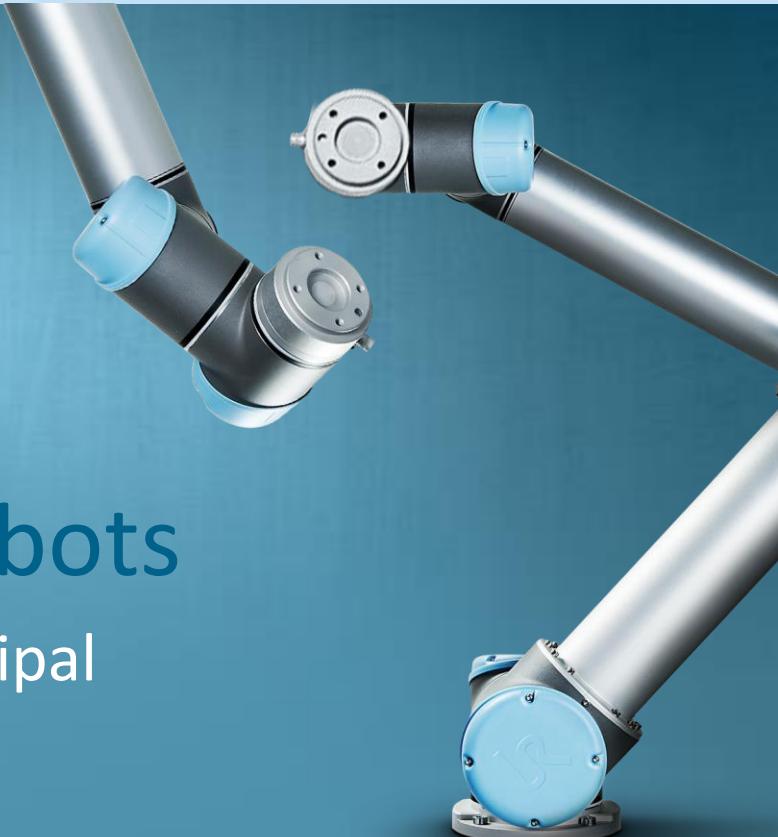




UNIVERSAL ROBOTS

Universal Robots

Formación Principal





Bienvenido a la Formación Principal

- 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
- Examen tipo test de 30 min. al finalizar el curso



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 E/S
- Leer y modificar programas
- Realizar operaciones de diagnóstico y mantenimiento

Presentación del Profesor

Nombre: *ingresa tu nombre aquí*

Rol: *ingresa tu posición de trabajo aquí*

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

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 viene en la caja?

- Caja 1
 - Brazo del Robot
- Caja 2
 - Caja de control
 - Cable de alimentación
 - Soportes de montaje
 - Cable para la herramienta
 - Puntero laser UR
 - Manuales
 - Certificado Test de Producción



- El brazo del Robot y el control están protegidos por bolsas antiestáticas ESD

Brazo del Robot

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



Brazo del Robot

- Denominación y tamaño de las juntas

	UR3	UR5	UR10
Base	Tamaño 2	Tamaño 3	Tamaño 4
Hombro	Tamaño 2	Tamaño 3	Tamaño 4
Codo	Tamaño 1	Tamaño 3	Tamaño 3
Muñeca 1	Tamaño 0	Tamaño 1	Tamaño 2
Muñeca 2	Tamaño 0	Tamaño 1	Tamaño 2
Muñeca 3	Tamaño 0	Tamaño 1	Tamaño 2



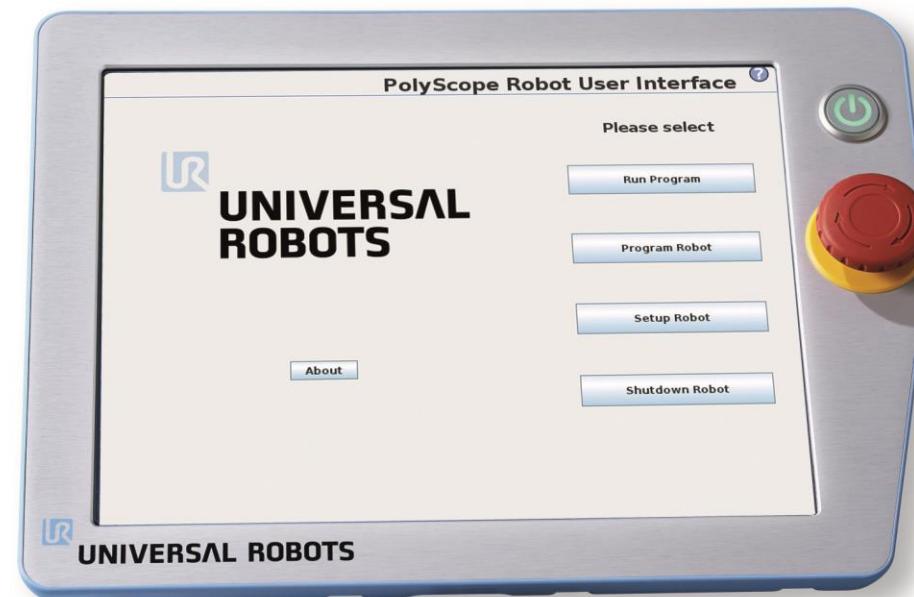
Controlador

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



Consola de programación

- Pantalla táctil
- Incluye
 - Botón encendido
 - Pulsador de emergencia
 - Pulsador movimiento libre
 - Conexión USB



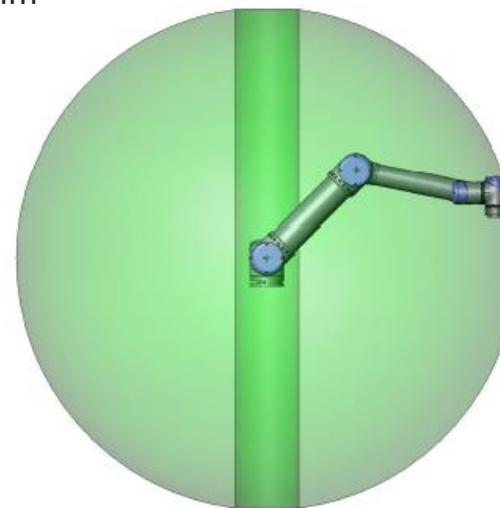
Especificaciones

	UR3	UR5	UR10
<i>Carga útil</i>	3 kg.	5 kg.	10 kg.
<i>Alcance</i>	500 mm	850 mm	1300 mm
<i>Rango de las articulaciones</i>	+/-360° Rotación infinita de la última articulación	+/-360°	+/-360°
<i>Repetitividad</i>	+/-0.1 mm	+/-0.1 mm	+/-0.1 mm
<i>Velocidad máx. Eje</i>	180°/seg y 360°/seg	180°/seg	120°/seg y 180°/seg
<i>Velocidad máx.PCH</i>	1000 mm/seg	1000 mm/seg	1000 mm/seg
<i>Peso</i>	11.0 kg	18.4 kg	28.9 kg
<i>Clasificación IP</i>	IP64	IP54	IP54
<i>Temperatura</i>	0-50°C	0-50°C	0-50°C
<i>Alimentación</i>	100-240V AC, 50-60 Hz	100-240V AC, 50-60 Hz	100-240V AC, 50-60Hz

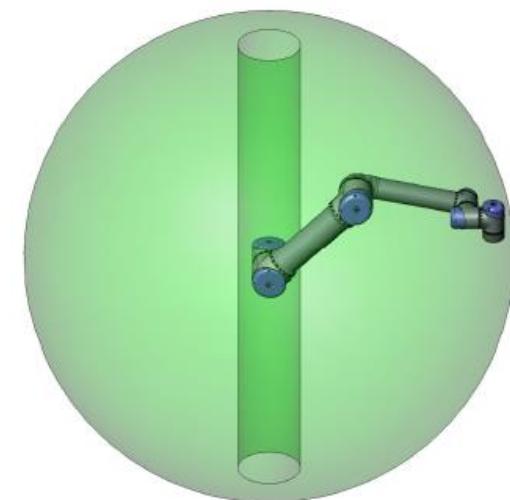
- Especificaciones adicionales en www.universal-robots.com

Área del trabajo del Robot

- Área de trabajo del UR5
 - Esfera aproximada de Ø1700 mm
- Limitación
 - Volumen cilíndrico alrededor del centro de la Base



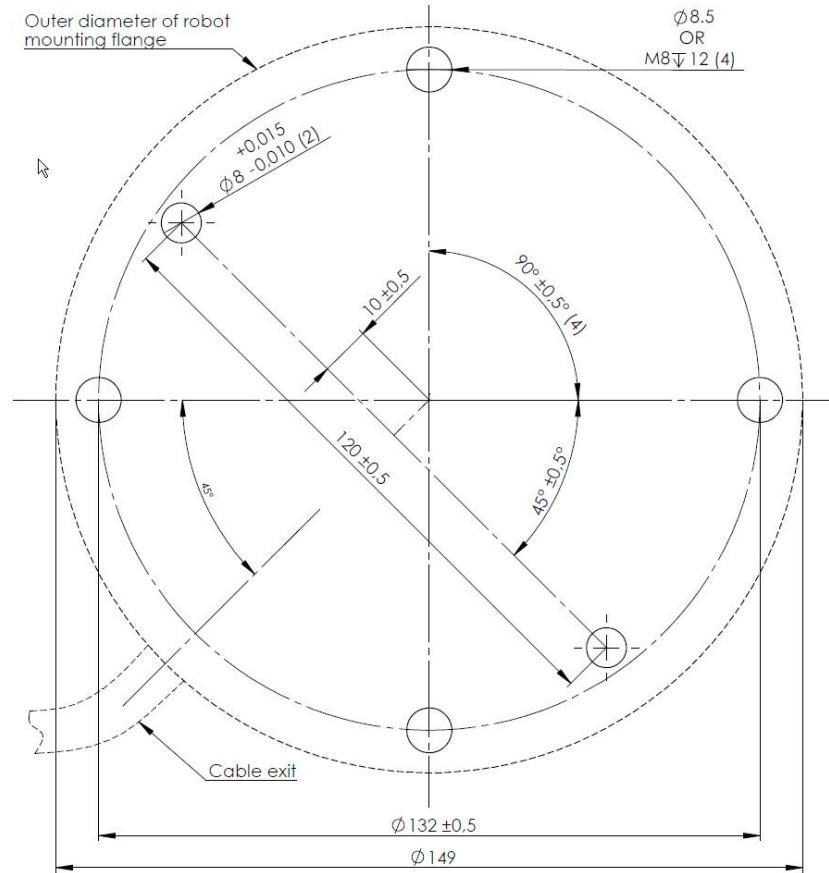
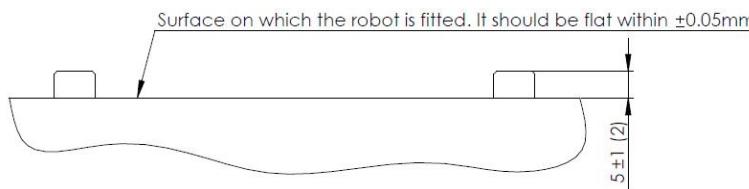
Vista frontal



Vista inclinada

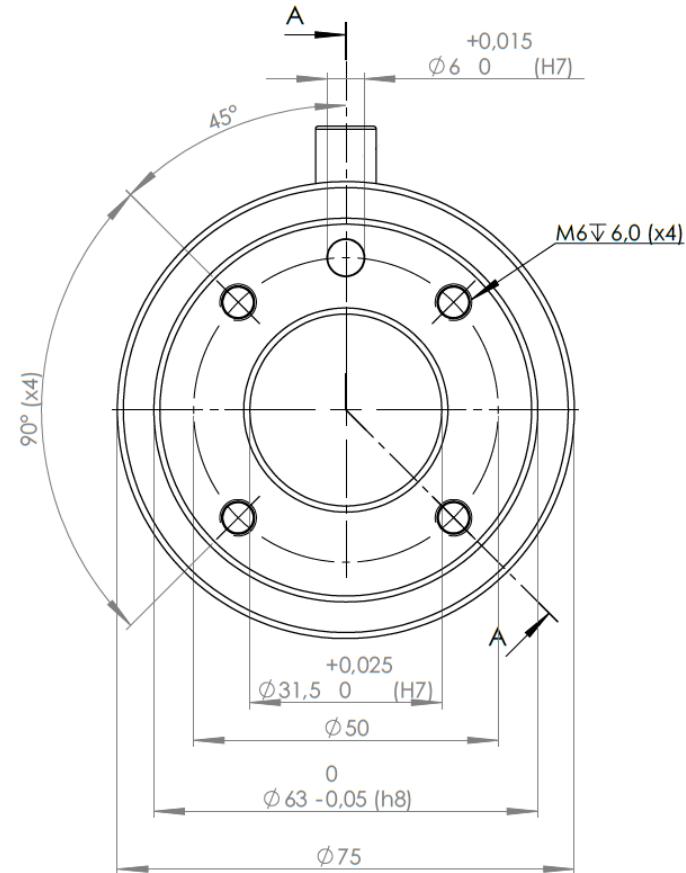
Montaje de la base

- Requisitos
 - Superficie sólida
 - Huella 149 mm
 - 4 tornillos M8



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
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

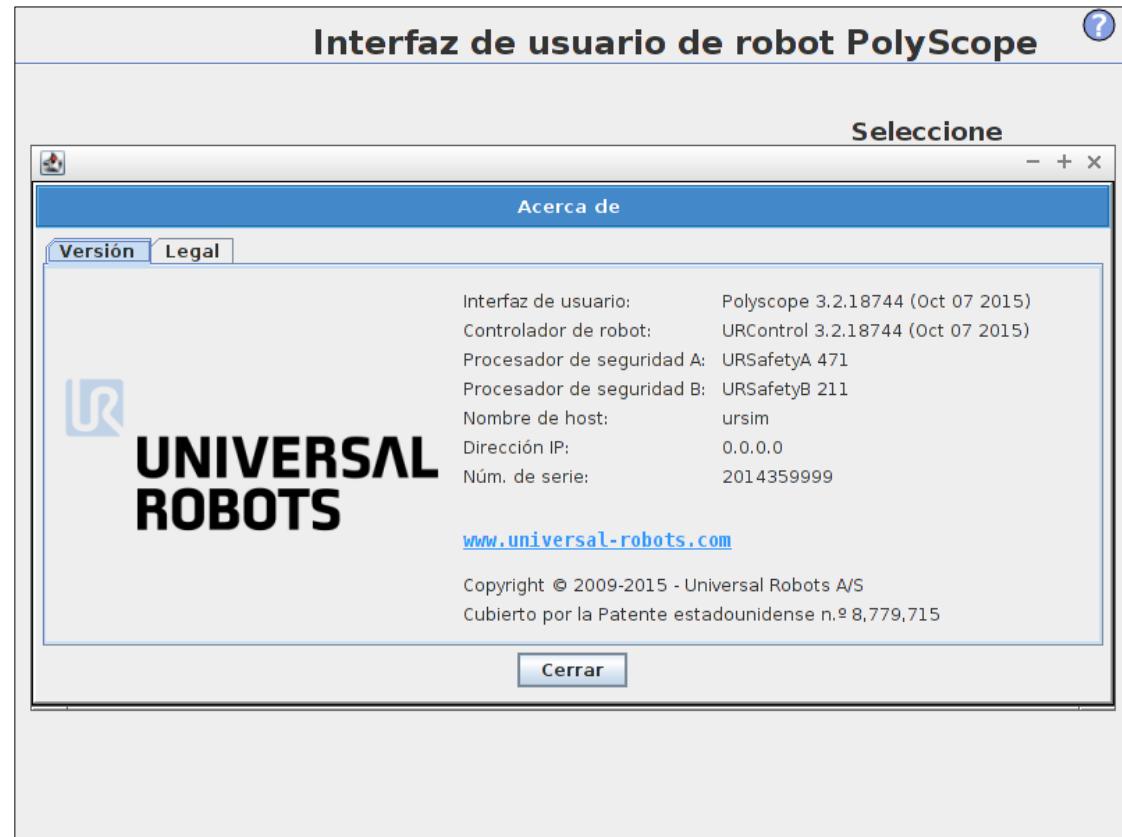
Introducción a PolyScope

- PolyScope
 - Desarrollado por UR
 - Actualizaciones gratuitas
- Sistema operativo
 - Debian Linux



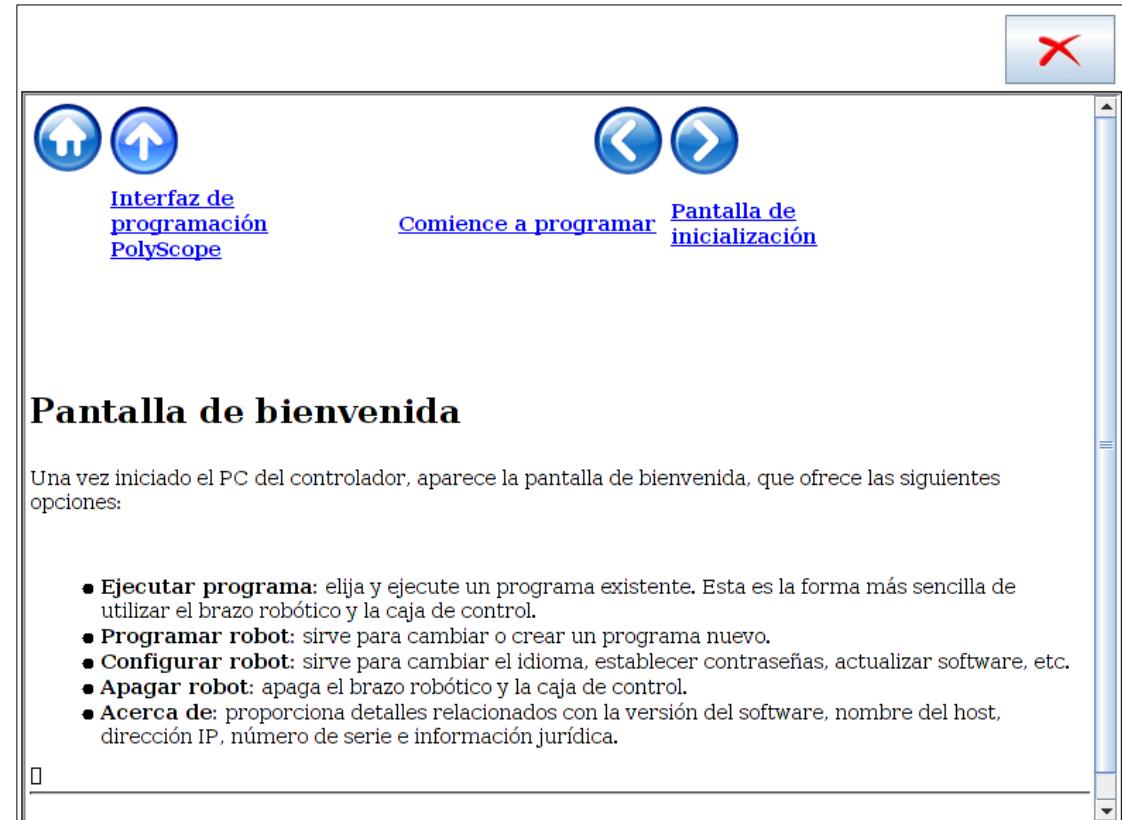
Acerca de

- Información
 - Número de serie
 - Versión de Software
 - Dirección IP



Manual en línea

- Características
 - Se muestra en el lenguaje seleccionado
 - Versión resumida del manual de software



Configurar robot

- Configurar robot
 - Ajusta parámetros de software



Configurar robot

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



Iniciar robot

- Estados del robot

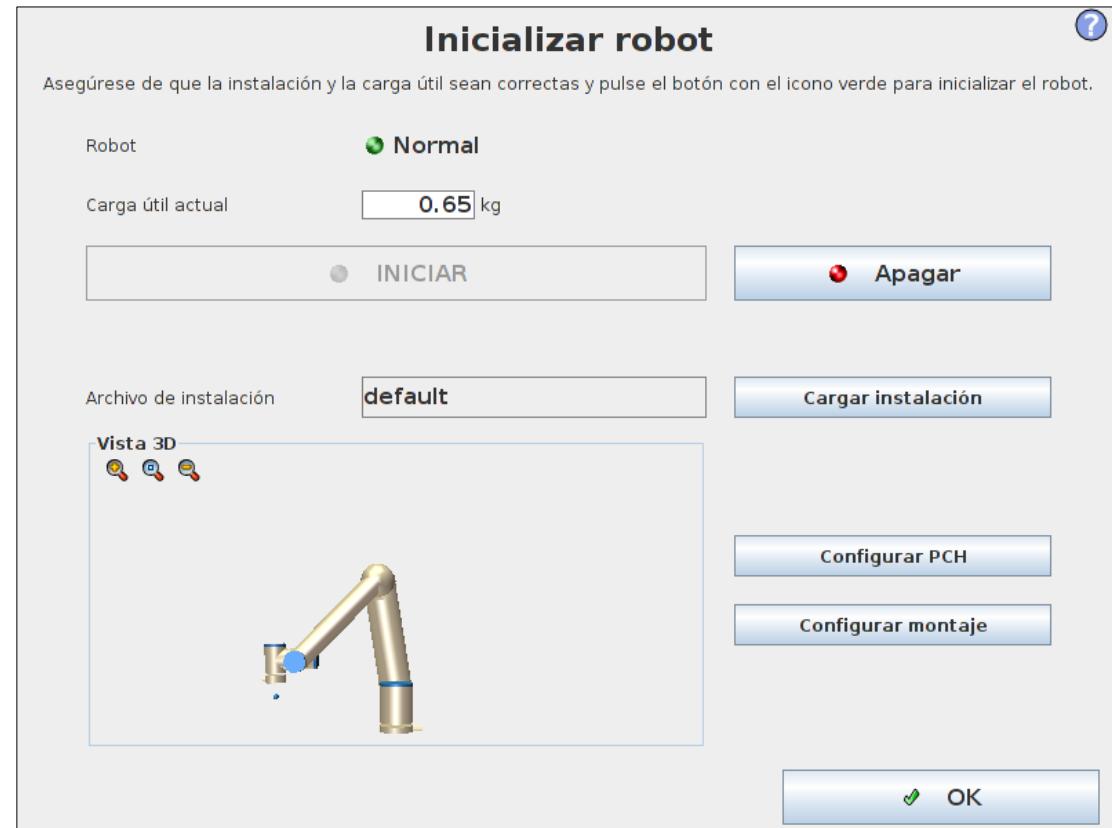
Estado	Tensión	Frenos
🔴 Apagado	OFF	Activos
🟡 Inactivo	ON	Activos
🟢 Normal	ON	Libres

- Iniciar robot

- Comprobar carga útil
- ENCENDER: Tensión ON
- INICIAR: Libera frenos

- Modo Retroceso

- Útil cerca de colisión



Idioma y unidades

- Idiomas
 - 20 idiomas
 - Programación en inglés
 - Mantiene las instrucciones en inglés
- Unidades
 - Sistema métrico
 - Sistema anglosajón



Actualizar robot

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



- Se realizará una demostración más adelante durante la sesión

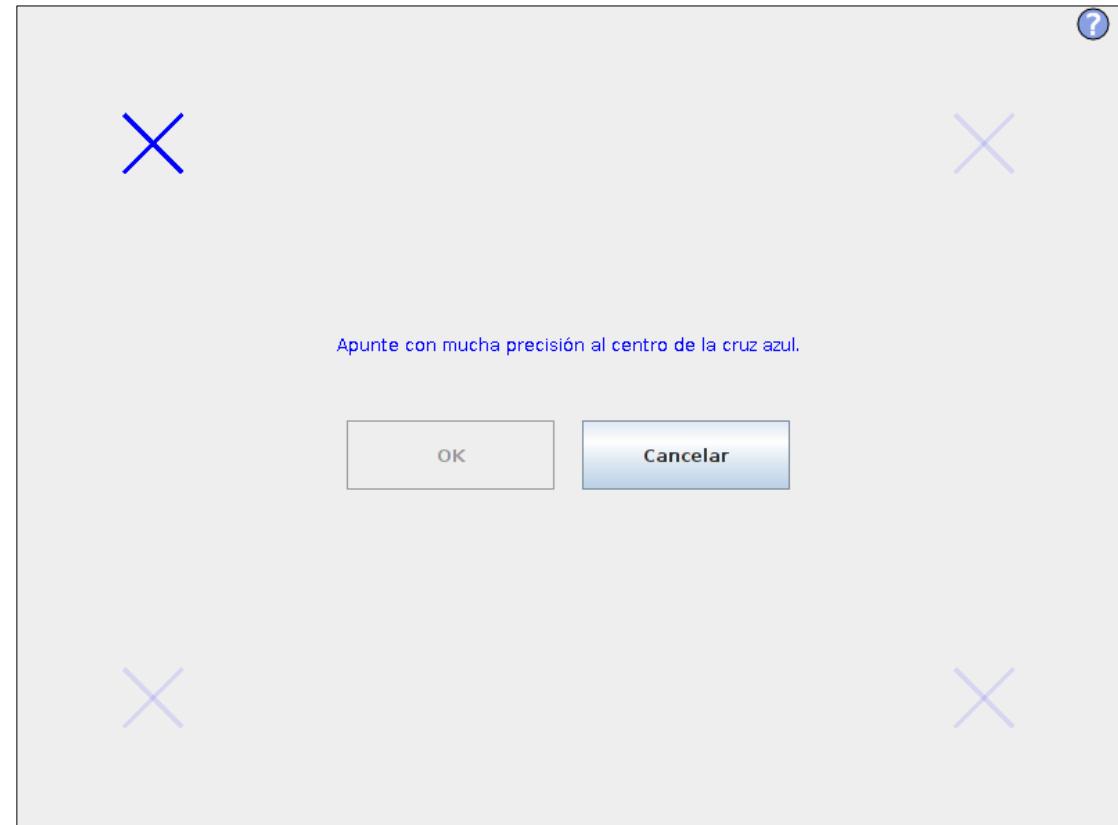
Fijar contraseña

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

The screenshot shows the 'Config. robot' software interface. On the left, there is a vertical menu with several options: 'Inicializar robot', 'Idioma y unidades', 'Actualizar robot', 'Fijar contraseña' (which is highlighted with a blue border), 'Pantalla Calibrar', 'Configurar red', and 'Ajuste de hora'. To the right of the menu, there are two main sections. The top section is titled 'Cambiar contraseña del sistema' and contains a descriptive text about how passwords protect changes made to robot functions and behavior. It includes fields for 'Contraseña' and 'Confirmar contraseña' with a 'Aplicar' button. The bottom section is titled 'Cambiar contraseña de seguridad' and contains instructions for setting a security password. It includes fields for 'Introducir contraseña actual', 'Contraseña', and 'Confirmar contraseña' with a 'Aplicar' button. There is also a 'Atrás' (Back) button at the bottom of the section.

Calibrar la pantalla

- Calibrado de la pantalla táctil
 - Marque las cuatro esquinas para calibrar



Configurar red

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



- Se realizará una demostración más adelante durante la sesión

Ajuste de hora y fecha

- Hora
 - Formato de hora
 - 24 horas
 - 12 horas
- Fecha
 - Formato de fecha



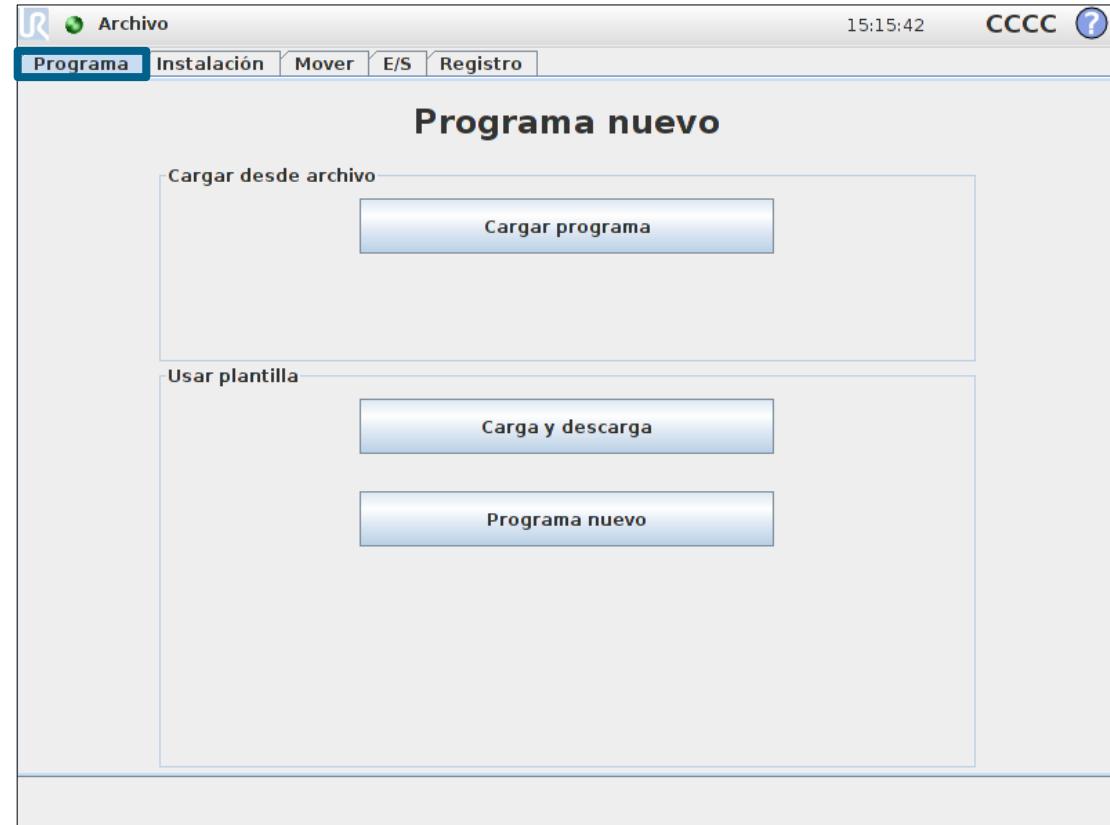
Programar el robot

- Programar robot
 - Resumen de pestañas



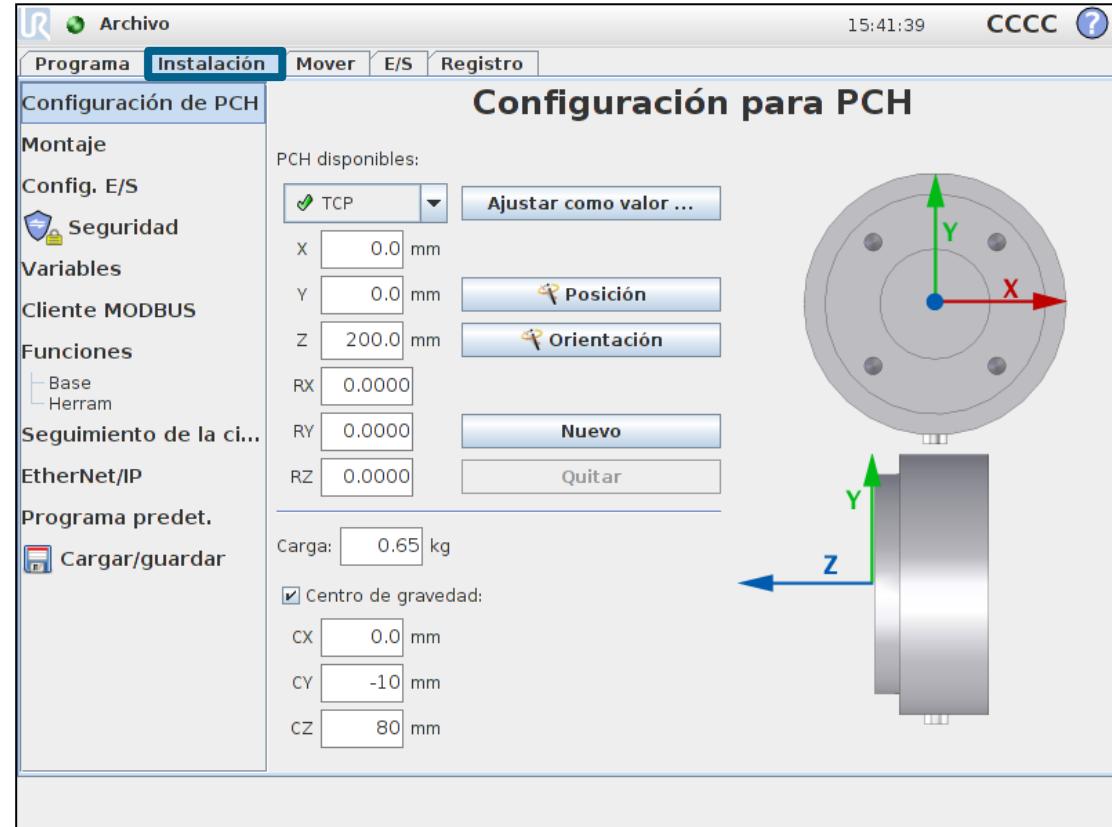
Pestaña Programa

- **Programa**
 - Cargar programa existente
 - Crear programa nuevo



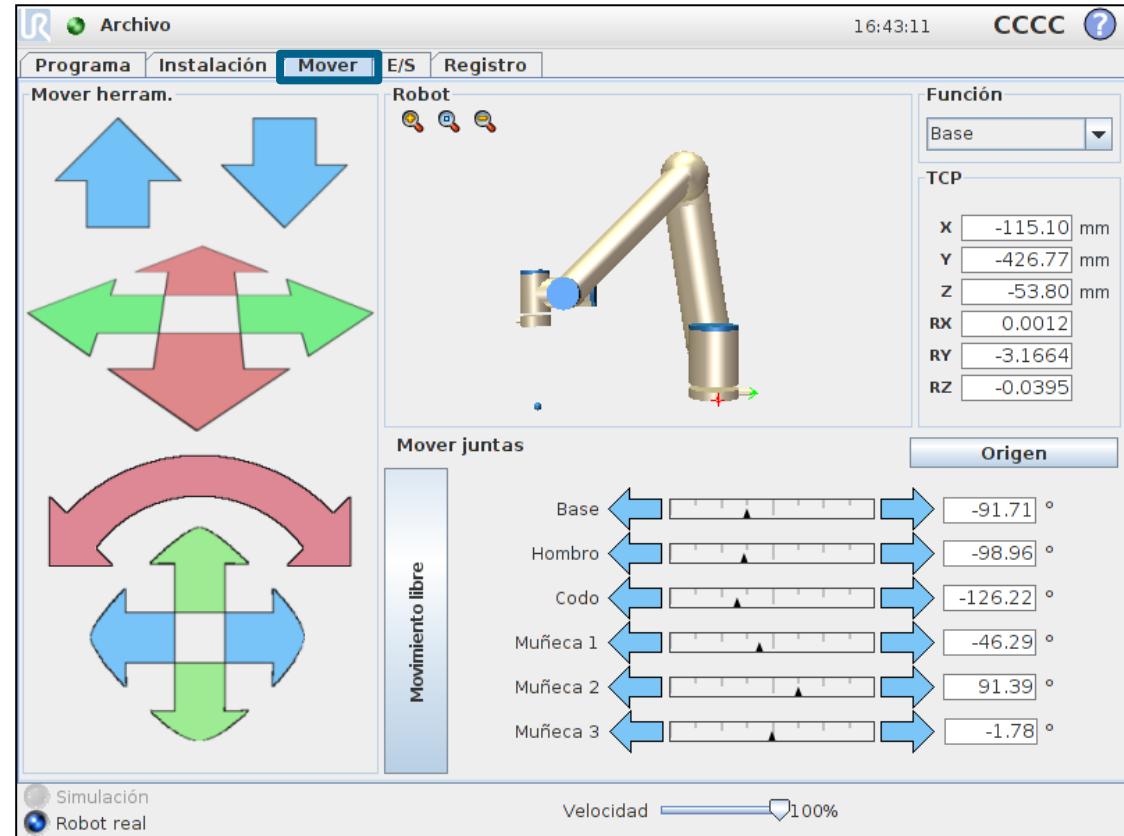
Pestaña Instalación

- Configuraciones
 - Configuración entorno
 - Configuración Seguridad



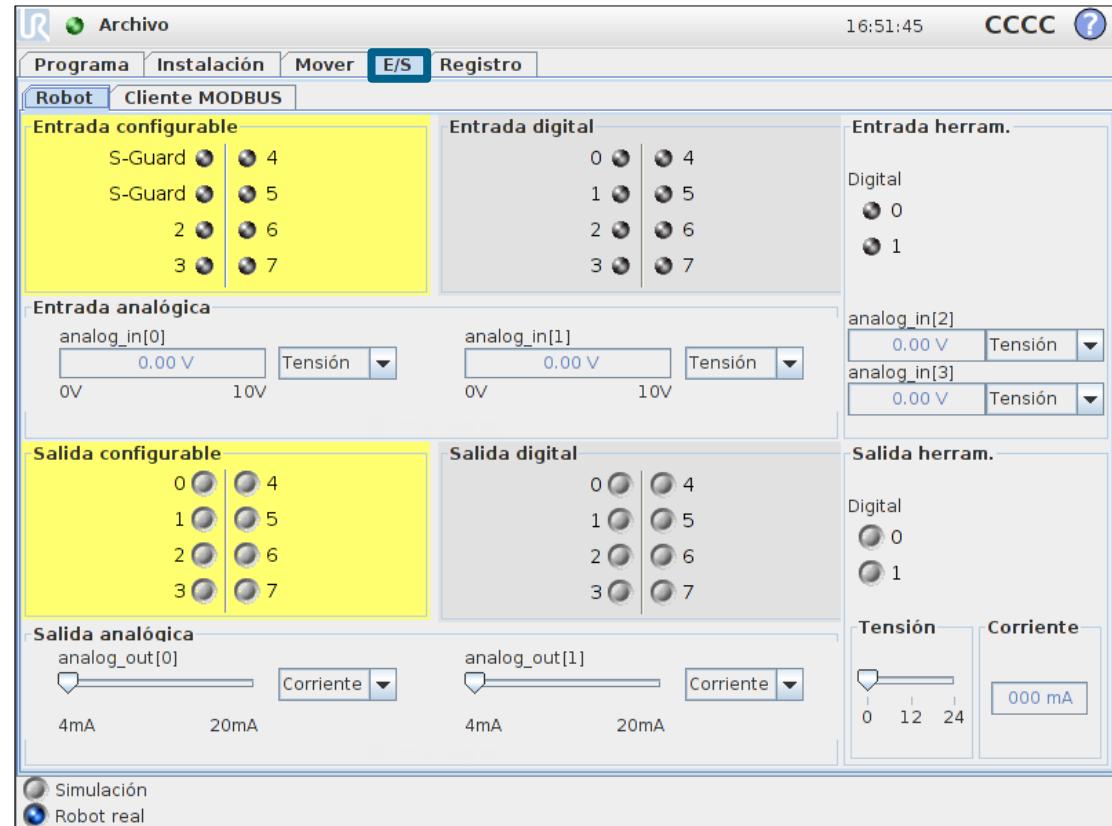
Pestaña Mover

- **Mover**
 - Desplazamiento manual del robot respecto la función seleccionada
 - Muestra la posición del PCH respecto la función seleccionada
 - Desplazamiento manual de cada junta individualmente
 - Muestra el ángulo de giro de cada una de las juntas



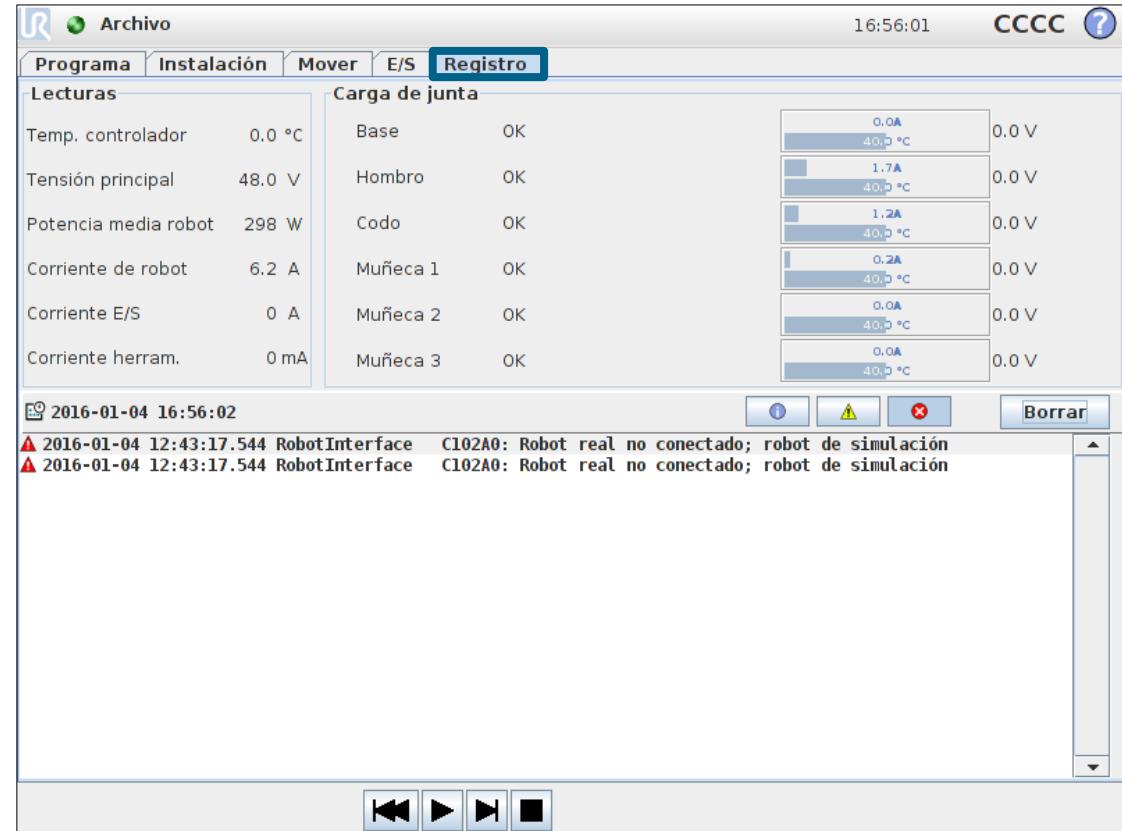
Pestaña E/S

- E/S
 - Monitoreo de señales
 - Activación de señales
 - Configuración de señales analógicas



Pestaña Registro

- Estado
 - Controlador
 - Juntas
- Registro de eventos
 - Mensajes de información
 - Advertencias
 - Errores



Moviendo el robot

- Desplazamiento

- Por junta individual

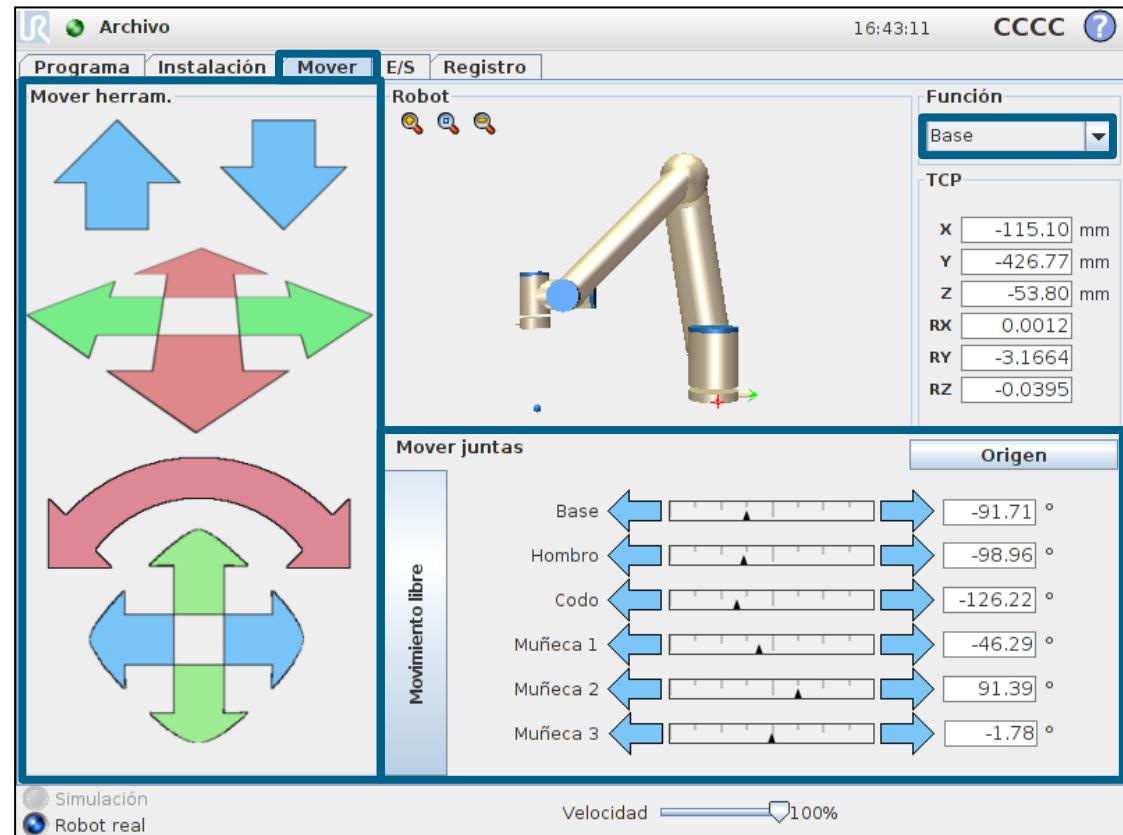
- Cambia la orientación del PCH
 - Límites mín./máx.
 - Muestra el ángulo de giro

- Lineal

- Orientación fija del PCH cuando se desplaza en XYZ
 - Relativo a la Función elegida
 - Ver
 - Base
 - Herramienta

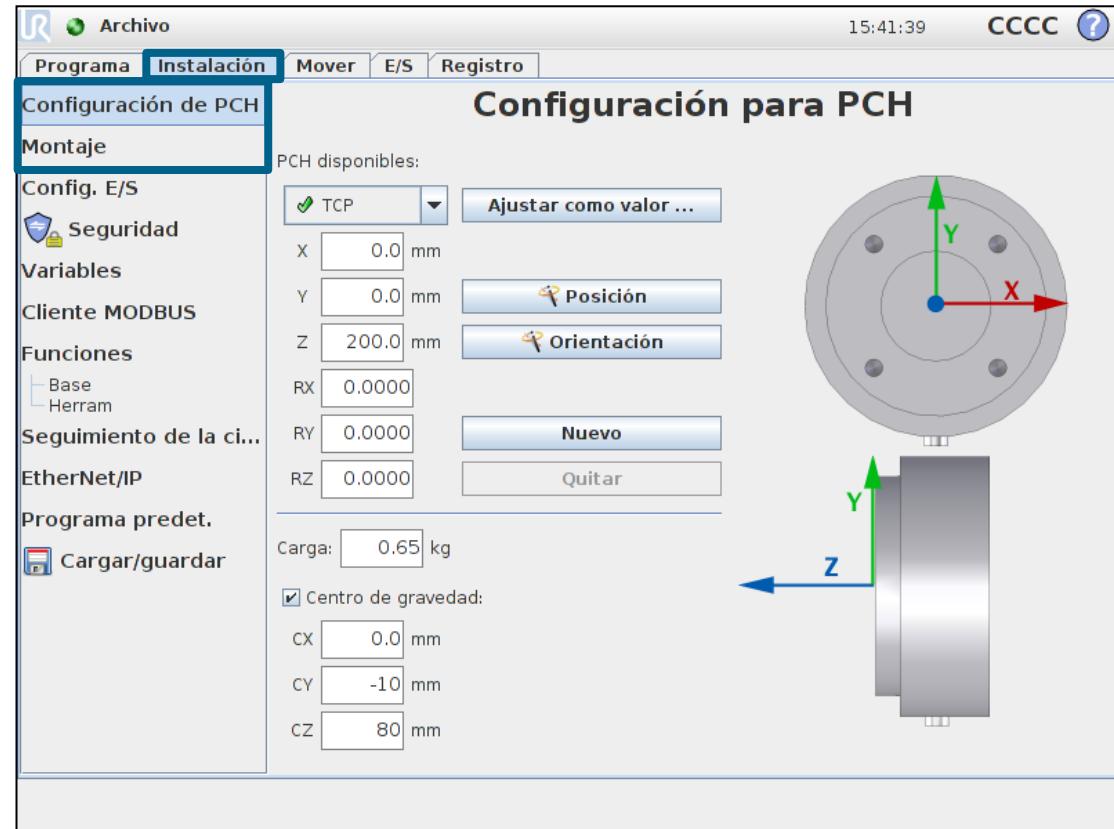
- Movimiento libre

- Botón Movimiento libre en TP
 - Botón Movimiento libre en GUI



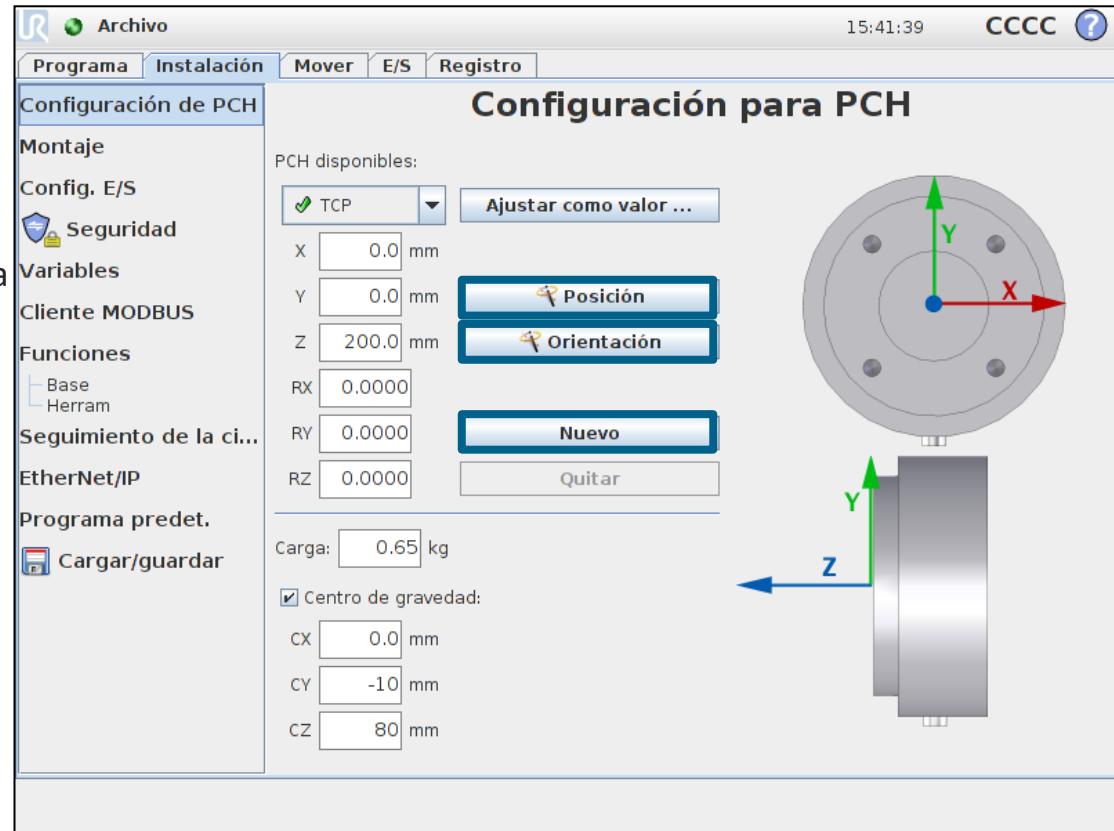
Configuración inicial

- Configuraciones iniciales
 - Configuración de PCH
 - Configuración Montaje



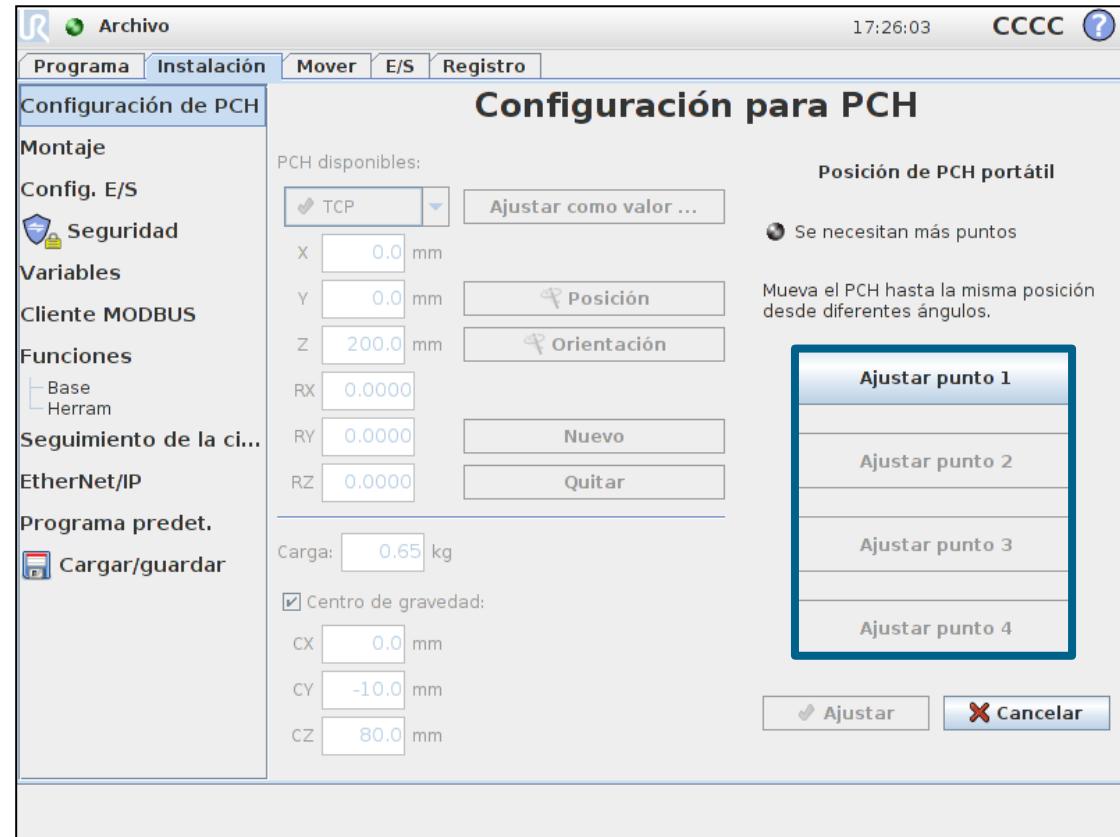
Configuración del PCH

- Punto Central de Herramienta (PCH)
 - 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 PCH
- Carga útil
 - Peso de la herramienta
 - Centro de gravedad



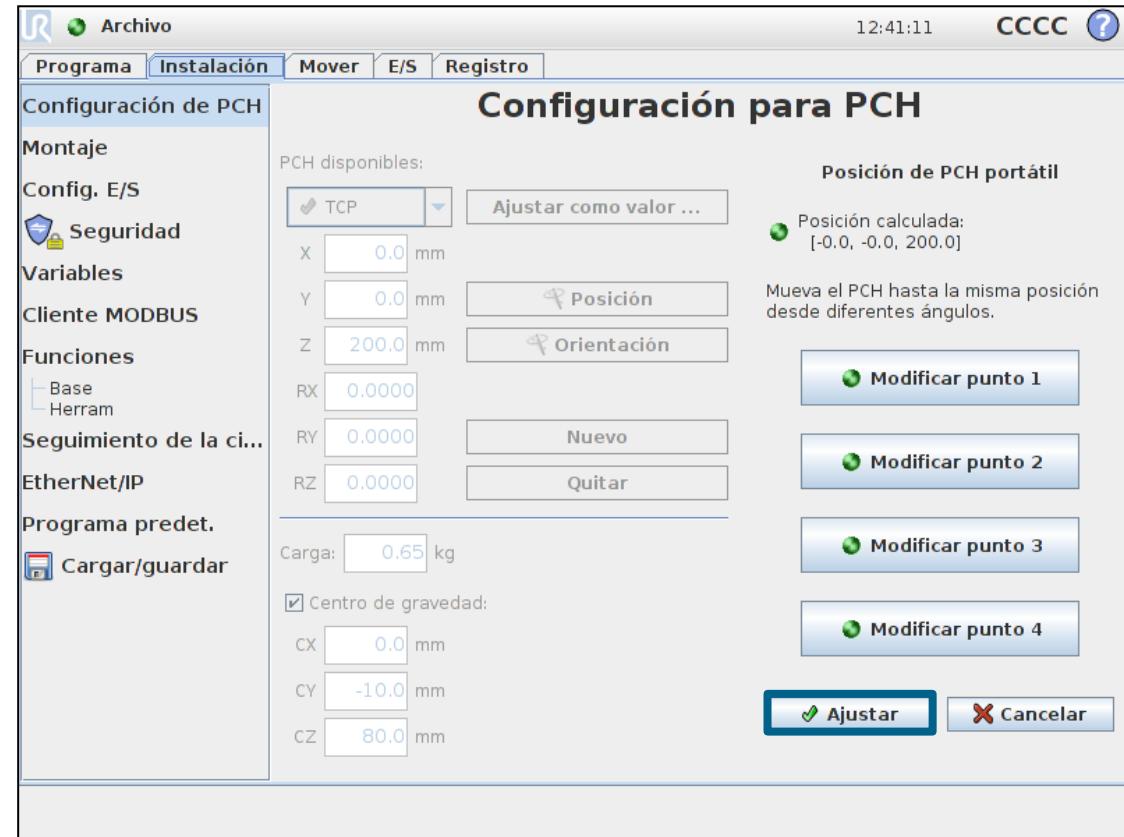
Configuración del PCH

- Cálculo Posición del PCH
 - Calcula la posición del PCH según 3-4 puntos ajustados manualmente



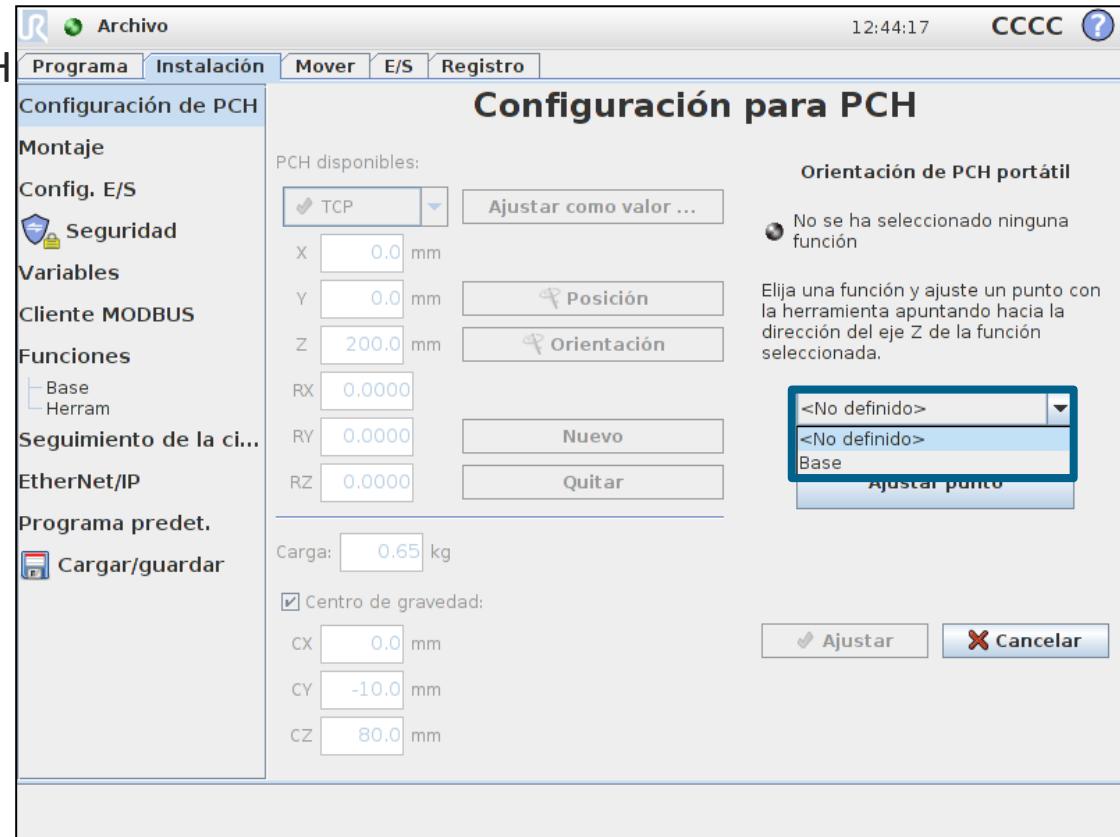
Configuración del PCH

- Cálculo Posición del PCH
 - Calcula la posición del PCH según 3-4 puntos ajustados manualmente



Configuración del PCH

- Cálculo Orientación del PCH
 - Elegir una función
 - Ajustar un punto



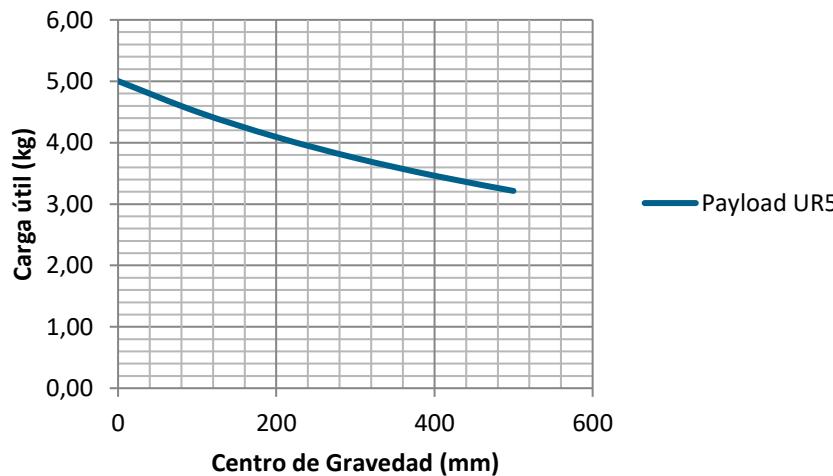
Configuración del PCH

- Cálculo Orientación del PCH
 - Elegir una función
 - Ajustar un punto

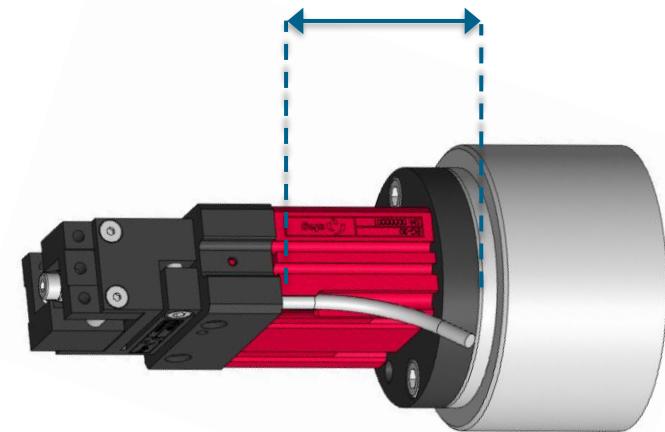


Cálculo de la Carga útil

- Ejemplo:
 - Cómo calcular la carga máxima permitida



Centro de Gravedad (CoG) = 100 mm



- Fórmula simplificada:

$$\text{Carga máx.} = \frac{4.5}{(0.9 + CoG)} = \frac{4.5}{(0.9 + 0.1)} = 4.50 \text{ kg}$$

Montaje

- Configurar cómo está montada la base del robot



Cargar/guardar

- **Guardar Instalación**

- Todas las configuraciones en Instalación se guardan en un fichero

- **Cargar Instalación**

- Carga un fichero de Instalación guardado

- **Nombre por defecto**

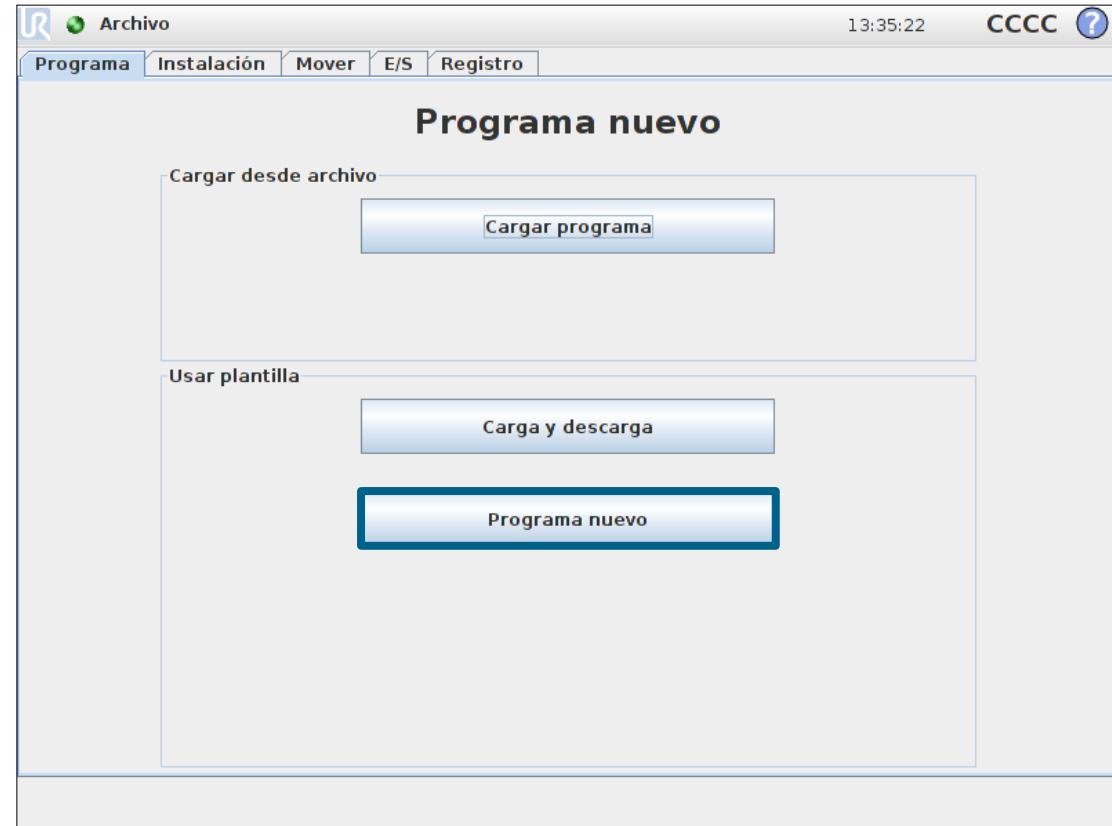
- default.installation
- Se carga cada vez que se reinicia el robot



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

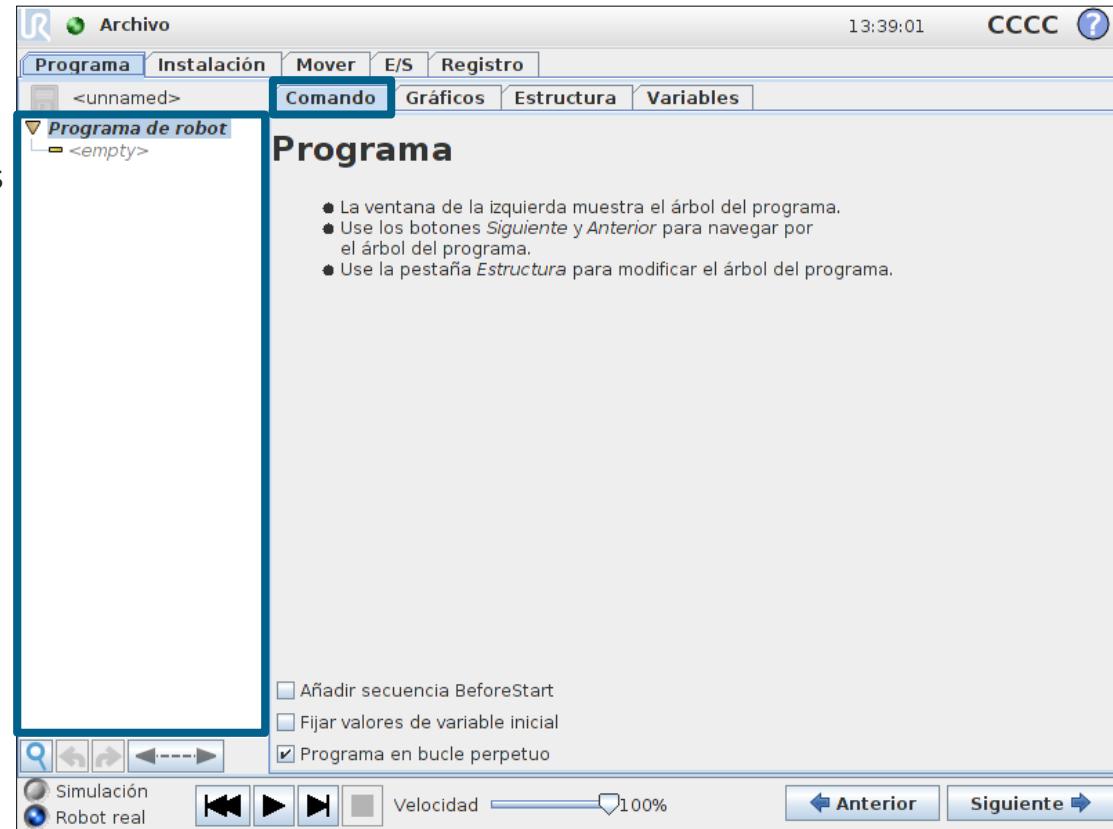
Crear un programa nuevo

- Usar plantilla
 - Programa nuevo



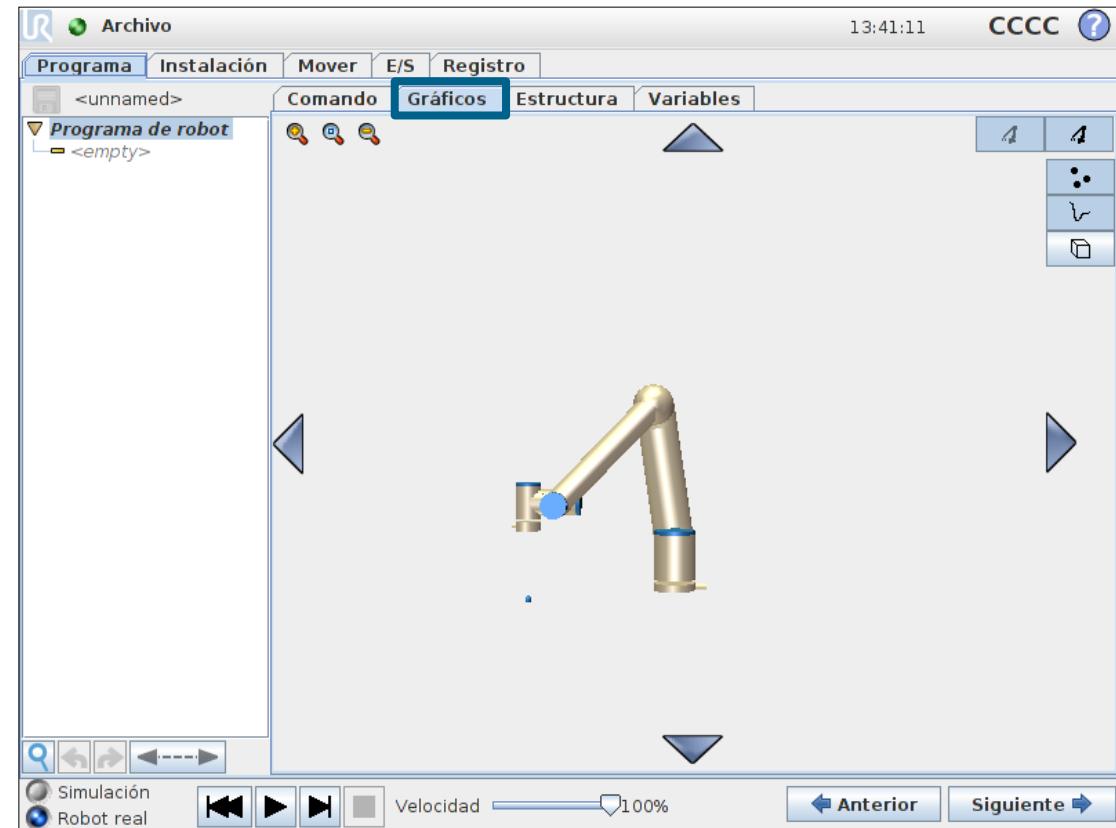
Pestaña Comando

- Árbol de programa
 - Estructura del programa
 - Contiene todos los comandos
 - Se ejecuta secuencialmente
- Pestaña comando
 - Edición del comando seleccionado
 - Configura parámetros generales del programa



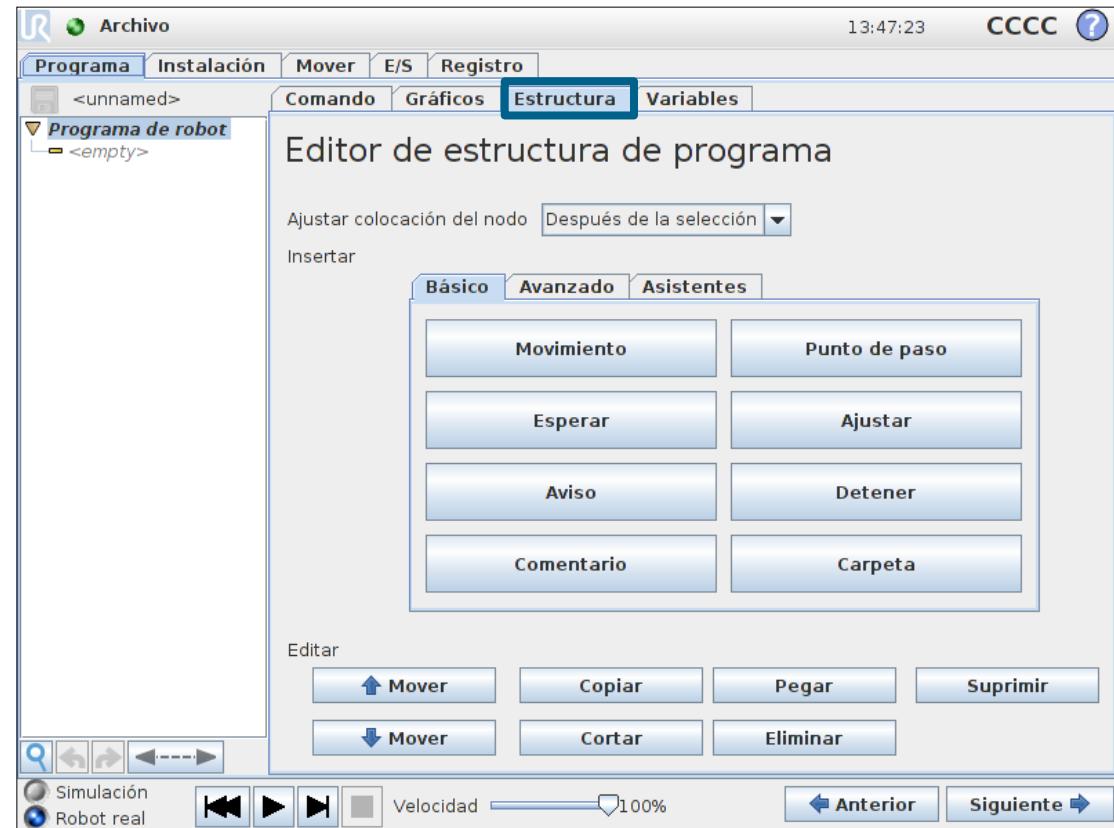
Pestaña Gráficos

- Pestaña de Gráficos
 - Posición actual
 - Posiciones aprendidas
 - Trayectorias



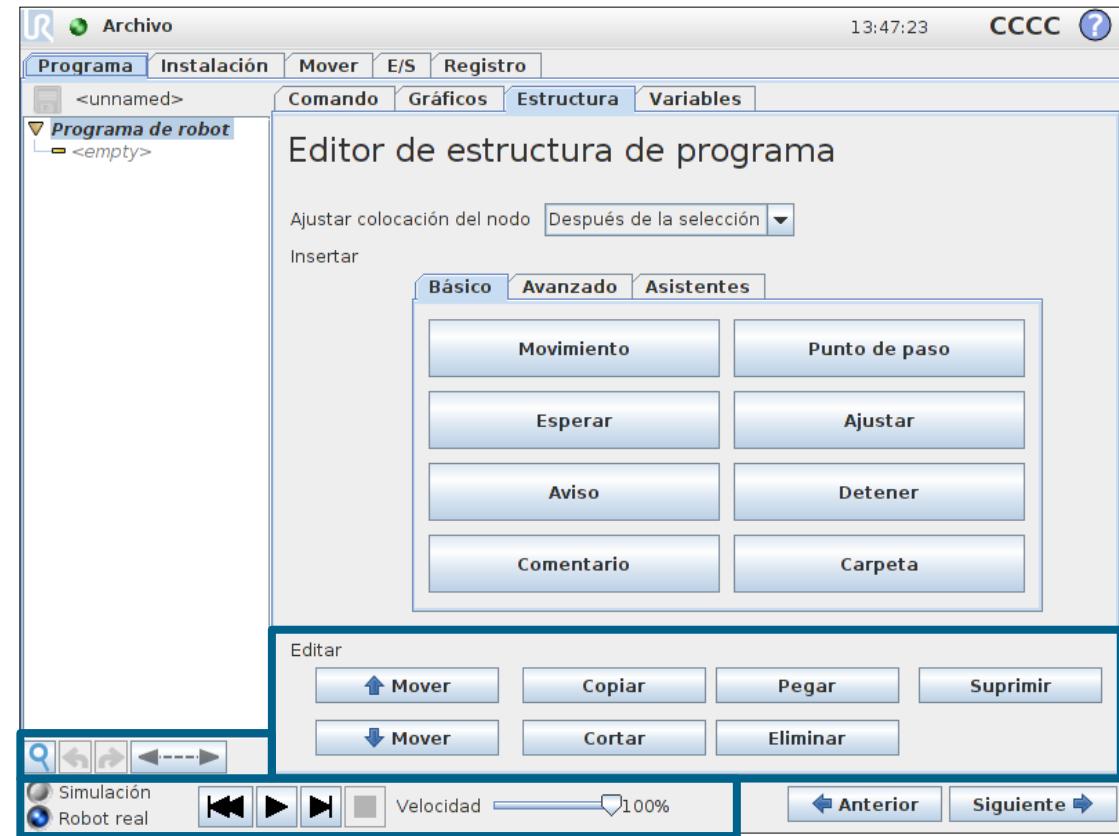
Pestaña Estructura

- Pestaña de estructura
 - Contiene todos los comandos
 - Se divide en
 - Básico
 - Avanzado
 - Asistentes



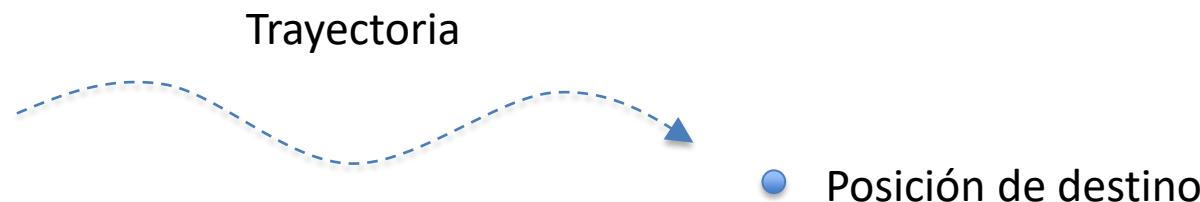
Área Editar / Panel ejecución

- Área de edición
 - Editar y organizar el árbol de programa
 - Herramientas de edición
 - Función de Suprimir
- Panel
 - Controla la ejecución de programa
 - Simulación / Robot real
- Deshacer/Rehacer
 - Deshacer/Rehacer cambios en el programa
 - Botón salto a programa
 - Función de búsqueda
 - Extender la ventana del árbol de programa



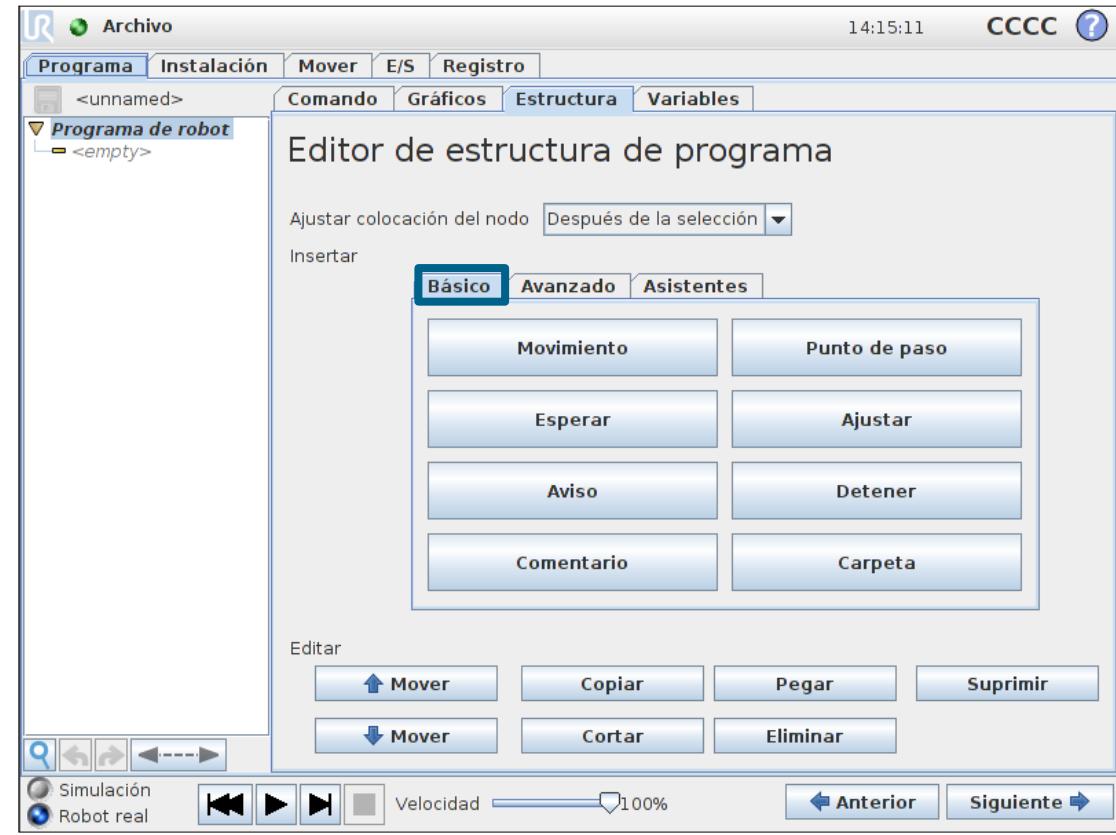
Insertar movimiento

- **Punto_de_paso** especifica la posición de destino
- **Movimiento** especifica la trayectoria que seguirá el robot para llegar a la posición de destino



Comandos básicos

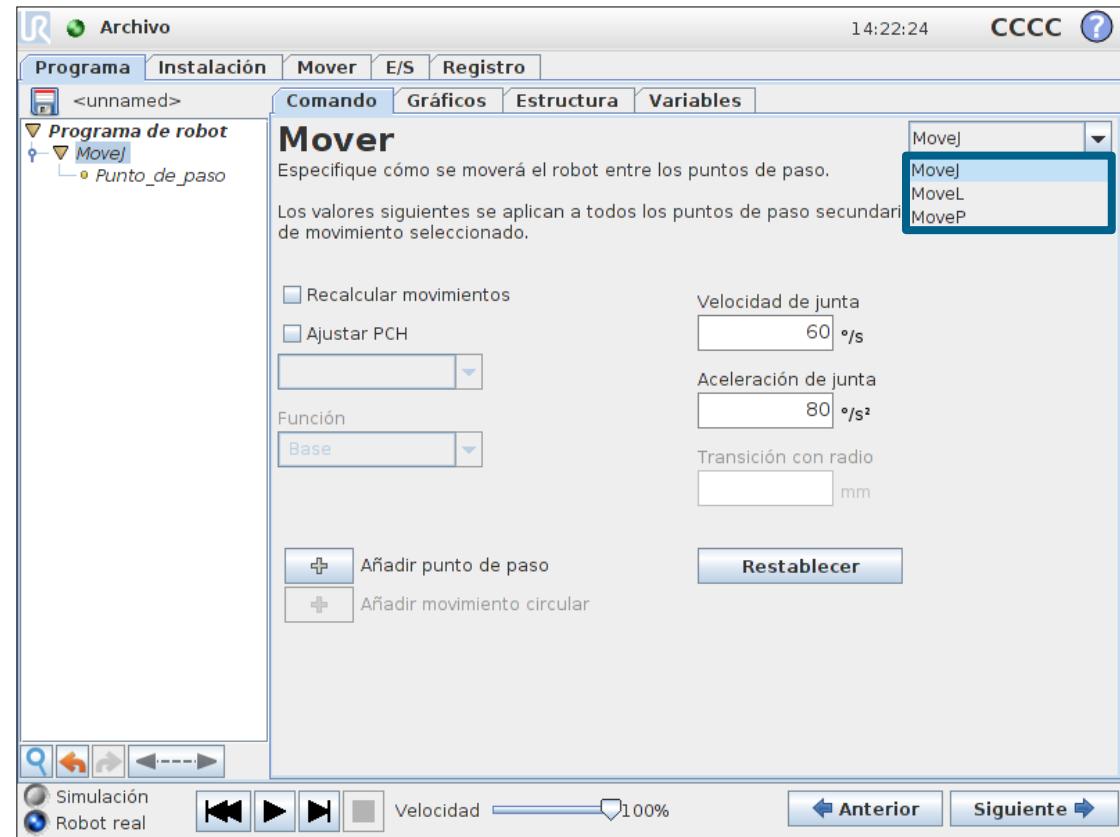
- Comandos básicos
 - Realizar operaciones sencillas



- Insertar comando Movimiento

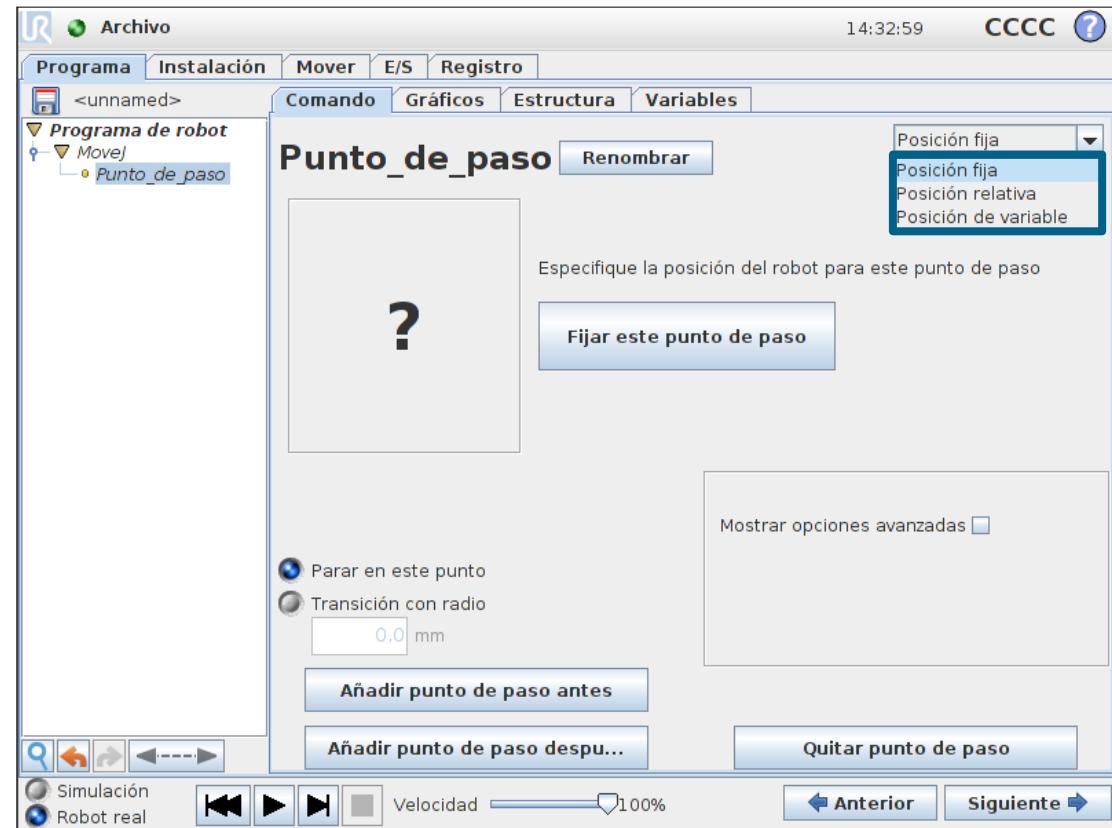
Comandos Movimiento

- Tipos de movimiento
 - MoveJ (*defecto*)
 - MoveL
 - MoveP
 - MoveC
- Parámetros compartidos
 - Velocidad de junta
 - Defecto: 60 °/s
 - Aceleración de junta
 - Defecto: 80 °/s²
- Se añade automáticamente un Punto_de_paso cuando se inserta un comando de Movimiento



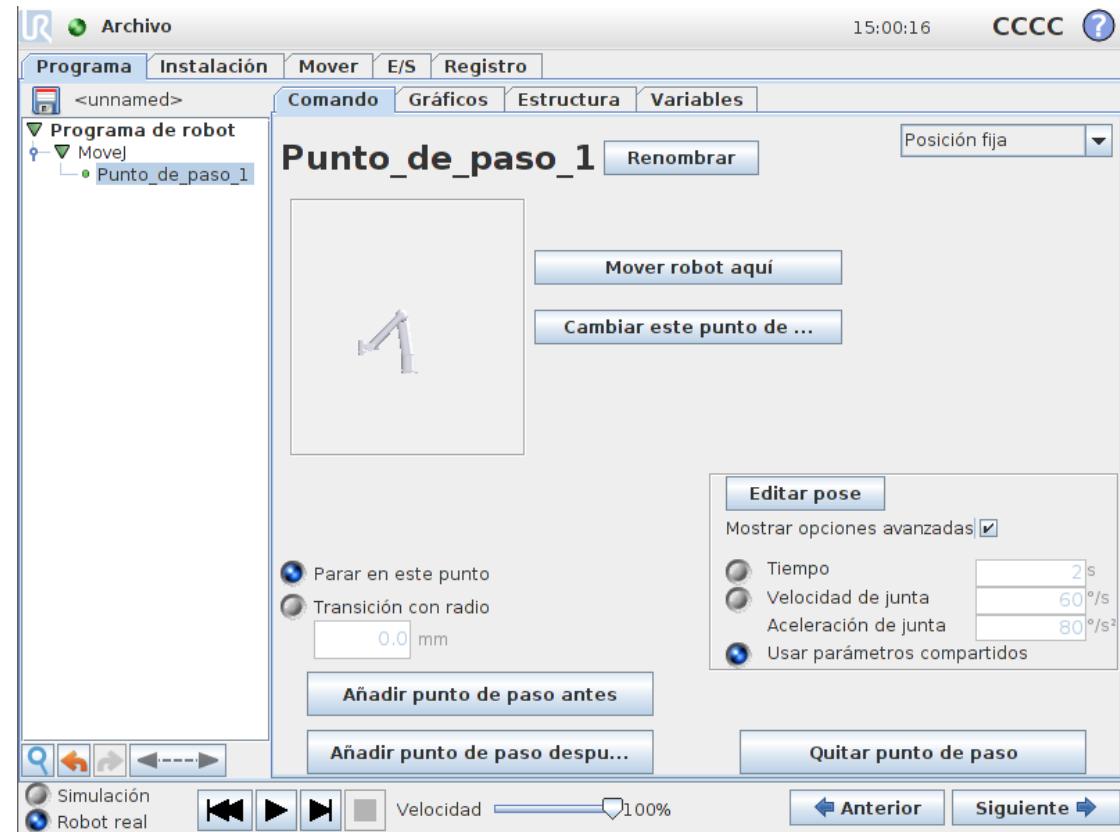
Punto_de_paso

- Tipos de Punto_de_paso
 - Fijo (*defecto*)
 - Relativo
 - De variable
- Enseñar Punto_de_paso
 - Pulsar *Fijar este punto de paso*
 - Llevar el robot a la posición deseada y pulsar OK para grabarla



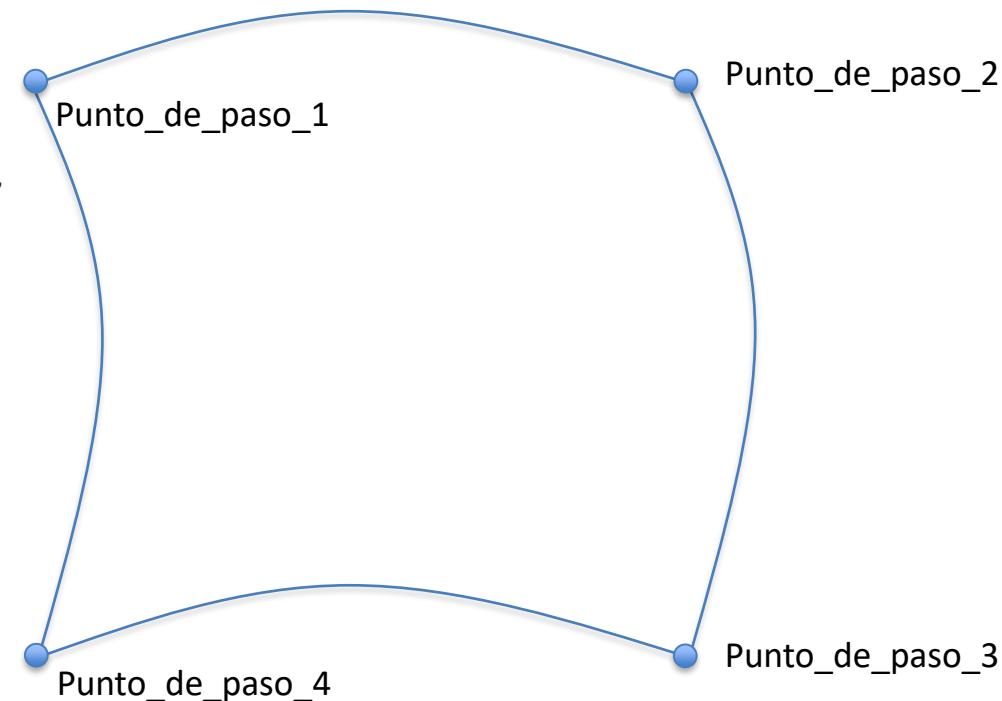
Punto_de_paso

- Opciones
 - Renombra Punto_de_paso
 - Mejora la lectura y comprensión del programa
 - Mover robot aquí
 - Mueve el robot a la posición programada
 - Cambiar este punto de paso
 - Modifica la posición



MoveJ

- Movimiento de juntas
 - Sin interpolación
 - Tipo de movimiento más rápido
 - Útil en movimientos con espacio "libre"



Programa de robot

```
MoveJ  
    Punto_de_paso_1  
    Punto_de_paso_2  
    Punto_de_paso_3  
    Punto_de_paso_4
```

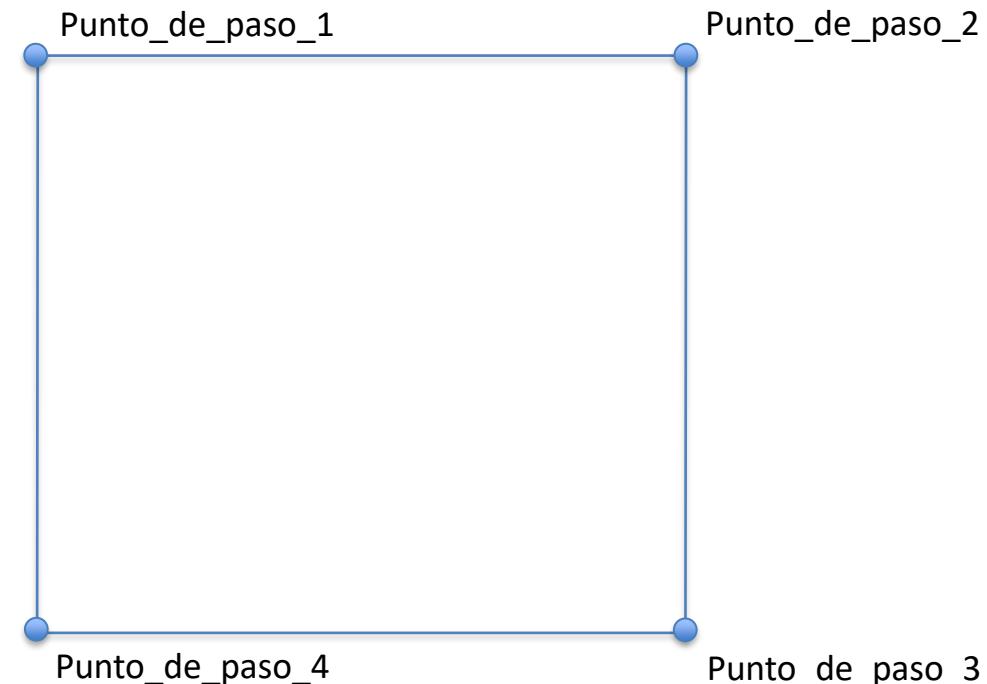
- Guardar programa de ejemplo como movej.urp

MoveL

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

Programa de robot

```
MoveL  
    Punto_de_paso_1  
    Punto_de_paso_2  
    Punto_de_paso_3  
    Punto_de_paso_4
```



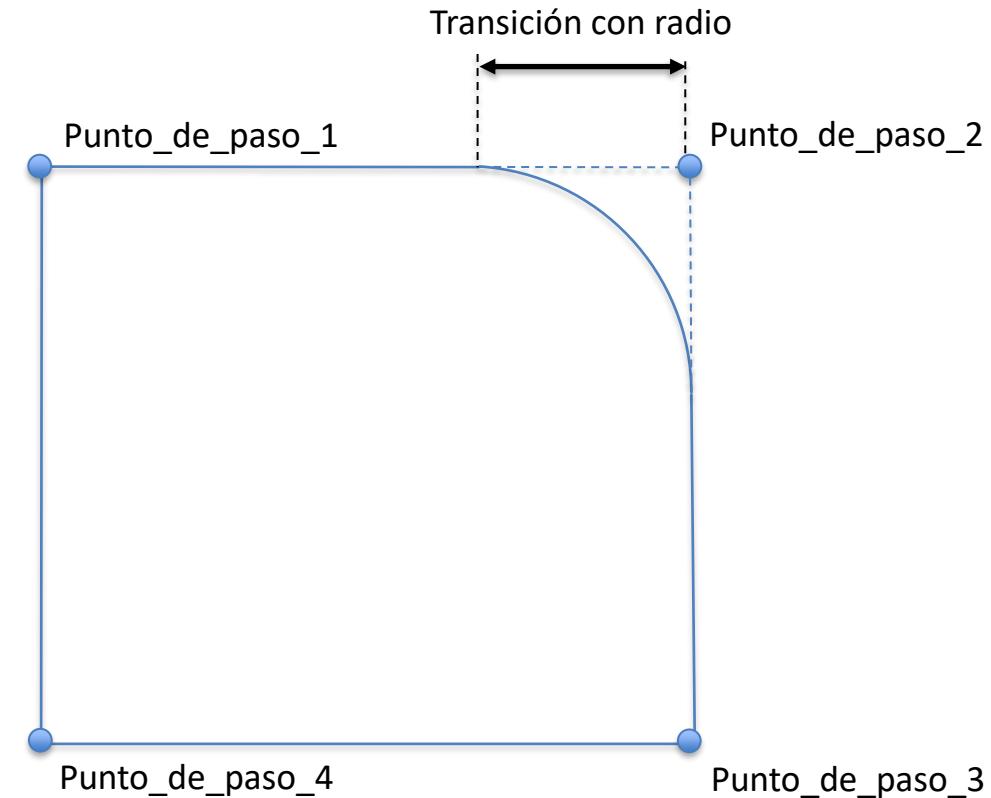
- Guardar programa de ejemplo como movel.urp

MoveL con radio de transición

- Transición con radio
 - Movimiento continuo
 - No se detiene en el punto de paso

Programa de robot

```
MoveL  
    Punto_de_paso_1  
    Punto_de_paso_2 (r=25)  
    Punto_de_paso_3  
    Punto_de_paso_4
```



- Guardar programa de ejemplo como move_with_blend.urp

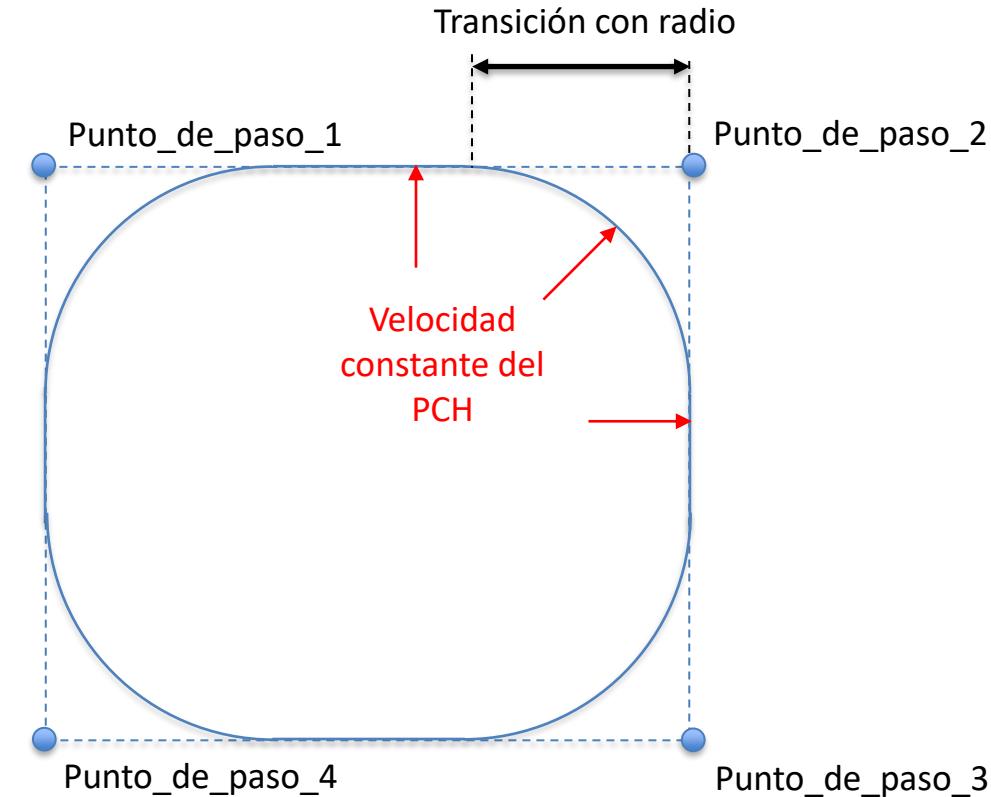
MoveP

- Movimiento de Proceso
 - Aplicaciones de proceso
 - Movimiento lineal
 - Velocidad de PCH constante
 - Transición con radio compartido

Programa de robot

MoveP

```
Punto_de_paso_1  
Punto_de_paso_2  
Punto_de_paso_3  
Punto_de_paso_4
```

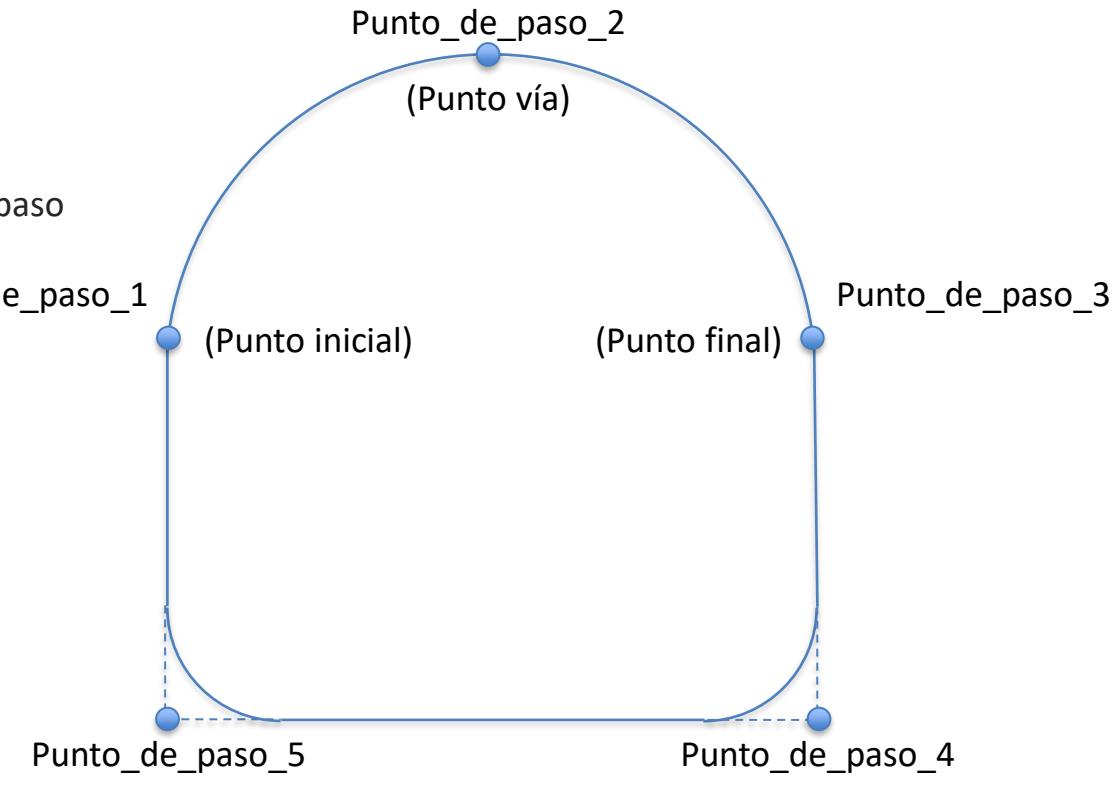


- Guardar programa de ejemplo como movep.urp

MoveC

- **Movimiento Circular**

- Trayectoria circular del PCH
- Son necesarios tres puntos de paso
 - Punto inicial
 - Punto vía
 - Punto final
- El radio se calcula de forma automática



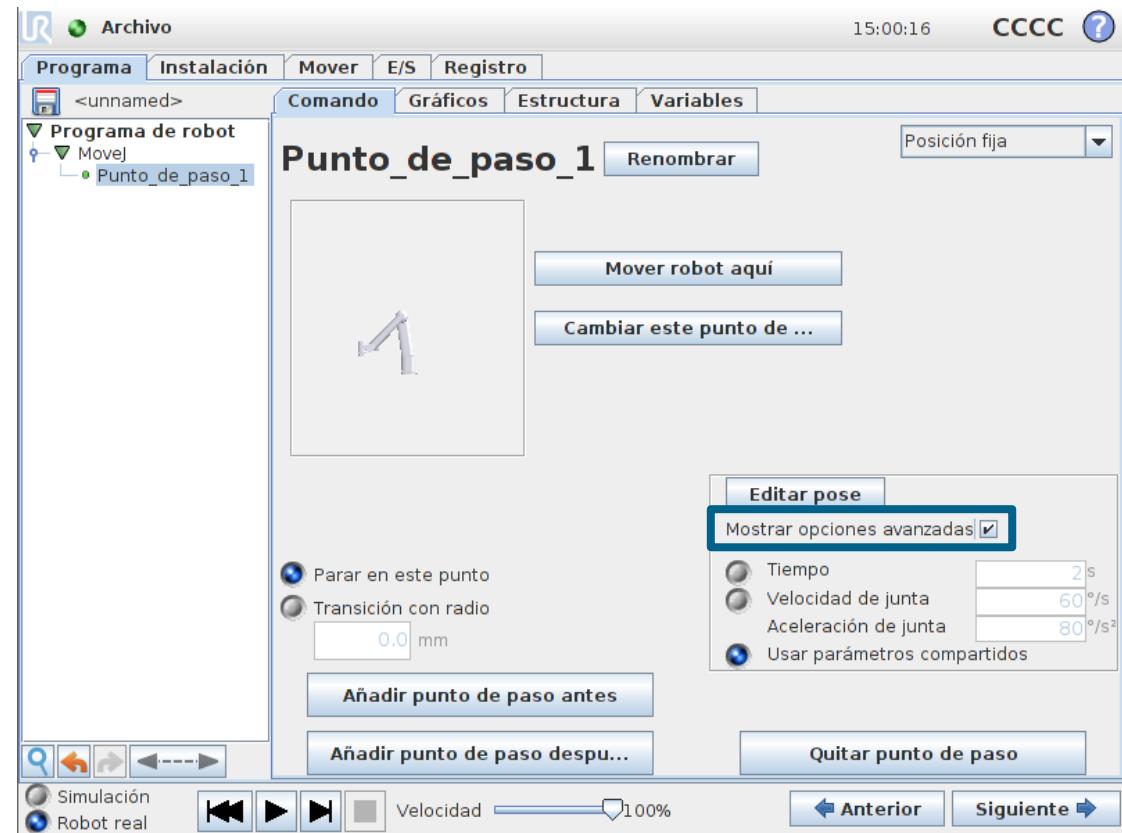
Programa de robot

```
MoveP  
  Punto_de_paso_1  
  CircleMove  
    Punto_de_paso_2  
    Punto_de_paso_3  
  Punto_de_paso_4  
  Punto_de_paso_5
```

- Guardar programa de ejemplo como movec.urp

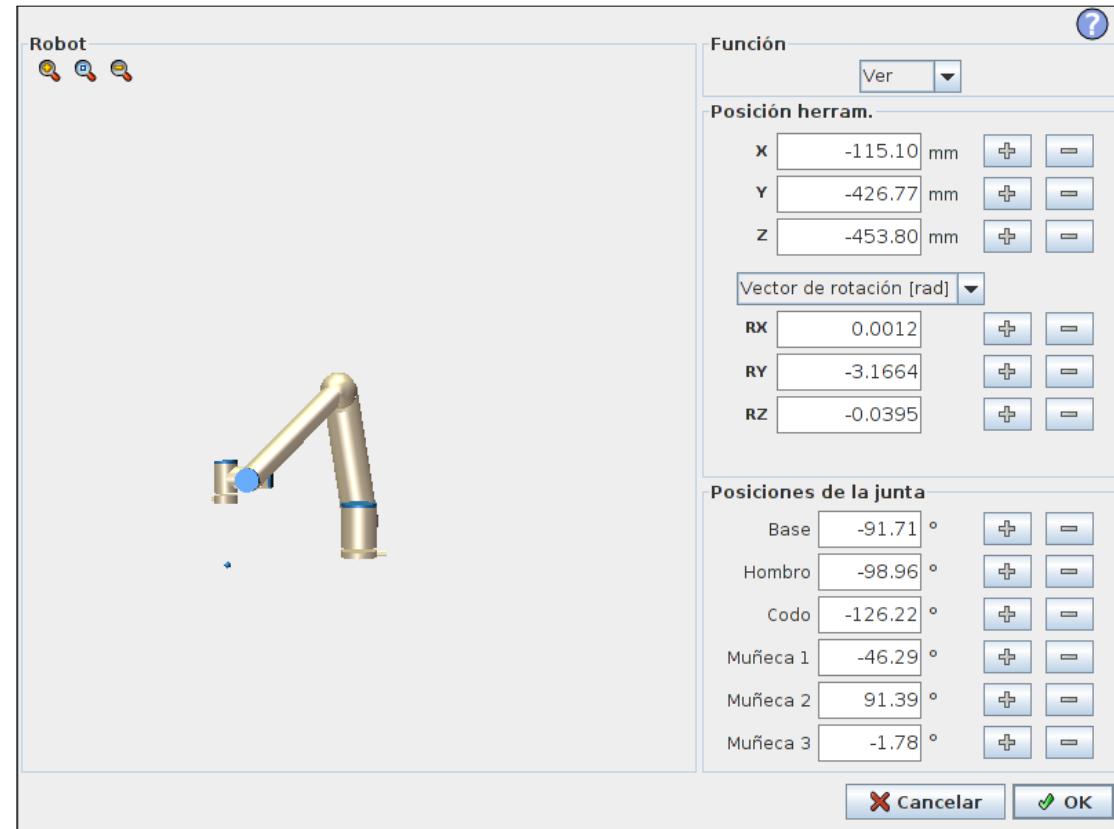
Opciones avanzadas

- Opciones avanzadas
 - Configuración individual
 - Velocidad & Aceleración
 - Tiempo
 - Editar pose
 - Introducir valores del Punto_de_paso



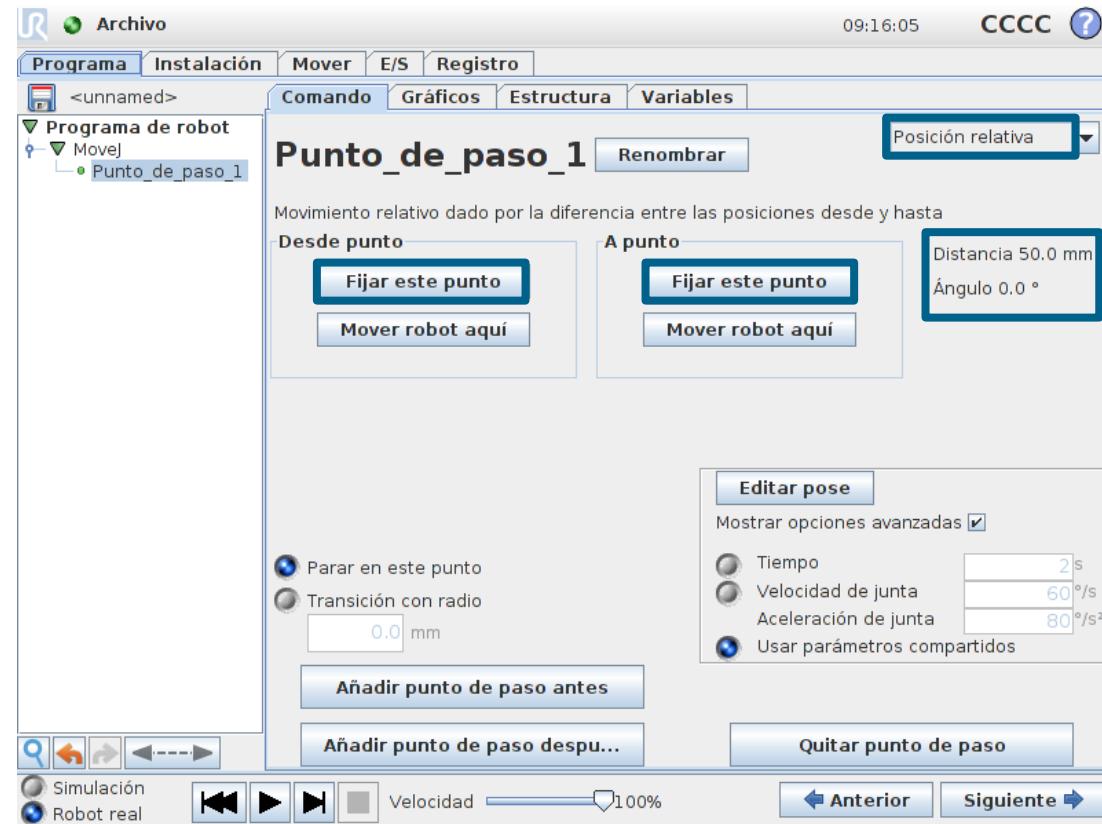
Editor de posiciones

- Modifica
 - Valores absolutos
 - Añadir/Sustraer a los valores
- Posiciones de juntas
 - Valores angulares en *grados*
- Posición del PCH
 - Valor cartesiano en *mm.*
 - Definir unidades de rotación



Punto_de_paso relativo

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

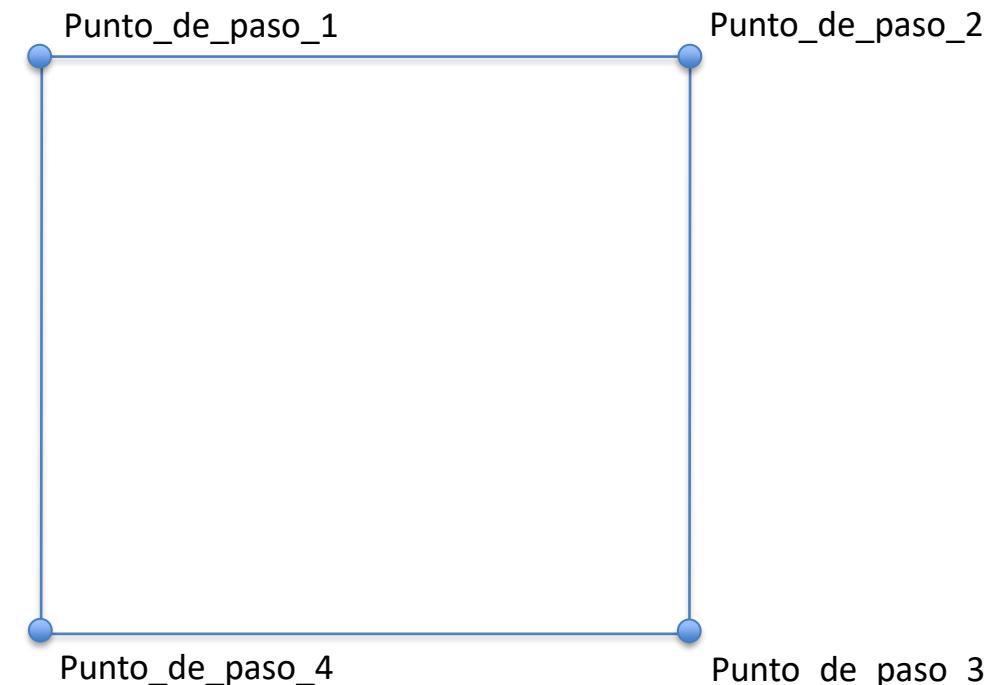


Punto_de_paso relativo

- Definir puntos de paso relativos
 - Definido por dos puntos de paso
 - Pueden definirse en cualquier posición del área de trabajo del robot

Programa de robot

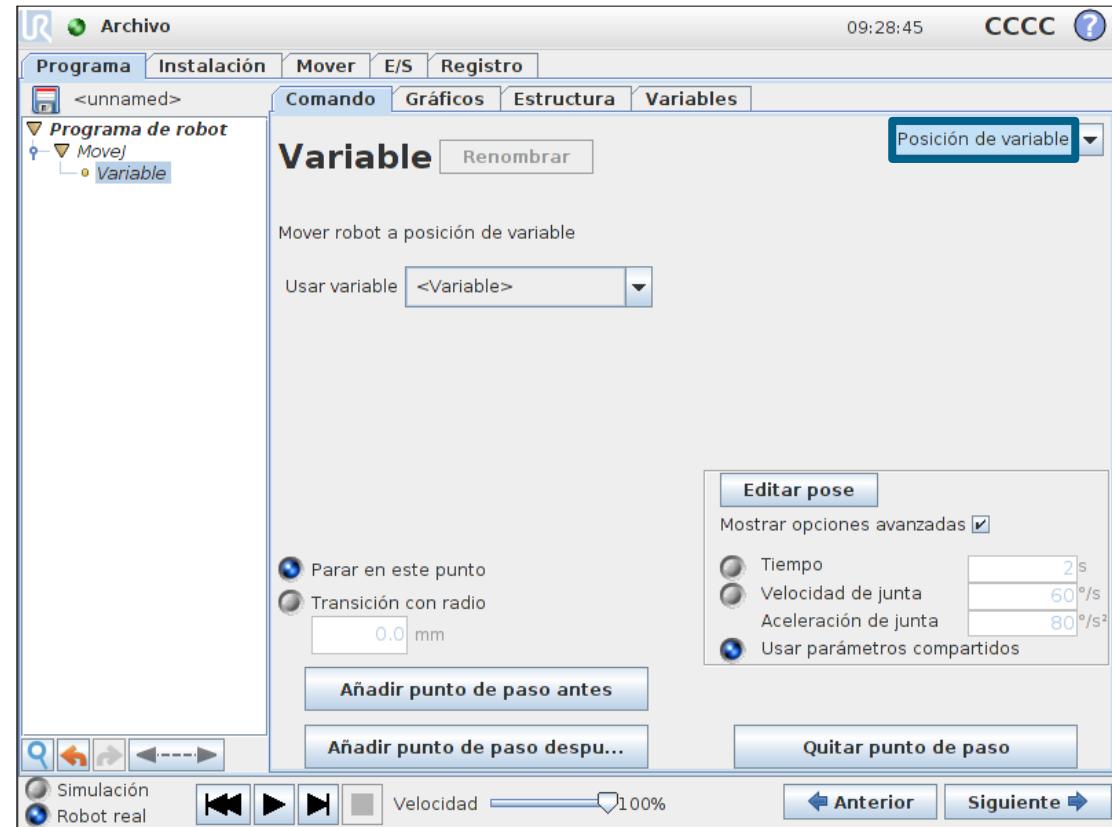
```
MoveL  
    Punto_de_paso_1  
    Punto_de_paso_2 rel. pos  
    Punto_de_paso_3 rel. pos  
    Punto_de_paso_4 rel. pos
```



- Guardar programa de ejemplo como movel_with_relative_waypoint.urp

Punto_de_paso de variable

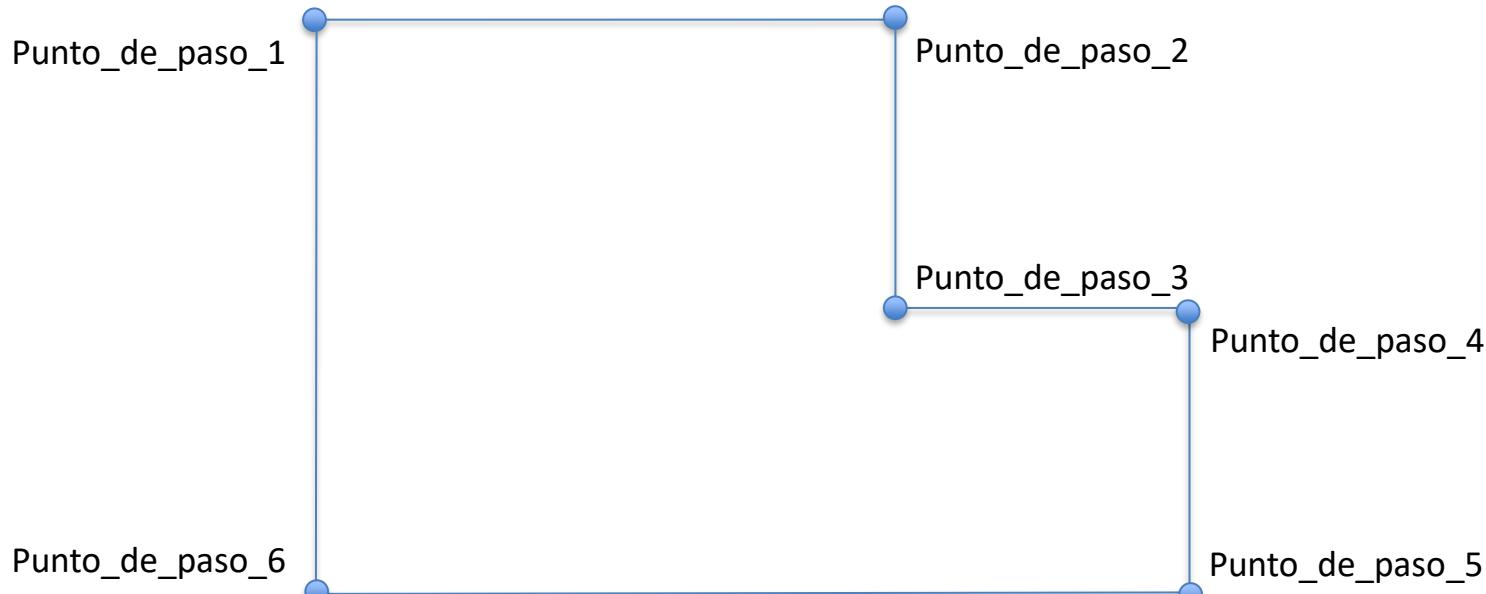
- Posición variable
 - Posición que se puede calcular / desplazar



- La definición de y cómo usar puntos de paso de variable son parte de la Formación Avanzada

Ejercicio práctico parte 1

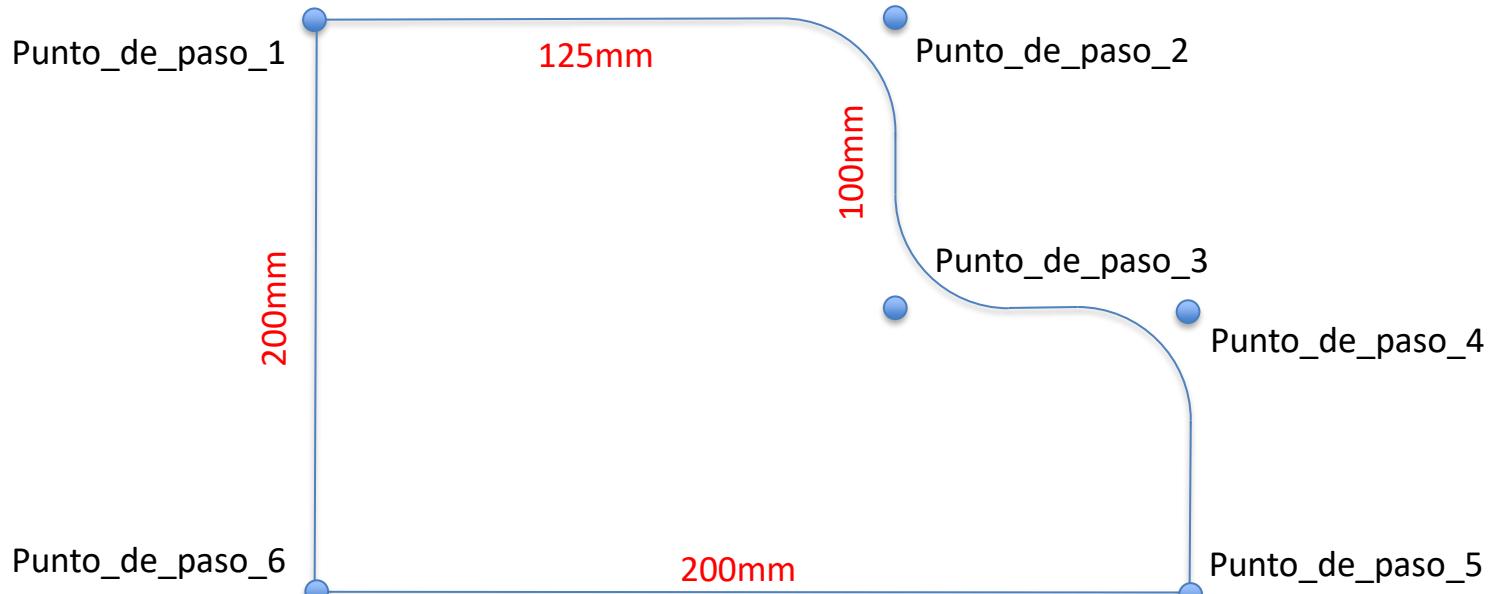
- Insertar instrucciones MoveL
 - Crear 6 puntos de paso en posiciones como en el diagrama
 - Usar la pestaña Mover para llevar la herramienta del robot a las posiciones adecuadas



- Guardar programa de ejemplo como Lab_exe_3_1.urp

Ejercicio práctico parte 2

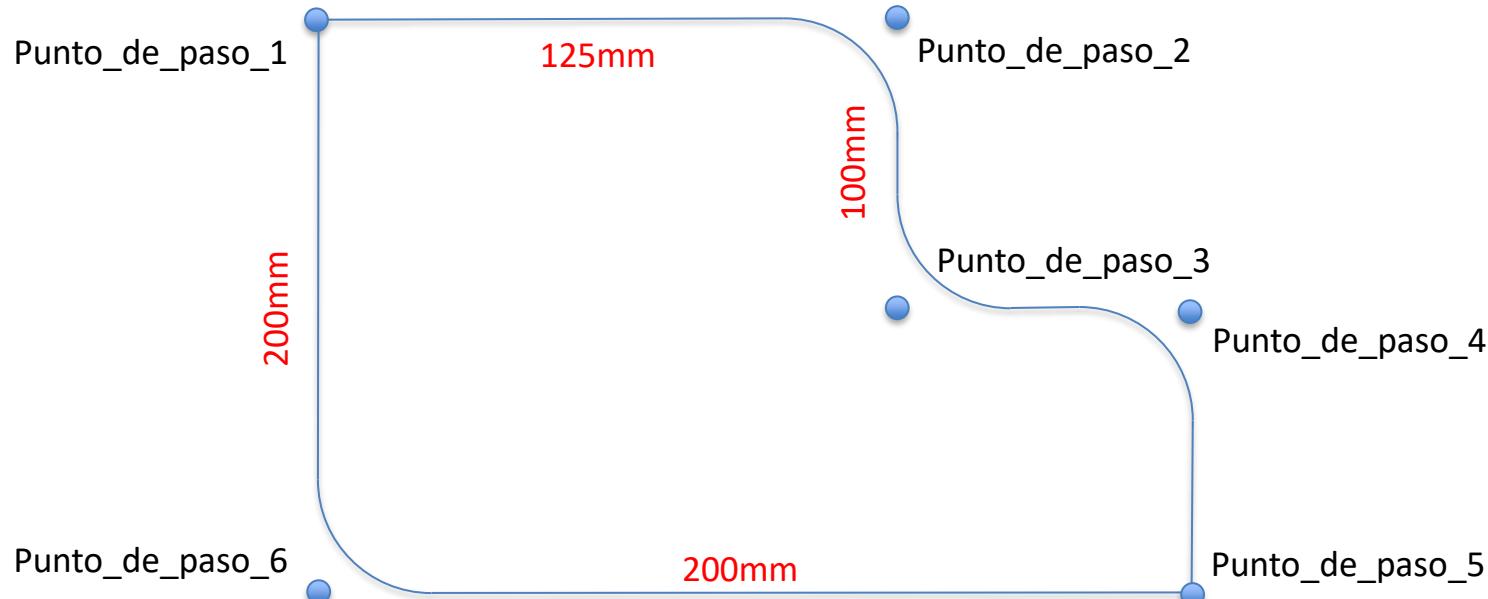
- Ajustar los puntos de paso
 - Ajustar los puntos de paso de acuerdo a las dimensiones del diagrama usando Editar pose
 - Añadir una transición con radio de 50mm a los puntos de paso 2, 3 and 4



- 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 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
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

Tratamiento de señales

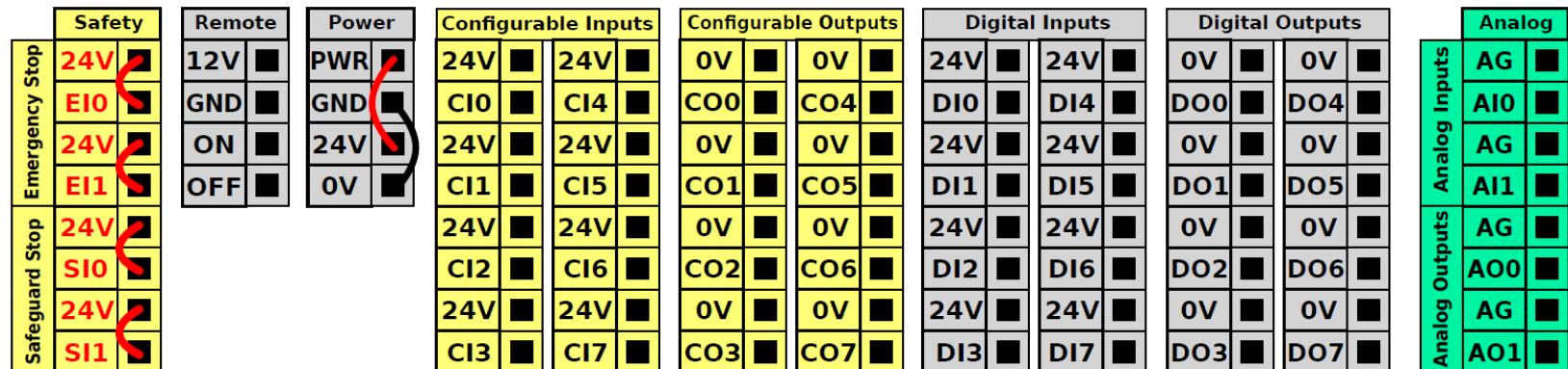
- Tratamiento de señales
 - Interacción con dispositivos externos



Interfaz eléctrica

- Placa de Control de Seguridad (SCB)

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

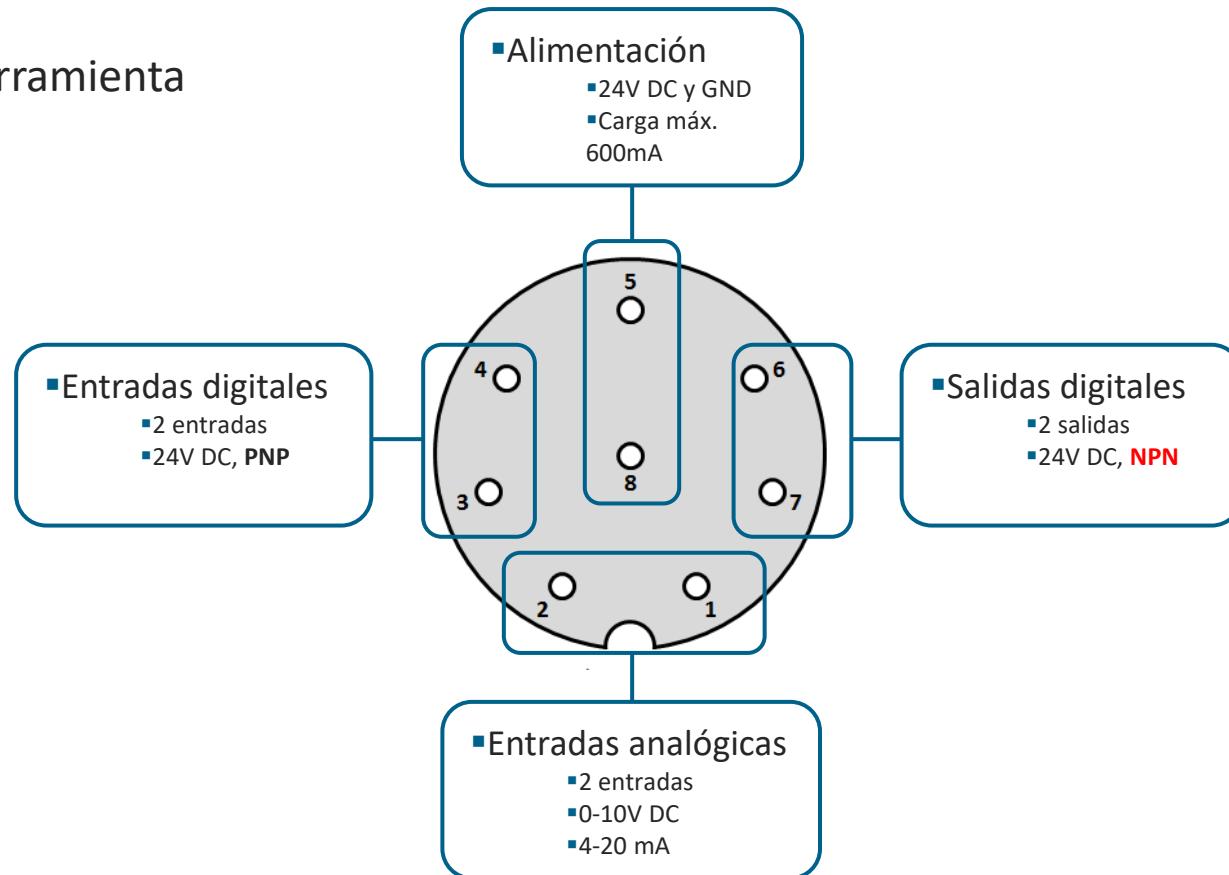


• Consultar en Manual de usuario, capítulo 4. Interfaz eléctrica

Interfaz eléctrica

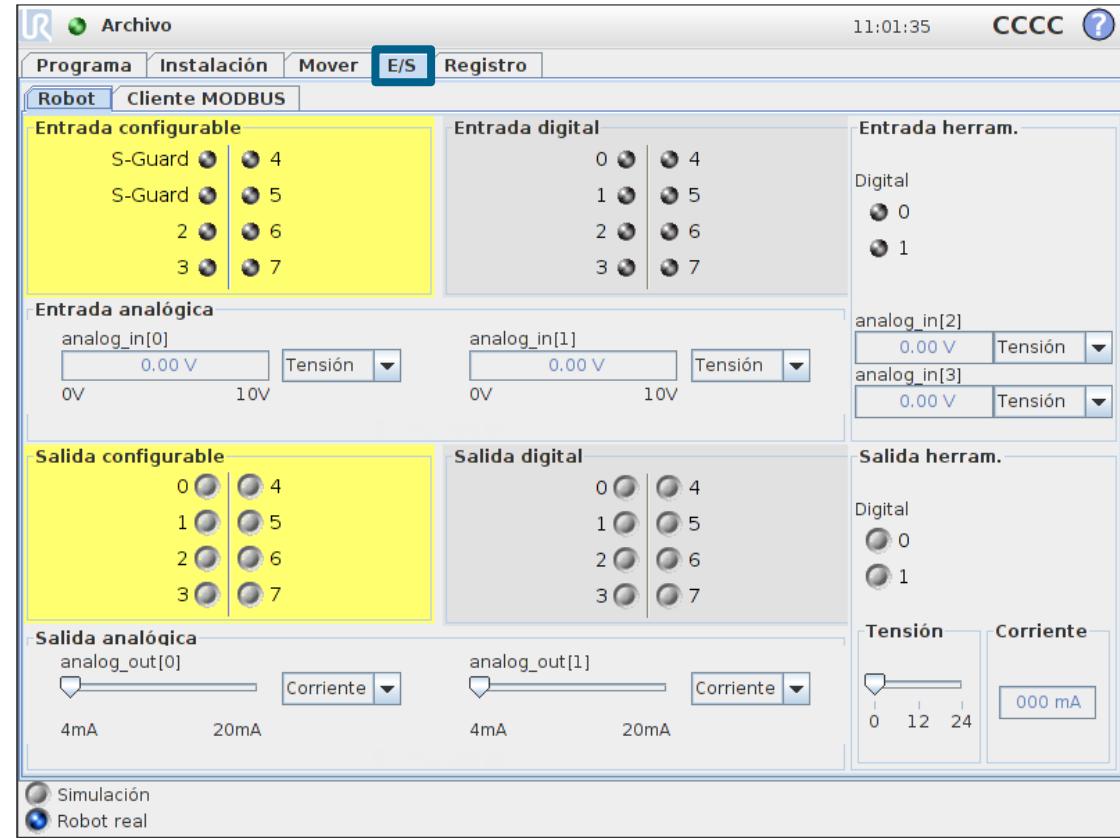
- Conector de herramienta

- 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



Tratamiento 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



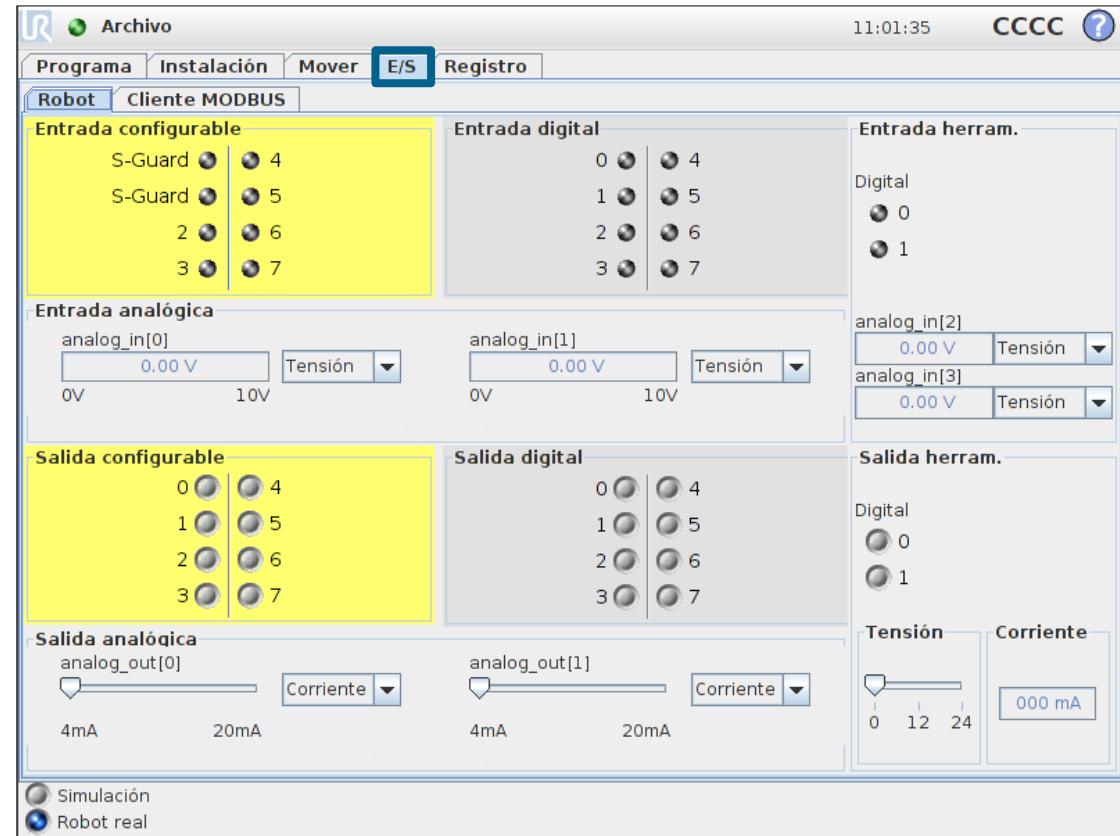
Tratamiento de señales

• Señales digitales

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

• Señales analógicas

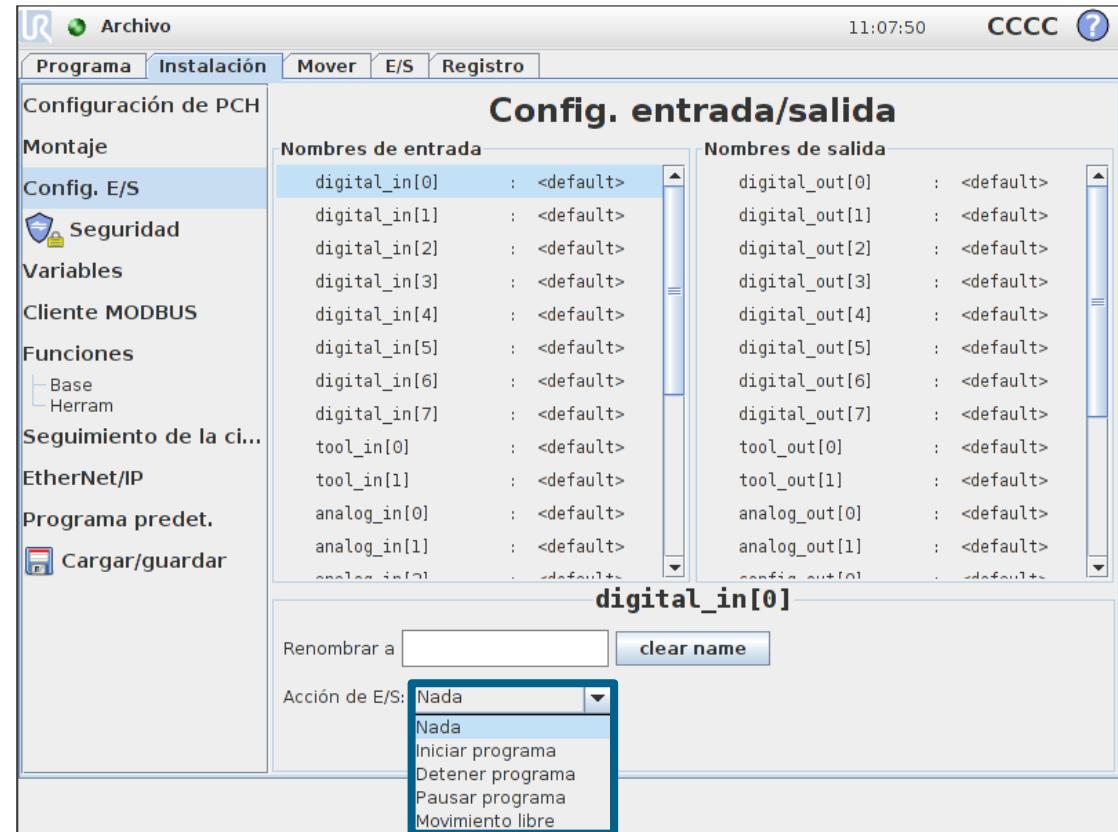
- Rango
 - Corriente 4-20 mA
 - Voltaje 0-10 V



Configuración de E/S

- Entradas

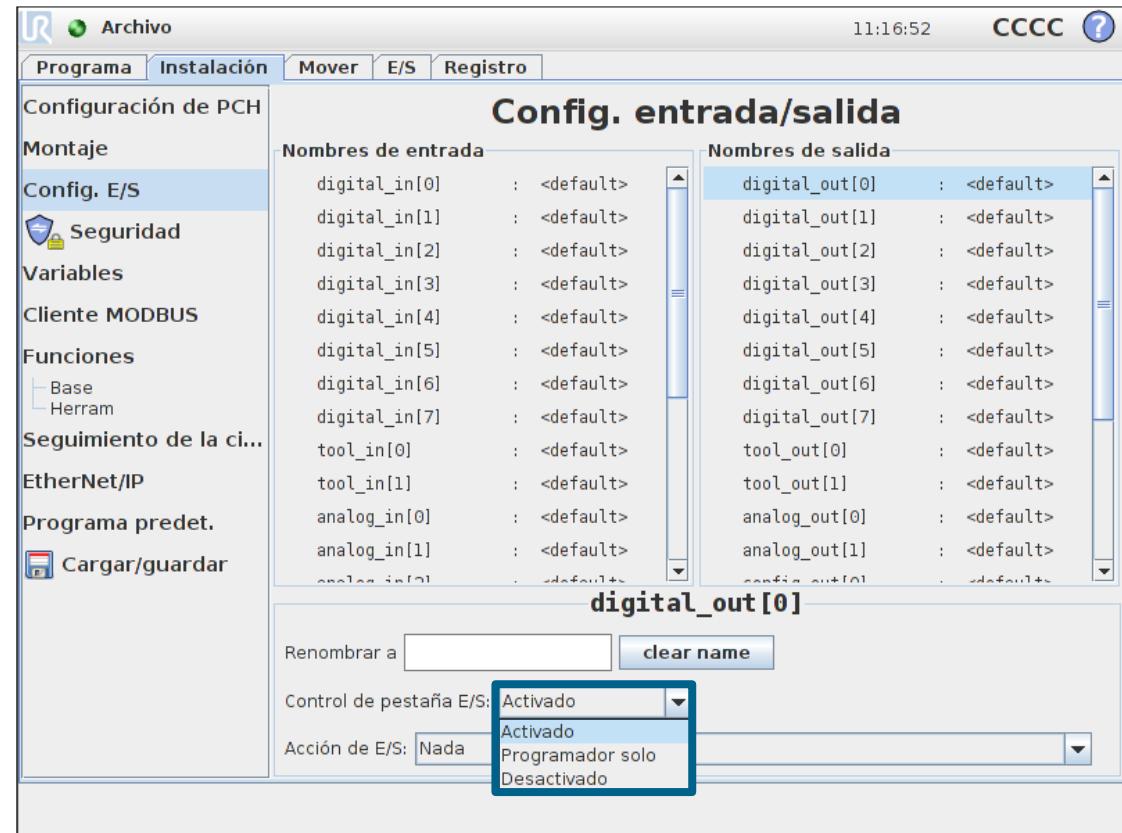
- Renombrar señales
- Asignar acción
 - Iniciar programa
 - Detener programa
 - Pausar programa
 - Movimiento libre



- Tanto el programa como la pestaña de E/S se actualizan al modificar el nombre de la señales

Configuración de E/S

- Salidas
 - Renombrar señales
 - Opciones para cambiar el estado de la salida desde la pestaña de E/S

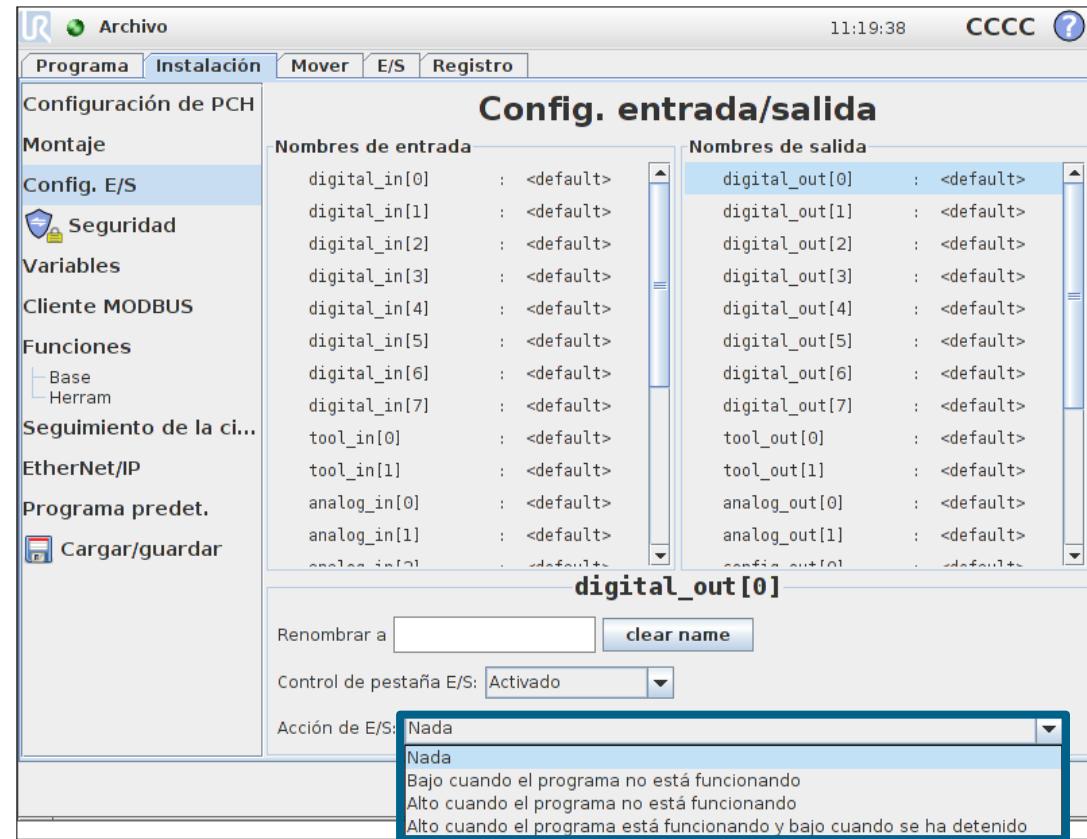


- Tanto el programa como la pestaña de E/S se actualizan al modificar el nombre de la señales

Configuración de E/S

- Salidas

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



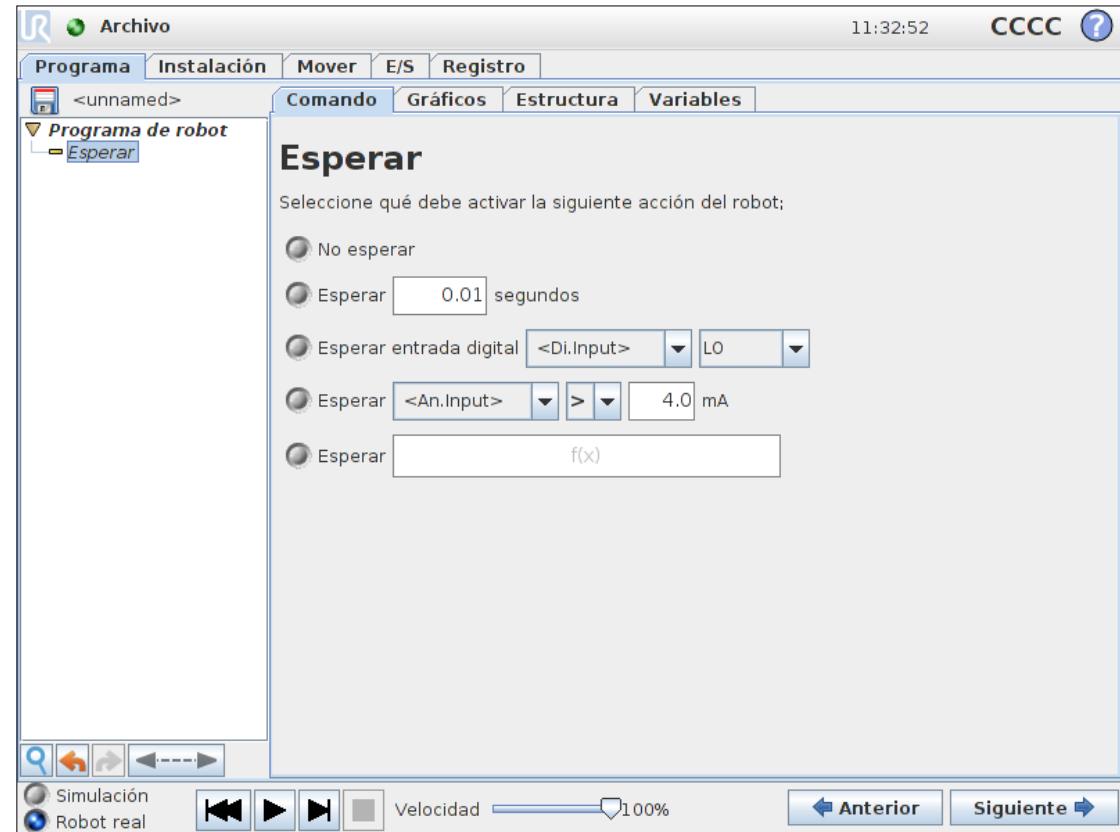
- Tanto el programa como la pestaña de E/S se actualizan al modificar el nombre de la señales

Comando Esperar

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

Programa de robot

```
MoveL  
    Punto_de_paso_1  
    Punto_de_paso_2  
    Esperar 1.0  
    Punto_de_paso_3  
    Punto_de_paso_4  
    Esperar DI[0] = HI
```



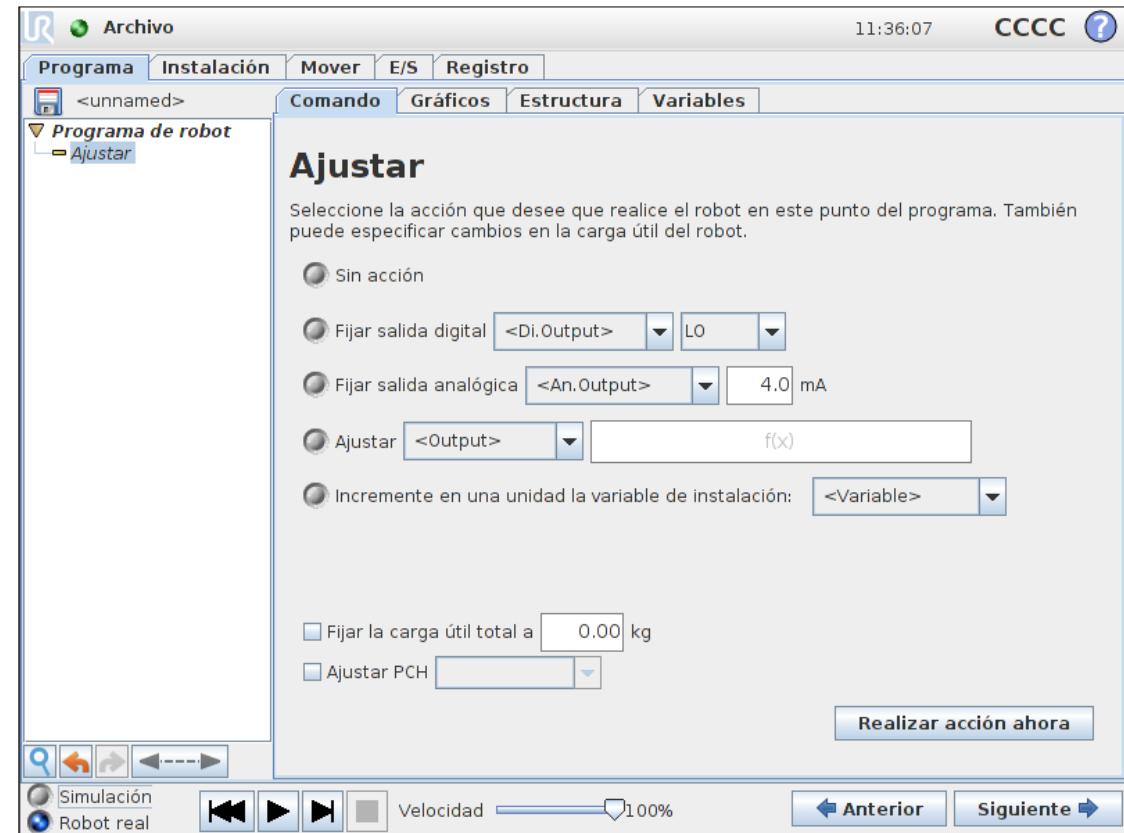
- Guardar programa de ejemplo como wait.urp

Comando Ajustar

- Realizar una acción
 - Sin acción
 - Fijar salida digital
 - Fijar salida analógica
 - Ajustar <expresión>
 - Incrementar variable
 - Set total payload
 - Fijar la carga útil

Programa de robot

```
MoveL  
    Punto_de_paso_1  
    Ajustar DO[0] = Encender  
    Punto_de_paso_2  
    Punto_de_paso_3  
    Punto_de_paso_4  
    Ajustar DO[0] = Apagar
```



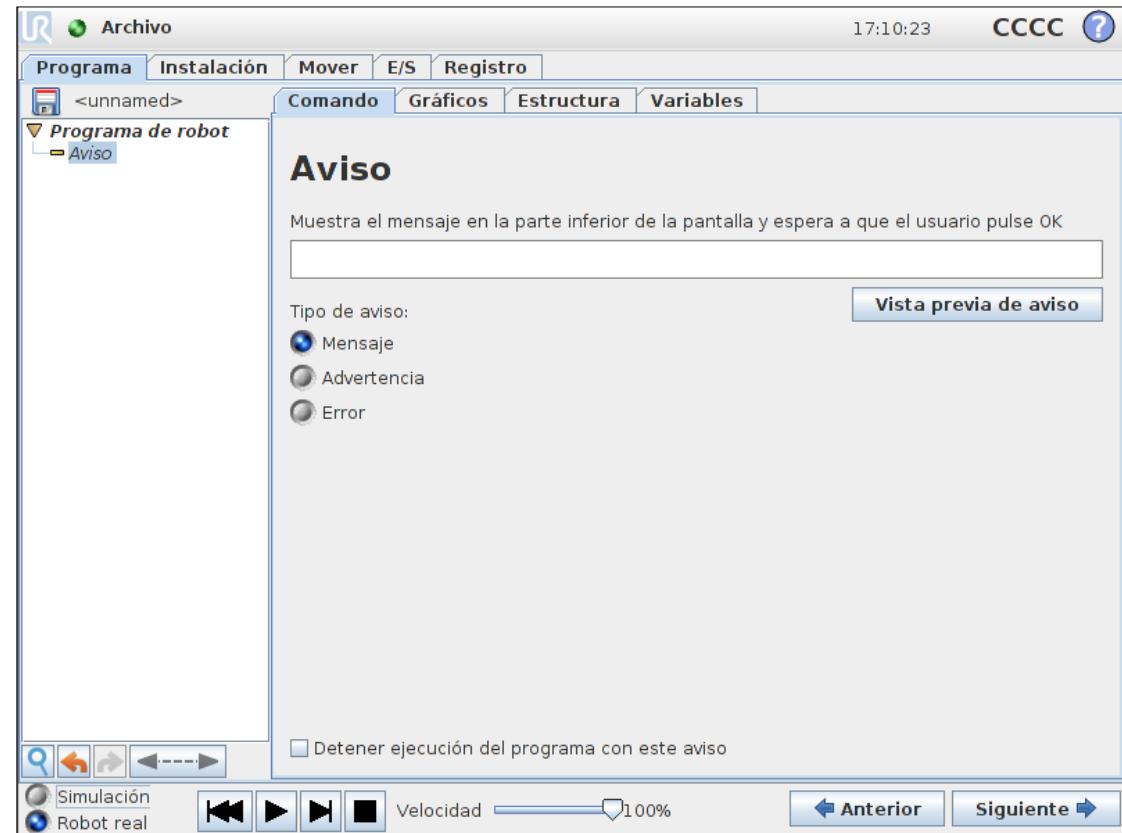
- Guardar programa de ejemplo como set.urp

Comando Aviso

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

Programa de robot

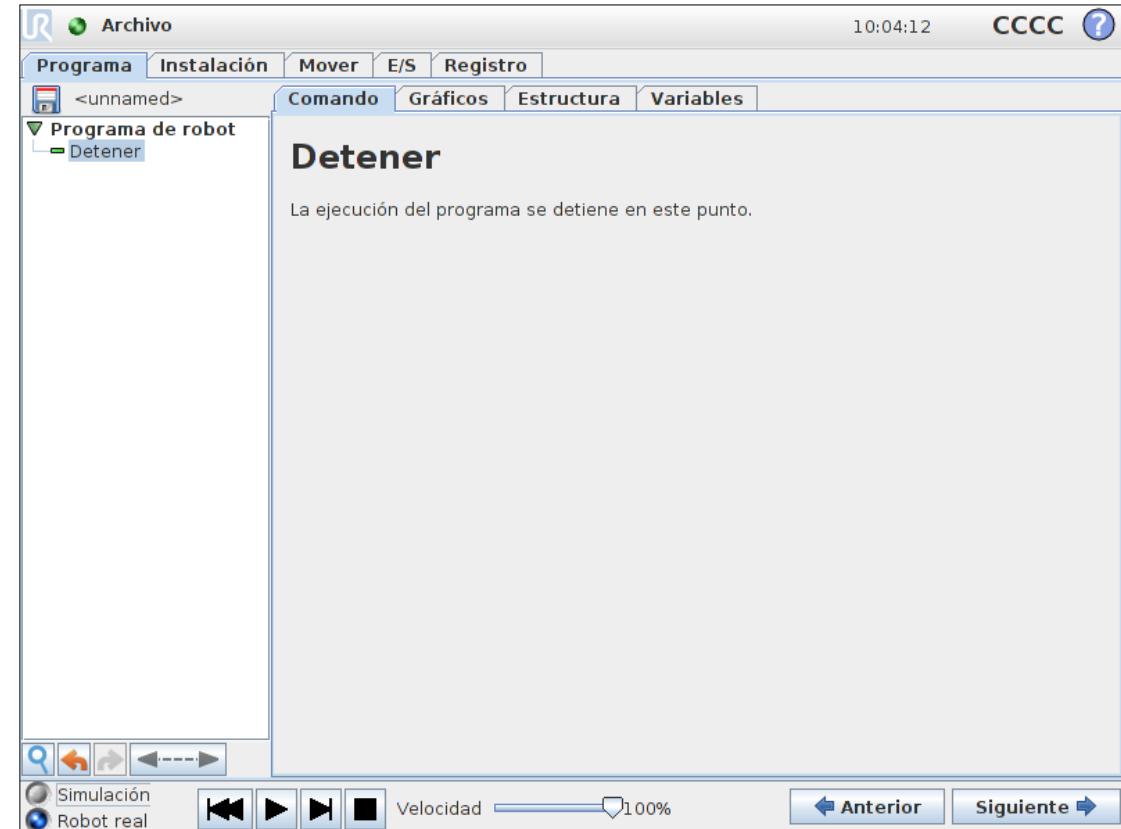
```
MoveL  
    Punto_de_paso_1  
    Punto_de_paso_2  
Espera 1.0  
    Punto_de_paso_3  
    Punto_de_paso_4  
Aviso
```



- Guardar programa de ejemplo como Aviso.urp

Comando Detener

- **Detener**
 - Detiene la ejecución del programa



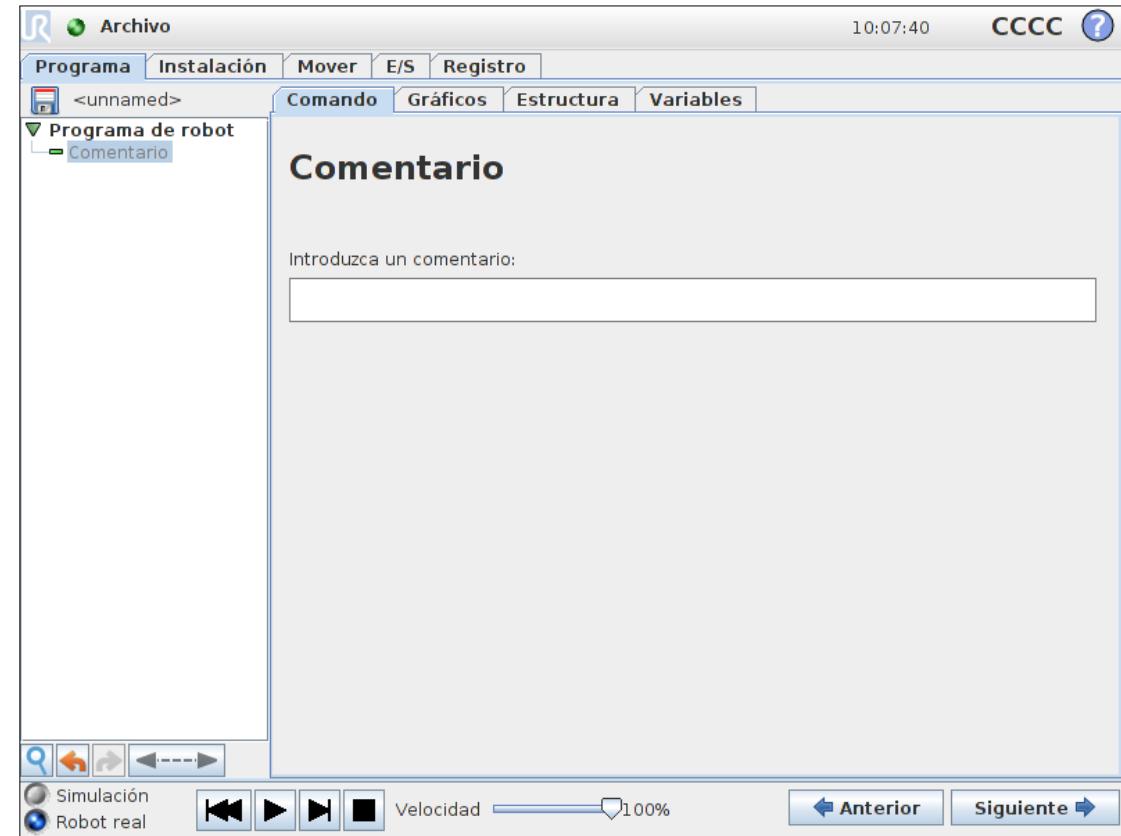
Programa de robot

```
MoveL  
Punto_de_paso_1  
Punto_de_paso_2  
Detener
```

- Guardar programa de ejemplo como halt.urp

Comando Comentario

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

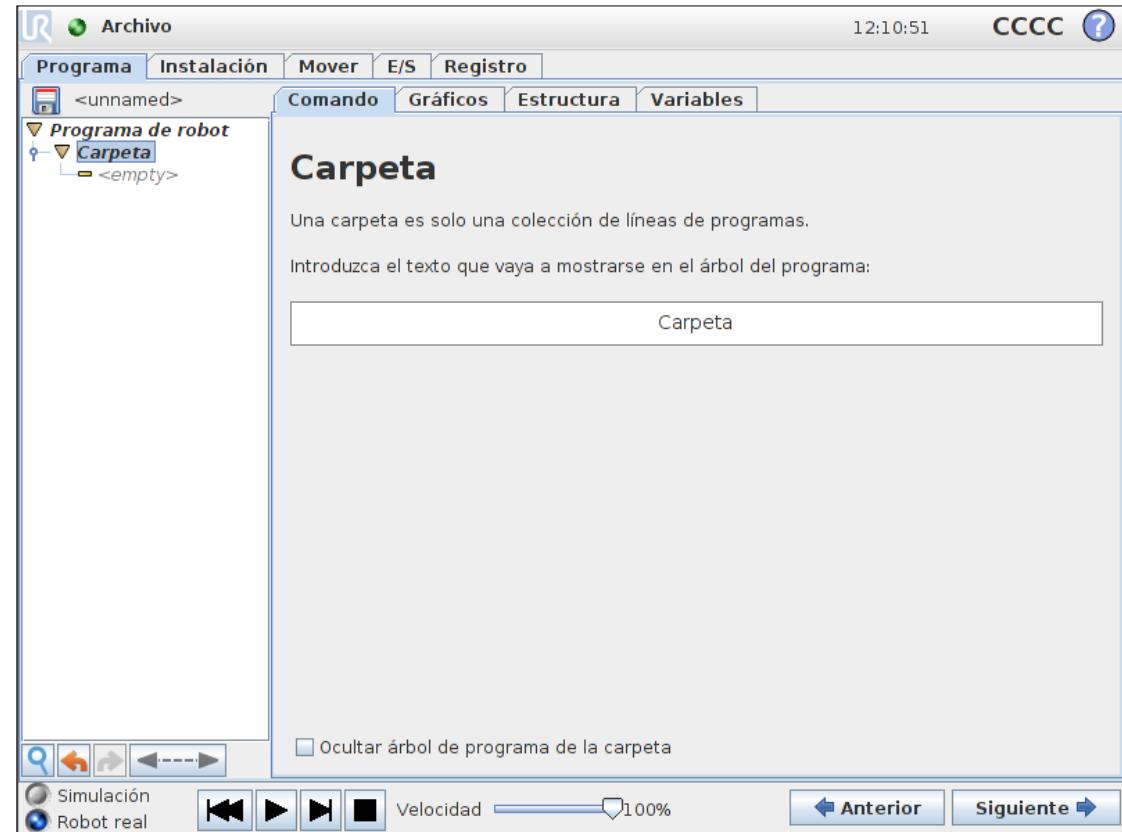


```
Programa de robot
MoveL
    Punto_de_paso_1
    Punto_de_paso_2
'Ingresar comentario
```

Comando Carpeta

- **Carpetas**

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



Comando Carpeta

- Programa de ejemplo coger y dejar
 - Dos carpetas
 - Coger_pieza
 - Dejar_pieza

Programa de robot

Coger_pieza

MoveL

Punto_de_paso_1

Punto_de_paso_2

Ajustar DO[0] = Encender

Esperar 0.5

Punto_de_paso_1

Dejar_pieza

MoveL

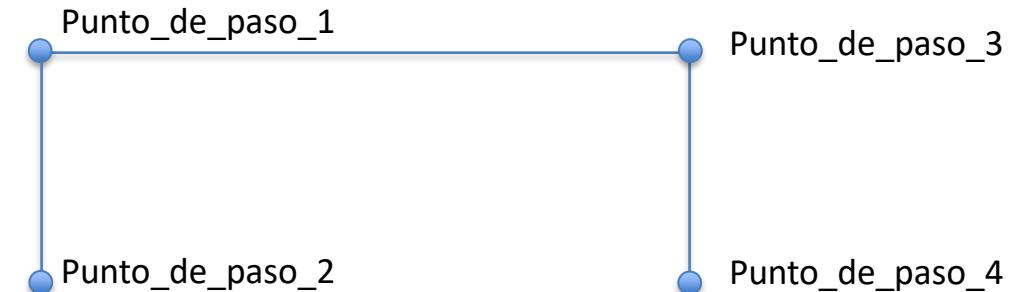
Punto_de_paso_3

Punto_de_paso_4

Ajustar DO[0] = Apagar

Esperar 0.5

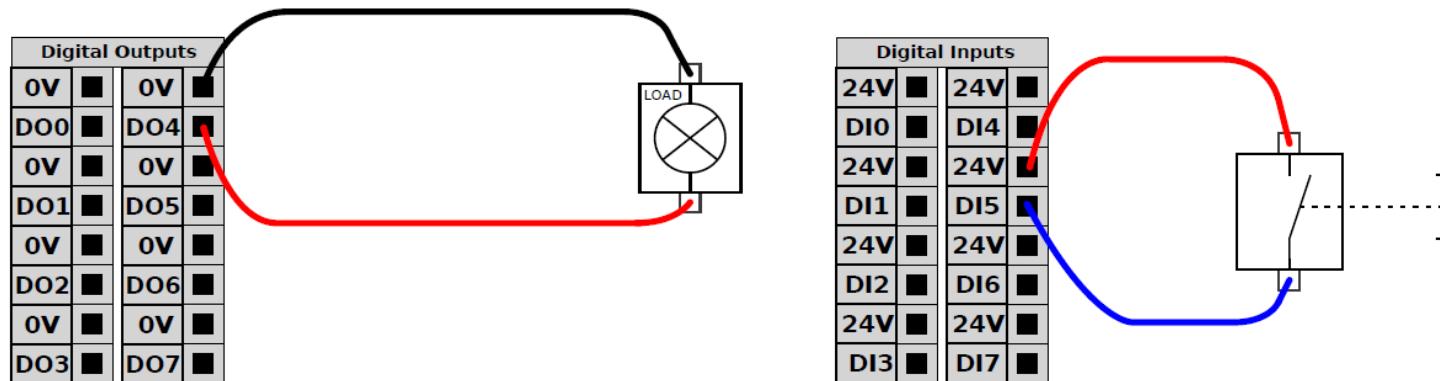
Punto_de_paso_3



- Guardar programa de ejemplo como pick_and_place.urp

Ejercicio práctico 1

- Crear un programa simple para el tratamiento de señales
- Configuración:
 - Renombrar digital_out[4] a "Lampara" y digital_in[5] a "Pulsador".
 - Conectar la lámpara a la Salida Digital 0 en el controlador como indicado en el esquema.
 - Conectar el pulsador a la Entrada Digital 0 en el controlador como indicado en el esquema.



- Guardar programa de ejemplo como Lab_exe_4_1.urp

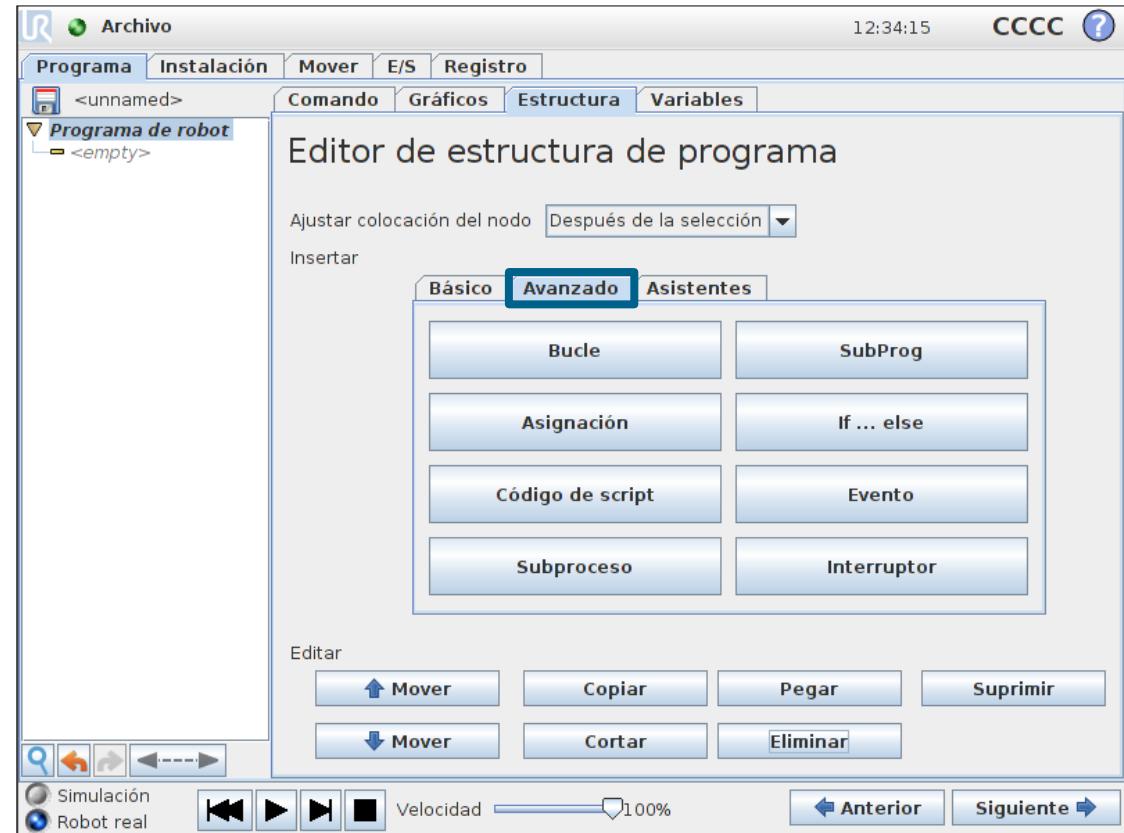
Ejercicio práctico 2

- Programa – Carpeta_Pulsador
 - Crear una carpeta y nombrarla como Carpeta_Pulsador.
 - Esperar la señal del pulsador para continuar el programa.
 - Añadir un comentario al inicio de la carpeta explicando lo que el código realiza.
- Programa – Carpeta_Lampara
 - Crear una carpeta y nombrarla como Carpeta_Lampara.
 - Conectar la lámpara durante 5 segundos para desconectarla posteriormente.
 - Crear un aviso para el operador que informe que la ejecución del programa se ha completado y detenga la ejecución del mismo.
 - Añadir un comentario al inicio de la carpeta explicando lo que el código realiza.
- Guardar programa de ejemplo como Lab_exe_4_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

Pestaña de comandos avanzados

- Comandos avanzados
 - Realizar operaciones avanzadas



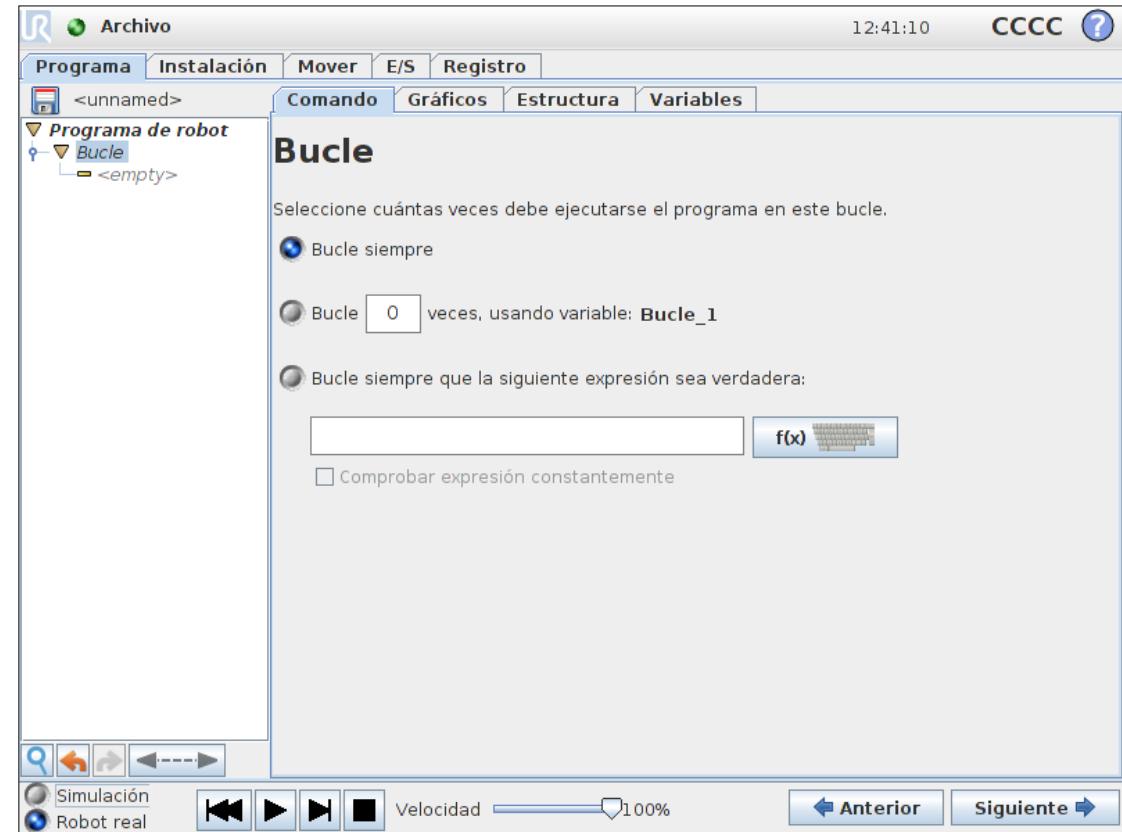
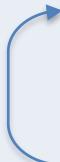
Comando Bucle

Definición

- Ejecuta cíclicamente las instrucciones que contiene
- Tipos de bucle
 - Bucle siempre
 - Bucle n veces
 - Bucle <expresión>

Programa de robot

```
MoveJ  
    Punto_de_paso_5  
Bucle 3 veces  
    MoveL  
        Punto_de_paso_1  
        Punto_de_paso_2  
        Punto_de_paso_3  
        Punto_de_paso_4  
MoveJ  
    Punto_de_paso_5
```



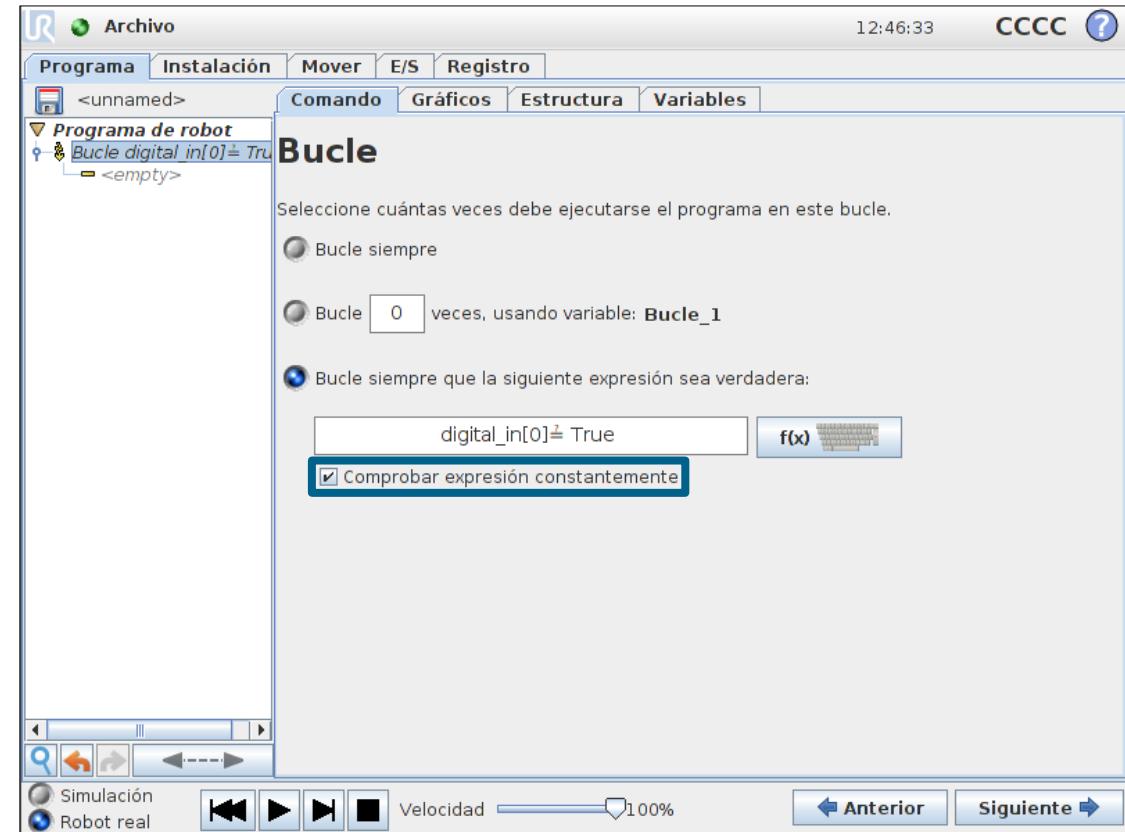
- Guardar programa de ejemplo como loop.urp

Comando Bucle

- Interrupción
 - Comprobar expresión constantemente

Programa de robot

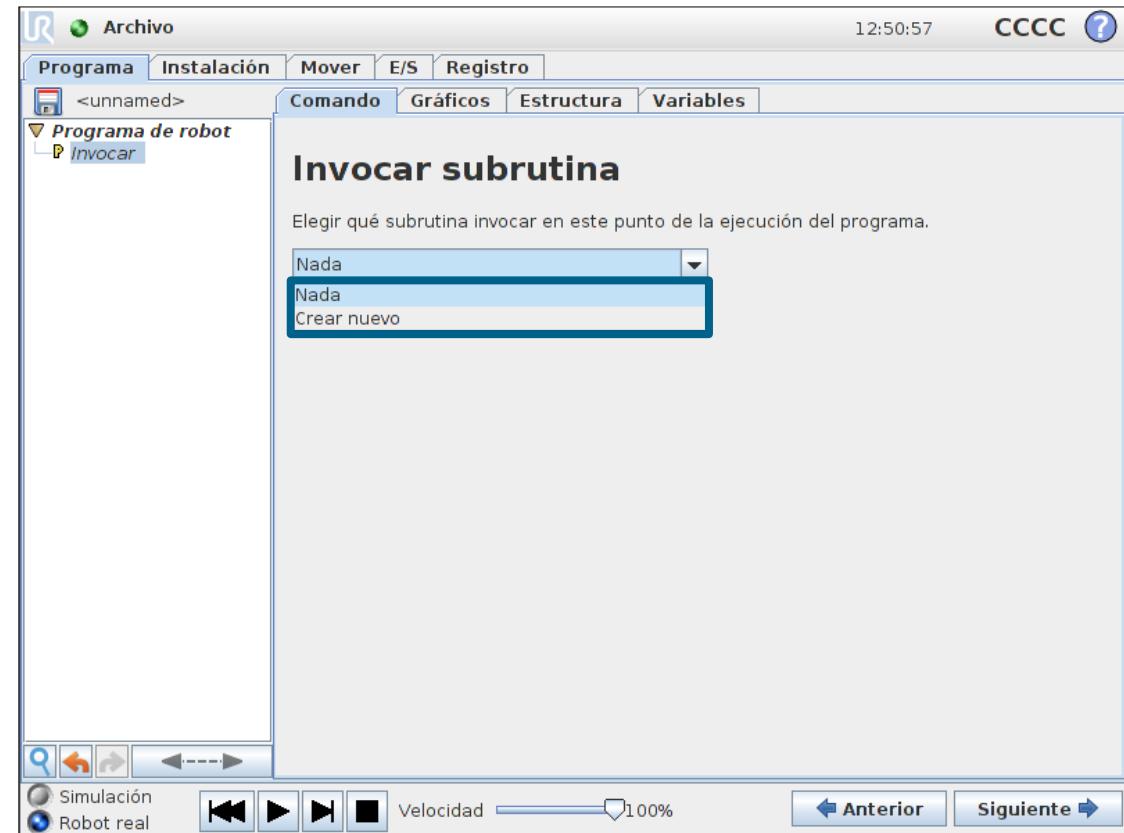
```
MoveJ  
    Punto_de_paso_5  
Bucle DI[0] = True  
    MoveL  
        Punto_de_paso_1  
        Punto_de_paso_2  
        Punto_de_paso_3  
        Punto_de_paso_4  
MoveJ  
    Punto_de_paso_5
```



- Guardar programa de ejemplo como `loop_interrupt.urp`

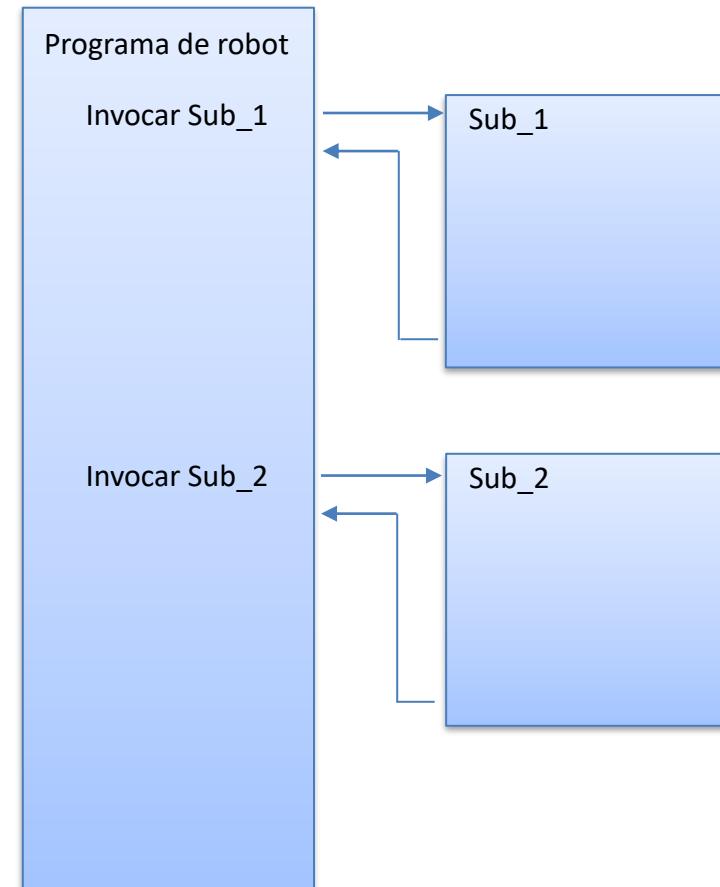
Comando SubProg

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



Comando *SubProg*

- Invocando una subrutina
 - Se ejecutan todos los comandos de la subrutina
 - Después regresa al programa "principal"

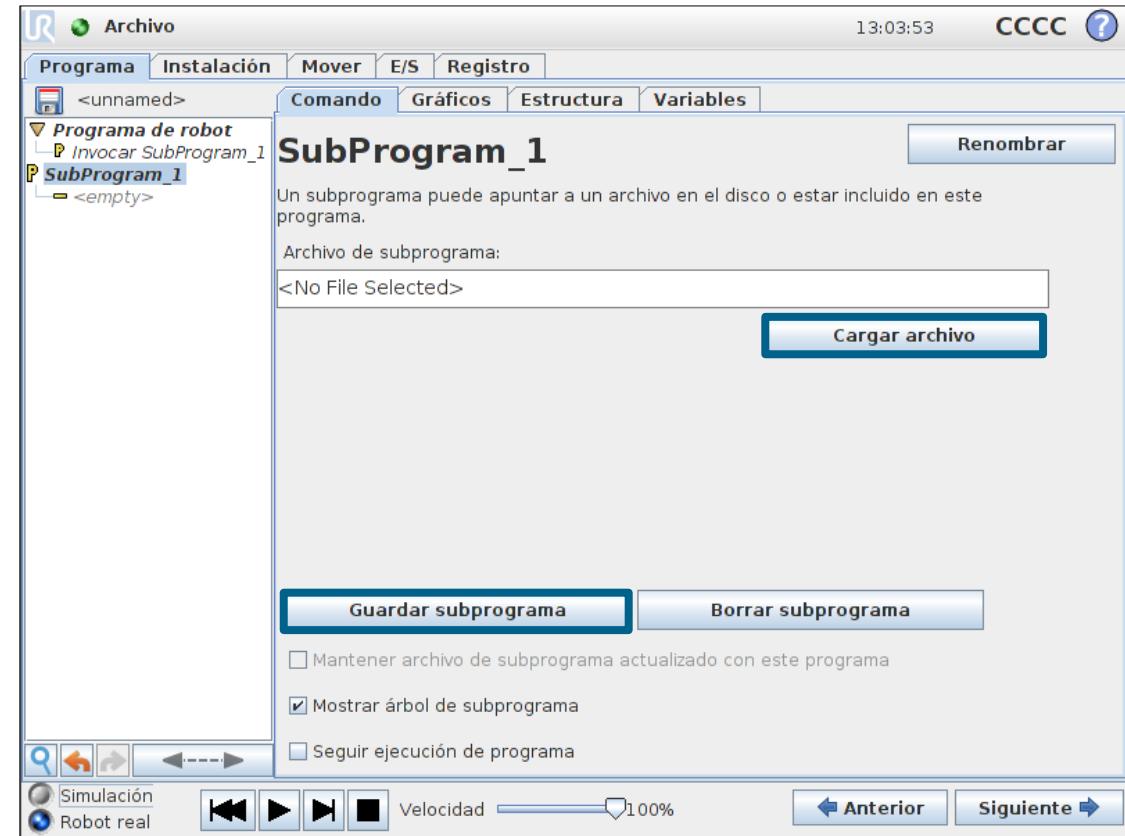


- Nota: No se soporta invocar una subrutina desde una subrutina

Comando SubProg

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

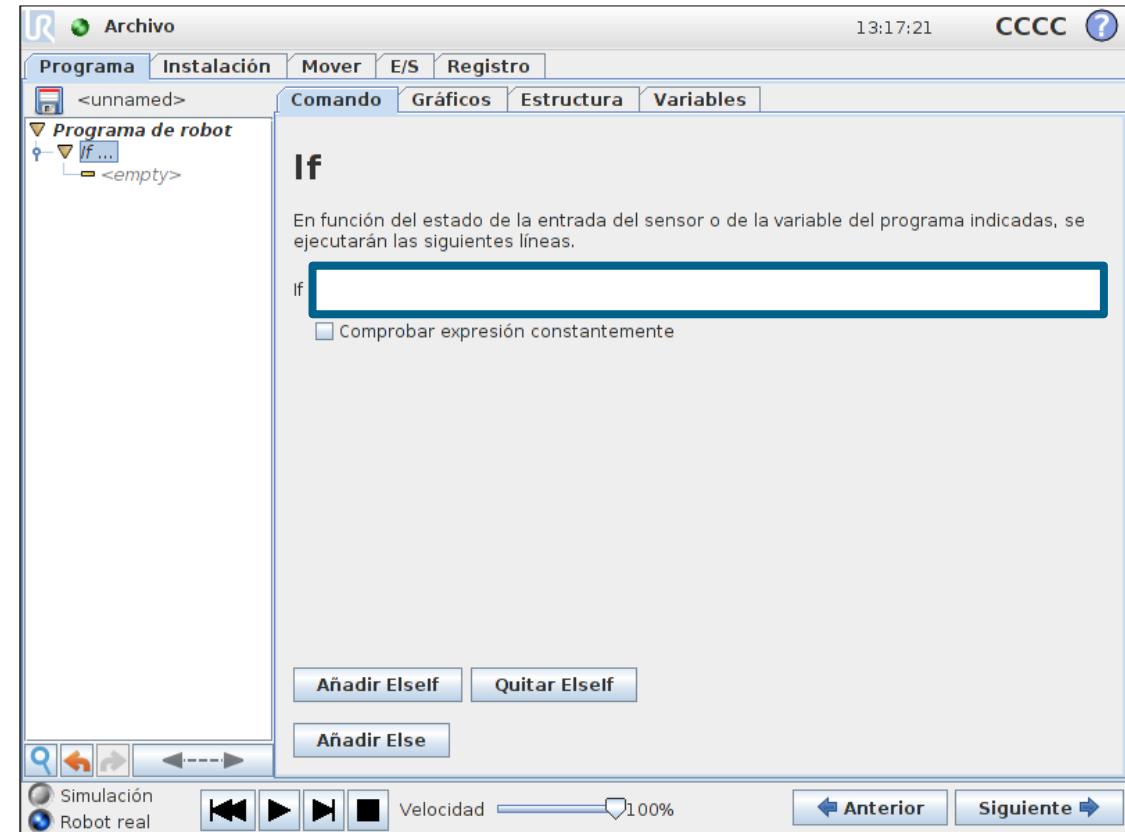
```
Programa de robot
MoveJ
    Punto_de_paso_5
Bucle 3 veces
MoveL
    Punto_de_paso_1
    Punto_de_paso_2
    Punto_de_paso_3
    Punto_de_paso_4
Invocar SubP_movec
MoveJ
    Punto_de_paso_5
```



- Guardar programa de ejemplo como call_sub.urp

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

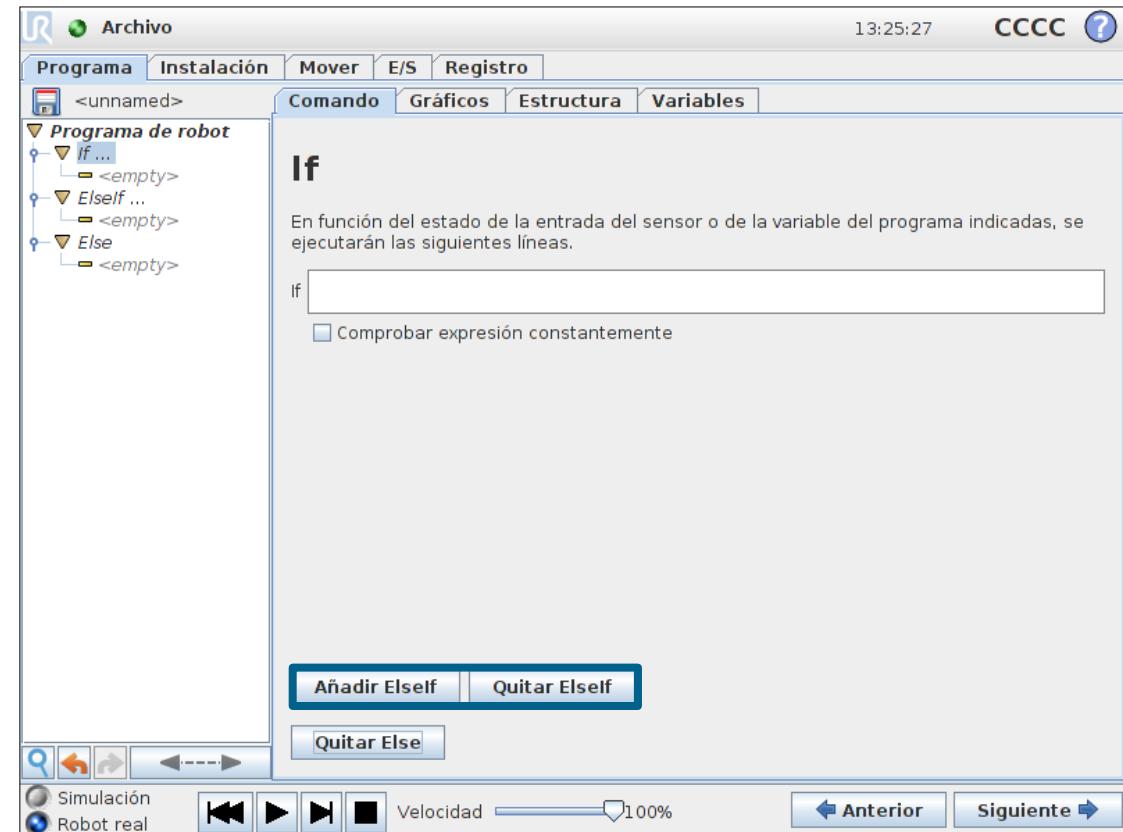


```
Programa de robot
MoveL
    Punto_de_paso_1
    IF DI[0] = True
        Punto_de_paso_2
    Punto_de_paso_3
```

- Guardar programa de ejemplo como if.urp

Comando *If ... else*

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



```
Programa de robot
MoveL
    Punto_de_paso_1
    IF DI[0] = True
        Punto_de_paso_2
    Else
        Punto_de_paso_3
```

- Guardar programa de ejemplo como if_else.urp

Ejercicio práctico 1

- Crear un programa usando subprogramas, bucles e instrucciones if
- Usar la instalación del ejercicio 2 de comandos básicos
- Primero crear un programa simple para conectar una lámpara de forma intermitente (este programa será usado como subprograma)
- Lampara_intermitente:
 - Crear un programa que conecte una lámpara de forma intermitente 5 veces usando un bucle
 - Conectar la lámpara durante 0.5 segundos, luego desconectarla durante 0.5 segundos (5 repeticiones)
 - Guardar el programa con el nombre Lampara_intermitente
- Guardar programa de ejemplo como Lab_exe_5_1.urp

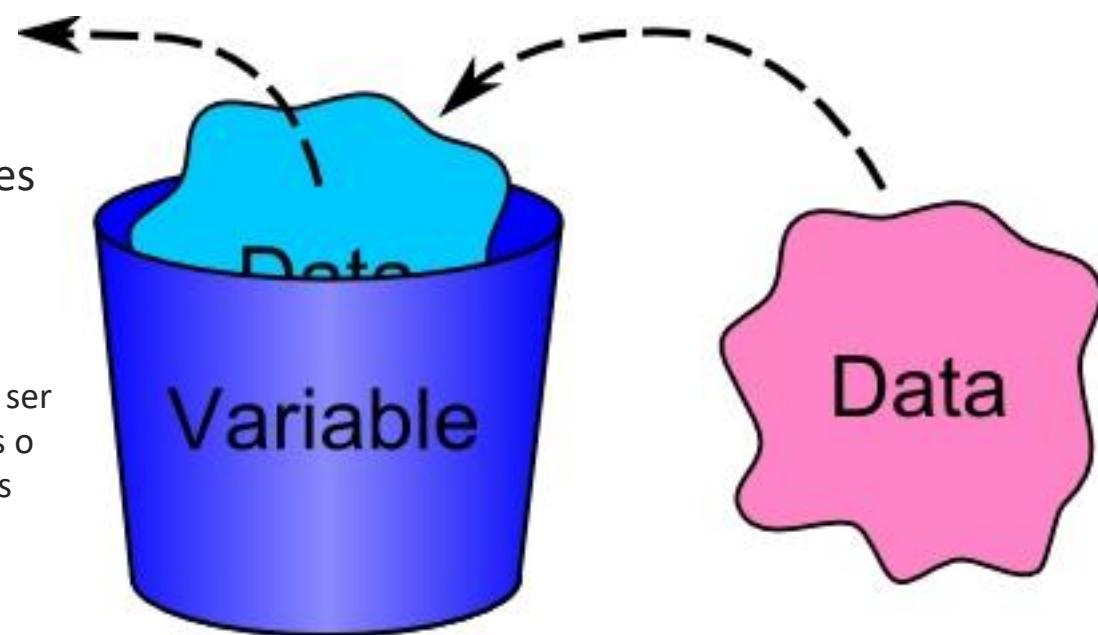
Ejercicio práctico 2

- Crear otro nuevo programa
 - Crear 3 puntos de paso en posiciones cualesquiera
 - Mover al Punto_de_paso_1
 - Esperar la pulsación del botón
 - Tras recibir la señal del botón llamar al subprograma Lampara_intermitente
 - Mover al Punto_de_paso_2
 - Esperar una posible pulsación del botón durante 1 segundo
 - Si el botón ha sido pulsado, mover al Punto_de_paso_1
 - En cualquier otro caso, mover al Punto_de_paso_3
 - Mover entre los puntos 1 y 2 repetidamente, detener inmediatamente si se pulsa el botón
- 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

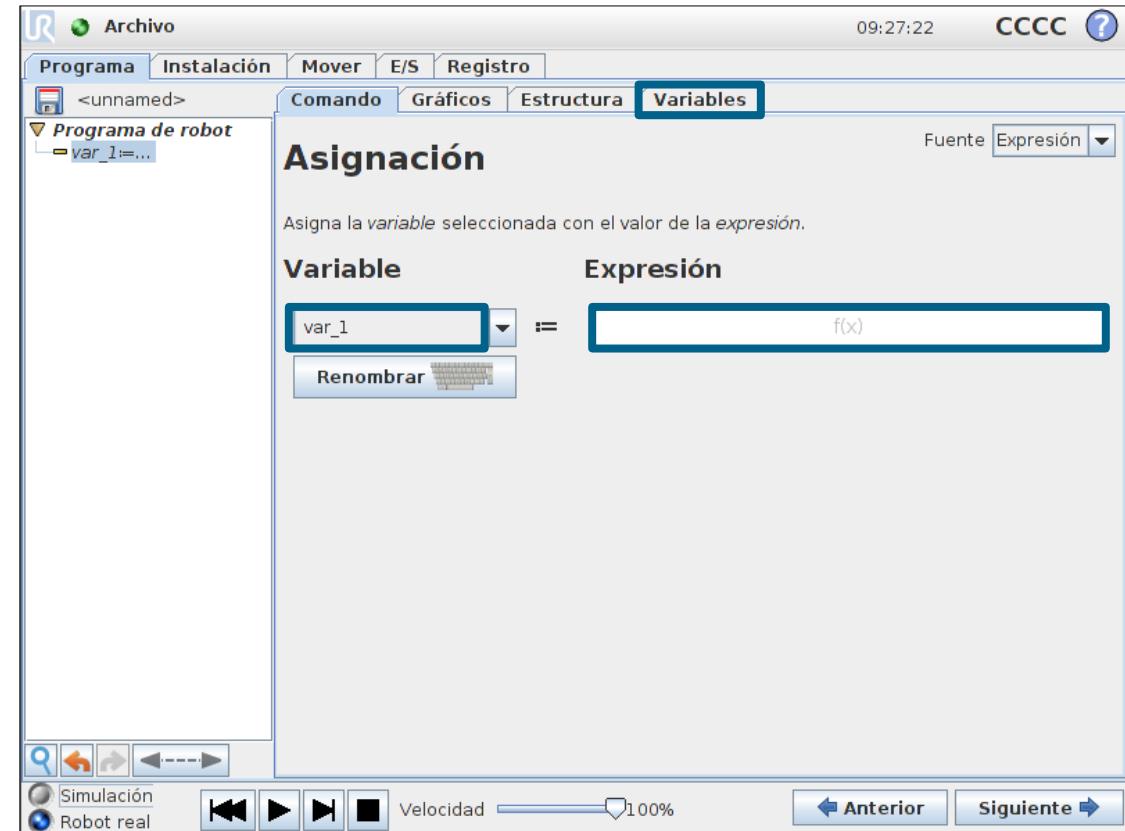
Comando Asignación

Opciones

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

Programa de robot

```
var_1 = True  
Espera 0.5  
var_1 = False  
Espera 0.5
```

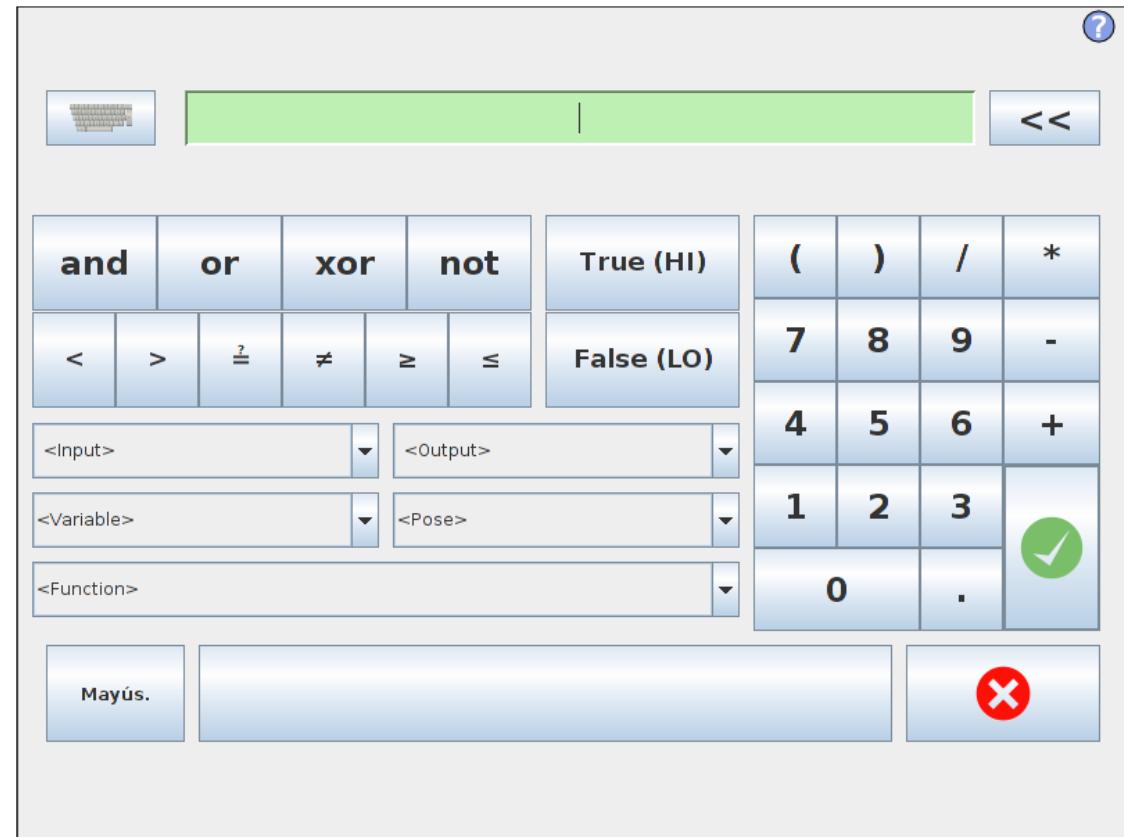


- Guardar programa de ejemplo como var_bool.urp

Editor de expresiones

- Opciones

- Valores numéricos
- Entradas
- Salidas
- Variables
- Posiciones
- Códigos Script
- Operadores lógicos
- Teclado

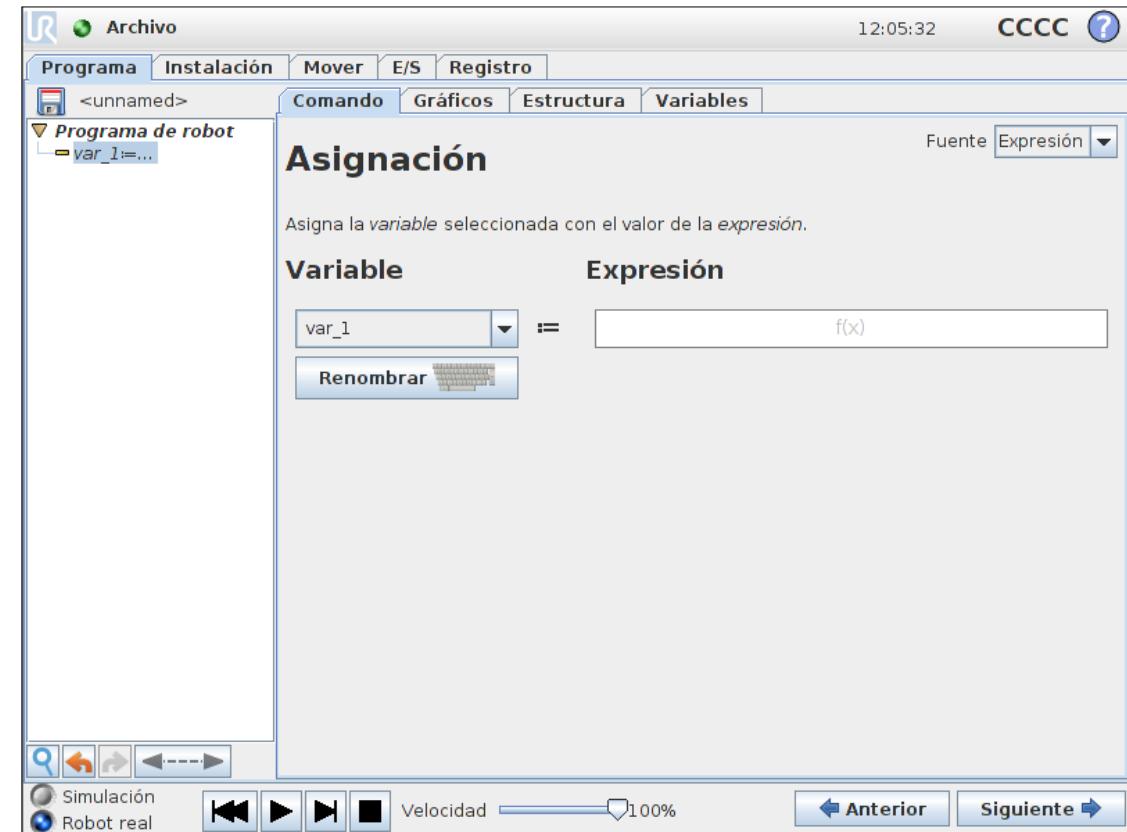


Usar una variable como contador

- Contador
 - Variable tipo entero
 - Incrementar variable en bucle
 - Comparar la variable con un número

Programa de robot

```
var_1 = 0  
Bucle var_1 < 5  
    Coger_pieza  
    Dejar_pieza  
    var_1 = var_1 + 1
```



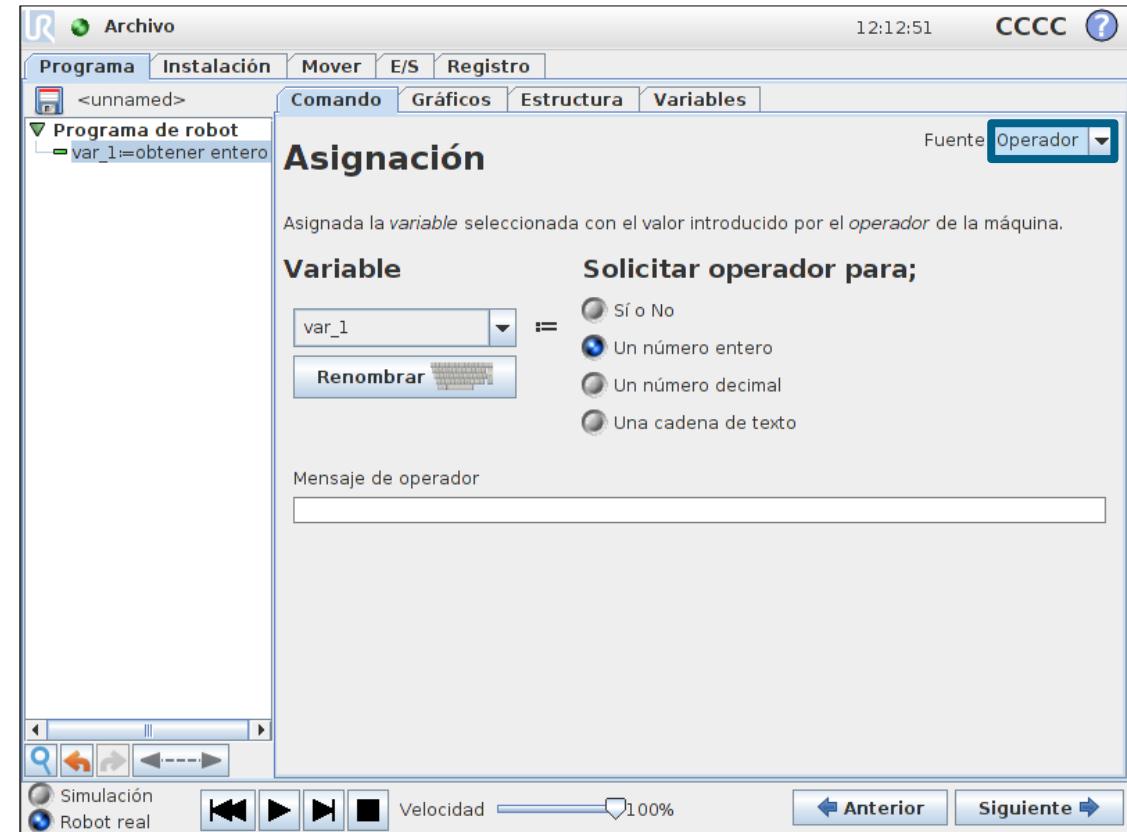
- Guardar programa de ejemplo como var_counter.urp

Asignación introducida por operador

- Asignación de operador
 - Seleccionar como fuente
 - Definir tipo de variable
 - Añadir mensaje

Programa de robot

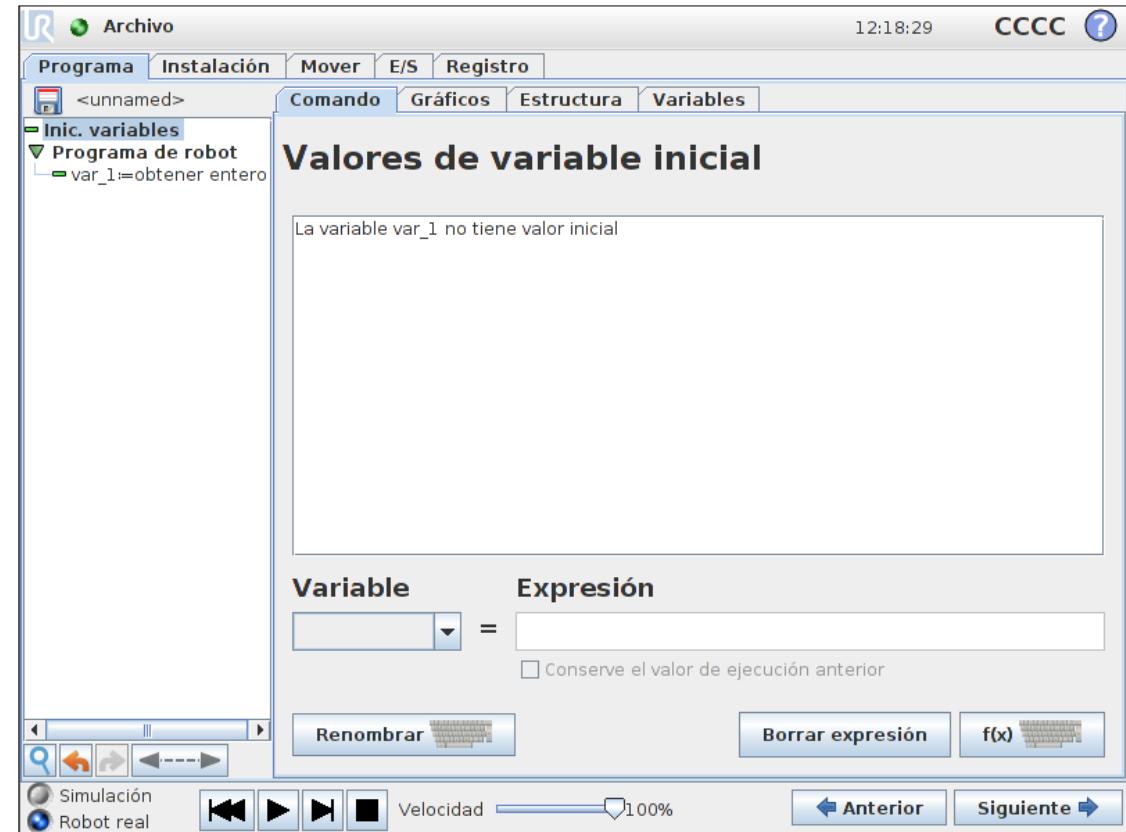
```
var_1 = 0  
var_2 = entrada de operador  
Bucle var_1 < var_2  
    Coger_pieza  
    Dejar_pieza  
    var_1 = var_1 + 1
```



- Guardar programa de ejemplo como var_operator_input.urp

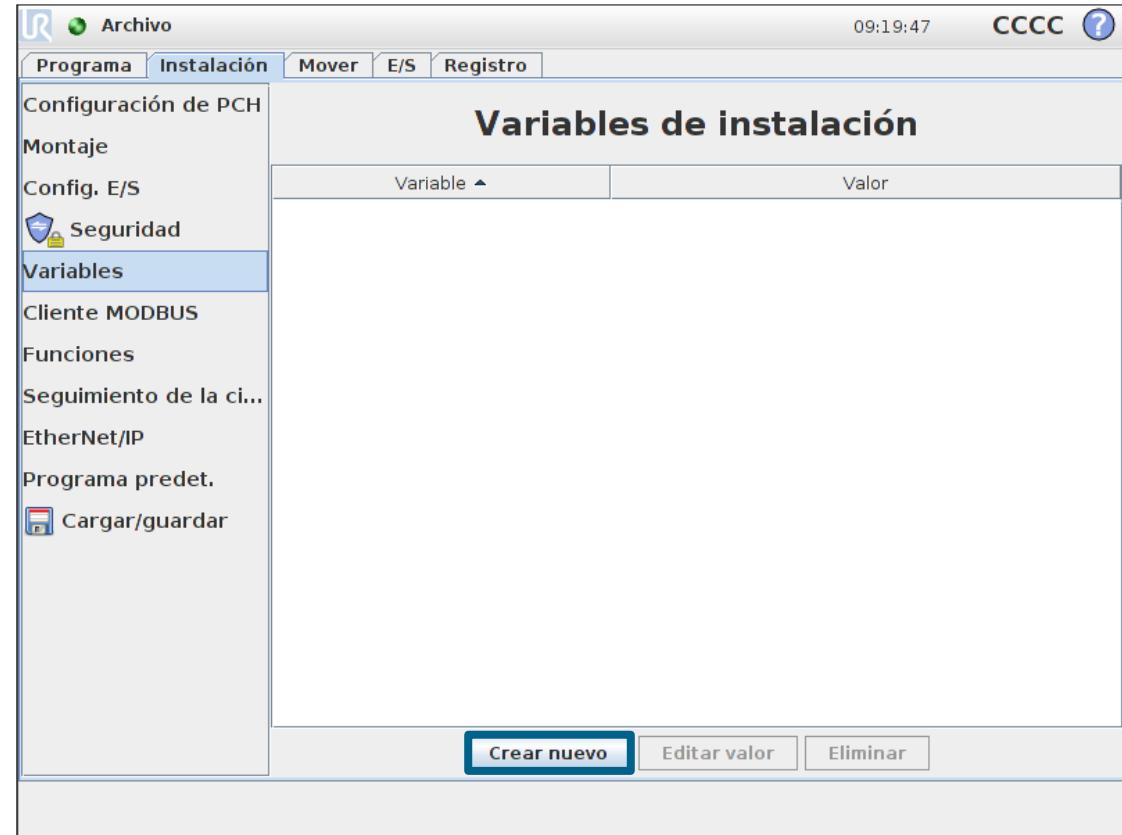
Inicialización de variables

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



Variables de instalación

- Características
 - Lista en ficha de instalación
 - Guardadas en fichero separado
 - Mantienen el valor tras reinicio
- Funcionalidad
 - Variables globales
 - Accesibles desde todos los programas
 - Mismas funcionalidades que las variables locales



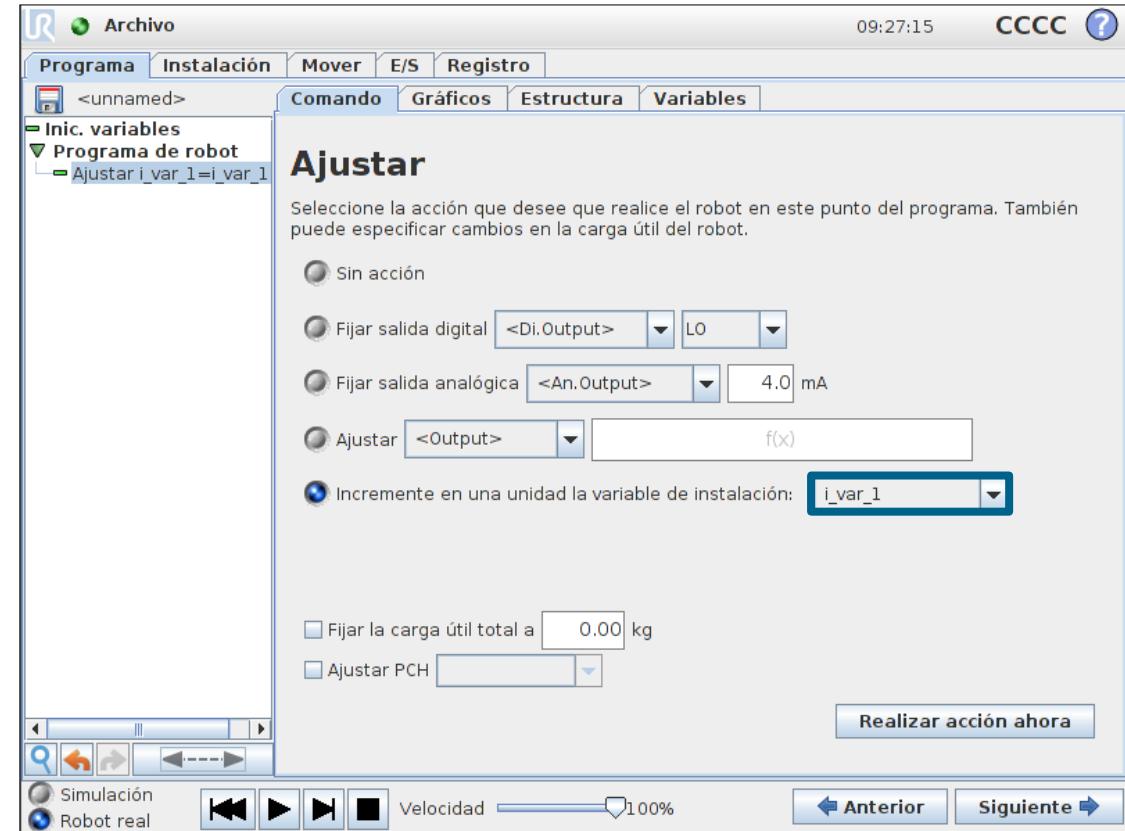
Use una variable de instalación como contador

- Comando Ajustar

- Incrementar una unidad la variable de instalación

Programa de robot

```
i_var_1 = 0  
var_2 = Entrada de operador  
Bucle i_var_1 < var_2  
    Coger_pieza  
    Dejar_pieza  
    Ajustar i_var_1 = i_var_1 + 1
```



- Guardar programa de ejemplo como inst_var_operator_input.urp

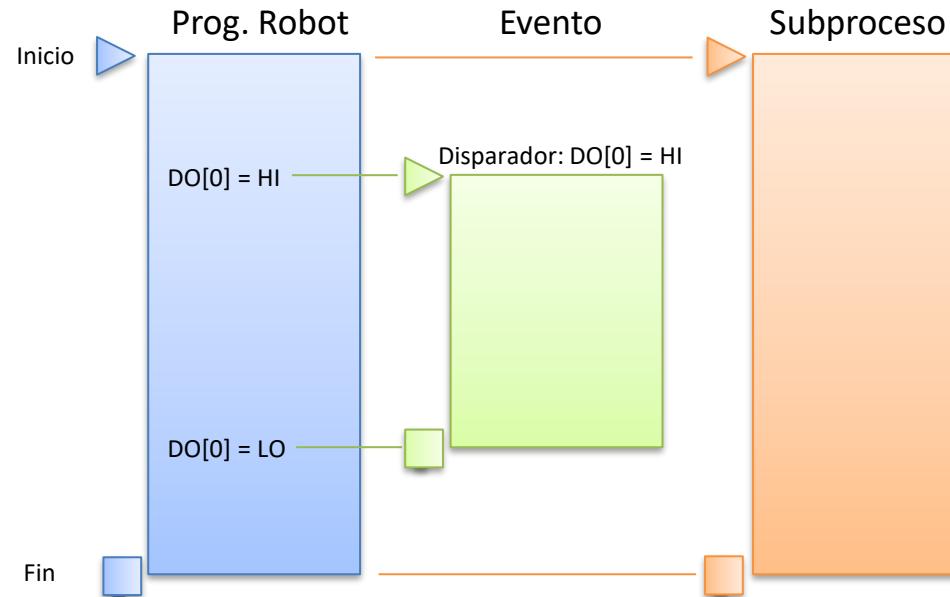
Ejercicio práctico

- 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 Punto_de_paso_1 y Punto_de_paso_2
 - Incrementar la variable *Contador* tras cada movimiento
 - Cuando el contador alcance los 10 ciclos mover ir a limpieza de herramienta (Punto_de_paso_3)
 - Cuando el contador alcance los 20 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

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



Subproceso

- Configuraciones
 - Bucle perpetuo
 - Seguir ejecución de prog.

Programa de robot

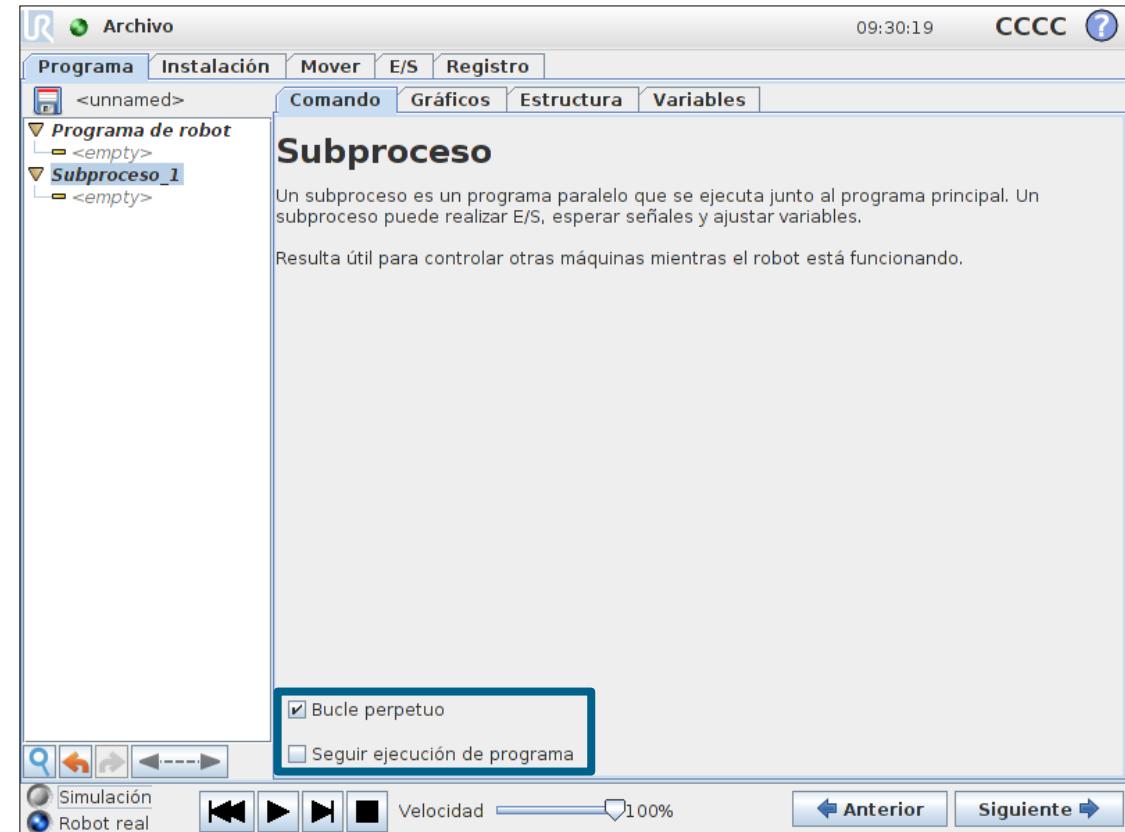
```
MoveJ  
    Punto_de_paso_1  
    Punto_de_paso_2  
IF DI[0] = True  
    Detener
```

Programa de robot

```
MoveJ  
    Punto_de_paso_1  
    Punto_de_paso_2
```

Subproceso_1

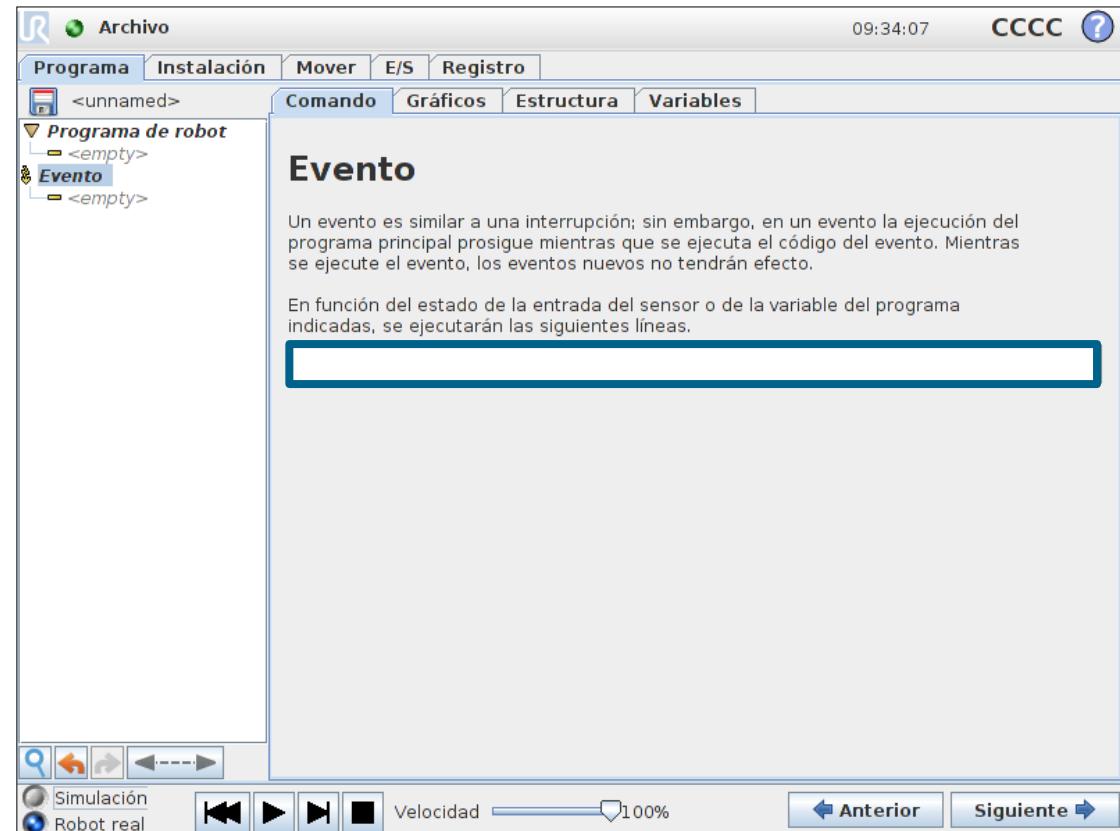
```
IF DI[0] = True  
    Detener  
Esperar 0.01
```



- Guardar programa de ejemplo como thread.urp

Evento

- Activación
 - Define la condición para ejecutar el evento

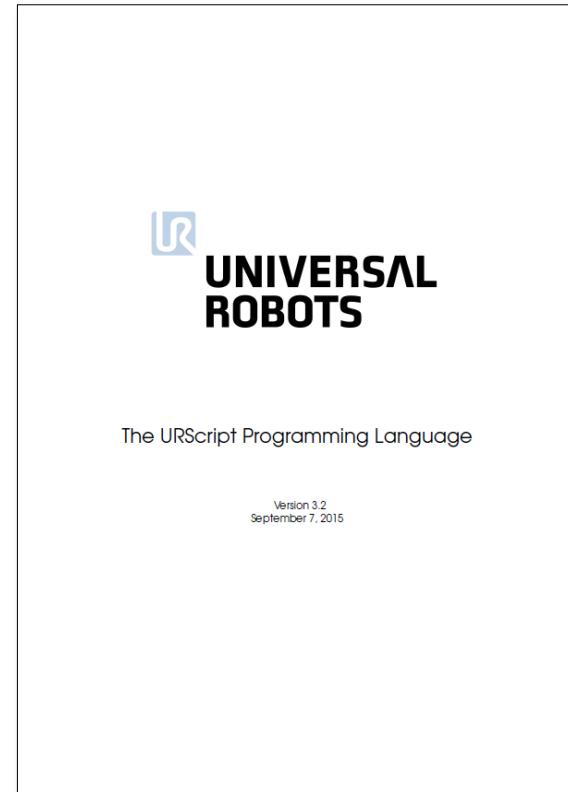


Programa de robot
MoveJ
 Punto_de_paso_1
 Punto_de_paso_2
Evento DI[0] = True
 Detener

- Guardar programa de ejemplo como event.urp

¿Qué es el script?

- Introducción a URScript
 - Una descripción más detallada se abordará en la Formación Avanzada
- URScript
 - Lenguaje script de alto nivel desarrollado por UR
 - Similar al lenguaje script Python
 - El manual Script contiene las definiciones de todos los códigos script disponibles

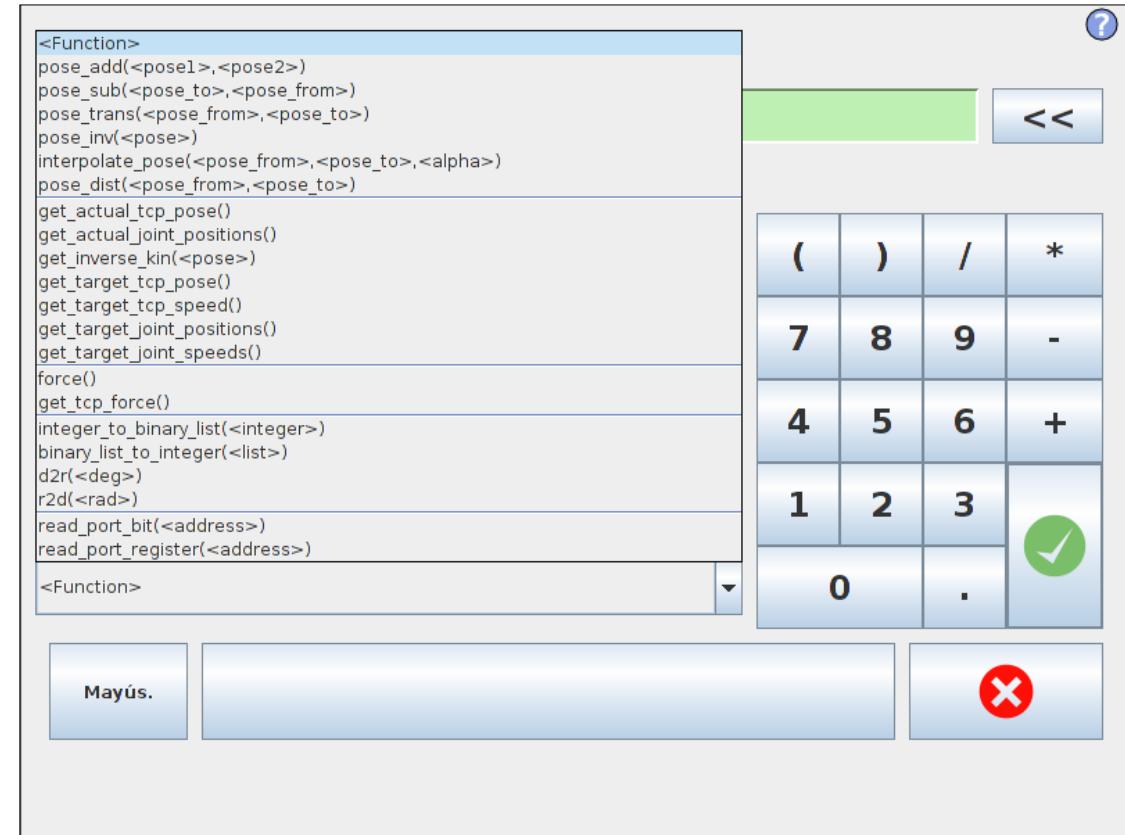


Cómo usar script

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

Programa de robot

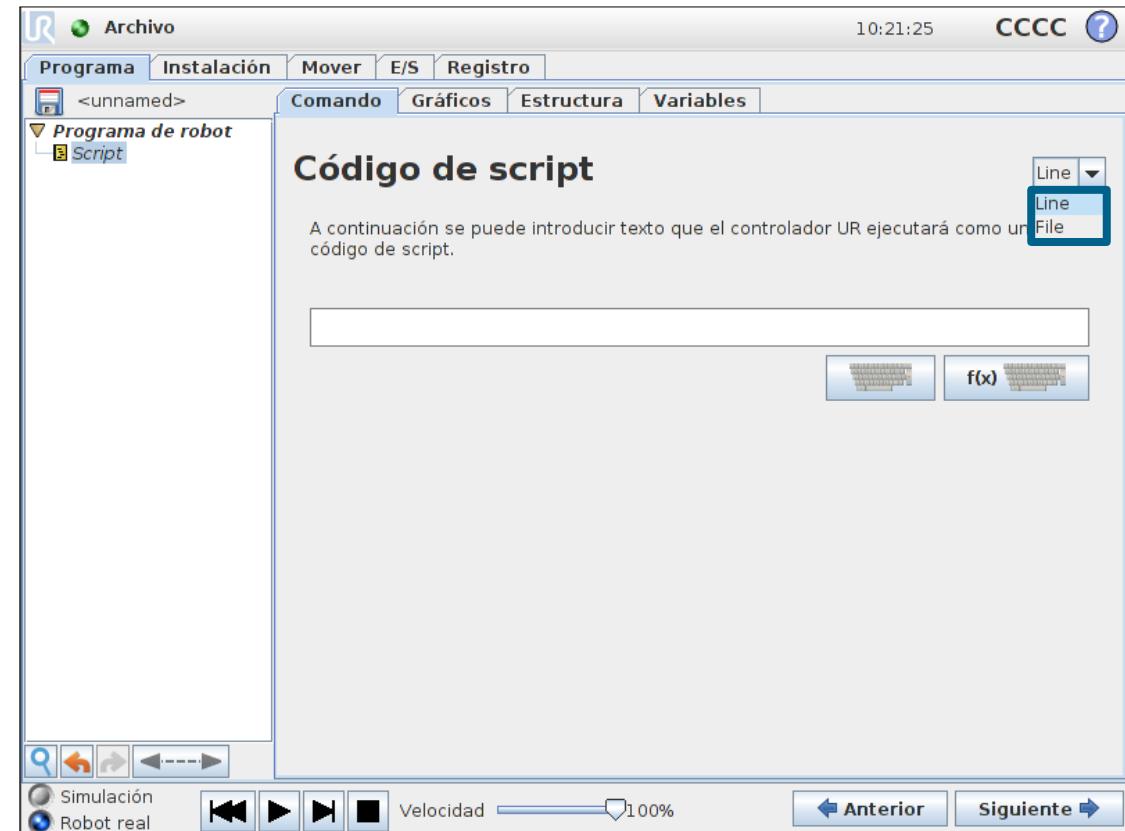
```
MoveL  
    Punto_de_paso_1  
    IF force() < 30  
        Punto_de_paso_2
```



- Guardar programa de ejemplo como *force_feedback.urp*

Comando Código de script

- Línea
 - Insertar un comando script
- Archivo
 - Llamada a un archivo múltiples códigos script

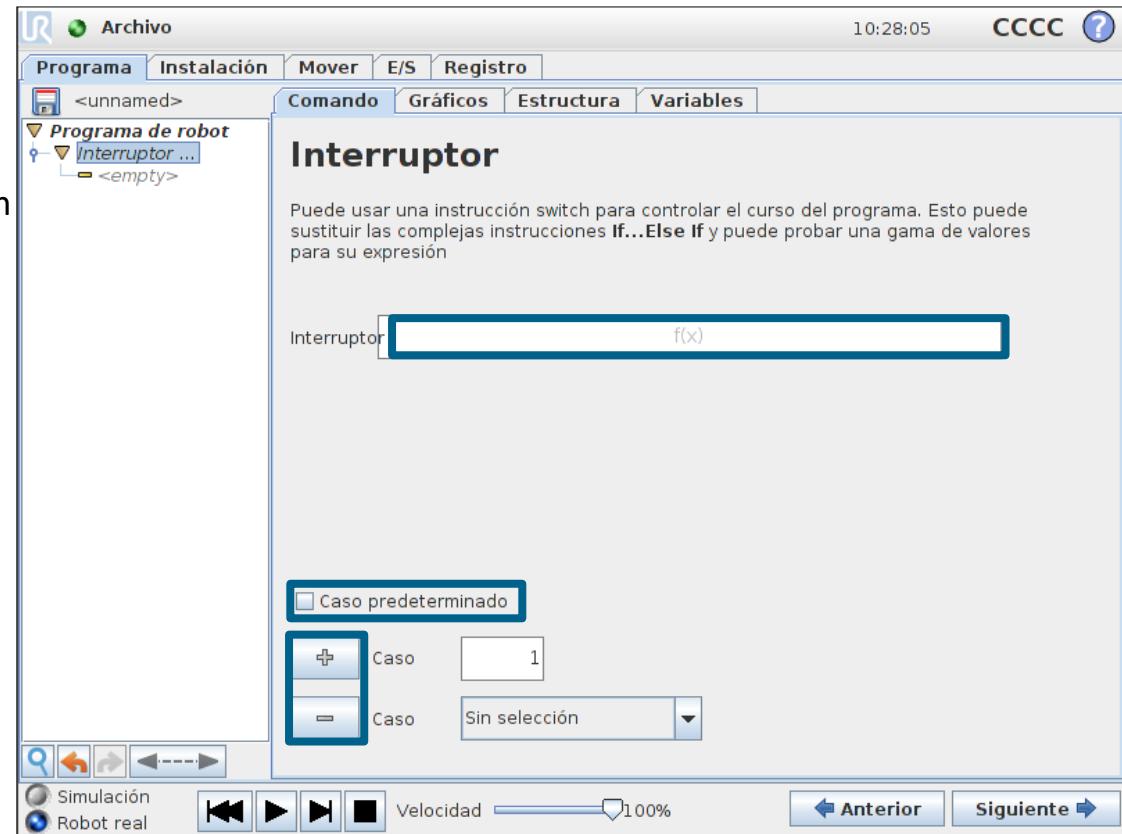


Programa de robot
set_digital_out(0,True)
Esperar 0.5
set_digital_out(0,False)
Esperar 0.5

- Guardar programa de ejemplo como script_line.urp

Comando *Interruptor*

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



Comando Interruptor

- Número de caso
 - Selecciona número de caso

Programa de robot

```
Seleccion_no:=‘Ingresar número entre 1 y 3’
```

```
Interruptor Seleccion_no
```

```
Caso 1
```

```
Aviso
```

```
Caso 2
```

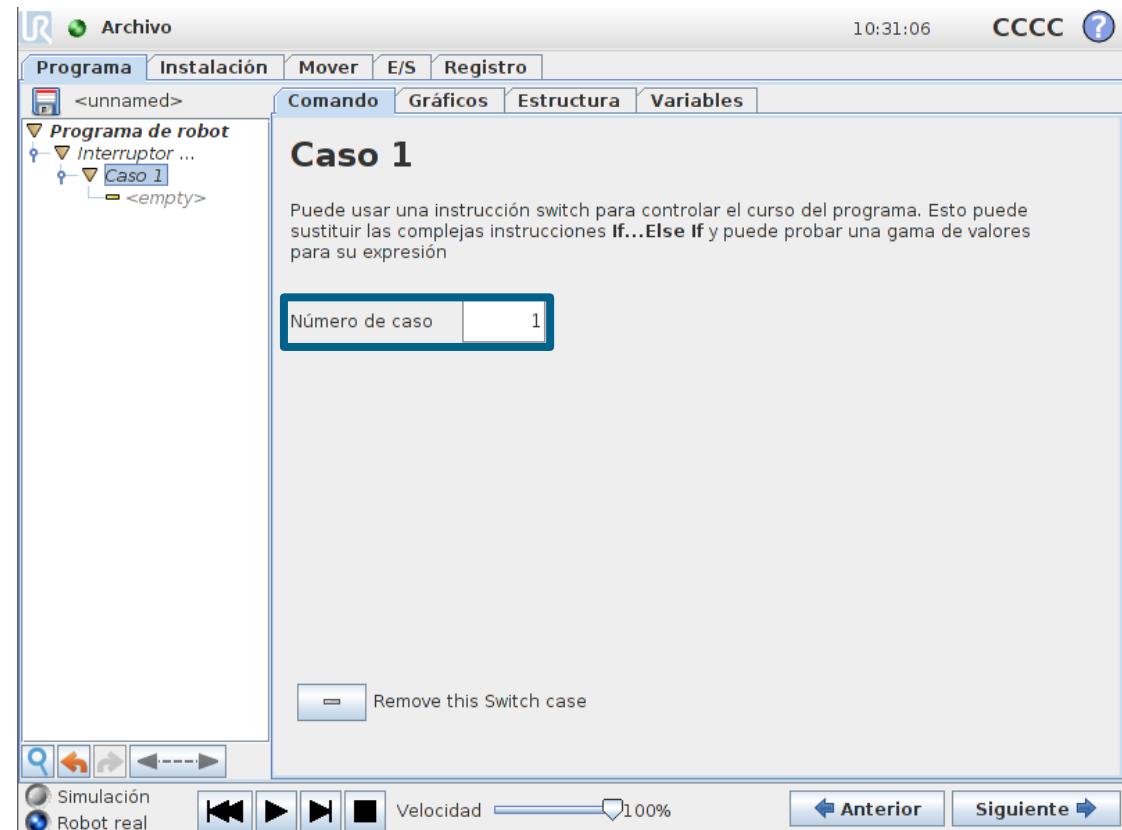
```
Aviso
```

```
Caso 3
```

```
Aviso
```

```
Caso predeterminado
```

```
Aviso
```



- Guardar programa de ejemplo como Interruptor.urp

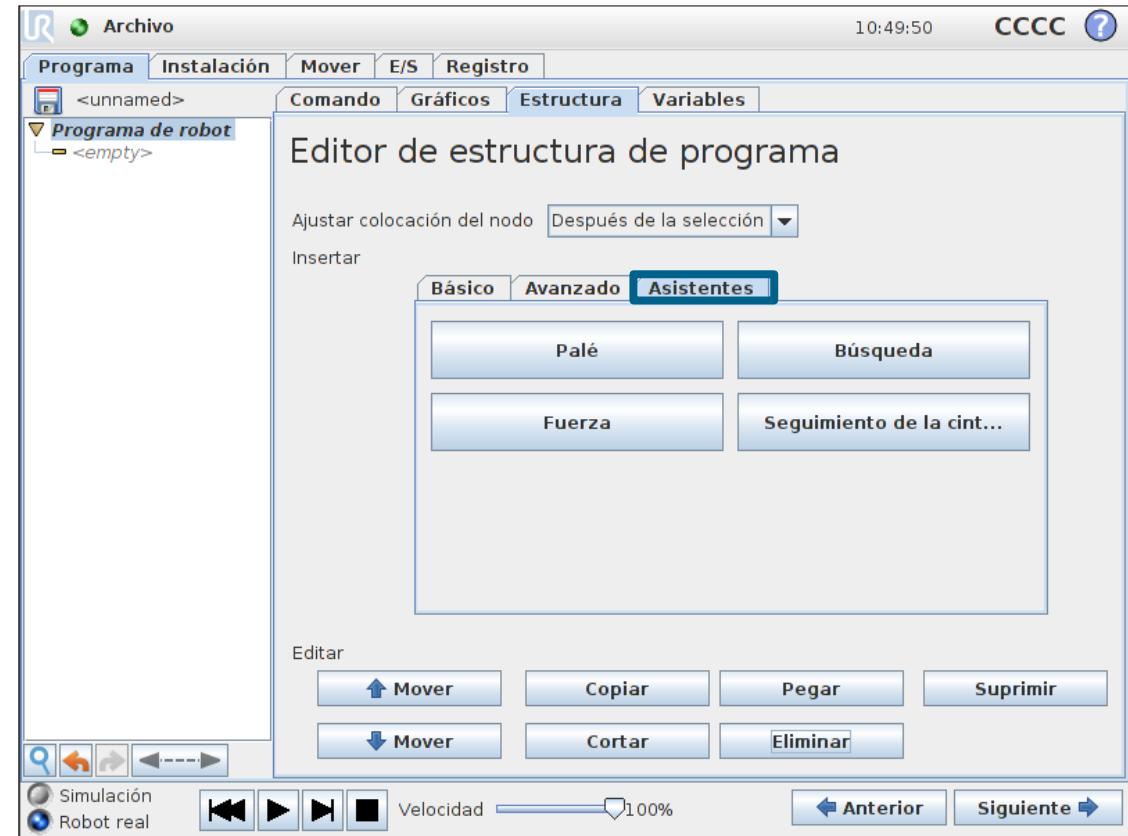
Ejercicio práctico

- Crear un Programa utilizando Subprocesos y URScript
 - Programa Principal:
 - Crear un simple MoveL entre Punto_de_paso_1 y Punto_de_paso_2
 - Subproceso_1:
 - Crear un subproceso con una asignación que llame a la función URScript force() y almacene el valor devuelto en una variable
 - Insertar una instrucción de espera de 0.01 segundos
 - Configurar la ejecución del subproceso en bucle perpetuo
 - Ejecutar el programa y seleccionar la pestaña de Variables.
 - Compruebe que el valor de la fuerza queda monitorizado en la variable, y este cambia si se opone resistencia al movimiento del brazo.
-
- Guardar programa de ejemplo como Lab_exe_7_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

Asistentes

- Palé
 - Función de paletizado
 - Patrones
- Búsqueda
 - Función de búsqueda
- Fuerza
 - Función de fuerza
- Seguimiento de la cinta transportadora
 - Sigue el movimiento del transportador
- Euromap
 - Máquinas de inyección
 - Opcional



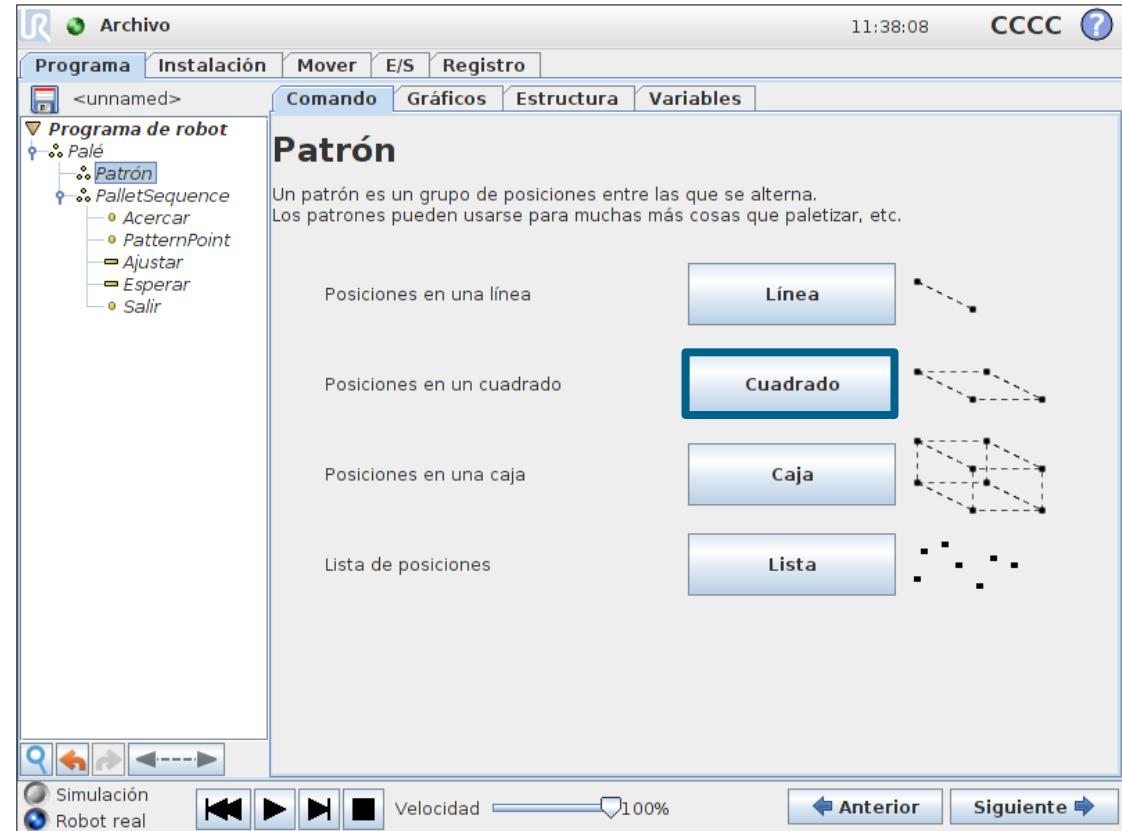
Asistente Palé

• Patrón

- Determina el patrón de paletizado
- Posiciones
 - Línea
 - Cuadrado
 - Caja
 - Lista

• PalletSequence

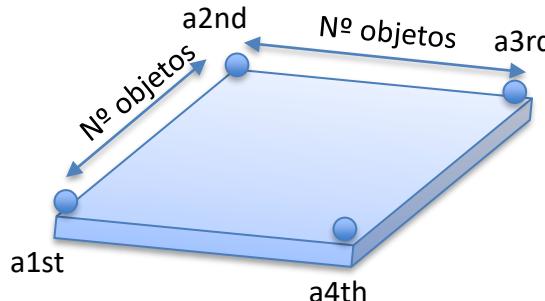
- Qué hace el robot en cada posición del patrón



Patrón: Cuadrado

- Patrón
 - Utilizar Patrón: Cuadrado
 - Indicar objetos entre
 - Punto 1 a 2
 - Punto 2 a 3
 - Enseñar las 4 esquinas

Patrón



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

- Archivo**: Archivo, Instalación, Mover, E/S, Registro
- Programa**: <unnamed>
- Comando**: Comando, Gráficos, Estructura, Variables
- Patrón: Cuadrado**: Cambiar tipo
- Descripción**: La variable de recuento "cnt_1" utilizada para el cruce
Recuerde la posición de recorrido entre ejecuciones del programa
- Parámetros**: Recuento de intervalo entre punto 1 y 2: 5
Recuento de intervalo entre punto 2 y 3: 5
- Parámetros compartidos**: Velocidad herramienta: 250 mm/s, Aceleración herramienta: 1200 mm/s², Restablecer valores predet.
- Botones**: Añadir punto nuevo, Quitar punto, Anterior, Siguiente
- Estado**: Simulación (radio botón seleccionado), Robot real, Velocidad: 100%

PalletSequence

- PalletSequence
 - Enseñar posiciones
 - PatternPoint
 - Acercar
 - Salir
 - Definir acciones

Programa de robot

Pallet

Patrón: Cuadrado

a1st_Corner
a2nd_Corner
a3rd_Corner
a4th_Corner

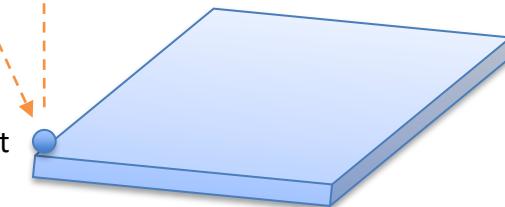
PalletSequence

Acercar
PatternPoint_1
Ajustar DO[0] =Encender
Esperar 0.5
Salir

PalletSequence

Acercar Salir

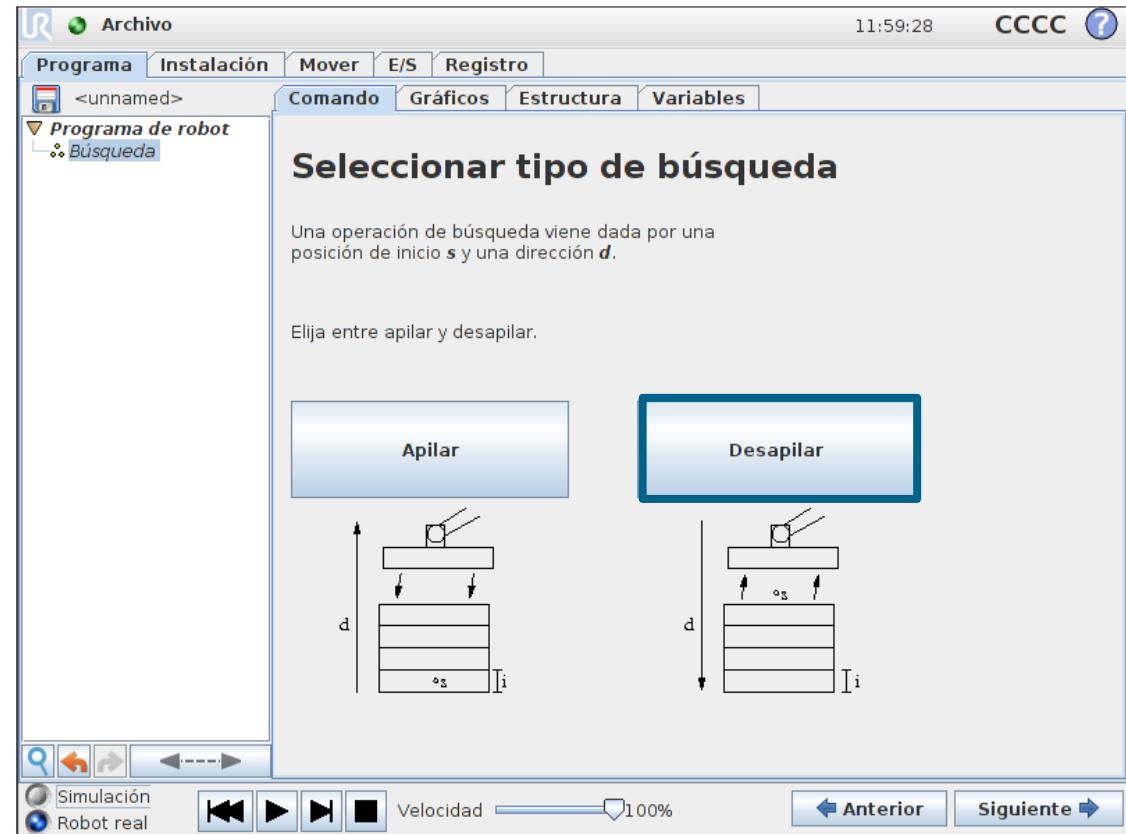
PatternPoint = a1st



- Consejo práctico: grabar PatternPoint como a1st_Corner
- Guardar programa de ejemplo como pallet.urp

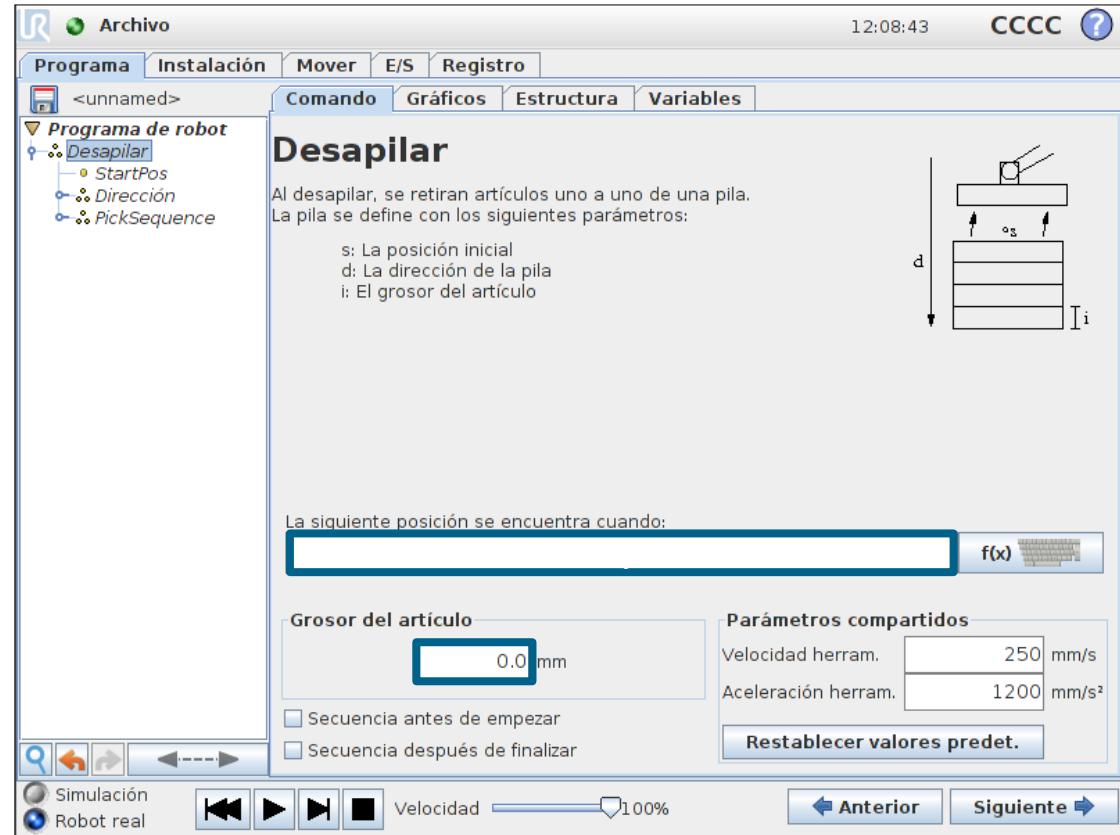
Asistente Búsqueda

- Apilar
 - Añadir objetos en la pila
- Desapilar
 - Recoger objetos de la pila



Desapilar

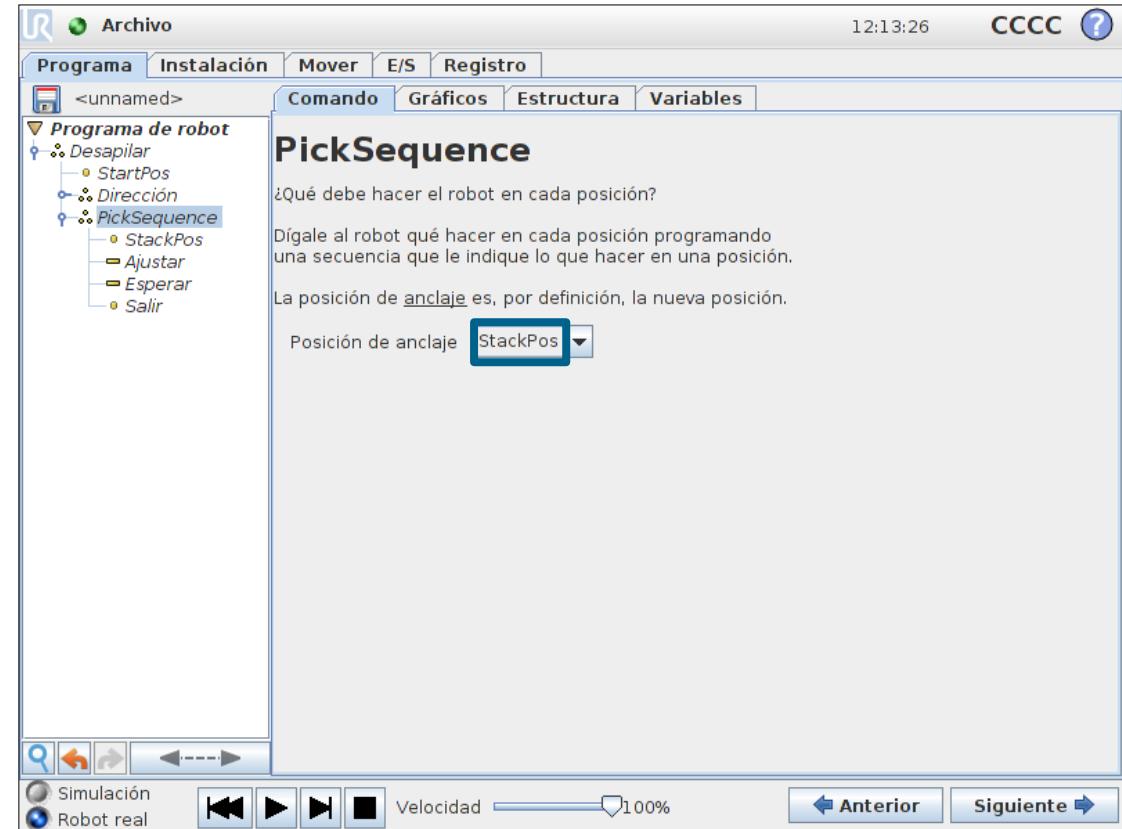
- Ejemplo
 - Insertar búsqueda
 - Seleccionar desapilar
- Desapilar
 - Fijar posición StartPos
 - Desde dónde empezar la búsqueda
 - Establecer dirección de pila
 - Cualquier dirección lineal
 - FromPos
 - ToPos
 - Establecer grosor de objeto
 - Grosor único para todos
 - Establecer condición d
 - Determina cuando se encuentra el objeto
 - Utilizar force() > 30



Desapilar

- **PickSequence**

- **StackPos**
 - Define donde coger el objeto al encontrarlo
- **Salir**
 - Define cómo salir después de recoger el objeto
- **Definir acciones**
 - Ajustar
 - Esperar



- Consejo práctico: Fijar StackPos igual que StartPos

Desapilar

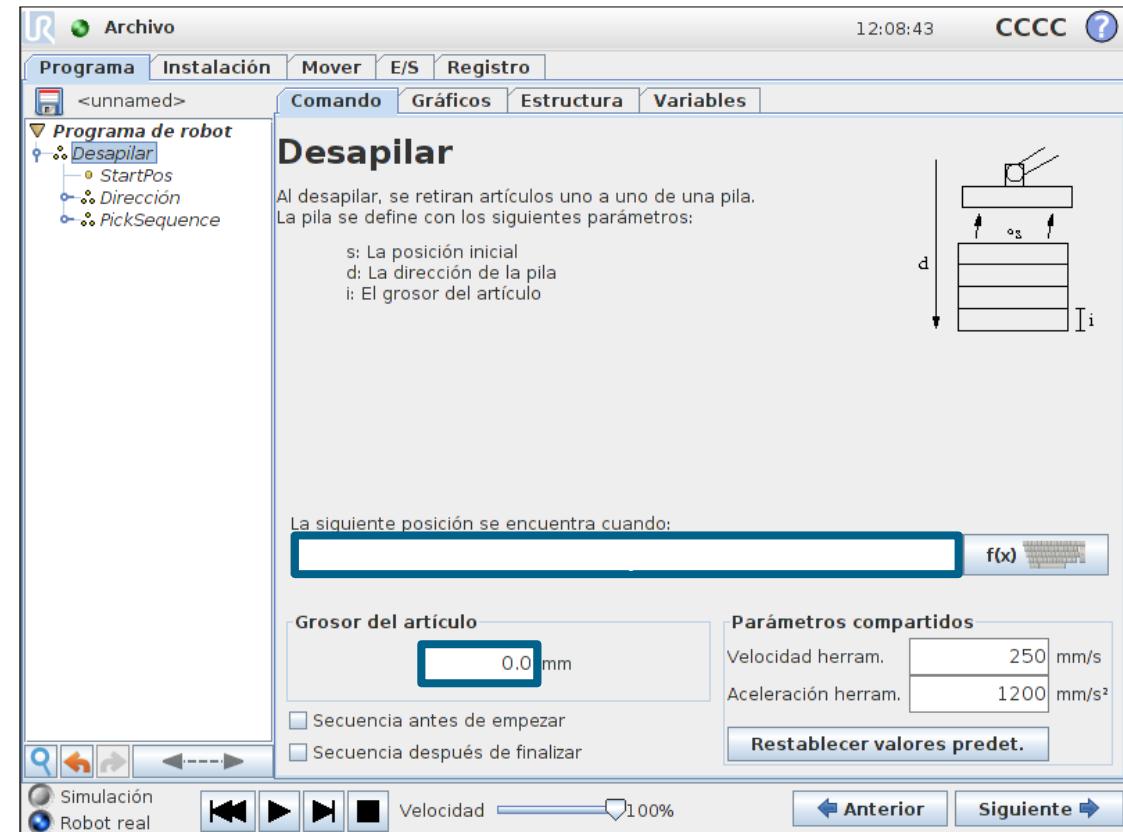
- Programa ejemplo
 - Añadir una carpeta para depositar el objeto

Programa de robot

```
Desapilar
  StartPos
  Dirección
    FromPos
    ToPos
  PickSequence
  StackPos
  Ajustar DO[0] = Encender
  Esperar 0.5
  Punto_de_paso_1
```

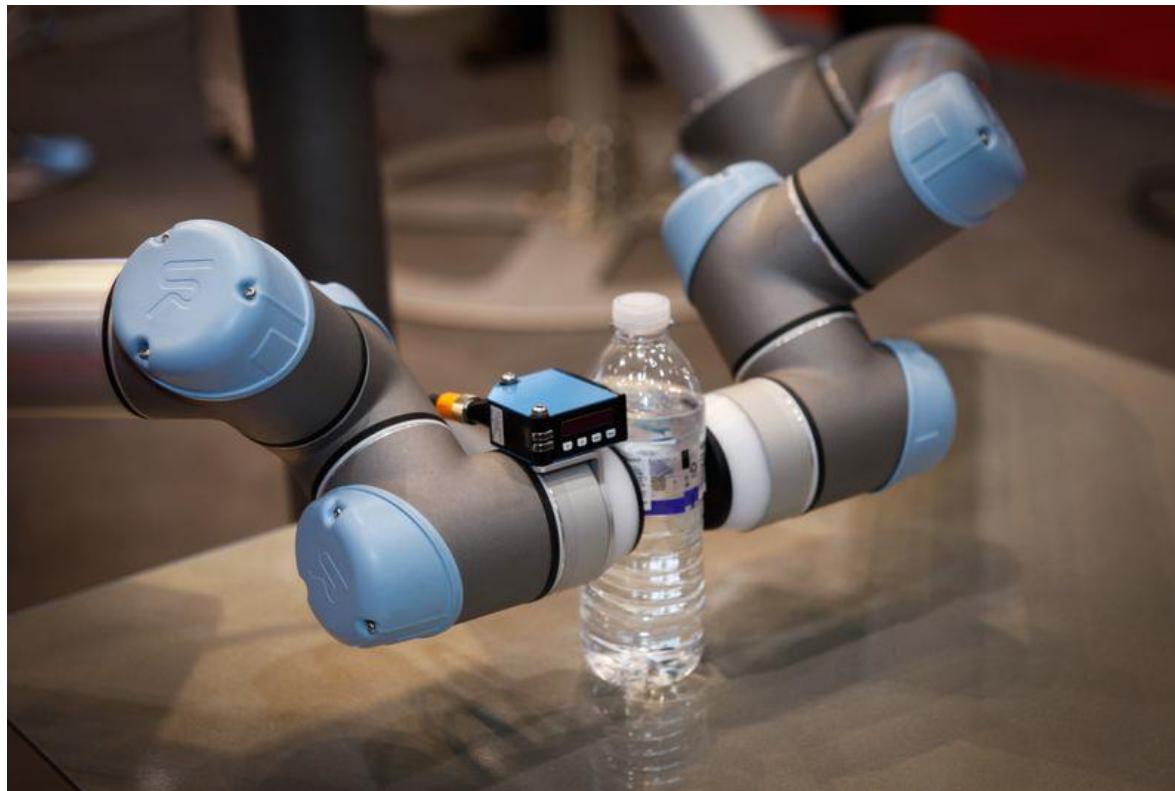
Dejar_objeto

```
  Punto_de_paso_2
  Punto_de_paso_3
  Ajustar DO[0] = Apagar
  Esperar 0.5
  Punto_de_paso_2
```



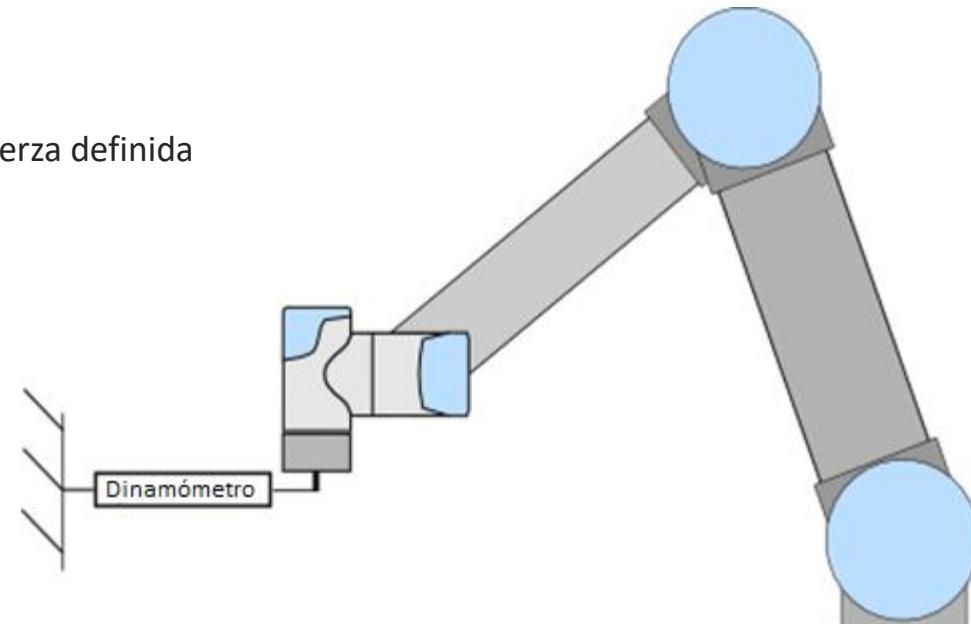
- Guardar programa de ejemplo como seek_destack.urp

Control de fuerza



Asistente Fuerza

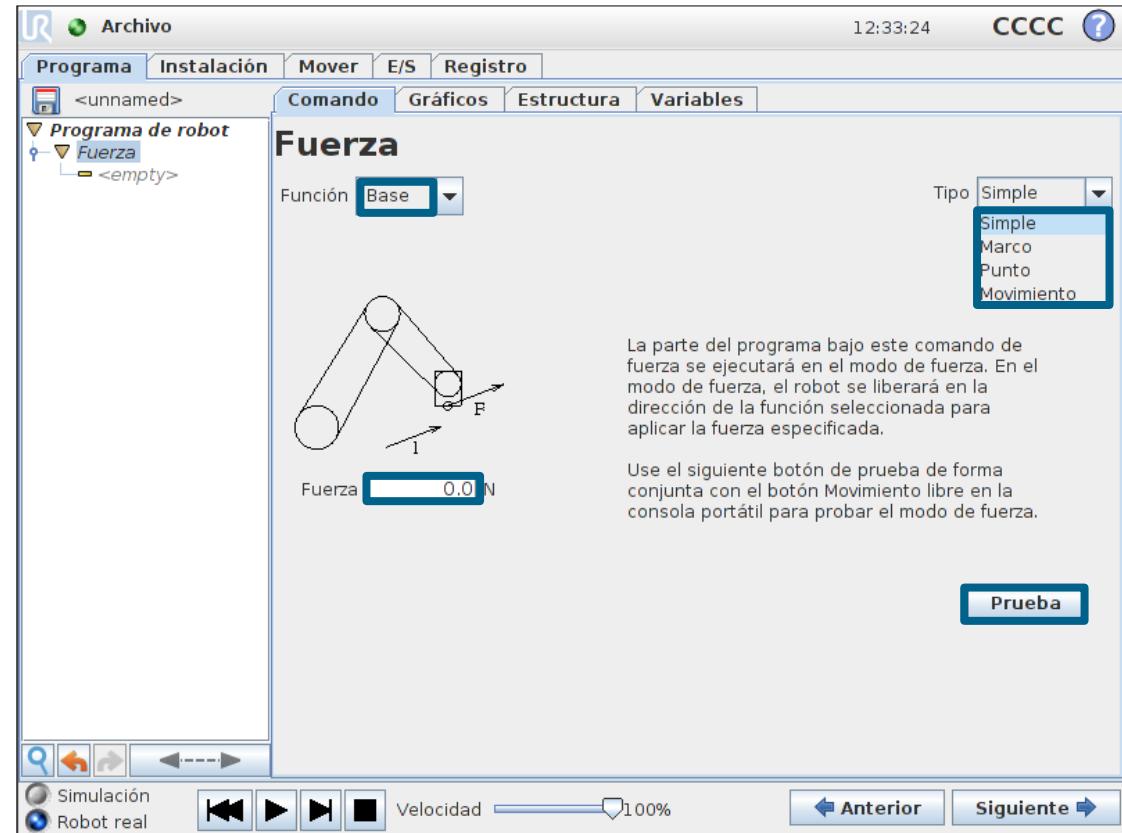
- Características
 - Interacción con el entorno
 - Ajuste de posición para lograr la fuerza definida



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

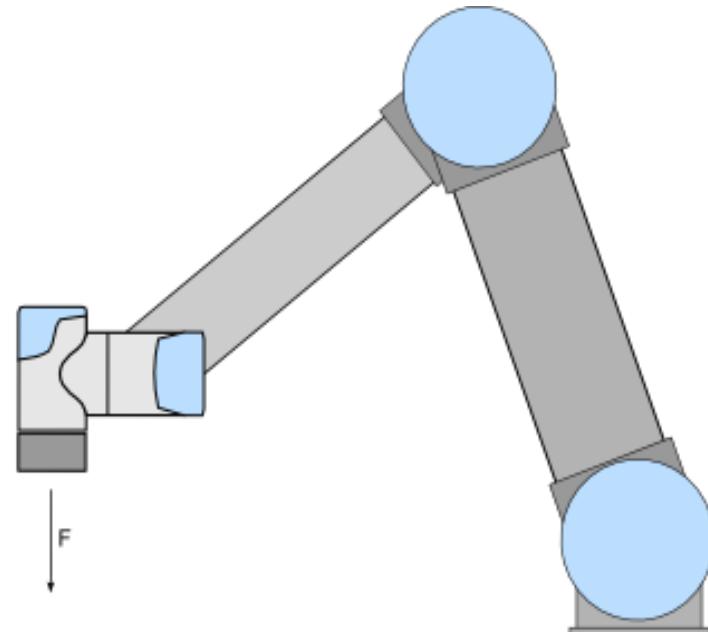
Cómo usar el asistente Fuerza

- Configuraciones
 - Tipo de fuerza
 - Valor de fuerza
 - Dirección de la fuerza
- Prueba sencilla
 - Prueba de aprendizaje



Tipo de fuerza: Simple

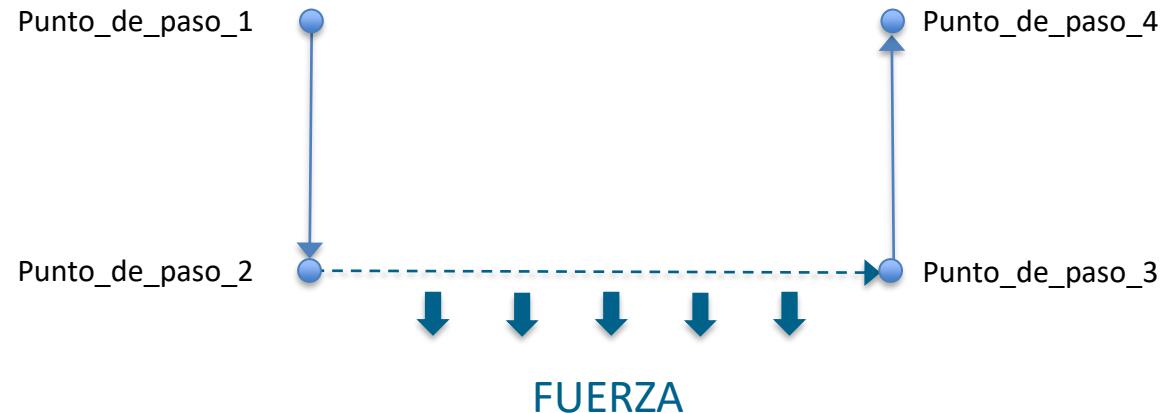
- Características
 - Un eje en modo adaptable
 - Fuerza en dirección del eje-Z de la función
- Test
 - Añadir Fuerza a un nuevo programa
 - Establecer Tipo: Simple
 - Establecer Función: BASE
 - Establecer Fuerza: 30 N
 - Prueba de aprendizaje
 - Establecer Fuerza: -30 N
 - Prueba de aprendizaje
 - Establecer Función: TOOL
 - Prueba de aprendizaje



Tipo de fuerza: Simple

- Programa ejemplo

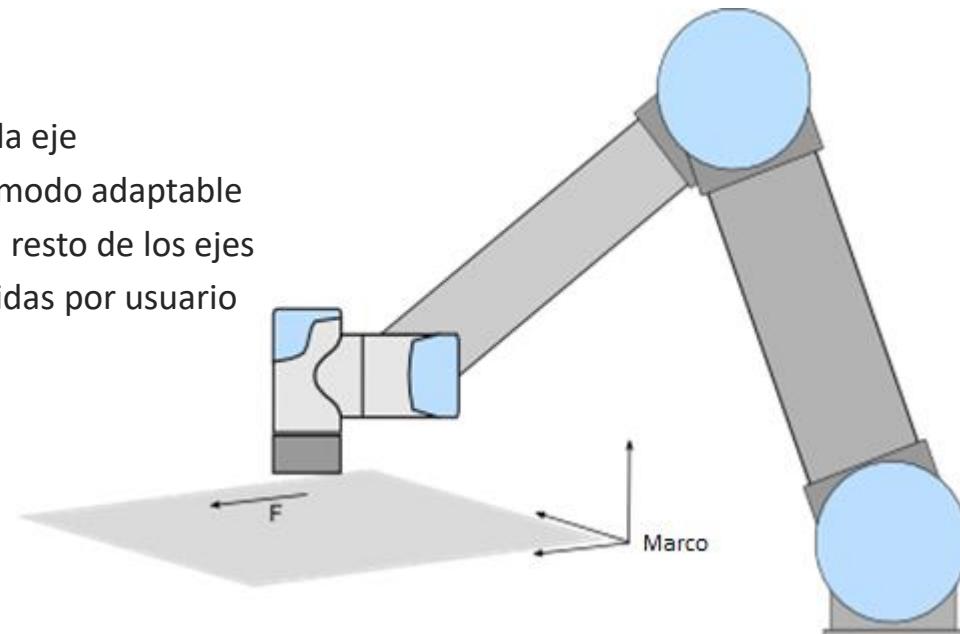
```
Programa de robot
MoveL
  Punto_de_paso_1
  Punto_de_paso_2
  Fuerza
    Punto_de_paso_2
    Punto_de_paso_3
    Punto_de_paso_4
```



- Guardar programa de ejemplo como force_simple.urp

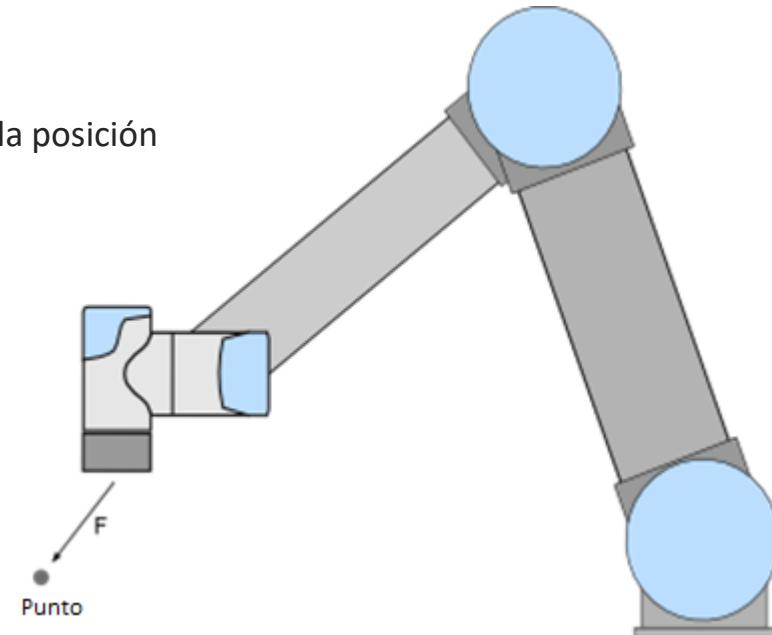
Tipo de fuerza: Marco

- Características
 - Múltiples ejes en modo adaptable
 - Valor de fuerza específico sobre cada eje
 - Velocidad limitada para los ejes en modo adaptable
 - Deriva de posición limitada sobre el resto de los ejes
 - Base, Herramienta, funciones definidas por usuario



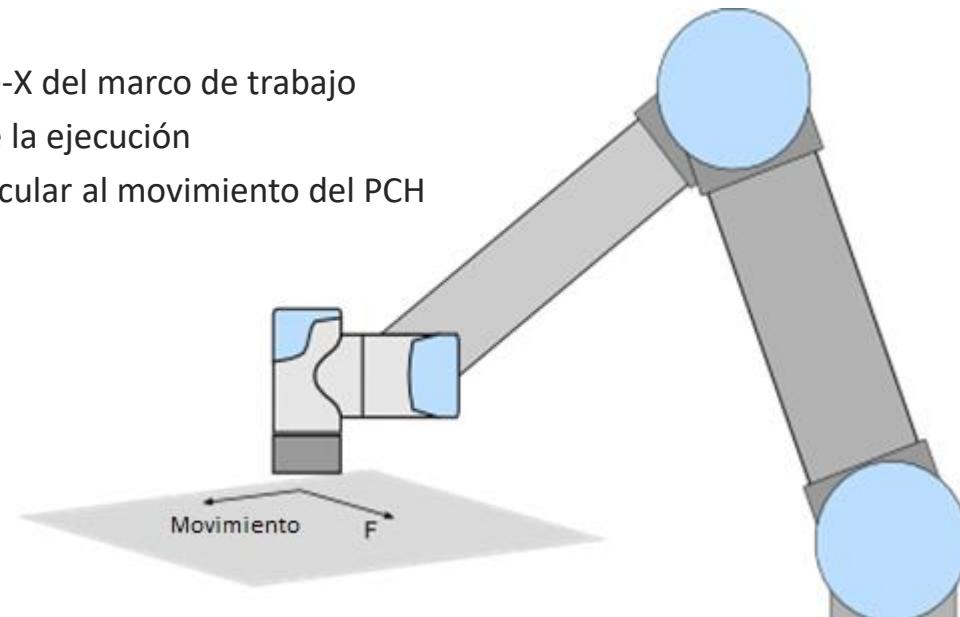
Tipo de fuerza: Punto

- Características
 - Específico sobre una función tipo punto
 - Fuerza sobre el eje-Y del marco de tarea, desde la posición del PCH hacia el punto definido como función
 - El marco de tarea cambia durante la ejecución



Tipo de fuerza: Movimiento

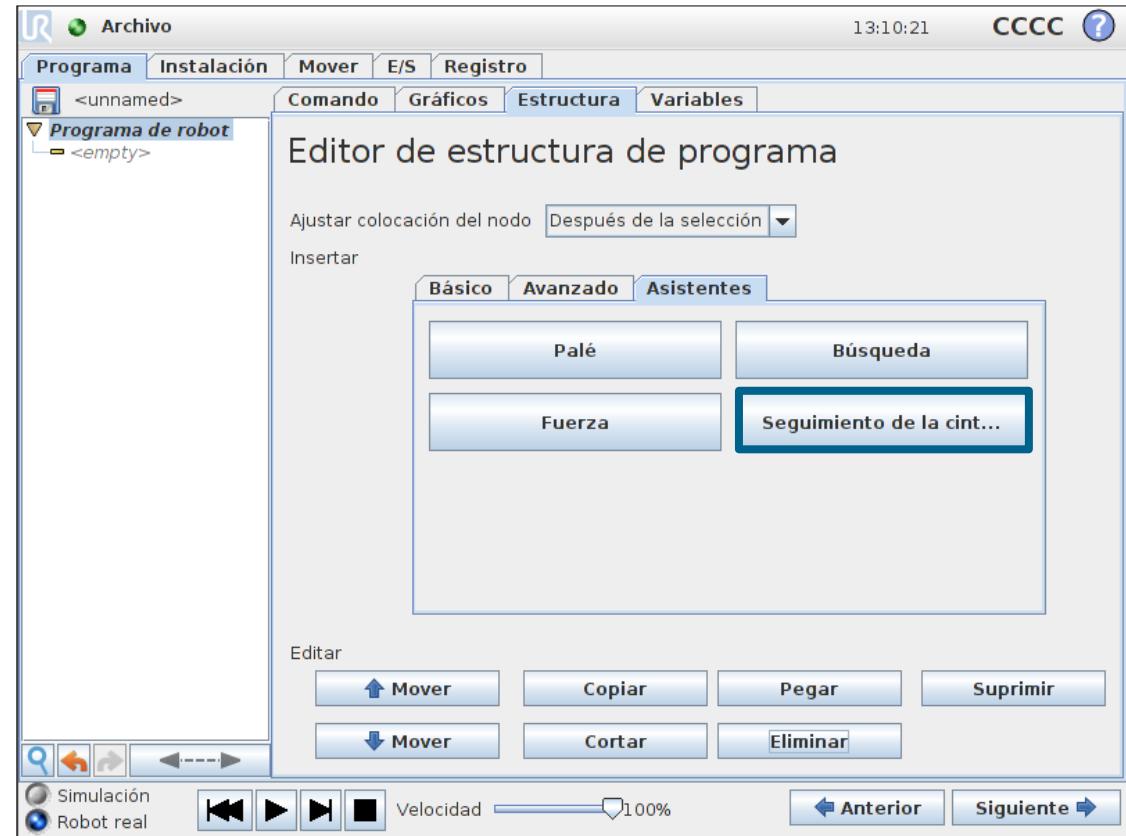
- Características
 - El movimiento del PCH define el eje-X del marco de trabajo
 - El marco de trabajo cambia durante la ejecución
 - El eje-Y, adaptable, queda perpendicular al movimiento del PCH



- Eje-X no adaptable
- *Prueba de aprendizaje no aplicable*

Asistente Seguimiento de la cinta transportadora

- Seguimiento de la cinta transportadora
 - Definen las partes del programa dónde el robot debe seguir el movimiento de la cinta transportadora
 - No incluido en la Formación Principal



Ejercicio práctico

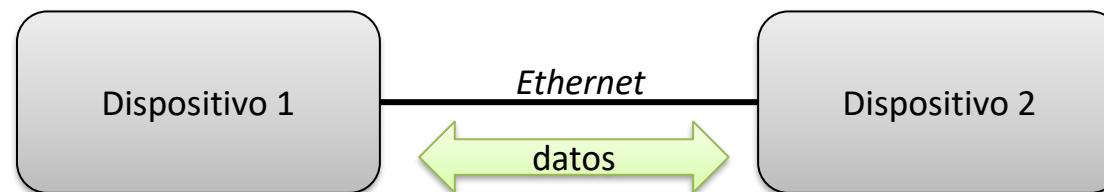
- Usar el asistente de paletizado para crear un programa que simule el colocar piezas en cada una de las posiciones del siguiente diagrama.
Utilizar el patrón impreso sobre formato A3 que tendrás disponible para realizar este ejercicio.
- Programa de paletizado
 - Seleccionar el asistente de paletizado y crear un programa nuevo
 - Definir el número de posiciones en cada dimensión
 - Especificar la posición de cada una de las 4 esquinas del Palé
 - Fijar las posiciones de Acercar y Salir a 100mm directamente sobre el PatternPoint
 - Fijar un punto de paso simulando recoger la pieza antes de dejarla en cada posición del Palé
 - Guardar programa de ejemplo como Lab_exe_8_1.urp

X	X	X	X
X	X	X	X
X	X	X	X
X	X	X	X
X	X	X	X
X	X	X	X

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?

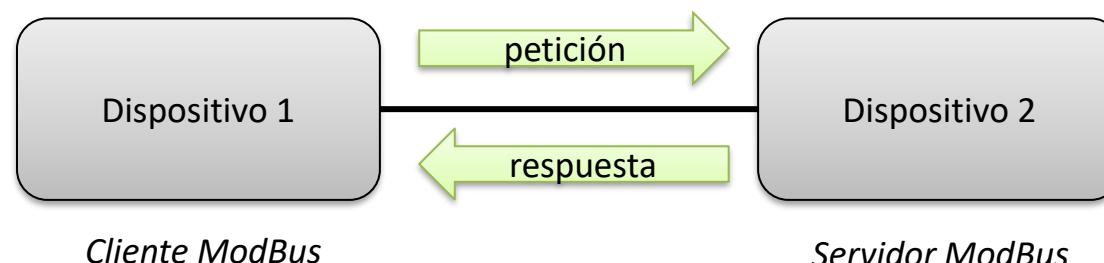
- Modbus TCP
 - Protocolo de comunicación basado en Ethernet
- Protocolo de comunicación
 - Un protocolo es un lenguaje común mediante el cual dos dispositivos pueden comunicarse
 - Posibilita la transmisión de datos entre los dispositivos



- 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

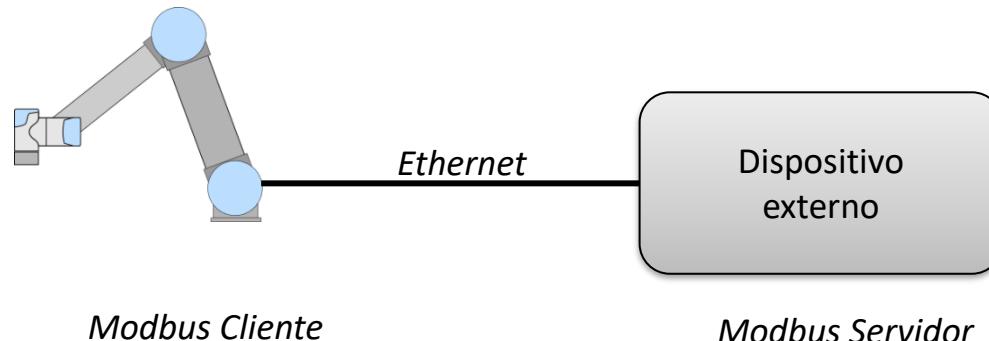
- Tipos de datos disponibles en Modbus TCP

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

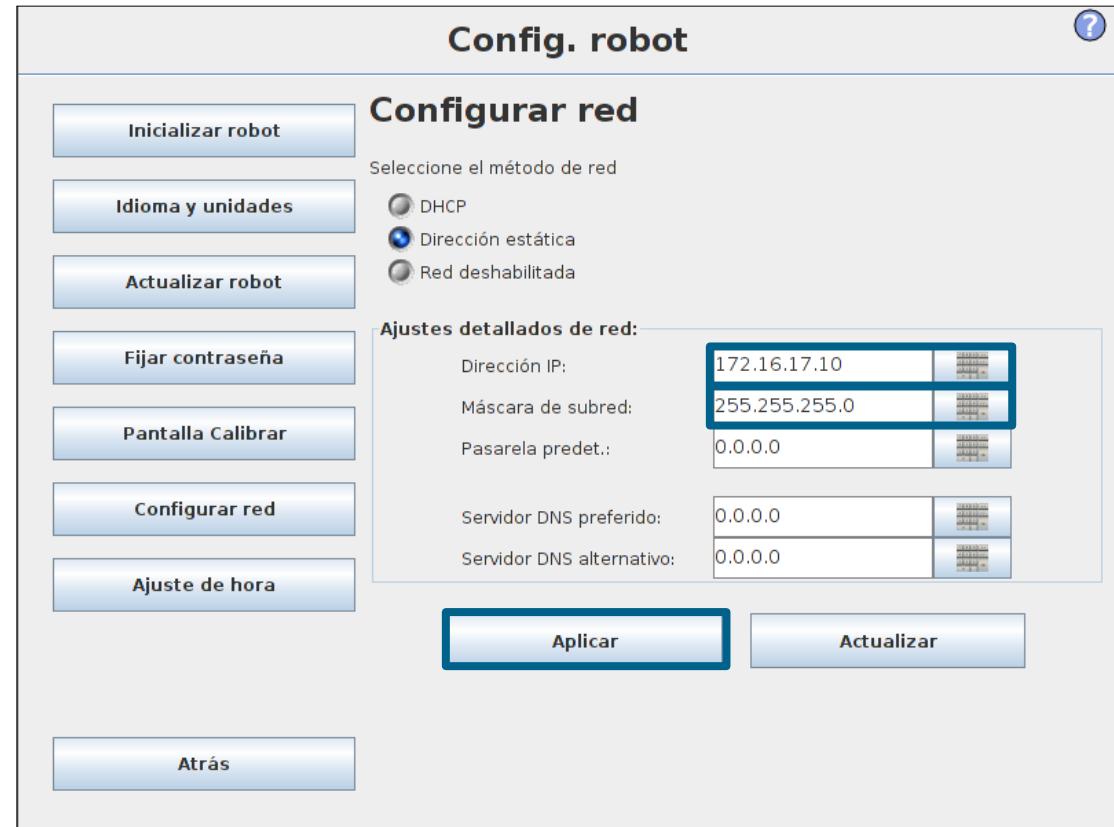


Modbus Cliente

Modbus Servidor

Configuración de red

- Ajustes
 - Método Dirección estática
 - Introducir la dirección-IP del robot
 - Introducir la máscara de subred
 - Aplicar 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



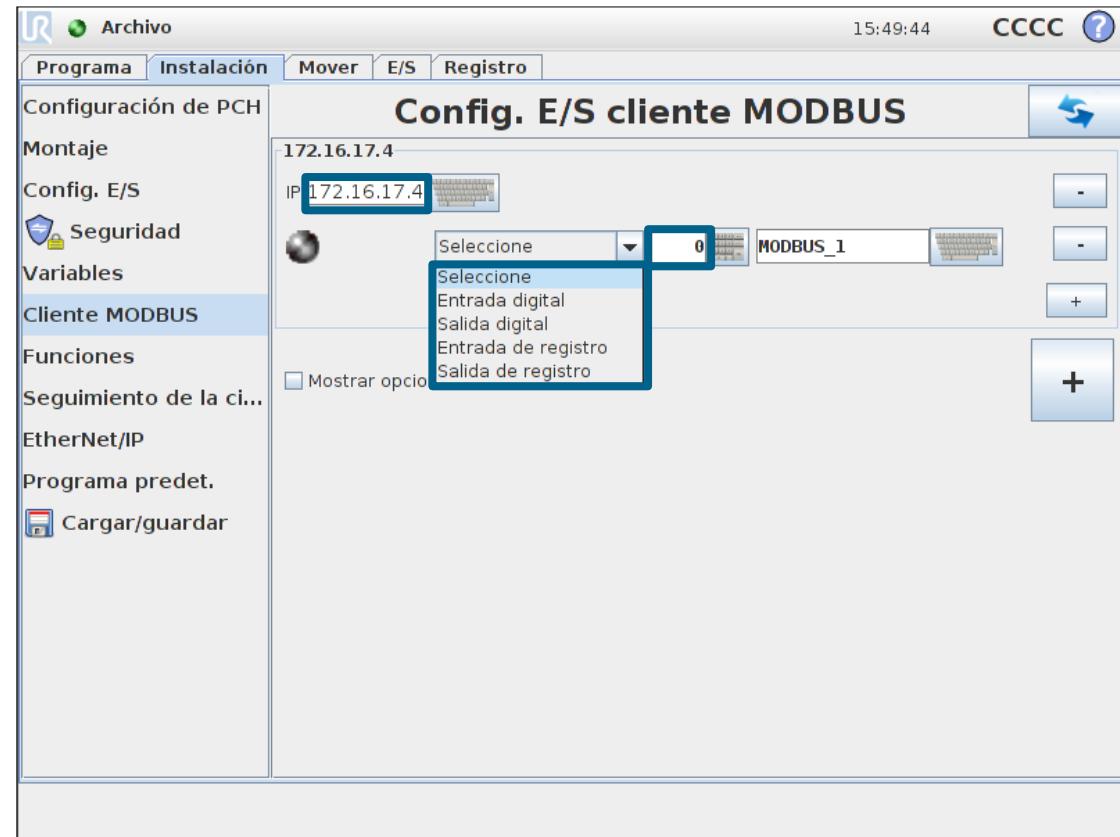
	Modbus Register Table (16-Bit Words)	Modbus Input Discretes Table (Bits)	Modbus Coil Table	Access	Function
Process data	0	0-15	—	Read only	Digital inputs (DIO)
	1	16-32	—	Read only	Digital inputs (DI)
	2	—	0-15	Read/write	Digital outputs
	3	—	—	Read only	Reserved

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

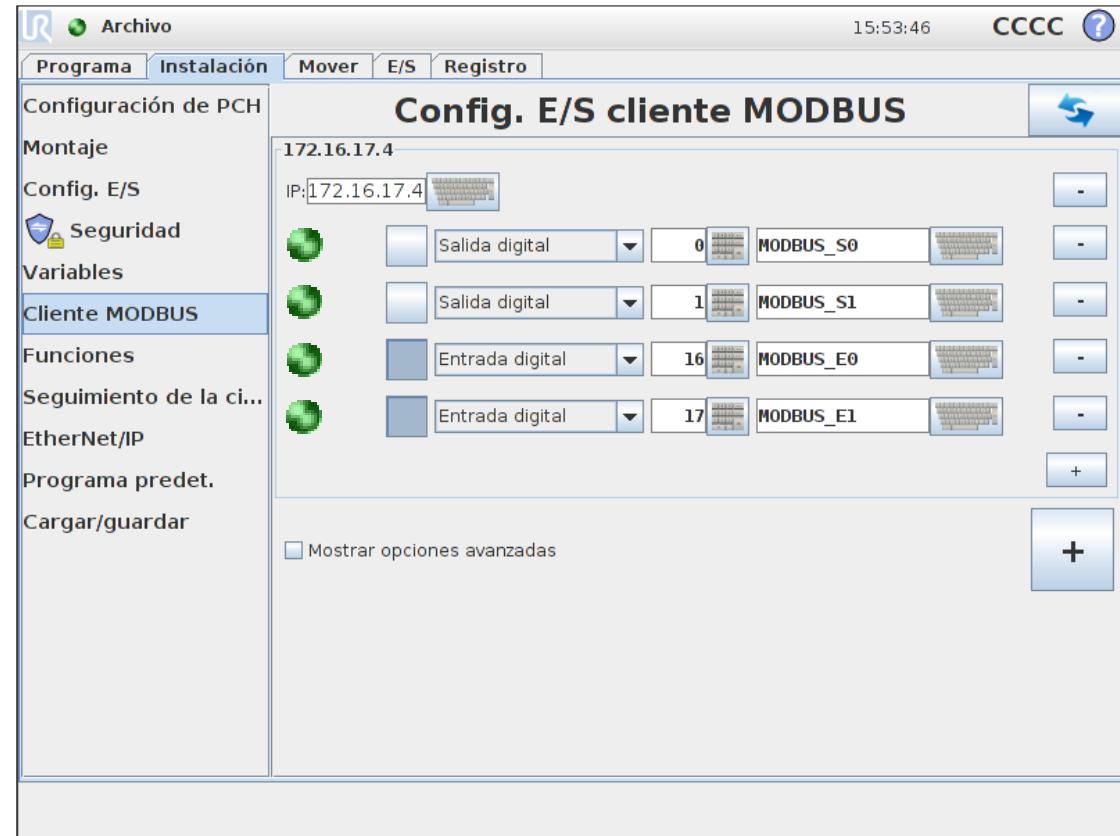
- Configurar dispositivo
 - Añadir nuevo dispositivo
 - Introducir la dirección-IP del dispositivo (Servidor)
- Configurar señales
 - Añadir señal en dispositivo
 - Definir tipo de dato
 - Establecer su dirección
 - Definir nombre de la señal



Configuración del servidor

- Configuración
 - 2 entradas digitales
 - 2 salidas digitales
 - Guardar Instalación
- Monitoreo
 - El estado de la señales aparece en la ficha E/S
- Estado de conexión

	Status
	Conexión ok
	Aviso frecuencia de refresco
	Sin conexión
E4	Código de excepción



Uso de las señales de Modbus en programa

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

Programa de robot

MoveL

Punto_de_paso_1

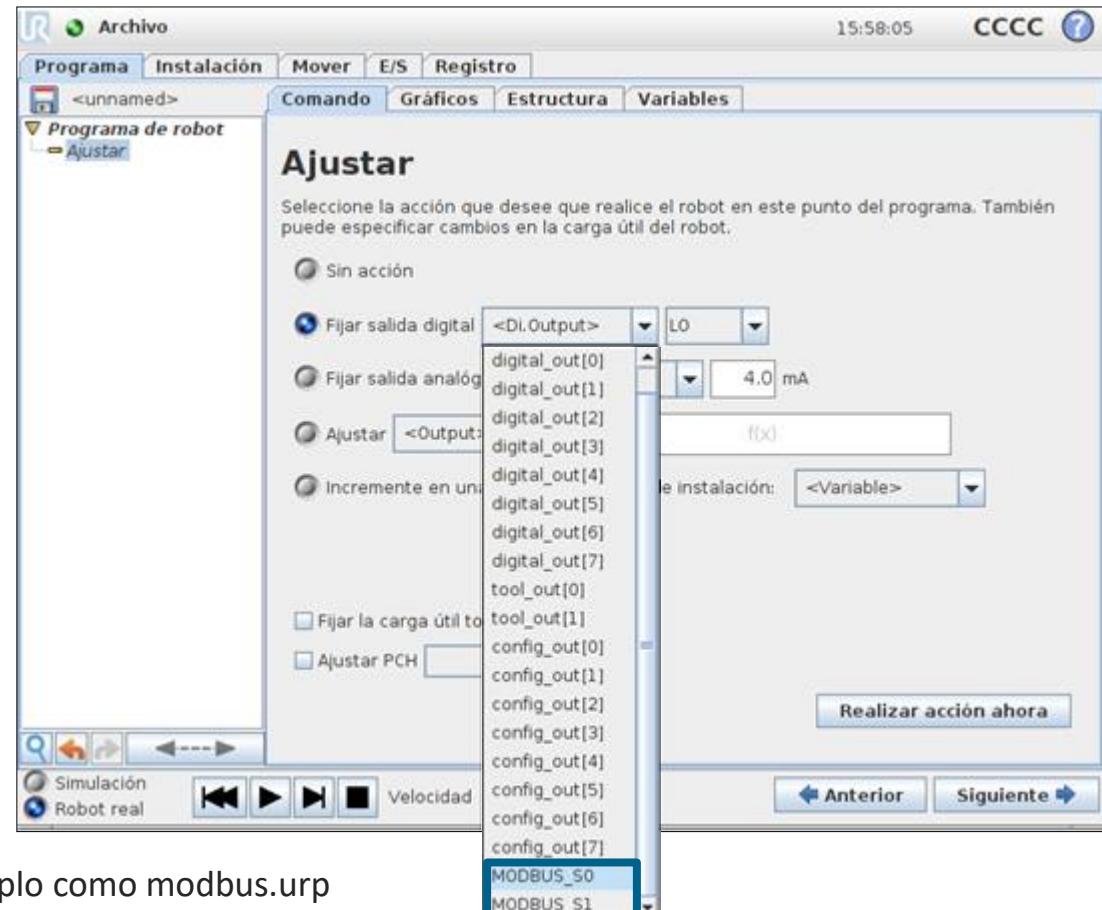
Ajustar MODBUS_S1 = Encender

Punto_de_paso_2

Punto_de_paso_3

Punto_de_paso_4

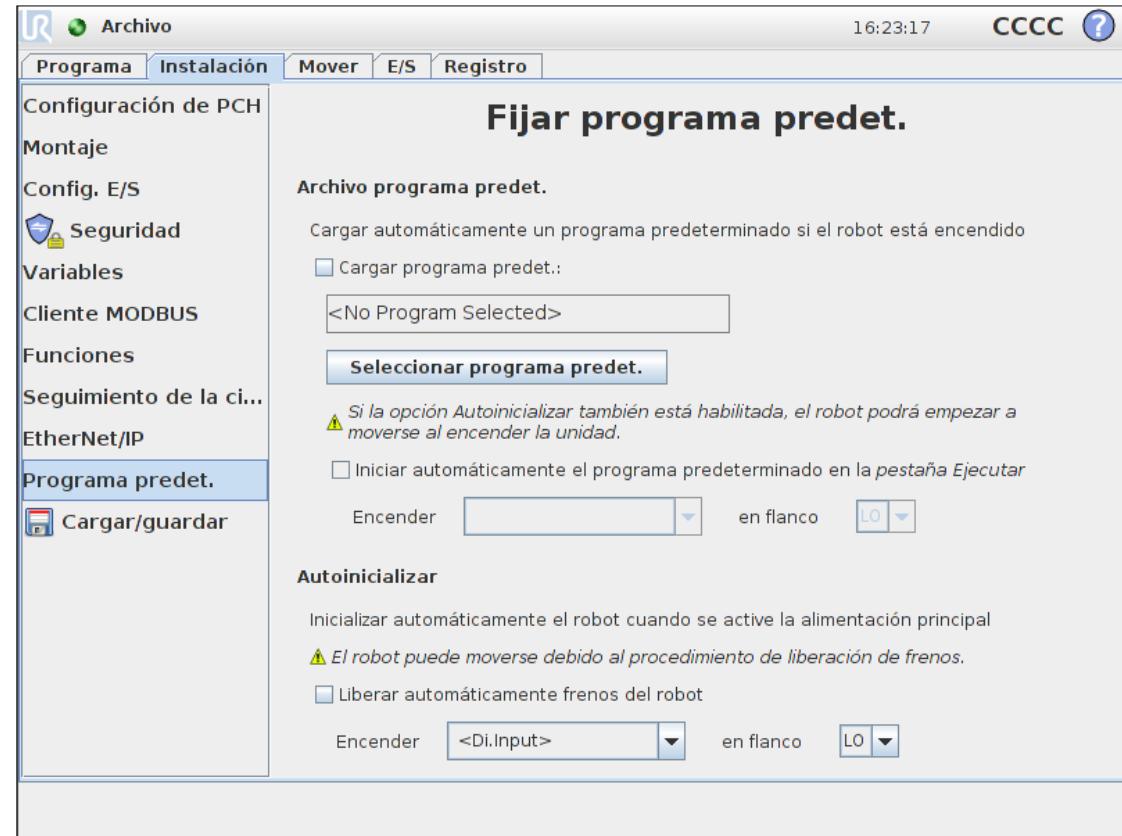
Ajustar MODBUS_S1 = Apagar



- Guardar programa de ejemplo como modbus.urp

Programa predeterminado

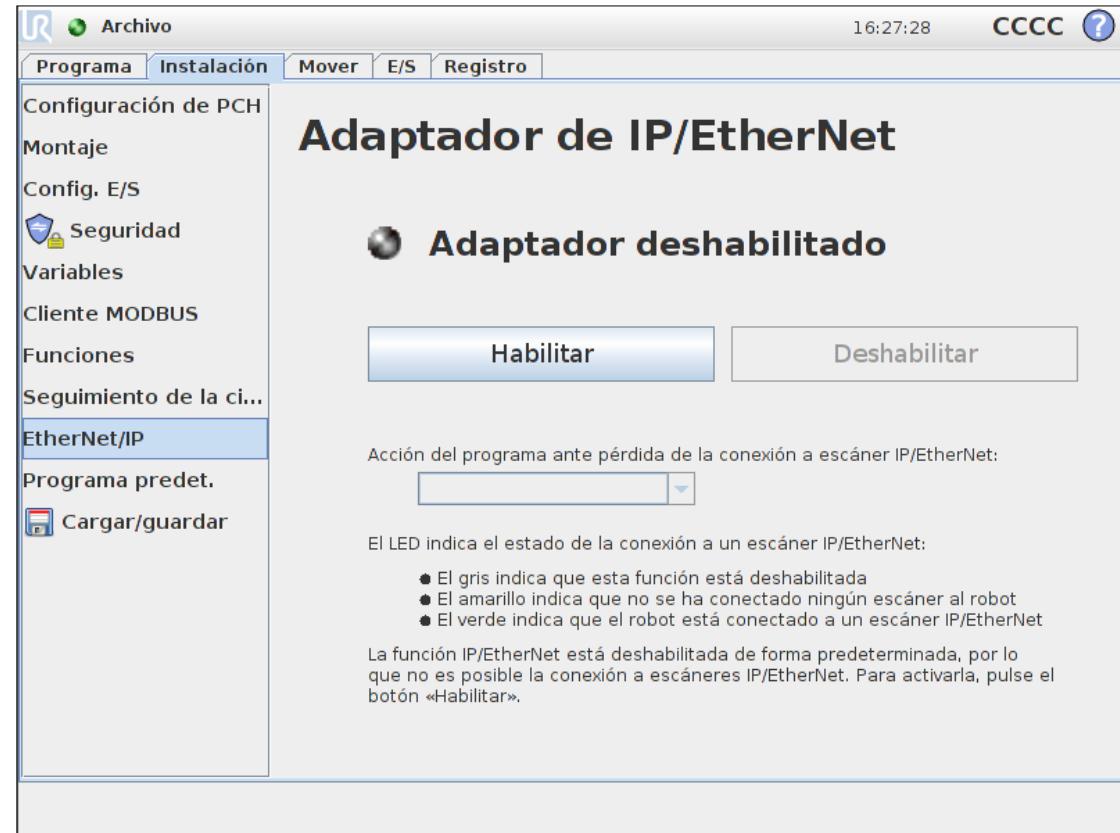
- Propósito
 - Iniciar el trabajo del robot tras el arranque sin acción requerida del operador
- Configuración
 - Elegir programa predeterminado
 - Establecer entrada digital para auto arranque
 - Establecer entrada digital auto inicialización
 - Guardar instalación
 - Reiniciar



- Nota: El robot **siempre** cargará la instalación predeterminada default.installation al arrancar

Ethernet IP

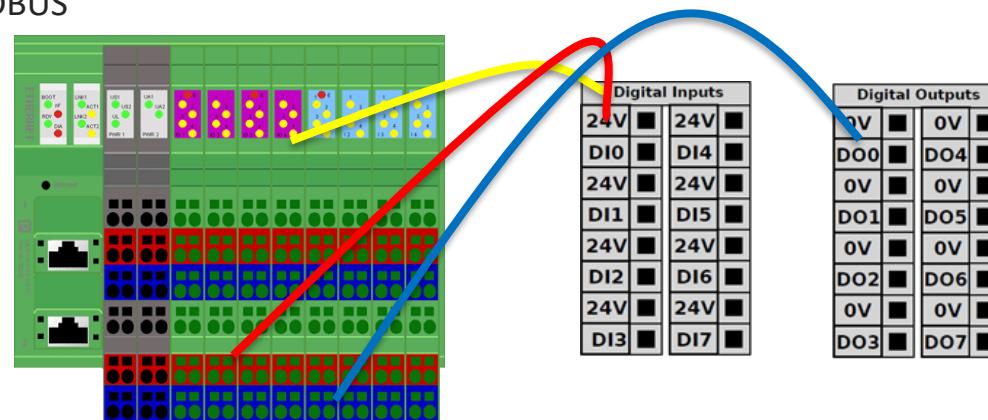
- Ethernet IP
 - Deshabilitado de forma predeterminada
- Disponible desde SW 3.2
 - No incluido en la Formación Principal



Ejercicio práctico 1

- Configuración de la conexión

- Conectar los terminales de alimentación del dispositivo MODBUS a bornes de alimentación de 0V y 24V en el controlador
- Conectar el puerto 1.1 en el bloque IO1 del dispositivo MODBUS a la borne DI0 en el controlador
- Configurar la red del robot y asegurar que ambos dispositivos se encuentren en la misma subred
- Configurar la salida digital del dispositivo MODBUS añadiéndola en la pestaña de instalación
- Activar/Desactivar la salida y comprobar que se activa/desactiva el correspondiente led en el dispositivo MODBUS



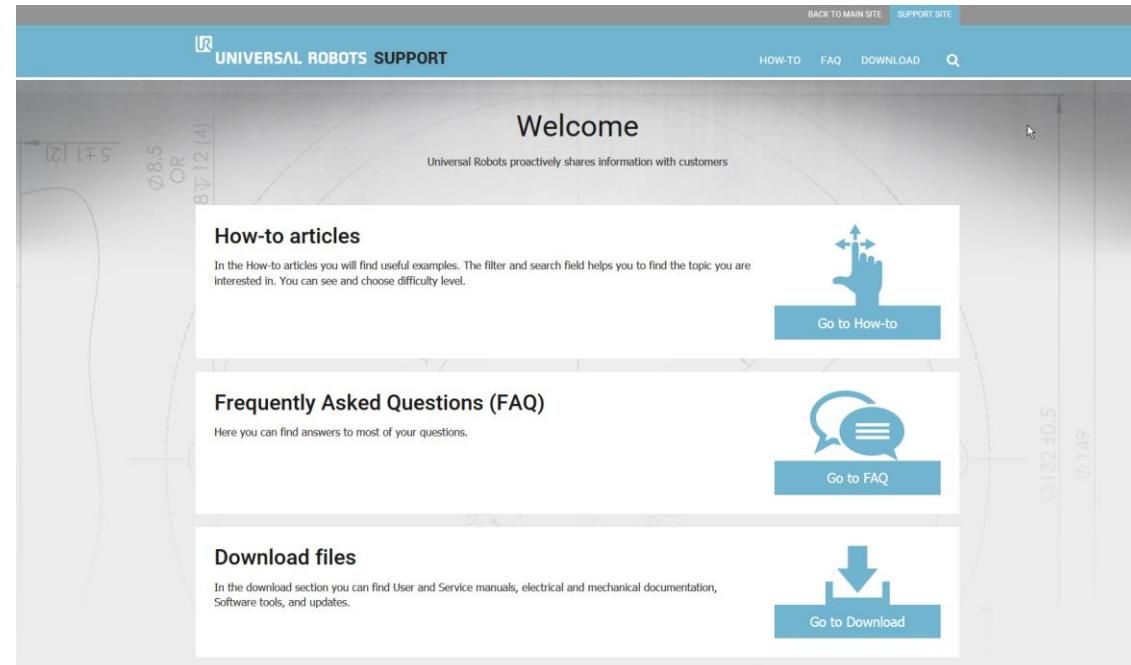
Ejercicio práctico 2

- 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

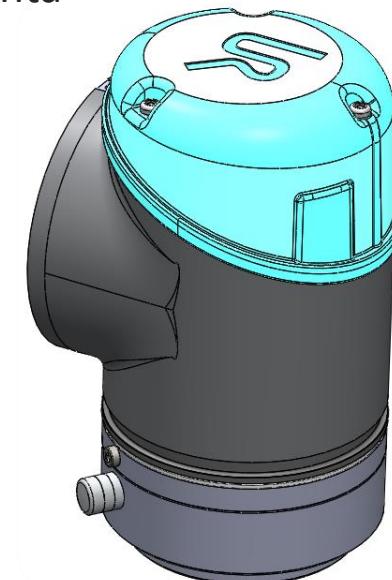
- Descargar última versión desde la página de soporte
- Copiar archivo en la memoria USB
- Insertar USB en TP
- Ir a Config. robot → Actualizar robot
- Pulse el botón "Buscar..." y seleccione la actualización
- Pulse el botón "Actualizar"
- NOTA:
El robot se desconectará tras actualizarse



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

Firmware

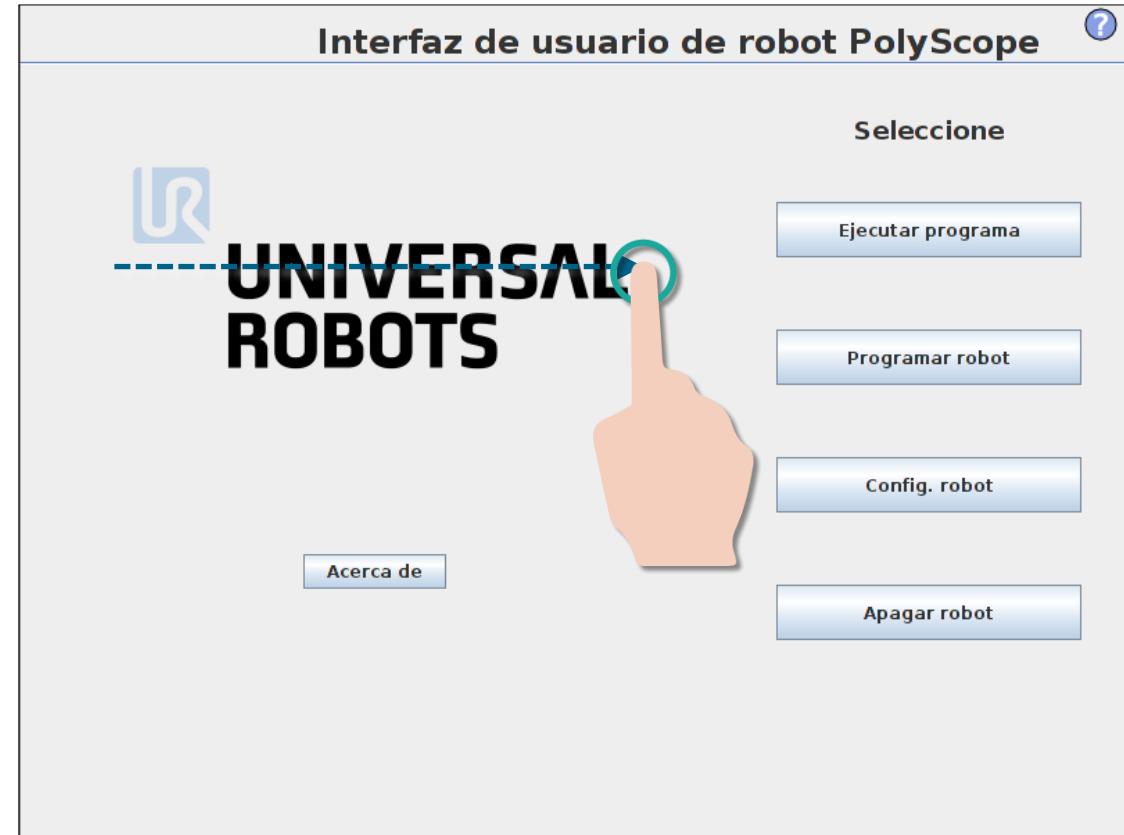
- El Firmware es el Software que se encuentra en cada junta
 - El Firmware controla la junta
 - Durante la actualización del Software, el Firmware se copia automáticamente a la memoria del robot –
No requiere ninguna descarga adicional
- Si es necesario puede ser actualizado
 - En versiones anteriores 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



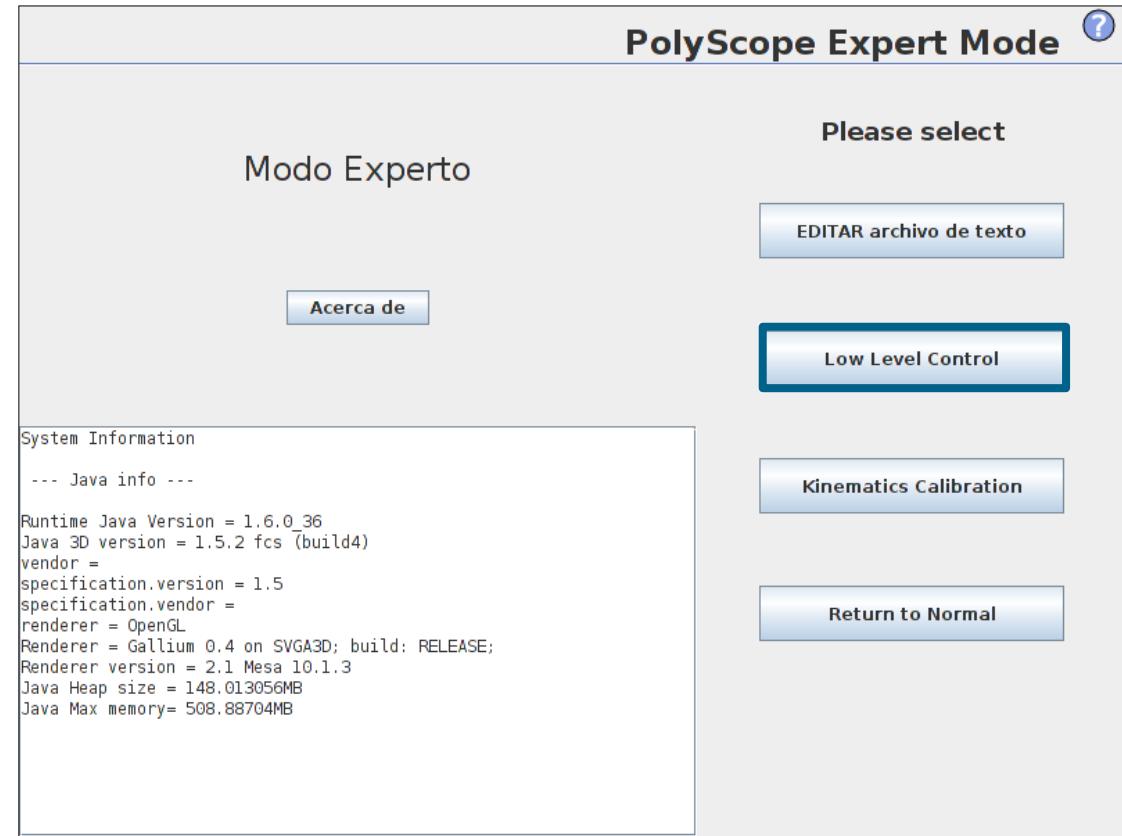
Actualización de Firmware (en versiones anteriores a SW3.1)

- 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

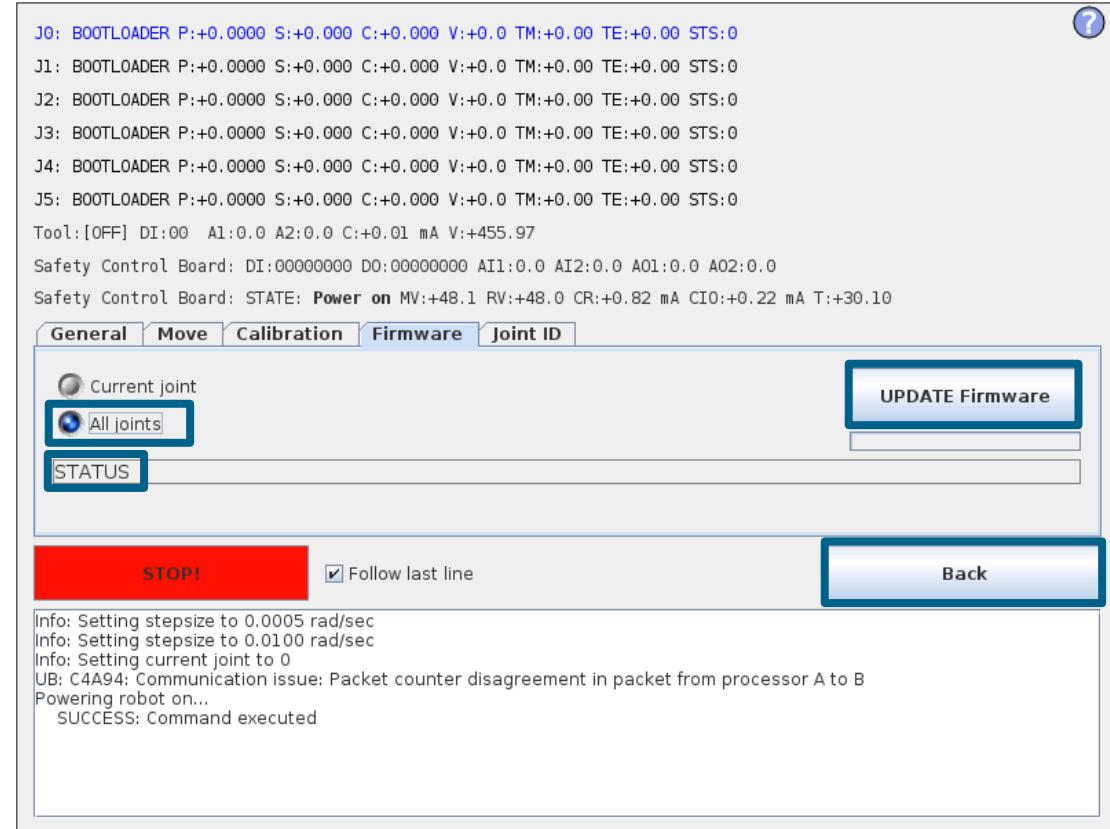
- Pestaña General

- 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:
 - Fichero de instalación (SO Linux)
 - Máquina Virtual (SO Windows)

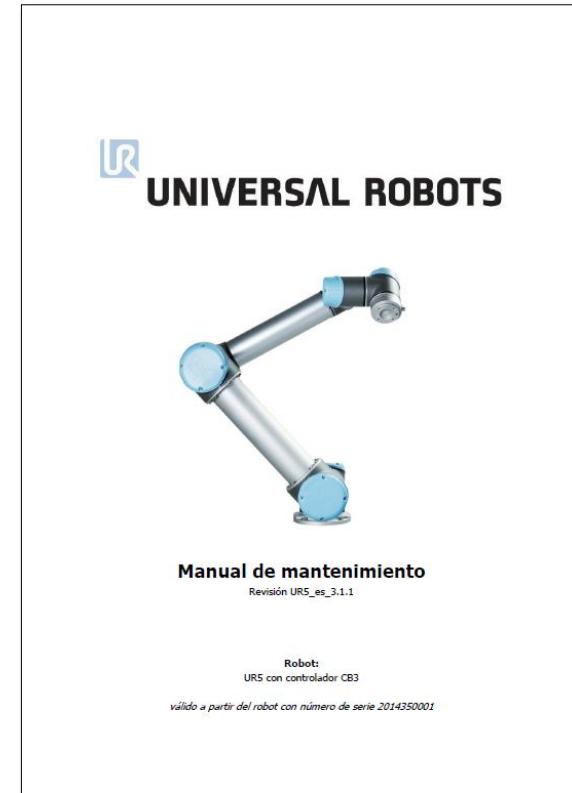


Siguiente...

Mantenimiento

Mantenimiento preventivo

- Contenido manual de mantenimiento
 - 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*

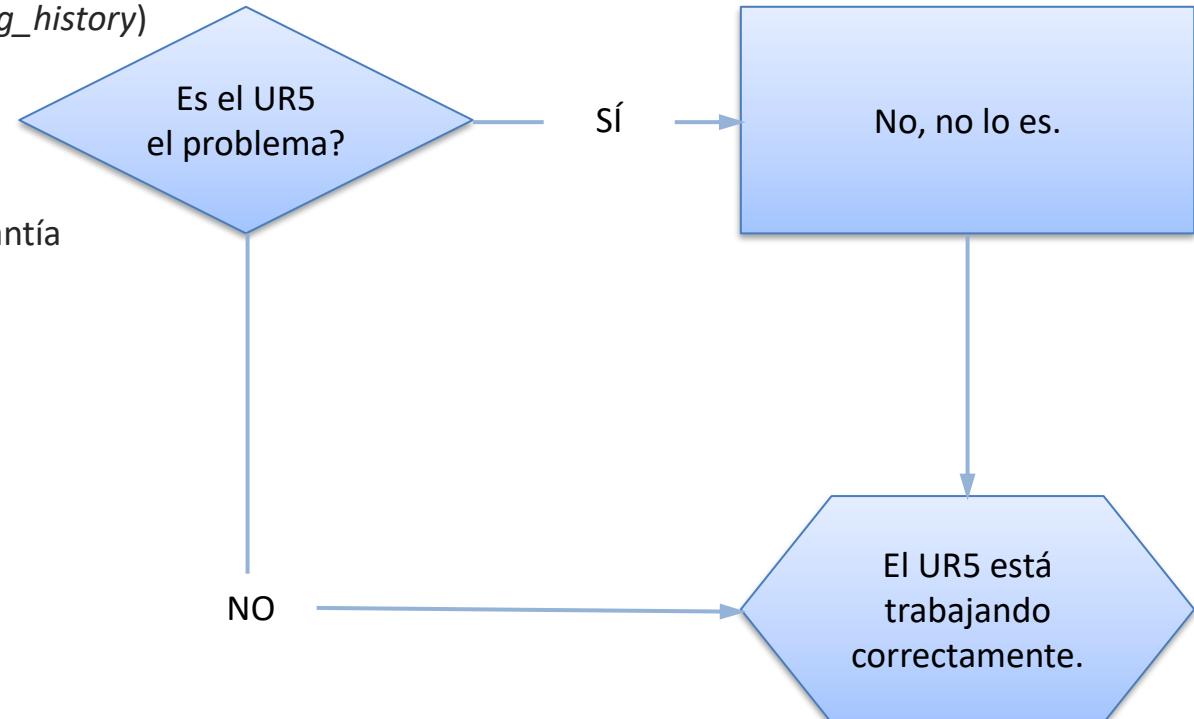


Manual de mantenimiento

- Resolución de problemas

- Registro histórico (*log_history*)
- Sustitución de junta
- Calibrado de junta
- Cambio ID de junta
- Reclamación de Garantía

UR5 Diagrama de flujo de resolución de problemas



Pestaña de Registro

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

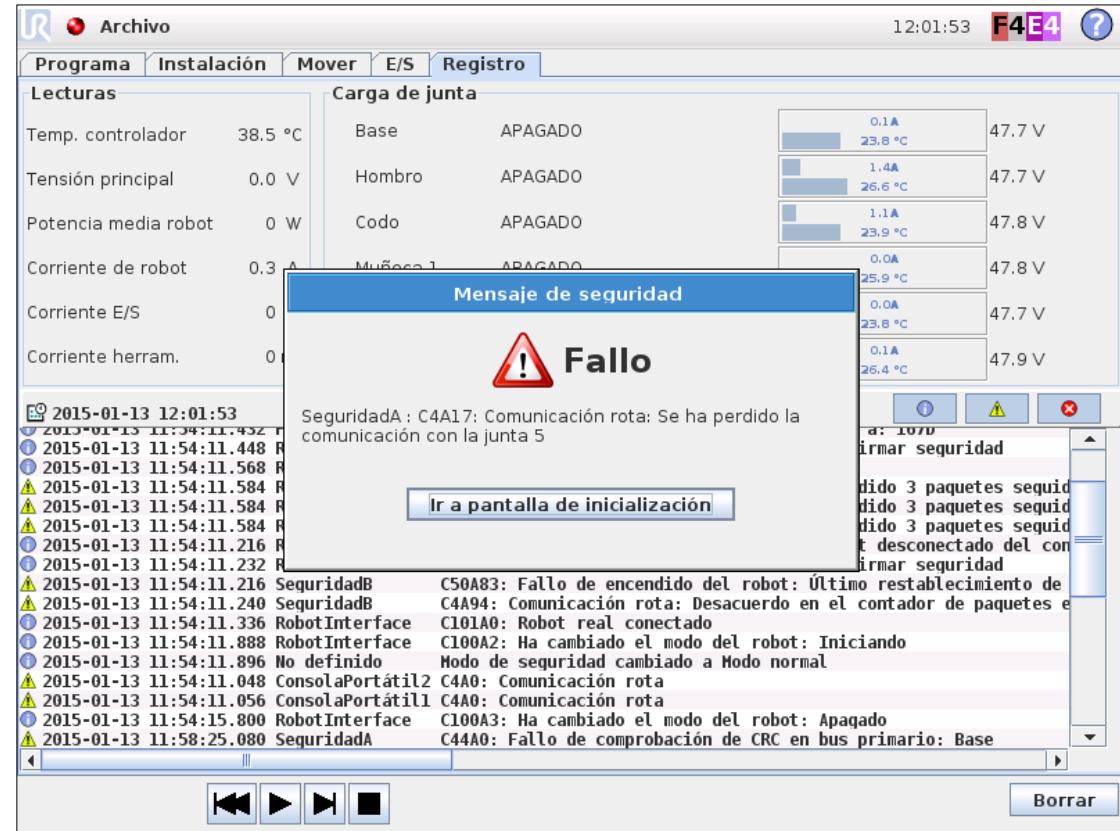
The screenshot shows the Universal Robots software interface with the 'Registro' (Log) tab selected. The top menu bar includes 'Archivo', 'Programa', 'Instalación', 'Mover', 'E/S', and 'Registro'. The status bar shows the time as 10:47:40 and the model as 'CCCC'. The main area is divided into several sections:

- Lecturas (Readings):** Displays sensor values:
 - Temp. controlador: 0.0 °C
 - Tensión principal: 48.0 V
 - Potencia media robot: 298 W
 - Corriente de robot: 6.2 A
 - Corriente E/S: 0 A
 - Corriente herram.: 0 mA
- Carga de junta (Joint Load):** Displays joint status and temperatures:

Joint	Estado	Temperatura (°C)	Voltaje (V)
Base	OK	0.0	0.0
Hombro	OK	40.0	0.0
Codo	OK	40.0	0.0
Muñeca 1	OK	40.0	0.0
Muñeca 2	OK	40.0	0.0
Muñeca 3	OK	40.0	0.0
- Historical Log (2016-02-18 10:47:40):** A list of events with timestamps and descriptions:
 - 2016-02-18 10:23:20.210 RobotInterface C100A1: ha cambiado el modo de
 - 2016-02-18 10:25:26.224 RobotInterface C100A3: Ha cambiado el modo del robot: Apagado
 - 2016-02-18 10:25:26.600 PolyScope Desconectado del controlador
 - 2016-02-18 09:57:15.712 PolyScope Conectado al controlador
 - 2016-02-18 09:57:15.712 RobotInterface URControl 3.2.18744 (Oct 07 2015)
 - 2016-02-18 09:57:15.712 RobotInterface URSafetyA 471: URSafetyB 211
 - 2016-02-18 09:57:15.712 PolyScope Error al activar robot real
 - 2016-02-18 09:57:15.712 PolyScope Error al activar robot real
 - 2016-02-18 09:57:15.712 PolyScope La suma de comprobación de seguridad cambió a: CCCC
 - 2016-02-18 10:27:44.152 PolyScope La suma de comprobación de seguridad cambió a: CCCC
 - 2016-02-18 10:27:44.160 RobotInterface C102A0: Robot real no conectado; robot de simulación
 - 2016-02-18 10:27:44.160 RobotInterface C102A0: Robot real no conectado; robot de simulación
 - 2016-02-18 10:27:44.168 RobotInterface C100A3: Ha cambiado el modo del robot: Apagado
 - 2016-02-18 10:27:44.184 RobotInterface C100A4: Ha cambiado el modo del robot: Encendido
 - 2016-02-18 10:27:44.192 RobotInterface C100A5: Ha cambiado el modo del robot: Inactivo
 - 2016-02-18 10:27:44.200 RobotInterface C100A7: Ha cambiado el modo del robot: Ok

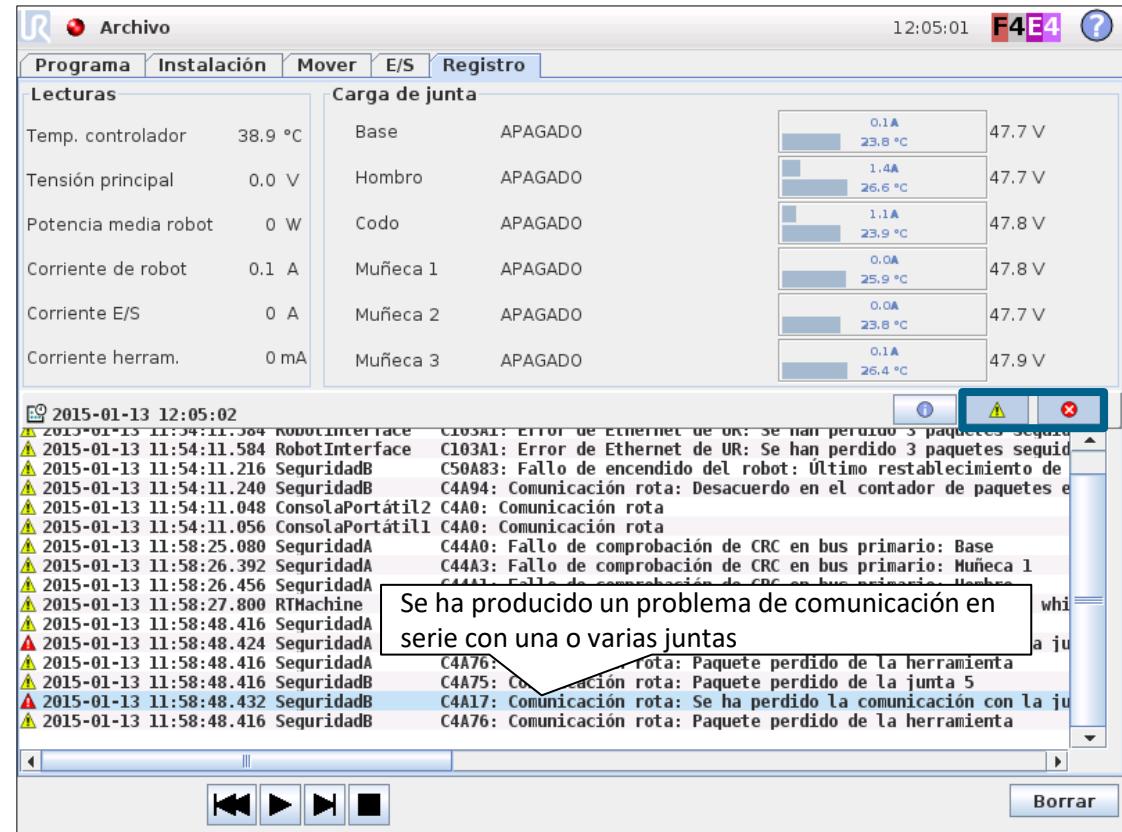
Pestaña de Registro

- 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



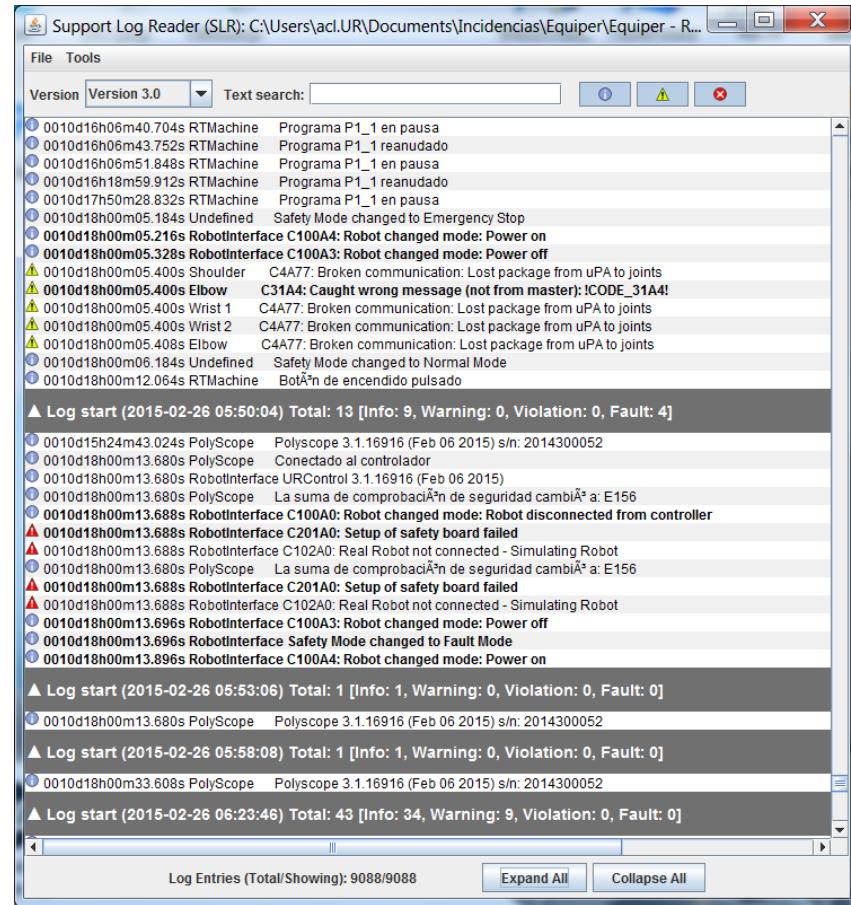
Pestaña de Registro

- 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

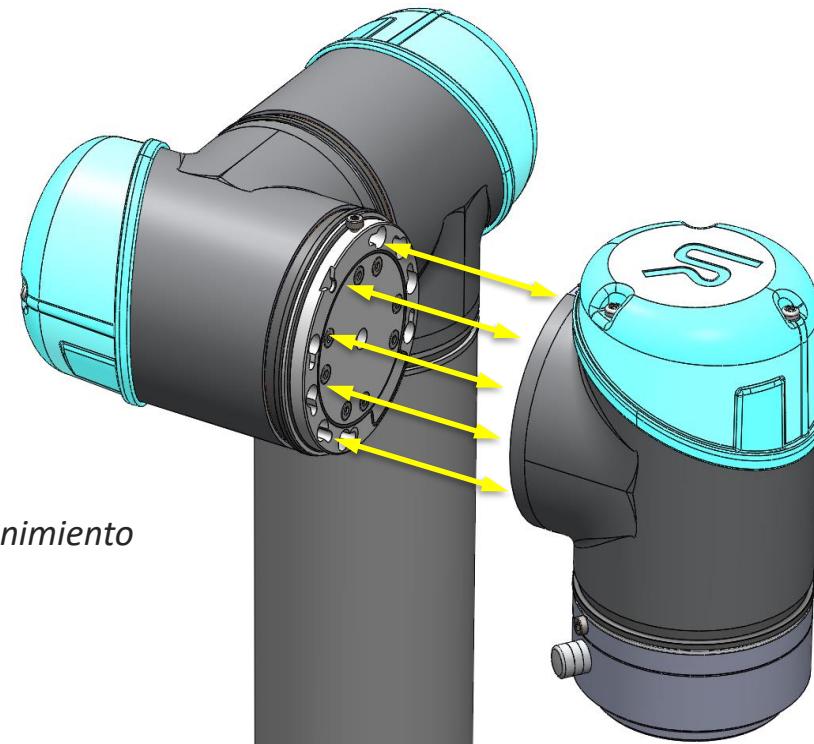


Siguiente...

Sustitución de juntas

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 junta 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



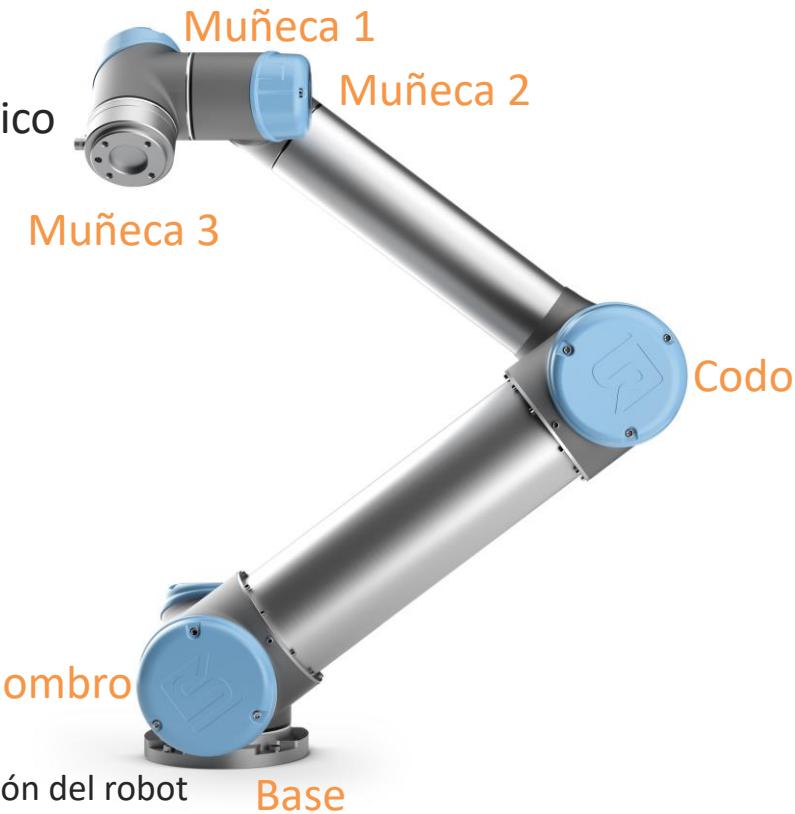
Siguiente...

Cambiar ID de juntas

Cambio del ID en las juntas

- Cada junta 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

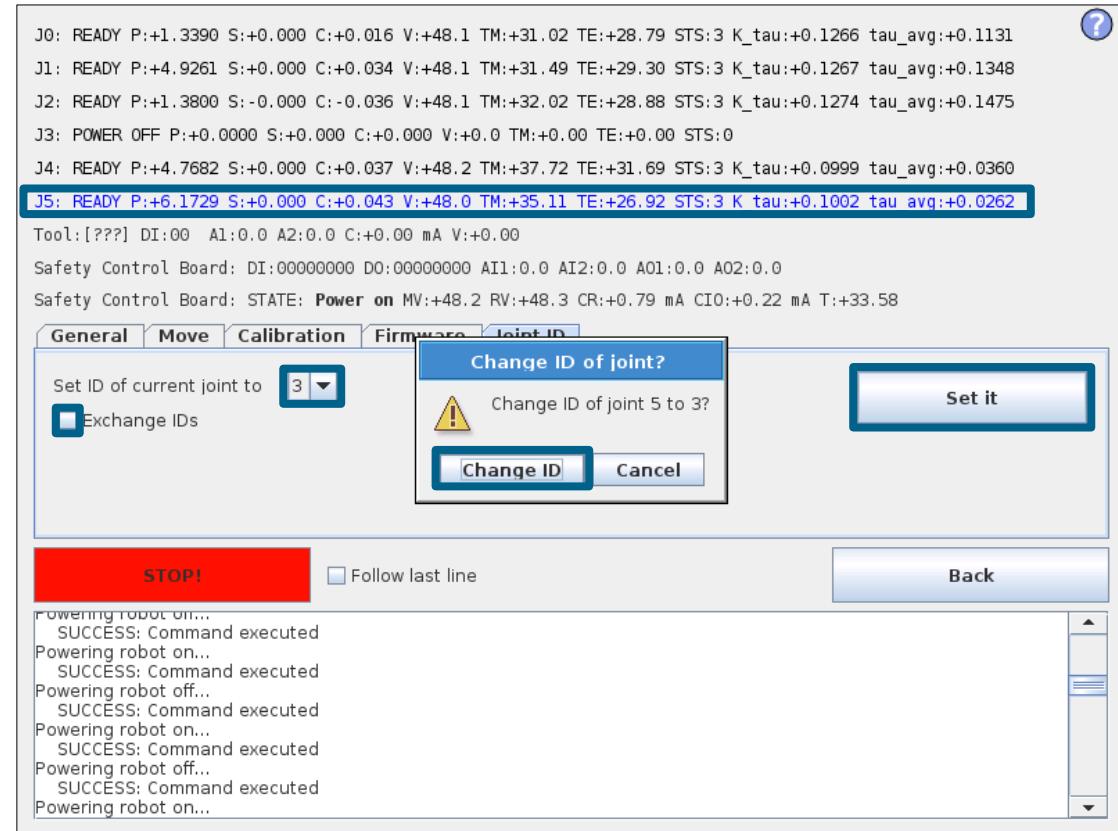
Cambio del ID en las juntas

- 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

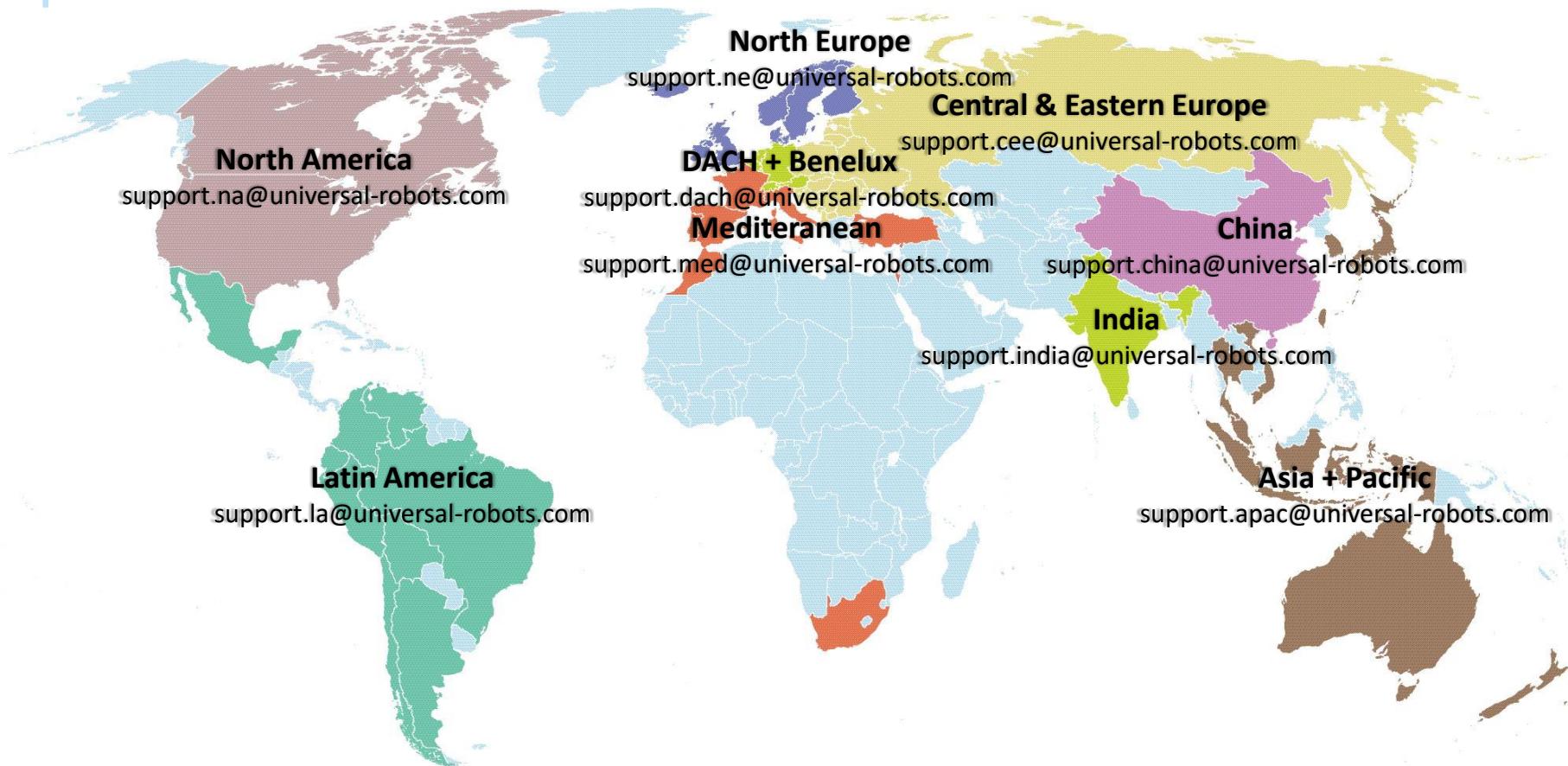
- Contactar por email con Soporte Regional de UR

Mediterráneo

support.med@universal-robots.com

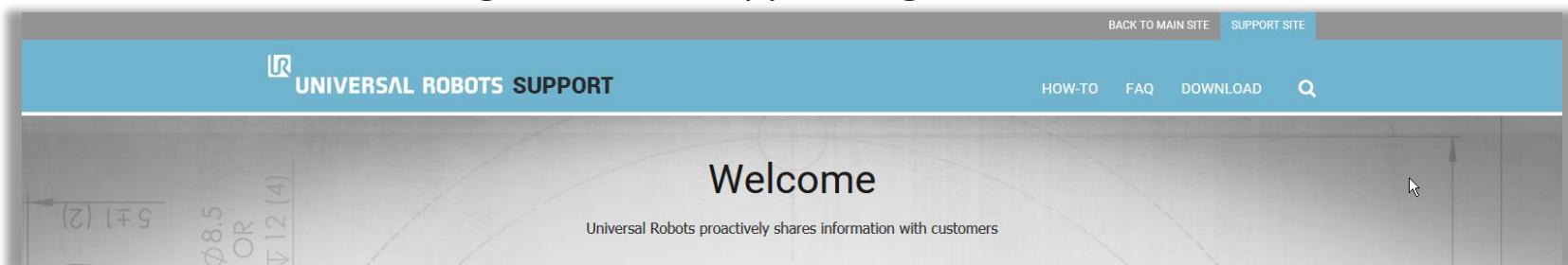
- Información requerida por UR
 - Número de serie del robot
 - Versión software
 - Descripción detallada del error
 - Adjuntar fichero de registro *log_history*
 - Tras recibir esta información, desde la oficina de Soporte Regional, se abrirá un RMA y se enviará el recambio necesario

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
 - Emplazamiento de elementos de mando y paros de emergencia
 - Diseño de la herramienta del robot
 - Movimiento y velocidades del robot
 - Posición del operario

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?
 - Constituye una guía para la operación colaborativa con robots dónde un sistema robótico y las personas comparten el mismo espacio de trabajo
- 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



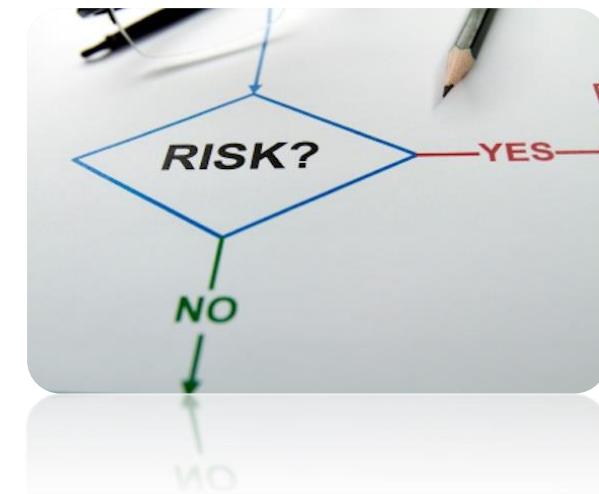
El integrador **DEBE** realizar una evaluación de riesgos

- NO OBLIGATORIO

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

- En caso de accidente:

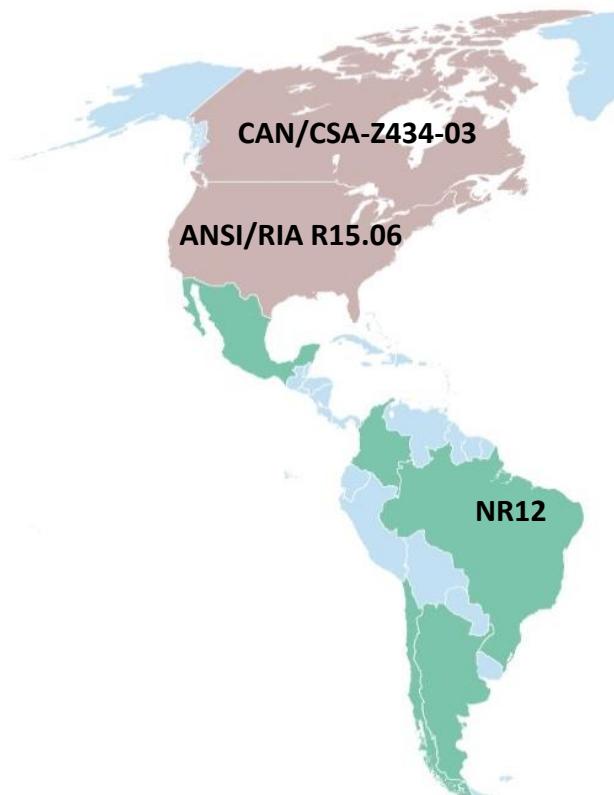
- Si el sistema cumple con los estándares
- Si el sistema **no** cumple con los estándares



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

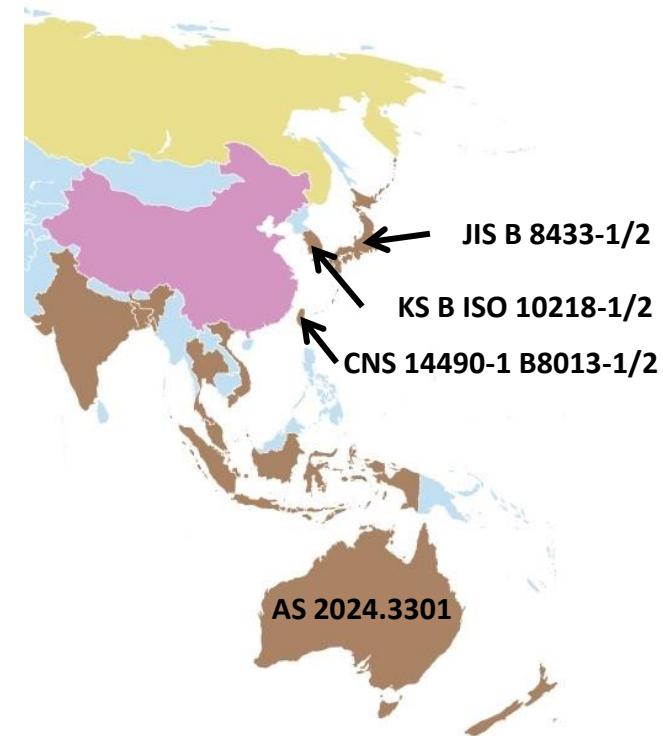
Diferencias regionales

- Canadá
 - CAN/CSA-Z434-03: 2013
 - Armonizado con estándares internacionales ISO
 - Consiste en la ISO 10218-1 e ISO 10218-2 con diferencias regionales
- EEUU
 - 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

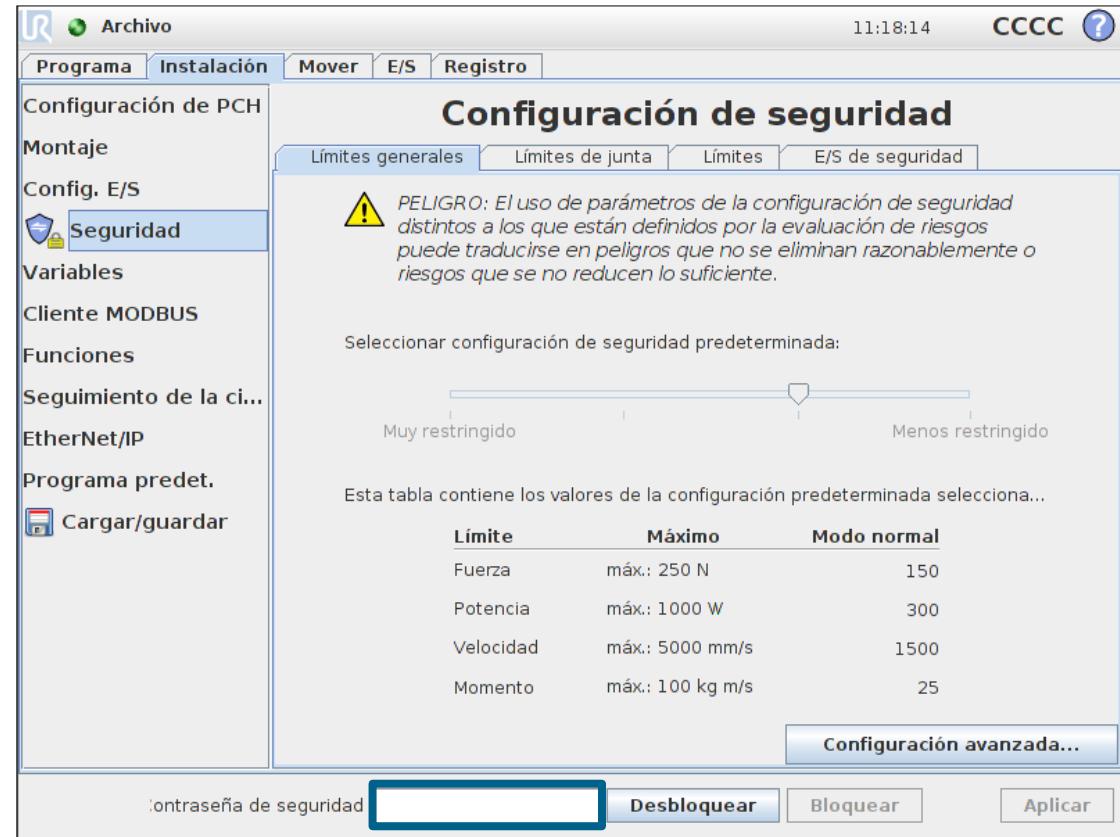
Características de seguridad

- 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



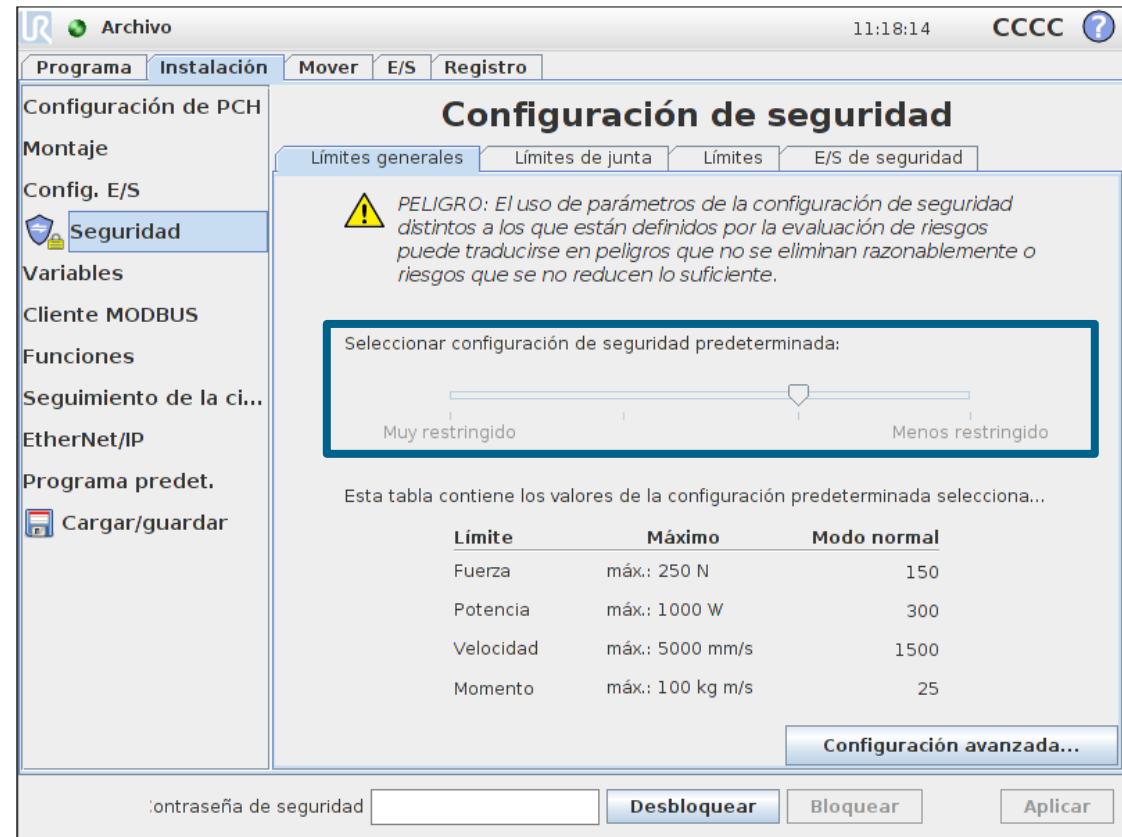
Contraseña de seguridad

- La configuración de seguridad está protegida por contraseña
 - Bloquear
 - Protege la configuración de seguridad
 - Desbloquear
 - Habilita la modificación de la configuración de seguridad
- Cambiar contraseña
 - Ir a Config. robot
 - Ir a Fijar contraseña
 - Cambiar contraseña de seguridad
 - Introducir contraseña
 - Aplicar cambios



Configuración básica

- Niveles de seguridad
 - Muy restringido
 - Restringido
 - Predeterminado
 - Menos restringido
- Estado de configuración
 - Sincronizada
 - Alterada
 - Configuración inválida
- Guardar configuración
 - Tocar "Aplicar"
 - Confirmar ajustes



Configuración básica

- Valores por defecto

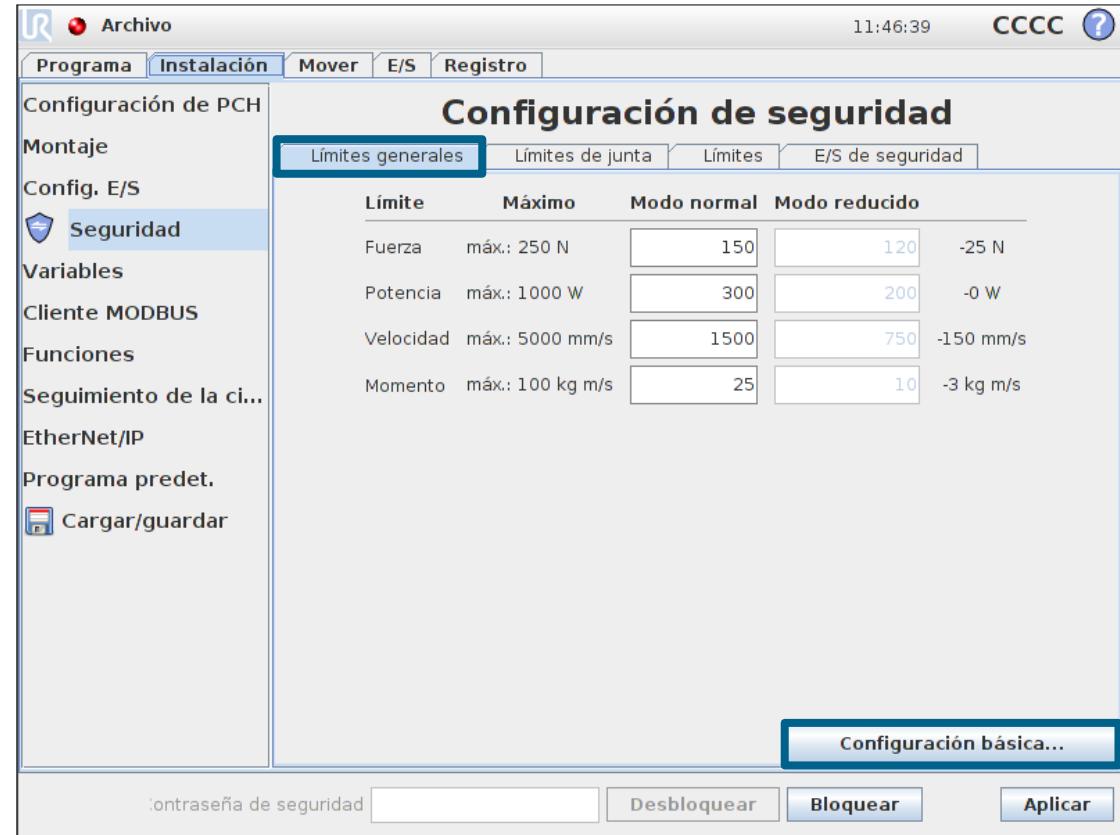
Modo configurado	Muy restringido	Restringido	Predeterminado	Menos restringido
Fuerza (N)	100*	120	150	250
Potencia (W)	80	200	300	1000
Velocidad (mm/s)	250	750	1500	5000
Momento (kg·m/s)	5	10	25	100

* 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 adaptados
 - Fuerza
 - Potencia
 - Velocidad
 - Momento
- Modos
 - Modo normal
 - Modo reducido
 - Requiere uso de entradas o límites de seguridad

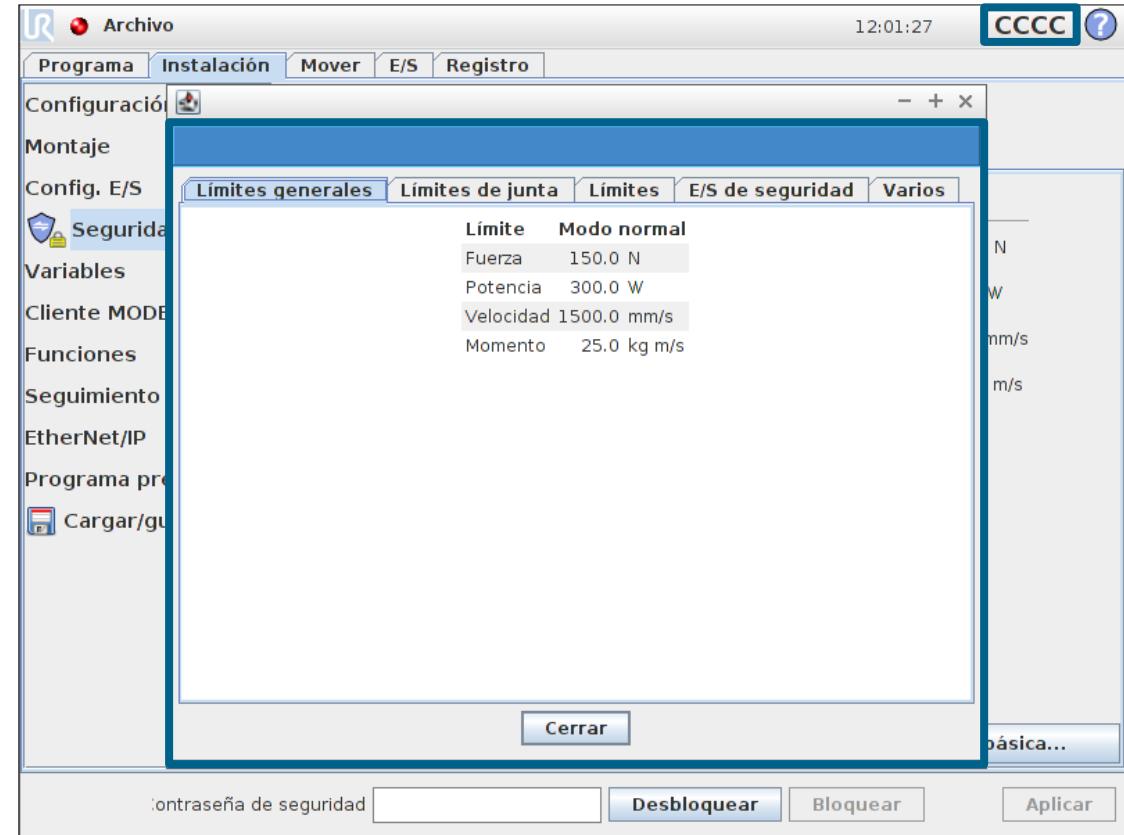


Modos de seguridad

- Modo normal
 - Modo de seguridad activo de forma predeterminada
- Modo reducido
 - Activo cuando el PCH del robot se posiciona más allá de un límite seguridad definido como *Modo Reducido con activador*
 - Activo cuando se usan las entradas configurables definidas como *Modo reducido*
- *En caso de violación de la seguridad:*
 - Modo recuperación
 - 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 junta y PCH están deshabilitadas en este modo

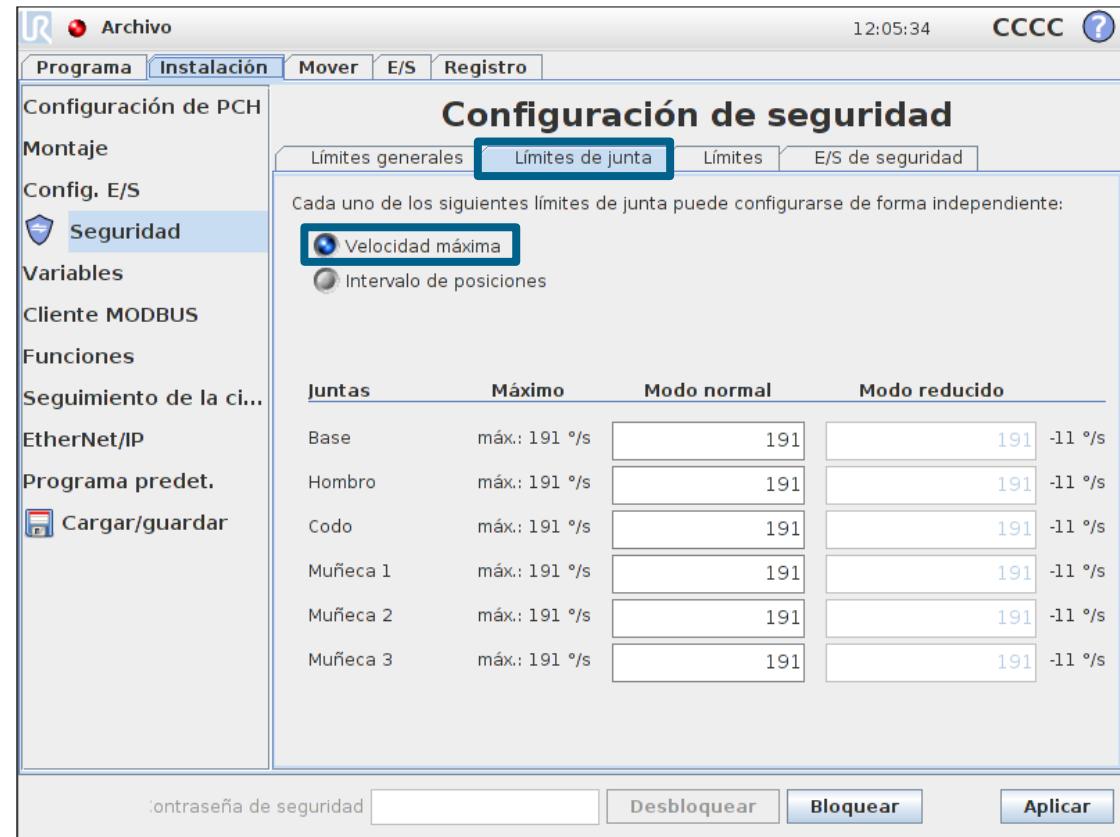
Suma de comprobación de seguridad

- Comprobación
 - 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



Límites de junta

- **Velocidad máxima**
 - Establece velocidad máxima para cada junta
- **Modo reducido**
 - Modo normal
 - Modo reducido
 - Requiere uso de entradas o límites de seguridad



Límites de junta

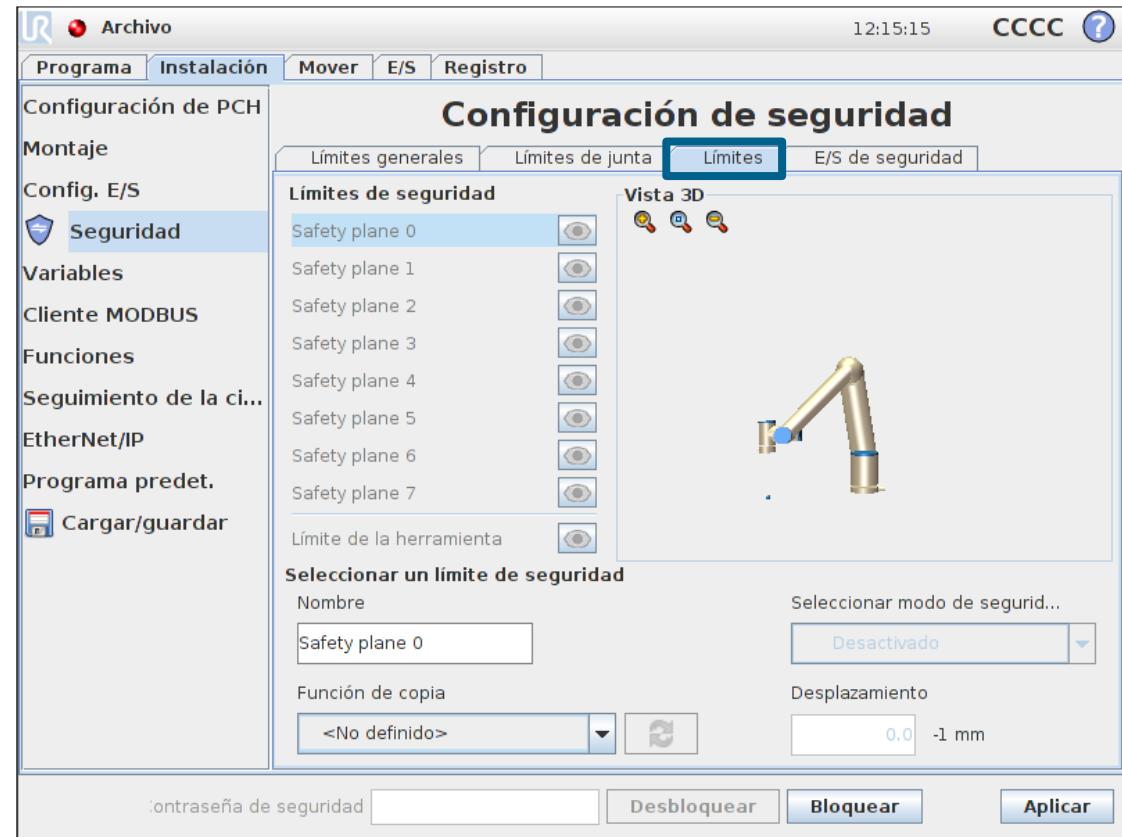
- Intervalo de posiciones
 - Establece rango mínimo y máximo para cada junta
- Modos
 - Modo normal
 - Modo reducido
 - Requiere uso de entradas o límites de seguridad

The screenshot shows the 'Configuración de seguridad' (Safety Configuration) screen in the Universal Robots software. The 'Límites de junta' tab is active. The interface includes a sidebar with various configuration options like 'Programa', 'Instalación', 'Mover', 'E/S', and 'Registro'. The main area displays a table for setting joint ranges. The table has columns for Juntas (Joints), Intervalo (Range), Modo normal (Normal Mode), and Modo reducido (Reduced Mode). For each joint, the range is listed as -363 - 363 °. In 'Modo normal', the minimum and maximum values are both 363. In 'Modo reducido', the minimum value is -363 and the maximum value is 363. The table also includes a column for 'Velocidad máxima' (Maximum speed) with values +3 ° / -3 °.

Juntas	Intervalo	Modo normal	Modo reducido			
Base	-363 — 363 °	Mínimo 363	Máximo 363	Mínimo -363	Máximo 363	+3 ° / -3 °
Hombro	-363 — 363 °	Mínimo -363	Máximo 363	Mínimo -363	Máximo 363	+3 ° / -3 °
Codo	-363 — 363 °	Mínimo -363	Máximo 363	Mínimo -363	Máximo 363	+3 ° / -3 °
Muñeca 1	-363 — 363 °	Mínimo -363	Máximo 363	Mínimo -363	Máximo 363	+3 ° / -3 °
Muñeca 2	-363 — 363 °	Mínimo -363	Máximo 363	Mínimo -363	Máximo 363	+3 ° / -3 °
Muñeca 3	-363 — 363 °	Mínimo -363	Máximo 363	Mínimo -363	Máximo 363	+3 ° / -3 °

Límites

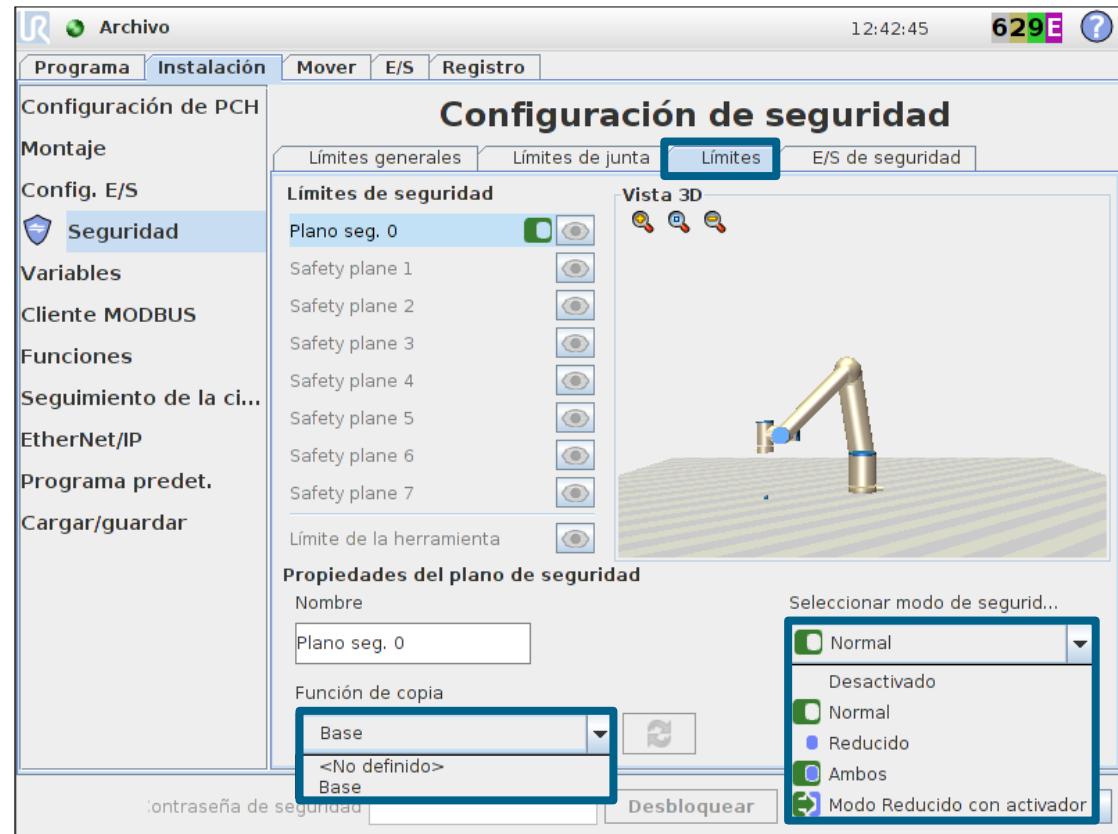
- Plano de seguridad
 - Área de trabajo restringida
 - Hasta 8 planos distintos
 - Activos en movimiento manual y en ejecución de programas
 - Los planos pueden activar *Modo Reducido* cuando el PCH los atraviesa



- IMPORTANTE: Los límites de seguridad limitan sólo al PCH, no al brazo del robot

Límites

- Establecer plano
 - Seleccionar Función de copia
 - Define qué función usar como límite
 - 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
 - Activa los cambios en la configuración



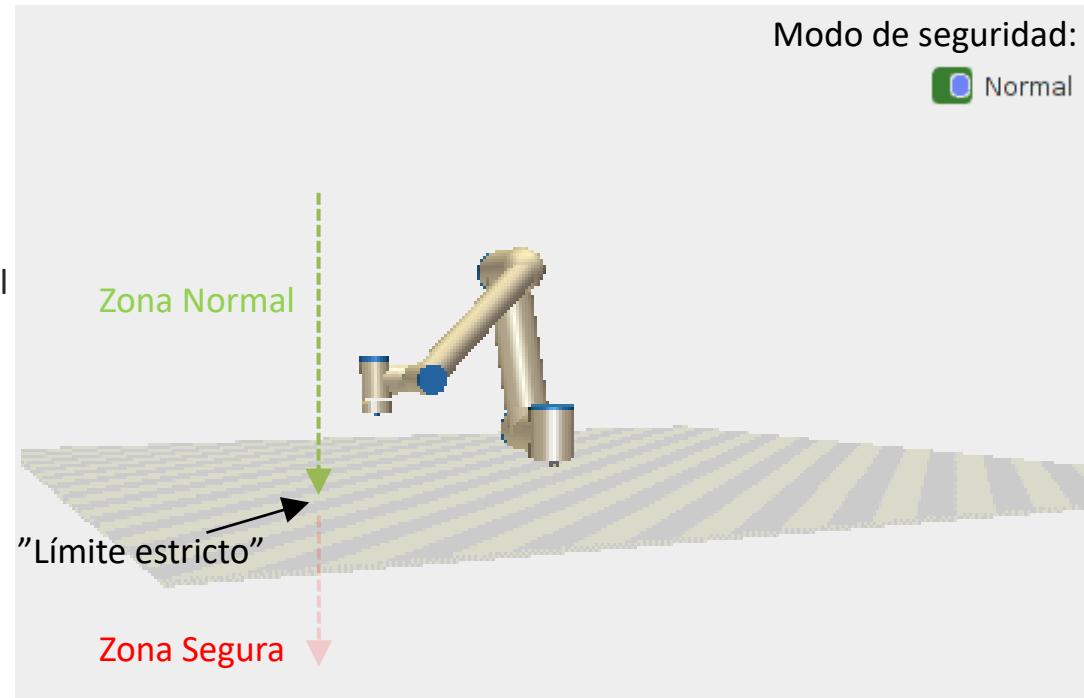
Modos de seguridad

- Comportamiento de los límites según el modo de seguridad

Modo de seguridad	Comportamiento
Desactivad	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	actúa como límite estricto en ambos modos
 Modo Reducido con activador	pasa el robot a modo reducido cuando el PCH atraviesa el plano

Comportamiento de los límites

- Prueba
 - Función de copia = *Base*
 - Modo de seguridad = *Normal*
 - Aplicar configuración
 - Prueba en movimiento manual
 - 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

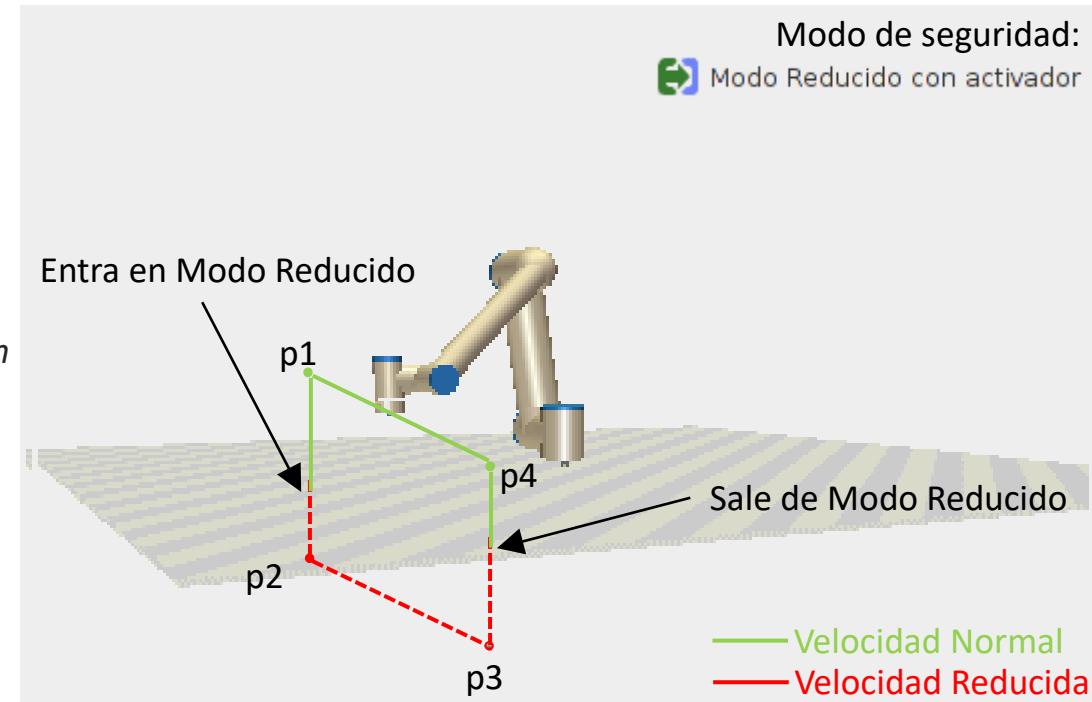
Modo Reducido con activador

- Prueba
 - Cambiar modo de seguridad a: Modo Reducido con activador
 - Seleccionar velocidad máxima en Modo Reducido a: 350 mm/s
 - Aplicar configuración
 - Guardar como *safety.installation*

Programa de robot

MoveL

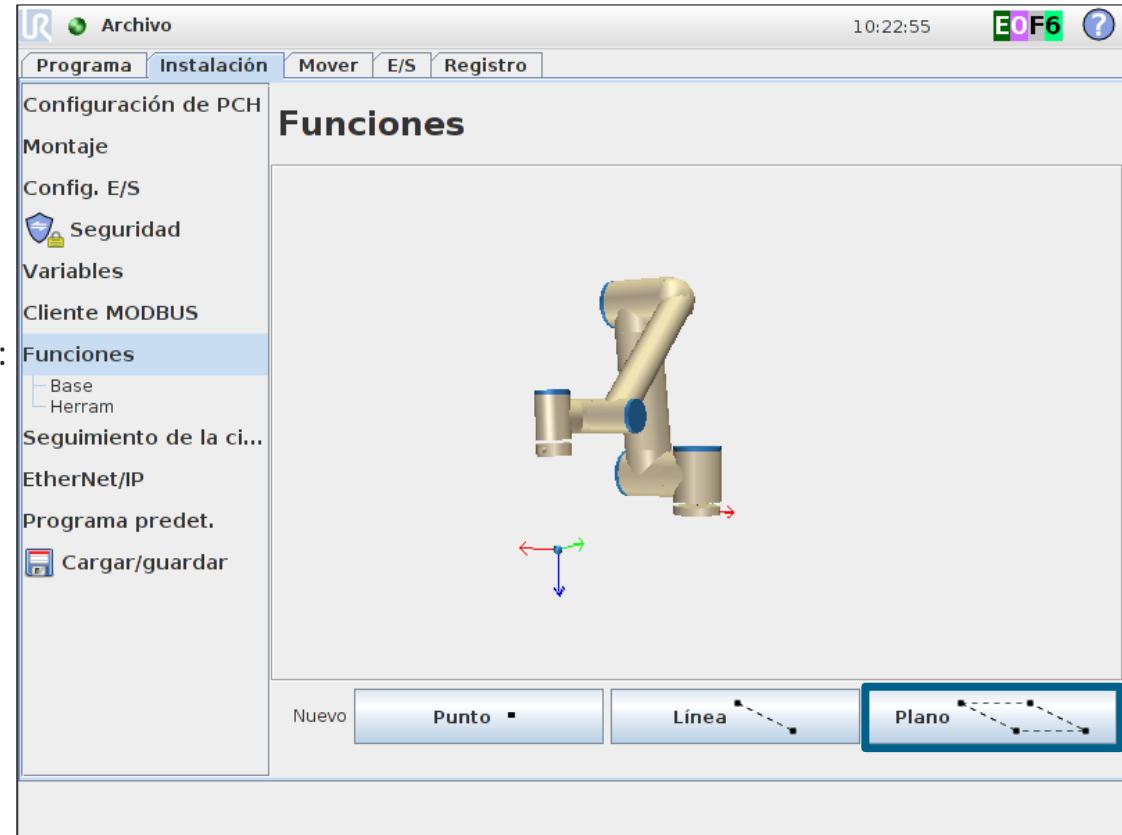
```
Punto_de_paso_1  
Punto_de_paso_2  
Punto_de_paso_3  
Punto_de_paso_4
```



- Guardar programa de ejemplo como *trigger_reduced_mode.urp*

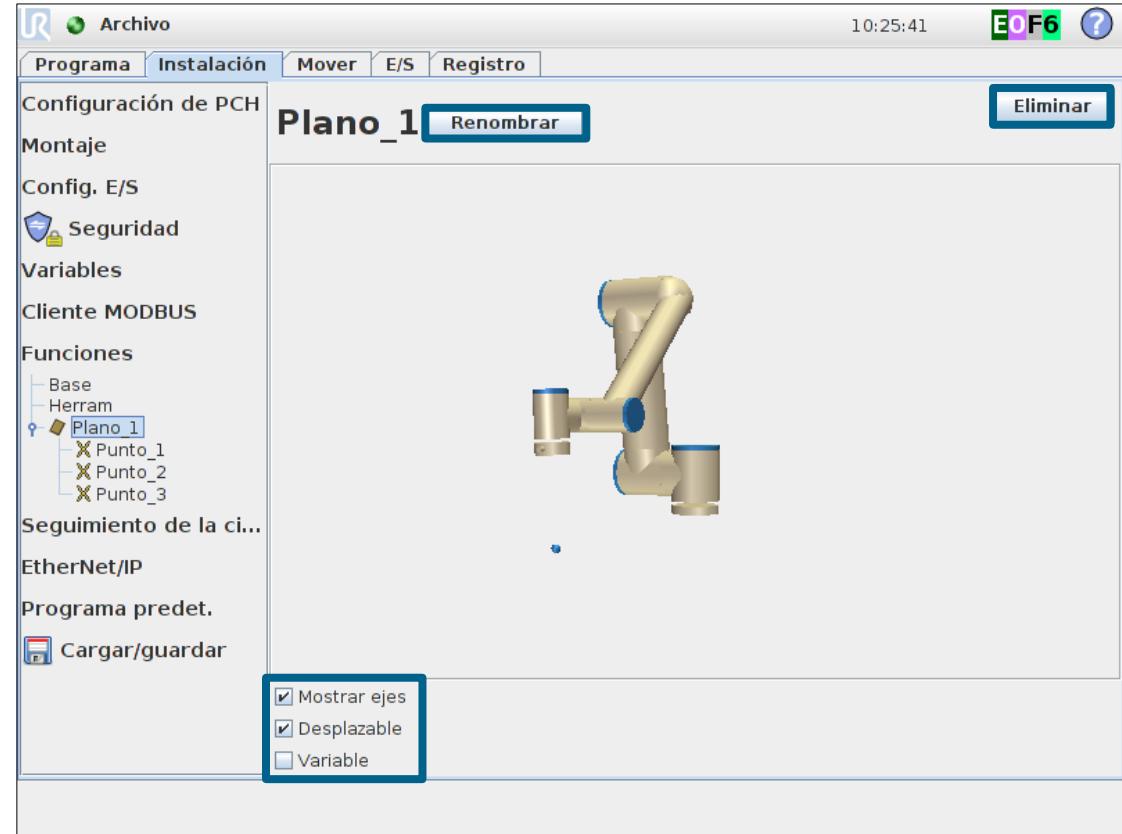
Plano de seguridad definido por el usuario

- **Funciones**
 - En PolyScope un Plano se define como una Función
 - Se pueden establecer múltiples Funciones
 - Establecer Funciones como:
 - Punto
 - Línea
 - Plano
- **Añadir Función**
 - Seleccionar Plano



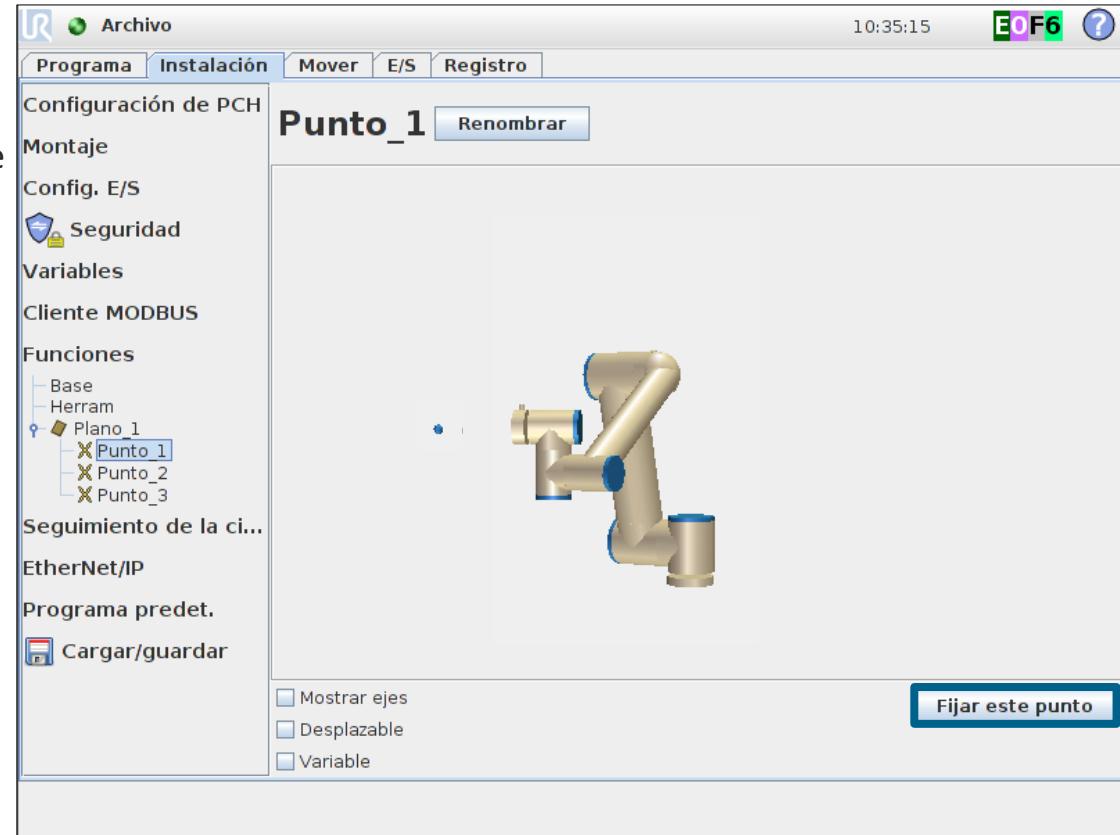
Plano de seguridad definido por el usuario

- Opciones
 - Renombrar Función
 - Eliminar Función
- Parámetros
 - Mostrar ejes
 - Desplazable
 - Variable



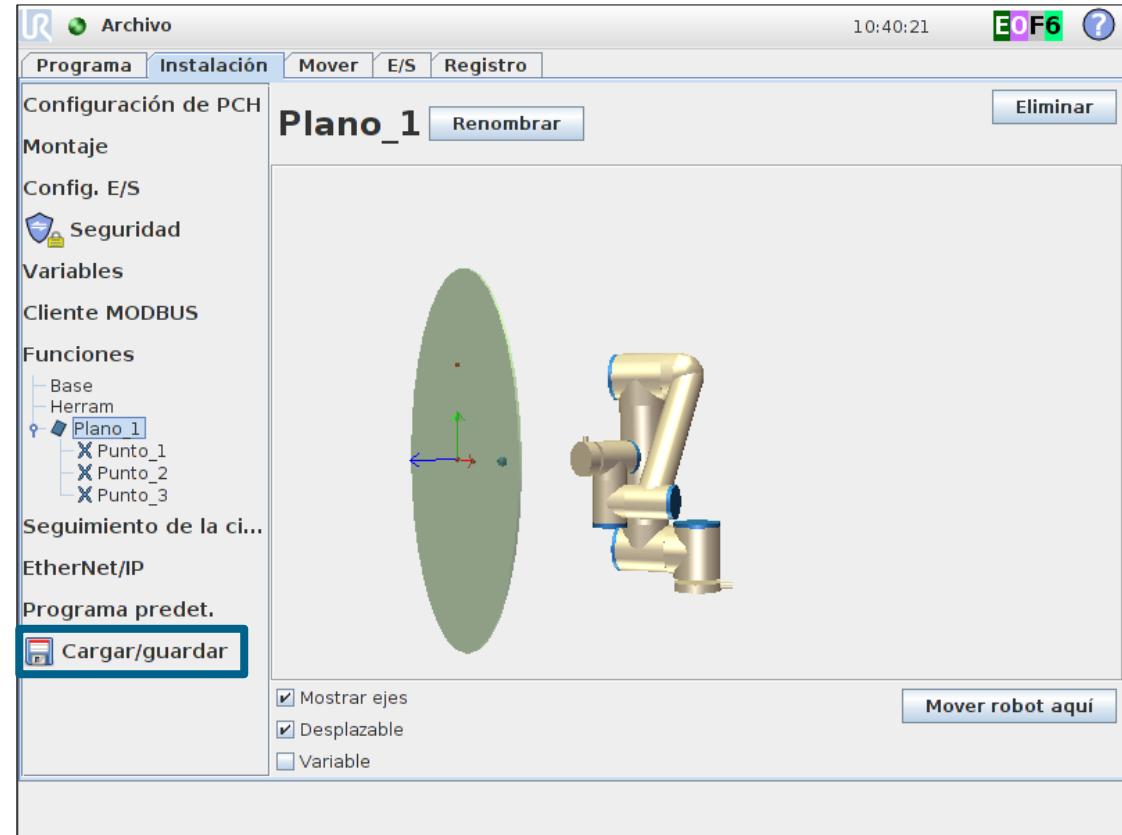
Plano de seguridad definido por el usuario

- Establecer Plano
 - El plano se define mediante tres puntos fijos
- Aprender plano vertical
 - Punto_1 = Origen
 - Punto_2 = Dirección-Y
 - Punto_3 = Dirección-X



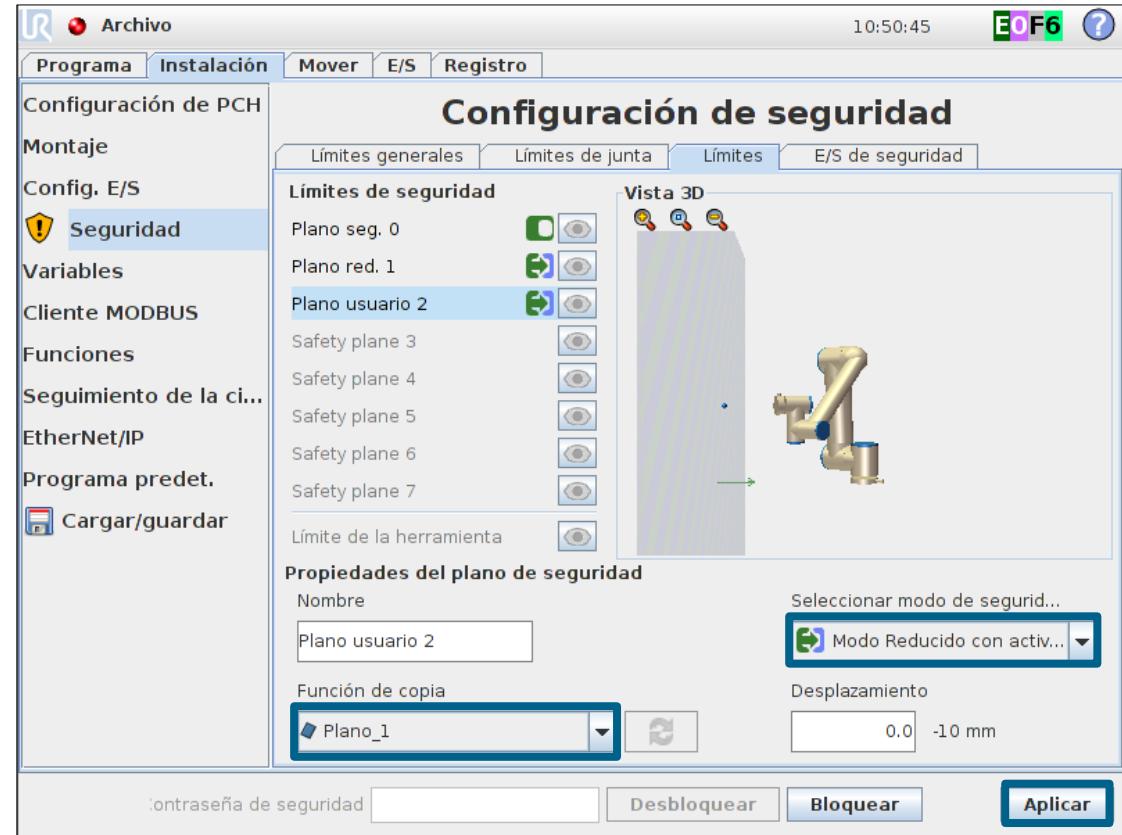
Plano de seguridad definido por el usuario

- Representación gráfica del Plano aprendido
 - Guardar instalación



Plano de seguridad definido por el usuario

- Establecer nuevo plano de seguridad:
 - Elegir Función de copia: *Plano_1*
 - Seleccionador modo de seguridad: *Modo Reducido con activador*
 - Aplicar configuración



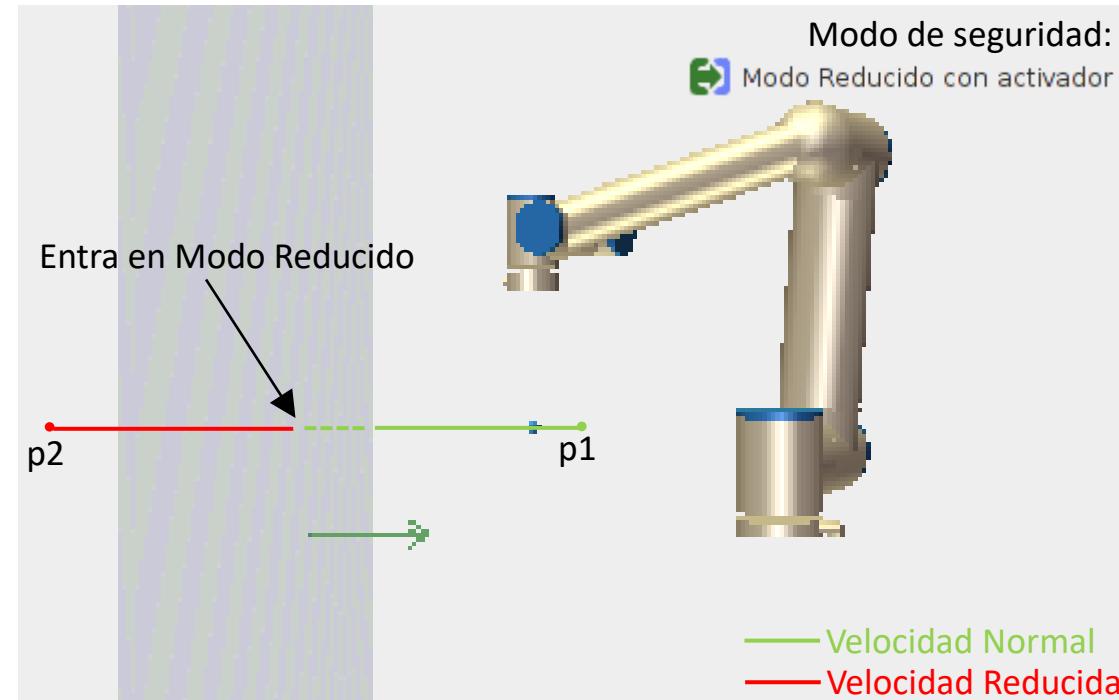
- Guardar instalación de ejemplo como *safety.installation*

Plano de seguridad definido por el usuario

- Prueba
 - Movimiento manual
 - Mover el robot desde la zona Normal hacia la zona Segura
 - Ejecución de programa
 - Crear un programa de ejemplo y verificar que cambie la velocidad del PCH

Programa de robot

```
MoveL  
Punto_de_paso_1  
Punto_de_paso_2
```



- Guardar programa de ejemplo como trigger_reduced_mode_feature.urp

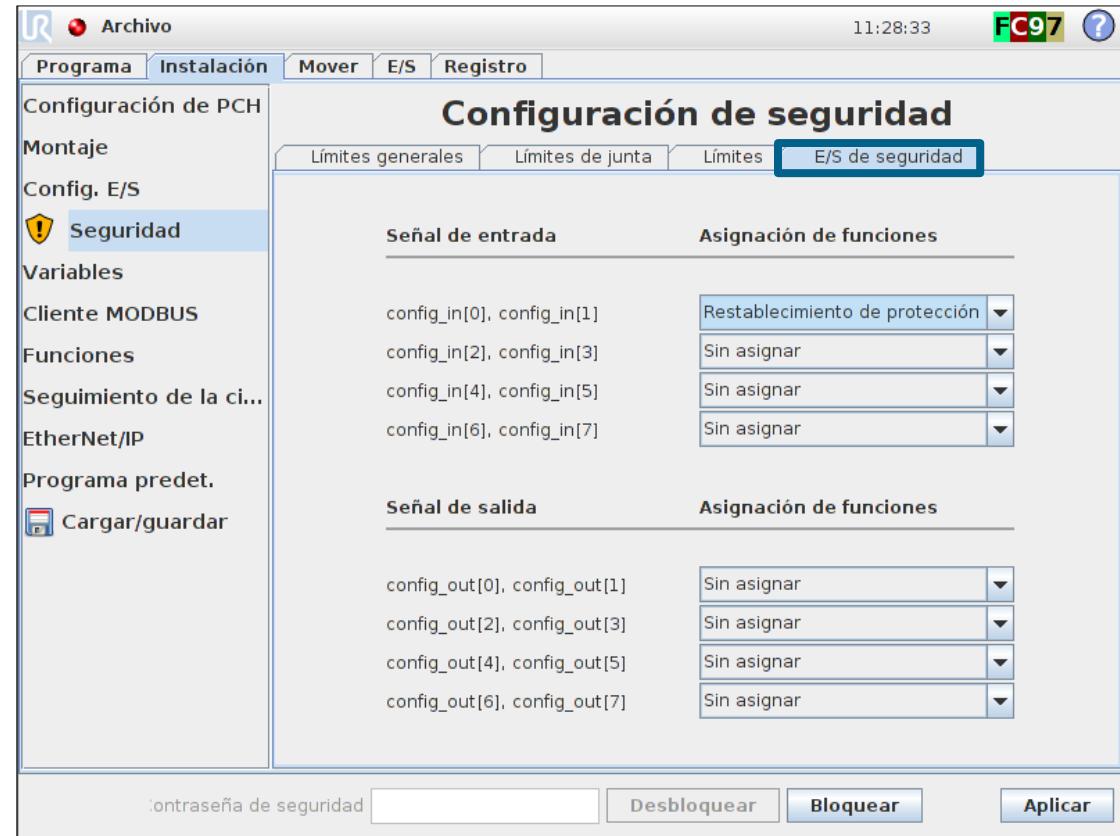
Límite de la herramienta

- Limita la variación angular del PCH
 - Establece la variación angular máxima del PCH respecto de la función elegida
- Prueba
 - Elegir Función de copia: *Base*
 - Establecer Desviación: 20°
 - Seleccionar modo de seguridad a: *Ambos*
 - Aplicar
 - Probar en movimiento manual



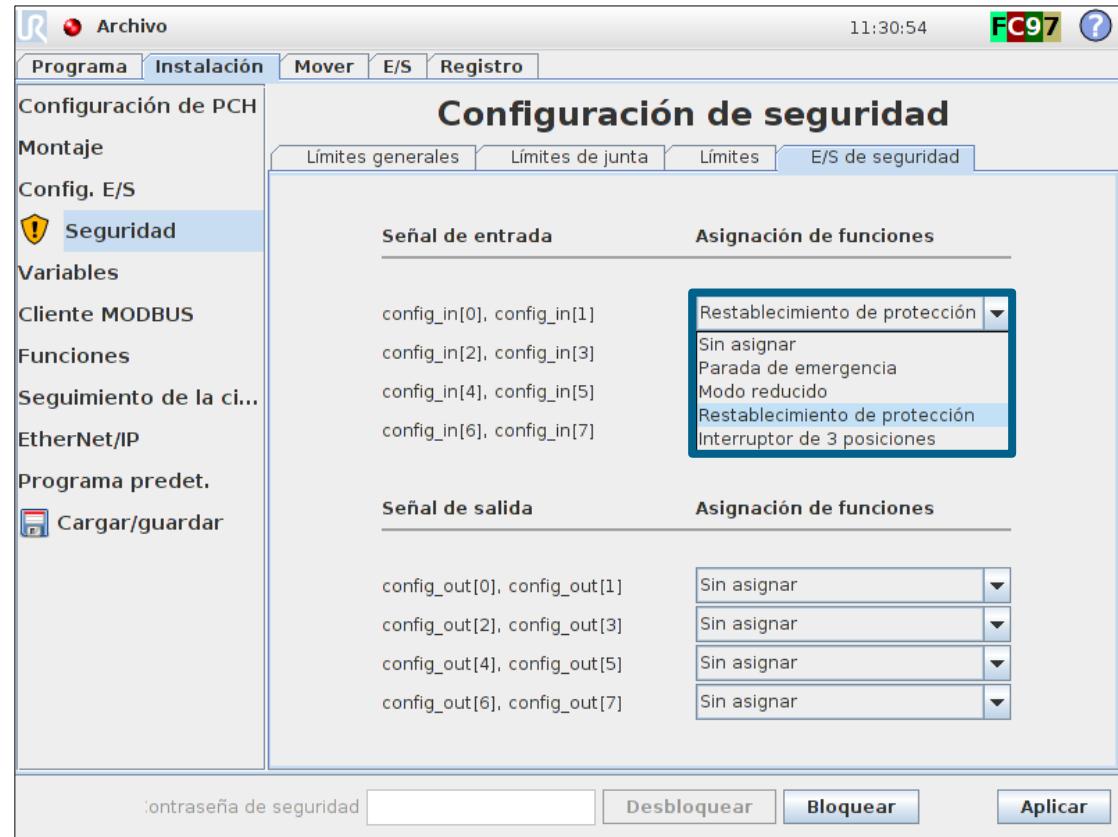
E/S de seguridad

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



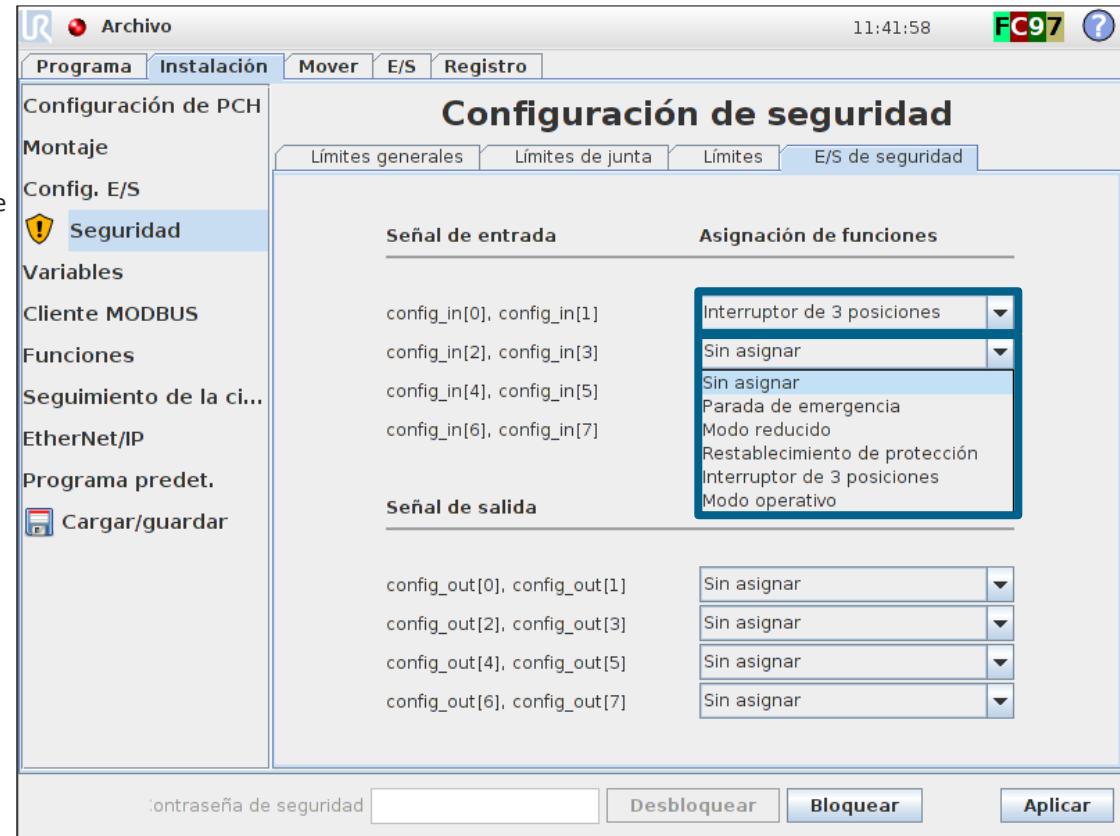
Entradas de seguridad

- Entradas de seguridad
 - Para de emergencia
 - Para la conexión de 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.



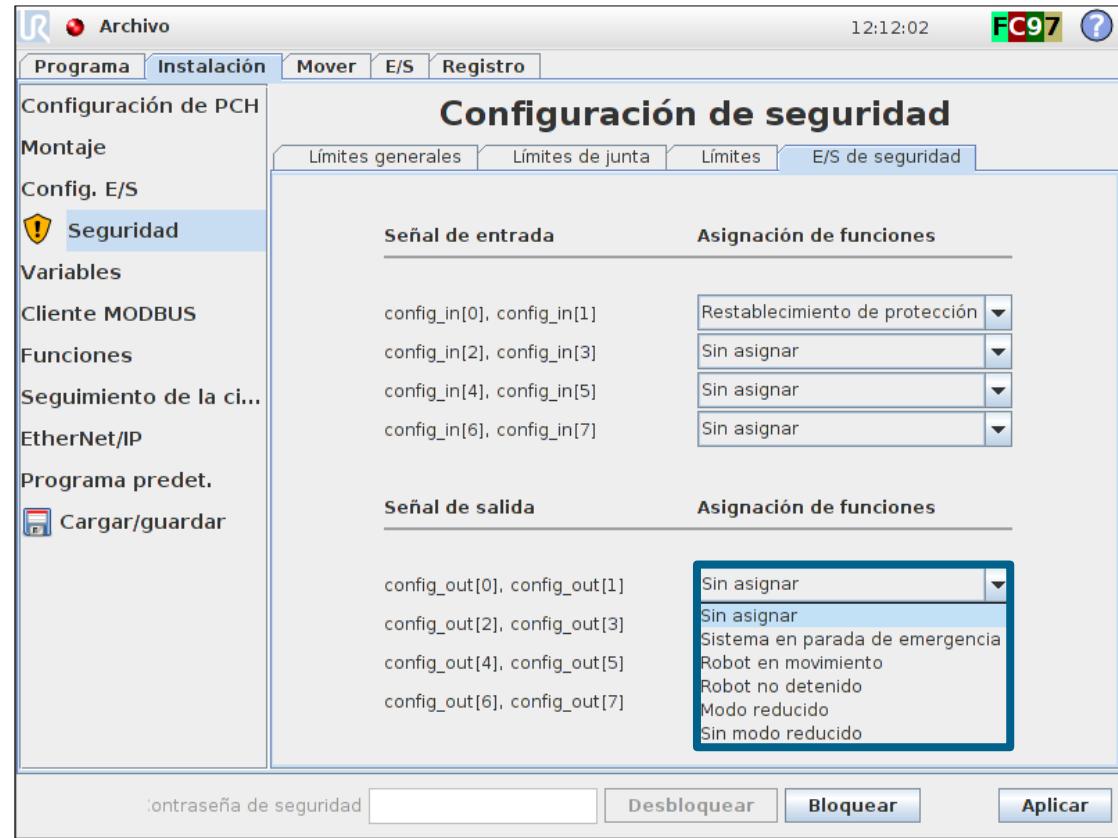
Entradas de seguridad

- Entradas de seguridad
 - Interruptor de 3 posiciones
 - Se puede cablear directamente a la placa de control de seguridad.
 - Modo operativo
 - Sólo se puede seleccionar cuando un interruptor de 3 posiciones ha sido previamente seleccionado.



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

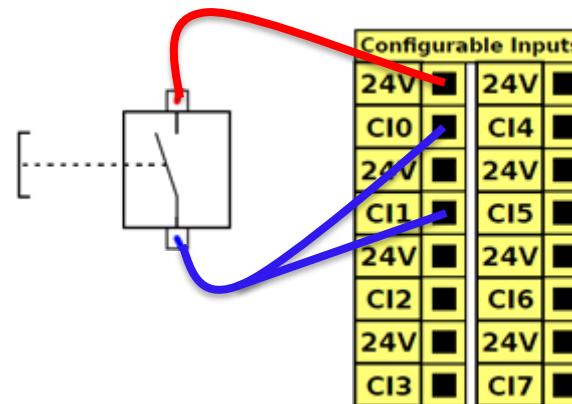


Ejercicio práctico 1

- Crear un programa que utilice zonas de seguridad y entradas de seguridad
- Configurar las Funciones de Seguridad:
 - Seleccione el nivel *Restringido* en la pestaña de *Límites generales*.
 - Crear un plano de seguridad horizontal sobre la *Base* (copiar la función *Base*) con un desplazamiento de -50mm y seleccionar el modo *Ambos* como modo de seguridad, de tal forma que el robot no pueda colisionar con la mesa sobre la que está montado.
 - Crear un plano de seguridad vertical sobre la cota x=400mm y asignarle el Modo Reducido con activador, simulando delimitar la zona de trabajo compartida con un humano.
- Aplicar los cambios y mover el robot manualmente en movimiento libre para sentir el efecto de los límites de seguridad
 - Guardar programa de ejemplo como Lab_exe_12_1.urp

Ejercicio práctico 2

- Configurar un pulsador para activar el modo reducido
 - Conectar el pulsador a las entradas configurables CI0 y CI1
 - Asignar a estas entradas la función Modo
 - Escribir un programa simple con dos o más posiciones y ejecútelo
 - Accione el pulsador para activar el modo de seguridad reducido y compruebe que el robot reduce su velocidad



- Guardar programa de ejemplo como Lab_exe_12_2.urp

Listado de programas de ejemplo

- **Programas de ejemplo**

- movej.urp
- movel.urp
- movel_with_blend.urp
- movep.urp
- movec.urp
- movel_with_relative_waypoint.urp
- wait.urp
- set.urp
- Aviso.urp
- pick_and_place.urp
- loop.urp
- loop_interrupt.urp
- call_sub.urp
- if.urp
- if_else.urp
- var_bool.urp
- var_counter.urp
- var_operator_input.urp
- inst_var_operator_input.urp
- thread.urp
- event.urp
- force_feedback.urp
- script_line.urp
- Interruptor.urp
- pallet.urp
- seek_destack.urp
- force_simple.urp
- modbus.urp
- trigger_reduced_mode.urp
- trigger_reduced_mode_feature.urp
- default.installation
- safety.installation
- Lab_exe_3_1.urp
- Lab_exe_3_2.urp
- Lab_exe_3_3.urp
- Lab_exe_4_1.urp
- Lab_exe_4_2.urp
- Lab_exe_5_1.urp
- Lab_exe_5_2.urp
- Lab_exe_6_1.urp
- Lab_exe_7_1.urp
- Lab_exe_8_1.urp
- Lab_exe_9_2.urp
- Lab_exe_12_1.urp
- Lab_exe_12_2.urp

Test

- Tiene 30 minutos para responder.
- 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!

Resultados del test

- 1. ¿Qué configuraciones en la pantalla de instalación deberemos establecer antes de ejecutar cualquier programa en el robot?
 - a. Configuración del cliente MODBUS
 - b. Datos de PCH, carga y posición de Montaje
 - c. Configuración de seguridad en CB3
 - d. (b) y (c)
 - e. Ninguna de las anteriores
- 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. Todos los anteriores
 - e. Ninguno de los 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 “Bucle”
 - d. Todas las anteriores
 - e. Ninguna de las anteriores

Resultados del test

- 4. ¿Cómo llamarías a otro programa desde el programa actual?
 - a. Usando el comando “Carpeta” para invocar el programa
 - b. Usando el comando “SubProg” para llamar al programa
 - c. Usando el comando “Código de Script”
 - d. Todas las anteriores
 - e. Ninguna de las anteriores
- 5. ¿Qué es una variable de instalación??
 - a. Es global a todos los programas que usen el mismo fichero de instalación
 - b. Son variables que mantienen su valor incluso si se desconecta el robot
 - c. ¿Variable de instalación? Eso no existe
 - d. (a) y (b)
 - e. (c) o (d)
- 6. ¿Cuál 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 PCH, carga y/o montaje no están correctamente definidos
 - c. Valores de aceleración muy elevados
 - d. Todas las anteriores
 - e. Ninguna de las anteriores

Resultados del test

- 7. ¿Por qué es necesario cambiar el valor de la carga útil?
 - a. Los parámetros en los motores se calculan dinámicamente según el dato de carga
 - b. Una definición incorrecta de la carga útil afecta la estabilidad del robot
 - c. No es necesario, trabajar con la carga máxima es suficiente
 - d. (a) y (b)
 - e. (c) y (d)
- 8. ¿Cómo crearías una variable que acepte y guarde los 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. Todas las anteriores
 - e. Ninguna de las anteriores
- 9. ¿Cuáles de las siguientes son posibles causas para el mensaje "C156Axx"?
 - a. Los valores de PCH, carga y/o montaje no están correctamente definidos
 - b. El movimiento del robot se encuentra obstruido
 - c. Valores de aceleración muy elevados
 - d. (a) o (c)
 - e. Este mensaje de error no existe

Resultados del test

- 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?
Solución :- **Es una posición almacenada como variable que contiene 6 valores que pueden ser modificados por programa.**
Se representa como: **p[x, y, z, rx, ry, rz]**

- 12. ¿Cuál es la diferencia entre las instrucciones de movimiento MoveL y MoveP?
 - 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 PCH, MoveL no
 - e. Ninguna de las anteriores

Resultados del test

- 13. ¿Cómo se puede trazar una trayectoria curva?
 - a. Dividiéndola en pequeñas curvas y usando el comando MoveC (dentro de MoveP)
 - b. Usando la función “movec” después de la función “movep” en el lenguaje UR script
 - c. No es posible conseguirlo en este robot
 - d. (a) o (b)
 - e. (b) o (c)
- 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. Ninguno de los anteriores
- 15. ¿Qué hace que los robots UR sean colaborativos?
 - a. No puede sonreír, así que no son colaborativos
 - b. Pueden ser limitados en velocidad
 - c. Pueden ser limitados en fuerza y potencia
 - d. Es necesaria una evaluación de riesgos para verificar la operación colaborativa
 - e. (b) (c) y (d)

Resultados del test

- 16. ¿Cómo se puede monitorizar el valor de una variable en paralelo a la ejecución del programa principal?
 - a. Usando el comando "Evento"
 - b. Añadiendo la secuencia "BeforeStart" en el programa
 - c. Usando el comando "Asignación" dentro de un "Subproceso"
 - d. Todos los anteriores
 - e. Ninguno de los anteriores
- 17. La evaluación de riesgos sólo es necesaria en el caso que un humano trabaje en la zona de trabajo del robot.
 - a. Verdadero
 - b. Falso
- 18. ¿Cuáles de las siguientes funciones han sido añadidas en el CB3?
 - a. Configuración de Seguridad ajustable
 - b. Nuevos alerones
 - c. Encoders absolutos verdaderos
 - d. Comprobación configuración de posición de montaje y carga útil
 - e. (a) (c) y (d)

Resultados del test

- 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 y 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 cuando:
 - a. Se substituye una junta
 - b. Se actualiza el firmware en la junta
 - 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