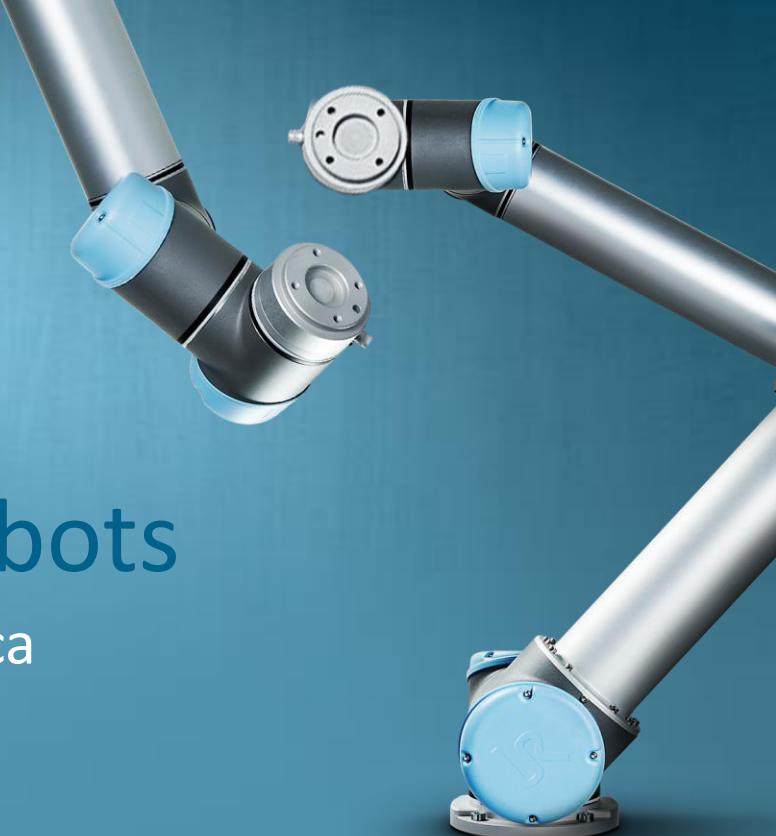




UNIVERSAL ROBOTS

Universal Robots

Capacitación Básica





Bienvenido a la Capacitación Básica

- Curso técnico de formación que incluye terminología básica y práctica en el manejo del robot
- Maximizar el conocimiento y ganar experiencia con la programación de los robots UR
- Curso teórico con ejercicios prácticos



objetivos de aprendizaje

Tras completar el curso deberás ser capaz de:

- Manejar el robot de forma segura
- Realizar programas de aplicación simples
- Gestionar archivos de programa
- Conectar señales de Entrada/Salida (I/O)
- Leer y modificar programas
- Realizar operaciones de diagnóstico y mantenimiento

Presentación del Profesor

Nombre: Felipe Marina

Rol: Tech Support Argentina

fmarina@murten.com.ar

COBOT v.s. Robot industrial

Categoría	COBOT	Robot industrial
Posicionamiento	Movible. Poco espacio.	Fijo/Requiere gran esfuerzo. Gran área de trabajo.
Seguridad	Sensores que permiten el trabajo en conjunto.	Requiere barreras de seguridad.
Versatilidad	Fácilmente reprogramable y reutilizable.	Único propósito.
Cooperación	Trabajo en conjunto con operario	Nula.

1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes / Wizard
3	Comandos básicos 1	9	Modbus TCP / Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes / Wizard
3	Comandos básicos 1	9	Modbus TCP / Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

¿Qué viene en la caja?

- Caja 1
 - Brazo del Robot
- Caja 2
 - Caja de control
 - Cable de alimentación
 - Soportes de montaje
 - Cable para la herramienta
 - Manuales
 - El brazo del Robot y el control están protegidos por bolsas antiestáticas ESD



Brazo del Robot

- Diseño del manipulador
 - 6 ejes / articulaciones
 - Robot articulado (*muy similar al brazo humano*)
 - Diseño modular
 - Rango movimiento +/- 360° (cada eje)
UR3: rotación infinita en el último eje
- Versiones disponibles
 - UR3
 - UR5
 - UR10
 - UR16



Brazo del Robot

- Denominación y tamaño de las articulaciones

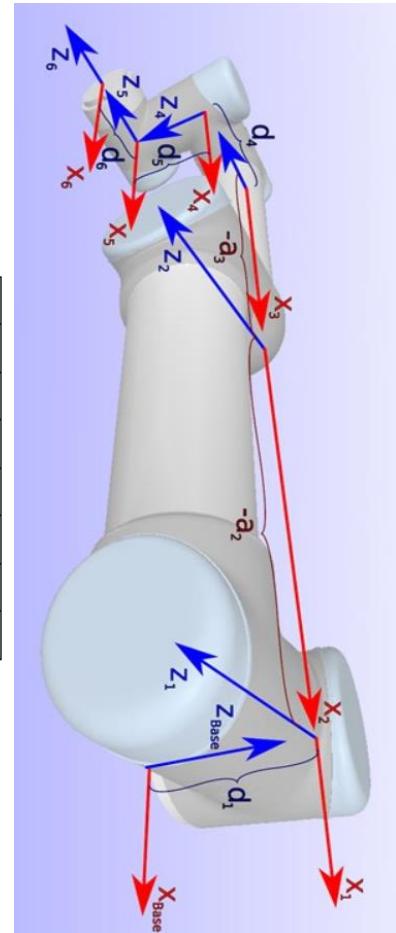
	UR3	UR5	UR10
[J1] Base/ <i>Base</i>	Size 2	Size 3	Size 4
[J2] Hombro/ <i>Shoulder</i>	Size 2	Size 3	Size 4
[J3] Codo/ <i>Elbow</i>	Size 1	Size 3	Size 3
[J4] Muñeca 1/ <i>Wrist 1</i>	Size 0	Size 1	Size 2
[J5] Muñeca 2/ <i>Wrist 2</i>	Size 0	Size 1	Size 2
[J6] Muñeca 3/ <i>Wrist 3</i>	Size 0	Size 1	Size 2



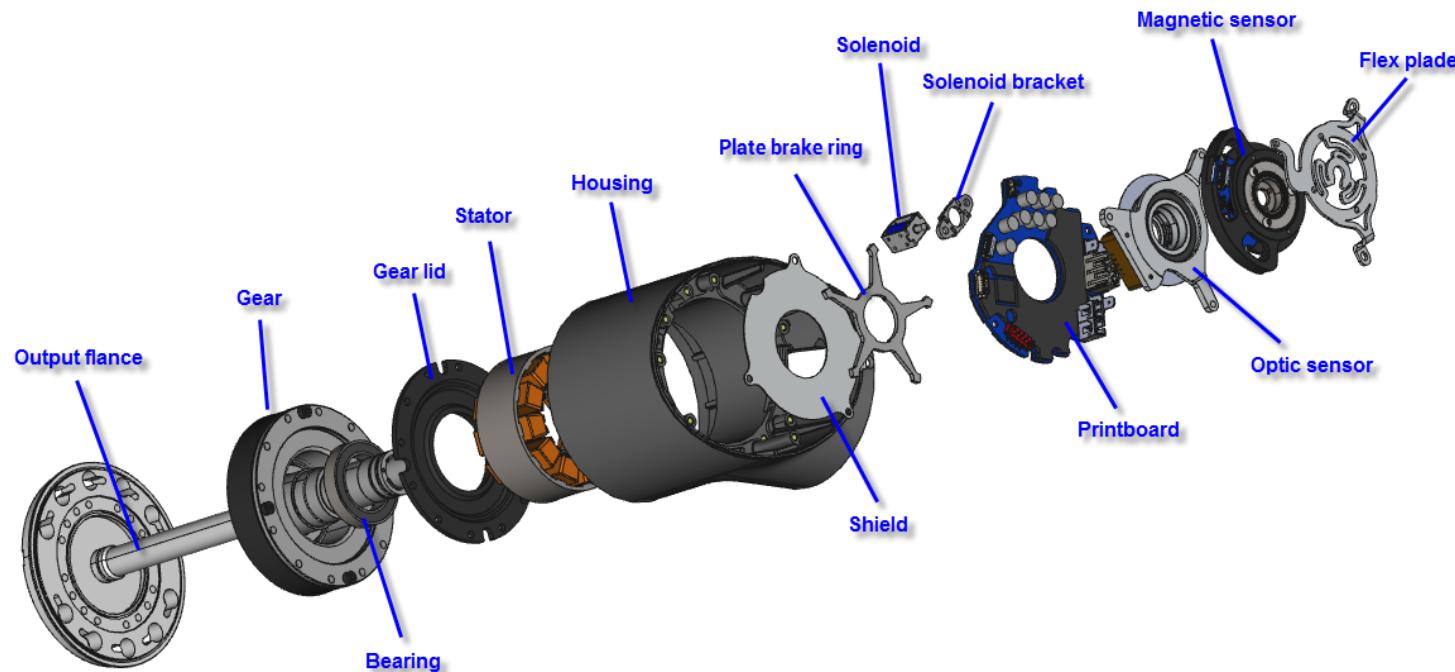
Brazo del Robot

- Denominación y tamaño de las articulaciones

UR10e							
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]	Dynamics	Mass [kg]	Center of Mass [m]
Joint 1	0	0	0.1807	$\pi/2$	Link 1	7.369	[0.021, 0.000, 0.027]
Joint 2	0	-0.6127	0	0	Link 2	13.051	[0.38, 0.000, 0.158]
Joint 3	0	-0.57155	0	0	Link 3	3.989	[0.24, 0.000, 0.068]
Joint 4	0	0	0.17415	$\pi/2$	Link 4	2.1	[0.000, 0.007, 0.018]
Joint 5	0	0	0.11985	$-\pi/2$	Link 5	1.98	[0.000, 0.007, 0.018]
Joint 6	0	0	0.11655	0	Link 6	0.615	[0, 0, -0.026]



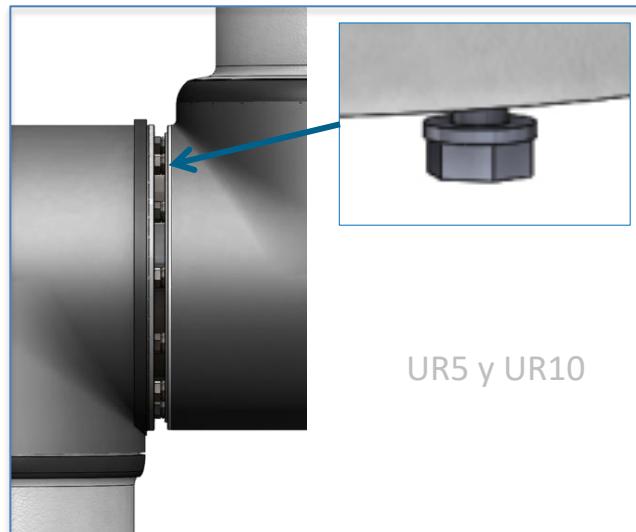
Explicación detallada de las articulaciones



Resumen – Tamaño de Articulaciones

Las Articulaciones del UR5 y UR10 son intercambiables

Ejemplo: “Size 3” de un UR5 puede ser usada como una “Size 3” en un UR10

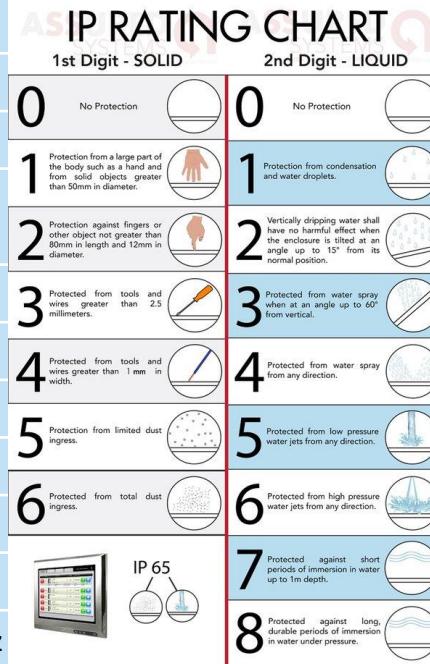


UR3 **no** es intercambiable con UR5 o UR10 debido a la diferente manera en las que se instalan juntas



Especificaciones

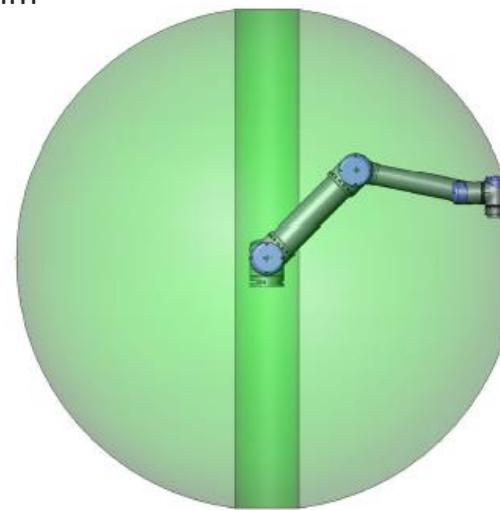
	UR3e	UR5e	UR10e
Carga útil	3 kg.	5 kg.	12.5 kg.
Alcance	500 mm	850 mm	1300 mm
Rango de las articulaciones	+/-360° Rotación infinita de la última articulación	+/-360°	+/-360°
Repetitividad	+/-0.03 mm	+/-0.03 mm	+/-0.05 mm
Velocidad máx. Eje	180°/s y 360°/s	180°/s	120°/s y 180°/s
Velocidad máx. Tool	1000 mm/s	1000 mm/s	1000 mm/s
Peso	11.0 kg	18.4 kg	33kg
Clasificación IP	IP64	IP54	IP54
Temperatura	0-50°C	0-50°C	0-50°C
Alimentación	100-240V AC, 50-60 Hz	100-240V AC, 50-60 Hz	100-240V AC, 50-60Hz



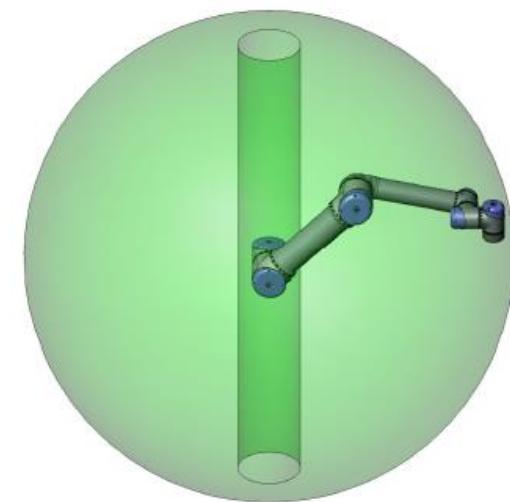
- Especificaciones adicionales en www.universal-robots.com

Área del trabajo del Robot

- Área de trabajo del UR10e
 - Esfera aproximada de Ø1300 mm
- Limitación
 - Volumen cilíndrico alrededor del centro de la Base



Vista frontal



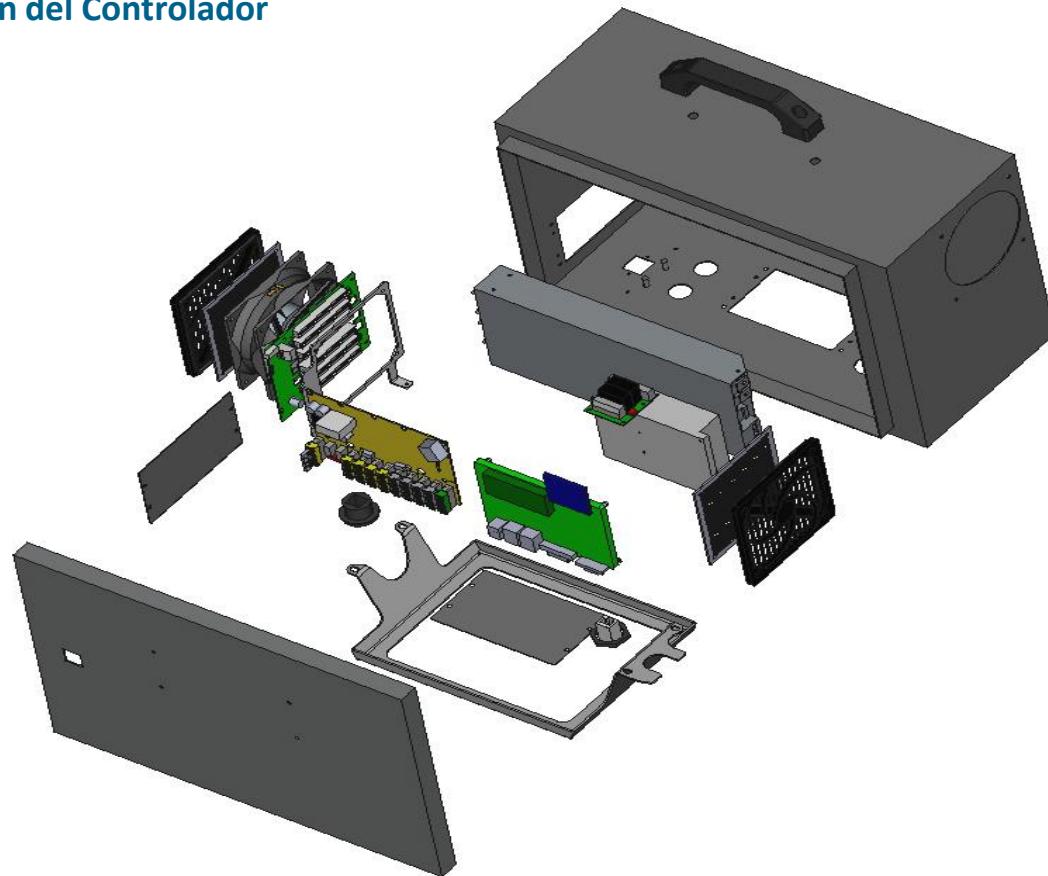
Vista inclinada

Controlador (*Control Box or Controller*)

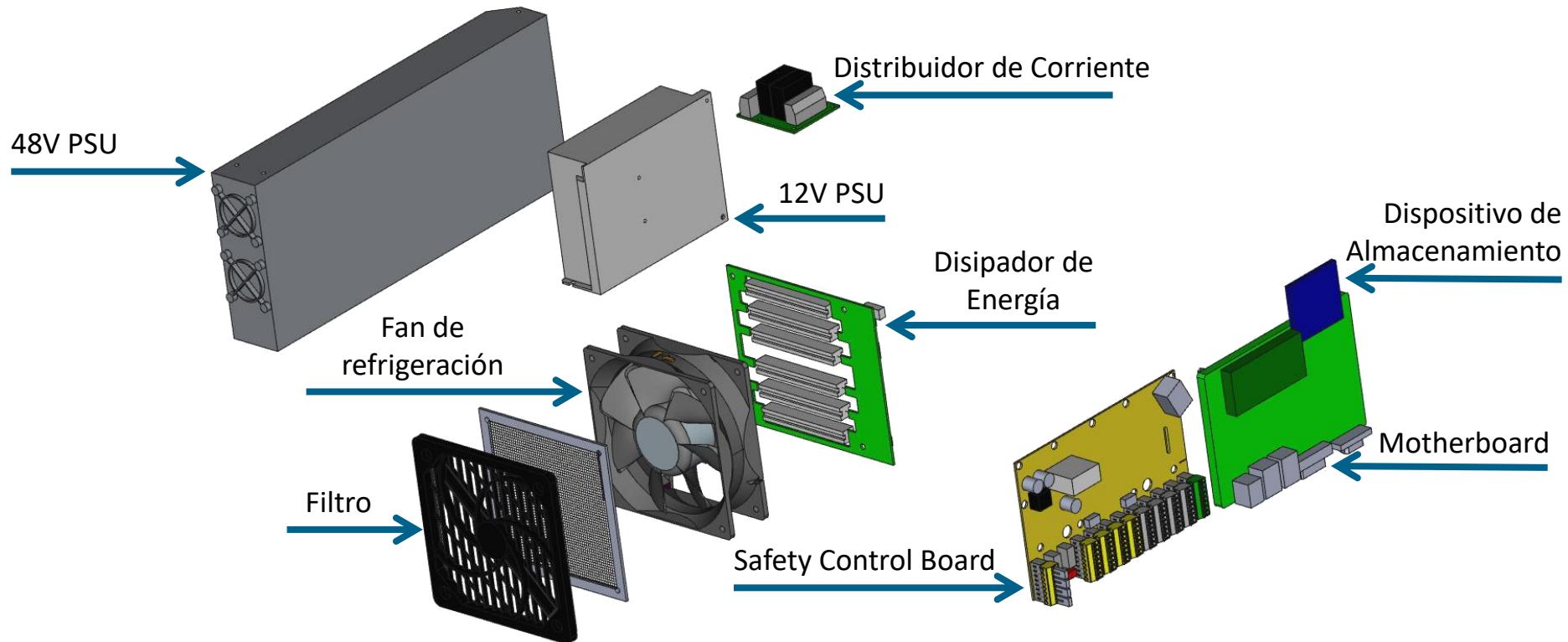
- Contiene
 - Flashcard/USB con el Software
 - Alimentación del brazo robótico
 - Sistema de Seguridad
 - Comunicación con equipos periféricos
- Conectores
 - Alimentación 220/110 V AC
 - Ethernet
 - USB
 - Brazo Robótico



Configuración del Controlador

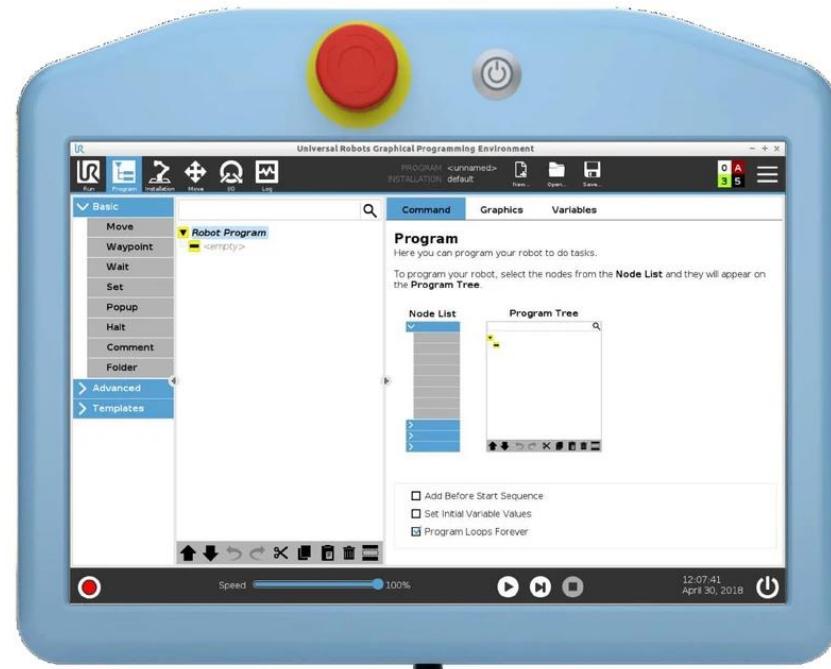


Configuración del Controlador



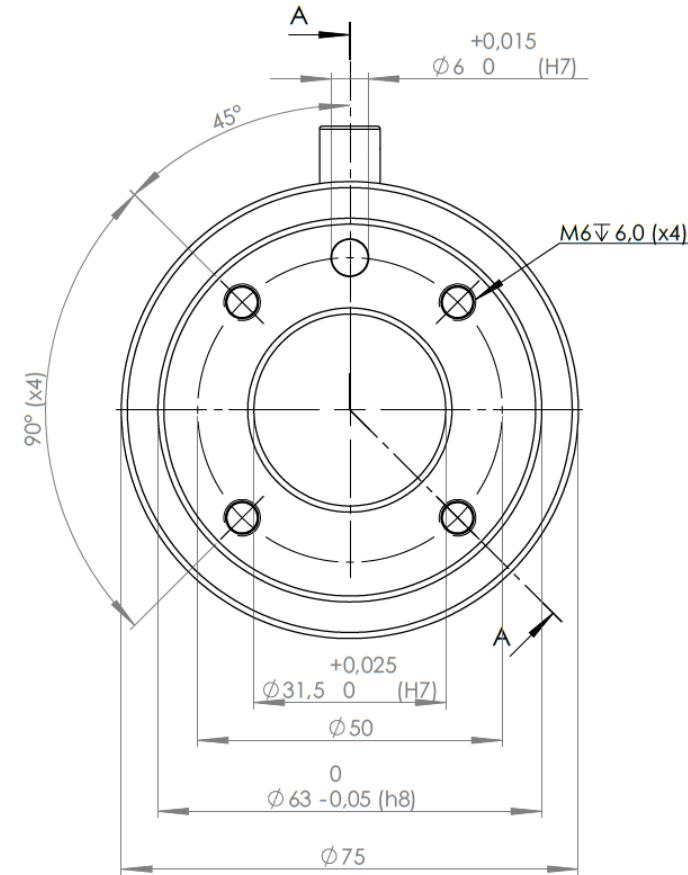
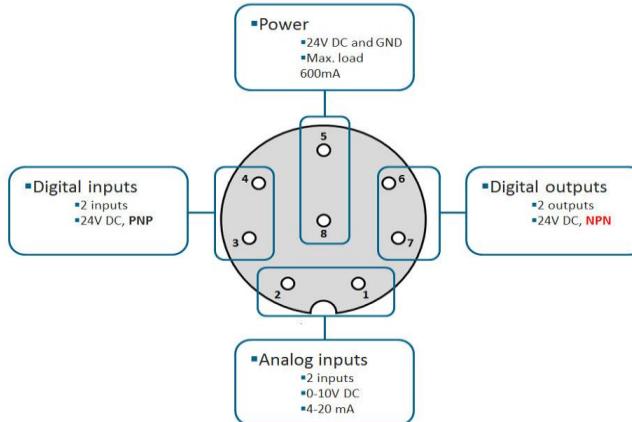
Consola de programación (*Teach Pendant*)

- Pantalla táctil
- Incluye
 - Botón encendido
 - Pulsador de emergencia (E-stop)
 - Pulsador movimiento libre (Freedrive)
 - Conexión USB



Montaje de la herramienta

- Estándar de montaje
 - ISO 9409-1-50-4-M6
- Conector de la herramienta
 - Conector de 8 pines
 - Lumberg RKMV-8-354



1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes / Wizards
3	Comandos básicos 1	9	Modbus TCP / Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

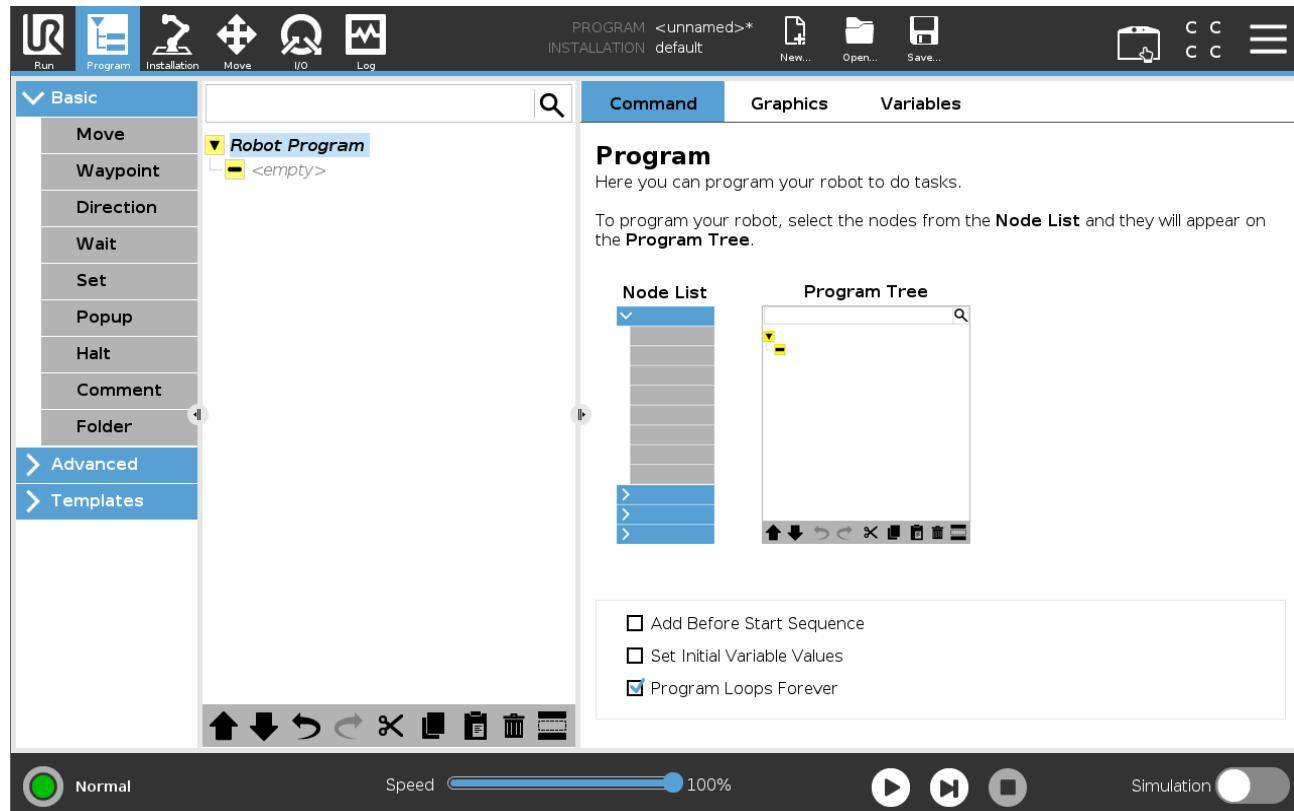
Introducción a PolyScope

PolyScope

- Desarrollado por UR
- Actualizaciones gratuitas, desde la página de UR.

Sistema operativo

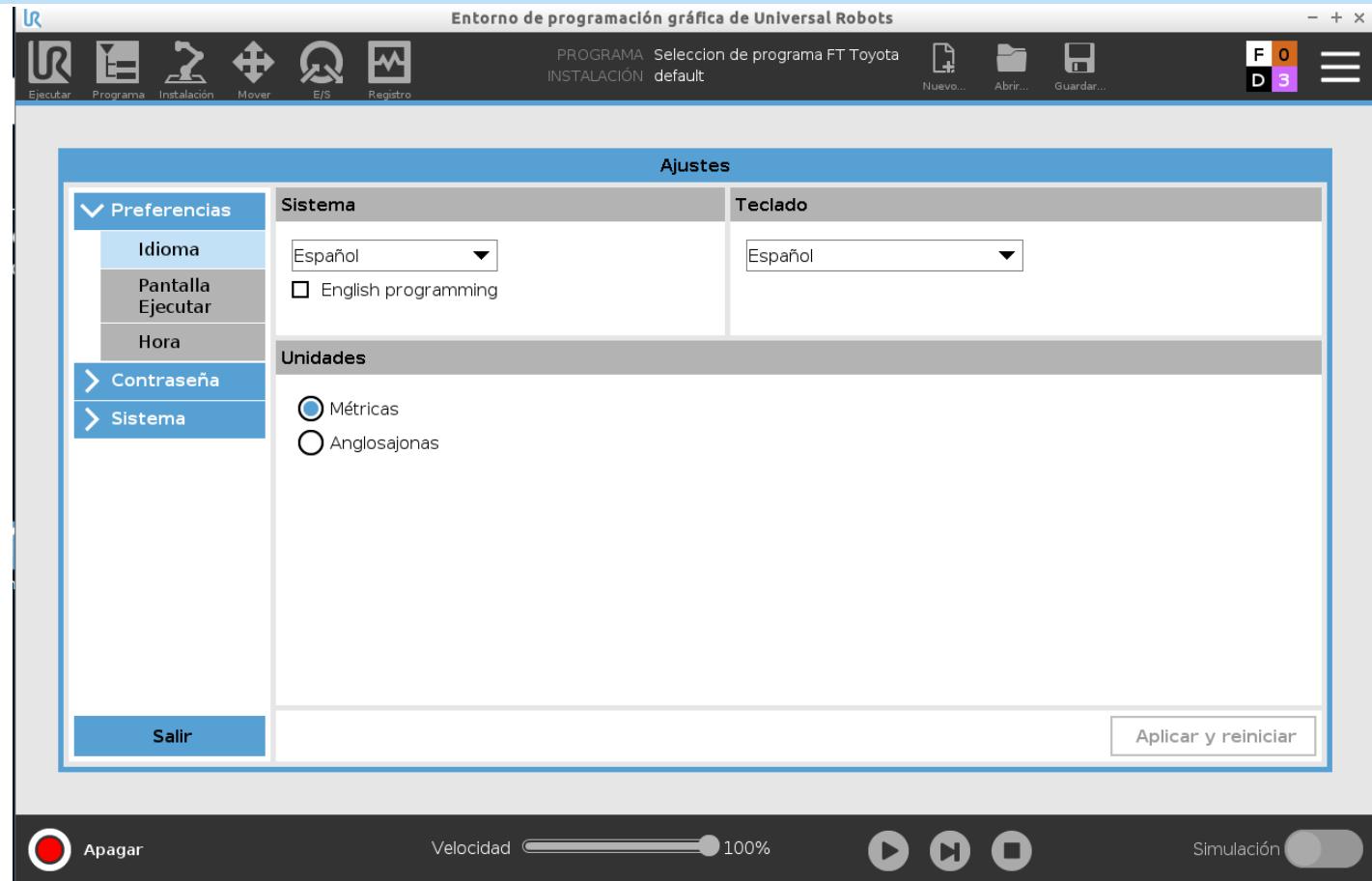
- Debian Linux



Configurar robot

Configurar robot

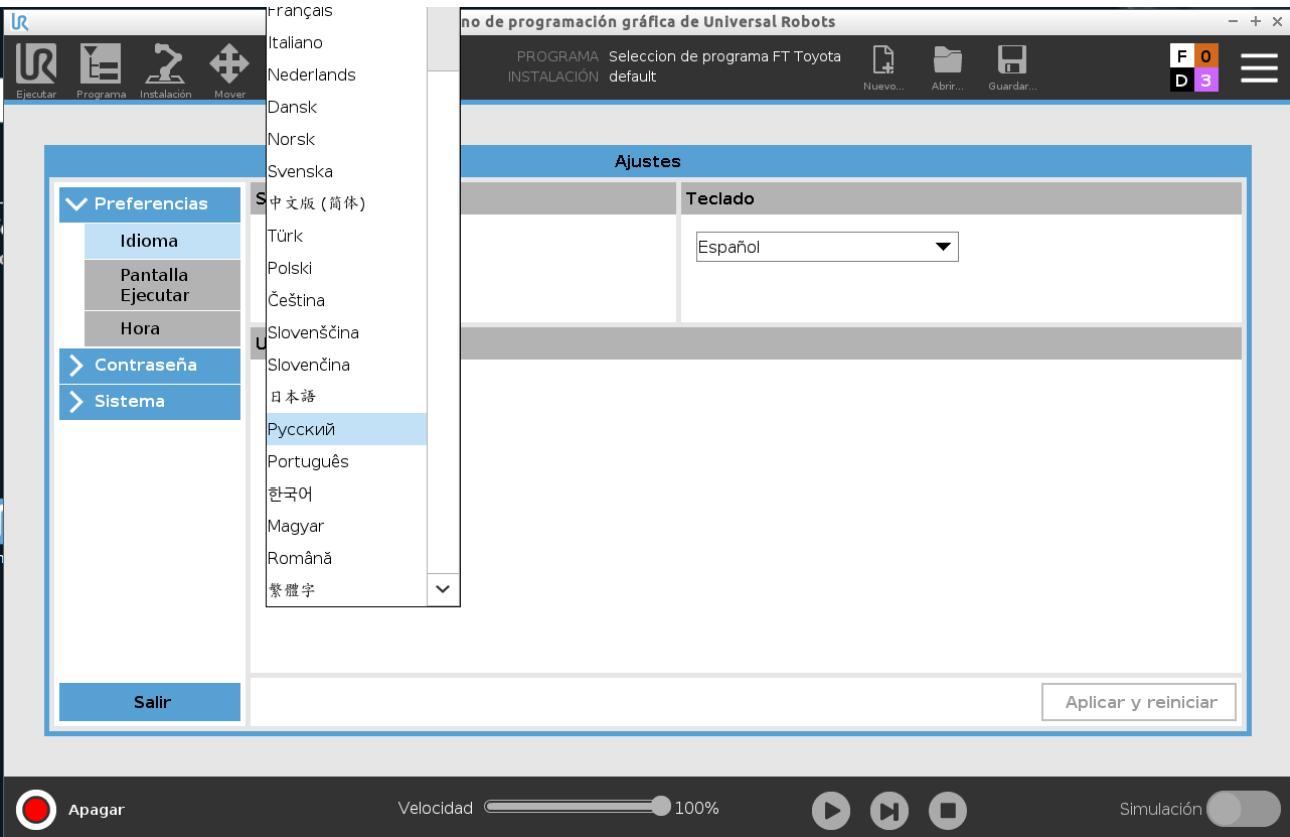
- Ajusta parámetros de software



Idioma y unidades

Idiomas

- 20 idiomas
- Programación en inglés
 - Mantiene las instrucciones y comandos en inglés



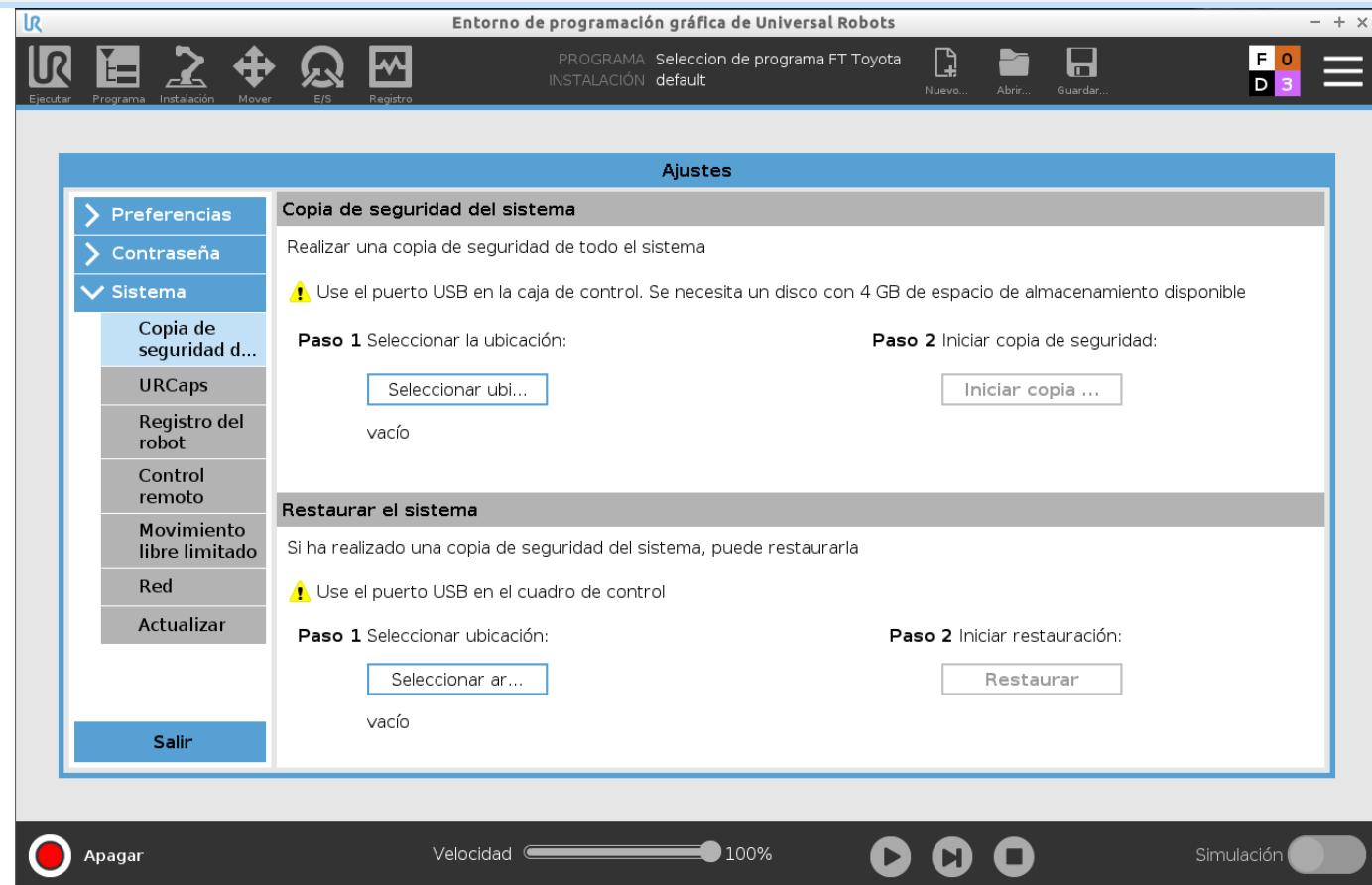
Unidades

- Sistema métrico
- Sistema anglosajón (U.S.)

Configurar robot

Ajustes de software

- Inicializar Robot
- Idioma y unidades
- Actualizar robot
- Fijar contraseña
- Calibrar Pantalla
- Configurar red
- Ajuste de hora
- Configuración de URCaps



Iniciar robot

- Estados del robot**

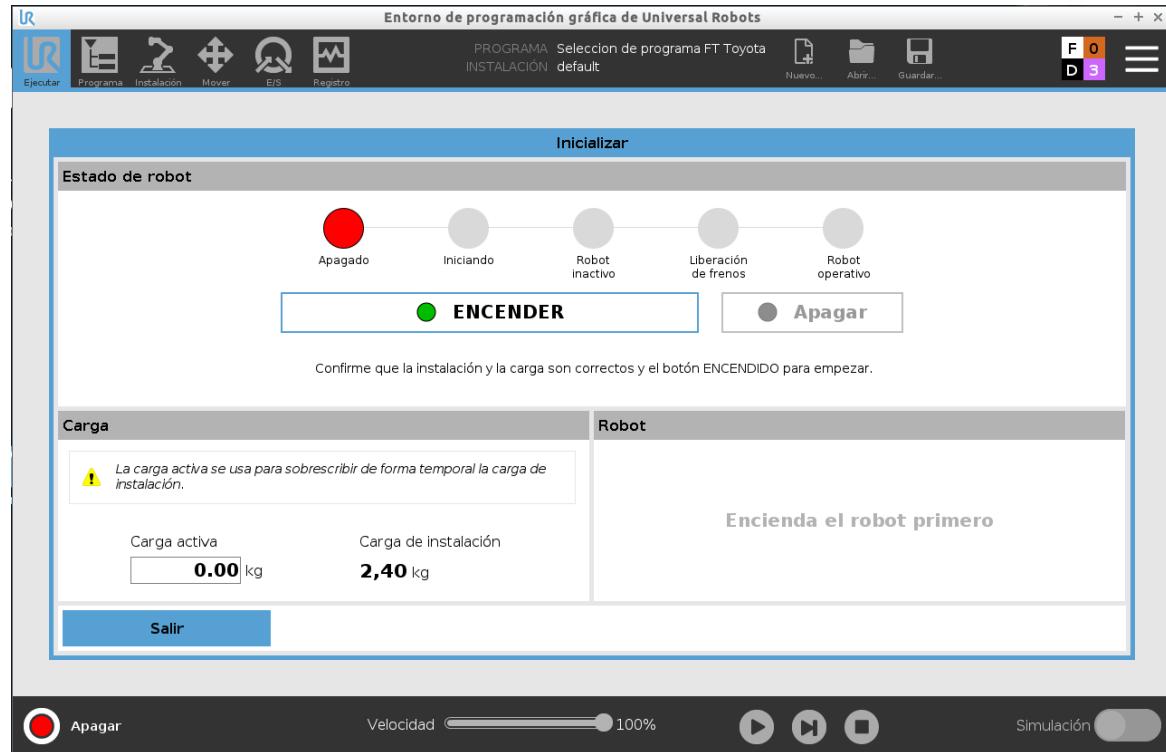
Estado	Poder	Frenos
Apagado	OFF	Activos
Inactivo	ON	Activos
Normal	ON	Libres

- Iniciar robot**

- Comprobar carga útil (payload)
- ENCENDER: Power ON
- INICIAR: Libera frenos

- Modo Retroceso (Backdrive)**

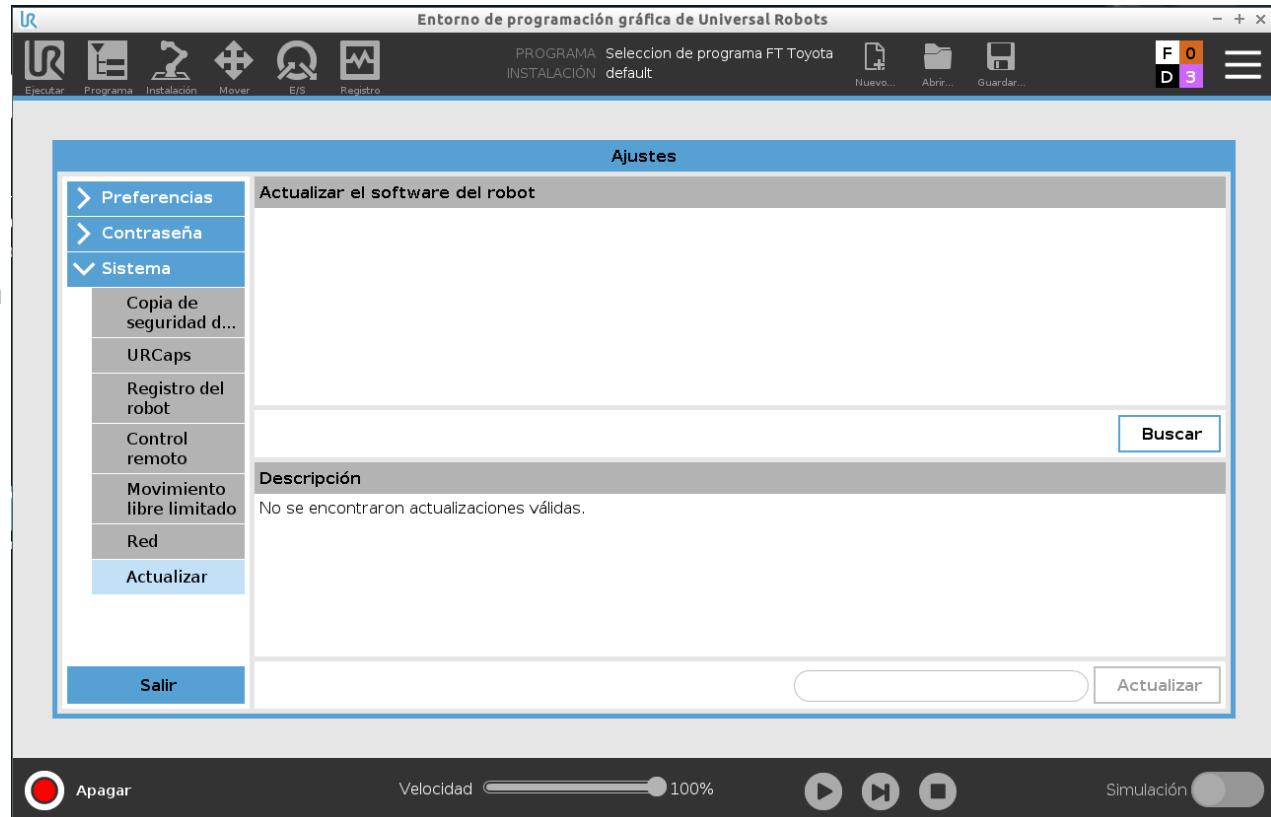
- Útil cerca de colisión
- Movimientos forzados para el usuario



Actualizar robot

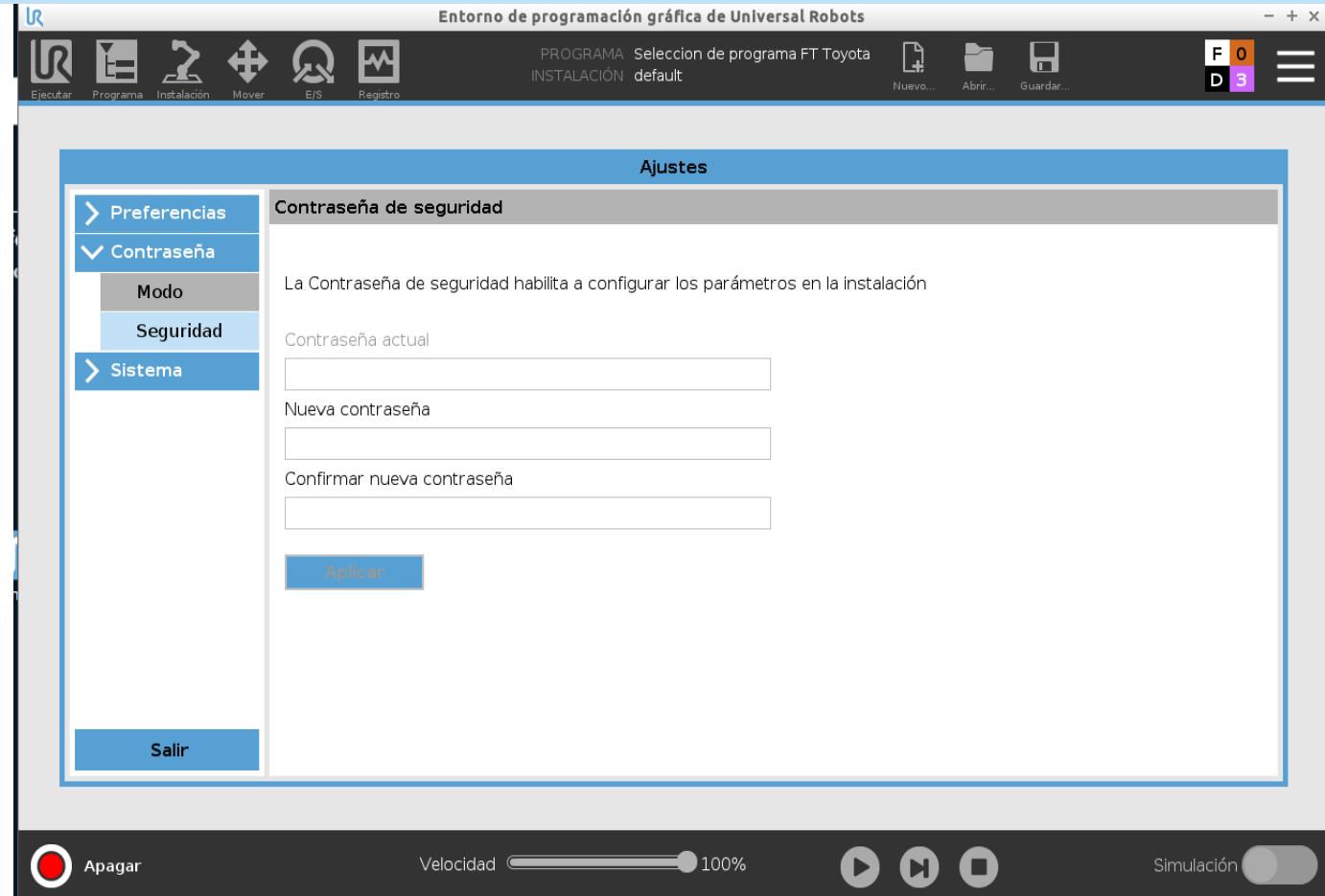
Software de Robot

- Actualizaciones gratuitas
- Se descarga desde la página de Soporte de UR



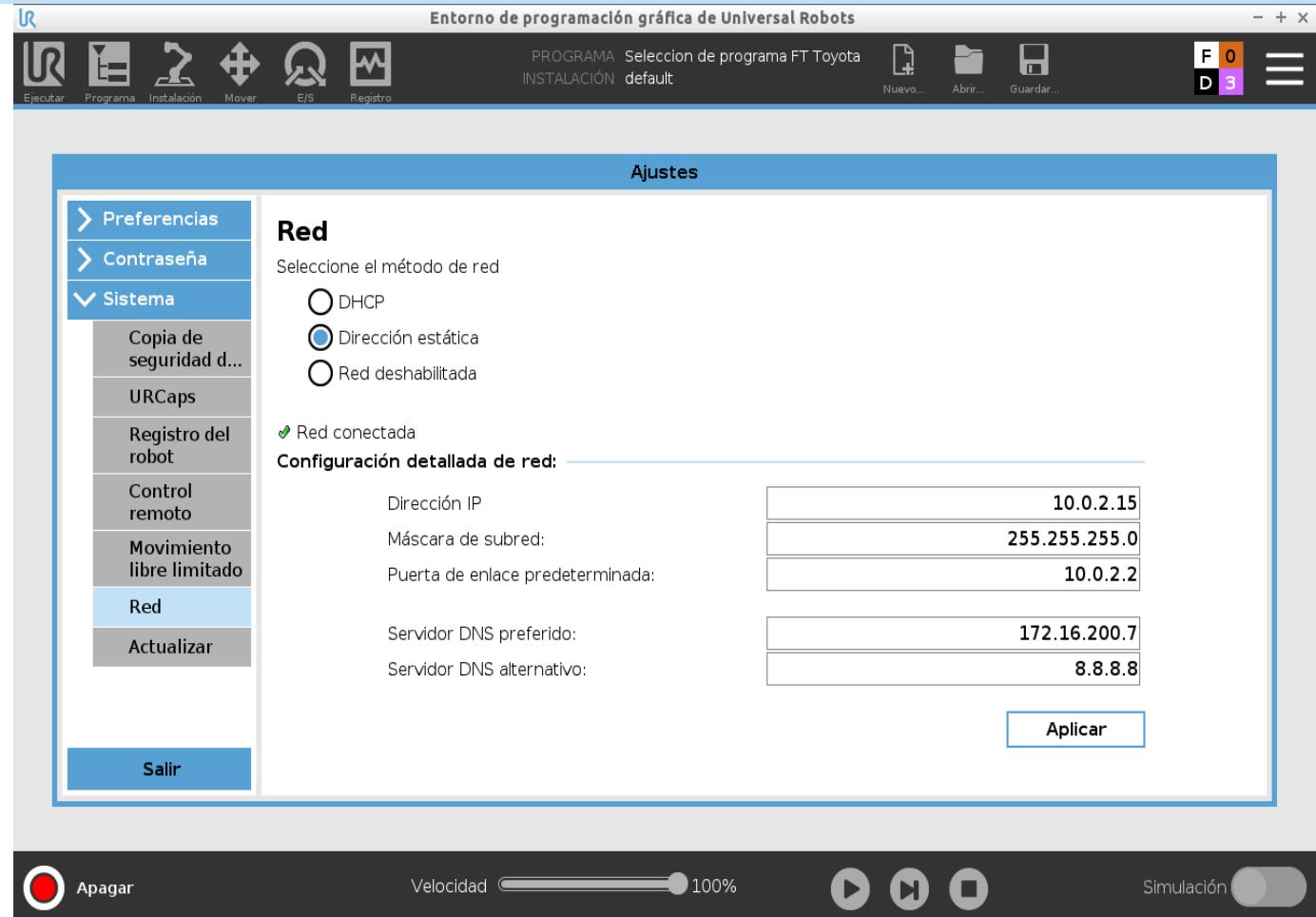
Fijar contraseña

- **Contraseña del sistema**
 - Limita el acceso a partes del software
- **Contraseña de seguridad**
 - Requerida para modificar parámetros de seguridad



Configurar red

- **Configuración de red**
 - La dirección IP se puede establecer en esta pantalla



Apagar

Velocidad

100%



Simulación

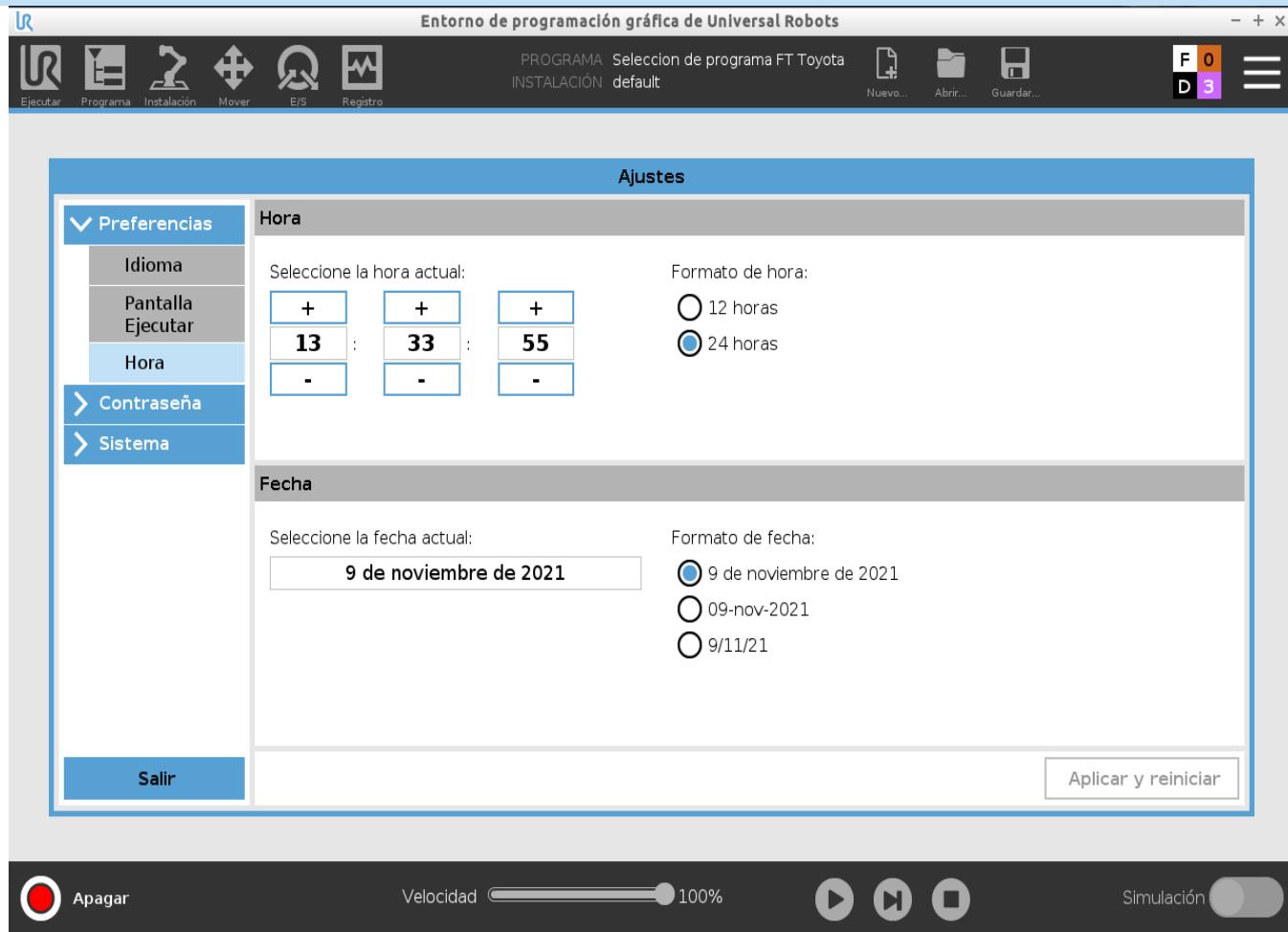
Ajuste de hora y fecha

- **Hora**

- Formato de hora
 - 24 horas
 - 12 horas

- **Fecha**

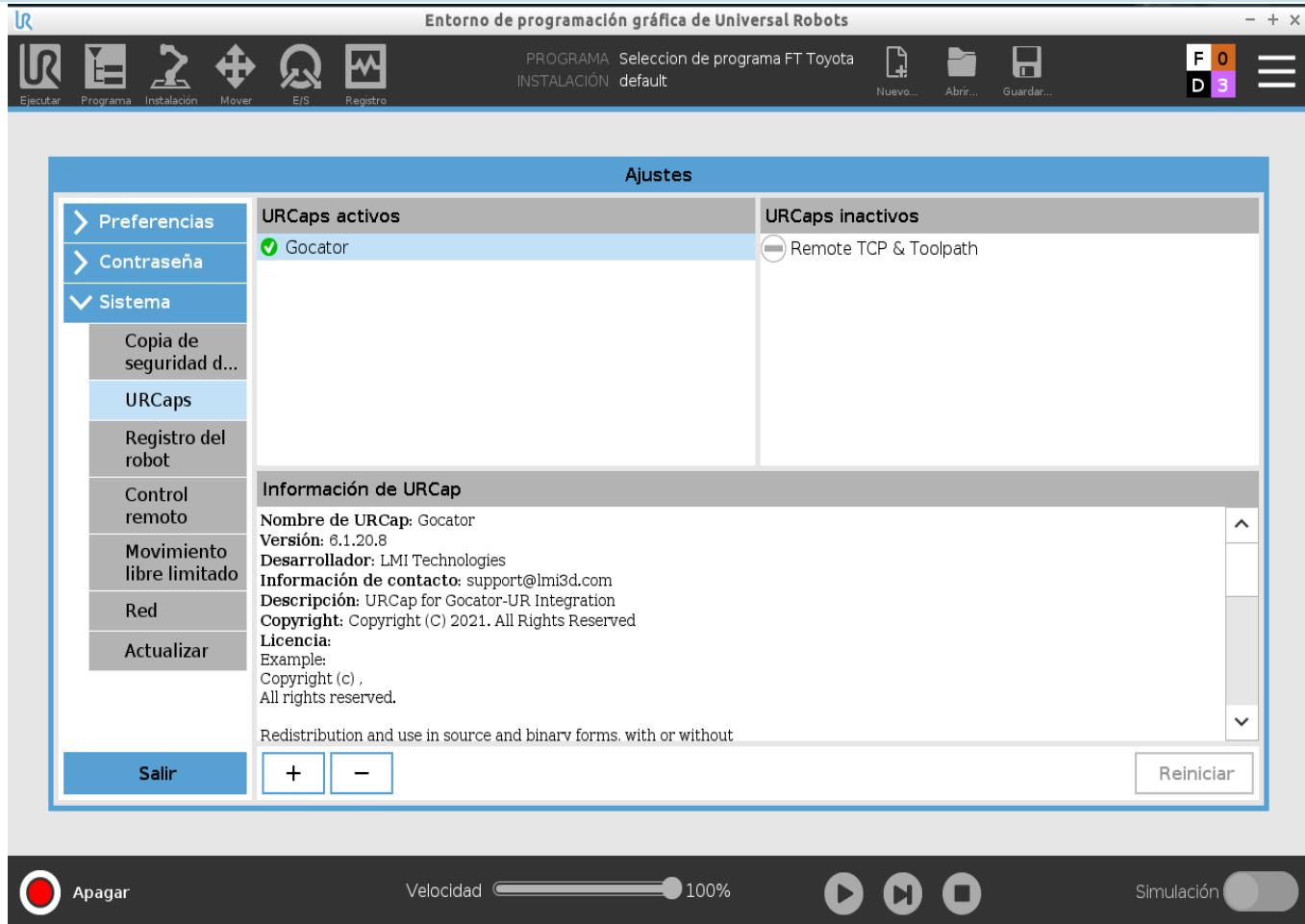
- Formato de fecha



URCaps

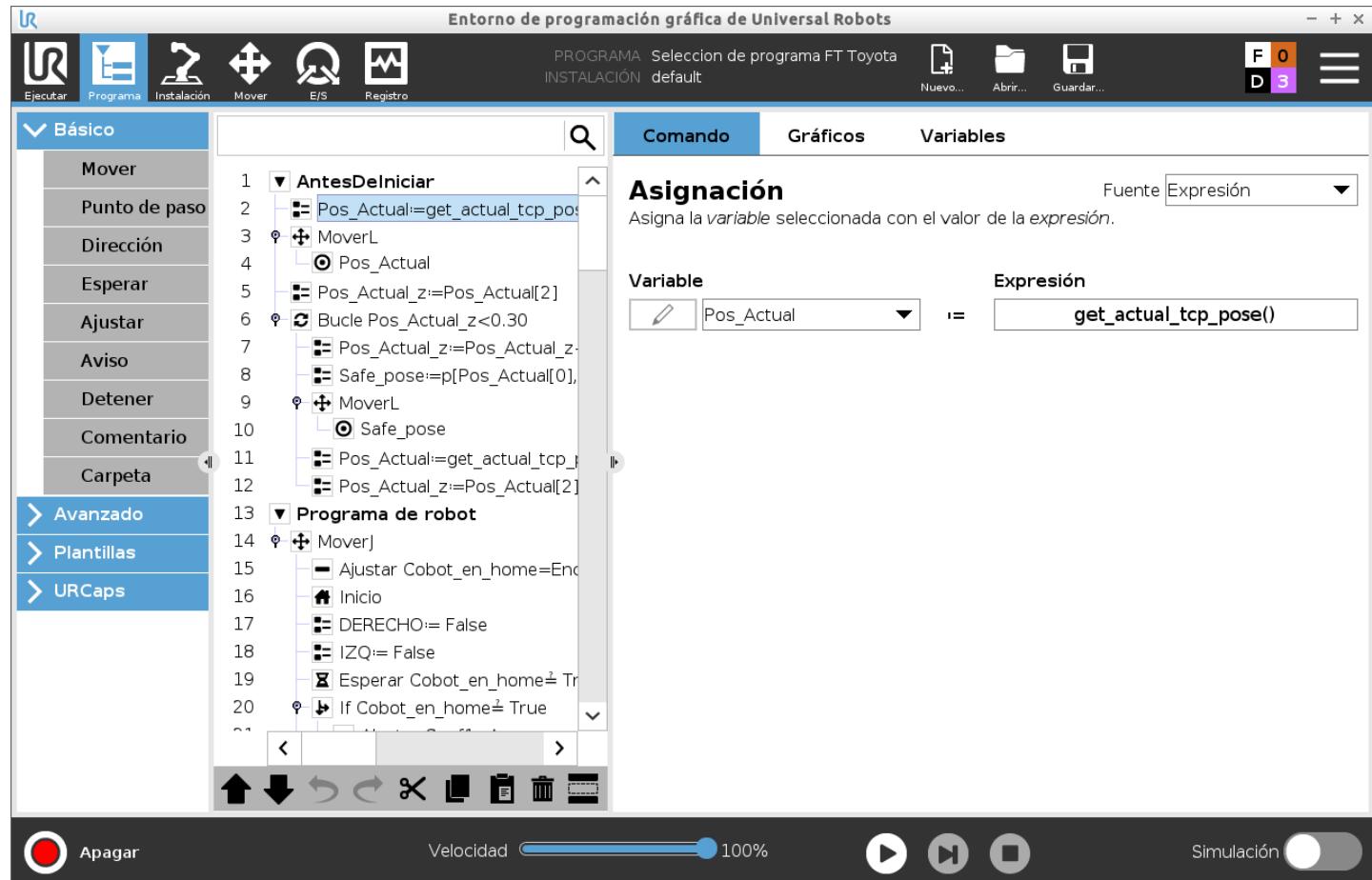
- **URCaps**

- Permite la integración de software de otros fabricantes con el “flujo de trabajo de UR”



Programar el robot

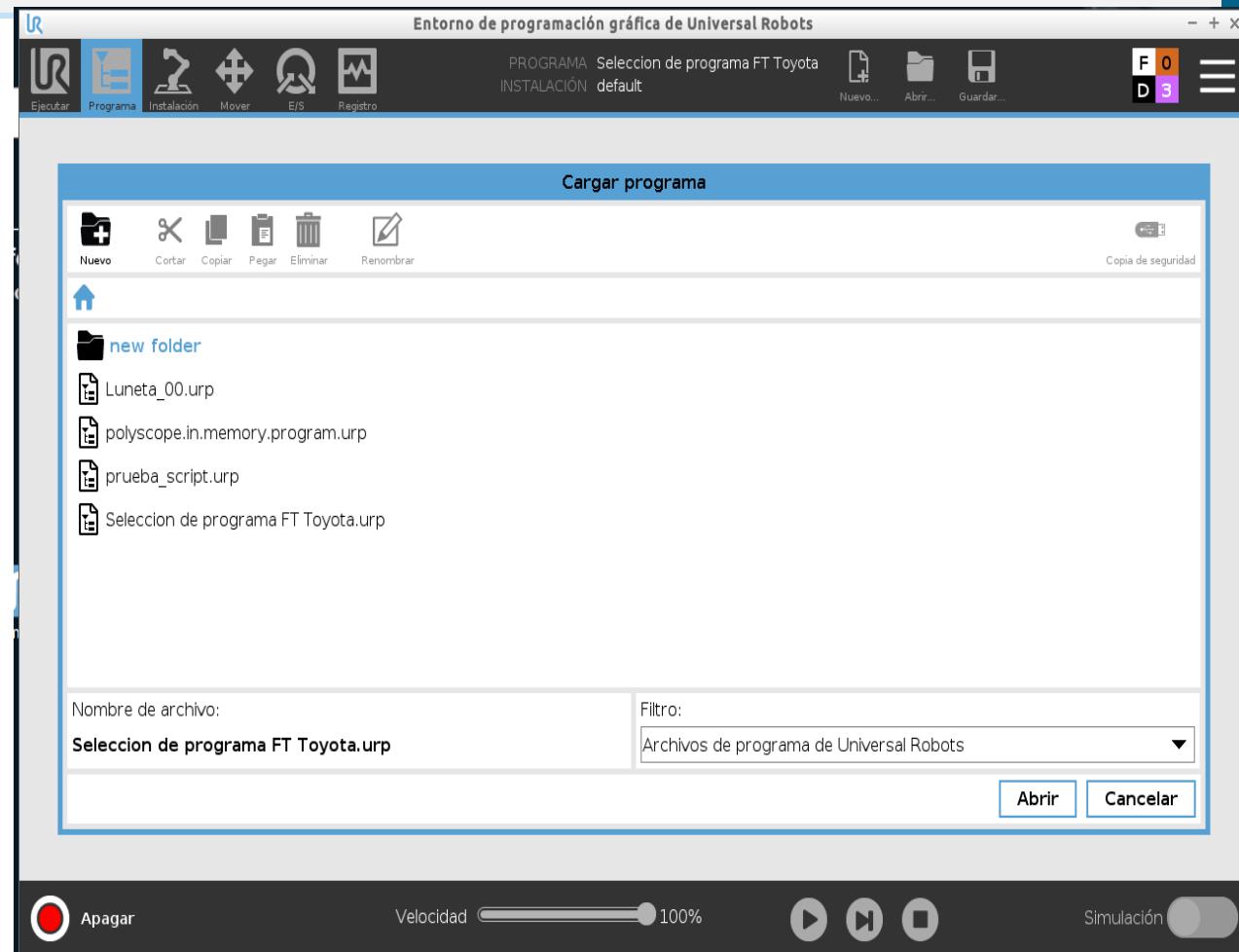
- Programar robot
 - Resumen de pestañas



Pestaña Programa

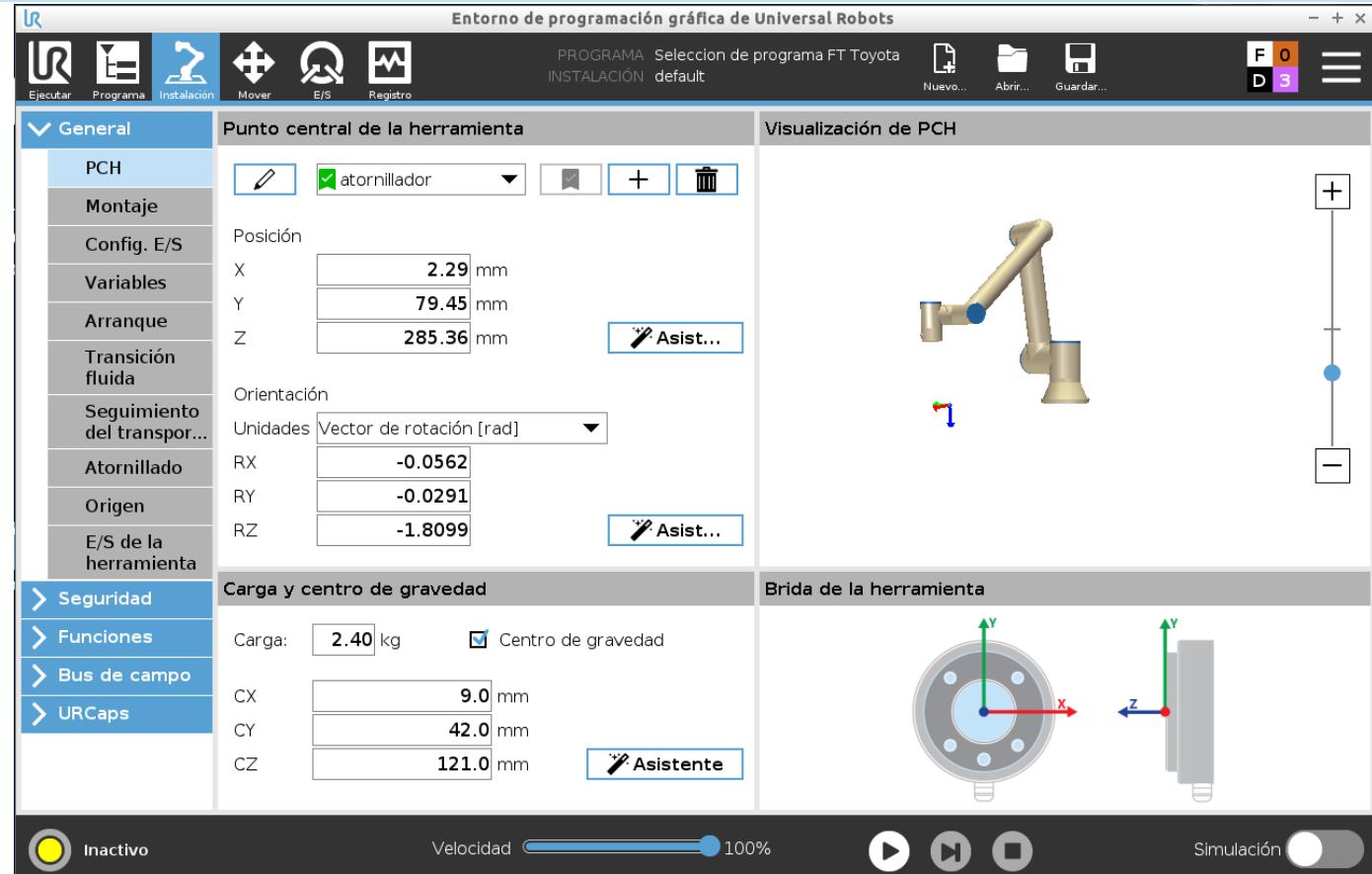
• Programa

- Cargar programa existente
- Crear programa nuevo



Pestaña Instalación

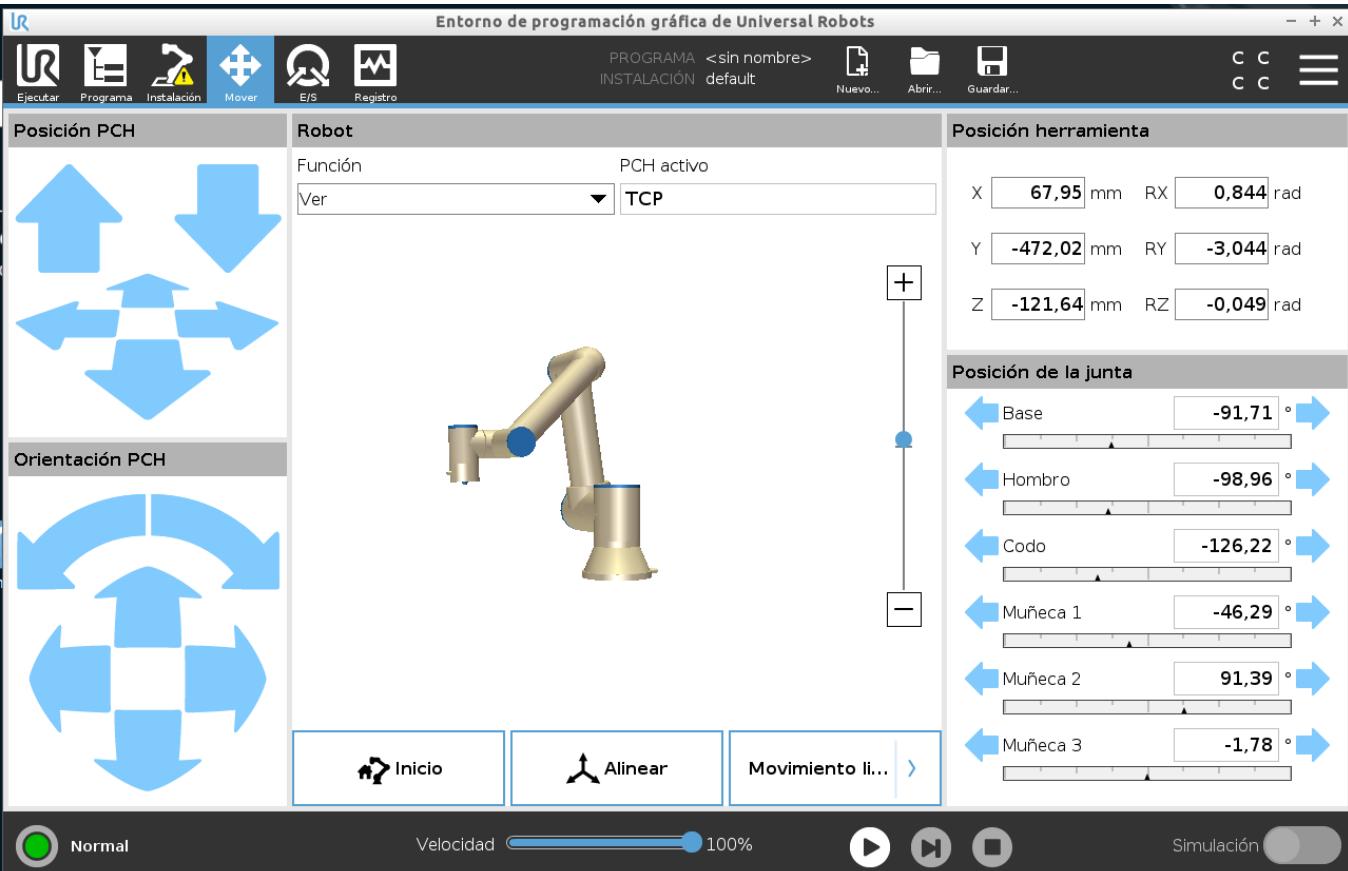
- **Configuraciones**
 - Configuración entorno
 - Configuración Seguridad



Pestaña Mover

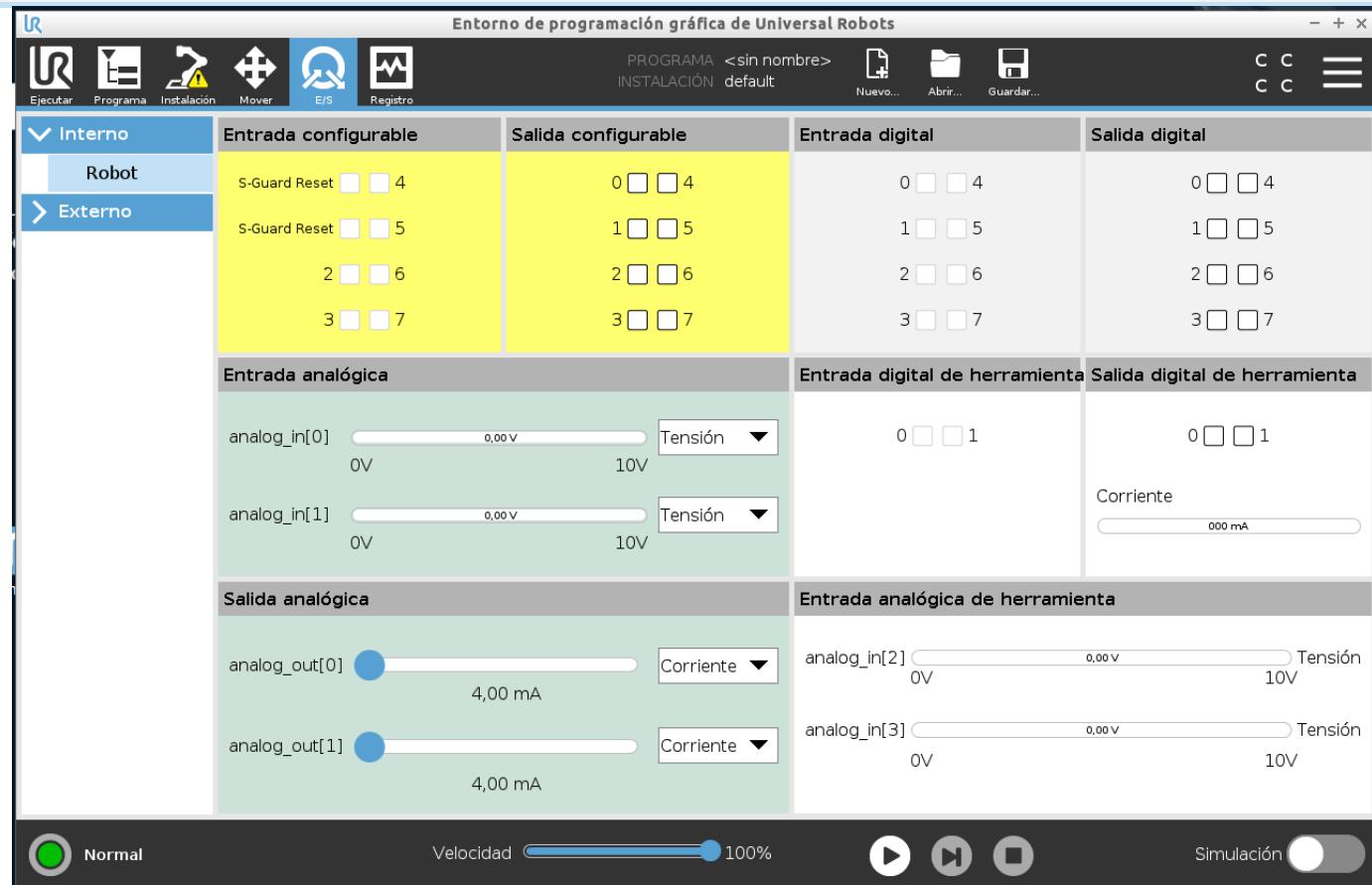
Mover

- Desplazamiento manual del robot respecto a la función seleccionada
- Muestra la posición del TCP (Tool Center Point) respecto
- la función seleccionada
- Desplazamiento manual de cada junta individualmente
- Muestra el ángulo de giro de cada una de las articulaciones



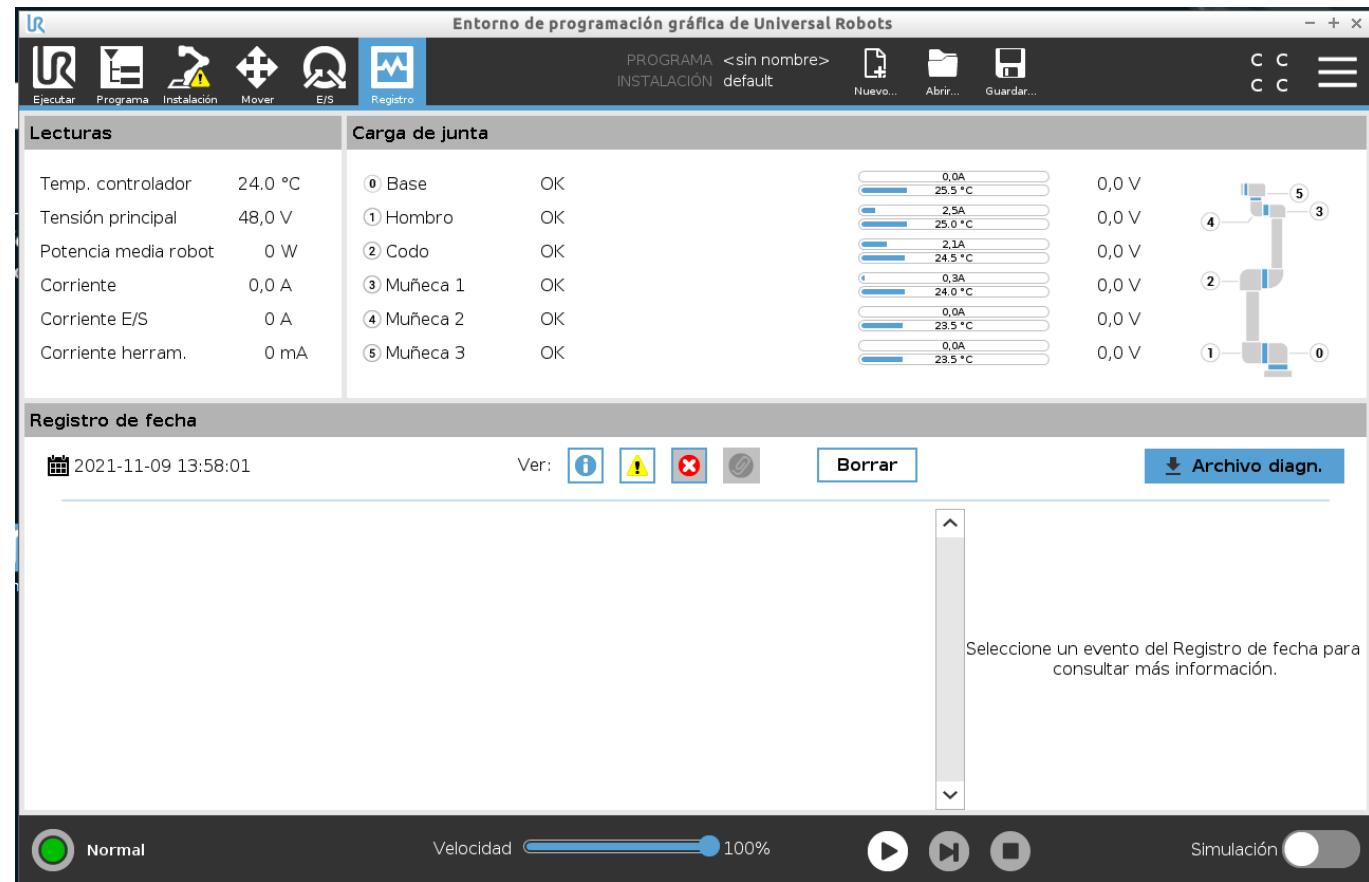
Pestaña I/O

- **I/O (Input/Output)**
 - Monitoreo de señales
 - Activación de señales
 - Configuración de señales analógicas



Pestaña Registro (Log)

- **Estado**
 - Controlador
 - Articulaciones
- **Registro de eventos (Log)**
 - Mensajes de información
 - Advertencias
 - Errores



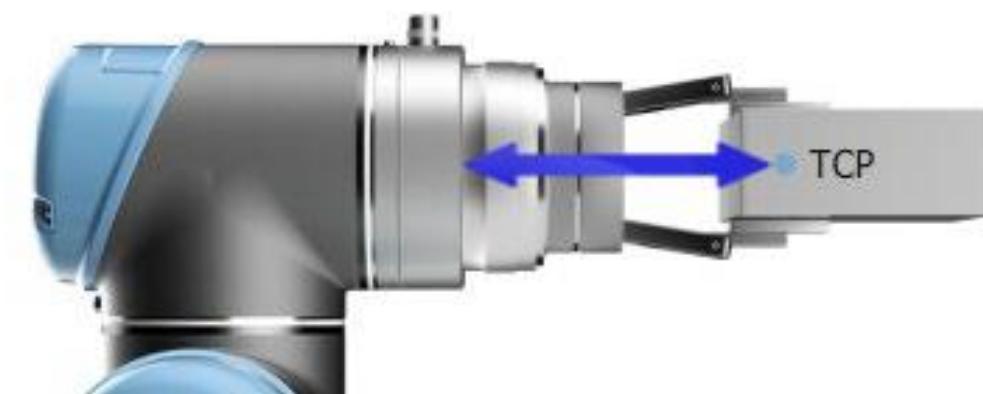
Configuración inicial

- **Configuraciones iniciales**
 - Configuración del TCP
 - Configuración Montaje (*Tool*)

TCP

Position

X 0.0 mm
Y 0.0 mm
Z 0.0 mm



Configuración inicial

• Configuraciones iniciales

- Configuración del TCP
- Configuración Montaje (*Tool*)

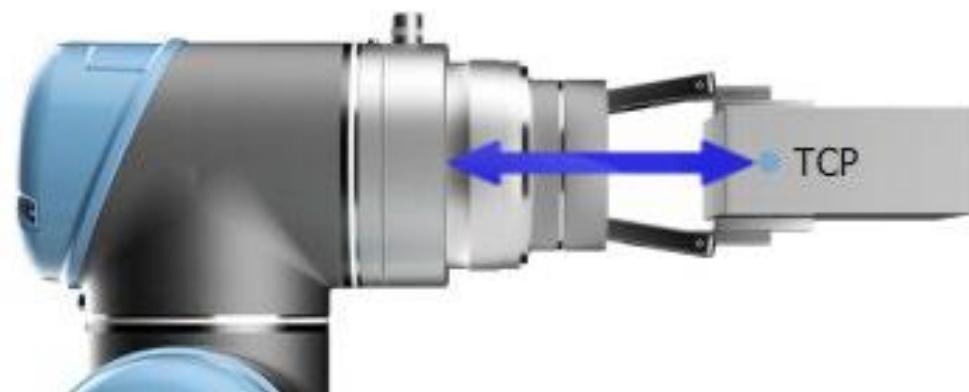
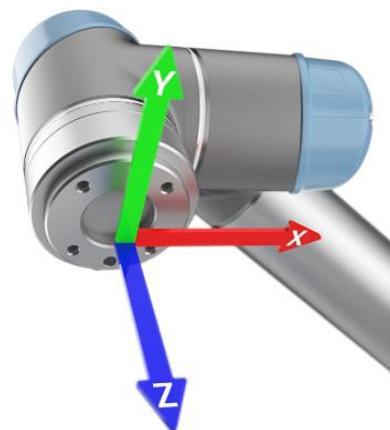
TCP

Position

X 0.0 mm

Y 0.0 mm

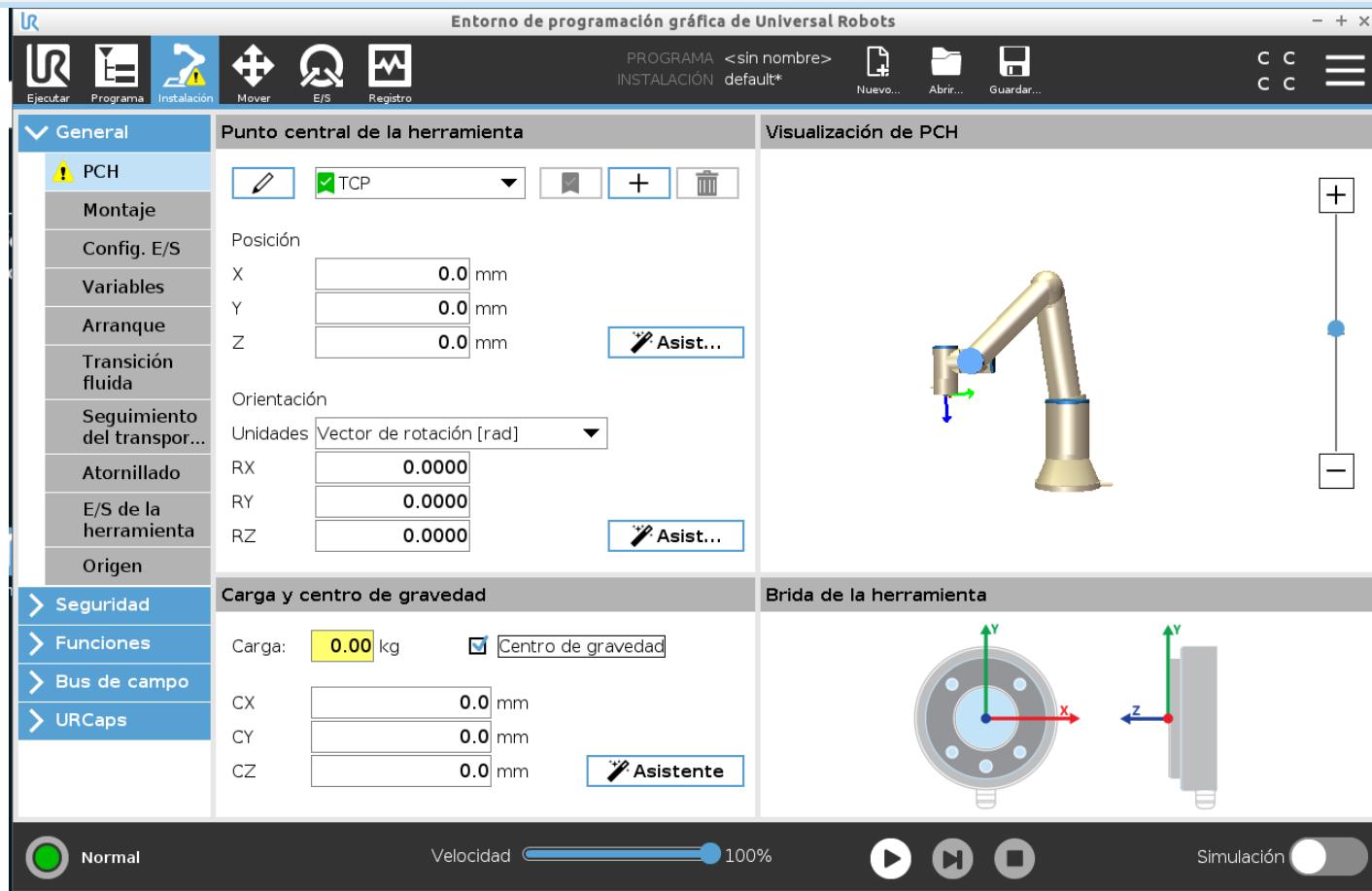
Z 0.0 mm



Configuración del TCP

Punto Central de Herramienta (TCP)

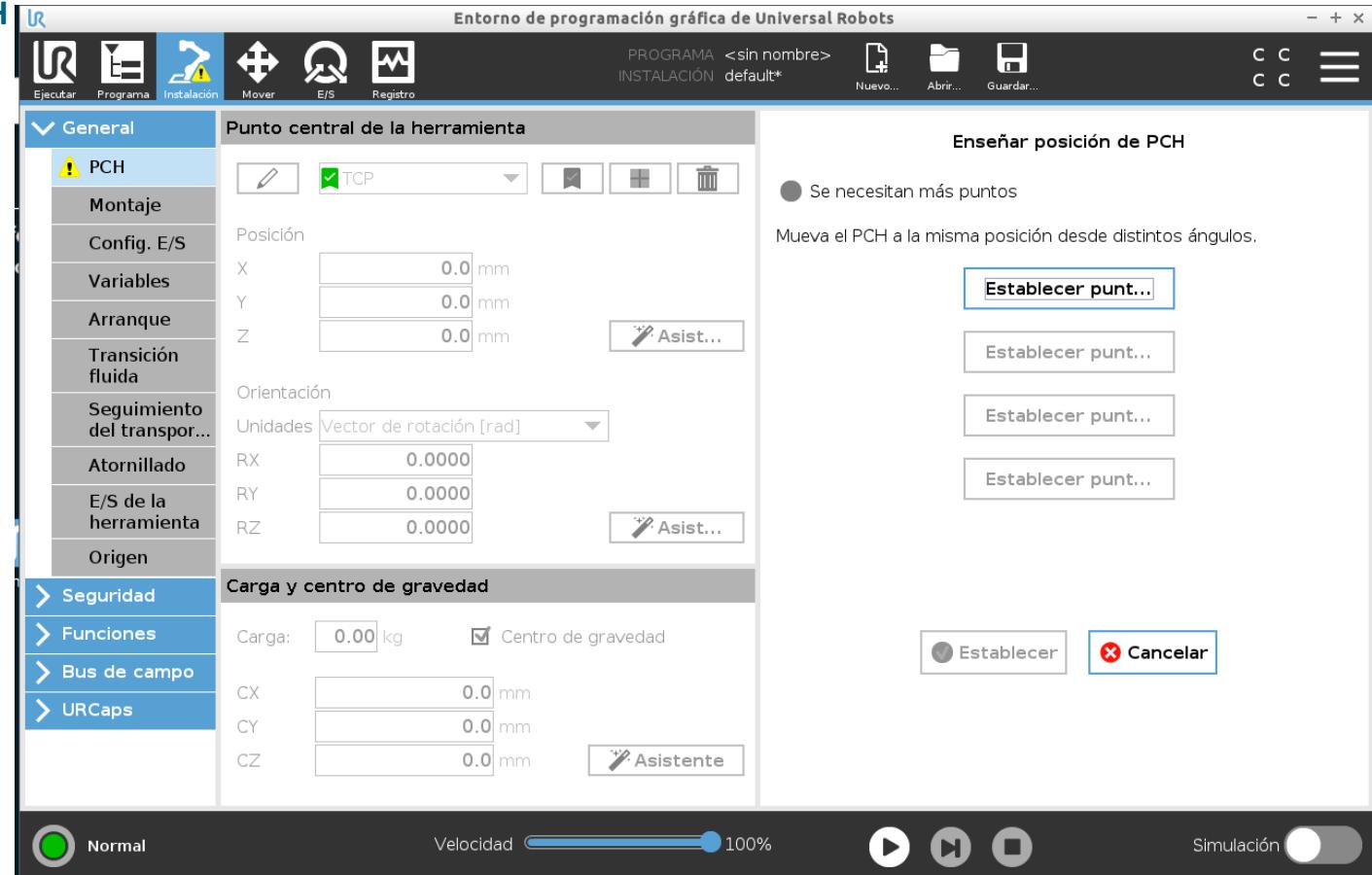
- Distancia lineal desde el centro de la brida de acople hasta la punta de herramienta
- Determina coordenadas XYZ de acuerdo a la ilustración
- Asistentes para cálculo de Posición y Orientación
- Posibilidad de definir múltiples TCP



Configuración del PCH

Cálculo Posición del TCP

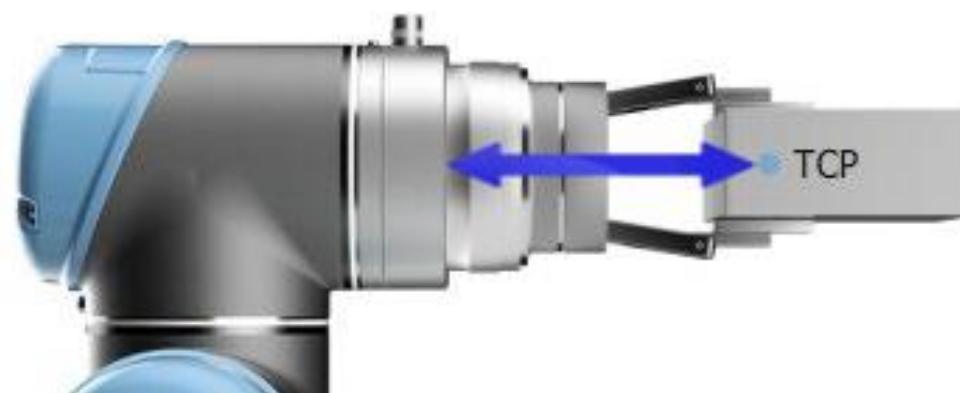
- Calcula la posición del TCP según 3-4 puntos ajustados manualmente



Configuración inicial

Configuraciones iniciales

- Configuración del Payload
- Configuración del centro de masa.



Configuración inicial

- **Configuraciones iniciales**

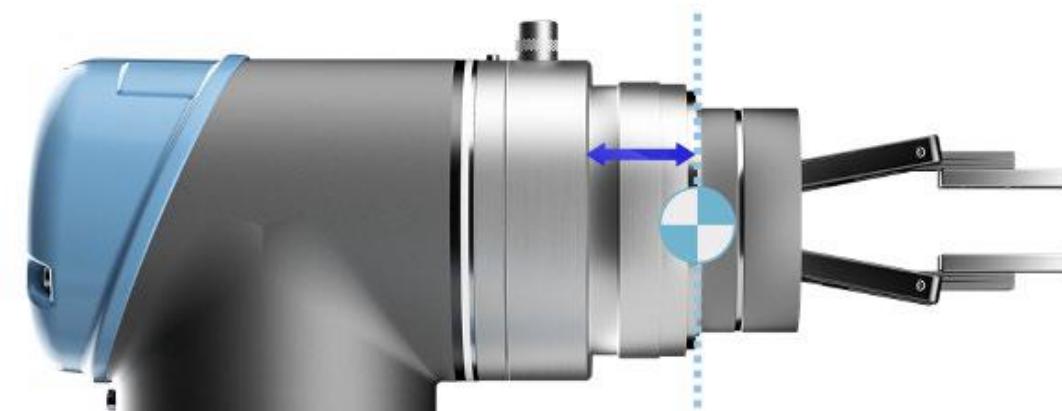
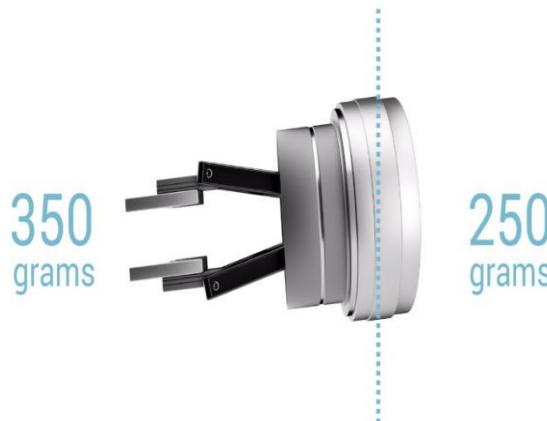
- Configuración del TCP
- Configuración Montaje (*Tool*)

Center of gravity:

CX **0.0** mm

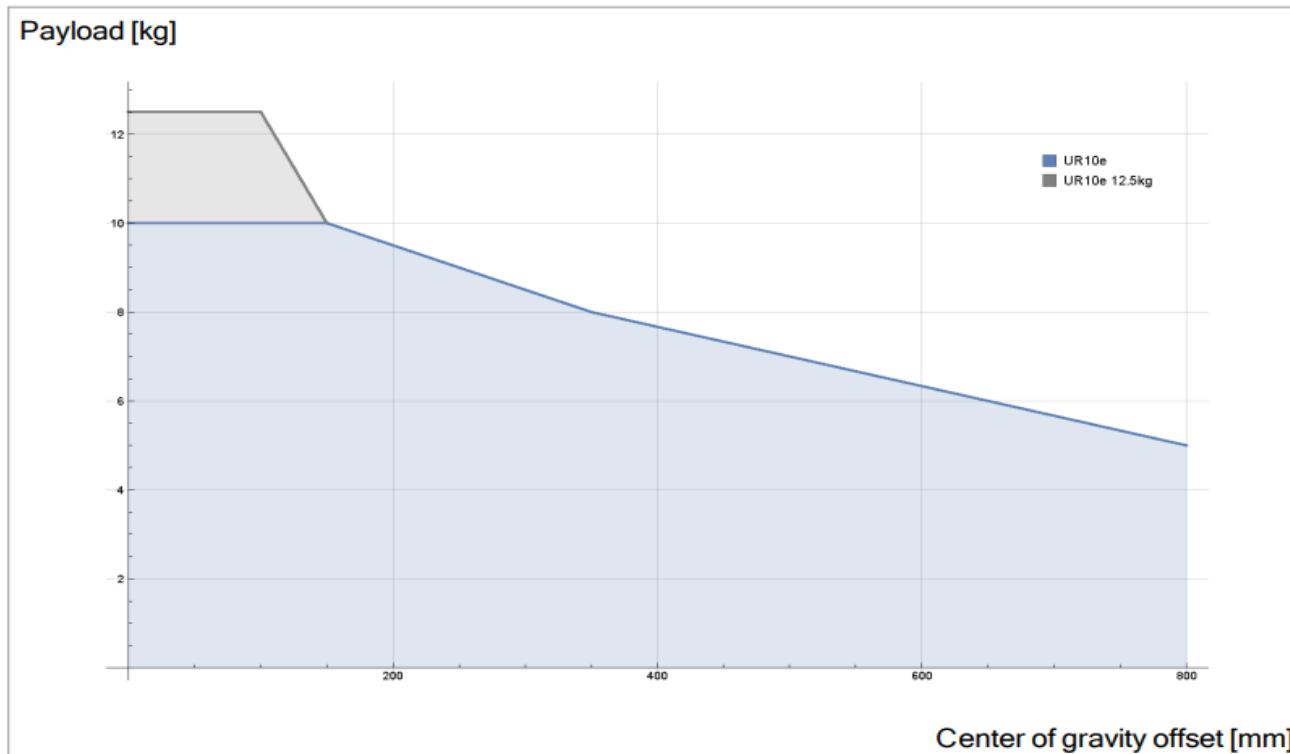
CY **0.0** mm

CZ **0.0** mm



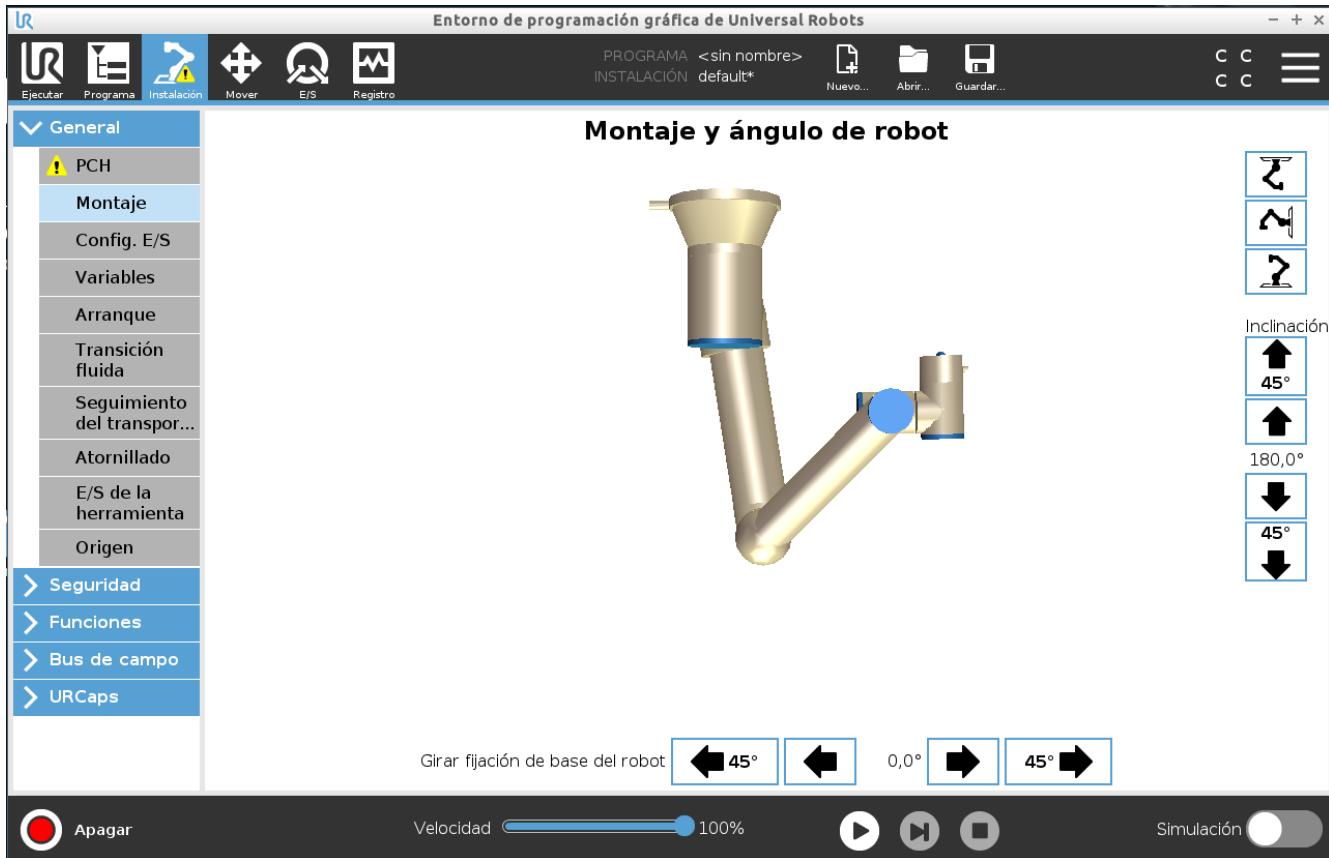
Cálculo de la Carga útil

- Ejemplo:
 - Cómo calcular la carga máxima permitida



Montaje

- Configurar cómo está montada la base del robot



Cargar/guardar

• Guardar Instalación

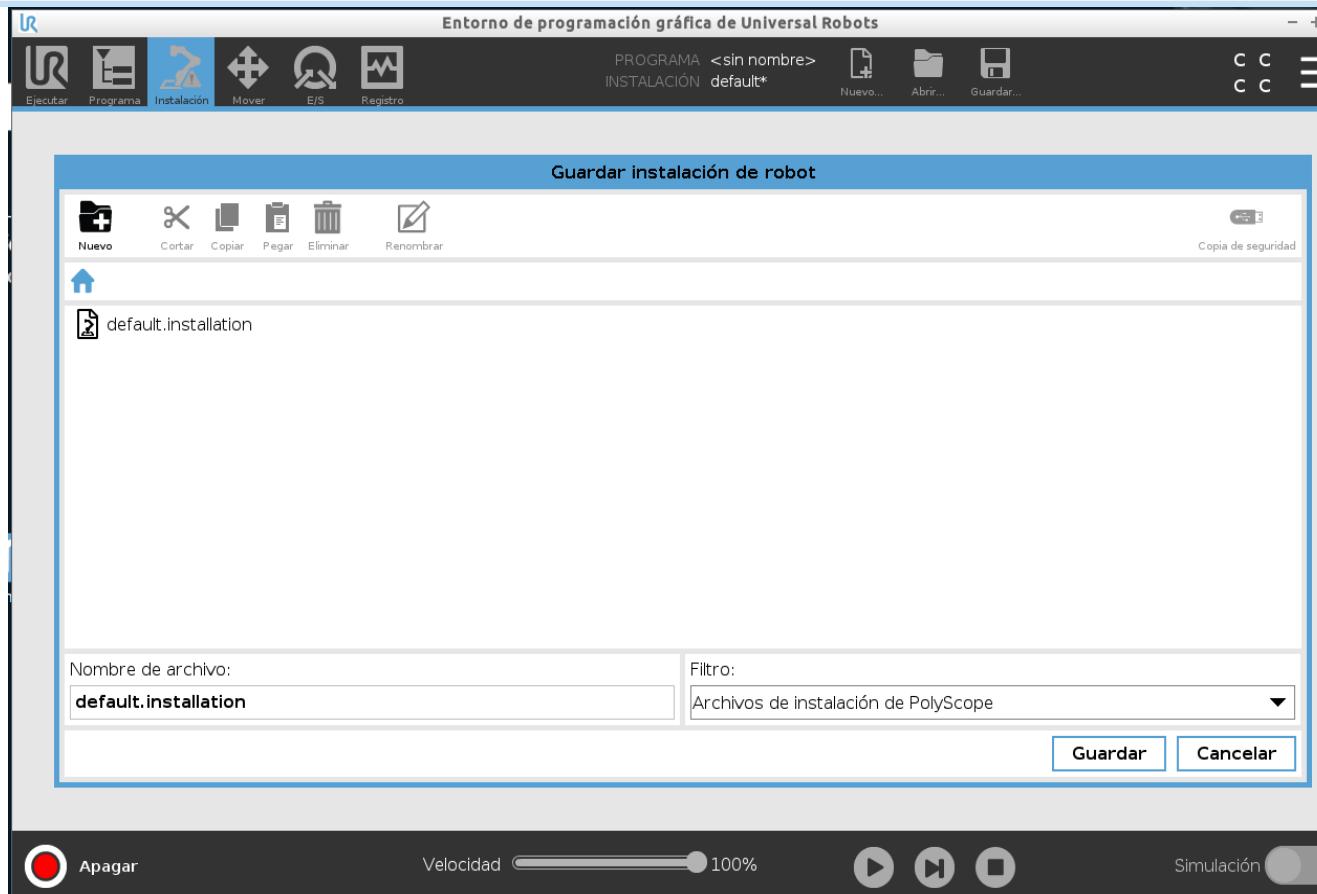
- Todas las configuraciones en Instalación se guardan en un archivo (*file*)

• Cargar Instalación

- Carga un archivo (*file*) de Instalación guardado

• Nombre por defecto

- default.installation
- Se carga cada vez que se reinicia el robot



Ejercicio Práctico 1

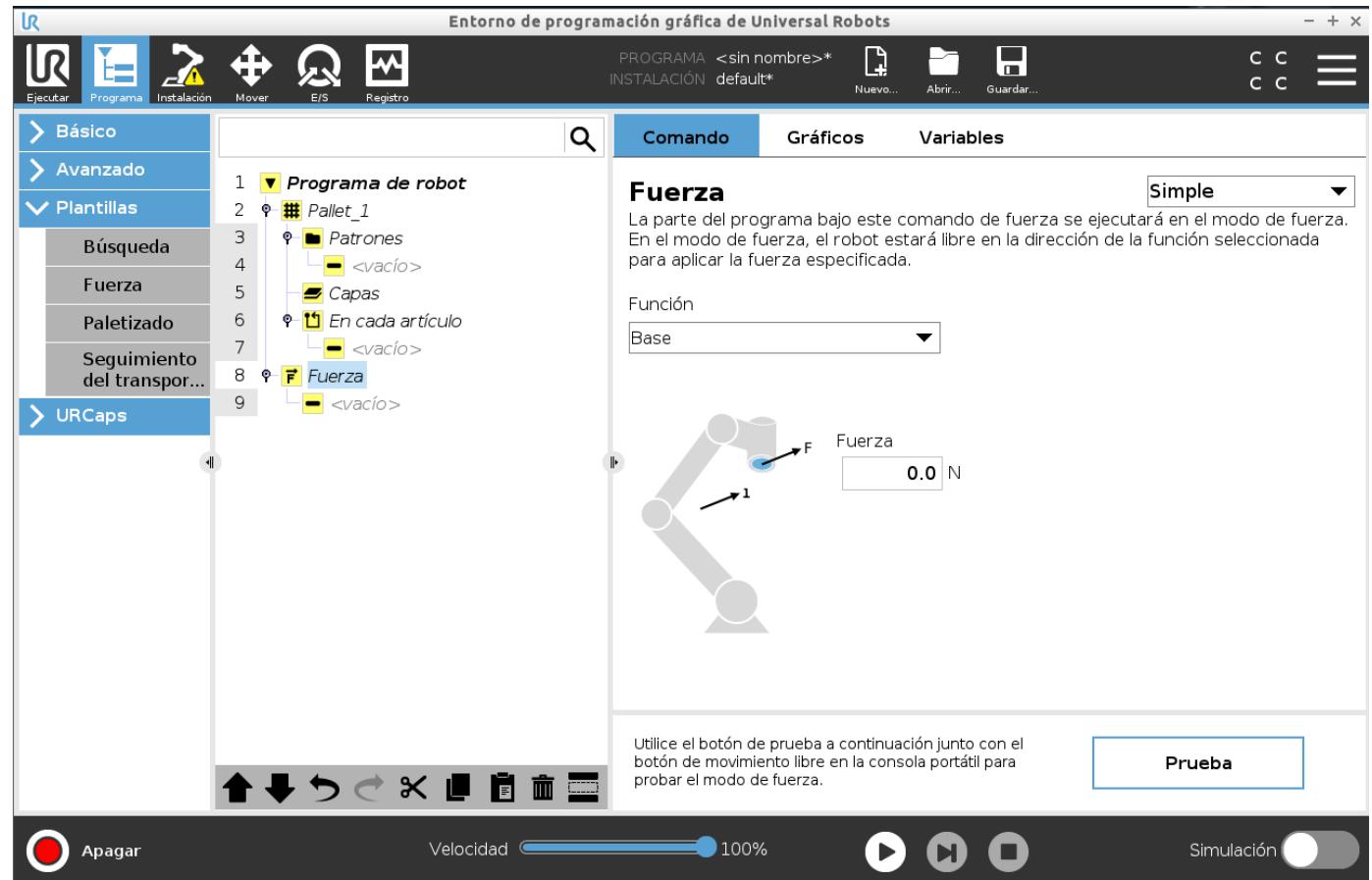
- Establecer el TCP
 - Establece el valor correcto del TPC de una herramienta
- Establecer la carga (*payload*)
 - Establece el valor correcto de carga de una herramienta
- Establecer el centro de gravedad (*CoG*)
 - Establece el nuevo centro de gravedad de la herramienta
- Probar si el robot se mueve en *freedrive* con los nuevos valores

1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes / Wizards
3	Comandos básicos 1	9	Modbus TCP / Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

Crear un programa nuevo

• Usar plantilla

- Programa nuevo
(Empty Program)



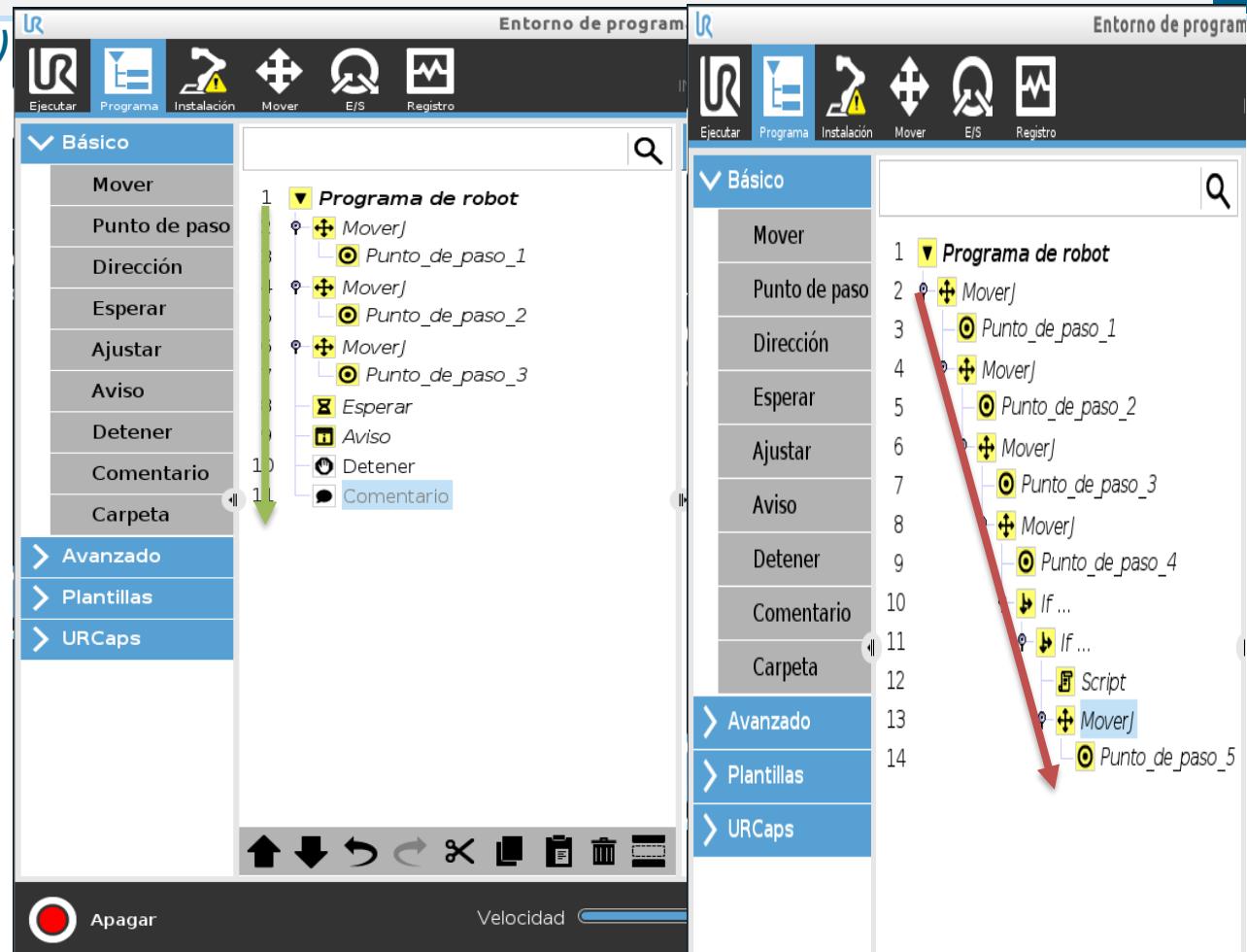
Pestaña Comando (Command)

Árbol de programa

- Estructura del programa
- Contiene todos los comandos
- Se ejecuta secuencialmente

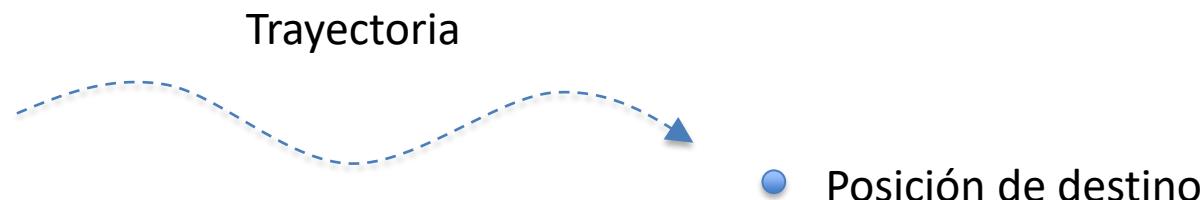
Pestaña comando

- Edición del comando seleccionado
- Configura parámetros generales del programa



Insertar movimiento

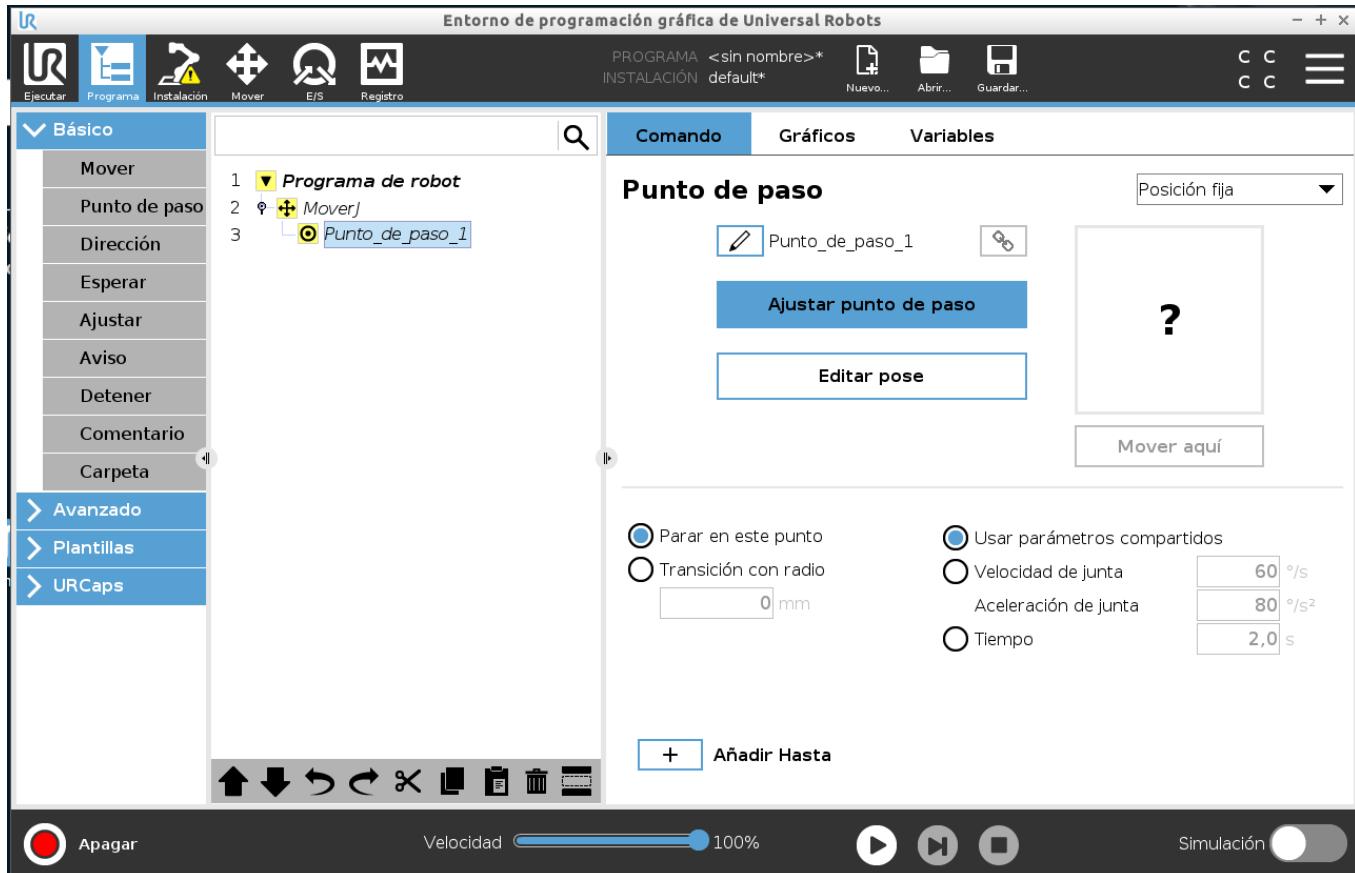
- **Waypoint (*Punto de paso*)** especifica la posición de destino
- **Move** especifica la trayectoria que seguirá el robot para llegar a la posición de destino



Comandos básicos (Basic)

• Comandos básicos

- Realizar operaciones sencillas
- Insertar comando “Move” (Movimiento)



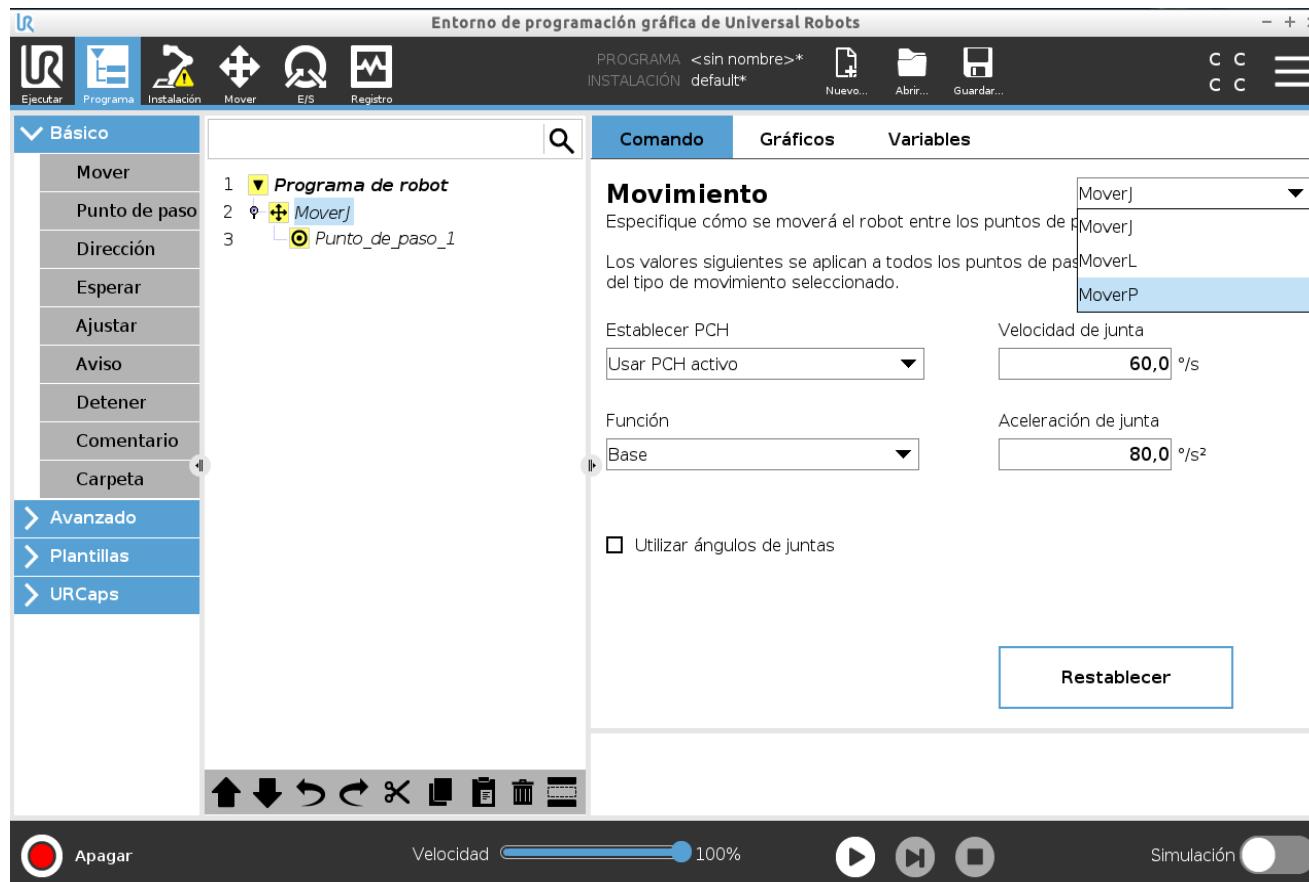
Comandos Movimiento

Tipos de movimiento

- MoveJ (*defecto*)
- MoveL
- MoveP
 - MoveC

Parámetros compartidos

- Velocidad de articulación
 - Defecto: $60^{\circ}/s$
- Aceleración de articulación
 - Defecto: $80^{\circ}/s^2$
- Se añade automáticamente un Waypoint cuando se inserta un comando de Movimiento



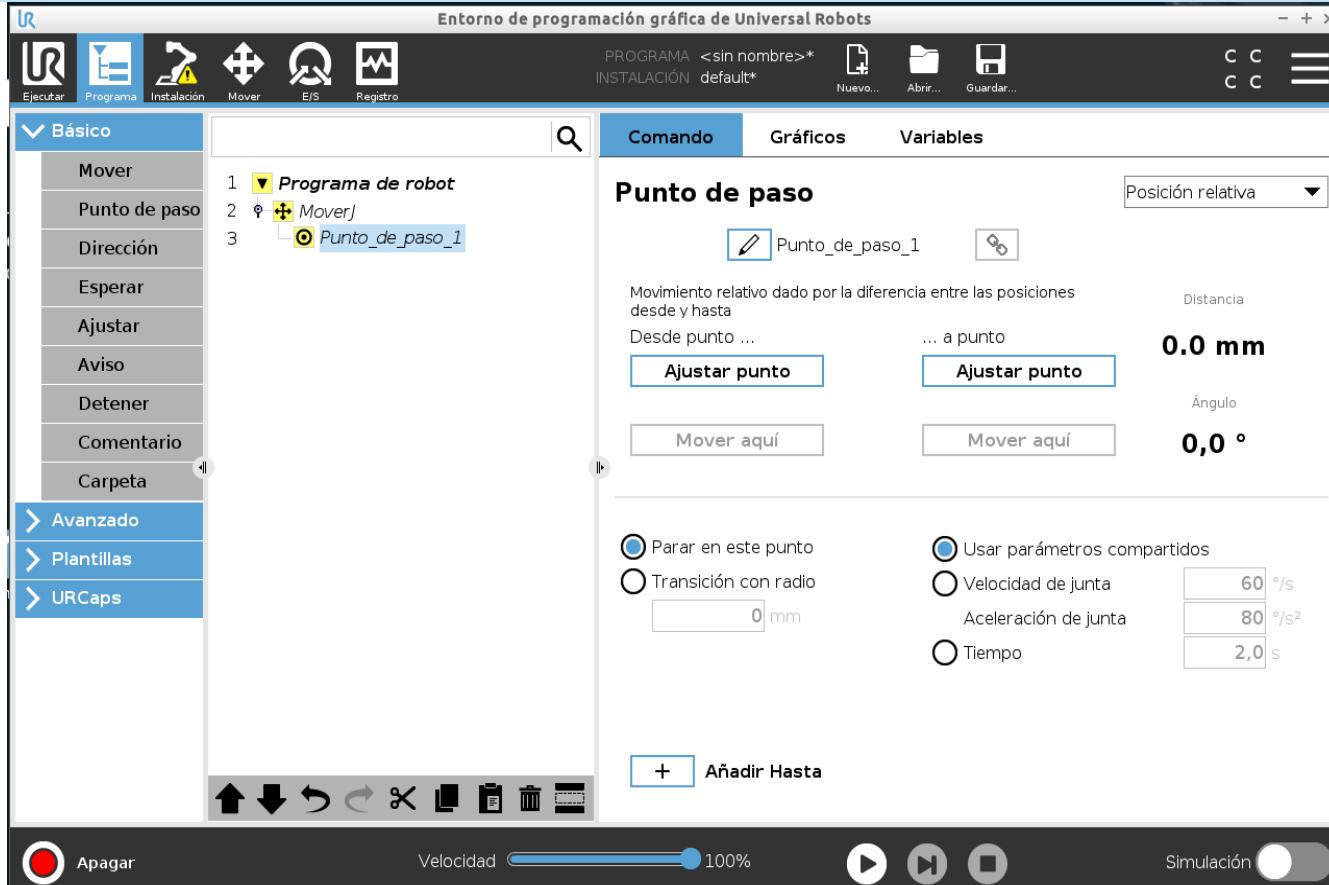
Waypoint (*Punto_de_paso*)

• Tipos de Waypoint

- Fijo (*defecto*)
- Relativo
- Variable

• Enseñar Waypoint

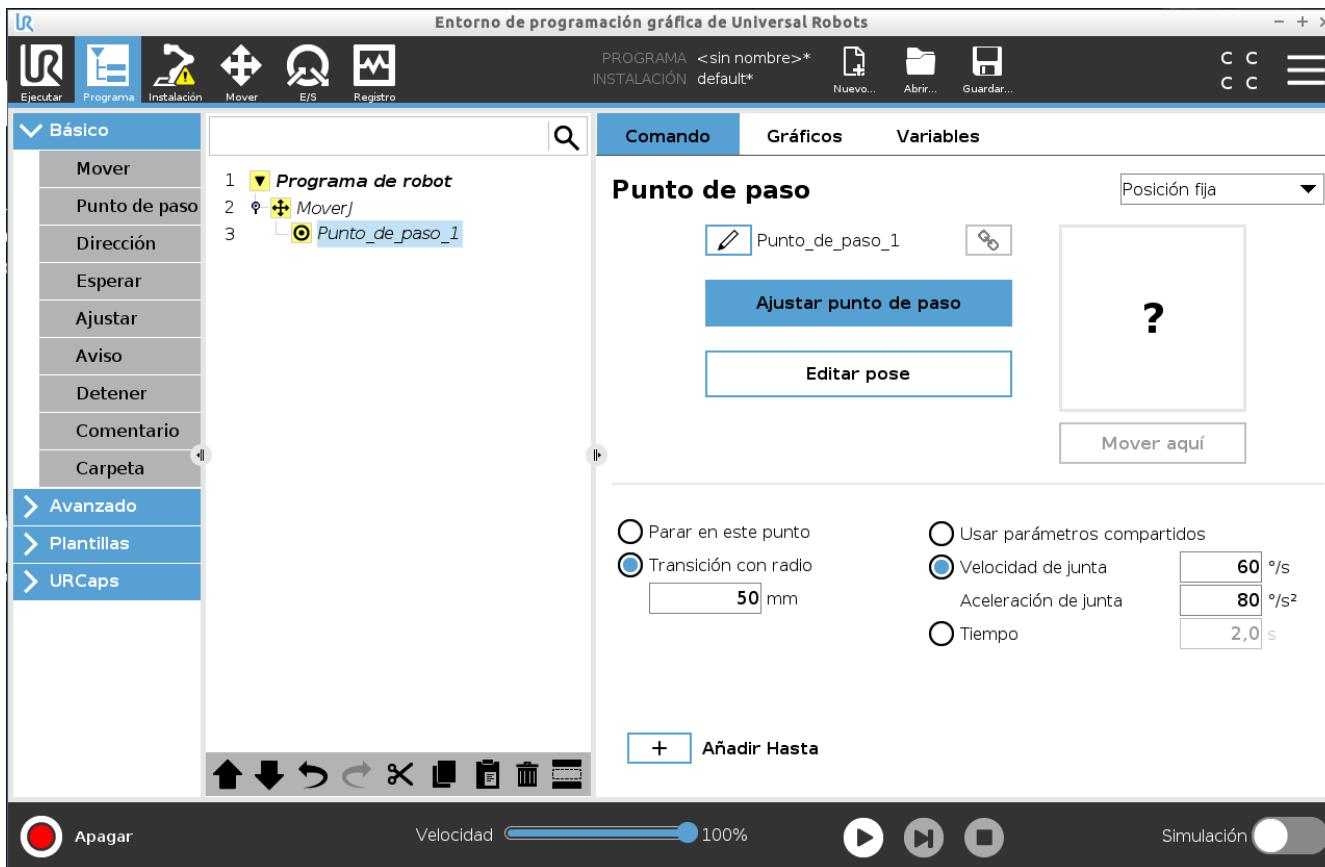
- Pulsar “Set this waypoint”
(*Fijar este punto de paso*)
- Enseñar la posición deseada y pulsar OK para grabarla



Waypoint (*Punto_de_paso*)

Opciones

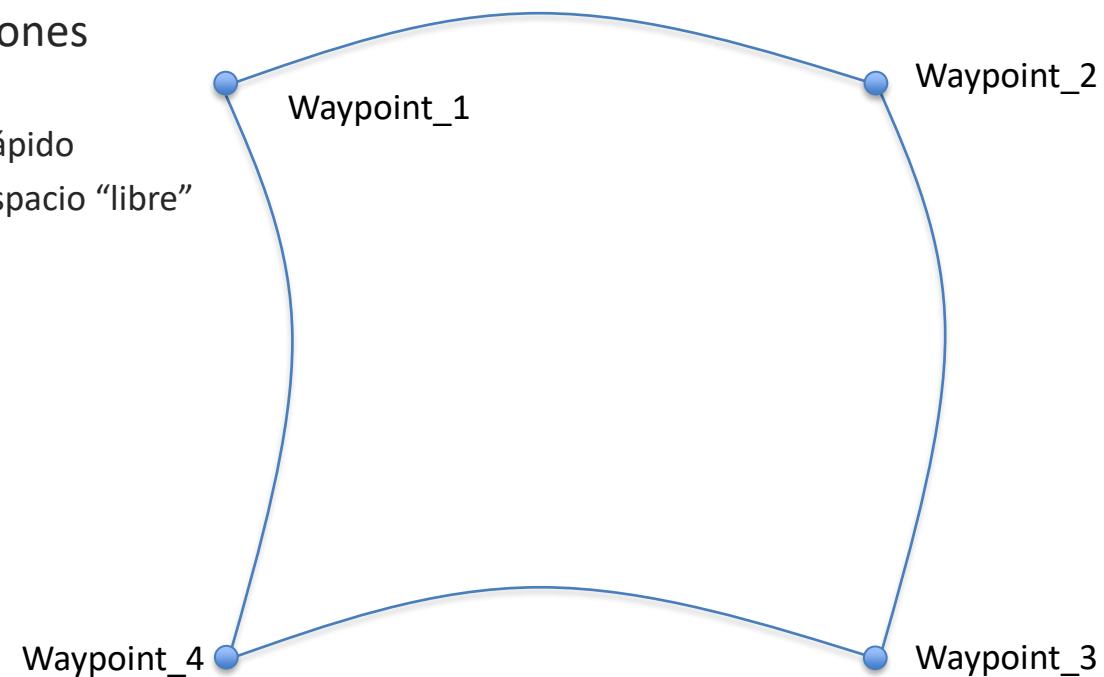
- Rename
(Renombrar)
 - Mejora la legibilidad
- Move robot here
(Mover robot aquí)
 - Mueve a la posición
- Change this
Waypoint (Cambiar punto de paso)
 - Modifica posición



MoveJ

- Movimiento de articulaciones

- Sin interpolación
- Tipo de movimiento más rápido
- Útil en movimientos con espacio “libre”



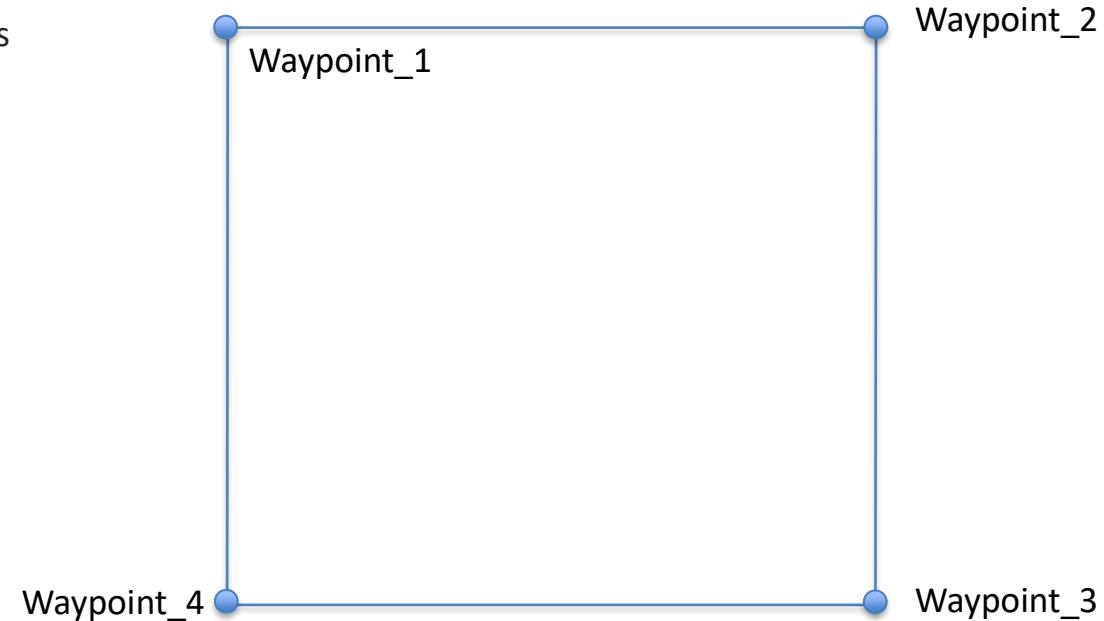
Robot Program

```
MoveJ  
    Waypoint_1  
    Waypoint_2  
    Waypoint_3  
    Waypoint_4
```

- Guardar programa de ejemplo como movej.urp

MoveL

- Movimiento lineal
 - Interpolación de posiciones
 - Trayectoria lineal del TCP



Robot Program

```
MoveL  
    Waypoint_1  
    Waypoint_2  
    Waypoint_3  
    Waypoint_4
```

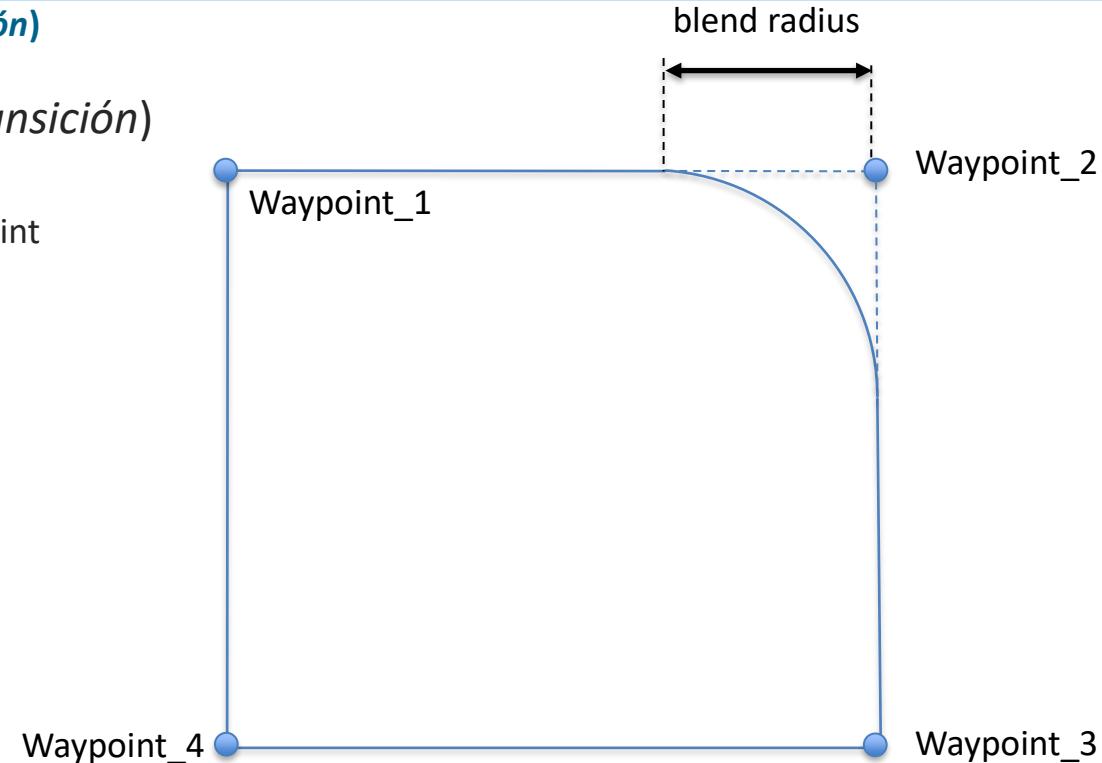
- Guardar programa de ejemplo como movel.urp

MoveL con blend (*radio de transición*)

- Blend radius (*radio de transición*)
 - Movimiento Continuo
 - No se detiene en el Waypoint

Robot Program

```
MoveL  
  Waypoint_1  
  Waypoint_2 (r=25)  
  Waypoint_3  
  Waypoint_4
```

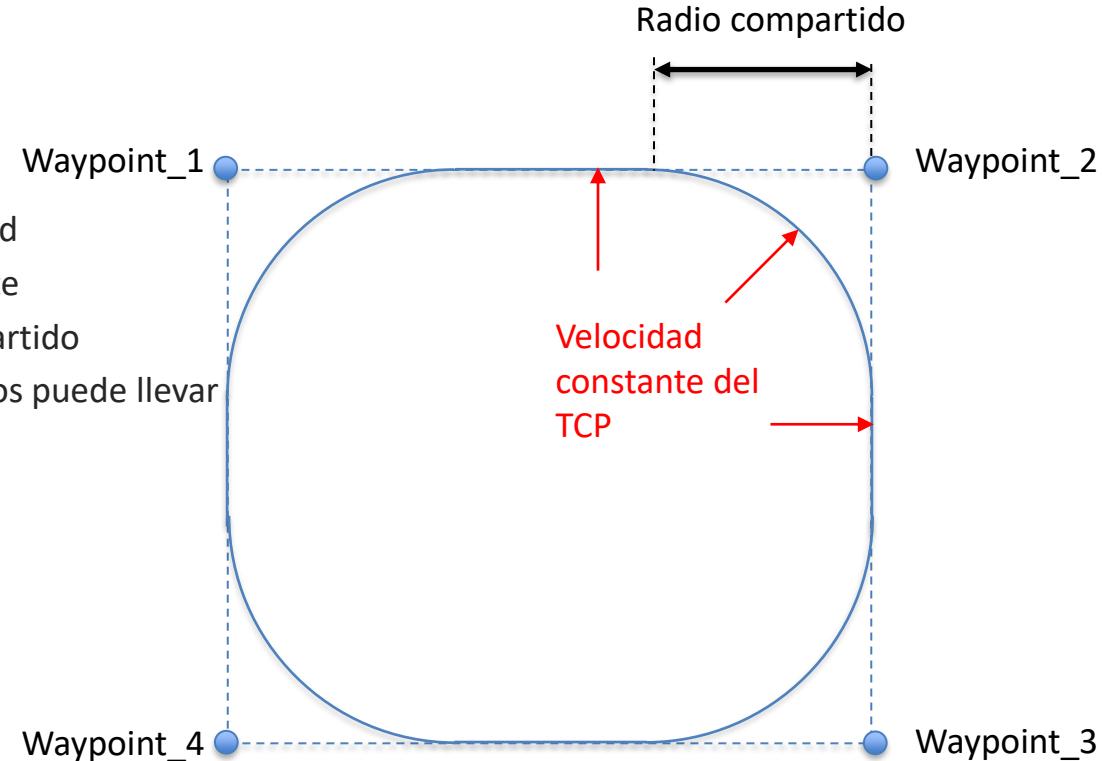


- Guardar programa de ejemplo como move_with_blend.urp

MoveP

- Movimiento de Proceso

- Aplicaciones de proceso
- Movimiento lineal con bend
- Velocidad del TCP constante
- Transición con radio compartido
- Alta velocidad y bajos radios puede llevar a parada.

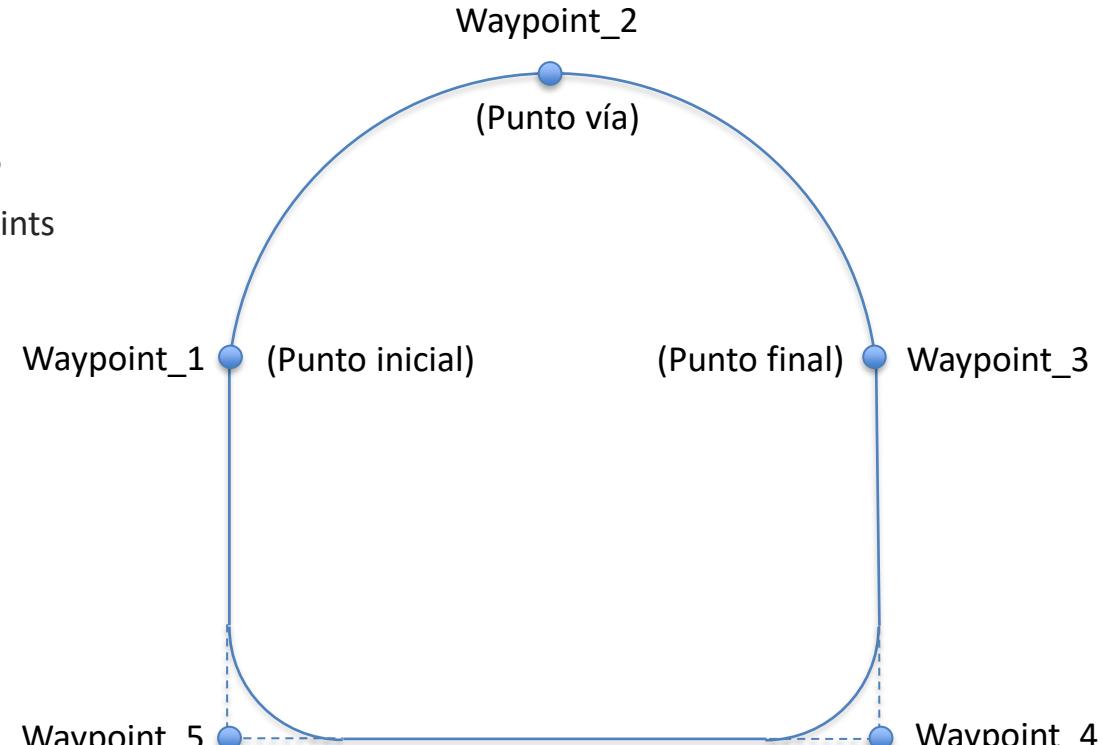


Robot Program

```
MoveP  
  Waypoint_1  
  Waypoint_2  
  Waypoint_3  
  Waypoint_4
```

MoveC

- Movimiento Circular
 - Trayectoria circular del TCP
 - Son necesarios tres Waypoints
 - Punto inicial
 - Punto vía
 - Punto final
 - El radio se calcula de forma automática



Robot Program

```
MoveP  
  Waypoint_1  
  CircleMove  
    Waypoint_2  
    Waypoint_3  
  Waypoint_4  
  Waypoint_5
```

- Guardar programa de ejemplo como movec.urp

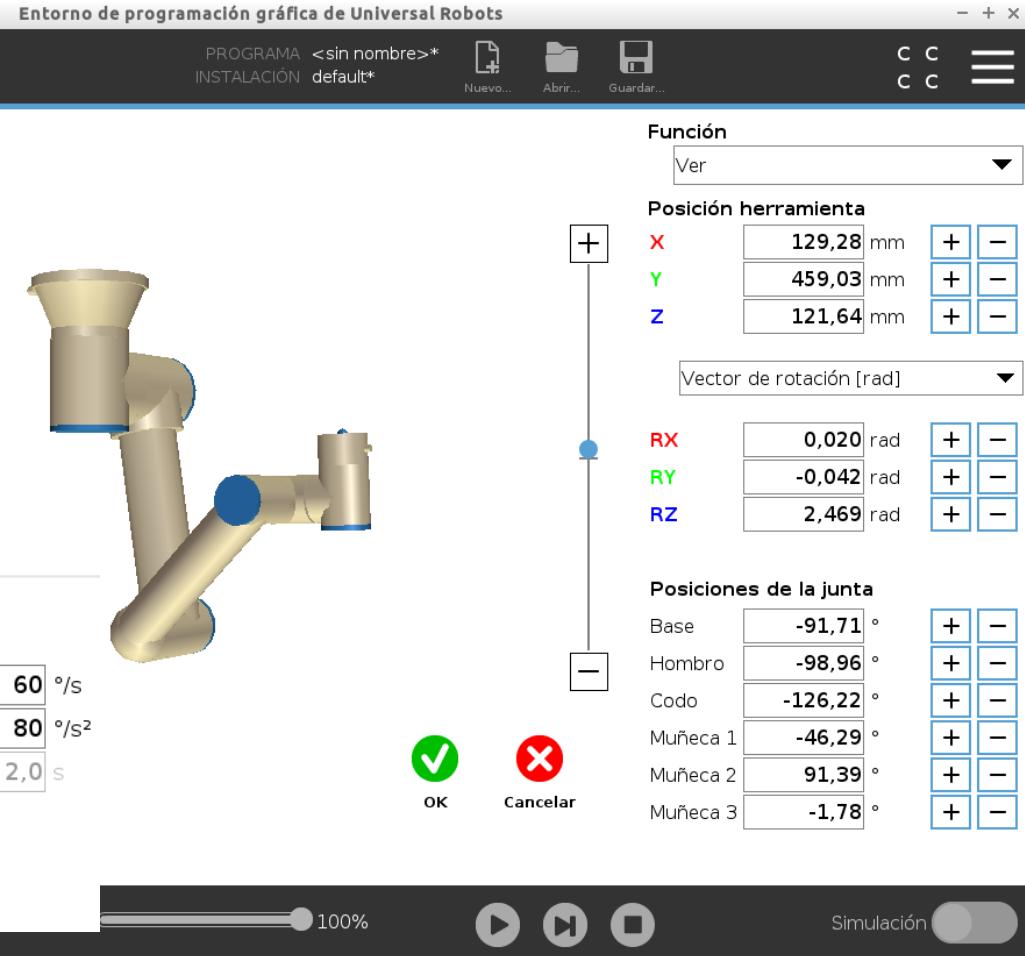
Opciones avanzadas

- Opciones avanzadas
 - Configuración individual
 - Velocidad & Aceleración
 - Tiempo
 - Editar pose
 - Modificar los valores de la pose manualmente

Parar en este punto
 Transición con radio
 50 mm

Usar parámetros compartidos
 Velocidad de junta
 Aceleración de junta
 Tiempo

60	°/s
80	°/s ²
2,0	s



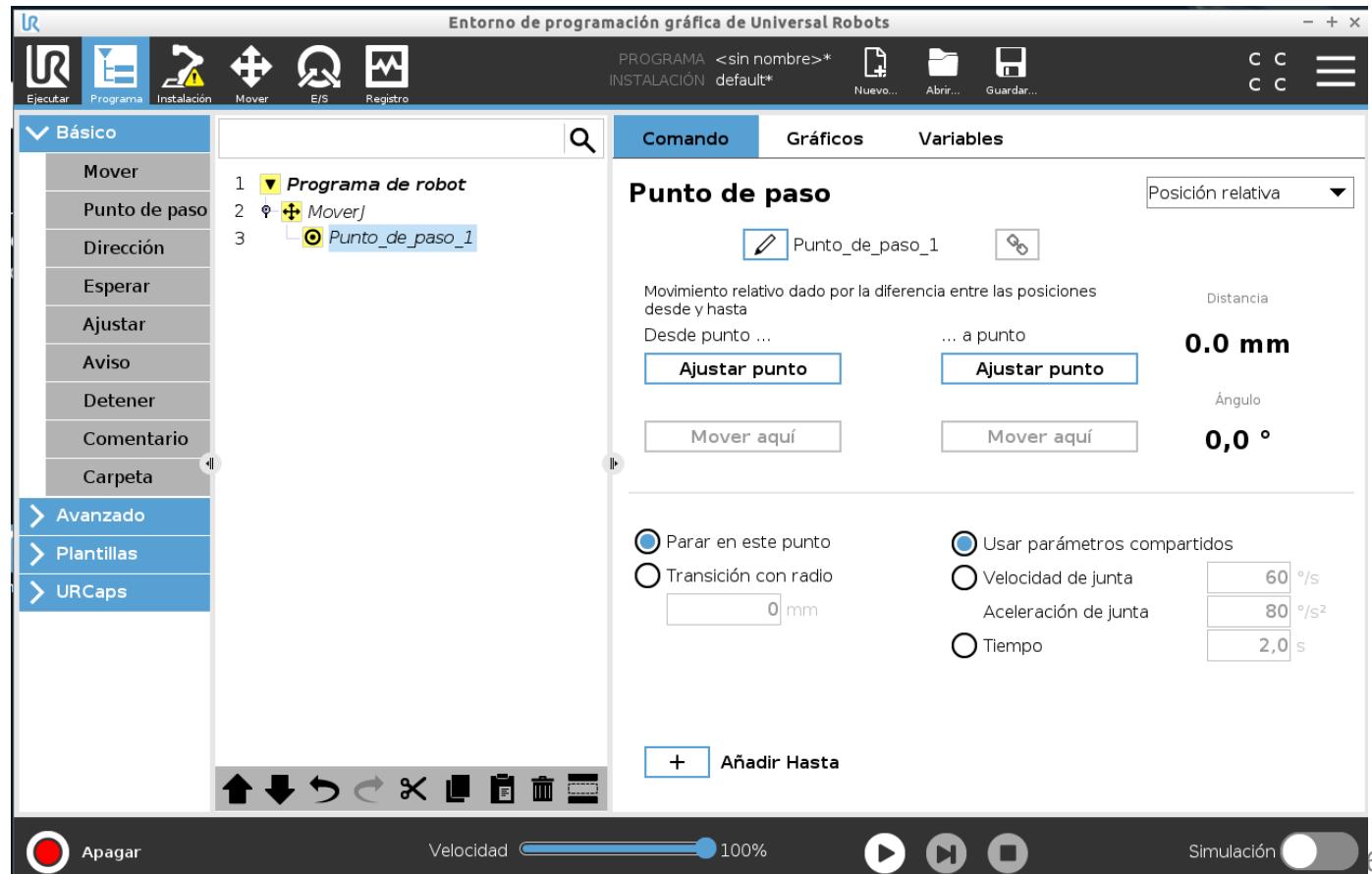
+ Añadir Hasta

Simulación

Relative waypoint (*Punto_de_paso relativo*)

Movimiento relativo

- Movimiento lineal
- Relativo a la posición previa en el programa
- Se muestran distancia y ángulo



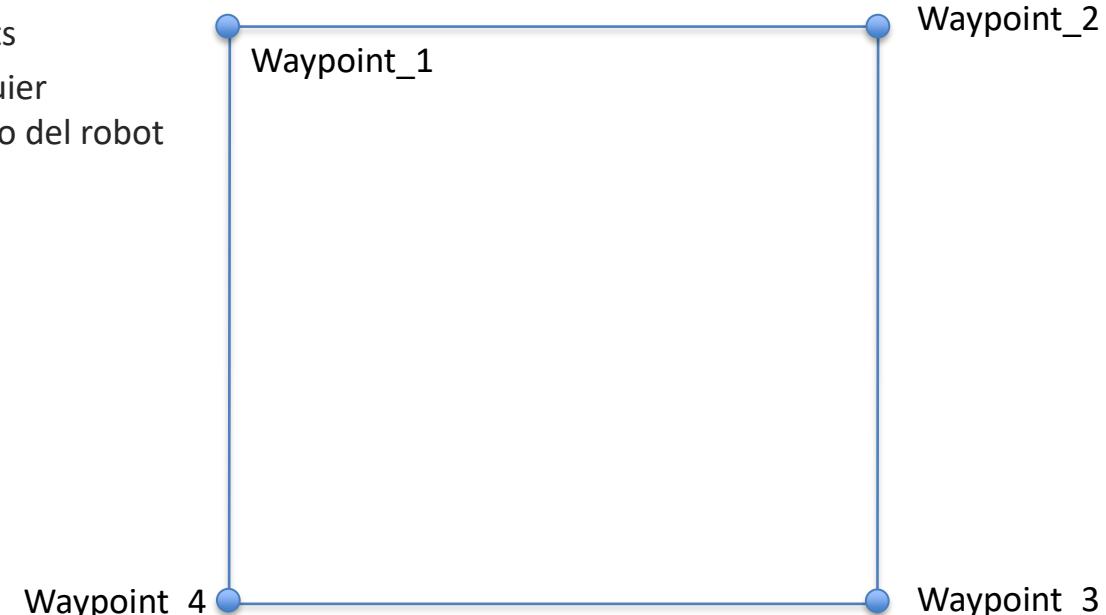
Relative waypoint (*Punto_de_paso relativo*)

- Definir waypoints relativos

- Definido por dos Waypoints
- Pueden definirse en cualquier posición del área de trabajo del robot

Robot Program

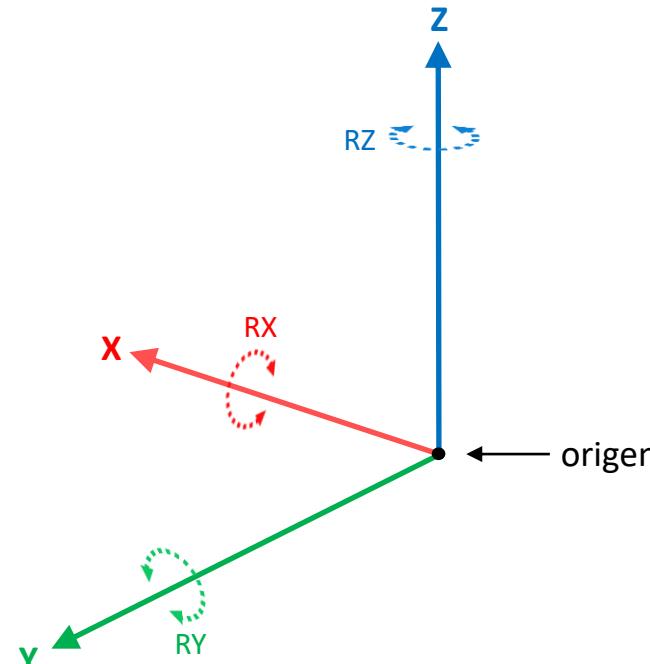
```
MoveL  
    Waypoint_1  
    Waypoint_2 rel. pos  
    Waypoint_3 rel. pos  
    Waypoint_4 rel. pos
```



- Guardar programa de ejemplo como movel_with_relative_waypoint.urp

¿Qué es una Feature (Función)?

- Los “Features” son sistemas de coordenadas cartesianos
 - Un sistema de coordenadas cartesiano define un plano o espacio mediante ejes desde un punto fijo denominado origen



- Las funciones son guardadas como variables tipo Pose

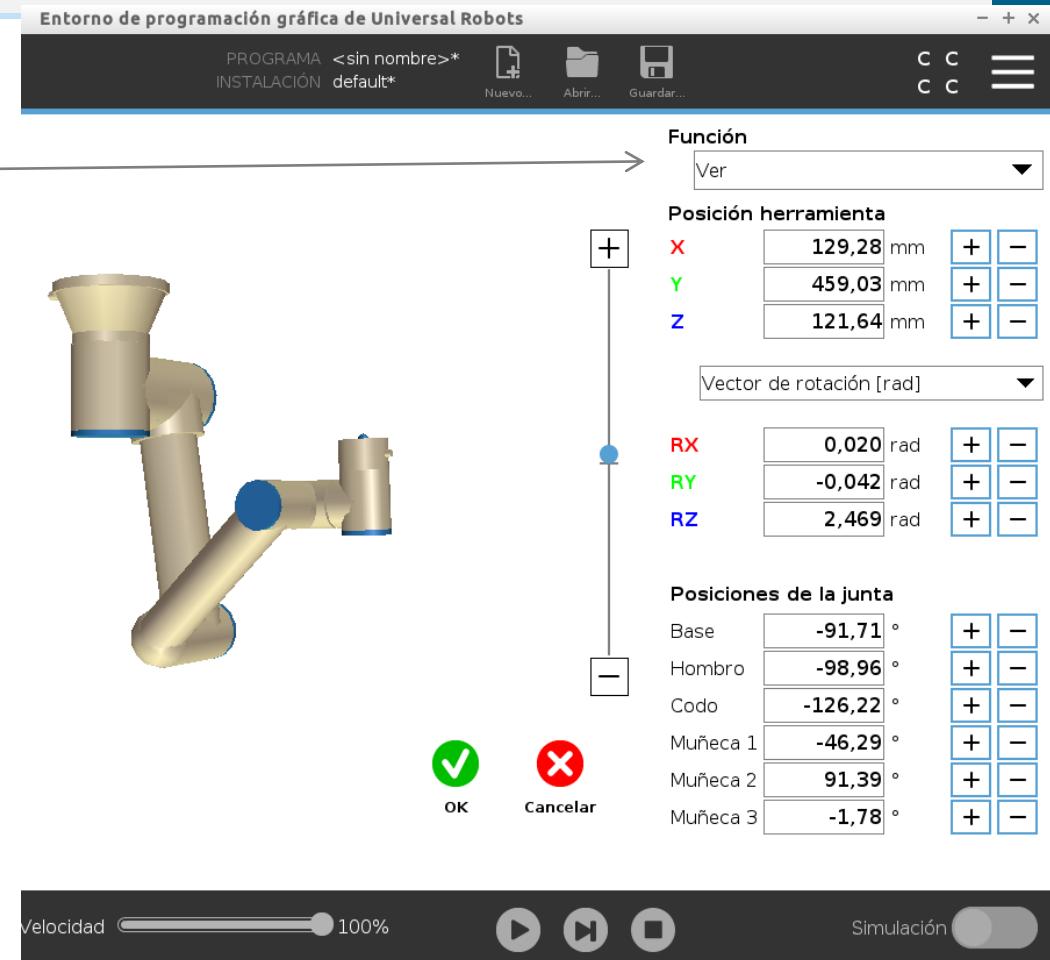
¿Qué es una Feature (Función)?

- El robot opera de forma relativa a “Features”

- Base Feature
- Tool Feature
- (Herramienta)

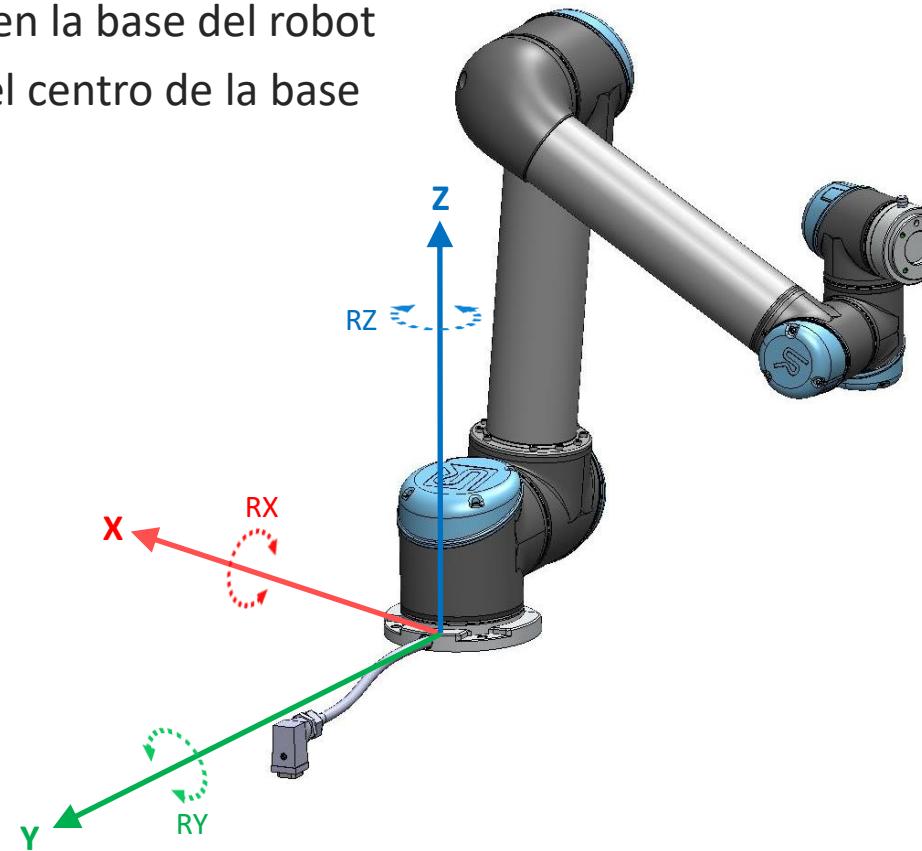
- Posición de la Herramienta

- X, Y, Z = posición del TCP
 - Unidad: mm / pulgadas
- Rx, Ry, Rz = orientación
 - Unidades: radianes
- NOTA: La orientación es dada por un vector de rotación, donde la longitud del vector es el ángulo a rotar en grados



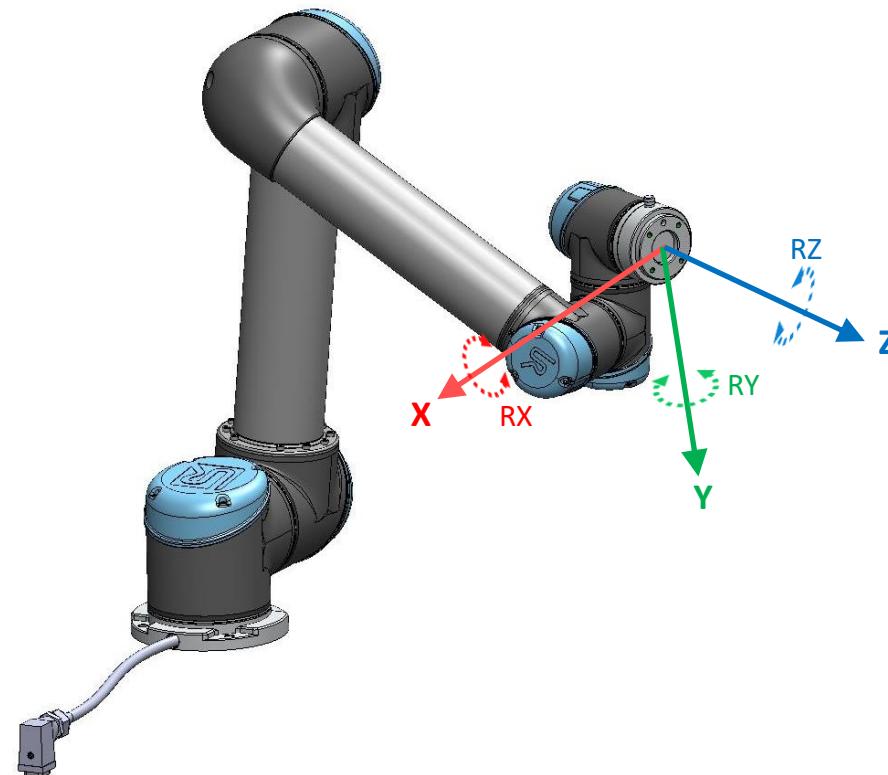
Base feature

- Localizada en la base del robot
- Origen en el centro de la base



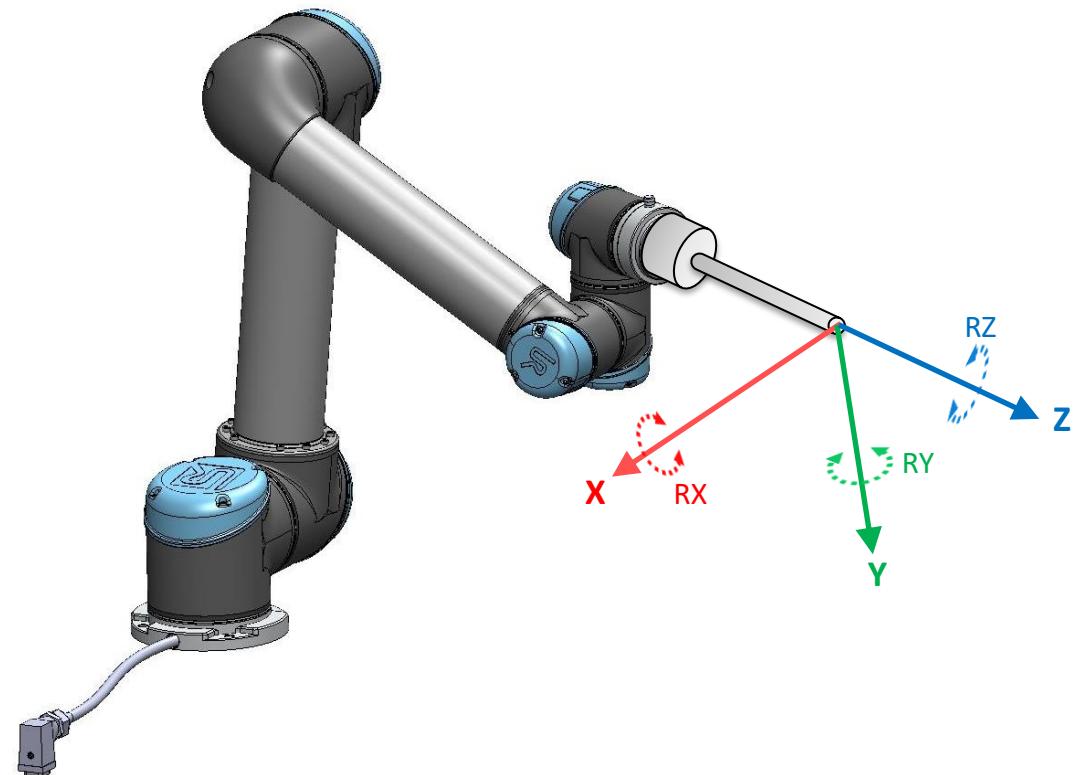
Tool feature (*Herramienta*)

- Origen en el centro de la brida de la herramienta



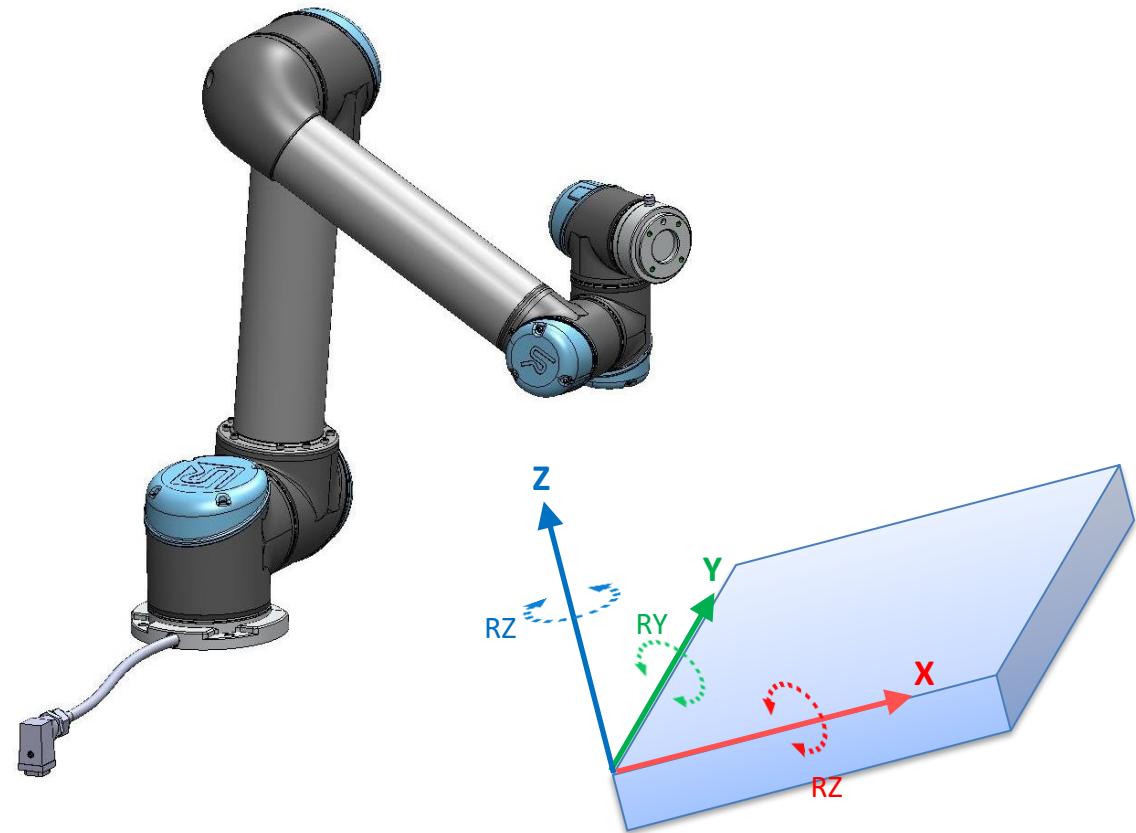
Tool feature (*Herramienta*)

- Origen en el TCP



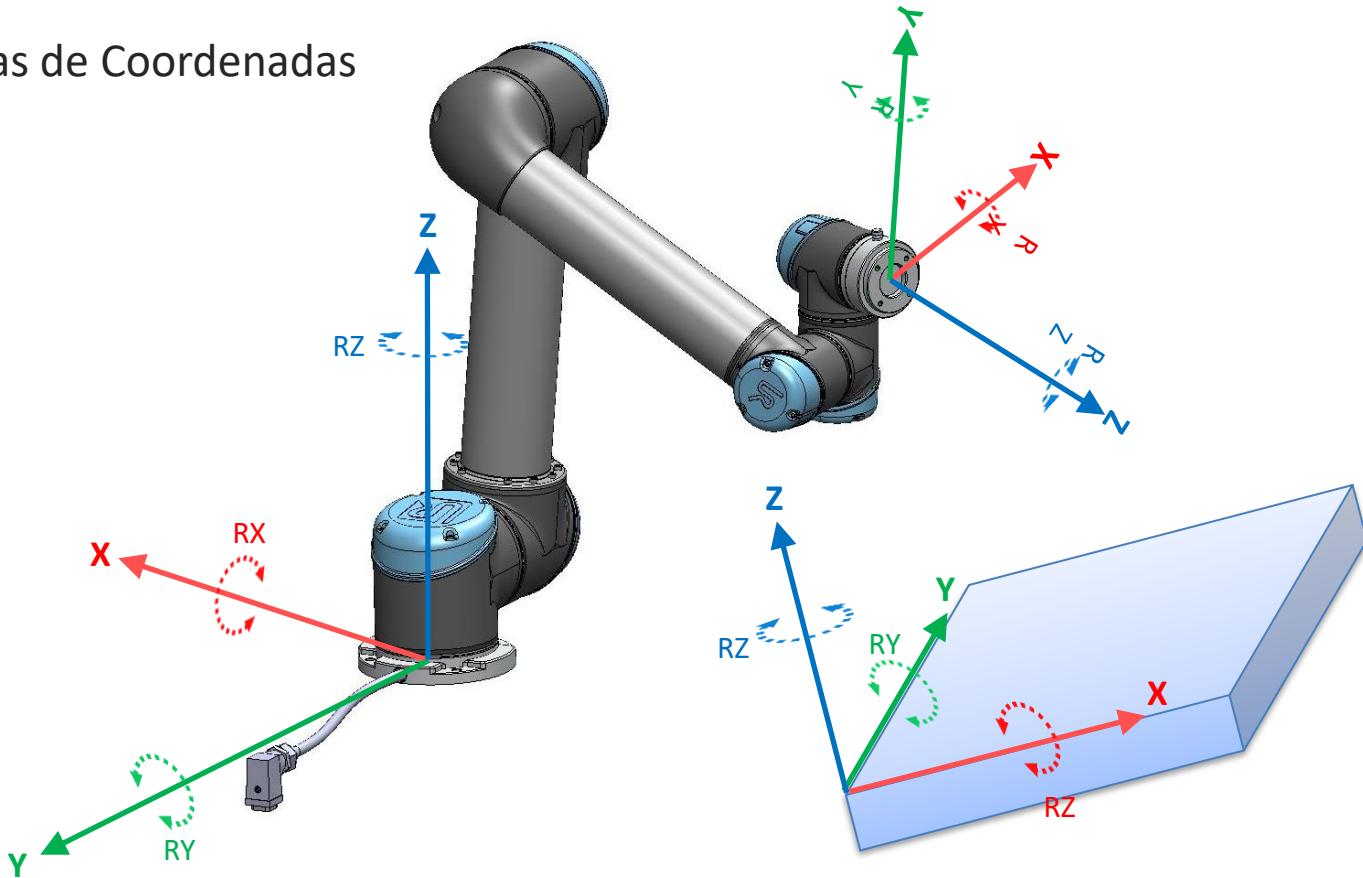
Feature definida por el usuario

- Definida por el operador



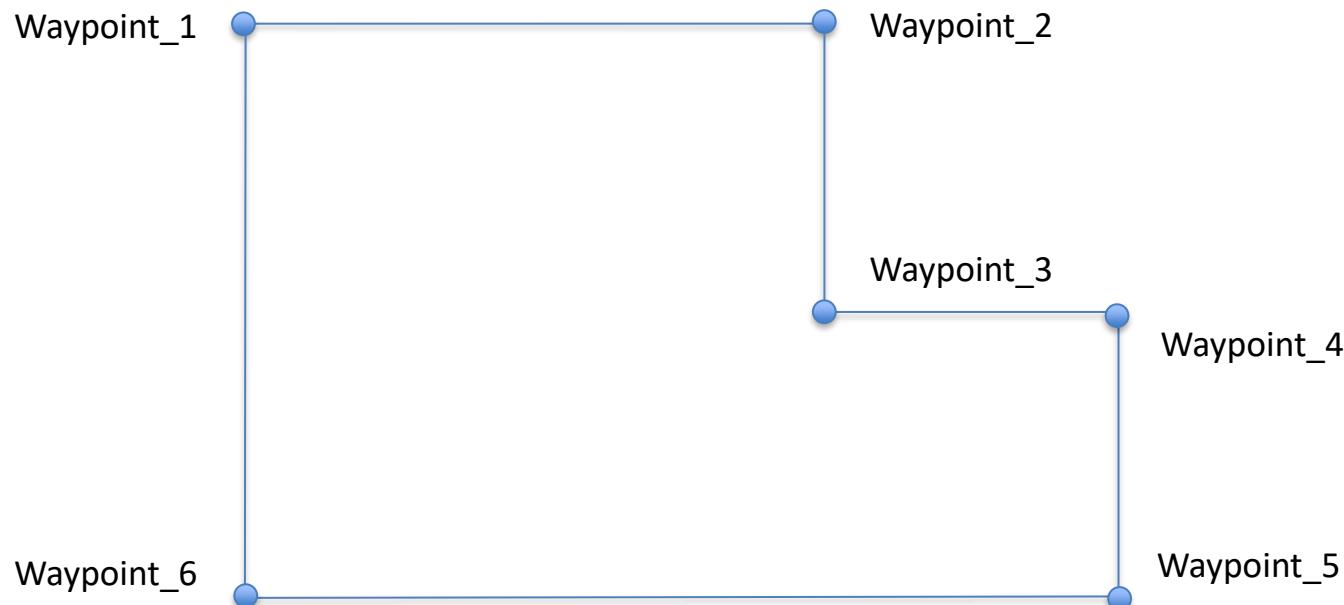
Features

- Los 3 Sistemas de Coordenadas



Ejercicio práctico parte 1

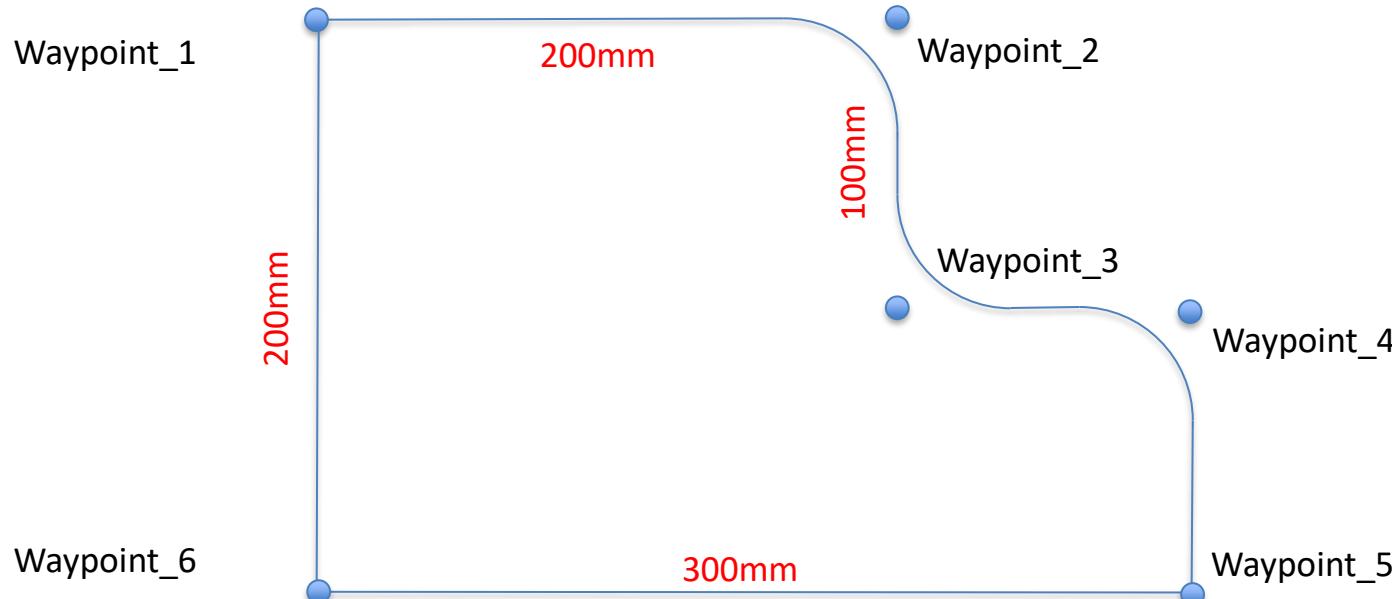
- Insertar instrucciones MoveL
 - Crear 6 puntos de paso en posiciones como en el diagrama
 - Usar la pestaña “Move” para llevar la herramienta del robot a las posiciones adecuadas



- Guardar programa de ejemplo como Lab_exe_3_1.urp

Ejercicio práctico parte 2

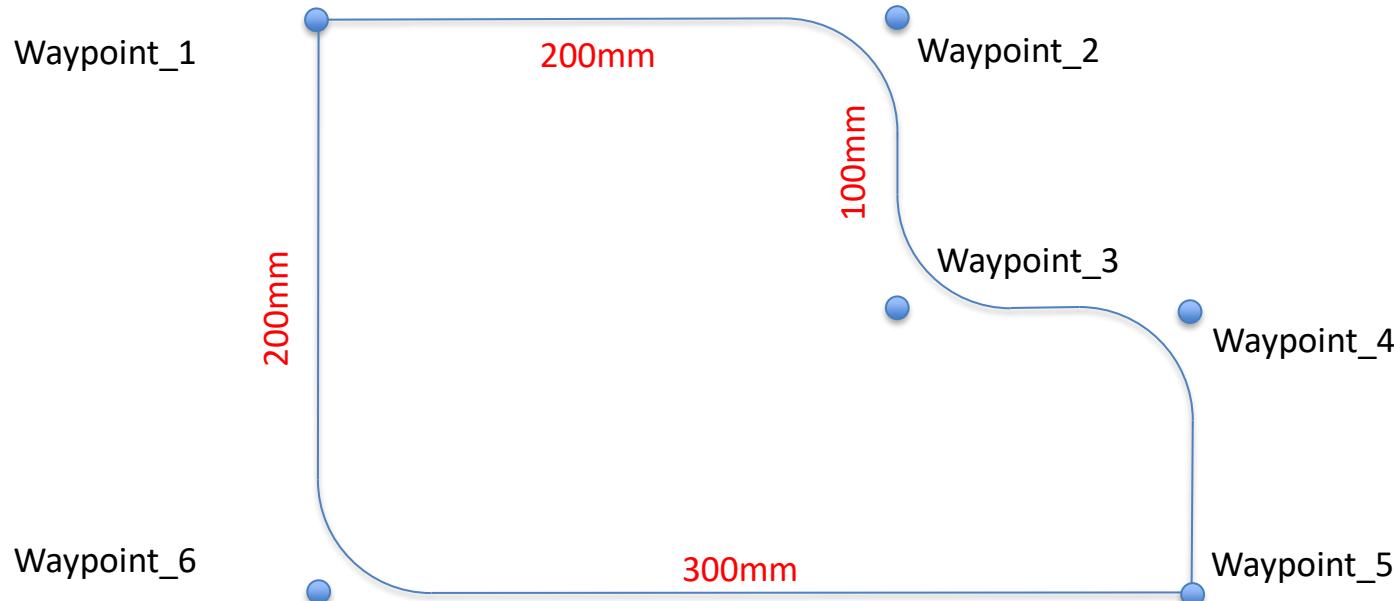
- Ajustar los Waypoints
 - Ajustar los Waypoints de acuerdo a las dimensiones del diagrama usando “Edit pose”
 - Añadir una transición con radio (*blend*) de 50mm a los Waypoints 2, 3 and 4



- Guardar programa de ejemplo como Lab_exe_3_2.urp

Ejercicio práctico parte 3

- Convertir a MoveP
 - Cambiar de MoveL a MoveP
 - Ajustar la transición con radio (*blend*) a 50mm para todos los puntos



- Guardar programa de ejemplo como Lab_exe_3_3.urp

1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes / Wizards
3	Comandos básicos 1	9	Modbus TCP / Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

Manejo de señales

- Manejo de señales
 - Interacción con dispositivos externos



Interfaz eléctrica

- Placa de Control de Seguridad - *Safety Control Board (SCB)*

- Seguridad
 - Parada de emergencia
 - Parada de protección
- Remoto
 - Encender/Apagar
- Alimentación
 - 2A PSU interna
 - PSU externa
- I/O configurables
 - 8 entradas
 - 8 salidas
 - 24V DC, PNP
- I/O digitales
 - 8 entradas
 - 8 salidas
 - 24V DC, PNP
- I/O analógicas
 - 2 entradas
 - 2 salidas
 - 0-10V DC
 - 4-20 mA

Safety	
Emergency Stop	24V
	EI0
Safeguard Stop	24V
	EI1
Emergency Stop	24V
	SI0
Safeguard Stop	24V
	SI1

Remote	
12V	■
GND	■
ON	■
OFF	■

Power	
PWR	■
GND	■

Configurable Inputs	
24V	■
CI0	■
24V	■
CI1	■
24V	■
CI2	■
24V	■
CI3	■
24V	■
CI4	■
24V	■
CI5	■
24V	■
CI6	■
24V	■
CI7	■

Configurable Outputs	
0V	■
CO0	■
0V	■
CO1	■
0V	■
CO2	■
0V	■
CO3	■
0V	■
CO4	■
0V	■
CO5	■
0V	■
CO6	■
0V	■
CO7	■

Digital Inputs	
24V	■
DI0	■
24V	■
DI1	■
24V	■
DI2	■
24V	■
DI3	■
24V	■
DI4	■
24V	■
DI5	■
24V	■
DI6	■
24V	■
DI7	■

Digital Outputs	
0V	■
DO0	■
0V	■
DO1	■
0V	■
DO2	■
0V	■
DO3	■
0V	■
DO4	■
0V	■
DO5	■
0V	■
DO6	■
0V	■
DO7	■

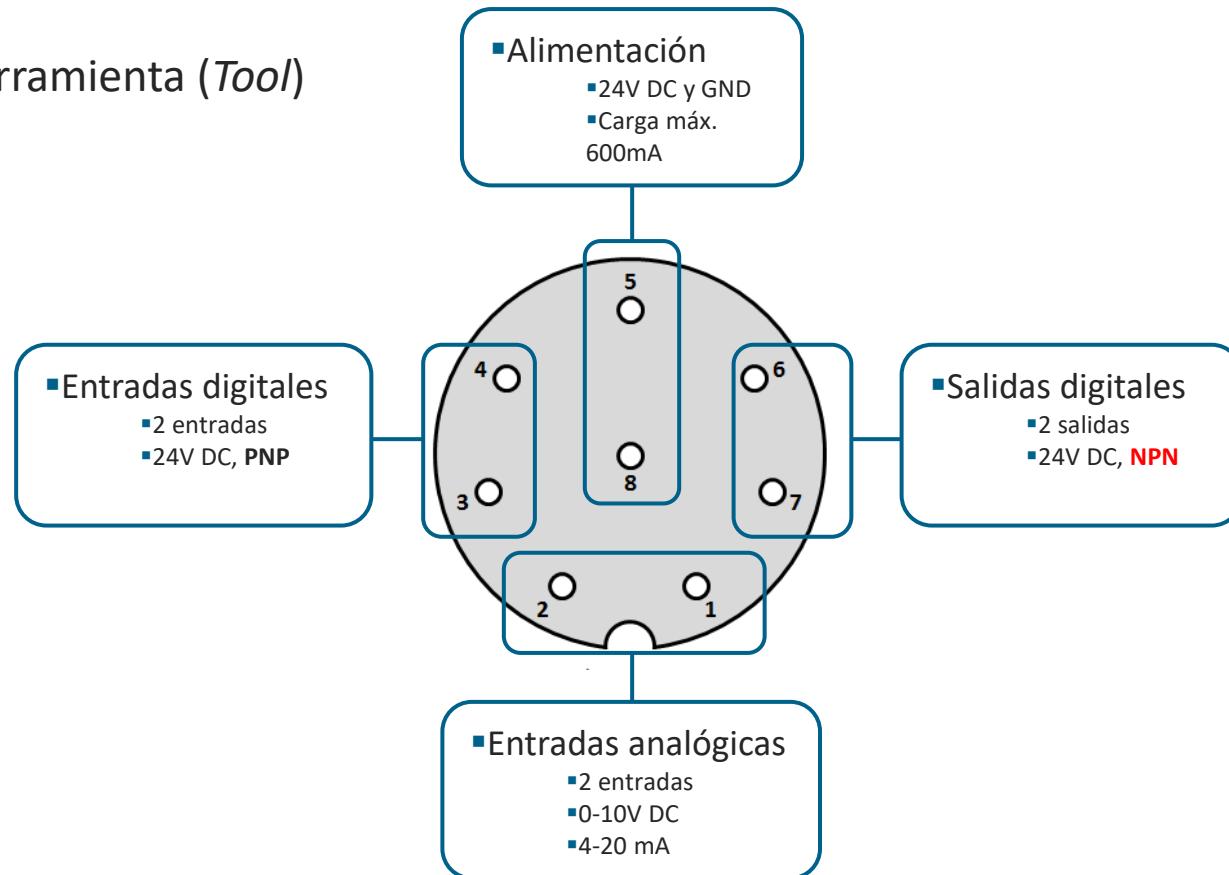
Analog	
AG	■
AI0	■
AG	■
AI1	■
AG	■
AI2	■
AG	■
A00	■
AG	■
A01	■

• Consultar en Manual de usuario, capítulo 4. Interfaz eléctrica

Interfaz eléctrica

- Conector de herramienta (*Tool*)

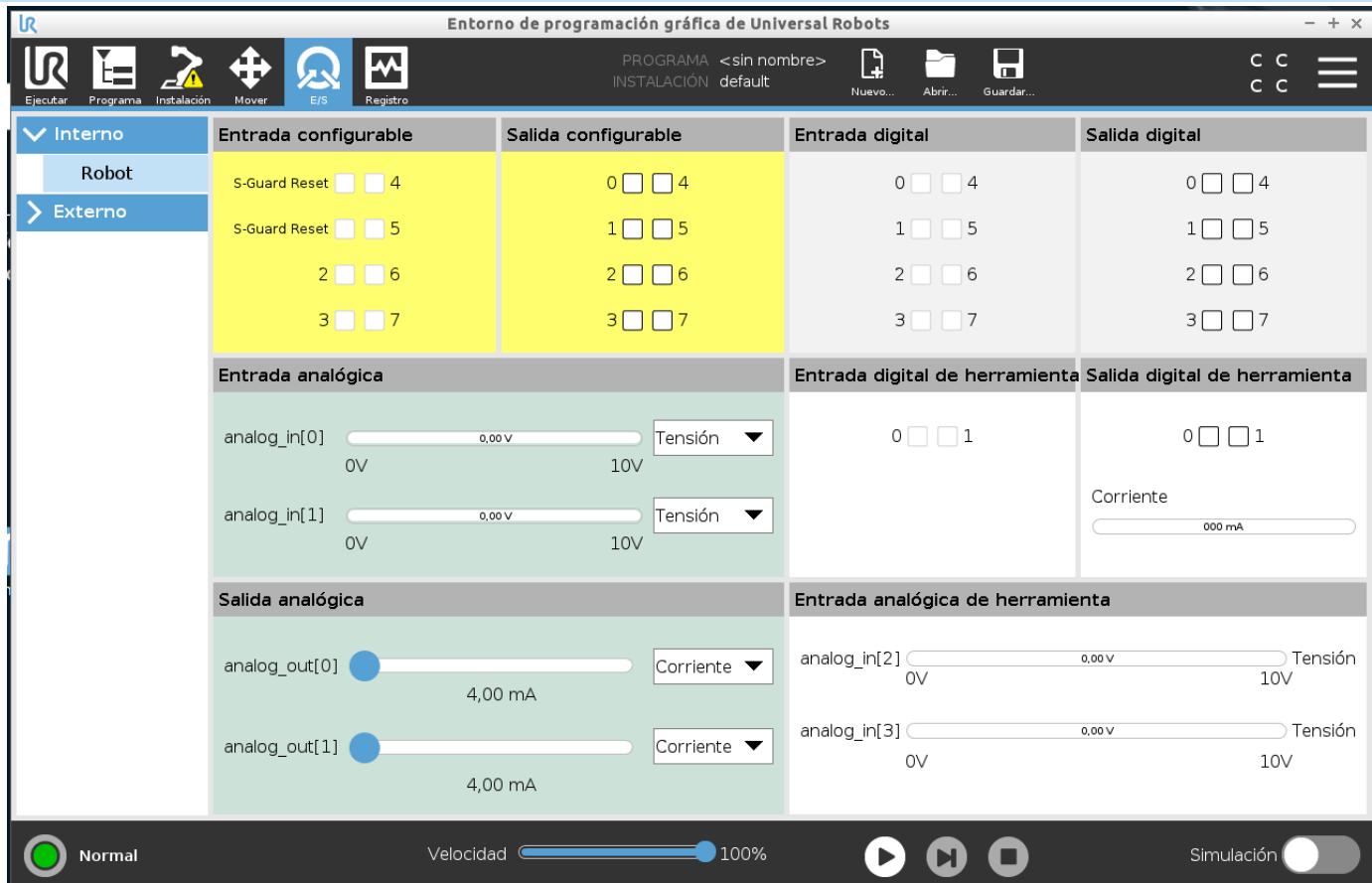
- pin 1 AI[2]
- pin 2 AI[3]
- pin 3 TI[0]
- pin 4 TI[1]
- pin 5 24V DC
- pin 6 TO[0]
- pin 7 TO[1]
- pin 8 GND



Manejo de señales

Propósito

- Interacción con dispositivos externos
- Cableadas al
 - Controlador
 - Conector de herramienta
- Configuración
 - Controlador
 - 16 DI
 - 16 DO
 - 2 AI
 - 2 AO
 - Conector de herramienta
 - 2 DI
 - 2 DO
 - 2 AI



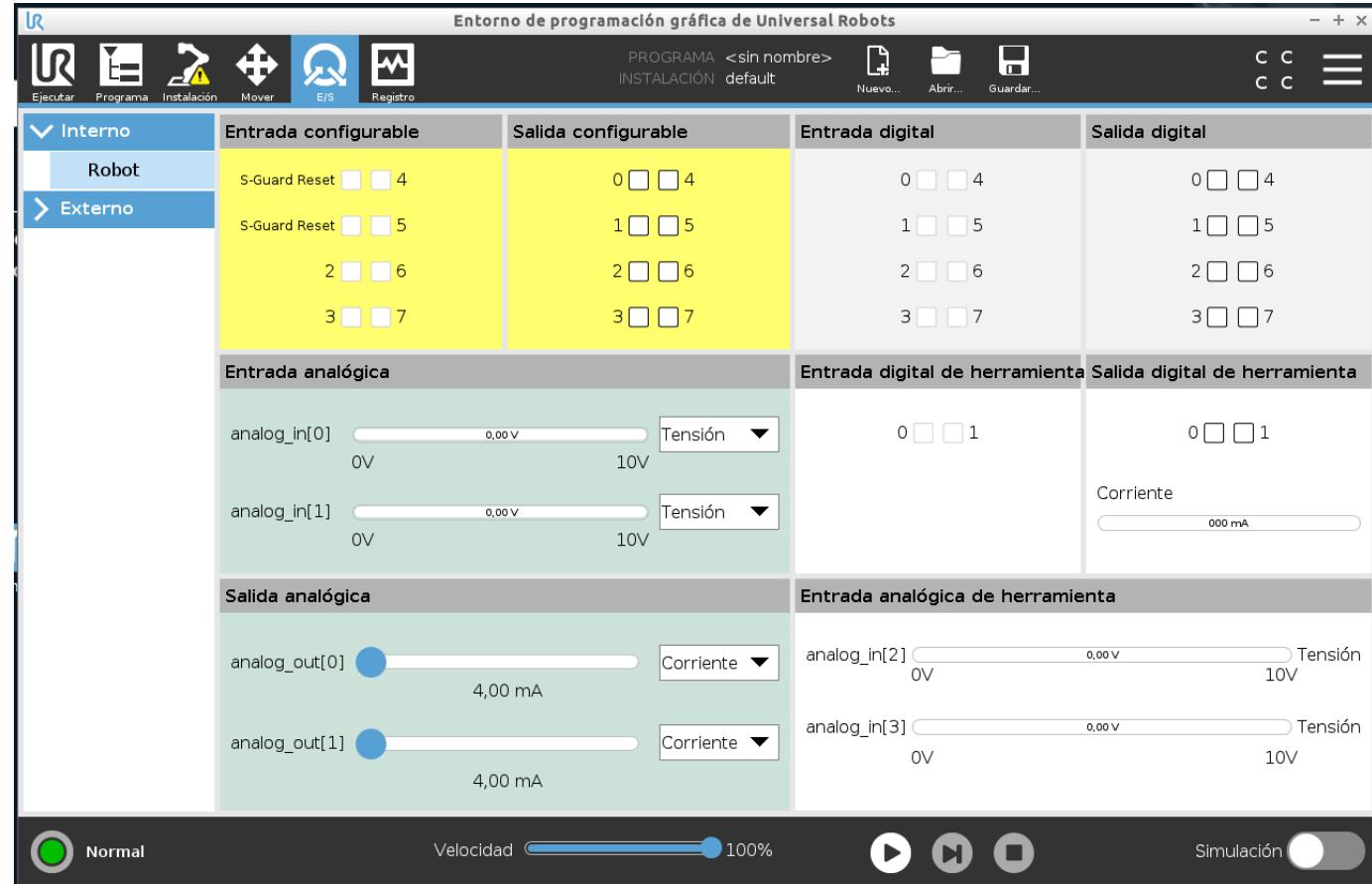
Manejo de señales

• Señales digitales

- Voltaje
 - 24V DC
 - GND
- Estados
 - OFF (Bajo)
 - ON (Alto)

• Señales analógicas

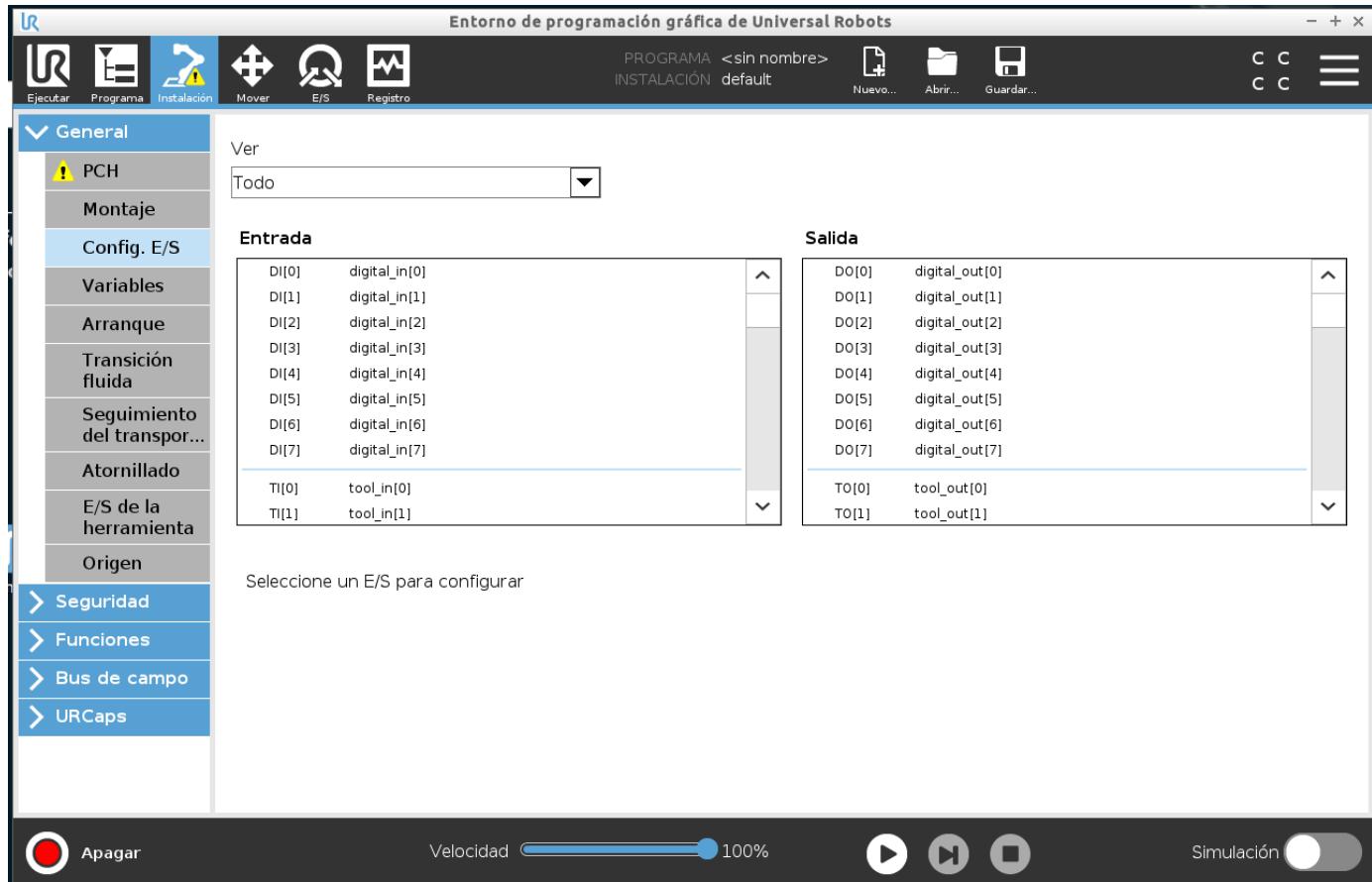
- Rango
 - Corriente 4-20 mA
 - Voltaje 0-10 V



Configuración de I/O

Entrada (Input)

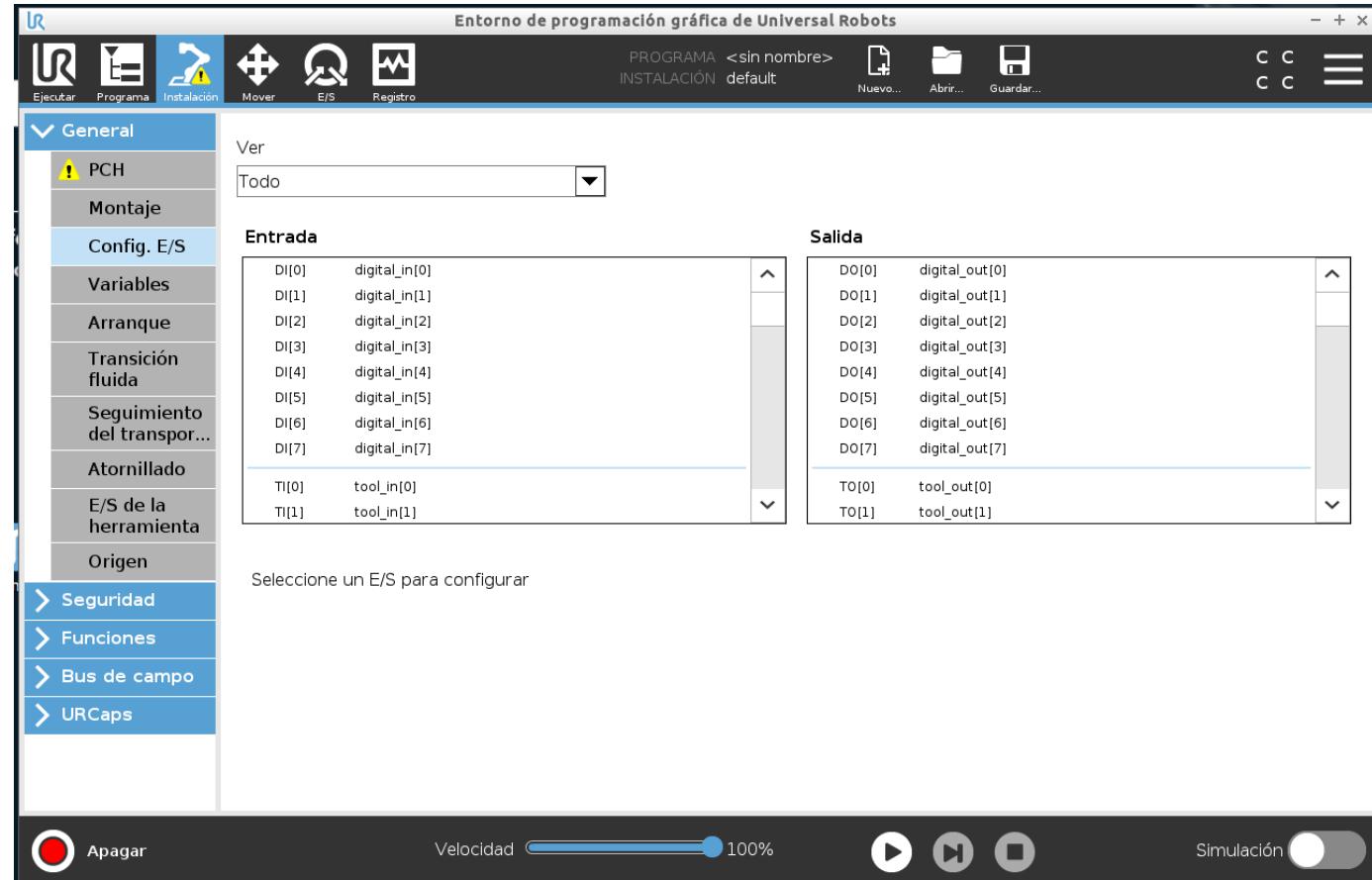
- Renombrar señales
- Asignar acción
 - Iniciar programa
 - Detener programa
 - Pausar programa
 - Movimiento libre
- Tanto el programa como la pestaña de I/O se actualizan al modificar el nombre de la señales



Configuración de I/O

Salida (Output)

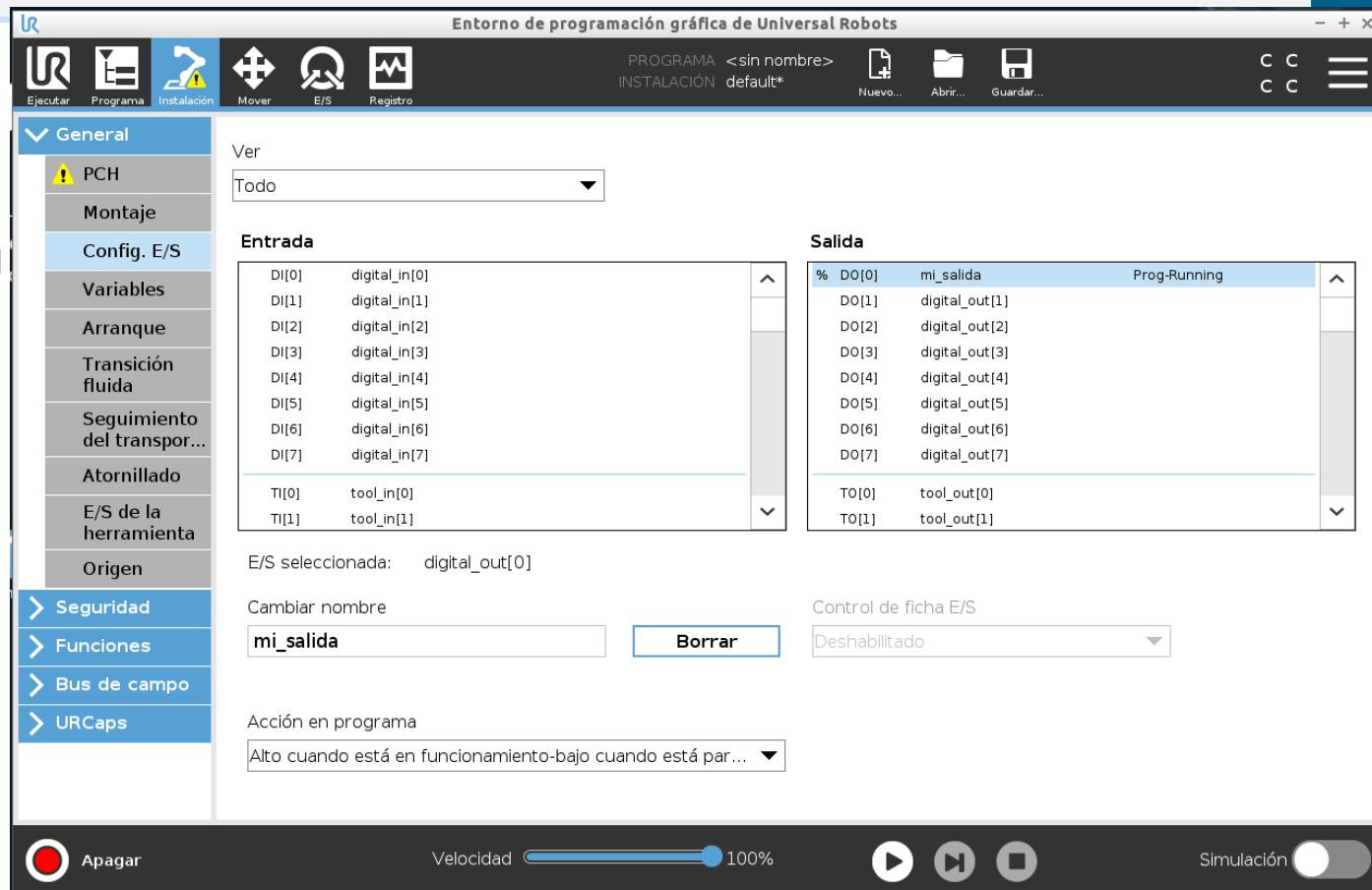
- Renombrar señales
- Opciones para cambiar el estado de la salida desde la pestaña de I/O



Configuración de I/O

Salida (Output)

- Renombrar señales
- Fijar el estado de la señal cuando el programa está parado



- Tanto el programa como la

Programa por defecto (Default program)

Propósito

- Configurar el robot para iniciar automáticamente sin la intervención del operador

Instrucciones

- Selecciona el programa “default”
- Establece una entrada digital a “Auto Start”
- Establece la entrada digital a “Auto Initialize”
- Guarda/salva la instalación
- Reinicia (Reboot)

The screenshot shows the 'General' configuration tab in the GRAP interface. It includes sections for 'PCH', 'Montaje', 'Config. E/S', 'Variables', 'Arranque', 'Transición fluida', 'Seguimiento del transpor...', 'Atornillado', 'E/S de la herramienta', and 'Origen'. Under 'Arranque', there is a note: 'Los cambios no se aplicarán hasta haber guardado la instalación'. The 'Auto inicializar' section contains a note: 'Si la opción Autoinitializar también está habilitada, el robot podrá empezar a moverse al encender la unidad. Esto solo está disponible en modo Control remoto.' At the bottom, there are buttons for 'Apagar', 'Velocidad' (100%), and playback controls.

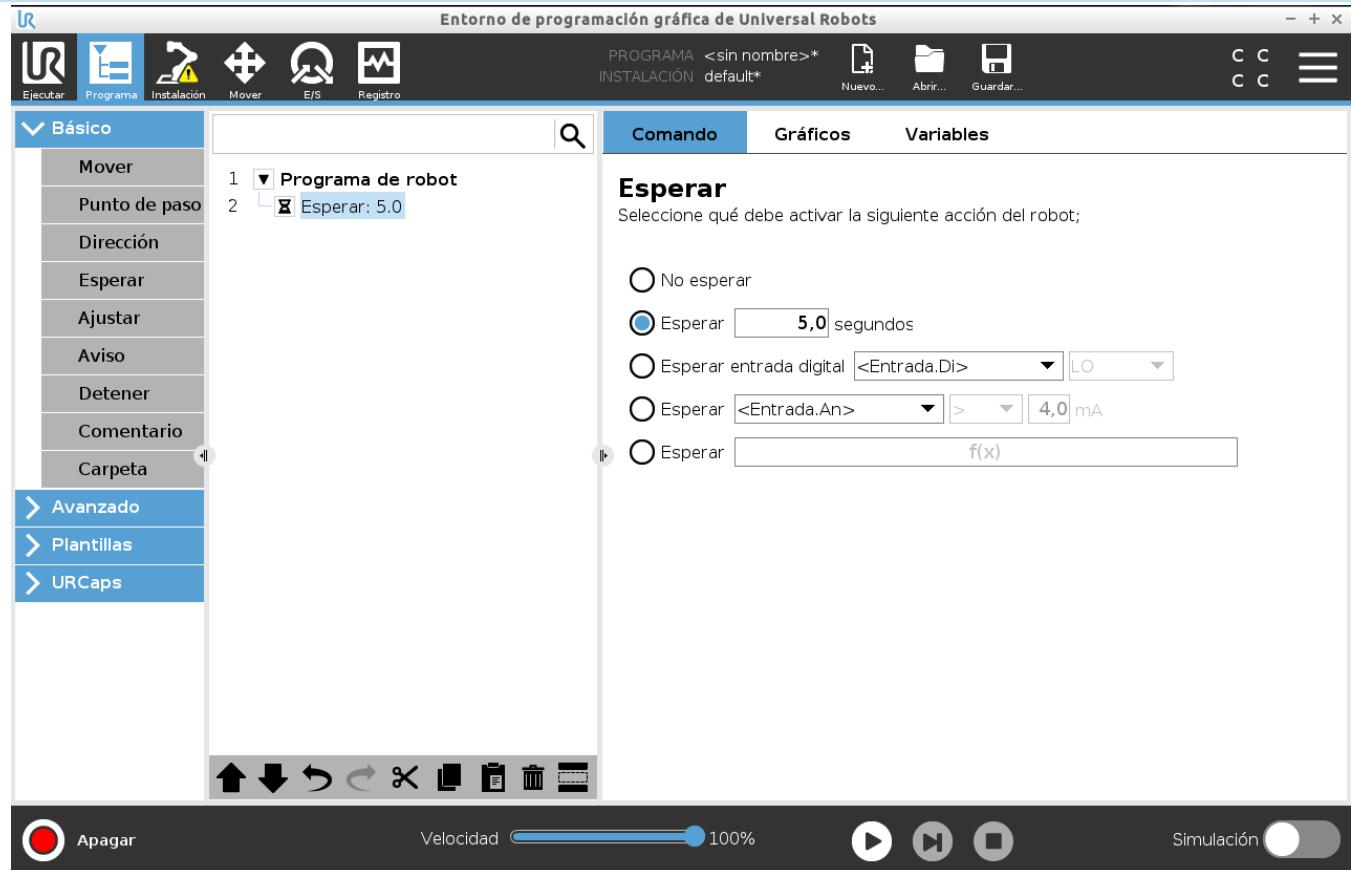
- Nota: El robot **siempre** carga “default.installation” en el arranque

Comando Wait (Esperar)

Espera por condiciones

- No esperar
- Esperar un tiempo
- Esperar entrada digital
- Esperar entrada analógica
- Esperar <expresión>

```
Robot Program
MoveL
    Waypoint_1
    Waypoint_2
    Wait 1.0
    Waypoint_3
    Waypoint_4
    Wait DI[0] = True
```



- Guardar programa de ejemplo como wait.urp

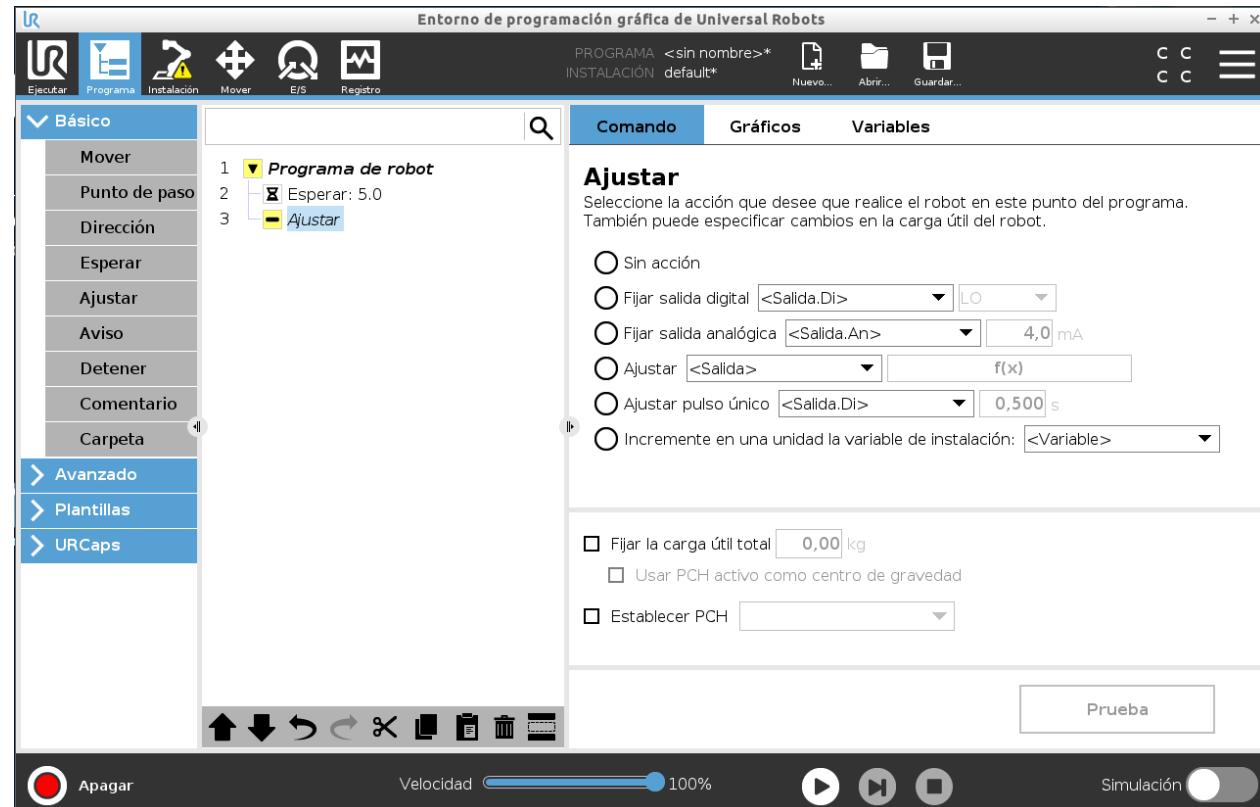
Comando Set (Ajustar)

- Realizar una acción

- Sin acción
- Fijar salida digital
- Fijar salida analógica
- Ajustar <expresión>
- Incrementar variable
- Fijar la carga útil
 - Usa el TCP activo como CoG
- Fijar TCP

```
Robot Program
MoveL
Waypoint_1
Waypoint_2
Wait 1.0
Waypoint_3
Waypoint_4
Set DO[0] = True
```

- Guardar programa de ejemplo como set.urp



Comando Popup (Aviso)

- Espera al operario
 - Pausa el programa
 - Define el mensaje de aviso
 - Tipos de aviso
 - Mensaje
 - Advertencia
 - Error

Robot Program

MoveL

Waypoint_1

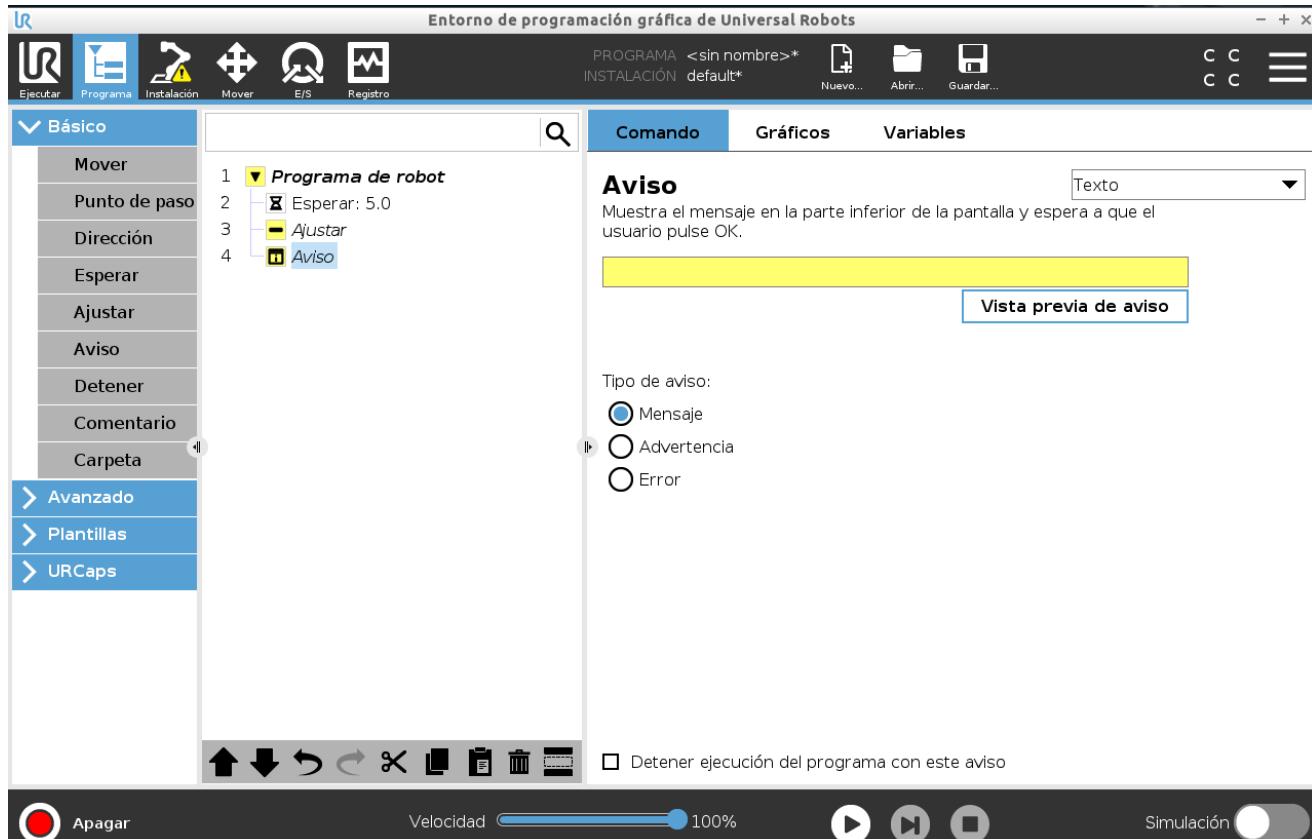
Waypoint_2

Wait 1.0

Waypoint_3

Waypoint_4

Popup



- Guardar programa de ejemplo como popup.urp

Comando Halt (*Detener*)

Detener (*Halt*)

- Detiene la ejecución del programa

Robot Program

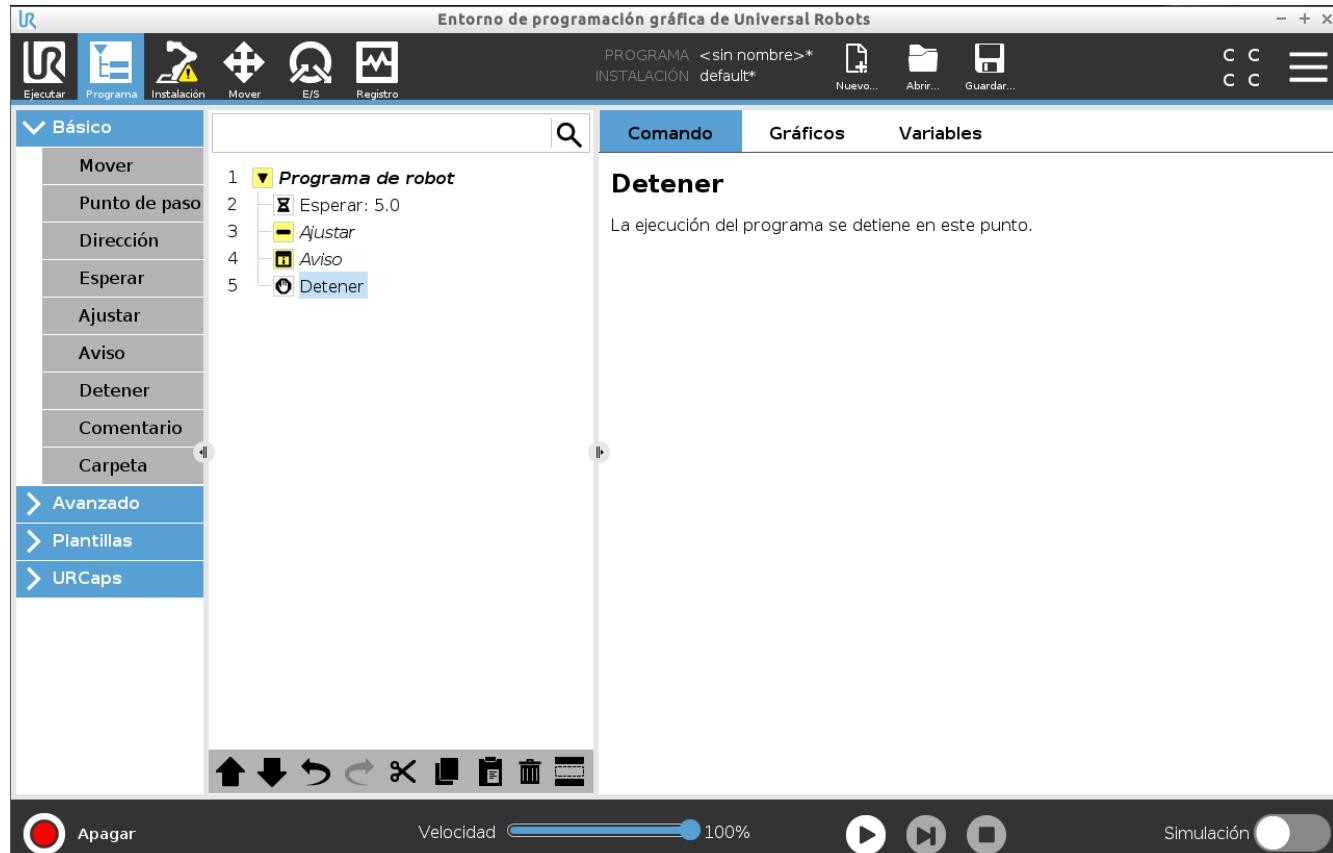
MoveL

 Waypoint_1

 Waypoint_2

 Halt

- Guardar programa de ejemplo como `halt.urp`



Comando Comment (Comentario)

Comentario (Comment)

- Añade texto al programa
- La ejecución del programa no se ve afectada

Robot Program
MoveL

Waypoint_1

Waypoint_2

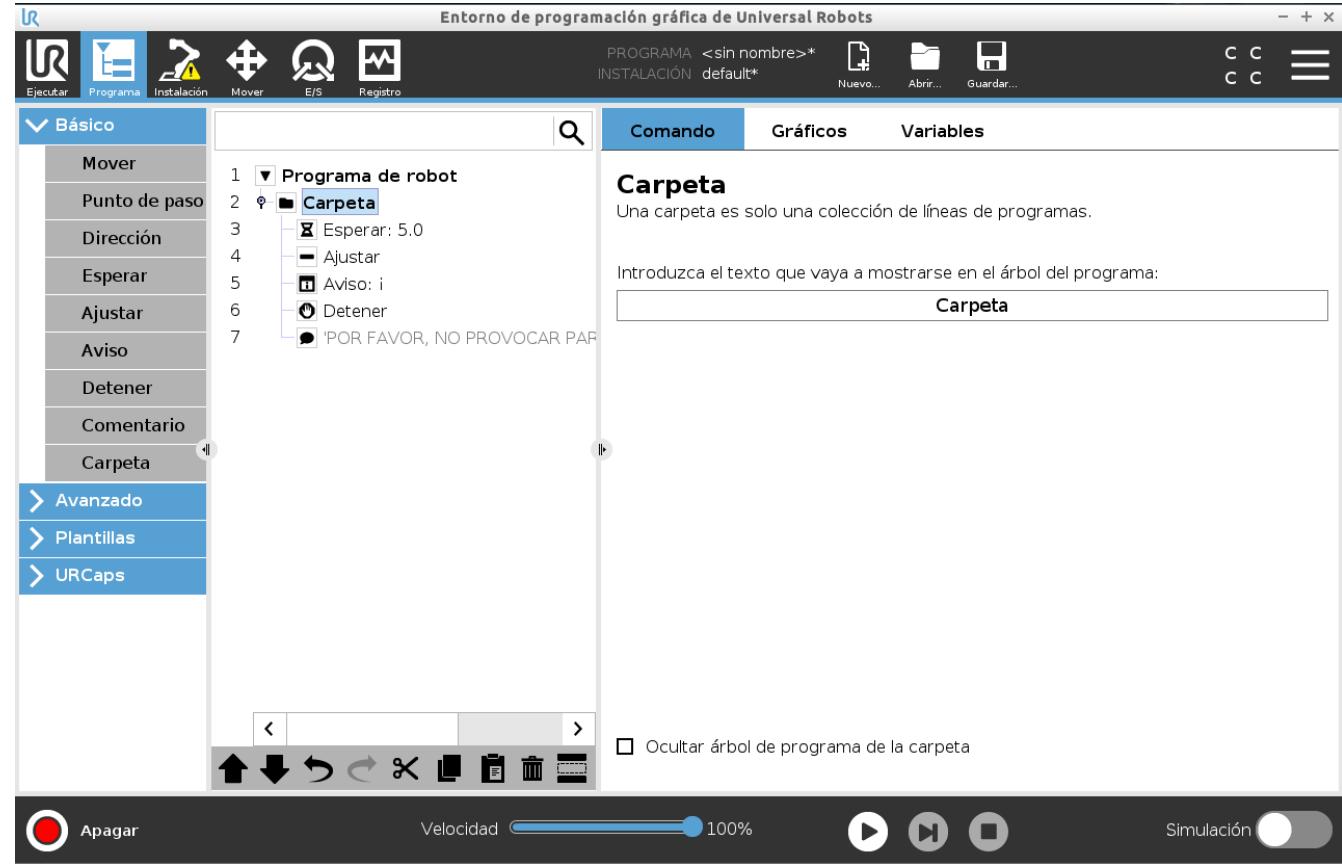
'Enter comment

The screenshot shows the Universal Robots graphical programming interface. The left sidebar has a tree view under 'Básico' with nodes like Mover, Punto de paso, Dirección, Esperar, Ajustar, Aviso, Detener, Comentario, and Carpeta. The 'Comentario' node is selected. The main workspace shows a sequence of blocks: 1. Programa de robot, 2. Esperar: 5.0, 3. Ajustar, 4. Aviso, 5. Detener, and 6. Comentario with the text 'POR FAVOR, NO PROVOCAR PARADAS DE PROTECCION'. To the right, there's a panel titled 'Comentario' with a text input field containing the same text. At the bottom, there are playback controls (play, stop, etc.) and a speed slider set to 100%.

Comando Folder (Carpeta)

Carpetas (Folders)

- Organiza el programa
- Agrupa parte del programa
- La ejecución del programa no se ve afectada



Comando Carpeta

- Programa de ejemplo recoger y colocar

- Dos carpetas

- Recoger_pieza (Pick_part)
 - Colocar_pieza (Place_part)

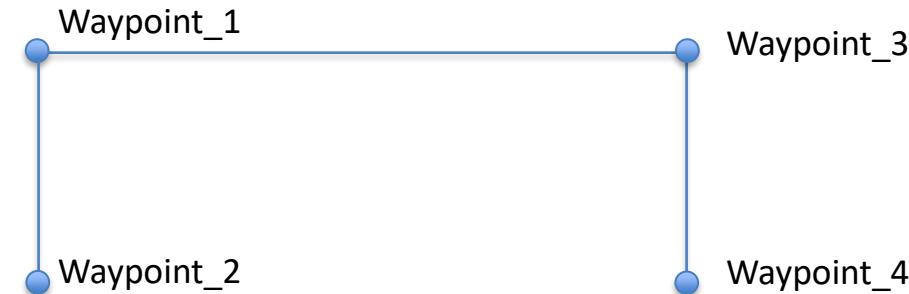
Robot Program

Pick_part

MoveL
Waypoint_1
Waypoint_2
Set DO[0] = On
Wait 0.5
Waypoint_1

Place_part

MoveL
Waypoint_3
Waypoint_4
Set DO[0] = Off
Wait 0.5
Waypoint_3



- Guardar programa de ejemplo como pick_and_place.urp

Ejercicio práctico 2

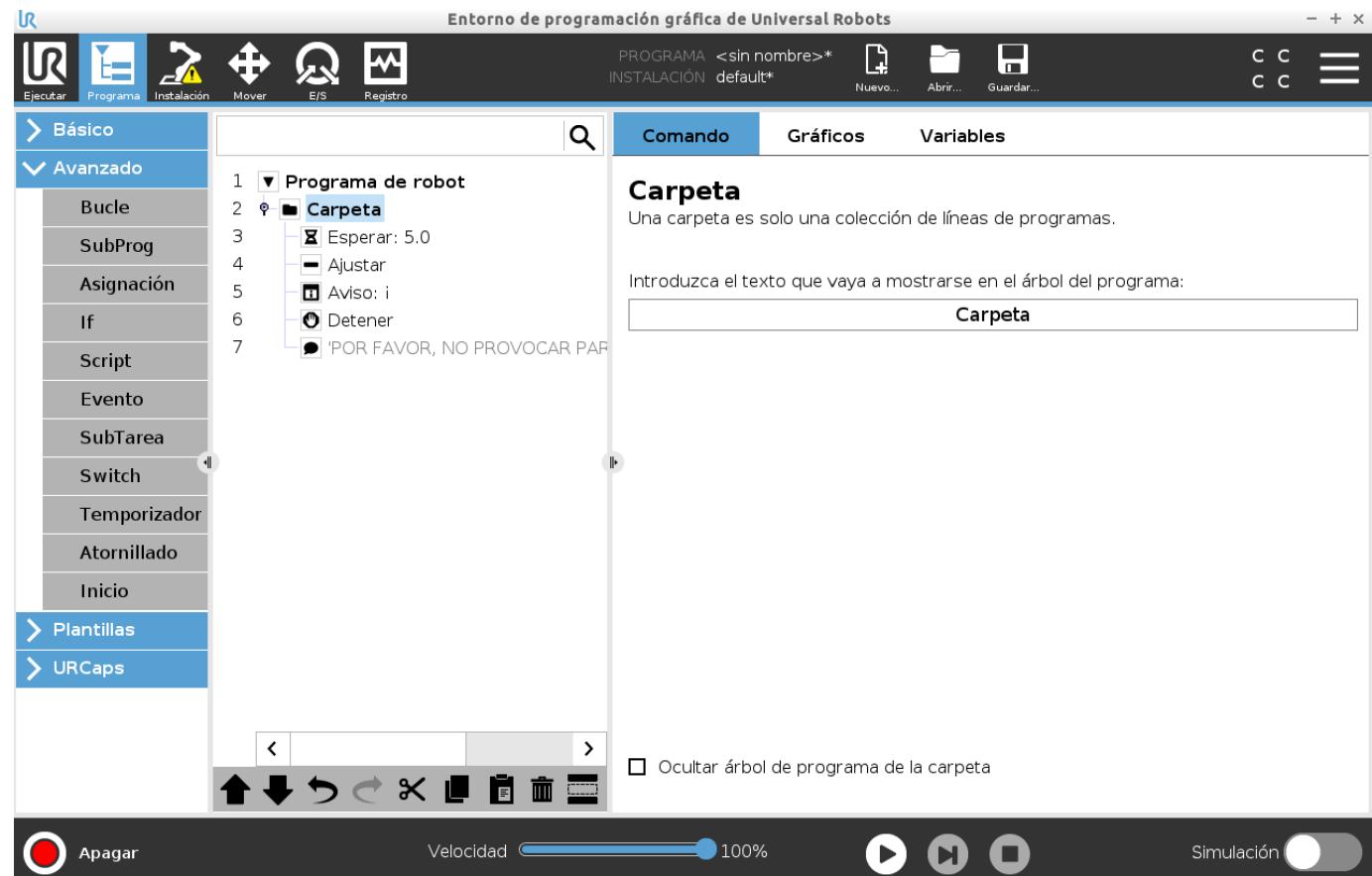
- Programa – Carpeta_Recoger
 - Crear una carpeta y nombrarla como Carpeta_Recoger.
 - Dentro de esta carpeta mueve el robot al Waypoint_1 y desde esa posición linealmente al Waypoint_2
 - Cierra el gripper (herramienta) en el Waypoint_2 (simulado), espera un momento y regresa linealmente al Waypoint_1
- Programa – Carpeta_Colocar
 - Crear una carpeta y nombrarla como Carpeta_Colocar.
 - Dentro de esta carpeta mueve el robot al Waypoint_3 y desde esa posición linealmente al Waypoint_4
 - Abre el gripper (herramienta) en el Waypoint_4 (simulado), espera un momento y regresa linealmente al Waypoint_3
- Guardar programa de ejemplo como Lab_exe_4_2.urp

1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes / Wizards
3	Comandos básicos 1	9	Modbus TCP / Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

Pestaña de comandos avanzados

Comandos avanzados

- Realizar operaciones avanzadas



Comando If ... else

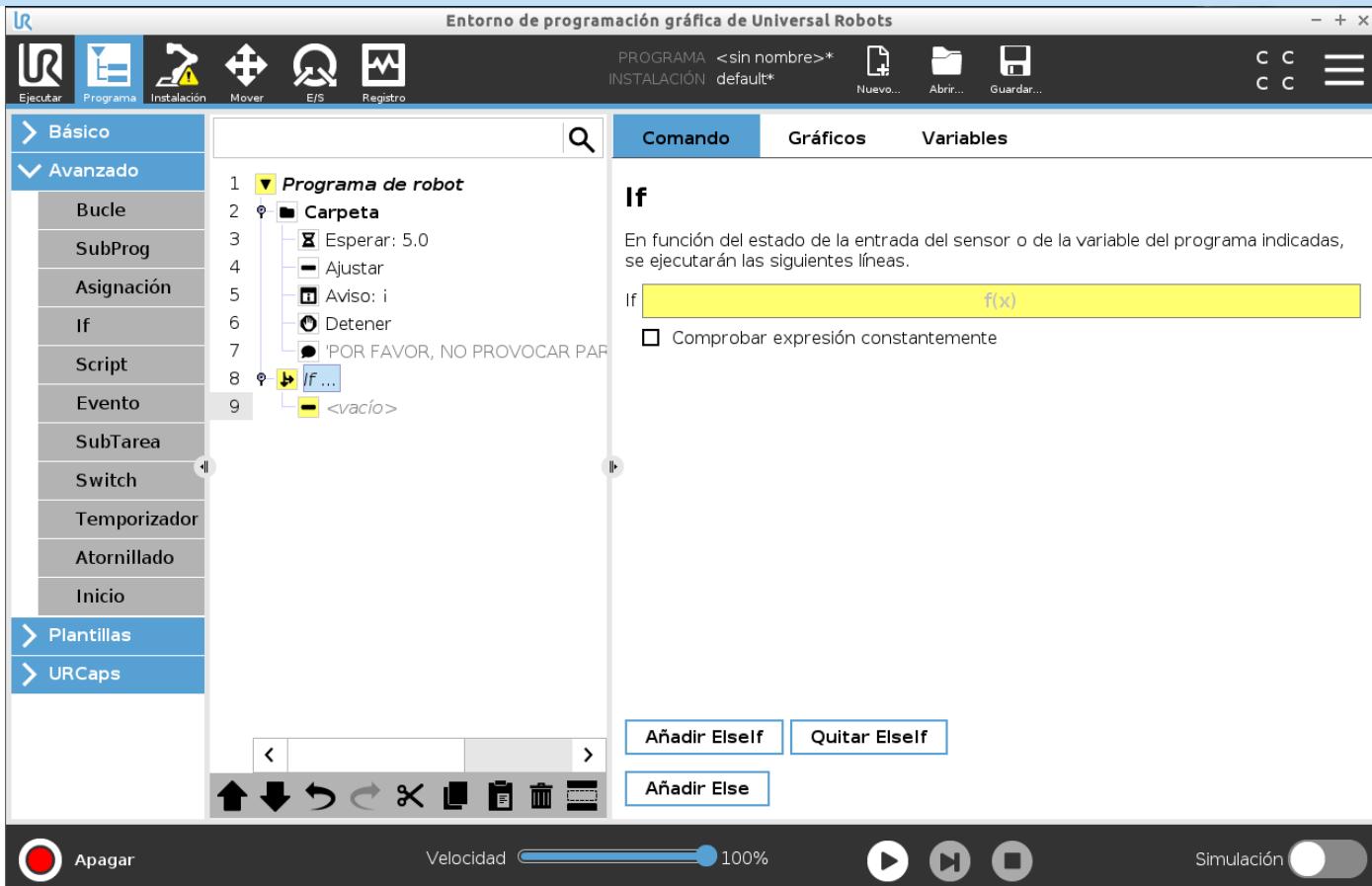
If

- Evalúa una condición:
 - Estado de un sensor
 - Valor de una variable
 - Combinación de varios estados
- If condición = True
 - Ejecuta los comandos contenidos

Robot Program

```
MoveL
  Waypoint_1
  IF DI[0] = True
    Waypoint_2
  Else
    Waypoint_3
```

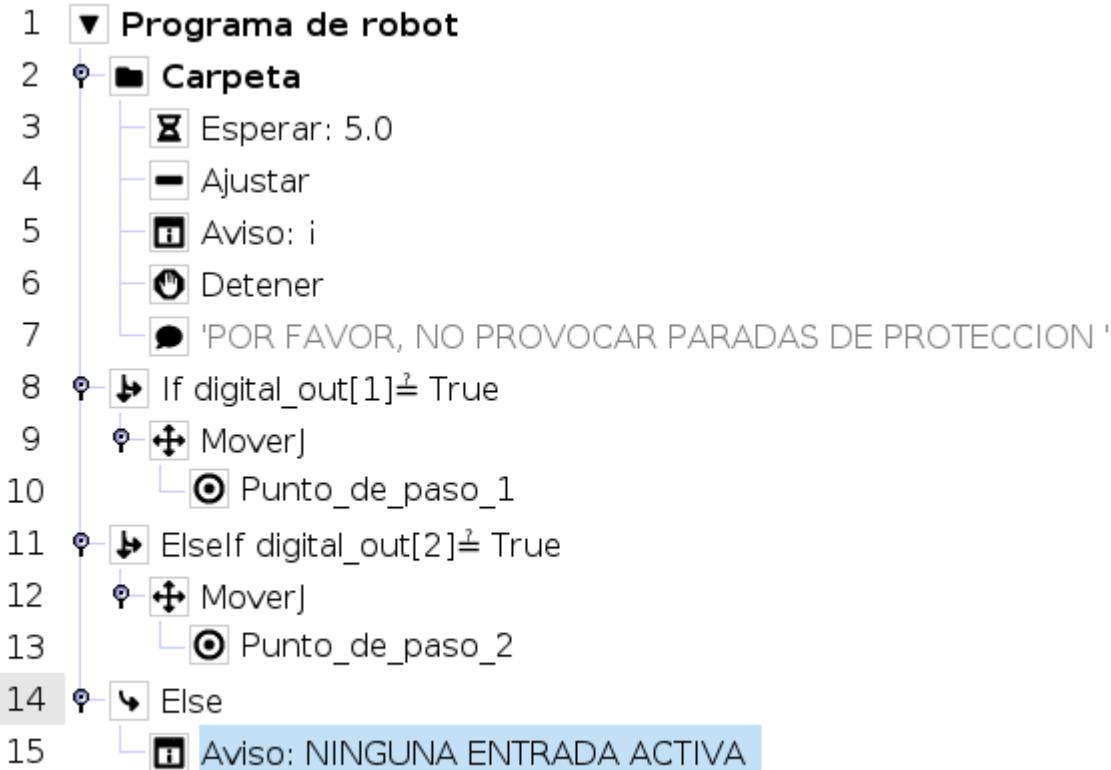
- Guardar programa de ejemplo como if.urp



Comando *If ... else*

- ... elseif
 - Examina nueva condición
- ... else
 - Define qué hacer si no se cumple ninguna condición

```
Robot Program
MoveL
    Waypoint_1
    IF DI[0] = True
        Waypoint_2
    Elseif
        Waypoint_3
```



- Guardar programa de ejemplo como if_else.urp

Usando sync()

```
Robot Program  
MoveL  
    Waypoint_1  
    if var = 1  
        Waypoint_2  
    elseif var = 2  
        Waypoint_3  
    elseif var = 3  
        Waypoint_4  
    sync()
```

```
Robot Program  
MoveL  
    Waypoint_1  
    if var = 1  
        if DI[0] = True  
            Waypoint_2  
            sync()  
    elseif var=2  
        Waypoint_3  
    elseif var=3  
        Waypoint_4  
    sync()
```

- Sync()
 - Utiliza el tiempo restante "físico" del procesador
- “Scan” de 8ms o 125Hz

Comando Loop (Bucle)

Definición

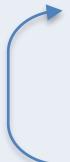
- Ejecuta cíclicamente las instrucciones que contiene
- Tipos de Loop (*bucle*)
 - Loop always (*Bucle siempre*)
 - Loop n times (*Bucle n veces*)
 - Loop <expresión>

Robot Program

```

MoveJ
  Waypoint_5
Loop 3 times
  MoveL
    Waypoint_1
    Waypoint_2
    Waypoint_3
    Waypoint_4
MoveJ
  Waypoint_5

```



```

1 ▼ Programa de robot
2 ⚡ Bucle
3 ⚡ If digital_out[1] == True
4   ⚡ Mover]
5     ⚡ Punto_de_paso_1
6   ⚡ Elseif digital_out[2] == True
7     ⚡ Mover]
8       ⚡ Punto_de_paso_2
9   ⚡ Else
10  ⚡ Aviso: NINGUNA ENTRADA ACT

```

Bucle

Seleccione cuántas veces debe ejecutarse el programa en este bucle.

Bucle siempre

Bucle X veces:

Recuento de bucles

Nombre de variable

Bucle_1

Bucle cuando la expresión es Verdadera

Comprobar expresión constantemente

Comando Bucle

Interrupción

- Monitorea expresión constantemente

Robot Program

```
MoveJ  
    Waypoint_5  
Loop DI[0] = True  
    MoveL  
    Waypoint_1  
    Waypoint_2  
    Waypoint_3  
    Waypoint_4  
MoveJ  
    Waypoint_5
```

```
1  ▼ Programa de robot  
2  ⚡ Bucle digital_in[5] ≠ False  
3  ⚡ If digital_out[1] ≠ True  
4      ⚡ Mover]  
5          ○ Punto_de_paso_1  
6  ⚡ ElseIf digital_out[2] ≠ True  
7      ⚡ Mover]  
8          ○ Punto_de_paso_2  
9  ⚡ Else  
10     ┌ Aviso: NINGUNA ENTRADA ACT
```

Bucle

Seleccione cuántas veces debe ejecutarse el programa en este bucle.

Bucle siempre

Bucle X veces:

Recuento de bucles

Nombre de variable

Bucle_1

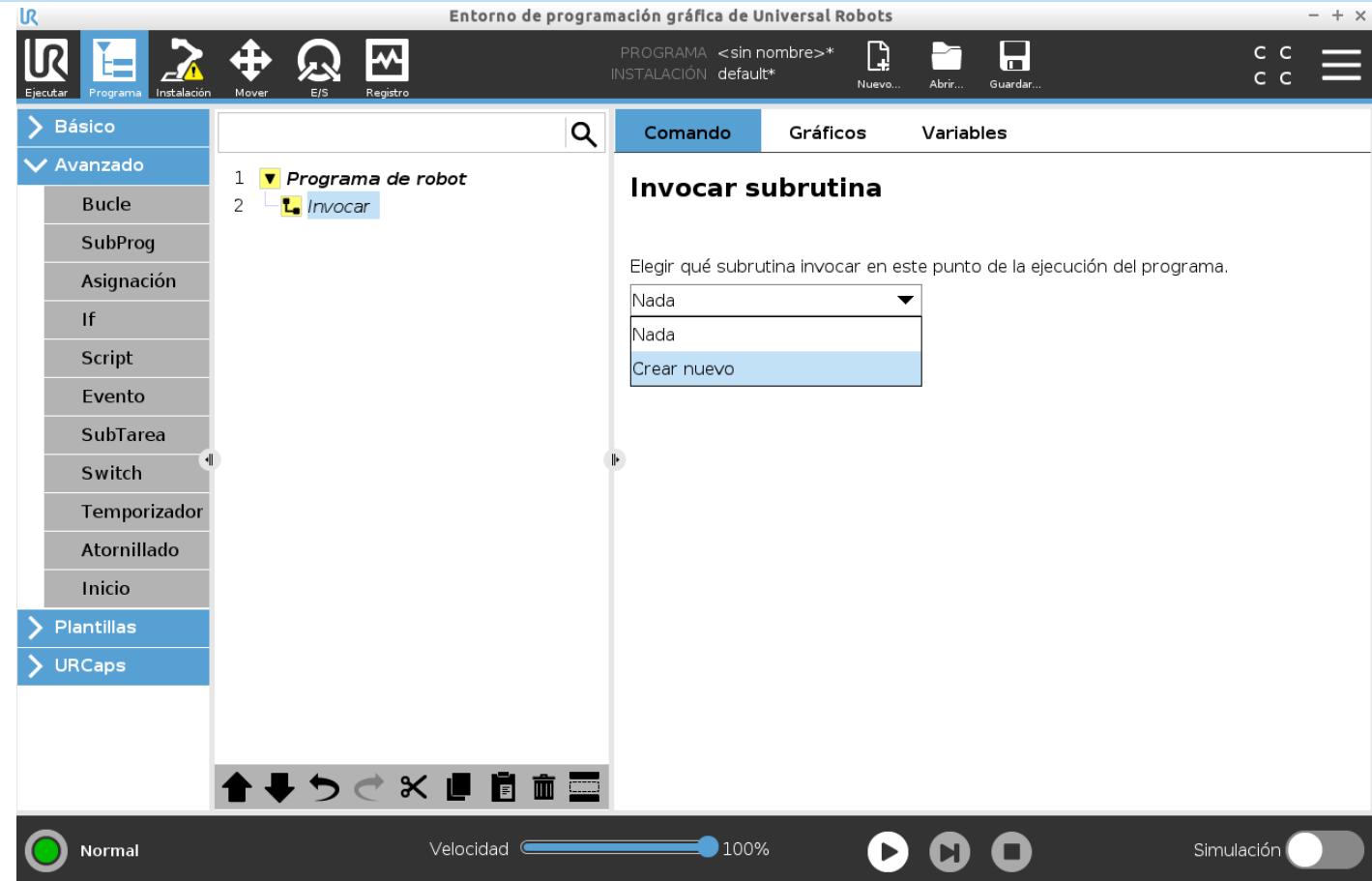
Bucle cuando la expresión es Verdadera

Comprobar expresión constantemente

Comando SubProg

Subrutinas

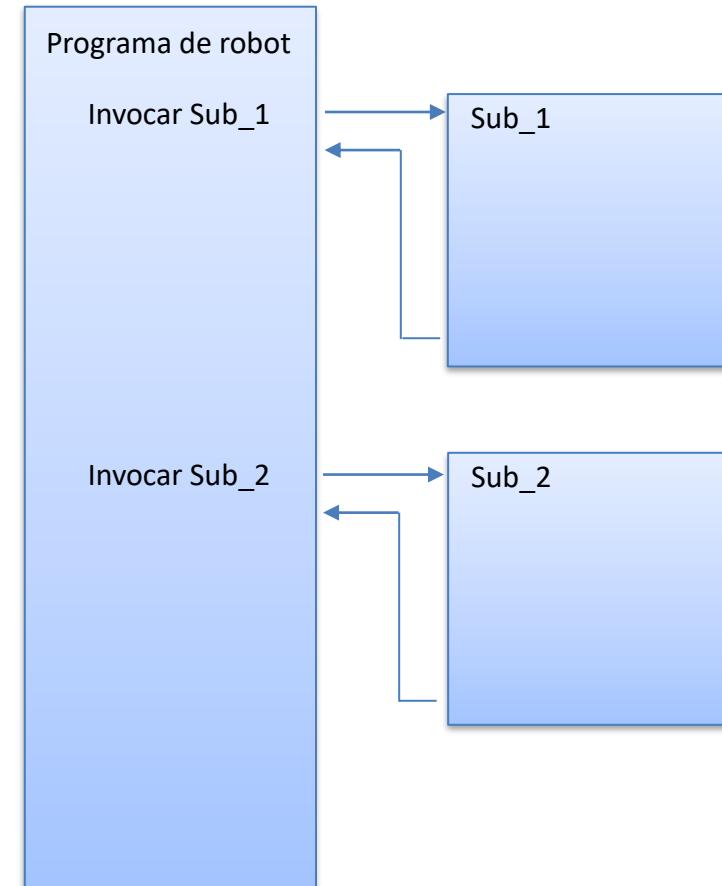
- Organizar programa
- Dividir programa en subprogramas
- Reutilizar subprogramas en múltiples programas



Comando *SubProg*

- **Invocando una subrutina**

- Primero se ejecuta el programa principal
- Se invoca una subrutina desde el programa principal
- Se “pausa” el programa principal
- Se ejecutan todos los comandos de la subrutina
- Después regresa al programa principal y continua con el comando posterior al de “invocar la subrutina”



- Nota: No se soporta invocar una subrutina desde una subrutina

Comando SubProg

Invocar subprograma

- Cargar archivo existente
- Crear subprograma en un programa nuevo
 - Guardar subprograma como archivo
 - Parte del programa principal

Robot Program

MoveJ
Waypoint_5

Loop 3 times

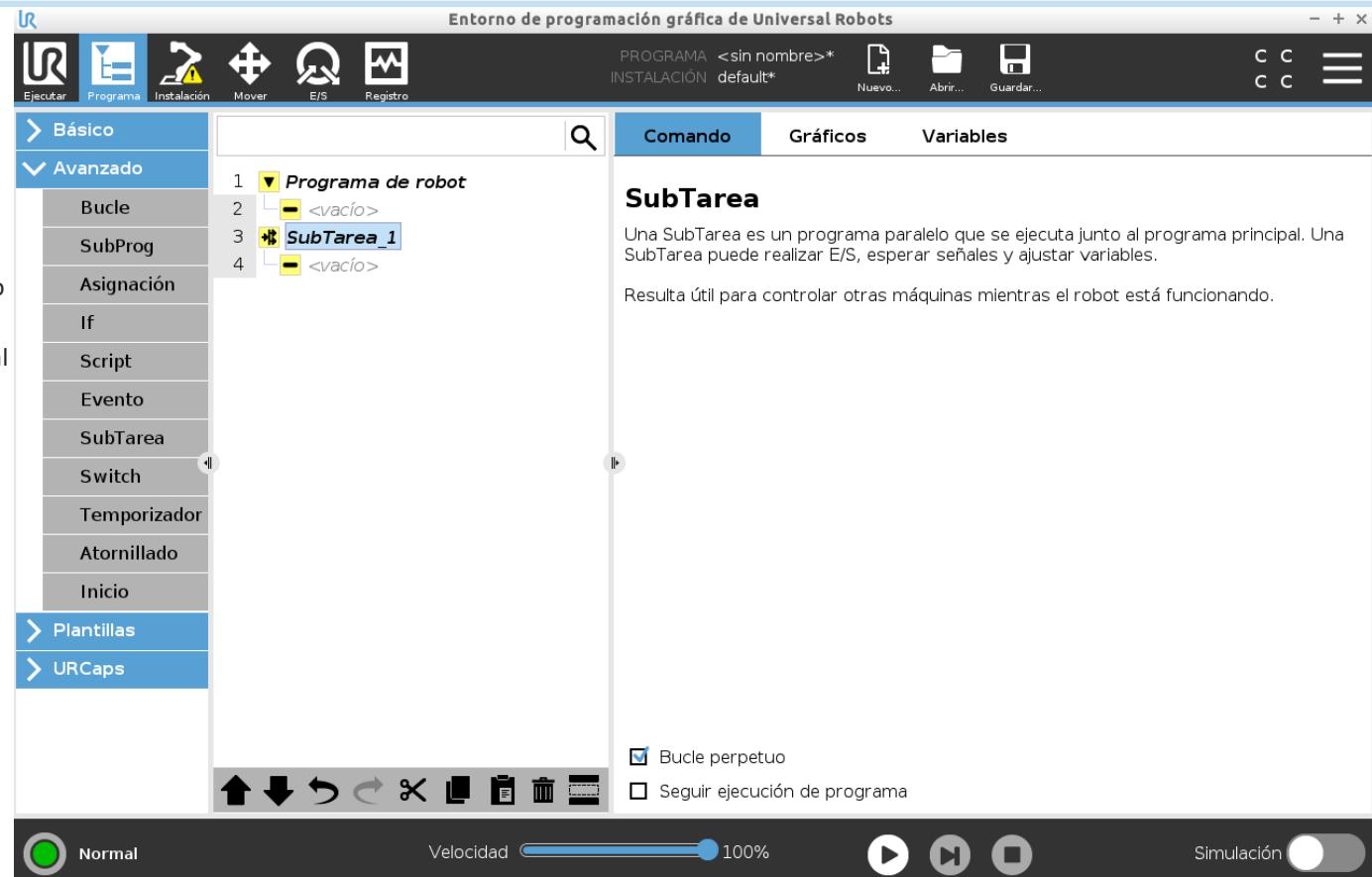
MoveL

Waypoint_1
Waypoint_2
Waypoint_3
Waypoint_4

Call SubP_movec

MoveJ

Waypoint_5



Ejercicio práctico 1

- Crear un programa usando subprogramas, “loops” e instrucciones “if”
- Primero crear un programa simple para prender/apagar una lámpara de forma intermitente (este programa será usado como subprograma)
- Lampa_intermitente:
 - Crear un programa que conecte una lámpara de forma intermitente 5 veces usando un loop
 - Prender la lámpara durante 0.5 segundos, luego apagarla durante 0.5 segundos (5 repeticiones)
 - Guardar el programa con el nombre Lampa_intermitente
- Guardar programa de ejemplo como Lab_exe_5_1.urp

Ejercicio práctico 2

- **Crear otro nuevo programa**

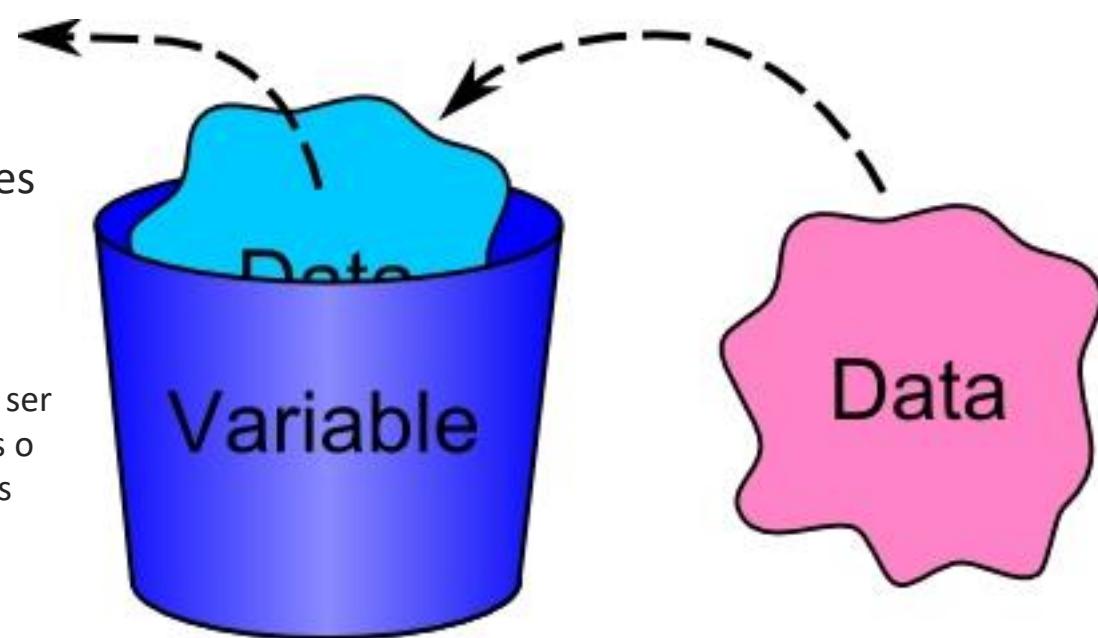
(Mover entre los puntos 1 y 2 varias veces, llamando subprograma. Detener inmediatamente si se pulsa el botón)

- Crear 3 “Waypoints” en posiciones de su elección
 - Mover al Waypoint_1
 - Llamar al subprograma Lampara_intermitente
 - Mover al Waypoint_2
 - Monitorear pulsación del botón
 - Si el botón ha sido pulsado, mover al Waypoint_3 y parar
 - En otro caso, mover al Waypoint_1
-
- Guardar programa de ejemplo como Lab_exe_5_2.urp

1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes
3	Comandos básicos 1	9	Modbus TCP/Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

¿Qué es una variable?

- Una variable es un espacio de almacenamiento (contenedor)
 - Su contenido puede cambiar
- Lectura/Escritura de variables
 - Su valor puede ser sobrescrito
 - Su valor puede ser leído
 - El valor de las variables puede ser comparado con otras variables o con los estados de los sensores



Tipos de variable

Tipo de variable	Valor
<code>boolean</code>	True / False
<code>integer</code>	Números enteros (32 bit)
<code>floating point</code>	Números reales (decimal)
<code>string</code>	Texto (caracteres ASCII)
<code>pose</code>	Variable de posición p[x,y,z,rx,ry,rz]
<code>list</code>	Lista de variables

Ámbito de las variables

Ámbito	Localización
local	Programa
global	Instalación

- Variables Locales
 - Declaradas en el programa
 - Accesibles desde el mismo programa
 - Su valor se pierde cuando se desconecta la alimentación
- Variables Globales
 - Declaradas en Instalación
 - Accesibles desde todos los programas que usan la misma Instalación
 - Su valor se almacena en un archivo en la memoria no volátil

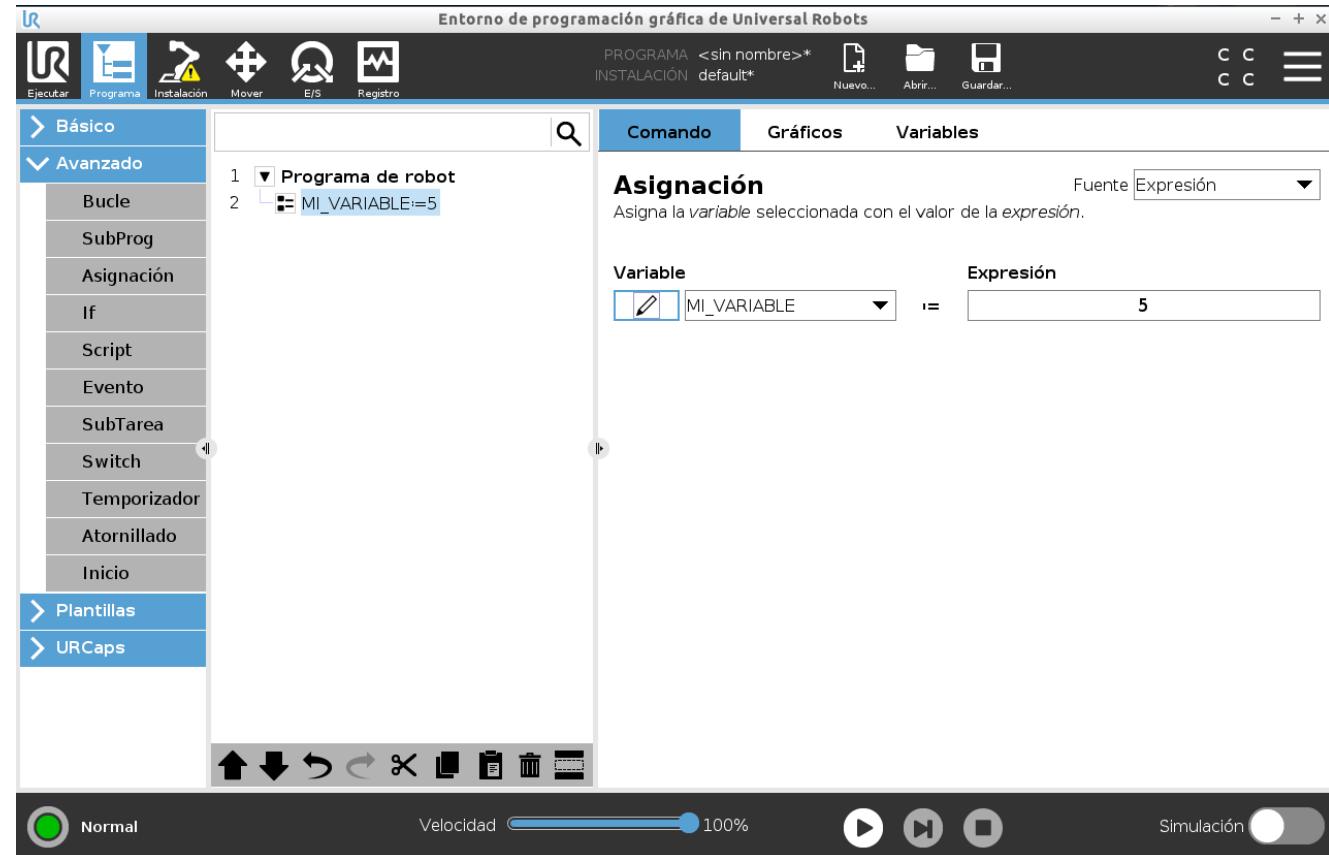
Comando Assigment (Asignación)

Opciones

- Definir nombre de variable
- Declarar tipo de variable
- Asignar valor a la variable

Robot Program

```
var_1 = True  
Wait 0.5  
var_1 = False  
Wait 0.5
```



Editor de expresiones

Opciones

- Valores numéricos
- Entradas (*Input*)
- Salidas (*Output*)
- Variables
- Posiciones
- Códigos Script
- Operadores lógicos
- Teclado

Entorno de programación gráfica de Universal Robots

Ejecutar Programa Instalación Mover E/S Registro

PROGRAMA <sin nombre>* INSTALACIÓN default* Nuevo... Abrir... Guardar...

C C C C

Básico

Avanzado

- Bucle
- SubProg
- Asignación
- If
- Script
- Evento
- SubTarea
- Switch
- Temporizador
- Atornillado
- Inicio

Plantillas

URCaps

Comando Gráficos Variables

Asignación

Fuente Operador

Asigne la variable seleccionada con el valor que deba introducir el operador de la máquina.

Variable: MI_VARIABLE

Solicitar operador para:

- Sí o No
- Un número entero
- Un número decimal
- Una cadena de texto

Mensaje de operador

Normal Velocidad 100% Simulación

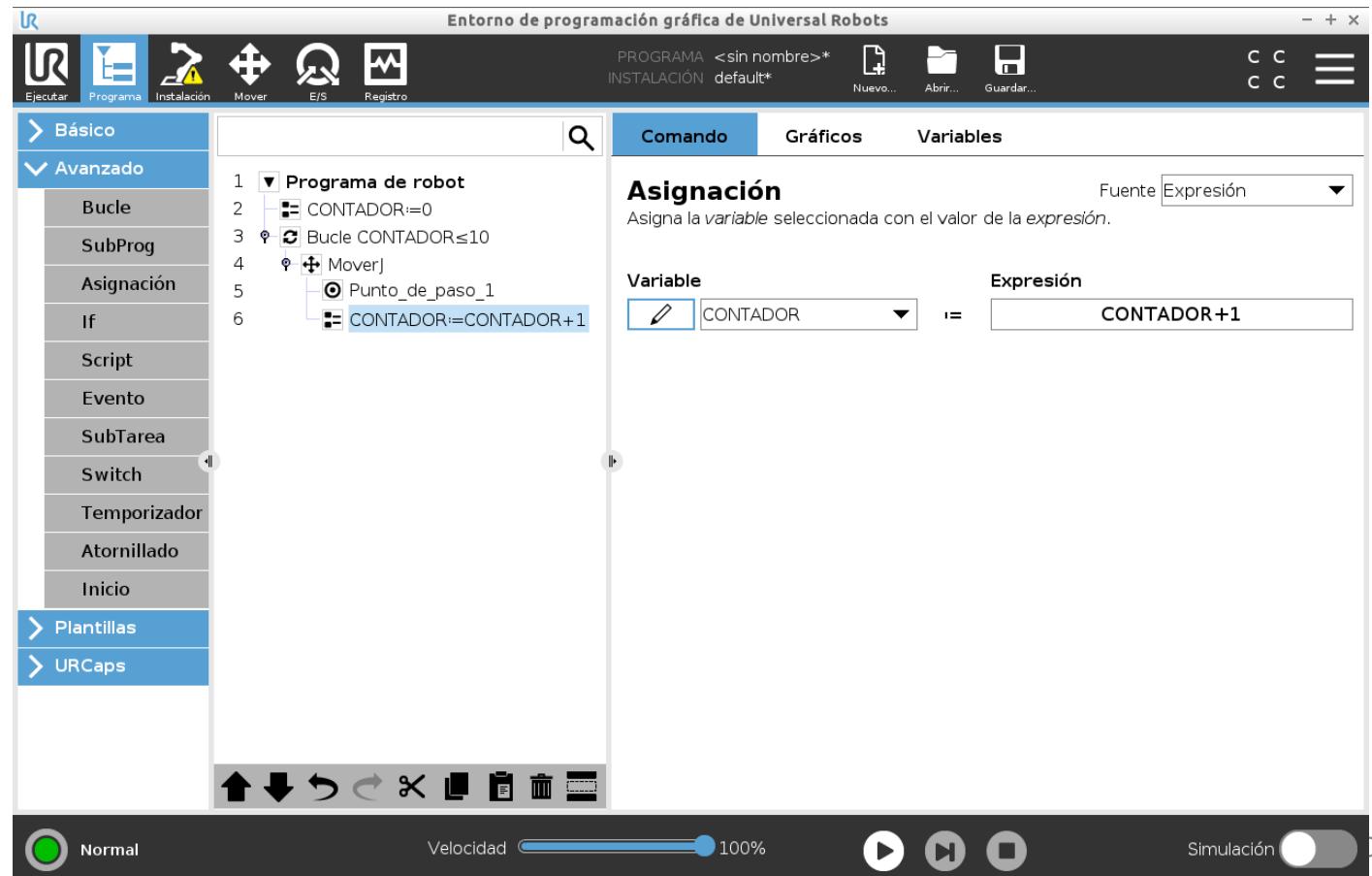
Usar una variable como contador

Contador

- Variable tipo entero
- Incrementar variable en loop (*bucle*)
- Comparar la variable con un número

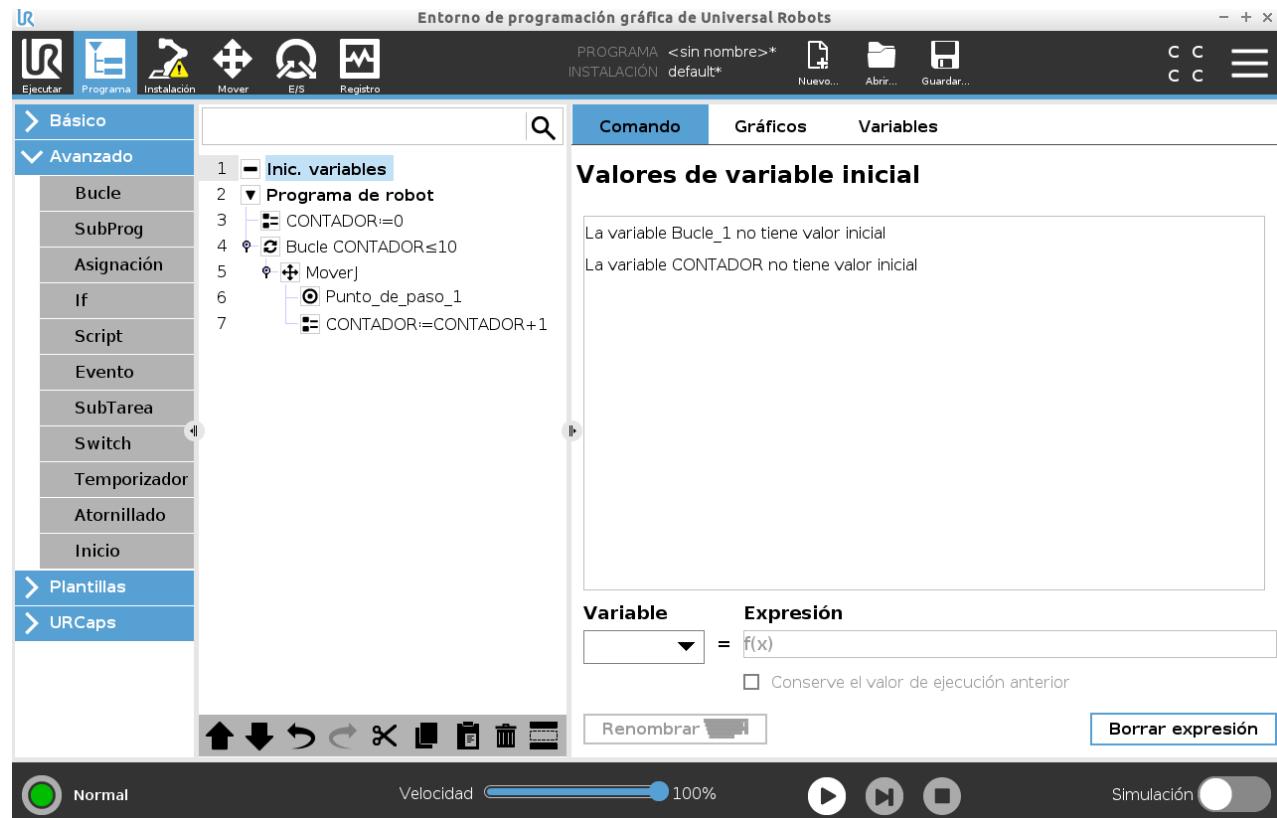
Robot Program

```
var_1 = 0  
Loop var_1 < 5  
    Pick_part  
    Place_part  
    var_1 = var_1 + 1
```



Inicialización de variables

- Inicializar variables
 - Lista de variables locales
 - Pre asignación de valores fijos
- Inicializar variables
 - Lista de variables locales
 - Pre asignación de valores fijos



Variables de instalación

Características

- Listado en pestaña de instalación
- Guardadas en fichero (*File*) separado
- Mantienen el valor tras reinicio

Funcionalidad

- Variables globales
- Accesibles desde todos los programas
- Mismas funcionalidades que las variables locales



Use una variable de instalación como contador

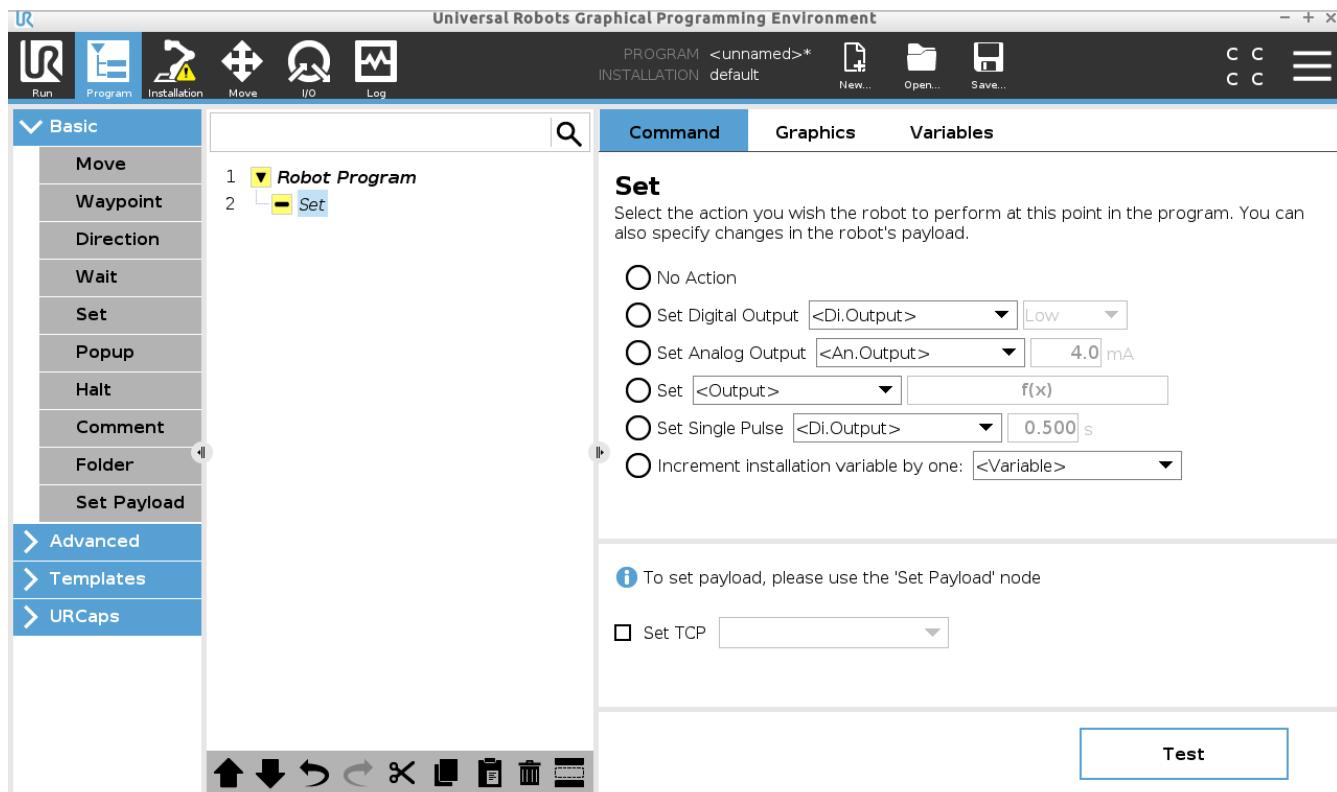
Comando Set (Ajustar)

- Incrementar una unidad la variable de instalación

Robot Program

```
i_var_1 = 0  
var_2 = operator input  
Loop i_var_1 < var_2  
    Pick_part  
    Place_part  
    Set i_var_1 = i_var_1 + 1
```

- Guardar programa de ejemplo como
inst_var_operator_input.urp



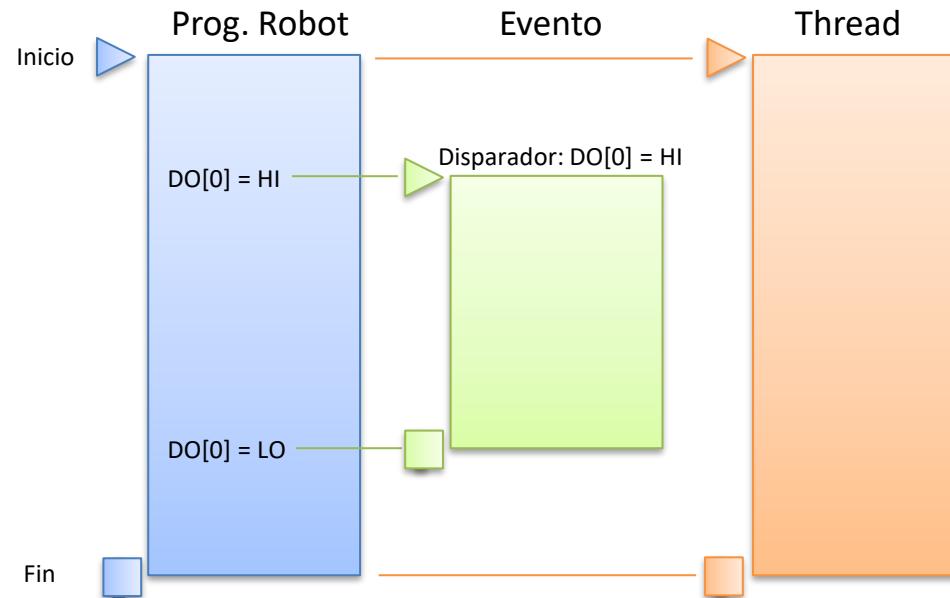
Ejercicio práctico

- Crear un programa usando variables de instalación para contar ciclos de ejecución
 - Crear una variable de instalación llamada *Contador* con valor 0
 - Crear un programa simple que mueva entre Waypoint_1 y Waypoint_2
 - Incrementar la variable *Contador* tras cada movimiento
 - Cuando el contador alcance los 3 ciclos mover a limpieza de herramienta (Waypoint_3)
 - Cuando el contador alcance los 5 ciclos, mostrar el mensaje “Cambiar bandeja en alimentador”.
 - Continuar sólo si se resetea manualmente el valor del contador
- Iniciar/Detener el programa y confirmar que el valor del contador se mantiene
- Reiniciar el robot y confirmar que el valor del contador se mantiene
 - Guardar programa de ejemplo como Lab_exe_6_1.urp

1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes
3	Comandos básicos 1	9	Modbus TCP/Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

Subprocesos/Eventos

- Thread (*Subproceso*)
 - Proceso paralelo
 - En ejecución continua
- Evento
 - Proceso paralelo
 - Iniciado por una condición
- Propósito
 - Útil para el control de I/O de comunicación con otras máquinas, para la realización de cálculos, supervisión de variables, temporizadores, etc.



Thread (Subproceso)

Configuraciones

- Loop forever (*perpetuo*)
- Seguir ejecución de prog.

```
Robot Program
MoveJ
    Waypoint_1
    Waypoint_2
IF DI[0] = True
    HALT
```

```
Robot Program
MoveJ
    Waypoint_1
    Waypoint_2
```

```
Thread_1
IF DI[0] = True
    HALT
WAIT 0.01
```

The screenshot shows the UR-Workshop software interface with the following details:

- Left Panel (Toolbox):**
 - Robot Program:** Contains MoveJ, Waypoint_1, Waypoint_2, IF DI[0] = True, and HALT.
 - Robot Program:** Contains MoveJ, Waypoint_1, Waypoint_2.
 - Thread_1:** Contains IF DI[0] = True, HALT, and WAIT 0.01.
- Center Panel (Main Area):**
 - Toolbars:** Basic, Advanced, Command, Graphics, Variables.
 - Robot Program Tree:**
 - Robot Program
 - <empty>
 - Thread_1
 - <empty>
 - Bottom Buttons:** Up, Down, Left, Right, Undo, Redo, Delete, Save, Open.
 - Status Bar:** Loops Forever (checked), Track program execution (unchecked).
- Right Panel (Description):**
 - Thread:** A thread is a parallel program that runs along with the main program. A thread can perform I/O, wait for signals, and set variables.
 - Useful for controlling other machines while the robot is running.

- Guardar programa de ejemplo como thread.urp

Evento

Activación

- Define la condición para ejecutar el evento
- Proceso paralelo

Robot Program

MoveJ

 Waypoint_1

 Waypoint_2

Event DI[0] = True

 HALT

1 Robot Program
2 <empty>
3 Event
4 <empty>

Event

An Event is similar to an Interrupt, however, in an event the execution of the main program continues while the event code is being executed. While the event is being executed, new events will have no effect.

Depending on the state of the given sensor input or program variable, the following lines will be executed.

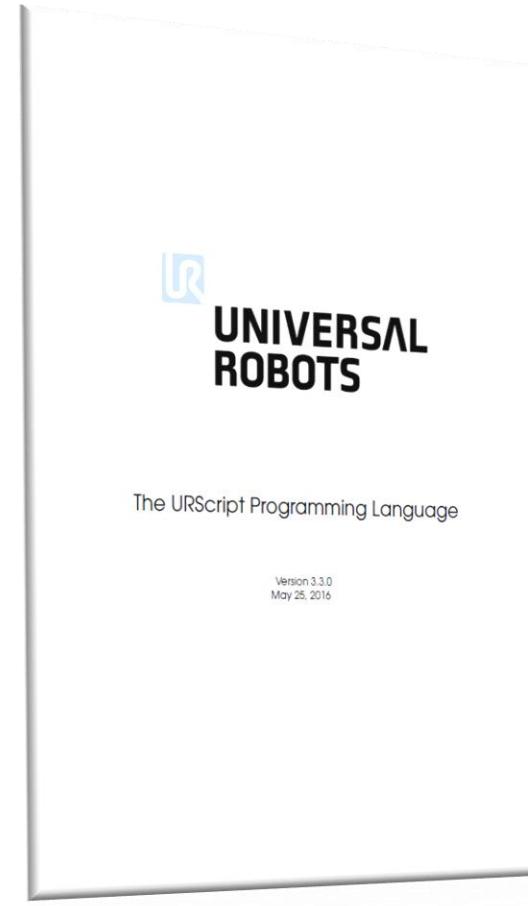
```
f(x)
```

↑ ↓ ↶ ↷ ✘ 🗑 🗑 🗑

Guardar programa de ejemplo
como event.urp

¿Qué es el script?

- Introducción a URScript
 - Una descripción más detallada se abordará en el Entrenamiento Avanzado
- URScript
 - Lenguaje script de alto nivel desarrollado por UR
 - Solo disponible en Inglés
 - Similar al lenguaje script Python
 - El manual Script contiene las definiciones de todos los códigos script disponibles



Cómo usar script

Editor de expresiones

- Aparecen listado los script más comunes
- Programa de ejemplo con *force()*
- Valor devuelto: Fuerza en el TCP

Robot Program

MoveL

Waypoint_1

IF force() < 30

Waypoint_2

The screenshot shows the Universal Robots Script Editor interface. On the left, there's a sidebar with categories like 'Basic' and 'Advanced'. Under 'Advanced', 'Loop' and 'SubProg' are collapsed, while 'Event' is expanded, showing its sub-functions. The main area displays a list of functions, many of which are preceded by a small blue 'UR' icon. The list includes:

- <Function>
- pose_add(<pose1>,<pose2>)
- pose_sub(<pose_to>,<pose_from>)
- pose_trans(<pose_from>,<pose_to>)
- pose_inv(<pose>)
- interpolate_pose(<pose_from>,<pose_to>,<alpha>)
- pose_dist(<pose_from>,<pose_to>)
- get_actual_tcp_pose()
- get_actual_joint_positions()
- get_inverse_kin(<pose>)
- get_target_tcp_pose()
- get_target_tcp_speed()
- get_target_joint_positions()
- get_target_joint_speeds()
- force()
- get_tcp_force()
- integer_to_binary_list(<integer>)
- <Function>

To the right of the list is a search bar with a magnifying glass icon. Below the search bar are tabs for 'Command', 'Graphics', and 'Variables'. The 'Command' tab is selected. Under the 'Event' heading, there's a detailed description of what an event does and how it differs from an interrupt. It also mentions that depending on the state of sensor inputs or program variables, certain lines will be executed. A large, empty rectangular area is provided for writing code. At the bottom right is a numeric keypad with standard arithmetic operators (+, -, *, /, =, <, >) and logical operators (or, xor, not). There are also buttons for 'Esc', 'Backspace', 'Submit', and a decimal point (.).

Comando Código de script

Línea

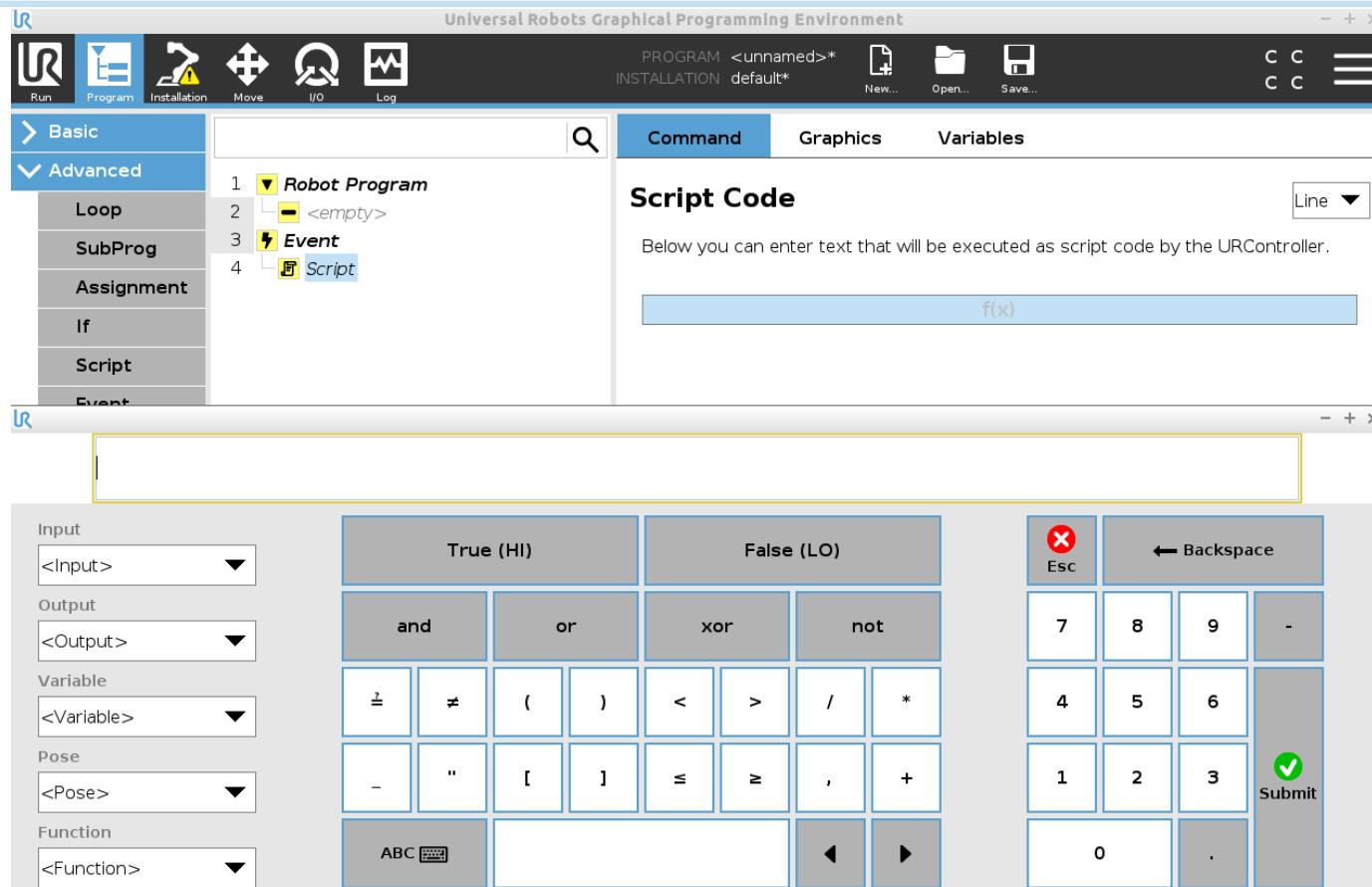
- Insertar un comando script

File (Archivo)

- Llama a un archivo que contiene múltiples códigos script

Robot Program

```
set.digital_out(0,True)
WAIT 0.5
set.digital_out(0,False)
WAIT 0.5
```



Comando Switch (*Interruptor*)

Switch (*Interruptor*) de caso

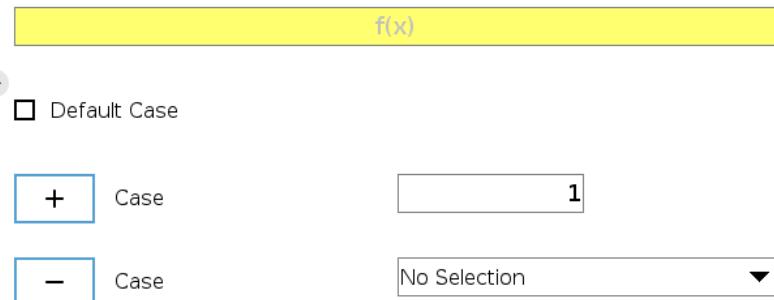
- El editor de expresiones se usa para definir la condición
- Para añadir un caso utilizar “+” y “-”
- Un caso predeterminado (*default*) se puede seleccionar de ser necesario

1 Robot Program
2 Switch ...

Switch

You can use a switch statement to control the flow of your program. These can replace complex **If...Else If** statements and can test a range of values for your expression

Switch



Comando Interruptor

Número de caso

- Selecciona número de caso

Robot Program

```
Select_no:='Enter a number  
between 1 to 3'
```

```
Switch Select_no
```

```
Case 1
```

```
  Popup
```

```
Case 2
```

```
  Popup
```

```
Case 3
```

```
  Popup
```

```
Default Case
```

```
  Popup
```

```
1 ▼ Robot Program
2 i_var_1:='select'
3 ↴ Switch i_var_1
4 ↴ Case 1
5    Popup: 1
6 ↴ Case 2
7    Popup: 2
```

Switch

You can use a switch statement to control the flow of your program. These can replace complex **If...Else If** statements and can test a range of values for your expression

Switch

```
i_var_1
```

□ Default Case

+ Case

```
3
```

- Case

```
Case 1
```



1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes / Wizard
3	Comandos básicos 1	9	Modbus TCP/Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

Wizards (Asistentes)



Command

Graphics

Variables

Pallet (Palé)

- Función de paletizado
- Patrones

- 1 ▾ Robot Program
- 2 ⚡ Force
- 3 ⚡ Loop
- 4 ↗ sync()

Force (Fuerza)

- Función de fuerza

Configuraciones

- Tipo de fuerza
- Valor de fuerza
- Dirección de la fuerza

Prueba sencilla

- Prueba de aprendizaje (Test

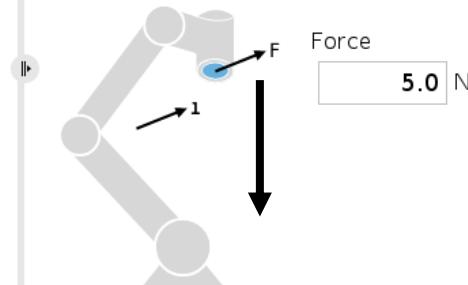
Force

Simple

The program part under this force command will be run in force mode. In force mode, the robot will be free in the direction of the selected feature to apply the specified force.

Feature

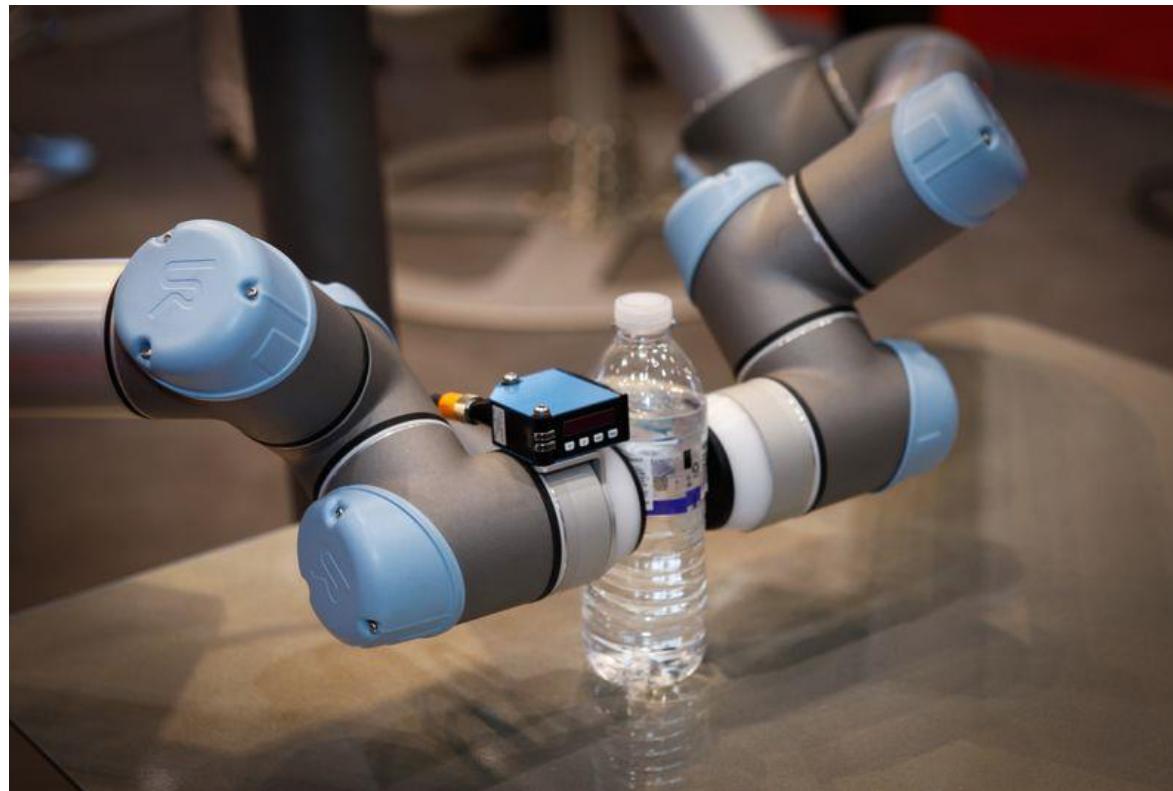
Tool



Force

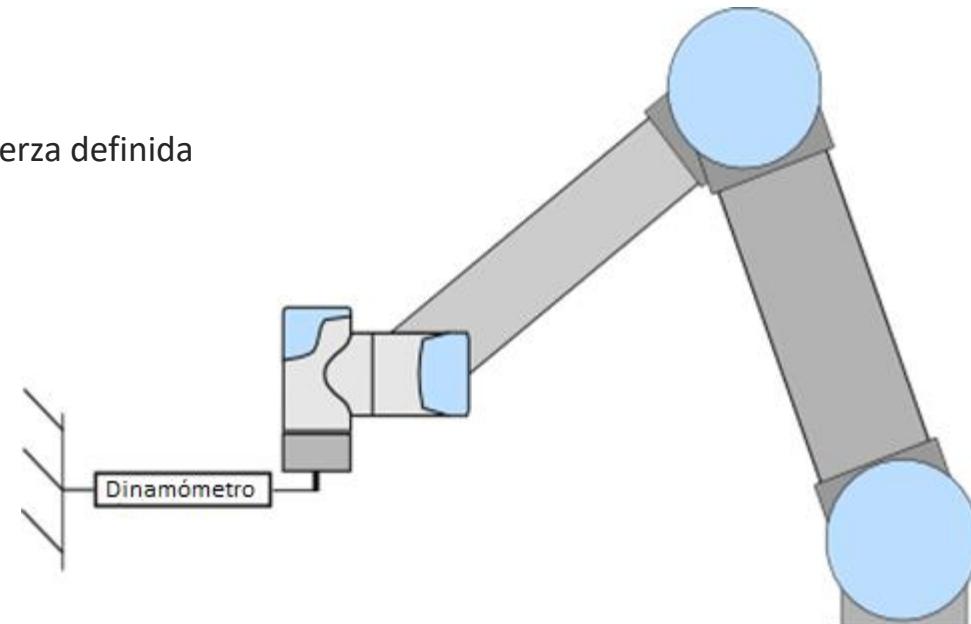
5.0 N

Control de fuerza



Asistente Fuerza

- Características
 - Interacción con el entorno
 - Ajuste de posición para lograr la fuerza definida



- Especificaciones
 - Precisión fuerza $\pm 10 \text{ N}$
 - Precisión torque $\pm 5 \text{ Nm}$
 - Precisión posición $\pm 5 \text{ mm}$
 - Precisión orientación $\pm 0.5^\circ$

Asistente Pallet (Palé)

The screenshot shows the Universal Robots Assistant interface. On the left, there is a tree view of a robot program structure:

- 1 Robot Program
- 2 Pallet_1
- 3 Patterns
- 4 <empty>
- 5 Layers
- 6 At Each Item
- 7 <empty>

At the bottom of the interface is a toolbar with various icons for navigation and file operations.

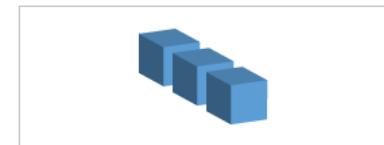
Command

Graphics

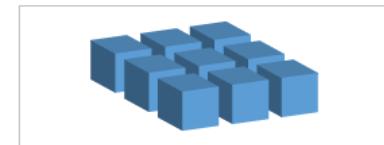
Variables

Patterns

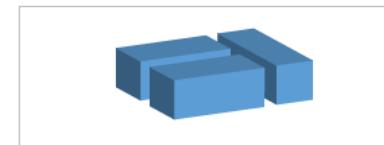
A pattern of items in a sequence. There are three different types of patterns:



Line



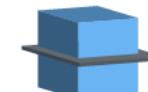
Grid



Irregular

Separator

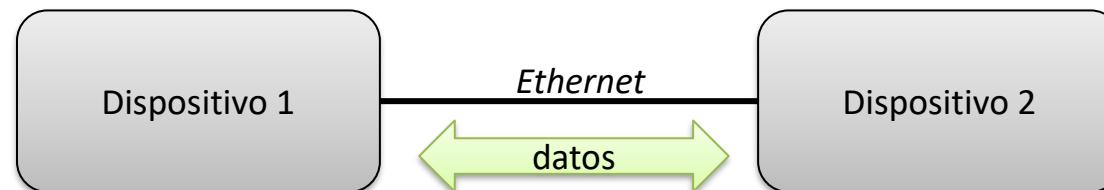
Separator Height



1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes
3	Comandos básicos 1	9	Modbus TCP/Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

¿Qué es ModBus TCP?

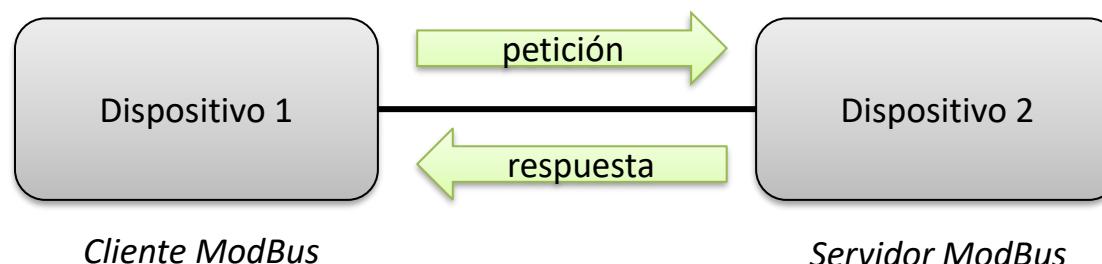
- Modbus TCP
 - Protocolo de comunicación basado en Ethernet
- Protocolo de comunicación
 - Un protocolo es un lenguaje común mediante el cual dos dispositivos pueden comunicarse
 - Posibilita la transmisión de datos entre los dispositivos



- Estructura Cliente / Servidor

Cliente / Servidor

- **Servidor (Esclavo)**
 - Uno de los dispositivos actúa como Servidor
 - A la escucha de peticiones desde el Cliente
- **Cliente (Maestro)**
 - Otro dispositivo en la red actúa como Cliente
 - Envía peticiones al Servidor



- Cada dispositivo debe tener configurada una dirección-IP única

Tipos de datos

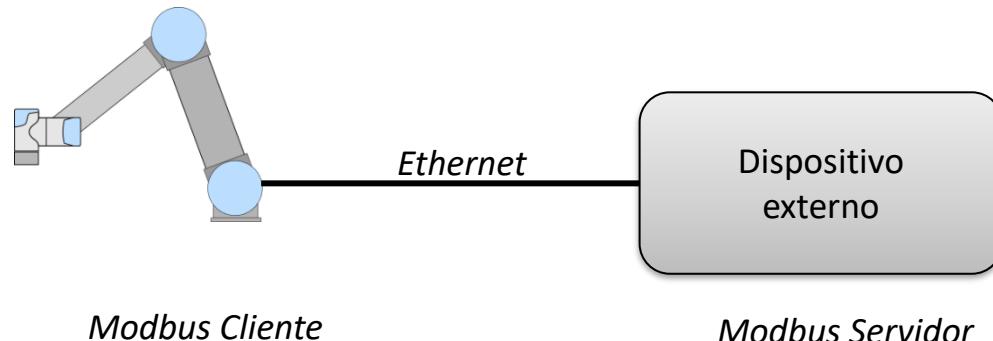
- Tipos de datos disponibles en Modbus TCP

Tipo de dato	Valor	Rango de direcciones
Entradas digitales	ON/OFF	<i>Consultar la documentación del fabricante del dispositivo Servidor</i>
Salidas digitales	ON/OFF	
Entradas de registro	16 bits	
Salidas de registro	16 bits	

- Rango de direcciones
 - Cada señal digital y cada registro poseen una única dirección
 - La dirección se especifica *siempre* en la documentación suministrada por el fabricante

Modbus TCP

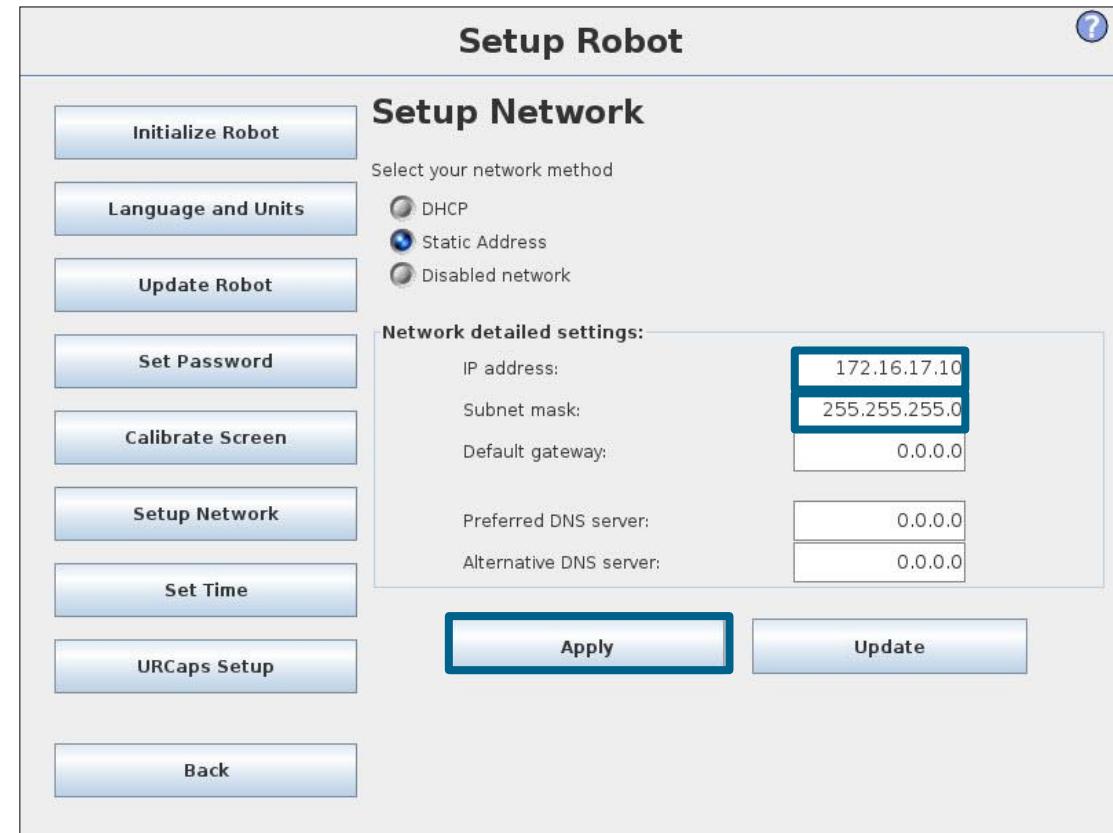
- Ejemplo
 - Usar el robot como Cliente y conectar a un Servidor



Configuración de red

Ajustes

- Método Dirección estática
- Introducir la dirección-IP del robot
- Introducir la máscara de subred
- Aplicar (*Apply*) para guardar la configuración



- Consejo: Use el botón "Actualizar" para hacer ping a otro dispositivo

Servidor

Phoenix Contact ILB ETH 24 DI16 DIO16 2TX

- 16 salidas digitales
- 16 entradas digitales
- 2-puertos de conexión



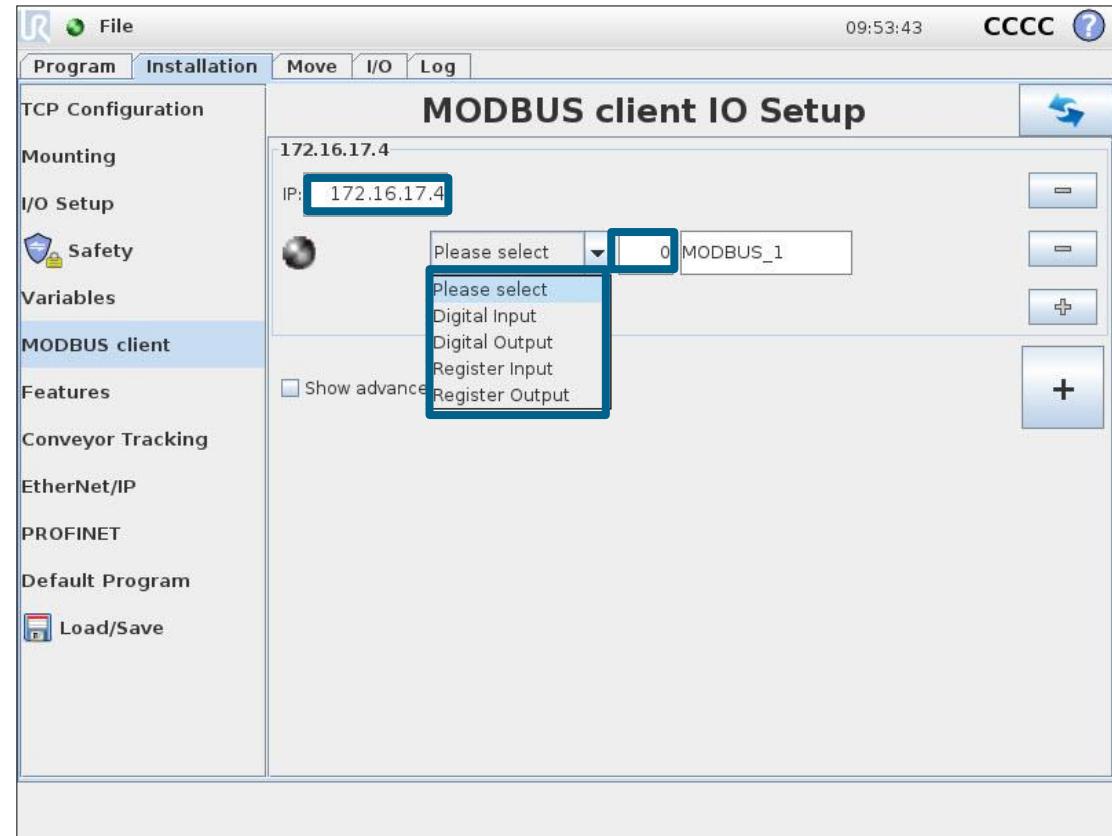
	Modbus Register Table (16-Bit Words)	Modbus Input Discretes Table (Bits)	Modbus Coil Table	Access	Function
Process data	0	0-15	—	Read only	Digital inputs (DIO)
	1	16-32	—	Read only	Digital inputs (DI)
	2	—	0-15	Read/write	Digital outputs
	3	—	—	Read only	Reserved

Diagnostics	4	—	—	Read only	Status register
	5	—	—	Read only	I/O diagnostic register
	6	—	—	Read only	NetFail reason
	7	—	—	Read only	IBS diagnostic register (for compatibility with FL IL 24 BK)
	8	—	—	Read only	IBS para register (for compatibility with FL IL 24 BK)

Special register	1280	—	—	Read/write	Modbus timeout connection monitoring
	2000	—	—	Read/write	Process data watchdog timeout
	2002	—	—	Read/write	Fault response mode
	2004	—	—	Read/write	NetFail test (same value as register 6)
	2006	—	—	Read/write	Command register

Configuración del servidor

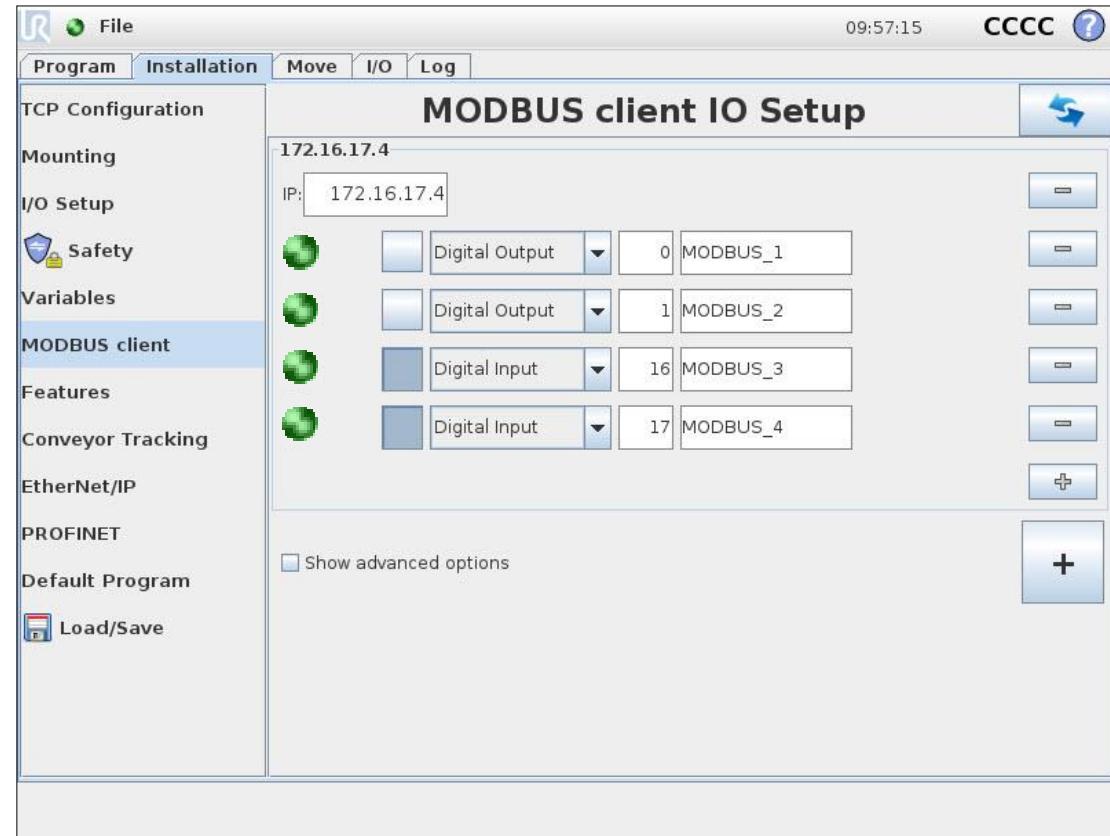
- Configurar dispositivo
 - Añadir nuevo dispositivo
 - Introducir la dirección-IP del dispositivo (Servidor)
- Configurar señales
 - Añadir señal en dispositivo
 - Definir tipo de dato
 - Establecer su dirección
 - Definir nombre de la señal



Configuración del servidor

- Configuración
 - 2 entradas digitales
 - 2 salidas digitales
 - Guardar Instalación
- Monitoreo
 - El estado de la señales aparece en el “tab” de I/O
- Estado de conexión

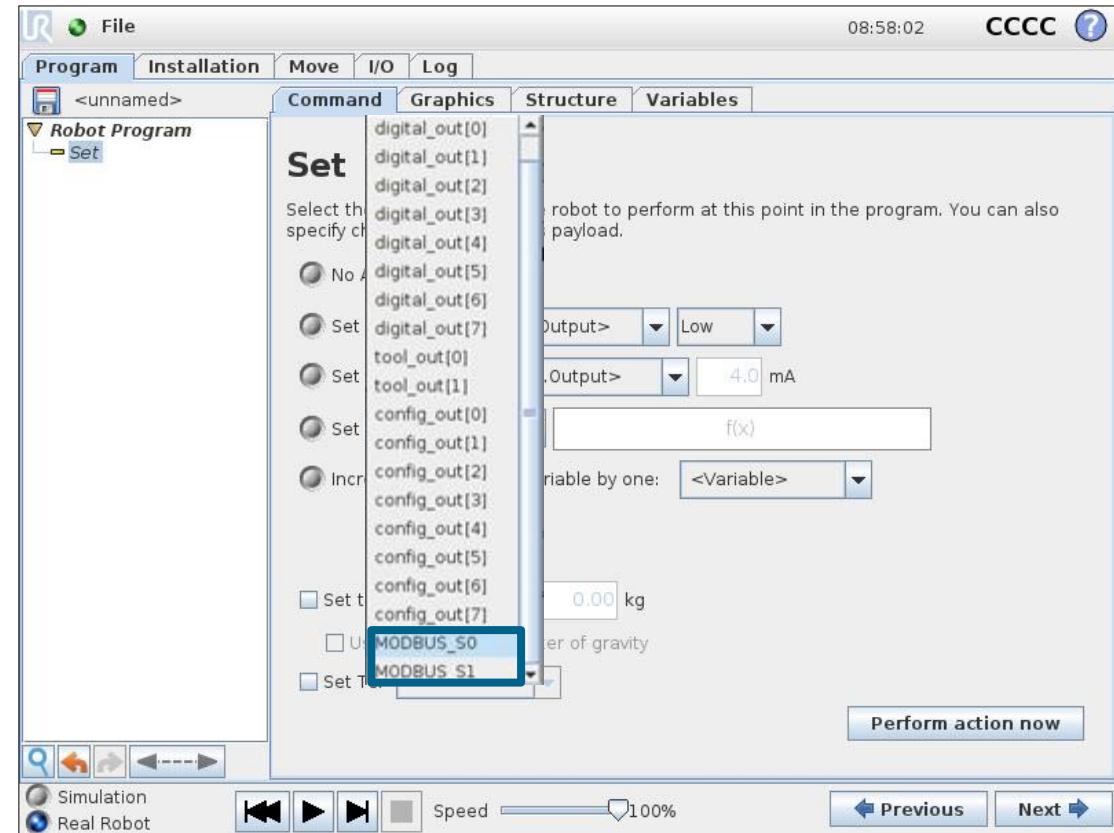
	Status
	Conexión ok
	Aviso frecuencia de refresco
	Sin conexión
	Código de excepción



Uso de las señales de Modbus en programa

- Uso de las señales
 - Misma funcionalidad que señales digitales normales

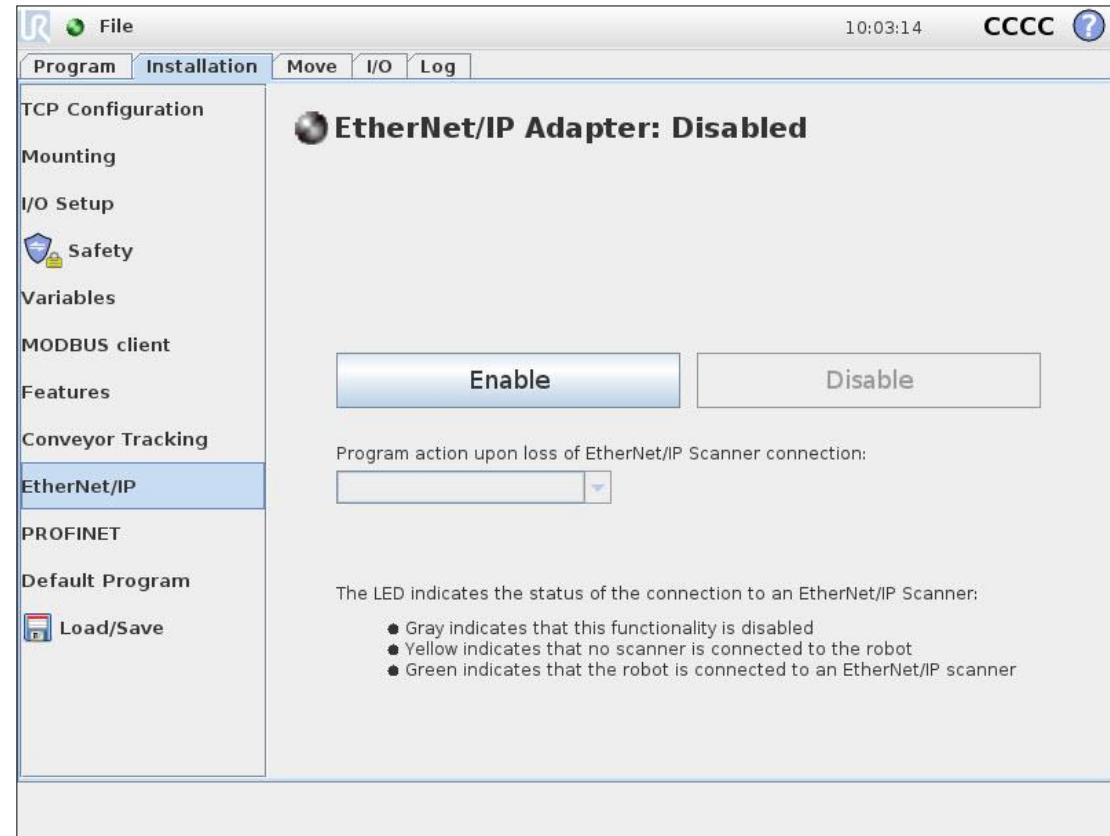
```
Robot Program
MoveL
    Waypoint_1
    Set MODBUS_1 = True
    Waypoint_2
    Waypoint_3
    Waypoint_4
    Set MODBUS_1 = False
```



- Guardar programa de ejemplo como modbus.urp

Ethernet IP

- Ethernet IP
 - Deshabilitado de forma predeterminada
- Disponible desde SW 3.2
 - No incluido en la Capacitación Básica



Profinet

- Profinet
 - Deshabilitado de forma predeterminada
- Disponible desde SW 3.3
 - No incluido en la Capacitación Básica



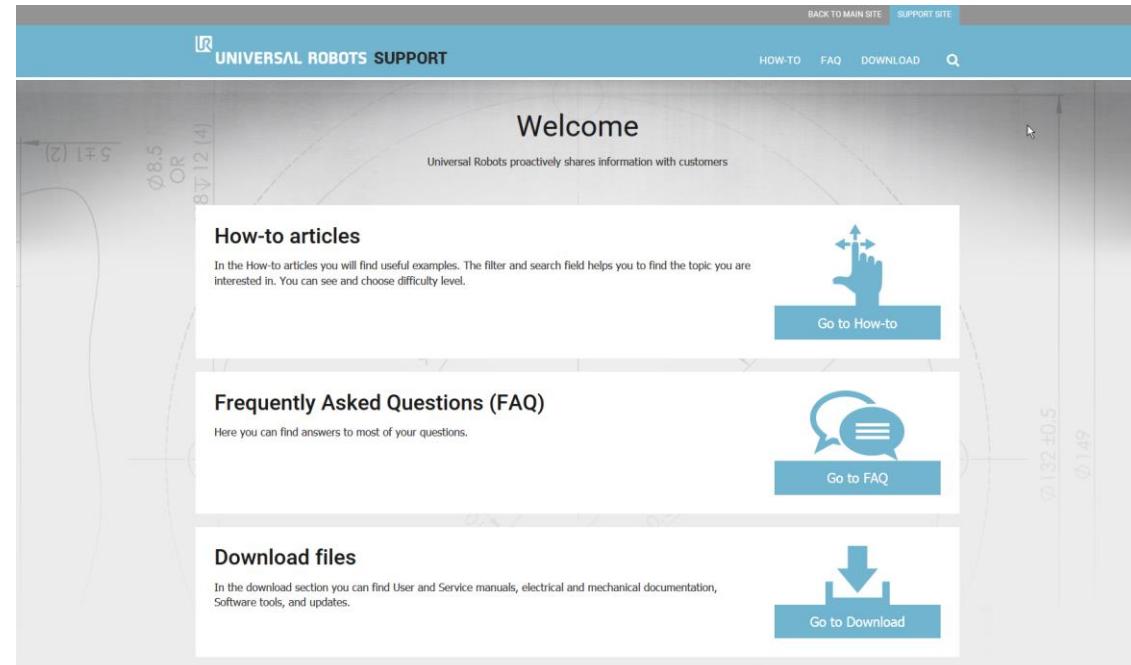
Ejercicio práctico 2

- Crear un programa para probar la conexión
 - Conectar la salida remota via MODBUS
 - Verificar que la entrada correspondiente en el robot se activa
 - Desconectar la salida remota via MODBUS
 - Verificar que la entrada correspondiente en el robot se desactiva
 - Mostrar un aviso con los resultados de la prueba
- Establecer un tiempo máximo
 - Usar un subprocesso para limitar a un tiempo máximo de 5 segundos la duración de la prueba
- Guardar programa de ejemplo como Lab_exe_9_2.urp

1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes
3	Comandos básicos 1	9	Modbus TCP/Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

Descargas y consejos

- *FAQ* - Preguntas frecuentes
- *How to's* - ¿Cómo?
- *Download* - Descargas
 - Magic files
 - Software/Firmware
 - URSim
 - Log reader
 - Planos CAD
 - Manuales



www.universal-robots.com/support

Magic files

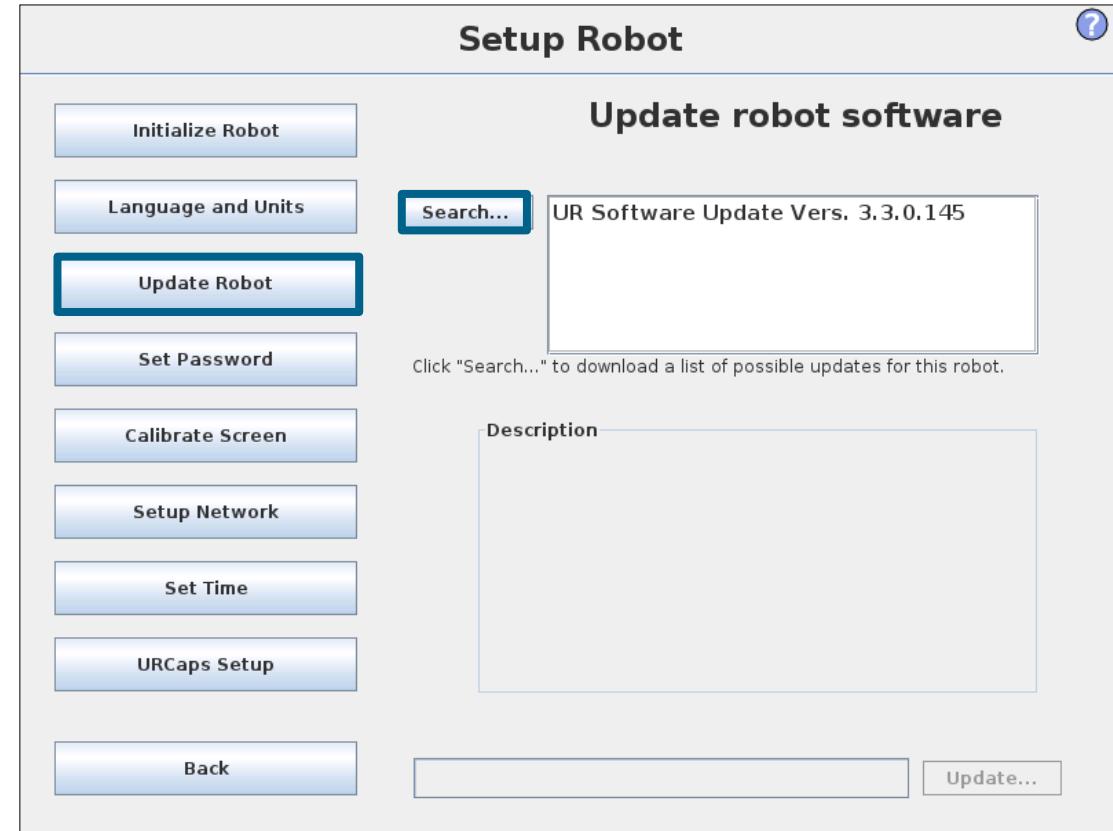
- Copia de seguridad sencilla
 - Backup programas
 - Backup histórico de avisos log_history
 - Backup archivos de configuración
- Otras funciones
 - Carga de programas
 - Captura de pantalla del GUI
- ¿Cómo?
 - Descargar la Magic file desde la página de soporte
 - Copiar el archivo en una memoria USB
 - Insertar la memoria USB en el TP » aparece el aviso ! USB ! rojo
 - Esperar el aviso <- USB verde » copia completada



Actualizar el Software

- Procedimiento

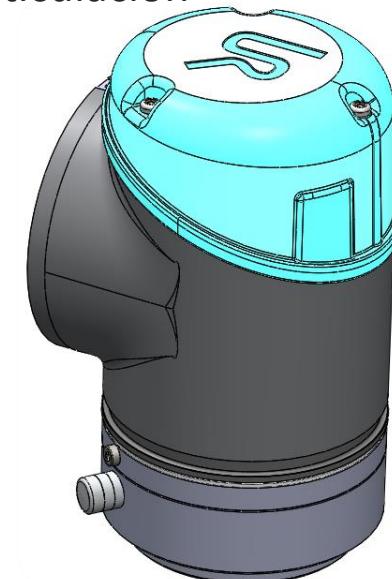
- Descargar última versión desde la página de soporte
- Copiar archivo en la memoria USB
- Insertar USB en TP
- Ir a Config. robot → Actualizar robot
- Pulse el botón "Buscar..." y seleccione la actualización
- Pulse el botón "Actualizar"
- NOTA:
El robot se apagara tras actualizarse



- **IMPORTANTE: ¡ REALIZAR SIEMPRE UNA COPIA COMPLETA DE ANTES DE ACTUALIZAR SOFTWARE !**

Firmware

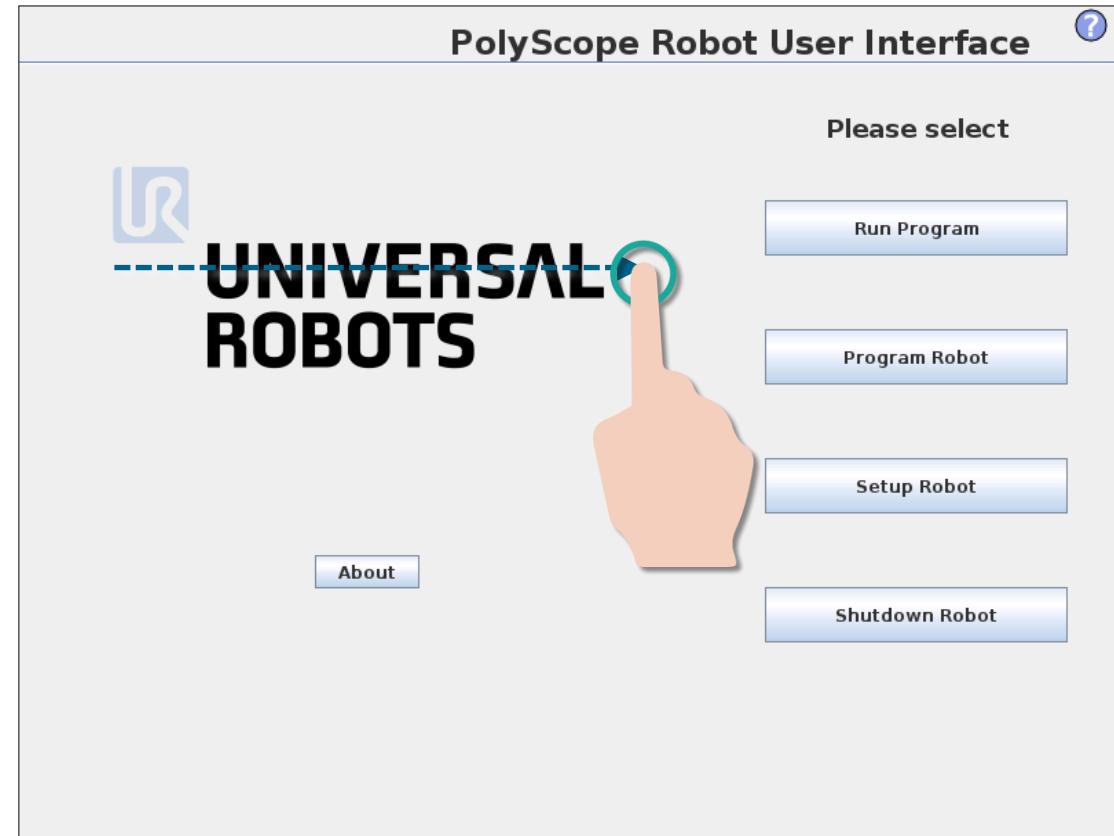
- El Firmware es el Software que se encuentra en cada articulación
 - El Firmware controla la articulación
 - Durante la actualización del Software, el Firmware se copia automáticamente a la memoria del robot –
No requiere ninguna descarga adicional
- Si es necesario puede ser actualizada
 - En versiones anteriores a SW3.1:
debe hacerse manualmente
 - A partir de la versión SW3.1:
se actualiza automáticamente la primera vez
tras actualizar el Software



Actualización de Firmware (en versiones anteriores a SW3.1)

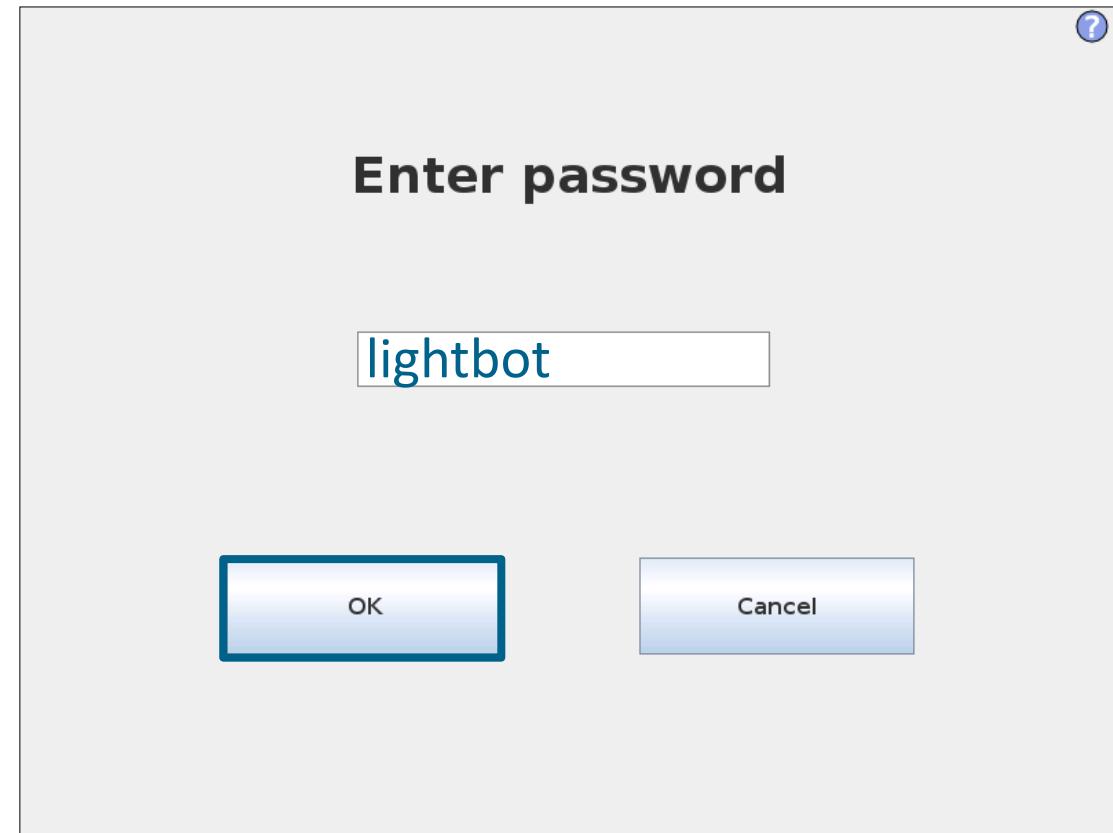
- Procedimiento

- Arrastre su dedo sobre la palabra UNIVERSAL en la pantalla de bienvenida



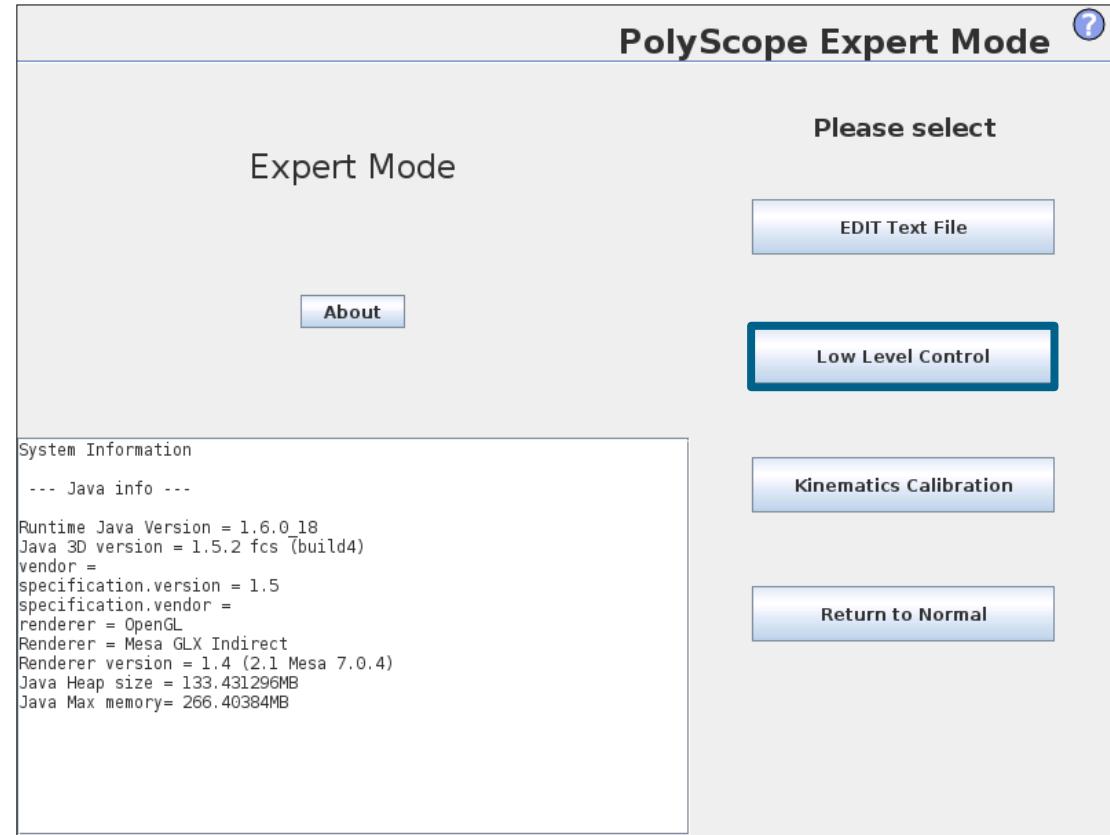
Actualización de Firmware (en versiones anteriores a SW3.1)

- Procedimiento
 - Introducir contraseña *lightbot*
 - Pulsar OK para acceder al Modo Experto



Actualización de Firmware (en versiones anteriores a SW3.1)

- Procedimiento
 - Pulsar "Low Level Control"



Actualización de Firmware (en versiones anteriores a SW3.1)

- Procedimiento

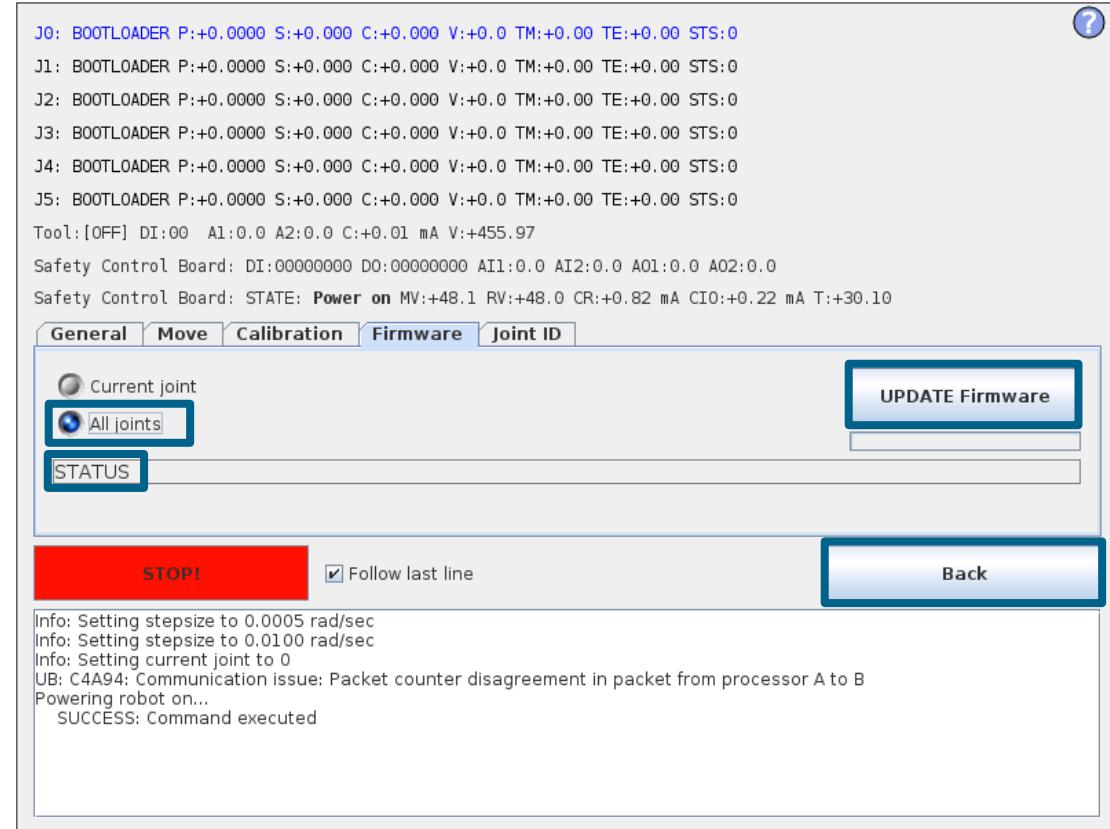
- Pestaña General

- Pulsar "Turn power on"
 - Verificar que el estado de todas las juntas cambia a: BOOTLOADER

- Pestaña Firmware

- Seleccionar "All joints"
 - Pulsar "UPDATE Firmware"
 - Esperar "Firmware update complete" en la línea de STATUS

- Pulsar "Back" y "Return to Normal"



Simulador fuera de línea (*Offline*)

- URSim

- Software para programación fuera de línea
- Ejecutable sólo en sistemas operativos Linux
- Disponible para su descarga desde la página de soporte como:
 - Archivo de instalación (SO Linux)
 - Máquina Virtual (SO Windows)



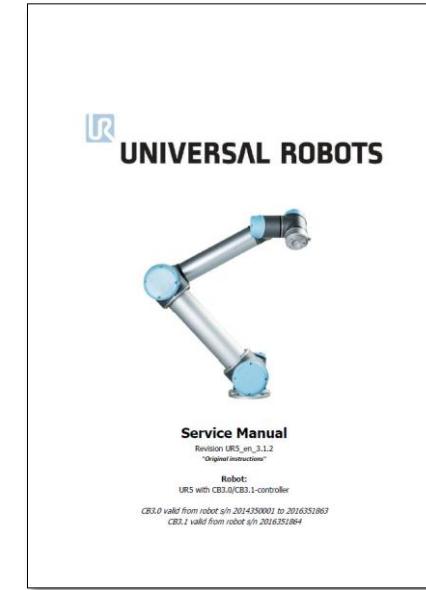
Siguiente...

Mantenimiento

Mantenimiento preventivo

- Contenido manual de mantenimiento (Service Manual)

- Mantenimiento preventivo
 - Brazo robot
 - Caja controlador
- Esquemas
- Limpieza y sustitución de las piezas
- Software
- Resolución de problemas
- Piezas de repuesto
- Embalaje del robot

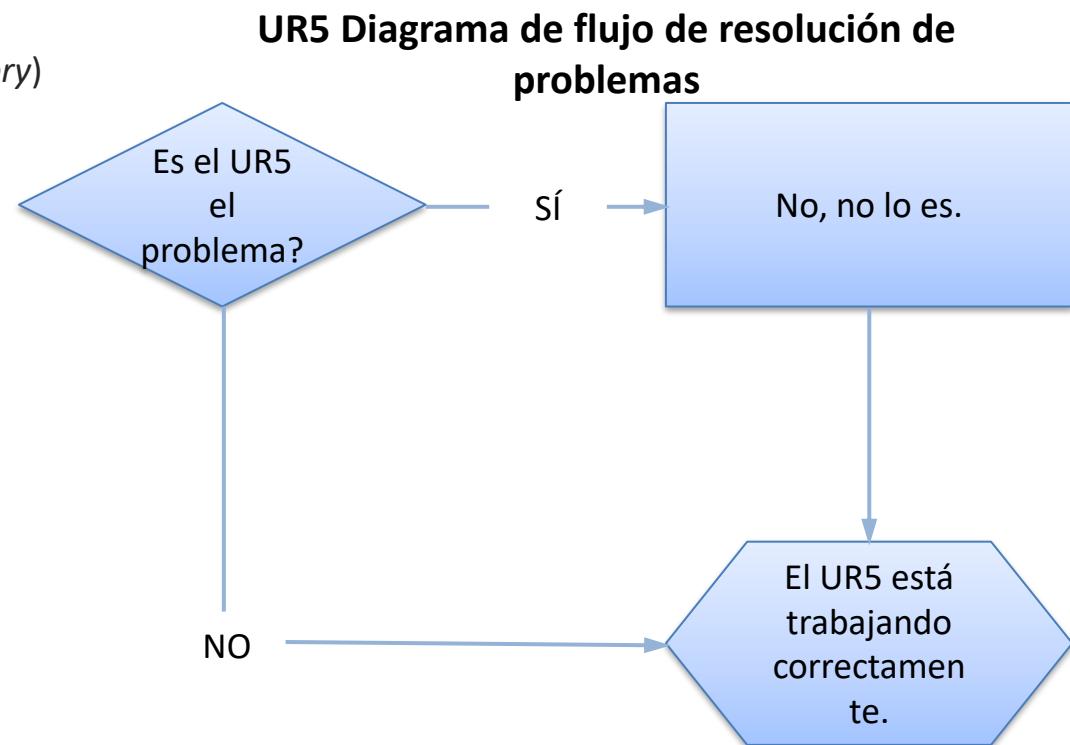


- Mantenimiento preventivo

- Descargar Manual de mantenimiento desde la página de soporte
- Revisar *Capítulo 2. Mantenimiento preventivo*

Manual de mantenimiento

- Resolución de problemas
 - Registro histórico (*log_history*)
 - Sustitución de articulación
 - Calibrado de articulación
 - Cambio ID de articulación
 - Reclamación de Garantía



Pestaña de Registro

- Lecturas
 - Medidas en controlador
- Carga de articulación
 - Estado de las articulaciones
- Registro histórico
 - Muestra información importante acerca de la salud del robot
 - Filtro Mostrar/Ocultar
 - Información
 - Avisos
 - Errores

The screenshot shows the Universal Robots software interface with the 'Log' tab selected. The top menu bar includes 'File', 'Program', 'Installation', 'Move', 'I/O', and 'Log'. The 'Log' tab is active, displaying a table of joint load data and a scrollable list of historical log entries.

Joint	Base	Shoulder	Elbow	Wrist 1	Wrist 2	Wrist 3	Power
Controller Temp.	37.8 °C						0.0 V
Main Voltage	0.0 V						0.0 V
Avg.Robot Power	0 W						0.0 V
Robot Current	0.2 A						0.0 V
I/O Current	0 A						0.0 V
Tool Current	0 mA						0.0 V

T 2014-04-01 12:38:40

```

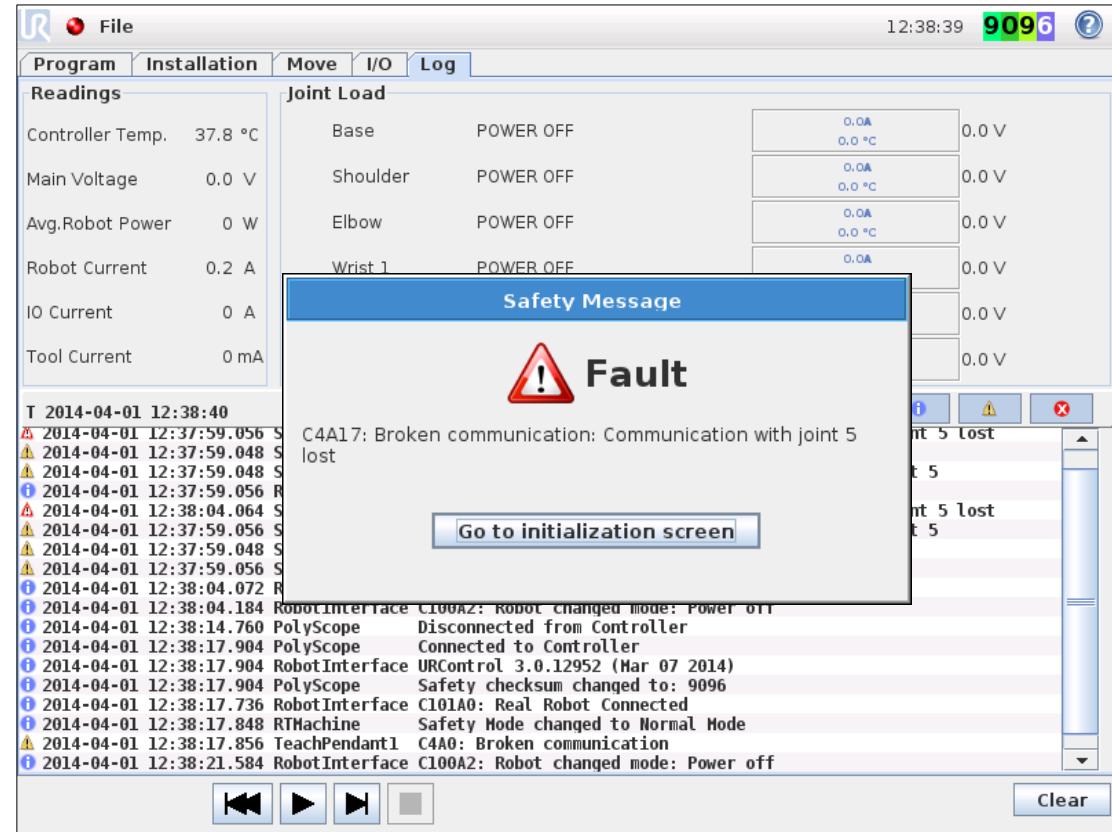
⚠ 2014-04-01 12:37:59.056 SafetyA C4A17: Broken communication: Communication with joint 5 lost
⚠ 2014-04-01 12:37:59.048 SafetyA C4A76: Broken communication: Lost package from tool
⚠ 2014-04-01 12:37:59.048 SafetyB C4A75: Broken communication: Lost package from joint 5
ℹ 2014-04-01 12:37:59.056 RTMachine Safety Mode changed to Fault Mode
⚠ 2014-04-01 12:38:04.064 SafetyB C4A17: Broken communication: Communication with joint 5 lost
⚠ 2014-04-01 12:37:59.056 SafetyA C4A75: Broken communication: Lost package from joint 5
⚠ 2014-04-01 12:37:59.048 SafetyB C4A76: Broken communication: Lost package from tool
⚠ 2014-04-01 12:37:59.056 SafetyA C4A76: Broken communication: Lost package from tool
ℹ 2014-04-01 12:38:04.072 RobotInterface C100A3: Robot changed mode: Power on
ℹ 2014-04-01 12:38:04.184 RobotInterface C100A2: Robot changed mode: Power off
ℹ 2014-04-01 12:38:14.760 PolyScope Disconnected from Controller
ℹ 2014-04-01 12:38:17.904 PolyScope Connected to Controller
ℹ 2014-04-01 12:38:17.904 RobotInterface URControl 3.0.12952 (Mar 07 2014)
ℹ 2014-04-01 12:38:17.904 PolyScope Safety checksum changed to: 9096
ℹ 2014-04-01 12:38:17.736 RobotInterface C101A0: Real Robot Connected
ℹ 2014-04-01 12:38:17.848 RTMachine Safety Mode changed to Normal Mode
⚠ 2014-04-01 12:38:17.856 TeachPendant1 C4A0: Broken communication
ℹ 2014-04-01 12:38:21.584 RobotInterface C100A2: Robot changed mode: Power off

```

Buttons at the bottom include arrows for navigating through the log and a 'Clear' button.

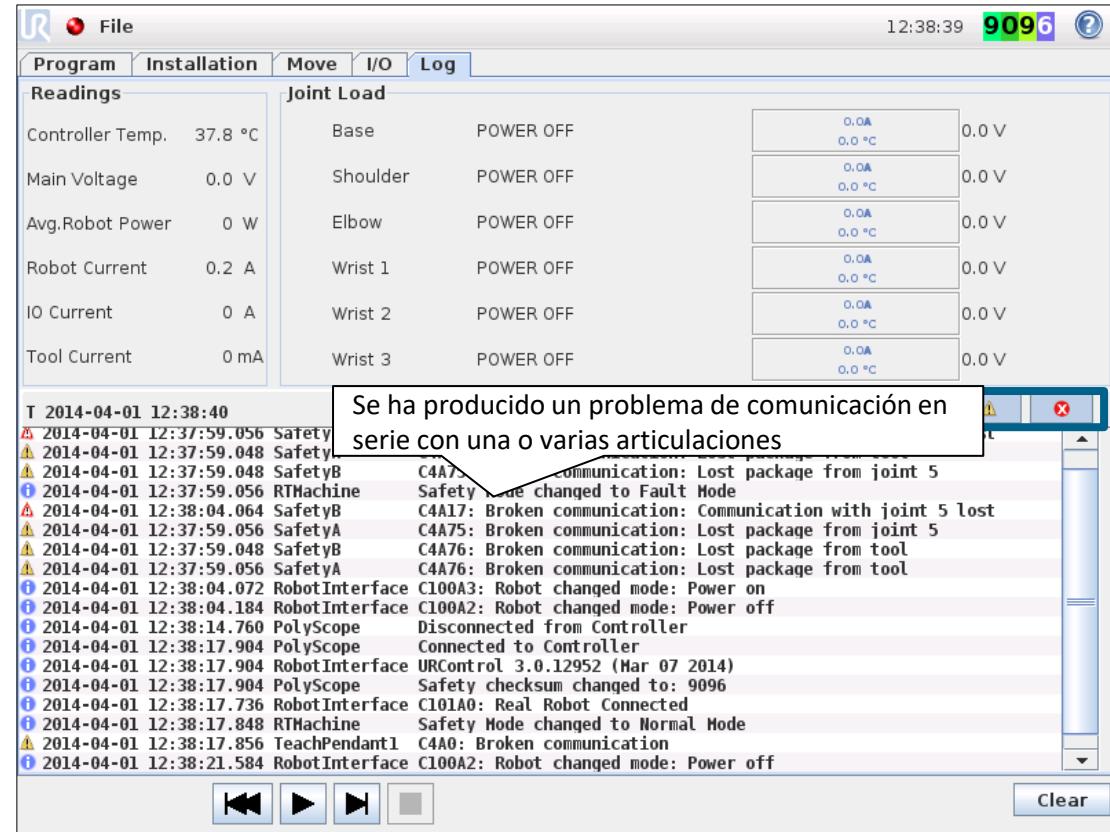
Pestaña de Registro

- Demostración de error
 - Desmontar la tapa azul en la junta Muñeca 3
 - Desconectar conector negro del cable de comunicación proveniente de la junta anterior
 - Verificar que el robot detecta el error y muestra un Fallo



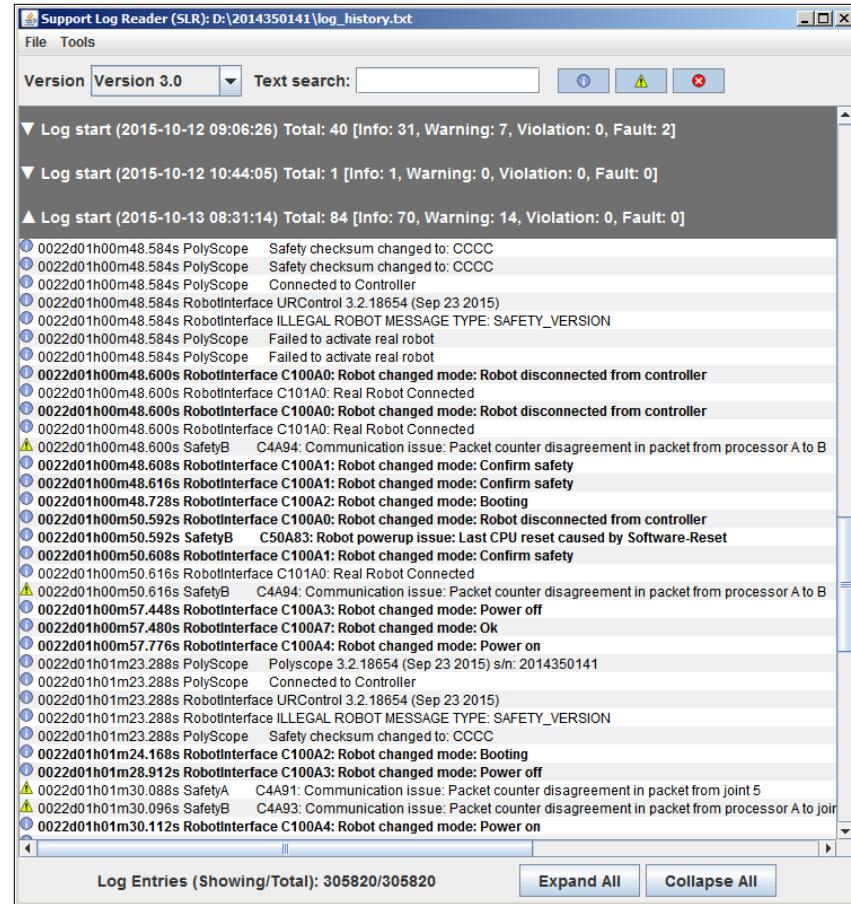
Pestaña de Registro

- Demostración de error
 - Comprobar registro
 - Mostrar/Ocultar filtrando por tipos de mensajes
 - Destacar un mensaje pulsando una vez sobre él
- Registro histórico
 - Se almacena en fichero de texto denominado: log_history.txt
 - Usar Magic file para recuperarlo



Support Log Reader (SLR)

- Support Log Reader
 - Lectura de registros históricos (*log_history*)
 - Conversión de idioma
 - Conversión a archivo-csv
 - Búsqueda de texto
 - Soporta
 - Formato de archivos CB3
 - Formato de archivos CB2
- Formato de archivos CB3
 - Conversión de idioma a Inglés
- Formato de archivos CB2
 - Se mantiene el idioma original



Siguiente...

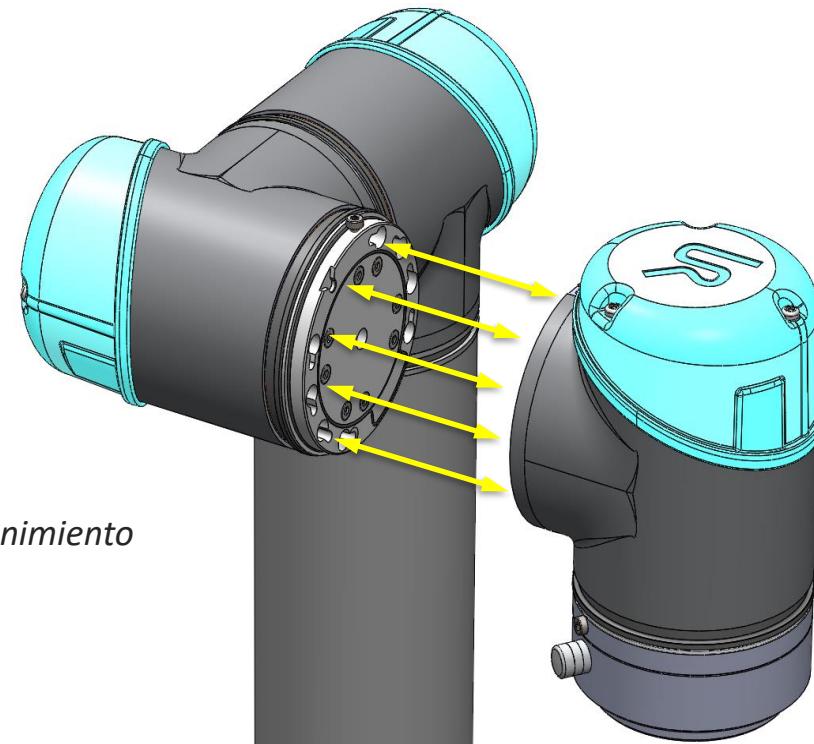
Sustitución de articulaciones

STEP 15: RECONNECT CABLES



Sustitución de juntas

- Procedimiento de sustitución de junta
 - Instrucciones disponibles en *Manual de mantenimiento*
- Cambiar la Muñeca 3
 - Herramientas necesarias
 - Llave fija 5.5 mm
 - Destornillador Torx T10
 - Destornillado dinamométrico Torx T10
 - Llave dinamométrica 5.5 mm
 - Destornillador plano pequeño
 - Seguir instrucciones en el *Manual de Mantenimiento* para separar la Muñeca 3 de la Muñeca 2
 - Montar la Muñeca 3 de nuevo



Calibración de juntas

- Calibración
 - Cada articulación tiene una posición cero
 - Esta posición se puede establecer por software
- Calibrar Muñeca 3
 - Seguir las instrucciones en el *Manual de Mantenimiento* para realizar la calibración de una junta



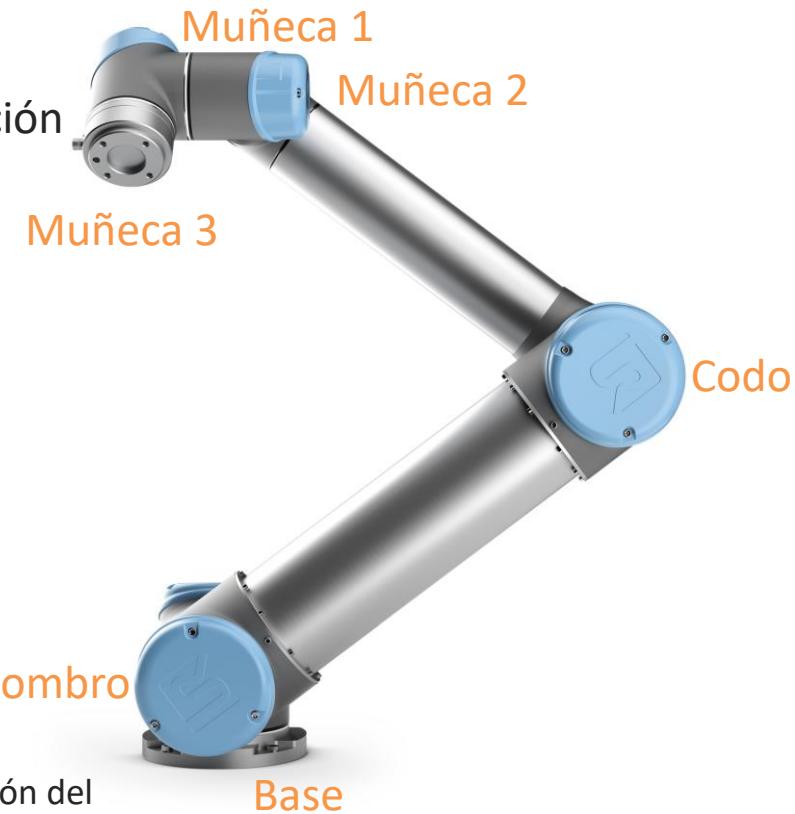
Siguiente...

Cambiar ID de juntas

Cambio del ID en las juntas

- Cada articulación posee un no. de identificación único

ID	Junta
J0	Base
J1	Hombro
J2	Codo
J3	Muñeca 1
J4	Muñeca 2
J5	Muñeca 3

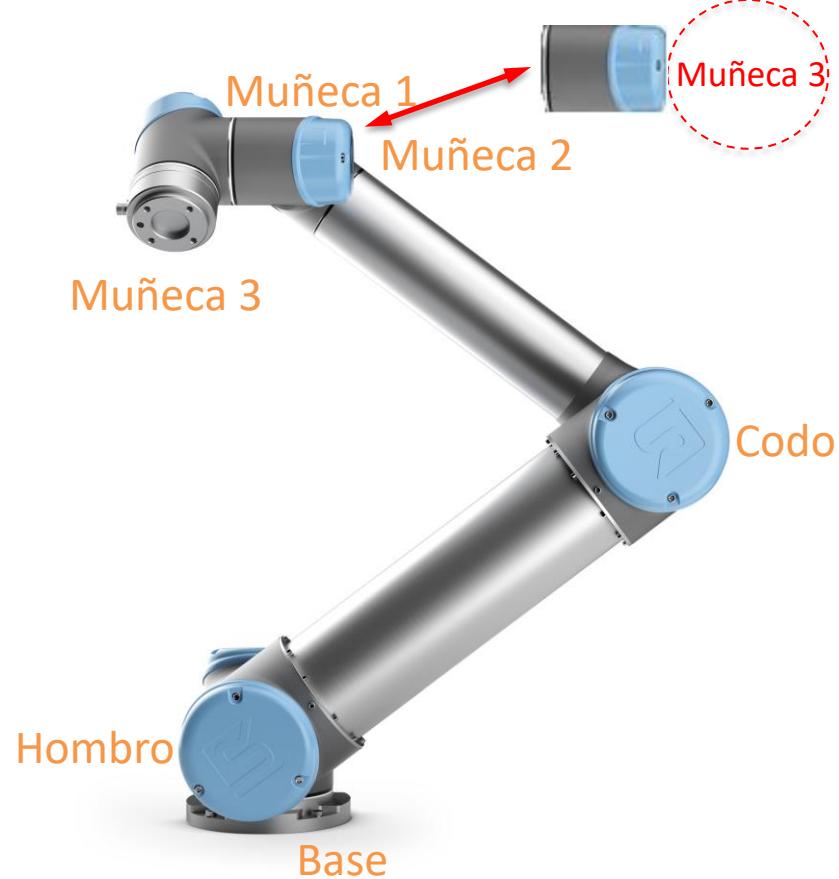


- **NO** es posible tener dos juntas con el mismo número de ID en el mismo robot

- » Habría un conflicto de IDs, provocando la disfunción del robot

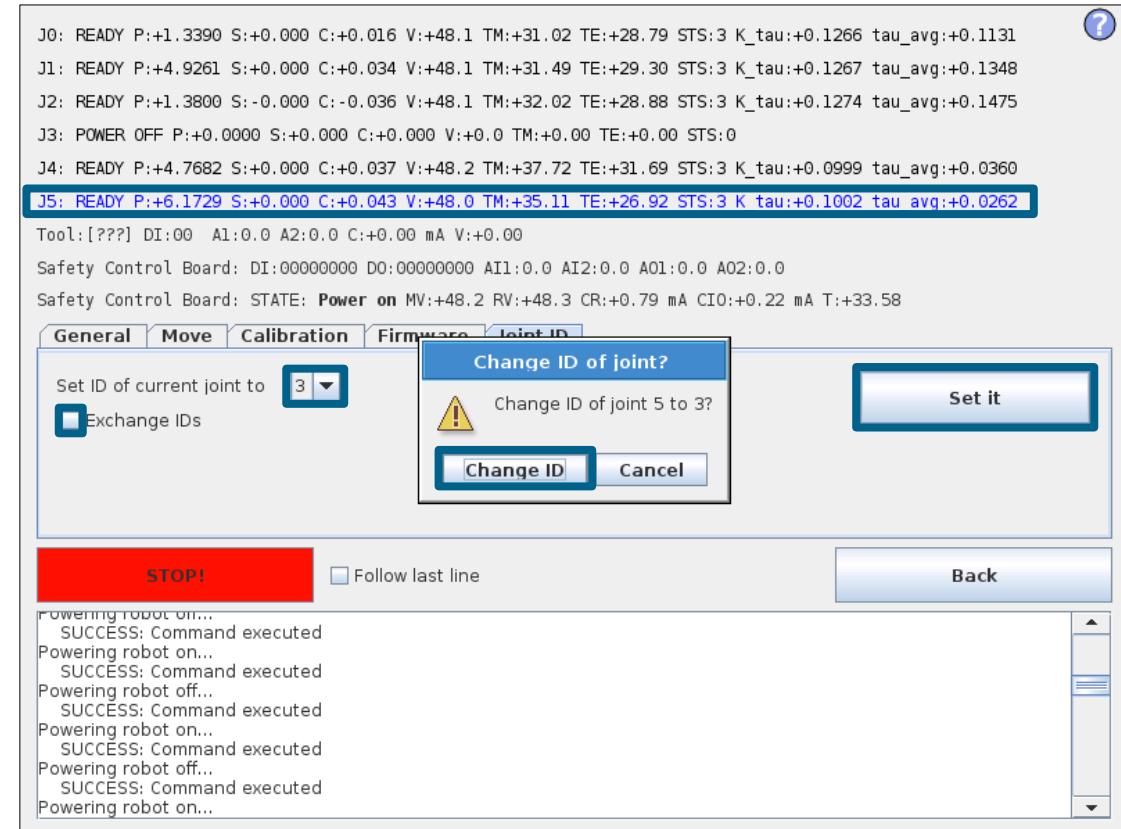
Cambio del ID en las juntas

- Ejemplo
 - La muñeca 1 (J3) debe reemplazarse
 - Disponemos de una Muñeca 3 (J5) de reserva
 - » Ocurrirá un conflicto, dado que el robot tendrá dos ID J5
- Cómo solventarlo
 - Desconectar conector de comunicación en la junta con ID correcto
 - » Ahora la Muñeca 3 no estará accesible
 - Ir a Low Level Control
 - » Cambiar el ID de la junta de recambio (J5) a J3



Cambio del ID en las juntas

- **ID de Junta**
 - Ir a “Low Level Control”
 - Ir a la ficha “General”
 - Click “Power on”
 - Click “Go to Idle”
 - Ir a la ficha “Joint ID”
 - Seleccionar J5 (el que debe cambiarse)
 - **IMPORTANTE:** desmarcar cuadro “Exchange IDs”
 - En el menú desplegable, seleccionar ID no. 3 y pulsar “Set it”



- Una guía completa para el cambio del ID de junta se encuentra en el Manual de mantenimiento

Reclamación de garantía

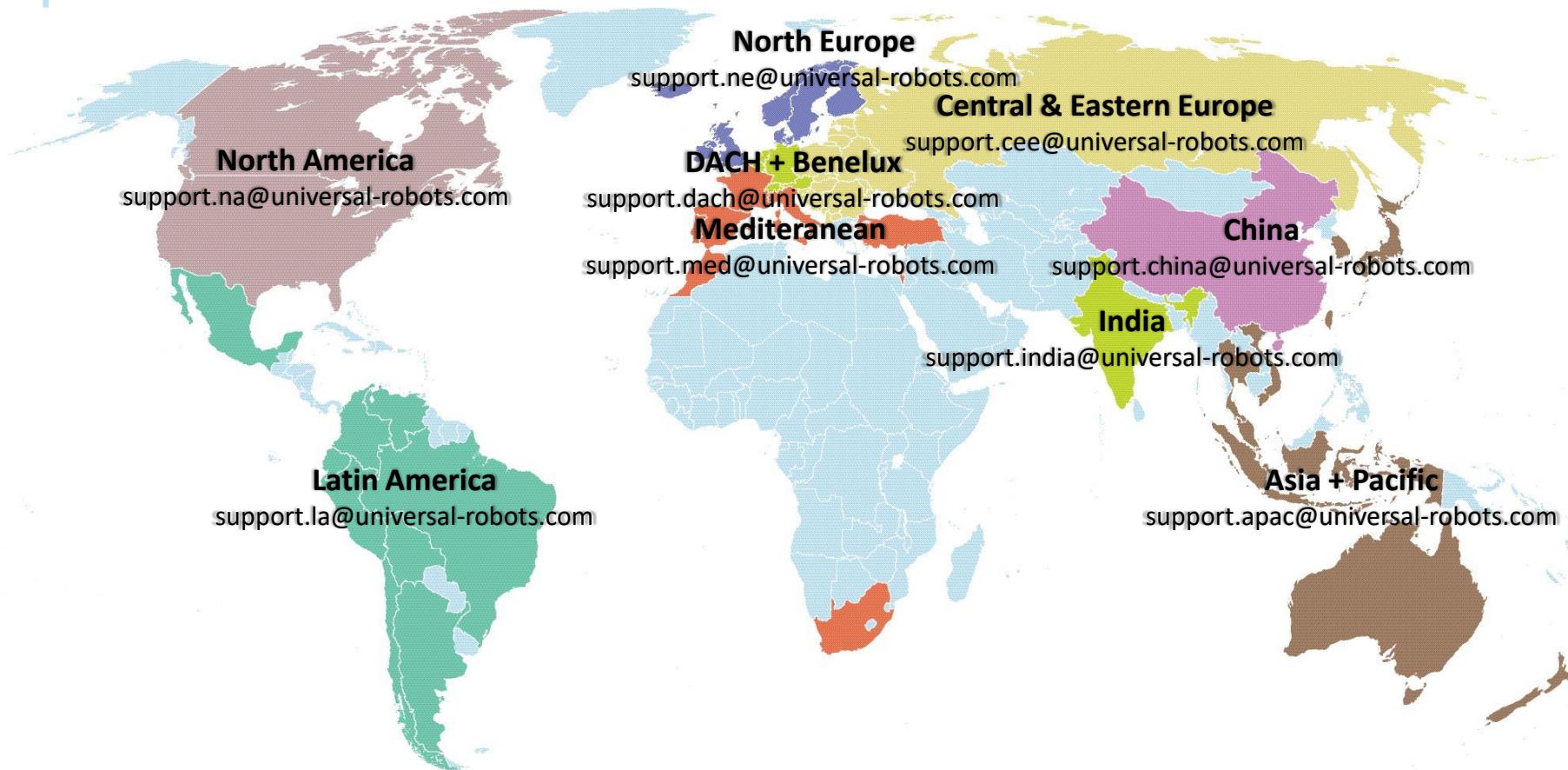
- Contactar por email con Soporte Regional de UR

LATAM

support.la@universal-robots.com

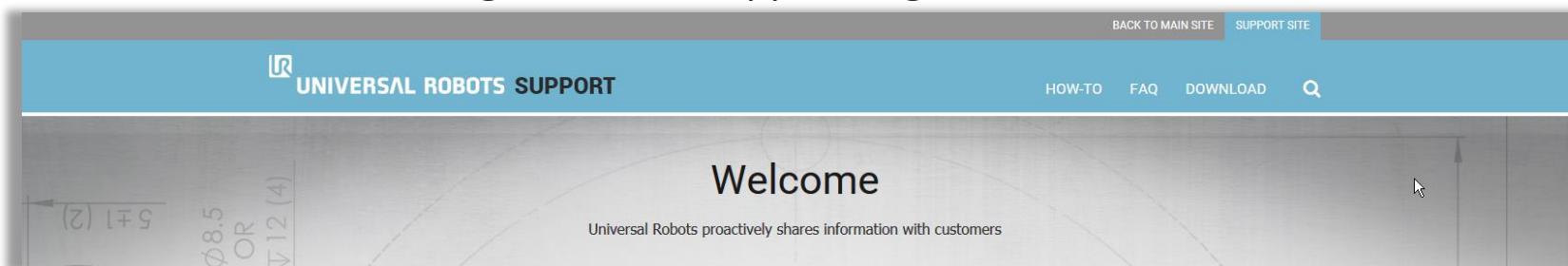
- Información requerida por UR
 - Número de serie del robot s/n
 - Versión software
 - Descripción detallada del error
 - Adjuntar archivo de registro *log_history*
 - Tras recibir esta información, desde la oficina de Soporte Regional, se abrirá un RMA y se enviará el recambio necesario

Soporte técnico regional



Ejercicio práctico 1

- Acceder a la web de soporte
- Descargar la *magic file* para obtener el registro histórico y el lector para estos archivos (*Support Log Reader*)
- Extraer el fichero log del robot usando la magic file
- Visualizar el fichero log usando el Support Log Reader



1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes
3	Comandos básicos 1	9	Modbus TCP/Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

... y ahora



Cumplimiento de estándares internacionales

- Un sistema robótico colaborativo debe cumplir con los requerimientos descritos en los siguientes estándares internacionales

Estándar	Describe	Responsable
ISO 13849-1	Principios generales para el diseño de las partes de los sistemas de mando relativas a la seguridad	Fabricante
ISO 10218-1	Requisitos de seguridad para robots industriales	
ISO 10218-2	Seguridad en la integración de sistemas robóticos	Integrador
ISO TS 15066	Especificación técnica para robots colaborativos	
ISO 12100	Principios generales para el diseño. Evaluación del riesgo y reducción del riesgo.	

ISO 13849-1: 2008

- ¿Qué describe el estándar?
 - Principios generales para el diseño de las partes de los sistemas de mando relativos a la seguridad
- ¿Cuál es su propósito?
 - Es una guía de principios a tener en cuenta por el fabricante de maquinaria industrial en cuanto al diseño de las partes relativas a la seguridad
- ¿Qué contiene?
 - Define los conceptos de Categoría de Seguridad y Nivel de Rendimiento *Performance Level (PL)*



Los robots UR obtienen un Nivel de Rendimiento “D” (PLd)

- PLd es el segundo nivel más alto de fiabilidad en esta clasificación, significa que las funciones de seguridad con este nivel son extremadamente confiables

ISO 10218-1: 2011

- ¿Qué describe el estándar?
 - Requisitos de seguridad para los robots industriales
- ¿Cuál es su propósito?
 - Es una guía de principios a tener en cuenta por el fabricante del robot industrial en el diseño en cuanto a las partes relativas a la seguridad
- 10218-1 fue creado para los robots industriales tradicionales
 - ISO 10218-1 Sección 5.10: Los robots diseñados para una operación colaborativa deben mostrar una indicación visual cuando el robot se encuentre en modo colaborativo y deben cumplir con uno o más de los requerimientos descritos en el apartado 5.10.2 a 5.10.5
 - 5.10.2 Paro monitorizado clasificado de seguridad
 - 5.10.3 Guiado manual
 - 5.10.4 Modo velocidad y separación
 - 5.10.5 Limitación de potencia y fuerza inherente por diseño y control



Los robots UR cumplen con el apartado 5.10.5, ya que la función limitadora de potencia y fuerza está siempre activa

Sistema de Seguridad certificado por TÜV NORD

- ISO 13849-1: 2008 PL d



Su responsabilidad como integrador

- La evaluación de riesgos es siempre obligatoria
 - Se recomienda además cumplir con los siguientes estándares

Estándar	Describe	Responsable
ISO 13849-1	Principios generales para el diseño de las partes de los sistemas de mando relativas a la seguridad	Fabricante
ISO 10218-1	Requisitos de seguridad para robots industriales	
ISO 10218-2	Seguridad en la integración de sistemas robóticos	Integrador
ISO TS 15066	Especificación técnica para robots colaborativos	
ISO 12100	Principios generales para el diseño. Evaluación del riesgo y reducción del riesgo.	



Identificar los riesgos y reducirlos a un nivel apropiado

ISO 10218-2: 2011

- 
- ¿Qué describe el estándar?
 - Requerimientos de seguridad en la integración de sistemas robóticos
 - ¿Cuál es su propósito?
 - Constituye una guía para integradores de robots industriales
 - **Considerar el diseño de la instalación dónde el robot va a ser usado**
 - Consideraciones
 - Definiciones del espacio de trabajo, espacios restringidos, espacios colaborativos
 - Ubicación de elementos de mando y paros de emergencia
 - Diseño de la herramienta del robot
 - Movimiento y velocidades del robot
 - Posición del operador

ISO/TS 15066

- ¿Qué describe el estándar?
 - Requerimientos de seguridad en espacio de trabajo industrial cooperativo. Especificación técnica para el uso de robots colaborativos.
- ¿Cuál es su propósito?
 - Ayuda integradores a completar las evaluaciones de riesgo.
 - Información en contacto quasi-estático y transitorio.
- Contiene
 - Guía detallada para integradores sobre la instalación de robots colaborativos
 - **Límites de fuerza relacionados con robots colaborativos**



ISO 12100: Evaluación y reducción del riesgo

- OBLIGATORIO



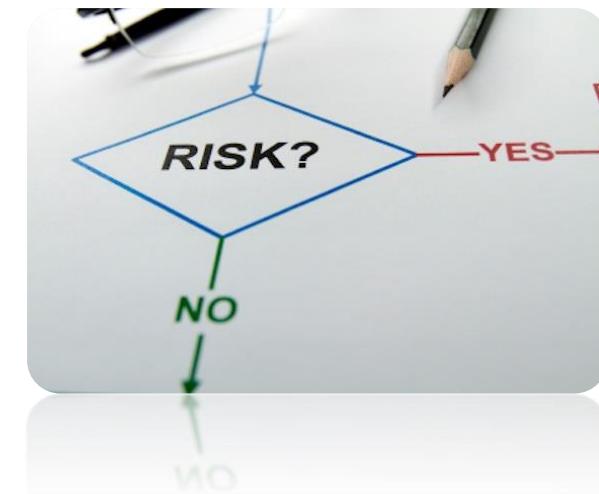
El integrador **DEBE** realizar una evaluación de riesgos

- NO OBLIGATORIO

- El cumplimiento con los estándares
- *Se recomienda cumplir con los estándares!*

- En caso de accidente:

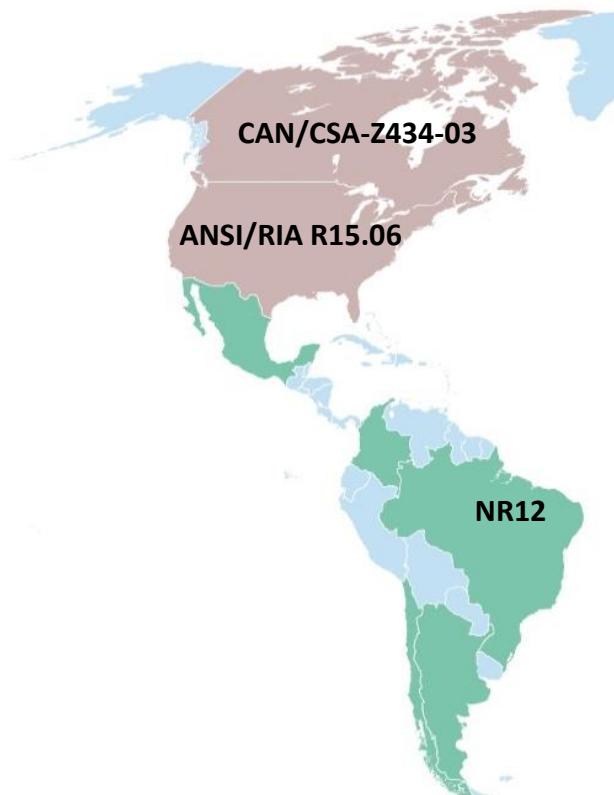
- Si el sistema cumple con los estándares
- Si el sistema **no** cumple con los estándares



- » El fiscal debe probar la culpabilidad
- » El integrador debe probar su inocencia

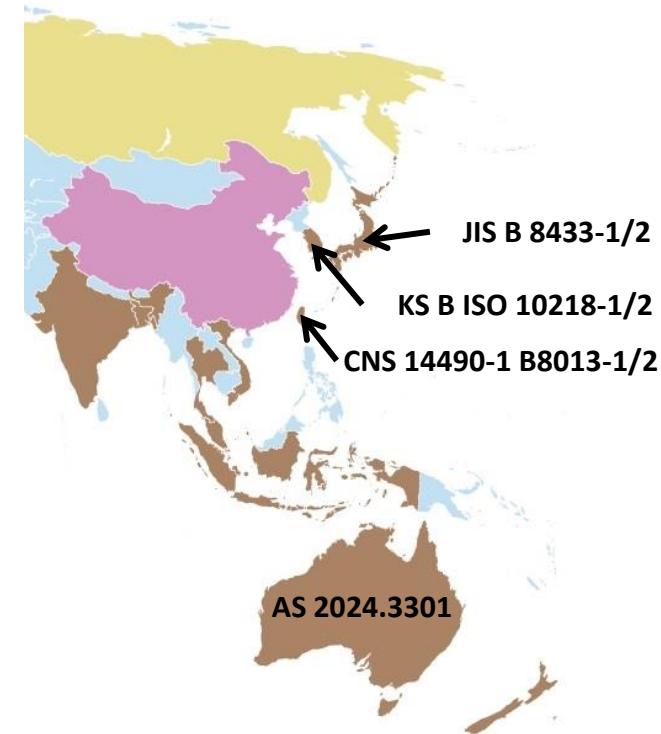
Diferencias regionales

- Canadá
 - CAN/CSA-Z434-03: 2013
 - Armonizado con estándares internacionales ISO
 - Consiste en la ISO 10218-1 e ISO 10218-2 con diferencias regionales
- USA
 - ANSI/RIA R15.06: 2012
 - Armonizado con estándares internacionales ISO
 - ISO 10218-1 e ISO 10218-2 están unidos en un único documento
- Brasil
 - NR 12
 - Estándar no armonizado con estándares internacionales ISO



Diferencias regionales

- Japón
 - JIS B 8433-1:2015 and JIS B 8433-2:2015
 - Traducción directa de las ISO 10218-1 y ISO 10218-2
- Corea del Sur
 - KS B ISO 10218-1 and KS B ISO 10218-2
 - Traducción directa de las ISO 10218-1 y ISO 10218-2
- Taiwán
 - CNS 14490-1 B8013-1 and CNS 14490-1 B8013-2
 - Traducción directa de las ISO 10218-1 y ISO 10218-2
- Australia
 - AS 2024.3301-2009
 - 2024: Seguridad de maquinaria
 - Parte 3301: Requerimientos de seguridad para robots industriales
- Listado completo: [Support site - Safety standards](#)



1	Hardware	7	Comandos avanzados 3
2	Primeros pasos	8	Asistentes
3	Comandos básicos 1	9	Modbus TCP/Ethernet IP
4	Comandos básicos 2	10	Mantenimiento
5	Comandos avanzados 1	11	Estándares seguridad
6	Comandos avanzados 2	12	Seguridad configurable

Características de seguridad

- Ajustes de Configuración de seguridad
 - Sistema de seguridad avanzado y patentado
 - Seguridad redundante
 - Protegida mediante contraseña
- Propósito
 - Adaptar la seguridad a cada aplicación
 - Evitar daños a personal y equipos periféricos
- Evaluación de riesgos
 - Siempre debe realizarse una evaluación de riesgos al instalar una aplicación robotizada
 - Los ajustes de la configuración de seguridad facilitan la evaluación de riesgos



Contraseña (password) de configuración

La configuración de seguridad está protegida por contraseña

- Bloquear (*Lock*)
 - Protege la configuración de seguridad
- Desbloquear (*Unlock*)
 - Habilita la modificación de la configuración de seguridad

Cambiar contraseña

- Ir a Config. Robot (*Setup*)
- Ir a Fijar contraseña
- Cambiar contraseña de seguridad
- Introducir contraseña
- Aplicar cambios

Universal Robots Graphical Programming Environment

PROGRAM <unnamed>
INSTALLATION default
New... Open... Save...

Run Program Installation
Move I/O Log

DANGER

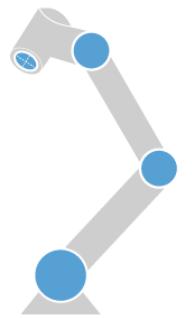
Use of Safety Configuration parameters different from those defined by the risk assessment can result in hazards that are not reasonably eliminated or risks that are not sufficiently reduced.

Factory Presets Custom

Most Restricted Least Restricted

Limit	Normal	Reduced
Power	300 W	200 W
Momentum	25.0 kg m/s	10.0 kg m/s
Stopping Time	400 ms	300 ms
Stopping Distance	500 mm	300 mm
Tool Speed	1500 mm/s	750 mm/s
Tool Force	150.0 N	120.0 N
Elbow Speed	1500 mm/s	750 mm/s
Elbow Force	150.0 N	120.0 N

Safety password **Unlock** **Lock** **Apply**



Configuración básica

Niveles de seguridad

- Muy restringido
- Restringido
- Predeterminado
- Menos restringido

Estado de configuración

- Sincronizada
- Alterada
- Configuración inválida

Guardar configuración

- Presionar "Apply" (Aplicar)
- Confirmar ajustes

Universal Robots Graphical Programming Environment

PROGRAM <unnamed>
INSTALLATION default
New... Open... Save...

C C
C C

	Normal	Reduced
Power	300 W	200 W
Momentum	25.0 kg m/s	10.0 kg m/s
Stopping Time	400 ms	300 ms
Stopping Distance	500 mm	300 mm
Tool Speed	1500 mm/s	750 mm/s
Tool Force	150.0 N	120.0 N
Elbow Speed	1500 mm/s	750 mm/s
Elbow Force	150.0 N	120.0 N

Safety password

Configuración básica

- Valores por defecto

Modo configurado	Muy restringido	Restringido	Predeterminado	Menos restringido
Fuerza (N)	100*	120	150	250
Potencia (W)	80	200	300	1000
Velocidad (mm/s)	250	750	1500	5000
Momento (kg·m/s)	5	10	25	100

* En el UR3 el ajuste de Fuerza (N) por defecto en el modo Muy restringido es 50

- Los límites son valores máximos teóricos, si se superan el robot se detendrá por seguridad

Configuración avanzada

Parámetros adaptables

- Fuerza
- Potencia
- Velocidad
- Momento

Custom

Limit

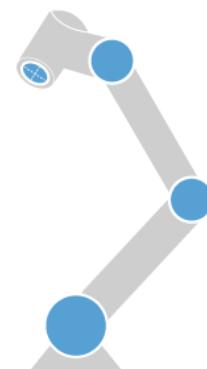
- Power
- Momentum
- Stopping Time
- Stopping Distance
- Tool Speed
- Tool Force
- Elbow Speed
- Elbow Force

Normal

300	W
25.0	kg m/s
400	ms
500	mm
1500	mm/s
150.0	N
1500	mm/s
150.0	N

Reduced

200	W
10.0	kg m/s
300	ms
300	mm
750	mm/s
120.0	N
750	mm/s
120.0	N



Modos

- Modo normal
- Modo reducido

- Requiere uso de entradas o límites de seguridad

Safety password

Unlock

Lock

Apply

Modos de seguridad

- Modo normal
 - Modo de seguridad activo de forma predeterminada
- Modo reducido
 - Activo cuando el TCP del robot se posiciona más allá de un límite seguridad definido como *Modo Reducido con activador (Trigger Reduced Mode Plane)*
 - Activo cuando se usan las entradas configurables definidas como *Modo reducido (Reduced Mode)*
- *En caso de violación de la seguridad:*
 - Modo recuperación (*recovery*)
 - Activo cuando se produce una violación de alguno de los otros modos
 - Este modo permite el movimiento manual del robot fuera del área violada
 - No es posible la ejecución de ningún programa en este modo
 - Las limitaciones de posicionamiento de articulación y TCP están deshabilitadas en este modo

Suma de comprobación de seguridad

Comprobación (Checksum)

- Indicación visual de la configuración de seguridad
- Indicada con colores y números
- La suma de comprobación cambia si se modifica la configuración
- Al pulsar sobre el código aparece la configuración de seguridad activa

Limit	Normal
Power	300 W
Momentum	25.0 kg m/s
Stopping Time	1000 ms
Stopping Distance	2000 mm
Tool Speed	1500 mm/s
Tool Force	150.0 N
Elbow Speed	5000 mm/s
Elbow Force	250.0 N

Close

Límites de junta

Velocidad máxima

- Establece velocidad máxima para cada articulación

Intervalo de posiciones

- Establece rango mínimo máximo para cada articulación

Position range

Joints	Range	Normal Mode		Reduced Mode		
		Minimum	Maximum	Minimum	Maximum	
Base	-363 — 363 °	-363	363	-363	363	+2 ° / -2 °
Shoulder	-363 — 363 °	-363	-50	-363	363	+2 ° / -2 °
Elbow	-363 — 363 °	-363	363	-363	363	+2 ° / -2 °
Wrist 1	-363 — 363 °	-363	363	-363	363	+2 ° / -2 °
Wrist 2	-363 — 363 °	-363	363	-363	363	+2 ° / -2 °
Wrist 3	-363 — 363 °	-363	363	-363	363	+2 ° / -2 °

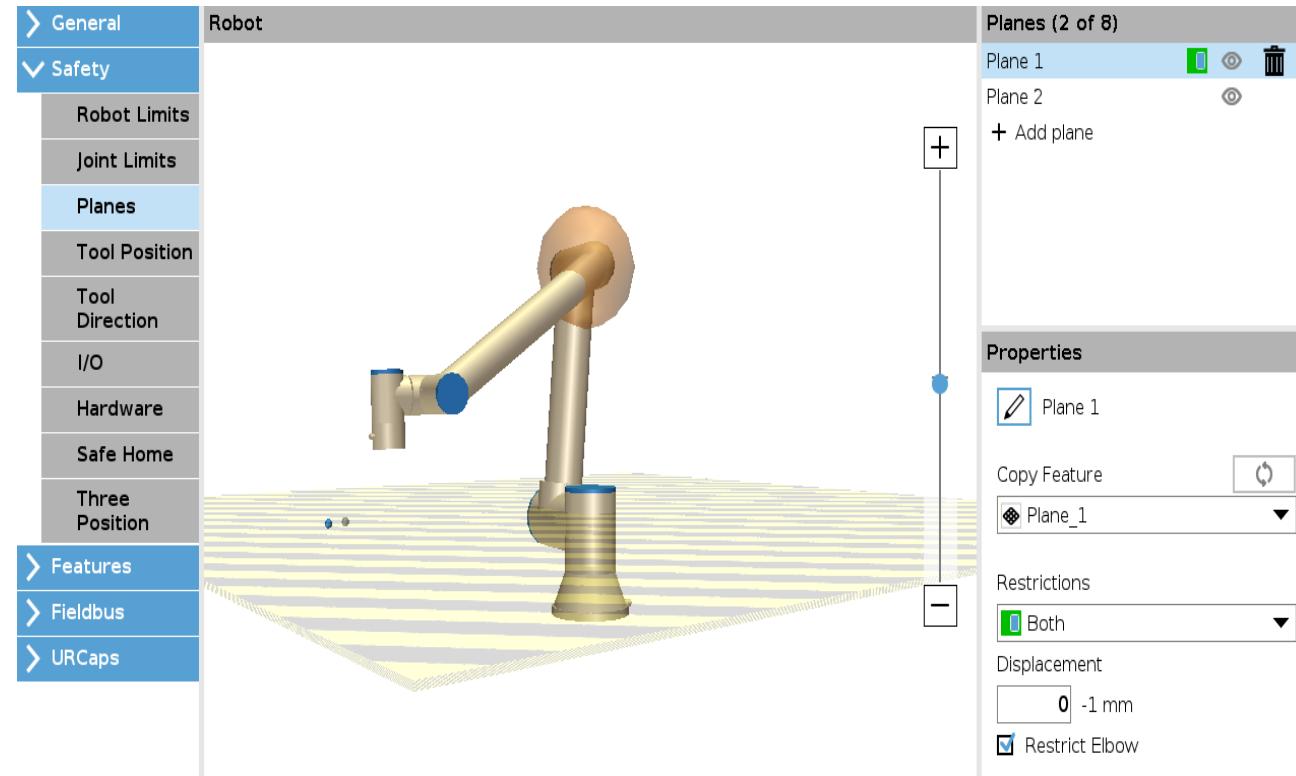
Maximum speed

Joints	Maximum	Normal Mode		Reduced Mode	
		131	131	-11 °/s	-11 °/s
Base	max: 131 °/s	131	131	-11 °/s	-11 °/s
Shoulder	max: 131 °/s	131	131	-11 °/s	-11 °/s
Elbow	max: 191 °/s	191	191	-11 °/s	-11 °/s
Wrist 1	max: 191 °/s	191	191	-11 °/s	-11 °/s
Wrist 2	max: 191 °/s	191	191	-11 °/s	-11 °/s
Wrist 3	max: 191 °/s	191	191	-11 °/s	-11 °/s

Límites (Boundaries)

Plano de seguridad

- Área de trabajo restringida
- Hasta 8 planos distintos
- Activos en movimiento manual y en ejecución de programas
- Los planos pueden activar *Modo Reducido* cuando el TCP los atraviesa

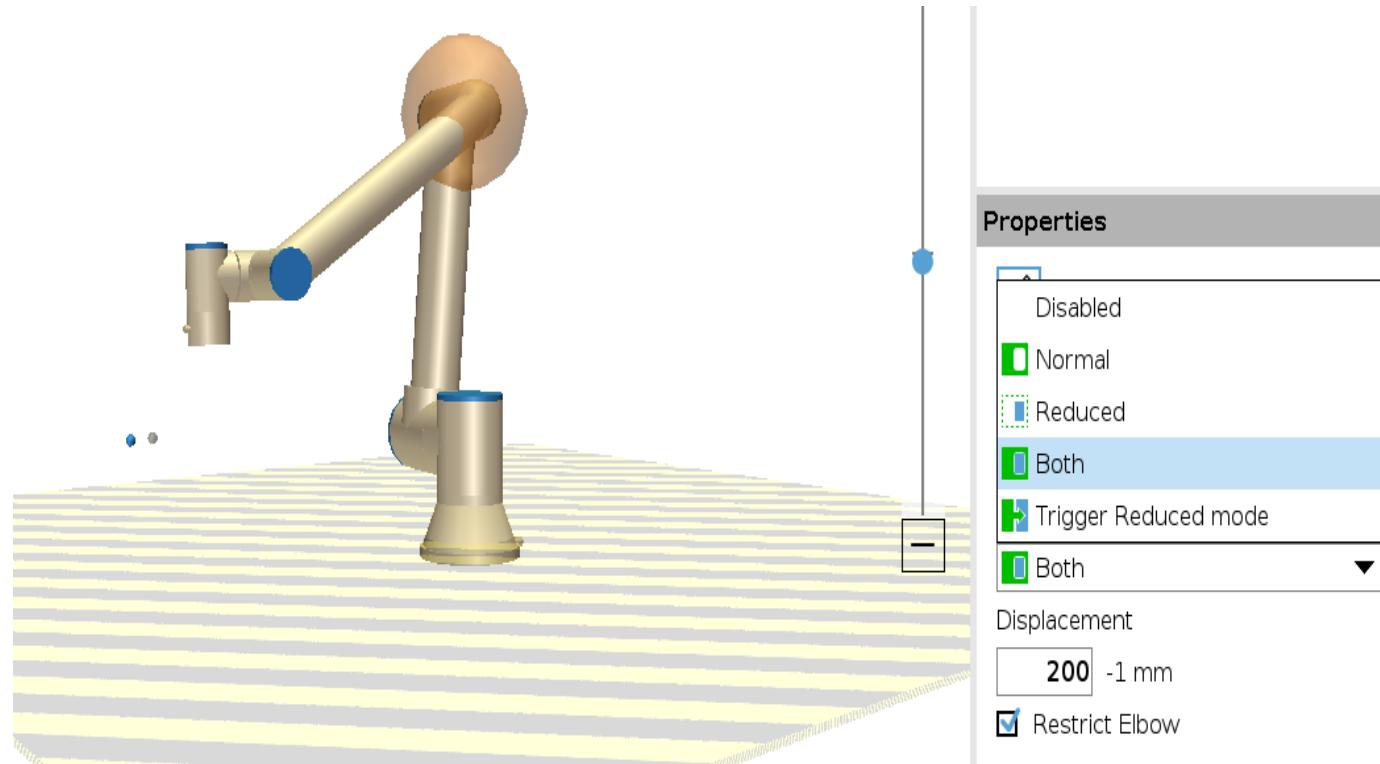


- IMPORTANTE: Los límites de seguridad limitan sólo al TCP, y al CODO , NO al brazo completo del robot

Límites (*Boundaries*)

Establecer plano

- Seleccionar Copy Feature
(*Función de copia*)
 - Define qué plano usar
- Seleccionar modo de seguridad
 - Define en qué modo se activa el límite, o si es el límite quien activa el modo reducido
- Desplazamiento
 - Desplaza el plano la distancia indicada
- Aplicar (*Apply*)
 - Activa los cambios en la configuración



Modos de seguridad

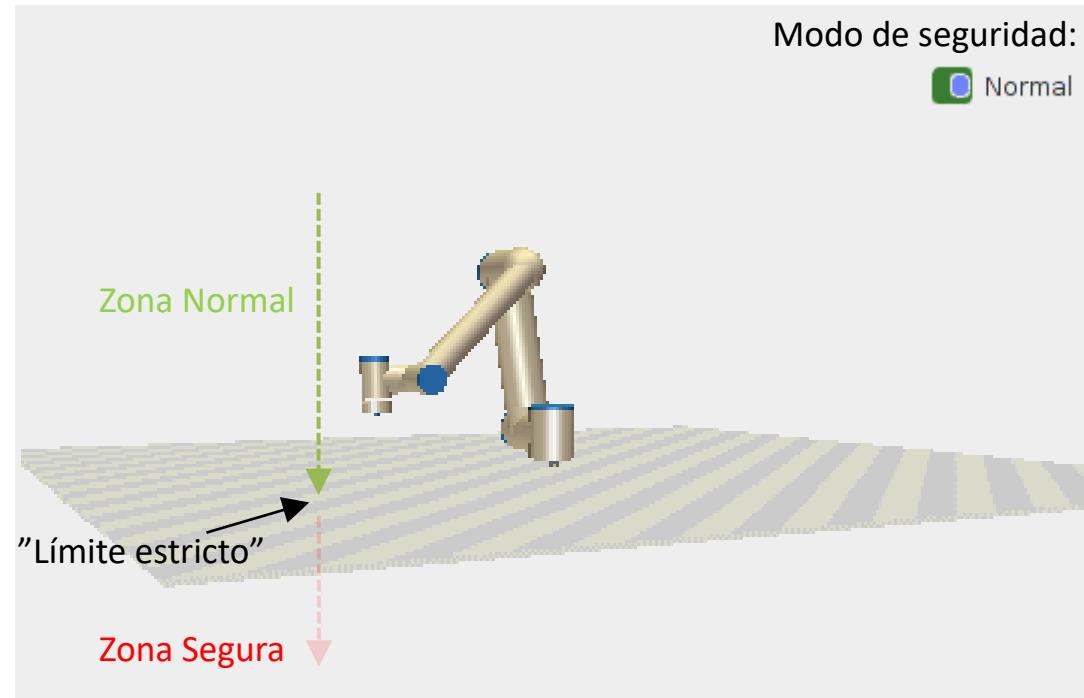
- Comportamiento de los límites según el modo de seguridad

Modo de seguridad	Comportamiento
Desactivad (<i>Disabled</i>)	inactivo
 Normal	actúa como límite estricto con el robot en modo normal
 Reducido	actúa como límite estricto sólo si el robot está en modo reducido
 Ambos (<i>Both</i>)	actúa como límite estricto en ambos modos
 Modo Reducido con activador (<i>Trigger Reduced Mode</i>)	pasa el robot a modo reducido cuando el TCP atraviesa el plano

Comportamiento de los límites (*boundary*)

Prueba

- Set Feature plane= *Base*
- Modo de seguridad = *Normal*
- Aplicar configuración
- Prueba en “freedrive”
 - Mover el robot desde la zona Normal hacia la zona Segura



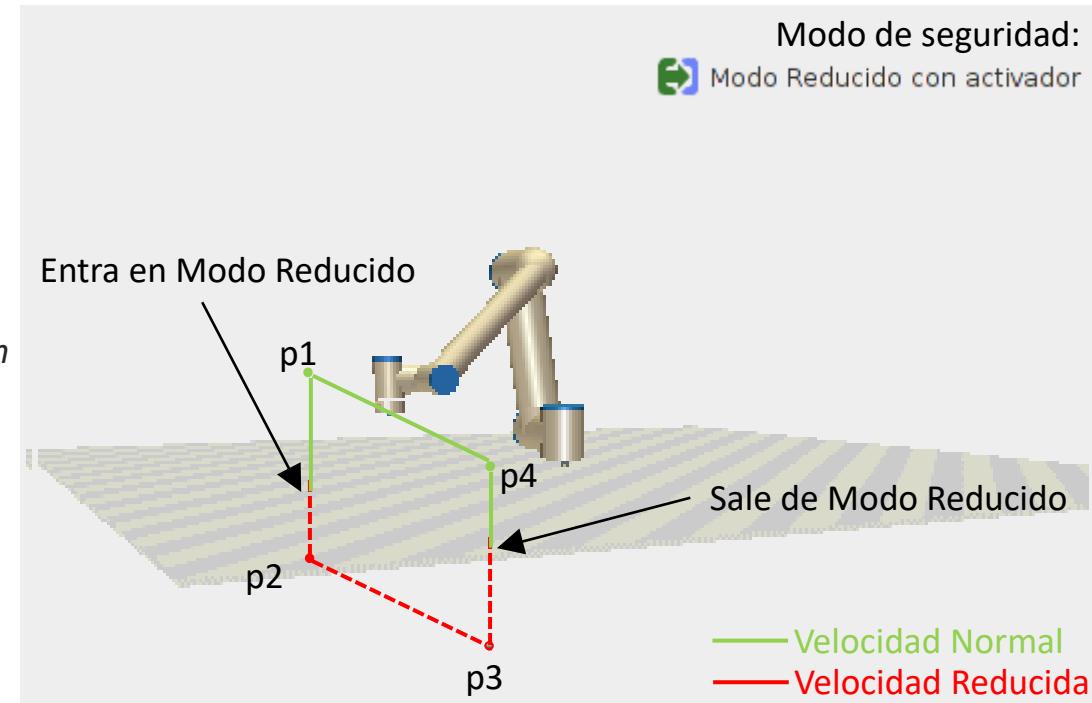
Comportamiento en ejecución de programa

- Se aborta la ejecución del programa, indicando la violación de seguridad mediante un mensaje emergente

Activar modo reducido (*Trigger reduced mode*)

- Prueba
 - Cambiar modo de seguridad a: “Trigger reduced mode”
 - Seleccionar velocidad máxima en Modo Reducido a: 350 mm/s
 - Aplicar configuración
 - Guardar como *safety.installation*

```
Robot Program
MoveL
  Waypoint_1
  Waypoint_2
  Waypoint_3
  Waypoint_4
```

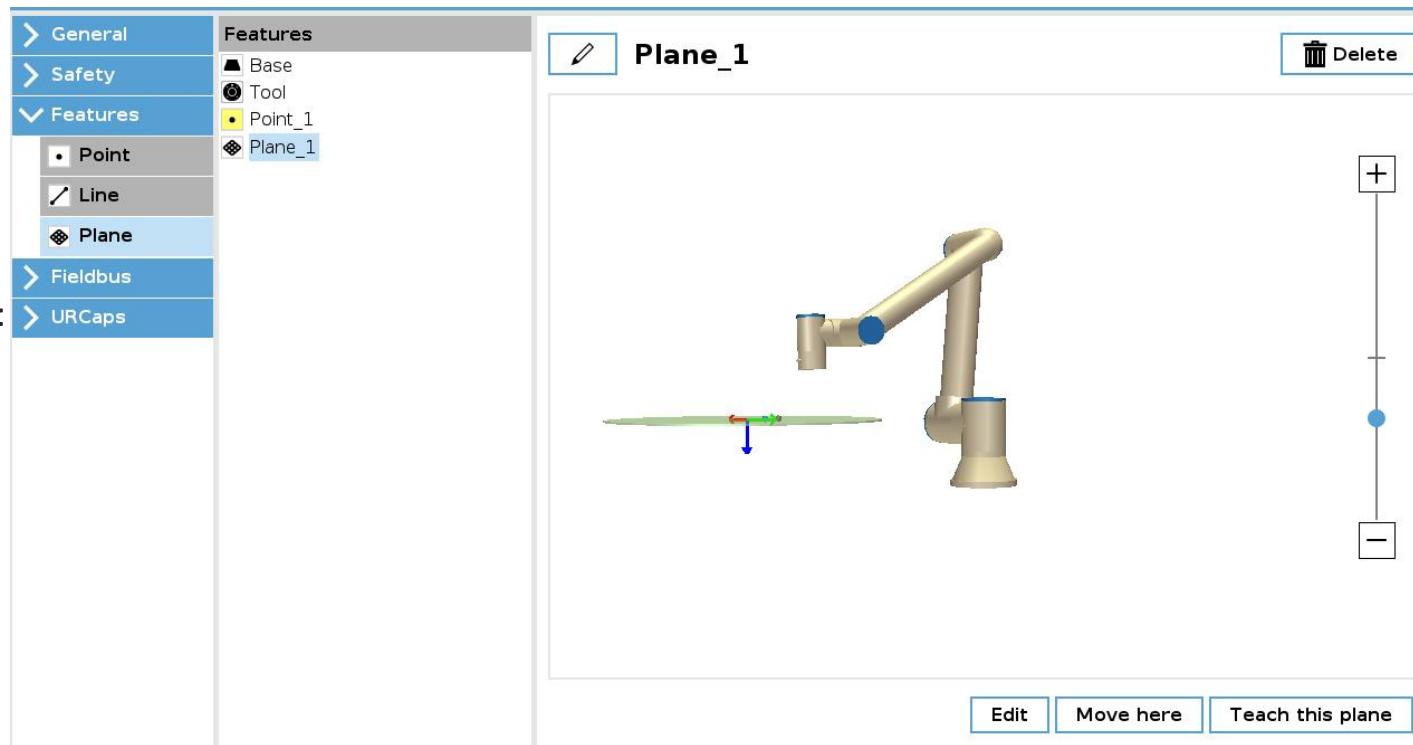


- Guardar programa de ejemplo como *trigger_reduced_mode.urp*

Plano de seguridad definido por el usuario

Features (Funciones)

- En PolyScope un Plano se define como una feature
- Se pueden establecer múltiples features
- Establecer Features como:
 - Punto
 - Línea
 - Plano



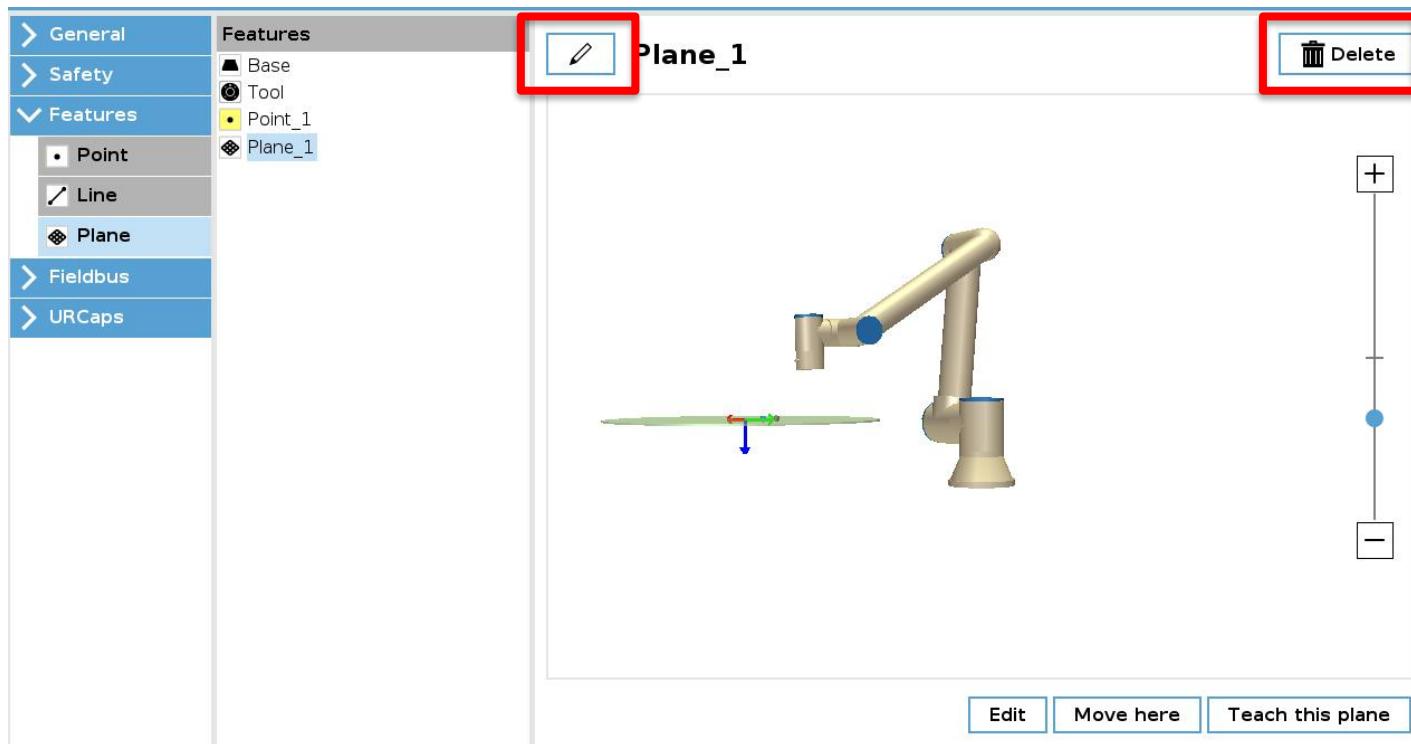
Añadir Feature

- Seleccionar Plano

Plano de seguridad definido por el usuario

Opciones

- Renombrar Feature
- Eliminar Feature
- Editar
- Enseñar



Plano de seguridad definido por el usuario

Establecer Plano

- El plano se define mediante tres Waypoints

Enseñar plano vertical

- Punto_1 = Origen
- Punto_2 = Dirección-Y
- Punto_3 = Dirección-X

Plane Teach-in

Plane

Origin

Positive X-axis

Direction of Positive Y-axis

Plane preview

Previous

Next

Finish

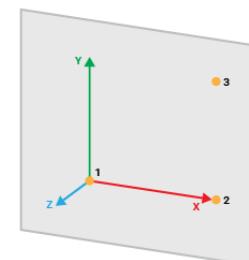
Cancel

Steps (1/5) - Plane

This wizard will assist you in creating a new plane using 3 positions

1 - Origin
2 - Positive X-axis
3 - Direction of Positive Y-axis

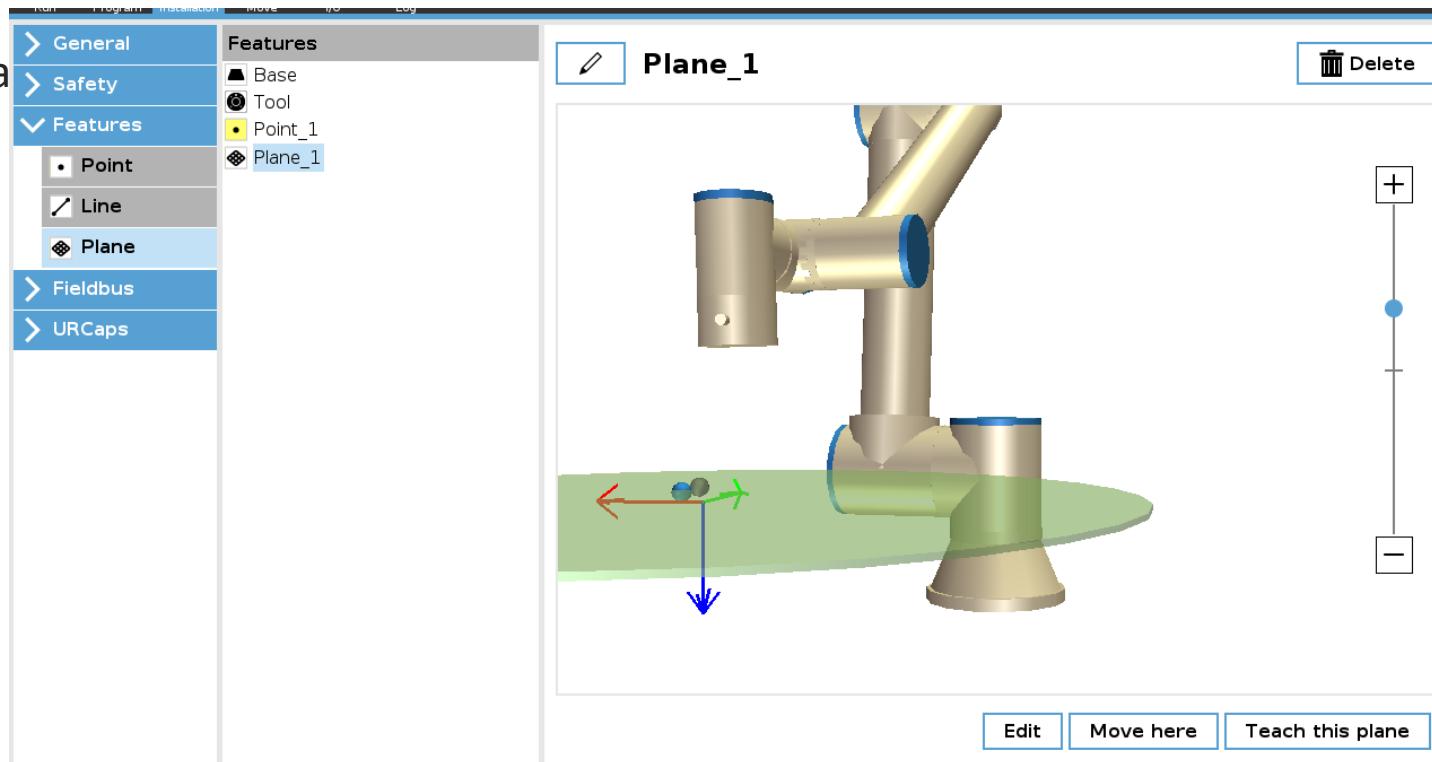
Example:



Plano de seguridad definido por el usuario

Representación gráfica del Plano aprendido

- Guardar instalación



Límite de la herramienta

- Limita el radio del TCP

The screenshot shows the Universal Robots software interface with the following details:

- Left Sidebar:** A vertical menu with sections: General, Safety (selected), Robot Limits, Joint Limits, Planes, Tool Position (selected), Tool Direction, I/O, Hardware, Safe Home, Three Position, Features, Fieldbus, URCaps.
- Central Area:** A 3D model of a robot arm with a blue sphere representing the tool center point (TCP) at its end effector.
- Right Panel:**
 - Tool (2 of 3):** A list of tools. "Tool_1" is selected, indicated by a blue background. A trash icon is next to it.
 - Add Tool:** A button labeled "+ Add Tool".
 - Tool_1 Configuration:**
 - Radius (max: 300mm):** Set to **250.0 mm**.
 - Copy TCP:** A dropdown set to **Custom**, with a refresh icon.
 - Edit Position:**
 - X: **200.0 mm**
 - Y: **-250.0 mm**
 - Z: **200.0 mm**
- Bottom Bar:** Safety password input field, **Unlock** button, **Lock** button (highlighted in blue), and **Apply** button.

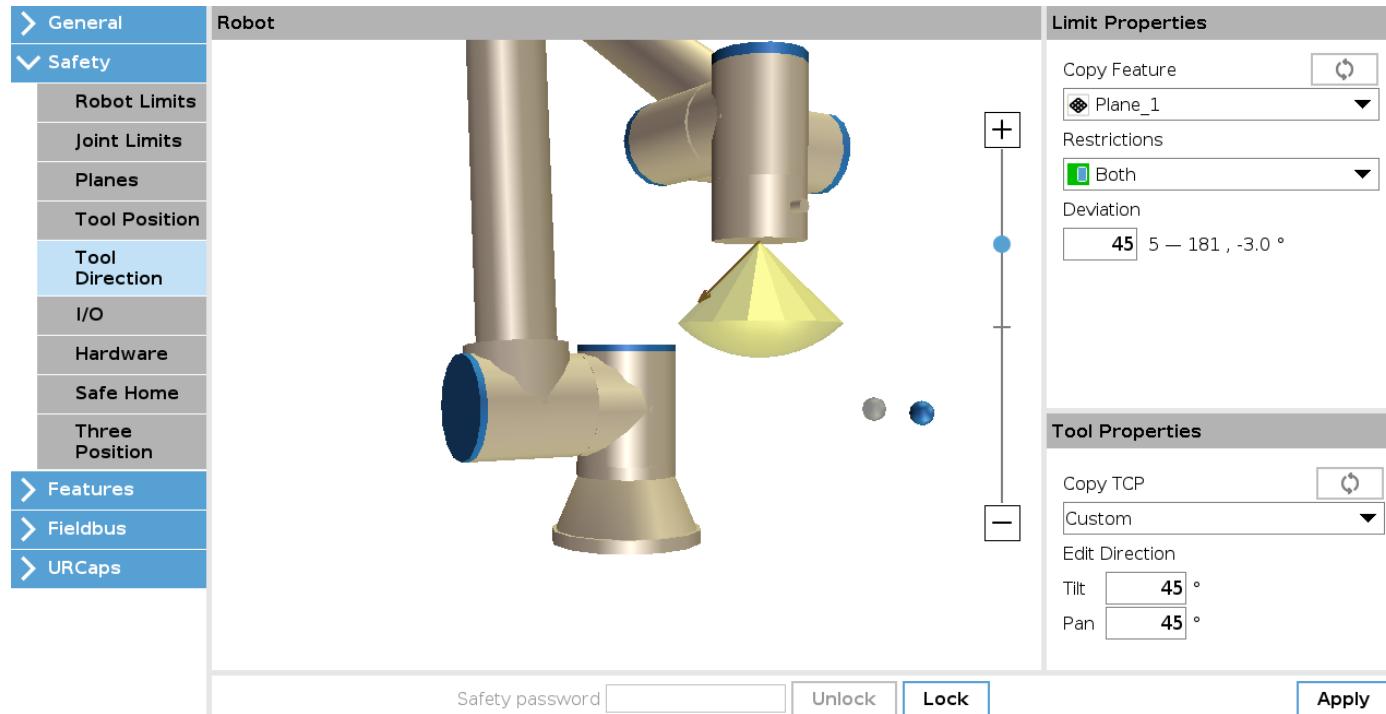
Límite de la herramienta

Limita la variación angular del TCP

- Establece la variación angular máxima del TCP respecto de la Feature elegida

Prueba

- Elegir “Copy feature”: *Base*
- Establecer Desviación: 20°
- Seleccionar modo de seguridad a: *Both (Ambos)*
- Aplicar
- Probar en movimiento manual



I/O de seguridad

Funciones de seguridad

- Se pueden asignar Funciones de Seguridad a las I/O Configurables
- Todas las funciones son redundantes
 - Dos señales para cada función (doble canal)
 - Categoría 3, PLD
- I/O Configurables
 - Entradas digitales
 - Salidas digitales

The screenshot shows the 'Safety' configuration page in the Universal Robots software. On the left is a sidebar with navigation links: General, Safety (selected), Robot Limits, Joint Limits, Planes, Tool Position, Tool Direction, I/O (selected), Hardware, Safe Home, Three Position, Features, Fieldbus, and URCaps. The main area has two tables: 'Input Signal' and 'Output Signal'. The 'Input Signal' table lists four signals: config_in[0], config_in[1] (Emergency Stop), config_in[2], config_in[3] (Reduced Mode), config_in[4], config_in[5] (Safeguard Reset), and config_in[6], config_in[7] (Operational Mode). The 'Output Signal' table lists four signals: config_out[0], config_out[1] (Unassigned), config_out[2], config_out[3] (System Emergency Stopped), config_out[4], config_out[5] (Unassigned), and config_out[6], config_out[7] (Unassigned). Below the tables are buttons for 'Safety password' (input field), 'Unlock' (button), 'Lock' (button), and 'Apply' (button).

Input Signal	Function Assignment	OSSD
config_in[0], config_in[1]	Emergency Stop	<input type="checkbox"/>
config_in[2], config_in[3]	Reduced Mode	<input type="checkbox"/>
config_in[4], config_in[5]	Safeguard Reset	<input type="checkbox"/>
config_in[6], config_in[7]	Operational Mode	<input type="checkbox"/>

Output Signal	Function Assignment	OSSD
config_out[0], config_out[1]	Unassigned	<input type="checkbox"/>
config_out[2], config_out[3]	System Emergency Stopped	<input type="checkbox"/>
config_out[4], config_out[5]	Unassigned	<input type="checkbox"/>
config_out[6], config_out[7]	Unassigned	<input type="checkbox"/>

Safety password

Entradas de seguridad

Entradas de seguridad

- Para de emergencia
 - Para la conexión desde un pulsador de emergencia o PLC de Seguridad
- Modo reducido
 - Señal baja: operación en modo reducido
 - Señal alta: operación en modo normal
- Restablecimiento de protección
 - Si la *Parada de protección* está conectada, se puede restablecer mediante esta señal
- Interruptor de 3 posiciones
 - Ver siguiente diapositiva.

Input Signal	Function Assignment	OSSD
config_in[0], config_in[1]	Emergency Stop	<input type="checkbox"/>
config_in[2], config_in[3]	Reduced Mode	<input type="checkbox"/>
config_in[4], config_in[5]	Safeguard Reset	<input type="checkbox"/>
config_in[6], config_in[7]	Operational Mode	<input type="checkbox"/>

Output Signal	Function Assignment	OSSD
config_out[0], config_out[1]	Unassigned	<input type="checkbox"/>
config_out[2], config_out[3]	System Emergency Stopped	<input type="checkbox"/>
config_out[4], config_out[5]	Unassigned	<input type="checkbox"/>
config_out[6], config_out[7]	Unassigned	<input type="checkbox"/>

Safety password

Salidas de seguridad

- Salidas de seguridad

- Sistema en parada de emergencia
 - HI: modo normal
 - LO: parado por emergencia

- Robot en movimiento

- HI: robot no se mueve
- LO: robot moviéndose

- Robot no detenido

- HI: la señal de petición de paro está activa hasta que el robot se ha detenido
- LO: sin petición de paro

- Modo reducido

- HI: modo normal
- LO: modo reducido

- Sin modo reducido

- Estado inverso al modo reducido

Output Signal	Function Assignment
config_out[0], config_out[1]	Robot Moving
config_out[2], config_out[3]	System Emergency Stopped
config_out[4], config_out[5]	Robot Not Stopping
config_out[6], config_out[7]	Reduced Mode

Examinación

- Tiene 30 minutos para responder las preguntas.
- Las preguntas pueden tener múltiples respuestas, así que revise todas las opciones cuidadosamente antes de responder.
- Indique la respuesta correcta mediante una marca y rellene los espacios en blanco con claridad cuando sea necesario.
- Puede usar cualquier material que tenga disponible. Evite hacer comentarios con los demás examinados.

Buena Suerte!

Nombre: _____ Fecha: _____

Resultado _____



Examinación

1. Que se necesita para prender un robot UR e iniciar un programa sin utilizar el “Teach Pendant”?
 - a. El robot tiene que estar encendido a través de los conectores de encendido / apagado en el controlador.
 - b. Todas las entradas y salidas necesitan ser ajustadas (set) a sus valores iniciales.
 - c. El robot no puede iniciarse por completo sin necesidad de utilizar el TP
 - d. El robot tiene que ser inicializado automática a través de una entrada que se define dentro del “Default program”
 - e. Tiene que haber un programa estándar configurado dentro del “Default program” que tiene que basarse en la “default.installation”
2. En qué casos se usa el comando Ajustar (Set)?
 - a. Activar salidas digitales o analógicas
 - b. Modificar los valores en registros de salida MODBUS
 - c. Cambiar el valor de la carga útil al cargar o descargar piezas
 - d. Ajustar el montaje del robot
 - e. Ninguna de las anteriores
3. ¿De qué opciones disponemos para actuar instantáneamente frente a cambios en el estado de una entrada digital?
 - a. Usar la función "Evento"
 - b. Activar la opción “Comprobar expresión constantemente” en comandos “If...Else”
 - c. Activar la opción “Comprobar expresión constantemente” en comandos “Loop”
 - d. Utilizando una variable de instalación que fue escrita con una entrada digital
 - e. Ninguna de las anteriores

Examinación

4. ¿Cómo llamarías a otro programa desde el programa actual?

- a. Usando el comando “Carpeta” para invocar al programa
- b. Usando el comando “SubProg” para llamar al programa
- c. Usando el comando “Código de Script”
- d. Usando el comando “Thread”
- e. Ninguna de las anteriores

5. ¿Qué es una variable de instalación?

- a. Es global a todos los programas que usen el mismo archivo de instalación
- b. Son variables que mantienen su valor incluso si se desconecta el robot
- c. ¿Variable de instalación? Eso no existe
- d. Una variable, que se guarda en la memoria física y no dentro del sistema RAM.
- e. Ninguna de las anteriores

6. ¿Cuáles de las siguientes son posibles causas para el mensaje "Parada de protección por limitación de fuerza"?

- a. El movimiento del robot se encuentra obstruido
- b. Los valores de TCP, carga y/o montaje no están correctamente definidos
- c. Valores de aceleración muy bajos
- d. Valores de aceleración muy elevados
- e. Ninguna de las anteriores

Examinación

7. ¿Por qué es necesario ajustar y reajustar el valor de la carga útil del TCP de un robot UR?
- a. Los parámetros en los motores se calculan dinámicamente según el dato de carga
 - b. Una definición incorrecta de la carga útil afecta a la estabilidad del robot
 - c. No es necesario, trabajar con la carga máxima funciona bien
 - d. Un ajuste incorrecto de la carga útil puede causar un corto circuito en la articulación
 - e. Ninguna de las anteriores
8. ¿Cómo crearías una variable que acepte y guarde valores introducidos por un operador?
- a. Usando el comando "Asignación" y seleccionando la fuente como "Operador"
 - b. Usando la opción para inicializar variables
 - c. Usando la opción de variables de instalación
 - d. Esto no es posible
 - e. Ninguna de las anteriores
9. Cuales de las siguientes son posibles causas para el mensaje "Infinite loop detected"?
- a. Programa sin un comando físico o tiempo de espera.
 - b. Loop sin un comando físico o tiempo de espera
 - c. La variable del Loop es muy grande
 - d. "Thread" o evento un comando sync() o tiempo de espera
 - e. Ninguna de las anteriores

Examinación

10. ¿Cómo puedo crear una variable de tipo float?
- a. Usar la pestaña de asignación para crear la variable
 - b. Renombrar la variable como “Coma flotante”
 - c. Asignando un valor de coma flotante al inicializar la variable
 - d. Por defecto, todas las variables se crean como float
 - e. Ninguna de las anteriores
11. ¿Qué es una variable de tipo “pose” y cómo se representa?
12. ¿Cuál es la diferencia entre una instrucción de movimiento MoveP y MoveL?
- a. No hay diferencia
 - b. En un MoveP se puede fijar un radio de transición, en un MoveL no es posible
 - c. MoveP mantiene constante la velocidad en los ejes, MoveL no
 - d. MoveP mantiene constante la velocidad en el TCP, MoveL no
 - e. Ninguna de las anteriores

Examinación

13. Qué declaraciones con respecto a un MoveC son correctas?
- a. El comando MoveC crea un medio circulo
 - b. EL comando MoveC se define por un punto de partida, punto de transición y un punto de salida
 - c. MoveC? No existe ese comando
 - d. El comando MoveC tiene que ser programado dentro de un MoveL
 - e. El comando MoveC tiene que ser programado dentro de un MoveP
14. ¿Con qué estándares de seguridad cumplen los robots UR?
- a. ISO 10218 secciones 1 y 2
 - b. ANSI/RIA R15.06-2012
 - c. CAN/CSA Z434-2003(R2013)
 - d. Todos los anteriores
 - e. Ninguna de los anteriores
15. Que declaraciones con respecto a la colaboración (collaboration) de los robots UR son correctas?
- a. Cumplen con EN ISO 10218 capítulo 5.10.5 (Limitados por poder y fuerza)
 - b. Una evaluación de riesgo tiene que hacerse para cada aplicación
 - c. Un UR es colaborativo debido a su diseño
 - d. La velocidad puede ser limitada para un robot colaborativo
 - e. Ninguna de las anteriores

Examinación

16. ¿Cómo se puede monitorizar el valor de una variable en paralelo a la ejecución del programa principal?

- a. Dentro de la estructura de un "Evento"
- b. Dentro de la condición inicial de un evento
- c. Usando un "Thread"
- d. Utilizando una variable de instalación
- e. Ninguna de las anteriores

17. La evaluación de riesgos sólo es necesario en el caso que un humano trabaje en la zona de trabajo del robot

- a. Verdadero
- b. Falso

18. Complete las siguientes especificaciones técnicas de un UR5:

Velocidad máxima de la herramienta:

Velocidad máxima de las articulaciones:

Repetibilidad:

Examination

19. En CB3, los límites de seguridad NO pueden ser configurados...
- a. Usando planos definidos por el usuario
 - b. Usando figuras complejas, tales como elipses o superficies curvas
 - c. Desplazando un plano determinado
 - d. Para activar el modo de seguridad reducido
 - e. Ninguna de las anteriores
20. Se debe realizar una calibración de la posición cero de las articulaciones cuando:
- a. Se substituye una articulación
 - b. Se actualiza el firmware en la articulación
 - c. Se instala el robot después de recibirlo desde fábrica
 - d. Cuando se registran errores “Parada de protección por limitación de fuerza”
 - e. Ninguna de las anteriores