

Universidad Miguel Hernández de Elche

**MASTER UNIVERSITARIO EN
ROBÓTICA**



UNIVERSITAS
Miguel Hernández

UNIVERSITAS Miguel Hernández

“Programación de robots ABB para tareas de soldadura y pick and place de bisagras en el sector de la automoción”

Trabajo de Fin de Máster

2019-2020

Autor: José Albarranch Martínez
Tutor/es: Carlos Pérez Vidal

AGRADECIMIENTOS

Agradecer a mi familia y amigos por su apoyo y comprensión y expresar el deseo de recuperar el tiempo que no pude estar a vuestro lado.

A mis profesores en el Master de Robótica por su constante apoyo e implicación y en especial a mi tutor Carlos Pérez Vidal por su constante ayuda y las facilidades prestadas.

A todos los investigadores y profesionales que con su trabajo nos ayudan en el aprendizaje de una materia tan compleja y apasionante como es la Robótica.



RESUMEN

El presente trabajo pretende llevar a cabo el diseño de una estación de soldadura y “pick-and-place” de precisión y gran flexibilidad para bisagras en el sector de la automoción. Para ello se diseñará una estación prototipo compuesta por dos robots ABB y parametrización de carácter temporal basada en algoritmos RAPID.

El trabajo está estructurado siguiendo una secuencia lógica en la que cada capítulo aporte los cimientos necesarios para los capítulos posteriores.

En primer lugar, se realiza una introducción al problema presentando las necesidades, limitaciones y objetivos a cumplir durante el transcurso del trabajo.

A continuación, y en base a nuestras necesidades se realiza un estudio del estado del arte, centrado en las industrias de la soldadura automatizada y la industria del empaquetado o “pick and place”, dedicado ampliar nuestros conocimientos en dichos sectores.

En tercer lugar, se emprenderá una etapa de adquisición de conocimientos y habilidades en RobotStudio haciendo uso de manuales del fabricante, así como, distintos tutoriales y videotutoriales existentes en la red.

En el cuarto capítulo se explotan los conocimientos y habilidades adquiridas en los capítulos anteriores para desarrollar una estación de soldadura y pick-and-place robotizada que resuelva las necesidades solicitadas y solvete las limitaciones impuestas.

Finalmente son expuestas las conclusiones y resultados obtenidos, presentando además posibles líneas de estudios futuros.

Palabras clave: ABB, RobotStudio, estación de soldadura, MIG/MAG, flexibilidad, FIFO, ventosa, simulación de robots.

ABSTRACT

The present work intends to carry out the design of a welding station and “pick-and-place” of precision and great flexibility for hinges in the automotive sector. For this, a prototype station consisting of two ABB robots and temporary parameterization based on RAPID algorithms will be designed.

The work is structured following a logical sequence in which each chapter provides the necessary foundations for the next chapters.

First, an introduction to the problem is presented presenting the needs, limitations and objectives to be met during along the work.

Next, and based on our needs, a study of the state of the art is carried out, focused on the industries of automated welding and the industry of packaging or "pick and place", dedicated to expand our knowledge in these sectors.

Thirdly, a stage of acquiring knowledge and skills in RobotStudio will be undertaken using the manufacturer's manuals, as well as different tutorials and video tutorials existing in the network.

In the fourth chapter, the knowledge and skills acquired in the previous chapters are exploited to develop a robotic pick-and-place welding station that meets the requested needs and solves the limitations imposed.

Finally, the conclusions and results obtained are presented, also presenting possible lines of future studies.

Keywords: ABB, RobotStudio, welding station, MIG / MAG, flexibility, FIFO, suction cup, robot simulation.

INDICE GENERAL

AGRADECIMIENTOS	I
RESUMEN	III
ABSTRACT	IV
1 INTRODUCCION	1
1.1 Motivación	1
1.2 Objetivos	1
2 ESTADO DEL ARTE	2
2.1 Introducción.....	2
2.2 Sistemas de soldadura automatizada	7
2.3 Sistemas pick and place	10
3 METODOLOGIA	16
3.1 Introducción a RobotStudio	16
3.2 Flujo de trabajo en RobotStudio	17
3.2.1 Creación de una nueva estación.....	17
3.2.2 Creación de la herramienta de un robot	23
3.2.3 Creación del controlador de un robot.....	26
3.2.4 Creación de ejes de Bases de coordenadas, objetivos y trayectorias.....	29
3.2.4.1 Creación de Base de coordenadas.	30
3.2.4.2 Creación de objetivos.....	32
3.2.5 Creación de trayectorias.....	38
3.2.6 Transportadores	41
3.2.7 Introducción a RAPID.....	49
3.3 Diseño y programación de una estación	56
3.3.1 Creación y diseño de Smart Components	56
3.3.2 Creación de señales E/S.....	70

3.3.3	Lógica de estación	71
3.4	Simulación de la estación	73
3.5	Flexpendant	74
4	DISEÑO Y PROGRAMACIÓN DE UNA ESTACIÓN DE SOLDADURA Y PICK AND PLACE DE BISAGRAS PARA EL SECTOR DE LA AUTOMOCIÓN	82
4.1	Determinación de necesidades y limitaciones	82
4.2	Equipos de la estación	83
4.2.1	Conveyors	83
4.2.2	Subestación de soldadura	84
4.2.3	Subestación de pick and place.....	88
4.3	Diseño de piezas y útiles.....	91
4.4	Programación en RobotStudio.....	94
4.4.1	Creación de la estación.....	94
4.4.2	Inserción de componentes	95
4.4.3	Creación de sistema.....	105
4.4.4	Diseño de la estación WPP	105
4.4.4.1	Subestación soldadura	105
4.4.4.1.1	Objetivos y Trayectorias.....	106
4.4.4.1.2	Smart Components (SC)	115
4.4.4.1.3	Lógica, Entradas y Salidas	118
4.4.4.1.4	Programación Rapid	119
4.4.4.2	Subestación pick and place	122
4.4.4.2.1	Objetivos y Trayectorias.....	122
4.4.4.2.2	Smart Components (SC)	129
4.4.4.2.3	Lógica, Entradas y Salidas	137
4.4.4.2.4	Programación Rapid	138

4.4.4.3	Nivel superior	146
4.4.4.3.1	Diseño de Smart Components de nivel superior	147
4.4.4.3.2	Entradas y Salidas.....	151
5	RESULTADOS	152
5.1	Simulación de la Sub-Estación de soldadura	153
5.2	Simulación de la sub-Estación de Pick-and-place	155
6	CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO.....	157
6.1	Conclusiones	157
6.2	LÍNEAS FUTURAS DE TRABAJO	158
7	BIBLIOGRAFÍA.....	159
8	ANEXOS.....	160
8.1	Glosario General	160
8.2	Aproximación STOP&GO.....	161
8.3	Fichas técnicas	163
8.3.1	Especificaciones técnicas IRB 1520ID	163
8.3.2	Especificaciones técnicas IRB 1600 1.45/10.....	165
8.3.3	Antorcha de soldadura BIZEL ABIROB	167
8.3.4	Estación de limpieza	169
8.4	Planos.....	171
8.5	Código RAPID	176
8.5.1	Tarea T_ROB1 (SOLDADURA)	176
8.5.2	Tarea SUPERVISION	178
8.5.3	Tarea T_ROB2 (Pick and Place).....	179

INDICE DE FIGURAS

Figura 2.1 Principales configuraciones Robótica Industrial.....	3
Figura 2.2 Robot antropomórfico FEDOR.....	3
Figura 2.3 ABB IRB 360	4
Figura 2.4 Robot colaborativo Festo.....	4
Figura 2.5 Robótica colaborativa -Niveles de colaboración	5
Figura 2.6 Analogía brazo humano/robótico.....	5
Figura 2.7 Soldadura por resistencia	7
Figura 2.8 Antorcha de soldadura MIG/MAG.....	8
Figura 2.9 Antorcha TIG	9
Figura 2.10 Robot cartesiano Europack.....	10
Figura 2.11 https://www.youtube.com/watch?v=n7Lxgu9uGJw	11
Figura 2.12 ABB IRB 360 FlexPicker Delta.....	11
Figura 2.13 https://www.youtube.com/watch?v=mz2Tdye6K0k	12
Figura 2.14 ABB IRB 8700	13
Figura 2.15 https://www.youtube.com/watch?v=RJcuz6yorvk&feature=emb_logo...	13
Figura 2.16 pinza de garras ABB Flexgripper	14
Figura 2.17 Pinza de palas Robotiq.....	14
Figura 2.18 pinza de vacío Schmalz	15
Figura 3.1 Menú principal.....	17
Figura 3.2 Opciones pestaña "Archivo"	18
Figura 3.3 Barra de herramientas "Posición Inicial"	19
Figura 3.4 Barra de herramientas "Modelado"	19
Figura 3.5 Barra de herramientas "Simulación"	19
Figura 3.6 Barra de herramientas "Controlador"	19
Figura 3.7 Barra de herramientas "Rapid"	20
Figura 3.8 Barra de herramientas "Complementos"	20
Figura 3.9 Robot IRB 2600	20
Figura 3.10 Biblioteca ABB.....	21
Figura 3.11 Configuración de IRB 2600.....	21
Figura 3.12 Estación de trabajo con IRB 2600	22

Figura 3.13 Detalle Efector Final en la muñeca	22
Figura 3.14 Equipamiento\ Herramientas	23
Figura 3.15 Robot IRB 2600 y herramienta en origen mundo.....	24
Figura 3.16 Herramienta conectada a la muñeca	24
Figura 3.17 Detalle herramienta y nuevo TCP	25
Figura 3.18 Pestaña "Posición"	25
Figura 3.19 Creación Sistema del robot.....	26
Figura 3.20 Configuración del sistema - Selección RobotWare.....	26
Figura 3.21 Configuración del sistema - Mecanismos	27
Figura 3.22 Configuración del sistema - Opciones - Idioma	27
Figura 3.23 Controladora correctamente cargada	28
Figura 3.24 Creación de Sólidos – distintos Sólidos	29
Figura 3.25 Edición de Sólidos	29
Figura 3.26 Posición inicial\ Creación de sistemas de coordenadas	30
Figura 3.27 Detalle nuevo sistema de referencia	31
Figura 3.28 Punto HOME	32
Figura 3.29 Crear punto.....	33
Figura 3.30 Objetivos en las esquinas	33
Figura 3.31 Orientación de objetivos VS TCP	34
Figura 3.32 Copia del primer objetivo reorientado	35
Figura 3.33 Aplicar orientación.....	35
Figura 3.34 Orientaciones corregidas	36
Figura 3.35 Alcanzabilidad Target_10	36
Figura 3.36 Alcanzabilidad Target_20	37
Figura 3.37 Alcanzabilidad Target_30	37
Figura 3.38 Alcanzabilidad Target_40	37
Figura 3.39 Alcanzabilidad Target_50	38
Figura 3.40 Árbol de trayectorias y objetivos.....	38
Figura 3.41 Path_10 automática	39
Figura 3.42 Parámetros	39
Figura 3.43 Trayectoria Path_10 con ajuste Zone "Fine"	40
Figura 3.44 Transportador - Opciones de sistema	41

Figura 3.45 Creación modelo simple de Transportador.....	42
Figura 3.46 Crear Transportador	42
Figura 3.47 Crear conexión.....	43
Figura 3.48 Editar conexión	43
Figura 3.49 Conexión creada	44
Figura 3.50 Pieza de trabajo	44
Figura 3.51 Objeto del transportador.....	45
Figura 3.52 Adjuntar objeto de trabajo	46
Figura 3.53 Objeto de trabajo adjuntado	46
Figura 3.54 posicionamiento "Jog"	47
Figura 3.55 Trayectorias transportador.....	47
Figura 3.56 Instrucciones de trabajo	48
Figura 3.57 Secuencia ordenada.....	48
Figura 3.58 Esquema básico de programa	49
Figura 3.59 Ejemplo Module1.....	50
Figura 3.60 Esquema básico RAPID	51
Figura 3.61 Trayectoria NO deseada	52
Figura 3.62 Trayectoria corregida en RAPID.....	54
Figura 3.63 Componente inteligente (SC)	56
Figura 3.64 Categorías de componentes	57
Figura 3.65 Señales y propiedades	57
Figura 3.66 Primitivos paramétricos.....	58
Figura 3.67 Sensores.....	59
Figura 3.68 Acciones	60
Figura 3.69 Manipuladores.....	61
Figura 3.70 Controlador.....	62
Figura 3.71 Física	62
Figura 3.72 Otros	63
Figura 3.73 Smart Gripper\Servo_fingers.....	64
Figura 3.74 SC Smart GRIPPER 14000.....	65
Figura 3.75 Componentes SC Smart GRIPPER 14000	65
Figura 3.76 Pestaña Diseño	66

Figura 3.77 Señales de E/S.....	66
Figura 3.78 Lógica Smart GRIPPER 14000.....	67
Figura 3.79 Pinza cerrada	67
Figura 3.80 Pinza abierta	68
Figura 3.81 Señales y conexiones	68
Figura 3.82 Sistema robot con herramienta.....	69
Figura 3.83 Señal activación de pinza.....	70
Figura 3.84 Sistema E/S estación de prueba	71
Figura 3.85 Lógica de la estación de prueba	72
Figura 3.86 Simulador E/S OFF	72
Figura 3.87 Simulador E/S ON	73
Figura 3.88 Configuración de la simulación.....	73
Figura 3.89 Virtual FlexPendant	74
Figura 3.90 Ejemplo FlexPendant	75
Figura 3.91 Menú FlexPendant.....	75
Figura 3.92 HotEdit	76
Figura 3.93 Inputs and Outputs.....	76
Figura 3.94 Jogging	77
Figura 3.95 Production Window	77
Figura 3.96 Program Editor.....	78
Figura 3.97 Program Data.....	78
Figura 3.98 Backup and Restore	79
Figura 3.99 Calibration.....	79
Figura 3.100 Control Panel	80
Figura 3.101 Event Log.....	80
Figura 3.102 FlexPendant Explorer.....	81
Figura 3.103 System Info	81
Figura 4.1 Conveyor de rodillos Estándar Robotstudio	83
Figura 4.2 Cinta modular acetal.....	84
Figura 4.3 IRB 1520ID	85
Figura 4.4 Esquema estación de soldadura robotizada.....	85
Figura 4.5 IRC-5.....	86

Figura 4.6 Fronius TPS 4000.....	86
Figura 4.7 Fronius VR1500.....	86
Figura 4.8 Antorcha ABIROB.....	87
Figura 4.9 Binzel ID22.....	87
Figura 4.10 Bidón de hilo 250 Kg.....	87
Figura 4.11 Equipo de limpieza Binzel.....	88
Figura 4.12 IRB 1600.....	89
Figura 4.13 Efactor final de ventosas SCHMALZ.....	89
Figura 4.14 VENTOSA SAOF 80x40 NBR-60 G1/4-AG de SCHMALZ.....	90
Figura 4.15 Socket_IRB1600.....	90
Figura 4.16 Bisagra Tipo-1.....	91
Figura 4.17 Bisagra Tipo-2.....	91
Figura 4.18 Bisagra Tipo-3.....	92
Figura 4.19 Bandeja de transporte.....	92
Figura 4.20 Cinta de transporte.....	92
Figura 4.21 Caja negra/Centrador.....	93
Figura 4.22 Referencia para soldadura.....	93
Figura 4.23 Crear estación.....	94
Figura 4.24 Estación vacía.....	95
Figura 4.25 Menú posición.....	95
Figura 4.26 Fijar posición.....	96
Figura 4.27 Posición de offset.....	96
Figura 4.28 Girar.....	97
Figura 4.29 Situar.....	97
Figura 4.30 Posición conveyors.....	98
Figura 4.31 Posición cintas.....	98
Figura 4.32 Posicionamiento IRB 1600.....	99
Figura 4.33 Posicionamiento IRB 1520ID.....	99
Figura 4.34 Conexión herramienta de soldar con IRB 1520ID.....	100
Figura 4.35 Conexión efactor final de ventosas con IRB 1600.....	100
Figura 4.36 Conexión wire feeder.....	101
Figura 4.37 IRB 1600 sobre pedestal.....	101

Figura 4.38 Colocación perímetro de seguridad	102
Figura 4.39 Equipo de soldadura	102
Figura 4.40 Componentes adicionales.....	103
Figura 4.41 Simulación hilo de alimentación.....	103
Figura 4.42 Detalle caja negra centradora	104
Figura 4.43 Vista en planta de la estación	104
Figura 4.44 Ejemplo Pieza (Rojo línea de soldadura)	106
Figura 4.45 Referencia Soldadura.....	107
Figura 4.46 Cotas Referencia Soldadura.....	108
Figura 4.47 Detalle avance Referencia Soldadura	108
Figura 4.48 Detalle longitud de trayectoria.....	109
Figura 4.49 Pieza y referencia en posición inicial	109
Figura 4.50 Pieza y referencia en posición final	109
Figura 4.51 Longitud cordón.....	110
Figura 4.52 Modelo de referencia en la estación	110
Figura 4.53 Creación Objeto de trabajo	111
Figura 4.54 Trayectoria automática.....	111
Figura 4.55 Trayectoria auxiliar	112
Figura 4.56 Trayectoria corregida.....	112
Figura 4.57 Menú Modificar instrucciones.....	113
Figura 4.58 Trayectoria y alcanzabilidad	113
Figura 4.59 Puntos de entrada y salida	114
Figura 4.60 Trayectoria "Path_WELD".....	114
Figura 4.61 Puntos y Trayectorias	115
Figura 4.62 SC SOLDADURA	117
Figura 4.63 Conexión señal SOLDAR.....	118
Figura 4.64 Creación de entradas y salidas	118
Figura 4.65 WO_PICKPLACE.....	122
Figura 4.66 Robot en Target_0_2 (PICK)	123
Figura 4.67 Robot en Target_0_1	124
Figura 4.68 Robot en Target_1_2 (PLACE).....	124
Figura 4.69 Objetivos Trayectoria 1.....	125

Figura 4.70 Objetivos Trayectoria 2.....	125
Figura 4.71 Objetivos Trayectoria 3.....	126
Figura 4.72 Objetivos creados Tarea T_ROB2	126
Figura 4.73 Trayectoria 1	127
Figura 4.74 Trayectoria 2	127
Figura 4.75 Trayectoria 3	128
Figura 4.76 Instrucciones de movimiento	128
Figura 4.77 SC PICK AND PLACE	129
Figura 4.78 SC APROCIMACION_PICK.....	130
Figura 4.79 SC CLASIFICACION	130
Figura 4.80 SC PIEZA_TIPO_1.....	132
Figura 4.81 SC PIEZA_TIPO_2.....	132
Figura 4.82 SC PIEZA_TIPO_OTRAS.....	133
Figura 4.83 SC SELECCION_PLACE	135
Figura 4.84 SC VENTOSA.....	136
Figura 4.85 Conexiones subestación Pick and Place	138
Figura 4.86 Creación de tarea SUPERVISION.....	139
Figura 4.87 Estación WPP	146
Figura 4.88 SC CINTA_SUMINISTRO	148
Figura 4.89 SC CAJA_NEGRA	148
Figura 4.90 SC PERIMETRO_SEGURIDAD	149
Figura 4.91 SC SINCRONIZACION_TAREAS	151
Figura 5.1 Estación parada.....	152
Figura 5.2 Estación en MARCHA	152
Figura 5.3 Pieza activando el sensor.....	153
Figura 5.4 Inicio de soldadura.....	153
Figura 5.5 Limpieza - corte	154
Figura 5.6 Limpieza - TCP	154
Figura 5.7 Pieza sujeta	155
Figura 5.8 Pieza 1 depositada	155
Figura 5.9 Pieza Tipo 2 Depositada.....	156

1 INTRODUCCION

1.1 Motivación

La industria de la automoción ha sido históricamente el mayor precursor de la robótica, siendo su primer consumidor intensivo, estas industrias, con altos estándares de calidad y elevadas exigencias en cuanto a productividad hacen necesaria, cuando no imprescindible, la continua adaptación de procesos, automatizándolos y robotizándolos haciéndolos más productivos, motivo por el cual se estima que en 2018 más del 30% del suministro total de robots industriales estaban destinados a esta industria.

La automatización y robotización de procesos involucra costes elevados, por lo que es necesario un estudio previo que permita valorar de forma objetiva su viabilidad e interés.

Los procesos que involucran altas producciones, tanto anuales, como diarias, a la par que altas exigencias de precisión y calidad son candidatos perfectos para robotización.

Si bien la productividad y mejora de costes serán los principales justificantes de las empresas por robotizar ciertos procesos, también se persigue librar a los operarios de tareas repetitivas, penosas y/o peligrosas que a su vez no pueden competir ni en velocidad ni en precisión con los robots industriales.

1.2 Objetivos

El objetivo de este Trabajo de Fin de Master (en adelante TFM) es el desarrollo de una estación robotizada de soldadura y pick and place, adquiriendo por el camino los conocimientos y habilidades necesarios para su completo diseño y programación mediante el software de simulación RobotStudio, ampliando, además, conocimientos en las distintas tareas implicadas y sus distintos componentes.

2 ESTADO DEL ARTE

2.1 Introducción

Para hablar de robótica en general, debemos comenzar por diferencias Robótica Industrial y Robótica de Servicio, esta primera clasificación debe estar reglamentada y recogida en las normativas ISO para el desarrollo, fabricación y comercialización de los distintos productos.

Las normativas ISO son definidas por la **International Federation of Robotics(IFR)**.

Según la ISO 8373 nos referimos como Robot Industrial a: “Manipulador multifuncional, controlado automáticamente, reprogramable en tres o más ejes, que puede estar fijo o móvil para uso en aplicaciones de automatización industrial”.

A su vez, clasifica el parque de robots actual en dos sentidos:

- Aplicaciones (Manipulación, soldadura, montaje...) y Sectores (Alimentación, textil, maquinaria industrial...).
- AUTO (Automóvil) y NO AUTO (No automóvil).

Por su parte, IFR entiende por robot de servicio: “Robot que opera de forma parcial o totalmente autónoma al servicio del bienestar de los seres humanos y de equipamientos, excluyendo operaciones manufactureras” [15].

En el contexto mundial, la Robótica Industrial será fundamental en la cuarta Revolución Industrial en la que estamos inmersos dado que tratamos con un producto que mejora la productividad, costes y calidad de producto terminado en la mayoría de plantas en las cuales se integran.

En este contexto y para el actual TFM nos interesan los Robots Industriales; A continuación, se hace un breve repaso a las tipologías y clasificaciones más frecuentes. (Figura 2.1)

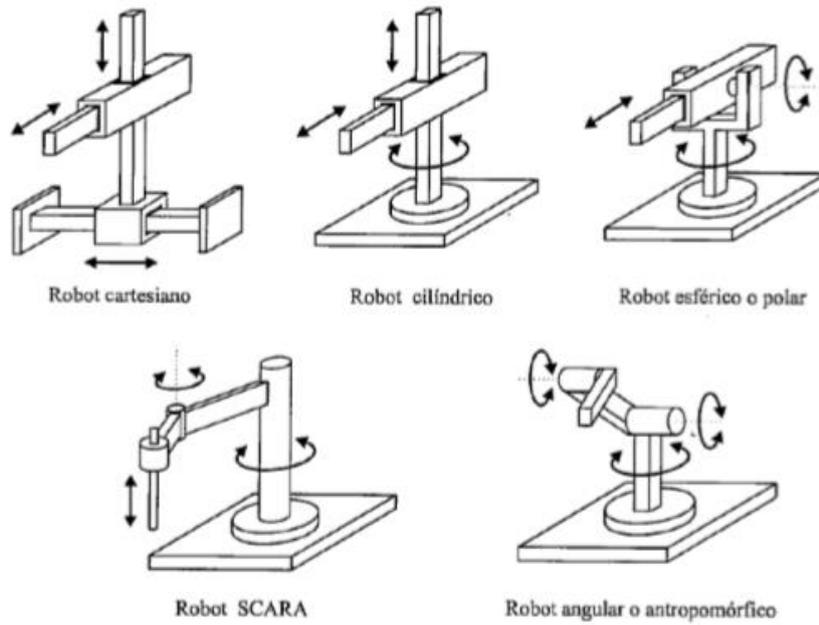


Figura 2.1 Principales configuraciones Robótica Industrial

Como vemos en la figura anterior, podemos hacer una primera distinción en función de la geometría de la estructura mecánica distinguiendo [7]:

- Cartesiano: Articulaciones lineales.
- Cilíndrico: Articulaciones lineales y rotacionales.
- Polar: Una articulación lineal sobre articulaciones rotacionales
- SCARA: Combinación de distintos tipos de articulaciones.
- Antropomórfico: Articulaciones rotacionales.



Figura 2.2 Robot antropomórfico FEDOR

También existen robots paralelos tipo “Araña” como la gama Delta de ABB con articulaciones prismáticas y de rotación, si bien son aptos para rápidas labores de paletización, su limitada flexibilidad (en comparación con los robots tipo brazo) los descartaron como unidad paletizadora en este proyecto.



Figura 2.3 ABB IRB 360

En la actualidad, cada vez más, van ganando popularidad los Robots Colaborativos, robots diseñados para trabajar en colaboración con humanos mejorando su productividad, liberándoles de tareas monótonas y repetitivas y permitiéndoles centrarse en trabajos más complejos o finalizar la tarea en colaboración con el robot en un espacio compartido. En un entorno colaborativo, una persona aporta destreza, flexibilidad y la capacidad de resolver problemas, mientras que un robot colaborativo ofrece fuerza, resistencia y precisión en la realización de la tarea en cuestión.



Figura 2.4 Robot colaborativo Festo

La programación de los “cobots” cada vez es más sencilla e intuitiva y ofrecen la máxima flexibilidad para que el robot ejecute una función diferente a la anterior. Son ligeros y se pueden mover fácilmente por la fábrica.

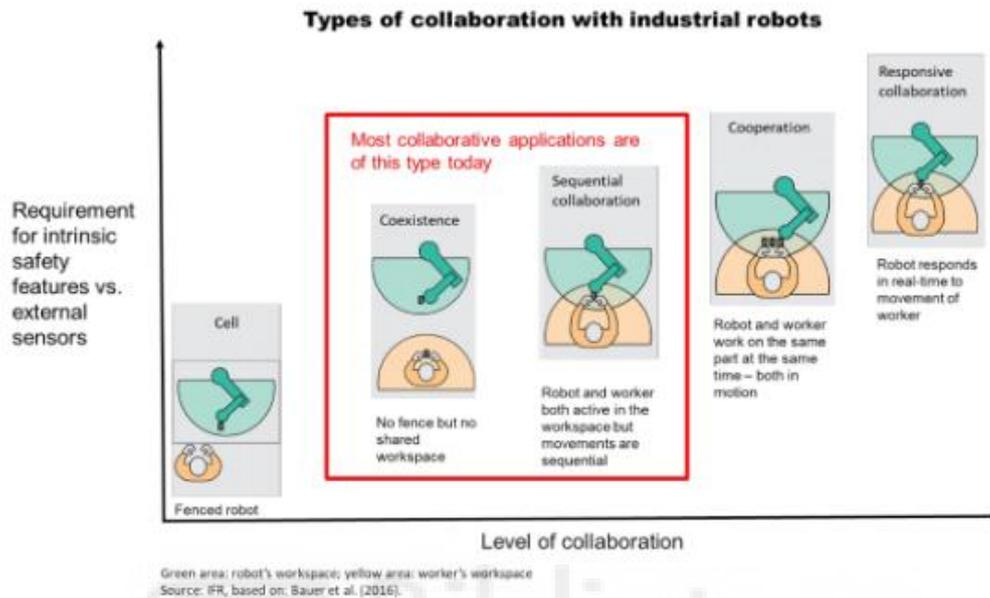


Figura 2.5 Robótica colaborativa -Niveles de colaboración

No obstante, es un producto aun en desarrollo cullas limitaciones en cuanto a fuerza y velocidad actuales los convierten más en herramientas y no los hacen apropiados para los objetivos tratados en este TFM.

De en adelante, el estudio del arte se basará principalmente, por tanto, en Robots Industriales de tipo antropomórfico.

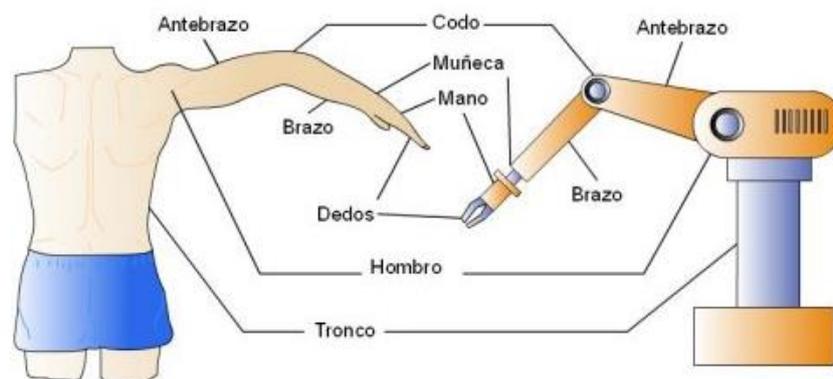


Figura 2.6 Analogía brazo humano/robótico

El brazo de antropomórfico es con diferencia el robot más usado en la industria de la soldadura, esto se debe a que sus seis grados de libertad le dan gran facilidad de acceso a todas las posiciones requeridas por una tarea, además, es la estructura más asequible económicamente al ser utilizada en muchos campos con mayor facilidad de amortización.

Estos robots están compuestos de forma genérica por una estructura mecánica, sistemas de transmisión, motores, encoders y un sistema de control; además, por cada motor, existe un accionamiento lineal dedicado a bloquear el robot en caso de fallo, ya sea caída de corriente o paro de emergencia, consiguiendo “que no se pueda producir la caída o proyección de ningún elemento móvil de la maquina o de ninguna de las piezas sujeta por ella” como reza el BOE, 2006/42/EC.

En cuanto a la industria, el uso de Robots es cada vez más frecuente, siendo habituales y casi indispensables en grandes industrias como: Automoción (Soldadura, pintura, mecanizado), Alimentación (Paletizado, manipulación), Química y Farmacéutica (Paletizado, posicionado, manipulación), Inyección y Matricería (manipulación, ensamblaje).

En los siguientes apartados se hace una aproximación a los sectores de aplicación relevantes en el actual TFM.

2.2 Sistemas de soldadura automatizada

Hablar de soldadura automatizada, es quizás, hablar de una de las aplicaciones más destacables dentro de la robótica, nacida en la industria de la automoción por su imperativa necesidad de precisión, elevadas producciones y reducción de costes.

Hablando de soldadura robotizada, podemos encontrar cuatro métodos principales [9]:

- **Soldadura por resistencia**

Se utiliza principalmente para soldar “por puntos”, se basa en la aplicación de una corriente eléctrica a través de dos electrodos en contacto con las chapas a unir, en este caso, la discontinuidad entre los metales a unir será la responsable de crear una elevada resistencia eléctrica que generará el calor necesario para la fusión del material y provocar la consecuente soldadura.



Figura 2.7 Soldadura por resistencia

- **Soldadura MIG/MAG**

Quizás el método de soldadura más utilizado en la industria en general y automoción en particular, hablando de automatización y robótica.

Se trata de un método de soldadura por fusión mediante un arco eléctrico protegido por un gas protector, además, se utiliza un material de aporte en forma de hilo que ejerce de electrodo.

La utilización de un gas protector está destinado a evitar la contaminación del cordón de soldadura por gases atmosféricos, el gas protector seleccionado será el factor

distintivo entre soldadura MIG, si utilizamos Helio o Argón como gas protector, o, MAG en el caso de utilizar CO₂.

Este proceso consiste en alimentar hilo mediante poleas dentadas hacia la pieza a soldar de manera que se genere un arco eléctrico constante para generar una fusión controlada, el hilo es fundido con los bordes de la pieza generando la unión mientras el gas protector es conducido por un conducto de manera concéntrica al hilo de aporte.

En la siguiente figura 2.8 podemos ver el esquema básico de estas antorchas.

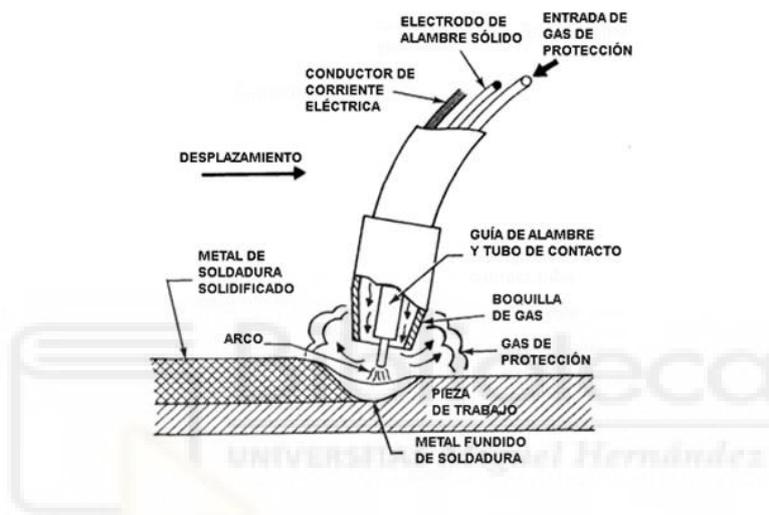


Figura 2.8 Antorcha de soldadura MIG/MAG

Puesto que esta metodología es la seleccionada para el actual TFM se desarrollará en mayor profundidad en cuanto a componentes en capítulos posteriores.

- **Soldadura TIG**

Reservado para aplicaciones especiales por su escasa productividad y elevados tiempos de preparación, generalmente se utiliza con aluminio y aleaciones especiales. Sus principales características son el empleo de electrodos de Tungsteno no consumibles dentro de una atmosfera de gas protector y el aporte de material realizado de manera externa a la antorcha.

Esta tipología de soldadura presenta la mayor dificultad a la hora de automatizar, debido principalmente a la difícil gestión de aporte de material, ya que un equipo TIG

convencional no gestiona de forma semiautomática el aporte de varilla dependiendo este del operario.



Figura 2.9 Antorcha TIG

Vistos los principales métodos de soldadura, cabe recordar, que, como cualquier proceso de automatización, el nivel de automatización será un reflejo de los sensores incorporados, haciendo de las máquinas más capaces de lidiar de forma autónoma con posibles variaciones en el proceso, si la variación de un proceso puede predecirse con precisión, posiblemente la implementación de sensores permitirá que la máquina detecte estas variaciones y realice los ajustes necesarios.

Sin entrar en detalle, los sensores estándar comerciales incluyen métodos como la triangulación láser, cámara de visión máquina, cámaras de soldadura, cámaras termográficas, monitores de parámetros de proceso, detectores de sonido por arco y espectrofotómetros [16].

Estos sensores pueden estar destinados a monitorizar distintas propiedades del proceso como el seguimiento de juntas, control de longitud de arco o la penetración de la soldadura.

2.3 Sistemas pick and place

En la actualidad podemos encontrar una enorme cantidad de empresas dedicadas a la producción, venta y distribución de robots destinados a él paletizado automático, si bien, la mayoría de las tipologías de robot vistas con anterioridad se podrían aplicar a labores de paletizado, históricamente destacan tres por su funcionalidad y eficiencia:

- **Robot cartesiano**

Ofreciendo movimientos lineales en 3 ejes (X, Y, Z), facilita mucho la programación simplificando las ecuaciones de control, contando, además, con la ventaja de poder agrupar pales de forma lineal.

Estos robots suelen estar montado sobre grandes estructuras lo que produce un área de trabajo de grandes dimensiones.

Diversos factores entran en juego en cuanto a los ciclos realizables por estos robots, siendo los principales, las especificaciones del robot, tamaño del producto, dimensiones de mosaico final...



Figura 2.10 Robot cartesiano Europack



Figura 2.11 <https://www.youtube.com/watch?v=n7Lxgu9uGJw>

- **Robot delta**

Estos robots tienen un funcionamiento similar a los cartesianos, suelen ir montados en techo o una estructura de soporte, ofrecen una elevada velocidad de manipulación de piezas a coste de una capacidad de carga muy limitada y un área de trabajo bastante inferior a los robots cartesianos, por ejemplo el ABB IRB 360 tiene una capacidad de carga de 3 Kg, en un espacio de trabajo entre 0,8 y 1,6 metros.

Su velocidad de ciclos es muy elevada, por lo que suele tratarse de la opción óptima para operaciones de pick and place siempre y cuando, las piezas de trabajo y áreas de trabajo necesarias se encuentren dentro de sus límites.



Figura 2.12 ABB IRB 360 FlexPicker Delta



Figura 2.13 <https://www.youtube.com/watch?v=mz2Tdye6K0k>

- **Robot antropomórfico**

Los robots antropomórficos, en su faceta paletizadora, trabaja en espacios mas reducidos que los robots cartesianos vistos previamente. En esta ocasión, la disposición de los pales u objetos de trabajo, solo puede realizarse en el espacio circundante del robot.

Estos robots se encuentran con una gran variedad de configuraciones en función, entre otras cosas, del peso a transportar y su distancia, siendo bastante superiores a los robot tipo delta.

Por ejemplo, podemos encontrar dentro del catálogo de ABB, configuraciones que posibilitan la manipulación de objetos de 800 Kg y hasta 3,5 metros como es el caso del IRB 8700.



Figura 2.14 ABB IRB 8700



Figura 2.15 https://www.youtube.com/watch?v=RJcuz6yorvk&feature=emb_logo

Una vez resuelta la selección del robot en función de las necesidades, una de las partes más importantes será la herramienta de efecto final responsable de la sujeción de las piezas para su transporte, si bien existen multitud de herramientas siendo incluso habitual el diseño de herramientas para procesos específicos, podemos distinguir las siguientes tipologías principales:

- **Pinza de garras:** Generalmente utilizadas para el transporte de palets.



Figura 2.16 pinza de garras ABB Flexgripper

- **Pinzas de palas:** Combinación de palas fijas y móviles de funcionamiento similar a dedos.



Figura 2.17 Pinza de palas Robotiq

- **Ventosas de vacío:** Conjunto de ventosas capaz de sujetar objetos mediante succión por vacío, suelen ir dotadas de sistemas de cierre y bloqueen las ventosas que no estén en contacto con objetos, además requieren del uso de equipos neumáticos y un compresor.



Figura 2.18 pinza de vacío Schmalz

- **Pinzas mixtas:** Configuraciones basadas en la combinación de las anteriores para aplicaciones específicas.

3 METODOLOGIA

3.1 Introducción a RobotStudio

Como reza en la página web de ABB [10]:

“El software de simulación y programación fuera de línea de ABB, RobotStudio, permite efectuar la programación del robot en un ordenador en la oficina sin interrumpir la producción.

RobotStudio proporciona las herramientas para incrementar la rentabilidad de su sistema robotizado mediante tareas como formación, programación y optimización, sin afectar la producción, lo que proporciona numerosas ventajas, como reducción de riesgos, arranque más rápido, transición más corta e incremento de la productividad.

RobotStudio se ha construido en el VirtualController de ABB, una copia exacta del software real que hace funcionar su robot en producción. Ello permite simulaciones muy realistas, con archivos de configuración y programas de robot reales e idénticos a los utilizados en su instalación.”

Si bien en la actualidad se encuentra disponible la versión RobotStudio 2019.2, para la realización de este proyecto se ha empleado RobotStudio 6.08 por cuestión de las licencias facilitadas por la universidad.

Para comenzar, deberá ser instalado el software RobotStudio en nuestra computadora, cabe decir que RobotStudio es un software privado de la empresa ABB, por lo tanto, para utilizarlo se deberá activar una versión de prueba de 30 días o utilizar alguna licencia disponible, tras valorarse la opción de trabajar con la última versión con licencia de prueba, se optó por utilizar una versión anterior gracias a la licencia facilitada por la Universidad Miguel Hernández de Elche evitando la restricción de 30 días de la versión de prueba ante la incertidumbre de la duración del proyecto.

3.2 Flujo de trabajo en RobotStudio

3.2.1 Creación de una nueva estación

Tras la pertinente instalación y activación de licencias nos encontramos la siguiente pantalla:

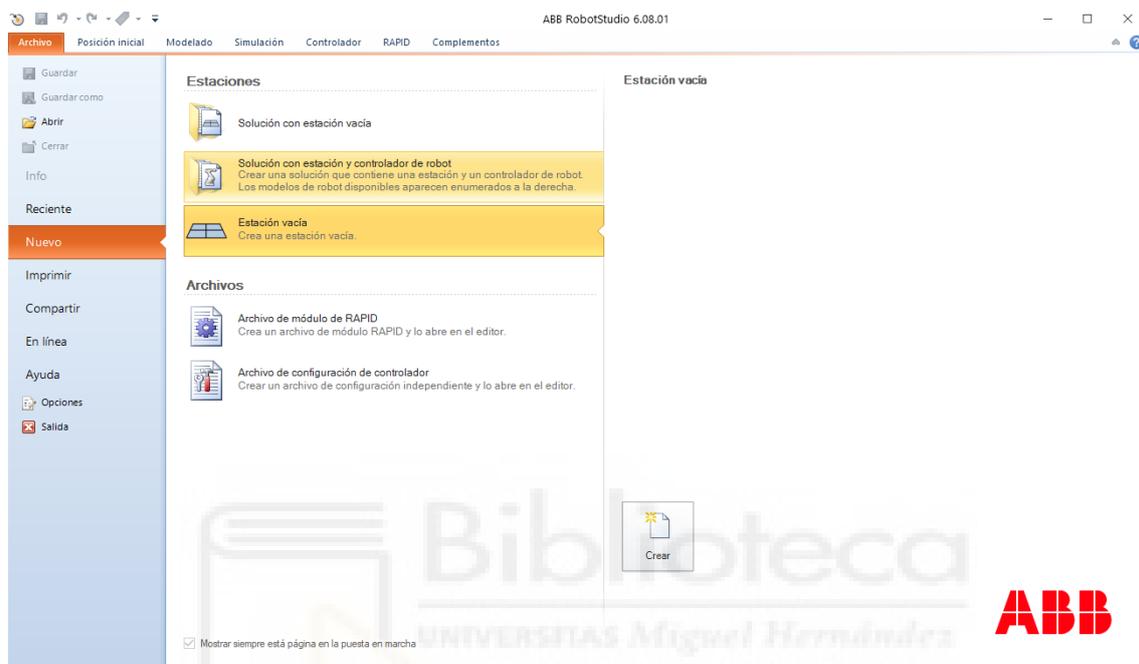


Figura 3.1 Menú principal

Podemos observar como la disposición de los elementos en cintas superiores se corresponde con el esquema clásico de la mayoría de programas de Windows.

Antes de hacer un repaso a las principales opciones que nos encontramos en el software crearemos nuestra estación, como se muestra en la figura anterior, disponemos de tres opciones:

- **Solución de estación vacía:** Crea una estructura de archivos de solución con estación vacía
- **Solución con estación y controlador de robot:** Contiene una lista con los robots que podemos utilizar, al cargarla crea una solución que contiene una estación y un controlador de robot.
- **Estación vacía:** Crea una estación vacía.

Para este pequeño tutorial seleccionamos “Estación vacía”.

Aprovechamos esta primera carga para hacer un breve repaso a la estructura del software RobotStudio, como se puede observar en la figura anterior, la ventana queda dividida en cuatro espacios que modificaremos, eliminaremos o añadiremos a nuestro interés y analizaremos más adelante.

Analizando la cinta de opciones superior, podemos observar que está dividida en siete pestañas que supone el grueso del programa, estas pestañas contienen multitud de opciones y características, si bien todas resultan interesantes y prácticas, a continuación, se explicarán los conceptos principales para abordar el proyecto y durante el desarrollo de este se irán explicando con más extensión de ser preciso.

Las distintas pestañas que podemos encontrar, son:

- **ARCHIVO**

Encontramos las clásicas opciones de Windows como: Nuevo, Reciente, Guardar, Guardar como, Abrir...

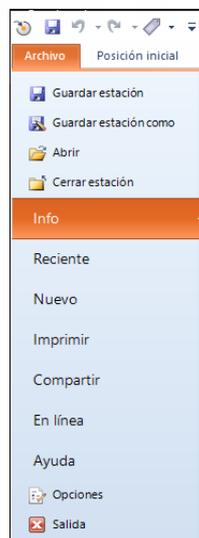
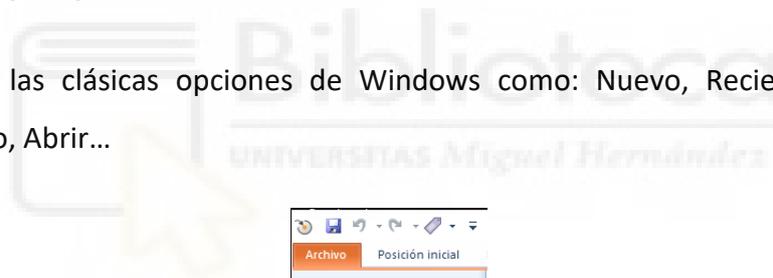


Figura 3.2 Opciones pestaña "Archivo"

- **POSICION INICIAL**

Podemos considerarla la pestaña principal donde podemos añadir el robot, controlador, bibliotecas, geometrías, puntos, trayectorias, herramientas y en definitiva todos los elementos que conformaran nuestro proyecto.



Figura 3.3 Barra de herramientas “Posición Inicial”

- **MODELADO**

Aquí crearemos componentes, componentes inteligentes, sólidos, superficies, mecanismos y demás elementos que deseemos y no se encuentren en las bibliotecas.



Figura 3.4 Barra de herramientas “Modelado”

- **SIMULACIÓN**

Permite realizar las tareas propias de la simulación como análisis de colisión, diseño de lógica de la estación, monitoreo...



Figura 3.5 Barra de herramientas “Simulación”

- **CONTROLADOR**

Opciones relativas al controlar, permite añadirlo, reiniciarlo, modificar herramientas, configurar el controlador, así como abrir la paleta del FlexPendant.



Figura 3.6 Barra de herramientas “Controlador”

- **RAPID**

Se utiliza para la programación textual mediante código del robot.



Figura 3.7 Barra de herramientas “Rapid”

- **COMPLEMENTOS**

Permite la descarga e instalación de algunos paquetes de complementos y su instalación como por ejemplo Painting, Machining, Cutting, Palletizing... mención especial a RobotWare, modulo imprescindible que si bien en esta versión venia incluido en Robot estudio, en otras versiones anteriores y posteriores no lo está y es imprescindible su instalación para empezar a trabajar.



Figura 3.8 Barra de herramientas “Complementos”

Habiendo creado nuestra estación, procedemos a cargar un robot, para este pequeño tutorial podríamos seleccionar cualquiera de la biblioteca de ABB disponible [11], no obstante, por estudios previos, así como, por ser uno de los más empleados en la industria nos decantamos por el IRB 2600, clásico robot de 6 grados de libertad, no se profundizará de momento en los detalles de este robot pues no es necesario para la cuestión que nos ocupa.



Figura 3.9 Robot IRB 2600

Para ello nos dirigimos a la pestaña Posición inicial y hacemos clic en Biblioteca ABB y posteriormente IRB 2600.

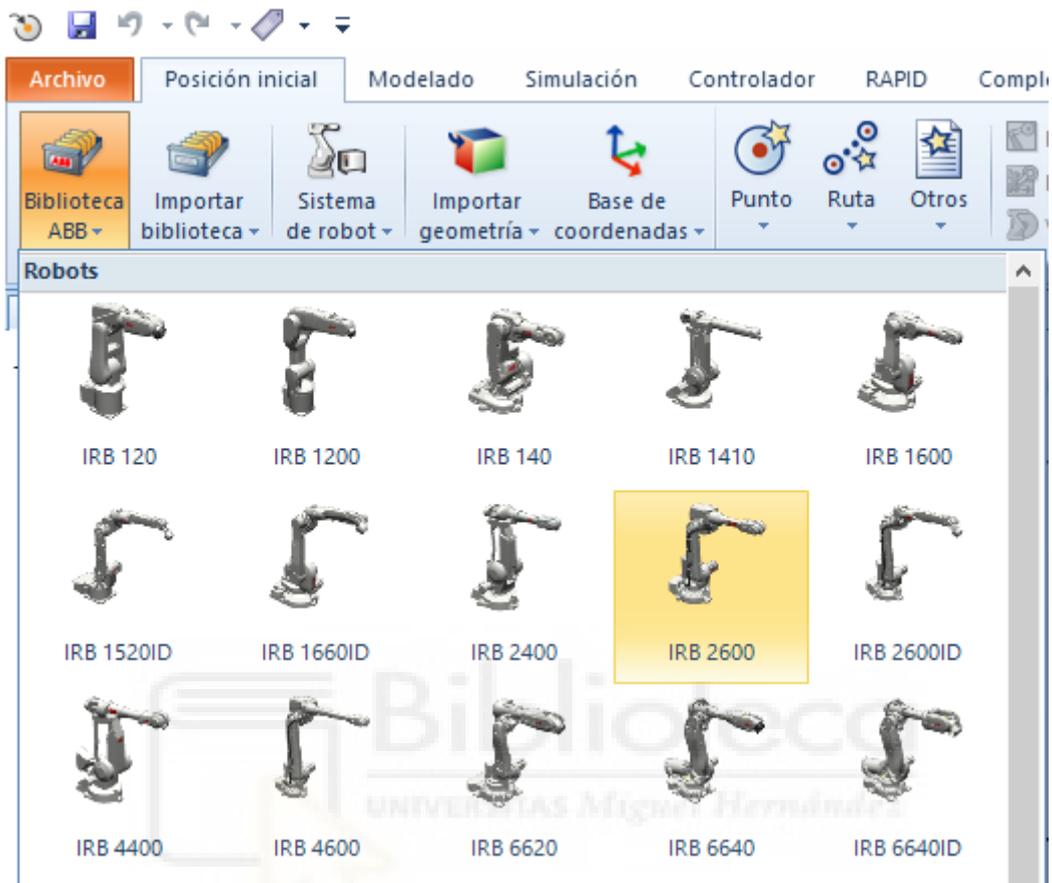


Figura 3.10 Biblioteca ABB

Nos aparece una ventana con las distintas configuraciones posibles, verificamos que se encuentre a nuestro gusto y aceptamos.

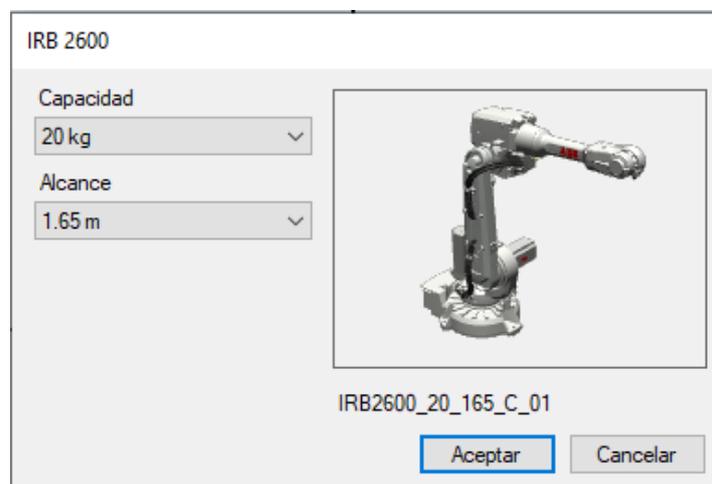


Figura 3.11 Configuración de IRB 2600

Podemos observar cómo se ha cargado de forma correcta nuestro robot, además, en la barra lateral izquierda podemos comprobar en la pestaña de diseño, que ahora el robot es un mecanismo perteneciente a nuestra estación.

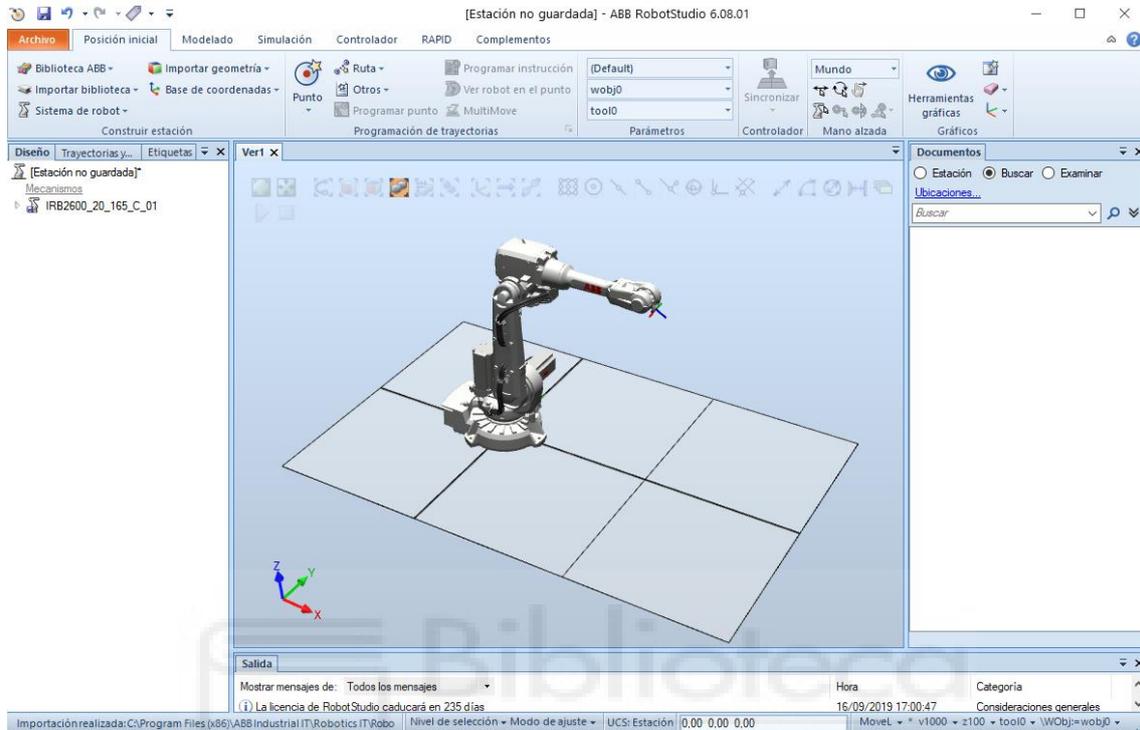


Figura 3.12 Estación de trabajo con IRB 2600

Interesante observar que nuestro robot tiene dos centros de coordenadas, por un lado, tiene uno en su base, inalterable, que en el origen coincide con las coordenadas mundo, por otro lado, se observa que tiene el TCP o efector final en su muñeca.

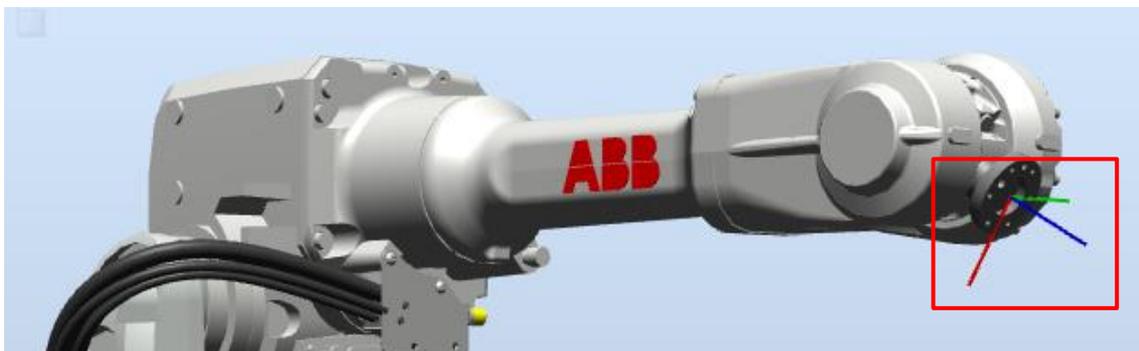


Figura 3.13 Detalle Efector Final en la muñeca

3.2.2 Creación de la herramienta de un robot

La mayoría de robots comerciales necesitan una herramienta como efector final anclada a la muñeca del robot, destinada a realizar las distintas tareas asignadas al robot, como por ejemplo un atomizador para tareas de pintura, pinzas y/o ventosas en el caso de pick-and-place o un soldador para trabajos de soldadura.

A nivel de programación también es importante contar con esta herramienta para el diseño de objetivos y rutas.

Podemos crear herramientas a nuestro antojo mediante soluciones como Autodesk Inventor o Solid Works y exportándolas en formatos neutros tipo *.sat *.stl; También es posible descargar e importar bibliotecas de otros fabricantes, no obstante, para este tutorial se hará uso de las herramientas disponibles en su biblioteca o en su centro de Complementos.

Para acceder a la biblioteca de herramientas disponible, tan solo accedemos a la pestaña Posición inicial, Equipamiento y descendemos por el panel hasta herramientas.

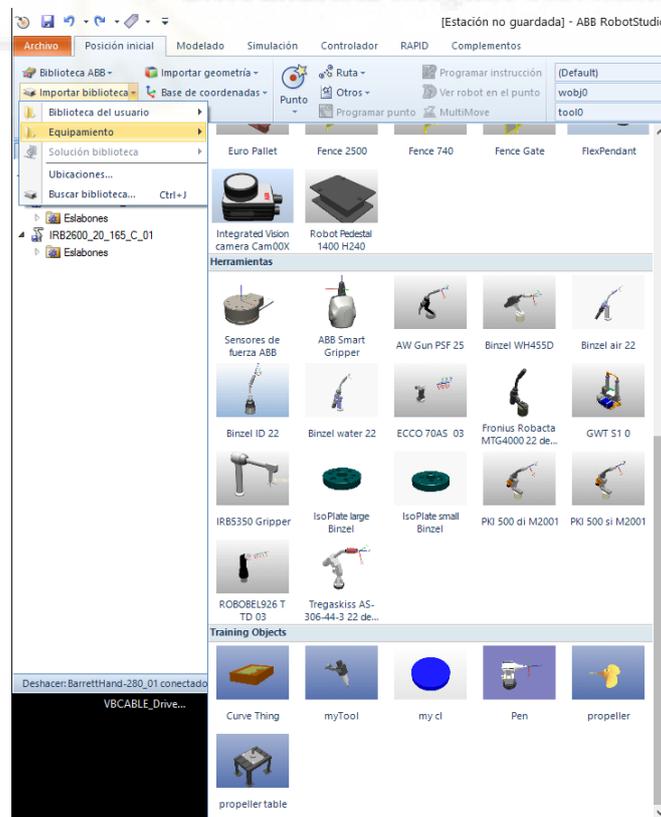


Figura 3.14 Equipamiento\Herramientas

Por ejemplo, vamos a utilizar la pistola de soldadura “Fronius Robacta MTG4000” disponible en la biblioteca.

Para cargarla simplemente pinchamos sobre ella, y observamos que, por defecto, el origen de la herramienta se encuentra en el origen mundo.

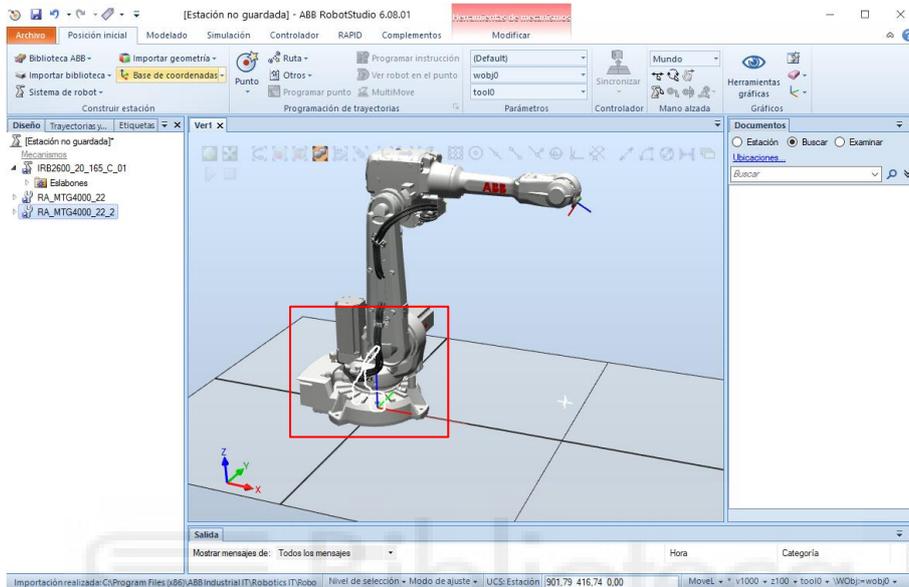


Figura 3.15 Robot IRB 2600 y herramienta en origen mundo

Para llevarla a su posición final y que coincida el origen de la herramienta con la muñeca del robot, en el panel de diseño en la barra lateral izquierda, arrastramos la herramienta dentro del robot, nos preguntara si deseamos actualizar la posición y aceptamos.

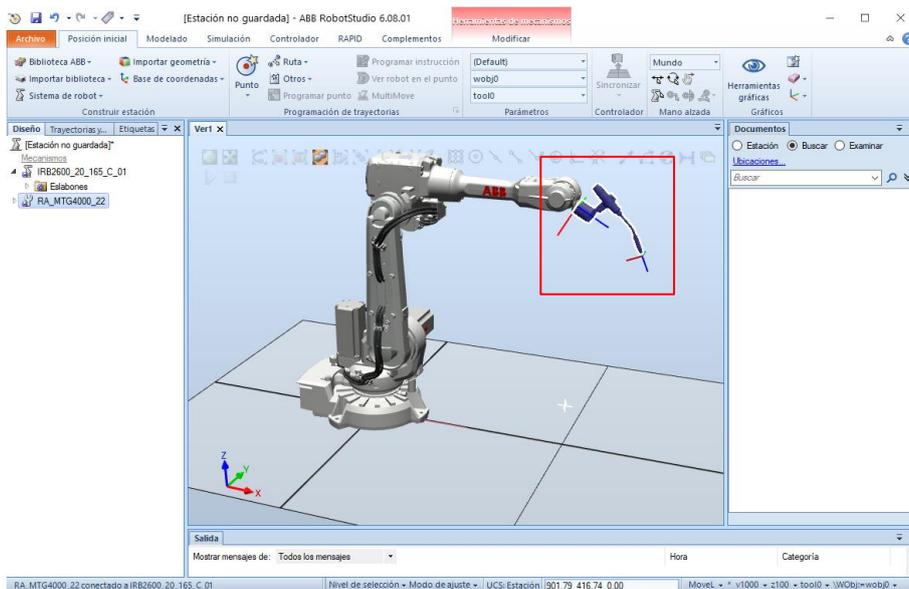


Figura 3.16 Herramienta conectada a la muñeca

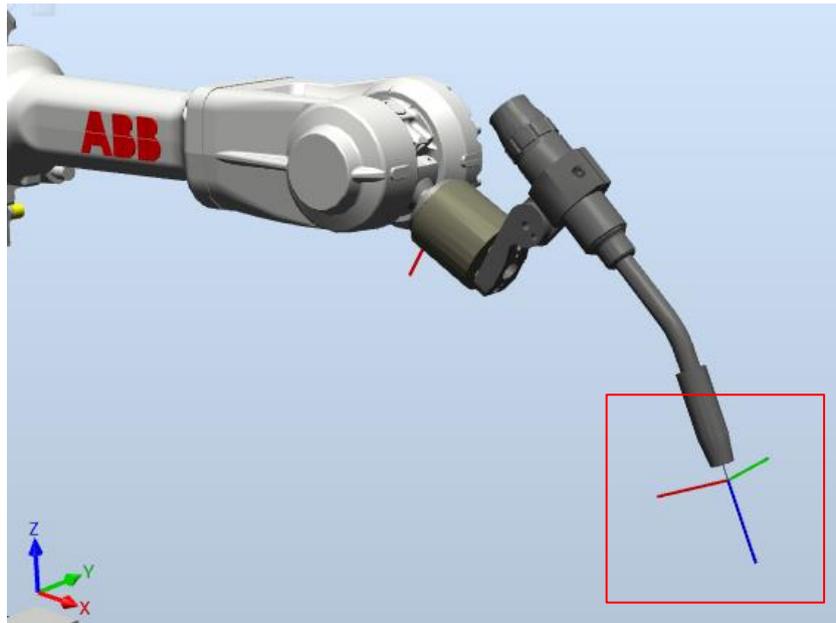


Figura 3.17 Detalle herramienta y nuevo TCP

Podemos comprobar como la herramienta se ha posicionado correctamente como deseábamos, no obstante, y de no ser así, podríamos modificar su posición mediante las distintas opciones que encontramos si presionamos con el botón derecho sobre la herramienta y desplegamos el menú posición.

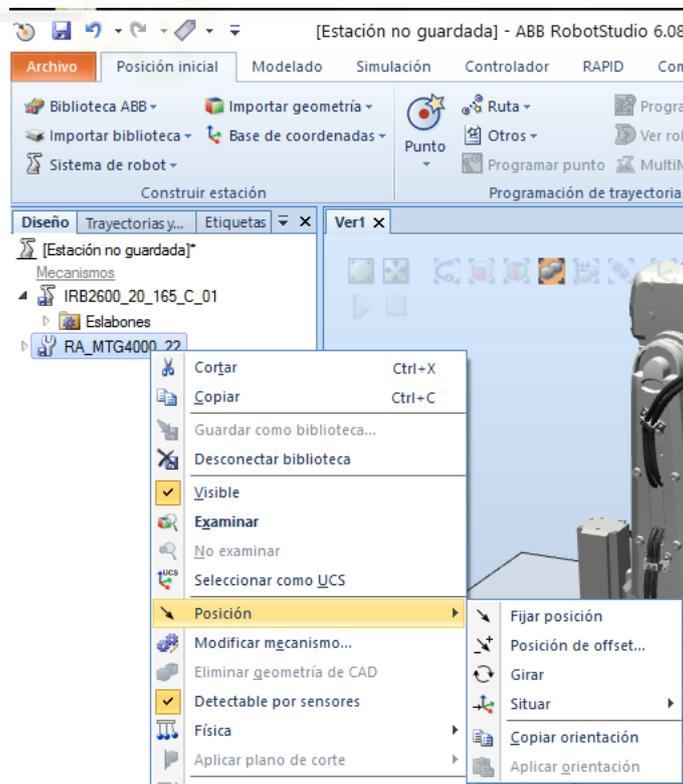


Figura 3.18 Pestaña "Posición"

3.2.3 Creación del controlador de un robot

Ahora tenemos un robot y una herramienta asociada a este en nuestra estación, el siguiente paso será dotar el conjunto de un controlador de proporcione inteligencia al robot y lo capacite para programar puntos, trayectorias, movimientos o accionamientos necesarios.

Para insertar un controlador acudiremos de nuevo a la pestaña “Posición inicial” y en “Sistema de robot” seleccionamos “Desde diseño” de entre las distintas opciones.

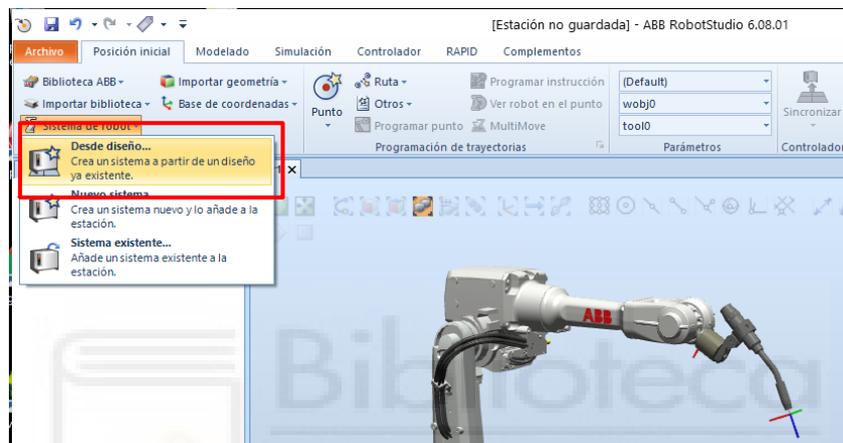


Figura 3.19 Creación Sistema del robot

Nos pedirá un nombre, así como una ubicación además de la versión de RobotWare

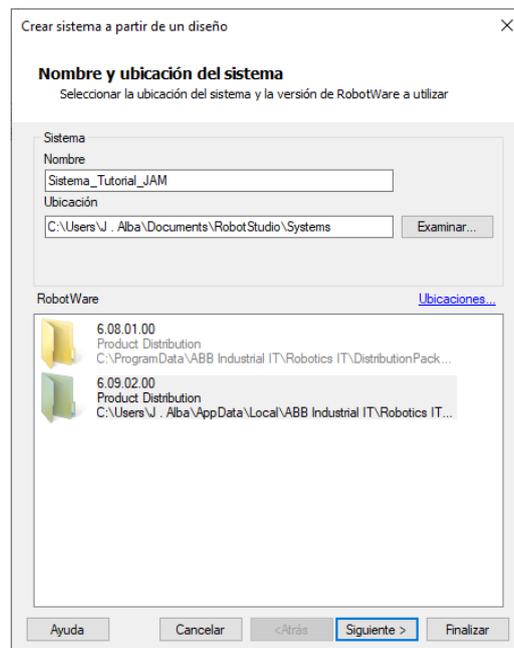


Figura 3.20 Configuración del sistema - Selección RobotWare

Seleccionamos nuestro robot y aceptamos.

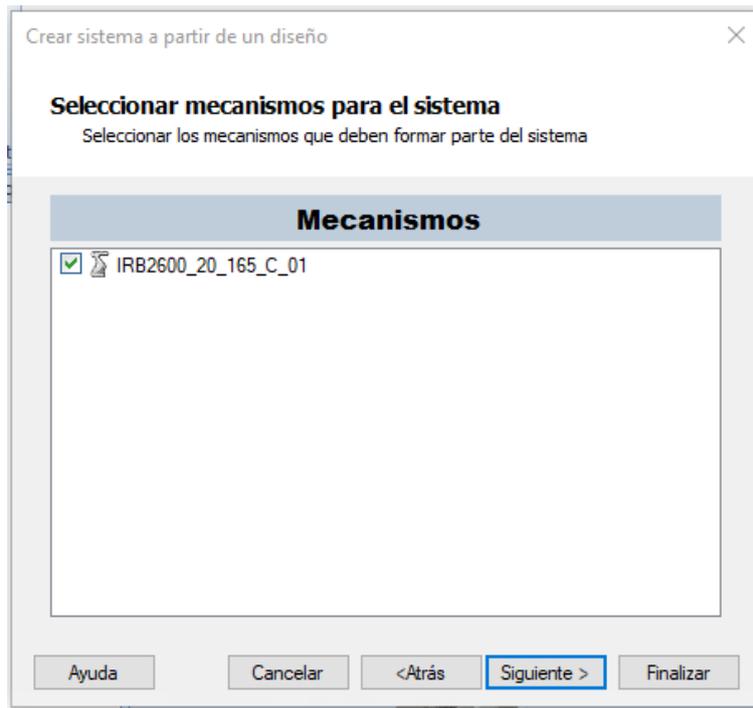


Figura 3.21 Configuración del sistema - Mecanismos

Se abre una ventana con el resumen del controlador configuramos, aprovechamos para acceder a opciones y seleccionar el castellano como idioma.

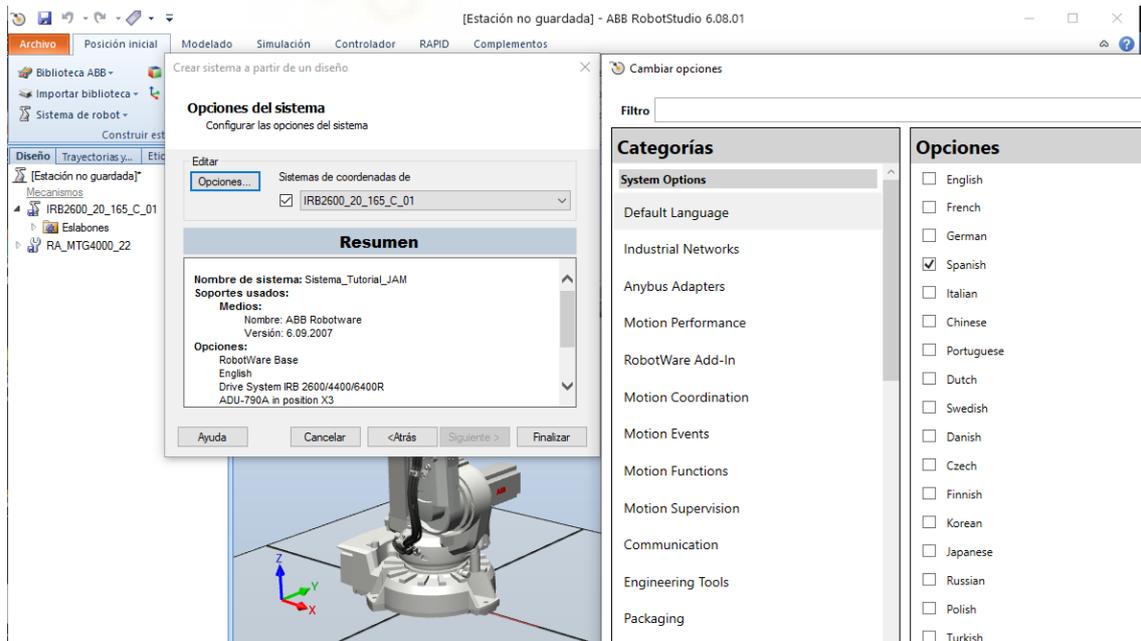


Figura 3.22 Configuración del sistema - Opciones - Idioma

Pulsamos finalizar y tras esperar unos segundos el programa a cargado correctamente el controlador como podemos comprobar en su estado en verde, situado en la esquina inferior derecha, además podemos comprobar que se han habilitado nuevas opciones como por ejemplo las correspondientes a “Mano alzada” en la pestaña “Posición inicial”.

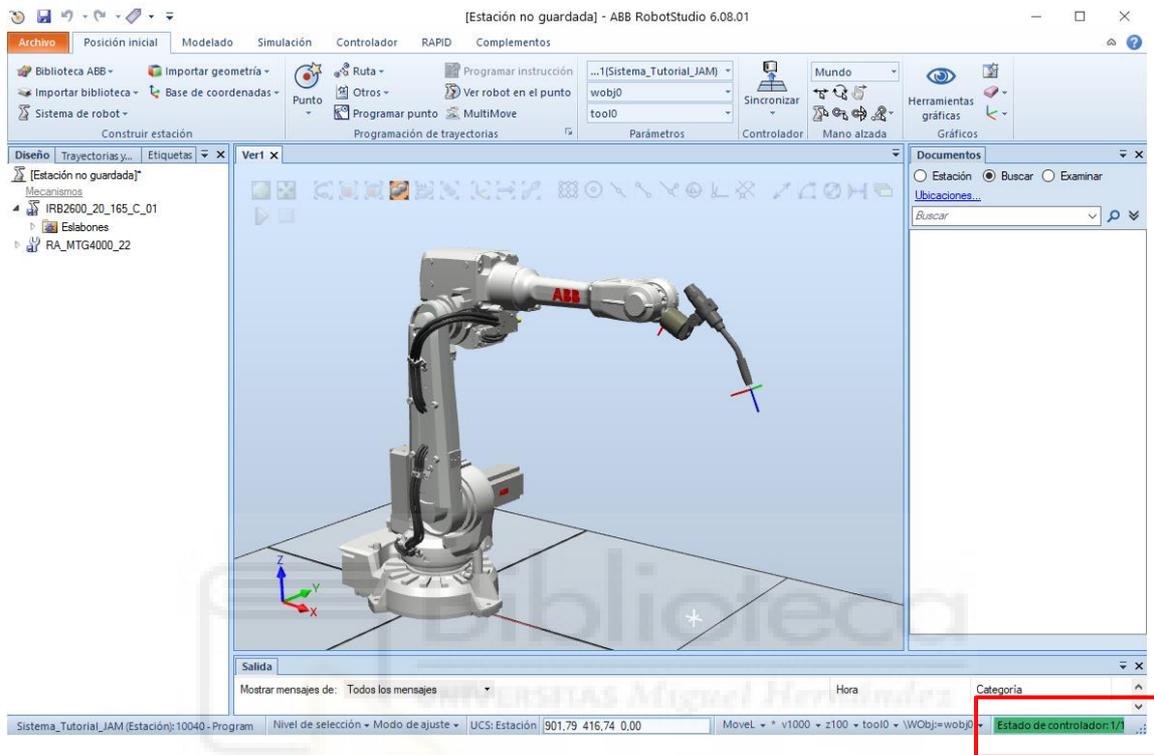


Figura 3.23 Controladora correctamente cargada

3.2.4 Creación de ejes de Bases de coordenadas, objetivos y trayectorias.

Para comenzar, crearemos un cubo sobre el que trabajar, para ello, y sin entrar en detalles nos dirigiremos a la pestaña “Modelado” y en “Sólido” seleccionaremos “tetraedro”.

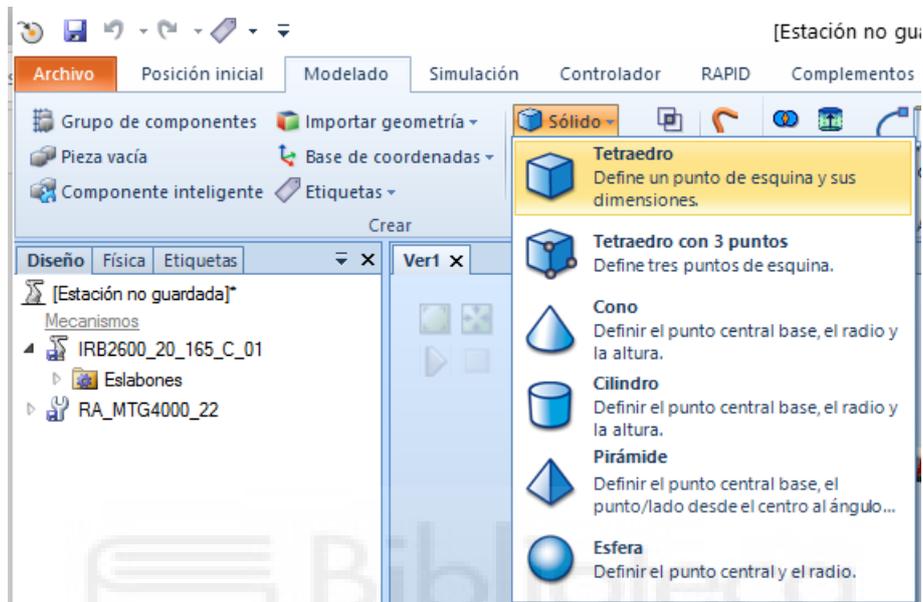


Figura 3.24 Creación de Sólidos – distintos Sólidos

Aparece la ventana “Crear tetraedro” donde solo restara indicar las dimensiones y posición para crear un cubo.

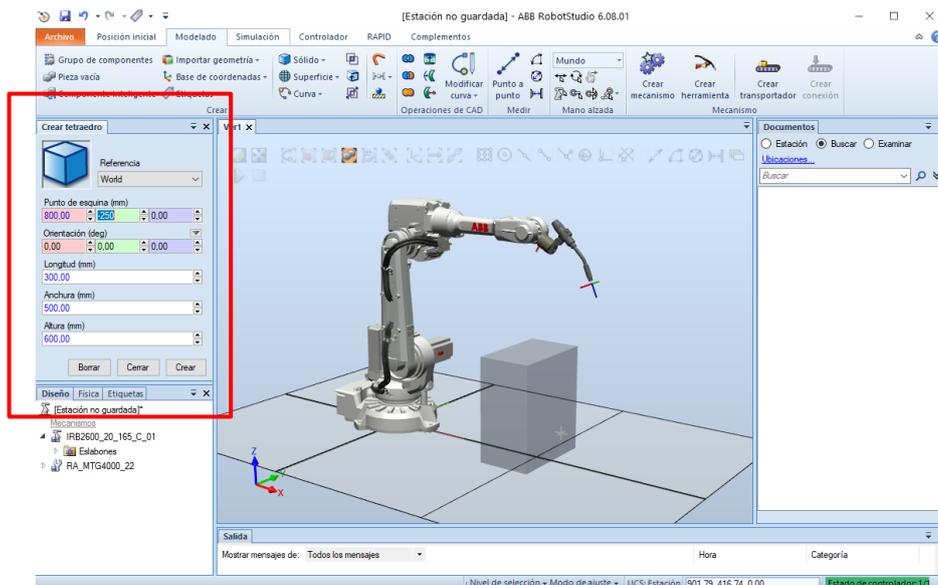


Figura 3.25 Edición de Sólidos

Para ilustrar la creación de centros de coordenadas, así como objetivos y sus distintas posibilidades, nos planteamos un escenario en el que deseamos llevar nuestra herramienta desde un punto inicial, hasta una esquina del cubo para posteriormente simular un proceso de soldadura en el contorno del cubo volviendo finalmente a la posición inicial.

Para ello, seguiremos un orden lógico en el que crearemos planos de trabajo, para continuar creando puntos objetivo sobre estos, para finalmente enlazarlos creando rutas o trayectorias.

3.2.4.1 Creación de Base de coordenadas.

Por defecto, RobotStudio trabaja con el centro de coordenadas del mundo, llamado “wob0”, esta base de coordenadas mundo es inamovible y el resto de sistemas estarán referidos en alguna instancia a este, esta no será la mejor manera de trabajar, ya que, por ejemplo, si referimos todos los puntos al mundo, y se modifica la posición de una pieza deberemos modificar toda la programación de nuevo, por otra parte, si previamente hemos asociado a los puntos su propio centro de referencias será mucho más simples las modificaciones necesarias.

Para crear una base de coordenadas, nos dirigimos a la pestaña posición inicial y en Base de coordenadas pulsamos sobre Creación de base de coordenadas.

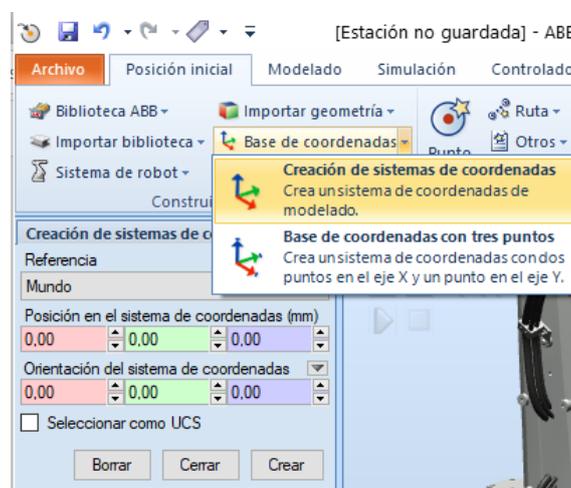


Figura 3.26 Posición inicial\ Creación de sistemas de coordenadas

En la ventana que nos aparece seleccionamos la referencia, posición y orientación. En este caso, queremos colocar en la esquina inferior del cubo que hemos creado anteriormente, una manera cómoda de hacer esto es activando la opción “Ajuste a final” en la cinta destinada a habilitar referencias en la ventana “Ver1” y tras hacer clic en alguno de los espacios habilitados para la edición de posición, pinchar en el punto deseado.

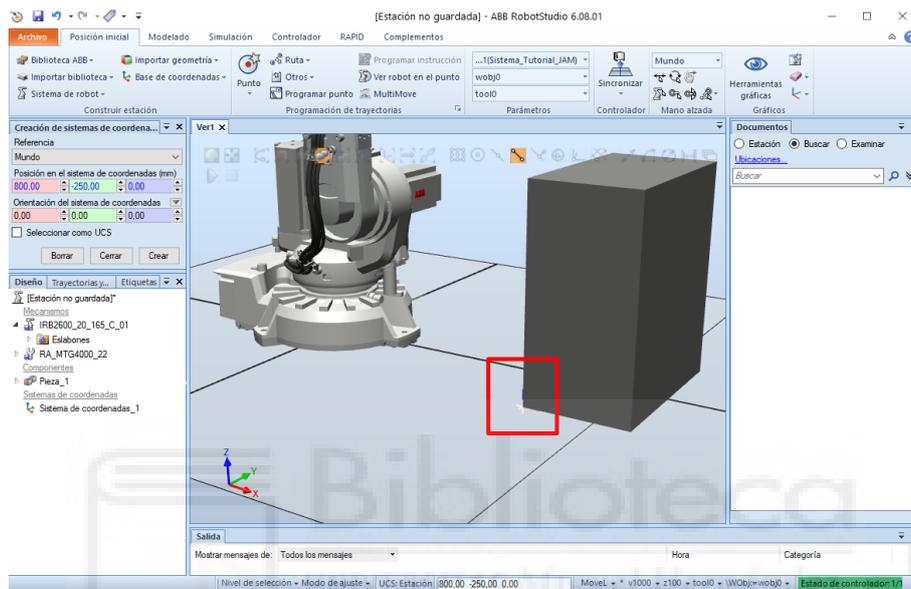


Figura 3.27 Detalle nuevo sistema de referencia

En la figura anterior, podemos comprobar cómo se ha creado el punto de referencia donde queríamos; para poder trabajar en RAPID deberemos convertirlo en un objeto de trabajo con botón derecho “Convertir en objeto de trabajo”, otra opción habría sido crearlo directamente como tal desde la pestaña “Posición inicial”, “Otros” y “Crear Objeto de trabajo”, donde indicaríamos el nombre y su ubicación de forma similar.

Si nos dirigimos a la pestaña de Trayectoria y puntos, y desplegamos nuestros Sistema>T_ROB1>&Objetos de trabajo y puntos podemos comprobar que ahora tenemos dos, wobj0 correspondiente a las coordenadas mundo y WOCUBO que es otro sistema de coordenadas referidas al primero.

Para visualizar los conceptos deseados nos será suficientes con estos dos sistemas de referencia creados.

3.2.4.2 Creación de objetivos.

Una vez creados los objetos de trabajo necesarios estamos en posición de crear puntos destino en referencia a estos.

Para comenzar crearemos un punto en la posición inicial, esta posición será el punto de partida y punto final de cada ciclo de operación; Este punto lo crearemos con referencia a las coordenadas mundo y con la herramienta seleccionada “weldgun”, seleccionados dichos parámetros, para crear este punto, en la pestaña “Posición inicial” en “Programación de trayectorias”, seleccionamos “Programar punto”, quedando así almacenada la posición en la que se encuentra el robot en este instante.

Podemos comprobar además en el árbol de trayectorias que se ha creado el punto “Target_10” que hemos renombrado como “HOME”.

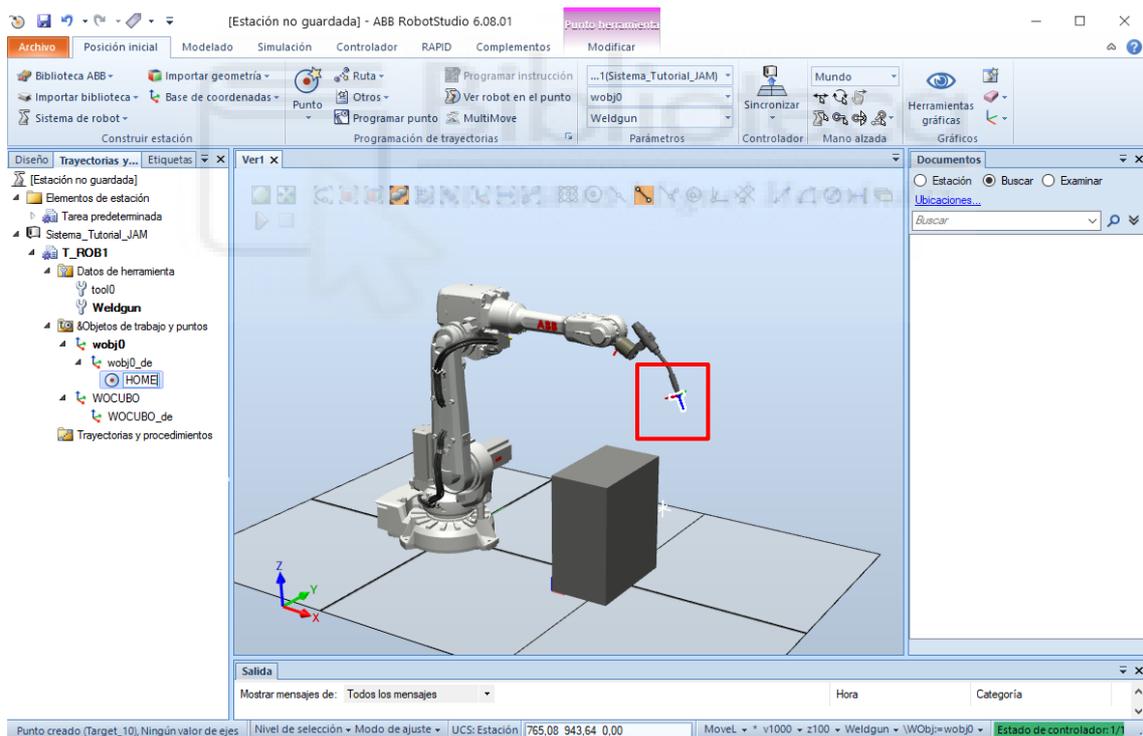


Figura 3.28 Punto HOME

A continuación, pretendemos crear 4 puntos en las esquinas superiores del cubo, pero ahora deseamos que estos puntos se encuentren referidos a el objeto de trabajo del cubo, por lo que lo modificamos en las opciones de parámetros.

Nos dirigimos a “Punto” y seleccionamos crear punto.

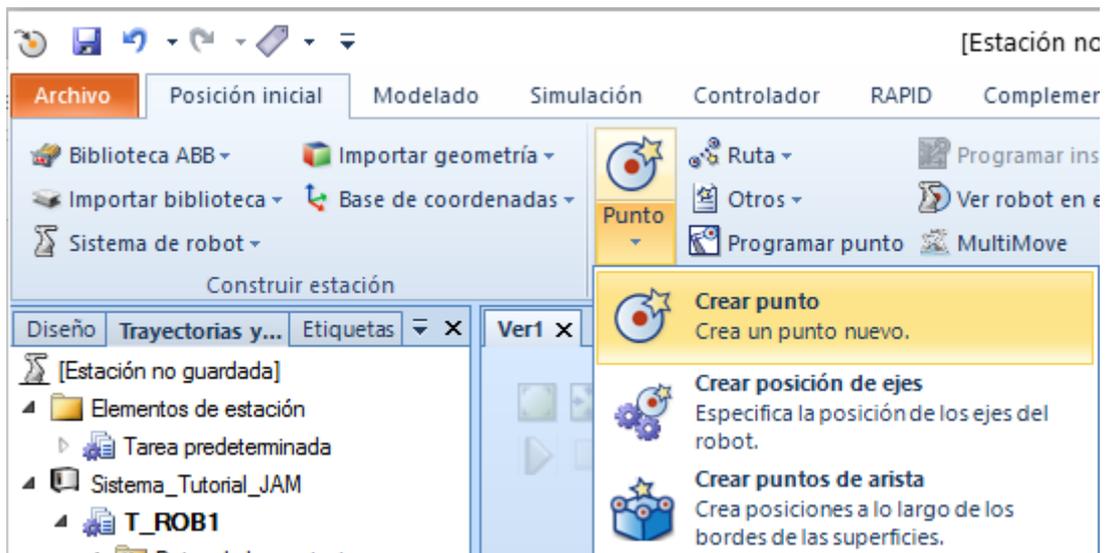


Figura 3.29 Crear punto

Con las opciones activas “Selección de objeto” y “Ajuste a final”, pinchamos sobre <Añadir nuevo> en la ventana izquierda y procedemos a seleccionar los puntos deseados en el cubo.

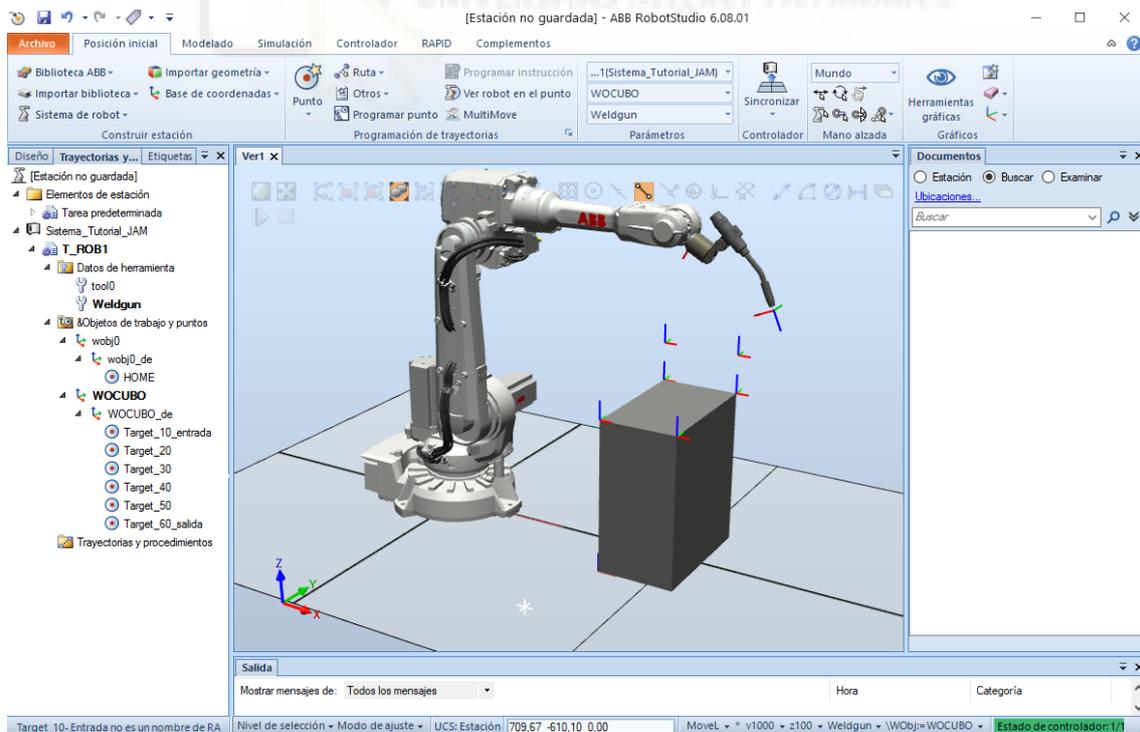


Figura 3.30 Objetivos en las esquinas

Como podemos observar en la figura anterior 3.30 ya se nos han creado los puntos, además se ha aprovechado para crear otro punto de aproximación para la entrada y la salida, no obstante, podemos observar que la orientación de la herramienta y los puntos no coincide, por lo que no serán alcanzables y de serlo seguirán siendo incorrectos para las operaciones deseadas.

Podemos modificar la posición y orientación de los puntos como cualquier otro objeto, para ello pinchando con el botón derecho sobre el punto a modificar, seleccionaremos “modificar punto” y “fijar posición” para orientarlo correctamente como se ha visto en casos anteriores.

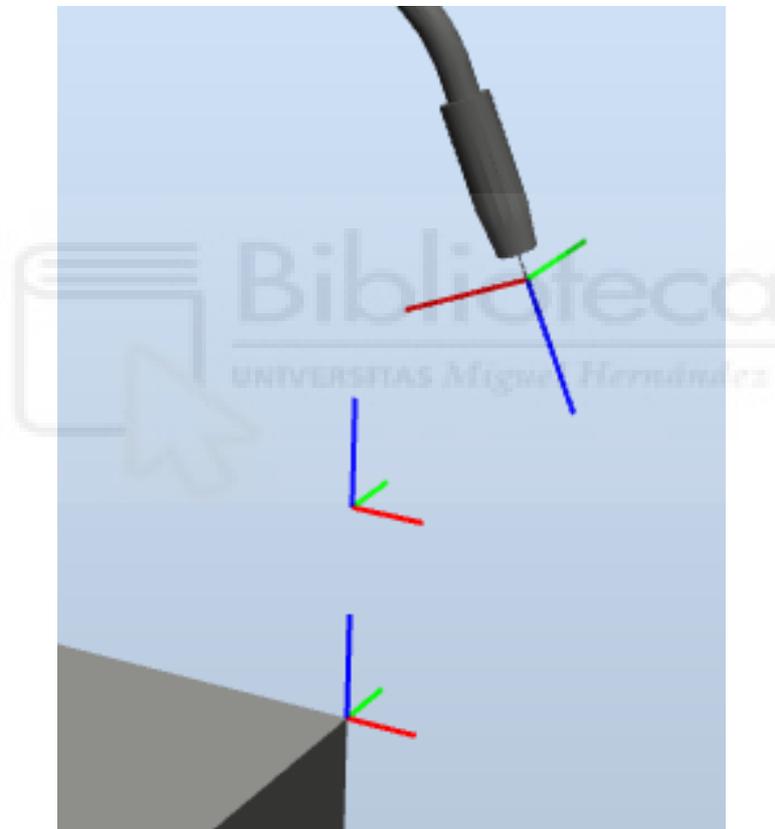


Figura 3.31 Orientación de objetivos VS TCP

Por proximidad visual se reoriento el Target_06_salida, se podría modificar el resto de puntos de forma individual, pero en este caso, con botón derecho sobre el punto modificado copiaremos su orientación, y posteriormente seleccionando el resto de puntos.

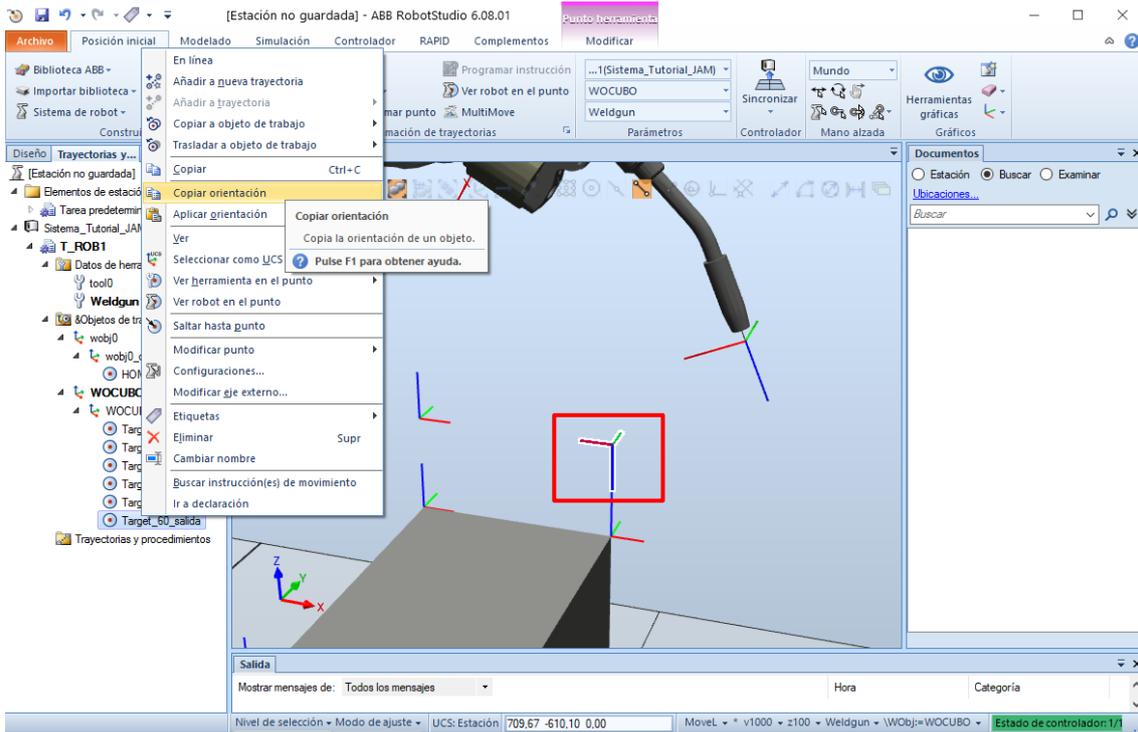


Figura 3.32 Copia del primer objetivo reorientado

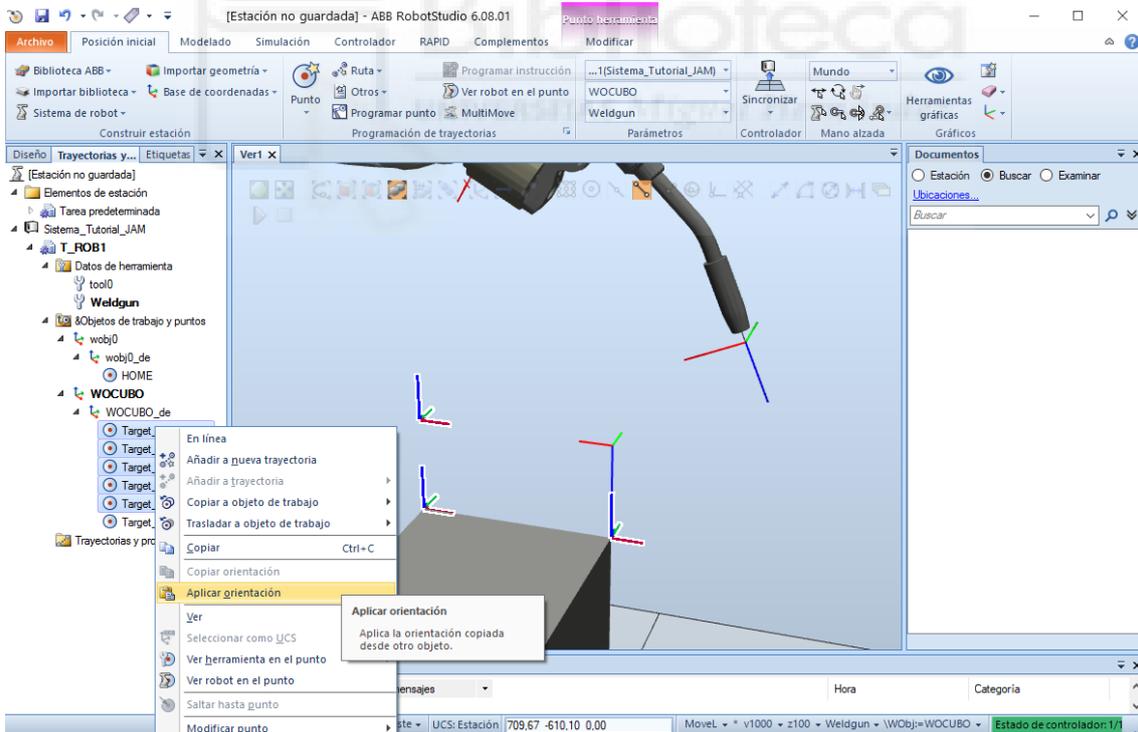


Figura 3.33 Aplicar orientación

Pudiendo comprobar como ahora todos los puntos tienen la misma orientación y de forma aparentemente correcta.

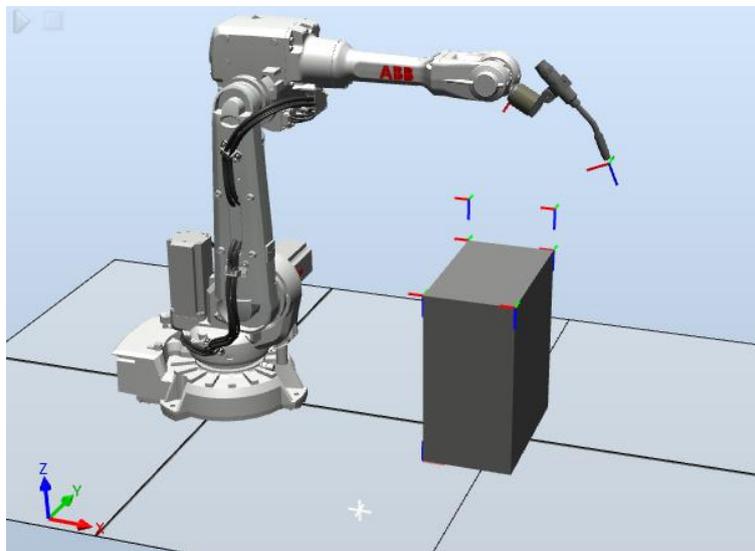


Figura 3.34 Orientaciones corregidas

Para verificar que los puntos son alcanzables en esta disposición por nuestro robot, tan solo deberemos pinchar con el botón derecho sobre el punto que queramos y accionar “ver robot en el punto”.

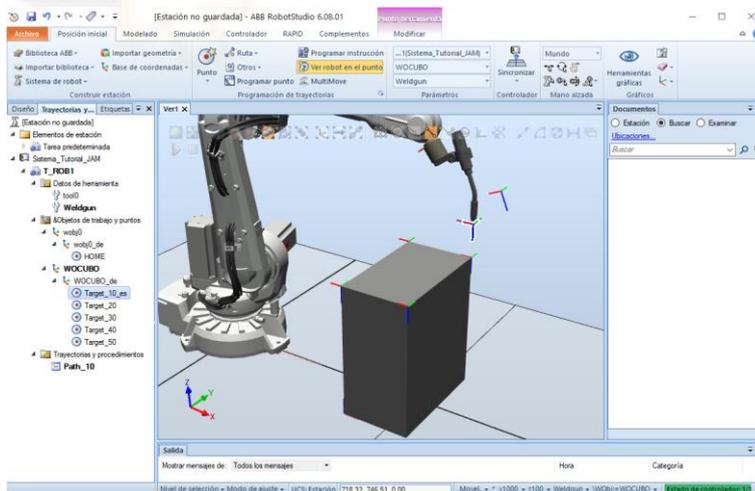


Figura 3.35 Alcanzabilidad Target_10

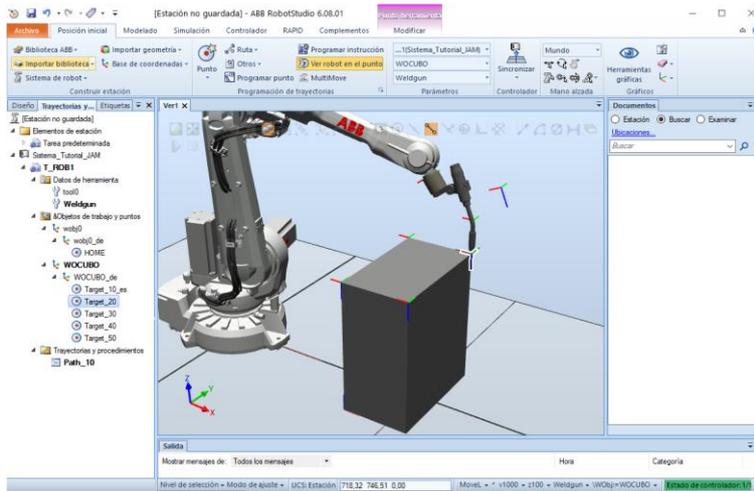


Figura 3.36 Alcanzabilidad Target_20

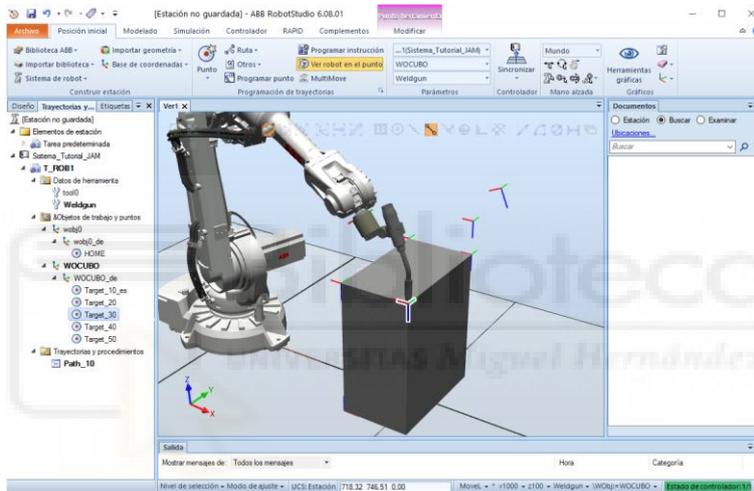


Figura 3.37 Alcanzabilidad Target_30

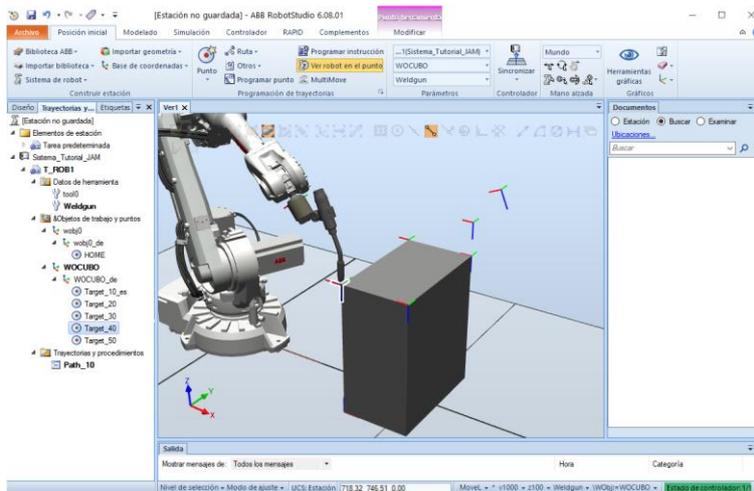


Figura 3.38 Alcanzabilidad Target_40

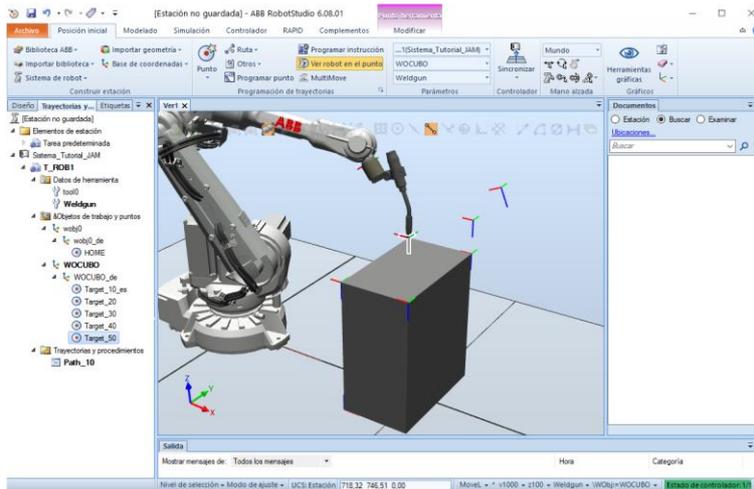


Figura 3.39 Alcanzabilidad Target_50

3.2.5 Creación de trayectorias

Una vez creados los puntos destino, estamos en disposición de crear trayectorias entre ellos, para ello, tenemos dos opciones, en la pestaña “Posición inicial”, desplegando las opciones de Ruta, podemos seleccionar “Trayectoria vacía” o “Trayectoria automática”, en el caso que nos ocupa seleccionaremos “Trayectoria vacía”, esto nos creara una ruta vacía llamada “Path_10” por defecto y orden numérico que podemos ver en el árbol de trayectorias y procedimientos.



Figura 3.40 Árbol de trayectorias y objetivos

Para crear la trayectoria a seguir por el robot, deberemos arrastrar los puntos deseados al path por el orden de ejecución requerido.

Arrastrando los puntos y ordenándolos podemos ver la trayectoria generada automáticamente.

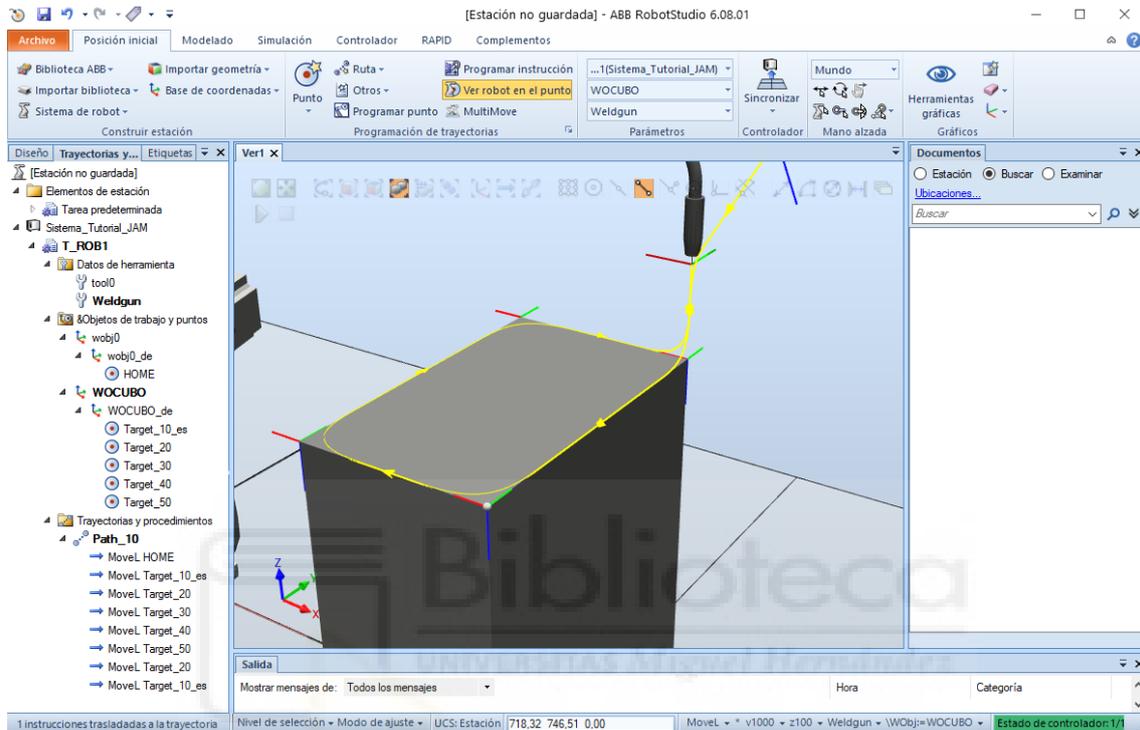


Figura 3.41 Path_10 automática

Podemos observar que la trayectoria no se ajusta mucho a lo que podríamos esperar, esto se debe a la configuración de la plantilla, si observamos la parte inferior de la pantalla podemos observar que entre otros parámetros, existe un parámetro de aproximación z establecido en 100.



Figura 3.42 Parámetros

El valor correcto para mayor precisión será “fine”, esto lo podemos modificar de forma manual e individual para cada instrucción o modificarlas en grupos, para ello seleccionaremos las instrucciones deseadas y con botón derecho seleccionaremos “Modificar instrucción” desplegándose el panel de opciones pertinente, en este panel, entre otras, destacan tres opciones, por un lado “Tipo de movimiento”, RobotStudio

distingue tres movimientos fundamentales, L, Linear, de manera que recorrerá la distancia entre dos puntos de forma linear, C, Circle, define trayectorias circulares a partir de tres puntos y, J, Joint, realiza la transición entre dos puntos de la forma que le resulte más “cómoda” a la trayectoria, es decir, la que requiera menos gasto computacional.

Las otras dos opciones de importancia son la velocidad V, y la zona, Z, en este caso modificamos la Z de todas las instrucciones dejándolas en “fine” y aplicamos.

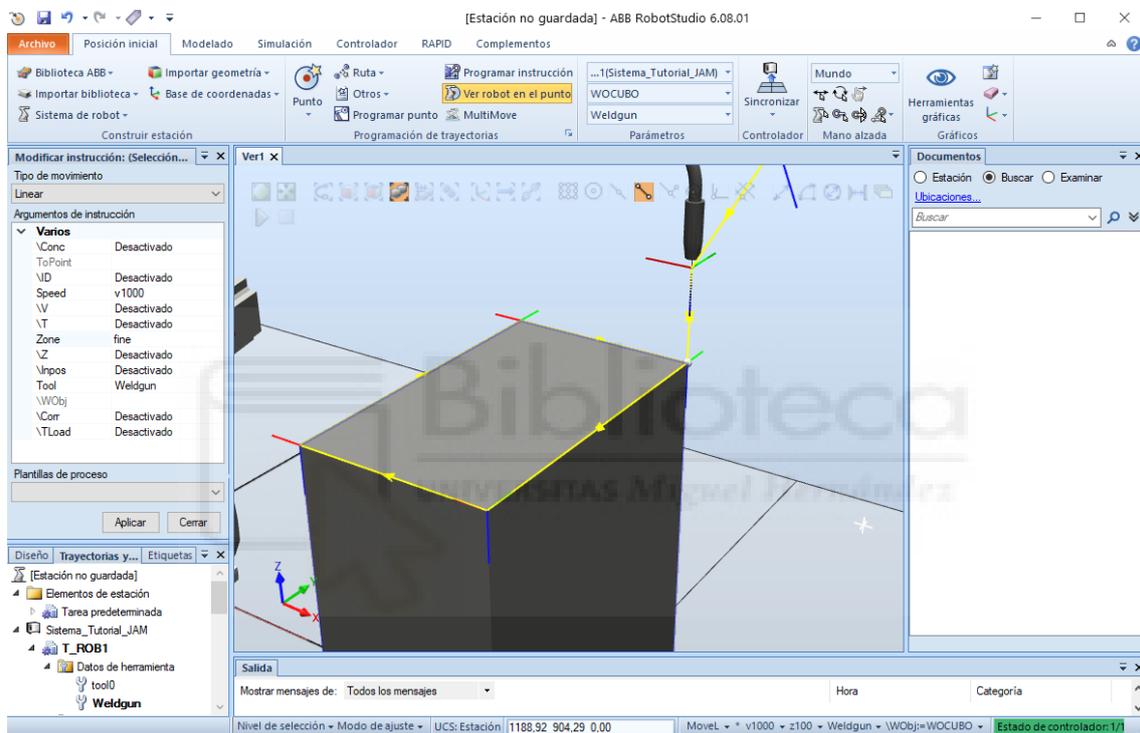


Figura 3.43 Trayectoria Path_10 con ajuste Zone "Fine"

Ahora se aprecia que el robot cumple con la trayectoria alcanzando de forma precisa todos los puntos.

Tras cargar el path en la configuración de la simulación y pulsando “play”, podemos comprobar la correcta ejecución de la secuencia.

3.2.6 Transportadores

Tras haber visto la aplicación de objetivos y trayectorias, es interesante ver la opción “Crear transportador”, hablamos de una potente herramienta que permite operar sobre los objetos en movimiento [2].

Debemos tener presente que para poder utilizarlo es necesario cargar las opciones “606-1 Transportador tracking” y “709-1 DeviceNet”.

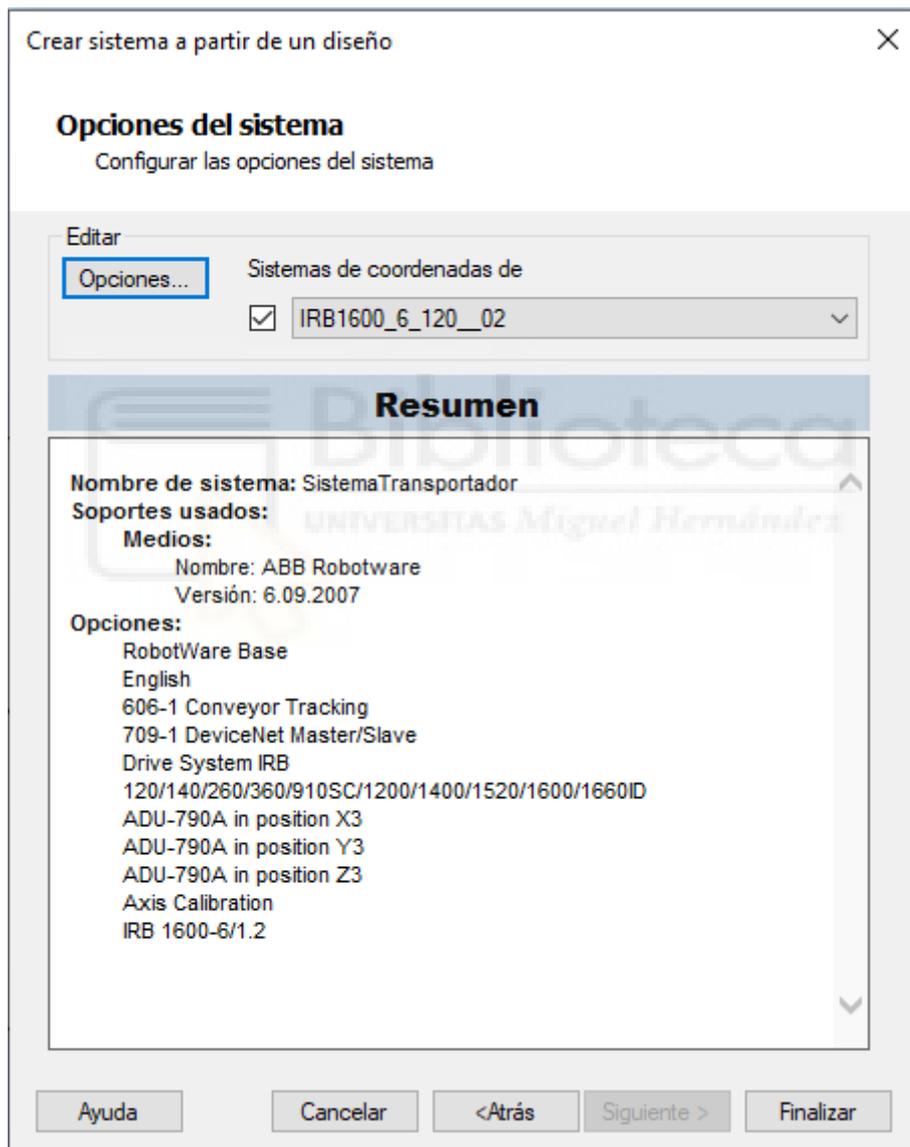


Figura 3.44 Transportador - Opciones de sistema

La metodología para la implementación de un transportador y un robot, tras la creación de un sistema y adición de un robot y herramienta de manera normal, se muestra a continuación:

- Creación y posterior posicionamiento de una representación simple del transportador.

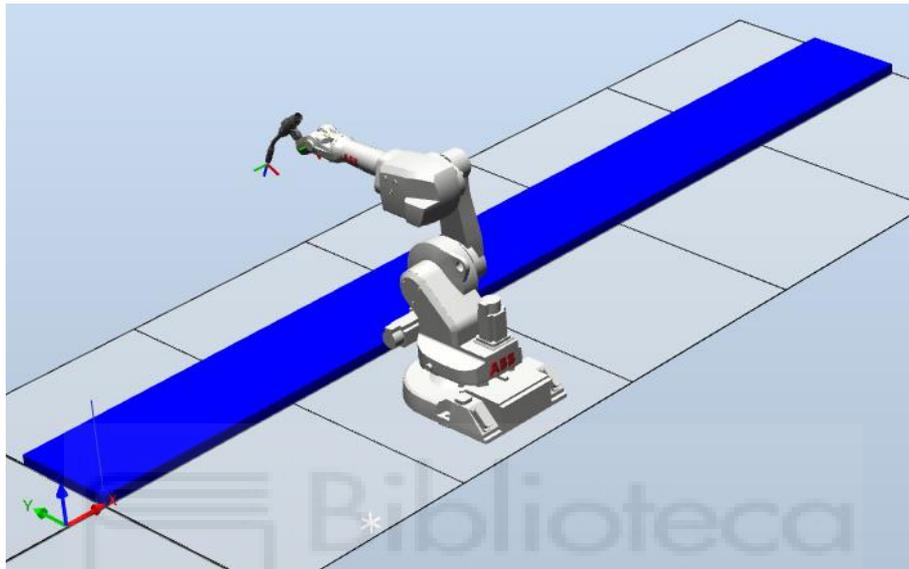


Figura 3.45 Creación modelo simple de Transportador

- Seleccionar la opción “Crear transportador” y seleccionar el modelo anterior en la propiedad “Transportador y geometría”, seleccionando además el tipo “Lineal” y definiendo la longitud de este en “Longitud del transportador”.

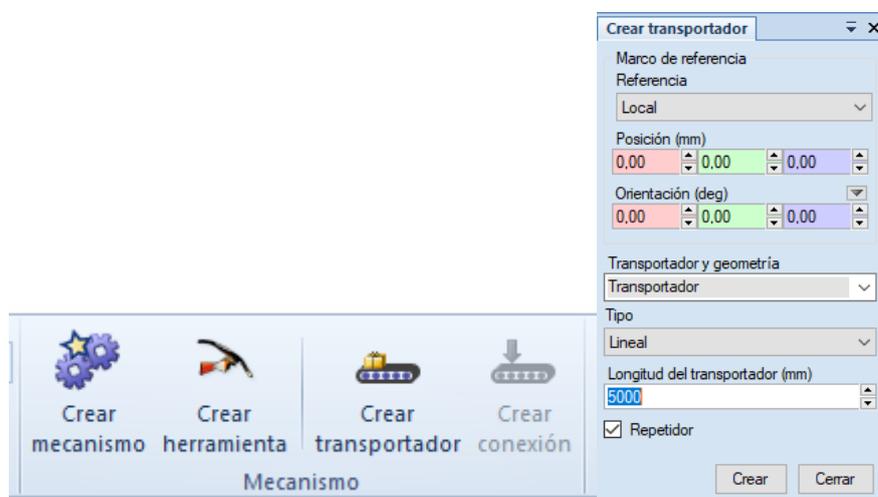


Figura 3.46 Crear Transportador

- Presionando el botón derecho del ratón sobre el transportador creado, seleccionamos “Crear conexión” y configuramos el espacio de trabajo del robot sobre la cinta.

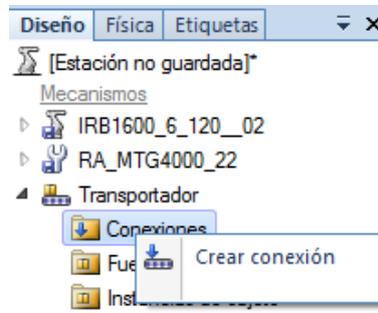


Figura 3.47 Crear conexión

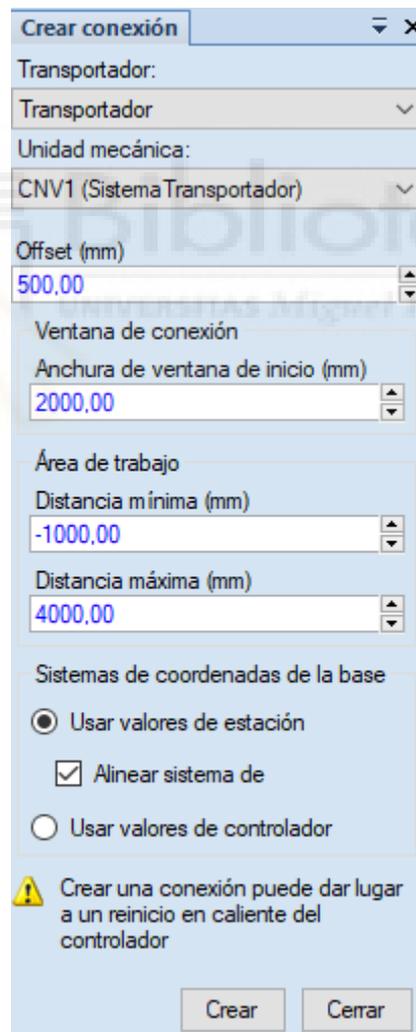


Figura 3.48 Editar conexión

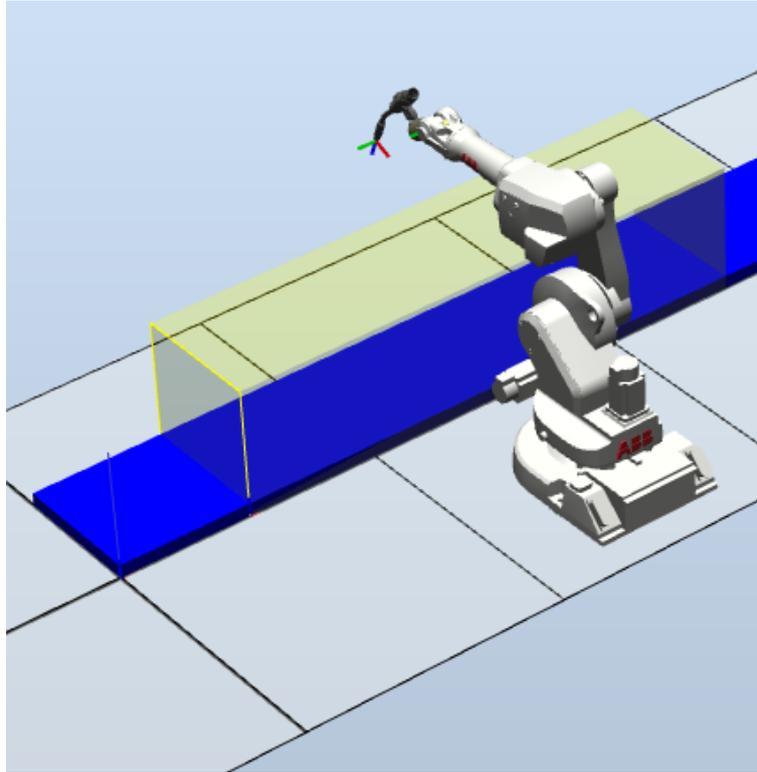


Figura 3.49 Conexión creada

- Añadimos o diseñamos la pieza sobre la que deseamos operar.

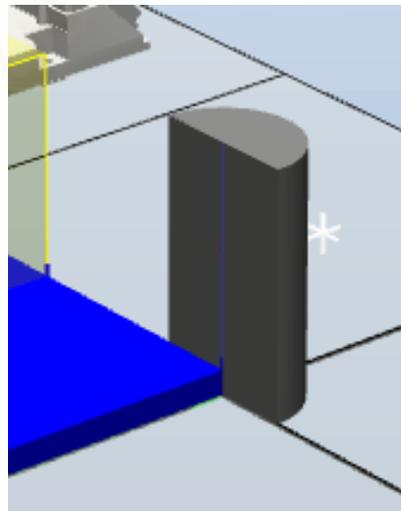
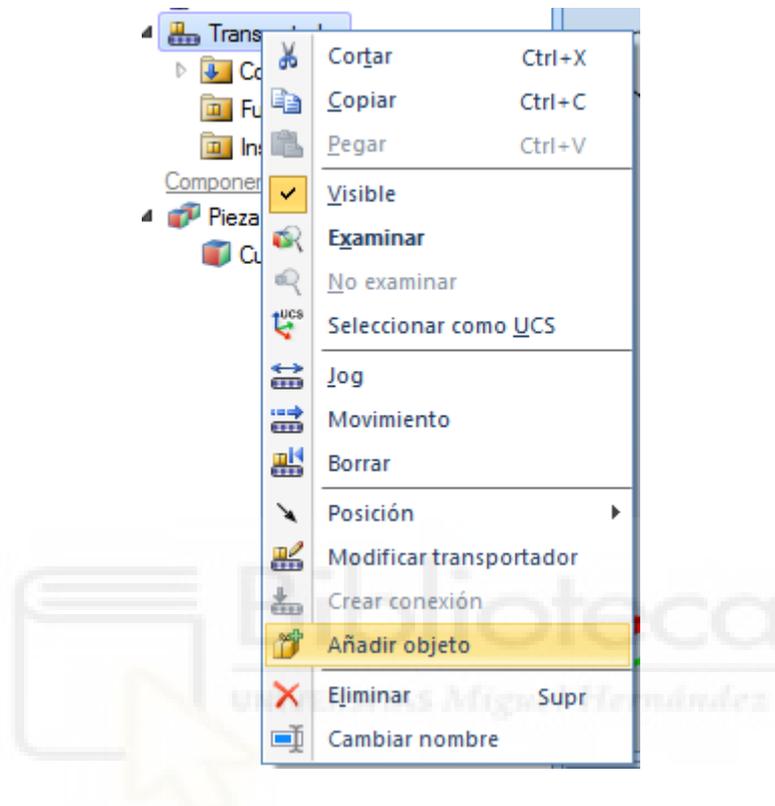


Figura 3.50 Pieza de trabajo

- Volviendo con el botón derecho sobre el transportador, presionamos “Añadir objeto”, ahora seleccionaremos la pieza de trabajo y configuraremos parámetros como la posición de inserción de las piezas y la separación entre piezas consecutivas.



Objeto del transportador		
Pieza		
Pieza_4		
Separación (mm)		
1000,00		
Compensar posición (mm)		
0,00	300,00	50
Compensar orientación (deg)		
0,00	0,00	0,00
Crear		
Cerrar		

Figura 3.51 Objeto del transportador

- Dentro del árbol del transportador, nos dirigimos a “Fuente de objeto”, y, con el botón derecho seleccionamos “Colocar en el transportador” y “Adjuntar objeto de trabajo”.

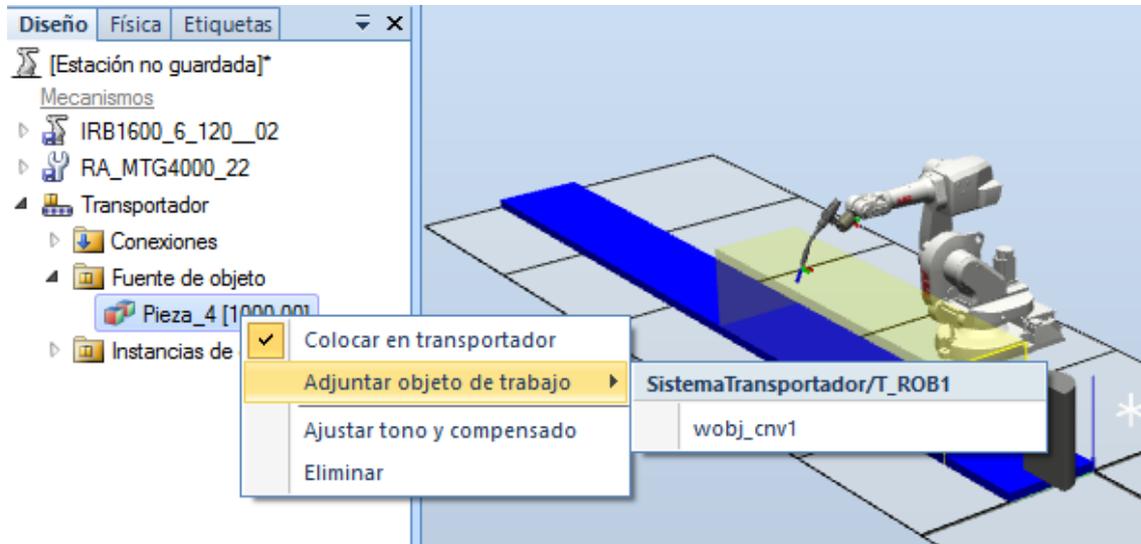


Figura 3.52 Adjuntar objeto de trabajo

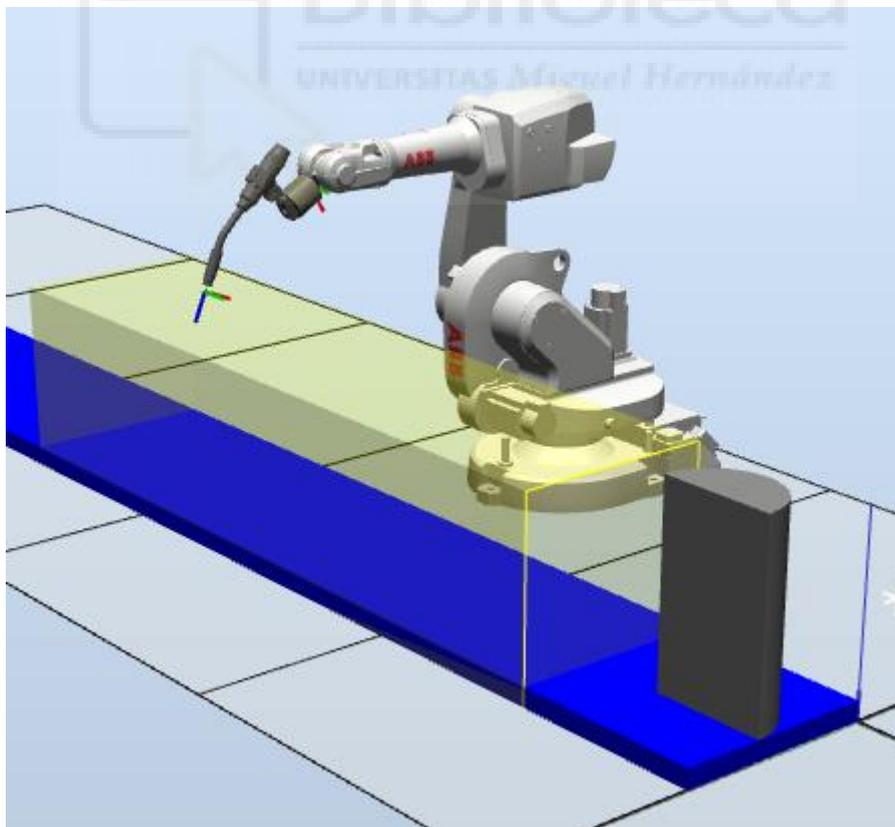


Figura 3.53 Objeto de trabajo adjuntado

- Utilizamos la opción “Jog” para arrastrar la pieza hasta la zona de seguimiento anteriormente creada, es decir, donde el robot comenzara a operar.

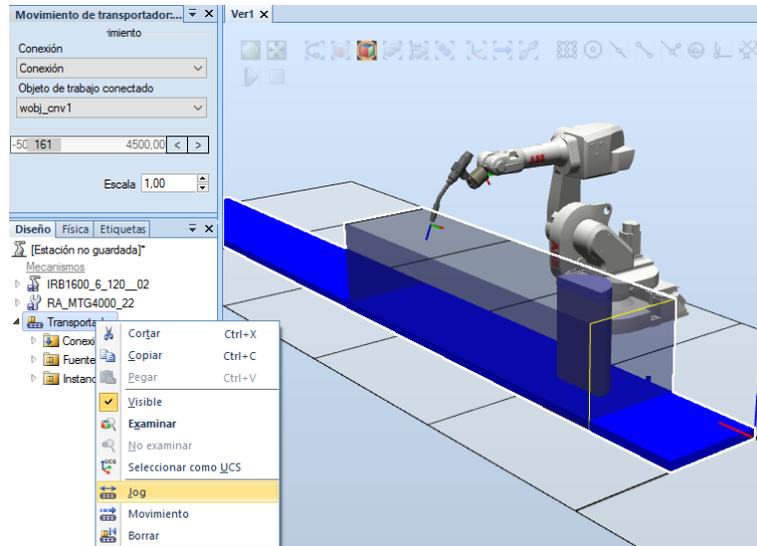


Figura 3.54 posicionamiento "Jog"

- Diseñamos los objetivos y trayectoria conforme a lo visto en apartados anteriores.

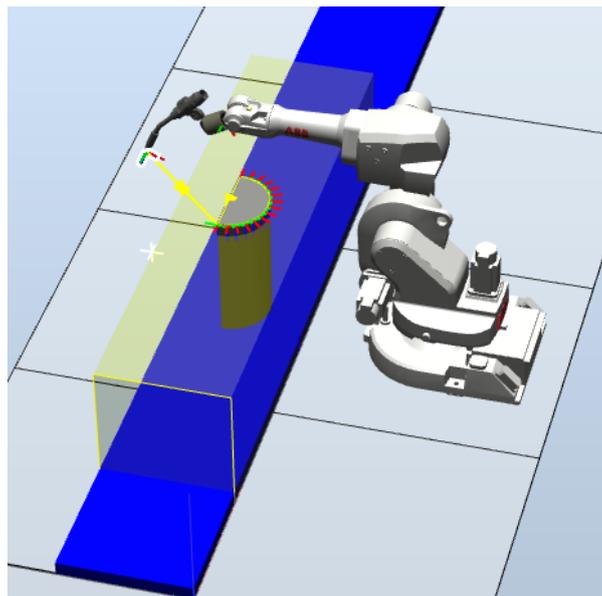


Figura 3.55 Trayectorias transportador

- Insertamos las siguientes instrucciones:
 - ActUnit CNV1 Responsable de la activación de la unidad mecánica
 - ConfL \Off
 - WaitWobjt wobj_cnv1 Espera la entrada del objeto de trabajo en el área de trabajo para operar.
 - DropWobjt wobj_cnv1 Suelta el objeto de trabajo

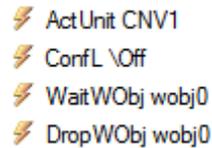


Figura 3.56 Instrucciones de trabajo

- Reordenamos las instrucciones quedando:

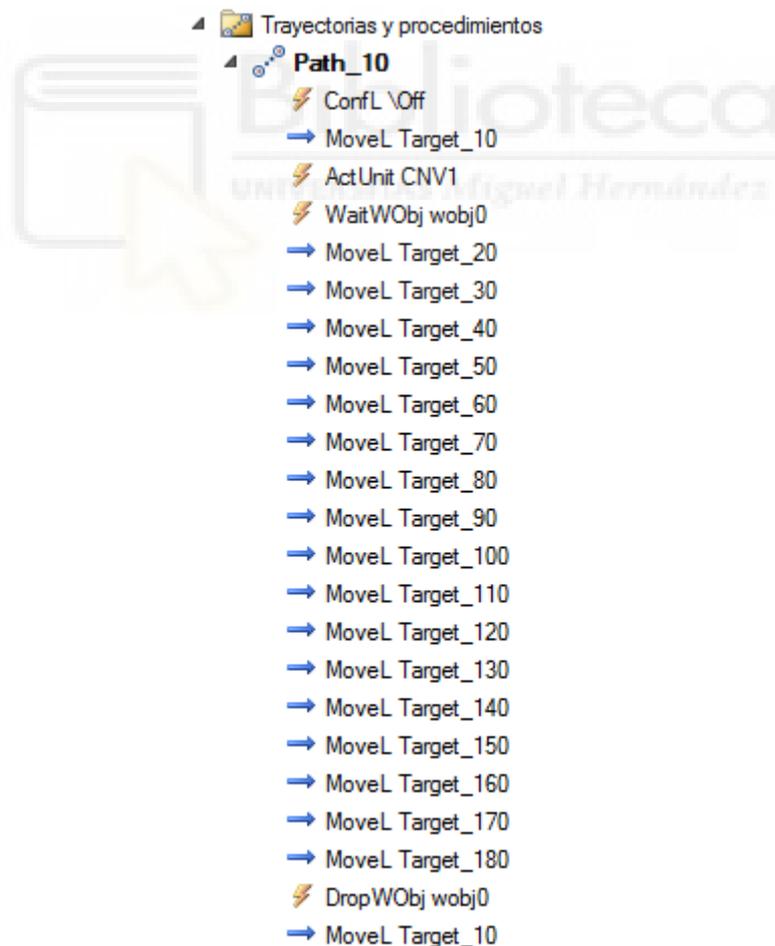


Figura 3.57 Secuencia ordenada

- Sincronizamos con RAPID.
- Simulamos.

Mencionar que esta herramienta es realmente potente y sería de gran interés adaptarla a el presente proyecto, no obstante, al menos en la instalación presente, tiene una serie de limitaciones que impide su empleo.

- Solo puede trabajar con una pieza, lo que limita la flexibilidad buscada.
- Trabaja a intervalos regulares preconfigurados.
- No se ha encontrado forma de crear señales para trabajar con el sistema general, solo una entrada para “Parada”.

Por estos motivos se descarta su implementación.

3.2.7 Introducción a RAPID

RAPID es un lenguaje de programación textual de alto nivel desarrollado por la empresa ABB “Robotics Application Programming Interactive Dialogue”.

Una aplicación RAPID consta de un programa y una serie de módulos del sistema.

El programa es una secuencia de instrucciones que controlan el robot y en general consta de tres partes:

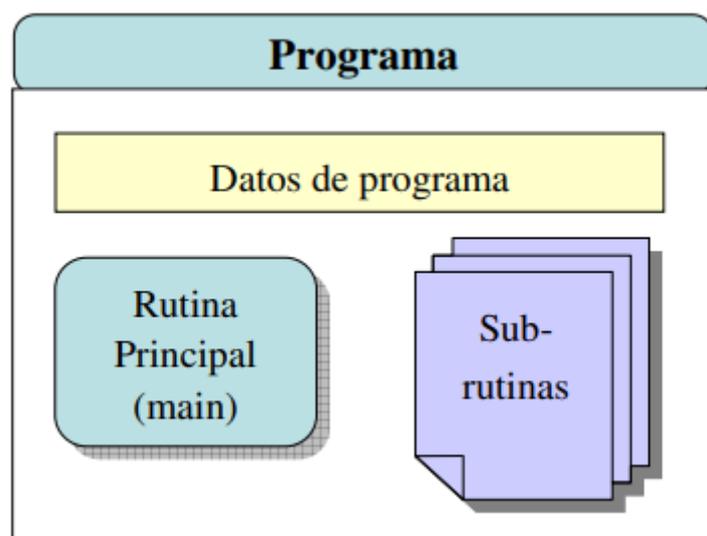


Figura 3.58 Esquema básico de programa

Una rutina principal (main), responsable de iniciar la ejecución.

Un conjunto de subrutinas que dividirán el programa en partes más pequeñas con el objetivo de obtener un programa modular.

Los datos del programa definirán posiciones, valores numéricos, sistemas de coordenadas...

El trabajo entre RAPID y RobotStudio es bidireccional, es decir, siempre que hagamos un cambio en nuestra estación y queramos reflejarlo en RAPID o viceversa deberemos sincronizar, de ahí, que si pinchamos sobre el icono “sincronizar” encontraremos las dos soluciones, sincronizar con RAPID y sincronizar con la estación.

Tras la sincronización, todos los datos, ya sean de herramientas, puntos o trayectorias, han sido volcados programa, si accedemos a la pestaña Rapid, podemos desarrollar el árbol del controlador y visualizar los módulos correspondientes al programa, CalibData y Module1.

El módulo del programa “Module1” incluye el main, correspondiente a el programa principal y es posible realizar cambios.

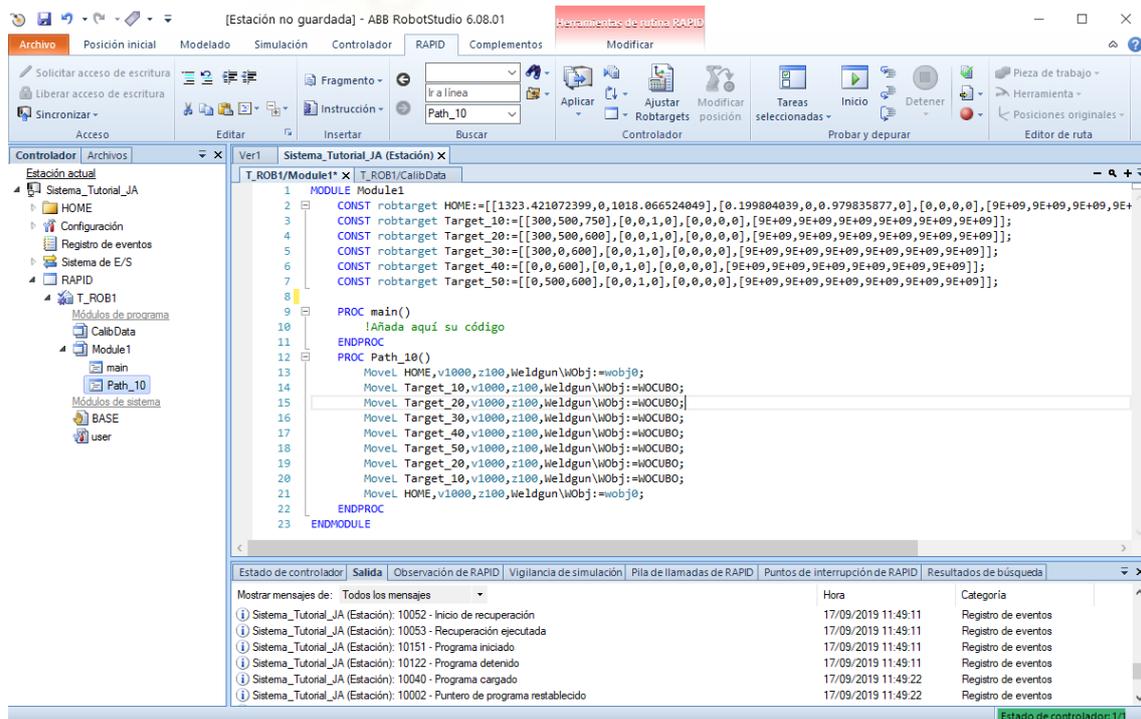


Figura 3.59 Ejemplo Module1

El esquema clásico de RAPID está formado por un módulo “Module1” que engloba todos los procedimientos de nuestro programa, este programa siempre comienza con la instrucción “MODULE Module1” y finalizar con “ENDMODULE”, por su parte, los distintos programas serán iniciados mediante “PROC Nombre programa” y finalizado con “ENDPROC”

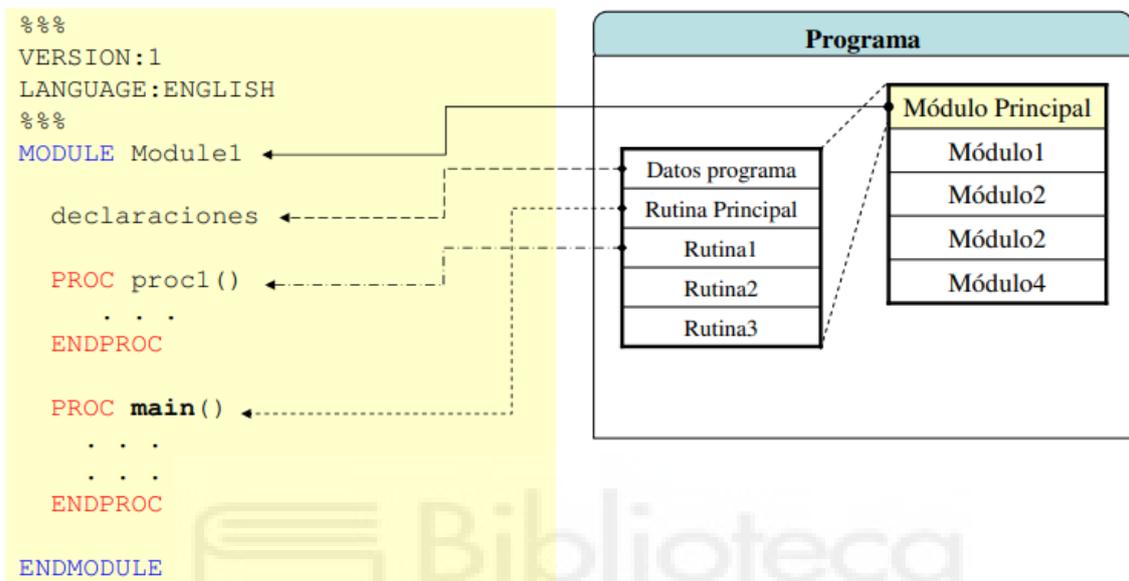


Figura 3.60 Esquema básico RAPID

```

MODULE Module1

CONST robtarget HOME:=[[1323.421072399,0,1018.066524049],[0.199804039,0,0.979835877,0],
[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_10:=[[300,500,750],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E
+09]];

CONST robtargetTarget_20:=[[300,500,600],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E
+09]];

CONST robtarget Target_30:=[[300,0,600],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_40:=[[0,0,600],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_50:=[[0,500,600],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PROC main()

  Path_10;

ENDPROC
  
```

```

PROC Path_10()

  MoveL HOME,v1000,z100,Weldgun\WObj:=wobj0;

  MoveL Target_10,v1000,z100,Weldgun\WObj:=WOCUBO;

  MoveL Target_20,v1000,z100,Weldgun\WObj:=WOCUBO;

  MoveL Target_30,v1000,z100,Weldgun\WObj:=WOCUBO;

  MoveL Target_40,v1000,z100,Weldgun\WObj:=WOCUBO;

  MoveL Target_50,v1000,z100,Weldgun\WObj:=WOCUBO;

  MoveL Target_20,v1000,z100,Weldgun\WObj:=WOCUBO;

  MoveL Target_10,v1000,z100,Weldgun\WObj:=WOCUBO;

  MoveL HOME,v1000,z100,Weldgun\WObj:=wobj0;

ENDPROC

ENDMODULE

```

Por ejemplo, fijándonos en el código anterior, podemos observar la estructura de las instrucciones de movimiento "Move_", estando compuestas por el tipo de movimiento, el punto de destino, velocidad, precisión y herramienta utilizada por el robot.

Podemos observar, como el valor de precisión es z100, haciendo memoria y observando la figura siguiente, esto resultaba en una trayectoria que no se correspondía con la tarea deseada.

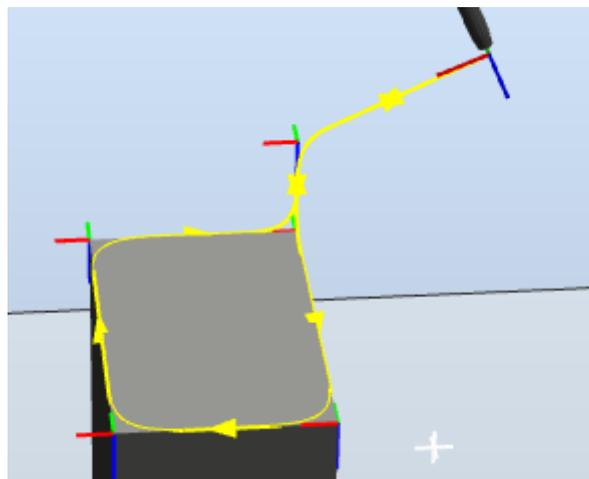


Figura 3.61 Trayectoria NO deseada

Anteriormente modificamos las instrucciones desde el árbol de trayectorias, no obstante, ahora realizaremos lo mismo con RAPID a modo de ejemplo, para ello, cambiaremos los valores “z100” por “fine”, además, se ha aprovechado para modificar las secciones de trayectoria HOME-Target_10 y Target_10-HOME de un movimiento lineal a uno joint, simplemente a modo de ejemplo, ya que en este ejemplo tan sencillo el coste computacional no es relevante; tras realizar estos cambios procedemos a sincronizar con la estación.

```
PROC Path_10()  
  
    MoveL HOME,v1000,z100,Weldgun\WObj:=wobj0;  
  
    MoveL Target_10,v1000,z100,Weldgun\WObj:=WOCUBO;  
  
    MoveL Target_20,v1000,fine,Weldgun\WObj:=WOCUBO;  
  
    MoveL Target_30,v1000,fine,Weldgun\WObj:=WOCUBO;  
  
    MoveL Target_40,v1000,fine,Weldgun\WObj:=WOCUBO;  
  
    MoveL Target_50,v1000,fine,Weldgun\WObj:=WOCUBO;  
  
    MoveL Target_20,v1000,fine,Weldgun\WObj:=WOCUBO;  
  
    MoveL Target_10,v1000,z100,Weldgun\WObj:=WOCUBO;  
  
    MoveL HOME,v1000,z100,Weldgun\WObj:=wobj0;  
  
ENDPROC
```

Modificando el código de esta manera y sincronizando con la estación obtenemos la trayectoria deseada, pero esta vez mediante programación RAPID.

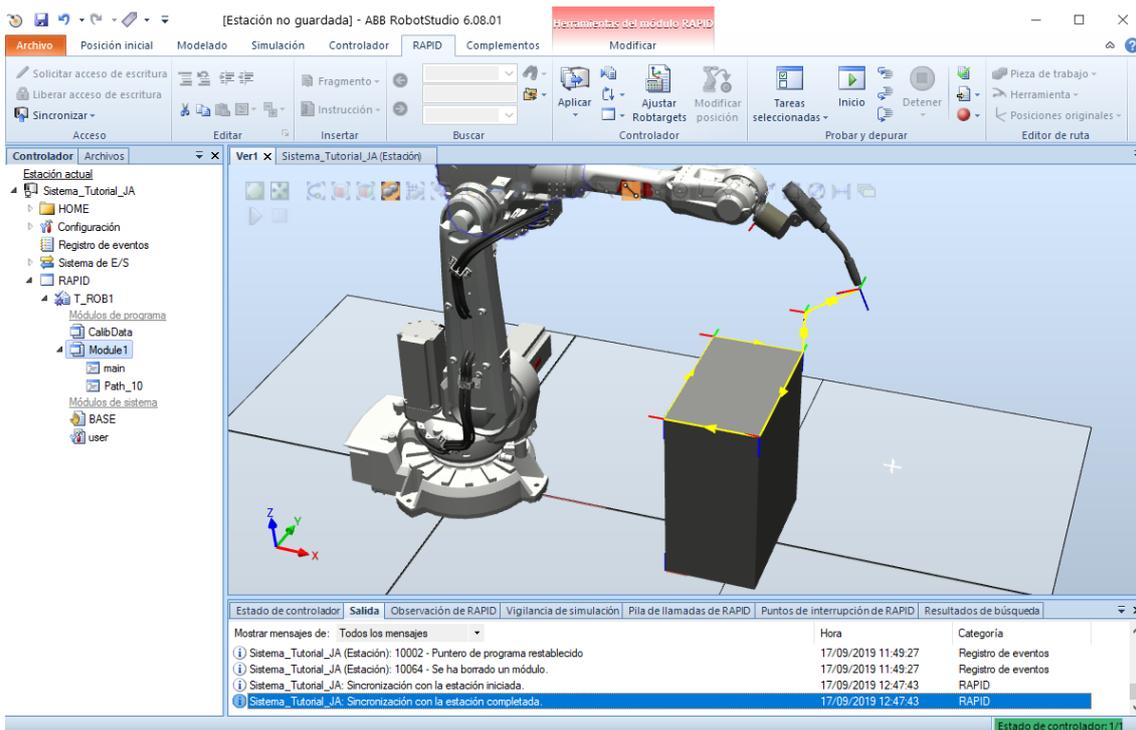


Figura 3.62 Trayectoria corregida en RAPID

Antes de continuar se hace un breve repaso a conceptos fundamentales en la programación RAPID [1].

- **Instrucciones para la utilización de entradas y salidas.**

Fundamentalmente, tratamos de identificar una entrada y fijar una salida en consecuencia.

La instrucción principal para comprobar una entrada es “WaitDI”, haciendo que el robot espere a que la señal alcance un valor, por ejemplo: (WaitDi Pieza,1;).

Para fijar el valor de las salidas:

Set	Fija el valor de una salida digital a 1.
Reset	Fija el valor de una salida digital a 0.
SetDO	Fija una salida digital a un valor simbólico.
SetAO	Fija el valor de una salida analógica.

- **Instrucciones de control de flujo de ejecución.**

Típicas instrucciones condicionales y de control comunes en otros lenguajes de programación:

IF THEN	Ejecuta una serie de instrucciones si se cumple una determinada condición.
FOR	Repite una secuencia de instrucciones un número de veces determinado.
WHILE	Repite una secuencia de instrucciones mientras se cumpla la condición.
TEST/CASE	Ejecuta diferentes instrucciones en función de un dato. (Switch/case)
GOTO	Salto incondicional a un punto del programa.

- **Variables y expresiones**

El lenguaje RAPID, permite trabajar con distintos tipos de variables con el fin de poder crear distintas expresiones aritméticas o lógicas.

Algunas de las variables más importantes son:

ConfData	Configura los ejes del robot	Pos	Posición X, Y y Z
JointTarget	Posición de los ejes	Robjoint	Posición de los ejes
LoadData	Carga en la muñeca del robot	RobTarget	Datos de posición
MotSetData	Parámetros de movimiento	StopPointData	Datos de punto de paro
Num	Valores numéricos	ToolData	Características de la herramienta
Orient	Orientación de un elemento	ZondData	Especifica como debe terminarse

3.3 Diseño y programación de una estación

3.3.1 Creación y diseño de Smart Components

Los componentes inteligentes o Smart Components (SC) son componentes asociados a los distintos elementos de la estación (robots, sólidos, piezas) caracterizados por un comportamiento asociado a señales y propiedades del sistema [2].

Para acceder al menú de Smart Components, deberemos dirigirnos a la pestaña “Modelado” y pulsar sobre la opción “Componente inteligente”.

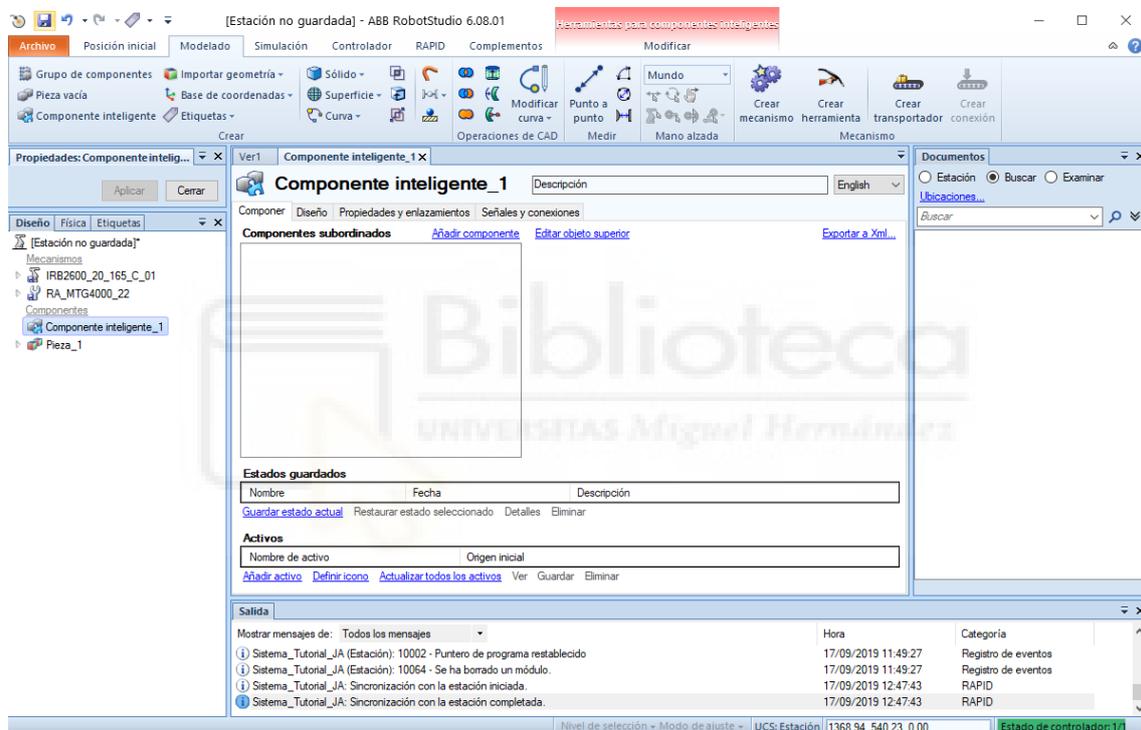


Figura 3.63 Componente inteligente (SC)

Si presionamos sobre “Añadir componente” en la subpestaña “Componer” se despliega una lista con los elementos utilizados recientemente y en la parte inferior todos los disponibles (Fig 3.63).

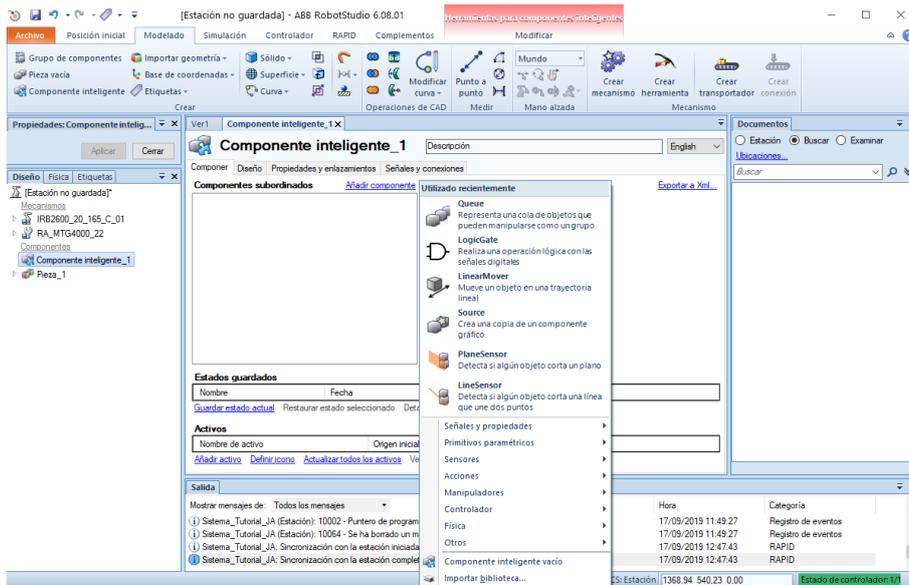


Figura 3.64 Categorías de componentes

Deteniéndonos en este punto podemos observar que disponemos de ocho categorías diferentes:

- **Señales y propiedades**

En esta categoría encontramos las operaciones lógicas, multiplexores, contadores, aritmética y el resto de elementos cuyo fin es modificar una señal o propiedad del sistema a programar.

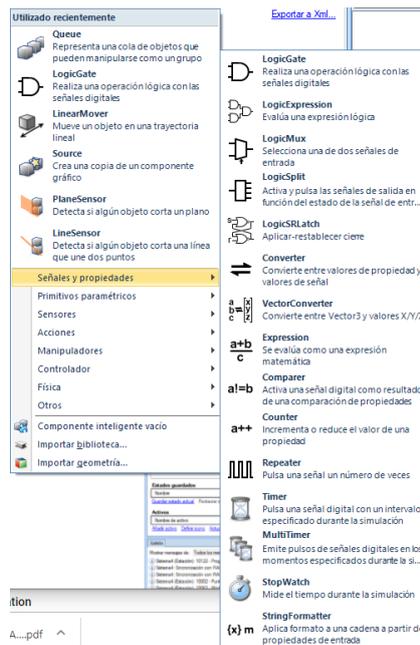


Figura 3.65 Señales y propiedades

- **Primitivos paramétricos**

Componentes destinados a crear de forma automática sólidos y líneas, así como trabajar con operaciones de matriz generando copias de elementos preexistentes.

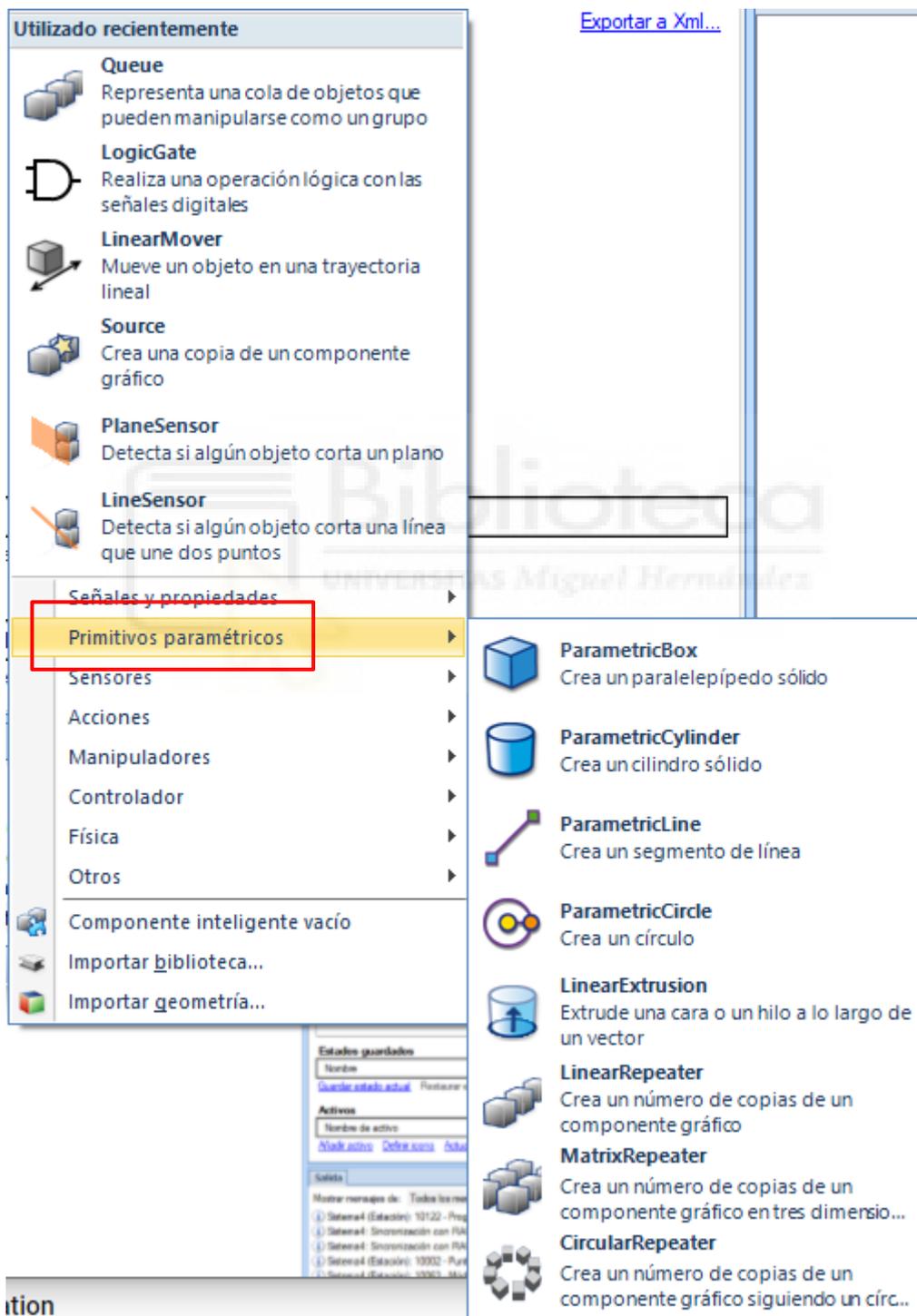


Figura 3.66 Primitivos paramétricos

- **Sensores**

Conjunto de sensores necesarios para la mayoría de procesos de producción automatizado. Incluye sensores lineales, planares, de colisión...

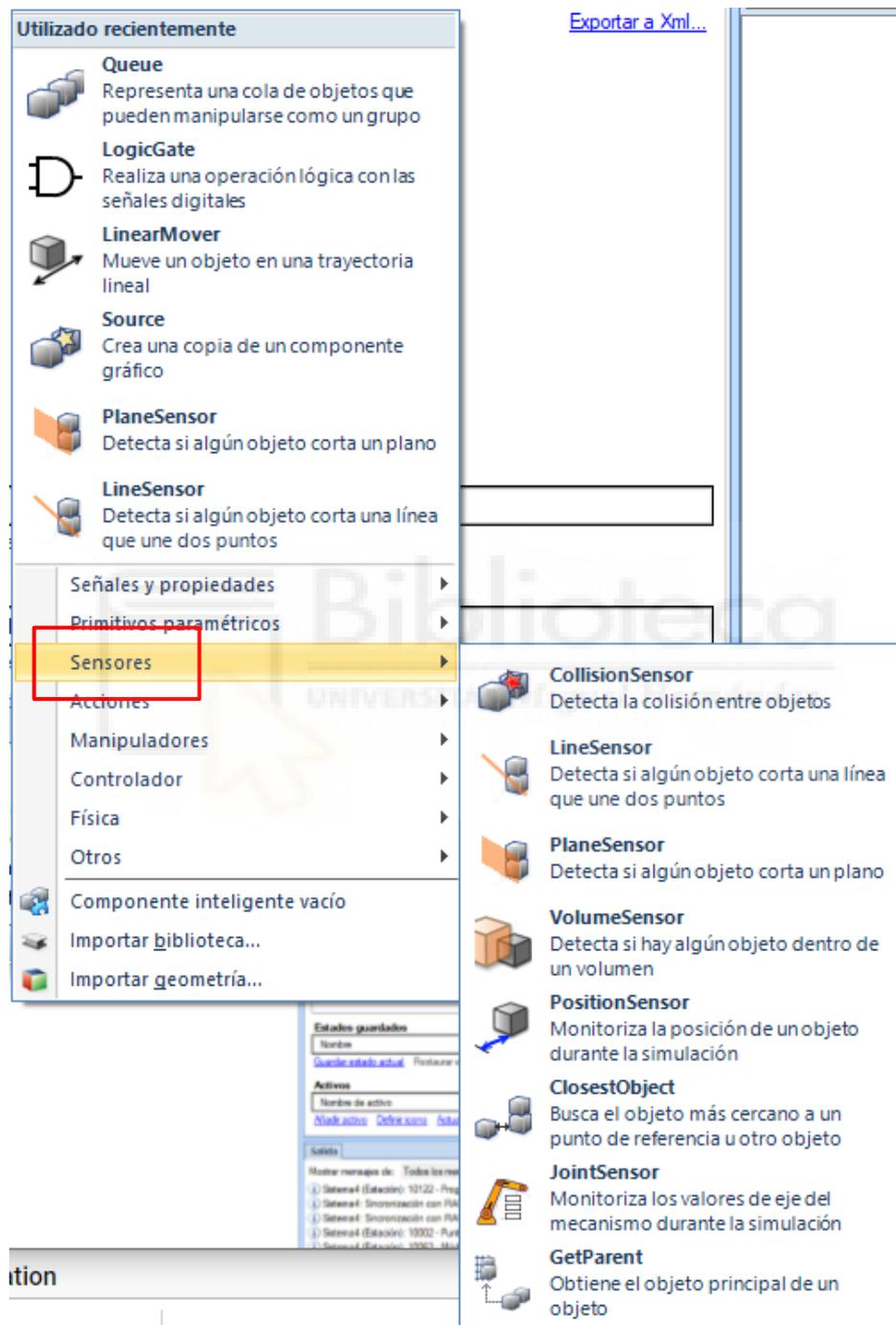


Figura 3.67 Sensores

- **Acciones**

Conjunto de operaciones simples pero imprescindibles como conectar/desconectar, mostrar/ocultar, fuentes...

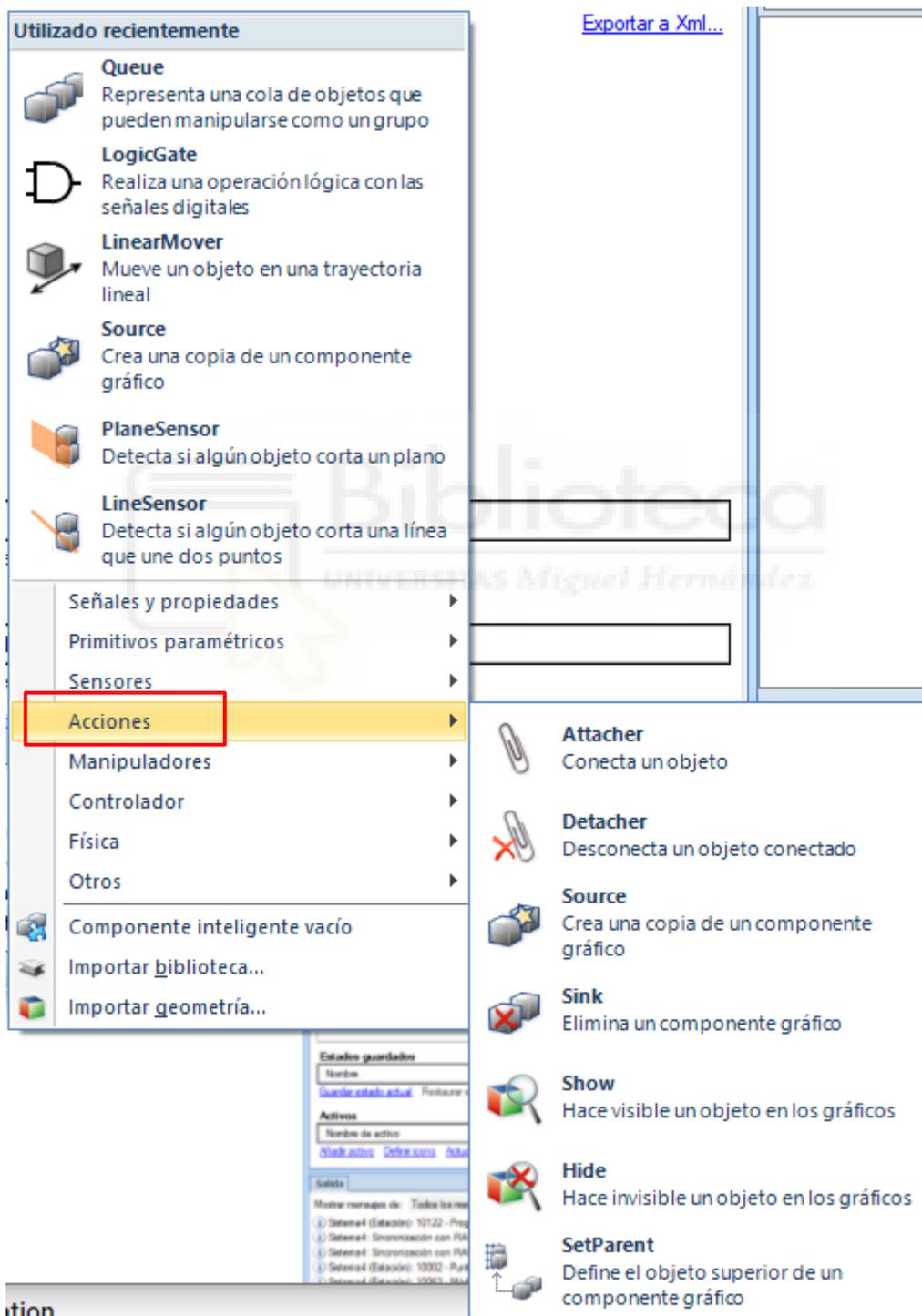


Figura 3.68 Acciones

- Manipuladores

Conjunto de componentes relativos a la posición y orientación de los objetos y robots.

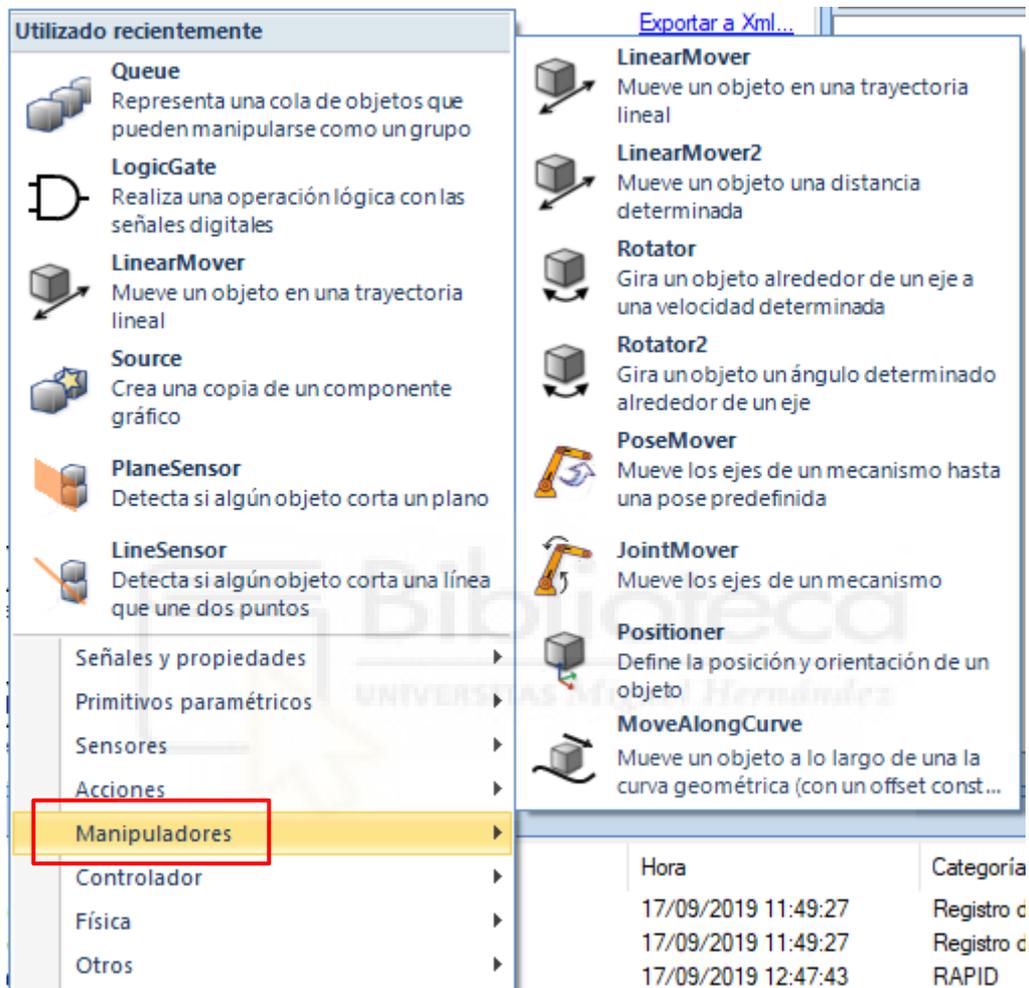


Figura 3.69 Manipuladores

- **Controlador**

Establece un vínculo bidireccional con RAPID sobre una variable

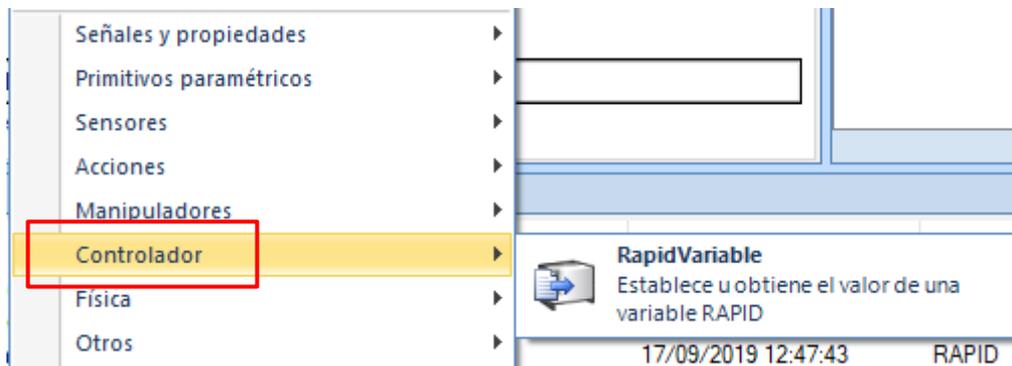


Figura 3.70 Controlador

Establece un vínculo bidireccional con RAPID sobre una variable

- **Física**

Permite modificar las propiedades físicas de los objetos y ejes.

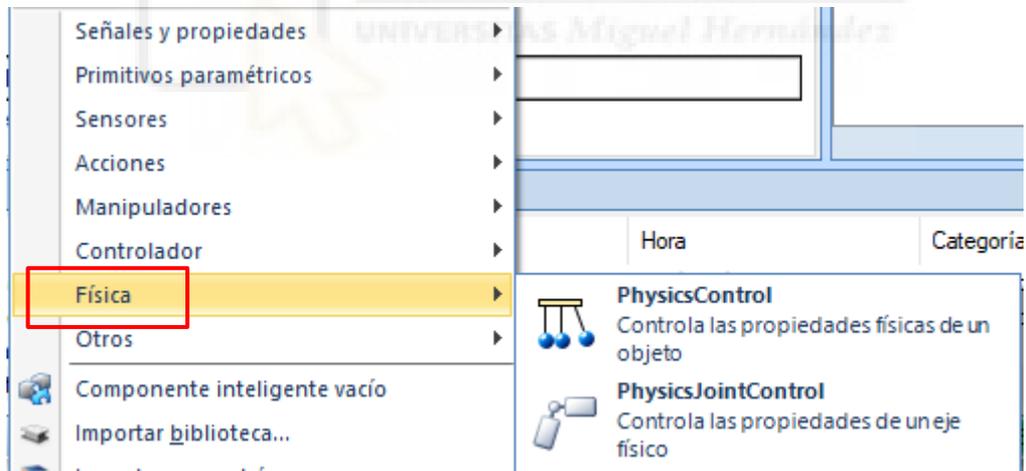


Figura 3.71 Física

- Otros

Importante sección construida con elementos de distintas indoles e importancia que no encajan en los apartados anteriores, entre estos podríamos destacar:

- Queue: Representa una cola de objetos que puede ser tratada como un conjunto.
- SoundPlayer: Emite un sonido frente a un evento.
- Random: Generador de números aleatorios.
- StopSimulation: Detiene la simulación.

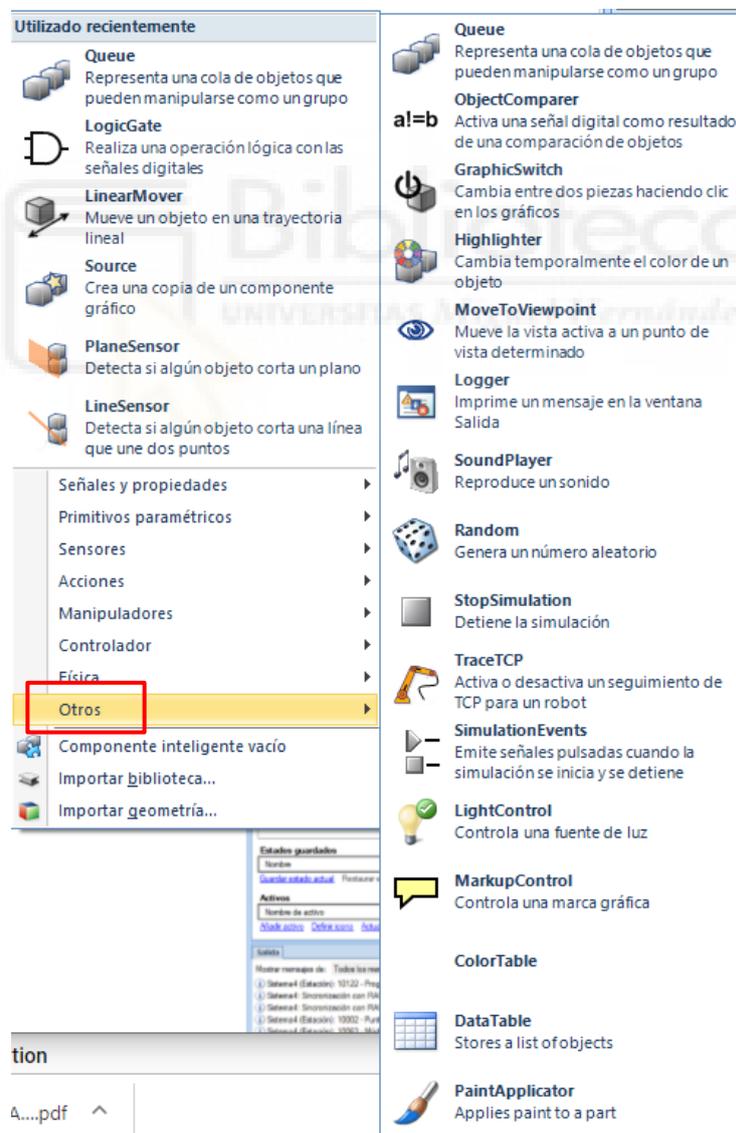


Figura 3.72 Otros

Dentro de componente inteligente, encontramos cuatro pestañas más: Diseño, Propiedades y enlazamientos y Señales y conexiones. En estas pestañas ordenaremos y conectaremos los objetos, modificaremos su configuración y realizaremos las gestiones relativas a entradas y salidas.

Como siempre, la mejor manera de visualizar el modo de empleo de los Componentes inteligentes mediante un ejemplo, a continuación, se propone la configuración de una pinza como componente inteligente.

Comenzaremos cargando la pinza de la biblioteca, por comodidad se ha seleccionado la pinza "Smart_Gripper" en su variante "Servo_fingers" herramienta clásica de los robots colaborativos de ABB dotados con brazos IRB14000, no obstante, es aplicable a cualquier robot.



Figura 3.73 Smart Gripper\Servo_fingers

Procedemos a crear y renombrar un componente inteligente como hemos visto previamente y arrastramos nuestra herramienta a su interior.

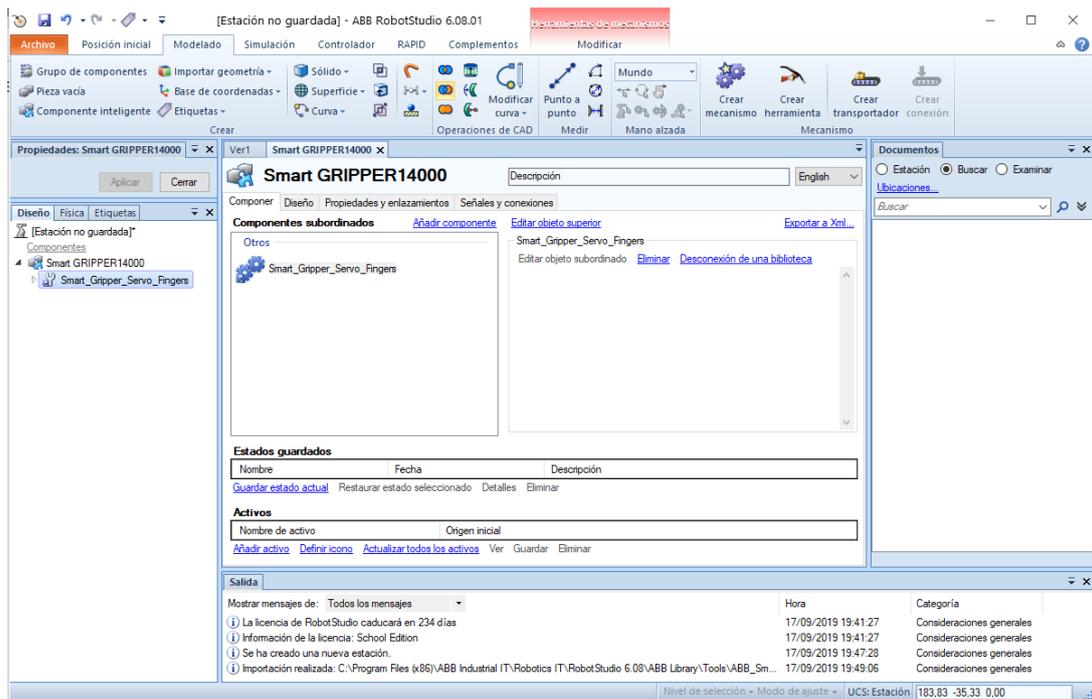


Figura 3.74 SC Smart GRIPPER 14000

A continuación, sin salir de la pestaña “componer” desde “añadir componentes”, seleccionaremos los elementos necesarios para el comportamiento deseado.



Figura 3.75 Componentes SC Smart GRIPPER 14000

Si en este punto nos desplazamos a la pestaña de diseño podemos observar cómo se nos han creado estos elementos en un entorno que nos permite vincularlos entre sí.

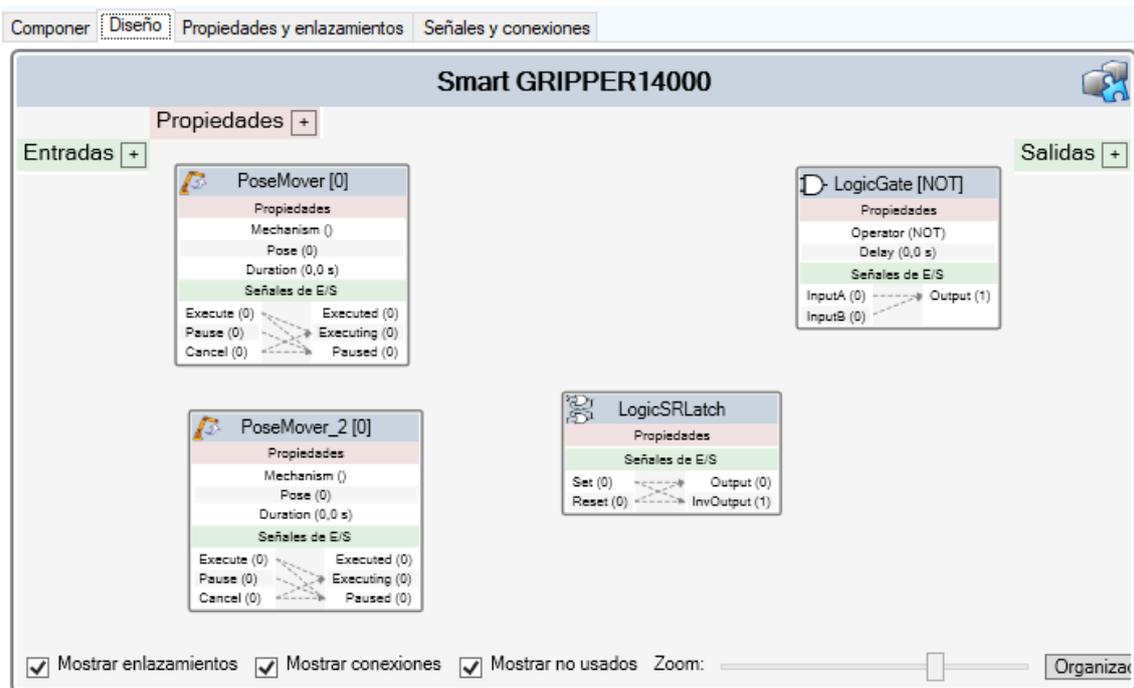


Figura 3.76 Pestaña Diseño

Aprovechamos para configurar sus entradas y salidas; Como entrada simplemente generaremos una señal que marcara el comportamiento del robot, por su parte, la salida constara simplemente de dos sensores a modo de testigo que nos indicara el estado de la pinza (abierta-cerrada)

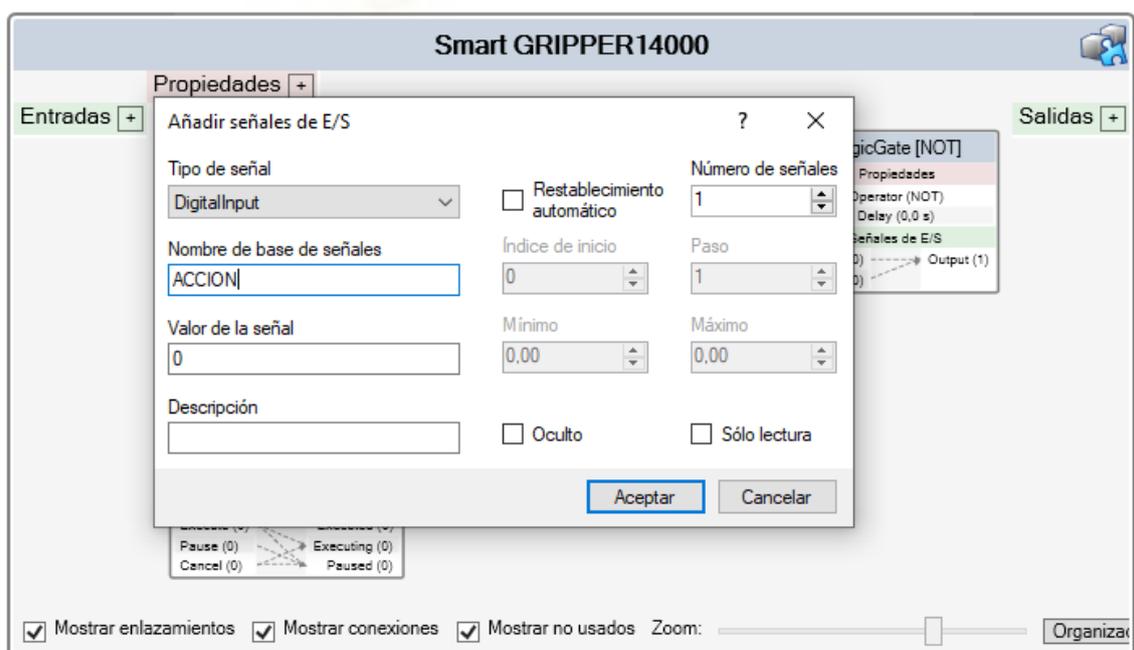


Figura 3.77 Señales de E/S

Realizando los vínculos pertinentes obtenemos el siguiente esquema de control, el cual nos permite accionar la pinza y obtener a su vez un output de esta posición.

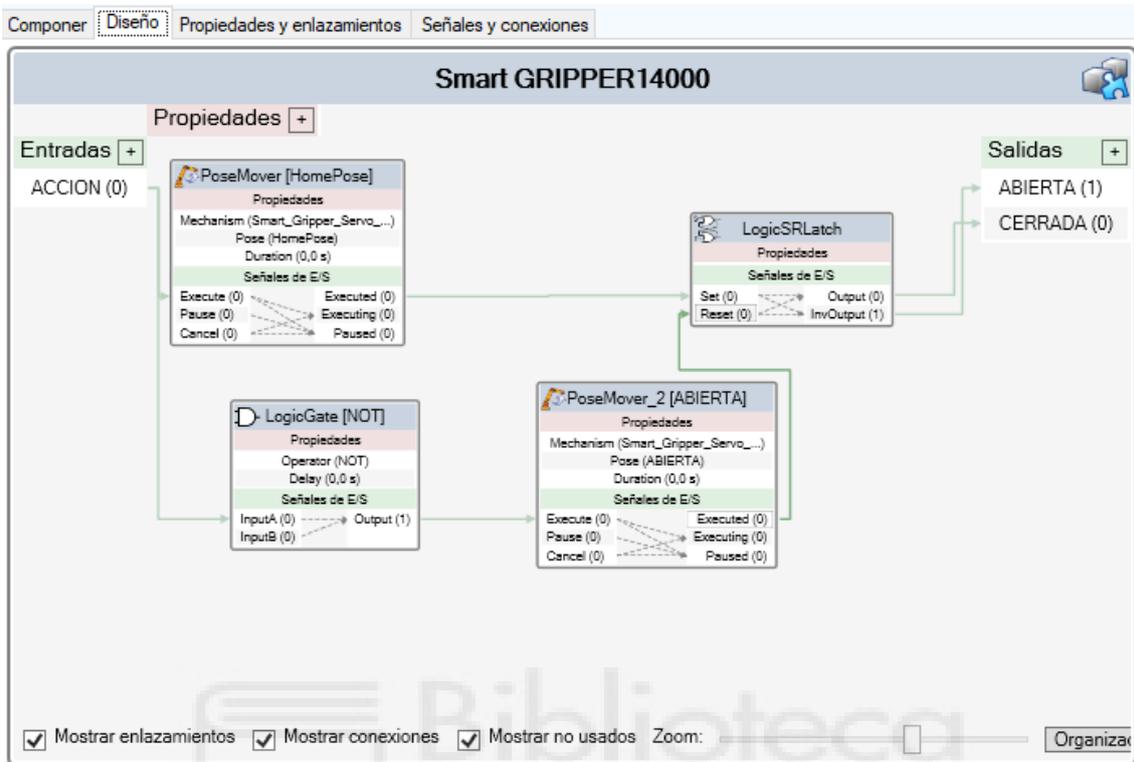


Figura 3.78 Lógica Smart GRIPPER 14000

Descriptivamente, accionando la señal “ACCION” haremos que la pinza se cierre y se active la señal de salida a modo de piloto que nos indique que la pinza está cerrada, por su parte la misma ACCION negada volverá a abrir la pinza y activara el piloto correspondiente.

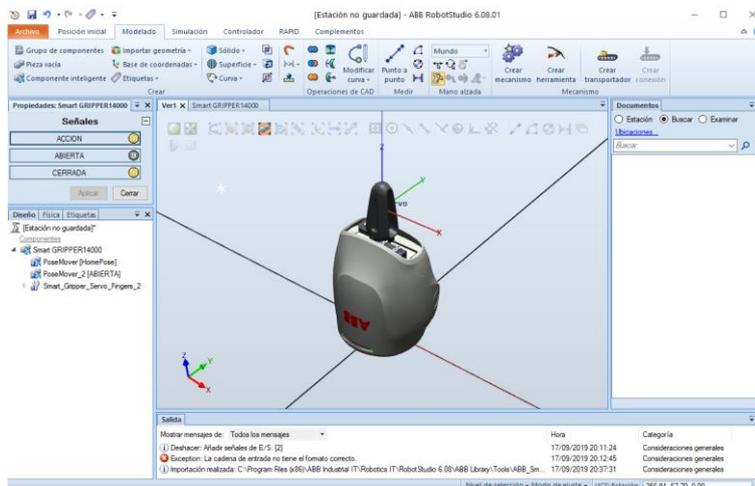


Figura 3.79 Pinza cerrada

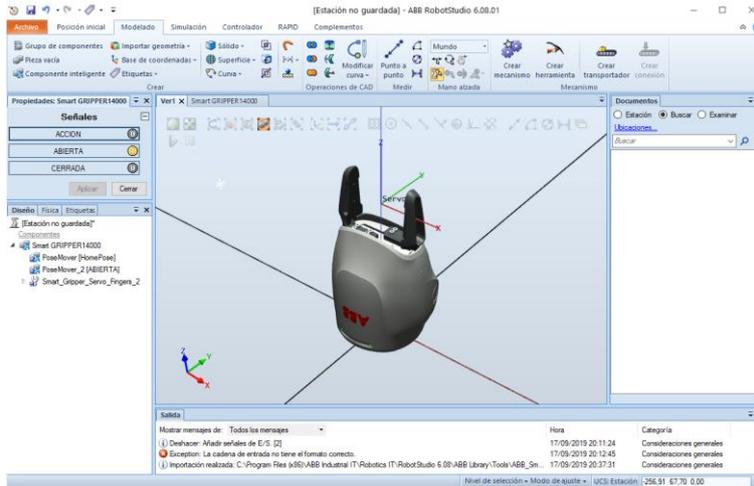


Figura 3.80 Pinza abierta

Si vamos a la pestaña “Señales y conexiones” podemos observar la misma información que tenemos en nuestra pestaña de diseño, pero en forma de tabla.

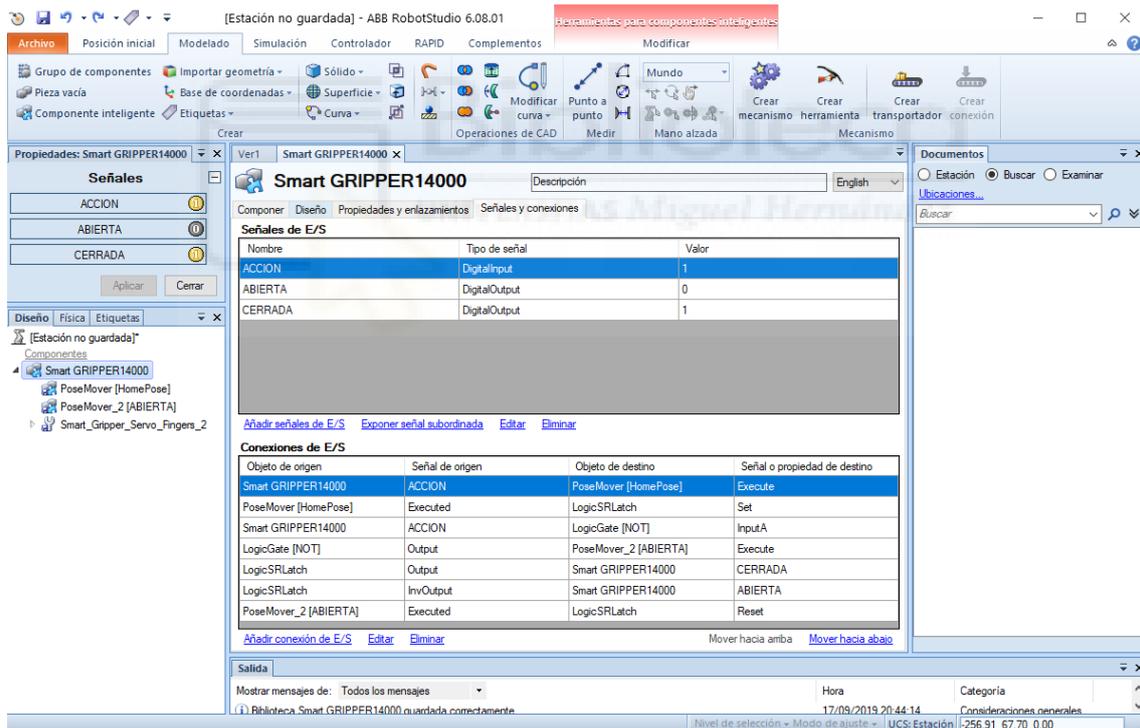


Figura 3.81 Señales y conexiones

Una vez creado nuestro componente inteligente aprovechamos para guardarlo en la biblioteca para futuros usos.

El siguiente paso lógico sería acoplar nuestra herramienta inteligente a el robot y realizar las configuraciones pertinentes para que la pinza responda a las señales digitales del robot.

Para ello, comenzamos cargando un robot y añadiéndole nuestra herramienta.

Se podría seleccionar cualquier robot de la biblioteca, no obstante, por ser la herramienta típica de este robot, se ha seleccionado el IRB14000, con interés adicional al tratarse de un robot colaborativo.

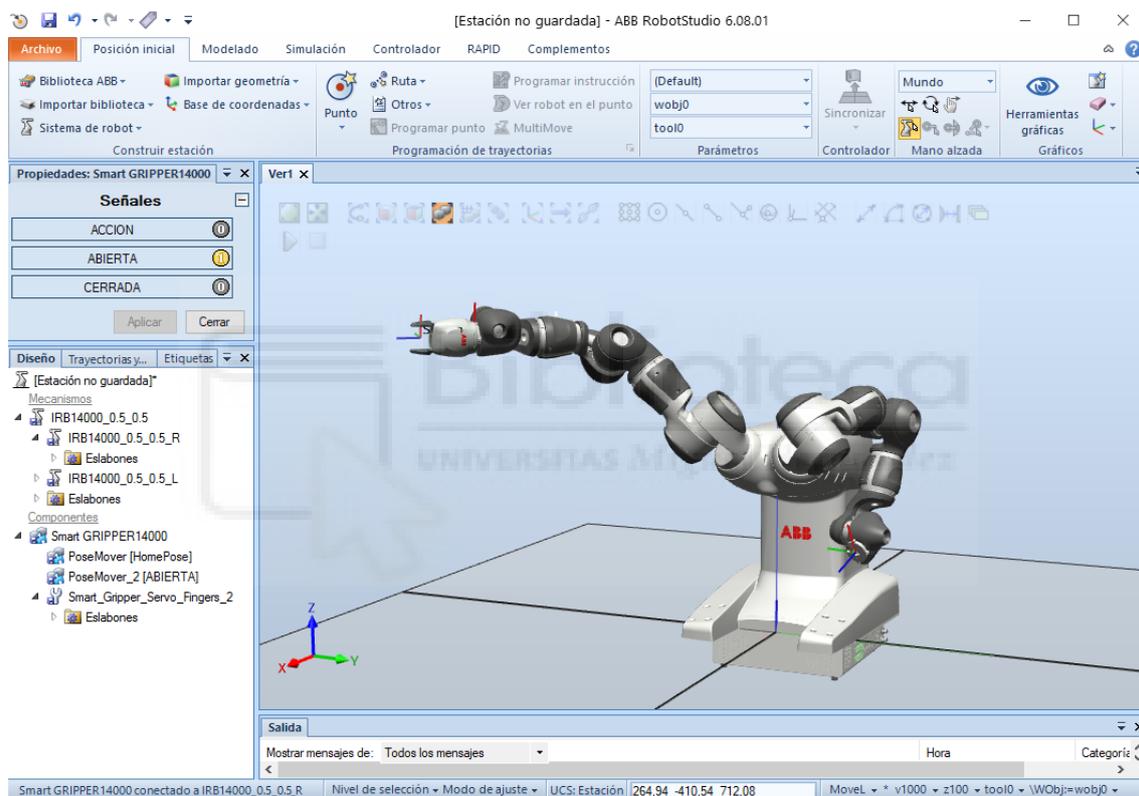


Figura 3.82 Sistema robot con herramienta

En este punto, tenemos nuestro elemento inteligente correctamente creado e incorporado al robot, el siguiente paso lógico será, por tanto, configurarlo para que funcione en sincronía con las señales del robot, para ello deberemos crear y editar de ser necesario las señales oportunas, así como la interconexión entre ellas, cuestiones que se analizan en los puntos siguientes.

3.3.2 Creación de señales E/S

La potencia de RobotStudio reside en la interconectividad de sus elementos, es decir y siguiendo con el ejemplo anterior, por el momento tenemos un robot con una herramienta en su muñeca y hemos visto como podíamos accionarla mediante una señal interna de acción, es lógico pensar que cuando programemos nuestra estación, la pinza, en este caso, deberá obedecer al controlador. Si trabajásemos con un controlador físico, deberíamos configurar nuestro software en base a este y aprovechar sus entradas y salidas, a nivel de programación y simulación era una cuestión más delicada en versiones anteriores, antes, RobotStudio marcaba una diferencia clara entre trabajar con una controladora real y una virtual, no obstante, a partir de la versión 6.08 desaparece esta separación y RobotStudio entiende que al tratarse de un software de programación siempre trabajara de una forma virtual aun conectándose a un software físico.

Para crear y/o editar señales disponemos de varios métodos, una manera es en la pestaña de “Controlador”, desplegando el árbol configuración en nuestro sistema encontramos las señales en “I/O System”, otra manera es pinchando en el icono de “Configuración” también en la pestaña de “Controlador” y presionar sobre “añadir señales”.

Siguiendo con nuestro ejemplo, pulsamos sobre añadir señal y aparece la ventana que ilustra la figura siguiente.

Añadir señales: Sistema8

Tipo de señal: Salida digital

Número de señales: 1

Nombre de base de señales: ACTIVAPINZA

Índice de inicio: 0

Paso: 1

Asignado a Dispositivo:

Inicio de mapeo de dispositivo: 0

Etiqueta de identificación de señal:

Categoría:

Avanzado

Nivel de acceso: Default

Valor predeterminado: 0

Invertir valor físico

Aceptar Cancelar

Figura 3.83 Señal activación de pinza

Rellenamos los campos pertinentes, en este caso, simplemente deseamos crear una salida del sistema de nuestro robot capaz de activar la pinza, por tanto, indicamos que se trata de una salida y le ponemos nombre, no le asignamos ningún dispositivo para visualizarlo mejor simplemente, además deberemos modificar el nivel de acceso a “All” para poder activar la señal a voluntad más tarde.

Si nos dirigimos a las opciones de sistema de entradas y salidas podemos comprobar la correcta creación de la señal.

Nombre	Tipo	Valor	Valor mínimo	Valor máximo	Simuladas	Red	Dispositivo	Mapeo de dispositivo
ACTIVAPINZA	DO	0	0	1	Sí	<ninguno>	<ninguno>	
Collision_Avoidance	DO	1	0	1	Sí	<ninguno>	<ninguno>	
custom_DI_0	DI	0	0	1	No	DeviceNet	D652_10	0
custom_DI_1	DI	0	0	1	No	DeviceNet	D652_10	1
custom_DI_2	DI	0	0	1	No	DeviceNet	D652_10	2
custom_DI_3	DI	0	0	1	No	DeviceNet	D652_10	3
custom_DI_4	DI	0	0	1	No	DeviceNet	D652_10	4
custom_DI_5	DI	0	0	1	No	DeviceNet	D652_10	5
custom_DI_6	DI	0	0	1	No	DeviceNet	D652_10	6
custom_DI_7	DI	0	0	1	No	DeviceNet	D652_10	7
custom_DO_0	DO	0	0	1	No	DeviceNet	D652_10	0
custom_DO_1	DO	0	0	1	No	DeviceNet	D652_10	1
custom_DO_2	DO	0	0	1	No	DeviceNet	D652_10	2
custom_DO_3	DO	0	0	1	No	DeviceNet	D652_10	3
custom_DO_4	DO	0	0	1	No	DeviceNet	D652_10	4

Figura 3.84 Sistema E/S estación de prueba

3.3.3 Lógica de estación

La lógica de la estación es un elemento de suma importancia, es la responsable de interconectar las señales entre los distintos dispositivos para su correcto funcionamiento.

En nuestro ejemplo teníamos un robot con una pinza acoplada a su muñeca y en el sistema del robot creamos una señal que esperamos, active la pinza, para ello, nos dirigimos a la pestaña “Simulación” y entramos en “Lógica de la estación”, aquí, en la subpestaña “Diseño” vincularemos la herramienta a las señales del robot de la misma

manera que hicimos de forma interna en el componente inteligente; desplegamos el menú de señales habilitadas en nuestro sistema, seleccionamos “ACTIVAPINZA” ya que es la señal dedicada que creamos para ello, y la unimos mediante una flecha a la señal interna de acción de la pinza que anteriormente llamamos “ACCION”.

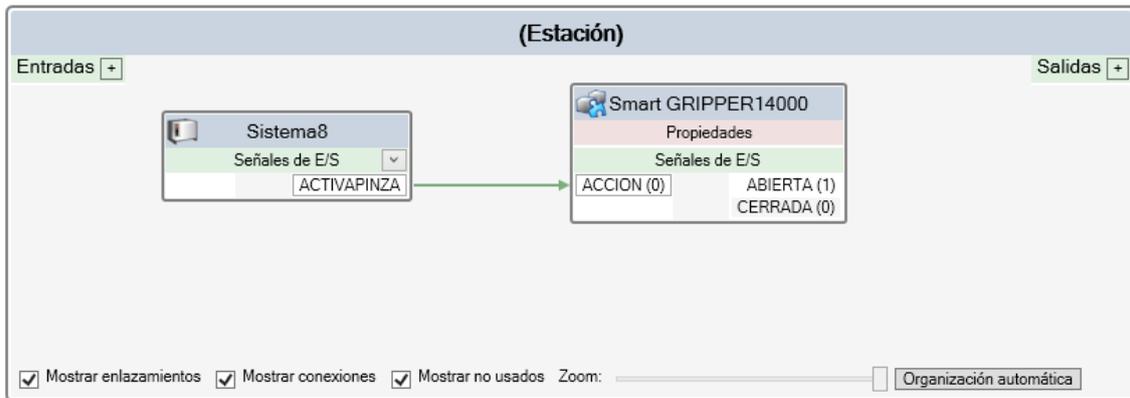


Figura 3.85 Lógica de la estación de prueba

Ya tendríamos lista esta pequeña prueba, como siempre que realizamos cambios en nuestro sistema, deberemos sincronizar antes de proceder a la simulación.

Puesto que no hemos programado nada, para comprobar el correcto funcionamiento de nuestra señal, deberemos configurar la simulación en ciclo continuo.

Hecho esto, podemos comprobar con el simulador de E/S de la pestaña de simulación como la pinza responde correctamente a las órdenes de nuestro sistema.

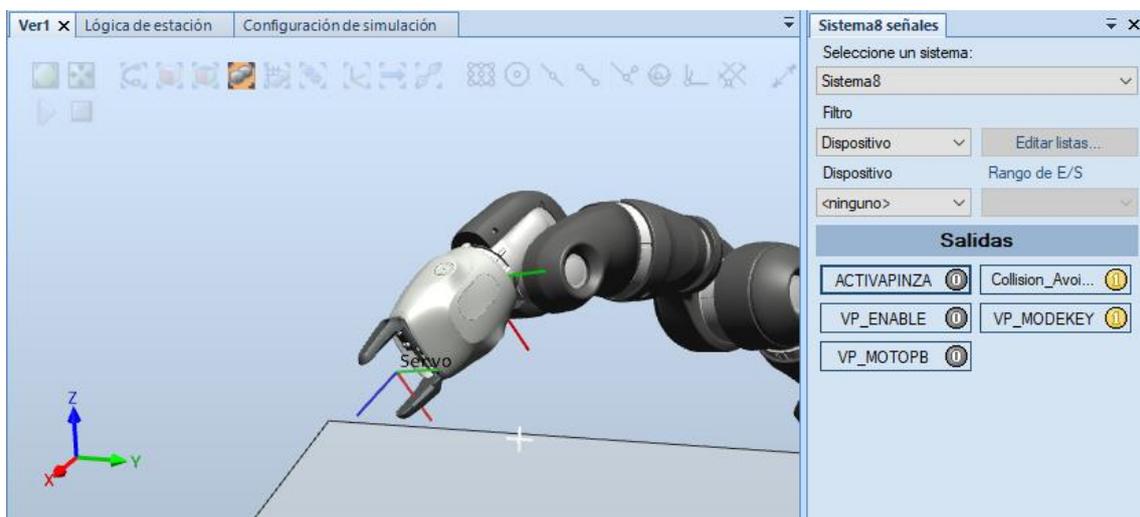


Figura 3.86 Simulador E/S OFF

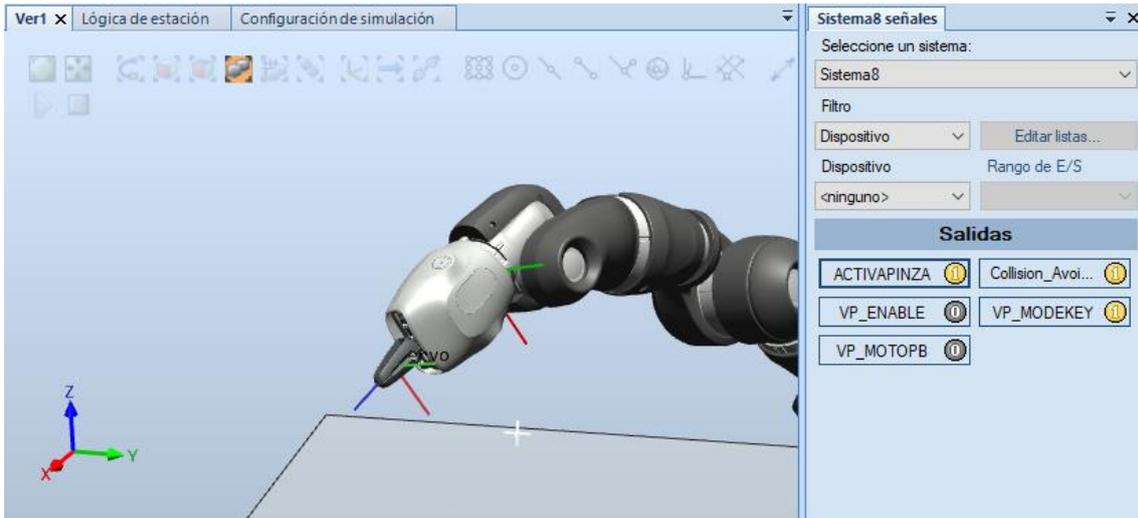


Figura 3.87 Simulador E/S ON

3.4 Simulación de la estación

En el punto anterior se ha realizado una pequeña aproximación a la simulación, no obstante, la simulación tiene distintas opciones y es interesante repasar algunas.

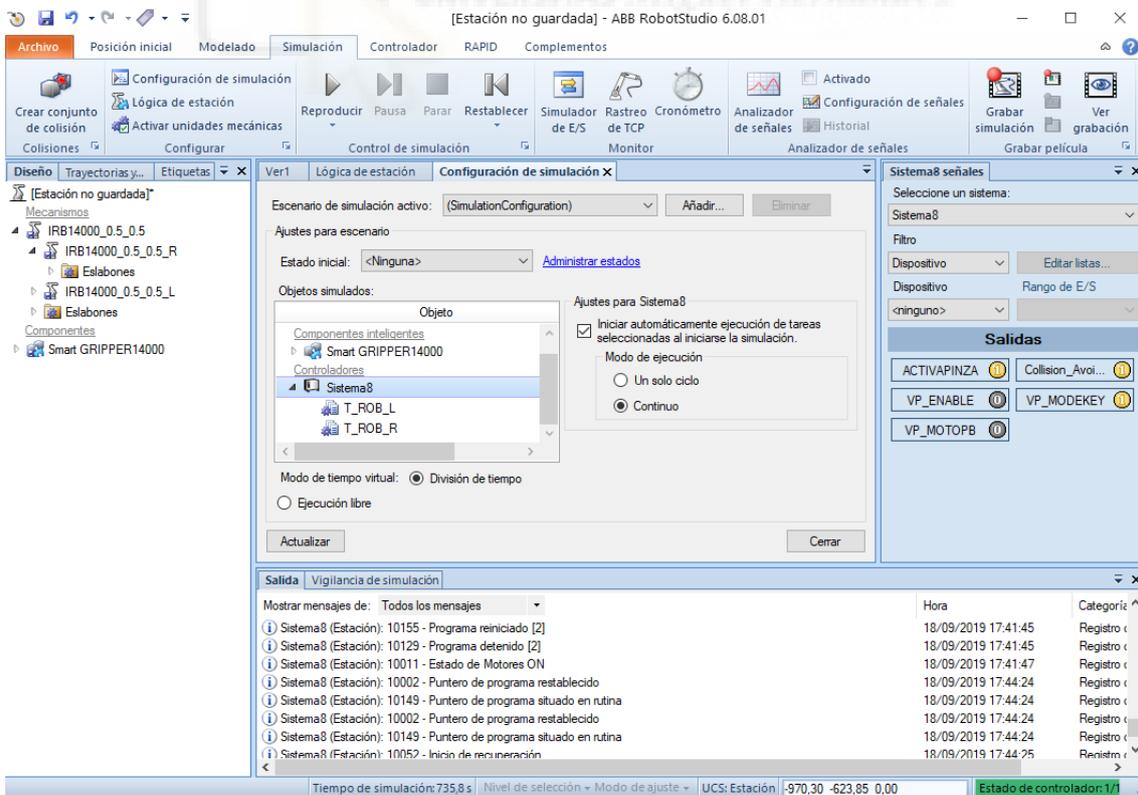


Figura 3.88 Configuración de la simulación

Una posibilidad interesante es simular paso por paso desde RAPID, función de gran utilidad para detectar posibles errores, para ello, tan solo nos dirigiremos a la pestaña RAPID y pulsaremos “verificar programa” y aplicamos guardando los cambios.

La configuración de la simulación se realiza desde el icono del mismo nombre en la pestaña “Simulación”.

Alguna opción interesante es la anteriormente vista para trabajar en ciclos únicos o en continuo, o la posibilidad de simular tanto el main, como solo alguna trayectoria u operación en concreto.

Es útil, también, posibilidad también la de grabar la simulación en distintos formatos de video.

3.5 Flexpendant

FlexPendant es la consola de mano que utiliza el operador para controlar el robot, permite realizar todas las tareas pertinentes, carga de programas, programación básica, modificación, calibración, movimiento de ejes... Está compuesto por una pantalla táctil, un joystick y botones de acceso rápido, RobotStudio permite simular de forma virtual el FlexPendant del robot, esta opción la encontramos en la barra de opciones de controlador [2].

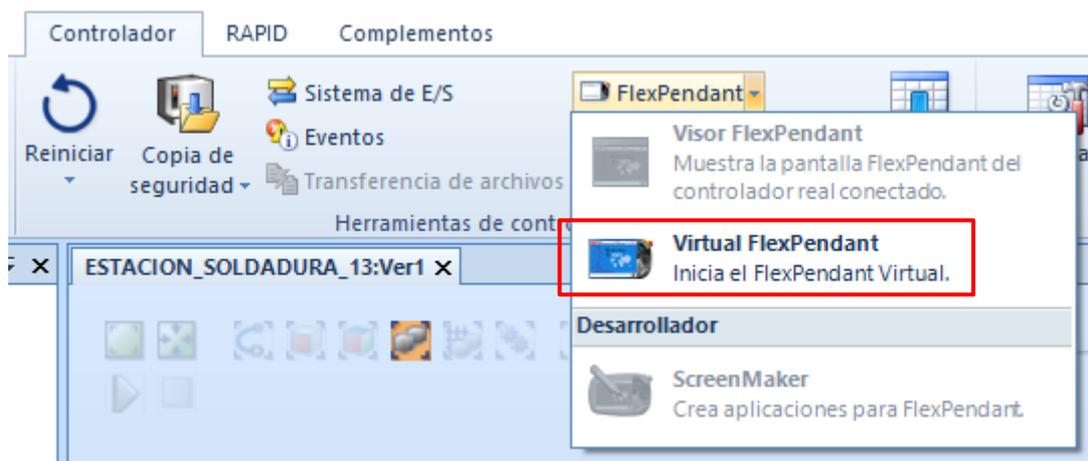


Figura 3.89 Virtual FlexPendant

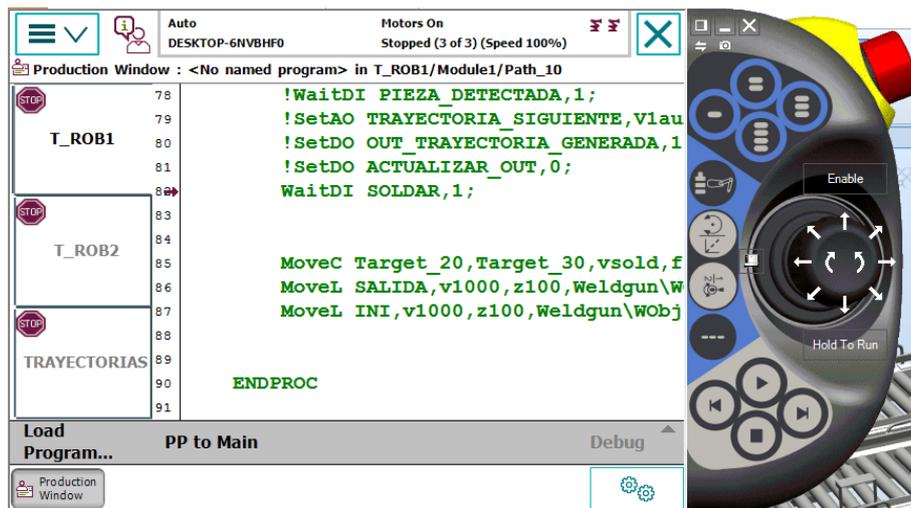


Figura 3.90 Ejemplo FlexPendant

Dispone de tres posiciones de trabajo:

- Modo automático.
- Modo manual a velocidad reducida: Debe ser empleado por motivos de seguridad cuando se trabaja dentro del área de trabajo del robot
- Modo manual a velocidad máxima: Se empleará estando fuera del espacio de trabajo para comprobar el correcto funcionamiento en las fases finales de programación.

El menú principal del FlexPendant dispone de los siguientes elementos:

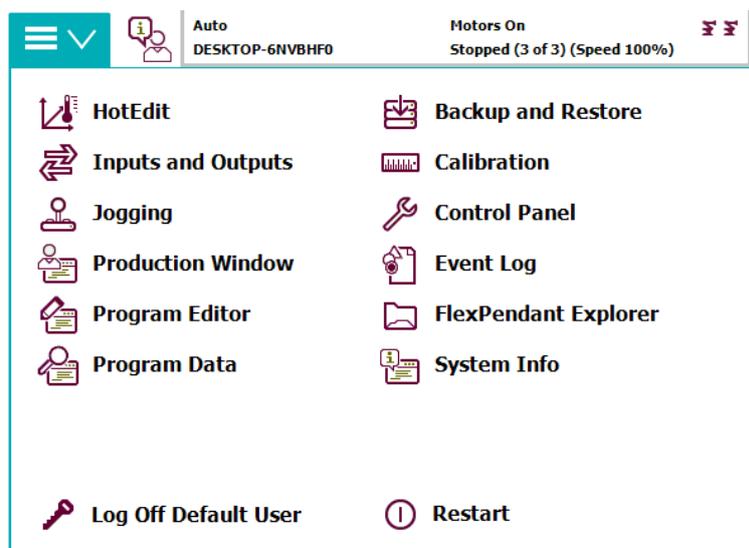


Figura 3.91 Menú FlexPendant

- Edición en caliente “**HotEdit**”: Posibilita la edición de posiciones programadas, tanto en coordenadas como en orientación en todos los modos de funcionamiento, con la única restricción de que solo se podrá utilizar con posiciones de tipo “Robtarget”.

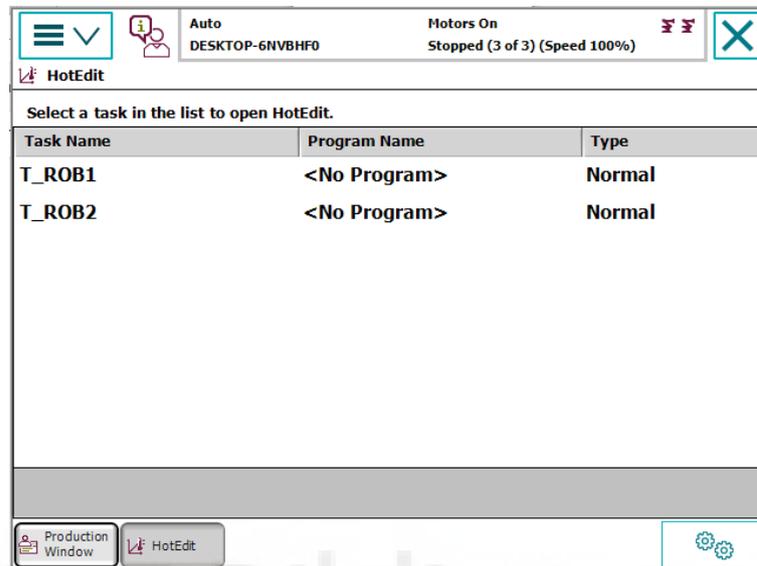


Figura 3.92 HotEdit

- Entradas y salidas “**Inputs and Outputs**”: Configuración de las señales del sistema con los parámetros de este.

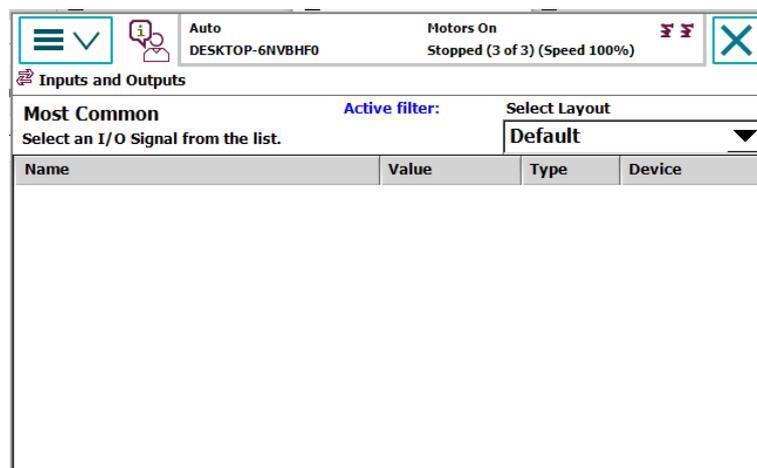


Figura 3.93 Inputs and Outputs

- Movimiento **“Jogging”**: Encontramos las distintas funciones de movimiento, las más comunes también se encuentran en el menú de configuración rápida.

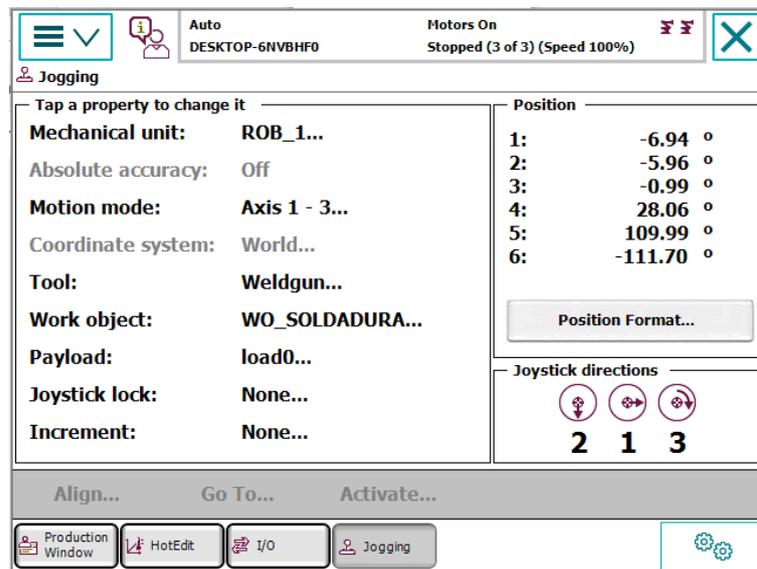


Figura 3.94 Jogging

- Ventana de producción **“Production Window”**: Permite operar con el código Rapid del programa durante la ejecución, permite la carga de nuevos programas y el desplazamiento del puntero.

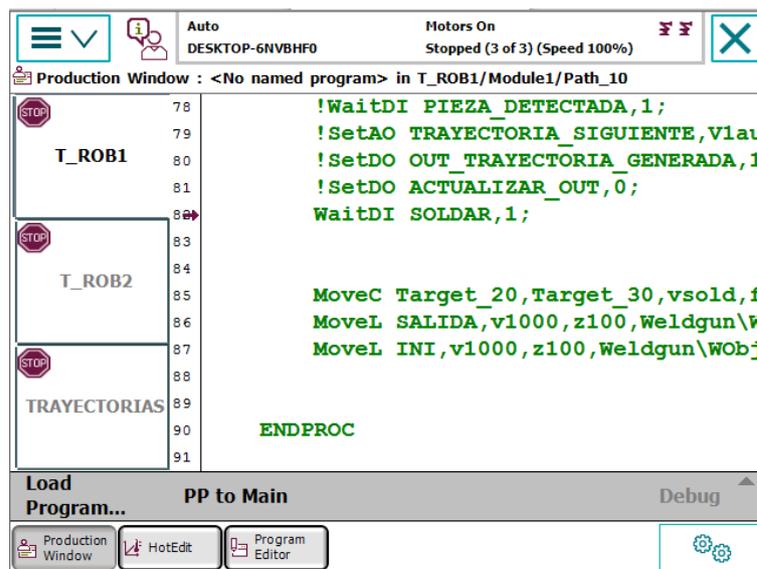


Figura 3.95 Production Window

- Editor de programas **“Program Editor”**: Creación y edición de programas, permite trabajar con distintos programas simultáneamente si se tiene instalada la opción de **“Multitasking”**.

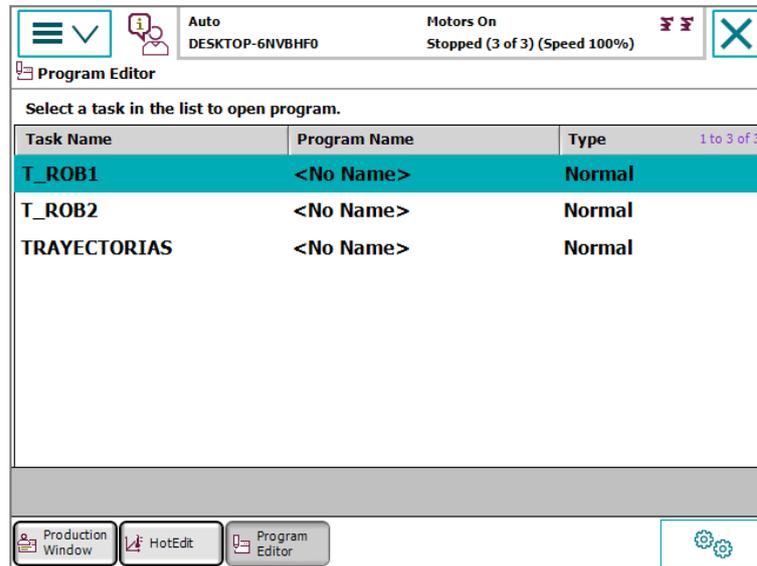


Figura 3.96 Program Editor

- Datos de programa **“Program Data”**: Útil cuando de trabajo con numerosas instancias.

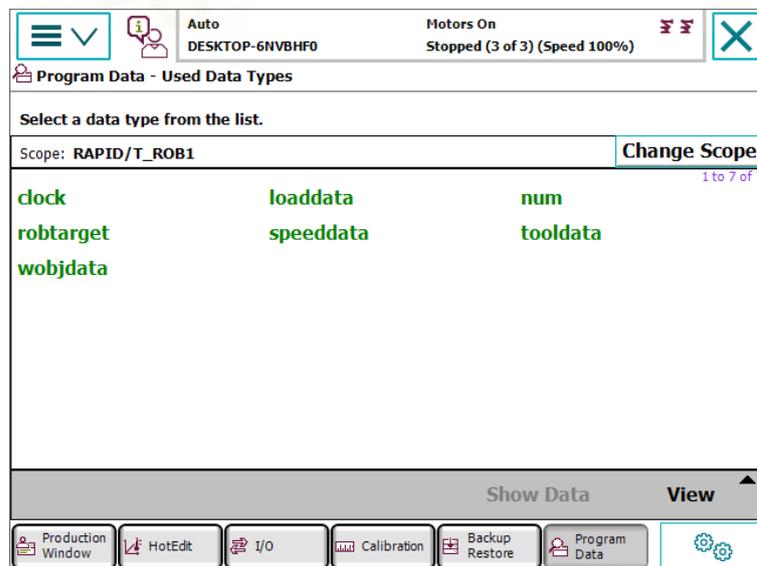


Figura 3.97 Program Data

- Copia de seguridad y restauración **“Backup and Restore”**.

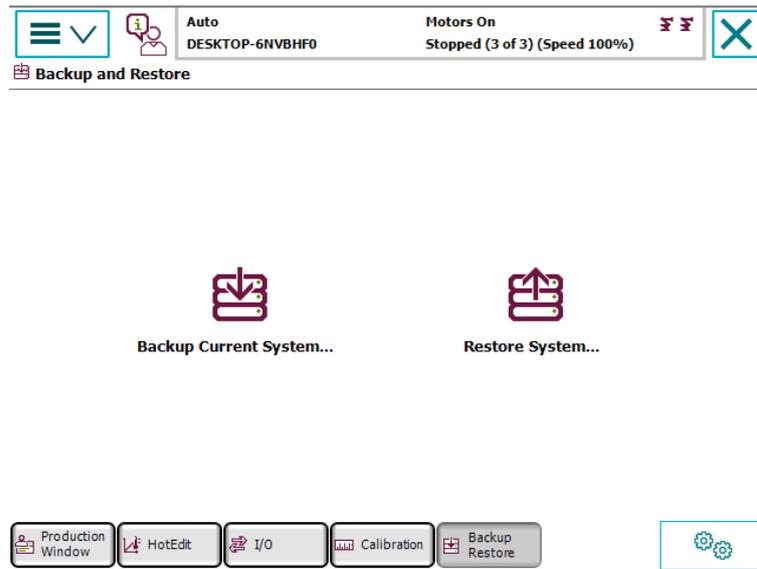


Figura 3.98 Backup and Restore

- Calibración **“Calibration”**: Permite calibrar las distintas unidades mecánicas que conforman el sistema.

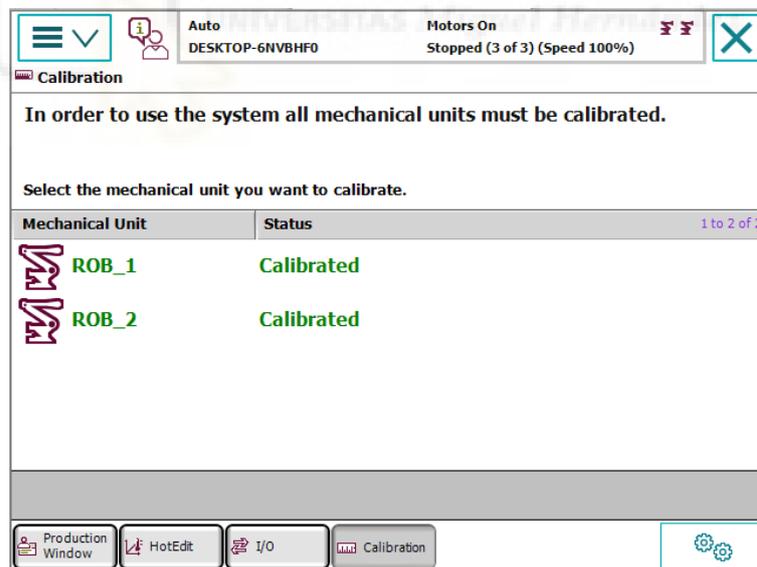


Figura 3.99 Calibration

- Panel de control **“Control Panel”**: Permite personalizar el sistema de robot y el Flexpendant.

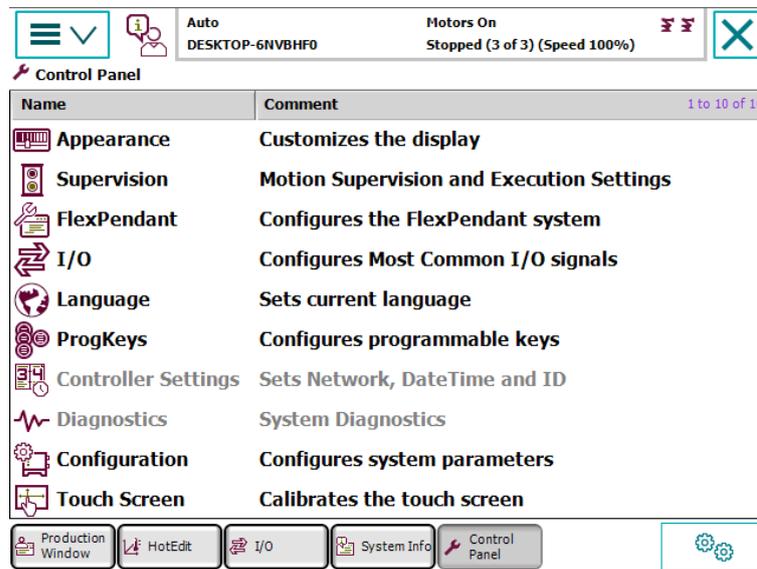


Figura 3.100 Control Panel

- Registro de eventos **“Event Log”**: Almacena los distintos eventos acontecidos durante las tareas.

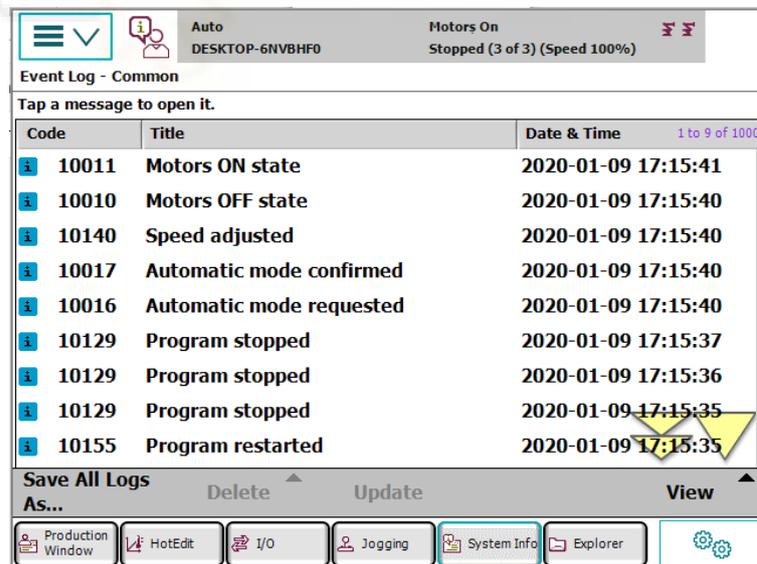


Figura 3.101 Event Log

- Explorador **“FlexPendant Explorer”**: Consiste básicamente en un explorador de archivos con el sistema de archivos del robot.

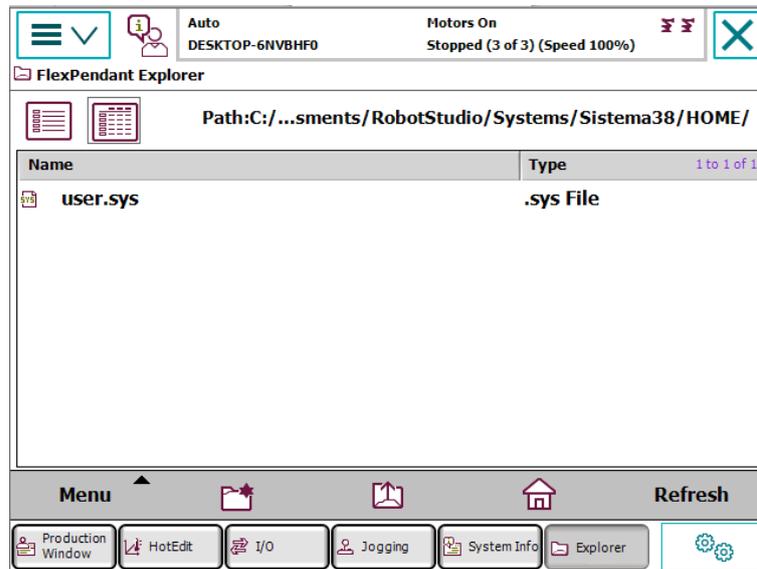


Figura 3.102 FlexPendant Explorer

- Información del sistema **“System Info”**: Información relevante del controlador y el sistema cargado, algunos de los atributos incluidos son la versión de RobotWare, las opciones en uso, conexiones de red...

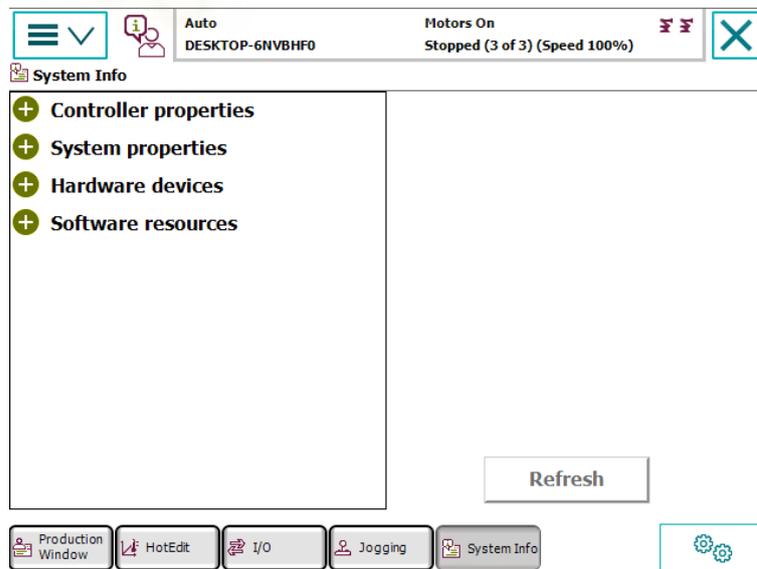


Figura 3.103 System Info

4 DISEÑO Y PROGRAMACIÓN DE UNA ESTACIÓN DE SOLDADURA Y PICK AND PLACE DE BISAGRAS PARA EL SECTOR DE LA AUTOMOCIÓN

Una vez adquiridos conocimientos sobre los distintos procesos involucrados en el proyecto y una aproximación a las herramientas software, RobotStudio en este caso, se procede al diseño y simulación de una estación capaz de cumplir las tareas deseadas, desde la identificación y selección de los componentes necesarios, hasta la completa simulación, detallando los distintos aspectos computacionales tanto a nivel RobotStudio como programación Rapid.

4.1 Determinación de necesidades y limitaciones

El presente proyecto, si bien, está muy abierto en cuanto a selección de componentes debido a su carácter de aproximación a un software muy específico, si está condicionado a ciertos parámetros previamente fijados, así como el logro de ciertos objetivos.

Objetivos:

- Continuidad: La estación robotizada debe de ser capaz de trabajar de forma continua sin interrupciones.
- Precisión, el robot debe realizar de forma precisa su tareas para ser un modelo productivo.
- Flexibilidad: La estación debe ser capaz de trabajar con distintas piezas que no han sido definidas explícitamente.

Limitaciones:

- Velocidad: Se deberá operar en continuo sobre una velocidad establecida en las cintas de 200 mm/s.
- Las cintas transportadoras no deben detenerse.
- El sistema debe detenerse en caso de incumplirse las condiciones de seguridad.
- Sensores: Solo se podrá utilizar un sensor, por tanto, las demás tareas se parametrizarán y ejecutar en función de la respuesta y precisión de este.

4.2 Equipos de la estación

Debido a la naturaleza del proyecto se deberá realizar una primera selección de componentes fundamentales que facilite su diseño y programación en RobotStudio, en este aspecto se pueden diferenciar tres grupos principales, en primer lugar, los conveyors o cintas transportadoras, necesarios durante todo el flujo de operación para el transporte de las piezas, por otro lado, necesitaremos los equipos necesarios para la subestación de soldadura y la subestación de “pick and place”.

4.2.1 Conveyors

Se han empleado 4 conveyors genéricos de la biblioteca de RobotStudio en su configuración de 3 metros de longitud, 400 mm de profundidad aprovechable y 775 mm de altura en los rodillos.

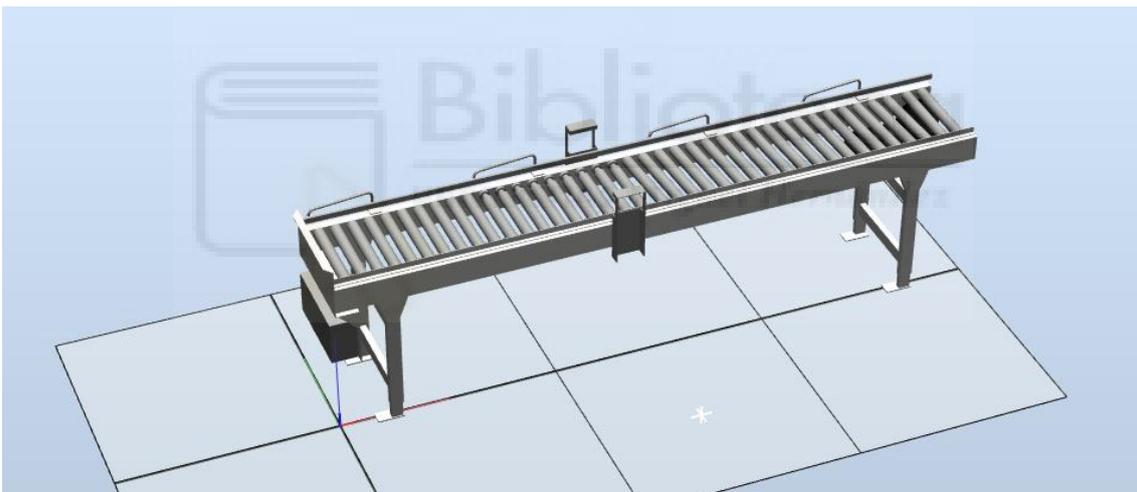


Figura 4.1 Conveyor de rodillos Estándar Robotstudio

Esta transmisión por rodillos presenta ciertos inconvenientes, por una parte, los rodillos por sí solos no aportan la precisión necesaria que requiere el proyecto, y por otra, para las operaciones de soldadura debe haber contacto eléctrico, en este aspecto los rodillos proporcionan poca superficie de contacto lo que puede dar lugar a chispas.

Estos inconvenientes son paliados mediante el uso de cintas modulares de acetal POM aportando contacto eléctrico al ser conductoras y la precisión necesaria (Figura 4.2).

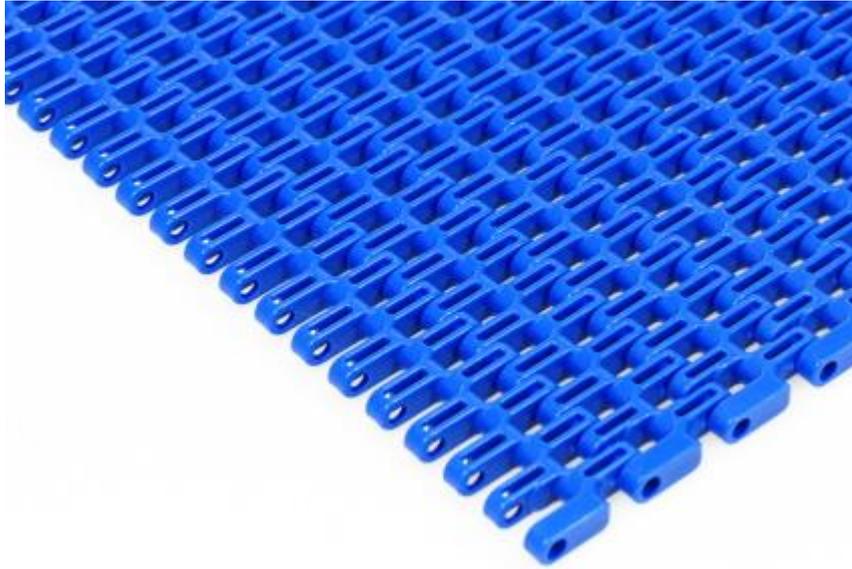


Figura 4.2 Cinta modular acetal

4.2.2 Subestación de soldadura

- **Robot**

Para la selección del robot se ha tenido en cuenta principalmente su alcanzabilidad, comprobando su correcto acceso a las trayectorias necesarias, además se ha seleccionado de un robot muy eficiente y adaptable para tareas de soldadura que siempre son necesarias en el sector de la automoción, facilitando así, su reutilización en otros procesos de ser necesario.

El robot seleccionado para la subestación de soldadura es el IRB 1520ID.

El nuevo robot IRB 1520ID de ABB [5] es un robot para la aplicación de soldadura al arco de alta precisión, con cableado de proceso integrado, que combina una capacidad de producción de 24 horas al día, 7 días a la semana, con un coste de mantenimiento un 50 % más bajo, proporcionando el menor coste por soldadura de su clase.

Proporciona una soldadura estable, alta precisión de la trayectoria, ciclos cortos y larga duración de las mangueras y cableados. Gracias a su cableado integrado, puede llevar a cabo una soldadura alrededor de objetos cilíndricos sin ninguna interrupción, y acceder más fácilmente a los cordones de soldadura de difícil alcance.

Versión	Alcance	Capacidad de carga	Peso en el brazo
IRB 1520ID	1.50m	4 Kg	10 Kg



Figura 4.3 IRB 1520ID

- **Herramientas y equipo.**

Como se aprecia en la figura siguiente, un equipo para soldadura robotizada consta de cinco elementos principales sin contar el cableado [14].

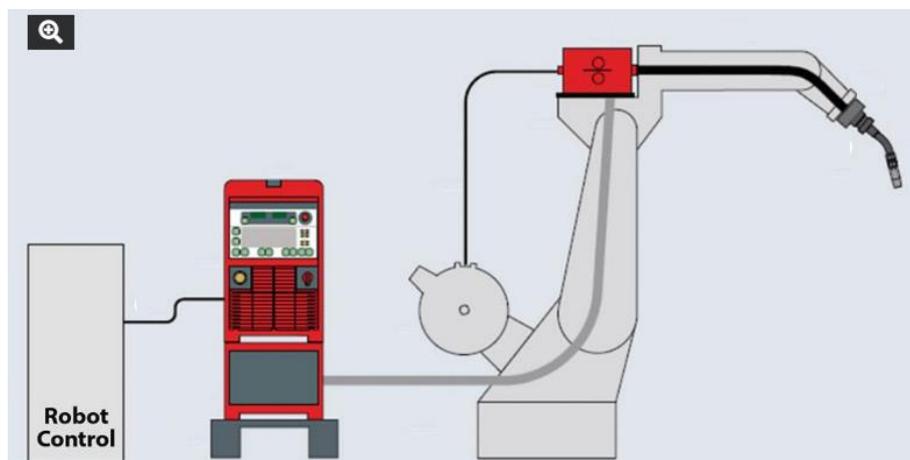


Figura 4.4 Esquema estación de soldadura robotizada

- Controlador IRC5, será el único responsable del movimiento de los robots.



Figura 4.5 IRC-5

- Fuente de energía y control Fronius TPS 4000 (TransPuls Synergic 4000)[14].



Figura 4.6 Fronius TPS 4000

- Alimentador de hilo Fronius VR1500 [14].



Figura 4.7 Fronius VR1500

- Antorcha ABIROB A360 [12]
(configuración para 1520ID)

ABIROB® A 360 (enfriada por aire)



Figura 4.8 Antorcha ABIROB



Figura 4.9 Binzel ID22

- Bidón de hilo de 250 Kg



Figura 4.10 Bidón de hilo 250 Kg

Además, se ha incluido una estación de limpieza para la antorcha de soldadura.



Figura 4.11 Equipo de limpieza Binzel

4.2.3 Subestación de pick and place

- **Robot**

Para la selección del robot se han tenido en cuenta como aspectos clave la alcanzabilidad, garantizando la factibilidad de realización de las distintas trayectorias necesarias, así como, la capacidad carga necesaria para las tareas de pick and place como objetivo específico.

Se ha buscado un robot fiable, adaptable a múltiples tareas y de uso frecuente en la industria de automoción lo que nuevamente facilita su reciclaje para futuros proyectos.

En este sentido, se ha optado por el robot IRB 1600 – 10/1.45 [4].

Versión	Alcance	Capacidad de carga
IRB 1600	1.45m	10 Kg



Figura 4.12 IRB 1600

- **Herramienta**

Para el efector final, responsable de operaciones de pick and place se ha optado por un sistema de ventosas de vacío personalizado a partir del configurador de efectores finales de "SCHMALZ" [13].

La configuración seleccionada incluye brida de sujeción al brazo de robot, así como un grupo de vacío de 80 l/s, por lo que su funcionamiento solo requerirá una conexión al compresor (habitual en la mayoría de industria) además del cableado de acción.

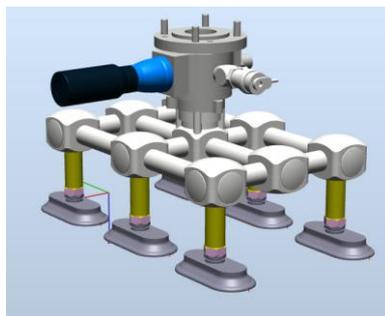


Figura 4.13 Efector final de ventosas SCHMALZ

Como vemos en la figura anterior (Figura 4.13), dispone de conexiones para seis ventosas, que han sido personalizadas añadiéndole prolongadores Parker Macho-Hembra G1/4 de 43 mm con el objetivo de alcanzar el plato sin colisionar con la pieza de trabajo.

Finalmente se han seleccionado ventosas SAOF 80x40 NBR-60 G1/4-AG de SCHMALZ con una capacidad de carga máxima de 140N cada unidad.



Figura 4.14 VENTOSA SAOF 80x40 NBR-60 G1/4-AG de SCHMALZ

- **Soporte**

El robot ira montado sobre un pedestal específico para este robot de 500 mm de altura.

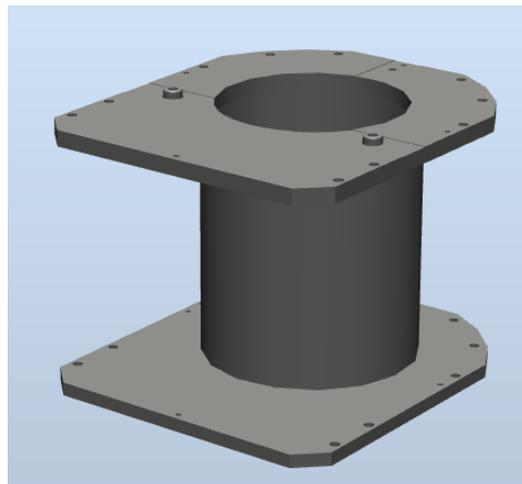


Figura 4.15 Socket_IRB1600

4.3 Diseño de piezas y útiles.

El diseño de las piezas de trabajo y resto de útiles necesarios para esta implementación han sido realizados mediante el software de diseño Autodesk Inventor 2020, siendo estas las ilustradas a continuación.

- Bisagra Tipo-1 (150 mm)

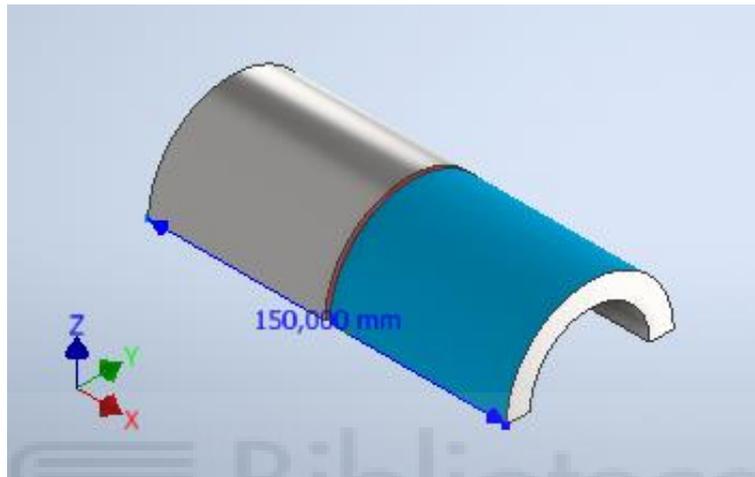


Figura 4.16 Bisagra Tipo-1

- Bisagra Tipo-2 (300 mm)

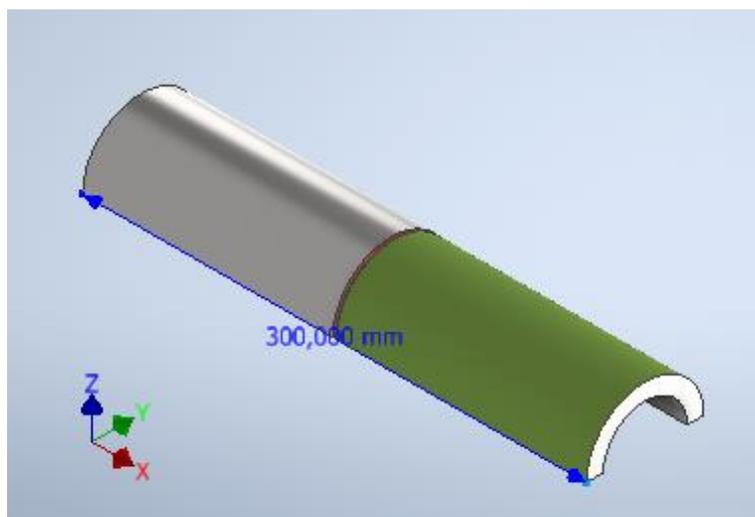


Figura 4.17 Bisagra Tipo-2

- Bisagra Tipo-3 (450 mm)

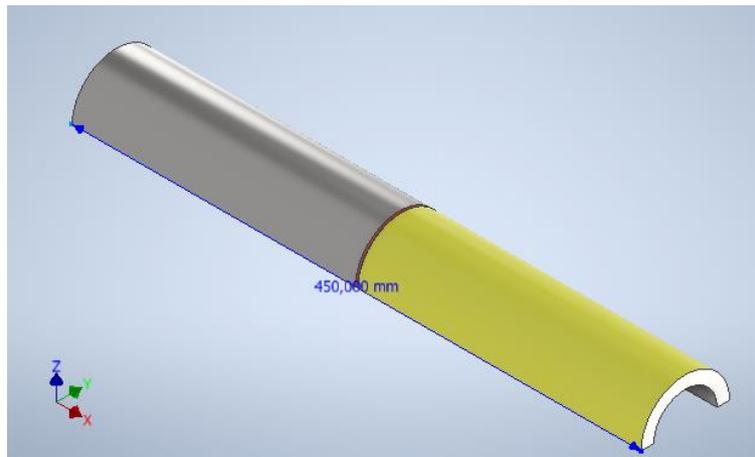


Figura 4.18 Bisagra Tipo-3

- Bandeja de transporte

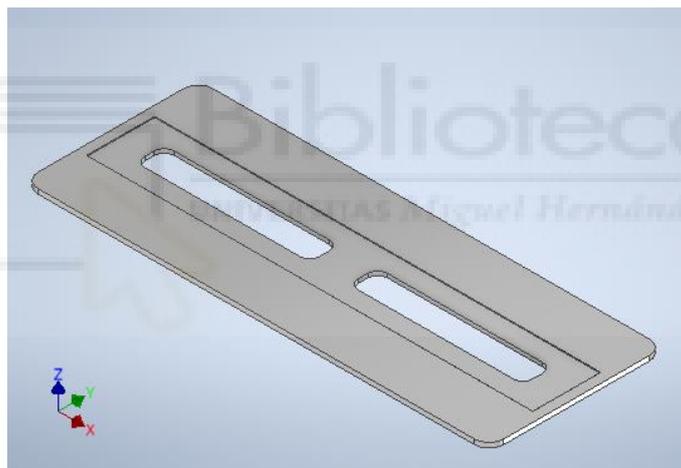


Figura 4.19 Bandeja de transporte

- Cintas de transporte

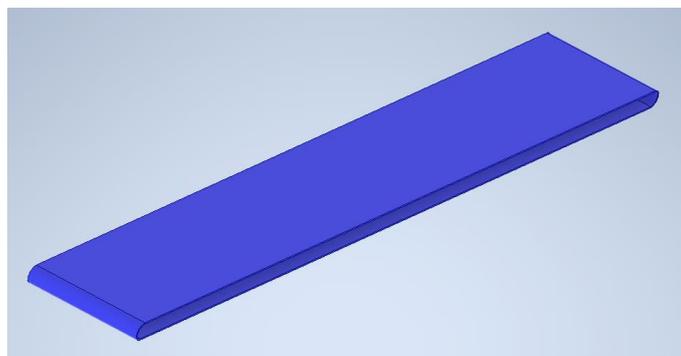


Figura 4.20 Cinta de transporte

- Caja negra-Centrador

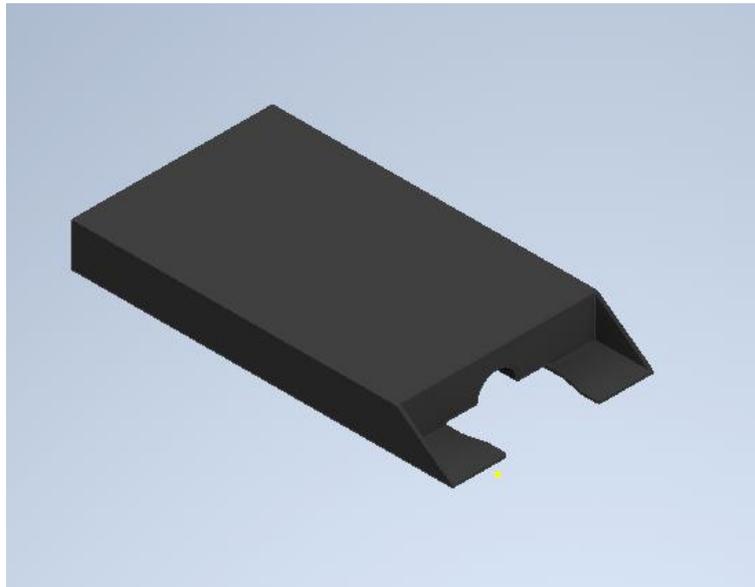


Figura 4.21 Caja negra/Centrador

- Referencia Soldadura

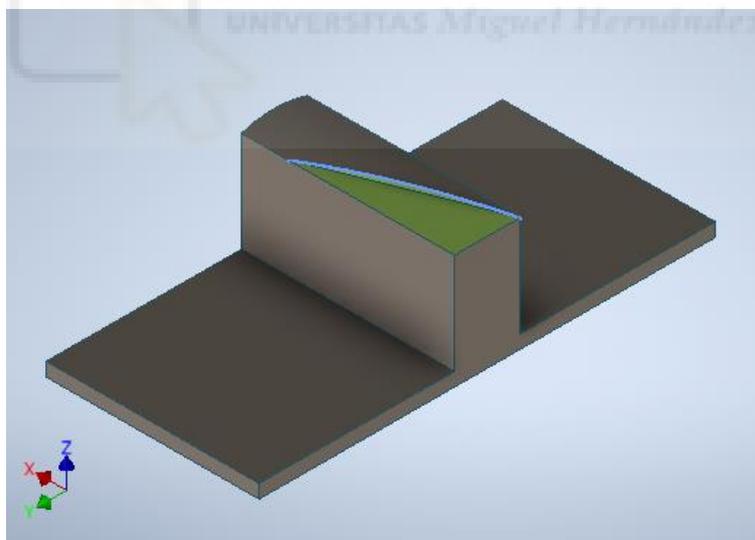


Figura 4.22 Referencia para soldadura

4.4 Programación en RobotStudio.

Puesto que tenemos flexibilidad en el diseño del espacio de trabajo, diseñaremos este in situ dentro de RobotStudio dándole prioridad a la posición de cintas y robots para que estos sean capaces de realizar sus tareas en términos de tiempos y trayectorias.

El flujo de trabajo que seguiremos es el siguiente: Procederemos a crear una estación vacía en el software, puesto que, ya tenemos seleccionado los componentes de nuestra solución y diseñado los sólidos necesarios procedemos a insertarlos en la estación, en primera instancia, conveyors, robots, herramientas y piezas de trabajo, es decir, aquellos que tienen una implicación directa con nuestra simulación y posteriormente, aquellos que simplemente aportan una representación gráfica de un elemento necesario en una implementación real pero sin relevancia en términos de programación, como por ejemplo los operarios. A continuación, crearemos el sistema de la estación con el fin de dotar a los robots de movimiento; procederemos a crear los distintos objetos de trabajo y objetivos, y, en función de estos, las trayectorias a realizar verificando su correcto funcionamiento, finalmente, dotaremos a nuestra estación de inteligencia mediante la creación de subestaciones en términos de Smart Components y programación RAPID.

4.4.1 Creación de la estación

En primer lugar, creamos una estación vacía en el menú principal de nuestro software RobotStudio en su versión 6.08 y guardaremos la estación asignándole un nombre en este caso ES_WPP (Estación Weld Pick and Place).

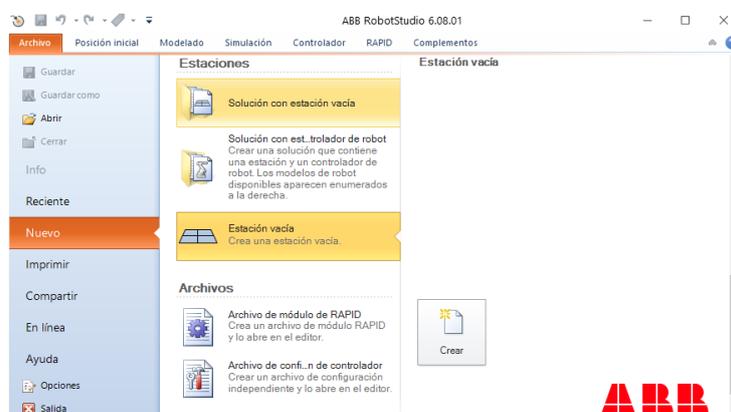


Figura 4.23 Crear estación

En este punto dispondremos solo de un espacio vacío con un sistema de coordenadas mundo (RS-WS Wobj0).

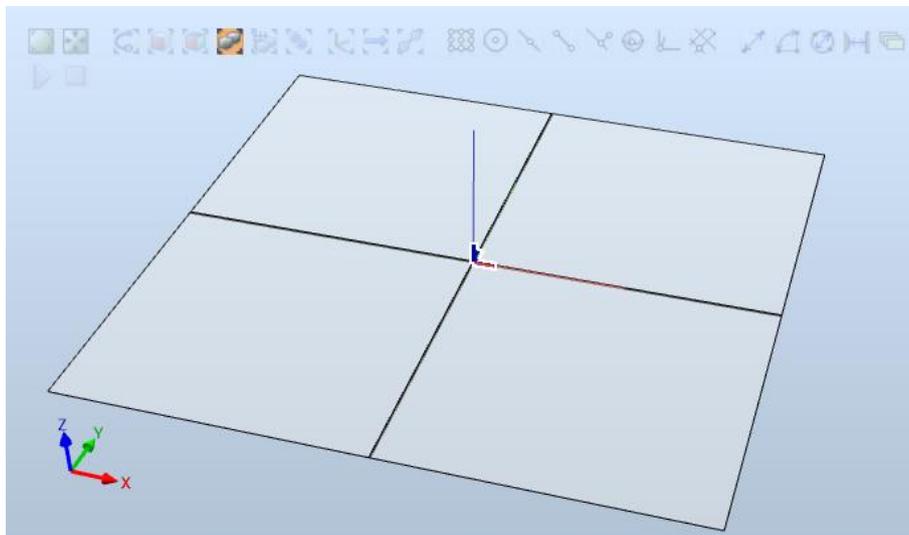


Figura 4.24 Estación vacía

4.4.2 Inserción de componentes

En este punto, insertaremos los distintos componentes que dan forma a nuestra estación virtual a partir de las bibliotecas ABB disponibles en RobotStudio en cuanto a robots y equipamiento, así como, las piezas diseñadas externamente en Autodesk Inventor o importadas de RoboApps en la pestaña complementos.

Tras la inserción de los distintos elementos de nuestra estación estos deberán ser situados en la posición deseada, para ello, haciendo click con el botón derecho del ratón sobre cualquier objeto desplegaremos el menú “Posición”.

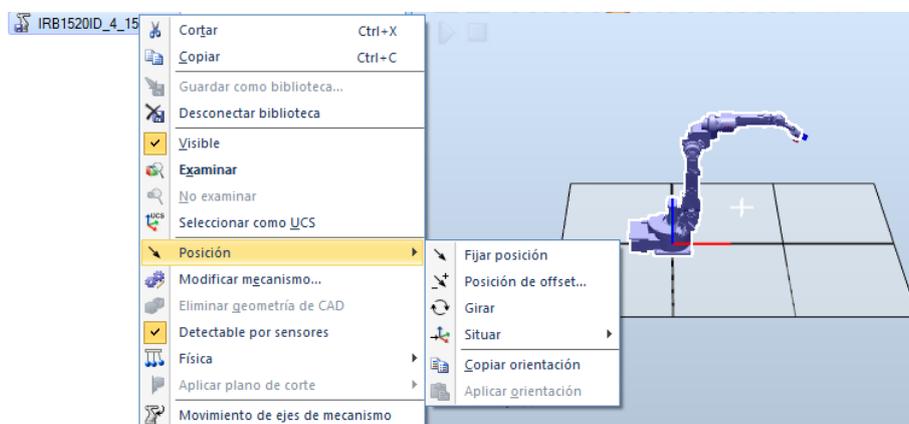


Figura 4.25 Menú posición

Como observamos en la figura anterior, el menú “Posición” nos permite:

- Fijar posición:

Sera la herramienta más utilizada durante todo el proceso, nos permite ubicar un objeto tanto en posición como orientación respecto a cualquier sistema de coordenadas seleccionados, las coordenadas se pueden introducir tanto manualmente, como haciendo uso de las distintas herramientas graficas disponibles.

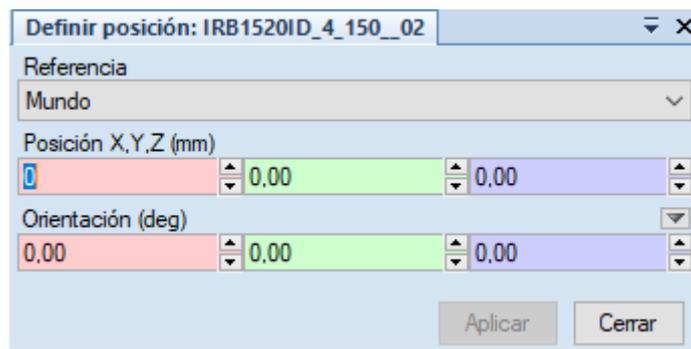


Figura 4.26 Fijar posición

- Posición de offset...:

Parecido al anterior, pero en términos relativos, nos permite trasladar/rotar el objeto respecto a un sistema de referencia seleccionado.

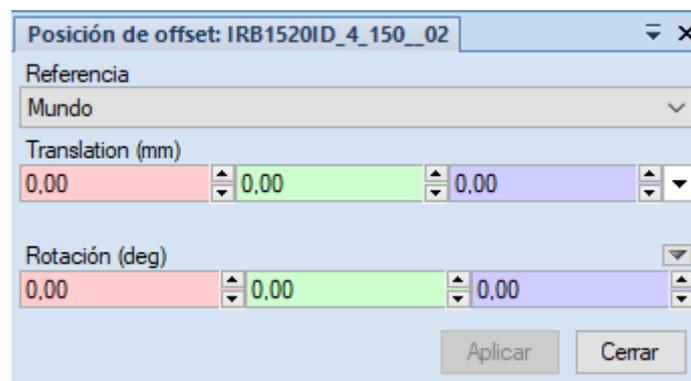


Figura 4.27 Posición de offset

- Girar:

Junto con “Fijar Posición” será el elemento más usado para las labores de posicionamiento. Aporta distintas opciones para la rotación de un objeto sobre uno o varios ejes respecto a un sistema de coordenadas seleccionado.

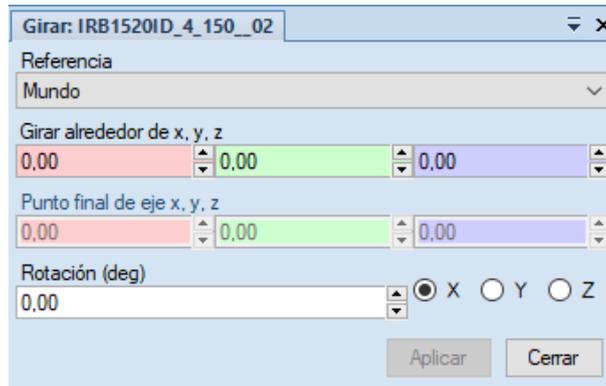


Figura 4.28 Girar

- Situar:

Presenta distintas opciones:

- Un punto: Mover objetos de un punto a otro.
- Dos puntos: Mover objetos según la relación entre planos.
- Tres puntos: Mover objetos según la relación entre tres puntos
- Base de coordenadas: Mover objetos al marco seleccionado.
- Dos bases de coordenadas: Mover objetos a un segundo marco usando el primero como referencia.

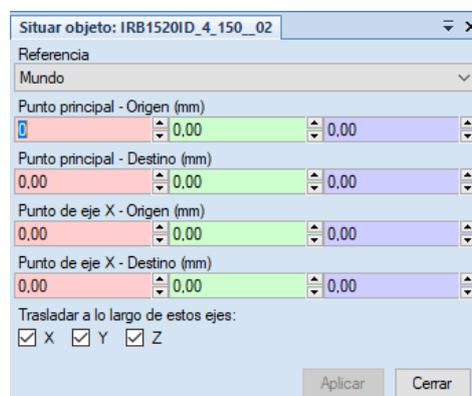


Figura 4.29 Situar

Vista las distintas maneras de ubicar lo componentes, procedemos a cargar y posicionar los equipos en el siguiente orden:

- Comenzamos ubicando las cintas transportadoras destinadas al aporte y retirada del material.

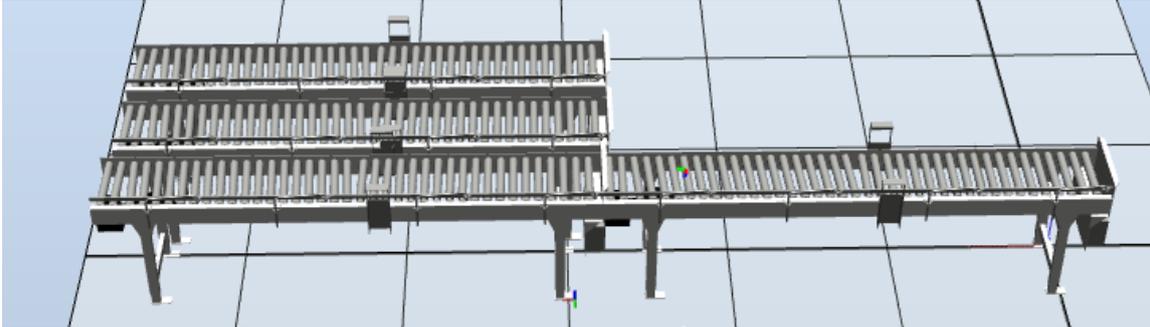


Figura 4.30 Posición conveyors

- Los conveyors son cubiertos con bandas que garantizan la precisa transmisión del movimiento y la conductividad en la fase de soldadura.

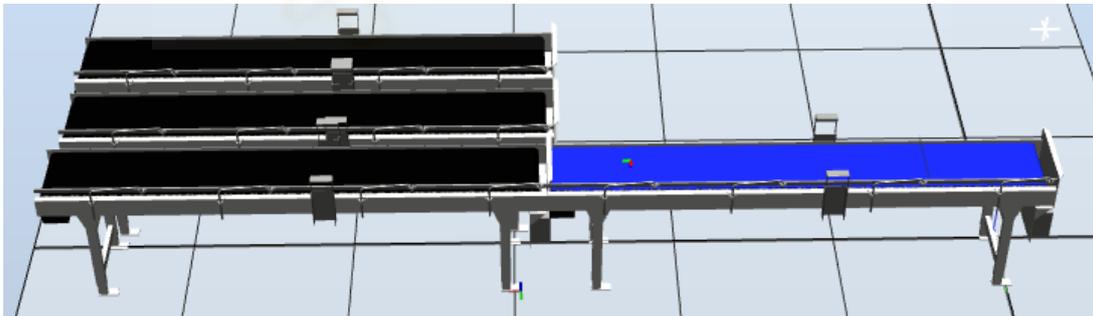


Figura 4.31 Posición cintas

- Acomodación del robot IRB1600 destinado al transporte en una posición que facilite la ejecución de las trayectorias.

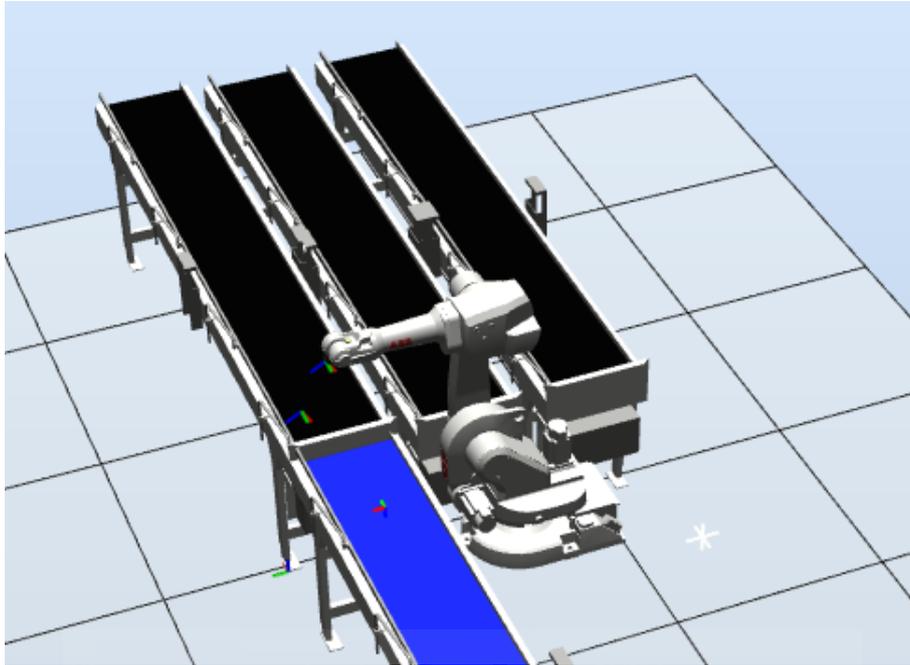


Figura 4.32 Posicionamiento IRB 1600

- Posicionamiento del robot IRB1520ID destinado a la soldadura.

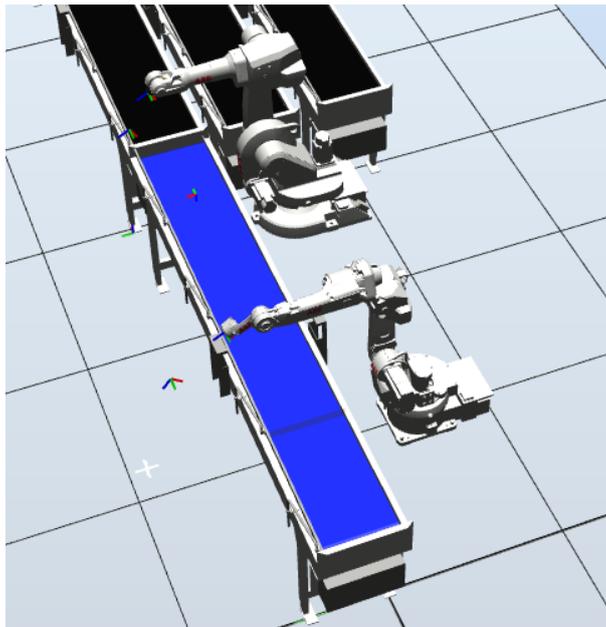


Figura 4.33 Posicionamiento IRB 1520ID

- Conectamos la antorcha de soldadura según lo visto en el apartado 3.2.2

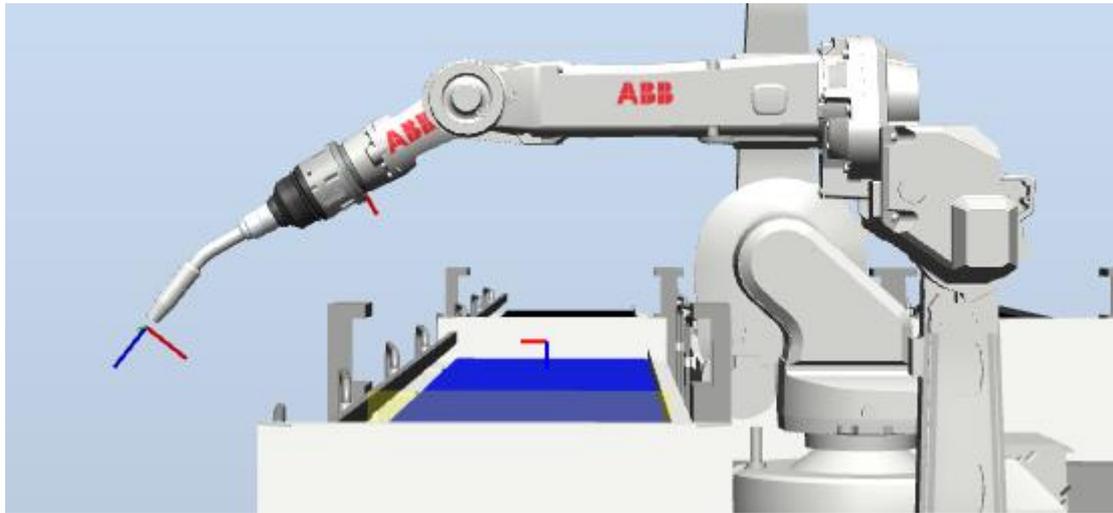


Figura 4.34 Conexión herramienta de soldar con IRB 1520ID

- Conectamos las ventosas de succión a su robot.

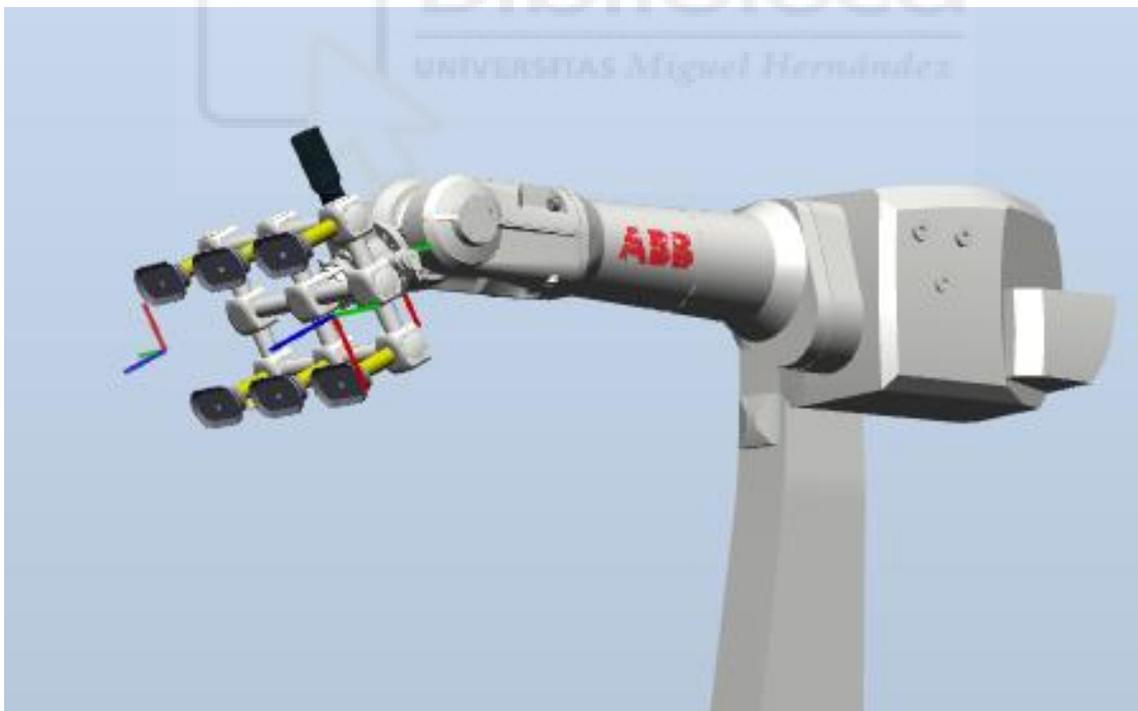


Figura 4.35 Conexión efector final de ventosas con IRB 1600

- Conectamos y posicionamos el alimentador de hilo Fronius al Link3 del robot de soldadura.

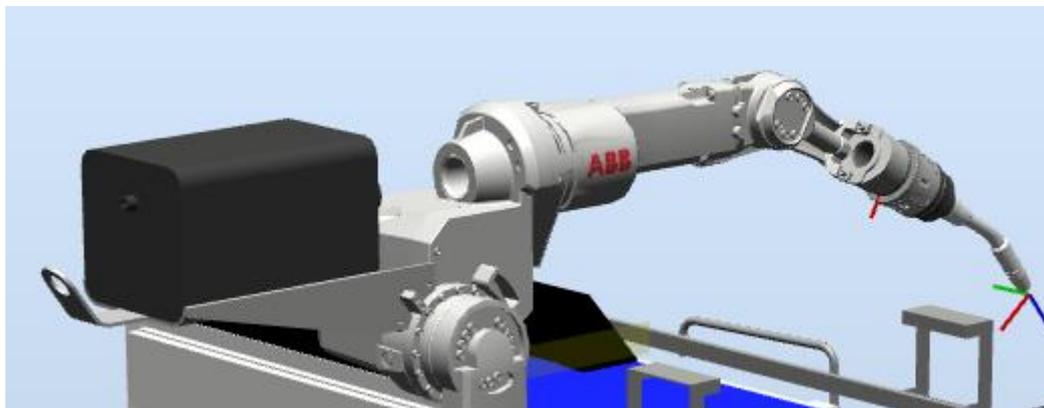


Figura 4.36 Conexión wire feeder

- Posicionamiento del robot IRB 1600 sobre su pedestal.

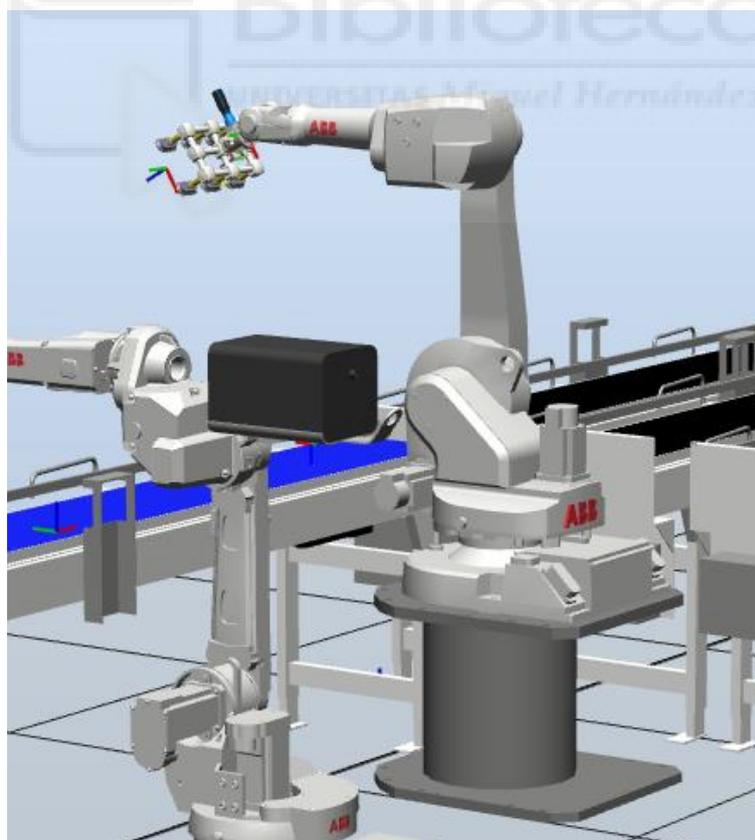


Figura 4.37 IRB 1600 sobre pedestal

- Colocación del perímetro de seguridad.

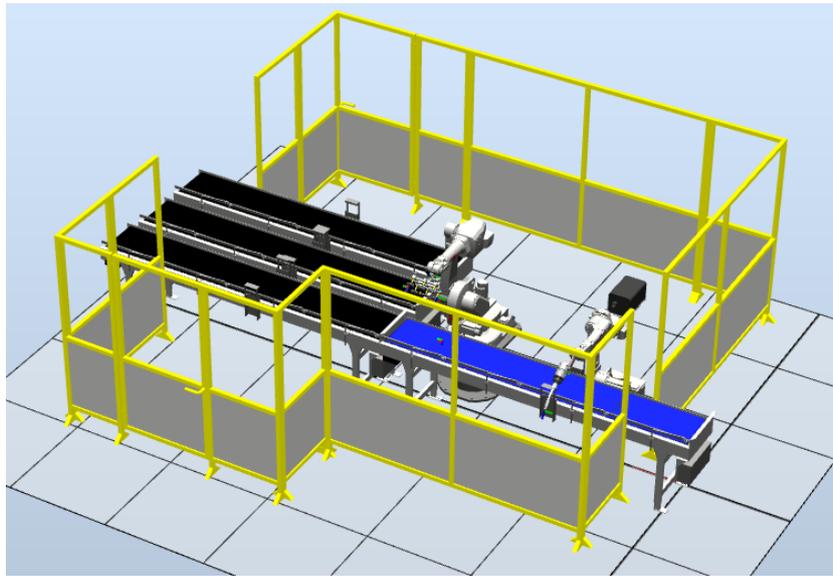


Figura 4.38 Colocación perímetro de seguridad

- Colocación del equipo del controlador y equipo de soldadura (Control, estación de limpieza y bidón.)

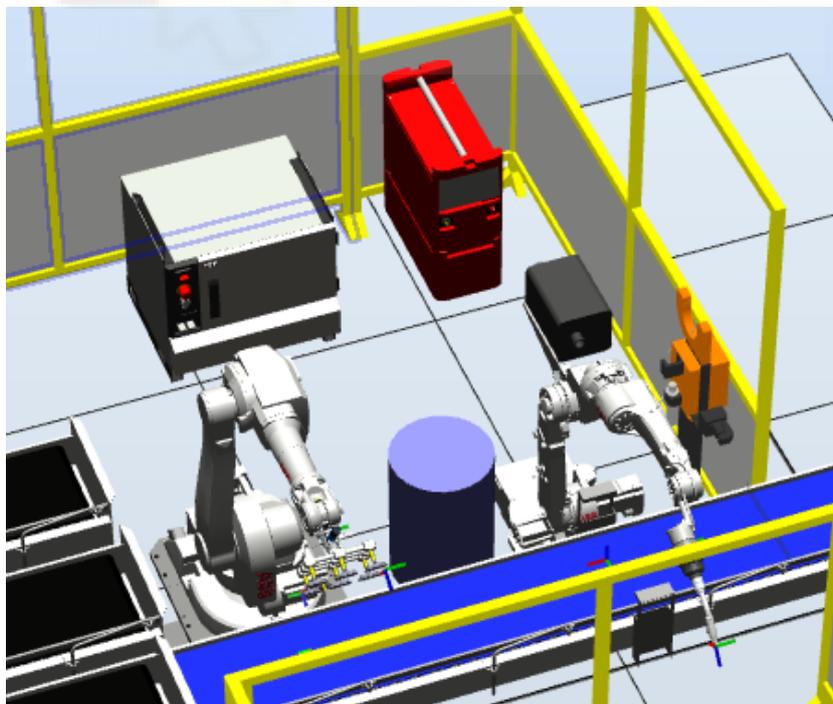


Figura 4.39 Equipo de soldadura

- Componentes necesarios: Operario, escritorio, silla, ordenador, FlexPendant.

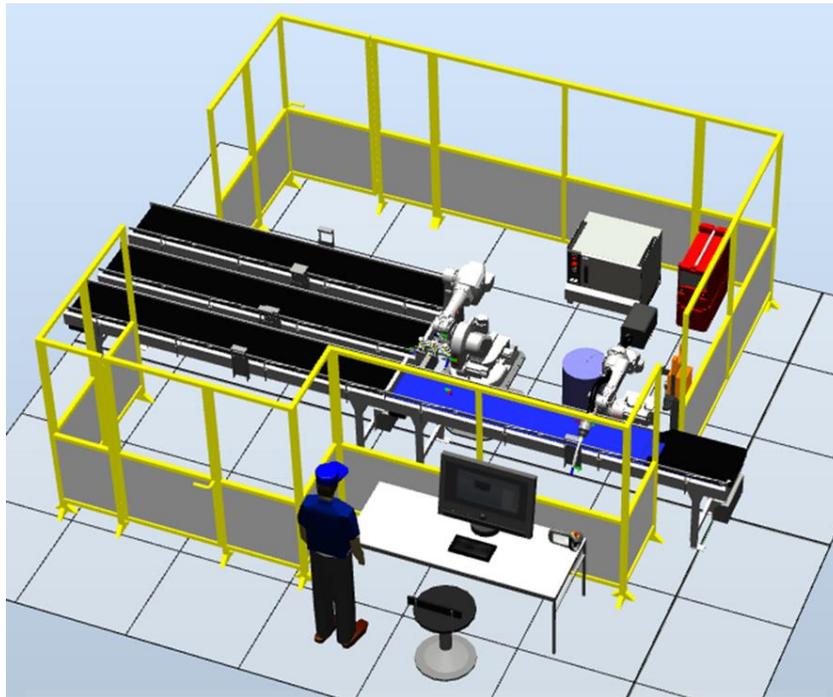


Figura 4.40 Componentes adicionales

- Simulación del hilo de alimentación en la estación de soldadura.

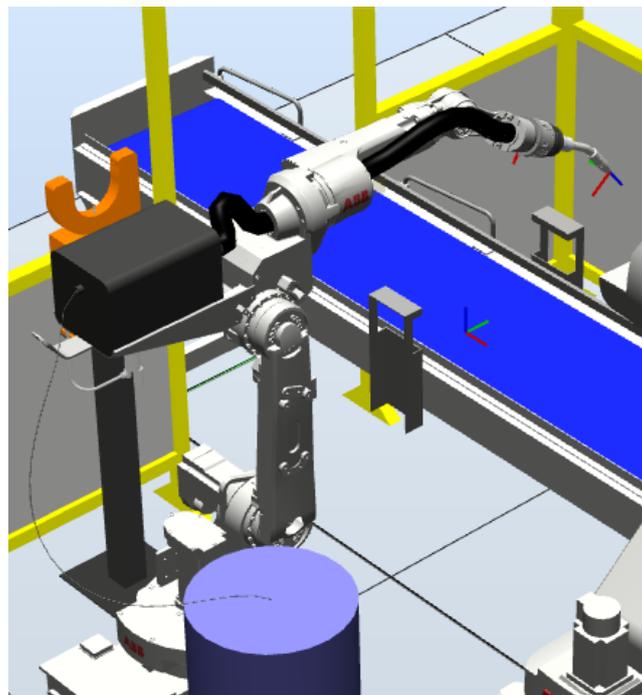


Figura 4.41 Simulación hilo de alimentación

- Colocación Caja negra/Centradora: Representa la entrada de piezas desde un proceso anterior con un elemento centrador que obliga a las piezas a entrar de forma correcta hacia el proceso.

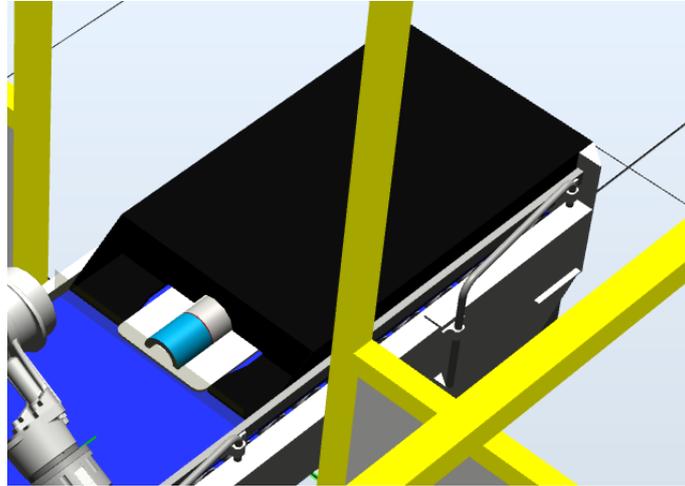


Figura 4.42 Detalle caja negra centradora

Obteniendo la siguiente vista en planta de la estación diseñada:

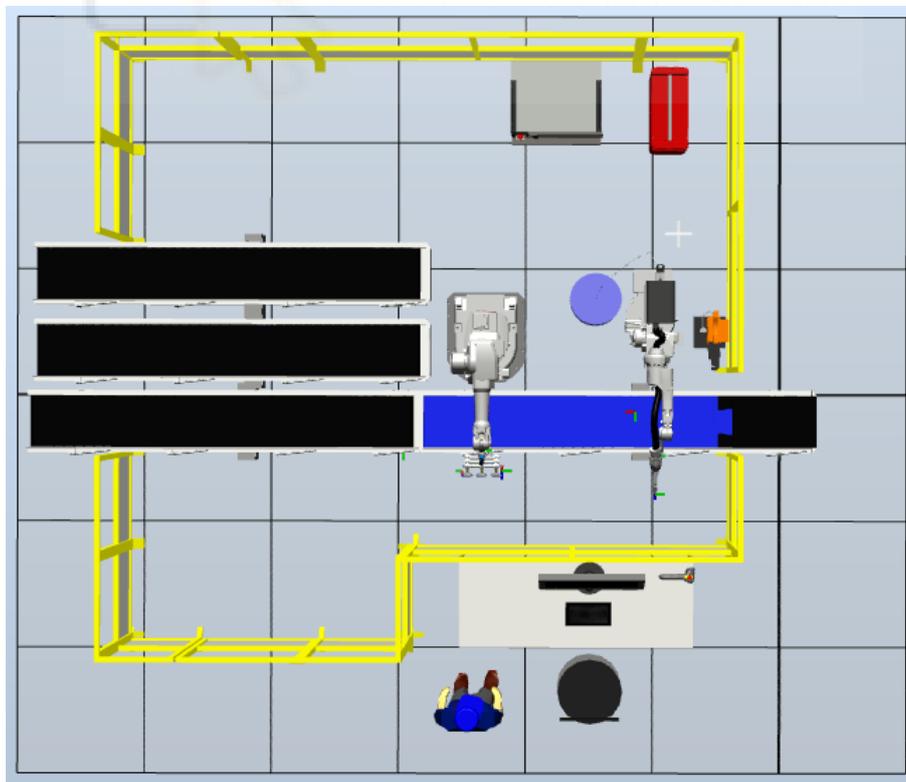


Figura 4.43 Vista en planta de la estación

4.4.3 Creación de sistema

Tras la inserción de los robots en nuestra estación, estamos en condiciones de crear el sistema. De esta manera RobotStudio selecciona automáticamente las opciones requeridas para el manejo de los robots mediante el controlador, además se pueden cargar manual y adicionalmente otras opciones como el idioma español, tarjetas externas como “DeviceNet”, controles para multitareas “MultiTasking” [3], seguimiento “Tracking” y un largo etcétera que dependerá de nuestras necesidades.

En este caso, puesto que se manejará mediante un controlador IRC5 y según el manual es capaz de controlar 3 robots simultáneamente y hasta 16 tareas independientes para tareas de supervisión, se creará un único sistema que englobe los dos robots.

Para ello, pincharemos en la opción “sistema de robot\desde diseño” y RobotStudio se encargará de la correcta configuración, en principio solo añadiremos “Idioma\Español”.

4.4.4 Diseño de la estación WPP

Este punto, comenzamos el diseño de nuestra estación a un nivel de programación que dotara a nuestro sistema de la “inteligencia” necesaria para cumplir sus objetivos.

Se ha diseñado bajo una estructura jerárquica en la cual un “Nivel superior” controla dos subestaciones responsables de las operaciones de soldadura y pick-and place respectivamente.

4.4.4.1 Subestación soldadura

Esta subestación, recibe a su entrada bisagras de “tipo 1”, “tipo 2” o “tipo otras”, todas ellas con un diámetro de 70 mm y longitudes definidas para el tipo 1 y 2 y “aleatoria” en el caso de “tipo otras”.

Las piezas son desplazadas por una cinta transportadora a una velocidad constante de 200 m/s y una cadencia aleatoria.

En cualquier caso, la estación debe detectar cuando la pieza alcanzara el punto inicial de la soldadura y ejecutar la trayectoria que permita la correcta soldadura sin parada de la cinta.

La soldadura debe cubrir 1/5 de la longitud del arco que forma la bisagra.

Para ello, se dependerá directamente de la precisión del sensor y el contador disponible, ya que se calibrarán y sincronizarán los movimientos del robot a partir de las señales de estos.

4.4.4.1.1 Objetivos y Trayectorias

En RobotStudio existen numerosos complementos destinados a facilitar las labores de soldadura en cuanto a diseño de trayectorias y control, de hecho, se podría completar un trabajo solo estudiando las distintas alternativas y su funcionamiento, además, estos complementos pueden suponer un sobrecoste al necesitar distintos equipos.

En este caso, se han diseñado los objetivos y trayectorias de una forma manual, mediante la interfaz de RobotStudio y código Rapid.

El sistema recibe en su entrada bisagras semicirculares de diámetro 70 mm y distintas longitudes, posicionadas sobre una bandeja; Y, deben ser soldadas por su centro sin detener la cinta transportadora.

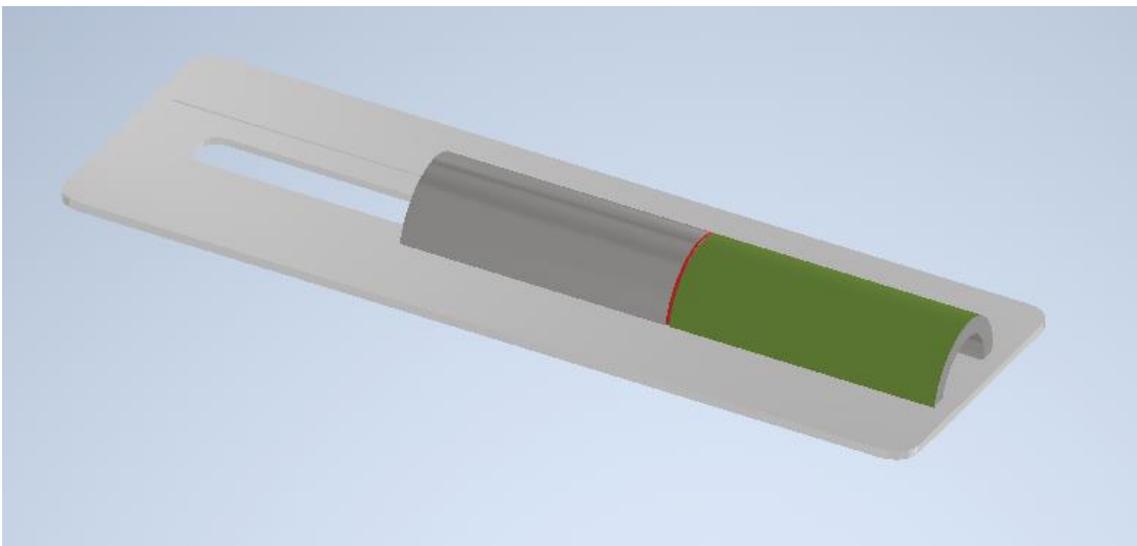


Figura 4.44 Ejemplo Pieza (Rojo línea de soldadura)

La soldadura se debe realizar a una velocidad de 5 m/s y debe ser perpendicular a la superficie en todo momento.

Para ello la antorcha de soldadura sujeta al robot deberá describir una trayectoria semi elíptica a una velocidad determinada que permita el acople y sincronización entre el avance de la cinta y el TCP (en el eje X en este caso).

Durante la investigación para este proyecto se valoró utilizar la herramienta de “mecanismos/transportador”, elemento visto previamente en el punto __, pero se descartó rápidamente por una serie de limitaciones, este componente, en las instalaciones más simples al menos, hace las operaciones de seguimiento y sincronización de forma perfecta, pero, solo puede trabajar con un objeto y con una cadencia predefinida, alejándonos de nuestra búsqueda de flexibilidad tanto en tamaño de piezas, como intervalo entre operaciones, además no se ha podido editarle entras o salidas virtuales con las que operar.

No obstante, hablamos de una herramienta potentísima y muy valorable para estudios futuros en profundidad y explotación de su potencial.

Por los motivos expuestos, se optó por crear las trayectorias en RobotStudio, para ello se diseñó un bloque de referencia de manera que facilitara la creación mediante las opciones disponibles.

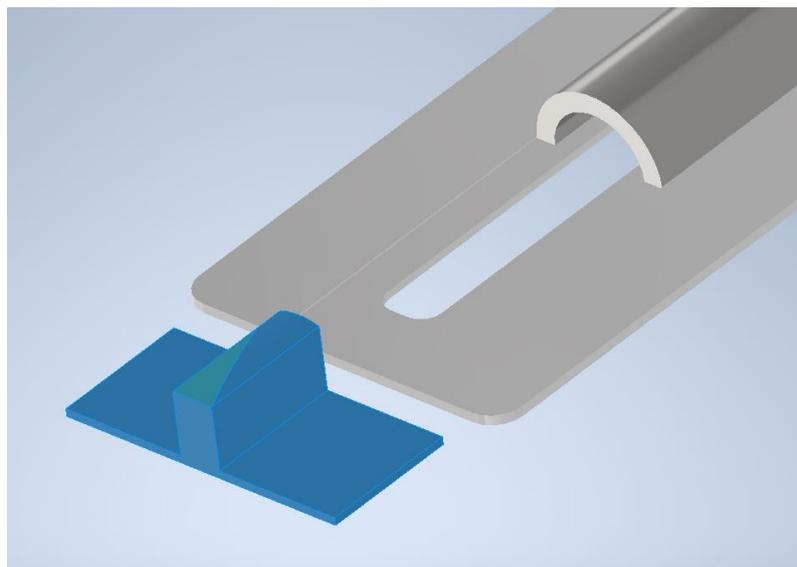


Figura 4.45 Referencia Soldadura

Este bloque virtual representa la sección de bisagra de 72 grados que debe ser soldada

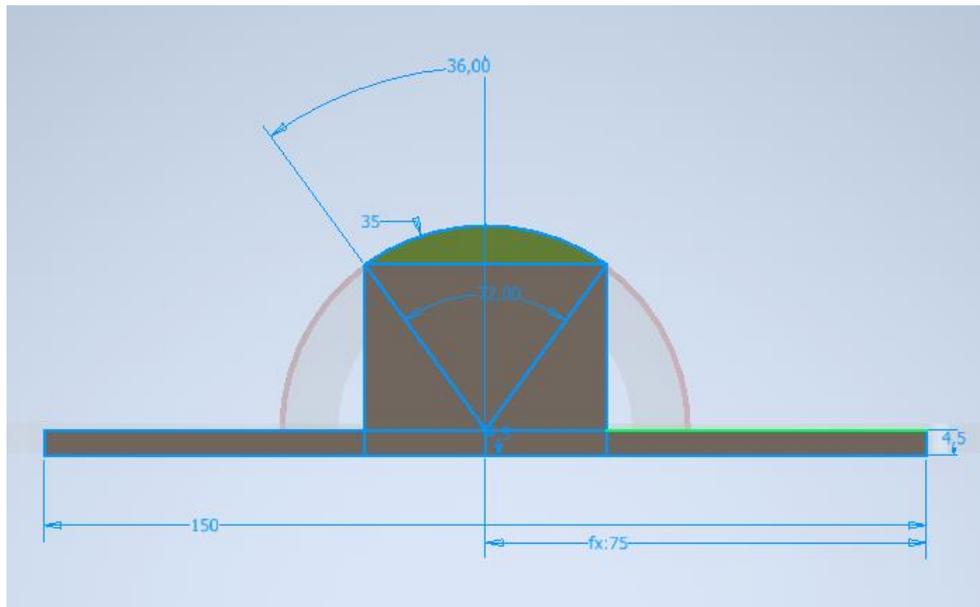


Figura 4.46 Cotas Referencia Soldadura

A través de distintos experimentos, se diseñó este útil para que la trayectoria de soldadura se correspondiera con 110 mm de avance de la cinta

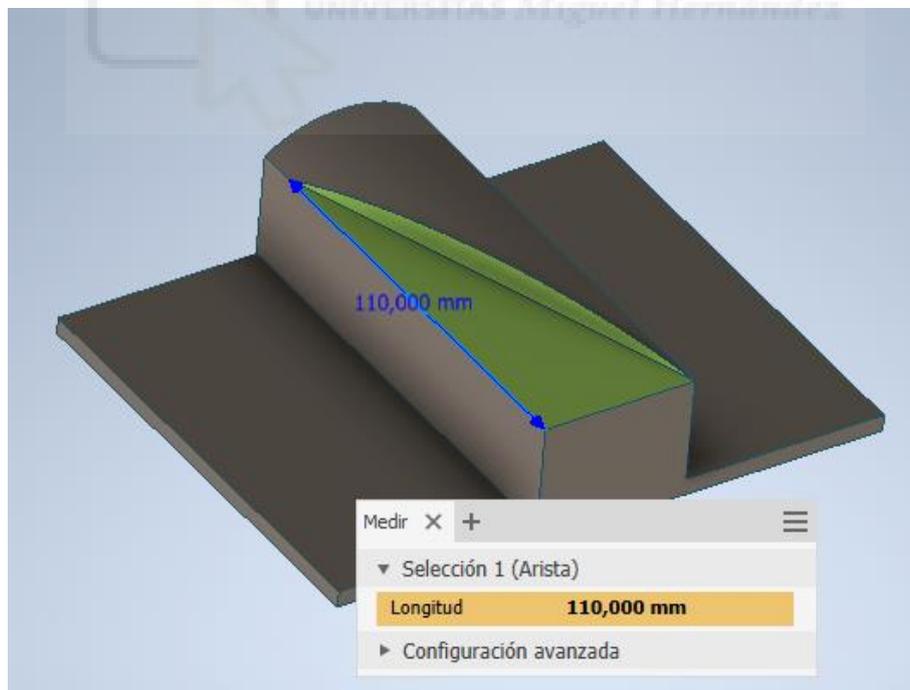


Figura 4.47 Detalle avance Referencia Soldadura

Como se observa en la siguiente figura, un corte que une los puntos inicial y final origina la “trayectoria” que deberá seguir nuestra herramienta.

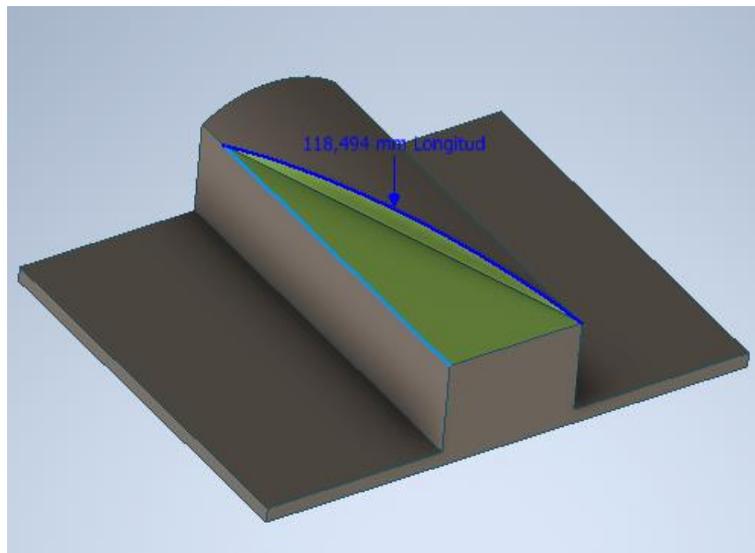


Figura 4.48 Detalle longitud de trayectoria

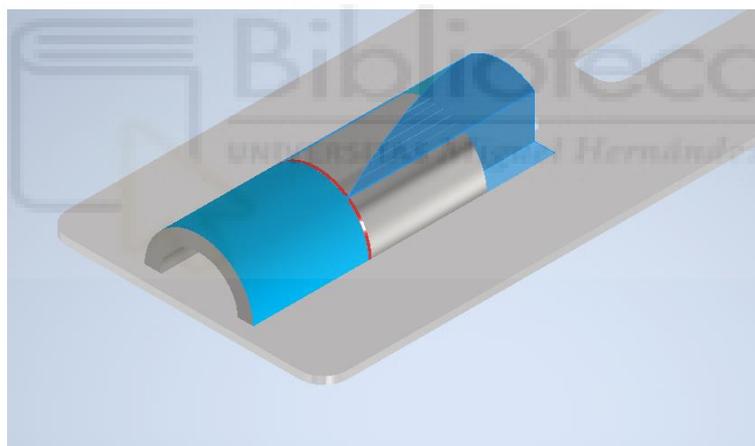


Figura 4.49 Pieza y referencia en posición inicial

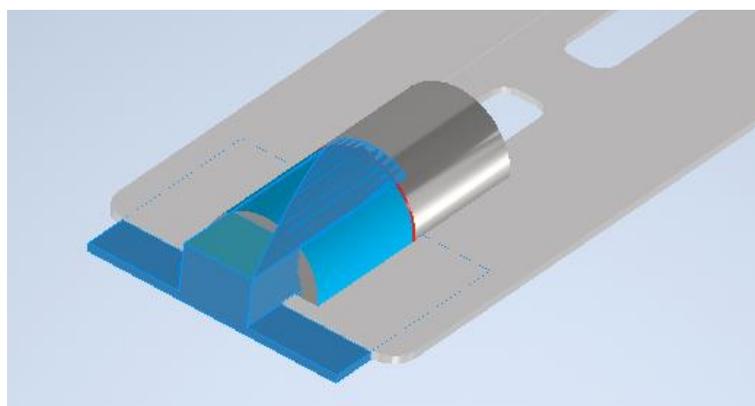


Figura 4.50 Pieza y referencia en posición final

Con los datos obtenidos y asumiendo velocidades constantes (debido a las elevadas aceleraciones disponibles), se procede a calcular la velocidad a la que debe desplazarse el TCP de la herramienta para una correcta sincronización.

Partiendo de que se debe soldar un cordón de 43,982 mm y, asumiendo una velocidad de 80 mm/s de soldadura, obtenemos un tiempo de activación de 0,549 s en cada ciclo.

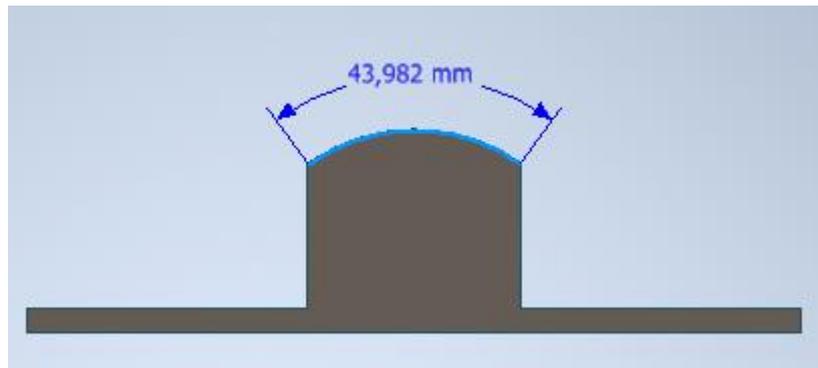


Figura 4.51 Longitud cordón

Sabemos por las acotaciones CAD que la distancia de la trayectoria es de 118.494 mm, que deberán ser recorridos en 0,549 s, esto implica una velocidad constante de 215,5 mm/s en el TCP.

Colocamos la referencia en la estación, en una posición cómoda para el robot.

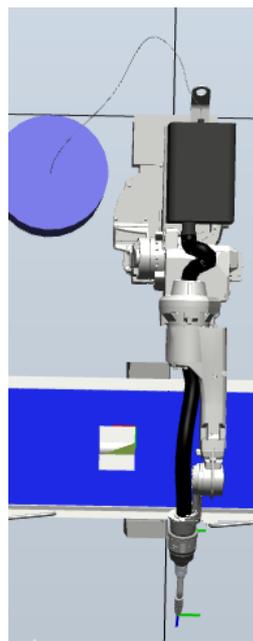


Figura 4.52 Modelo de referencia en la estación

En primer lugar, programaremos un Objeto de trabajo, para ello nos dirigiremos la cinta de “programación de trayectorias” ubicada en la pestaña “Posición inicial” y en “Otros”, presionamos sobre crear “objeto de trabajo”; le damos nombre “WO_WELD” y, puesto que queremos vincularlo a la geometría de referencia, pinchamos sobre una esquina con la opción de “ajuste final” activado.

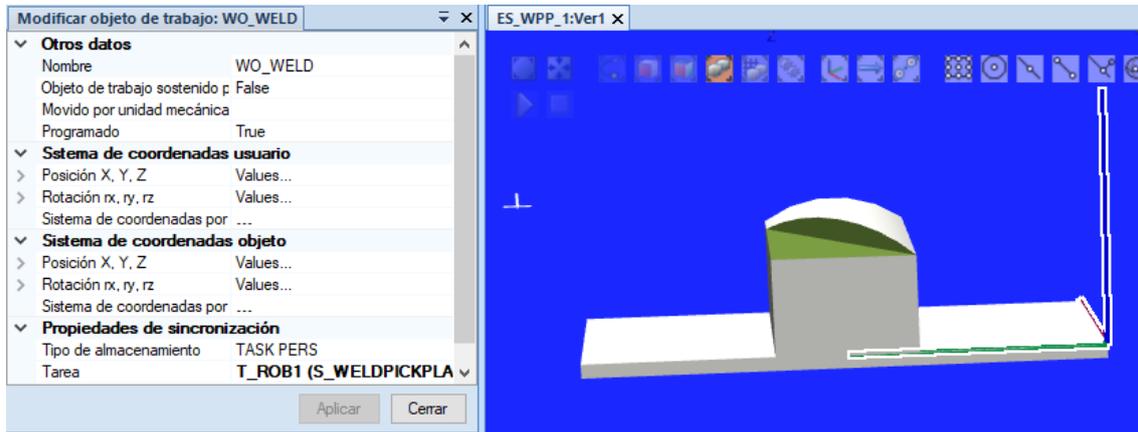


Figura 4.53 Creación Objeto de trabajo

En esta ocasión, comenzamos a diseñar la trayectoria principal en primer lugar, para ello nos dirigiremos la cinta de “programación de trayectorias” ubicada en la pestaña “Posición inicial” y en “Ruta” seleccionamos trayectoria automática, prestando atención a tener activado el objeto de trabajo y la herramienta correcta, hacemos click sobre la arista de trayectoria en el modelo de referencia obtenemos una trayectoria muy parecida a lo que buscamos.

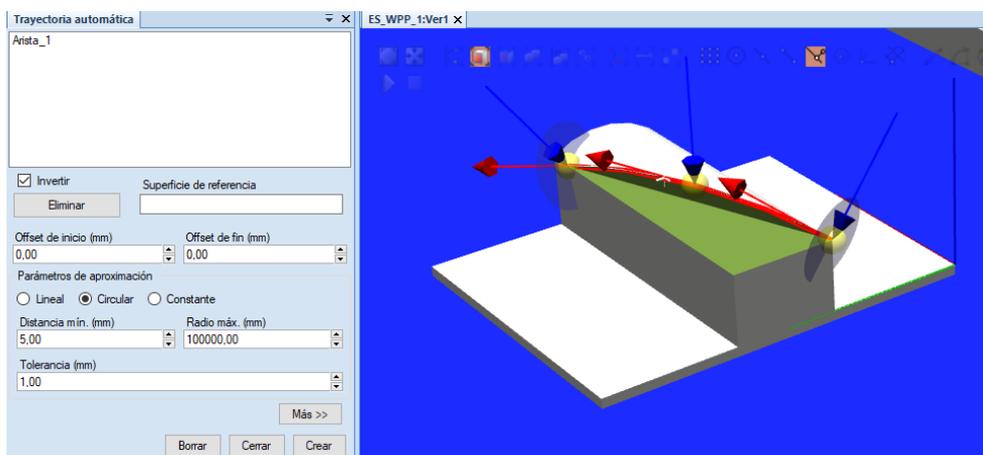


Figura 4.54 Trayectoria automática

Presionamos “Crear” y renombramos como “Path_WELD”

En este punto hemos creado los objetivos que configuraran la trayectoria, pero por imposición de la técnica siempre deben ser perpendiculares a la superficie, con este fin, generamos una trayectoria auxiliar que eliminaremos más tarde, para ello hacemos uso de la sección plana final y operamos de la misma manera.

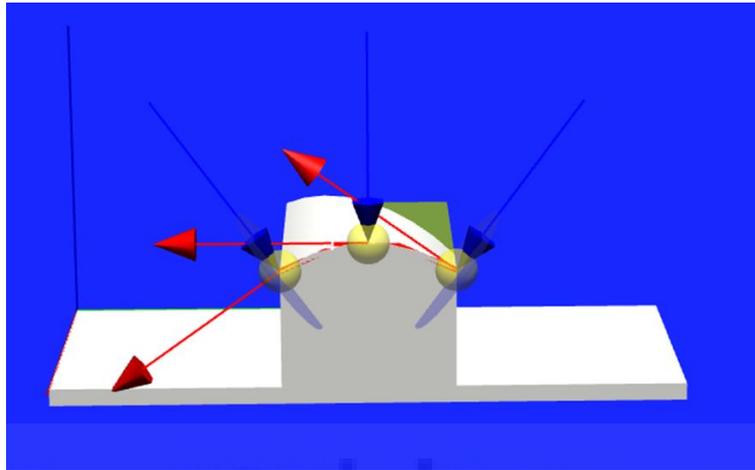


Figura 4.55 Trayectoria auxiliar

Como se observa en la figura anterior 4.55 los objetivos creados, ahora son perpendiculares a la superficie.

Utilizamos las opciones “Copiar orientación” y “Aplicar orientación” entre estos puntos y sus correspondientes en el Path_WELD asegurándonos así que nuestra trayectoria es perpendicular a la superficie (figura 4.56).

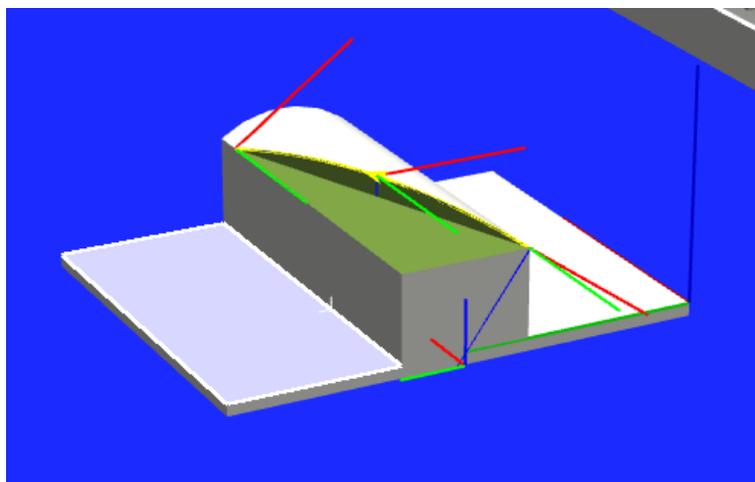


Figura 4.56 Trayectoria corregida

Realizando pruebas con la alcanzabilidad y la comodidad de la herramienta se realiza un giro de 24 grados en el eje z para que este más próximo a la posición “natural” de la antorcha. Además, este tramo es el que requiere máxima precisión ya que recorre la trayectoria que debe ser soldada, para ello se modifican las instrucciones “fine” y la velocidad específica que se programará posteriormente en Rapid.

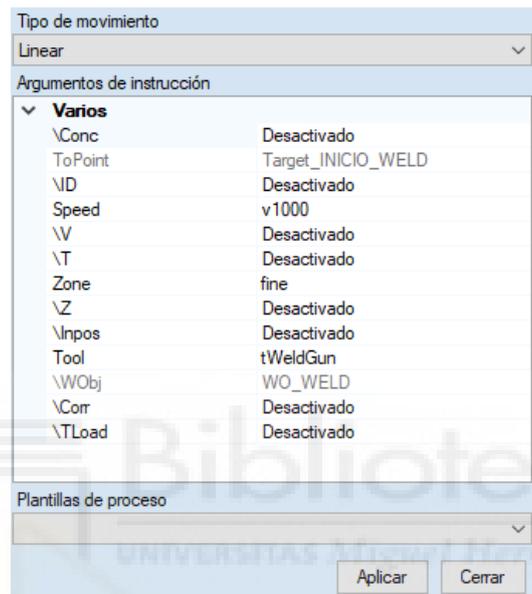


Figura 4.57 Menú Modificar instrucciones

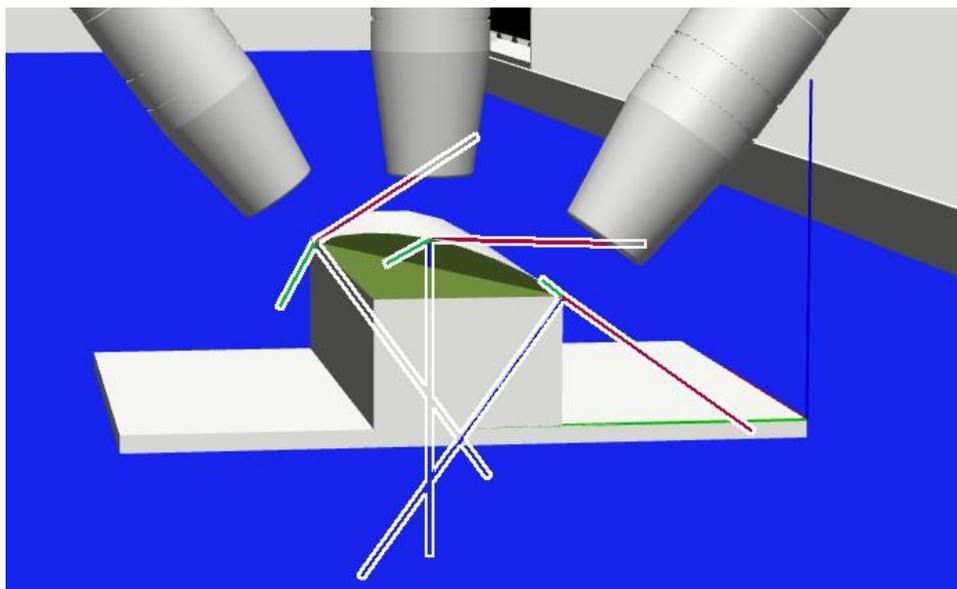


Figura 4.58 Trayectoria y alcanzabilidad

Con las herramientas vistas en el capítulo 3 de metodología, en concreto copiar, pegar y fijar posición, copiamos los puntos de salida y entrada, y los desplazamos en torno a los 50 mm de forma que puedan trabajar en ciclo sin colisionar con las piezas.

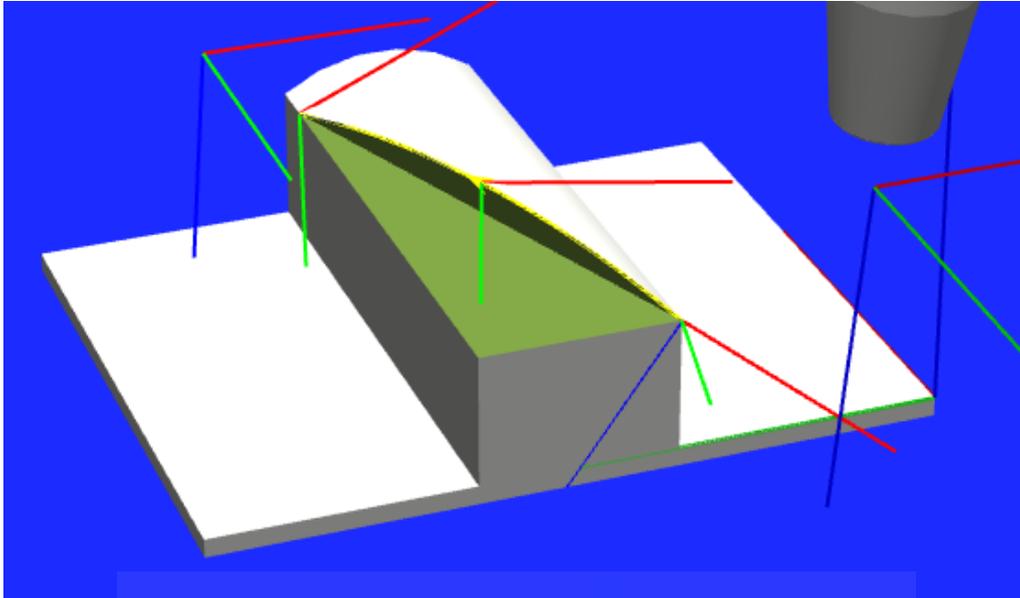


Figura 4.59 Puntos de entrada y salida

Tras darles un nombre para facilitar su identificación, se arrastran y ordenan en el Path_WELD de manera que ejecute la trayectoria deseada.

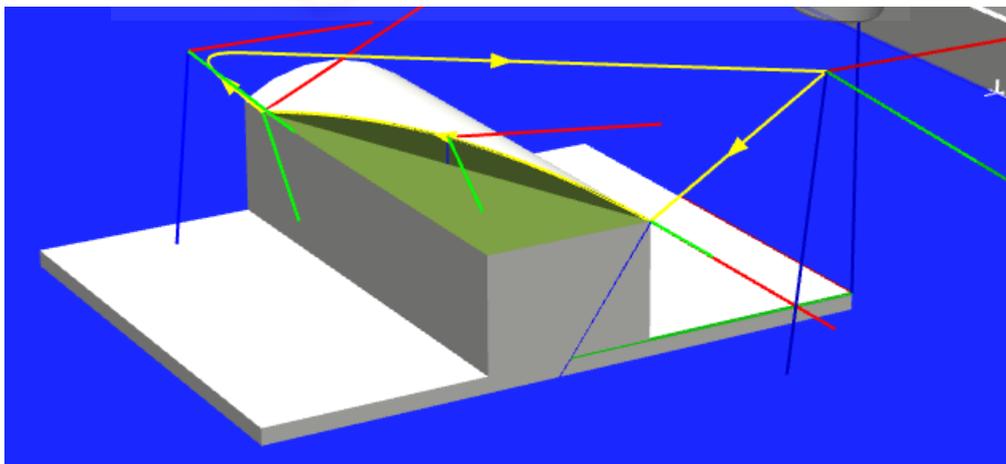


Figura 4.60 Trayectoria "Path_WELD"

Se configuran los ejes de forma automática si, con el botón derecho pulsamos sobre la trayectoria en el árbol de "Trayectorias y puntos" y dentro de "configuración

automática” ejecutamos “todas las instrucciones”; No obstante, de encontrar fallos se pueden modificar las configuraciones de cada punto de forma independiente.

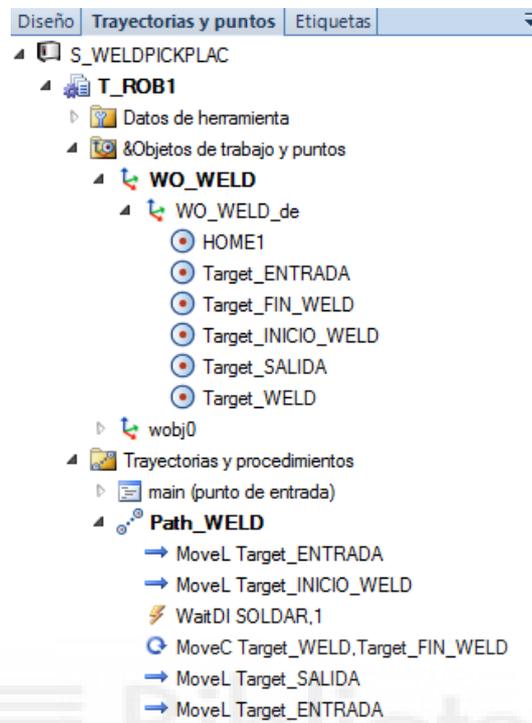


Figura 4.61 Puntos y Trayectorias

Tras crear los puntos de destino y las rutas, procedemos a sincronizar con Rapid, para su supervisión y modificación posterior.

4.4.4.1.2 Smart Components (SC)

- **SC SINCRONIZACION_TAREAS/ SC SOLDADURA**

El SC SOLDADURA, se encuentra en el marco de la sincronización de tareas siendo el responsable de la correcta ejecución de la tarea de soldadura, aportándole a la estación gran flexibilidad permitiéndole trabajar con cualquier tamaño de pieza. Su función consiste en predecir cuándo se debe activar la soldadura, para ello, calcula el tiempo que la pieza está activando el sensor planar y en base a los datos obtenidos parametriza el momento de ejecución.

Como se observa en la siguiente figura, recibe como entrada la señal digital proveniente del detector y activa una pulso como señal de salida denominada "SOLDAR" en el momento que debe dar comienzo el proceso.

En cuanto a sus componentes, está formado por los seis siguientes:

- Stopwatch:

Se trata de un cronometro, y de su precisión dependerá la precisión del conjunto, recibe como entrada la señal del detector que activa el contador por su entrada "Active" y un nuevo ciclo a traves de su entrada "Lap" y se reinicia a través de la entrada "Reset" cuando la señal del detector se vuelve "0", como salida, direccionamos su valor en la señal analógica "LapTime" que nos indica el tiempo que ha durado el ciclo, y, en combinación con la velocidad de la cinta podemos deducir la dimensión de la pieza que ha pasado el detector.

- Expression:

Este componente nos permite realizar operaciones matemáticas, lo utilizaremos para, en base a el tiempo obtenido en el Stopwatch, parametrizar el tiempo que debe tardar en activarse la señal de soldadura para que el punto medio de la pieza alcance el punto inicial de soldadura en la trayectoria programada en el robot, en este caso se ha establecido su propiedad "Expression" como $3 + \text{TIEMPO} * 0.5 - \text{TIEMPO} - 0.072$, lo que se traduce como 3 segundos hasta que la pieza alcanza el punto inicial de la trayectoria, más la mitad del tiempo que la pieza ha activado el sensor para alcanzar el punto medio, menos el tiempo dedicado a la detección y menos 0.06 segundos, dato obtenido experimentalmente en base a distintas simulaciones y teniendo en cuenta el paso configurado en la simulación. Finalmente extraemos el resultado en el atributo "Result".

- LogicSplit:

Componente que nos permite emitir distintas señales en función de la entrada obtenida, en este caso recibe como entrada la señal del detector y tomamos como salida "PulseLow" que enviara un pulso cuando la señal del detector cambie de "1" a "0" indicando que la pieza ha terminado de pasar por el detector.

- LogicSRLatch:

Funciona como un interruptor, establece su salida "Output" en "1" cuando recibe la señal de que el detector se ha activado y cambiará a "0" al mismo tiempo que activa la señal "InOutput" cuando sea reseteado mediante su entrada "Reset", que a su vez será responsable de resetear el reloj del Stopwatch.

- LogicGate[AND]:

Puerta lógica configurada como "AND", activara su salida cuando reciba la señal "Output" que proviene del LogicSRLatch indicando que se está detectando una pieza y el pulso recibido desde el LogicSplit indicando que ha dejado de detectarse, finalizada pues la detección, su salida es redirigida al Reset del LogicSplit.

- LogicGate[OR]:

Esta puerta lógica introduce en su propiedad "Delay" el tiempo calculado por el complemento "Expression" y comienza su cuenta atrás cuando recibe la señal de fin de pieza que obteníamos con el LogicGate[AND] enviando la señal de salida definitiva en el momento pertinente.

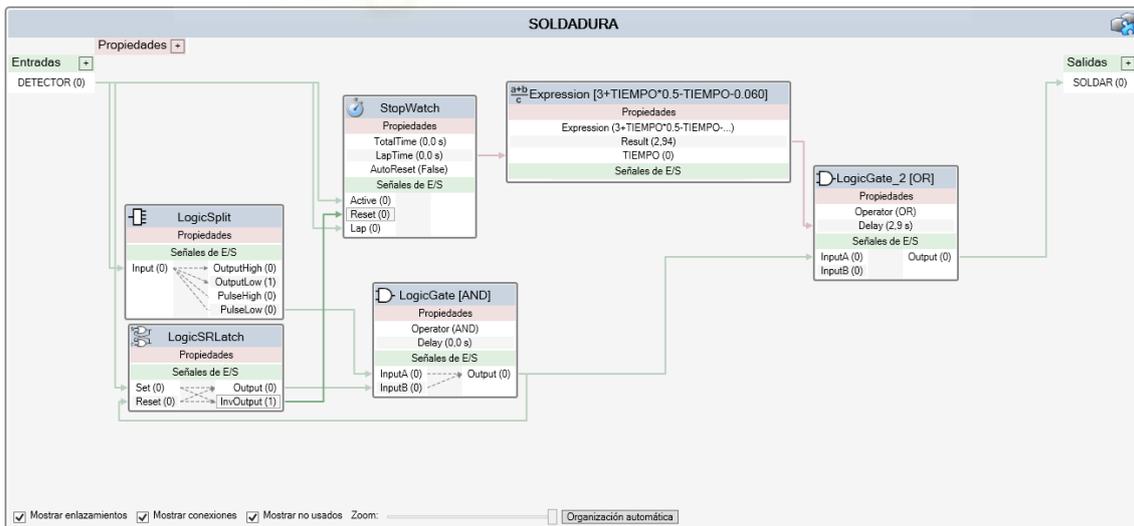


Figura 4.62 SC SOLDADURA

4.4.4.1.3 Lógica, Entradas y Salidas

Cuando la estación se pone en marcha el robot deberá ponerse en posición de espera para comenzar a soldar cuando reciba una señal, para tal efecto, debe haber una conexión entre el componente inteligente y el sistema de nuestro robot "S_WELDPICKPLACE".

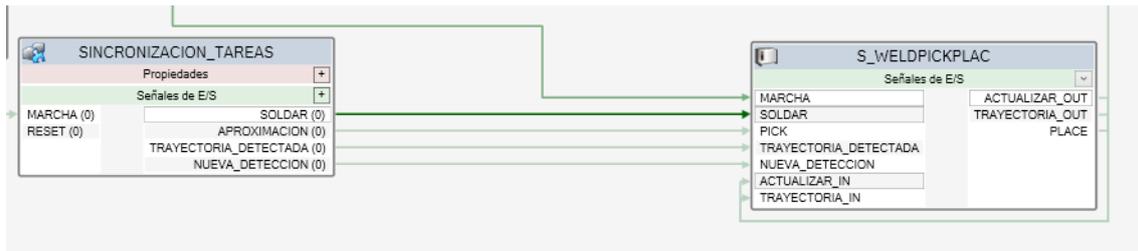


Figura 4.63 Conexión señal SOLDAR

Ya hemos visto en el punto anterior el funcionamiento del componente inteligente y sus entradas y salidas, a continuación, deberemos crear señales de entrada en el sistema, para ello en la pestaña Rapid entraremos en el árbol del controlador y dentro de configuración encontraremos "I/O System" y en él, dentro del menú "signal", el editor de señales.

Puesto que recibiremos una señal analógica en forma de pulsos, configuraremos esta entrada como una entrada Digital de nombre SOLDAR, será la responsable de indicar el preciso momento de realizar la soldadura en cada ciclo.

También necesitaremos una señal digital de entrada denominada "MARCHA" que dominara el funcionamiento del sistema.

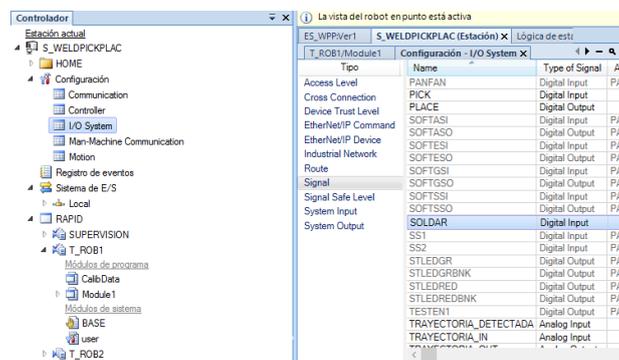


Figura 4.64 Creación de entradas y salidas

Tras la creación de señales deberemos reiniciar el controlador para poder utilizarlas, una vez reiniciado podremos operar con estas entradas tanto desde las instrucciones del editor de trayectorias, como a través de lenguaje RAPID.

4.4.4.1.4 Programación Rapid

Tras la sincronización con Rapid se nos ha creado el archivo de calibración "CalibData" y el archivo principal "Module1", está vacío en cuanto a instrucciones, pero contiene todos los objetivos y trayectorias creadas en la estación.

Procedemos a editarlo y adaptarlo a nuestras necesidades.

- Creamos una variable para la velocidad de soldadura.

```
VAR speeddata Vel_WELD:=[215.53,500,5000,1000];
```

- En la trayectoria creamos una espera para la señal SOLDAR, el robot permanecerá en el punto de inicio hasta que se active dicha señal ordenando la ejecución de la soldadura.

```
MoveL Target_INICIO_WELD,v1000,fine,tWeldGun\WObj:=WO_WELD;  
WaitDI SOLDAR,1;      !ESPERA LA ORDEN DE SOLDAR  
                       !ENCENDER ANTORCHA  
MoveC Target_WELD,Target_FIN_WELD,Vel_WELD,fine,
```

- Se agrupan y corrigen los movimientos del tramo de soldadura para que describan una trayectoria circular entre los objetivos (cambiar L por C), y se asigna la velocidad de soldadura "Vel_WELD".

```
MoveC Target_WELD,Target_FIN_WELD,Vel_WELD,fine,
```

Por último, programamos el bloque main de la tarea deseada.

En este caso pretendemos que mientras el robot este en marcha, espere en una posición específica, una señal de activación para la soldadura, al recibirla realiza la operación, volviendo al punto de inicio a la espera de una nueva señal. También se desea que

cuando se interrumpa la marcha (ya sea por parada o la apertura de alguna puerta), el robot ejecute una trayectoria a una posición “HOME1” o de parada que facilite labores de mantenimiento.

Esto resulta en un bloque sencillo main compuesto por un condicional “WHILE” aunque se podría programar de distintas maneras.

```

PROC main()
WHILE MARCHA=1 DO !FUNCIONAMIENTO MIENTRAS ESTE LA MARCHA PUESTA
(INCLUIDA SEGURIDAD DE LAS PUERTAS)
    Path_WELD;
ENDWHILE
    Path_PARADA;
ENDPROC

```

Tras programar una trayectoria adicional, se ha creado una señal llamada “LIMPIEZA” y ha sido vinculada con una rutina de interrupción TRAP “LIMPIEZA_ANTORCHA”, para ejecutar las tareas de limpieza de la antorcha mediante un pulsador.

```

IDelete intLIMPIEZA;
CONNECT intLIMPIEZA WITH LIMPIEZA_ANTORCHA;
ISignalDI LIMPIEZA,1,intLIMPIEZA;

```

- Programa RAPID resultante (Sin targets):

```

MODULE Module1
VAR speeddata Vel_WELD:=[215.53,500,5000,1000];
VAR intnum intLIMPIEZA;
|*****
! Módulo: Module1
!
! Descripción: Modulo responsable de las tareas de soldadura y limpieza
!
! Autor: J . Albarranch
|*****
PROC main()
    IDelete intLIMPIEZA;
    CONNECT intLIMPIEZA WITH LIMPIEZA_ANTORCHA;
    ISignalDI LIMPIEZA,1,intLIMPIEZA;

```

```
    WHILE MARCHA=1 DO !FUNCIONAMIENTO MIENTRAS ESTE LA MARCHA PUESTA
(INCLUIDA SEGURIDAD DE LAS PUERTAS)
```

```
    Path_WELD;
```

```
ENDWHILE
```

```
    Path_PARADA;
```

```
ENDPROC
```

```
PROC Path_WELD()
```

```
    MoveL Target_ENTRADA,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
    MoveL Target_INICIO_WELD,v1000,fine,tWeldGun\WObj:=WO_WELD;
```

```
    WaitDI SOLDAR,1; !ESPERA LA ORDEN DE SOLDAR
```

```
    !ENCENDER ANTORCHA
```

```
MoveC Target_WELD,Target_FIN_WELD,Vel_WELD,fine, tWeldGun\WObj:=WO_WELD;
```

```
    !APAGAR ANTORCHA
```

```
    MoveL Target_SALIDA,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
    MoveL Target_ENTRADA,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
ENDPROC
```

```
PROC Path_PARADA()
```

```
    MoveL HOME1,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
ENDPROC
```

```
TRAP LIMPIEZA_ANTORCHA
```

```
    Path_LIMPIEZA;
```

```
    MoveL Target_INICIO_WELD,v1000,fine,tWeldGun\WObj:=WO_WELD;
```

```
ENDTRAP
```

```
PROC Path_LIMPIEZA()
```

```
    MoveL HOME1,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
    MoveL Target_L1,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
```

```
    MoveL Target_L2,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
```

```
    MoveL Target_L3,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
```

```
    WaitTime 3;
```

```
    MoveL Target_L1,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
```

```
    MoveL Target_L4,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
```

```
    MoveL Target_L5,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
```

```
    MoveL Target_L6,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
```

```
    WaitTime 1;
```

```
    MoveL Target_L5,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
```

```
    MoveL HOME1,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
ENDPROC
```

```
ENDMODULE
```

4.4.4.2 Subestación pick and place

Esta subestación recibe las piezas terminadas en la subestación de soldadura, a la misma velocidad y la misma cadencia aleatoria y debe catalogarlas y redirigirlas a sus cintas transportadoras finales.

En función de las señales enviadas por un detector planar, la subestación deberá identificar el tipo de pieza e incluirla en una cola FIFO, además, deberá predecir el momento en el que debe comenzar las operaciones de pick and place para alcanzar la pieza en movimiento.

4.4.4.2.1 Objetivos y Trayectorias

A continuación, se pretenden crear tres trayectorias que faciliten el transporte de piezas terminadas entre la cinta de suministros y las cintas de material terminado

- **Objeto de trabajo**

Comenzamos creando un objeto de trabajo que servirá de referencia para los objetivos y trayectorias creadas a continuación, De nuevo, nos dirigimos a la a el menú de “Programación de trayectorias” y en “otros”, hacemos click en “Creación de objeto de trabajo” , en este caso se llamó “WO_PICKPLACE” y se decidió conectarlo en una pata de la primera cinta de transporte de material terminado, parece un punto correcto por su proximidad a las dos superficies de importancia de cara a futuros ajustes o modificaciones.

Para su correcta colocación se despliega el submenú “Posición” en el editor de objetos de trabajo y con la opción de ajuste final activado se pincha en la esquina de interés.

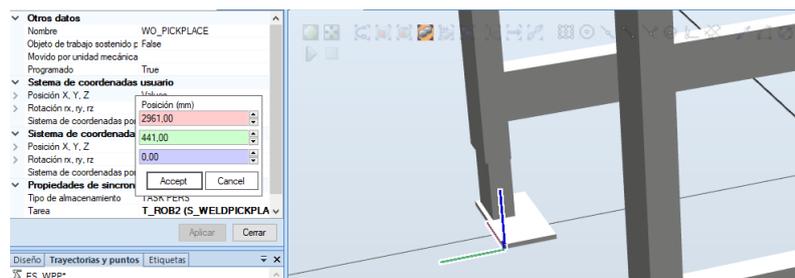


Figura 4.65 WO_PICKPLACE

- **Objetivos**

Se deberán crear todos los objetivos necesarios para cubrir las tres trayectorias deseadas.

Comenzamos creando una parada de la simulación un tiempo conocido después de que la pieza corte el detector para estimar el punto en el que deseamos coger la pieza, utilizando esta de referencia y puesto que la herramienta tiene ubicados los TCP en los extremos finales dentro del plano medio, con la opción “Ajuste a línea media” activado, seleccionaremos nuestro primer punto, pulsando directamente en “programar punto”, el punto se crea con su correcta orientación al basarse en la posición de TC de la herramienta.

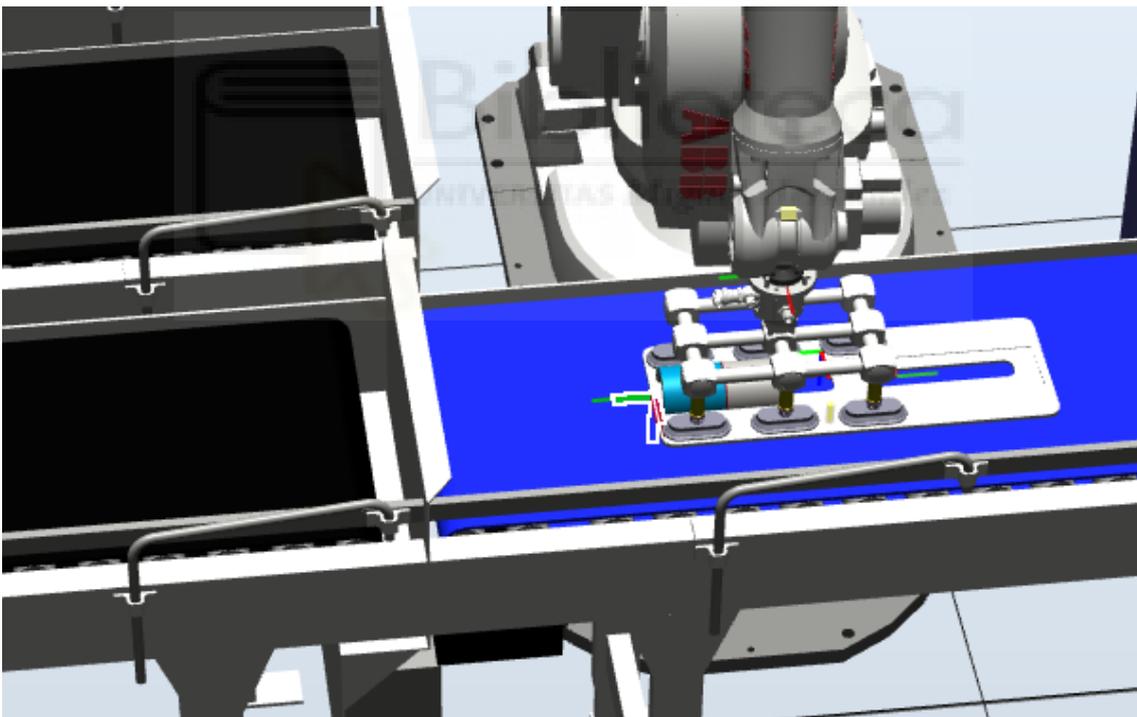


Figura 4.66 Robot en Target_0_2 (PICK)

Se ha diseñado para una aproximación de un segundo desde 50 mm de altura, para esto se ha copiado y pegado el primer objetivo modificando su posición -200 mm en Y y 50 mm en Z (según el sistema de referencia TCP).

Estos dos puntos serán comunes en todas las trayectorias por lo que los renombramos como “Target_0_1” y “Target_0_2” respectivamente.

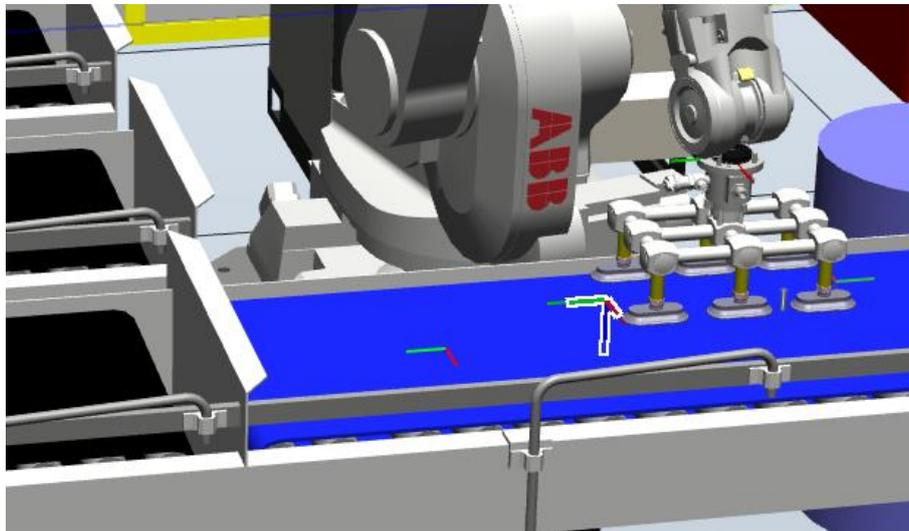


Figura 4.67 Robot en Target_0_1

Sin soltar la pieza, y con la opción “mover” lineal de las opciones “Mano alzada” desplazamos la pieza desde el Target_0_2 hasta alcanzar una posición adecuada y programamos el punto.

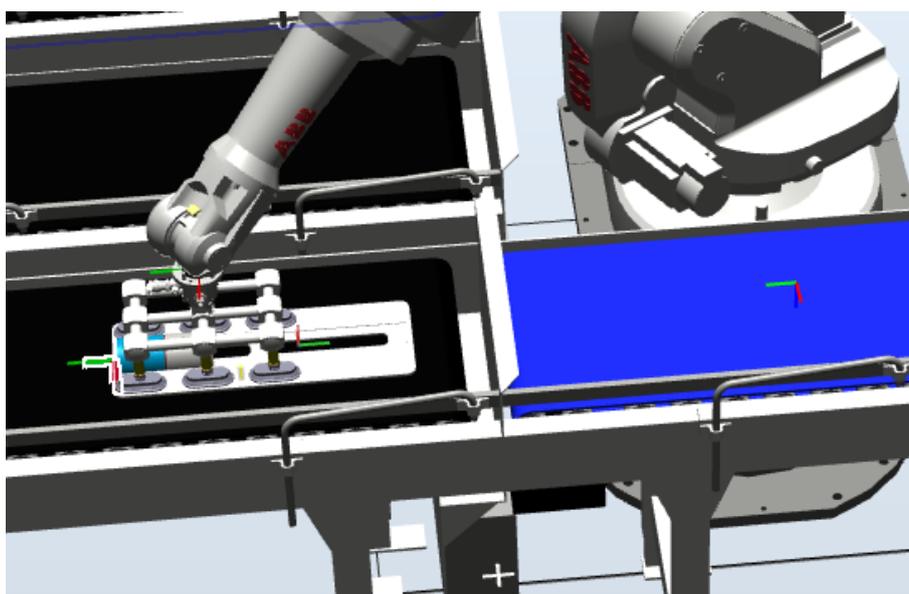


Figura 4.68 Robot en Target_1_2 (PLACE)

Repetimos la operación moviendo la herramienta en horizontal y vertical generando el resto de puntos intermedios. Además, son modificados mediante “fijar posición” para darle unas coordenadas mas intuitivas y sin decimales.

Todos los puntos se encuentran en el plano medio de la cinta.

Estos puntos los renombramos como “Target_1_X” en alusión a pertenecer a la primera trayectoria.

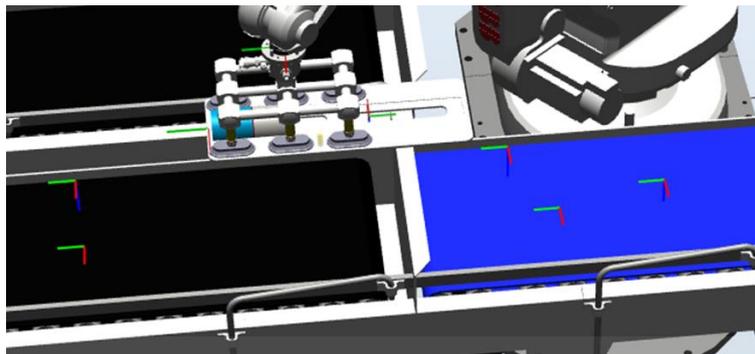


Figura 4.69 Objetivos Trayectoria 1

Copiamos los puntos “Target_1_X”, modificamos su posición ubicandola en el plano medio de la segunda cinta y los renombramos como “Target_2_X”, destinados a la trayectoria dos.

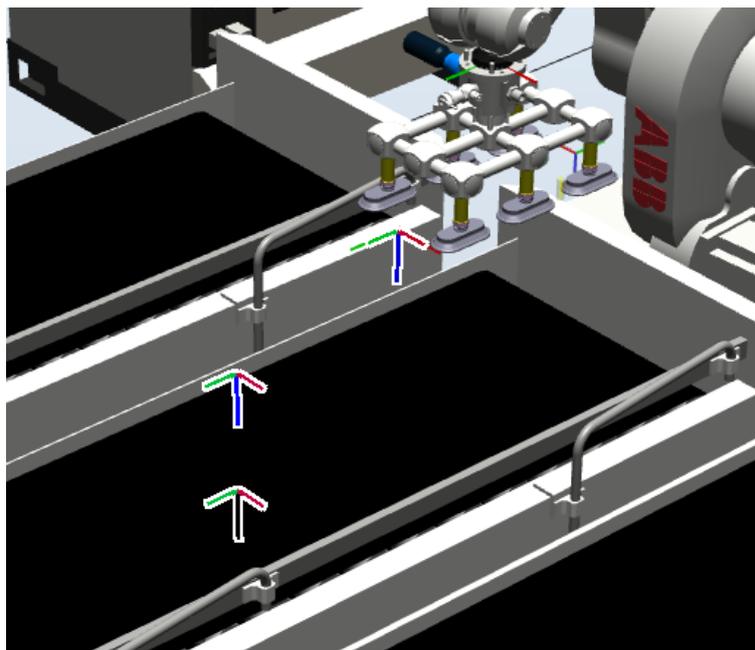


Figura 4.70 Objetivos Trayectoria 2

Repetimos el proceso anterior, pero ubicándolos en la tercera cinta, añadiendo además un punto intermedio para el retorno de la herramienta.

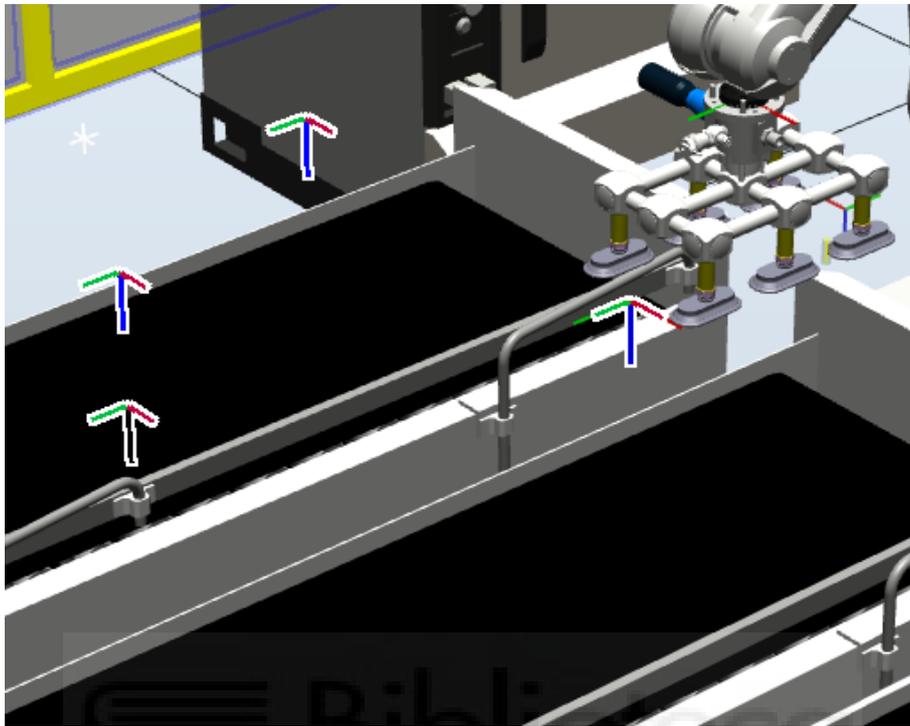


Figura 4.71 Objetivos Trayectoria 3

Obteniendo de esta manera la siguiente lista de targets:

- ▲ T_ROB2
 - ▾ Datos de herramienta
 - ▲ &Objetos de trabajo y puntos
 - ▲ WO_PICKPLACE
 - ▲ WO_PICKPLACE_de
 - Target_0_1
 - Target_0_2
 - Target_0_3
 - Target_1_1
 - Target_1_2
 - Target_1_3
 - Target_2_1
 - Target_2_2
 - Target_2_3
 - Target_3_1
 - Target_3_2
 - Target_3_3
 - Target_3_4

Figura 4.72 Objetivos creados Tarea T_ROB2

- **Creación de trayectorias**

Como se ha visto en puntos anteriores, para crear un trayectoria nos dirigimos a “Ruta” en la cinta de opciones de “Posición inicial” y creamos una ruta vacía, en esta ocasión creamos tres que se corresponderán con las tres cintas destino de material finalizado.

Las rutas estarán compuestas por tres tramos cada una; Un primer tramo responsable de aproximarse a la pieza objetivo (Pick), este tramo será común en las tres trayectorias, un segundo tramo de alzamiento de desplazamiento y posicionamiento de la pieza en su destino (Place), y, finalmente un tercer tramo que devolverá la herramienta a su punto de espera.

Arrastramos los destinos creados en el punto anterior a su trayectoria y ordenamos las secuencias.

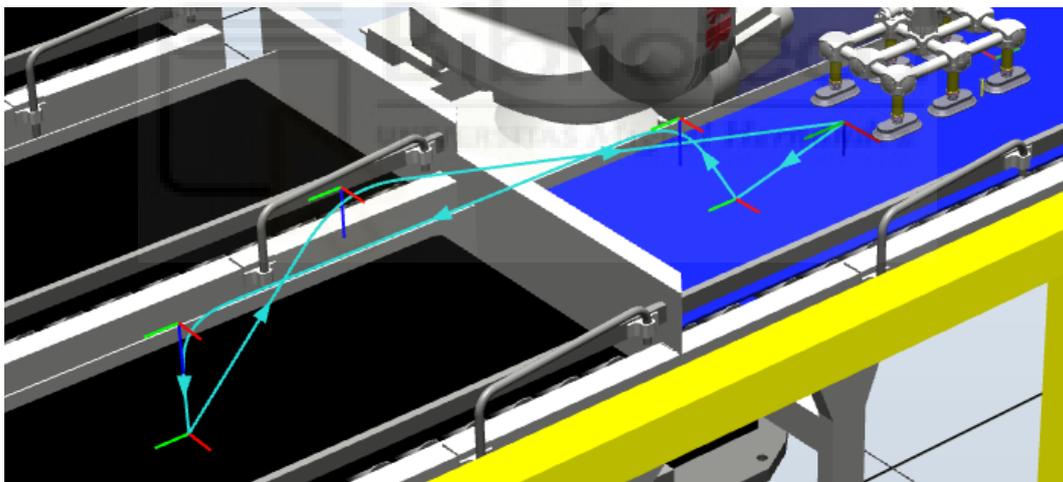


Figura 4.73 Trayectoria 1

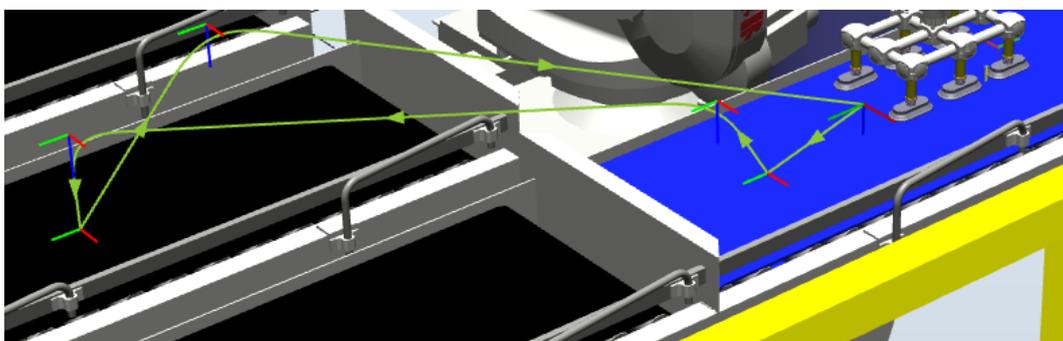


Figura 4.74 Trayectoria 2

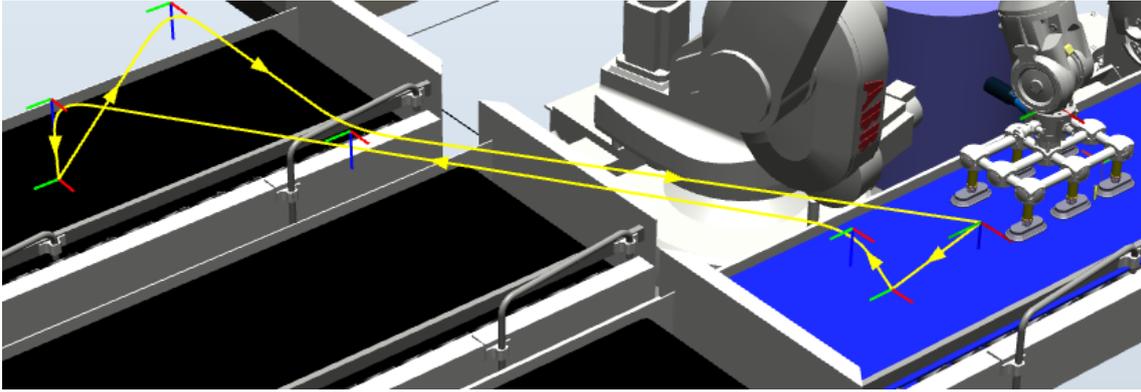


Figura 4.75 Trayectoria 3

Quedando estructuradas las trayectorias como muestra la siguiente figura:

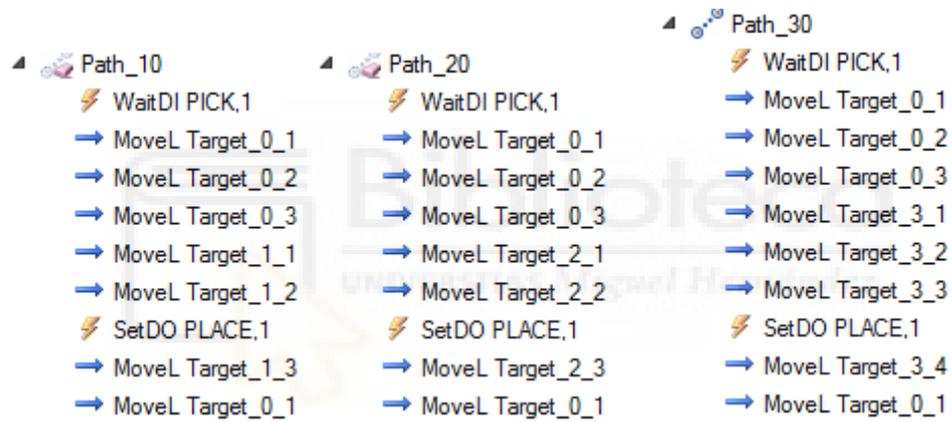


Figura 4.76 Instrucciones de movimiento

Solo se han modificado las instrucciones responsables de las operaciones de pick and place cambiando su atributo “zone” a “fine” para mejor precisión en estos puntos.

Las instrucciones de acción se explicarán más adelante en los apartados destinados a la lógica y Rapid.

4.4.4.2.2 Smart Components (SC)

- **SINCRONIZACION_TAREAS/SC PICK_AND_PLACE**

Agrupación de SC relativos a las tareas de pick and place en el contexto de sincronización de tareas.

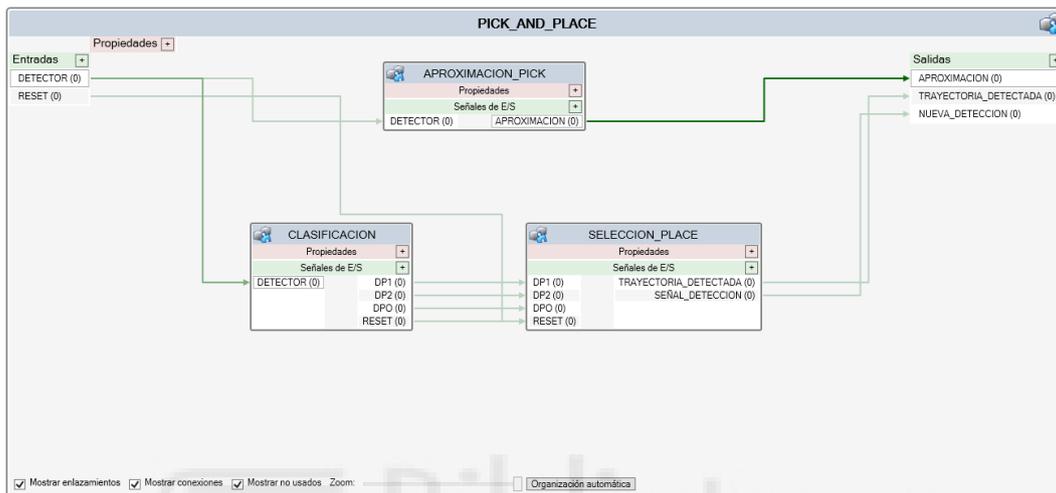


Figura 4.77 SC PICK AND PLACE

- **SC APROXIMACION:**

Es un componente inteligente sencillo dedicado a calcular cuando debe comenzar la tarea de Pick-and-place, para ello hace uso, como todos los SC incluidos en la sincronización de tareas como entrada, la señal recibida del detector. Y emite una señal de salida digital denominada "APROXIMACION" cuando debe comenzar el proceso; Para ello, hace uso de dos componentes inteligentes:

- LogicSplit:

Recibiendo como entrada la señal que viene del detector planar, y esta vez utilizando la salida "PulseHight" que enviara un pulso "1" cuando la señal cambie de "0" a "1", indicando que la pieza ha comenzado a ser detectada.

- LogicGate[OR]:

Puerta lógica con la única finalidad de introducir un retraso a la señal de salida mediante su atributo “Delay” que para la configuración utilizada en este sistema es establecida en 9s.

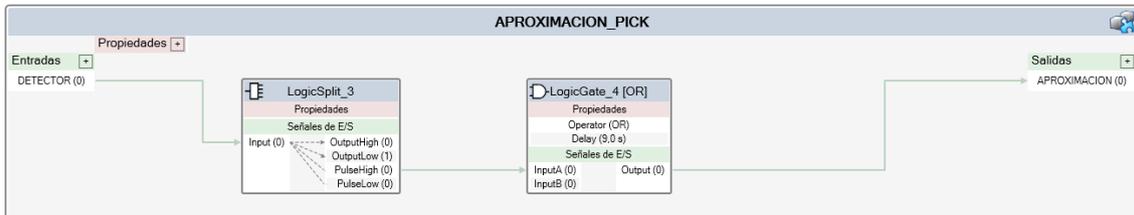


Figura 4.78 SC APROXIMACION_PICK

- **SC CLASIFICACION**

Responsable de la clasificación de las piezas que entran en el sistema, recibe como entrada la señal del detector, y en base al resultado obtenido en el cronometro “StopWatch”, los componentes inteligentes “SC PIEZA_TIPO_1,2 o Otros” discrimina la pieza de la que se trata y activa la señal pertinente.

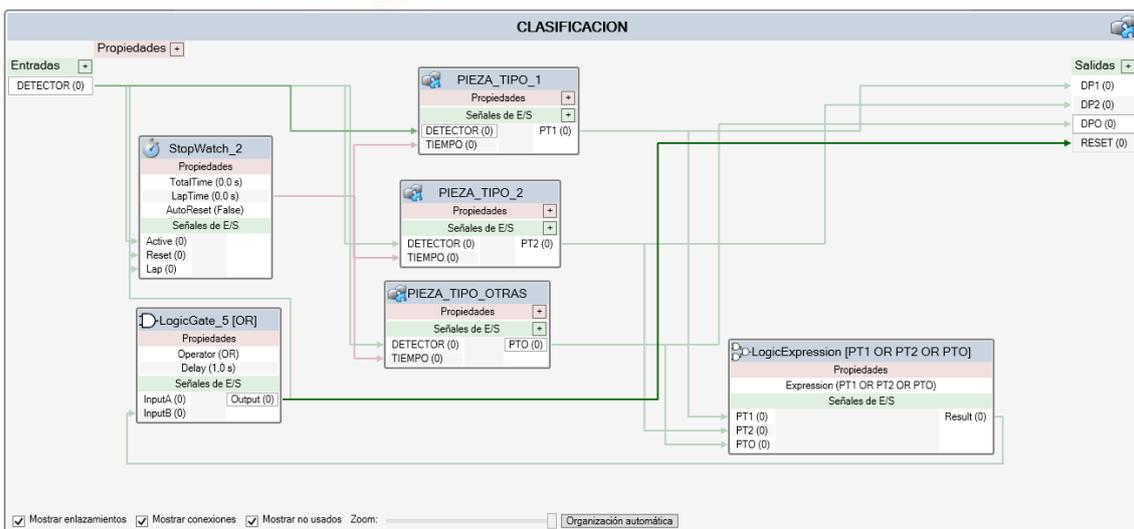


Figura 4.79 SC CLASIFICACION

- **SC PIEZA_TIPO_1**

Componente dedicado a detectar piezas de tipo 1; Recibe como entradas, por un lado, la señal digital que indica que hay una pieza cruzando el detector, y por otro, una señal analógica con el tiempo que la pieza está siendo detectada. Como salida, activa la señal digital PT1 en caso de verificar que se trata de una pieza tipo 1. (Figura 4.80)

Esta diseñado con cuatro complementos:

- LogicSplit:

Recibe la señal del detector y tomamos como salida el pulso "PulseLow" indicativo de que la pieza ha atravesado el detector por completo.

- Comparer:

Encontramos dos complementos de este tipo, envían una señal cuando se cumple una condición matemática, para esta pieza se ha establecido que la pieza debe cortar el detector más de 0.744 segundo y menos de 0.768, estos valores han sido fijados el atributo "ValueB" mientras que el valor de entrada dinámico es recibido en "ValueA", matemáticamente, se podría haber utilizado un solo comparador, pero empíricamente se ha comprobado un mejor funcionamiento dejando un pequeño margen de error de 12ms.

- LogicExpresion:

Expresión lógica configurada como un triple "AND" recibe las señales de los dos comparadores y el LogicSplic y envía una señal "1" de verificarse los tres, es decir, cumple las dos condiciones de verificación y la pieza ha terminado de ser detectada.

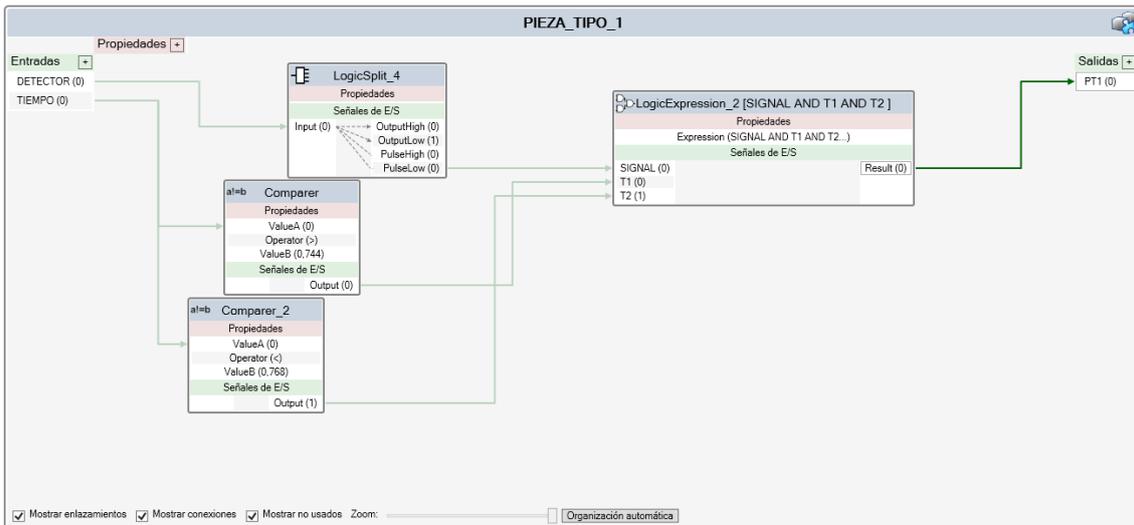


Figura 4.80 SC PIEZA_TIPO_1

- SC PIEZA_TIPO_2

Componente exactamente igual que el anterior con la única diferencia de los parámetros configurados en los comparadores, en este caso $1.488 < A < 1.512$, para definir las piezas de tipo 2.

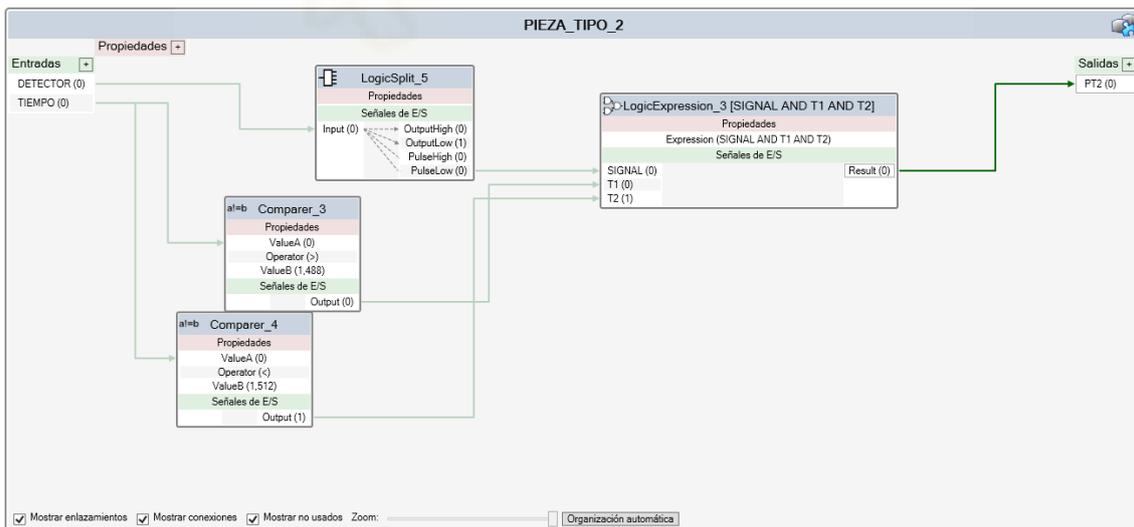


Figura 4.81 SC PIEZA_TIPO_2

- **SC PIEZA_TIPO_OTRAS**

Debido a la flexibilidad de este proyecto, capaz de asumir piezas de distintos tamaños, este bloque no identifica una pieza en concreto, simplemente detecta que no se trata de piezas especificadas de forma concreta, para ello, utiliza una estructura similar a los bloques anteriores combinados con una negación, se podría resumir como “Si ha activado y desactivado el detector, y no se trata de una pieza tipo 1 o tipo 2, es “otra pieza””.

Hace uso de las mismas entradas y salidas que los componentes anteriores, además usa los mismos comparadores y puertas lógicas anteriores añadiendo en esta ocasión:

- LogicExpresion:

En este caso configurado como un doble “NOT” que se activara si no se activan la señal de pieza 1 ni la de pieza 2 resultado de los dobles comparadores con sus respectivas puertas lógicas.

- LogicGate:

Configurado como un [AND], activara la señal “PTO” (piezas tipo otras) en caso de recibir la señal de pieza detectada y la negación de que se trate de otro tipo de pieza como se anticipaba en la descripción del complemento.

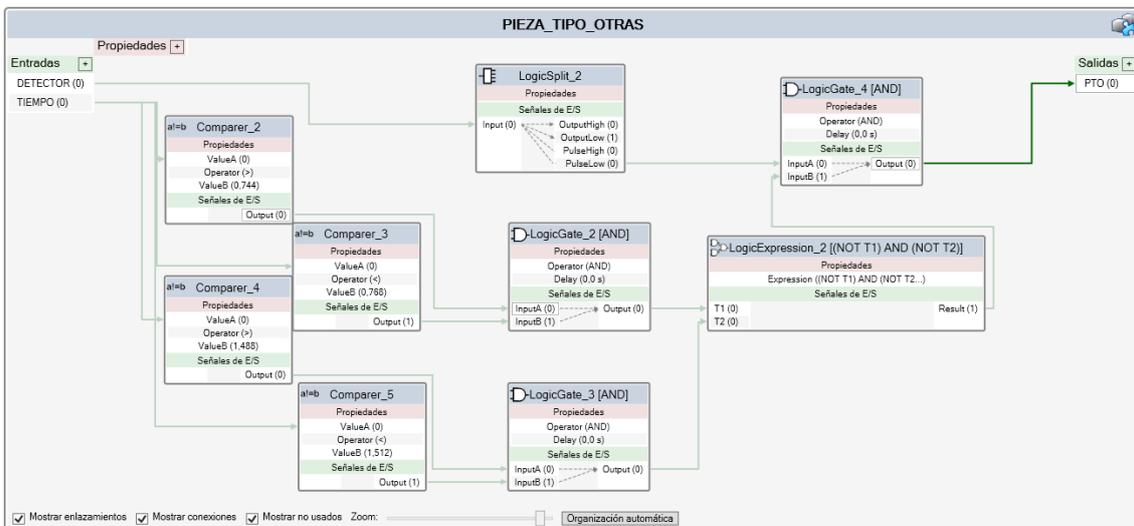


Figura 4.82 SC PIEZA_TIPO_OTRAS

- **SC SELECCIÓN_PLACE**

Complemento dedicado a decidir el destino de cada pieza en función del tipo detectado. En este caso se toman como entradas tres señales digitales DP1,DP2,DPO correspondientes a las detecciones en los componentes anteriores, y una señal RESET para reiniciar los valores tras cada lectura; En cuanto a las salidas se configura una salida digital "SEÑAL_DETECCION" que indica que la pieza ha sido correctamente catalogada y una señal analógica "TRAYECTORIA_DETECTADA" que será 1, 2 o 3 en función de la pieza tratada, estas señales serán de vital importancia de cara a su implementación en RAPID.

Para esta labor se han utilizado ocho complementos (Figura 4.83):

- LogicSRLatch:

Encontramos tres complementos de este tipo, uno por cada tipo de pieza, interruptores que se activaran en función de la pieza detectada.

- Expression:

Encontramos nuevamente tres componentes asociados cada uno a los LogicSRLatch anteriores, su función es convertir las señales analógicas en digitales mediante una operación matemática, siendo $DP1*1$ en caso de piezas tipo 1, $DP2*2$ en caso de piezas tipo 2 y $DP3*3$ en caso de piezas tipo otras, siendo las responsables de alimentar la salida analógica "TRAYECTORIA_DETECTADA".

- Comparer:

Bloque comparador que recibe como entrada en su atributo "ValueA" el resultado de las trayectorias anteriores y activándose en el caso de ser mayores que 0, parámetro establecido en "ValueB".

- LogicSplit:

Recibe la señal generada en el comparer anterior y la convierte en un pulso mediante su salida "PulseHight".

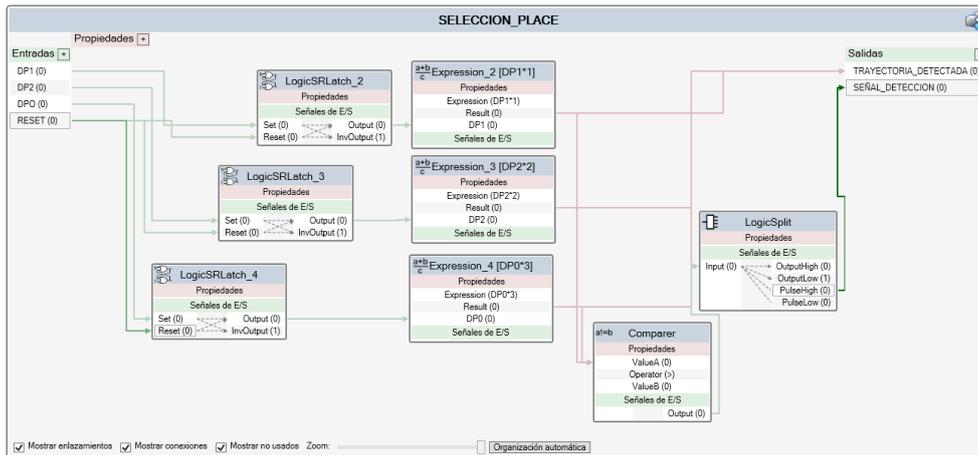


Figura 4.83 SC SELECCION_PLACE

- **SC VENTOSA**

Componentes responsables de las operaciones de Pick-and-place, retirando las piezas de una cola para su manipulación y reinsertándolas en otra cola para continuar su camino. Se compone de la herramienta de ventosas seleccionada como fin de brazo y los siguientes componentes:

- LineSensor:

Tras la vinculación con la herramienta, detectara cuando una pieza corta su haz enviando una señal "SensorOut" e indicara la pieza detectada con su atributo (SensePart()).

- Attacher:

Conecta físicamente el objeto indicado en su entrada "Child" al SmartComponent, lo que nos permite ligarlo al movimiento del robot de forma solidario y desplazarlo con las trayectorias del robot, debe estar ligado en su atributo "Parent" con un complemento "Detacher" responsable de las funciones opuestas, es decir, soltar el objeto cuando sea preciso. Recibirá la señal para conectarse cuando el detector lineal le envíe una señal a través de su entrada "Execute" y enviará una señal tras conectarse correctamente a través de su salida "Executed"

- Detacher:

Como se mencionaba previamente, desconectará los objetos del attacher, para ello recibirá una señal de desconexión en su entrada "Execute" y el objeto a desconectar en su atributo "Child" vinculado al "Child" del Attacher como observamos en la figura del esquema.

- Queue:

Representa la cola de objetos finalizados, tras recibir la confirmación de desconexión del "Detacher" previo en su entrada Execute, añade el objeto indicado por el Detacher en su entrada "Back" añadiéndolo así a la nueva cola.

- LinearMover:

Representa las cintas transportadoras de piezas finalizadas, estas cintas se encuentran activas siempre que la estación este en marcha, desplazando en este caso el objeto indicado en "Object" en este caso las colas de objetos finalizados a una velocidad y con una orientación indicados, está configurado con una velocidad de 200 mm/s en la dirección positiva de "X".

En cuanto a entradas y salidas, podemos observar que está compuesto por dos entradas y una salida digitales en todos los casos.

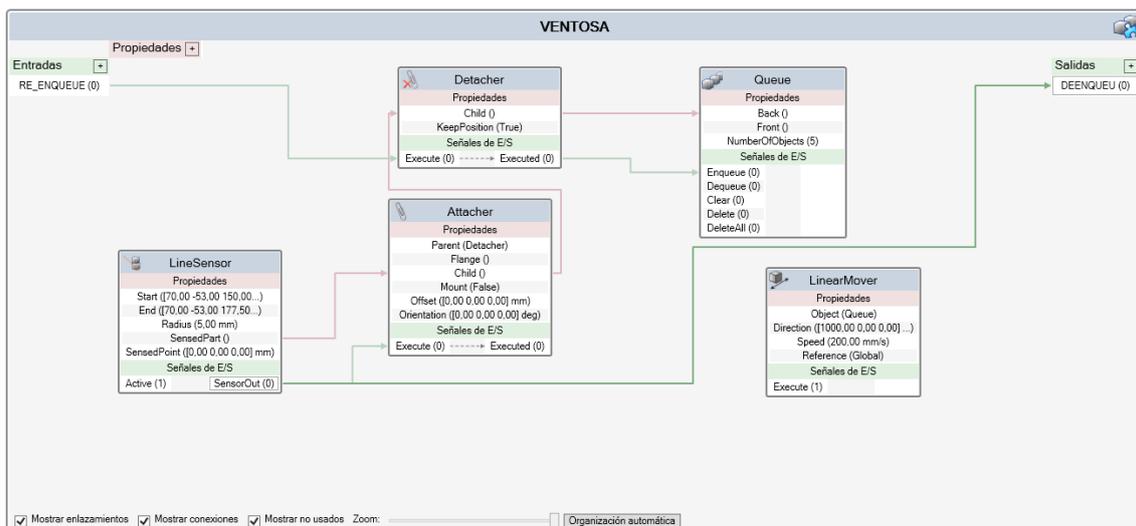


Figura 4.84 SC VENTOSA

En las entradas encontramos “MARCHA” responsable de activar el LinearMover y “RE_ENQUEUE”, que recibirá una señal cuando el robot dedicado al pick-and-place se encuentre en la posición deseada para desconectar el objeto reincorporándolo a la cola mediante el Detacher; Por su parte como salida encontramos “DEENQUEUE” para enviar una señal cuando se coja un objeto y se desee retirar de la primera cola incluida en el SC CINTA_SUMINISTROS.

4.4.4.2.3 Lógica, Entradas y Salidas

La subestación Pick and Place funciona gracias a la interconexión de los componentes inteligentes vistos en el punto anterior englobados en la sincronización de tareas, la ventosa y el sistema de nuestro robot.

Para ello deberemos crear una serie de señales en el controlador de nuestro robot.

Comenzamos creando las entradas para la conexión con el bloque “SINCRONIZACION _TAREAS”; La entrada MARCHA la tenemos disponible de la subestación anterior.

A continuación, creamos dos entradas digitales, “PICK” que recibirá la señal APROXIMACION cuando el robot deba iniciar el seguimiento y NUEVA_DETECCION que recibirá una señal cuando esté disponible una nueva trayectoria para añadir a la cola FIFO.

También deberemos crear una entrada analógica “TRAYECTORIA_DETECTADA” para poder gestionar las distintas trayectorias catalogadas, estas serán las introducidas en la cola FIFO.

Con carácter interno del sistema se crean dos parejas de entradas-salidas, con la intención de vincular dos tareas paralelas que controlan el sistema.

La salida “TRAYECTORIA_OUT” enviara una señal analógica a la entrada “TRAYECTORIA_IN” con la instrucción de la siguiente trayectoria a realizar.

La salida “ACTUALIZAR_OUT” enviara una señal digital a la entrada “ACTUALIZAR_IN” cuando se haya cargado una trayectoria en el sistema y se deba actualizar el vector de la cola.

Finalmente, el sistema dispone de una salida digital “PLACE” que se encargara de desactivar el attach de la pinza y reintroducir las piezas finalizadas en sus cintas de transporte.

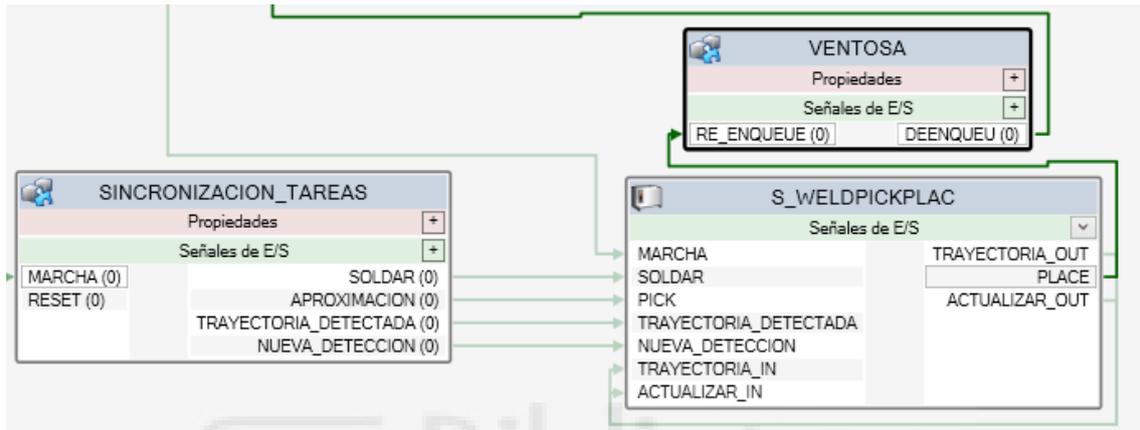


Figura 4.85 Conexiones subestación Pick and Place

4.4.4.2.4 Programación Rapid

La programación Rapid de la subestación de pick and place está estructurada en dos tareas dependientes entre sí.

“T_ROB2” será la tarea principal y controlará los movimientos del robot IRB 1600.

La tarea “SUPERVISION” permanecerá en un bucle constante esperando la detección de nuevas piezas, almacenándolas en una cola que se actualizará en combinación con las acciones realizadas en la tarea T_ROB2.

- Tarea “SUPERVISION”

Para crear una tarea adicional desplegaremos el árbol del controlador en configuración accederemos al menú “Controller” donde encontraremos la pestaña de “Task”, con el botón derecho crearemos una nueva tarea.

Configuramos la tarea como tipo Normal, indicamos que NO es una tarea de movimiento, y la asociamos al grupo mecánico “rob2”.

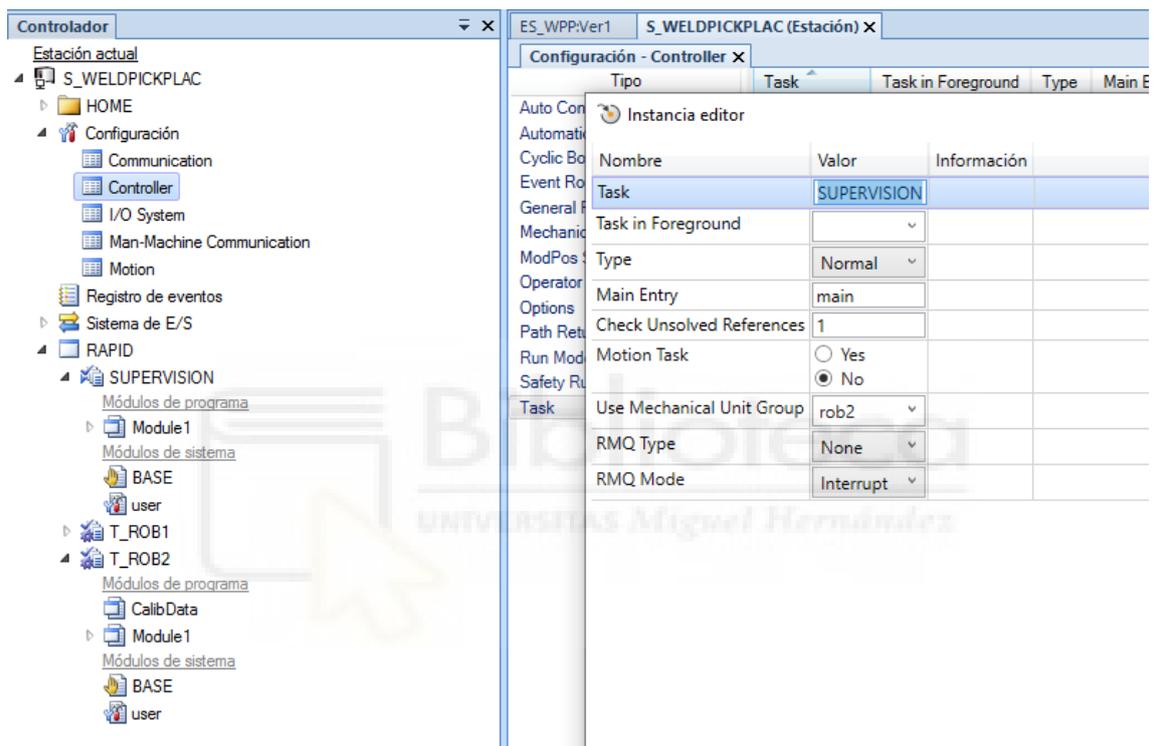


Figura 4.86 Creación de tarea SUPERVISION

A continuación, se explica el flujo de trabajo de esta tarea en paralelo a la tarea RAPID programada.

Comenzamos creando dos variables numéricas, la primera “contador” hace referencia a un contador simple, “VECTOR_TRAYECTORIAS” es un vector que acumulara y actualizara las trayectorias representando la cola FIFO.

Aprovechamos para establecer estas variables en “0” de manera que se inicialicen con cada simulación.

```

MODULE Module1

VAR num CONTADOR;           !Variable contador
VAR num VECTOR_TRAYECTORIAS{3}; !vector de trayectorias almacenadas

PROC main()

    CONTADOR:=0;           !INICIALIZA CONTADOR
    VECTOR_TRAYECTORIAS:=[0,0,0]; !INICIALIZA VECTOR DE RAYECTORIAS

ENDPROC
ENDMODULE

```

Se espera que este módulo funcione de forma continua siempre y cuando se encuentre en “MARCHA”, recordemos que la señal MARCHA llega filtrada por los sistemas de seguridad.

Para ello creamos un bloque “WHILE” que seguirá funcionando siempre que esté en marcha.

```

MODULE Module1

VAR num CONTADOR;           !Variable contador
VAR num VECTOR_TRAYECTORIAS{3}; !vector de trayectorias almacenadas

PROC main()

    CONTADOR:=0;           !INICIALIZA CONTADOR
    VECTOR_TRAYECTORIAS:=[0,0,0]; !INICIALIZA VECTOR DE TRAYECTORIAS

    WHILE MARCHA=1 DO

    ENDWHILE

ENDPROC
ENDMODULE

```

En el interior del bucle WHILE introduciremos las instrucciones a realizar.

El programa espera una activación en la señal digital del sistema “NUEVA_DETECCION” indicando que una pieza ha sido correctamente catalogada, como consecuencia directa se actualiza el contador y se carga en la posición del vector indicada en el contador el valor de la trayectoria obtenida en la entrada analógica “TRAYECTORIA_DETECTADA”;

En este paso se espera a que la señal de “NUEVA_DETECCION” se deshabilite antes de continuar, esto se hace para evitar que la lectura en bucle tome datos repetidos. Finalmente se establece la salida analógica “TRAYECTORIA_OUT” con el primer valor contenido en el vector de trayectorias almacenados, esto le indicara a la tarea Rob2 la trayectoria a seguir

```

MODULE Module1

VAR num CONTADOR;           !Variable contador
VAR num VECTOR_TRAYECTORIAS{3}; !vector de trayectorias almacenadas

PROC main()

    CONTADOR:=0;           !INICIALIZA CONTADOR
    VECTOR_TRAYECTORIAS:=[0,0,0]; !INICIALIZA VECTOR DE TRAYECTORIAS

    WHILE MARCHA=1 DO

        WaitDI NUEVA_DETECCION,1; !ESPERA A DETECTAR TRAYECTORIA

        CONTADOR:=CONTADOR+1; !ACTUALIZA CONTADOR
        VECTOR_TRAYECTORIAS{CONTADOR}:=TRAYECTORIA_DETECTADA;
!ACTUALIZA EL VECTOR
        WaitDI NUEVA_DETECCION,0;

        SetAO TRAYECTORIA_OUT,VECTOR_TRAYECTORIAS{1}; !TRAYECTORIA A
EJECUTAR

    ENDWHILE

ENDPROC
ENDMODULE

```

En este punto tenemos un bucle que toma datos, pero esperamos actualizar estos también, para ello introducimos una comprobación mediante un bucle “IF”. En cada ciclo se comprueba la activación de la señal “ACTUALIZAR_IN” por parte de la salida “ACTUALIZAR_OUT” que dispara la tarea T_Rob2 cuando ya ha cargado una trayectoria. Si la señal no se encuentra activa, el bucle continua y almacena nuevas trayectorias de existir estas, en el caso de recibir la señal de actualización, se actualiza el contador restando uno y se actualiza el vector de trayectorias desplazando sus elementos hacia la izquierda, y continua el bucle a la espera de recibir una nueva señal.

```

MODULE Module1

VAR num CONTADOR;           !Variable contador
VAR num VECTOR_TRAYECTORIAS{3}; !vector de trayectorias almacenadas

PROC main()

```

```

CONTADOR:=0;           !INICIALIZA CONTADOR
VECTOR_TRAYECTORIAS:=[0,0,0]; !INICIALIZA VECTOR DE TRAYECTORIAS

WHILE MARCHA=1 DO

    WaitDI NUEVA_DETECCION,1; !ESPERA A DETECTAR TRAYECTORIA

    CONTADOR:=CONTADOR+1; !ACTUALIZA CONTADOR
    VECTOR_TRAYECTORIAS{CONTADOR}:=TRAYECTORIA_DETECTADA;
!ACTUALIZA EL VECTOR
    WaitDI NUEVA_DETECCION,0;

    SetAO TRAYECTORIA_OUT,VECTOR_TRAYECTORIAS{1}; !TRAYECTORIA A
EJECUTAR

    IF ACTUALIZAR_IN=1 THEN !ACTUALIZACION DEL VECTOR
TRAS SU LECTURA
        CONTADOR:=CONTADOR-1;
        VECTOR_TRAYECTORIAS{1}:=VECTOR_TRAYECTORIAS{2};
        VECTOR_TRAYECTORIAS{2}:=VECTOR_TRAYECTORIAS{3};
        VECTOR_TRAYECTORIAS{3}:=0;
        SetDO ACTUALIZAR_OUT,0;
    ENDIF

ENDWHILE

ENDPROC
ENDMODULE

```

- Tarea T_ROB2

Tras sincronizar la estación con Rapid, ya tenemos cargados en esta tarea todos los objetivos y trayectorias diseñados en la estación.

Aprovechamos para declarar dos variables, una variable “speeddata” con la velocidad de aproximación calculada, hemos llamado a esta velocidad “Vel_PICK”. Y una variable numérica auxiliar para guardar la trayectoria recibida en la entrada analógica “TRAYECTORIA_IN” que llamamos “TRAYECTORIA_CARGADA”.

```

MODULE Module1
    CONST robtarget Target_0_1:=[[[-661,-241,835], [0,0.707106781,0.707106781, 0],[0,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_0_2:=[[[-461,-241,785],[0,0.707106781,0.707106781, 0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_0_3:=[[[-361,-241,950],[0,0.707106781,0.707106781, 0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_1_1:=[[[439,-241,950],[0,0.707106781,0.707106781, 0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

```

CONST robtarget Target_1_2:=[[439,-241,785],[0,0.707106781,0.707106781, 0],[-1,0,-
3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1_3:=[[189,-241,1050],[0,0.707106781,0.707106781, 0],[-1,0,-
3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_2_1:=[[439,-788.5,950],[0,0.707106781,0.707106781, 0],[-
1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_2_2:=[[439,-788.5,785],[0,0.707106781,0.707106781, 0],[-
1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_2_3:=[[189,-788.5,1050],[0,0.707106781, 0.707106781,0],[-
1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_3_1:=[[439,-1378.5,950],[0,0.707106781, 0.707106781,0],[-
1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_3_2:=[[439,-1378.5,785],[0,0.707106781, 0.707106781,0],[-
1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_3_3:=[[189,-1378.5,1050],[0,0.707106781, 0.707106781,0],[-
1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_3_4:=[[189,-851.168,1050],[0,0.707106781, 0.707106781,0],[-
2,0,-4,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR speeddata Vel_PICK:=[206.15,500,5000,1000];
VAR num TRAYECTORIA_CARGADA;

PROC main()

ENDPROC

PROC Path_10()
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_1_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_1_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_1_3,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
ENDPROC
PROC Path_20()
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_2_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_2_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_2_3,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
ENDPROC
PROC Path_30()
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_3_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_3_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_3_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_3_4,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
ENDPROC
ENDMODULE

```

Programaremos el cuerpo del programa dentro de un bucle “WHILE” nuevamente condicionado a la señal de “MARCHA” y programamos la secuencia de operaciones deseada en cada ciclo:

- 1- El robot se mueve a su posición de espera.
- 2- En condicional IF, comprueba en cada ciclo si hay una trayectoria en la entrada analógica “TRAYECTORIA_IN” vinculada a la salida “TRAYECTORIA_OUT” de la tarea supervisión.
- 3- De haber una trayectoria a la espera, esta es almacenada en la variable auxiliar “TRAYECTORIA_CARGADA”. Reestablece a “0” la señal analógica “TRAYECTORIA_OUT” y activa la señal de “ACTUALIZAR_OUT” que indicara a la tarea de supervisión que la presente trayectoria ha sido almacenada y puede actualizar el vector.
- 4- Finalmente, una función “TEST” Mandará la instrucción de ejecutar la trayectoria pertinente.

```
MODULE Module1
```

```
!CONSTANTES Y VARIABLES
```

```
PROC main()
```

```
  WHILE MARCHA=1 DO !EJECUCION MIENTRAS ESTA EN MARCHA (INCLUIDOS  
  SENSORES DE SEGURIDAD)
```

```
    !POSICION INICIAL
```

```
    MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
```

```
    !COMPROBACION DE TRAYECTORIA A LA ESPERA
```

```
    IF TRAYECTORIA_IN>0 THEN
```

```
      !CARGA DE TRAYECTORIA
```

```
      TRAYECTORIA_CARGADA:=TRAYECTORIA_IN;
```

```
      !REESTRABLECIMIENTO
```

```
      SetAO TRAYECTORIA_OUT,0;
```

```
      !ORDEN DE ACTUALIZAR VECTOR
```

```
      SetDO ACTUALIZAR_OUT,1;
```

```
    ENDIF
```

```
  TEST TRAYECTORIA_CARGADA
```

```
  CASE 1:
```

```

    Path_10;
    CASE 2:
    Path_20;
    CASE 3:
    Path_30;
ENDTEST

SetDO PLACE,0;  !REESTABLECER PLACE

    ENDWHILE
ENDPROC
!Path_10 Path_20 Path_30
ENDMODULE

```

Solo nos resta modificar las trayectorias incorporando las instrucciones necesarias para su completa ejecución.

Deberemos incorporar en cada Path una instrucción “WaitDI” para, una vez cargada la trayectoria, ejecutarla en el momento adecuado para una aproximación sincronizada a la pieza.

La segunda instrucción que debemos incorporar a todas las trayectorias es la activación de la señal “PLACE” que envíe una señal para que el efector final de ventosas suelte la pieza y se reincorpore a la cola de piezas terminadas. Esta señal será reestablecida a cero tras finalizar la trayectoria actual.

```

    PROC Path_10()
    WaitDI PICK,1;  !ESPERAR SEÑAL PARA EL PICK
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_1_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_1_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    SetDO PLACE,1;  !SOLTAR PIEZA
MoveL Target_1_3,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    ENDPROC
    PROC Path_20()
    WaitDI PICK,1;  !ESPERAR SEÑAL PARA EL PICK
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_2_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_2_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    SetDO PLACE,1;  !SOLTAR PIEZA
MoveL Target_2_3,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    ENDPROC
    PROC Path_30()
    WaitDI PICK,1;  !ESPERAR SEÑAL PARA EL PICK
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;

```

```

MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_3_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_3_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_3_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    SetDO PLACE,1;    !SOLTAR PIEZA
MoveL Target_3_4,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    ENDPROC
ENDMODULE

```

El código completo se puede encontrar el Anexo Rapid.

4.4.4.3 Nivel superior

En el nivel superior, la estación completa se encuentra formada por cinco componentes inteligentes diferenciados y dedicados a tareas concretas, vinculados entre sí para su correcto funcionamiento.

Además, han sido utilizadas como medio de simplificación y organización, agrupando todos los SC vistos hasta ahora, ya que tener todos los Smart Components de más bajo nivel en pantalla la saturaría y daría lugar complejos laberintos de señales de entrada y salidas que dificultaría su comprensión y modificación.

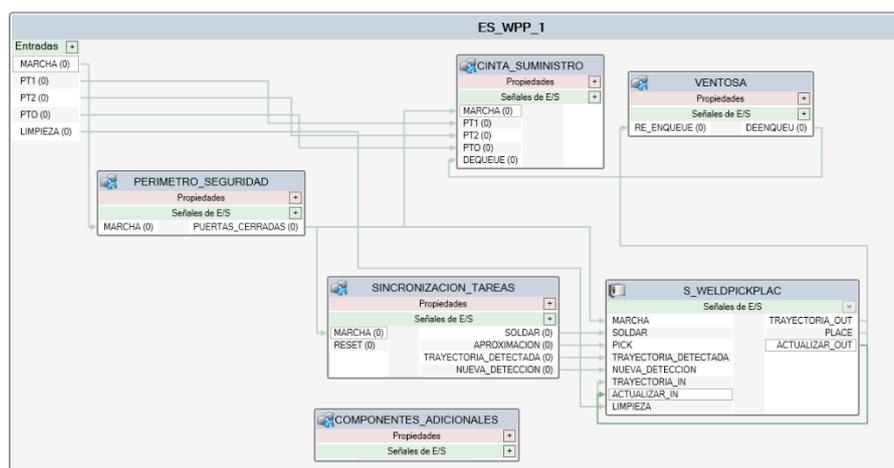


Figura 4.87 Estación WPP

4.4.4.3.1 Diseño de Smart Components de nivel superior

- **SC CINTA_SUMINISTRO**

El primer SC asignado, llamado "CINTA_SUMINISTRO" será el responsable de crear las piezas con las que operar e introducirlas o sacarlas de la cola de procesos, para ello, ha sido configurada con los siguientes componentes:

- Source:

Indica el objeto que debe clonarse, la propiedad Source indica la pieza que debe clonarse al recibir una señal en la entrada "Execute", los atributos "Position" y "Orientation" señalan la posición y orientación en la que debe realizarse la copia y tras la ejecución envían la señal "Executed" indicando que la clonación se ha realizado con éxito. En este SC existen tres componentes "Source" referenciando a las tres piezas modelo con las que deseamos trabajar.

- Queue:

El componente Queue representa una cola, en función de su configuración puede representar una cola FIFO (First In First Out) o una cola LIFO (Last In First Out), en este caso se ha configurado como una cola FIFO al asignar las copias creadas en los distintos Source a la entrada "Back" (En "Front" podremos ubicar la pieza situada en cabeza); Se podrían eliminar los objetos de la cola mediante las órdenes "Clear" o "ClearAll" pero no es necesario en este caso, si lo es por otro lado, conectar las señales "Executed" de las distintas fuentes a la entrada "Enqueue" para añadir las copias a la cola y tener presente la entrada "Dequeue" para retirarlas cuando sea preciso.

- LinearMover:

Responsable de mover los objetos asignados en su atributo "Object" que en este caso será la cola de objetos previamente creada, en la dirección y con la velocidad asignada, en este caso en la dirección "X" y con una velocidad de 200mm/s. Este movimiento se realiza siempre que este recibiendo una señal en su entrada "Execute".

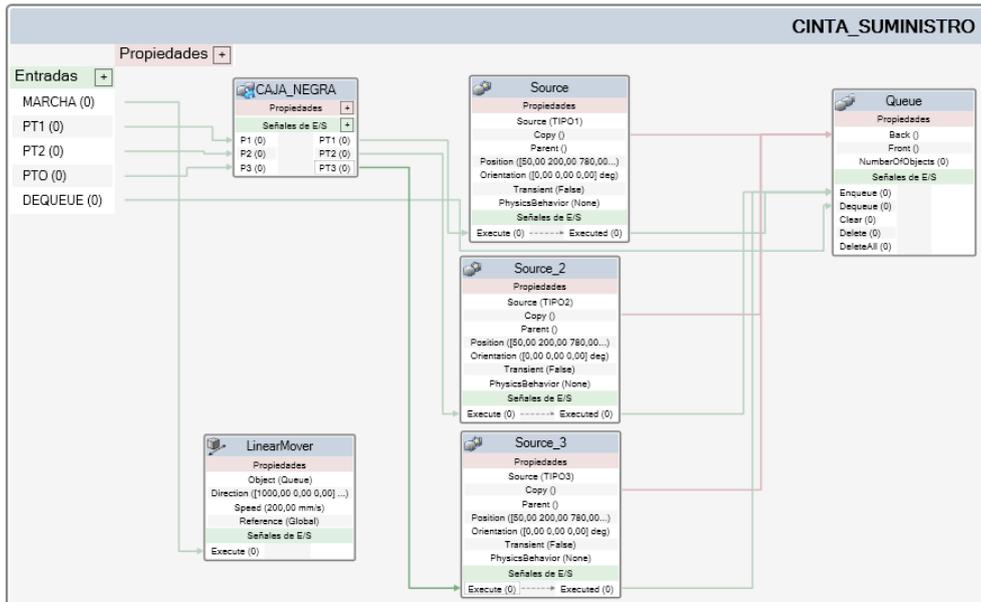


Figura 4.88 SC CINTA_SUMINISTRO

Como se observa en la figura anterior se han añadido cinco entradas, una entrada “MARCHA” que accionara el LinearMover mediante la entrada “Execute”; Tres entradas “PT1”, “PT2” y “PTO” para indicar la copia de las piezas tipo 1, tipo 2 o tipo otras en la cola cuando sea preciso conectadas a los Execute de cada pieza respectivamente; Una señal Dequeue conectada al Dequeue de la cola “Queue” para retirar los objetos en el momento deseado.

Además, se incorpora un pequeño SC “CAJA_NEGRA” que emulara la llegada de piezas de un proceso anterior no permitiendo que se genere una nueva pieza si no ha transcurrido un tiempo mínimo.

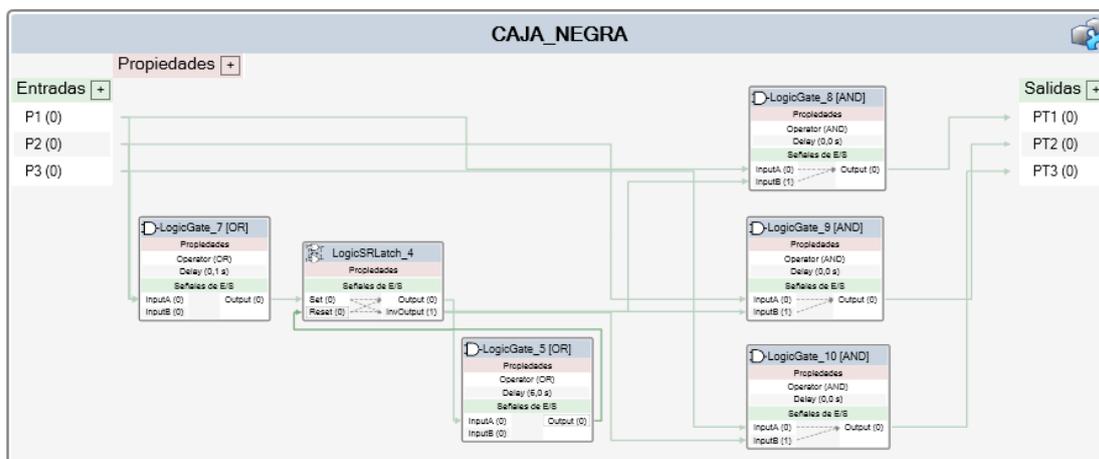


Figura 4.89 SC CAJA_NEGRA

- **SC COMPONENTES_ADICIONALES**

Se han agrupado en este bloque todos los elementos físicos, con una representación gráfica, necesarios en una estación real pero carentes de funcionalidad en el entorno de RobotStudio, como son una mesa, un Pc compuesto por pantalla y teclado, operarios, unidad de control IRC5, unidad de control de soldadura...

- **SC PERIMETRO_SEGURIDAD**

Este Smart Component incluye las medidas de seguridad necesarias además de los componentes físicos que la componen como las vallas de seguridad y puertas.

Está compuesto por tres componentes:

- LineSensor:

Representan sensores lineales que envían una señal "SensorOut" cuando su haz se corta, estos sensores están siempre activos y tras configurar su geometría y su posición en las puertas mandan una señal indicando que estas se encuentran cerradas. Como alternativa más económica y con la misma programación podría tratarse detectores magnéticos.

- LogicExpresion:

Representa una función lógica de triple AND que activara la salida cuando se verifiquen las tres entradas: que las dos puertas estén cerradas y la señal de MARCHA activa.

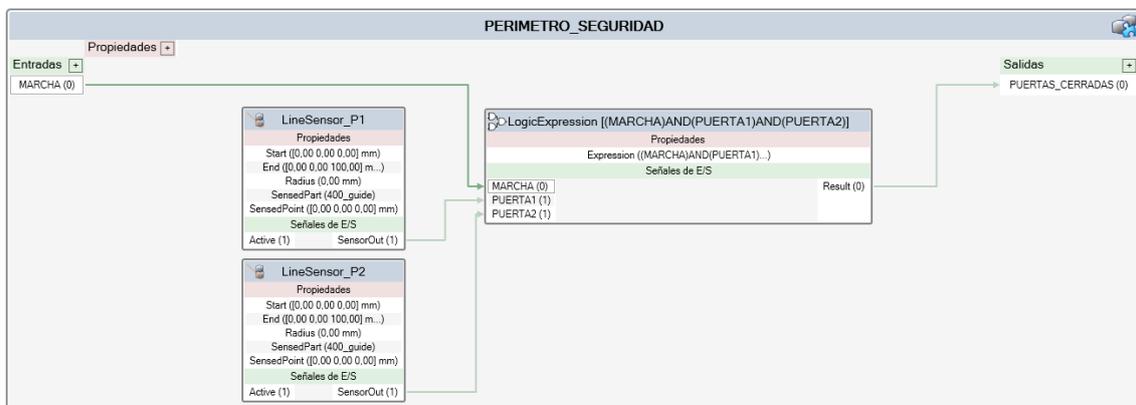


Figura 4.90 SC PERIMETRO_SEGURIDAD

Como se observa en la figura anterior(4.90), este SC solo requiere una entrada para recibir la entrada de MARCHA y una señal de salida “PUERTAS_CERRADAS” para permitir posteriores ejecuciones. La apertura de alguna puerta o la parada de marcha implicara una detención del proceso.

- **SC SINCRONIZACION_TAREAS**

Dentro del esquema de alto nivel encontramos el SC más complejo y responsable principal de la corrección del resto de procesos, si bien sus componentes se exploran con mayor detalle en sus apartados correspondientes aquí se realiza una primera aproximación a su funcionalidad.

Este bloque pretende sincronizar todas tareas a realizar a partir de un detector planar, para ello está compuesto por dos componente inteligentes “SOLDADURA” responsable de las operaciones de soldadura y “PICK_AND_PLACE” responsable de la catalogación y colocación de cada pieza en su cinta pertinente, en cuanto a componentes de programa solo está formado por uno:

- PlaneSensor:

Tras configurar su posición y dimensiones, detectará cuando un objeto corta su superficie e igualmente importante, cuando deja de hacerlo, este sensor esta activo siempre que la estación está en marcha y envía una señal “SensorOut” siempre que este detectando un objeto, esta señal será la entrada principal de los bloques de Soldadura y Pick-and-Place respectivamente.

Como entradas, dispone de entradas de “Marcha” y “Reset”, en ambos casos digitales y responsables de devolver el sistema a un estado inicial.

En cuanto a salidas dispone de cuatro, que serán las distintas señales de activación para el sistema de robots mediante instrucciones RAPID, siendo estas:

SOLDAR: Salida digital que indicara a la subestación de soldadura el instante en el que debe comenzar la operación.

APROXIMACION: Salida digital que marca el inicio del movimiento de la subestación de pick_and_place para enganchar el objeto finalizado.

NUEVA_DETECCION: Señal digital que indicara cuando una pieza ha sido correctamente catalogada.

TRAYECTORIA_DETECTADA: Señal analógica que tras clasificar las piezas en "Tipo 1", "Tipo 2" o "Tipo Otras" indicara la trayectoria para colocarlas en sus cintas correspondientes.

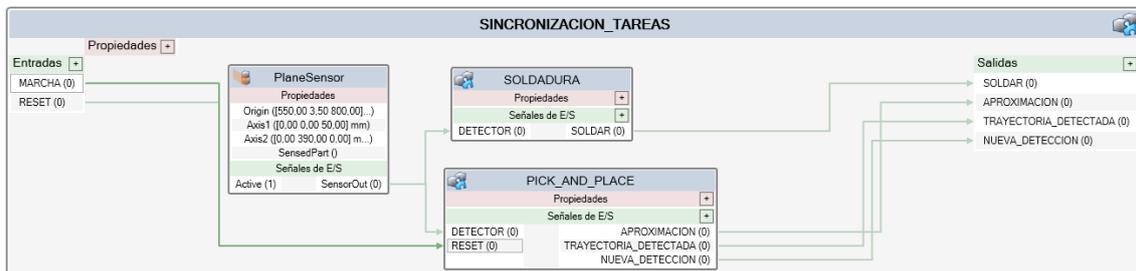


Figura 4.91 SC SINCRONIZACION_TAREAS

4.4.4.3.2 Entradas y Salidas

El nivel ha sido provisto de 5 entradas:

Marcha/Paro: Responsable de la activación de la estación.

Limpieza: Permite ejecutar las tareas de limpieza presionando un botón a deseo del operario.

PT1, PT2 y PTO: Nos permite simular la entrada de piezas a nuestro sistema.

5 RESULTADOS

Una vez diseñada y programada la estación, tras guardar y sincronizar pertinentemente, se procede a la simulación de la estación en funcionamiento.

Inicialmente se encuentra la estación en paro.

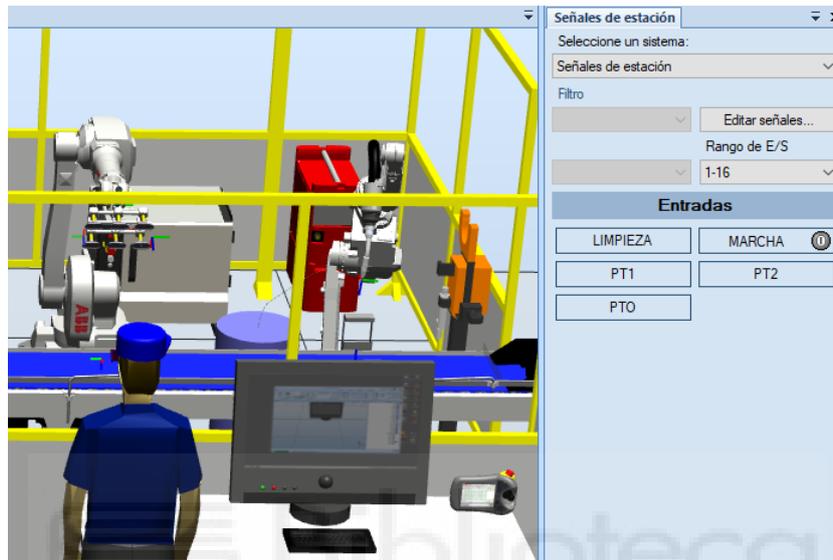


Figura 5.1 Estación parada

Tras activar la señal de MARCHA, se verifica que los sensores ubicados en las puertas (SC PERIMETRO_SEGURIDAD) están activos indicando que estas están cerradas y los robots adquieren su posición inicial hasta nuevas órdenes.

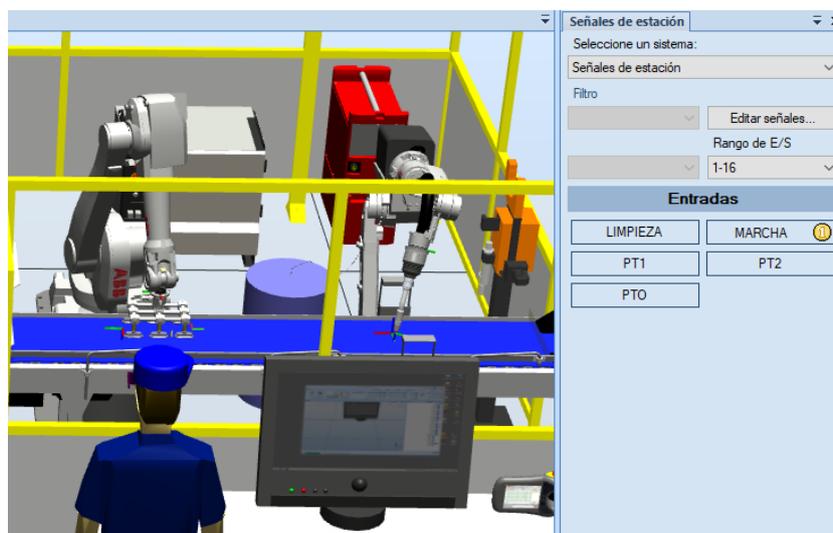


Figura 5.2 Estación en MARCHA

Pulsando sobre PT1, PT2 o PTO simulamos la entrada de una pieza al sistema proveniente de un proceso anterior. (Recordamos que, aunque pulsemos más veces no se generaran más piezas hasta alcanzar el tiempo de seguridad).

Comenzamos generando una pieza de Tipo 1.

La pieza generada avanza y corta el haz del sensor, activándolo e iniciando todos los procesos de sincronización de tareas (SC_SINCRONIZACION_TAREAS).

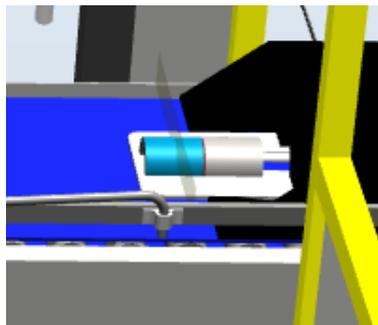


Figura 5.3 Pieza activando el sensor

Pasado el tiempo de admisión cargaremos otra pieza de Tipo 2.

5.1 Simulación de la Sub-Estación de soldadura

El SC SOLDADURA, en base al tiempo que la pieza corte el haz, calcula el tiempo que el punto medio de esta tardará en alcanzar la posición inicial de soldadura y llegado el momento activará la señal SOLDAR que dará lugar a la acción deseada.

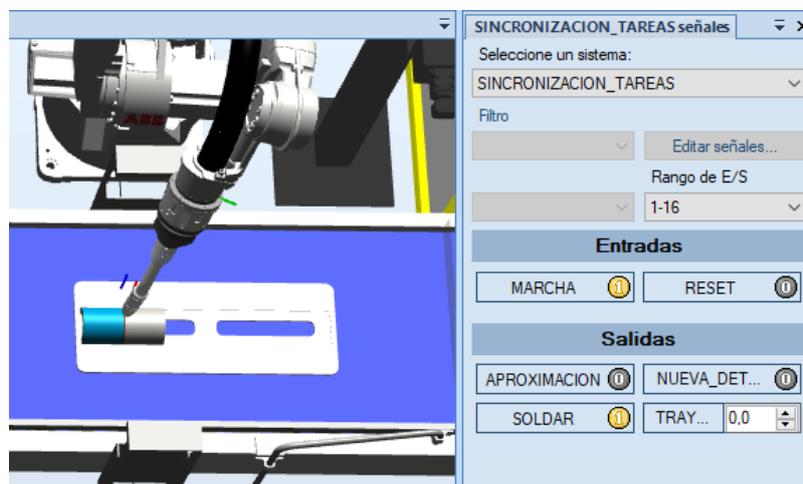


Figura 5.4 Inicio de soldadura

Finalizada la soldadura, la antorcha de soldadura se apaga y retorna a su posición de espera para la pieza siguiente. (Aprovechamos para generar una segunda pieza de test.)

Testeamos también la funcionalidad de la estación de limpieza de la antorcha, para ello, en cualquier momento que no se espere pieza para operar pulsamos el botón de limpieza.

Observamos como el Robot ejecuta de forma correcta la trayectoria posicionándose en el cortador de la estación de limpieza.

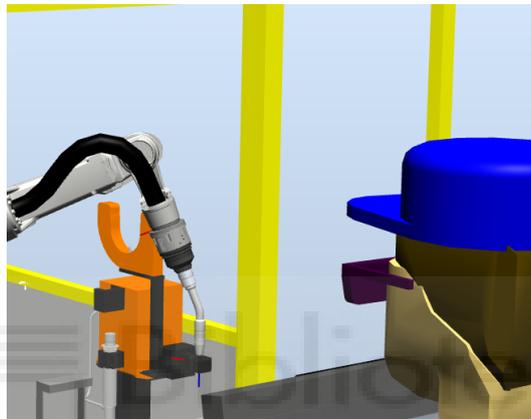


Figura 5.5 Limpieza - corte

Tras el tiempo necesario para la operación de corte, comprobamos como el robot se dirige a la posición de comprobación de TCP, para posteriormente regresar a su posición de espera hasta la recepción de nuevas piezas de trabajo.

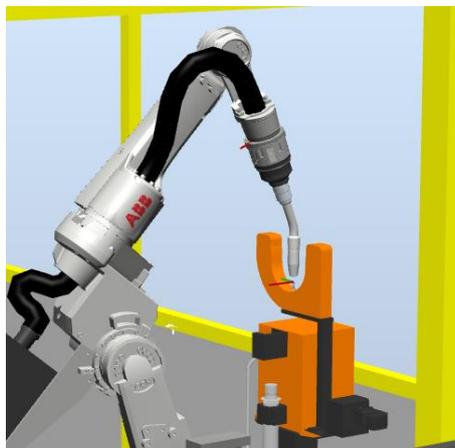


Figura 5.6 Limpieza - TCP

5.2 Simulación de la sub-Estación de Pick-and-place

Desde que la pieza fue detectada, el SC_APROXIMACION estimo el tiempo que debía tardar hasta enviar la señal, para aproximarse a la pieza.

Llegado el momento, se ejecuta el seguimiento de forma correcta y alcanzada la pieza se activa el sistema de ventosas mediante la señal "PICK", sujetándola.

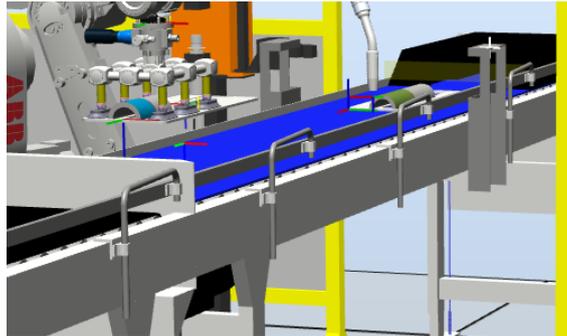


Figura 5.7 Pieza sujeta

Gracias al Smart Component PICKANDPLACE, en función del tiempo que corto el haz, la pieza fue catalogada en una de las tipologías definidas y, gracias a la multitarea (Tarea SUPERVISION), fue incluida en una cola FIFO la trayectoria que debería seguir para su posicionamiento en su cinta destino.

Llegado al punto de depositar la pieza o PLACE, se emite una señal "PLACE", soltando está en su cinta destino.

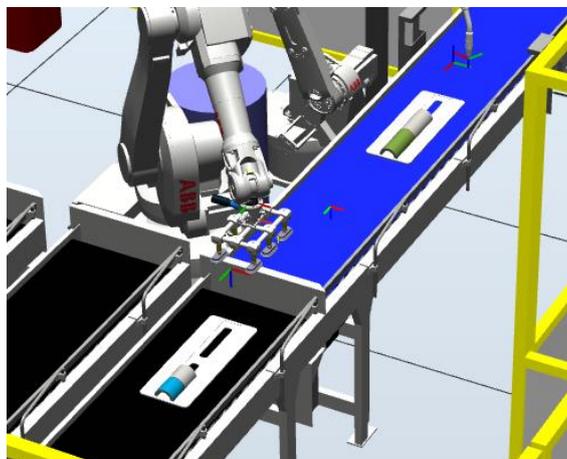


Figura 5.8 Pieza 1 depositada

Sigue evolucionando la simulación y comprobamos como efectivamente se repite satisfactoriamente el proceso con la pieza TIPO 2 y la Tipo Otros.

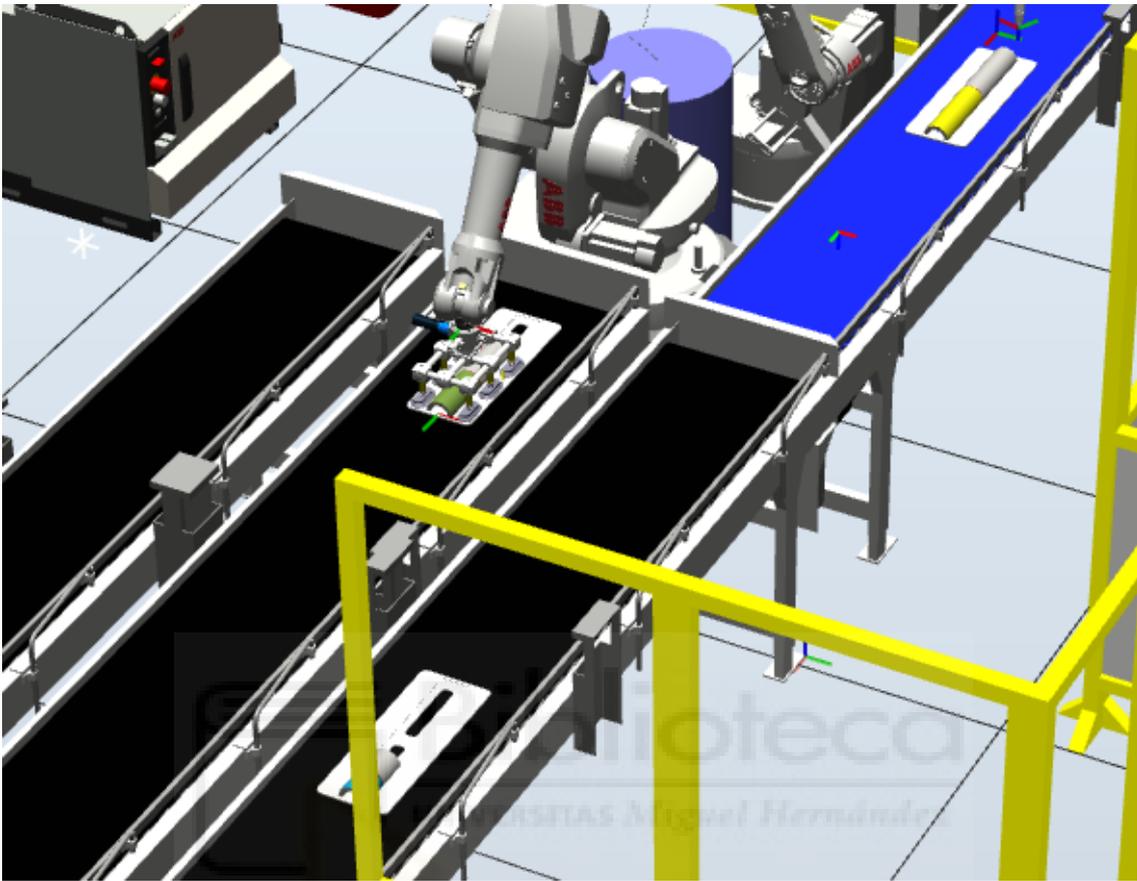


Figura 5.9 Pieza Tipo 2 Depositada

6 CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO

6.1 Conclusiones

Puesto que el principal objetivo de este TFM era la adquisición de conocimientos sobre el software RobotStudio y el desarrollo de las habilidades necesarias para resolver posibles complicaciones, considero que se han alcanzado las metas dentro del alcance del proyecto. Dicho esto, la profundidad del software, añadida a la cantidad de versiones y la ingente cantidad de módulos y aplicaciones disponibles imposibilita abarcarlo en su totalidad. De la misma manera RAPID es un lenguaje que requeriría mucha dedicación y estudio dado que sus posibilidades paralelizan con su complejidad.

El aprendizaje en ciertos puntos y detalles concretos ha sido lento, llegando a puntos muertos que han requerido la búsqueda de soluciones alternativas, acompañado por un proceso de investigación en manuales muy técnicos y faltos de ejemplos didácticos, y un continuo proceso de prueba-error.

Hablando de la implantación, dentro de las limitaciones impuestas y las decisiones tomadas parece correcta consiguiendo que la estación admita una cadencia de una pieza cada 5 segundos, teniendo en cuenta que no se han empleado velocidades máximas para evitar saltos, ni estas han sido debidamente optimizadas.

Por otro lado, la combinación de la búsqueda de un proceso tan flexible, continuo y la utilización de un único sensor, forzando una ejecución basada en la parametrización en función del tiempo y en ausencia de experimentación con una implementación real hace pensar que es una metodología muy dependiente de una calibración muy precisa en todos sus componentes. Por lo que, de tratarse de un proyecto real, quizás, se deberían valorar otras opciones como la utilización de varios sensores y realizar operaciones en parada. En este sentido, se hace una breve aproximación al proceso implantando un paro para el proceso de soldadura como se expone brevemente en el Anexo 8.2.

6.2 LÍNEAS FUTURAS DE TRABAJO

Dadas las complejidades expuestas en el punto anterior, así como la profundidad del software, las posibles líneas futuras de trabajo son muy numerosas, por lo que procedo a enumerar algunas de ellas aplicables a este TFM o futuros TFM de terceros.

Valorar la redefinición del proyecto principalmente incluyendo sensores y procesos stop-go.

Inclusión de controles de calidad y predicción mediante visión artificial.

Un estudio profundo del lenguaje RAPID como lenguaje universal de robótica sería inmensamente útil.

Implementación de seguridad más realista, ya que , según normativa, el enclavamiento del robot y paradas de emergencia se deberían realizar mediante un módulo externo al controlador del robot.

La exploración de RobotStudio en detalle y principalmente sus funciones de seguimiento de transportadores parece una herramienta muy potente pero limitada en cuanto a su aplicación directa en RobotStudio, quizás en Rapid existan mejores formas de emplearla.

Existen diversas Apps que facilitan y mejoran tareas específicas como la soldadura o el pick and place, un estudio de estas en búsqueda de soluciones rápidas y eficaces resultaría interesante.

7 BIBLIOGRAFÍA

- [1] Manual de referencia técnica, Instrucciones , funciones y tipos de datos RAPID. 3HAC16581-5
- [2] Manual del operador RobotStudio 3HAC032104-005_revD. ABB Robotics.
- [3] Application manual MultiMove 3HAC050961-001 ABB Robotics.
- [4] IRB 1600, Especificaciones del producto. ABB Robotics (2019)
- [5] IRB 1520, Especificaciones del producto. ABB Robotics (2019)
- [6] Guía de implantación de robots. Sector metal. FEMETAL.
- [7] Barrientos, A. et al. (2007) Fundamentos de Robótica. Madrid. McGrawHill/Interamericana de España, S.A.U. 2ª edición.
- [8] Roldán, D. J. Programación de un robot ABB para una estación de trabajo de soldadura de chasis para maquinaria. Universidad Miguel Hernández. TFM del Master de Robótica UMH. (2019)
- [9] Otón, P. Diseño, programación y simulación de una celda de soldadura robotizada para operaciones en carrocería. Universidad Politécnica de Cartagena. TFG del Grado en ingeniería electrónica industrial y automática.(2017)
- [10] <https://new.abb.com/es>
- [11] <https://library.abb.com/es>
- [12] <https://www.binzel-abicor.com/ES/spa/home/>
- [13] <https://www.schmalz.com/>
- [14] <https://www.fronius.com/es-es/spain>
- [15] <https://web.archive.org/web/20180609101518/http://www.aer-automation.com/aer-atp/robotica-industrial-y-de-servicio/>
- [16] <https://www.thefabricator.com/thefabricatorenspanol/article/automationrobotics/los-sensores-cambian-en-la-soldadura-automata>
- [17] <https://www.youtube.com/user/jcmcastillo>

8 ANEXOS

8.1 Glosario General

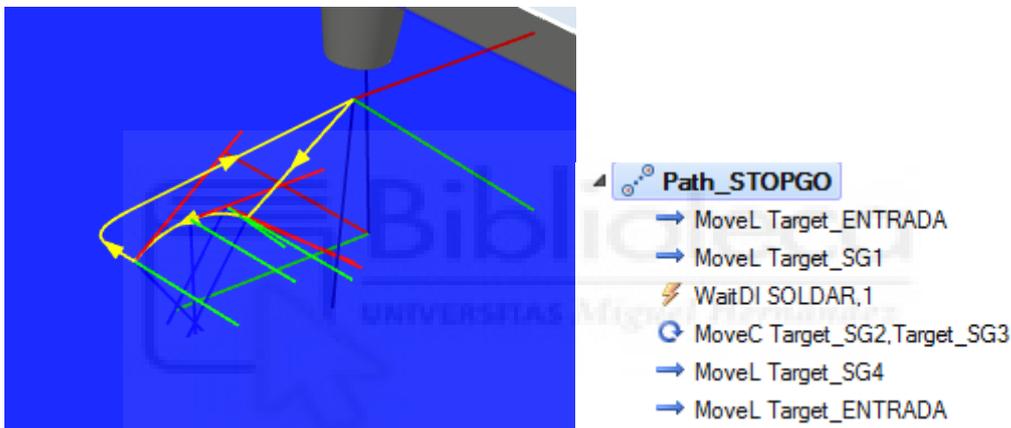
- **Devicenet:** Protocolo de comunicaciones de campo basado en CANBUS. Las tarjetas adicionales de los robots ABB se comunican con el armario del robot mediante Devicenet.
- **Flexcontroller:** Armario controlador IRC5, compuesto por el módulo de control y el módulo de accionamiento de motores.
- **FlexPendant:** Consola de mano para el operador.
- **Herramienta:** Dispositivo de fin de brazo montado en la muñeca del robot.
- **Path:** Trayectoria.
- **SC:** Smart Component.
- **Target:** Objetivo
- **TCP:** Tool Central Point, sistema de coordenadas establecido para la herramienta.
- **TFM:** Trabajo de fin de Master.
- **Pieza de trabajo:** Pieza sobre la que opera la estación.

8.2 Aproximación STOP&GO

A modo ilustrativo, una aproximación más tradicional al proceso de soldadura en cinta implica una parada para la realización de la tarea, suponiendo un considerable aumento de la precisión incrementando ligeramente el tiempo de ciclo, para ello, a la estación presentada durante el proyecto se le han realizado ligeras modificaciones como se resume a continuación.

Trayectoria

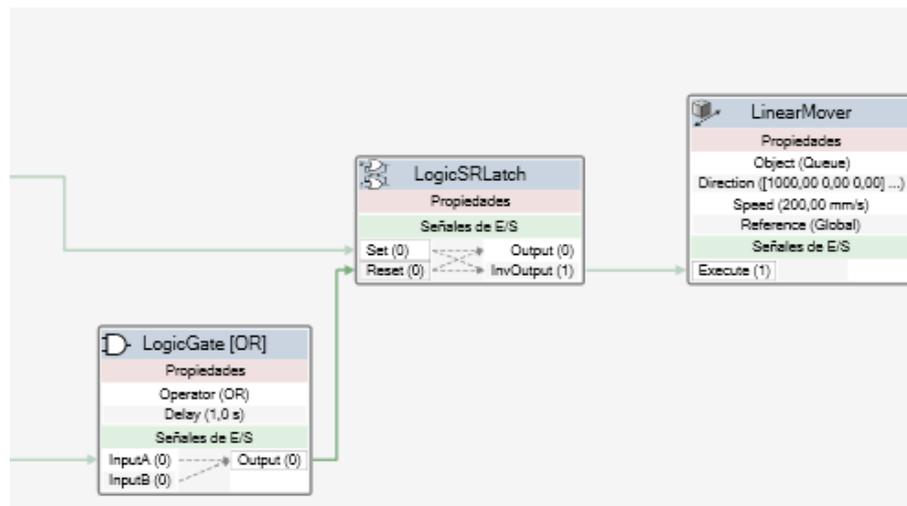
Se ha creado una trayectoria para la ejecución de la soldadura encontrándose la pieza de trabajo en reposo.



Smart Components

En el componente inteligente CINTA_SUMINISTRO, se ha añadido una entrada digital "STOPGO" enlazada con la señal de la soldadura, de manera que aprovechemos esta señal para parar la cinta mediante un LogicSRLatch que es reestablecido mediante una puerta lógica OR tras un segundo reactivando la cinta.

En el componente APROXIMACION, se ha incrementado un segundo (10 s) para adaptar el pick a las nuevas condiciones.



En esta configuración y sin entrar detalles de diseño se consigue una precisión absoluta a coste de un segundo por ciclo.



8.3 Fichas técnicas

8.3.1 Especificaciones técnicas IRB 1520ID

Robótica

IRB 1520ID

El robot eficiente para la soldadura arco

Este robot de soldadura arco de alta precisión, con configuración de proceso integrada, combina la exigencia de una producción 24/7 con una reducción del 50% en el coste de mantenimiento, para ofrecer el coste más bajo por cordón de soldadura de su categoría.



El robot específico para soldadura arco

En el IRB 1520ID (Integrated Dressing), el paquete de mangueras está totalmente integrado en el brazo superior y a través de la base del robot. Esto significa que todos los medios necesarios para la soldadura arco, como la potencia, el hilo de soldadura, el gas de protección y el aire comprimido, están orientados para conseguir las mejores prestaciones y eficiencia energética. El IRB 1520ID proporciona una soldadura estable, excelente precisión de trayectoria, reducción en el tiempo de ciclo y una mayor durabilidad del paquete de mangueras. Gracias a la configuración integrada, pueden efectuarse soldaduras en torno a objetos cilíndricos sin ninguna parada y con una mayor facilidad de acceso a espacios reducidos.

Instalación flexible

Con una capacidad de carga de 4 kg y un radio de trabajo de 1,5 metros, el compacto IRB 1520ID puede montarse tanto en el suelo como en posición invertida. Esta flexibilidad de colocación ofrece reducidos tiempos de ciclo y una amplia gama de posibilidades de producción.

Excelente precisión y velocidad

Los robots de ABB son famosos por su óptimo control de trayectoria. Con la tecnología TrueMove™ el IRB 1520ID ofrece una extraordinaria precisión, con la tecnología QuickMove™ el robot puede utilizar una aceleración máxima entre soldaduras aumentando la producción y al mismo tiempo reducir el consumo de energía.

Fácil de usar, fácil de programar

Tanto el robot como el proceso de soldadura arco se programan y se mantienen fácilmente desde la unidad de programación FlexPendant de ABB. Una interfaz gráfica e intuitiva que permite a los operarios controlar el robot y el equipo de soldadura en su propio idioma. Equipado con una pantalla táctil y con el joystick exclusivo de ABB para conseguir un posicionamiento rápido y sencillo del robot. Respecto a las ventajas de la simulación y la programación "offline", ABB ofrece los paquetes de software más populares, fiables y rentables, RobotStudio™ y RobotStudio Arc Welding Power-Pac con VirtualArc™. Con VirtualArc™, usted dispone de una herramienta que le permite la correcta parametrización de los datos de soldadura de un modo totalmente virtual, evitando errores. En unas pocas horas, el robot estará produciendo con los tiempos de ciclo previstos y con unas soldaduras de alta calidad.

Servicio y asistencia mundial

Para un funcionamiento sin problemas, ABB ofrece asimismo RemoteService, que proporciona acceso a distancia al equipo para su supervisión y soporte. Además, los clientes de ABB pueden aprovechar la organización de servicio de la empresa, que con más de 35 años de experiencia en el sector de la soldadura arco, ABB proporciona soporte de servicio en más de 100 emplazamientos en 53 países.

www.abb.com/robotics

Power and productivity
for a better world™ **ABB**

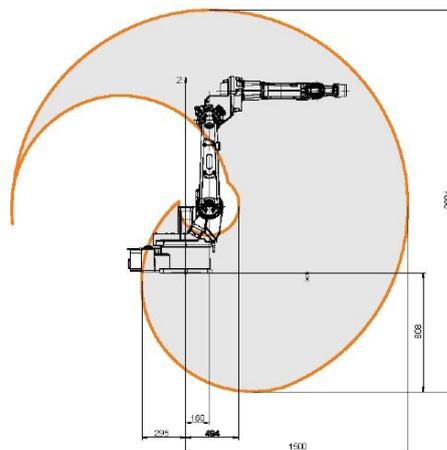
IRB 1520ID

Principales aplicaciones		
Soldadura arco		
Especificación		
Carga útil	4 kg	
Carga en el brazo	10 kg	
Radio de trabajo	1,50 m	
Número de ejes	6	
Protección	IP40	
Montaje	En suelo, invertido	
Variantes de controlador IRC5	Módulo de accionamiento, armario sencillo	
Características físicas		
Dimensiones de la base del robot	300 x 300 mm	
Peso del robot	170 kg	
Prestaciones (según la norma ISO 9283)		
Repetibilidad de la posición (RP)	0,05 mm	
Repetibilidad del recorrido (RT)	0,35 mm	
Movimiento		
Movimientos de los ejes	Campo de trabajo	Velocidad máxima
Eje 1	+ 170° a -170°	130°/s
Eje 2	+ 150° a -90°	140°/s
Eje 3	+80° a -100°	140°/s
Eje 4	+ 155° a -155°	320°/s
Eje 5	+ 135° a -90°	380°/s
Eje 6*	+200° a -200°	460°/s
* Máx +288 rev. a -288 rev.		
Conexiones eléctricas		
Tensión de alimentación	380 V	
Consumo	0,6 kW cubo ISO	
Condiciones ambientales		
Temperatura ambiente para la unidad mecánica:		
Durante el funcionamiento	+5°C (41°F) a +45°C (113°F)	
Durante el transporte y el almacenamiento	-25°C (13°F) a +55°C (131°F)	
Durante periodos cortos (máximo 24h)	hasta +70° C (158° F)	
Humedad relativa	Máx.95% a temperatura constante	
Seguridad	Circuitos dobles con supervisión, paradas de emergencia y funciones de seguridad, dispositivo que permite 3 posiciones	
Emisiones	Protección EMC/EMI	

Los datos y las dimensiones pueden cambiarse sin previo aviso

Campo de trabajo

IRB 1520ID-4/1.5



8.3.2 Especificaciones técnicas IRB 1600 1.45/10



ROBOTICS

IRB 1600

The highest performance 10 kg robot



Performance is often a trade off, optimizing for speed or accuracy. With ABB's IRB 1600, you don't have to choose. The robot's cycle times are shorter, sometimes half that of other robots, allowing you to increase throughput. Meanwhile, you will enjoy the work piece quality that only an ABB robot can offer. Extra everything.

Double your throughput

The IRB 1600 has up to 50 percent shorter cycle times than competing robots in material handling, machine tending and process applications. It speeds up and slows down faster than other robots, saving time while moving between tasks. This is possible due to ABB's patented second generation QuickMove motion control, combined with the robot's strong motors and low friction losses in the spur gears.

No more cutting corners

At high speed, most robots will cut corners. With the IRB 1600, the path will be the same regardless of speed, thanks to the robot's unique combination of brains and brawn. Intelligent second generation TrueMove motion control means that "what-you-program-is-what-you-get". Add muscle – a heavy and stiff design – low vibrations and low friction – and you have a robot that will deliver consistently high work piece quality, high yield and few rejects.

Outstanding reliability

The IRB 1600 offers outstanding reliability, even in the toughest environments and the most demanding 24/7 duty cycles. The entire manipulator is IP 54 classed and sensitive parts are IP 67 classed as standard.

The optional protection Foundry Plus offers IP 67, special paint, rust protection and is tailor made for tough foundry environments. The rigid and heavy design combined with spur gears, make the robot extremely robust. Smart collision detection software further adds to the robot's outstanding reliability.

Easy to integrate

Mounting is fully flexible: on a shelf, on the wall, tilted or inverted. By choosing the compact short-arm version with the 1.2 m reach, you can even fit the IRB 1600 inside a machine, while ensuring sufficient payload as the maximum total load is as high as 36 kg. Sustainable and healthy Low friction spur gears, and no unnecessary moves due to QuickMove and TrueMove, reduces power consumption down to 0.58 kW at max speed, and even less at low speeds. The airborne noise level of just <70 dB (A) secures a healthy sound environment.

Main applications

- Assembly
- Arc Welding
- Material Handling
- Machine Tending
- Material Removal
- Cleaning/Spraying
- Dispensing
- Packing

Specification

Robot version	Reach (m)	Payload (kg)	Armload (kg)
IRB 1600-6/1.2	1.2	6	30.5
IRB 1600-6/1.45	1.45	6	30.5
IRB 1600-10/1.2	1.2	10	20.5
IRB 1600-10/1.45	1.45	10	20.5
Number of axes	6+3 external (up to 36 with MultiMove)		
Protection	Standard IP54 Option FoundryPlus 2 (IP67)		
Mounting	Floor, wall, shelf, tilted, inverted		
Controller	IRC5 Single Cabinet/IRC5 Compact		

Performance (according to ISO 9283)

	Position repeatability	Path repeatability
IRB 1600-6/1.2	0.02 mm	0.13 mm
IRB 1600-6/1.45	0.02 mm	0.19 mm
IRB 1600-10/1.2	0.02 mm	0.06 mm
IRB 1600-10/1.45	0.05 mm	0.13 mm

Technical information

Electrical Connections

Supply voltage	200-600 V, 50-60 Hz
Energy consumption	0.58 kW

Physical

Robot base	484 x 648 mm
Robot height	
IRB 1600-6/1.2	1069 mm
IRB 1600-10/1.2	1069 mm
IRB 1600-6/1.45	1294 mm
IRB 1600-10/1.45	1294 mm
Robot weight	250 kg

Environment

Ambient temperature for mechanical unit	
During operation	+ 5°C (41°F) to + 45°C (113°F)
During transportation and storage	- 25°C (- 13°F) to + 55°C (131°F)
During short periods (max. 24 h)	up to + 70°C (158°F)
Relative humidity	Max. 95%
Safety	Double circuits with supervisions, emergency stops and safety functions, 3-position enable device

Emission

EMC/EMI shielded

Data and dimensions may be changed without notice.

Movement

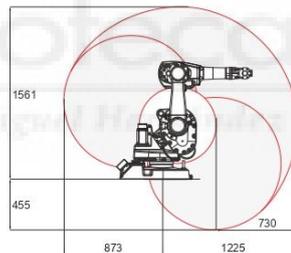
Axis movement	Working range IRB 1600-6/1.2, IRB 1600-10/1.2	Working range IRB 1600-6/1.45, IRB 1600-10/1.45
Axis 1 rotation	+180° to -180°	+180° to -180°
Axis 2 arm	+110° to -63° +136° to -63° ¹	+120° to -90° +150° to -90° ²
Axis 3 arm	+55° to -235°	+65° to -245°
Axis 4 rotation	Default: +200° to -200° Max. rev: +190° to -190°	Default: +200° to -200° Max. rev: +190° to -190°
Axis 5 bend	+115° to -115°	+115° to -115°
Axis 6 turn	Default: +400° to -400° Max. rev: +288 to -288	Default: +400° to -400° Max. rev: +288 to -288

¹With axis 1 limited to ±100° ²With axis 1 limited to ±95°

Axis max. speed

	IRB 1600-6/1.2, IRB 1600-6/1.45	IRB 1600-10/1.2, IRB 1600-10/1.45
Axis 1 rotation	150°/s	180°/s
Axis 2 arm	160°/s	180°/s
Axis 3 arm	170°/s	185°/s
Axis 4 rotation	320°/s	385°/s
Axis 5 bend	400°/s	400°/s
Axis 6 turn	460°/s	460°/s

Working range, IRB 1600-6/1.2, IRB 1600-10/1.2



Working range, IRB 1600-6/1.45, IRB 1600-10/1.45

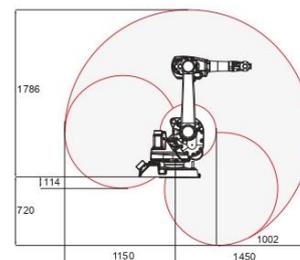


abb.com/robotics

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB. Copyright © 2019 ABB. All rights reserved.

PR10282EN Rev. E January 2019

8.3.3 Antorcha de soldadura BIZEL ABIROB

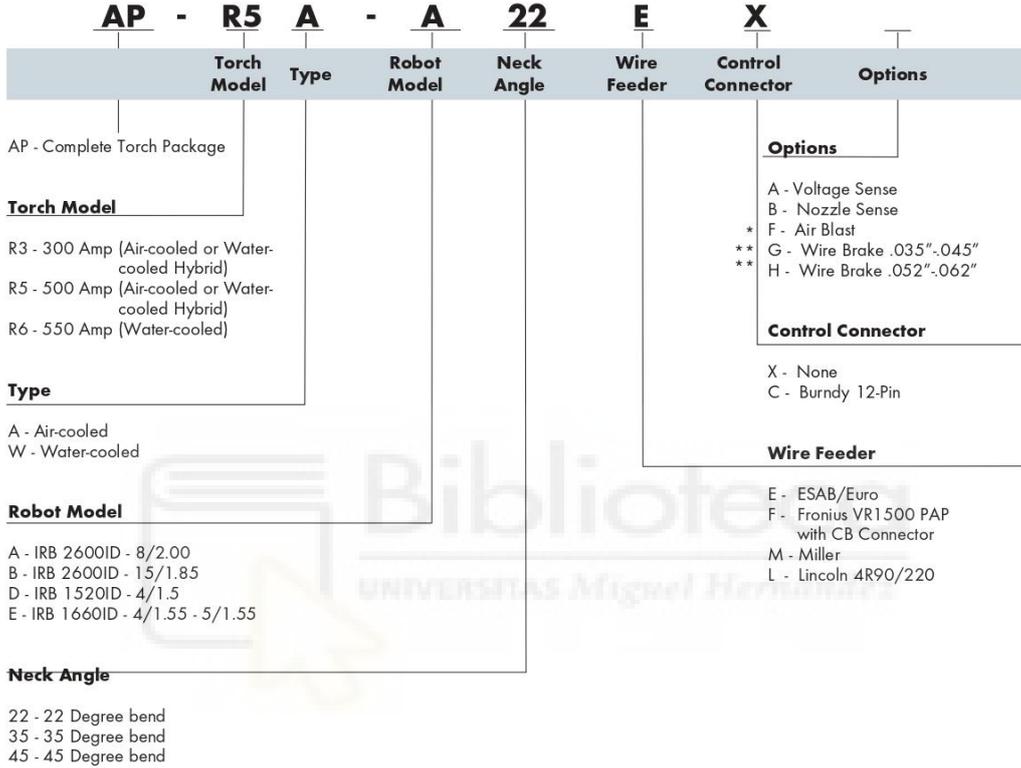


TECHNOLOGY FOR THE WELDER'S WORLD.

ROBO TORCHES

ABB IRB 2600ID, IRB 1520ID, IRB 1660ID
ABIROB 300/500/550

Ordering Instructions for Complete ROBO Torches



* Note: Air Blast is not available for cables with wire brake options.
** Wire brake available on air-cooled models only.

Binzel S.A DE C.V.
Municipio Pabellon de Arteaga 102
Valle De Aguascalientes, 20358
San Francisco de los Romo, Ags. Mexico
Phone: +52(499) 973.0133
Fax: +52(499) 973.1388
E-Mail: info@binzel.com.mx

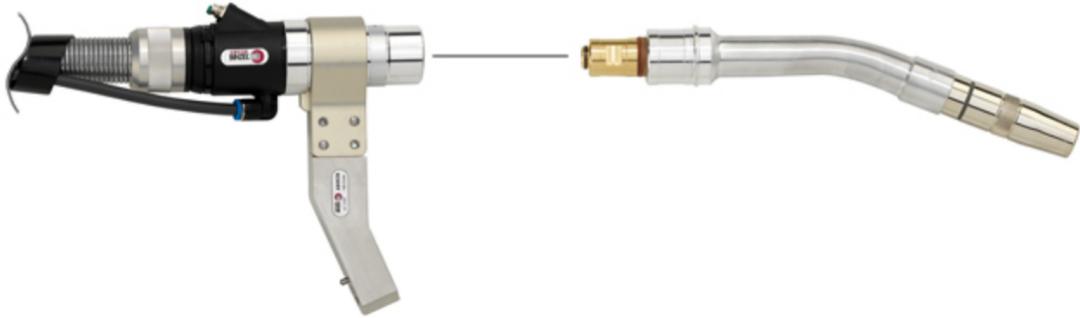


ABICOR BINZEL USA, Inc.
650 Medimmune Court, Suite 110
Frederick, MD 21703
Phone: 800.542.4867
Fax: 301.846.4497
E-Mail: customerservice@abicorusa.com

www.binzel-abicor.com

ABB.0003 • 05 - 18 • © ABICOR BINZEL USA, Inc. • Printed USA

ABIROB® A 360 (enfriada por aire)



Información técnica según EN 60 974-7:

Enfriamiento: Aire

Capacidad: 360 A CO₂ / 290 A Mezcla de gases M 21 según DIN EN ISO 14175

Ciclo de trabajo: 100%

Alambre-Ø: 0.8 - 1.4 mm (.030 - .052")

Datos técnicos

VR 1500-PAP

Alimentación de tensión	55 V DC
Corriente nominal	4 A
Diámetro del hilo	0,8 - 1,6 mm 0.03 - 0.06 in.
Velocidad de hilo	0,5 - 22 m/min 19.69 - 866.14 ipm.
Par	4 Nm
Tipo de protección	IP 21
Dimensiones (longitud x anchura x altura)	320 x 230 x 265 mm 12.60 x 9.06 x 10.43 in.
Peso	5,7 kg 12.57 lb.
Accionamiento	Accionamiento a 4 rodillos
Presión máxima del gas protector	7 bar 101 psi.

8.3.4 Estación de limpieza



Technical data

Technical data

General data

Total weight:	approx. 16 kg (incl. TMS-VI and DAV)
Ambient temperature:	+ 5°C to + 50°C
Air consumption:	approx. 380 l/min.
Pneumatic motor (Nominal speed):	with lubricated air: approx. 650 rpm / with non-lubricated air: approx. 550 rpm

Pneumatic connection – manifold block

Compressed air supply:	G 1/4
Clear width:	min. Ø 6 mm
Nominal pressure:	6 bar
Operating pressure:	6–8 bar

Electrics – terminal block

Control:	4 inputs for triggering the 5/2 directional control valves
Control voltage:	24 V DC
Power consumption:	4.5 W
Output:	1 output from inductive proximity switch (pnp)
Operating voltage:	10 – 30 V DC
Tolerated residual ripple:	V _{ss} < 10%
Continuous current:	max. 200 mA
Current consumption:	approx. 4 mA (24 V)
Drop in voltage:	approx. 1.2 V (200 mA)

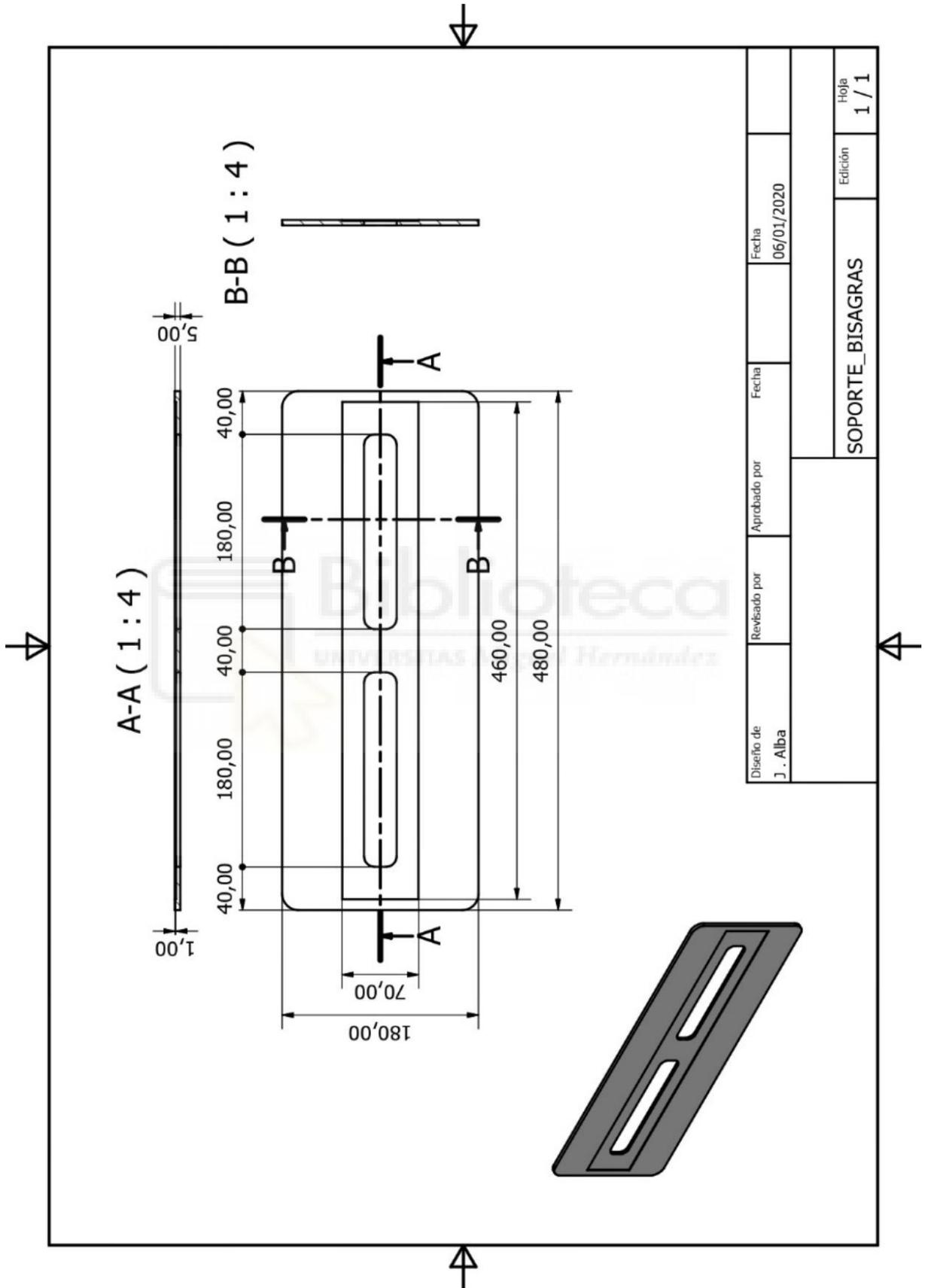
Front injector “TMS-VI”

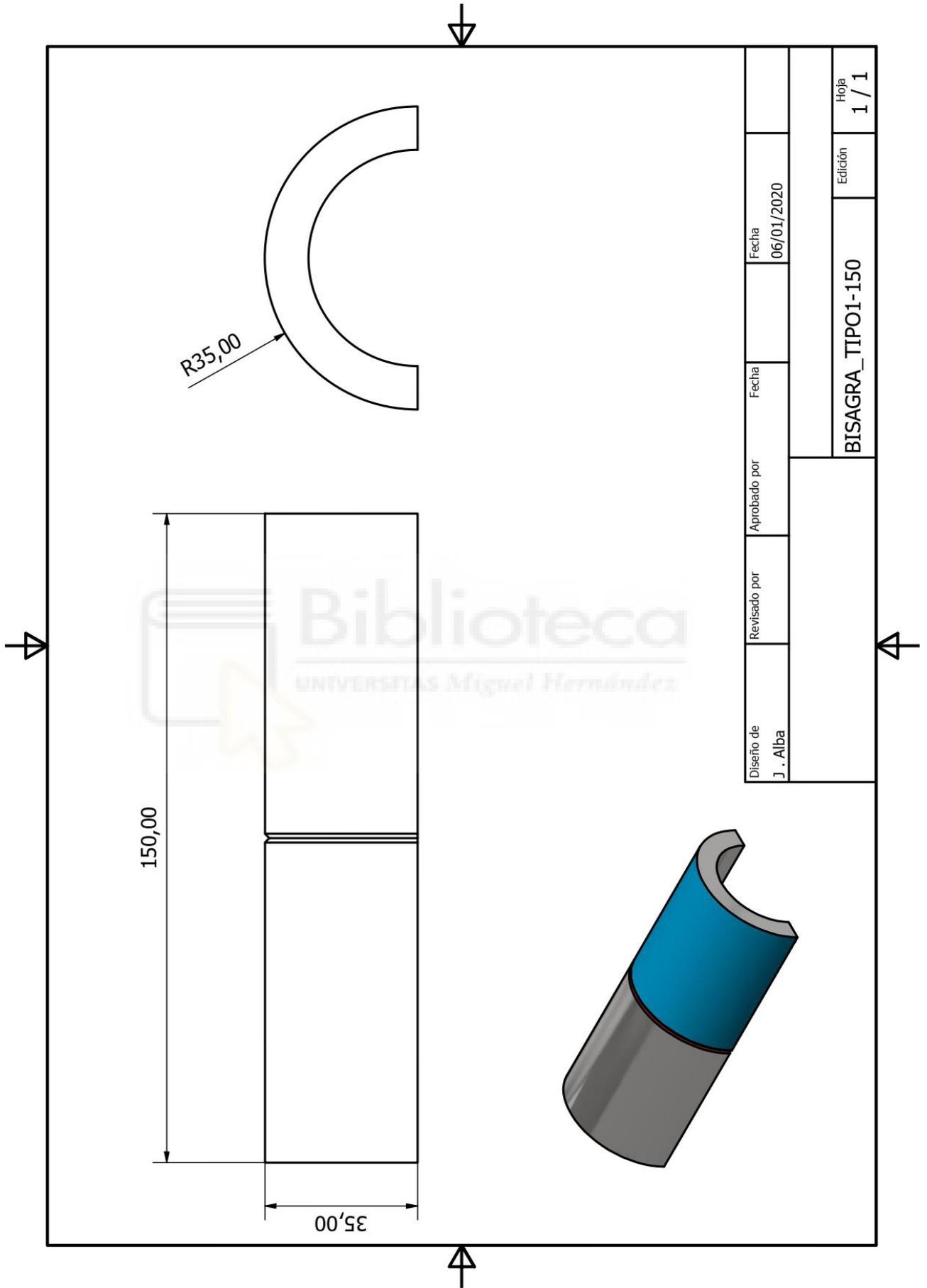
Capacity of the bottle:	1 litre
--------------------------------	---------

Wire cutting fixture “DAV”

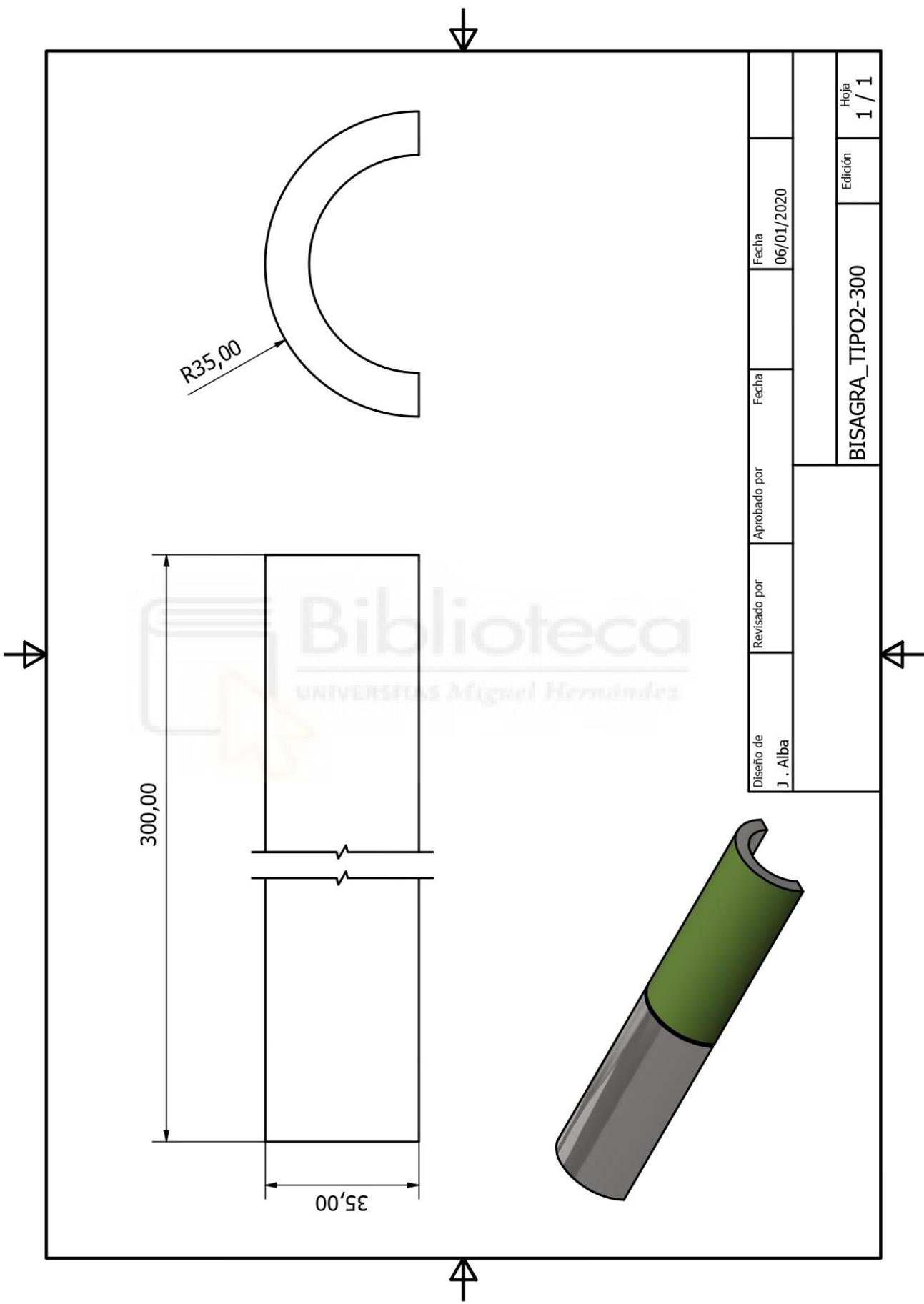
Cutting rate at 6 bar:	Solid wire: up to 1.6 mm / Flux cored wire: up to 3.2 mm
Cutting time:	0.5 sec.

8.4 Planos



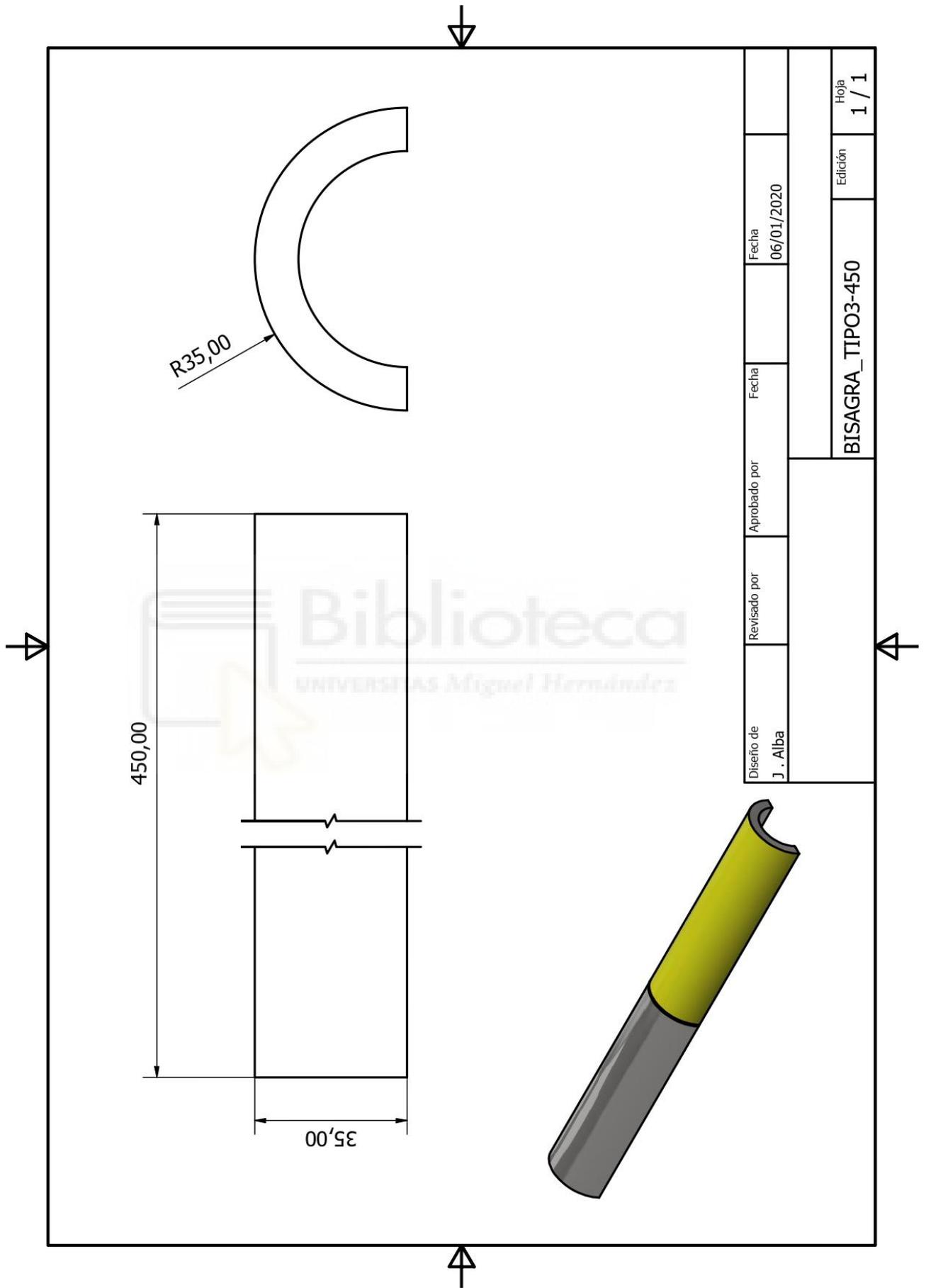


Diseño de J . Alba	Revisado por	Aprobado por	Fecha	Fecha 06/01/2020
BISAGRA_TIPO1-150			Edición	Hoja 1 / 1

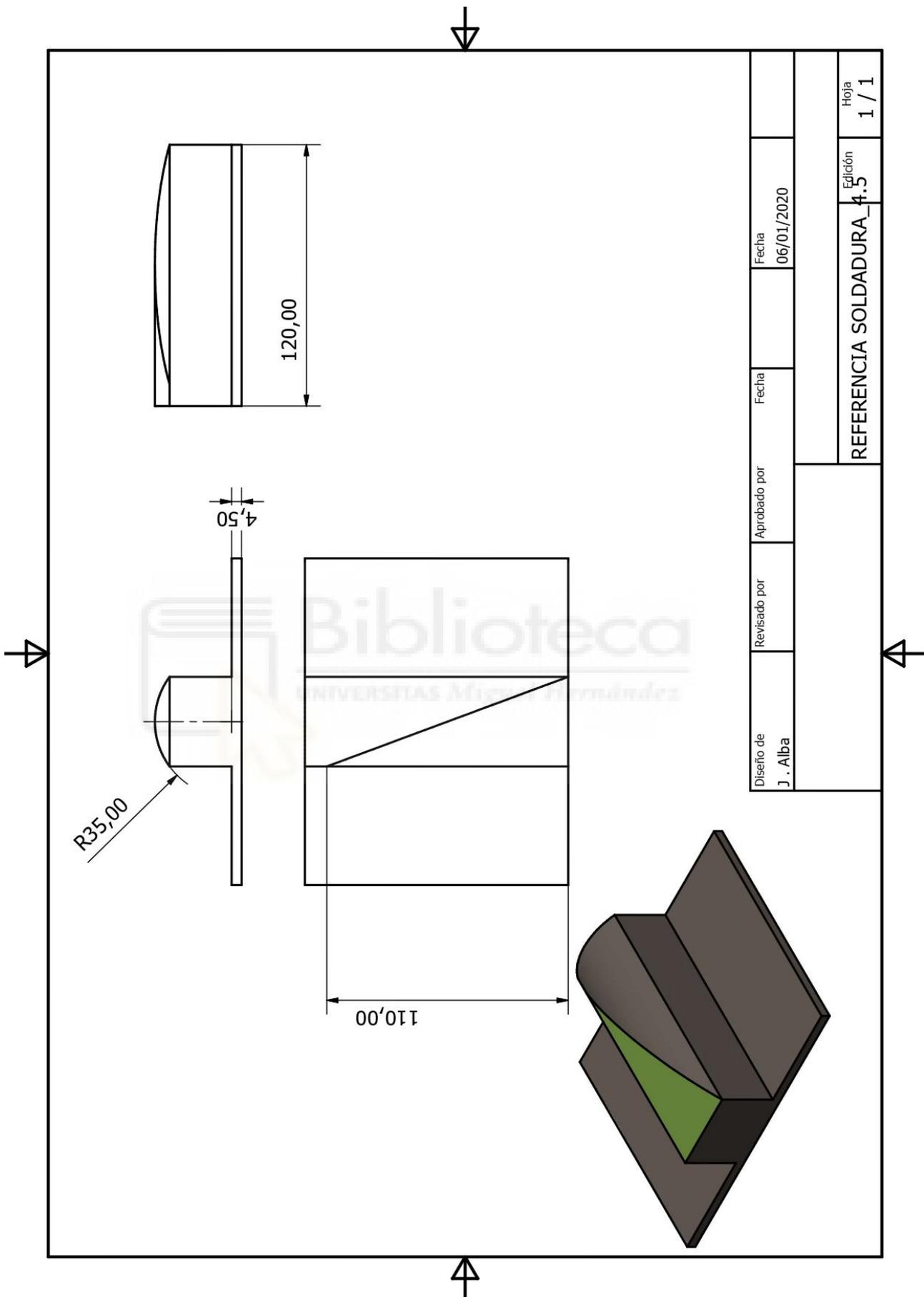


Biblioteca
UNIVERSITAS Miguel Hernández

Diseño de J . Alba	Revisado por	Aprobado por	Fecha	Fecha 06/01/2020
BISAGRA_TIPO2-300			Edición	Hoja 1 / 1



Diseño de J . Alba	Revisado por	Aprobado por	Fecha	Fecha 06/01/2020
BISAGRA_TIPO3-450			Edición	Hoja 1 / 1



Diseño de J . Alba	Revisado por	Aprobado por	Fecha	Fecha
			06/01/2020	
REFERENCIA SOLDADURA_4.5				Edición 4.5
				Hoja 1 / 1

8.5 Código RAPID

8.5.1 Tarea T_ROB1 (SOLDADURA)

```
MODULE Module1

CONST robtarget Target_ENTRADA:=[[-32.279227934,30.791856098,68.543028738],
[0.053490515,-0.726913262,0.684177283,-0.025245209],[-1,-1,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_SALIDA:=[101.623907132,118.208675704,49.927278354],
[0.053490515,-0.726913262,0.684177283,-0.025245209],[-1,0,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_WELD:=[55,75,35],[0,-0.573576437,0.819152044,0],[-1,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_INICIO_WELD:=[0,54.428,28.316],[0.177244866,-
0.545503608,0.779059889,0.253131902],[-
1,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_FIN_WELD:=[110,95.572,28.316],[0.177244866,0.545503608,-
0.779059889,0.253131902],[-1,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget HOME1:=[[-32.27919825,-
180.210372518,343.346033508],[0.053490495,-0.726913264,0.684177282,-
0.025245222],[-1,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
VAR intnum intLIMPIEZA;
CONST robtarget Target_L1:=[[-7,273,914],[0,0.923879533,-0.382683432,0],[0,-1,0,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_L2:=[[-7,273,884],[0,0.923879533,-0.382683432,0],[0,-1,0,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_L3:=[[-7,273,879],[0,0.923879533,-0.382683432,0],[0,-1,0,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_L4:=[114.129,-4.863,1215],[0,1,0,0],[0,0,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_L5:=[0.086,-21.438,1215],[0,1,0,0],[0,0,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_L6:=[0.086,-21.438,1151.641],[0,1,0.000000001,0],[0,0,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
VAR speeddata Vel_WELD:=[215.53,500,5000,1000];

!*****
! Módulo: Module1
!
! Descripción: Soldadura y limpieza de antorcha.
!
! Autor: J . Albarranch
!*****

PROC main()

IDelete intLIMPIEZA;
CONNECT intLIMPIEZA WITH LIMPIEZA_ANTORCHA;
ISignalDI LIMPIEZA,1,intLIMPIEZA;
```

```
    WHILE MARCHA=1 DO !FUNCIONAMIENTO MIENTRAS ESTE LA MARCHA
PUESTA (INCLUIDA SEGURIDAD DE LAS PUERTAS)
```

```
    Path_WELD;
    ENDWHILE
    Path_PARADA;
```

```
ENDPROC
```

```
PROC Path_WELD()
```

```
    MoveL Target_ENTRADA,v1000,z100,tWeldGun\WObj:=WO_WELD;
    MoveL Target_INICIO_WELD,v1000,fine,tWeldGun\WObj:=WO_WELD;
    WaitDI SOLDAR,1; !ESPERA LA ORDEN DE SOLDAR
    MoveC
Target_WELD,Target_FIN_WELD,Vel_WELD,fine,tWeldGun\WObj:=WO_WELD;
    MoveL Target_SALIDA,v1000,z100,tWeldGun\WObj:=WO_WELD;
    MoveL Target_ENTRADA,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
    ENDPROC
```

```
PROC Path_PARADA()
```

```
    MoveL HOME1,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
ENDPROC
```

```
TRAP LIMPIEZA_ANTORCHA
```

```
    Path_LIMPIEZA;
    MoveL Target_INICIO_WELD,v1000,fine,tWeldGun\WObj:=WO_WELD;
```

```
ENDTRAP
```

```
PROC Path_LIMPIEZA()
```

```
    MoveL HOME1,v1000,z100,tWeldGun\WObj:=WO_WELD;
    MoveL Target_L1,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
    MoveL Target_L2,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
    MoveL Target_L3,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
    WaitTime 3;
    MoveL Target_L1,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
    MoveL Target_L4,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
    MoveL Target_L5,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
    MoveL Target_L6,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
    WaitTime 1;
    MoveL Target_L5,v1000,z100,tWeldGun\WObj:=WO_LIMPIEZA;
    MoveL HOME1,v1000,z100,tWeldGun\WObj:=WO_WELD;
```

```
ENDPROC
```

```
ENDMODULE
```

8.5.2 Tarea SUPERVISION

```
MODULE Module1
!*****
! Módulo: Module1
!
! Descripción:
! Modulo de supervision para la creacion y actualizacion de una cola FIFO y la
generacion de trayectorias siguientes
!
! Autor: J . Albarranch
!
!*****
VAR num CONTADOR;           !Variable contador
VAR num VECTOR_TRAYECTORIAS{3}; !vector de trayectorias almacenadas

PROC main()

    CONTADOR:=0;           !INICIALIZA CONTADOR
    VECTOR_TRAYECTORIAS:=[0,0,0]; !INICIALIZA VECTOR DE
TRAYECTORIAS

    WHILE MARCHA=1 DO

        WaitDI NUEVA_DETECCION,1; !ESPERA A DETECTAR TRAYECTORIA

        CONTADOR:=CONTADOR+1; !ACTUALIZA CONTADOR
        VECTOR_TRAYECTORIAS{CONTADOR}:=TRAYECTORIA_DETECTADA;
!ACTUALIZA EL VECTOR
        WaitDI NUEVA_DETECCION,0;

        SetAO TRAYECTORIA_OUT,VECTOR_TRAYECTORIAS{1};
!TRAYECTORIA A EJECUTAR

        IF ACTUALIZAR_IN=1 THEN !ACTUALIZACION DEL VECTOR
            CONTADOR:=CONTADOR-1;
            VECTOR_TRAYECTORIAS{1}:=VECTOR_TRAYECTORIAS{2};
            VECTOR_TRAYECTORIAS{2}:=VECTOR_TRAYECTORIAS{3};
            VECTOR_TRAYECTORIAS{3}:=0;
            SetDO ACTUALIZAR_OUT,0;
        ENDIF

    ENDWHILE

ENDPROC
ENDMODULE
```

8.5.3 Tarea T_ROB2 (Pick and Place)

```
MODULE Module1
  CONST robtarget Target_0_1:=[[-661,-241,835],[0,0.707106781,0.707106781,
0],[0,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_0_2:=[[-461,-241,785],[0,0.707106781,0.707106781,
0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_0_3:=[[-361,-241,950],[0,0.707106781,0.707106781,
0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_1_1:=[[439,-241,950],[0,0.707106781,0.707106781,
0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_1_2:=[[439,-241,785],[0,0.707106781,0.707106781,
0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_1_3:=[[189,-241,1050],[0,0.707106781,0.707106781,
0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_2_1:=[[439,-788.5,950],[0,0.707106781,0.707106781,
0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_2_2:=[[439,-788.5,785],[0,0.707106781,0.707106781,
0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_2_3:=[[189,-788.5,1050],[0,0.707106781,
0.707106781,0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_3_1:=[[439,-1378.5,950],[0,0.707106781,
0.707106781,0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_3_2:=[[439,-1378.5,785],[0,0.707106781,
0.707106781,0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_3_3:=[[189,-1378.5,1050],[0,0.707106781,
0.707106781,0],[-1,0,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_3_4:=[[189,-851.168,1050],[0,0.707106781,
0.707106781,0],[-2,0,-4,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR speeddata Vel_PICK:=[206.15,500,5000,1000];
VAR num TRAYECTORIA_CARGADA;

!*****
! Módulo: Module1
!
! Descripción:
! Módulo Responsable de las operaciones de Pick and Place
!
! Autor: J . Albarranch
!
!*****

PROC main()

!EJECUCION MIENTRAS ESTA EN MARCHA (INCLUIDOS SENSORES DE
SEGURIDAD)
  WHILE MARCHA=1 DO
    IPOSICION INICIAL
    MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;

    !COMPROBACION DE TRAYECTORIA A LA ESPERA
    IF TRAYECTORIA_IN>0 THEN
```

```

!CARGA DE TRAYECTORIA
TRAYECTORIA_CARGADA:=TRAYECTORIA_IN;
!REESTRABLECIMIENTO
SetAO TRAYECTORIA_OUT,0;
!ORDEN DE ACTUALIZAR VECTOR
SetDO ACTUALIZAR_OUT,1;

ENDIF

TEST TRAYECTORIA_CARGADA
CASE 1:
  Path_10;
CASE 2:
  Path_20;
CASE 3:
  Path_30;

ENDTEST

SetDO PLACE,0;  !REESTABLECER PLACE

ENDWHILE

ENDPROC

  PROC Path_10()
    WaitDI PICK,1;  !ESPERAR SEÑAL PARA EL PICK
    MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_1_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_1_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    SetDO PLACE,1;  !SOLTAR PIEZA
    MoveL Target_1_3,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
  ENDPROC
  PROC Path_20()
    WaitDI PICK,1;  !ESPERAR SEÑAL PARA EL PICK
    MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_2_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_2_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    SetDO PLACE,1;  !SOLTAR PIEZA
    MoveL Target_2_3,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
  ENDPROC
  PROC Path_30()
    WaitDI PICK,1;  !ESPERAR SEÑAL PARA EL PICK
    MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_0_2,Vel_PICK,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_0_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_3_1,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_3_2,v50,fine,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
    MoveL Target_3_3,v1000,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;

```

```
SetDO PLACE,1;  !SOLTAR PIEZA
MoveL Target_3_4,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
MoveL Target_0_1,vmax,z100,VENTOSA_6x40x80\WObj:=WO_PICKPLACE;
  ENDPROC
ENDMODULE
```



