



**Universidad  
Europea**

**UNIVERSIDAD EUROPEA DE MADRID**

**ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO**

**GRADO EN FÍSICA**

**TRABAJO FIN DE GRADO**

**Aplicación de Visión Artificial y Redes Neuronales  
Recurrentes para predecir la evolución temporal  
de un movimiento caótico**

**VÍCTOR MANUEL SUÁREZ FERRER**

**Dirigido por**

**Dr. JOSE MANUEL LÓPEZ LÓPEZ**

**CURSO 2024/2025**

Victor Manuel Suárez Ferrer

---

**TÍTULO:** APLICACIÓN DE VISIÓN ARTIFICIAL Y REDES NEURONALES RECURRENTE PARA PREDECIR LA EVOLUCIÓN TEMPORAL DE UN MOVIMIENTO CAÓTICO.

**AUTOR:** VÍCTOR SUÁREZ FERRER

**TITULACIÓN:** GRADO EN FÍSICA

**DIRECTOR/ES DEL PROYECTO:** Dr. JOSE MANUEL LÓPEZ LÓPEZ

**FECHA:** MAYO de 2025

## RESUMEN

Este proyecto explora la predicción del movimiento caótico de un péndulo doble mediante inteligencia artificial, utilizando visión artificial y redes neuronales recurrentes, sin depender de ecuaciones dinámicas. De esta forma se resalta el potencial de la IA para analizar fenómenos no lineales de manera práctica, accesible y eficiente, abriendo nuevas posibilidades en el estudio de dinámicas complejas.

El método combina un modelo de visión artificial (YOLOv11n-pose) para detectar puntos clave en secuencias de video y una red LSTM para predecir la evolución temporal del movimiento caótico. Los resultados muestran una precisión del 99.94% en la detección de puntos clave, con errores medios de  $1.83^\circ$  y  $7.67^\circ$  en los ángulos predichos de los dos eslabones del péndulo, demostrando la robustez del modelo.

Este enfoque resulta valioso en aplicaciones prácticas, como el diseño de sistemas robóticos que requieren adaptarse a perturbaciones impredecibles, o en entornos educativos para ilustrar conceptos de caos de forma interactiva. No obstante, la extrema sensibilidad del sistema a pequeñas variaciones y las restricciones del equipo utilizado revelan áreas de mejora.

**Palabras clave:** Inteligencia Artificial, IA, Visión Artificial, VA, YOLO, Redes Neuronales, LSTM, Caos, Sistemas dinámicos.

## ABSTRACT

This project investigates the prediction of chaotic motion in a double pendulum using artificial intelligence, leveraging computer vision and recurrent neural networks without relying on dynamic equations. It showcases AI's ability to tackle nonlinear phenomena in a practical, accessible, and efficient way, paving the way for new approaches to studying complex dynamics.

The approach uses a computer vision model (YOLOv11n-pose) to identify key points in video sequences and an LSTM network to forecast the temporal evolution of chaotic motion. The results demonstrate a remarkable 99.94% accuracy in detecting key points, with average prediction errors of  $1.83^\circ$  and  $7.67^\circ$  for the angles of the pendulum's two links, highlighting the method's robustness.

This method could hold significant value for practical applications, such as designing robotic systems that adapt to unpredictable disturbances or creating interactive educational tools to illustrate chaos concepts vividly. However, the system's extreme sensitivity to minor variations and limitations in the equipment used point to areas for improvement.

**Keywords:** Artificial Inteligencia, IA, Artificial Vision, VA, YOLO, Neural Networks, LSTM, Chaos, Dynamic Systems.

Victor Manuel Suárez Ferrer

---

## TABLA RESUMEN

	<b>DATOS</b>
<b>Nombre y apellidos:</b>	Victor Manuel Suarez Ferrer
<b>Título del proyecto:</b>	Aplicación de Visión Artificial y Redes Neuronales Recurrentes para predecir la evolución temporal de un movimiento caótico
<b>Directores del proyecto:</b>	Dr. Jose Manuel López López
<b>El proyecto se ha realizado en colaboración de una empresa o a petición de una empresa:</b>	NO
<b>El proyecto ha implementado un producto:</b>	SI
<b>El proyecto ha consistido en el desarrollo de una investigación o innovación:</b>	SI
<b>Objetivo general del proyecto:</b>	Predecir movimiento caótico con inteligencia artificial

# INDICE

RESUMEN .....	3
ABSTRACT .....	4
TABLA RESUMEN .....	5
Capítulo 1. INTRODUCCIÓN / ESTADO DEL ARTE .....	10
1.1 Estado del arte .....	11
1.2 Contexto y justificación .....	12
1.3 Planteamiento del problema .....	13
Capítulo 2. OBJETIVOS .....	14
2.1 Objetivos generales .....	14
2.2 Objetivos específicos .....	14
2.3 Beneficios del proyecto .....	14
Capítulo 3. MARCO TEÓRICO .....	15
3.1 MECÁNICA LAGRANGIANA .....	15
3.2 LIGADURAS .....	16
3.3 COORDENADAS GENERALIZADAS .....	18
3.3.1 ENERGIA CINETICA EN COORDENADAS GENERALIZADAS .....	19
3.4 ECUACIONES DE LAGRANGE DESDE LA MECÁNICA LAGRANGIANA .....	20
3.5 SISTEMAS DINÁMICOS .....	23
3.5.1 SISTEMAS DINÁMICOS DISCRETOS .....	24
3.5.2 SISTEMAS DINÁMICOS CONTINUOS .....	24
3.5.3 SISTEMAS LINEALES .....	25
3.5.4 SISTEMAS NO LINEALES .....	26
3.5.5 SISTEMAS CONSERVATIVOS .....	28
3.5.6 SISTEMAS DISIPATIVOS .....	29
3.6 TEORIA DEL CAOS .....	31
3.6.1 SISTEMAS CAÓTICOS .....	31
3.7 REDES NEURONALES .....	34
3.7.1 REDES NEURONALES RECURRENTE (RNNs) .....	35
3.7.2 REDES LSTM .....	36

3.8	VISIÓN ARTIFICIAL .....	38
3.9	POSE DETERMINATION .....	39
3.9.1	POSE DETERMINATION CON YOLO .....	39
Capítulo 4.	DESARROLLO .....	43
4.1	Planificación del trabajo fin de grado .....	43
4.2	PÉNDULO SIMPLE PLANO .....	44
4.3	PÉNDULO DOBLE .....	45
4.4	CONSTRUCCION DEL PENDULO .....	48
4.4.1	BASE .....	49
4.4.2	ESLABONES .....	49
4.5	SIMULACIÓN EN PYTHON .....	51
4.6	MODELO DE VISIÓN ARTIFICIAL .....	53
4.7	RED NEURONAL LSTM .....	57
Capítulo 5.	RESULTADOS .....	61
5.1	Simulación Python .....	61
5.2	VISIÓN ARTIFICIAL .....	61
5.3	RED NEURONAL LSTM .....	68
Capítulo 6.	DISCUSIÓN .....	73
6.1	LIMITACIONES DEL ESTUDIO .....	73
6.2	LIMITACIONES DE LA TECNOLOGIA USADA .....	73
6.3	CAMBIO RESPECTO A LOS OBJETIVOS INICIALES .....	74
6.4	IMPACTO DEL PROYECTO .....	75
Capítulo 7.	CONCLUSIONES .....	76
7.1	Conclusiones del trabajo .....	76
Capítulo 8.	FUTURAS LÍNEAS DE TRABAJO .....	77
Capítulo 9.	REFERENCIAS .....	78
Capítulo 10.	ANEXOS .....	80
10.1	Codigos .....	80
10.2	Declaración de uso de IA .....	80

## Índice de Figuras

Figura 1: Particula en aro.....	15
Figura 2: Péndulo doble.....	17
Figura 3: Disco rodando sin deslizar.....	18
Figura 4 Oscilador armónico simple.....	25
Figura 5 Ejemplo sistema no lineal: Péndulo simple sin aproximaciones.....	26
Figura 6 Diagrama de fases péndulo simple.....	33
Figura 7 Diagrama de fases péndulo doble.....	33
Figura 8 Arquitectura de una red RNN tradicional.....	35
Figura 9 Capacidad de la red LSTMS para mantener información a largo plazo.....	37
Figura 10 Representación de la puerta de salida [15].....	38
Figura 11 Péndulo doble con información.....	45
Figura 12 Esquema eslabón (barra y rodamiento).....	46
Figura 13 Base vista desde atrás en plano picado.....	49
Figura 14 Eslabones.....	50
Figura 15 Resultado final del montaje experimental.....	51
Figura 16 Captura de pantalla para ejemplificar el funcionamiento de CVAT.....	53
Figura 17 Mosaico batch.....	55
Figura 18 Simulación péndulo doble.....	61
Figura 19 Conjunto de validación.....	62
Figura 20 Falsos positivos del eje principal como masa.....	63
Figura 21 Gráficos de línea VA.....	64
Figura 22 Gráfico de barras VA.....	65
Figura 23 Porcentaje de frames incorrectos.....	66
Figura 24 Gráfico de barras métricas de rendimiento.....	67
Figura 25 Pivote.....	67
Figura 26 Ángulos experimentales.....	68
Figura 27 Resultados entrenamiento LSTM.....	68
Figura 28 Histograma LSTM.....	69
Figura 29 Gráfico de líneas LSTM.....	70
Figura 30 Grafica de la evolución de perdida LSTM.....	71
Figura 31 Gráficas de comparación entre ángulos.....	72

## Índice de Tablas

Tabla 1 Bibliotecas usadas en la simulación del péndulo doble en Python	¡Error! Marcador no definido.
Tabla 2 Datos experimentales introducidos en la simulación .....	52
Tabla 3 Metricas conjunto de validación en el entrenamiento 1 .....	61
Tabla 4 Metricas conjunto de prueba en el entrenamiento 1 .....	63

## Capítulo 1. INTRODUCCIÓN / ESTADO DEL ARTE

La física es la ciencia que estudia, mediante leyes fundamentales, la energía, la materia, el tiempo y el espacio, es decir, el universo mismo. La física examina experimental y teóricamente la naturaleza, y una de las preguntas que ha llevado a la realización de este proyecto es el movimiento caótico. Un sistema que experimenta un movimiento caótico nunca se repite a sí mismo, sino que más bien se comporta de forma diferente en distintas situaciones. Uno de los aspectos más importantes o incluso el más importante del caos es su extrema sensibilidad a las condiciones iniciales. En este contexto, la visión artificial emerge como una herramienta clave, permitiendo medir el estado de sistemas como el péndulo doble de manera no invasiva, sin alterar su dinámica física, lo que garantiza la integridad de los datos experimentales.

Un ejemplo muy sencillo de caos es el péndulo. Cuando lo dejamos oscilar desde una posición inicial, su movimiento sigue un patrón predecible. Sin embargo, si acoplamos más péndulos o aplicamos fuerzas externas, su comportamiento se vuelve más complicado, de tal forma que pequeñas variaciones dan lugar a patrones de movimiento distintos. De hecho, el péndulo doble es uno de los sistemas más simples que experimentan comportamiento caótico

El hecho de que lo consideremos un sistema simple mecánicamente puede llevar a pensar que la física que lo gobierna también es sencilla. Sin embargo, se deben aplicar métodos avanzados como la formulación lagrangiana, la cual simplifica considerablemente muchos problemas físicos. Este modelo introducido por Joseph-Louis de Lagrange en 1788, permite obtener las ecuaciones de movimiento que imperan en el péndulo doble, dotándolo así de cierto determinismo.

Es interesante notar que el caos es determinista, en el sentido de que estos sistemas evolucionan siguiendo unas ecuaciones bien definidas pero que dan lugar a esta peculiaridad debido a la complejidad de estas. Esto nos conduce a la idea de simular el comportamiento del péndulo doble a través de una simulación, mostrando así el movimiento caótico el cual queremos predecir.

El siglo XX fue indudablemente la era de la física, marcada por una gran cantidad de avances científicos que transformaron nuestro entendimiento del universo. Desde la mecánica cuántica y la relatividad general de Einstein hasta teorías aún más complejas. Ahora, en el primer cuarto del siglo XXI, estamos sin lugar a duda ante la era de la Inteligencia Artificial.

Los seres humanos tenemos cierta intuición y no dudamos entre distinguir un perro de un lobo, pero enseñar a una máquina a distinguirlos no es tan sencillo ya que tienen rasgos similares. Es aquí donde entran grandes desarrolladores de machine learning como Ultralytics y sus modelos de visión artificial.

La visión artificial es una tecnología que utilizan las máquinas para reconocer automáticamente las imágenes y describir lo que “ven” en ellas. Puede reconocer rostros en las fotos, distinguir objetos e incluso detectar galaxias en una imagen inmensa abarrotada de ruido. Sin embargo, en este trabajo se ira un paso más allá, utilizando un modelo de pose estimation.

Pose estimation es una técnica que permite detectar la posición y la orientación de objetos o cuerpos en imágenes y videos. Se basa en identificar puntos clave (keypoints) que representan partes significativas que queremos determinar. Por ejemplo, en un péndulo doble, puede detectar los centros de masa, el pivote central o cualquier elemento que se desee.

En este proyecto, por medio de una técnica no invasiva se detectarán los keypoints de un péndulo doble previamente fabricado y se devolverán las posiciones de los centros de masa en cada frame del video. Estos datos se usarán para extraer los ángulos experimentales y de esta forma alimentar una red neuronal recurrente (RNN), en concreto una red neuronal de memoria de corto-largo plazo (LSTM) y con ella predecir el comportamiento futuro.

## 1.1 Estado del arte

Se plantea resolver un sistema caótico sin el uso de las ecuaciones físicas que lo gobiernan, empleando nuevas tecnologías como visión artificial y redes neuronales.

La visión artificial (VA) permite rastrear y procesar datos visuales de sistemas caóticos, como el movimiento de un péndulo, a través de un entrenamiento previo con una serie de imágenes que ofrezcan los parámetros de correlación pertinentes. Mientras que las redes neuronales recurrentes (RNN), especialmente las redes de memoria a largo plazo (LSTM), son ideales para predecir secuencias temporales y capturar dependencias a largo plazo.

En el artículo [1] se presenta un sistema de VA que permite recolectar datos sobre el movimiento caótico de un péndulo doble utilizando un software de rastreo de video de código abierto (Tracker). Los datos experimentales que se obtuvieron coinciden con simulaciones numéricas, confirmando así la ventaja que supone la VA en movimientos caóticos. Otro de los fenómenos caóticos comunes que actualmente se está estudiando con inteligencia artificial es la meteorología, donde artículos como [2] y [3] ofrecen modelos que combinan VA y arquitecturas convolucionales para modelos predictivos.

Por otro lado, en [4] se propone un enfoque basado en RNN como solución a predicciones en sistemas caóticos. Los autores lograron predicciones en sistemas como el de Lorenz o el péndulo doble, lo que aúpa la integración de esta tecnología en este proyecto. El desarrollo de las redes neuronales ha proporcionado modelos capaces de mitigar los efectos de ruido, logrando predicciones más robustas como se expone en [5].

La combinación de ambas técnicas en sistemas mecánicos caóticos aún no está muy introducida, ya que la predicción es limitada debido a la divergencia exponencial de las trayectorias, así como el ruido a causa de iluminaciones variables o limitaciones de la cámara. Es por eso que en artículos como [6] proponen integrar teoría de sistemas dinámicos para mejorar las predicciones. La proyección de combinar VA y RNN va más allá de modelos simples como un péndulo doble, expandiéndose a futuras aplicaciones en finanzas, biología o astrofísica.

## 1.2 Contexto y justificación

En primer lugar, debemos ser conscientes de la revolución tecnológica que está experimentando nuestra sociedad. Nuevos avances como el desarrollo de inteligencia artificial, ayuda y “obliga” en cierta medida a los científicos a implementarlas en sus investigaciones.

A lo largo de la carrera los estudiantes de física se enfrentan a problemas sumamente complejos que desarrollan su capacidad resolutoria y promueven la búsqueda de nuevos problemas cada vez más irresolubles. Impulsado por esta misma idea, decidí integrar estas nuevas tecnologías a lo que considero un problema complejo matemáticamente pero relativamente fácil de reproducir. De esta forma, el público que lea este artículo podrá ver el enfoque clásico y el enfoque actual que se le podría dar, sirviendo como ejemplo para futuras soluciones a los problemas que se ha enfrentado la física a lo largo de la historia o incluso a las múltiples cuestiones aun sin resolver.

El concepto de caos se popularizó en la segunda mitad del siglo XX, cuando los científicos empezaron a notar que incluso sistemas relativamente simples podían mostrar comportamientos extremadamente complejos e impredecibles. El péndulo doble es uno de los ejemplos clásicos de sistemas caóticos. A pesar de su simplicidad mecánica, su movimiento resulta extremadamente difícil de predecir a largo plazo. Edward Lorenz, en 1963, contribuyó al campo con el famoso "efecto mariposa", demostrando que pequeñas variaciones en las condiciones iniciales podían tener efectos masivos en el resultado final, lo que hizo evidente la complejidad y dificultad de predecir estos sistemas.

Actualmente, con conocimientos de programación en Python o herramientas como Matlab o Wolfram Mathematica, podemos facilitar la solución del péndulo doble. Estos modelos nos permiten resolver las ecuaciones diferenciales del modelo dinámico, sin embargo, necesitamos darle al programa estas ecuaciones. Esto quiere decir que se requiere un mínimo de base física y matemáticas.

Por otro lado, el desarrollo de nuevas tecnologías como la inteligencia artificial, ofrece un nuevo enfoque para estudiar y predecir estos sistemas caóticos. Este proyecto busca extraer información directamente de los datos visuales para hacer predicciones sin la necesidad explícita del uso de ecuaciones diferenciales. La idea es desarrollar un modelo que utilice Visión Artificial (Yolo) y que proporcione los ángulos experimentales del péndulo doble en tiempo real. Además de una Red Neuronal Recurrente (LSTM) para predecir la evolución temporal del sistema y que aprenda las relaciones temporales de estos datos sin conocer las ecuaciones físicas que gobiernan este sistema. Lo que lo hace aplicable incluso cuando las aproximaciones matemáticas son difíciles de manejar o conocer.

La motivación detrás de este proyecto radica en la posibilidad de aplicar sus resultados en una variedad de campos científicos y de ingeniería. En concreto, el péndulo doble es un sistema dinámico bastante presente en la ingeniería, dado que es asimilable a un brazo robótico. Por ejemplo, si el péndulo doble es predecible, significa que podemos conseguir un brazo que se estabilice solo tras una perturbación.

Si bien existen investigaciones previas que utilizan redes neuronales para la predicción de sistemas dinámicos, este proyecto se diferencia al proponer una combinación directa de Visión Artificial en tiempo real con predicciones basadas en secuencias temporales, lo cual representa un enfoque novedoso.

### **1.3 Planteamiento del problema**

Los sistemas caóticos, como el péndulo doble, el clima o los flujos turbulentos, se caracterizan por su extrema sensibilidad a las condiciones iniciales, lo que genera trayectorias que divergen rápidamente y dificulta su predicción a largo plazo. Esta sensibilidad, conocida como el efecto mariposa, plantea un desafío tanto teórico como práctico en disciplinas como la física, la meteorología y la ingeniería. En el caso del péndulo doble, un sistema mecánico clásico, el comportamiento caótico emerge bajo ciertas condiciones (por ejemplo, grandes amplitudes o energías), lo que lo convierte en un modelo ideal para estudiar la predicción de dinámicas no lineales.

La combinación de VA y RNN para este tipo de sistemas ha sido estudiada en meteorología, sin embargo, ha sido muy poco explorada en sistemas mecánicos. La falta de soluciones en este campo genera la necesidad de una nueva propuesta que ayude a predecir datos experimentales. Con esto en mente se prevé capturar datos usando un modelo de VA, usar una RNN lo más robusta al ruido posible para predecir las trayectorias y eliminar el componente físico y matemático que hay detrás de estos sistemas tan complejos, ofreciendo así una solución práctica con aplicaciones en investigación y educación.

## Capítulo 2. OBJETIVOS

### 2.1 Objetivos generales

Diseñar un sistema que, a partir de vídeo del movimiento de un péndulo doble, utilice visión artificial para extraer sus coordenadas y una red neuronal recurrente (LSTM) para predecir su evolución temporal.

### 2.2 Objetivos específicos

- Llegar al mayor público posible, presentando modelos visuales (simulaciones) del problema a abordar.
- Dar una base sólida de los fundamentos físicos necesarios para comprender el proyecto.
- Introducir nociones de programación de inteligencia artificial, abordando temas como redes neuronales, visión artificial y estimación de pose.
- Generar un sistema caótico simple que sirva como fuente de datos para la visión artificial.
- Desarrollar un modelo de visión artificial lo más robusto posible para la detección precisa de un péndulo doble.
- Diseñar y entrenar una red neuronal LSTM, con la capacidad de predecir el comportamiento futuro del sistema.

### 2.3 Beneficios del proyecto

Con este trabajo se aportará una mayor comprensión del caos a un público extenso, tanto para estudiantes o investigadores de la rama de ciencias que quieran profundizar en esta área, y lectores sin una base tan sólida que puedan comprender los conceptos clave gracias a las simulaciones. Además, cuenta con una aplicación práctica de temas en auge actualmente como son las redes neuronales y la visión artificial, ofreciendo un sistema experimental simple y reproducible al que se le pueden aplicar estas tecnologías.

Este proyecto puede suponer un antes y un después en la física ya que uno de los mayores problemas que tiene, es el conocimiento matemático que requiere, limitando su acceso a un gran número de personas. Por lo tanto, este artículo podría servir como punto de partida para futuras investigaciones que permitan resolver sistemas complejos mediante inteligencia artificial sin necesidad de las ecuaciones dinámicas que lo gobiernan.

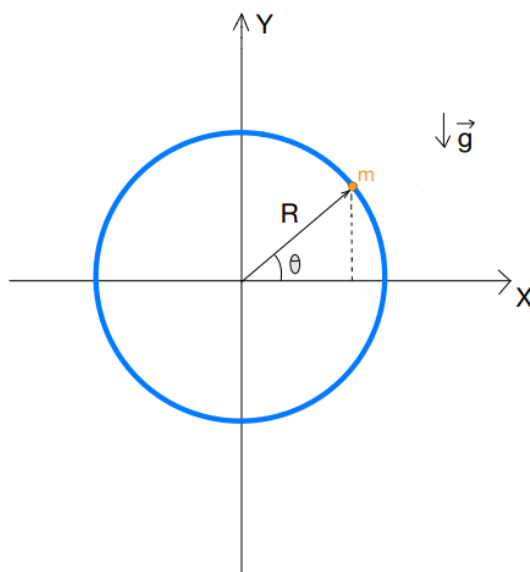
## Capítulo 3. MARCO TEÓRICO

### 3.1 MECÁNICA LAGRANGIANA

La mecánica lagrangiana es una reformulación de la mecánica clásica introducida por Joseph-Louis de Lagrange en 1788. En la mecánica lagrangiana, la trayectoria de un objeto es obtenida encontrando la trayectoria que minimiza la acción, que es la integral del lagrangiano en el tiempo; siendo este la energía cinética del objeto menos la energía potencial del mismo.

$$L = T - U \quad (1)$$

Este método es especialmente útil para sistemas complejos, como los caóticos que se estudiarán en este TFG, ya que simplifica el tratamiento de restricciones y dinámicas no lineales. Sin embargo, la utilidad de la formulación lagrangiana se aprecia incluso en ejemplos sencillos. Por ejemplo, se considera una partícula que se mueve en un aro:



*Figura 1* Esquema representativo de una partícula moviéndose dentro de un aro bajo la acción de la gravedad, utilizado para ilustrar la aplicación de la mecánica lagrangiana en sistemas dinámicos.

---

Si se calculara el movimiento de la partícula usando la mecánica newtoniana, se obtendría un sistema complicado de ecuaciones donde se considerarían fuerzas como la gravedad y la normal. En cambio, en la aproximación de Lagrange, se mira todos los movimientos posibles que la partícula podría tomar en el aro y se encuentra matemáticamente el que reduce al mínimo la acción.

Otro ejemplo son los movimientos referidos a un sistema que gira, como por ejemplo observaciones astronómicas vistas desde la Tierra. En la formulación newtoniana es necesario introducir a mano las fuerzas ficticias o fuerzas de inercia como la fuerza centrífuga o el efecto Coriolis mientras que en la formulación lagrangiana estas fuerzas aparecen de modo natural como se verá más adelante al resolver el péndulo doble por este método.

Por otro lado, las ecuaciones de movimiento de Lagrange (*Sección 3.4*) tienen un formalismo escalar debido a sus consideraciones puramente energéticas, facilitando así el análisis para cualquier sistema mecánico. Además, presentan la ventaja de escribirse igual para cualquier coordenada generalizada.

Para el uso de las ecuaciones de Lagrange es necesario seguir cuatro etapas:

- 1) Cálculo de la energía cinética.
- 2) Cálculo de la energía potencial.
- 3) Cálculo del lagrangiano.
- 4) Resolver las ecuaciones para cada grado de libertad.

Además, debemos de tener en cuenta conceptos importantes como el de coordenada generalizada y ligadura. Es por ello que en este capítulo se abordaran todos los conceptos importantes necesarios para la resolución de un sistema dinámico desde la mecánica lagrangiana.

## 3.2 LIGADURAS

Las ligaduras, o restricciones, en mecánica son condiciones que limitan el movimiento de un sistema, un concepto fundamental para analizar sistemas dinámicos como péndulos, ruedas o cuerpos rígidos. A veces estas ligaduras dependen del tiempo: por ejemplo, una pelota girando encima de una mesa unida a una cuerda que llegue hasta un agujero central y con el tiempo se va deslizando hasta el suelo, estrechando el radio de giro de la pelota.

Estas limitaciones pueden ser físicas, como barreras, o matemáticas, como relaciones entre coordenadas. Por ejemplo, en un péndulo simple, la restricción es que la distancia del pivote a la masa del extremo es constante.

En mecánica de Lagrange tenemos dos posibilidades:

**Ligaduras holónomas:** Son ligaduras que pueden expresarse por una relación matemática que involucre solamente las coordenadas geométricas  $y$ , quizás, el tiempo. Si el sistema tiene  $N$  partículas, la expresión matemática general de una ligadura holónoma es [7]:

$$f(x_1, x_2, \dots, x_{3N}, t) = 0 \quad (2)$$

Como vemos las velocidades no aparecen en la expresión. Si el tiempo no aparece explícitamente en la expresión la ligadura se denomina esclerónoma. Si el tiempo sí aparece explícitamente se llama reónoma.

Dentro de las ligaduras holónomas, podemos diferenciar entre geométricas bilaterales y cinemáticas integrables.

Las geométricas bilaterales establecen una relación de igualdad entre las coordenadas. Por ejemplo, una partícula rodando adherida en una circunferencia de radio  $R$  centrada en el origen

Victor Manuel Suárez Ferrer

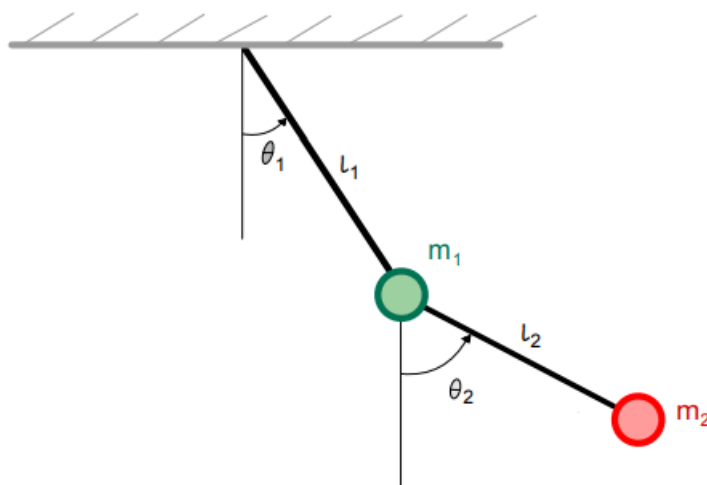
---

y en el plano XY (*Figura 1*) es un ejemplo de ligadura holónoma geométrica bilateral y esclerónoma.

$$x^2 + y^2 = R \quad (3)$$

Las ligaduras cinemáticas son aquellas en las que se imponen condiciones sobre alguna o todas las velocidades. Si la expresión puede integrarse para eliminar las velocidades se convierte en una ligadura geométrica bilateral [7]. Por tanto, es holónoma.

Las ligaduras que restringen nuestro sistema de péndulo doble son ligaduras holónomas geométricas bilaterales y esclerónomas.



*Figura 2* Esquema péndulo doble fuera del equilibrio, con eslabones de longitud  $l_1$  y  $l_2$ , masas  $m_1$  y  $m_2$  en los extremos y ángulos  $\theta_1$  y  $\theta_2$  respecto a la vertical, ilustrando un sistema con ligaduras holónomas.

---

Tenemos un sistema de 2 dimensiones con 2 partículas y dos ligaduras (las longitudes de los eslabones son constantes), por tanto:

$$s = 2 \times 2 - 2 = 2 \text{ c. g} \quad (4)$$

**Las ligaduras no holónomas** son más complicadas de tratar y en este caso no pueden expresarse con una relación matemática de la forma de la ecuación (2).

Lo ideal es que la ligadura sea integrable, es decir, que sea del tipo cinemática que se ha mencionado anteriormente. Un ejemplo sencillo es el de un disco que rueda sobre un plano inclinado, cuya ligadura en realidad se escribe más apropiadamente en términos de velocidades [8]. El disco rueda sin deslizar si la velocidad del punto de contacto del disco con el plano es nula. Esta velocidad es igual a la velocidad del centro de masas, que es  $\dot{x}$ , más la velocidad del punto con respecto al centro de masas, que es  $-R\dot{\theta}$  (con signo negativo porque va en la dirección negativa de las x). La ligadura es entonces  $\dot{x} - R\dot{\theta} = 0$ , que es una ligadura no holónoma. Pero esta ligadura se puede integrar y transformar en holónoma:

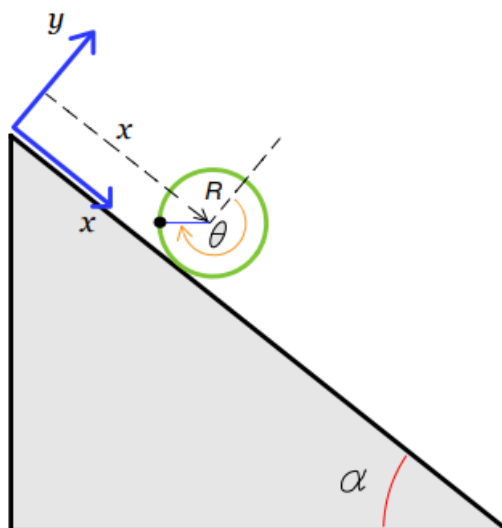


Figura 3 Esquema de un disco de radio  $R$  sin deslizar en un plano inclinado con ángulo  $\alpha$ , utilizado para ilustrar una ligadura cinemática integrable en mecánica lagrangiana. Fuente: adaptada de [8].

$$\frac{dx}{dt} = R \frac{d\theta}{dt} \rightarrow dx = R d\theta \quad (5)$$

$$\int_{x(0)}^{x(t)} dx = R \int_{\theta(0)}^{\theta(t)} d\theta \rightarrow x(t) - x(0) = R(\theta(t) - \theta(0))$$

### 3.3 COORDENADAS GENERALIZADAS

Se llaman grados de libertad de un sistema,  $s$ , al número de maneras independientes en que se puede mover éste (incluyendo implícitamente las ligaduras). Entonces:

$$s = \text{Dimensiones del movimiento} \times N - \text{Ligaduras} (M) \quad (6)$$

Llamamos coordenadas generalizadas a cualquier conjunto de  $s$  coordenadas que nos permite especificar de manera única el estado del sistema. Pueden ser coordenadas cartesianas, polares, etc. El conjunto de coordenadas generalizadas es [8]:

$$q_1(t), q_2(t), q_3(t), \dots, q_s(t) \quad (7)$$

A este conjunto asociamos un conjunto de velocidades generalizadas:

$$\dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dots, \dot{q}_s(t) \quad (8)$$

A partir de las coordenadas cartesianas, y siempre que las ecuaciones de ligadura sean holónomas (esto es, involucren sólo coordenadas y no velocidades), vamos a poder poner unas coordenadas cartesianas en términos de las otras y definir un conjunto posible de coordenadas generalizadas.

$$x_{\alpha i} = x_{\alpha i}(q_j, t) \rightarrow \dot{x}_{\alpha i} = \sum_j \frac{\partial x_{\alpha i}}{\partial q_j} \dot{q}_j + \frac{\partial x_{\alpha i}}{\partial t} \quad (9)$$

Vemos que el ultimo termino solo aparecerá si hay una dependencia explicita de la coordenada cartesiana con el tiempo.

### 3.3.1 ENERGIA CINETICA EN COORDENADAS GENERALIZADAS

La energía cinética en coordenadas cartesianas es de una forma cuadrática homogénea de las velocidades [8]:

$$T = \frac{1}{2} \sum_{\alpha} \sum_I m_{\alpha} \dot{x}_{\alpha i}^2 \quad (10)$$

Donde el índice  $\alpha$ , representa las diferentes partículas en el sistema y el índice  $i$  representa las componentes de las coordenadas cartesianas de cada partícula.

Como se había visto en la ecuación (9), podemos expresar el cuadrado de  $\dot{x}$  como:

$$\dot{x}_{\alpha i}^2 = \sum_{kj} \frac{\partial x_{\alpha i}}{\partial q_j} \frac{\partial x_{\alpha i}}{\partial q_k} \dot{q}_j \dot{q}_k + 2 \sum_j \frac{\partial x_{\alpha i}}{\partial q_j} \frac{\partial x_{\alpha i}}{\partial t} \dot{q}_j + \left( \frac{\partial x_{\alpha i}}{\partial t} \right)^2 \quad (11)$$

De modo que:

$$T = \sum_{\alpha i k j} \frac{1}{2} m_{\alpha} \frac{\partial x_{\alpha i}}{\partial q_j} \frac{\partial x_{\alpha i}}{\partial q_k} \dot{q}_j \dot{q}_k + \sum_{\alpha i j} m_{\alpha} \frac{\partial x_{\alpha i}}{\partial q_j} \frac{\partial x_{\alpha i}}{\partial t} \dot{q}_j + \sum_{\alpha i} \frac{1}{2} m_{\alpha} \left( \frac{\partial x_{\alpha i}}{\partial t} \right)^2 \quad (12)$$

Esta expresión la podemos reescribir, lo que será de gran ayuda a la hora de realizar la animación.

$$T = \sum_{\alpha i k j} \frac{1}{2} m_{\alpha} A_{jk} \dot{q}_j \dot{q}_k + \sum_{\alpha i j} m_{\alpha} B_j \dot{q}_j + C = T_2 + T_1 + T_0 \quad (13)$$

Siendo:

$$A_{jk} = A_{kj} = \sum_{\alpha i} \frac{1}{2} m_{\alpha} \frac{\partial x_{\alpha i}}{\partial q_j} \frac{\partial x_{\alpha i}}{\partial q_k} ; B_j = \sum_{\alpha i} m_{\alpha} \frac{\partial x_{\alpha i}}{\partial q_j} \frac{\partial x_{\alpha i}}{\partial t} ; C = \sum_{\alpha i} \frac{1}{2} m_{\alpha} \left( \frac{\partial x_{\alpha i}}{\partial t} \right)^2 \quad (14)$$

Veamos un ejemplo práctico sobre esta formulación. Para ello se realizará el cambio de coordenadas de cartesianas a polares. Pensemos en una única partícula moviéndose en el plano XY:

$$T = \frac{1}{2} m (\dot{x}^2 + \dot{y}^2) \rightarrow x = r \cos \varphi ; y = r \sin \varphi \quad (15)$$

$$T_0 = 0 ; \text{No dependen explicitamente del tiempo} \quad (16)$$

$$T_1 = \sum_{\alpha i j} m_{\alpha} B_j \dot{q}_j = B_1 \dot{r} + B_2 \dot{\varphi} = 0 ; \text{Tampoco depende explicitamente de } t \quad (17)$$

$$T_2 = \sum_{\alpha i k j} \frac{1}{2} m_{\alpha} A_{jk} \dot{q}_j \dot{q}_k = \frac{1}{2} A_{11} \dot{r}^2 + A_{12} \dot{r} \dot{\varphi} + \frac{1}{2} A_{22} \dot{\varphi}^2 \quad (18)$$

$$A_{11} = m ; A_{12} = 0 ; A_{22} = mr^2 \quad (19)$$

$$T_2 = T = \frac{1}{2} (m\dot{r}^2 + mr^2\dot{\varphi}^2) \quad (20)$$

### 3.4 ECUACIONES DE LAGRANGE DESDE LA MECÁNICA LAGRANGIANA

Llegados a este punto, se debería poder identificar las coordenadas generalizadas a través del análisis de las ligaduras del sistema. Una vez conocidas, el cálculo de la energía cinética y potencial es trivial, pudiendo construir de esta forma el lagrangiano como la resta de una con respecto a la otra. Sin embargo, la verdadera potencia de este modelo de la mecánica reside en las ecuaciones conocidas como ecuaciones de Euler-Lagrange que son las que permiten conocer la evolución temporal de una partícula.

Ya que el sistema que se estudia en este proyecto es el péndulo doble y es un dispositivo ampliamente estudiado por personajes importantes en la historia de la mecánica clásica como Newton, Galileo y Foucault, desarrollaremos las ecuaciones de lagrange desde la mecánica clásica [8].

Consideremos un sistema de N partículas sometido a ciertas ligaduras que restringen los movimientos posibles de las partículas. Supongamos que el sistema se encuentra en equilibrio, de modo que las fuerzas sobre las partículas son nulas,  $F_i = 0$ . Introducimos el concepto de desplazamiento virtual  $\delta r_i$ , que es un desplazamiento infinitesimal hipotético compatible con las ligaduras, pero que no necesariamente ocurre en la realidad. Por ejemplo, si una partícula está fija en una superficie,  $\delta r_i$  debe ser tangente a esa superficie. Dado que  $F_i = 0$ , el trabajo virtual total es

$$\sum_i F_i \cdot \delta r_i = 0 \quad (21)$$

Imaginemos ahora el problema dinámico, gobernado por la ecuación de Newton  $F_i = \dot{p}_i$ . El problema dinámico lo podemos considerar como un problema estático, introduciendo la fuerza de inercia:

$$F_i - \dot{p}_i = 0 \rightarrow \sum_i (F_i - \dot{p}_i) \cdot \delta r_i = 0 \quad (22)$$

Esta es la base del Principio de D'Alembert, que transforma el problema dinámico en una forma parecida al equilibrio estático, pero que incluye las fuerzas de inercia.

Dividiendo las fuerzas en fuerzas externas  $F_i$  (gravedad o fuerzas aplicadas) y fuerzas de ligadura  $f_i$  (que mantienen las restricciones, como la tensión en una cuerda):

$$\sum_i (F_i + f_i - \dot{p}_i) \cdot \delta r_i = \sum_i F_i \cdot \delta r_i + \sum_i f_i \cdot \delta r_i - \sum_i \dot{p}_i \cdot \delta r_i = 0 \quad (23)$$

Aquí entra una propiedad clave: las fuerzas de ligadura son ortogonales a los desplazamientos virtuales porque estos desplazamientos respetan las ligaduras. Por ejemplo, si una partícula está en una superficie, la fuerza normal (ligadura) es perpendicular al desplazamiento tangente a la superficie. Por lo tanto:

$$\sum_i F_i \cdot \delta r_i - \sum_i \dot{p}_i \cdot \delta r_i = 0 \quad (24)$$

Ahora se transforman los desplazamientos virtuales en las coordenadas generalizadas. Como  $r_i = r_i(\{q_j\}; t)$ , donde  $\{q_j\}$  son coordenadas generalizadas independientes con  $j = 1, 2, \dots, s$ , se tiene:

$$\delta r_i = \sum_j \frac{\partial r_i}{\partial q_j} \delta q_j \quad (25)$$

Sustituyendo en el principio de D'Alembert:

$$\sum_i F_i \cdot \delta r_i - \sum_i \dot{p}_i \cdot \delta r_i = \sum_{ij} F_i \cdot \frac{\partial r_i}{\partial q_j} \delta q_j - \sum_{ij} m_i \ddot{r}_i \cdot \frac{\partial r_i}{\partial q_j} \delta q_j = 0 \quad (26)$$

El objetivo es expresar el segundo termino de tal forma que nos facilite la conexión con el formalismo lagrangiano. Para manipularlo, podemos aplicar una identidad matemática basada en la regla del producto de la derivada temporal. Así, el segundo termino se puede escribir como [8]:

$$m_i \ddot{r}_i \cdot \frac{\partial r_i}{\partial q_j} = \frac{d}{dt} \left( m_i \dot{r}_i \cdot \frac{\partial r_i}{\partial q_j} \right) - m_i \dot{r}_i \cdot \frac{d}{dt} \left( \frac{\partial r_i}{\partial q_j} \right) \quad (27)$$

Sin embargo, para que esta expresión sea útil y se pueda relacionarla con la energía cinética del sistema, se necesita desarrollar dos igualdades esenciales:

$$\frac{d}{dt} \left( \frac{\partial r_i}{\partial q_j} \right) = \frac{\partial \dot{r}_i}{\partial q_j} \quad (A)$$

$$\frac{\partial r_i}{\partial q_j} = \frac{\partial \dot{r}_i}{\partial \dot{q}_j} \quad (B)$$

Estas dos propiedades actúan como un puente matemático que transforma los términos dinámicos de la mecánica newtoniana en una forma que se integra naturalmente en el marco lagrangiano, como se verá en los pasos siguientes al aplicar estas igualdades.

La primera igualdad (A) indica que la derivada temporal de  $\frac{\partial r_i}{\partial q_j}$  es igual a la derivada de la velocidad  $\frac{\partial \dot{r}_i}{\partial q_j}$  lo cual nos permite sustituir directamente en la ecuación (27). Para demostrarla, se calculan ambos miembros por separado y se ve que son iguales. Se tiene lado izquierdo:

$$\frac{d}{dt} \left( \frac{\partial r_i}{\partial q_j} \right) = \sum_k \frac{\partial}{\partial q_k} \left( \frac{\partial r_i}{\partial q_j} \right) \dot{q}_k + \frac{\partial}{\partial t} \left( \frac{\partial r_i}{\partial q_j} \right) \rightarrow \sum_k \frac{\partial^2 r_i}{\partial q_k \partial q_j} \dot{q}_k + \frac{\partial^2 r_i}{\partial q_j \partial t} \quad (30)$$

Y lado derecho:

$$\dot{r}_i = \sum_k \frac{\partial r_i}{\partial q_k} \dot{q}_k + \frac{\partial r_i}{\partial t} \rightarrow \frac{\partial \dot{r}_i}{\partial q_j} = \sum_k \frac{\partial^2 r_i}{\partial q_j \partial q_k} \dot{q}_k + \frac{\partial^2 r_i}{\partial q_j \partial t} \quad (31)$$

Como las derivas segundas mixtas son conmutativas (si  $r_i$  es suficientemente suave), el resultado es idéntico al lado izquierdo

La segunda igualdad (B) establece una relación que será crucial más adelante para expresar la energía cinética en términos de  $q_j$  y  $\dot{q}_j$ . Al igual que en el caso anterior se parte de un lado de la igualdad para llegar a esa misma expresión desde el otro lado de la igualdad.

Si se parte del lado derecho:

$$\frac{\partial \dot{r}_i}{\partial \dot{q}_j} = \frac{\partial}{\partial \dot{q}_j} \left( \sum_k \frac{\partial r_i}{\partial q_k} \dot{q}_k + \frac{\partial r_i}{\partial t} \right) \quad (32)$$

El segundo termino del parentesis  $\frac{\partial r_i}{\partial t}$  no depende de  $\dot{q}_j$  por lo que su derivada es 0. Y en el primero como  $\frac{\partial r_i}{\partial q_k}$  no depende de  $\dot{q}_j$ , la derivada actúa solo sobre  $\dot{q}_k$ :

$$\frac{\partial}{\partial \dot{q}_j} \left( \sum_k \frac{\partial r_i}{\partial q_k} \dot{q}_k \right) = \sum_k \frac{\partial r_i}{\partial q_k} \frac{\partial \dot{q}_k}{\partial \dot{q}_j} \quad (33)$$

La derivada de  $\frac{\partial \dot{q}_k}{\partial \dot{q}_j}$  es 1 si  $k = j$  y 0 si  $k \neq j$ . Entonces solo queda el termino con  $k = j$

$$\frac{\partial \dot{r}_i}{\partial \dot{q}_j} = \frac{\partial r_i}{\partial q_j} \cdot 1 = \frac{\partial r_i}{\partial q_j} \quad (34)$$

Entonces, llamando fuerza generalizada a:

$$Q_j = \sum_i F_i \cdot \frac{\partial r_i}{\partial q_j} \quad (35)$$

Y usando los resultados anteriores, tenemos que la ecuación (26) se puede poner como [8]:

$$\sum_j \left( Q_j - \sum_i \left[ \frac{d}{dt} \left( m_i \dot{r}_i \cdot \frac{\partial \dot{r}_i}{\partial \dot{q}_j} \right) - m_i \dot{r}_i \cdot \frac{\partial \dot{r}_i}{\partial \dot{q}_j} \right] \right) \delta q_j = 0 \quad (36)$$

Ahora:

$$\sum_i m_i \dot{r}_i \cdot \left( \frac{\partial \dot{r}_i}{\partial \dot{q}_j} \right) = \frac{\partial}{\partial \dot{q}_j} \left( \frac{1}{2} \sum_i m_i |\dot{r}_i|^2 \right) = \frac{\partial T}{\partial \dot{q}_j} \quad (37)$$

$$\sum_i m_i \dot{r}_i \cdot \left( \frac{\partial \dot{r}_i}{\partial q_j} \right) = \frac{\partial}{\partial q_j} \left( \frac{1}{2} \sum_i m_i |\dot{r}_i|^2 \right) = \frac{\partial T}{\partial q_j} \quad (38)$$

Luego llegamos a la ecuación:

$$\sum_j \left[ Q_j - \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} \right] \delta q_j = 0 \quad (39)$$

Como los desplazamientos virtuales  $\delta q_j$  son arbitrarios e independientes, llegamos a las ecuaciones de Lagrange:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} = Q_j, \quad j = 1, 2, \dots, s \quad (40)$$

Estas ecuaciones son generales, sin hacer distinción de si las fuerzas son conservativas o no. Sin embargo, las ecuaciones que se van a usar para resolver el péndulo doble tienen una forma distinta ya que existe una energía potencial  $U$  (y, por tanto, que el sistema es conservativo). Es por ello que se va a desarrollar la ecuación (40) para llegar a las ecuaciones de Lagrange de interés.

Para ello, se parte de la suposición de que las fuerzas son conservativas  $F_i = -\nabla_{r_i} U$ . Entonces las fuerzas generalizadas son:

$$Q_j = \sum_i F_i \cdot \left( \frac{\partial r_i}{\partial q_j} \right) = - \sum_i \nabla_{r_i} U \cdot \left( \frac{\partial r_i}{\partial q_j} \right) = - \frac{\partial U}{\partial q_j} \quad (41)$$

Si la energía potencial  $U$  no depende de  $\dot{q}_j$ , entonces:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} = - \frac{\partial U}{\partial q_j} \quad (42)$$

De donde se obtiene:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} + \frac{\partial U}{\partial q_j} = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = 0 \quad (43)$$

Que son las ecuaciones de Lagrange que se utilizarán en este proyecto para resolver el sistema dinámico en cuestión.

### 3.5 SISTEMAS DINÁMICOS

Los sistemas dinámicos son modelos matemáticos que explican sistemas cuyo estado va cambiando con el tiempo. Por ejemplo, podríamos utilizar un sistema dinámico para explicar el movimiento de un péndulo, de una noria o de un coche; pero podríamos también utilizar sistemas dinámicos para explicar la evolución de una población de conejos en un ecosistema, la volatilidad en la bolsa durante unos días o el número de usuarios de un portal web en función de algunas innovaciones en el mismo [9].

### 3.5.1 SISTEMAS DINÁMICOS DISCRETOS

Desde el punto de vista matemático la evolución de algunos sistemas dinámicos puede describirse dando pequeños saltos en el tiempo y entonces hablamos de sistemas dinámicos discretos. Se trabajan matemáticamente con ecuaciones recursivas o iteraciones [9]:

$$x_{k+1} = f(x_k) \quad (44)$$

donde  $x$  representa el estado del sistema,  $k$  es el momento discreto en el tiempo, y  $f$  es una función de  $x$ . Estas relaciones permiten expresar el estado del sistema en el instante  $k+1$  como función de su estado inmediatamente anterior  $k$ .

Donde  $x$  representa el estado del sistema,  $k$  es el momento discreto en el tiempo, y  $f$  es una función de  $x$ . Estas relaciones permiten expresar el estado del sistema en el instante  $k+1$  como función de su estado inmediatamente anterior  $k$  [9].

### 3.5.2 SISTEMAS DINÁMICOS CONTINUOS

En el caso concreto de la mecánica, el cálculo nos permite relacionar entre sí las variables que gobiernan el comportamiento de los sistemas: su posición, velocidad y aceleración. Las relaciones que se establezcan entre dichas variables recogerán la dinámica del sistema. Por ejemplo, una carga eléctrica describirá una trayectoria en presencia de un campo eléctrico distinta a la que seguiría en presencia de un campo magnético. Cada tipo de campo ejercerá una fuerza característica sobre la carga y le impartirá una aceleración distinta en magnitud y dirección, lo que determinará la posición de la partícula en un instante de tiempo posterior [9]. Matemáticamente las relaciones entre posición, velocidad y aceleración se expresan mediante ecuaciones diferenciales. Éstas pueden ser ordinarias o parciales, dependiendo de la existencia de una o más variables de las que dependa cada magnitud.

Por ejemplo, la velocidad de una partícula se define como el cambio de su posición en función del tiempo mediante la ecuación diferencial ordinaria:

$$v = \dot{x} = \frac{dx}{dt} \quad (45)$$

De manera similar, su aceleración se define como el cambio de velocidad en función del tiempo que, a su vez, podemos expresar también en función del cambio de posición utilizando la ecuación (45):

$$a = \dot{v} = \frac{d}{dt} \left( \frac{dx}{dt} \right) = \frac{d^2x}{dt^2} = \ddot{x} \quad (46)$$

La trayectoria de la partícula vendrá determinada tras la integración de las ecuaciones diferenciales anteriores. De esta manera podremos expresar la posición, por ejemplo, como función del tiempo.

Las ecuaciones dinámicas integradas, nos permiten manejar el tiempo directamente como variable y no solamente a través de su efecto sobre la tasa de cambio de las magnitudes de interés, como en el caso de las ecuaciones diferenciales. Ambas expresiones deben contener la misma información para que sean intercambiables: unas resultan más cómodas para describir

la física del problema que tratamos de resolver mientras que las otras nos permiten visualizar más fácilmente las trayectorias de los sistemas estudiados. Pero esencialmente se trata de dos maneras equivalentes de decir lo mismo [9].

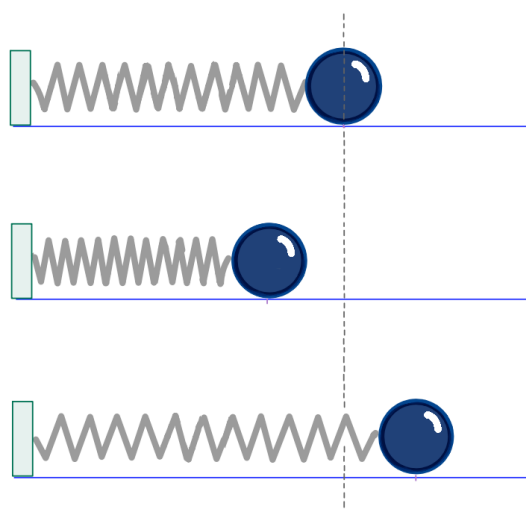
Sin embargo, existe una cierta asimetría entre los procesos de integración y diferenciación. El paso de una expresión diferencial a otra algebraica mediante la integración introduce términos adicionales, antes ausentes en las expresiones. Dichos términos no son arbitrarios, sino que deben escogerse de manera que el camino inverso, es decir, la diferenciación de la expresión integrada resulte en la expresión diferencial original y, por tanto, desaparezcan [9]. Estas características funcionales de los términos entrantes y salientes implican que dichos términos deben ser independientes de la variable con respecto a la que derivamos e integramos, el tiempo. Es decir, son constantes. Dichas constantes de integración son los grados de libertad que nos permiten fijar las condiciones iniciales de nuestro problema. La integración de la ecuación (45), por ejemplo, nos permitirá fijar una posición inicial de nuestra partícula (que no necesariamente coincidirá con el origen de coordenadas que utilicemos), mientras que la integración de la ecuación (46) nos permitirá fijar la velocidad inicial con la que se desplazaba la partícula antes de sufrir el efecto del campo eléctrico o magnético [9].

### 3.5.3 SISTEMAS LINEALES

En los sistemas lineales, pequeñas diferencias en las condiciones iniciales conducen a pequeñas diferencias en el estado final después de la evolución. Un sistema dinámico lineal cumple que [9]:

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y), \quad \alpha, \beta \in R \quad (47)$$

Cuando el sistema dinámico es lineal, se cumple el principio de superposición para las soluciones, se pueden separar las distintas variables aplicando la ecuación (47) y, sobre todo, tiene solución analítica, es decir, una expresión explícita de  $x$  en función del tiempo  $x(t)$ .



*Figura 4 Ejemplo de OSCILADOR ARMÓNICO SIMPLE: Muelle con masa en su extremo.*

---

La ecuación de movimiento de un oscilador armónico simples es:

$$m\ddot{x} + kx = 0 \tag{48}$$

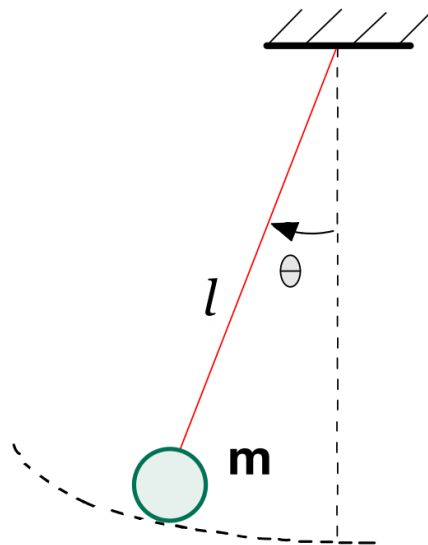
Si  $x_1(t)$  y  $x_2(t)$  son soluciones del sistema, entonces:

$$\begin{aligned} m\ddot{x}_1 + kx_1 &= 0 ; m\ddot{x}_2 + kx_2 = 0 \\ m((\ddot{x}_1 + \ddot{x}_2) + k(x_1 + x_2)) &= 0 \end{aligned} \tag{49}$$

Esto demuestra que  $x_1 + x_2$  también es solución, cumpliendo la aditividad.

### 3.5.4 SISTEMAS NO LINEALES

En los sistemas no lineales, pequeñas diferencias en las condiciones iniciales pueden conducir a grandes diferencias en el estado final después de la evolución. Este fenómeno, es absolutamente crucial para la predicción del futuro del sistema y se llama sensibilidad a las condiciones iniciales. La diferencia entre dos estados iniciales del sistema, a menudo llamada incertidumbre o error por estar asociada a la máxima precisión con la que podemos identificar dichos estados de manera diferenciada [9].



*Figura 5 Ejemplo sistema no lineal: Péndulo simple sin aproximaciones*

La ecuación de movimiento de un péndulo simple es:

$$\ddot{\theta} + \frac{g}{L}\sin(\theta) = 0 \tag{50}$$

Supongamos que  $\theta_1(t)$  y  $\theta_2(t)$  son soluciones, entonces:

$$\ddot{\theta}_1 + \frac{g}{L}\sin(\theta_1) = 0 ; \ddot{\theta}_2 + \frac{g}{L}\sin(\theta_2) = 0 \tag{51}$$

Para que  $\theta_1$  y  $\theta_2$  sean solución, debería cumplirse que:

$$\sin(\theta_1 + \theta_2) = \sin \theta_1 + \sin \theta_2 \quad (52)$$

Sin embargo, esto no ocurre porque el seno no es una función lineal.

### 3.5.4.1 LINEALIZACIÓN

En la historia de las ciencias físicas se ha resuelto una enorme variedad de problemas mediante la técnica de aproximación denominada linealización: si el sistema dinámico que representa matemáticamente el problema no tiene solución analítica (solución por simulación numérica) porque no es lineal, podemos encontrar a menudo un conjunto de estados en los que el sistema puede considerarse aproximadamente lineal; se simplifican las ecuaciones correspondientes, linealizándolas, de modo que puedan tener solución analítica. [9]

Si se está interesado en estudiar el comportamiento del sistema en torno a un punto de equilibrio, se puede aproximar por una ecuación lineal si  $x$  se mantiene en un rango pequeño alrededor de dicho punto.

La clave es que si  $x$  es muy pequeño ( $|x| \ll 1$ ), la función  $f(x)$  puede ser aproximada por una expansión en serie de Taylor en torno a un punto de equilibrio  $x_0$ :

$$f(x) = f(x_0) + \left. \frac{df}{dx} \right|_{x_0} (x - x_0) + \frac{1}{2} \left. \frac{d^2f}{dx^2} \right|_{x_0} (x - x_0)^2 + \dots \quad (53)$$

Si elegimos un punto de equilibrio, es decir, un punto donde  $f(x_0) = 0$  (el sistema no cambia en el tiempo), podemos despreciar los términos cuadráticos y de orden superior, dejando solo el término lineal:

$$x \sim A(x - x_0); \quad A = \left. \frac{df}{dx} \right|_{x_0} \quad (54)$$

donde  $A$  es una matriz jacobiana (si el sistema es multidimensional) o una derivada en el caso escalar. En este caso el sistema tiene siempre solución analítica.

Grandes teorías y técnicas de considerable éxito contienen ecuaciones linealizadas en física, química, economía, biología, ingeniería, etc [9]. Por ejemplo, en el caso del oscilador armónico o del péndulo simple, se asume a menudo que las amplitudes de las oscilaciones son pequeñas, lo que permite linealizar el problema.

Ya que una parte muy importante de este proyecto recae sobre el péndulo, se demostrará este caso para una mejor comprensión.

La ecuación de movimiento viene descrita por la ecuación (50), una ecuación diferencial no lineal como ya se ha visto. Sin embargo, si las oscilaciones son pequeñas, es decir  $|\theta| \ll 1$ , se puede usar la aproximación de Taylor para el seno alrededor de  $\theta=0$  que es un punto de equilibrio:

$$\sin \theta \sim \theta \rightarrow \ddot{\theta} + \frac{g}{L} \theta = 0 \rightarrow \ddot{\theta} + \omega^2 \theta; \quad \omega = \sqrt{\frac{g}{L}} \quad (55)$$

Es una ecuación diferencial de segundo orden con coeficientes constantes y raíces imaginarias. La solución general es de la forma:

$$\theta(t) = C_1 e^{i\omega t} + C_2 e^{-i\omega t} \quad (56)$$

Usando la identidad de Euler:

$$e^{i\omega t} = \cos(\omega t) + i \sin(\omega t) \rightarrow \theta(t) = A \cos(\omega t) + B \sin(\omega t) \quad (57)$$

Donde A y B son constantes de integración que se determinan con las condiciones iniciales. Alternativamente, podemos escribir la solución en la forma más compacta:

$$\theta(t) = \theta_0 \cos(\omega t + \varphi) \quad (58)$$

Siendo  $\theta_0$  la amplitud de oscilación, que depende de las condiciones iniciales y  $\varphi$  la fase inicial, determinada por la condición inicial de  $\theta$ .

### 3.5.5 SISTEMAS CONSERVATIVOS

Los sistemas conservativos son aquellos en los que la energía total se conserva. Esto significa que la energía se puede transformar de una forma a otra dentro del sistema, pero su cantidad total permanece constante. En un sistema conservativo, la energía potencial y la energía cinética están relacionadas y se pueden intercambiar entre sí. Un ejemplo común de un sistema conservativo es un péndulo simple. A medida que el péndulo oscila hacia arriba y hacia abajo, la energía potencial se convierte en energía cinética y viceversa [10].

La segunda ley de Newton o ley fundamental de la física establece que la fuerza aplicada a un sistema mecánico es igual al producto de su masa por la aceleración:

$$m \left( \frac{d^2 x}{dt^2} \right) = f \quad (59)$$

Donde  $f = f\left(t, x, \frac{dx}{dt}\right)$ . Se dice que un sistema es conservativo si la fuerza aplicada sólo depende de la posición  $x$ :  $f = f(x)$ . Normalizando la masa ( $m = 1$ ), la ecuación de la segunda ley de Newton da lugar a una ecuación diferencial de segundo orden del tipo:

$$\ddot{x} = f(x) \quad (60)$$

La ecuación se convierte en el sistema:

$$\begin{cases} \dot{x} = y \\ \dot{y} = f(x) \end{cases} \quad (61)$$

definido en el dominio  $\Omega = I \times R \subseteq R^2$ , donde  $I \subseteq R$  es un intervalo abierto en el que  $f: I \rightarrow R$  es una función continua y diferenciable.

Eliminamos la dependencia explícita del tiempo  $t$  combinando las ecuaciones del sistema (61). Aplicamos la regla de la cadena para expresar la derivada de  $y$  respecto a  $x$ :

$$\frac{dy}{dx} = \frac{dy}{dt} \frac{dt}{dx} = f(x) \frac{1}{y} \quad (62)$$

La ecuación (62) es una ecuación diferencial ordinaria de variables separables, cuya solución nos permitirá caracterizar las trayectorias del sistema en el plano  $(x,y)$ :

$$\int y dy = \int f(x) dx \rightarrow \frac{1}{2}y^2 + C_1 = F(x) + C_2 \quad (63)$$

Para conectar esta expresión con una interpretación física, definimos la energía potencial  $U(x)$  como:

$$U(x) = - \int_{x_0}^x f(s) ds \quad (64)$$

Si agrupamos todas las constantes de integración en una única y la llamamos  $E$ , llegamos a la siguiente expresión:

$$E = \frac{1}{2}y^2 + U(x) \quad (65)$$

Volviendo a la interpretación física, el término  $\frac{1}{2}y^2$  representa la energía cinética del sistema de masa  $m = 1$  y  $U(x)$  representa la energía potencial. La ecuación de la trayectoria significa que cada trayectoria corresponde a un nivel de energía constante  $E$ , lo que se traduce en que la energía total se conserva a lo largo de cada trayectoria.

### 3.5.6 SISTEMAS DISIPATIVOS

Los sistemas disipativos son aquellos en los que la energía se pierde gradualmente debido a la fricción o la resistencia en el sistema. En estos sistemas, la energía se disipa en forma de calor u otras formas de energía no utilizables. Un ejemplo de sistema disipativo es un péndulo doble no ideal. A medida que los eslabones oscilan la energía se disipa debido al rozamiento de las articulaciones y del propio sistema con el aire. Para introducir este sistema se utilizarán los conceptos y ecuaciones de la mecánica clásica descritos en el apartado (3.4). Para tratar fuerzas no conservativas en mecánica lagrangiana, se pueden abordar tres enfoques distintos [11]:

- Expandir el número de grados de libertad, redefiniendo el sistema.
- Añadir fuerzas no conservativas directamente en las ecuaciones de movimiento de Lagrange como fuerzas generalizadas. Si las fuerzas disipativas son proporcionales a la velocidad se puede usar la función de disipación de Rayleigh.
- Modificar la función de Lagrange agregando términos adicionales.

El caso más general es el segundo y por tanto será el que se aborde en esta sección. Cuando la fuerza disipativa que aparece en el sistema depende linealmente de la velocidad como el rozamiento con un fluido ( $\vec{F} = -b \cdot \vec{v}$ ), se puede incluir las ecuaciones de Lagrange mediante la función de disipación de Rayleigh, definida como:

$$R(\dot{q}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n b_{ij} \dot{q}_i \dot{q}_j \quad (66)$$

Donde  $b_{ij}$  son coeficientes de rozamiento intrínsecas de cada fluido. Este método ha sido generalizado mediante la definición de una función de disipación de Rayleigh dependiente linealmente de la velocidad [11]:

$$F_i^f = -\frac{\partial R(q, \dot{q})}{\partial \dot{q}} \quad (67)$$

donde la función de disipación de Rayleigh generalizada  $R(q, \dot{q})$  satisface la ecuación general de la mecánica lagrangiana [11]:

$$\frac{\delta L}{\delta q} - \frac{\partial R}{\partial \dot{q}} = 0 \quad (68)$$

Cuando las fuerzas no son conservativas, dichas ecuaciones se derivan también de un principio extremal, específicamente el principio de Hamilton para sistemas generales, expresado como [8]:

$$\delta S[\{q_j\}] = \delta \int_{t_1}^{t_2} dt (T + V) = 0 \quad (69)$$

Donde T es la energía cinética y V es el virial, definido como:

$$V = \sum_i F_i \cdot r_i \quad (70)$$

Partiendo de este punto, calculamos la variación de la acción S:

$$\begin{aligned} \delta S &= \delta \int_{t_1}^{t_2} dt T + \delta \int_{t_1}^{t_2} dt \sum_i F_i \cdot r_i = \delta \int_{t_1}^{t_2} dt T + \int_{t_1}^{t_2} dt T + \int_{t_1}^{t_2} dt \sum_i F_i \cdot \delta r_i = \\ &= \delta \int_{t_1}^{t_2} dt T + \int_{t_1}^{t_2} dt \sum_j Q_j \delta q_j \end{aligned} \quad (71)$$

El último paso se deriva de la definición de las fuerzas generalizadas (41):

$$\sum_j Q_j \delta q_j = \sum_{ij} F_i \frac{\partial r_i}{\partial q_j} \delta q_j = \sum_i F_i \cdot \sum_j \frac{\partial r_i}{\partial q_j} \delta q_j = \sum_i F_i \cdot \delta r_i \quad (72)$$

Dado que  $T = T(\{q_j, \dot{q}_j\}; t)$ , aplicamos la técnica variacional estándar al primer término de (71):

$$\delta \int_{t_1}^{t_2} dt T = \int_{t_1}^{t_2} dt \sum_j \left[ \frac{\partial T}{\partial q_j} - \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) \right] \delta q_j \quad (73)$$

Sumando ambos términos, obtenemos [8]:

$$\int_{t_1}^{t_2} dt \sum_j \left[ \frac{\partial T}{\partial q_j} - \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) + Q_j \right] \delta q_j = 0 \quad (74)$$

Puesto que las variaciones  $\delta q_j$  son independientes, el integrando debe anularse para cada j, resultando en las ecuaciones de Lagrange generales para sistemas con fuerzas no conservativas:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} = Q_j \quad (75)$$

Las fuerzas disipativas, vendrán dadas por un término adicional al lado derecho, resultando en la ecuación final:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \left( \frac{\partial L}{\partial q_i} \right) + \left( \frac{\partial R_i}{\partial \dot{q}_i} \right) = Q_i, \quad i = 1, 2, \dots, n \text{ grados de libertad} \quad (76)$$

### 3.6 TEORIA DEL CAOS

La Teoría del Caos una rama de las matemáticas, la física y otras ciencias que trata ciertos tipos de sistemas dinámicos, con la particularidad de ser muy sensibles a las variaciones en las condiciones iniciales. Pequeñas variaciones en dichas condiciones iniciales pueden implicar grandes diferencias en el comportamiento futuro, haciendo complicada la predicción a largo plazo [12].

La metáfora de la “mariposa” en la teoría del caos subraya cómo una variación ínfima en el estado inicial de un sistema puede desencadenar transformaciones de gran alcance. Es como si el batir de sus alas, casi imperceptible, se amplificara progresivamente hasta provocar sucesos extraordinarios. A pesar de que estos sistemas siguen reglas deterministas (su evolución está teóricamente predeterminada por sus condiciones de partida), su comportamiento real escapa a toda predicción a largo plazo.

El caos sería un punto intermedio entre lo estrictamente determinista y lo aleatorio. Las principales características de los sistemas caóticos es que son impredecibles en períodos amplios de tiempo y muy sensibles a las condiciones iniciales del sistema. Una vez iniciado con valores específicos, el futuro sistema podría desarrollarse de forma totalmente diferente si comenzase bajo condiciones ligeramente diferentes.

#### 3.6.1 SISTEMAS CAÓTICOS

##### 3.6.1.1 DIAGRAMA DE FASES

Una manera práctica de describir la evolución de un sistema como el péndulo simple consiste en hacer un mapa con los valores de los parámetros involucrados en la evolución del sistema. ¿Qué valores de posición y velocidad puede tomar la masa de un péndulo simple en acción? Evidentemente no cualesquiera, sino únicamente aquellos que sean compatibles con su dinámica. Ese mapa de valores posibles de los parámetros del sistema se conoce como diagrama de fases [9]. En el caso del péndulo simple, el diagrama de fases representa parejas de valores de posición y velocidad de la masa del péndulo en cada instante de tiempo. De esta forma, cada punto en el diagrama corresponde a la fotografía del sistema en un cierto instante de tiempo, caracterizado por que la masa del péndulo se encuentra en una determinada posición viajando a una cierta velocidad.

La *Figura 6* representa el diagrama de fases de un péndulo simple con una familia de trayectorias posibles [\(Anexo-Codigos-02\)](#). La elección de una curva particular viene impuesta por las condiciones iniciales del problema. Por ejemplo, si se desplaza la masa del péndulo una distancia

$x_1$  y la liberamos en el instante  $t = 0$ , de manera que  $v = 0$ , el punto indicado en el diagrama con coordenadas  $(x_1, 0)$  será el fotograma de dicho sistema en el instante inicial.

La evolución del sistema estará determinada a partir de ese instante por los puntos comprendidos en la curva correspondiente. Por ejemplo, en la simulación la flecha representa que el péndulo, al ser soltado desde  $x_1$ , comienza a caer hacia la posición de equilibrio ( $\theta=0$ ), y su velocidad angular aumenta en la dirección positiva. Se puede predecir también qué ocurriría con el sistema si se desplazara la masa del péndulo un poco más allá de  $x_1$  en el instante inicial: la nueva trayectoria elíptica será entonces un poco más amplia, de manera que el valor máximo de la velocidad que alcanza la masa en su movimiento al pasar por el punto de equilibrio ( $x = 0$ ) será algo mayor también. Se dice entonces que se trata de un sistema determinista.

Los diagramas de fases permiten identificar rápidamente propiedades importantes de la evolución de los sistemas dinámicos. Por ejemplo, si las trayectorias son cerradas el sistema describirá un movimiento periódico, es decir, parejas de valores de posición y velocidad se repetirán periódicamente con el paso del tiempo. Si, por el contrario, las trayectorias en el diagrama de fases son abiertas, el movimiento del sistema será aperiódico. Existen situaciones en las que las trayectorias tienden a converger en un punto del diagrama de fases al que, en un lapso de tiempo suficientemente largo, el sistema terminará aproximándose arbitrariamente cerca [9]. Estos puntos particulares se conocen como atractores y se asocian a situaciones de equilibrio al que determinados sistemas evolucionan de manera natural. Por ejemplo, el movimiento de un péndulo amortiguado termina por detenerse en su posición de equilibrio. En este caso, las trayectorias del espacio de fases son espirales que convergen hacia el origen  $(0,0)$ , el atractor de este sistema.

Por otro lado, si se tiene un sistema caótico, las trayectorias no tendrán una geometría clara ni serán perfectamente regulares. Además, se podrá apreciar superposición de trayectorias con condiciones iniciales ligeramente diferentes. Para demostrar esta divergencia y como trayectorias que comienzan muy cerca una de otras, acaban por separarse con el tiempo (mezcla topológica), formando patrones complejos, se muestra en la siguiente página una simulación en Python ([Anexo-Codigos-03](#)) del diagrama de fases de un péndulo doble (Figura 7).

La complejidad de las trayectorias con bucles y superposiciones es un indicio del caos del sistema. De esta forma cada vez que se vea un diagrama de fases que no sigue un patrón simple y predecible, se estará frente a un sistema caótico. Cabe destacar que, aunque solo se muestre el diagrama del primer péndulo, el comportamiento caótico es el resultado de la interacción no lineal entre los dos péndulos, ya que el segundo influye en el movimiento del primero a través de términos de acoplamiento descritos en las ecuaciones de movimiento. La concentración de líneas que se observa en  $\theta = 0$ , representa la presencia de un atractor y es que se trata de un sistema disipativo (se ha introducido pérdida energética por rozamiento de los rodamientos y con el fluido circundante) que genera una pérdida energética al péndulo hasta que transcurrido un cierto tiempo alcanza el equilibrio en  $\theta = 0$ .

Victor Manuel Suárez Ferrer

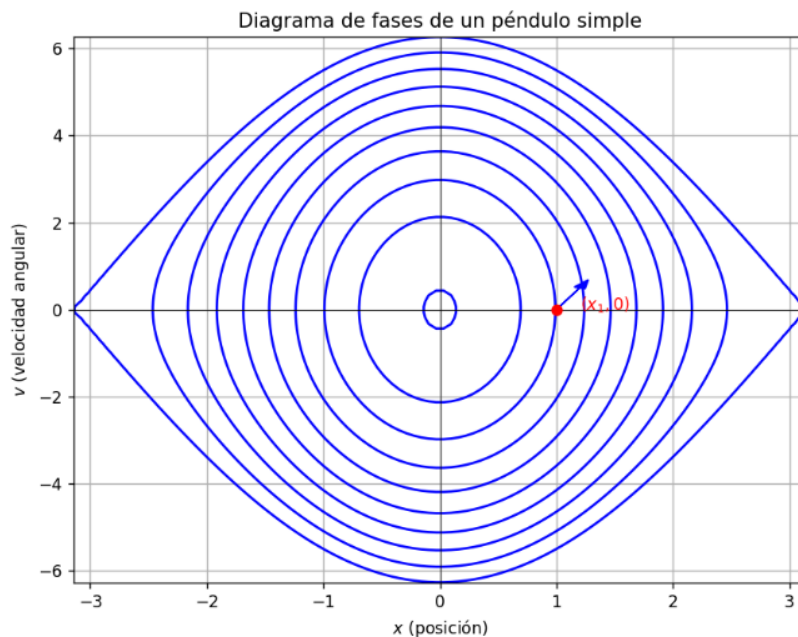


Figura 6 Diagrama de fases de un péndulo simple calculado a partir de sus ecuaciones dinámicas mediante un programa en Python, mostrando trayectorias elípticas que representan un movimiento periódico en el espacio de posición y velocidad angular.

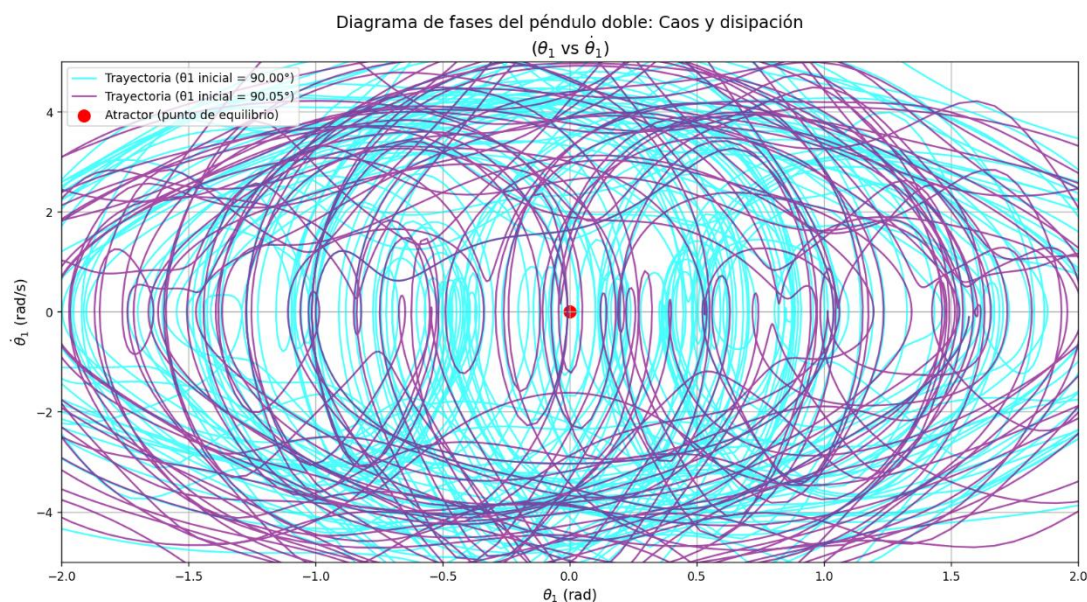


Figura 7 Diagrama de fases de un péndulo doble calculado a partir de sus ecuaciones dinámicas mediante un programa en Python. Enfrentamiento de trayectorias con condiciones iniciales prácticamente iguales para demostrar el caos. Atractor en  $(0,0)$ .

### 3.7 REDES NEURONALES

Unos de los métodos más exitosos del aprendizaje automático son las redes neuronales artificiales. Son un método de aprendizaje y procesamiento inspirado en la forma en que funciona el sistema nervioso de los animales. Del mismo modo, las redes neuronales artificiales están compuestas por unidades de procesamiento llamadas neuronas, organizadas en capas interconectadas. Cada neurona recibe entradas de información, las procesa mediante una cierta función de activación y transmite la salida a las neuronas de la capa siguiente. Existe gran cantidad de formas en las que se pueden conectar estas capas y hay mucha libertad a la hora de elegir como se transmite la información entre ellas. Esto hace que el diseño de redes neuronales sea un campo desafiante y en evolución.

Toda red neuronal consta de capas de nodos o neuronas artificiales: una capa de entrada, una o varias capas ocultas y una capa de salida. Cada nodo se conecta a los demás y tiene su propia ponderación y umbral asociados. Si la salida de cualquier nodo individual está por encima del valor umbral especificado, ese nodo se activa y envía datos a la siguiente capa de la red. De lo contrario, no se pasa ningún dato a la siguiente capa de la red [13].

Para entender cómo funciona una red neuronal de manera sencilla, es importante conocer la fórmula bajo la cual operan [14]:

$$\sum w_i x_i + sesgo = w_1 x_1 + w_2 x_2 + w_3 x_3 + sesgo \quad (77)$$

$$salida f(x) = \begin{cases} 1 & \text{si } \sum w_i x_i \geq 0 \\ 0 & \text{si } \sum w_i x_i < 0 \end{cases} \quad (78)$$

Los distintos  $w_i$ , reciben el nombre de pesos o ponderaciones. Estas ponderaciones ayudan a determinar la importancia de cualquier variable, ya que las más grandes contribuyen de forma más significativa a la salida en comparación con otras entradas. A continuación, todas las entradas se multiplican por sus respectivas ponderaciones y se suman. Después, la salida se pasa a través de una función de activación, que determina la salida. Si esa salida supera un umbral determinado, se «dispara» (o activa) el nodo, pasando los datos a la siguiente capa de la red. Esto da como resultado que la salida de un nodo se convierta en la entrada del siguiente nodo [14].

Para entender el funcionamiento de forma más sencilla, se expondrá un ejemplo simple:

Pensemos en una empresa de discos de freno que quiere determinar si su producto está en buen estado para su venta o no (SI=0, NO=1). Supongamos que hay tres factores que influyen en la decisión:

1. Los discos tienen las medidas correctas (Si=1, NO=0)
2. Los discos tienen alguna anomalía (Si=1, NO=0)
3. La cadena de montaje se ha detenido al menos una vez durante la elaboración de este. (Si=1, NO=0)

Entonces, supongamos lo siguiente, dándonos las siguientes entradas:

- $X_1 = 1$ , ya que los discos tienen las medidas correctas
- $X_2 = 0$ , ya que no tienen ninguna anomalía
- $X_3 = 1$ , ya que hubo un problema con el montaje

Ahora, tenemos que asignar algunas ponderaciones para determinar la importancia. Unas ponderaciones mayores significan que determinadas variables son más importantes para la decisión o el resultado:

- $W_1 = 6$ , ya que un disco con unas medidas incorrectas no sirve
- $W_2 = 4$ , ya que una pequeña anomalía como una grieta puede ocasionar problemas muy graves
- $W_3 = 2$ , ya que no es un factor determinante en la calidad del disco

Ahora supongamos un valor de umbral de 6, lo que supone un sesgo de -6, que es lo mismo que exigir que la suma ponderada supere 6 para activarse. Con todas las entradas, podemos empezar a introducir valores en la fórmula para obtener la salida deseada:

$$\sum w_i x_i + sesgo = (1 \cdot 6) + (0 \cdot 4) + (1 \cdot 2) - 6 = 2 \quad (79)$$

Por lo tanto, la salida sería 1, lo que quiere decir que nuestro disco no está en buen estado para su venta.

### 3.7.1 REDES NEURONALES RECURRENTE (RNNs)

Las RNNs son redes neuronales diseñadas para procesar datos donde el orden importa, como series temporales. A diferencia de otras redes, las RNNs tienen conexiones que permiten recordar información pasada, lo que las hace ideales para tareas como predecir movimientos.

Las RNNs se distinguen de otras redes neuronales por su estructura recurrente, que permite que las salidas de un paso de tiempo se utilicen como entradas en el siguiente, manteniendo un estado oculto que actúa como una memoria de los estados previos. Las RNNs procesan datos secuenciales mediante una unidad recurrente que actualiza su estado oculto en cada paso de tiempo  $t$ .

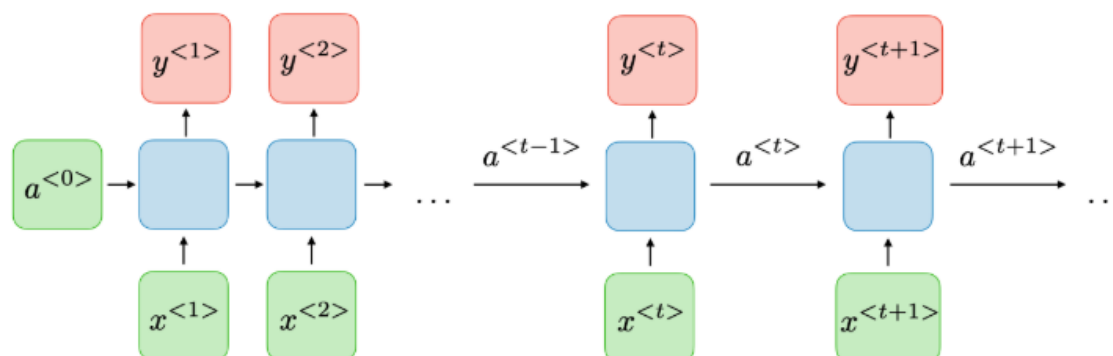


Figura 8 Arquitectura de una red RNN tradicional [15].

Matemáticamente, la activación  $a^{(t)}$  y la salida  $y^{(t)}$  se calculan como [15]:

$$\begin{aligned} a^{(t)} &= g_1(W_{aa}a^{(t-1)} + W_{ax}x^{(t)} + b_a) \\ y^{(t)} &= g_2(W_{ya}a^{(t)} + b_y) \end{aligned} \quad (80)$$

La primera ecuación calcula el estado oculto  $a^{(t)}$ , que actúa como memoria. Combina el estado anterior  $a^{(t-1)}$  (ponderado por el peso de  $W_{aa}$ ), la entrada actual  $x^{(t)}$  (ponderada por el peso  $W_{ax}$ ), y un sesgo  $b_a$ . La función  $g_1$  (como tanh) introduce no linealidad para modelar relaciones complejas. La salida  $y^{(t)}$  funciona de la misma forma, transformando el estado oculto. Esta arquitectura es idónea modelar sistemas con movimiento continuo, incluyendo sus variantes LSTM y GRU, lo que las hace directamente aplicables a la predicción de ángulos del péndulo.

Pero a pesar de sus ventajas, las RNNs presentan limitaciones que deben considerarse. Estas redes sufren del problema conocido como gradiente desvaneciente. En la Ec. (80) los pesos que se observan se ajustan por cálculo de gradientes, el problema ocurre cuando estos gradientes se vuelven extremadamente pequeños a medida que se propagan hacia pasos de tiempo anteriores, lo que dificulta capturar dependencias a largo plazo, y su computación secuencial puede ser lenta [15]. Para superar esta limitación, se desarrollaron las LSTMs, una variante avanzada de las RNNs.

### 3.7.2 REDES LSTM

Las Redes de Memoria a Largo Plazo (LSTM, Long Short-Term Memory) son una evolución avanzada de las redes RNNs, diseñadas específicamente para superar las limitaciones de estas últimas en el aprendizaje de dependencias a largo plazo en datos secuenciales.

Las RNNs, como se explicó anteriormente, procesan datos secuenciales mediante un estado oculto que se actualiza iterativamente, pero sufren del problema del gradiente desvaneciente, donde los gradientes se reducen exponencialmente durante la retropropagación a través del tiempo, dificultando el aprendizaje de dependencias a largo plazo [15]. Este fenómeno, causado por la multiplicación repetida de pesos y derivadas de funciones de activación no lineales como la tangente hiperbólica, limita la capacidad de las RNNs para modelar secuencias extensas donde eventos iniciales influyen en resultados distantes. Las LSTM se desarrollaron para mitigar esta limitación, introduciendo una arquitectura que permite preservar información relevante a lo largo de múltiples pasos de tiempo, lo que las hace idóneas para tareas teóricas que requieren memoria extendida, como el modelado de secuencias complejas.

La arquitectura de las LSTM se distingue por la incorporación de un estado de celda y tres puertas reguladoras: la puerta de olvido, la puerta de entrada y la puerta de salida. El estado de celda actúa como una memoria a largo plazo, permitiendo que la información fluya a través de la secuencia con modificaciones mínimas, mientras que las puertas controlan selectivamente qué información se retiene, se añade o se utiliza como salida.

Matemáticamente, el funcionamiento de una LSTM en un paso de tiempo  $t$  se describe mediante un conjunto de ecuaciones que reflejan la interacción entre el estado de celda, las puertas y el

estado oculto. Dado un input  $x_t$ , el estado oculto anterior  $h_{t-1}$  y el estado de celda anterior  $c_{t-1}$ , las operaciones son las siguientes [16]:

- Puerta de Olvido ( $f_t$ ):

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (81)$$

La puerta de olvido decide qué información del estado de celda anterior ( $c_{t-1}$ ) debe descartarse u olvidarse. Es como un filtro que determina qué partes de la memoria a largo plazo ya no son relevantes para el contexto actual. Donde  $\sigma$  es una función que transforma la suma ponderada en valores entre 0 (olvidar) y 1 (retener).

- Puerta de Entrada ( $i_t$ ) y Vector Candidato ( $\tilde{C}_t$ ):

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (82)$$

La puerta de entrada determina qué partes del input actual son lo suficientemente importantes como para añadirse al estado de celda. Actúa como un filtro que evalúa la relevancia de la nueva información.

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (83)$$

Este vector propone los valores potenciales que podrían añadirse al estado de celda. Es como un borrador de la nueva información que la red considera incorporar. Aplica la función tangente hiperbólica, que produce valores entre  $-1$  y  $1$ . Esto normaliza los valores para mantenerlos estables.

Ahora que la red ha decidido que olvidar y que añadir, es hora de actualizar el estado de celda  $C_t$ . Para ello usa la función:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (84)$$

Esta función multiplica elemento a elemento la puerta de olvido con el estado de celda anterior. Esto conserva solo las partes de la memoria que la puerta de olvido marcó como relevantes y descarta las irrelevantes. A esto se le suma la multiplicación elemento a elemento de la puerta de entrada con el vector candidato. Esto añade la nueva información, escalada según la relevancia asignada por la puerta de entrada. La suma de ambos términos produce el nuevo estado de celda.

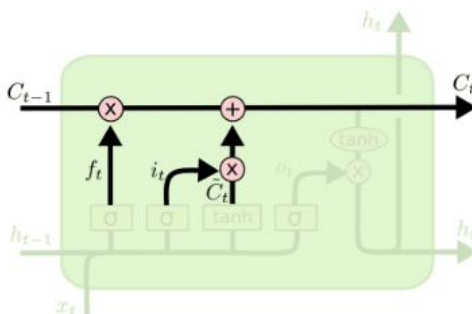


Figura 9 Esquema de la capacidad de la red LSTM para mantener información a largo plazo [16].

Estas redes tienen una especie de filtro que decide que partes de este estado actual cuentan con información relevante y por tanto deben emitirse como salida de la celda. Esto recibe el nombre de puerta de salida y se rige por la siguiente ecuación:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (85)$$

Finalmente se genera la salida del estado oculto que será la señal de salida final o la información que llegue a la siguiente celda.

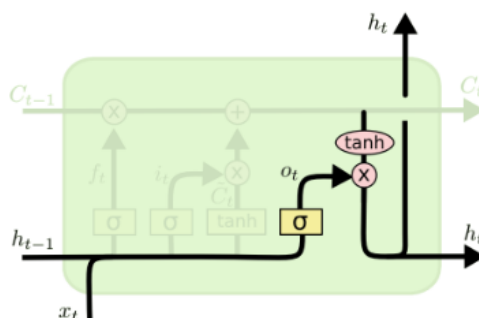


Figura 10 Representación de la puerta de salida [16].

---

### 3.8 VISIÓN ARTIFICIAL

La visión artificial o visión por computador intenta emular la capacidad de algunos seres vivos para ver una escena (imagen), entenderla y actuar en consecuencia, con el fin de producir información numérica o simbólica para que puedan ser tratados por un ordenador. Si la IA permite que los ordenadores piensen, la visión artificial les permite ver, observar y comprender.

La visión artificial necesita una gran cantidad de datos. Ejecuta análisis de datos una y otra vez hasta que discierne distinciones y finalmente reconoce imágenes. Por ejemplo, para un entrenamiento de modo que sea capaz de reconocer coches, necesita recibir grandes cantidades de imágenes de coches en distintos escenarios y elementos relacionados con coches para aprender las diferencias y reconocer el objeto.

Para lograrlo, se emplean dos tecnologías esenciales: un tipo de machine learning llamado aprendizaje profundo y una red neuronal convolucional (CNN).

El machine learning emplea modelos algorítmicos que permiten que un ordenador se enseñe a sí mismo sobre el contexto de los datos visuales. Si se alimentan suficientes datos a través del modelo, el ordenador "mirará" los datos y aprenderá a distinguir una imagen de otra. Los algoritmos permiten que la máquina aprenda por sí misma, en lugar de que alguien la programe para reconocer una imagen [17].

Una Red Neuronal Convolucional (CNN) permite a un modelo de machine learning o aprendizaje profundo procesar imágenes descomponiéndolas en píxeles etiquetados. Con estas etiquetas, realiza convoluciones (una operación matemática que combina dos funciones para generar una

tercera) y predice qué está observando. La red ajusta la precisión de sus predicciones mediante iteraciones hasta mejorar su reconocimiento, logrando identificar imágenes de forma similar a los humanos.

### 3.9 POSE DETERMINATION

La estimación de la pose es una tarea que consiste en identificar la ubicación de puntos específicos en una imagen, normalmente denominados puntos clave. Los puntos clave pueden representar diversas partes del objeto, como articulaciones, puntos de referencia u otras características distintivas. Las ubicaciones de los puntos clave suelen representarse como un conjunto de 2D  $[x, y]$  o 3D  $[x, y, visible]$  coordenadas [18].

Pose determination puede abordarse desde dos perspectivas principales:

- **Top-down:** Este modelo primero detecta el objeto completo (por ejemplo, el péndulo completo) y luego se localizan los puntos clave dentro de su región (como podrían ser las masas de los extremos de los eslabones). Este enfoque es común en modelos como YOLO cuando se combina con detección de objetos [19].
- **Bottom-up:** Por otro lado, en este modelo primero se identifican todos los puntos clave en la imagen y luego se agrupan para formar la pose del objeto. Este método es más común en modelos como OpenPose, pero menos eficiente para objetos específicos como un péndulo [15].

#### 3.9.1 POSE DETERMINATION CON YOLO

YOLO (You Only Look Once) es un modelo de visión artificial que ha evolucionado para incluir capacidades de estimación de pose en sus versiones recientes. Los modelos de pose de YOLO extienden la detección de objetos al predecir un conjunto de keypoints asociados a cada objeto detectado, con aplicaciones que van desde humanos hasta objetos personalizados [18]. Esto es gracias a que todos los modelos que ofrece YOLO han sido preentrenados con 80 clases distintas (personas, maletas, coches, etc) y esto ofrece un punto de partida muy bueno para aplicaciones específicas como la que se pretende abordar en este documento.

El funcionamiento de YOLO para pose estimation sigue estos pasos:

- **División en cuadrícula:** La imagen se divide en una cuadrícula. Cada celda predice si contiene un objeto y sus características asociadas [19].
- **Detección del objeto:** Para cada celda, se predicen bounding boxes con coordenadas  $(x, y, anchura, alto)$  y una confianza. Esto localizaría el péndulo doble en la imagen, por ejemplo.
- **Predicción de keypoints:** Dentro del bounding box, se estiman las coordenadas de los keypoints relativas a las dimensiones del cuadro. YOLO ofrece la ventaja de que las coordenadas se normalizan entre 0 y 1 respecto al tamaño del bounding box, lo que facilita la generalización [18].

- **Post-procesamiento:** Se aplica un umbral de confianza para eliminar predicciones redundantes, dejando solo las más precisas [20]. Supongamos que YOLO predice un bounding box con  $c=0.8$  y tres keypoints con  $v_0 = 0.9$ ,  $v_1 = 0.7$ ,  $v_2 = 0.3$ . Si el umbral es 0.5, el keypoint 2 se descartaría, pero el bounding box y los keypoints 0 y 1 se mantendrían.

Es muy importante tener en cuenta que YOLO requiere una estructura de directorios específica para organizar los datos de manera eficiente y garantizar que el modelo pueda acceder a ellos sin complicaciones durante las fases de entrenamiento, validación y prueba. Esta estructura incluye:

**Carpetas y Subcarpetas:** Como train, val, test, donde se encuentra toda la información que va a usar YOLO para su modelo, dividida en subcarpetas para las imágenes y las etiquetas. El directorio "train" contiene las imágenes y etiquetas empleadas para entrenar el modelo YOLO, "val" incluye datos para validar y ajustar hiperparámetros durante el entrenamiento, y "test" se reserva para evaluar el rendimiento final del modelo en datos no vistos, asegurando su capacidad de generalización.

Es importante que cada imagen.png tenga asociado un archivo de etiquetas.txt. La división del conjunto de datos en subconjuntos de entrenamiento, validación y prueba es una práctica fundamental en el aprendizaje supervisado, ya que permite entrenar, ajustar y evaluar el modelo de manera robusta.

**Archivo dataset.yaml:** Es un elemento crítico en la configuración de YOLO, ya que centraliza toda la información necesaria para cargar y procesar el conjunto de datos. Define las rutas a las subcarpetas e incluye el número de clases y sus nombres, así como la estructura de los keypoints (número de valores).

### 3.9.1.1 HIPERPARAMETROS

- **Épocas (Epoch):** Una época es una pasada completa del modelo a través de todo el conjunto de datos de entrenamiento. El número de épocas determina cuánto tiempo el modelo aprende de los datos. Un número elevado de épocas es particularmente importante en tareas de *fine-tuning* (ajuste fino), donde el modelo preentrenado se adapta a un conjunto de datos específico
- **Resolución de imagen (Imgsz):** Determina el tamaño (en píxeles) al que se redimensionan las imágenes o frames de video antes de ser procesados por el modelo. Una resolución más alta mejora la precisión al capturar más detalles, pero aumenta el consumo de memoria y tiempo de procesamiento. Una resolución alta podría saturar la memoria, mientras que una demasiado baja podría comprometer la detección de puntos clave pequeños.
- **Tamaño de lote (Batch):** Indica el número de imágenes procesadas simultáneamente en cada iteración del entrenamiento
- **Aumentación Mosaic:** Combina cuatro imágenes en una sola, creando una imagen compuesta que simula escenas complejas con múltiples objetos y posibles oclusiones.

Esto es particularmente útil en el movimiento caótico, donde las masas pueden aparecer en configuraciones inesperadas en los frames.

- **Paciencia (Patience):** Determina el número de épocas sin mejora en la métrica de validación (mAP50-95) antes de detener el entrenamiento. Esto evita desperdiciar recursos en un modelo que ha convergido o está sobreajustado.
- **Workers:** Controla cuántos procesos paralelos se utilizan para cargar los datos durante el entrenamiento. Un valor mayor permite cargar datos más rápido usando varios núcleos, pero puede causar problemas en algunos sistemas; si se establece en 0, la carga se hace sin paralelización, en un solo proceso.
- **Tasa de aprendizaje (lr0):** Controla el tamaño de los pasos que el optimizador toma al actualizar los pesos del modelo. Un valor bajo es típico para ajuste-finio, ya que el modelo preentrenado ya tiene pesos cercanos al óptimo.
- **Aumento (augment:True):** Aumentan los datos mediante transformaciones aplicadas a las imágenes de entrenamiento para simular variaciones realistas y mejorar la robustez del modelo.

### 3.9.1.2 Métricas de entrenamiento

- **Intersection over Union (IoU):** Mide la superposición entre la caja delimitadora predicha y la caja delimitadora real. Se calcula como la relación entre el área de intersección y el área de unión [18]:

$$IoU = \frac{\text{Área de intersección}}{\text{Área de unión}} \quad (86)$$

Es un parámetro fundamental para determinar si una detección es un verdadero positivo o un falso positivo.

- **Precisión:** Es la proporción de detecciones correctas (verdaderos positivos, TP) entre todas las detecciones positivas realizadas por el modelo, incluyendo falsos positivos (FP) [17]:

$$P = \frac{TP}{TP + FP} \quad (87)$$

- **Recall:** Mide la proporción de objetos reales detectados correctamente (TP) respecto al total de objetos reales, incluyendo falsos negativos (FN) [18]:

$$R = \frac{TP}{TP + FN} \quad (88)$$

- **F1-Score:** Proporciona una única puntuación que equilibra otras dos métricas importantes: la precisión y la recuperación. Este equilibrio hace que la Puntuación F1 sea especialmente valiosa en situaciones en las que la distribución de las clases es desigual (conjuntos de datos desequilibrados) o cuando tanto los falsos positivos como los falsos negativos conllevan costes significativos. Se calcula como la media armónica de la precisión y la recuperación, lo que le da un rango entre 0 y 1, donde 1 significa precisión y recuperación perfectas.

$$F1 - Score: 2 \frac{P \cdot R}{P + R} \quad (89)$$

- **Average Precision (AP):** Es el área bajo la curva de precisión-recall

$$AP = \sum_{r=0}^1 P(r) \Delta r \quad (90)$$

Donde  $r$  es el recall y  $P(r)$  es la precisión máxima para un nivel de recall dado.

- **Mean Average Precision (mAP):** Es el promedio de AP sobre todas las clases, siendo la métrica más común para evaluación global. Tenemos dos tipos de mAP. En primer lugar mAP@0.5 usa un umbral de IoU de 0.5:

$$mAP@0.5 = \frac{1}{N} \sum_{i=1}^N AP_i \quad (91)$$

Donde  $N$  es el número de clases y  $AP_i$  es la AP de la clase  $i$ .

Por otro lado mAP@0.5:0.95 es una métrica más estricta que promedia la mAP sobre un rango de umbrales de IoU, desde 0.5 hasta 0.95, en pasos de 0.05. La fórmula es:

$$mAP@0.5:0.95 = \frac{1}{10} \sum_{IoU=0.5}^{0.95} mAP@IoU \quad (92)$$

Donde mAP@IoU es la mAP calculada para un umbral específico.

## Capítulo 4. DESARROLLO

### 4.1 Planificación del trabajo fin de grado

#### 1. Estado del Arte

- Duración: 1 semana
- Fechas: 30/01/2025 - 05/02/2025
- Descripción: Investigación sobre trabajos previos en modelización de sistemas caóticos, redes neuronales LSTM y visión artificial con YOLO, revisando artículos científicos, videos y repositorios de código.

#### 2. Marco Teórico

- Duración: 3 semana
- Fechas: 06/02/2025 - 27/02/2025
- Descripción: Desarrollo de la base teórica sobre dinámica del péndulo doble, redes neuronales LSTM y visión artificial, consultando artículos y documentación técnica.

#### 3. Elaboración Física del Péndulo Doble

- Duración: 1 semana
- Fechas: 27/02/2025 - 06/03/2025
- Descripción: Construcción del modelo físico del péndulo doble para grabar videos y obtener datos experimentales.

#### 4. Resolución Física y Matemática del Péndulo Doble

- Duración: 1 semana
- Fechas: 06/03/2025 - 13/03/2025
- Descripción: Derivación de las ecuaciones de movimiento del péndulo doble con Lagrange y adición de efectos disipativos.

#### 5. Simulación en Python del Péndulo Doble

- Duración: 2 semana
- Fechas: 13/03/2025 - 27/03/2025
- Descripción: Desarrollo de un script en Python para simular el movimiento del péndulo doble y validar los resultados teóricos.

#### 6. Modelo de Visión Artificial

- Duración: 3 semana
- Fechas: 27/03/2025 - 17/04/2025
- Descripción: Implementación de un sistema de visión artificial con YOLO para extraer los ángulos  $\theta_1$  y  $\theta_2$  del péndulo doble a partir de videos.

#### 7. Modelo de Red Neuronal LSTM

- Duración: 2 semanas
- Fechas: 17/04/2025 - 01/05/2025
- Descripción: Diseño, entrenamiento y optimización de una red neuronal LSTM para predecir los ángulos del péndulo doble y generación de gráficas.

#### 8. Revisión Final y Documentación

- Duración: 2 semana y 3 días
- Fechas: 01/05/2025 - 17/05/2025
- Descripción: Revisión de los resultados finales, ajustes menores y redacción de las conclusiones para el TFG.

## 4.2 PÉNDULO SIMPLE PLANO

Antes de trabajar con el péndulo doble, estudiaremos el caso del péndulo simple como ejemplo introductorio.

El péndulo, es un sistema físico que puede oscilar bajo la acción de una fuerza como puede ser la gravitatorio o la elástica, por ejemplo. Está configurado por una masa suspendida de un punto o de un eje horizontal fijos mediante un hilo, una varilla, u otro dispositivo que pueda mantener unido el sistema (*Figura 5*).

Desde el punto de vista que nos interesa podemos definir este instrumento como un sistema de dos dimensiones sin ligaduras, por lo que se tendrá dos coordenadas generalizadas  $(l, \theta)$ . Haciendo el cambio de variable obtenemos:

$$\begin{aligned}x &= l \sin \theta \\y &= l \cos \theta\end{aligned}\tag{93}$$

Es un sistema con una coordenada generalizada ya que tenemos dos dimensiones de movimiento, una partícula y una ligadura (restricción del movimiento) holónoma.

Escribimos la energía cinética  $T$  y la energía potencial  $U$  en las coordenadas escogidas  $(\theta, \dot{\theta})$ :

$$\begin{aligned}T &= \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) = \frac{1}{2}m(l^2 \cos^2 \theta \dot{\theta}^2 + l^2 \sin^2 \theta \dot{\theta}^2) = \frac{1}{2}ml^2 \dot{\theta}^2 \\U &= -mgl \cos \theta\end{aligned}\tag{94}$$

Por lo tanto, la función lagrangiana en coordenadas polares tiene el siguiente aspecto:

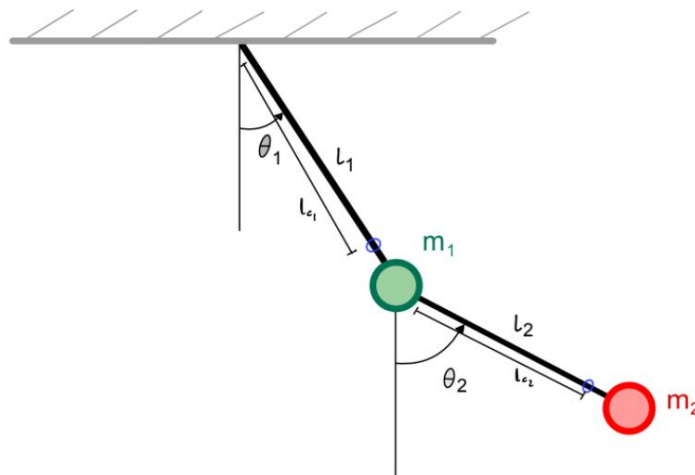
$$L(\theta, \dot{\theta}) = T - U = \frac{1}{2}m(l^2 \dot{\theta}^2) + mgl \cos \theta$$

Las ecuaciones de movimiento:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = 0 \rightarrow \frac{d}{dt} (ml^2 \dot{\theta}) + mgl \sin \theta = 0 \rightarrow \ddot{\theta} + \frac{g}{l} \sin \theta = 0\tag{95}$$

### 4.3 PÉNDULO DOBLE

Consideremos un mecanismo de dos coordenadas generalizadas (dos dimensiones, dos cuerpos, dos ligaduras). El mecanismo está formado por dos cuerpos rígidos (articulados en las uniones de los mismos) de longitudes  $l_1$  y  $l_2$  y masas  $m_1$  y  $m_2$  respectivamente. Las coordenadas generalizadas están asociadas a los ángulos  $\theta_1$  y  $\theta_2$  que miden desde la posición vertical hacia abajo, siendo ambos positivos en el sentido contrario a las agujas del reloj.



*Figura 11* Esquema péndulo doble con un primer eslabón de longitud  $l_1$ , con una masa  $m_1$  colgando de su extremo formando un ángulo  $\theta_1$  con el eje de ordenadas. Con un segundo eslabón unido al primero que a su vez cuenta con una mas  $m_2$  formando un ángulo  $\theta_2$ . A su vez, se indican los centros de masa  $l_{c1}$  y  $l_{c2}$  en azul.

Las componentes de posición del centro de masas del eslabón 1 y 2 expresadas en el plano x-y mostrado son:

$$x_1 = l_{c1} \sin \theta_1 ; x_2 = l_1 \sin \theta_1 + l_{c2} \sin \theta_2 \quad (96)$$

$$y_1 = l_{c1} \cos \theta_1 ; y_2 = l_1 \cos \theta_1 + l_{c2} \cos \theta_2 \quad (97)$$

Para determinar los centros de masa de cada eslabón se dividió el problema en dos cuerpos distintos para posteriormente calcular el CM total (*Figura 11*). El eslabón se puede desglosar en una barra rígida y en un disco con un agujero en el centro.

Para la barra:

$$\vec{r}_{CM} = \frac{\int \vec{r} dm}{\int dm} ; dm = \lambda dy \quad (98)$$

Siendo  $\lambda$  la densidad lineal de la barra.

$$\vec{r}_{CM} = \frac{\int_0^L y \hat{y} \frac{M}{L} dy}{\int_0^L \frac{M}{L} dy} \rightarrow \vec{r}_{CM} = \frac{L}{2} \hat{y} \quad (99)$$

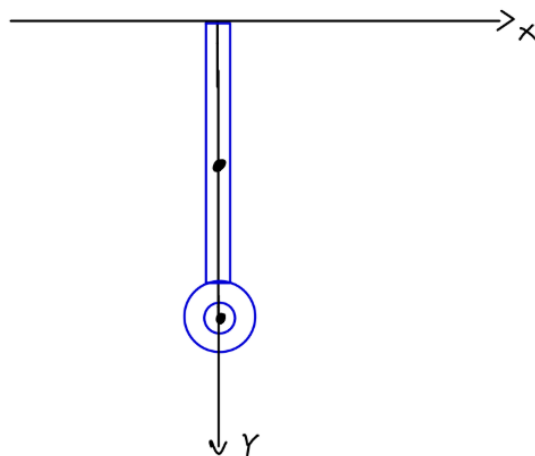


Figura 12 Esquema eslabón (barra y rodamiento).

Para el rodamiento se modelizará como si fuera un disco de radio  $R_2$  con un agujero concéntrico de radio  $R_1$ , siendo  $R_2 > R_1$

Primero se calcula el centro de masas del disco entero y luego se calcula el centro de masas del agujero, de esta forma se puede establecer una relación como si fuera el rodamiento. En ambos casos el centro de masas se encuentra en  $(0,0)$ , por lo tanto:

$$X_{rodamiento+agujero} = \frac{m_r x_r + m_a x_a}{m_r + m_a} = 0 \rightarrow x_r = 0 = y_r \quad (100)$$

De tal forma que:

$$X_{CM} = \frac{\sum_i x_i m_i}{m_T} = 0; Y_{CM} = \frac{\sum_i y_i m_i}{m_T} = \frac{\frac{L}{2} m_b + (L + R_2) m_r}{m_b + m_r} \quad (101)$$

Este mismo razonamiento se puede aplicar a los dos eslabones sin ninguna distinción más que introducir correctamente los valores de cada uno.

Volviendo a las ecuaciones (96) y (97), la velocidad será:

$$\dot{x}_1 = l_{c1} \cos \theta_1 \dot{\theta}_1; \dot{x}_2 = l_1 \cos \theta_1 \dot{\theta}_1 + l_{c2} \cos \theta_2 \dot{\theta}_2 \quad (102)$$

$$\dot{y}_1 = -l_{c1} \sin \theta_1 \dot{\theta}_1; \dot{y}_2 = -l_1 \sin \theta_1 \dot{\theta}_1 + l_{c2} \sin \theta_2 \dot{\theta}_2 \quad (103)$$

Partiendo de aquí se puede expresar la energía cinética como la suma de la energía de ambos péndulos. Al tratarse un sólido con un movimiento oscilatorio se debe añadir la energía cinética de rotación correspondiente:

$$T_{rot} = \sum_i \frac{1}{2} m_i v_i^2 = \frac{1}{2} \sum_i m_i r_i^2 \omega^2 = \frac{1}{2} I \omega^2 \quad (104)$$

Donde  $I$  representa el momento de inercia. En este caso y para simplificar los cálculos se asumió el momento de inercia de una barra delgada respecto a un eje perpendicular que pasa por su extremo:

$$I = \frac{1}{3}ml^2 \quad (105)$$

Teniendo esto en cuenta y empleando las identidades trigonométricas pertinentes, podemos llegar a la expresión final de la energía cinética como el cómputo de la suma de la energía cinética de traslación y de rotación de ambas partículas:

$$T = T_1 + T_2 = \frac{1}{2}(m_1l_{c1}^2 + m_2l_1^2 + I_1)\dot{\theta}_1^2 + m_2l_1l_{c2} \cos(\theta_1 - \theta_2)\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}(m_2l_{c2}^2 + I_2)\dot{\theta}_2^2 \quad (106)$$

De forma similar, la energía potencial puede descomponerse como la suma de dos partes. Partiendo del origen de la energía potencial y en función a los ejes de la *figura 11*, la energía potencial del sistema queda definida como:

$$U = U_1 + U_2 = -m_1gy_1 - m_2gy_2 = -(m_1l_{c1} + m_2l_1)g \cos\theta_1 - m_2gl_{c2} \cos\theta_2 \quad (107)$$

Tomando las ecuaciones (106) y (107) se tiene que:

$$L = T - U = \frac{1}{2}(m_1l_{c1}^2 + m_2l_1^2 + I_1)\dot{\theta}_1^2 + \frac{1}{2}(m_2l_{c2}^2 + I_2)\dot{\theta}_2^2 + m_2l_1l_{c2} \cos(\theta_1 - \theta_2)\dot{\theta}_1\dot{\theta}_2 + m_1gl_{c1} \cos\theta_1 + m_2g(l_1 \cos\theta_1 + l_{c2} \cos\theta_2) \quad (108)$$

Teniendo definido el Lagrangiano, las ecuaciones dinámicas que modelan el péndulo doble se obtienen aplicando las ecuaciones de Euler-Lagrange:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \left( \frac{\partial L}{\partial q_i} \right) + \left( \frac{\partial R_i}{\partial \dot{q}_i} \right) = Q_i, \quad i = 1, 2, \dots, n \text{ grados de libertad} \quad (109)$$

Se tienen dos coordenadas generalizadas, por lo tanto:

$$D_1 = \frac{1}{2}b_1\theta_1^2; D_2 = \frac{1}{2}b_2\theta_2^2 \quad (110)$$

Teniendo esto en cuenta, se puede desarrollar la ecuación de Euler-Lagrange para cada c.g:

$$\theta_1 \rightarrow \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \left( \frac{\partial L}{\partial \theta_1} \right) + \left( \frac{\partial D_1}{\partial \dot{\theta}_1} \right) = 0 \quad (111)$$

$$\frac{\partial L}{\partial \theta_1} = (m_1l_{c1}^2 + I_1 + m_2l_1^2)\dot{\theta}_1 + [m_2l_1l_{c2} \cos(\theta_1 - \theta_2)]\dot{\theta}_2 \quad (112)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) = (m_1l_{c1}^2 + I_1 + m_2l_1^2)\ddot{\theta}_1 + m_2l_1l_{c2} [\cos(\theta_1 - \theta_2)\ddot{\theta}_2 - \sin(\theta_1 - \theta_2)(\dot{\theta}_1 - \dot{\theta}_2)\dot{\theta}_2] \quad (113)$$

$$\frac{\partial L}{\partial \theta_1} = -m_2l_1l_{c2}\dot{\theta}_1\dot{\theta}_2 \sin(\theta_1 - \theta_2) - (m_1l_{c1} + m_2l_1)g \sin\theta_1 \quad (114)$$

$$\frac{\partial D_1}{\partial \dot{\theta}_1} = b_1 \dot{\theta}_1 \quad (115)$$

Finalmente, sustituyendo los términos de arriba y desarrollando la expresión completa, se obtiene:

$$\begin{aligned} \theta_1 \rightarrow (m_1 l_{c1}^2 + m_2 l_1^2 + I_1) \ddot{\theta}_1 + m_2 l_1 l_{c2} \cos(\theta_1 - \theta_2) \ddot{\theta}_2 + m_1 l_1 l_{c2} \sin(\theta_1 - \theta_2) \dot{\theta}_2^2 + \\ + (m_1 l_{c1} + m_2 l_1) g \sin \theta_1 + b_1 \dot{\theta}_1 = 0 \end{aligned} \quad (116)$$

El desarrollo para la coordenada restante  $\theta_2$  se hace de manera homóloga a lo hecho para  $\theta_1$ , por lo tanto, la ecuación de Euler Lagrange queda como a continuación se muestra:

$$\theta_2 \rightarrow \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \left( \frac{\partial L}{\partial \theta_2} \right) + \left( \frac{\partial D_2}{\partial \dot{\theta}_2} \right) = 0 \quad (117)$$

$$\begin{aligned} \theta_2 \rightarrow (m_2 l_{c2}^2 + I_2) \ddot{\theta}_2 + m_2 l_1 l_{c2} \cos(\theta_1 - \theta_2) \ddot{\theta}_1 - m_2 l_1 l_{c2} \sin(\theta_1 - \theta_2) + \\ + m g l_{c2} \sin \theta_2 + b_2 \dot{\theta}_2 = 0 \end{aligned} \quad (118)$$

Las ecuaciones (116) y (117) son las ecuaciones dinámicas del sistema, las cuales son un conjunto de ecuaciones diferenciales no lineales simultaneas para las coordenadas generalizadas  $\theta_1$  y  $\theta_2$ .

Para que las ecuaciones sean más manejables para la simulación en Python, se expresaran en forma de matrices:

$$\begin{aligned} \begin{pmatrix} m_1 l_{c1}^2 + m_2 l_1^2 + I_1 & m_2 l_1 l_{c2} \cos(\theta_1 - \theta_2) \\ m_2 l_1 l_{c2} \cos(\theta_1 - \theta_2) & m_2 l_{c2}^2 + I_2 \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + \begin{pmatrix} 0 & m_2 l_1 l_{c2} \sin(\theta_1 - \theta_2) \\ -m_2 l_1 l_{c2} \sin(\theta_1 - \theta_2) & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{pmatrix} + \\ + \begin{pmatrix} 0 & m_2 l_1 l_{c2} \sin(\theta_1 - \theta_2) \\ -m_2 l_1 l_{c2} \sin(\theta_1 - \theta_2) & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{pmatrix} + \begin{pmatrix} b_1 & 0 \\ 0 & b_2 \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} + \begin{pmatrix} (m_1 l_{c1} + m_2 l_1) g \sin \theta_1 \\ m_2 g l_{c2} \sin \theta_2 \end{pmatrix} = 0 \end{aligned} \quad (119)$$

#### 4.4 CONSTRUCCION DEL PENDULO

Uno de los objetivos de este trabajo es construir el mecanismo del péndulo doble. El diseño y la fabricación de los eslabones que conforman el péndulo doble es una parte fundamental para el diseño del mecanismo, sin embargo, no es lo único indispensable, también es necesario diseñar y fabricar la base donde será montado el sistema.

Los materiales que se usaron en este proceso fueron los siguientes:

1. Tablón de fibras de densidad media (MDF crudo) 60x120x1 cm
2. Tablón de madera virgen de pino 40x104x2 cm
3. Rodamientos 608 2RS 22x7x8 mm
4. Tornillos, tuercas y arandelas M8
5. Teflón
6. Pegatinas de colores (verde y rojo)
7. Cinta de carrozero

Victor Manuel Suárez Ferrer

---

8. Pintura de pizarra negro mate de secado rápido
9. Sierra de calar BOSCH PST-650
10. Taladro percutor 710W
11. Juego de brocas de corona bimetálica de M3
12. Cinta métrica

#### 4.4.1 BASE

Se uso un tablón de madera de pino de 40x104x2 cm como punto de partida y por medio de la sierra de calar BOSCH PST-650 se dio la forma deseada. Se aposto por un diseño estable y practico, formado por un tablón horizontal de 14x104x2cm y un tablón vertical de 7x35x2 cm. Para aportar mayor solidez a la base se añadió un pequeño tablón uniendo tanto el horizontal como el vertical (*Figura 13*).

Para facilitar la tarea de reconocimiento de la visión artificial se aplicó pintura negra mate sobre la madera, obteniendo así un alto contraste con el fondo y los eslabones.



*Figura 13 Base vista desde atrás en plano picado.*

---

#### 4.4.2 ESLABONES

Para el diseño de los eslabones, los parámetros de interés son la longitud, la ubicación del centro de masas y la masa de los eslabones. En primer lugar, los eslabones deberán tener las dimensiones apropiadas para aportar imágenes lo suficientemente claras para que el proceso de anotación de puntos. Las piezas deben tener un bajo peso de forma que la base pueda soportar el peso dinámico del sistema en funcionamiento. El parámetro más importante es que los centros de masa estén ubicados en los extremos opuestos a los ejes de giro (mayor momento de fuerza). Se debe tener en cuenta que el primer eslabón debe ser lo suficientemente robusto para sostener al segundo sin prestar flexiones o deformaciones.

En realidad, como la geometría del diseño preliminar es invariable, el único parámetro a proponer es la longitud y el ancho total de los eslabones; un cambio en las longitudes y los anchos implica la modificación en su masa y en su centro de masas. Con base a estas

Victor Manuel Suárez Ferrer

---

especificaciones se realizó una simulación en Python con el objetivo de determinar la longitud más eficiente para llegar al caos en una situación en la que el sistema se lanzara con el primer eslabón formando  $90^\circ$  con respecto al eje de ordenadas y el segundo eslabón en posición natural ( $0^\circ$ ). El resultado fue una proporción de 1,5 entre el primer y el segundo eslabón, siendo el primero más grande.

Por otro lado, desplazar el centro de masas a un extremo, resulto relativamente fácil ya que el propio diseño exigía de la aparición de rodamientos, tornillos y tuercas con un peso notablemente mayor al del resto en esta posición.

Finalmente se optó por un tablón de MDF crudo. Se eligió este material principalmente por su baja densidad y por su alta resistencia a técnicas como el corte y el torneado, además de que en general presenta buenas propiedades mecánicas. La proporción de 1,5 que se obtuvo en las simulaciones y el balance entre reducir al máximo la longitud y ancho de los tabloncillos sin perder precisión en el modelo de VA, concluyo en diseñar un primer eslabón de longitud  $l_1 = 20\text{ cm}$ , ancho  $g_1 = 4\text{ cm}$  y un segundo eslabón de longitud  $l_2 = 15\text{ cm}$  con un ancho  $g_2 = 4\text{ cm}$ . Para obtener las medidas deseadas, se empleó la sierra de calar, cinta de carroceros para evitar astillamientos y una cinta métrica para marcar las dimensiones. Hay que tener en cuenta errores de paralaje o los propios errores sistemáticos de la cinta métrica (precisión dispositivo analógico), todo esto puede llevar a pequeñas desviaciones entre la simulación y el sistema físico.

Una vez obtenidos los tabloncillos, se procedió a perforarlos con el taladro percutor y una broca de corona de 8mm (igual que el rodamiento). El resultado fue un agujero de unos milímetros superior al diámetro del rodamiento lo que daba lugar a un cabeceo no deseado (*Figura 14*).



*Figura 14* Eslabones 1 y 2 de longitudes 20 y 15 cm respectivamente y ancho de 4 cm con agujeros de 8 mm de diámetro

Para solucionarlo, se encinto cada rodamiento con una fina capa de cinta de carroceros y se introdujo en los correspondientes agujeros a presión. Finalmente se debían ensamblar entre si y con la propia base, para ello se utilizaron tornillos M8, arandelas (reduciendo la superficie de contacto entre los tornillos y los rodamientos) y tuercas que permitían mantener fijos estos enganches. De este modo la masa 1 quedo conformada por 2 rodamientos de 13,15g, dos tuercas de 4,51 g y un tornillo de 13,43g, haciendo un total de 35,6g. En el extremo del segundo

eslabón se añadió un eslabón superpuesto de 9,45g. Por último, se colocaron las pegatinas correspondientes en cada masa para que el etiquetado de las imágenes que iba a entrenar el modelo de YOLO fuera más sencillo, dando como resultado final el mostrado en la *Figura 15*.



*Figura 15* Resultado final del montaje experimental.

---

## 4.5 SIMULACIÓN EN PYTHON

Se desarrolló un código en Python ([Anexo-Codigos-01](#)) a modo de herramienta que mostrara el comportamiento dinámico y la naturaleza caótica del péndulo doble, así como su sensibilidad a las condiciones iniciales. Para ello se hizo uso de las siguientes librerías (Tabla 1):

*Tabla 1* Bibliotecas usadas en la simulación del péndulo doble en Python

<b>Numpy</b>	Para cálculos numéricos y manejo de matrices
<b>Matplotlib.pyplot</b>	Para graficar la simulación
<b>FuncAnimation</b>	Para crear la animación del péndulo
<b>TextBox</b>	Para agregar campos de texto interactivo
<b>Os</b>	Para manejar directorios y rutas de archivo

Para construir este modelo, se definieron parámetros físicos como las masas de los eslabones, sus longitudes, los centros de masa y los momentos de inercia. Además, se incorporaron

Victor Manuel Suárez Ferrer

términos de fricción para simular un comportamiento más realista, incluyendo la resistencia del aire proporcional a la velocidad angular y la fricción en las articulaciones.

*Tabla 2 Datos experimentales introducidos en la simulación*

<b>Parámetro</b>	<b>En el código</b>	<b>Valor</b>
<i>Masa extremo eslabón 1</i>	masa_1	0,036 kg
<i>Masa extremo eslabón 2</i>	masa_2	0,009 kg
<i>Longitud del primer eslabón</i>	longitud1	0,200 m
<i>Longitud del segundo eslabón</i>	longitud2	0,150 m
<i>Ancho eslabones</i>	radio_barra	0,040 m
<i>Momento de inercia del primer péndulo</i>	inercia1	0,002 kg m <sup>2</sup>
<i>Momento de inercia del segundo péndulo</i>	inercia1	0,001 kg m <sup>2</sup>
<i>Coefficiente de fricción del aire</i>	fricción_aire	0,100
<i>Fricción primer rodamiento</i>	fricción_rod1	0,05
<i>Fricción primer rodamiento</i>	fricción_rod1	0,05

El modelo dinámico del péndulo doble se basó en la formulación de Lagrange descrita en la sección (4.3). El resultado de las ecuaciones en formato matricial fue lo que se utilizó para las físicas detrás de la animación.

Dado que las ecuaciones de movimiento del péndulo doble son no lineales y no admiten una solución analítica, se optó por un método de integración numérica para resolverlas. Se optó por el método Runge-Kutta de cuarto orden (RK4) debido a su precisión y estabilidad en sistemas dinámicos no lineales. Este método calcula la evolución de los ángulos y velocidades angulares en pequeños pasos de tiempo, utilizando cuatro estimaciones intermedias para aproximar la derivada en cada intervalo.

La integración se implementó de manera que, en cada iteración, se resolviera la ecuación dinámica para la aceleración angular, invirtiendo la matriz de inercia y aplicando las fuerzas correspondientes. Este enfoque permitió capturar el comportamiento caótico característico del sistema bajo ciertas condiciones iniciales

Para representar gráficamente el movimiento, se desarrolló un módulo de cinemática directa que transforma los ángulos del péndulo en coordenadas cartesianas. Las posiciones de las masas se calcularon mediante relaciones trigonométricas triviales, considerando la geometría del sistema. La visualización incluyó elementos adicionales como una trayectoria que sigue el

movimiento de la segunda masa (representado de manera más visual el caos), así como indicadores del tiempo transcurrido.

Se diseñó una interfaz interactiva que permite modificar los parámetros físicos del sistema, como las masas, longitudes, ángulos iniciales y coeficientes de fricción, durante la ejecución de la simulación. Este enfoque facilita el análisis de cómo diferentes configuraciones afectan el comportamiento del péndulo, lo que resulta especialmente útil para estudiar transiciones entre movimientos periódicos y caóticos.

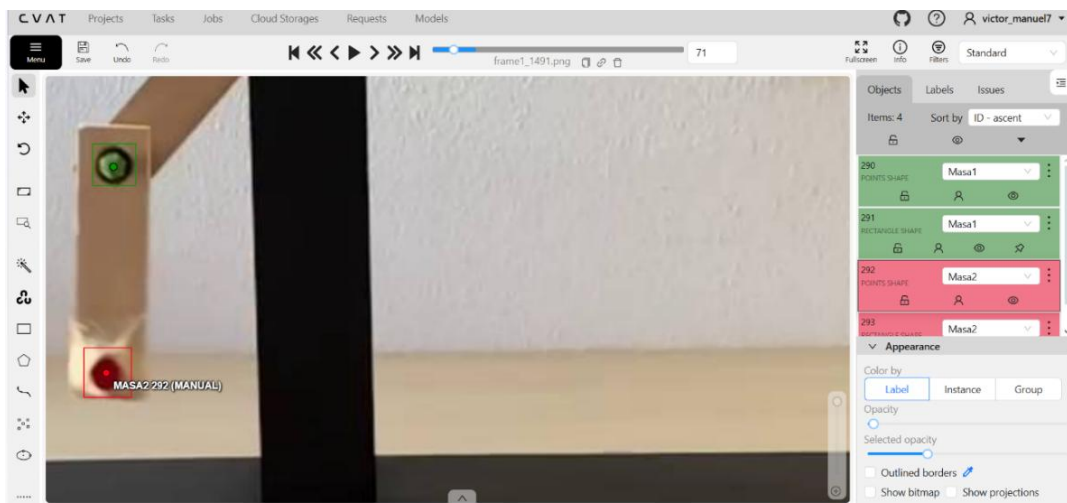
Cabe destacar algunos factores clave que se tuvieron en cuenta, como una constante de gravedad basada en un valor regional (Madrid, España), la elección de un paso de tiempo pequeño para asegurar una animación fluida, la ausencia de torques externos para centrarse en el movimiento libre del sistema, y una elección de colores fiel a la teoría y al péndulo físico.

## 4.6 MODELO DE VISIÓN ARTIFICIAL

En primer lugar, se debe grabar la cinemática del péndulo doble, lo que producirá los frames de entrenamiento para el modelo. Se grabó un total de 4 videos a 240 fps y 720p con un iPhone 11, variando ligeramente la orientación, iluminación y condiciones iniciales, de forma que el futuro modelo fuera lo más robusto posible.

Se desarrolló un código en Python ([Anexo-Codigos-04](#)) que seleccionara 200 frames de un vídeo dividiendo el número total de frames entre un intervalo calculado dinámicamente, asegurando una distribución uniforme a lo largo del video. De esta forma se obtuvieron un total de 800 imágenes de entrenamiento.

Para la tarea de anotación se optó por CVAT. CVAT es una herramienta de código abierto diseñada para anotaciones precisas en tareas de visión artificial, como detección de objetos, segmentación, y detección de keypoints. Se definieron dos clases (Masa1 y Masa2) y se procedió a anotar en cada una de ellas tanto los keypoints como las boundingbox (*Figura 16*).



*Figura 16* Captura de pantalla para ejemplificar el funcionamiento de CVAT en este proyecto.

---

Una vez se han etiquetado las 800 fotos, se deben exportar las anotaciones. Actualmente, CVAT no exporta los datos correctamente para YOLO pose, es por eso que se implementó un código en Python ([Anexo-Codigos-05](#)) capaz de convertir el formato CVAT en un formato legible para YOLO. En este caso, el archivo XML que se obtiene directamente de CVAT contiene información sobre las distintas bounding boxes y puntos clave etiquetados con clases como Masa1 y Masa2. El código convierte estas anotaciones en archivos de texto con coordenadas normalizadas, siguiendo el formato YOLO.

El formato YOLO para detección con puntos clave requiere, para cada objeto, una línea de texto con la siguiente estructura:

```
<class_id> <x_center> <y_center> <width> <height> <x_point> <y_point> <visibility>
```

Donde:

- class\_id: Identificador numérico del objeto.
- x\_center, y\_center: Coordenadas normalizadas del centro del bounding box.
- width, height: Ancho y alto normalizados del bounding box.
- x\_point, y\_point: Coordenadas normalizadas de los keypoints.
- visibility: Indicador binario que representa si el keypoint es visible o está ocluido.

Por lo que finalmente obtendremos un total de 800 archivos txt que contiene dicha sintaxis para cada una de las dos masas.

El siguiente paso es dividir el conjunto total de datos en la estructura que se mencionó en el punto (3.9.1). Para esta tarea también se desarrolló un código en Python ([Anexo-Codigos-06](#)) capaz de organizar un conjunto de datos de imágenes y anotaciones en formato YOLO, dividiéndolo en subconjuntos de entrenamiento (train, 70%), validación (val, 20%) y prueba (test, 10%), creando una estructura de carpetas estándar y generando un archivo dataset.yaml para el modelo YOLO.

Para la tarea de pose-estimation se utilizó **YOLO11n-pose**, un modelo preentrenado optimizado para detección de objetos y estimación de puntos clave. Se aplicó un “fine-tuning” con el conjunto de datos nombrado anteriormente para poder detectar las masas.

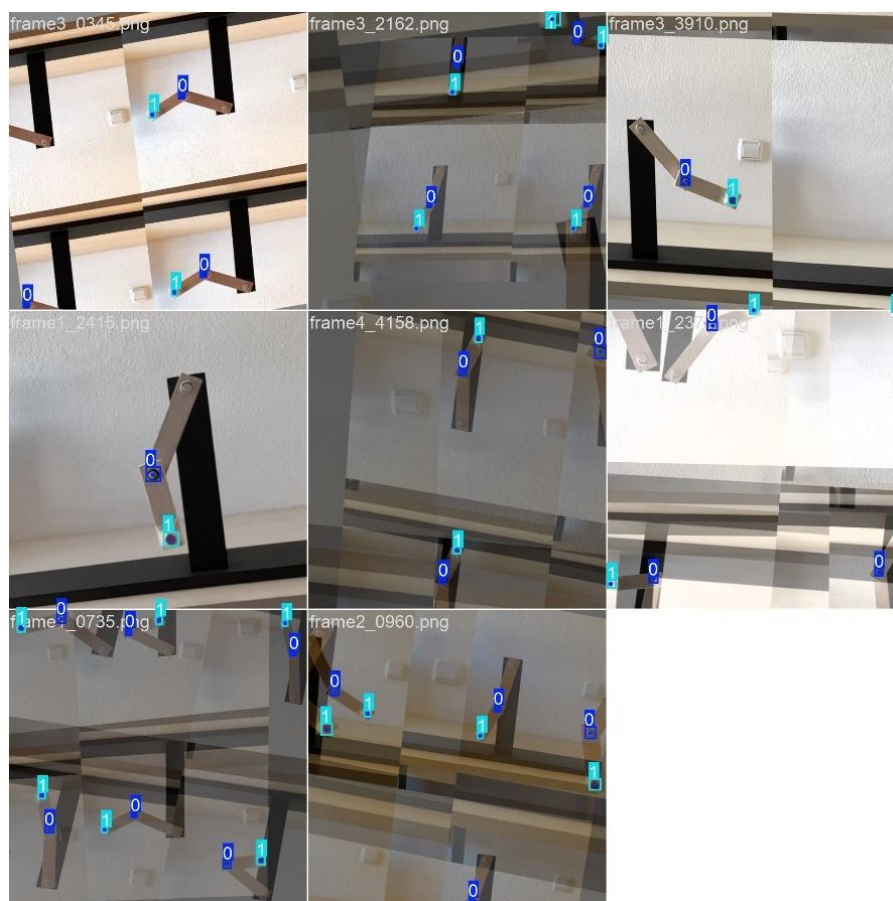
El código diseñado para este fin emplea varias bibliotecas: *os* para gestionar el entorno y verificar la existencia de archivos, *ultralytics.YOLO* para cargar y entrenar el modelo, y *torch* para verificar la disponibilidad de una GPU con soporte CUDA, lo que acelera el proceso de entrenamiento. La estructura del código se organiza en varias etapas clave ([Anexo-Codigos-07](#)). Primero, realiza una verificación de la GPU, mostrando detalles como el nombre del dispositivo, la versión de CUDA y la memoria disponible, lo que asegura que el entrenamiento pueda optimizarse según el hardware disponible. Segundo, carga el modelo preentrenado *yolo11n-pose.pt*, una variante ligera diseñada para GPUs con memoria limitada, como la GTX 1050 con la que se realizó el entrenamiento en este caso. Tercero, define una configuración de hiperparámetros optimizada específicamente para el caso del péndulo doble.

En el primer modelo se establecieron 200 épocas para capturar la complejidad del movimiento caótico, una resolución de imagen *kjimgs=736* (ajustada al video de 720 y reescalada a 736

porque YOLO trabaja con múltiplos de 32), un tamaño de lote batch=8 compatible con la memoria de la GTX 1050, y se incluyeron aumentos de datos como mosaic, flipud (flip vertical) y degrees=10.0 (rotación máxima) para simular variaciones en el movimiento y mejorar la robustez ante frames borrosos o condiciones difíciles.

En cuanto a la optimización y regularización, se empleó el optimizador AdamW, que incorpora decaimiento de pesos para mejorar la regularización, con una tasa de aprendizaje inicial (lr0) de 0.001 y una tasa final (lrf) de 0.01, utilizando un scheduler de aprendizaje tipo cosine annealing (cos\_lr=True) que reduce progresivamente la tasa de aprendizaje para una convergencia más estable. Para mejorar la robustez del modelo frente a variaciones en las imágenes, como borrosidad o cambios de iluminación, se implementó un aumento de datos agresivo: hsv\_h, hsv\_s y hsv\_v se configuran en 0.015, 0.7 y 0.4, respectivamente, para variar el tono, la saturación y el brillo.

A continuación (*Figura 18*) se muestran los lotes de imágenes procesadas por el modelo. Estas imágenes generadas automáticamente por YOLO permiten verificar visualmente los datos de entrada, las etiquetas (bounding boxes y keypoints de Masa1 y Masa2), y la aplicación del aumento de datos.



*Figura 17* Mosaico generado por el modelo pose mostrando un lote de 8 imágenes procesadas durante el entrenamiento de visión artificial, con etiquetas de bounding boxes y keypoints para las masas del péndulo doble, ilustrando la aplicación de aumento de datos.

---

El entrenamiento se realizó optimizando el modelo sobre el conjunto de entrenamiento, y posteriormente se validó en los conjuntos de validación y prueba. Durante la validación, se calcularon métricas de rendimiento para las tareas de detección de objetos (bounding boxes) y estimación de poses (keypoints), incluyendo precisión (Precision), sensibilidad (Recall), mAP@50 y mAP@50:95, presentando los resultados como valores promedio (usando np.mean) por clase (Masa1 y Masa2), lo que permite un análisis detallado del rendimiento del modelo para cada componente del péndulo doble.

Para calcular el centro de masa de cada objeto (Masa1 y Masa2) en el péndulo doble se estimó a partir de las detecciones proporcionadas por el modelo de visión artificial que se seleccionó finalmente. La idea era combinar la información de las cajas delimitadoras y los puntos clave para obtener una representación más precisa de la posición central de cada masa.

Para cada frame que el modelo analiza, se devuelven 6 valores. Para cada boundingbox se obtiene  $x_{min}, x_{max}, y_{min}, y_{max}$ . Y para cada keypoint:  $kp_{centrox}, kp_{centroy}$ . Partiendo de esta información de entrada se puede calcular el centro de masas como el promedio ponderado entre el centro de las cajas (media de las coordenadas) y el centro de los keypoints:

$$cm = 0,5 \cdot x_{centro} + 0,5 \cdot kp_{centrox} \quad (120)$$

La caja delimitadora proporciona una estimación general, mientras que los keypoints ofrecen una localización más fina. Esto asume una contribución igual (50%) de cada fuente, suavizando errores de detección y dando una posición representativa del centro de masa en píxeles. Sin embargo, para conocer los ángulos experimentales no vale con conocer la posición del centro de masas en cada instante ya que también se ha de conocer la posición del pivote.

Si se tiene en cuenta que la trayectoria de la Masa1 describe un arco de círculo centrado en el pivote, no es necesario reetiquetar los 800 frames de entrenamiento, sino que se puede utilizar la ecuación de un círculo para obtener la posición.

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (121)$$

donde  $(x_c, y_c)$  representa las coordenadas del centro del círculo (el pivote) y  $r$  es el radio, que corresponde aproximadamente a la longitud del primer eslabón del péndulo. Cada punto  $(x_i, y_i)$  registrado en la trayectoria de Masa1 debería cumplir esta ecuación, con pequeñas desviaciones debido a ruido o imperfecciones en las detecciones.

El desafío es encontrar los valores óptimos de  $x_c, y_c$  y  $r$  que mejor se ajusten a todos los puntos observados. Para ello, se define una función de pérdida que cuantifica el error entre la distancia de cada punto al centro estimado y el radio esperado. Esta distancia se calcula como:

$$distancia = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad (122)$$

La función de pérdida mide la diferencia entre esta distancia y el radio  $r$ :

$$perdida = \sqrt{(x - x_c)^2 + (y - y_c)^2} - r \quad (123)$$

El objetivo es minimizar la suma de los errores al cuadrado de esta función para todos los puntos  $(x_i, y_i)$  de la trayectoria de Masa1. Esto se alinea con el principio de mínimos cuadrados, que

busca la mejor aproximación a un modelo geométrico dado un conjunto de datos experimentales. La minimización asegura que las desviaciones de los puntos respecto al círculo sean las menores posibles, reflejando la hipótesis de que la trayectoria es circular.

Para guiar el proceso de optimización, se necesita una estimación inicial razonable. Se parte de la media de las coordenadas  $x$  e  $y$  de todas las posiciones de Masa1, calculada como:

$$x_{media} = \frac{1}{n} \sum_{i=1}^n x_i, \quad y_{media} = \frac{1}{n} \sum_{i=1}^n y_i \quad (124)$$

donde  $n$  es el número total de frames analizados. Esta media actúa como una aproximación inicial del centro, asumiendo que las oscilaciones tienden a distribuirse simétricamente alrededor del pivote. El radio inicial se estima como la distancia máxima desde esta media al punto más lejano en la trayectoria:

$$r_{inicial} = \sqrt{(x_i - x_{media})^2 + (y_i - y_{media})^2} \quad (125)$$

Esta elección es lógica porque el punto más alejado de la media probablemente define el extremo del arco oscilatorio, proporcionando una cota superior inicial para el radio.

Una vez detectado el pivote, la obtención de los ángulos es más trivial, ya que basta con trazar una recta vertical en el pivote y en la Masa1 y un par de rectas que unan el pivote con la Masa1 y la Masa1 con la Masa2. Finalmente, para calcular el ángulo habría que aplicar la función arcotangente. Se diseñó un código en Python ([Anexo-Codigos-10](#)) que contara con la estimación del pivote y los cálculos de  $\theta_1$  y  $\theta_2$ , dando lugar a un CSV que contiene columnas para el frame, el tiempo (en segundos),  $\theta_1$  y  $\theta_2$ .

## 4.7 RED NEURONAL LSTM

En el contexto de un péndulo doble, los ángulos  $\theta_1$  y  $\theta_2$  son las variables fundamentales que describen el estado del sistema en cada instante de tiempo. Dado que los frames están espaciados uniformemente en el tiempo, estos pares forman una serie temporal bivariada, es decir, una secuencia de observaciones donde cada punto temporal tiene dos dimensiones:  $\theta_1(t)$  y  $\theta_2(t)$ .

Para modelar la dinámica de un péndulo doble con una red LSTM, necesitamos transformar la serie temporal en un problema de aprendizaje supervisado, donde el modelo aprenda a mapear un historial de datos (entrada) a un conjunto de valores futuros (salida). En series temporales, esto se logra mediante el uso de ventanas temporales deslizantes:

- Entrada (X): Una ventana de  $T_{input}$  pasos temporales que contiene el historial reciente de los ángulos.

$$X(t) = \begin{bmatrix} \theta_1(t - T_{input} + 1) & \theta_2(t - T_{input} + 1) \\ \theta_1(t - T_{input} + 2) & \theta_2(t - T_{input} + 2) \end{bmatrix} \quad (126)$$

Y así sucesivamente hasta alcanzar los  $T_{input}$  definidos, por lo que obtenemos una matriz  $T_{input} \times 2$  dimensiones. El + 1 se añade porque en Python los intervalos se

definen con un índice de inicio (inclusive) y un índice de fin (exclusivo). Sin embargo, en el contexto de contar frames, queremos incluir explícitamente el frame inicial del historial.

- Salida( $y$ ): Una ventana de  $T_{output}$  pasos temporales que contiene los valores futuros de los ángulos

$$y(t) = \begin{bmatrix} \theta_1(t+1) & \theta_2(t+1) \\ \vdots & \vdots \\ \theta_1(t+T_{output}) & \theta_2(t+T_{output}) \end{bmatrix} \quad (127)$$

En un principio se optó por escoger  $T_{input} = 10$  y  $T_{output} = 1$ , ya que 10 frames de entrada permiten capturar un historial suficiente para que la LSTM identifique los patrones del péndulo y una salida de 1 frames para una predicción precisa a corto plazo. Como las ecuaciones de movimiento del péndulo doble son diferenciales y el movimiento es el resultado de la integración temporal de estas dinámicas, la LSTM, al ser una red recurrente, puede aprender estas dependencias al procesar la secuencia de entrada y generar una secuencia de salida.

Para generar múltiples pares entrada-salida, se desliza la ventana temporal a lo largo de toda la serie. Si  $N$  es el número total de frames en el video, los índices  $t$  para los que se puede crear una muestra válida van desde  $t = T_{input}$  hasta  $t = N - T_{output}$ . Esto asegura que siempre haya suficientes datos para formar la entrada y la salida. El número total de muestras generadas será:

$$M = N - T_{input} - T_{output} + 1 \quad (128)$$

Obteniendo finalmente un tensor de entrada de forma  $(M, T_{input}, 2)$  y un tensor de salida de forma  $(M, T_{output}, 2)$ . Al estructurar los datos de esta manera, permitimos que la red aprenda estas dependencias temporales de manera implícita, sin necesidad de modelar explícitamente las ecuaciones diferenciales del sistema, que son complejas y no lineales.

Al igual que se hizo en el modelo de visión artificial, se divide el conjunto de datos en entrenamiento (80%) y validación (20%). Esto asegura que el modelo aprenda patrones generales sin sobreajustarse a los datos observados, y permite medir su capacidad de generalización. La red LSTM no empleó un conjunto de prueba independiente porque el objetivo era evaluar su capacidad para predecir datos no vistos utilizando validación cruzada y métricas de error (como MAE) en el conjunto de validación.

Para este propósito se diseñó un código en Python ([Anexo-Codigos-11](#)) que generara los datos de entrenamiento para la red LSTM y eliminara los primeros 5 segundos del video ya que al revisar la validación del modelo se percibió que la aparición de una mano estaba generando datos corruptos. Utilizando la biblioteca pandas, se carga el archivo que contiene las series temporales de los ángulos provenientes de 6 videos distintos, y se combinan en un array bivariado. Para manejar posibles valores faltantes (NaN), se aplica una interpolación lineal con el método `interpolate()` seguido de un relleno hacia atrás (`fillna(method='bfill')`).

Un aspecto clave es la detección automática de la tasa de frames por segundo (FPS) del video original mediante `cv2.VideoCapture`, lo que permite calcular dinámicamente el número de

frames a recortar. Posteriormente, los datos se normalizan al rango  $[1, 1]$  usando `MinMaxScaler`, asumiendo un rango de  $[-180^\circ, 180^\circ]$ .

El siguiente paso es definir la arquitectura del modelo LSTM ([Anexos-Codigos-12](#)). Se definió el tamaño del input ( $x_t$ ) y del output ( $x_t$ ) igual a 2 ya que esto corresponde con un vector de dos dimensiones, representando los ángulos del péndulo doble en un instante. Por otro lado, el estado oculto ( $h_t$ ) y el estado de celda ( $C_t$ ) se definieron con una dimensión de 64. Este parámetro determina la capacidad de representación de la LSTM, es decir, cuanta información puede codificar sobre patrones complejos en la secuencia. Un valor de 64 es común en problemas de series temporales de complejidad moderada, como el péndulo doble, ya que proporciona suficiente expresividad para modelar trayectorias caóticas sin ser excesivamente costoso computacionalmente y sin correr el riesgo de sobreajuste. Ambos estados iniciales se definieron como cero, ya que el modelo no cuenta con ninguna información previa sobre la secuencia.

El modelo que se diseñó contaba con dos capas apiladas, esto quiere decir que el primer estado oculto se usa como input para la segunda capa. Esto aumenta la profundidad de la red, permitiendo modelar relaciones no lineales más complejas y dependencias a largo plazo. La elección de dos capas vino de iterar el código cambiando este hiperparámetro hasta que se obtuvo la mejor precisión. El aumento de capas se tradujo en una precisión notablemente peor, seguramente por el sobreajuste.

Para procesar una secuencia de ángulos, como una serie de 10 pasos de tiempo (10 frames del video del péndulo), la LSTM aplica las siguientes operaciones en cada paso  $t$ . Por ejemplo, en el paso  $t = 5$  la puerta de olvido decide qué información olvidar, como una oscilación no relevante de  $\theta_1$ , y que información añadir, como por ejemplo una nueva tendencia de  $\theta_1$ , asignando valores bajos (por ejemplo, 0.1) a partes irrelevantes del estado de celda anterior ( $C_{t-1}$ ). La puerta de entrada evalúa que cambios son importante para predecir el movimiento futuro y así añadirlos a la memoria, mientras el vector candidato prepara una propuesta de nueva información sobre los ángulos, pero espera la aprobación de la puerta de entrada para incluirla. El estado de celda ( $C_t$ ) se actualiza combinando la información retenida con la nueva, formando una memoria a largo plazo. Finalmente, la puerta de salida selecciona que información sobre el movimiento de los ángulos es útil para la predicción, generando el estado oculto ( $h_t$ ) final. Usando el ejemplo anterior de  $t = 5$ , en este paso el estado oculto de salida sería como un resumen de lo que sabe la red sobre los ángulos a este instante, listo para usarse en el siguiente paso o para predecir los ángulos futuros.

Al igual que con el modelo de visión artificial se usó un optimizador "Adam2" que asegura que los gradientes fluyan sin desvanecerse, permitiendo aprender dependencias a largo plazo. En cuanto a la tasa de aprendizaje y al número de épocas, se determinó por iteración también, estableciendo un valor de 0.0008 de tasa de aprendizaje y de 75 épocas. Con esta elección se consiguió que la red aprendiera los patrones complejos del péndulo y se evitó el sobreajuste.

Por último, se añadió una opción para imprimir por pantalla valores de error absoluto medio (MSE) y raíz del error cuadrático medio (RMSE). De esta forma se puede conocer la diferencia entre los ángulos predichos y los experimentales.

Victor Manuel Suárez Ferrer

---

Para analizar de forma más rigurosa el modelo de red neuronal que se había diseñado, se implementó al código la generación de ciertas graficas para complementar los valores numéricos. En primer lugar, se generó un histograma de errores que muestra la distribución y magnitud de los errores, revelando si estos son simétricos, sesgados o presentan valores atípicos; por ejemplo, un sesgo en  $\theta_2$  hacia errores positivos podría indicar problemas en la detección de los ángulos, indicando que ciertos valores negativos los convierte a positivos al pasar de cierto limite. En segundo lugar, se imprimió una gráfica de la evolución de la perdida. Esta grafica indica cómo el modelo converge durante el entrenamiento, ayudando a validar si las métricas reflejan un proceso estable, lo que es crucial para confiar en las métricas de MAE y RMSE como indicadores de un aprendizaje efectivo. Complementando esto, se añadió una gráfica de degradación de la precisión que permite evaluar cómo el error acumulado crece con el tiempo, mostrando la robustez de las predicciones a largo plazo y destacando posibles discontinuidades, como las introducidas por el reinicio periódico en los ángulos. A su vez, una gráfica del espacio de fases proporciona una visión cualitativa de la dinámica predicha frente a la medida, complementando los errores cuantitativos con información sobre la coherencia del movimiento, lo que puede revelar si las desviaciones en  $\theta_2$  afectan la dinámica global del sistema. Por último, se generó la gráfica más visual de todas, una comparación temporal entre valores medidos y predichos que ofrece una visión directa de patrones y desviaciones.

## Capítulo 5. RESULTADOS

### 5.1 Simulación Python

La simulación cumplió con su objetivo de representar el caos de un péndulo doble ante los pequeños cambios de posición, además de que su interfaz permite su uso para cualquier usuario que lea el proyecto y quiera replicar su propio péndulo con sus respectivas condiciones. A continuación, se muestra una captura de pantalla del resultado final de dicha simulación (Figura 18):

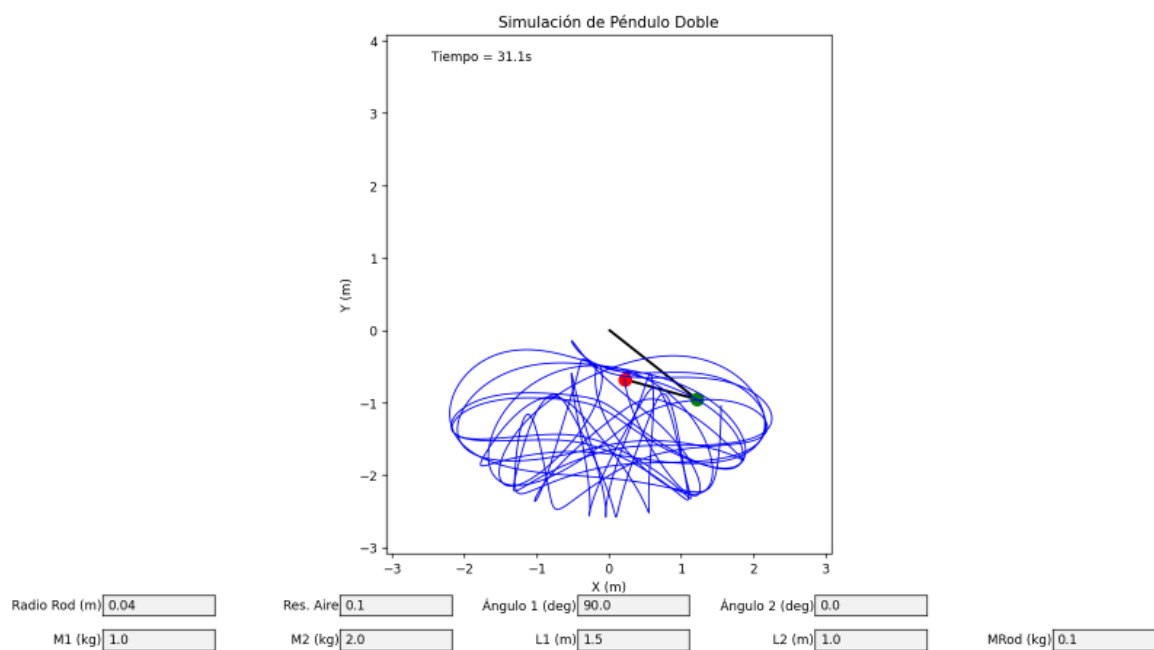


Figura 18 Captura de la interfaz gráfica de una simulación en Python que representa el movimiento caótico de un péndulo doble, permitiendo a los usuarios ajustar condiciones iniciales y visualizar trayectorias.

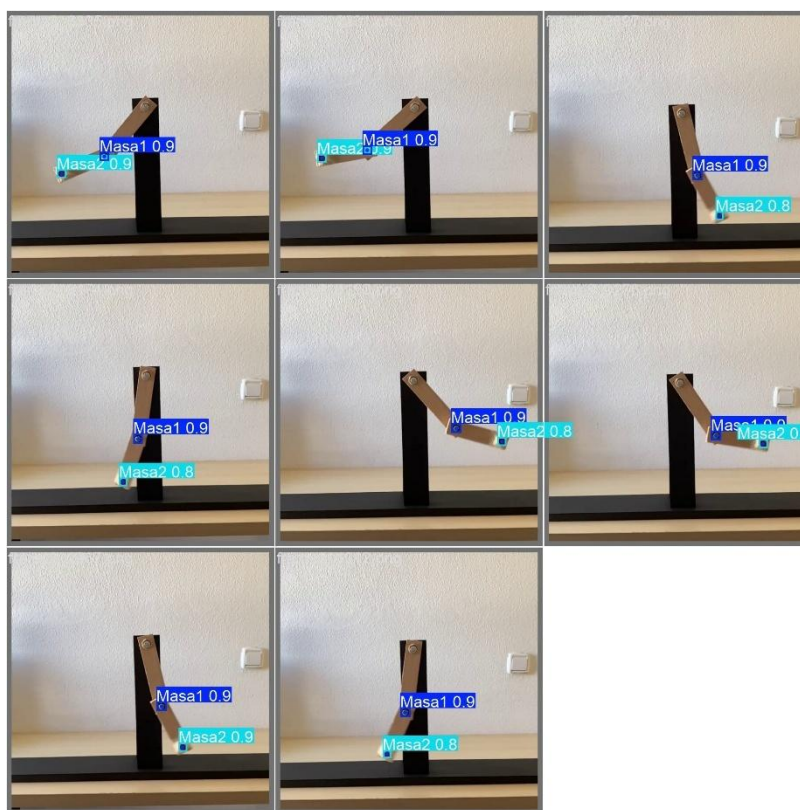
### 5.2 VISIÓN ARTIFICIAL

Para el conjunto de validación se obtuvieron las siguientes métricas:

Tabla 3 Métricas conjunto de validación en el entrenamiento 1

Entrenamiento 1	Validación			
<b>Bounding-Box</b>	Precisión	Sensibilidad	mAP@50	mAP@50:95
	0.9994	1.0000	0.9950	0.8194
<b>Key-points</b>	Precisión	Sensibilidad	mAP@50	mAP@50:95
	0.9994	1.0000	0.9950	0.9950

Estos valores indican un rendimiento excepcional en la detección de las masas del péndulo doble, con una precisión y sensibilidad cercanas al valor óptimo (1.0), lo que refleja que el modelo identifica correctamente casi todas las instancias de Masa1 y Masa2 presentes en las imágenes, con un número mínimo de falsos positivos. El mAP@50 de 0.9950 confirma una excelente capacidad de detección, mientras que el mAP@50:95 de 0.8194, aunque más bajo, sigue siendo elevado, indicando una buena precisión en la localización de los bounding boxes incluso a umbrales de IoU más estrictos (hasta 0.95), lo que es crucial para garantizar que las posiciones de las masas sean precisas para el análisis dinámico (Figura 19).



*Figura 19* Lote de imágenes del conjunto de validación con las predicciones del modelo superpuestas, mostrando bounding boxes y keypoints detectados para las masas del péndulo doble.

---

En cuanto a la estimación de keypoints, los resultados son igualmente sobresalientes, mostrando que el modelo no solo detecta las masas, sino que también predice con alta precisión la ubicación de sus puntos clave (centros de masa). Sin embargo, el hecho de que mAP@50 y mAP@50:95 sean idénticos (0.9950) resulta inusual y sugiere posibles limitaciones en el cálculo de las métricas o en las etiquetas de los keypoints. Esto puede deberse a que las etiquetas de los keypoints son demasiado simples: cada masa (Masa1 y Masa2) tiene un solo keypoint, y si este keypoint está etiquetado con una tolerancia alta (una región amplia donde se considera correcto), las predicciones siempre cumplen tanto el umbral de IoU 0.5 como los de 0.95, resultando en métricas idénticas. Alternativamente, aunque mucho menos probable, podría ser un error en la implementación de Ultralytics, donde el cálculo de mAP@50:95 para keypoints no

está correctamente diferenciado de  $mAP@50$ , lo que impide reflejar variaciones en la precisión de la localización a distintos umbrales de IoU.

En el conjunto de prueba, las métricas promedio para bounding boxes fueron:

Tabla 4 Métricas conjunto de prueba en el entrenamiento 1

Entrenamiento 1		Prueba		
Bounding-Box	Precisión	Sensibilidad	$mAP@50$	$mAP@50:95$
	0.9994	1.0000	0.9950	0.8135
Key-points	Precisión	Sensibilidad	$mAP@50$	$mAP@50:95$
	0.9990	1.0000	0.9950	0.9950

Ligeramente inferiores a las de validación, lo que indica una buena generalización pese a desafíos como borrosidad o iluminación. Para keypoints, las métricas fueron iguales a las de validación, mostrando un rendimiento excepcional.

Los resultados obtenidos son altamente satisfactorios para los objetivos del proyecto, ya que el modelo YOLOv11n-pose demuestra una capacidad sobresaliente para detectar las masas del péndulo doble y extraer sus puntos clave con una precisión y sensibilidad cercanas al 100%. Esto asegura que los keypoints generados son fiables para alimentar el modelo LSTM de predicción de movimiento caótico. Sin embargo, algunos aspectos requieren atención adicional. En primer lugar, la diferencia entre  $mAP@50$  y  $mAP@50:95$  para los bounding boxes (0.9950 frente a 0.8194 en validación, y 0.9950 frente a 0.8135 en prueba) indica que, aunque el modelo detecta correctamente la presencia de las masas, la precisión en la localización de los bounding boxes podría mejorarse, especialmente a umbrales de IoU más estrictos; esto podría deberse a etiquetas de bounding boxes que no son lo suficientemente precisas o a la detección ocasional del pivote como una masa adicional, un problema identificado al validar el modelo en un video

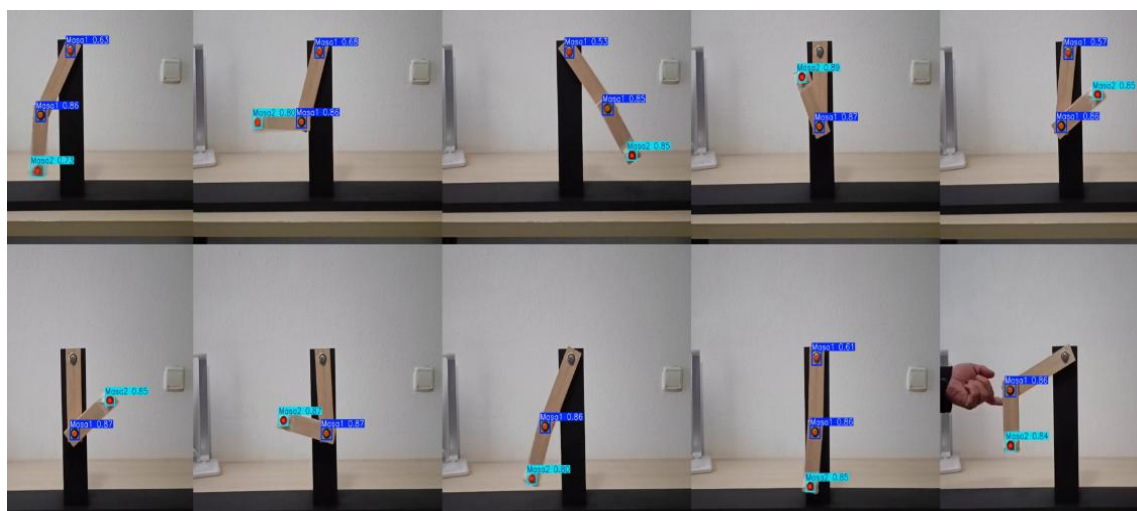


Figura 20 Mosaico de imágenes extraídas del video de prueba con sus etiquetas correspondientes. El video de prueba estaba grabado a una tasa de captación más baja, con menor calidad y con una posición ligeramente diferente al conjunto de videos de entrenamiento. Se aprecian los falsos positivos del eje principal como masa.

de prueba (Figura 20).

En segundo lugar y como ya se comentó anteriormente la igualdad persistente entre  $mAP@50$  y  $mAP@50:95$  para los keypoints es un indicativo de alguna especie de error. Finalmente, la ligera disminución del  $mAP@50:95$  en el conjunto de prueba (0.8135) respecto a la validación (0.8194) indica que el modelo tiene dificultades para generalizar a datos no vistos, posiblemente por variaciones en la calidad de las imágenes o un leve sobreajuste, lo que resalta la necesidad de optimizarlo para mejorar su rendimiento en escenarios reales. En el entrenamiento inicial, se utilizó un aumento de datos agresivo, lo que podría haber introducido transformaciones que hicieron que el pivote se asemejara a una masa en algunas imágenes procesadas, contribuyendo a los falsos positivos. Un *fine-tuning* sin aumento de datos puede abordar esto al entrenar con imágenes originales, reduciendo falsos positivos como la detección del pivote como masa y mejorando la discriminación y generalización del modelo en imágenes nuevas

Este resultado, llevo a una búsqueda más exhaustiva de cuál era el posible error y como mejorar el modelo. Para ello se examinaron una a una las gráficas generadas por YOLO en el apartado results (Figura 21)

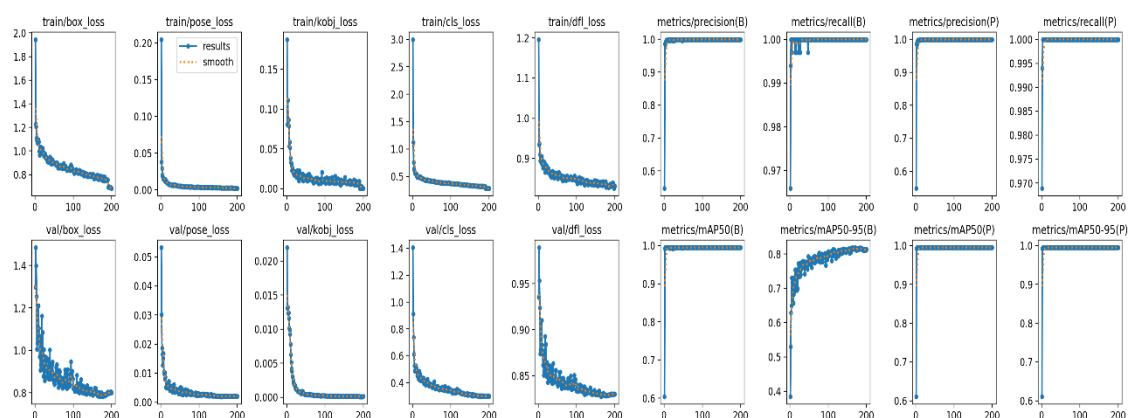


Figura 21 Gráficos de línea que muestran la evolución de las pérdidas y métricas de rendimiento durante el entrenamiento y la validación a lo largo de 200 épocas. Cada grafico incluye una línea azul solida ("results") que representa los valores medidos y una línea naranja discontinua ("smooth") que refleja una versión suavizada de los datos.

Las pérdidas disminuyen de manera consistente tanto en entrenamiento como en validación, y los valores finales son bajos, lo que indica que el modelo ha convergido correctamente. La similitud entre las pérdidas de entrenamiento y validación sugiere que no hay sobreajuste significativo, lo cual nos elimina una de las posibles causas de fallo. Sin embargo, la pérdida de clasificación (cls\_loss) sigue siendo relativamente alta (0.4 en validación), lo que podría estar relacionada con los falsos positivos del eje superior.

Las métricas de rendimiento como ya se había visto son excepcionales, por lo que todo apunta al aumento de datos. La configuración inicial (mosaic=1.0, flipud=0.5, fliplr=0.5, degrees=10.0, etc.) introduce transformaciones que podrían distorsionar las imágenes, haciendo que el pivote se asemeje a una masa en ciertas condiciones. Por ejemplo, el aumento de mosaico combina varias imágenes, lo que puede crear solapamientos o patrones artificiales que el modelo

interpreta como una masa adicional. Las rotaciones y flips también podrían alterar la percepción del pivote, confundiéndolo con las masas reales.

Todo esto conlleva a lo que ya se prevía, a realizar un fine tuning sin aumentación. En esta ocasión se redujo el número de épocas a 30, ya que se parte de pesos preentrenados, también se bajó la paciencia a 15 para evitar desperdiciar recursos y se optó por una tasa de aprendizaje menor con el objetivo de evitar cambios drásticos en los pesos, preservando el conocimiento previo. Por último, el umbral de confianza se fijó en 0.25 para hacer al modelo más selectivo y se desactivaron completamente todos los hiperparámetros de aumentación.

Los resultados que se obtuvieron no fueron los deseados, ya que las métricas seguían siendo excepcionales, aunque con leves diferencias (Figura 22) pero la detección de falsos positivos se disparó en más del doble (Figura 23).

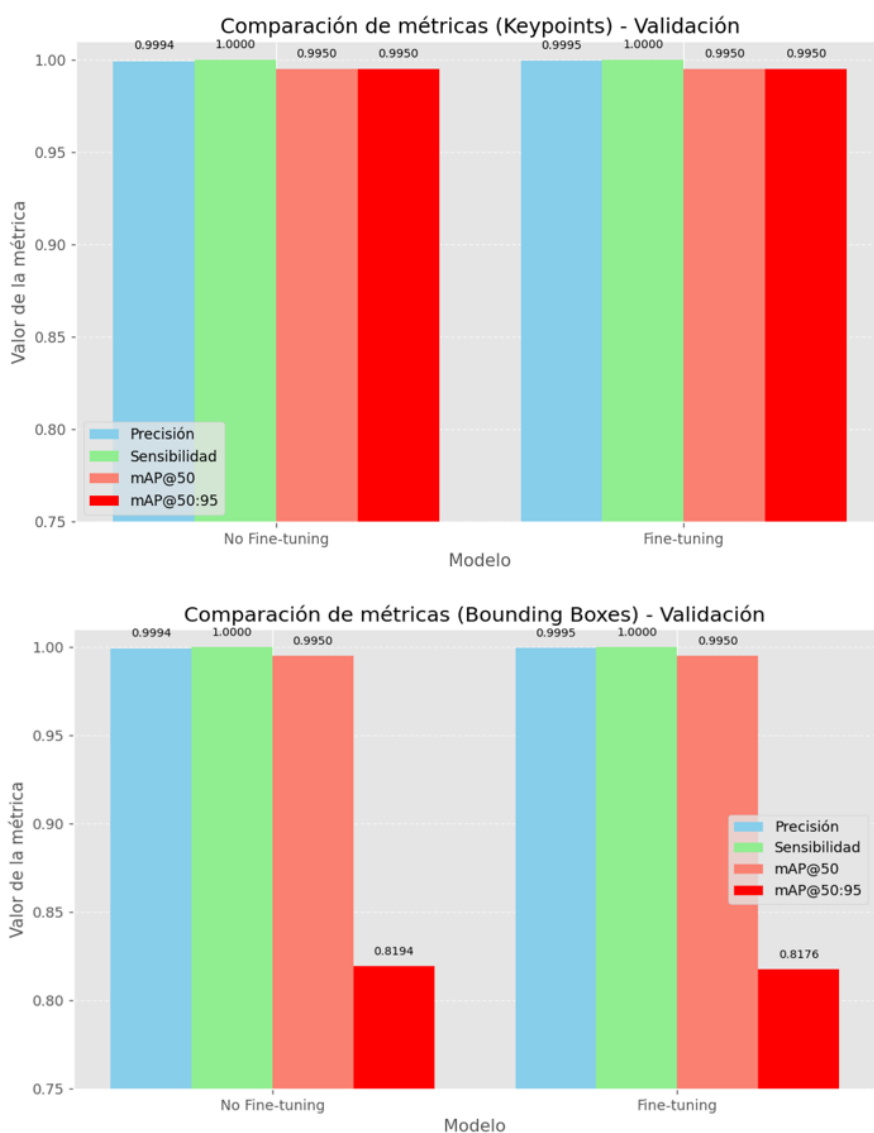


Figura 22 Gráfico de barras que compara las métricas de rendimiento (precisión, sensibilidad, mAP@50, mAP@50:95) entre el modelo inicial y el modelo ajustado mediante fine-tuning para la detección del péndulo doble.

Victor Manuel Suárez Ferrer

---

=== Análisis de detecciones incorrectas (Imágenes) ===	=== Análisis de detecciones incorrectas (Imágenes) ===
Total de frames procesados (imágenes): 213	Total de frames procesados (imágenes): 213
Frames con más de 2 masas (incorrectos): 38	Frames con más de 2 masas (incorrectos): 17
Porcentaje de frames incorrectos (imágenes): 17.84%	Porcentaje de frames incorrectos (imágenes): 7.98%

*Figura 23 A la derecha el porcentaje de frames incorrectos para el modelo fine tuning y a la derecha el valor de detecciones incorrectas para el primer modelo.*

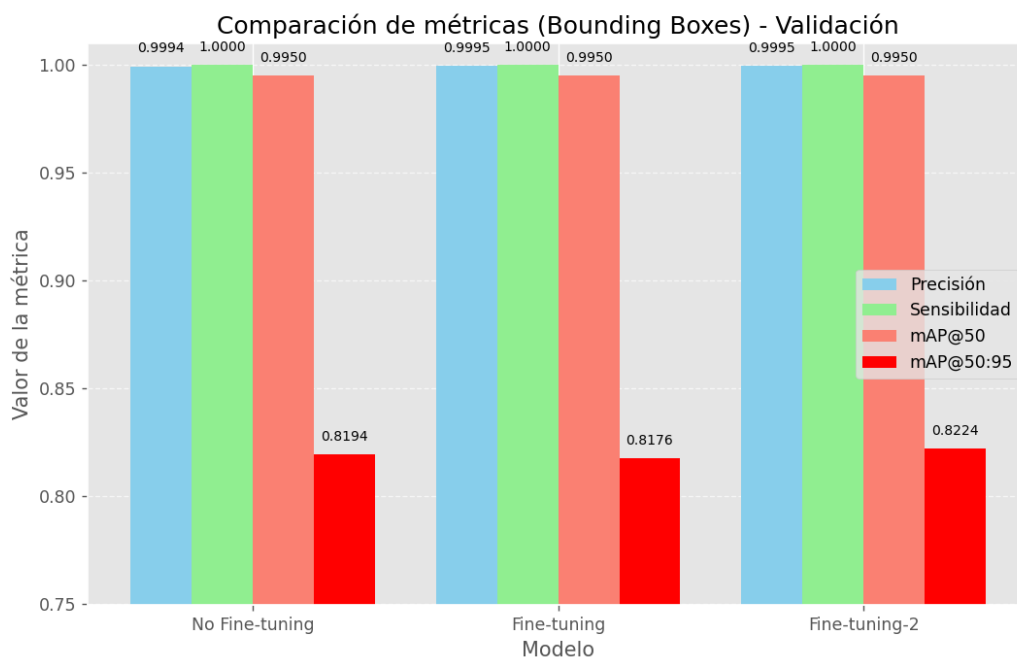
---

Es trivial ver que este preentrenamiento no dio el resultado que se buscaba, por lo que la posibilidad de que una aumentación tan severa estuviera confundiendo al modelo, también quedó parcialmente descartada. Sin embargo, analizando las gráficas del modelo se podía ver que había un sobreajuste, por lo que se dio una última oportunidad al fine-tuning sin aumentación cambiando el número de épocas de 30 a 50, dando más tiempo al modelo para converger y aprender a clasificar mejor las masas, reduciendo la tasa de aprendizaje a 0.0003 y aumentando el peso de pérdida de 1,5 a 2. El parámetro más importante introducido en este entrenamiento fue el umbral de IoU (antes no especificado ahora en 0.7). Un umbral más alto elimina detecciones superpuestas, que es lo que en un principio se cree que puede estar ocurriendo, de esta forma se reducen los falsos positivos.

Los resultados fueron excepcionales, se conservaron las métricas sobresalientes de los otros dos modelos y además se redujo en más de un 75% la detección incorrecta de frames con más de dos masas. Para un video de 1696 frames solo 34 de ellos fueron detecciones incorrectas (2%), unos valores que sobresalen por encima de los otros modelos. Sin embargo, para el objetivo final de entrenar una red neuronal con valores experimentales se necesitaba que este porcentaje fuera 0.

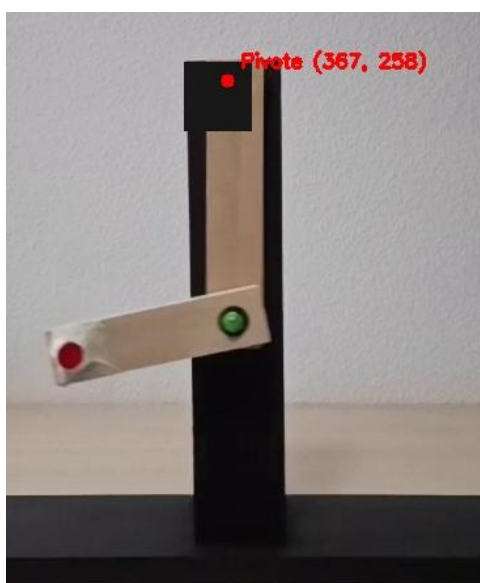
Se analizaron las imágenes obtenidas al aplicar el modelo al video de prueba y se pudo observar que los falsos positivos que aparecían tenían una confianza de entorno al 0.5-0.6, mientras que para las masas solía ser más alta del 0.7. Por esa misma razón, se subió la confianza a 0.5, tratando de eliminar los frames incorrectos restantes, pero sin perder detecciones válidas de Masa1 o Masa2. También se pensó en aumentar el IoU pero teniendo en cuenta la dinámica del péndulo que implica solapamientos en varios instantes, no sería una buena opción. Con todo esto, el modelo no mejoró la detección de frames incorrectos y además bajo la sensibilidad (recall), por lo que nunca se tuvo en cuenta.

Finalmente se optó por el segundo fine-tuning, ya que mejoraba las métricas (*Figura 24*) y además nos ofrecía el menor número de falsos positivos en cada uno de los frames. No obstante, tras revisar en repetidas ocasiones los frames del video de prueba se pudo comprobar que solo se anotaban en 2% de las veces bounding-boxes negativas, es decir, nunca se produjo una anotación de key-point errónea, lo que lo hace perfecto para filtrar los datos y poder obtener ángulos experimentales que nutran la red LSTM.



*Figura 24* Gráfico de barras que compara las métricas de rendimiento (precisión, sensibilidad, mAP@50, mAP@50:95) entre el modelo inicial, el primer fine-tuning y el último modelo de YOLOv11n-pose para la detección del péndulo doble.

Para el cálculo del pivote, se necesitaba que la probabilidad de un falso positivo de Masa1 fuera 0, ya que un valor erróneo aumentaría la desviación y afectaría a la estimación del pivote. Para corregir esto, se editó el video de prueba colocando una sección negra en donde estaba el pivote. Al implementar estos cálculos previos en un código de Python se obtuvo una estimación muy buena de la posición del pivote como se puede ver en la *Figura 25*.



*Figura 25* Frame de un video de prueba del péndulo doble mostrando la estimación de la posición del pivote central mediante el método de mínimos cuadrados, basada en la trayectoria de Masa1 detectada por YOLO.

Finalmente, como comprobar visualmente que los ángulos son correctos es muy complicado, se añadió unas líneas de código usando OpenCV (cv2) que permitiera ver sobre el video original las detecciones de YOLO por un lado y por otro una simulación de un péndulo basado en los ángulos experimentales extraídos (Figura 26)

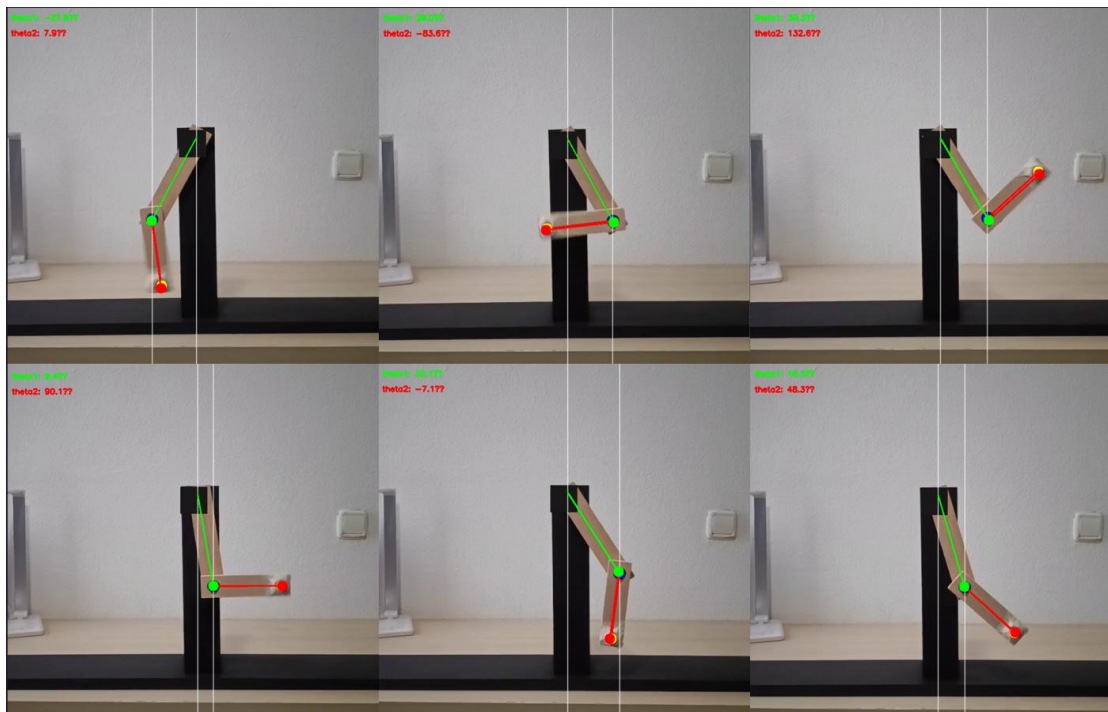


Figura 26 Representación gráfica de las coordenadas extraídas por Yolo (azul para Masa1 y amarillo para Masa2) vs la simulación de un péndulo doble en base a los ángulos extraídos (Masa1 en verde y Masa2 en rojo), así como de las líneas verticales que definen los ángulos. También se muestran los ángulos exactos en el instante del frame.

Como se puede apreciar la superposición es casi perfecta, lo que es un indicativo de que ya se cuenta con un modelo de visión artificial adecuado y con un método de extracción de ángulos igual de adecuado para entrenar la red LSTM.

### 5.3 RED NEURONAL LSTM

Los resultados que se obtuvieron fueron los siguientes:

```
MAE  $\theta_1$  (hasta datos medidos): 1.83 grados  
MAE  $\theta_2$  (hasta datos medidos): 7.67 grados  
RMSE  $\theta_1$  (hasta datos medidos): 2.32 grados  
RMSE  $\theta_2$  (hasta datos medidos): 9.66 grados
```

Figura 27 Resumen de los resultados del entrenamiento de la red neuronal LSTM, mostrando los valores de error absoluto medio (MAE) y error cuadrático medio (RMSE) para los ángulos  $\theta_1$  y  $\theta_2$  del péndulo doble

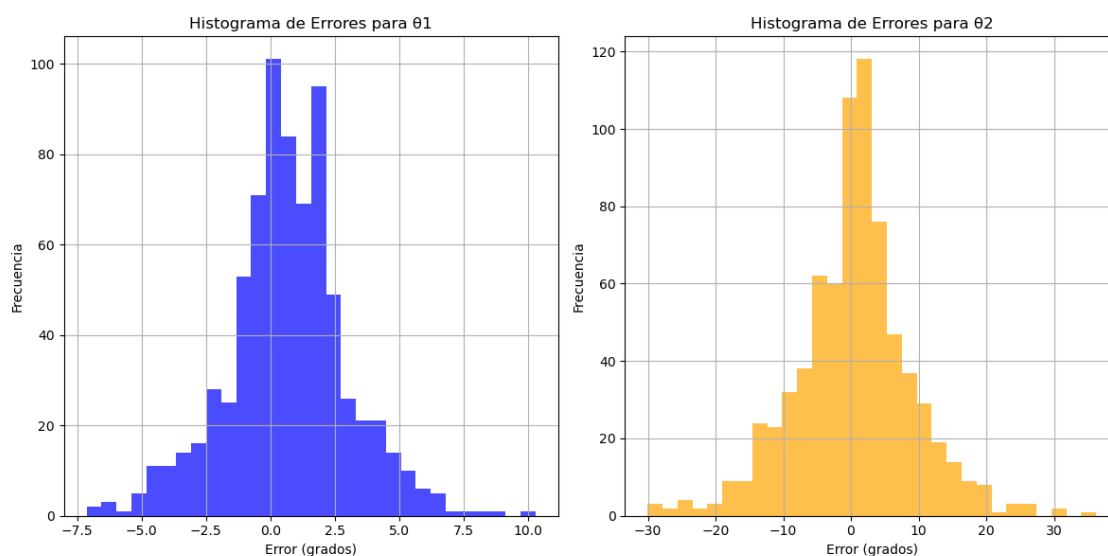
Un MAE de 1.83 grados y un RMSE de 2.32 grados indican que las predicciones para el ángulo del primer eslabón del péndulo doble son bastante precisas. Esto sugiere que el modelo LSTM

Victor Manuel Suárez Ferrer

captura bien la dinámica de  $\theta_1$ , probablemente porque el primer eslabón tiene un movimiento más estable y predecible en comparación con el segundo. Un MAE de 7.67 grados y un RMSE de 9.66 grados muestran un error significativamente mayor para el ángulo del segundo eslabón. Aunque un error de 7.67 grados sigue siendo razonable en términos absolutos, es mucho mayor que el del primer eslabón, esto indica que el modelo tiene más dificultades para predecir  $\theta_2$ . Esto puede deberse a la dinámica más caótica y sensible del segundo eslabón, que está influenciado tanto por su propia inercia como por el movimiento del primer eslabón.

Por otro lado, la diferencia entre el MAE (1.83) y el RMSE (2.32) es relativamente pequeña (0.49 grados). Dado que el RMSE penaliza más los errores grandes al elevar las diferencias al cuadrado, esta diferencia sugiere que los errores en las predicciones de  $\theta_1$  son bastante uniformes, con pocos errores grandes. Sin embargo, la diferencia entre el MAE (7.67) y el RMSE (9.66), implica que hay algunos errores más grandes que están influyendo en la detección de  $\theta_2$ . Esto sugiere que el modelo comete errores significativos en ciertos puntos de la predicción, posiblemente en momentos donde la dinámica del segundo eslabón cambia abruptamente o donde la dinámica se vuelve extremadamente caótica.

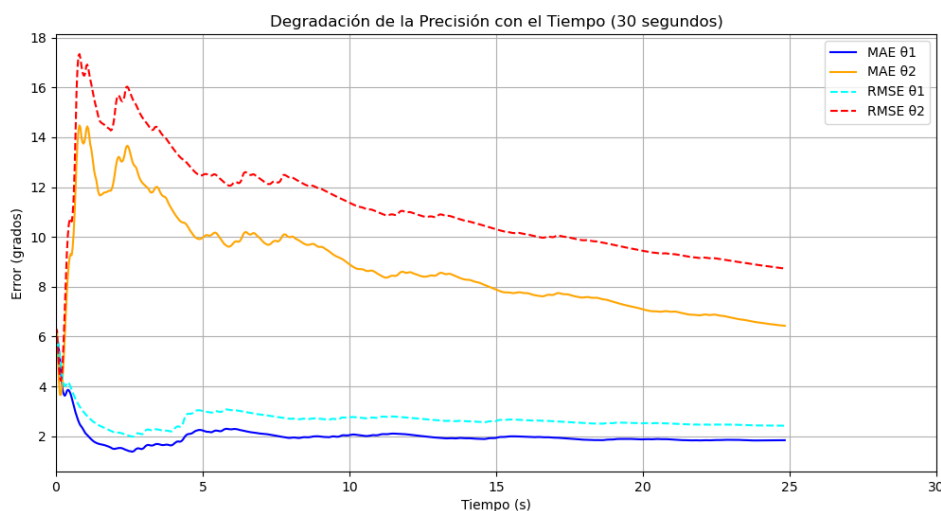
En cuanto a las gráficas generadas, el histograma de errores mostro para  $\theta_1$  una distribución con una forma aproximadamente simétrica, con un pico pronunciado ubicado entre 0 y 2.5 grados, lo que indica que la mayoría de las predicciones están muy cerca de los valores experimentales. No obstante, se observa una ligera cola hacia valores negativos que se extiende hasta aproximadamente -7.5 grados, junto con una cola más corta hacia valores positivos hasta 5 grados. En contraste, el histograma de  $\theta_2$  muestra una distribución sesgada hacia la derecha, con un pico prominente entre 0 y 10 grados, pero con una cola extensa que alcanza errores de hasta 30 grados, evidenciando la presencia de predicciones con desviaciones sustanciales en ciertos casos.



*Figura 28* Histograma que muestra la distribución de los errores de predicción de la red LSTM para los ángulos  $\theta_1$  y  $\theta_2$  del péndulo doble, comparando la precisión entre el primer y segundo eslabón.

La ligera asimetría hacia errores negativos sugiere una tendencia del modelo a subestimar  $\theta_1$  en ciertas condiciones, lo que podría estar relacionado con un sesgo sistemático en los datos de entrenamiento o con la incapacidad del modelo para capturar completamente ciertas dinámicas específicas del primer brazo del péndulo, como transiciones rápidas o efectos de fricción no modelados. Por otro lado, el histograma de  $\theta_2$  refleja un desempeño más variable. El pico entre 0 y 10 grados indica que muchas predicciones son razonablemente precisas, pero la cola mayor hacia 30 grados señala la presencia de errores significativos en el subconjunto de las predicciones.

Teniendo en cuenta la dinámica del péndulo en los videos donde se estaba aplicando la red neuronal, lo más probable es que esa desviación se debería a los primeros instantes donde el movimiento es mucho más caótico. Para comprobar esta hipótesis se generó una gráfica para la degradación de la precisión (*Figura 29*).

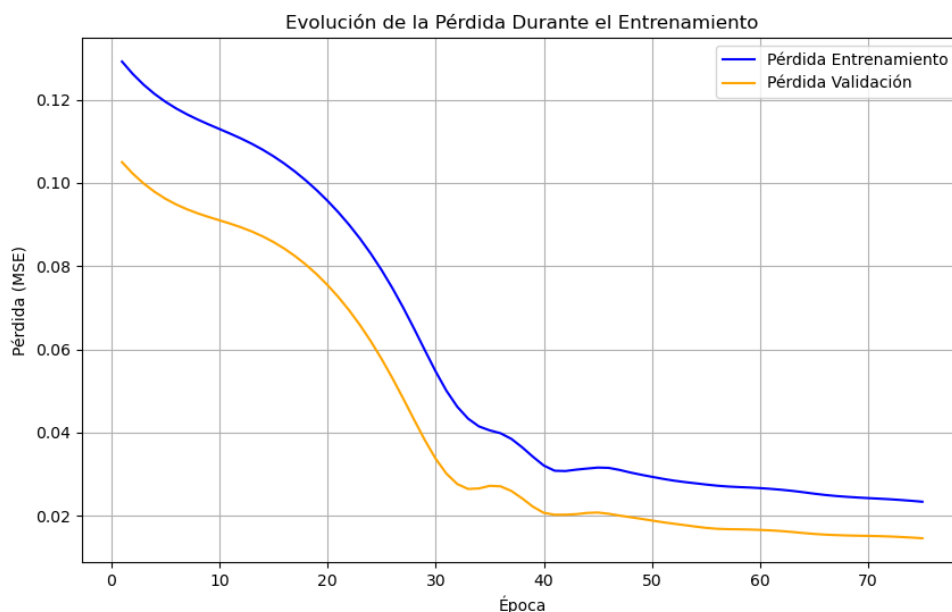


*Figura 29* Gráfico de líneas que muestra la evolución temporal de las métricas de error acumulado (MAE y RMSE) de la red LSTM para los ángulos  $\theta_1$  y  $\theta_2$  del péndulo doble, a lo largo de un intervalo de 30 segundos.

Como se esperaba, el pico inicial elevado en ambas métricas sugiere que el modelo enfrenta dificultades significativas al comienzo de la predicción, esto revela que le cuesta predecir ángulos cuando el movimiento es rápido y extremadamente caótico como se puede ver en el comportamiento de  $\theta_2$ .

Por otro lado, al igual que pasaba en el modelo de visión artificial, otra hipótesis es que estos errores se podían deber a un sobreajuste que no permitiera al modelo aprender la dinámica real y compleja del péndulo, sin embargo, la gráfica de evolución de la pérdida (*Figura 30*) muestra que inicialmente, ambas curvas parten de valores relativamente altos: aproximadamente 0.12 para el conjunto de entrenamiento y 0.10 para el de validación en la época 0. Durante las primeras 10 épocas, se observa una disminución abrupta y casi exponencial, cayendo a valores cercanos a 0.04. A partir de ahí, la reducción se vuelve más gradual, y después de unas 20 épocas, ambas curvas se estabilizan en un rango bajo, entre 0.02 y 0.03, manteniéndose casi constantes

hasta las 75 épocas totales. Esto se traduce en que no hay evidencia de divergencia significativa entre las curvas de entrenamiento y validación, lo que sugiere una ausencia de sobreajuste.



*Figura 30* Gráfica de la evolución de pérdida. La curva azul representa la pérdida calculada sobre el conjunto de entrenamiento, mientras que la curva naranja corresponde al conjunto de validación

---

La rápida caída inicial de la pérdida en ambas curvas indica que el modelo LSTM aprende eficientemente los patrones fundamentales presentes en los datos del péndulo doble durante las primeras etapas del entrenamiento. La pérdida residual de 0.02-0.03 implica que, aunque el modelo es efectivo, no elimina completamente el error, lo que es esperable dado el carácter caótico del sistema. La similitud entre las curvas de entrenamiento y validación es una señal positiva de generalización. Si la pérdida de entrenamiento hubiera continuado disminuyendo mientras la de validación aumentaba, esto habría indicado sobreajuste, pero la estabilidad conjunta sugiere que el modelo no está memorizando los datos de entrenamiento, sino que captura patrones aplicables a datos no vistos.

Teniendo una visión más clara de las debilidades del modelo, ahora toca ver cómo se ajusta realmente a los ángulos experimentales (*Figura 31*).

En líneas generales se puede apreciar una muy buena aproximación de la red neuronal, lo que ya se podía intuir con las métricas MSE y RMSE tan buenas que se obtuvieron. En el espacio de fases vemos una especie de atractor en 10 grados, lo que podría corresponder a un equilibrio parcial o a una oscilación dominante en los datos. Esto seguramente se deba a los efectos disipativos que sufre el péndulo a los pocos segundos de ser soltado, reduciendo el movimiento a pequeñas oscilaciones que seguramente sean de este orden. Por otro lado, estas gráficas nos permiten apreciar que a pesar de que las métricas para  $\theta_2$  fueran significativamente peores que

Victor Manuel Suárez Ferrer

para  $\theta_1$ , el modelo es capaz de adaptarse muy bien a la dinámica caótica del péndulo incluso en los primeros segundos donde el movimiento es mucho más frenético.

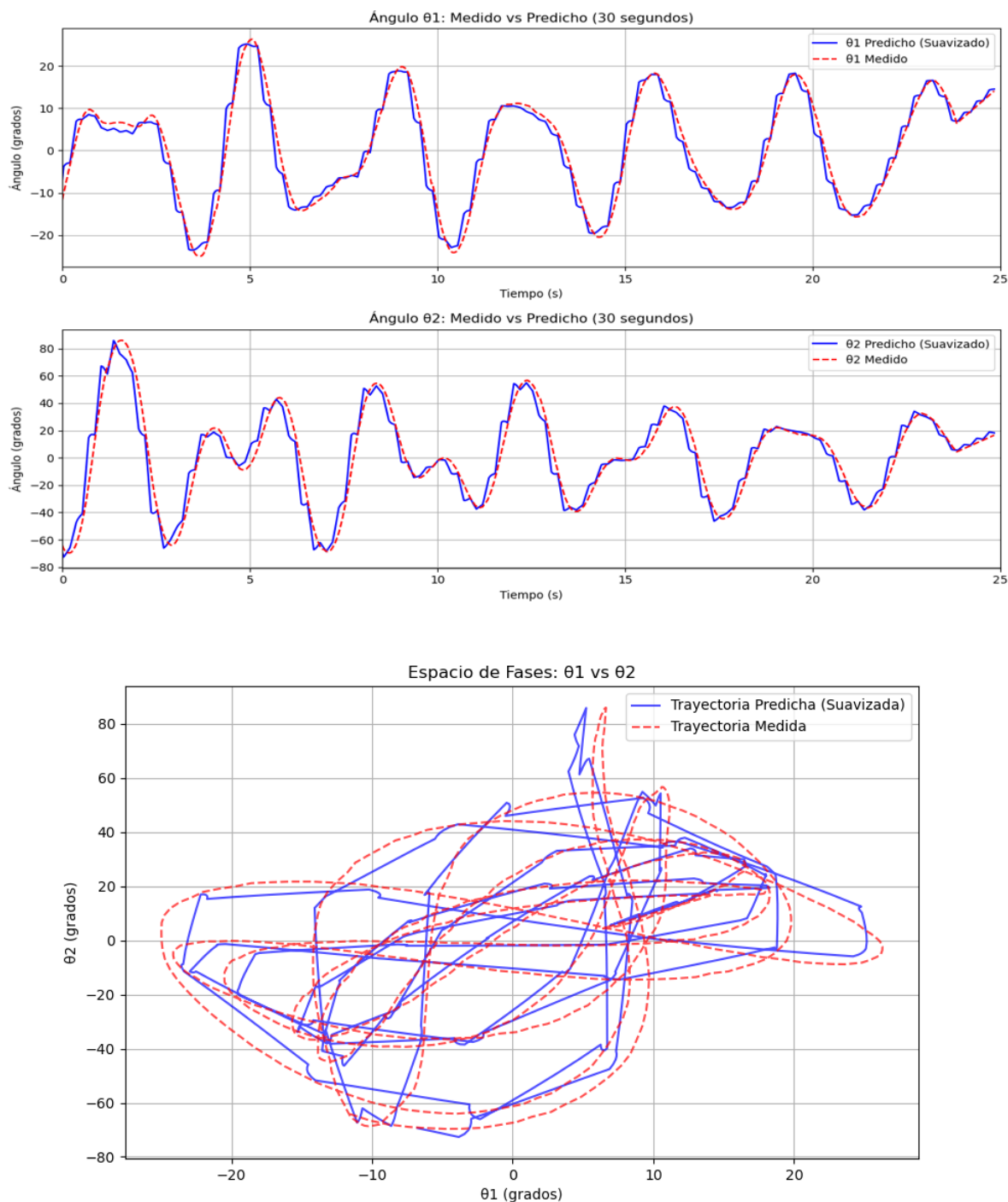


Figura 31 Gráficas de comparación entre ángulos. En azul la dinámica predicha por el modelo y en naranja y línea discontinua los ángulos experimentales medidos por la VA

## Capítulo 6. DISCUSIÓN

### 6.1 LIMITACIONES DEL ESTUDIO

El péndulo doble, como sistema caótico, exhibe una extrema sensibilidad a las condiciones iniciales, lo que significa que incluso diferencias mínimas en las mediciones de posición o velocidad iniciales pueden generar trayectorias completamente divergentes a largo plazo. Esta propiedad intrínseca del caos impone un límite fundamental a la capacidad predictiva de cualquier modelo, ya que los errores iniciales, por pequeños que sean, se amplifican exponencialmente con el tiempo. En este estudio, el modelo LSTM logró capturar con precisión las trayectorias del péndulo a corto plazo, pero su rendimiento decayó significativamente en predicciones extendidas. Comparado con métodos numéricos como el Runge-Kutta de cuarto orden (RK4), que resuelve las ecuaciones diferenciales del sistema con alta fidelidad cuando las condiciones iniciales son conocidas con exactitud, el enfoque basado en datos del LSTM resulta menos preciso. Sin embargo, en este caso, las condiciones iniciales se derivaron de datos experimentales, lo que introdujo incertidumbre adicional y limitó aún más la ventaja teórica de métodos como RK4.

Además, los datos experimentales utilizados para entrenar el modelo LSTM provinieron de un modelo de visión artificial que, aunque alcanzó una precisión impresionante del 99.94% en la detección de *keypoints*, no estuvo exento de problemas. Se observaron falsos positivos persistentes, como la detección errónea del pivote o la mano como masas adicionales, lo que requirió intervenciones manuales como la edición de videos para ocultar elementos no deseados. Este proceso, aunque efectivo, no es óptimo para un sistema que idealmente debería ser completamente automatizado. Por otro lado, aunque se emplearon 800 imágenes para entrenar el modelo YOLO, la complejidad de la dinámica caótica del péndulo doble sugiere que este número podría no haber sido suficiente. Un *dataset* más amplio y diverso, que incluyera una mayor variedad de configuraciones iniciales y regímenes caóticos, habría permitido al modelo generalizar mejor y capturar con mayor precisión la riqueza del comportamiento del sistema

### 6.2 LIMITACIONES DE LA TECNOLOGIA USADA

El desarrollo del proyecto se vio restringido por varias limitaciones tecnológicas que afectaron tanto la calidad de los datos como el desempeño de los modelos de inteligencia artificial empleados. En primer lugar, la construcción física del péndulo doble no contó con rodamientos de alta precisión, lo que resultó en rozamientos no controlados que alteraron la dinámica del sistema. Aunque estos efectos se intentaron modelar en las simulaciones, las discrepancias entre los datos experimentales y los simulados persistieron. Asimismo, la falta de un sistema de lanzamiento estandarizado impidió determinar con exactitud las condiciones iniciales, como el ángulo y la velocidad de lanzamiento, lo que dificultó la validación de las predicciones frente a los datos reales.

Otro factor limitante fue la cámara utilizada para registrar el movimiento del péndulo. Con una resolución de 720 píxeles y una tasa de 240 frames por segundo (fps), se capturó eficazmente el movimiento del péndulo doble en la mayoría de los escenarios, sin embargo, no logró capturar detalles finos del movimiento, especialmente en fases de alta velocidad o cuando las masas del péndulo se superponían visualmente. Una cámara con mayor resolución (por ejemplo, 1080p) y una tasa de fps más alta (como 480 fps) habría mejorado la calidad de los datos, permitiendo un análisis más detallado y fiable.

En cuanto a los modelos de IA, el modelo YOLOv11n-pose, pese a su efectividad tras un fine-tuning exhaustivo, mostró falsos positivos recurrentes, como la confusión del pivote o la mano con masas del péndulo. Estas detecciones erróneas requirieron correcciones manuales, lo que introdujo ruido en el dataset y afectó al entrenamiento del modelo LSTM. Por su parte, la arquitectura LSTM demostró ser adecuada para capturar dependencias temporales a corto plazo, pero su capacidad para predecir a largo plazo se vio limitada por la naturaleza caótica del sistema. En sistemas caóticos, el horizonte predictivo es finito, y los errores se acumulan exponencialmente, un efecto que se hizo más evidente en la predicción del segundo eslabón del péndulo, cuya dinámica es más sensible a las perturbaciones. Aunque la LSTM es una herramienta poderosa para series temporales, su aplicación a dinámicas caóticas requiere técnicas adicionales para mitigar estas limitaciones.

### **6.3 CAMBIO RESPECTO A LOS OBJETIVOS INICIALES**

La metodología inicialmente resultó efectiva para modelar el comportamiento caótico del péndulo doble sin recurrir a ecuaciones dinámicas, no obstante, en un principio se previó entrenar un modelo estándar de YOLO con su dataset correspondiente. Sin embargo, fue necesario un fine-tuning para reducir falsos positivos y mejorar la detección de keypoints. Esto incluyó múltiples iteraciones de entrenamiento y ajustes en hiperparámetros.

No se previó tener que usar una bounding-box extra para el pivote, por lo que el ajuste para su detección y el tiempo que conllevo tampoco. Además, se desarrolló un código específico para calcular los ángulos a partir de las posiciones detectadas, un paso no contemplado inicialmente.

La metodología propuesta inicialmente resultó efectiva para modelar el comportamiento caótico del péndulo doble sin depender de ecuaciones dinámicas explícitas, cumpliendo así el objetivo principal del proyecto. Sin embargo, durante la implementación surgieron desafíos imprevistos que obligaron a ajustar el enfoque original. Se había planeado utilizar un modelo estándar de YOLO con su dataset predeterminado para la detección de keypoints, pero este presentó un número elevado de falsos positivos, particularmente en la identificación del pivote y la mano como elementos del péndulo. Para solucionar esto, se llevó a cabo un fine-tuning intensivo, que incluyó múltiples iteraciones de entrenamiento, ajustes en hiperparámetros como la tasa de aprendizaje y el umbral de confianza, y la desactivación de ciertas técnicas de aumento de datos que generaban confusión. Este proceso, aunque necesario, no estaba contemplado en el plan inicial y extendió significativamente el tiempo de desarrollo.

Además, no se había previsto la necesidad de implementar una bounding-box adicional para detectar el pivote ni de desarrollar un método basado en mínimos cuadrados para estimar su posición con precisión. Este paso extra fue esencial para calcular los ángulos del péndulo a partir de las posiciones detectadas, pero implicó un esfuerzo adicional en términos de programación y validación. Asimismo, la extracción de los ángulos experimentales requirió un algoritmo basado en la función arcotangente y la geometría del sistema, un desarrollo no considerado inicialmente que añadió complejidad al procesamiento de datos. Estos cambios, aunque permitieron alcanzar los objetivos, resaltaron la necesidad de una planificación adaptable en proyectos que integran hardware experimental y técnicas de IA.

## **6.4 IMPACTO DEL PROYECTO**

El proyecto demostró con éxito que es posible predecir el comportamiento de un sistema caótico como el péndulo doble utilizando inteligencia artificial, sin necesidad de recurrir a sus ecuaciones dinámicas subyacentes. Este avance tiene implicaciones profundas para la física y la ingeniería, especialmente en el estudio de sistemas complejos o poco conocidos donde derivar modelos matemáticos es costoso. Al integrar visión artificial y redes neuronales recurrentes, este enfoque basado en datos ofrece una alternativa no invasiva para analizar dinámicas caóticas, con aplicaciones potenciales en campos como la predicción de flujos turbulentos, el modelado de sistemas biológicos o incluso el análisis de mercados financieros.

En términos prácticos, este enfoque podría aplicarse al control de sistemas dinámicos en robótica, como brazos manipuladores que enfrentan perturbaciones externas, o en meteorología, para predecir patrones climáticos caóticos a partir de datos observacionales. Incluso podría explorarse en la música generativa, donde las trayectorias del péndulo se convierten en secuencias sonoras caóticas pero estructuradas, ofreciendo una nueva forma de composición algorítmica.

## Capítulo 7. CONCLUSIONES

### 7.1 Conclusiones del trabajo

El propósito central fue diseñar e implementar un sistema capaz de predecir la evolución temporal del movimiento caótico de un péndulo doble. Para ello, se integró un modelo de visión artificial basado en YOLO11n-pose para la extracción de datos experimentales a partir de grabaciones visuales, y una red neuronal recurrente LSTM para modelar y predecir la dinámica temporal del sistema. Este enfoque permitió resolver el problema sin depender de ecuaciones físicas explícitas, lo que representa un avance significativo en la aplicación de IA a problemas de física experimental.

El modelo de visión artificial YOLO11n-pose se entrenó y optimizó para detectar con precisión las masas del péndulo doble en cada fotograma. Los resultados obtenidos fueron notablemente robustos: en el conjunto de validación, el modelo alcanzó una precisión de 0.9994 y una sensibilidad de 1.0000, lo que demuestra la fiabilidad para extraer datos precisos y consistentes.

A pesar de este rendimiento sobresaliente, se observaron ciertas limitaciones en el conjunto de prueba, donde el  $mAP@50:95$  disminuyó ligeramente a 0.8135. Este descenso puede atribuirse a pequeñas variaciones en las condiciones experimentales no capturadas completamente durante el entrenamiento, como cambios sutiles en la velocidad del movimiento o ruido visual. Aunque lo más probable es que se debiera a la aparición de falsos positivos con las detecciones erróneas del pivote central como masa y el desajuste que generaba la aparición de una mano en los primeros instantes. Sin embargo, estas limitaciones no comprometen la utilidad del modelo, ya que, tras ciertos ajustes, los datos generados fueron suficientemente precisos para alimentar la siguiente etapa del sistema: la predicción temporal mediante la red LSTM.

La red neuronal LSTM, por su parte, fue diseñada para capturar las relaciones temporales en los ángulos del péndulo extraídos por el modelo de visión artificial. Tras un proceso exhaustivo de entrenamiento y ajuste de hiperparámetros, la LSTM logró predecir los ángulos del primer eslabón con un error medio absoluto (MAE) de  $1.83^\circ$ , un resultado que destaca su capacidad para modelar movimientos relativamente estables. Para el segundo eslabón, donde el comportamiento caótico es más pronunciado, el MAE aumentó a  $7.67^\circ$ , y el error cuadrático medio (RMSE) alcanzó  $9.66^\circ$ . Estos valores reflejan la dificultad inherente de predecir dinámicas altamente sensibles al caos, pero también subrayan el éxito del modelo al capturar patrones generales en un sistema tan complejo sin recurrir a un modelo físico tradicional.

Los resultados obtenidos tienen implicaciones profundas tanto para la física como para la inteligencia artificial. En el ámbito de la física, este trabajo demuestra que es posible analizar y predecir el comportamiento de sistemas caóticos utilizando únicamente datos experimentales procesados por IA, eliminando la necesidad de derivar ecuaciones diferenciales complejas o de realizar simulaciones numéricas intensivas. Este enfoque podría aplicarse a otros sistemas físicos no lineales, como fluidos turbulentos o fenómenos climáticos. Desde la perspectiva de la IA, el proyecto resalta el potencial de combinar visión artificial y redes neuronales recurrentes para resolver problemas del mundo real.

## Capítulo 8. FUTURAS LÍNEAS DE TRABAJO

Una de las principales áreas de mejora se centra en el modelo de visión artificial. Aunque el modelo alcanzó una precisión sobresaliente, la detección de keypoints podría optimizarse mediante la incorporación de arquitecturas alternativas como HRNet, que ofrece mayor resolución espacial y podría mejorar la localización precisa de las articulaciones del péndulo, especialmente en condiciones donde el movimiento se vuelve más caótico, que como ya se ha visto presenta ciertas limitaciones. Además, sería valioso trabajar en la automatización completa del proceso de detección, eliminando la necesidad de intervenciones manuales como tapar el pivote para evitar falsos positivos o eliminar frames iniciales para evitar la detección de la mano. Esto podría lograrse integrando técnicas de segmentación semántica que identifiquen y excluyan automáticamente elementos no deseados en el video o reetiquetando todo el conjunto de datos de nuevo. Otra mejora importante sería ampliar el dataset de entrenamiento para incluir una mayor variedad de condiciones experimentales, como diferentes niveles de iluminación, velocidades del péndulo o resoluciones de video, lo que contribuiría a no tener que usar una aumentación tan extrema y que se produzcan esas detecciones no deseadas.

En cuanto a la predicción temporal, la red neuronal LSTM demostró ser efectiva para capturar patrones a corto plazo, con un MAE de  $1.83^\circ$  para el primer eslabón, pero presentó limitaciones en el segundo eslabón (MAE de  $7.67^\circ$ ) debido a la naturaleza caótica del sistema. Una línea de trabajo prometedora sería explorar arquitecturas alternativas, como redes GRU (Gated Recurrent Units), que han mostrado ventajas en la modelización de series temporales complejas y podrían mejorar la precisión, especialmente en regímenes altamente caóticos. Además, enriquecer el conjunto de datos de entrada incorporando variables adicionales, como la velocidad angular, el momento angular o la energía total del sistema, podría proporcionar al modelo información más completa sobre la dinámica, reduciendo los errores de predicción. Otro aspecto por investigar es la predicción a largo plazo, que actualmente está limitada por la acumulación exponencial de errores en sistemas caóticos. Esto podría extender el horizonte predictivo, haciendo que el sistema sea más útil para aplicaciones prácticas.

El enfoque desarrollado en este TFG tiene un gran potencial para aplicarse a otros sistemas dinámicos más allá del péndulo doble. Una línea de trabajo interesante sería generalizar el sistema a otros sistemas caóticos, como movimientos brownianos, sistemas de fluidos turbulentos o modelos simplificados de fenómenos climáticos, lo que ampliaría el impacto del proyecto en campos como la física, la meteorología o la ingeniería. En el ámbito de la robótica, los resultados podrían adaptarse para mejorar el control de sistemas mecánicos complejos, como brazos robóticos sometidos a perturbaciones externas, permitiendo una mayor estabilidad y precisión en entornos dinámicos.

Otro aspecto relevante para futuros desarrollos es la creación de una interfaz interactiva que permita a los usuarios cargar videos de péndulos dobles y obtener predicciones en tiempo real, junto con visualizaciones detalladas de los resultados, como trayectorias, ángulos predichos o errores de predicción. Esta interfaz podría servir como una herramienta educativa para enseñar conceptos de caos, dinámica no lineal y aprendizaje automático.

## Capítulo 9. REFERENCIAS

- [1]. Myers AD, Tempelman JR, Petrushenko D, Khasawneh FA. Low-cost double pendulum for high-quality data collection with open-source video tracking and analysis. HardwareX [Internet]. 2020 Sep 11 [cited 2025 May 7];8:e00138–8. Available from: [https://www.hardware-x.com/article/S2468-0672\(20\)30047-X/fulltext](https://www.hardware-x.com/article/S2468-0672(20)30047-X/fulltext)
- [2]. Chattopadhyay A, Hassanzadeh P, Pasha S. Predicting clustered weather patterns: A test case for applications of convolutional neural networks to spatio-temporal climate data. Scientific Reports [Internet]. 2020 Jan 28 [cited 2025 May 7];10(1). Available from: <https://www.nature.com/articles/s41598-020-57897-9>
- [3]. Ionescu VS, Czibula G, Eugen Mihuleț. DeePS at: A deep learning model for prediction of satellite images for nowcasting purposes. Procedia Computer Science [Internet]. 2021 Jan 1 [cited 2025 May 7];192:622–31. Available from: <https://www.sciencedirect.com/science/article/pii/S1877050921015519?via%3Dihub>
- [4]. Fan H, Jiang J, Zhang C, Wang X, Lai YC. Long-term prediction of chaotic systems with recurrent neural networks [Internet]. arXiv.org. 2020 [cited 2025 May 7]. Available from: <https://arxiv.org/abs/2004.01258>
- [5]. Sangiorgio M, Dercole F, Giorgio Guariso. Forecasting of noisy chaotic systems with deep neural networks. Chaos Solitons & Fractals [Internet]. 2021 Nov 16 [cited 2025 May 7];153:111570–0. Available from: <https://www.sciencedirect.com/science/article/pii/S0960077921009243?via%3Dihub>
- [6]. Wang M, Li J. Interpretable predictions of chaotic dynamical systems using dynamical system deep learning. Scientific Reports [Internet]. 2024 Feb 7 [cited 2025 May 7];14(1). Available from: <https://www.nature.com/articles/s41598-024-53169-y>
- [7]. MR 09 Ligaduras - laplace [Internet]. Laplace.us.es. 2023 [cited 2025 May 7]. Available from: [http://laplace.us.es/wiki/index.php?title=MR\\_09\\_Ligaduras](http://laplace.us.es/wiki/index.php?title=MR_09_Ligaduras)
- [8]. Velasco E. MECÁNICA LAGRANGIANA y HAMILTONIANA Universidad Autónoma de Madrid 1 Contents [Internet]. Available from: <http://casanchi.org/fis/mlagrangiana01.pdf>
- [9]. Agencia Estatal de Meteorología (AEMET). Física del caos en la predicción meteorológica. Madrid: AEMET; 2012 [citado 7 May 2025]. Available from: [https://www.aemet.es/documentos/es/conocerlas/recursos\\_en\\_linea/publicaciones\\_y\\_estudios/publicaciones/Fisica\\_del\\_caos\\_en\\_la\\_predicc\\_meteo/Fisica\\_del\\_caos\\_en\\_la\\_prediccion\\_meteorologica\\_baja.pdf](https://www.aemet.es/documentos/es/conocerlas/recursos_en_linea/publicaciones_y_estudios/publicaciones/Fisica_del_caos_en_la_predicc_meteo/Fisica_del_caos_en_la_prediccion_meteorologica_baja.pdf)
- [10]. Parte II2-2. [Internet]. Cartagena99.com; [s.f.] [citado 7 May 2025]. Available from: <https://www.cartagena99.com/recursos/alumnos/apuntes/Parte%20II2-2.pdf>
- [11]. S: Nonconservative systems (Summary) [Internet]. Physics LibreTexts. 2020 [cited 2025 May 7]. Available from: [https://phys.libretexts.org/Bookshelves/Classical\\_Mechanics/Variational\\_Principles\\_in\\_Classic](https://phys.libretexts.org/Bookshelves/Classical_Mechanics/Variational_Principles_in_Classic)

[al Mechanics \(Cline\)/10%3A Nonconservative Systems/10.S%3A Nonconservative systems \(Summary\)](#)

[12]. TEORÍA DEL CAOS [Internet]. APRENDEMOS MATEMÁTICAS. 2015 [cited 2025 May 7]. Available from: <https://www3.gobiernodecanarias.org/medusa/ecoblog/mrodperv/fractales/teoria-del-caos/>

[13]. Alex W, Roldan B, Huincho FV. Artificial neural networks: a measurement of forecast learnings as potential demand. Universidad Ciencia y Tecnología [Internet]. 2023 Feb 27 [cited 2025 May 13];27(118):51–60. Available from: [https://ve.scielo.org/scielo.php?script=sci\\_arttext&pid=S1316-48212023000100051](https://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212023000100051)

[14]. IBM. Red neuronal [Internet]. Ibm.com. 2021 [cited 2025 May 27]. Available from: <https://www.ibm.com/es-es/think/topics/neural-networks>

[15]. CS 230 - Recurrent Neural Networks Cheatsheet [Internet]. Stanford.edu. 2025 [cited 2025 May 8]. Available from: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>

[16]. Understanding LSTM Networks -- colah's blog [Internet]. Github.io. 2015 [cited 2025 May 8]. Available from: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[17]. IBM. Visión artificial [Internet]. Ibm.com. 2021 [cited 2025 May 8]. Available from: <https://www.ibm.com/es-es/topics/computer-vision>

[18]. Ultralytics. Pose [Internet]. Ultralytics.com. 2023 [cited 2025 May 7]. Available from: <https://docs.ultralytics.com/es/tasks/pose/>

[19]. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection [Internet]. 2016 May. Available from: <https://arxiv.org/pdf/1506.02640>

[20]. Cao Z, Hidalgo Martinez G, Simon T, Wei SE, Sheikh YA. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019;

## Capítulo 10. ANEXOS

### 10.1 Codigos

Códigos utilizados durante la elaboración del proyecto:

<https://github.com/VmSuarez25/TFG-Victor-Suarez/tree/main>



### 10.2 Declaración de uso de IA

- **Redacción académica:**
  - Herramienta: Chat GPT
  - Funciones: Refinar textos escritos personalmente para una mayor cohesión y una comprensión más clara.
  - Prompt utilizado: “Redacta este texto de manera más clara y formal sin cambiar el contenido”
- **Código:**
  - Herramienta: Chat GPT y Copilot
  - Funciones: Solucionar errores puntuales en líneas de código y sugerir códigos más compactos.
  - Prompt utilizado: “Me esta saliendo el siguiente error: (...) en este código en Python (.....) Explicame a que se refiere y como podría solucionarlo. Trata de agrupar todo lo que puedas en funciones sin alterar las características del código.”