

# Informe de planificación y desarrollo de nuevas funcionalidades en Spotify

## Introducción

Spotify es una de las plataformas líderes en streaming de música y podcasts. Para mantener su competitividad e innovación, se plantea el desarrollo de una nueva actualización que incorpora dos funcionalidades:

- Requisito funcional: Modo Karaoke con puntuación en tiempo real.
- Requisito no funcional: Optimización de rendimiento para reducir el consumo de batería en dispositivos móviles.

Estas dos funcionalidades buscan mejorar la experiencia del usuario, tanto en el uso interactivo como en la eficiencia técnica del sistema.

## Etapa de Planificación

En esta fase inicial del proyecto, se define el qué, por qué y para quién se desarrolla la funcionalidad de modo karaoke con puntuación en Spotify, con el objetivo principal de incrementar la interacción social y el entretenimiento entre usuarios, al mismo tiempo que se mejora la eficiencia de la aplicación mediante una reducción en el consumo energético en dispositivos móviles. Para hacerlo se requerirá un equipo de desarrolladores frontend y backend, ingenieros de audio y machine learning especializados en reconocimiento de voz. Entre los riesgos identificados destacan problemas de precisión en el reconocimiento de voz para el karaoke y posibles incompatibilidades con dispositivos de baja gama en la optimización.

## Etapa de Análisis

Se detallan los requisitos funcionales y no funcionales del sistema para asegurar una implementación precisa. Entre los requisitos funcionales, el sistema debe permitir a los usuarios activar el modo karaoke, mostrar letras sincronizadas y otorgar puntuaciones en tiempo real basadas en afinación y ritmo. En cuanto a los no funcionales, la aplicación debe reducir el consumo de batería en un 15% durante sesiones de más de una hora, optimizando procesos en segundo plano. Los actores involucrados incluyen al usuario final como cantante, el servidor de Spotify que procesa datos de voz y puntuaciones, y la app móvil que muestra letras y resultados. Como ejemplo de casos de uso para el karaoke, se consideran UC-01: selección de canción por el usuario; UC-02: activación del modo karaoke; UC-03: visualización de letras y evaluación en tiempo real por el sistema; y UC-04: almacenamiento y muestra de la puntuación.

## Etapa de Diseño

En la etapa de diseño, se optó por una arquitectura de microservicios para dividir el sistema en componentes independientes, cada uno con una función específica: un microservicio para sincronizar las letras de las canciones, otro para el reconocimiento de voz y evaluación, y uno más para gestionar las puntuaciones y estadísticas de los usuarios. La app de Spotify integrará un nuevo "Karaoke Mode", permitiendo a los usuarios ver las letras sincronizadas y su puntuación en tiempo real. Esta estructura modular permite una mayor escalabilidad, flexibilidad y facilidad de mantenimiento del sistema, al mismo tiempo que asegura una experiencia de usuario fluida y eficiente.

## Etapa de Desarrollo

La etapa de desarrollo se centra en la implementación técnica utilizando lenguajes y tecnologías específicas para construir la funcionalidad de manera eficiente. En el frontend móvil, se emplean Kotlin para Android, Swift para iOS y React Native para componentes compartidos; en el backend, Java/Scala con microservicios en Kubernetes; y para machine learning, Python con TensorFlow o PyTorch para el modelo de reconocimiento de voz. Se aplican buenas prácticas como el encapsulamiento y modularidad (separando el módulo de karaoke), control de versiones en Git con commits claros, e integración continua (CI/CD) mediante Jenkins o GitHub Actions. El entregable es el código fuente completamente documentado y gestionado en un repositorio Git, asegurando un desarrollo colaborativo y mantenible.

## Etapa de Pruebas

En la etapa de pruebas, se verifica la calidad y funcionalidad del sistema a través de diversos niveles para identificar y corregir errores antes del lanzamiento.

- **Unitarias:** verificar funciones básicas (ej. sincronización de letra).
- **Integración:** validar que karaoke interactúe bien con backend.
- **Pruebas de usabilidad:** validar experiencia de usuario con grupos piloto.
- **Pruebas de rendimiento:** medir consumo de batería en móviles.
- **Pruebas de seguridad:** evitar abuso del reconocimiento de voz y fugas de datos.

## Etapa de Implementación y Despliegue

Finalmente, en la etapa de implementación y despliegue, se pone en producción la nueva versión de la app de manera controlada para minimizar impactos. Se aplica un despliegue gradual mediante canary release, comenzando con el 5% de usuarios Premium, y se utilizan feature flags para activar o desactivar el karaoke según regiones. El monitoreo se realiza con logs en tiempo real a través de Grafana y Prometheus, y se incluye un plan de rollback en caso de errores graves. El entregable es la nueva versión de Spotify disponible en la App Store y Google Play, asegurando una transición suave y segura para los usuarios.

## Conclusión

La nueva actualización de Spotify incluiría:

1. **Funcionalidad Karaoke con puntuación en tiempo real (requisito funcional):** mejora la interacción social y el entretenimiento.
2. **Optimización del consumo de batería (requisito no funcional):** mejora la experiencia técnica y la eficiencia de la app.

Seguir las **etapas del SDLC** asegura que el desarrollo sea ordenado, seguro y escalable.

## Repositorio de referencia

En GitHub existen proyectos que abordan funcionalidades similares:

- **ArturBa/spotify-karaoke** → Cliente de Spotify con letra sincronizada estilo karaoke.  
<https://github.com/MiguelGalleg05/spotify-karaoke>
- **gedrick/SpotifyKaraoke** → Muestra letras en tiempo real de Spotify.  
<https://github.com/MiguelGalleg05/SpotifyKaraoke-ref>