**Etapa de Desarrollo**



BusTrack

**Fase de Desarrollo para BusTrack**

**1. Configuración del Entorno de Desarrollo**

* **Lenguaje de Programación:**
  + **Dart** (con Flutter) para crear una aplicación multiplataforma.
  + **Node.js o Python** para el backend si necesitas lógica compleja en el servidor.

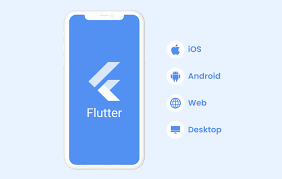


Ilustración Flutter

* **Frameworks y Herramientas:**
  + **Flutter**: Desarrollo de la app móvil para Android e iOS.
  + **Firebase**: Backend como servicio para autenticación, base de datos y notificaciones push.
  + **Google Maps API o Mapbox**: Integración de mapas y rastreo GPS.
  + **Git y GitHub**: Control de versiones para colaborar y gestionar el progreso.
* **Organización de Ramas en GitHub:**
  + **Main**: Contendrá la versión estable del proyecto.
  + **Develop**: Uso para desarrollo activo.
  + **Feature branches**: Ramas separadas para funcionalidades específicas, como "rutas", "notificaciones" o "mapa".

**2. Desarrollo del Backend**

* **APIs y Servidores:**
  + Creación de endpoints para:
    - Recibir datos GPS de los buses.
    - Proveer información sobre rutas, líneas y precios filtrados.
  + Almacenamiento de datos en Firebase Realtime Database o Firestore.
* **Almacenamiento de Datos:**
  + Base de datos para rutas, buses y usuarios.
  + Almacenamiento temporal para la ubicación GPS en tiempo real.
* **Seguridad:**
  + Autenticación con Firebase Authentication para usuarios registrados.
  + Encriptación de datos sensibles durante las transmisiones.

**3. Desarrollo del Frontend**

* **Interfaz de Usuario:**
  + **Pantalla principal:**
    - Mapa interactivo con ubicación en tiempo real de los buses.
  + **Filtros:**
    - Opción para seleccionar rutas, líneas de bus y rango de precios.
  + **Notificaciones:**
    - Alertas sobre la proximidad de buses o cambios en rutas.
* **Flujo de Interacción:**
  + Selección de destino → Visualización de rutas sugeridas → Rastreo del bus en tiempo real.
  + Configuración personalizada para guardar rutas frecuentes.

**4. Integración y Pruebas**

* **Pruebas de Integración:**
  + Verificación de la conexión entre API, bases de datos y frontend.
  + Medición de la latencia en actualizaciones del GPS.
* **Pruebas Automatizadas:**
  + Uso de herramientas como **GitHub Actions** para CI/CD.
  + Pruebas de:
    - Precisión de localización.
    - Usabilidad y rendimiento en diferentes dispositivos.

**5. Metodología de Desarrollo**

* **Metodología Ágil (Scrum):**
  + Organización en sprints (2 semanas cada uno).
  + Ejemplo de objetivos por sprint:
    - Sprint 1: Configuración del mapa y rastreo básico.
    - Sprint 2: Implementación de filtros y notificaciones.
    - Sprint 3: Optimización de interfaz y pruebas de rendimiento.
  + Revisión y ajustes al final de cada sprint según retroalimentación.

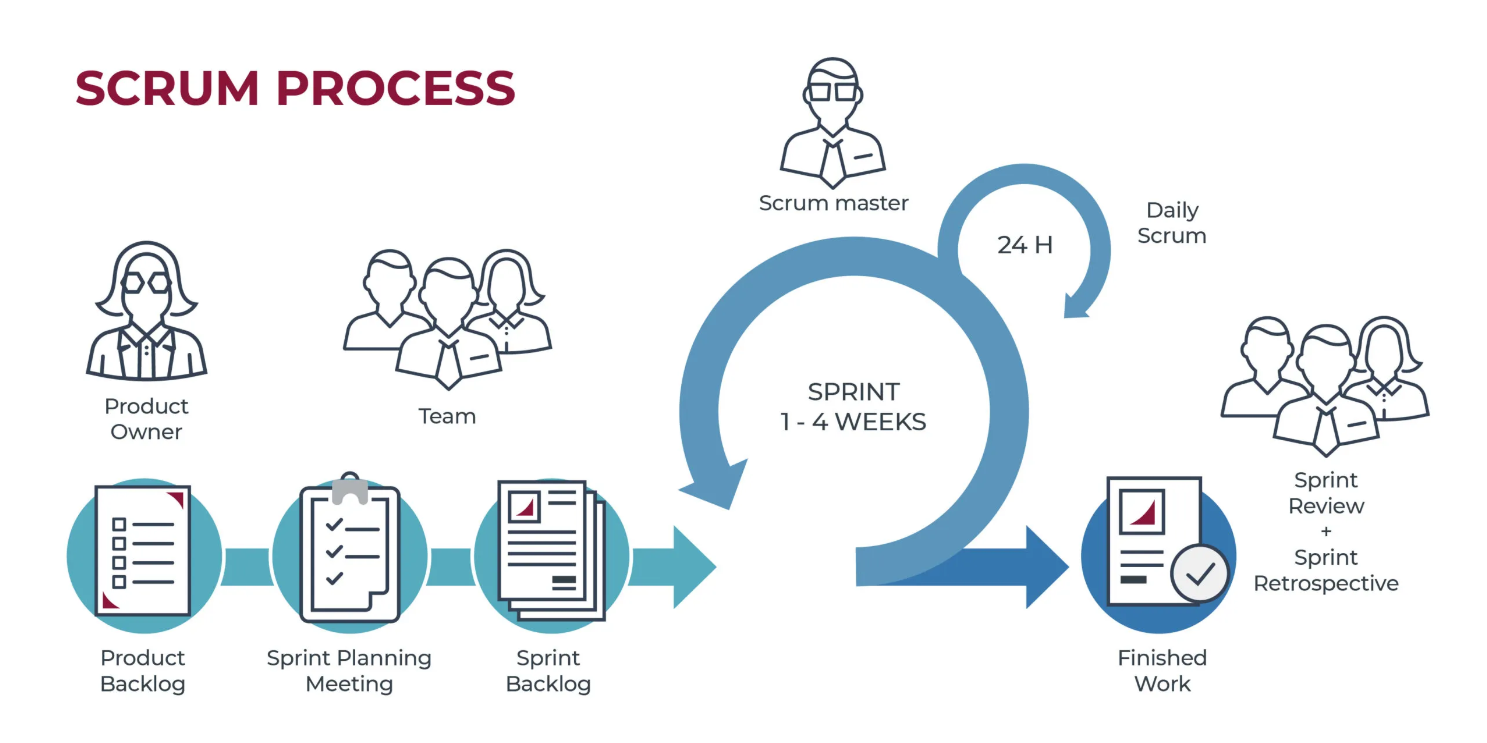


Ilustración Scrum Process

**6. Documentación**

* **Documentación Técnica:**
  + Configuración del entorno de desarrollo.
  + Guía sobre las dependencias y la estructura del código.
* **Manual de Usuario:**
  + Instrucciones sobre cómo usar la app para rastrear buses y configurar preferencias.
  + Resolución de problemas comunes (ejemplo: permisos de GPS).

**7. Entrega Final**

* Publicación de la app en **Google Play Store** y **Apple App Store**.
* Configuraciones predeterminadas para facilitar la instalación y uso inicial.
* Revisión final para asegurar compatibilidad con las últimas versiones de Android e iOS.