

Universidad Miguel Hernández de Elche

**MASTER UNIVERSITARIO EN
ROBÓTICA**



**“Programación de robot ABB para tareas de
reparación de defectos en carrocerías de
automóvil”**

Trabajo de Fin de Máster

2016-2017

**Autor: Cristian Querol Gómez
Tutor: Carlos Pérez Vidal**

AGRADECIMIENTOS

Siempre se recuerda a todos los que de alguna manera han colaborado técnica o moralmente de forma desinteresada, sin ella no hubiera sido posible el desarrollo del presente trabajo.

En primer lugar, dar las gracias a mi director de proyecto Carlos Pérez Vidal, por su implicación en el desarrollo del presente trabajo.

Agradecer a la Universidad Miguel Hernández y al Departamento de Ingeniería de Sistemas y Automática su colaboración y poner a mi disposición los programas de software utilizados, así como sus licencias y un servidor para la utilización de la licencia del software RobotStudio.

Por último, evidentemente agradecer a mi familia y a mis compañeros, que con su apoyo incondicional han estado en todo momento a mi lado.



RESUMEN

El presente trabajo trata de llevar a cabo el diseño de una metodología predictiva de precisión y pulido superficial aplicado al mecanizado robotizado como alternativa viable de calidad al pulido manual o auxiliado con pulidora eléctrica convencional. Para ello, desarrollaremos dos estaciones prototipo funcionales basadas en robótica e inteligencia artificial que permitan pronosticar el resultado de los procesos en una línea de fabricación de vehículos automóviles de la marca Volkswagen.

Lo primero de todo se realiza una revisión bibliográfica del estado del arte actual respecto a los robots en la industria de automoción, la utilización de herramientas de pulido sobre superficies y la implantación y auge de los robots colaborativos en la industria 4.0.

A continuación, fue necesario la recopilación de la información necesaria y un periodo de aprendizaje a través de video-tutoriales que permita adquirir los conocimientos necesarios para el manejo del programa RobotStudio 6.05.

Una vez adquiridos estos conocimientos se implantarán dos estaciones virtuales robotizadas de pulido. Una pulimentará los defectos propios de las capas internas o imperfecciones de la superficie del acabado final del pintado de las carrocerías de vehículos automóviles ayudado por un puente móvil para trasladar el robot a las carrocerías en movimiento, y la otra estación pulimentará los bordes de la estructura metálica de los salpicaderos trasladados por cinta transportadora mediante robot suspendido sobre pórtico fijo.

Se especificarán los elementos operativos de cada estación y su disposición en sus áreas de trabajo: elección de robots, herramientas de pulido, controladores, puente y pórtico de sujeción, sistemas de transporte y soportes de las piezas a pulir, medidas de seguridad de operarios e intrusos, programas informáticos adecuados y PCs, necesarios para el pulido automatizado de la línea de fabricación de vehículos automóviles.

Una vez especificado e integrado del sistema nos centraremos en el testado del sistema y se realizarán una serie de simulaciones de las respectivas estaciones, con análisis y evaluación de los resultados obtenidos para comprobar sobre pantalla su buen funcionamiento.

Palabras clave: ABB, carrocerías de automóviles, controlador, estación robotizada, proceso industrial, programación RAPID, pulido superficies, robot, RobotStudio, simulación robótica, salpicaderos.

ABSTRACT

The present work tries to carry out the design of a predictive methodology of precision and superficial polishing applied to the robotized machining as a viable alternative of quality to the manual polishing or helped with conventional electrical polishing machine. For this, we will develop two functional prototype stations based on robotics and artificial intelligence that allow to predict the result of the processes in a line of manufacture of vehicles of the Volkswagen brand.

First of all, we aim to present a bibliographical review of the state of the art in relation to robots in the automotive industry, the use of polishing tools on surfaces and the implementation and boom of collaborative robots in the industry 4.0.

Next, it is necessary to collect the most relevant information. Furthermore, a period of learning based on several video-tutorials will allow us to get the knowledge for the suitable management of the RobotStudio 6.05 programme.

Once acquired this knowledge two robotic virtual polishing stations will be implemented. One will polish the internal layer defects or surface imperfections of the final paint finish of motor vehicle bodies assisted by a mobile bridge to move the robot to the moving bodies, whereas the other station will polish the border of the metallic structure of the dashboards transported by conveyor belt by a robot suspended on a fixed portico.

The operational elements of each station and their layout in their workspace will be specified: choice of robots, polishing tools, controllers, bridge and fixed portico, transportation systems and supports of the parts to be polished, security measures of workers and intruders, appropriate software and PCs, necessary for the automatized polishing of the automobile manufacturing line.

Once specified and integrated the system will focus on the testing of the system and a series of simulations of the respective stations will be carried out, with analysis and evaluation of the results obtained to check on display its good operation.

Keywords: ABB, car bodies, controller, robotic station, industrial process, RAPID programming, polishing surfaces, robot, RobotStudio, robotic simulation, dashboard.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	I
RESUMEN.....	II
ABSTRACT	III
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS.....	X
1. INTRODUCCIÓN.....	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Resumen del Trabajo	3
1.4. Estructura de la Memoria del Tfm.....	7
2. ESTADO DEL ARTE	9
2.1. Introducción.....	9
2.2. Sistemas de Pulido de Superficies Mediante Robot.....	10
2.3. Aplicaciones de la Robótica Colaborativa Hombre-Robot.....	11
3. METODOLOGÍA	15
3.1 Introducción al Programa RobotStudio.....	15
3.1.1 Creación de una nueva estación.....	15
3.1.2 Creación de la herramienta de un robot.....	18
3.1.3 Creación del controlador de un robot.....	24
3.1.4 Creación de planos de trabajo y posiciones.....	25
3.1.5 Creación de trayectorias	29
3.1.6 Introducción a RAPID.....	36
3.2. Diseño y Programación de una Estación.....	40
3.2.1 Diseño en Inventor de la geometría de la estación	40
3.2.2 Creación y diseño de los Smart Components	42
3.2.3 Creación de señales E/S.....	47
3.2.4 Lógica de estación	49
3.3 Simulación de una Estación	50
4. IMPLEMENTACIÓN A UN PROCESO DE PULIDO EN UNA LÍNEA DE PRODUCCIÓN DE AUTOMÓVILES	53
4.1. Estación de Carrocerías	53
4.1.1 Elección del Robot	53
4.1.2 Elección de la Herramienta	54
4.1.3 Diseño de los Sólidos de la Estación	54
4.1.4 Creación de la Estación en RobotStudio.....	62
4.1.4.1 SC Carrocerías	62

4.1.4.2 SC Puente	63
4.1.4.3 Controlador y Lógica de Estación	64
4.1.4.4 Programación en RAPID.....	65
4.2. Estación de Salpicaderos	68
4.2.1 Elección del Robot	68
4.2.2 Elección de la Herramienta	69
4.2.3 Diseño de los Sólidos de la Estación	69
4.2.4 Creación de la Estación en RobotStudio.....	77
4.2.4.1 SC Salpicaderos	77
4.2.4.2 SC Cintas.....	78
4.2.4.3 Controlador y Lógica de Estación	79
4.2.4.4 Programación en RAPID.....	79
5. RESULTADOS	83
5.1 Simulación de la Estación de Carrocerías.....	83
5.2 Simulación de la Estación de Salpicaderos.....	91
6. CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO.....	97
6.1. Conclusiones.....	97
6.2. Líneas futuras de trabajo.....	98
7. BIBLIOGRAFÍA.....	99
8. ANEXOS.....	105
8.1 Planos	105
8.1.1 Planos de la Estación de Carrocerías	105
8.1.2 Planos de la Estación de Salpicaderos	111
8.2 Señales	114
8.2.1 Señales de la Estación de Carrocerías	114
8.2.2 Señales de la Estación de Salpicaderos	114
8.3 Programas.....	114
8.3.1 Código de RAPID de la Estación de Ejemplo.....	114
8.3.2 Código de RAPID de la Estación de Carrocerías.....	115
8.3.3 Código de RAPID de la Estación de Salpicaderos.....	130
8.4. Hojas de Características de los Robots.....	142
8.4.1 Hoja de Característica del Robot IRB140.....	142
8.4.2 Hoja de Característica del Robot IRB4600.....	145
8.4.3 Hoja de Característica del Robot IRB1600.....	148

ÍNDICE DE FIGURAS

Figura 2.1. Diferencia entre pulido robot UR10 y pulido manual	11
Figura 2.2. Diferencia entre robots colaborativos y no colaborativos	12
Figura 3.1. Creación nueva estación	15
Figura 3.2. Pestañas programa RobotStudio.....	16
Figura 3.3. Robots de la Biblioteca ABB.....	17
Figura 3.4. Robot ABB IRB 140	18
Figura 3.5. Creación de Sólidos	18
Figura 3.6. Creación de un Cono.....	19
Figura 3.7. Crear herramienta	19
Figura 3.8. Configuración herramienta	20
Figura 3.9. Conexión de la herramienta al robot.....	20
Figura 3.10. Herramienta conectada al robot	21
Figura 3.11. Importar geometría	21
Figura 3.12. Biblioteca RobotStudio	22
Figura 3.13. Herramienta de la biblioteca	23
Figura 3.14. Herramienta de la biblioteca conectada al robot.....	23
Figura 3.15. Creación de un controlador desde diseño	24
Figura 3.16. Creación del controlador.....	24
Figura 3.17. Opciones del sistema	25
Figura 3.18. Creación de un tetraedro	25
Figura 3.19. Crear un objeto de trabajo	26
Figura 3.20. Posición objeto de trabajo	26
Figura 3.21. Crear objetivo	27
Figura 3.22. Creación puntos objetivo	28
Figura 3.23. Creación de la posición inicial del robot.....	29
Figura 3.24. Ver herramienta en la posición	30
Figura 3.25. Girar posición de la herramienta.....	30
Figura 3.26. Giros de la posición respecto al eje X y al eje Z	31
Figura 3.27. Ver robot en posición.....	31
Figura 3.28. Creación de una trayectoria vacía	32
Figura 3.29. Trayectoria Path_10	33
Figura 3.30. Copiar y aplicar orientación en los demás objetivos.....	34
Figura 3.31. Configuración automática de la trayectoria.....	34
Figura 3.32. Instrucciones de los puntos de la trayectoria.....	35

Figura 3.33. Modificar instrucciones	36
Figura 3.34. Sincronización con RAPID.....	37
Figura 3.35. Sincronización de datos, objeto de trabajo y trayectorias.....	37
Figura 3.36. Programa RAPID	38
Figura 3.37. Ejemplo de un punto en RAPID.....	38
Figura 3.38. Ejemplo de instrucción de movimiento MoveL en RAPID.....	39
Figura 3.39. Mejora de la precisión en la trayectoria	40
Figura 3.40. Cambios en RAPID	40
Figura 3.41. Exportar sólido desde Inventor	41
Figura 3.42. Importar geometría desde RobotStudio.....	42
Figura 3.43. Crear componente inteligente (SC)	42
Figura 3.44. Categorías SC	43
Figura 3.45. Señales y propiedades en SC	43
Figura 3.46. Primitivos paramétricos en SC	44
Figura 3.47. Sensores en SC	44
Figura 3.48. Acciones en SC	45
Figura 3.49. Manipuladores en SC	45
Figura 3.50. Otros en SC.....	46
Figura 3.51. Señales y conexiones del SC	46
Figura 3.52. Diseño del SC.....	47
Figura 3.53. Añadir señales.....	47
Figura 3.54. Añadir una señal de un controlador.....	48
Figura 3.55. Reinicio del controlador	48
Figura 3.56. Configuración de una señal.....	49
Figura 3.57. Lógica de una estación.....	50
Figura 3.58. Compilación de un programa RAPID e inicio de la simulación	50
Figura 3.59. Configuración de la simulación, ajustes de controlador.....	51
Figura 3.60. Configuración de la simulación, ajustes del programa	51
Figura 3.61. Simulación y grabación.....	52
Figura 4.1. Robot ABB IRB 4600.....	53
Figura 4.2. Herramienta de pulido	54
Figura 4.3. Puente móvil.....	54
Figura 4.4. Herramienta pulidora	55
Figura 4.5. Bancada carrocerías	55
Figura 4.6. Raíles	56
Figura 4.7. Carrocería Volkswagen modelo Golf	56
Figura 4.8. ISO Plate Large	57

Figura 4.9. Socket IRB4600.....	57
Figura 4.10. Vallas.....	58
Figura 4.11. Controlador.....	58
Figura 4.12. PLC.....	59
Figura 4.13. Mesa.....	59
Figura 4.14. Silla.....	60
Figura 4.15. Ordenador	60
Figura 4.16. Carretilla elevadora	61
Figura 4.17. Operario.....	61
Figura 4.18. SC Carrocerías.....	63
Figura 4.19. SC Puente	64
Figura 4.20. Lógica de Estación Carrocerías.....	65
Figura 4.21. Robot ABB IRB 1600.....	68
Figura 4.22. Herramienta de pulido	69
Figura 4.23. Pórtico	69
Figura 4.24. Herramienta pulidora	70
Figura 4.25. Bandeja salpicadero	70
Figura 4.26. Salpicadero de fabricante Faurecia.....	71
Figura 4.27. Cinta transportadora.....	71
Figura 4.28. ISO Plate Large.....	72
Figura 4.29. Grundplate IRB1600.....	72
Figura 4.30. Vallas.....	73
Figura 4.31. Controlador.....	73
Figura 4.32. Mesa.....	74
Figura 4.33. Silla.....	74
Figura 4.34. Ordenador	75
Figura 4.35: Palé	75
Figura 4.36. Carretilla elevadora	76
Figura 4.37. Operario.....	76
Figura 4.38. SC Salpicaderos.....	78
Figura 4.39. SC Cintas	79
Figura 4.40. Lógica de Estación Salpicaderos.....	79
Figura 5.1. Estación Carrocerías	83
Figura 5.2. Primera bancada con carrocería	83
Figura 5.3. Bancada con carrocería en movimiento	84
Figura 5.4. Segunda carrocería	84
Figura 5.5. Sensor detector objetos.....	85

Figura 5.6. El sensor detecta la bancada	85
Figura 5.7. Robot en movimiento.....	86
Figura 5.8. Pulido en lateral y capó de la carrocería.....	86
Figura 5.9. Pulido en el techo de la carrocería	87
Figura 5.10. Detalle de la herramienta del robot puliendo	87
Figura 5.11. Robot en reposo tras pulido.....	88
Figura 5.12. Puente móvil regresando a su posición	88
Figura 5.13. Carrocería tras finalizar el pulido	89
Figura 5.14. Sincronización puente móvil con la segunda carrocería.....	89
Figura 5.15. Carrocerías finalizado el pulido	90
Figura 5.16. Sensor al final de los carriles.....	90
Figura 5.17. Detección bancada	91
Figura 5.18. Estación Salpicaderos	91
Figura 5.19. Primera bandeja con salpicadero	92
Figura 5.20. Segunda bandeja sobre la cinta transportadora	92
Figura 5.21. Salpicaderos sobre la cinta transportadora	92
Figura 5.22. Sensor detecta objetos	93
Figura 5.23. Parada de la cinta.....	93
Figura 5.24. Robot en marcha	93
Figura 5.25. Robot realizando el pulido en salpicadero	94
Figura 5.26. Robot realizando el pulido en parte posterior salpicadero	94
Figura 5.27. Detalle de la herramienta de pulido	94
Figura 5.28. Robot en su posición inicial	95
Figura 5.29. Cintas transportadoras en marcha de nuevo.....	95
Figura 5.30. Proceso de pulido en salpicaderos.....	95
Figura 5.31. Sensor final cintas transportadoras	96
Figura 5.32. Pale con bandejas	96

ÍNDICE DE TABLAS

Tabla 4.1. Modelos de robot IRB4600	53
Tabla 4.2. Modelos de robot IRB1600	68
Tabla 8.1. Señales E/S del controlador robot IRB4600	114
Tabla 8.2: Señales E/S del controlador robot IRB1600	114





1. INTRODUCCIÓN

1.1. Motivación

El pulido se emplea para eliminar imperfecciones superficiales, tales como marcas dejadas por herramientas o asperezas e imperfecciones de acabado dejando la superficie suave y brillante mediante un material abrasivo.

En la actualidad la mayoría de las operaciones de pulido de las partes metálicas en el sector industrial se llevan a cabo mediante piezas movidas a mano o herramientas con movimiento mecánico y platos de sujeción.

Las herramientas más comunes son las pulidoras de disco. Los discos de pulido se componen de una base flexible de papel o tela a la cual se une un material microabrasivo ultrafino por medio de un aglutinante natural o sintético formando un cuerpo rígido, generalmente poroso.

Este proceso en metales de alta porosidad y dureza superficial tanto manual o auxiliado de alguna herramienta está realizado por operarios altamente especializados, con características de pesadez y reiteración, con un alto consumo de herramientas, elevados tiempos y costes de ejecución. Otra consideración a tener en cuenta sobre el pulido es el perjuicio para la salud, ya que el metal utilizado libera polvo durante el pulido e irrita las membranas mucosas donde el operario está siempre obligado a trabajar con mascarilla respiratoria haciendo necesario la implementación de cabinas con extracción eficaz del polvo o realizado directamente por la herramienta, existiendo además riesgo de cortes y otros peligros para el operario.

Estos defectos apenas visibles por el ojo humano son fuente de numerosos errores y su baja calidad obtenible no cumple con los requisitos actuales en cuanto a precisión y reproductibilidad.

La principal alternativa al método manual o con herramientas manuales, es el mecanizado convencional con máquina-herramienta muy especializada y de grandes dimensiones, aplicadas a materiales blandos fácilmente mecanizables (espumas de poliuretano, resinas, aleaciones de aluminio, madera, granito, etc.).

La principal diferencia del mecanizado entre estos materiales y los aceros es su dureza, aunque es posible desarrollar sistemas más versátiles y económicos a través de la utilización del mecanizado robotizado.

Las estaciones mecanizadas mediante robot antropomórfico representan una alternativa viable para el pulido, combinando dos tecnologías desarrolladas independientemente; por un lado el desarrollo de los robots y por otro el mecanizado clásico de máquina-herramienta.

La robótica ha ocupado un importante lugar en el sector automovilístico, formando un área tecnológicamente indivisible, de hecho este sector es el primer consumidor de robots y de sistemas de automatización. A su vez, es el mejor ejemplo de sustitución de trabajos penosos, duros o en entornos peligrosos.

En las líneas de fabricación de vehículos automóviles una de las mayores preocupaciones actuales es la calidad de sus productos. Para tender al concepto de "cero defectos" se aumentan los controles de calidad resolviendo de inmediato cualquier problema en la cadena de producción.

La aplicación de pinturas de acabado para carrocerías está cada vez más robotizadas, desarrollando estos procesos muy rápidos. En ellos se pueden generar una serie de defectos como motas de polvo u otros propios de las capas internas que se producen en el barniz o en el acabado final del pintado, sea por un pulido incompleto, deficiente o por las condiciones de aplicación.

Estas posibles imperfecciones en la pintura, que difícilmente son perceptible por el ojo humano, con el desarrollo de un sistema laser 3D son identificadas a la salida del horno de curado y con un adecuado pulido estos defectos superficiales, sin necesidad de un repintado, se pueden minimizar o eliminar totalmente.

Por todo ello, hay razones de peso para implantar un proceso robotizado de pulido de los defectos superficiales de las partes metálicas de la pintura de las carrocerías de vehículos automóviles y de los bordes metálicos de los salpicaderos; con ello se pretende mejorar la calidad en este tipo de procesos de acabado y por consiguiente aumenta la productividad.

1.2. Objetivos

El presente trabajo tiene como objetivo principal el diseño de una metodología predictiva de precisión y acabado superficial aplicado al mecanizado robotizado. Para ello, desarrollaremos dos prototipos funcionales inteligentes que permitan pronosticar el resultado de los procesos, consistentes en dos estaciones basadas en robótica e inteligencia artificial en una línea de fabricación de vehículos automóviles de la marca Volkswagen mediante el desarrollo teórico informático integrado.

La primera de las estaciones pulimentará las imperfecciones, aspereza y defectos de las estructuras de carrocerías tras su pintado y la segunda pulimentará los bordes metálicos de los salpicaderos.

Esta solución innovadora será el método alternativo al pulido manual o auxiliado con pulidora manual eléctrica convencional. Para esta implantación en el sistema establecido se deben evitar las barreras técnicas de los nuevos componentes.

Estos procesos de pulido exigen un detallado estudio previo de los defectos más comunes encontrados en las carrocerías y de las técnicas y materiales más adecuados de pulido.

Del planteamiento realizado se extrae que estas mejoras son posibles mediante la adecuada selección de robots y herramientas de pulido que encaje con los elementos terminales del brazo del robot.

La desviación y vibración de las herramientas durante la trayectoria de trabajo dependen fundamentalmente de la rigidez del robot, la magnitud de las fuerzas de pulido y la dirección de la acción predominante. La interfaz para estos sistemas deberá dar precisión y rigidez, incluyendo la generación automática de trayectorias, así como interfaces hápticas de realimentación de velocidades y fuerzas, con ello no se deformarán los elementos a pulir.

Además, cumplirá con los condicionamientos de métodos y tiempos acoplada a la línea automatizada convencional existente de producción de los vehículos automóviles.

Siempre será preciso estar dispuesto a admitir cambios en el desarrollo del proceso proyectado, modificando el diseño de herramientas, programas o sistemas de ejecución que faciliten y hagan viable las tareas encomendadas al robot.

La implantación de los sistemas robotizado de pulido que se propone desarrollar mejorará estratégicamente la competitividad de esta empresa de automóviles, obteniendo grandes beneficios, entre los que se destacan:

- **Flexibilidad:** Los sistemas robotizados son flexibles, es decir, se adaptan a una serie de tareas sin la necesidad de modificar o parar la cadena de producción.
- **Integración de calidad y producción:** La repetitividad y el control del proceso encomendado al robot asegura un acabado homogéneo y de alta calidad del producto final. Esta calidad se puede detectar en origen, beneficio que se trasladará a cliente.
- **Reducción en los costes de producción:** La robotización de los procesos de pulido supondrá una reducción significativa del tiempo invertido respecto al proceso manual de ejecución, con la consecuencia lógica del aumento de la productividad y por tanto, la reducción de costes de dichos procesos. Repercutirá en el coste final del vehículo, por tanto su competitividad y posicionamiento de la empresa en el mercado.
- **Reducción del “time-to-market”:** Mediante la aceleración de estos procesos se reducirá el tiempo global del proyecto, con lo que se podrán sacar al mercado los automóviles antes, tendencia actual del mercado.
- **Seguridad laboral:** Los robots pasan a encargarse de actividades que pueden suponer una tarea desagradable y perjudicial para la salud de los operarios, lo que disminuye los accidentes laborales, empleándose estos en tareas de supervisión y mantenimiento.

1.3. Resumen del Trabajo

Desde que se plantea la conveniencia de diseñar un prototipo funcional e implementar un sistema inteligente basados en la robótica e inteligencia artificial aplicable al pulido de asperezas o imperfecciones de acabado en una línea de producción de una fábrica de vehículos automóviles de la marca Volkswagen, hasta que se ejecuta tal decisión, hay que utilizar una metodología para el desarrollo del sistema, esta metodología se realizó mediante fases, tomando una serie de decisiones para el correcto progreso de este, las cuales determinaron la ejecución del trabajo.

Hay que desarrollar dos estaciones robotizadas que permitan pronosticar el resultado del proceso, una que pulimente los defectos propios de las capas internas o imperfecciones de la superficie del acabado final del pintado de las carrocerías de vehículos automóviles, y otra que pulimente los bordes de la estructura metálica de los salpicaderos, previo análisis tridimensional mediante láser de los defectos detectados.

A continuación nombraremos y explicaremos las fases que hemos seguido en este trabajo de investigación:

Planeamiento del Sistema: Se inició con la fase de planeamiento, fase primordial en la que se definió lo requisitos más importante para la elaboración del sistema, visitando el Departamento de Ingeniería de Sistemas y Automática.

Se efectuó una visita de reconocimiento al taller de Centro-chapa del Grupo Marcos de Orihuela, en el que se observó la técnica manual del pulido de defectos una vez que el vehículo sale del horno de secado tras el pintado.

La línea de producción posee un pórtico con un sistema láser 3D, que escanea toda la superficie de las carrocerías, construyendo un mapa tridimensional que incluye las pequeñas imperfecciones, normalmente no visibles al ojo humano. En base a la información recopilada, y comparando con los datos virtuales del CAD almacenado, captura con precisión una nueva geometría en tres dimensiones, real e individual para cada carrocería que determina las coordenadas de puntos o zonas que debe ser pulimentada.

Hecha una composición de lugar, se decidió actuar sobre las principales dificultades que presentaban la implantación de un sistema robótico de pulido para mejorar las condiciones de la línea de fabricación de vehículos automóviles.

Así, pues se establecieron las directrices de las siguientes fases bajo la coordinación del Departamento de Ingeniería de Sistemas y Automática de la Universidad Miguel Hernández de Elche (Alicante).

Especificación del Sistema: Las variables recogidas por los PCs deben ser transmitidas a los robots para ejecutar las órdenes de pulido en las coordenadas asignadas de las carrocerías con imperfecciones y de los bordes de los salpicaderos.

Diseño del Sistema: Se especifican los elementos operativos de la estación y su disposición en la planta: robots, herramientas de pulido, controladores, puente y pórtico de sujeción de los robots, los sistemas de transporte y soportes de las piezas a pulir, la adecuación del área de trabajo para seguridad de operarios e intrusos, programas informáticos adecuados y PCs, necesarios para el pulido automatizado de la línea de fabricación de vehículos automóviles.

Para la estación robotizada de pulido de imperfecciones de pintura se optó por los siguientes componentes:

Robot antropomórfico de 6 ejes marca ABB modelo IRB 4600, con capacidad de peso de herramienta de 20 kg. y alcance de 2,50 m.

Herramienta pulidora con acoplamiento a robot mediante plato, arquitectura tipo tambor con actuación radial, disco compuesto por tejido resistente a la tracción y desgaste por rozamiento, con abrasivo fino de óxido de aluminio fijado al disco por medio de un aglutinante.

Controlador marca ABB modelo IRB-4600.

Estructura de suspensión y anclaje de robot compuesto por perfiles metálicos de acero laminado tipo puente móvil accionado por motor eléctrico sobre ruedas y guiado mediante raíles sujetos al suelo de la planta. Esta base móvil es capaz de

desplazar al robot dentro del área de trabajo en caso de ciclos de procesamientos largos, maximizando el tiempo, no originando tiempos muertos.

Barandillas de cerramiento y puertas para adecuación del espacio anti intrusos y seguridad de operarios.

Bancada móvil accionada por motor eléctrico sobre ruedas y guiado por raíles para transporte de las carrocerías de los vehículos automóviles.

Programas informáticos:

- ABB - RobotStudio Machining PowerPac 6.05: Software CAD/CAM de generación de trayectorias que incluye un control de fuerza y velocidad durante el mecanizado del pulido.
- Autodesk Inventor Profesional 2015.

La visualización de los resultados se realizó mediante PC portátil de la marca Lenovo, modelo G50, con procesador Intel Core i7-5500U, CPU a 2,40 GHz y 4,0 Gb de memoria RAM y sistema operativo Windows 10.

Para la estación robotizada de pulido de los bordes de los salpicaderos se optó por los siguientes componentes:

Robot marca ABB modelo IRB 1600, con capacidad de peso de herramienta de 6 kg. y alcance de 1,45 m.

Herramienta pulidora con acoplamiento a robot mediante plato, arquitectura tipo tambor con actuación radial, disco compuesto por tejido resistente a la tracción y desgaste por rozamiento, con abrasivo fino de óxido de aluminio fijado al disco por medio de un aglutinante.

Controlador marca ABB modelo IRB-1600.

Estructura de suspensión y anclaje compuesto por cuatro perfiles metálicos de acero laminado tipo pórtico fijado mediante placas de anclaje y pernos con tornillos al suelo.

Transportadores de bandas con guías laterales y apoyo telescópico para la regulación de altura, sobre rodillos para transporte de los salpicaderos de los vehículos. Además cuenta con motor con variador de velocidad y caja eléctrica de marcha-paro.

Barandillas de cerramiento y puertas para adecuación del espacio anti intrusos y seguridad de operarios.

Bandejas de acero galvanizado para acople y transporte de salpicaderos.

Palets de madera para apilado de bandejas de salpicaderos una vez pulidos sus bordes.

Programas informáticos:

- ABB - RobotStudio Machining PowerPac 6.05: Software CAD/CAM de generación de trayectorias que incluye un control de fuerza y velocidad durante el mecanizado del pulido.

- Autodesk Inventor Profesional 2015.

La visualización de los resultados se realizó mediante PC portátil de la marca Lenovo, modelo G50, con procesador Intel Core i7-5500U, CPU a 2,40 GHz y 4,0 Gb de memoria RAM y sistema operativo Windows 10.

Desarrollo del Sistema: Se llevó a cabo la implantación del sistema de pulido en la línea de fabricación de automóviles, con dos estaciones, una para el pulido de las imperfecciones de las carrocerías de los vehículos automóviles a la salida del horno de pintado y otra para el pulido de los bordes metálicos de los salpicaderos.

Todos los componentes de las estaciones diseñadas se ubicaron en sus respectivos emplazamientos especificados y se instalaron los programas necesarios en ambos PCs. Además se ha establecido un cerramiento perimetral que abarque el espacio de movimiento de robot de tal forma que no quede ningún punto de acceso más que los habilitados para tal fin.

Integración del Sistema: A la salida del horno de secado tras el pintado final de carrocerías de vehículos automóviles se ha establecido un control de calidad para evitar posibles desviaciones de los valores deseados mediante un sistema láser 3D de escaneado. Este sistema escanea la superficie de toda carrocería que pasa bajo el mismo, construyendo un mapa tridimensional que incluye las pequeñas imperfecciones de la misma.

En base a la información recopilada, y comparando con los datos virtuales del CAD almacenado en el PC, captura con precisión una nueva geometría en tres dimensiones, real e individual para cada carrocería, que en caso de conformidad el vehículo seguirá su línea de transferencia a almacén.

Si la carrocería es rechazada, se determinará las coordenadas de los puntos o zonas que deben ser pulimentadas y se desviará a la estación para el pulimento de imperfecciones, en donde se calcula y transmite las fuerzas y orientación del vector de la herramienta en función de la normal a pulir. Las fotocélulas detectan la posición de la carrocería sobre la bancada e iniciará el robot sobre la superficie el proceso de pulido teniendo en cuenta la realidad morfológica del defecto.

Para la estación de pulido de los bordes metálicos del salpicadero se ha introducido en el programa RobotStudio las coordenadas de los bordes que se han de pulimentar. Una vez que las fotocélulas han detectado la presencia del salpicadero sobre el equipo de transporte de rodillos iniciará el proceso de pulimento y al finalizarlo el operador controlará el proceso y colocará el salpicadero con su bandeja en palé para ser retirado en carretilla por otro operario.

Validación del Sistema: En las numerosas pruebas de simulación con el programa RobotStudio se verifica la idoneidad de las estaciones implantadas. El sistema láser en 3D ha transmitido al PC los datos recogidos de las superficies escaneadas y comparadas con las almacenadas en CAD. Se verifica el cumplimiento de las órdenes transmitidas al robot y la coordinación de tiempos de toda la línea de producción, comprobando el buen funcionamiento de ambas estaciones; con ello se valida el sistema.

Simulación del Proceso: La recopilación de secuencias simuladas de los robots de ambas estaciones se grabó con el programa RobotStudio para ser reproducidas, para un posterior análisis.

Análisis de la Simulación: Este análisis sirvió como soporte para la toma de decisiones adecuadas, identificando y estudiando el establecimiento de estas alternativas para mejorar las condiciones de productividad y flexibilidad del pulido en ambas estaciones, con calidad uniforme, sistematización de los procesos y la posibilidad de supervisar y/o controlar las estaciones según los diferentes parámetros y criterios oportunos que permita realizar los cambios pertinentes a las circunstancias del momento, además se podrán abordar nuevas experiencias en este campo con un menor riesgo.

Como resultado se puede destacar la alta flexibilidad y el aumento de la productividad, la mejora de la calidad del producto, las condiciones de trabajo y de seguridad, sustituyendo estas tareas peligrosas de los operarios como ventajas principales de estos sistemas robotizados.

Redacción del Trabajo: Toda la información recopilada en la bibliografía consignada, las simulaciones por ordenador y los datos obtenidos en la investigación sirvieron como base para la redacción del presente trabajo.

1.4. Estructura de la Memoria del Tfm

La memoria del presente trabajo fin de master está distribuida en ocho capítulos, en los que se procede a explicar en detalle su ejecución, de los que daremos una breve explicación en las siguientes líneas:

Capítulo 1: Introducción. Se expone la motivación y el contexto del presente trabajo, los objetivos que persiguen, se resume y estructura la memoria para una mejor visión general de sus contenidos.

Capítulo 2: Estado del arte. Se ha analizado el estado del arte, con una revisión de las herramientas y métodos de pulido y de la tecnología actual aplicada en la industria, en especial, la de automoción.

Capítulo 3: Metodología. Se ha realiza una pequeña introducción al programa de software RobotStudio ABB versión 6.05, con aprendizaje y creación de una estación en un entorno de simulación. Se ha incorporado un robot y su controlador asociado, creando y definiendo los sólidos y herramienta de pulido, planos de trabajo, objetivos y trayectorias. Además se realiza una sincronización de los datos del programa, aprendiendo el lenguaje de programación textual RAPID, creación de componentes inteligentes (SC), señales E/S y la lógica de la estación. Finalmente se muestra como realizar una simulación de la estación y su grabación para reproducción posterior en PC.

Capítulo 4: Implementación a un Proceso de Pulido en una Línea de Producción de Automóviles. En cada una de las dos estaciones vamos a implementar el proceso de pulido. En primer lugar realizaremos la elección del robot y herramienta de cada estación, creación de los sólidos e implementación en la estación.

Capítulo 5: Resultados. Al finalizar las programaciones de las estaciones se han realizado una serie de simulaciones para observar en pantalla el funcionamiento del sistema automatizado creado y se han grabado para poderlas reproducir en cualquier momento. Además se realizará una interpretación y discusión de las simulaciones obtenidas.

Capítulo 6: Conclusiones y Líneas Futuras de Trabajos. Se exponen las conclusiones del proyecto, comparándolas con los objetivos marcados inicialmente. Se proponen mejoras o nuevas líneas de investigación para futuros trabajos.

Capítulo 7: Bibliografía. Se indican todas las referencias bibliográficas utilizadas en este trabajo, ordenadas con código para su cita en la memoria.

Capítulo 8: Anexos. En este apartado se añade información complementaria, incluyendo los planos de los objetos diseñados, señales de entrada/salida de cada estación, programas de cada estación en código RAPID y las hojas de características de cada robot utilizado.



2. ESTADO DEL ARTE

2.1. Introducción

En un entorno industrial tan cambiante, se hace patente la necesidad de cambios sustanciales para fabricar productos más versátiles y económicos que reporten beneficios. Estos productos de alto valor añadido se pueden conseguir a través de los novedosos sistemas flexibles tecnológicos de producción que actualmente se están introduciendo en el entorno industrial.

La robótica no es ajena a las diversas soluciones que se producen en este sector, por ello, en este apartado se trata de resumir el estado actual de desarrollo de las aplicaciones robóticas empleadas en los entornos industriales.

En los años sesenta, los ingenieros americanos Joe Engelberger y George Devol diseñaron el primer robot industrial útil, llamado Unimate, fue puesto al servicio de la empresa General Motors, a partir de aquí, se empezaron a introducir los primeros robots industriales, que aumentaron aún más la flexibilidad del sistema productivo. En poco más de treinta años las investigaciones y desarrollo robótico industrial ha permitido que los robots tome posiciones en casi todas las áreas productivas de todo tipo de industrias. [33]

A partir de 2010 se inicia el desarrollo de los robots de cuarta generación que prometen cambios importantes. Robots más inteligentes, más móviles y adaptables, con sensores sofisticados y control en tiempo real. Estos ayudarán cada vez más a los humanos en todas las áreas productivas y tipos de industrias. La inteligencia artificial está pasando rápidamente de los laboratorios universitarios a la aplicación práctica, existe prototipos que en un futuro podrá fabricar autopistas, construir estructuras de acero para edificios, limpiar conducciones subterráneas o cortar el césped.

La inversión tecnológica en el campo de la automatización industrial con la utilización de robots está produciendo cambios sustanciales en los sistemas de producción de la industria, extendiéndose a todo tipo de fábricas y empresas industriales, corriendo riesgos de quedarse estancados en el mercado en los procesos de fabricación si descartamos su utilización.

El principal papel de los robots mediante sus brazos articulados es ejercer diferentes funciones productivas; transporte, manejo de materiales, maquinado, carga y descarga, etc. mediante su capacidad para desempeñar diversas tareas u operaciones. El robot industrial ha sido descrito como el elemento más visible de la fabricación asistida por computador y como la base técnica para la mayor automatización de la producción.

Aunque en un principio se pensaba que el montaje automático podría ser completamente realizado por robots. Sin embargo, siempre hay necesidad de técnicos humanos. Además, el costo de algunos robots para hacer ensamblajes menores o control de calidad no son tan eficiente como la contratación de un ser humano en tales tareas.

2.2. Sistemas de Pulido de Superficies Mediante Robot

La industria del automóvil es un campo en donde los robots han evolucionado y han adquirido un grado de perfeccionamiento tecnológico en tareas repetitivas, monótonas y perjudiciales para los operarios acorde con las exigencias mecánicas actuales. Actualmente, el 50% de todos los robots en uso hoy en día se utilizan en la fabricación de automóviles.

Los robots aunque inicialmente eran empleados en realizar tareas precisas de soldadura, hoy en día son muy utilizados en la industria del automóvil como trabajadores incansables de la línea de montaje. A partir de sus muñecas existen múltiples efectores finales o herramienta, dependiendo de sus necesidades, que son las que realmente ejecutan el trabajo con precisión programada. Según las operaciones a ejecutar se distinguen: soldador de arco para fusionar piezas de metal, ventosas accionadas por vacío para poder manejar el vidrio liso de lunas del automóvil, fresadoras, pistolas de pintura o lijadoras y pulidoras de piezas metálicas entre otras, aunque las más comunes son las pinzas. Estos robots no solo le ahorran al empresario tiempo, si no también más seguridad para sus empleados, ya que ahora son las maquinas quienes hacen las labores más peligrosas.

El robot SAMMS es bastante flexible para poder ejecutar tareas de lavado, pintado, lijado, pulido y hasta reconocimientos de inspección. Para la limpieza de aviones, sustituye las celdas cilíndricas similares al lavacoches, por una herramienta con rodillos compactos contra rotativos de eje neumático. Este equipo le permitir al cabezal ligero de limpieza adaptarse a las formas curvadas de la superficie del avión, pudiendo llegar hasta las áreas de acceso más difícil, como alas y aletas. Un avión limpio ahorra hasta el 7% de combustible. [52]

El consorcio empresarial compuesto por: Emte SLU, Technologies for Advanced Manufacturing and Robotics, S.L. y la Universidad Politécnica de Cataluña como universidad pública, todos ellos con experiencia en el sector industrial, tienen por objetivo el diseño y desarrollo de una estación prototipo funcional para implementar a nivel industrial, un nuevo sistema óptico de reconocimiento y escaneo automatizado mediante tecnología 3D y un sistema robotizado dotado de una herramienta de pulido de grandes superficies acopladas apta para vagones de trenes y aviones, juntamente con el software necesario. [59]

En el artículo "Improving machining accuracy with robot deformation compensation" desarrollado por el centro de investigación ABB en 2009, se presenta un método práctico para compensar la deformación del robot causada por la fuerza de mecanizado sobre partes blandas metálicas, que era uno de los problemas del lijado y pulido.

El UR10 es un robot colaborativo fabricado por Universal Robots de Canadá al que se la ha asignado la tarea de pulir superficies planas. La implementación de este robot ante la escasez de operarios cualificados en la línea de fabricación de altavoces y subwoofers ha aumentado el rendimiento de la producción un 50% en la empresa Paradigm Electronics.

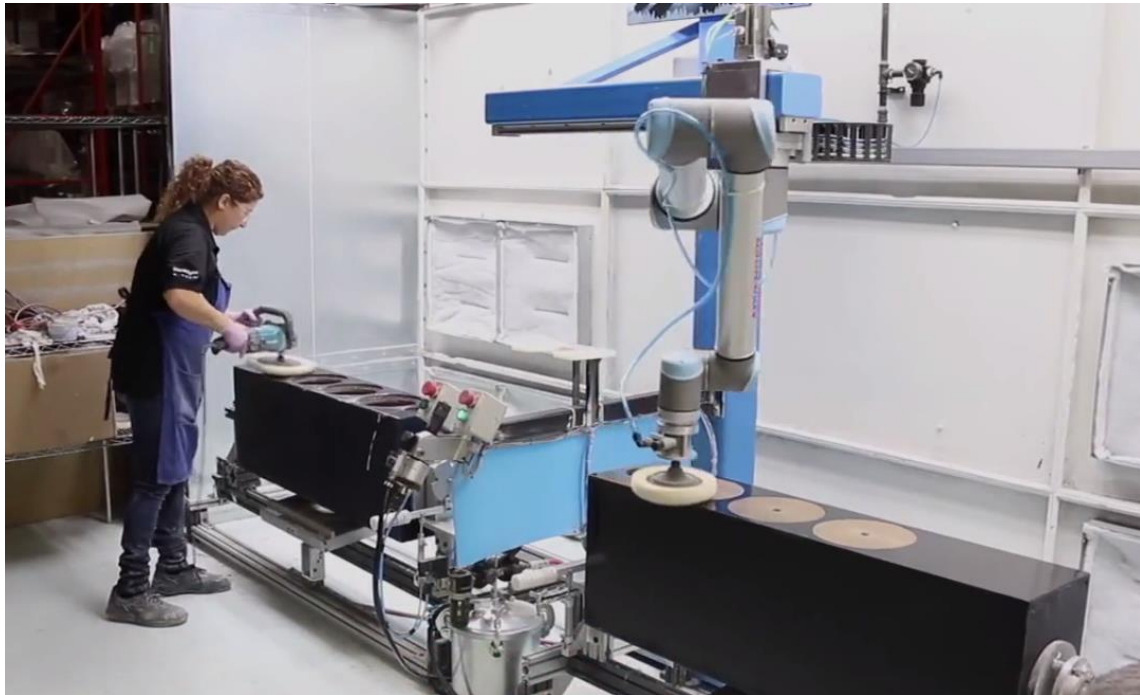


Figura 2.1: Diferencia entre pulido robot UR10 y pulido manual

Este éxito es debido a la incorporación de la retroalimentación de fuerzas dentro de su estructura; la utilización de esta fuerza regulable, permite aplicar una fuerza concreta en un punto específico del recorrido de pulido. En caso de aplicar demasiada fuerza, la superficie se calientan y los resultados serán totalmente contrarios a los esperados.

Como proyecto inmediato se va a introducir una nueva línea de altavoces con forma parabólica. Para resolver este nuevo hándicap y tras verificar su corto periodo de amortización de entre 12-14 meses, se incorporará un nuevo robot colaborativo que lije y pulimente las caras no planas de los altavoces. [61]

2.3. Aplicaciones de la Robótica Colaborativa Hombre-Robot

Los últimos avances en la tecnología robótica ha permitido el nacimiento de una nueva era en la automatización industrial: la de los robots colaborativos (cobot), tecnología en la que se basa la industria 4.0.

La filosofía de esta nueva generación de robots que aúnan esfuerzos con los humanos en entornos de fabricación, es hacer la robótica accesible a todo el mundo, y en particular facilitar su implementación en las PYMES para incrementar la competitividad.

Se caracterizan por estar constituidos por brazos robóticos ligeros, flexibles y fáciles de instalar que permiten ser montados y programados por personal no cualificado de manera sencilla y sin conocimiento de programación, aunque a veces no se habla de programación de robot, sino de entrenamiento moviendo manualmente el robot a las posiciones deseadas, posiciones que quedan registradas para que luego las repita el robot de forma automática.

Lejos de reemplazar como en anteriores generaciones de robots a trabajadores humanos, están diseñados especialmente para trabajar en su proximidad e interactuar directamente en los procesos productivos en un espacio de trabajo compartido.

La introducción de un robot colaborativo en un entorno de trabajo, no es percibido por el operario como una tecnología que le va a sustituir, sino como una herramienta de ayuda que le hace su trabajo más fácil, es como trabajar “con un compañero”, con posibilidades ilimitadas, incluyendo la integración laboral para personas con discapacidad.

En este entorno colaborativo, el operario aporta destreza, flexibilidad y la capacidad de resolver problemas y aunque los robots son, por defecto, más lentos y menos potentes que los industriales tradicionales, ofrecen fuerza, resistencia y precisión en la realización de la tarea en cuestión, sin restricciones de seguridad, pues los sensores, cámaras u otras tecnologías instaladas en robots colaborativos pueden sentir la cercanía de un humano y detener su movimiento para no ponerle en peligro.

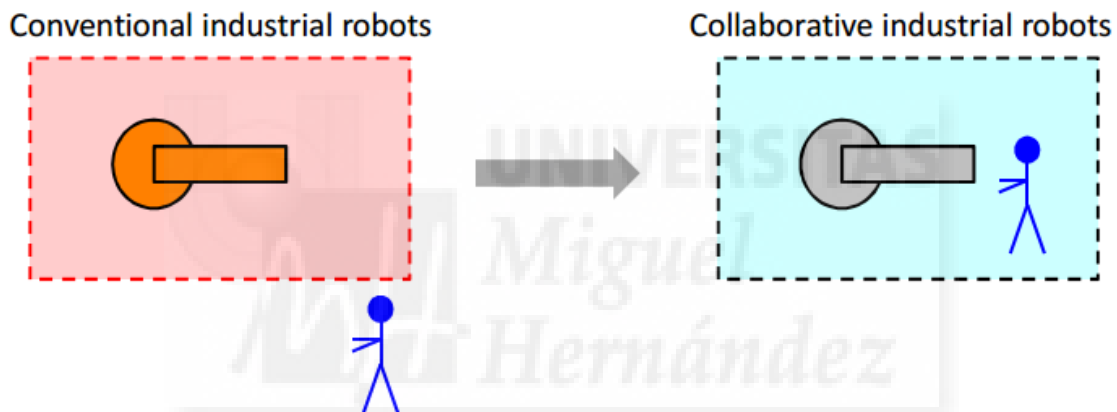


Figura 2.2: Diferencia entre robots colaborativos y no colaborativos

Los robots colaborativos se adaptan fácilmente a la automatización de procesos y son aptos para reconfigurarlos en la planificación de la producción. El precio asequible y el rápido retorno de su inversión los diferencian de los tradicionales y los hace idóneos para su implantación en las PYMES en las que, hasta ahora, no había sido viable.

La irrupción de estos nuevos Robots ha definido los requisitos de seguridad y aplicaciones colaborativas promulgando nuevas legislaciones como la ISO/TS 15066 – Safety of Collaborative. [53]

Esta colaboración en el futuro será de interacción con humanos.

Los principales beneficios de estos robots son:

- Los robots están capacitados para tareas de manejo simples y repetitivas.
- Los seres humanos, por otro lado, con sus habilidades cognitivas únicas comprenden y pueden adaptarse a cualquier cambio en la tarea.

- Se mejora el rendimiento con esta combinación de ser humano y robot, compartiendo óptimamente el trabajo.
- La colaboración entre ellos permite variar los niveles de automatización e intervención humana. Si una solución totalmente automatizada no es económica o demasiado compleja, se pueden automatizar parcialmente las tareas.
- Con la ayuda de estos robots se pueden mejorar mucho las estaciones de trabajo no ergonómicas.
- La seguridad de los seres humanos impera y es un requisito previo absoluto.

Las principales tareas que puede realizar el robot colaborativo son:

Pick & Place: En la paletización y despaletización se reduce la duración del ciclo.

Control de Calidad: Con la ayuda de una cámara de visión artificial los brazos robóticos pueden inspeccionar e identificar piezas defectuosas. Aumenta la calidad final del producto.

Ensamblaje: Ensamblaje por brazos robóticos de Plásticos, metales y otros materiales como la madera. Aumenta y mejora la velocidad y por tanto la calidad del proceso.

Paletizado y Empaquetado: Tareas realizadas por el robot con los niveles de calidad y exigencia más elevados.

Atornillado: Aporta precisión y velocidad exacta e idéntica. Mejora la calidad y la consistencia del producto final.

Pulido: Resultados constantes en el lijado y pulido de las superficies más irregulares.

Análisis de Laboratorio: El grado de objetividad aumenta y libera al equipo de las tareas más repetitivas.

Aplicaciones Médicas: cirugía mínimamente invasiva, cirugía de radiación en tratamiento de tumores, terapia de partículas, angiografías, hipoterapia neuronal y rehabilitación física.

Aunque en el mercado hay números fabricantes y robot colaborativos, a continuación se enumeran los principales.

YuMi es un robot colaborativo de ABB. Se caracteriza por tener dos brazos, fácil de utilizar. Está destinado a trabajos industriales como por ejemplo el montaje de pequeños componentes.

CR-35iA es el robot colaborativo de FANUC.

Roberta es el robot colaborativo de GOMETEC. Dispone de 6 ejes, es ágil y ligero y fácil de desplazar.

JACO del fabricante KINOVA, tiene 6 ejes y dispone, de una pinza de 3 dedos con movimientos independiente.

LBR iiwa es el robot sensitivo de KUKA, tiene 7 ejes y dispone de control de fuerza en cada uno de ellos.

NEXTAGE es el robot colaborativo de ROLLOMATIC. Tiene aspecto de humanoide, con sus dos brazos realiza trabajos tediosos. Con cuatro cámaras de video, dos en la cabeza y una en cada brazo que le ayuda a realizar con mayor seguridad las tareas. [54]

BAXTER es un robot industrial de dos brazos y cara animada fabricado por RETHINK ROBOTIC, realiza una amplia gama de tareas repetitivas de producción con precisión, envasado, elevación, clasificación, manipulado, carga y descarga, máquina-herramienta. Los propios trabajadores de la línea pueden entrenar a Baxter en cuestión de minutos, sin experiencia en software, robótica o la ingeniería requeridos.

Con la evolución de la robotización, su uso se extiende cada vez más hacia sectores como la automoción y la electrónica, actualmente considerados como los principales impulsores de su crecimiento.

La industria automovilística europea es una de las más automatizadas a nivel mundial, a pesar de ello deben mejorar su eficacia y productividad en sus cadenas de montaje, enfrentarse a la creciente competencia de sus rivales asiáticos y al envejecimiento de sus plantillas, además, los fabricantes de coches con modelos Premium de altos niveles de personalización en volúmenes relativamente bajos, crean una gran complejidad que requiere habilidades de destreza y resolución de problemas, algo que los robots hasta ahora no son capaces de hacer.

Por ello las fábricas de automóviles dependen de los trabajadores humanos, ya que crear robots con destreza humana para resolver estos problemas sería demasiado costoso e ineficiente.

Para paliar estas tendencias y para hacer frente a los volúmenes de producción de automóviles cada vez mayor; de la mano de la Cuarta Revolución Industrial (Industria 4.0) la industria de la automoción está introduciendo robots colaborativos y estaciones de trabajo optimizadas.

La aplicación en la industria del automóvil representó la mayor parte del mercado de robots colaborativos en 2015 esperando que esta tendencia continúe.

3. METODOLOGÍA

3.1 Introducción al Programa RobotStudio

El programa Robotstudio ABB es un software que permite crear, programar y simular estaciones de robots industriales, diseñado y patentado por la empresa ABB, líder en este tipo de robot. Utiliza el lenguaje RAPID.

RobotStudio posee un controlador virtual, una copia exacta del software real que emplean los robots en la producción. Ello permite programar un robot fuera de línea (offline) en un PC sin necesidad de parar la producción en la industria, exportando los resultados obtenidos en simulación a la estación real.

El programa además proporciona 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.

Para la creación y programación de las estaciones de este trabajo se ha utilizado la versión 6.05.

3.1.1 Creación de una nueva estación

Tras ser instalado el programa RobotStudio en el ordenador, es necesaria tener una licencia o bien utilizar los 30 días de prueba que tiene el producto de ABB. En nuestro caso hemos utilizado una licencia de red proporcionada por la Universidad Miguel Hernández de Elche, conectada mediante un servidor de licencias de red.

Una vez instalada la licencia y ejecutado el programa, lo primero que hay que realizar es crear una nueva estación en “Nuevo” y “Estaciones”, tenemos varias opciones, tal y como aparece en la figura 3.1.

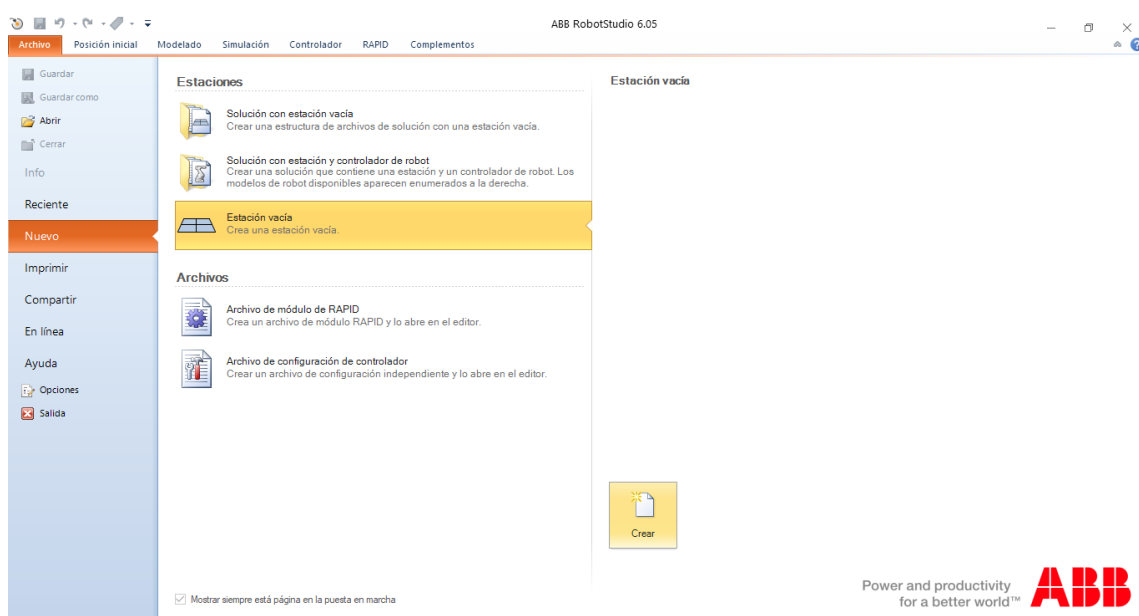


Figura 3.1: Creación nueva estación

- **Solución con estación vacía:** Crea una estructura de archivos de solución con una estación vacía.
- **Solución con estación y controlador de robot:** Crea una solución que contiene una estación y un controlador de robot, teniendo una lista con los modelos de robot ABB que podemos utilizar. Esta opción es la utilizada en este trabajo, debido a que al elegir el robot aparece en la estación el robot y el controlador, además de ser la mejor opción a la hora de exportar.
- **Estación vacía:** Crea una estación vacía. Esta opción la hemos utilizado en este tutorial introductorio de RobotStudio.

Una vez creada la estación vacía, el programa dispone de 7 pestañas

- **Archivo:** Tenemos diferentes opciones como guardado, abrir, nuevo, compartir, imprimir y las opciones de programa.
- **Posición Inicial:** Principal pestaña del programa donde se puede añadir robot, controlador, bibliotecas, coordenadas e importar geometría. Además de realizar la programación de trayectorias, seleccionar la herramienta y espacio de trabajo, sincronizar, movimientos del robot y herramientas gráficas.
- **Modelado:** Podemos crear un grupo de componentes, componentes inteligentes, importar geometría, también crear sólidos, superficies y curvas, operaciones de edición de CAD y crear mecanismo y herramientas.
- **Simulación:** Se pueden crear conjunto de colisión, lógica de estación, configuración de simulación, controlar la simulación, monitorizar, analizar las señales y poder realizar una grabación de la simulación.
- **Controlador:** En esta pestaña se puede añadir un controlador, reiniciarlo, varias herramientas, configuración del controlador y abrir la paleta FlexPendant.
- **RAPID:** Se utiliza para la programación del robot.
- **Complementos:** Desde esta pestaña se pueden descargar algunos complementos e instalar paquetes RobotPac como Painting, Machinig, Cutting, Palletizing.

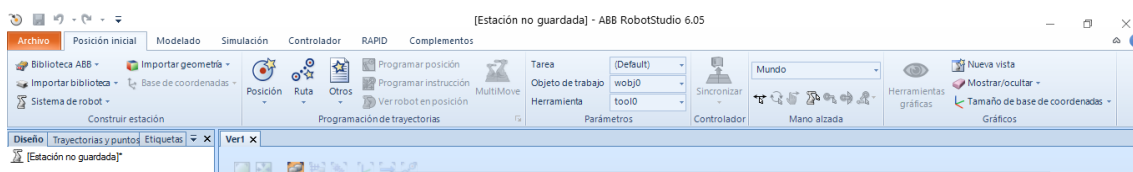


Figura 3.2: Pestañas programa RobotStudio

Ya familiarizados con el programa, lo primero que vamos a hacer es importar un robot, en la pestaña “Posición Inicial”, pinchamos sobre Biblioteca ABB y se despliega un listado con los modelos de robots ABB, seleccionamos el robot IRB 140 como en la figura 3.3 para realizar esta introducción.

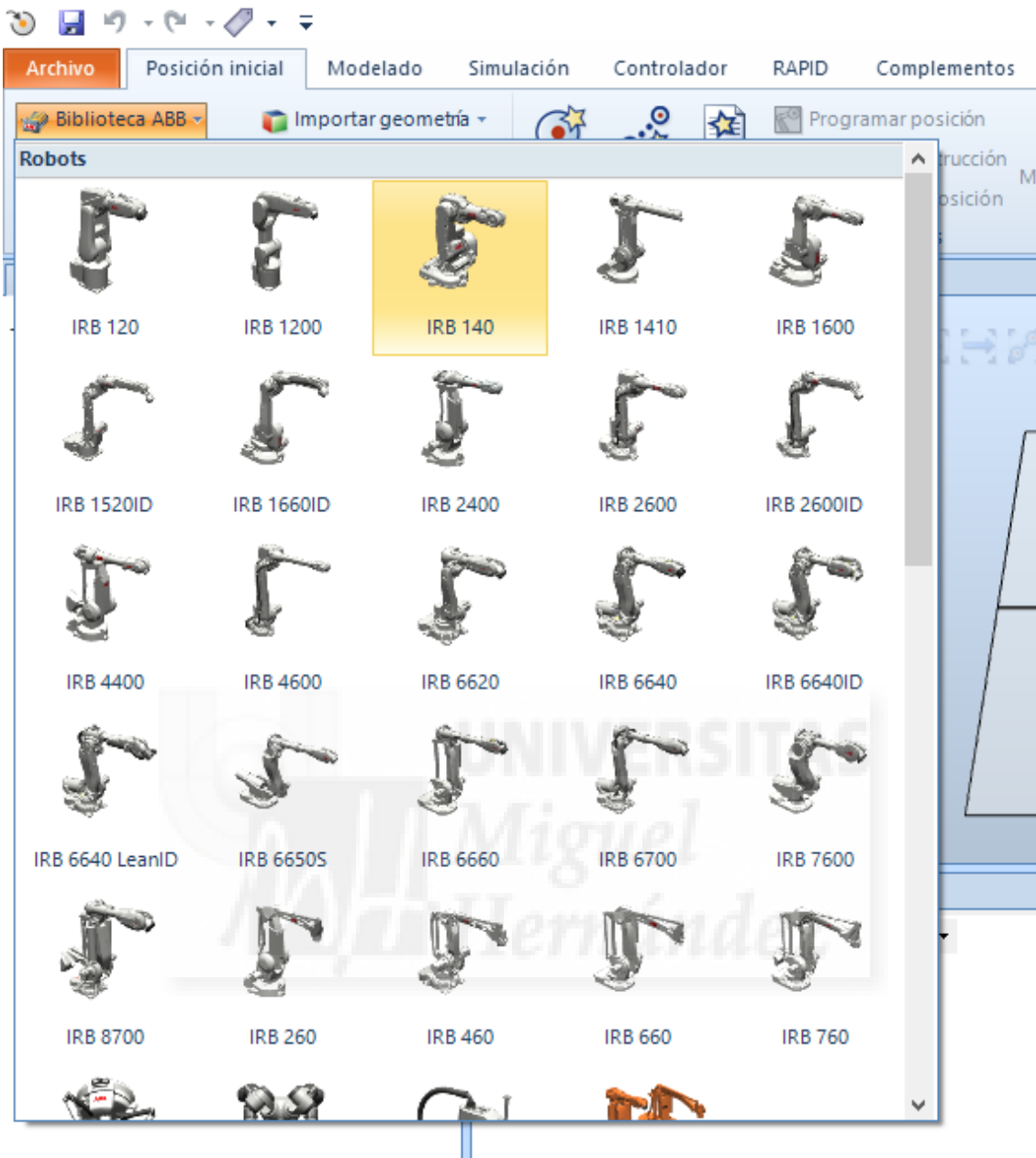


Figura 3.3: Robots de la Biblioteca ABB

Se carga el robot IRB 140 de la figura 3.4 en la estación.

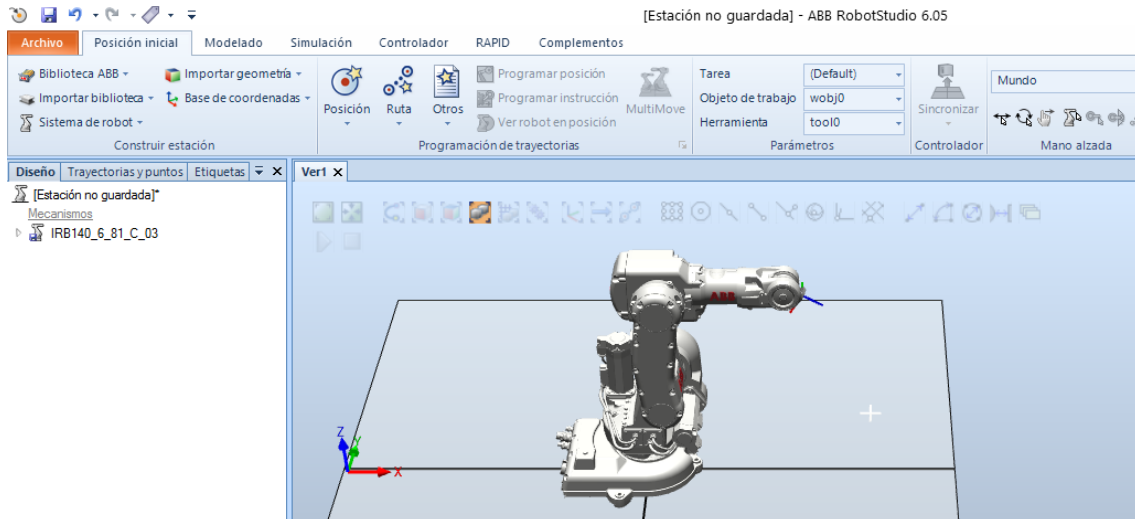


Figura 3.4: Robot ABB IRB 140

3.1.2 Creación de la herramienta de un robot

Todos los robots llevan una herramienta como efector final para poder realizar determinadas tareas como pintura, soldadura, pulido, pick and place. Además es necesaria una herramienta de trabajo del robot para poder realizar operaciones y crear objetivos, la herramienta se encuentra incorporada en la muñeca.

Vamos a crear una herramienta a través de modelado, podremos crear un tetraedro, cilindro, cono, pirámide o esfera. Para este tutorial crearemos un cono, en la pestaña “Modelado”, pinchamos en “Sólido” y escogemos el “Cono” tal y como muestra la figura 3.5.

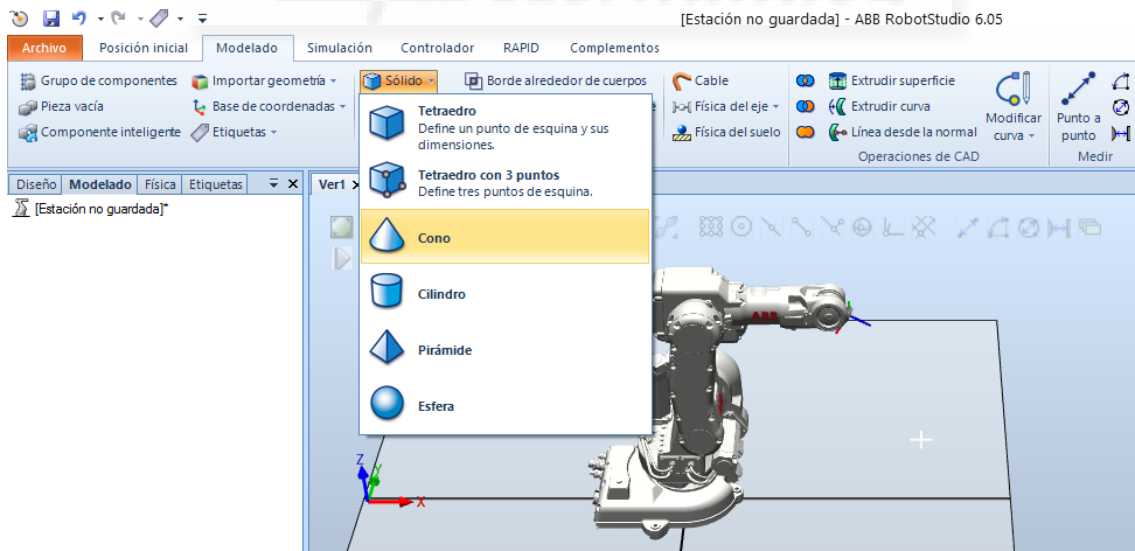


Figura 3.5: Creación de Sólidos

Se configuran los parámetros del cono en la figura 3.6, con radio de 50 mm, diámetro de 100 mm y una altura de 100 mm y se selecciona crear. Al no realizar cambios en la posición, la pieza será creada por defecto en el (0,0,0).

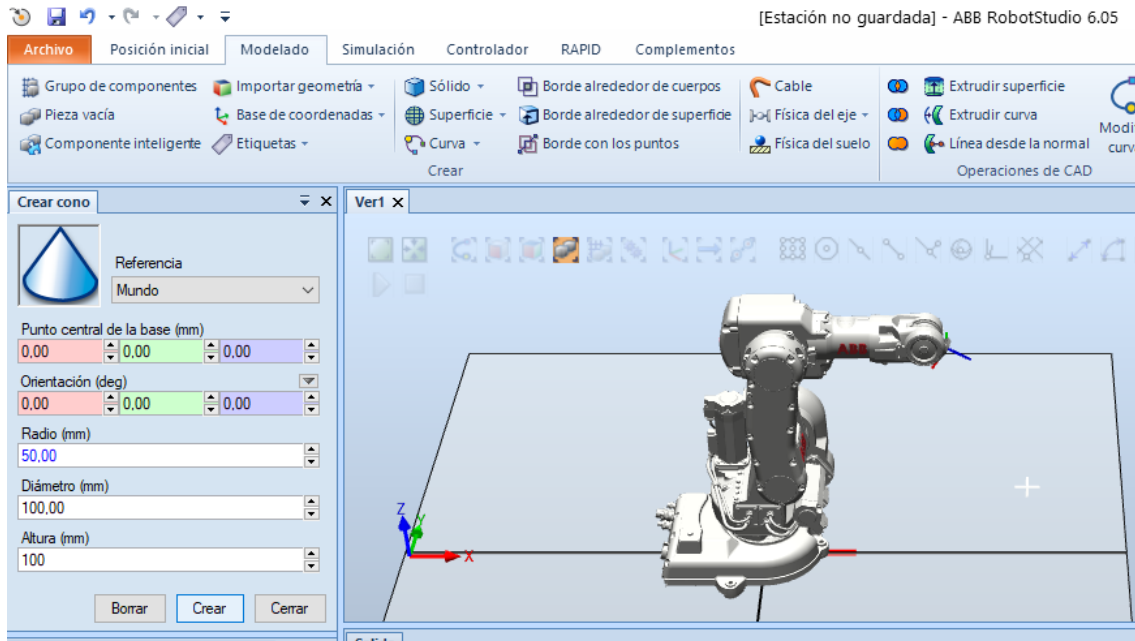


Figura 3.6: Creación de un Cono

En este paso se transforma la pieza creada en herramienta, consiste en transformar en herramienta la geometría creada, paso necesario para que el programa interprete la pieza como una herramienta del robot. En la pestaña “Modelado”, seleccionamos la opción “Crear herramienta” de la figura 3.7.



Figura 3.7: Crear herramienta

Lo primero que vamos a hacer es ponerle nombre a la herramienta, le hemos llamado “Cono”, seleccionamos que es una pieza existente “Pieza_1”, le indicamos la masa y si sabemos el centro de gravedad y los momentos de inercia se pueden añadir. En el segundo paso se puede añadir posición y orientación relativa al extremo de la herramienta, para que las posiciones del robot que se crean posteriormente se realicen respecto a dicho extremo. Esta posición se denomina TCP (Tool Central Point) y hay que seleccionar el botón de la flecha “→” y se selecciona “Terminado” como en la figura 3.8.

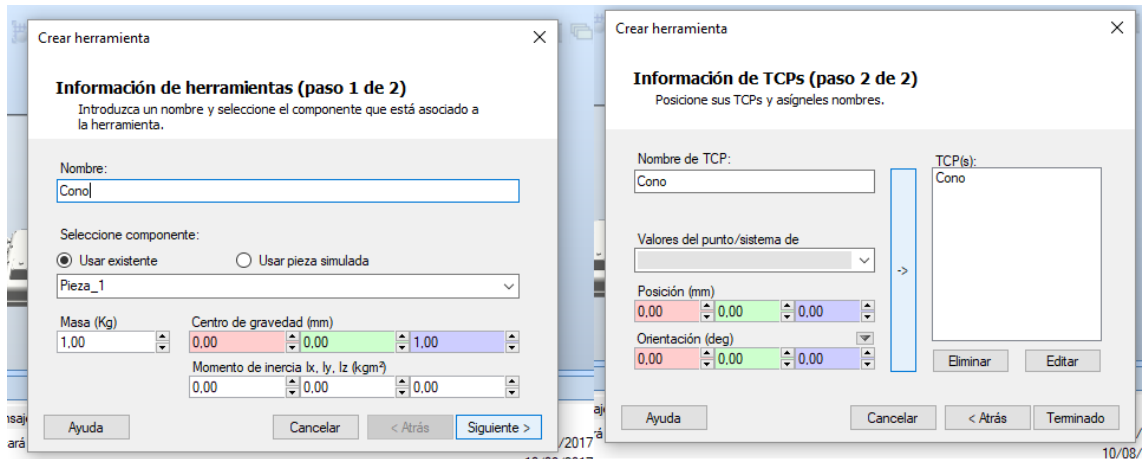


Figura 3.8: Configuración herramienta

Para finalizar hay que adjuntarla al robot en la figura 3.9, seleccionando “Cono” en diseño y arrastrarlo al robot IRB140, cuando pregunta si deseas actualizar la posición, le decimos que “Sí” y automáticamente la herramienta se pondrá en la muñeca del robot siendo parte de este.

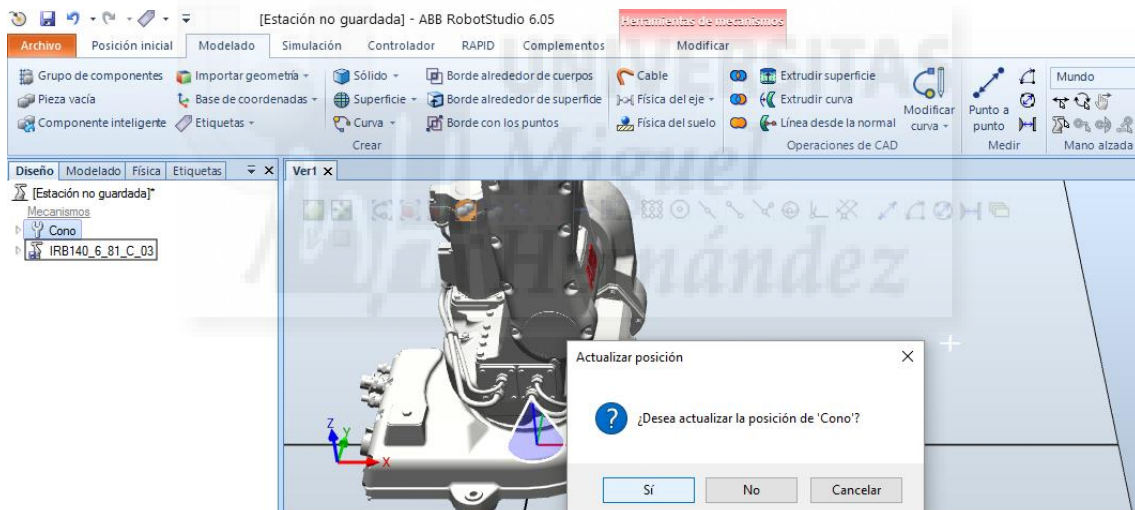


Figura 3.9: Conexión de la herramienta al robot

En la figura 3.10 aparece la herramienta conectada al robot.

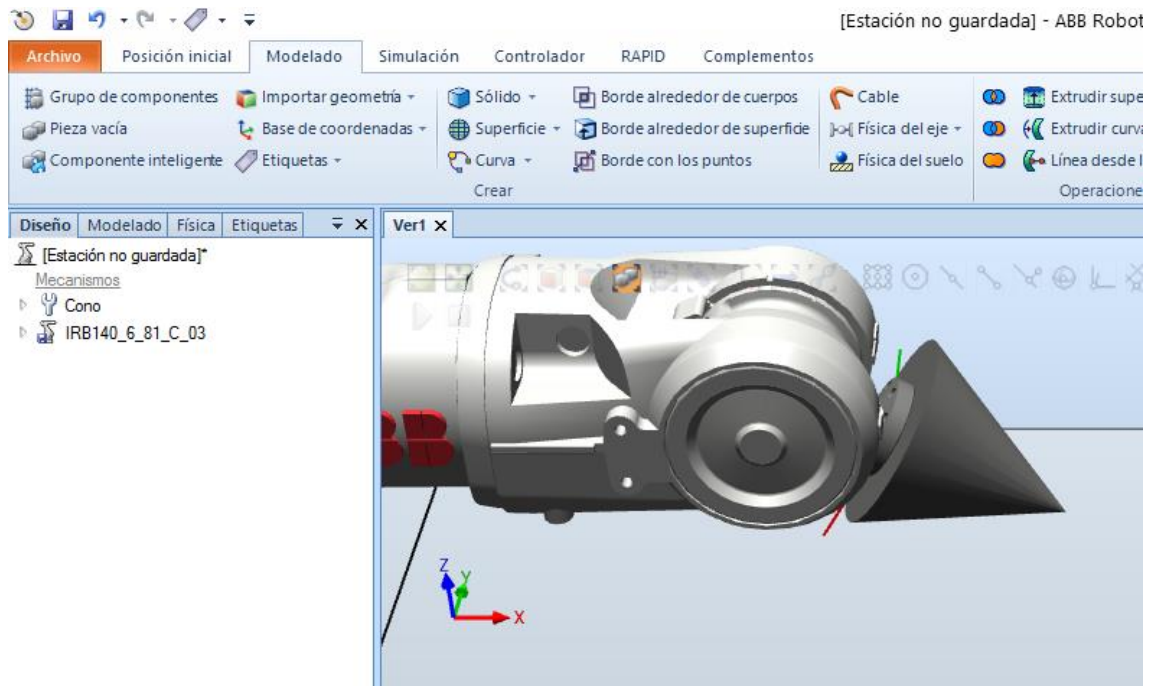


Figura 3.10: Herramienta conectada al robot

También tenemos la posibilidad de “Importar geometría” en la figura 3.11 desde otros programas para realizar la herramienta.

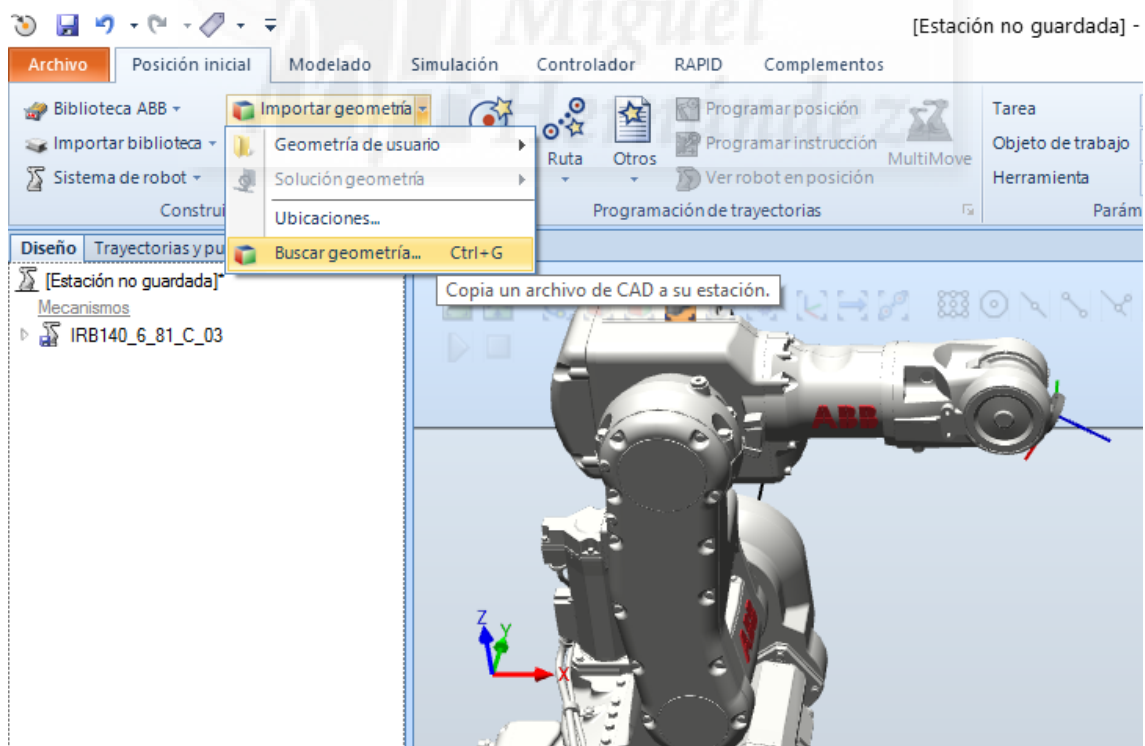


Figura 3.11: Importar geometría

Otra posibilidad es añadir la herramienta desde la propia biblioteca del programa, podemos encontrar algunas herramientas, para añadir en la figura 3.12, en la

pestaña de “Posición Inicial”, realizando clic en “Importar Biblioteca” y luego dentro de “Equipamiento” tendremos varias herramientas y otros objetos para añadir a nuestra estación.

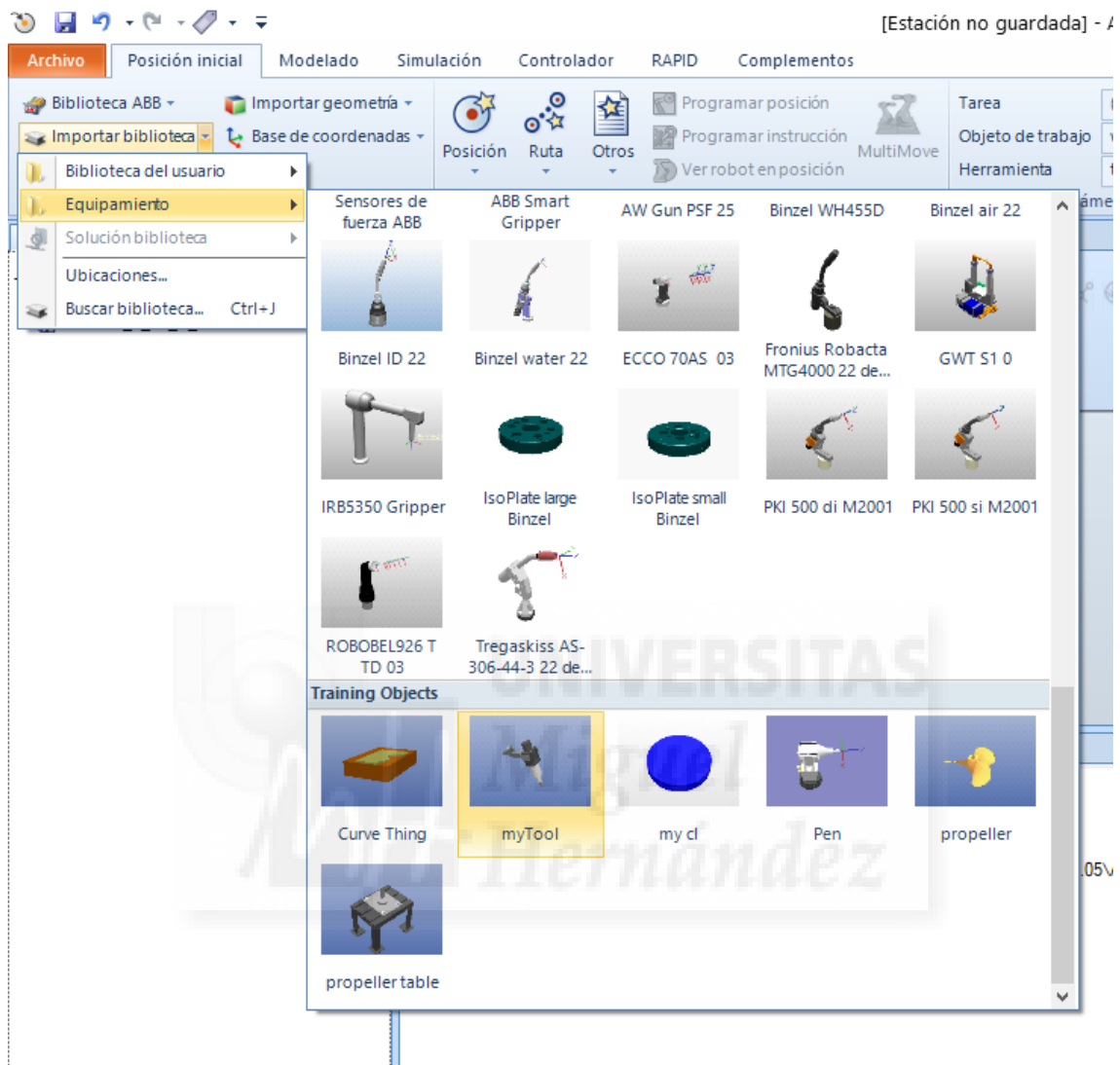


Figura 3.12: Biblioteca RobotStudio

En la figura 3.13 hemos seleccionado “myTool” de las herramientas disponibles.

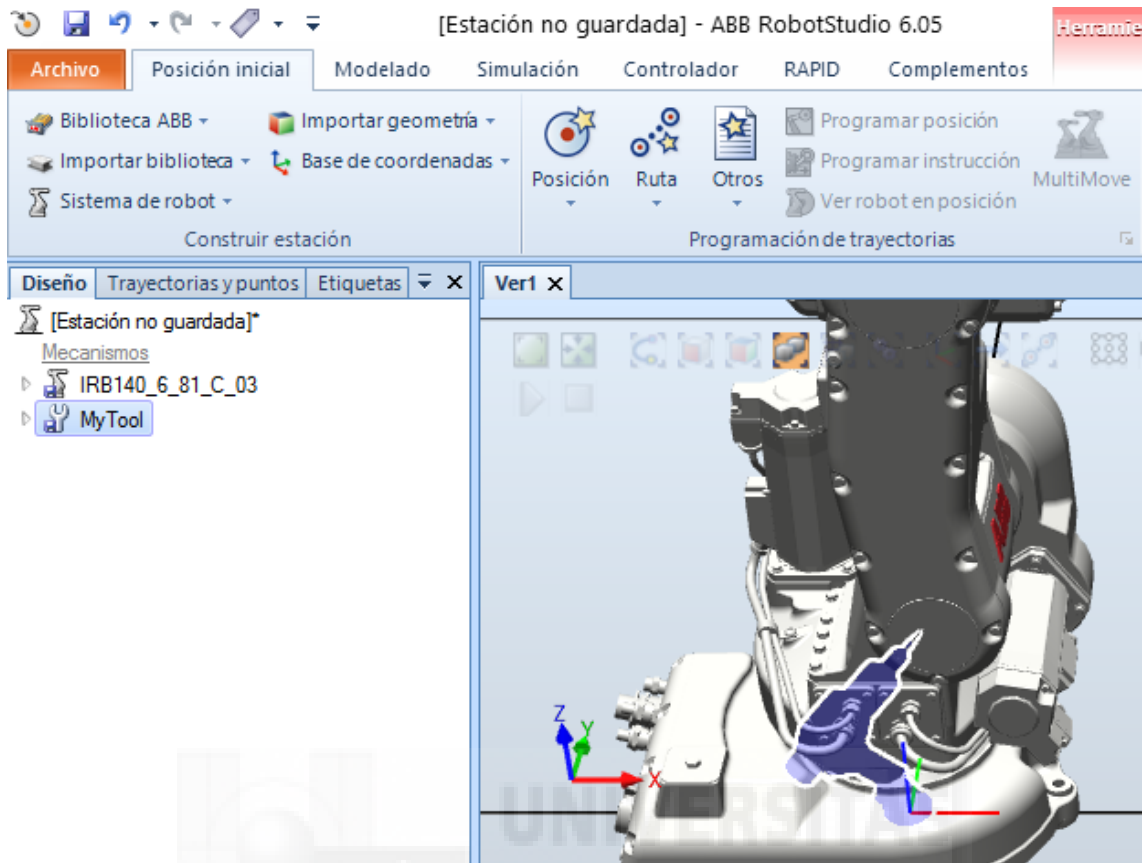


Figura 3.13: Herramienta de la biblioteca

Realizamos el paso que anteriormente hemos hecho para conectar la herramienta con el robot, en la figura 3.14 arrastrando “MyTool” al “IRB140”.

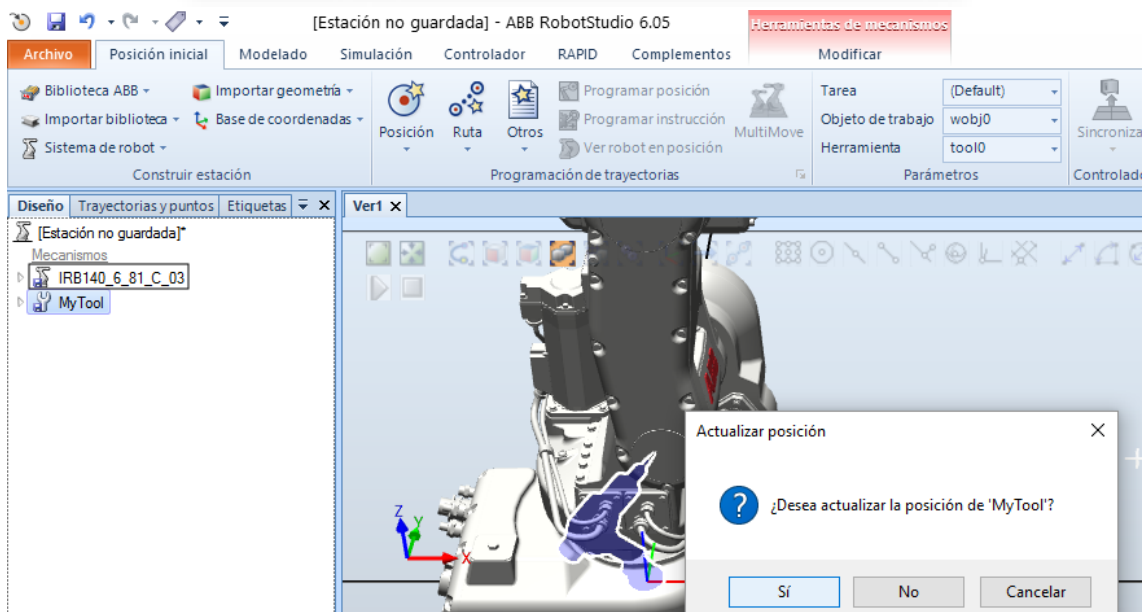


Figura 3.14: Herramienta de la biblioteca conectada al robot

3.1.3 Creación del controlador de un robot

De momento solo tenemos la herramienta adjunta al robot, sin embargo no se pueden programar trayectorias ni realizar movimientos, para realizar esto es necesario crear un controlador que va asociado al robot y dotarlo de inteligencia. Para añadir el controlador en la figura 3.15, en la pestaña “Posición Inicial”, en “Sistema de robot” seleccionamos “Desde diseño”.

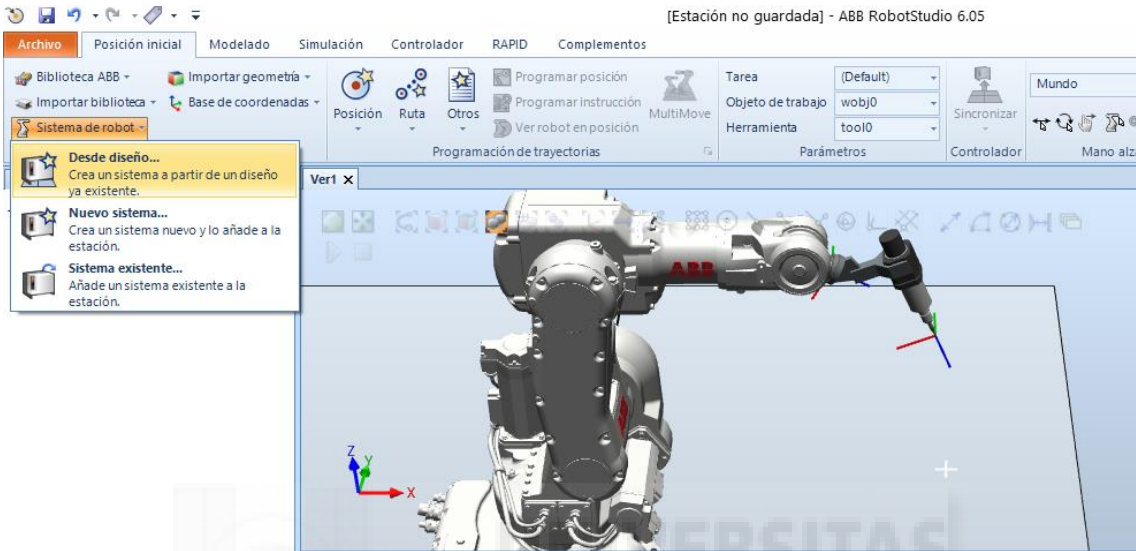


Figura 3.15: Creación de un controlador desde diseño

Le damos un nombre al controlador en la figura 3.16 y pulsamos en “Siguiente”, seleccionamos el mecanismo “IRB140” y pulsamos de nuevo “Siguiente”.

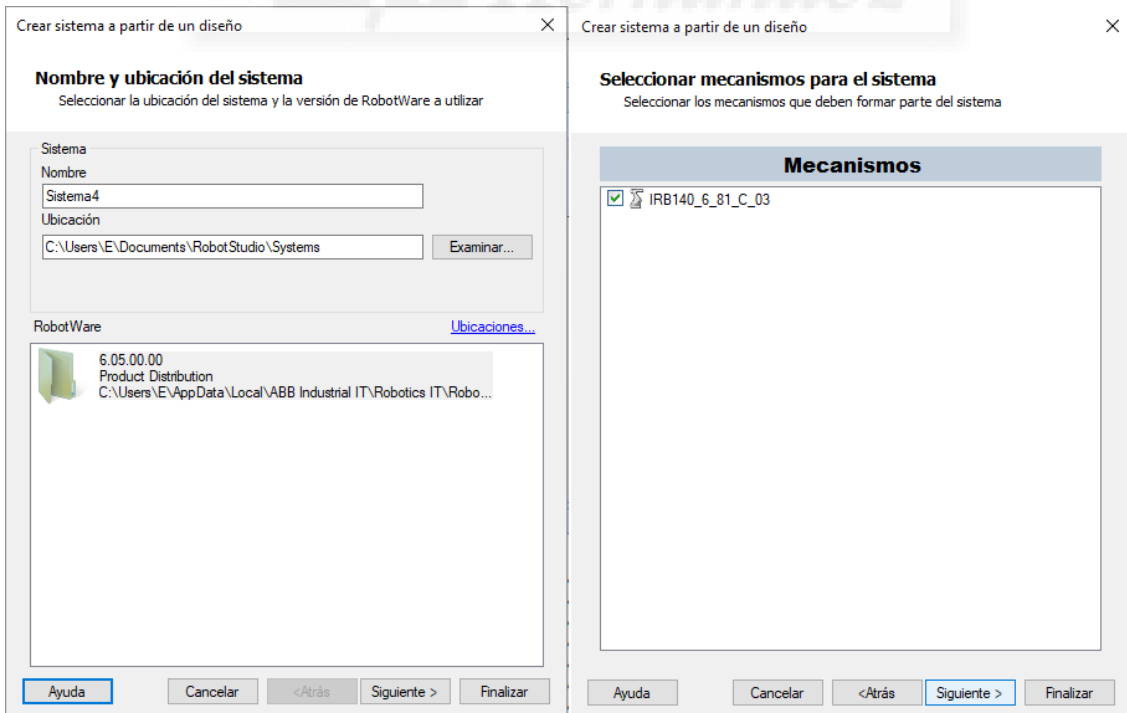


Figura 3.16: Creación del controlador

Para acabar de configurar el controlador en la figura 3.17, se pueden cambiar las opciones que vienen por defecto en el botón “Opciones del sistema”, por ejemplo podemos cambiar el idioma en “Default Language” a español, le damos a “Aceptar” y por último en “Finalizar” creándose el controlador del robot.

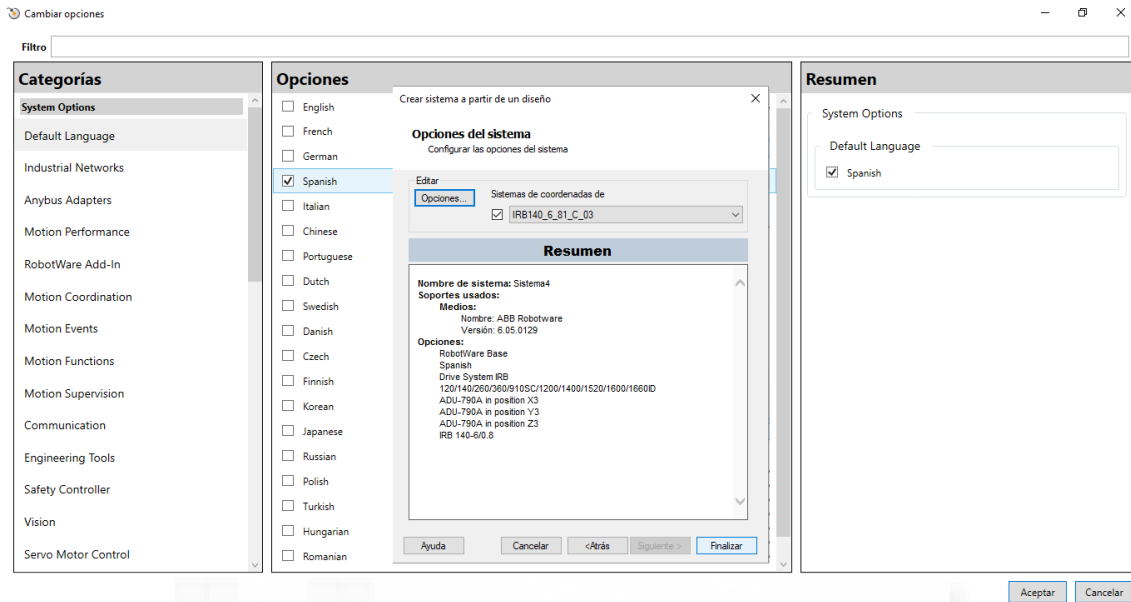


Figura 3.17: Opciones del sistema

3.1.4 Creación de planos de trabajo y posiciones

Antes de crear el plano de trabajo y los objetivos, vamos a necesitar un objeto donde se realizará la trayectoria. La pieza creada en la figura 3.18 va a ser un sólido, en concreto un tetraedro con 200 mm de longitud, anchura de 200 mm y una altura de 200 mm, lo vamos a posicionar respecto al (0,0,0) a 500 mm en el eje X y -100 mm del eje Y, donde la herramienta del robot va a poder alcanzar el tetraedro. Le damos a “Crear” y le cambiamos el nombre a “Cubo”.

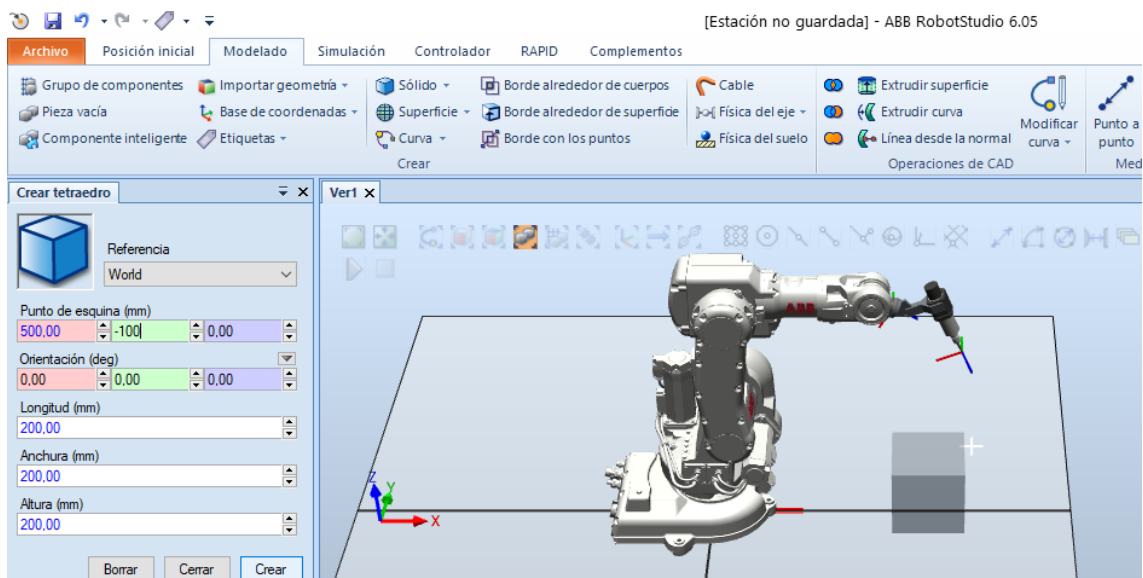


Figura 3.18: Creación de un tetraedro

Para crear el plano de trabajo hacemos clic en “Pestaña de Inicio”, seleccionamos “Otros” y después “Crear objeto de trabajo” como la figura 3.19.

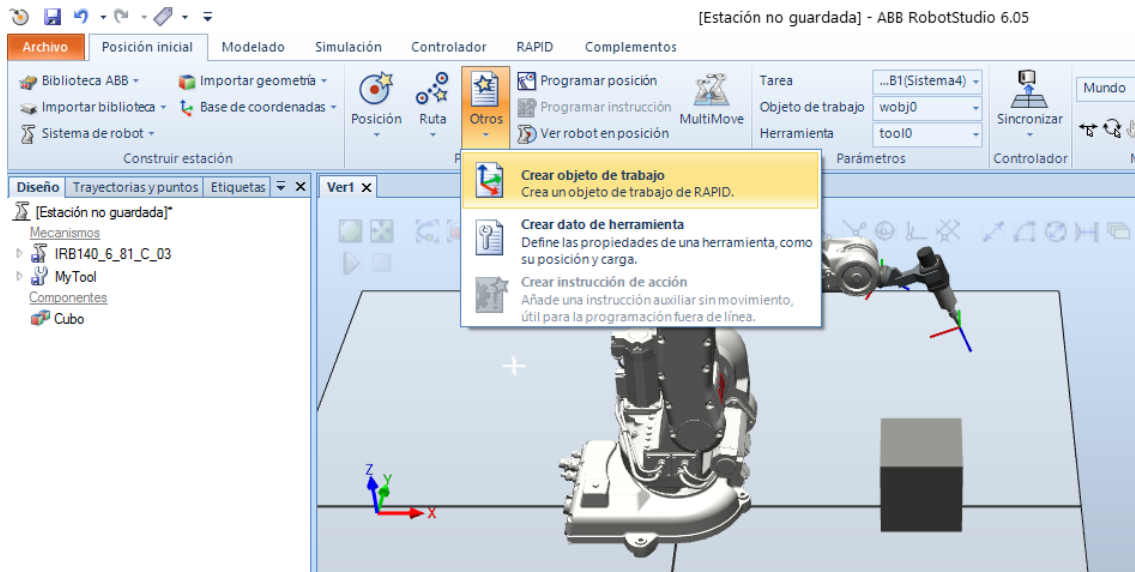


Figura 3.19: Crear un objeto de trabajo

En la figura 3.20, el nombre del objeto de trabajo lo dejamos por defecto “Workobject_1” (le podemos llamar como queramos), en Sistema de coordenadas usuario, seleccionamos en posición X para cambiar las coordenadas por defecto, pinchamos en la casilla roja y finalmente seleccionamos el punto que queremos que sea el punto de trabajo. Vamos a seleccionar la esquina superior izquierda cercana a nosotros (anteriormente debemos seleccionar en los iconos tanto Selección de pieza y punto final). Pinchamos en “Accept” y después en “Crear”, ya tendremos el objeto de trabajo creado.

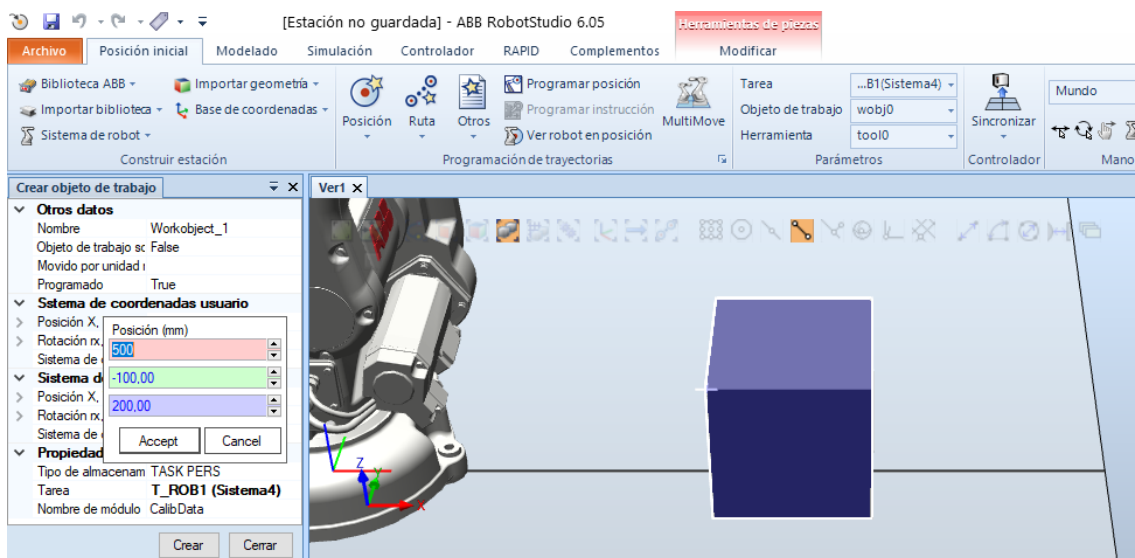


Figura 3.20: Posición objeto de trabajo

Antes de crear el objetivo en “Posición Inicial”, “Parámetros” debemos modificar el Objeto de trabajo a Workobject_1 y la Herramienta a MyTool, así los objetivos creados pertenecerán al objeto de trabajo que hemos creado y utilizamos la herramienta añadida.

Una vez seleccionado los parámetros en la figura 3.21, crearemos posiciones en la pestaña “Posición Inicial”, “Posición” y pinchamos en “Crear objetivo”.

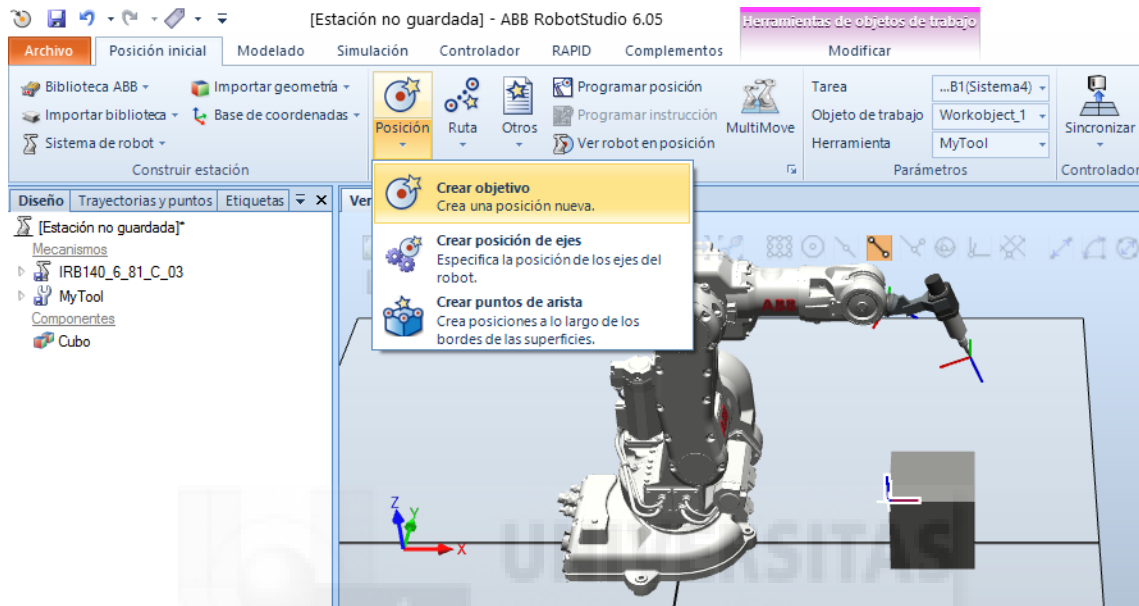


Figura 3.21: Crear objetivo

A continuación en la figura 3.22 creamos cuatro puntos señalando las esquinas superiores del cubo, es necesario seleccionar la casilla roja de Posición para poder coger la posición de los puntos. Para finalizar se pincha en crear.

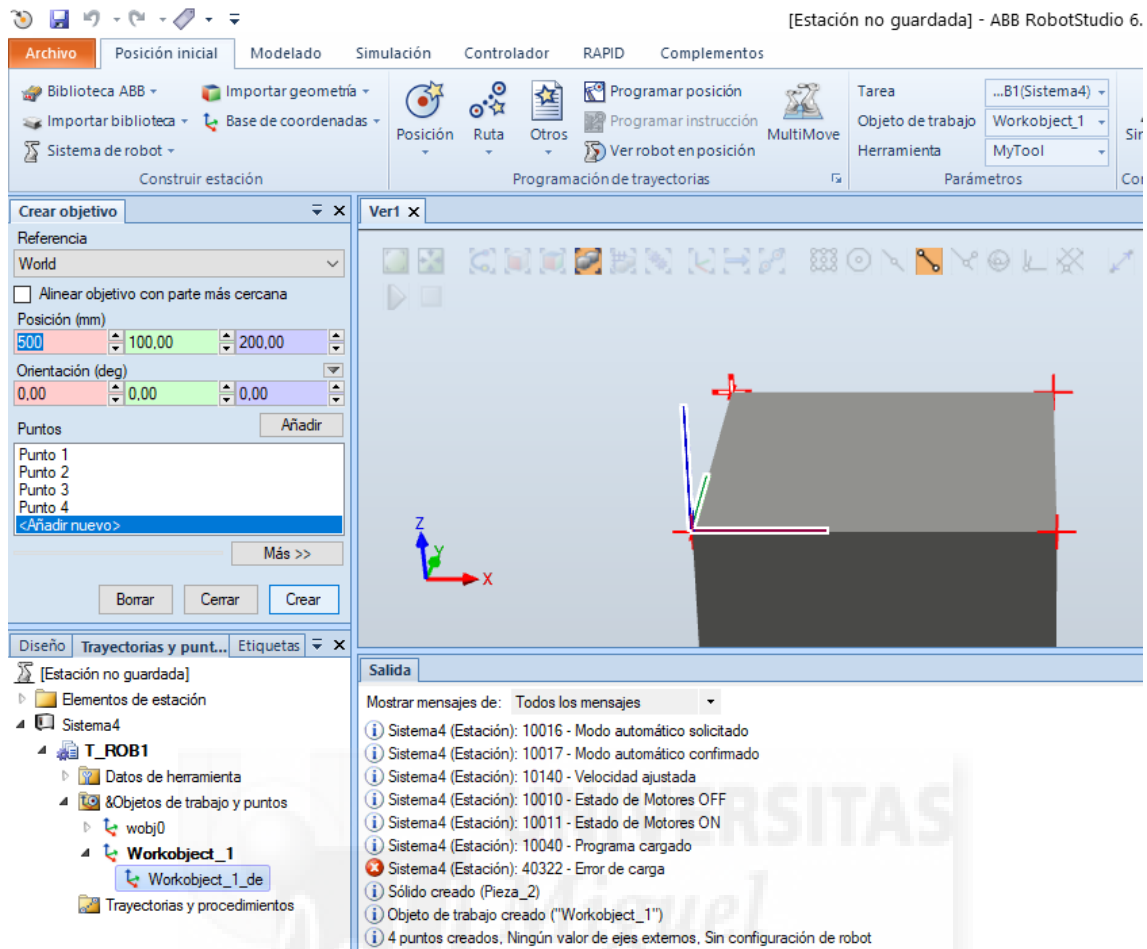


Figura 3.22: Creación puntos objetivo

Crearemos una posición inicial en la figura 3.23, para así situar la herramienta en una posición fuera de los puntos objetivo. Esta posición inicial la crearemos en el objeto de trabajo "wobj0".

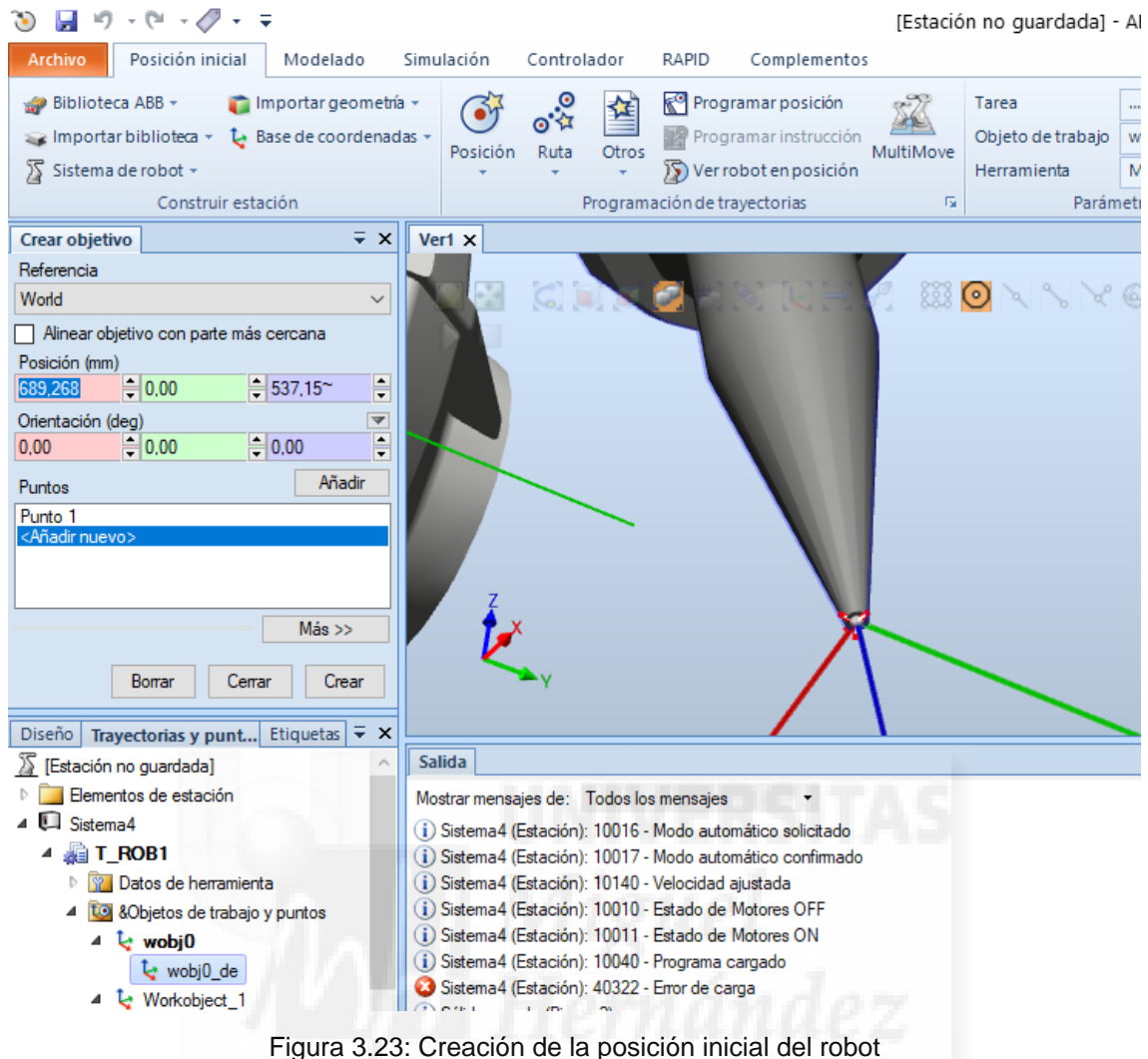


Figura 3.23: Creación de la posición inicial del robot

3.1.5 Creación de trayectorias

Una vez creado el conjunto de puntos, vamos a visualizar la posición de la herramienta "MyTool" en los puntos creados, la orientación que viene por defecto puede que no sea la idónea o que el robot no pueda alcanzarla.

Escogemos el primer punto creado "Target_10", clic con el botón derecho del ratón y seleccionamos "Ver herramienta en la posición" y pinchamos en "MyTool", entonces la herramienta se visualiza en el punto 1 que hemos creado anteriormente.

Como se aprecia en la figura 3.24, este punto con dicha orientación la herramienta no alcanza el punto Target_10, ya que al hacer clic en "Ver robot en posición", nos saldrá un mensaje de que el punto Target_10 se encuentra fuera de alcance.

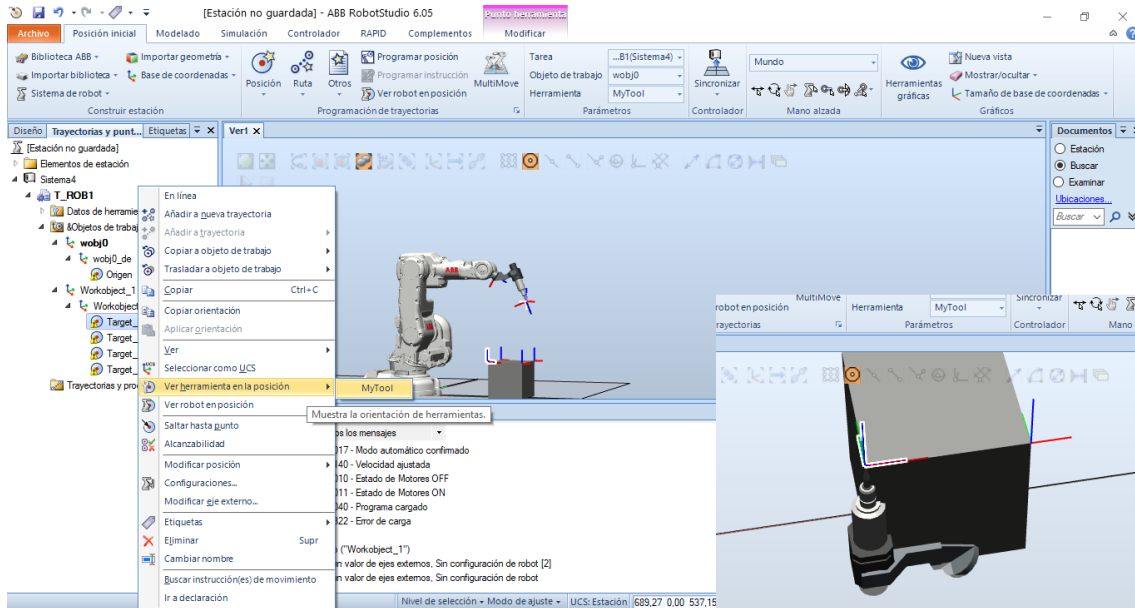


Figura 3.24: Ver herramienta en la posición

Para poder colocar la herramienta de forma que pueda ser alcanzada por el robot, será necesario realizar un giro al sistema de eje de coordenadas. Pulsamos otra vez clic derecho del ratón sobre “Target_10”, seleccionando “Modificar posición” y pinchamos la opción girar como aparece en la figura 3.25.

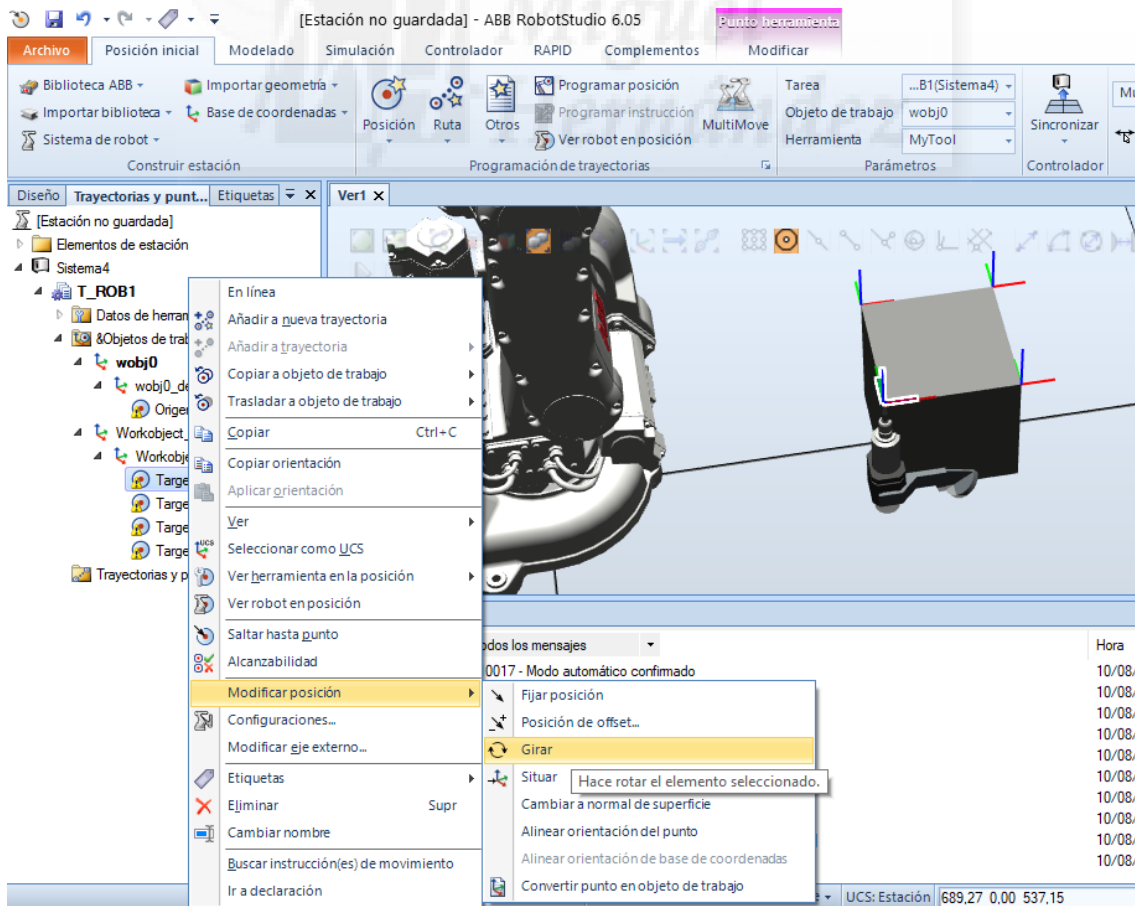


Figura 3.25: Girar posición de la herramienta

Se puede ir buscando giros hasta llegar a una posición donde el robot alcance el objetivo, estos giros pueden ser en los ejes de coordenadas X, Y, Z. En nuestro caso realizaremos dos giros en la figura 3.26, el primer giro 180 grados respecto al eje X y el segundo giro se realiza también a 180 grados pero respecto al eje Z.

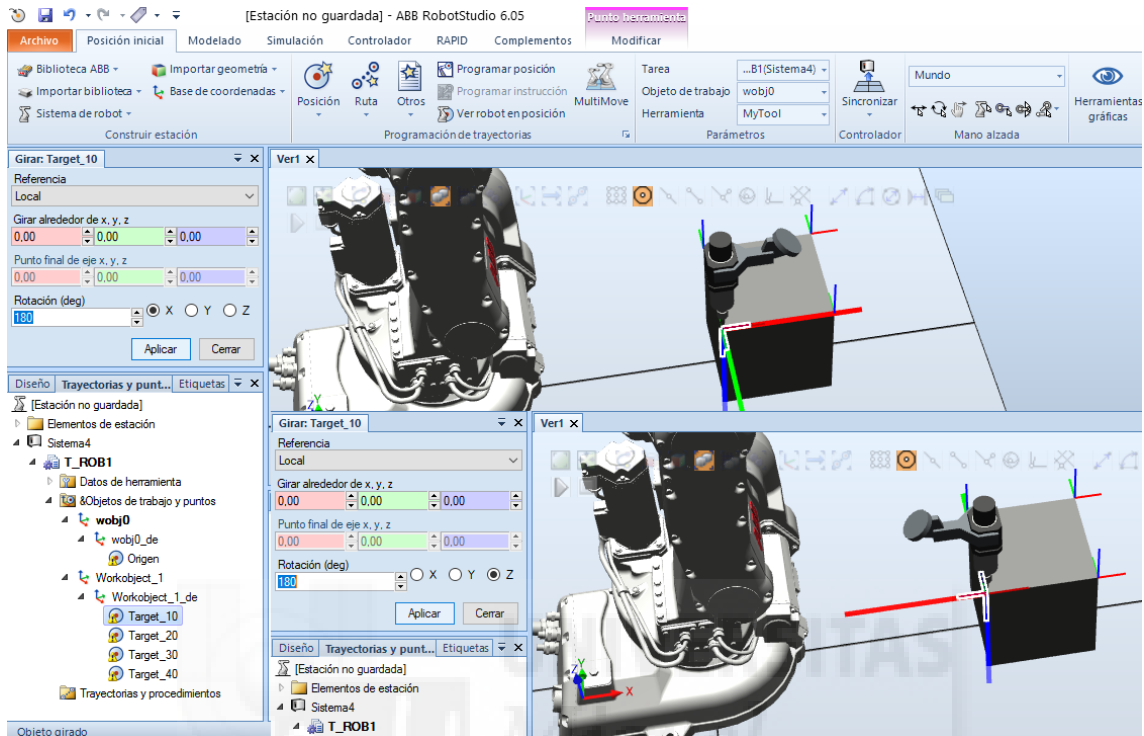


Figura 3.26: Giros de la posición respecto al eje X y al eje Z

Si seleccionamos la opción de “Ver robot en posición” que se encuentra en la pestaña “Posición Inicial” o haciendo clic en el botón derecho también se puede seleccionar. Como se ve en la imagen 3.27 el robot alcanza el “Target_10”.

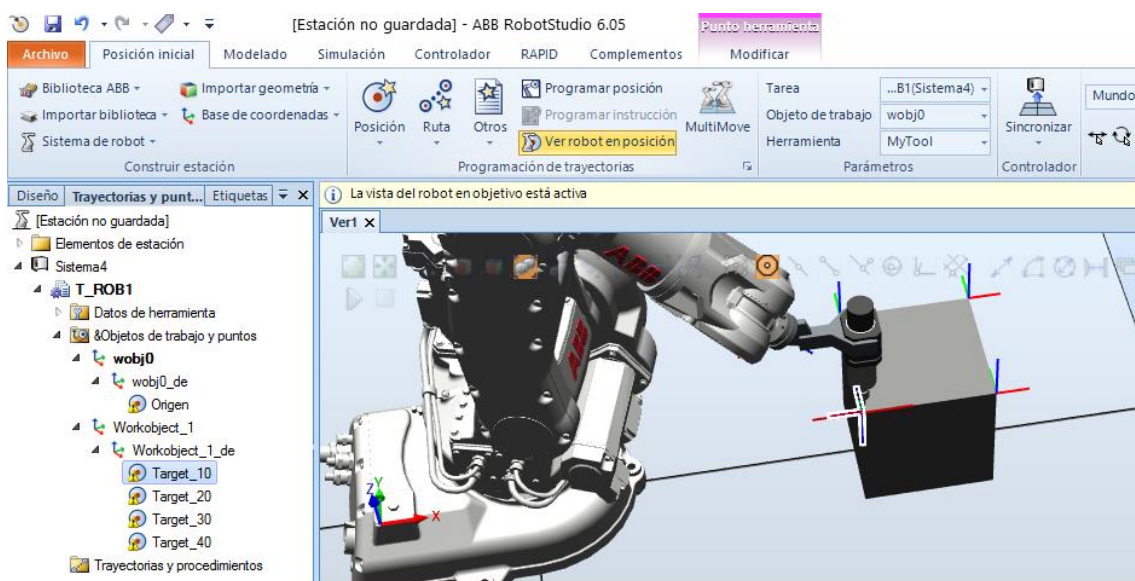


Figura 3.27: Ver robot en posición

Para realizar la trayectoria tendremos dos opciones como se puede ver en la figura 3.28, crear una trayectoria vacía o crear una trayectoria automática en la que necesitaremos una geometría o curva. En nuestro caso vamos a realizar una “Trayectoria vacía”, seleccionándolo en la pestaña “Posición Inicial” y en la opción “Ruta”.

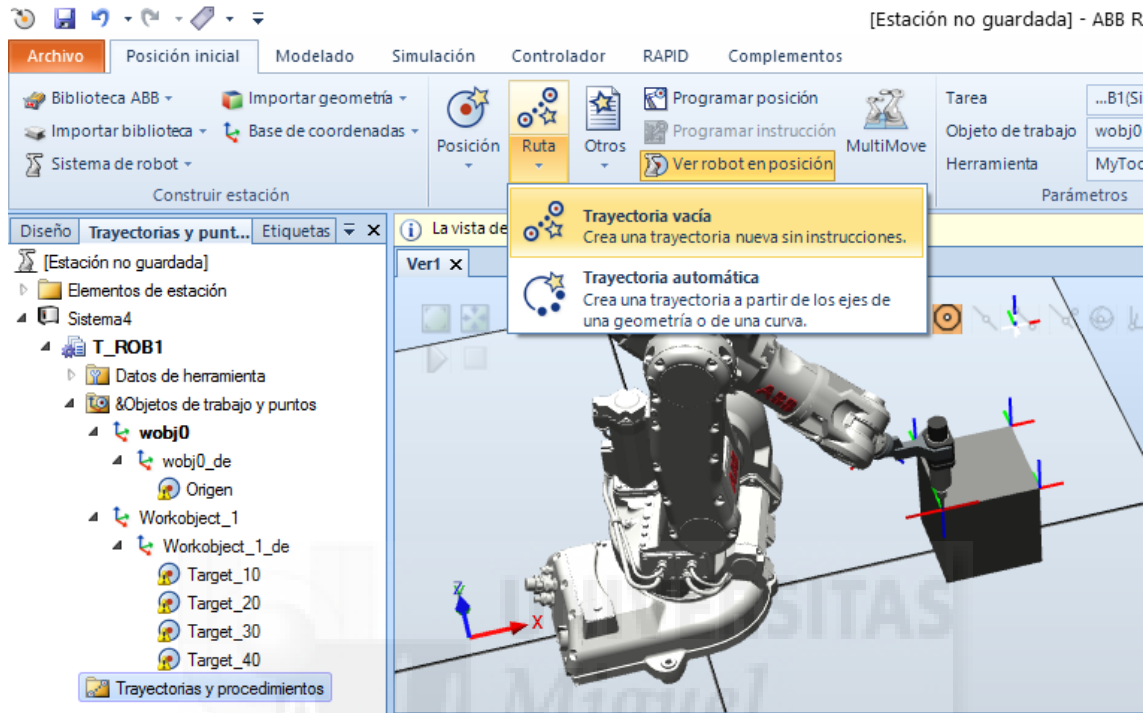


Figura 3.28: Creación de una trayectoria vacía

Al crear una Trayectoria vacía, se creará la trayectoria Path_10, la cual se encuentra vacía, seleccionaremos el punto “Origen” y arrastramos los puntos del “Target_10” al “Target_40” y volvemos a arrastrar el punto “Origen” para que la trayectoria finalice en el inicio, el orden será el orden que llevara la trayectoria. Hemos realizado la trayectoria que empiece en el origen, recorra las cuatro esquinas del cubo y vuelva al origen. Por defecto se creará el movimiento de la trayectoria en MoveL como ocurre en la figura 3.29.

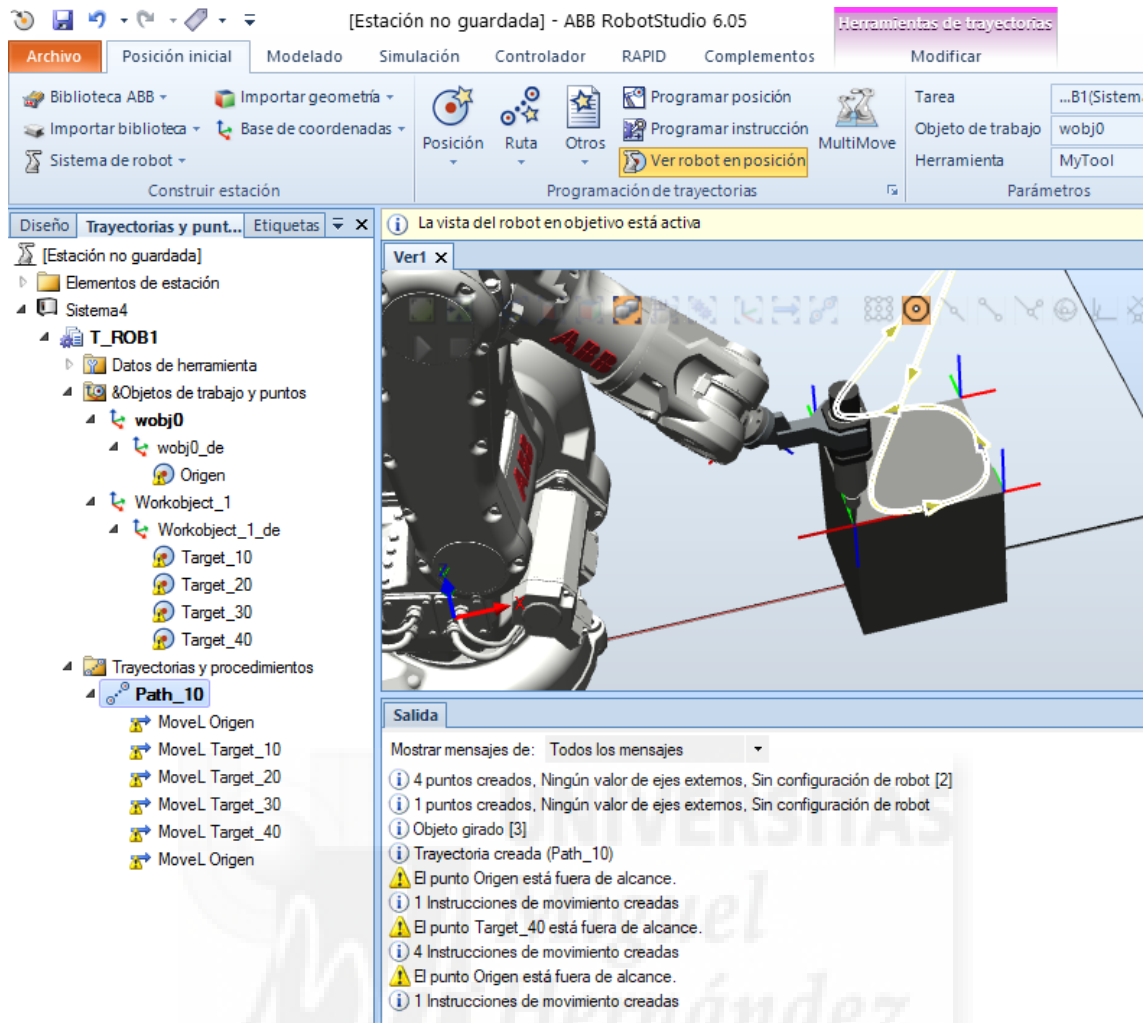


Figura 3.29: Trayectoria Path_10

Para que la trayectoria pueda pasar por todos los puntos, hay que hacer que estos puntos sean alcanzables por la herramienta del robot, por tanto habrá que girar todos los puntos tal y como hemos hecho con el punto "Target_10". Para ser más rápido en la figura 3.30 tenemos la opción de "Copiar orientación" haciendo clic en el botón derecho del ratón en Target_10 y seleccionamos "Aplicar orientación" en los demás puntos.

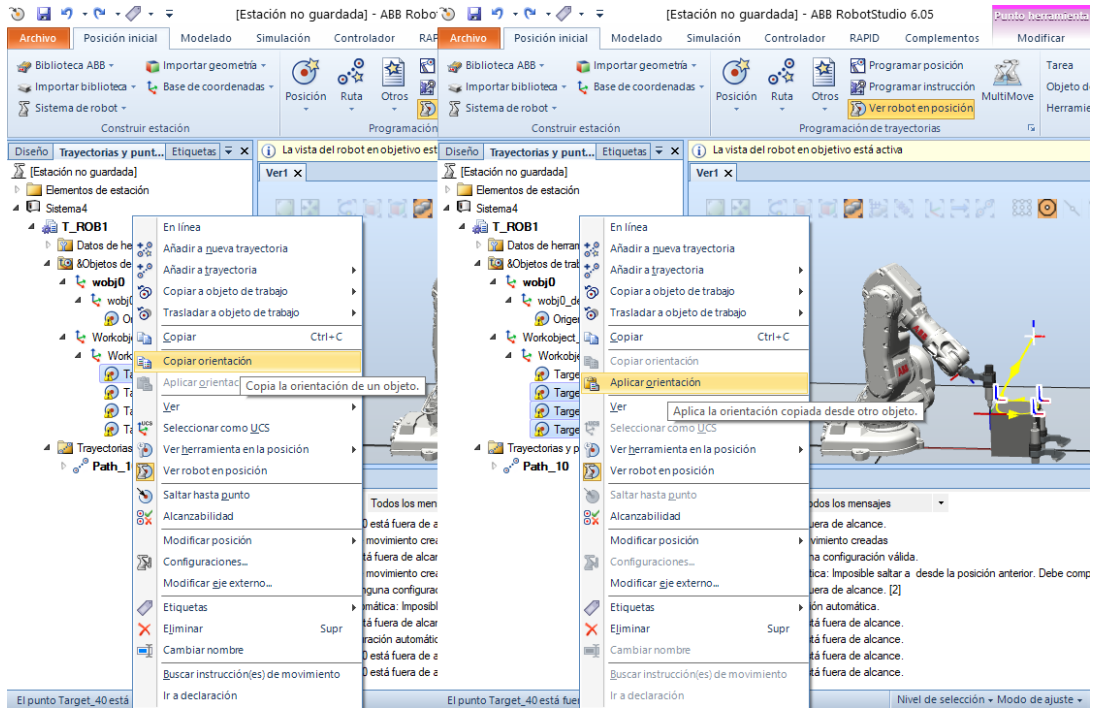


Figura 3.30: Copiar y aplicar orientación en los demás objetivos

Una vez modificada la orientación, ya se puede realizar la configuración de los ejes del robot. Nos aparece un símbolo de advertencia en cada posición, ya que el robot puede acceder a cada punto con una configuración diferente de sus ejes. Para realizar la configuración, haciendo clic botón derecho sobre Path_10, "Configuraciones", seleccionamos "Configuración automática" y nos aparecerán diferentes configuraciones que pueda tener el robot, en nuestro caso seleccionamos como en la figura 3.31 la configuración "Cfg1 (0,0,0,0)".

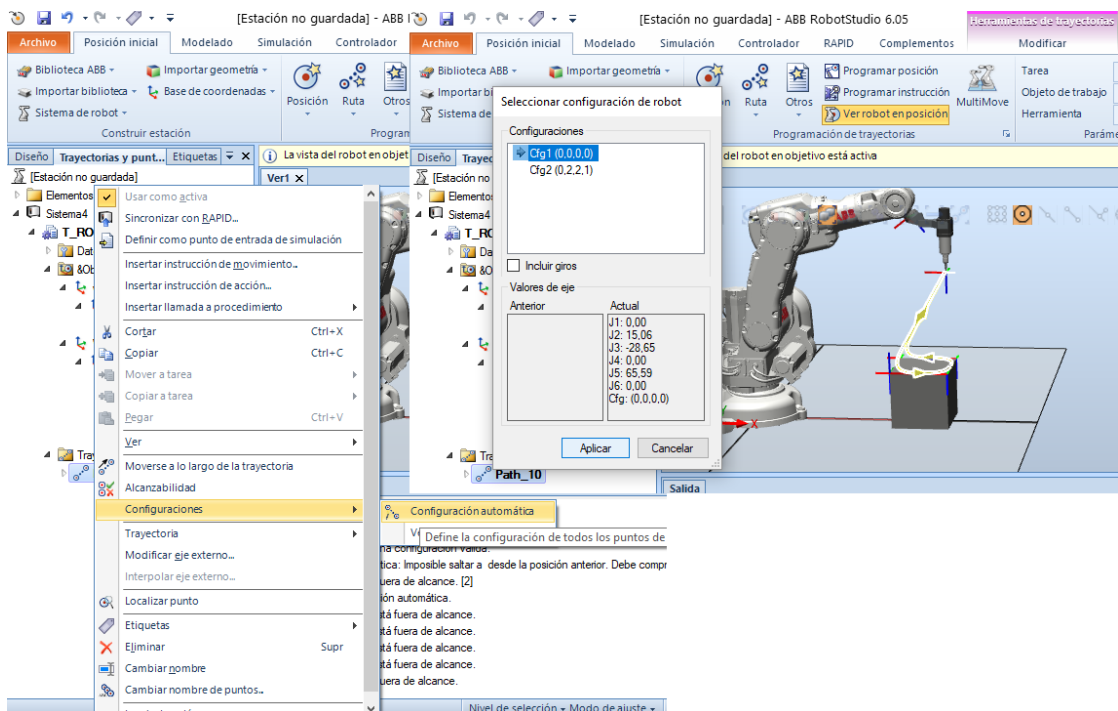


Figura 3.31: Configuración automática de la trayectoria

Tenemos varias opciones en las trayectorias que podemos cambiar como el movimiento, la velocidad, precisión, herramienta y objeto de trabajo. Seleccionando un punto de trayectoria o varios y en la zona de abajo a la derecha en la figura 3.32 tenemos estas opciones:

- Plantilla de Instrucciones: podremos cambiar el tipo de movimiento MoveL a MoveJ o MoveC, en el apartado XX veremos que significa cada tipo de movimiento.
- Speed: se puede cambiar la velocidad con la que se mueve el robot, por defecto aparecerá v1000, siendo la velocidad en mm/s.
- Zone: esta opción se refiere a la precisión del movimiento, los valores referencian al error en distancia (en mm) que hay desde el TCP de la herramienta del robot al punto objetivo en el momento en el que se ejecuta la instrucción y el robot está pasando por dicha posición. Siendo fine el más preciso y z200 el menos preciso, por defecto aparece z100.
- Herramienta: se podrá escoger entre varias herramientas que tengamos asociadas como herramientas.
- Objeto de trabajo: finalmente también podremos cambiar el objeto de trabajo por otro que nos interese.

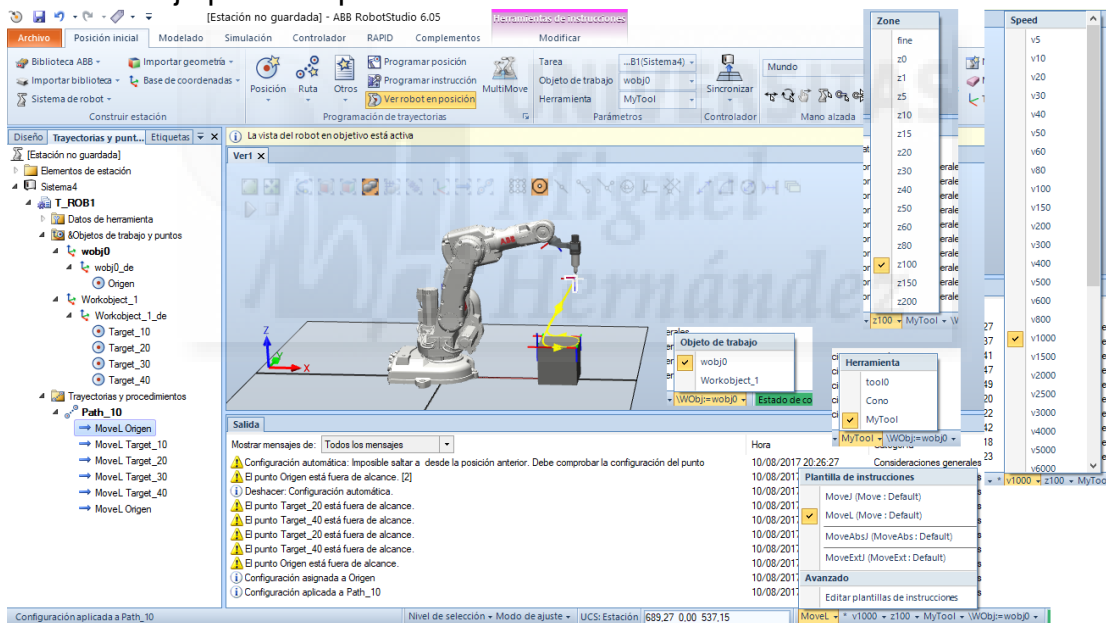


Figura 3.32: Instrucciones de los puntos de la trayectoria

Otra forma de editar las instrucciones es haciendo clic en el botón derecho en cada punto de trayectoria o en varios y seleccionando "Editar instrucción", tendremos las mismas opciones y alguna más que aparecen en la figura 3.33.

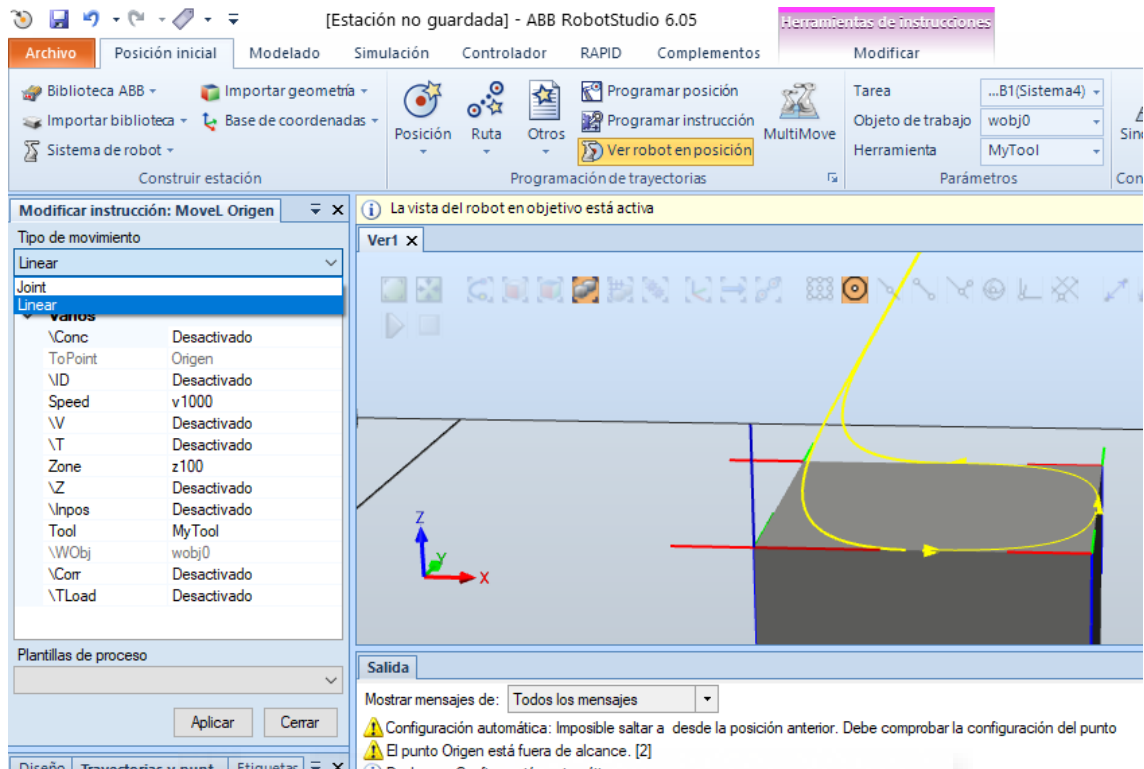


Figura 3.33: Modificar instrucciones

3.1.6 Introducción a RAPID

RAPID es un lenguaje de programación textual de alto nivel desarrollado por la empresa ABB. 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:

- Una rutina principal (main): Rutina donde se inicia la ejecución.
- Un conjunto de sub-rutinas: Sirven para dividir el programa en partes más pequeñas a fin de obtener un programa modular.
- Los datos del programa: Definen posiciones, valores numéricos, sistemas de coordenadas, etc.

Para realizar la programación en RAPID, en primer lugar hay que sincronizar la estación con el controlador virtual. Como en la figura 3.34 se encuentra en la pestaña "Posición Inicial", seleccionamos "Sincronizar" y pinchamos en "Sincronizar con Rapid".

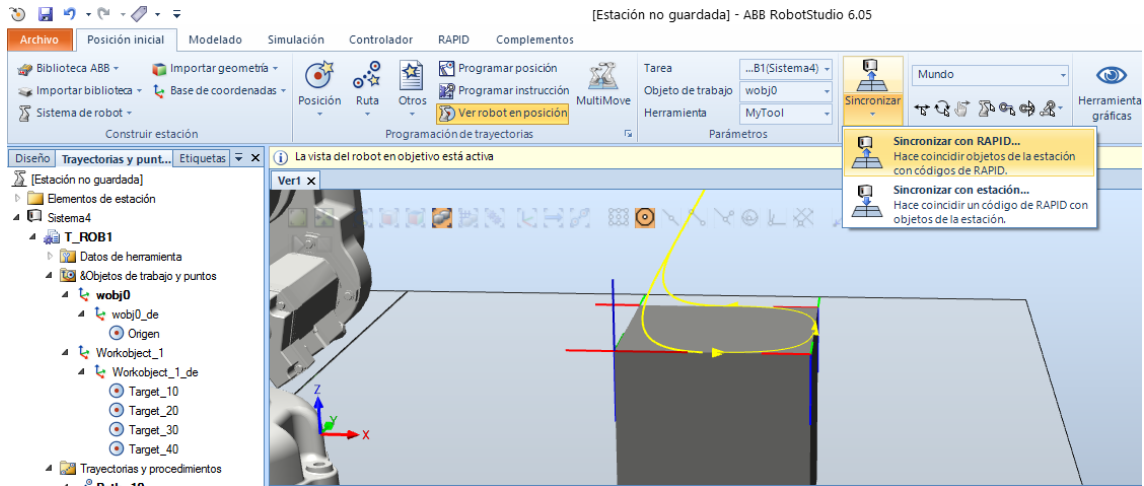


Figura 3.34: Sincronización con RAPID

Con la sincronización conseguimos que los datos de la herramienta, objeto de trabajo con sus puntos correspondientes y las trayectorias sean volcadas al programa en la figura 3.35, apareciendo los módulos de programa “CalibData” y “Module1” necesarios para la programación del robot.

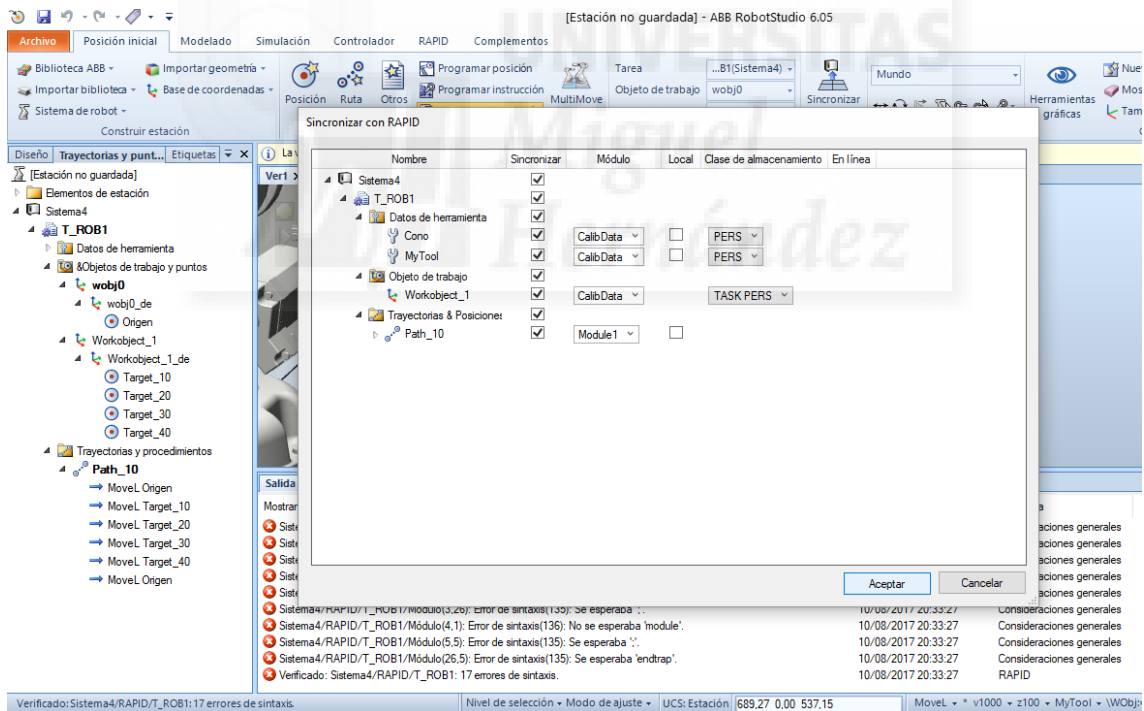


Figura 3.35: Sincronización de datos, objeto de trabajo y trayectorias

Para poder programar entramos en la pestaña RAPID y hacemos clic dentro del controlador, abriendo Rapid y en “T_ROB1” seleccionando el Module1. Gracias a la sincronización en la figura 3.36 nos aparecerán definidos los puntos como constantes robtarjet y también las trayectorias con sus movimientos. En el main tendremos el programa principal, es posible realizar cambios del programa en este apartado y aplicar los cambios realizados. Por defecto es posible que nos

aparezca un “Module2” dándonos unos errores, entonces eliminamos este módulo.

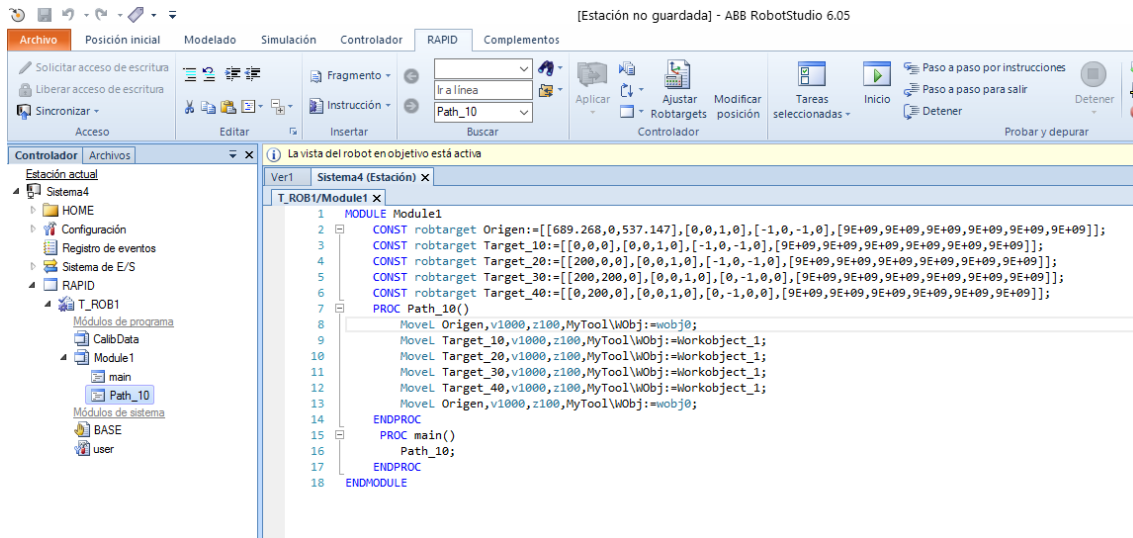


Figura 3.36: Programa RAPID

El módulo del programa “Module1” engloba todos los programas. La instrucción inicial es “MODULE Module 1” y debe finalizar el programa con “ENDMODULE”. Seguidamente hay que realizar la declaración de los datos mediante el comando CONST seguido de la palabra robtarget, el nombre de la posición, sus coordenadas relativas al objeto de trabajo y la configuración de ejes en dicho punto, en la figura 3.37 tenemos un ejemplo.

```
CONST robtarget Origen=[[689.268,0,537.147],[0,0,1,0],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

Figura 3.37: Ejemplo de un punto en RAPID

Y para cada programa, debe ser inicializado con “PROC Nombre_del_programa ()” y finalizarlo con “ENDPROC”.

Respecto a la instrucción de movimiento ‘Move’ que anteriormente hemos utilizado, existen 3 principales tipos de movimiento:

- MoveL: desplazamiento del extremo del robot hasta el punto indicado siguiendo una línea recta.
- MoveC: desplazamiento del extremo del robot hasta el punto indicado siguiendo un círculo, habiendo dos posiciones, una definiendo un punto intermedio del círculo y otra que define el punto destino.
- MoveJ: desplazamiento del extremo del robot hasta el punto indicado rápidamente, sin garantizar cuál es la trayectoria seguida (no hay coordinación de velocidad entre los distintos ejes del robot).

En la imagen 3.38 tenemos un ejemplo de instrucción de movimiento, donde en primer lugar se define el tipo de movimiento, seguido del punto de destino, la velocidad del movimiento, la precisión y la herramienta que está utilizando el robot.

```
MoveL Origen,v1000,z100,MyTool\WObj:=wobj0;
```

Figura 3.38: Ejemplo de instrucción de movimiento MoveL en RAPID

Instrucciones para la utilización de entradas y salidas

Principalmente interesarán dos tipos de operaciones, comprobar el valor de una entrada y fijar el valor de una salida.

Para fijar el valor de las salidas utilizamos las siguientes instrucciones:

- 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 (activado o desactivado).
- SetAO: fija el valor de una salida analógica.

La instrucción principal para comprobar el valor de una entrada es la instrucción WaitDI (WaitDI Pieza,1;), esta hace que el robot espere hasta que la señal alcance el valor deseado (en este caso espera un 1). También comprobamos el valor de una señal de entrada con instrucciones de comparación, IF señal = 1 THEN..., IF señal < 5 THEN...

Instrucciones de control de flujo de ejecución

Son instrucciones utilizadas en otros lenguajes de programación:

- IF THEN: ejecuta una serie de instrucciones si se cumple una determinada condición.
- FOR: repite una sección del programa un determinado número de veces.
- WHILE: repite una sección del programa mientras se cumpla una condición dada.
- TEST/CASE: ejecuta diferentes instrucciones en función del valor de un dato (similar al switch/case del lenguaje C).
- GOTO: salto incondicional a un punto del programa.

Variables y expresiones

El lenguaje RAPID permite definir variables o datos de distintos tipos. Con estas variables es posible crear expresiones aritméticas o lógicas mediante una serie de operadores comunes (suma, producto, comparación...).

Los principales tipos de datos en RAPID son ConfData, JointTarget, LoadData, MotSetData, Num, Orient, Pos, Robjoint, RobTarget, StopPointData, ToolData, ZoneData.

Existen dos modos de introducir programas al IRB140, una es generando el programa directamente a través de la paleta de programación y la segunda es escribir el programa en un PC y transferirlo al controlador.

Hemos realizado cambios en la instrucción de zone, haciendo que tenga mayor precisión como se puede ver en la figura 3.39, para ello hemos utilizado "fine", así puede llegar hasta la esquina de forma sin tener errores.

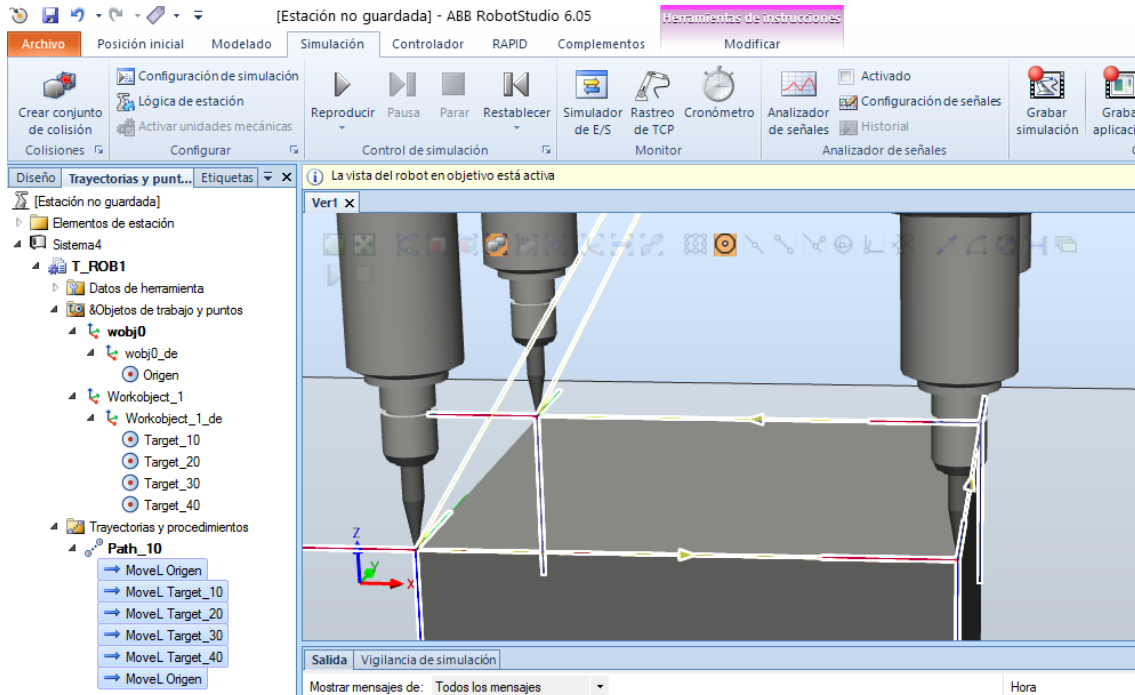


Figura 3.39: Mejora de la precisión en la trayectoria

Hemos vuelto a sincronizar con Rapid y en el código de la figura 3.40 se puede apreciar que en los MoveL en lugar de aparecer z100 ahora aparece fine.

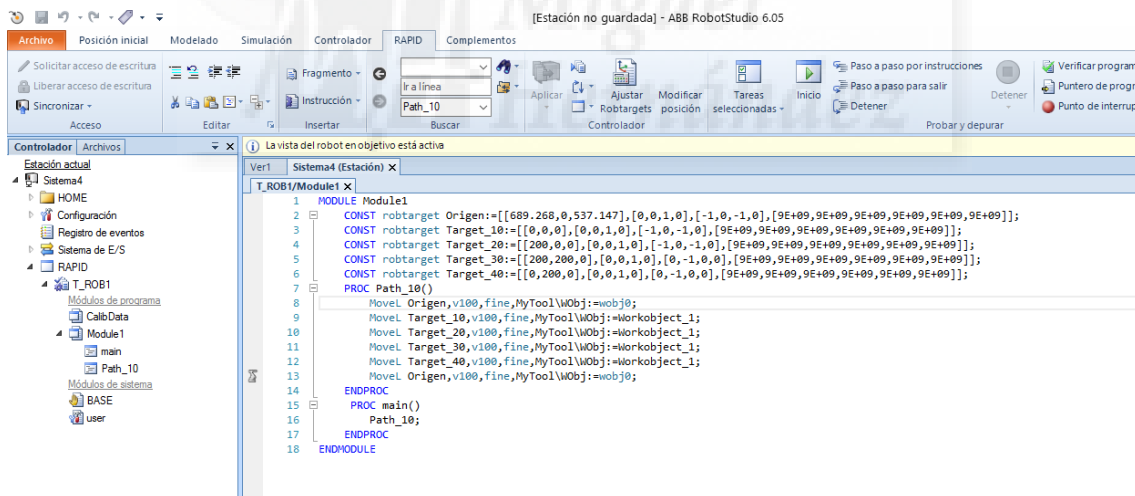


Figura 3.40: Cambios en RAPID

3.2. Diseño y Programación de una Estación

3.2.1 Diseño en Inventor de la geometría de la estación

En primer lugar, para el diseño de la estación se han de crear los sólidos que la componen, pero el programa RobotStudio no es suficiente ni a la hora de crear un sólido de diseño ni con los objetos que ya vienen incluidos en el programa, cintas transportadoras, mesas, vallas...

Entonces para la realización de este trabajo se ha utilizado el software de diseño Inventor 2015, realizando el diseño de las piezas necesarias. El conjunto de planos de las piezas se encuentran en el Anexo X de este trabajo.

Para adjuntar la pieza en el programa de RobotStudio, lo primero que hay que realizar es la exportación desde el programa Inventor, en la figura 3.41 en el Icono de Inventor, “Exportar” como “Formato de CAD” y nos pedirá el directorio donde exportarlo, a continuación se ha de guardar el archivo como tipo ACIS (.sat), ya que es el formato admitido por Robotstudio.

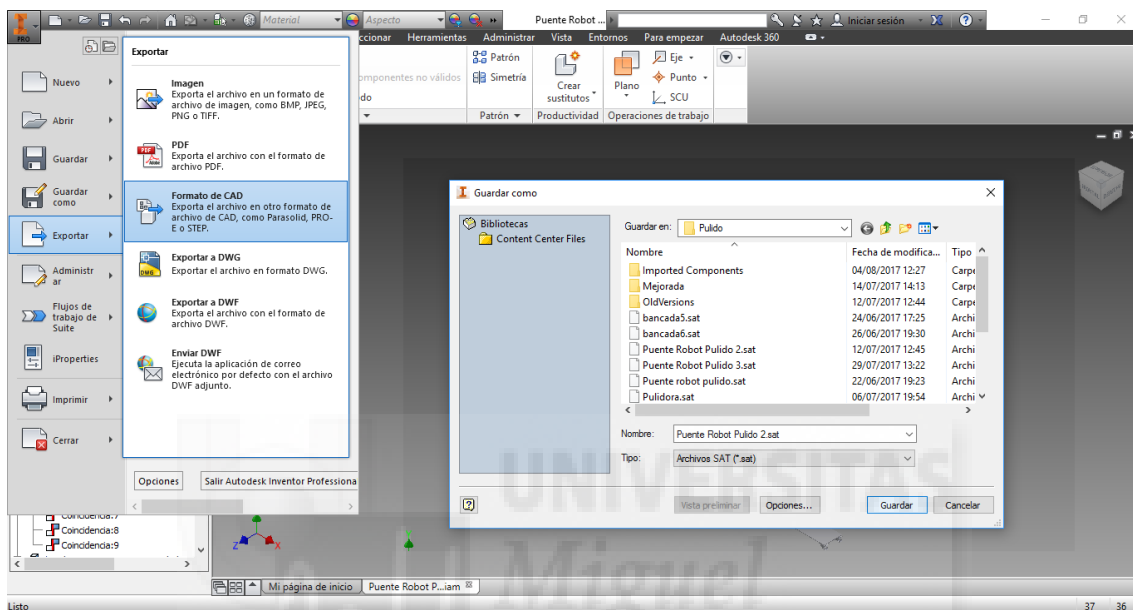


Figura 3.41: Exportar sólido desde Inventor

Una vez guardado, nos vamos al programa RobotStudio y en la pestaña “Posición Inicial”, seleccionamos “Importar Geometría”, hacemos clic en “Buscar geometría” y entonces buscamos el archivo .sat que queremos importar y pulsar en Abrir como en la figura 3.42. Solo se permite exportar sólidos y superficies.

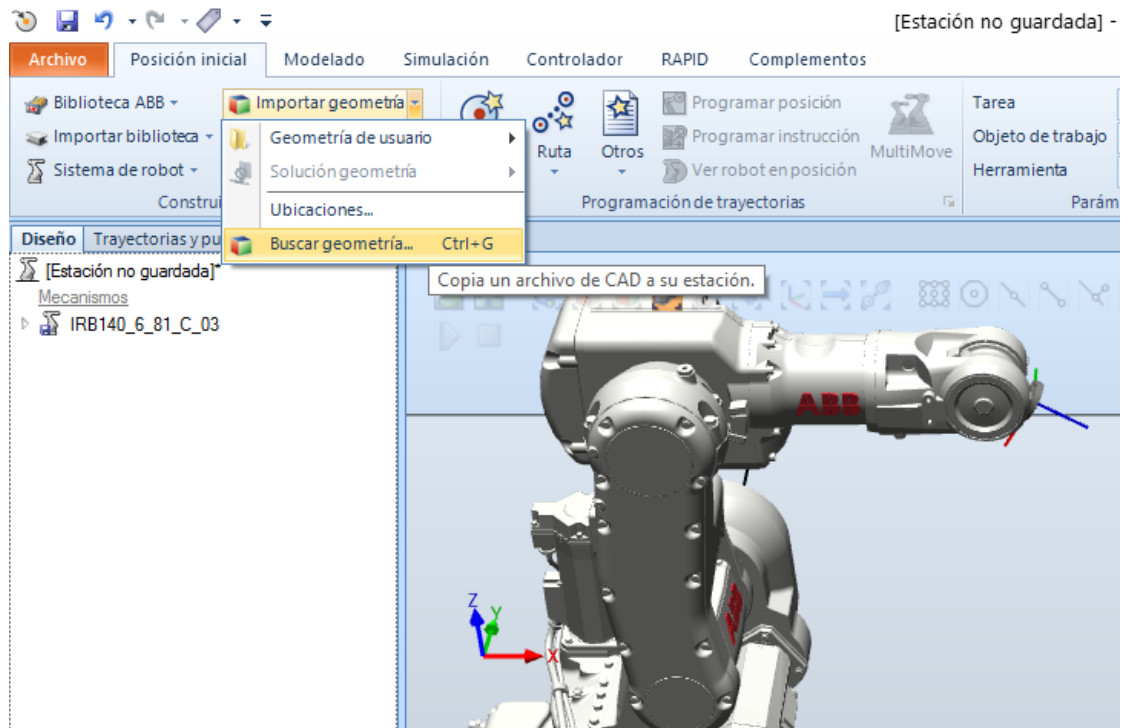


Figura 3.42: Importar geometría desde RobotStudio

3.2.2 Creación y diseño de los Smart Components

Los Componentes Inteligentes o Smart Components (SC) son elementos asociados a los sólidos, piezas o robots de la estación, los cuales tienen un comportamiento controlado por señales y propiedades del sistema.

Se pueden añadir en la pestaña “Modelado” y clicando en “Componente Inteligente” tal y como parece en la figura 3.43.

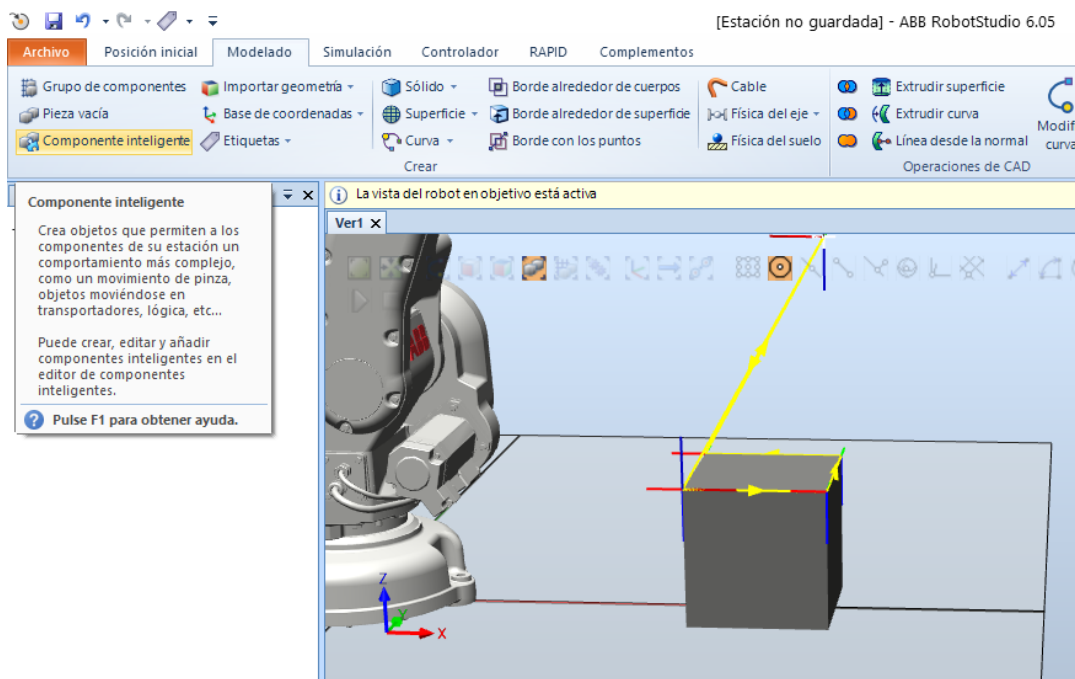


Figura 3.43: Crear componente inteligente (SC)

Los Smart Components podemos agruparlos en 6 categorías diferentes que aparecen en la figura 3.44: Señales y propiedades, primitivos paramétricos, sensores, acciones, manipuladores y otros.

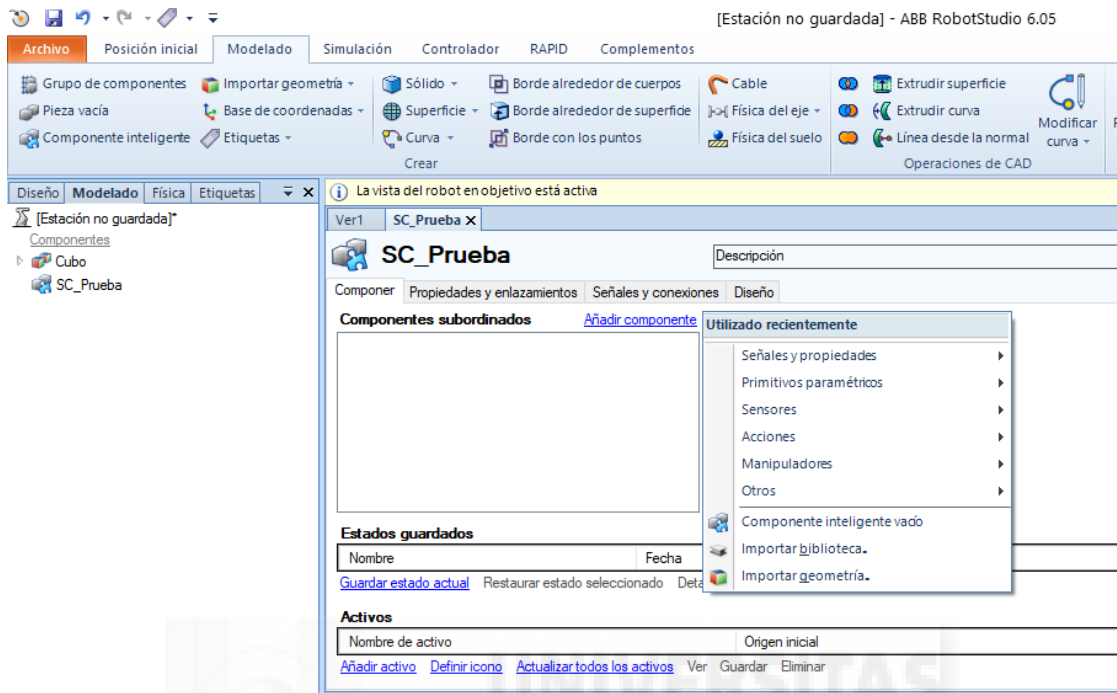


Figura 3.44: Categorías SC

Señales y propiedades: En esta categoría se pueden encontrar puertas lógicas, multiplexores, contadores, temporizadores, expresiones matemáticas, convertidores y demás elementos para modificar una señal o una propiedad del sistema a programar. Figura 3.45.

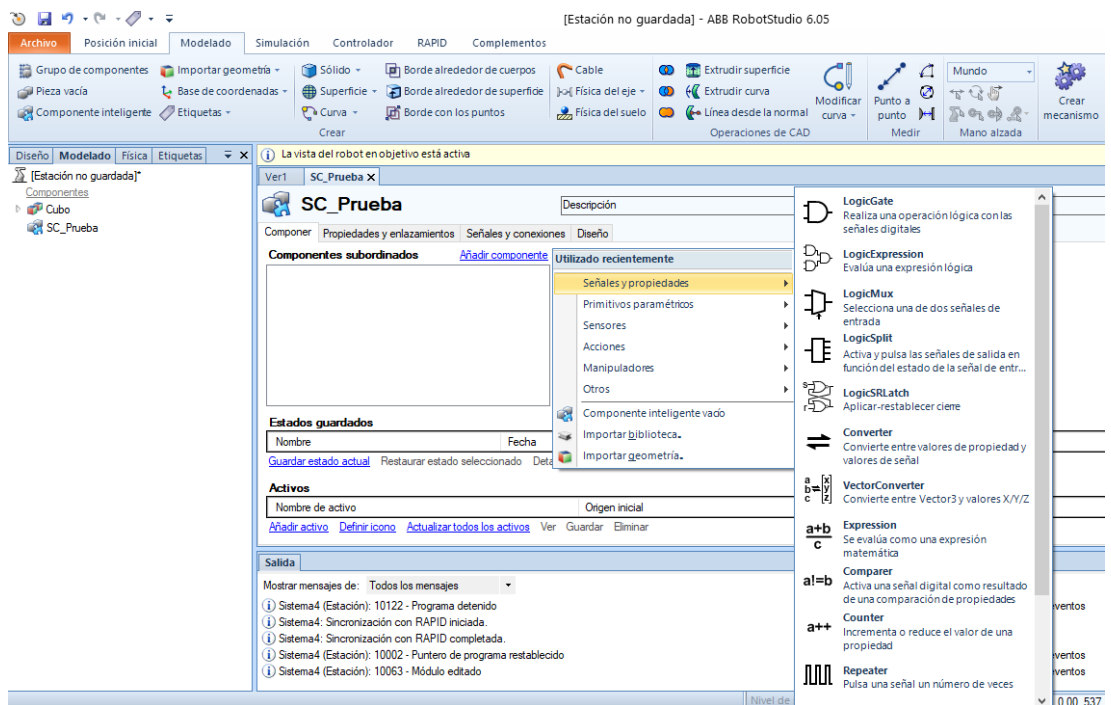


Figura 3.45: Señales y propiedades en SC

Primitivos paramétricos: En este apartado existen los componentes necesarios para crear de forma automática sólidos y líneas, así como para generar copias de componentes gráficos ya existentes. Figura 3.46.

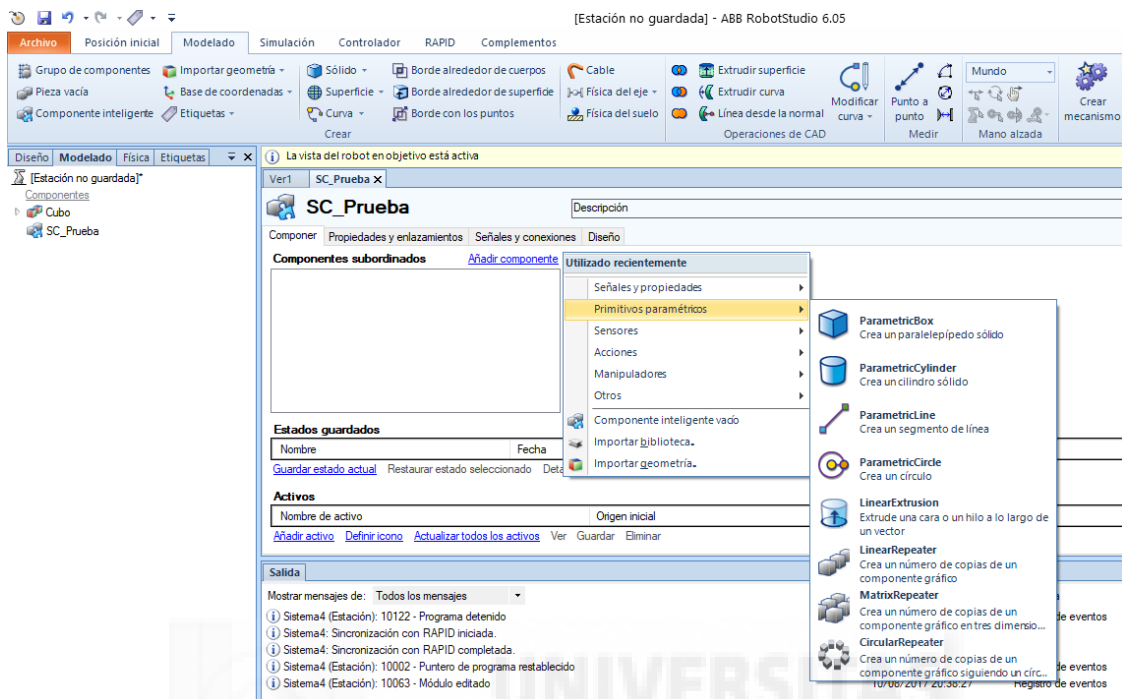


Figura 3.46: Primitivos paramétricos en SC

Sensores: Esta categoría reúne el conjunto de sensores a utilizar en los procesos de producción automático de este trabajo. Incluye sensores de colisión, de línea, de superficie, volumétricos, etc. Figura 3.47.

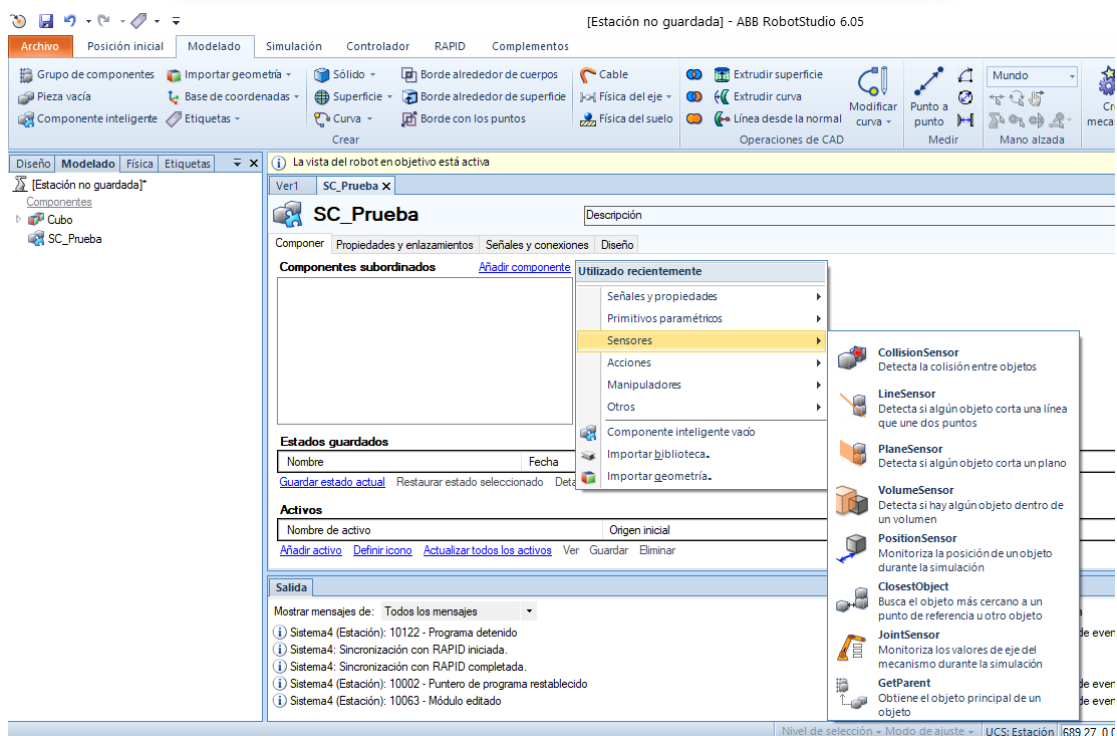


Figura 3.47: Sensores en SC

Acciones: Hace referencia a componentes como conectar o desconectar dos objetos entre sí, eliminar un objeto o simplemente hacerlo visible/invisible. Figura 3.48.

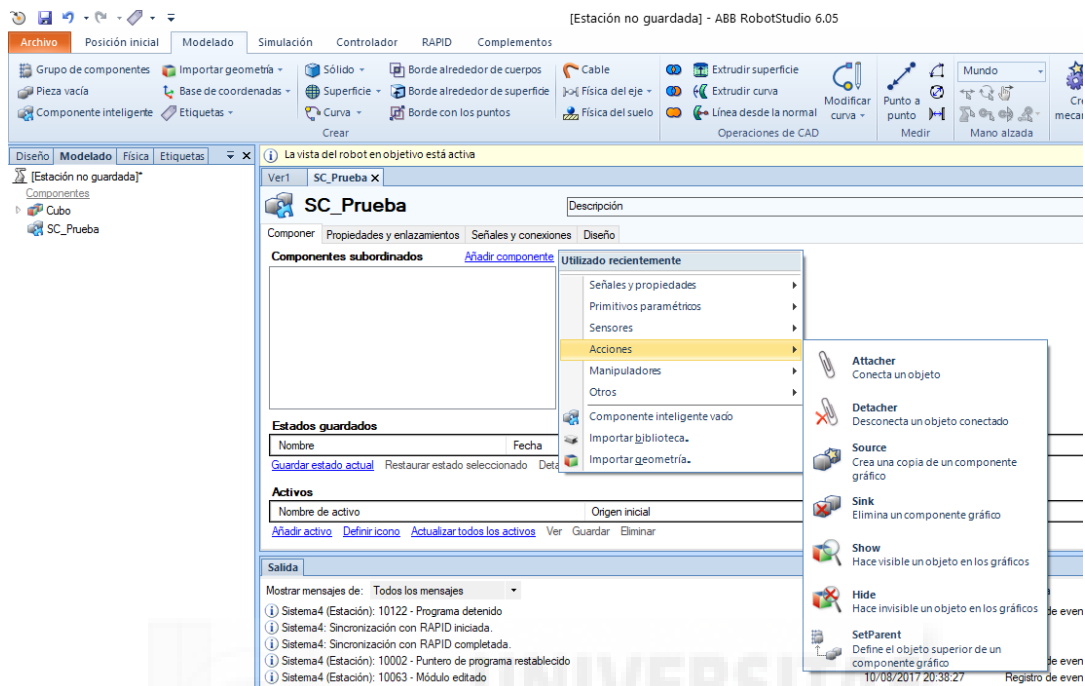


Figura 3.48: Acciones en SC

Manipuladores: Este apartado reúne los componentes necesarios para mover un objeto de forma lineal, hacerlo rotar un ángulo dado, moverse a lo largo de una curva o posicionarlo en un lugar determinado. También permite mover los ejes del mecanismo de un brazo robótico a una posición elegida. Figura 3.49.

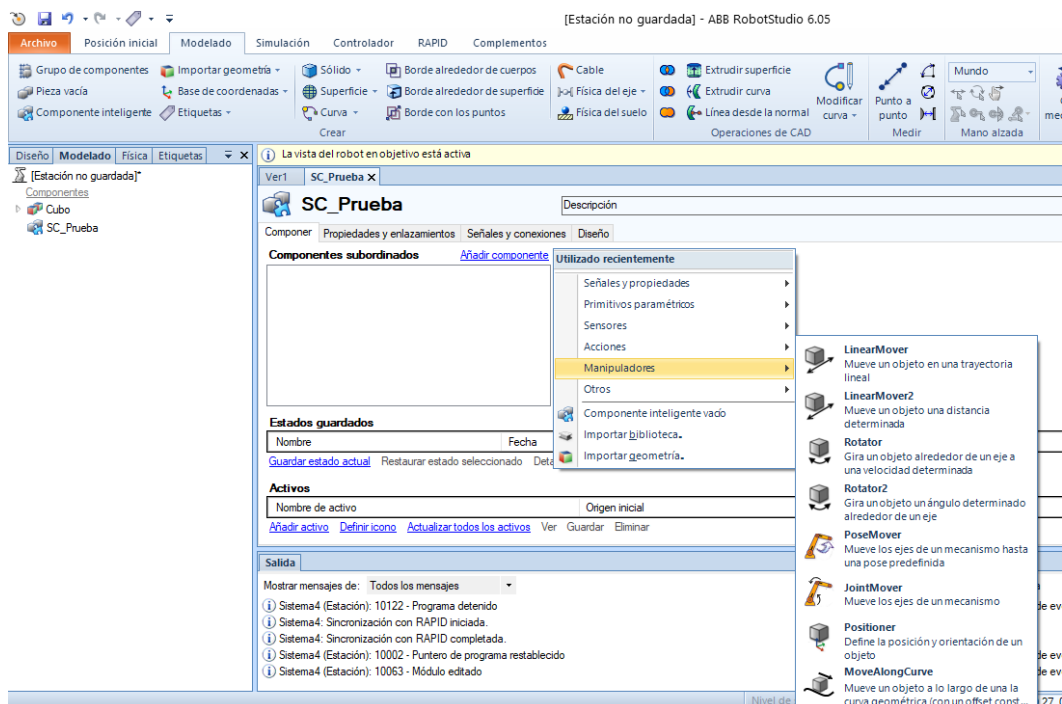


Figura 3.49: Manipuladores en SC

Otros: En esta última categoría se encuentra un conjunto de SC de distinta variedad. Representación de una cola de objetos, generación de un número aleatorio, detención de la simulación, reproducción de un sonido, etc. Figura 3.50.

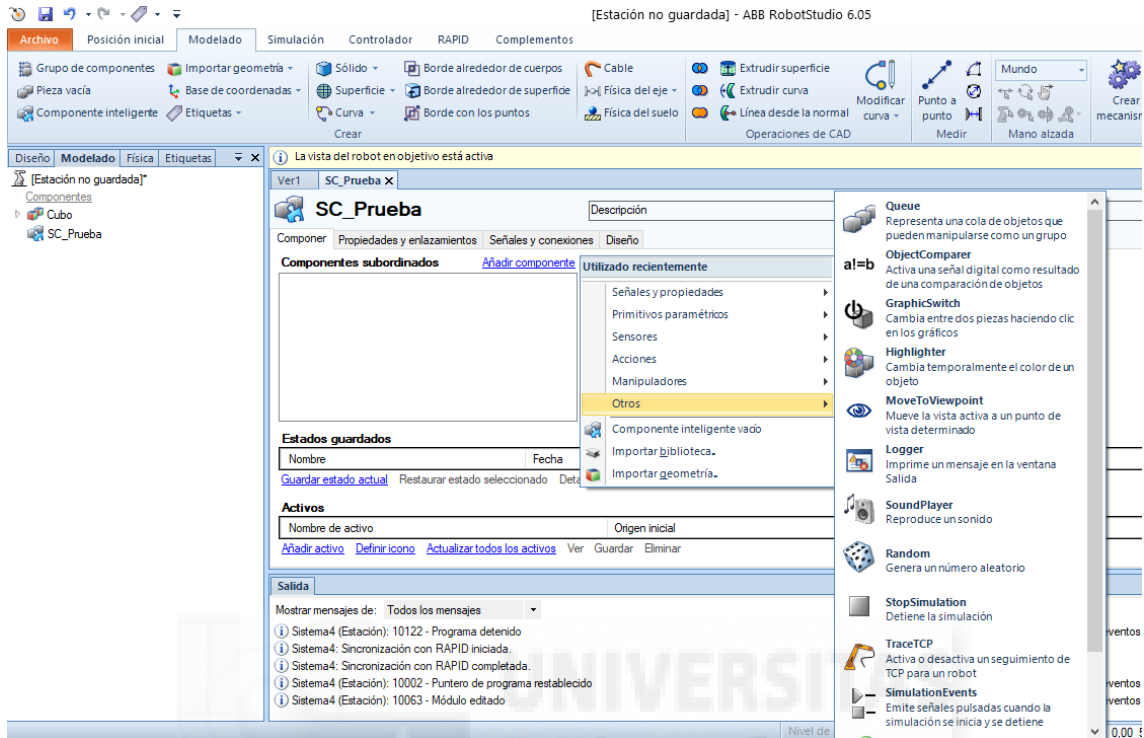


Figura 3.50: Otros en SC

Dentro del SC de la figura 3.51, en la pestaña de Señales y conexiones, podemos añadir señales de entrada o salida correspondientes a este componente, además podemos realizar las conexiones entre diferentes objetos.

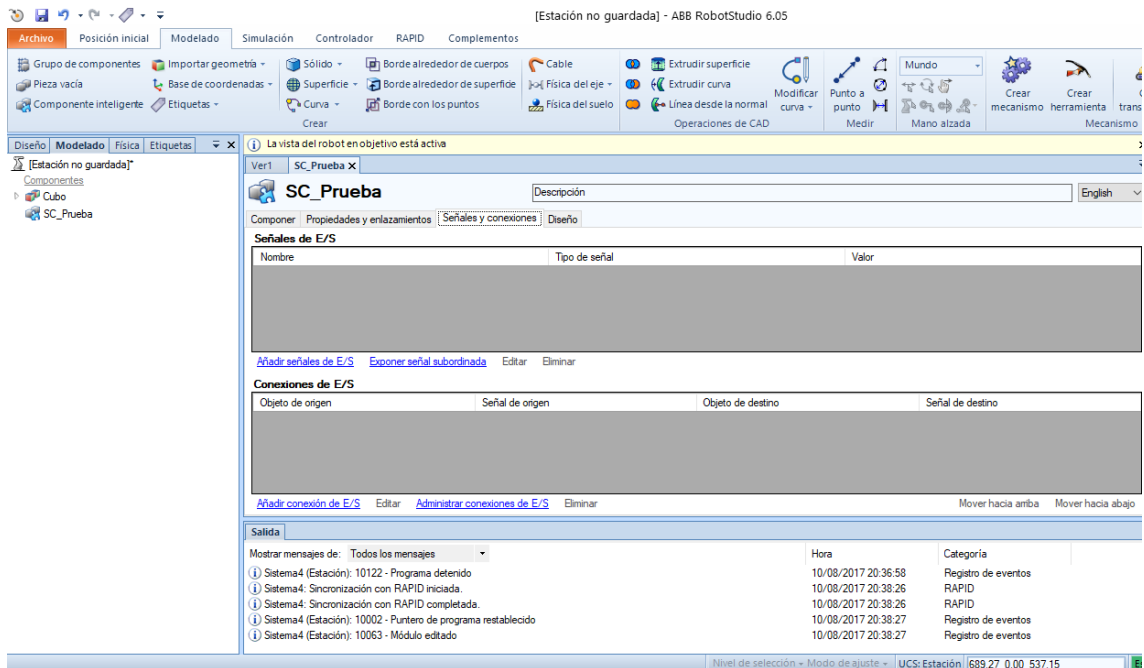


Figura 3.51: Señales y conexiones del SC

En la pestaña de “Diseño” de la figura 3.52 podremos ordenar los objetos, cambiar su configuración, añadir entradas/salidas y conectar mediante flechas los diferentes objetos.

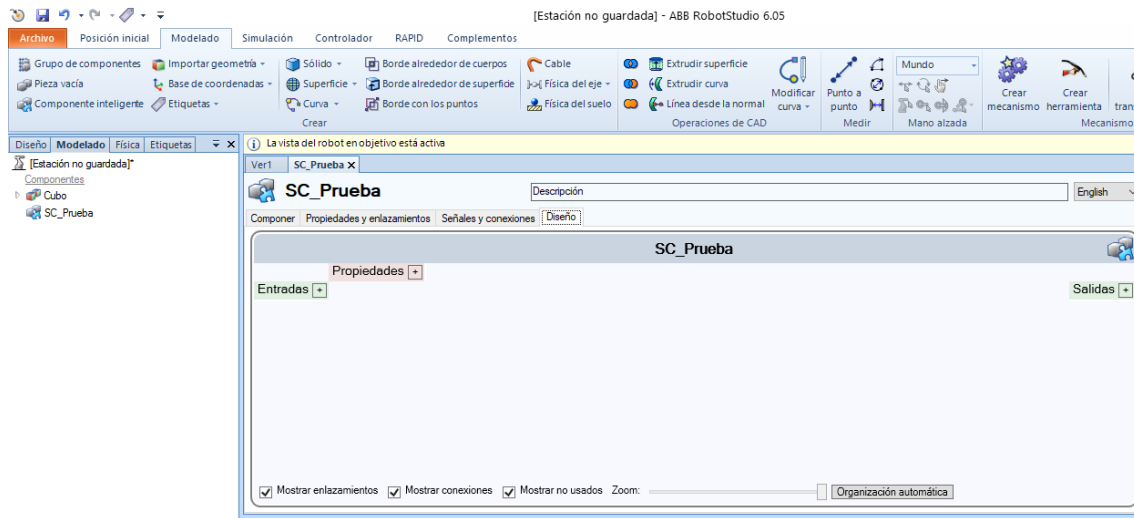


Figura 3.52: Diseño del SC

3.2.3 Creación de señales E/S

Tras crear la geometría de la estación, el siguiente paso es “atribuirle inteligencia” al robot que compone la estación.

El sistema de E/S maneja las señales de entrada y salida intercambiadas con el controlador.

Podemos añadir una señal en la figura 3.53 desde la pestaña “Controlador”, seleccionando en “Editor de configuración” y haciendo clic en “Añadir señales”.

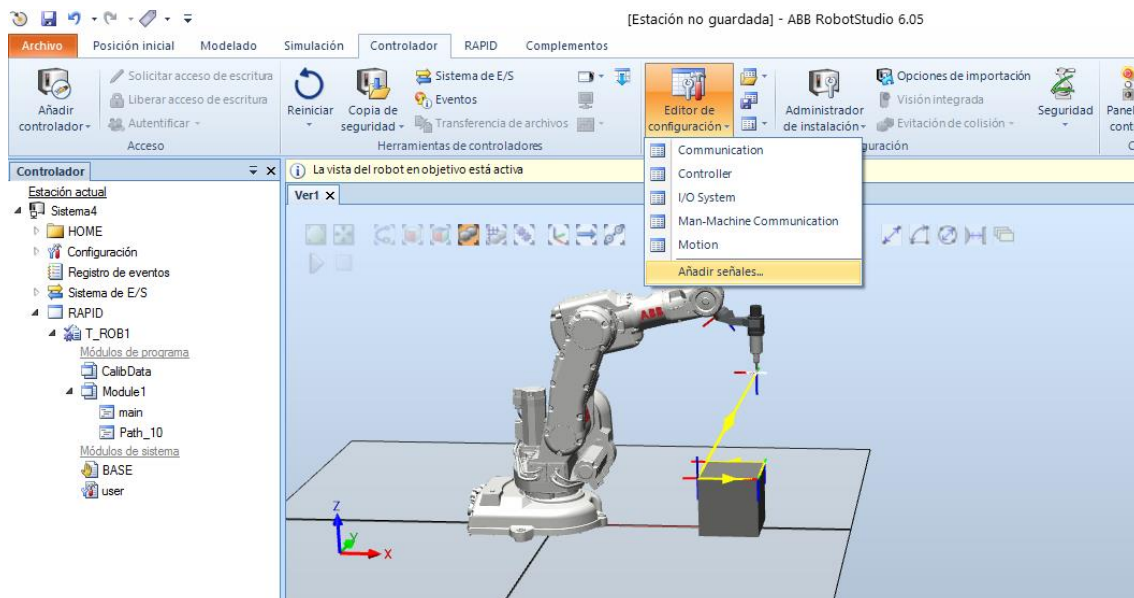


Figura 3.53: Añadir señales

Para añadir la señal, tendremos que decirle que tipo de señal es, normalmente añadiremos señales de entrada digital o salida digital, le indicamos un nombre a la señal, le asignamos a ningún dispositivo, seleccionar el cuadrado de Avanzado, como nivel de acceso indicamos All, finalmente le asignamos un valor predeterminado (0 o 1) y pulsamos en Aceptar como la figura 3.54.

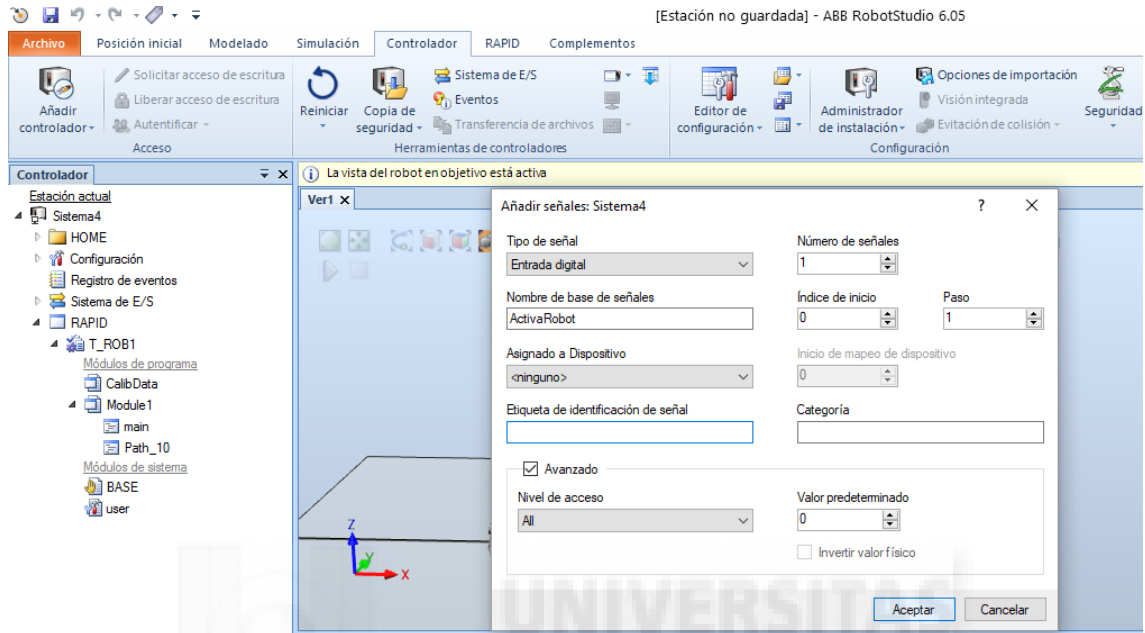


Figura 3.54: Añadir una señal de un controlador

Es necesario reiniciar el controlador para que los cambios realizados se guarden y sincronicen con el controlador, esto se realiza en la figura 3.55 en la pestaña “Controlador” y se hace clic en “Reiniciar”.

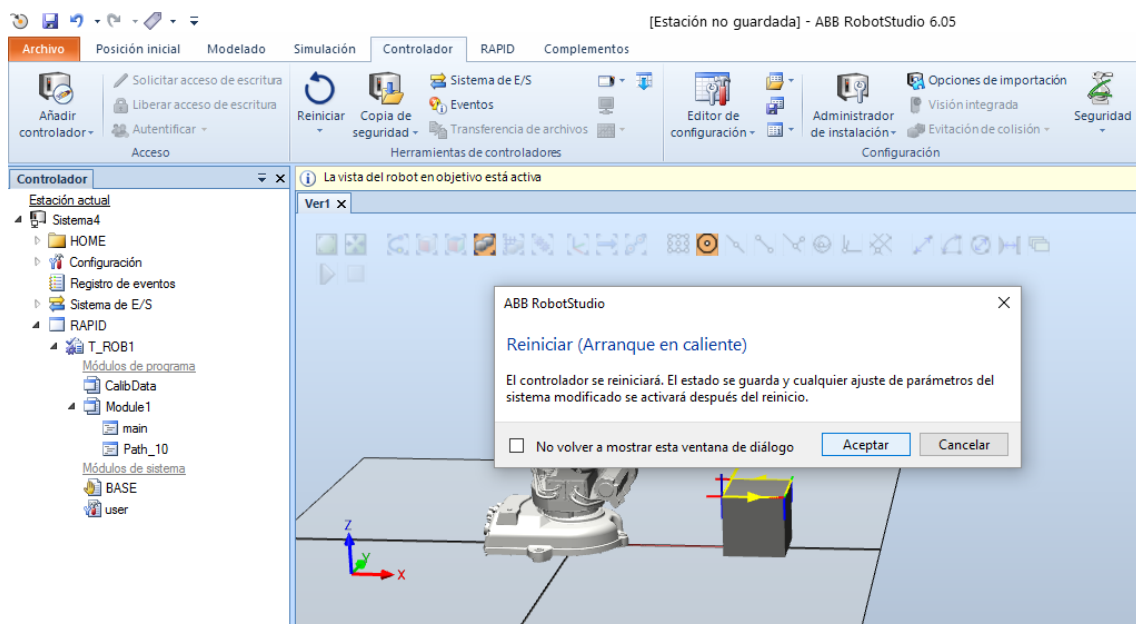


Figura 3.55: Reinicio del controlador

Otra forma de añadir señales es en la misma pestaña del “Controlador” de la figura 3.56, en la ventana de la izquierda, dentro de “Sistema 4”, “Configuración”, hacemos clic en “I/O System”. Para añadir una nueva señal haciendo clic en el botón derecho en “Signal” y pinchamos en “Nuevo Signal”. También es posible editar, copiar o eliminar las señales anteriormente creadas.

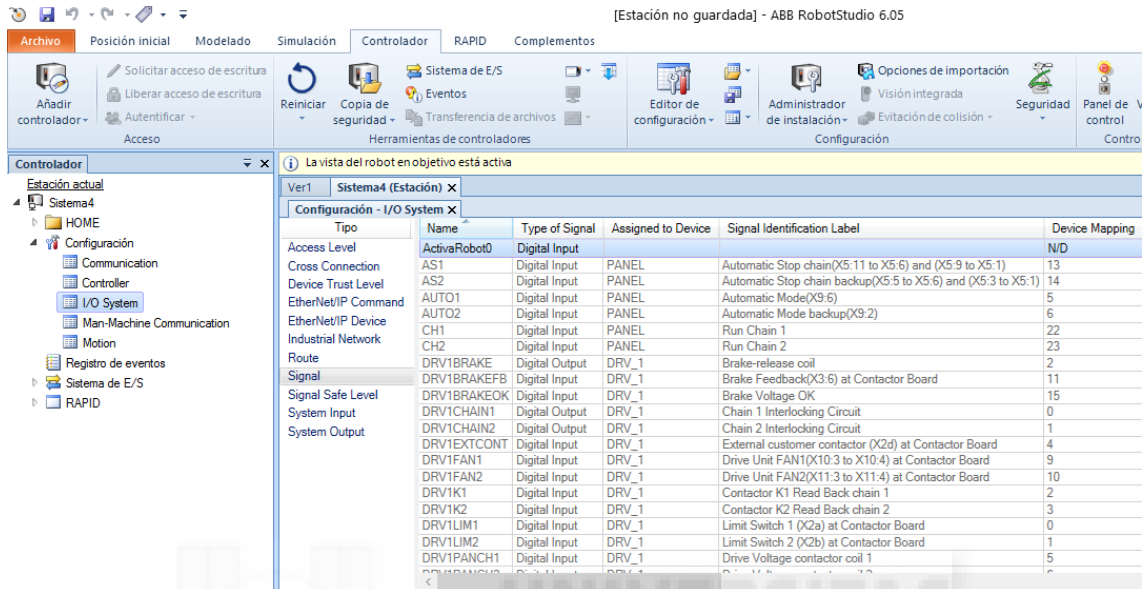


Figura 3.56: Configuración de una señal

Se crearán un conjunto de señales de entrada digitales del controlador que corresponderán a las señales de salida de los SC que sean necesarias para el control de algún robot. También se establecerán un conjunto de señales de salida del controlador correspondientes a señales de entrada de los SC cuyo control provenga de alguno de los robots, en vez de un controlador externo ajeno al sistema automatizado.

En el Anexo 8.2 se encuentra el listado de entradas y salidas de este trabajo.

3.2.4 Lógica de estación

Tras crear el controlador del robot y los SC es necesario de forma adecuada interconectar estos entre sí, paso indispensable para que el sistema coordinando los movimientos automáticos de los robots y de los sólidos inteligentes funcionen correctamente. Para ello, hay que acceder a la pestaña “Simulación” y dentro de ésta en “Lógica de estación”.

La lógica de la estación está formado por un conjunto de bloques correspondientes a SC y el controlador anteriormente creado. Además, se pueden añadir otros componentes inteligentes subordinados como los vistos en el diseño de los SC, aunque en este sistema eso no será necesario. Para poder observar dicho esquema de bloques una vez abierta la lógica de estación, al igual que con los SC, hay que pulsar sobre la pestaña “Diseño”, como en la figura 3.57.

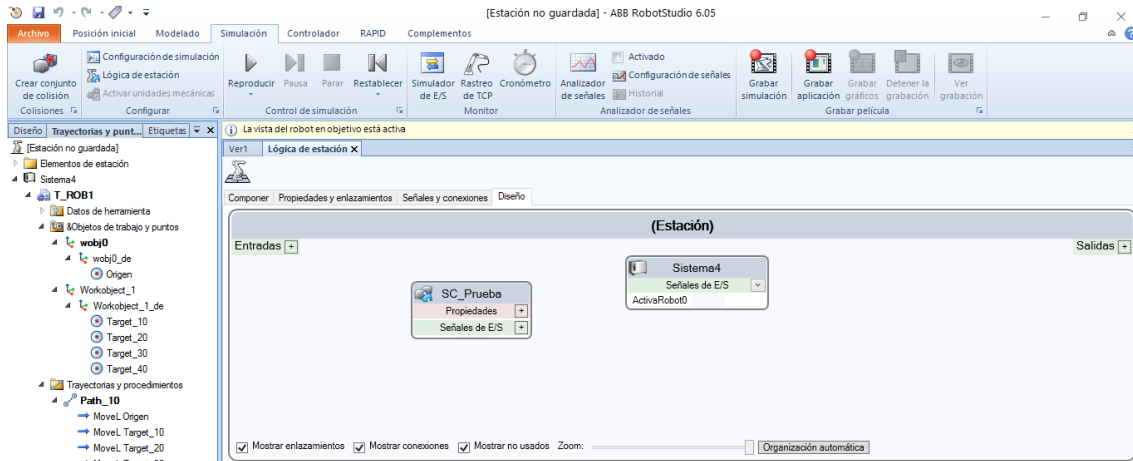


Figura 3.57: Lógica de una estación

3.3 Simulación de una Estación

Cuando hemos finalizado la programación de la estación, solo nos queda realizar la simulación para así observar el funcionamiento del sistema automatizado que hemos creado. En la figura 3.58 si queremos observar su paso a paso por el programa RAPID, desde la pestaña RAPID pulsaremos en “Verificar programa”, pinchamos en “Aplicar” para que se guarden los cambios dándole a sí y finalmente clic en “Inicio” para comenzar la simulación.

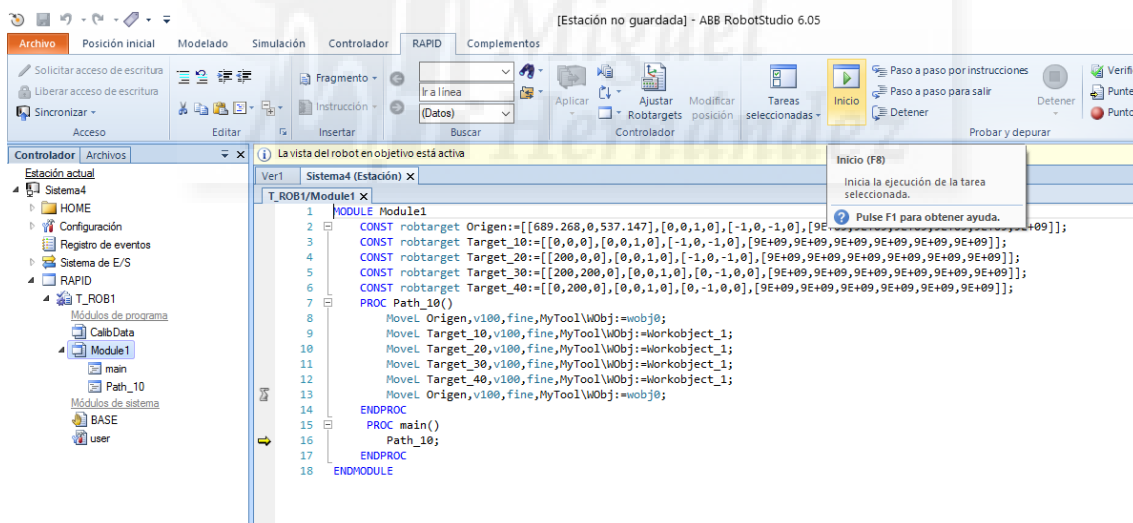


Figura 3.58: Compilación de un programa RAPID e inicio de la simulación

Para realizar la simulación podemos configurarlo de varios modos, esto se realiza en la pestaña “Simulación”, haciendo clic en “Configuración de simulación” y nos saldrá un panel con las configuraciones de la figura 3.59.

Si pinchamos sobre Sistema 4, que es nuestro controlador, nos aparece unos ajustes que si queremos que se inicie automáticamente la ejecución de las tareas al comenzar la simulación o no. Y también podremos seleccionar el modo de ejecución, si la tarea queremos que se realice una vez seleccionamos “Un solo

ciclo” o si no escogemos “Continuo”. En el tutorial se ha utilizado Un solo ciclo y en el trabajo hemos utilizado Continuo.

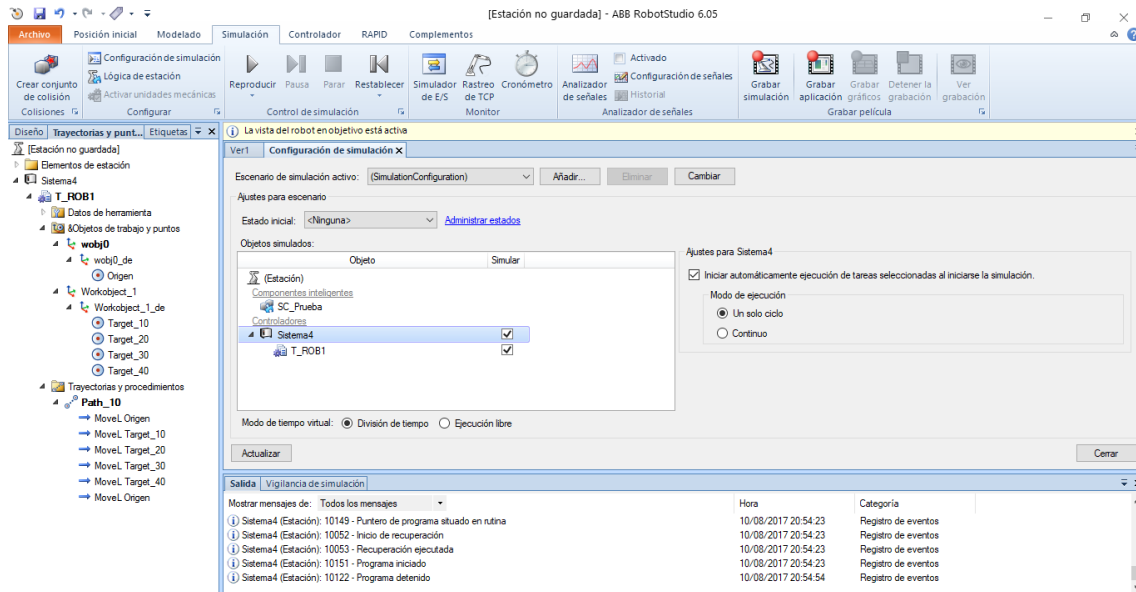


Figura 3.59: Configuración de la simulación, ajustes de controlador

Otra opción que podemos cambiar de la simulación son los ajustes de T_ROB1 y podremos indicar que solamente haga una trayectoria seleccionando “Path_10” o que realice el programa principal indicando “main” como ocurre en la figura 3.60, esta última es la que hemos utilizado.

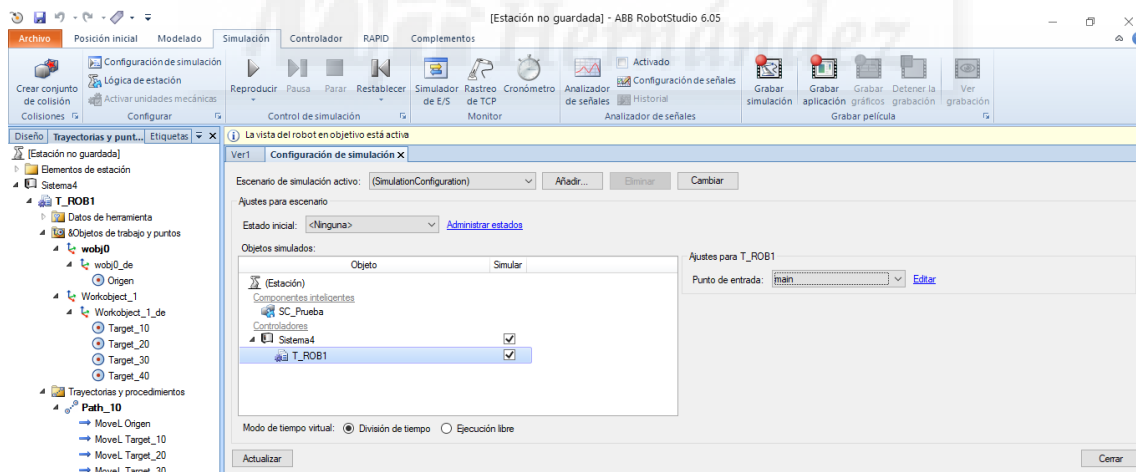


Figura 3.60: Configuración de la simulación, ajustes del programa

Además podremos cambiar las opciones de grabación de simulación en “Archivo”, “Opciones” y “Grabadora de pantalla”.

En la figura 3.61, para reproducir la simulación en la pestaña “Simulación” hay que darle a “Reproducir”, se puede pausar la simulación y si se va clicando varias veces en “Pausar” se moverá todo lentamente, otra opción es la de “Parar” la simulación y de “Restablecer” para así si se ha movido los objetos o el robot vuelvan al estado inicial de la simulación.

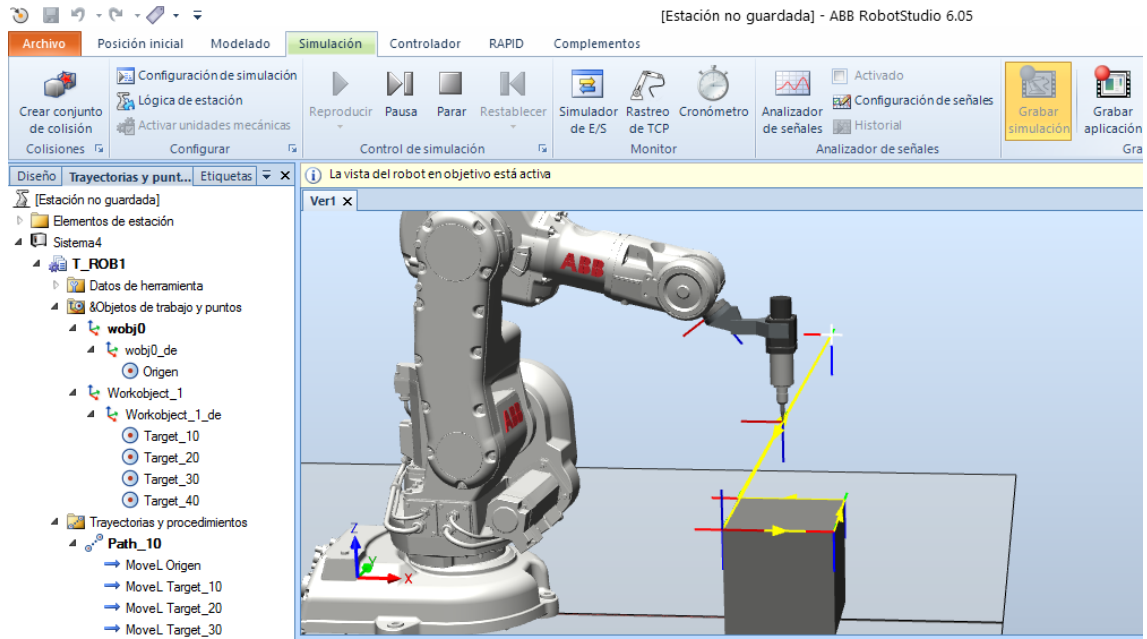


Figura 3.61: Simulación y grabación



4. IMPLEMENTACIÓN A UN PROCESO DE PULIDO EN UNA LÍNEA DE PRODUCCIÓN DE AUTOMÓVILES

4.1. Estación de Carrocerías

4.1.1 Elección del Robot

En la estación de pulimentado de las imperfecciones de las carrocerías se va a instalar un puente móvil que se desplazará a lo largo de los carriles instalados en el suelo. En este puente se situará el socket del robot en la parte superior, que irá incrustado en el propio puente y al que se le atornillará el robot escogido.

El robot debe tener una serie de características a cumplir: ser rápido para realizar este proceso, ser preciso para realizar el pulido en la zona donde se ha producido el defecto y poder ser montado sobre el puente.

Hemos escogido el robot IRB4600 porque cumple esta serie de características y principalmente porque es un robot utilizado en pulimiento de superficies.

Una vez elegido el modelo de robot podemos observar en la tabla 4.1, que tenemos el mismo robot pero con diferentes variantes de tamaño y de carga.

Variantes	Reach (m)	Payload (kg)	Armload (kg)
IRB 4600-60/2.05	2.05	60	20
IRB 4600-45/2.05	2.05	45	20
IRB 4600-40/2.55	2.55	40	20
IRB 4600-20/2.50	2.51	20	11

Tabla 4.1: Modelos de robot IRB4600

Dentro de los robot de la tabla anterior, se ha seleccionado el robot IRB 4600-20/2.50 de la figura 4.1, por tener un gran tamaño con el que se alcanza todas las partes de la carrocería y por ser el de menor capacidad de carga, ya que el robot no va a soportar cargas, sino solamente contará como carga el peso de la herramienta.

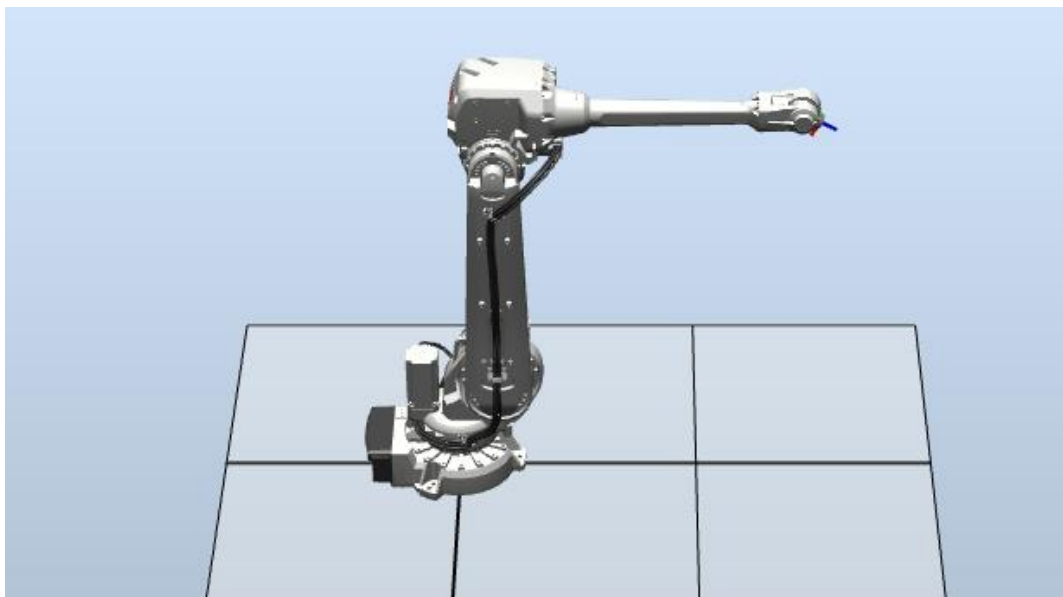


Figura 4.1: Robot ABB IRB 4600.

4.1.2 Elección de la Herramienta

Herramienta pulidora con acoplamiento adaptativo a robot mediante plato, arquitectura tipo tambor con actuación radial, disco compuesto por tejido resistente a la tracción y desgaste por rozamiento, con abrasivo fino de óxido de aluminio fijado al disco por medio de un aglutinante, mostrada en la figura 4.2.



Figura 4.2: Herramienta de pulido

Esta herramienta de mecánica pasiva posee muelles que se adaptan a cualquier superficie.

4.1.3 Diseño de los Sólidos de la Estación

Para el diseño de los sólidos de la estación hemos utilizado el software de CAD 3D Autodesk Inventor Profesional 2015, para ver los planos de estos objetos se puede consultar el anexo 8.1.1.

En primer lugar se ha diseñado el puente de la figura 4.3, soporte del robot, de una altura y anchura suficiente que permita el paso de las carrocerías por debajo sin peligro de colisión. Además, para dar estabilidad a la estructura son necesarios dos soportes móviles de adecuada longitud soldados a la base de los pilares metálicos con seis ruedas en cada soporte para su desplazamiento mediante railes.

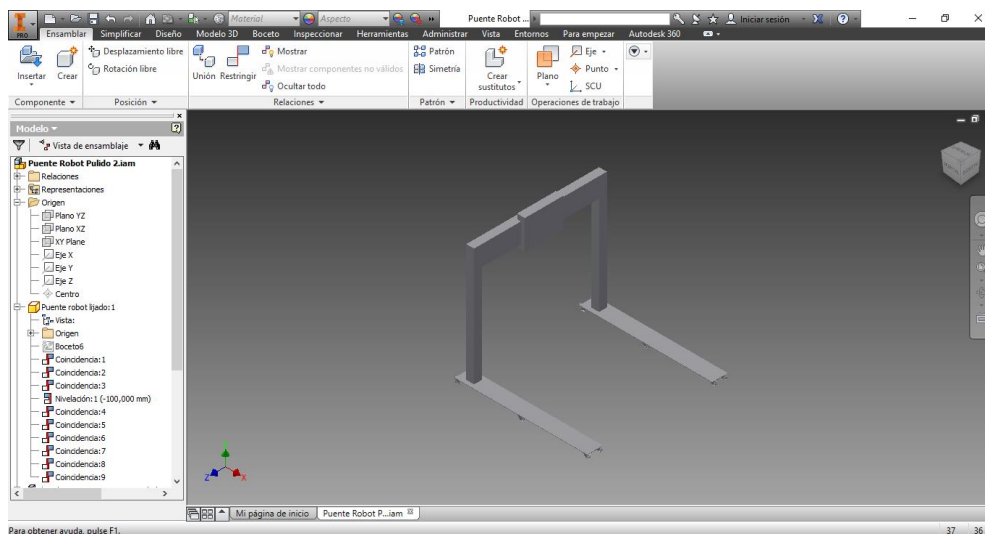


Figura 4.3: Puente móvil

El robot necesita para el pulido una herramienta adecuada que no ha sido posible encontrar en las bibliotecas de RobotStudio y otras bibliotecas de CAD Open Source, por tanto ha tenido que diseñar una como la mostrada en la figura 4.4, a partir de una pulidora dedicada a realizar tareas de pulido de superficies.

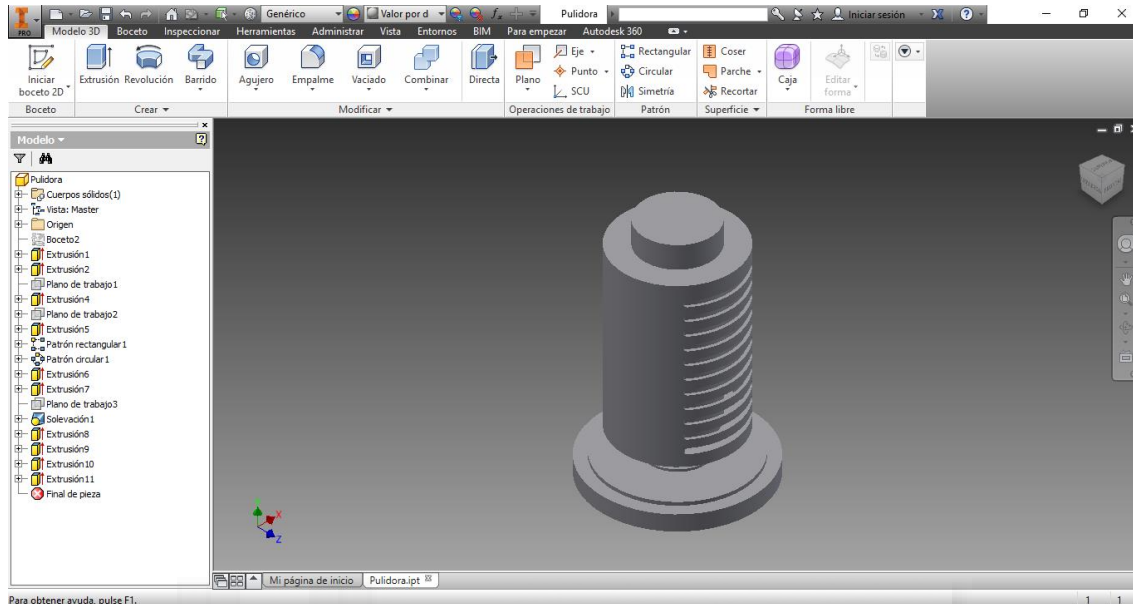


Figura 4.4: Herramienta pulidora

Ha sido necesario diseñar una bancada móvil con ruedas a base de perfiles metálicos y desplazamiento a través de railes como se muestra en la figura 4.5, donde se apoye las carrocerías de los vehículos.

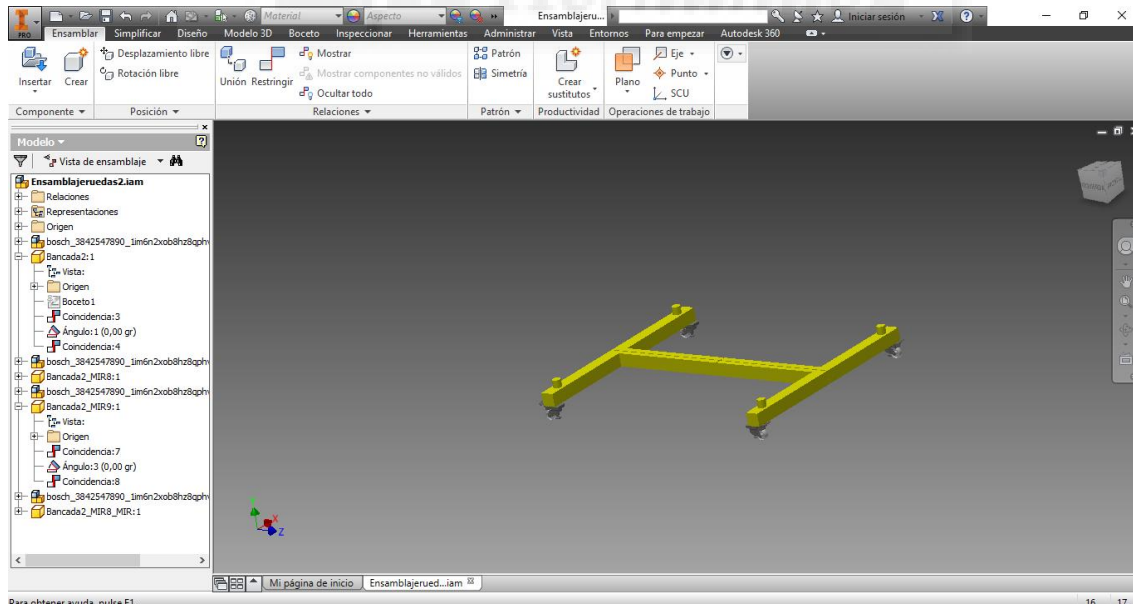


Figura 4.5: Bancada carrocerías

Para que las carrocerías y el puente móvil se desplacen desde el inicio al final, se diseñan dos railes metálicos incrustados en el suelo a ambos lados del eje

central por donde se desplazaran las bancadas junto a las carrocerías. Igualmente y para que se desplace el puente móvil otros dos raíles, con una longitud total de 55 metros. Los raíles se pueden observar en la figura 4.6.

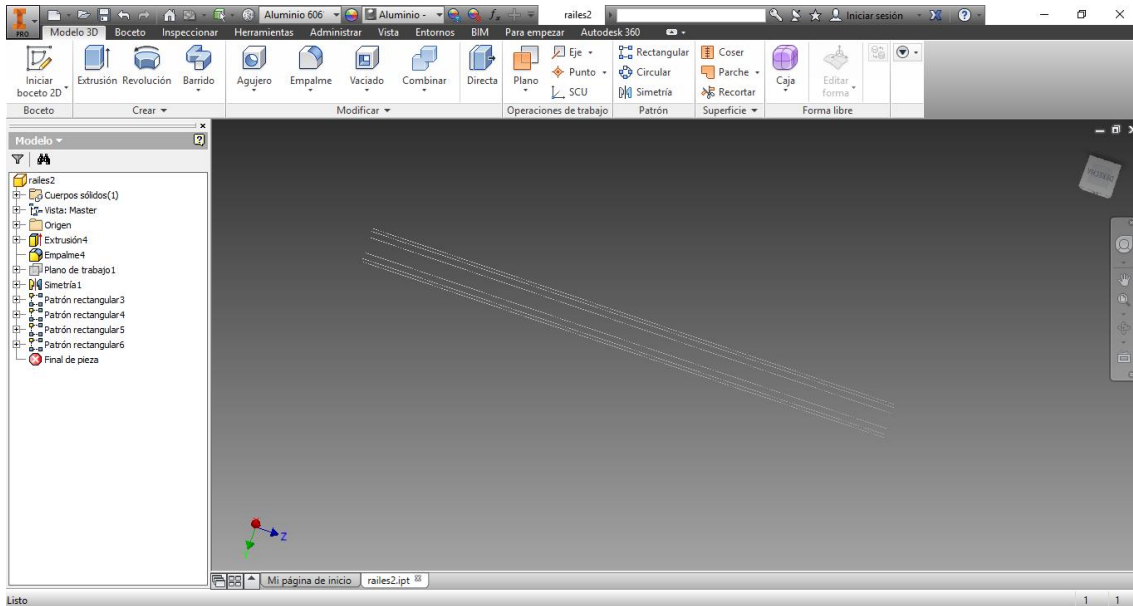


Figura 4.6: Raíles

Otros sólidos importados

Se han utilizado también otros sólidos que ya estaban diseñados y compartidos en librerías CAD Open Source y GrabCAD, se ha descargado en STL para incorporarlos a la estación.

En esta estación se ha importado la carrocería de la figura 4.7, de la marca Volkswagen modelo Golf, donde se realizará el pulido de imperfecciones.



Figura 4.7: Carrocería Volkswagen modelo Golf

Otros sólidos de la biblioteca RobotStudio

Para la unión de la muñeca del robot con la herramienta de pulido, vamos a utilizar un disco de plato como el de la figura 4.8.

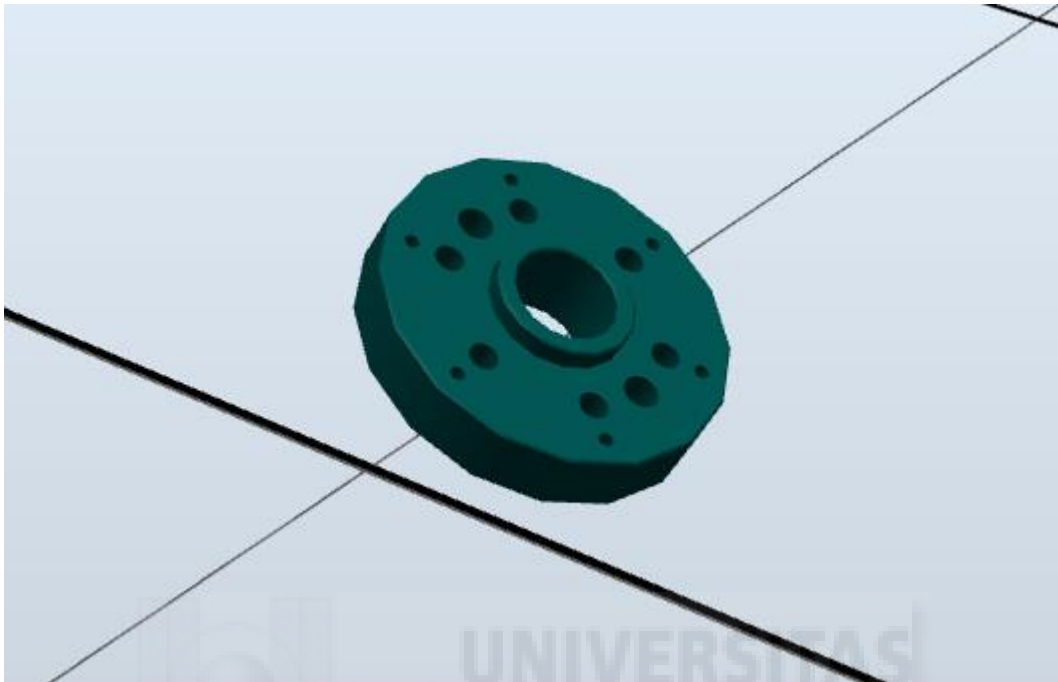


Figura 4.8: ISO Plate Large

Para la unión entre el puente móvil y el robot necesitamos una pieza o socket IRB4600 de la figura 4.9.

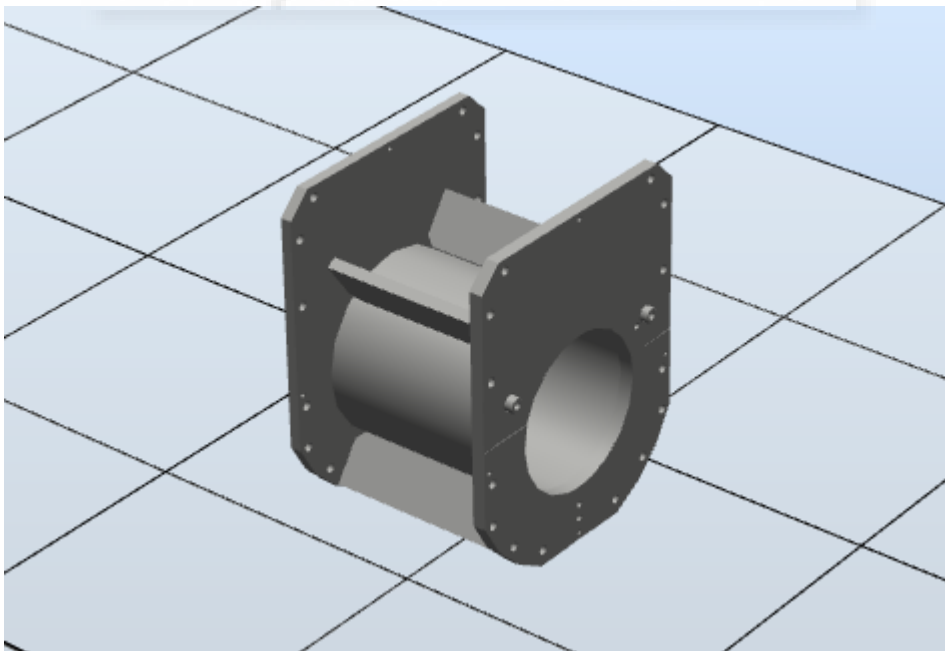


Figura 4.9: Socket IRB4600

Las vallas de la figura 4.10 se utilizan para delimitar un perímetro anti-intrusos.

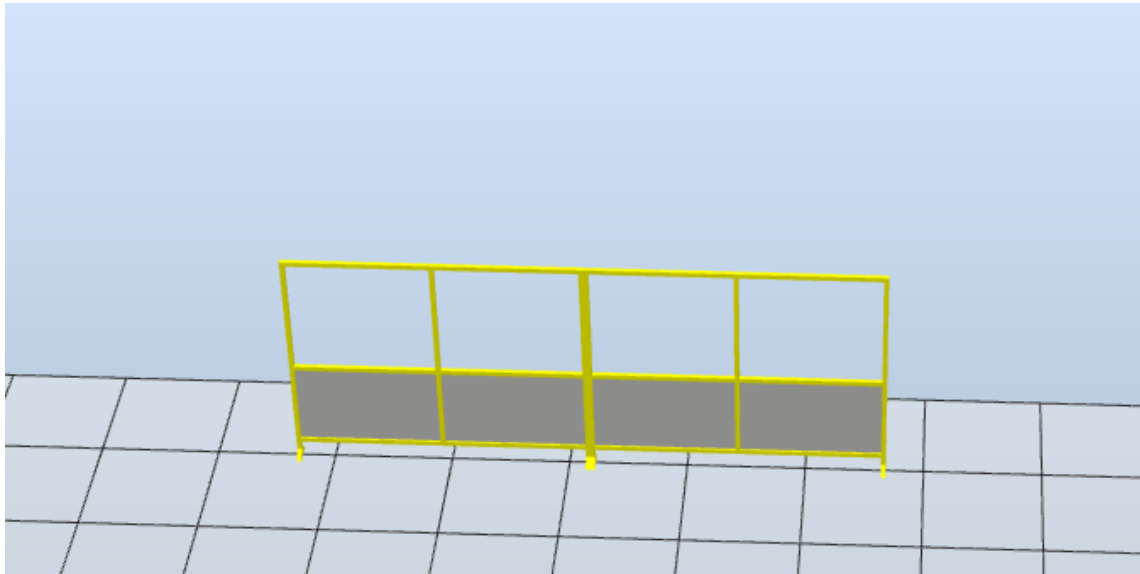


Figura 4.10: Vallas

La estación debe tener un controlador para el robot como el de la figura 4.11.

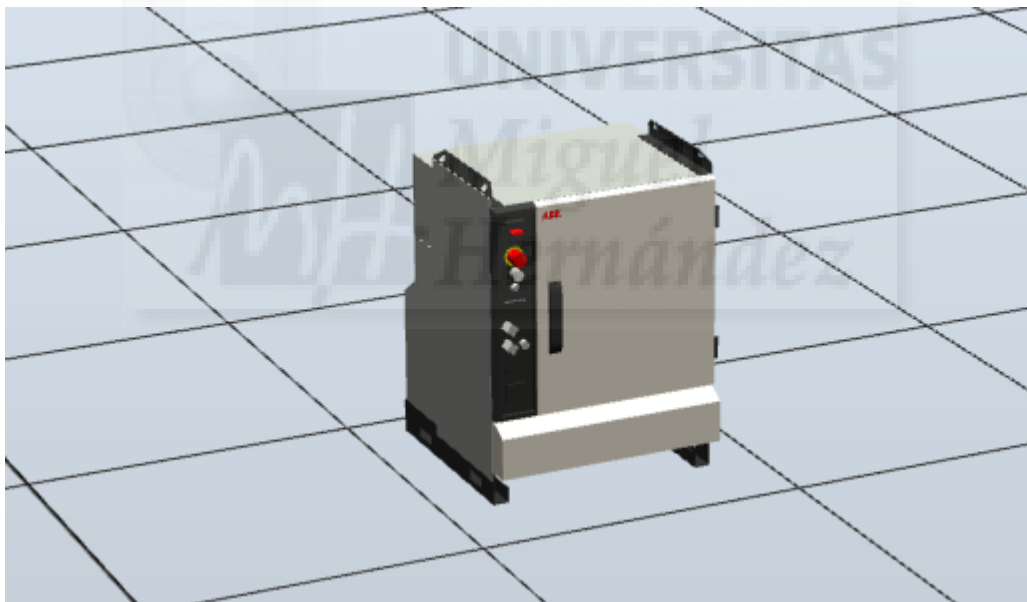


Figura 4.11: Controlador

El PLC de la figura 4.12 pertenece al puente móvil.

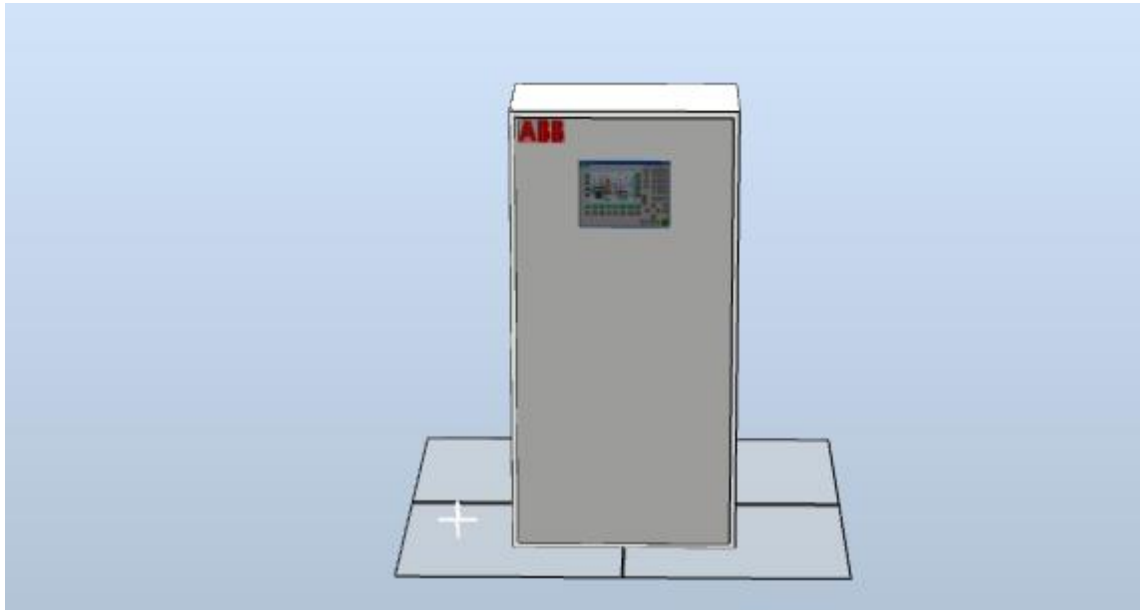


Figura 4.12: PLC

La mesa de la figura 4.13 es utilizada para colocar un ordenador controlador de la estación.

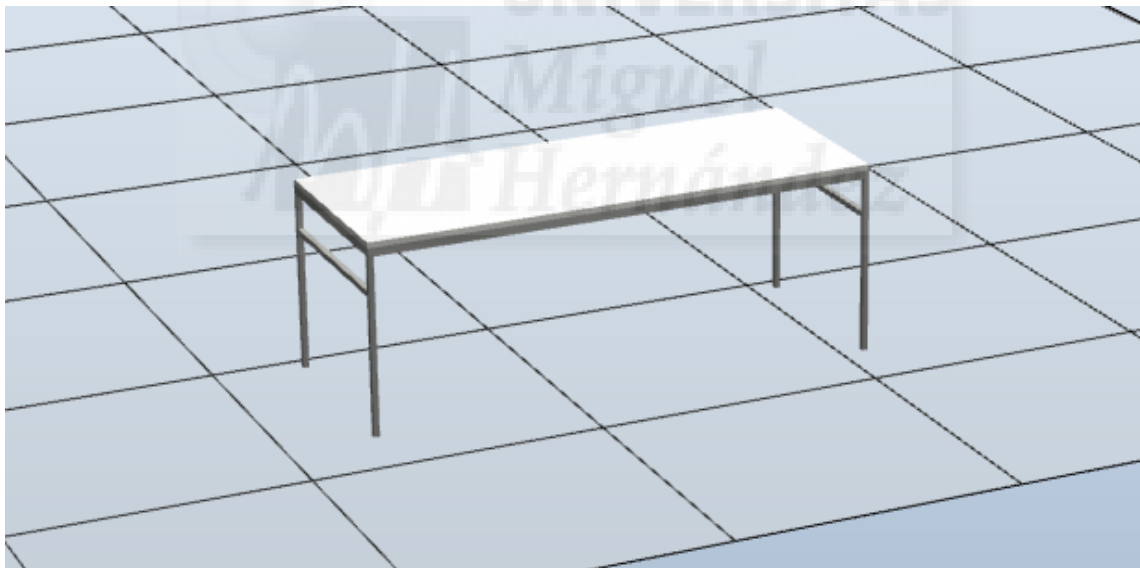


Figura 4.13: Mesa

En la figura 4.14, la silla puede ser utilizada el operario que esté a cargo del ordenador.

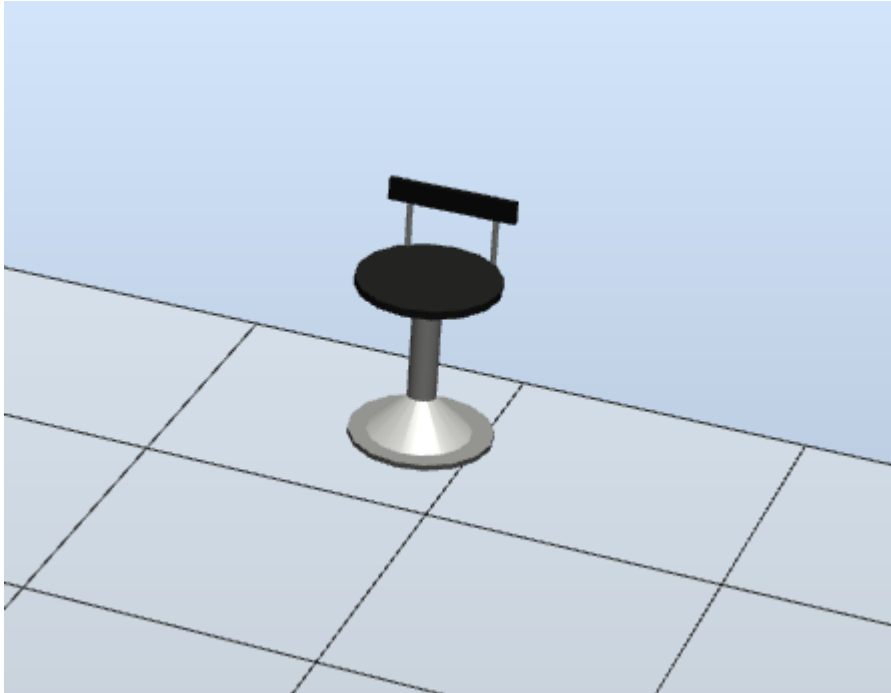


Figura 4.14: Silla

Para el control de la estación es necesario un ordenador como la figura 4.15.

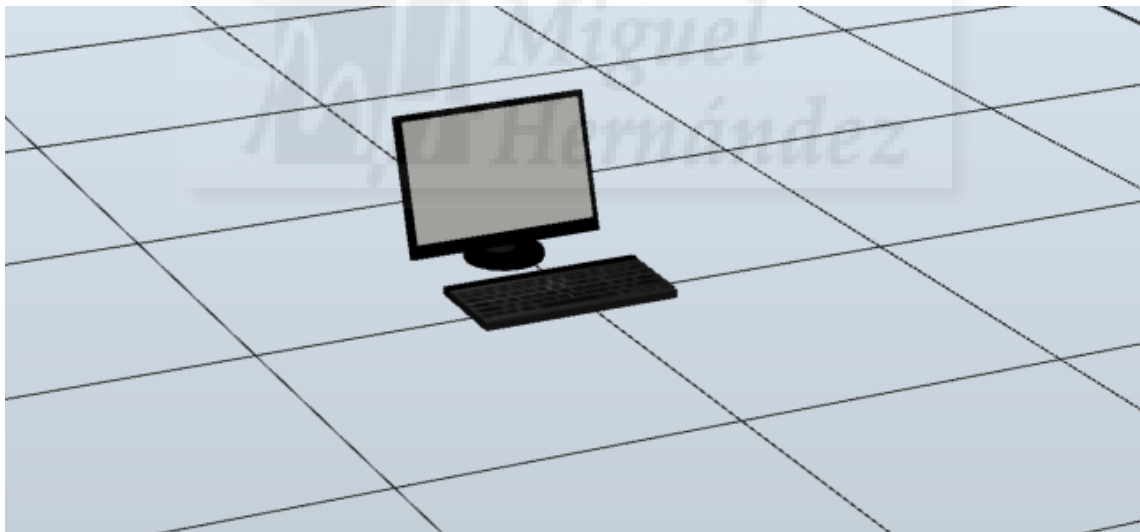


Figura 4.15: Ordenador

Se utilizará la carretilla de la figura 4.16, para transportar las bancadas de las carrocerías si fuera necesario y otros objetos.

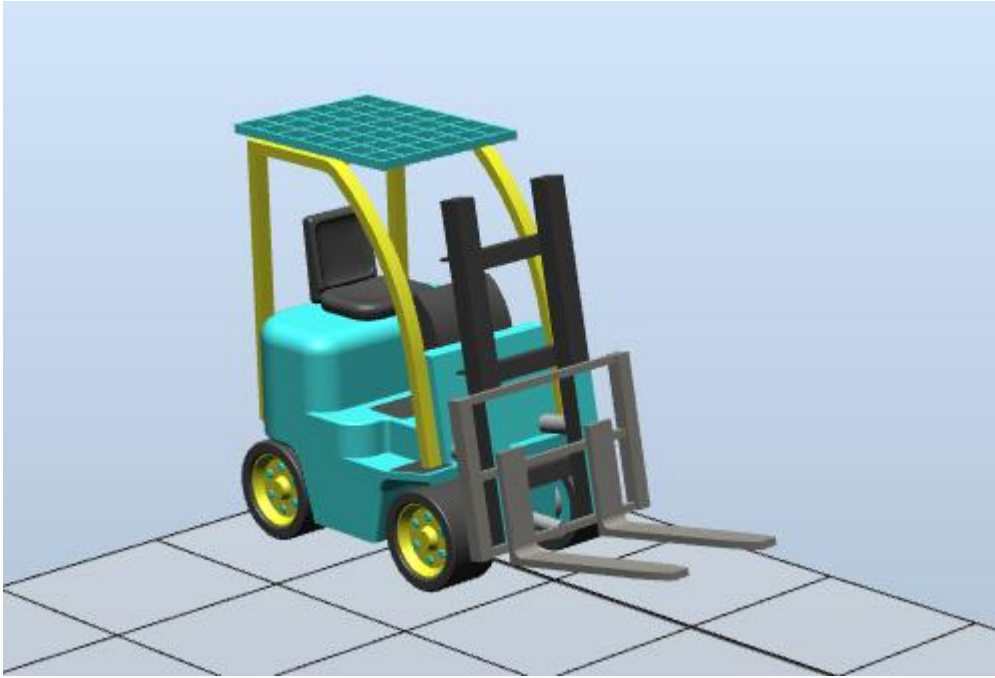


Figura 4.16: Carretilla elevadora

Se necesitarán varios operarios como el de la figura 4.17. Su labor es controlar que el robot haya realizado el pulido de las imperfecciones de la carrocería correctamente, supervisar la movilidad del puente, el controlador y el robot mediante un ordenador.



Figura 4.17: Operario

4.1.4 Creación de la Estación en RobotStudio

4.1.4.1 SC Carrocerías

El componente inteligente Carrocerías de la figura 4.18 se encuentra formado por los siguientes componentes:

- Linear Mover: mueve el objeto al que se hace referencia en Object (Queue), con una velocidad indicada por la propiedad Speed (300 mm/s) y en la dirección indicada en la propiedad Direction en mm (1000,0,0). El movimiento se inicia cuando se activa la señal de entrada Execute (MotorCinta) y se detiene cuando se desactiva la señal Execute.
- Timer: pulsa la señal Output basándose en el intervalo indicado. Deben transcurrir 4 segundos antes del primer pulso. El tiempo de simulación entre los pulsos es de 30 segundos, la señal debe ser repetida.
- Source: la propiedad Source del componente de origen indica el objeto que debe clonarse (Carroceria) cuando se recibe la señal de entrada Execute. El objeto superior de los objetos clonados se especifica mediante la propiedad Parent y una referencia al objeto clonado se especifica mediante la propiedad Copy. La señal de salida Executed significa que la clonación se ha completado.
- Queue: representa una cola FIFO (primero en entrar, primero en salir). El objeto de Back se añade a la cola cuando se activa la señal Enqueue. El objeto delantero de la cola se muestra en Front. El objeto de Front es eliminado de la cola cuando se activa la señal Dequeue. Si hay más objetos en la cola, el siguiente objeto se muestra en Front. Todos los objetos de la cola son eliminados de la cola cuando se activa la señal Clear. Si un componente transformador (como por ejemplo LinearMover) tiene un componente de cola como Object, transformará el contenido de la cola en lugar de la cola en sí.
- SimulationEvents: emite señales cuando la simulación se inicia o se detiene.
- LineSensor: define una línea por sus parámetros Start, End y Radius. Cuando una señal Active tiene el valor alto, el sensor detecta los objetos que están en intersección con la línea. Los objetos que están en intersección se muestran en la propiedad ClosestPart y el punto de la pieza en intersección más cercana al punto inicial del sensor de línea se muestra en la propiedad ClosestPoint. Cuando se produce la intersección, se activa la señal de salida SensorOut. El sensor va de Start (25000,-2500, 150) a End (25000,-500, 150).

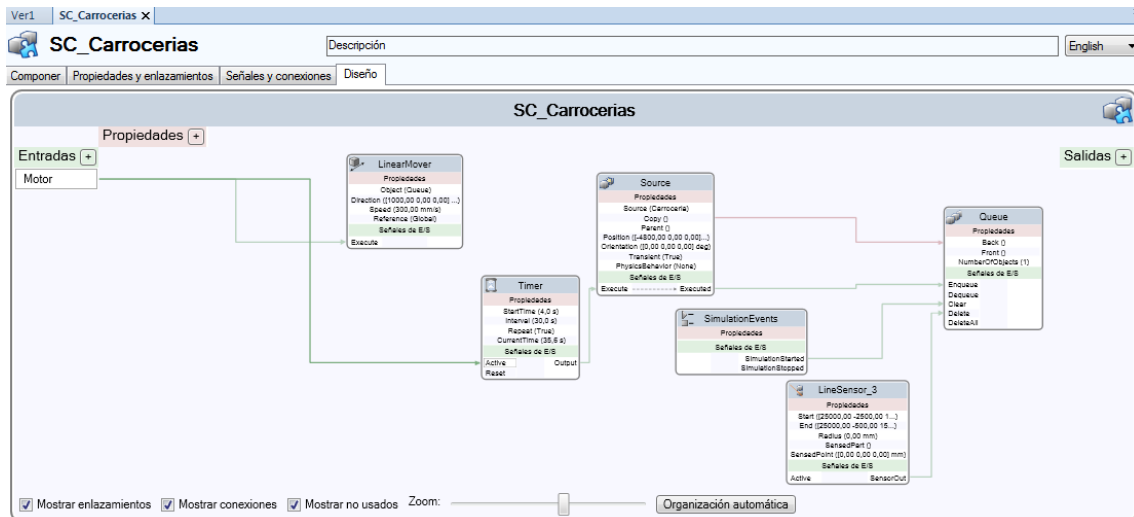


Figura 4.18: SC Carrocerías

Se añade una entrada “Motor” que es la señal que activará y pondrá en movimiento las bancadas, cuando la señal se encuentre a 0 las bancadas no se moverán y cuando esté a 1 se moverán e irán apareciendo bancadas cada 30 segundos.

La entrada “Motor” del SC Carrocerías estará conectada a la entrada del “LinearMover” y al active de “Timer”. La salida de “Timer” se conecta a la entrada de “Source”. Las copias de la Carrocería “Copy” se conectan a “Back” de “Queue”. Finalmente las señales de entrada de “Queue: Enqueue” vendrá la conexión de la señal de salida de “Source”, “Clear” se conecta con la salida “SimulationStarted” del componente “SimulationEvents” y “Delete” conectado con la salida del “LineSensor_3”.

4.1.4.2 SC Puente

Este componente realiza el movimiento del puente móvil de la estación. El puente cuenta con unas ruedas haciendo que se desplacen por los carriles construidos para estas. El componente “LineSensor” se encuentra situado el Start en (2250,-2500,150) y End (2250,-500,150), situado debajo de la posición inicial del puente. La salida del LineSensor se encuentra conectada a Execute del “LinearMover2”.

El “LinearMover” realizará el movimiento del puente hacia delante sincronizado con la misma velocidad que las carrocerías. Este componente se moverá con Direction en mm (1000,0,0), y una velocidad de 300 mm/s.

El “LineSensor_2” se encuentra el Start en (-175’806,500,150) y el End en (-175’806,-300,150), este sensor está situado en uno de los pilares de la base del puente.

El “LinearMover_2” realizará el movimiento del puente hacia atrás con mayor velocidad que las carrocerías. Este componente se moverá con Direction en mm (-1000,0,0), y una velocidad de 968 mm/s.

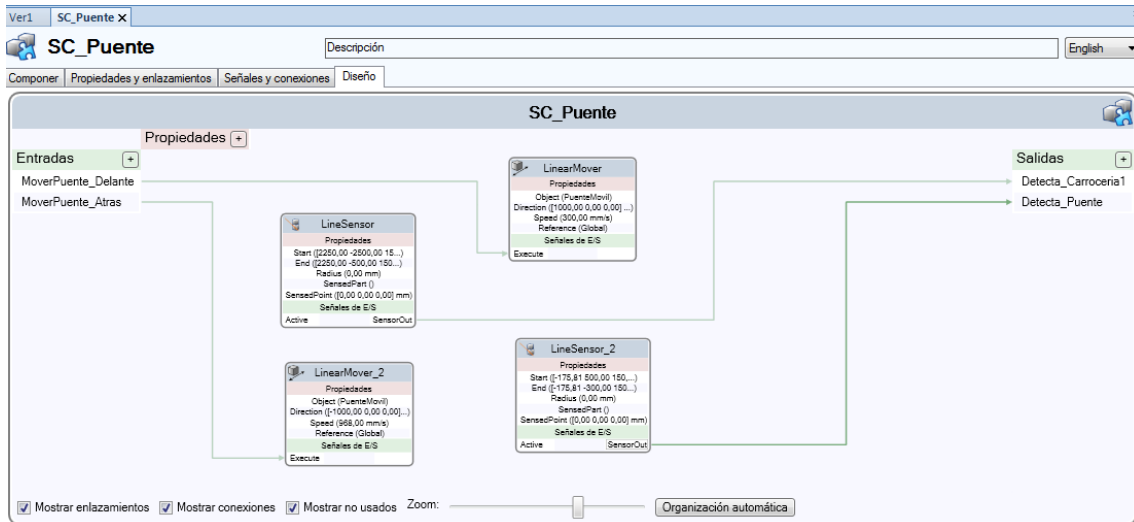


Figura 4.19: SC Puente

En el SC Carrocerías de la figura 4.19 tenemos dos entradas, “MoverPuente_Delante” conectada a Execute de LinearMover y “MoverPuente_Atras” con Execute de LinearMover_2.

Este SC tiene dos salidas, “Detecta_Carroceria1” conectada a la salida del LineSensor y “Detecta_Puente” conectada a la salida de LineSensor_2.

Cuando LineSensor detecte una bancada la señal se pondrá a 1, siendo esta señal la de Detecta_Carroceria1 y si LineSensor_2 detecta el puente entonces Detecta_Puente será 1. Cuando no sea detectado nada la señal será 0.

4.1.4.3 Controlador y Lógica de Estación

El controlador del robot IRB4600 tendrá 5 señales, 2 de entrada y 3 de salida. Las señales de entrada del controlador son “DetectaCarroceria1” y “DetectaPuente” conectadas con las salidas del SC Puente llamadas “Detecta_Carroceria1” y “Detecta_Puente”. Una señal de salida del controlador es “ActivaMotor” conectada con la entrada “Motor” del SC Carrocerías y las otras dos salidas son “MoverPuenteDelante” y “MoverPuenteAtras”, conectadas con las entradas del SC_Puente llamadas “MoverPuente_Delante” y “MoverPuente_Atras”. Estas conexiones se realizan en la lógica de estación de la figura 4.20.

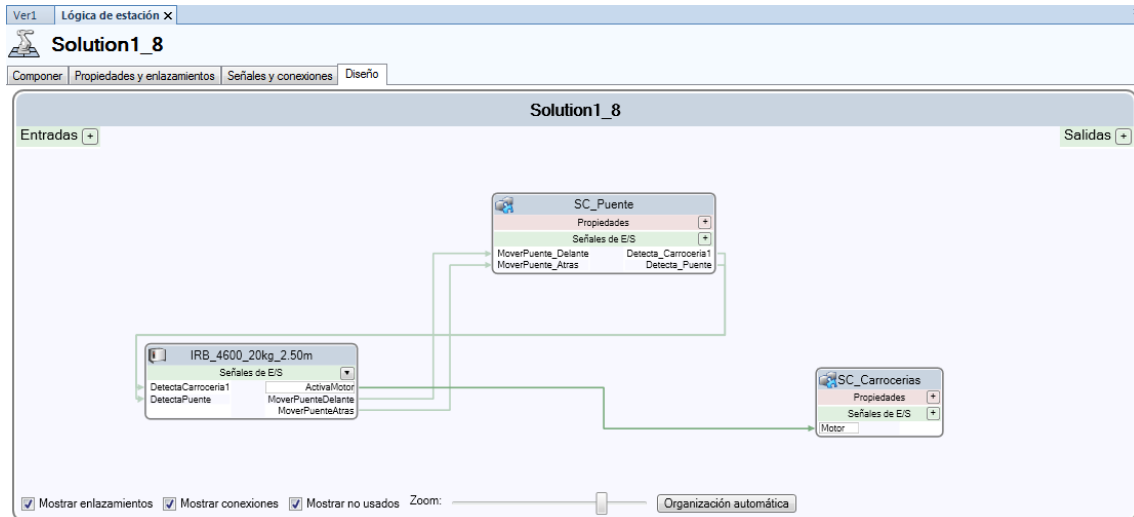


Figura 4.20: Lógica de Estación Carrocerías

4.1.4.4 Programación en RAPID

Por último lo único que quedaría para el diseño de la estación sería la programación del robot, el código desde RAPID de las señales de E/S del controlador.

Hemos creado un programa con tres subprogramas PROC, se trata de la trayectoria del robot "Path_10" y "Path_20" y finalmente el subprograma principal denominado "main".

En el subprograma Path_10 se encuentra la programación de las trayectorias de cada punto objetivo, estos puntos son definidos en nuestro programa. El subprograma Path_20 también encontramos la programación de trayectorias en otro punto.

En el subprograma main, en primer lugar se coloca el robot en su posición inicial es decir en el punto "Origen", esto se realiza por si el robot se encontrara movido y fuera de su posición inicial. Se ponen a 0 MoverPuenteDelante y MoverPuenteAtras, por si se encontrara el puente en movimiento. A continuación el motor de las bancadas se pone en marcha, siendo "ActivaMotor = 1". Una bancada junto a una carrocería aparecerá en los raíles a los 4 segundos y las siguientes bancadas cada 30 segundos, tal y como se realizó en el diseño de los componentes inteligentes.

El programa esperará a que DetectaCarroceria1 cambie a 1, "WaitDI DetectaCarroceria1,1".

Las bancadas se moverán a través de los raíles hasta que el sensor que se encuentra debajo del puente móvil de la estación detecte la bancada con la carrocería, entonces al tener "DetectaCarroceria1 = 1" se ejecutarán las siguientes líneas del programa.

Entonces el robot se pone en marcha y realiza las trayectorias "Path_10" y "Path_20", realizando el pulido sobre la carrocería.

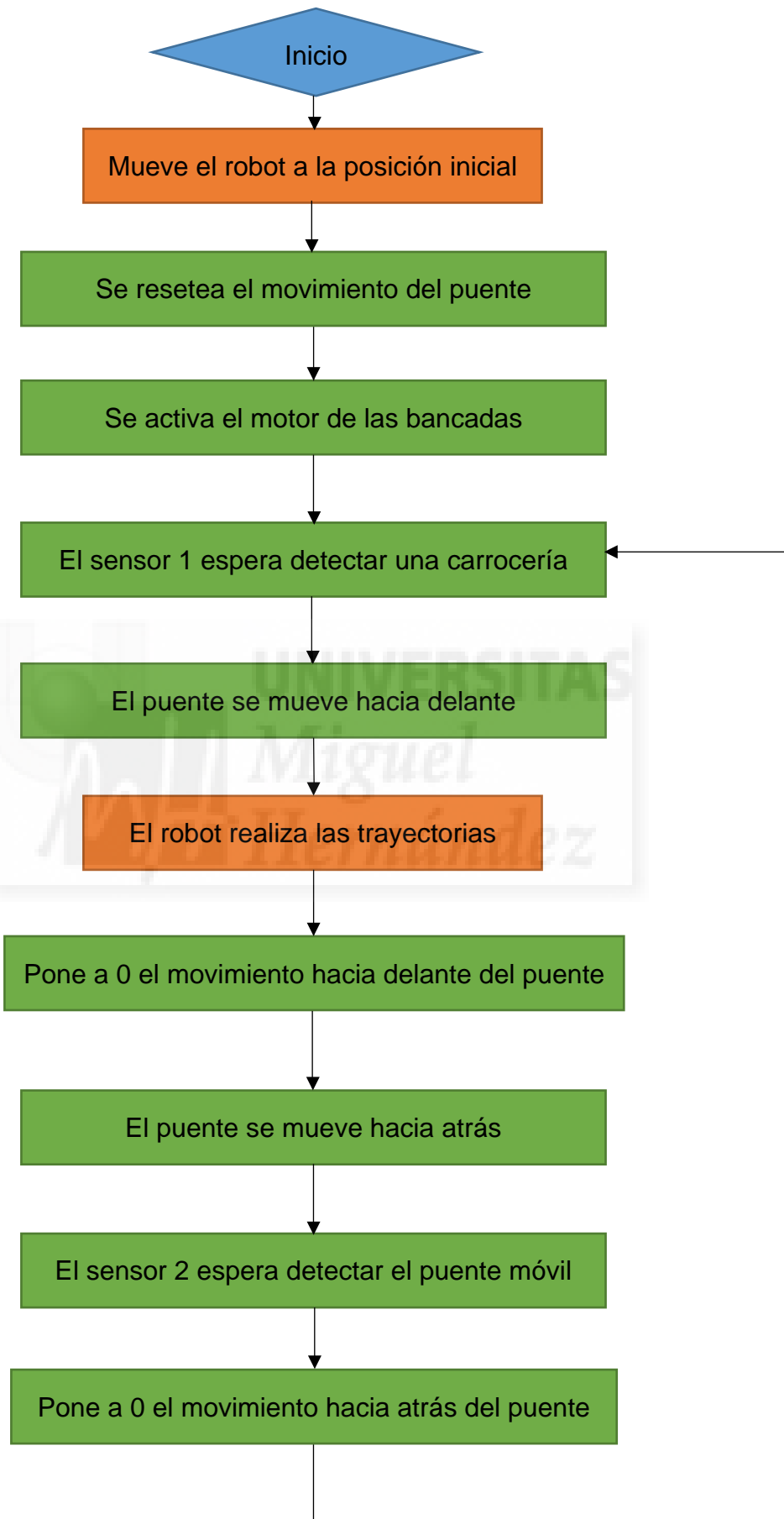
Al finalizar el robot, el programa reseteará MoverPuenteDelante, poniendo la señal a 0, parando el movimiento hacia delante. Para poner en marcha el puente hacia atrás ponemos a 1 "MoverPuenteAtras". Las bancadas junto a las carrocerías continúan en movimiento y el puente móvil regresa a su posición inicial. El puente continuará hacia atrás esperando detectar un sensor colocado en la posición inicial del puente, "WaitDI DetectaPuente,1". Si el sensor 2 detecta el puente móvil, este parará el movimiento hacia atrás, haciendo que la señal MoverPuenteAtras sea 0.

Una vez detectada la siguiente bancada junto con la carrocería, el programa vuelve a realizar la misma tarea cíclicamente.

En el anexo 8.3.2 se encuentra el código RAPID de la estación robotizada de carrocerías.

En la página siguiente se muestra un diagrama de flujo con la explicación del programa de una forma más resumida.





4.2. Estación de Salpicaderos

4.2.1 Elección del Robot

En la estación de salpicaderos se va a montar un pórtico que estará fijo al suelo, mediante placas de anclaje y tornillos. En este puente se situará una placa de sujeción en la parte media superior, que irá incrustado en el propio puente y al que se le atornillará el robot escogido.

En esta estación el robot realizará el pulido sobre la superficie de los bordes de los salpicaderos, por tanto el robot debe tener una serie de características a cumplir: rapidez en el proceso, realizar las tareas ininterrumpidamente sin errores, ser muy preciso en el pulido y poder ser montado sobre el pórtico.

Hemos escogido el robot IRB1600 porque cumple esta serie de características y es un robot utilizado en diversas de aplicaciones. No hemos escogido el mismo robot que la estación de pulido de carrocerías debido a que necesitaríamos un pórtico de mayor altura, no buscamos un robot tan grande sino un robot, fiable y con precisión antes que veloz.

Una vez elegido el modelo de robot podemos observar en la tabla 4.2, que tenemos el mismo robot pero con diferentes variantes de tamaño y de carga.

Variantes	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

Tabla 4.2: Modelos de robot IRB1600

Finalmente hemos escogido el robot IRB 1600-6/1.45 de la figura 4.21, por tener el tamaño adecuado para que el robot alcance todas las partes necesarias a pulir del salpicadero y ser el de menor carga, no siendo necesario otro, ya que el robot no va a suspender cargas elevadas, sino solamente cargará el peso de la herramienta.

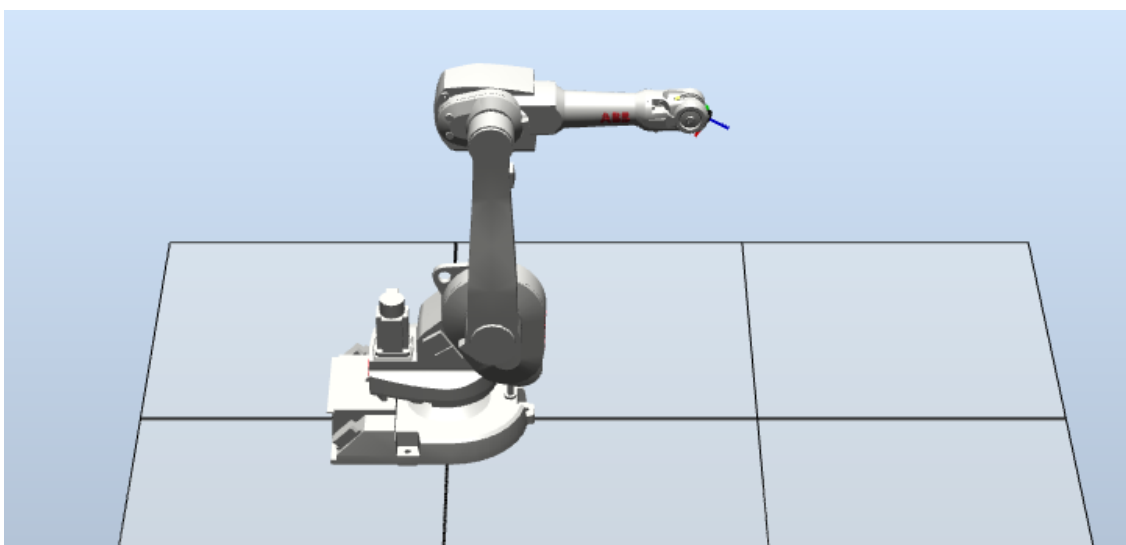


Figura 4.21: Robot ABB IRB 1600

4.2.2 Elección de la Herramienta

Herramienta pulidora con acoplamiento adaptativo a robot mediante plato, arquitectura tipo tambor con actuación radial, disco compuesto por tejido resistente a la tracción y desgaste por rozamiento, con abrasivo fino de óxido de aluminio fijado al disco por medio de un aglutinante, mostrada en la figura 4.22.



Figura 4.22: Herramienta de pulido

Esta herramienta de mecánica pasiva posee muelles que se adaptan a cualquier superficie.

4.2.3 Diseño de los Sólidos de la Estación

Para el diseño de los sólidos de la estación hemos utilizado el software de CAD 3D Autodesk Inventor Profesional 2015, para ver los planos de estos objetos se puede consultar el anexo 8.1.2.

En primer lugar se ha diseñado el pórtico de la figura 4.23, soporte del robot, de una altura y anchura suficiente que permita el paso de los salpicaderos sobre la cinta transportadora por debajo sin peligro de colisión, con los salpicaderos o cinta transportadora. Además, para dar estabilidad a la estructura del pórtico son necesarias cuatro placas de anclaje atornilladas al suelo de 30x30 cm.

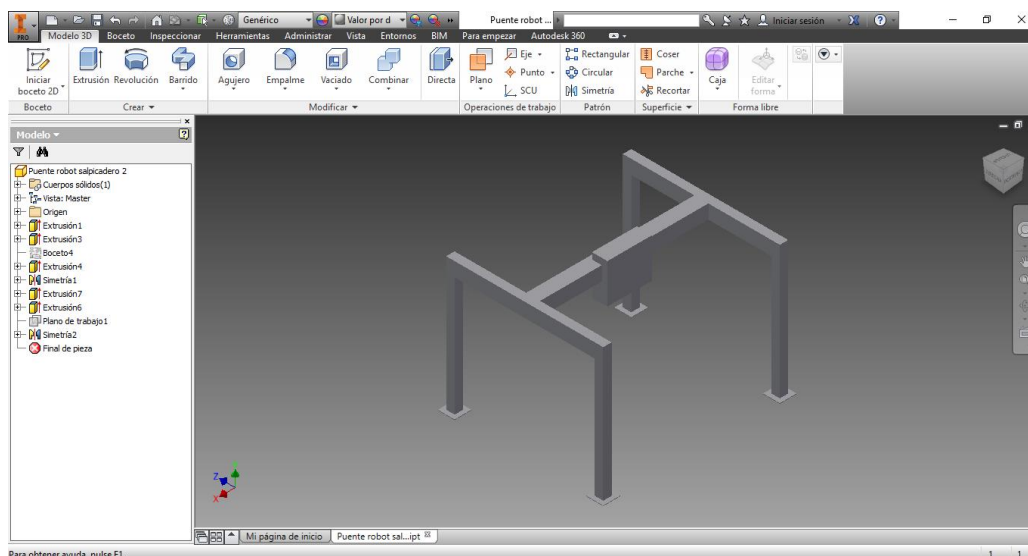


Figura 4.23: Pórtico

El robot necesita para el pulido una herramienta adecuada que no ha sido posible encontrar en las bibliotecas de RobotStudio y otras bibliotecas de CAD Open Source, por tanto se ha tenido que diseñar una como la mostrada en la figura 4.24, a partir de una pulidora dedicada a realizar tareas de pulido de superficies. Es la misma herramienta utilizada en la estación de carrocerías.

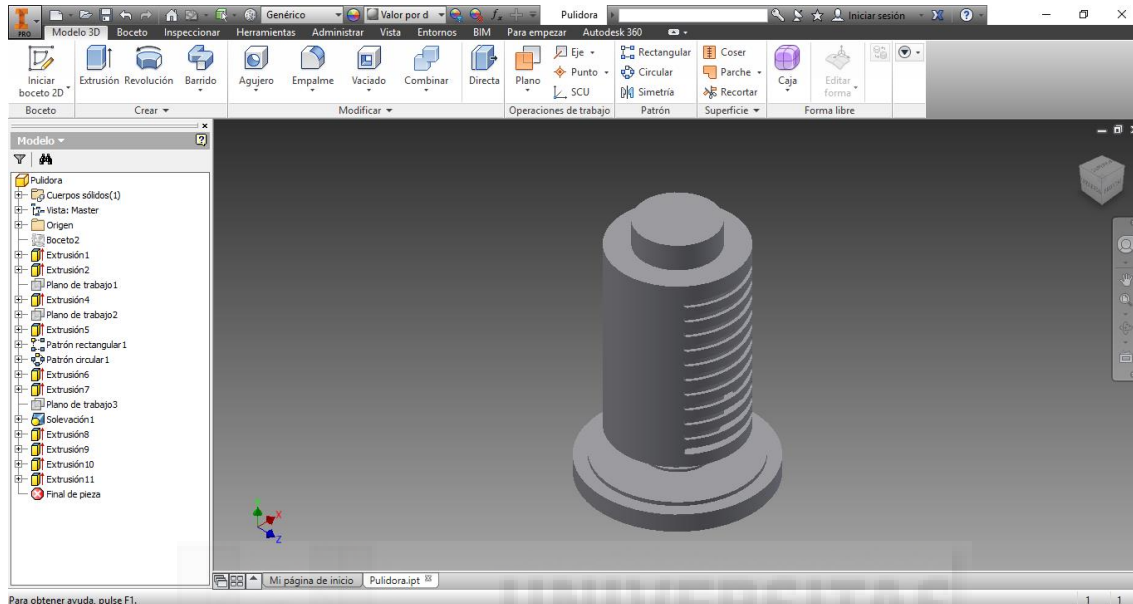


Figura 4.24: Herramienta pulidora

Es necesario diseñar una bandeja, donde se apoyen y den estabilidad por la presión que se ejerza por el brazo robótico a los salpicaderos, en su traslado a través de la cinta transportadora. Su diseño se puede comprobar en la figura 4.25.

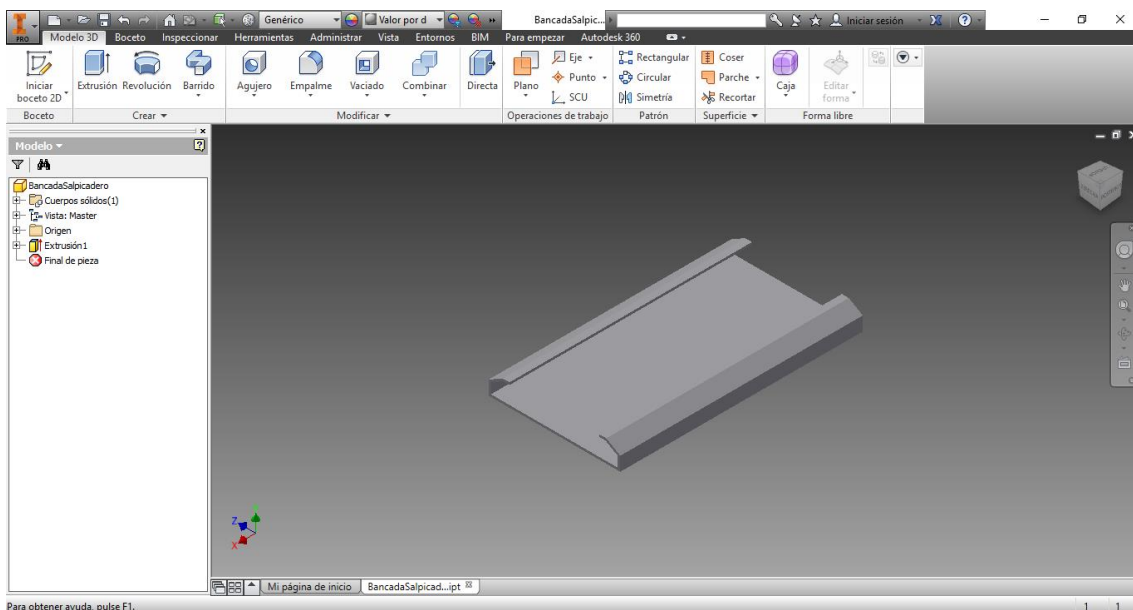


Figura 4.25: Bandeja salpicadero

Otros sólidos importados

Se han utilizado también otros sólidos que ya estaban diseñados y compartidos en librerías CAD Open Source y GrabCAD, se ha descargado en STL para incorporarlos a la estación.

En esta estación se ha importado el salpicadero de la figura 4.26, de fabricante Faurecia, donde se pulirán sus bordes.

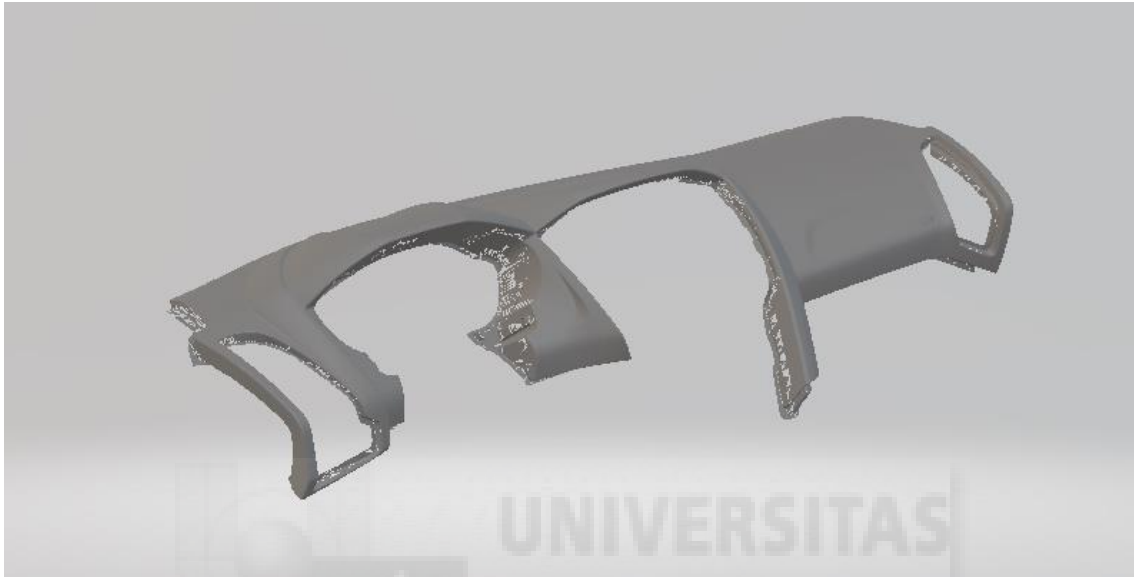


Figura 4.26: Salpicadero de fabricante Faurecia

En la estación se van a instalar 3 cintas transportadoras como la de la figura 4.27, por donde se desplazarán las bandejas junto a los salpicaderos.

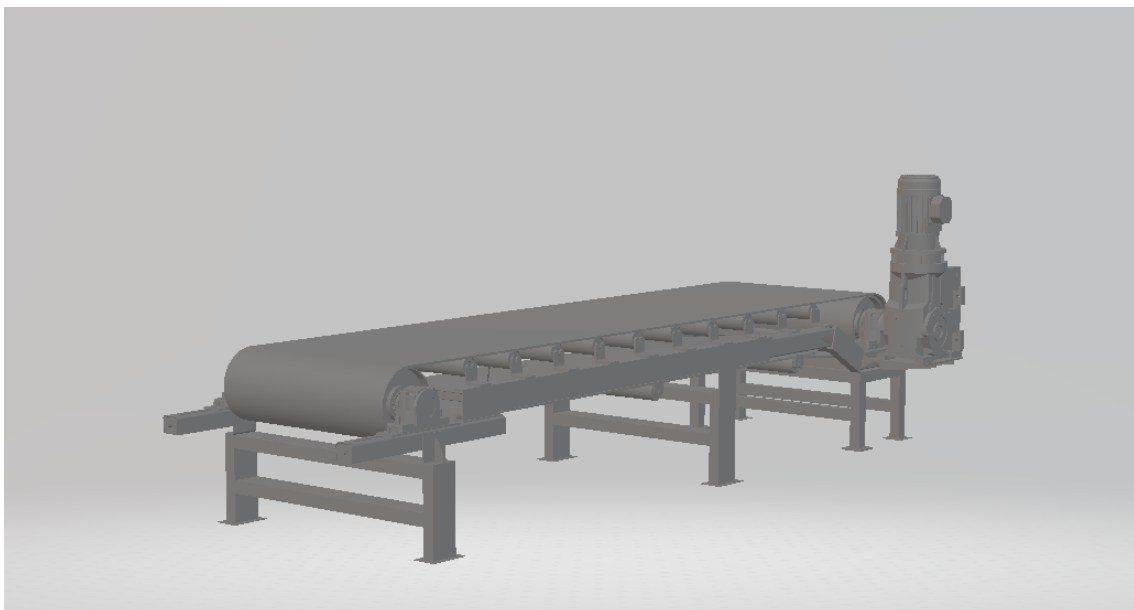


Figura 4.27: Cinta transportadora

Otros sólidos de la biblioteca RobotStudio

Para la unión de la muñeca del robot con la herramienta de pulido, vamos a utilizar un disco de plato como el de la figura 4.28.

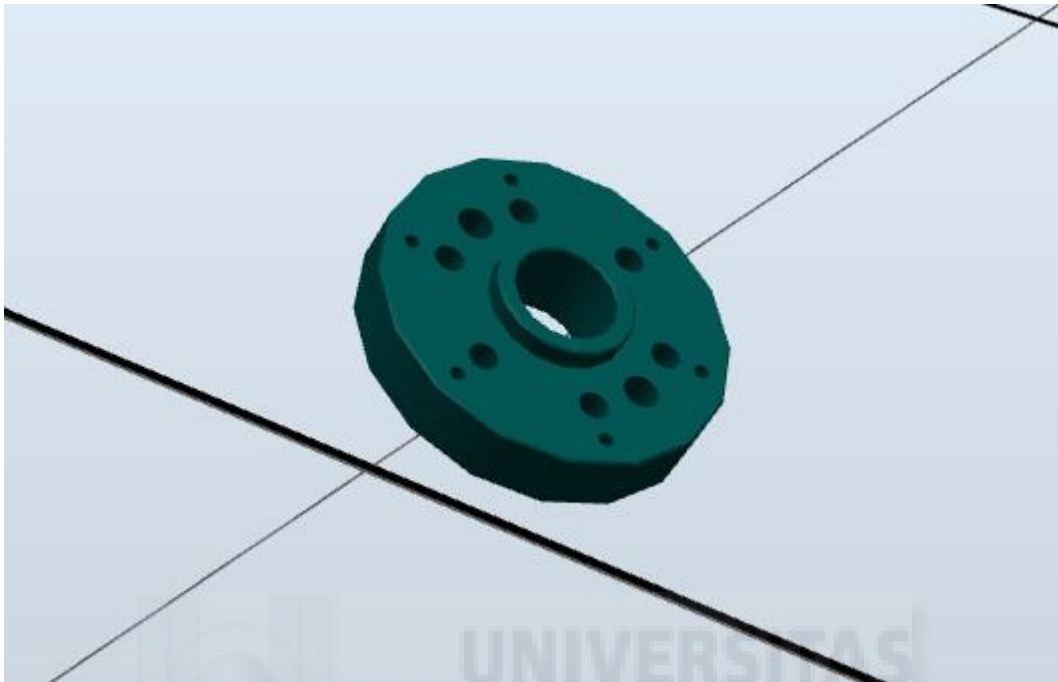


Figura 4.28: ISO Plate Large

Para la unión entre el pórtico y el robot necesitamos una pieza o Grundplate IRB1600 de la figura 4.29.

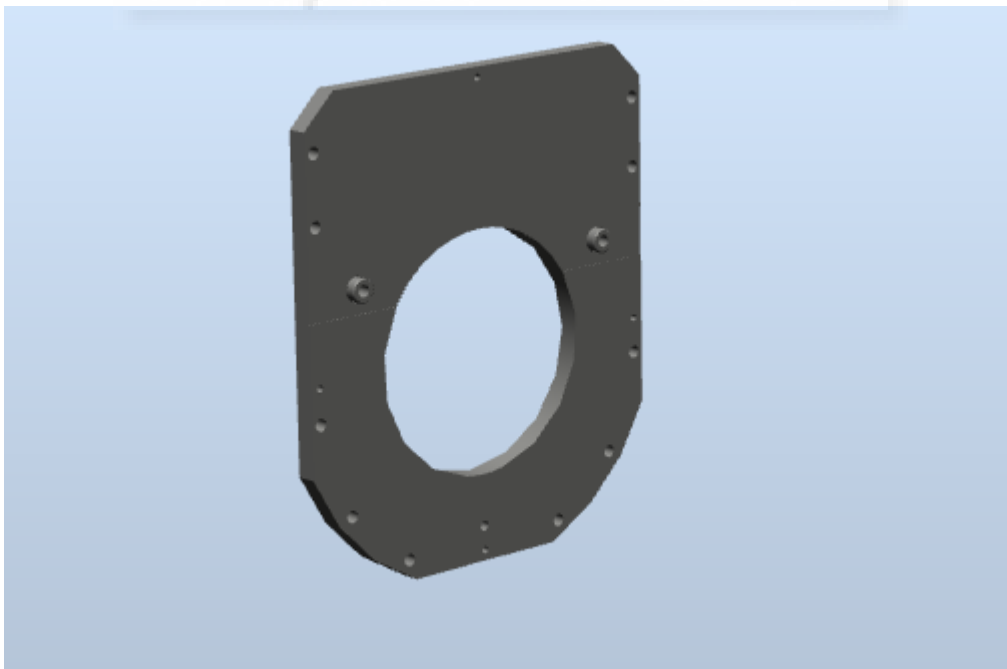


Figura 4.29: Grundplate IRB1600

Las vallas de la figura 4.30 se utilizan para delimitar un perímetro anti-intrusos.

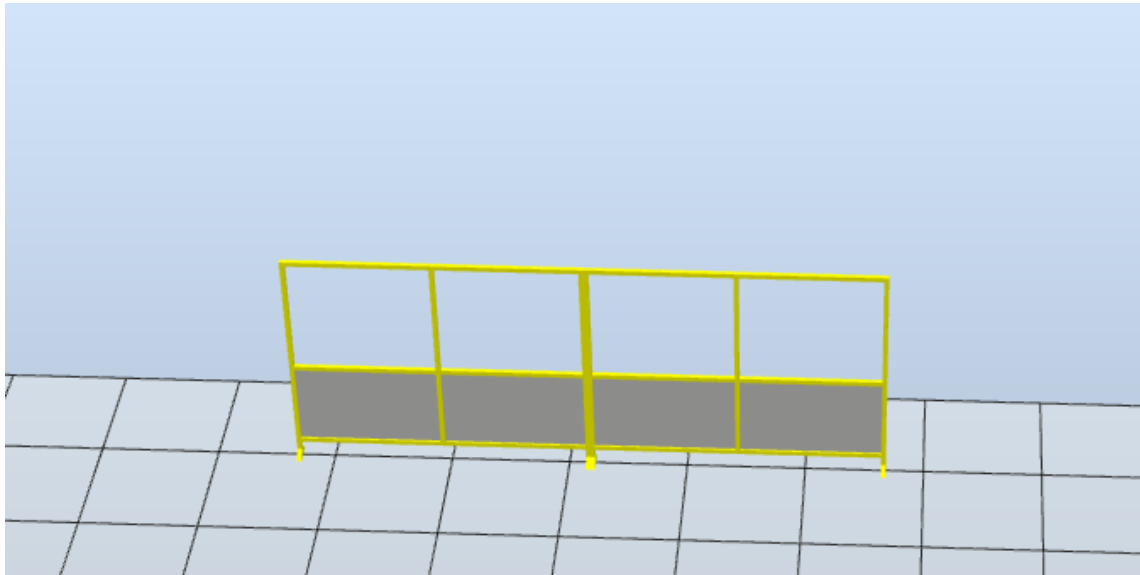


Figura 4.30: Vallas

La estación debe tener un controlador para el robot como el de la figura 4.31.

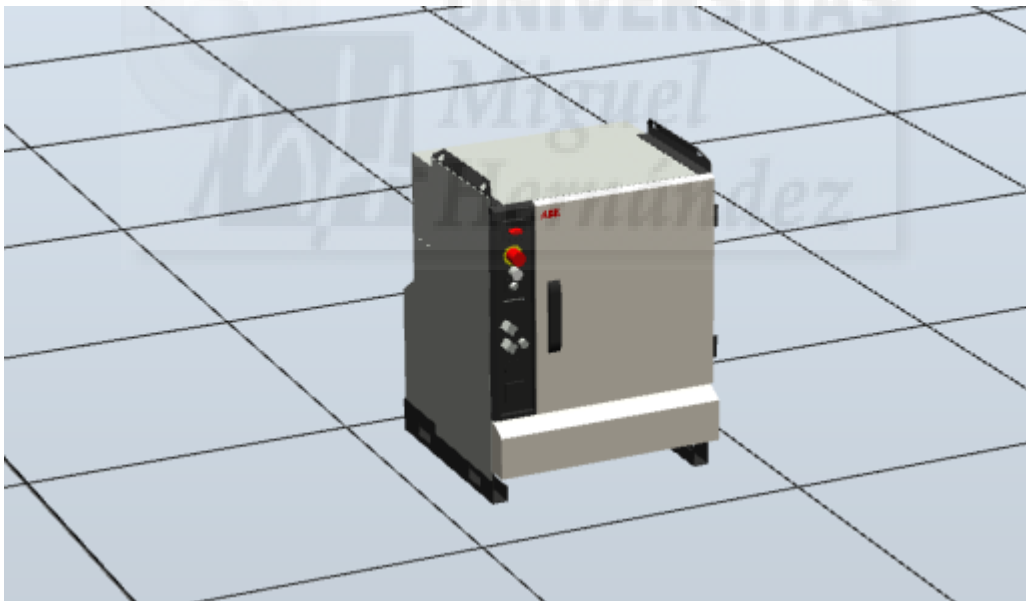


Figura 4.31: Controlador

La mesa de la figura 4.32 es utilizada para colocar un ordenador controlador de la estación.

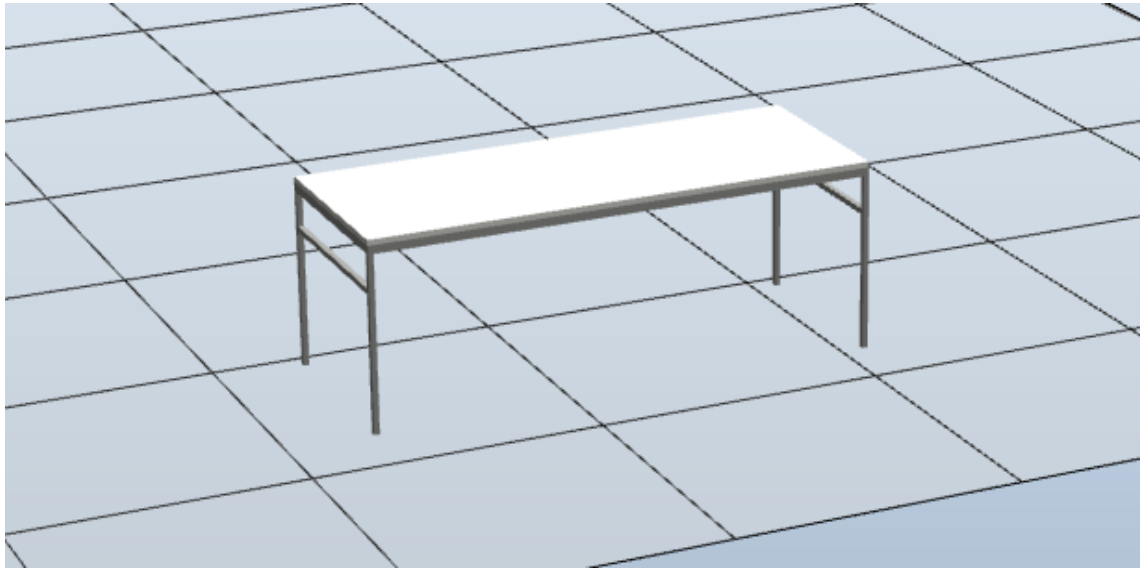


Figura 4.32: Mesa

En la figura 4.33, la silla puede ser utilizada el operario que esté a cargo del ordenador.

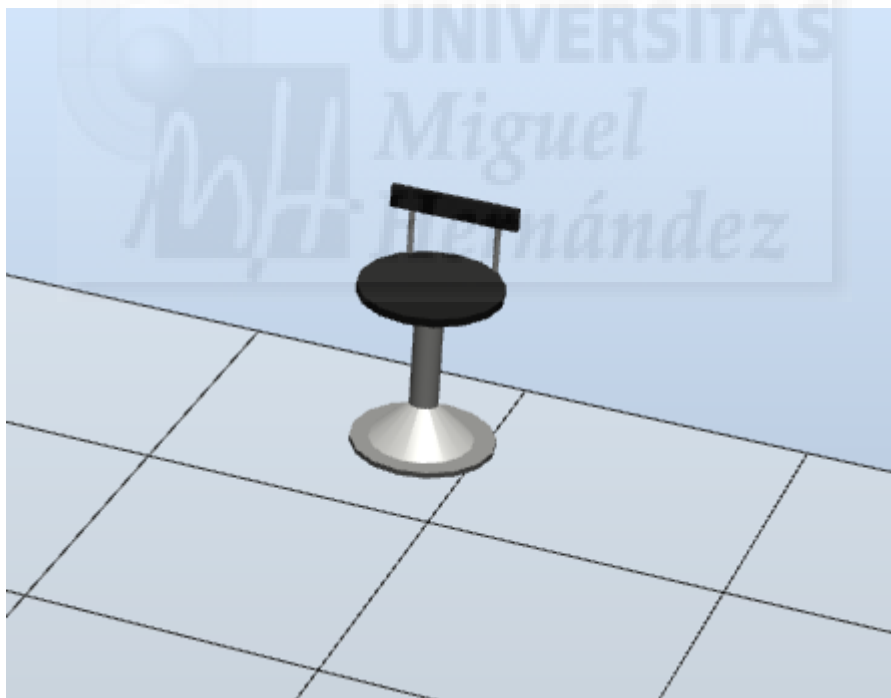


Figura 4.33: Silla

Para el control de la estación es necesario un ordenador como la figura 4.34.

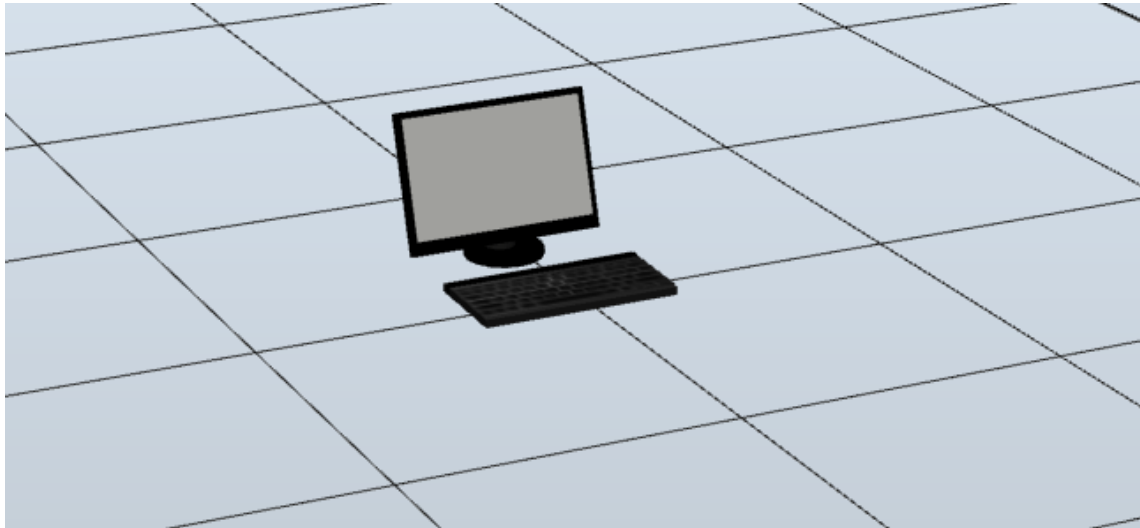


Figura 4.34: Ordenador

El palé de la figura 4.35 se utiliza para almacenar las bandejas de los salpicaderos.

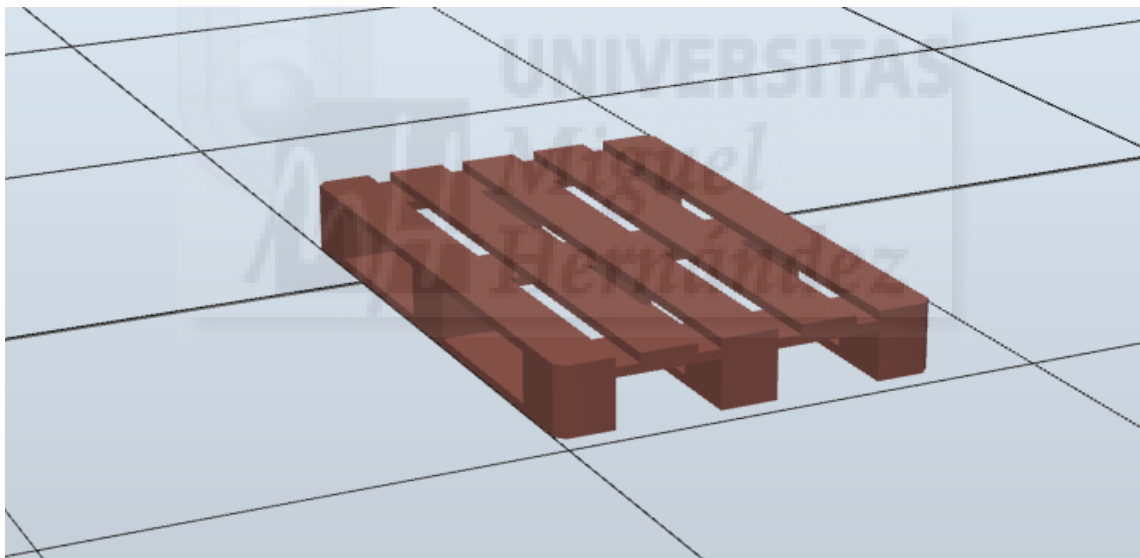


Figura 4.35: Palé

La carretilla de la figura 4.36 se utilizará para transportar las bandejas de los salpicaderos una vez que estos han sido pulidos.

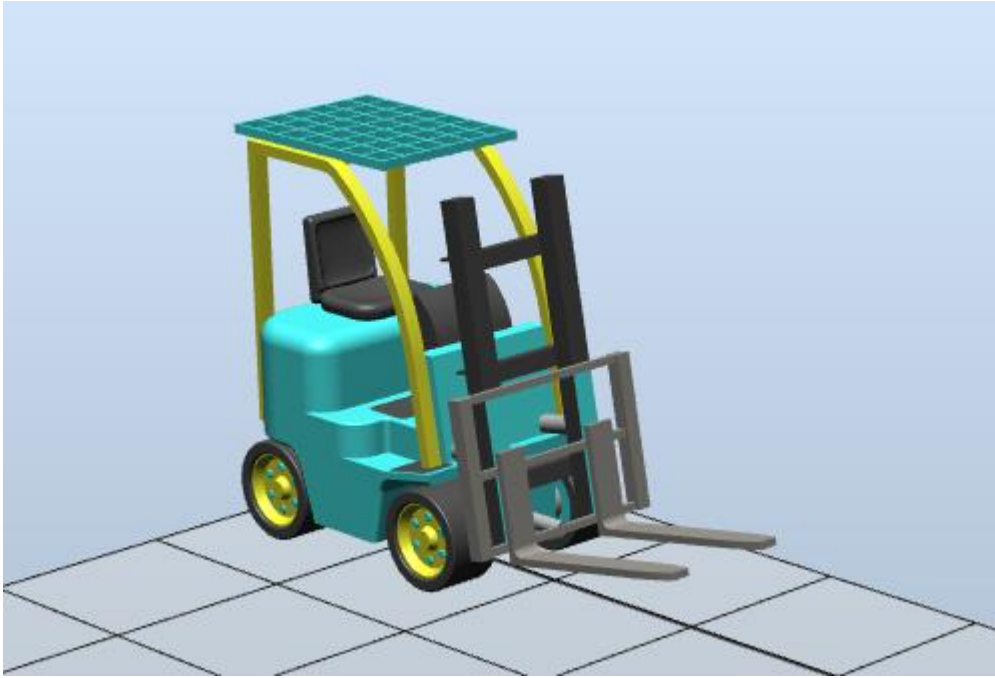


Figura 4.36: Carretilla elevadora

Se necesitarán varios operarios como el de la figura 4.37, su labor es controlar que el robot haya realizado el pulido de los bordes de los salpicaderos correctamente, supervisar el controlador y el robot mediante un ordenador. Una vez pulimentados los bordes de los salpicaderos el operario apilará las bandejas en un palé para que sean retiradas por la carretilla elevadora al inicio del proceso para ser reutilizadas.



Figura 4.37: Operario

4.2.4 Creación de la Estación en RobotStudio

4.2.4.1 SC Salpicaderos

El componente inteligente Salpicaderos de la figura 4.38 está formado por los siguientes componentes:

- Linear Mover: mueve el objeto al que se hace referencia en Object (Queue), con una velocidad indicada por la propiedad Speed (300 mm/s) y en la dirección indicada en la propiedad Direction (-1000,0,0). El movimiento se inicia cuando se activa la señal de entrada Execute (MotorCinta) y se detiene cuando se desactiva la señal Execute.
- Timer: pulsa la señal Output basándose en el intervalo indicado. Deben transcurrir 4 segundos antes del primer pulso. El tiempo de simulación entre los pulsos es de 6 segundos, la señal debe ser repetida.
- Source: la propiedad Source del componente de origen indica el objeto que debe clonarse (Salpicaderos) cuando se recibe la señal de entrada Execute. El objeto superior de los objetos clonados se especifica mediante la propiedad Parent y una referencia al objeto clonado se especifica mediante la propiedad Copy. La señal de salida Executed significa que la clonación se ha completado.
- Queue: representa una cola FIFO (primero en entrar, primero en salir). El objeto de Back se añade a la cola cuando se activa la señal Enqueue. El objeto delantero de la cola se muestra en Front. El objeto de Front es eliminado de la cola cuando se activa la señal Dequeue. Si hay más objetos en la cola, el siguiente objeto se muestra en Front. Todos los objetos de la cola son eliminados de la cola cuando se activa la señal Clear. Si un componente transformador (como por ejemplo LinearMover) tiene un componente de cola como Object, transformará el contenido de la cola en lugar de la cola en sí.
- SimulationEvents: emite señales cuando la simulación se inicia o se detiene.
- LineSensor: define una línea por sus parámetros Start, End y Radius. Cuando una señal Active tiene el valor alto, el sensor detecta los objetos que están en intersección con la línea. Los objetos que están en intersección se muestran en la propiedad ClosestPart y el punto de la pieza en intersección más cercana al punto inicial del sensor de línea se muestra en la propiedad ClosestPoint. Cuando se produce la intersección, se activa la señal de salida SensorOut. El sensor va de Start (-7750,-1000, 810) a End (-7750,-2350, 810).

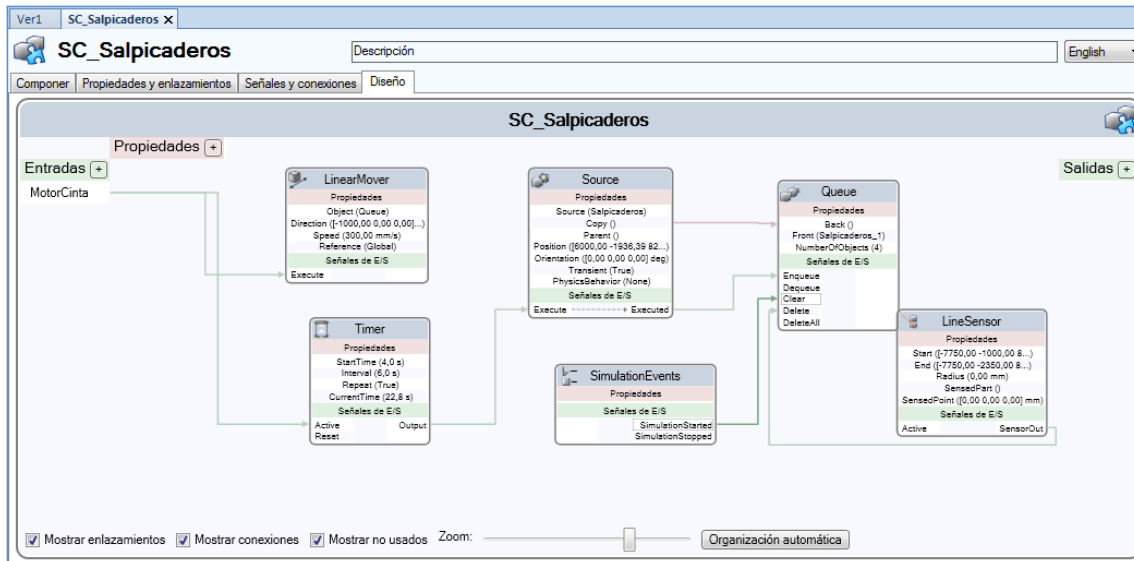


Figura 4.38: SC Salpicaderos

Se añade una entrada “MotorCinta” que es la señal que activará y pondrá en movimiento la cinta, cuando la señal se encuentre a 0 la cinta no se moverá y cuando esté a 1 la cinta se moverá e irán apareciendo bandejas cada 6 segundos.

La entrada “MotorCinta” del SC Salpicaderos estará conectada a la entrada del “LinearMover” y al active de “Timer”. La salida de “Timer” se conecta a la entrada de “Source”. Las copias de los Salpicaderos “Copy” se conectan a “Back” de “Queue”. Finalmente las señales de entrada de “Queue: Enqueue” vendrá la conexión de la señal de salida de “Source”, “Clear” se conecta con la salida “SimulationStarted” del componente “SimulationEvents” y “Delete” conectado con la salida del “LineSensor”.

4.2.4.2 SC Cintas

Este componente realiza la detección de una bandeja sobre un sensor lineal en la cinta transportadora. El componente LineSensor se encuentra situado el Start en (1000,-1000,810) y End (1000,-2350,810), justo debajo del robot.

Tenemos en el SC Cintas de la figura 4.39 una entrada “ActivaSensores” conectada a la entrada de LineSensor y la salida de este a una entrada creada llamada “Detecta_Salpicadero”.

Cuando ActivaSensores esté a 0 el sensor no detectará cuando pase por la línea algún objeto, por tanto debe encontrarse a 1 para que cuando pase un objeto a cortando la línea del sensor este la detecte. Cuando la salida del sensor lineal no detecte algún objeto será 0 y 1 cuando un objeto sea detectado, la salida del SC Cintas será 0 o 1 según sea la salida del sensor lineal.

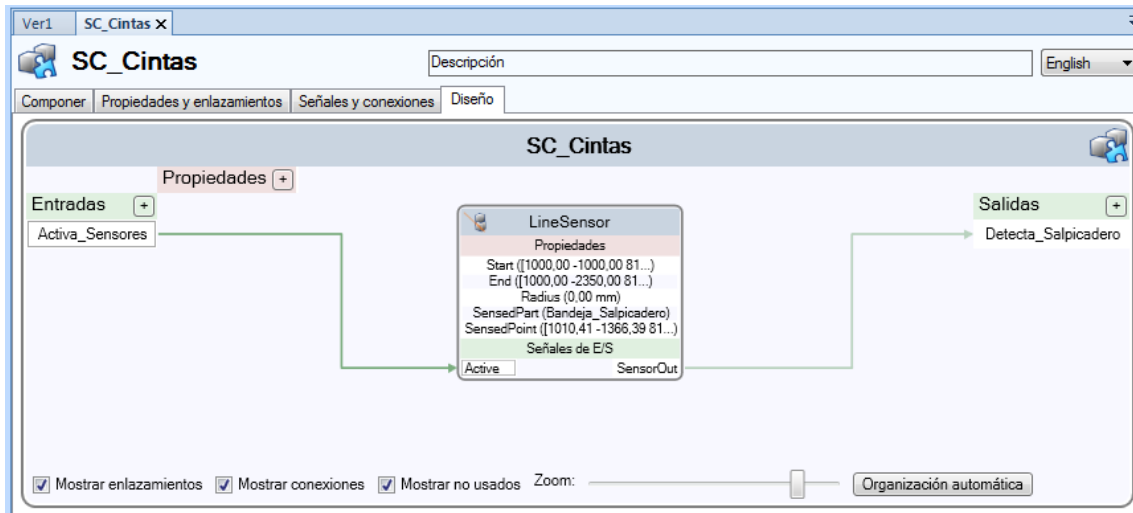


Figura 4.39: SC Cintas

4.2.4.3 Controlador y Lógica de Estación

El controlador del robot IRB1600 tendrá 3 señales, 1 de entrada y 2 de salida. La señal de entrada del controlador es “DetectaSalpicadero” conectada con la salida del SC Cintas llamada “Detecta_Salpicadero”. La señal de salida del controlador “ActivaSensores” se conecta con “ActivaSensores” de SC Cintas y la otra salida del controlador “ActivaMotorCinta” es conectada con la entrada “MotorCinta” del SC Salpicaderos. Estas conexiones se realizan en la lógica de estación de la figura 4.40.

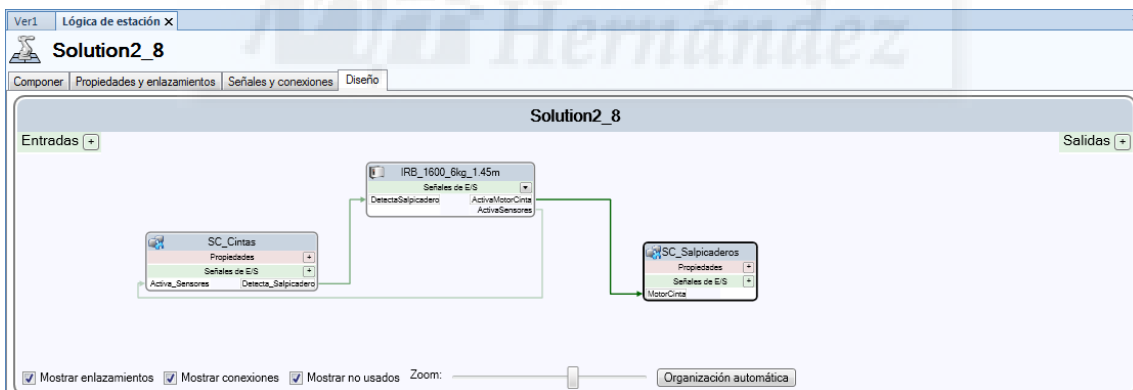


Figura 4.40: Lógica de Estación Salpicaderos

4.2.4.4 Programación en RAPID

Por último lo único que quedaría para el diseño de la estación sería la programación del robot, el código desde RAPID de las señales de E/S del controlador.

Hemos creado un programa con dos subprogramas PROC, uno se trata de la trayectoria del robot “Path_10” y otro subprograma principal denominado “main”.

En el subprograma Path_10 se encuentra la programación de las trayectorias de cada punto objetivo, estos puntos son definidos en nuestro programa.

En el subprograma main, en primer lugar se coloca el robot en su posición inicial es decir en el punto "Origen", esto se realiza por si el robot se encontrara movido y fuera de su posición inicial. Una vez colocado, la cinta transportadora se pone en marcha, siendo "ActivaMotorCinta = 1". Una bandeja junto a los salpicaderos aparecerá en la cinta a los 4 segundos y las siguientes bandejas cada 6 segundos, tal y como se realizó en el diseño de los componentes inteligentes.

Las bandejas serán transportadas por la cinta hasta que el sensor que se encuentra debajo del robot de la estación detecte el salpicadero, entonces al tener "DetectaSalpicadero = 1" se ejecutarán las siguientes líneas del programa.

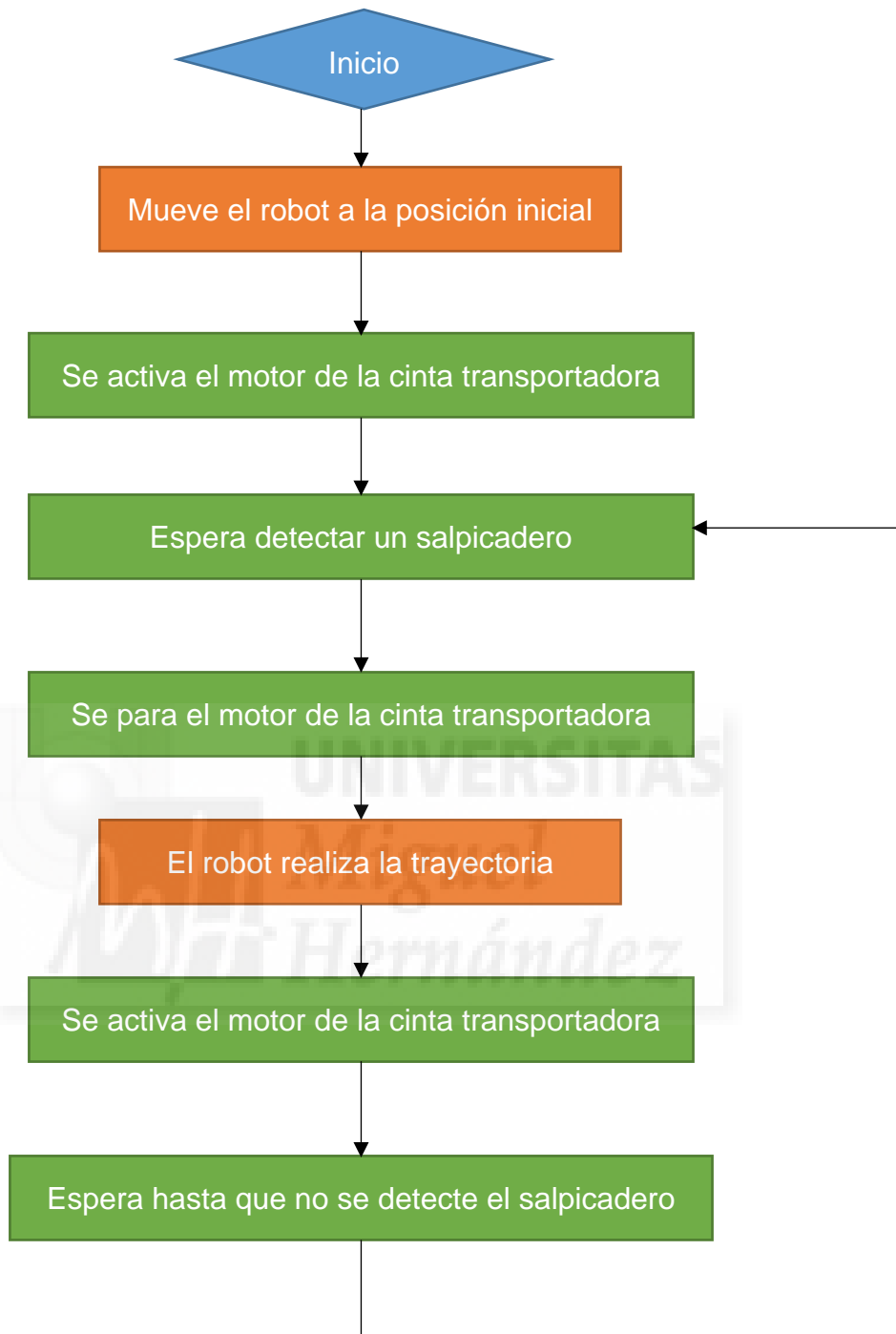
Una vez el salpicadero ha sido detectado por el sensor, entonces las cintas transportadoras se pararan, siendo "ActivaMotorCinta =0". Cuando se ha parado el robot se pone en marcha y realiza la trayectoria "Path_10", realizando el pulido sobre el salpicadero.

Al finalizar el robot, la cinta transportadora vuelve a ponerse en marcha, siendo "ActivaMotorCinta = 1". Los salpicaderos continúan en movimiento, pero el programa espera hasta que el salpicadero pulido deje de ser detectado por el sensor "WaitDI DetectaSalpicadero,0", porque si no las cintas transportadoras se volverían a parar.

Una vez dejado de detectar la bandeja junto con el salpicadero, el programa vuelve a realizar la misma tarea cíclicamente, esperando que la siguiente bandeja sea detectada por el sensor.

En el anexo 8.3.3 se encuentra el código RAPID de la estación robotizada de salpicaderos.

En la página siguiente se muestra un diagrama con la explicación del programa de una forma más resumida.





5. RESULTADOS

5.1 Simulación de la Estación de Carrocerías

En este apartado se muestra la simulación de la estación de pulido de salpicaderos, en la figura 5.1 se puede observar la estación antes de iniciar la simulación.

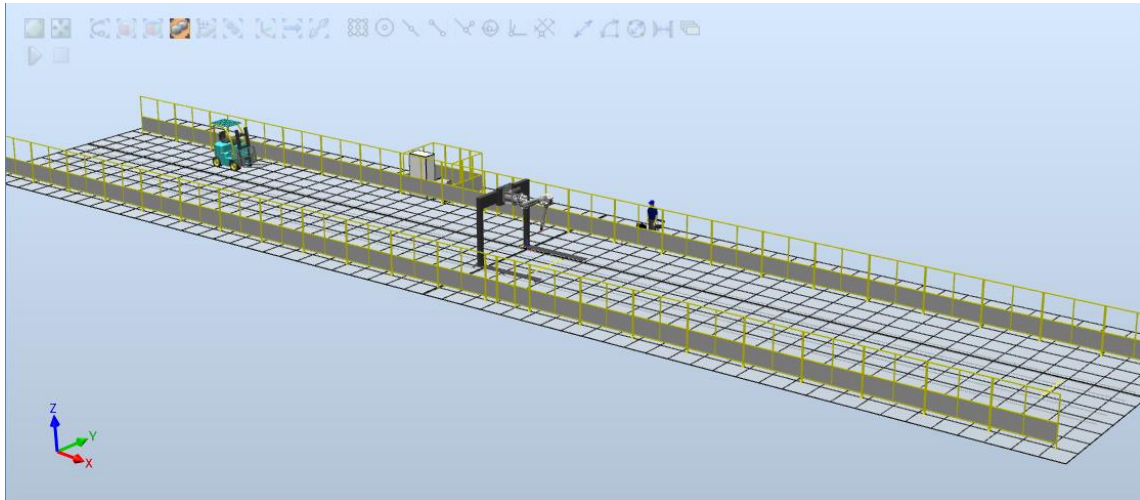


Figura 5.1: Estación Carrocerías

En primer lugar el robot se colocará en su posición inicial si no está en ella, se pondrá en marcha el sistema y a los 4 segundos aparecerá la primera bancada con la carrocería como se muestra en la figura 5.2.

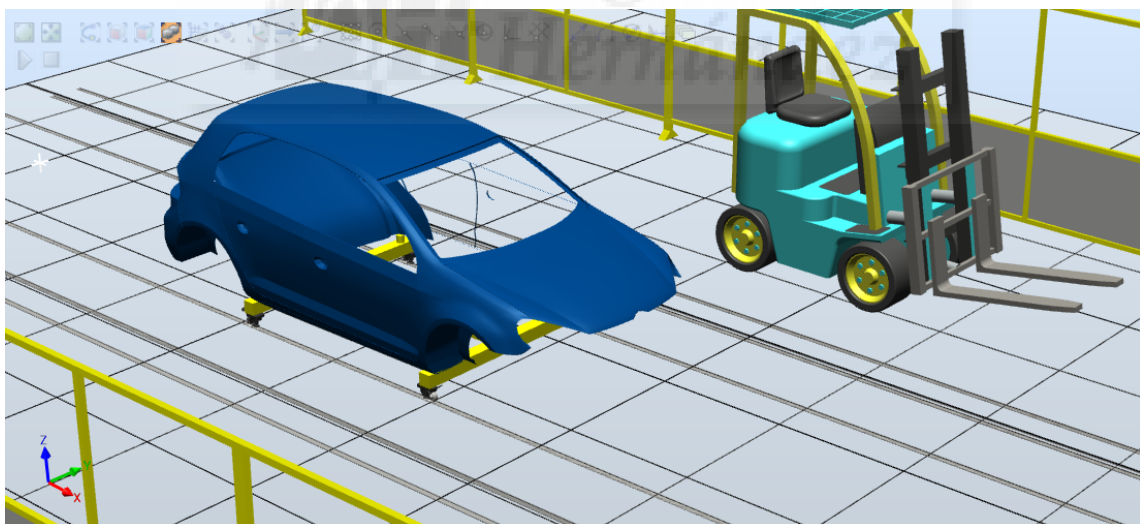


Figura 5.2: Primera bancada con carrocería

Una vez que aparece la bancada, esta se mueve como en la figura 5.3 a través de los carriles.

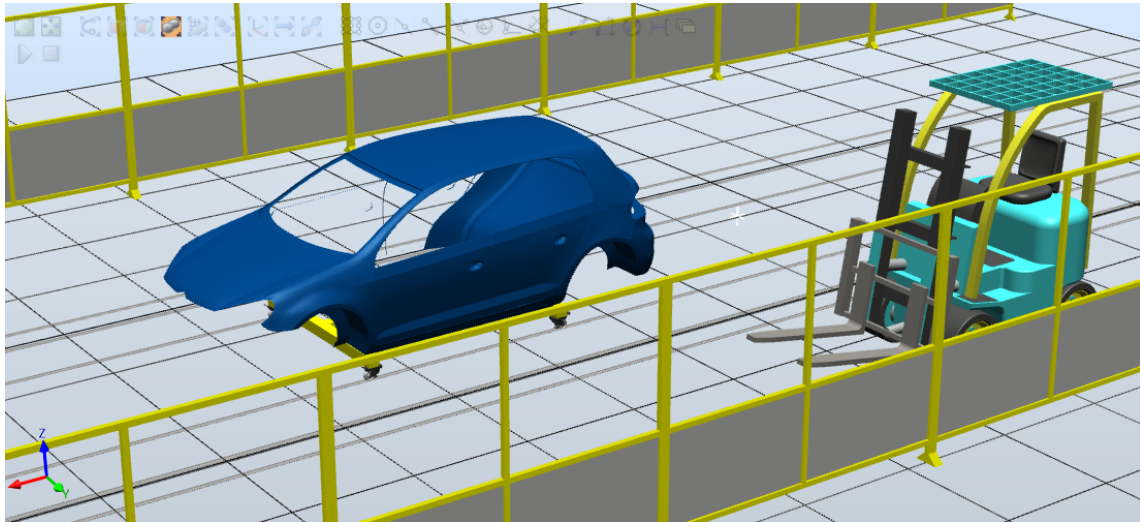


Figura 5.3: Bancada con carrocería en movimiento

La segunda carrocería aparecerá a los 30 segundos, figura 5.4.

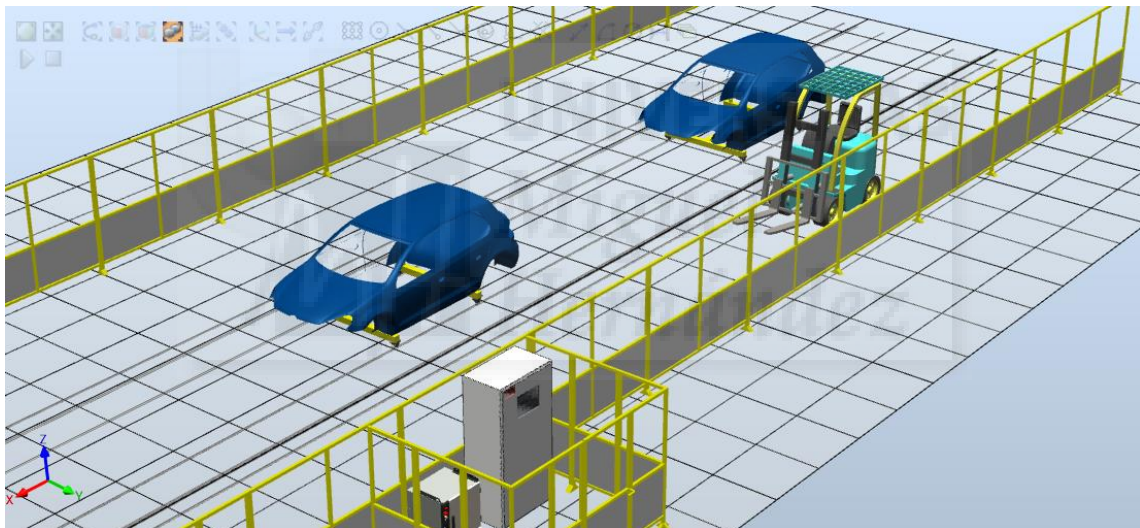


Figura 5.4: Segunda carrocería

El sensor en la figura 5.5, detecta el objeto cuando este corta el rayo.

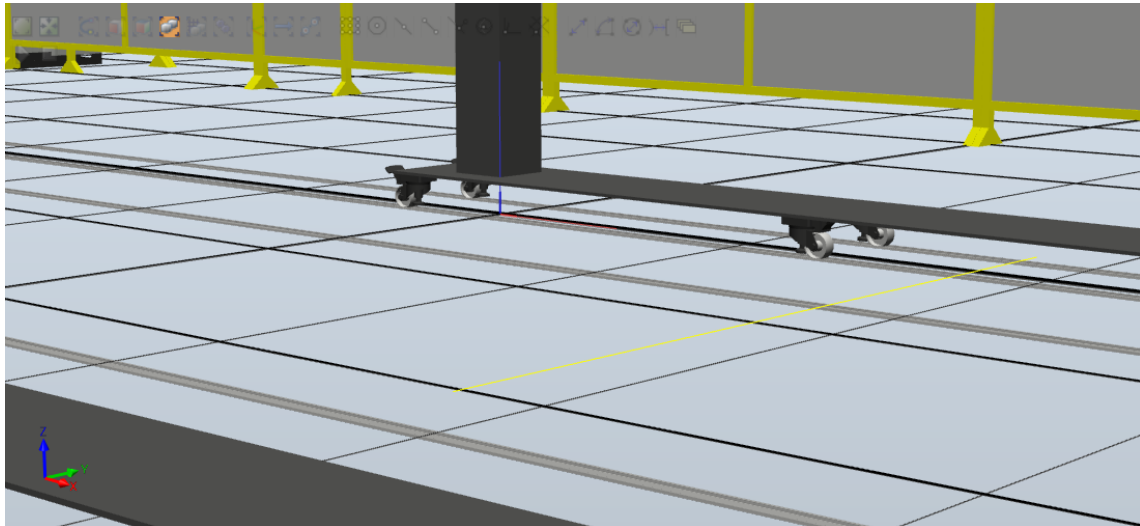


Figura 5.5: Sensor detector objetos

Cuando el sensor detecta la bancada en la figura 5.6, el puente móvil se pone en marcha a la misma velocidad con la que se mueven las carrocerías de los automóviles, realizando una sincronización.

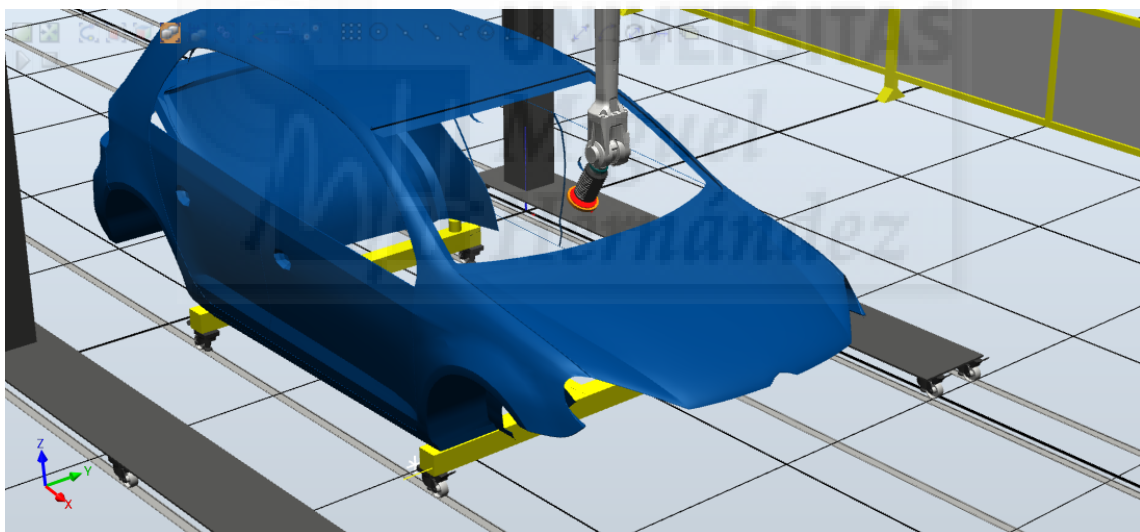


Figura 5.6: El sensor detecta la bancada

En base a la información recopilada por el escáner en 3D al pasar la carrocería por él, y comparando con los datos virtuales del CAD almacenado en el PC, el sistema detecta las imperfecciones de las carrocerías. El robot en la figura 5.7 se pone en marcha y comienza a realizar el pulido de superficies en las imperfecciones detectadas, pudiendo hacerlo sobre una o varias zonas.

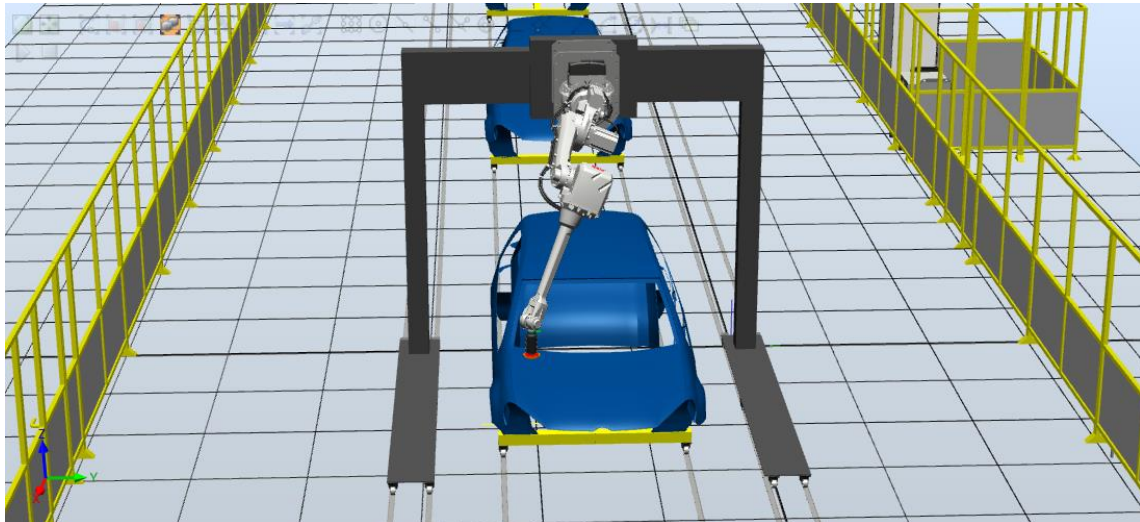


Figura 5.7: Robot en movimiento

Podemos observar en la figura 5.8, el robot realiza un pulido en la zona lateral y parte del capó de la carrocería del automóvil.

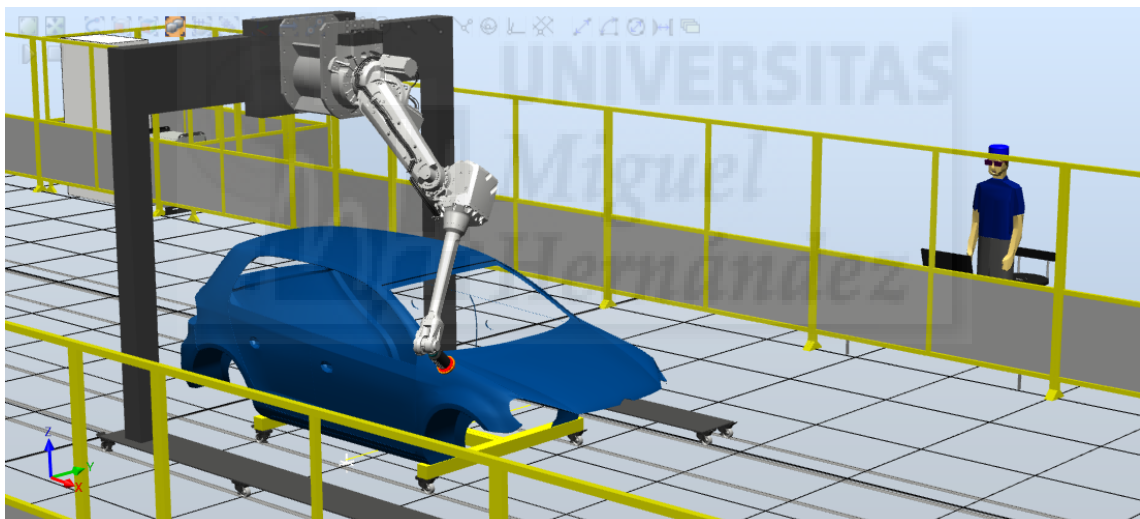


Figura 5.8: Pulido en lateral y capó de la carrocería

Otra zona donde el robot realiza el pulido, tiene lugar en el techo de la carrocería en la figura 5.9.

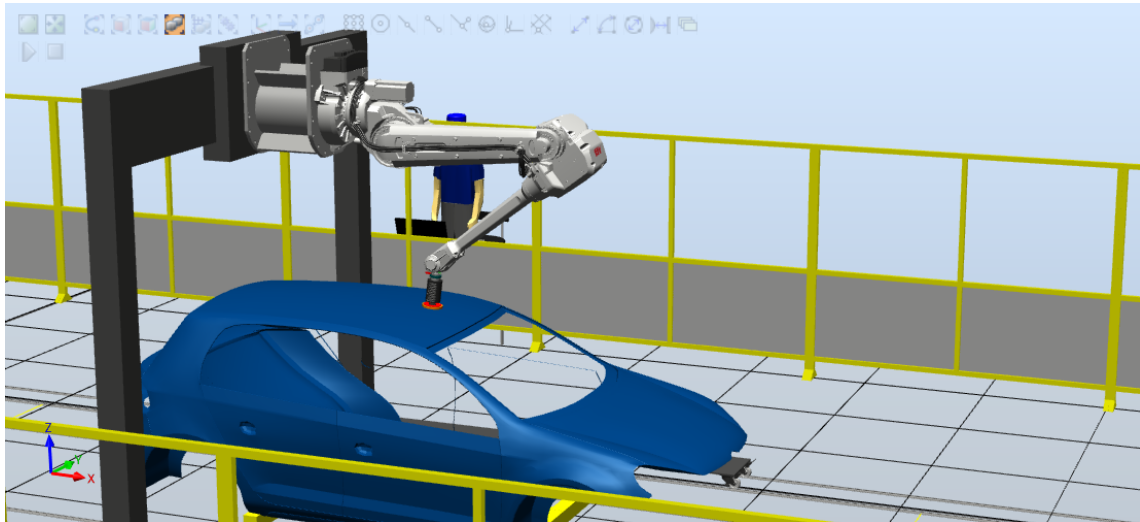


Figura 5.9: Pulido en el techo de la carrocería

En la figura 5.10 se observa con detalle la herramienta del robot en su proceso de pulido.

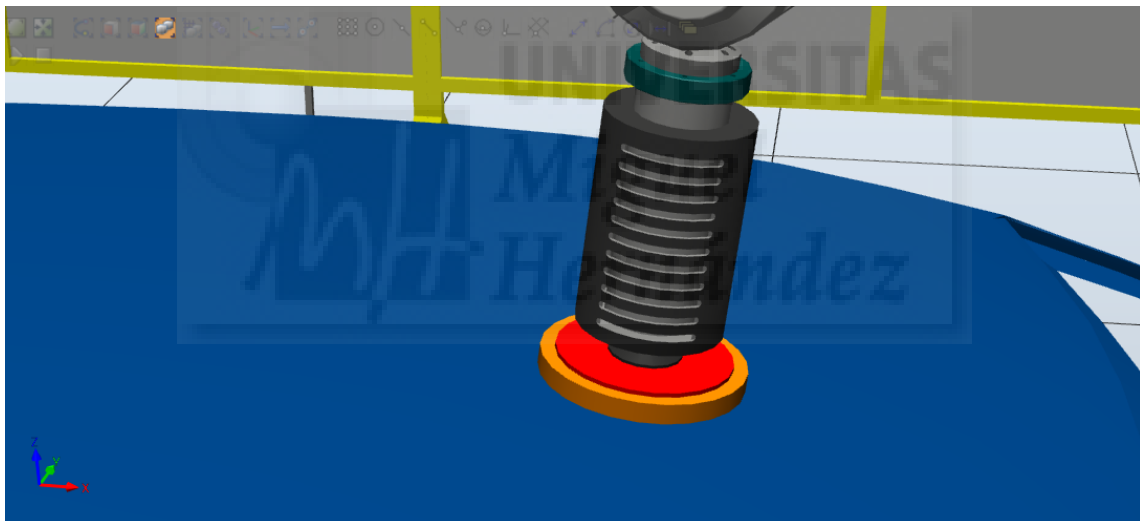


Figura 5.10: Detalle de la herramienta del robot puliendo

Una vez acabado el pulido, el robot en la figura 5.11 se coloca en una posición donde no colisione con la carrocería.

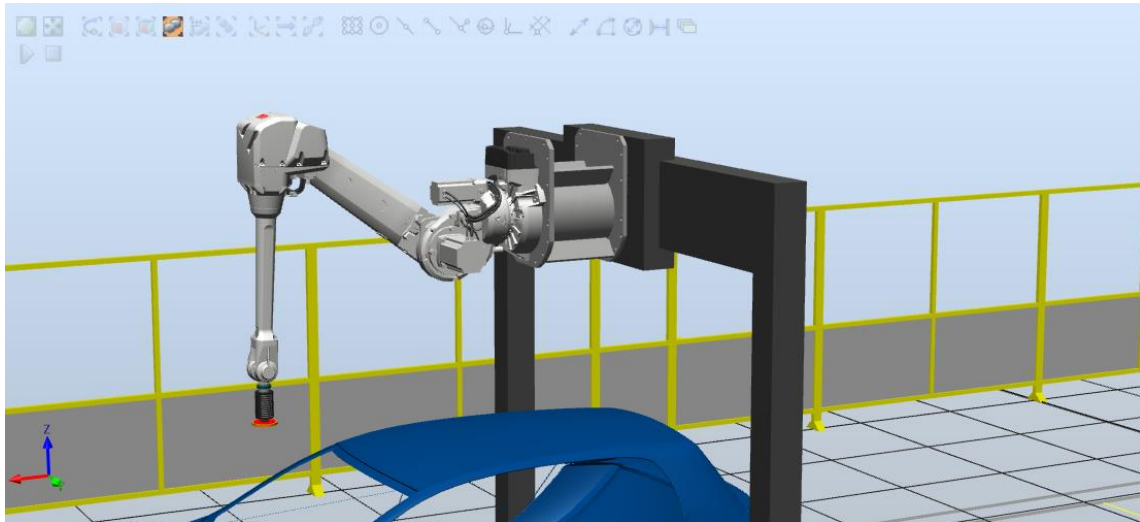


Figura 5.11: Robot en reposo tras pulido

A continuación el puente móvil de la figura 5.12 regresará a su posición inicial para sincronizarse con la siguiente carrocería.

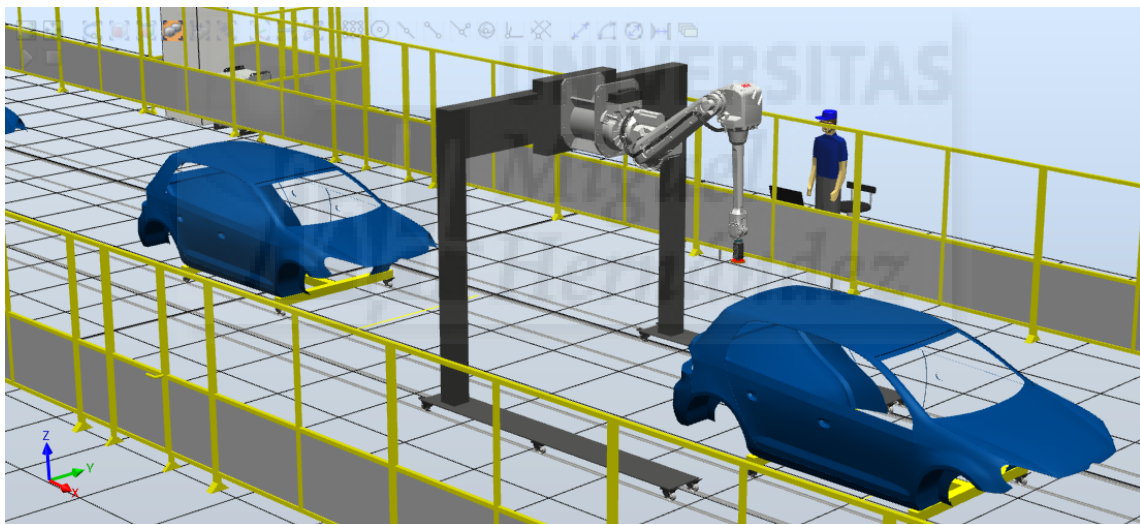


Figura 5.12: Puente móvil regresando a su posición

La carrocería de la figura 5.13 continúa su camino a través de los raíles.

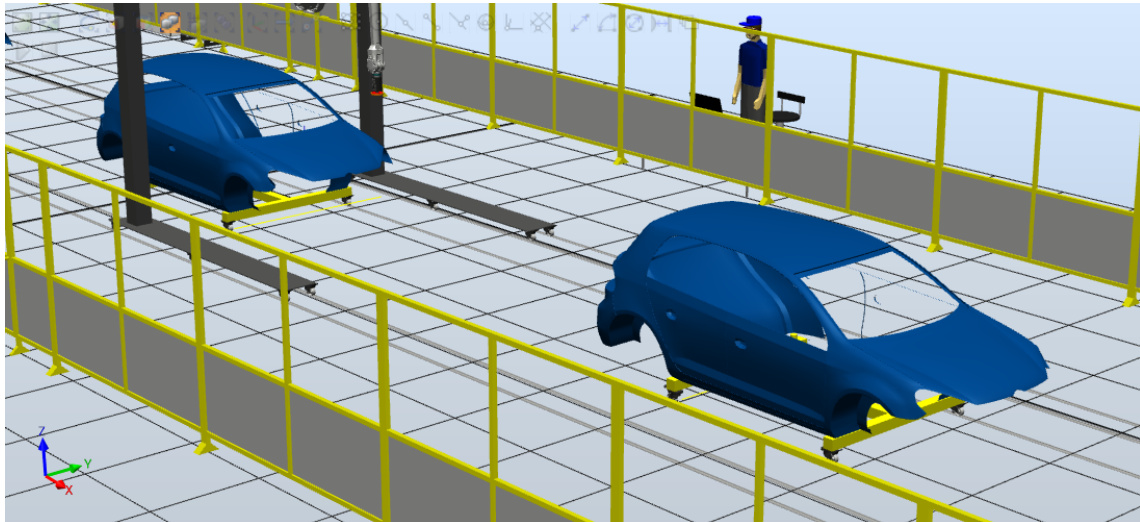


Figura 5.13: Carrocería tras finalizar el pulido

Una vez detectado la siguiente carrocería de vehículo de la figura 5.14, el puente móvil vuelve a sincronizarse con esta.

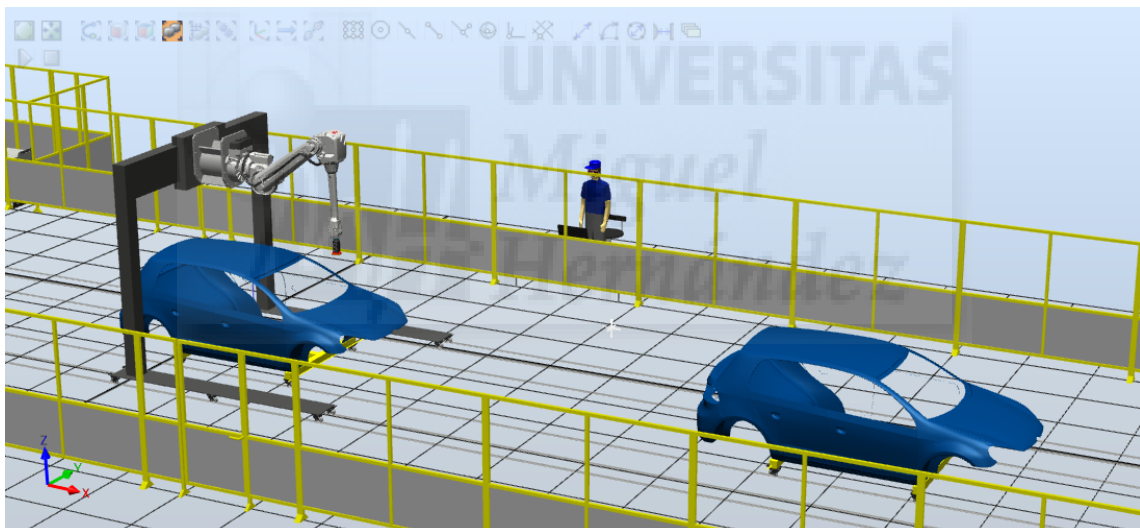


Figura 5.14: Sincronización puente móvil con la segunda carrocería

Este proceso se irá repitiendo sucesivamente.

Una vez que los vehículos han sido pulidos continúan su camino a través de los carriles en la figura 5.15.

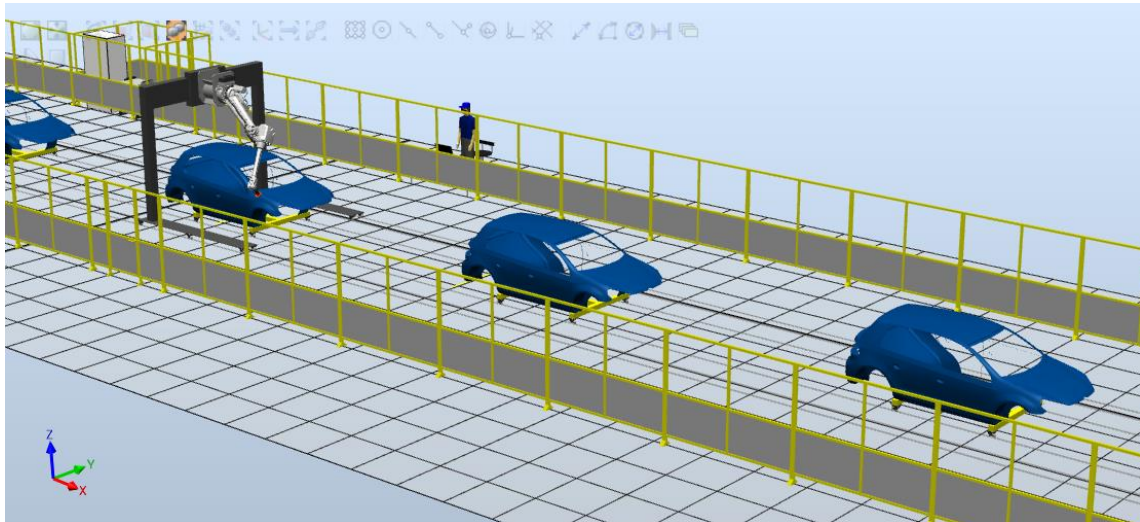


Figura 5.15: Carrocerías finalizado el pulido

Al final de la estación se encuentra un sensor, que detecta objetos cuando el rayo es cortado. Este sensor de la figura 5.16, realiza la eliminación del objeto de la estación.

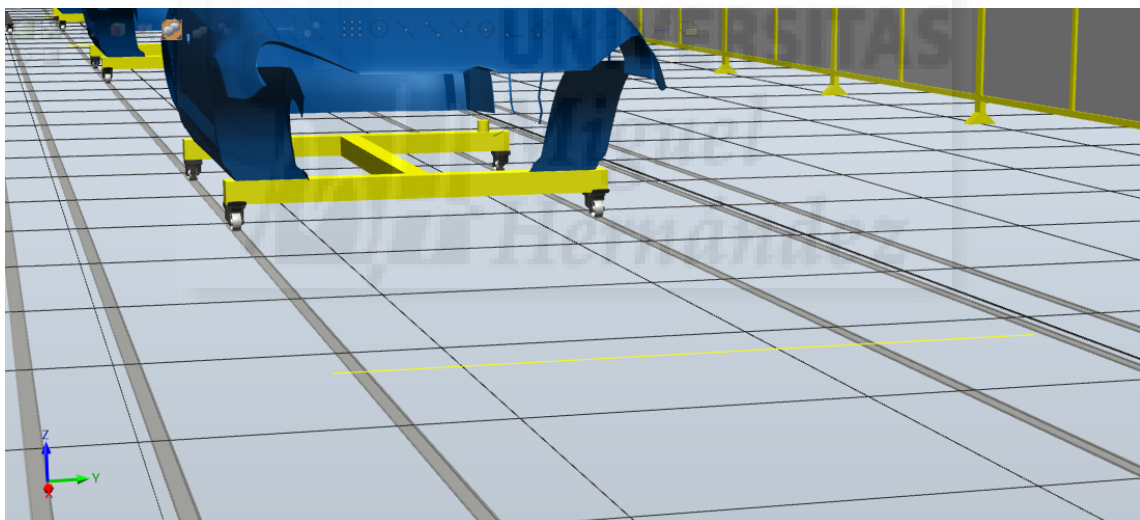


Figura 5.16: Sensor al final de los carriles

El sensor al detectar la bancada con la carrocería como en la figura 5.17, esta desaparecerá.

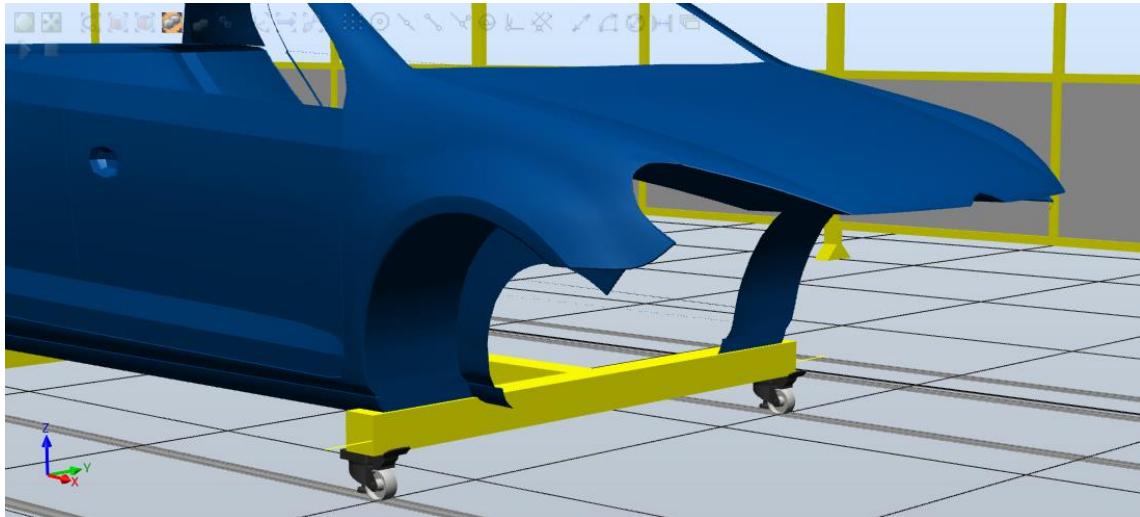


Figura 5.17: Detección bancada

La simulación de este proceso puede ser infinita, hasta que el usuario decida parar o detener la simulación en cualquier momento.

5.2 Simulación de la Estación de Salpicaderos

En este apartado se muestra la simulación de la estación de pulido de salpicaderos, en la figura 5.18 se puede observar la estación antes de iniciar la simulación.

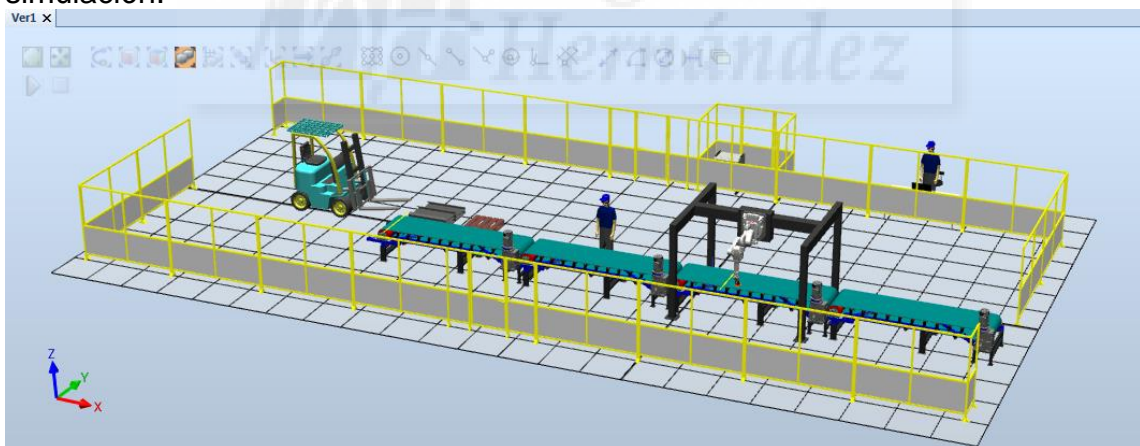


Figura 5.18: Estación Salpicaderos

En primer lugar el robot se colocará en su posición inicial si no está en ella, se pondrá en marcha la cinta transportadora y a los 4 segundos aparecerá la primera bandeja con el salpicadero como en la figura 5.19.

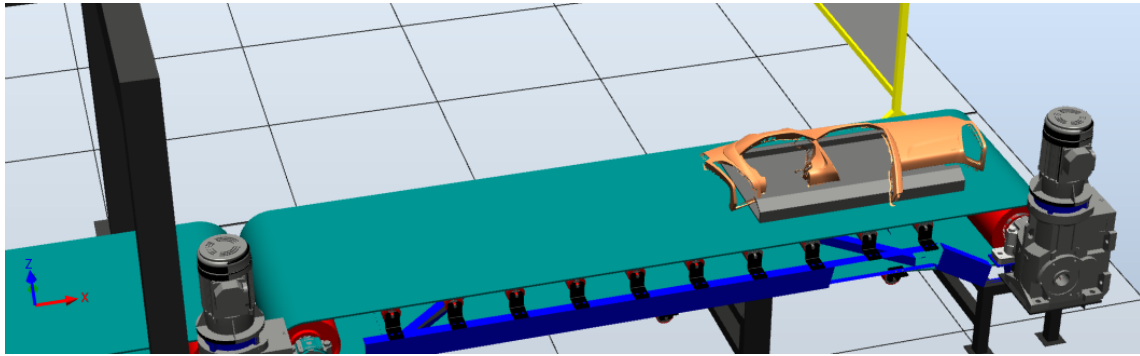


Figura 5.19: Primera bandeja con salpicadero

Transcurrido unos 6 segundos aparecerá la segunda bandeja de la figura 5.20, así sucesivamente.

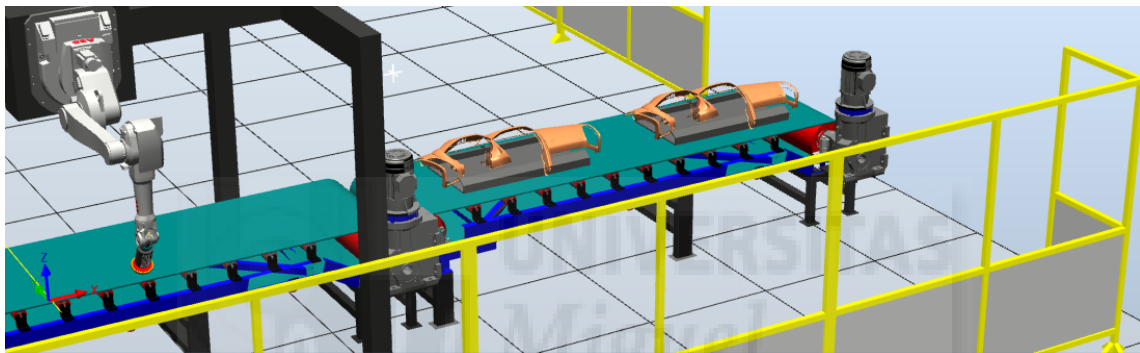


Figura 5.20: Segunda bandeja sobre la cinta transportadora

En la figura 5.21, las bandejas con los salpicaderos continúan sobre la cinta transportadora.

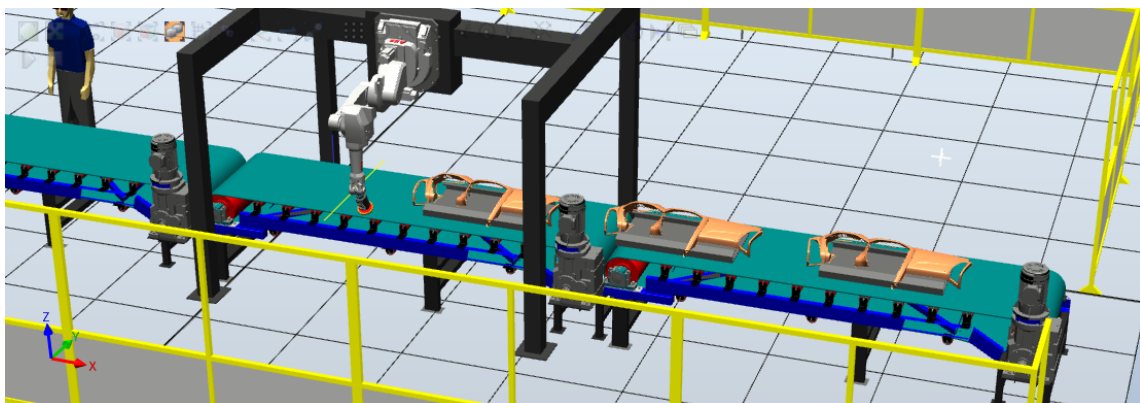


Figura 5.21: Salpicaderos sobre la cinta transportadora

El sensor de la figura 5.22 es el encargado de detectar cuando un objeto corta el rayo de este.

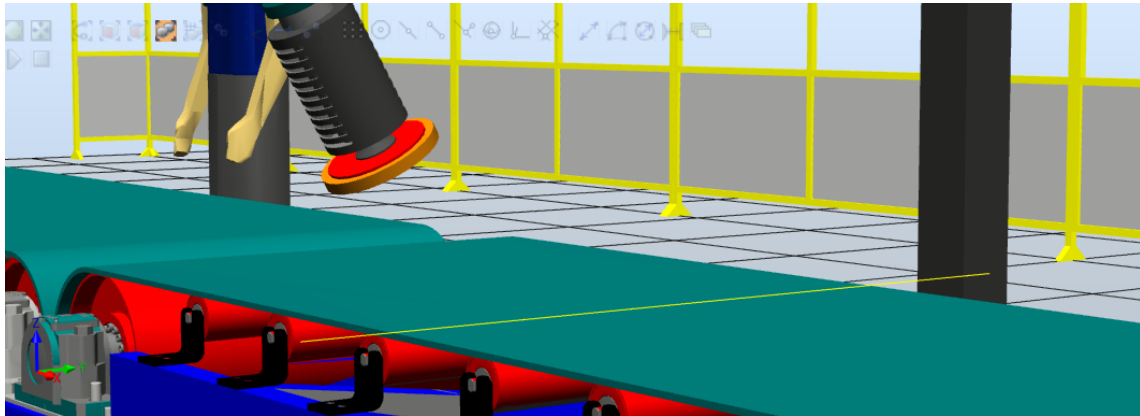


Figura 5.22: Sensor detecta objetos

Cuando la bandeja con salpicadero ha sido detectada por el rayo, las cintas transportadoras se pararán como en la figura 5.23.

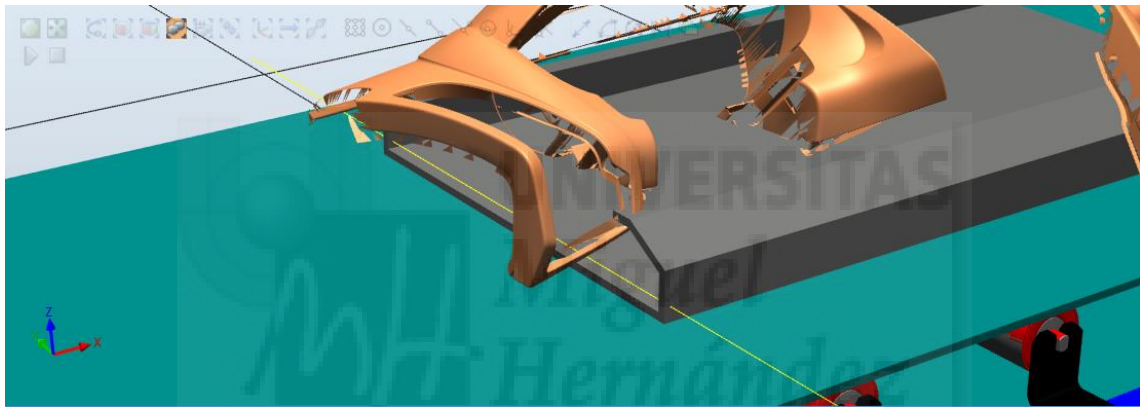


Figura 5.23: Parada de la cinta

Tras pararse la cinta, el robot de la figura 5.24 se pondrá en marcha para realizar su tarea de pulido.

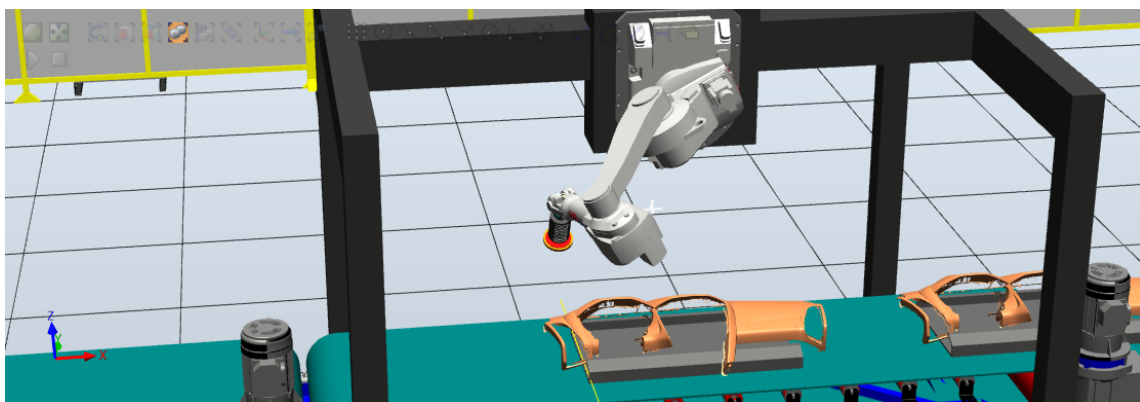


Figura 5.24: Robot en marcha

El robot realizará el pulido de los bordes de los salpicaderos como en las figuras 5.25 y 5.26.

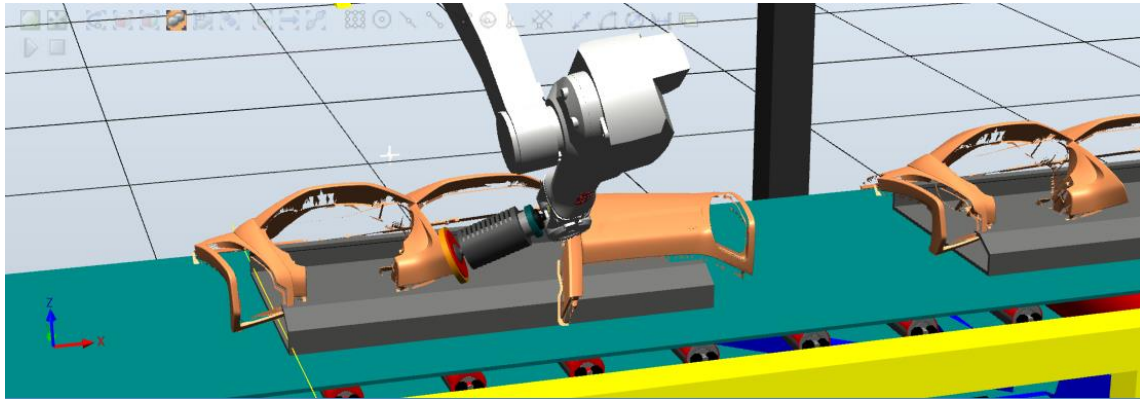


Figura 5.25: Robot realizando el pulido en salpicadero

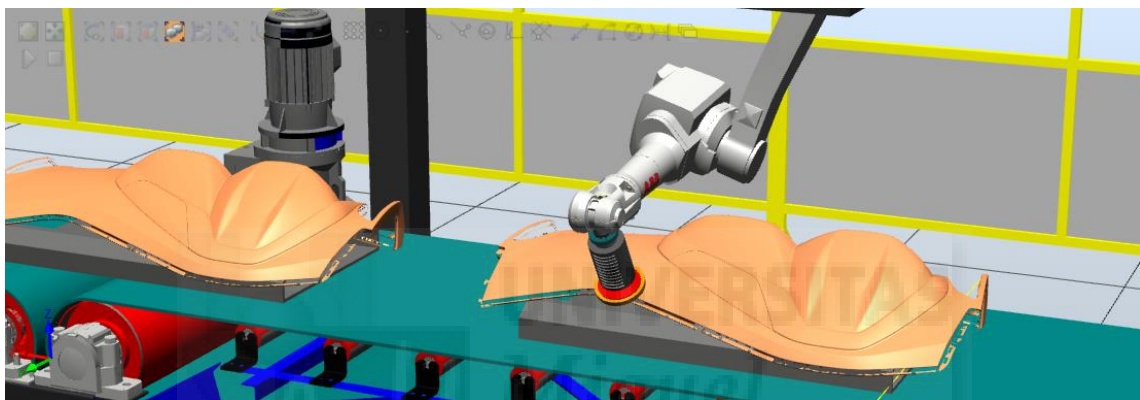


Figura 5.26: Robot realizando el pulido en parte posterior salpicadero

En la figura 5.27 vemos el detalle de la herramienta del robot realizando el pulido sobre superficie.

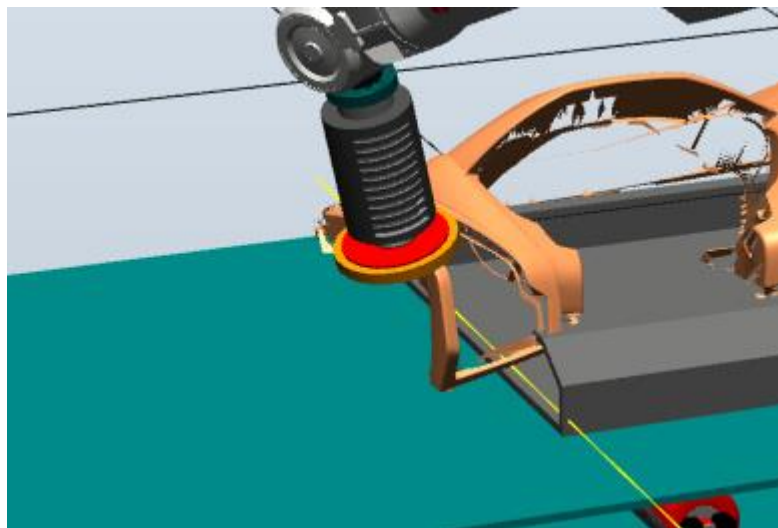


Figura 5.27: Detalle de la herramienta de pulido

Al finalizar el pulido por todos los bordes del salpicadero, el robot de la figura 5.28 vuelve a su posición inicial.

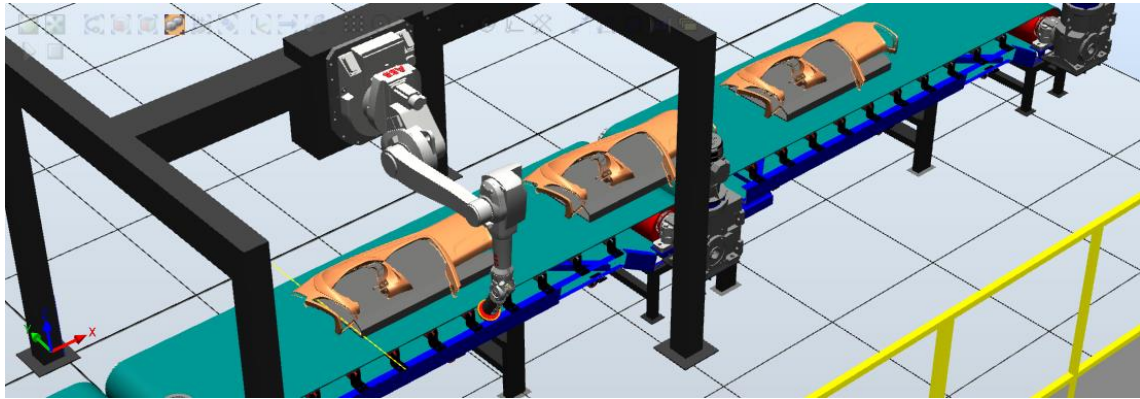


Figura 5.28: Robot en su posición inicial

Una vez que el robot se encuentre en su posición inicial, las cintas transportadoras se pondrán en marcha de nuevo, moviendo los salpicaderos junto a sus bandejas por la cinta como en la figura 5.29.

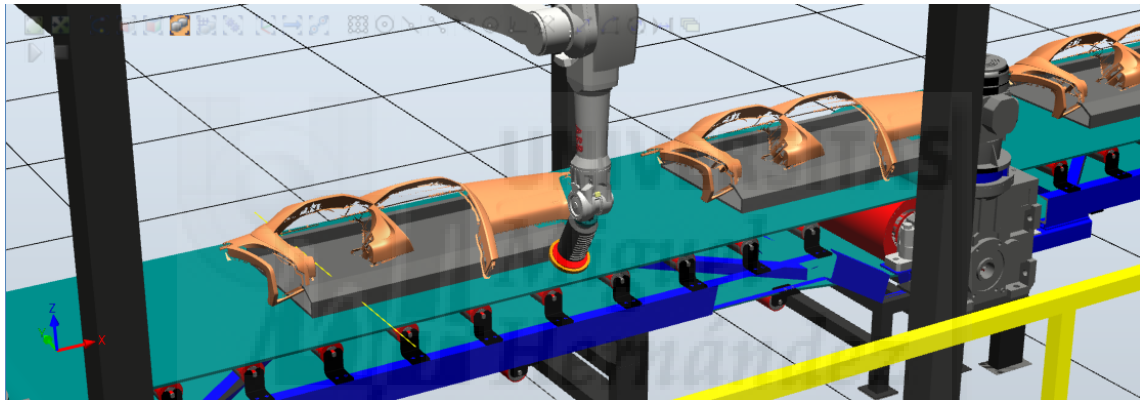


Figura 5.29: Cintas transportadoras en marcha de nuevo

Este proceso se irá repitiendo con cada uno de los salpicaderos, figura 5.30.

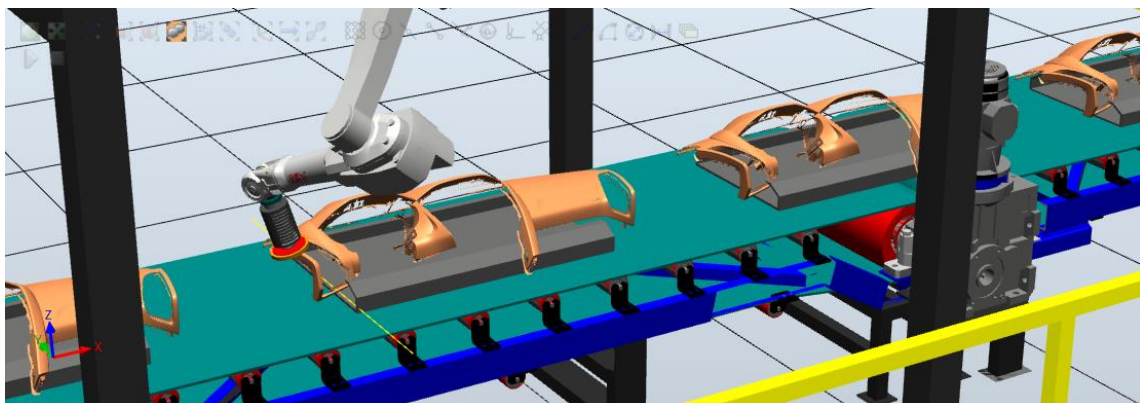


Figura 5.30: Proceso de pulido en salpicaderos

Al final de las cintas transportadoras se encuentra el sensor de la figura 5.31, detecta los objetos cuando el rayo se corta.

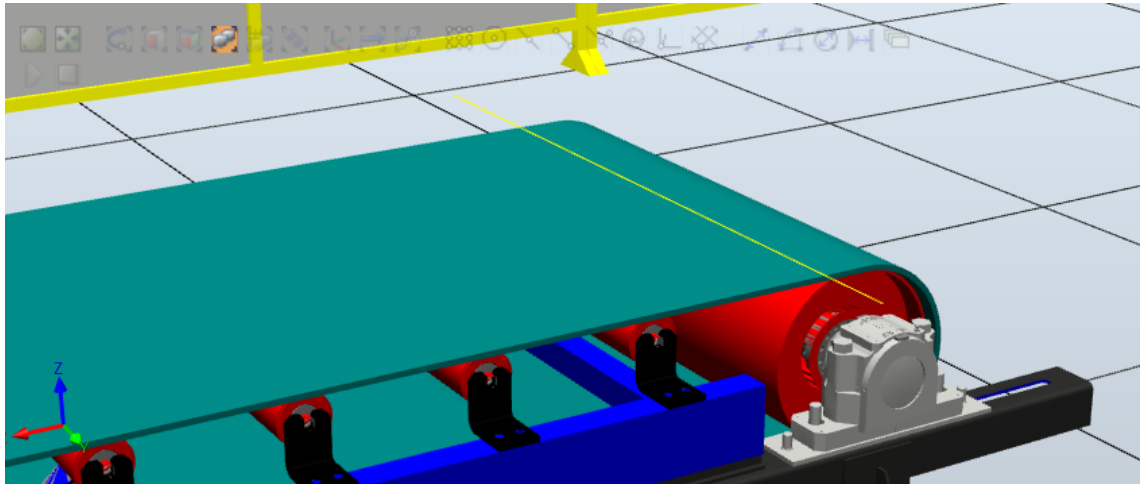


Figura 5.31: Sensor final cintas transportadoras

El objeto al llegar al final de la cinta es detectado por el sensor de la figura 5.32 y el operario separa el salpicadero de la bandeja que lo portaba, colocando la bandeja en el pale donde será retirado por la carretilla elevadora.

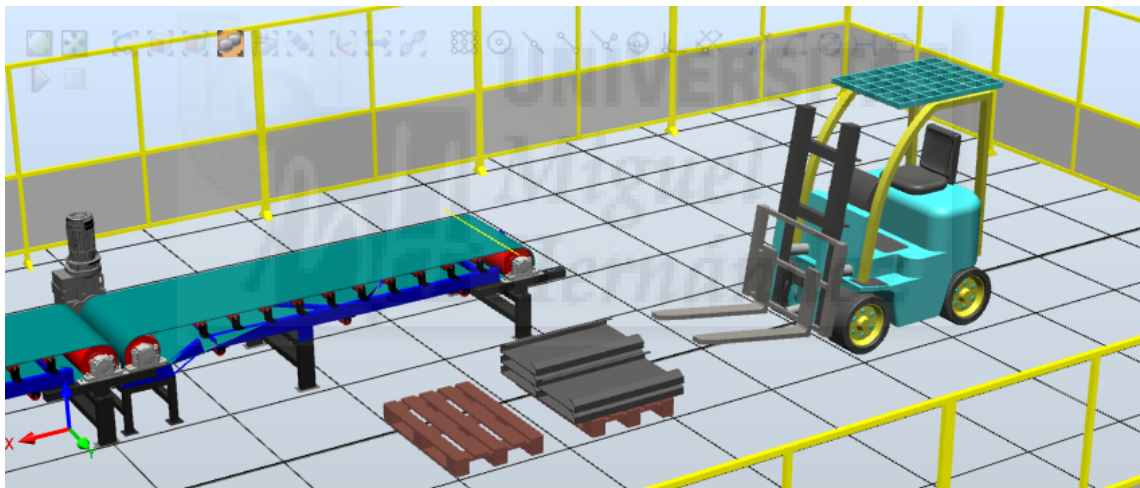


Figura 5.32. Pale con bandejas

La simulación de este proceso puede ser infinita, hasta que el usuario decida parar o detener la simulación en cualquier momento.

6. CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO

6.1. Conclusiones

Una vez finalizado el proyecto fin de Master, podemos concluir que se ha cumplido el objetivo general, así como los requisitos mínimos inicialmente planteados.

El software RobotStudio 6.05, es uno de los más utilizados en la programación “offline” en estaciones robóticas industriales, que permite realizar simulaciones muy cercanas a la realidad en un escenario virtual.

Para cumplir con las fases del trabajo y debido a la complejidad de utilización del programa ha sido necesarios adquirir unos conocimientos previos de robótica y el seguimiento minucioso de unos videotutoriales de aprendizaje para entender el funcionamiento de dicho programa.

Este programa nos ha permitido adquirir la destreza en el desarrollo de dos estaciones robotizadas para el pulido de superficies metálicas, que puede ser de gran utilidad a la hora de implementarlas en un escenario real, ya que hoy en día no están incorporados estos procesos en la industria del automóvil.

Para la generación de las simulaciones en las estaciones se ha utilizado elementos comunes de la industria como vallas, sensores, planos de trabajo, etc. de las librerías de RobotStudio ABB. Debido a no disponer de todas las piezas y elementos necesarios para la creación de las estaciones robóticas en estas librerías específicas, ha sido necesario descargarlas de otras librerías de CAD. Además se han diseñado las estructuras de soporte de los robots y herramienta rotativa de pulido en el programa Inventor 2015, llevando a cabo la captura y el envío de datos al controlador de manera satisfactoria.

En el transcurso del trabajo han surgido algunos inconvenientes, a la hora de la simulación del pulido en ambas estaciones, el brazo daba giros innecesarios en el cambio de trayectorias, habiendo sido parcialmente corregidos estos defectos a base de prueba y la creación de más puntos sobre las superficies. Además el servidor junto a la licencia dejó de funcionar, dificultando el trabajo hasta que se pudieron subsanar.

La aplicación de los resultados de estos modelos simulados de pulido en estaciones reales, frente al pulido manual o pulido con máquinas auxiliares, supondría aumentar la seguridad y las condiciones de trabajo de los operarios como ventajas principales de estos sistemas robotizados.

Esta metodología incrementa la productividad, destacando la reducción del tiempo de proceso, alta flexibilidad, homogeneidad y repetitividad de los resultados y por tanto un aumento en la calidad final del producto.

6.2. Líneas futuras de trabajo

Finalmente se enunciarán las principales líneas futuras derivadas de este trabajo como punto de partida, que debido a su alcance no ha permitido poder desarrollar por completo todas las posibilidades que ofrece.

Esta metodología predictiva desarrollada en este trabajo se podrá aplicar a un área de trabajo real. Se propone implantar en la industria del automóvil estas dos estaciones, ver su comportamiento, optimizarlos y aplicar las correcciones necesarias para un buen funcionamiento. Estas optimizaciones y correcciones se pueden hacer sin necesidad de estar físicamente trabajando con ellos, sin afectar a la producción, lo que supone una gran ventaja.

Además, se deberá cumplir con las condiciones de métodos y tiempos para ser acopladas a la línea automatizada convencional de producción existente.

La herramienta de pulido rotativa acoplada a la muñeca del robot es uno de los elementos esenciales del sistema, por ello deberán ser primero testeadas manualmente con objeto de poder definir su funcionalidad y posteriormente serán de nuevo testeadas como elementos terminales de la muñeca del robot para evaluar su comportamiento en un brazo robótico real.

Se realizará un detallado estudio previo de los defectos más comunes encontrados en las carrocerías, de las técnicas y materiales más adecuados que exigen estos procesos de pulido.

Uno de los mayores problemas que se puede presentar a la hora de su implantación en una estación real es la desviación y vibración de las herramientas en la trayectoria de trabajo; dependiendo fundamentalmente de la rigidez del robot, la magnitud de las fuerzas de pulido y la dirección de la acción predominante. Por ello las interfaces hápicas deben controlar especialmente la retroalimentación de fuerzas de la estructura y las velocidades del robot. La utilización de esta fuerza regulable, permite aplicar una fuerza concreta en un punto específico del recorrido de pulido mejorando la precisión y calidad superficial del mecanizado robotizado, en caso de aplicar demasiada fuerza la superficie se calienta y puede llegar a deformarla, siendo los resultados totalmente contrarios a los esperados y el robot implantado no cumplirá los objetivos.

Se propone implantar una segunda línea de transportadores de bandas, junto a la de pulimentado de salpicaderos donde otro robot acople la carcasa con la estructura del salpicadero pulido, con lo que tendremos el salpicadero listo para colocar en el vehículo.

Hay que dar solución al polvo metálico y el óxido de aluminio abrasivo en suspensión que se produce, constituye un problema especialmente grave en las operaciones de pulido, y lo más eficaz para su eliminación sería la ventilación local por extracción o incorporar al brazo del robot un extractor de polvo, recogiendo en el mismo punto en que se produce.

7. BIBLIOGRAFÍA

- [1] ABB Robotics. (2017) Manual del operador. RobotStudio 6.05. 648 p.
- [2] ABB Robotics. (2016) IRB 4600 Industrial Robot. 2 p.
- [3] ABB Robotics. (2014) IRB 140 Industrial Robot. 2 p.
- [4] ABB Robotics. (2014) IRB 1600 The highest performance 10 kg robot. 2 p.
- [5] ABB Robotics. (2010) Manual de referencia técnica. Instrucciones, funciones y tipos de datos de RAPID. RobotWare 5.13. 1322 p.
- [6] ABB Robotics. ABB Flexible Automation AB. RAPID Reference On-line Manual. 757 p.
- [7] Ariza, C. Amaya, D. (2011) Laboratorio remoto para la enseñanza de la programación de un robot industrial. Ing. USBMed, Vol. 2, No. 1, Ene-Jun, pp. 33-39.
- [8] Ávila, J. A. (2015) Modelado de una célula robótica con fines educativos usando el programa Robot-Studio. Universidad de Valladolid. TFG Escuela de Ingenierías Industriales. 103 p.
- [9] Barrientos, A. et al. (2007) Fundamentos de Robótica. Madrid. McGraw-Hill/Interamericana de España, S.A.U. 2ª edición. 624 p.
- [10] Beaupre, M. (2015) Collaborative Robot Technology and Applications. KUKA Robotics Corporation.
- [11] Bernal, M.E et. al. (2015) Caso productivo en una celda de manufactura flexible. Scientia et Technica año XX, vol. 20, junio, pp. 139-145.
- [12] Bredin, C. (2005) Compañeros de equipo. La funcionalidad MultiMove de ABB anuncia una nueva era en las aplicaciones de robots. Revista ABB Automatización Tecnológica Suecia nº 1, pp. 26-29.
- [13] Castillo, C. 2011 Disseny d'una cèl·lula de treball robotitzada que utilitzarà un robot industrial asea irb 120. Universitat Politècnica de Catalunya. TCM Escola Univeristària Politècnica de Mataró. 236 p.
- [14] CEPYME. (2015) Manual de Procedimientos para Evaluación de Riesgos y Condiciones de Trabajo en el Subsector de Metal Fabricación de Carrocerías y Chasis para Vehículos de Transporte de Viajeros y Mercancías. Zaragoza. Fundación para la prevención de Riesgos Laborales. CEPYME. 317 p.
- [15] Córdoba, A.J. (2016) Puesta en marcha de brazo robótico y desarrollo de aplicaciones. Universidad de Jaén. TFG Escuela Politécnica Superior de Linares. 145 p.
- [16] COTEC (2006) Robótica y automatización. Documentos COTEC sobre oportunidades tecnológicas. Madrid, Graficas Arias Montoro, S.A.
- [17] Doria, J.J. et al. (2013) Implementación De Proceso De Pick And Place Mediante Grúa-Robot Aplicado A Celda De Manufactura Flexible. Scientia et Technica Año XVIII, Vol. 18, No. 4, pp. 633-639.

- [18] Dosta, J. et al. (2003) Defectos en la Aplicación de Pinturas en Carrocerías. Técnica Industrial 250 – septiembre.
- [19] Gámez, J. et al. (2007) Estimación de la fuerza de contacto para el control de robots manipuladores con movimientos restringidos. Revista Iberoamericana de Automática e Informática Industrial. Vol. 4, Núm. 1, Enero 2007, pp. 70-82.
- [20] García, J.P. et al, (1999) Secuenciación para la línea de montaje de autmóviles. Un caso práctico. Barcelona. III Jornadas de Ingeniería de Organización. 16-17 septiembre.
- [21] García, M. et al. (2014) Empleo de robots en sistemas de fabricación flexible. 3C Tecnología (Edición núm. 12) Vol.3 – Nº 5, pp. 283-295.
- [22] García, S. et al. (2011) Robótica General. Centro de Formación de Seat. Norwww.infoPLC.net.
- [23] Gil, A. (2017) Practical session 5: Rapid programming. ARTE. Universidad Miguel Hernández. 18 p.
- [24] Gil, A. (2017) Tema 8: Introducción a Rapid. Control de robots y sistemas sensoriales. Universidad Miguel Hernández. 94 p.
- [23] Gil, P. et al. (2012) Recursos y herramientas didácticas para el aprendizaje de la robótica. Universidad de Salamanca Revista TESI, 13(2), pp. 18-47.
- [24] Heyer, C. (2010) Human-Robot Interaction and Future Industrial Robotics Applications. IEEE/RSJ International Conference on Intelligent Robots and Systems October pp.18-22, Taipei, Taiwan.
- [25] IDF (2015) Control de calidad mediante visión artificial. Universidad Politécnica de Valencia. Instituto de Diseño y Fabricación automoción. 8 p.
- [26] Iglesias, I. (2015) Desarrollo de una metodología predictiva de precisión y acabado superficial aplicada al mecanizado robotizado. Universidad Nacional de Educación a Distancia. Tesis Doctoral, 103 p.
- [27] ISA UMH. Ejemplo de programación en RAPID. Control de robots y sistemas sensoriales. Universidad Miguel Hernández. 14 p.
- [28] Khamis A. (2006) Lenguaje RAPID. Robotics Application Programming Interactive Dialogue. Automatización Industrial II. Universidad Carlos III de Madrid. 44 p.
- [29] Kosuge, K. et al. (1993) Robot-Human Collaboration for New Robotic Applications. Japan. Nagoya University. Department of Mechano-Informatics and Systems. pp. 713-718.
- [30] Lamas, J. (2015) Sistema de posicionamiento, automatización, muestreo y control para la fabricación en tres dimensiones utilizando el láser como herramienta. Universidad de la Coruña, Tesis Doctoral. 187 p.
- [31] Laplace, J. (2015) Human-robot interaction finally reaches the work floor. www.Humarobotics.com. 17 p.

- [32] Márquez, J.J. Pérez, J.M. (1996) Desarrollo de un sistema automatizado para el pulido de moldes. Universidad Politécnica de Madrid. Depto. De Ingeniería de Fabricación. Información Tecnológica, Vol nº 3.
- [33] Martínez, A. et al. (2000) Las tecnologías de automatización flexible en España. Boletín ICE económico nº 2654, pp. 7-16.
- [34] Martínez, V. (2008) Simulación de procesos de producción robotizados mediante el programa Robotstudio. Universidad Carlos III de Madrid. PFC Escuela Politécnica Superior. 179 p.
- [35] Mato, M.A. (2014) Simulación, control cinemático y dinámico de robots comerciales usando herramienta de Matlab, Robotic Tolboc. Universidad de Valladolid. TFG Escuela de Ingenierías Industriales. 110 p.
- [36] Merino, L. et al. (2010) Robótica cooperativa e integración con sensores en el ambiente. Aplicaciones en entornos urbanos. Universidad Pablo de Olavide Sevilla y Escuela Superior de Ingenieros, Universidad de Sevilla. Grupo de Robótica, Visión y Control Escuela Politécnica Superior.
- [37] Muñelo, G. 2015. Simulación de una célula robotizada para la fabricación de piezas mecanizadas en un entorno industrial, utilizando RobotStudio. Universidad de Valladolid. TFG Escuela de Ingenierías Industriales. 147 p.
- [38] Olier, I. (1999) Una introducción a la robótica industrial. Universidad Militar de Nueva Granada. Revista de la Facultad de Ingeniería noviembre 1999, p.p. 53-67.
- [39] Olier, I. (1999) Diseño del robot industrial. Universidad Militar de Nueva Granada. Revista Ciencia e Ingeniería noviembre 1999, p.p. 9-18.
- [40] Pérez, E. (2015) Los sistemas SCADA en la automatización industrial. Revista Tecnología en Marcha. Vol. 28, Nº 4, Octubre-Diciembre, pp. 3-14.
- [41] Pires, J.N. et al. (2008) Control difuso de fuerza en robótica industrial. Revista Ingenierías, octubre-diciembre 2008, Vol. XI Nº 41.
- [42] Poch, R (2009) Estudio de la implantación de un robot antropomórfico para la mecanización de elevada complejidad. Universidad Politécnica de Cataluña. PFC. Escuela Técnica Superior de Ingenieros Industriales y Aeronáuticos de Tarrasa. 121 p.
- [43] Ramos, A. (2016) Diseño, programación y simulación de estaciones robotizadas industriales con Robotstudio. Universidad de Sevilla. TFG Escuela Técnica Superior de Ingeniería. 111 p.
- [44] Rosado, A. et al. (2008) Herramienta e-learning para la programación de robots mediante entorno web. @tic revista de innovación educativa nº 1. Universidad de Valencia pp. 45-48.
- [45] Rubio, J.C. (2006) Selección de una nueva arquitectura de control y supervisión de una célula robotizada de pintado de carrocerías. Universitat Politècnica de Catalunya. PFM Fundació CIM. 154 p.
- [46] Santos, P. (2014) Procesos de Pulido. Revista Carrocería y pintura, núm. 62 Octubre-Diciembre, p.p. 1-4.

[47] Soto, A. (2014) Robot pintor artístico. Universidad de Alcalá. TFG Escuela Politécnica Superior. 131 p.

[48] Vicente, D. (2016) Generación de escenarios y programación de brazos robóticos con RobotStudio y RAPID (ABB). Universidad de Zaragoza. TFG Escuela Universitaria Politécnica de Teruel. 57 p.

[49] Welinder, K. (1996) Metalurgia y Metalistería. Rectificado y Pulido de las Industrias Manufactureras. Enciclopedia de Salud y Seguridad en el Trabajo, p.p. 1-66.

Páginas web consultadas:

[50] <http://www.aitiip.com/noticias/robot-kuka-kr60-ha.html>

[51] <http://www.ayri11.com/abb/271-robotstudio.html>

[52] <https://es.scribd.com/document/230901923/Robotica-Industrial>

[53] <http://www.infoplz.net/blogs-automatizacion/item/102143-robots-colaborativos>

[54] <http://www.interempresas.net/MetalMecanica/Articulos/116573-Sistema-inteligente-acabado-piezas-utilidades-mediante-sistemas-pulido-desbarbado.html>

[55] <https://manuelguerrero.com/robots-colaborativos-la-industria-4-0/>

[56] <http://www.metalmecanica.com/temas/Pulido-automatico-de-moldes+7076234>

[57] <http://www.monografias.com/trabajos102/introduccion-utilizacion-robotstudio-abb/introduccion-utilizacion-robotstudio-abb.shtml#ixzz4pX4e6oBV>

[58] http://www.revistatope.com/158_art_PROVA_Eolica.html

[59] http://www.shl.ag/cms/es/pulido_automatico/

[60] <http://www.tamautomation.com/lang/esp/noticias.php>

[61] <https://www.universal-robots.com/es/casos-prácticos/paradigm-electronics/>

Vídeo-tutoriales:

<https://www.youtube.com/watch?v=oDnHaVMZjp4&list=PLVdvHpsfqw1bX194i7Slup6ri5se2668p>

<https://www.youtube.com/watch?v=4l2FT-MdqZU&list=PL7fQXFqChkTe0snNKx67OHyaZxYfdIRBz>

<https://www.youtube.com/watch?v=-T8YE9S8bHg>

<https://www.youtube.com/watch?v=bm-Tyz4Gmzw&list=PLE6CB81A3320750CB>

<https://www.youtube.com/watch?v=agbhTUxJAeg&list=PLEF161D764568EC82>

<https://www.youtube.com/watch?v=YBQFIX4wk8I>

<https://www.youtube.com/watch?v=1CvnROQ3LA8>

<https://www.youtube.com/watch?v=RNTSGDYGRr4>

<https://www.youtube.com/watch?v=FwgFHF8vRc>

<https://www.youtube.com/watch?v=Qri8IHGqon4>

<https://www.youtube.com/watch?v=-D1HBTr-8T4>

<https://www.youtube.com/watch?v=H56AUpT7ELI>



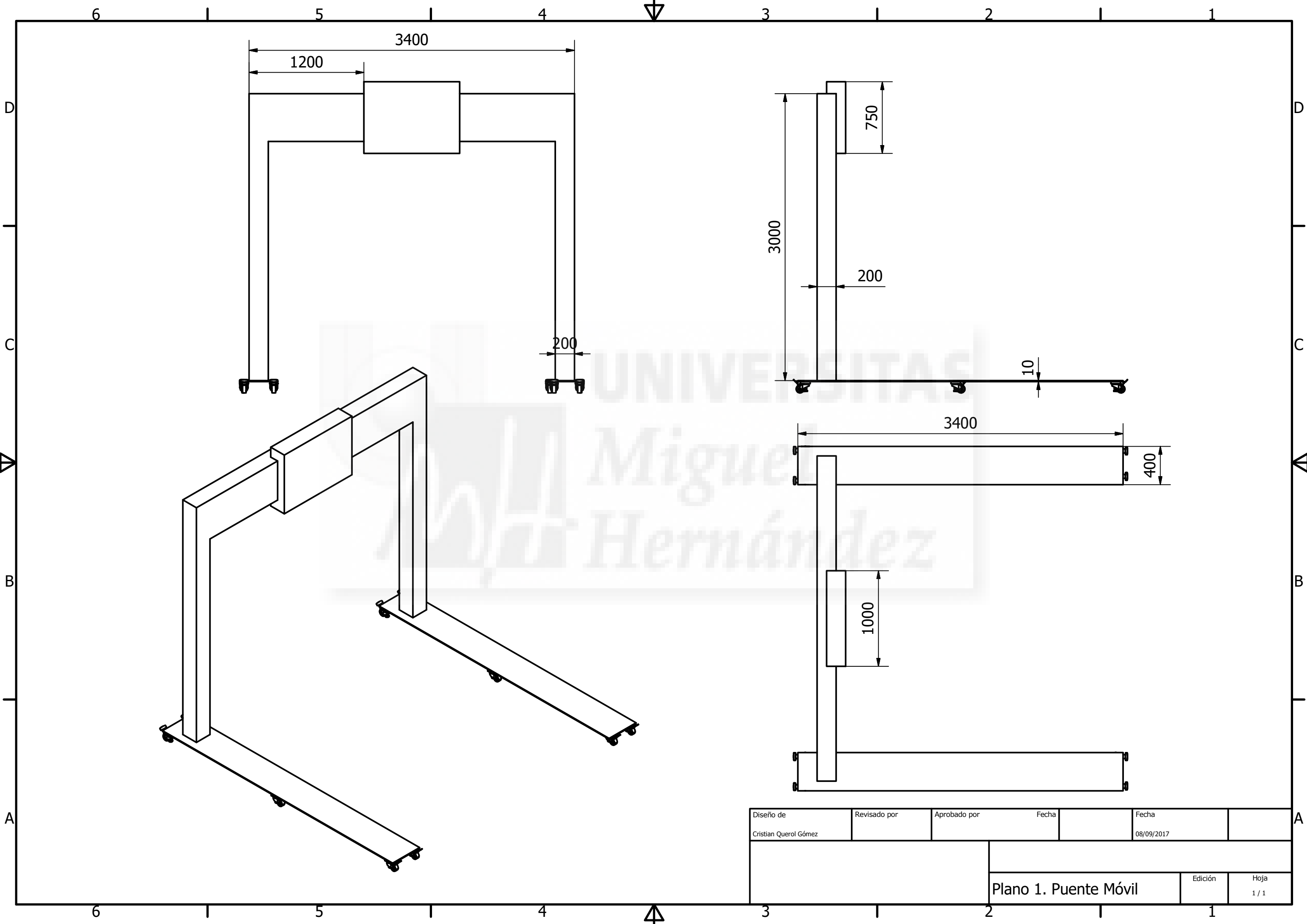


8. ANEXOS

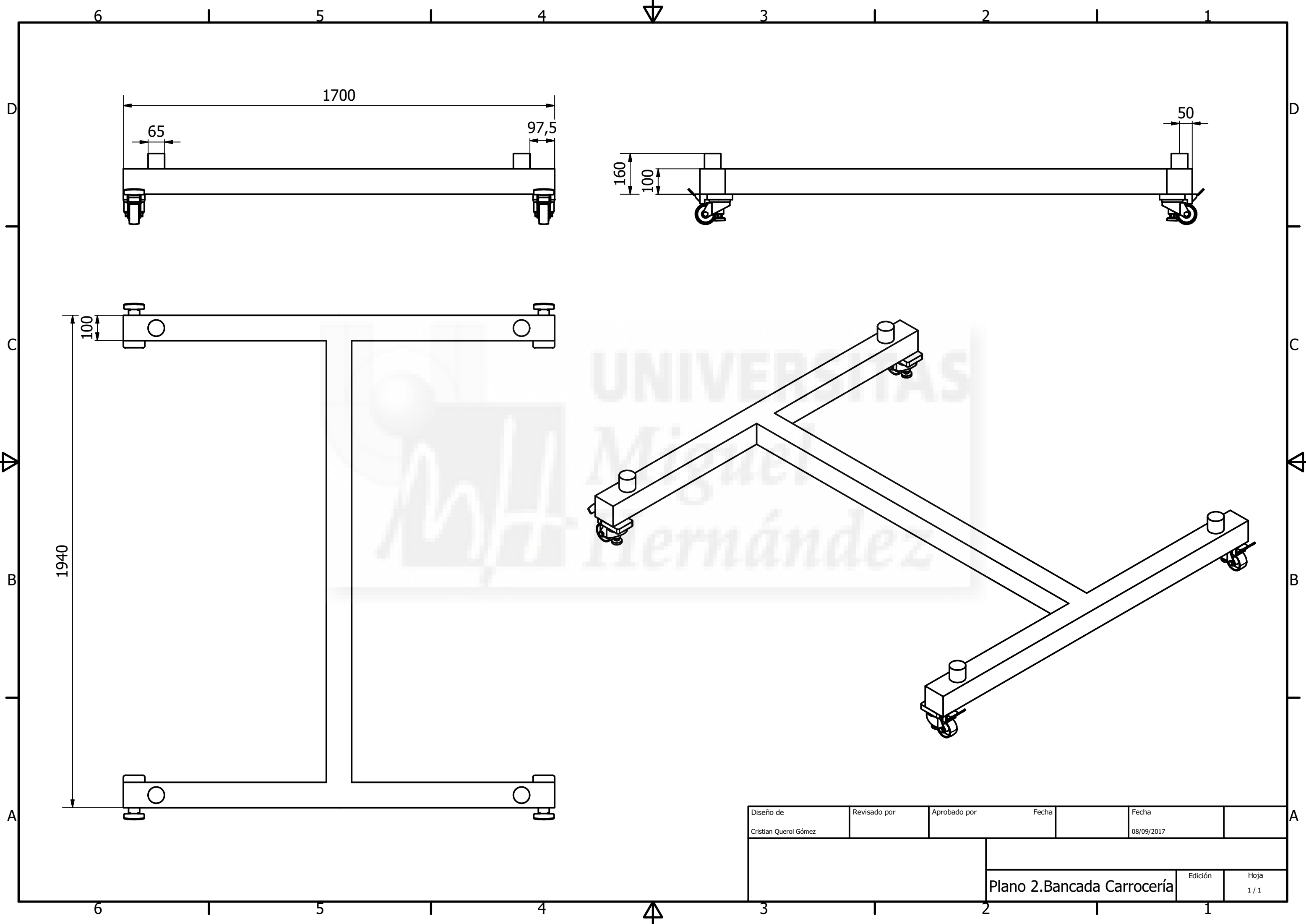
8.1 Planos

8.1.1 Planos de la Estación de Carrocerías

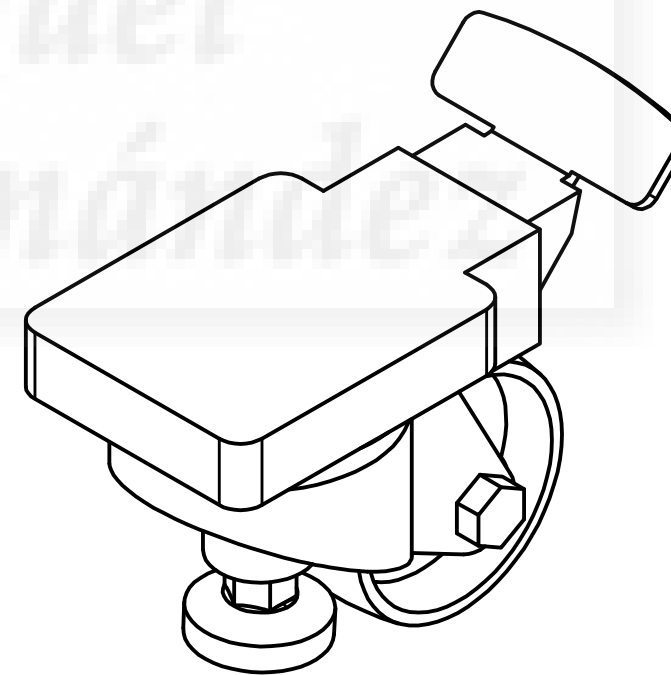
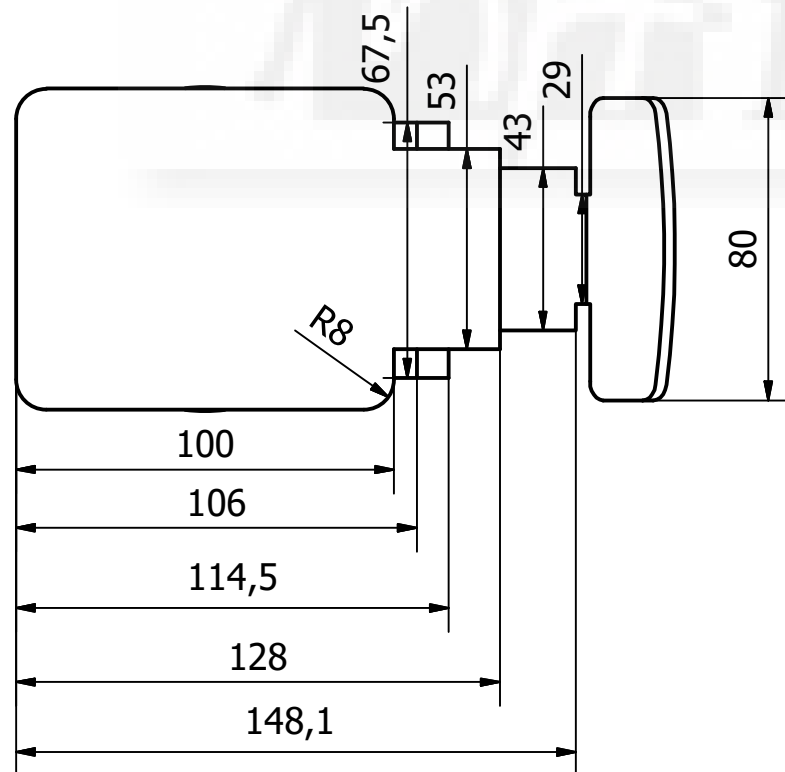
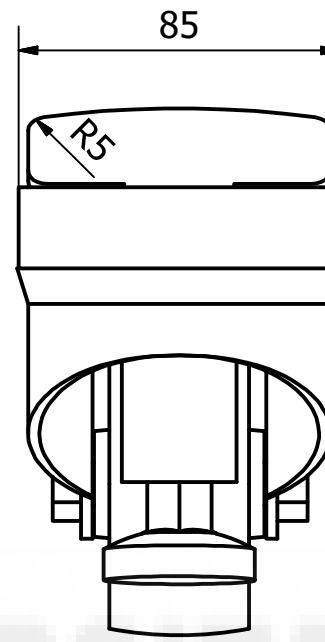
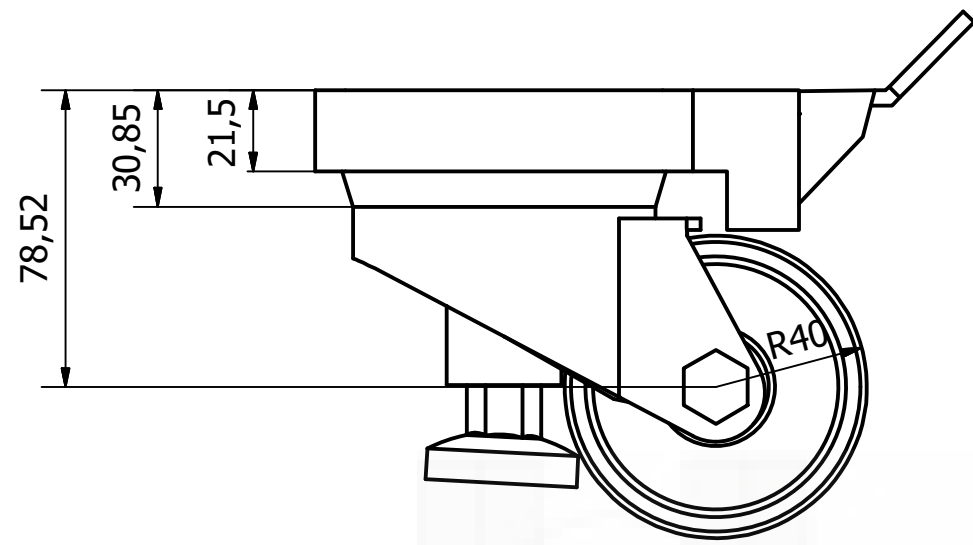




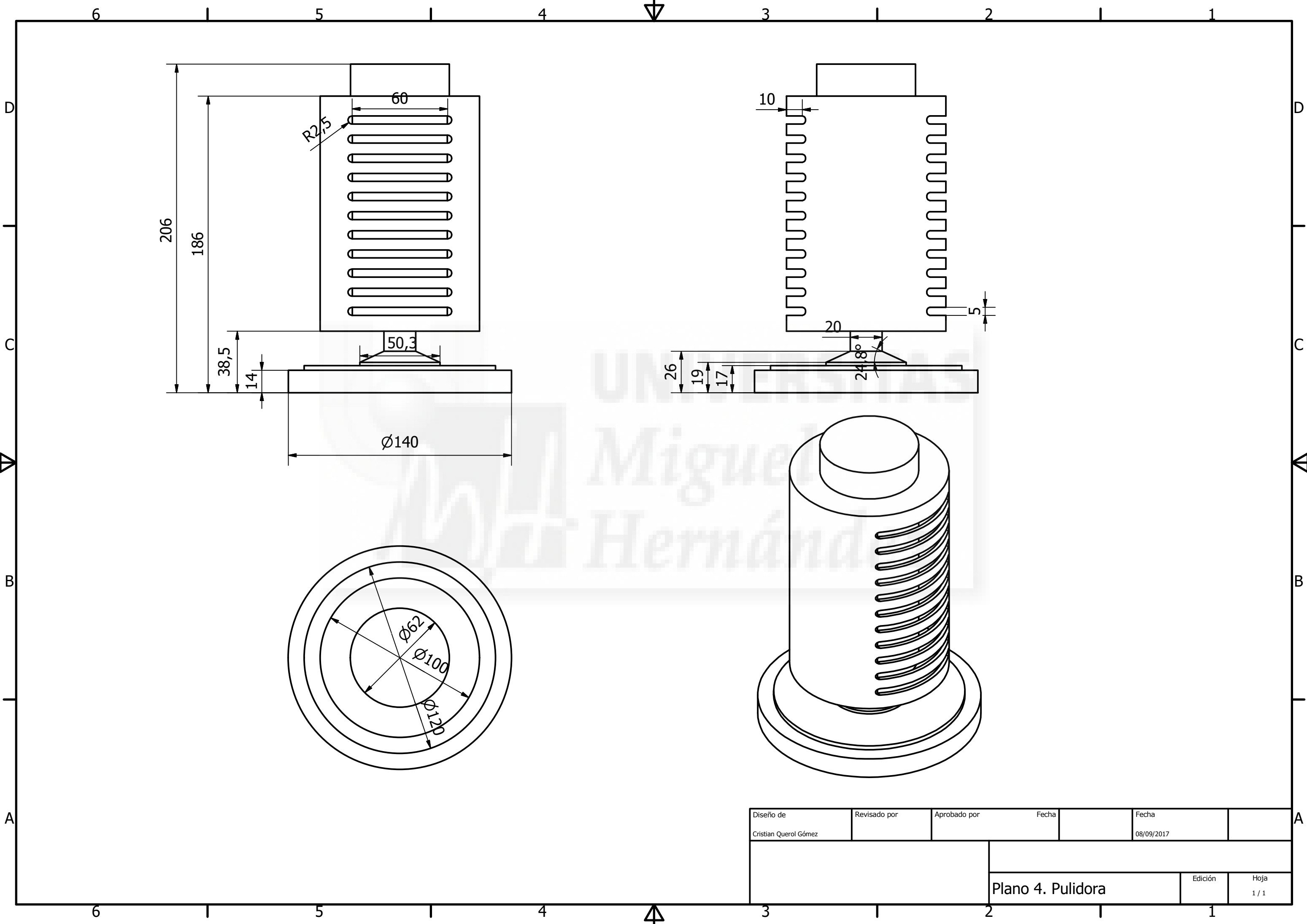
Diseño de Cristian Querol Gómez	Revisado por	Aprobado por	Fecha	Fecha 08/09/2017	
			Edición		Hoja 1 / 1
			Plano 1. Puente Móvil		

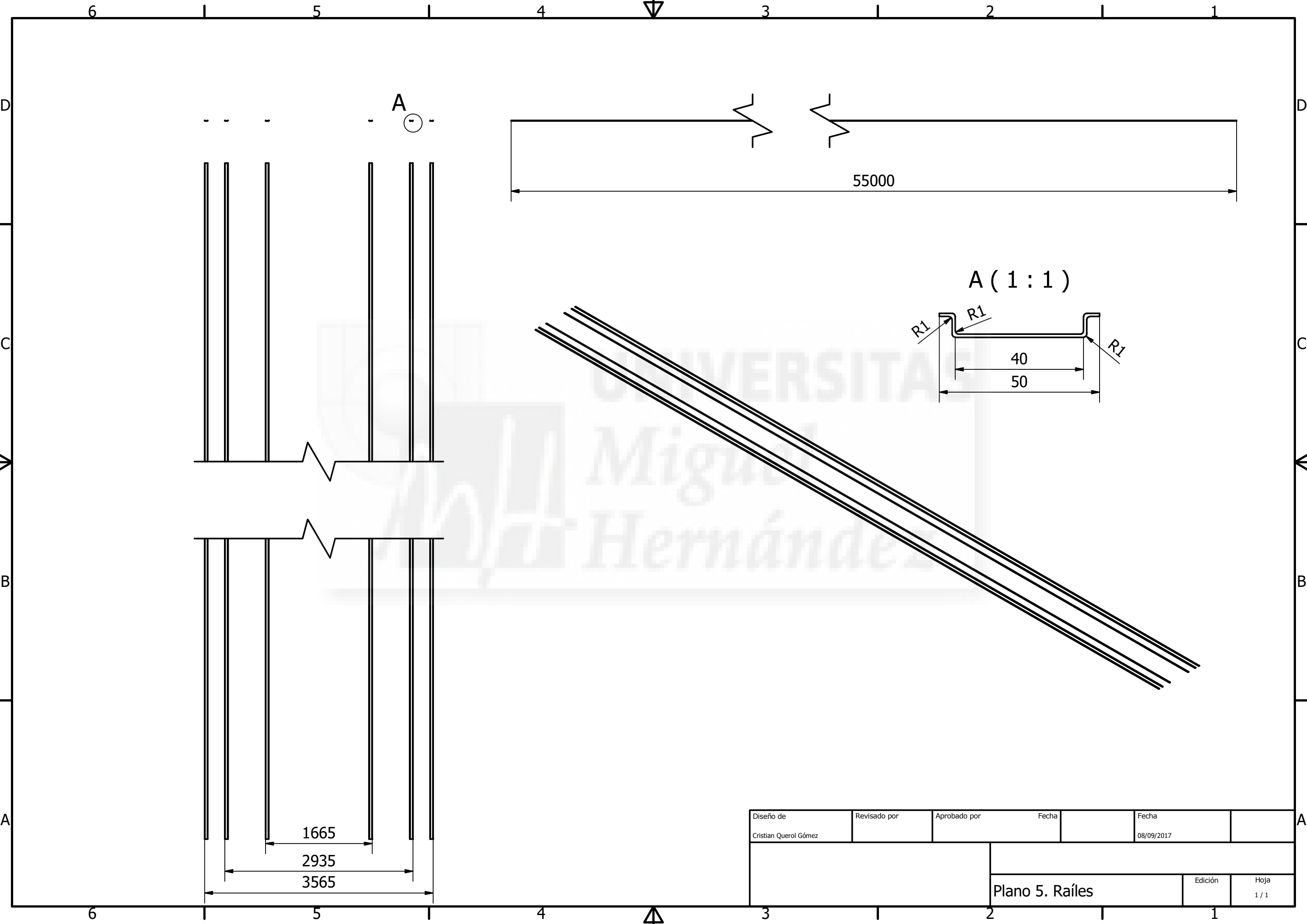


Diseño de	Revisado por	Aprobado por	Fecha	Fecha	
Cristian Querol Gómez				08/09/2017	
			Plano 2.Bancada Carrocería		Edición
					Hoja
					1 / 1



Diseño de Cristian Querol Gómez	Revisado por	Aprobado por	Fecha	Fecha 08/09/2017	
			Edición		Hoja 1 / 1
			Plano 3. Ruedas		

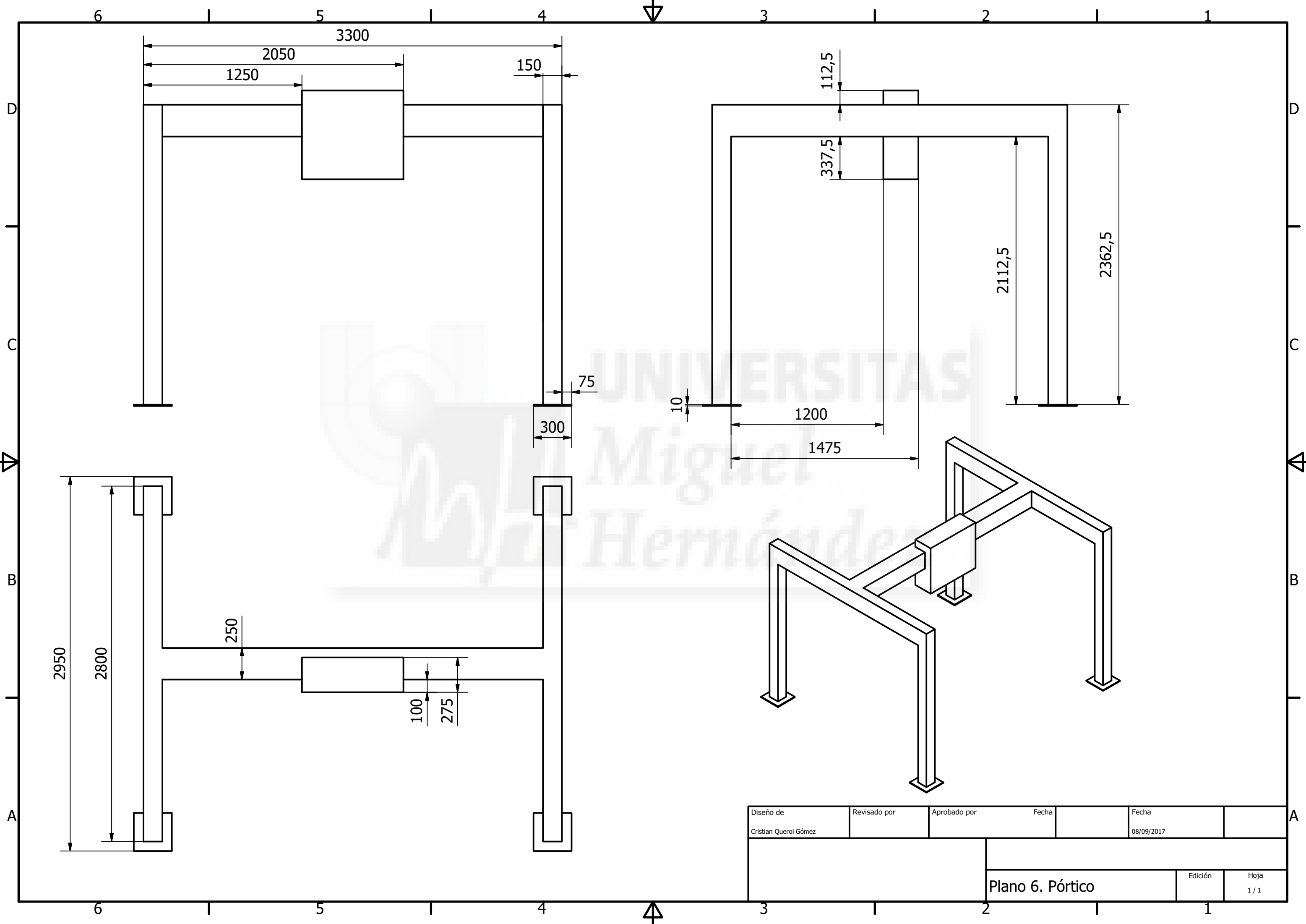




Diseño de Cristian Querol Gómez	Revisado por	Aprobado por	Fecha	Fecha 08/09/2017	
			Edición		Hoja 1 / 1
			Plano 5. Raíles		

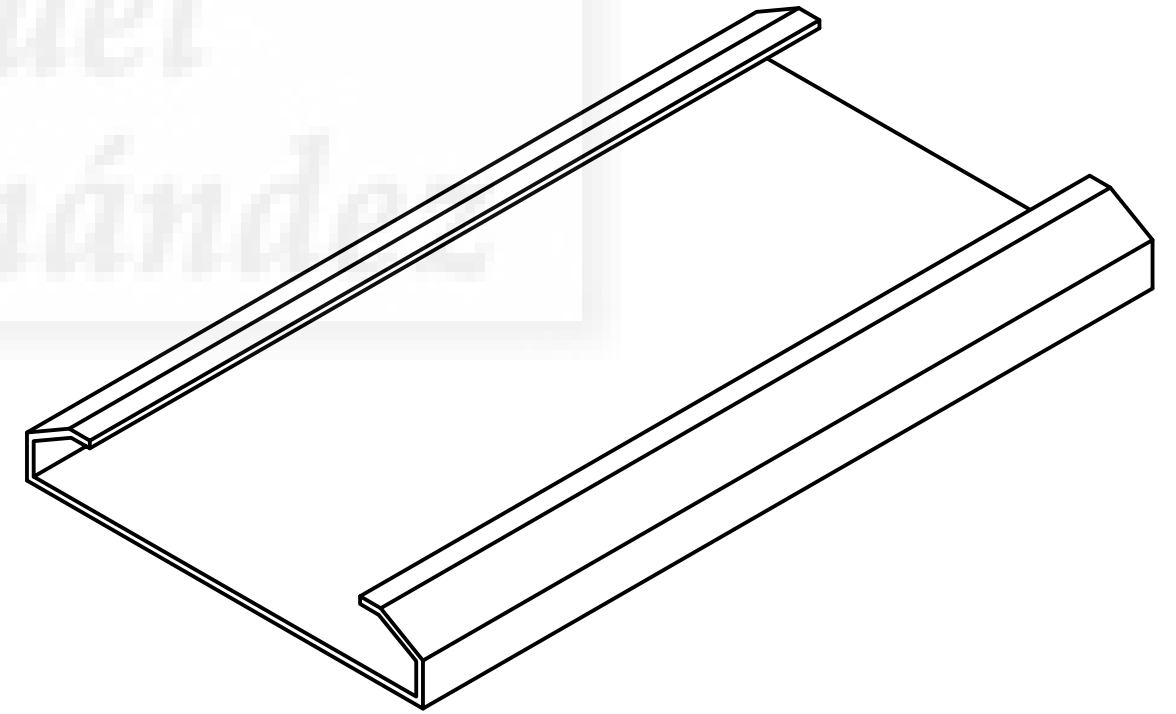
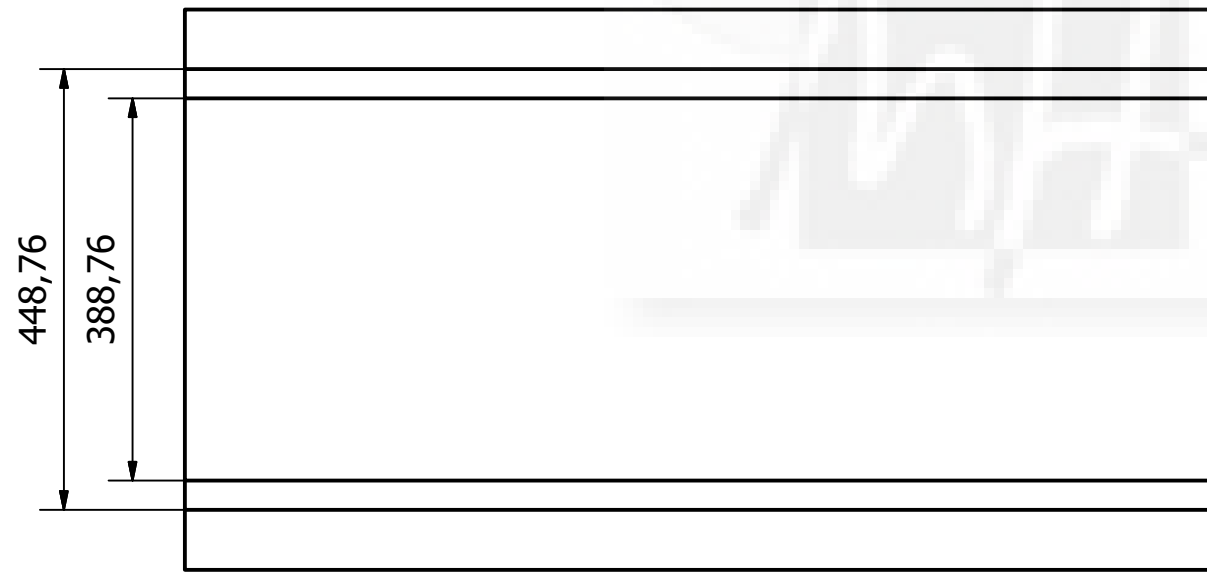
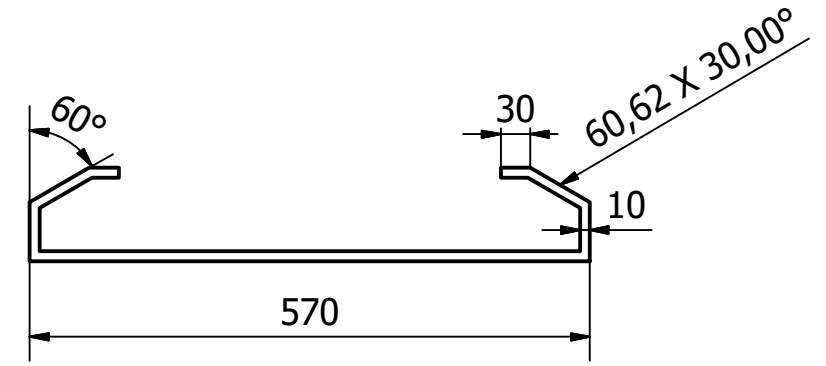
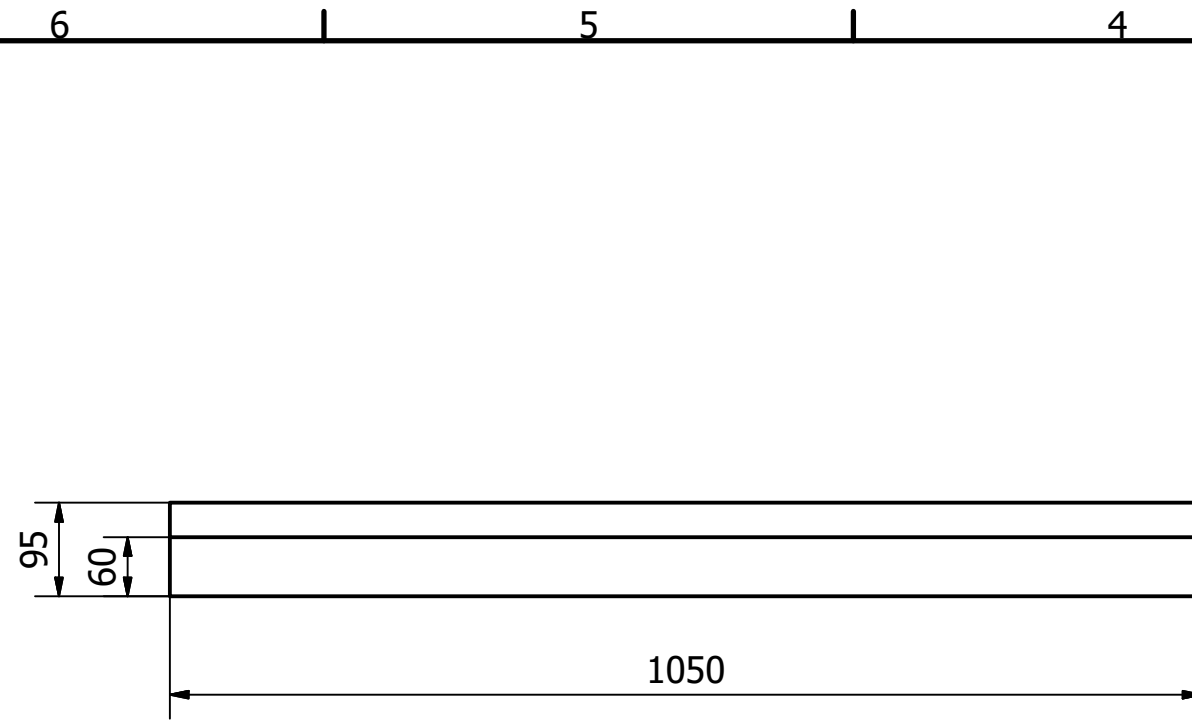
8.1.2 Planos de la Estación de Salpicaderos





UNIVERSITAS
Miguel
Hernández

Diseño de Cristian Querol Gómez	Revisado por	Aprobado por	Fecha	Fecha 08/09/2017	
			Edición		Hoja 1 / 1
			Plano 6. Pórtico		



UNIVERSITAS
Miguel
Hernández

Diseño de Cristian Querol Gómez	Revisado por Carlos Pérez	Aprobado por	Fecha	Fecha 08/09/2017	
			Edición		Hoja 1 / 1
			Plano7.Bandeja Salpicadero		

8.2 Señales

8.2.1 Señales de la Estación de Carrocerías

Módulo E/S. Controlador Robot IRB4600			
Entradas digitales	Mapeo	Salidas digitales	Mapeo
DetectaSalpicadero1	0	ActivaMotorCinta	0
DetectaSalpicadero2	1	-	-

Tabla 8.1: Señales E/S del controlador robot IRB4600

8.2.2 Señales de la Estación de Salpicaderos

Módulo E/S. Controlador Robot IRB1600			
Entradas digitales	Mapeo	Salidas digitales	Mapeo
DetectaSalpicadero	0	ActivaMotor	0
DetectaPuente	1	MoverPuenteDelante	1
-	-	MoverPuenteAtras	2

Tabla 8.2: Señales E/S del controlador robot IRB1600

8.3 Programas

8.3.1 Código de RAPID de la Estación de Ejemplo

```
MODULE Module1
  CONST robtarget Origen:=[[689.268,0,537.147],[0,0,1,0],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_10:=[[0,0,0],[0,0,1,0],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_20:=[[200,0,0],[0,0,1,0],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_30:=[[200,200,0],[0,0,1,0],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_40:=[[0,200,0],[0,0,1,0],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

  PROC Path_10()
    MoveL Origen,v100,fine,MyTool\WObj:=wobj0;
    MoveL Target_10,v100,fine,MyTool\WObj:=Workobjobject_1;
    MoveL Target_20,v100,fine,MyTool\WObj:=Workobjobject_1;
    MoveL Target_30,v100,fine,MyTool\WObj:=Workobjobject_1;
    MoveL Target_40,v100,fine,MyTool\WObj:=Workobjobject_1;
    MoveL Origen,v100,fine,MyTool\WObj:=wobj0;
  ENDPROC

  PROC main()
    Path_10; !!Realiza la trayectoria
  ENDPROC
ENDMODULE
```

8.3.2 Código de RAPID de la Estación de Carrocerías

```
MODULE Module1
  CONST robtarget Target_10:=[[366.639,744.873,97.081],[1,0,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_20:=[[366.639,-255.127,97.081],[1,0,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_30:=[[102.112879524,-
206.153749266,73.317520891],[0.904100067,0.421588785,0.063220835,0.02948036],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_40:=[[74.278259409,-54.43587901,-
53.988888092],[0.904100067,0.421588785,0.063220835,0.02948036],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_50:=[[64.961967124,-48.698779538,-
47.768558657],[0.904100067,0.421588785,0.063220835,0.02948036],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_60:=[[71.917101008,-39.214397213,-
40.363715635],[0.921629008,0.381751235,0.061874748,0.032209955],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_70:=[[82.309643506,-38.648680059,-
41.52212498],[0.921629008,0.381751235,0.061874748,0.032209955],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_80:=[[91.291135726,-41.270598839,-
45.466855237],[0.928151117,0.365608449,0.061303183,0.033284913],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_90:=[[96.388,-44.008,-
50.413],[0.928151117,0.365608449,0.061303183,0.033284913],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_100:=[[98.607135726,-51.705598839,-
56.842855237],[0.928151117,0.365608449,0.061303183,0.033284913],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_110:=[[89.260643505,-60.00268006,-
64.418124981],[0.921629008,0.381751236,0.061874748,0.032209955],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_120:=[[76.690932136,-64.649137034,-
68.811617202],[0.8858401,0.461019548,0.048993998,0.018651588],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_130:=[[64.40522184,-65.946887388,-
68.433502774],[0.87346846,0.484050159,0.04946545,0.017362684],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_140:=[[46.917,-63.158,-
63.11],[0.87346846,0.484050159,0.04946545,0.017362684],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget Target_150:=[[41.826535566,-54.808037549,-
50.123183178],[0.911359851,0.405641452,0.063679083,0.028691602],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```


CONST robtarger Target_160:=[[45.637372124,-46.532706812,-
41.339755212],[0.911359851,0.405641453,0.063679082,0.028691601],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_170:=[[53.375616971,-37.166758491,-
33.816151783],[0.927482284,0.36515312,0.072083311,0.035267675],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_180:=[[62.313740107,-28.636983391,-
28.554525478],[0.928076005,0.365446981,0.06398686,0.03207981],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_190:=[[76.516147899,-20.872196236,-
25.844042548],[0.942530976,0.322350246,0.078084776,0.040354009],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_200:=[[94.618244871,-18.777499558,-
28.725475698],[0.955352815,0.28073737,0.080419505,0.044946983],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_210:=[[106.388258941,-23.853383154,-
33.421852788],[0.962019485,0.255425278,0.083413755,0.048151676],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_220:=[[117.861448332,-27.901894643,-
37.989965172],[0.96633075,0.23859682,0.082560688,0.049600113],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_230:=[[124.893964148,-32.140064282,-
43.762769168],[0.955352815,0.28073737,0.080419506,0.044946982],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_240:=[[126.137103354,-40.01889686,-
51.050785403],[0.943176497,0.322690122,0.069856774,0.037539471],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_250:=[[126.470075971,-49.409108905,-
59.820536111],[0.912835717,0.404252359,0.052961927,0.022494837],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_260:=[[116.621,-58.978,-
69.54],[0.912835717,0.404252359,0.052961927,0.022494837],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_270:=[[106.405809667,-66.314378334,-
75.966043747],[0.89433366,0.443684935,0.05389273,0.02016326],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_280:=[[93.994857523,-70.507930772,-
79.539894122],[0.892154603,0.447469579,0.058042708,0.021498478],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_290:=[[79.051708325,-74.391994127,-
82.863479475],[0.89251538,0.447600701,0.052202368,0.018569357],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_300:=[[66.407213488,-73.746482396,-
80.561884704],[0.872141835,0.486105659,0.052962667,0.016274648],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_310:=[[53.682,-73.089,-
79.124],[0.872141835,0.486105659,0.052962667,0.016274648],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarger Target_320:=[[41.846978681,-71.492128609,-
73.625930066],[0.880605531,0.470847045,0.050749508,0.016169305],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_330:=[[34.636067847,-68.899381842,-
67.165399394],[0.880605531,0.470847045,0.050749508,0.016169305],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_340:=[[27.204867992,-62.889524984,-
57.116352681],[0.892153501,0.447472126,0.058027137,0.021533211],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_350:=[[23.702777993,-52.614885834,-
43.550560413],[0.906405093,0.415706958,0.069203854,0.028781221],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_360:=[[28.396765118,-43.832228531,-
33.940405794],[0.91352213,0.399824694,0.068691012,0.029984611],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_370:=[[34.89395739,-34.828756131,-
26.675312964],[0.91352213,0.399824694,0.068691012,0.029984611],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_380:=[[42.80873502,-26.38627682,-
22.672097461],[0.943630031,0.31794213,0.082906096,0.040021827],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_390:=[[55.199234096,-14.358889418,-
17.128029632],[0.956237455,0.276286083,0.085254632,0.044805996],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_400:=[[69.530343191,-2.863166949,-
14.197029315],[0.982596294,0.148116181,0.095840177,0.058144476],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_410:=[[83.864156038,0.999996386,-
15.743443763],[0.989738865,0.07858506,0.100228574,0.064773448],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_420:=[[100.243362292,-3.078366822,-
20.648356087],[0.985784225,0.121982343,0.098628193,0.060185123],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_430:=[[119.653261697,-5.668848664,-
25.816036721],[0.986408627,0.122373637,0.092174184,0.059385467],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_440:=[[134.839298135,-11.745105286,-
32.613537225],[0.973175887,0.208530554,0.084267959,0.048400544],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_450:=[[143.554415303,-19.246100452,-
39.063150183],[0.963153664,0.250781415,0.086298957,0.044678761],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_460:=[[150.51255249,-32.326737348,-
48.7260595],[0.952370577,0.293064832,0.07570602,0.037039002],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_470:=[[152.6667597,-42.777053565,-
58.567021605],[0.939611569,0.334740216,0.064956012,0.029322397],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_480:=[[148.233744075,-54.245000824,-
68.464086069],[0.924396041,0.375518784,0.062140378,0.024822879],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_490:=[[136.916831029,-64.510819954,-
79.476831585],[0.907885396,0.415736499,0.051284533,0.016648332],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_500:=[[122.070918867,-73.738718881,-90.671289726],[0.888887119,0.454942215,0.051961912,0.014395487],[-1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_510:=[[104.704474778,-80.597288516,-98.388344918],[0.884883204,0.462681797,0.052085556,0.013941491],[-1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_520:=[[86.427880984,-84.823603096,-102.800980737],[0.863859095,0.50083949,0.052644101,0.011656282],[-1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_530:=[[64.435,-85.419,-101.72],[0.863859095,0.50083949,0.052644101,0.011656282],[-1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_540:=[[42.193025328,-86.283653264,-96.705208685],[0.864080574,0.500885582,0.048874315,0.009470854],[-1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_550:=[[19.544543963,-82.456772548,-85.005015529],[0.863621169,0.500783862,0.056412884,0.013841489],[-1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_560:=[[5.807438204,-71.176440262,-62.809565168],[0.884122842,0.462476838,0.06347318,0.020325604],[-1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_570:=[[2.464129543,-62.524997625,-49.316061166],[0.884122842,0.462476838,0.06347318,0.020325604],[-1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_580:=[[6.629199975,-47.50332929,-31.346617713],[0.919853089,0.38315223,0.07752561,0.032472185],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_590:=[[18.420387521,-28.57937963,-17.452930092],[0.934615023,0.342165862,0.088276702,0.040305164],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_600:=[[25.095640399,-15.845503535,-10.889567704],[0.969697293,0.216315259,0.099199263,0.055266408],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_610:=[[39.694400764,4.840804963,-5.713711515],[0.988803213,0.087299259,0.099769511,0.068506127],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_620:=[[57.566827542,27.582549656,-5.969974528],[0.990176196,0.070028967,0.098558719,0.070236911],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_630:=[[74.044018232,44.249162888,-8.078235755],[0.990176196,0.070028967,0.098558719,0.070236911],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_640:=[[92.232036463,46.457325775,-11.306471509],[0.989527383,0.078667108,0.099167891,0.06937416],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_650:=[[111.851934041,40.709182158,-15.826113923],[0.989527383,0.078667108,0.099167891,0.06937416],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_660:=[[128.514817686,28.283656905,-21.262372348],[0.989527383,0.078667108,0.09916789,0.06937416],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

CONST robtarget Target_670:=[[160.001726382,16.446722383,-
29.76231759],[0.989950664,0.07896906,0.094849332,0.069030251],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_680:=[[171.166084478,-5.525768001,-
34.579395049],[0.989950664,0.07896906,0.094849332,0.069030251],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_690:=[[177.692762503,-19.515765075,-
42.501786938],[0.980532663,0.165726245,0.08767752,0.058336614],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_700:=[[182.735812031,-32.282323728,-
54.575704499],[0.963484847,0.251193865,0.079823463,0.047189043],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_710:=[[181.970651563,-44.600841631,-
65.533473928],[0.939242247,0.334859343,0.067250712,0.034213511],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_720:=[[168.72350222,-57.645522112,-
75.814415813],[0.907566432,0.415874725,0.054145735,0.020967211],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_730:=[[159.182345389,-69.690330157,-
89.088445561],[0.888562429,0.455066396,0.055008777,0.01858545],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
Target_1000:=[[52.129804365,53.349670392,43.572759696],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1010:=[[46.905172216,52.040823053,-
6.136295189],[0.99854398,-0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1020:=[[50.725,39.87,-6.235],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1030:=[[51.387,27.764,-5.974],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1040:=[[47.759,19.041,-5.343],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1050:=[[34.825,11.961,-3.852],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1060:=[[25.042,20.624,-2.941],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1070:=[[17.872,46.872,-2.89],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1080:=[[21.223,70.813,-3.883],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

```

CONST robtarget Target_1090:=[[34.626,84.549,-5.621],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1100:=[[50.129,85.87,-7.282],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1110:=[[62.981,81.239,-8.522],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1120:=[[68.447,62.325,-8.646],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1130:=[[74.179,44.988,-8.835],[0.99854398,-
0.013071656,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1140:=[[76.056,22.178,-8.476],[0.998049286,-
0.013077136,0.061043309,0.00057096],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1150:=[[76.232,2.476,-8.243],[0.998125401,-
0.004367126,0.061045967,0.000038242],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1160:=[[67.053,-11.023,-7.154],[0.998125401,-
0.004367126,0.061045967,0.000038242],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1170:=[[51.677,-24.664,-5.391],[0.998125401,-
0.004367126,0.061045967,0.000038242],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1180:=[[30.775,-28.094,-
3.094],[0.998629535,0.000000001,0.052335956,-0.000000001],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1190:=[[10.869,-23.437,-
0.974],[0.998629535,0.000000001,0.052335956,-0.000000001],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1200:=[[ -1.008,-8.92,0.182],[0.99859151,-
0.008714575,0.052333963,0.000456711],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1210:=[[ -9.152,16.435,0.846],[0.99859151,-
0.008714575,0.052333963,0.000456711],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1220:=[[ -10.818,42.765,0.311],[0.99854398,-
0.013071655,0.052331472,0.000685056],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1230:=[[ -6.388,73.289,-1.004],[0.998513086,-
0.015250107,0.052329853,0.000799224],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1240:=[[4.557,91.508,-2.625],[0.998477439,-
0.017428487,0.052327985,0.000913388],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1250:=[[23.08,111.986,-5.34],[0.998477439,-
0.017428487,0.052327985,0.000913388],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

```

CONST robtarget Target_1260:=[[39.257,121.052,-7.392],[0.998477439,-
0.017428487,0.052327985,0.000913388],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1270:=[[62.453,114.281,-9.569],[0.998477439,-
0.017428487,0.052327985,0.000913388],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1280:=[[84.925,91.847,-11.083],[0.998306136,-
0.013074519,0.05668793,0.000628014],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1290:=[[97.505,65.885,-12.16],[0.997773434,-
0.013079502,0.065397525,0.000513895],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1300:=[[103.876,46.267,-12.59],[0.997821006,-
0.008725783,0.065399145,0.00022854],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1310:=[[105.14,20.769,-12.178],[0.997821006,-
0.008725783,0.065399145,0.00022854],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1320:=[[103.123,-5.497,-11.672],[0.997849581,-
0.004371898,0.06539952,-0.000056819],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1330:=[[89.879,-38.642,-9.519],[0.997849581,-
0.004371898,0.06539952,-0.000056819],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1340:=[[73.545,-48.526,-7.609],[0.997849581,-
0.004371898,0.06539952,-0.000056819],[-1,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1350:=[[56.178,-58.511,-5.763],[0.998391818,-
0.000025896,0.056688637,-0.000417997],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1360:=[[24.704,-65.9,-2.39],[0.998629711,-
0.000022407,0.052331474,-0.000341913],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1370:=[[7.555,-54.497,-0.449],[0.998629711,-
0.000022407,0.052331474,-0.000341913],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1380:=[[-13.006,-38.196,1.721],[0.998629711,-
0.000022407,0.052331474,-0.000341913],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1390:=[[-27.629,-6.381,2.84],[0.999010148,-
0.00873565,0.043616228,0.000191011],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1400:=[[-40.124,36.67,3.125],[0.999010148,-
0.00873565,0.043616228,0.000191011],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1410:=[[-45.833,71.832,2.632],[0.999010148,-
0.00873565,0.043616228,0.000191011],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1420:=[[-39.709,102.398,1.081],[0.998810215,-
0.021811541,0.043609991,0.000761913],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

```

CONST robtarget Target_1430:=[[ -31.027,130.323,-0.776],[0.998810215,-
0.021811541,0.043609991,0.000761913],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1440:=[[ -6.661,154.768,-4.052],[0.998810215,-
0.021811541,0.043609991,0.000761913],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1450:=[[ 19.516,170.533,-7.387],[0.998391619,-
0.021817359,0.052324483,0.000571545],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1460:=[[ 42.433,171.355,-9.756],[0.998391619,-
0.021817359,0.052324483,0.000571545],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1470:=[[ 62.697,162.614,-11.468],[0.998391619,-
0.021817359,0.052324483,0.000571545],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1480:=[[ 89.678,146.041,-13.812],[0.998391619,-
0.021817359,0.052324483,0.000571545],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1490:=[[ 101.142,127.201,-14.695],[0.997411997,-
0.017456221,0.069746337,0.000190525],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1500:=[[ 111.039,108.282,-15.35],[0.997411997,-
0.017456221,0.069746337,0.000190525],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1510:=[[ 122.227,79.73,-16.017],[0.997411997,-
0.017456221,0.069746337,0.000190525],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1520:=[[ 130.333,52.456,-16.509],[0.997526351,-
0.008751605,0.069745344,-0.000418126],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1530:=[[ 135.679,25.27,-16.693],[0.997526351,-
0.008751605,0.069745344,-0.000418126],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1540:=[[ 134.051,-6.879,-16.164],[0.997555042,-
0.004399006,0.069742856,-0.000722442],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1550:=[[ 133.99,-31.864,-15.995],[0.997555042,-
0.004399006,0.069742856,-0.000722442],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1560:=[[ 122.736,-55.585,-14.266],[0.99756474,-
0.000046323,0.06973904,-0.001026745],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1570:=[[ 105.671,-70.009,-11.875],[0.99756474,-
0.000046323,0.06973904,-0.001026745],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1580:=[[ 79.511,-88.66,-
8.503],[0.997555446,0.004306361,0.069733896,-0.001331028],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1590:=[[ 58.189,-96.932,-
6.239],[0.998097656,0.008644363,0.061019813,-0.001710786],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

```

CONST robtarget Target_1600:=[[32.543,-101.243,-
3.51],[0.998592143,0.008629105,0.052307555,-0.001786156],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1610:=[[10.013,-89.777,-
0.989],[0.998592143,0.008629105,0.052307555,-0.001786156],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1620:=[[ -17.662,-68.737,2.169],[0.998629422,-
0.000085473,0.05232115,-0.001329624],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1630:=[[ -32.055,-37.772,3.465],[0.998848209,-
0.000091274,0.047963323,-0.001329238],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1640:=[[ -52.249,4.981,4.753],[0.999038096,-
0.004454401,0.043609969,-0.001100522],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1650:=[[ -68.705,68.118,4.571],[0.999141994,-
0.013175389,0.03925872,-0.000662434],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1660:=[[ -73.751,122.94,2.829],[0.998988974,-
0.021893935,0.039263006,-0.000319816],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1670:=[[ -62.629,158.421,0.434],[0.998988974,-
0.021893935,0.039263006,-0.000319816],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1680:=[[ -27.583,195.699,-3.952],[0.998988974,-
0.021893935,0.039263006,-0.000319816],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1690:=[[ 0.964,213.27,-7.873],[0.998503308,-
0.026245347,0.04798183,-0.000263493],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1700:=[[ 33.914,219.981,-11.66],[0.998160419,-
0.030602844,0.052337304,-0.000225699],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1710:=[[ 71.694,222.956,-15.64],[0.998160419,-
0.030602844,0.052337304,-0.000225699],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1720:=[[ 99.134,212.326,-18.323],[0.998046578,-
0.026247032,0.056689981,-0.000606589],[0,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Origen:=[[ 1070.504,0,1612],[0.707106781,0,-0.707106781,0],[0,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PROC Path_10()
MoveL Origen,v1000,z100,Pulidora\WObj:=wobj0;
MoveJ Target_10,v1000,z100,Pulidora\WObj:=Workobject_1;
MoveL Target_20,v1000,z100,Pulidora\WObj:=Workobject_1;
MoveJ Target_30,v1000,z100,Pulidora\WObj:=Workobject_1;
MoveJ Target_40,v1000,z100,Pulidora\WObj:=Workobject_1;
MoveL Target_50,v1000,z100,Pulidora\WObj:=Workobject_1;
MoveL Target_60,v1000,z100,Pulidora\WObj:=Workobject_1;

```



```

MoveL Target_1390,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1400,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1410,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1420,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1430,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1440,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1450,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1460,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1470,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1480,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1490,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1500,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1510,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1520,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1530,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1540,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1550,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1560,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1570,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1580,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1590,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1600,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1610,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1620,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1630,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1640,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1650,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1660,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1670,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1680,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1690,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1700,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1710,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1720,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveL Target_1010,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveJ Target_1000,v1000,z100,Pulidora\WObj:=Workobject_2;
MoveJ Origen2,v1000,fine,Pulidora\WObj:=Workobject_2;

```

ENDPROC

PROC main()

```

MoveJ Origen,v1000,fine,Pulidora\WObj:=wobj0; !!Robot en Origen
Reset MoverPuenteDelante; !!El movimiento del puente hacia delante se detiene
Reset MoverPuenteAtras; !!El movimiento del puente hacia atrás se detiene
Set ActivaMotor; !!Se activa el motor de las bancadas
WaitDI DetectaCarroceria1,1; !!El programa espera a que detecte la bancada
WHILE TRUE DO
IF DetectaCarroceria1 = 1 THEN
Set MoverPuenteDelante; !!El puente se mueve hacia delante
Path_10; !!Realiza la trayectoria
Path_20; !!Realiza la trayectoria
Reset MoverPuenteDelante; !!El movimiento del puente hacia delante se detiene

```

```

Set MoverPuenteAtras; !!El puente se mueve hacia atrás
WaitDI DetectaPuente,1; !!El programa espera a que detecte el puente
IF DetectaPuente = 1 THEN
    Reset MoverPuenteAtras; !!El movimiento del puente hacia atrás se detiene
ENDIF
ENDIF
ENDWHILE
ENDPROC
ENDMODULE

```

8.3.3 Código de RAPID de la Estación de Salpicaderos

```

MODULE Module1
    CONST robtarget Target_10:= [[-653.472,-
158.255,708.511],[0.866025385,0.500000033,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_20:= [[-
682.694121017,35.626223055,187.734379339],[0.968254684,0.208825189,-
0.136671108,0.01399696],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_30:= [[-
683.146356404,24.461211069,188.105850075],[0.968254684,0.208825189,-
0.136671108,0.01399696],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_40:= [[-
684.11622988,12.479763293,190.228173007],[0.968254684,0.208825189,-
0.136671108,0.01399696],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_50:= [[-685.797157791,-
11.074914364,193.597392771],[0.956442281,0.244693141,-0.146743521,0.061723321],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_60:= [[-665.630622976,-
75.035999254,201.894017062],[0.957739376,0.250753606,-0.139613807,0.019128576],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_70:= [[-665.853216666,-
86.199405941,201.320603429],[0.957739376,0.250753606,-0.139613807,0.019128576],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_80:= [[-666.075810356,-
97.362812627,200.747189798],[0.945890101,0.292290949,-0.138646549,0.025200239],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_90:= [[-666.298404046,-
108.526219314,200.173776165],[0.945890101,0.292290949,-0.138646549,0.025200239],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_100:= [[-666.248954251,-
119.60452871,198.667583736],[0.945890101,0.29229095,-0.138646549,0.025200239],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_110:= [[-666.199504455,-
130.682838107,197.161391308],[0.945890101,0.292290949,-0.138646549,0.025200239],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_120:= [[-666.620846988,-
143.021068041,197.135386307],[0.932240271,0.3332719,-0.137415369,0.031223931],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

```

CONST robtarget Target_130:=[[-667.073265384,-
144.912484258,199.419650365],[0.937346959,0.33159273,-0.096620831,0.045731329],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_140:=[[-666.084664544,-
146.037991066,201.086422101],[0.941939375,0.322786772,0.026554103,0.088621628],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_150:=[[-664.906669712,-
152.331078554,201.183262354],[0.941939375,0.322786772,0.026554103,0.088621628],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_160:=[[-647.692,-
172.896787609,196.811044444],[0.939692621,0.342020143,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_170:=[[-647.692,-
178.655372654,195.895239729],[0.939692621,0.342020143,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_180:=[[-647.692,-
184.542596042,194.861559691],[0.923879533,0.382683432,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_190:=[[-647.692,-
194.645552572,191.615736698],[0.917060075,0.398749068,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_200:=[[-647.692,-
198.413104659,189.937715124],[0.887010833,0.461748613,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_210:=[[-647.692,-
202.91929093,186.989107776],[0.866025404,0.5,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_220:=[[-647.692,-
206.432013673,183.768840341],[0.843391446,0.537299608,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_230:=[[-647.692,-
216.245572911,173.484577668],[0.819152044,0.573576436,0,0],[-1,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_240:=[[-647.692,-
217.955673629,168.786114564],[0.79335334,0.608761429,0,0],[-1,-
2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_250:=[[-647.692,-
217.955673629,168.786114564],[0.766044443,0.64278761,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_260:=[[-647.692,-
217.955673629,168.786114564],[0.737277337,0.675590208,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_270:=[[-647.692,-
217.955673629,168.786114564],[0.707106781,0.707106781,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_280:=[[-641.797,-
214.50542326,156.617055469],[0.64278761,0.766044443,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_290:=[[-641.797,-
215.569034028,151.175378613],[0.608761429,0.79335334,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```



```

CONST robtarget Target_300:=[[-641.797015235,-
219.346915511,144.375076933],[0.573576437,0.819152044,-0.000000012,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_310:=[[-641.797,-
225.495317415,135.505665257],[0.537299609,0.843391446,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_320:=[[-641.797,-
225.800673742,124.329496078],[0.5,0.866025404,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_330:=[[-641.797,-
228.348576822,114.197597133],[0.422618262,0.906307787,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_340:=[[-641.797,-
380.521,114.198],[0.422618262,0.906307787,0,0],[-1,-
3,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_350:=[[-474.764,-
380.521,114.198],[0.422618262,0.906307787,0,0],[-1,-
2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_360:=[[-474.764,-
230.521,178.511],[0.422618262,0.906307787,0,0],[-1,-
3,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_370:=[[-474.764,-
230.521,178.511],[0.707106782,0.707106781,0,0],[-1,-
3,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_380:=[[-474.764,-
230.521,178.511],[0.696364241,0.69636424,0.122787804,0.122787804],[-1,-1,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_390:=[[-474.764,-
155.521,178.511],[0.69636424,0.69636424,0.122787804,0.122787804],[-1,-1,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_400:=[[-454.681393664,-
250.169570643,204.466923189],[0.599512975,0.781300521,0.137764362,0.105710313],[-1,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_410:=[[-463.943000001,-
175.927508076,158.519923789],[0.836516304,0.482962913,0.224143868,0.129409523],[-1,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_420:=[[-458.777153691,-
142.182096974,168.475759394],[0.875426098,0.408217894,0.25783416,0.022557566],[-1,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_430:=[[-483.84445543,-
118.465282873,185.400075472],[0.776176165,0.422994833,0.21294187,0.416271176],[-1,-2,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_440:=[[-516.254137817,-
71.375560219,206.37278826],[0.892399101,0.36964381,0.205334954,0.157559052],[-1,-3,-3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_450:=[[-439.927415094,-
164.906280496,308.776351678],[0.766320481,0.588018386,0.239117619,0.099045761],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_460:=[[-468.905189964,-
113.675047012,297.12949462],[0.588018387,0.76632048,0.256604813,0.033782665],[-1,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

CONST robtarget Target_470:=[[-469.644,-
14.762608073,272.211241841],[0.414693243,0.909961271,0,0],[-
1,1,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_480:=[[-389.644,-
14.762608072,272.211241842],[0.414693243,0.909961271,0,0],[-
1,1,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_490:=[[-339.644,-
22.369025641,305.349473716],[0.414693243,0.909961271,0,0],[-
1,1,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_500:=[[-278.081,-
25.325190527,315.191999997],[0.5,0.866025404,0,0],[-
1,2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_510:=[[-234.442294784,-
28.909581815,291.925155062],[0.5,0.866025404,0,0],[-
1,2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_520:=[[-179.50192069,-
146.729594099,291.925154955],[0.674379723,0.701057385,0.092295956,0.21263111],[-
1,0,3,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_530:=[[-194.588608062,-
149.32989362,289.320432273],[0.675288742,0.732202415,-0.085391268,0.023938877],[-
1,-2,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_540:=[[-244.87756597,-
157.997558687,280.638023346],[0.646082999,0.685152336,-0.271989561,0.197900767],[-
1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_550:=[[-283.211983871,-
140.039942262,283.465816926],[0.801398728,0.494587751,-0.313942229,0.120761383],[-
1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_560:=[[-244.87756597,-
157.997558687,280.638023346],[0.841455327,0.422859194,-0.323272632,0.092939981],[-
1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_570:=[[-193.172053589,-
76.649449977,223.547240824],[0.722759721,0.090861843,-0.217900019,0.649524512],[-
1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_580:=[[-205.675285765,-
103.765795352,212.351122038],[0.715864645,0.134815793,-0.177841069,0.661615497],[-
1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_590:=[[-198.665543915,-
133.894977587,204.650963771],[0.715864646,0.134815793,-0.177841069,0.661615496],[-
1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_600:=[[-217.8073827,-
127.847946752,194.800916763],[0.715864646,0.134815793,-0.177841069,0.661615496],[-
1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_610:=[[-238.444619023,-
127.519040365,206.055459338],[0.692514235,0.298836238,-0.236898577,0.612372436],[-
1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_620:=[[-214.322343257,-
188.897914361,163.317205732],[0.640856382,0.400633602,-0.35355339,0.551176755],[-
1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_630:=[[-215.47374679,-
279.221716521,206.215749219],[0.640856382,0.400633602,-0.35355339,0.551176755],[-
1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

CONST robtarget Target_640:=[[-151.720041608,-
239.753208461,161.136852993],[0.673335249,0.343254756,-0.400246232,0.518265153],[-
1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_650:=[[-17.525543162,-
293.500477245,181.464473039],[0.640856382,0.400633602,-0.35355339,0.551176755],[-
1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_660:=[[-17.525543162,-
293.500477245,181.464473039],[0.515328124,0.55284818,-0.198851309,0.623902213],[-
1,-2,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_670:=[[-17.525543162,-
293.500477245,181.464473039],[0.354681106,0.667387068,-0.03059784,0.654109767],[0,-
1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_680:=[[-17.525543162,-
293.500477245,181.464473039],[0.169863157,0.73644463,0.139740821,0.639740821],[0,-
1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_690:=[[-130.119493649,-
234.1515127,194.896240474],[0.524519053,0.59150635,0.433012703,0.433012702],[0,-1,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_700:=[[-130.119493649,-
234.1515127,194.896240474],[0.484783556,0.551515946,0.477079807,0.482918133],[0,-1,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_710:=[[-126.201000179,-
184.42872363,198.401901035],[0.46351222,0.529926432,0.497771694,0.50651529],[0,-
2,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_720:=[[-157.216,-114.081,251.548],[0.5,0.5,0.5,0.5],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_730:=[[-27.216,-114.081,251.548],[0.5,0.5,0.5,0.5],[0,-
2,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_740:=[[-27.216,-
114.081,251.548],[0.640856382,0.640856382,0.298836239,0.298836239],[0,-
2,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_750:=[[-27.216,-
114.081,251.548],[0.704416026,0.704416026,0.061628417,0.061628417],[0,-
2,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_760:=[[-88.352170393,-
35.730273525,307.697065329],[0.633022222,0.754406507,-0.111618897,-0.133022221],[-
1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_770:=[[-88.010150231,-
41.698511219,306.644704001],[0.64278761,0.766044443,0.000000001,0.000000001],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_780:=[[-28.010150234,-
40.713703296,306.818352192],[0.64278761,0.766044443,0.000000001,0.000000001],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_790:=[[1.989849767,-
40.789230918,307.159035671],[0.608761429,0.79335334,0.000000001,0.000000001],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_800:=[[95.199678536,-
38.648891074,310.710423874],[0.64268971,0.765927771,0.011218192,0.01336932],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

```

CONST robtarget Target_810:=[[95.199678536,-
38.648891074,310.710423874],[0.635727142,0.757630106,0.095010059,0.11322858],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_820:=[[123.888821177,-
30.01099331,312.233518343],[0.63097779,0.751970049,0.122649659,0.146168173],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_830:=[[100.434603544,-
186.280066261,284.679064456],[0.64268971,0.765927771,0.011218192,0.01336932],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_840:=[[100.434603544,-
186.280066261,284.679064456],[0.562194711,0.669997567,-0.311629616,-
0.371385715],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_850:=[[100.434603544,-
186.280066261,284.679064456],[0.462382711,0.551046257,-0.446517793,-0.532139184],[-
1,1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_860:=[[177.63977166,-
111.870144822,242.132037228],[0.673533315,0.427423775,-0.56063042,-
0.222160577],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_870:=[[156.107922139,-
252.703060433,187.710999997],[0.69636424,0.353553391,-0.612372436,-
0.122787804],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_880:=[[116.460237628,-
360.885199136,84.28622214],[0.69636424,0.353553391,-0.612372436,-0.122787804],[-
1,1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_890:=[[12.324806053,-
376.80816069,101.616755804],[0.581129853,0.521738679,-0.623286173,0.039889738],[-
1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_900:=[[4.34347273,-
442.593759685,91.197865108],[0.426292427,0.654368338,-0.591723988,0.19984886],[-
1,1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_910:=[[117.225748181,-
356.119728554,85.591704322],[0.522567879,0.609723622,-0.508709468,0.310442577],[-
1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_920:=[[114.929216523,-
370.4161403,81.675257777],[0.522567879,0.609723622,-0.508709468,0.310442577],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_930:=[[116.460237628,-
360.885199136,84.28622214],[0.346953725,0.724198313,-0.411027162,0.431528201],[0,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_940:=[[174.618149346,-
481.510529788,92.482984284],[0.346953725,0.724198313,-0.411027162,0.431528201],[-
1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_950:=[[236.247195579,-
360.333259714,101.274519091],[0.147695248,0.789320086,-
0.285334034,0.523205892],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_960:=[[199.163751775,-
333.746532356,121.718689991],[0.278078304,0.836274539,-
0.071320442,0.467151739],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_970:=[[199.163751775,-
333.746532356,121.718689991],[0.099505298,0.737763972,0.175784404,0.644129537],[0,-
1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

```

CONST robtarget Target_980:=[[180.822923637,-
308.898428698,156.177331724],[0.203969095,0.751033404,0.426791242,0.460645927],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_990:=[[185.049106264,-
292.318748212,166.533661932],[0.203969095,0.751033404,0.426791242,0.460645927],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1000:=[[316.161388317,-
315.049359035,149.419666888],[0.203969095,0.751033404,0.426791242,0.460645927],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1010:=[[316.161388317,-
315.049359035,149.419666888],[0.390038039,0.88073697,0.267627511,0.023414373],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1020:=[[284.314103137,-
210.505109463,185.57668866],[0.499524111,0.86520114,0.021809693,0.037775497],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1030:=[[458.098410279,-
191.120291915,172.685100089],[0.537881995,0.840764472,0.058776065,0.018532022],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1040:=[[549.774541822,-
202.98815725,159.282090059],[0.465759975,0.883134531,0.006850955,-0.055624719],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1050:=[[617.79207313,-
203.328105798,103.828745466],[0.427205576,0.899446939,-0.023758319,-
0.089028878],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1060:=[[641.432067897,-
207.682725318,137.83499229],[0.466032295,0.87995642,-0.019852321,-
0.089980466],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1070:=[[647.099886,-
204.686717299,175.12833555],[0.540952157,0.835990529,-0.011934462,-
0.091368307],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1080:=[[645.443105754,-
213.709526269,171.147883626],[0.540952157,0.835990529,-0.011934462,-
0.091368307],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1090:=[[653.297128756,-
194.941317983,210.409342511],[0.611755046,0.785662246,-0.003925775,-
0.09206078],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1100:=[[651.802522919,-
204.546746653,208.063730981],[0.67790211,0.729354598,0.004112789,-0.092052614],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1110:=[[651.040499438,-
214.122695417,210.84231434],[0.794254351,0.600557597,0.020035076,-0.08993995],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1120:=[[660.579598093,-
167.734198563,238.484568236],[0.865847891,0.491748748,0.031603191,-
0.086555399],[0,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1130:=[[662.099593374,-
124.882694259,212.766127513],[0.93808499,0.333925071,0.046153254,-0.079752592],[0,-
2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1140:=[[661.766307675,-
118.544075492,205.038849031],[0.93808499,0.333925071,0.046153254,-0.079752592],[0,-
2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

CONST robtarget Target_1150:=[[674.707915112,-
83.418584131,201.409485989],[0.554301076,0.139310097,-0.21547005,-0.791780065],[0,-
2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_1160:=[[685.96352065,-
55.820071475,199.134677457],[0.544374833,0.104640543,-0.239443244,-
0.797103087],[0,-2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_1170:=[[696.685960583,-
28.190722744,197.757705666],[0.533412341,0.0697718,-0.262960644,-0.800908778],[0,-
2,3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1180:=[[696.178097305,64.714187152,181.706837262],[0.484334056,0.029442893,-
-0.338279238,-0.806300686],[0,-2,3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1190:=[[657.099291667,188.522270809,141.866703714],[0.469117544,-
0.005755472,-0.359083626,-0.806817547],[0,-
2,3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1200:=[[625.616914844,215.583305509,136.873178462],[0.469117544,-
0.005755472,-0.359083626,-0.806817547],[0,-
2,3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1210:=[[653.363015125,215.766392929,134.694474326],[0.999657325,0,0.0261769
48,0],[0,-2,3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1220:=[[653.316671911,216.462,130.72125907],[0.999657325,0,0.026176948,0],[0,-
2,3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1230:=[[646.829679634,276.44933238,95.514386821],[0.999505072,-
0.01734022,0.026172961,0.001975668],[0,-
2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1240:=[[497.913792188,246.46323064,118.383676582],[0.999050214,-
0.034781348,0.026134495,0.002432148],[0,-
2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1250:=[[415.066068039,225.969477177,136.031186988],[0.998863013,-
0.016459984,-0.018185032,-0.040878542],[0,-2,-
3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1260:=[[266.104136069,211.337954349,156.108373976],[0.998863013,-
0.016459984,-0.018185032,-0.040878542],[0,-
2,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1270:=[[265.754306155,211.681647285,166.096341447],[0.99612922,-
0.015651098,-0.018885698,-0.084409427],[0,-2,-
3,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_1280:=[[64.39,269.13155742,134.255946982],[0.999048222,0,0,-
0.043619387],[0,-2,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget
Target_1290:=[[4.61831811,274.360901939,135.255946982],[0.996194698,0,0,-
0.087155742],[0,-2,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

CONST robtarget Target_1300:=[[-
94.938000002,312.27277871,116.057973492],[0.998097349,-0.043577871,-0.001902651,-
0.043577871],[0,-2,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1310:=[[-
154.709681891,317.482223996,115.602206081],[0.998097349,-0.043577871,-
0.001902651,-0.043577871],[0,-2,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1320:=[[-283.963,345.57227686,79.163760556],[0.976296007,-
0.216439613,0,0],[-1,-3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1330:=[[-343.963,345.57227686,79.163760556],[0.976296007,-
0.216439613,0,0],[-1,-3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1340:=[[-373.963,326.31777034,108.036612464],[0.976296007,-
0.216439613,0,0],[-1,-3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1350:=[[-474.764,326.318,108.037],[0.976296007,-
0.216439614,0,0],[-1,-3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1360:=[[-474.764,319.479,108.037],[0.976296007,-
0.216439614,0,0],[-1,-3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1370:=[[-
474.764,298.629040319,134.784164662],[0.976296007,-0.216439614,0,0],[-1,-3,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1380:=[[-574.764,298.629,134.784],[0.976296007,-
0.216439614,0,0],[-1,-3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1390:=[[-
574.764,268.904056822,142.024701135],[0.976296007,-0.216439614,0,0],[-1,-3,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1400:=[[-624.764,268.904,142.025],[0.976296007,-
0.216439614,0,0],[-1,-3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1410:=[[-
635.626544445,248.89278618,144.46123138],[0.976296007,-0.216439614,0,0],[-1,-3,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1420:=[[-635.627,248.893,198.511],[0.976296007,-
0.216439614,0,0],[-1,1,-2,2],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1430:=[[-
642.366959661,224.086,134.022197148],[0.992403877,0.087155742,-0.086824088,0],[-
1,1,-2,2],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1440:=[[-
643.42935076,165.469899138,155.16538307],[0.992403877,0.087155742,-
0.086824088,0],[-1,1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1450:=[[-
664.896726805,101.513675825,173.899515271],[0.982962913,0.130401958,-0.129409522,-
0.005693472],[-1,1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Target_1460:=[[-
664.385638088,75.328000006,182.839025406],[0.982962913,0.129409521,-0.129409522,-
0.017037087],[-1,1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Origen:=[997.682928758,0,1043.499980715],[0.683012709,-
0.183012676,-
0.183012676,0.683012709],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PROC Path_10()
    MoveJ Target_10,v400,z1,Pulidora\WObj:=Workobject_1;
    MoveJ Target_20,v400,z1,Pulidora\WObj:=Workobject_1;
    MoveJ Target_30,v400,z1,Pulidora\WObj:=Workobject_1;

```



```
MoveJ Target_1050,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1060,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1070,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1080,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1090,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1100,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1110,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1120,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1130,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1140,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1150,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1160,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1170,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1180,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1190,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1200,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1210,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1220,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1230,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1240,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1250,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1260,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1270,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1280,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1290,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1300,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1310,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1320,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1330,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1340,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1350,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1360,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1370,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1380,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1390,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1400,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1410,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1420,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1430,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1440,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1450,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_1460,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Target_10,v400,z1,Pulidora\WObj:=Workobject_1;
MoveJ Origen,v400,fine,Pulidora\WObj:=wobj0;
```

ENDPROC

PROC main()

WHILE TRUE DO

MoveJ Origen,v1000,fine,Pulidora\WObj:=wobj0; !!Robot en Origen

Set ActivaMotorCinta; !!Se activa el motor de la cinta transportadora

```
IF DetectaSalpicadero = 1 THEN !!Si detecta un salpicadero entonces
  Reset ActivaMotorCinta; !!El motor de la cinta transportadora se para
  Path_10; !!Realiza la trayectoria
  Set ActivaMotorCinta; !!Al finalizar la trayectoria se vuelve a poner en marcha la cinta
  transportadora
  WaitDI DetectaSalpicadero,0; !!El programa espera hasta que no se detecte el
  salpicadero
ENDIF
ENDWHILE
ENDPROC

ENDMODULE
```

8.4. Hojas de Características de los Robots

8.4.1 Hoja de Característica del Robot IRB140



IRB 140 Industrial Robot

Main Applications

Arc welding
Assembly
Cleaning/Spraying
Machine tending
Material handling
Packing
Deburring



Small, Powerful and Fast

Compact, powerful IRB 140 industrial robot.

Six axis multipurpose robot that handles payload of 6 kg, with long reach (810 mm). The IRB 140 can be floor mounted, inverted or on the wall in any angle. Available as Standard, Foundry Plus 2, Clean Room and Wash versions, all mechanical arms completely IP67 protected, making IRB 140 easy to integrate in and suitable for a variety of applications. Uniquely extended radius of working area due to bend-back mechanism of upper arm, axis 1 rotation of 360 degrees even as wall mounted.

The compact, robust design with integrated cabling adds to overall flexibility. The Collision Detection option with full path retraction makes robot reliable and safe.

Using IRB 140T, cycle-times are considerably reduced where axis 1 and 2 predominantly are used.

Reductions between 15-20 % are possible using pure axis 1 and 2 movements. This faster versions is well suited for packing applications and guided operations together with PickMaster.

IRB 140 Foundry Plus 2 and Wash versions are suitable for operating in extreme foundry environments and other harsh environments with high requirements on corrosion resistance and tightness. In addition to the IP67 protection, excellent surface treatment makes the robot high pressure steam washable. Also available in white Clean Room ISO class 6 version, making it especially suited for environments with stringent cleanliness standards.

IRB 140

Specification

Robot versions	Handling capacity	Reach of 5th axis	Remarks
IRB 140/IRB 140T	6 kg	810 mm	
IRB 140F/IRB 140TF	6 kg	810 mm	Foundry Plus 2 Protection
IRB 140CR/IRB 140TCR	6 kg	810 mm	Clean Room
IRB 140W/IRB 140TW	6 kg	810 mm	SteamWash Protection
Supplementary load (on upper arm alt. wrist)			
on upper arm		1 kg	
on wrist		0.5 kg	
Number of axes			
Robot manipulator		6	
External devices		6	
Integrated signal supply	12 signals on upper arm		
Integrated air supply	Max. 8 bar on upper arm		
IRC5 Controller variants:	Single cabinet, Dual cabinet, Compact, Panel mounted		

Performance

Position repeatability 0.03 mm (average result from ISO test)

Axis movement	Axis	Working range
	1	360°
	2	200°
	3	280°
	4	Unlimited (400° default)
	5	230°
	6	Unlimited (800° default)

Max. TCP velocity 2.5 m/s

Max. TCP acceleration 20 m/s²

Acceleration time 0-1 m/s 0.15 sec

Velocity *)

Axis no.	IRB 140	IRB 140T
1	200°/s	250°/s
2	200°/s	250°/s
3	260°/s	260°/s
4	360°/s	360°/s
5	360°/s	360°/s
6	450°/s	450°/s

*) Max velocity is reduced at single phase power supply, e.g. Compact controller. Please, see the Product specification for further details.

Cycle time

5 kg Picking side	IRB 140	IRB 140T
cycle 25 x 300 x 25 mm	0.85s	0.77s

Electrical Connections

Supply voltage	200–600 V, 50/60 Hz
Rated power	
Transformer rating	4.5 kVA
Power consumption typically	0.4 kW

Physical

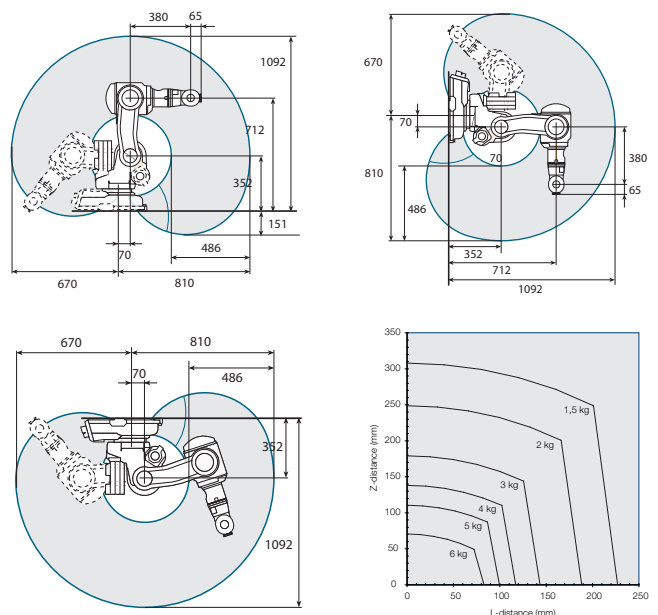
Robot mounting	Any angle
Dimensions	
Robot base	400 x 450 mm
Robot controller H x W x D	950 x 800 x 620 mm
Weight	
Robot manipulator	98 kg

Environment

Ambient temperature for	
Robot manipulator	5 – 45°C
Relative humidity	Max. 95%
Degree of protection,	
Manipulator	IP67
Options	
	Foundry Plus 2
	SteamWash
	(High pressure steam washable)
	Clean Room, class 6
	(certified by IPA)
Noise level	Max. 70 dB (A)
Safety	
	Double circuits with supervision, emergency stops and safety functions,
	3-position enable device
Emission	EMC/EMI-shielded

Data and dimensions may be changed without notice

Working range



8.4.2 Hoja de Característica del Robot IRB4600



IRB 4600 Industrial Robot

IRB 4600 is a highly productive general purpose robot optimized for short cycle times where compact robots can help create high density cells.



The IRB 4600 enables more compact manufacturing cells with increased production output and higher quality - and that means improved productivity.

Shortest cycle times

Thanks to the new compact and optimized design resulting in a low weight, the IRB 4600 can cut the cycle times of the industry benchmark by up to 25%. The maximum acceleration achievable is highest in its class, together with high maximum speeds. The high acceleration is possible to use to avoid obstacles or to follow the path. The benefit is increased production capacity and higher productivity.

Ultra-wide working range

You can position the IRB 4600 in the most favourable way with regard to reach, cycle time and auxiliary equipment. Flexible mounting with floor, tilted, semi-shelf or inverted mounting is very useful when you are simulating the best position for your application.

Compactness

The small footprint, the slim swing base radius around axis 1, the fine elbow behind axis 3, the small lower and upper arms, and the compact wrist all contribute to the most compact robot in its class. With the IRB 4600 you can create your production cell with reduced floorspace by placing the robot closer to the served machines, which also increases your output per m² and your productivity.

Best protection available

ABB has the most comprehensive protection program on the market and it will be even further enhanced with the IRB 4600. Foundry Plus includes IP 67, resistant paint, rust-protected mounting flange and protection for molten metal spits on non-moving cables on the rear of the robot and extra protection plates over the floor cable connections on the foot.

Optimize and go sharp

To get the IRB 4600 ready for the targeted applications you have access to high performing workpiece positioners, Track motions, and the motor and gear unit range.

To simulate your production cell to find the optimal position for the robot and program it offline, RobotStudio is available on subscription together with PowerPacs for several applications.

If you would like to learn more about how to use the IRB 4600 in your applications and environments, you can watch simulations on several applications at www.abb.com/robotics

Specification

Variants:	Reach	Payload	Armload
IRB 4600-60/2.05	2.05 m	60 kg	20 kg
IRB 4600-45/2.05	2.05 m	45 kg	20 kg
IRB 4600-40/2.55	2.55 m	40 kg	20 kg
IRB 4600-20/2.50	2.51 m	20 kg	11 kg
Number of axes	6+3 external (up to 36 with MultiMove)		
Protection	Standard IP67, as option Foundry Plus 2		
Mounting	Floor, shelf, inverted or tilted		
IRC5 Controller variants	Single cabinet, Dual cabinet		

Physical

Dimensions robot base	512 x 676 mm	
Robot height IRB 4600-60/2.05 and IRB 4600-45/2.05	1727 mm	
Robot height IRB 4600-40/2.55 and IRB 4600-20/2.50	1922 mm	
Robot weight	412 to 435 kg	

Performance

Position repeatability (RP)	0.05 - 0.06 mm	
Path repeatability (RT)	0.13 - 0.46 mm	
	(measured at speed 250 mm/s)	

Movement

Axis movements	Working range	Maximum speed
Axis 1	+180° to -180°	175°/s
Axis 2	+150° to -90°	175°/s
Axis 3	+75° to -180°	175°/s
Axis 4	+400° to -400°	250° (20/2.50 has 360°/s)
Axis 5*	+120° to -125°	250° (20/2.50 has 360°/s)
Axis 6	+400° to -400°	360° (20/2.50 has 500°/s)

* Axis 5 for IRB 4600-20/2.50 +120°-120°

Electrical connections

Supply voltage	200-600 V, 50-60 Hz
----------------	---------------------

Environment

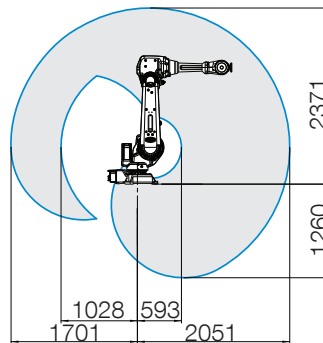
Ambient temperature for mechanical unit:

During operation	+5° C (41° F) to + 45° C (113° F)
During transportation and storage	-25° C (-13° F) to +55° C (131° F)
For 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

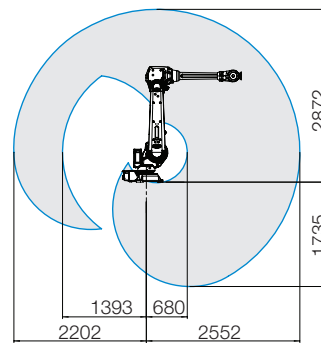
Data and dimensions may be changed without notice.

Main applications

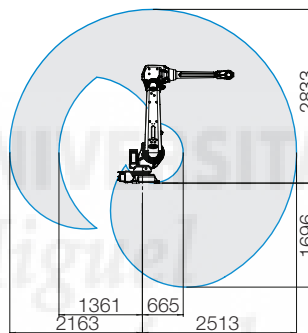
- Machine tending
- Material handling
- Arc welding
- Cutting
- Dispensing
- Assembly
- Palletizing and packing



IRB 4600-60/2.05
IRB 4600-45/2.05



IRB 4600-40/2.55



IRB 4600-20/2.50

For more information please contact:

ABB AB

Robotics

Hydrovägen 10
SE-721 36 Västerås, Sweden
Phone: +46 21 325000

www.abb.com/robotics

Note

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© 2016 ABB All rights reserved

8.4.3 Hoja de Característica del Robot IRB1600



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.

IRB 1600

Main applications

Machine tending, material handling, arc welding, cutting, dispensing, assembly, palletizing and packaging, measuring, die casting, injection moulding

Specification

Variants	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; opt. FoundryPlus 2 (IP 67)		
Mounting	Floor, wall, shelf, tilted, inverted		
IRC5 Controller variants	Single cabinet, Dual cabinet, Compact		

Physical

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

Performance (according to ISO 9283)

	6/1.2	6/1.45	10/1.2	10/1.45
Pos. repeatability (RP)	0.02 mm	0.02 mm	0.02 mm	0.05 mm
Path repeatability (RT)	0.13 mm	0.19 mm	0.06 mm	0.13 mm

Movement

Working range	1.2	1.45
Axis 1	+180° to -180°	+180° to -180°
Axis 2	+136° to -63°	+150° to -90°
Axis 3	+55° to -235°	+65° to -245°
Axis 4	+200° to -200° def. +/-190° revolution	+200° to -200° def. +/-190° revolution
Axis 5	+115° to -115°	+115° to -115°
Axis 6	+400° to -400° def. +/-288 revolution	+400° to -400° def. +/-288 revolution
Maximum Speed	6 kg	10 kg
Axis 1	150°/s	180°/s
Axis 2	160°/s	180°/s
Axis 3	170°/s	185°/s
Axis 4	320°/s	385°/s
Axis 5	400°/s	400°/s
Axis 6	460°/s	460°/s

Electrical connections

Supply voltage	200-600 V, 50-60 Hz
Power consumption	ISO-Cube at max speed 0.58 kW

Environment

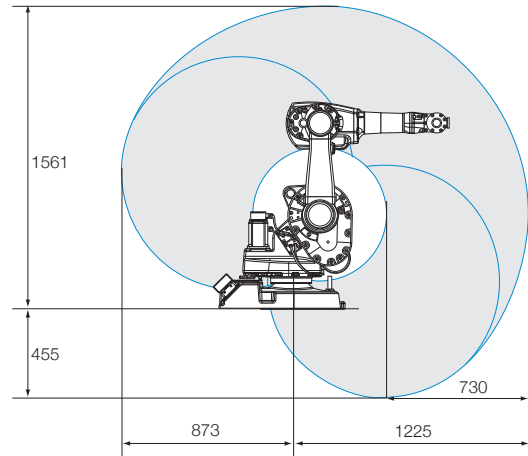
Ambient temperature for mechanical unit:

During operation	+ 5°C (41°F) to + 45°C (113°F)
During transportation and storage	- 25°C (- 13°F) to + 55°C (131°F)
For short periods (max 24h)	up to + 70°C (158°F)
Relative humidity	Max. 95% at constant temperature
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

Working range

IRB 1600-x/1.2



IRB 1600-x/1.45

