

**Instituto Tecnológico de Costa Rica**

**Área Académica de Ingeniería Mecatrónica**



**Implementación de Inteligencia Artificial en un módulo práctico de Robótica  
para la enseñanza a estudiantes preuniversitarios**

**Proyecto de graduación para optar por el título de Ingeniero en Mecatrónica  
con el grado académico de Licenciatura**

**Mayda Lucía Arce Gómez**

**2017100116**

**Alajuela, julio del 2024**

Implementación de Inteligencia Artificial en un módulo práctico de Robótica para la enseñanza a estudiantes preuniversitarios © 2024 by Mayda Lucía Arce Gómez is

licensed under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/) 

**INSTITUTO TECNOLÓGICO DE COSTA RICA**  
**PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA**  
**PROYECTO FINAL DE GRADUACIÓN**  
**ACTA DE APROBACIÓN**

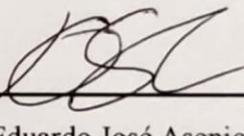
Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

Estudiante: Mayda Lucía Arce Gómez

Proyecto: Implementación de Inteligencia Artificial en un módulo práctico de robótica para la enseñanza a estudiantes preuniversitarios.

Miembros del jurado evaluador

  
\_\_\_\_\_  
Dr. -Ing. Juan Luis Crespo Mariño  
Jurado

  
\_\_\_\_\_  
Ing. Eduardo José Asenjo Castillo  
Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

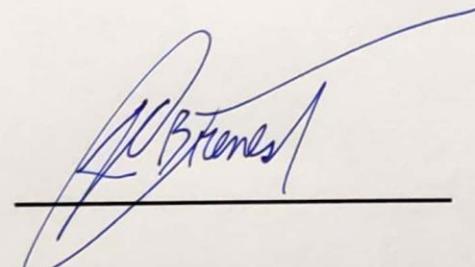
Cartago, 27 de agosto 2024

**INSTITUTO TECNOLÓGICO DE COSTA RICA  
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA  
PROYECTO FINAL DE GRADUACIÓN  
ACTA DE APROBACIÓN**

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: Mayda Lucía Arce Gómez

Proyecto: Implementación de Inteligencia Artificial en un módulo práctico de robótica para la enseñanza a estudiantes preuniversitarios.



Msc. -Ing. Juan Carlos Brenes Torres

Asesor

Cartago, 27 de agosto 2024

## Resumen

En este documento se describirá el desarrollo necesario para la creación de lecciones de Inteligencia Artificial con la implementación de un brazo robótico *Magician* de **Dobot** para el aprendizaje de estudiantes preuniversitarios.

Se verá la metodología de Eppinger y Ulrich, la cual se aplicará para la creación de las lecciones. Donde además se seleccionará un brazo robótico para su compra e implementación de Inteligencia Artificial.

También se realizará un análisis de cuatro modelos de Inteligencia Artificial de tipo de aprendizaje supervisado (árboles de decisión, regresión polinomial, vecinos cercanos/kNN's y red neuronal).

Para la unión del brazo robótico con la Inteligencia Artificial, se decidió usar el tipo de sensor de cámara. Esta se añade al brazo robótico en la muñeca y de la cual se sacaron imágenes para la creación de la base de datos a usar para el aprendizaje supervisado.

Donde se implementó cada modelo con datos obtenidos por el robot y se obtuvieron resultados de MSE y  $R^2$  y diferentes gráficas. Se detallará cada gráfica, junto con un análisis de resultados respectivo y se seleccionará un modelo ganador para las lecciones.

Palabras *clave*:

Árboles de decisión, Dobot, inteligencia artificial, kNN's, *Magician*, red neuronal, regresión polinomial, vecinos cercanos.

## Abstract

In this document is going to be described the development necessary for the creation of Artificial Intelligence lessons with the implementation of a robotic arm (*Magician*, from **Dobot**) for the learning of pre-university students.

Eppinger and Ulrich methodology will be used for the creation of the teaching units. In which the robotic arm will be chosen for purchase and the Artificial Intelligence implementation.

In addition, there will be an analysis of four Artificial Intelligence models of supervised learning (decision trees, nearest neighbors/kNN's, neural network and polynomial regression).

For the union of Artificial Intelligence in the robotic arm, the usage of the sensor type was decided: camera. It was added to the wrist of the robot and employed to take snapshots for the dataset used in supervised learning.

Each model was implemented in the software, in which results and graphics were obtained. Each graphic will be detailed with its respective analysis and only one model will be selected for the teaching units.

### *Keywords:*

Artificial Intelligence, decision trees, Dobot, kNN's, Magician, nearest neighbors, neural network, polynomial regression.

## Dedicación

A mis padres y hermano que me han apoyado y aconsejado toda la vida.

## **Agradecimientos**

Primero agradezco mucho a mis padres, Olger Arce y Ana Leonor Gómez, por todo el apoyo que me han brindado durante toda mi vida y mis estudios; velando siempre que no me faltara nada. Además de mi amigo, compañero de cuarto y hermano Ólger Arce; porque sin su ayuda este camino habría sido mucho más difícil.

También quiero agradecer a todos los profesores que me han enseñado en el Tecnológico de Costa Rica, en el colegio y escuela; porque de todos he aprendido algo que me ayudó a llegar a este punto y a ser quien soy. En especial, a mi profesor asesor Juan Carlos Brenes, por todo el apoyo académico y orientación crítica durante el desarrollo del proyecto.

Igualmente, en la Universidad de Coruña, mis asesores Francisco Bellas, Abraham Prieto y el Grupo Integrado de Ingeniería por la oportunidad de realizar el proyecto en sus instalaciones y brindar orientación necesaria para el proyecto.

De igual manera le agradezco a todo el personal y funcionarios de las instituciones en las que he estudiado y la Escuela de Mecatrónica, porque han influenciado a mis estudios desde contar con instalaciones limpias hasta la posibilidad de los cursos llevados. Con particular agradecimiento al coordinador de los laboratorios LAIMI del Tecnológico sede de Cartago, Bryan Aguirre, que me brindó oportunidades de realizar asistencias que me ayudaron con la continuación de mis estudios y en donde adquirí experiencia en labores tecnológicas y de supervisión.

Por último, quiero agradecer a los amigos que he tenido en este camino y a toda mi familia, porque fueron vitales en mis estudios. Aunque su apoyo pareciera sutil, me ayudó a perseverar y mantenerme enfocada en mis metas.

## **Declaratoria de Autenticidad**

Declaro que el presente Anteproyecto de Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema y asesoramiento técnico de miembros del Grupo Integrado de Ingeniería (GII) de la Universidad de Coruña.

Asumo completa responsabilidad sobre el trabajo realizado y por el correspondiente contenido.

Ferrol, España, enero 2024

Cédula: 1-1734-0770

Mayda Lucía Arce Gómez

## **Contenido**

1.	Introducción .....	1
1.1	Entorno del proyecto .....	1
1.2	Definición del problema.....	1
1.2.1	Generalidades .....	1
1.3	Síntesis del problema .....	2
1.4	Objetivo General .....	2
1.5	Objetivos Específicos.....	2
2.	Marco Teórico .....	4
2.1	Robótica .....	4
2.1.1	Anatomía de los robots.....	4
2.1.2	Espacio de trabajo.....	7
2.2	Sistemas de Visión .....	7
2.2.1	La cámara .....	8
2.2.2	Segmentación: Espacios de color y máscaras de color.....	8
2.2.3	Binarización de imágenes.....	10
2.3	Inteligencia Artificial .....	10
2.3.1	Tipos de aprendizaje en Inteligencia Artificial.....	10
2.3.2	Base de datos .....	11
2.3.3	Modelos de entrenamiento supervisado .....	12
3.	Descripción de la metodología .....	15
3.1	Etapa 1: Identificación y determinación de necesidades .....	15
3.2	Etapa 2: Establecimiento de especificaciones objetivo .....	15
3.3	Etapa 3: Generación de conceptos.....	16
3.4	Etapa 4: Selección de conceptos .....	16
3.5	Etapa 5: Pruebas de concepto.....	16
3.6	Etapa 6: Documentación del proyecto .....	17
3.7	Subetapa: Modelado y construcción de prototipos .....	17
4.	Desarrollo.....	18
4.1	Primera Etapa: Definición de necesidades .....	18
4.2	Segunda Etapa: Establecimiento de especificaciones .....	20
4.3	Tercera y cuarta etapa: Generación y selección de conceptos .....	22

4.3.1	Brazo robótico: Generación y selección de conceptos.....	24
4.3.2	Sensor: Generación y selección de conceptos .....	34
4.3.3	Inteligencia Artificial: Generación y selección de conceptos.....	37
4.4	Quinta etapa: Pruebas de concepto.....	41
4.4.1	Prueba de concepto del robot .....	41
4.4.2	Prueba de concepto del sensor .....	41
4.4.3	Prueba de concepto de Inteligencia Artificial.....	42
4.5	Sexta etapa: Documentación .....	42
5.	Creación de la Inteligencia Artificial .....	43
5.1	Lógica de movimiento para la base de datos .....	43
5.2	Creación de la base de datos .....	47
5.3	Programación de los modelos de Inteligencia Artificial .....	51
6.	Lecciones y Actividades Docentes .....	54
6.1	Learning by doing.....	54
6.2	Descripción de actividades y retos .....	54
1.1.1	Información necesaria de las lecciones: Implementación en el robot real .....	56
6.2.2	Información necesaria de la cuarta lección: Base de datos.....	57
7.	Análisis de resultados.....	59
7.1	Resultados de los modelos de Inteligencia Artificial .....	59
7.2	Curva de aprendizaje: Análisis de resultados.....	60
7.2.1	Curvas de aprendizaje: Resultados obtenidos.....	61
7.3	Predicciones vs Datos Reales: Análisis de resultados .....	64
7.3.1	Predicciones vs Datos reales: Resultados Obtenidos .....	65
7.4	Gráficos de residuos: Análisis de resultados .....	67
7.4.1	Gráfico de residuos: Resultados obtenidos.....	67
7.5	Gráfica de Q-Q: Análisis de resultados .....	70
7.5.1	Gráfica de Q-Q: Resultados obtenidos .....	70
7.6	Errores de Distancia: Análisis de resultados.....	73
7.6.1	Errores Q-Q: Resultados obtenidos.....	73
7.7	Resumen de análisis de los modelos de Inteligencia Artificial: Análisis de resultados ....	76
1.1.1	Resultados de MSE y R <sup>2</sup> : Resultados obtenidos .....	77
7.8	Movimientos del robot simulado y real .....	77

7.8.1	Primera lección: movimientos del robot simulado y real .....	77
7.8.2	Segunda lección: movimientos del robot simulado .....	78
7.8.3	Tercera lección: movimientos del robot simulado .....	78
7.8.4	Quinta lección: movimientos del robot simulado y real.....	80
7.9	Población muestra .....	80
7.9.2	Primera Lección: resultados de la población muestra.....	83
7.9.3	Segunda Lección: resultados de la población muestra.....	83
7.9.4	Tercera Lección: resultados de la población muestra .....	83
7.9.5	Cuarta Lección: resultados de la población muestra.....	84
8.	Análisis económico.....	85
8.1	Recursos Empleados para el proyecto .....	86
8.2	Recursos generados para el proyecto .....	87
9.	Conclusiones y recomendaciones .....	90
9.1	Conclusiones .....	90
9.2	Recomendaciones .....	91
10.	Referencias.....	92
11.	Anexos.....	94
	Anexo A. Carta de Aceptación .....	94
	Anexo B. Hoja de información del proyecto .....	95
	Anexo C: Cuestionario de la primera lección: Robótica y actuadores .....	97
	Anexo D: Cuestionario de la segunda lección: Sensores.....	98
	Anexo E: Cuestionario de la tercera lección: Base de datos .....	99
	Anexo F: Cuestionario de la cuarta lección: Modelo de Inteligencia Artificial .....	100
	Anexo G: Códigos creados para las lecciones .....	101

## Contenido de figuras

Figura 2.1.	Tipos de articulaciones. Referencia: [14].....	5
Figura 2.2.	Anatomías del robot. Referencia [14].....	6
Figura 2.3.	Manipulador de 6 DOF. [14] .....	6
Figura 2.4.	Área de trabajo de los robots. [14].....	7
Figura 2.5.	Espacios de color. [15].....	9
Figura 2.6.	Lógica de los árboles de decisión [16] .....	12
Figura 2.7.	Ejemplo de regresión lineal con características polinómicas. [16] .....	13
Figura 2.8.	Predicciones hechas por el modelo kNN's para vecinos=1. [16].....	13
Figura 2.9.	Lógica tras el perceptrón multicapa.[16].....	14
Figura 2.10.	Ejemplo de un perceptrón multicapa con una sola capa oculta. [16].....	14
Figura 3.1.	Diagrama de las actividades iniciales que comprenden la fase de desarrollo de un producto. [21] 15	15
Figura 4.1.	Diagrama de caja negra del problema del proyecto. Elaboración propia. ....	22
Figura 4.2.	Diagrama de refinamiento que muestra subfunciones. Elaboración propia. ....	23
Figura 4.3.	Diagrama de flujo del desarrollo del proyecto. Elaboración propia. ....	23
Figura 4.4.	Diagramas de árbol para los subproblemas del brazo robótico. Elaboración propia. 24	24
Figura 4.5.	Diagrama de árbol de los subproblemas de los sensores. Elaboración propia. ....	34
Figura 4.6.	Diagrama de árbol para la inteligencia artificial. Elaboración propia .....	38
Figura 5.1.	Entorno del robot y la visión de la cámara. Elaboración propia. ....	43
Figura 5.2.	Gráfica de rangos de movimiento. La posición ideal que considera el umbral es la zona naranja. El rango para hacer pequeños movimientos es la zona verde. El resto del rango se hacen grandes movimientos. Elaboración propia. ....	45
Figura 5.3.	Rangos de acción para el robot según la posición de la pieza objetivo. Elaboración propia. 46	46
Figura 5.4.	Ejemplo de trayectoria de movimiento del robot para llegar a la pieza a base de casos definidos. Elaboración propia. ....	47
Figura 5.5.	Diagrama de flujo para el código de la base de datos. Elaboración propia .....	49
Figura 5.6.	Ejemplo de movimiento para la base de datos, asumiendo dos posiciones nuevas. a) Ejemplo de resultado del código P0. b) Ejemplo de resultado del código Pi. c) Ejemplo de resultado del código "Selección de Pi". d) Ejemplo 2 de resultado del código Pi. Elaboración propia.	

Figura 5.7.	Ejemplo de trayectoria y posibles posiciones aleatorias que realiza la base de datos. Elaboración propia. ....	50
Figura 7.1.	Curva de aprendizaje para el modelo de árbol de decisión. ....	61
Figura 7.2.	Curva de aprendizaje para el modelo de árbol de decisión multisalida. ....	62
Figura 7.3.	Curva de aprendizaje 1: Regresión polinomial. ....	62
Figura 7.4.	Curva de aprendizaje 2: Regresión polinomial (con acercamiento). ....	63
Figura 7.5.	Curva de aprendizaje: kNN's. ....	63
Figura 7.6.	Curva de aprendizaje: Red neuronal. ....	64
Figura 7.7.	Curva de aprendizaje 2: Red neuronal (acercamiento). ....	64
Figura 7.8.	Predicciones vs Datos reales: Árbol de decisión. ....	65
Figura 7.9.	Predicciones vs Datos Reales: Árbol de decisión multisalida. ....	65
Figura 7.10.	Predicciones vs Valores Reales: Regresión Polinomial. ....	66
Figura 7.11.	Predicciones vs Valores reales: kNN's. ....	66
Figura 7.12.	Predicciones vs Valores reales: Red Neuronal. ....	67
Figura 7.13.	Gráfico de residuos: Árbol de decisión. ....	68
Figura 7.14.	Gráfico de residuos: Árbol de decisión multisalida. ....	68
Figura 7.15.	Gráfico de residuos: Regresión Polinomial. ....	69
Figura 7.16.	Gráfico de residuos: kNN's. ....	69
Figura 7.17.	Gráfico de residuos: red neuronal. ....	70
Figura 7.18.	Gráfica Q-Q: Árbol de decisión. ....	71
Figura 7.19.	Gráfica Q-Q: Árbol de decisión multisalida. ....	71
Figura 7.20.	Gráfica Q-Q: Regresión Polinomial. ....	72
Figura 7.21.	Gráfica Q-Q: kNN's. ....	72
Figura 7.22.	Gráfica de Q-Q: red neuronal. ....	73
Figura 7.23.	Errores de distancia: Árbol de decisión. ....	74
Figura 7.24.	Errores de distancia: Árbol de decisión multisalida. ....	74
Figura 7.25.	Errores de distancia: Regresión polinomial. ....	75
Figura 7.26.	Errores de distancia: kNN's. ....	75
Figura 7.27.	Errores de distancia: red neuronal. ....	76

## **Contenido de tablas**

Tabla 4.1.	Clasificación de necesidades para el proyecto. Donde va de numeración de 5 (definitivamente la solución debe contar con esto) a 1 (no deseable en la solución). .....	19
Tabla 4.2.	Matriz de necesidades-métricas del proyecto. ....	20
Tabla 4.3.	Lista comparativa de brazos robóticos: Características físicas .....	25
Tabla 4.4.	Lista comparativa de brazos robóticos: Efector y sensor.....	26
Tabla 4.5.	Lista comparativa de brazos robóticos: Software.....	27
Tabla 4.6.	Lista comparativa de brazos robóticos: Precios.....	29
Tabla 4.7.	Posibles conceptos por elegir para el proyecto. ....	31
Tabla 4.8.	Matriz de selección para los conceptos del brazo robótico. ....	32
Tabla 4.9.	Matriz de evaluación para los conceptos del brazo robótico .....	33
Tabla 4.10.	Matriz de selección de conceptos para los sensores.....	35
Tabla 4.11.	Matriz de selección de conceptos para las cámaras.....	36
Tabla 4.12.	Matriz de selección de conceptos para el modelo de base de datos. ....	38
Tabla 4.13.	Matriz de selección de conceptos para los modelos de IA supervisados. ....	39
Tabla 4.14.	Matriz de selección de conceptos: Modelo de Inteligencia artificial.....	40
Tabla 6.1.	Plan de lecciones propuestas para el proyecto. ....	55
Tabla 7.1.	Matriz de resultados de cada modelo de Inteligencia Artificial. ....	76
Tabla 7.2.	Resultados obtenidos para los modelos de Inteligencia Artificial usados.....	77
Tabla 7.3.	Parte de los datos obtenidos para la trayectoria del robot. ....	79
Tabla 7.4.	Tabla resumen de evaluación a la población muestra .....	82
Tabla 8.1.	Resumen de análisis económico del proyecto (Costos e inversiones) .....	85
Tabla 8.2.	Resumen de análisis económico del proyecto (Beneficios). ....	86

# 1. Introducción

## 1.1 Entorno del proyecto

Este proyecto se lleva a cabo con el Grupo Integrado de Ingeniería (GII). Este es un grupo de investigación aplicada en ingeniería orientado a generar conocimiento científico y la creación de nuevos productos en el entorno industrial. Este grupo se creó en 1999 en el Campus de Ferrol de la Universidad de Coruña (UDC), España. Tiene diversas líneas para la investigación científica, como lo son la Ingeniería Naval, Dinámica de Fluidos, Inteligencia Computacional, Robótica, Sistemas de Medida y Sensorización y Organización Industrial. [1]

Además, el proyecto del GII se integra con el proyecto europeo AIM@VET, el cual busca implementar módulos de aprendizaje para la educación VET (educación y entrenamiento vocacional, por sus siglas en inglés) con una perspectiva del profesor. Aplicando tres áreas clave para el aprendizaje: visión por computadora, robótica e inteligencia ambiental. [2]

El proyecto por realizar se centra en las áreas de Robótica e Inteligencia Artificial, las cuales son las líneas de enfoque que poseen tanto el GII como AIM@VET. El enfoque buscado es la integración de un brazo robótico en los módulos de enseñanza de estudiantes preuniversitarios. Inicialmente poseía cuatro módulos completos con un quinto y sexto módulo en desarrollo durante la creación de este proyecto. Esas primeras lecciones se realizaron utilizando un robot móvil denominado “Robobo”.

En cuanto al brazo robótico y el simulador a utilizar se definen durante la realización de este proyecto.

## 1.2 Definición del problema

### 1.2.1 Generalidades

El proyecto se desarrolla en el Grupo Integrado de Ingeniería (GII) en el Campus de Ferrol de la Universidad de Coruña (UDC). El GII es un grupo

interdisciplinario de ingenieros que desarrollan investigación en diversas áreas, como lo es la robótica y la inteligencia artificial. Dentro de los proyectos que se desarrollan actualmente es ampliar los módulos de enseñanza que cuenta el proyecto AIM@VET, donde actualmente utilizan el robot llamado “Robobo”.

Actualmente el robot existente consiste en una base con ruedas al que se le acopla un celular en un atril, el cual funciona como el cerebro y cara del robot. La base cuenta con 5 sensores de proximidad ubicados en la parte frontal de la base los cuales poseen diferentes orientaciones y tres en la parte trasera. Con estos sensores se puede detectar la presencia de objetos y la continuidad de la superficie donde este se desplaza. Por lo que ahora se busca la implementación de un brazo robótico fijo como parte del proyecto para que los estudiantes también aprendan con otro producto.

### **1.3 Síntesis del problema**

Necesidad del GII de agregar nuevos módulos lectivos con la implementación de un brazo robótico y utilizar Inteligencia Artificial para configurar algunas acciones que pueda realizar.

### **1.4 Objetivo General**

Desarrollar lecciones de Inteligencia Artificial mediante el uso de un brazo robótico en un currículum preuniversitario.

### **1.5 Objetivos Específicos**

- 1) Contrastar las necesidades del cliente con posibles opciones de brazo robótico en el mercado para su implementación en el programa AIM@VET para estudiantes preuniversitarios
- 2) Desarrollar posibles objetivos de aprendizaje de Inteligencia Artificial para un plan de estudiantes preuniversitarios mediante el uso de un brazo robótico.

- 3) Implementar las lecciones de aprendizaje desarrolladas mediante el uso de simuladores.
- 4) Validar el correcto funcionamiento de las lecciones de aprendizaje mediante el uso del robot físico.

## 2. Marco Teórico

### 2.1 Robótica

La industria de la automatización se inventó para que las máquinas pudieran realizar los trabajos más duros, repetitivos o dañinos que nos podemos encontrar. Por ejemplo, a la hora de llenar botellas de refresco, la industria se orienta hacia máquinas automatizadas que, cuando una botella llega a una determinada posición, abren las tuberías para llenar la botella y, tras unos segundos, el flujo se detiene hasta que llega la siguiente botella. Sin embargo, las máquinas automatizadas se utilizan cuando la tarea a resolver requiere una mínima necesidad de toma de decisiones o de adaptabilidad. Pero ¿qué ocurre cuando hay un trabajo que tiene tareas un poco más complejas que las que puede realizar un sistema automatizado? En esos casos, los robots pueden ser nuestra mejor respuesta.

#### 2.1.1 Anatomía de los robots

La anatomía del robot se refiere a la construcción física de la máquina. Veremos tres partes importantes para esta unidad: el robot, la base (en la que reposa el robot) y, en el extremo del robot, el efector final (que tiene la herramienta que realizará la tarea deseada). Dentro del robot, hay articulaciones que unifican toda la estructura y le permiten moverse de varias formas. [14]

Existen diferentes tipos de movimientos articulares que se pueden observar en la Figura 2.1.

- ❖ Lineal: Este tipo de articulación permite el movimiento de deslizamiento o traslación entre las partes conectadas. (a).
- ❖ Rotacional: Permite la rotación entre un eje fijo. (b).
- ❖ Giratoria: Esta tiene un movimiento circular alrededor de un eje en el robot. (c).
- ❖ Torsión: Implica rotación a lo largo de un eje longitudinal. (d)

- ❖ Cilíndrica: Es una combinación entre movimientos giratorios y lineales. (f)
- ❖ esférico: permite que los eslabones se muevan en los tres ejes. (g)

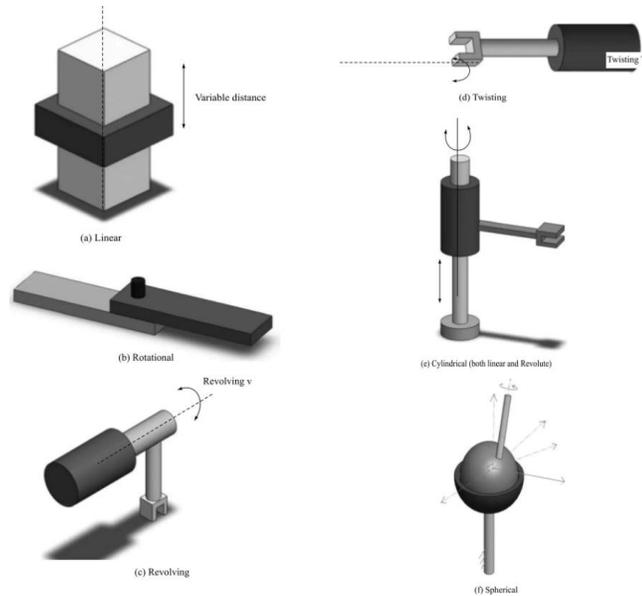


Figura 2.1. Tipos de articulaciones. Referencia: [14]

También hay cuatro configuraciones de robot comunes que utilizan el movimiento de las articulaciones que se ve en Figura 2.2. [14]

- ❖ Configuración polar. (a)
- ❖ Configuración cilíndrica. (b)
- ❖ Configuración cartesiana. (c)
- ❖ Configuración de brazo articulado o manipulador. (d)

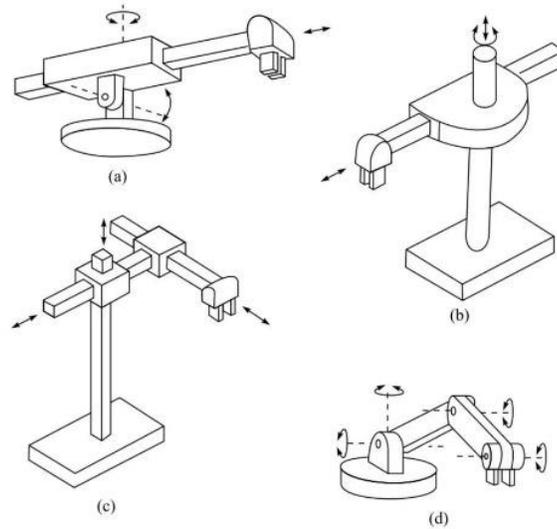


Figura 2.2. Anatomías del robot. Referencia [14]

Cada movimiento de una articulación individual está asociado al término grados de libertad (DOF), que significa de cuántas maneras puede moverse una articulación. Por ejemplo: tanto los movimientos lineales como los giratorios tienen 1 DOF, los cilíndricos tienen 2 DOF y los esféricos tienen 3 DOF. También podemos clasificar los robots por DOF y, en el caso de los robots industriales, es típico tener de 4 a 6 DOF. En la Figura 2.3 se muestra un ejemplo de un manipulador de 6 grados de libertad.

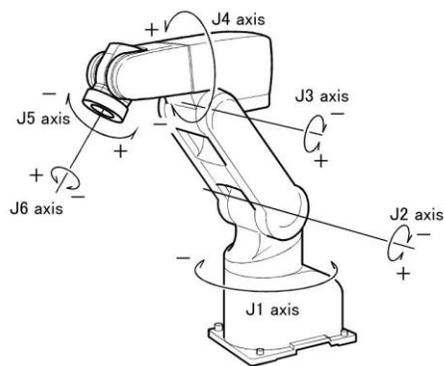


Figura 2.3. Manipulador de 6 DOF. [14]

### 2.1.2 Espacio de trabajo

Los grados de libertad y el tipo de movimiento de la articulación también contribuyen al espacio de trabajo, que se refiere al espacio dentro del extremo del robot que puede alcanzar sin el efector final. El efector final no debe considerarse, ya que es un agregado al robot y puede variar en diferentes tamaños. En la Figura 2.4 se puede ver algunas de las cargas de trabajo comunes que pueden tener las cuatro anatomías comunes. En (a) corresponde a un robot polar y su volumen de trabajo parece una esfera parcial. En (b) se ve un robot cilíndrico tiene un volumen de trabajo de un cilindro (de ahí el nombre). Finalmente, (c) con un robot cartesiano forma un prisma rectangular.

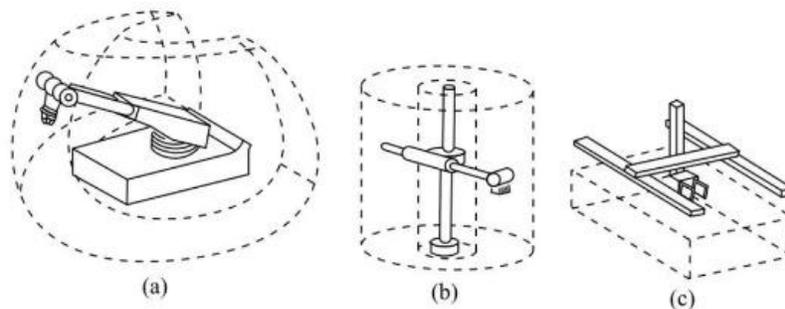


Figura 2.4. Área de trabajo de los robots. [14]

## 2.2 Sistemas de Visión

Los sistemas de visión forman una especialidad por sí mismos, con varios conceptos, subtemas, algoritmos matemáticos que sirven para poder analizar las imágenes y videos que toman los sensores de visión. Sirve para convertir en valores o especificar lo que nosotros podemos reconocer. Con el análisis y la conversión, hace que nos podemos comunicar con un computador sobre lo que estamos buscando y lo que se quiere realizar.

Como se mencionaba, sistemas de visión es un tema amplio, por lo que se visitarán solo los conceptos que fueron fundamentales para la ejecución del proyecto y que se les transmitirá a los estudiantes con las lecciones creadas.

### 2.2.1 La cámara

El primer componente de nuestro sistema de visión será el sensor de la misma categoría. Cualquier tipo de cámara puede utilizarse. Tanto imágenes como videos pueden ser procesados.

Esta funciona como los ojos de un ser vivo, con ella se puede analizar el entorno y digitalizar los datos para que la computadora sea capaz de procesar la información. Con esta información, la computadora puede actuar según lo que encuentre.

Al igual que los ojos de los seres vivos, como águilas, topos y caballos; existen diferentes características para las cámaras. Algunas pueden ver más largo, otras necesitan una menor distancia, algunas tienen mayor ángulo de visión, otras uno reducido, entre otras características. Estas características son capaces de definir una cámara y con ellas uno puede basar la compra según lo requiera una aplicación específica.

Dentro de las características principales que se utilizan, se encuentran:

- Longitud focal: Define la distancia entre el centro de proyección de la cámara y el plano de la imagen.
- Campo de visión: Es el área observable que la cámara puede capturar.
- Máxima longitud: Es la distancia máxima de trabajo.
- Tamaño de cámara y/o de imagen: Es el tamaño de píxeles que se pueden capturar o almacenan.

### 2.2.2 Segmentación: Espacios de color y máscaras de color

#### **Espacios de color**

Los colores son una percepción visual que surge de la forma en que nuestros ojos y cerebro interpretan las ondas de luz. Podemos ver e interpretar diferentes longitudes de onda y determinar qué color estamos viendo.

Existen modelos matemáticos para describir a los colores por medio de valores. Hay algunos espacios de color comunes:

- 🌈 RGB (rojo, verde y azul): en estos modelos, cada color se describe mediante la combinación de luz roja, verde y azul. El negro será todos

los valores como 0, y el blanco todos los valores como el número máximo en la escala respectiva.

- ✚ CMYK (Cyan, Magenta, Yellow, Key/Black): al igual que el último, cada color se crea por la combinación de los colores en el nombre. Probablemente hayas visto este modelo en impresoras con el color de tinta que necesita para imprimir en el papel.
- ✚ HSV (Hue, Saturation, Value): en este modelo, se describen en términos de su tono (que representa el tipo de color, la longitud de onda dominante), saturación (que indica la pureza del color) y valor (también conocido como brillo).
- ✚ Escala de grises: representa imágenes en tonos de gris sin ninguna información de color, cada píxel varía dependiendo de la intensidad de la luz, desde el negro hasta el blanco.

En la Figura 2.5 se puede ver cómo cambiando cada variable se obtiene un color diferente.

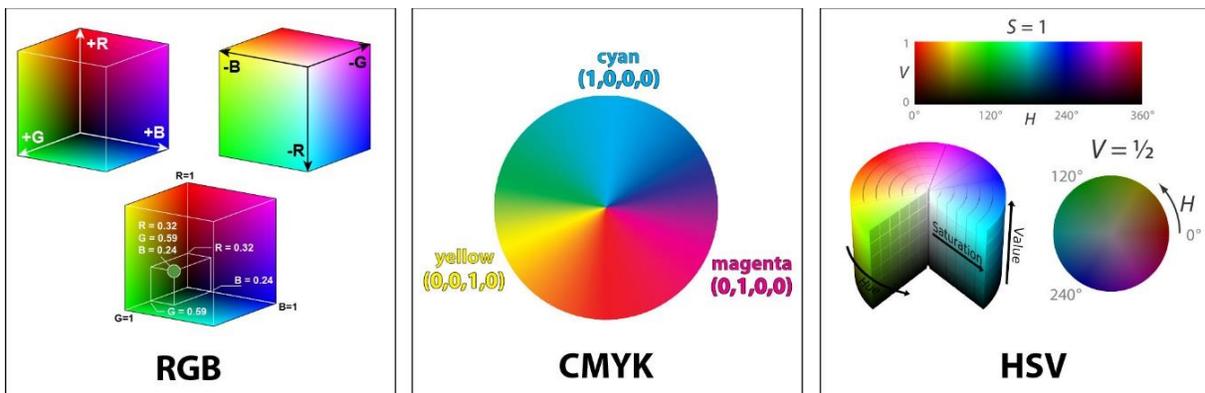


Figura 2.5. Espacios de color. [15]

### **Máscaras de color**

Como aprendimos en la sección anterior, los colores no son solo un valor específico, sino un rango de valores. Podemos tomar una manzana, una fresa y una cereza y clasificarlas como rojas, pero no son el mismo tipo de rojo. Lo mismo con cada color que vemos, puede tener muchas variaciones dependiendo de cómo se componga. Cuando usamos los espacios de color, convirtiendo cada color en valores numéricos, podemos tomar colores que a simple vista parecen iguales, y ver la diferencia entre ellos por los números que muestran. Lo mismo ocurre al revés, si queremos que la computadora identifique todo el color que identificamos como rojo, necesita los valores respectivos o un rango para que esta sepa exactamente lo que queremos identificar.

Con las máscaras de color, lo que se logra es que digitalmente se separan las imágenes según el color de los píxeles. Para esto se requiere dar un rango de los valores en HSV para determinar los píxeles que quedan en la máscara.

#### **2.2.3 Binarización de imágenes**

Tal como indica el nombre, cambia los valores de una imagen en dos valores. La binarización específicamente cambia la imagen a una donde solo hay dos valores en la escala de grises, píxeles negros y blancos.

Esto es útil en sistemas de visión porque de esta forma se pueden definir dos áreas: la de interés y la que no se va a analizar, permitiendo que el proceso sea más fácil.

## **2.3 Inteligencia Artificial**

La Inteligencia Artificial (IA) busca que las computadoras puedan tomar decisiones como la mente humana puede.

### **2.3.1 Tipos de aprendizaje en Inteligencia Artificial**

La Inteligencia Artificial tiene tres grandes clases. Aprendizaje supervisado, sin supervisar y por refuerzo. [9]

### **Supervisado:**

En el aprendizaje supervisado, el programador entrena un sistema al definir las salidas deseadas para el conjunto de entradas y brindar retroalimentación constante para definir si se llegó a la meta. Esto implica tener ejemplos clasificados con los cuales entrenar la Inteligencia Artificial. [9]

### **Sin supervisor:**

En este tipo de aprendizaje, el programador no le entrega a la Inteligencia Artificial la salida deseada ni mensajes de error. Este tipo de aprendizaje puede ser implementado en aplicaciones donde se quiere descubrir conocimiento, de modo que los programadores no necesitan conocer qué patrones existen en los datos y la IA los encuentra por sí sola. [9]

### **Aprendizaje por refuerzo:**

Este tipo de aprendizaje funciona con recompensas y castigos que la misma retroalimentación les muestra si la acción que acababan de hacer es buena o mala. Estos mensajes de retroalimentación no tienen que ser binarios, pero pueden estar representados con números. [9]

## 2.3.2 Base de datos

Al enseñarle a los bebés y niños a hablar y conocer objetos, los adultos les enseñan tarjetas con imágenes representativas para explicar la palabra que se busca enseñar. Al igual que la enseñanza con los niños, la Inteligencia Artificial requiere de información y datos de los que pueda aprender. Para esto se requiere un conjunto de datos, lo más variado posible para que la IA pueda desarrollarse en el mayor caso de escenarios posibles.

Según los tipos de aprendizaje mencionados, la Inteligencia Artificial requiere de datos deseables de salida (clasificados o no) y/o mensajes de retroalimentación para poder aprender de su entorno y llegar a tener las acciones deseadas cuando se necesiten. De acuerdo con el tipo de aprendizaje seleccionado para la implementación de una IA, así se deberá escoger su base de datos. [9]

### 2.3.3 Modelos de entrenamiento supervisado

#### **Árboles de decisión**

Los árboles de decisión son modelos altamente usados para aplicaciones de clasificación y regresión. En este modelo se aprenden la jerarquía de preguntas if/else que llevan a una decisión. Este tipo de preguntas se pueden expresar como árboles de decisión, donde cada nodo representa una pregunta, las líneas que conectan las preguntas contienen respuestas a la pregunta previa y los nodos terminales son llamados hojas y tienen una solución. [16]

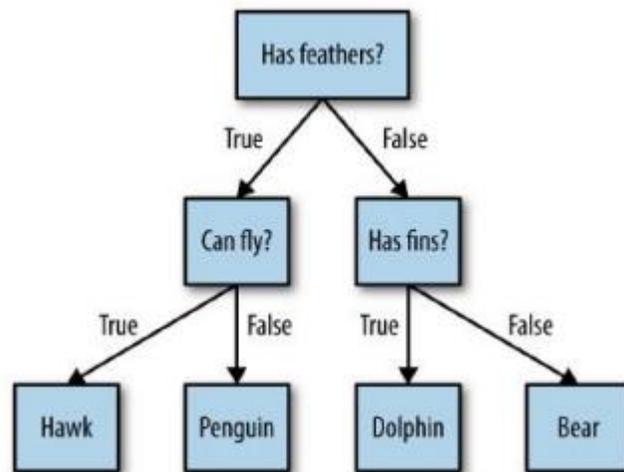


Figura 2.6. Lógica de los árboles de decisión [16]

#### **Regresión Lineal y Polinomial**

La regresión lineal es modelo que asume una relación lineal entre la salida y las entradas. La fórmula de un modelo de este estilo es:

$$y = b_0 + b_1x \quad (1)$$

Para la salida  $y$ , entrada  $x$ ,  $b_0$  es el término constante y  $b_1$  la pendiente de la línea.

Una forma de enriquecer la representación de características de los modelos lineales es agregando características polinómicas de los datos originales. De esta forma se pueden capturar las relaciones no lineales en la aplicación y permite que la curva de ajuste tenga formas menos simples como la lineal. [16]

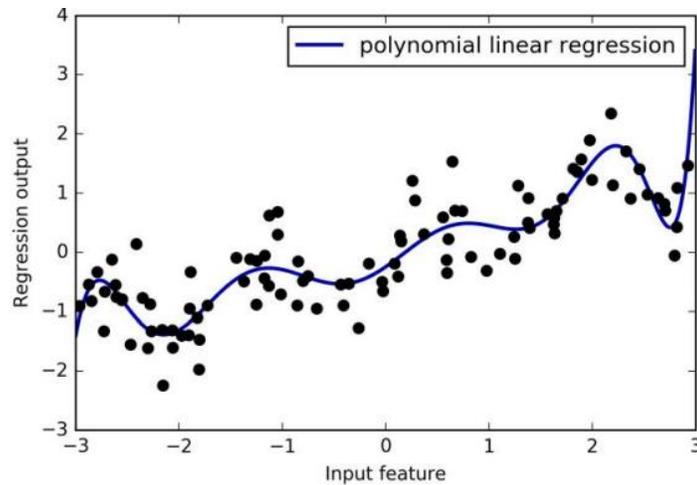


Figura 2.7. Ejemplo de regresión lineal con características polinómicas. [16]

### kNN's

El algoritmo de este modelo de IA tiene un dato nuevo y busca los datos más cercanos en la base de datos para hacer una predicción, y selecciona la salida del vecino cercano seleccionado. [16]

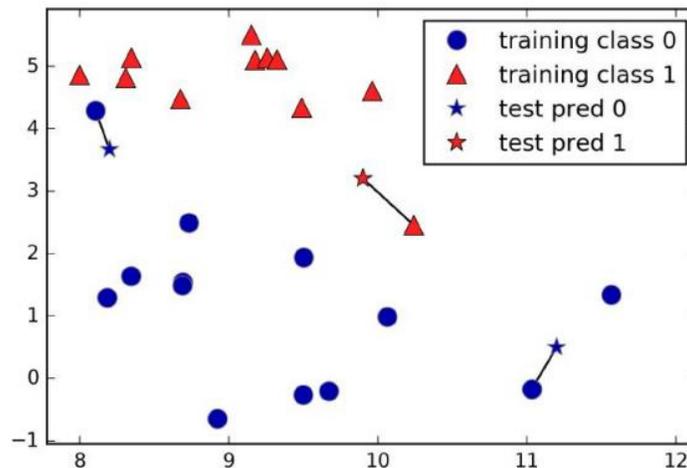


Figura 2.8. Predicciones hechas por el modelo kNN's para vecinos=1. [16]

### Red Neuronal

Las redes neuronales pueden ser un tema complejo dependiendo de su aplicación, por lo que solo se valorará los métodos simples llamados Perceptrones Multicapa (MLP, Multi-layered perceptron) para clasificación y regresión.

Este modelo puede ser generalizado en modelos lineales que realizan múltiples etapas para llegar a tomar una decisión.

En la Figura 2.9 cada nodo a la izquierda representa las entradas de la capa, las líneas representan los coeficientes aprendidos y el nodo de la derecha representa la salida. [16]

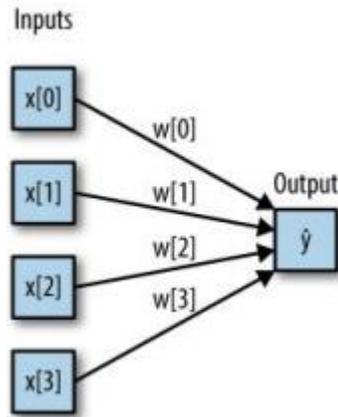


Figura 2.9. Lógica tras el perceptrón multicapa.[16]

En un MLP este proceso es repetido múltiples veces, las cuales crean capas ocultas como en la Figura 2.10 donde se tiene más coeficientes para aprender.

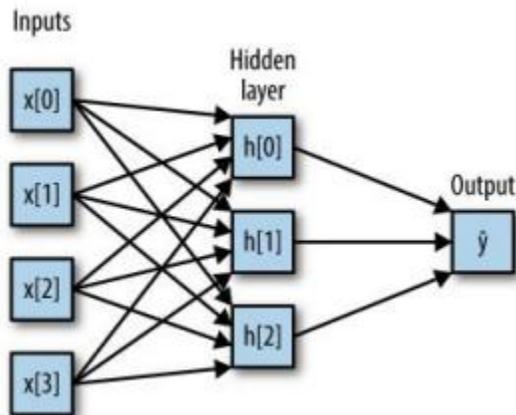


Figura 2.10. Ejemplo de un perceptrón multicapa con una sola capa oculta. [16]

### 3. Descripción de la metodología

Este proyecto se desarrolló utilizando la metodología de diseño de ingeniería planteada en Ulrich y Eppinger [21]. Con dicha metodología se implementaron los pasos a seguir para el desarrollo de proyectos los cuales se pueden a ver a detalle en la Figura 3.1.

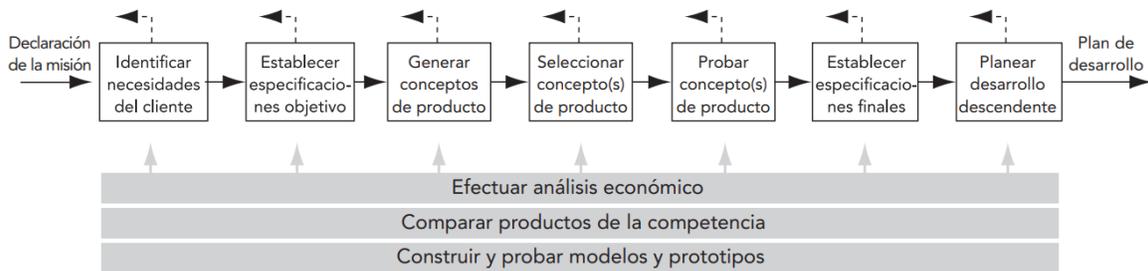


Figura 3.1. Diagrama de las actividades iniciales que comprenden la fase de desarrollo de un producto. [21]

#### 1.1 Etapa 1: Identificación y determinación de necesidades

Como primer paso en la solución del proyecto, es la comunicación con el cliente para identificar y determinar el conjunto de necesidades que se tienen relacionadas al problema establecido.

Tras la identificación de las necesidades, se realiza una jerarquización de estas en primarias y secundarias. Con el fin de encontrar las ideas principales y las que varían de estas.

Después se vuelve a tener comunicación con el cliente para que este analice las necesidades y les dé un orden de prioridad.

Es importante recalcar, que durante este proceso aún no se buscan soluciones al problema, hay que mantener un espacio mental libre de búsqueda de soluciones para evitar sesgos más adelante.

#### 3.2 Etapa 2: Establecimiento de especificaciones objetivo

Con las necesidades en un orden establecido, se requiere establecer las especificaciones objetivo. Estas especificaciones se podrán evaluar los conceptos

propuestos más adelante. En esta etapa también es importante estar fuera de sesgos para la solución.

En esta etapa se puede crear la matriz de necesidades-métricas para verificar la correspondencia entre ambas, con el fin de encontrar la dependencia entre estas.

Por último, se realiza un breve estudio de cada característica correspondiente a las métricas y se definen valores objetivo para cada una.

### **3.3 Etapa 3: Generación de conceptos**

Esta etapa comienza con la descomposición del problema. El fin es encontrar los subproblemas relacionados para encontrar un punto de partida en la definición de conceptos. En esta etapa es importante obtener una lista exhaustiva y completa, ya que de estos conceptos se basará la solución, por lo mismo no se debería implementar ningún tipo de filtro.

Además, para la creación de conceptos es importante hacer una búsqueda interna por experiencia personal; pero también es importante complementar con una búsqueda externa donde se usan fuentes de información.

### **3.4 Etapa 4: Selección de conceptos**

Con los conceptos planteados, se procede a evaluarlos para verificar su utilidad en nuestra solución. Primero se realiza un filtrado, para determinar los conceptos que requieran mejoras, deban ser combinados o rechazados. Al tener una lista de conceptos más reducidas con solo propuestas que serán de utilidad para la solución, se pasa al segundo paso, donde se selecciona un concepto ganador del cual partirá nuestra solución.

Esta metodología implementa la iteración para poder conseguir los mejores resultados posibles. Durante cualquiera de las etapas es posible encontrarse con la situación que hay que regresar y añadir o modificar algo. Es parte del proceso para lograr llegar a la mejor solución posible.

### **3.5 Etapa 5: Pruebas de concepto**

En esta etapa se prueban los conceptos ganadores para verificar que las necesidades del cliente se han satisfecho. Además, se localiza cualquier error o

defecto que debe ser corregido posteriormente. También se busca la opinión del cliente, si esta no es satisfactoria el desarrollo del proyecto puede terminar o volver a los pasos anteriores para repetirlos.

### **3.6 Etapa 6: Documentación del proyecto**

En esta última etapa, se redacta el proceso realizado, junto con los datos y decisiones de las etapas previas. Además, se realiza un análisis económico, resultados obtenidos, conclusiones, recomendaciones y toda información relevante usada para el desarrollo del proyecto.

### **3.7 Subetapa: Modelado y construcción de prototipos**

Si bien esta no se expresa como una etapa fundamental en la línea de desarrollo de proyectos, a veces es crear modelos iniciales para la etapa 6 (“Pruebas de concepto”). Estos se crean con el fin de demostrar la factibilidad del concepto ganador de la etapa anterior.

Las posibles formas de desarrollar esta etapa son:

- ❖ Modelo previo a la Prueba de concepto
- ❖ Modelo de forma de la solución
- ❖ Modelo de hoja de cálculo de concesiones técnicas
- ❖ Modelos experimentales de prueba

## 4. Desarrollo

Este proyecto se desarrolló utilizando la metodología de diseño de ingeniería explicada en el capítulo pasado.

En el Capítulo 3: Metodología, se hablaron de 6 etapas. Durante el desarrollo del proyecto se notó que este contaba con tres áreas que dependían de la finalización de la anterior y para cada una se aplicó la metodología de diseño. Por lo que este capítulo se dividirá en 5 secciones. La primera consta de las primeras etapas del proyecto, hasta donde se dividen las áreas. Las siguientes tres secciones se enfocarán en cada área y el desarrollo de la metodología hasta la quinta etapa. La última sección trata de la etapa final con todas las áreas unificadas.

Como una etapa previa, se realizó una reunión con el cliente para conocer y definir el problema y conocer lo que se había hecho previamente. Se mencionaron las primeras lecciones que se realizaron con el carro robótico: Robobo. Deseando continuar con el curso con un brazo robótico para las siguientes lecciones.

### 4.1 Primera Etapa: Definición de necesidades

En la primera etapa, se realizaron una serie de reuniones y entrevistas con el cliente para identificar y determinar el conjunto de necesidades que ellos encontraban con el problema planteado.

Después, por cuenta propia se identificaron de las necesidades y se realizó una jerarquización de estas dividiéndolas en primarias y secundarias. Las necesidades jerarquizadas son:

- El robot es compatible con las capacidades de los estudiantes.
  - ❖ El robot es programable.
  - ❖ El robot se tiene que poder programar con un lenguaje conocido para los estudiantes.
  - ❖ El robot tiene librerías públicas (open source).
  - ❖ El robot es apropiado para que el estudiante lo mueva a su área de trabajo.
- El robot tiene un simulador compatible.
  - ❖ El simulador tiene un nivel de programación conocido para los estudiantes.
  - ❖ El simulador posee las características físicas del robot.
  - ❖ El simulador es versión 3D.

- El robot tiene características atractivas para los estudiantes.
  - ❖ El robot tiene un efector compatible para agarrar objetos.
  - ❖ El robot se asemeja a los robots industriales.
  - ❖ El robot tiene que estar disponible actualmente en el mercado.
  - ❖ El robot puede usar cinemática inversa.
- Al robot se le pueden agregar sensores y cámaras.
  - ❖ La cámara debe tener acceso a los datos sin aplicación propia.
  - ❖ La cámara se puede usar en el simulador.
- El robot se mantiene dentro del presupuesto.
- La solución se puede implementar este semestre en un colegio.
- El simulador se usa solo en Linux y Ubuntu.

Una vez necesidades fueron jerarquizadas, se contactó al cliente para obtener su retroalimentación en cuanto a las prioridades de las necesidades y asegurarse que las necesidades encontradas, efectivamente eran lo que el cliente buscaba. En otras palabras, se realizó la clasificación de las necesidades. En la Tabla 4.1. se puede observar las necesidades encontradas y su número de clasificación.

Tabla 4.1. Clasificación de necesidades para el proyecto. Donde va de numeración de 5 (definitivamente la solución debe contar con esto) a 1 (no deseable en la solución).

Clasificación de necesidades	
Clas.	Necesidades
5	El robot es programable.
5	El robot se tiene que poder programar con un lenguaje conocido para los estudiantes.
4	El robot tiene librerías públicas (open source).
5	El robot es apropiado para que el estudiante lo mueva a su área de trabajo.
4	El robot tiene un efector compatible para agarrar objetos.
5	El robot es compatible con algún simulador.
4	El simulador tiene un nivel de programación conocido para los estudiantes.
3	El simulador posee las características físicas del robot.
3	El simulador es versión 3D.
4	Al robot se le puede agregar sensores y cámaras.
4	La cámara debe tener acceso a los datos sin aplicación propia.
4	La cámara se puede usar en el simulador.
3	El robot se asemeja a los robots industriales.
5	El robot tiene que estar disponible actualmente en el mercado.
2	El robot puede usar cinemática inversa.
4	El robot se mantiene dentro del presupuesto.
5	La solución se puede implementar este semestre en un colegio.
1	El simulador se usa solo en Linux y Ubuntu.

Al tener los resultados del cliente, se pudo notar las necesidades que el cliente no deseaba, las que tenían baja prioridad y las que sí eran correctas y con alta prioridad.

#### 4.2 Segunda Etapa: Establecimiento de especificaciones

Con las necesidades indeseables descartadas, se pasó a la segunda etapa donde se elaboró una lista de métricas relacionadas a las necesidades y se creó la matriz de necesidades-métricas para verificar la correspondencia entre ambas. En la Tabla 4.2 se puede analizar la matriz mencionada.

Tabla 4.2. Matriz de necesidades-métricas del proyecto.

Matriz de necesidades-métricas														
	Lenguajes de progra. compatibles.	Librerías Open Source.	Simuladores compatibles.	Peso del robot.	Tamaño del robot.	.Costo del robot.	Accesibilidad a los datos de sensores.	Material del robot.	Geometría del robot.	Disponibilidad actual en tiendas.	Eefector disponible.	Tiempo de envío del robot.	Tiempo de desarrollo de las lecciones.	Peso máximo que soporta el robot.
El robot es programable.	X													
El robot se tiene que poder programar con un lenguaje conocido para los estudiantes.	X													
El robot tiene librerías públicas (open source).		X												
El robot es apropiado para que el estudiante lo mueva a su área de trabajo.				X	X				X					

El robot tiene un efector compatible para agarrar objetos.											X			
El robot es compatible con algún simulador.			X											
El simulador tiene un nivel de programación conocido para los estudiantes.	X		X											
El simulador posee las características físicas del robot.			X											X
El simulador es versión 3D.			X											
Al robot se le puede agregar sensores y cámaras.			X											
La cámara debe tener acceso a los datos sin aplicación propia.						X								
La cámara se puede usar en el simulador.			X											
El robot se asemeja a los robots industriales.							X	X						
El robot tiene que estar disponible actualmente en el mercado.														
El robot puede usar cinemática inversa.														
El robot se mantiene dentro del presupuesto.					X									
La solución se puede implementar este semestre en un colegio.												X	X	

Después se realizó un breve estudio de cada característica correspondiente a las métricas y se definieron los valores objetivo.

<b><u>Valores objetivo</u></b> Peso: Val. Marginal: 15kg Val. Ideal: 10kg  Lenguaje de programación: Val.: Python  Material Val. Marginal: Plástico Val. Ideal: Metal
---

### 4.3 Tercera y cuarta etapa: Generación y selección de conceptos

Con los requerimientos definidos, se realizó una descomposición del problema. Entrando así a la tercera etapa de la metodología de Ulrich y Eppinger [21].

En la Figura 4.1 y 4.2 se muestra el diagrama de caja negra y el diagrama de refinamiento de las subfunciones del proyecto. Cada entrada, se consideró como un área a la cual se realizó su propio análisis de subproblemas. Para cada área se implementó el resto de la metodología individualmente. Por esto, a partir de aquí se continuará el desarrollo del proyecto dividido en cada área. Además, cada área fue desarrollada según el diagrama de flujo de la Figura 4.3.

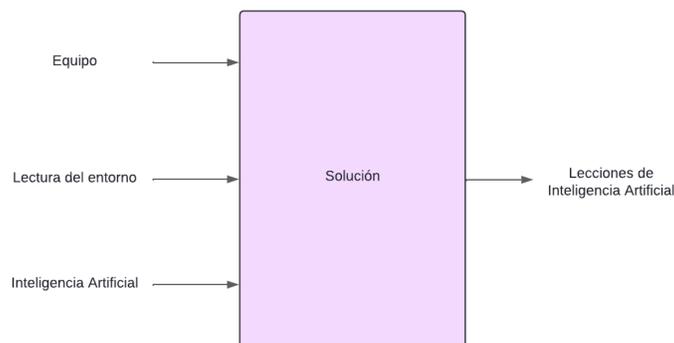


Figura 4.1. Diagrama de caja negra del problema del proyecto. Elaboración propia.

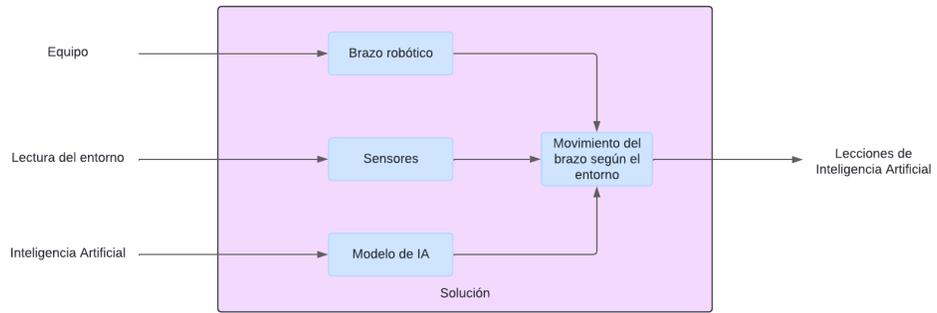


Figura 4.2. Diagrama de refinamiento que muestra subfunciones. Elaboración propia.

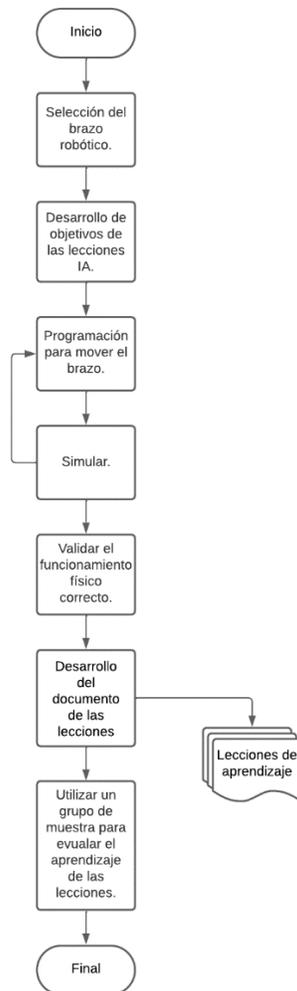


Figura 4.3. Diagrama de flujo del desarrollo del proyecto. Elaboración propia.

### 4.3.1 Brazo robótico: Generación y selección de conceptos

Como continuación de la tercera etapa, se analizan los subproblemas del área del brazo robótico. En la Figura 4.4 se representan los árboles de clasificación usados para analizar los subproblemas.

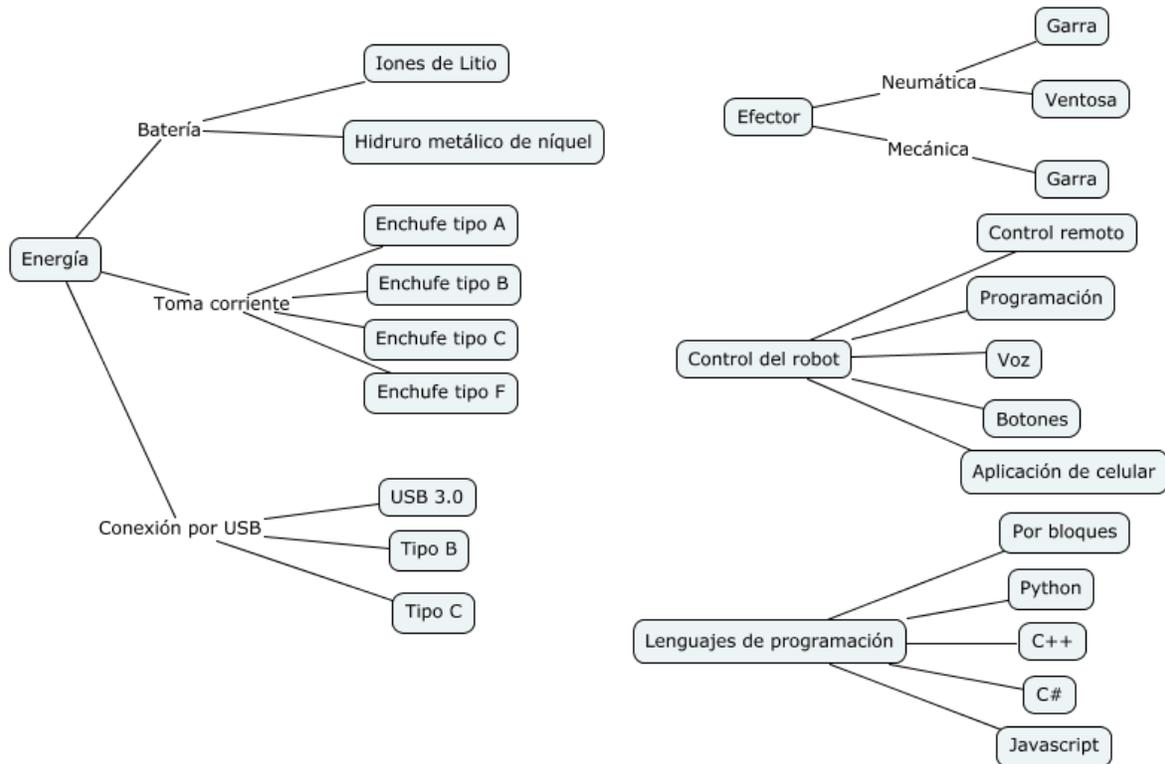


Figura 4.4. Diagramas de árbol para los subproblemas del brazo robótico. Elaboración propia.

Con los subproblemas que se plantearon y sus variables, se analizaron los posibles conceptos que se encontraban en el mercado. En las siguientes tablas (de la Tabla 4.3 a la 4.6, se encuentran las tablas comparativas del documento entregable del listado de los robots encontrados en el mercado junto su precio, diferentes posibilidades de efector que permiten el agarre de objetos, tamaño del robot, número de grados de libertad, si posee sensores o cámara, simuladores compatibles y si posee librerías u otro tipo de soporte.

Tabla 4.3. Lista comparativa de brazos robóticos: Características físicas

<b>Tabla comparativa de brazos robóticos 1</b>					
<b>Subject/Item</b>	<b>Tipo de objeto</b>	<b>Característica dif.</b>	<b>Tamaño (cm)</b>	<b>Radio de área de trabajo (cm)</b>	<b>Peso que soporta (g)</b>
<b>MyCobot 280 Arduino</b>	Brazo	USB, WIFI, Bluetooth	35*11	28	250
<b>MechArm M5Stack</b>	Brazo	Parecido a los brazos industriales. Bluetooth, USB, cable C		27	250
<b>MyCobot 280 M5Stack</b>	Brazo	USB, WIFI, Bluetooth	35*11	28	250
<b>MyCobot 280 Raspberry Pi</b>	Brazo	USB, HDMI	35*11	28	250
<b>MyPalletizer 260 M5Stack</b>	Brazo	USB, HDMI	32*	26	250
<b>MyPalletizer 260 Pi</b>	Brazo	USB, HDMI	32*	26	250
<b>MechArm Pi</b>	Brazo	USB, HDMI		26	250
<b>Brazo Robótico Adept</b>	Brazo	Con pantalla OLED			
<b>RoArm-M1</b>	Brazo	Toma de US. Wi-Fi y Bluetooth	16,33*28,46	40	
<b>Brazo robótico Braccio</b>	Brazo		52*14	32	150
<b>Brazo robot de madera</b>	Brazo	Micro USB o baterías			
<b>Brazo robótico AL5A</b>	Brazo				

<b>Brazo robótico Arduino uno R3</b>	Brazo	Baterías			
<b>Brazo robótico AL5D</b>	Brazo	Sin soporte			
<b>RaspoArm-S</b>	Brazo	Baterías 18650	35*18,3	30	
<b>Brazo robótico AL5D PLTW</b>	Brazo	ensamblado			
<b>ArmPi mini</b>	Brazo		30,9*20,8	21,5	
<b>ArmPi mini</b>	Brazo	No disponible por el momento.	30,9*20,8	21,5	
<b>ArmPi Pro</b>	Brazo	No disponible por el momento Con carro.			

Tabla 4.4. Lista comparativa de brazos robóticos: Efecto y sensor

<b>Tabla comparativa de brazos robóticos 2</b>				
<b>Subject/Item</b>	<b>Efecto incluido</b>	<b>Sensores</b>	<b>Cámara incluida</b>	<b>Grados de libertad</b>
<b>MyCobot 280 Arduino</b>	No	Posibilidad de agregar	Se puede agregar	6
<b>MechArm M5Stack</b>	No		Se puede agregar	6
<b>MyCobot 280 M5Stack</b>	No	Posibilidad de agregar	Se puede agregar	6
<b>MyCobot 280 Raspberry Pi</b>	No	Posibilidad de agregar	Se puede agregar	6
<b>MyPalletizer 260 M5Stack</b>	No		Se puede agregar	4

<b>MyPalletizer 260 Pi</b>	No		Se puede agregar	4
<b>MechArm Pi</b>	No		Se puede agregar	6
<b>Brazo Robótico Adept</b>	Sí	No	No	5
<b>RoArm-M1</b>	Sí		No	5
<b>Brazo robótico Braccio</b>	Sí	No	No	3
<b>Brazo robot de madera</b>	Sí		No	5
<b>Brazo robótico AL5A</b>	Sí		No	4
<b>Brazo robótico Arduino uno R3</b>	Sí		No	5
<b>Brazo robótico AL5D</b>				4
<b>RaspoArm-S</b>	Sí		No	4
<b>Brazo robótico AL5D PLTW</b>	Sí	No	No	5
<b>ArmPi mini</b>	Sí		Sí	4
<b>ArmPi mini</b>	Sí		Sí	4
<b>ArmPi Pro</b>	Sí		Sí	4

Tabla 4.5. Lista comparativa de brazos robóticos: Software

Tabla comparativa de brazos robóticos 3			
Subject/Item	Microprocesador	Soporte	Open source

MyCobot 280 Arduino	Arduino Uno/Mega/MKR	ROS, Python, Arduino, myBlockly, C++, C#, JavaScript	Sí
MechArm M5Stack	Raspberry Pi4B	ROS, Arduino, Python, C#, C++, JavaScript myBlockly, RoboFlow	Sí
MyCobot 280 M5Stack	Arduino	ROS, ROS2, Arduino, Python, C++, C# myBlockly, RoboFlow	Sí
MyCobot 280 Raspberry Pi	Raspberry Pi	ROS, ROS2, C++, MyBlockly, Python	Sí
MyPalletizer 260 M5Stack	Arduino	Arduino, Python y ROS	Sí
MyPalletizer 260 Pi	Raspberry Pi4B	ROS, Python, myBlockly, C#, C++, JavaScript	Sí
MechArm Pi	Raspberry Pi	ROS, Python, myBlockly, C#, C++, JavaScript	Sí
Brazo Robótico Adept	Arduino Uno		Sí
RoArm-M1	ESP32	ESP32-WROOM-32	Sí
Brazo robótico Braccio		Arduino	Sí
Brazo robot de madera		Arduino, C++	
Brazo robótico AL5A			
Brazo robótico Arduino uno R3		Arduino Uno R3	Sí
Brazo robótico AL5D			
RaspoArm-S		Raspberry Pi	Sí

Brazo robótico AL5D PLTW		Solamente se puede usar en FlowArm PLTW	No
ArmPi mini	Raspberry Pi		Sí
ArmPi mini	Raspberry Pi		Sí
ArmPi Pro	Raspberry Pi		Sí

Tabla 4.6. Lista comparativa de brazos robóticos: Precios

Tabla comparativa brazos robóticos 4					
Subject/Item	Precio (€)	Precio con pinza adaptativa y una base	Precio con 2 efectores, base y cámara	Precio envío solo robot	Precio envío + Robot
MyCobot 280 Arduino	€ 420,95	€ 559,95		€ 68,63	€ 489,58
MechArm M5Stack	€ 560,95	€ 723,95	€ 952,95	€ 75,70	€ 636,65
MyCobot 280 M5Stack	€ 560,95	€ 723,95		€ 68,63	€ 629,58
MyCobot 280 Raspberry Pi	€ 750,95	€ 896,95		€ 68,63	€ 819,58
MyPalletizer 260 M5Stack	€ 513,95	€ 701,95	€ 963,95	€ 68,63	€ 582,58
MyPalletizer 260 Pi	€ 609,95	€ 797,95	€ 060,95 1	€ 68,63	€ 678,58
MechArm Pi	€ 750,95	€ 896,95	€ 129,95 1	€ 75,70	€ 826,65

Brazo Robótico Adept	€ 76,62			€ 54,05	€ 130,67
RoArm-M1	€ 345,71			€ 59,39	€ 405,10
Brazo robótico Braccio	€ 285,97			€ 69,84	€ 355,81
Brazo robot de madera	€ 74,89			€ 53,75	€ 128,64
Brazo robótico AL5A	€ 252,62				€ 252,62
Brazo robótico Arduino uno R3	€ 76,62			€ 53,75	€ 130,37
Brazo robótico AL5D					€ -
RaspoArm-S	€ 51,27			€ 53,75	€ 105,02
Brazo robótico AL5D PLTW	€ 505,79				€ 505,79
ArmPi mini	€ 780,45			€ 3,90	€ 784,35
ArmPi mini	€ 250,34			€ 79,50	€ 329,84
ArmPi Pro	€ 519,24			€ 73,72	€ 592,96

Dado que este es un proceso iterativo, se volvió varias veces a la etapa de la selección de conceptos para analizar nuevas opciones cuando los posibles conceptos no llenaban las necesidades del cliente. Por eso, con el fin de mantener la claridad del documento y evitar la redundancia en la explicación de las iteraciones, el resto de la subsección estará enfocado en detallar la última iteración. Esta iteración final captura de manera representativa el proceso completo de la selección de conceptos, por lo que las iteraciones previas no son vitales. En la Tabla 4.7. se muestran los conceptos finales por seleccionar.

Tabla 4.7. Posibles conceptos por elegir para el proyecto.

<b>Últimos conceptos de robot</b>	
<b>Nombre del robot- Compañía</b>	<b>Imagen demostrativa del robot</b>
Magician- De la compañía Dobot:	
myCobot 280- De Elephant Robotics:	
DexArm- De Rotrics	
Mirobot- De WLkata	

Después de plantear los conceptos, se procedió con la cuarta etapa: evaluarlos. Primero se realizó un proceso de filtrado, para determinar los conceptos que requerían mejoras, podían combinarse o debían ser rechazados. Este proceso se representa en la Tabla 4.8.

Tabla 4.8. Matriz de selección para los conceptos del brazo robótico.

<b>Matriz de selección: Conceptos de brazos robóticos</b>					
<b>Criterios de selección</b>	<b>A Magician</b>	<b>B myCobot 280</b>	<b>C DexArm</b>	<b>D Mirobot</b>	<b>E Sensores externos</b>
<b>Programable</b>	0	0	0	0	
<b>Población meta (código, simulador)</b>	0	-	0	0	+
<b>OpenSource</b>	+	+	+	+	+
<b>Efactor</b>	0	-	+	+	-
<b>Simulador</b>	0	0	-	+	-
<b>Sensores</b>	0	0	0	0	+
<b>Disponibilidad inmediata</b>	+	0	0	-	+
<b>Cinemática inversa.</b>	+	0	0	0	-
<b>Aprobación del cliente</b>	+	-	-	0	+
<b>Suma +</b>	3	1	2	5	5
<b>Suma 0</b>	5	5	5	5	0
<b>Suma -</b>	0	3	2	1	3
<b>Evaluación neta</b>	3	-2	0	4	2
<b>Lugar</b>	2	5	4	1	3
<b>¿Continuar?</b>	Combinar	No	No	Combinar	Combinar

Como se explicó en la etapa anterior, esta metodología implementa la iteración para poder conseguir los mejores resultados posibles. Con el proceso de filtrado, se encontraron conceptos que no eran adecuados ya que no se alineaban a las necesidades del cliente y se rechazaron o necesitaban combinarse. Al

rechazar o buscar alternativas, se volvía a la etapa de generación de conceptos para encontrar otras opciones.

Como parte esta cuarta etapa de selección de un concepto ganador. Para el desarrollo de este proyecto se aplicaron varios métodos para poder elegir la mejor opción debido a que algunos robots quedaron en la misma categoría. Todos estos métodos se mencionan en el libro de “Diseño y desarrollo de productos” de Ulrich y Eppinger [21]. Primeramente, se realizó la metodología de “Pros y contras” donde se creó una lista de puntos débiles y fuertes de cada concepto. Después se realizó la metodología de “Decisión externa” en conjunto con “Votación múltiple” donde los votantes fueron el Dr. Abraham Prieto García, el Dr. Francisco Bellas Bouza y mi persona. Hay que aclarar que este fue un proceso iterativo y se recurrió varias veces a buscar nuevos conceptos y hacer nuevas listas de “Pros y contras” y repetir las votaciones de las nuevas opciones en contraste del o de los ganadores de la vez anterior.

Ya con las preferencias claras, se llegó a la última ronda de la iteración, donde se acudió al método “Campeón del producto” donde mi persona indicó cuál era el robot que consideraba adecuado según las tablas realizadas en Tabla 4.8 y Tabla 4.9. En la Tabla 4.9 se puede apreciar la última matriz de evaluación realizada para los conceptos del brazo robótico.

Con el concepto ganador definido (*Magician* de **Dobot**), se pasó al debido proceso en la empresa del trámite de la compra y se pasó a la siguiente etapa.

Tabla 4.9. Matriz de evaluación para los conceptos del brazo robótico

<b>Matriz de evaluación: Conceptos de brazos robóticos</b>					
<b>Criterios de selección</b>	<b>A+E Magician + Sensores externos</b>			<b>D+E Mirobot + Sensores externos</b>	
	<b>Peso</b>	<b>Ponderado</b>	<b>Calificación</b>	<b>Ponderado</b>	<b>Calificación</b>
<b>Programable</b>	<b>30%</b>	<b>1,5</b>	<b>5</b>	<b>1,5</b>	<b>5</b>

<b>Población meta (código, simulador)</b>	<b>30%</b>	<b>1,5</b>	<b>5</b>	<b>1,5</b>	<b>5</b>
<b>OpenSource</b>	<b>10%</b>	<b>0,5</b>	<b>5</b>	<b>0,5</b>	<b>5</b>
<b>Efector</b>	<b>2%</b>	<b>0,1</b>	<b>5</b>	<b>0,1</b>	<b>5</b>
<b>Simulador</b>	<b>2%</b>	<b>0,1</b>	<b>5</b>	<b>0,08</b>	<b>4</b>
<b>Sensores</b>	<b>5%</b>	<b>0,2</b>	<b>4</b>	<b>0,2</b>	<b>4</b>
<b>Disponibilidad inmediata</b>	<b>15%</b>	<b>0,75</b>	<b>5</b>	<b>0,15</b>	<b>1</b>
<b>Cinemática inversa.</b>	<b>1%</b>	<b>0,05</b>	<b>5</b>	<b>0,05</b>	<b>5</b>
<b>Aprobación del cliente</b>	<b>5%</b>	<b>0,2</b>	<b>4</b>	<b>0,15</b>	<b>3</b>
<b>Total</b>	<b>1</b>	<b>4,9</b>	<b>43</b>	<b>4,23</b>	<b>37</b>
<b>Lugar</b>			<b>1</b>		<b>2</b>

#### 4.3.2 Sensor: Generación y selección de conceptos

Para el área de sensores, se continuó la tercera etapa de generación de conceptos y se analizaron los subproblemas que se tenían. En la Figura 4.5 se puede analizar el árbol para los subproblemas del sensor.

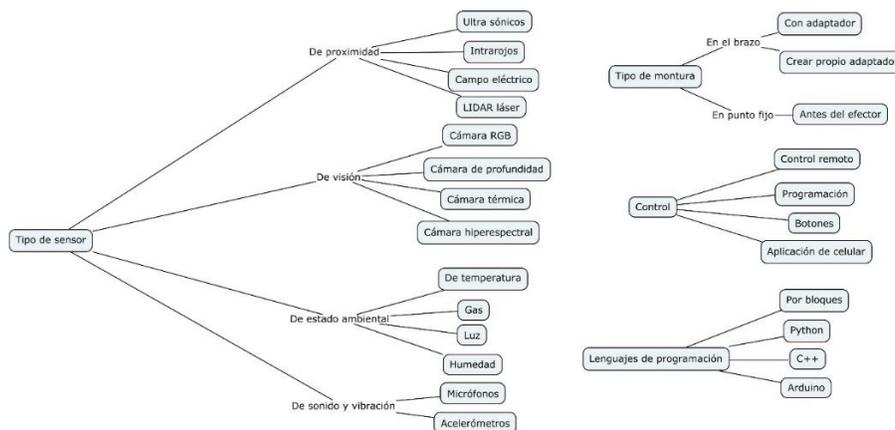


Figura 4.5. Diagrama de árbol de los subproblemas de los sensores. Elaboración propia.

Con los subproblemas que se plantearon y sus variables, se analizaron los posibles conceptos que se encontraban en el mercado.

Después de plantear los conceptos, se procedió con la cuarta etapa: seleccionarlos. Primero se realizó un filtrado, para determinar los conceptos que requerían mejoras, podían combinarse o debían ser rechazados. En la Tabla 4.10 se puede apreciar la matriz de selección de algunos de los conceptos encontrados para representar el proceso de desarrollo.

Tabla 4.10. Matriz de selección de conceptos para los sensores

<b>Matriz de selección: Conceptos de sensores</b>				
<b>Criterios de selección</b>	<b>A Cámara RGB</b>	<b>B Sensor de proximidad infrarrojo</b>	<b>C Sensor de proximidad ultrasónico</b>	<b>D Micrófono</b>
<b>Programable</b>	0	0	0	-
<b>Población meta (código, simulador)</b>	+	0	0	0
<b>OpenSource</b>	0	0	0	0
<b>Efactor</b>	0	0	0	0
<b>Simulador</b>	+	0	0	0
<b>Sensores</b>	0	0	0	0
<b>Disponibilidad inmediata</b>	+	0	0	0
<b>Cinemática inversa.</b>	0	0	0	-
<b>Aprobación del cliente</b>	+	0	0	0
<b>Suma +</b>	3	0	0	0
<b>Suma 0</b>	5	9	9	7
<b>Suma -</b>	0	0	0	2
<b>Evaluación neta</b>	3	0	0	-2
<b>Lugar</b>	1	2	3	4
<b>¿Continuar?</b>	<b>Sí</b>	<b>No</b>	<b>No</b>	<b>No</b>

Con esta matriz se seleccionó el concepto ganador. Dado que el “Concepto A” quedó en primer lugar con gran diferencia, este fue el concepto ganador.

Combinar conceptos no era opción porque la población meta del proyecto (estudiantes preuniversitarios) tienen ciertas limitaciones en el conocimiento técnico de sensores, además que las lecciones cuentan con limitaciones de tiempo. En caso de combinar ideas, el nivel del proyecto superaría el esperado para los alumnos.

Otro factor por destacar es que, en el proceso de filtrado, se descartaron las demás opciones. Esto significó que, en este caso, no fue necesario realizar la matriz de evaluación de conceptos, como indican Ulrich y Eppinger en “*Diseño y desarrollo de productos*”:

“La evaluación de conceptos se usa cuando una mayor resolución va a lograr una mejor diferencia entre conceptos que compiten.” [21]

### **Selección de cámara**

Para la selección de cámara, se realizó un proceso de filtrado donde el criterio de evaluación de mayor peso era la disponibilidad con el proveedor de la universidad y alrededores.

Con los conceptos ya filtrados por este criterio, quedaron dos posibles conceptos a elegir. A los cuales se les aplicó la matriz de selección que se puede apreciar en la Tabla 4.11.

Tabla 4.11. Matriz de selección de conceptos para las cámaras.

<b>Matriz de selección: Conceptos de cámaras</b>		
	<b>A Huskylens</b>	<b>B Pixy2</b>
<b>Criterios de selección</b>		
<b>Programable</b>	0	0
<b>Población meta (código, simulador)</b>	+	0
<b>OpenSource</b>	0	0
<b>Acceso a los datos</b>	0	0
<b>Disponibilidad inmediata</b>	+	0
<b>Aprobación del cliente</b>	+	0
<b>Suma +</b>	3	0

<b>Suma 0</b>	3	6
<b>Suma -</b>	0	0
<b>Evaluación neta</b>	3	0
<b>Lugar</b>	1	2
<b>¿Continuar?</b>	Sí	No

Al igual que para la selección de conceptos de sensor, en este caso tampoco fue necesario recurrir a una matriz de evaluación.

El concepto ganador fue la cámara *SEN0305 HUSKYLENS*, la cual cuenta con las siguientes características obtenidas de la hoja de datos [13]:

- Longitud focal = 4.8
- Campo de visión = 51
- Tamaño = 320x240 píxeles

La Huskylens es además una cámara basada en Inteligencia Artificial. Esta tiene diferentes modos de funcionamiento las cuales incluyen la detección de caras, reconocimiento de objetos, seguimiento de líneas, seguimiento de objetos, reconocimiento de colores, identificación de etiquetas basadas en códigos QR y con la posibilidad de guardar una cara, objeto y color para ser identificado [13].

Para acceder a los datos de esta cámara, se necesita un microcontrolador externo (como un Arduino UNO) y con el código y la conexión que se encuentra en la página oficial [13].

#### 4.3.3 Inteligencia Artificial: Generación y selección de conceptos

Para el área de Inteligencia Artificial, como se hizo en las partes pasadas, se regresó a continuar la tercera etapa donde se evaluaron los subproblemas de esta área. En la Figura 4.6 se puede observar el diagrama de árbol creado.

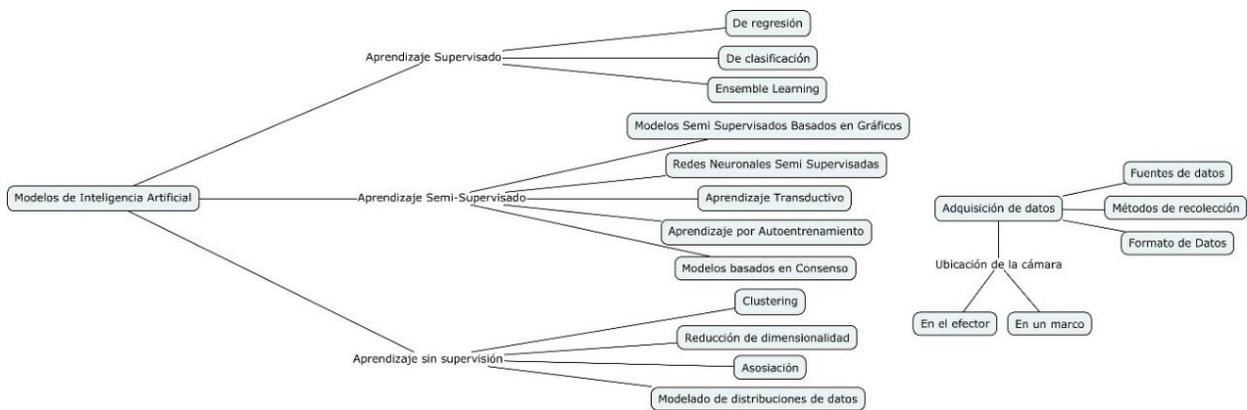


Figura 4.6. Diagrama de árbol para la inteligencia artificial. Elaboración propia

Con los subproblemas identificados, se pasó a la cuarta etapa de selección de conceptos. En la Tabla 4.12 se puede apreciar la matriz de datos seleccionados para el modelo de base de datos para la inteligencia artificial a implementar.

Tabla 4.12. Matriz de selección de conceptos para el modelo de base de datos.

Matriz de selección: Conceptos de base de datos			
Criterios de selección	A Supervisado	B Semi-supervisado	C Sin supervisar
Población meta (nivel de complejidad)	+	0	-
Tiempo de programación	+	0	-
Tiempo de toma de datos	+	0	+
Tiempo de aprendizaje	+	0	+
Aprobación del cliente	+	0	-
Suma +	5	0	2
Suma 0	0	5	0
Suma -	0	0	3
Evaluación neta	5	0	-1

Lugar	1	2	3
¿Continuar?	Sí	No	No

Con el concepto de modelo de IA se pasó a la subetapa: modelado y construcción de prototipos. Para esto se creó un código con la lógica con la que se obtendrían los datos para entrenar la Inteligencia Artificial.

Con la lógica de la toma de datos, se pasó a la quinta etapa, pruebas de concepto, donde se creó el código para la toma de datos.

El tema de Inteligencia Artificial y sus códigos son un área que abarca muchas partes, las cuales no todas se buscaron que el estudiante replicara, por eso las pruebas de concepto serán mayormente detalladas en el *Capítulo 5: Creación de la Inteligencia Artificial*. En el *Capítulo 6: Lecciones y Actividades Docentes* se verá lo que se le pide al estudiante.

Una vez con el tipo de IA seleccionado, se realizó otra reunión con el cliente. En esta se indicó que se encontraba interesado en varios modelos de inteligencia artificial del tipo supervisado. De esta manera, haciendo una matriz de selección con el criterio del interés del cliente, se llegaron a varios conceptos pendientes de revisión. La matriz de selección corresponde a la mostrada en la Tabla 4.13.

Tabla 4.13. Matriz de selección de conceptos para los modelos de IA supervisados.

<b>Matriz de selección: Conceptos de modelos de IA Supervisados</b>					
<b>Criterios de selección</b>	<b>A Regresión Polinomial</b>	<b>B Árbol de Decisión</b>	<b>C Árbol de Decisión Mult. (Referencia)</b>	<b>D Red Neuronal</b>	<b>E k-NN's</b>
<b>Complejidad de programación</b>	-	0	0	-	-
<b>Tiempo de aprendizaje</b>	-	-	0	-	+
<b>Aprobación del cliente</b>	+	+	0	+	+
<b>Suma +</b>	1	1	0	1	2
<b>Suma 0</b>	0	1	3	0	0
<b>Suma -</b>	2	1	0	2	1
<b>Evaluación neta</b>	-1	0	0	-1	1
<b>Lugar</b>	4	1	2	5	3

¿Continuar?	Revisar	Revisar	Revisar	Revisar	Revisar
-------------	---------	---------	---------	---------	---------

Para los modelos propuestos, se tomó la decisión de implementarlos en el desarrollo y realizar un análisis con los resultados. Para luego usar el análisis como el criterio de evaluación en la matriz de evaluación de conceptos. De esta manera se puede tomar el concepto que mejor se adecua a la aplicación, no fuera ya estudiado en lecciones pasadas por los estudiantes y que sea un nivel apropiado para los estudiantes.

De esta forma se realizó la quinta etapa, pruebas de concepto, con los cinco modelos a revisar. La prueba de concepto se encuentra más detallada en el *Capítulo 5: Creación de la Inteligencia Artificial*.

Después de realizar el análisis de resultados para cada modelo de IA y considerando estos criterios de evaluación, se desarrolló una matriz de selección de conceptos, la cual se encuentra en la Tabla 4.14.

Tabla 4.14. Matriz de selección de conceptos: Modelo de Inteligencia artificial.

Matriz de selección: Conceptos de Inteligencia Artificial					
Criterios de selección	A Árbol de Decisión (Referencia)	B Árbol de Decisión Mult.	C k-NN's	D Regresión Polinomial	E Red Neuronal
MSE Salida 1	3	2	4	3	1
MSE Salida 2	5	4	2	1	
R <sup>2</sup> Salida 1	4	3	5	3	2
R <sup>2</sup> Salida 2	5	4	3	2	
Gráfico de Residuos	5	5	5	5	5
Valores Reales vs Predichos	1	1	3	5	3
Q-Q Plot	3	5	5	4	4
Curva de Aprendizaje	5	5	5	5	1
Error	1	5	5	1	1
<b>Total</b>	32	34	37	29	17
<b>Lugar</b>	3	2	1	4	5

Finalmente, con el concepto ganador claro (vecinos cercanos, kNN's) se da como concluida la cuarta etapa de selección de conceptos.

#### **4.4 Quinta etapa: Pruebas de concepto**

##### **4.4.1 Prueba de concepto del robot**

En la quinta etapa, pruebas de concepto, se adaptó para utilizar el simulador. Ya que este es un modelo inicial para la siguiente etapa de pruebas de concepto (el robot). Para ello se empezó con lo básico, aprendiendo a usar el simulador, la barra del menú, la de herramientas (como se agregan objetos, marcos de referencia, cómo realizar los movimientos del robot, entre otros comandos), controles del ratón y de teclado.

Estos primeros pasos en el simulador eran esenciales para conocer las capacidades del robot e identificar si este cumplía con las necesidades de crear lecciones de inteligencia artificial. Después de analizar el funcionamiento básico del simulador y del robot en el mismo, se procedió a la creación de códigos en Python para crear los mismos comandos que permite la barra de herramientas.

Dado que estos códigos y conocimientos se usaron para la creación de lecciones, estos se explicarán a detalle en el Capítulo 6: Lecciones y Actividades docentes.

##### **4.4.2 Prueba de concepto del sensor**

Para la quinta etapa, probar conceptos, se adaptó para también ser implementado en el simulador. Se buscó en el simulador cómo agregar una cámara y la forma de agregar las configuraciones propias de esta para asemejar el proceso lo máximo posible al real. También se trabajó la implementación de la cámara por medio de código de Python.

Al igual que en la sección del robot, estos códigos y conocimientos se usaron para la creación de lecciones, por lo que los códigos se explicarán a detalle en el Capítulo 6: Lecciones y Actividades docentes.

#### 4.4.3 Prueba de concepto de Inteligencia Artificial

Para la quinta etapa de Inteligencia Artificial, se realizaron las pruebas de cada modelo de Inteligencia Artificial en el simulador. Dado que esta sección requiere mayor detalle, se abarcará el *Capítulo 5: Creación de la Inteligencia Artificial* para la descripción y desarrollo de esta, considerando la creación de la base de datos y el desarrollo de los cuatro modelos de Inteligencia Artificial seleccionados en la Tabla 4.13. En el *Capítulo 7: Análisis de resultados* se discutirán los resultados obtenidos, análisis que fue la base para la generación de la Tabla 4.14.

#### **4.5 Sexta etapa: Documentación**

Después de tener las pruebas de concepto realizada, se procedió a la sexta etapa, documentación, donde se llevó a cabo la implementación de los códigos creados más los conceptos básicos y se escribieron las lecciones.

Cabe destacar que las lecciones fueron escritas en inglés. Si bien la empresa y el cliente mantienen como idioma oficial el español, las lecciones se piensan implementar a nivel europeo. Por lo tanto, el idioma estándar para las lecciones es el inglés.

La descripción de las lecciones realizadas se encuentra en el *Capítulo 6: Lecciones y Actividades Docentes*.

## 5. Creación de la Inteligencia Artificial

### 5.1 Lógica de movimiento para la base de datos

El primer paso es poder analizar el entorno. Para esto se conoce que se tiene el robot en la mesa, con la cámara adherida arriba de la muñeca del robot. Con la cámara se puede observar la mesa de trabajo y el objeto que queremos alcanzar (la torre de ajedrez). De manera gráfica, esto se puede apreciar en la Figura 5.1.

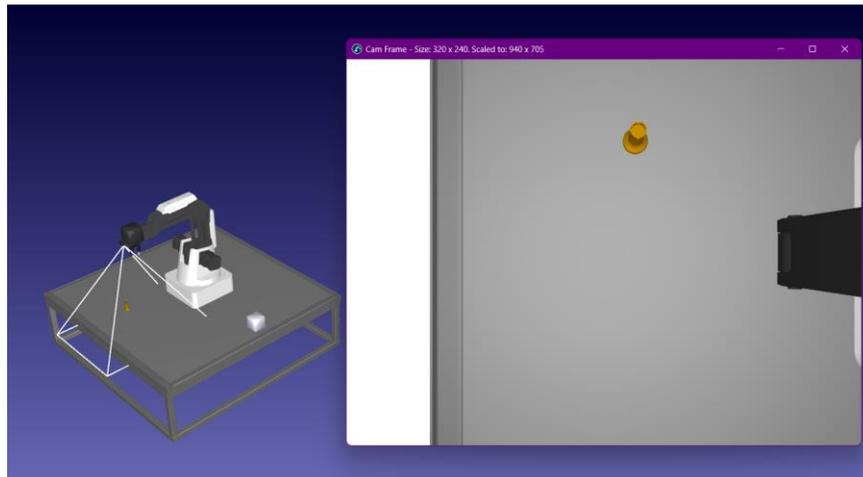


Figura 5.1. Entorno del robot y la visión de la cámara. Elaboración propia.

Para el proceso completo de la Inteligencia Artificial, se mantendrán dos articulaciones estáticas, esto debido a que el nivel que se busca para todo el proceso (en especial la solución) requiere que sea una forma sencilla de resolver nuestra aplicación debido a que la población meta está apenas comenzando a estudiar robótica e Inteligencia Artificial.

Las primeras dos articulaciones del robot mueven principalmente al robot en  $(x,y)$ . Mientras que las articulaciones 3 y 4 (efector final), cambian mayoritariamente la altura y ángulo del efector. Por lo tanto, si se definen las articulaciones 1 y 2 como las variables, permiten que el robot pueda analizar gran parte de la mesa. Además, si la articulación 3 se usa específicamente en  $25^\circ$ , permite que la cámara se mantenga relativamente a la misma altura en todo el rango posible de las articulaciones variables. Por ello, las articulaciones se verán definidas en  $(c1, c2,$

25, 0), siendo  $c_1$  y  $c_2$  variables y correspondientes a las articulaciones 1 y 2 respectivamente.

Para ayudar también a que la cámara se mantenga a una altura similar durante toda la aplicación, se define que la torre solo puede estar en un área determinada. Para esto se crea el código RandomRook, que define el área en el que puede estar la torre para ser alcanzada. El rango corresponde a:

- $x = [-121, -57]$  mm del simulador
- $y = [-104, 104]$  mm del simulador

Con esto definido, se empieza con el movimiento del robot. Para esto primero se implementa el código para el análisis de las imágenes (el cual se encuentra explicado en el Capítulo 6: Lecciones y Actividades Docentes). Este código devuelve la posición media del objeto.

Después se define la posición final en la que se debe encontrar el objeto para que quede justo abajo del efector. Esta posición corresponde a los valores de referencia (' $x_0 = 120$ ', ' $y_0 = 205$ ') para las coordenadas ' $x$ ' e ' $y$ ', con un umbral (' $umbral = 2$ ') para considerar pequeñas desviaciones.

Para hacer un movimiento más eficaz, si el objetivo se encuentra lejos de la posición ideal, el robot debe de hacer grandes movimientos para llegar rápido. Pero para que sea más preciso, es necesario que después de pasar cierto rango, el movimiento debe de ser menor. Así se definen rangos aceptables alrededor de la posición ideal (estos se llaman: ' $x_{o1}$ ', ' $x_{o2}$ ', ' $y_{o1}$ ', ' $y_{o2}$ ') y rangos de error permitidos (' $ex_1$ ', ' $ex_2$ ', ' $ey_1$ ', ' $ey_2$ '). En la Figura 5.2 se puede apreciar el rango definido.

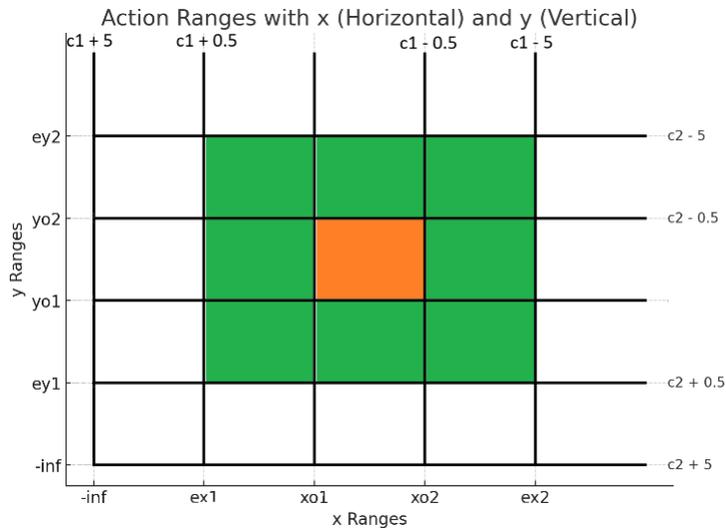


Figura 5.2. Gráfica de rangos de movimiento. La posición ideal que considera el umbral es la zona naranja. El rango para hacer pequeños movimientos es la zona verde. El resto del rango se hacen grandes movimientos. Elaboración propia.

Dependiendo de la posición de 'x' e 'y' y con los rangos definidos se ajustan las articulaciones  $c1$  y  $c2$ . El movimiento se define con variaciones de 5 grados y 0.5 grados, los cuales se detallan de la forma:

- Si 'x' está por debajo de 'ex1', se reduce 'c1' en 5 grados.
- Si 'x' está entre 'ex1' y 'xo1', se reduce 'c1' en 0.5 grados.
- Si 'x' está entre 'xo2' y 'ex2', se incrementa 'c1' en 0.5 grados.
- Si 'x' está por encima de 'ex2', se incrementa 'c1' en 5 grados.
- Si 'x' está entre 'xo1' y 'xo2', 'c1' no se modifica.

Se realiza un proceso similar para  $c2$  con respecto a la coordenada 'y'. Estas condiciones pueden observarse mejor en la Figura 5.3.

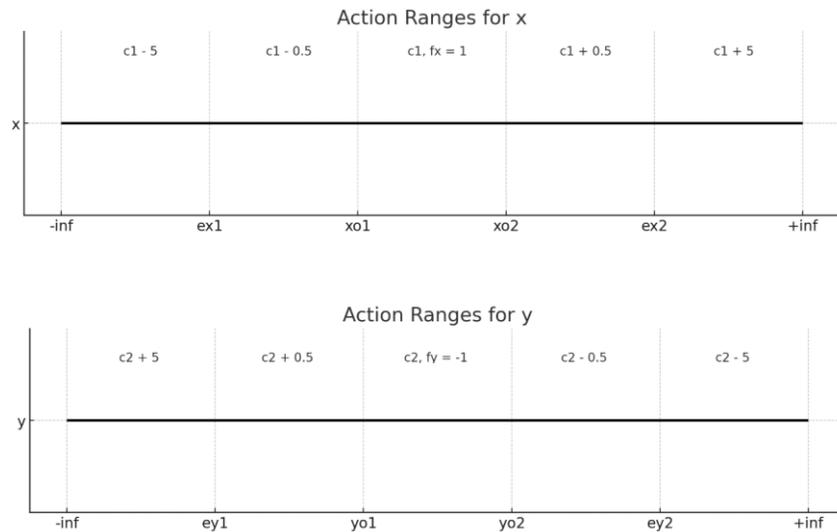


Figura 5.3. Rangos de acción para el robot según la posición de la pieza objetivo. Elaboración propia.

Se aplican los nuevos valores a las articulaciones del robot y se procesa una nueva posición. Esto continúa en bucle hasta que el objetivo alcance el rango de valores ideales.

En resumen, el código ajusta las posiciones de las articulaciones del robot para que la posición leída de la torre de ajedrez se la de las coordenadas de referencia donde el efector se encuentra justo sobre esta. Un ejemplo de movimiento se puede apreciar en la Figura 5.4 donde la posición inicial del robot (considerada en el medio del efector) se denomina P0 y de ahí inicia una trayectoria en búsqueda de llegar a la pieza.

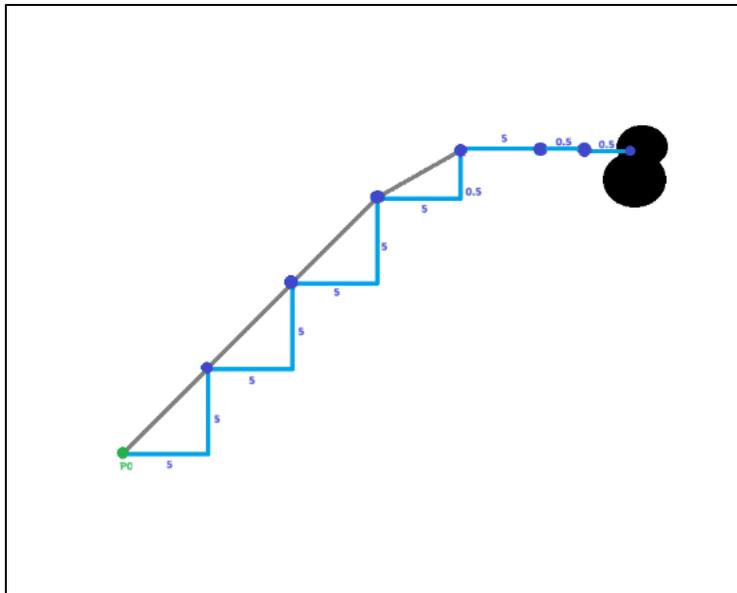


Figura 5.4. Ejemplo de trayectoria de movimiento del robot para llegar a la pieza a base de casos definidos. Elaboración propia.

## 5.2 Creación de la base de datos

El caso pasado muestra la lógica de una trayectoria con pasos definidos para conseguir que el robot llegue a la pieza. Este será el objetivo de la Inteligencia Artificial. Para que la IA aprenda qué movimiento realizar, se le debe enseñar a esta con una serie de datos de los que ella pueda aprender. Además, se quiere que esta sea capaz de elegir movimientos de tal manera que no se defina en cuánto mover al robot, si no que sea la Inteligencia Artificial quien decida esto.

Para ello es vital tener una base de datos que abarque la mayor cantidad de escenarios posibles. Es ideal crear un código para hacer esta base de datos y que sea lo más autónoma posible, ya que de otra forma se duraría mucho tiempo en conseguir la base de datos.

Para la lógica del código de la generación de la base de datos se utilizaron 4 archivos .CSV por lo que primero se explicará cada uno:

- ❖ DataBase0: almacena todos los puntos iniciales buscados, incluso los que no son válidos.
- ❖ DataBaseRamas: almacena los 20 puntos aleatorios cercanos al punto inicial donde se elige un mínimo de estos.

- ❖ DataBaseMinimos: almacena solo los datos mínimos seleccionados, esto incluye si el robot decide que la mejor posición es no moverse.
- ❖ DataBaseTotal: almacena todos los datos de los puntos aleatorios. A diferencia de Ramas, este no se sobrescribe.
- ❖ DataBaseTrayectoria: almacena únicamente los datos de la trayectoria que realiza el robot

En esos archivos se guardaron las variables y datos de interés se encuentran: grados de las articulaciones 1 y 2, distancia entre el objeto y la posición actual del objeto y la variación de las articulaciones con respecto a la posición pasada.

A continuación se describe brevemente la lógica detrás de la creación del código para la generación de la base de datos y en la Figura 5.5 se encuentra el diagrama de flujo del código.

- P0: Se genera un punto aleatorio inicial P0 para que el robot empiece en un punto diferente cada vez. Las posiciones aleatorias están limitadas a que exista la presencia del objeto en la imagen. En caso de no encontrarse, debe buscar una nueva posición. Se guardan los datos en los archivos correspondientes.
- Pi: Se lee la última posición válida almacenada en DataBaseTrayectoria. Se crea un ciclo donde se generan nuevas posiciones aleatorias cercanas a la posición inicial para 20 posiciones nuevas y las almacena en los archivos correspondientes. Cada vez que este código corre, se sobrescribe el archivo, de esta manera solo se mantienen los datos de la última ronda.
- Selección de Pi: Se leen los datos de las posiciones generadas en el archivo DataBaseRamas, con esto se selecciona la posición que tenga la menor distancia con el objetivo y se guardan los datos en los archivos correspondientes.
- Preprocesado de datos: Aquí se eliminan las filas del archivo DataBaseTrayectoria donde la decisión fue no moverse y se ajustan los datos para que solo las decisiones válidas se incluyan en la base de datos final.

- Dataset100: este código busca que se reitere el proceso de generación y selección de nuevas posiciones 100 veces. En caso de que el robot cumpla el objetivo, genera un nuevo P0 y modifica la ubicación de la torre y continua con el proceso de generación y selección de Pi hasta que se termine el ciclo o que llegue a la meta.

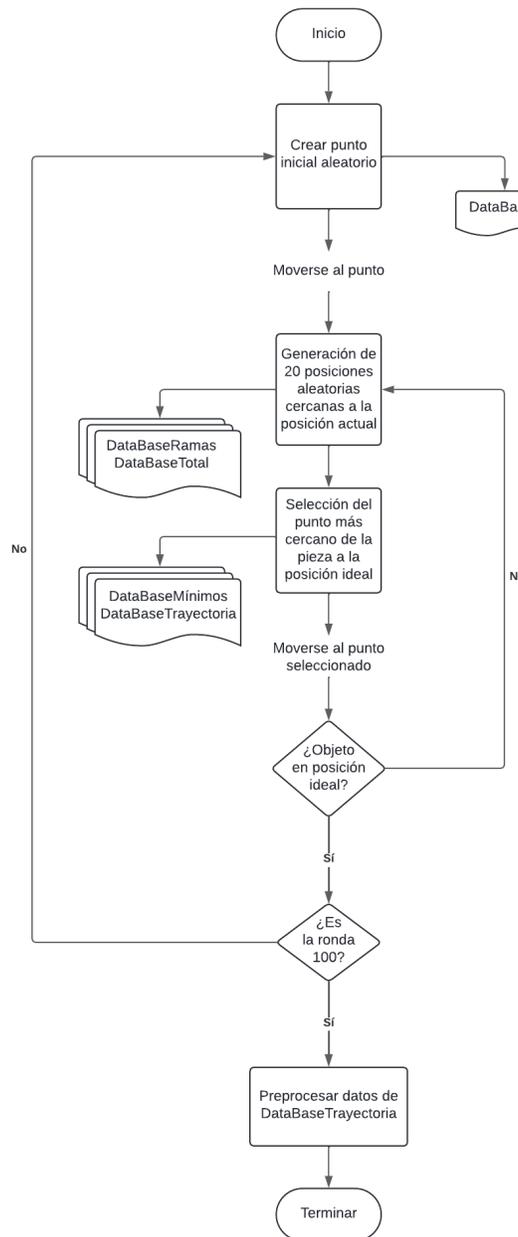


Figura 5.5. Diagrama de flujo para el código de la base de datos. Elaboración propia

De esta forma el resultado de cada código puede ser explicado en los ejemplos de la Figura 5.6.y Figura 5.7.

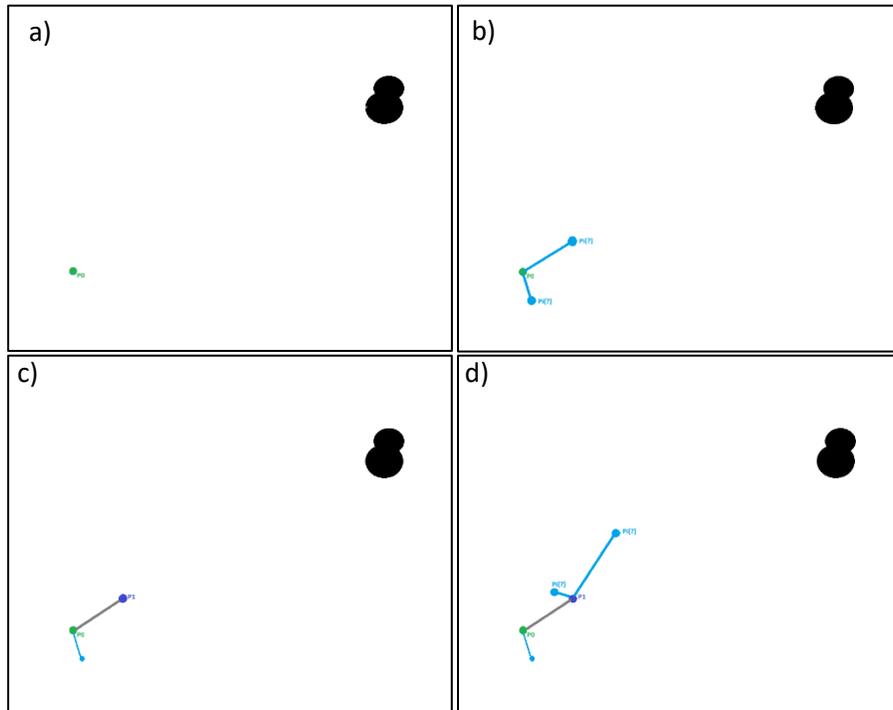


Figura 5.6. Ejemplo de movimiento para la base de datos, asumiendo dos posiciones nuevas. a) Ejemplo de resultado del código P0. b) Ejemplo de resultado del código Pi. c) Ejemplo de resultado del código "Selección de Pi". d) Ejemplo 2 de resultado del código Pi. Elaboración propia.

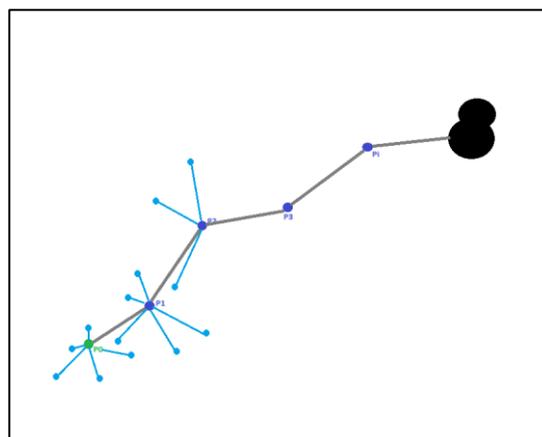


Figura 5.7. Ejemplo de trayectoria y posibles posiciones aleatorias que realiza la base de datos. Elaboración propia.

Este proceso asegura que la base de datos generada sea adecuada para entrenar un modelo de aprendizaje supervisado que pueda predecir la posición óptima del robot basada en las detecciones de la cámara. Para las lecciones, los estudiantes implementarán la lógica básica del manejo de los datos para entender mejor el flujo completo de la creación y uso de estas en aplicaciones de robótica y visión artificial; esto se ve con mayor detalle en el Capítulo 6: Lecciones y Actividades Docentes.

### **5.3 Programación de los modelos de Inteligencia Artificial**

Para la implementación de todos los modelos de Inteligencia Artificial, se utilizó la librería de sklearn. La cual es una biblioteca de aprendizaje automático gratuita y de código abierto para Python y que está basada en SciPy.

Dado que se usó la misma librería para cada modelo de Inteligencia Artificial, los códigos son bastante similares entre sí. Las dos diferencias son: el comando para llamar al modelo y que árboles multisalida y redes neuronales permiten el entrenamiento de solo un modelo para las dos salidas; mientras que para el resto (árboles de decisión, vecinos cercanos y regresión polinómica) se requiere el entrenamiento de dos modelos para cada variable objetivo. Para los modelos que solo cuentan con una salida, se entrenaron dos tipos iguales de modelo de Inteligencia Artificial. Igualmente, no se consideró la combinación de los modelos por la población meta del alcance del proyecto.

Para cada modelo se siguió el siguiente formato:

Dentro de las librerías necesarias, se usó la librería sklearn del modelo a implementar para construir y utilizarlo.

Después se cargaron y prepararon los datos desde el archivo DataBaseTrayectoria.csv y se cargó en un DataFrame de la librería “pandas”.

Luego se separan los datos de entrada (x,y) y las variables objetivo (las variaciones v1 y v2).

Se dividen los datos en conjuntos de entrenamiento y prueba para cada variable objetivo, usando el 20% de los datos para prueba y el 80% para entrenamiento.

Después con la función del modelo de IA a implementar se crea un modelo para cada variable objetivo (a excepción de árboles multisalida y redes neuronales). Además, se definió un valor "RandomState", el cual permite que siempre que se corra el código va a empezar en el mismo punto, lo que hará que tenga siempre la misma salida de datos. Esto con el fin de que se pueda hacer el mismo análisis de resultados sin importar si se requiere correr de nuevo.

Luego se hizo el entrenamiento de los modelos con el comando respectivo de cada modelo.

Para esto se realizaron los códigos correspondientes a: árbol de decisión, árbol de decisión de múltiple salida, regresión polinomial, vecinos cercanos kNN's y redes neuronales. Los códigos tienen lógicas similares entre sí, solo varía la línea que llama al modelo de IA.

Se cargaron los datos del archivo DataBaseTrayectoria, donde los valores de importancia serán las coordenadas (x,y) que la cámara detecta como posición del objeto y la variación de las articulaciones del robot.

Luego se dividen los conjuntos de datos de entrenamiento y de prueba, utilizando una proporción del 20% para las pruebas y el 80% para el entrenamiento.

Después se realiza la creación y entrenamiento del modelo correspondiente. Para árbol de decisión, kNN's y regresión polinomial, se definen dos veces el mismo modelo. Esto porque estos modelos solo manejan una salida. Para el árbol de multisalida y la red neuronal, ambos fueron programados para tener dos salidas.

Para el árbol de vecinos, se solicitaron 10 vecinos para cada salida.

Posteriormente, se usan los modelos entrenados para hacer predicciones sobre el conjunto de prueba.

Por último, para cada modelo se implementaron evaluaciones de estos. Como cálculos de métricas de rendimiento, error cuadrático medio (MSE) y el coeficiente de determinación  $R^2$  para cada salida. Además, se implementaron gráficas como: Predicciones vs Valores Reales, Gráfico de residuos, Gráfico Q-Q, Cálculo de distancias de la gráfica Q-Q, curvas de aprendizaje, superficie de decisión. Estos resultados se mostrarán en el Capítulo 7: Análisis de resultados.

Cada uno de estos resultados proporciona una vista completa de cómo trabajar con cada modelo meta, desde la preparación de datos hasta la evaluación y visualización de los resultados.

Después de realizar esto, se creó un código donde se implementaron todos los modelos, pero sin los resultados, para que este pudiera ser llamado con una entrada que definiera el modelo a usar y que guardara los datos para poder ser abiertos en el código que necesitara la IA.

## **6. Lecciones y Actividades Docentes**

### **6.1 Learning by doing**

El formato de aprendizaje implementado por AIM@VET en su proyecto es “Learning by doing”. Este método consta de aprender de las experiencias resultando directamente de las acciones que la propia persona hace, a diferencia de aprender de observar a otros hacer el trabajo, leer las instrucciones o descripciones de otras personas o de escuchar las lecturas de otros. A pesar de que escuchar, leer u observar sean acciones, “learning by doing” se refiere a la acción de hacer y crear ya que es el campo de experiencia directa la que el estudiante aprende. [22]

Esta forma de aprendizaje fue usada para el proyecto ya que las primeras lecciones implementaron este método y dado que es una continuación de un módulo de enseñanza de Inteligencia Artificial, se debe mantener la uniformidad. Además, esta forma de aprendizaje ha demostrado su efectividad con las primeras lecciones del Robobo.

### **6.2 Descripción de actividades y retos**

Para la creación de lecciones se tomó en cuenta los pasos necesarios para aprender el uso del robot (tanto en simulación como el real) y completar la meta de que el brazo llegue al punto deseado para tomar una pieza.

Para esto se desarrollaron posibles objetivos de aprendizaje de Inteligencia Artificial en el plan de estudio, tomando como base las guías para el Robobo (el robot de las lecciones ya implementadas, previas al desarrollo de este proyecto). Los objetivos del plan para las lecciones propuestas se encuentran en la Tabla 6.1, el cual cumple con el entregable planteado en los objetivos, donde se crea el documento base de las lecciones y con el indicador de al menos dos lecciones de aprendizaje de Inteligencia Artificial.

Tabla 6.1. Plan de lecciones propuestas para el proyecto.

Plan de lecciones				
Número de lección	Objetivo general	Objetivos específicos	Contenido didáctico	Actividades experimentales
<b>Primera Lección</b>	Conocer el uso de los actuadores del robot.	<p>1.1. Introducir los actuadores y grados de libertad.</p> <p>1.2. Introducir sistemas de coordenadas y transformaciones (traslación y rotación)</p> <p>1.3. Trabajar con el simulador.</p>	<p>Conceptos básicos de robótica.</p> <p>Bases del simulador RoboDK.</p> <p>Configuración del entorno del simulador RoboDK.</p>	<p>Creación de una trayectoria del robot para que se acerque a un objeto meta y luego se desplace al punto de depósito.</p> <p>Implementación de movimientos para acercarse a objetos, recogerlos y colocarlos en el punto de depósito.</p>
<b>Segunda Lección</b>	<p>Implementar herramientas de detección de objetos.</p> <p>Modelar el comportamiento de aproximación.</p>	<p>2.1. Conocer el uso de las cámaras en el ambiente industrial.</p> <p>2.2. Describir el entorno experimental</p> <p>2.3. Conocer librerías de detección de objetos.</p>	<p>Introducción a la cámara.</p> <p>Segmentación de imágenes.</p> <p>Binarización de imágenes.</p>	<p>Detectar varios objetos de colores con segmentación y binarización de imágenes.</p> <p>Detectar la pieza objetivo (una torre de ajedrez) con segmentación y binarización de imagen.</p> <p>Crear un código que movilice el efector final al punto promedio de la pieza de ajedrez detectada.</p>
<b>Tercera Lección</b>	Generar una base de datos y conocer de aprendizaje supervisado	<p>3.1. Crear una base de datos preliminar para el aprendizaje del movimiento del robot.</p> <p>3.2. Implementar de funciones de recompensa</p>	<p>Bases de aprendizaje supervisado. Bases de datos y su importancia.</p> <p>Explicación base de la realización de</p>	<p>Hacer el código para encontrar el punto promedio de la torre de ajedrez.</p> <p>Hacer el código base similar a P0, Pi y selección de Pi (del Capítulo 5).</p>

			la base de datos que se usará en la siguiente lección.	
<b>Cuarta Lección</b>	Generar una técnica de aproximación con un modelo de Inteligencia Artificial	<p>4.1. Representar gráficamente los datos iniciales.</p> <p>4.2. Conocer de modelos de Inteligencia Artificial.</p> <p>4.2. Programar una técnica de aproximación con un modelo preseleccionado.</p>	<p>Descripción de modelos de entrenamiento supervisado.</p> <p>Importancia de bases de datos.</p> <p>Modelo de vecinos cercanos (kNN's).</p>	<p>Hacer un código para representar gráficamente la base de datos.</p> <p>Crear el código del modelo de vecinos cercanos (kNN's).</p>
<b>Quinta Lección</b>	Implementar en el robot simulado y real el movimiento con Inteligencia Artificial	<p>5.1. Integración del modelo de aproximación en el simulador.</p> <p>5.2. Integración de modelos de aproximación en el robot real.</p> <p>5.3. Discusión sobre resultados: simulación vs real.</p>	<p>Explicación de la integración del modelo creado con el simulador y robot real.</p> <p>Explicación de los resultados deseables con el modelo.</p> <p>Explicación de las diferencias entre simulador y robot real.</p>	<p>Código para leer datos de la cámara y encontrar la distancia con la torre de ajedrez.</p> <p>Crear un bucle de corrección de posición del robot implementando el modelo de IA entrenado para que el efector llegue a la pieza meta.</p> <p>Desplazar el robot verticalmente hasta llegar al objeto para su toma.</p>

### 1.1.1 Información necesaria de las lecciones: Implementación en el robot real

Durante el desarrollo y validación de los códigos, se encontró que la aplicación del robot real (DobotLab) fue actualizada y no hay forma de descargar

versiones previas. Esto influyó a la implementación del robot real, ya que el simulador realiza un código análogo para implementar los comandos de movimiento específicos de RoboDK a la aplicación, pero el código que crea es para versiones previas de DobotLab.

Para esto se creó una clase que se utilizó para que los estudiantes creen el código del simulador y este mismo pueda ser implementado en el robot real, haciendo la mínima cantidad de cambios en el código creado. El código de esta clase se puede encontrar en el enlace del Anexo G.

### 6.2.2 Información necesaria de la cuarta lección: Base de datos

En esta lección, se consideró que el código de la base de datos era muy extenso y complejo para la población meta, además que dura un tiempo considerable en la toma de datos que los estudiantes no poseen. Por lo que en la lección se solicita la creación de un código simplificado y se les proporciona la base de datos creada durante la realización de este proyecto para la siguiente lección. Con el código simplificado, ellos pueden apreciar la forma que se usó para la creación de la base de datos, además de las consideraciones para tener en cuenta al momento de crear una base de datos.

A los estudiantes se les solicita los siguientes cambios en el código de la base de datos:

- P0: En un bucle generar posiciones aleatorias dentro del rango establecido para las articulaciones. Llamar la función de la cámara. Calcular la distancia entre el objeto y las coordenadas ideales. No se solicita que se guarden los datos en ningún archivo.
- Pi: luego se les pide que generen un nuevo código donde se formen posiciones aleatorias cercanas a P0. Implementen el código de la cámara para obtener (x, y) y calcular la distancia (h). El proceso se realiza para 2 posibles posiciones.
- Selección de Pi: implementar unas líneas de código al pasado donde tome la opción de pasarse a la posición más cercana al objeto.

- Dataset: Por último, crear un bucle donde se seleccionen en total 5 nuevas posiciones.

## 7. Análisis de resultados

Después de obtener los códigos del Capítulo 5: Creación de la Inteligencia Artificial, se busca tener los siguientes resultados:

- ✓ **Valores MSE:** Valores del error cuadrático medio para cada salida debe ser lo más bajo posible.
- ✓ **R<sup>2</sup>:** Coeficiente de determinación para cada salida debe ser cercano a 1.
- ✓ **Gráfico de Residuos:** Evaluación de la adecuación del gráfico de residuos. Esto significa que no se muestran patrones claros
- ✓ **Valores Reales vs. Predichos:** Calidad del ajuste en la gráfica de valores reales vs. predichos.
- ✓ **Q-Q Plot:** Adecuación de los Q-Q plots, deben acercarse a la línea identidad.
- ✓ **Curva de Aprendizaje:** Evaluación de la convergencia en la curva de aprendizaje.
- ✓ **Error de Entrenamiento y Validación sean bajos:** Ambos errores deben ser lo más bajos posible. Esto que indica un buen rendimiento del modelo tanto en los datos de entrenamiento como en los de validación.
- ✓ **Estabilidad de las Curvas de aprendizaje:** las curvas deben estabilizarse y no mostrar una tendencia creciente en el error de validación.

Todos los resultados obtenidos junto con su análisis se detallarán en este capítulo.

### 7.1 Resultados de los modelos de Inteligencia Artificial

Al realizar los códigos de cada modelo de Inteligencia Artificial, se imprimen los resultados de MSE y R<sup>2</sup>, además que se muestran las gráficas de: curva de aprendizaje, predicciones vs datos reales, gráfico de residuos, gráfica de Q-Q y los errores Q-Q. Los resultados obtenidos se pondrán al terminar cada subsección del análisis respectivo.

Para estos resultados, se empieza generando las predicciones del conjunto de prueba. Con las predicciones listas, se calcula el error cuadrático medio (MSE) y el coeficiente de determinación R<sup>2</sup>. Ambas funciones ya están definidas dentro de

la librería de sklearn, por lo que solamente es llamarlas. Para la red neuronal, se utiliza la función `multioutput='variance_weighted'` en el cálculo de  $R^2$  para considerar ambas variables objetivo en el resultado final. Además, para este modelo se usó una sola vez con doble salida, por lo que el valor de MSE considerando ambas salidas.

Después se crea un DataFrame con los valores reales (los de la base de datos) y los predichos (los de la IA) para ambas variables objetivo y se guardan en un archivo CSV.

Luego se realizan los gráficos:

- Curva de aprendizaje: Estas curvas son importantes para observar el rendimiento del modelo en función al tamaño del conjunto de entrenamiento. Se usa el MSE negativo como métrica para visualizar los valores de entrenamiento forma descendente y facilitar la lectura.
- Valores Reales vs Predichos: Se grafican las curvas de los valores reales vs los predichos para comprar el desempeño del modelo.
- Gráfico de residuos: Luego se grafican las diferencias entre los valores reales y predichos (residuos) para evaluar el ajuste del modelo.
- Gráficos Q-Q: Estos gráficos se crearon para evaluar la normalidad de los residuos.

## **7.2 Curva de aprendizaje: Análisis de resultados**

En el caso de todos los modelos, el error de entrenamiento disminuye a medida que aumenta el número de ejemplos, esto significa que el modelo va aprendiendo conforme pasa la cantidad de datos de entrenamiento.

También, de diferencia entre el error de entrenamiento y el error de validación cruzada se reduce con el aumento de ejemplos, esto indica que los modelos se benefician de la cantidad de datos conforme aumenta y que los modelos no presentan sobre ajuste.

También, en los modelos que presentan una gráfica por salida, se puede ver que el comportamiento para cada una es similar, con la salida v2 empezando el MSE en un valor más alto que el de la salida v1.

En general, estas gráficas muestran que los modelos están aprendiendo y no tiene un sobreajuste significativo.

Estas gráficas fueron las principales para asegurarse que el modelo estaba correctamente programado durante la creación de cada modelo de Inteligencia Artificial.

### 7.2.1 Curvas de aprendizaje: Resultados obtenidos

De la Figura 7.1 a la Figura 7.7, se pueden apreciar las curvas de aprendizaje obtenidas para cada modelo a evaluar. Además, para el modelo de Regresión Polinómica y la Red Neuronal se adjunta una segunda curva, la cual corresponde a un acercamiento en el punto de interés de las curvas.

Para estas curvas lo más importante a notar es que converjan, que la curva de entrenamiento y la de validación cruzada se acerquen entre ellas. Si la curva de validación se empieza a alejar de la curva de entrenamiento, entonces existe un sobreajuste en el modelo de IA.

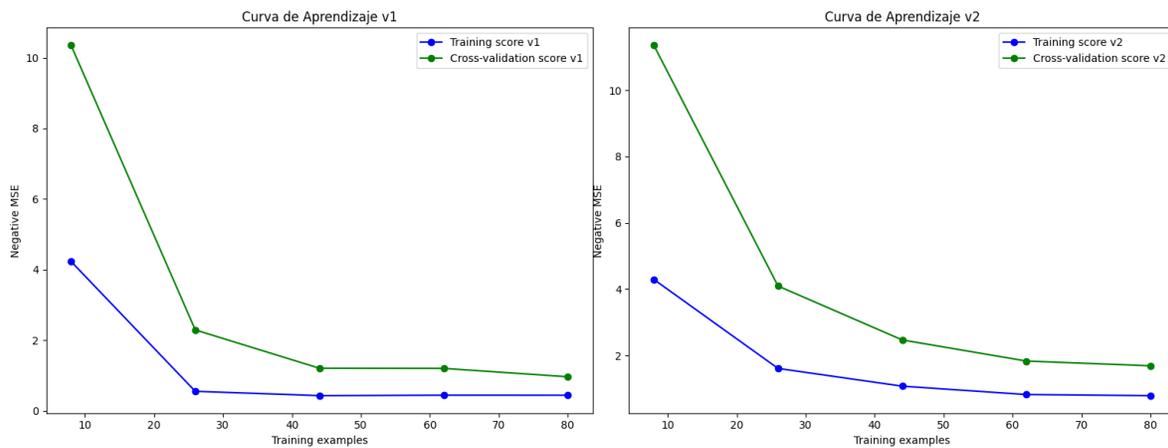


Figura 7.1. Curva de aprendizaje para el modelo de árbol de decisión.

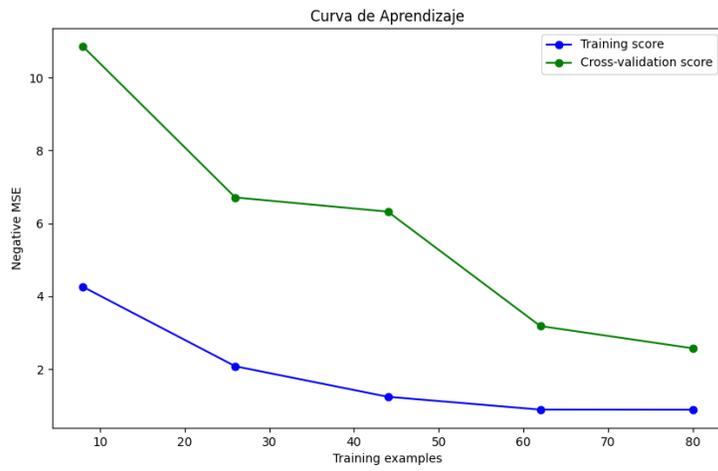


Figura 7.2. Curva de aprendizaje para el modelo de árbol de decisión multisalida.

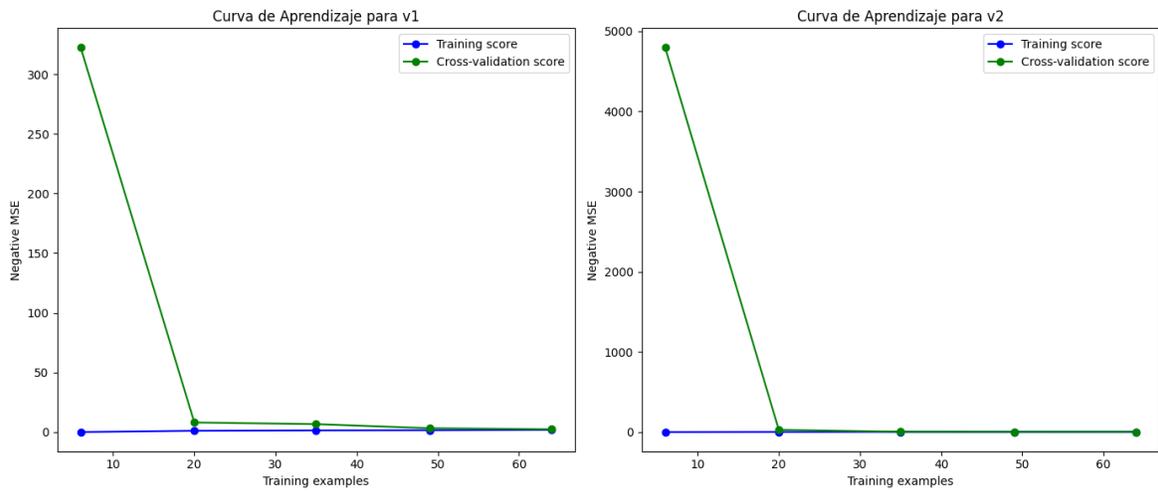


Figura 7.3. Curva de aprendizaje 1: Regresión polinomial.

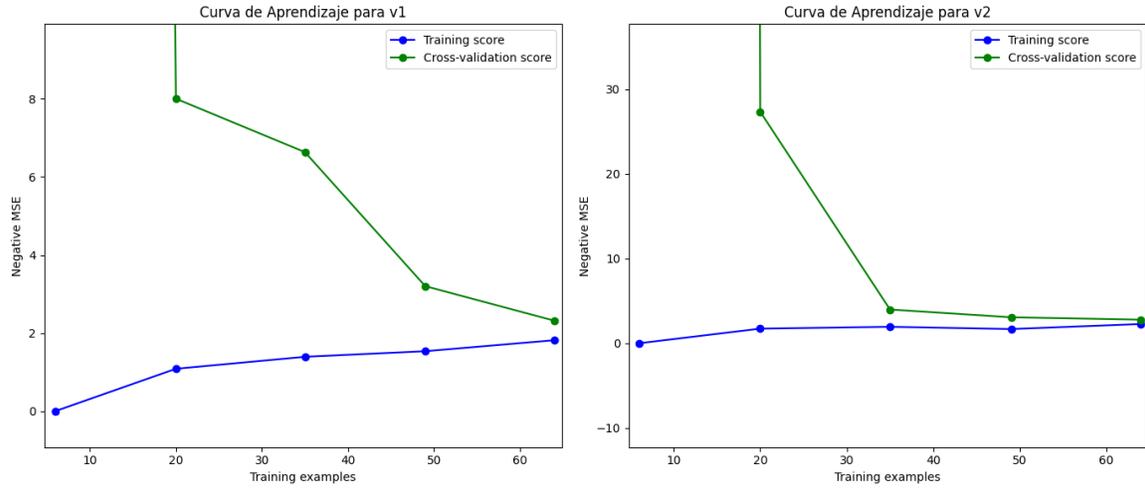


Figura 7.4. Curva de aprendizaje 2: Regresión polinomial (con acercamiento)

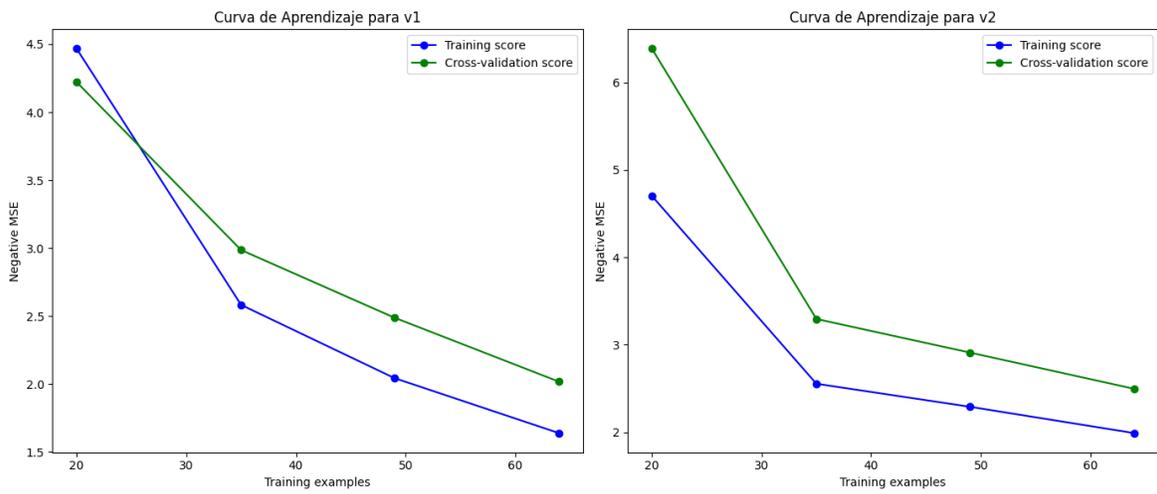


Figura 7.5. Curva de aprendizaje: kNN's.

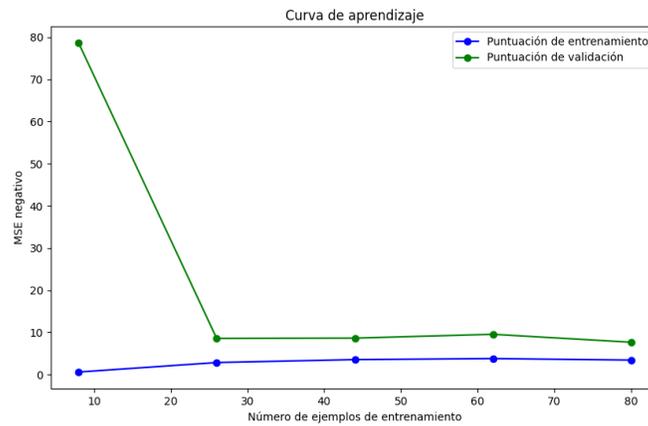


Figura 7.6. Curva de aprendizaje: Red neuronal

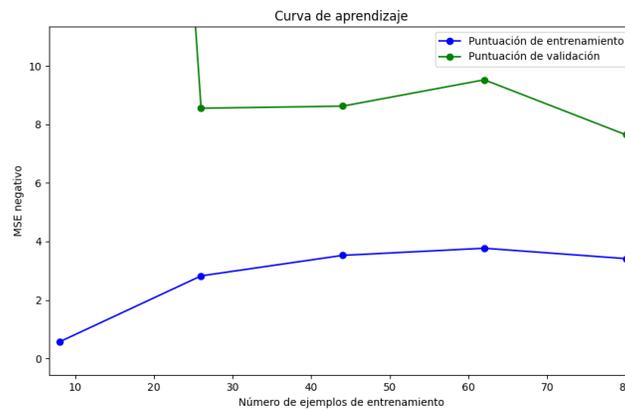


Figura 7.7. Curva de aprendizaje 2: Red neuronal (acercamiento).

### 7.3 Predicciones vs Datos Reales: Análisis de resultados

Este gráfico muestra la relación entre las predicciones de cada modelo y los valores reales obtenidos en la base de datos.

En los gráficos de cada salida para cada modelo, los puntos deberían alinearse en la normal para indicar un buen ajuste.

Al analizar entre modelos, se puede apreciar que hay uno que se ajusta correctamente (Regresión Polinómica), dos que tienen un error, pero igual siguen la normal (kNN's y red neuronal) y los dos modelos de árboles no siguen la función

ideal. También, entre los gráficos de las salidas v1 y v2, se nota que las salidas v2 suelen ser menos precisas que las v1.

### 7.3.1 Predicciones vs Datos reales: Resultados Obtenidos

De la Figura 7.8 a la 7.12 se muestran las gráficas de predicciones vs datos reales. En estas gráficas lo que se espera encontrar es que los datos se alineen a la normal.

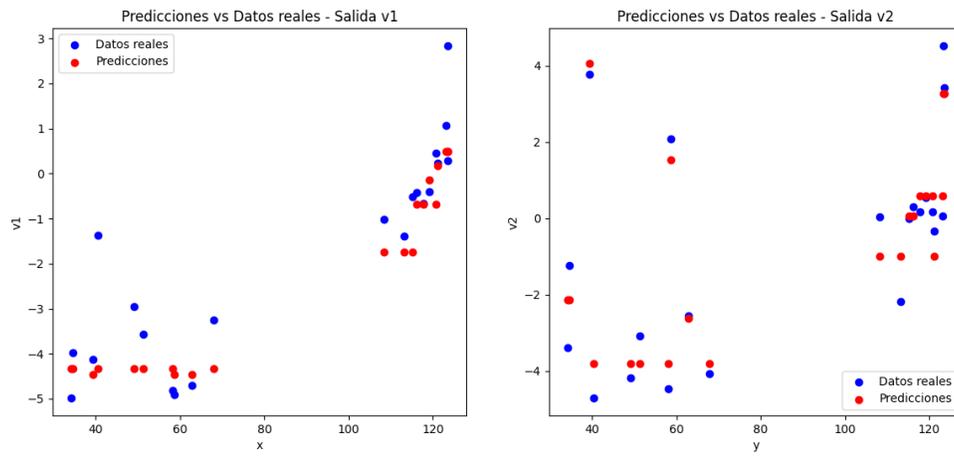


Figura 7.8. Predicciones vs Datos reales: Árbol de decisión

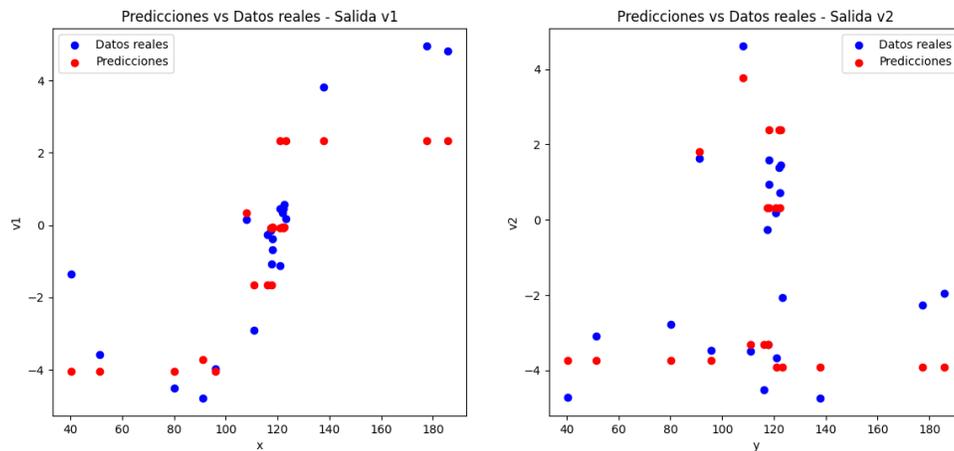


Figura 7.9. Predicciones vs Datos Reales: Árbol de decisión multisalida.

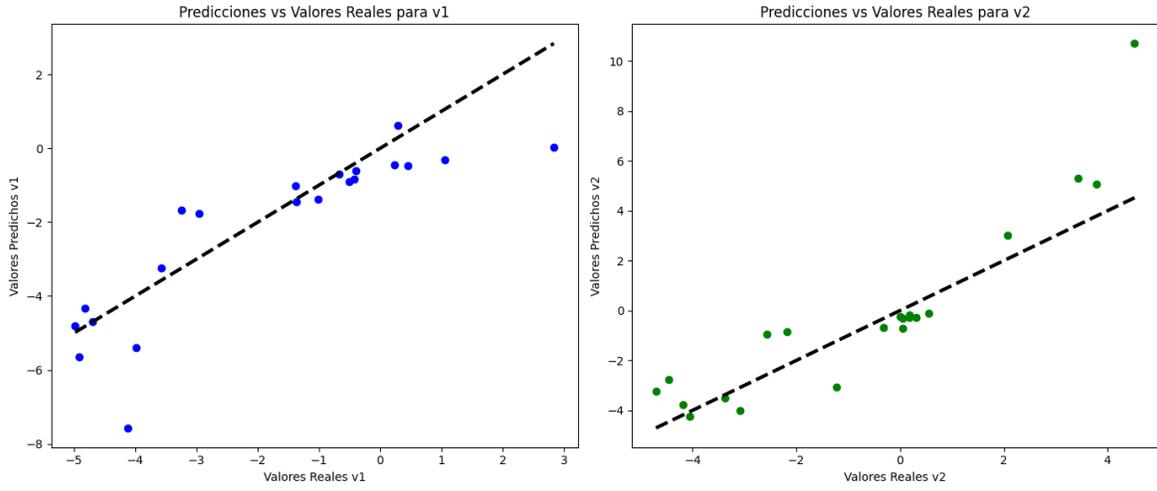


Figura 7.10. Predicciones vs Valores Reales: Regresión Polinomial

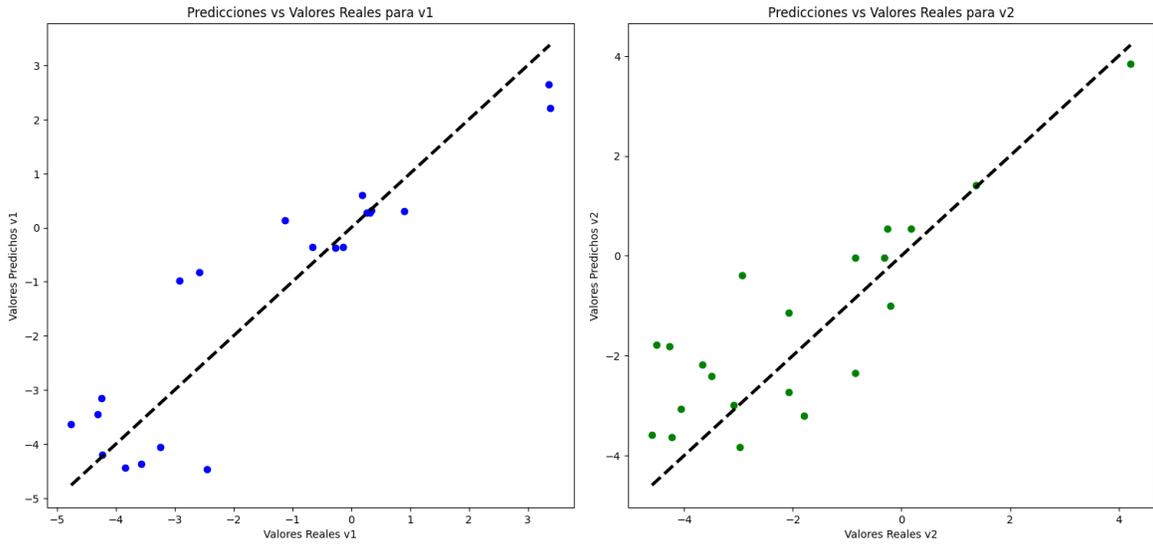


Figura 7.11. Predicciones vs Valores reales: kNN's.

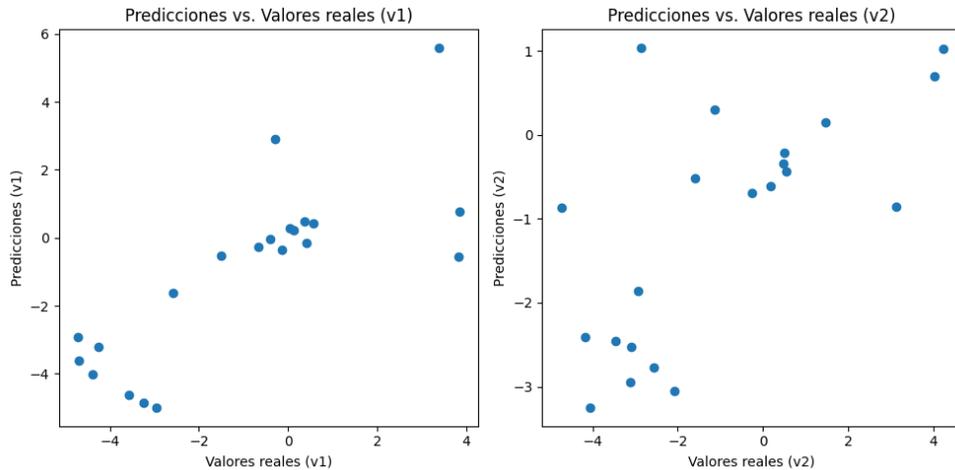


Figura 7.12. Predicciones vs Valores reales: Red Neuronal

## 7.4 Gráficos de residuos: Análisis de resultados

Este gráfico muestra los residuos entre las salidas v1 y v2 en función al índice de muestra. Lo que se busca en este gráfico, es que los valores estén distribuidos alrededor de 0. Además, no debería haber patrones claros, para indicar que los modelos están razonablemente bien especificados.

Para este gráfico todos los modelos cumplen con lo que se busca, por lo que se pasará al siguiente gráfico.

### 7.4.1 Gráfico de residuos: Resultados obtenidos

De la Figura 7.13 a la Figura 7.17 se encuentran los resultados de los gráficos de residuos. En estas gráficas se busca que no haya ningún patrón definido y los valores se encuentren dispersos.

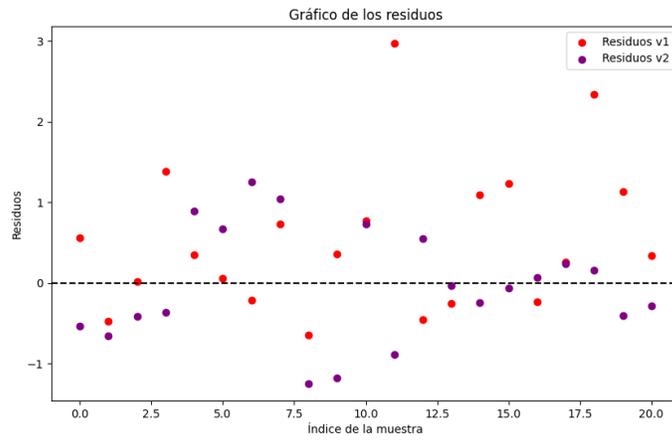


Figura 7.13. Gráfico de residuos: Árbol de decisión.

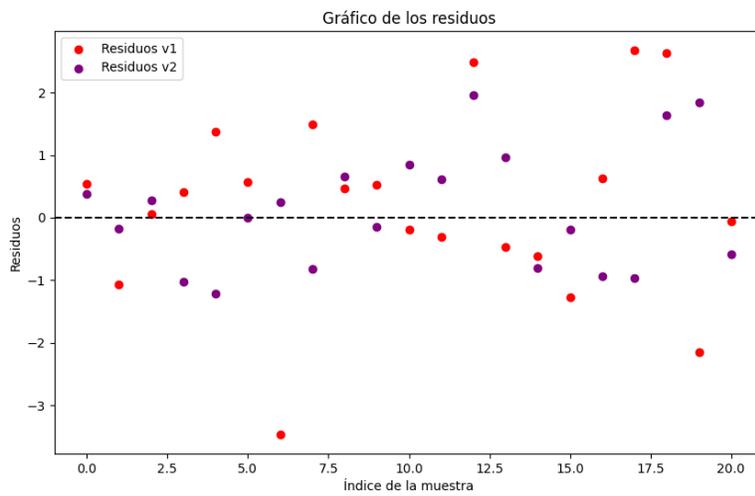


Figura 7.14. Gráfico de residuos: Árbol de decisión multisalida.

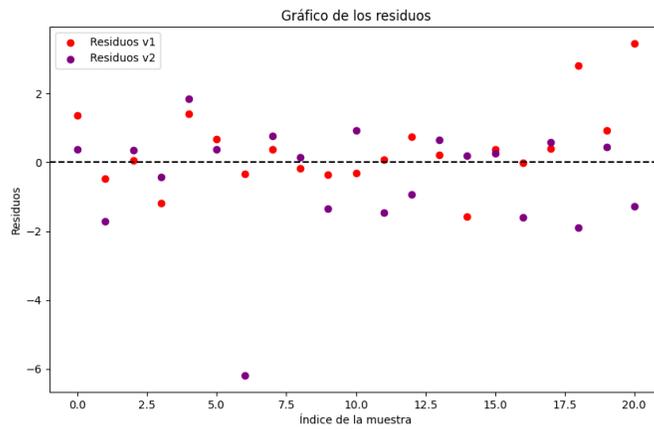


Figura 7.15. Gráfico de residuos: Regresión Polinomial

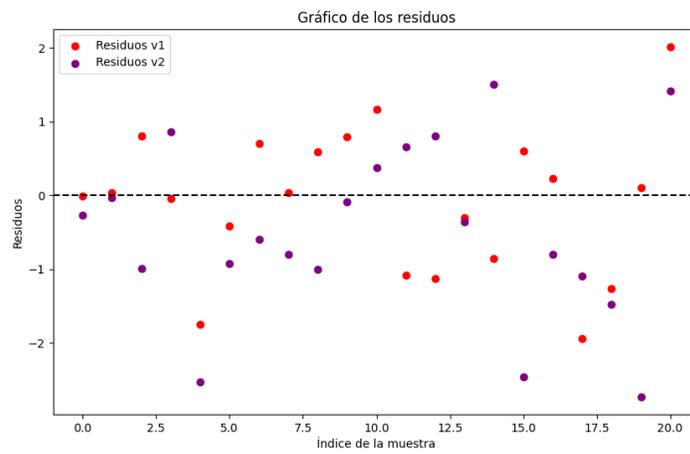


Figura 7.16. Gráfico de residuos: kNN's.

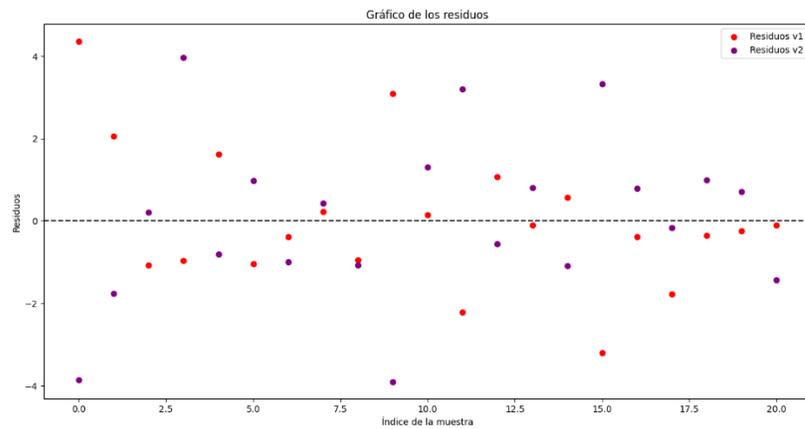


Figura 7.17. Gráfico de residuos: red neuronal

## 7.5 Gráfica de Q-Q: Análisis de resultados

Estos gráficos comparan la distribución de los residuos de cada modelo con una distribución normal, siendo un indicador que el modelo está bien especificado.

En esta gráfica, se puede ver que todos los residuos siguen la normal, aunque con algún error asociado. En general, todos los modelos parecen ser apropiados, con el árbol de decisión multisalida y kNN's siendo los mejores dos.

### 7.5.1 Gráfica de Q-Q: Resultados obtenidos

De la Figura 7.18 a la Figura 7.22 se encuentran las gráficas de QQ. Para estas gráficas se buscan que los datos se acerquen a la línea de identidad.

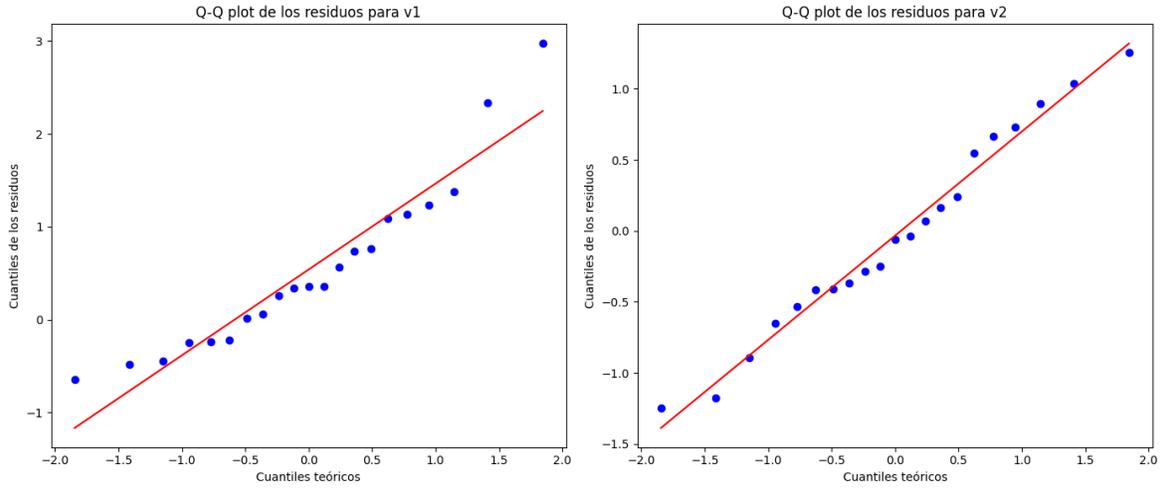


Figura 7.18. Gráfica Q-Q: Árbol de decisión.

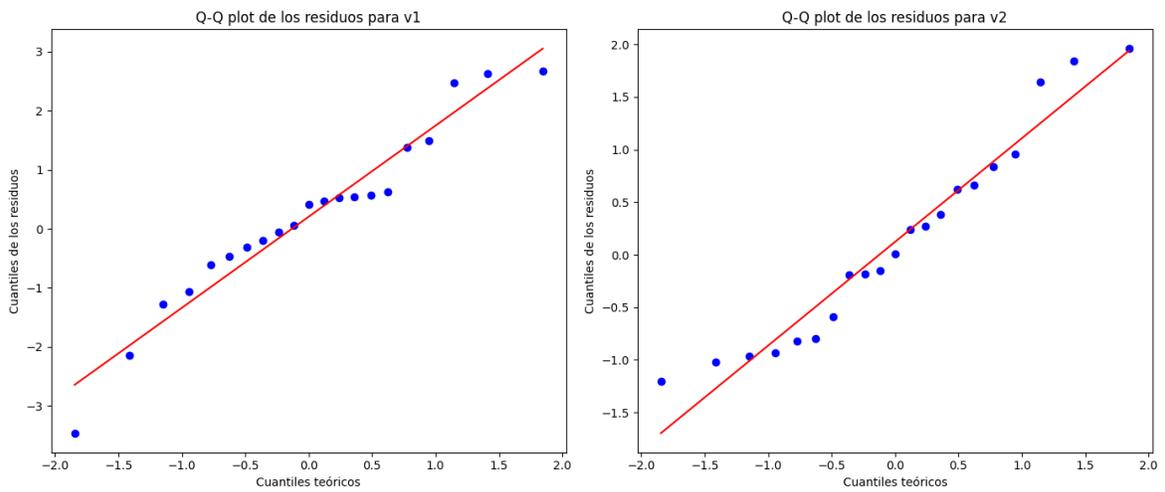


Figura 7.19. Gráfica Q-Q: Árbol de decisión multisalida.

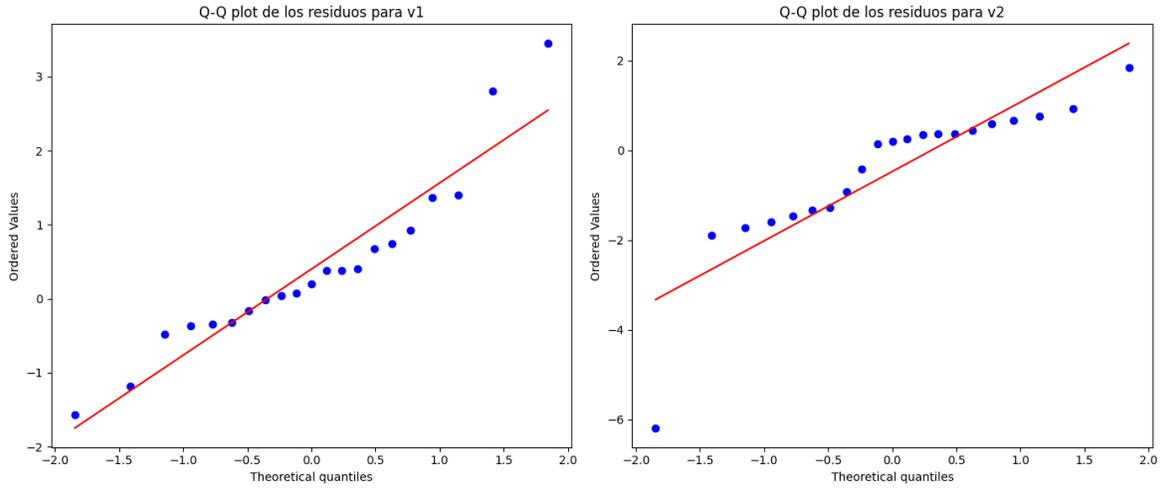


Figura 7.20. Gráfica Q-Q: Regresión Polinomial.

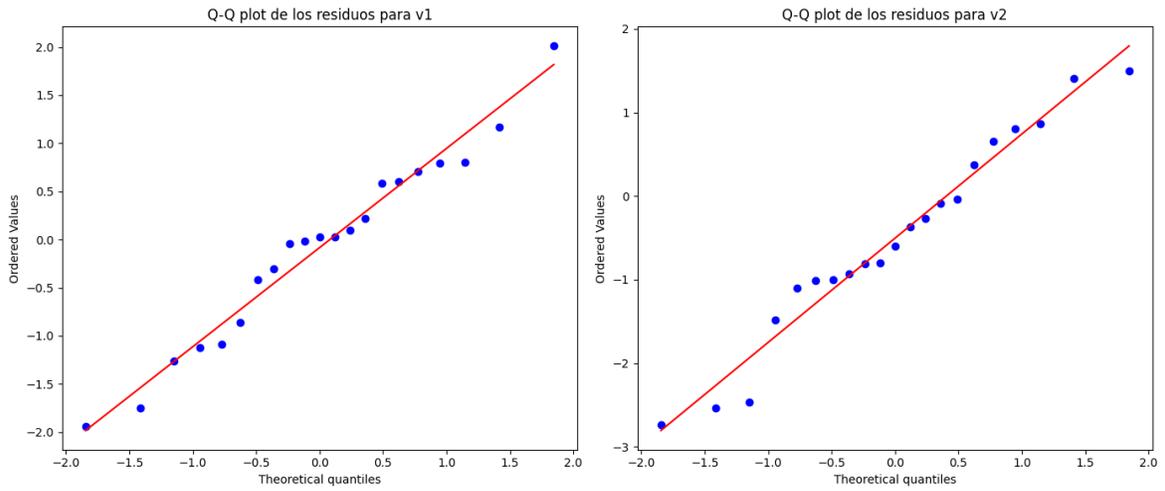


Figura 7.21. Gráfica Q-Q: kNN's.

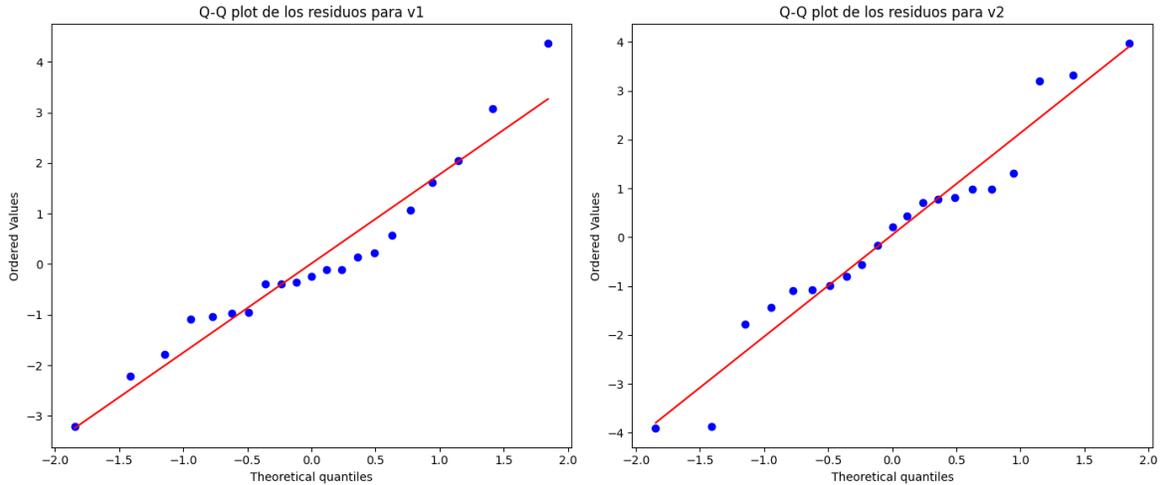


Figura 7.22. Gráfica de Q-Q: red neuronal.

## 7.6 Errores de Distancia: Análisis de resultados

Este gráfico muestra las distancias entre los puntos obtenidos en Q-Q y la normal, para poder analizar el error de ellos y es una forma para analizar mejor la gráfica de Q-Q. Se busca que los errores sean lo mínimo posible

Dado que para este análisis se busca el menor error posible, se considera una puntuación binaria para este criterio. El resultado es que el modelo kNN's es el mejor ya que es el que cuenta con el menor rango de error en la gráfica.

### 7.6.1 Errores Q-Q: Resultados obtenidos

De la Figura 7.23 a la Figura 7.27 se observan las gráficas de los errores de los valores QQ.

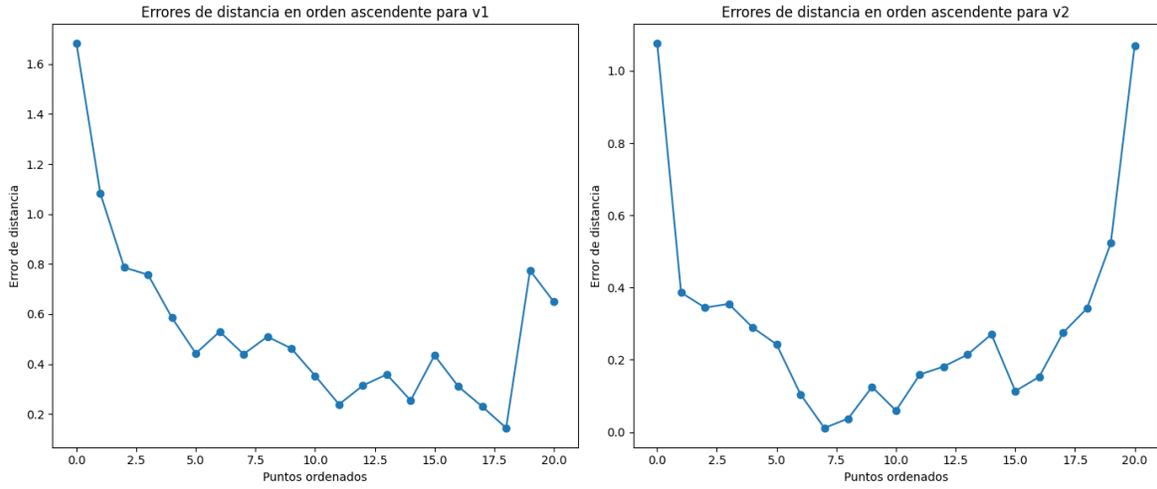


Figura 7.23. Errores de distancia: Árbol de decisión.

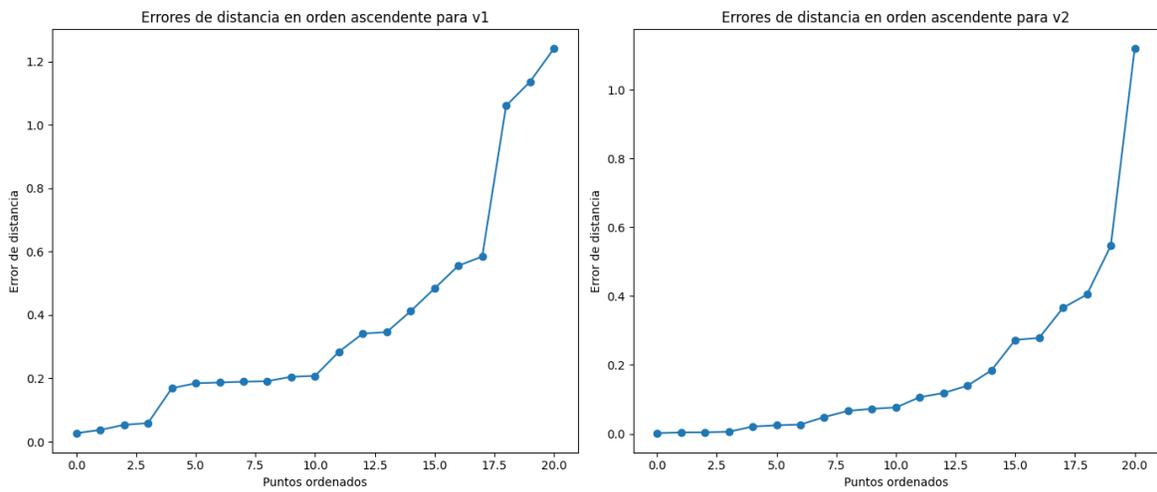


Figura 7.24. Errores de distancia: Árbol de decisión multisalida.

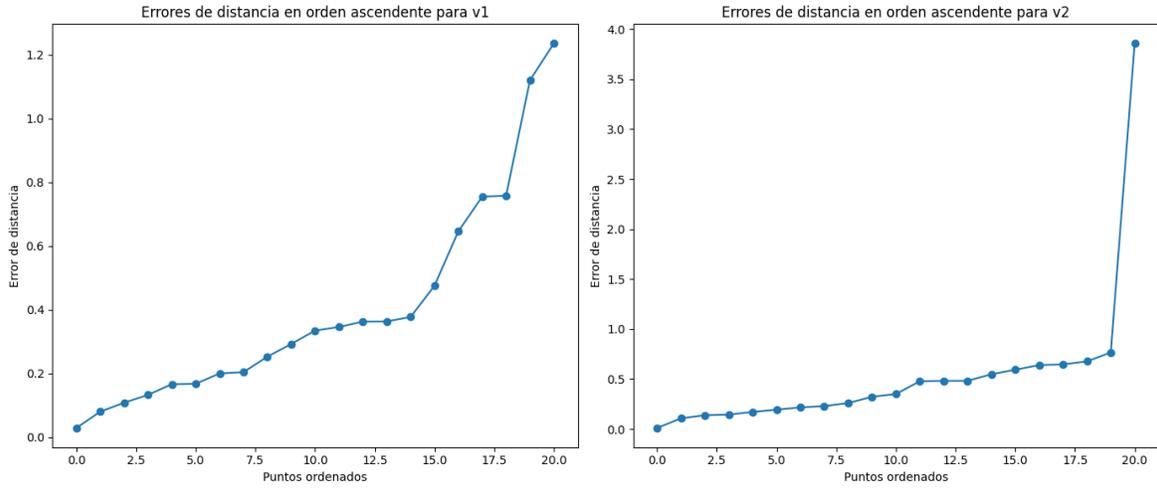


Figura 7.25. Errores de distancia: Regresión polinomial.

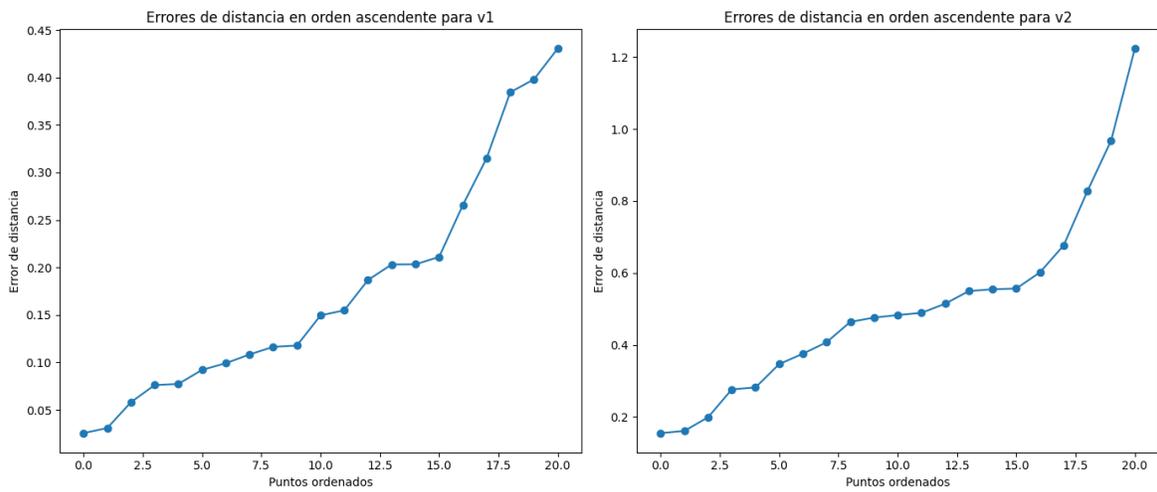


Figura 7.26. Errores de distancia: kNN's.

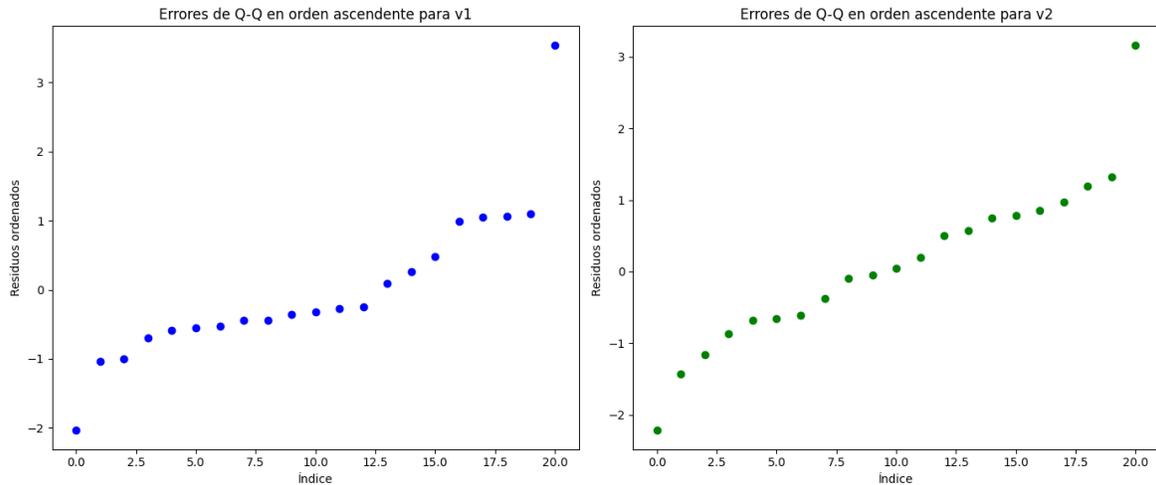


Figura 7.27. Errores de distancia: red neuronal.

## 7.7 Resumen de análisis de los modelos de Inteligencia Artificial: Análisis de resultados

Considerando estos criterios de evaluación, para facilidad de estudio se desarrolló una matriz para evaluar cada concepto la cual se encuentra en la Tabla 6.1.

Tabla 7.1. Matriz de resultados de cada modelo de Inteligencia Artificial.

Resumen de resultados de los modelos de Inteligencia Artificial					
	Árbol de Decisión	Árbol de Decisión Mult.	k-NN's	Regresión Polinomial	Red Neuronal
<b>MSE Salida 1</b>	1,1013	2.1931	0.9444	1.4483	3.3883
<b>MSE Salida 2</b>	0.4718	0.8908	1.6320	2.8687	
<b>R<sup>2</sup> Salida 1</b>	0.7801	0.6814	0.8351	0.7108	0.5053
<b>R<sup>2</sup> Salida 2</b>	0.9341	0.8658	0.6637	0.5995	
<b>Gráfico de Residuos</b>	Adecuado	Adecuado	Adecuado	Adecuado	Adecuado
<b>Valores Reales vs Predichos</b>	Malo	Malo	Regular	Bueno	Regular
<b>Q-Q Plot</b>	Regular	Bueno	Bueno	Bueno	Bueno
<b>Curva de Aprendizaje</b>	Convergente	No	Convergente	Convergente	No
<b>Error</b>	Malo	Malo	Bueno	Malo	Malo

Además, se marcaron en verde los mejores resultados. Según se aprecian en los resultados del Capítulo 5: Creación de la Inteligencia Artificial.

### 1.1.1 Resultados de MSE y R<sup>2</sup>: Resultados obtenidos

En la matriz de la Tabla 7.2, se pueden apreciar los resultados obtenidos para cada modelo.

Tabla 7.2. Resultados obtenidos para los modelos de Inteligencia Artificial usados.

<b>Resultados MSE y R<sup>2</sup> para los modelos de Inteligencia Artificial</b>					
	<b>Árbol de Decisión</b>	<b>Árbol de Decisión Mult.</b>	<b>k-NN's</b>	<b>Regresión Polinomial</b>	<b>Red Neuronal</b>
<b>MSE Salida 1</b>	1,1013	2.1931	0.94 44	1.4483	3.3883
<b>MSE Salida 2</b>	0.4718	0.8908	1.63 20	2.8687	
<b>R<sup>2</sup> Salida 1</b>	0.7801	0.6814	0.83 51	0.7108	0.5053
<b>R<sup>2</sup> Salida 2</b>	0.9341	0.8658	0.66 37	0.5995	

## 7.8 Movimientos del robot simulado y real

Durante el proceso de desarrollo de las lecciones, los códigos establecidos se verificaron tanto con la simulación como en el robot físico. A continuación, se comentarán los resultados obtenidos para cada lección.

Cabe destacar que, según lo obtenido, la simulación desarrollada cumple con al menos 90% de los movimientos esperados para cada lección de aprendizaje. También, se nota que el robot real cumple con al menos 90% de los movimientos establecidos por el programa realizado en cada lección de aprendizaje.

Por último, hacer el comentario que en la cuarta lección no se implementa ningún movimiento del robot (ni simulado ni real), ya que únicamente se entrena el modelo de Inteligencia Artificial. También hay que recordar que, en la segunda y tercera lección, se hace el movimiento del robot únicamente por simulación.

### 7.8.1 Primera lección: movimientos del robot simulado y real

Para la primera lección, los movimientos del robot en simulador y real constaban de posiciones ya establecidas. Las posiciones se brindaron en las

instrucciones con coordenadas XYZ y sus respectivos ángulos o con los ángulos correspondientes a cada articulación del robot. Dado que son posiciones fijas, el robot simulado y real cumplió con el 100% de los movimientos esperados para la primera lección de aprendizaje.

#### 7.8.2 Segunda lección: movimientos del robot simulado

La segunda lección se basa en el aprendizaje de sistemas de visión para la aplicación del proyecto, usando el sensor en el simulador y no se implementa el robot real. Sin embargo, al final de la lección para crear una aplicación con la cámara, se realizan en el simulador los movimientos básicos del robot en dirección al objeto meta (la torre de ajedrez). Al analizar los movimientos efectuados tanto por el robot como la simulación, el robot efectivamente se acerca a la pieza con cada movimiento. De esta forma, el robot simulado cumple con el 100% de los movimientos esperados para la segunda lección de aprendizaje.

#### 7.8.3 Tercera lección: movimientos del robot simulado

En esta lección se creó la base de datos, al igual que la lección pasada, esta lección solo posee la implementación en el simulador. Además, se realizó un código principal para la creación de la base, todos datos fueron almacenados en un .CSV. Cada archivo contiene información importante para analizar los resultados de movimientos de esta lección, sin embargo, para representar los movimientos definidos por el programa para el robot y la trayectoria a seguir se mostrará una parte de los datos guardados en el archivo “*DataBaseTrayectoria*” y “*DataBaseMinimos*”. Estos datos se muestran en la Tabla 7.3. para esto se muestran los valores de distancia entre el objeto y el punto ideal y se usó una escala de color para apreciar cómo la distancia varía. En la tabla se encuentran los ángulos de las articulaciones 1 y 2, la distancia en (x,y), la distancia del objeto (hipotenusa de x,y) y la variación de las articulaciones que se toma para llegar a la siguiente posición. En la tabla se puede observar cómo se realiza un movimiento donde la distancia disminuye donde se busca un valor menor a 2 píxeles, cuando se llega a este valor, se pasa a otra posición aleatoria y comienza el acercamiento.

Dado que en todos los movimientos el robot se acerca al objeto, se demuestra que el robot simulado cumple con el 100% de los movimientos esperados para la segunda lección de aprendizaje.

Tabla 7.3. Parte de los datos obtenidos para la trayectoria del robot.

Datos de la trayectoria del robot						
Articulación 1	Articulación 2	Distancia total	Distancia x	Distancia y	Variación de articulación 1	Variación de articulación 2
-12,5411	30,17721	99,58482	51,3021	277,0953	-3,57126	-3,08229
-16,1124	27,09493	75,26816	71,00909	262,1418	-4,69121	-4,64456
-20,8036	22,45037	45,11059	95,88085	243,1213	-3,972	-3,46792
-24,7756	18,98245	25,96859	115,8791	230,6395	-1,32052	-4,83509
-26,0961	14,14736	10,41739	122,3822	215,1414	0,901701	-4,2668
-25,1944	9,880564	3,84969	118,0114	201,7037	-0,67907	1,596414
-25,8735	11,47698	2,117024	121,2389	206,7167	0,236122	-0,32739
-25,6373	11,14958	0,711019	120,0782	205,7067	0	0
11,51182	26,85163	126,714	14,79316	275,6254	-4,25894	-2,53677
7,252876	24,31486	99,86779	36,65009	260,0124	-4,90276	2,786942
2,350112	27,1018	83,62104	60,18662	263,4366	-3,75576	-3,40316
-1,40565	23,69864	58,70236	80,37126	248,3074	-3,84904	-4,59348
-5,25468	19,10515	32,5278	100,3295	230,9062	-4,16857	-3,99959
-9,42325	15,10557	12,47909	120,9254	217,4447	-1,13084	-3,66967
-10,5541	11,4359	6,432063	126,3581	205,9725	0,371518	-1,14126
-10,1826	10,29463	5,277231	124,5621	202,3475	0,846078	0,736945
-9,3365	11,03158	0,643522	120,5337	204,6404	0	0
-6,81412	2,279148	114,0999	225,9952	162,765	4,957224	1,711434
-1,85689	3,990582	93,42146	201,6962	159,6866	4,723599	3,874757
2,866704	7,865339	69,211	176,7527	165,3856	3,109733	3,447074
5,976438	11,31241	51,44514	160,1136	172,7898	3,375505	4,219388
9,351943	15,5318	31,05298	142,0439	183,1287	4,196597	4,426843
13,54854	19,95864	9,51346	119,4441	195,5028	-0,13734	3,619823
13,4112	23,57847	1,938505	120,165	206,9315	0	0
-9,92687	46,50629	138,9756	230,5983	289,156	4,896823	-3,18513
-5,03004	43,32116	108,6509	202,7446	275,4155	3,352468	-4,22659
-1,67757	39,09457	76,27784	177,4962	255,1248	4,954647	-2,26949
3,277072	36,82509	45,52032	144,6591	243,2626	1,722304	-4,05982
4,999376	32,76526	26,96799	133,0301	228,6112	0,465666	-4,4638
5,465042	28,30147	13,16861	129,7425	213,8598	2,169802	-3,24609
7,634843	25,05537	3,596722	116,8528	203,2589	-0,84985	0,793666
6,784989	25,84904	2,008344	121,8781	205,7114	0,176089	0,314697
6,961078	26,16374	1,933675	120,842	206,7407	0	0

-9,71579	36,18372	73,81102	58,13607	245,2607	-4,8189	-4,46673
-14,5347	31,71699	36,68001	89,90101	225,9636	-4,24359	-2,07331
-18,7783	29,64368	13,20344	116,0869	217,6102	-0,26976	-4,51373
-19,048	25,12995	2,87472	117,8342	203,1097	0,025569	0,141581
-19,0225	25,27153	2,735254	117,734	203,468	-0,66462	0,173917
-19,6871	25,44545	1,808676	121,5315	204,0378	0,130257	0,467315
-19,5568	25,91277	1,013783	120,85	205,5525	0	0
-0,74311	30,12095	88,56806	185,6319	264,4706	4,808904	-1,95088
4,065793	28,17006	61,83541	159,0277	252,9631	3,842867	-2,86313
7,90866	25,30693	40,2066	137,8836	241,0104	3,821544	-4,73553
11,7302	20,57141	19,90566	117,7173	224,7743	-1,07954	-3,30066
10,65067	17,27074	9,712987	123,23	214,1602	0,183661	-2,06962
10,83433	15,20112	3,595553	122,3886	207,6875	-0,16963	-1,39126
10,66469	13,80987	3,565929	123,118	203,2697	1,062335	0,058745
11,72703	13,86861	2,729875	117,7303	203,4831	-0,90601	0,265462
10,82102	14,13407	2,51347	122,4206	204,3231	0,037121	0,499524
10,85814	14,6336	2,315564	122,1599	205,8347	0,313107	-0,85197
11,17125	13,78163	1,896667	120,5391	203,1816	0	0
-4,20848	5,84519	12,97831	109,5262	212,6639	-2,58015	-2,92984
-6,78862	2,915349	1,788855	120,7989	203,3994	0,448781	0,181614
-6,33984	3,096963	1,556817	118,8092	203,9971	0	0
-7,81178	10,87982	29,62972	91,06094	198,6399	-4,78855	1,637812
-12,6003	12,51763	6,134701	114,5568	202,1705	-1,53204	0,634545
-14,1324	13,15218	2,355044	122,15	204,0389	0,462592	0,710729
-13,6698	13,86291	1,255041	119,7576	206,2314	0	0

#### 7.8.4 Quinta lección: movimientos del robot simulado y real

En esta lección se realizó la implementación del modelo de Inteligencia Artificial programado. Los movimientos esperados del robot para esta lección son el acercarse a la pieza usando el modelo de vecinos cercanos. Tanto para el simulador como el robot real, cada vez que se ejecutó el código, el robot fue capaz de llegar a la pieza. Por lo tanto, el simulador y el robot real cumplen con el 100% de los movimientos esperados para cada lección de aprendizaje.

### 7.9 Población muestra

Para este análisis, se desarrollaron las lecciones con una población muestra de 1 persona. A pesar de tener en cuenta un grupo de estudiantes de colegio para aplicar la prueba, el cual había realizado las primeras pruebas con el “Robobo” (la parte previa de las lecciones realizadas), el cronograma lamentablemente no logró

coincidir con las fechas de realización del proyecto. Esto porque las fechas fueron aplazadas dado que los estudiantes requirieron un mayor tiempo del esperado para desarrollar de las últimas lecciones del Robobo, además que se realizó un viaje académico en el colegio. Lo que provocó que los estudiantes llegaran a las evaluaciones finales del año lectivo y salieran a las vacaciones antes de implementar las lecciones propuestas. La unión de estos factores más la limitación que el total de cada lección planteada es de 12 horas y el cronograma del proyecto que finaliza antes del nuevo año académico, no fue posible implementar la población muestra prevista.

También, se consideró hacer un grupo de muestra dentro del Grupo Integrado de Ingeniería, sin embargo, los presentes ya eran graduados de la universidad, incluso estudiantes de maestría o con doctorados, los cuales además trabajan en el grupo con un enfoque de Inteligencia Artificial y Robótica (conociendo más de lo que abarcan las lecciones) o Ingeniería Naval (sin conocimientos de programación). Por lo que no cumplen con los requisitos para ser población muestra.

La cantidad de la población muestra no es ideal. Sin embargo, se trató de asemejar los conocimientos de los estudiantes, considerando que no debería tener grandes conocimientos sobre Inteligencia Artificial y Robótica. La posibilidad de usar a esta persona es porque desde el comienzo se presentó la oportunidad para analizar las lecciones antes de revisarlas con el cliente. Por lo tanto, el análisis creará posibles indicadores si las lecciones están bien o deben implementar algún cambio.

La persona usada para la población muestra es un hombre de 23 años, con la secundaria completa y estudiante de Ingeniería Electromecánica en el Tecnológico de Costa Rica, con una pasantía en una constructora. Además, tiene un manejo del inglés regular, poseía conocimientos limitados de programación (dado que llevó un curso básico hace unos años) y sin conocimientos previos de Robótica, Sistemas de Visión ni Inteligencia Artificial, y con un interés de aprender del tema regular.

A la población muestra se le aplicaron las evaluaciones creadas durante el desarrollo de las lecciones (las cuales se encuentran desde el Anexo C al Anexo F). Para lo cual, parte de la evaluación constó de que se cumplieran los objetivos de cada actividad, se consideró el tiempo de duración de las actividades y por último un cuestionario sobre teoría. Estas evaluaciones son un anexo de las lecciones creadas para que los profesores puedan aplicarlas a los estudiantes y constan de 3 partes: evaluación del desempeño de las tareas para cada actividad (40%), evaluación general del estudiante (35%) y el cuestionario de lo aprendido (25%).

Para la evaluación de las lecciones con la población muestra se evaluaron cuatro lecciones de aprendizaje, esto debido al tiempo disponible de la población para ejecutar las lecciones. Se dedicó 40 horas en un plazo de 3 meses en los tiempos libres para la realización de 4 lecciones y sus respectivos cuestionarios.

Las actividades de cada lección se evaluaron de manera binaria solo para asegurar que las actividades fueran realizables y alcanzaran el objetivo, mientras que para cada cuestionario se realizó la evaluación planteada del 25%.

Después de realizar las lecciones a la población muestra fue capaz de realizar todas las actividades propuestas, por lo que se concluye que el nivel desarrollado era el adecuado.

Por último, para los cuestionarios se desarrollará un análisis por lección, pero en la Tabla 7.4 se muestra un resumen de los resultados obtenidos en las evaluaciones de cada lección.

Es importante destacar que en la mayoría de las lecciones (la segunda, tercera y cuarta lección) el nivel de aprendizaje fue mayor a 70%. Además, al sumar los puntos obtenidos (dado que se evaluaron 4 lecciones, cada una equivale a 25% del total) se tiene un aprendizaje total de 77.5%. Con esto se demuestra el cumplimiento del objetivo del proyecto ya que el nivel de aprendizaje fue mayor a 70% en el grupo de muestra validado mediante la herramienta de evaluación de cuestionarios.

Tabla 7.4. Tabla resumen de evaluación a la población muestra

<b>Resumen de evaluación</b>		
<b>Lección</b>	<b>Puntaje (Total 25 pts.)</b>	<b>Nota (sobre 100)</b>

<b>Primera lección</b>	16 puntos	64
<b>Segunda lección</b>	24 puntos	96
<b>Tercera lección</b>	17,5 puntos	70
<b>Cuarta lección</b>	20 puntos	80

#### 7.9.2 Primera Lección: resultados de la población muestra

Para esta lección se aplicó un cuestionario con 5 preguntas. De las cuales 2 eran de selección única, 2 de respuesta corta y la última de desarrollo. Este se puede ver en el Anexo C.

Tras la evaluación, la población muestra obtuvo un 64% de la nota (16 puntos de 25). Con dos preguntas con gran fallo, esto indicó que había que reforzar esas dos áreas en las lecciones. Estas áreas fueron la cantidad de grados de libertad de la imagen de muestra y las 6 variables que afectan los marcos de referencia.

#### 7.9.3 Segunda Lección: resultados de la población muestra

El cuestionario de la segunda lección constaba de 6 preguntas. Donde habían 2 de selección única y 4 de respuesta breve. El cuestionario se puede ver en el Anexo D.

Tras la evaluación, se consiguió una puntuación del 96% (24 puntos de 25). Esta nota indica que esta lección está correctamente desarrollada y logra el objetivo de aprendizaje.

#### 7.9.4 Tercera Lección: resultados de la población muestra

En la tercera lección, el cuestionario tenía 5 preguntas. La primera de selección única, dos de respuesta corta y dos más de desarrollo. El cuestionario se puede ver en el Anexo E.

En este caso, la nota obtenida fue del 70% con leves faltas durante la solución (17.5 de 25 puntos). Por lo que es necesario un análisis más profundo con una población muestra mayor para determinar si la lección tiene que adquirir ajustes o si cumple con la enseñanza que se desea conseguir.

#### 7.9.5 Cuarta Lección: resultados de la población muestra

El cuestionario de esta lección consta de 5 preguntas. La primera de selección única, dos de respuesta corta y dos más de desarrollo. EL cuestionario se puede ver en el Anexo F.

En la nota final, la población muestra tuvo una nota de 80% (20 puntos de 25). Esto indica que esta lección es satisfactoria y cumple con el objetivo.

## 8. Análisis económico

Este proyecto consistió en la creación de un material lectivo con actividades interactivas para estudiantes preuniversitarios basado en Inteligencia Artificial y robótica. El proyecto incluyó la compra de un brazo robótico y una cámara para realizar los experimentos de las lecciones, lo que influyó en el análisis económico ya que son activos que se compraron por la universidad. Estos activos, a pesar de tener una depreciación, mantienen un valor económico para la universidad. En la Tabla 8.1 se encuentra el resumen de los gastos considerados.

Además, por el contexto del proyecto, no se esperan ingresos monetarios directos, ya que el objetivo principal es educativo. Sin embargo, el mayor impacto al aplicar las lecciones es la mejora del capital humano. También otros beneficios son la mejora en el nivel de escolaridad, preparación educativa para los estudiantes, innovación de la enseñanza y aumento en el prestigio de la universidad. En la Tabla 8.2 se encuentra el resumen de los beneficios que el proyecto genera.

Tabla 8.1. Resumen de análisis económico del proyecto (Costos e inversiones)

<b>Gastos del proyecto</b>				
<b>Descripción</b>	<b>Cantidad</b>	<b>Valor por unidad (euros)</b>	<b>Subtotal (euros)</b>	<b>Disponible en la empresa previa al proyecto</b>
<b>Materiales y Herramientas</b>				
<b>Brazo robótico</b>	1	2 000	2 000	NO
<b>Cámara</b>	1	66.5	66.5	NO
<b>Arduino UNO</b>	1	37.2	37.2	SI
<b>Simulador</b>	1	200	200	NO
<b>Computadora y accesorios</b>	1	1000	1000	SI
<b>Servicios Generales</b>				
<b>Remuneración económica</b>	6	400	2 400	---
<b>Alimentación</b>	120	7.5	900	---
<b>Imprevistos</b>	1	30	30	---
<b>Total</b>			<b>6633.7</b>	

Tabla 8.2. Resumen de análisis económico del proyecto (Beneficios).

Beneficios del proyecto	
Beneficio	Descripción
Mejora en el Capital Humano	Mejora las habilidades de los alumnos para poder producir, permitiendo una ventaja en el mercado laboral.
Mejora en el Nivel de Escolaridad	Genera un grado superior de escolaridad a los estudiantes ya que conocen las bases de la materia.
Preparación educativa	El alumno empieza a conocer sobre ámbitos tecnológicos relacionados, preparándolo para continuar sus estudios en una carrera a fin.
Innovación de la enseñanza	Mejora la enseñanza de colegios al implementar lecciones con tecnología avanzada.
Aumento en el prestigio de la universidad	La universidad se posiciona como líder de la innovación educativa, lo que aumenta su reputación e interés para nuevos estudiantes.

### 8.1 Recursos Empleados para el proyecto

Dentro de los costos del proyecto se incluyen las herramientas y equipo necesarios para la realización del proyecto. Dentro del equipo comprado se incluye un brazo robótico que cuenta con varios efectores finales (el *Magician* de **Dobot**) y un sensor (la cámara Huskylens). La compra de este equipo no solo significa un gasto, pero es además un activo para la institución, ya que tanto el robot como la cámara se pueden usar a lo largo de varios años, lo que aporta un valor en su uso continuo.

Dentro del equipo usado, se aprovechó la computadora portátil, teclado externo, ratón y pantalla conectada por HDMI. Este equipo se necesitó para la creación de los códigos y el uso del simulador y control del robot. Además, para el uso de la cámara se necesitó un Arduino UNO el cual la universidad ya tenía.

En cuanto a las herramientas, todo el software usado fue gratuito. Para el simulador (RoboDK), el cliente prefirió mantener la versión gratuita, la cual limita pocas herramientas. Para el control del robot se usó el software proporcionado por el productor (la aplicación DobotLab), en este caso, no existe versión paga. Por último, para el uso de la cámara,

Por último, dentro de los gastos del proyecto se debe considerar la remuneración económica que tiene el personal creador del proyecto. En este caso, el equipo consta de una persona para el tiempo de desarrollo del proyecto que fueron 6 meses y se consideró la alimentación de almuerzo para esos días con el precio de la cafetería institucional.

## **8.2 Recursos generados para el proyecto**

Como se mencionó en el inicio de este capítulo, el proyecto no tiene un beneficio económico directo, si no consta de múltiples beneficios hacia el cliente (la universidad) como para la población meta (los estudiantes). A continuación, se detallan los beneficios comentados en la Tabla 8.2.

Se puede enseñar Inteligencia Artificial sin grandes equipos, pero al usar tecnología avanzada (como lo son los brazos robóticos) en la enseñanza colegial, se proporciona a los estudiantes una experiencia interactiva y moderna. Los jóvenes se sienten atraídos por el equipo y la posibilidad de poder manejarlo, al mismo tiempo que los inmerge en la industria actual. De esta forma se mejora la calidad educativa, ya que el primer paso para la enseñanza es tener alumnos interesados en aprender.

Los estudiantes al realizar las lecciones también adquieren habilidades en Inteligencia Artificial y Robótica, las cuales son claves para entender el futuro mercado laboral en el que se desarrollarán. Esto puede verse como una preparación laboral gracias a las lecciones creadas para el proyecto.

También, en caso de que los estudiantes quieran continuar con sus estudios en un área relacionado a la Inteligencia Artificial, ya cuentan con algunas de las bases de conceptos por las lecciones proporcionadas. Esto facilitará el entendimiento de algunos cursos en el trayecto universitario especializado. Además, al haber desarrollado proyectos de robótica, sistemas de visión e Inteligencia Artificial, le permite al estudiante conocer si le gustan esas áreas para buscar carreras afines.

Al implementar proyectos a nivel europeo, la universidad se posiciona como líder en innovación educativa, aumentando el prestigio y el atractivo para los futuros estudiantes quienes quieran estudiar en esta universidad.

Al empezar el proyecto, se investigó algunos cursos de aprendizaje de Inteligencia Artificial y además algunos de los robots estudiados podían complementarse con kits de aprendizaje del mismo robot. La mayoría de los cursos en línea explican las bases de Inteligencia Artificial, mas no permiten al estudiante implementar esos conocimientos. Con los kits que proporciona el robot, limitan el robot ya que su aplicación es con programación por bloques, donde si bien se pueden aprender las bases de los sensores y del ambiente robótico, no permiten entender línea por línea el código de Inteligencia Artificial ya que usa bloques preconfigurados desde la fábrica. Este proyecto al implementar módulos de enseñanza basados en “Learning by Doing” permite obtener un aprendizaje inmersivo, el cual además de abarcar más tipos de aprendizaje.

La población meta (colegios y estudiantes) no son los únicos con beneficios con el proyecto. Al implementar esta tecnología (Inteligencia Artificial y el robot Dobot Magician) el cliente (la UDC) puede tener nuevas oportunidades para colaboraciones con otras instituciones (interesadas en Inteligencia Artificial o Robótica) y empresas tecnológicas, como más colegios o la casa madre del robot Magician (Dobot). Incluso, durante la realización del proyecto se estuvo analizando la posibilidad de colaborar con Dobot ya que se pretenden comprar más robots para la implementación en el colegio donde se harán las pruebas de las lecciones.

En resumen, el proyecto ayuda al cliente (la Universidade da Coruña) al ampliar su reputación, atraer nuevos estudiantes y crear oportunidades de

financiación y colaboración. Además, ayuda a la población meta de las lecciones ya que les genera beneficios futuros para el mercado laboral o la continuación de sus estudios ya que se generaron las bases del área de robótica, sistemas de visión e Inteligencia Artificial y con esto conocen si les gusta esas áreas para su desarrollo profesional.

## 9. Conclusiones y recomendaciones

### 9.1 Conclusiones

En el proyecto se realizó un análisis exhaustivo de las opciones disponibles de robots en el mercado, considerando criterios como precio, efectores, tamaños, grados de libertad, sensores y cámaras.

Se seleccionó el robot más adecuado para el programa AIM@VET, garantizando que el modelo estuviera acorde a las necesidades educativas y presupuestarias del cliente. El robot seleccionado fue el Magician del productor Dobot.

Se elaboró un plan de aprendizaje que incluye lecciones de Inteligencia Artificial utilizando un brazo robótico.

Las lecciones creadas están diseñadas para introducir a los estudiantes los conceptos básicos de robótica e Inteligencia Artificial, como movimientos autónomos y la interacción con el entorno mediante sensores como una cámara.

Además, las lecciones creadas incluyen cinco lecciones cumpliendo con el objetivo establecido de crear al menos dos lecciones.

Las lecciones fueron implementadas según el simulador seleccionado, donde se desarrollaron y probaron los movimientos y algoritmos de control del brazo robótico.

Las simulaciones replicaron los movimientos esperados y dado que siempre se lograba llegar a la pieza meta, indica que se cumplió el objetivo de realizar el 90% de los movimientos planteados para el robot en la simulación.

Las lecciones de aprendizaje fueron validadas con el robot físico y se aseguró que los conceptos y movimientos desarrollados en el simulador se transfirieran efectivamente al entorno real.

Las pruebas con el robot físico además replicaron el 90% de los movimientos planteados, cumpliendo así con el objetivo de la implementación física.

Las pruebas a la población muestran manifestaron un funcionamiento consistente del brazo robótico, confirmando la aplicabilidad de las lecciones en un entorno educativo real.

También los resultados de las evaluaciones aplicadas a la población muestran indicaron con un 77,5% que las lecciones cumplían con los objetivos de enseñar las bases de robótica, sistemas de visión e Inteligencia Artificial para la aplicación desarrollada.

El proyecto cumplió con éxito los objetivos establecidos, demostrando la viabilidad de integrar conceptos de Inteligencia Artificial en un entorno preuniversitario utilizando un brazo robótico.

Por último, el proyecto mostró el potencial de utilizar robots como herramientas educativas para el aprendizaje de Inteligencia Artificial para estudiantes preuniversitarios. Lo que sienta bases para futuras iniciativas educativas para que integren tecnología avanzada en la enseñanza secundaria.

## 9.2 Recomendaciones

- Es importante efectuar las lecciones y evaluaciones a un grupo de muestra mayor para considerar diferentes tipos de personas, horarios de aplicación, formación académica y tipos de aprendizaje.
- Realizar la compra de la versión paga del simulador para poder controlar el robot desde el simulador facilitando su enseñanza. Además de poder crear ambientes educativos más complejos y disfrutar de las herramientas que solo se tienen en la versión paga.
- En caso de querer usar el simulador y la aplicación de Dobot para el uso del robot, se recomienda crear o usar una librería o clase que incluya los comandos del simulador ya que estos varían entre aplicaciones.
- Por último, se recomienda continuar con el proyecto para abarcar todas las posibilidades de enseñanza Robótica e Inteligencia Artificial que posee el **Dobot Magician**, ya que en las lecciones creadas solo se utilizó una parte del potencial del robot.

## 10. Referencias

- [1] GRUPO INTEGRADO DE INGENIERÍA, «GRUPO INTEGRADO DE INGENIERÍA: Presentación,» [En línea]. Available: <https://gii.udc.es/presentacion> . [Último acceso: 22 de enero 2024].
- [2] GRUPO INTEGRADO DE INGENIERÍA, «GRUPO INTEGRADO DE INGENIERÍA: Investigación,» [En línea]. Available: <https://gii.udc.es/investigacion/detalle/5> . [Último acceso: 22 enero 2024].
- [3] AIM@VET, «AIM@VET,» [En línea]. Available: <https://aim4vet.udc.es/> [Último acceso: 22 enero 2024].
- [4] IMT ITCR, 2018, Guía para la presentación de anteproyectos.
- [5] Reglamento para Proyectos de Graduación, IMT, versión 10; marzo 2022.
- [6] Bloom, B. (s.f.), Verbos para investigación.
- [7] 2. *RoboDK API (robodk package) — RoboDK API Documentation.* (s. f.). <https://robodk.com/doc/en/PythonAPI/robodk.html#>
- [8] *Arduino Dobot Magician GetPose.* (2019, 6 julio). Dobot Forum. <https://forum.dobot.cc/t/arduino-dobot-magician-getpose/2376>
- [9] Boden, M. A. (2016). *AI: Its Nature and Future.* <https://cds.cern.ch/record/2295640>
- [10] *Dobot Magician: Communication Protocol.* (2019, 5 agosto). Dobot Robots. Recuperado 1 de febrero de 2024, de <https://www.dobot-robots.com/products/education/magician.html>
- [11] *Dobot Magician: User Guide.* (2019, 5 agosto). Dobot Robots. Recuperado 1 de febrero de 2024, de <https://www.dobot-robots.com/products/education/magician.html>
- [12] *Dobot Magician: User Manual V1.2.4.* (2019, 5 agosto). Dobot Robots. Recuperado 1 de febrero de 2024, de <https://www.dobot-robots.com/products/education/magician.html>
- [13] *Gravity: HUSKYLENS AI Machine Vision Sensor - DFRobot Wiki.* (s. f.). [https://wiki.dfrobot.com/HUSKYLENS\\_V1.0\\_SKU\\_SEN0305\\_SEN0336](https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336)

- [14] Groover, M. P., Weiss, M. . . , Nagel, R. . . N., & Odrey, N. . . G. (1986). *Industrial robotics: Technology, Programming, and Applications*.
- [15] *Mini HW 6 - Color Conversion: Computer Graphics: Spring 2022*. (s. f.). <http://15462.courses.cs.cmu.edu/spring2022/article/10>
- [16] Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media.
- [17] Nixon, M., & Aguado, A. S. (2012). *Feature Extraction and Image Processing for Computer Vision*. Academic Press.
- [18] *pydobot2*. (2020, 21 septiembre). PyPI. <https://pypi.org/project/pydobot2/>
- [19] *¿Qué es el aprendizaje supervisado? | IBM*. (s. f.). <https://www.ibm.com/es-es/topics/supervised-learning>
- [20] *scikit-learn: machine learning in Python — scikit-learn 1.5.1 documentation*. (s. f.). <https://scikit-learn.org/stable/index.html>
- [21] Ulrich, K. T., & Eppinger, S. D. (2013). *Diseño y desarrollo de productos*.
- [22] Reese, H. W. (2011). The learning-by-doing principle. *Behavioral Development Bulletin*, 17(1), 1–19. <https://doi.org/10.1037/h0100597>

# 11. Anexos

## Anexo A. Carta de Aceptación



GRUPO INTEGRADO DE INGENIERÍA



UNIVERSIDADE DA CORUÑA

Ferrol, 14 de Noviembre de 2023

**Mayda Lucía Arce Gómez**  
Tecnológico de Costa Rica  
Cartago, Costa Rica

Estimada Lucía,

En mi calidad de Catedrático de Universidad en el Departamento de Ciencias de la Computación y Tecnologías de la Información de la Universidade da Coruña, tengo el gusto de comunicarle que ha sido aceptada para poder realizar su Trabajo Final de Graduación en el Grupo Integrado de Ingeniería (GII) en el campus de Ferrol de la Universidade da Coruña, desde el día 1 de Febrero hasta el 15 de Junio de 2024. El trabajo será supervisado por uno de los investigadores del GII. Durante el desarrollo del mismo, tendrá acceso a equipos, laboratorios y transferencia de conocimientos que permitan concluir con los objetivos establecidos.

Agradeciendo de antemano su atención me despido de usted, enviándole un cordial saludo.

**Prof. Francisco Bellas Bouza**  
Grupo Integrado de Ingeniería  
Campus de Ferrol  
Universidade da Coruña (España)  
[francisco.bellas@udc.es](mailto:francisco.bellas@udc.es)  
+34881013886

## **Anexo B. Hoja de información del proyecto**

### **Datos del estudiante:**

**Nombre:** Mayda Luía Arce Gómez

**Cédula:** 1-1734-0770

**Carné ITCR:** 2017100116

**Dirección de su residencia:** San Ramón, Alajuela

**Teléfono de residencia:** 2445 0714

**Teléfono celular:** +506 8630 8246

**Correo electrónico:** [mlarce9@gmail.com](mailto:mlarce9@gmail.com)

### **Información del proyecto:**

**Nombre del proyecto:** Proyecto de implementación de Inteligencia Artificial en un módulo práctico de Robótica para la enseñanza a estudiantes preuniversitarios.

#### **Información de la empresa:**

**Nombre:** Grupo Integrado de Ingeniería (GII) de la Universidad de Coruña (UDC).

**Actividad principal:** Investigación en ingeniería aplicada.

**Zona:** España, A Coruña, Ferrol, 15403

**Dirección:** Calle Mendizábal s/n. Edificio Talleres Tecnológicos. Campus Esteiro.

**Teléfono:** +34 881013866

### **Información del encargado/asesor en la empresa:**

**Nombre:** Francisco Javier Bellas Bouza

**Puesto que ocupa:** Personal Docente e Investigador (PDI)

**Departamento:** Ciencias de la Computación y Tecnología de la Información

**Profesión:** Docente e Investigador

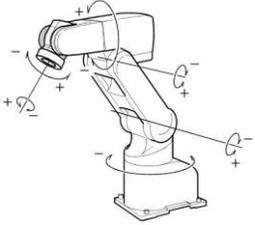
**Grado académico:** Catedrático de universidad

**Teléfono:** +34 881013886

**Correo electrónico:** [francisco.bellas@udc.es](mailto:francisco.bellas@udc.es)



## Anexo C: Cuestionario de la primera lección: Robótica y actuadores

TU7 – Questionnaire – RUBRIC		
<b>Name(s)</b>		
<b>Activity 1</b>		<b>MAX Score</b>
<p>1. What are the 6 variables that frames have?</p> <p>a) x, y, z, joint 1, joint 2, joint 3</p> <p>b) x, y, z, <math>\alpha</math>, <math>\beta</math>, <math>\gamma</math></p> <p>c) joint 1, joint 2, joint 3, joint4, joint 5, joint 6</p>		5
<p>2. Write corresponding number of how many joints the following robot has.</p> 		5
<p>3. What is the main difference between joint movement and linear movement?</p>		5
<b>Activity 2</b>		<b>Evaluation</b>
<p>1. What is the logic behind Inverse Kinematics?</p> <p>a) It calculates how to move forward based on the joint's angle.</p> <p>b) It involves simulating random movements until a desired position is found.</p> <p>c) It moves the robot in reverse.</p>		5
<p>2. Was the Magician able to reach every target during the whole teaching unit? If not, why not?</p>		5
<b>Final evaluation</b>		<b>25</b>

## Anexo D: Cuestionario de la segunda lección: Sensores

TU8 – Questionnaire – RUBRIC		
<b>Name(s)</b>		
<b>Activity 1</b>		<b>MAX Score</b>
1. What space color it is used for color segmentation? a) RGB b) CMYK c) HSV		2.5
o 2.How can you pass from one image color model to another?		5
3. What is the field of view of a camera?		5
<b>Activity 2</b>		<b>Evaluation</b>
3. Why is used a color mask in image processing? a) To mark the area with that specific color. b) To be able to change the space color. c) It gives the robot an animal face.		2.5
4. How is defined the color range of interest for a color mask?		5
What is the purpose of image binarization?		5
<b>Final evaluation</b>		<b>25</b>

## Anexo E: Cuestionario de la tercera lección: Base de datos

TU9 – Questionnaire – RUBRIC		
<b>Name(s)</b>		
<b>Activity 1</b>		<b>MAX Score</b>
1. What is the importance of the dataset? a) To recollect data and teach AI based on it. b) To understand our environment. c) To be able to move robots		5
2. What characteristics the dataset must have?		5
3. Explain how we can recollect datasets using a camera		5
<b>Activity 2</b>		<b>Evaluation</b>
4. Explain briefly the idea for the dataset implemented.		5
5. Why is important to save the data during the creation of the data set?		5
<b>Final evaluation</b>		<b>25</b>

**Anexo F: Cuestionario de la cuarta lección: Modelo de Inteligencia Artificial**

TU10 – Questionnaire – RUBRIC		
<b>Name(s)</b>		
Activity 1		MAX Score
1. What does “k” mean in the name kNN’s? a) It is the times the model can be used b) It is the number of neighbors it looks for. c) It does not have a clear meaning.		5
2. What Python libraries are used in kNN’s IA model?		5
3. Mention an example of how the model kNN’s can be implemented in the industry.		5
Activity 2		Evaluation
1. How can the kNN’s model be more efficient?		5
2. Describe briefly the process to teach an AI with kNN’s model.		5
Final evaluation		25

## **Anexo G: Códigos creados para las lecciones**

Creación propia:

<https://github.com/LucyArce/TFG-Mayda-Lucia-Arce-Gomez>