

IMPLEMENTACION BUS ROUTRACK

1. Lanzamiento

Objetivo: Hacer que la aplicación esté disponible para todos los usuarios, tanto en dispositivos móviles como en la web.

Publicación en tiendas de aplicaciones:

Google Play Store (Android): Lo primero es crear una cuenta de desarrollador en Google Play, lo que implica un pago único de \$25 USD. Luego, hay que preparar la aplicación generando un archivo APK o AAB firmado. También se deben proporcionar descripciones, capturas de pantalla y videos promocionales. El proceso de revisión puede tardar entre 2 y 7 días.

Apple App Store (iOS): Aquí, debes crear una cuenta de desarrollador en Apple, que cuesta \$99 USD al año. La aplicación se sube a través de Xcode, cumpliendo con las guías de diseño de Apple. Este proceso de revisión generalmente toma de 1 a 2 semanas.

Despliegue en servidores: Escoger un proveedor de servicios en la nube como AWS, Google Cloud, Heroku o Azure es esencial. Hay que configurar varios servicios: para el backend, desplegar la API y las bases de datos; para la CDN, utilizar una red de distribución de contenido que mejore los tiempos de carga; y, por último, asegurarse de que la comunicación sea segura configurando HTTPS y certificados SSL.

Comunicación con los usuarios: Es fundamental crear una campaña de lanzamiento que incluya redes sociales, anuncios locales y contacto con empresas de transporte. También es útil tener material de apoyo

como guías rápidas, tutoriales y un sistema de ayuda dentro de la aplicación.

2. Monitoreo

Objetivo: Hacer un seguimiento constante del rendimiento de la aplicación para identificar y resolver problemas rápidamente.

Métricas clave:

Se deben evaluar varias métricas técnicas, como la latencia de la API y el tiempo de carga de las pantallas principales. También es importante monitorear la interacción del usuario, revisando el número de usuarios activos diarios y mensuales (DAU/MAU), así como las rutas más buscadas y los puntos de congestión en el sistema.

Herramientas de monitoreo:

En el backend, se pueden usar New Relic o Datadog para el monitoreo de servidores y aplicaciones, y centralizar logs con ELK Stack (ElasticSearch, Logstash, Kibana). Para el frontend, Google Analytics o Firebase Analytics pueden ayudar a rastrear eventos de usuario. Para errores y fallos, Sentry es útil para capturar y seguir errores en tiempo real.

Automatización: Es recomendable configurar alertas para abordar problemas críticos, como caídas de servidor o tiempos de respuesta altos. También se deben generar reportes semanales o mensuales sobre rendimiento y uso.

3. Mantenimiento

Objetivo: Asegurar que la aplicación siga actualizada y funcione correctamente a medida que evolucionan las necesidades de los usuarios.

Actualizaciones regulares: Esto implica incorporar nuevas funcionalidades según lo que los usuarios necesiten, como rutas adicionales o un modo offline. También se debe trabajar en la optimización continua para mejorar el rendimiento y reducir errores.

Corrección de errores:

Es vital priorizar los errores: aquellos que son críticos y afectan el uso básico de la aplicación deben ser resueltos en un plazo de 24 a 48 horas, mientras que los errores menores, que son más estéticos o de baja prioridad, se pueden resolver en futuros lanzamientos.

Retroalimentación: Es esencial monitorear las reseñas en las tiendas de aplicaciones y mantener un canal de comunicación abierto con los usuarios para que puedan reportar problemas directamente.

Soporte técnico: Crear una sección de ayuda dentro de la aplicación es clave, al igual que configurar un sistema de tickets o soporte, como Zendesk o correo electrónico.

Estrategia a largo plazo: Hay que evaluar y ajustar la arquitectura para soportar un mayor número de usuarios. También es importante realizar auditorías de seguridad de manera periódica para proteger los datos sensibles y analizar métricas de uso que puedan identificar áreas de mejora y nuevas oportunidades.