



**Universidad de Valladolid**

**TRABAJO FIN DE MÁSTER**

**MÁSTER EN PROFESOR DE EDUCACIÓN SECUNDARIA  
OBLIGATORIA Y BACHILLERATO, FORMACIÓN  
PROFESIONAL Y ENSEÑANZAS DE IDIOMAS**

**Especialidad de Tecnología e Informática**

**PROPUESTA DIDÁCTICA:**

**“APRENDIZAJE DE PROGRAMACIÓN DE  
ROBOTS EN EL SECTOR EDUCATIVO”**

**DIDACTIC PROPOSAL:**

**“LEARNING ROBOTS PROGRAMMING IN THE  
EDUCATION SECTOR”**

**Autor:**

**D. Luis José Cenizo Revuelta**

**Tutor:**

**Dr. D. Eduardo J. Moya de la Torre**

**Valladolid, 05 de Julio de 2021**





## Resumen

El objetivo de la presente memoria es la creación de una herramienta que permita a los alumnos de ciclos de grado medio o superior de las ramas de automatización y electrónica, profundizar en el conocimiento de los principios básicos de funcionamiento de los Robots industriales, así como iniciarse en el manejo de una herramienta de programación profesional que les permita incrementar sus posibilidades de integración en el mundo profesional una vez finalizados sus estudios.

Por tanto, ésta memoria se construye con el objetivo de ser un complemento a la formación académica recibida durante su formación, pero también con la posibilidad de ser integrada dentro de la programación reglada de los diferentes ciclos descritos en ésta memoria. La memoria está diseñada cómo si fuera una unidad didáctica con sus objetivos didácticos, contenidos, resultados de aprendizaje, metodología utilizada y su temporalización.

Dentro de los contenidos el autor ha decidido incluir aquellos conceptos teóricos que, bajo su experiencia profesional, considera claves para la formación del estudiante, explicados siempre de una forma sencilla y lo más visual posible, apoyándonos siempre en el software profesional en cuestión, de modo que cada concepto siempre tenga una aplicación inmediata.

Ésta memoria se completa con seis actividades realizadas en modalidad de “video learning”, dónde el estudiante repasará los conocimientos teóricos adquiridos y a su vez se iniciará en el manejo de una herramienta profesional de programación utilizada en el mundo de la Robótica Industrial.

Palabras claves: Aprendizaje, Programación, Robots, Educativo, RobotStudio.



Máster en profesor de E.S.O y Bachillerato, Formación Profesional y Enseñanza de Idiomas  
Especialidad Tecnología e Informática



## Abstract

The main goal of this report is the creation of a tool that allows students of middle or higher degree cycles of the branches of automation and electronics increase their knowledge of the basic principles of operation of industrial robots. In addition, to start in the management of a professional programming tool that allows them to increase their possibilities of integration into the professional world.

Therefore, this report is built with the aim of being a complement to the academic formation received during their training stage, but also with the possibility to be integrated into the regulated programming of the different cycles described in this report.

The memory is designed like a didactic unit including objectives, contents, learning results, methodology used and scheduling. Within the contents, the author has decided to include the main theoretical concepts that under his professional experience he considers keys to the student's training developed always in a simple and visual way.

This report is completed with six activities carried out in a "video learning" mode, by which the students will be able to review the main concepts included in this report as well as take their first steps managing a professional programming tool used in the world of Industrial Robots.

Keywords: Learning, Programming, Robots, Education, RobotStudio.



Máster en profesor de E.S.O y Bachillerato, Formación Profesional y Enseñanza de Idiomas  
Especialidad Tecnología e Informática



## Índice de Contenido

|          |  |    |
|----------|--|----|
| 1.       | INTRODUCCIÓN Y OBJETIVOS.....              | 1  |
| 1.1.     | Conocimientos previos.....                 | 1  |
| 1.2.     | Contexto.....                              | 1  |
| 1.3.     | Justificación.....                         | 2  |
| 1.4.     | Objetivos Didácticos.....                  | 5  |
| 1.5.     | Estructura Memoria.....                    | 6  |
| 2.       | MARCO TEÓRICO.....                         | 7  |
| 2.1.     | HARDWARE ROBOT.....                        | 7  |
| 2.1.1.   | Unidad de programación.....                | 7  |
| 2.1.2.   | Controlador IRC5.....                      | 8  |
| 2.1.3.   | La unidad mecánica o Robot.....            | 8  |
| 2.1.4.   | RobotStudio.....                           | 9  |
| 2.1.5.   | Cadena seguridad.....                      | 10 |
| 2.2.     | CONCEPTOS ROBOTWARE Y RAPID.....           | 11 |
| 2.2.1.   | Software Robotware.....                    | 11 |
| 2.2.2.   | Marco Rapid.....                           | 11 |
| 2.3.     | HERRAMIENTAS Y OBJETO DE TRABAJO.....      | 13 |
| 2.3.1.   | Herramienta trabajo.....                   | 13 |
| 2.3.2.   | Objeto trabajo.....                        | 14 |
| 2.4.     | SISTEMAS DE COORDENADAS.....               | 14 |
| 2.5.     | CINEMÁTICA DEL ROBOT.....                  | 16 |
| 3.       | APLICACIÓN PRÁCTICA. ROBOTSTUDIO.....      | 19 |
| 3.1.1.   | Creación de una estación.....              | 19 |
| 3.1.2.   | Incorporar componentes a la estación.....  | 20 |
| 3.1.3.   | Creación de un sistema.....                | 23 |
| 4.       | PROGRAMACIÓN DE ROBOTS.....                | 25 |
| 4.1.1.   | Pasos en la programación de un Robot.....  | 25 |
| 4.1.2.   | Creación de una herramienta.....           | 25 |
| 4.1.3.   | Sistema coordenadas herramienta.....       | 27 |
| 4.1.3.1. | Sistema coordenadas RobotStudio.....       | 27 |
| 4.1.3.2. | Sistema coordenadas Flexpendant.....       | 29 |
| 4.1.4.   | Creación objeto trabajo.....               | 32 |
| 4.1.5.   | Sistema coordenadas objeto de trabajo..... | 35 |



|          |  |    |
|----------|--|----|
| 4.1.6.   | Creación de objetivos o puntos de trabajo. ....                  | 37 |
| 4.1.6.1. | Ajuste sistema coordenadas objeto-herramienta.....               | 38 |
| 4.1.7.   | Trayectorias.....  | 39 |
| 4.1.8.   | Tipos de datos. ....   | 43 |
| 4.1.8.1. | Datos de configuración del robot. Confdata. ....                 | 43 |
| 4.1.8.2. | Orientación. Orient. ....  | 44 |
| 4.1.8.3. | Datos de posición. RobTarget. ....                               | 46 |
| 4.1.8.4. | Herramienta de trabajo. Tooldata. ....                           | 47 |
| 4.1.8.5. | Datos del objeto de trabajo. Wobjdata.....                       | 48 |
| 4.2.     | Señales Entrada/Salida.....                                      | 49 |
| 5.       | SIMULAR PROGRAMAS. ....  | 51 |
| 5.1.1.   | Conceptos básicos.....   | 51 |
| 5.1.2.   | Detección de colisiones.....                                     | 51 |
| 5.1.3.   | Movimiento objetos.....  | 54 |
| 5.1.4.   | Componentes inteligentes. ....                                   | 55 |
| 5.1.5.   | Simulación de señales de E/S.....                                | 58 |
| 6.       | ACTIVIDADES A DESARROLLAR.....                                   | 61 |
| 6.1.     | Actividad1. Inicio al Software RobotStudio.....                  | 62 |
| 6.2.     | Actividad 2. Controladora y modelado de piezas. ....             | 63 |
| 6.3.     | Actividad 3. Creación de herramientas de trabajo. ....           | 64 |
| 6.4.     | Actividad 4. Creación de trayectorias. ....                      | 65 |
| 6.5.     | Actividad 5. Creación de objetos de trabajo. ....                | 66 |
| 6.6.     | Actividad 6. Entradas y Salidas virtuales. Smart Components..... | 67 |
| 7.       | TEMPORALIZACIÓN.....   | 69 |
| 8.       | RESULTADOS DE APRENDIZAJE. ....                                  | 71 |
| 9.       | METODOLOGÍA Y EVALUACIÓN .....                                   | 73 |
| 9.1.     | METODOLOGÍA.....   | 73 |
| 9.1.1.   | Método trabajo parte teórica. ....                               | 73 |
| 9.1.2.   | Método trabajo parte práctica.....                               | 73 |
| 9.2.     | VIDEO LEARNING.....  | 74 |
| 9.2.1.   | Ventajas del video learning .....                                | 74 |
| 9.3.     | CRITERIOS DE EVALUACIÓN. ....                                    | 75 |
| 10.      | CONCLUSIONES .....   | 77 |
|          | BIBLIOGRAFÍA.....  | 79 |





|                  |    |
|------------------|----|
| ANEXOS .....     | 81 |
| Actividad 1..... | 81 |
| Actividad 2..... | 82 |
| Actividad 3..... | 83 |
| Actividad 4..... | 84 |
| Actividad 5..... | 85 |
| Actividad 6..... | 86 |



Máster en profesor de E.S.O y Bachillerato, Formación Profesional y Enseñanza de Idiomas  
Especialidad Tecnología e Informática



## Índice de Tablas.

|   |    |
|---|----|
| Tabla 1. Secuenciación y Distribución horaria semanal de los módulos profesionales..... | 3  |
| Tabla 2. Espacios y equipamientos mínimos.....  | 4  |
| Tabla 3. Secuenciación Unidades didácticas IES "Segundo de Chomón" Teruel. ....         | 5  |
| Tabla 4. Partes Aplicación RAPID. ....  | 12 |
| Tabla 5. Temporalización contenidos y actividades TFM.....                              | 69 |



Máster en profesor de E.S.O y Bachillerato, Formación Profesional y Enseñanza de Idiomas  
Especialidad Tecnología e Informática



## Índice de Figuras.

|   |    |
|---|----|
| Figura 1. Componentes principales de la unidad de programación. ....                | 7  |
| Figura 2. Botones hardware unidad de programación. ....                             | 7  |
| Figura 3. Tipos de controlador IRC5. ....   | 8  |
| Figura 4. Robot o unidad mecánica.....  | 8  |
| Figura 5. Interfaz gráfica de usuario.....  | 9  |
| Figura 6. Esquema general funcionamiento de la cadena de seguridades.....           | 10 |
| Figura 7. Aplicación RAPID .....  | 11 |
| Figura 8. Herramienta de trabajo.....   | 13 |
| Figura 9. Orientación muñeca según TCP. ....  | 13 |
| Figura 10. Jerarquía sistemas de coordenadas. ....                                  | 14 |
| Figura 11.Sistema de coordenadas de la base. ....                                   | 15 |
| Figura 12. Sistema coordenadas objeto trabajo. ....                                 | 15 |
| Figura 13. Sistema de coordenadas de la herramienta.....                            | 16 |
| Figura 14. Interrelación sistemas coordenados. ....                                 | 16 |
| Figura 15. Matriz de transformación homogénea. ....                                 | 17 |
| Figura 16. Interpretación geométrica de la matriz de transformación homogénea. .... | 17 |
| Figura 17. Matriz básica de rotación. ....  | 18 |
| Figura 18. Ejemplo cálculo matriz transformación. ....                              | 18 |
| Figura 19. Diagrama flujo creación solución RobotStudio. ....                       | 19 |
| Figura 20. Solución con estación vacía.....   | 19 |
| Figura 21. Selección del robot.....   | 20 |
| Figura 22. Importar herramienta. ....   | 21 |
| Figura 23. Herramienta insertada en la estación. ....                               | 21 |
| Figura 24. Importar Geometría. ....   | 22 |
| Figura 25. Guardar como biblioteca.....   | 22 |
| Figura 26. Creación sistema. ....   | 23 |
| Figura 27. Pasos creación sistema.....  | 23 |
| Figura 28. Secuencia programación Robots.....                                       | 25 |
| Figura 29. Modelado herramienta. ....   | 25 |
| Figura 30. Crear herramienta. ....  | 26 |
| Figura 31. Crear herramienta. TCP.....  | 26 |
| Figura 32. Asociar herramienta TCP robot.....                                       | 27 |
| Figura 33. Modelado herramienta doble. ....   | 28 |
| Figura 34. Agrupación objetos herramienta doble. ....                               | 28 |
| Figura 35. Herramienta con dos TCP.....   | 29 |
| Figura 36. Herramienta con dos TCP en robot.....                                    | 29 |
| Figura 37. Métodos definir TCP.....   | 31 |
| Figura 38. MétodoTCP (orient. predet.).....   | 31 |
| Figura 39. Crear "Objeto Trabajo".....  | 32 |
| Figura 40. Menú creación objeto trabajo. ....                                       | 33 |
| Figura 41. Modificar objeto de trabajo. ....  | 34 |
| Figura 42. Conectar objeto de trabajo. ....   | 34 |
| Figura 43. Menú selección Flexpendant. ....   | 35 |
| Figura 44. Menú definir objeto trabajo flexpendant. ....                            | 36 |
| Figura 45. Definir base coordenadas usuario y objeto. ....                          | 36 |



|  |    |
|--|----|
| Figura 46. Método tres puntos sistema coordenadas objeto .....             | 37 |
| Figura 47. Crear objetivo o punto de trabajo.....                          | 37 |
| Figura 48. Crear objeto o punto trabajo. Parámetros.....                   | 38 |
| Figura 49. Ver herramienta en punto.....                                   | 38 |
| Figura 50. Orientación puntos con objeto trabajo.....                      | 39 |
| Figura 51. Creación trayectoria vacía. ....                                | 39 |
| Figura 52. Trayectoria con puntos. ....                                    | 40 |
| Figura 53. Menú modificar punto. ....                                      | 41 |
| Figura 54. Modificar posición punto. Mano alzada.....                      | 42 |
| Figura 55. Copiar orientación de un punto. ....                            | 42 |
| Figura 56. Cuadrantes configuración eje 6.....                             | 43 |
| Figura 57. Configuraciones robot, misma posición.....                      | 44 |
| Figura 58. Rotación de un sistema de coordenadas mediante cuaternión. .... | 45 |
| Figura 59. Definición sistema de coordenadas de la herramienta.....        | 48 |
| Figura 60. Definición sistema de coordenadas del objeto de trabajo. ....   | 49 |
| Figura 61. Creación señales E/S.....                                       | 49 |
| Figura 62. Crear señal de simulación. ....                                 | 50 |
| Figura 63. Reiniciar controladora. ....                                    | 50 |
| Figura 64. Configuración de simulación. ....                               | 51 |
| Figura 65. Grupos de colisión A y B. ....                                  | 52 |
| Figura 66. Configuración comprobar colisiones.....                         | 53 |
| Figura 67. Visualización de colisión.....                                  | 53 |
| Figura 68. Cambio comportamiento físico pieza. ....                        | 54 |
| Figura 69. Guardar estado de simulación. ....                              | 54 |
| Figura 70. Creación componente inteligente. Modelado. ....                 | 55 |
| Figura 71. Creación componente inteligente. Simulación. ....               | 55 |
| Figura 72. Tipos de componentes inteligentes. ....                         | 56 |
| Figura 73. Diseño lógica componente inteligente.....                       | 57 |
| Figura 74. Diseño lógico de la estación. ....                              | 57 |
| Figura 75. Simular señales E/S. ....                                       | 58 |
| Figura 76. Listas de señales. ....   | 59 |
| Figura 77. Instrucción de acción con entrada. ....                         | 59 |



## 1. INTRODUCCIÓN Y OBJETIVOS.

Cualquiera que sea la aplicación, existe un robot que se ajusta a sus necesidades. Con ésta rotundidad los fabricantes de robots garantizan que sus productos se pueden adaptar a cualquier aplicación industrial y no tan industrial con la aparición de los cada vez más presentes robots colaborativos. La tecnología robótica está progresando con paso firme hacia los nuevos conceptos de robots, más móviles, autónomos e inteligentes, y muchas industrias están planeando recurrir a este tipo de sistemas para automatizar procesos y para complementar el trabajo humano, tanto en las fábricas y almacenes como en otros entornos de trabajo.

Ésta creciente demanda crea necesidad de profesionales con la formación adecuada para trabajar en entornos robotizados, o en colaboración con este tipo de máquinas. Por ello, los expertos instan a las empresas, gobiernos e instituciones académicas a que aborden este problema, que podría tener consecuencias tan graves como la escasez de talento en nuevas materias digitales, como la inteligencia artificial o el big data.

Con éste trabajo se pretende dotar de una herramienta de apoyo al docente que desempeña su labor en los centros educativos en los que se imparten las especialidades asociadas a éste sector, facilitando el aprendizaje del alumno, así como aumentar su motivación por continuar progresando en el conocimiento de las herramientas y desarrollo de las habilidades adecuadas necesarias para trabajar con robots y sistemas de automatización inteligentes.

### 1.1. Conocimientos previos.

El documento pretende servir de apoyo a estudiantes de las especialidades tecnológicas continuando la formación ya adquirida en etapas anteriores, de cuyos conocimientos se parte. Profundizaremos en las destrezas claves y en la adquisición de los conocimientos básicos de los que se parte en la elaboración de cualquier proyecto robotizado. Para ello utilizaremos un software profesional que permite diseñar la programación y simular el funcionamiento de una célula robotizada. Se describirá paso a paso y mediante información gráfica, teoría de robots vinculada al software, el manejo del software, aprendizaje práctico mediante la realización de actividades tutorizadas con videos.

Entre los conocimientos previos deseados para garantizar el máximo aprovechamiento del contenido del documento podemos citar:

- Conocimientos de algún lenguaje de programación.
- Tipos de datos usados.
- Uso de funciones y rutinas en programación.
- Conocimientos lógica digital.
- Manejo genérico Windows.

### 1.2. Contexto.

Valladolid se sitúa en el centro de una comunidad como es Castilla y León, dónde la industria alimentaria siempre ha jugado un papel muy importante. Dada la situación actual, debido al coronavirus, se está comprobando como el sector de la alimentación se ha mostrado como uno de los sectores con mayor resiliencia, que, gracias a su esencialidad, ha podido mantener, e incluso aumentar, la producción en durante los meses de confinamiento de la población.

Se trata de un sector estratégico, con una alta capilaridad, cuya implantación territorial es clave en la fijación de la población y en el desarrollo local de las zonas rurales. Para adaptarse a las circunstancias y mantener su carácter vital en el desarrollo de la economía necesita implantar reformas de digitalización y adaptación a las nuevas tecnologías.

El avance de la tecnología ha permitido cumplir con todos los estándares de calidad necesarios y, hoy en día, las ventajas y aplicaciones de la robótica colaborativa en la alimentación y en la



agricultura han revolucionado la forma de trabajar de esta industria. Los cobots, por ejemplo, asumen las tareas más duras, tediosas y peligrosas para los trabajadores, mejoran la uniformidad de los procesos y por ende la calidad, y, sobre todo, permiten aumentar la productividad.

Éste documento se dirige principalmente a centros educativos dónde se impartan ciclos de grado medio o superior con un grado de vinculación a la robótica importante, como pueden ser:

- Técnico Superior en Automatización y Robótica Industrial.
- Técnico Superior en Mantenimiento Electrónico.
- Técnico Superior en Sistemas Electrotécnicos y Automatizados.

### 1.3. Justificación.

El encaje del presente trabajo fin de master podemos encontrarlo en el módulo profesional "Robótica Industrial", incluido dentro del ciclo de grado superior de Técnico Superior en Automatización y Robótica Industrial. Para ello vamos a proceder a analizar los contenidos teóricos que establece el BOE-A-2013-1033 para dicho módulo, que podemos ver a continuación.

1. Reconocimiento de diferentes tipos de robots y/o sistemas de control de movimiento.
  - Aplicaciones de robots y/o sistemas de control de movimiento (motion control).
  - Tipología de los robots.
  - Análisis de sistemas de seguridad en entornos robotizados.
  - Morfología de un robot. Elementos constitutivos. Grados de libertad.
  - Sistemas mecánicos: elementos mecánicos. Sistemas de transmisión. Transformación de movimiento.
  - Útiles y herramientas del robot.
  - Unidades de control de robots.
  - Sistemas de control de movimiento.
  - Unidades de programación.
  - Sistemas teleoperados para el control de manipuladores y/o robots.
  - Sistemas de guiado.
2. Configuración de instalaciones de robots y/o sistemas de control de movimiento en su entorno.
3. Programación de robots y sistemas de control de movimiento:
  - Posicionamiento de robots. Operaciones lógicas aplicadas a la programación de robots.
  - Lenguajes de programación de robots.
  - Programación secuencial.
  - Programación de sistemas de control de movimiento.
4. Verificación del funcionamiento de robots y/o sistemas de control de movimiento:
  - Técnicas de verificación.
  - Monitorización de programas.
  - Instrumentos de medida.
  - Reglamentación vigente.
5. Reparación de averías en entornos industriales robotizados y/o de control de movimiento.

Si bien los contenidos que establece la ley, son bastante orientativos y sujetos a interpretación, podemos concluir que los puntos 1,3 y 4, es dónde podríamos encajar el contenido de éste TFM.





La temporización es la que podemos ver en la Tabla 1, de 80 horas repartidas en 4 h semanales, dentro del segundo curso del ciclo superior. Podemos considerar que el número de horas asignadas al módulo resulta escaso para la cantidad de contenidos citados anteriormente, es por ello que con el presente TFM pretendemos conseguir un mayor aprovechamiento del horario lectivo y además constituir una herramienta de consulta y guía para las actividades extraescolares.

*Ciclo Formativo de Grado Superior: Automatización y Robótica Industrial*

| Módulo profesional  | Duración (horas) | Primer curso (h/semana) | Segundo curso           |                     |
|---|------------------|-------------------------|-------------------------|---------------------|
|   |                  |                         | 2 trimestres (h/semana) | 1 trimestre (horas) |
| 0959. Sistemas eléctricos, neumáticos e hidráulicos.        | 135              | 4                       |                         |                     |
| 0960. Sistemas secuenciales programables.                   | 150              | 5                       |                         |                     |
| 0961. Sistemas de medida y regulación.                      | 135              | 4                       |                         |                     |
| 0962. Sistemas de potencia.                                 | 175              | 5                       |                         |                     |
| 0963. Documentación técnica.                                | 95               | 3                       |                         |                     |
| 0964. Informática industrial.                               | 90               | 3                       |                         |                     |
| 0970. Formación y orientación laboral.                      | 90               | 3                       |                         |                     |
| Horario reservado para el módulo impartido en inglés.       | 90               | 3                       |                         |                     |
| 0965. Sistemas programables avanzados.                      | 90               |                         | 5                       |                     |
| <b>0966. Robótica industrial.</b>                           | <b>80</b>        |                         | 4                       |                     |
| 0967. Comunicaciones industriales.                          | 165              |                         | 8                       |                     |
| 0968. Integración de sistemas de automatización industrial. | 165              |                         | 8                       |                     |
| 0971. Empresa e iniciativa emprendedora.                    | 60               |                         | 3                       |                     |
| Horario reservado para el módulo impartido en inglés.       | 40               |                         | 2                       |                     |
| 0972. Formación en centros de trabajo.                      | 400              |                         |                         | 400                 |
| 0969. Proyecto de automatización y robótica industrial.     | 40               |                         |                         | 40                  |
| <b>Total en el ciclo formativo.....</b>                     | <b>2.000</b>     | <b>30</b>               | <b>30</b>               | <b>440</b>          |

Tabla 1. Secuenciación y Distribución horaria semanal de los módulos profesionales.

Así mismo el ANEXO IV del mismo documento, establece los equipamientos mínimos que deben incluir las aulas asignadas en la impartición del módulo. Como podemos observar en la Tabla 2 es de obligado cumplimiento la posesión de software de simulación de sistemas de automatización y robótica industrial. Por tanto, podemos considerar que el software utilizado para el TFM se adapta perfectamente a los requerimientos de la ley, como herramienta a utilizar en la impartición del módulo.



## ANEXO IV

### Espacios y equipamientos mínimos

Espacios:

| Espacio formativo                    | Superficie m <sup>2</sup> |            |
|--------------------------------------|---------------------------|------------|
|                                      | 30 alumnos                | 20 alumnos |
| Aula polivalente.                    | 60                        | 40         |
| Aula de informática.                 | 120                       | 80         |
| Laboratorio de sistemas automáticos. | 180                       | 120        |
| Taller de sistemas automáticos.      | 200                       | 130        |

Equipamientos mínimos:

| Espacio formativo                    | Equipamiento   |
|--------------------------------------|--|
| Aula polivalente.                    | Sistema de proyección.<br>Ordenadores en red y con acceso a Internet.<br>Dispositivos de almacenamiento en red.<br>Escáner.<br>Impresoras.<br>Equipos audiovisuales.   |
| Aula de informática.                 | Sistema de proyección.<br>Ordenadores en red y con acceso a Internet.<br>Escáner.<br>Plotter.<br>Programas de gestión de proyectos.<br>Impresoras.<br>Equipos audiovisuales.<br>Software de diseño y simulación de sistemas de automatización y robótica industrial.<br>Software de desarrollo de SCADA. |
| Laboratorio de sistemas automáticos. | Sistema de proyección.<br>Ordenadores en red y con acceso a Internet.<br>Impresoras.   |

Tabla 2. Espacios y equipamientos mínimos.

Si bien el contenido del BOE nos sirve de guía, de cómo debe estructurarse el módulo profesional, podemos ir un poco más allá analizando la programación que un centro en concreto establece en cuanto a unidades didácticas y temporización.

#### UNIDADES DIDÁCTICAS.

U.D.0.- Presentación.

U.D.1.- Análisis de robots y sistemas de control de movimiento.

U.D.2.- Implementación de robots en entornos simulados.

U.D.3.- Puesta en servicio y ajustes sobre robot real.

U.D.4.- Comunicación con PLC.



| TRIMESTRE                          | BLOQUES | UNIDADES DIDÁCTICAS |   |              |
|------------------------------------|---------|---------------------|---|--------------|
|                                    |         | Nº                  | DENOMINACION  | SESIONES (*) |
| 1º<br>21/09/2016<br>a<br>7/12/2016 |         | 0                   | Presentación  | 3            |
|                                    | 1       | 1                   | Análisis de robots y sistemas de control de movimiento. | 12           |
|                                    | 2       | 2                   | Implementación de robots en entornos simulados.         | 15           |
| 2º<br>14/12/2016<br>a<br>8/03/2017 | 3-4-5   | 3                   | Puesta en servicio y ajustes sobre robot real.          | 18           |
|                                    | 2       | 4                   | Comunicación con PLC.                                   | 9            |
|                                    |         |                     | Examen 1ª-2ª y Recuperación Final                       | 9            |
| Total:                             |         |                     |   | <b>66</b>    |

*Este módulo tiene de un total de 63 sesiones, el curso dispone de 66 sesiones, se pueden dedicar a la recuperación o refuerzo de contenidos. La temporización para cada trimestre se ajusta al calendario escolar del presente curso académico. (\*) Una sesión corresponde a 50 minutos.*

Tabla 3. Secuenciación Unidades didácticas IES "Segundo de Chomón" Teruel.

Podemos concluir, por tanto, que los contenidos del presente TFM encajan perfectamente en el módulo profesional de Robótica Industrial, aportando contenido teórico y práctico que puede ser utilizado como herramienta didáctica de apoyo en los centros en los que se imparte. Como herramienta de complemento que permite profundizar en la materia con el fin último de que el alumno consiga los conocimientos adecuados para poder afrontar el paso del mundo educativo al profesional con garantías.

#### 1.4. Objetivos Didácticos.

Los objetivos, en el proceso de enseñanza-aprendizaje, son los que guían la acción docente, son proposiciones razonadas para que el alumnado alcance una capacitación completa. Son el marco de referencia y la "ayuda para desarrollar con mayor calidad y eficacia el proceso educativo".

A partir del Real Decreto 1581/2011, de 4 de noviembre, por el que se establece el currículo del ciclo formativo de grado superior correspondiente al título de Técnico Superior en Automatización y Robótica Industrial, se organizan los objetivos generales, entre los que destacan en relación con la unidad didáctica en la que podríamos incluir éste TFM:

- Programación de robots y sistemas de control de movimiento:
  - Posicionamiento de robots. Operaciones lógicas aplicadas a la programación de robots.
  - Lenguajes de programación de robots.
  - Programación secuencial.
  - Programación de sistemas de control de movimiento.

Dentro de los objetivos didácticos se establecen las intenciones del profesor respecto al módulo profesional "Robótica Industrial" y de cada una de las unidades didácticas que lo componen. En nuestro caso podemos definir unos objetivos didácticos o de aprendizaje, que servirán de punto de partida en la elaboración del TFM y que a su vez constituyen el fin último de su creación.



Los objetivos que se establecen en el desarrollo de éste TFM son los siguientes:

- Creación estación robotizada con software profesional.
- Creación herramientas, objetos, variables según estándar profesional.
- Aprendizaje lenguaje programación robot.
- Aprendizaje del manejo de un software profesional de diseño.
- Comprensión de los principios teóricos básicos de la estructura y funcionamiento de los Robots.
- Utilización de E/S digitales en un sistema automatizado como elementos de activación.

### 1.5. Estructura Memoria.

Comienza la memoria con el marco teórico en el que se sustenta, por tanto, describe los diferentes elementos físicos a los que se hace referencia y que serán usados, como son el hardware del robot, la unidad de programación o el software que utilizaremos para el desarrollo de las actividades.

Continúa el marco teórico, con una explicación teórica de los conceptos esenciales que se trabajan en la programación de robots y que posteriormente serán utilizados en las actividades, con el objetivo de que el alumno adquiera una base sólida de conocimientos teóricos que pueda asociar en la fase posterior de realización de actividades y utilización del software.

A continuación, se pasa a la aplicación práctica de los conocimientos adquiridos mediante el software de programación de Robots. En ésta sección se comenzará a utilizar el software de programación y comenzaremos a crear los diferentes componentes que constituyen una estación robotizada. Se realizará de forma visual, mediante capturas de pantalla del software que incluirán textos explicativos de los pasos que se deben ir dando.

Una vez iniciados en el software, comenzaremos con el aprendizaje de programación con el software RobotStudio. Igual que en el apartado anterior, se irán creando elementos de programación paso por paso, mediante imágenes capturadas del software que ayuden al estudiante en su proceso de aprendizaje. Se crearán todos los elementos vistos durante el marco teórico.

El siguiente apartado, aborda la simulación de estaciones robotizadas mediante el software. Se aprenderá a simular estaciones mediante el software RobotStudio, el concepto de entrada/salida digital, como simularlas y por último aprenderemos a usar una herramienta que incluye el software denominada componentes inteligentes, que nos servirán para simular estaciones de una forma más real mediante la programación por bloques.

Ésta memoria se complementa con una serie de videos en los que se trabajan todos los conceptos aprendidos a lo largo de seis actividades. En el apartado seis se describen cada una de ellas de forma pormenorizada con sus objetivos, materiales, desarrollo y duración y en los anexos aparecen las portadas de cada una de ellas con un código QR para visualizarlas.

Finalmente, la memoria incluye un apartado de temporalización, resultados de aprendizaje, así como de metodología y evaluación que indican la forma de integrar ésta memoria en los contenidos oficiales del módulo de Robótica Industrial dentro del ciclo superior de Técnico Superior en Automatización y Robótica Industrial.



## 2. MARCO TEÓRICO.

### 2.1. HARDWARE ROBOT.

En éste apartado daremos una descripción general de la unidad de programación, el controlador IRC5 y RobotStudio.

Un sistema robot IRC5 básico suele componerse de un controlador, la unidad de programación y uno o varios robots o unidades mecánicas de otros tipos. En ésta unidad se describe un sistema IRC5 básico.

#### 2.1.1. Unidad de programación.

La unidad de programación es un dispositivo que maneja muchas de las funciones relacionadas con el uso del sistema de robot, como ejecutar programas, mover el manipulador, crear y editar programas de aplicación, etc.

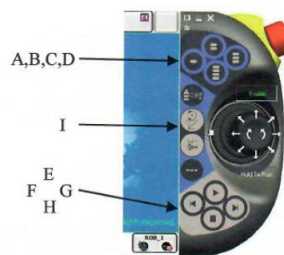
Se compone de elementos de hardware, como botones y un joystick, y de software. Se conecta al módulo de controlador a través de un cable con conector integrado.



|   |                                |
|---|--------------------------------|
| A | Cable conexión                 |
| B | Pantalla táctil                |
| C | Pulsador de paro de emergencia |
| D | Dispositivo de habilitación    |
| E | Joystick                       |
| F | Interface USB                  |

Figura 1. Componentes principales de la unidad de programación.

La unidad de programación cuenta con varios botones de hardware dedicados. Cuatro de ellos son programables y otros cuatro están pre-programados.



|         |  |
|---------|--|
| A,B,C,D | Teclas programables.   |
| E       | Botón INICIAR. Inicia la ejecución del programa.   |
| F       | Botón RETROCEDER un paso. Ejecuta el programa una instrucción hacia atrás.   |
| G       | Botón AVANZAR un paso. Ejecuta el programa una instrucción hacia delante.  |
| H       | Botón DETENER. Detiene la ejecución del programa.  |
| I       | 4 botones adicionales para el acceso rápido a la selección de movimientos (Unidad de Robot / ejes Externos, Reorientación / Lineal, Ejes 1,2,3 / Ejes 4,5,6) y Activar / Desactivar Incrementos. |

Figura 2. Botones hardware unidad de programación.



### 2.1.2. Controlador IRC5.

El controlador IRC5 contiene todas las funciones necesarias para mover y controlar el robot. La variante básica del controlador IRC5, puede componerse de un solo armario o estar dividida en dos módulos separados: el módulo de control y el módulo de accionamiento.

El módulo de control contiene todos los elementos electrónicos de control, como el ordenador principal, las tarjetas de E/S y la unidad de almacenamiento.

El módulo de accionamiento contiene todos los elementos electrónicos de alimentación que proporcionan la alimentación a los motores del robot. Un módulo de accionamiento IRC5 puede contener nueve unidades de accionamiento y manejar seis ejes internos, además de dos o más ejes externos en función del modelo de robot.

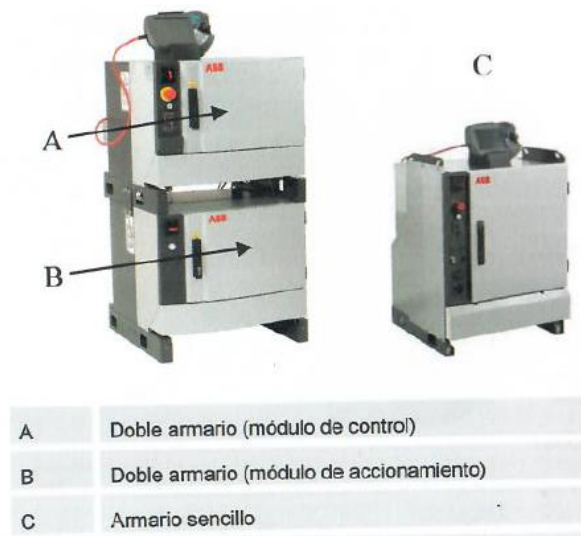


Figura 3. Tipos de controlador IRC5.

### 2.1.3. La unidad mecánica o Robot.

El Robot o unidad mecánica es el elemento del sistema robot encargado de realizar las tareas de manipulación y movimiento para las que ha sido programado. La configuración mecánica más común es la que tiene forma de brazo articulado y consta de un total de seis ejes.

Su base puede ser instalada directamente sobre un elemento de apoyo anclado al suelo (robot fijo), o sobre una unidad mecánica móvil como un TRACK para dotar al robot de más movilidad. En el extremo opuesto a la base, en el eje 6, es dónde se ubica la herramienta o útil con el que va a realizar las tareas el robot.

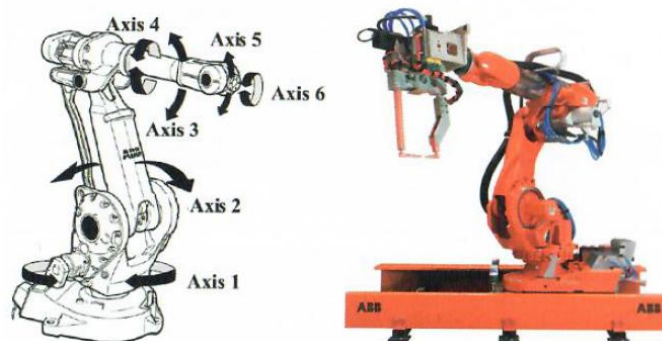


Figura 4. Robot o unidad mecánica.



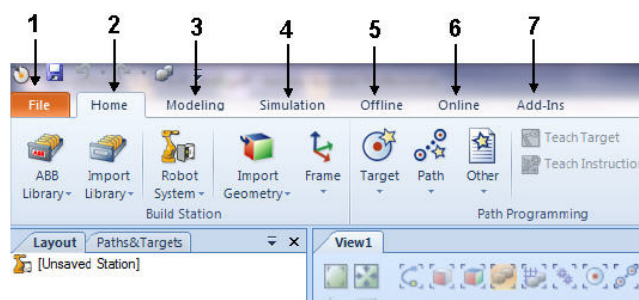
### 2.1.4. RobotStudio.

Robotstudio se ejecuta en un PC que debe estar conectado a la red de un controlador o al puerto de servicio del controlador, permite la programación y edición del programa, así como la carga del sistema operativo al robot. Para nuestro trabajo usaremos RobotStudio versión 6.08.

RobotStudio está clasificado en los dos niveles de funciones siguientes:

- **Basic:** Ofrece funciones seleccionadas de RobotStudio para configurar, programar y ejecutar un controlador virtual. También contiene funciones en línea para programación, configuración y monitorización de un controlador real conectado a través de Ethernet.
- **Premium:** Ofrece toda la funcionalidad de RobotStudio para programación fuera de línea y simulación de múltiples robots. El nivel Premium incluye las características del nivel Basic y requiere activación.

Además de la funcionalidad Premium, existen complementos, como los PowerPacs, y opciones para convertidores de CAD.



|   | Repuestos              | Descripción   |
|---|------------------------|---|
| 1 | Menú Archivo           | Contiene las opciones necesarias para crear una nueva estación, crear un sistema de robot, conectarse a un controlador, guardar la estación como un visor y otras opciones de RobotStudio. Para obtener más información, consulte |
| 2 | Pestaña Inicio         | Contiene los controles necesarios para construir estaciones, crear sistemas, programar trayectorias y colocar elementos. Para obtener más información, consulte   |
| 3 | Pestaña Modelado       | Contiene los controles necesarios para crear y agrupar componentes, crear cuerpos, mediciones y operaciones de CAD. Para obtener más información, consulte  |
| 4 | Pestaña Simulación     | Contiene los controles necesarios para crear, configurar, controlar, monitorizar y grabar simulaciones. Para obtener más información, consulte la   |
| 5 | Pestaña Fuera de línea | Contiene los controles necesarios para la sincronización, configuración y tareas asignadas al controlador virtual (VC). Para obtener más información, consulte las  |
| 6 | Pestaña En línea       | Contiene controladores para gestionar los controladores reales. Para obtener más información, consulte las  |
| 7 | Pestaña Complementos   | Contiene los controles de los PowerPac y de VSTA. Para obtener más información, consulte la   |

Figura 5. Interfaz gráfica de usuario.

### 2.1.5. Cadena seguridad.

La carta de seguridades, también conocida como carta panel, es el elemento del armario del robot dedicado a gestionar y controlar todas las seguridades asociadas al sistema robot (cadena de seguridades). Ésta unidad consta de una serie de bornes donde poder conectar eléctricamente los distintos dispositivos de seguridad, y de una serie de LED's que nos indican los mecanismos de emergencia y protección que han sido activados.

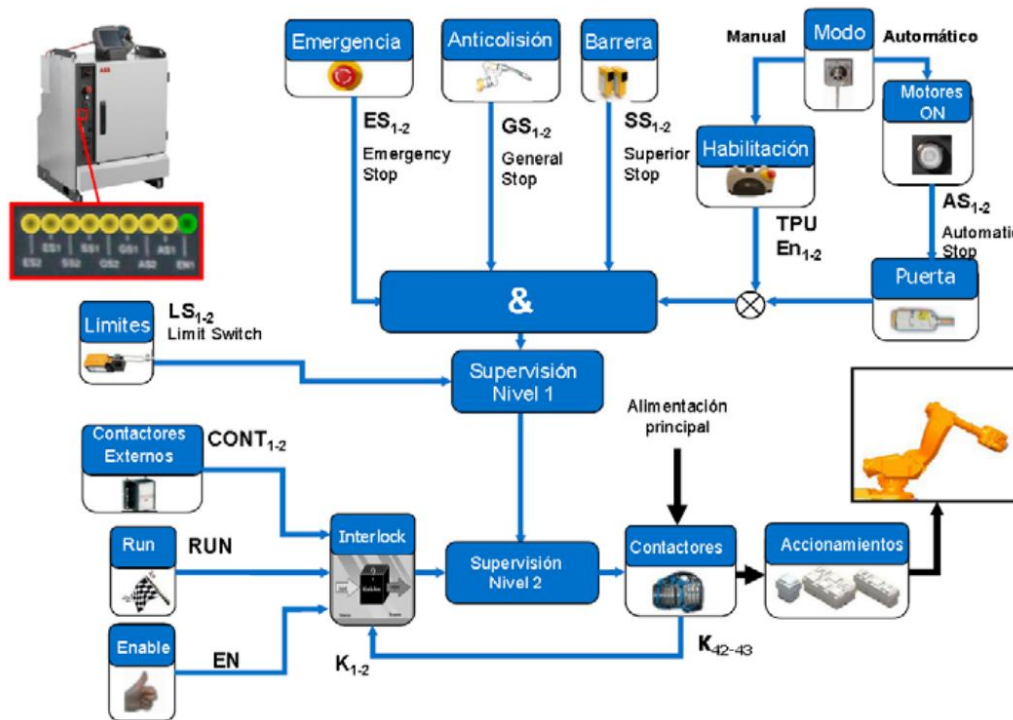


Figura 6. Esquema general funcionamiento de la cadena de seguridades.

Como se puede observar en el esquema el controlador cuenta además de con sus paros de emergencia integrados en la unidad de programación y armario del controlador (ES), con tres mecanismos de protección separados, el paro protegido general (GS), el paro protegido automático (AS) y el paro protegido supervisor (SS), que concurren en una doble cadena de seguridad constituida por dos contactores (k42-k43) que garantizan el corte de la alimentación principal al robot en caso de fallo o emergencia y detienen el robot hasta que el problema quede resuelto.

Podemos observar también como en modo manual la habilitación de movimientos pasa por tener pulsada la seguridad de hombre-muerto de la unidad de programación y en el nivel de supervisión 2 existe la posibilidad de programar un dispositivo de interlock que, en caso de parada por pérdida de una condición del mismo, requiera de cierto procedimiento para el rearme como puede ser realizar un reconocimiento de evento en la unidad de programación con un selector en posición off.





## 2.2. CONCEPTOS ROBOTWARE Y RAPID.

### 2.2.1. Software Robotware.

RobotWare Machine Tending es un software controlador flexible para el desarrollo y el control de los robots ABB. Incluye una interfaz gráfica intuitiva para el funcionamiento seguro y sin incidencias, el control y la supervisión fáciles de la producción, y la selección automática de piezas y programas.

Entre sus características se incluyen:

- Pantalla gráfica para el operador.
- Control de programas y gestión de ciclo.
- Gestión de errores y configuración de la estación.
- Edición sin necesidad de reinicio y control de la pinza.
- Configuración de piezas y señales.
- Vuelta segura y automática a la posición de inicio.
- Rutina de configuración y servicio.
- Marco RAPID.

Por tanto, podemos afirmar que Robotware es el sistema operativo del robot y sobre el que se pueden instalar diferentes aplicativos diseñados específicamente para un fabricante, que incluya por ejemplo partes comunes, relacionadas con los procesos de mantenimiento y partes específicas a una aplicación o proceso. Ejemplo de aplicativo es PROMIA utilizado en Renault con modificaciones para zona soldadura o montaje.

### 2.2.2. Marco Rapid.

En IRC5, un programa de RAPID es una colección de archivos de módulo (.mod) y el archivo de programa (.pgf.) que hace referencia a todos los archivos de módulo. Al cargar un archivo de programa, todos los módulos de programa antiguos se reemplazan por aquéllos a los que se hace referencia en el archivo .pgf. Los módulos de sistema no se ven afectados por la carga de programas.

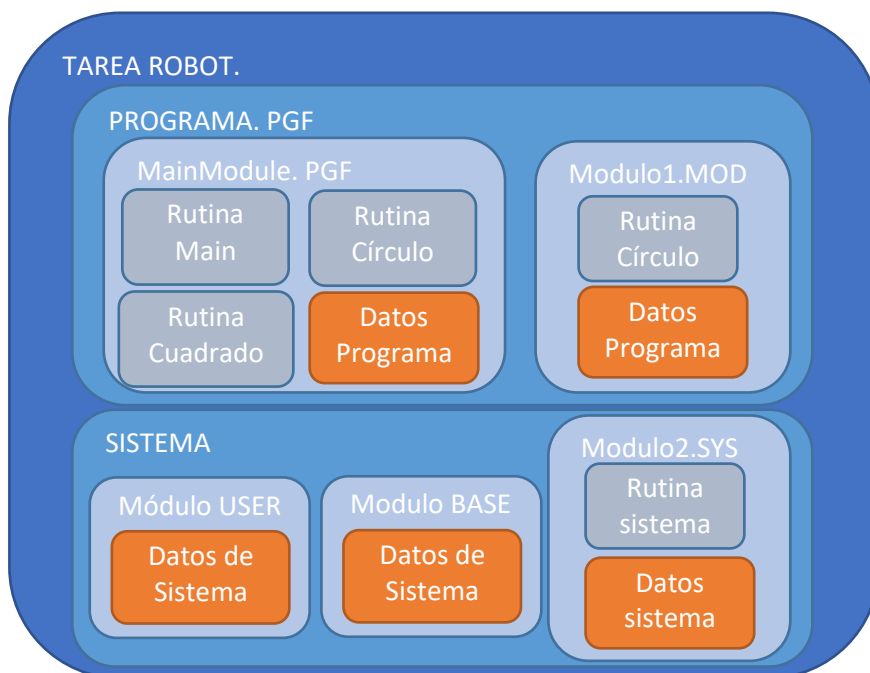


Figura 7. Aplicación RAPID



| Parte              | Función   |
|--------------------|---|
| Tarea              | Cada tarea contiene un programa de RAPID y módulos de sistema destinados a realizar una función determinada.<br>Cada aplicación de RAPID puede contener una tarea, y si tiene instalada la opción Multitarea, puede tener más de una tarea.   |
| Programa           | Cada programa contiene uno o varios módulos de programa con código de RAPID para distintos fines, por ejemplo el modulo que se crea por defecto es el módulo principal "Main Module".<br>Cada programa debe tener una rutina main definida como ejecutable  |
| Módulo de programa | Cada módulo de programa contiene datos y rutinas para una finalidad determinada.<br>El programa está dividido en módulos principalmente para ofrecer una mejor visión general y facilitar el manejo. Normalmente, cada módulo representa una acción concreta del robot.<br>Los módulos de programa suelen ser escritos por el usuario.<br>Todos los módulos de programa se eliminan al eliminar un programa de la memoria del controlador           |
| Rutina             | Una rutina contiene conjuntos de instrucciones, es decir, define qué debe hacer el sistema de robot.<br>Las rutinas también pueden contener los datos necesarios para las instrucciones.  |
| Rutina main        | Cada programa debe tener una rutina especial llamada "main" definida como punto de inicio de la ejecución del programa. De lo contrario, no es posible ejecutarlo.  |
| Instrucción        | Cada instrucción es una petición para que tenga lugar un evento determinado, por ejemplo "Mover el manipulador hasta un punto determinado", "Cambiar el valor de una salida digital determinada", "Esperar el valor de una entrada digital" .   |
| Datos              | Los datos son valores y definiciones establecidos en los módulos de programa o de sistema. Estos datos son utilizados por las instrucciones del mismo módulo o de varios módulos (su disponibilidad depende del tipo de dato).  |
| Módulo de sistema  | Cada módulo de sistema contiene datos y rutinas encaminadas a la realización de una función determinada.<br>El programa está dividido en módulos principalmente para ofrecer una mejor visión general y facilitar el manejo. Normalmente, cada módulo representa a una acción concreta del robot o una operación similar.<br>Los módulos de sistema suelen ser desarrollados por el fabricante del robot o los creadores de la línea de producción. |

Tabla 4. Partes Aplicación RAPID.

Es de reseñar que solo cabe un programa en el sistema. Los datos no guardados en el módulo USER se borrarán al borrar el módulo. Los datos en el módulo "user" son accesibles desde cualquier módulo y por tanto deben tener diferentes denominaciones.



## 2.3. HERRAMIENTAS Y OBJETO DE TRABAJO.

### 2.3.1. Herramienta trabajo.

Una herramienta es un objeto que puede montarse directa o indirectamente sobre la brida de sujeción del robot, o montarse en una posición fija dentro del área de trabajo del robot.

Todas las herramientas deben tener definido un TCP (punto central de la herramienta).

Cada herramienta que pueda ser utilizada por el robot debe ser medida y sus datos almacenados, para conseguir un posicionamiento exacto del punto central de la herramienta.

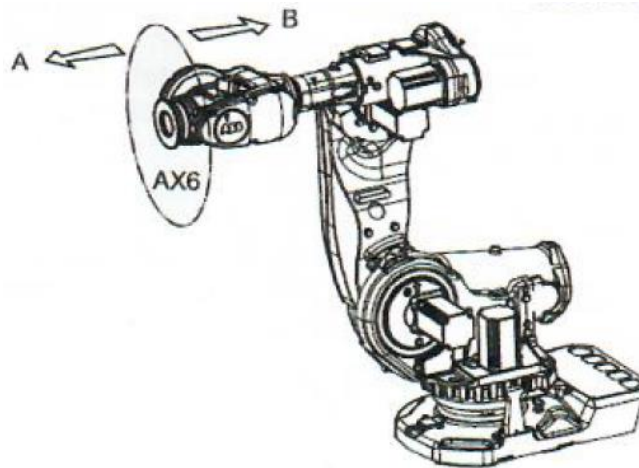


Figura 8. Herramienta de trabajo.

#### Punto central de la herramienta.

El punto central de la herramienta (TCP) es el punto respecto del cual se definen todas las posiciones del robot. Normalmente, el punto central de la herramienta se define respecto de una posición de la brida de sujeción del manipulador.

El punto central de la herramienta se mueve hasta la posición de destino programada. Este punto también constituye el origen del sistema de coordenadas de la herramienta.

La siguiente figura muestra el hecho de que el punto central de la herramienta (TCP) es el punto alrededor del cual se define la orientación de la muñeca existente entre la herramienta y el manipulador.

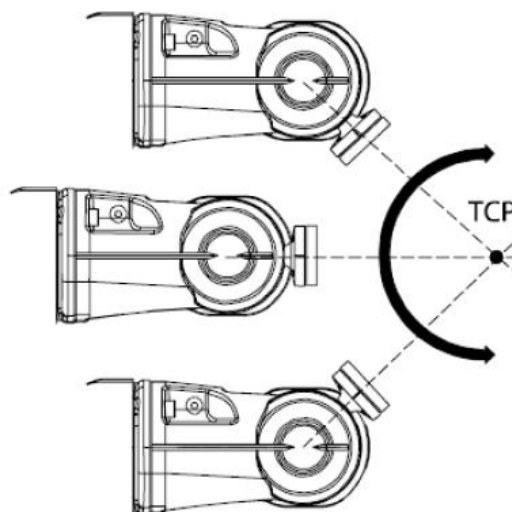


Figura 9. Orientación muñeca según TCP.



### 2.3.2. Objeto trabajo.

Un objeto de trabajo es un sistema de coordenadas que tiene asociadas determinadas propiedades específicas. Se utiliza principalmente para simplificar la programación durante la edición de programas debido a los desplazamientos asociados a tareas, objetos, procesos y otros elementos concretos.

El **sistema de coordenadas** del objeto de trabajo debe ser definido en dos bases de coordenadas: la base de coordenadas del usuario (dependiente de la base de coordenadas mundo) y la base de coordenadas del objeto (dependiente de la base de coordenadas del usuario).

## 2.4. SISTEMAS DE COORDENADAS.

Un sistema de coordenadas define un plano o un espacio con ejes, partiendo de un punto fijo conocido como origen. Los objetivos y las posiciones de robot se localizan mediante medidas a lo largo de los ejes de los sistemas de coordenadas.

Los robots utilizan varios sistemas de coordenadas, cada uno de ellos adecuado para tipos concretos de movimientos o programaciones.

- El sistema de coordenadas de la base se encuentra en la base del robot. Es la forma más fácil de mover únicamente el robot de una posición a otra.
- El sistema de coordenadas del objeto de trabajo depende de la pieza de trabajo y con frecuencia es el más adecuado para la programación del robot.
- El sistema de coordenadas de la herramienta define la posición de la herramienta que utiliza el robot al alcanzar los objetivos programados.
- El sistema de coordenadas mundo define la célula de robot. Todos los demás sistemas de coordenadas dependen del sistema de coordenadas mundo, ya sea de forma directa o indirectamente. Resulta útil en los movimientos, los movimientos en general y el manejo de estaciones y células con varios robots o bien robots movidos por ejes externos.
- El sistema de coordenadas del usuario resulta útil a la hora de representar equipos que sostienen otros sistemas de coordenadas, por ejemplo, objetos de trabajo.

Los sistemas de coordenadas están correlacionados jerárquicamente. El origen de cada sistema de coordenadas se define como una posición en uno de sus ascendientes.

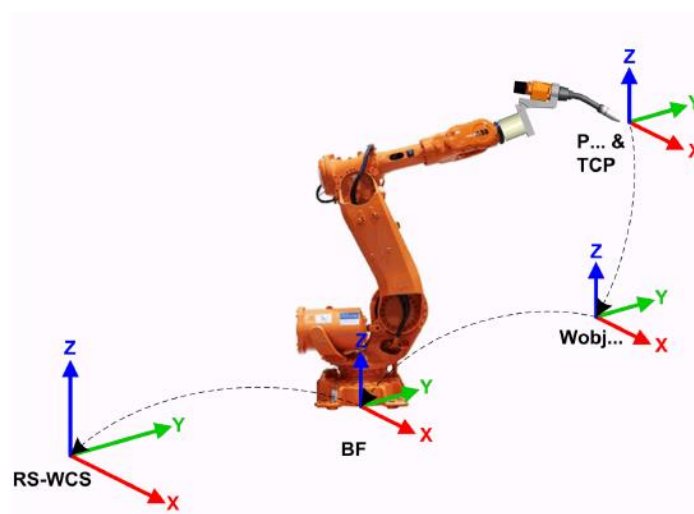


Figura 10. Jerarquía sistemas de coordenadas.

- **Sistema de coordenadas de la base.**

El sistema de coordenadas de la base tiene su punto cero en la base del robot, lo que hace que sus movimientos resulten predecibles en el caso de los robots con montaje fijo. Por tanto, resulta útil a la hora de mover un robot de una posición a otra. A la hora de programar un robot, suelen resultar más adecuados otros sistemas de coordenadas, como el sistema de coordenadas del objeto de trabajo.

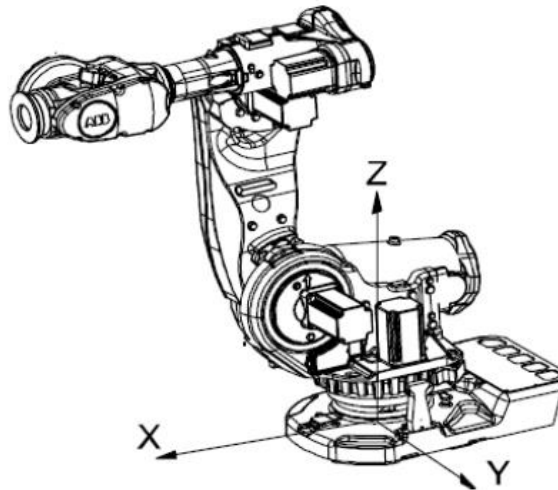


Figura 11. Sistema de coordenadas de la base.

- **Sistema coordenadas del objeto de trabajo.**

Con frecuencia, los objetos de trabajo se crean para simplificar el movimiento a lo largo de las superficies del objeto. Puede tener creados más de un objeto de trabajo, de forma que debe decidir cuál debe usarse para el movimiento.

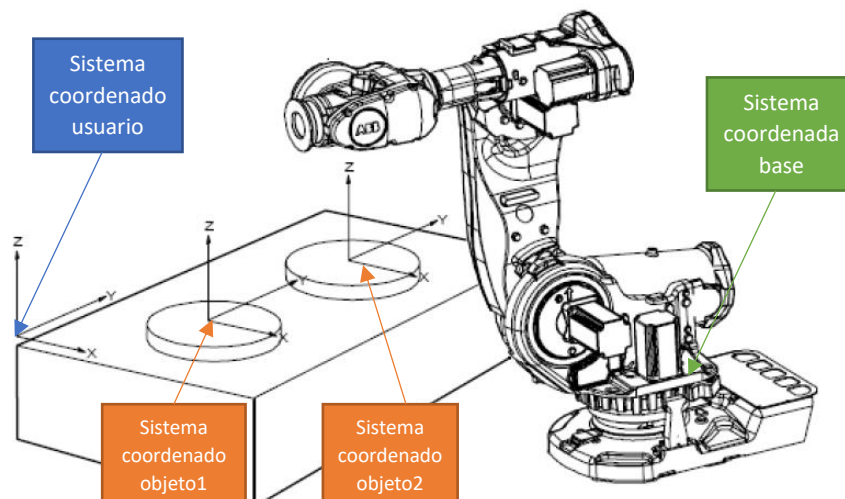


Figura 12. Sistema coordenadas objeto trabajo.

Es en estos sistemas de coordenadas de objetos de trabajo donde se crean los objetivos y trayectorias durante la programación del robot. Con ello se consiguen un sinnúmero de ventajas, al repositionar el objeto de trabajo en la estación, sólo es necesario cambiar la posición del sistema de coordenadas del objeto de trabajo para que todas las trayectorias se actualicen a la vez.

- **Sistema de coordenadas de la herramienta.**

El sistema de coordenadas de la herramienta tiene su origen, como se ha indicado anteriormente, en el punto central de la herramienta seleccionada. Por tanto, define la posición y la orientación de la herramienta. El sistema de coordenadas de la herramienta se abrevia con frecuencia como TCP.

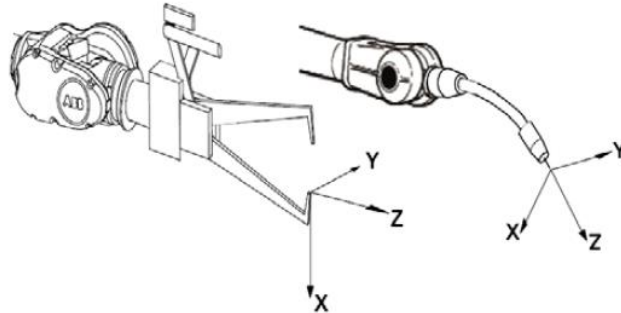


Figura 13. Sistema de coordenadas de la herramienta.

El TCP es el punto movido por el robot hasta las posiciones programadas al ejecutar el programa. Esto significa que si se cambia en una herramienta (la orientación del sistema de coordenadas de la herramienta), los movimientos del robot cambiarán de forma que el nuevo TCP alcance su objetivo.

Todos los robots tienen un sistema de coordenadas de herramienta predefinido, denominado "tool0" situado en el centro de la brida de sujeción del robot.

Al mover un robot, el sistema de coordenadas de la herramienta resulta útil para facilitar el cambio de la orientación de la herramienta durante la programación de movimientos.

- **Sistema de coordenadas del usuario.**

El sistema de coordenadas del usuario puede usarse para representar equipos como los accesorios o los bancos de trabajo. Con ello se consigue un nivel más en la cadena de sistemas de coordenadas relacionados entre sí, lo cual puede resultar útil a la hora de manejar equipos que sostienen objetos de trabajo u otros sistemas de coordenadas.

## 2.5. CINEMÁTICA DEL ROBOT.

Llegado a éste punto y conocidos todos los sistemas de referencia que se puede utilizar en Robotstudio, debemos tener claro que para que el robot pueda manipular una pieza, es necesario conocer su localización, es decir la posición y orientación de ésta con respecto a la base del robot.

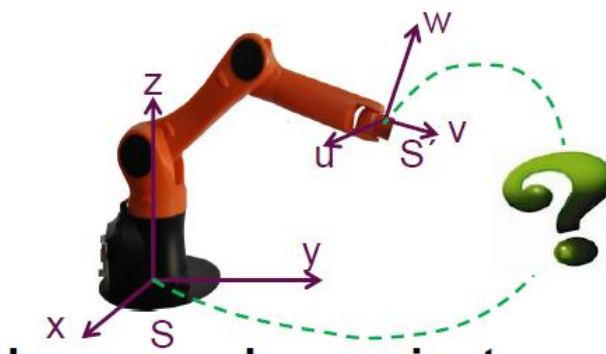


Figura 14. Interrelación sistemas coordenados.

Como hemos indicado anteriormente los sistemas de referencia definidos están jerarquizados de modo que conociendo la posición de un objeto con respecto a su sistema de referencia (herramienta de trabajo) es posible obtener la posición de ese mismo objeto con respecto al sistema de referencia de la base del Robot y se realiza mediante la utilización de una herramienta matemática conocida como matriz de transformación homogénea, la cual incluye las operaciones de traslación y orientación.

La matriz de transformación homogénea (MTH) es una matriz 4x4 que representa la transformada de un vector de coordenadas homogéneas de un sistema de coordenadas a otro.

$$T = \begin{bmatrix} \text{matriz de rotación} & \text{vector de posición} \\ f_{1 \times 3} & \text{escalado} \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 15. Matriz de transformación homogénea.

En robótica la submatriz f1x3 que representa una transformación de perspectiva, es nula, y la submatriz W1x1 que representa un escalado global es la unidad.

La matriz de rotación representa el giro de un sistema-j con respecto a un sistema-i y el vector posición representa la posición del origen del sistema-j en el sistema-i de coordenadas.

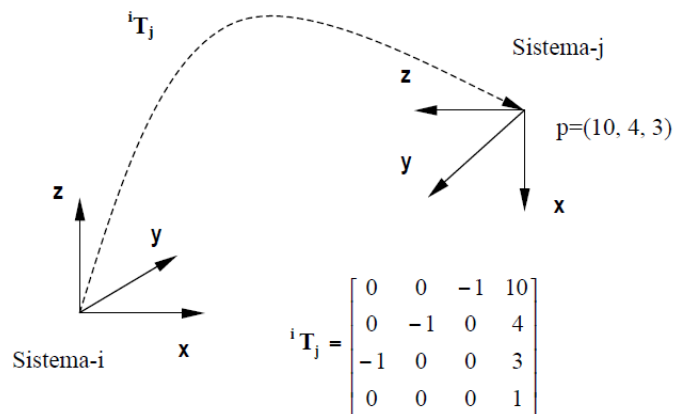


Figura 16. Interpretación geométrica de la matriz de transformación homogénea.

La rotación de un sistema-i con respecto a un sistema-j se puede componer de varias rotaciones. Debemos partir, por tanto, de conocer cómo se construye la matriz básica de rotación con un único giro y construir la matriz final a base de multiplicar matrices de rotación de un solo giro.



► **Matriz básica de rotación: sistema O'UVW girado únicamente un ángulo**

|   |   |   |
|---|---|---|
| $\alpha$ respecto al eje OX<br>$\left[ \begin{array}{ccc c} 1 & 0 & 0 & 0 \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$ | $\phi$ respecto al eje OY<br>$\left[ \begin{array}{ccc c} C\phi & 0 & S\phi & 0 \\ 0 & 1 & 0 & 0 \\ -S\phi & 0 & C\phi & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$ | $\theta$ respecto al eje OZ<br>$\left[ \begin{array}{ccc c} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$ |
|---|---|---|

► **Matriz básica de traslación: sistema únicamente trasladado**

$$\left[ \begin{array}{ccc|c} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Figura 17. Matriz básica de rotación.

Regla para componer ordenadamente las rotaciones:

- Inicialmente se suponen coincidentes ambos sistemas de coordenadas. **Matriz de rotación inicial=I 4x4.**
- Si el sistema O'UVW se obtiene mediante **rotaciones y traslaciones** definidas con **respecto al sistema fijo** OXYZ, la matriz homogénea que representa cada transformación deberá **pre multiplicar** sobre las matrices de transformación previas.
- Si el sistema O'UVW se obtiene mediante **rotaciones y traslaciones** definidas **con respecto al sistema móvil**, la matriz homogénea que representa cada transformación deberá **post multiplicar** sobre las matrices de transformación previas.

$$T = T_{xyz} | T_{uvw}$$

- 1) giro 90° en el eje X respecto al stma fijo
- 2) giro 90° en el eje Z respecto del stma móvil
- 3) traslado el vector (4,5,-3) respecto del stma fijo
- 4) giro 90° en el eje Y respecto del stma movil.

$${}^S T = T(p) T(x,90) T(z,90) T(y,90)$$

$${}^S T = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^S T = \begin{bmatrix} 0 & -1 & 0 & 4 \\ 1 & 0 & 0 & 5 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 18. Ejemplo cálculo matriz transformación.





### 3. APLICACIÓN PRÁCTICA. ROBOTSTUDIO.

Para la aplicación práctica utilizaremos el software RobotStudio versión 6.08.

En RobotStudio cuando iniciamos el trabajo creamos soluciones y el término SOLUCIÓN lo define como “el nombre colectivo para la carpeta que contiene la estructura para estaciones, bibliotecas y todos los elementos relacionados”. Dicha carpeta se crea cuando el usuario crea una nueva solución y contiene:

- **Estaciones:** las estaciones creadas como parte de la solución.
- **Sistemas:** controladores virtuales creados.
- **Bibliotecas:** definidas por el usuario o utilizadas en la estación.
- **Archivo de solución:** el archivo que abre la solución.

El flujo de trabajo para crear una solución **básica** con un robo en RobotStudio es el siguiente:

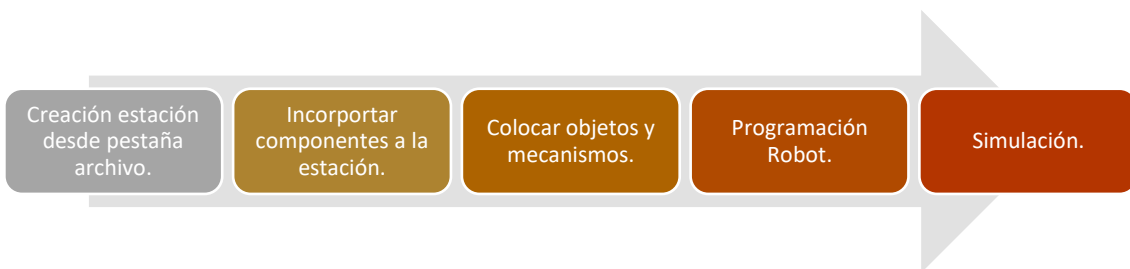


Figura 19. Diagrama flujo creación solución RobotStudio.

#### 3.1.1. Creación de una estación.

Al iniciar la aplicación RobotStudio, nos muestra la pestaña “Archivo” desde la cual nos ofrece tres opciones para crear una nueva solución:

- Solución con estación vacía.
- Solución con estación y controlador de robot.
- Estación vacía.

Seleccionaremos estación vacía y posteriormente añadiremos el robot y el sistema de controlador virtual.

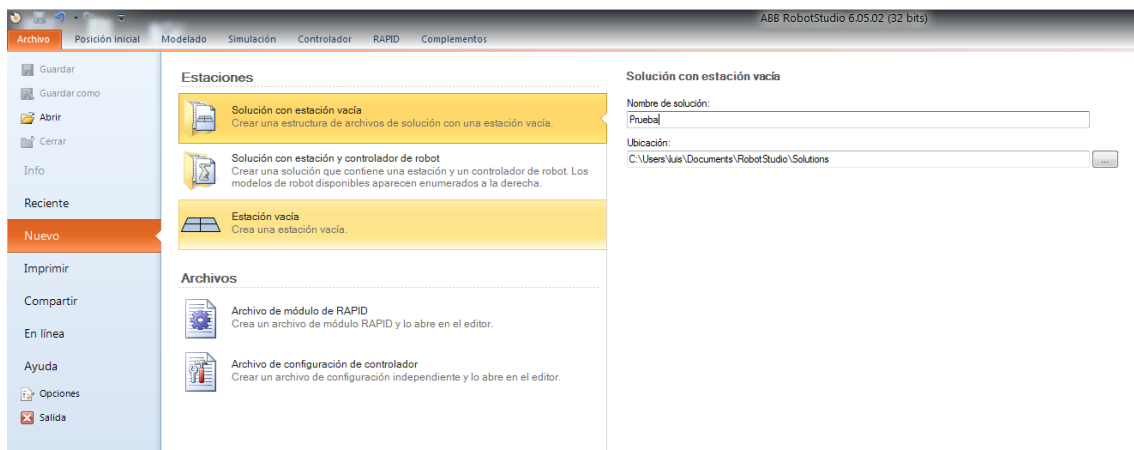


Figura 20. Solución con estación vacía.



### 3.1.2. Incorporar componentes a la estación.

Una vez creada la solución o estación vacía conforme al apartado anterior, incorporamos los componentes necesarios.

- **Importación de un modelo de robot.**

En la pestaña “Posición inicial” hacer clic en “Biblioteca ABB” y seleccionar el robot en el menú desplegable. Dependiendo del robot seleccionado, es posible que tenga diferentes versiones (capacidad y/o alcance) que podremos escoger en la pantalla que nos aparece al seleccionar un robot.

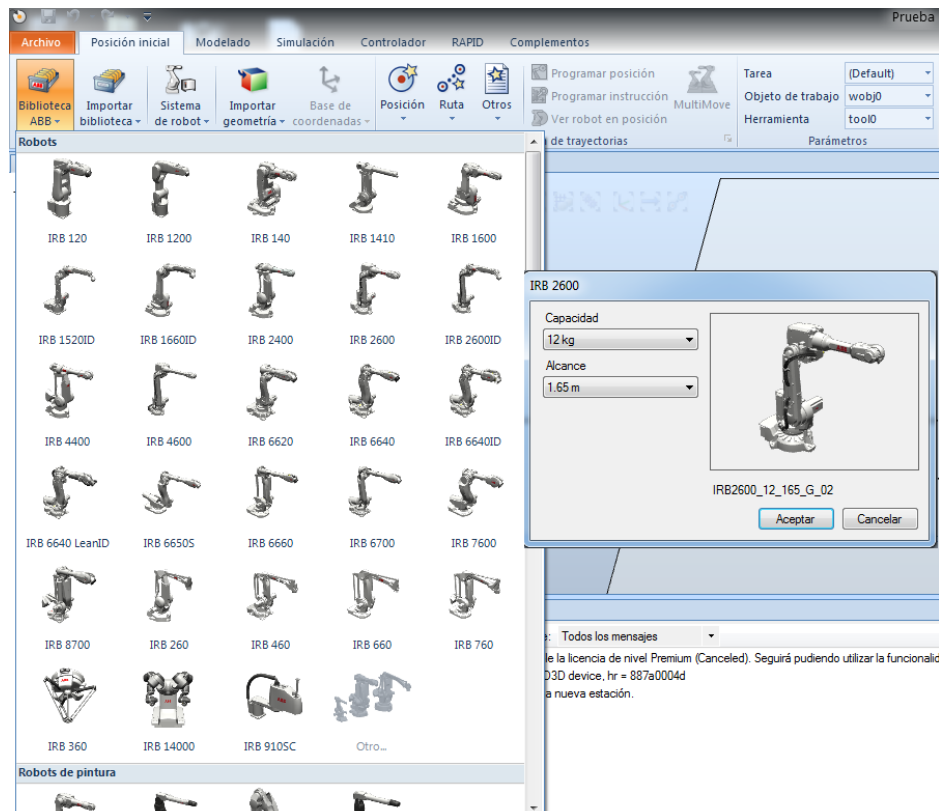


Figura 21. Selección del robot.

El robot insertado es solo un modelo gráfico, debe asociarse a un sistema (controlador virtual) para que pueda programarse y realizar movimientos, etc.

- **Importación de una herramienta.**

Una herramienta es un objeto que actúa sobre una pieza de trabajo, por ejemplo, una pistola de soldadura, una garra, un atornillador, etc.

Para poder obtener el posicionado correcto del robot con la herramienta es necesario definir correctamente los parámetros de la herramienta y entre ellos su TCP (ver punto 2.3.1).

RobotStudio dispone de algunas herramientas creadas que podemos usar. Desde la pestaña “Posición inicial” hacemos clic sobre “Importar biblioteca” y seleccionamos una de las herramientas disponibles.



Si es necesario podemos crear una herramienta a partir de un modelo CAD importado o dibujado en RobotStudio.

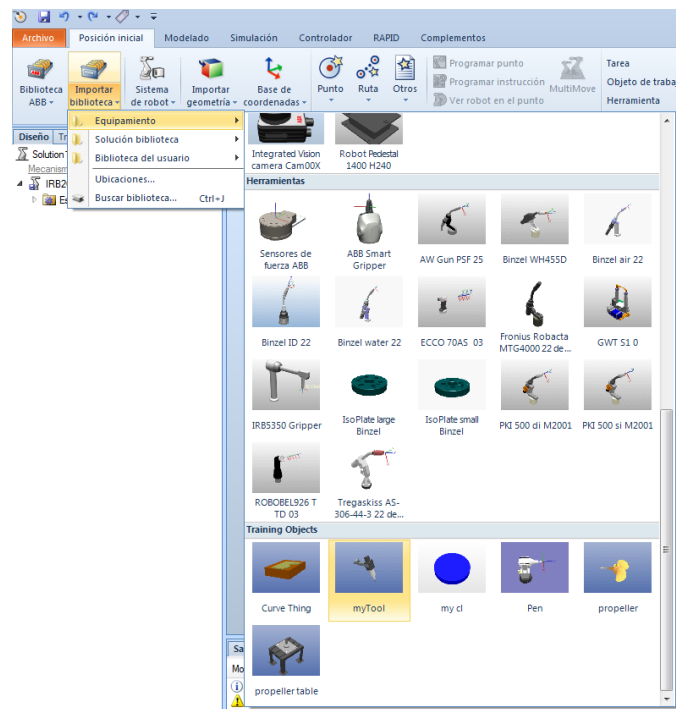


Figura 22. Importar herramienta.

Al importar la herramienta ésta se sitúa en el origen de coordenadas gráfico de la estación de RobotStudio, por lo que es necesario conectarla al robot posteriormente para poder usarla.

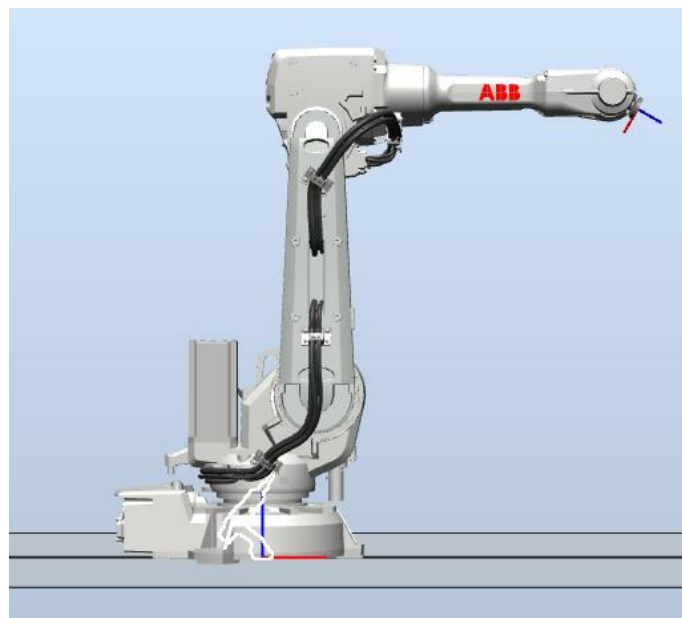


Figura 23. Herramienta insertada en la estación.

- **Importación de una biblioteca, geometría o elemento de equipo.**

En RobotStudio se pueden crear objetos para simular piezas o herramientas o bien importarlas una vez construidas en un software externo de diseño 3D.



Los formatos de archivo de CAD más habituales que admite RobotStudio son .3ds, .sat, .dxf, .dwg etc.. Para importar una geometría externa, desde la pestaña “Posición inicial”, hacemos clic en el icono “Importar Geometría”.

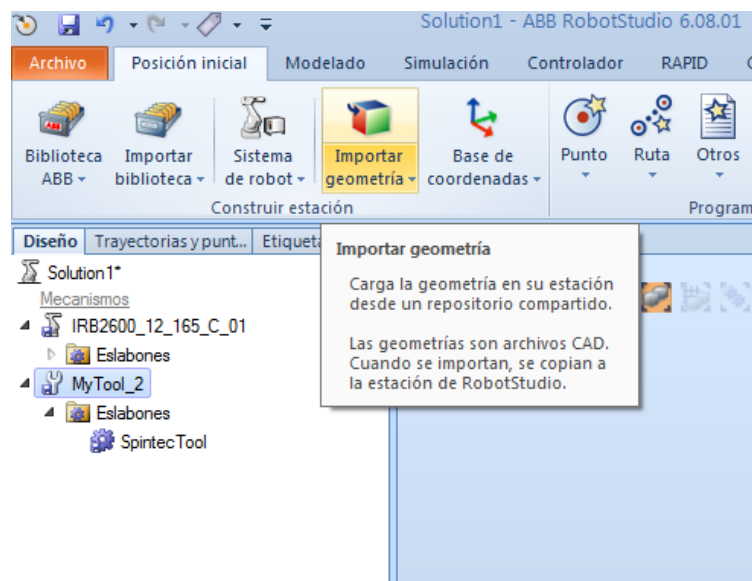


Figura 24. Importar Geometría.

Se puede crear una biblioteca a partir de una geometría creada en RobotStudio o importada, seleccionamos la geometría en el árbol de navegación “Diseño” haciendo clic sobre ella con el botón derecho del ratón, en la ventana flotante que aparece seleccionamos “Guardar como biblioteca”.

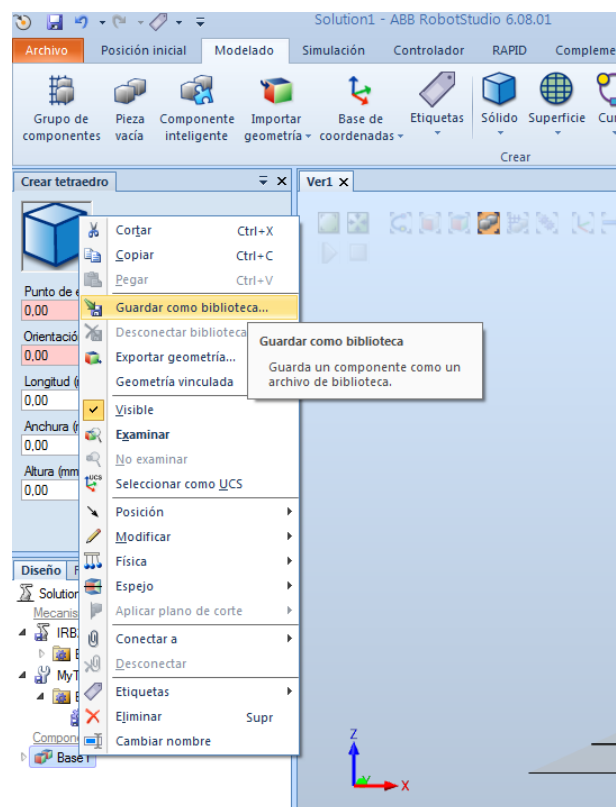


Figura 25. Guardar como biblioteca.



### 3.1.3. Creación de un sistema.

El sistema o controladora virtual se puede crear al inicio cuando se crea la estación o bien crearlo posteriormente. Es recomendable, antes de crear el sistema, incorporar a la estación todos los sistemas a controlar tales como robots, posicionadores, tracks o transportadores colocándoles en su posición de trabajo, y posteriormente crear el sistema que nos incluirá automáticamente todas las opciones de configuración necesarias.

Desde la pestaña “Posición inicial” el botón “Sistema Robot” nos permite crear el sistema controlador a través de tres opciones distintas.

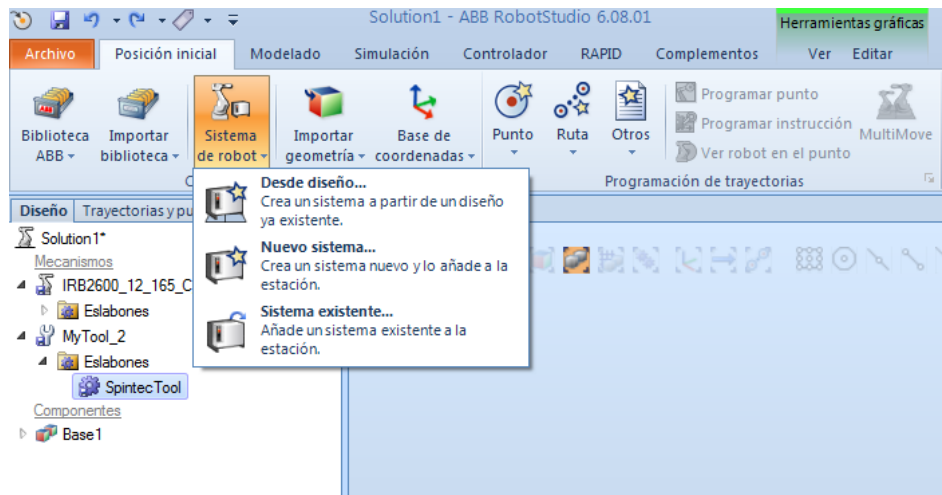


Figura 26. Creación sistema.

La opción “Desde diseño” es la más directa si ya hemos añadido a nuestra estación los sistemas a controlar.

Las siguientes pantallas nos permiten cambiar el nombre del sistema, seleccionar la ruta de guardado, seleccionar los robots, tracks, posicionadores o transportadores a incluir en el controlador y mediante el botón “opciones” incluir todas las opciones de configuración del sistema necesarias tales como idioma, tarjetas de comunicación, opciones de coordinación entre robots, tracks, etc. de acuerdo a las licencias que se tengan adquiridas.

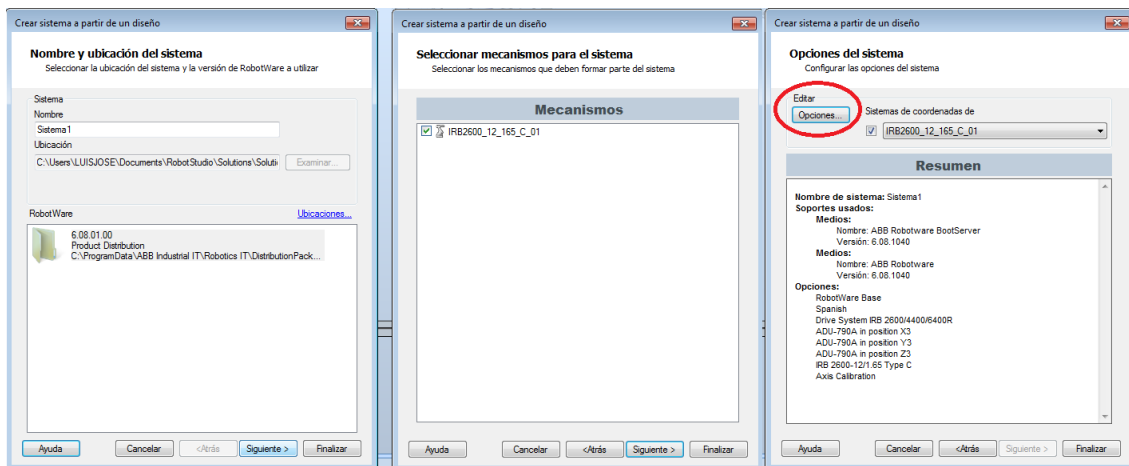


Figura 27. Pasos creación sistema.



## 4. PROGRAMACIÓN DE ROBOTS.

### 4.1.1. Pasos en la programación de un Robot.

Se puede usar tanto la unidad de programación como el RobotStudio para las tareas de programación. La unidad de programación es más adecuada para crear programas sencillos y la modificación de programas, por ejemplo, posiciones, mientras que el RobotStudio es preferible para programaciones complejas.

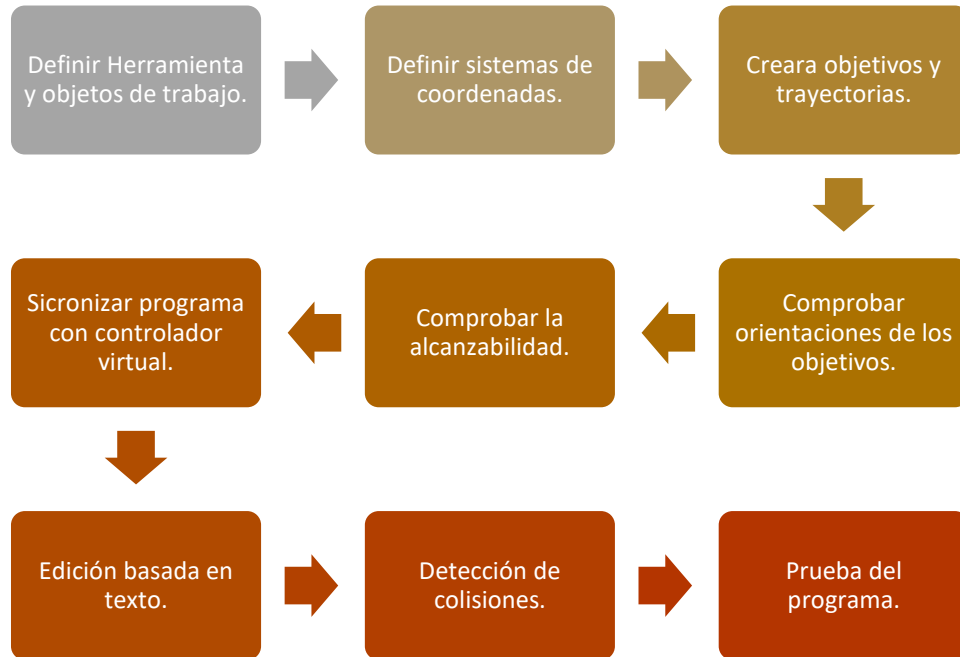


Figura 28. Secuencia programación Robots.

### 4.1.2. Creación de una herramienta.

Como hemos visto anteriormente (ver punto 3.1.2) podemos añadir a nuestra estación una herramienta directamente importándola desde la biblioteca y conectándola al TCP del robot, o podemos crear nuestra propia herramienta mediante la pestaña de modelado.

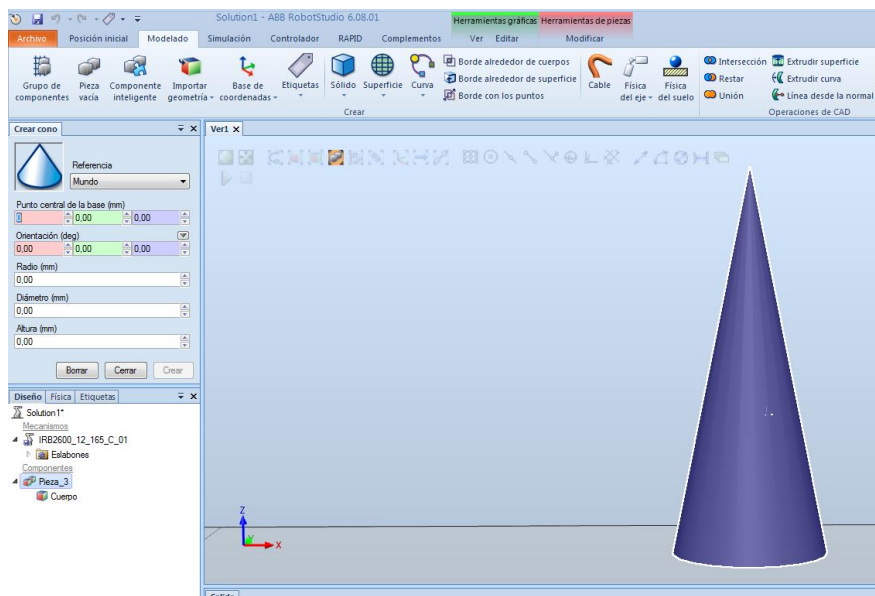


Figura 29. Modelado herramienta.



Simplemente elegiremos la forma que queremos dar a nuestra herramienta y rellenaremos las propiedades de punto central, radio y altura. Posteriormente en la misma pestaña de modelado seleccionaremos la opción “Crear herramienta”. Se abrirá una ventana flotante que nos irá guiando con los pasos necesarios para crear la herramienta. Al crear una herramienta se crea un dato del tipo “tooldata”, con valores predeterminados para el peso, centro de gravedad y otros, que deben modificarse para poder usar la herramienta.

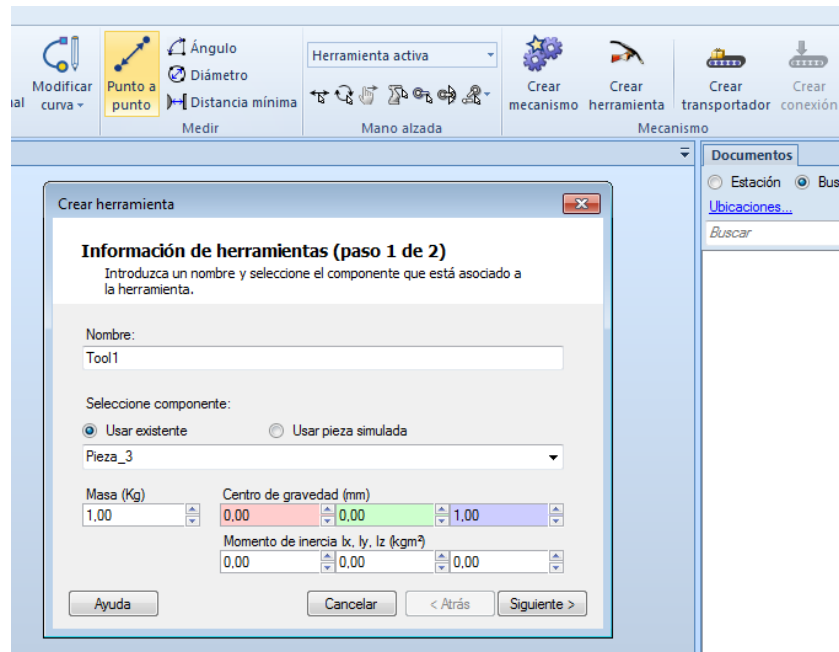


Figura 30. Crear herramienta.

En el paso final le damos un nombre al TCP, una posición y una orientación, pulsamos el botón alargado para añadir el TCP y pulsamos en terminado.

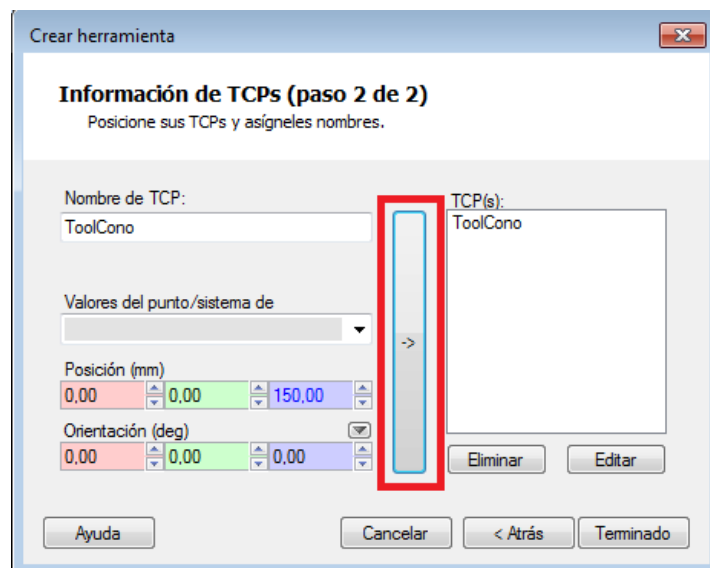


Figura 31. Crear herramienta. TCP





Finalmente, para añadir la herramienta al TCP del robot podemos hacerlo de dos maneras. Arrastrando la herramienta en el navegador de la izquierda hasta el icono del robot o haciendo clic con el botón derecho sobre la herramienta en el árbol de la izquierda, seleccionando la opción “Conectar a” y elegir la opción robot.

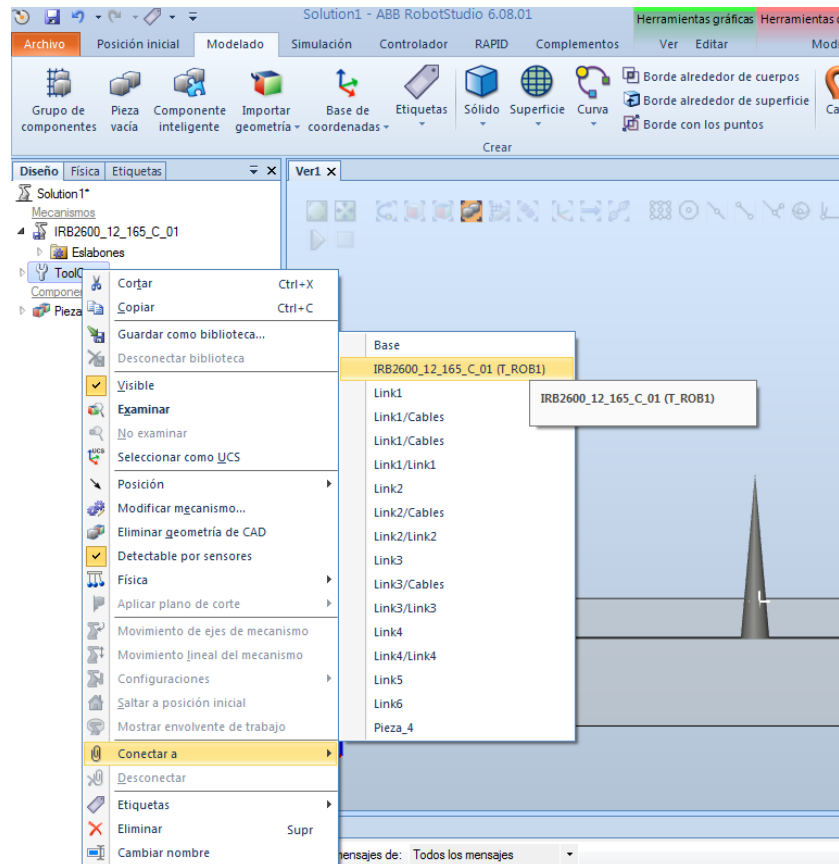


Figura 32. Asociar herramienta TCP robot.

#### 4.1.3. Sistema coordenadas herramienta.

##### 4.1.3.1. Sistema coordenadas RobotStudio.

En RobotStudio resulta fácil colocar el sistema de coordenadas de nuestra herramienta si tenemos el modelo mecánico. Para comprobarlo vamos a construir una herramienta compleja con dos TCP.

Durante el proceso de creación de la herramienta con la opción de modelado de Robotstudio es muy importante conocer dónde se encuentra el sistema de referencia de la pieza, si es local o coincide con la referencia world porque será el punto que se utilizará cuando lo queramos conectar al robot. Cuando creamos un objeto compuesto de varias piezas deben coincidir en su sistema de referencia para facilitar la colocación de unas piezas con respecto a otras para construir la herramienta final y luego conocer el punto que se usara cuando se conecte al robot.

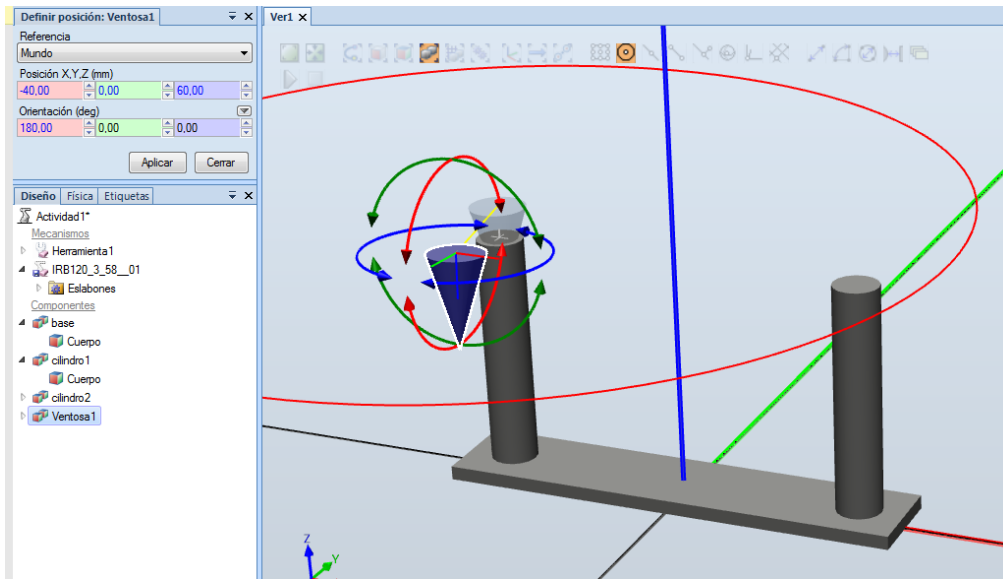


Figura 33. Modelado herramienta doble.

En la Figura 33. Modelado herramienta doble. vemos como el sistema de referencia de todas las piezas es el formado por los ejes rojo, azul y verde y luego cada una de ellas (el cono el centro de su base, el rectángulo una de las esquinas) tiene su punto para la ubicación o rotación. Finalmente, para crear el objeto del que crearemos la herramienta creamos un objeto vacío al que iremos añadiendo los cuerpos de cada uno de los objetos previamente creados.

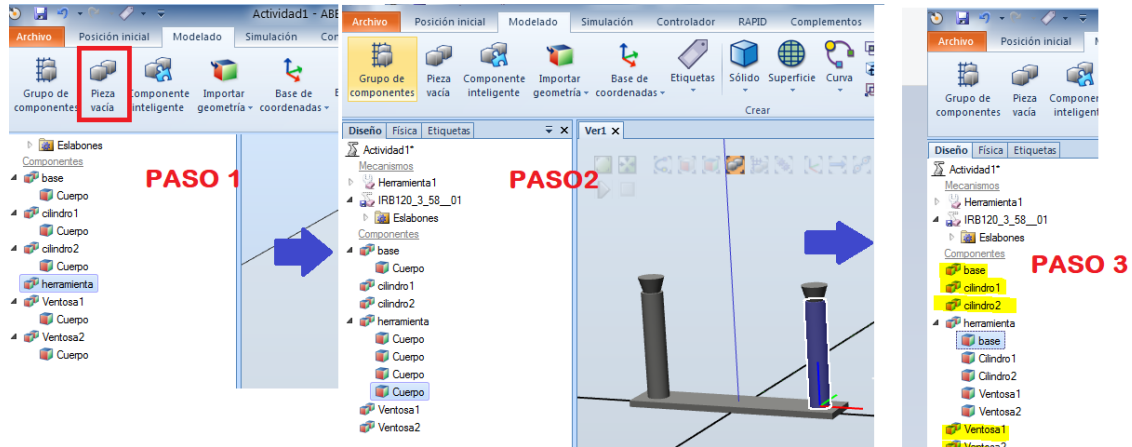


Figura 34. Agrupación objetos herramienta doble.

Finalmente podríamos eliminar los objetos que han quedado vacíos (señalados en amarillo en el paso 3) y pasar a crear la herramienta que en éste caso tendrá dos TCP, uno por ventosa situados en el centro exacto de la base.

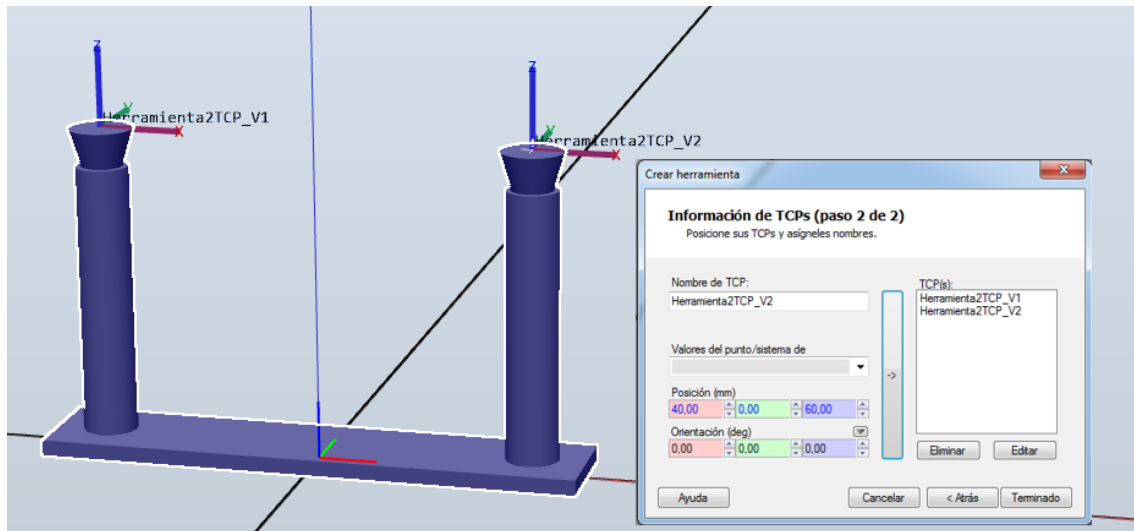


Figura 35. Herramienta con dos TCP.

Cuando unimos la herramienta al robot vemos que usa el punto medio de la pletina que habíamos llamado base para conectarlo a la brida 6 del robot, que coincide con el sistema de referencia que habíamos utilizado en la colocación de las piezas.

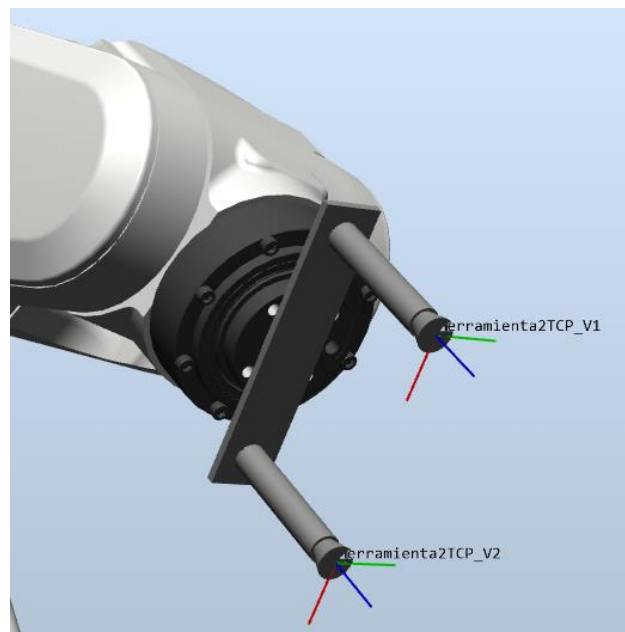


Figura 36. Herramienta con dos TCP en robot

#### 4.1.3.2. Sistema coordenadas Flexpendant.

En ocasiones no disponemos de la mecánica de la herramienta en 3D o resulta muy difícil de diseñar. En éstos casos, para establecer el sistema de coordenadas de la herramienta, se recurre a un punto de referencia fijo en la zona de trabajo del robot (suele ser una varilla terminada en punta) y en la herramienta del robot se coloca otra punta que se moverá a cuatro posiciones del robot con distintas orientaciones (cuatro por lo menos), lo más cerca posible al punto de referencia fijo anterior. Éstas posiciones se denominan punto 1,2,3 y 4.



Para definir la orientación completa de una herramienta, se deberá mover una posición en el eje Z deseado y una posición en el eje X deseado al punto de elongación, estos puntos de elongación deberán tener la misma orientación que el último punto almacenado. Estas posiciones se denominan puntos de elongación Z y X.

### Métodos disponibles.

Todos los métodos requieren que defina las coordenadas cartesianas del punto central de la herramienta. Estos métodos ofrecen distintas posibilidades de configuración y definición de la orientación.

| Método                           | Si desea...                              |
|----------------------------------|--|
| TCP (orientación predeterminada) | mantener la orientación actual           |
| TCP y Z                          | cambiar la orientación en el eje Z       |
| TCP y Z , X                      | cambiar la orientación en los ejes Z y X |

Este tipo de método se realiza utilizando la consola de programación y el software flexpendant.

|   | Acción  |
|---|---|
| 1 | En el menú ABB, seleccione Datos de programa.   |
| 2 | Seleccione tooldata y Mostrar datos para ver la lista de herramientas disponibles.  |
| 3 | Seleccione la herramienta que desee definir y seleccione Editar.  |
| 4 | En el menú, seleccione Definir y aparece la ventana de definición del sistema de coordenadas.   |
| 5 | Seleccione el método que prefiera en el menú emergente Método.<br>( ver Figura 37. Métodos definir TCP.)  |
| 6 | Seleccione el número de puntos de aproximación en el menú emergente Nº de puntos. Normalmente basta con 4 puntos. Si elige más puntos para obtener un resultado más exacto, debe poner el mismo cuidado al definir cada uno de ellos. |
| 7 | Definir la base de coordenadas de la herramienta tal como se explica en el siguiente apartado   |

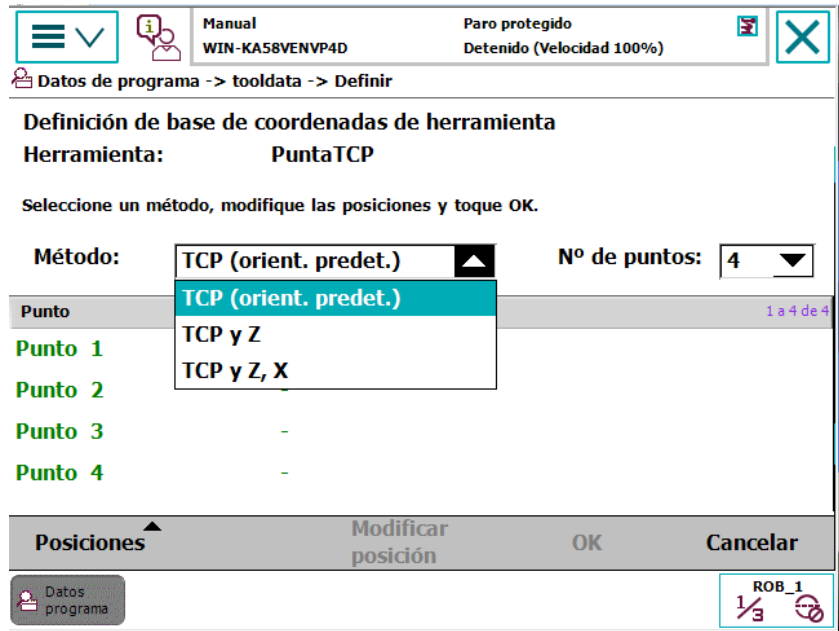


Figura 37. Métodos definir TCP.

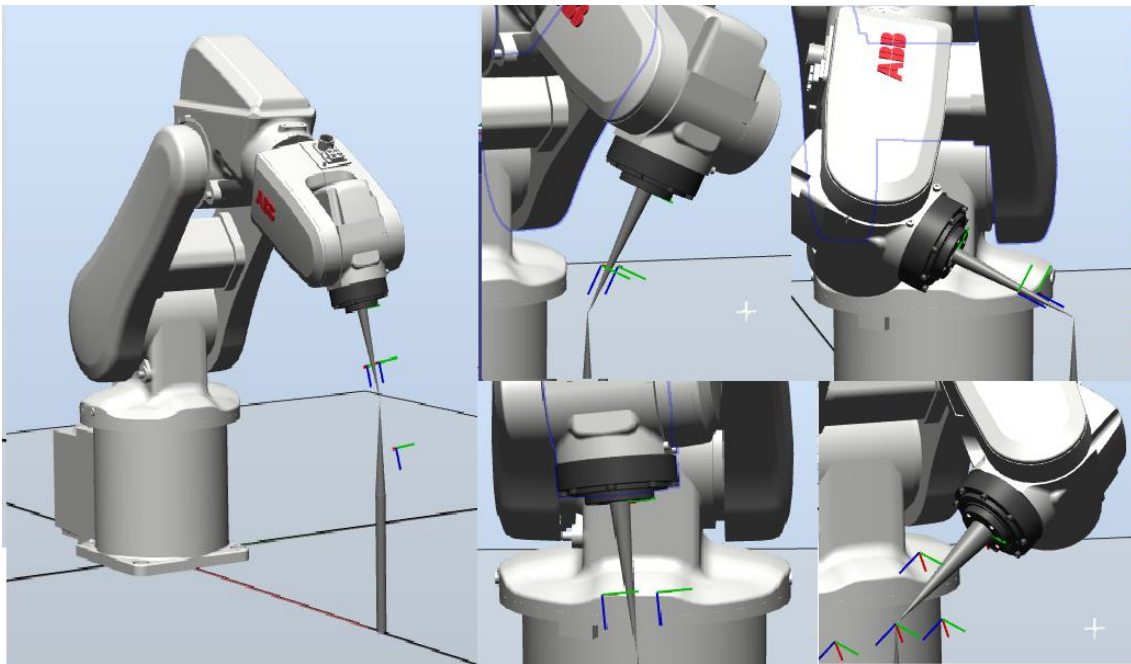


Figura 38. MétodoTCP (orient. predet.).

La ventana de diálogo Resultado de cálculo muestra el resultado calculado de la definición de la base de coordenadas de la herramienta. Es necesario confirmar que acepta el resultado antes de que pueda tener efecto en el controlador. La alternativa es repetir la definición de la base de coordenadas para conseguir un mejor resultado. El resultado Error medio es la distancia media de los puntos de aproximación con respecto al TCP (punto central de la herramienta) calculado. Error máximo Es el error máximo existente entre todos los puntos de aproximación.



Resulta difícil afirmar qué resultado es aceptable. Depende de la herramienta, el tipo de robot, etc. que esté utilizando. Normalmente, un error medio de algunas décimas de milímetro es un buen resultado. Si el posicionamiento ha sido realizado con una exactitud razonable, el resultado será correcto.

#### 4.1.4. Creación objeto trabajo.

Los objetos de trabajo son los sistemas de coordenadas dónde se sitúan los puntos objetivos del robot. Un objeto de trabajo está formado por dos sistemas de coordenadas denominados “sistema de coordenadas usuario” y “sistema de coordenadas objeto”, siendo el segundo descendiente en la jerarquía del primero, de modo que si al crear el objeto sus valores son cero coincidirá exactamente con el sistema de coordenadas usuario. Normalmente se define únicamente el sistema de coordenadas de usuario. Para crear el objeto de trabajo nos situamos en la ventana “Trayectorias y puntos”.

En ésta pestaña podemos ver que tenemos tareas y sistemas. Podemos tener varios sistemas y tareas y en cada uno de ellos herramientas y objetos cuyo ámbito de aplicación dependerá de dónde están creados. Nosotros crearemos el objeto de trabajo en el único sistema que tenemos que le hace utilizable en cualquier tarea que posteriormente podamos crear.

Por defecto tenemos un objeto de trabajo 0 (wobjt0) cuyo sistema de coordenadas coincide con el sistema de coordenadas del sistema o también denominado mundo (RS-WCS) y con el de la base del robot (BF).

Los objetos de trabajo se utilizan para crear sistemas de referencias que podemos asociar a un objeto mecánico o pieza y sobre el que queremos situar nuestros puntos objetivos en una trayectoria. Con ello conseguimos que si el objeto se desplaza todos los puntos creados en ese objeto de trabajo se desplazarán y no tendremos que reprogramarlos.

#### Creación de un objeto de trabajo.

- Seleccionamos “Crear objeto de trabajo” en el menú desplegable “Otros”, en la pestaña “Posición Inicial”.

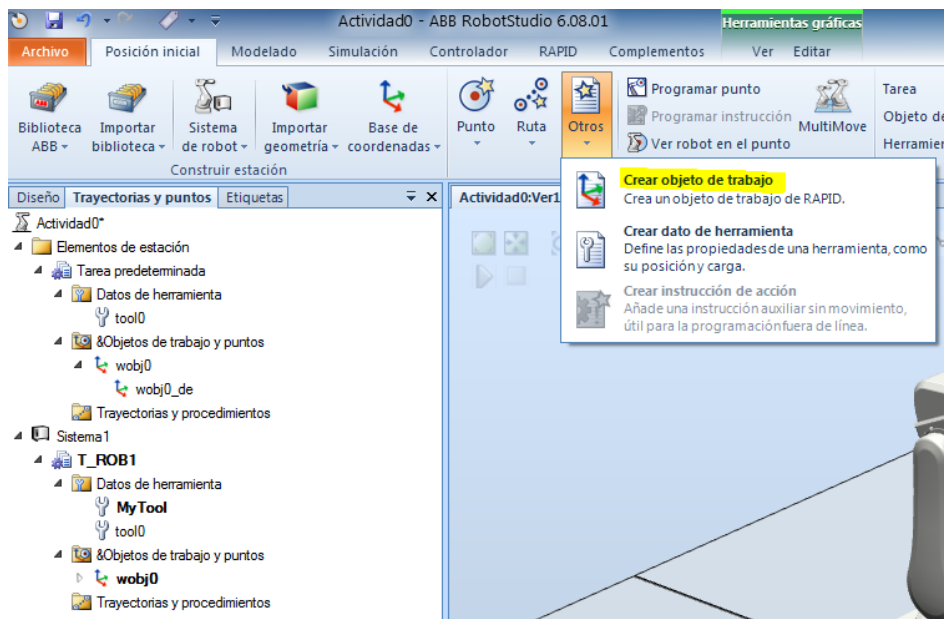


Figura 39. Crear "Objeto Trabajo"

- Aparece la ventana de creación del “objeto de trabajo”, que nos permite definir los sistemas de coordenadas objeto y usuario a través de su posición y rotación o mediante la creación de un sistema de coordenadas utilizando la opción “posición” o “Tres puntos”.

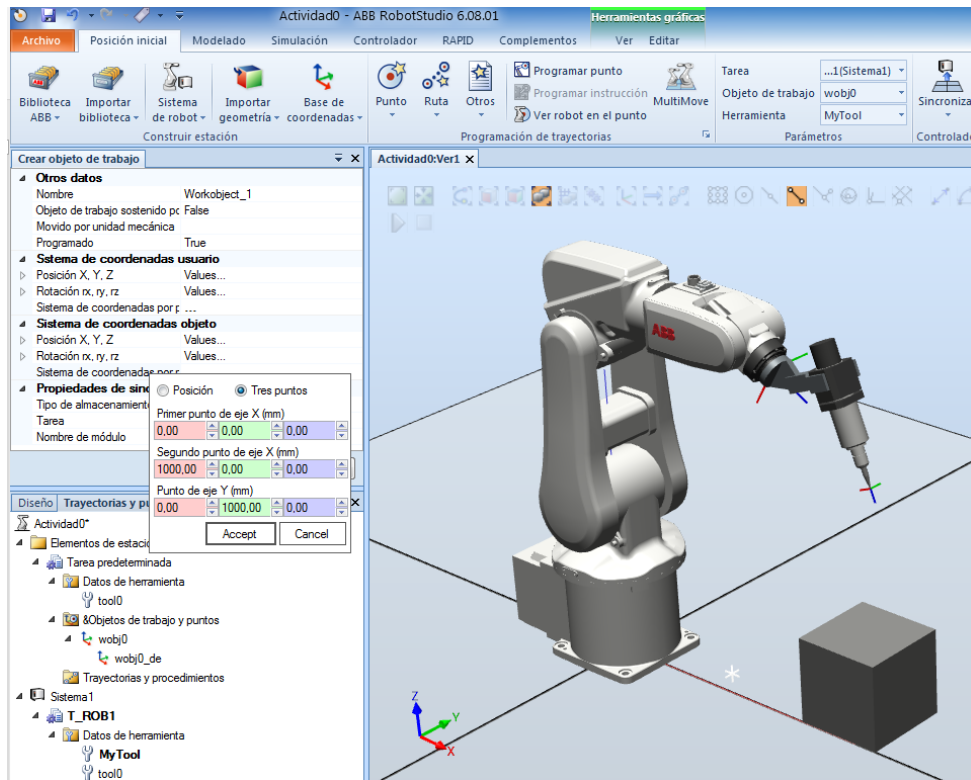


Figura 40. Menú creación objeto trabajo.

Si nuestro objeto de trabajo va a estar asociado a un cubo que hemos creado previamente, lo más sencillo es seleccionar los tres puntos sobre los vértices del cubo en el orden que deseamos. Debemos tener en cuenta que el orden en el que escojamos los puntos definirán la orientación del sistema de coordenadas del objeto, lo cual será importante más tarde cuando establezcamos nuestros puntos objetivos.

- Posteriormente si queremos modificar nuestro objeto de trabajo podemos hacerlo pulsando sobre él con el botón secundario del ratón o a través de la opción “Modificar” de herramientas de objetos de trabajo que nos aparece en la parte superior.

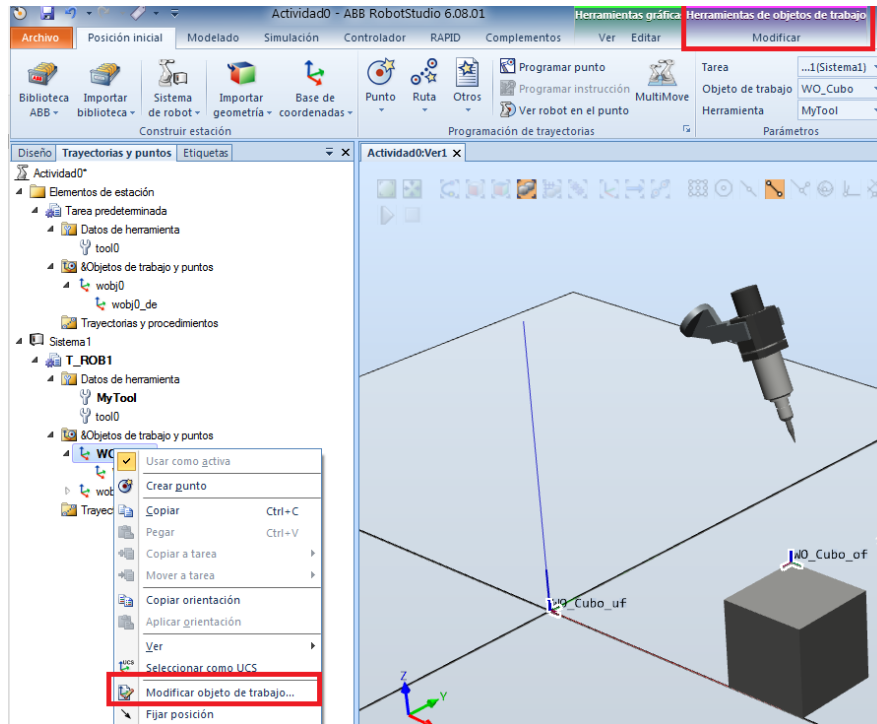


Figura 41. Modificar objeto de trabajo.

- Una vez tenemos nuestro objeto de trabajo lo podemos asociar al objeto mecánico “cubo” que habíamos creado previamente. De ésta manera cuando desplazemos el cubo el sistema de referencias de nuestro objeto de trabajo se desplazará con él. Pulsamos botón secundario del ratón sobre el objeto de trabajo y tendremos la opción “conectar a”, nos preguntará “actualizar posición” y responderemos que, no.

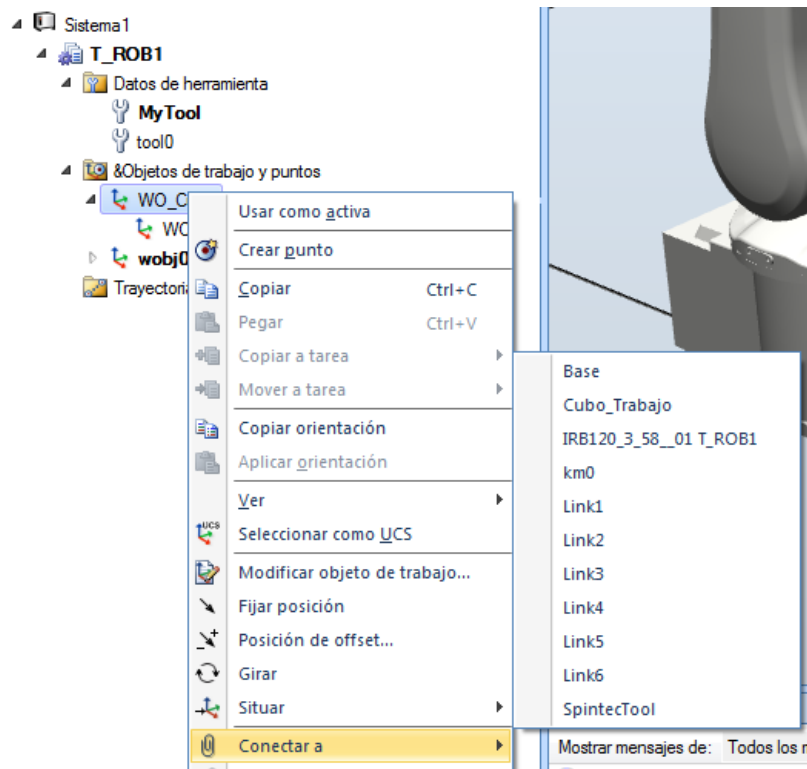


Figura 42. Conectar objeto de trabajo.





#### 4.1.5. Sistema coordenadas objeto de trabajo.

Como hemos visto durante la creación de un objeto de trabajo, se puede definir fácilmente su sistema de coordenadas asociado, pero normalmente en una instalación en campo la estructura mecánica o elementos de la estación presentan desviaciones con respecto a los elementos que hemos utilizado o importado en Robotstudio. Es fácil imaginar, que el suelo dónde se asienta la estación y el robot presenta pequeñas desviaciones o simplemente la colocación de los diferentes elementos mecánicos dentro de la estación no es perfecta.

En éstos casos se procede a calcular el sistema de referencia asociado a un objeto de trabajo mediante la unidad de programación y el software flexpendant. Cuando se crea un objeto de trabajo se genera un nuevo dato en la aplicación de tipo “wobjdata” cuyo sistema de coordenadas es idéntico al sistema de coordenadas Wobj0.

Debemos seleccionar un método para la creación y orientación del sistema de coordenadas del objeto de trabajo creado.

|   | Acción.  |
|---|--|
| 1 | En el menú principal, seleccione Datos de programa. (ver Figura 43)                      |
| 2 | Seleccione Wobjdata y Mostrar datos para ver la lista de objetos de trabajo disponibles. |
| 3 | Seleccione el objeto de trabajo que desee definir y seleccione menú Editar.              |
| 4 | En el menú, seleccione Definir. (ver Figura 44)  |
| 5 | Seleccione un método en el menú Método de usuario o el menú Método de objeto.            |

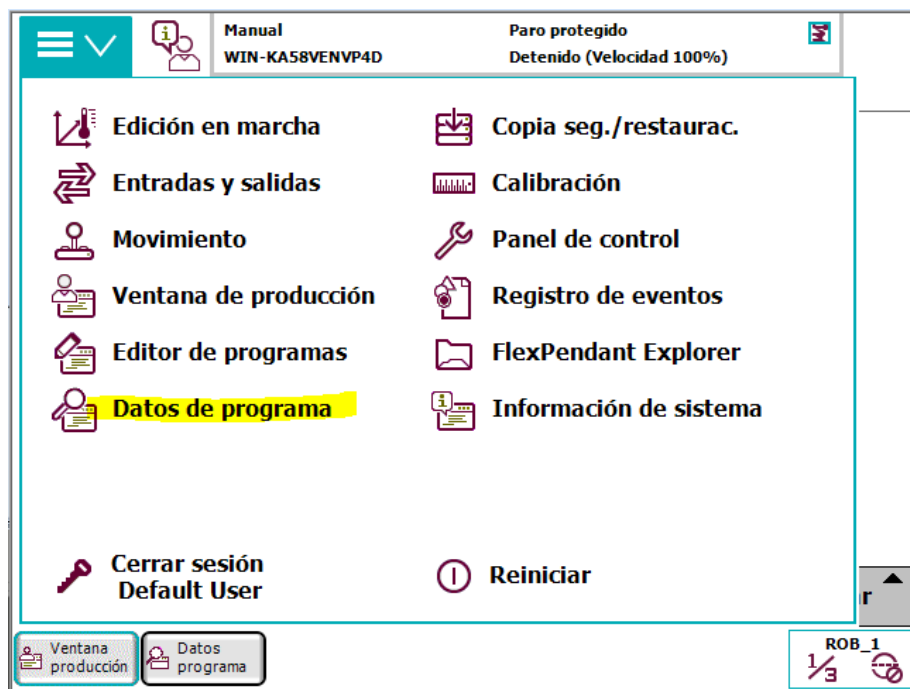


Figura 43. Menú selección Flexpendant.

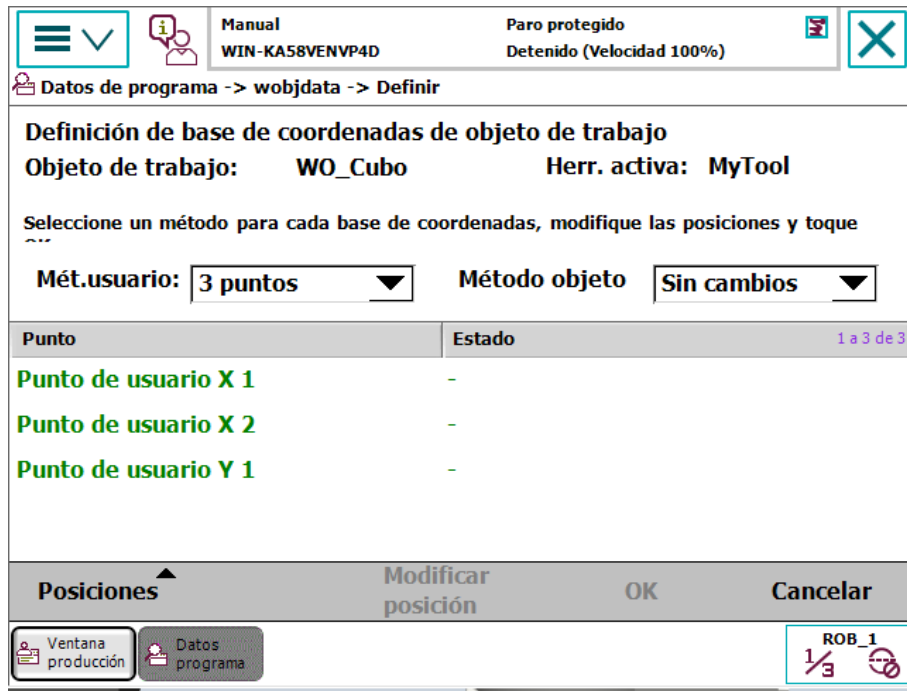


Figura 44. Menú definir objeto trabajo flexpendant.

|   | Acción   | Información   |
|---|--|---|
| 1 | Seleccione la base de coordenadas que desee definir: Método de usuario o Método de objeto y seleccione la opción 3 puntos. |   |
| 2 | Presione el dispositivo de habilitación y mueva el robot hasta el primer punto X1.   | Es recomendable que la distancia entre X1, X2 y Y1 sea grande lo que permite obtener una definición más exacta. |
| 3 | Seleccione el punto en la lista.   |   |
| 4 | Seleccione Modificar posición para definir el punto.   |   |
| 5 | Repita los pasos del 2 al 4 con los puntos, X2 y Y1.   |   |

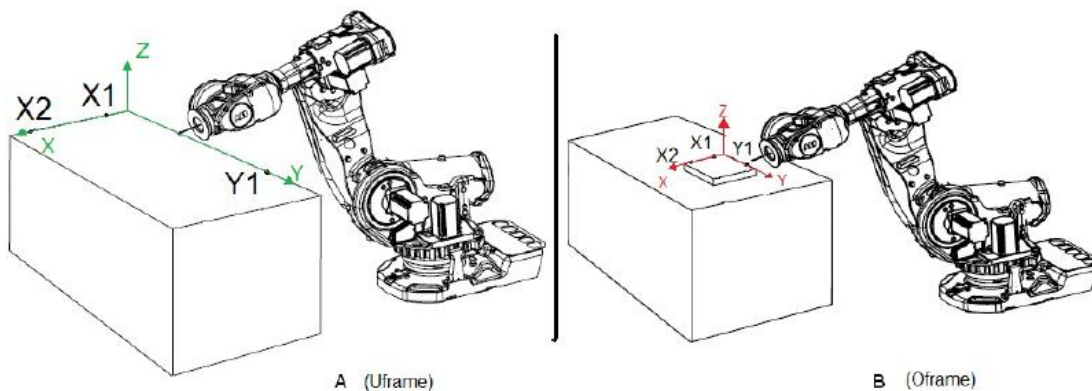


Figura 45. Definir base coordenadas usuario y objeto.



Para realizar éste método se suele emplear una punta que se acopla a la herramienta de trabajo que va utilizar el robot y otras tres puntas que van acopladas al objeto de trabajo de modo que la aproximación a los puntos X1,X2 e Y1, sea más precisa, con lo que conseguiremos que nuestro sistema de coordenadas tenga una orientación lo más exacta posible al plano asociado.

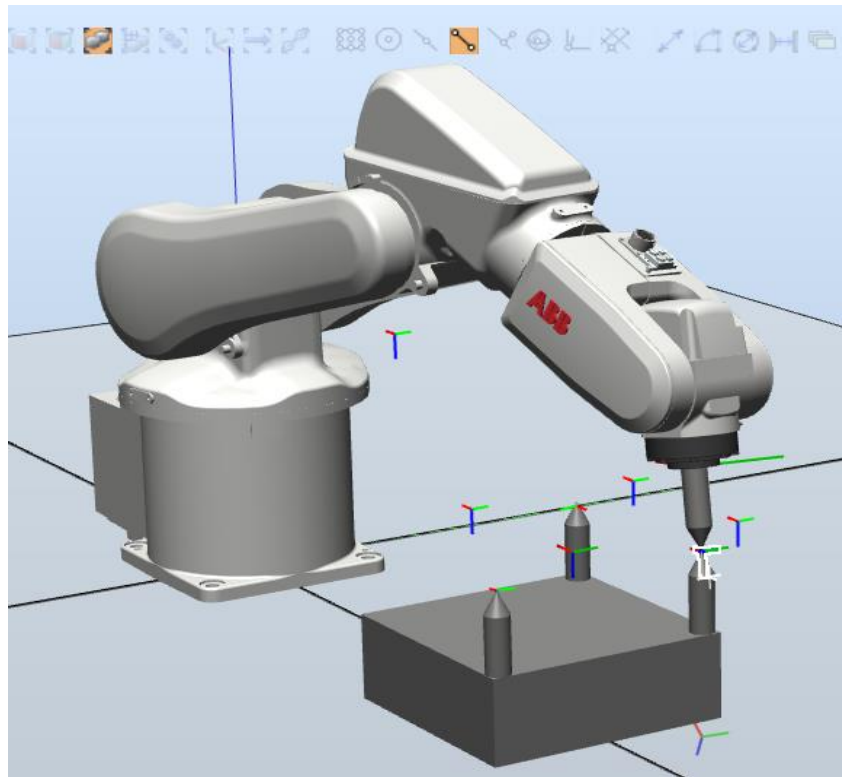


Figura 46. Método tres puntos sistema coordenadas objeto

#### 4.1.6. Creación de objetivos o puntos de trabajo.

Los objetivos o puntos de trabajo son los puntos que el robot utilizará como destino cuando generemos una trayectoria. Se crean desde la pestaña de “trayectorias y puntos” de dos maneras diferentes desde la pestaña superior de “programación de trayectorias”.

- Utilizando la posición actual del robot y el botón “programar punto”.
- Con el botón crear punto.

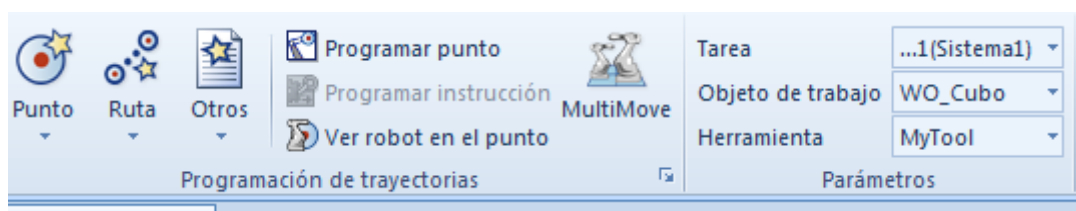


Figura 47. Crear objetivo o punto de trabajo.

En ambos casos debemos tener en cuenta que valores están seleccionados en la pestaña parámetros pues definirán a que objeto de trabajo se asocian esos puntos y la herramienta utilizada. Ver que dependiendo de éstos valores en el árbol de la pestaña “trayectoria y puntos” aparecen resaltados el objeto y la herramienta correspondiente.

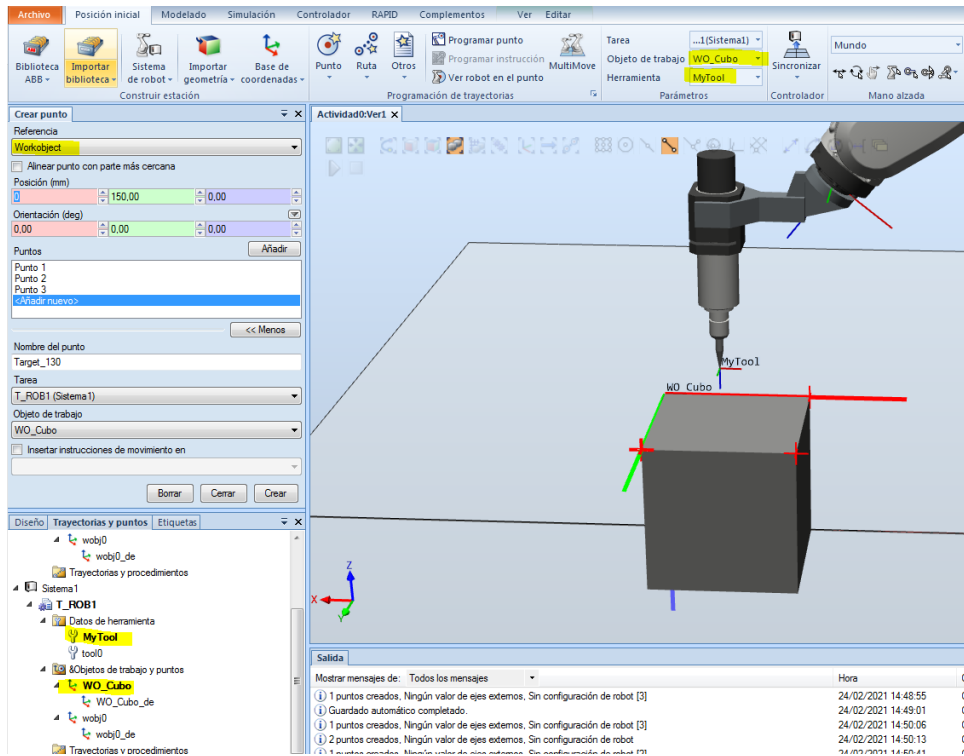


Figura 48. Crear objeto o punto trabajo. Parámetros.

#### 4.1.6.1. Ajuste sistema coordenadas objeto-herramienta.

Una vez creados los puntos, podemos ver cómo queda la herramienta cuando el robot se posiciona sobre esos puntos.

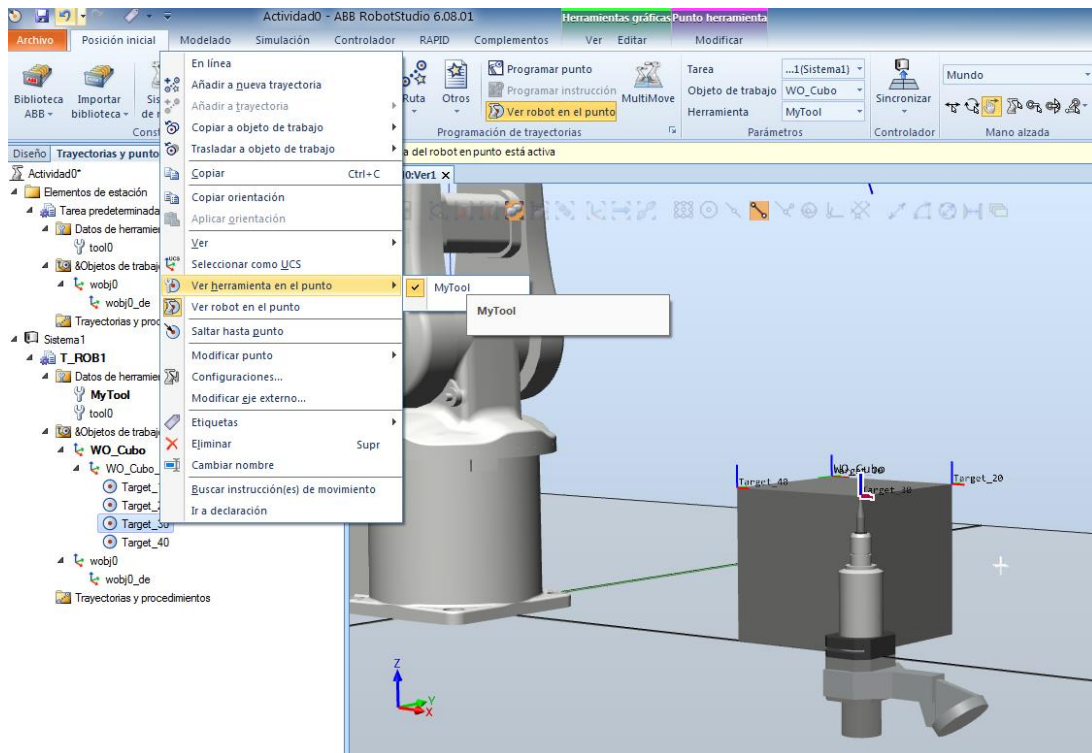


Figura 49. Ver herramienta en punto.

Como vemos la herramienta queda orientada en una posición a la que el robot nunca podrá llegar. Para solucionar éste problema podemos reorientar la herramienta en ese punto o modificar el sistema de coordenadas del objeto de trabajo de modo que la herramienta cuando llegue a cualquiera de los puntos que se sitúen sobre el objeto de trabajo quedará bien posicionada. Para ello modificamos el objeto de trabajo de modo que al seleccionar los tres puntos los sentidos de los ejes coincidan con el de la herramienta.

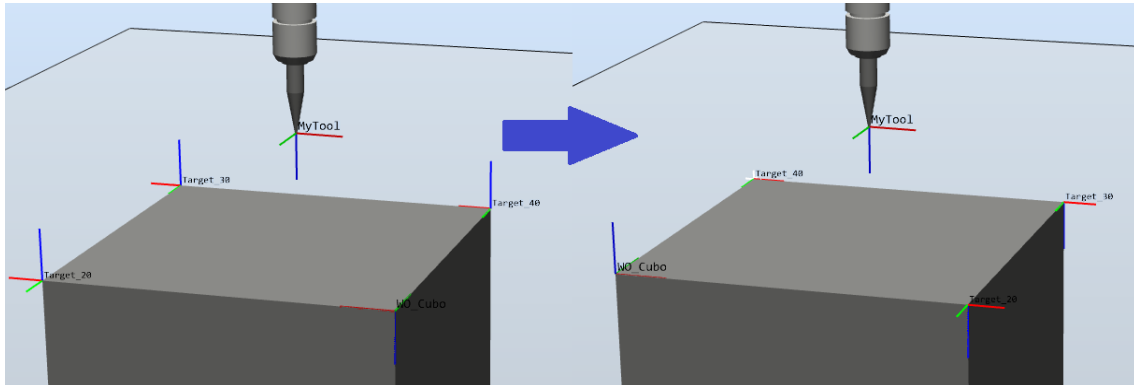


Figura 50. Orientación puntos con objeto trabajo.

#### 4.1.7. Trayectorias.

Las trayectorias son secuencias de objetivos con instrucciones de movimiento que debe seguir el robot. Las trayectorias se pueden crear manualmente o mediante las herramientas de que dispone RobotStudio.

- **Trayectoria vacía:** desde la pestaña “Posición inicial” podemos, en el menú “Programación de trayectorias”, pulsar sobre “Ruta” y a continuación sobre “Trayectoria vacía” generando una carpeta dentro de la carpeta “Trayectoria y movimientos”.

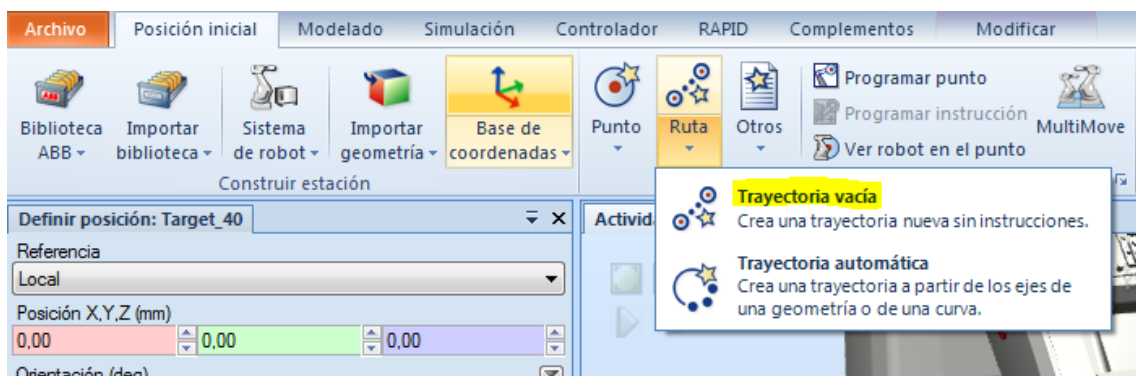


Figura 51. Creación trayectoria vacía.

A ésta trayectoria le faltarán los puntos objetivos que podemos ir arrastrando hasta su carpeta si ya les hemos creado anteriormente.

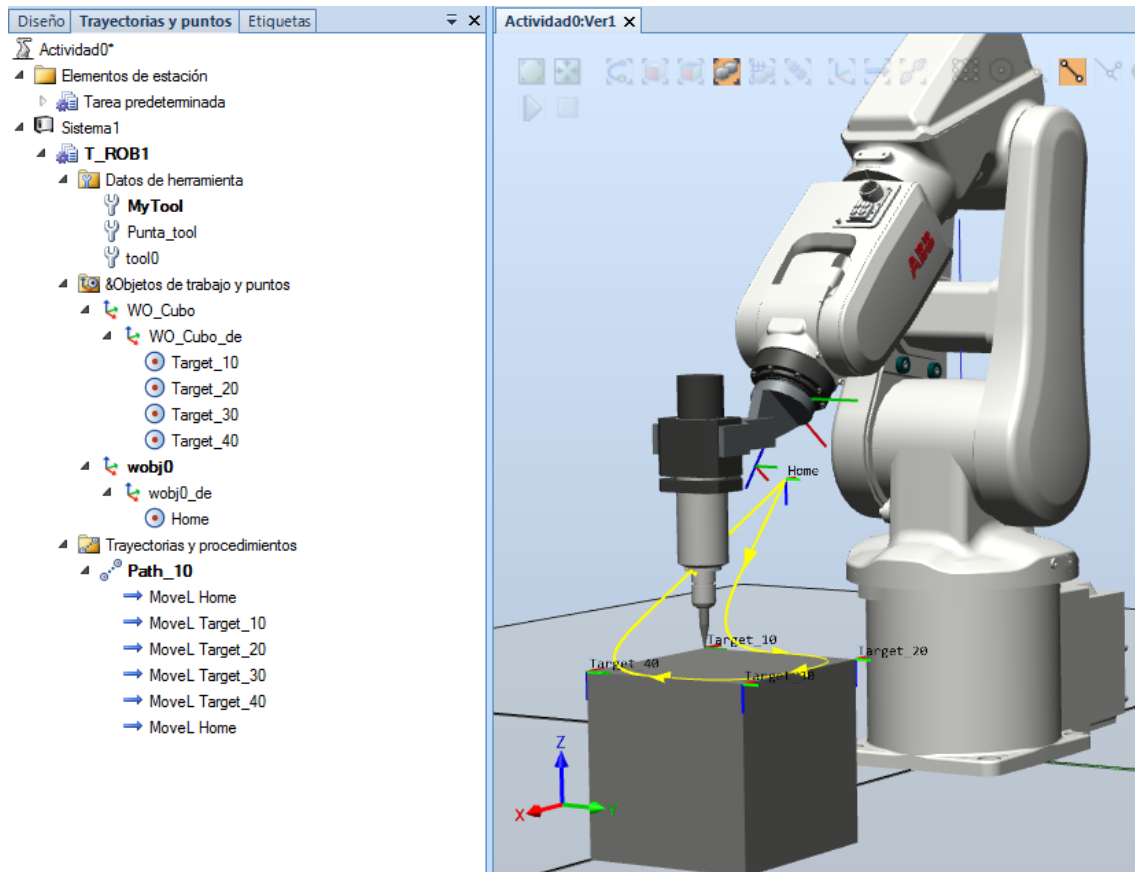


Figura 52. Trayectoria con puntos.

- **Trayectoria a partir de borde/curva:** RobotStudio dispone de herramientas para crear trayectorias automáticamente (objetivos e instrucciones de movimiento) siguiendo curvas o contornos de la pieza de trabajo.
- **Configuración de ejes para trayectorias:** Ya hemos comentado anteriormente lo importante que es que el robot tenga la herramienta bien orientada en cada uno de los puntos de trabajo de una trayectoria y que las trayectorias estén dentro del área de trabajo del robot. Al realizar una trayectoria es posible que el robot pase cerca o por uno de sus puntos singulares<sup>1</sup> y no pueda realizar el movimiento.

Cuando un punto no es alcanzable en la configuración actual pero sí en otra configuración RobotStudio muestra un icono de advertencia amarillo, en ese caso, hacemos clic con el botón derecho del ratón sobre el path y seleccionamos “Configuración automática” << “Todas las instrucciones del movimiento”. En el caso que no sea alcanzable en ninguna configuración muestra un icono rojo y en ese caso debemos optar por cambiar la orientación de la herramienta, posición de la pieza o del robot para lograr alcanzar el punto.

<sup>1</sup> Punto singular, consiste en un alineamiento de dos más ejes, que resta grados de libertad al robot ocasionando un comportamiento impredecible tal que pequeños movimientos del extremo final del robot pueden suponer velocidades articulares de magnitud descontrolada.

Para modificar la orientación de la herramienta en el punto podemos usar el menú contextual que aparece al hacer clic con el botón secundario del ratón sobre el punto y seleccionando fijar posición podremos girar la posición de la herramienta en el punto.

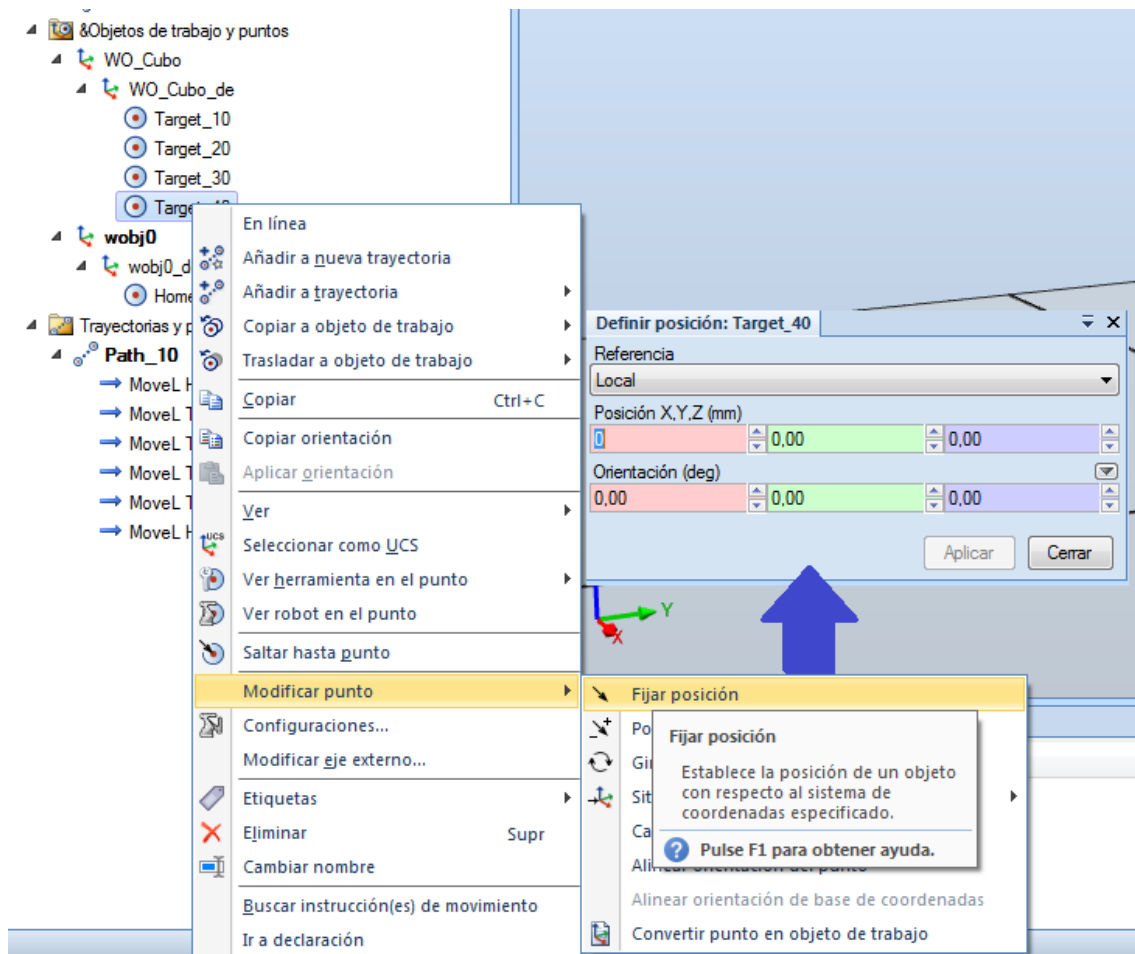


Figura 53. Menú modificar punto.

Modificar la posición de la herramienta en el punto mediante la opción “Mano alzada”. Primero visualizamos la herramienta en el punto y después seleccionaremos “Mover” o “Girar” del menú “Mano alzada” seleccionando el sistema de referencia adecuado del mismo menú.

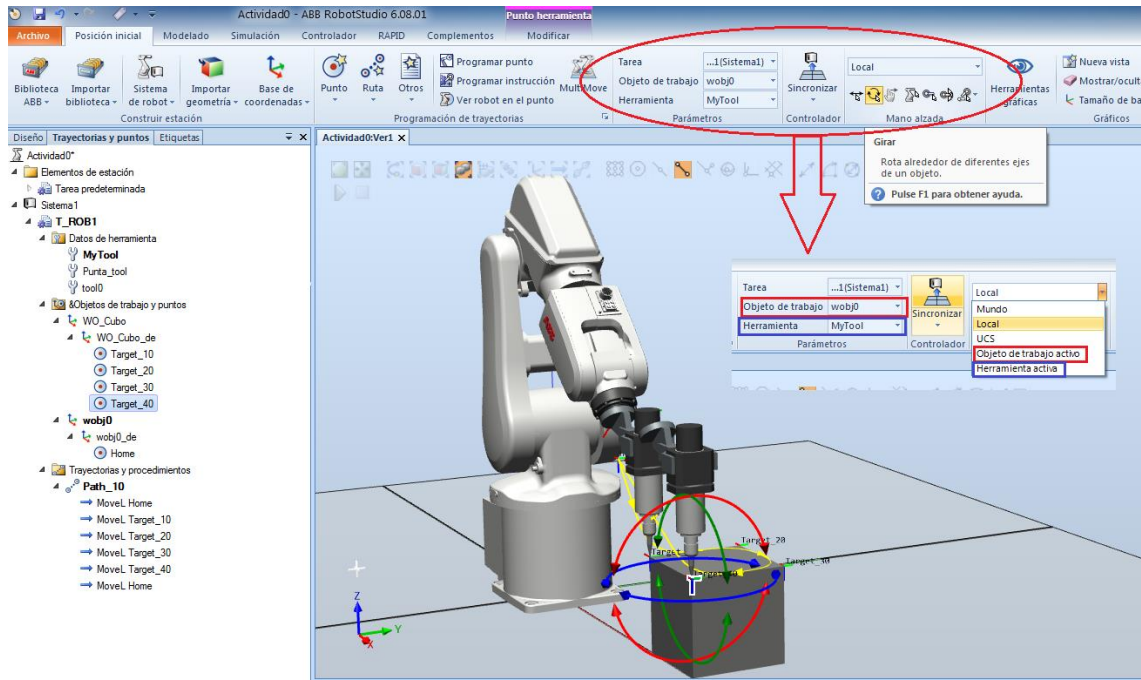


Figura 54. Modificar posición punto. Mano alzada.

Una vez conseguida una orientación adecuada de la herramienta en un punto podemos copiar esa orientación para el resto de puntos simplemente con la opción “copiar orientación” y “aplicar orientación” del menú que aparece al pulsar con el botón secundario del ratón sobre el punto.

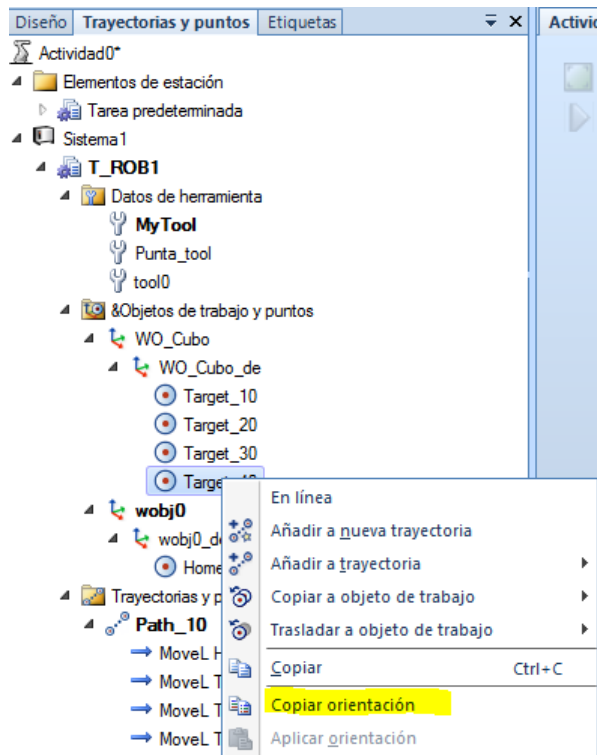


Figura 55. Copiar orientación de un punto.





#### 4.1.8. Tipos de datos.

La creación de elementos como herramientas, objetos de trabajo y puntos de trabajo genera una serie de variables en la controladora que posteriormente si trabajamos en el entorno RAPID debemos modificar sus valores. La comprensión de las propiedades de éstas variables ayuda a entender el funcionamiento de los diferentes objetos en RobotStudio.

##### 4.1.8.1. Datos de configuración del robot. Confdata.

La variable “Confdata” sirve para definir las configuraciones de los ejes del robot.

Todas las posiciones del robot están definidas y almacenadas utilizando coordenadas rectangulares. Al calcular las posiciones de los ejes correspondientes, pueden existir muchas veces dos o más soluciones posibles. Ello significa que el robot es capaz de alcanzar la misma posición, o sea, la herramienta es capaz de llegar a la misma posición y con la misma orientación, mediante diferentes posiciones y configuraciones de los ejes del robot.

Para determinar sin ambigüedad cual es la posición y orientación correcta, se deberá definir la configuración del robot utilizando los valores de cuatro ejes. Estos valores definen el cuadrante de la vuelta en donde están situados estos cuatro ejes del robot. Los cuadrantes están numerados según el orden 0, 1, 2, etc. (también pueden ser negativos). El número del cuadrante está asociado al ángulo del eje. Para cada eje, el cuadrante 0 es el primer cuarto, de 0 a 90°, en una dirección positiva partiendo desde la posición 0; el cuadrante 1 es el cuarto siguiente, de 90 a 180°, etc. El cuadrante -1 es el cuarto 0° a (-90°), etc.

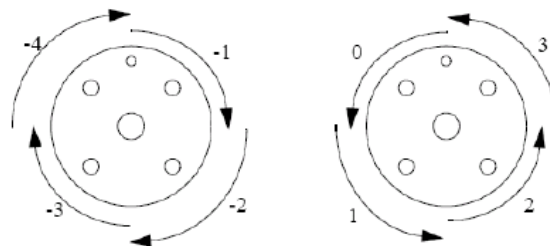


Figura 56. Cuadrantes configuración eje 6.

Dentro de la configuración de la mayor parte de los robots existen tres singularidades dentro de su área de trabajo.

Cf1 se usa para el eje 1. Cf4 se usa para el eje 4. Cf6 se usa para el eje 6. Cfx se usa para seleccionar uno de las 8 posibles configuraciones del robot numeradas desde el 0 hasta el 7. En la siguiente tabla se describe cada una de estas configuraciones en términos de como el robot se posiciona con respecto a las tres singularidades.

| Cfx | Centro muñeca respecto al eje 1 | Centro muñeca respecto eje inferior | Angulo eje 5 |
|-----|---------------------------------|-------------------------------------|--------------|
| 0   | Delante                         | Delante                             | Positivo     |
| 1   | Delante                         | Delante                             | Negativo     |
| 2   | Delante                         | Detrás                              | Positivo     |
| 3   | Delante                         | Detrás                              | Negativo     |
| 4   | Detrás                          | Delante                             | Positivo     |
| 5   | Detrás                          | Delante                             | Negativo     |
| 6   | Detrás                          | Detrás                              | Positivo     |
| 67  | Detrás                          | Detrás                              | Negativo     |

En las siguientes figuras se puede ver cómo se pueden conseguir la misma posición y orientación de la herramienta usando diferentes configuraciones.

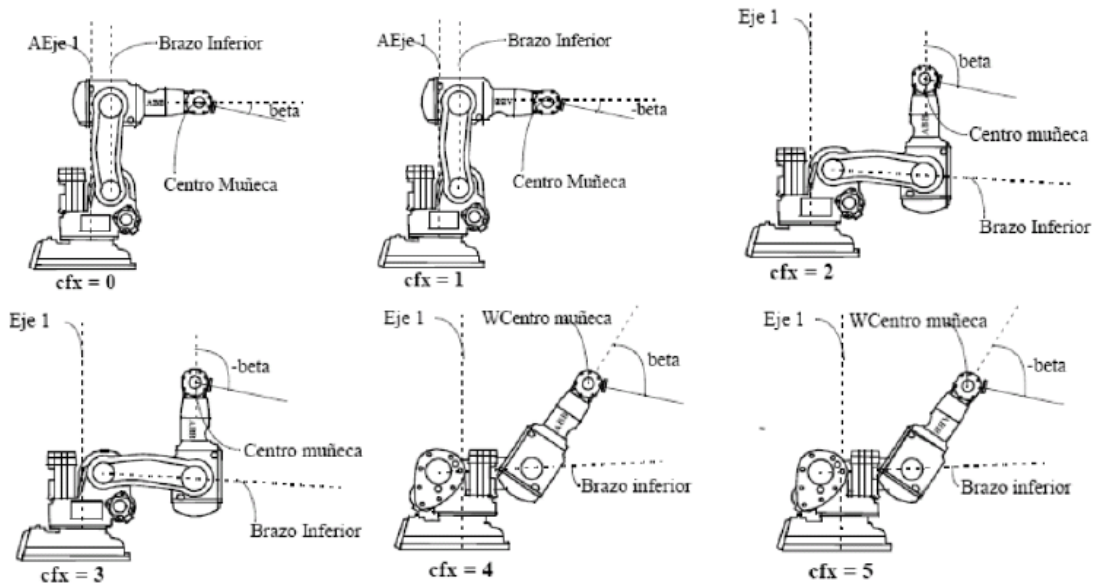


Figura 57. Configuraciones robot, misma posición.

**Cf1.** Eje rotacional: El cuadrante utilizado por el eje 1, expresado con un número entero positivo o negativo.

**Cf4.** Eje rotacional: El cuadrante utilizado por el eje 4, expresado con un número entero positivo o negativo.

**Cf6.** Eje rotacional: El cuadrante utilizado por el eje 6, expresado con un número entero positivo o negativo.

**Cfx.** Eje rotacional: Configuración actual del robot expresada por un entero comprendido entre 0 y 7.

#### 4.1.8.2. Orientación. Orient.

La variable orientación se usa para las orientaciones (como la orientación de una herramienta) y las rotaciones (como la rotación de un sistema de coordenadas).

La orientación se describe bajo la forma de un cuaternión formado por cuatro elementos:  $q_1$ ,  $q_2$ ,  $q_3$  y  $q_4$ . De ésta manera una variable orientación puede tener un valor como  $[1,0,0,0]$  que indicaría que el elemento asociado no tiene ninguna rotación.

La orientación debe ser normalizada, es decir que la suma de los cuadrados de los cuatro debe ser igual a 1:

$$Q_1^2 + Q_2^2 + Q_3^2 + Q_4^2 = 1$$

Un cuaternión permite definir la orientación de un sistema de coordenadas (como la de una herramienta) mediante una matriz rotacional que describe la dirección de los ejes del sistema de coordenadas respecto a un sistema de referencia.

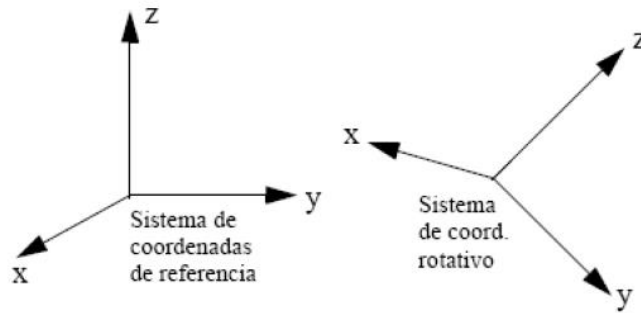


Figura 58. Rotación de un sistema de coordenadas mediante cuaternión.

Los ejes de los sistemas de coordenadas rotativos (x,y,z) son vectores que pueden ser expresados en el sistema de coordenadas de referencia según lo siguiente:

$$X = (x_1, x_2, x_3)$$

$$Y = (y_1, y_2, y_3)$$

$$Z = (z_1, z_2, z_3)$$

Éstos tres vectores pueden ser puestos juntos en una matriz rotacional, donde cada uno de los vectores forma una columna:

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}$$

Un cuaternión no es más que una forma más concisa de describir esa matriz rotacional; los cuaternios se calculan basándose en los elementos de la matriz rotacional:

$$q_1 = \frac{\sqrt{x_1 + y_2 + z_3 + 1}}{2}$$

$$q_2 = \frac{\sqrt{x_1 - y_2 - z_3 + 1}}{2}$$

$$q_3 = \frac{\sqrt{y_2 - x_1 - z_3 + 1}}{2}$$

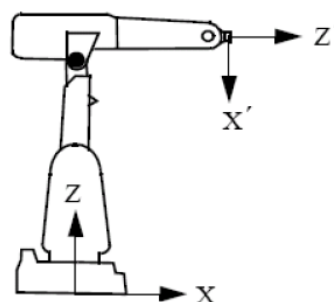
$$q_4 = \frac{\sqrt{z_3 - x_1 - y_2 + 1}}{2}$$

$$\text{signo } q_2 = \text{signo } (y_3 - z_2)$$

$$\text{signo } q_3 = \text{signo } (z_1 - x_3)$$

$$\text{signo } q_4 = \text{signo } (x_2 - y_1)$$

Lo podemos ver de forma sencilla con el siguiente ejemplo en el que una herramienta está orientada de forma que su eje Z' apunte recto hacia delante (en la misma dirección que el eje X del sistema de coordenadas en la base). El eje Y' de la herramienta corresponde al eje Y del sistema de coordenadas en la base.





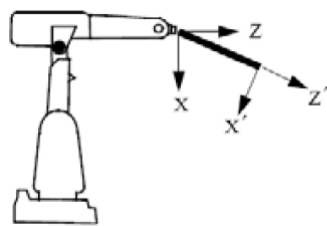
Los ejes estarán relacionados de la siguiente forma:

$$\begin{aligned} x' &= -z = (0, 0, -1) \\ y' &= y = (0, 1, 0) \\ z' &= x = (1, 0, 0) \end{aligned} \Rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Matriz rotacional que corresponde al siguiente cuaternión:

$$\begin{aligned} q_1 &= \frac{\sqrt{0+1+0+1}}{2} = \frac{\sqrt{2}}{2} = 0,707 \\ q_2 &= \frac{\sqrt{0-1-0+1}}{2} = 0 \\ q_3 &= \frac{\sqrt{1-0-0+1}}{2} = \frac{\sqrt{2}}{2} = 0,707 \quad \text{signo } q_3 = \text{signo}(1+1) = + \\ q_4 &= \frac{\sqrt{0-0-1+1}}{2} = 0 \end{aligned}$$

Podemos complicarlo un poco más girando por ejemplo la dirección de la herramienta  $30^\circ$  en los ejes  $X'$  y  $Z'$  respecto al sistema de coordenadas de la muñeca y el resultado sería el siguiente.



$$\begin{aligned} x' &= (\cos 30^\circ, 0, -\sin 30^\circ) \\ y' &= (0, 1, 0) \\ z' &= (\sin 30^\circ, 0, \cos 30^\circ) \end{aligned}$$

$$\begin{bmatrix} \cos 30^\circ & 0 & \sin 30^\circ \\ 0 & 1 & 0 \\ -\sin 30^\circ & 0 & \cos 30^\circ \end{bmatrix}$$

$$\begin{aligned} q_1 &= \frac{\sqrt{\cos 30^\circ + 1 + \cos 30^\circ + 1}}{2} = 0,965926 \\ q_2 &= \frac{\sqrt{\cos 30^\circ - 1 - \cos 30^\circ + 1}}{2} = 0 \\ q_3 &= \frac{\sqrt{1 - \cos 30^\circ - \cos 30^\circ + 1}}{2} = 0,258819 \quad \text{signo } q_3 = \text{signo}(\sin 30^\circ + \sin 30^\circ) = + \\ q_4 &= \frac{\sqrt{\cos 30^\circ - \cos 30^\circ - 1 + 1}}{2} = 0 \end{aligned}$$

#### 4.1.8.3. Datos de posición. RobTarget.

El "RobTarget" sirve para definir la posición del robot y de los ejes externos.

Los datos de posición sirven para definir la posición en las instrucciones de posicionamiento a donde deben moverse, tanto los ejes del robot y los ejes externos.

Dado que el robot es capaz de alcanzar una misma posición de varias formas diferentes, también se deberá especificar la configuración de los ejes. Esto servirá para definir los valores de los ejes en el caso de que sean ambiguos, por ejemplo:

- si el robot está en una posición hacia adelante o hacia atrás,
- si el eje 4 esta girado hacia abajo o hacia arriba,
- si el eje 6 esta en la vuelta positiva o negativa.



La posición será definida basándose en el sistema de coordenadas del objeto de trabajo, incluyendo cualquier desplazamiento de programa. Si la posición es programada con cualquier otro objeto de trabajo que el utilizado en la instrucción, el robot no se moverá de la forma esperada.

Componentes:

- **Trans** (tipo de dato pos). La posición (x, y, z) del punto central de la herramienta expresada en mm. La posición se especifica respecto al sistema de coordenadas del objeto utilizado, incluyendo el desplazamiento del programa. Si no se especifica ningún objeto de trabajo, se usará el sistema de coordenadas mundo.
- **Rot** (tipo de dato orient). La orientación de la herramienta expresada bajo la forma de un quaternion (q1, q2, q3 y q4). La orientación se especifica respecto al sistema de coordenadas del objeto utilizado, incluyendo el desplazamiento del programa. Si no se especifica ningún objeto de trabajo, se usará el sistema de coordenadas mundo.
- **Robconf** (tipo de dato confdata). La configuración de los ejes del robot (cf1, cf4, cf6 y cfx). Esto se define bajo la forma del cuarto de vuelta donde está el eje 1, del eje 4 y del eje 6. El primer cuarto de vuelta positivo, 0-90 o, está definido como 0.

#### 4.1.8.4. Herramienta de trabajo. Tooldata.

Tooldata se usa para describir las características de una herramienta, como por ejemplo, una pistola de soldadura o una pinza.

Si la herramienta está fija en el espacio (si se trata de una herramienta estacionaria), se definirán los datos de herramienta normales correspondientes así como la pinza que sujeta el objeto de trabajo.

Los datos de herramienta afectarán a los movimientos del robot en la siguiente medida:

- El punto central de la herramienta (TCP) se refiere a un punto que cumplirá con la trayectoria especificada y con la exigencia de velocidad. En el caso en que se reoriente la herramienta o si se usan ejes externos coordinados, solamente este punto seguirá la trayectoria deseada a la velocidad programada.
- En el caso en que se use una herramienta estacionaria, tanto la velocidad como la trayectoria programadas se referirán al objeto de trabajo.
- Las posiciones programadas se refieren a la posición del TCP utilizado y a la orientación respecto al sistema de coordenadas de la herramienta. Ello significa que si, por ejemplo, se ha reemplazado una herramienta porque está dañada, se podrá todavía utilizar el programa antiguo si se vuelve a definir el sistema de coordenadas de la herramienta.

Los datos de herramienta también se usan al mover el robot para:

- Definir el TCP que no debe moverse cuando se reorienta la muñeca.
- Definir el sistema de coordenadas de la herramienta con vistas a facilitar el movimiento o rotación de las direcciones de la herramienta.

Es importante definir siempre la carga actual de la herramienta y cuando se use, la carga útil del robot. Definiciones incorrectas de la carga pueden provocar una sobrecarga en la estructura mecánica del robot



Componentes:

- **robhold** (tipo de dato bool). Define si el robot está sujetando la herramienta o no:
  - TRUE-> El robot está sujetando la herramienta.
  - FALSE -> El robot no está sujetando la herramienta, es decir, se trata de una herramienta estacionaria.
- **tframe** (tipo de dato pose). El sistema de coordenadas de la herramienta, es decir:
  - La posición del TCP (x, y, z) en mm, expresado en coordenadas de la muñeca.
  - La orientación del sistema de coordenadas de la herramienta, expresado en el sistema de coordenadas de la muñeca bajo la forma de un cuaternión (q1, q2, q3 y q4). (Véase la Figura.)

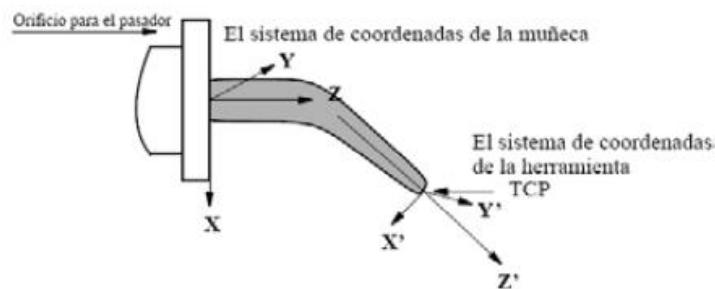


Figura 59. Definición sistema de coordenadas de la herramienta.

Si se utiliza una herramienta estacionaria, la definición se realizará respecto al sistema de coordenadas del mundo.

Si la dirección de la herramienta no está especificada, el sistema de coordenadas de la herramienta y el sistema de coordenadas de la muñeca coincidirán.

#### 4.1.8.5. Datos del objeto de trabajo. Wobjdata.

Se usa para describir el objeto de trabajo que el robot está soldando, procesando, moviendo, etc..

Componentes:

- **robhold** (tipo de dato bool). Define si el robot está sujetando el objeto de trabajo o no:
  - TRUE-> El robot está sujetando el objeto de trabajo, es decir, que está utilizando una herramienta estacionaria.
  - FALSE-> El robot no está sujetando el objeto de trabajo, es decir, que el robot está sujetando la herramienta.
- **Ufprog** (tipo de dato bool). Defise si se usa un sistema de coordenadas fijo del usuario.
  - TRUE-> Sistema de coordenadas fijo del usuario.
  - FALSE->Sistema de coordenadas móvil del usuario, es decir, que se están usando ejes externos coordinados.
- **uframe** (tipo de dato pose). El sistema de coordenadas del usuario, es decir, la posición de la superficie de trabajo o el utillaje utilizados (véase Figura):
  - La posición del origen del sistema de coordenadas (x, y, z) en mm.
  - La rotación del sistema de coordenadas, expresado como un cuaternión (q1,q2,q3,q4).

Si el robot está sujetando la herramienta, el sistema de coordenadas del usuario será definido en el sistema de coordenadas mundo.



- **oframe** (tipo de dato pose). El sistema de coordenadas del objeto, es decir, la posición del objeto de trabajo utilizado (véase Figura):

- La posición del origen del sistema de coordenadas (x, y, z) en mm.
- La rotación del sistema de coordenadas, expresado como un cuaternión (q1,q2,q3,q4).
- El sistema de coordenadas del objeto está definido en el sistema de coordenadas del usuario.

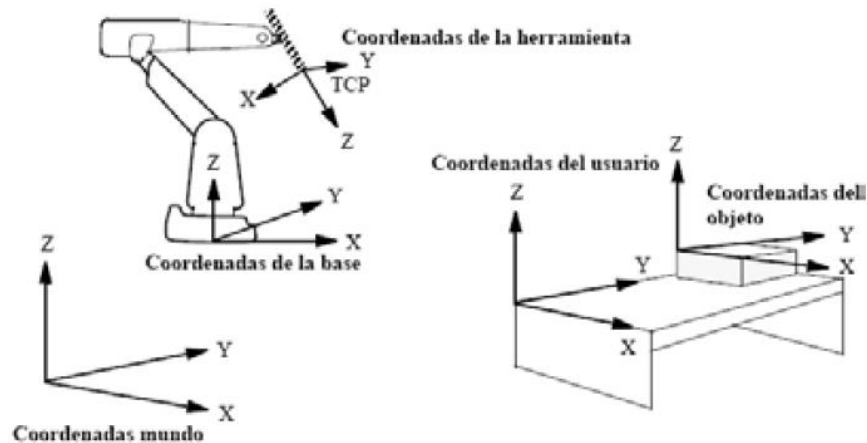


Figura 60. Definición sistema de coordenadas del objeto de trabajo.

#### 4.2. Señales Entrada/Salida.

En RobotStudio podemos crear señales de entrada o salida desde la pestaña **Controlador** seleccionando **I/O system** dentro del menú de **Configuración**. Se pueden asignar a un módulo real de entradas/salidas del robot (Ethernet/IP Device, Industrial Network) o si no se especifica nada, crearlas para uso exclusivo de simulación.

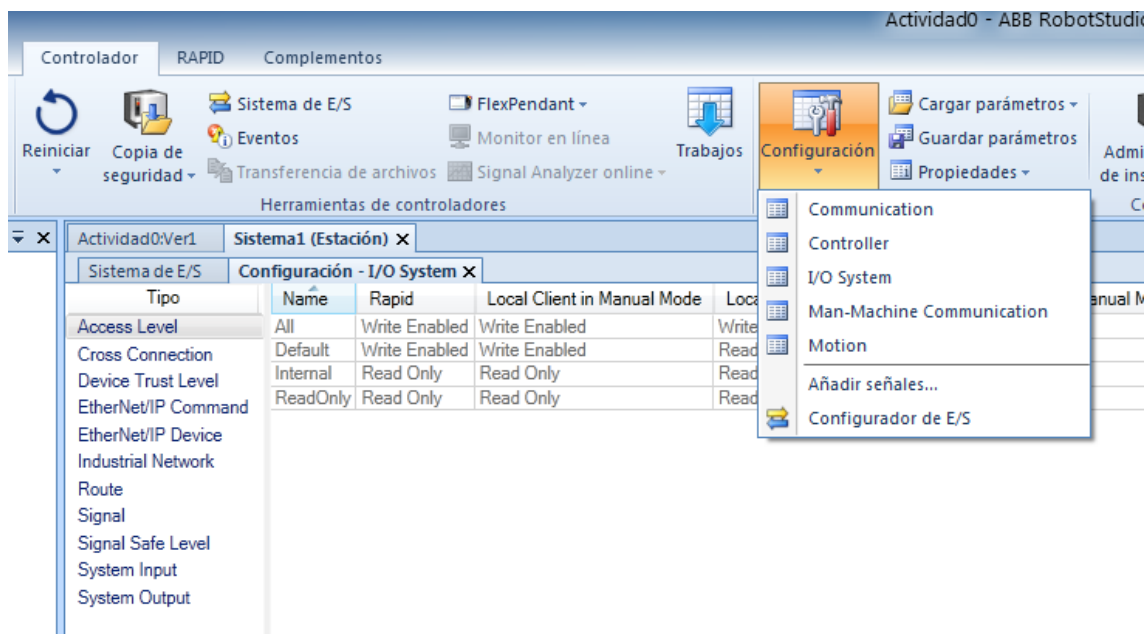


Figura 61. Creación señales E/S.



Para crear una señal de simulación seleccionaremos el tipo **Signal** e indicamos nueva señal. En el menú que se nos abre daremos un nombre a la señal, indicaremos el tipo (entrada o salida) y no la asignaremos a ningún dispositivo al ser únicamente de simulación.

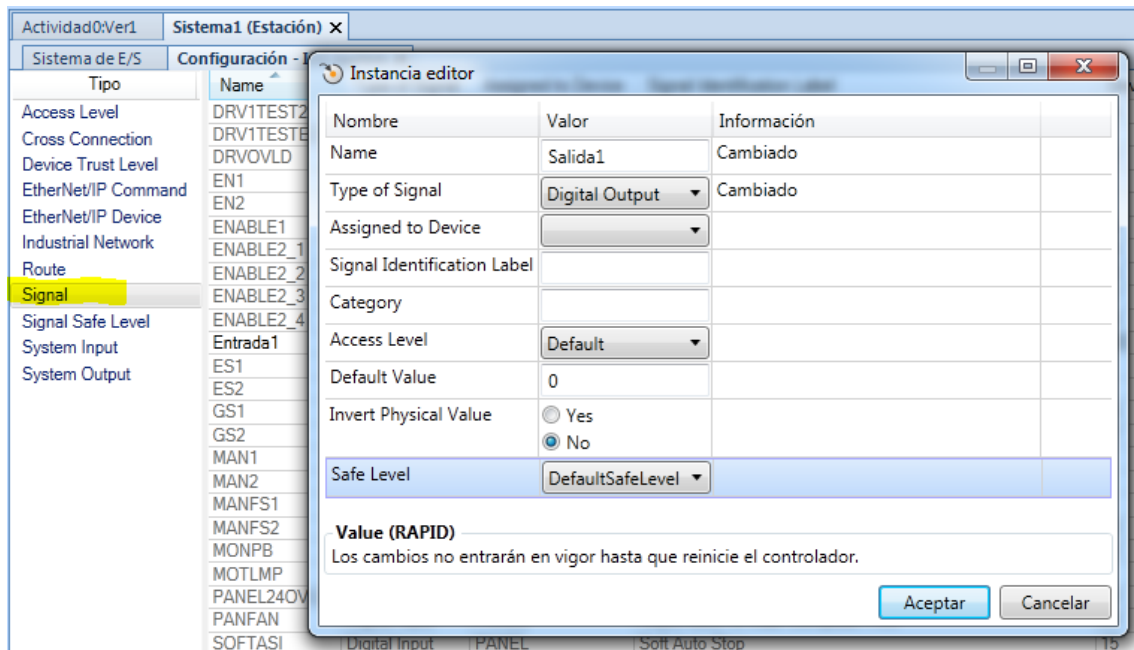


Figura 62. Crear señal de simulación.

Es importante reiniciar el controlador una vez se han creado las señales para que tengan efecto los cambios.

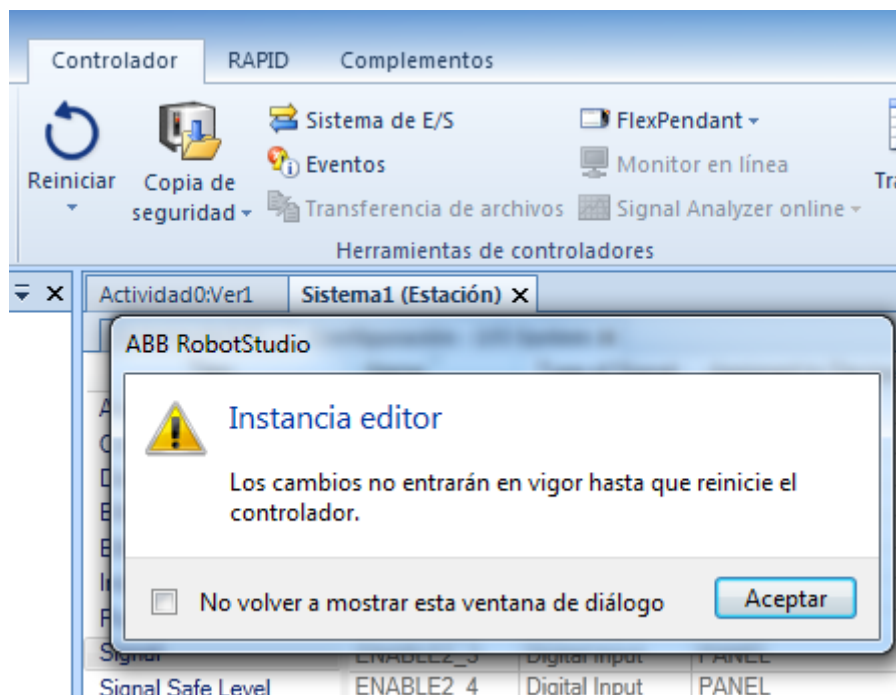


Figura 63. Reiniciar controladora.





## 5. SIMULAR PROGRAMAS.

### 5.1.1. Conceptos básicos.

Antes de simular el programa debemos guardar todos los cambios realizados en el controlador haciendo clic en “Sincronizar” en la pestaña “Posición inicial” y luego hacemos clic en la pestaña “Simulación” de la cinta superior para acceder a todas las herramientas de simulación.

Las simulaciones ejecutan programas de robot completos en un controlador virtual. Antes de ejecutar una simulación debe decidir que trayectorias desea simular.

- Configurar simulación.  
La ventana de diálogo configuración de simulación se utiliza para realizar las dos tareas principales.
  - Configuración de la secuencia y el punto de entrada del programa del robot.
  - Creación de escenarios de simulación para distintos objetos simulados.

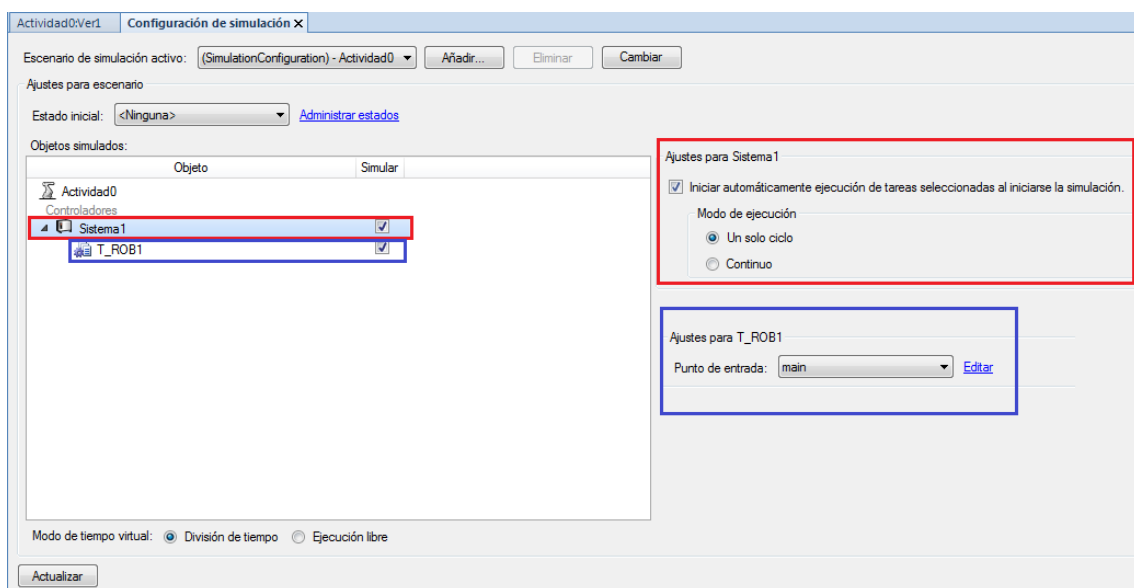


Figura 64. Configuración de simulación.

**Punto de entrada:** La tarea empieza a ejecutarse en la rutina especificada por Punto de entrada. El punto de entrada predeterminada es “main” y el módulo “module1”. Si solo queremos simular una trayectoria la buscaremos en el menú desplegable.

**Modo de ejecución:** Existe modo continuo dónde la rutina seleccionada se ejecuta repetidamente hasta que se detenga el programa y el modo un solo ciclo, dónde la rutina seleccionada sólo se ejecuta una vez y el programa se detiene a continuación.

### 5.1.2. Detección de colisiones.

La detección de colisiones muestra y registra las colisiones y las casi colisiones para los objetos especificados de la estación. Se utiliza normalmente durante la simulación de programas de robot. También puede usarse durante la construcción de la estación.

**Conjunto de colisión:** Un conjunto de colisión contiene dos grupos de objetos, *Objetos A* y *Objetos B*, en los que puede situar objetos para detectar las colisiones existentes entre ellos. Cuando cualquier objeto de *Objetos A* colisiona con cualquier objeto de *Objetos B*, la colisión se representa en la vista gráfica y se registra en la ventana de salida. Puede tener varios conjuntos de colisión en la estación, pero cada conjunto de colisión sólo puede contener dos grupos.



Un uso habitual de los conjuntos de colisión es crear un conjunto de colisión para cada robot de la estación. A continuación, para cada conjunto de colisión coloca el robot y su herramienta en un grupo y en el otro grupo todos los objetos con los que no debe colisionar.

Cada conjunto de colisión puede activarse y desactivarse separadamente.

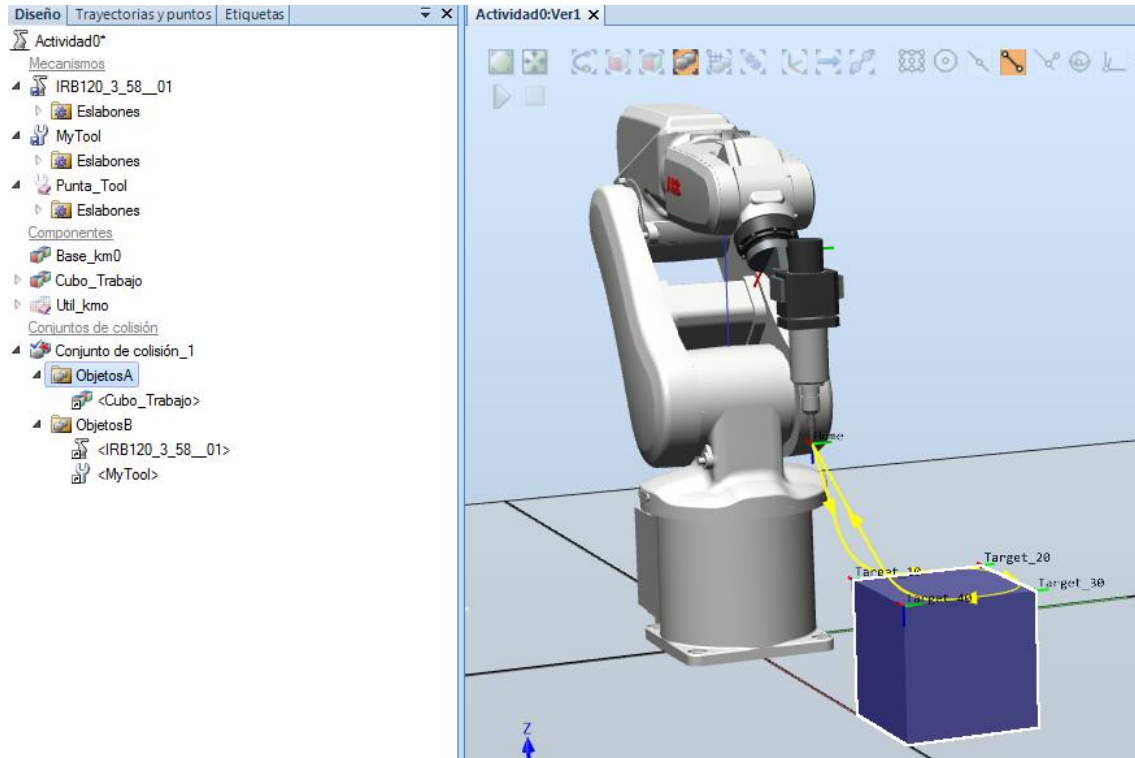


Figura 65. Grupos de colisión A y B.

La detección de colisiones comprueba si los robots u otras piezas móviles colisionan con los equipos de la estación. En el caso de las estaciones complejas, puede utilizar varios conjuntos de colisiones para detectar las colisiones entre varios grupos de objetos.

Una vez configurada la detección de colisiones, no es necesario iniciarla, sino que ésta detecta automáticamente las colisiones de acuerdo con la configuración.

Para definir si las colisiones deben detectarse siempre o sólo durante la simulación hay que dirigirse al menú **Archivo** y hacer clic en **Opciones**, se selecciona **Simulación->Colisión**.

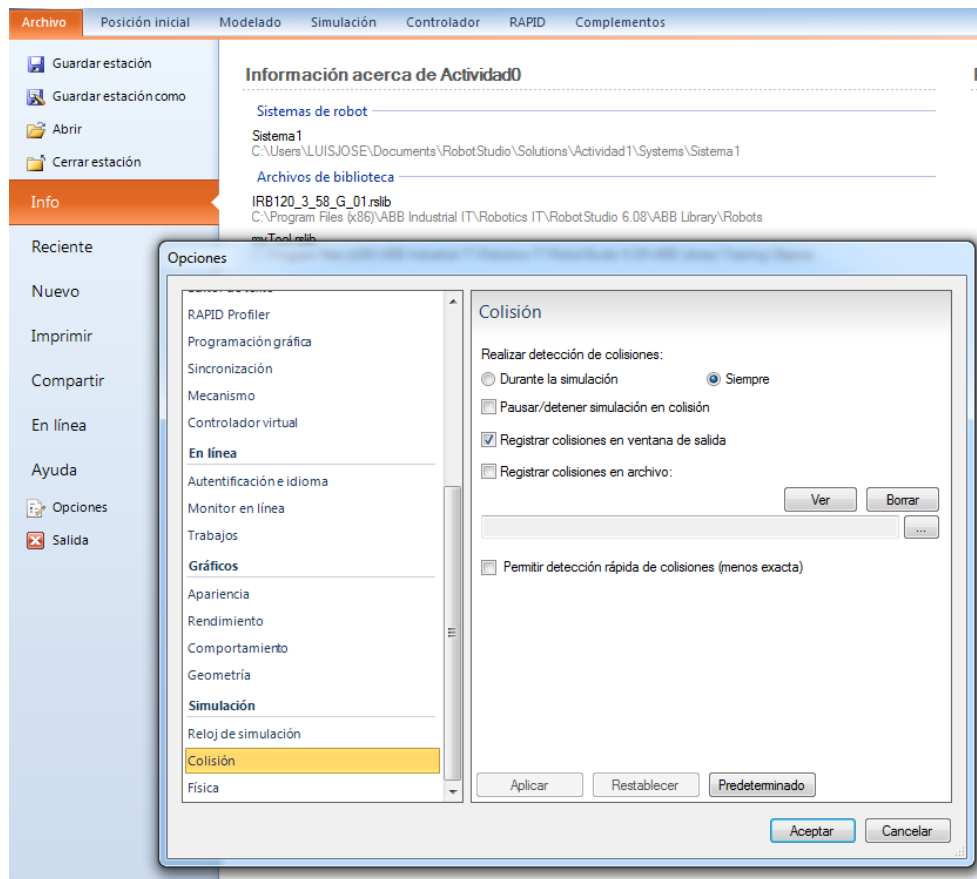


Figura 66. Configuración comprobar colisiones.

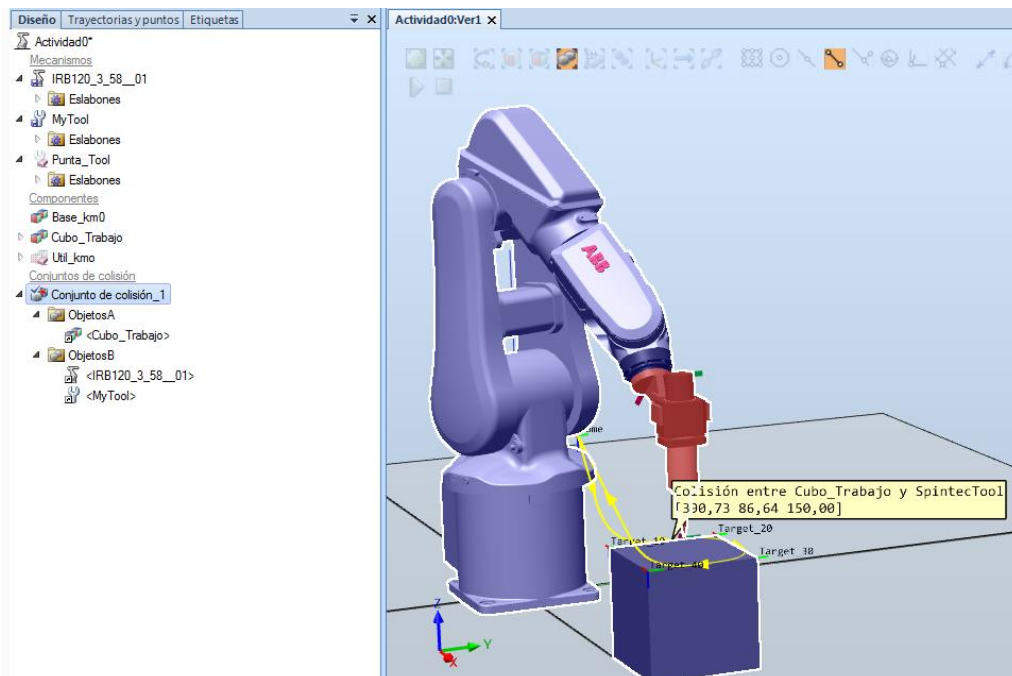


Figura 67. Visualización de colisión.

### 5.1.3. Movimiento objetos.

La simulación nos permite también simular el movimiento de objetos a los que les hemos asociado un comportamiento concreto. RobotStudio permite modificar las características físicas y mecánicas de piezas que hayamos modelado. Por ejemplo, podemos hacer que un tetraedro actúe de cinta transportadora, para ello modificamos sus propiedades desde la pestaña de modelado.

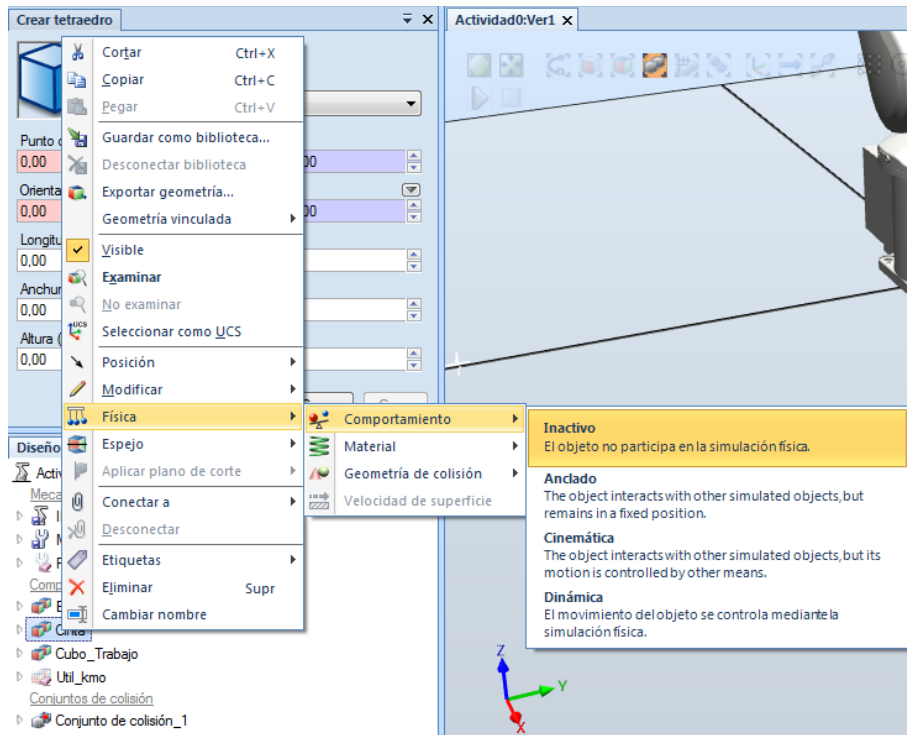


Figura 68. Cambio comportamiento físico pieza.

Cambiando las propiedades de material (anclado) y asignándole una velocidad a su superficie podemos hacer que el tetraedro transporte piezas que caen sobre su superficie.

Después creando otra pieza a la que asignamos un comportamiento dinámico podemos conseguir que la pieza caiga sobre la cinta y se desplace utilizando la simulación de RobotStudio. La simulación de Robotstudio puede guardar estado con lo que podemos fijar una situación estática de las piezas como posición original y volver a ella siempre que queramos.

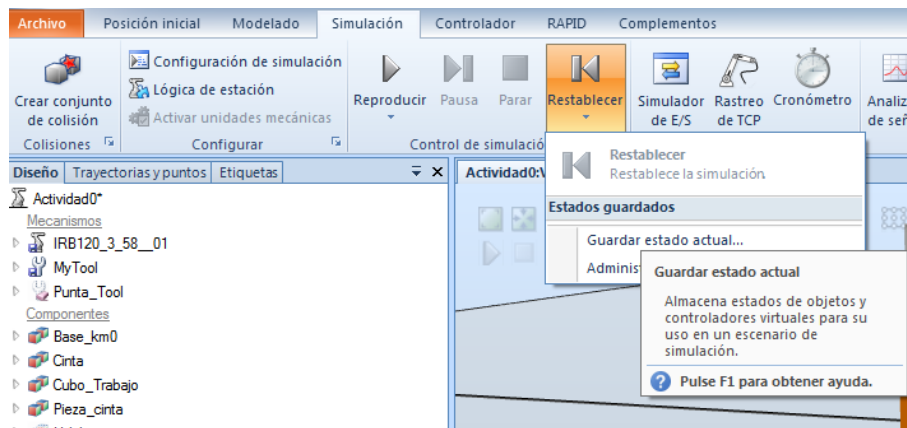


Figura 69. Guardar estado de simulación.



#### 5.1.4. Componentes inteligentes.

RobotStudio permite crear los denominados “Componentes inteligentes” que son objetos con o sin representación gráfica que permiten representar el comportamiento de objetos de una estación tales como sensores, movimiento de piezas, señales de entrada/salida, etc.

Podemos añadir componentes inteligentes de dos formas, la primera desde la pestaña “Modelado” que nos permite crear un contenedor al cual podemos añadir diferentes componentes inteligentes y establecer las propiedades y realizar las conexiones entre ellos para implementar la lógica del comportamiento necesario del conjunto de elementos incluidos en ese contenedor.

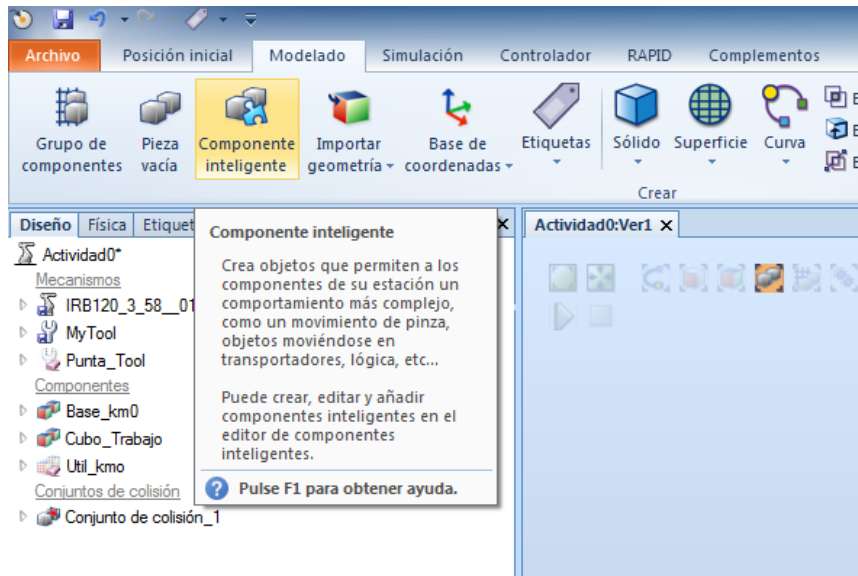


Figura 70. Creación componente inteligente. Modelado.

La segunda forma de añadir un componente inteligente es desde la pestaña “Simulación” con la opción “Lógica de la estación”, en donde podemos añadir componentes inteligentes fuera de un contenedor de componentes inteligentes y configurar las propiedades del componente, así como implementar la lógica de la estación.

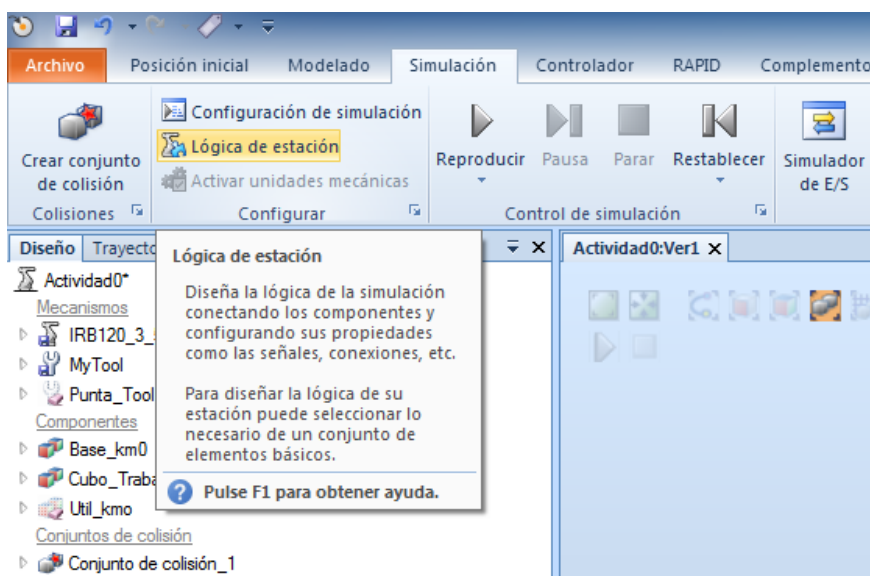


Figura 71. Creación componente inteligente. Simulación.



### Tipos de componentes inteligentes:

- Señales y propiedades. Contiene puertas lógicas, operaciones lógicas, temporizadores.
- Primitivos paramétricos. Contiene prismas, cilindros, repetidores, que pueden variar de tamaño/acción en función de algún parámetro.
- Sensores. Para la detección de objetos, posiciones.
- Acciones. Permite conectar/desconectar objetos, crear copias de objetos o eliminarlos.
- Otros. Permite acciones varias y entre ellas añadir eventos de simulación que podemos utilizar para disparar acciones o cambiar el color de objetos.

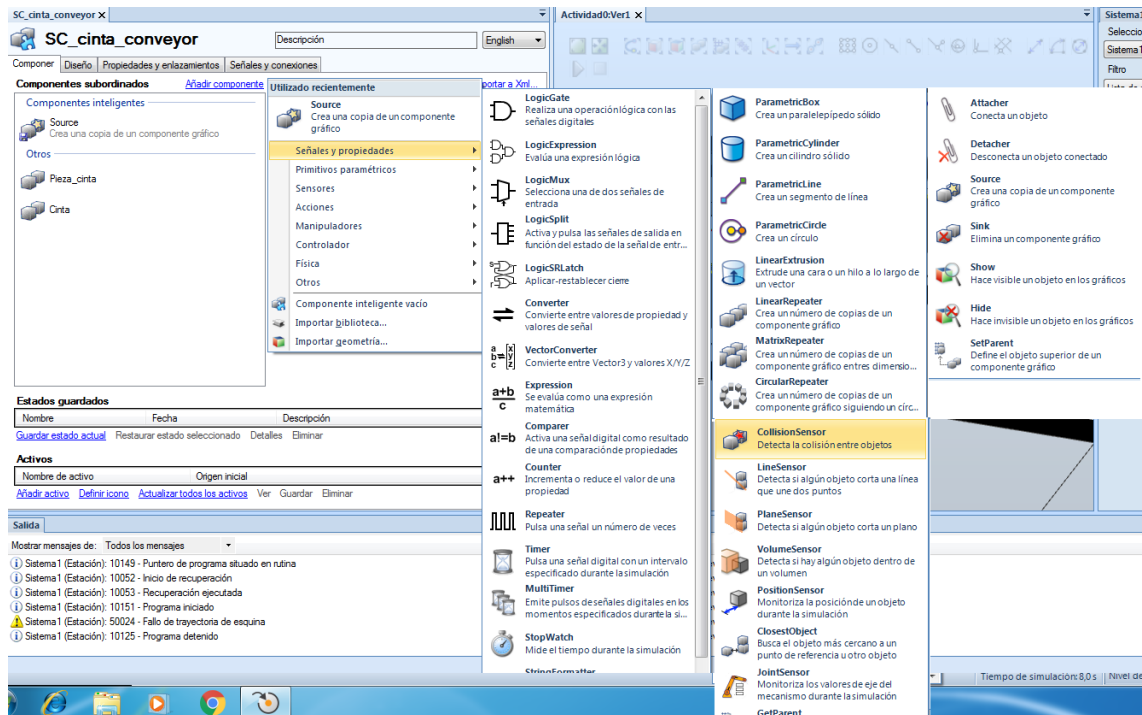


Figura 72. Tipos de componentes inteligentes.

### Lógica de un contenedor inteligente o de la estación:

Cuando hemos creados nuestros componentes inteligentes, podemos establecer la lógica de cómo interactúan entre ellos utilizando sensores, acciones y eventos que podemos ir añadiendo. En la imagen siguiente, por ejemplo, tenemos dos sensores uno en el medio de la cinta y otro al final, todos están dentro de un contenedor. El sensor medio detecta las piezas que el componente “source” genera y hace que genere otra pieza, el sensor fin cuando detecta elimina el objeto detectado. El generador de piezas, genera el primero objeto cuando el bloque “simulations events” detecta que la simulación ha comenzado.

La lógica se construye como en una programación de bloques, conectando las propiedades o señales E/S de cada bloque que necesitemos mediante flechas que establece su interacción.

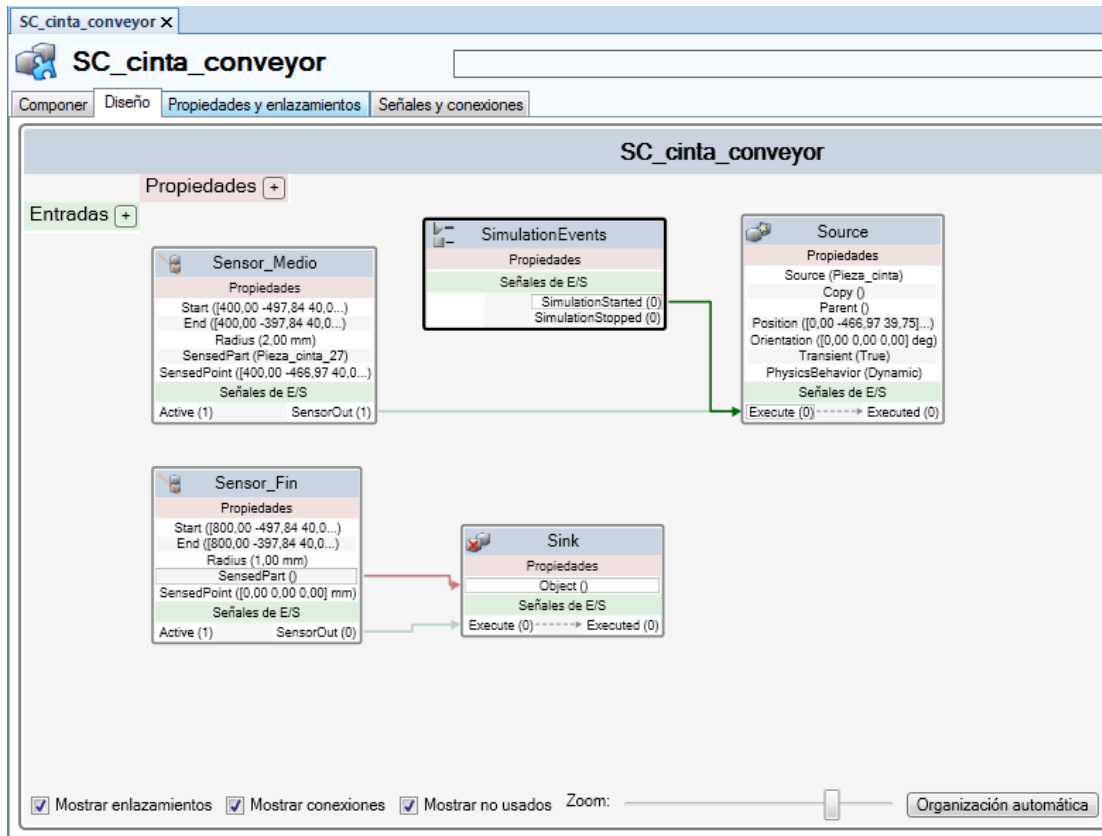


Figura 73. Diseño lógica componente inteligente.

Como hemos indicado anteriormente podemos diseñar la lógica de los contenedores de componentes inteligentes y la lógica de la estación dónde aparecerán los contenedores previamente creados como un bloque más. Cada contenedor puede tener señales de entrada y salida que serán con las que podamos interactuar en el diseño de la lógica de la estación.

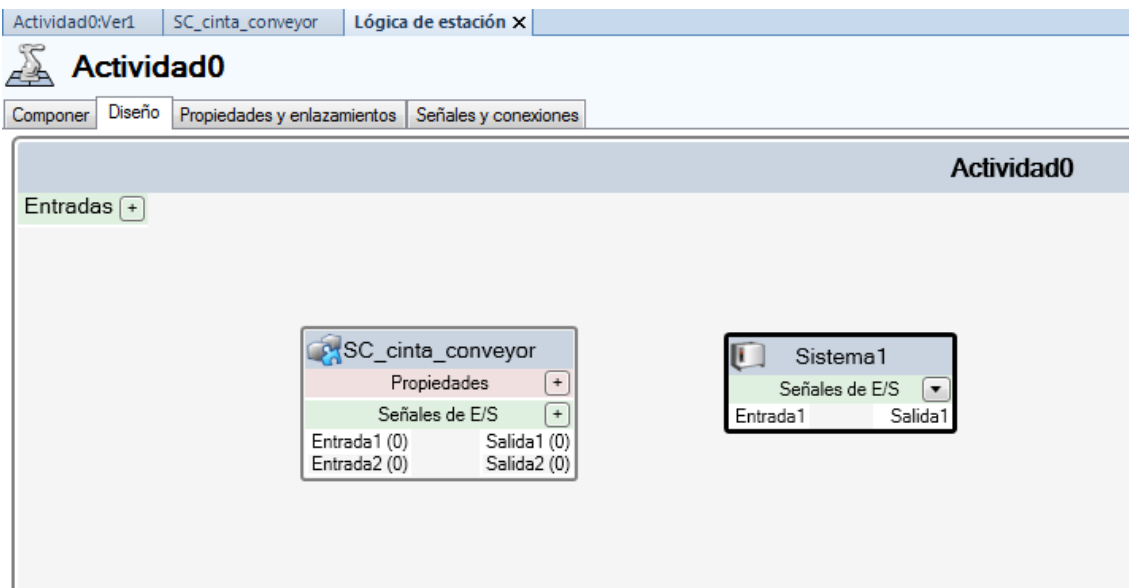


Figura 74. Diseño lógico de la estación.



### 5.1.5. Simulación de señales de E/S.

A la hora de simular señales de E/S, puede crear eventos que establecen los valores de las señales cuando se cumplen determinadas condiciones de disparo, o bien puede establecer los valores de las señales manualmente.

Para simular las señales una vez creadas (ver punto 4.2), primeramente debemos sincronizar con rapid y una vez hecho esto en la pestaña de simulación, elegimos el menú **Simular E/S** dónde nos aparecerá un panel en el que seleccionaremos las señales que deseamos visualizar durante la simulación, pudiendo de ésta manera modificar su valor durante la simulación.

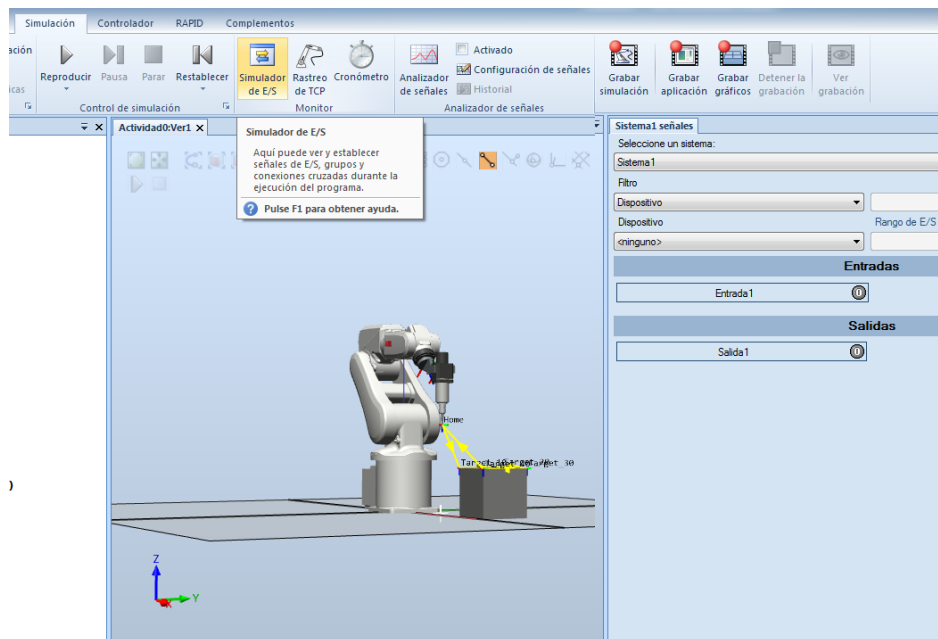


Figura 75. Simular señales E/S.

Para visualizar nuestras señales en el panel podemos crear una nueva lista que agrupe las señales que hemos creado y de ésta manera cuando queramos visualizarlas accederemos a ellas rápidamente.



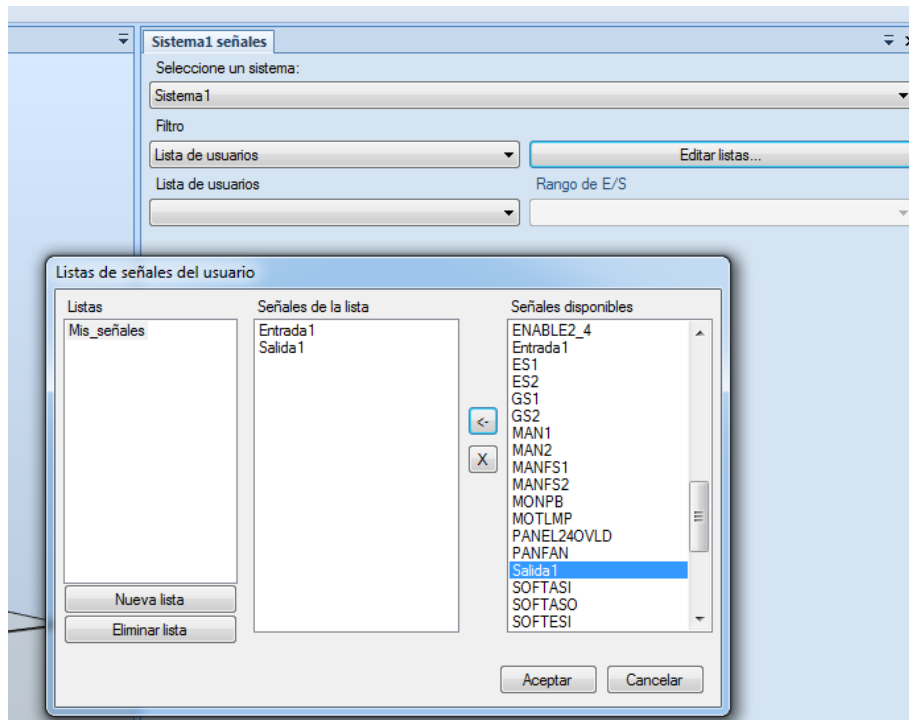


Figura 76. Listas de señales.

Podemos utilizar fácilmente una señal de entrada para detener el movimiento de nuestro robot simulando que espera un evento de un dispositivo externo que le autorice a iniciar o continuar el movimiento, utilizando una instrucción de acción.

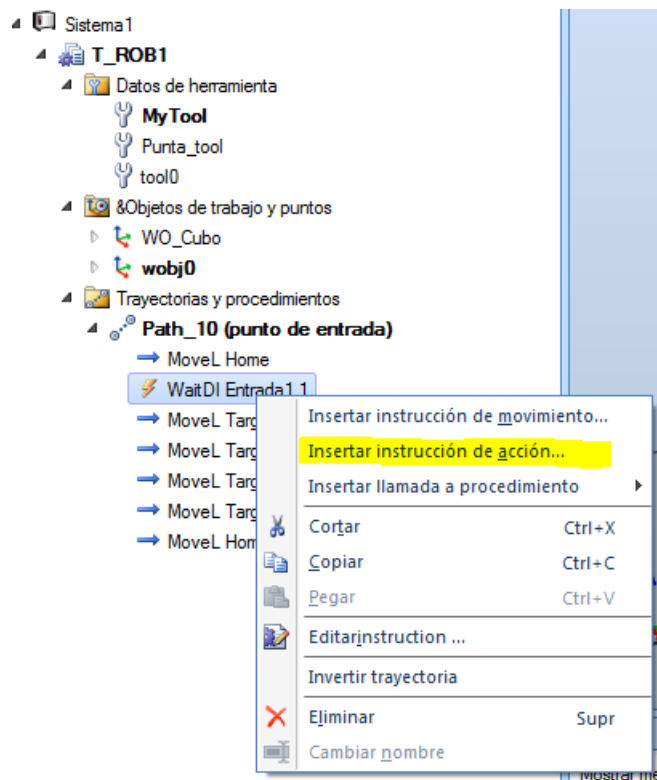


Figura 77. Instrucción de acción con entrada.





## 6. ACTIVIDADES A DESARROLLAR

Desarrollaremos 6 actividades según el siguiente esquema:

- Actividad 1. Inicio al Software RobotStudio.
- Actividad 2. Controladora y modelado de piezas.
- Actividad 3. Creación de herramientas de trabajo.
- Actividad 4. Creación de trayectorias.
- Actividad 5. Creación de objetos de trabajo.
- Actividad 6. Entradas y Salidas virtuales. SmartComponents.



## 6.1. Actividad1. Inicio al Software RobotStudio.

1. Objetivos.
  - Creación estación vacía.
  - Familiarización con el entorno de trabajo.
  - Inserción de componentes.
  - Desplazamiento de componentes.
  - Salvaguardas del proyecto.
  
2. Materiales.

Video disponible en anexo.  
Contenidos teóricos de éste TFM.
  
3. Desarrollo de la actividad.

En ésta actividad el alumno comenzará a trabajar con el software RobotStudio. Se pretende que el alumno se familiarice con las teclas que se utilizan para el manejo del software y con los diferentes menús disponibles. Se añadirá un Robot y aprenderemos a moverlo eje a eje. También se añadirán otros componentes habituales en una estación robotizada y serán ubicados en una posición de la estación.
  
4. Duración.

La duración de la actividad será aproximadamente de 45 minutos. El video disponible tiene una duración de 10 minutos que el alumno podrá utilizar durante toda la actividad cómo apoyo o a través de una única visualización al comienzo de la actividad.



## 6.2. Actividad 2. Controladora y modelado de piezas.

### 1. Objetivos.

- Familiarización con el entorno de trabajo.
- Concepto y finalidad de una controladora.
- Diferencia entre movimientos lineales y de orientación.
- Aprender a crear objetos mediante modelado.
- Asociación ideas de contenidos teóricos y prácticos.

### 2. Materiales.

Video disponible en anexo.

Contenidos teóricos de éste TFM.

### 3. Desarrollo de la actividad.

Cómo continuación a la actividad anterior se añadirá una controladora a la estación de trabajo. Esto nos permitirá poder realizar movimientos con el robot de tipo lineal y de reorientación. Realizaremos movimientos mediante otras herramientas del software que permitirán al alumno conocer mejor la morfología del robot, sus grados de libertad y por tanto la dependencia de los diferentes ejes de movimiento del robot. Con ésta actividad el alumno tendrá una visión más real de todo el contenido teórico recibido hasta el momento, realizando una asociación de ideas que facilitará la comprensión y aprendizaje significativo de conceptos básico en la realización de movimientos de un Robot. Finalmente se crearán diferentes piezas mediante la herramienta de modelado del software como paso previo a la creación de nuestras propias herramientas de trabajo.

### 4. Duración.

La duración de la actividad será aproximadamente de 45 minutos. El video disponible tiene una duración de 10 minutos que el alumno podrá utilizar durante toda la actividad cómo apoyo o a través de una única visualización al comienzo de la actividad.



### 6.3. Actividad 3. Creación de herramientas de trabajo.

#### 1. Objetivos.

- Compresión concepto herramienta trabajo.
- Creación herramienta trabajo simple.
- Creación herramientas trabajo complejas.
- Aprender a crear objetos mediante modelado.
- Asociación ideas de contenidos teóricos y prácticos.

#### 2. Materiales.

Video disponible en anexo.

Contenidos teóricos de éste TFM.

#### 3. Desarrollo de la actividad.

En ésta actividad se trata que el alumno profundice en el concepto de herramienta de trabajo. El concepto y creación de herramientas de trabajo es parte fundamental del contenido teórico de éste TFM y en ésta actividad repasaremos el concepto de herramienta de trabajo, para que sirve, que funcionalidad tiene y el nexo de unión que establece con el Robot. Crearemos primeramente una herramienta de trabajo simple para finalmente crear una herramienta de trabajo más compleja y de gran similitud a las herramientas que podemos encontrarnos en el ámbito industrial.

#### 4. Duración.

La duración de la actividad será aproximadamente de 45 minutos. El video disponible tiene una duración de 18 minutos que el alumno podrá utilizar durante toda la actividad cómo apoyo o a través de una única visualización al comienzo de la actividad.



#### 6.4. Actividad 4. Creación de trayectorias.

##### 1. Objetivos.

- Creación puntos de trabajo.
- Modificación de puntos de trabajo.
- Creación de trayectorias.
- Simulación de trayectorias.
- Asociación ideas de contenidos teóricos y prácticos.

##### 2. Materiales.

Video disponible en anexo.

Contenidos teóricos de éste TFM.

##### 3. Desarrollo de la actividad.

Una vez afianzado el concepto de herramienta de trabajo, en ésta actividad avanzaremos hacia la creación de trayectorias. Para ello comenzaremos añadiendo una herramienta de trabajo de la biblioteca a nuestro Robot para a continuación comenzar a aprender a crear puntos de trabajo, su vinculación con un objeto de trabajo, a modificarlos, así como conocer las diferentes orientaciones que puede tener la herramienta en cada punto dependiendo de la configuración que tome el Robot. Finalmente, con los puntos creados generaremos una trayectoria y visualizaremos el movimiento del robot a través de la trayectoria creada.

##### 4. Duración.

La duración de la actividad será aproximadamente de 45 minutos. El video disponible tiene una duración de 15 minutos que el alumno podrá utilizar durante toda la actividad cómo apoyo o a través de una única visualización al comienzo de la actividad.



## 6.5. Actividad 5. Creación de objetos de trabajo.

### 1. Objetivos.

- Creación objetos de trabajo.
- Vínculo objetos y puntos de trabajo.
- Modificación de objetos de trabajo.
- Dependencias sistemas de referencia objeto y herramienta de trabajo.
- Asociación ideas de contenidos teóricos y prácticos.

### 2. Materiales.

Video disponible en anexo.

Contenidos teóricos de éste TFM.

### 3. Desarrollo de la actividad.

Otro punto importante de nuestro TFM es la comprensión del concepto objeto de trabajo. En ésta actividad aprenderemos a crear un objeto de trabajo y asociarlo a un determinado elemento mecánico de la estación. También a orientar el sistema de referencia del objeto de trabajo y conocer que vinculación tiene con el sistema de referencia de la herramienta de trabajo, y como ésta relación repercute en la orientación que toma la herramienta de trabajo cuando alcanza un determinado punto de la trayectoria. Finalmente veremos la ventaja de trabajar con diferentes objetos de trabajo asociados a diferentes elementos mecánicos de la estación cuando la estación se debe reconfigurar o trasladar.

### 4. Duración.

La duración de la actividad será aproximadamente de 45 minutos. El video disponible tiene una duración de 15 minutos que el alumno podrá utilizar durante toda la actividad cómo apoyo o a través de una única visualización al comienzo de la actividad.





## 6.6. Actividad 6. Entradas y Salidas virtuales. Smart Components.

### 1. Objetivos.

- Creación entradas/salidas digitales.
- Simulación entradas/salidas digitales.
- Creación elementos Smart components.
- Diseño lógica estación.
- Simulación funcionamiento estación real mediante Smart components.
- Asociación ideas de contenidos teóricos y prácticos.

### 2. Materiales.

Video disponible en anexo.

Contenidos teóricos de éste TFM.

### 3. Desarrollo de la actividad.

En ésta actividad profundizaremos en la idea de entrada/salida digital. Mediante RobotStudio crearemos entradas y salidas virtuales que simularemos y veremos que utilidad tienen en un sistema de control y en nuestro caso como elementos condicionantes del movimiento de un Robot. A partir de aquí, nos centraremos en los Smart componentes y en su potencia en la simulación de estaciones reales, mediante la creación de elementos animados controlables mediante la programación en bloques.

### 4. Duración.

La duración de la actividad será de dos sesiones de aproximadamente de 45 minutos. El video disponible tiene una duración de 45 minutos que el alumno podrá utilizar durante toda la actividad cómo apoyo o a través de una única visualización al comienzo de la actividad.





## 7. TEMPORALIZACIÓN.

Para llevar a cabo una integración del presente TFM dentro de la unidad didáctica de Robótica Industrial del grado superior de Técnico Superior en Automatización y Robótica Industrial podemos establecer una temporalización ideal cómo la de la tabla siguiente.

| NUMERO     | TEORÍA     | PRÁCTICA   |
|------------|------------|------------|
| ACTIVIDAD1 | 1 SESIÓN   | 1 SESIÓN   |
| ACTIVIDAD2 | 1 SESIÓN   | 1 SESIÓN   |
| ACTIVIDAD3 | 1 SESIÓN   | 1 SESIÓN   |
| ACTIVIDAD4 | 1 SESIÓN   | 1 SESIÓN   |
| ACTIVIDAD5 | 1 SESIÓN   | 1 SESIÓN   |
| ACTVIDIAD6 | 2 SESIONES | 2 SESIONES |

*Tabla 5. Temporalización contenidos y actividades TFM*

Para cada actividad tendríamos sesiones dedicadas a la parte teórica y a la parte práctica. Cada sesión tendría una duración estimada de 45 minutos y quedarían integradas dentro de las 80h dedicadas a la unidad didáctica de “Robótica Industrial”. Teniendo en cuenta la asignación horaria de la asignatura a la que le pertenecen cuatro horas semanales, se prevé una duración de cuatro semanas para abordar los contenidos y las actividades.

El número de alumnos óptimo sería desde 10 hasta un máximo de 16 alumnos, que podríamos agrupar en grupos de 2 dependiendo del número de equipos de trabajo disponibles. El número de alumnos queda limitado por la cantidad de equipos disponibles debido a que éstos deben cumplir unos requerimientos de hardware mínimos en cuanto a memoria RAM, disco duro y número de núcleos para un correcto funcionamiento del software.

Si bien, el presente TFM está orientado a ser usado como complemento a la formación académica oficial y con el objetivo de ampliar los conocimientos adquiridos por los alumnos del ciclo, por tanto, su utilización está dirigida a una formación extraescolar autodidacta, para aquellos alumnos interesados en proyectar su formación más allá de los contenidos académicos reglados.





## 8. RESULTADOS DE APRENDIZAJE.

La organización de los contenidos del TFM responde al criterio elegido por el alumno que lo elabora en base a su experiencia profesional y formación educativa. Incorpora parte de los cinco bloques que se corresponden con los resultados de aprendizaje del módulo profesional de “Robótica Industrial”, sacados del documento curricular base del ciclo profesional, pero dándole un punto de vista más práctico y haciendo hincapié en contenidos teóricos que son considerados básicos para la resolución de cualquier problema robótico profesional, mediante de su adecuada interpretación y comprensión.

El fin último, por tanto, es la creación de un aprendizaje significativo que constituya una base para que el alumno pueda profundizar en los contenidos de la robótica a partir de la generación de una destreza que le ayudará tanto en su etapa académica como profesional.

Como resultados de aprendizaje podemos citar:

- Comprensión de los principios teóricos básicos de la estructura y funcionamiento de los Robots.
- Creación estación robotizada con software profesional.
- Creación herramientas, objetos, variables según estándar profesional.
- Programación y simulación de una estación robot.
- Se conoce el concepto E/S digital y su utilización como elementos disparadores de eventos en un sistema automatizado.





## 9. METODOLOGÍA Y EVALUACIÓN

### 9.1. METODOLOGÍA.

A pesar de tratarse de una propuesta didáctica de gran contenido práctico, existen conocimientos teóricos de gran importancia que deben ser conocidos y que su aprendizaje puede convertirse en meramente conceptual por no resultar motivador para el alumno. Se debe, por tanto, busca la combinación adecuada de parte práctica y teórica, desarrollando actividades desde el principio y buscando que la parte teórica resulte atrayente para el alumno buscando ejemplos de aplicación práctica de la parte teórica y como un buen conocimiento de la misma puede afectar al resultado final de diseño.

#### 9.1.1. Método trabajo parte teórica.

El sistema empleado para desarrollar las clases teóricas, es en general, el exponer por parte del profesor o profesora los contenidos a tratar, así como la resolución de problemas, esto sin restar importancia a la participación activa de los alumnos y alumnas planteando cuestiones, dudas e intercambios de opiniones en la discusión de los problemas prácticos.

La exposición de los conceptos se realiza apoyándose en el uso de la pizarra, proyector y fotocopias principalmente, pudiéndose en cualquier momento utilizar cuantos recursos materiales sean necesarios y se encuentren a disposición del profesor o profesora, incluso apoyándonos en el software de diseño apoyándonos de su potencial para mostrar conceptos teóricos, acercando de ésta forma al alumno al software que posteriormente utilizará.

El comienzo de los contenidos teóricos se efectúa con una introducción motivadora, poniendo de manifiesto la utilidad de la misma en el mundo profesional. De esta manera el alumno y la alumna emprende los conceptos nuevos de una forma más receptiva y buscando el carácter práctico del mismo.

#### 9.1.2. Método trabajo parte práctica.

Dentro de la parte práctica podemos distinguir entre la que realizará el propio alumno individualmente y la que se realiza con ayuda del profesor, que consiste en realizar actividades paso a paso mediante el software, que puedan ser visualizadas por los alumnos a través del proyector. Durante éste tipo de clases el alumno puede ir siguiendo los pasos a través de su ordenador de modo que el aprendizaje sea mayor basándonos en el descubrimiento propio del software por parte del alumno.

En la elaboración del TFM se han generado unos videos que simulan éstas actividades, que son resueltas paso a paso y con audio explicativo que permita entender al alumno la secuencia seguida y la asociación de los conceptos teóricos con los prácticos. Éste tipo de metodología es cada vez más usada y lo podemos ver como un tipo de aproximación al “**video learning**”, que pasaremos a explicar a continuación y que se está constituyendo como una de las herramientas más demandas y eficaz en el ámbito educativo, sobre todo en la enseñanza a distancia.



## 9.2. VIDEO LEARNING.

Son vídeos pedagógicos creados usualmente para complementar una formación presencial o ser el recurso principal de un programa de e-learning, como un máster a distancia. Pueden ser vídeos producidos en tiempo real o grabados.

Los contenidos grabados pueden ser videotutoriales, video explicativo, documental, video divulgativo, entrevistas, presentaciones y más. En este sentido, la duración es variada de un minuto a más de una hora.

### 9.2.1. Ventajas del video learning.

El vídeo se presenta como una herramienta más de aprendizaje, para no decir la favorita, y no sólo para los estudiantes sino también para los profesores, pues de esta manera pueden cumplir su labor de coach como demanda el sector educativo hoy en día. Así lo ven las casas de estudio más vanguardistas.

Otras de las ventajas pedagógicas del vídeo son su facilidad de uso y adaptabilidad, al igual que su alto poder de captación de atención, sin contar que favorece la memorización y la comprensión.

Asimismo, facilita la comprensión de contenidos complejos y largos gracias a que este pueden incluirse ejemplos visuales de conceptos y procedimientos concretos. Además, es el medio que estimula más canales sensoriales del estudiante.

Desde el punto de vista técnico, es una herramienta que puede ser reproducida varias veces y en dicho proceso se va aumentando la comprensión del receptor. Puede alojarse en la nube con lo cual puede compartirse con facilidad, en cualquier momento y lugar, es decir, puede visualizarse en ordenadores, tabletas y smartphones.





### 9.3. CRITERIOS DE EVALUACIÓN.

La evaluación no forma parte de los objetivos de ésta propuesta didáctica y por tanto queda fuera de su ámbito de aplicación. Sin embargo, vamos a presentar unas tablas con los criterios de evaluación que podemos asociar a cada resultado de aprendizaje.

#### **RESULTADO DE APRENDIZAJE 1**

Comprensión de los principios teóricos básicos de la estructura y funcionamiento de los Robots.

#### **CRITERIO DE EVALUACIÓN**

- a) Se identifica grados de libertad del robot.
- b) Se identifica los diferentes sistemas de referencia asociados al robot.
- c) Se comprende dependencia posición punto final del robot con la posición de sus brazos.

#### **RESULTADO DE APRENDIZAJE 2**

Creación estación robotizada con software profesional.

#### **CRITERIO DE EVALUACIÓN**

- a) Se crea una estación con todos sus elementos funcionales.
- b) Es capaz de añadir diferentes elementos en función de los requerimientos de la aplicación.
- c) Se conoce la función de cada uno de los elementos de una estación.

#### **RESULTADO DE APRENDIZAJE 3**

Creación herramientas, objetos, variables según estándar profesional.

#### **CRITERIO DE EVALUACIÓN**

- a) Se comprende el concepto y utilidad de herramienta y objeto de trabajo.
- b) Se crean diferentes herramientas con diferente tcp.
- c) Se crean objetos de trabajo ligados a la morfología de la estación.

#### **RESULTADO DE APRENDIZAJE 4**

Programación y simulación de una estación robot.

#### **CRITERIO DE EVALUACIÓN**

- a) Se diseñan diferentes tipos de trayectorias asociadas a la estación.
- b) Se modifica las instrucciones de movimiento de la trayectoria.
- c) Se conocen los diferentes parámetros asociados las funciones de movimiento.

#### **RESULTADO DE APRENDIZAJE 5**

Se conoce el concepto E/S digital y su utilización como elementos disparadores de eventos en un sistema automatizado.

#### **CRITERIO DE EVALUACIÓN**

- a) Sabe utilizar las herramientas de simulación de E/S digitales.
- b) Utiliza E/S como elementos de decisión en sus programas.
- c) Crea diferentes elementos móviles en la estación que interactúan.





## 10. CONCLUSIONES

Los objetivos inicialmente planteados en éste trabajo, los cuales podemos agrupar en:

- Comprensión de los principios teóricos básicos de la estructura y funcionamiento de los Robots.
- Introducción a la programación de Robots mediante un software de diseño profesional.

Se han completado y plasmado en éste documento.

De su elaboración podemos concluir, en primer lugar, que el desarrollo de contenidos destinado a alumnos de ciclos formativos de grado medio o superior incluidos en la formación profesional, requiere de gran esfuerzo y conocimiento de la materia.

Éste trabajo, partió de la detección de una laguna en la formación de los estudiantes, en materia de robótica, a partir del estudio de los contenidos de los diferentes módulos ligados a ésta especialidad incluidos en los ciclos formativos mencionados al inicio del documento y de la propia experiencia del autor durante su actividad como responsable de recursos humanos dentro de su actividad profesional.

Los puntos incluidos constituyen la base de los conocimientos que se deben adquirir, tanto a nivel teórico como a nivel práctico, para iniciar una actividad profesional dentro del campo de la robótica, se han desarrollado de manera minuciosa, sistemática y lo más detallada posible, con el objetivo de facilitar al alumno su comprensión y la utilización del documento.

A pesar de ser un trabajo teórico, que no se ha podido poner en práctica, la metodología utilizada para su elaboración, así como los recursos didácticos utilizados, garantizan su aplicación tanto a nivel particular como a nivel educativo, constituyendo una herramienta que cualquier docente de la especialidad podría integrar fácilmente en sus clases, suponiendo un elemento de apoyo que unido a las habilidades pedagógicas del docente, por un lado, con su dominio de la materia, por otro, puede garantizar la adquisición de conocimientos por parte del alumno.

Cómo líneas futuras de trabajo, existe un amplio abanico de posibilidades. Desde continuar con el desarrollo de actividades, en una línea similar a la planteada, con el objetivo de completar la formación de una manera más significativa, hasta continuar con el aprendizaje del manejo del software, profundizando en aspectos más complejos, de modo que el alumno alcance un nivel de destreza en el manejo del mismo notable. Para ello, se requeriría de una dedicación personal por parte de un docente o grupo de docentes, apasionados por la materia y por su trabajo, que de una manera organizada fueran aportando contenidos en la misma línea planteada, es decir, mediante contenidos teóricos desgranados paso a paso y directamente ligados a su aplicación, así como mediante la elaboración de videos que mostrasen de una manera dinámica su aplicación práctica.

Finalmente, podemos añadir, que si se pudieran analizar los resultados de una supuesta aplicación práctica, podríamos concluir que la línea de trabajo escogida, puede ser fácilmente corregida y modificable por parte de cada docente adaptándose a su contexto particular.





## BIBLIOGRAFÍA

Connolly, C. (2009). Technology and applications of ABB RobotStudio. *Industrial Robot: An International Journal*.

Neto, P. (2014). A Guide for ABB Robot Studio.

Robotics, A. B. B. (2008). Manual del Operador RobotStudio. *Västerås, Sweden*.

BOE-A-2013-1033 Orden ECD/102/2013

BOE-A-2011-19530 Real Decreto 1581/2011





## ANEXOS

### Actividad 1.

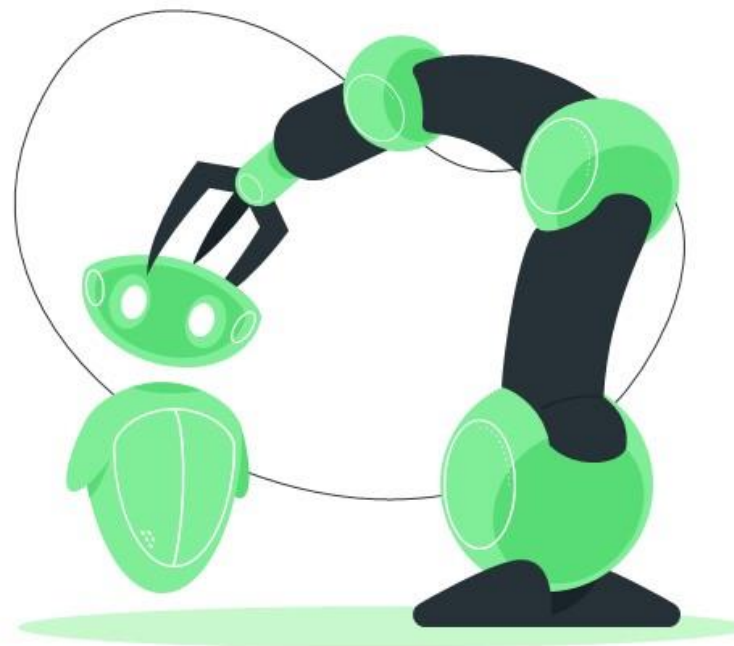
# ACTIVIDAD 1

RobotStudio:  
Inicio al Software RobotStudio.

Realizado por: Luis José Cenizo Revuelta

MÁSTER EN PROFESOR DE EDUCACIÓN SECUNDARIA OBLIGATORIA Y  
BACHILLERATO, FORMACIÓN PROFESIONAL Y ENSEÑANZAS DE IDIOMAS

UVa





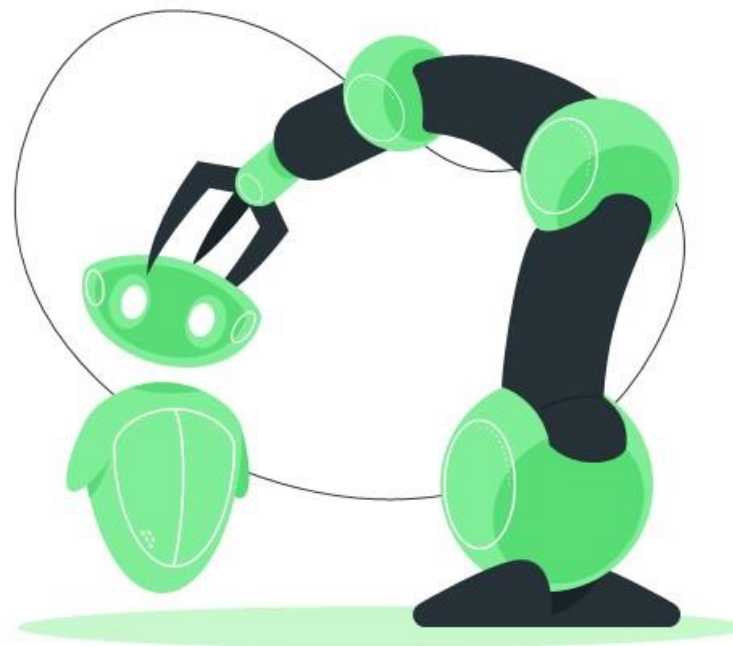
## Actividad 2.



# ACTIVIDAD 2

RobotStudio:  
Controladora y modelado de piezas.

Realizado por: Luis José Cenizo Revuelta



MÁSTER EN PROFESOR DE EDUCACIÓN SECUNDARIA OBLIGATORIA Y  
BACHILLERATO, FORMACIÓN PROFESIONAL Y ENSEÑANZAS DE IDIOMAS





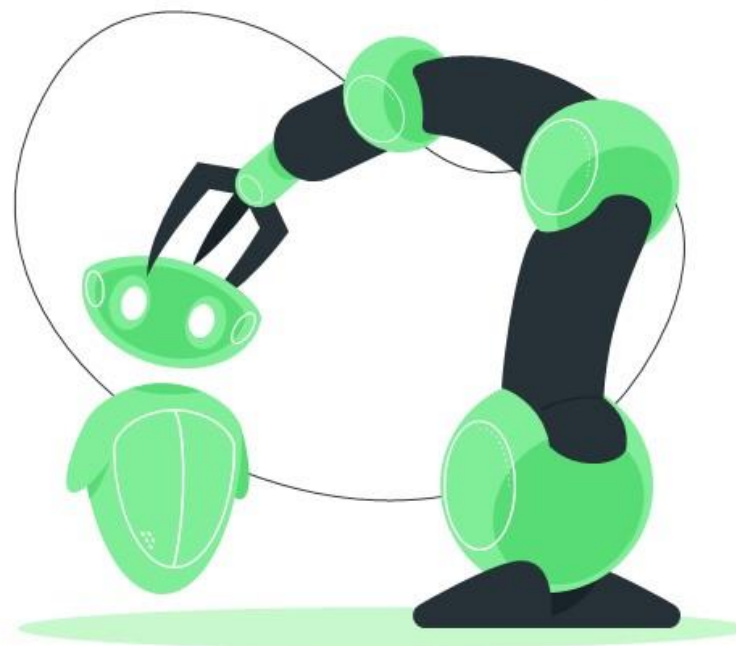
Actividad 3.



# ACTIVIDAD 3

RobotStudio:  
Creación de herramientas de trabajo.

Realizado por: Luis José Cenizo Revuelta



MÁSTER EN PROFESOR DE EDUCACIÓN SECUNDARIA OBLIGATORIA Y  
BACHILLERATO, FORMACIÓN PROFESIONAL Y ENSEÑANZAS DE IDIOMAS



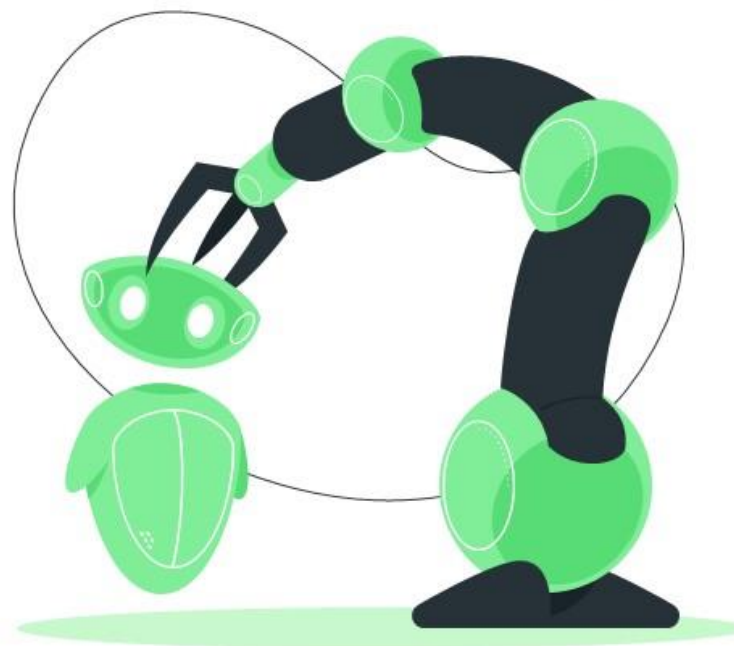
Actividad 4.



# ACTIVIDAD 4

RobotStudio:  
Creación de trayectorias.

Realizado por: Luis José Cenizo Revuelta



MÁSTER EN PROFESOR DE EDUCACIÓN SECUNDARIA OBLIGATORIA Y  
BACHILLERATO, FORMACIÓN PROFESIONAL Y ENSEÑANZAS DE IDIOMAS



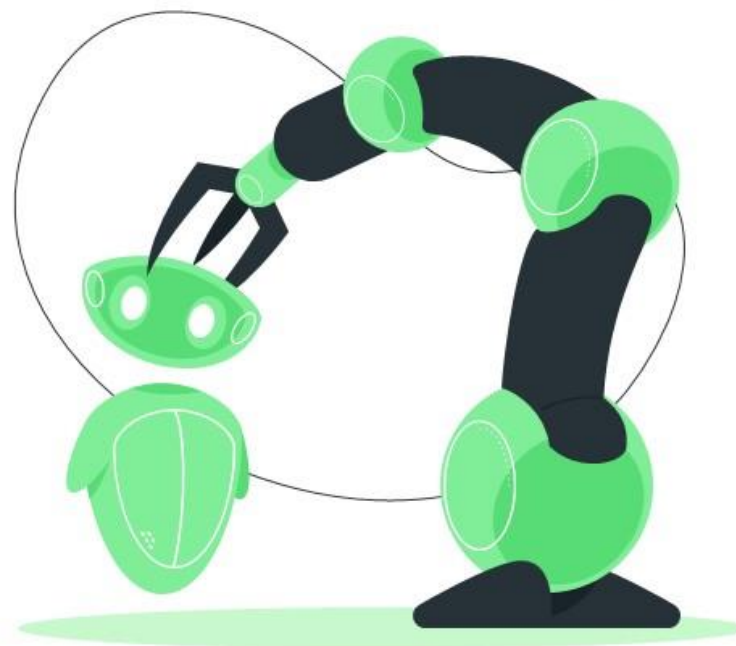
Actividad 5.



# ACTIVIDAD 5

RobotStudio:  
Creación de objetos de trabajo.

Realizado por: Luis José Cenizo Revuelta



MÁSTER EN PROFESOR DE EDUCACIÓN SECUNDARIA OBLIGATORIA Y  
BACHILLERATO, FORMACIÓN PROFESIONAL Y ENSEÑANZAS DE IDIOMAS



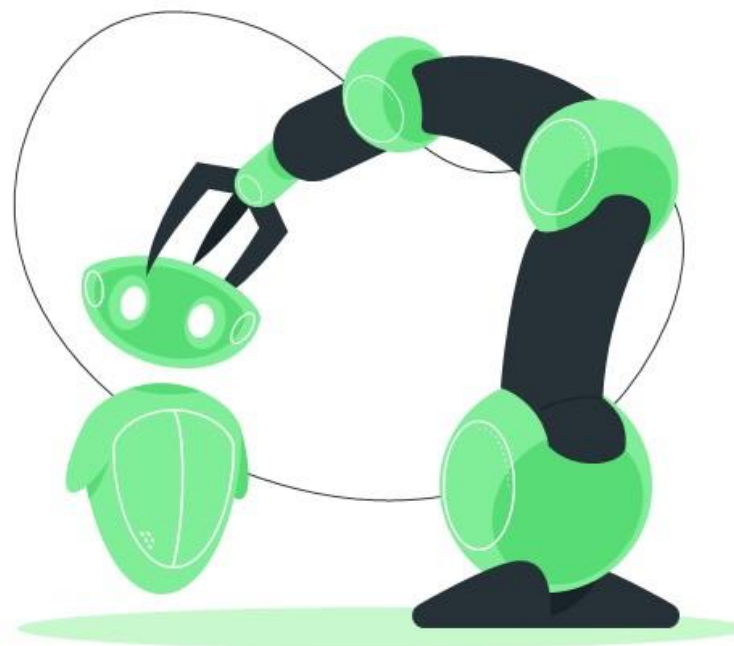
Actividad 6.



# ACTIVIDAD 6

RobotStudio:  
Entradas y Salidas virtuales.  
Smart Components.

Realizado por: Luis José Cenizo Revuelta



MÁSTER EN PROFESOR DE EDUCACIÓN SECUNDARIA OBLIGATORIA Y  
BACHILLERATO, FORMACIÓN PROFESIONAL Y ENSEÑANZAS DE IDIOMAS