

# **Laboratorio A: Programación de un robot industrial UR5**

Julio Salvador Lora Millán  
Juan Alejandro Castaño Peña

## Índice

1	Introducción y objetivos .....	3
2	Elementos del sistema.....	3
3	Primeros pasos con el robot UR .....	5
4	Configuración de herramienta.....	8
4.1	Configuración del TCP .....	8
4.2	Configuración del centro de gravedad y carga útil .....	9
5	Medidas de seguridad.....	10
6	Desarrollo de un programa .....	11
6.1	Instrucciones de movimiento .....	11
6.1.1	Definición de puntos de paso dinámicos.....	15
6.2	Manejo de Entradas y Salidas .....	16
6.2.1	Lectura de entradas digitales.....	17
6.2.2	Escritura de salidas digitales .....	18
6.3	Flujo del programa .....	19
6.3.1	Asignación de variables.....	19
6.3.2	Condicionales If/Else .....	21
6.3.3	Bucles .....	21
6.3.4	Subprogramas de robot .....	22
6.4	Coordenadas de la función.....	23
7	Ejercicio práctico de programación de un robot UR5 .....	25
7.1	Aplicación de dispensación de pegamento.....	25
7.2	Aplicación de pick and place .....	27
8	Referencias .....	29

## 1 Introducción y objetivos

Esta práctica pretende ser una primera aproximación al manejo de un robot industrial real. Concretamente, vamos a trabajar con el robot UR5 de Universal Robot.

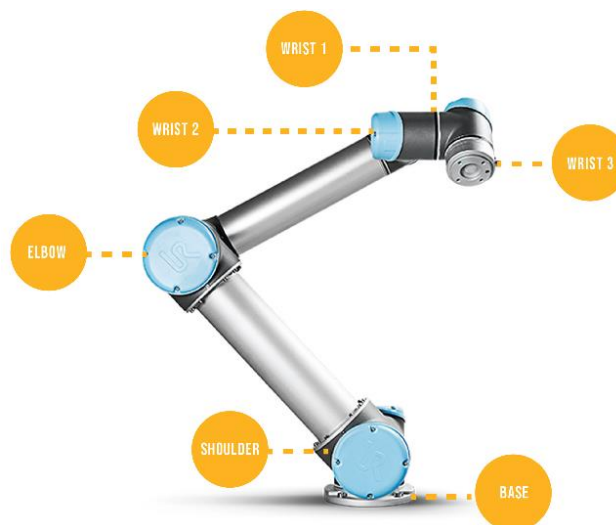


*Figura 1: Equipo UR5 con el que se va a trabajar durante esta práctica.*

Aunque en la asignatura, hasta ahora, sólo hemos trabajado con equipos de *ABB* a través de simulación en RobotStudio. Vamos a ver que los conceptos que ya hemos trabajado son extensibles también a los equipos de *Universal Robots*. De hecho, salvo por las diferencias debidas a la consola de programación, los mismos procedimientos son aplicables a cualquier otro robot industrial.

## 2 Elementos del sistema

El robot UR5 con el que vamos a trabajar está compuesto por 6 grados de libertad, capaces de rotar 360° sin limitación física:



*Figura 2: Grados de libertad del robot*

El controlador, por su parte, incluye la placa base, la tarjeta SD y el tablero de control de seguridad. El tablero de control de seguridad maneja todas las entradas y salidas desde el controlador y la conectividad con equipos periféricos como sensores, actuadores o elementos de seguridad. Se cuenta con entradas y salidas digitales de carácter general o dedicadas para equipos de seguridad, así como señales analógicas. Además, cuenta con un micro-ordenador, conectividad USB y Ethernet y la tarjeta SD donde está almacenado el Software del robot.



*Figura 3: Unidad de control*

La consola de programación (denominada en este entorno *TechPendant*), permite encender el robot, programarlo para que realice tareas específicas y manejar las entradas y salidas del controlador.



*Figura 4: TechPendant para la programación de robots UR*

A pesar de que los equipos UR son robots colaborativos (*Cobots*), seguros por su propia naturaleza, el *TechPendant* incluye una seta de emergencia que detiene de manera inmediata la ejecución de la rutina del robot por motivos de seguridad.

### 3 Primeros pasos con el robot UR

El arranque del robot se realiza a través del pulsador correspondiente en la parte superior del Tech Pendant. Una vez encendida la consola, hay que hacer *click* en el botón rojo de la esquina inferior izquierda para acceder a la pantalla de inicialización de la unidad mecánica. Al hacer *click* en “Encender” transmitiremos energía a las articulaciones, y por último con “Iniciar” liberaremos los frenos mecánicos del motor. Haciendo *click* en “Salir” volveríamos a la pantalla de inicio. Observa como el piloto de la esquina inferior izquierda ha ido cambiando de color para indicar el estado de la unidad mecánica.

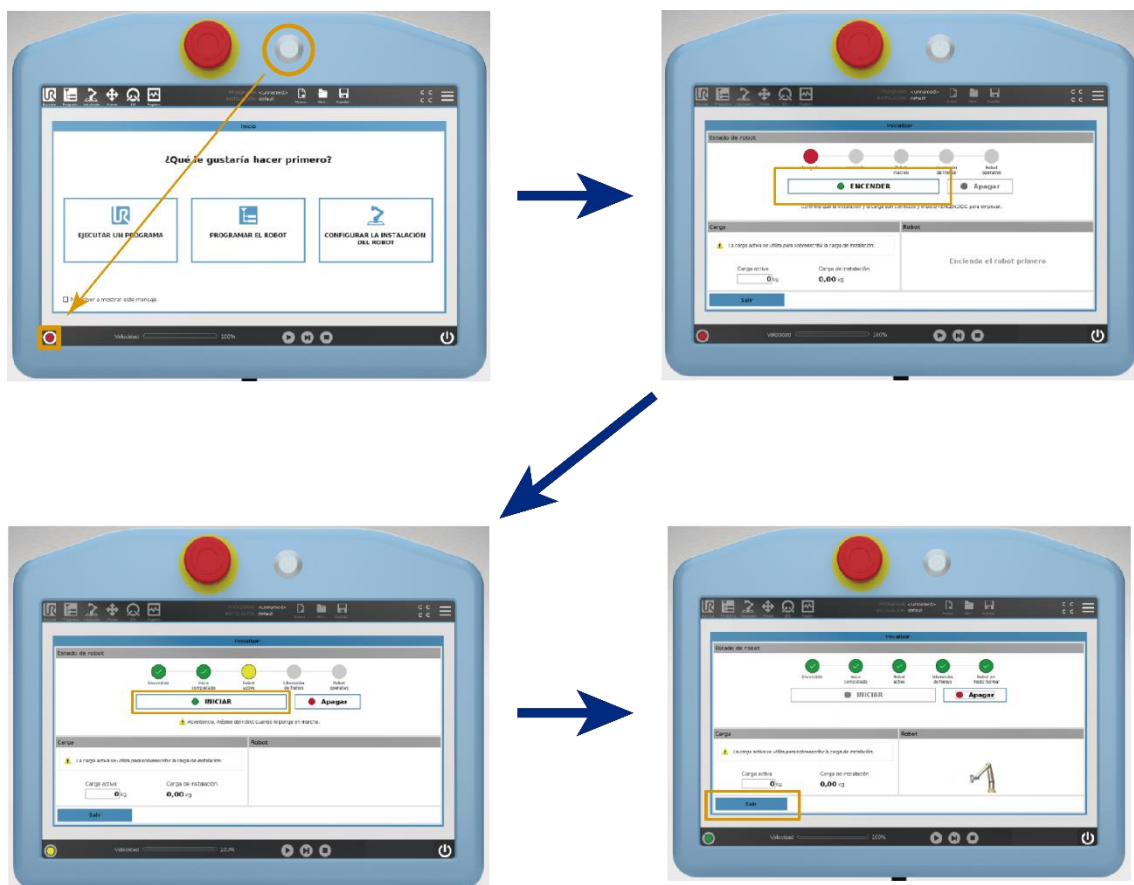
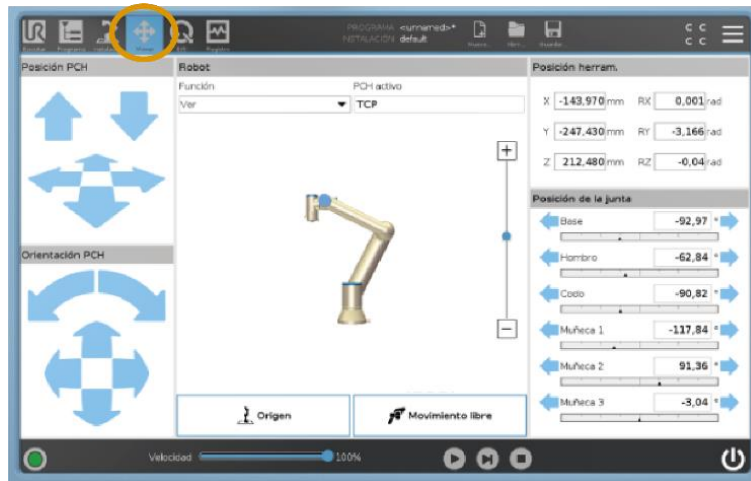


Figura 5: Puesta en marcha del robot

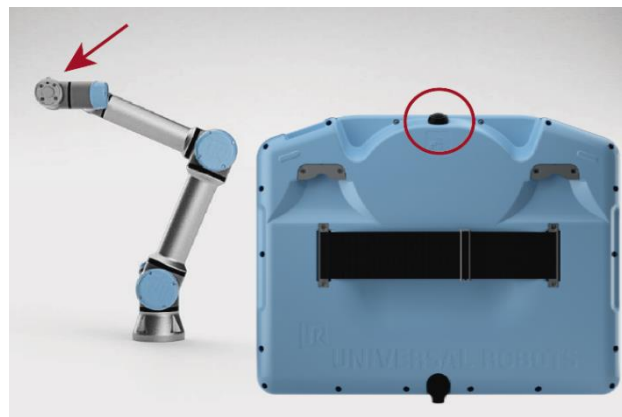
Desde la pestaña de movimiento de la consola de programación, se puede comandar directamente el movimiento de la unidad mecánica. Este movimiento puede hacerse

articulación a articulación del robot, utilizando los selectores del menú derecho, o bien actuando sobre la posición y la orientación del TCP activo (flechas del menú izquierdo).



*Figura 6: Movimiento del robot desde el TechPendant*

Puesto que se trata de un *Cobot*, otra opción para mover el robot es hacerlo mediante el movimiento manual. Para ello, se ha de pulsar el botón de la parte trasera del TechPendant mientras se mueve manualmente la unidad mecánica hasta la posición deseada:



*Figura 7: Guiado manual del UR*

De igual manera, una característica del *Cobot* es que es capaz de trabajar en el mismo entorno que los operarios. Por este motivo, incluye medidas de seguridad que detienen el robot si encuentra demasiada resistencia al movimiento. Cuando esto ocurre, el robot entra en parada de protección, deteniendo la ejecución de lo que estuviera ocurriendo. Para salir del modo de parada de protección, basta con volver a habilitar el funcionamiento del UR, una vez que el origen de la parada ha desaparecido.

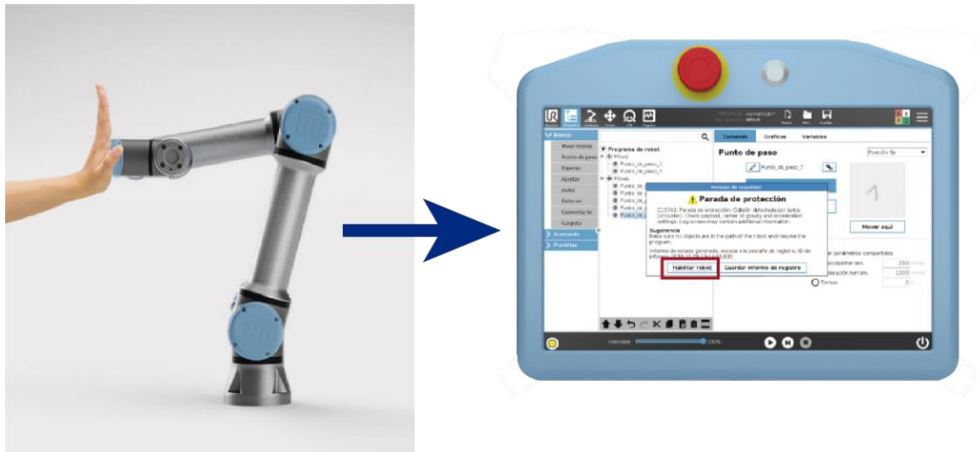


Figura 8: Parada de protección

Por último, es preciso comentar que en cualquier momento del funcionamiento del robot se tienen disponibles los botones del encabezado y pie de página. En el encabezado se encuentran los botones para crear un nuevo programa, abrir uno ya creado previamente o guardar el programa actual para recuperarlo más adelante.



Figura 9: Opciones para guardar y abrir programas del robot

Por su parte, los botones del pie de página permiten controlar la ejecución del programa del robot:

- El selector de velocidad permite disminuir proporcionalmente la velocidad del movimiento del robot.
- El botón de simulación permite simular el movimiento del robot en lugar de ejecutarlo en la unidad mecánica. De esta manera se asegura que el robot no realizará ninguna colisión ni se dañará a él mismo. Este modo debe usarse para probar programas cuando no se está seguro de su ejecución.
- Los botones de control de ejecución permiten ejecutar el programa cargado, una única instrucción o detener la ejecución del programa.

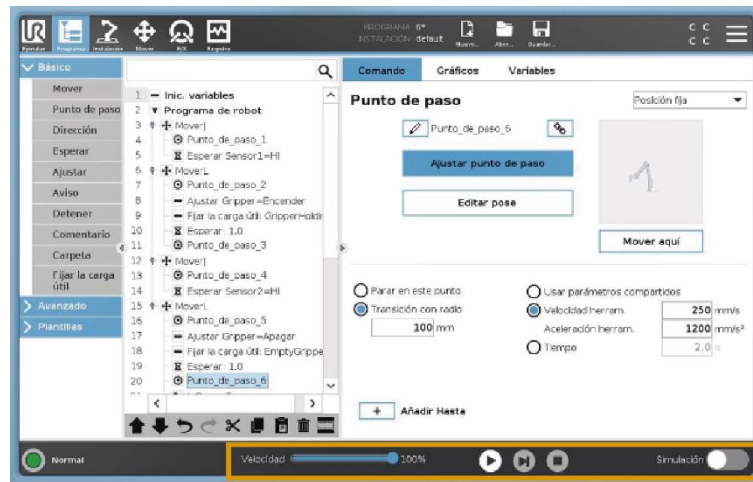


Figura 10: Botones de pie de página

## 4 Configuración de herramienta

Para que el robot funcione de manera correcta, es necesario definir los parámetros que caracterizan el efector final. Concretamente, tenemos que definir:

- El TCP o PCH (*Tool Center Point*, Punto Central de Herramienta).
- La posición del centro de gravedad del efector final
- La carga útil, es decir, el peso del end-effector.

### 4.1 Configuración del TCP

Conocidas las distancias desde el centro de la brida al TCP, es posible configurar directamente la posición del TCP, introduciéndolas en el menú de configuración del TCP (Instalación → PCH → Posición).

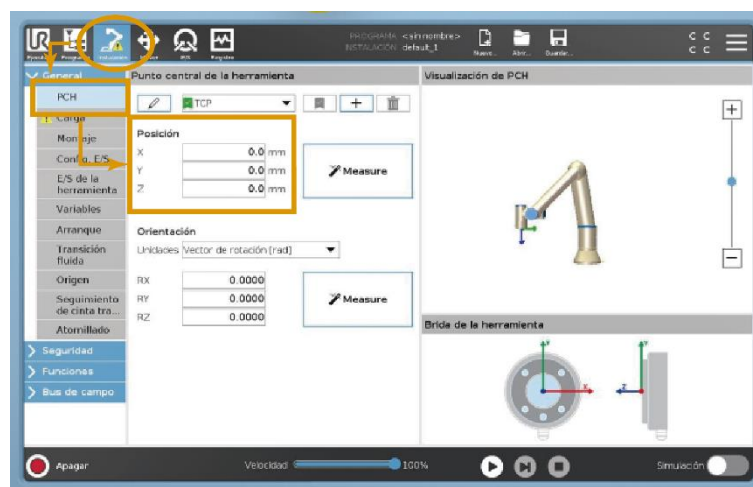


Figura 11: Configuración directa del TCP



Si no es posible medir directamente la posición del TCP con respecto al centro de la brida, el sistema es capaz de calcularlo mediante un proceso iterativo. Para ello, hay que llevar la herramienta al mismo punto con 4 orientaciones diferentes (método de los 4 puntos).

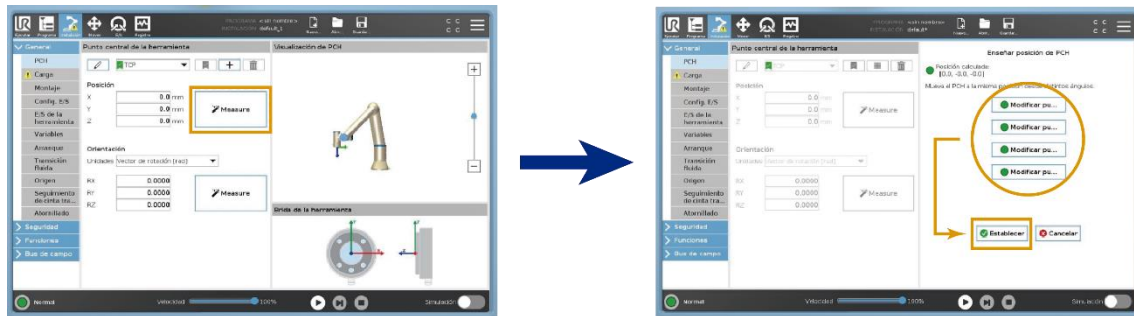


Figura 12: Configuración del TCP mediante el método de los 4 puntos

El sistema también es capaz de calcular de manera automática la orientación de la herramienta en caso de que su eje Z (la dirección de aproximación) no sea paralelo al eje Z de la brida. Para ello hay que llevar el robot a una posición tal que la dirección de la herramienta sea paralela al eje Z de la base (dirección vertical si el robot está sobre una superficie horizontal).

## 4.2 Configuración del centro de gravedad y carga útil

El software es capaz de estimar ambos de manera autónoma, mediante un asistente (Instalación → PCH → Carga). Al igual que el caso del cálculo del TCP hay que llevar al robot a 4 posiciones diferentes. En esta ocasión es importante que no se toque la brida ni la herramienta durante el movimiento, para evitar errores en la estimación de la masa del efector terminal. Por este motivo el robot deberá moverse desde la consola o tocando la muñeca del robot.

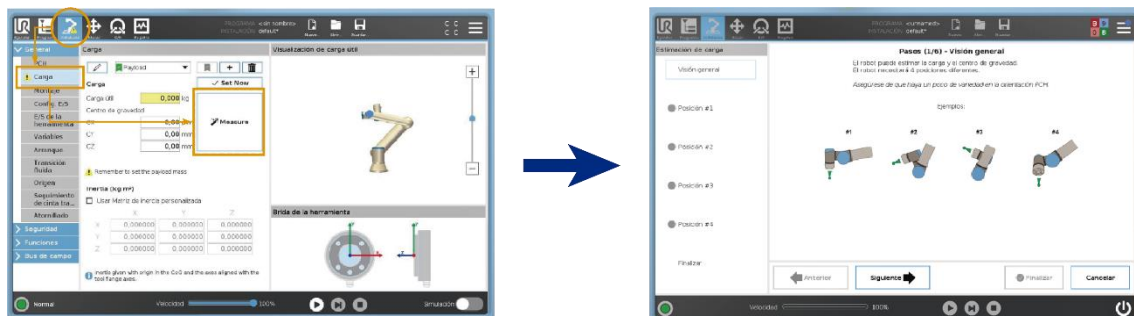
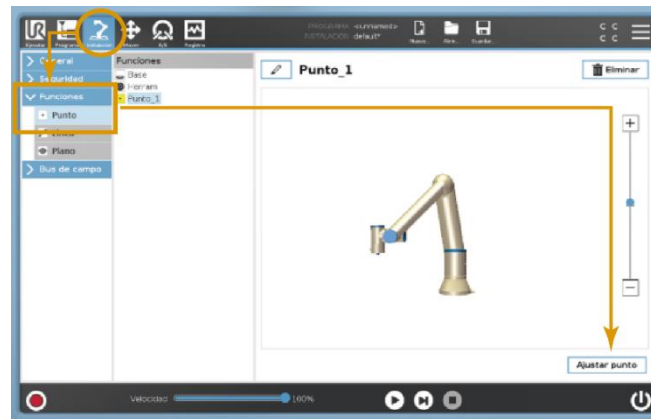


Figura 13: Asistente para el cálculo del centro de gravedad y la carga útil

## 5 Medidas de seguridad

Un aspecto muy importante de la robótica industrial son las medidas de seguridad, especialmente cuando se trata de robots colaborativos que trabajarán en el mismo entorno que los operarios. La medida de seguridad más básica que se puede imponer a estos robots es la creación de áreas en las que no se permite que funcione el robot, o en las que sólo se le permite el funcionamiento en modo reducido. Las restricciones se aplican tanto para el TCP como para el codo del sistema mecánico.

Estas áreas se pueden definir utilizando la posición del TCP. Almacenando el punto en el que se encuentra la herramienta, se define un plano que pasa por dicho punto y es perpendicular a la dirección de aproximación del efector final.



*Figura 14: Definición de un punto para crear un plano de seguridad.*

Una vez definido el punto, se puede definir directamente el plano de seguridad para nuestra aplicación. Para ello hay que acceder a la sección “Planos” dentro del menú “Seguridad”. En primer lugar, es preciso añadir el plano, en el menú superior de la derecha, a continuación, especificamos el punto que vamos a utilizar para definir el plano, y por último seleccionamos el comportamiento de la restricción que vamos a crear. Existen cuatro tipos de restricciones:

1. Normal: no se permite traspasar el plano de seguridad cuando se trabaja en modo normal.
2. Reducido: no se permite traspasar el plano cuando se trabaja en modo reducido
3. Ambos: no se permite traspasar en ninguno de los dos modos de funcionamiento.
4. Modo reducido con activador: hace que robot pase a funcionar en modo reducido cuando atravesase el límite.

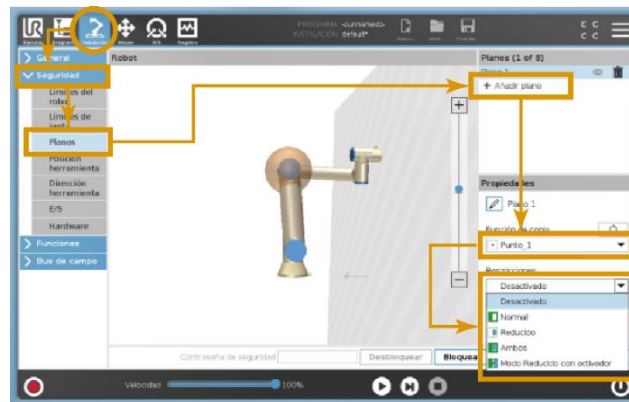


Figura 15: Creación de un plano de seguridad

La configuración de estas medidas de seguridad requiere de una Contraseña de Seguridad. En nuestra instalación, dicha contraseña es “admin”.

## 6 Desarrollo de un programa

La programación se realiza desde la pestaña de programa. La interfaz de usuario para programar el robot incluye:

- El árbol de programas donde aparecen los comandos que ejecutará el robot.
- Un listado con las instrucciones más habituales para el robot, como movimientos, puntos de paso, esperas o acciones sobre entradas y salidas.

### 6.1 Instrucciones de movimiento

Para que el robot se mueva a una primera posición, tenemos que incluir la instrucción de movimiento “Move” del menú “Básico”. A continuación, tendremos que definir el “punto de paso” hasta el que se va a desplazar el robot, para ello basta con hacer *click* sobre él, y después sobre ajustar punto de paso.

Dicho punto de paso será la posición actual de la unidad mecánica. Por ello, antes de guardar el punto, el *TechPendant* nos permite modificar la posición del brazo mediante la consola de programación. Haciendo *click* en OK, guardaremos la posición como punto de paso.

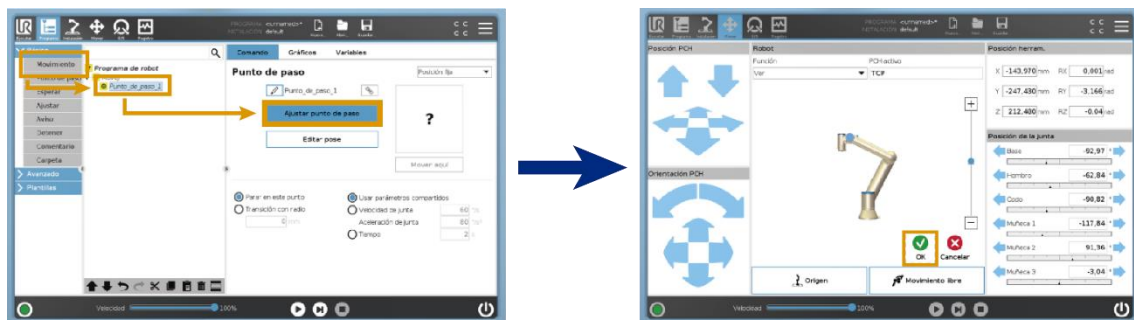
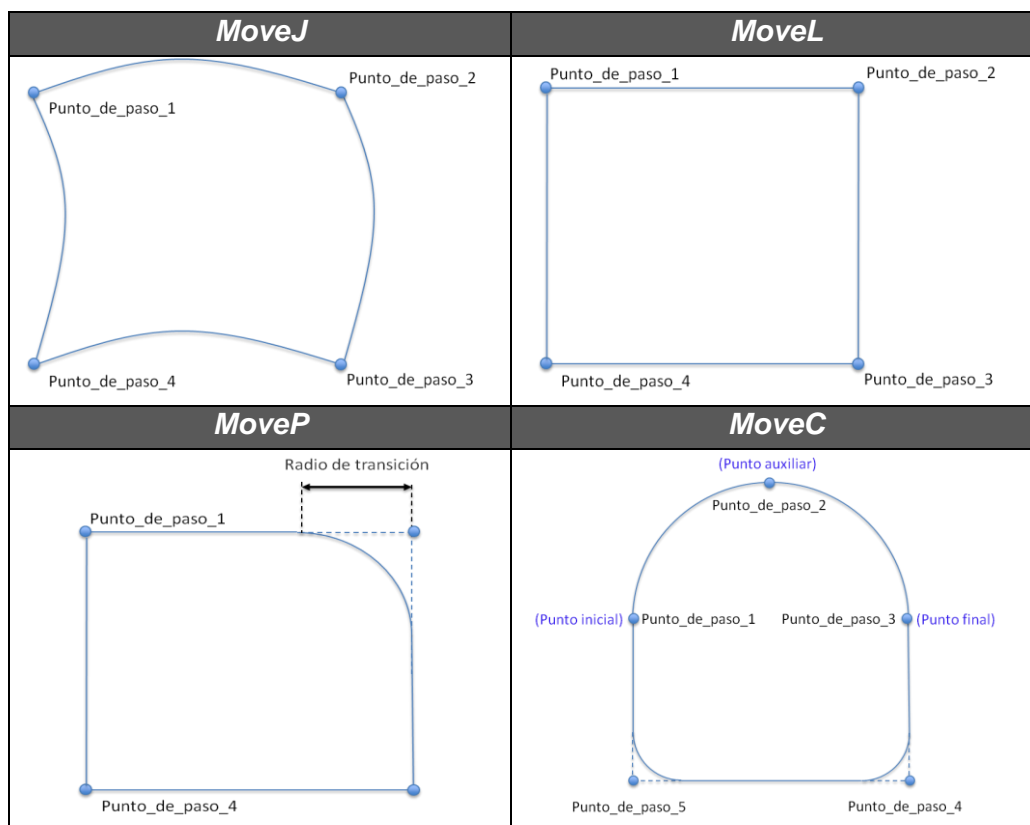


Figura 16: Instrucción de movimiento y definición de un punto de paso

Los UR definen cuatro tipos de movimientos, cada uno de ellos con características propias:

- MoveJ: es el movimiento más rápido ya que no se fuerza a las articulaciones (Joints) a seguir ninguna restricción.
- MoveL: el TCP describe una trayectoria rectilínea desde el punto de origen al destino, el robot se detiene momentáneamente cada vez que alcanza el punto de paso..
- MoveP: o movimiento de proceso, se trata también de movimientos lineales entre los puntos, pero en este caso a velocidad constante. Puesto que no se puede acelerar y decelerar el movimiento, no es posible realizar giros bruscos. Por este motivo se tiene que definir el radio de transición para unir dos segmentos rectilíneos.
- MoveC: el TCP describe una trayectoria circular desde el punto de origen hasta el destino pasando por un punto auxiliar. Se trata de un sub-tipo del MoveP.



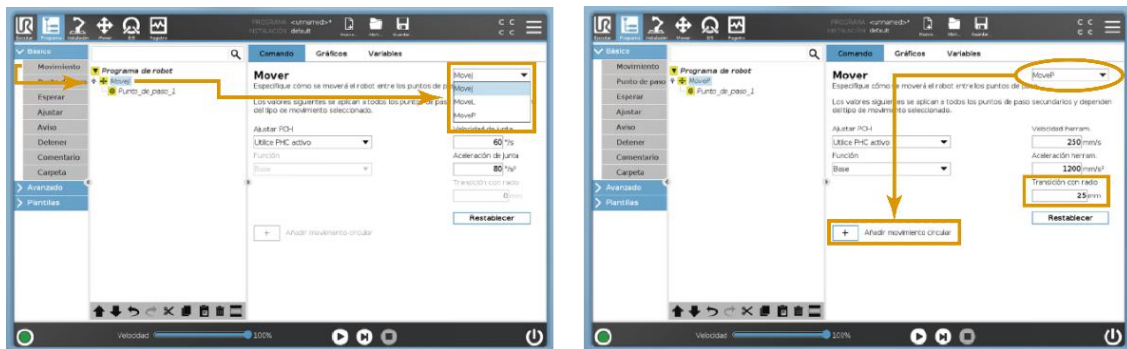


Figura 17: Configuración de tipos de movimiento

A la hora de utilizar movimientos circulares, es posible configurar el cálculo de la orientación de la herramienta a lo largo de la trayectoria. Este cálculo de la orientación puede ser:

- Fijo: sólo se utiliza el punto de inicio para definir la orientación de la herramienta, manteniéndose la orientación relativa al círculo del punto inicial durante todo el movimiento y sin tener en cuenta la orientación del punto intermedio y final.
- Ilimitado: la orientación del punto inicial se transforma en la orientación definida para el punto final durante el transcurso de la trayectoria, de manera que cada punto intermedio de la trayectoria tiene una orientación relativa al círculo diferente.

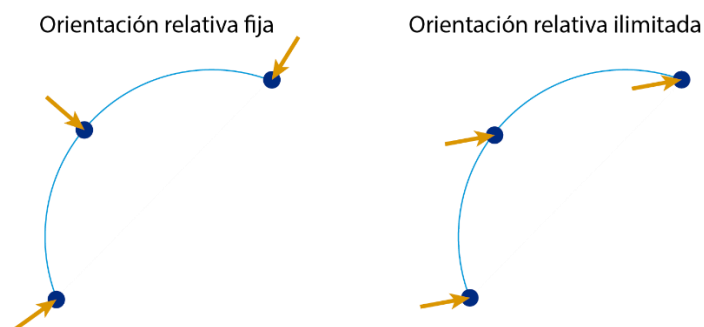


Figura 18: Métodos de cálculo de la orientación en trayectoria circular

Para incluir movimientos nuevos, es importante hacer *click* sobre la última instrucción de movimiento, así evitaremos anidar movimientos que dificulten la lectura y depuración del programa.

Para concatenar movimientos del mismo tipo, no es necesario introducir instrucciones nuevas, basta con incluir puntos de paso sucesivos. Para ello, basta con hacer *click* en la opción "Punto de paso" dentro del menú "Básico".

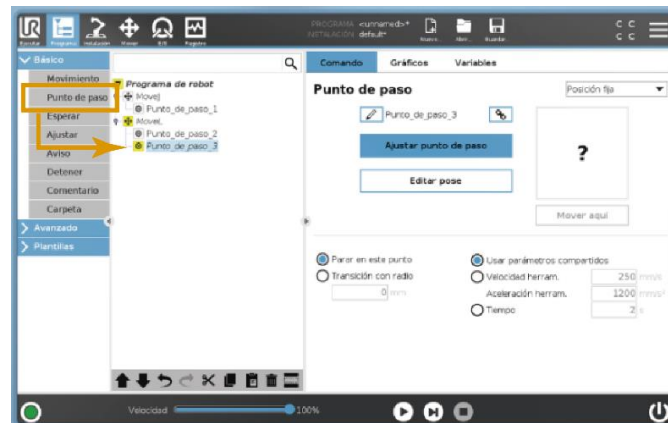


Figura 19: Introducir punto de paso

Una vez definidos los puntos de paso, se pueden configurar ciertos parámetros adicionales para mejorar el rendimiento del programa:

- Transición con radio: permite no detenerse en un punto de paso concreto, sino que realiza la transición a la siguiente trayectoria mediante un arco de radio configurable:

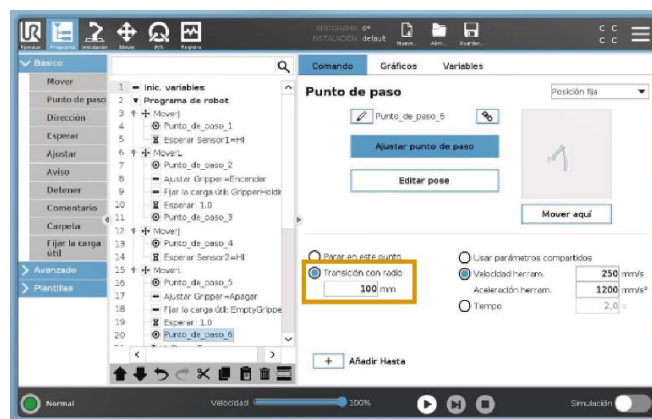


Figura 20: Definición de radio de transición

- Velocidad del movimiento: se puede especificar para todos los puntos de paso dentro de un movimiento o bien para un único punto de paso.

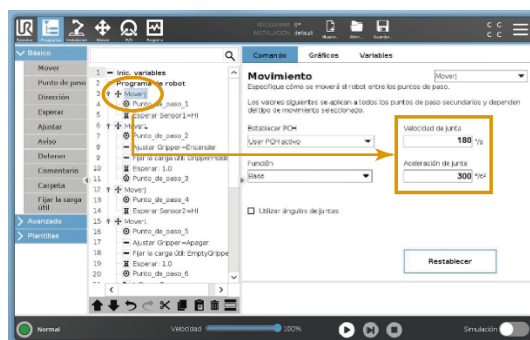


Figura 21: Definición de velocidad para un movimiento

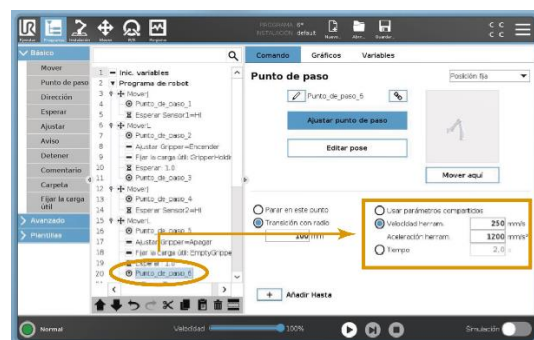


Figura 22: Definición de velocidad para un punto de paso

### 6.1.1 Definición de puntos de paso dinámicos

Hasta ahora, los puntos de paso se han definido fijos, es decir, se encuentran expresados en coordenadas cartesianas absolutas con respecto a la base del robot. Esta configuración es la que aparece por defecto, tal y como se muestra señalado en la siguiente figura:

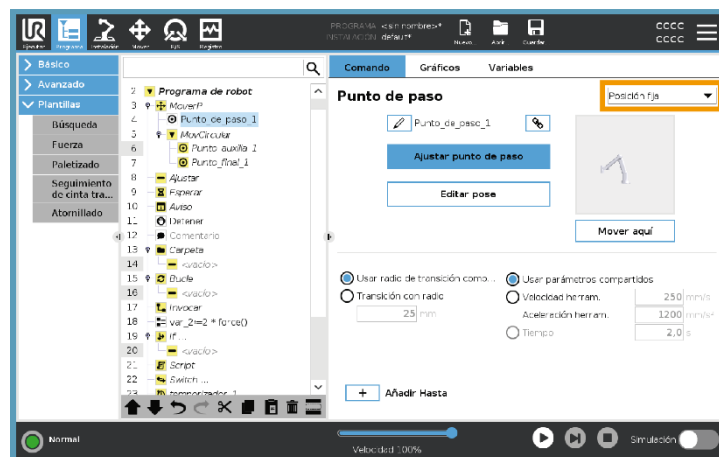


Figura 23: Punto de paso definido como posición fija.

Sin embargo, es posible definir los puntos de paso también de forma dinámica, de manera que sean el resultado de un cálculo. En este sentido, se disponen de dos opciones: puntos de paso relativos y puntos de paso variables.

- Los puntos de paso relativos se configuran como un movimiento desde la posición anterior, como por ejemplo “dos centímetros a la izquierda”. Este movimiento relativo se define como la diferencia entre los puntos almacenados en los campos “Desde” y “Hasta” del menú. *Nota: posiciones relativas repetidas pueden sacar a la unidad mecánica de su espacio de trabajo.*

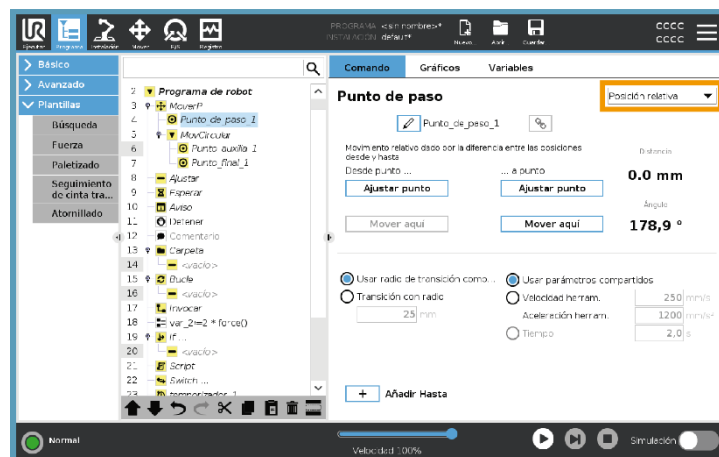


Figura 24: Punto de paso relativo.



- Los puntos de paso variables se definen como el contenido de una variable del programa según el formato *pose*, como por ejemplo:

$$ptoPaso=p[0.5, 0.0, 0.0, 3.14, 0.0, 0.0]$$

Donde las tres primeras componentes del vector son las posiciones  $x$ ,  $y$ ,  $z$  y las tres últimas son la orientación dada como un vector de rotación  $rx$ ,  $ry$ ,  $rz$ . La longitud del eje es el ángulo que se debe rotar en radianes, y el vector en sí proporciona el eje sobre el que rotar. El sistema de referencia utilizado para la definición viene marcado por la función empleada en la instrucción de movimiento (consultar apartado 6.4 - *Coordenadas de la función* para más información).

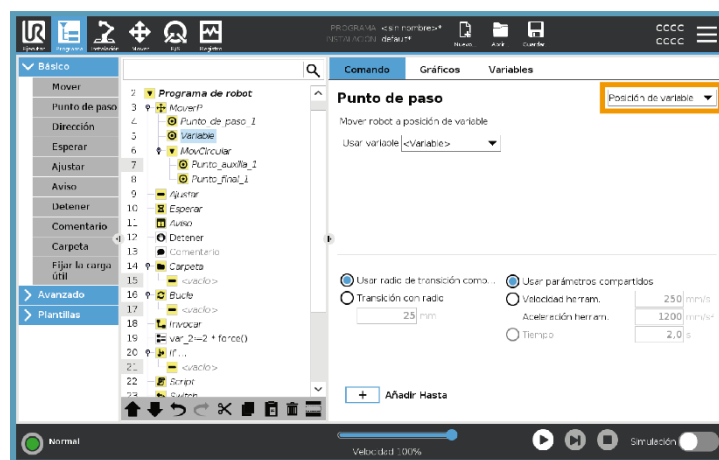


Figura 25: Punto de paso variable.

Estas variables pueden ser el resultado de operaciones aritméticas llevadas a cabo a través de instrucciones de asignación de variables (consultar apartado 6.3.1 - *Asignación de variables* para más información).

## 6.2 Manejo de Entradas y Salidas

El programa del robot debe ser capaz de interactuar con elementos externos a través del módulo de entradas y salidas digitales del controlador.

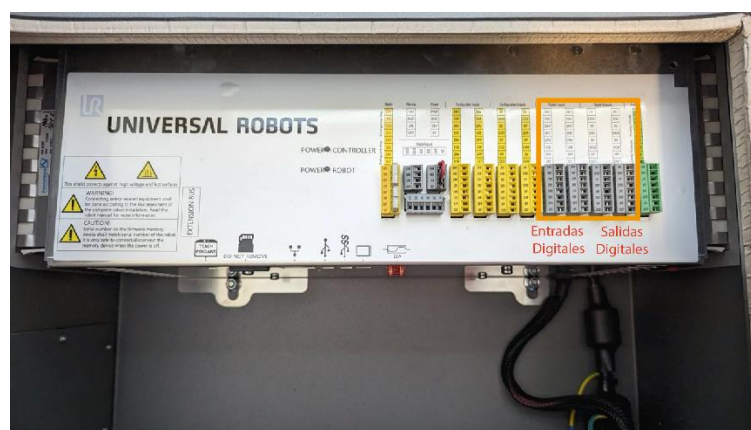


Figura 26: Módulo de E/S digitales.



Dentro del menú instalación, en la opción Config. E/S, se puede renombrar las entradas y salidas digitales de la instalación. De esta manera, es posible darle nombres coherentes con los sensores y actuadores de nuestro sistema.

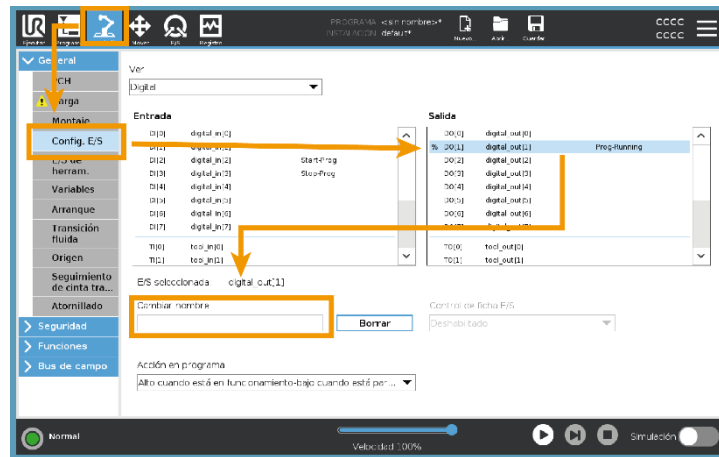


Figura 27: Configuración de E/S digitales.

En el menú E/S de la cinta superior se tiene acceso directo al módulo de entradas y salidas, pudiendo ver directamente los niveles lógicos de las entradas y accionar directamente los puertos de salida.

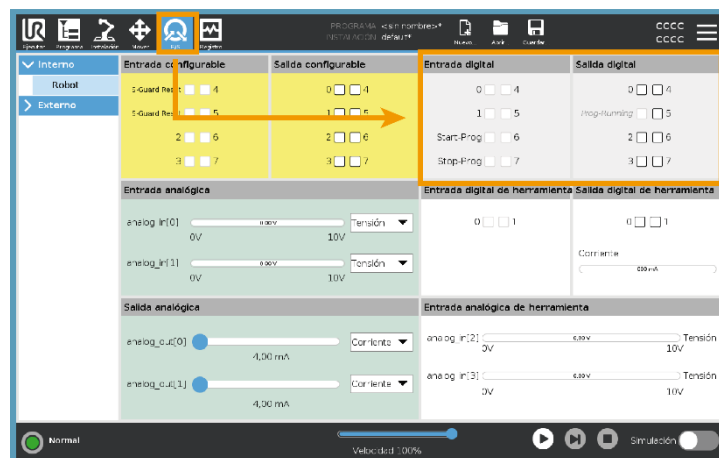


Figura 28: Manejo directo de E/S digitales.

### 6.2.1 Lectura de entradas digitales

Las entradas digitales se utilizan para captar información del entorno del sistema. Normalmente, se utilizan para detener el flujo del programa hasta que se detecta un evento determinado a través de un sensor. Este comportamiento se programa mediante el comando “Esperar”.

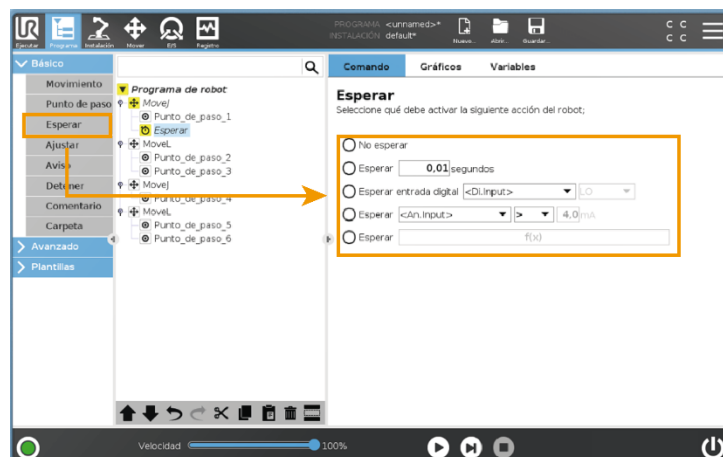


Figura 29: Instrucción Esperar para lectura de entradas digitales

El comando “Esperar” detiene el funcionamiento normal del sistema hasta que se cumple una condición determinada. Esta condición es configurada por el usuario y puede ser:

- Una cantidad de tiempo, especificada en segundos, para garantizar que se ha completado una tarea externa al controlador del robot.
- Un valor lógico alto (HI) o bajo (LO) en una entrada digital.
- Un nivel de corriente determinado, especificado en mA en una entrada analógica.

### 6.2.2 Escritura de salidas digitales

Las salidas digitales se utilizan para forzar una acción en el entorno del sistema, por ejemplo, actuar sobre la herramienta o comandar elementos auxiliares. Para ello, es necesario fijar el valor eléctrico en el módulo de salidas digitales. Esto se hace a través de la instrucción “Ajustar”.

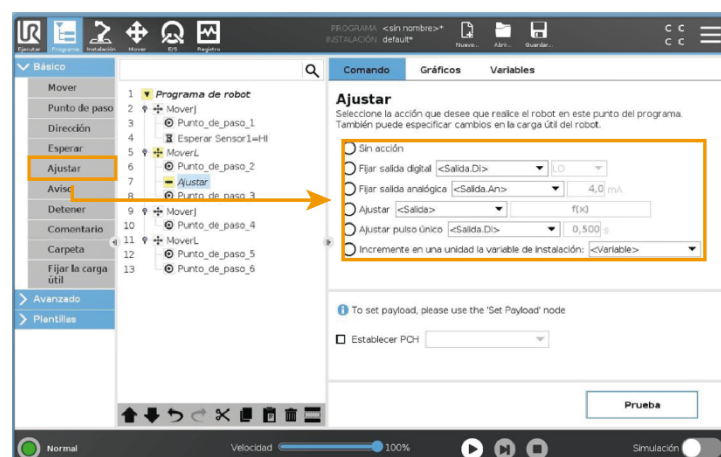


Figura 30: Instrucción Ajustar para escritura de entradas digitales

El comando “Ajustar” permite establecer el valor de un elemento, concretamente se puede trabajar con:

- Salidas digitales, configurando su nivel lógico a alto (HI) o bajo (LO) o mediante un pulso de duración determinada
- Salidas analógicas, estableciendo el nivel de corriente
- Incrementando (+1) el contenido de una variable.

### 6.3 Flujo del programa

Para dar mayor flexibilidad a la rutina de funcionamiento del robot, es posible introducir sentencias de control de flujo.

### 6.3.1 Asignación de variables

Es posible utilizar variables para almacenar información que condicionen el comportamiento del programa. Estas variables pueden tomar valores fijos o siguiendo una expresión aritmética en base a alguna variable. Para ello, basta con incluir una instrucción de “Asignación”, disponible en el menú “Avanzado”, crear una nueva variable e introducir la expresión correspondiente en el campo disponible.

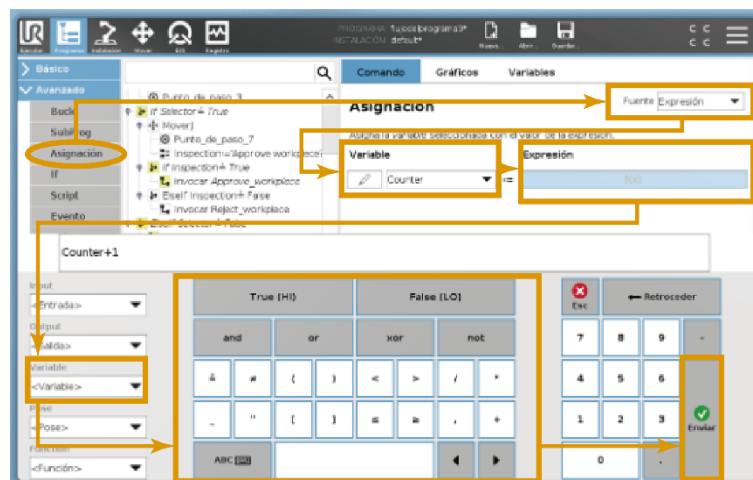


Figura 31: Asignación de variable por expresión

Hay ocasiones en los que la rutina de funcionamiento del robot ha de pedir información al operario para ajustar su comportamiento. Esto se hace a través de asignación de valores a variables a través de la consola de programación del robot.

Para ello, se ha de incluir una “Asignación” dentro del árbol de programa en el punto en el que se ha de pedir esta información al operario. A continuación, basta con darle nombre y seleccionar como fuente de la información al “Operador” en el menú de la esquina superior derecha de la ventana “Comando”. Los tipos de asignaciones pueden ser:

- Sí o No, es decir, directamente valores *True* o *False*.
- Un número entero.
- Un número decimal.
- Una cadena de texto.

Además, es posible solicitar esta información al operador, configurando el mensaje que se muestra al operador para ello.

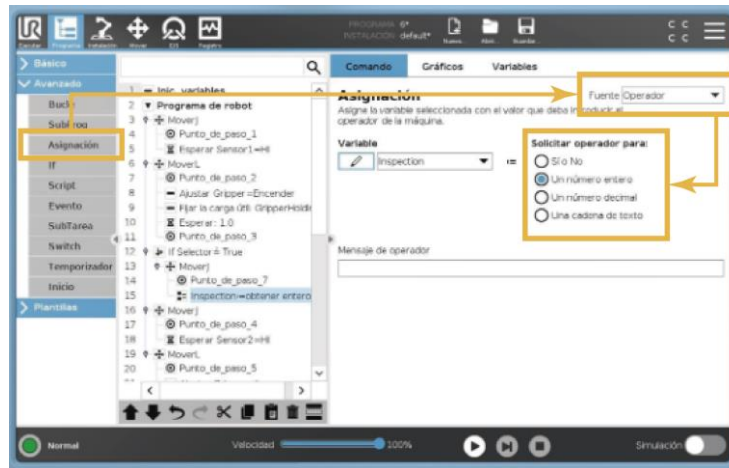


Figura 32: Asignación de valores a variables por parte del operador

Las asignaciones también permiten definir variables que funcionen como puntos de paso siempre que se mantenga el formato *pose* (ver sección 6.1.1). Además, es posible utilizar funciones para modificar puntos de paso (*poses*) ya definidos. Para ello, en el menú asignación se da la posibilidad de usar como datos de partida poses ya almacenadas, así como utilizar funciones para manipular dichas poses.



Figura 33: Uso de poses y funciones en una asignación de variables.

Dentro de las funciones, podemos encontrar algunas como:

- *pose\_add(<pose1>, <pose2>)*: permite sumar poses para generar un movimiento relativo
- *pose\_sub(<pose\_to>, <pose\_from>)*: permite calcular la diferencia entre poses para obtener la transformación intermedia

### 6.3.2 Condicionales If/Else

Esta estructura permite la ejecución condicional de acciones, siguiendo el esquema If-Else clásico de todos los lenguajes de programación. Haciendo *click* sobre la opción “If” del menú avanzado, se inserta la estructura. La condición que se ha de cumplir para que se ejecuten las instrucciones correspondientes se ha de especificar en el editor de expresiones de la ventana comando. Esta condición puede estar vinculada a señales de entrada o salida, o a valores de variables. Por último, las correspondientes sentencias “Else” o “Elseif” se incluirían mediante los botones de la parte inferior del menú “Comando”.

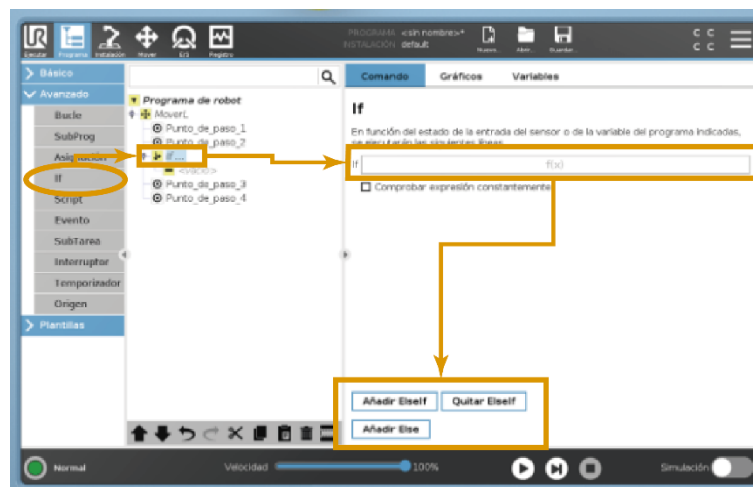


Figura 34: Sentencias de ejecución condicional If/Else

### 6.3.3 Bucles

Existe la posibilidad de utilizar bucles tipo “For” o “While”. Todas estas estructuras de programación aparecen bajo el epígrafe “Bucle” del menú “Avanzado”.

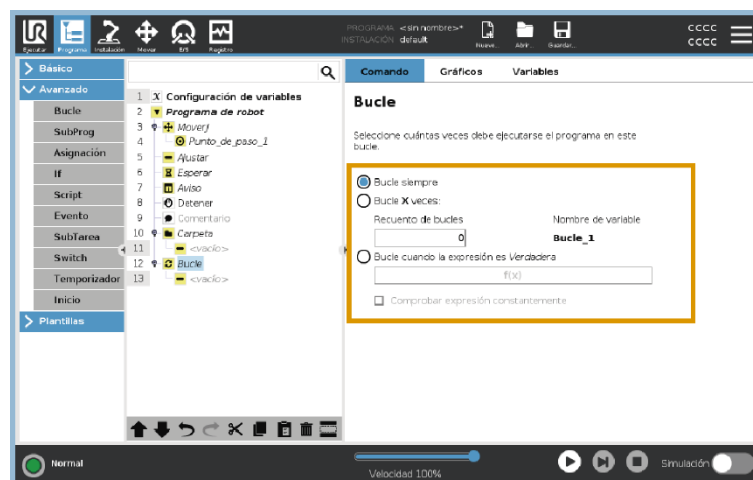


Figura 35: Bucles

Los bucles tipo “For” utilizan una variable para ejecutar el código interno un número determinado de veces (X). Esta variable se actualiza de manera automática tomando

valores de 0 a X-1. Por su parte, los bucles “While” repiten la ejecución del código en su interior mientras se cumpla una condición.

#### 6.3.4 Subprogramas de robot

Se trata de conjuntos de instrucciones que se tratan como funciones, es decir, se definen fuera del flujo normal de instrucciones del robot, y pueden ser llamadas desde cualquier punto del programa. Son especialmente útiles para estructurar el código y facilitar la posterior depuración.

Para definir un nuevo subprograma, primero hay que hacer *click* en el “Programa de robot” del árbol de programa y, estando este activo, hacer *click* en “SubProg”

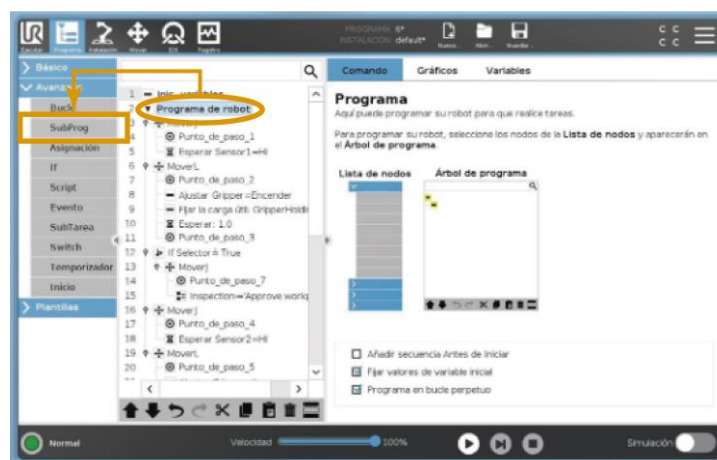


Figura 36: Definición de subprogramas.

Una vez definido el subprograma y renombrado de manera acorde, se puede invocar al mismo en cualquier punto del programa del robot. Basta con hacer *click* en la posición correcta el programa principal y a continuación hacer *click*, de nuevo, en “SubProg”. Una vez incluida la instrucción para la invocación, se ha de seleccionar la rutina correspondiente dentro del menú desplegable de la ventana “Comando”.

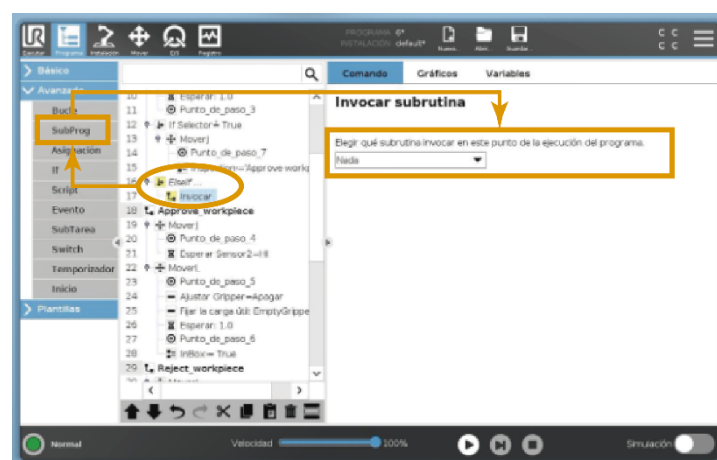


Figura 37: Invocación de subprograma.

En este sentido, lo único que quedaría es incluir las instrucciones pertinentes dentro del programa. Esto se hace de manera idéntica a cuando se trabaja dentro del árbol de programa, incluso puede hacerse uso de herramientas de “cortar”, “copiar” y “pegar” que están disponibles en la cinta inferior de herramientas.

#### 6.4 Coordenadas de la función

En el entorno de Universal Robots, se denomina “función” a los sistemas de referencia utilizados para la definición de puntos de pasos. Los UR consideran por defecto únicamente dos funciones, una asociada a la base y otra asociada al TCP.

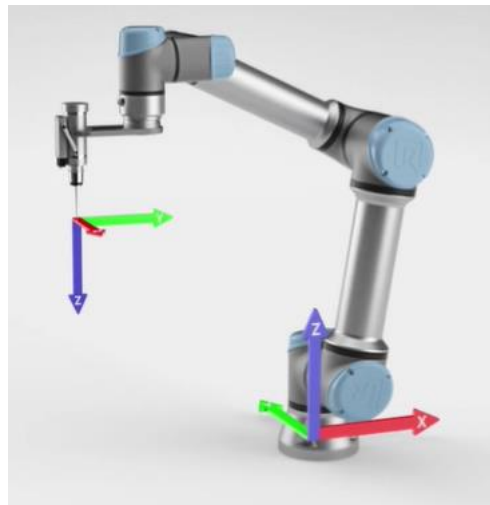


Figura 38: Funciones de la base y la herramienta en un robot UR5.

Sin embargo, es posible definir nuevas funciones de usuario que puedan utilizarse como referencia para definir puntos de paso. De esta manera, al re-definir la función con otra posición u orientación, todos los puntos de paso definidos en base a ella reajustarán de manera automática también su posición y orientación.

Para definir nuevas funciones de usuario, hay que entrar en el menú “Funciones” de la pestaña “Instalación”. Para crear una función alineada con una superficie se utilizará la opción “Plano”. Una vez insertada, es preciso definir la posición del origen y de los sentidos positivos del Eje X e Y.

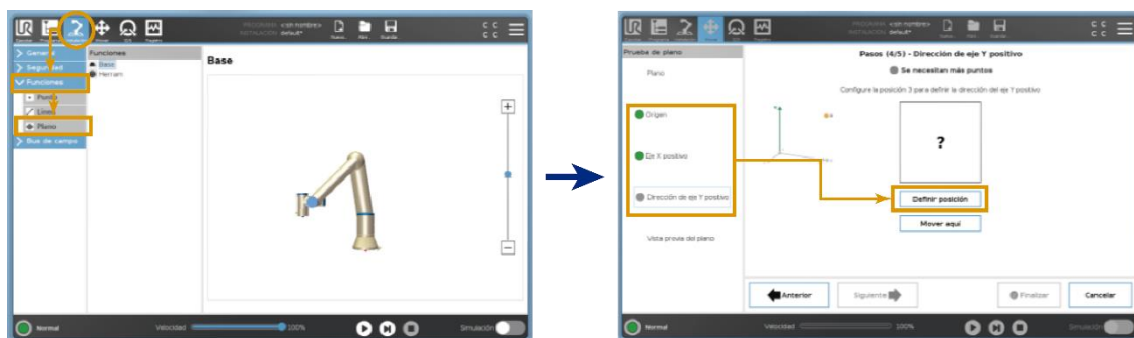


Figura 39: Definición de una función de plano



Una vez definido el plano de función, es posible utilizarlo como base para definir los puntos de pasos de movimiento tipo “MoveP” o “MoveL”. Para ello, al introducir la instrucción de movimiento correspondiente, antes de definir los puntos de paso, es necesario especificar la función con respecto a la que se quieren definir dichos puntos.



Figura 40: Configuración de la función de referencia en una instrucción de movimiento.

Cuando se manipula la posición y orientación de los puntos de pasos, puede resultar útil modificar el sistema de referencia con respecto al que se especifican. Para ello, se puede seleccionar en el menú desplegable “Función” el sistema de referencia que se quiere utilizar:

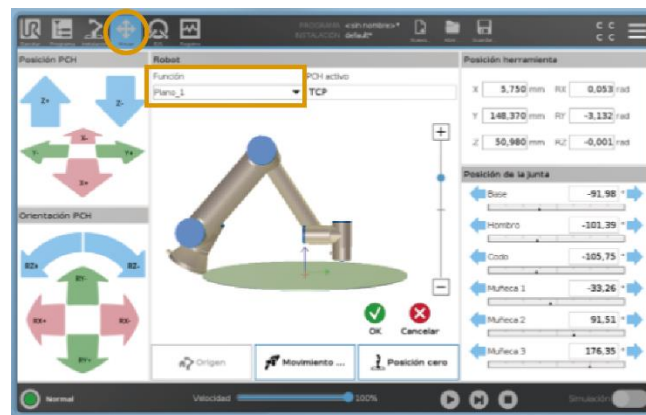


Figura 41: Uso de funciones para definir la posición y orientación de puntos de paso

Una vez definidos todos los puntos de paso necesarios, es posible reajustar el sistema de referencia de la función para que todos estos puntos de paso se redefinan de manera acorde. Para ello, solamente hay que modificar el plano utilizado para la definición de dichos puntos, desde la interfaz en la que se definió.



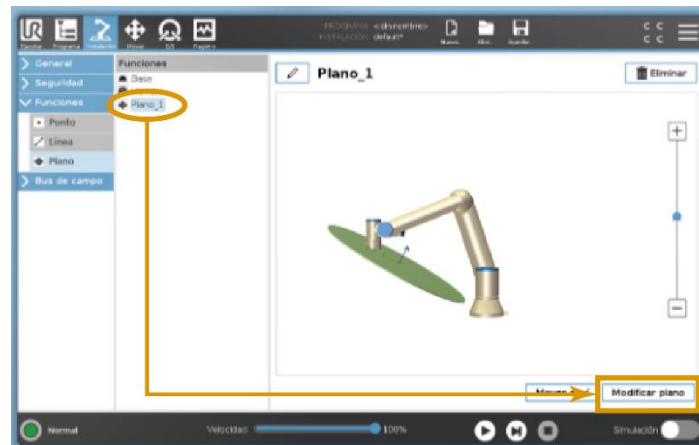


Figura 42: Modificación del plano de función

## 7 Ejercicio práctico de programación de un robot UR5

El trabajo a desarrollar con el UR5 consta de tres partes diferenciadas: entrenamiento previo a la sesión experimental, programación de una aplicación de dispensación de pegamento y programación de una aplicación de *pick and place*. Toda la práctica se evaluará sobre 10 puntos.

### 7.1 Formación online previo al trabajo experimental

En primer lugar, cada alumno ha de realizar de manera individual una introducción a la programación del equipo. Para ello tienen a su disposición el material formativo proporcionado por Universal Robots en su programa de formación en línea:

<https://academy.universal-robots.com/es/formacion-en-linea-gratuita/formacion-en-linea-de-e-series/>

Se habrá de realizar los itinerarios básico y profesional y, opcionalmente, el itinerario de aplicaciones. La realización de este entrenamiento previo es **condición indispensable** para poder trabajar con las unidades reales disponibles en el laboratorio.

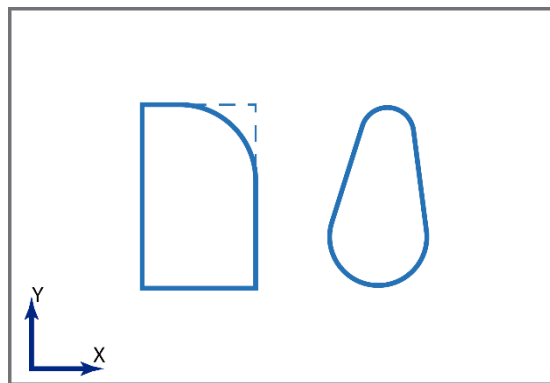
**Evaluación:** se habilitará una entrega en aula virtual para entregar los diplomas acreditativos de haber superado con éxito los dos itinerarios de formación obligatorios (1 punto)

### 7.2 Aplicación de dispensación de pegamento

Se trata de una aplicación en la que el robot ha de recorrer el contorno de dos piezas para dispensar adhesivo. El plano de trabajo debe ser configurable, de manera que, al cambiar de posición, el sistema sea capaz de volver a ejecutar las trayectorias sin tener que volver a tomar todos los puntos.

Los requisitos para dicha aplicación son los que se listan a continuación:

- Se definirá una región de protección en torno a la mesa de trabajo, de manera que el robot no podrá salir de la misma bajo ninguna condición.
- El operador habrá de introducir en qué tipo de pieza el robot tiene que aplicar pegamento y cuantas repeticiones se deberá hacer sobre la misma pieza. Basándose en lo que el operador introduzca, el robot ejecutará la parte correspondiente del programa y luego volverá a una posición "Origen". Un ejemplo de las piezas seleccionadas aparece en la siguiente figura. Se trata de una pieza rectangular con una de las esquinas definida como un arco en el que la herramienta habrá de moverse a velocidad constante, y de una pieza compuesta por dos semicircunferencias y dos lados rectos.



*Figura 43: Ejemplos de objetos de trabajo.*

- El programa debe ser lo suficientemente flexible para que, en caso de que la mesa de trabajo cambie de posición u orientación, se pueda reajustar el programa a las nuevas condiciones de trabajo de manera fácil.
- La herramienta sólo es capaz de realizar 5 trayectorias completas antes de requerir mantenimiento. La aplicación necesita llevar un conteo de las figuras completadas, de manera que una vez completada la quinta, se interrumpa la ejecución del programa y se lleve la herramienta a una posición de mantenimiento en la que el operario debe actuar sobre ella. Hasta que el operario no valide el mantenimiento en la consola, no se continuará con la ejecución normal de la rutina.

Para simular esta aplicación se utilizará una pizarra a modo de superficie de trabajo y una herramienta rotulador a modo de dispensador de pegamento. En la pizarra estarán definidos los puntos necesarios para la ejecución de las tres trayectorias pedidas. Se considerará como espacio de trabajo del robot una zona similar a la representada en la siguiente figura, en la que los planos limitan dicho espacio.



*Figura 44: Espacio de trabajo del robot*

**Evaluación:** Este trabajo deberá desarrollarse, orientativamente en 3-4h, una vez finaliza la aplicación, se pedirá a los alumnos que se ejecute para comprobar su funcionamiento. Se valorará el adecuado uso de puntos de aproximación y la selección de velocidades y tipos de movimientos. Si durante la ejecución se produce un choque o colisión con cualquier elemento (mesa, pizarra o el propio cuerpo del robot), la calificación máxima posible que se podrá obtener será del 50% del total.

Los ítems que se valorarán serán:

- Creación de la zona de seguridad: 0.75 puntos.
- Definición de la herramienta y comprobación de la situación del TCP: 0.75 puntos.
- Definición y ejecución de las trayectorias de trabajo: 1 puntos.
- Funcionamiento correcto de la aplicación (selección de piezas, repeticiones y mantenimiento): 1.5 puntos.
- Adaptación de las trayectorias tras un cambio en el plano de trabajo: 1 puntos.

### 7.3 Aplicación de pick and place

En este caso, se utilizará el robot para implementar una tarea de manipulación de elementos. Para ello, en la brida del robot se instalará una pinza de acción neumática como la de la siguiente figura.

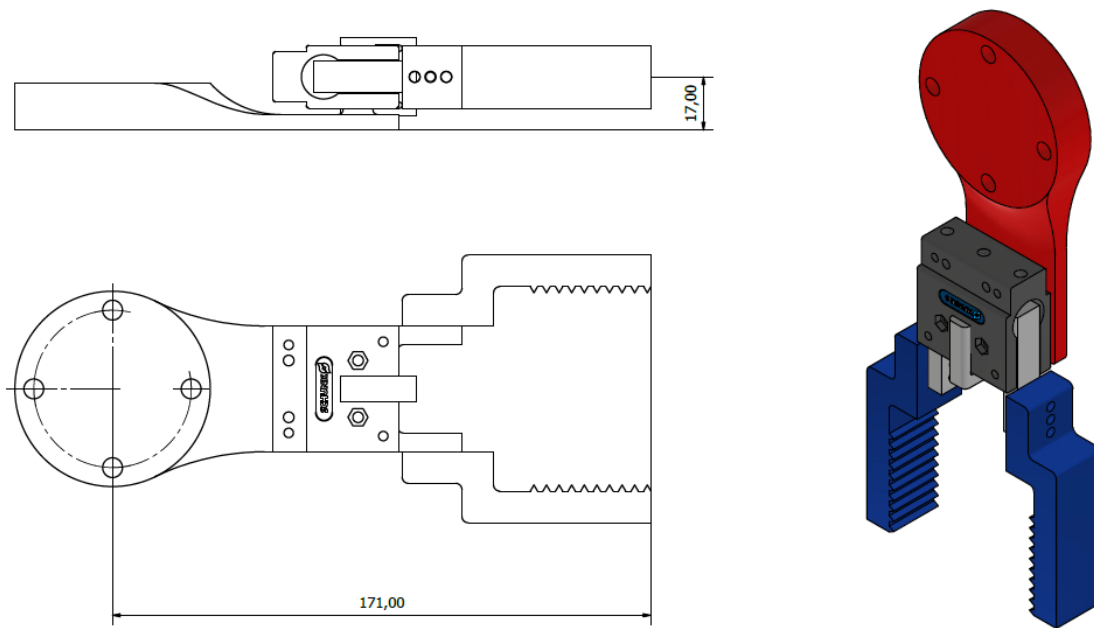


Figura 45: Pinza (dimensiones en mm)

El objetivo de la aplicación es simular el proceso de empaquetado y paletizado en una instalación industrial. Las piezas a manipular son bricks de 7x7x20cm.

Se cuenta con una plataforma sensorizada que será utilizada a modo de dispensador. Cuando las piezas se encuentran en dicha plataforma, el sistema ha de detectarlo y llevar la pieza a su posición correcta dentro del palé. Cada palé consta de 6 piezas, dispuestas según la siguiente estructura:

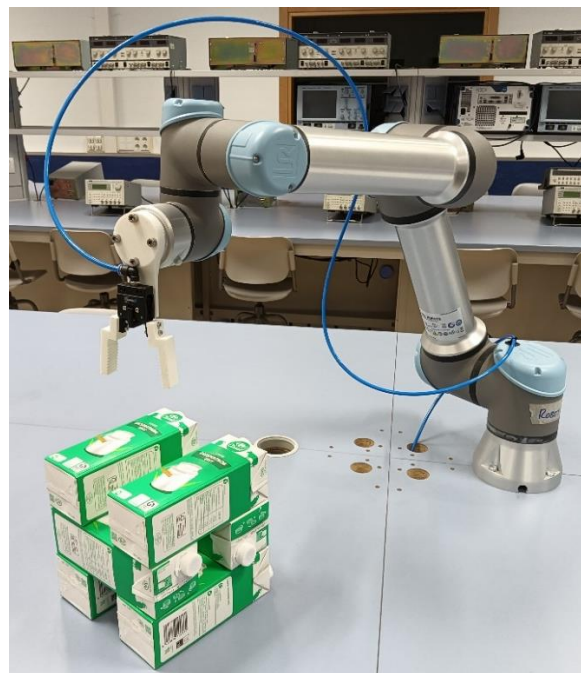


Figura 46: Ejemplo de palé completo

**Evaluación:** Este trabajo deberá desarrollarse, orientativamente en 2-3h, una vez finaliza la aplicación, se pedirá a los alumnos que se ejecute para comprobar su funcionamiento. Se valorará el adecuado uso de puntos de aproximación y la selección de velocidades y tipos de movimientos. Si durante la ejecución se produce un choque o colisión con cualquier elemento (mesa, piezas o el propio cuerpo del robot), la calificación máxima posible que se podrá obtener será del 50% del total.

Los ítems que se valorarán serán:

- Definición de la herramienta y comprobación de la situación del TCP: 0.5 puntos.
- Detección de piezas en el dispensador y ejecución de proceso “pick and place”: 1.5 pts
- Construcción de la estructura de paletizado: hasta 2 puntos. Para conseguir la puntuación completa deben apilarse seis bricks, definiendo el menor número posible de puntos y utilizando movimiento basados en variables. Se penalizará una construcción incompleta y la definición de puntos innecesarios.

## 8 Referencias

1. Itinerarios de Formación en Línea para los cobots de la e-Series (<https://academy.universal-robots.com/es/formacion-en-linea-gratuita/>). Universal Robots.
2. Manual de Usuario del UR5e. Universal Robots (<https://www.universal-robots.com/download/manuals-e-series/user/ur5e/513/user-manual-ur5e-e-series-sw-513-spanish-es/>)
3. Material Didáctico de Universal Robots. Curso Básico, libro de ejercicios.