MÁSTER EN EN INTELIGENCIA ARTIFICIAL Y DEEP LEARNING

ANTEPROYECTO

ANÁLISIS DE RENDIMIENTO EN GIMNASIO MEDIANTE VISIÓN POR COMPUTADOR

TFM elaborado por: Daniel_Romero_de_Miguel

Tutor/a de TFM: Daniel_Rubio_Yagüe

- Madrid 28/05/2025 -

Índice

Descripción del proyecto

Título

Análisis de rendimiento en gimnasio mediante visión por computador

Categoría

Inteligencia artificial: orientado a la aplicación de algoritmos y modelos de inteligencia artificial en sets de datos.

Tipo

Visión por computador + Series temporales

Contexto de negocio o tecnológico

El creciente interés por el fitness en España ha impulsado la demanda de herramientas que ayuden al usuario amateur a mejorar su técnica y medir su progreso de forma objetiva.

Este proyecto se centra en el análisis automático de vídeo de entrenamientos en gimnasio (pesas, máquinas y ejercicios de fuerza), empleando visión por computador para evaluar la postura, contar repeticiones y calcular métricas como velocidad de ejecución y rango de movimiento.

Técnicamente, se apoyará en librerías de estimación de poses como MediaPipe que permiten extraer los puntos clave del cuerpo en tiempo real desde la cámara del smartphone.

A esos datos de vídeo se suman series temporales de sensores procedentes de dispositivos wearables, integrándolos en una plataforma que cruza información multimodal para detectar anomalías de técnica y ofrecer feedback inmediato.

Aunque existen apps globales como Freeletics o Fitbod que ofrecen planes de entrenamiento con IA, ninguna se centra en análisis automático de vídeo para ejercicios de fuerza en el segmento amateur.

Esta solución aporta valor al usuario al sustituir la supervisión presencial, no siempre accesible ni económica, por un sistema automatizado capaz de identificar errores posturales y cuantificar el rendimiento en cada sesión.

Nuestra aplicación envía el vídeo de tus ejercicios a servidores en la nube, donde se ejecutan los modelos de visión por computador. De ese modo, el móvil solo recibe los resultados (postura, velocidad, número de repeticiones), evitando sobrecargar el dispositivo y manteniendo una respuesta rápida.

En pocas palabras, combinamos análisis de vídeo y datos de sensores para proporcionarte un informe automático de tu técnica y métricas clave de entrenamiento, sin necesidad de entrenadores presenciales ni equipos adicionales.

Objetivos de aprendizaje

General

- Desarrollar un sistema automatizado que, a partir de vídeo y datos de wearables, ofrezca métricas y recomendaciones de técnica deportiva y planes de entrenamiento personalizados.

Específicos

1. Implementar detección de ejercicios y estimación de pose con MediaPipe.
2. Diseñar pipeline de ingestión y preprocesado de datos de wearables.
3. Entrenar y evaluar modelos de detección de errores posturales.
4. Desplegar API REST contenedorizada para inferencia.

Resultados esperados

Prototipo operativo que reciba un vídeo de entrenamiento y datos de sensor, y devuelva:

- Métricas de forma (ángulos de articulaciones, cadencia, velocidad).
- Conteo automático de repeticiones y detección de errores (por ejemplo, rodilla colapsada al hacer una sentadilla).
- Dashboard web o demo en vídeo con los resultados.

Asignaturas / módulos relacionados

- **M1 - Las Herramientas del científico de datos:**
  Base para la ingestión y preprocesado de los datos de vídeo y sensores: manejo de Python, estructuras de datos (listas, diccionarios), lectura de ficheros CSV/JSON y limpieza de datos con Pandas.

- **M9 - Deep learning aplicada: NLP y visión artificial:**
  Implementación del análisis de vídeo: OpenCV, estimación de poses con CNN preentrenadas (OpenPose/MediaPipe), data augmentation y detección de puntos clave del cuerpo.

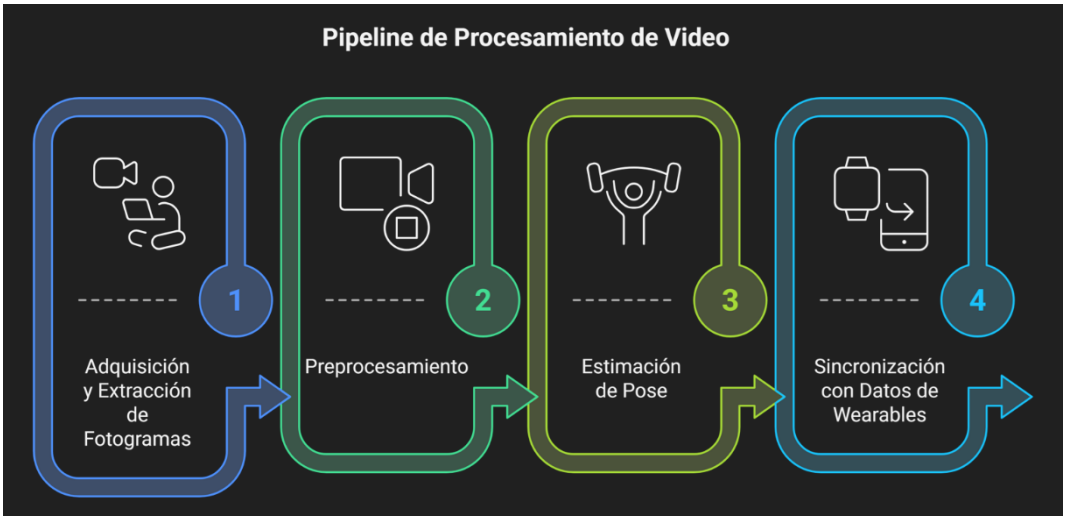
- **M8 - Series temporales y modelos predictivos: Optimización. Modelos de grafos:**
  Procesamiento y transformación de las señales de los wearables (acelerómetro, giroscopio, frecuencia cardíaca): creación de lags, medias móviles y extracción de características para integrarlas con el análisis de vídeo.

- **M7 - Cloud, MLops, productivización de modelos. Introducción a process mining:**
  Despliegue de la API y procesamiento en la nube: contenedores Docker, funciones serverless y pipelines CI/CD.

Métodos, materiales y tecnologías de uso potencial

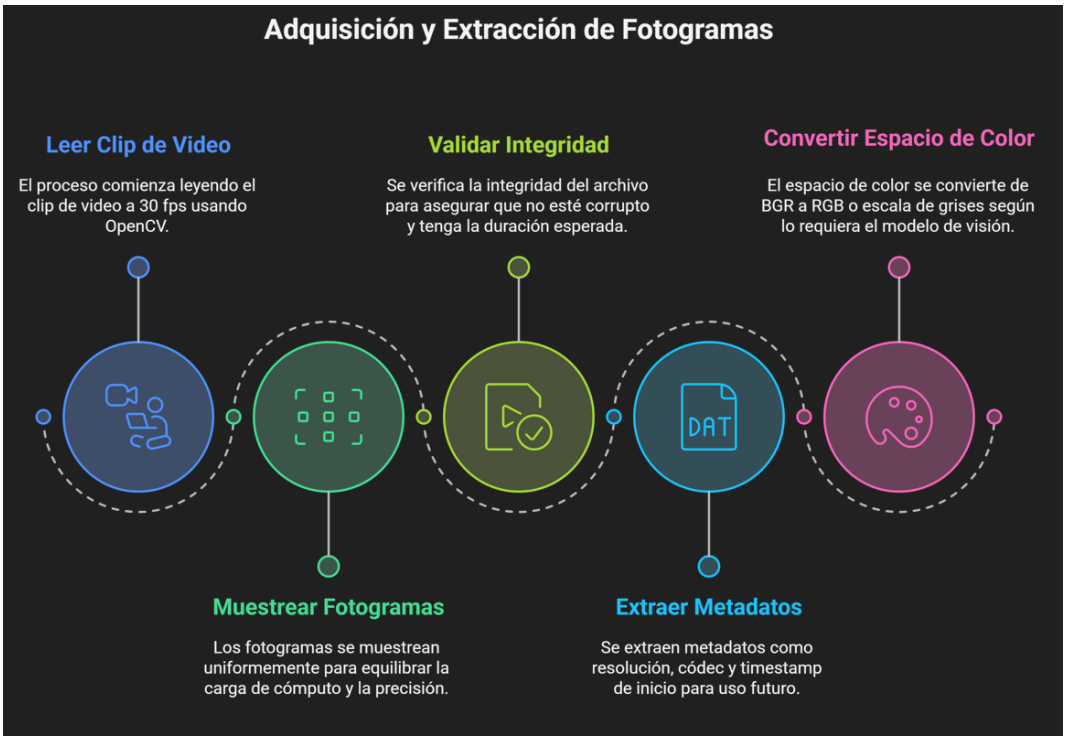
Pipeline de procesamiento de vídeo

El siguiente diagrama muestra de forma general las cuatro fases principales de nuestro pipeline de vídeo, desde la lectura del clip hasta la fusión con los datos de los wearables.



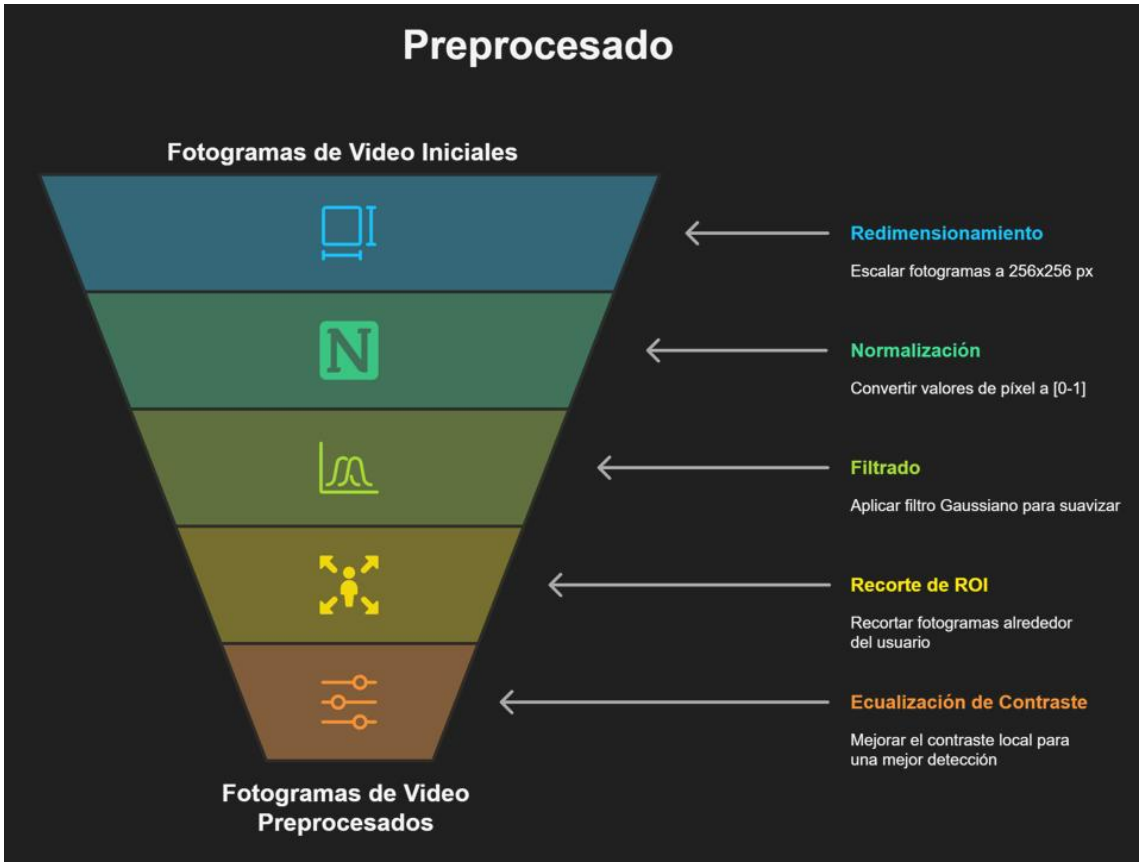
1. Adquisición y extracción de fotogramas

- Lectura de cada clip a 30 fps con OpenCV.
- Muestreo uniforme de fotogramas para equilibrar carga de cómputo y precisión.
- Validación de integridad: comprobar que el fichero no está corrupto y posee la duración esperada antes de procesar.
- Extracción de metadatos: capturar resolución original, códec y timestamp de inicio para futuras sincronizaciones.
- Conversión de espacio de color: pasar de BGR (OpenCV) a RGB o escala de grises según requiera el modelo de visión.



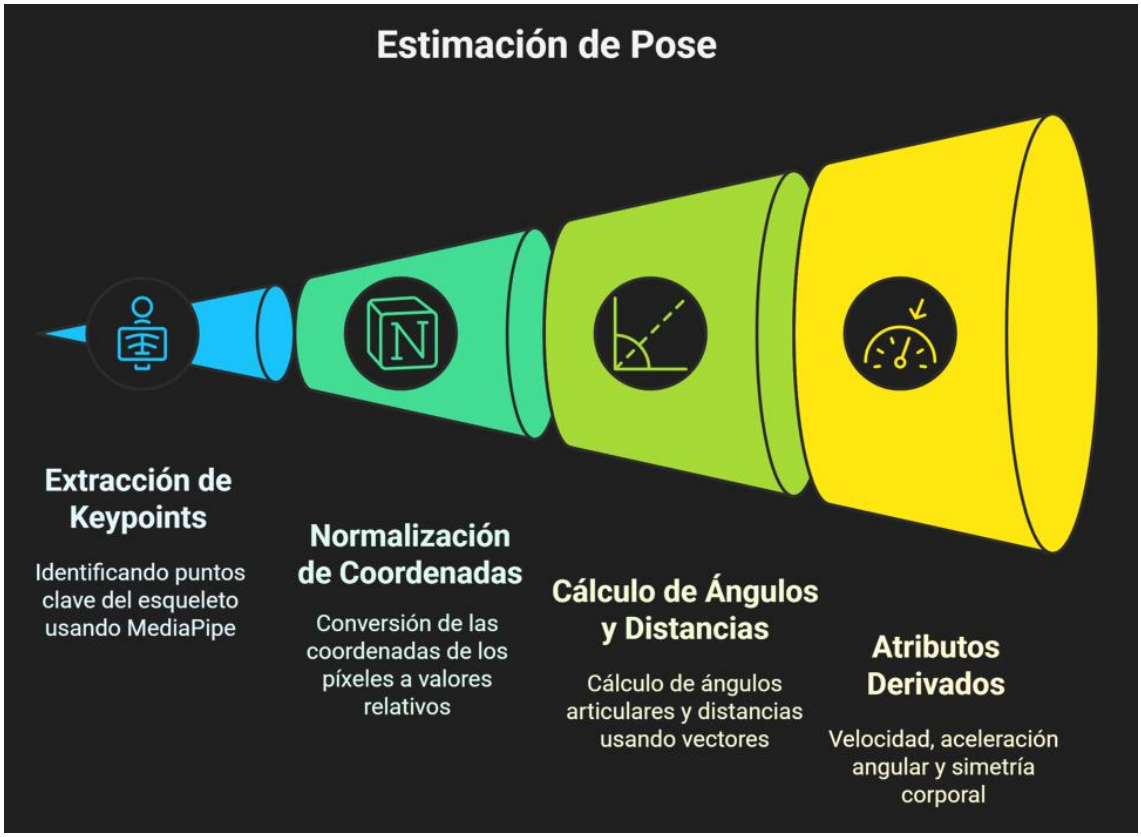
2. Preprocesado

- **Redimensionado:** Todos los fotogramas se escalan a 256×256 píxeles, garantizando un tamaño uniforme y reduciendo el coste computacional.
- **Normalización:** Convertimos los valores de píxel a tipo float32 y los escalamos a la franja [0 – 1], lo que favorece la estabilización del entrenamiento y acelera la convergencia de los modelos.
- **Filtrado básico:** Aplicamos un filtro de suavizado Gaussiano (kernel 3×3) para atenuar artefactos de compresión y reducir el ruido en los bordes, mejorando la precisión de la estimación de pose.
- **Recorte de ROI (Región de Interés):** Con un detector rápido de persona (HOG + SVM preentrenado o MobileNet SSD), recortamos cada fotograma al área donde aparece el usuario, centrando el análisis de poses y descartando fondos irrelevantes.
- **Ecualización adaptativa de contraste**
  Para asegurar que la estimación de pose no falle en condiciones de luz irregular, aplicamos CLAHE, una forma de ecualización de histograma que actúa por regiones y limita la amplificación excesiva directamente sobre el canal de luminancia de cada fotograma. Esto mejora el contraste local (resalta contornos y articulaciones) sin generar demasiado ruido, garantizando una detección de puntos clave más estable.



3. Estimación de pose

- **Extracción de keypoints con MediaPipe:**
  - Detectamos los puntos clave del esqueleto que el modelo proporcione (MediaPipe Pose, por defecto, extrae hasta 33 landmarks).
  - Ajustaremos el número y la selección de estos puntos en función de los ejercicios y de los resultados empíricos que vayamos obteniendo, priorizando siempre aquellos que aporten más información para el análisis de técnica.
- **Normalización de coordenadas**
  - Convertimos las coordenadas de píxel a valores relativos (0–1) basados en el ancho y alto del fotograma.
  - Además, centramos el origen en la cadera (o torso) para que las distancias y ángulos sean comparables independientemente de la posición de la cámara.
- **Cálculo de ángulos articulares y distancias**
  - Definimos vectores entre puntos clave (p. ej. cadera–rodilla, rodilla–tobillo) y calculamos los ángulos con la fórmula del producto escalar.
  - Medimos distancias relativas (p. ej. separación de pies como diferencia de coordenadas X normalizadas) para extraer features de anchura y postura.
- **Atributos derivados**
  - Velocidad y aceleración angular: derivadas temporales de los ángulos para cuantificar la rapidez de movimiento.
  - Simetría corporal: comparación de ángulos y posiciones izquierda vs. derecha para detectar desequilibrios.



4. Sincronización con datos de wearables

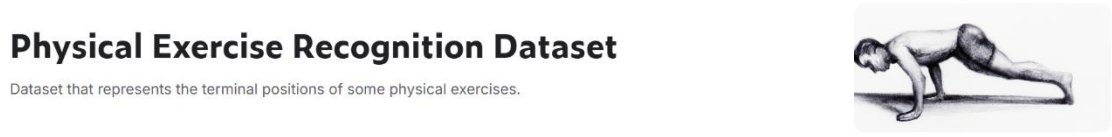
- Registro del timestamp tanto en el vídeo como en el sensor.
- Corrección de desajustes de reloj: aplicar un offset único si el smartphone y el wearable no van perfectamente sincronizados.
- Interpolación lineal de las lecturas de acelerómetro y pulso para estimar sus valores en cada timestamp de vídeo (calculando el valor intermedio entre las dos muestras de sensor más cercanas a cada fotograma).
- Relleno de huecos en la señal de los wearables en caso de desconexión, copiando el último valor registrado antes de la interrupción, para que cada fotograma disponga de datos completos.
- Unión de ambas fuentes en un solo DataFrame por fotograma con columnas: keypoints, ángulos, aceleración y pulso.

Etiquetado de fallos posturales

Dataset (Kaggle)

“Exercise-Recognition” de Kaggle aporta 15 000 clips en 14 ejercicios, etiquetados por tipo de movimiento (no por fallo).

https://www.kaggle.com/datasets/muhannadtuameh/exercise-recognition



El uso del dataset público “Exercise-Recognition” bajo licencia **CC BY-NC-SA 4.0** impide su explotación comercial directa. Para garantizar la viabilidad futura de la aplicación y evitar conflictos legales, se proponen dos estrategias de mitigación:

- **Negociación de licencia comercial**
  Contactar al autor original (Muhannad Tuameh) para solicitar una licencia que permita el uso con fines comerciales.
  Evaluar posibles términos: pago único, royalties por unidad vendida o condiciones de uso limitado.

- **Desarrollo de un dataset propio**
  Grabar un mínimo de 1 000 clips de ejercicios de fuerza (squat, bench press, deadlift), bajo consentimiento informado de voluntarios.

  Definir un protocolo de grabación homogéneo (iluminación, ángulo de cámara, duración) y criterios de etiquetado claros.

Nota: Durante la fase de prototipado y validación académica se mantendrá el entrenamiento con “Exercise-Recognition”. En la transición a la versión comercial, el modelo se reajustará o reentrenará exclusivamente con datos propios, de modo que ningún componente liberado esté sujeto a la restricción “NoComercial” de la licencia CC BY-NC-SA.

Etiquetado propio

- Generación de un subconjunto de vídeos de squat, deadlift y bench press grabados con smartphone.
- Etiquetado manual por experto (Yo): cada clip recibe etiquetas binarias de “correcto” vs. “incorrecto” según desviaciones de ángulo (p. ej. rodilla que pasa de vertical, posición del hombro incorrecta en un bench…).

Estrategia de validación

Validación cruzada

Para evaluar la robustez de nuestro modelo, dividimos todo el conjunto de datos (clips de Kaggle + grabaciones propias) en cinco bloques. Entrenamos el modelo cinco veces, usando cuatro bloques para entrenar y reservando uno distinto para probar cada vez. De este modo garantizamos que las secuencias de un mismo voluntario no aparezcan simultáneamente en el entrenamiento y en la prueba y evitar que el modelo aprenda patrones particulares de una persona y no se produzca un overfitting específico enfocado a esa persona

Métricas de evaluación

- **Precisión:** porcentaje de fallos detectados que eran realmente fallos.
- **Recall:** porcentaje de fallos reales que el modelo logra detectar.
- **F1:** combinación de precisión y recall en una única medida.
- **MAE:** medida de cuánto error, en promedio, existe entre los ángulos articulares reales y los estimados por el modelo.

Pruebas con usuarios

- Piloto con 5 - 10 voluntarios realizando 3 series de 10 repeticiones de cada ejercicio.
- Comparación de conteo automático de repeticiones vs. observación manual.

Aspectos legales y éticos

Firmaremos consentimiento informado con todos los voluntarios.

Los vídeos se almacenarán de forma cifrada y se borrarán al concluir la fase de desarrollo.

Toda información personal (nombre, demografía) se tratará de forma anónima y bajo normativa RGPD.

Tecnologías empleadas:

- Lenguaje y librerías: Python (OpenCV, MediaPipe, Pandas, NumPy), TensorFlow/Keras.
- Contenerización y MLOps: Docker, GitHub Actions, MLflow.
- Front-end de demostración**:** Streamlit o React para visualizar resultados en tiempo real.

MVP

- Detección de tipo de ejercicio (clasificación vídeo → ejercicio).
- Cálculo de ángulos clave y conteo de repeticiones con MediaPipe.
- Visualización básica de resultados (ángulos, repeticiones).

ANEXO: Elección de Frameworks y Tecnologías del proyecto

Estimación pose:

MediaPipe y OpenPose son dos frameworks de código abierto muy utilizados para la estimación de pose humana (pose estimation), es decir, para detectar y seguir las articulaciones y los movimientos del cuerpo a partir de imágenes o vídeo.

OpenPose

- **Origen**: Desarrollado por el grupo de investigación CMU Perceptual Computing Lab de la Universidad Carnegie Mellon.
- **Funcionamiento**:
    - Procesa cada fotograma con una red neuronal profunda (CNN) que localiza puntos clave del cuerpo (cabeza, hombros, codos, muñecas, caderas, rodillas, tobillos, etc.).
    - Genera un esqueleto 2D de la persona y puede detectar múltiples individuos en la misma imagen.
- **Características**:
    - Soporta también estimación de manos y rostro.
    - Preciso, pero relativamente pesado (requiere GPU para tiempo real).
- **Casos de uso**: análisis de técnica deportiva, interacción persona-máquina, animación, realidad aumentada.

MediaPipe

- **Origen**: Creado y mantenido por Google Research.
- **Funcionamiento**:
    - Ofrece “pipelines” optimizados para móviles y entornos web, basados en modelos ligeros de Machine Learning.
    - Detecta puntos clave de cuerpo, manos y cara en tiempo real incluso en dispositivos con recursos limitados.
- **Características**:
    - Muy eficiente y preparado para producción en Android, iOS y navegador (WebAssembly).
    - Incluye componentes modulares (por ejemplo, face mesh, hand tracking, holistic tracking) que se pueden combinar.

- **Casos de uso**: aplicaciones móviles de fitness, filtros faciales en vídeo, control gestual, telemedicina.

Diferencias clave

Aspecto	OpenPose	MediaPipe
Precisión	Muy alta	Alta, pero ligeramente menos precisa
Rendimiento	Requiere GPU potente	Optimizado para CPU y móviles
Facilidad de uso	Instalación y despliegue más complejos	Integración sencilla con ejemplos y APIs
Flexibilidad	Focalizado en pose y manos	Múltiples soluciones (cara, manos, cuerpo)

He optado por MediaPipe como tecnología de estimación de pose principal debido a su equilibrio entre precisión y eficiencia.

Al estar especialmente optimizado para ejecutarse en CPU de dispositivos móviles y navegadores web, permite ofrecer un análisis en tiempo real sin depender de infraestructuras de alto coste ni de GPUs dedicadas. Su compatibilidad nativa con Android, iOS y WebAssembly, junto con el soporte oficial de Google y ejemplos listos para usar, facilita su integración en la aplicación y garantiza un bajo consumo de recursos y batería.

De esta forma, es posible centralizar el procesamiento de vídeo directamente en el dispositivo del usuario (o, en casos de hardware limitado, con un servicio de inferencia ligero en la nube), asegurando una experiencia fluida, accesible y escalable para deportistas amateurs.