### **Épica 1: Gestión de Recorridos en el Campo**

Como usuario, quiero poder registrar y gestionar los recorridos realizados en el campo, de manera que pueda optimizar mis operaciones agrícolas con datos precisos de posición, hora, precisión y velocidad promedio.

#### **Historia de Usuario 1.1: Registro de Recorridos**

* **Como**: Operador en terreno
* **Quiero**: Registrar la posición, hora, precisión y velocidad promedio durante mis recorridos en el campo.
* **Para**: Obtener datos precisos y detallados de mis movimientos y actividades en el terreno.
* **Criterios de Aceptación**:
  + Los datos deben ser registrados en un archivo JSON con el formato correcto.
  + La interfaz debe ser fácil de usar y permitir iniciar y terminar recorridos de manera intuitiva.
  + En caso de baja señal GPS, la aplicación debe continuar registrando utilizando datos inerciales.

#### **Historia de Usuario 1.2: Uso de Sensores Inerciales**

* **Como**: Operador en terreno
* **Quiero**: Que la aplicación utilice los sensores inerciales del dispositivo móvil para complementar los datos de GPS.
* **Para**: Asegurar una mayor precisión en la georreferenciación incluso en condiciones de baja señal GPS.
* **Criterios de Aceptación**:
  + La aplicación debe combinar datos del GPS con los de los sensores inerciales (acelerómetro y giroscopio) para mejorar la precisión.
  + Los datos mejorados deben visualizarse en tiempo real durante el recorrido.
  + La aplicación debe verificar la disponibilidad de los sensores antes de iniciar el recorrido.

### **Épica 2: Sincronización de Datos con la Base de Datos Central**

Como usuario, quiero sincronizar los datos de mis recorridos con una base de datos central para que puedan ser analizados y utilizados en la toma de decisiones.

#### **Historia de Usuario 2.1: Sincronización de Recorridos**

* **Como**: Operador en terreno
* **Quiero**: Sincronizar mis datos de recorrido con la base de datos central al finalizar un recorrido.
* **Para**: Asegurar que la información esté disponible para análisis posterior por parte de los administradores.
* **Criterios de Aceptación**:
  + Los datos deben ser enviados correctamente a través de la API y almacenados en la base de datos sin errores.
  + En caso de falta de conexión a Internet, los datos deben almacenarse localmente y sincronizarse cuando la conexión se restablezca.
  + El usuario debe recibir una confirmación de la sincronización exitosa.

### **Épica 3: Optimización y Seguridad de la Aplicación**

Como usuario, quiero que la aplicación funcione de manera eficiente y segura, para que mis datos estén protegidos y la experiencia de usuario sea fluida.

#### **Historia de Usuario 3.1: Manejo de Errores y Seguridad de Datos**

* **Como**: Desarrollador
* **Quiero**: Implementar medidas de seguridad y manejo de errores en la aplicación.
* **Para**: Garantizar que los datos sean protegidos y que cualquier problema sea informado claramente al usuario.
* **Criterios de Aceptación**:
  + Implementar validaciones para todas las entradas y salidas de datos.
  + Proporcionar mensajes de error claros y orientados al usuario final.
  + Confirmar cada intercambio de información con la base de datos para garantizar la integridad de los datos.

#### **Historia de Usuario 3.2: Compatibilidad y Rendimiento en Dispositivos Android**

* **Como**: Usuario final
* **Quiero**: Que la aplicación funcione correctamente en dispositivos Android, optimizando el uso de sensores y gestionando eficientemente los recursos del dispositivo.
* **Para**: Asegurar un rendimiento fluido y evitar interferencias con los datos de posición debido a optimizaciones del sistema operativo.
* **Criterios de Aceptación**:
  + La aplicación debe ser desarrollada utilizando React Native y Expo para aprovechar herramientas y librerías avanzadas.
  + Considerar los constantes cambios en Android para evitar problemas de compatibilidad.

Sprints:

### **Sprint 0: Configuración de Entornos**

* **Objetivo**: Configuración de entornos de desarrollo, instalación de herramientas necesarias. (React Native, Expo, TailwindCSS)
* **Entregable**: Entorno de desarrollo completamente configurado.

### **Sprint 1: Implementación del Registro de Recorridos**

* **Objetivo**: Desarrollo de la funcionalidad de registro de recorridos.
* **Entregable**: Módulo que registra posición, hora, precisión y velocidad promedio, y guarda los datos en formato JSON.

### **Sprint 2: Navegación Inercial Básica**

* **Objetivo**: Implementar la integración con sensores inerciales.
* **Entregable**: Módulo que captura datos de navegación inercial y los integra con los datos de GPS.

### **Sprint 3: Sincronización de Datos**

* **Objetivo**: Desarrollo de la sincronización de datos con la base de datos mediante API.
* **Entregable**: Funcionalidad que permite sincronizar recorridos con la base de datos central.

### **Sprint 4: Validación y Seguridad de Datos**

* **Objetivo**: Implementar validaciones de datos y seguridad en la transmisión.
* **Entregable**: Módulo que asegura la integridad de los datos y maneja errores y desconexiones.

### **Sprint 5: Optimización del modulo**

* **Objetivo**: Mejorar el rendimiento del modulo y adaptarla a diferentes dispositivos Android.
* **Entregable**: Aplicación optimizada y lista para diferentes condiciones de uso.

### **Sprint 6: Pruebas de Integración**

* **Objetivo**: Realizar pruebas de integración de todos los módulos.
* **Entregable**: Aplicación completamente integrada y funcional, lista para pruebas finales.