

Activity 01

Luis Mario Ayala Castellanos 174902

Luis Julian Alvarez Armenta 163247

Abraham Yair Silva Figueroa 173089

Emilio Ortiz Jaquim 172360

1. Tabla comparativa de opciones de programación

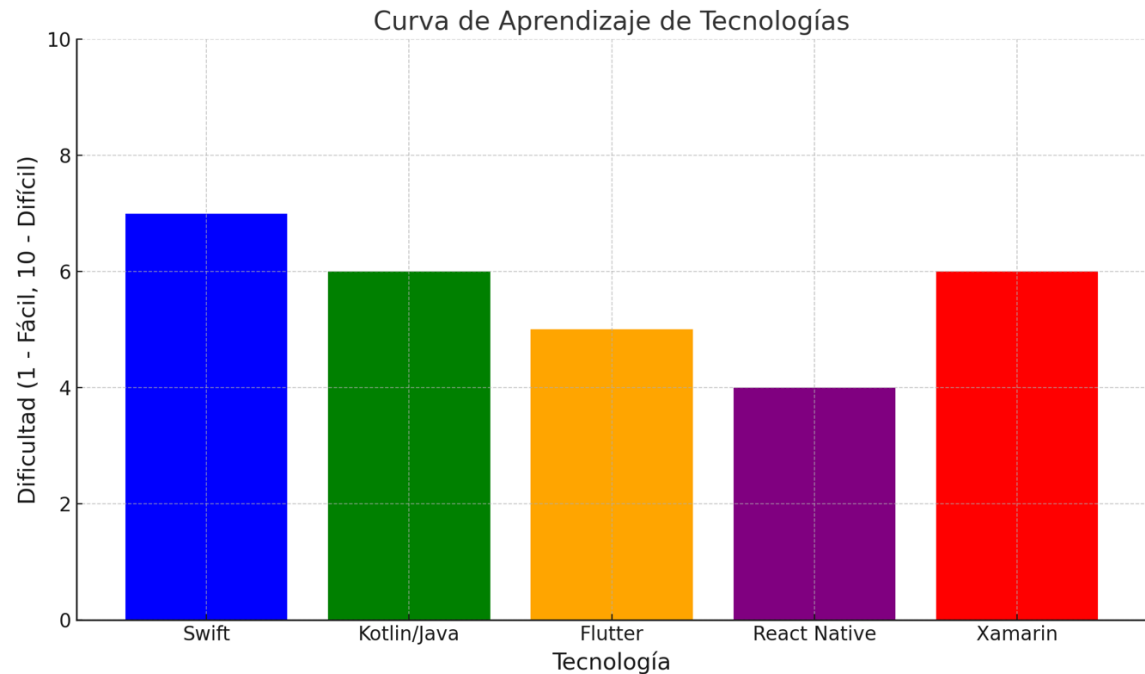
Framework/Plataforma	Lenguaje Principal	Soporte de Comunidad	Curva de Aprendizaje	Ventajas	Desventajas
Swift (iOS)	Swift	Amplio	7	Integración nativa, rendimiento óptimo, soporte de Apple.	Exclusivo de iOS, curva de aprendizaje más alta.
Kotlin/Java (Android)	Kotlin/Java	Amplio	6	Excelente para Android, herramientas bien integradas.	Específico para Android, puede ser más complejo que Flutter.
Flutter	Dart	Muy amplio	5	UI altamente personalizable, una sola base de código.	Necesidad de aprender Dart, tamaño mayor de las apps.
React Native	JavaScript	Muy amplio	4	Fácil de aprender, compatible con múltiples plataformas.	Rendimiento menor en comparación con nativo.
Xamarin	C#	Amplio	6	Soporte de Microsoft, código reutilizable.	Requiere entorno de Microsoft, menos flexible en UI.

2. Curva de aprendizaje

La siguiente tabla muestra los valores estimados para la curva de aprendizaje (escala del 1 al 10):

Tecnología	Dificultad estimada
Swift	7

Kotlin/Java	6
Flutter	5
React Native	4
Xamarin	6



3. Análisis de políticas de Android y iOS

Aspecto	iOS (Apple)	Android (Google)
Publicación en tienda	Aprobación estricta (calidad y privacidad).	Menos restrictiva, pero más riesgos de apps.
Costo	\$99/año para desarrolladores.	\$25 pago único para desarrolladores.
Privacidad	Obligatorio App Tracking Transparency (ATT).	Permisos más flexibles, pero fragmentados.
Soporte técnico	Documentación completa, pero cerrada.	Comunidad más amplia y diversa.

Implementar políticas en iOS es más desafiante debido a las estrictas regulaciones de calidad y privacidad, mientras que Android es más accesible pero menos seguro.

4. Conclusión

Basado en el análisis presentado, podemos concluir que iOS presenta políticas más desafiantes para los desarrolladores debido a su riguroso control de calidad, sus requisitos de privacidad y el costo recurrente. Por otro lado, Android ofrece un entorno más accesible, pero menos regulado, lo cual podría implicar

riesgos para los usuarios finales.

En cuanto al desarrollo nativo versus multiplataforma, el desarrollo nativo es ideal para proyectos que requieren un rendimiento óptimo y una experiencia de usuario altamente personalizada. Sin embargo, las opciones multiplataforma son más prácticas para startups y pequeños equipos debido a su accesibilidad y eficiencia en tiempo.