

## Desarrollo del Backend

**Objetivo:** Construir la parte del servidor encargada de gestionar la lógica de negocio, administrar los datos y facilitar la comunicación entre la base de datos y el frontend.

### Tecnologías a usar:

- **Lenguaje:** Node.js (con gran velocidad y capacidad de escalado) o Python (utilizando Flask o Django).
- **Base de datos:** PostgreSQL para almacenar datos estructurados como rutas, paradas y usuarios.
- **Redis (opcional):** Para manejar información en tiempo real, como la ubicación de los autobuses.
- **Servicios en la nube:** AWS, Google Cloud o Azure para el despliegue de la aplicación.

### Tareas principales:

- Configurar la base de datos y establecer la conexión con el servidor.
- Implementar la lógica de negocio: registro y autenticación de usuarios, gestión de rutas y paradas, y actualización en tiempo real de la ubicación de los autobuses.
- Crear una API RESTful que permita: enviar datos al frontend (como ubicación, tiempos estimados y rutas disponibles) y recibir solicitudes del usuario (incluyendo búsquedas y configuraciones).
- Realizar pruebas unitarias e integradas para asegurar el funcionamiento correcto del servidor.

## Desarrollo del Frontend

**Objetivo:** Diseñar y crear una interfaz gráfica que interactúe directamente con los usuarios.

### **Tecnologías a usar:**

- **Framework:** React Native para desarrollar una aplicación móvil que funcione en múltiples plataformas (Android e iOS).
- **API de mapas:** Google Maps API para mostrar ubicaciones y rutas interactivas.
- **Librerías de UI:** Material UI o Ant Design para componentes estilizados y responsivos.

### **Tareas principales:**

- Desarrollar pantallas clave que incluyan: un mapa interactivo con autobuses y rutas en tiempo real.
- Funcionalidades de búsqueda para rutas y paradas.
- Información detallada de las rutas seleccionadas.
- Implementar filtros y opciones de personalización: elegir un destino o mostrar solo los autobuses cercanos.
- Integrar con la API del backend para obtener datos dinámicos.
- Realizar pruebas de usabilidad y corregir errores en dispositivos reales.

### **Integración de APIs**

**Objetivo:** Conectar la aplicación con los servicios externos que son necesarios para su funcionamiento.

### **APIs a integrar:**

- **Google Maps API:** Para mostrar mapas interactivos con marcadores para rutas, paradas y autobuses, además de calcular rutas óptimas y tiempos estimados.
- **Firebase:** Para enviar notificaciones push a los usuarios (como alertas de llegada y cambios en rutas).
- **Servicios de geolocalización:** Para rastrear en tiempo real la posición de los autobuses usando GPS integrado.
- **Servicios de análisis:** Google Analytics o Mixpanel para monitorear el uso de la aplicación y mejorarla a partir de datos reales.

## **Control de Versiones**

**Objetivo:** Asegurar que los cambios en el código sean rastreables, seguros y colaborativos.

**Plataforma:** Git, utilizando un repositorio en GitHub, GitLab o Bitbucket.

### **Estrategia de ramas:**

- Main branch: Código estable y listo para producción.
- Develop branch: Código en desarrollo con nuevas funcionalidades.
- Feature branches: Ramas específicas para cada funcionalidad (por ejemplo, feature/map-integration).

### **Herramientas complementarias:**

- CI/CD: Automatizar el proceso de despliegue con herramientas como Jenkins o GitHub Actions.