ENTREGABLE
PROYECTOS—
2023-2024

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

Entregable: E6.1. Sistema de comunicación desarrollado

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

1 Introducción y objetivos del entregable

1.1 Objeto del entregable

El presente entregable recoge toda la documentación asociada al sistema de comunicación de instrucciones al robot, desarrollado dentro del proyecto.

Este sistema es el encargado de construir a partir de los inputs del PT4 (interpretación de instrucciones emitidas mediante voz) y el PT5 (posición y orientación de piezas y del usuario) instrucciones de trabajo para el robot (movimientos del robot y acciones de su herramienta) que son comunicadas y ejecutadas.

El desarrollo de este sistema es uno de los resultados del proyecto que permite avanzar en los métodos de interacción humano-robot, y genera el componente final de un sistema que integra, además, un sistema de reconocimiento del lenguaje natural por voz (paquete de trabajo PT4) y un sistema de reconocimiento del entorno (paquete de trabajo PT5).

1.2 Objetivos del proyecto con los que se relaciona

Este documento se relaciona con dos de los objetivos específicos identificados en la memoria de solicitud del proyecto:

- Desarrollo de un conjunto de operaciones elementales que pueda realizar un robot colaborativo, y que puedan combinarse para ejecutar operaciones complejas de más alto nivel.
- Desarrollo de los elementos software adecuados para comunicar los subsistemas de interpretación y de reconocimiento del entorno con el control del robot, y transferir los comandos adecuados para ejecución de tareas.

1.3 Paquete de trabajo en el que está encuadrado y principales tareas a desarrollar

El entregable se encuadra en el paquete de trabajo PT6 (Desarrollo del sistema de comunicación. Realización prueba piloto), donde se llevan a cabo las siguientes tareas:

- Tarea 6.1 Programación de instrucciones elementales
- Tarea 6.2 Desarrollo de software de comunicación

1.4 Resumen del trabajo realizado

En primer lugar, se ha definido un flujo de acciones a realizar por el robot cada vez que reciba una instrucción de trabajo. Este flujo de acciones se basa en movimientos de aproximación a las mesas de trabajo en puntos fijos del espacio de trabajo, así como movimientos finales de agarre / liberación de pieza a las coordenadas capturadas por el sistema de reconocimiento del entorno (desarrollo del PT5).

AIDIMME INSTITUTO TECNOLÓGICO · GENERALITAT VALENCIANA · IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN · Financiado por la Unión Europea

En segundo lugar, se analizan los diferentes métodos de interacción con un robot enfocándose en detalle en aquellos que permiten el control de un modelo UR16e de forma remota (dashboard, RTDE e interfaz Secundaria de programación).

Finalmente, se documentan los diferentes pasos llevados a cabo para el desarrollo del sistema completo de generación y comunicación de instrucciones con el robot, detallando la configuración del protocolo modbus en el robot, el servidor modbus, la aplicación de control del robot, y la ejecución del programa.

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN
Financiado por la Unión Europea

ENTREGABLE PROYECTOS

2 Actividades realizadas

2.1 Tarea 6.1 Programación de instrucciones elementales

2.1.1 Definición del flujo de tareas del robot

Según el escenario de trabajo definido en el PT5, se dispone de 1 robot UR16e dos mesas de trabajo (una donde están las piezas y otra donde trabaja el operario) y dos cámaras (cámara 1 para identificar coordenadas de piezas en la mesa 1, cámara 2 para identificar coordenadas de usuario). Para el escenario final, dado que el usuario no se va a mover, las piezas se pueden ubicar en un punto fijo y por tanto no es necesaria la tercera cámara para identificar la posición del usuario.

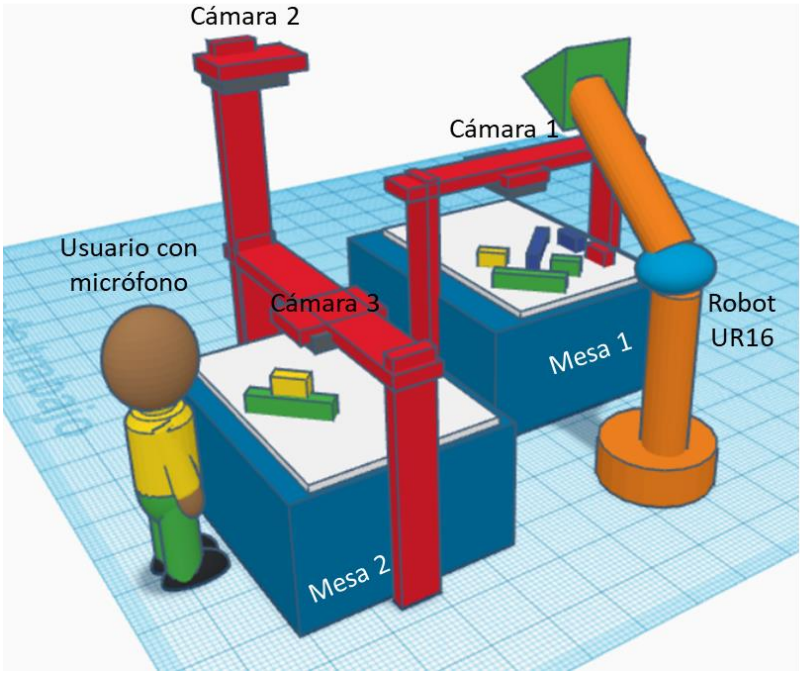

Figura 1: Escenario de trabajo colaborativo
Fuente: elaboración propia

En este escenario el operario trabaja en la mesa 2 y las piezas a manipular están en la mesa 1. El operario pide piezas y el robot las coge de la mesa 1 y las coloca en la mesa 2. Si el operario quiere devolver piezas, el robot las coge de la mesa 2 y las coloca en una posición de la mesa 1.

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).

AIDIMME INSTITUTO TECNOLÓGICO · GENERALITAT VALENCIANA · IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN · Financiado por la Unión Europea

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 partir de aquí, 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.

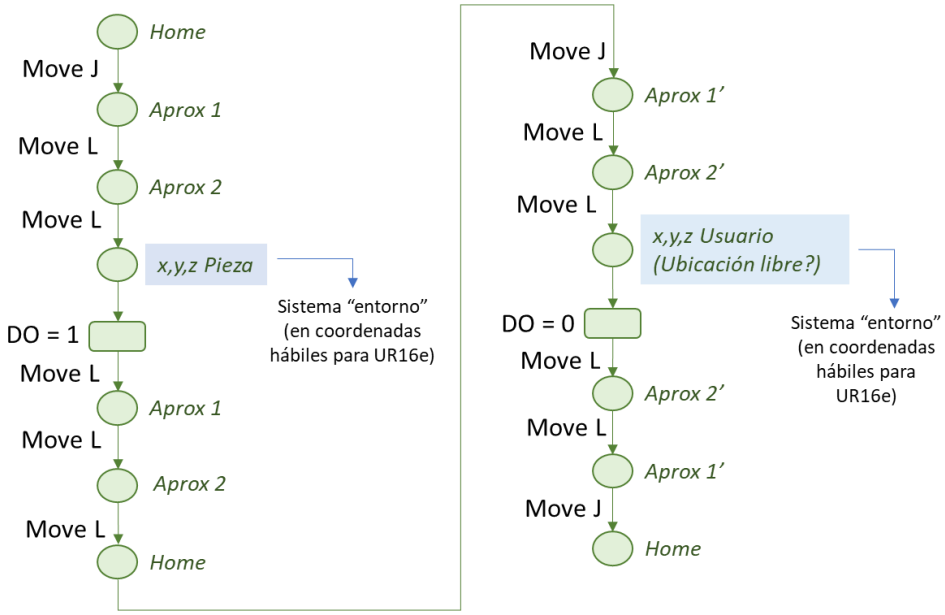

Figura 2: Secuencia de operaciones del robot para ejecutar una instrucción
Fuente: elaboración propia

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

AIDIMME INSTITUTO TECNOLÓGICO · GENERALITAT VALENCIANA · IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN · Financiado por la Unión Europea

A continuación, se muestran imágenes de la definición de coordenadas para los puntos de aproximación.

Punto HOME

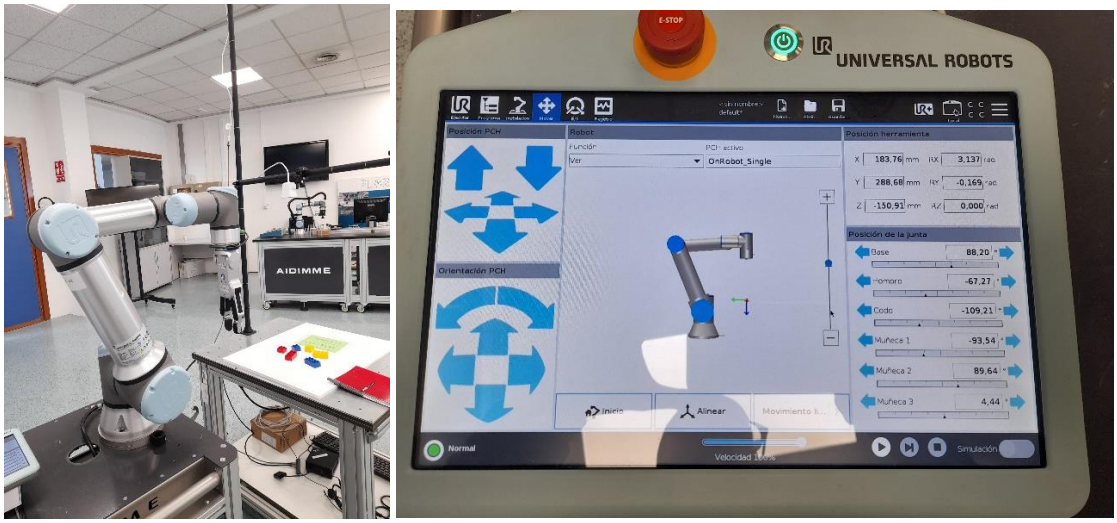

Figura 3: Ubicación y coordenadas del punto HOME
Fuente: elaboración propia

Punto de aproximación 1

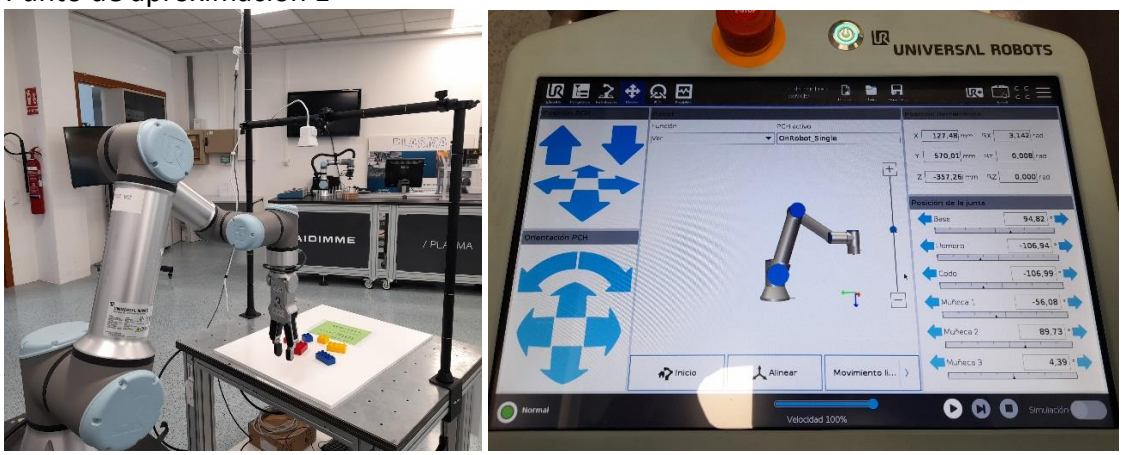

Figura 4: Ubicación y coordenadas del punto de aproximación 2
Fuente: elaboración propia

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN
Financiado por la Unión Europea

Punto de aproximación 2

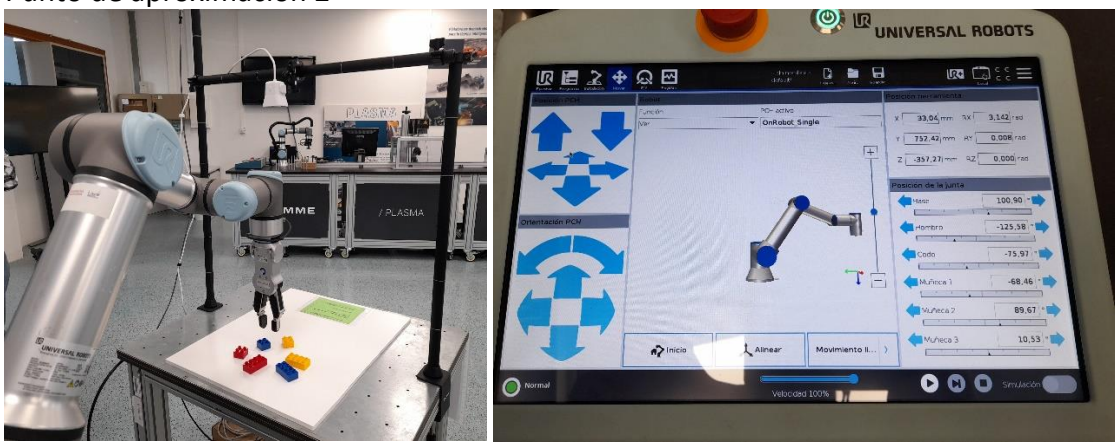

Figura 5: ubicación y coordenadas del punto de aproximación 2
Fuente: elaboración propia

2.2 Tarea 6.2 Desarrollo de software de comunicación

2.2.1 Métodos de interacción general con un robot

A continuación, se enumeran los principales métodos de interacción que un humano tiene con un robot.

1) Interfaz de usuario del Teach Pendant: Es la pantalla táctil y el panel de control del robot que permite programarlo y controlarlo directamente a través de su software integrado.
2) Programación por software (UR Polyscope, ABB Robostudio, RoboDK, etc.): Permite la programación y control del robot UR mediante una interfaz gráfica intuitiva.
3) APIs (Application Programming Interfaces): Permiten a los desarrolladores crear programas personalizados para controlar el robot y facilitar su integración en sistemas automatizados.
4) Conexión a través de redes y protocolos estándar:
   - Ethernet/IP: Protocolo basado en Ethernet, ampliamente utilizado en entornos industriales, ofrece una buena velocidad y compatibilidad con una amplia gama de dispositivos.
   - Modbus TCP/IP: Protocolo conocido por su simplicidad y facilidad de implementación, permite la comunicación entre dispositivos a través de TCP/IP.
   - Profinet: Protocolo basado en Ethernet que ofrece altas velocidades, determinismo y funcionalidades avanzadas para la configuración y diagnóstico de dispositivos.

AIDIMME INSTITUTO TECNOLÓGICO GENERALITAT VALENCIANA IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN Financiado por la Unión Europea

- EtherCAT: Protocolo de alta velocidad y determinismo extremadamente alto, ideal para aplicaciones que requieren tiempos de respuesta rápidos y sincronización precisa.

2.2.2 Métodos de interacción con robot UR de forma remota

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.

2.2.3 Configuración del protocolo Modbus en el robot.

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.

AIDIMME INSTITUTO TECNOLÓGICO GENERALITAT VALENCIANA IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN Financiado por la Unión Europea

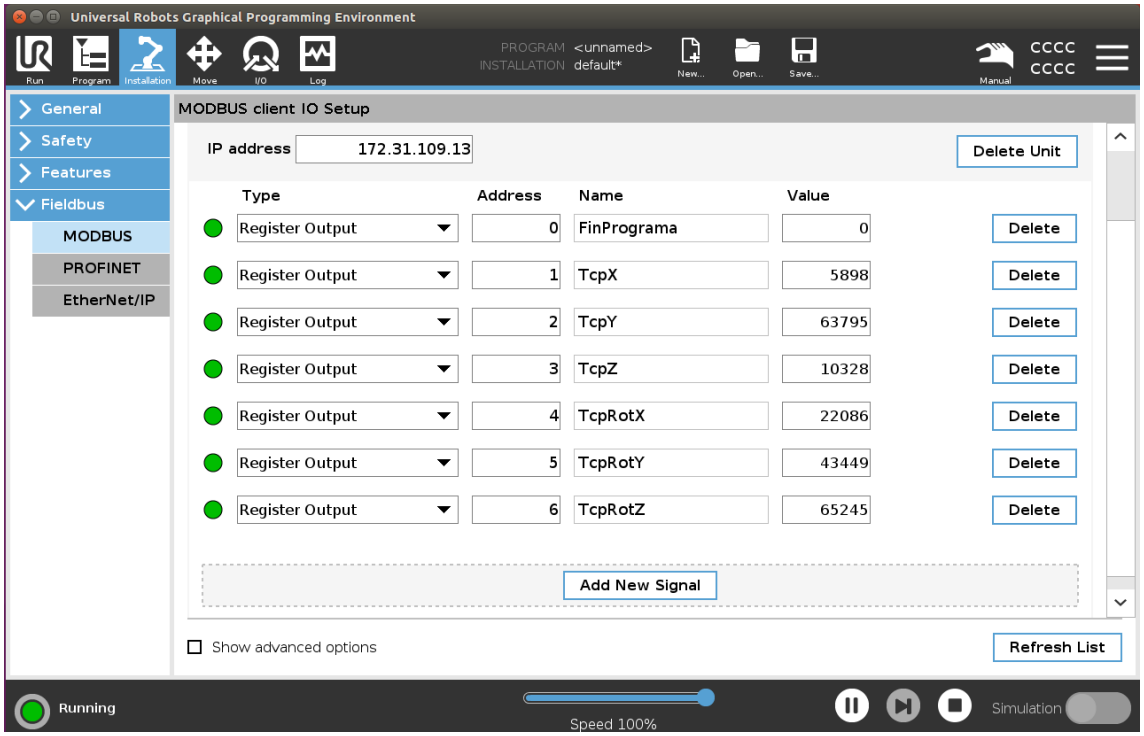

Figura 6: pantalla general Modbus en Polyscope
Fuente: elaboración propia

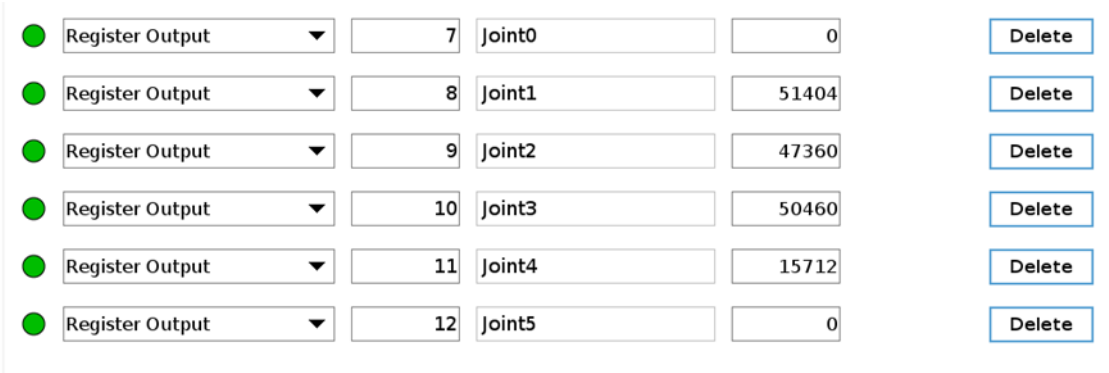

Figura 7: Relación entre las articulaciones del robot y los registros modbus
Fuente: elaboración propia

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.

AIDIMME INSTITUTO TECNOLÓGICO GENERALITAT VALENCIANA IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN Financiado por la Unión Europea

●	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 8: dos registros para cada dato
Fuente: elaboración propia

De esta forma se evitó el truncamiento de los datos numéricos.

2.2.4 Servidor Modbus.

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.

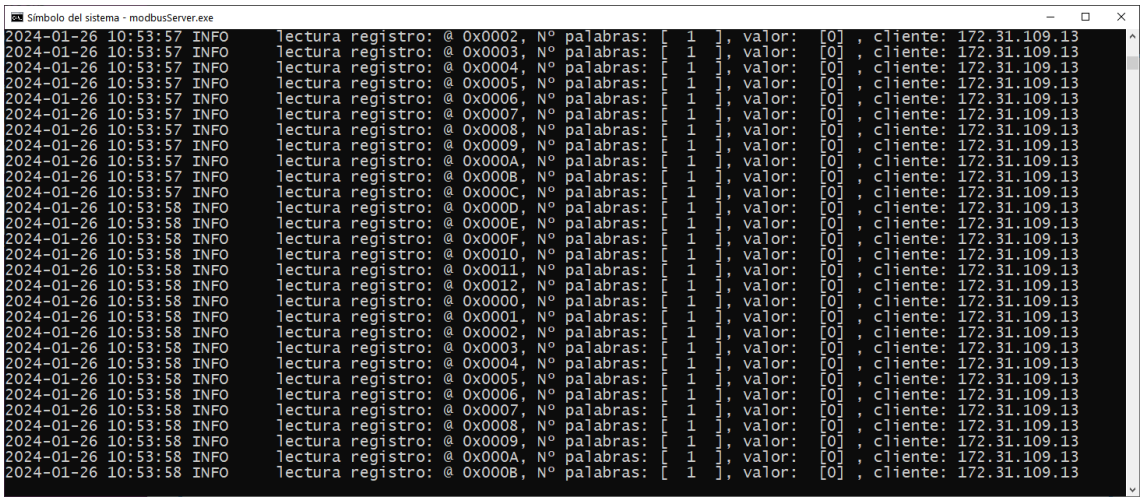

Figura 9: consulta de valores almacenados en el servidor modbus
Fuente: elaboración propia

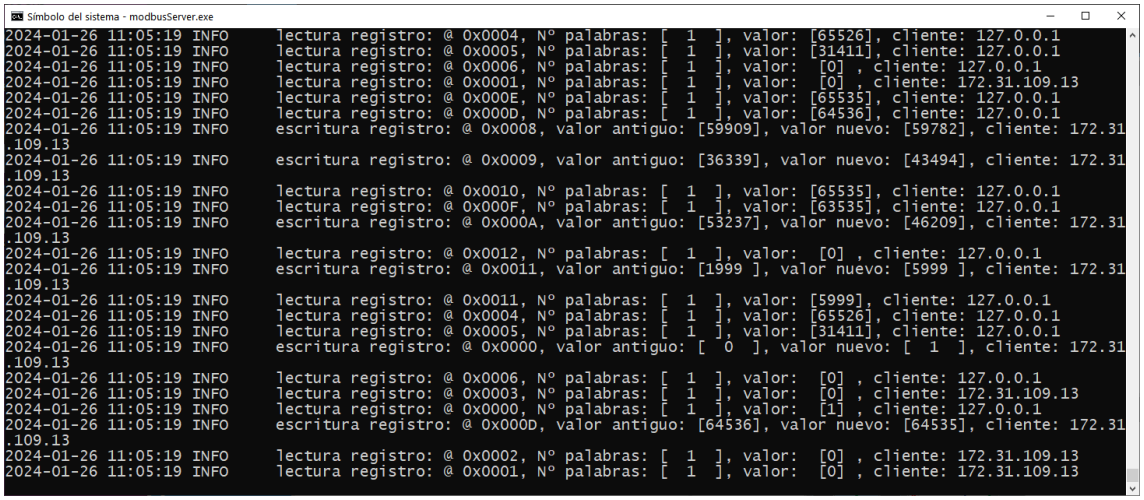

Figura 10: intercambio de valores a través del servidor modbus
Fuente: elaboración propia

2.2.5 Aplicación de control del robot.

El objeto de esta aplicación 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. Tal y como se muestra en la imagen siguiente.

```json
{
    "id": "3483478347",
    "coger": {
        "x": 0.0348,
        "y": 0.7653,
        "z": 0.0225,
        "rot_x": 3.139,
        "rot_y": -0.129,
        "rot_z": 0.0
    },
    "dejar": {
        "x": -0.82974,
        "y": 0.04010,
        "z": -0.04090,
        "rot_x": 2.215,
        "rot_y": 2.228,
        "rot_z": 0.0
    }
}
```

Figura 11: estructura archivo json
Fuente: elaboración propia

La aplicación se estructura en las siguientes capas de funcionamiento:

- Conectividad TCP IP

```
class SecondaryClient:

    def __init__(self, modbusclient, hostname="192.168.0.210", port=30002):···

    def connect(self):···

    def disconnect(self):···

    def is_connected(self):···
```

Figura 12: conectividad TCP IP
Fuente: elaboración propia

- Protocolo Modbus. Lectura de los datos publicados por el servidor Modbus. Finalización del programa y posicionamiento tanto de la herramienta del robot como de las posiciones angulares de cada una de las articulaciones del mismo.

```
    def __modbus2int32(self, msw, lsw):···

    @staticmethod
    def reinterpretSigned16(val):···

    @staticmethod
    def reinterpretSigned32(val):···

    def __get_tcp_pose(self):···

    def __get_joints(self):···

    def is_command_finished(self):···
```

Figura 13: protocolo modbus
Fuente: elaboración propia

- Protocolo de la interfaz secundaria de programación del robot. Envío de programas a la controladora.

```
    def __send_command(self, command):···

    def __get_response(self):···
```

Figura 14: protocolo interfaz secundaria
Fuente: elaboración propia

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN
Financiado por la Unión Europea

- Operaciones básicas de movimiento y control de salidas digitales. Movimientos lineales y angulares.

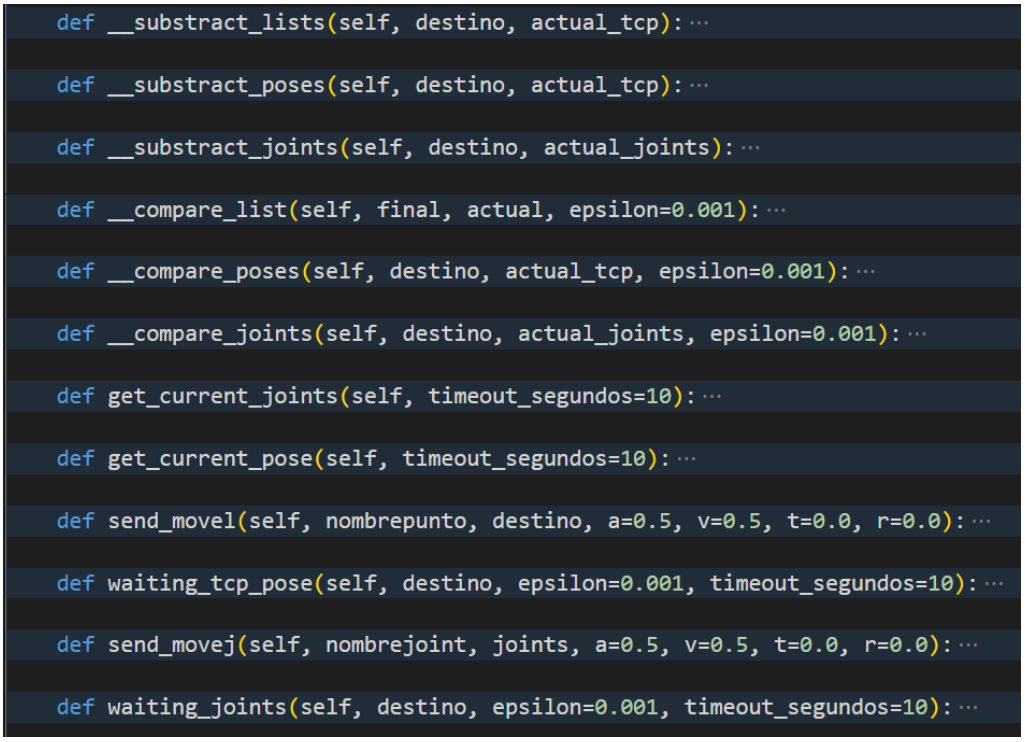

Figura 15: manejo de señales digitales
Fuente: elaboración propia

- Gestión de señales digitales para la operación de la herramienta.

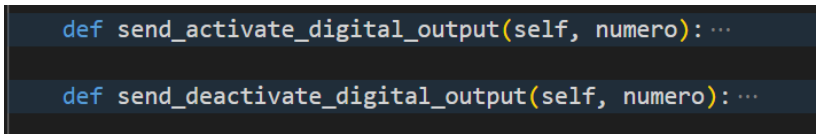

Figura 16: operaciones con la herramienta de agarre
Fuente: elaboración propia

- Controlador. Obtención de órdenes y gestión de notificaciones de estado

```
@staticmethod
def notificar_estado(*args, **kwargs):···

@staticmethod
def __notificar_estado_post(controlador_estado_url, numero_orden, estado, extra="", timeout=5):

@staticmethod
def __notificar_estado_get(controlador_estado_url, numero_orden, estado, extra="", timeout=5):···

@staticmethod
def obtener_orden(controlador_orden_url, timeout=5):···

@staticmethod
def obtener_numero_orden(orden):···

@staticmethod
def obtener_posicion_coger(orden):···

@staticmethod
def obtener_posicion_dejar(orden):···
```

Figura 17: órdenes y gestión de notificación de estado
Fuente: elaboración propia

- Secuencias complejas de movimientos.

```
def secuencia_coger(self, numero_orden, controlador_estado, p_inicial, p_aproximacion, p_final,

def secuencia_dejar(self, numero_orden, controlador_estado, p_inicial, p_aproximacion, p_final,

def posicion_intermedia_coger_dejar(self, j_posicion, timeout_segundos=10):···
```

Figura 18: secuencias complejas de movimiento
Fuente: elaboración propia

2.2.6 Ejecución del programa.

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.

AIDIMME INSTITUTO TECNOLÓGICO GENERALITAT VALENCIANA IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN Financiado por la Unión Europea

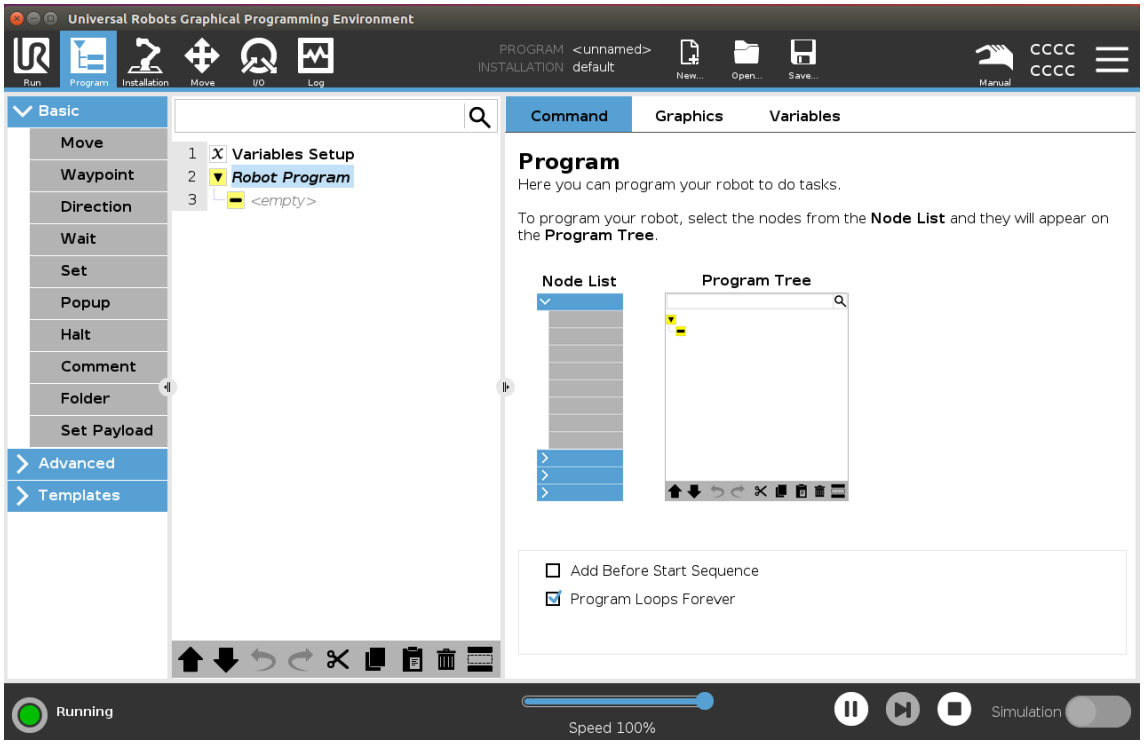

Figura 19: configuración inicial de consola del robot
Fuente: elaboración propia

Sin embargo, se puede observar el registro de incidencias del robot donde se muestra el inicio y finalización de cada uno de los programas ejecutados por el robot

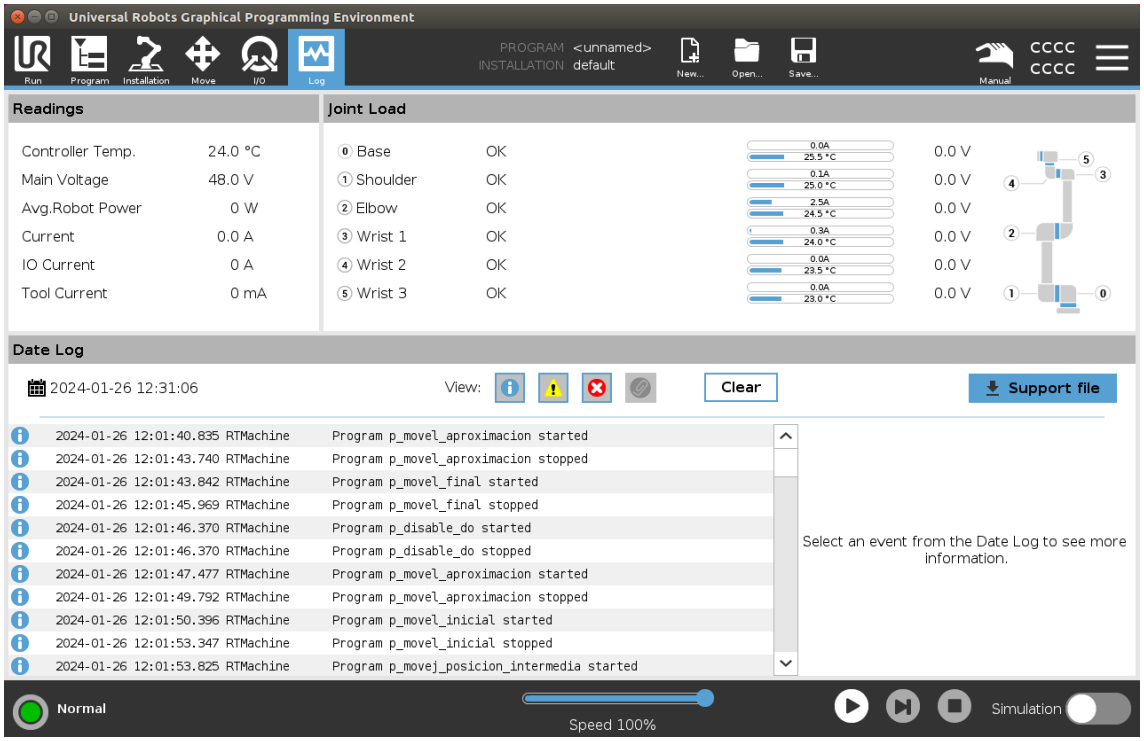

Figura 20: registro de incidencias
Fuente: elaboración propia

Durante la ejecución también se puede observar a través de la consola el cambio de estado de las señales digitales de la controladora gracias a las cuales se opera la herramienta del robot. En este caso la ejecución remota activa la señal digital de salida número 0.

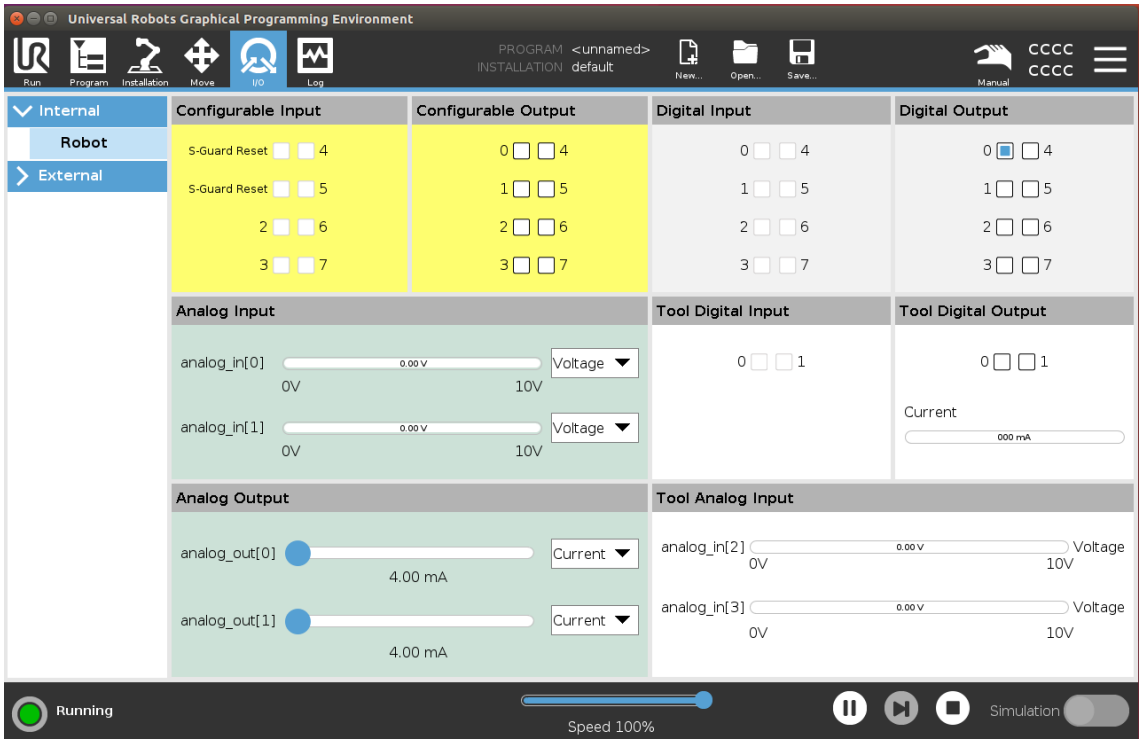

Figura 21: vista de señales
Fuente: elaboración propia

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 22: traza de ejecución del robot en el PC
Fuente: elaboración propia

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN
Financiado por la Unión Europea

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.

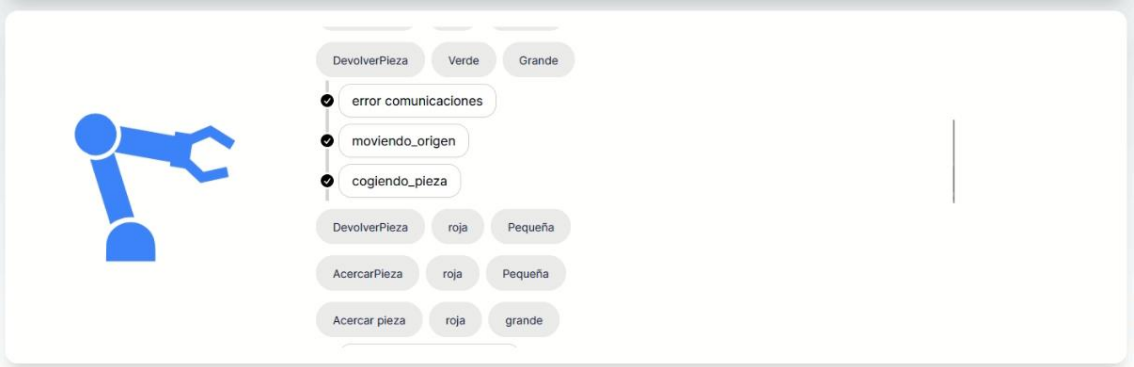

Figura 23: ejecución en la interfaz de usuario desarrollada
Fuente: elaboración propia

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN
Financiado por la Unión Europea

3 Resultados obtenidos

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.

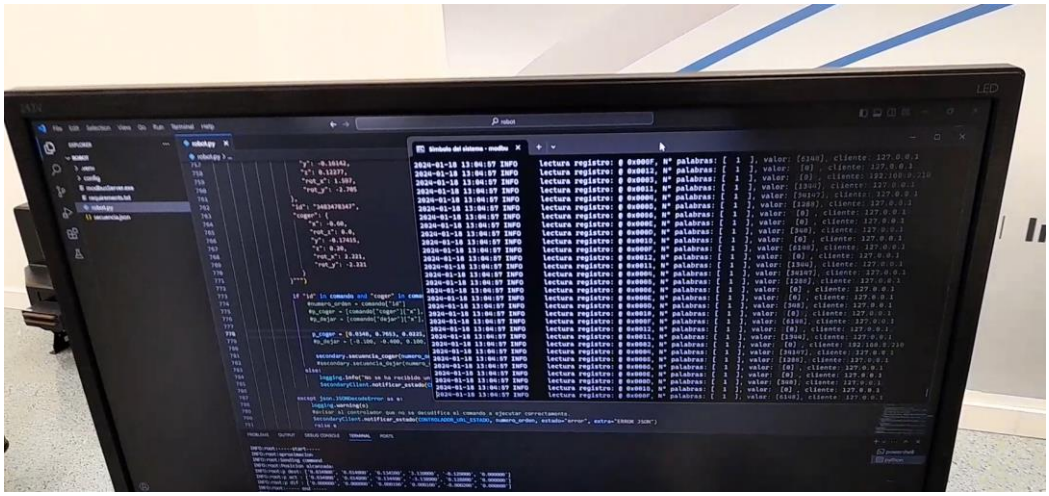

Figura 24: ejecución del programa de comunicaciones en pantalla
Fuente: elaboración propia

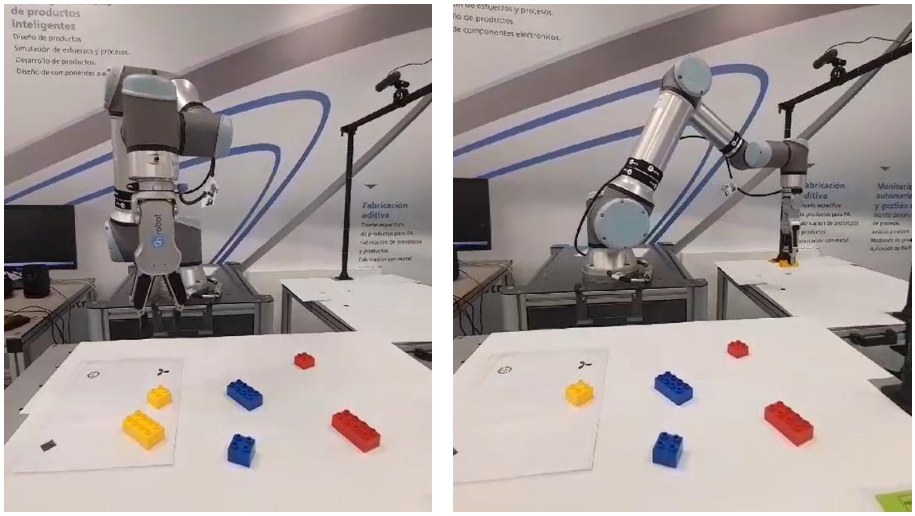

Figura 25: Robot recogiendo y entregando una pieza ubicada en una posición conocida
Fuente: elaboración propia

AIDIMME INSTITUTO TECNOLÓGICO GENERALITAT VALENCIANA IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN Financiado por la Unión Europea

A la hora de validar el sistema desarrollado se contó con la participación de las empresas colaboradoras en el proyecto.

En diferentes reuniones se les encuestó acerca de la idoneidad y adecuación de las instrucciones básicas definidas, obteniendo las siguientes valoraciones.

Cuestión	HURTADO R.	CFZ	DYMSA	Promedio
Las instrucciones elementales actuales permiten al robot realizar tareas complejas.	5	5	4	4,7
El número de instrucciones elementales es adecuado para cubrir las operaciones requeridas.	5	5	5	5,0
Otro caso de uso puede requerir del uso de nuevas instrucciones elementales	5	5	5	5,0

Figura 26: Tabla de cuestiones T6.1
Fuente: elaboración propia.

Las valoraciones se realizaron bajo una escala Likert entre 1 y 5, donde 1 implicaba estar totalmente en desacuerdo con la afirmación y 5 totalmente desacuerdo.

De forma generalizada las tres afirmaciones se consideraron pertinentes (ya que obtienen una media de puntuación mayor a 4).

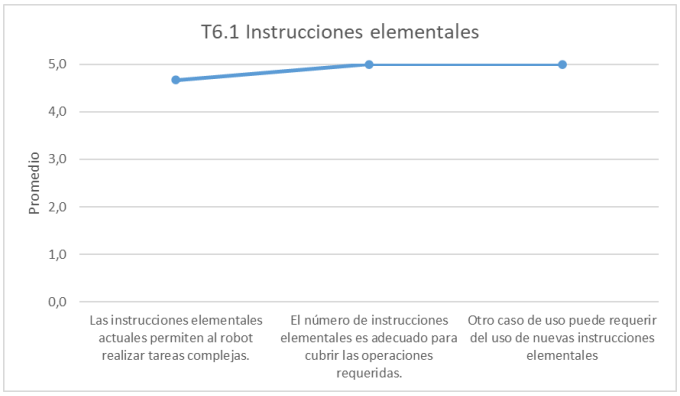

Figura 27: Grafica promedio T6.1
Fuente: elaboración propia.

Respecto del sistema de comunicaciones se encuestó acerca del nivel de errores en la comunicación y la ausencia de retrasos, tras mostrar a las empresas el funcionamiento del sistema.

Cuestión	HURTADO R.	CFZ	DYMSA	Promedio
El sistema de comunicación de instrucciones al robot funciona sin fallos o errores.	5	4	4	4,3
El sistema de comunicación de instrucciones no presenta retrasos significativos.	4	5	4	4,3

Figura 28: Tabla de cuestiones T6.2
Fuente: elaboración propia.

AIDIMME INSTITUTO TECNOLÓGICO · GENERALITAT VALENCIANA · IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN · Financiado por la Unión Europea

Las empresas participantes en el proyecto han considerado que el sistema consigue funcionar sin errores y con un tiempo de respuesta satisfactorio para los objetivos del proyecto.

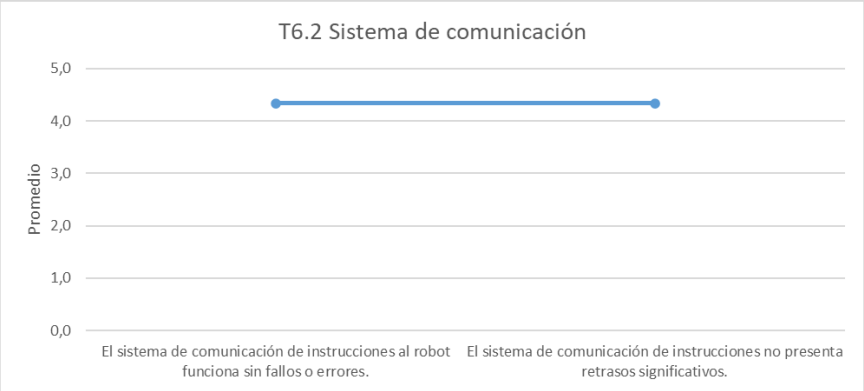

Figura 29: Grafica promedio T6.2
Fuente: elaboración propia.

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN
Financiado por la Unión Europea

4 Conclusiones

Como conclusiones al desarrollo de las actividades ejecutadas en el paquete de trabajo PT6 asociado al presente entregable podemos destacar las siguientes:

- Se ha definido un sistema de generación de instrucciones basado en movimientos básicos del robot a puntos predefinidos de las mesas de trabajo 1 y 2, se completa con movimientos a las posiciones específicas de la pieza que el usuario desea manipular.

- El control en remoto del robot UR16e se ejecuta por medio de la 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.

- Para conocer el estado del robot (posicionamiento y orientación cartesiana, posicionamiento y orientación angular, así como estado de ejecución de programa) se accede a diferentes registros Modbus.

- Se han realizado pruebas preliminares del sistema de comunicaciones para verificar que el robot ejecutaba correctamente la secuencia de tareas enviada desde el PC.

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN
Financiado por la Unión Europea

5 Anexos y bibliografía

No aplica.

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE+i INSTITUTO VALENCIANO DE COMPETITIVIDAD E INNOVACIÓN
Financiado por la Unión Europea

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
www.aidimme.es