

# Hito 1 - IPD441

Integrantes: Ignacio Barrera 2019094093-6  
Francisca Soto 201704175-7

## Contexto:

El objetivo de este proyecto es implementar un algoritmo de pose, siendo nuestro enfoque personal realizar un seguimiento al pie de un jugador, buscamos, por una parte, que el modelo sea estable, osea que siga de forma acertada y no pierda la posición del pie, por otra parte también debemos buscar un modelo, o crear, uno que tenga al menos 2 keypoints para la pose del pie, idealmente, en el talón y punta del pie.

## Métricas:

A continuación, listamos las métricas que usaremos para evaluar nuestros algoritmos propuestos en este proyecto:

- Error de Punto Clave Promedio (PCK - Percentage of Correct Keypoints):  
Es una métrica de precisión que mide si el keypoint (lo que predijo el algoritmo) y la articulación verdadera (ground truth) están dentro de un cierto umbral de distancia.
- Error Medio de Posición (MPE - Mean Position Error): métrica utilizada para medir la precisión de un modelo predictivo en relación de las predicciones y los valores reales en el conjunto de datos. Se calcula como el promedio de los errores porcentuales individuales para cada punto de datos del dataset.
- AP: área bajo la curva precision-recall para clase específica, teniendo un valor más alto de AP implica un mejor rendimiento en la detección de clase.
- mAP: media de AP, es una métrica que proporciona una medida general del rendimiento del modelo en la detección de objetos.
- Precisión: proporción de puntos clave correctamente detectados sobre el total de puntos clave detectados por el modelo.
- Recall: proporción de puntos claves correctamente detectados sobre el total de puntos claves reales en la imagen.
- F1-Score: combinación de precisión y recall.
- Curva ROC: nos servirá para analizar el rendimiento del modelo y así podemos obtener AUC.

## Dataset:

Planteamos utilizar modelos pre-entrenados, por lo que planteamos la opción de crear un dataset nuestro con fines de validación.

De igual forma, para modelos los cuales tengan pocos keypoints en el pie, evaluaremos la opción de realizar un entrenamiento personalizado de estos modelos con datasets que

contengan multiples keypoints, esto podría implicar nosotros crear un dataset, ya que al menos en un primer acercamiento no encontramos un dataset open-source que cumpla nuestros requerimientos.

Ahora. se listaran algunos datasets de relevancia para la evaluación de algoritmos de pose en humanos:

- COCO: <https://cocodataset.org/#keypoints-2020>



- OCHuman: <https://github.com/liruiflong940607/OCHumanApi>



## Modelos:

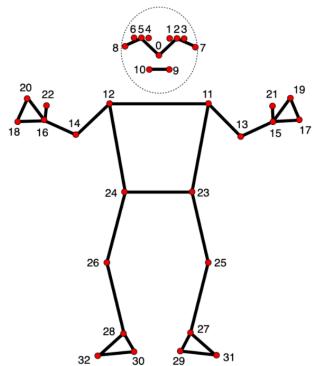
Para la elección de modelos, nos basamos en los keypoints que entregan y su evaluación de métricas, consideraremos evaluar diferentes modelos:

- YoloVX (Ultralytics) [<https://docs.ultralytics.com/es/tasks/pose/>]: Es un modelo que realiza múltiples tareas, es ampliamente conocido por su rendimiento y constante actualización, su evaluación se muestra en la siguiente tabla:

Modelo	tamaño (píxeles)	mAPpose 50-95	mAPpose 50
YOLOv8n- pose	640	50.4	80.1
YOLOv8s- pose	640	60.0	86.2
YOLOv8m- pose	640	65.0	88.8
YOLOv8l- pose	640	67.6	90.0
YOLOv8x- pose	640	69.2	90.2
YOLOv8x- pose-p6	1280	71.6	91.2

Notemos que ofrece diferentes modelos, los cuales obtienen valores crecientemente mejores respecto a los demás a cambio de mayor costo computacional, estos valores se obtuvieron utilizando el dataset [COCO Puntos clave val2017](#). Dentro de las tareas que realiza encontramos detección y seguimiento de objetos, segmentación de instancias y estimación de pose.

- Media Pipe (Google) [\[https://developers.google.com/mediapipe/solutions/vision/pose\\_landmarker#get\\_started\]](https://developers.google.com/mediapipe/solutions/vision/pose_landmarker#get_started): Modelo que provee múltiples características que incluyen detección de rostros y gestos, seguimiento de manos y detección de objetos móviles. Posee 3 keypoints para los pies sin la necesidad de re-entrenamiento, lo cual es bastante bueno para nuestro objetivo:



datasets enfocados en la detección de pose de personas. (#1 en COCO), esto se puede revisar en

<https://paperswithcode.com/paper/vitpose-simple-vision-transformer-baselines>

Model	Dataset	Resolution	AP	AR
ViTPose-B	COCO+AIC+MPII	256x192	77.1	82.2
ViTPose-L	COCO+AIC+MPII	256x192	78.7	83.8
ViTPose-H	COCO+AIC+MPII	256x192	79.5	84.5
ViTPose-G	COCO+AIC+MPII	576x432	81.0	85.6

Notemos que al igual que Yolo ofrece diferentes modelos, los cuales obtienen valores crecientemente mejores respecto a los demás a cambio de mayor costo computacional, valores obtenidos para MS COCO Keypoint test-dev set.

Prueba con modelo pre-entrenado desde Hugging Face(<https://huggingface.co/spaces/Gradio-Blocks/ViTpose> )