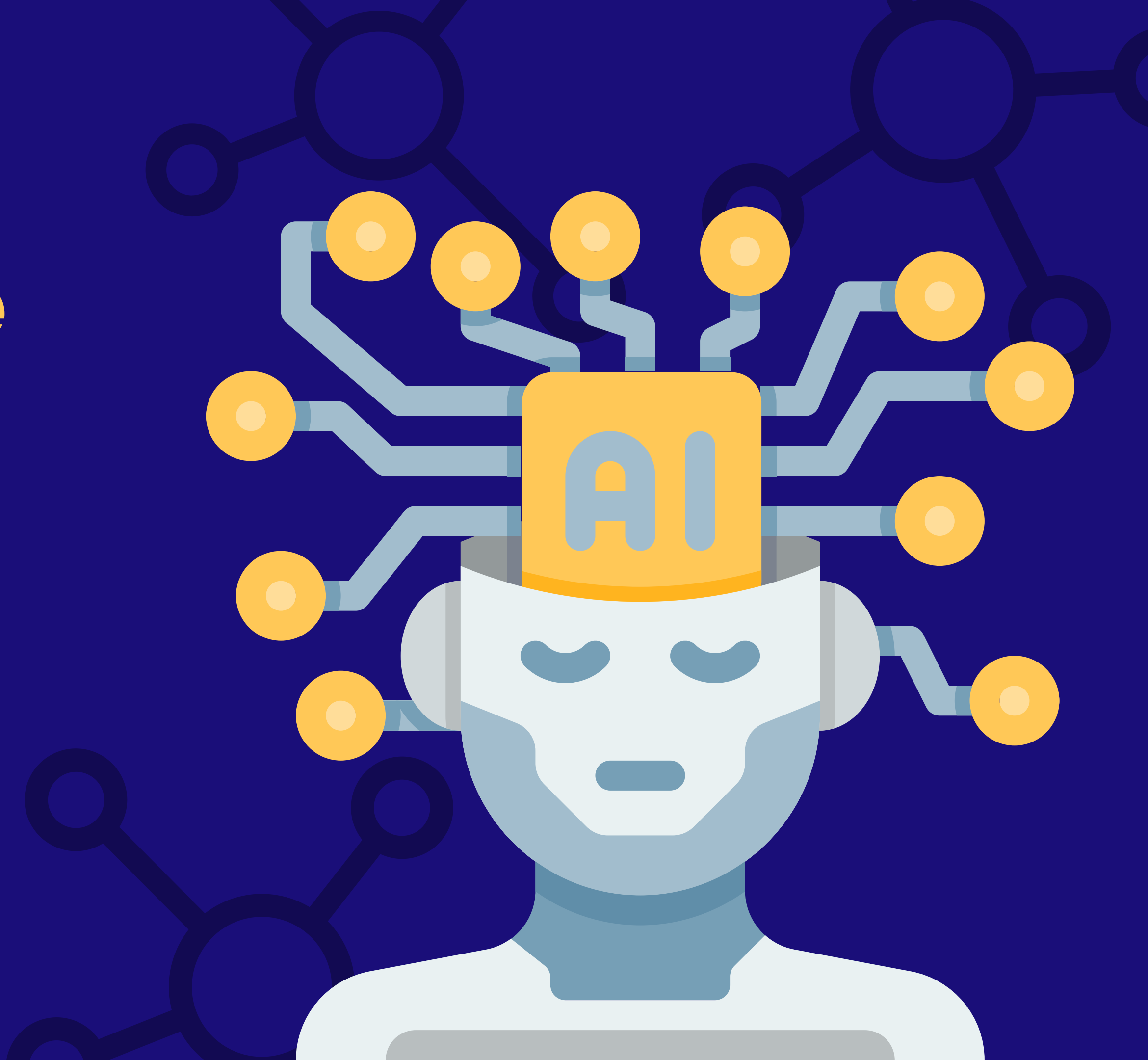
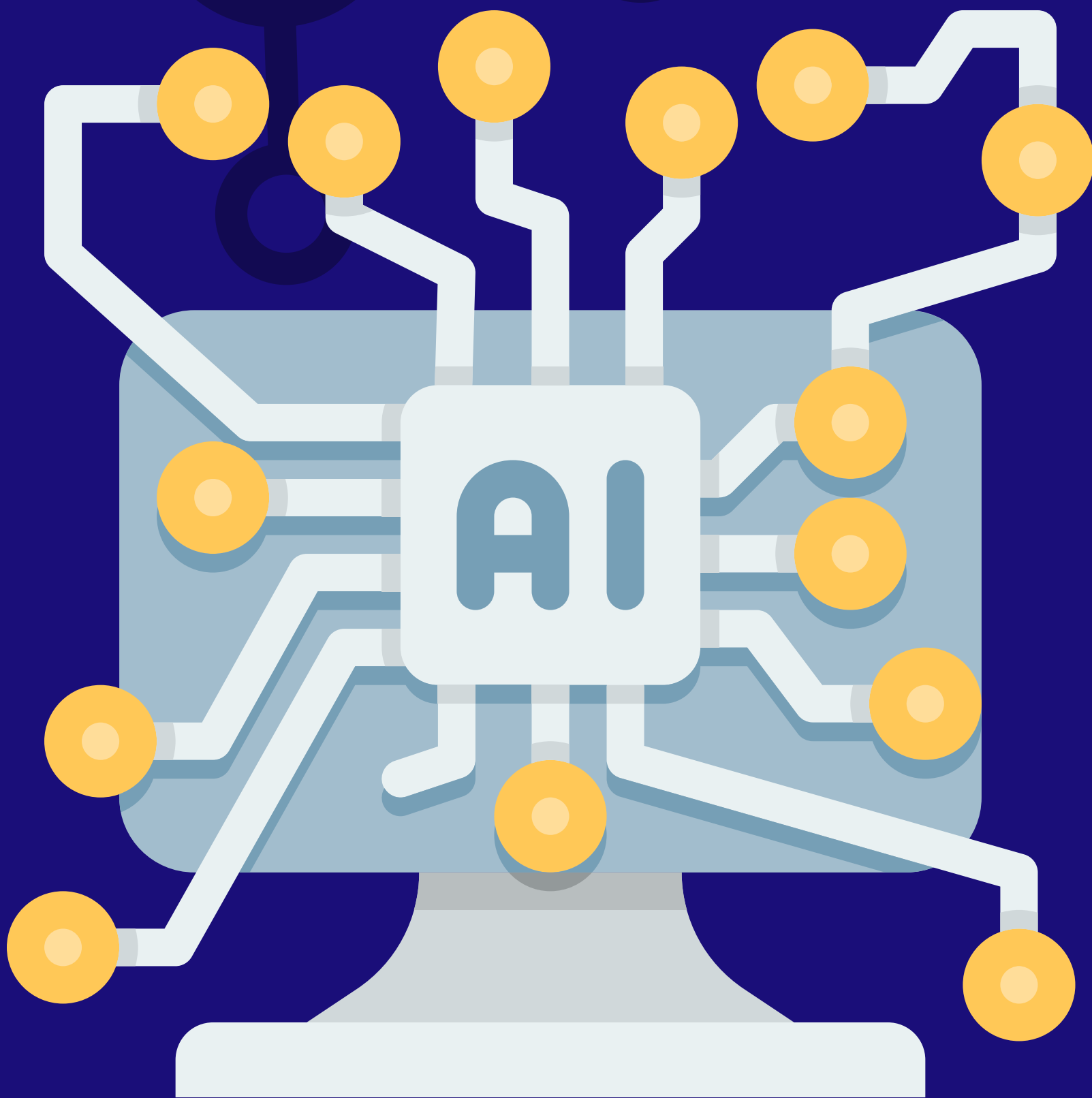


# MediaPipe

**Juan Medina**  
**Cristian Montañez**  
**Estephanie Perez**  
**Sergio Ruiz**



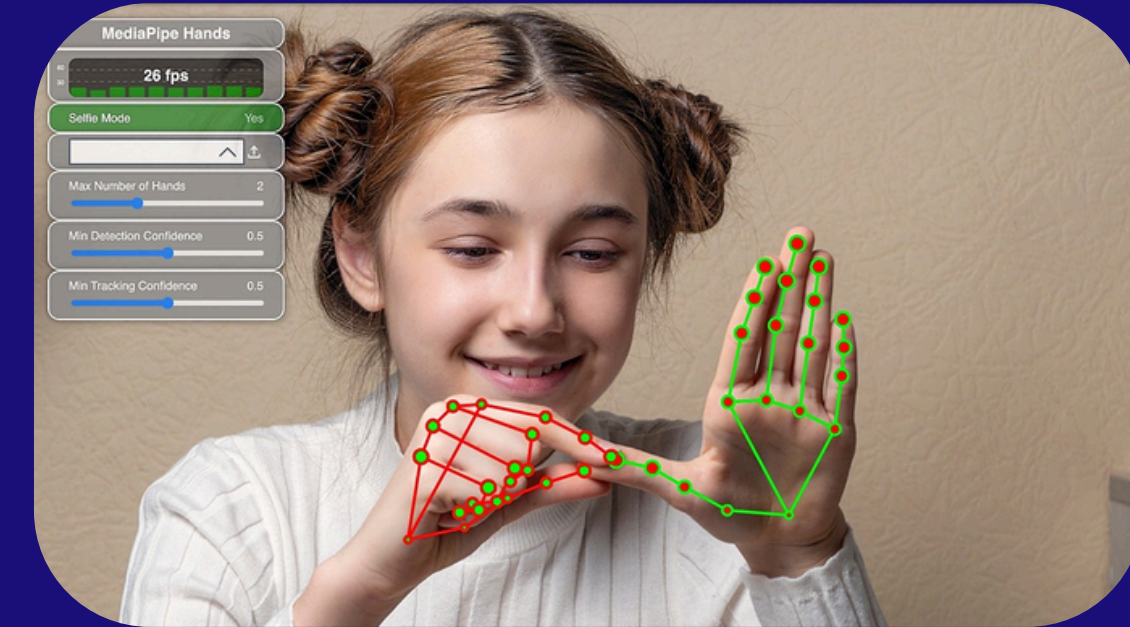


## Índice

- 01.** Sector en el que se Aplica
- 02.** Tecnologías Compatibles
- 03.** Librerías de Software
- 04.** Herramientas de Desarrollo
- 05.** Algoritmos

# 1. Sector en el que se Aplica

MediaPipe es un framework de Google que permite el procesamiento de medios en tiempo real con una alta precisión y bajo costo computacional. Se aplica en los siguientes sectores clave:



## Entretimiento

1

- Filtros en tiempo real
- Reconocimiento de gestos para videojuegos
- Producción audiovisual

## Salud

2

- Rehabilitación física
- Telemedicina
- Terapias de lenguaje y gestual

## Realidad Aumentada

3

- Aplicaciones móviles y wearables
- Espejos inteligentes y fitting rooms virtuales

## 2. Tecnologías Compatibles

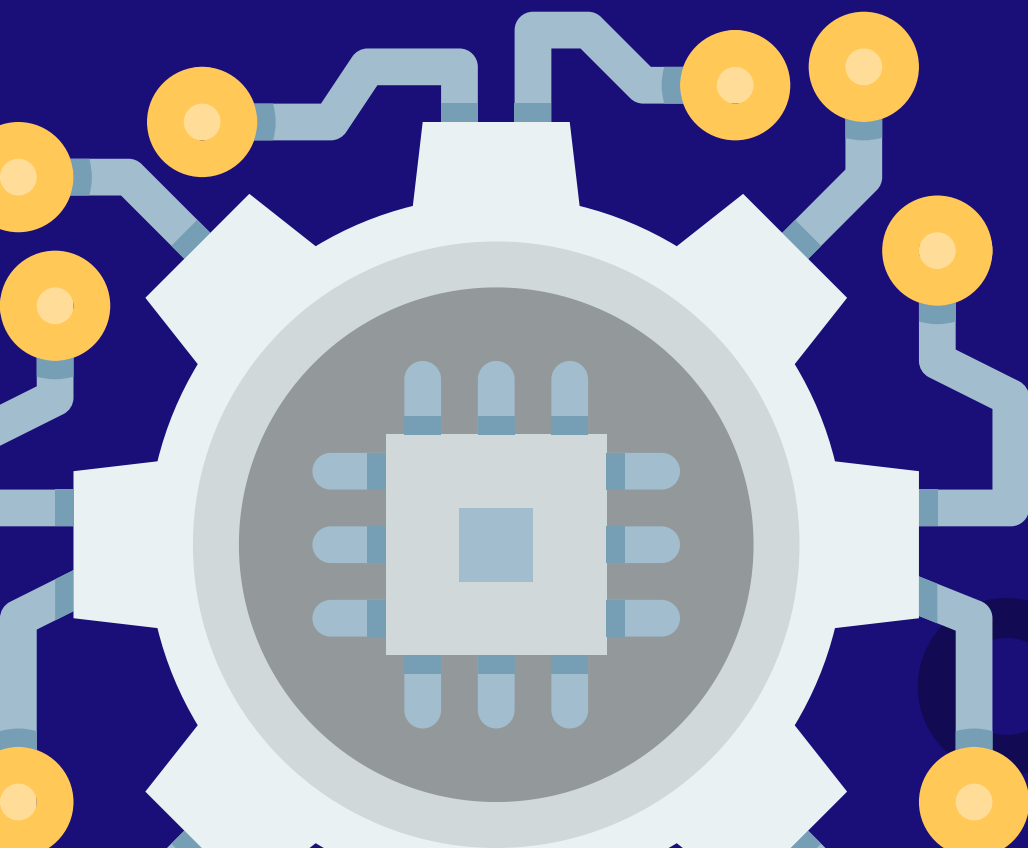
MediaPipe ofrece soporte multi-plataforma y está construido para integrarse fácilmente con varios lenguajes y tecnologías:

### LENGUAJES DE PROGRAMACIÓN

- Python: Ideal para prototipos rápidos, pruebas, y visualizaciones. Muy usado en investigación y desarrollo.
- C++: Usado para procesamiento de alto rendimiento y despliegue en producción.
- JavaScript (a través de WebAssembly): Soporte para la web, permitiendo correr modelos en navegadores.
- Java/Kotlin (Android): Implementación móvil optimizada para dispositivos Android.

### SISTEMAS COMPATIBLES

- Linux, Windows, macOS
- Android, iOS
- Web (mediante WebAssembly)



# 3. Librerías de Software Comúnmente Usadas

MEDIAPIPE SE INTEGRA CON OTRAS LIBRERÍAS PARA MEJORAR SUS CAPACIDADES:

## OpenCV

OpenCV se utiliza para tareas como la lectura de imágenes, conversión de espacios de color y aplicación de filtros antes de que MediaPipe procese los datos.

## NumPy y Pandas

Los resultados obtenidos de MediaPipe, como las coordenadas de puntos clave, pueden convertirse en arrays de NumPy o DataFrames de Pandas para facilitar su análisis.

## PyTorch

Es posible entrenar modelos en PyTorch y luego convertirlos a TensorFlow Lite para su uso con MediaPipe.

## Matplotlib

Permiten graficar las coordenadas de los puntos clave detectados por MediaPipe, facilitando la interpretación y análisis de los datos.



## 4. Herramientas útiles junto a Media Pipe

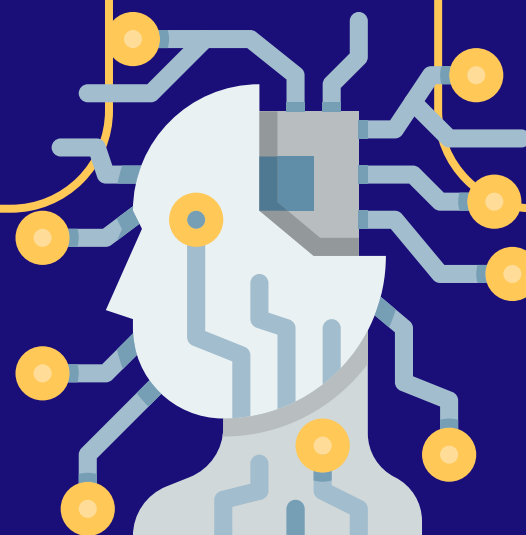
### IDEs

- **VSCode:** Ligero y con muchas extensiones útiles para Python, C++ y web.
- **PyCharm:** Excelente para trabajar con proyectos en Python.
- **Android Studio:** Recomendado para aplicaciones móviles.

&

### PLATAFORMAS DE DESPLIEGUE

- **Navegadores web:** A través de WebAssembly y soluciones JavaScript.
- **Aplicaciones móviles:** Desarrollos en Android e iOS para experiencias interactivas.





## 5. ALGORITMOS

1

### **Convolutional Neural Networks (CNNs)**

Detección de rostros, manos y cuerpo en imágenes.

2

### **BlazeFace / BlazePose / BlazePalm**

Modelos ligeros diseñados por Google para detección eficiente en tiempo real.

3

### **3D Landmark Regression**

Estimación precisa de puntos clave en 3D para rostro, manos y cuerpo.

## 5. ALGORITMOS

4

### **Kalman Filter / Smoothing Filters**

Suavizado del movimiento para evitar jitter en detección continua.

5

### **Bounding Box Tracking**

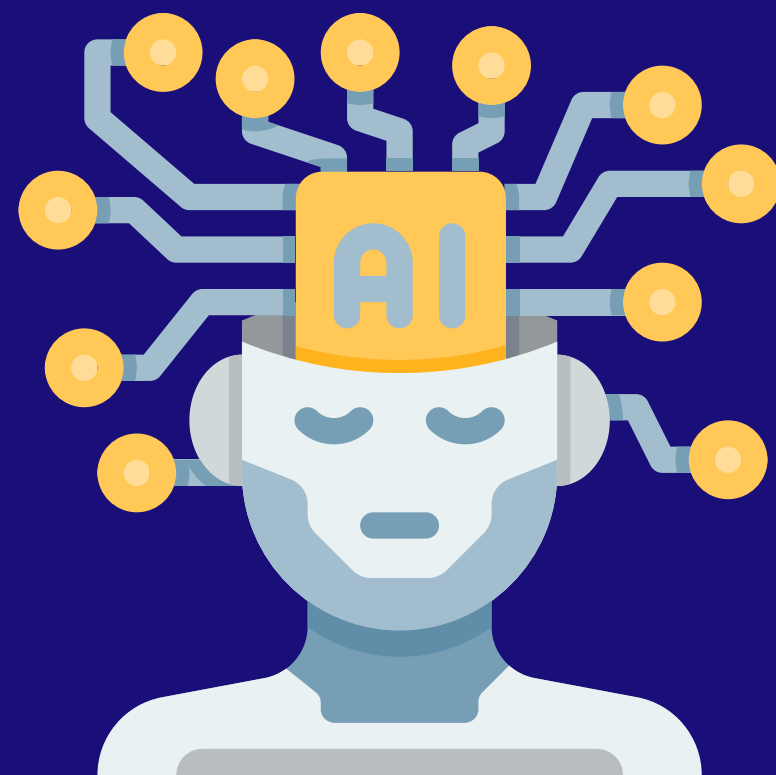
Seguimiento de regiones de interés (como la cara) entre frames.



# Referencias

**(1) A. Lugaresi, J. Smith, y otros, "MediaPipe: A Framework for Building Perception Pipelines," in Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Venice, Italy, 2019, pp. 1000–1007.**

**(2) M. Singh and R. Kumar, "Integration of OpenCV with MediaPipe for Real-Time Face and Pose Estimation," IEEE Access, vol. 9, pp. 45678–45686, 2021.**



**Gracias**