teams. For this reason, development environments have become popular that allow development and maintenance to be performed using common code, but separate code will appear for different platforms. However, when a company decides to use a cross-platform development environment, unless it decides to redevelop an existing application from scratch, it is tempting to be able to migrate at least part of the application. mentioned new development environment. Android is the mobile platform with the largest number of natively developed applications. On the other hand, Flutter is a recently released cross-platform development framework by Google, as well as having built-in support.

**Keywords:** Tags: Android; Xamari; Flutter; App migration; Cross-platform development.

## INTRODUCCIÓN

La naturaleza de las relaciones humanas y los patrones de comunicación entre personas han cambiado por completo con la llegada de las tecnologías de la información y la comunicación (TIC). Los datos y la información en un espacio casi infinito llamado Internet ingresan a Internet. Desde la llegada de los teléfonos inteligentes, parece casi posible llevar el mundo en el bolsillo simplemente integrando todas las formas posibles de información y comunicación en un solo dispositivo.

Los dispositivos móviles como los teléfonos inteligentes (smartphones) y las tabletas son un factor importante en el cambio de la forma en que las personas y las organizaciones interactúan con ellos, abriendo un sinfín de posibilidades en una variedad de áreas comerciales. Este trabajo tiene como objetivo analizar estas oportunidades y consta de tres partes principales.

El mundo de hoy, que es tan cambiante y acelerado, difiere de otros tiempos en sus patrones culturales, tradiciones, comportamientos y, lo más importante, las relaciones interpersonales que se manifiestan en el hogar y la escuela, padres y maestros parecen interesados en lo que están pasando y ellos también. El fenómeno de estos rápidos cambios en la sociedad global adquiere un nuevo significado. Parece cambiar la vida misma y cambiar radicalmente el mundo.

Para intentar dar respuesta a los cuestionamientos, se ha planteado los siguientes objetivos:

1. Investigar los principales entornos de desarrollo móvil que se encuentra en auge.
2. Indagar sobre el uso de cada una de ellas dentro del desarrollo de aplicaciones móviles
3. Obtener resultados sobre los entornos de desarrollo móvil más usados en los últimos años.

Tradicionalmente, cuando hablamos de tecnologías multiplataforma, pensamos en un enfoque minimalista: reducir nuestra implementación/desarrollo al mínimo denominador en todas las plataformas. Esto es lo que se ofrece en aplicaciones híbridas desarrolladas con tecnología tipo PhoneGap/Cordova, donde el código html y javascript escrito debe ser interpretado por el navegador web. Estos tipos de soluciones no pueden igualar los comentarios y la experiencia del usuario de una aplicación nativa. Para dar un ejemplo, Mark Zuckerberg (CEO de Facebook), admite que el mayor error que cometieron en la historia de Facebook fue crear inicialmente una aplicación híbrida en lugar de una aplicación nativa.

El desarrollo de aplicaciones móviles es actualmente un gran desafío, debido a los requisitos específicos y las limitaciones técnicas del entorno móvil, tales como dispositivos con cambios limitados pero constantes; diferentes estándares de red, protocolos y tecnologías, la necesidad de operar en diferentes plataformas, requisitos específicos del usuario y estrictos requisitos de tiempo del mercado. (Thomas,2022)

## **METODOLOGÍA**

Para crear este documento, partimos de un enfoque cualitativo a partir de nuestra investigación anterior. Porque esta metodología nos permite analizar las principales características del tema en discusión y formular una propuesta que ayudará a nuestros lectores a comprender. Una interpretación más amplia de los conceptos que rodean el proceso de desarrollo basado en diferentes entornos móviles.

También se emplea una estrategia documental mediante la cual se realiza una búsqueda bibliográfica de carácter intelectual relevante al tema a través de búsquedas bibliográficas de revistas, artículos y sitios web, y se analiza su contenido para orientar mejor la estructura y así poder tener un mejor conocimiento del tema que se está tratando.

El investigador no solo debe ser capaz de conceptualizar el problema, sino también redactarlo de manera clara, precisa y accesible. A veces sabe lo que quiere hacer, pero no sabe cómo comunicárselo a los demás, y tiene que esforzarse más en traducir sus pensamientos en términos comprensibles, porque hoy en día la mayoría de las encuestas requieren la cooperación de muchas persona (Hernández, 2014)

## RESULTADOS

- FLUTTER

### La nueva era de flutter.

Para crear un proyecto Flutter primero se instala el SDK de Flutter para Windows que se encuentra en: <https://flutter.dev/docs/get-started/install/windows>. Se utiliza el editor de código Visual Studio Code y es necesario descargar las extensiones: Flutter, Awesome Flutter Snippets, Flutter Widget Snippets, Flutter Color y Dart. En Android Studio se deben instalar los plugins Flutter plugin y Dart plugin. En Visual Studio Code se construye el código desde la paleta de comandos escribiendo 'Flutter: new Project'. Si existen problemas para crear el proyecto, en la paleta de comandos se coloca 'Flutter: Run Flutter Doctor' para ver que dependencias falta por instalar.(Guamba,2021)

### Flutter

Flutter es el framework por excelencia diseñado para la interacción con Dart, optimizando la sintaxis utilizada por su lenguaje raíz. Posee el mejor tiempo de ejecución dentro de los frameworks ofertados en el mercado además de integrar múltiples herramientas gratuitas fáciles y livianas de utilizar para depurar y debuggear el código. Flutter transforma el proceso de desarrollo de aplicaciones desde una perspectiva cómoda y eficiente para los desarrolladores ya que posee librerías fáciles y cómodas de usar e instalar desde un repositorio virtual, con el soporte de cientos de desarrolladores a nivel mundial que realizan constantes contribuciones a la comunidad.(Pilalumbo,2019)

Para implementar una o más dependencias (librerías) para el proyecto que se esté desarrollando, únicamente basta con añadir la librería correspondiente a la necesidad de los programadores, mediante una sencilla búsqueda dentro del repositorio virtual de Google Flutter <https://pub.dev>(Pilalumbo,2019)

### Lenguaje de programación de flutter

En Flutter, todas las aplicaciones están escritas con Dart. El cual es un lenguaje de programación desarrollado y mantenido por Google.. Es ampliamente utilizado dentro de

Google y se ha demostrado que tiene la capacidad de desarrollar aplicaciones web masivas, como AdWords.

## **Dart**

Se desarrolló originalmente como reemplazo y sucesor de JavaScript. Por lo tanto, implementa la mayoría de las características importantes del próximo estándar de JavaScript (ES7), como las palabras clave "async" y "await". Sin embargo, para atraer a los desarrolladores que no están familiarizados con JavaScript, Dart tiene una sintaxis similar a Java.

### **Los objetivos de diseño de Dart fueron:**

- Crear un lenguaje estructurado pero flexible para la programación web.
- Hacer que Dart sea familiar y natural para los programadores, por lo que es fácil de aprender.
- Asegúrese de que Dart funcione bien en todos los entornos y navegadores web modernos, desde pequeños dispositivos móviles hasta la ejecución del lado del servidor.

Al igual que otros sistemas que utilizan vistas reactivas, la aplicación Flutter actualiza el árbol de vistas en cada cuadro nuevo. Este comportamiento genera el inconveniente de que se crearán muchos objetos, que pueden durar solo un fotograma. Dart, como lenguaje de programación moderno, está optimizado para manejar este escenario en el nivel de memoria con la ayuda de "Recolección de basura generacional" (Wu, 2014)

## **Widgets**

Los widgets son pequeños programas informáticos que realizan funciones específicas y suelen estar incluidos en la pantalla del ordenador o en páginas web. También son conocidos por términos equivalentes como componente de 2 páginas, "startlets", "widgets", "web badges" o "módulos", aunque el nombre más comúnmente aceptado es widget esperado (Perez, 2022).

En lugar de reutilizar los widgets OEM, tal como lo hace React Native, el equipo de Flutter decide proporcionar sus propios widgets. Esto significa que Flutter, como plataforma, puede decidir cuándo y cómo se representan los widgets. En cierto modo, Flutter mueve los widgets y el renderizador del nivel del sistema a la propia aplicación, lo que les permite

ser más personalizables y extensibles (Carlos,2020). Sin embargo, tener los widgets y el renderizador dentro de la aplicación aumenta el tamaño de la aplicación.

Hay dos tipos principales de widgets en Flutter, que son el widget sin estado y el widget con estado. (Wu,2014).

### **Base de datos integrada sqlite**

La base de datos con la que trabaja Flutter es SQLite ya que nos permite trabajar de forma fácil pero dentro del lenguaje *Dart*

- **SQLite**

SQLite es una base de datos integrada de código abierto escrita por D. Richard Hipp en 2000 usando lenguaje C. Es completamente independiente sin dependencias externas que pueden usarse más fácilmente en sistemas integrados, ejecutarse en todos los principales sistemas operativos y admite la mayor parte del lenguaje informático(Carlos,2020).

- **Características**

Este Framework tiene la característica que en lugar de utilizar widgets propios de la plataforma para la cual se está desarrollando, mediante su propio motor de renderizado, construye cada componente de vista.

Este comportamiento y el beneficio que ofrece Dart con la compilación AOT (Ahead of Time), brinda la posibilidad de crear aplicaciones que tengan un rendimiento igual o muy cercano al nativo

- **Ventajas y desventajas**

#### **Desventajas**

- ✓ No es 100% nativo
- ✓ Es muy nueva solo puede hacer aplicaciones 2D

#### **Ventajas**

- ✓ Es multiplataforma.
- ✓ Su velocidad es de 120fps.
- ✓ Es nuevo y brinda una mejor oportunidad laboral.

- ✓ Es muy robusto.
- ✓ Es gratuito

### Definición Xamarin

Es un entorno de desarrollo que le permite crear aplicaciones multiplataforma para Android, iOS y Windows utilizando un lenguaje de programación común, en este caso C#. No es necesario aprender el lenguaje de una plataforma en particular, es decir, le facilita la vida al programador ya que solo puede trabajar conociendo las características y el lenguaje de programación de Xamarin.

Hay varios marcos integrados de Xamarin, sin embargo, el más recomendado es Visual Studio, que es propiedad de Microsoft<sup>3</sup>



*Ilustración 1 Funcionamiento de Xamarin*

### Definición de xaml

XAML se ha portado a varias tecnologías de .NET Framework, sobre todo en Windows Presentation Foundation (WPF), donde XAML se usa para definir el diseño de la interfaz de usuario. XAML también se ha portado a Silverlight, Windows Runtime, Universal Windows Platform (UWP) y Xamarin.Forms. En el archivo XAML, los desarrolladores de Xamarin.Forms pueden definir la interfaz de usuario mediante controles (vistas), diseños, páginas o clases personalizadas.

Los archivos XAML se pueden compilar o incrustar en un ejecutable. En ambos casos, la información XAML se analiza en tiempo de compilación para ubicar objetos con nombre y nuevamente en tiempo de ejecución para inicializar e inicializar objetos y dispositivos, así como para poner restricciones entre esos objetos y el código



Xamarin es una empresa de software adoptada por Microsoft para ofrecer aplicaciones originales nativas para plataformas Android, Windows e iOS. Xamarin es parte de la plataforma .NET de código abierto que cuenta con una sólida comunidad de colaboradores de más de 3700 empresas. (Mohammed,2019)

WPF es la abreviatura de Windows Presentation Foundation. Es una parte del framework de desarrollo .NET. Pero no una parte cualquiera, es la más grande de todas ellas. Abarca muchos conceptos desde interfaces de usuario con XAML hasta patrones de diseño como MVVM. De todo esto hablaré en el siguiente artículo.(Hernandez,2016).

### **¿Con qué lenguajes de programación trabaja xamarin?**

Xamarin es una herramienta para desarrolladores de aplicaciones móviles, y la novedad de esta herramienta es la capacidad para que los desarrolladores escriban su aplicación en lenguaje C# y el mismo código se puede traducir para que funcione en iOS, Android y Windows Phone (Univision, 2014)

### **Objective-C, Java, C y C.**

Permite el uso de código de terceros en nuestras aplicaciones para código específico de plataforma. Compatibilidad multiplataforma móvil: brinda compatibilidad con las tres principales plataformas móviles, con una API unificada que brinda acceso a los recursos de las tres. Reducir el tiempo de desarrollo de aplicaciones(Muñoz, 2018).

## **VENTAJAS Y DESVENTAJAS**

### **Ventajas**

XAML tiene varias ventajas sobre el código imperativo equivalente.

- ✓ El código XAML es más compacto y legible que el código semejante.
- ✓ La jerarquía padre-hijo inherente de XML permite que XAML imite más visualmente la jerarquía padre-hijo de los objetos de la interfaz de usuario.
- ✓ Los programadores pueden escribir fácilmente código XAML a mano, pero también se puede crear utilizando herramientas visuales como Microsoft Blend (Muñoz, 2018).

## Desventajas

- ✓ XAML también tiene desventajas, la mayoría de las cuales están relacionadas con las limitaciones inherentes de los lenguajes de marcado:
- ✓ XAML no puede contener bucles para procesamiento iterativo. Sin embargo, algunos objetos de visualización de Xamarin.Forms pueden generar varios elementos, como el objeto ListView que puede crear varios elementos en función de los objetos de su colección ItemsSource.
- ✓ XAML en general no puede crear instancias de clases que no definen un constructor sin parámetros. Sin embargo, a veces existe un mecanismo para eludir esta restricción.
- ✓ XAML en general no puede llamar a métodos. Sin embargo, a veces existe un mecanismo para eludir esta restricción. (Muños 2018).

## Definición de android

Según Figueredo (2006), el sistema operativo móvil se considera como el programa principal y puede administrar todos sus recursos para que se utilicen de manera eficiente, cómoda y sin interrupciones, para que los usuarios puedan mantener la comunicación sin problemas. utilizando los recursos proporcionados por el hardware.(Malave, 2011)

Sistema operativo de Google y el número uno en popularidad, en 2003, Andy Rubin, Rich Miner, Nick Sears y Chris White fundaron Android Inc. En sus inicios, la empresa se centró en el "desarrollo de software móvil".

En agosto de 2007, Android Inc. fue contactado por Google, trayendo éxito a Android el 5 de noviembre de 2007. Ese día, se formó OHA (Open Handset Alliance), una sólida alianza comercial de 35 personas iniciada por Google. incluidos fabricantes de dispositivos móviles, empresas de telecomunicaciones y fabricantes de chips y desarrolladores de software.

Disponible con todas las versiones de Android existentes. Además, la posibilidad de que cada fabricante incluya su propia capa sobre la original significa que la experiencia del usuario no siempre es la que Google quiere. Cada vez más desarrolladores intentan diseñar sus aplicaciones para usuarios de Android.



### **Android está compuesto por:**

- ✓ Aplicaciones: Como mencionamos antes, todas están programadas en Java. Van desde los más básicos como Email , Calendario, SMS, hasta conseguir una app compleja y que se ejecute de manera correcta.
- ✓ La arquitectura de este sistema operativo está orientada a la reutilización de tantos componentes como sea posible. Una aplicación puede utilizar las capacidades de otra aplicación. Los desarrolladores tienen acceso total a la misma API de Workbench<sup>1</sup> utilizada por las aplicaciones principales.
- ✓ Bibliotecas: incluye varias bibliotecas C/C utilizadas por componentes diferentes en el sistema. Las bibliotecas están expuestas a los desarrolladores a través del marco de aplicación de Android<sup>5</sup>.
- ✓ Android Runtime: Android incluye varias bibliotecas principales que brindan la mayor parte de la funcionalidad disponible en las bibliotecas de Java. Cada aplicación de Android ejecuta su propio proceso.
- ✓ Linux Kernel: Android es un sistema que depende de Linux para servicios básicos sistema de seguridad, administración de memoria, procesos, etc. Linux kernel actúa como una abstracción

### **Lenguajes más populares para android studio**

- ✓ Kotlin: Este es el lenguaje más utilizado para la programación de Android. Su popularidad se debe a las ventajas que ofrece a los desarrolladores: código simple e intuitivo, sin comprometer la eficiencia. Básicamente está inspirado en Scala, pero mejora la velocidad de compilación y Java, entre otras cosas. En 2019, Android anunció ser Kotlin First. Desde entonces, Google recomienda desarrollar nuevos proyectos exclusivamente en Kotlin.
- ✓ JavaAndroid: está diseñado con Java en mente, por lo que ha sido durante mucho tiempo el lenguaje de programación predeterminado para desarrollar aplicaciones en este sistema operativo. El lenguaje es rápido, simple y ofrece muchas aplicaciones. Por ello, sigue siendo una de las apps más populares en la programación de Android.
- ✓ Javascript: Este lenguaje de programación multiplataforma es generalmente el más utilizado, según Github. En el contexto específico del desarrollo móvil, se

puede utilizar a la hora de crear aplicaciones híbridas, aunque destaca principalmente por optimizar la funcionalidad de los sitios web y las interfaces de usuario, haciéndolas más dinámicas e interactivas.

- ✓ C#: C# también es multiplataforma y, por lo tanto, una buena alternativa para crear aplicaciones híbridas. Orientado a objetos, destaca la influencia de Java en su sintaxis, así como su simplicidad y flexibilidad.
- ✓ PHP: Aunque este lenguaje se utiliza principalmente en el entorno web, también se puede utilizar para la programación de Android. Su atractivo se basa en la sencillez de su sintaxis y accesibilidad (es de código abierto) ( KeepCoding,2022)

### Versiones

Entre las versiones que podemos encontrar en la página web oficial [www.android.com](http://www.android.com) están desde las más antiguas a las más actuales, entre las menos utilizadas están las siguientes(Louis,2017):

- ✓ Android 1.0. Donut: Con innovación en navegación y multimedia.
- ✓ Android 2.0. Eclair: le permite organizar aplicaciones y widgets en varias pantallas y carpetas, así como fondos de pantalla en vivo.
- ✓ Android 2.2.Froyo: Input, Voice te permite ingresar texto y las acciones de voz te permiten controlar tu teléfono, todo esto con solo reconocimiento de voz.
- ✓ Android 3.0, Honeycomb: Optimizado para tabletas, este lanzamiento abre nuevos horizontes estés donde estés.
- ✓ Android 4.0, Ice Cream Sandwich: Android regresa con un nuevo diseño elegante. Sencillo, bonito y sobre todo inteligente.
- ✓ Android 5.0 Lollipop: los cambios más significativos de Lollipop incluyen una interfaz de usuario rediseñada basada en un diseño de lenguaje receptivo conocido como "Diseño de materiales", así como mejoras en el sistema de información para acceder a él desde la pantalla del escritorio. 'Bienvenidos. entre otros.
- ✓ Android 6.0 Marshmallow: Android Marshmallow brinda soporte nativo para el reconocimiento de huellas dactilares, lo que permite usar las huellas dactilares para desbloquear el dispositivo y autenticarse en Play Store o Android Pay.

- ✓ Android 7.0 Nougat: Presentado en el evento Google I/O de mayo de 2016, Nougat se puede resumir como una actualización de las mencionadas novedades de Marshmallow, la versión anterior de Android, entre otras, se pueden resumir sus principales características en los siguientes puntos: Multiventana, realidad virtual, Doze, Google Play.
- ✓ Android 8.0 O: el nombre en clave de la próxima versión de Android es Android "O" (O para Oreo, no Zero "0") y fue anunciado por Google el 21 de marzo de 2017 como la versión "Alpha". primero. lanzado para dispositivos Google Pixel y Nexus el 22 de marzo de 2017.

### **Características del sistema operativo android**

Como se mencionó anteriormente, Android es un sistema operativo para dispositivos móviles, como teléfonos inteligentes y tabletas, desarrollado originalmente por Google y luego por Open Handset Alliance y es un software basado en Linux (gratis, multiplataforma). y gratis).

El hecho de que Android sea gratuito significa que se publica bajo licencias diferentes, la más destacada es la licencia Apache, lo que permite a cualquier desarrollador crear aplicaciones y compilarlas en código ARM root (API de Android).

La estructura del sistema operativo Android se compone de aplicaciones escritas en el lenguaje de programación Java , pero no existe tal máquina virtual Java en la plataforma . En versiones anteriores, utilizaba una máquina virtual basada en Dalvik JIT (solo in time) compiló, ejecutó el código de bytes de Java y lo convirtió en código nativo de terminal en el propio tiempo de ejecución.(Angela,2018)

### **Ciclo de ejecución de procesos en Android**

- ✓ Procesos activos: son los procesos que contienen las aplicaciones que están interactuando con el usuario en ese preciso momento. Android liberará recursos para intentar que estos procesos activos siempre respondan sin retrasos. Los procesos activos solo se cancelarán como último recurso.
- ✓ Procesos visibles: Los procesos son visibles pero no activos, ya sea porque sus respectivas aplicaciones se muestran detrás de otras o porque no están respondiendo a ninguna entrada del usuario. Esto sucede cuando una actividad

está parcialmente oscurecida por otra, ya sea porque esta última es transparente o porque no ocupa toda la pantalla. Estos procesos solo se detuvieron en condiciones extremas.

- ✓ Procesos relacionados con la ejecución de servicios: Los servicios permiten el procesamiento sin una interfaz de usuario para mostrar. Dado que estos servicios no interactúan directamente con el usuario, tienen una prioridad ligeramente inferior a la de los procesos visibles. Sin embargo, los procesos aún se consideran activos y solo se cancelarán si es absolutamente necesario.
- ✓ Procesos Inactivos: estos son los procesos que alojan actividades que no son visibles ni manejadas y no están ejecutando ningún servicio. El orden en que se detendrán estos procesos está determinado por la cantidad de tiempo que han estado inactivos desde que se vio por última vez, de mayor a menor.
- ✓ Manejo vacío: este es el resultado de que Android intenta mantener las aplicaciones en la memoria después de que hayan terminado en el caché. Esto asegura que el lanzamiento de la aplicación llevará menos tiempo.

#### **Principales características que incluye Android Studio<sup>7</sup>:**

- ✓ Admite la programación de aplicaciones para Android Wear (sistema operativo para dispositivos portátiles como relojes).
- ✓ Herramientas Lint (detectar incompatibilidad entre diferentes arquitecturas o código ofuscado que el compilador no puede manejar) para detectar rendimiento, usabilidad y compatibilidad de versiones.
- ✓ Use ProGuard para optimizar y reducir el código del proyecto al exportar a APK (muy útil para dispositivos de gama baja con memoria interna limitada).
- ✓ Integración de la herramienta Gradle responsable de administrar y automatizar proyectos de compilación, como tareas de prueba, compilación o empaquetado.
- ✓ Nuevo diseño de editor con soporte para editar temas.
- ✓ Nueva interfaz específica para el desarrollo de Android.
- ✓ Permite la importación de proyectos realizados en entorno Eclipse, a diferencia de Android Studio (Gradle) que utiliza ANT.
- ✓ Habilite el control de versiones accediendo a un repositorio desde el cual puede descargar Mercurial, Git, Github o Subversion.

- ✓ Advertencias en tiempo real sobre errores de sintaxis, compatibilidad o rendimiento antes de la compilación de la aplicación.
- ✓ Vista previa en diferentes dispositivos y resoluciones.
- ✓ Integración con Google Cloud Platform, para acceder a diversos servicios que ofrece Google en la nube.
- ✓ El editor de diseño muestra una vista previa de los cambios realizados directamente en el archivo xml.

### Requerimientos del sistema

Windows: Microsoft Windows 8/7/Vista/2003 (32 o 64 bit).

Mac OS : Mac OS X 10.8.5 o superior, hasta la 10.9 (Mavericks).

Linux : GNOME o entorno de escritorio KDE.

- ✓ Mínimo de 2 GB de RAM, recomendado 4 GB de RAM.
- ✓ 400 MB de espacio en disco.
- ✓ Necesita de al menos 1 GB para Android SDK, emulador de imágenes del sistema, y cachés.
- ✓ Resolución mínima de pantalla de 1280 x 800.
- ✓ Java Development Kit (JDK) 7 o superior(Android,2018).

### Ventajas y desventajas

#### Ventajas

- ✓ Android Studio se ha convertido en el entorno recomendado para desarrollar aplicaciones de Android, ya que es el IDE oficial de Google en colaboración con JetBrains (una empresa de desarrollo de software que se especializa en la creación de IDE).
- ✓ Android Studio permite crear módulos nuevos en el mismo proyecto sin cambiar el espacio de trabajo para administrar proyectos, lo cual es común en Eclipse.
- ✓ Con una simple descarga de Android Studio, todas las herramientas necesarias para desarrollar aplicaciones para la plataforma Android están disponibles.
- ✓ Su nueva forma de construir paquetes .apk, mediante el uso de Gradle, ofrece una serie de ventajas más propias de un proyecto Java.



- ✓ Facilita la distribución de código y así trabajar en equipo. Reutilización de código y recursos.
- ✓ Permite la compilación desde la línea de comandos, para los casos en que no se disponga de un entorno de desarrollo.
- ✓ Es más fácil crear diferentes versiones de la misma aplicación, brindando muchos beneficios, como crear una versión paga y una versión gratuita, o por ejemplo, diferentes dispositivos o almacenamiento de datos.

### **Desventajas**

- ✓ Curva de aprendizaje más lenta para nuevos desarrolladores de Android.
- ✓ El sistema de construcción de proyectos Gradle puede resultar complicado inicialmente.
- ✓ En comparativa con Eclipse, menor número de plugins(Hernández,2017).
- ✓ Aunque se ha lanzado la primera versión estable, v1.0, que se encuentra en sus primeras etapas, todavía existe la posibilidad de introducir más cambios que pueden causar inestabilidad entre proyectos de otras versiones juntos.

### **Migración de android a multiplataforma**

Para migrar completamente las aplicaciones de Android al marco multiplataforma, se deben migrar la interfaz de usuario y la lógica. Puede ser una operación arriesgada a corto plazo, pero proporcionará la minimización necesaria de costos y recursos a mediano y largo plazo para que la empresa migre sus aplicaciones, ya que no solo tiene que programar las funciones y la interfaz de de una vez por todas, tanto los usuarios de Android como los de iOS podrán beneficiarse de estos cambios más rápidamente. 0 Antes de iniciar la migración, por ejemplo, es necesario evaluar si beneficiará a la aplicación 0 a migrar. Si se migra una aplicación de Android al marco multiplataforma dirigido solo a usuarios de Android , hacer esta migración a corto plazo es un riesgo evitable, ya que no exprimirá todo el potencial que este tipo de marco trae.(Millan,2019)

### **Migración de android a xamarin**

Una de las publicaciones más relevantes para hacer esto fue en StackOverflow, donde el usuario Lex Li respondió a un usuario que le preguntó qué podía hacer para migrar su aplicación de Android a iOS usando Xamarin. En la respuesta, Lex Li afirma que la interfaz de usuario de Android se puede reutilizar fácilmente porque la interfaz de usuario de los proyectos se crea con Xamarin. Android comparte el mismo formato XML con los proyectos nativos de Android, por lo que las máscaras existentes se pueden reutilizar. También se menciona que se pueden importar bibliotecas de terceros a Xamarin, aunque algunos han incluido directamente sus componentes de Xamarin. Finalmente, agregó que el código genérico Java se puede convertir a código C# usando la herramienta Sharpen. Reddit, escrito por el usuario "unndunn", responde a los usuarios que quieren migrar sus aplicaciones de Android a iOS usando Xamarin. En este anuncio, el usuario que desee migrar la aplicación debe aprender C# y .NET, pero para un programador familiarizado con Java, convertir a C# no será demasiado costoso. (Millan, 2019)

## **RESULTADOS Y DISCUSIÓN**

### **DISCUSIÓN**

En relación a los temas tratados en el presente artículo podemos notar que cada entorno de desarrollo móvil tiene una funcionalidad diferente, pero todas se enfocan en el mismo objetivo, es decir brindar una mejor experiencia al usuario al momento de utilizar el entorno de desarrollo que necesitan para el desarrollo de una app.

### **ILUSTRACIONES, TABLAS, FIGURAS.**

Se puede observar en la gráfica que durante los últimos años flutter está comenzando a tomar fuerza dentro del mundo del desarrollo móvil, además que Xamarin se puede observar que no tiene tanta acogida ya sea por ser un entorno nuevo, pero dentro la investigación realizada se pudo observar que Android Studio es uno de los entornos más utilizados en la actualidad ya que Android desde sus inicios se ha ido actualizando y los

usuarios de las aplicaciones realizadas en Android han tenido una mejor aceptación.

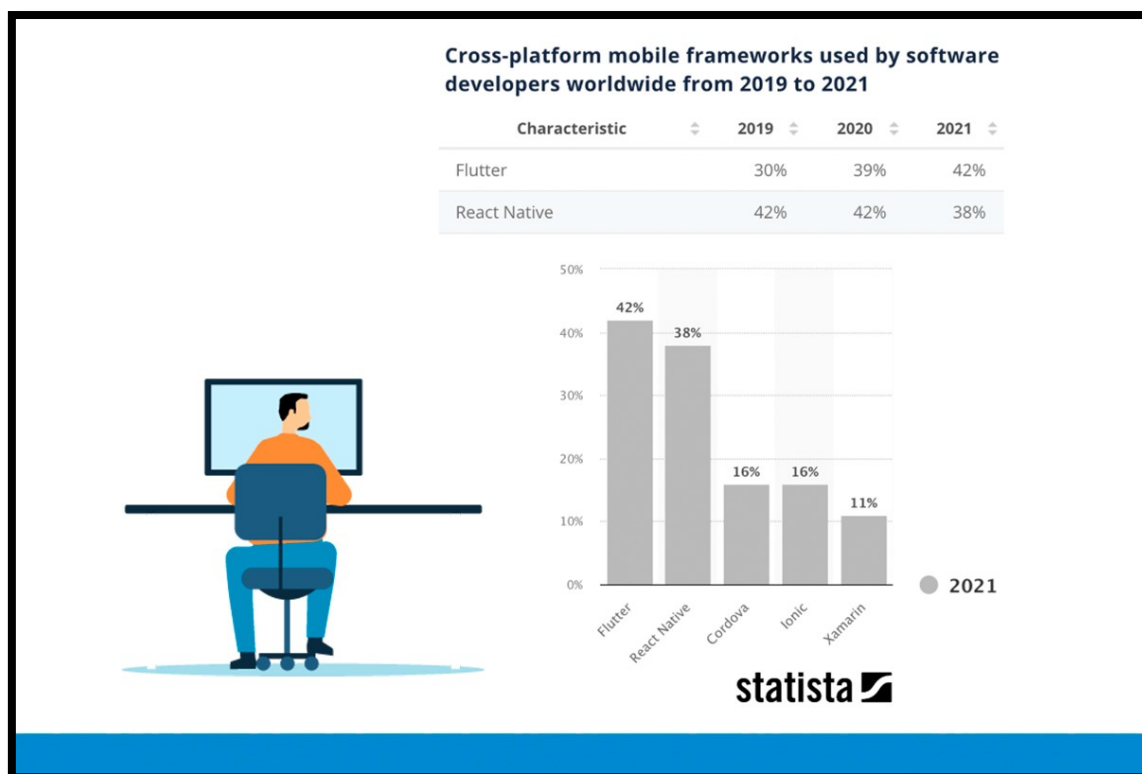


Ilustración 2 Tabla de estadísticas del uso de entornos de desarrollo móvil.

Tabla comparativa de los entornos de desarrollo móvil.

Entornos	Característica principal	Tipo de desarrollo
Xamarin	Xamarin permite compartir hasta el 90% de código entre plataformas	Semi-hibrido
Flutter	Crea aplicaciones para Android e iOS, en diseño móvil, web y escritorio, desde una única base de código	Nativo
Android Studio	Entorno unificado donde puedes desarrollar para todos los dispositivos Android	Nativo

Tabla 1 Tabla comparativa

## CONCLUSIONES

Concluimos que SQLite al ser una base relacional liviana y de alto rendimiento que se usa netamente para diseño de Software nos proporciona un nivel extra a la hora de desarrollar aplicaciones para dispositivos móviles ya que no todas las aplicaciones requieren una gran potencia de base de datos.

Se debe tener en cuenta los factores que se deben analizar a la hora de de programar una Apps Nativa, ya que la misma no permite reutilizar el código fuente para reutilizarlo, sin embargo, hay lenguajes de programación que permiten migrar una aplicacion android a flutter o xamari, ademas debemos considerar el tiempo que nos llevará construir una aplicación nativa ya que al ser nativa de android se logrará acceder en su totalidad al sistema operativo de cada plataforma.

## LISTA DE REFERENCIAS

- Android Studio 文档教程* - *Android Studio 安卓开发者社区*. (n.d.). <http://ask.android-studio.org/?/explore/category-studio>
- APOLINARIO ROJAS CARLOS ALBERTO *Trabajo de Titulación Previo a la obtención del Título de*. (2020).
- Fernandez, C. (2018). *Programación de aplicaciones Android para aprendizaje de Idiomas*. 23–30.
- Hernández, L., & Lozada, M. (2017). *Aplicación Para Dispositivos Móviles Con Sistema Operativo Android Para El Aprendizaje De Conceptos Básicos De Lógica De Programación Estudiantes: Louis Steven Hernández Sánchez-20131078022 Martín Leonardo Lozada Cortés-20102078110*.
- Icaza Guamba, C. A. (2021). Análisis, diseño y desarrollo de un sistema inteligente y automatizado de monitoreo y control de cultivos con IoT. *Universidad Central Del Ecuador*, 258.
- KeepCoding. (2022). *Los 5 lenguajes más usados para la programación en Android*. <https://keepcoding.io/blog/programacion-en-android/>
- Mainez, P., & Ángela. (2018). *Diseño y desarrollo de una aplicación Android para el envío de formularios mediante la tecnología Wifi Direct*.
- Millán, C. Z. (2019). Migración de aplicaciones Android hacia Flutter , un framework para desarrollo de apps multiplataforma. *Universitat Politècnica de València*. <https://riunet.upv.es/handle/10251/128486>
- Muñoz, M., & Antón, M. (2018). *Desarrollo de una aplicación móvil mediante Xamarin para el apoyo a terapeutas en la rehabilitación cognitiva de pacientes*.



- Pilalumbo Tulpa, J. I. (2019). Universidad Politécnica Salesiana Sede Quito. *Tesis*, 80.
- Wu, W. (2018). *Flutter Vs React. March*.
- Hernández, R., Fernández, C., & Baptista, P. (2014). Metodología de la investigación. Chile: Editorial McGraw-Hill.
- Fundación, M., Venezuela, J., Polanco, M., Taibo, B., Luis, J., Maracaibo, J., Polanco, K. M., & Beauperthuy, J. L. (s/f). 79 / 96 "ANDROID" GOOGLE'S OPERATING SYSTEM FOR MOBILE DEVICES "ANDROID" EL SISTEMA OPERATIVO DE GOOGLE PARA DISPOSITIVOS MÓVILES. Redalyc.org. Recuperado el 24 de agosto de 2022, de <https://www.redalyc.org/pdf/782/78219156004.pdf>
- Thomas, P., Galdamez, N., Lisandro, D., Federico, C., Dapoto, S., & Pesado, P. (n.d.). *Dispositivos Móviles: Desarrollo de Aplicaciones y Conectividad*. Core.ac.uk. Retrieved August 22, 2022, from <https://core.ac.uk/download/pdf/296374627.pdf>
- Mohammed, M. (2019). Xamarin: Xamarin for beginners, building your first mobile app with C# .net and xamarin. Independently Published.
- Del Valle Hernández, L. (2016, junio 22). ¿Qué es WPF? un repaso por sus características. Programar fácil con Arduino. <https://programarfacil.com/blog/programacion-net-blog/que-es-wpf/>
- Lopez, P (2021). Desarrollo de una aplicación móvil para gestionar colecciones.
- Univision, P. (2014, abril 18). Conociendo Xamarin, la herramienta para desarrolladores multiplataforma. Univision. <https://www.univision.com/explora/conociendo-xamarin-la-herramienta-para-desarrolladores-multiplataforma>
- De, C., En, I., & Computacionales, S. (s/f). UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS. Edu.ec. Recuperado el 24 de agosto de 2022, de <http://repositorio.utn.edu.ec/bitstream/123456789/5611/3/ARTICULO.pdf>
- Gonzalez, A. N. (2011, febrero 8). ¿Qué es Android? Xatakandroid.com; Xataka Android. <https://www.xatakandroid.com/sistema-operativo/que-es-android>