



# INFORME

## PROYECTOS—

### 2023-2024

Investigación en sistemas de conversión de lenguaje natural para la programación de robots en tareas colaborativas  
“NATURBOT”

Informe: “Final de Resultados”

Programa: Proyectos de I+D en colaboración con empresas

Número de proyecto: 22300046

Expediente: IMDEEA/2023/13

Duración: 16 meses

Coordinado en AIDIMME por: José Luís Sánchez



GENERALITAT  
VALENCIANA

IVACE+i INSTITUTO VALENCIANO  
DE COMPETITIVIDAD  
E INNOVACIÓN



Financiado por  
la Unión Europea

**AIDIMME**  
INSTITUTO TECNOLÓGICO



## ÍNDICE

<b>1</b>	<b><u>INTRODUCCIÓN, OBJETIVOS DEL PROYECTO</u></b>	<b><u>1</u></b>
<b>2</b>	<b><u>ACTIVIDADES REALIZADAS, DESARROLLO DEL PROYECTO</u></b>	<b><u>3</u></b>
<b>2.1</b>	<b>PT4. INVESTIGACIÓN EN SISTEMAS DE INTERPRETACIÓN DEL LENGUAJE NATURAL</b>	<b>3</b>
2.1.1	TAREA 4.1 DEFINICIÓN DE ONTOLOGÍAS EN LA INTERACCIÓN HUMANO-ROBOT	3
2.1.2	TAREA 4.2 INVESTIGACIÓN SOBRE SISTEMAS DE INTERPRETACIÓN DE LENGUAJE NATURAL	4
2.1.3	TAREA 4.3 GENERACIÓN DE ALGORITMOS Y DESARROLLO DE LA APLICACIÓN	6
2.1.4	TAREA 4.4 DESARROLLO DE PRUEBAS	9
<b>2.2</b>	<b>PT5. DESARROLLO DEL SISTEMA DE DESCRIPCIÓN DEL CONTEXTO</b>	<b>10</b>
2.2.1	TAREA 5.1 ANÁLISIS DE ESCENARIOS EN LA INTERACCIÓN HUMANO-ROBOT	10
2.2.2	TAREA 5.2 INVESTIGACIÓN SOBRE METODOLOGÍAS DE REPRODUCCIÓN DE ESCENARIOS	11
2.2.3	TAREA 5.3 GENERACIÓN DE ALGORITMOS Y DESARROLLO DEL SISTEMA DE TRANSFERENCIA DE POSICIONES	13
2.2.4	TAREA 5.4 DESARROLLO DE PRUEBAS	21
<b>2.3</b>	<b>PT6. DESARROLLO DEL SISTEMA DE COMUNICACIÓN. REALIZACIÓN DE PRUEBA PILOTO</b>	<b>24</b>
2.3.1	TAREA 6.1 PROGRAMACIÓN DE INSTRUCCIONES ELEMENTALES	24
2.3.2	TAREA 6.2 DESARROLLO DE SOFTWARE DE COMUNICACIÓN	25
2.3.3	TAREA 6.3 DESARROLLO DEL DEMOSTRADOR	30
2.3.4	TAREA 6.4 VALIDACIÓN DEL SISTEMA Y CONCLUSIONES	34
<b>3</b>	<b><u>RESULTADOS OBTENIDOS</u></b>	<b><u>40</u></b>
<b>4</b>	<b><u>RESUMEN Y CONCLUSIONES</u></b>	<b><u>41</u></b>



## 1 Introducción, objetivos del proyecto

Este proyecto se enmarca en las investigaciones que AIDIMME está realizando en los últimos años, relativas al desarrollo de sistemas de programación de robots, especialmente colaborativos, que faciliten esta tarea incluso a personal no experto. En este marco, el manejo de un brazo robot mediante lenguaje natural facilitaría la interacción estrecha entre un humano y un robot, tanto a nivel de programación como a nivel de ejecución de tareas complejas.

Una forma de avanzar en este campo es a través del uso de sistemas de diálogo en lenguaje natural, que permiten a los usuarios interactuar con el brazo robot de manera natural y fluida utilizando voz o texto. Estos sistemas pueden estar diseñados para reconocer órdenes específicas, tales como "mueve el brazo hacia la izquierda" o "agarras el objeto rojo", así como para procesar preguntas y respuestas complejas que involucren información específica sobre el entorno y los objetos presentes.

Los últimos avances en este campo incluyen el desarrollo de modelos de lenguaje natural avanzados, basados en la inteligencia artificial, como los modelos de lenguaje GPT-3, que pueden generar texto con un nivel de complejidad y naturalidad cada vez mayor. Estos modelos pueden ser entrenados para interactuar con los usuarios de una manera más inteligente y personalizada, y también para interpretar y comprender la intención detrás de las preguntas y comandos que se les dan.

Otro avance importante en este campo es el uso de tecnologías de procesamiento de lenguaje natural y reconocimiento de voz en tiempo real para mejorar la precisión y la velocidad de la comunicación entre el usuario y el brazo robot. Esto hace que la interacción sea más fluida y natural, permitiendo que el usuario hable con el robot de manera similar a como hablaría con otra persona.

Dados estos avances, se ha planteado el proyecto actual, pero centrándolo en la colaboración entre humanos y robots en tareas de manufactura complejas.

En un proyecto ya desarrollado por AIDIMME (INTERBOT), se planteó el desarrollo de una interfaz para la programación de robots mediante comandos verbales. Durante el desarrollo del proyecto, se pusieron de manifiesto los problemas que presenta realizar un programa de robot completo sólo utilizando la voz. Esta estrategia de "programación desde cero" utilizando la voz, no combina bien con los actuales sistemas de programación de los robots industriales, ya que se requiere que el robot pueda ser programado mediante el movimiento de su efector; esto se realiza habitualmente en robots colaborativos, pero este sistema de programación es muy poco habitual en robots industriales.

Por ello, con el presente proyecto se ha abordado el problema con una visión diferente y más práctica. Por un lado, se parte de instrucciones elementales preprogramadas en el control del robot; por otra parte, en lugar de tener que adaptar la forma de expresarse de las personas al sistema del robot, se ha conseguido que el robot “entienda” a las personas cuando hablan de forma normal. Este planteamiento ha requerido trabajar en tres áreas diferentes: “traducción” del lenguaje natural a instrucciones comprensibles por el control del robot, reconocimiento del entorno para situar las órdenes en el contexto adecuado y por último la transmisión de las instrucciones al control del robot en tiempo real.

En este proyecto se ha investigado, de forma más intensa, en el primer problema y enfocándose a trabajo industrial: cómo hacer que una orden expresada en lenguaje natural (p.ej. *ayúdame a encontrar la pieza A, o no veo la pieza A, ayúdame, o busca la pieza A y dámela, etc.*), se convierta en una secuencia de instrucciones correctas, precisas y ejecutables para el robot, considerando el entorno real de colaboración en el que se manifiesta la orden.

Obviamente, se han trabajado también técnicas para el reconocimiento del entorno, basadas en técnicas de reconocimiento de imágenes. Y también se ha abordado la transmisión al controlador del robot de la secuencia de instrucciones derivadas de la orden dada en lenguaje natural, mediante los scripts necesarios.

## 2 Actividades realizadas, desarrollo del proyecto

### 2.1 PT4. INVESTIGACIÓN EN SISTEMAS DE INTERPRETACIÓN DEL LENGUAJE NATURAL

#### 2.1.1 Tarea 4.1 Definición de ontologías en la interacción humano-robot

El principal objetivo de esta tarea es la propuesta y definición de una ontología que represente el dominio abordado por el escenario de aplicación propuesto en el proyecto, el cual esencialmente consiste en la colaboración entre humano y robot mediante la asignación de tareas sencillas de envío y recogida de piezas a/desde el puesto del trabajador.

La ontología para el entorno de aplicación del escenario ha sido desarrollada mediante el software Protégé en formato Web Ontology Language (OWL). El fichero generado contiene las entidades, propiedades y relaciones necesarias que conforman el escenario propuesto.

A continuación, se muestran algunas imágenes que ilustran algunos aspectos de la ontología desarrollada:

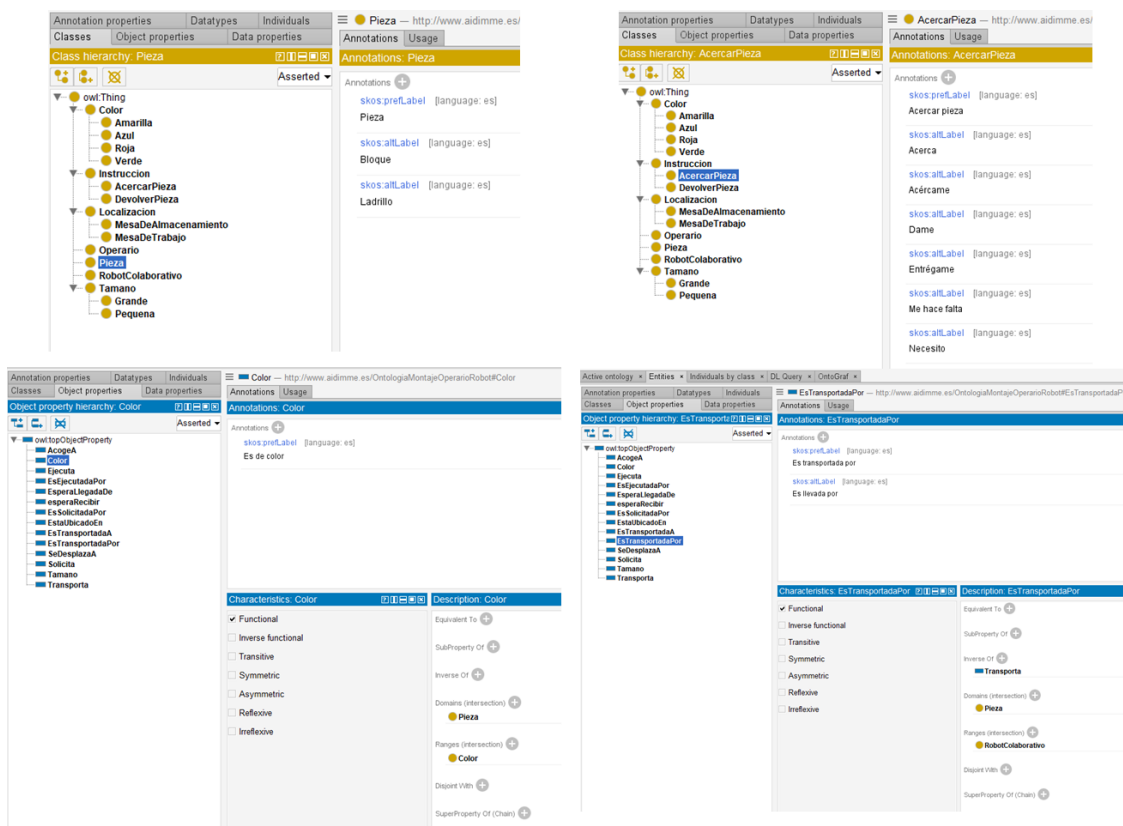


Figura 1: extracto de diferentes elementos definidos en la ontología para el entorno de aplicación  
Fuente: elaboración propia

En la parte superior de la imagen se muestran ejemplos de Entidades y la gestión de sus vocablos sinónimos mediante el uso de la anotación *skos:altLabel*. En la parte inferior se muestran ejemplos de relaciones representadas como Propiedades, como es el color de una pieza, o el transporte de piezas por parte del robot colaborativo.

La imagen a continuación muestra el mapa de relaciones entre las distintas Entidades definidas en la ontología.

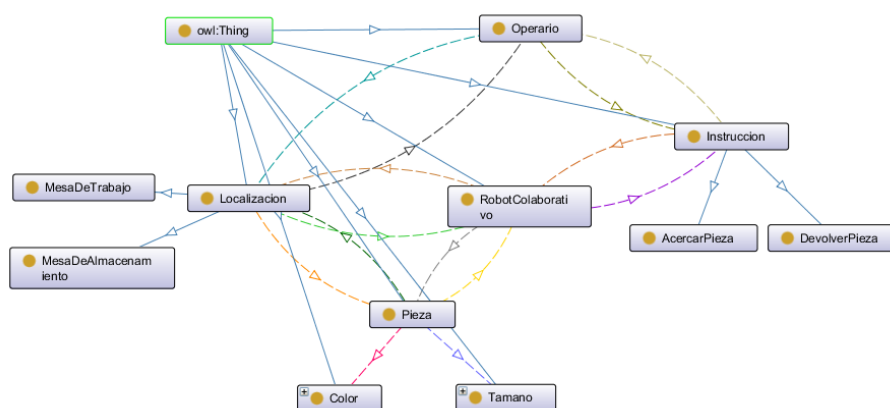


Figura 2: mapa de relaciones entre las entidades definidas en la ontología para el caso de uso  
Fuente: elaboración propia

Como puede observarse en la figura, el mapa de relaciones entre Conceptos generado es relativamente sencillo, pues el dominio de aplicación considera fundamentalmente las características del caso de uso, focalizado en el envío y retirada de piezas a/desde el puesto del trabajo del trabajador mediante el robot colaborativo.

Sin embargo, el diseño de la ontología permite su ampliación mediante la inserción de nuevas Entidades, Propiedades y Relaciones según actualizaciones del escenario o su aplicación en entornos de distintas características.

#### 2.1.2 Tarea 4.2 Investigación sobre sistemas de interpretación de lenguaje natural

Se han revisado diferentes sistemas de reconocimiento de voz, que permiten trasladar lo que habla una persona (audio) a una transcripción en forma de texto, lista para poder ser interpretada: CMU Sphinx, Julius, Vosk.

También se han analizado servicios en la nube de distintos proveedores, que, a cambio de un pago por uso, proporcionan el servicio de transcripción: Google Cloud Speech-to-Text, Amazon Transcribe Speech-to-Text, Microsoft Azure Speech Services, IBM Watson Speech to Text.

Todos estos sistemas se basan en la transcripción del audio a texto, pero no en su interpretación, que es lo que se busca con NATURBOT.



Un sistema de interpretación del lenguaje natural por voz va más allá. Se trata una tecnología diseñada para permitir la interacción entre humanos y computadoras a través del habla. Utilizando algoritmos avanzados de procesamiento del lenguaje natural (PLN) y reconocimiento de voz, estos sistemas son capaces de comprender y procesar los comandos y consultas emitidas por los usuarios mediante su voz. Sus principales características incluyen la capacidad de transcribir la voz a texto con alta precisión, entender el significado y la intención detrás de las palabras pronunciadas, y tomar acciones en función de esa comprensión. Además, algunos sistemas también pueden generar respuestas habladas o texto a partir de las solicitudes del usuario, lo que permite una interacción bidireccional natural.

En el presente proyecto se aplica esta tecnología para interpretar las acciones que un usuario desea que un robot ejecute en tareas colaborativas en un entorno industrial. El reto consiste en lograr la aplicación de una tecnología del ámbito doméstico o de oficinas a un ámbito industrial donde se requiere mayor precisión y velocidad de respuesta, en un entorno con ruido de maquinaria u otros operarios.

En los últimos diez años, han surgido diferentes sistemas encaminados a ofrecer servicios de interpretación de lenguaje natural por voz. Se ha llevado a cabo una revisión y análisis de este tipo de sistemas y la siguiente tabla lo resume especificando algunas ventajas e inconvenientes detectados.

Nombre del Sistema	Empresa	Año Lanzamiento	Ventajas	Inconvenientes
<b>AWS Lex</b>	Amazon	2016	<ul style="list-style-type: none"> <li>- Integración con otros servicios de AWS como Lambda y Polly.</li> <li>- Fácil de usar con la consola de AWS.</li> <li>- Escalabilidad y fiabilidad debido a la infraestructura de AWS.</li> </ul>	<ul style="list-style-type: none"> <li>- La personalización puede requerir habilidades de desarrollo.</li> <li>- Algunas características avanzadas pueden ser limitadas en comparación con otros sistemas.</li> </ul>
<b>Google Dialogflow</b>	Google	2016	<ul style="list-style-type: none"> <li>- Potente capacidad de procesamiento del lenguaje natural.</li> <li>- Integración con Google Cloud Platform y otros servicios de Google.</li> <li>- Soporte para múltiples canales de entrada y salida.</li> </ul>	<ul style="list-style-type: none"> <li>- La configuración inicial puede ser compleja para usuarios no técnicos.</li> <li>- Algunas características avanzadas pueden requerir conocimientos de desarrollo avanzado.</li> </ul>
<b>IBM Watson</b>	IBM	2013	<ul style="list-style-type: none"> <li>- Amplia gama de capacidades de procesamiento del lenguaje natural, incluyendo análisis de sentimientos y extracción de entidades.</li> <li>- Integración con servicios de IBM Cloud.</li> <li>- Personalización y entrenamiento del modelo avanzado.</li> </ul>	<ul style="list-style-type: none"> <li>- El precio por uso puede ser elevado para ciertos casos.</li> <li>- La curva de aprendizaje puede ser elevada para usuarios no técnicos.</li> </ul>
<b>Microsoft Azure Language Understanding</b>	Microsoft	2017	<ul style="list-style-type: none"> <li>- Integración con servicios de Microsoft Azure, como Azure Bot Service y Azure Functions.</li> <li>- Herramientas de entrenamiento y evaluación fáciles de usar.</li> <li>- Soporte para varios idiomas y canales de entrada.</li> </ul>	<ul style="list-style-type: none"> <li>- Limitaciones en la personalización avanzada del modelo.</li> <li>- Algunas características pueden requerir conocimientos técnicos elevados para la configuración.</li> </ul>

Figura 3: Tabla comparativa sistemas de servicios de interpretación de lenguaje natural

Fuente: elaboración propia

El desarrollo final de la aplicación se va a llevar a cabo en AWS Lex debido a su entorno de desarrollo amigable y a facilidad de escalado de los desarrollos que se realicen. Todos estos sistemas son sistemas acotados y guiados, suficiente para el ámbito de este proyecto.

### 2.1.3 Tarea 4.3 Generación de algoritmos y desarrollo de la aplicación

Se plantea un escenario de trabajo inicial donde el usuario quiere realizar un montaje con diferentes tipos de piezas. El usuario dará instrucciones verbales en lenguaje natural al robot indicando que piezas desea utilizar, y el robot debe ejecutar las acciones que le han sido solicitadas.

El SISTEMA VOZ debe de interpretar el lenguaje natural empleado por el usuario, entender que acción quiere llevar a cabo el usuario y con qué tipo de pieza. Además, deberá de comunicar la información adecuada para que se activen los sistemas de reconocimiento del entorno.

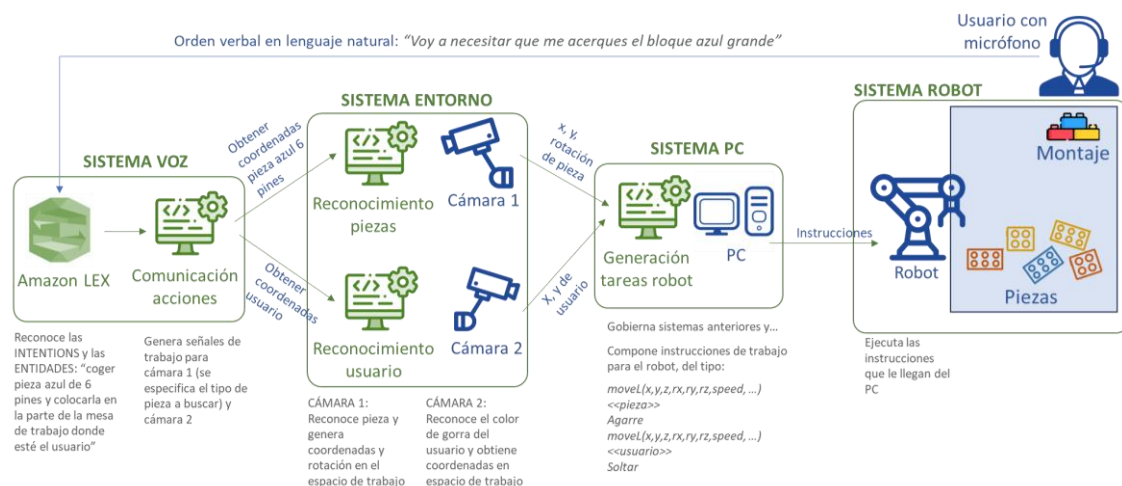


Figura 4: esquema de funcionamiento general de la solución  
Fuente: elaboración propia

Tal y como se ha definido en la ontología de la Tarea 4.1, en este entorno de trabajo, el usuario puede desarrollar 2 tipos de acciones (intentions), utilizando 6 tipos diferentes de piezas.

Acciones a realizar por el robot:

- Coger una pieza de lego de la mesa y colocarla en la zona de la mesa próxima al usuario
- Devolver una pieza que ya no quiere el usuario a la zona de la mesa donde están el resto de las piezas.

Piezas a manipular:

- Piezas amarillas pequeñas o grandes
- Piezas rojas pequeñas o grandes
- Piezas azules pequeñas o grandes

Para la implementación del sistema AWS Lex cada posible petición del usuario al sistema debe de basarse en el reconocimiento de “intenciones”. Estas "intenciones" (intentions, en el entorno de desarrollo de AWS Lex) son representaciones de las acciones o propósitos que un usuario puede tener al interactuar con un chatbot o una interfaz de voz. Cada intención define qué tipo de solicitud o pregunta está realizando el usuario.

Cada “intention” en Amazon Lex debe tener información asociada que defina su propósito, cómo se debe manejar y cómo interactuar con los usuarios. La información asociada a una intención puede incluir los siguientes aspectos:

- Nombre de la intención: Un nombre descriptivo que identifica de manera única la intención. Este nombre se utiliza para referirse a la intención en la configuración y programación.
- Ejemplos o expresiones: Una lista de ejemplos de expresiones que los usuarios podrían utilizar para activar la intención. Estos ejemplos sirven para entrenar el modelo de lenguaje y permiten a Amazon Lex comprender cuando un usuario está expresando esa intención.
- Respuestas o acciones: La lógica que define lo que el chatbot debe hacer cuando se activa esta intención. Esto puede incluir la ejecución de una función, la búsqueda en una base de datos, el acceso a servicios web, el cálculo de una respuesta o cualquier otra acción específica para satisfacer la intención del usuario.
- Parámetros o entidades: Si la intención requiere información adicional del usuario para completarse, se debe definir los parámetros o entidades asociados a la intención.
- Preguntas de clarificación: Se pueden definir preguntas de clarificación que el chatbot realizará si no entiende completamente la intención del usuario o necesita más información. Esto permite aclarar la solicitud del usuario y garantizar una interacción más precisa.

A continuación, se desarrolla la información asociada a cada una de estas dos “intentions”.

*2.1.3.1 Coger una pieza y colocarla en la zona de la mesa próxima al usuario*

**Intention:** Recibir una pieza que necesita el usuario

**Entidades:** color (amarillo, rojo, azul), tamaño (pequeña, grande)

**Locuciones:**

- Para “dar”: Usar, utilizar, dar, necesitar, hacer falta
- Para “color”: amarillo, rojo, azul
- Para “Tamaño”: pequeña, grande, corta, larga, alargada, 4 pines, 8 pines,

**Ejemplos:**

- Necesito que me des la pieza grande roja
- Me hace falta la amarilla más pequeña
- Voy a utilizar la pieza azul de 8 pines

**Respuestas:** ¿Quieres que te de la pieza <<color>> de <<tamaño>>?

*2.1.3.2 Devolver una pieza que ya no quiere el usuario a la zona de la mesa donde están el resto de las piezas.*

**Intention:** Devolver una pieza que ya no necesita el usuario

**Entidades:** color (amarillo, verde, azul), tamaño (pequeña, grande)

**Locuciones:**

- Para “devolver”: Devolver, retornar, no necesitar, no hacer falta
- Para “color”: amarillo, roja, azul
- Para “Tamaño”: pequeña, grande, corta, larga, alargada, 4 pines, 8 pines,

**Ejemplos:**

- Ya no necesito la pieza grande roja
- No me hace falta la amarilla más pequeña
- Voy a devolver la pieza azul de 8 pines

**Respuestas:** ¿Quieres que devuelva a la mesa de trabajo la pieza <<color>> de <<tamaño>>?

El desarrollo de los distintos elementos que componen el sistema se basa en el esquema siguiente:

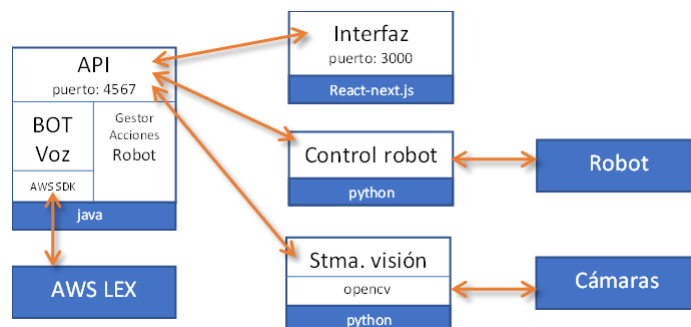


Figura 5: esquema de componentes del sistema completo

Fuente: elaboración propia

Se ha desarrollado una interfaz de usuario, utilizando para ello react-next.js, un framework basado en React, orientado para la realización de aplicaciones de front-end.

Desde este desarrollo se puede acceder al API del módulo principal, ver los comandos que se han almacenado, el estado del comando actual que se está reconociendo por el sistema, el estado en el que se encuentra el resto de los comandos, controlar la puesta en marcha del sistema de voz y ver el estado del robot y el sistema de control.

Desde este interfaz, se puede gestionar la parada o el inicio del sistema de control de voz, y se visualiza la información de lo que está pasando con el sistema de voz, el sistema de visión y el robot. Aquí el usuario debe de conocer:

- Si se han interpretado correctamente su orden verbal.
- Si se ha transmitido
- Si el robot lo está ejecutando
- Cuando se ha terminado de ejecutar una acción

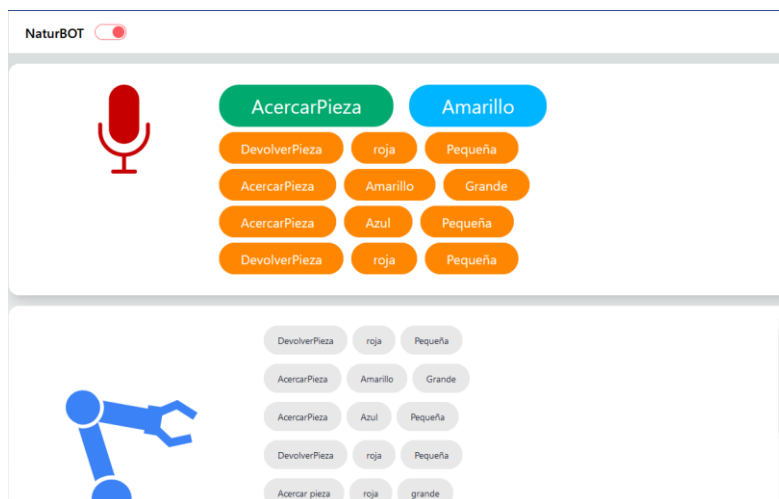


Figura 6: Diseño de interfaz de usuario  
Fuente: elaboración propia.

#### 2.1.4 Tarea 4.4 Desarrollo de pruebas

El objetivo de las pruebas se puede resumir en:

- Verificar la eficacia del sistema para reconocer e interpretar la INTENTION formulada por un usuario (entregar o devolver una pieza), el COLOR (rojo, amarillo o azul) y el TAMAÑO de la pieza (grade o pequeño).

Para ello, se han llevado a cabo pruebas preliminares de funcionamiento del sistema desarrollado emitiendo diferentes instrucciones verbales que siguen la siguiente estructura, expresada en lenguaje natural:

<INTENTION> + <COLOR> + <TAMAÑO>

Algunos ejemplos de frases de testeo han sido:

“Por favor, me puedes entregar una pieza amarilla pequeña”

“Dame una pieza azul grande”

“Me gustaría devolver una pieza de color amarilla pequeña”

Las pruebas han sido llevadas a cabo por diferentes usuarios (técnicos de AIDIMME participantes en el proyecto). En las siguientes imágenes se muestra un ejemplo de algunos momentos durante la realización de las pruebas.



Figura 7: Ejemplo de pruebas de funcionamiento realizadas  
Fuente: elaboración propia.

## 2.2 PT5. DESARROLLO DEL SISTEMA DE DESCRIPCIÓN DEL CONTEXTO

### 2.2.1 Tarea 5.1 Análisis de escenarios en la interacción humano-robot

Para abordar los aspectos de seguridad de robots colaborativos hay que considerar la norma ISO/TS 15066: Se trata de una especificación técnica con directrices adicionales que ayudan al integrador a evaluar y adecuar medidas para el uso de robots colaborativos.

Las aplicaciones colaborativas son diferentes a las de los sistemas robóticos tradicionales, ya que las personas pueden trabajar próximos al sistema robot cuando está en funcionamiento y se permite el contacto físico humano-robot bajo ciertas condiciones, sin necesidad de aislar el robot mediante resguardos o vallas.

Grado de Interacción	Comparten Espacio Físico	Realizan Tareas al Mismo Tiempo	Ejemplo	Programación del Robot	Parámetros del Programa del Robot que podrían Cambiar
Colaboración física directa	Sí	Sí	Un operario y un robot ensamblan componentes en una línea de producción.	Fija	N/A
Colaboración física indirecta	Sí	No	Un operario carga y descarga piezas mientras un robot realiza operaciones de mecanizado.	Requiere instrucciones	Velocidad de mecanizado, herramientas utilizadas
Colaboración temporal	Sí	No	Un robot y un operario trabajan en turnos alternados en una tarea de fabricación.	Requiere instrucciones	Secuencia de cambio entre tareas, ajuste de parámetros
Colaboración de supervisión y asistencia	Sí	Sí	Un operario supervisa y asiste a un robot que realiza operaciones de pintura en una pieza.	Fija	N/A
Colaboración independiente	No	No	Varios robots trabajan en células de fabricación separadas dentro de una planta automatizada.	Fija	N/A
Colaboración en espacios de trabajo separados	No	Sí	Un operario realiza la inspección final de productos mientras varios robots ensamblan componentes en líneas de producción separadas.	Requiere instrucciones	Parámetros de inspección, criterios de calidad

Figura 8: Tabla comparativa de potenciales grados de colaboración humano-robot  
Fuente: elaboración propia

### 2.2.2 Tarea 5.2 Investigación sobre metodologías de reproducción de escenarios

Para el reconocimiento de piezas, herramientas o cualquier otro objeto que deba ser manipulado por un robot durante la tarea colaborativa pueden emplearse sensores, dispositivos o marcadores adheridos a los propios objetos para identificar su posición, o bien optar por sistemas basados en cámaras o escáneres. En la siguiente tabla se desarrolla un resumen de diferentes tecnologías que pueden emplearse para capturar la posición de objetos en un entorno de trabajo como el planteado en el proyecto.

Objetivo de Captura	Tecnología	Ejemplo de Dispositivo	Aplicación en Robótica Colaborativa
Identificación de Posición de Piezas	Sistemas de Visión por Computadora	Cámaras RGB-D	Las cámaras RGB-D pueden capturar la posición y forma de las piezas en el espacio de trabajo, permitiendo al robot localizarlas y manipularlas en colaboración con el operario.
	Sistemas de Localización en Tiempo Real	Sensores de ultrasonido de Tecnología RFID	Los sensores de ultrasonido o tags de RFID pueden utilizarse para determinar la posición relativa de las piezas o herramientas en el entorno de trabajo, permitiendo al robot interactuar con ellas de manera precisa.
	Marcadores de Posición	Códigos QR de Etiquetas de marcadores AR	Los marcadores de posición pueden colocarse en las piezas o herramientas para identificar su posición y orientación en el espacio, facilitando la interacción con el robot.
	Sistemas de Escaneo 3D	Escáneres 3D	Los escáneres 3D pueden crear modelos precisos de las piezas o herramientas en el entorno de trabajo, permitiendo al robot planificar y ejecutar movimientos con mayor precisión durante la colaboración.
	Cámaras convencionales con algoritmos de IA	Cámaras convencionales	Las cámaras convencionales equipadas con algoritmos de inteligencia artificial pueden analizar el entorno de trabajo y reconocer las piezas o herramientas, permitiendo al robot interactuar de manera inteligente y adaptativa con su entorno.

Figura 9: Tabla de tecnologías de identificación de posición de piezas.

Fuente: elaboración propia

Se define un escenario de trabajo donde humano y robot deben desempeñar una tarea colaborativa donde el humano emite de forma secuencial una serie de instrucciones que el robot debe ir ejecutando paso a paso hasta completar la tarea.

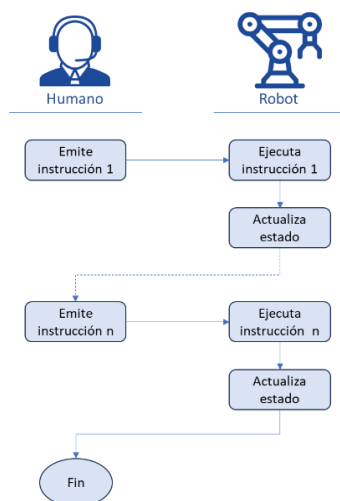


Figura 10: Secuencia de tareas

Fuente: elaboración propia

Se plantea un escenario con dos zonas de trabajo. En una primera zona (mesa 1) trabaja solo el robot recogiendo piezas, y en la segunda zona (mesa 2) trabajan el humano y el robot (que entrega y recoge las piezas solicitadas).



De este modo el robot asiste recogiendo piezas de una zona de trabajo donde el humano no alcanza sin tener que realizar un desplazamiento, que le haría perder tiempo en la ejecución de la tarea, al mismo tiempo que incrementaría su fatiga.

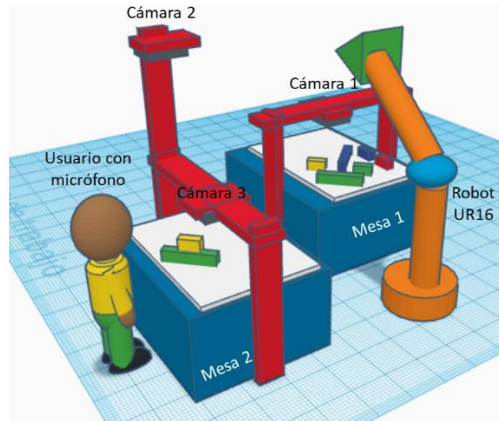


Figura 11: boceto del escenario de trabajo colaborativo nº2  
Fuente: elaboración propia

### 2.2.3 Tarea 5.3 Generación de algoritmos y desarrollo del sistema de transferencia de posiciones

En primer lugar, se genera un dataset con un conjunto de diferentes fotografías de las piezas de trabajo que recojan las diferentes ubicaciones y orientaciones que estas puedan llegar a tener durante la fase de pruebas del demostrador en el PT6.

Tal y como se ha documentado en el entregable asociado al PT4, para el demostrador se han empleado seis piezas diferentes, formadas por dos tamaños (grande y pequeño) y tres colores (amarillo, rojo y azul).

Para tomar estas fotografías iniciales, se definen 9 posiciones donde se colocará una pieza para realizar las instantáneas. Las posiciones 1, 2 y 3 serán las más próximas a la cámara, y las 7,8 y 9 las más alejadas

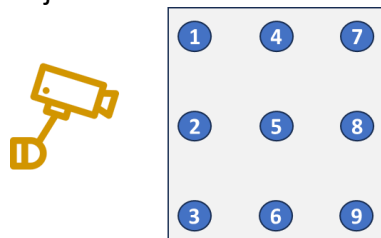


Figura 12: Posiciones de las piezas  
Fuente: elaboración propia

Por tanto, se realizan 54 fotografías en total (6 piezas x 9 posiciones).

Cada fotografía se codificará de la siguiente manera:

- G=Piezas grandes (8 pines), P= Piezas pequeñas (4 pines)
- AM= Piezas amarillas, AZ= Piezas Azules, RJ= piezas rojas
- Número de la posición en la mesa= del 1 al 9

Los parámetros del montaje final para la toma de fotografías en un banco de pruebas inicial son:

- A (altura mesa de trabajo) = 74'5 cm
- B (distancia horizontal cámara a la mesa de trabajo) = 0 cm
- C (distancia vertical cámara a la mesa de trabajo) = 31 cm

Siendo el esquema el siguiente.

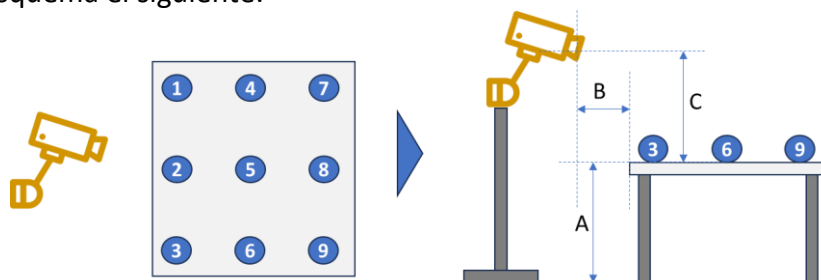


Figura 13:Esquema de montaje de cámara

Fuente: elaboración propia

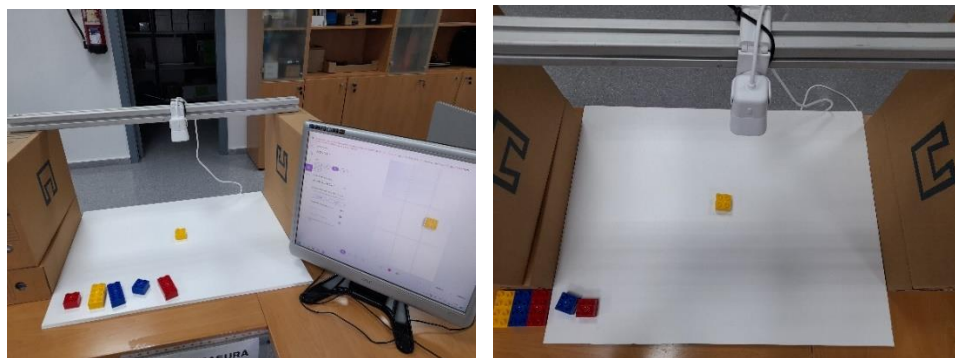


Figura 14:Ejemplo de realización de fotografías

Fuente: elaboración propia

A continuación, se muestra el dataset completo de fotografías para las piezas catalogadas como "Piezas Grandes".

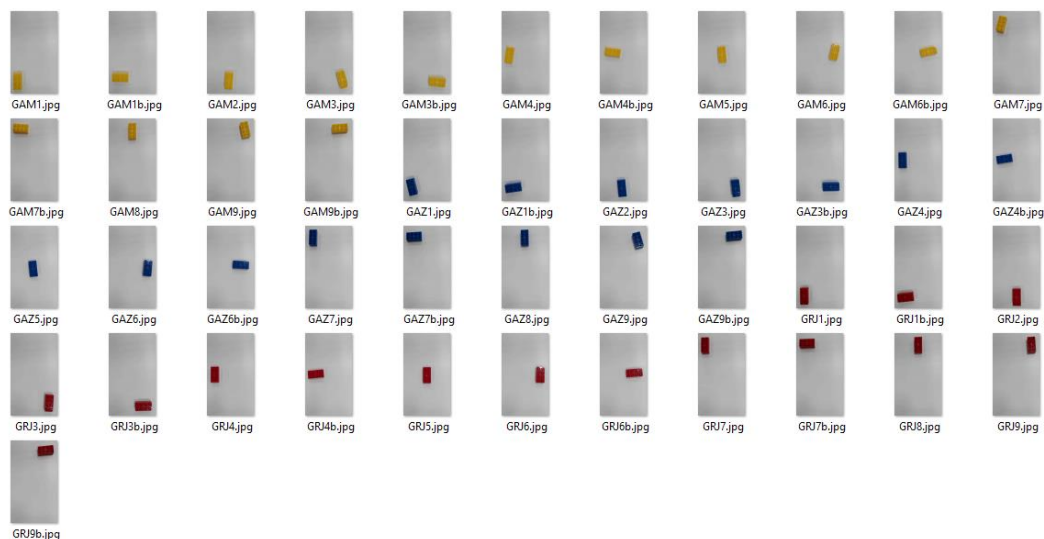


Figura 15: Archivo de entrenamiento de piezas grandes  
Fuente: elaboración propia

Y aquí se muestra el dataset completo para las “Piezas pequeñas”. Este dataset contiene menos fotografías que el anterior ya que al ser las piezas cuadradas, no ha hecho falta repetir fotografías con las piezas en orientación vertical y horizontal.

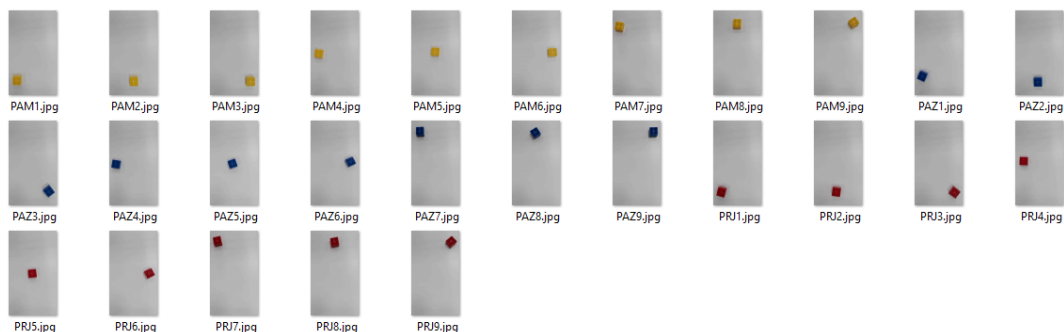


Figura 16: Archivo de entrenamiento de piezas pequeñas  
Fuente: elaboración propia

### 2.2.3.1 Máscara de filtrado de color

Para obtener un algoritmo que sea capaz de identificar las piezas que se posicionen dentro del rango de visión de la cámara, es necesario identificar cuál es la característica de estas que más destaca en la imagen. En este caso, por el fondo blanco de la mesa, el color es la característica visual más fácil de identificar. Por eso se diseñará e implementará un sistema de filtrado de color capaz de aislar cada color de pieza del fondo.

Para realizar el filtrado, se ha utilizado la codificación de color HSV. La librería OpenCV guarda las imágenes con codificación BGR. Este formato almacena para cada píxel de la

imagen un vector de dimensión 3, donde cada una de sus coordenadas representa la cantidad de azul, verde y rojo que debe iluminarse el píxel respectivamente. Por contra, el formato HSV utiliza igualmente vectores de dimensión 3, donde la información que se guarda es el color (Hue), la saturación y el valor. De esta manera, toda la información del color está almacenada en una única componente.

En la siguiente figura se puede observar el modelo HSV de manera gráfica:

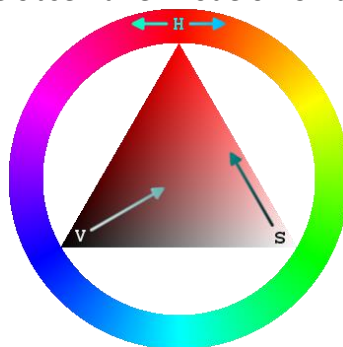


Figura 17: Modelo HSV

Fuente: [https://es.wikipedia.org/wiki/Modelo\\_de\\_color\\_HSV#/media/Archivo:Triangulo\\_HSV.png](https://es.wikipedia.org/wiki/Modelo_de_color_HSV#/media/Archivo:Triangulo_HSV.png)

Se utiliza a continuación OpenCV para aplicar a la imagen recibida de la cámara 3 filtros que identifiquen únicamente los valores de Hue que correspondan al rojo, el amarillo y el azul, descartando todos los demás valores. Es importante destacar que el color rojo está incluido tanto en el inicio del espectro como en el final, lo que lleva a que realmente se definan 4 máscaras, al dividir el rojo en *red\_low* y *red\_high*, aunque el resultado de estas será tratado como único. Mediante este proceso se obtienen resultados como los siguientes:



Figura 18: Resultado tratamiento de imagen

Fuente: elaboración propia

### 2.2.3.2 Cierre morfológico

Tal como se puede ver en la imagen anterior, debido a la propia iluminación, el sombreado y la posición respecto a la cámara de la pieza, el sistema de máscaras identifica las piezas, pero normalmente no identifica algunos puntos internos de las

piezas. Es necesario cerrar estos huecos ya que, sino produciría problemas en los pasos siguientes. Para conseguir esto se han utilizado dos procedimientos de modificación morfológica de las imágenes conocidos como dilatación y erosión. En la siguiente imagen muestra el resultado de un ejemplo de su aplicación.

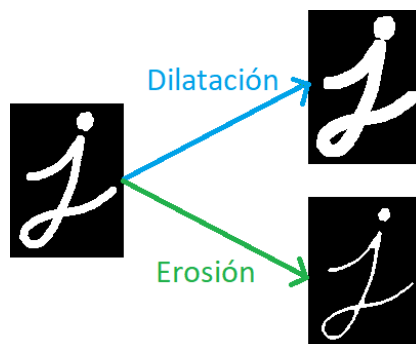


Figura 19: Ejemplo de procesos de “dilatación” y “erosión” de la imagen  
Fuente: [https://docs.opencv.org/4.x/d9/d61/tutorial\\_py\\_morphological\\_ops.html](https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html)

La combinación de estas dos operaciones se conoce como cierre morfológico y permite cerrar los huecos internos de la imagen, tal como se ve en la siguiente ilustración.



Figura 20: Ejemplo de cierre morfológico  
Fuente: [https://docs.opencv.org/4.x/d9/d61/tutorial\\_py\\_morphological\\_ops.html](https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html)

Podemos ver en la siguiente ilustración el resultado de aplicar un cierre morfológico al resultado obtenido tras la aplicación de máscaras de color con algunas de las piezas del piloto demostrativo del proyecto.

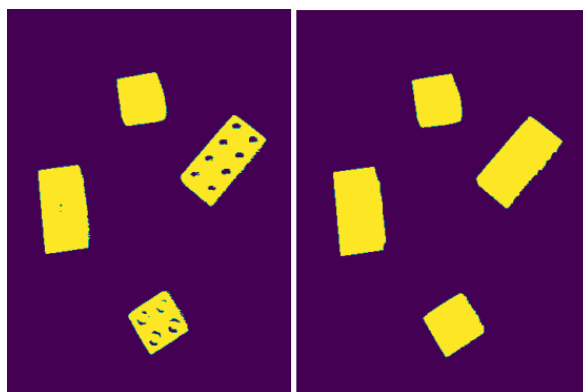


Figura 21: Resultado del cierre morfológico  
Fuente: elaboración propia

### 2.2.3.3 Identificación de pieza

Para identificar las piezas desde la imagen, se ha utilizado la extracción de componentes conexas que permite openCV. Esta herramienta busca aquellas regiones de la imagen que se encuentren sólidamente conectadas entre sí y las separa de las demás, obteniendo como resultado la posición y el tamaño de cada componente conexas.

Es importante destacar que el fondo en sí es una componente conexas también. Esta se descarta manualmente evitando la componente conexas de mayor tamaño. En la siguiente imagen se muestran las componentes conexas.

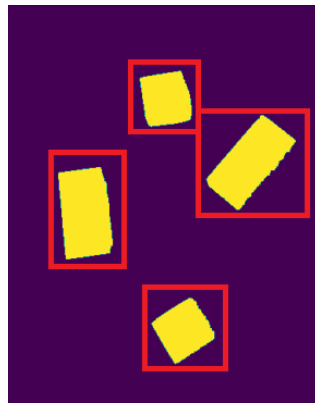


Figura 22: Identificación de cuatro componentes conexas.  
Fuente: elaboración propia

Cada una de estas componentes es extraída y tratada de manera individual. El centro de este rectángulo resultante es el empleado para obtener las **coordenadas x, y de la pieza**. El siguiente paso es encontrar el contorno de la imagen. Se ha llevado a cabo mediante la herramienta de “find countours” de openCV. Esta herramienta esta aplica una máscara que permite encontrar los contornos de la imagen.

Se ha tomado siempre el contorno más largo encontrado, para descartar cualquier efecto que pueda ser provocado por dos piezas que están muy cerca como se muestra en esta imagen.



Figura 23: Selección del contorno más largo  
Fuente: elaboración propia

Por último, se ha buscado el rectángulo de menor área que envuelve al contorno encontrado. Para realizar este proceso se envuelve el contorno en un rectángulo.

Para ello, se lleva a cabo un proceso iterativo de rotado y se calcula el área que engloba, al girar 90°, y se devuelve el rectángulo más pequeño considerado de cada una de las iteraciones. En la imagen se muestra en verde el rectángulo inicial que irá girando de manera iterativa.

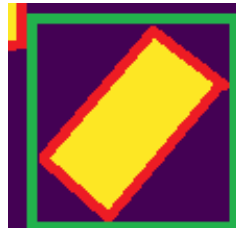


Figura 24: Rectángulo inicial en las iteraciones  
Fuente: elaboración propia

Este rectángulo tiene una triple utilidad:

- Es el rectángulo que se muestra en el programa de demostración para identificar la posición de la pieza encontrada por el sistema de visión
- El ángulo rotado indica el giro de la pieza que se pasará al robot
- Para clasificar la pieza en grande o pequeña, se utiliza la proporción entre el lado largo y corto del rectángulo. Si esta proporción es cercana a 2 se clasifica como pieza grande, mientras que, si es cercana a 1, se clasificará como pequeña.

#### 2.2.3.4 Desarrollo del sistema de referencia

Tal como se ha explicado en el apartado anterior, las coordenadas de la pieza se obtienen en función de los píxeles de la cámara y la distancia fija de la cámara a la mesa. Sin embargo, surge la problemática de que estas coordenadas no son útiles cuando se intenta indicar al robot dónde se encuentra la pieza (ya que el robot requiere de coordenadas en base a sus propios ejes, no respecto de la cámara).

Por tanto, es necesario un sistema de calibración que permita actuar como puente entre las coordenadas que genera la cámara y las coordenadas con las que es capaz de trabajar el robot.

El sistema de calibración se basa en el desarrollo de unas coordenadas sobre la mesa y la obtención de la referencia de estas coordenadas respecto al sistema de coordenadas del robot y de la cámara.

Con este sistema, se han conseguido unas coordenadas intermedias que permiten pasar de un sistema a otro. Para esta tarea, se ha generado un modelo de 3 figuras que se imprimen sobre un folio DIN A4, estas figuras consisten en una cruz, un rombo y un trébol, tal y como se muestra en la siguiente figura.

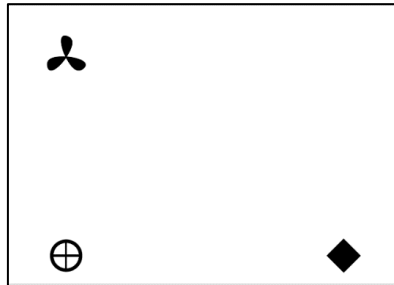


Figura 25: Figuras del sistema de coordenadas desarrollado  
Fuente: elaboración propia

El vector con punta en el trébol y cola en la cruz es el vector generador del eje “y” de coordenadas. El tamaño de este vector es siempre 15 cm. Por otra parte, el vector con punta en el rombo y cola en la cruz genera el eje “x” y mide 20 cm.

Una vez situada esta hoja sobre la mesa de trabajo, es necesario obtener en qué coordenadas se encuentra cada una de las figuras respecto del robot y respecto a la cámara, de los dos.

En el caso del robot, ha bastado con mover la pinza del robot a cada una de las figuras y anotar en un archivo JSON cuál es la coordenada correspondiente.

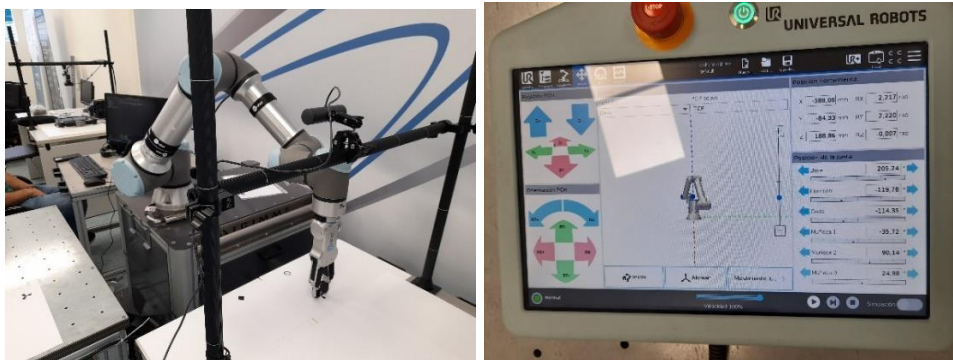


Figura 26: Obtención de las coordenadas del sistema de referencia para el robot  
Fuente: elaboración propia

El segundo paso, algo más complejo, ha sido obtener las coordenadas en las que se encuentran las figuras respecto a la cámara. Para realizar esta función, ha sido necesario desarrollar un nuevo sistema de visión artificial que sea capaz de encontrar las figuras en una imagen.

En este caso, todas las figuras son del mismo color, por lo que no es posible utilizar la misma metodología que se ha explicado en apartados anteriores. Por esa razón el método de identificación se basa en la utilización de redes neuronales convolucionales. Al disponer del diseño de las figuras en un archivo digital, se ha desarrollado un script en Python que genera posibles variaciones de la figura, incluyendo giros, estiramientos y encogimientos para obtener un dataset completo con el que entrenar al modelo. Por



otra parte, se ha desarrollado una categoría de objetos desconocidos para que el sistema aprenda a descartar aquellos objetos que no sean estrictamente las figuras. Con este dataset conformado, se ha empleado Keras y Tensorflow para entrenar un modelo que es capaz de reconocer todas las figuras. Una vez las identifica, se muestran por pantalla los ejes y pregunta al usuario si son correctos. Y finaliza generando un fichero JSON que muestra las coordenadas de las figuras respecto de la cámara.

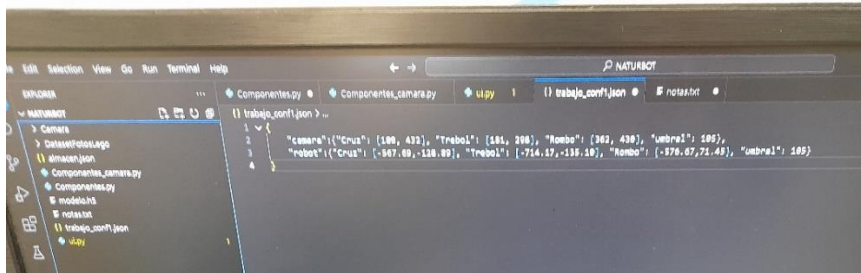


Figura 27: Ficheros con coordenadas de robot y de cámara del sistema de ejes de símbolos  
Fuente: elaboración propia

Una vez creados estos dos ficheros, el sistema de visión identifica dónde se encuentran las piezas y se envían las coordenadas de las piezas identificadas al sistema de cambio de coordenadas que calcula las coordenadas del robot.

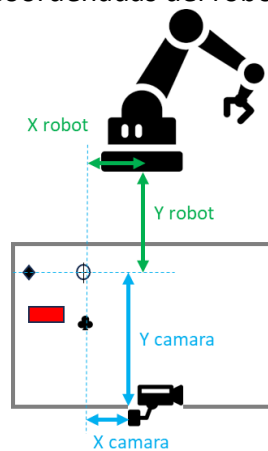


Figura 28: Esquema de coordenadas desde cámara y desde el robot al sistema de ejes  
Fuente: elaboración propia

#### 2.2.4 Tarea 5.4 Desarrollo de pruebas

El objetivo de las pruebas ha sido verificar que los seis tipos de piezas fueran perfectamente reconocidos por el sistema y se pudieran generar sus coordenadas y ángulos de agarre.

Para ello, se han realizado pruebas con las piezas en diferentes posiciones del espacio de trabajo, con diferentes orientaciones. Se han realizado pruebas con diferente cantidad de piezas en el espacio de trabajo.

También se han realizado pruebas con objetos diferentes a las seis piezas con los que se ha entrenado al sistema para verificar si interferían en el reconocimiento de las piezas. Finalmente, se han realizado pruebas para validar el método de generación de coordenadas y ángulo de agarre, primero en el simulador de Universal Robots y luego en un entorno real.

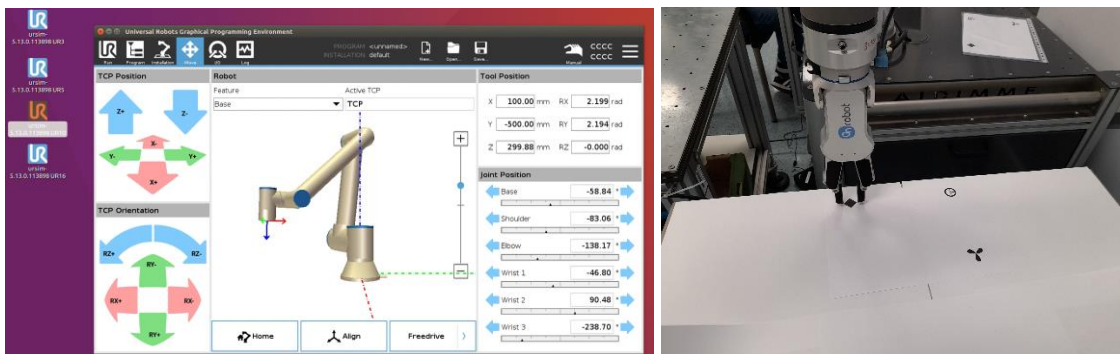


Figura 29: Pruebas en simulador y en entorno real.  
Fuente: elaboración propia.

El resultado de las pruebas realizadas ha sido satisfactorio en cuanto que se reconoce el tipo de pieza a manipular y se obtienen tanto sus coordenadas de posición, como su ángulo de agarre. En las siguientes imágenes se muestran ejemplos del resultado del reconocimiento de las piezas, con diferentes posiciones y ubicaciones de estas.

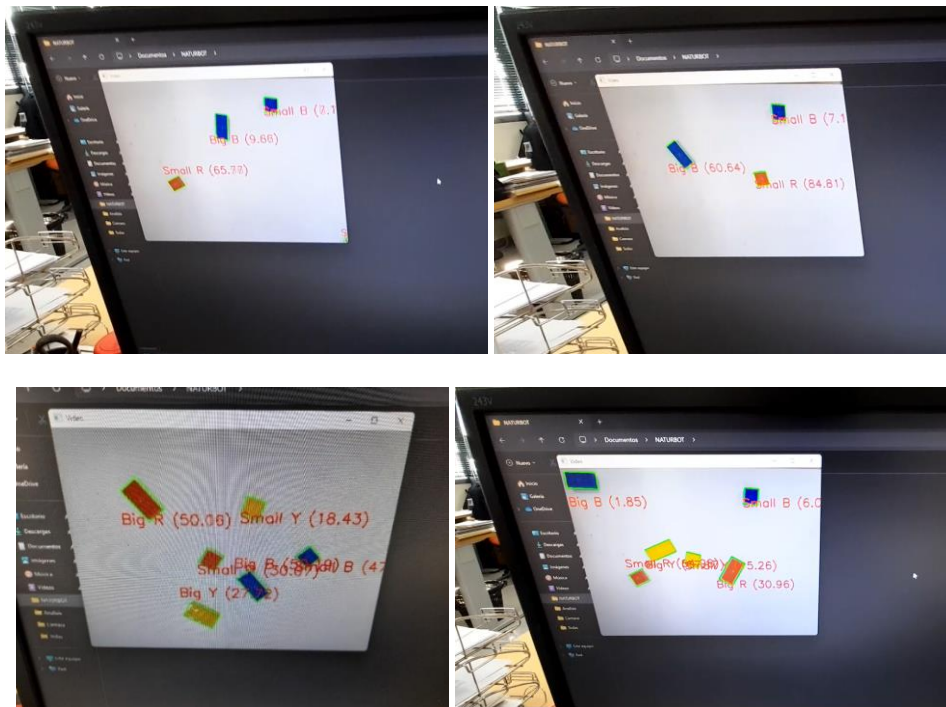


Figura 30: Ejemplos de pruebas del sistema de descripción del contexto  
Fuente: elaboración propia.

Se realizaron pruebas también introduciendo piezas o elementos “extraños” con los que no se había entrenado al sistema, por ejemplo, un bolígrafo, unas pinzas de electrónica, o cinta adhesiva.

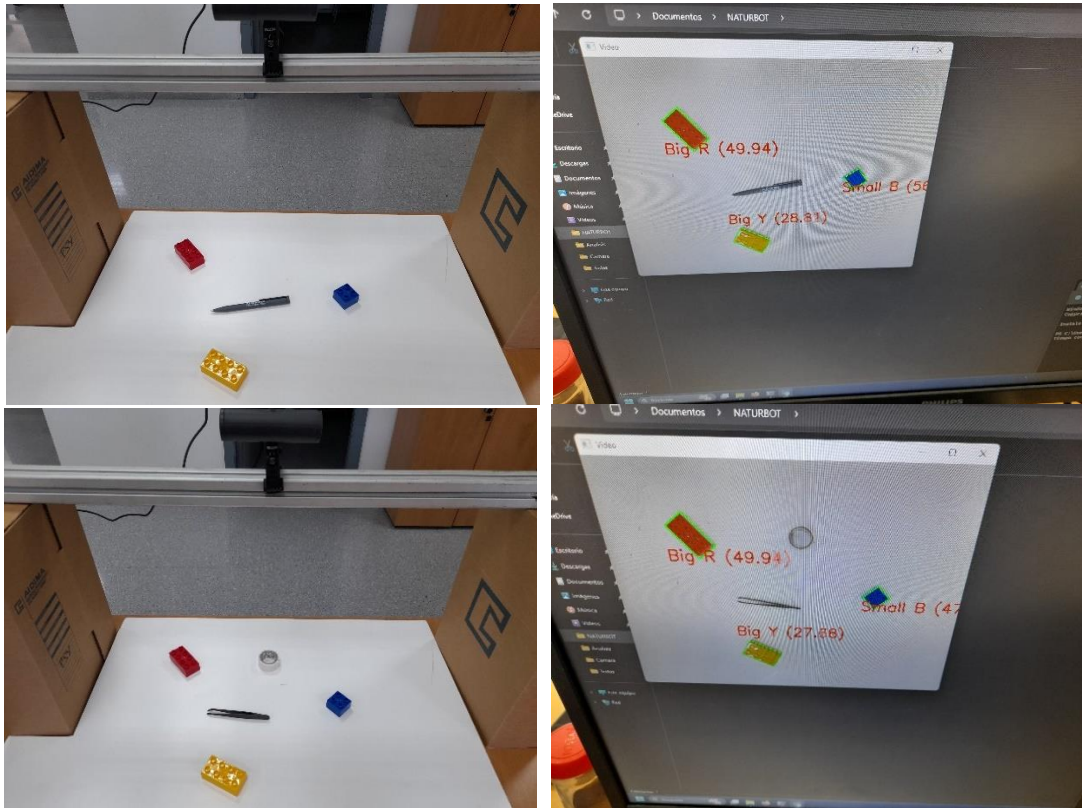


Figura 31: Pruebas con un bolígrafo, pinzas o cinta adhesiva.  
Fuente: elaboración propia.

Se comprueba como el sistema no hace caso a estos nuevos elementos y no los considera en sus cálculos de posición y ángulo de agarre.

Si embargo, si se introduce en el campo de la imagen un objeto con forma cuadrada o rectangular del color de las piezas con las que se ha realizado el entrenamiento, se interpreta como una de las piezas a manipular.

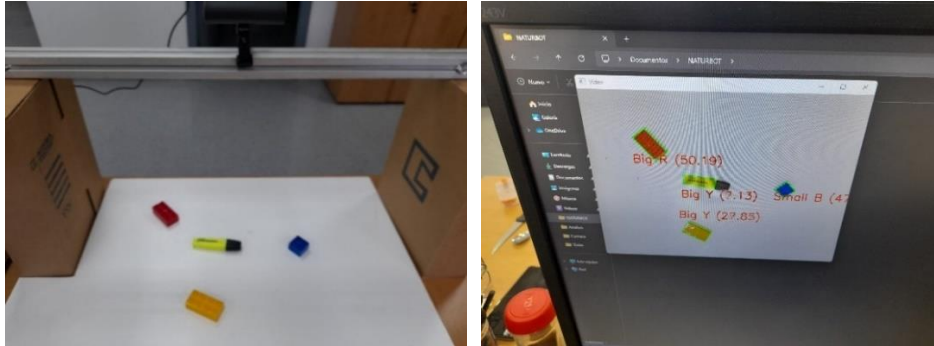


Figura 32: Pruebas con un rotulador rectangular amarillo  
Fuente: elaboración propia.

Tras las pruebas realizadas se realizó un ajuste más fino del sistema para evitar este tipo de resultados.

## 2.3 PT6. DESARROLLO DEL SISTEMA DE COMUNICACIÓN. REALIZACIÓN DE PRUEBA PILOTO

### 2.3.1 Tarea 6.1 Programación de instrucciones elementales

Se ha definido la secuencia de operaciones estandarizada que el robot deberá de ejecutar cada vez que el usuario emita una orden verbal relacionada con utilizar una nueva pieza o devolver una ya utilizada.

Esta secuencia estándar parte de una posición “home” del robot. A continuación, el robot se moverá acercándose a la mesa de trabajo (punto de aproximación 1), y posteriormente se posicionará sobre la mesa de trabajo (punto de aproximación 2). Según el tipo de pieza a recoger identificado por el sistema de reconocimiento del lenguaje natural (PT4) y las coordenadas y orientación definidas por el sistema de reconocimiento del entorno (PT5), se compone un movimiento a la ubicación exacta de la pieza a manipular.

A partir de este punto se activa el agarre de la pieza mediante la herramienta del robot, y se retira al punto “home”, pasando previamente por los puntos de aproximación 1 y 2.

A continuación, el robot debe de avanzar a la posición de entrega de la pieza, que tendrá sus propios puntos de aproximación predefinidos en la segunda mesa de trabajo (aproximación 1' y aproximación 2').

La posición de entrega estará definida por las coordenadas generadas por el sistema de reconocimiento del entorno. En esa posición la herramienta del robot se desactiva para soltar la pieza. Para el caso de uso final, se plantea la posibilidad de que se utilicen unas coordenadas fijas para la posición del usuario, ya que el espacio de trabajo es limitado y no se producen movimientos.

Finalmente, el robot regresa a la posición “home” pasando previamente por los puntos de aproximación 2’ y 1’ respectivamente.

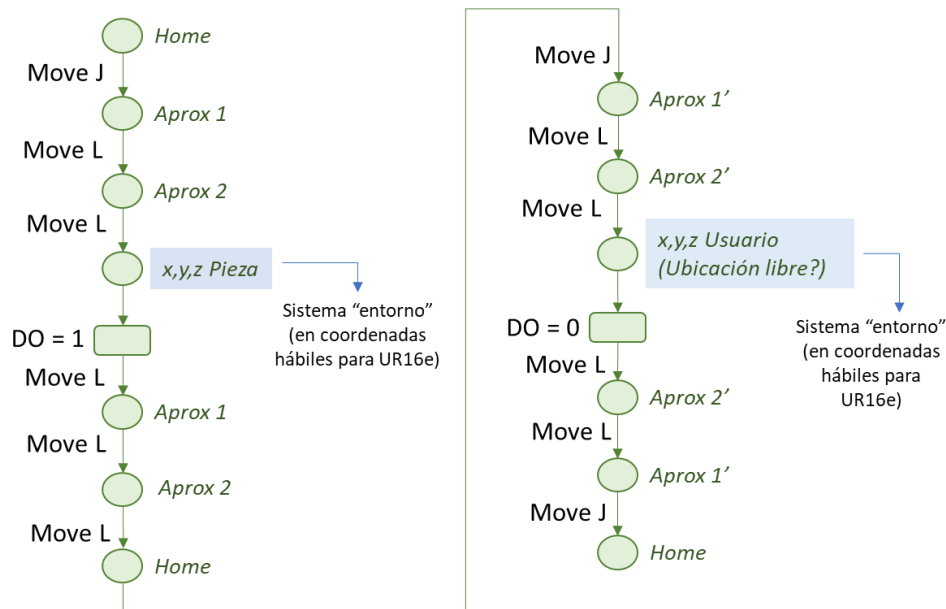


Figura 33: Secuencia de operaciones del robot para ejecutar una instrucción  
Fuente: elaboración propia

Cada vez que el robot tiene que ejecutar una instrucción se repite la secuencia de tareas, actualizándose las coordenadas de la pieza a manipular.

### 2.3.2 Tarea 6.2 Desarrollo de software de comunicación

Los robots de Universal Robotics permiten su control de forma remota sin necesidad de utilizar la consola de programación TeachPendant. Para ello se dispone de los siguientes protocolos accesibles a través de diferentes puertos de comunicación TCP/IP.

#### 1. Dashboard.

Este protocolo de comunicaciones funciona en el puerto TCP 29999 y permite mediante el envío de comandos en formato texto la realización de las siguientes operaciones:

- Carga y ejecución de programas
- Puesta en marcha de los motores y liberación de frenos
- Consultar el estado del robot
- Especificar el modo de operación manual o automático del robot

Como este protocolo requiere el almacenamiento previo del programa a ejecutar en el almacenamiento local del robot como requisito necesario antes de iniciar su ejecución de forma remota no resulta de utilidad en el proyecto.

#### 2. RTDE.

Protocolo propietario de UR para el intercambio de datos en tiempo real. Este protocolo proporciona una manera de sincronizar aplicaciones externas con el controlador UR a través de una conexión TCP/IP estándar, a través del puerto TCP 30004, sin alterar ninguna propiedad en tiempo real del controlador UR. RTDE se divide en dos etapas: el procedimiento de configuración y el bucle de sincronización.

El procedimiento de configuración es logrado a través de un fichero XML, o paquete de datos, con las especificaciones necesarias que crea la combinación de entradas y salidas de datos del controlador necesarias para el funcionamiento de la aplicación. Una vez que el robot ha recibido la configuración debe iniciarse el bucle de sincronización. Esto es, un programa almacenado localmente en la controladora del robot que evalúe los datos recibidos, ejecute las acciones necesarias y envíe al control remoto los datos del estado del robot convenientemente actualizados tras cada acción.

Al igual que el protocolo anterior, como el protocolo RTDE también necesita del almacenamiento previo en la controladora del robot del programa que realiza el bucle de sincronización se descarta su utilización en el proyecto.

### 3. Interfaz Secundaria de programación.

Accesible a través del puerto de comunicaciones TCP 30002. Esta interfaz permite el envío de comandos o programas URScript directamente a la controladora del robot. También permite el envío de datos desde el robot a través de un flujo constante que debe ser analizado mediante un programa personalizado externo. Sin embargo, como el orden y significado de los bits de datos está sujeto a cambios producidos en el robot al aplicar actualizaciones de su software, el fabricante recomienda la utilización adicional de un protocolo de intercambio de datos como por ejemplo MODBUS o RTDE para el intercambio de datos de una forma más organizada.

Como este protocolo no necesita del almacenamiento local en la controladora del robot de ningún programa se opta por esta alternativa en el desarrollo del proyecto. Adicionalmente escogemos el protocolo MODBUS para el intercambio de datos entre la aplicación de control externa al robot y la controladora UR.

Se ha configurado la instalación del robot especificando los registros de datos a intercambiar entre el robot y la aplicación externa.

Para ello se indican los siguientes registros.

- Registro que indica si un programa ha finalizado su ejecución
- Posición de la herramienta del robot
- Posición de las articulaciones del robot.

●	Register Output	7	Joint0	0	Delete
●	Register Output	8	Joint1	51404	Delete
●	Register Output	9	Joint2	47360	Delete
●	Register Output	10	Joint3	50460	Delete
●	Register Output	11	Joint4	15712	Delete
●	Register Output	12	Joint5	0	Delete

Figura 34: Relación entre las articulaciones del robot y los registros MODBUS  
Fuente: elaboración propia

Durante el desarrollo del proyecto se constató que la precisión de los registros MODBUS que es de 16 bits no era suficiente para publicar la posición de la herramienta del robot. Debido a este motivo se amplió la precisión a 32 bits obligando a utilizar dos registros para cada uno de los datos. De esta forma se evitó el truncamiento de los datos numéricos.

●	Register Output	13	TcpX0	64535	Delete
●	Register Output	14	TcpX1	65535	Delete
●	Register Output	15	TcpY0	63536	Delete
●	Register Output	16	TcpY1	65535	Delete
●	Register Output	17	TcpZ0	5999	Delete
●	Register Output	18	TcpZ1	0	Delete

Figura 35: dos registros para cada dato  
Fuente: elaboración propia

Para el intercambio de datos entre la controladora del robot y la aplicación externa que la gobernará es necesaria la utilización de un servidor MODBUS. Este protocolo de intercambio de datos permite la consulta o modificación de los datos publicados mediante una arquitectura cliente servidor a través del puerto de comunicaciones TCP 502.

Se ha desarrollado una aplicación de control del robot cuyo objetivo es generar el programa a ejecutar en la controladora del robot de las órdenes recibidas por el controlador. Para ello obtiene las posiciones tanto del objeto a recoger como de la posición del usuario donde se entregará la pieza a través de la consulta al controlador mediante una API de los datos en formato JSON necesarios para generar el programa. De forma simultánea la aplicación informa al controlador de la etapa que se encuentra en ejecución en el robot, así como de los errores que se pudieran detectar.

Los datos recibidos en formato JSON se componen de un código identificador de orden, de la posición donde se encuentra el objeto a recoger y de la posición más próxima al usuario donde tiene que entregar el objeto recogido.

Para la ejecución del programa debe habilitarse el control remoto del mismo a través de la consola TeachPendant. Una vez iniciada la secuencia se puede observar en la consola TeachPendat el estado de ejecución del programa. También se confirma que a pesar de estar en ejecución no aparece ningún programa cargado en la consola debido a la ejecución remota.

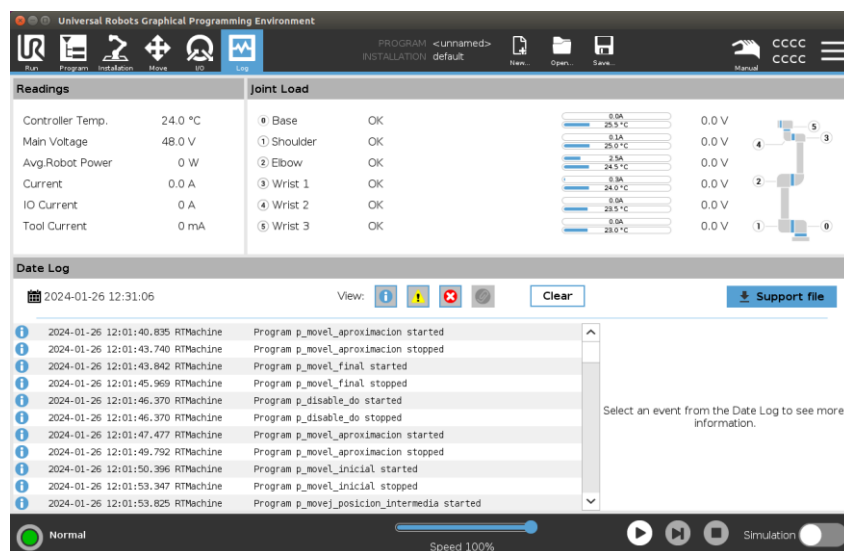


Figura 36: registro de incidencias

Fuente: elaboración propia

Alternativamente, en el PC donde se encuentra la aplicación que gobierna el robot podemos observar la siguiente traza de ejecución.

```

INFO:root:Inicio orden coger nº 3483478347
INFO:root:-----start-----
INFO:root:inicial
INFO:root:Sending command
INFO:root:Posicion alcanzada:
INFO:root:p dest: ['-0.100000', '-0.200000', '0.600000', '0.001000', '-3.142000', '0.000000']
INFO:root:p act : ['-0.100000', '-0.200000', '0.600000', '-0.001000', '3.141100', '0.000000']
INFO:root:p dif : ['0.000000', '0.000000', '0.000000', '0.000000', '-0.000900', '0.000000']
INFO:root:----- end -----
INFO:root:-----start-----
INFO:root:aproximacion
INFO:root:Sending command
INFO:root:Posicion alcanzada:
INFO:root:p dest: ['-0.100000', '-0.200000', '0.200000', '0.001000', '-3.142000', '0.000000']
INFO:root:p act : ['-0.100000', '-0.200100', '0.200000', '-0.001000', '3.141100', '0.000000']
INFO:root:p dif : ['0.000000', '0.000100', '0.000000', '0.000000', '-0.000900', '0.000000']
INFO:root:----- end -----
INFO:root:-----start-----
INFO:root:punto final
INFO:root:Sending command
    
```

Figura 37: traza de ejecución del robot en el PC

Fuente: elaboración propia



Donde entre otra información se muestra el error producido en un movimiento de la secuencia. Un ejemplo de error producido en la imagen anterior es el siguiente.

```
INFO:root:p dif : ['0.000000', '0.000100', '0.000000', '0.000000', '-0.000900', '-0.000000']
```

Por último, podemos observar la actualización en el controlador (interfaz del usuario) del estado en la ejecución de cada uno de los comandos recibidos.

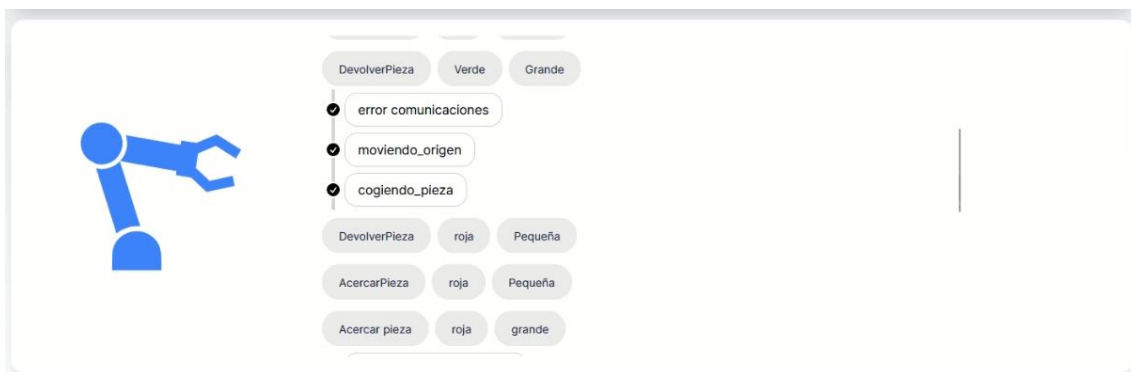


Figura 38: ejecución en la interfaz de usuario desarrollada

Fuente: elaboración propia

Se han realizado pruebas preliminares del sistema de comunicaciones desarrollado. Estas pruebas han consistido en la definición de instrucciones de agarre y entrega de piezas desde diferentes puntos de la mesa de trabajo 1 hasta la mesa de trabajo 2.

El robot ha ejecutado correctamente las secuencias de instrucciones, partiendo de su punto de inicio (HOME), recogiendo la pieza en una mesa, entregándola en la segunda mesa, y volviendo al punto de inicio.

En las siguientes imágenes se muestran algunos instantes de las pruebas realizadas.

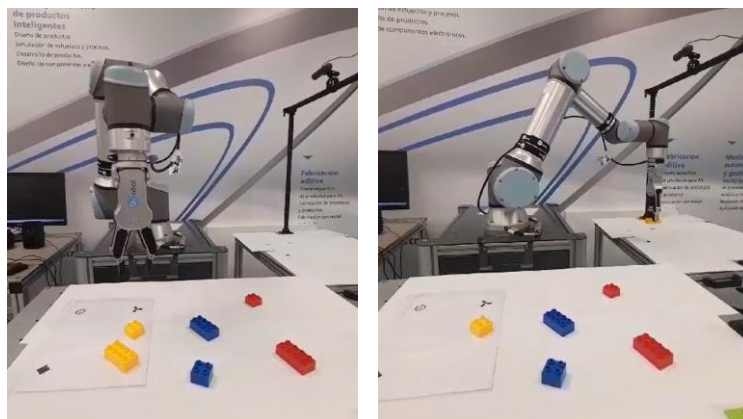


Figura 39: Robot recogiendo y entregando una pieza ubicada en una posición conocida

Fuente: elaboración propia

### 2.3.3 Tarea 6.3 Desarrollo del demostrador

#### 2.3.3.1 *Diseño y montaje del demostrador*

En los paquetes de trabajo PT4 y PT5 se generaron los inputs necesarios para definir el piloto demostrador. Las tareas donde se consideró que los resultados de NATURBOT mejoran una aplicación de robótica colaborativa fueron: “Selección y montaje de componentes”, “Inspección de piezas”, y “Alimentación de piezas a diferentes líneas”. Estas aplicaciones se corresponden con actividades donde el tipo de colaboración es “físico indirecta” o “temporal”. Es decir, robot y humano comparten el mismo espacio de trabajo, pero no al mismo tiempo.

Según el escenario de trabajo definido, el demostrador se compone de un robot, dos mesas de trabajo (una donde están las piezas y otra donde trabaja el operario). En el piloto el usuario que realiza las pruebas no se va a mover, las piezas se pueden ubicar en un punto fijo y por tanto solo es necesaria una cámara para identificar y capturar la posición de las piezas.

El flujo de trabajo se ejecuta de la siguiente manera:

- Un robot trabaja en dos mesas de trabajo (mesa1 y mesa 2).
- En la mesa 2 un operario con un micrófono emite órdenes verbales pidiendo una de las piezas de la mesa 1 con la que quiere trabajar.
- El sistema reconoce la orden verbal.
- La cámara 1 situada sobre un pórtico en la mesa 1, reconoce las coordenadas “x,y” de la pieza y se las pasa al robot.
- El robot recoge la pieza de la mesa 1.
- El robot deposita la pieza en la mesa 2.
- Si el usuario quiere devolver la pieza desde la mesa 2 hacia la mesa 1, emite esa orden verbal y es la cámara 2 de la mesa 2 la que indica las coordenadas de donde está la pieza.

En la siguiente imagen se muestra un ejemplo del montaje final del demostrador.



Figura 40: Piloto demostrativo final  
Fuente: elaboración propia

### 2.3.3.2 *Indicadores para validar la tecnología*

#### Sistema de interpretación del lenguaje natural.

El objetivo del sistema de interpretación del lenguaje natural es identificar si el usuario necesita que el robot le entregue una pieza o bien que el robot le retire la pieza de su zona de trabajo. Además, se requiere interpretar que tipo de pieza concreta se desea que sea manipulada por el robot.

Para la validación del sistema de interpretación del lenguaje natural se han empleado tres indicadores que fueron medidos durante la realización de las pruebas:

- a) **Interpretación de la frase completa.** Hace referencia a la capacidad del sistema para interpretar correctamente la “intention” y las “ranuras” de una frase emitida por el usuario. El resultado del indicador es binario: si se registra un “1”, significa que si lo ha reconocido correctamente; si se registra un “0” el sistema no ha podido reconocer correctamente alguno (o varios) de los parámetros. Por ejemplo, en la frase “dame una pieza amarilla grande” se registraría un “1” si el sistema reconoce la “intention” ENTREGAR, el color AMARILLO, y el tamaño GRANDE. Si falla en alguno (o varios) de estos parámetros se registra un “0”.
- b) **Interpretación parcial de la frase.** Hace referencia a la capacidad del sistema para interpretar correctamente al menos la “intention” o alguna de las “ranuras” de una frase emitida por el usuario. El resultado del indicador es binario: si se registra un “1”, significa que si ha reconocido correctamente algún parámetro; si se registra un “0” el sistema no ha podido reconocer correctamente ninguno de los parámetros. Por ejemplo, en la frase “dame una pieza amarilla grande” se registraría un “1” si el sistema reconoce la “intention” ENTREGAR, el color AMARILLO, y el tamaño GRANDE. Si falla en alguno de estos parámetros se registra también un “1”. Solo cuando falla en todos los parámetros se registra un “0”.
- c) **Tiempo de reconocimiento.** Hace referencia al número de veces que hay que repetir una sentencia. Se ha considerado que, si hay que emitir 1 o 2 veces una frase se registra un “1”, asimilándose a un tiempo de reconocimiento breve. Si hay que emitir más veces la sentencia se registra un “0”, asimilándose a un tiempo de respuesta elevado.

#### Sistema de descripción del contexto.

El objetivo del sistema de descripción del contexto es identificar correctamente cada una de las piezas y extraer sus coordenadas y ángulo de giro en el plano de trabajo, en base al eje de referencia del robot que tiene que manipular las piezas.

Para la validación del sistema se definen dos indicadores que fueron medidos durante la realización de las pruebas:

- d) **Identificación de la pieza.** Hace referencia a la capacidad del sistema para reconocer una pieza. Para cada prueba donde el sistema de voz y de comunicaciones ha funcionado correctamente se asigna un “1” si la pieza ha sido reconocida satisfactoriamente y un “0” si no es así.
  
- e) **Coordenadas de pieza.** Hace referencia a la idoneidad de las coordenadas y ángulo de agarre generados para que una pieza pueda ser manipulada por el robot. Con cada prueba se asigna un “1” si el robot puede correctamente agarrar la pieza y un “0” si no la puede agarrar.

#### Sistema de comunicaciones

El objetivo del sistema de comunicaciones es construir y transmitir las instrucciones de movimiento y agarre de pieza al robot según las órdenes emitidas por voz, y la información de contexto capturada por cámara.

Para la validación del sistema se define un indicador que fue medido durante la realización de las pruebas:

- f) **Instrucciones comunicadas.** Hace referencia a la capacidad del sistema para generar y transmitir instrucciones de robot, según los inputs generados por el sistema de interpretación del lenguaje natural y de descripción del contexto. Para cada prueba donde el sistema de voz y de descripción del contexto ha funcionado correctamente se asigna un “1” si las instrucciones de trabajo han sido emitidas al robot y un “0” si no es así.

#### *2.3.3.3 Pruebas de funcionamiento*

Se han identificado seis variables que pueden interferir en el resultado de las pruebas:

- Usuario. En función de la persona que realice las pruebas el volumen y velocidad de la voz o el acento pueden diferir. Los usuarios de estas pruebas son personal técnico de AIDIMME participante en el proyecto.
- “intention”. Son dos las acciones que puede demandar el usuario al robot: que le entregue una pieza, o que la devuelva a su mesa de trabajo original.
- Color de la pieza. El usuario podrá solicitar acciones sobre tres colores de pieza: roja, amarilla o azul.
- Tamaño de pieza. El usuario podrá solicitar acciones sobre dos tamaños de pieza: grande y pequeña.
- Tipo de ruido. Uno de los factores que se ha identificado pueden afectar a la eficacia y eficiencia del sistema es el nivel de ruido ambiente cuando se están emitiendo las órdenes verbales. Se planifica realizar las pruebas en un entorno de laboratorio sin ruido, con ruido de maquinaria y con ruido de voces humanas.
- Intensidad de ruido. El volumen del ruido será fijo durante las pruebas.

En la siguiente tabla se resumen los parámetros considerados en las pruebas.

Parámetros	Tipo	Posibles valores
Usuario	Características del propio usuario	<<Personal participante en el proyecto>>
Intención	Características de la orden a emitir	Entregar pieza, Devolver pieza
Color pieza	Características de la orden a emitir	Roja, Amarilla, Azul
Tamaño pieza	Características de la orden a emitir	Grande, Pequeña
Tipo ruido	Elementos externos	Sin ruido, Ruido maquinaria, Ruido humano
Intensidad Ruido	Elementos externos	Fija

Figura 41: Parámetros de las pruebas  
Fuente: elaboración propia.

Con el objetivo de guiar al usuario durante la realización de las pruebas se han redactado doce frases de ejemplo que cubren las doce posibles acciones que el usuario puede llegar a solicitar al robot. Estas doce frases se basan en dos acciones (entregar o devolver), tres colores de pieza (rojo, amarillo y azul) y dos tamaños de pieza (grande o pequeña).

En la siguiente imagen se muestran las doce frases de ejemplo y su correspondiente acción, color y tamaño de pieza.

Frase ejemplo	Acción	Color pieza	Tamaño pieza
1. "Por favor, tráeme una pieza roja grande."	Entregar	rojo	grande
2. "Necesito utilizar una pieza pequeña de color amarillo, por favor."	Entregar	amarillo	pequeña
3. "¿Podrías conseguirme una pieza azul grande, por favor?"	Entregar	azul	grande
4. "Quisiera una pieza roja pequeña, por favor."	Entregar	rojo	pequeña
5. "¡Hola! ¿Podrías ayudarme a obtener una pieza amarilla grande?"	Entregar	amarilla	grande
6. Dame una pieza azul pequeña	Entregar	azul	pequeña
7. "Por favor, devuelve la pieza roja grande."	Devolver	rojo	grande
8. "Necesito que devuelvas la pieza amarilla pequeña, por favor."	Devolver	amarillo	pequeña
9. "¿Podrías retornar la pieza azul grande, por favor?"	Devolver	azul	grande
10. "Quisiera devolver a su sitio la pieza pequeña roja, por favor."	Devolver	rojo	pequeña
11. "¡Hola! ¿Podrías ayudarme a retornar la pieza amarilla grande?"	Devolver	amarillo	grande
12. Devuelve la pieza pequeña azul	Devolver	azul	pequeña

Figura 42: Frases de ejemplo para las pruebas  
Fuente: elaboración propia.

Para la realización de las pruebas y registro de datos se prepara una plantilla donde para cada usuario que realiza las pruebas se definen treinta y seis frases que se corresponden a las doce frases definidas anteriormente, multiplicadas por tres diferentes escenarios de ruido: sin ruido de fondo, con ruido de maquinaria y con ruido humano.

Las pruebas han sido llevadas a cabo por cinco usuarios (técnicos de AIDIMME participantes en el proyecto).

Los diferentes niveles de ruido han sido generados mediante archivos de audio que reproducen ruido de maquinaria industrial y ruido de voces humanas respectivamente.

Los archivos de audio empleados se pueden encontrar en las siguientes direcciones web.

- Ruido de maquinaria industrial: [https://www.youtube.com/watch?v=ed\\_KdQ77K10](https://www.youtube.com/watch?v=ed_KdQ77K10)

- Ruido de voces humanas: <https://www.youtube.com/watch?v=ANcYacARSdU>

En la propia pantalla principal de la interfaz de usuario desarrollada se puede comprobar si la acción verbalizada por el usuario (“intention”, color y tamaño) ha sido reconocida correctamente por el sistema y cuantas iteraciones han hecho falta para ello.

La parte de reconocimiento de piezas, cálculo de coordenadas, ángulo de agarre y comunicaciones se comprueba visualmente tras la ejecución de la tarea por parte del robot.

#### 2.3.4 Tarea 6.4 Validación del sistema y conclusiones

##### 2.3.4.1 Resultados de las pruebas

###### 2.3.4.1.1 Sistema de interpretación del lenguaje natural

De manera general los tres indicadores propuestos para la validación quedan calculados de forma promedio tal y como se muestran en la siguiente tabla.

Id	Indicador	Valor	Unidades
a	Interpretación de la frase completa	48'6%	%
b	Interpretación parcial de la frase	84'0	%
c	Tiempo de reconocimiento	76'4	%

Figura 43: Resultados KPIs de validación (i)

Fuente: elaboración propia.

La interpretación de la frase completa queda por debajo del 50% promedio (48'6%), y la interpretación parcial de la frase y el tiempo de reconocimiento en el entorno del 80% (84'0% y 76'4% respectivamente).

Sin embargo, el valor de los indicadores son variables en función de la agrupación de resultados que se hace según los diferentes parámetros con los que se configuraron las pruebas. A continuación, se muestran los resultados agrupados según las diferentes variables de validación.

#### Análisis por “INTENTION”.

La siguiente gráfica muestra el resultado de los tres indicadores agrupado por promedio según la “intention” a reconocer: Devolver una pieza o Entregar una pieza.

En la gráfica se observa como el reconocimiento completo de la frase es mucho menor en la “intention” devolver que en la “intention” entregar. Esto indica que debe de realizarse un ajuste fino del entrenamiento de del reconocimiento de esta expresión. El resto de los indicadores son similares en ambos casos.

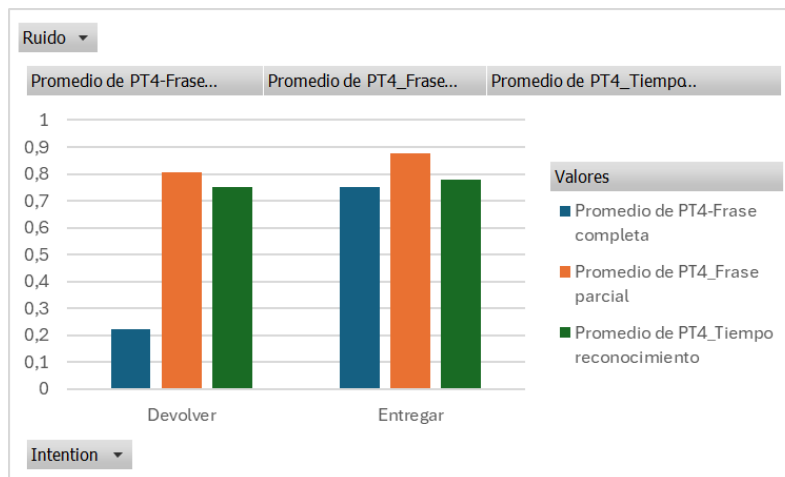


Figura 44: Resultado promedio agrupado por “INTENTION”  
Fuente: elaboración propia

#### Análisis por COLOR.

En el caso del color no se han encontrado grandes diferencias entre ninguno de los tres colores, aunque en el caso del color amarillo todos los indicadores son algo más bajos.

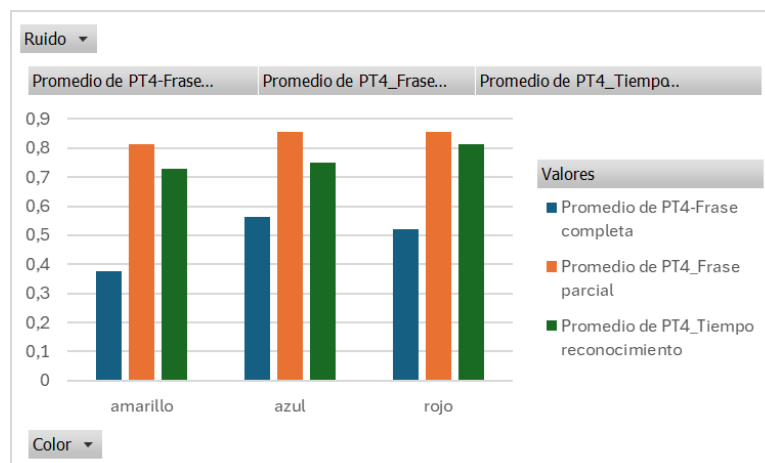


Figura 45: Resultado promedio agrupado por COLOR  
Fuente: elaboración propia

#### Análisis por TAMAÑO.

En cuanto al promedio por tamaño tampoco se han identificado grandes diferencias.

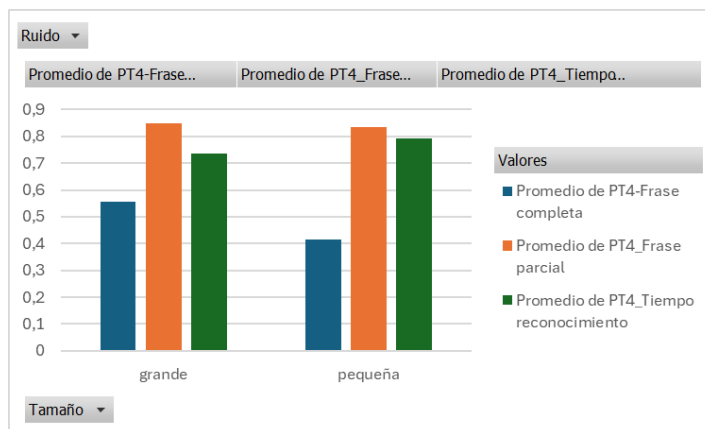


Figura 46: Resultado promedio agrupado por TAMAÑO  
Fuente: elaboración propia

Análisis por RUIDO.

Si se analiza por nivel de ruido se comprueba como tanto el ruido industrial, como el ruido humano afectan al sistema, siendo este último el que mayor distorsión genera.

Si bien, en el promedio de reconocimiento de frase completa no se aprecia grandes diferencias, en los otros dos indicadores si que hay un deterioro con ruido.

Ruido	NO			
<b>Etiquetas de fila</b>	<b>Promedio de PT4-Frase completa</b>	<b>Promedio de PT4_Frase parcial</b>	<b>Promedio de PT4_Tiempo reconocimiento</b>	
1. "Por favor, devuelve la pieza roja grande."	0,8	1,0	1,0	
1. "Por favor, tráeme una pieza roja grande."	1,0	1,0	1,0	
2. "Necesito que devuelvas la pieza amarilla pequeña, por favor."	0,0	1,0	1,0	
2. "Necesito utilizar una pieza pequeña de color amarillo, por favor."	0,8	1,0	1,0	
3. "¿Podrías conseguirme una pieza azul grande, por favor?"	1,0	1,0	1,0	
3. "¿Podrías retornar la pieza azul grande, por favor?"	0,0	0,8	0,8	
4. "Quisiera devolver a su sitio la pieza pequeña roja, por favor."	0,0	1,0	1,0	
4. "Quisiera una pieza roja pequeña, por favor."	0,8	1,0	1,0	
5. "¡Hola! ¿Podrías ayudarme a obtener una pieza amarilla grande?"	1,0	1,0	0,5	
5. "¡Hola! ¿Podrías ayudarme a retornar la pieza amarilla grande?"	0,0	1,0	1,0	
6. Dame una pieza azul pequeña	0,8	0,8	0,8	
6. Devuelve la pieza pequeña azul	0,8	1,0	1,0	
<b>Total general</b>	<b>0,5625</b>	<b>0,958333333</b>	<b>0,916666667</b>	
<b>Ruido</b>	<b>INDUSTRIAL</b>			
<b>Etiquetas de fila</b>	<b>Promedio de PT4-Frase completa</b>	<b>Promedio de PT4_Frase parcial</b>	<b>Promedio de PT4_Tiempo reconocimiento</b>	
1. "Por favor, devuelve la pieza roja grande."	0,5	1,0	1,0	
1. "Por favor, tráeme una pieza roja grande."	0,5	0,8	0,5	
2. "Necesito que devuelvas la pieza amarilla pequeña, por favor."	0,0	0,8	0,8	
2. "Necesito utilizar una pieza pequeña de color amarillo, por favor."	0,3	0,8	0,8	
3. "¿Podrías conseguirme una pieza azul grande, por favor?"	1,0	1,0	0,8	
3. "¿Podrías retornar la pieza azul grande, por favor?"	0,0	0,8	0,5	
4. "Quisiera devolver a su sitio la pieza pequeña roja, por favor."	0,3	0,5	0,8	
4. "Quisiera una pieza roja pequeña, por favor."	0,8	0,8	0,8	
5. "¡Hola! ¿Podrías ayudarme a obtener una pieza amarilla grande?"	0,8	0,8	0,8	
5. "¡Hola! ¿Podrías ayudarme a retornar la pieza amarilla grande?"	0,3	0,8	0,5	
6. Dame una pieza azul pequeña	1,0	1,0	0,8	
6. Devuelve la pieza pequeña azul	0,0	1,0	0,8	
<b>Total general</b>	<b>0,4375</b>	<b>0,8125</b>	<b>0,708333333</b>	



Ruido	CONVERSACIÓN		
<b>Etiquetas de fila</b>	<b>Promedio de PT4-Frase completa</b>	<b>Promedio de PT4_Frase parcial</b>	<b>Promedio de PT4_Tiempo reconocimiento</b>
1. "Por favor, devuelve la pieza roja grande."	0,5	1,0	1,0
1. "Por favor, tráeme una pieza roja grande."	0,5	0,8	0,5
2. "Necesito que devuelvas la pieza amarilla pequeña, por favor."	0,0	0,5	0,5
2. "Necesito utilizar una pieza pequeña de color amarillo, por favor."	0,8	1,0	1,0
3. "¿Podrías conseguirme una pieza azul grande, por favor?"	1,0	0,8	0,8
3. "¿Podrías retornar la pieza azul grande, por favor?"	0,5	0,8	0,8
4. "Quisiera devolver a su sitio la pieza pequeña roja, por favor."	0,0	0,5	0,3
4. "Quisiera una pieza roja pequeña, por favor."	0,8	1,0	1,0
5. "¡Hola! ¿Podrías ayudarme a obtener una pieza amarilla grande?"	0,8	0,8	0,5
5. "¡Hola! ¿Podrías ayudarme a retornar la pieza amarilla grande?"	0,0	0,5	0,5
6. Dame una pieza azul pequeña	0,3	0,8	0,8
6. Devuelve la pieza pequeña azul	0,5	0,8	0,5
<b>Total general</b>	<b>0,458333333</b>	<b>0,75</b>	<b>0,666666667</b>

Figura 47: Resultado promedio agrupado por RUIDO  
Fuente: elaboración propia

### 2.3.4.1.2 Sistema de descripción del contexto

Durante las pruebas realizadas todas las piezas han sido correctamente reconocidas.

Id	Indicador	Valor	Unidades
d	Identificación de la pieza.	100%	%

Figura 48: Resultados KPIs de validación (ii)  
Fuente: elaboración propia.

El cálculo de coordenadas realizado ha sido suficiente para realizar el agarre de las piezas.

Id	Indicador	Valor	Unidades
e	Coordenadas de pieza.	100%	%

Figura 49: Resultados KPIs de validación (iii)  
Fuente: elaboración propia.

Sin embargo, se detectó que con el paso de las iteraciones se han producido pequeñas desviaciones que han hecho que el robot no agarrara la pieza por su centro geométrico.

En la siguiente imagen se muestra un ejemplo de la pinza agarrando una pieza por su centro geométrico.



Figura 50: Ejemplo de pieza agarrada por su centro geométrico  
Fuente: elaboración propia

En las dos siguientes imágenes se muestra un ejemplo de piezas que no fueron agarradas por su centro geométrico, indicando que el cálculo de coordenadas de la pieza ha sufrido un pequeño error.

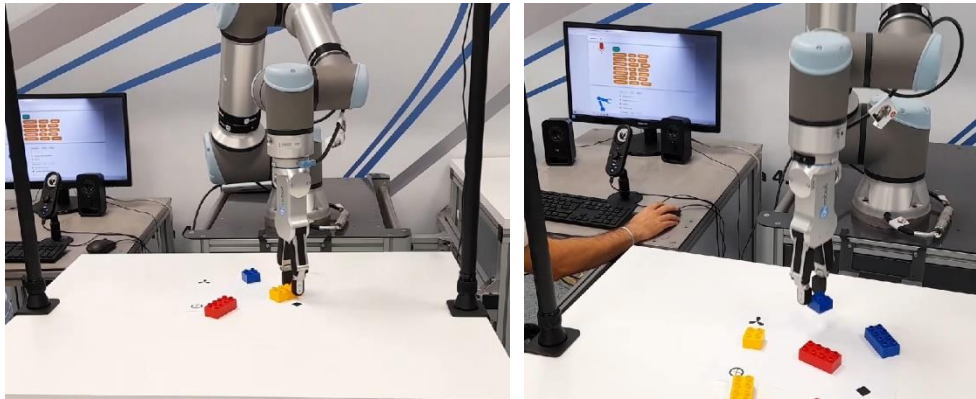


Figura 51: Ejemplo de piezas donde el centro calculado se ha desviado del centro real  
Fuente: elaboración propia

Se ha detectado que mínimos movimiento de la mesa de trabajo respecto del robot o de la cámara respecto de la mesa de trabajo obligan a recalibrar todo el sistema para evitar este tipo de desviaciones.

El sistema de montaje de las cámaras se ha realizado mediante bridas que se fijan y ajustan manualmente a los soportes de las mesas. Este sistema es muy sensible a pequeños golpes o movimientos involuntarios de un usuario en las mesas de trabajo que provocan a su vez pequeños desplazamientos en el ángulo de la cámara respecto del plano de trabajo de la mesa, y por tanto induciendo un pequeño error en el cálculo de coordenadas. Estas desviaciones han podido ser corregidas con una recalibración del sistema.

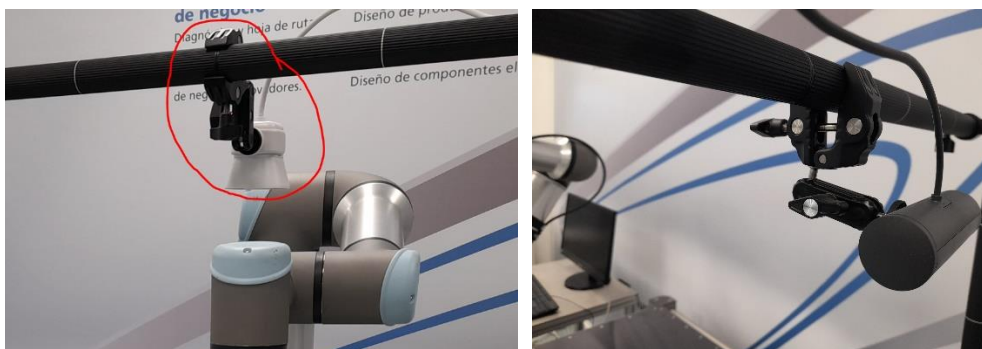


Figura 52: Sujeción de cámaras con bridas a los soportes  
Fuente: elaboración propia

#### 2.3.4.2 Sistema de comunicaciones

Durante las pruebas realizadas el sistema de comunicaciones ha transmitido en todos los casos el 100% de las instrucciones reconocidas por el sistema de interpretación del lenguaje natural, recogiendo las piezas según los datos generados por el sistema de descripción del contexto.

Id	Indicador	Valor	Unidades
f	Instrucciones comunicadas	100%	%

Figura 53: Resultados KPIs de validación (iv)

Fuente: elaboración propia.

No se ha detectado ninguna incidencia en el sistema, funcionando satisfactoriamente.

### 3 Resultados obtenidos

Como resultado de la ejecución de las actividades ejecutadas en el proyecto se han generado los siguientes sistemas:

- Se ha desarrollado un sistema que extrae instrucciones de bajo nivel, que permiten gestionar un robot colaborativo, a partir de expresiones en lenguaje natural utilizadas en tareas colaborativas.
- Se ha desarrollado un sistema de reconocimiento de piezas de diferente color y tamaño, que contextualiza las instrucciones enviadas al robot y obtiene la posición de cada pieza en coordenadas base para el sistema robot.
- Se han desarrollado elementos software de comunicación entre los sistemas anteriores y el control del robot, que permiten transferir los comandos adecuados para ejecución de tareas de manipulación de piezas entre dos puestos de trabajo.
- Se ha desarrollado un demostrador piloto de los desarrollos realizados, para mostrar a las empresas la aplicación de la tecnología.

## 4 Resumen y conclusiones

Como conclusión, con el proyecto NATURBOT se ha desarrollado un sistema que permite de manera sencilla e intuitiva interactuar con un robot colaborativo mediante lenguaje natural por voz, de manera que el humano y el robot puedan colaborar en tareas como el montaje de componentes.

Tras las pruebas de validación llevadas a cabo se han identificado dos posibles áreas de mejora del sistema desarrollado.

- Entrenar el sistema de interpretación del lenguaje natural con una mayor cantidad de expresiones, frases y sinónimos para mejora el porcentaje de acierto en la interpretación de la “intention” DEVOLVER una pieza.
- Realizar en una única bancada solidaria el montaje de robot, mesas de trabajo y cámara. De esta manera, se evitarían calibraciones del sistema debido a pequeños movimientos de la cámara o de las mesas de trabajo respecto del robot.

A la vista de los resultados obtenidos, el sistema NATURBOT se ha mostrado como un prototipo de tecnología que puede hacer que un usuario no experto en programación, pueda interactuar con un robot para ejecutar tareas no repetitivas de forma fácil e intuitiva.



# **AIDIMME**

**INSTITUTO TECNOLÓGICO**

Domicilio fiscal —

C/ Benjamín Franklin 13. (Parque Tecnológico)  
46980 Paterna. Valencia (España)  
Tlf. 961 366 070 | Fax 961 366 185

Domicilio social —

Leonardo Da Vinci, 38 (Parque Tecnológico)  
46980 Paterna. Valencia (España)  
Tlf. 961 318 559 - Fax 960 915 446

[aidimme@aidimme.es](mailto:aidimme@aidimme.es)

[www.aidimme.es](http://www.aidimme.es)