

**Escuela Superior Politécnica del Litoral**

**Facultad de Ingeniería en Electricidad y Computación**

Sistema de ensamblaje autónomo de un eje de transmisión usando un brazo robótico y algoritmos de aprendizaje autónomo.

**Proyecto Integrador**

Previo la obtención del Título de:

**Ingeniera en Electrónica y Automatización**

Presentado por:

Doménica Isabel Córdova Bermeo

Guayaquil - Ecuador

Año: 2023

## **Dedicatoria**

---

El presente proyecto lo dedico a mi familia, sin su amor, apoyo y sacrificio nada de esto habría sido posible. A mis amigos, Paula, Dolores, Siza, Carlos y Andrés con quienes he compartido los mejores momentos, aligerado los tempestuosos y con los cuales mantengo la fe de que lo mejor está por venir.

## Agradecimientos

---

Mis más sinceros agradecimientos a mi tutor PhD. Edwin Valarezo por su guía y dedicación durante esta investigación. Un especial agradecimiento a mis profesores por su dedicación y pasión inquebrantable por la enseñanza. A mis colegas, Bolívar Núñez, Carlos Jiménez, Josué Cajamarca y Michael Quezada por brindarme su apoyo. A mis amigos durante la carrera Andrés, Javier, Francisco Henry, Ronald, Kleber, Stalin y a todos con quienes avanzamos hombro a hombro. Sobre todo, un inmenso agradecimiento a Enrique Estupiñan por su apoyo incondicional, que me ha impulsado a mantenerme enfocada en la meta.

## Declaración Expresa

---

“Los derechos de titularidad y explotación, me(nos) corresponde conforme al reglamento de propiedad intelectual de la institución; (nombre de los participantes) y doy(damos) mi(nuestro) consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Domé Córdoba B

---

Doménica Isabel Córdoba Bermeo

## **Evaluadores**

---

**PhD.Efrén Herrera Muentes**

Profesor de Materia

---

**PhD. Edwin Valarezo Añazco**

Tutor de proyecto

## Resumen

El proyecto se enfoca en abordar los desafíos de la industria de ensamblaje de ejes, donde las tareas manuales a menudo resultan en inexactitudes y baja eficiencia. Se identifican diversos errores, lo que motiva la implementación de un algoritmo de aprendizaje autónomo. Este algoritmo controla un brazo robótico para ensamblar piezas ubicadas de manera aleatoria en una superficie de trabajo. El sistema utiliza una cámara RGB-D en conjunto con un modelo basado en YOLO y transformación de coordenada para detectar y ubicar las piezas. El sistema de control hace uso de un algoritmo desarrollado con TD3 para dirigir las acciones del brazo robótico. Los resultados verifican la precisión del modelo de visión y la eficacia del algoritmo de control, validando así la capacidad del robot para aprender y adaptarse en función del progreso del entrenamiento. La recompensa alcanzada y la tendencia de mejora constante brindan respaldo a la viabilidad de este enfoque autónomo en la automatización de los procesos de ensamblaje de ejes, contribuyendo al incremento de la producción y a la reducción de errores de naturaleza humana.

**Palabras Clave:** Aprendizaje Profundo por Refuerzo, Visión, Ensamblaje, Robótica

## **Abstract**

The project focuses on solving the challenges of the shaft assembly industry, where manual tasks often result in inaccuracies and low efficiency. Several errors are identified, which motivates the implementation of an autonomous learning algorithm. This algorithm controls a robotic arm to assemble randomly placed parts on a workstation. The system uses an RGB-D camera in conjunction with a YOLO-based model and coordinate transformation to detect and locate the parts. The control system makes use of an algorithm developed with TD3 to direct the actions of the robotic arm. The results verify the accuracy of the vision model and the effectiveness of the control algorithm, thus validating the robot's ability to learn and adapt as training progresses. The reward achieved and the trend of constant improvement provide support for the feasibility of this autonomous approach in the automation of shaft assembly processes, contributing to the increase of production and the reduction of errors of human nature.

**Keywords:** Deep Reinforcement Learning, Vision, Assembly, Robotics,

## Índice general

Evaluadores .....	V
Resumen .....	VI
Abstract.....	VII
Abreviaturas.....	XI
Simbología.....	XII
Índice de figuras .....	XIII
Índice de tablas .....	XIII
Capítulo 1 .....	1
1.1    Introducción.....	2
1.2    Descripción del problema.....	3
1.3    Justificación del problema.....	5
1.4    Objetivos.....	6
1.4.1    Objetivo general .....	6
1.4.2    Objetivos específicos.....	6
1.5    Marco teórico.....	7
1.5.1    Ejes de transmisión.....	7
1.5.2    Fundamentos de la robótica.....	7
1.5.3    Inteligencia Artificial (IA).....	8
1.5.4    Aprendizaje Profundo por Refuerzo (DRL) .....	10
1.5.5    Visión Artificial.....	13
1.5.6    Simulación.....	13
1.6    Estado de Arte .....	14
1.6.1    Trabajos Previos .....	14



1.6.2	Contribución .....	16
2.	Capítulo 2 .....	17
2.1	Metodología.....	18
2.2	Simulación.....	19
2.2.1	Descripción.....	19
2.2.2	Modelado del entorno.....	20
2.3	Visión .....	22
2.3.1	Adquisición de imagen .....	22
2.3.2	Localización e identificación de las piezas.....	23
2.3.3	Creación y etiquetado de datos .....	24
2.3.4	Transformación de coordenadas .....	25
2.4	Entrenamiento.....	28
2.4.1	Desarrollo del algoritmo de entrenamiento .....	28
2.4.2	Algoritmo TD3 .....	29
2.4.3	Definición de la secuencia de ensamblaje .....	32
2.4.4	Definición de la función recompensa .....	33
3.	Capítulo 3 .....	35
3.1	Resultados y Análisis .....	36
3.1.1	Entorno de Trabajo .....	36
3.1.2	Sistema de Visión .....	36
3.1.3	Transformación de coordenadas .....	40
3.1.4	Resultados del DRL.....	43
3.1.5	Análisis de Costos .....	47

4.	Capítulo 4 .....	48
4.1	Conclusiones y recomendaciones .....	49
4.1.1	Conclusiones.....	49
4.1.2	Recomendaciones .....	50

## Abreviaturas

IA	Inteligencia Artificial
RGB-D	Red Green Blue y Depth
ML	Machine Learning
RL	Reinforcement Learning
DL	Deep Learning
DRL	Deep Reinforcement Learning
CNN	Convolutional Neural Network
YOLO	You Only Look Once
XML	Extensible Markup Language
MJCF	MuJoCo XML Format
URDF	Unified Robot Descripción Format
STL	Standard Triangle Language
SMVs	Support-Vector Machines
TD3	Twin Delayed Deep Deterministic policy gradient

## Simbología

Kg	Kilogramo
K	Kilo
m	Metro
cm	Centímetro
s	Segundo
N	Newton

## Índice de figuras

Figura 1 Eje de Transmisión .....	4
Figura 2 Interaccion entre el agente y el entorno.....	9
Figura 3 Ciclo de Reinforcement Learning .....	10
Figura 4 Diagrama de flujo de la metodología.....	18
Figura 5 Entorno de trabajo en MuJoCo .....	19
Figura 6 Descripción de las dimensiones del entorno de trabajo.....	20
Figura 7 Diseño tridimensional de las piezas .....	21
Figura 8 Imagen RGB e imagen de profundidad del entorno.....	22
Figura 9 Ejemplos de imágenes etiquetadas en el conjunto de datos.....	25
Figura 10 Modelo de proyección central.....	26
Figura 11 Algoritmo TD3.....	31
Figura 12 Matriz de Confusión del modelo de visión .....	37
Figura 13 Esfera de error de la transformación de coordenadas.....	42
Figura 14 Gráfica de Recompensa Promedio y Pasos Inicial.....	44
Figura 16 Gráfica de Recompensa Promedio y Pasos.....	45
Figura 16 Gráfica de Recompensa Promedio y Pasos Detalle.....	46
Figura 17 Secuencia de ensamblaje.....	47

## Índice de tablas

Tabla 1 Estadísticas de desempeño del modelo de visión.....	41
Tabla 2 Radio de la esfera de error de la transformación de coordenadas.....	42
Tabla 3 Error Relativo de la transformación de coordenadas.....	43
Tabla 4 Rubros de la implementación del proyecto.....	47

# **CAPÍTULO 1**

## 1.1 Introducción

El sector manufacturero es un pilar fundamental en la búsqueda de la prosperidad, esto se debe a que genera ingresos e innovación y cataliza la activación de otros sectores económicos. En el Ecuador, representa el 20.8 % del valor agregado bruto nacional y presentó una tasa de crecimiento promedio del 1.5 % anual durante el período 2007-2020 (Ochoa-Jiménez et al., 2022). Considerando este crecimiento, las industrias manufactureras buscan la manera de mejorar su productividad, por ende, optimizan sus sistemas de fabricación.

Dentro del sector manufacturero, se encuentra la industria de ensamblaje de ejes. En ella, la mayoría de los procesos son de alta precisión y repetitividad. Usualmente, estas tareas son realizadas manualmente, lo que conduce a errores humanos. Las estimaciones estadísticas muestran que entre el 20 % y el 90 % de los fallos de los sistemas tienen relación directa o indirecta con el agotamiento laboral (Di Pasquale et al., 2015; Griffith & Mahadevan, 2011).

Dhillon (2009) señala a los errores de montaje como uno de los errores humanos más comunes en un contexto de fabricación. Estos errores están asociados a las capacidades del trabajador, como conocimientos y habilidades; la fatiga psicofísica; los parámetros de la tarea, como la carga de trabajo y la repetitividad; y el entorno de trabajo (Elmaraghy et al., 2008; Fruggiero et al., 2017). Algunos de los principales errores son tareas realizadas de forma no secuencial, uso de la pieza/objeto equivocado, y la aplicación de la fuerza incorrecta. Dichos errores pueden causar accidentes, defectos en la calidad y retrasos en la producción (Michalos et al., 2013). Los accidentes son fácilmente rastreables; sin embargo, los fallos menores pueden aumentar drásticamente los tiempos de producción, costes, tasas de reprocesamiento y desechos (Di Pasquale et al., 2015; Elmaraghy et al., 2008).

Para la resolución de esta problemática, se hace hincapié en la automatización y la fiabilidad de la maquinaria y los equipos, una opción más rentable en comparación con la mejora del rendimiento de los trabajadores (Fruggiero et al., 2016). Es por ello por lo que, el presente proyecto busca el desarrollo de un sistema de ensamblaje autónomo de un eje de transmisión usando un brazo robótico y algoritmos de aprendizaje autónomo.

## **1.2 Descripción del problema**

Dentro de la industria de ensamblaje de ejes, se realiza tareas que demandan gran precisión y son de alta repetitividad, tales como, la colocación de rodamientos, la fijación de tornillos y la aplicación de lubricantes. A menudo estas tareas se realizan manualmente. Ahora bien, debido a las condiciones y requerimientos de estos trabajos, se cometen muchos errores humanos durante el ensamblaje.

Las categorías de errores posibles según la “Técnica para la predicción de la tasa de error humano” son: errores de omisión (omisión de un paso de la tarea) y errores de acción. Estos últimos pueden ser de cuatro tipos:

- Error de selección (selección del control erróneo, elección de procedimientos de forma incorrecta).
- Error en la secuencia (acción llevada a cabo en orden incorrecto)
- Error temporal (acción llevada a cabo muy pronto / muy tarde)
- Error cualitativo (acción realizada mucho / poco)(Salas-Arias et al., 2018,p. 28).

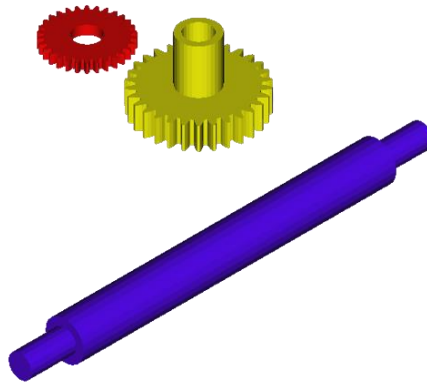
Con el fin de evitar estos errores, en este proyecto se creó un algoritmo basado en aprendizaje autónomo que controla un brazo robótico para ensamblar automáticamente piezas de un eje de transmisión, ubicadas de forma aleatoria en la mesa de trabajo. La



Figura 1, muestra la composición del eje de transmisión, dos engranes y un eje. Las piezas se ubican aleatoriamente en una mesa de trabajo, por lo que se requiere de un sistema de visión y control.

### **Figura 1**

*Eje de Transmisión.*



*Nota:* Eje de transmisión detallado. (1) Primer engrane en amarillo, (2) Segundo engrane en rojo, (3) Eje en azul

El sistema de visión incluye una cámara Red Green, Blue y Depth, (por sus siglas en inglés, RGB-D), para visualizar cada pieza y localizarla en el espacio de trabajo, con ayuda de un algoritmo que la detecte. El sistema de control debe priorizar la manipulación de cada pieza, de acuerdo con el orden de armado del eje de transmisión. Estos sistemas autónomos son capaces de operar con menor intervención y supervisión humana, optimizando el tiempo de productividad y reduciendo drásticamente el error humano.

### **1.3 Justificación del problema**

En el panorama actual, la automatización de procesos e incorporación de nuevas tecnologías son una necesidad. Según las encuestas del Banco Mundial, un análisis de dos indicadores de desempeño muestra que las capacidades industriales juegan un papel crucial a la hora de suavizar el impacto frente a una crisis (United Nations Industrial Development Organization, 2021). Las empresas manufactureras de los países con mayores capacidades industriales han sido más robustas durante crisis como la pandemia del COVID 19.

Con la modernización de las capacidades industriales, podemos mejorar significativamente la ejecución de muchos procesos, como en este caso es el ensamblaje de un eje de transmisión usando un brazo robótico y algoritmos de aprendizaje autónomo. La integración de este sistema a las industrias permite reducir las fallas humanas, los riesgos laborales durante la operación, mejorará la calidad y consistencia de las piezas terminadas y por ende mejorar los tiempos de producción y confiabilidad, lo que propiciará una mejor consolidación de las industrias a futuro.

## **1.4 Objetivos**

### ***1.4.1 Objetivo general***

El objetivo de este proyecto es utilizar técnicas de aprendizaje autónomo para automatizar los procesos de ensamblaje de ejes de transmisión, incrementado el volumen de producción y disminuyendo fallas e interrupciones debidas a errores humanos.

### ***1.4.2 Objetivos específicos***

- Implementar un sistema de visión capaz de detectar y localizar las partes del eje de transmisión.
- Diseñar una inteligencia artificial basada en algoritmos de aprendizaje autónomo para controlar el brazo robótico.
- Entrenar la inteligencia artificial para ensamblar las piezas en el orden lógico establecido por un usuario.

## **1.5 Marco teórico**

En esta sección se presenta los fundamentos de la robótica, los algoritmos de aprendizaje autónomo y la visión artificial, con el objetivo de comprender su aplicación en el presente proyecto.

### ***1.5.1 Ejes de transmisión***

Los ejes de transmisión son componentes que transmiten la potencia del motor al diferencial. Son usualmente tubulares y están hechos de acero o aluminio para soportar las altas tensiones y torsiones durante la operación (Gilles, 2015). Estos poseen diferentes componentes dependiendo de su diseño, los más comunes suelen ser, engranajes, cojinetes, soportes, acoples y lubricantes.

El ensamblaje preciso de los ejes de transmisión es esencial para garantizar un funcionamiento suave y eficiente del sistema de transmisión. Cualquier desalineación o defecto en el ensamblaje puede ocasionar vibraciones, ruidos, desgaste prematuro e incluso fallas en el sistema. Por lo tanto, es crucial emplear métodos y tecnologías adecuadas para lograr un ensamblaje preciso y confiable de los ejes de transmisión.

### ***1.5.2 Fundamentos de la robótica***

Un robot industrial se define como un manipulador multipropósito, reprogramable, controlado automáticamente, programable en tres o más ejes; que puede estar fijo o móvil para el uso en aplicaciones de automatización industrial (Groover, 2008). Algunos ejemplos de estos robots son los Universal Robots UR5, ABB YuMi, Techman TM5 y Kuka LBR iiwa.

El control de los robots es una parte crucial de la robótica y puede ser complejo, dependiendo del nivel de sofisticación y funcionalidad del robot. Los métodos más comunes de control de robots son control manual o teleoperación; control por sensores o retroalimentación; control por inteligencia artificial (IA); control predictivo y control adaptativo. (Russel & Norvig, 2016; Siciliano & Khatib, 2016).

### ***1.5.3 Inteligencia Artificial (IA)***

La Inteligencia Artificial es una rama de la informática, que se centra en la creación de sistemas que incluyen tareas las cuales normalmente requieren inteligencia humana. Estas tareas incluyen reconocer patrones, tomar decisiones y resolver problemas (Russel & Norvig, 2016).

Un subcampo que evolucionó de la IA es el Aprendizaje Automático (ML), el cual involucra algoritmos de autoaprendizaje que adquieren conocimiento a partir de datos para crear predicciones (Raschka & Mirjalili, 2020).

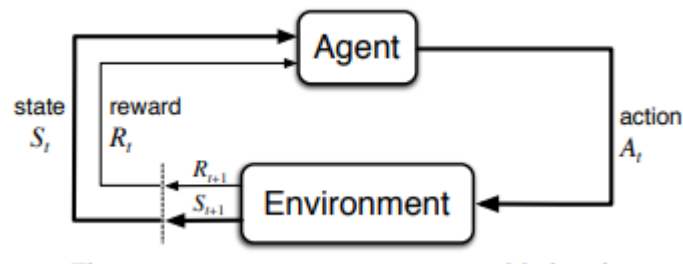
En el ML podemos identificar varios actores, de manera general estos son agente y entorno. El aprendiz y tomador de decisiones se llama el agente. El ambiente con el que interactúa, que comprende todo lo que está fuera del agente, se llama entorno. Estos interactúan continuamente, el agente seleccionando acciones y el ambiente respondiendo a estas acciones y presentando nuevas situaciones al agente. El ambiente también da lugar a recompensas, valores numéricos especiales que el agente busca maximizar con el tiempo a través de su elección de acciones (Sutton & Barto, 2018).

Dependiendo la manera en la que interactúan estas partes podemos establecer varios tipos de aprendizaje Hay tres tipos de fundamentales de aprendizaje, el

aprendizaje no supervisado, el agente aprende patrones de los datos, aunque no se le proporcione una retroalimentación explícita. El aprendizaje por refuerzo (RL), el agente aprende a partir de experiencias interactuando en un ambiente para resolver un problema específico. En el aprendizaje supervisado, el agente observa algunos pares de entrada-salida de ejemplo y aprende una función que asigna la entrada a la salida. (Russel & Norvig, 2016).

**Figura 2**

*Interacción entre el agente y el entorno*



*Nota.* Adaptado de La interacción agente-entorno en un proceso de decisión de Markov, (p. 70), de Sutton & Barto, 2018.

La Figura 2 corresponde a la interacción entre el agente y el entorno fundamental en el ciclo de aprendizaje basado en RL.

Dentro de ML encontramos el Aprendizaje Profundo (DL), que es un subcampo caracterizado por el uso de redes neuronales artificiales, a menudo llamadas redes neuronales profundas. Estas redes neuronales son capaces de aprender representaciones de los datos que son demasiado complejas para que las capturen los modelos más simples (Goodfellow et al., 2017).

Una red neuronal es una serie de nodos, o “neuronas” organizados en capas, donde se incluye una capa de entrada, una o varias capas de procesamiento intermedias y una

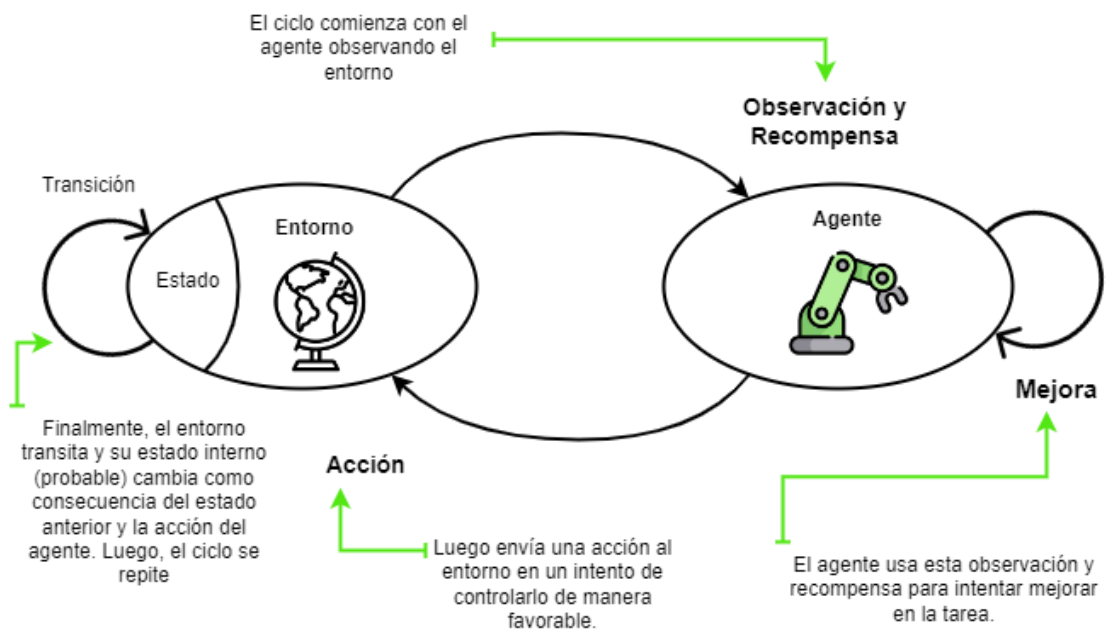
capa de salida (Russel & Norvig, 2016). Cada neurona en una capa está conectada a todas las neuronas en la capa siguiente, y estas conexiones tienen pesos asociados que la red ajusta para aprender a partir de los datos (Goodfellow et al., 2017).

#### 1.5.4 Aprendizaje Profundo por Refuerzo (DRL)

Al combinar el DL con el RL obtenemos el Aprendizaje Profundo por Refuerzo, el cual permite aprender a través del entrenamiento en distintos entornos y medir los resultados generados por las acciones que realiza para alcanzar sus objetivos (Goodfellow et al., 2017). En el campo de la robótica es un algoritmo de aprendizaje autónomo usualmente utilizado, y se puede representar como en la Figura 3.

**Figura 3**

*Ciclo de RL*



*Nota.* Adaptado de Ciclo del Aprendizaje por Refuerzo, (p. 33), de Morales, 2020.

Las interacciones entre el agente y el entorno continúan durante varios ciclos, cada ciclo se denomina pasos. En cada paso el agente observa el entorno, actúa y recibe una nueva observación y recompensa. El conjunto del estado, la acción, la recompensa y el nuevo estado se denomina experiencia. Cada experiencia tiene una oportunidad para aprender y mejorar el desempeño. (Morales, 2020)

Más allá del entorno y el agente se pueden identificar varios elementos que interactúan en este tipo de aprendizaje

*Actor (Política):* El actor es una función, generalmente representada por una red neuronal, que genera una correspondencia entre los estados percibidos del entorno y las acciones a realizar en esos estados. La política puede ser estocástica (elige acciones basadas en una distribución de probabilidad) o determinista (elige siempre la misma acción para un estado dado).

*Crítico (Función de Valor Q):* La función de valor Q, que suele estar representada por una red neuronal, toma un estado y una acción, y proporciona un valor numérico esperado basado en tomar una acción en un estado específico y seguir la política del actor en el futuro.

*Espacio de Estados:* Representa todas las situaciones posibles en las que el agente puede encontrarse. Un estado es una descripción completa de una situación.

*Recompensas:* Posterior a que el agente toma una acción en un estado este recibe una recompensa, esta define cuáles son los sucesos buenos y malos para el para el agente.



*Búfer de Repetición de Experiencia:* Es la memoria en donde se almacenan las experiencias (estado, acción, recompensa, estado siguiente). Mediante este búfer podemos acceder al muestreo aleatorio de experiencias pasadas para mejorar la estabilidad del aprendizaje.

*Aprendizaje:* El agente mejora sus decisiones (política) y estimaciones (función Q) utilizando gradientes derivados del error entre la recompensa obtenida y la recompensa estimada. Al muestrear experiencias del búfer y calcular el error de predicción de recompensa, se actualizan las redes neuronales del actor y del crítico.

*Tasa de Descuento ( $\gamma$ ):* Es un factor que disminuye el valor de las recompensas futuras. Esto hace que las recompensas inmediatas sean más valiosas que las recompensas a largo plazo.

*Exploración vs. Explotación:* Durante el aprendizaje, el agente debe decidir si sigue su política actual (explotación) o si prueba acciones aleatorias (exploración) para descubrir nuevas estrategias. Típicamente, esto se gestiona mediante algoritmos de selección de acciones como epsilon-greedy.

En resumen, en DRL, el actor decide qué acción tomar basándose en su política actual, y el crítico estima los valores futuros de esas acciones. Con las recompensas recibidas del entorno y las estimaciones del crítico, el actor ajusta su política para maximizar la recompensa acumulada en el futuro.

### ***1.5.5 Visión Artificial***

La visión artificial es un subcampo de la inteligencia artificial que se centra en proporcionar a las máquinas la capacidad de "percibir" e interpretar imágenes y videos.

Debido a las bondades y precisión alcanzadas con algoritmos de DL, en particular las redes neuronales convolucionales (CNN), se ha popularizado el uso de CNN en problemas de visión robótica y de máquina. Lo cual permite que sea utilizada para realizar tareas como detección, segmentación de objetos, reconocimiento fácil entre otros.

En el DRL la visión artificial nos permite captar el entorno en el que estamos interactuando. En el entrenamiento de un brazo robótico, el agente tiene acceso a observaciones de su entorno a partir de imágenes de una cámara, que luego de un procesamiento le permite conocer su estado. (Morales, 2020)

### ***1.5.6 Simulación***

Los entornos simulados son una herramienta muy útil para el entrenamiento de un algoritmo de inteligencia artificial, ya que ofrece ventajas significativas, por ejemplo, proporciona ambientes controlados, iteraciones más rápidas y una mayor escalabilidad y eficiencia.

MuJoCo es un motor de física versátil para investigación en robótica, biomecánica, gráficos y aprendizaje automático. Permite la simulación precisa y rápida de estructuras articuladas que interactúan con el entorno. (MuJoCo, 2022).

## **1.6 Estado de Arte**

### ***1.6.1 Trabajos Previos***

En esta sección, se realizará una revisión y contraste de trabajos anteriores relevantes, con el fin de obtener una perspectiva actualizada de los avances existentes y así establecer una base sólida del contexto de la investigación.

#### ***1.6.1.1 Robotic assembly strategy via reinforcement learning based on force and visual Information***

En este estudio Ahn et al. (2023), proponen un generador de trayectorias para robots que realizan tareas de ensamblaje, utilizando tanto información visual como de fuerza. El generador de trayectorias basado en imágenes, correspondiente a la red del actor de SAC, determina la dirección de movimiento del robot en función de la información visual obtenida de una cámara RGB en el efector final. Mediante un método de adaptación propuesto, este generador de trayectorias logró reducir el número de pasos de aprendizaje necesarios. En las pruebas realizadas, se obtuvo una alta tasa de éxito, incluso en presencia de errores iniciales significativos. Estos resultados demuestran la robustez de la estrategia propuesta para enfrentar errores en el ensamblaje y manejar formas complejas de piezas.

A pesar de estos avances, aún existen desafíos para implementar esta estrategia en entornos industriales reales, debido a la influencia del entorno en la información visual. Se propone mejorar el rendimiento mediante el desarrollo de métodos de segmentación de imágenes más robustos.

#### ***1.6.1.2 Learning 6-DoF grasping with dual-agent deep reinforcement learning***

En este trabajo Hou & Li (2023), presentan un método de aprendizaje llamado 6DGL, que se utiliza para realizar agarres en seis grados de libertad (6-DoF). El enfoque

se basa en una arquitectura cooperativa de aprendizaje por refuerzo multi agente (MARL), en la cual se utilizan agentes de posición y orientación. El método utiliza prestaciones planas y esféricas para representar el espacio de acción de los agentes y emplea QMIX-PSA para promover la colaboración entre ellos y lograr una comprensión de los agarres en 6-DoF. QMIX-PSA consta de tres componentes: PA-Net, SA-Net y una red de mezcla llamada QMIX. Estos componentes permiten predecir las prestaciones planas y esféricas, así como estimar el valor de acción conjunta de los agentes. Además, se propone una recompensa mejorada para mejorar la calidad de los agarres en 6-DoF.

### ***1.6.1.3 Robotic Arm Control and Task Training through Deep Reinforcement Learning***

En este trabajo Franceschetti et al., (2020) realizan una comparación entre los algoritmos DQN, NAF, TRPO, VPG y DDPG en tareas clásicas de destreza del efector final y Pick & Place utilizando un brazo robótico colaborativo de 10 grados de libertad. Se observa que es posible obtener buenos resultados en un número razonable de episodios, y que los tiempos de entrenamiento pueden ser mejorados mediante la asignación de más recursos computacionales. El algoritmo DQN-NAF muestra un rendimiento destacado en la tarea de alcance, logrando una política subóptima. Por otro lado, TRPO se destaca por su versatilidad al adaptarse a diferentes recompensas y parámetros invariables. VPG tiende a tener un aprendizaje más lento, mientras que DDPG resulta más inestable y difícil de ajustar debido a su alta sensibilidad a la escala de recompensas. Además, se observa que la arquitectura de la red neuronal (ancho/profundidad) no es un factor determinante en el proceso de aprendizaje, y su elección depende del algoritmo específico utilizado.

### ***1.6.2 Contribución***

A diferencia de investigaciones anteriores, este proyecto propone implementar un algoritmo que ensamble automáticamente un eje de transmisión, considerando que las piezas son ubicadas de forma aleatoria en la mesa de trabajo. Para ello, se realiza reconocimiento de las partes del eje mediante imágenes capturadas por una cámara RGB-D y utilizando You Only Look Once (por sus siglas en inglés, YOLO). El sistema se compone de un agente representado por un robot UR5 de Universal Robots, y el entorno será simulado mediante MuJoCo. Durante el entrenamiento, se emplean tres componentes de la función de recompensa, para asegurar un avance adecuado, al combinar todas estas partes, se espera lograr un algoritmo con un comportamiento eficiente y robusto.

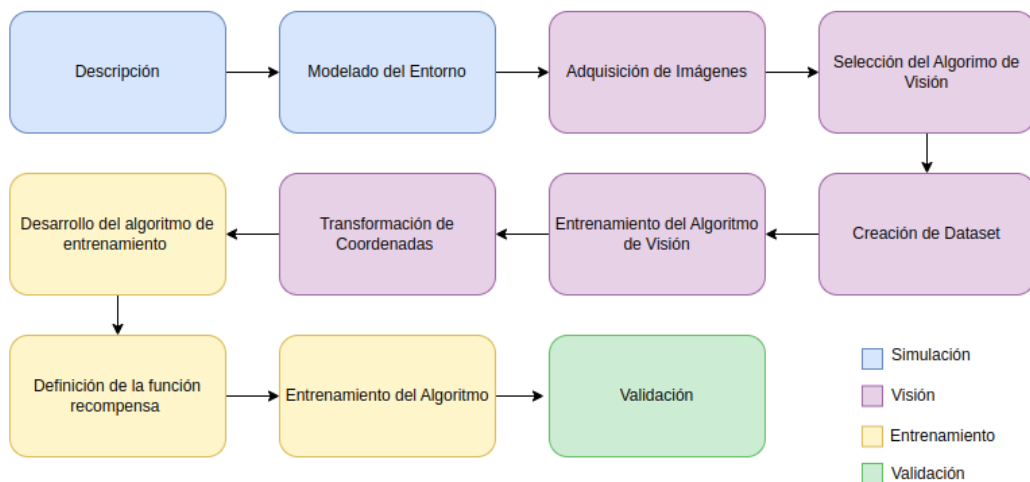
## **2. CAPÍTULO 2**

## 2.1 Metodología

En este capítulo se describe la metodología utilizada para el desarrollo de un sistema de ensamblaje autónomo de un eje de transmisión usando un brazo robótico y algoritmos de aprendizaje autónomo. En este proceso, se pueden identificar varias etapas con requerimientos específicos. La primera etapa, denominada Simulación, muestra la descripción y modelado del entorno. La etapa, denominada Visión, se enfoca en la identificación del tipo de pieza y su localización. La etapa siguiente, denominada Entrenamiento, se refiere al control, en la cual se busca desarrollar un algoritmo de aprendizaje para trazar una trayectoria hacia un objeto específico y posterior a esto la reubicación de las piezas. En la Figura 4 se presentan los pasos a seguir organizados por etapas.

**Figura 4**

*Diagrama de flujo de la metodología*



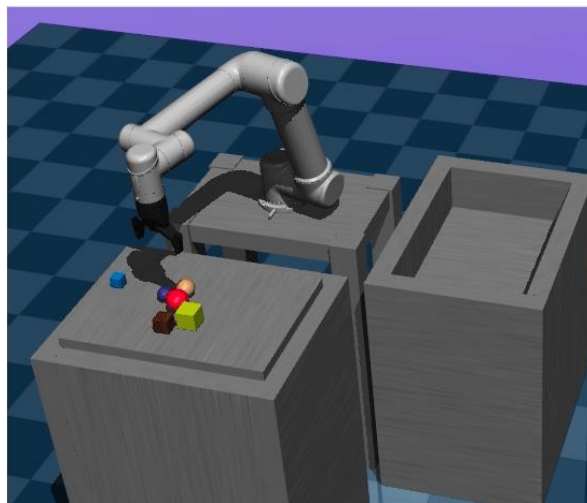
## 2.2 Simulación

### 2.2.1 Descripción

Para la simulación del ambiente de este proyecto se utiliza MuJoCo de la abreviatura Multi-Joint dynamics with Contact. Este es un motor de física y simulación altamente eficiente para sistemas de múltiples articulaciones y contactos; presenta alta fidelidad para la simulación de sistemas robóticos complejos y entornos interactivos. MuJoCo se utiliza, ya que facilita el desarrollo y prueba de algoritmos de control y aprendizaje automático en sistemas robóticos. En este caso como punto de partida se usa el ambiente modelado por (Daniel, 2022).

#### Figura 5

*Entorno de trabajo en MuJoCo*



*Nota:* Entorno de trabajo y sus principales elementos tomado de (Daniel, 2022)

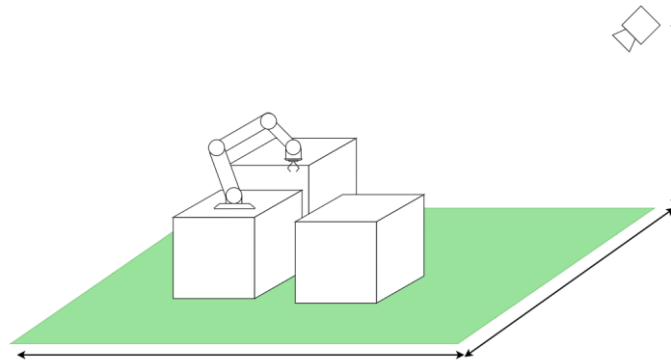
Como se muestra en la Figura 5, este ambiente consta de un brazo robótico Universal Robots UR5 equipado con un gripper de tipo pinza de dos dedos, una mesa de trabajo frontal (mesa de trabajo) y la mesa de trabajo lateral (mesa de ensamblaje) y una cámara RGB-D.



El simulador MuJoCo, basa las unidades métricas en el Sistema Internacional (SI), lo que significa que las cantidades físicas como masa se miden en kilogramos (kg), longitud en metros (m), tiempo en segundos (s) y fuerza en newtons (N). Las dimensiones del entorno de trabajo son de 2.5 metros de largo, 2.5 metros de ancho y 2 metros de alto como se muestra en le Figura 6

### Figura 6

*Descripción de las dimensiones del entorno de trabajo*



*Nota.* Descripción de la ubicación y dimensiones del entorno de trabajo

#### 2.2.2 Modelado del entorno

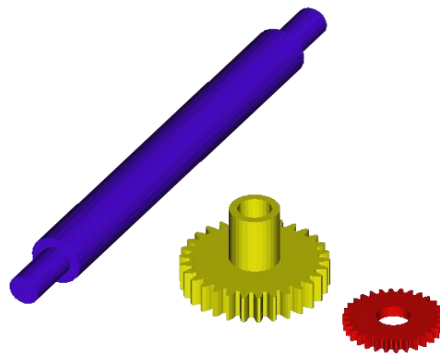
Las estructuras de datos de MuJoCo son de bajo nivel con el propósito de maximizar su legibilidad y editabilidad. Por lo tanto, como se menciona en (MuJoCo, 2022), se puede cargar archivos de modelo XML (por sus siglas en inglés, Extensible Markup Language), en su formato nativo MJCF (MuJoCo XML Format), así como en formato URDF (Unified Robot Descripción Format), los cuales permiten representar la geometría, la dinámica y la cinemática de un robot.

El entorno de trabajo es creado a partir de elementos XML y sus atributos MJCF, el brazo robótico es cargado en el entorno a partir de un archivo URDF, la cámara es un objeto de tipo “body/cámara”, la cual nos permite acceder a la visualización del entorno. Las mesas son objetos de tipo “body/geom”, la cual permite crear geometrías dentro del entorno de trabajo. Para finalizar los elementos personalizables como en este caso son las piezas son objetos del tipo “body/freejoint”.

Para incorporar las piezas al entorno, el primer paso es crear un archivo 3D de formato STL (Standard Triangle Language), con el modelado de cada pieza individualmente, con las dimensiones apropiadas como se muestra en el Figura 7.

### **Figura 7**

*Diseño tridimensional de las piezas*



Luego se crea un archivo XML donde se detalla la física de este grupo de piezas, como su densidad, su material o su fricción. Por último, se agrega este archivo XML en el archivo XML principal, donde se detallan todos los elementos del modelo.

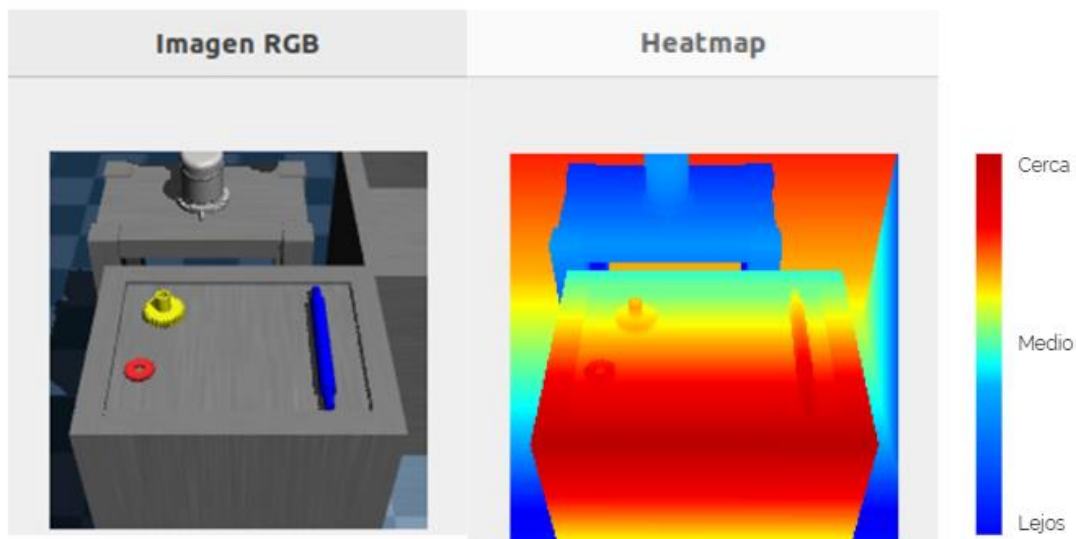
## 2.3 Visión

### 2.3.1 Adquisición de imagen

El sistema de visión adquiere información del área de trabajo mediante el objeto cámara, el cual emula una cámara RGB-D y proporciona información de color (RGB) y datos de profundidad (D). En la Figura 8 se observa la visualización de la cámara para el color y la profundidad.

#### Figura 8

*Imagen RGB e imagen de profundidad del entorno*



*Nota.* En esta figura se presenta las imágenes de componentes de color que devuelve la cámara, así como la componente profundidad, esta última como un mapa de calor en la que dependiendo su calor se indica la profundidad de los objetos.

### ***2.3.2 Localización e identificación de las piezas.***

Para la tarea de localización e identificación de las piezas existen muchos algoritmos que se pueden utilizar, entre esos están:

Redes Neuronales Convolucionales (CNNs). Las CNNs son efectivas para el procesamiento de imágenes lo que las hace excelentes para la localización e identificación de objetos. Sin embargo, requieren una gran cantidad de datos etiquetados para su entrenamiento y considerables recursos computacionales.

YOLO Este algoritmo es especialmente bueno en tareas de detección y localización de objetos en una imagen, una particularidad de este algoritmo es que su rendimiento se mantiene en detección de objetos en tiempo real. Si bien requieren gran cantidad de datos etiquetados y recursos computacionales para el entrenamiento. Roboflow es una herramienta disponible online diseñada para el etiquetado de datos y el aumento de datos. Además, ofrece capacidades de entrenamiento en la nube, eliminando la necesidad de utilizar recursos computacionales locales.

Algoritmos de Clasificación como Máquinas de Vectores de Soporte (SVMs). Este tipo de algoritmos son útiles siempre y cuando podamos identificar características fundamentales extraídas de manera manual, su eficacia depende de las características seleccionadas, la precisión promedio de reconocimiento en tiempo real es superada por YOLO debido a su precisión y detección de múltiples objetos en una sola imagen.

### **2.3.3 Creación y etiquetado de datos**

La etapa inicial en la identificación y localización de piezas implica la construcción de un conjunto de datos robusto que represente el espacio diseñado. Estos datos sirven como base, para llevar a cabo el proceso de entrenamiento. Para evitar la detención errónea, el conjunto de datos debe ser diverso e incorporar todas las variantes del objeto en términos de iluminación, ángulo y tamaño. Además, es fundamental que las imágenes proporcionadas sean de una calidad similar a la que recibe el modelo una vez entrenado. Por último, para optimizar el entrenamiento, es importante disponer de una cantidad adecuada de datos.

En este caso la identificación está basada en el tipo de pieza, donde se distinguieron dos clases: "gear", que representa los engranajes, y "axi", que representa al eje. En la creación de los datos, se incluyen imágenes del área de trabajo para simular las condiciones a las que el modelo se enfrentará. Por otro lado, el etiquetado de los datos se realizó utilizando herramientas de Roboflow, lo que permitió generar cuadros delimitadores más precisos y agilizar el proceso de etiquetado. El total de imágenes del conjunto, incluidas las generadas por aumentación, fue de 1053 imágenes, de las cuales un 82% se usa para el entrenamiento, un 12% para la validación y un 6% para el testeo. En la Figura 9 podemos observar un ejemplo de la data que se utiliza

## Figura 9

*Ejemplos de imágenes etiquetadas en el conjunto de datos.*



*Nota.* Imágenes etiquetadas y separadas según su papel en el entrenamiento.

### 2.3.4 Transformación de coordenadas

Luego de procesar la imagen mediante YOLO, las predicciones obtenidas respecto a la posición de los objetos se presentan en función de los píxeles de la imagen. Por lo tanto, es necesario convertir estas coordenadas de imagen a coordenadas referenciadas al sistema de la base del robot.

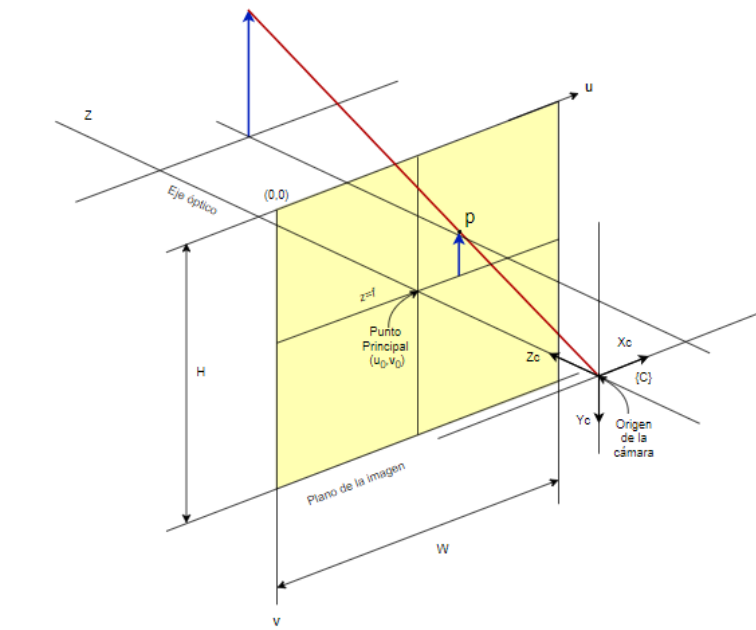
#### 2.3.4.1 Transformación de perspectiva

Una imagen es una representación bidimensional de una escena tridimensional. Como menciona Corke (2011), en una cámara, se utiliza un modelo de perspectiva central en el cual los rayos se enfocan en el origen del sistema de referencia de la cámara  $\{C\}$  y

se proyecta una imagen no invertida sobre el plano de imagen ubicado en  $z=f$ . Esto se puede apreciar en la Figura 10.

**Figura 10**

*Modelo de proyección central.*



*Nota.* En esta figura se detalla la imagen bidimensional obtenida a partir de la proyección central de un objeto tridimensional en un plano.

Para poder pasar de una representación a otra es necesario aplicar una matriz de transformación la cual está estructurada como se muestra en la Ecuación 1.

$$T = \begin{bmatrix} \frac{f}{\rho_w} & 0 & u_o \\ 0 & \frac{f}{\rho_h} & v_o \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 373.2 & 0 & 100 \\ 0 & 373.2 & 100 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Donde:

$f$ : Punto focal de la cámara

$\rho_w$ : Ancho de los pixeles

$\rho_h$ : Alto de los pixeles

$u_o, v_o$ : Punto central de la imagen

Con las características de la cámara se determina la matriz de transformación descrita en la Ecuación 2

Conociendo la matriz de transformación, el cambio de perspectiva de una coordenada expresada en píxeles a una coordenada tridimensional en el sistema de referencia de la cámara se puede expresar mediante la Ecuación 2.

$$P = T^{-1} p \quad (2)$$

Donde:

$P$ : Coordenadas del punto en el espacio

$p$ : Cordenadas del punto en el imagen

$T^{-1}$ : Inversa de la matriz de trasformación

#### 2.3.4.2 Matriz de trasformación

Para realizar la transformación de coordenadas desde el sistema de referencia de la cámara al sistema de referencia de la base del robot, se emplea una matriz de transformación. Esta matriz permite establecer una relación entre ambos sistemas considerando la posición y rotación del sistema de referencia de la cámara con respecto al de la base, como se muestra en la Ecuación 3.

$$T = \begin{bmatrix} R_{3x3} & P_{1x3} \\ f_{1x3} & w_{1x1} \end{bmatrix} = \begin{bmatrix} Rotación & Traslación \\ Perspectiva & Escalado \end{bmatrix} \quad (3)$$



Con la matriz de transformación el cambio entre un sistema a otro se puede expresar como se detalla en la Ecuación 4.

$${}^0p = {}^0T_{cam} {}^{cam}p \quad (4)$$

Donde:

${}^0p$ : punto respecto al sistema de referencia del robot

${}^0T_{cam}$  ; Matriz de Transformación

${}^{cam}p$ : punto respecto al sistema de referencia de la cámara

Las posiciones y rotaciones de estos sistemas en este caso se obtienen a partir de la experimentación, se usan las posiciones de puntos conocidos en ambos ejes de referencia, obteniendo así la matriz de transformación de la Ecuación 5.

$${}^0T_{cam} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.707 & 0.707 & -1.7 \\ 0 & -0.707 & -0.707 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

## 2.4 Entrenamiento

### 2.4.1 Desarrollo del algoritmo de entrenamiento

Dentro del DRL, existen varios algoritmos que permiten el entrenamiento de modelos. Cada uno de estos algoritmos presenta diferentes enfoques para abordar los problemas de mejor manera.

En el algoritmo Deep Deterministic Policy Gradient (por sus siglas en inglés, DDPG), combina DL y aprendizaje por refuerzo determinista y se caracteriza por

aprender políticas competitivas en espacios de acción continuos usando observaciones de baja dimensión(Lillicrap et al., 2015).

El algoritmo Twin Delayed Deep Deterministic policy gradient (por sus siglas en inglés, TD3), se aplica en aprendizaje por refuerzo basado en valores. Este algoritmo se fundamenta sobre el Double Q-learning, tomando el valor mínimo entre un par de críticos para limitar la sobreestimación de valores y entrenar un actor, donde los críticos y el actor están modelados usando redes neuronales convolucionales (Fujimoto et al., 2018) .

El sistema de control de este proyecto utiliza TD3, debido a sus características distintivas. En primer lugar, permite la reducción del ruido mediante una técnica conocida como "suavizado regularizado". Esta técnica agrega ruido a la acción seleccionada por la política, con el fin de fomentar la exploración.

Además, el algoritmo TD3 incorpora una estrategia de retraso en la actualización del actor. En lugar de actualizar el actor con la misma frecuencia que el crítico, se actualiza con menor frecuencia. Esto ayuda a mitigar el error generado al actualizar la política en cada paso de tiempo.

Por último, TD3 utiliza la función de valor doble. En este enfoque, se emplean dos redes criticas independientes y se toma el valor mínimo de ambas para estimar el valor de una acción. Esto permite una limitación efectiva y ayuda a evitar la sobreestimación de críticos, así como actores subóptimos.

#### ***2.4.2 Algoritmo TD3***

El Algoritmo TD3, puede ser articulado de manera estructurada descomponiéndolo en secciones o etapas diferenciadas. Entre sus etapas están:

*Inicialización:* Inicializamos dos críticos (las funciones de valor Q) y un actor (la política), así como sus correspondientes objetivos.

*Recopilación de experiencias:* El actor interactúa con el entorno y genera trayectorias (secuencias de estados, acciones y recompensas). Estas trayectorias se almacenan en el búfer de repetición para su posterior uso.

*Actualización del crítico:* Se selecciona una mini-muestra aleatoria de trayectorias del búfer de repetición. Para cada muestra, se utiliza el actor de destino para generar una acción para el siguiente estado en la trayectoria, y esta acción se utiliza para calcular el valor Q de destino utilizando los críticos de destino. Luego, actualizamos los críticos para minimizar la diferencia entre su valor Q predicho y el valor Q de destino.

*Actualización del actor:* Después de un cierto número de actualizaciones de los críticos, se actualiza el actor para maximizar el valor Q promedio de los críticos.

*Actualización de los objetivos:* Después de actualizar los críticos y el actor, se actualizan lentamente los actores y críticos de destino para reflejar los valores de los actores y críticos actualizados. Esto se hace a través de un proceso llamado "actualización suave", donde en lugar de copiar directamente los pesos, se mueven lentamente hacia los valores de los actores y críticos actualizados.

*Repetir:* Se repiten los pasos posteriores a la inicialización muchas veces hasta que la política (el actor) se desempeña de manera satisfactoria. La idea clave del TD3 es utilizar dos críticos en lugar de uno y retrasar la actualización del actor. De esta manera, se pueden evitar algunos de los problemas comunes en los métodos de actor-crítico, como la sobreestimación de las recompensas futuras y la inestabilidad del aprendizaje.

En términos de pseudo código este algoritmo se presenta como se muestra en la Figura 11.

## Figura 11

### Algoritmo TD3

#### Algoritmo TD3

Inicializamos la redes de criticos  $Q_{\theta_1}, Q_{\theta_2}$  y a red del actor  $\pi_\phi$

con parámetros aleatorios,  $\theta_1, \theta_2$  y  $\phi$

Inicializamos las redes objetivo,  $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$  y  $\phi' \leftarrow \phi$

Inicializamos el buffer de reproducción  $\beta$

**for**  $t = 1$  **to**  $T$  **do**

Seleccionamos una acción con ruido de exploración  $a \sim \pi_\phi(s) + \epsilon$

$\epsilon \sim \mathcal{N}(0, \sigma)$  y observamos la recompensa  $r$  y el nuevo estado  $s'$

Almacenamos la tupla de transición  $(s, a, r, s')$  en el buffer  $\beta$

Probamos un mini – muestra de  $N$  transiciones  $(s, a, r, s')$  desde  $\beta$

$\bar{a} \leftarrow \pi_{\phi'}(s') + \epsilon, \epsilon \sim \text{clip}(\mathcal{N}(0, \bar{\sigma}), -c, c)$

$y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \bar{a})$

Actualizamos los críticos  $\theta_i \leftarrow \text{argmin}_\theta N^{-1} \sum (y - Q_{\theta_1}(s, a))^2$

**if**  $t \bmod d$  **then**

Actualizamos  $\phi$  el gradiente de politica deterministica:

$$\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$$

Actualizamos las redes objetivo:

$$\theta'_i \leftarrow \tau \theta_i + (1 - \tau) \theta'_i$$

$$\phi' \leftarrow \tau \phi + (1 - \tau) \phi'$$

**end if**

**end for**

*Nota.* Adaptado de Algorithm 1 TD3, de Fujimoto et al., 2018

### **2.4.3 Definición de la secuencia de ensamblaje**

Par el ensamblaje del eje de transmisión se define la siguiente secuencia de acciones divididas por etapas:

#### Etapa 1

1. Acercarse al engranaje primario.
2. Levantar el engranaje primario.
3. Situar el engranaje primario en una posición específica en la mesa de ensamblaje.

#### Etapa 2

4. Acercarse al eje.
5. Levantar el eje.
6. Situar el eje en una posición específica en la mesa de ensamblaje.

#### Etapa 3

7. Acercarse al engranaje secundario.
8. Levantar el engranaje secundario.
9. Situar el engranaje secundario en una posición específica en la mesa de ensamblaje.

Considerando esta secuencia de acciones DRL se encargará de las dos primeras acciones de la primera etapa, Acercamiento y levantamiento del engranaje principal, la reubicación se realizará con una trayectoria fija hasta el punto final.

Para las siguientes etapas se usará las coordenadas obtenidas en las predicciones para trazar una trayectoria fija hacia el eje y el engranaje secundario, posterior al

levantamiento de estas piezas se define una trayectoria fija para la reubicación de las mismas en la mesa de ensamblaje.

#### **2.4.4 Definición de la función recompensa**

La función recompensa desempeña un papel fundamental en el aprendizaje del algoritmo, ya que proporciona retroalimentación al agente sobre el rendimiento de las acciones. En este caso la función recompensa será definida para las dos primeras acciones de la primera etapa.

La etapa consta de 3 movimientos acercamiento, levantamiento y reubicación, considerando la secuencia de los objetos. Para el acercamiento, se construye una trayectoria con las coordenadas transformadas de la predicción realizada por YOLO. Para el levantamiento el robot debe sujetar la pieza y elevarla hasta una altura específica. Por último, la reubicación se realiza siguiendo una trayectoria preestablecida en la cual se tiene una posición específica en la mesa de ensamblaje. Para que el robot llegue a estas posiciones con la configuración correcta de sus articulaciones, se emplea cinemática directa, la cinemática directa consiste en calcular la posición y orientación final en función de las coordenadas y ángulos de sus articulaciones. De esta manera, se obtiene una configuración de articulaciones para cada reubicación.

Para cada etapa se ha definido una componente para cuantificar su desempeño.

*Acercamiento* Se basa en la distancia entre la posición del gripper y la posición del objeto, si el gripper del robot está en posiciones muy cercanas, menores a 7 cm, esta recompensa toma el valor de 15; caso contrario toma el valor definido en la Ecuación 6.

$$A = 10 * e^{|10*(p1-p2)|} \quad (6)$$

Donde:

*p1: Posición del gripper*

*p2: Posición actual*

*Levantamiento* Se basa en el aumento de altura. Si la pieza supera un umbral se le asigna una recompensa de 400; caso contrario se le asigna una recompensa proporcional al cambio de distancia descrita en la Ecuación.7.

$$L = 100 *(h2 - h1) \quad (7)$$

Donde:

*h1: Altura inicial*

*h2: Altura actual*

*Reubicación* Si la posición del objeto está en la posición de ensamblaje obtiene una ganancia de 1000.  $U = 1000$

Sin embargo, dentro de esta tarea también existe una restricción importante, durante el desarrollo de una etapa, el robot no puede golpear las otras piezas que se encuentran en la mesa de trabajo. En caso de que esto ocurra, el robot recibirá una penalización y esta se evalúa en función de la posición de los objetos restantes en la Mesa de trabajo.

Considerando todo lo mencionado previamente la función recompensa se estructura como se muestra en el Ecuación 8.

$$R_1 = Acercamiento + Levantamiento + Reubicación \quad (8)$$

### **3. CAPÍTULO 3**



### **3.1 Resultados y Análisis**

En este capítulo, se expone de forma estructurada y detallada los hallazgos más relevantes de esta investigación.

#### **3.1.1 Entorno de Trabajo**

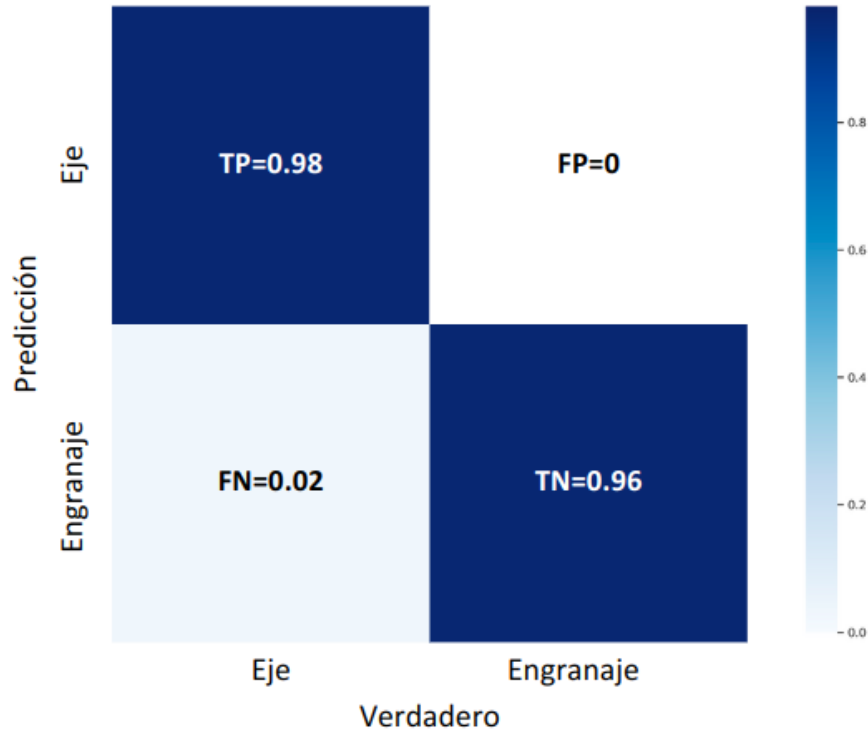
Tras integrar y ajustar los elementos en el entorno de trabajo, se consiguió adaptarlo con precisión a nuestras necesidades. Esta adaptación ofreció una representación clara y fiel, facilitando un análisis más detallado y controlado de la situación. Dentro de este marco, es posible observar las tres piezas (Engranaje primario, engranaje secundario y eje ) llegar de forma aleatoria a la mesa de trabajo. Además, se optimizó la ubicación de la cámara para garantizar una visión sin obstrucciones, lo que brindó una perspectiva mejorada. La representación completa de este entorno modificado se muestra en la figura.

#### **3.1.2 Sistema de Visión**

En el campo de la inteligencia artificial, es indispensable evaluar los modelos para medir su desempeño. En los modelos de visión por computadora, se utiliza la matriz de confusión. Esta matriz destaca cuatro métricas esenciales, verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN). Con estas métricas, no solo se puede determinar cuántas veces un modelo realizó predicciones correctas, sino también identificar las ocasiones en las que se equivocó. El modelo de identificación y localización desarrollado en este proyecto cuenta con una matriz de confusión que se muestra en la Figura 12

**Figura 12**

*Matriz de Confusión del modelo de visión*



En la matriz de confusión, el valor TP es 0.98, representa el porcentaje de ejes estimados correctamente. FP tiene un valor de 0, indicando que ningún engranaje fue erróneamente clasificado como eje. Por su parte, FN muestra un valor de 0.02, reflejando los ejes que fueron incorrectamente clasificados como engranajes. Finalmente, TN tiene un valor de 0.96, correspondiente al porcentaje de engranajes identificados de forma acertada.

A partir de estas métricas podemos calcular estadísticas que nos informan sobre el desempeño de este modelo. Estas estadísticas son exactitud, precisión, sensibilidad y F1 score.

### *Exactitud*

La exactitud nos indica que porcentaje de predicciones acertó el modelo esta se calcula como se indica en la Ecuación 9.

$$Exactitud = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Donde:

TP : Verdaderos Positivos.

TN : Verdaderos Negativos.

FP : Falsos Positivos.

FN : Falsos Negativos.

Para nuestro modelo este valor corresponde al 99.4% lo que indica que el modelo tiene un excelente desempeño en la clasificación.

### *Precisión*

La precisión es una estadística la cual se refiere a cuan cercanos son los valores de predicción referente a los valores reales esta se calcula como se muestra en la Ecuación 10.

$$Precisión = \frac{TP}{TP+FP} \quad (10)$$

Donde:

TP : Verdaderos Positivos.

FP : Falsos Positivos.

El valor de precisión de este modelo es de 99.4% lo que quiere decir que el modelo presenta una alta fiabilidad de la predicción respecto a los valores reales.

Sensibilidad.

También conocida como la tasa de verdaderos positivos, es una estadística que permite identificar la proporción de casos positivos reales que fue identificada correctamente por el modelo. Se calcula como lo expresa la Ecuación 11.

$$\text{Sensibilidad} = \frac{TP}{TP+FN} \quad (11)$$

Donde:

TP : Verdaderos Positivos.

FN : Falsos Negativos.

En este modelo la sensibilidad es del 98.9% con lo que se indica que todos los casos positivos reales fueron identificados por el modelo.

*F1-score*

F1-score es una medida armónica de la precisión y la sensibilidad, para obtener un balance entre ambas. En la Ecuación 12 se muestra cómo se calcula.

$$F1 - score = 2 * \frac{\text{Precisión} * \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}} \quad (12)$$

El valor de F1 del modelo es de 99.1%, e indica que tanto la precisión como la sensibilidad del modelo son altas. En otras palabras, el modelo tiene un buen rendimiento al identificar positivos y, al mismo tiempo, limita la cantidad de falsos positivos.

**Tabla 1***Estadísticas de desempeño del modelo de visión*

	Exactitud	Precisión	Sensibilidad	F1-score
Porcentaje	99.4	99.4	98.9	99.1

Como se aprecia en la Tabla 1 este modelo ha demostrado un desempeño sobresaliente considerando sus estadísticas. Con una exactitud del 99.4%, queda claro que el modelo acierta en la mayoría de sus predicciones. Su precisión del 99.4% reafirma que, cuando el modelo señala un resultado como positivo, hay una alta probabilidad de que efectivamente sea así. Sin embargo, lo que resalta notablemente es su sensibilidad del 98.9%, lo que indica que el modelo logra identificar y reconocer la gran mayoría de los casos positivos reales, fallando muy pocas veces en este aspecto. Acompañando a estos indicadores, el valor F1-score de 99.1% confirma que el modelo ha conseguido un equilibrio casi perfecto entre precisión y sensibilidad, garantizando una confiabilidad integral en sus predicciones. En conjunto, estos valores no solo indican la precisión global del modelo, sino que también destacan su capacidad para ser altamente confiable en situaciones críticas donde se requiere identificar casos positivos de manera certera.

### ***3.1.3 Transformación de coordenadas***

Para la tarea de transformación, es posible identificar discrepancias entre la posición real y la estimadas de los objetos luego de la transformación. Dado que cada pieza tiene una geometría única, presentan errores distintos en su localización.

Para determinar el error en la transformación de coordenadas se utilizó la Ecuación

$$Error = \|PR - PT\| \quad (13)$$

Donde:

PR : Coordenadas de la posición real

PT : Coordenadas estimadas

Se llevaron a cabo 100 simulaciones del entorno, en cada una se registraron los errores correspondientes a cada pieza. Estos valores fueron almacenados y posteriormente se calculó el promedio de dichos errores. Al comparar la ubicación real con la transformada, se descubrió que el modelo presenta los errores descritos en la Tabla 2

**Tabla 2**

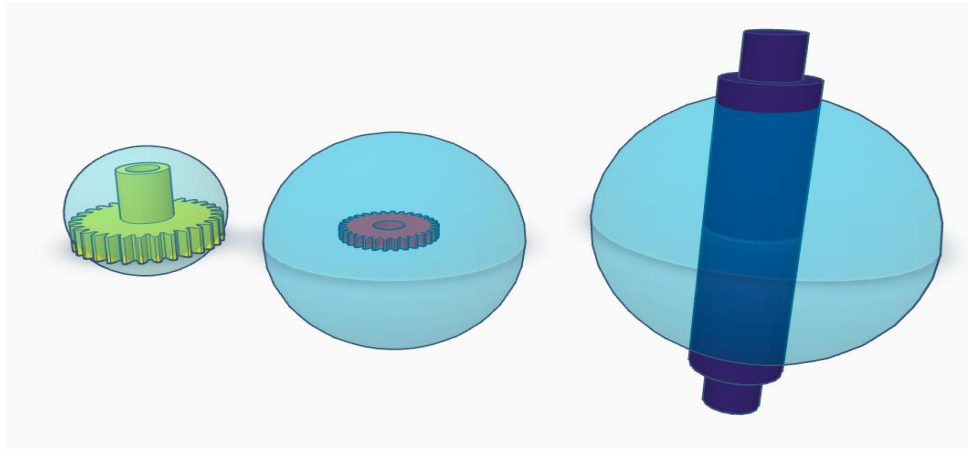
*Radio de la esfera de error de la transformación de coordenadas*

	Engranaje Primario	Engranaje Secundario	Eje
Error [cm]	4.072	6.906	18.709

Considerando este error, se identifica la “esfera de error”, la cual se utiliza para describir el error en las mediciones de posición. Es decir, la transformación del punto en el espacio podría estar en cualquier lugar dentro de esa esfera, tomando el centro de la esfera como la posición real. En la Figura 13 se observa las esferas de error en contraste con las piezas.

**Figura 13**

*Esfera de error de la transformación de coordenadas*



Para evaluar la precisión de los resultados obtenidos, es importante considerar el cálculo del error relativo. Este cálculo se detalla en la Ecuación 14.

$$Error\ Relativo\ \% = \frac{\|PR-PT\|}{\|PR\|} * 100 \quad (14)$$

Donde:

PR : Coordenadas de la posición real

PT : Coordenadas estimadas

Para pieza se detallan los errores relativos promedio, como se especifica en la Tabla 3. Un análisis de estos datos sugiere que el modelo está logrando predicciones que se asemejan considerablemente a los resultados reales en la mayoría de las instancias.

**Tabla 3**

*Error Relativo de la transformación de coordenadas*

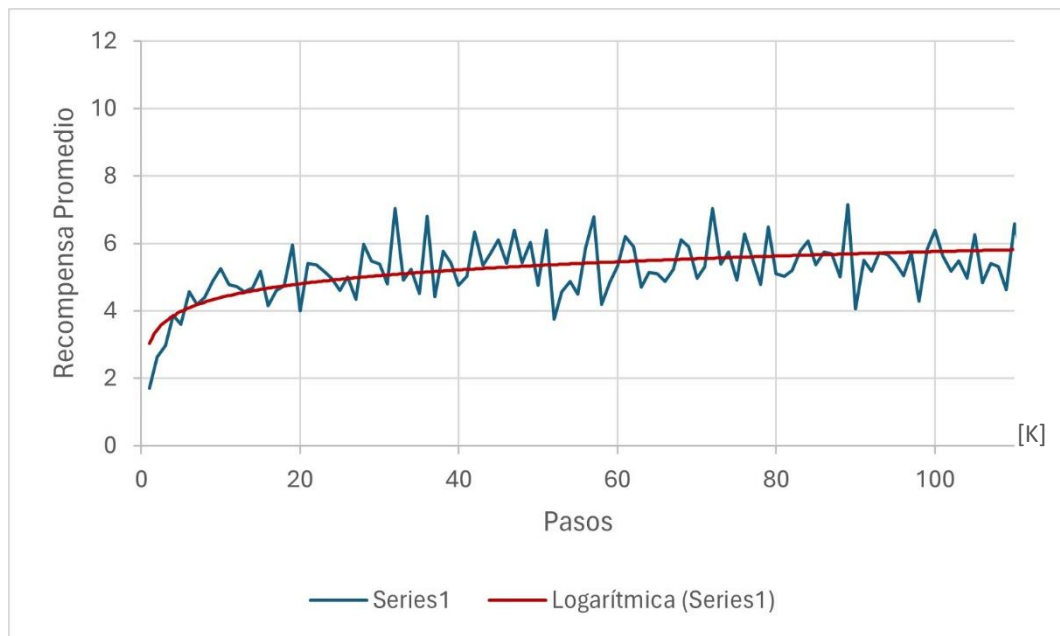
	<b>Engranaje Primario</b>	<b>Engranaje Secundario</b>	<b>Eje</b>
<b>Error Relativo %</b>	3.931	6.396	18.275

### 3.1.4 Resultados del DRL

Para el desarrollo del entrenamiento, se definieron episodios, que son secuencias compuestas por 200 pasos cada uno, a su vez los pasos se refieren a una acción tomada por el agente. Cada iteración del entrenamiento consistió en la ejecución de 1000 de estos episodios. Esta estrategia permitió examinar y verificar la evolución del sistema a lo largo de múltiples conjuntos de episodios, asegurando una evaluación completa. Adicionalmente, se guardaron los parámetros y pesos resultantes del entrenamiento, para respaldar un análisis detallado y futuras investigaciones en el mismo ámbito. En la Figura 14 se muestra la relación entre el número de pasos y la recompensa promedio obtenida, en función de la proximidad al objetivo, durante la etapa inicial del entrenamiento.

**Figura 14**

*Gráfica de Recompensa Promedio y Pasos en la etapa inicial*



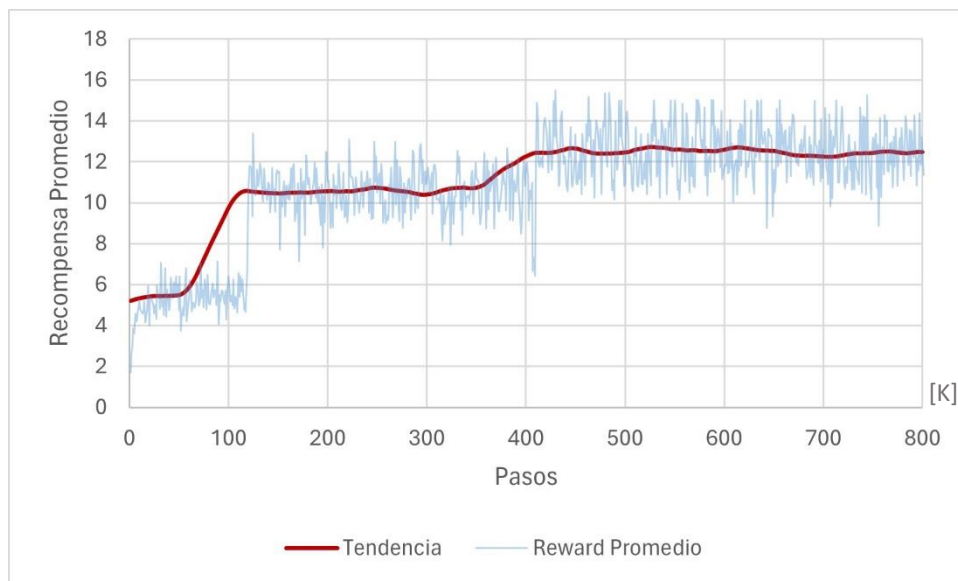
*Nota.* El gráfico representa con azul la recompensa obtenida durante cada paso y con rojo la línea de tendencia que forman estos datos.



En los primeros pasos, la recompensa es baja, señalando la distancia del agente al objeto objetivo. Durante esta fase, explora el entorno y experimenta con distintas acciones. Conforme los pasos aumentan, la recompensa crece de forma gradual, indicando el aprendizaje del agente en la aproximación al objetivo y la toma de decisiones más acertadas. La línea ascendente en la gráfica refleja la mejora y adaptación positiva del sistema. A medida que avanza el entrenamiento, logra mejores resultados, indicando una evolución favorable en su capacidad para alcanzar el objetivo deseado.

**Figura 15**

*Gráfica de Recompensa Promedio y Pasos*



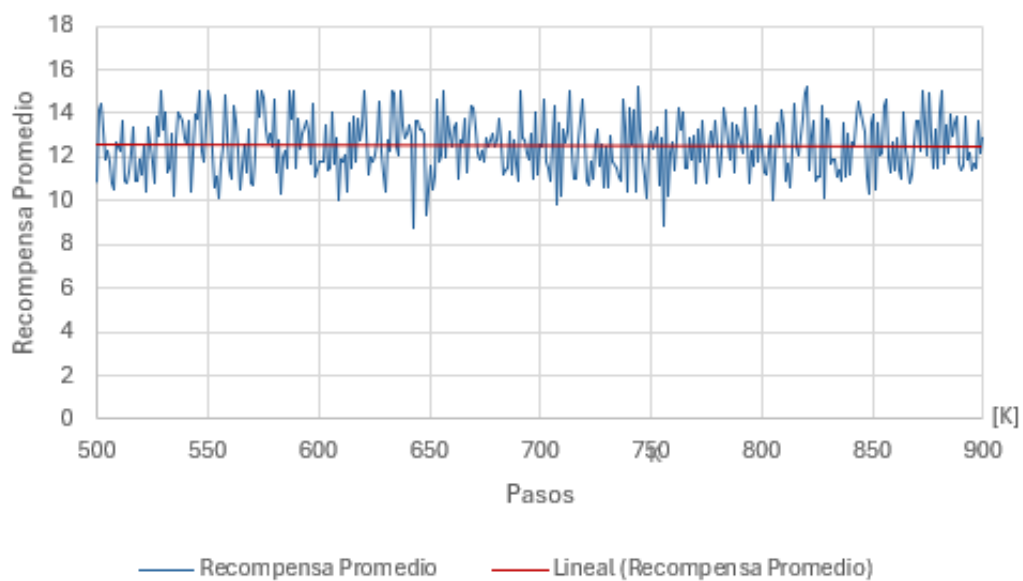
*Nota.* El gráfico representa con azul la recompensa obtenida durante cada paso y con rojo la línea de tendencia que forman estos datos.

La Figura 15 muestra una progresión gradual y consistente en el desempeño del sistema. En la primera etapa, se aprecia un crecimiento hasta alcanzar un valor alrededor de 5, indicando un proceso inicial de adaptación y aprendizaje. Posteriormente, en la siguiente fase, esta mejora se intensifica, llegando hasta un punto cercano a 10, lo que

sugiere un refinamiento y ajuste más profundo de las capacidades del sistema. La evolución parece consolidarse en la última etapa, manteniendo un nivel constante, lo que podría indicar que el sistema ha encontrado una estrategia óptima y estable para la tarea de acercamiento en cuestión.

### Figura 16

*Gráfica de Recompensa Promedio y Pasos Detalle*



*Nota.* El gráfico representa con azul la recompensa obtenida durante cada paso y con rojo la línea de tendencia que forman estos datos

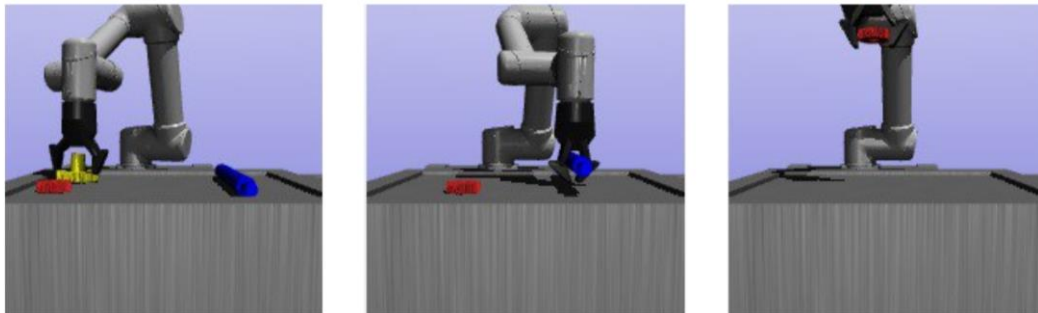
En la Figura 16, se observa una estabilización con un valor cercano a 15. Este punto de equilibrio sugiere que el sistema ha alcanzado un nivel de rendimiento consistente y predecible. La convergencia en este valor señala una fase en la que el proceso de entrenamiento ha afinado el comportamiento del sistema hasta lograr una respuesta sólida y confiable. Este hallazgo resalta la efectividad del proceso de aprendizaje.

En resumen, la gráfica muestra una tendencia positiva en la recompensa a medida que el agente aprende a acercarse al objeto objetivo. El agente pasa por una fase de exploración y aprendizaje antes de lograr una mejor recompensa, lo que sugiere que ha aprendido con éxito a realizar acciones que lo acercan significativamente al objetivo deseado.

En la Figura 17 se muestra la secuencia de imágenes que documenta el proceso de ensamblaje durante la secuencia descrita previamente.

### **Figura 17**

*Secuencia de ensamblaje*



### 3.1.5 Análisis de Costos

En caso de necesitar llevar este proyecto al ámbito práctico, se han determinado los rubros generales necesarios para su desarrollo, los cuales están detallados en la Tabla 4.

**Tabla 4**

*Rubros de la implementación del proyecto*

<b>Rubro</b>	<b>Descripción</b>	<b>Cantidad</b>	<b>Precio Unitario</b>	<b>Precio Total</b>
<b>Hardware</b>	GPU y componentes de hardware	1	\$1500	\$1500
<b>Rubro</b>	<b>Descripción</b>	<b>Tiempo mensual</b>	<b>Precio Unitario</b>	<b>Precio Total</b>
<b>Energía Eléctrica</b>	Consumo eléctrico por hardware	4	\$20	\$80
<b>Personal</b>	Recursos humanos	4	\$1000	\$4000
			<b>Total</b>	<b>\$5580</b>

La viabilidad de esta investigación se establece según las exigencias de la industria en la que se implemente. Basándonos en el análisis de costos y los recursos requeridos, se puede concluir que el proyecto es factible en términos de desarrollo.

## **4. CAPÍTULO 4**

## 4.1 Conclusiones y recomendaciones

### 4.1.1 Conclusiones

El presente proyecto se ha enfocado en la incorporación de técnicas de aprendizaje autónomo para automatizar procesos de ensamblaje de ejes de transmisión, con el objetivo clave de aumentar la producción y reducir las fallas debidas a errores humanos. La información obtenida del análisis de resultados y desempeño brinda una visión sobre el potencial de esta estrategia y permite concluir lo siguiente:

- El sistema de visión, integra el modelo desarrollado con YOLO y la transformación de coordenadas. El modelo presenta un valor F1 de 99.1%, lo que corrobora que ha logrado un equilibrio sobresaliente entre precisión y sensibilidad, asegurando así su capacidad predictiva. En relación con la transformación de coordenadas, estas muestran una fiabilidad superior al 80%, lo que implica que, las coordenadas transformadas son muy próximas a las reales. El sistema en conjunto es capaz de detectar y localizar con precisión las distintas piezas del eje de transmisión localizadas de forma aleatoria en el espacio de trabajo y se demuestra que se ha desarrollado con éxito un sistema confiable y robusto.
- El diseño de un sistema de control, basado en el algoritmo TD3 y complementado con una estructura de recompensa por etapas, ha demostrado gran eficacia al guiar el entrenamiento, esto se puede evidenciar en rendimiento del robot en la tarea. La estructura de recompensas ha ofrecido una guía para el aprendizaje del robot, asegurando la evolución de sus acciones con los objetivos de cada etapa.
- Considerando las recompensas obtenidas, queda claro que el algoritmo ha demostrado la capacidad para aprender y adaptarse a

medida que avanza en su proceso de entrenamiento. El comportamiento inicial de exploración se transforma en una evolución constante de decisiones más acertadas, permitiendo al agente acercarse de manera efectiva al objetivo. La recompensa máxima alcanzada, marcada por un valor de 415, refleja una proximidad muy cercana al objetivo y el levantamiento exitoso de la pieza durante el entrenamiento, indicando un alto grado de eficiencia en el aprendizaje. Además, el análisis de la tendencia de entrenamiento confirma que a medida que aumenta la cantidad de entrenamiento, se observan mejoras significativas en los resultados.

#### ***4.1.2 Recomendaciones***

- Una recomendación crucial al trabajar en el entrenamiento de modelos de visión es proporcionar conjuntos de datos de entrenamiento que sean lo más representativos posible de las situaciones que se encontrarán en la implementación práctica. Al hacerlo, se fortalece la capacidad del modelo de visión para adaptarse y responder eficazmente a las complejidades y variaciones del entorno real.
- Para enfrentar desafíos complejos en contexto del aprendizaje automático, es esencial descomponer los problemas en etapas más manejables. Esta división facilita el aprendizaje y asegura la toma correcta de acciones en cada fase. En entornos desafiantes, abordar problemas de manera estructurada garantiza decisiones más precisas y resultados óptimos.

- Basados en los resultados obtenidos, se sugiere seguir enfocando los esfuerzos en el proceso de entrenamiento y refuerzo, que ha demostrado ser efectivo. Es importante continuar con la iteración y optimización del algoritmo para lograr una estabilización en la recompensa.



## Referencias

- Ahn, K.-H., Na, M., & Song, J.-B. (2023). Robotic assembly strategy via reinforcement learning based on force and visual information. *Robotics and Autonomous Systems*, 104399. <https://doi.org/10.1016/j.robot.2023.104399>
- Corke, P. (2011). *Robotics, Vision and Control*. <https://doi.org/10.1007/978-3-642-20144-8>
- Daniel, P. (2022, November 21). *Deep Reinforcement Learning for robotic pick and place applications using purely visual observations*. [https://github.com/PaulDanielML/MuJoCo\\_RL\\_UR5](https://github.com/PaulDanielML/MuJoCo_RL_UR5)
- Dhillon, B. (2009). *Human reliability, error, and human factors in engineering maintenance: with reference to aviation and power generation*. <https://books.google.com/books?hl=es&lr=&id=211nOYpB7kYC&oi=fnd&pg=PP1&ots=FSiyN0inbK&sig=vPARsUETN8VrqKwu1mBMCza-2Fg>
- Di Pasquale, V., Miranda, S., Iannone, R., & Riemma, S. (2015). A Simulator for Human Error Probability Analysis (SHERPA). *Reliability Engineering & System Safety*, 139, 17–32. <https://doi.org/10.1016/J.RESS.2015.02.003>
- Elmaraghy, W. H., Nada, O. A., & Elmaraghy, H. A. (2008). Quality prediction for reconfigurable manufacturing systems via human error modelling. *International Journal of Computer Integrated Manufacturing*, 21(5), 584–598. <https://doi.org/10.1080/09511920701233464>
- Franceschetti, A., Tosello, E., Castaman, N., & Ghidoni, S. (2020). *Robotic Arm Control and Task Training through Deep Reinforcement Learning*. <http://arxiv.org/abs/2005.02632>

- Fruggiero, F., Fera, M., Lambiase, A., Maresca, P., & Caja, J. (2017). The role of human fatigue in the uncertainty of measurement. *Procedia Manufacturing*, *13*, 1320–1327. <https://doi.org/10.1016/J.PROMFG.2017.09.092>
- Fruggiero, F., Riemma, S., Ouazene, Y., Macchiaroli, R., & Guglielmi, V. (2016). Incorporating the Human Factor within Manufacturing Dynamics. *IFAC-PapersOnLine*, *49*(12), 1691–1696. <https://doi.org/10.1016/J.IFACOL.2016.07.825>
- Fujimoto, S., van Hoof, H., & Meger, D. (2018). *Addressing Function Approximation Error in Actor-Critic Methods*.
- Gilles, T. (2015). *Automotive service: Inspection, maintenance, repair*. 177–176. [https://books.google.com/books/about/Automotive\\_Service\\_Inspection\\_Maintenanc.html?hl=es&id=Ji5-BAAAQBAJ](https://books.google.com/books/about/Automotive_Service_Inspection_Maintenanc.html?hl=es&id=Ji5-BAAAQBAJ)
- Goodfellow, I., Bengio, Y., Courville, A., & Heaton, J. (2017). Deep learning. *Genetic Programming and Evolvable Machines* *2017 19:1*, *19*(1), 305–307. <https://doi.org/10.1007/S10710-017-9314-Z>
- Griffith, C. D., & Mahadevan, S. (2011). Inclusion of fatigue effects in human reliability analysis. *Reliability Engineering & System Safety*, *96*(11), 1437–1447. <https://doi.org/10.1016/J.RESS.2011.06.005>
- Groover, M. P. (2008). *Automation, production systems, and computer-integrated manufacturing*. 815. [https://books.google.com/books/about/Automation\\_Production\\_Systems\\_and\\_Comput.html?hl=es&id=I-dQPgAACAAJ](https://books.google.com/books/about/Automation_Production_Systems_and_Comput.html?hl=es&id=I-dQPgAACAAJ)
- Hou, Y., & Li, J. (2023). Learning 6-DoF grasping with dual-agent deep reinforcement learning. *Robotics and Autonomous Systems*, *166*. <https://doi.org/10.1016/j.robot.2023.104451>

- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). *Continuous control with deep reinforcement learning*.
- Michalos, G., Makris, S., & Chryssolouris, G. (2013). The effect of job rotation during assembly on the quality of final product. *CIRP Journal of Manufacturing Science and Technology*, 6(3), 187–197. <https://doi.org/10.1016/J.CIRPJ.2013.03.001>
- Morales, M. (2020). *Grokking Deep Reinforcement Learning*.
- MuJoCo. (2022). *Descripción general - Documentación de MuJoCo*. <https://mujoco.readthedocs.io/en/latest/overview.html>
- Ochoa-Jiménez, D., Armas-Herrera, R., & Pereira, C. (2022). Manufacturas y crecimiento económico en Ecuador bajo una perspectiva regional. Un modelo de panel dinámico, 2007 -2020. *Revista Vista Económica*, 10(1), 31–44. <https://doi.org/10.54753/rve.v10i1.1290>
- Raschka, S., & Mirjalili, V. (2020). *Python Machine Learning*. <https://www.worldcat.org/title/1237379311>
- Russel, S., & Norvig, P. (2016). Artificial intelligence A modern approach . *The Knowledge Engineering Review*. <https://doi.org/10.1017/S0269888900007724>
- Salas-Arias, K. M., Elena Madriz-Quirós, C., Sánchez-Brenes, O., Sánchez-Brenes, M., & Bautista Hernández-Granados, J. (2018). *Factores que influyen en errores humanos en procesos de manufactura moderna Factors that influence human errors in modern manufacturing processes*. 28. <https://doi.org/10.18845/tm.v31i1.3494>
- Siciliano, B., & Khatib, O. (2016). Springer handbook of robotics. *Springer Handbook of Robotics*, 1–2227. <https://doi.org/10.1007/978-3-319-32552-1/COVER>
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning An Introduction*.

United Nations Industrial Development Organization. (2021). *INFORME SOBRE EL  
DESARROLLO INDUSTRIAL 2022.*

[www.unido.org](http://www.unido.org) [unido@unido.org](mailto:unido@unido.org) [www.unido.org](http://www.unido.org) [unido@unido.org](mailto:unido@unido.org)