

Propuesta Técnica de Migración (RFC)de Animación Móvil basada en Lottie

Lead Developer

12 de diciembre de 2025

Introducción

En el contexto de crecimiento de la aplicación móvil de la startup, el equipo de Marketing ha solicitado la inclusión de animaciones de alta calidad durante el proceso de onboarding con el objetivo de mejorar la experiencia del usuario. Sin embargo, el equipo de QA ha identificado problemas técnicos significativos relacionados con el uso actual de secuencias PNG y animaciones GIF, tales como el incremento considerable del tamaño del APK y caídas de rendimiento en dispositivos de gama media.

El presente informe tiene como objetivo proponer formalmente la migración hacia la tecnología **Lottie**, basada en el documento técnico “*Introducing Lottie (Airbnb Engineering)*”, evaluando el impacto técnico, la arquitectura de la solución y la estrategia de despliegue recomendada.

1. Diagnóstico del Problema

1.1. Impacto técnico de archivos de imagen pesados

El uso de secuencias PNG y archivos GIF para representar animaciones complejas genera múltiples problemas a nivel técnico:

- **Incremento del tamaño del APK:** cada frame se almacena como una imagen independiente, lo que provoca una duplicación innecesaria de información visual. Como resultado, el tamaño del APK ha aumentado aproximadamente en 40 MB.

- **Consumo elevado de memoria:** durante la ejecución, los frames deben decodificarse continuamente y cargarse en memoria, incrementando el uso de RAM y la frecuencia del recolector de basura.
- **Caídas de framerate:** el renderizado de imágenes rasterizadas y su escalado en tiempo real generan sobrecarga en el hilo principal de la interfaz, afectando la fluidez de la animación y produciendo pérdidas de frames, especialmente en dispositivos de gama media.

Estos factores impactan directamente en la experiencia del usuario y en la percepción de calidad de la aplicación.

1.2. Costo-beneficio del código nativo manual

Una alternativa tradicional para evitar archivos pesados consiste en reescribir las animaciones manualmente en código nativo. No obstante, este enfoque presenta desventajas significativas:

- Requiere miles de líneas de código altamente acoplado al diseño visual.
- El código resulta frágil y difícil de mantener ante cambios menores en la animación.
- Incrementa el tiempo de desarrollo, pruebas y corrección de errores.

Tal como se menciona en el documento de Airbnb, este enfoque produce “*thousands of lines of brittle animation code*”, lo que eleva el costo de mantenimiento a largo plazo. En contraste, Lottie permite describir las animaciones como datos estructurados, reduciendo significativamente la complejidad del código nativo.

2. Arquitectura de la Solución

La solución propuesta se basa en un pipeline de producción de animaciones que separa claramente el diseño visual del renderizado en la aplicación.

2.1. Pipeline de producción

El flujo completo de la arquitectura es el siguiente:

1. Diseño de animaciones vectoriales en **Adobe After Effects**.
2. Exportación mediante el plugin **Bodymovin**, que convierte la animación a un archivo **JSON**.
3. Almacenamiento del archivo JSON en un repositorio o CDN.
4. Descarga dinámica del archivo mediante **Network Loading**.
5. Renderizado nativo de la animación usando la librería **Lottie** en la aplicación móvil.

El formato JSON describe keyframes, interpolaciones, formas vectoriales y tiempos, permitiendo un renderizado eficiente y adaptable a múltiples resoluciones sin pérdida de calidad.

3. Estrategia de Despliegue

3.1. Actualización de animaciones sin publicar una nueva versión

Para campañas estacionales como Navidad o Black Friday, se propone utilizar el mecanismo de **Network Loading**. Este enfoque permite descargar las animaciones desde un servidor remoto sin necesidad de lanzar una nueva versión de la aplicación en las tiendas oficiales.

Las ventajas principales de esta estrategia son:

- Reducción del tiempo de despliegue de campañas.
- Mayor autonomía para el equipo de Marketing.
- Posibilidad de modificar o revertir animaciones de forma inmediata.

3.2. Optimización de memoria mediante caché

En el caso de animaciones repetitivas, como botones de “Like” o indicadores de carga, Lottie permite almacenar en caché los archivos JSON ya parseados. De esta forma:

- Se evita reprocesar la animación en cada uso.
- Se reduce el consumo de CPU y memoria.
- Se mejora la fluidez general de la interfaz.

Esta optimización resulta clave para mantener un rendimiento estable en dispositivos con recursos limitados.

4. Conclusión

La migración desde animaciones basadas en PNG y GIF hacia una arquitectura apoyada en Lottie representa una mejora significativa tanto a nivel técnico como operativo. Esta solución reduce el tamaño del APK, mejora el rendimiento gráfico, disminuye la complejidad del código y permite una mayor flexibilidad en la actualización de contenidos animados.

Se recomienda aprobar esta propuesta y adoptar Lottie como estándar de animación para el onboarding y las microinteracciones de la aplicación móvil.