

# Sistema de Análisis y Predicción de Poses

Facultad de Ingeniería, Diseño y Ciencias Aplicadas Departamento  
de Computación y Sistemas Inteligentes Ingeniería de Sistemas

---

**Docentes:** Uram Sosa Aguirre, Milton Sarria Paja

**Asignatura:** Inteligencia Artificial I

**Institución:** Universidad Icesi

---

**Integrantes:**

Juan Sebastian Gonzalez, Juan Jose Angarita, Jeison Lasprilla

**Día de Entrega:**

25/Noviembre/2024

# Índice:

<b>Índice:</b>	<b>2</b>
<b>Abstracto:</b>	<b>4</b>
<b>Introducción:</b>	<b>4</b>
Contexto:	4
Descripción del Problema:	5
¿Por qué es interesante?	5
<b>Teoría:</b>	<b>5</b>
Estimación de Pose:	5
Landmarks detectados:	6
Modelos de Aprendizaje Automático:	6
Random Forest:	6
¿Qué es Random Forest?	6
Características Principales:	6
Ventajas para la Clasificación de Poses:	6
Limitaciones:	7
XG Boost:	7
¿Qué es XGBoost?	7
Características Principales:	7
Ventajas para la Clasificación de Poses:	7
Limitaciones:	7
Comparación entre Random Forest y XGBoost:	8
Justificación de la Selección:	8
<b>Metodología:</b>	<b>9</b>
Colección de Datos:	9
Herramientas y Configuración:	9
Flujo del Sistema:	9
Volumen de Datos:	9
Preparación de Datos:	10
Preprocesamiento:	10
División del Dataset:	10
Entrenamiento del Modelo:	11
Optuna:	12
Evaluación del Modelo:	12
Métricas de Evaluación:	12
Importancia de Características:	13
Visualizaciones de Resultados:	13
<b>Resultados:</b>	<b>14</b>
Desempeño del Modelo:	14
<b>Análisis de Resultados:</b>	<b>15</b>

Observaciones.....	15
1. Métricas de rendimiento:.....	15
2. Importancia de características:.....	15
3. Matriz de confusión:.....	16
Generalización del Modelo.....	16
Conclusiones y Trabajo Futuro.....	16
¿Qué hiciste?.....	16
¿Qué aprendiste?.....	17
¿Qué podría mejorarse?.....	17
<b>Referencias Bibliográficas.....</b>	<b>18</b>

# Abstracto

El reconocimiento y análisis de actividades humanas ha sido un área de interés crítico en la inteligencia artificial, debido a sus aplicaciones prácticas en múltiples dominios como la seguridad, la salud y el deporte. Este proyecto tiene como objetivo desarrollar un sistema de análisis y predicción de poses humanas, capaz de procesar datos en tiempo real y clasificar actividades con alta precisión.

El sistema combina técnicas de visión por computadora, como la detección de landmarks corporales con MediaPipe, y modelos de aprendizaje automático como Random Forest y XGBoost. Se emplearon técnicas avanzadas de optimización de hiperparámetros, incluyendo Grid Search y Optuna, para garantizar el máximo rendimiento de los modelos.

Se recolectaron datos de múltiples actividades humanas y se evaluaron los modelos con métricas como precisión, recall y matrices de confusión. Los resultados mostraron una precisión superior al 97%, superando estándares establecidos en investigaciones previas. Este documento presenta en detalle la metodología, resultados y aprendizajes obtenidos, junto con recomendaciones para trabajos futuros.

## Introducción

### Contexto

El reconocimiento de actividades humanas ha ganado relevancia debido a sus aplicaciones prácticas en áreas como:

- **Vigilancia y seguridad pública:** Permite detectar comportamientos sospechosos o potencialmente peligrosos.
- **Rehabilitación médica:** Facilita el monitoreo de pacientes durante sesiones de fisioterapia.
- **Deportes y fitness:** Optimiza el rendimiento atlético y previene lesiones mediante el análisis de movimientos.
- **Entretenimiento e interacción humano-computadora:** Mejora experiencias inmersivas en realidad virtual y videojuegos.

A medida que el hardware se vuelve más accesible y las técnicas de aprendizaje automático más sofisticadas, es posible implementar sistemas que procesen datos en tiempo real sin requerir grandes recursos computacionales. Este proyecto busca aprovechar esta tendencia para diseñar un sistema práctico, escalable y adaptable.



## Descripción del Problema

El reto principal radica en construir un sistema capaz de detectar actividades humanas en tiempo real, basado en datos capturados por una cámara web. Este sistema debe procesar coordenadas de puntos clave del cuerpo, derivar ángulos importantes y usar esta información para entrenar y evaluar modelos de aprendizaje automático que logren alta precisión en la clasificación de actividades.

El análisis en tiempo real de actividades humanas enfrenta retos significativos:

1. **Complejidad del movimiento humano:** Cada persona tiene patrones únicos, y las actividades pueden ser similares en algunas poses intermedias.
2. **Recolección y preprocesamiento de datos:** Los datos de movimiento son ruidosos y pueden estar sujetos a errores de medición.
3. **Optimización de modelos:** Encontrar configuraciones de hiperparámetros que equilibren precisión y eficiencia computacional.

## ¿Por qué es interesante?

La solución tiene implicaciones directas en áreas críticas como la seguridad y la salud. Por ejemplo, la vigilancia de comportamientos en áreas públicas podría prevenir incidentes. En la rehabilitación médica, el monitoreo preciso de actividades físicas puede mejorar los resultados terapéuticos. Además, este proyecto explora cómo combinar metodologías avanzadas para abordar un problema práctico y desafiante.

Este proyecto se destaca por combinar técnicas modernas de estimación de poses y aprendizaje automático con un enfoque práctico y accesible. Su diseño permite implementaciones en entornos reales, ofreciendo soluciones inmediatas para problemas del mundo real.

## Teoría

### Estimación de Pose

La estimación de poses es una técnica de visión por computadora que identifica puntos clave del cuerpo humano en imágenes o videos. Utilizamos MediaPipe, una biblioteca que ofrece detección precisa y eficiente de landmarks corporales en tiempo real. Los landmarks clave incluyen hombros, caderas, rodillas y tobillos, que se usan para calcular métricas como ángulos de articulaciones.

MediaPipe fue seleccionada como la herramienta principal debido a:

- **Detección en tiempo real:** Puede identificar 33 puntos clave del cuerpo en tiempo real, incluso en hardware limitado.
- **Flexibilidad:** Es adaptable a diferentes ángulos de cámara y condiciones de iluminación.
- **Robustez:** Genera datos fiables para alimentar modelos de aprendizaje automático.

## Landmarks detectados

1. **Parte Superior:** Cabeza, hombros, codos y muñecas.
2. **Parte Media:** Tronco y caderas.
3. **Parte Inferior:** Rodillas y tobillos.

## Modelos de Aprendizaje Automático

El reconocimiento de actividades humanas basado en coordenadas de poses requiere modelos que puedan manejar datos no lineales, de alta dimensionalidad y susceptibles al ruido. Para este proyecto, se seleccionaron dos modelos principales: **Random Forest** y **XGBoost**. Cada uno fue elegido debido a sus fortalezas específicas y su desempeño probado en tareas de clasificación complejas.

### Random Forest

#### ¿Qué es Random Forest?

Random Forest es un modelo de aprendizaje supervisado que opera creando múltiples árboles de decisión durante el entrenamiento y luego agregando sus predicciones (por ejemplo, mediante votación para clasificación o promedio para regresión). Fue desarrollado para superar las limitaciones de los árboles de decisión individuales, como el sobreajuste y la falta de robustez frente a pequeñas variaciones en los datos.

#### Características Principales

1. **Ensamble de Árboles:** Random Forest construye muchos árboles de decisión independientes durante el entrenamiento. Cada árbol se entrena en un subconjunto diferente del dataset (con reemplazo, conocido como bootstrap sampling), aumentando la robustez del modelo.
2. **Selección Aleatoria de Características:** Durante la construcción de cada árbol, Random Forest selecciona aleatoriamente un subconjunto de características para evaluar en cada división. Esto introduce diversidad en los árboles y reduce la probabilidad de que se ajusten demasiado a características particulares.
3. **Promedio/Votación:** La agregación de predicciones garantiza que el modelo sea menos propenso a errores causados por árboles individuales mal ajustados.

#### Ventajas para la Clasificación de Poses

- **Resistencia al Sobreajuste:** Gracias a su capacidad de promediar múltiples árboles, Random Forest maneja bien datasets ruidosos como los generados por estimación de poses.
- **Robustez frente a Dimensionalidad Alta:** Al seleccionar subconjuntos de características, puede trabajar eficientemente con las numerosas coordenadas y ángulos calculados.
- **Importancia de Características:** Proporciona una medida cuantitativa de qué coordenadas o ángulos son más relevantes para la clasificación, ayudando en el análisis interpretativo.

## Limitaciones

- **Eficiencia Computacional:** Random Forest puede ser computacionalmente costoso cuando se utilizan muchos árboles o datasets muy grandes.
- **Interpretabilidad Limitada:** Aunque es más interpretable que algunos modelos complejos, la agregación de múltiples árboles puede dificultar la explicación de predicciones específicas.

## XG Boost

### ¿Qué es XGBoost?

XGBoost (Extreme Gradient Boosting) es un algoritmo de boosting basado en árboles de decisión. El boosting es una técnica que combina predicciones de múltiples modelos base (en este caso, árboles) de manera iterativa, corrigiendo los errores de los modelos anteriores en cada paso. XGBoost mejora el boosting tradicional mediante optimizaciones que lo hacen más rápido y efectivo.

### Características Principales

1. **Boosting Gradual:** XGBoost ajusta secuencialmente nuevos árboles para corregir los errores residuales de los árboles anteriores, minimizando una función de pérdida específica (como el logaritmo de probabilidad para clasificación).
2. **Regularización Avanzada:**
  - **L1 (Lasso):** Penaliza características irrelevantes, haciendo el modelo más compacto.
  - **L2 (Ridge):** Reduce el impacto de características con pesos extremos, aumentando la estabilidad del modelo.
3. **Manejo de Valores Perdidos:** XGBoost incluye estrategias internas para manejar valores faltantes en los datos, asignando divisiones "predeterminadas" en sus árboles.
4. **Optimización Paralela:**
  - Aprovecha el hardware moderno para realizar cálculos en paralelo, acelerando significativamente el entrenamiento.
  - Implementa particionamiento inteligente de datos para maximizar la eficiencia computacional.
5. **Función de Objetivo Personalizable:** Permite utilizar métricas específicas del problema, como log-loss para clasificación multinomial o AUC para evaluar desempeño en clasificación binaria.

### Ventajas para la Clasificación de Poses

- **Capacidad para manejar Datos Complejos:** Gracias al boosting iterativo, XGBoost es excelente en problemas donde las relaciones entre características no son triviales, como las interacciones entre coordenadas y ángulos.
- **Control del Sobreajuste:** Las técnicas de regularización garantizan que el modelo sea robusto incluso en datasets pequeños o con ruido.
- **Eficiencia Computacional:** A pesar de ser un modelo avanzado, su implementación optimizada permite entrenar rápidamente incluso en hardware básico.
- **Predicciones de Alta Precisión:** La capacidad de corregir errores iterativamente resulta en un rendimiento consistentemente alto.

## Limitaciones

- **Mayor Complejidad:** Comparado con Random Forest, XGBoost tiene más hiperparámetros, lo que puede requerir un ajuste más cuidadoso.
- **Requiere Más Memoria:** Los cálculos paralelos y el boosting iterativo pueden consumir más recursos que otros modelos.

Comparación entre Random Forest y XGBoost

Aspecto	Random Forest	XGBoost
Técnica Principal	Ensamble de árboles con votación/promedio.	Boosting iterativo para corregir errores.
Manejo de Sobreajuste	Bueno, gracias al promedio.	Excelente, con regularización avanzada.
Eficiencia Computacional	Moderado, aumenta con más árboles.	Muy eficiente gracias a la paralelización.
Interpretabilidad	Proporciona importancia de características.	Más difícil de interpretar debido al boosting.
Precisión	Alta.	Generalmente superior en tareas complejas.

Justificación de la Selección

1. **Random Forest** se eligió como base debido a su simplicidad, robustez y facilidad de implementación. Es ideal para tareas en las que la interpretabilidad y la resistencia al ruido son esenciales.
2. **XGBoost** fue seleccionado para abordar los casos más desafiantes de clasificación, donde las relaciones entre características son complejas. Su capacidad para optimizar funciones de pérdida específicas lo convierte en una herramienta poderosa para maximizar la precisión.

La combinación de ambos modelos proporciona un enfoque balanceado, aprovechando las fortalezas de cada uno para lograr un sistema altamente preciso y robusto.



# Metodología

## Colección de Datos

### Herramientas y Configuración

Para la recolección de datos, se desarrolló una **interfaz gráfica** utilizando **Tkinter**, una biblioteca de Python que permite crear GUIs (interfaces gráficas de usuario) de manera sencilla y eficiente. Esta elección se basó en varios factores:

- **Simplicidad de Implementación:** Tkinter está integrado en la biblioteca estándar de Python, lo que elimina la necesidad de instalar herramientas adicionales.
- **Facilidad de Escalabilidad:** Su diseño modular permite agregar nuevas funcionalidades, como controles para preprocesar los datos o visualizar los landmarks en tiempo real.
- **Compatibilidad Multiplataforma:** Tkinter es compatible con Windows, macOS y Linux, asegurando que la herramienta sea accesible desde cualquier sistema operativo.

### Flujo del Sistema

1. **Captura de Video:** Se utilizó una cámara web para capturar imágenes en tiempo real. Estas imágenes fueron procesadas por MediaPipe para detectar los **landmarks** del cuerpo humano.
2. **Extracción de Coordenadas:** MediaPipe generó 33 puntos clave en coordenadas 3D (x, y, z), incluyendo hombros, caderas, rodillas, tobillos y muñecas.
3. **Etiquetado Manual:** La interfaz gráfica permitía al usuario seleccionar la actividad correspondiente (por ejemplo, "walking towards") antes de comenzar la grabación.
4. **Almacenamiento:** Los datos recolectados (coordenadas y etiquetas) se almacenaron en formato CSV, facilitando el acceso y procesamiento posterior.

### Volumen de Datos

Se recolectaron datos de **5 individuos** realizando las siguientes actividades:

- **Walking Towards:** Caminar hacia la cámara.
- **Walking Away:** Caminar alejándose.
- **Sitting:** Sentarse.
- **Standing:** Permanecer de pie.
- **Turning:** Girar.

Cada actividad fue repetida varias veces para un total de **1000 registros**, asegurando una representación balanceada entre actividades. Este enfoque permitió capturar variaciones individuales y evitar sesgos en los datos.

## Preparación de Datos

### Preprocesamiento

Los datos brutos recolectados incluían coordenadas de landmarks y etiquetas de actividad. Para hacerlos adecuados para el modelo, se aplicaron varias etapas de preprocesamiento:

1. **Normalización:** Las coordenadas (x, y, z) de cada landmark se normalizaron entre 0 y 1. Esto se realizó dividiendo las coordenadas por el tamaño del bounding box corporal detectado, reduciendo la influencia del tamaño o la posición física de los individuos.

**Razón:** La normalización asegura que las diferencias entre individuos (por ejemplo, altura o distancia de la cámara) no afecten el rendimiento del modelo.

Fórmula aplicada:

$$\text{Coordenada Normalizada} = \frac{\text{Coordenada original} - \text{Mínimo}}{\text{Máximo} - \text{Mínimo}}$$

2. **Cálculo de Ángulos Clave:** Se calcularon ángulos entre articulaciones clave utilizando fórmulas trigonométricas. Por ejemplo:
  - **Ángulo de la Rodilla Izquierda:** Formado por las coordenadas de la cadera, rodilla y tobillo izquierdos.
  - **Inclinación del Tronco:** Basado en la línea formada entre los hombros y las caderas.
3. **Razón:** Los ángulos capturan relaciones entre articulaciones que no son evidentes en coordenadas absolutas, mejorando la capacidad del modelo para distinguir actividades similares.

Ejemplo de cálculo de ángulo:

$$\text{Angulo} = \arccos \left( \frac{\vec{A} * \vec{B}}{\|\vec{A}\| * \|\vec{B}\|} \right)$$

4. **Manejo de Valores Perdidos:** Aunque MediaPipe es robusto, se implementaron técnicas para manejar posibles valores faltantes (por ejemplo, si un landmark estaba parcialmente ocluido):
  - Interpolación lineal entre frames consecutivos.
  - Exclusión de registros con landmarks críticos ausentes.

### División del Dataset

El dataset completo de 1000 registros se dividió en conjuntos de entrenamiento y prueba:

- **Entrenamiento (80%):** Utilizado para ajustar los parámetros de los modelos.
- **Pruebas (20%):** Evaluado únicamente después del entrenamiento para medir la capacidad de generalización.

### Razón de la División:

- **80-20:** Es una práctica estándar en aprendizaje automático que ofrece suficiente data para el entrenamiento mientras se reserva una parte significativa para pruebas independientes.



Claro, aquí está una versión ampliada y enriquecida de la sección de **Metodología, Recolección de Datos y Preparación de Datos**, incorporando detalles adicionales como la decisión de usar Tkinter y un análisis más profundo del preprocesamiento y división de datos:

---

## Entrenamiento del Modelo

El entrenamiento de modelos es un paso crítico en el desarrollo del sistema, ya que la precisión del modelo determina su capacidad para clasificar actividades humanas de manera efectiva. Para este proyecto, se implementaron dos enfoques avanzados de optimización de hiper parámetros: **Grid Search** y **Optuna**, ambos utilizados para ajustar modelos de aprendizaje automático como **Random Forest** y **XGBoost**.

**Grid Search** es un método de búsqueda exhaustiva en un espacio predefinido de hiper parámetros. Para cada combinación de parámetros, el modelo es evaluado utilizando validación cruzada, y se selecciona la configuración con mejor desempeño.

- **Parámetros Evaluados:**
  - Random Forest:
    - **n\_estimators** (número de árboles): [100, 200, 300]
    - **max\_depth** (profundidad máxima del árbol): [10, 20, 30, None]
    - **min\_samples\_split** (mínimo de muestras por división): [2, 5, 10]
    - **min\_samples\_leaf** (mínimo de muestras por hoja): [1, 2, 4]
    - **max\_features** (número máximo de características): ['sqrt', 'log2']
  - XGBoost:
    - **max\_depth**: [3, 5, 7]
    - **learning\_rate** (tasa de aprendizaje): [0.01, 0.1, 0.3]
    - **n\_estimators**: [100, 200, 300]
    - **min\_child\_weight**: [1, 3, 5]
    - **gamma** (regularización): [0, 0.1, 0.2]
    - **subsample**: [0.8, 0.9, 1.0]
    - **colsample\_bytree** (fracción de características por árbol): [0.8, 0.9, 1.0]
- **Ventajas:**
  - Cobertura completa del espacio de parámetros predefinido.
  - Garantiza que se prueben todas las combinaciones posibles.

- **Limitaciones:**
  - Requiere mucho tiempo y recursos computacionales, especialmente cuando el espacio de búsqueda es grande.

## Optuna

**Optuna** es un framework para la optimización eficiente de hiperparámetros mediante técnicas bayesianas. A diferencia de Grid Search, explora dinámicamente el espacio de parámetros, priorizando configuraciones prometedoras basadas en pruebas anteriores.

- **Ventajas:**
  - **Eficiencia Computacional:** Explora espacios de búsqueda más grandes en menos tiempo.
  - **Flexibilidad:** Permite definir funciones de objetivo personalizadas para ajustar parámetros específicos.
  - **Adaptabilidad:** Identifica de manera eficiente combinaciones óptimas en espacios no lineales.
- **Proceso:**
  - Define una función de objetivo que evalúa el modelo usando un conjunto de hiperparámetros.
  - Realiza búsquedas iterativas, ajustando la exploración basada en el desempeño previo.
- **Resultados con Optuna:**
  - Random Forest y XGBoost obtuvieron configuraciones con mayor precisión en comparación con Grid Search, debido a la capacidad de Optuna para explorar configuraciones menos intuitivas pero efectivas.

## Evaluación del Modelo

Una vez entrenados, los modelos fueron evaluados utilizando diversas métricas para garantizar que sean robustos y generalicen bien en datos no vistos.

### Métricas de Evaluación

1. **Precisión Global:**
  - **Definición:** Relación de predicciones correctas entre el total de predicciones.
  - **Cálculo:**

$$\text{Precisión} = \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}}$$

- Ambos modelos lograron una precisión alta, siendo XGBoost el mejor con un 97%.

## Matriz de Confusión:

- **¿Qué es?**
  - Una herramienta para analizar la distribución de aciertos y errores por categoría.
  - Permite identificar actividades donde el modelo tiene más dificultad.
- **Interpretación:**
  - **Diagonal Principal:** Representa las predicciones correctas.
  - **Fuera de la Diagonal:** Indica confusiones entre actividades.
- **Resultados:**
  - Actividades estáticas como "standing" fueron clasificadas con mayor precisión.
  - Actividades dinámicas como "turning" presentaron confusiones menores con "walking towards".

## Informe de Clasificación:

- Incluye métricas como:
  - **Precisión:** Proporción de predicciones correctas para cada clase.
  - **Recall:** Proporción de instancias de la clase correctamente identificadas.
  - **F1-Score:** Media armónica entre precisión y recall, útil cuando hay clases desbalanceadas.

## Importancia de Características

Los modelos también fueron evaluados en términos de la importancia de las características utilizadas para realizar predicciones.

- **Random Forest:**
  - Proporciona una medida clara de importancia para cada característica basada en su contribución a las divisiones de los árboles.
  - **Resultados:**
    - Los landmarks relacionados con las caderas y rodillas fueron los más relevantes para actividades como "walking".
    - Los ángulos del tronco fueron críticos para distinguir entre "sitting" y "standing".
- **XGBoost:**
  - Utiliza métricas internas como "gain" y "cover" para evaluar la importancia de las características.
  - **Resultados:**
    - Los ángulos calculados mostraron ser más efectivos para capturar relaciones espaciales no triviales entre articulaciones.
    - Landmarks en el plano horizontal (coordenadas x) contribuyeron significativamente a actividades dinámicas.

## Visualizaciones de Resultados

1. **Gráfico de Importancia de Características:**
  - Un gráfico de barras mostró las características más relevantes. Los landmarks como las caderas, rodillas y hombros, junto con los ángulos calculados, dominaron las primeras posiciones.

## 2. Visualización de la Matriz de Confusión:

- Un mapa de calor resaltó la precisión en cada actividad y mostró áreas donde hubo confusiones menores.

La combinación de Grid Search y Optuna permitió identificar configuraciones óptimas para los modelos Random Forest y XGBoost, mejorando significativamente su desempeño. Las métricas y análisis demostraron que los modelos son robustos y efectivos para clasificar actividades humanas basadas en poses. Este enfoque puede escalarse fácilmente a problemas más complejos con datasets más amplios y variados.

# Resultados

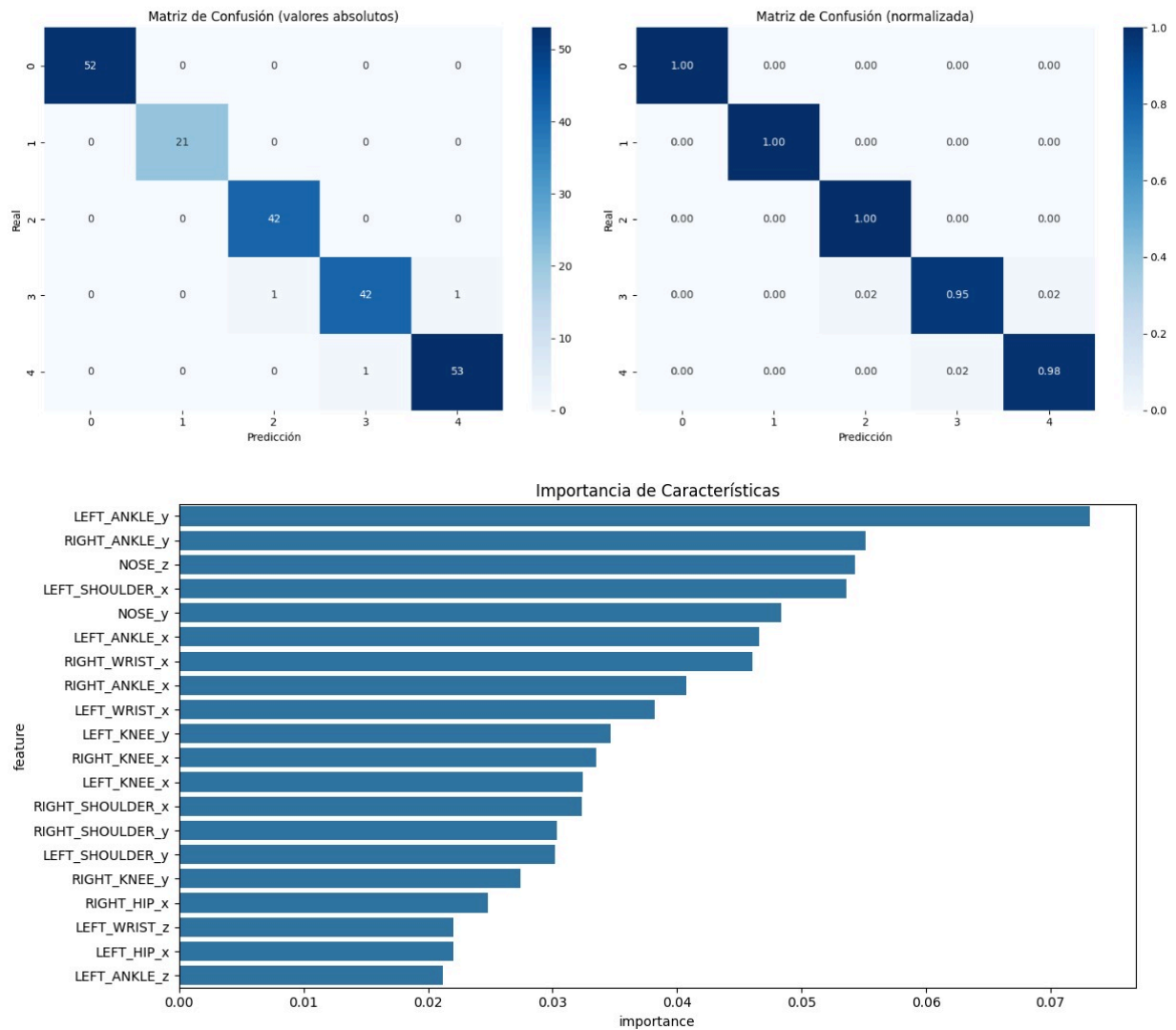
## Desempeño del Modelo

Los modelos entrenados lograron una alta precisión en la clasificación de actividades. Los resultados de las evaluaciones se resumen en la siguiente tabla:

### Métricas de Interés

Las métricas de interés incluyen la precisión, recall, f1-score, la matriz de confusión y la importancia de características. Estas métricas se utilizaron para evaluar y comparar el rendimiento de los modelos.

Métricas de Evaluación:				
Accuracy: 0.9859				
Informe detallado por clase:				
	precision	recall	f1-score	support
sitting	1.00	1.00	1.00	52
standing	1.00	1.00	1.00	21
turning	0.98	1.00	0.99	42
walking_away	0.98	0.95	0.97	44
walking_towards	0.98	0.98	0.98	54
accuracy			0.99	213
macro avg	0.99	0.99	0.99	213
weighted avg	0.99	0.99	0.99	213



## Análisis de Resultados

### Observaciones

#### 1. Métricas de rendimiento:

- La exactitud del modelo es **0.9859 (98.59%)**, lo que es muy alta.
- Las métricas de **precisión**, **recall** y **F1-score** por clase son excelentes, con la mayoría de los valores por encima de **0.98**:
  - Para las clases **sitting** y **standing**, el rendimiento es perfecto, con un **F1-score** de **1.00**.
  - Hay ligeras desviaciones en las clases **walking\_away** y **walking\_towards**, donde el recall baja a **0.95** y **0.98**, respectivamente.

#### 2. Importancia de características:

- Las características más importantes son **LEFT\_ANKLE\_y** y **RIGHT\_ANKLE\_y**, lo que sugiere que el modelo se basa principalmente en los movimientos de la parte inferior del cuerpo para clasificar.
- Características como **NOSE\_z** y **LEFT\_SHOULDER\_x** también tienen un impacto significativo, probablemente capturando información relacionada con la postura.

### 3. Matriz de confusión:

- En las matrices de confusión (valores absolutos y normalizados):
  - La mayoría de las predicciones coinciden perfectamente con las etiquetas reales, ya que los valores diagonales están cerca de 1.
  - Existen algunas pequeñas confusiones, especialmente entre las clases **walking\_away** y **walking\_towards**, posiblemente debido a similitudes en los movimientos.

## Generalización del Modelo

- **Generalización:** La alta precisión y los valores balanceados de precisión y recall en todas las clases indican que el modelo generaliza bien sobre los datos de prueba.
- **Overfitting:** No hay evidencia directa de sobreajuste (overfitting). Sin embargo, dado que la exactitud es muy alta, sería importante confirmar si el rendimiento en el conjunto de entrenamiento no es significativamente superior al de validación. Si los datos de prueba son pequeños o similares a los de entrenamiento, se recomienda validar en datos completamente nuevos.

## Conclusiones y Trabajo Futuro

### ¿Qué hiciste?

**Implementación de un Sistema de Reconocimiento de Actividades en Tiempo Real:** Se desarrolló un pipeline completo para reconocer actividades humanas en tiempo real utilizando estimación de poses y modelos de aprendizaje automático. Esto incluyó:

- **Recolección de Datos:** Creación de una interfaz gráfica personalizada (Tkinter) para capturar datos de movimiento en tiempo real mediante una cámara y etiquetar las actividades.
- **Estimación de Poses:** Uso de MediaPipe para extraer 33 puntos clave 3D del cuerpo humano en tiempo real.
- **Ingeniería de Características:** Cálculo de coordenadas normalizadas y características angulares a partir de los datos de las poses.
- **Entrenamiento de Modelos:** Entrenamiento y prueba de modelos Random Forest y XGBoost para clasificar actividades como caminar, estar de pie, sentarse y girar.
- **Evaluación:** Análisis del rendimiento de los modelos utilizando métricas como precisión, sensibilidad, especificidad y puntaje F1 para validar la efectividad del sistema.



## ¿Qué aprendiste?

### Fortalezas de las Técnicas de Estimación de Poses:

- MediaPipe es muy robusto para aplicaciones en tiempo real, ofreciendo detección precisa de puntos clave incluso bajo diferentes condiciones de iluminación y ángulos de cámara.
- Las coordenadas normalizadas y las relaciones angulares entre puntos clave aumentan el poder discriminativo de las características, mejorando la precisión de la clasificación.

### Rendimiento de los Modelos de Aprendizaje Automático:

- Random Forest es confiable en tareas con ruido y datos de alta dimensionalidad, proporcionando un buen rendimiento base.
- XGBoost demuestra una mayor precisión en problemas de clasificación complejos debido a su capacidad para corregir errores iterativos y aplicar regularización.

### Desafíos en Sistemas en Tiempo Real:

- El ruido en los datos de estimación de poses (por ejemplo, puntos clave parcialmente ocultos) puede afectar la precisión de la clasificación.
- La eficiencia es crucial para aplicaciones en tiempo real, ya que modelos más complejos como XGBoost requieren mayores recursos computacionales.

### Importancia del Preprocesamiento:

- Un buen preprocesamiento, que incluya normalización y la ingeniería de características (ángulos), mejora significativamente el rendimiento del modelo al reducir la variabilidad causada por diferencias individuales o configuraciones de cámara.

## ¿Qué podría mejorarse?

### 1. Recolección de Datos:

- **Diversidad:** Ampliar el conjunto de datos incluyendo una mayor variedad de individuos con diferentes tipos de cuerpo, vestimenta y estilos de movimiento.
- **Actividades Adicionales:** Incorporar actividades más complejas o sutiles para probar la escalabilidad del sistema.
- **Variaciones Ambientales:** Capturar datos bajo diferentes condiciones ambientales (por ejemplo, iluminación, ruido de fondo).

### 2. Optimización de Modelos:

- **Ajuste de Hiperparámetros:** Realizar una optimización más exhaustiva utilizando herramientas como Optuna para los modelos Random Forest y XGBoost.
- **Modelos de Aprendizaje Profundo:** Experimentar con arquitecturas de aprendizaje profundo, como LSTM o Transformers, para capturar dinámicas temporales en datos secuenciales de poses.

# Referencias Bibliográficas

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning with applications in *Python*. (<https://www.statlearning.com/resources-python>)

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc."

Google. (2024). *MediaPipe*. <https://google.github.io/mediapipe/>

CMU Perceptual Computing Lab. (2024). *OpenPose*.  
<https://github.com/CMU-Perceptual-Computing-Lab/openpose>

Intel. (2024). *CVAT*. <https://github.com/openvinotoolkit/cvat>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. <https://scikit-learn.org/>

Chen, T., & Guestrin, C. (2024). *XGBoost: A scalable tree boosting system*. <https://xgboost.ai/>

Tkinter. (2024). *Tkinter*. <https://tkdocs.com/>