Aforos - Especificación funcional y prompt técnico para Android

1. Objetivo del producto

Crear una app Android llamada "Aforos" que permita estimar caudal en campo a partir de un video de 10 s y una sección rectangular definida por el usuario. La app calcula: velocidad media en la sección del recuadro usando visión por computadora, área hidráulica con los datos ingresados y reporta caudal en L/s y m³/s. Genera además una imagen del fotograma con la sección marcada y un recuadro de resultados.

2. Alcance inicial v0.1

- Grabación de video de 10 s en modo horizontal con CameraX.
- Pantalla de selección de ROI rectangular sobre el video grabado.
- Entrada manual de ancho [m], largo [m] y profundidad promedio [m] de la sección del recuadro.
- Cálculo de área hidráulica: A = ancho × profundidad. Largo se usa para escala métrica en píxeles y control de calidad.
- Estimación de velocidad superficial en la ROI con flujo óptico denso. Conversión a velocidad media mediante un factor configurable.
- Cálculo de caudal: Q = A × Vmedia. Salida en m³/s y L/s.
- Overlay de resultados sobre un fotograma del video y opción para guardar la imagen PNG sin fondo adicional al overlay.
- Menú inicial con: Iniciar medición, Instrucciones, Ir a garuas.com.

3. Flujo de usuario

1. Home
    - Botón Iniciar medición
    - Botón Instrucciones
    - Link externo a https://garuas.com/
2. Iniciar medición
    - Vista cámara con contador 10 s y botón Grabar
    - Al finalizar: previsualización del clip y botón Continuar
3. Definir sección
    - Slider para elegir fotograma base o usar promedio
    - Herramienta para dibujar un rectángulo ROI ajustable
    - Campos de entrada: Ancho [m], Largo [m], Profundidad promedio [m]
    - Botón Calcular
4. Proceso
    - Barra de progreso con mensaje Analizando velocidad del agua...
5. Resultado

    - Imagen del fotograma con ROI y vectores de flujo estilizados
    - Velocidad estimada [m/s]
    - Área [m²]
    - Caudal [m³/s] y [L/s]
    - Botones: Guardar imagen, Recalcular, Nueva medición, Compartir

4. Requisitos funcionales

- RF1: Grabación de video 1080p a 30 fps durante 10 s usando CameraX.
- RF2: Permitir selección de ROI rectangular editando vértices y arrastre.
- RF3: Ingresar manualmente ancho, largo y profundidad. Validar valores positivos y rangos razonables.
- RF4: Calcular área A = ancho × profundidad.
- RF5: Estimar velocidad superficial Vs con flujo óptico denso sobre la ROI usando OpenCV. Filtrar vectores no fiables.
- RF6: Convertir Vs a velocidad media Vmedia aplicando coeficiente k entre 0.75 y 0.95 configurable en Ajustes. Valor por defecto 0.85.
- RF7: Calcular Q = A × Vmedia y mostrar m³/s y L/s.
- RF8: Generar imagen PNG del fotograma con overlay de ROI y cuadro de resultados. Guardar en almacenamiento local y permitir compartir.
- RF9: Persistir últimos parámetros usados para agilizar siguientes mediciones.
- RF10: Mostrar pantalla de Instrucciones con guía breve de toma del video y medición.

5. Requisitos no funcionales

- RNF1: Kotlin, Android Studio Iguana o superior.
- RNF2: minSdk 26, targetSdk 35.
- RNF3: Procesamiento en dispositivo, sin conexión a internet.
- RNF4: Tiempo total de procesamiento menor a 10 s en dispositivos medios.
- RNF5: Interfaz clara y minimalista, modo oscuro automático.

6. Permisos

- CAMERA
- RECORD_AUDIO opcional si se requiere audio, en v0.1 desactivado
- READ_MEDIA_VIDEO o READ_EXTERNAL_STORAGE según versión
- WRITE_EXTERNAL_STORAGE solo si aplica en versiones antiguas

7. Librerías sugeridas

- CameraX: camera-core, camera-video, camera-lifecycle, camera-view
- OpenCV Android SDK para flujo óptico denso (Farnebäck) y estabilización ligera
- Jetpack: ViewModel, LiveData o Flow, Navigation, DataStore
- Kotlin Coroutines para tareas en background

8. Algoritmo de velocidad - resumen

- Estabilizar ligeramente el clip dentro de la ROI con feature tracking.
- Convertir frames a escala de grises y aplicar blur suave.
- Calcular flujo óptico denso Farnebäck en pares de frames consecutivos dentro de la ROI.
- Proyectar vectores al eje principal del flujo estimado por PCA o por dirección dominante.
- Filtrar outliers por magnitud y ángulo. Promediar percentil 50 a 90.
- Convertir de píxeles por frame a m/s usando la escala métrica derivada de la longitud conocida dentro del rectángulo y fps real.
- Aplicar coeficiente k para obtener velocidad media.

9. Datos derivados y fórmulas

- Área [m^2] = ancho × profundidad
- Escala [m/píxel] = longitud_real [m] ÷ longitud_en_píxeles [px] del lado mayor del rectángulo
- Velocidad superficial [m/s] = magnitud_promedio [px/frame] × escala [m/px] × fps
- Velocidad media [m/s] = k × Vs
- Caudal [m^3/s] = Área × Vmedia
- Caudal [L/s] = Caudal [m^3/s] × 1000

10. UI y estados

- Home: 3 botones principales, logotipo Aforos opcional
- Grabación: visor con retícula, indicador fps, temporizador 10 s
- ROI: canvas con puntos de control, campos numéricos con teclado decimal
- Cálculo: barra de progreso lineal con mensajes
- Resultado: tarjeta con datos y fotograma anotado

11. Manejo de errores

- Video no válido o fps bajo: sugerir repetir toma
- ROI muy pequeña: solicitar ampliar
- Sin textura detectable: instrucción para añadir marcadores flotantes visibles o repetir video
- Valores fuera de rango: bloquear cálculo con mensaje claro

12. Exportación

- Imagen PNG en carpeta Aforos/Resultados con nombre YYYYMMDD_HHMMSS.png

- Archivo JSON opcional con metadatos de la medición: ancho, largo, profundidad, k, fps, Vs, Vmedia, Q, dispositivo

13. Optimización propuesta

- Opción de guía de calibración rápida con un objeto de referencia de longitud conocida en el plano de la superficie
- Suavizado temporal con media móvil para reducir ruido en Vs
- Preprocesado adaptativo para mejorar contraste en agua

14. Roadmap breve

- v0.1: MVP sin exportación JSON
- v0.2: Exportación JSON y pantalla Historial
- v0.3: Ajustes avanzados de flujo óptico y coeficiente k por tipo de canal

15. Prompt base para generador de código

Copia y pega el siguiente prompt en tu asistente de código para Android Studio. Ajusta según sea necesario.

"""
Quiero que generes una app Android llamada Aforos en Kotlin con minSdk 26 y targetSdk 35. Usa Jetpack Compose para UI y CameraX para video. Estructura por módulos si conviene o un solo módulo app.

Características:

- Mostrar el logo de Garúa en la pantalla Home, centrado arriba (drawable `garua_logo`).
- Implementar un splash screen cuyo ícono sea la gota de la tilde de la "Ú" del logo (Adaptive Icon `ic_aforos`).

1. Pantalla Home con tres acciones: Iniciar medición, Instrucciones, Ir a garuas.com. El tercero abre navegador del sistema a https://garuas.com/.
2. Iniciar medición: grabación de video de 10 s a 1080p y 30 fps con CameraX VideoCapture. Guardar temporalmente en cache interna.
3. Seleccionar ROI: luego de grabar, mostrar fotograma representativo y permitir dibujar y ajustar un rectángulo. Campos para Ancho [m], Largo [m], Profundidad [m] con validaciones. Botón Calcular.
4. Al Calcular, mostrar barra de progreso y ejecutar en un Dispatcher.IO el análisis con OpenCV: flujo óptico denso Farnebäck en la ROI sobre el clip. Estimar dirección dominante, filtrar outliers, convertir px/frame a m/s usando fps y escala por la longitud real indicada. k por defecto 0.85.

5. Calcular Área = ancho × profundidad, Vmedia = k × Vs, Q = Área × Vmedia. Mostrar resultados y fotograma con overlay del rectángulo y setas de flujo y un recuadro con Velocidad estimada, Área y Caudal en m³/s y L/s. Botones Guardar PNG, Compartir, Nueva medición.
6. Implementa capa DataStore para persistir el último k, y los últimos valores ingresados.
7. Maneja permisos CAMERA y lectura de media según API. Maneja estados de error con mensajes claros.

Tecnologías:

- Kotlin, Jetpack Compose, CameraX, OpenCV Android SDK, Coroutines, DataStore, Navigation Compose.

Entregables:

- Proyecto compilable en Android Studio con instrucciones README para configurar OpenCV.
- Código limpio con ViewModels y separación UI-logic. Comentarios en puntos clave del procesamiento de video.
  """

16. Instrucciones de uso para campo - borrador de pantalla

- Grabe en modo horizontal a 1 o 2 m de altura y enfoque la zona con textura visible en la superficie del agua.
- Dibuje el rectángulo cubriendo la sección a analizar.
- Ingrese ancho, largo y profundidad promedio en metros.
- Presione Calcular y espere el resultado.
- Guarde la imagen si necesita evidencia.

17. Notas

- Evitar usar guion largo. Usar guiones simples para listas.
- Nombre del paquete sugerido: com.garua.aforos

18. Branding en Home y Splash

- Incluir la imagen de marca provista (logo Garúa) en la pantalla Home, centrada arriba, con altura aprox. 160 dp y ajuste `ContentScale.Fit`. Debe respetar fondo blanco y márgenes laterales de 24 dp.
- Agregar un splash de inicio cuyo ícono sea la **gota** de la tilde de la “Ú” del logo. Usar Adaptive Icon:

    - `mipmap-anydpi-v26/ic_aforos.xml` (foreground vector con la gota) y `ic_aforos_background.xml` (color sólido del brand #117D73 aprox.).
    - En Android 12+, configurar `windowSplashScreenAnimatedIcon` con el mismo vector; animación opcional por escala.
- Colocar el archivo del logo como `res/drawable/garua_logo.png` (o `webp`). El gota se sugiere exportar como SVG para foreground.

Detalles de UI Home

- Layout en Compose: Column con `Image(garua_logo)` arriba, `Spacer`, y tres botones grandes: Iniciar medición, Instrucciones, Ir a garuas.com.
- Mantener contraste AA, tipografía sans geométrica.

19. Notas de assets

- Fuente sugerida: Inter/Roboto.
- Paleta base: verde Garúa (#117D73), gris oscuro (#243239), gris claro (#F2F5F5).
- Nombre de paquete se mantiene: `com.garua.aforos`.