

Informe Técnico

Operación Campus UIDE

Virginia Mora

Juan Fuentes

Universidad Internacional del Ecuador

Materia: Programación Móvil

13 de febrero de 2026

Índice

1. Introducción	3
2. Arquitectura del Sistema	3
3. Gestión de Permisos y Mapa Base	3
3.1. Arquitectura de Permisos	3
3.2. Mapa Base	4
4. Geofencing y Lógica de Aproximación	4
4.1. Indicador de Proximidad Dinámico	4
4.2. Restricción de Precisión	4
5. Diagnóstico por Visión Artificial	4
6. Intervención en Realidad Aumentada	5
7. Optimización Energética	5
8. Test de Campo y Validación	5
9. Conclusiones	6

1. Introducción

El presente documento describe el desarrollo técnico de la aplicación móvil *Operación Campus UIDE*, creada como un laboratorio experimental en el campus Loja de la Universidad Internacional del Ecuador. El objetivo principal fue integrar tecnologías avanzadas como geolocalización de alta precisión, visión artificial mediante Machine Learning y Realidad Aumentada (AR), dentro de una arquitectura limpia y optimizada energéticamente.

La aplicación permite detectar focos de contaminación virtual y ejecutar una intervención digital inmersiva para su “sanación”.

2. Arquitectura del Sistema

La aplicación fue desarrollada en Flutter siguiendo una arquitectura por capas:

- **Capa Presentation:** Widgets, interfaz gráfica, animaciones y navegación.
- **Capa Domain:** Lógica de negocio, validaciones, cálculo de distancia y reglas de activación.
- **Capa Data:** Acceso a sensores (GPS, cámara), integración con TensorFlow Lite y servicios del sistema.

Para la gestión de estados se implementó **Riverpod**, lo que permite desacoplar completamente la interfaz de usuario de la lógica del negocio. Esta decisión garantiza escalabilidad, mantenibilidad y mejor control del ciclo de vida de los servicios activos.

3. Gestión de Permisos y Mapa Base

3.1. Arquitectura de Permisos

Se utilizó el paquete `permission_handler`. No se emplearon diálogos estándar del sistema como única respuesta. En su lugar, se diseñaron vistas personalizadas que explican al usuario por qué la aplicación requiere acceso a:

- Ubicación
- Cámara

Si el usuario niega permisos, la aplicación despliega una pantalla informativa que impide continuar hasta su habilitación.

3.2. Mapa Base

Se integró Google Maps para mostrar:

- Ubicación actual del usuario
- Marcador en coordenadas críticas (zona de laboratorios)

El mapa fue configurado en modo oscuro para coherencia visual con la temática Cyber-Ecología.

4. Geofencing y Lógica de Aproximación

Se implementó un sistema de escucha continua de cambios de ubicación. La lógica principal incluye:

- Cálculo dinámico de distancia al objetivo
- Activación progresiva de eventos según proximidad

4.1. Indicador de Proximidad Dinámico

Se diseñó un radar animado cuya:

- Velocidad aumenta al reducir la distancia
- Color cambia de rojo (lejos) a verde (cerca)

4.2. Restricción de Precisión

La cámara de reconocimiento solo se habilita cuando el *accuracy* del GPS es menor a 5 metros del punto exacto. Esto reduce falsos positivos y mejora eficiencia.

5. Diagnóstico por Visión Artificial

Se integró un modelo TensorFlow Lite para clasificación en tiempo real desde el flujo de cámara.

La intervención solo se habilita cuando:

- Se detecta un residuo válido (botella, papel, etc.)
- La confianza del modelo supera el 80 %

El procesamiento se realiza localmente en el dispositivo, evitando dependencia de red y reduciendo latencia.

6. Intervención en Realidad Aumentada

Se utilizó un plugin de AR para:

- Anclar un objeto 3D interactivo en el entorno real
- Activar animación de éxito al tocar el objeto
- Mostrar panel con datos simulados de radiación UV (inspirado en el proyecto Sol-máforo)

La activación ocurre únicamente tras validación exitosa del modelo de visión artificial.

7. Optimización Energética

La aplicación implementa un contador interno de peticiones GPS y adapta la frecuencia de muestreo:

- Distancia mayor a 500 m: muestreo reducido
- Distancia menor a 50 m: alta frecuencia

Este enfoque reduce consumo energético durante desplazamientos largos y aumenta precisión solo en la zona crítica.

8. Test de Campo y Validación

Se realizó una prueba real documentada en video continuo (máximo 2 minutos), mostrando:

1. Inicio de la aplicación
2. Gestión de permisos
3. Caminata hacia el punto objetivo
4. Activación de proximidad
5. Reconocimiento ML
6. Aparición e interacción con modelo 3D

9. Conclusiones

La combinación de Riverpod, arquitectura por capas y procesamiento local con TensorFlow Lite permitió construir una aplicación robusta, escalable y eficiente. La optimización del muestreo GPS indica una preocupación real por la eficiencia energética, mientras que la integración de AR y visión artificial posiciona la solución dentro de estándares tecnológicos actuales del mercado móvil.