

Reto 1: Comparativa entre Android e IOS

1. Características (Herramientas, lenguaje...) de Android y de IOS

1.1. Android.

Android es un SO muy utilizado en el día de hoy. Presenta una cuota de mercado del 70%, cifras que logra gracias a su flexibilidad, lo que la dota de capacidad de funcionar con alta compatibilidad en distintos dispositivos y con elevadas capacidades de personalización gracias a su condición de código abierto

- **Herramientas usadas en Android:** La más usada sin duda es AndroidStudio
- **Lenguaje: JAVA** (también kotlin)
 - Lenguaje robusto y con larga trayectoria (lo que la dota de una amplia comunidad)
 - Alto rendimiento y eficacia
 - Permite crear Apps multiplataforma con las IDE ReactNative o Xamorin)
 - Lenguaje Orientado a Objetos: Facilita la organización y la reutilización de código.
 - Mejora la estabilidad y la mantenibilidad de proyectos a gran escala
- **Tienda de Aplicaciones: Google Play**, aunque los dispositivos android permiten la instalación de .apk no oficiales ni registradas en la propia Store.
 - En la Google Play hay que pagar un coste de 25\$ para el registro como desarrollador. Exige un 15-30% según los ingresos como comisión, sin embargo la publicación es más rápida, siendo automatizada (horas) al igual que en el caso de la publicación de actualizaciones.
 - Los controles de calidad son menos rigurosos y tienen políticas de privacidad más permisivas, con lo que hay más riesgo de malware.
- **Directrices de diseño: Material Design para Android.**
 - Sistema de diseño que se basa en color, tipografía, elevación, movimiento y accesibilidad. Las animaciones suelen ser microinteracciones que aportan feedback a las acciones del usuario
 - **Componentes principales:** App bar, FAB, tarjetas, bottom navigation, listas, campos de texto.

1.2. IOS

- **Herramientas usadas en IOS:** La más usada sin duda es Xcode
- **Lenguaje: Swift** (también Objective C, aunque es más rudimentario)
 - Lenguaje de código cerrado que aporta gran seguridad y calidad al entorno, disminuyendo las posibilidades de personalización.
 - Actualizado y más moderno que java. Presenta incluso **seguridad integrada** y una sintaxis más clara.
 - Dirigida a equipos de alta gama (Entorno Apple)
 - Funcionamiento cohesionado con el entorno Apple en sus diversas plataformas garantizando un rendimiento nativo en dispositivos IOS.
- **Tienda de Aplicaciones: App Store.** Los dispositivos con el IOS tienen unas restricciones muy marcadas a la hora de instalar aplicaciones de terceros al no usar el modelo .apk de android.
 - En la App Store hay que pagar un coste de 99\$/año para el registro como desarrollador. Exige un 15-30% según los ingresos como comisión y los métodos de pago son mucho más restringidos (sólo dispositivos apple)
 - Los controles de calidad son más rigurosos y manuales (pudiendo tardar días en publicar la app o las actualizaciones, que también se revisan manualmente).
 - Su entorno tan controlado reduce enormemente la cantidad de malware disponible desde su tienda, además presenta unas políticas de privacidad muy detalladas y exigentes
- **Directrices de diseño: Human Interface Guidelines.**
 - Sistema de diseño de Apple que se basa en minimalismo y claridad, con jerarquía visual marcada y respuesta táctil.
 - **Componentes principales:** Barra de navegación y pestañas, tablas, colecciones, botones etc... Siempre tratando de usar componentes nativos para aportar mayor coherencia y accesibilidad
 - El Theming y el Estilo se ven remarcados por la tipografía "SanFrancisco", una paleta de colores limitada y un cuidadoso uso del color para marcar diversos puntos de énfasis.

2. Ventajas y Desventajas de Android e IOS

Plataforma	Ventajas	Inconvenientes
Android	<ul style="list-style-type: none"> - Flexibilidad - Personalización - Publicación de app y actualizaciones mas rápida (Revisión automatizada) - Lenguaje Java (Extrapolable a muchos otros tipos de sistema) - Facilidad en la publicación 	<ul style="list-style-type: none"> - Menor seguridad - Menor cohesión entre los apartados estéticos - Directrices de diseño más laxas (+ Creatividad, - Cohesión) - Mucha competencia, dificultad de destacar. - Mayor riesgo de Malware (Revisiones automatizadas)
IOS	<ul style="list-style-type: none"> - Su entorno cerrado facilita una mejor seguridad - Mayor calidad por revisiones manuales - Directrices de diseño estrictas (+ Cohesión) - Funcionamiento nativo en dispositivos IOS - Lenguaje SWIFT (más moderno y compacto) 	<ul style="list-style-type: none"> - Lenguaje SWIFT: No permite uso fuera de IOS. - Las publicaciones y actualizaciones se demoran más por la revisión manual - Menor flexibilidad de dispositivos admitidos (sólo Apple) - APP Store más cara, que puede suponer una barrera de acceso - Menor personalización y control sobre el sistema y las apps (Entorno cerrado)

3. Tabla comparativa

	Android	IOS
Herramienta	Android Studio	XCode
Lenguaje	JAVA (también kotlin)	Swift
Diseño	Menos estricto, permitiendo más variabilidad en el diseño, lo que aporta un punto extra a la creatividad a costa de que algunas apps pierdan intuitividad o accesibilidad en función del diseño	Muy estricto y de carácter minimalista (menos es más). Aporta un punto extra a la cohesión de las App, haciendo que su intuitividad se traspase entre la mayoría de Apps del entorno a costa de reducir la libertad artística de los Devs.
Otros Factores	<ul style="list-style-type: none"> - Acceso más económico a ser Dev en la PlayStore (25\$ en pago unico) - Libertad creativa - Entorno de codigo abierto (mayor personalización) - Flexibilidad en cuanto a los dispositivos disponibles 	<ul style="list-style-type: none"> - App Store con cuotas de acceso más altas (99\$ anuales) - Cohesión de diseño en las diferentes App. - Funcionamiento nativo, pero en dispositivos IOS solamente - Entorno de código cerrado (mayor seguridad y consistencia)

4. ¿Cuál es mejor y por qué?

Si queremos diseñar una app para el entorno Apple lo mejor es usar Xcode con lenguaje Swift, pues correrán de forma nativa en ese tipo de dispositivos, sin embargo, en caso de no tener claro el alcance que queremos dar a nuestra App lo más inteligente es optar por una IDE que permita el trabajo multiplataforma como ReactNative (JAVAScript)

Analizando el factor económico y las barreras de entrada al entorno, destacamos que la PlayStore de Google (Android) es más accesible, con cuotas más reducidas (25\$ en pago único) y su sistema de verificación automatizada permite una entrada rápida de los desarrollos y actualizaciones. La AppStore (IOS) tiene un acceso más limitado, revisiones manuales y unas cuotas de entrada más elevadas (99\$ anuales)

En cuanto a las directrices de diseño de los entornos: El de Apple es más estricto dando cohesión e intuitividad entre las Apps (aprendiendo a usar una, casi las conoces todas) mientras que la libertad que permite Android hace que, en ocasiones, pueda complicarse la intuitividad y accesibilidad.

¿Cuál es mejor?

Dependerá principalmente de tus objetivos. Si no se tienen claros estos objetivos, en mi opinión, la mejor opción podría ser recurrir a una IDE que permita el desarrollo multiplataforma (ReactNative con lenguaje JAVAScript).

Así las incompatibilidades entre entornos se minimizan enormemente, facilitando desarrollar una app que funcione de forma fluida, diligente e intuitiva en ambos sistemas, a pesar de que en este caso, si queremos que ambas versiones sean idénticas, tendremos que fijarnos por las directrices de diseño de Apple, que al ser más estrictas podrían ser una barrera real de acceso a la AppStore en caso de usar las directrices de Guidelines de Android, mientras que al revés este problema se minimiza por la mayor flexibilidad presente en la publicación en PlayStore.