



ACTIVIDAD COLABORATIVA

PROCESO DE DESARROLLO DE SOFTWARE (SPOTIFY)

Integrantes:

Ximena Guerrero Carvajal

Marco Antonio Zhoe
Vargas Lira

Proceso de desarrollo de software en Spotify

Acciones que se realizan: Definición del alcance del proyecto: Se establecen las funcionalidades iniciales (streaming de música, creación de playlists, perfil de usuario, reproducción offline) y las características deseadas a largo plazo (podcasts, recomendaciones personalizadas, interacción social).

Estudio de mercado y benchmarking: Análisis de plataformas similares como Apple Music, Amazon Music o YouTube Music. Se identifican sus puntos fuertes y débiles para mejorar la propuesta de valor.

Identificación de públicos objetivo: Determinar a quién va dirigido (jóvenes, adultos, creadores de contenido, empresas, etc.).

Formación del equipo de desarrollo: Se conforman equipos especializados: frontend, backend, base de datos, inteligencia artificial, diseño UI/UX, QA (aseguramiento de calidad), seguridad y soporte.

Cronograma de trabajo: Se establece una línea de tiempo con fases clave y entregables:

- Semana 1: Reunión inicial y análisis de competencia.
- Semana 2: Definición de requerimientos y tecnología.
- Semana 3: Diseño preliminar y plan de pruebas.
- Semana 4 en adelante: Desarrollo por sprints de 2 semanas.

Gestión de presupuesto: Estimación de gastos en servidores, derechos de autor/licencias, sueldos, herramientas de desarrollo y marketing.

2. Fase de Análisis

Objetivo principal: Profundizar en lo que necesita el usuario y lo que debe hacer el sistema.

Acciones que se realizan:

Recopilación de requisitos funcionales:

- ✓ Registro e inicio de sesión.
- ✓ Búsqueda de música por artista, canción o género.
- ✓ Crear, editar y eliminar playlists.
- ✓ Guardar canciones para escuchar sin conexión.
- ✓ Compartir canciones o listas.
- ✓ Recopilación de requisitos no funcionales:
- ✓ Alta disponibilidad del servicio (24/7).
- ✓ Reproducción fluida sin interrupciones (mínimo buffering).
- ✓ Compatibilidad multiplataforma (Android, iOS, Windows, Web).
- ✓ Seguridad en el manejo de datos personales y pagos.

Análisis de riesgos:

- Caídas del servidor en horas pico.

- Infracciones de derechos de autor.
- Uso no autorizado de cuentas.

Análisis técnico: Se evalúan tecnologías a usar, como bases de datos (PostgreSQL, Cassandra), servicios de nube (AWS, Google Cloud), y lenguajes de programación (JavaScript, Python, Node.js).

3. Fase de Diseño

Objetivo principal: Crear el plan técnico y visual del sistema antes de programar.

Acciones que se realizan:

Diseño de arquitectura del software: Se decide una arquitectura de microservicios, donde cada parte (reproducción, autenticación, recomendaciones) funciona como un módulo independiente.

Diseño de la interfaz gráfica (UI): Se diseñan los componentes visuales que verá el usuario (botones, reproductor, menús, sliders) usando herramientas como Figma o Adobe XD.

Experiencia de usuario (UX): Se piensa en la navegación intuitiva, fluidez, tiempos de carga, accesibilidad y estética de la app.

Diseño de base de datos: Se crean esquemas para almacenar canciones, usuarios, playlists, historial de reproducción y preferencias.

Documentación técnica: Se escriben manuales de arquitectura, diagramas UML, flujos de navegación, y criterios de aceptación para el equipo de desarrollo.

4. Fase de Desarrollo

Objetivo principal: Construir el software siguiendo lo planeado y diseñado.

Acciones que se realizan:

Programación de la interfaz de usuario (Frontend): Se desarrolla la app con tecnologías como React Native (móvil) o React.js (web), creando pantallas de inicio, búsqueda, perfil, reproducción, etc.

Desarrollo del backend: Se construyen APIs REST o GraphQL con Node.js o Python para gestionar usuarios, subir canciones, autenticar pagos, generar recomendaciones, etc.

Integración con servicios de streaming: Se usan códecs de audio como Ogg/Vorbis o AAC para transmitir archivos de manera eficiente.

Desarrollo de funciones inteligentes: Se implementa inteligencia artificial y machine learning para sugerir música según el historial del usuario, preferencias de otros similares, o hábitos de escucha.

Gestión de pagos y suscripciones: Integración de sistemas de pago como Stripe o PayPal para planes Premium.

5. Fase de Pruebas

Objetivo principal: Asegurarse de que el sistema funciona correctamente y sin errores.

Acciones que se realizan:

Pruebas unitarias:

- Validar que al hacer clic en “Play” una canción se reproduzca correctamente.
- Verificar que la función de búsqueda devuelva los resultados esperados según el texto ingresado.

Pruebas de integración:

- ❖ Verificar que el login funcione con bases de datos y tokens de sesión.
- ❖ Confirmar que las playlists creadas se guardan correctamente.
- ❖ Pruebas de rendimiento y carga:
- ❖ Medir cuántos usuarios pueden estar activos sin que el sistema se ralentice.
- ❖ Simular miles de peticiones simultáneas para ver la estabilidad.

Pruebas de compatibilidad: Asegurar que la app funcione correctamente en diferentes dispositivos, resoluciones de pantalla y sistemas operativos.

6. Fase de Mantenimiento

Objetivo principal: Garantizar la estabilidad del software a lo largo del tiempo.

Acciones que se realizan:

Actualizaciones regulares: Se añaden nuevas funciones como podcasts, videos musicales, integración con redes sociales o mejoras en la personalización.

Corrección de errores: Se solucionan bugs reportados por los usuarios (por ejemplo, canciones que no cargan o fallos de sesión).

Soporte técnico: Se responde a dudas o problemas reportados mediante correo o chat.

Monitoreo continuo: Se utiliza software de monitoreo como Datadog o New Relic para detectar caídas o errores en tiempo real.

Respaldos y restauración:

- Se hacen copias de seguridad automáticas diarias en la nube de los datos más importantes.
- Se crean puntos de restauración para volver al estado anterior en caso de fallas.
- Se implementa replicación de servidores para evitar pérdidas por fallos de hardware.

FASES DEL DESARROLLO DE SOFTWARE DE SPOTIFY

Spotify es un servicio de música, podcasts y vídeos digitales que te da acceso a millones de canciones y a otro contenido de creadores de todo el mundo. Las funciones básicas, como escuchar música, son totalmente gratis, pero también tienes la opción de mejorar tu cuenta con Spotify Premium.

Fase de planeación



Alcance del proyecto:

Desarrollar una plataforma de streaming de música que permita a los usuarios buscar, reproducir, guardar y compartir canciones, crear listas de reproducción, y recibir recomendaciones personalizadas según sus gustos.

CRONOGRAMA DE TRABAJO:



Semana 1:
Estudio de mercado y definición de funcionalidades principales (búsqueda, reproducción, listas de reproducción).



Semana 2:
Identificación de socios musicales y licencias necesarias.



Semana 3:
Definición del equipo de desarrollo y asignación de tareas técnicas (frontend, backend, base de datos).

Fase de análisis



Requerimiento 1:

El sistema debe permitir a los usuarios registrarse, iniciar sesión y mantener sus preferencias (listas, historial, canciones favoritas).

Requerimiento 2:

El sistema debe ser capaz de transmitir audio en tiempo real (streaming) sin interrupciones, con adaptabilidad a distintas velocidades de conexión.

Fase de diseño



Metodología propuesta:

Se adopta la metodología ágil Scrum, permitiendo dividir el desarrollo en sprints de dos semanas. Esto favorece la entrega continua de funcionalidades, revisión constante del producto y adaptación a los comentarios de los usuarios y stakeholders.

Fase de desarrollo



Procedimiento 1:

Programación de la interfaz de usuario en React Native (para app móvil) y React.js (para versión web), permitiendo una experiencia fluida y consistente.

Procedimiento 2:

Desarrollo del backend en Node.js con integración de servicios de streaming, almacenamiento de datos en bases como PostgreSQL y gestión de archivos de audio en servidores distribuidos.

Fase de pruebas



Prueba unitaria 1:

Verificar que el sistema reproduzca correctamente canciones al hacer clic en "Play", incluyendo funciones como pausar, avanzar y retroceder.

Prueba unitaria 2:

Probar que la funcionalidad de búsqueda devuelva resultados precisos según el nombre del artista, canción o álbum.

Fase de mantenimiento



Estrategia de respaldos:

Implementar respaldos automáticos diarios de la base de datos de usuarios y música, con replicación en la nube (por ejemplo, AWS S3 y RDS). Además, se crean puntos de restauración cada semana y se realiza monitoreo constante para detectar posibles fallos del sistema.

Reflexión sobre la importancia de cada fase en el desarrollo del software de Spotify

El desarrollo de una plataforma tan compleja como Spotify requiere que cada fase del proceso sea ejecutada con precisión, ya que todas son fundamentales para garantizar un servicio estable, intuitivo y con alto rendimiento.

La Planeación: En Spotify, esta fase es esencial para definir el alcance del producto: desde qué funciones incluir (reproducción, playlists, podcasts) hasta las alianzas con sellos discográficos. Sin esta planificación estratégica, sería imposible coordinar equipos globales y recursos técnicos, afectando la escalabilidad de la plataforma.

El Análisis: Esta etapa permite identificar lo que realmente necesita el usuario: recomendaciones personalizadas, reproducción sin cortes, sincronización entre dispositivos, entre otros. Si se omite esta fase, Spotify podría ofrecer funciones poco relevantes o una experiencia deficiente, perdiendo competitividad frente a otros servicios como Apple Music o YouTube Music.

El Diseño: Spotify se destaca por su diseño visual atractivo y una experiencia de usuario sencilla e intuitiva. Esta fase define la arquitectura técnica (microservicios, bases de datos, servidores de audio) y visual (interfaz en móviles, web y escritorio). Un mal diseño afectaría tanto la estética como el rendimiento del sistema.

El Desarrollo: Aquí se construye todo lo planificado. En Spotify, se programan funciones clave como el reproductor, los algoritmos de recomendación musical y la gestión de contenido. Si esta fase no está bien organizada, se pueden generar fallos graves que afecten millones de usuarios en tiempo real.

Las pruebas: Spotify debe asegurar que su plataforma funcione perfectamente bajo demanda global. Se realizan pruebas unitarias e integradas para garantizar que funciones como "reproducir", "buscar" o "guardar canción" operen sin errores. Omitir esta fase generaría errores visibles que dañarían la reputación del producto.

El Mantenimiento: Una plataforma como Spotify está en constante evolución. Esta fase permite lanzar actualizaciones, corregir errores detectados por los usuarios y mantener la seguridad del sistema. También incluye el respaldo continuo de datos y la protección de información sensible.

Conclusiones

Spotify es un claro ejemplo de cómo cada fase del desarrollo de software es clave para el éxito y sostenibilidad de una aplicación a gran escala. Si alguna de estas fases no se ejecuta correctamente, podrían surgir fallas técnicas, mala experiencia de usuario o incluso problemas legales (por ejemplo, en derechos de autor).

La fase más crítica en el caso de Spotify es el análisis, ya que todo el diseño del sistema, las recomendaciones musicales y las decisiones técnicas dependen de una comprensión profunda de lo que el usuario necesita. Sin un buen análisis, se corre el riesgo de invertir tiempo y dinero en funciones que no aportan valor.

En conclusión, el desarrollo de Spotify no es solo programación, sino una serie de pasos estratégicos, técnicos y creativos que, al integrarse correctamente, hacen posible una experiencia musical fluida, personalizada y confiable para millones de usuarios en todo el mundo.