

Creación de aplicaciones multiplataforma en móviles

Índice

[3. Introducción](#)

[4. Marco Teórico](#)

[5. Descripción de Herramientas para la Creación de Aplicaciones Multimedia](#)

[6. Comparativa de Herramientas \(Tabla\)](#)

[7. Gráficos y Elementos Visuales](#)

- [Resumen de Hallazgos](#)
- [Reflexión. ¿Cuál Herramienta es Más Conveniente?](#)
- [Comentarios Finales](#)

[9. Bibliografía](#)

[10. Formato y pautas de entrega](#)

POR GRUPO A

Manuel Azzalin DNI 35669618E

Rubén Agra Casal DNI 53797243J

Andrés Amado Cibreiro DNI 46290349C

1. Introducción

Las aplicaciones multimedia son plataformas en línea que combinan diferentes formatos de multimedia, como texto, imágenes, sonido y video, para proporcionar una experiencia de usuario interactiva y atractiva. Estas aplicaciones se utilizan comúnmente en el mundo educativo, la rehabilitación y el entretenimiento.

En este trabajo, para investigar y comparar 2 aplicaciones de creación de apps multimedia en múltiples campos, como si son gratuitas o de pago, las plataformas en las que están disponibles, etc...

Esto es relevante por que así podemos aprender que programas no son más útiles a la hora de crear la aplicación que deseamos y en que plataformas son mejores para esta a la hora de su expansión.

2. Marco Teórico

Plataformas de desarrollo:

Android: Está basado en el núcleo Linux y otros software de código abierto. Está diseñado para dispositivos móviles con pantalla táctil, lanzado inicialmente en 2008, y que para finales de 2020, abarcaba un 84,1% del mercado de la industria móvil.

IOS: sistema operativo móvil de código cerrado desarrollado por Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se utilizó en dispositivos como el iPod touch y el iPad. Para finales de 2020, abarcaba un 15,9% del mercado de la industria móvil.

Windows Mobile: Se basó en el núcleo del sistema operativo Windows CE y contó con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows. Estaba diseñado para ser similar a las versiones de escritorio de Windows estéticamente. Si bien ahora ya no está activo, tuvo aplicaciones diseñadas para este las cuales también se tomarán en cuenta.

Smart TV: también conocido como TV conectada (CTV), es un televisor tradicional con Internet integrado y funciones interactivas de la Web 2.0 que permiten a los usuarios transmitir música y vídeos, navegar por Internet y ver fotos. Además de las funciones tradicionales de los televisores proporcionados a través de los medios de transmisión tradicionales, estos dispositivos pueden proporcionar acceso a servicios de medios de transmisión como transmisión de televisión y radio por Internet, junto con acceso a redes domésticas. Esto han ido evolucionando a ser la principal forma de televisión vendida en los 2010s, y sigue aumentando, lo que implica una gran demanda de mercado.

Tipos de herramientas:

Por acceso:

- **Gratuito:** El acceso a la herramienta es gratis, sin embargo puede tener funciones bloqueadas por pagos externos.
- **Pago:** Se tiene que pagar la Licencia de uso; existen diferentes tipos de licencias dependiendo de la escala o uso, y hay algunos casos en los cuales las licencias son gratuitas, como por ejemplo un colegio autorizado permite que la licencia de Microsoft office sea gratuita para los estudiantes y profesores..

Por plataforma:

- **Nativo:** Las aplicaciones nativas son programas diseñados y **desarrollados específicamente para un sistema operativo particular**, como iOS o Android. A diferencia de las aplicaciones web o las híbridas, las nativas aprovechan al máximo las capacidades y características únicas de cada plataforma. Por eso, el desarrollo de aplicaciones multiplataforma es cada vez más demandado.
- **Multiplataforma:** Estas aplicaciones son diseñadas para funcionar en diferentes sistemas operativos, como iOS, Android, Windows y más. La idea principal detrás de estas aplicaciones es brindar una **experiencia consistente a los usuarios, sin importar el dispositivo que utilicen**. Utilizando frameworks o herramientas específicas que permiten escribir el código una vez y luego compilarlo para funcionar en diferentes plataformas. Estas herramientas facilitan la tarea de los desarrolladores al proporcionar un conjunto de bibliotecas y funciones comunes que pueden ser utilizadas en múltiples sistemas operativos.

Lenguajes de programación:

Java: Java es un lenguaje de programación y plataforma informática lanzado por primera vez por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, proporcionando la plataforma fiable sobre la que se construyen muchos servicios y aplicaciones.

Kotlin: Kotlin es un lenguaje de programación multiplataforma, estáticamente tipado, de alto nivel y propósito general con inferencia de tipos. Creado en el 2016, Kotlin está diseñado para ser totalmente interoperable con Java, y la versión JVM de la biblioteca estándar de Kotlin depende de la Java Class Library, pero la inferencia de tipos permite que su sintaxis sea más concisa.

C: El lenguaje C es un lenguaje de programación de propósito general originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

Está orientado a sistemas operativos, lenguaje de tipos de datos estáticos, débilmente tipado, y de nivel medio (estructuras típicas de los lenguajes de alto nivel, pero, a su vez, dispone de construcciones del lenguaje que permiten un control a bajo nivel).

C++: es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C y añadir mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido.

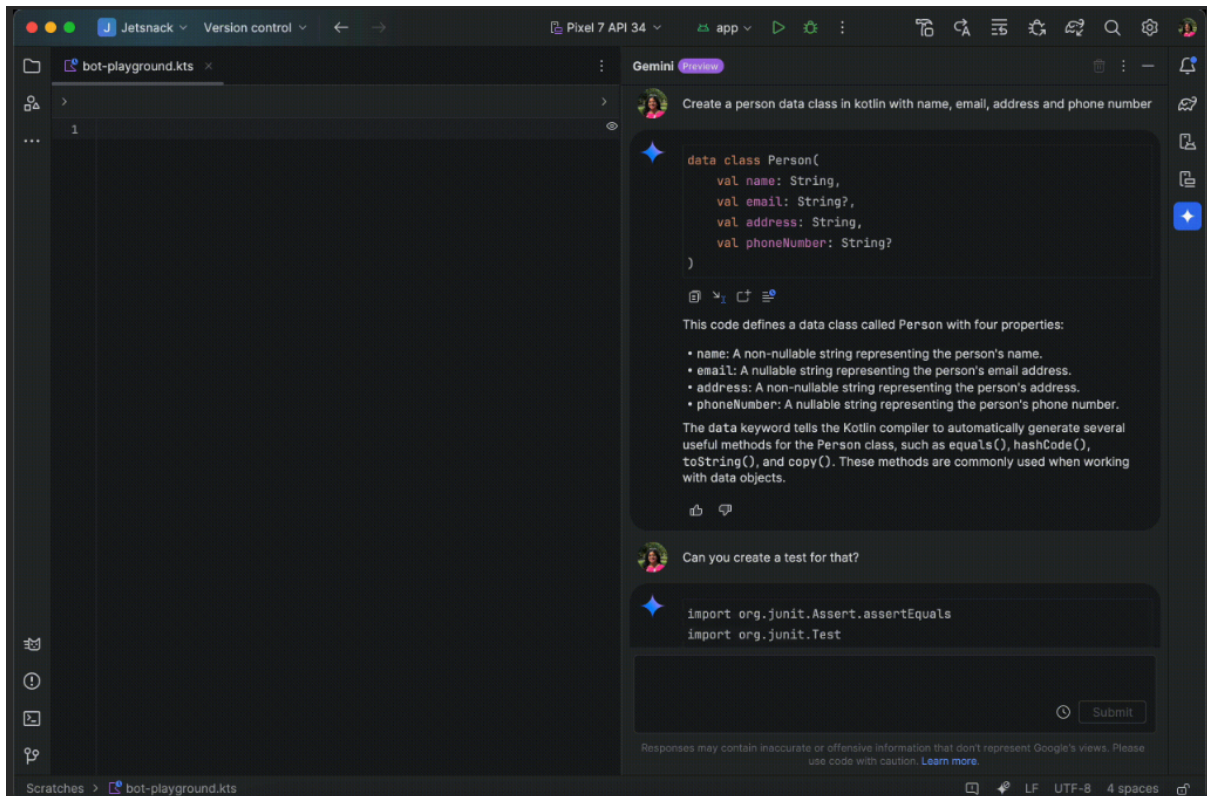
Dart: Creado en el 2011, El lenguaje Dart es de tipo seguro; Utiliza la comprobación de tipos estáticos para garantizar que el valor de una variable siempre coincida con el tipo estático de la variable. Aunque los tipos son obligatorios, las anotaciones de tipo son opcionales debido a la inferencia de tipos. El sistema de tipificación Dart también es flexible, lo que permite el uso de un tipo dinámico combinado con comprobaciones en tiempo de ejecución, lo que puede ser útil durante la experimentación o para el código que necesita ser especialmente dinámico.

3. Descripción de Herramientas para la Creación de Aplicaciones Multimedia

Android Studio

Android Studio es un entorno de desarrollo integrado (IDE) **oficial** y gratuito para Android.

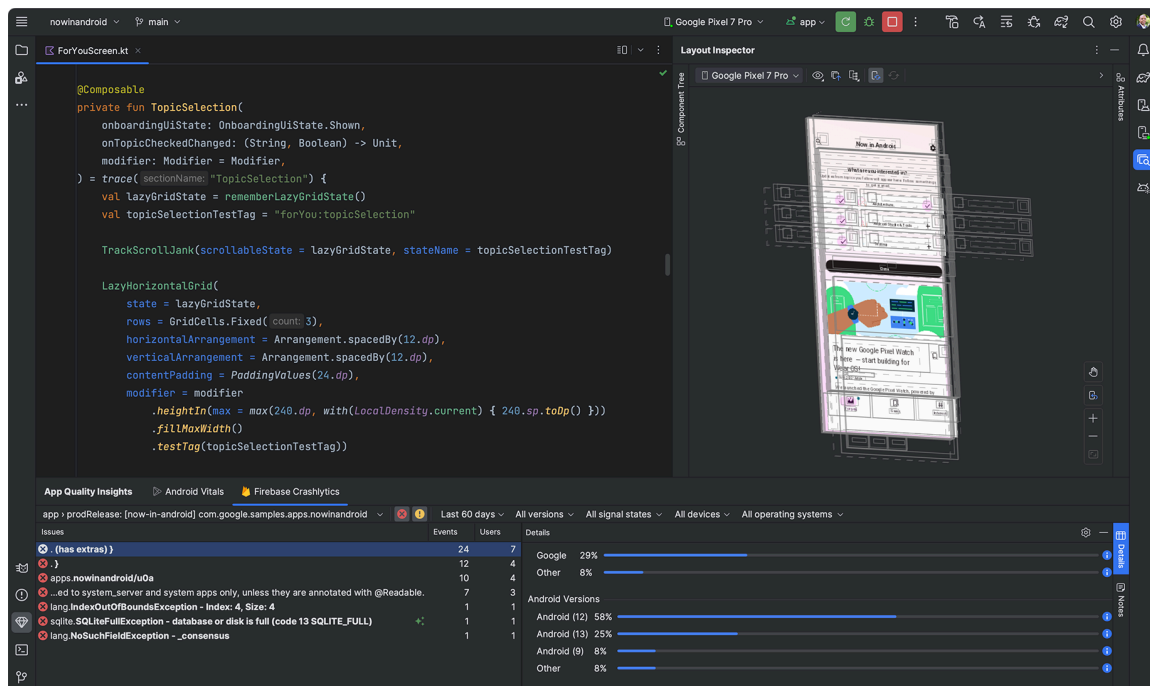
Al ser oficial de Android, este incorpora diversas integraciones con varios servicios de Google como Cloud Platform, Google App Engine, Google Maps o la nueva IA Gemini. Este entorno está basado en el software de IntelliJ de JetBrains por lo que admite los mismos lenguajes de programación como Java, C++ o Go utilizando extensiones, aunque el lenguaje de programación más usado es Kotlin.



Gemini integrado en Android Studio

Además de esto, incorpora diversas características para su desarrollo como un editor de código inteligente, herramientas Lint para detectar problemas de rendimiento, una refactorización específica para Android, etc.

En cuanto a herramientas orientadas a diseño, Android Studio aporta diversas plantillas para crear diseños comunes de Android y un editor bastante intuitivo que permite arrastrar y soltar componentes que se podrá comprobar en tiempo real utilizando el dispositivo virtual que trae el propio entorno.

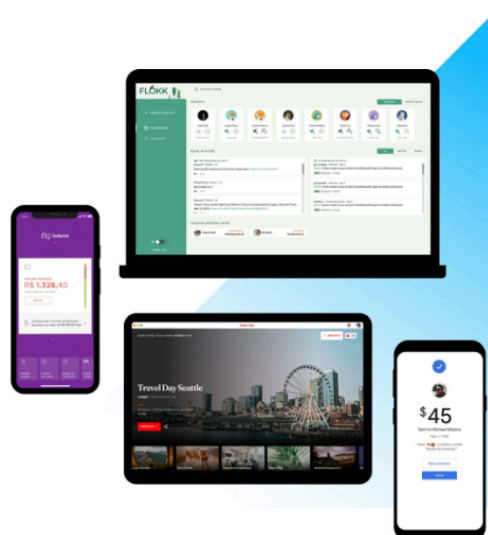


Interfaz de Android Studio

Android Studio está disponible para Windows 2003, Vista, 7, 8, y 10, tanto plataformas de 32 como de 64 bits, GNU/Linux y macOS, desde 10.8.5 en adelante.

Flutter (Herramienta gratuita)

Flutter es un kit de desarrollo de software de código abierto creado por Google para desarrollar aplicaciones multiplataforma desde una sola base de código para web, Android, iOS, Fuchsia, Linux, macOS y Windows. Antes Flutter era solo compatible para el desarrollo de aplicaciones móviles. Google usa este kit para aplicaciones suyas como Google Pay o Google Earth.



Una de las peculiaridades que tiene Flutter, es que está desarrollado con Dart, un lenguaje de programación también desarrollado por Google que se compila en código máquina, por lo que el rendimiento será bastante rápido y eficaz.

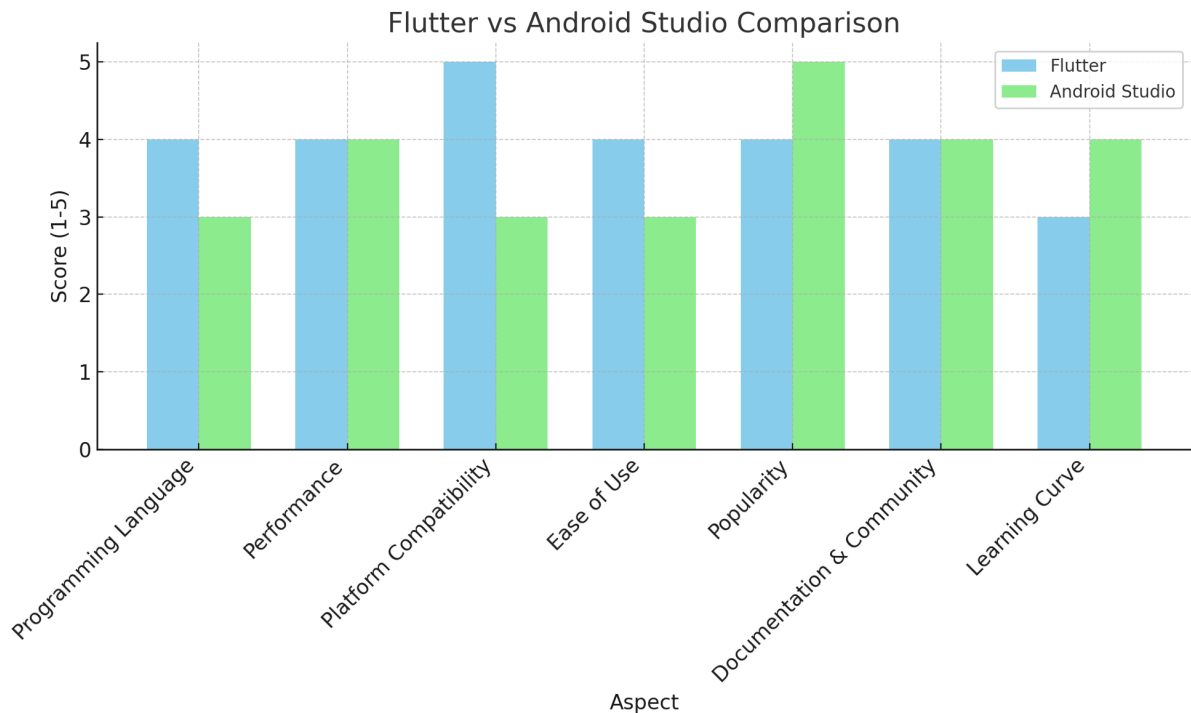
Además, Flutter utiliza una biblioteca gráfica de código abierto llamada Skia para renderizar la interfaz de usuario y proporciona a los desarrolladores diversas herramientas para que sea fácil de usar, como la recarga en caliente o como el inspector de widgets.

4. Comparativa de Herramientas (Tabla)

Nombre	Gratuita/D e pago	Plataformas Soportadas	Facilidad de uso	Capacidades multimedia	Plugins / Extensi ones	Coste (si aplica)	Lenguaje de programación
Android Studio	Gratuita	Android, Windows, macOs y Linux	Alta	Imágenes, audio, vídeo, gráficos 3D y 2D	Sí		Kotlin, Java, C++ y C
Flutter	Gratuita	iOS, Android, Windows, MacOS y Linux	Alta	Audio, video, imágenes y gráficos 2D	Sí		Dart

Esta tabla permite comparar de forma rápida y visual las características más importantes.

5. Gráficos y Elementos Visuales



En esta gráfica podemos ver distintas skills de ambas herramientas. Versatilidad en cuando a lenguajes de programación, rendimiento, compatibilidad entre plataformas, facilidad de uso, popularidad, comunidad activa y curva de aprendizaje.

● Diagrama de Flujo sobre Android Studio

Inicio



Crear nuevo proyecto en Android Studio

Selecciona "Nuevo Proyecto".

Configura el nombre, la ubicación y el tipo de proyecto (actividad vacía o plantilla básica). ↓

Configurar dependencias y permisos en AndroidManifest.xml

Agrega las dependencias necesarias (por ejemplo, para reproducción multimedia o acceso a la cámara).

Establece permisos (acceso a archivos, internet, micrófono, etc.). ↓

Diseñar la interfaz de usuario (UI)

Utiliza el Editor de Layout para crear botones, sliders o vistas donde se mostrarán los archivos multimedia (audio, video, imágenes).

Personaliza los elementos visuales según la funcionalidad que se desea. ↓

Agregar archivos multimedia a la aplicación

Inserta imágenes, videos o audios en las carpetas del proyecto (como res/raw o assets).

Asegúrate de que los archivos estén organizados correctamente. ↓

Configurar la reproducción multimedia

Utilizá las clases de Android como MediaPlayer o ExoPlayer para manejar la reproducción de audio y video.

Configura los controles (play, pause, stop) y las interfaces asociadas. ↓

Probar la aplicación

Ejecuta la app en un emulador o dispositivo físico para asegurarte de que los archivos multimedia se reproduzcan correctamente.

Depura errores si es necesario. ↓

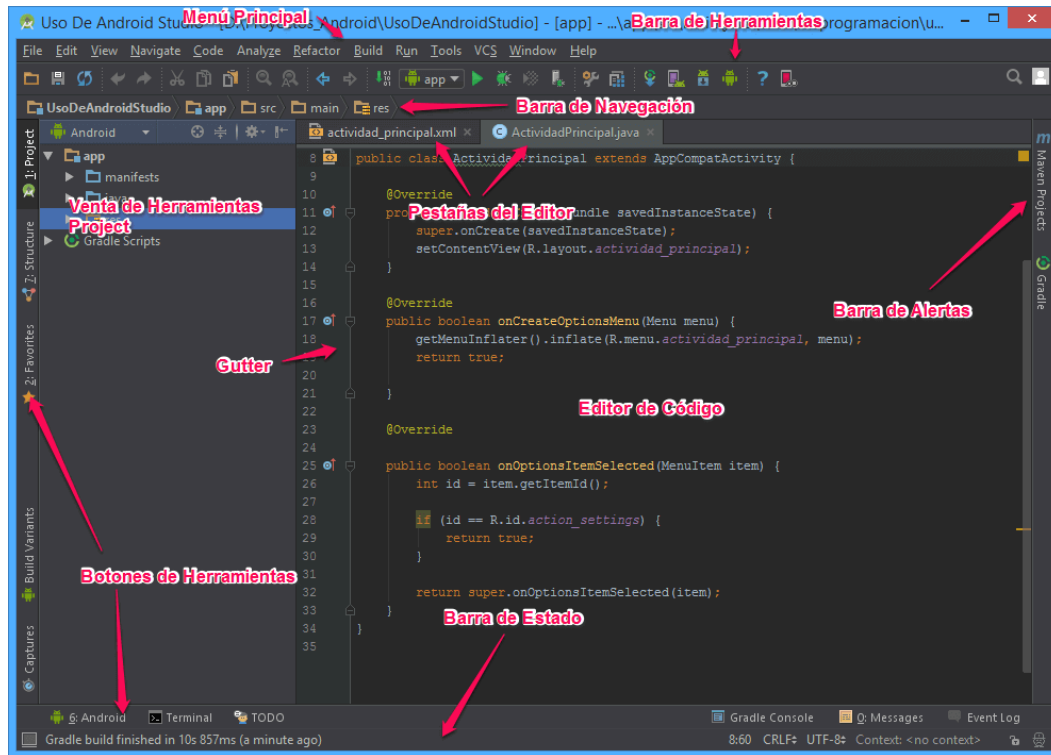
Compilar y distribuir la aplicación

Compila el APK o crea un bundle de aplicación.

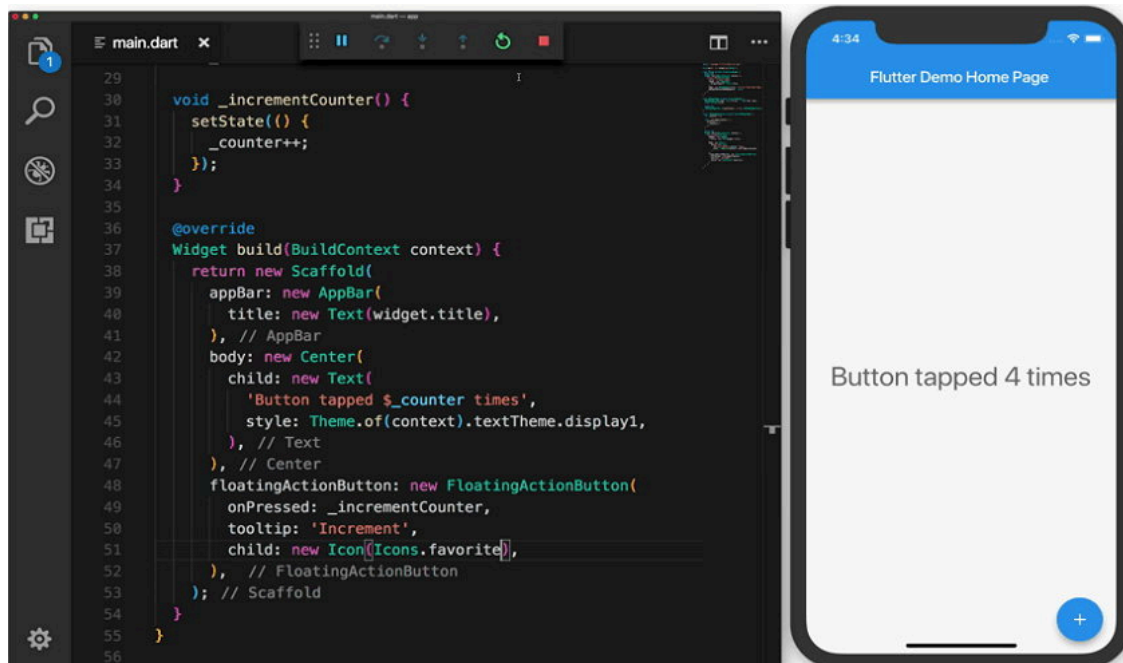
Firma digitalmente la app para su distribución en Google Play u otras plataformas. ↓

- Capturas de pantalla de las interfaces de las herramientas

Android Studio



Flutter



8. Conclusiones

En este apartado, se realiza un resumen de los hallazgos más importantes en el análisis de las herramientas de desarrollo de aplicaciones multimedia, Android Studio y Flutter. Ambas plataformas tienen características únicas que pueden ser ventajosas según las necesidades del proyecto.

- **Resumen de Hallazgos**

Android Studio, como entorno oficial de desarrollo para Android, proporciona una integración completa con los servicios de Google y un control profundo sobre el sistema Android. Sus características avanzadas, como el editor de código inteligente, la simulación en tiempo real, y las herramientas de diseño, lo convierten en una opción robusta para desarrollos multimedia en dispositivos Android. Sin embargo, está limitado en términos de soporte multiplataforma, ya que requiere herramientas adicionales para desarrollar en iOS o Windows, lo que puede aumentar la complejidad de los proyectos multiplataforma.

Flutter, por otro lado, se destaca como una herramienta gratuita y de código abierto para el desarrollo multiplataforma. Permite crear aplicaciones para iOS, Android, web, y sistemas de escritorio (Windows, macOS y Linux) con una única base de código, lo que puede reducir significativamente el tiempo de desarrollo y los costos. Su uso de Dart y la herramienta de recarga en caliente ofrecen un flujo de trabajo eficiente para desarrolladores, aunque puede tener una curva de aprendizaje inicial para quienes no estén familiarizados con el lenguaje.

- **Reflexión. ¿Cuál Herramienta es Más Conveniente?**

La elección entre Android Studio y Flutter depende de varios factores:

Presupuesto: Ambas herramientas son gratuitas, por lo que la decisión no depende del coste de la herramienta en sí. Sin embargo, Flutter podría reducir los costos si el objetivo es desarrollar una aplicación multiplataforma, ya que permite escribir el código una sola vez y desplegarlo en múltiples plataformas.

Complejidad del proyecto: Si el proyecto implica el uso intensivo de funcionalidades nativas de Android, como interacciones profundas con hardware o servicios de Google, Android

Studio sería más adecuado. Sin embargo, si el objetivo es lanzar una aplicación en múltiples plataformas con funcionalidades comunes (audio, video, gráficos), Flutter puede ser más eficiente.

Facilidad de uso y curva de aprendizaje: Android Studio es una herramienta madura con una comunidad activa y muchos recursos de aprendizaje. Flutter, a pesar de ser más reciente, tiene una comunidad creciente y bien organizada, pero puede resultar más desafiante para desarrolladores nuevos en Dart.

● Comentarios Finales

La importancia de contar con la herramienta adecuada para el desarrollo de aplicaciones multimedia es crucial, ya que influye directamente en el tiempo de desarrollo, los costos, la calidad de la aplicación y la capacidad de esta para llegar a múltiples plataformas. Android Studio es la elección ideal para desarrolladores que buscan aprovechar al máximo el ecosistema de Android. Mientras tanto, Flutter proporciona una solución potente y flexible para aquellos que desean desarrollar aplicaciones que puedan ser ejecutadas en diferentes sistemas operativos con un esfuerzo mínimo de adaptación.

Ambas herramientas tienen sus fortalezas, y la selección correcta dependerá del contexto específico del proyecto y de las prioridades del equipo de desarrollo.

6. Bibliografía

- Mia , M (2024,). 10 Ejemplos de Aplicaciones multimedia online: Definición, Que es, Diferencias, Significado y Usos. Ejemplos verdes. <https://ejemplosverdes.com/aplicaciones-multimedia-online-definicion-que-es-diferencias-significado-usos/>
-
- Monica, M(2021, 31 agosto). Android e iOS dominan el mercado de los smartphones. Statista <https://es.statista.com/grafico/18920/cuota-de-mercado-mundial-de-smartphones-por-sistema-operativo/>

- Windows Mobile. Wikipedia https://es.wikipedia.org/wiki/Windows_Mobile
- Smart TV. Wikipedia https://en.wikipedia.org/wiki/Smart_TV
- Lenguajes de programación. Wikipedia https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n
- App nativas. Universidad Europea <https://universidadeuropea.com/blog/apps-nativas/>
- Brendon V. (2024, 17 diciembre). Cómo funcionan las aplicaciones multiplataforma: Una guía completa. MBlog.

<https://www.mediummultimedia.com/apps/como-funcionan-las-aplicaciones-multiplatforma/>

- Kotlin. Wikipedia [https://es.wikipedia.org/wiki/Kotlin_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Kotlin_(lenguaje_de_programaci%C3%B3n))
- What is Java? Java https://www.java.com/en/download/help/whatis_java.html
- C (lenguaje de programación). Wikipedia [https://es.wikipedia.org/wiki/C_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n))
- C++. Wikipedia <https://es.wikipedia.org/wiki/C%2B%2B>
- Dart Overview. Dart <https://dart.dev/overview>
- AndroidDev(2024, 14 Junio). Cómo descargar Android Studio? <https://developer.android.com/studio?hl=es-419>
- Android Studio https://es.wikipedia.org/wiki/Android_Studio
- Ricardo Gottheil (2022, 11 Mayo). Maximizando Flutter: La magia de las extensiones y cómo implementarlas <https://www.ricardogottheil.com/maximizando-flutter-la-magia-de-las-extensions-y-como-implementarlas/#:~:text=Una%20de%20las%20caracter%C3%ADsticas%20que,uso%20de%20estas%20extensionsen%20Flutter.>
- Descripción general de MediaPlayer <https://developer.android.com/media/platform/mediaplayer?hl=es-419>
- Página oficial de Flutter. <https://flutter.dev/>
- Qué es Flutter? - Explicación de la Aplicación Flutter - AWS <https://aws.amazon.com/es/what-is/flutter/>

10. Formato y pautas de entrega

Pautas y Formato de Entrega del Proyecto

1. Formato del Documento

Formato de archivo: El proyecto debe entregarse en **PDF**.

Nombre del archivo: El archivo debe nombrarse de la siguiente manera: Grupo_Nºmiembros_Título_corto_del_trabajo.pdf. Ejemplo: Grupo-A_5_Tools4Apps.pdf.

2. Tamaño y tipo de letra

Tamaño de letra:

Texto principal: 12 puntos. LISTO

Títulos y subtítulos: Tamaño 14 o 16 puntos, negrita. LISTO

Espaciado:

Texto principal: Interlineado de 1.5.
LISTO

Títulos y subtítulos: Espacio antes y después de 12 puntos para diferenciarlos del texto.

Alineación del texto: Justificado para el cuerpo del texto, y alineación centrada para títulos principales.

3. Paginación

Todas las páginas deben estar numeradas en el **pie de página**, comenzando en la **Introducción** (página 1).

Portada e **índice** no deben llevar número de página, pero sí deben contarse.

4. Estructura del Documento

a) Portada

Título corto.

Título del trabajo (centrado, en negrita, tamaño de letra 18 puntos).

Nombre del grupo.

Nombre y primer apellido de cada integrante.

Nota: Tanto grupo como integrantes alineados parte derecha fondo portada.

b) Índice

Listado de secciones y subsecciones con el número de página correspondiente.

Debe tener una estructura clara y reflejar el contenido de cada apartado.

c) Introducción

Un párrafo que describa el tema del trabajo y los objetivos generales.

Explicación breve del propósito del trabajo y la importancia de las herramientas para la creación de aplicaciones multimedia.

d) Cuerpo del Trabajo

Subtítulos claros para organizar la información.

Cada apartado debe comenzar en una nueva página (salto de página).

Descripciones de las herramientas, plataformas y características, con explicaciones claras y concisas.

Incluir gráficos, tablas o imágenes que aporten al contenido. Todos los gráficos deben tener títulos y numeración. Ejemplo: “Gráfico 1. Comparación de herramientas”.

e) Tabla Comparativa

Una tabla clara y organizada, que cumpla con los criterios establecidos (nombre de la herramienta, coste, plataformas, características, etc.).

La tabla debe estar alineada al centro y tener un título, por ejemplo: “Tabla 1. Comparativa de herramientas para creación de apps multimedia”.

5. Entrega digital

El documento debe subirse a la plataforma virtual en el formato **PDF** antes de la fecha límite.

El archivo debe estar libre de errores de ortografía y gramática. Es recomendable revisarlo detenidamente antes de la entrega.