



UNIVERSIDAD MODELO

INGENIERÍA MECATRÓNICA

PROYECTOS VI
REPORTE DE ACTIVIDADES

PROFESOR:

FREDDY ANTONIO IX ANDRADE

REALIZADO POR:

JOSHUA EMMANUEL GÓNGORA ÁLVAREZ

HUGO ADRIEL ESTRELLA CANCINO

MAYO 2026

Contenido

1. Introducción	3
2. Objetivo general	3
3. Objetivos específicos.....	3
4. Alcance del proyecto	3
5. Arquitectura general de la solución.....	3
6. Entorno de desarrollo	4
6.1 Comandos sugeridos en Windows.....	4
7. Librerías utilizadas	4
8. Fundamento matemático del cálculo del ángulo	5
9. Desarrollo por fases.....	5
9.1 Fase 1: detección de pose en tiempo real	5
Funciones principales.....	6
9.2 Fase 2: medición del ángulo del codo	6
Decisiones de diseño implementadas	6
9.3 Fase 3: exportación del ángulo a Excel	7
Datos almacenados	7
9.4 Comparación resumida de fases	8
10. Controles del sistema	8
11. Validación del funcionamiento	8
12. Ventajas de la solución implementada	9
13. Conclusión.....	9

1. Introducción

Este documento presenta de forma ordenada el desarrollo del proyecto KineTech, un sistema en Python orientado a la detección de pose humana en tiempo real, la medición del ángulo del codo y el almacenamiento de los resultados en un archivo de Excel para su análisis posterior.

El proyecto se estructuró por fases para validar progresivamente cada componente: primero la captura y detección corporal, después el cálculo angular del codo (utilizado como referencia, pero se puede usar el que le convenga al fisioterapeuta), y finalmente la exportación de datos. Esta metodología facilita la depuración, mejora la comprensión del código y deja una base escalable para futuras aplicaciones en biomecánica, fisioterapia, rehabilitación o análisis de movimiento.

2. Objetivo general

Desarrollar un sistema de visión por computadora capaz de detectar puntos estratégicos del cuerpo humano en tiempo real, calcular el ángulo de movimientos a partir de los puntos de interés y registrar dichas mediciones en un archivo Excel.

3. Objetivos específicos

- Capturar video en vivo desde la cámara web del equipo.
- Detectar landmarks corporales utilizando MediaPipe Pose Landmarker.
- Visualizar el esqueleto corporal sobre la imagen capturada.
- Calcular el ángulo del codo con base en los puntos hombro, codo y muñeca.
- Mostrar el ángulo medido directamente sobre la interfaz de video.
- Guardar el tiempo y el ángulo del codo en un archivo Excel para análisis posterior.

4. Alcance del proyecto

La implementación actual se centra en el brazo derecho y mide únicamente el ángulo del codo. El sistema conserva la visualización general del cuerpo, excluye la cara para reducir ruido visual y guarda en Excel las mediciones de tiempo y ángulo. La solución está preparada para extenderse posteriormente a otras articulaciones, al brazo izquierdo o al análisis bilateral.

5. Arquitectura general de la solución

La solución funciona como una cadena de procesamiento compuesta por cinco bloques principales:

1. Adquisición de imagen mediante OpenCV.
2. Detección de pose con MediaPipe Pose Landmarker.
3. Conversión de landmarks normalizados a coordenadas de píxel.
4. Cálculo angular mediante geometría vectorial.
5. Visualización en pantalla y exportación de datos a Excel.

El modelo utilizado en la versión final fue `pose_landmarker_lite.task`, colocado en la misma carpeta que los scripts de Python para evitar errores de ruta.

Bloque	Función	Herramienta principal
Captura de video	Obtiene los frames desde la cámara web.	OpenCV
Detección corporal	Identifica landmarks del cuerpo humano.	MediaPipe
Procesamiento geométrico	Convierte puntos y calcula el ángulo del codo.	math / Python
Visualización	Dibuja el esqueleto y el valor del ángulo en pantalla.	OpenCV
Registro de datos	Guarda tiempo y ángulo en un archivo .xlsx.	openpyxl

6. Entorno de desarrollo

Se requiere utilizar Python 3.12 dentro de un entorno virtual, ya que ofrece buena compatibilidad con las librerías requeridas y reduce conflictos entre dependencias.

6.1 Comandos sugeridos en Windows

```
py -3.12 -m venv venv
```

```
venv\Scripts\activate
```

```
python -m pip install --upgrade pip
```

```
pip install mediapipe opencv-python numpy openpyxl
```

7. Librerías utilizadas

Librería	Uso principal	Fase
mediapipe	Detección de pose y obtención de landmarks corporales.	1, 2 y 3
opencv-python	Apertura de cámara, lectura de frames, dibujo y ventanas.	1, 2 y 3
numpy	Soporte numérico general para ampliaciones y procesamiento.	Preparación
openpyxl	Creación y edición de archivos Excel .xlsx.	3
pathlib	Manejo robusto de rutas del proyecto.	1, 2 y 3
math	Cálculo del ángulo mediante producto punto y trigonometría.	2 y 3
time	Control de FPS, marcas de	1, 2 y 3

Librería	Uso principal	Fase
	tiempo y muestreo para Excel.	

8. Fundamento matemático del cálculo del ángulo

Para medir el ángulo del codo se utilizan tres puntos: hombro, codo y muñeca. El vértice del ángulo es el codo, por lo que la referencia geométrica queda definida por la relación hombro-codo-muñeca.

- A = hombro
- B = codo
- C = muñeca

$$\theta = \cos^{-1} [(\mathbf{BA} \cdot \mathbf{BC}) / (|\mathbf{BA}| |\mathbf{BC}|)]$$

El programa forma dos vectores desde el codo: uno hacia el hombro y otro hacia la muñeca. Después calcula el producto punto, divide entre las magnitudes de ambos vectores y obtiene el ángulo mediante la función arccoseno. El resultado se expresa en grados.

Esta medición es relativa a los segmentos corporales, no a una referencia absoluta como el piso o la vertical del mundo. Por ello, el valor representa la flexión o extensión del codo según la posición detectada por la cámara.

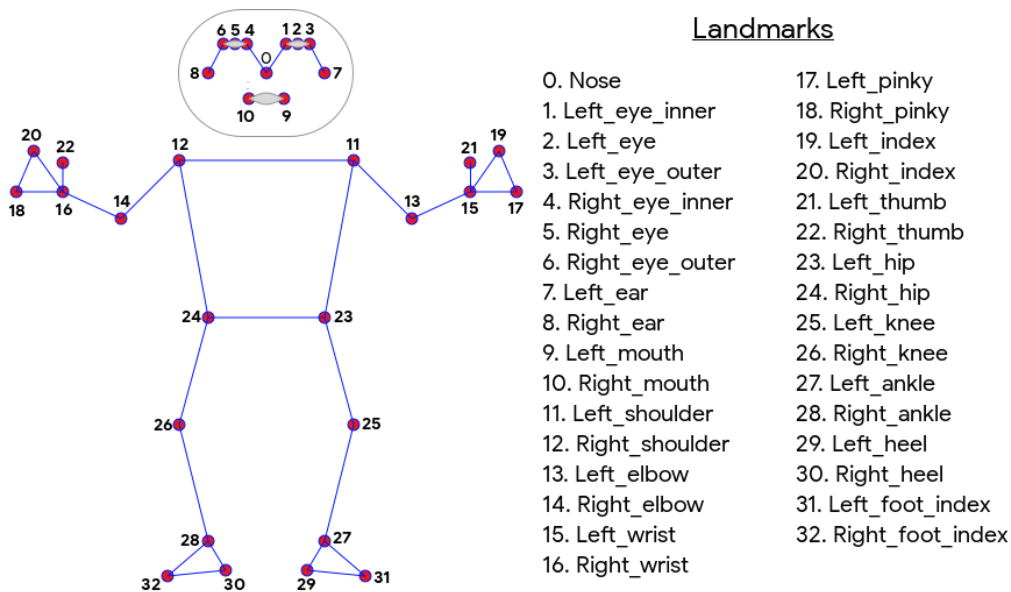


Figura 1. Puntos de referencia (“landmarks”)

9. Desarrollo por fases

El desarrollo se dividió en tres fases para validar cada parte del sistema antes de integrar nuevas funciones. A continuación se describe el propósito, funcionamiento y evidencia sugerida para cada una.

9.1 Fase 1: detección de pose en tiempo real

La Fase 1 valida la captura de video y la detección corporal. El programa abre la cámara, procesa cada frame, detecta landmarks del cuerpo y dibuja el esqueleto en tiempo real.

Funciones principales

- Abrir la cámara web.
- Convertir el frame de BGR a RGB para MediaPipe.
- Procesar el frame con Pose Landmarker.
- Dibujar landmarks y conexiones del cuerpo.
- Mostrar FPS en pantalla para observar el rendimiento.

Archivo asociado: fase1_pose.py

Comando de ejecución: python fase1_pose.py



Figura 2. Salida de la fase 1

9.2 Fase 2: medición del ángulo del codo

En la Fase 2 se calcula únicamente el ángulo del codo derecho. Para ello se toman como referencia los puntos del hombro, codo y muñeca derechos. Se mantienen visibles las líneas generales del cuerpo, pero se elimina la cara para simplificar el análisis visual.

Decisiones de diseño implementadas

- Conservar el esqueleto corporal general.

- Eliminar líneas y puntos faciales.
- Resaltar hombro, codo y muñeca del lado derecho.
- Calcular únicamente el ángulo del codo.
- Mostrar el valor del ángulo cerca de la articulación y en un texto general en pantalla.

Archivo asociado: fase2_codo.py

Comando de ejecución: python fase2_codo.py



Figura 3. Salida de la fase 2

9.3 Fase 3: exportación del ángulo a Excel

La Fase 3 toma la base funcional de la Fase 2 y agrega el registro de datos en un archivo Excel. Para evitar un crecimiento excesivo del archivo, el guardado se realiza a intervalos regulares y no en cada frame.

Datos almacenados

- Tiempo (s)
- Ángulo_codo (deg)

El archivo se genera como `angulos_codo.xlsx`, aunque puede adaptarse para incluir fecha y hora en el nombre del archivo.

Archivo asociado: fase3_excel_codo.py

Comando de ejecución: python fase3_excel_codo.py



Figura 4. Salidas de la fase 3

9.4 Comparación resumida de fases

Fase	Propósito	Salida principal	Evidencia recomendada
1	Validar cámara y detección corporal.	Video con esqueleto detectado.	Captura del cuerpo con landmarks.
2	Calcular el ángulo del codo derecho.	Ángulo visible en pantalla.	Captura con hombro, codo y muñeca resaltados.
3	Registrar mediciones para análisis posterior.	Archivo Excel con tiempo y ángulo.	Captura del Excel y del sistema en ejecución.

10. Controles del sistema

Durante la ejecución de los programas, la tecla q se utiliza para cerrar la ventana y finalizar la ejecución de forma controlada. Este cierre es importante especialmente en la Fase 3, ya que permite guardar correctamente el archivo Excel.

11. Validación del funcionamiento

Para considerar que el sistema funciona correctamente, deben cumplirse las siguientes condiciones:

- La cámara abre sin errores.
- El cuerpo es detectado en tiempo real.
- Las líneas del esqueleto se dibujan correctamente.

- Hombro, codo y muñeca se resaltan adecuadamente.
- El valor del ángulo del codo cambia conforme se mueve el brazo.
- En la Fase 3, el archivo Excel se crea y guarda filas con tiempo y ángulo.

12. Ventajas de la solución implementada

- Estructura modular por fases.
- Código entendible y escalable.
- Uso de herramientas modernas de visión por computadora.
- Base útil para fisioterapia, rehabilitación o biomecánica básica.
- Exportación directa de datos a Excel para análisis posterior.

13. Conclusión

El proyecto KineTech cumple con el objetivo de detectar pose humana en tiempo real, medir el ángulo del codo y exportar la información a Excel. La metodología por fases permitió validar cada etapa antes de avanzar a la siguiente, facilitando la depuración y consolidando una base técnica clara.

Además de ser funcional, el sistema queda preparado para crecer hacia aplicaciones más completas de análisis corporal, fisioterapia, rehabilitación o monitoreo de movimiento. La documentación organizada facilita que el proyecto pueda explicarse, replicarse y mejorarse en futuras etapas académicas o técnicas.