



**DEPARTAMENTO DE INGENIERÍA
DE SISTEMAS Y AUTOMÁTICA**



MÁSTER EN AUTOMÁTICA E INFORMÁTICA INDUSTRIAL

Trabajo Final de Máster

TAREAS DE ENSAMBLADO CON UN ROBOT COLABORATIVO DE DOS BRAZOS

Autor: Sara Rabadán Mayordomo

Tutor: Martín Mellado Arteché

Cotutor: Carlos Blanes Campos

Valencia, 2019

Agradecimientos

A todas las personas que han formado parte en la realización de este proyecto, a mi familia y amigos por el apoyo, en especial, a Iván Salvador por estar conmigo durante todo el proceso de este trabajo, de principio a fin. A mis compañeros de laboratorio, por contribuir con sus conocimientos y amenizar el trabajo realizado.

Mis más sinceros agradecimientos a Martín Mellado como tutor y a Carlos Blanes como cotutor por prestarme ayuda y atención siempre que la he necesitado, sobre todo en aquellas competencias requeridas en el proyecto que están fuera del plan de estudios del máster. Gracias por el esfuerzo y la confianza.

Resumen

Con el actual trabajo se presenta el desarrollo de una aplicación de ensamblaje de piezas mediante la utilización de un robot colaborativo de dos brazos robot, YuMi de ABB, que requiere, además, la colaboración de los dos brazos de un operario.

Dicho trabajo fue planteado por el Instituto de Automática e Informática industrial (ai2) de la UPV en calidad de un proyecto de investigación y estudio de los robots.

Para llevar a cabo dicho estudio se diseñó el ensamblaje de la rueda de un carro de compra debido a los grados de libertad de movimiento que requiere. Dichas piezas fueron impresas en material de impresora 3D para, posteriormente, probar su aplicación con el robot. Para ello, se realizó la programación de un software en RAPID, el lenguaje de programación de ABB, donde se hizo uso de todas las ventajas que presenta este robot de dos brazos para realizar el montaje. En concreto, se utilizaron los módulos de visión y vacío que incorpora, las herramientas de Smart Gripper y los distintos tipos de movimientos que permite su cinemática ampliada a siete articulaciones.

Finalmente, la realización del proyecto resultó exitosa ya que la metodología seguida fue comprobada físicamente en una situación real, obteniendo los resultados esperados con el ensamblado del producto final realizado.

Palabras clave: Ensamblado, robot colaborativo y brazo dual

Resum

Amb l'actual treball es presenta el desenvolupament d'una aplicació d'acoblament de peces per mitjà de la utilització d'un robot col·laboratiu de dos braços robot, YuMi d'ABB, que requerix, a més, la col·laboració dels dos braços d'un operari.

El treball va ser plantejat per l'Institut d'Automàtica i Informàtica industrial (ai2) de la UPV en qualitat d'un projecte d'investigació i estudi dels robots.

Per a dur a terme dit estudi es va dissenyar l'acoblament de la roda d'un carro de compra a causa dels graus de llibertat de moviment que requereix. Les peces van ser impreses en material d'impressora 3D per a, posteriorment, provar la seua aplicació amb el robot. Per a això, es va realitzar la programació d'un programari en RAPID, el llenguatge de programació d'ABB, on es va fer ús de tots els avantatges que presenta aquest robot de dos braços per a realitzar el muntatge. En concret, es van utilitzar els mòduls de visió i buit que incorpora, les ferramentes de Smart Gripper i els distints tipus de moviments que permet la seua cinemàtica ampliada a set articulacions.

Finalment, la realització del projecte va resultar reeixida, ja que la metodologia seguida va ser comprovada físicament en una situació real, obtenint els resultats esperats amb l'acoblament del producte final realitzat.

Paraules clau: Acoblament, robot col·laboratiu i braç dual.

Abstract

With the current paper, the development of an assembly application of workpieces is presented through using a collaborative robot with two robot arms, YuMi from ABB, which also requires the collaboration of the two operator arms.

This project was proposed by the Institute of Automation and Industrial Informatics (ai2) of the UPV as a research and study project for robots.

To carry out this study, the wheel assembly of a shopping cart was designed due to the movement freedom degrees it requires. These workpieces were printed on 3D printer material to, subsequently, test their application with the robot. For this, the software programming was performed in RAPID, the ABB programming language, where all the advantages of this two-arm robot for assembly were made. Specifically, the vision and vacuum modules that it incorporates, the Smart Gripper tools and the different types of movements that allow its extended kinematics to seven joints were used.

Finally, the performance of the project was successful since the methodology followed was physically verified in a real situation, obtaining the expected results with the final product assembly made.

Keywords: Assembly, collaborative robot and dual arm.

Contenido

1. Introducción.....	14
1.1 Antecedentes	14
1.2 Motivación.....	15
1.3 Necesidad del trabajo	16
2. Objetivo y especificaciones	17
3. Contexto teórico.....	19
3.1 Robots tradicionales vs robots colaborativos	19
3.2 Cobots	22
3.1 Normativa	24
4. Estado del arte.....	27
4.1 Robots de un brazo robótico.....	27
4.2 Robots de dos brazos robóticos.....	32
5. Aplicación colaborativa persona-robot	34
5.1 Aplicación de ensamblaje.....	34
5.2 Puesta en marcha YuMi.....	39
5.3 Software	49
6. Validación de la aplicación.....	67
7. Conclusiones.....	70
8. Referencias	72
1. PRESUPUESTO.....	77
2. PLANOS.....	79
3. CÓDIGO.....	86

Índice de ilustraciones

Ilustración 1: Digitalización industria 4.0	14
Ilustración 2: Robot colaborativo de Universal Robots.....	15
Ilustración 3: Robots industriales tradicionales	19
Ilustración 4: Cobots en el mundo empresarial	22
Ilustración 5: Espacio de trabajo común.....	23
Ilustración 6: Niveles de cooperación entre un humano y un robot	23
Ilustración 7: Parada monitorizada.....	25
Ilustración 8: Guiado manual	26
Ilustración 9: Control de velocidad.....	26
Ilustración 10: Limitación de fuerza.....	26
Ilustración 11: Gama de robots Universal Robots.....	27
Ilustración 12: LBR iiwa 7 ejes	29
Ilustración 13: Modelos pequeños de cobots FANUC	30
Ilustración 14: Modelo CR-35iA de FANUC.....	31
Ilustración 15: Robot colaborativo de dos brazos TIAGo++.....	32
Ilustración 16: Robot colaborativo YuMi de ABB.....	33
Ilustración 17: Modelo diseño de piezas	34
Ilustración 18: Casquillos de fricción libre	35
Ilustración 19: Encaje de SmartGripper y la rueda diseñada.....	36
Ilustración 20: Diseño CAD del soporte de la rueda	36
Ilustración 21: CAD piezas laterales	37
Ilustración 22: CAD soporte de piezas laterales.....	37
Ilustración 23: CAD pieza superior.....	38
Ilustración 24: Disposición del material de manipulación del operario.....	38
Ilustración 25: Área de trabajo en el laboratorio de Industria 4.0	39
Ilustración 26: Alimentación del robot IRB 14000.....	39
Ilustración 27: Conexiones periféricas	40
Ilustración 28: Unidad de programación FlexPendant	41
Ilustración 29: Menú principal FlexPendant.....	41
Ilustración 30: Conexión al controlador del robot.....	42
Ilustración 31: Creación de una estación en RobotStudio	42
Ilustración 32: Simulación IRB 14000.....	43
Ilustración 33: FlexPendant Virtual	43

Ilustración 34: Botones liberadores de frenos.....	44
Ilustración 35: Movimiento guiado	44
Ilustración 36: Pantalla de movimiento para el uso del joystick.....	45
Ilustración 37: Posición de calibración.....	46
Ilustración 38: Selección de articulaciones a calibrar	47
Ilustración 39: Flujograma del software.....	49
Ilustración 40: Posición inicial T_ROB_L.....	51
Ilustración 41: Posición de agarre primera pieza.....	52
Ilustración 42: Pantalla Smart Gripper	52
Ilustración 43: Programación de la primera posición de encaje.....	53
Ilustración 44: Movimiento coordinado.....	53
Ilustración 45: Cambio de agarre al brazo derecho	54
Ilustración 46: Ir a la posición de encaje final.....	54
Ilustración 47: Retirada brazo izquierdo	54
Ilustración 48: Comunicación mediante la pinza izquierda	55
Ilustración 49: Finalización T_ROB_L.....	55
Ilustración 50: Posición inicial brazo derecho	56
Ilustración 51: Módulo de vacío IRB 14000	56
Ilustración 52: Código succión de pieza lateral	56
Ilustración 53: Pantalla del módulo neumático en FlexPendant.....	57
Ilustración 54: Código referente al primer encaje de piezas	57
Ilustración 55: Primera parte de la aplicación de ensamblado	58
Ilustración 56: Comunicación operario mediante SmartGripper	58
Ilustración 57: Movimiento coordinado brazo derecho.....	59
Ilustración 58: Posición de ensamblado final	59
Ilustración 59: Comunicación a través del módulo de aire.....	60
Ilustración 60: Giro de muñeca del brazo derecho	60
Ilustración 61: Deposición pieza final.....	60
Ilustración 62: Cámara Cognex A3.....	61
Ilustración 63: Patrón de calibración con origen.....	62
Ilustración 64: Captura de la pieza tras la calibración.....	63
Ilustración 65: Reconocimiento de la imagen por el patrón PatMax.....	63
Ilustración 66: Datos enviados a RAPID	64
Ilustración 67: Declaración de variables de visión.....	64
Ilustración 68: Código de captura de imagen.....	65

Ilustración 69: Homografía de la cámara al mundo real	65
Ilustración 70: Posición de alcance de la pieza por visión	65
Ilustración 71: Creación de un nuevo wobjdata	66
Ilustración 72: Pieza final ensamblada.....	67
Ilustración 73: Torniqueta de apriete.....	69

Índice de tablas

Tabla 1: Comparación robots tradicionales vs colaborativos	20
Tabla 2: Tipos de interacción de robots industriales	21
Tabla 3: Comparativa de características robots UR	28
Tabla 4: Tabla comparativa robots KUKA	29
Tabla 5: Comparativa robots FANUC	31
Tabla 6: Características YuMi de ABB	33
Tabla 7: Conexiones periféricas YuMi	40
Tabla 8: Ángulos de calibración de los ejes	47
Tabla 9: Fuerzas y pares del robot en distintos modos:.....	48
Tabla 10: Par en las articulaciones correspondientes de la muñeca	48
Tabla 11: Velocidades máximas de los ejes del robot	48
Tabla 12: Especificaciones cámara Cognex A3	61
Tabla 14: Tiempos de la aplicación humano-robot	67
Tabla 15: Tiempos aplicación humano	68
Tabla 16: Presupuesto de la maquinaria.....	77
Tabla 17: Presupuesto del material	77
Tabla 18: Presupuesto de mano de obra.....	78
Tabla 19: Presupuesto total.....	78

Trabajo Final de Máster

MEMORIA

Sara Rabadán Mayordomo

1. Introducción

A continuación, se van a exponer las causas y motivos por lo que se cree necesario el trabajo de estudio y se planteará un método de automatización de ensamblado con un robot colaborativo de dos brazos.

1.1 Antecedentes

Desde hace ya varios años, el término alemán Industria 4.0 hace referencia a la cuarta revolución industrial que afecta directamente a las nuevas tecnologías. En concreto, se pretende automatizar la industria y llevarla a un nivel más avanzado, a la digitalización, donde todos los dispositivos y tecnología estén comunicados entre sí y operen independientemente formando, así, una industria “inteligente”.

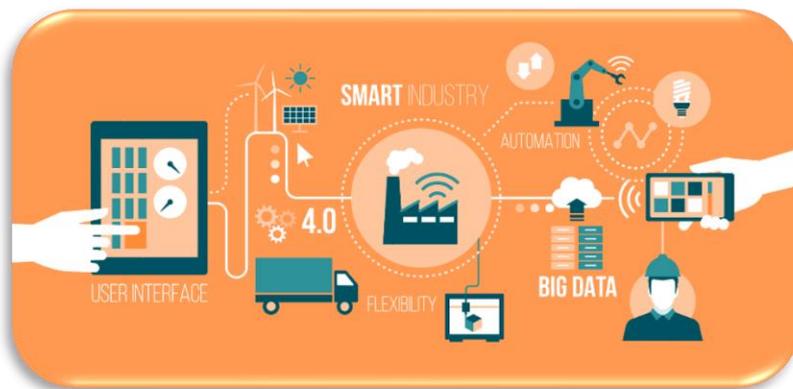


Ilustración 1: Digitalización industria 4.0

Entre la maquinaria de esta industria se encuentran los robots donde se pueden diferenciar los tradicionales robots industriales y los robots colaborativos. Es en estos últimos donde se va a hacer un enfoque especial debido a las características que proporcionan respecto a la integración en el entorno con los humanos.

La principal función de estos robots es realizar tareas repetitivas, pesadas, aburridas o precisas liberando así a los operarios de tener que hacerlas y encomendarlos a otro tipo de trabajos como la supervisión, control o programación ofreciendo trabajos de mayor calidad que benefician tanto a la empresa, por su mayor rendimiento y producción, y al empleado en calidad de vida (Pelegrí, 2018).

1.2 Motivación

Según la empresa Universal Robots (UR) la venta de robots en el entorno industrial aumentó en un 29% en 2018, en concreto, en el sector automovilístico donde, según la Federación Internacional de Robótica, España se sitúa como país importante dentro del mercado europeo (Pelegri, 2019).



Ilustración 2: Robot colaborativo de Universal Robots

Las ventajas de proporcionan los robots colaborativos son las que incentivan a las empresas a invertir cada vez más en su incorporación en las plantas de producción. Entre estas ventajas se encuentran:

- **Accesibilidad:** Cualquier operario puede ser capaz de programar el robot con una formación básica previa.
- **Seguridad:** Los sensores que incorpora, como el sensor de fuerza, detecta colisiones y para de inmediato el movimiento del robot para garantizar la seguridad del operario.
- **Flexibilidad:** Son manejables y pueden realizar diversas tareas.

El conjunto de estas características proporciona un producto final de mayor calidad y un personal más satisfecho con su trabajo al poder colaborar con dichos robots.

1.3 Necesidad del trabajo

Es debido al gran avance tecnológico y nivel de integración en el mercado de los robots colaborativos por lo que se plantea este proyecto como necesidad de estudio de investigación y poder implantar el robot YuMi de ABB, o cualquier otro robot colaborativo, en cadenas de producción donde los empleados puedan programar, casi de forma intuitiva, nuevas tareas para el robot y poder modificar tareas ya implantadas que se adapten a nuevas necesidades.

En concreto, es necesario el uso de un robot colaborativo para la aplicación de ensamblado que se plantea, por lo que, este robot colaborativo de dos brazos y siete articulaciones es idóneo para ello y, además, mucho más conveniente que el ensamblado entre dos empleados debido a la rapidez, precisión y flexibilidad de sus articulaciones.

2. Objetivo y especificaciones

El presente trabajo de estudio tiene como objetivo el diseño y realización de una aplicación de ensamblado de piezas que requiera la ayuda de un robot colaborativo de dos brazos además del propio operario y el uso del sistema de visión que incorpora en la pinza.

El primer paso de este proyecto es conocer el funcionamiento y programación del robot IRB 14000, YuMi de ABB. Para empezar a trabajar con este robot es necesario calibrar los resolvers de sus articulaciones y establecer así las restricciones cinemáticas de cada una de las articulaciones. Para ello se puede utilizar directamente una rutina ya programada.

Una vez calibrado el robot ya se pueden programar tareas sencillas de pick & places para conocer los movimientos y área de trabajo del robot para poder realizar correctamente la programación de la tarea de ensamblado, el cual es el principal objetivo del proyecto.

Como segundo objetivo se establece realizar una buena identificación de las piezas diseñadas a través del sistema de visión y que aporte flexibilidad al reconocimiento de dichas piezas.

En el caso de haber realizado dicha aplicación sería útil hacer comparativas de producción respecto a la cooperación del robot y el operario o, simplemente, el trabajo entre dos empleados en los aspectos de tiempo de ensamblaje o calidad del acabado de la pieza como objetivos secundarios.

Dadas estas necesidades se requiere el diseño de un sistema que:

- Realice una aplicación de ensamblado que incluya la ayuda de un operario y un robot colaborativo.
- Incorpore el sistema de visión integrado en el robot y aporte flexibilidad al reconocimiento de piezas.
- Aporte flexibilidad al ensamblado del producto final.
- El operario se pueda comunicar con el robot a través de los sensores que incorpora.

A parte de las especificaciones técnicas también se han de cumplir los siguientes requisitos:

- Poder trabajar en la misma tarea el robot y el operario conjuntamente certificando que la robótica colaborativa que está en auge en la industria.
- Formar un área de trabajo segura para el ser humano.
- La programación sea una tarea sencilla para poder ser realizada por usuarios con una formación básica que no estén especializados.
- Sistema viable económicamente.

3. Contexto teórico

El mundo de la robótica evoluciona de forma exponencial a medida que surgen nuevas tecnologías integrándose en la fase de producción del mundo industrial tanto de pequeñas como medianas y grandes empresas.

Es por ello por lo que se han de diferenciar las distintas generaciones de robots que han ido surgiendo a lo largo de los años.

3.1 Robots tradicionales vs robots colaborativos

Hasta ahora los robots que incorporaban las empresas en sus líneas de producción eran robots grandes, pesados y robustos dedicados a realizar una única tarea de forma repetitiva. Este tipo de robots son los llamados robots industriales tradicionales cuya definición según la norma ISO 10218-1 es:

“Manipulador reprogramable para usos múltiples, controlado automáticamente con tres o más ejes programables, ya sea fijo o móvil, para uso en aplicaciones de automatización industrial.” (ISO/TC 299 Robotics, 2011).



Ilustración 3: Robots industriales tradicionales

Debido a las exigencias del mercado y la necesidad de versatilidad y flexibilidad en distintas aplicaciones surge un nuevo tipo de robot llamados robots colaborativos cuya definición según la norma ISO 10218-2 es:

“Robot diseñado para la interacción directa con humanos en un espacio colaborativo definido.” (ISO/TC 299 Robotics, 2011).

Al formar parte de procesos industriales ambos robots tienen también similitudes en común como:

- Sistemas software que permiten opciones de integración con otras máquinas o robots a través de sistemas embebidos.
- Son capaces de realizar los procesos de producción sin empleados a cargo, aunque sí que puedan necesitar supervisión.
- A lo largo del tiempo sus procesos de automatización no sufren ningún cambio, se mantienen invariables.

Sin embargo, se ha de decir que cada tipo de robot es aplicado a situaciones distintas por lo que es necesario clasificarlos según las características del proceso industrial que se requiera (Universal Robots, 2018):

Tabla 1: Comparación robots tradicionales vs colaborativos

ROBOT TRADICIONAL	ROBOT COLABORATIVO
Gran volumen	Similar al trabajo de un humano
Soporta grandes cargas	Fácil de programar, instalar y reprogramar nuevas tareas
Precisión extrema	Colaboración con el trabajo humano
Alta velocidad de producción	Cambios mínimos del área de trabajo

En resumen, las ventajas que presenta un robot colaborativo respecto a un robot industrial tradicional son:

- No necesidad de una barrera de separación del espacio debido a la integración de sensores de colisión en sus brazos robóticos que detectan sobrecorrientes y la fuerza aplicada para frenar el movimiento del robot.
- Interfaces de programación más sencillas que se ajustan al perfil de un operario de nivel medio sin conocimientos avanzados de programación.

- Robots más pequeños y manipulables con nuevas morfologías que proporcionan funcionalidades más versátiles, incluido el cambio de herramienta en la muñeca del brazo robot.
- Menor inversión por parte del empresario debido que su coste inicial es menor, enfocados para empresas pequeñas y medianas.

Empresarialmente aparece una reestructuración del espacio de trabajo ya que, como se ha citado anteriormente, los nuevos robots sí que permiten el contacto físico con operarios donde se rigen unas normas de seguridad completamente distintas a la de los robots tradicionales, las cuales se explicarán más adelante.

Estas nuevas áreas de trabajo se pueden distribuir de la siguiente manera:

Tabla 2: Tipos de interacción de robots industriales

TIPO DE INTERACCIÓN	No simultáneo	Simultáneo
Mismo espacio		
Distinto espacio		

Una vez presentadas las diferencias respecto a los robots tradicionales tanto en características del producto como en el área de producción, se procede a explicar, más en concreto, a los robots colaborativos y toda la normativa vigente que atañe a los robots industriales.

3.2 Cobots

Puesto que los robots colaborativos son el tema principal de este proyecto se procede a detallar sus orígenes y funcionalidades.

3.2.1 Origen de los cobots

La creación de los cobots surge por dos profesores de la Universidad del Noroeste, J. Edward Colgate y Michael Peshkin, en 1996, definidos como un aparato y método de interacción física directa entre una persona y un manipulador general controlado por un ordenador. Sin embargo, no fue hasta 2004 la empresa *KUKA* lanzó el primer robot pionero, el modelo *LBR 3* (Wikipedia, 2019).

A partir de este momento otras grandes empresas de robótica diseñaron modelos distintos como el cobot *UR5* de la empresa *Universal Robots* en 2008, capaz de trabajar de forma segura con la gente o la empresa *Rethink Robotics* con el modelo *Baxter* en 2012.

3.2.2 Brecha de habilidades

La finalidad que se pretende alcanzar con los robots colaborativos es combinar las habilidades humanas y las capacidades de los robots permitiendo resolver en la industria aplicaciones que de otro modo necesitarían máquinas más complejas, caras y de hecho más peligrosas (Gömemann, 2017).

Todo esto se puede resumir en que los cobots ayudan a superar la brecha de habilidades presente en las empresas y así aumentar la producción, mejorar la calidad del producto final y reducir los costes ya que:

- Se reduce el número de operarios en tareas repetitivas o peligrosas.
- Mejora la cualificación de los trabajadores.
- Añade valor a los procesos de producción.



Ilustración 4: Cobots en el mundo empresarial

3.2.3 Niveles de cooperación

Cuando un humano y un robot trabajan conjuntamente se llama una operación de colaboración.

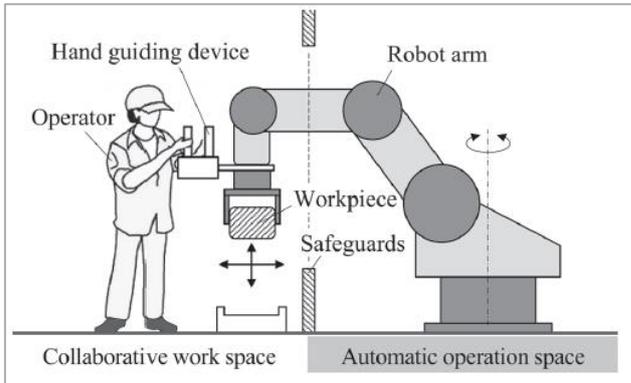


Ilustración 5: Espacio de trabajo común

Esta se define como un estado donde un robot diseñado intencionalmente trabaja en cooperación directa con una persona dentro de un espacio de trabajo común definido en el que se realizan tareas simultáneas (Tsuyoshi SAITO1, 2015).

Las posibles operaciones de colaboración entre un humano y un robot, cuyas medidas de seguridad se explican más adelante, son:

- a. Parada segura monitorizada
- b. Guiado manual
- c. Control de velocidad y separación
- d. Limitación de fuerza y energía

Estas tareas pueden integrar distintos niveles de cooperación según la relación que haya entre el humano y el robot.

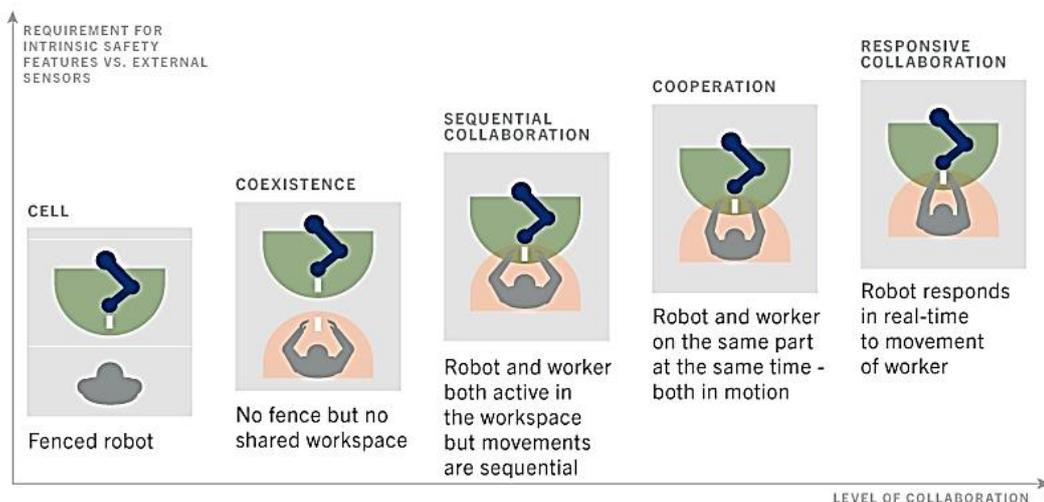


Ilustración 6: Niveles de cooperación entre un humano y un robot

Como se puede observar en el gráfico anterior existen 5 niveles:

- **Celda:** Este caso es el nivel más bajo de cooperación donde el robot, por seguridad, está vallado y protegido en un área.
- **Coexistencia:** En este segundo nivel el robot no se encuentra restringido en un área inaccesible, pero el trabajo del humano no se realiza en el mismo sitio que el trabajo del robot, sino que están separados.
- **Colaboración secuencial:** Este es el nivel de colaboración más típico en la industria actualmente, donde el robot y el trabajador actúan de forma activa en el mismo espacio de trabajo de forma secuencial.
- **Cooperación:** Este cuarto nivel hace referencia al trabajo que se realiza al mismo tiempo y sobre el mismo componente donde tanto el movimiento del operario como el del robot son a la vez.
- **Colaboración receptiva:** Este es el nivel más alto de cooperación entre un humano y el robot ya que existe una respuesta por parte del robot en tiempo real a los movimientos realizados por la persona.

3.1 Normativa

A continuación, se van a mencionar aquellas normas que afectan directamente en la realización del proyecto en relación al uso de los robots industriales, y más en concreto, a los robots colaborativos.

UNE-EN ISO 10218-1:2011: Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 1: Robots.

Esta primera norma especifica los requisitos de seguridad, pautas y medidas de protección que se aplican durante la manipulación de robots industriales con el fin de eliminar o reducir los riesgos laborales.

En esta norma no se contempla al robot como una máquina completa ni se considera el ruido como un peligro a contemplar (ISO/TC 299 Robotics, 2011).

UNE-EN ISO 10218-2:2011: Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 2: Sistemas robot e integración.

Esta normal complementa a la norma anterior descrita ampliando el aspecto de integración del robot donde se incluye el diseño, la fabricación, la instalación, operación, mantenimiento y desmantelamiento del sistema del robot y la información necesaria para ello. Además, describe los riesgos asociados a estas tareas identificadas como peligrosas para poder reducirlos.

En esta norma se considera al robot industrial como parte de un sistema de fabricación integrado (ISO/TC 299 Robotics, 2011).

ISO/TS 15066:2016: Robots y dispositivos robóticos. Robots colaborativos.

Debido al auge de los robots colaborativos se propone una norma concreta para ellos que complementa las dos anteriores donde se especifican los requisitos de seguridad en los sistemas industriales de los robots colaborativos y su entorno de trabajo, cuyos ejemplos se mostrarán más adelante (ISO/TC 299 Robotics, 2016).

Por otra parte, se pueden contemplar la norma **ISO 8373:2012** donde se definen los términos de uso relacionados con robots y dispositivos robóticos en el ámbito industrial o no industrial (ISO/TC 299 Robotics, 2012) y la norma **ISO 14539:2000** referente a la manipulación industrial de robots con objetos de agarre tipo *gripper* (ISO/TC 299 Robotics, 2000).

Además de las normas ISO existen, también, otras medidas de seguridad para las distintas operaciones de colaboración mencionadas:

a. Parada segura monitorizada

El robot debe tener una función de parada segura monitorizada y debe detenerse y permanecer en ese estado cuando una persona se encuentra en el espacio de trabajo común y puede reiniciar el movimiento cuando el operario se retire de dicho espacio.



Ilustración 7: Parada monitorizada

b. Guiado manual

Para la función de guiado manual el robot debe tener funciones que limiten los ejes y las velocidades de operación y su movimiento se ha de accionar a través de un mando de validación con velocidad reducida para no ocasionar ningún daño.



Ilustración 8: Guiado manual

c. Control de velocidad y separación

Para reducir el riesgo hace falta mantener una separación entre el operador y el robot. Además, la velocidad del robot ha de estar limitada y si se supera dicha distancia de seguridad se ha de reducir la velocidad del robot alterando su trayectoria.

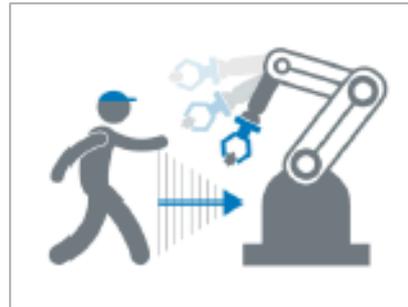


Ilustración 9: Control de velocidad

d. Limitación de fuerza y energía

Puesto que hay contacto físico hay que asegurar que no se originen lesiones o daños. Para ello, hay que determinar las partes del cuerpo humano que estén el peligro de colisión mediante un análisis de riesgo y hay que estimar las fuerzas, presiones y energías de impacto.

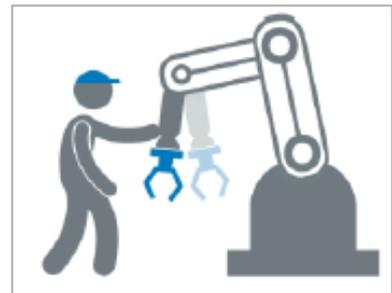


Ilustración 10: Limitación de fuerza

4. Estado del arte

Se procede a exponer el estado del arte referente a la robótica colaborativa actual atendiendo a las características que presenta cada robot con la intención de obtener soluciones alternativas y similares a la propuesta en este proyecto y observar las distintas tecnologías desarrolladas por parte de las ingenierías de cada empresa.

4.1 Robots de un brazo robótico

4.1.1 Universal Robots

Universal Robots es una de las empresas más sumergidas en la robótica colaborativa de los últimos años. Su triunfo en el mercado se debe a su oferta de robots de tamaño pequeño, flexibles y ligeros consiguiendo ser una herramienta muy fácil de manejar para los usuarios.

Esta empresa ha sacado al mercado tres modelos de robot de seis ejes y de distintos tamaños para adaptarse a diversas situaciones. Además, en breve anunciarán su nueva gama *e-Series* con los mismos modelos de robots, pero mucho más intuitivos, versátiles y con mayor gama de accesorios compatibles.

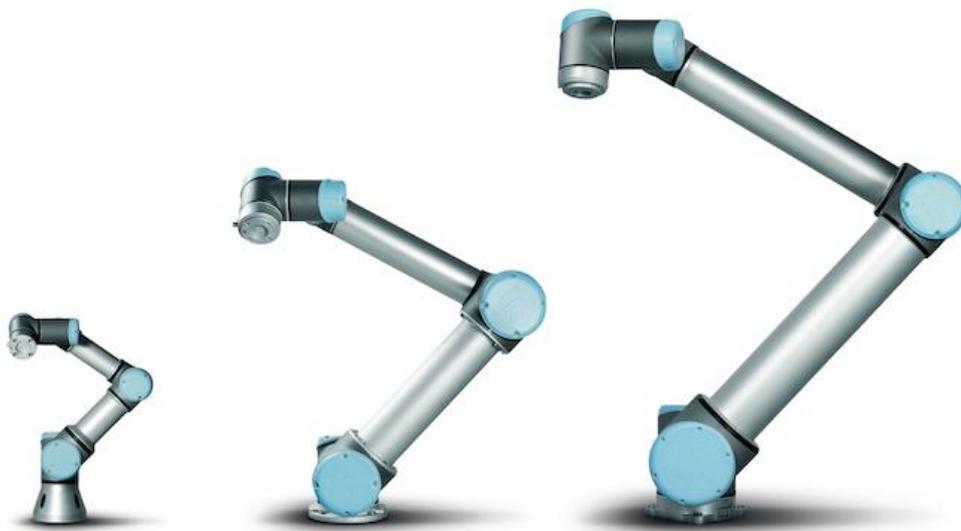


Ilustración 11: Gama de robots Universal Robots

A continuación, se procede a explicar cada uno de los modelos y sus distintas utilidades dentro de los procesos de fabricación de las empresas.

- **UR3:** Este robot es el más pequeño de toda la gama y el más ligero del mercado lo que le hace idóneo para tareas de montaje poco pesadas o bancos de trabajo automatizados. Sus características lo convierten en el mejor asistente para las aplicaciones de montaje, pulido, encolado y atornillado que requieran una calidad uniforme de la producción con menos del 1% de robots defectuosos producidos (Universal Robots, s.f.).
- **UR5:** Es el robot de tamaño mediano de la gama. Se caracteriza por ser muy flexible y ligero, pero a su vez ser capaz de automatizar tareas repetitivas y peligrosas. Sus características lo hacen idóneo para optimizar procesos colaborativos de poco peso como pick & place permitiendo a los operarios realizar otros procesos de producción, ya que según su director ejecutivo Geoff Escalette logra realizar el trabajo en 4 horas lo que un operario tardaría 2 o 3 días manualmente (Escalette, s.f.).
- **UR10:** Es el mayor brazo robótico industrial de toda la empresa, diseñado para tareas de gran envergadura donde priman la precisión y fiabilidad. Sus características lo hacen idóneo para procesos como el empaquetado, paletizado, montaje, pick & place o donde la distancia sea un factor clave. No por ello deja de ser económico o flexible en cuanto a su programación y sigue siendo un gran manipulador (Universal Robots, s.f.).

Con todo ello se realiza un resumen de las características propias de cada robot:

Tabla 3: Comparativa de características robots UR

	UR3	UR5	UR10
Huella	128 mm	149 mm	190 mm
Peso	11 kg	18.4 kg	33.5 kg
Alcance	500 mm	850 mm	1300 mm
Carga	3 kg	5 kg	10 kg
Rotación	360°	360°	360°

4.1.2 KUKA

Esta empresa lleva en el sector de los robots industriales desde sus comienzos orientada a una programación más avanzada y con muchas posibilidades. Con ello han desarrollado dos modelos de robots de la misma gama, LBR iiwa 7 y LBR iiwa 14 con siete articulaciones.

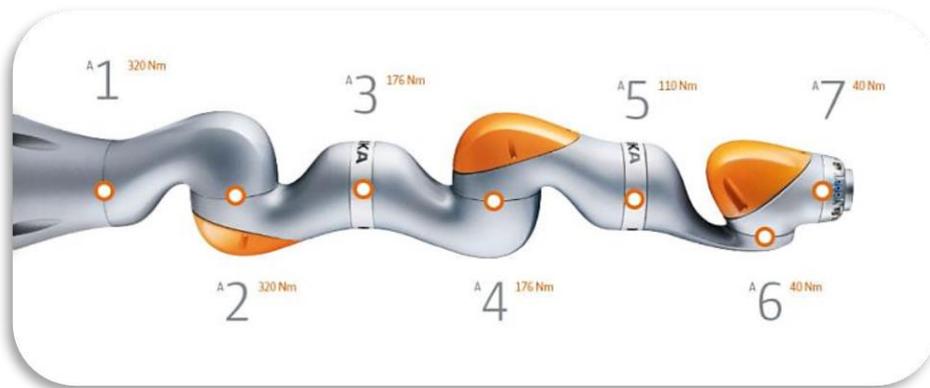


Ilustración 12: LBR iiwa 7 ejes

La gama *LBR* (robot de estructura liviana) *iiwa* (intelligent industrial work assistant) (KUKA, s.f.) permite tareas de montaje sensibles ya que sus robots incorporan una gran capacidad sensitiva dotándole de características como:

- **Tiempo mínimo de reacción:** Debido a sus sensores de esfuerzo articulados se detectan contactos inmediatamente reduciendo la fuerza y velocidad.
- **Gran capacidad de aprendizaje:** Tres modos de servicio y programación por simulación.
- **Sensitivo:** Detecta contornos y la posición de montaje correcta con una precisión de momento de eje del 2%.

Los dos modelos de la gama contienen las siguientes características:

Tabla 4: Tabla comparativa robots KUKA

	LBR iiwa 7	LBR iiwa 14
Carga	7 kg	14 kg
Alcance	800 mm	820 mm
Protección	IP54	IP54

4.1.3 FANUC

La empresa FANUC es pionera del control numérico con más de 4 millones de controles CNC y 550.000 robots instalados por lo que es conocida como el primer fabricante del mundo de sistemas de automatización de fábricas.

Es de esperar que o se quedase atrás en el área de la robótica colaborativa con 3 modelos distintos de robots de seis ejes.

- **CR-4iA:** Es el modelo más pequeño y compacto de la gama, con un brazo corto que permite realizar trabajos en áreas pequeñas siendo montado en la pared o de forma invertida (FANUC, s.f.).
- **CR-7iA/L:** También es de tamaño pequeño pero flexible encargado de tareas manuales y repetitivas, aunque se puede programar para flujos de producción que requieran calidad y fiabilidad constantes y debido a su largo alcance (modelo L) puede realizar tareas de carga, descarga y paletizado. Además, lleva incorporado un sensor de seguridad para detenerse y es compatible con accesorios como iRVision (FANUC, s.f.).



Ilustración 13: Modelos pequeños de cobots FANUC

- **CR-35iA:** Es el robot colaborativo con más capacidad de carga del mundo ideal para tareas como elevación de cargas pesadas y posicionamiento. Incluye sensores de fuerza y está rodeado por gomas para la seguridad del operario (FANUC, s.f.).

En la siguiente figura se muestra este último modelo de FANUC en colaboración con un operario:



Ilustración 14: Modelo CR-35iA de FANUC

Todos los robots pueden incluir distintos accesorios como sensores de visión o reconocimiento de área 3D siendo sus características las siguientes:

Tabla 5: Comparativa robots FANUC

	CR - 4iA	CR - 7iA/L	CR - 35iA
Carga	4 kg	7 kg	35 kg
Alcance	550 mm	717/911 mm	1813 mm
Ejes	6	6	6

Todos los modelos presentados de los distintos fabricantes de robots colaborativos solo contienen un brazo robótico, por lo que para realizar la aplicación de este proyecto harían falta dos robots ya que se requieren dos brazos robóticos. Este hecho hace que se complique la programación, la coordinación de los movimientos y que se pierda seguridad ya que no podrían evitar colisiones entre los dos brazos anticipadamente.

Por este motivo, en el siguiente apartado, se van a presentar dos soluciones de robots colaborativos con dos brazos robóticos.

4.2 Robots de dos brazos robóticos

4.2.1 TIAGo++

Este robot colaborativo con forma humanide de dos brazos robóticos de 7 articulaciones cada uno es fabricado por la empresa PAL Robotics. Su novedad es el movimiento coordinado entre los dos brazos.

Sus características principales son su lenguaje de programación basado completamente en ROS, es personalizable, robusto y puede expandir sus aplicaciones con sensores y dispositivos adicionales. (Viladomat, 2019).



Ilustración 15: Robot colaborativo de dos brazos TIAGo++

Además, también puede incluir una pantalla táctil en su torso para facilitar la creación de aplicaciones de la combinación de los dos brazos, con una extensión horizontal o vertical de ellos de 1.75 m y con un sistema de suspensión para bajar o subir durante sus movimientos. La diferencia con otros robots es que es un robot con una plataforma móvil pensado para la colaboración directa con humanos en áreas domésticas o industriales.

Por último, puede incluir un kit de inteligencia artificial para su desarrollo donde se pueden encontrar aplicaciones basadas en *deep learning*.

4.2.2 YuMi

El robot YuMi es una creación de ABB del año 2015 diseñado especialmente para el ensamblaje de piezas pequeñas en líneas donde robots y personas trabajan mano a mano en las mismas tareas.

Es un robot colaborativo que cuenta con dos brazos robóticos de siete articulaciones cada uno donde, además, se incluyen dos manos flexibles, sistemas de alimentación de piezas, localización de piezas mediante cámaras y un control robótico de última generación debido a su software flexible o las señales precisas de control de fuerza (ABB, s.f.).



Ilustración 16: Robot colaborativo YuMi de ABB

Las principales características del robot IRB 14000 (YuMi) son:

Tabla 6: Características YuMi de ABB

IRB 14000	
Carga	0.5 kg/brazo
Alcance	559 mm
Huella	399 mm * 497 mm
Precisión	0.02 mm
Protección	IP 30

Debido a sus dos brazos robóticos con 7 ejes permite alcanzar posiciones que los demás brazos robóticos no pueden y permite movimientos coordinados entre sí y conjunto al operador. Además, su sistema de visión integrado ayuda al reconocimiento de piezas y está específicamente diseñado para tareas de ensamblaje.

Por todas estas razones, el robot IRB 14000 es el más adecuado para la aplicación de ensamblado que se propone en este proyecto, por lo que se va a proceder a explicar el desarrollo práctico y detallar en profundidad las características de este robot y su control.

El diseño de las piezas se ha realizado a través del programa de diseño CAD de SolidWorks teniendo en cuenta que hay piezas que se van a imprimir en material 3D, permitiendo así su diseño completamente libre y adaptable a las necesidades.

Las piezas han sido impresas en las instalaciones del instituto de investigación ai2 de la UPV y se ha utilizado el modelo de impresora BCN3d Sigma R17. En concreto, se ha utilizado el filamento de impresión PLA muy usual en esta área que permite un rango más amplio de colores y, además, no emite gases nocivos durante su utilización en impresoras 3D. A continuación, se va a explicar cada una de ellas.

I. RUEDA

Esta es la pieza principal del montaje ya que es la que ha de rodar a través del eje sin causar fricción ni con él ni con las paredes laterales. Para conseguir este objetivo se han tenido en cuenta dos cosas: la holgura del diámetro del eje y el uso de casquillos para no permitir la fricción lateral. Estos casquillos van adheridos a la rueda mediante cinta de teflón no permitiendo su rotación interior y consiguiendo un encaje seguro.



Ilustración 18: Casquillos de fricción libre

Por otra parte, esta pieza ha sido específicamente diseñada para el agarre de las pinzas que integra el robot YuMi dotándole además de una simetría propia de una rueda que no intercepte con su estética exterior. Como se puede comprobar la matriz octagonal de los laterales de la rueda coincide perfectamente con el dedo del SmartGripper del robot, teniendo en cuenta siempre un margen de holgura.

En la siguiente ilustración se muestra el agarre entre el robot y la pieza diseñada:



Ilustración 19: Encaje de SmartGripper y la rueda diseñada

Además, cada una de las piezas diseñadas van acompañadas de su propio soporte ya que cada una se situará en un lugar específico de la mesa de trabajo dependiendo de la herramienta utilizada para su agarre o el uso del operario. Todos los soportes se sitúan de forma permanente en el espacio de trabajo por lo que se han diseñado sus correspondientes agujeros para fijarlos mediante tornillos.

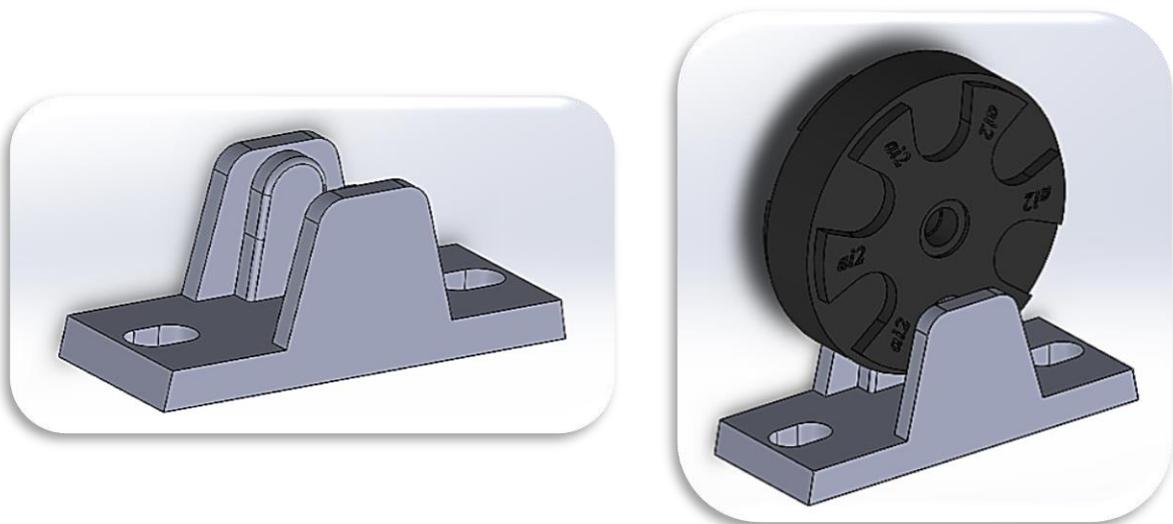


Ilustración 20: Diseño CAD del soporte de la rueda

II. TRIÁNGULOS LATERALES

La siguiente pieza a diseñar son las paredes laterales en forma triangular para adaptarse a la rueda ya diseñada y, además, permitir la unión con el carro.

Estas dos piezas son casi idénticas excepto en la apertura del eje ya que se ha de amoldar tanto al cabezal del tornillo de métrica seis, por una parte, como a la tuerca de palometa por la otra. Se ha escogido este tipo de tuerca ya que resulta más manejable a la hora de que el operario trate con ella.

En la siguiente imagen se muestra como encajan ambas piezas:

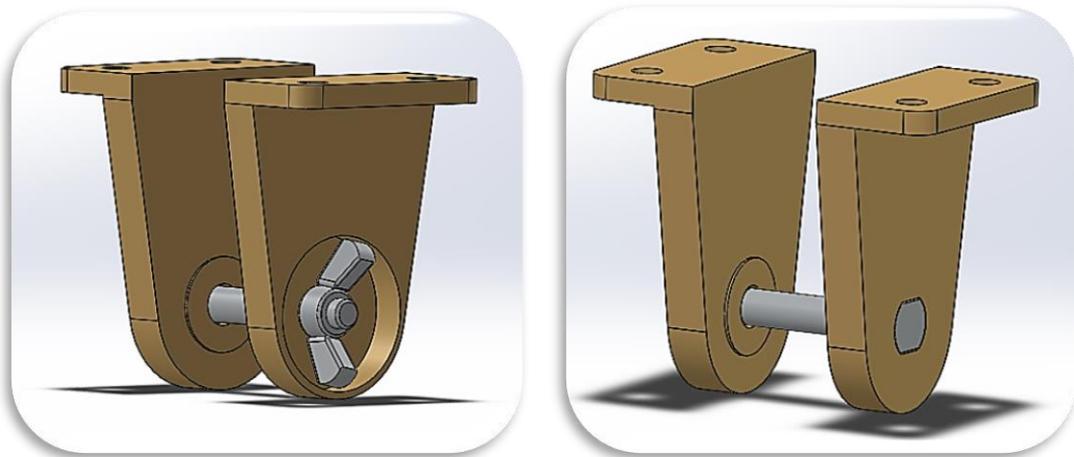


Ilustración 21: CAD piezas laterales

Como se ha comentado anteriormente se han creado soportes para fijar la posición de las piezas en el área de trabajo. Para esta pieza se ha optado por un diseño plano que permita su encaje horizontal con cierta holgura, ya que es la mejor posición para el tipo de agarre que utilizará el robot.

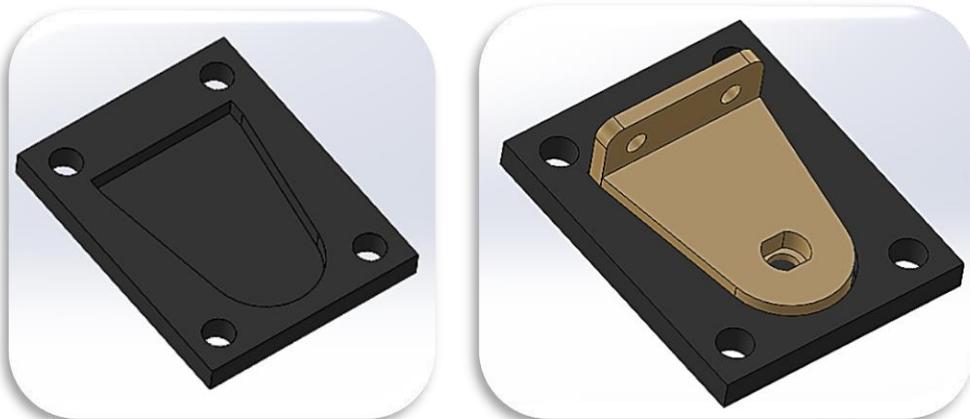


Ilustración 22: CAD soporte de piezas laterales

III. SOPORTE SUPERIOR

La última pieza a diseñar es la parte superior de la rueda que es atornillada a cada uno de las piezas laterales descritas en el apartado anterior.

Esta pieza es de forma rectangular con cuatro agujeros en las esquinas que permiten la unión. En este caso se han utilizado cuatro tornillos de métrica 5 con sus tuercas hexagonales y las arandelas correspondientes.

Para esta pieza no se ha creado ningún soporte ya que será la pieza donde se utilice la herramienta de visión para su localización y posterior agarre. El hecho de mantener su posición directamente en la mesa de trabajo permite un contraste de colores ideal para el procesamiento e identificación de la imagen.



Ilustración 23: CAD pieza superior

Cabe comentar que la unión de esta pieza con las dos piezas laterales son las que en su montaje necesitarán tanto al robot como la ayuda del operario, ya que ambas piezas laterales no tienen fricción con el eje lo que hace imposible que se queden fijas en una posición y caigan siempre debido a la gravedad.

A parte de las piezas que se han diseñado también se han utilizado otras piezas ya fabricadas. Estas piezas se han ido comentando a lo largo de las piezas explicadas, como los distintos tipos de tornillos y tuercas que se usan para las uniones. Todas ellas estarán situadas en la mesa de trabajo en su correspondiente recipiente para la correcta manipulación del operario.

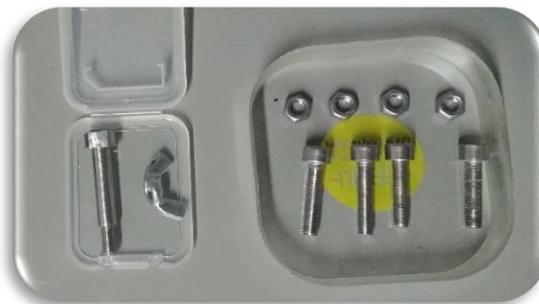


Ilustración 24: Disposición del material de manipulación del operario

5.2 Puesta en marcha YuMi

Una vez se ha diseñado la aplicación de ensamblado se dispone a la puesta en marcha del robot, ya que se trata de una nueva adquisición en el laboratorio de Industria 4.0 del ai2.

Primeramente, se ha de montar el robot con su correspondiente controlador, basado en IRC5 y dotarle de un área de trabajo estable y fija. De todo este trabajo previo se encargaron los técnicos de laboratorio cuyo resultado es el siguiente espacio de trabajo:



Ilustración 25: Área de trabajo en el laboratorio de Industria 4.0

El robot va conectado a la red eléctrica mediante el conector XP0 y lleva incorporado un interruptor de ON-OFF para su encendido. Estas conexiones se encuentran en la parte baja de la mesa donde también se encuentra el conector del Flexpendant.

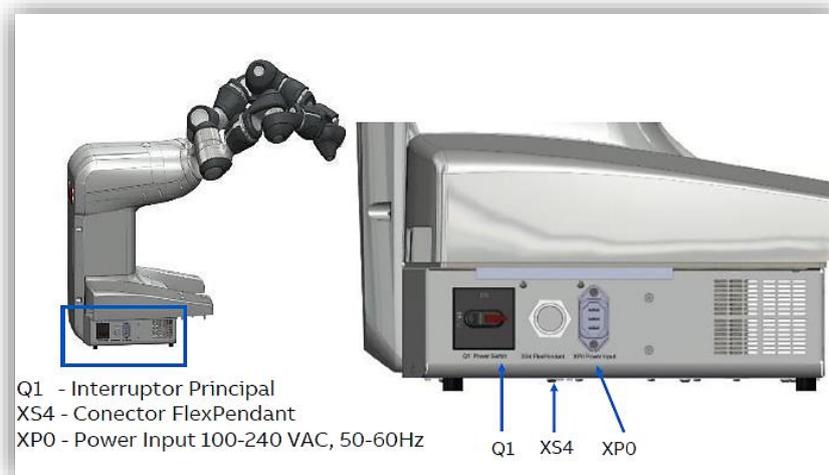


Ilustración 26: Alimentación del robot IRB 14000

A la otra parte del controlador se encuentran todas las conexiones de entradas digitales, analógicas o de red que puede soportar el robot. Algunas de ellas serán utilizadas y explicadas más hacia adelante.

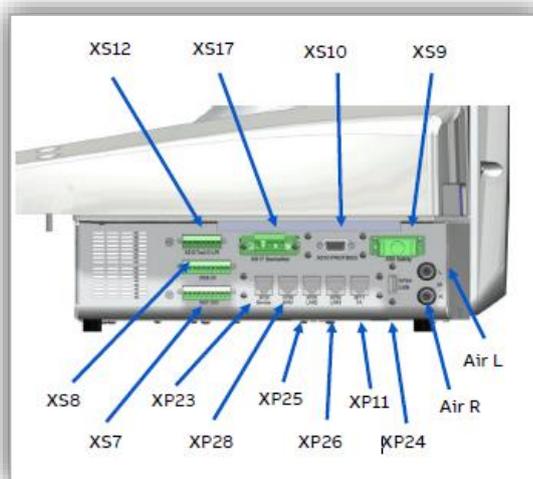


Ilustración 27: Conexiones periféricas

Las referencias se muestran en la siguiente tabla:

Tabla 7: Conexiones periféricas YuMi

XS12	4x4 E/S digitales para herramientas
XS17	DeviceNet Master/Slave
XS10	PROFIBUS
XS9	Señales de seguridad
XS8	8 señales de entrada digitales, 5mA
XS7	8 señales de salida digitales, 150mA
XP23	Service
XP28	WAN
XP25	LAN2
XP26	LAN3
XP11	PROFINET o EtherNet/IP
XP24	Puerto USB
Air L	Suministro de aire brazo izquierdo
Air R	Suministro de aire brazo izquierdo

Una vez el robot está en un lugar fijo ya se puede empezar a trabajar con él. Existen diversas maneras de programar el robot y de realizar sus movimientos, las cuales se explican a continuación.

5.2.1 Modos de programación

I. FlexPendant

Se trata de una unidad propia de programación móvil conectada al controlador del robot la cual permite todos los tipos de movimiento ya sea a través de un código de programación o rutina, con un movimiento guiado o a través del movimiento dirigido a partir del joystick que incorpora. En la siguiente imagen se pueden observar las opciones que ofrece esta unidad de programación:

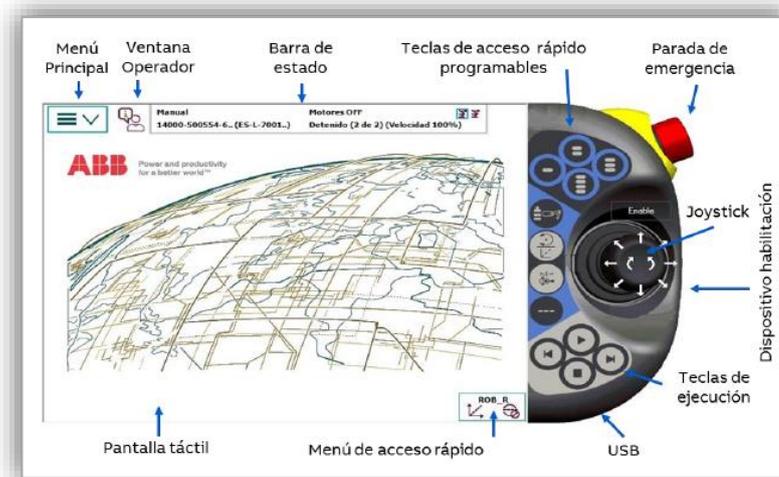


Ilustración 28: Unidad de programación FlexPendant

En cuyo menú principal se pueden escoger los tipos de movimiento, módulos que lleva incorporados, además de la configuración, ajustes o copia de seguridad propias.

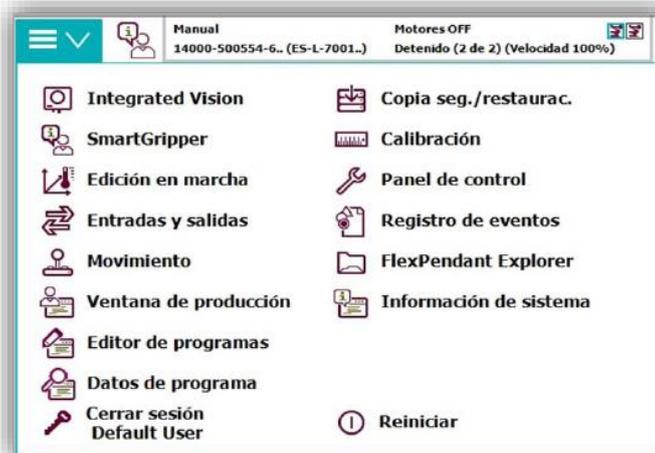


Ilustración 29: Menú principal FlexPendant

II. RobotStudio

Otra manera de programación es a través del software proporcionado por ABB, RobotStudio, en el cual se puede escoger cualquier robot fabricado por la empresa para realizar simulaciones, crear nuevos códigos o utilizar las extensiones como el módulo de visión. Para este proyecto se ha utilizado la versión ABB RobotStudio 6.07.01 de 32 bits, ya que la de 64 bits no incluye el módulo de visión.

Para poder utilizarlo con el YuMi se ha de conectar un cable tipo RJ45 al puerto de servicio del controlador del robot (Ilustración) y al puerto Ethernet del ordenador que se vaya a utilizar. Una vez conectado, en la pestaña de *Controlador* de la barra de menú superior, se selecciona “*Conexión en un click*” que detecta automáticamente el puerto. Además, hay otras formas de poder conectarse al controlador del robot como utilizar una red local a través de un router para que la conexión sea inalámbrica.

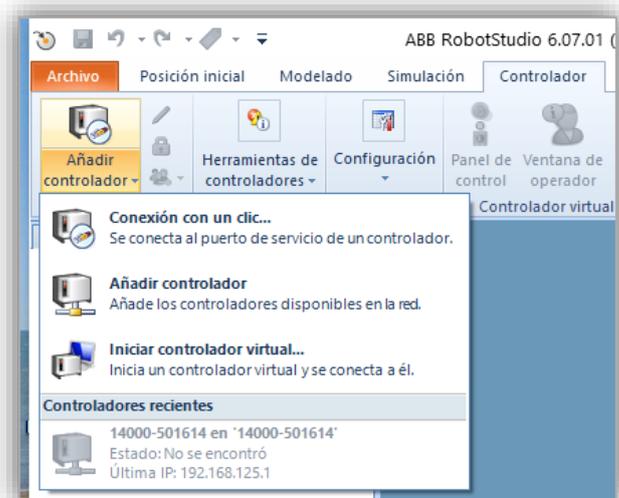


Ilustración 30: Conexión al controlador del robot

Como ya se ha comentado, a través de RobotStudio también se pueden realizar simulaciones del robot en la pestaña de *Simulación*. Para ello, previamente se ha de crear una estación con el modelo del robot eligiendo la opción de *Solución con estación y controlador de robot* y elegir el modelo IRB 14000.

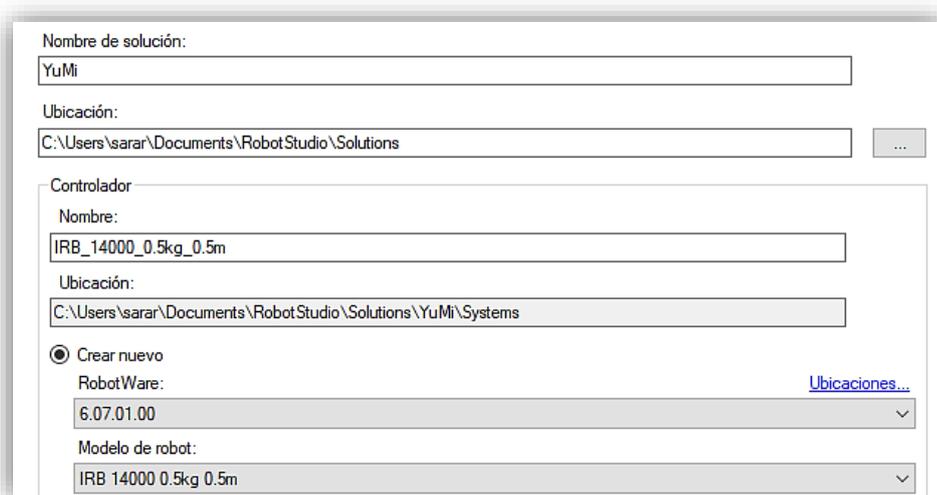


Ilustración 31: Creación de una estación en RobotStudio

Una vez creada la estación aparecerá una simulación del robot escogido. En esta pestaña se pueden simular movimientos dados por posiciones introducidas directamente por la persona, se puede simular un código o se pueden grabar los movimientos, entre otras tantas opciones.

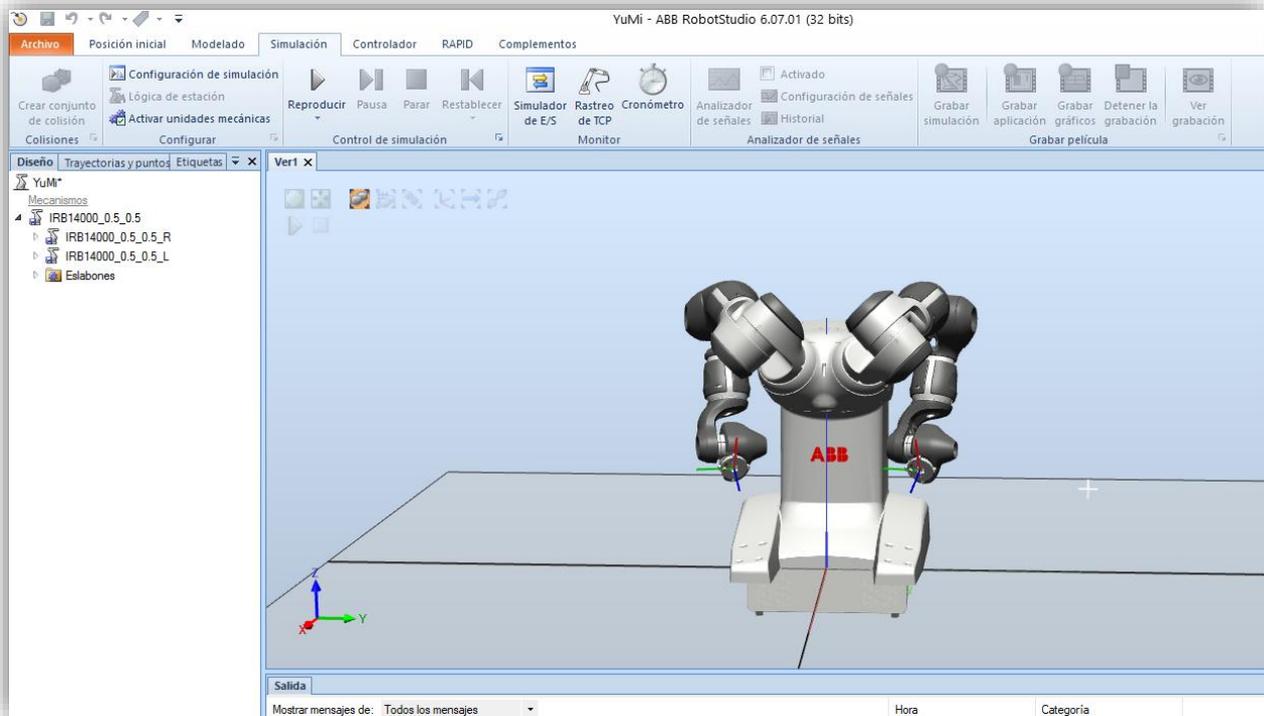


Ilustración 32: Simulación IRB 14000

Durante este proyecto no ha sido muy útil la función de la simulación ya que el robot simulado no incorpora la herramienta de SmartGripper que hace falta para el ensamblado por lo que no se puede realizar un ajuste fino de las posiciones. Sin embargo, sí se ha utilizado para detectar errores en la configuración de las articulaciones del robot por problemas de cinemática.

A través de RobotStudio también se puede simular la unidad de programación y poder utilizar el FlexPendant virtual sin tenerlo físicamente en tus manos.



Ilustración 33: FlexPendant Virtual

5.2.2 Tipos de movimiento

Se prosigue explicando las distintas formas que tiene el operario de realizar movimientos con el robot. Este modelo de ABB incluye cuatro formas de movimiento como son el movimiento manual, movimiento guiado, movimiento dirigido a través del joystick de la unidad de programación o movimiento programado.

I. Movimiento manual

Es el caso más sencillo de movimiento donde se desactivan los motores de las articulaciones, por lo tanto, quedan libres de ninguna fuerza y se pueden manejar manualmente hacia cualquier posición.

Para desactivar los motores se han de pulsar los botones de liberación de frenos del robot situados en su parte inferior.

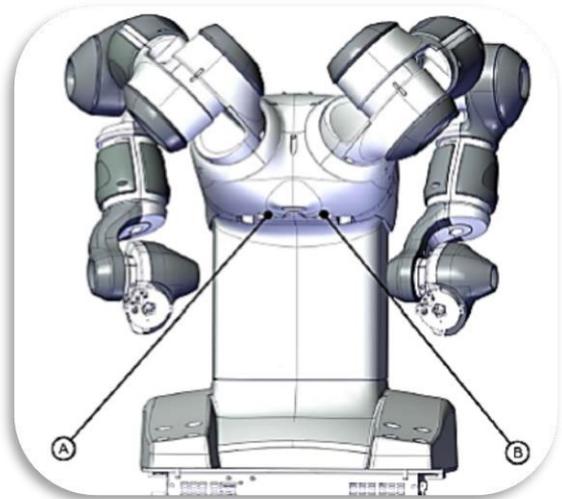


Ilustración 34: Botones liberadores de frenos

II. Movimiento guiado

Este tipo de movimiento es muy parecido al anterior debido a que también se ha de realizar manualmente. La principal diferencia es que el anterior movimiento libera completamente los frenos de las articulaciones haciéndolas caer por la gravedad ya que es la única fuerza aplicada. Sin embargo, en este caso las articulaciones se mantienen en la posición en la que están permitiendo ser movidas y modificadas por el operador en cualquier sentido. Esta opción está disponible a través de la pantalla de *Movimiento* del Flexpendant.

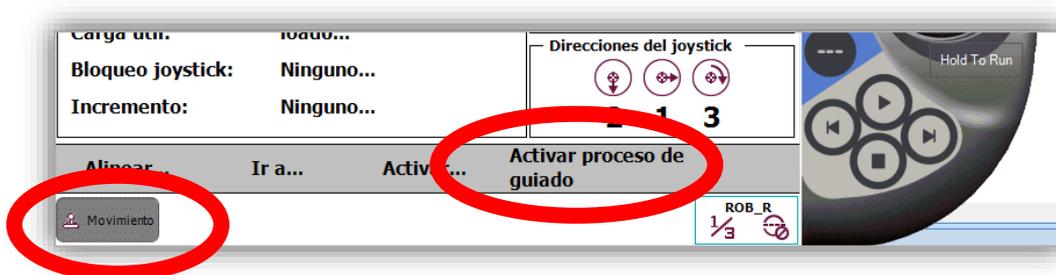


Ilustración 35: Movimiento guiado

III. Joystick

El movimiento a través del joystick proporciona la precisión que no se obtiene con un movimiento manual. Se trata de un joystick de 3 ejes con velocidad regulada, que cuanto más lo mantienes hacia una dirección más rápido va.

Se utiliza a través de la unidad de programación donde se pueden escoger diversas características del movimiento como:

- Modo movimiento: Puede ser lineal, por articulaciones, de orientación o movimiento respecto de la herramienta.
- Sistema de coordenadas: Puede ser respecto al mundo, a la base, al *workobject* al TCP. El modo de uso normal es respecto a la base.
- Posición: Se puede leer la posición actual de las articulaciones del robot respecto al sistema cartesiano X, Y, Z o respecto a los ángulos de Euler.

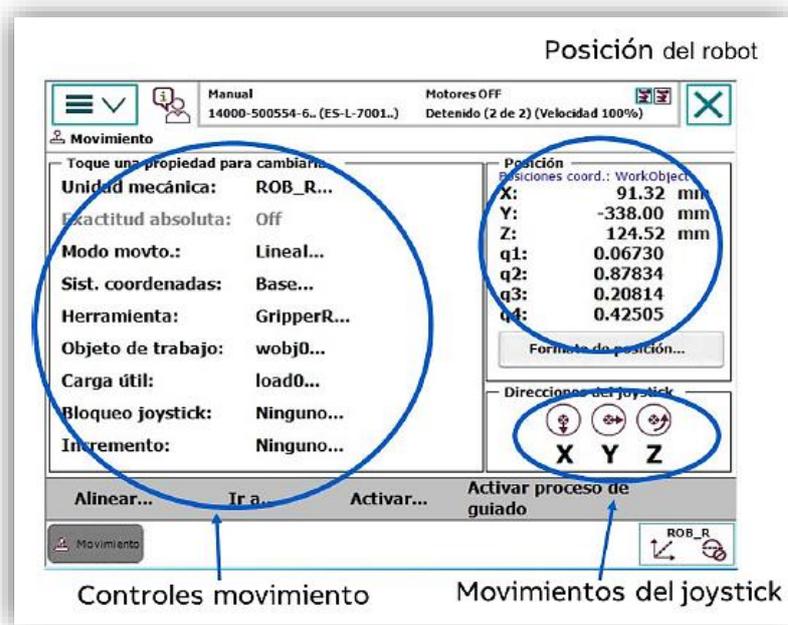


Ilustración 36: Pantalla de movimiento para el uso del joystick

IV. Movimiento programado

Por último, se encuentra el movimiento programado a través de un código de ejecución. Para ello, se han de escribir las instrucciones en el lenguaje RAPID e ir a la pestaña de *Ventana de producción* y pulsar el botón de *Play* de las teclas de ejecución de la ilustración 27.

5.2.3 Calibración

Antes de probar cualquier rutina o código se ha de realizar la calibración de las articulaciones. Para ello, simplemente se ha de seguir el manual que proporciona el fabricante, ABB y que lo podemos encontrar en el *Manual del Producto IRB 14000* (ABB AB, Robotics, 2015).

En el apartado 5 del manual se encuentra la información referente a la calibración. Este paso es necesario al principio de la instalación ya que hay que actualizar los cuentarrevoluciones de los resolvers que hay en cada articulación. Se ha utilizado el tipo de calibración estándar con el método de calibración fina con tecnología de sensor de efecto Hall donde el TCP del robot está vinculado a la base.

Para utilizar este método es necesario poner el robot en una posición predeterminada de manera que se puedan calibrar los 7 ejes a la vez de manera secuencial. En la siguiente imagen se muestra la posición adecuada de los ejes del robot donde deben coincidir todas las marcas con sus hendiduras.

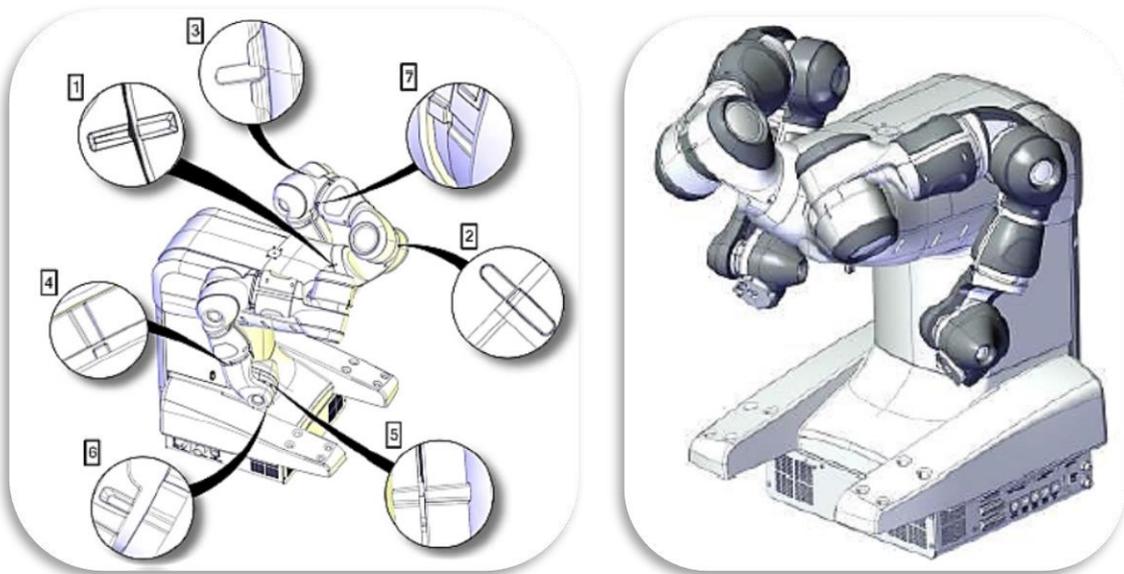


Ilustración 37: Posición de calibración

Para comenzar la calibración es necesario crear un nuevo programa en la pestaña de *Editor de programas* del FlexPendant y llamar a la rutina de *CalHall* con los motores en marcha. Esta acción se ha de realizar para los dos brazos por separado, tanto el derecho como el izquierdo.

Dentro de esta rutina de calibración hay que seleccionar el método de “Calibración Fina”, como se ha especificado anteriormente, marcando todas las articulaciones que se quieren calibrar al mismo tiempo.

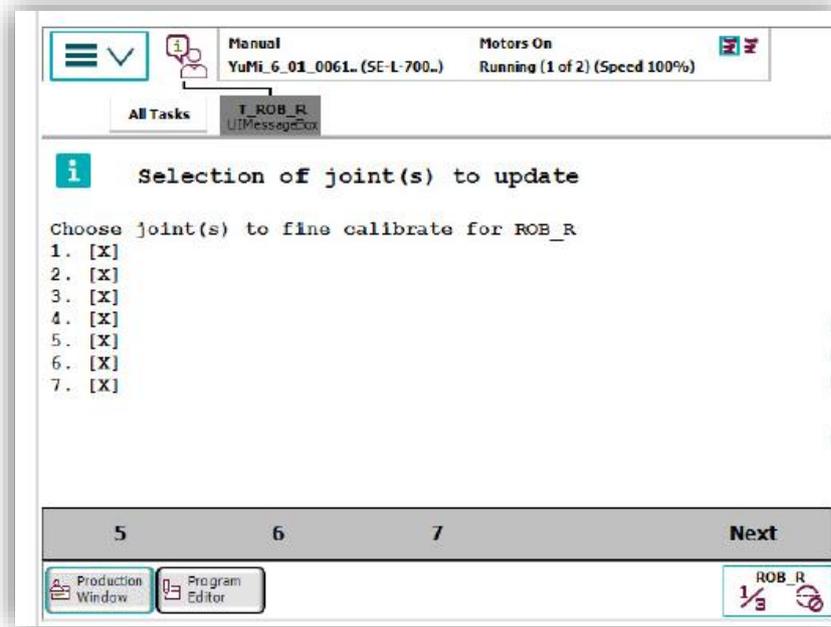


Ilustración 38: Selección de articulaciones a calibrar

Una vez seleccionadas las articulaciones a calibrar se ha de pulsar el botón de OK y el robot comenzará desde la primera articulación a realizar la calibración mediante movimientos rotatorios hasta hacer coincidir perfectamente las marcas y pasará a la siguiente articulación.

Tras la calibración se podrá comprobar que los ejes están en las posiciones que indica la siguiente tabla:

Tabla 8: Ángulos de calibración de los ejes

Eje	IRB 14000 ROB_R	IRB 14000 ROB_L
1	0°	0°
2	-130°	-130°
3	30°	30°
4	0°	0°
5	40°	40°
6	0°	0°
7	-135°	135°

Una vez definidos el rango de ángulos que abarca el movimiento del robot, también se ha de tener en cuenta otras características del robot, así como fuerza que se puede ejercer en las articulaciones o velocidades de las mismas.

En la siguiente tabla se muestran las fuerzas y pares que puede soportar el robot comparando su modo de funcionamiento normal con su modo de emergencia.

Tabla 9: Fuerzas y pares del robot en distintos modos:

Fuerza	Carga de resistencia (en funcionamiento)	Carga máxima (paro de emergencia)
Fuerza x	±89 N	±178 N
Fuerza y	±147 N	±294 N
Fuerza z	+380 ±140 N	+380 ±280 N
Par x	±101 Nm	±202 Nm
Par y	+14 ±98 Nm	+14 ±172 Nm
Par z	±61 Nm	±122 Nm

Por otra parte, durante la aplicación de ensamblado el robot será capaz de coger las piezas que se necesiten para el montaje final, por lo que se deben tener en cuenta los pares que se ejercerán sobre la muñeca, lo que atañe a los ejes 4, 5 y 6, teniendo en cuenta que la carga máxima es de 0.5 kg sumando el peso de la propia pinza.

Tabla 10: Par en las articulaciones correspondientes de la muñeca

Varlante de robot	Par de muñeca máx. ejes 4 y 5	Par muñeca máx. eje 6	Par máx. válido con carga
IRB 14000	0,64 Nm	0,23 Nm	0,5 kg

Por último, se muestran las velocidades que puede soportar el eje de cada articulación del robot, sabiendo que la superación de las mismas detendrá el robot al instante.

Tabla 11: Velocidades máximas de los ejes del robot

Varlante de robot	Eje 1	Eje 2	Eje 7	Eje 3	Eje 4	Eje 5	Eje 6
IRB 14000	180 °/s	180 °/s	180 °/s	180 °/s	400 °/s	400 °/s	400 °/s

5.3 Software

En este apartado se va a describir en detalle todo el programa utilizado para crear la aplicación de ensamblado, además de la explicación del software de los módulos que integra el robot para el reconocimiento de piezas.

En la siguiente figura se muestra el flujograma seguido en el módulo principal del programa para poder obtener una visión general de los movimientos del robot.

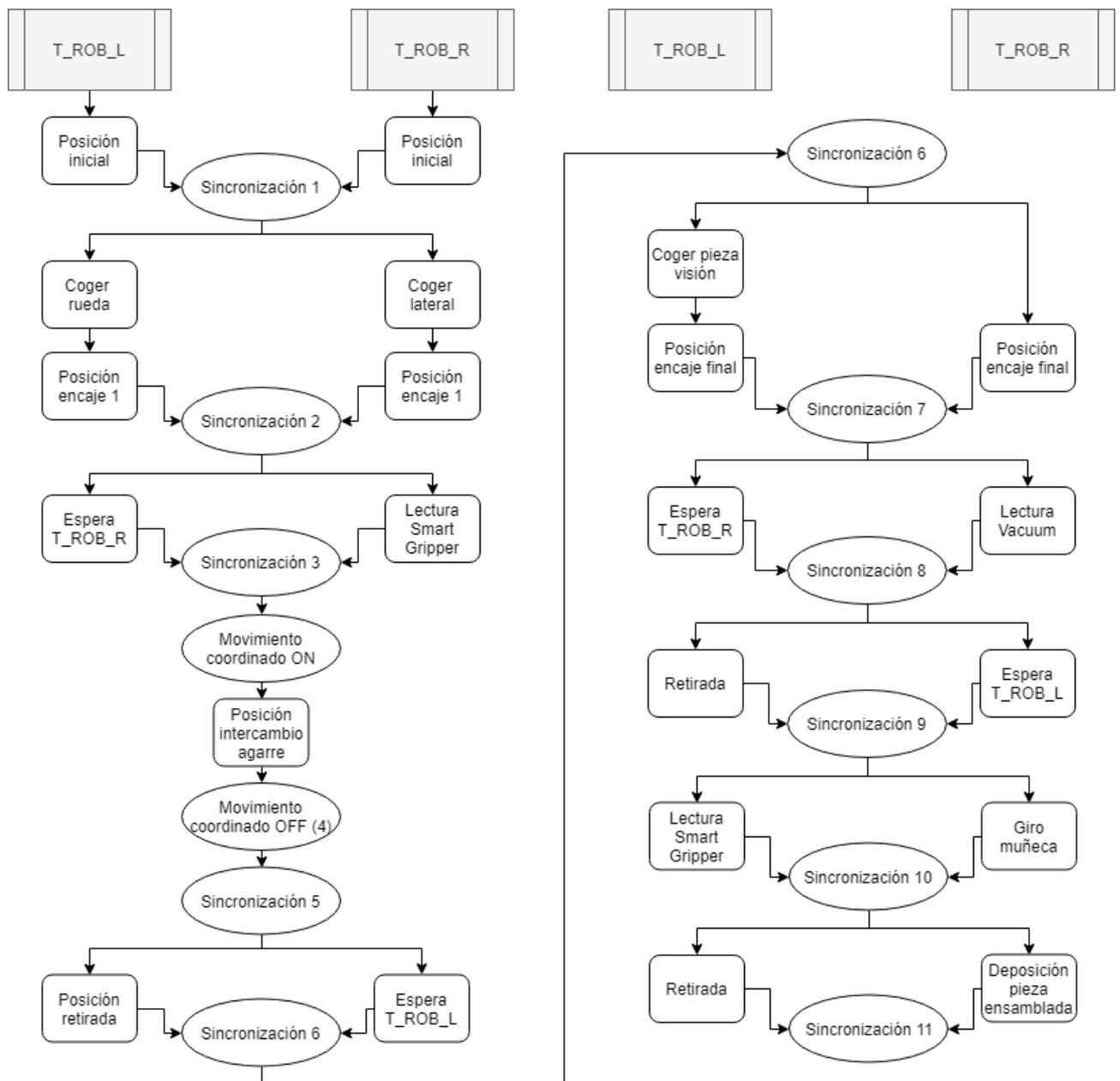


Ilustración 39: Flujograma del software

Previamente al módulo principal del programa es necesario definir las herramientas que se utilizan en el robot, así como la lista de tareas y variables.

Respecto a las herramientas, el robot tiene definida por defecto la herramienta con su TCP denominado *tool0*, pero al utilizar las pinzas de *Smart Gripper* es necesario importar estas nuevas herramientas ya que su TCP ha cambiado. El TCP hace referencia al punto central de la herramienta y es donde se dirigirá el robot cuando se le programe una posición. Por ello, se han creado los TCP de *GripperL* y *GripperR* para hacer referencia a ambos brazos.

```
PERS tooldata GripperR:=[TRUE,[[0,0,136],[1,0,0,0]], [0.23,[8.2,11.7,52],  
[1,0,0,0],0.00021,0.00024,0.00009]];
```

```
PERS tooldata GripperL:=[TRUE,[[0,0,136],[1,0,0,0]], [0.23,[8.2,11.7,52],  
[1,0,0,0],0.00021,0.00024,0.00009]];
```

Al tener dos tareas de ambos brazos la lista de tareas queda de la siguiente forma:

```
PERS tasks YuMi_App_task_list {2} := [{"T_ROB_L"}, {"T_ROB_R"}];
```

Y cada vez que se quiera definir una variable, como un movimiento sincronizado o una variable numérica, se ha de seguir la estructura:

```
VAR syncident YuMi_App_sync1;
```

```
VAR num presion;
```

Los movimientos sincronizados son necesarios para la cooperación entre los dos brazos del robot, dado que, como ambos tienen un código que seguir por separado lo seguirían secuencialmente sin tenerse en cuenta entre ellos mismos. Con la declaración de la sincronización, un brazo robótico ha de esperar a que el otro llegue a la parte del código donde esté la instrucción del mismo movimiento sincronizado.

Dado que el robot tiene dos brazos robóticos que se programan de forma independiente, se va a proceder a explicarlos por separado desarrollando los movimientos que realicen de manera secuencial y realizando las aclaraciones oportunas cuando interactúen ambas partes a la vez.

5.3.1 Módulo izquierdo

Se va a comenzar explicando el código seguidor por el brazo izquierdo del robot que sigue la parte del flujograma denominada T_ROB_L.

El primer paso será abrir la pinza para que esté disponible para un nuevo agarre y llevar el robot a la posición inicial. Para ello, usamos la instrucción *MoveJ* ya que se trata de un movimiento que no requiere linealidad y, por tanto, puede llegar a la posición deseada a través del movimiento de los ejes del robot.

```
!Posicion inicial
g_GripOut;
MoveJ p_LIni, v1000, fine, GripperL\WObj:=wobj0;
WaitSyncTask YuMi_App_sync1, YuMi_App_task_list;
```

Ilustración 40: Posición inicial T_ROB_L

Para definir las posiciones deseadas se mueve el robot activando el modo de guiado manual de la pantalla de *Movimiento* de la unidad de programación hacia el punto que se desea, y a partir de ahí, se realiza un ajuste fino con el joystick.

Estas posiciones se quedan guardadas como constantes de tipo *Robtarget* las cuales se definen al inicio del código.

```
CONST robtarget p_LIni:=[[495.60,21.70,370.94],[0.0297595,0.0516422,-  
0.701001,0.710665],[-1,-2,2,5],[167.681,9E+09,9E+09,9E+09,9E+09]];
```

Tras este primer movimiento, se realiza el agarre de la rueda a través del *Smart Gripper*. En todos los movimientos de agarre o encaje se realiza un paso previo de seguridad, es decir, un acercamiento a la posición real que se desea a través de un offset de dicha posición.

Este offset se realiza a través de la instrucción *MoveJ* a una velocidad de 1000mm/seg, mientras que la posición deseada se alcanza a través de la instrucción *MoveL*. Este movimiento es lineal por lo que el TCP de la herramienta se mueve acorde al sistema de coordenadas X, Y, Z linealmente y siempre con una velocidad media para que el sistema sea seguro.

En la siguiente imagen se muestran los pasos utilizados para tratar la pieza:

```
!Posicion de seguridad y agarre de la rueda
MoveJ Offs(p_LAgarre1,0,0,180), v1000, fine, GripperL\WObj:=wobj0;
MoveL p_LAgarre1, v500, fine, GripperL\WObj:=wobj0;
g_GripIn;
GripLoad load1;
MoveL Offs(p_LAgarre1,0,0,180), v500, fine, GripperL\WObj:=wobj0;
```

Ilustración 41: Posición de agarre primera pieza

Para el uso del Smart Gripper se utilizan las instrucciones de *g_GripOut* si se quieren abrir las pinzas y *g_GripIn* si se quiere agarrar una pieza. Este movimiento cerrará las pinzas hasta colisionar con el objeto ejerciendo una fuerza de agarre sobre él. Esta fuerza se puede programar por código o definir en el FlexPendant donde hay un módulo especial para esta herramienta.

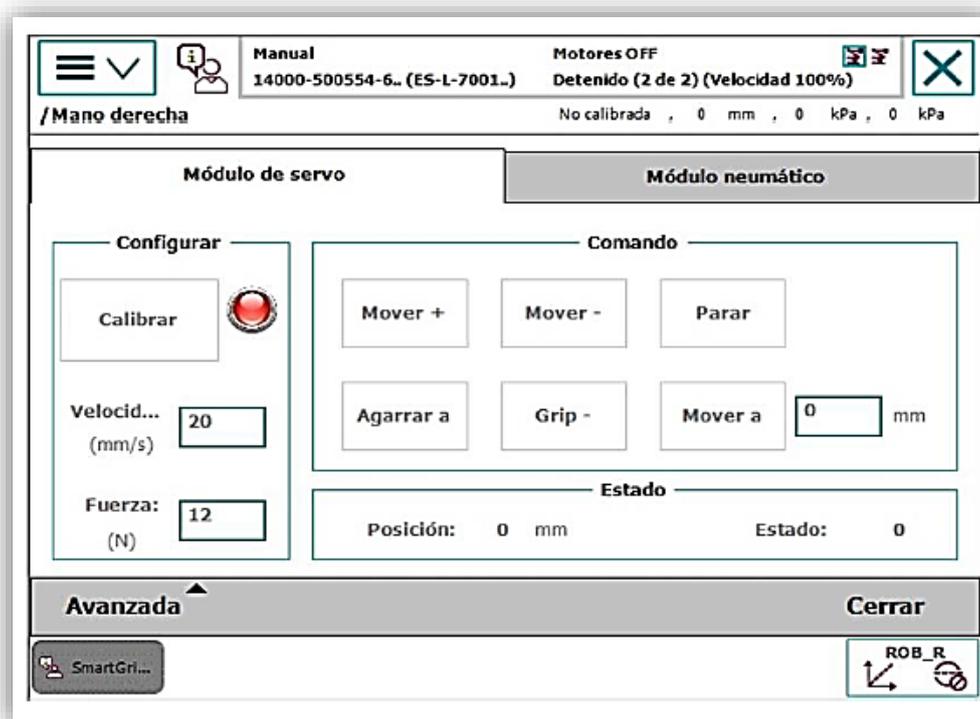


Ilustración 42: Pantalla Smart Gripper

Además, previamente a usar cualquier instrucción que implique esta herramienta se han de calibrar las pinzas para definir la posición cero de las mismas, por lo que se recomienda realizar la calibración una vez encendido el robot para poder empezar a

trabajar con él. Para calibrarlo llevamos las pinzas a la posición cerrada y pulsamos en *Calibrar* donde se el botón cambiará al color verde.

En el código de la Ilustración 41 también se puede observar la instrucción *Gripload* que hace referencia a una carga, *load1*. Esta instrucción se usa para indicarle al robot que ha cogido una carga, de manera que tenga en cuenta su peso y sus características para realizar la cinemática del robot. Esta carga se ha de definir previamente tal y como indica el *Manual del producto de la Pinza del IRB 14000* (ABB, 2018).

```
PERS loaddata load1:=[0.11,[6.8,-0.6,-27.4],[1,0,0,0],0,0,0.01];
```

Se prosigue con llevar el movimiento hacer la posición de encaje. En este caso no hace falta posición de seguridad ya que es el brazo derecho el que se aproxima hacia el izquierdo, considerando este como fijo.

```
!Posicion de encaje de la rueda y los triangulos  
MoveJ p_LEncaje1, v1000, fine, GripperL\WObj:=wobj0;  
WaitSyncTask YuMi_App_sync2, YuMi_App_task_list;
```

Ilustración 43: Programación de la primera posición de encaje

El robot ha de esperarse hasta el próximo movimiento sincronizado sin hacer ninguna tarea ya que es el momento en que el operario encaja las piezas oportunas que se explicarán en el código del brazo robótico derecho.

Cuando se realiza la sincronización se procede a realizar un movimiento coordinado de los brazos a través de la instrucción *SyncMoveOn*. Este se diferencia de los movimientos anteriores en que uno de los brazos tiene como objeto de trabajo la posición del otro, obteniendo así las coordenadas de las posiciones como un offset respecto al otro brazo robótico. En este caso, se considera como el brazo líder al izquierdo, por lo que, el brazo derecho dependerá de él.

```
!Movimiento sincronizado para posicion de seguridad de agarre  
SyncMoveOn YuMi_App_sync3, YuMi_App_task_list;  
MoveJ pCoord1_L10\ID:=10, v1000, fine, GripperL\WObj:=wobj0;  
SyncMoveOff YuMi_App_sync4;
```

Ilustración 44: Movimiento coordinado

En el siguiente paso el brazo derecho procede a enganchar las piezas que tiene agarradas el brazo izquierdo de manera que este las suelte y quede libre.

```
!Cambio de pinza de agarre
WaitSyncTask YuMi_App_sync5, YuMi_App_task_list;
g_GripOut;
MoveL Offs(pCoord1_L10,0,100,0), v500, fine, GripperL\WObj:=wobj0;
WaitSyncTask YuMi_App_sync6, YuMi_App_task_list;
```

Ilustración 45: Cambio de agarre al brazo derecho

La siguiente línea de código hace referencia al reconocimiento de la pieza superior por visión cuyo procedimiento se explicará en el apartado *5.3.3 Reconocimiento por visión*. Suponiendo que la pieza superior ya ha sido recogida por la pinza izquierda, se procede a realizar el último encaje entre piezas de la aplicación de ensamblaje.

```
!Posición de encaje final
MoveJ Offs(p_LEncaje2,0,220,0), v1000, fine, GripperL\WObj:=wobj0;
WaitSyncTask YuMi_App_sync7, YuMi_App_task_list;
MoveL p_LEncaje2, v100, fine, GripperL\WObj:=wobj0;
```

Ilustración 46: Ir a la posición de encaje final

Una vez ambos brazos están en la posición de encaje han de esperar a que el operador realice la unión de las piezas, la cual se comentará en el módulo del brazo derecho. Una vez se complete la espera, el brazo izquierdo cede el agarre del soporte superior y se retira hacia atrás para permitir el movimiento de la pieza. Para que no arrastre la pieza consigo, se realiza una parada de 1 segundo.

```
!Retirada
WaitSyncTask YuMi_App_sync8, YuMi_App_task_list;
g_JogIn;
WaitTime 1;
MoveJ Offs(p_LEncaje2,-50,0,0), v50, fine, GripperL\WObj:=wobj0;
WaitSyncTask YuMi_App_sync9, YuMi_App_task_list;
```

Ilustración 47: Retirada brazo izquierdo

Mientras el operario completa la unión de las piezas de la parte inferior, el brazo robótico izquierdo sirve como comunicación entre el robot y la persona. Para ello, se crea un bucle de lectura de la posición de la pinza izquierda del robot con la condición de continuar con el hilo del programa si se supera la posición de 18mm. Hay que tener en cuenta que las pinzas han de estar siempre calibradas. En el momento en que el operario abra las pinzas del robot, esta lectura será mayor y se continuará con el movimiento tras 1 segundo de espera por seguridad.

Esta comunicación proporciona un nivel de interacción humano-robot de nivel 5, máximo nivel, ya que la reacción del robot al movimiento de la persona se realiza en tiempo real consiguiendo una *colaboración receptiva* comentada anteriormente en el apartado *3.2.3 Niveles de cooperación*.

```
!Segunda espera del operador mediante las pinzas
posOpen:= g_GetPos();

WHILE posOpen < 18 DO
    posOpen:= g_GetPos();
ENDWHILE

WaitTime 1;
WaitSyncTask YuMi_App_sync10, YuMi_App_task_list;
```

Ilustración 48: Comunicación mediante la pinza izquierda

Concluyendo el módulo de programación izquierdo se realiza un movimiento de retirada del brazo robótico a una posición cercana a la inicial y se realiza la última sincronización de espera mientras el brazo derecho completa su respectiva tarea.

```
!Fin de programa
MoveJ Offs(p_LIni,100,100,0), v1000, fine, GripperL\WObj:=wobj0;
WaitSyncTask YuMi_App_sync11, YuMi_App_task_list;
```

Ilustración 49: Finalización T_ROB_L

5.3.2 Módulo derecho

El próximo punto a desarrollar es la parte del flujograma referenciada al código seguido por el brazo derecho del robot, denominada T_ROB_R.

El primer paso de sendos brazos, como se ha citado anteriormente, es ir hacia la posición inicial y esperar a que ambos lo hayan completado.

```
!Posicion inicial
MoveJ p_RIni,v1000,fine,GripperR;
WaitSyncTask YuMi_App_sync1, YuMi_App_task_list;
```

Ilustración 50: Posición inicial brazo derecho

Para este brazo, la primera pieza a reconocer es el lateral izquierdo en forma de triángulo explicando en el apartado 5.1 Aplicación de ensamblaje. En esta ocasión, el agarre se va a realizar a través del módulo de vacío que incluye dicho brazo.

En particular, se ha utilizado el modo de vacío de manera que el robot se acercará a la posición donde esté la pieza en la mesa de trabajo, siempre



Ilustración 51: Módulo de vacío IRB 14000

con un movimiento anterior y posterior de seguridad, y la succionará a una presión de 104 kPa manteniéndola sujeta a él hasta que se indique lo contrario. El código que se ha programado para ello es el siguiente:

```
!Posicion de seguridad y succion del triangulo
MoveJ Offs(p_RAgarre1, 0,0,35),v1000,fine,GripperR;
MoveL p_RAgarre1,v500,fine,GripperR;
g_VacuumOn1;
WaitTime 1;
MoveL Offs(p_RAgarre1, 0,0,35), v500, fine, GripperR\WObj:=wobj0;
```

Ilustración 52: Código succión de pieza lateral

El módulo de aire se puede editar y controlar a través del FlexPendant donde, también, tiene una pantalla para su uso. En ella se puede editar el modo eligiendo entre succión o soplado, además de comprobar a la presión que actúa cada modo.

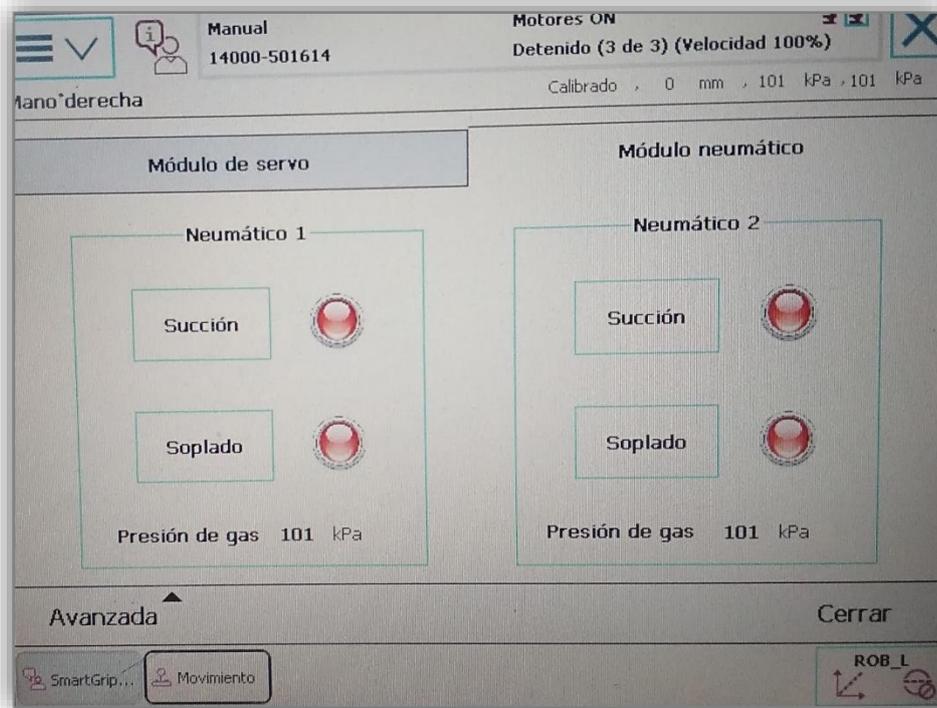


Ilustración 53: Pantalla del módulo neumático en FlexPendant

Se prosigue con el encaje del lateral derecho con la rueda del brazo izquierdo. Como se ha explicado en el apartado anterior, es el brazo derecho el que se aproxima a una distancia de seguridad y, seguidamente, encaja la pieza.

```
!Posición de seguridad y encaje de la rueda y los triangulos
MoveJ Offs(p_REncaje1, 0,-60,0), v1000, fine, GripperR\Wobj:=wobj0;
WaitSyncTask YuMi_App_sync2, YuMi_App_task_list;
MoveL p_REncaje1, v500, fine, GripperR\Wobj:=wobj0;
```

Ilustración 54: Código referente al primer encaje de piezas

Es el momento de actuación del operador donde su tarea es encajar a la parte derecha el tornillo de 7mm que hace función de eje principal a través de la rueda, sus casquillos y las dos piezas laterales, atornillando al lado izquierdo la tuerca de palometa.

El primer ensamblado de las piezas resulta de la siguiente forma:

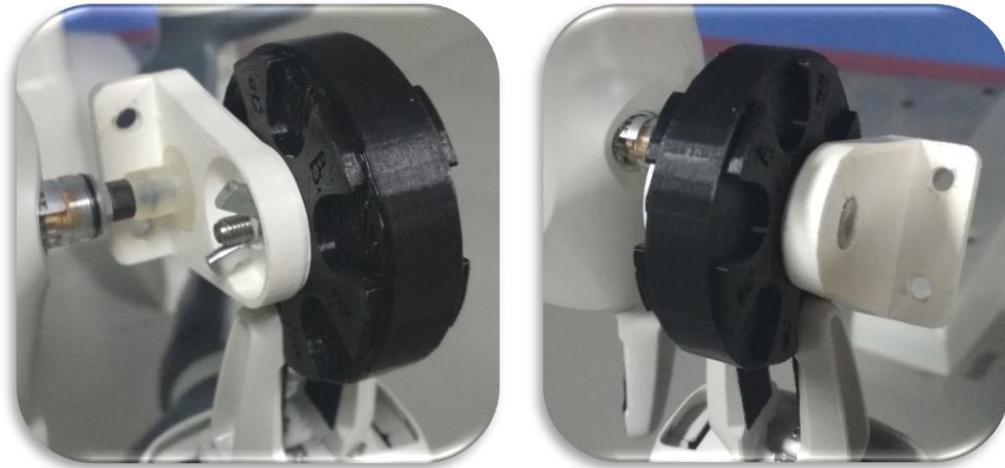


Ilustración 55: Primera parte de la aplicación de ensamblado

Una vez está todo encajado, el operario da la instrucción al robot para que pueda seguir con el código programado. Esta instrucción se realiza a través de la reducción de la distancia entre los dedos de la pinza del brazo derecho, ya que es la que está disponible. Esta función es análoga a la descrita en el módulo izquierdo con la diferencia de que esta vez la reacción del robot se pretende a través del cierre de la pinza y no de la apertura, por eso en el bucle de lectura la condición de salida es que la posición sea inferior a 24mm.

Una vez realizada la señal, el brazo derecho queda libre, el módulo de succión de aire se apaga y el robot se retira.

```
!Espera del operador mediante las pinzas
g_JogOut;
posClose:= g_GetPos();

WHILE posClose > 24 DO
    posClose:= g_GetPos();
ENDWHILE

!Retirada
WaitTime 1;
g_VacuumOff1;
MoveL Offs(p_REncaje1, 0,-100,0), v500, fine, GripperR\WObj:=wobj0;
```

Ilustración 56: Comunicación operario mediante SmartGripper

Posteriormente, se ejecuta el movimiento coordinado expuesto en el apartado anterior en que el brazo derecho depende del izquierdo por lo que la instrucción tiene que ir referenciada mediante el código de identificación ID:=10. Además, se ha creado un nuevo objeto de trabajo, *wobjCoord*, de las siguientes características:

TASK PERS wobjdata

```
wobjCoord:=[FALSE,FALSE,"ROB_L",[[0,0,0],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
```

Una vez terminada la sincronización de ambos brazos se procede al agarre de la pieza por parte del brazo derecho, quedando el izquierdo libre para el siguiente movimiento:

```
!Movimiento sincronizado para posicion de seguridad de agarre  
SyncMoveOn YuMi_App_sync3, YuMi_App_task_list;  
MoveJ pCoord1_R10\ID:=10, v1000, fine, GripperR\WObj:=wobjCoord;  
SyncMoveOff YuMi_App_sync4;  
  
!Cambio de pinza de agarre  
MoveL p_RAgarre2, v500, fine, GripperR\WObj:=wobj0;  
g_GripIn;  
WaitSyncTask YuMi_App_sync5, YuMi_App_task_list;  
WaitSyncTask YuMi_App_sync6, YuMi_App_task_list;
```

Ilustración 57: Movimiento coordinado brazo derecho

La última parte de la aplicación es dirigirse hacia la posición de encaje final donde el robot ha de esperar a la nueva señal del operario.

```
!Posicion de seguridad y encaje final  
MoveJ Offs(p_REncaje2, 0,-60,100), v1000, fine, GripperR\WObj:=wobj0;  
WaitSyncTask YuMi_App_sync7, YuMi_App_task_list;  
MoveJ p_REncaje2, v500, fine, GripperR\WObj:=wobj0;
```

Ilustración 58: Posición de ensamblado final

Dado que ambas pinzas están ocupadas, la comunicación entre el operario y el robot se realizará a través del módulo de aire. Para eso, también se programa un bucle de lectura de la presión de soplado del brazo derecho, teniendo en cuenta que la presión en condiciones normales durante la aplicación es de 104 kPa. Esta vez, el movimiento del operario será presionar con el dedo la ventosa de aire tamponando su salida de

manera que la presión aumentará superando la condición del bucle de lectura. En este momento habrá una espera de seguridad del 1 segundo prosiguiendo con la retirada del brazo.

```
!Primera espera del operador mediante soplado
g_BlowOn1;
presion1 := g_GetPressure1();

WHILE presion1 < 108 DO
    presion1:= g_GetPressure1();
ENDWHILE

WaitTime 1;
g_BlowOff1;
WaitSyncTask YuMi_App_sync8, YuMi_App_task_list;
```

Ilustración 59: Comunicación a través del módulo de aire

El siguiente movimiento se trata de un giro de muñeca de 180° de maneja que se puedan introducir, de forma sencilla, los tornillos en los agujeros de la parte inferior de la pieza quedando, esta, totalmente ensamblada.

```
!Giro de la posicion de encaje
WaitSyncTask YuMi_App_sync9, YuMi_App_task_list;
MoveL p_REncaje3, v1000, fine, GripperR\WObj:=wobj0;
WaitSyncTask YuMi_App_sync10, YuMi_App_task_list;
```

Ilustración 60: Giro de muñeca del brazo derecho

El último movimiento será depositar la pieza ya totalmente construida sobre la mesa de trabajo y sincronizarse con el brazo izquierdo de manera que puedan comenzar el programa, de nuevo, a la vez.

```
!Deposicion de la pieza ensamblada
MoveJ p_RFinal, v1000, fine, GripperR\WObj:=wobj0;
WaitSyncTask YuMi_App_sync11, YuMi_App_task_list;
```

Ilustración 61: Deposición pieza final

5.3.3 Reconocimiento por visión

Este apartado del software hace referencia al programa utilizado durante el reconocimiento de las piezas a través de la visión. Se va a hacer uso de la cámara Cognex A3 que lleva integrada la pinza del robot. Las características de dicha cámara se muestran en la siguiente tabla:

Tabla 12: Especificaciones cámara Cognex A3

Descripción	Datos
Resolución	1,3 megapíxeles
Objetivo	6,2 mm f/5
Iluminación	LED integrado con intensidad programable
Motor de software	Alimentación desde Cognex In-Sight
Software de programación de aplicaciones	ABB Integrated Vision o Cognex In-Sight Explorer

Previamente a la programación del código se ha de establecer un área de trabajo acorde a una aplicación donde se incluye la parte de visión ya que ha de tener unas características de iluminación adecuadas, por lo que el operario debe tener en cuenta su posición a la hora de realizar la captura de la imagen para no proyectar sombras.

El primer paso para poder emplear la cámara es el enfoque del objetivo, que ha de realizarse de forma manual desmontando la pinza del robot tal y como indica su manual *Pinza del IRB 14000*. Este enfoque dependerá de la altura a la cual se realice la fotografía, por lo que hay que tener claro cuál será la posición de la pieza en la mesa de trabajo.



Ilustración 62: Cámara Cognex A3

Seguidamente, se procede a efectuar la calibración de la cámara. Este paso es el más importante a lo largo del proceso de visión, ya que de él dependerá la afinidad con la que se calcule la posición de la pieza.

Esta calibración se realiza con la ayuda de un patrón de calibración tipo *checkboxoss* de 5mm con origen. Se ha escogido este tamaño de patrón debido a que las piezas con las que va a manipular el robot son pequeñas, por lo que se ha de tener la suficiente resolución como para detectarlas de forma óptima.

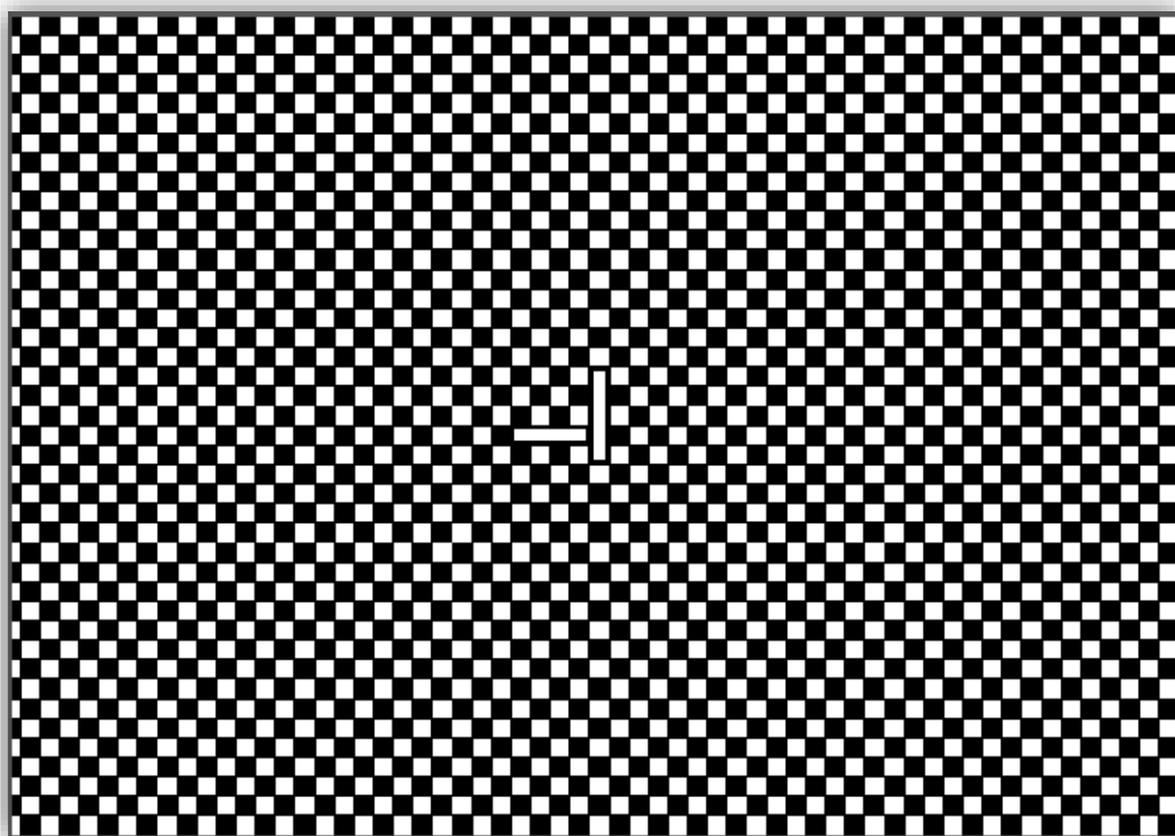


Ilustración 63: Patrón de calibración con origen

Para poder realizar una captura al patrón de calibración se ha de conectar a desde RobotStudio a la cámara del robot proporcionándole un nombre. Para este proyecto el nombre escogido es *camara1*. Dentro de la pantalla de Visión Integrada se ha de pulsar el botón de *Calibrar* seleccionando como tipo de calibración la opción de *Grid*, marcando la casilla de origen e indicando la distancia que hay entre los cuadros de calibración. Una vez está completada la calibración con los puntos suficientes se guarda el proyecto para poder trabajar con la pieza real.

El siguiente paso es realizar una captura con la pieza colocada en su lugar de trabajo. Como ya se ha realizado la calibración se indicará de la siguiente forma:

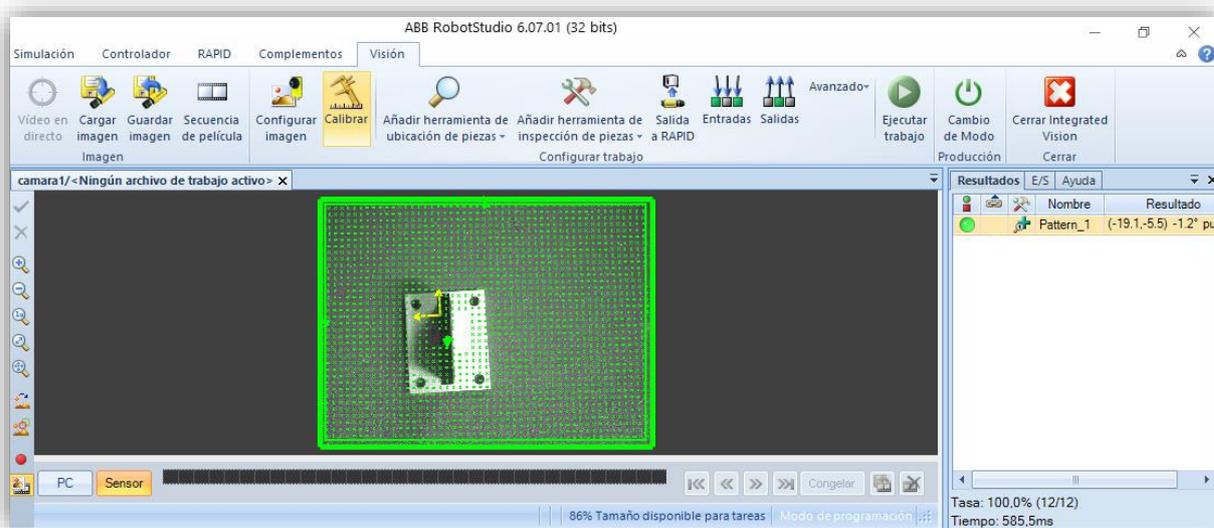


Ilustración 64: Captura de la pieza tras la calibración

Es el momento de proceder al reconocimiento de la pieza. Para ello se va a utilizar la herramienta *PatMax Pattern* que proporciona el software en la pestaña de localización de herramientas. Se recuadra la pieza a localizar utilizando el modelo de rectángulo y se ajusta el parámetro de la rotación de la imagen a 180°. Con esto la imagen queda entrenada de la siguiente manera:

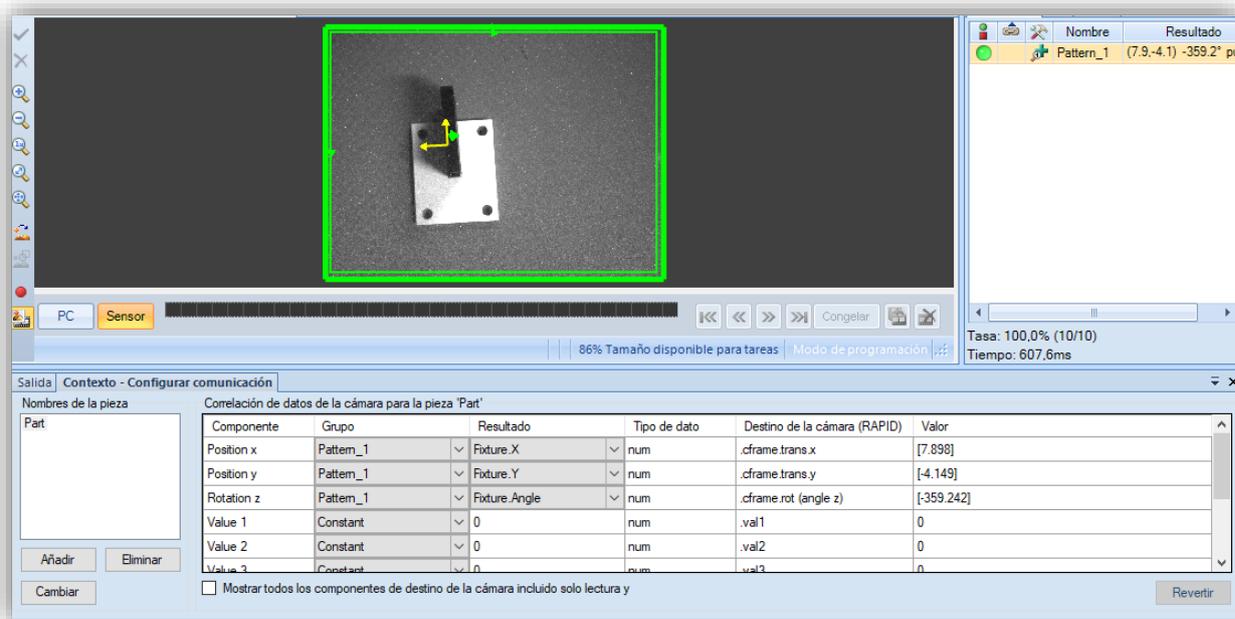


Ilustración 65: Reconocimiento de la imagen por el patrón PatMax

El último paso a realizar en la pestaña de Visión Integrada es el envío de los datos de la imagen que sean necesarios para el agarre de la pieza a través de la opción de *Salida a RAPID*. En esta ocasión, la información que se necesita es la posición del centroide de la pieza en los ejes X e Y, y la rotación del ángulo en el eje Z.

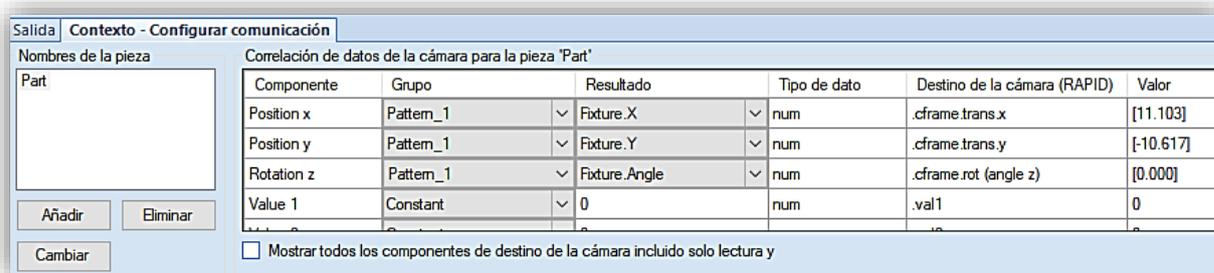


Ilustración 66: Datos enviados a RAPID

Tras todo este proceso del reconocimiento de la pieza, se ha de generar el código para poder realizar la captura de la foto durante el proceso de ensamblado. Para ello, es necesario dirigirse a la pestaña de RAPID donde esté situado el resto del código de programar e incluir la función de *MoveToDetectObject* en el apartado de Visión Integrada a través de la opción de *Fragmento*.

La primera parte del código generado es la declaración de las variables necesarias, tales como, el nombre del proyecto de visión, las posiciones del robot desde las cuales realiza la captura y toma la pieza y el objeto de trabajo creado para la cámara.

```
!Declaracion de variables para la vision
CONST string myjob := "calibracion5mm.job";
VAR cameratarget mycameratarget;

CONST robtarget p_LVision=[[606.83,176.98,355.25],[0.519616,-0.471904,0.515616,-0.491371],
[-1,0,1,5],[94.1762,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget p_LAgarre2=[[34.03,-17.55,20.75],[0.041352,0.0252738,-0.998726,0.014075],
[-1,-1,2,5],[-142.469,9E+09,9E+09,9E+09,9E+09,9E+09]];
TASK PERS wobjdata mywobj:=[FALSE,TRUE,"",[[479.846,197.671,0.842837],
[1,5.88045E-05,-2.2628E-05,7.44836E-05]],[[-33.5589,4.3408,0],[0.99465,0,0,0.103301]]];
```

Ilustración 67: Declaración de variables de visión

Se prosigue con el contenido de la propia función donde el robot se desplaza a la posición de captura de la imagen, realiza la fotografía y obtiene los datos que se le han especificado anteriormente.

El código al que hace referencia es el siguiente:

```
!Posicion de captura de imagen  
MoveJ p_LVision, v1000, fine, GripperL\WObj:=wobj0;  
CamReqImage camara1;  
CamGetResult camara1, mycameratarget;
```

Ilustración 68: Código de captura de imagen

Una vez obtenidos los datos, se realiza la homografía entre los puntos del centroide respecto a la cámara y a los puntos de la pieza en el mundo real, para lo cual se ha creado un nuevo objeto de trabajo, *myobj*, que se explicará a continuación.

```
!Cambio de coordenadas de la camara al objeto  
mywobj.oframe := mycameratarget.cframe;
```

Ilustración 69: Homografía de la cámara al mundo real

El último paso que realiza el robot es ir hacia la posición donde atrapa la pieza. Este punto de agarre no siempre es el mismo, ya que depende del objeto de trabajo según los datos obtenidos del programa de visión.

Para mayor comprensión, durante la primera ejecución del código donde la pieza está situada de la misma forma en la que se ha creado el programa de visión con su calibración correspondiente, se ha de llevar el robot al punto de agarre a través del modo de guiado. Este paso solo será necesario una única vez ya que, en próximas ejecuciones, el robot interpolará la posición de agarre a partir de los nuevos datos introducidos en RAPID desde el programa.

```
!Ir a posicion de agarre calculada  
MoveL p_LAgarre2, v500, fine, GripperL \WObj:=mywobj;  
g_GripIn;
```

Ilustración 70: Posición de alcance de la pieza por visión

Solo queda una última tarea por realizar, la definición del nuevo objeto de trabajo a través de la unidad de programación. Para ello, se sigue la guía del manual del producto en la cual se indica como crear un nuevo *wobjdata* en la pestaña de *Datos de programa*.

Se escoge la opción de crear el objeto de trabajo a través de 3 puntos de coordenadas. Es necesario utilizar el patrón de calibración de la cámara en la misma posición en la cual se ha realizado el resto del programa. Estos puntos serán tomados a ras del patrón de calibración donde la Z sea la mínima posible.

Los puntos necesarios para el nuevo objeto de trabajo son:

- Primer punto a lo largo del eje X.
- Primer punto a lo largo del eje Y manteniendo la X.
- Segundo punto a lo largo del eje X.

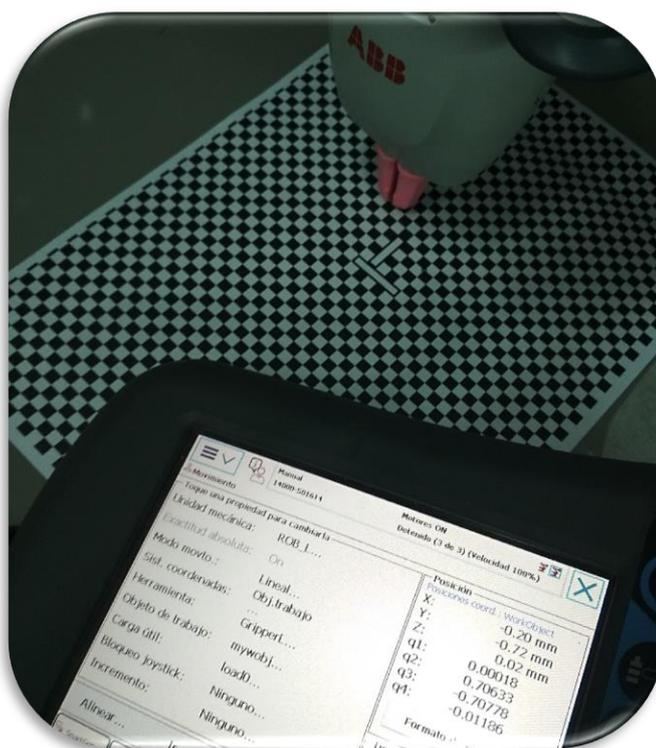


Ilustración 71: Creación de un nuevo *wobjdata*

Una vez creado el *wobj* el código ya podrá ser utilizado dando por finalizada la programación de la aplicación de ensamblado.

6. Validación de la aplicación

Es el momento de realizar la aplicación del ensamblaje de todas las piezas de forma práctica y continuada con la ayuda del robot para, posteriormente, poder analizar los resultados y extraer las conclusiones oportunas.

Para validar el proyecto, se han realizado varios ensayos con el robot completando todo el proceso y, por otra parte, se ha realizado el mismo ensayo sin la ayuda del robot. A través de estos dos ensayos se han extraído los tiempos de duración de ambas aplicaciones y se han valorado los resultados.



Ilustración 72: Pieza final ensamblada

Respecto al ensayo donde operan conjuntamente el robot con el humano se han separado los tiempos en los cuales el robot actúa por sí solo y los tiempos donde el humano actúa en colaboración con el robot. Los resultados son los siguientes:

Tabla 13: Tiempos de la aplicación humano-robot

APLICACIÓN HUMANO-ROBOT			
Robot		Humano	
Intervención	Tiempo(s)	Intervención	Tiempo(s)
1	5	Eje	11
2	13	Tornillo 1	12
3	4	Tornillo 2	12
4	2	Tornillo 3	14
		Tornillo 4	9
TOTAL	24	TOTAL	58

Con ello se obtiene una suma total de 82 segundos desde que comienza el programa hasta que la pieza final es depositada en la mesa de trabajo.

Se puede observar como la tarea del operario es mucho más costosa que la del robot, ya que tener que atornillar las piezas, lo que resulta una tarea más delicada y pueden ocurrir errores de no encajar las tuercas a la primera. Sin embargo, la tarea del robot es precisa y automática lo que aporta rapidez al conjunto. Aún con estos datos, sería posible reducir los tiempos del robot ya que se podrían eliminar las esperas del código y aportar mayor velocidad a ciertos movimientos, siempre dentro de la seguridad.

El segundo tipo de ensayo realizado es el ensamblado de la rueda del carro sin la ayuda del robot, solo con las dos manos del operario.

En este ensayo se encuentra una gran dificultad a la hora de unir la pieza rectangular superior a las dos piezas laterales, ya que estas últimas no se mantienen por sí solas y rotan alrededor de la rueda para permitir su movimiento. Es por ello que es necesario de apoyar toda la pieza sobre una superficie plana y centrar los agujeros de ambas piezas con una mano para poder encajar el tornillo con la otra, y en el siguiente movimiento enroscar la tuerca. Esta es la tarea que hace el robot en la aplicación de colaboración, por lo que el operario solo ha de encajar el tornillo sobre las dos piezas fijadas por las pinzas del robot.

A pesar de la libertad de las piezas para rotar se consigue acabar de montar la pieza, por lo que se han comparado los datos de tres pruebas realizadas por el operario obteniendo una media de tiempo de 106 segundos.

Tabla 14: Tiempos aplicación humano

APLICACIÓN HUMANO	
Intervención	Tiempo(s)
Prueba 1	115
Prueba 2	100
Prueba 3	102
MEDIA	106

Este resultado supera al tiempo de la aplicación de colaboración en 24 segundos, lo que supone que la aplicación sin la ayuda del robot sea un 22,64% más lenta.

Otra forma de realizar el montaje manual sin la ayuda del robot es haciendo uso de otras herramientas, como puede ser el caso de una torniqueta de apriete. Esta herramienta podría ser útil para fijar la pieza superior a las piezas laterales quedando libres las manos del operario para meter el tornillo y enroscar la tuerca.



Ilustración 73: Torniqueta de apriete

Con la comparación de ambos ensayos se demuestra cómo el uso del robot aporta flexibilidad a esta aplicación ya que resulta de gran ayuda cuando las piezas a ensamblar son móviles. También aporta otras ventajas como movimientos más rápidos y precisos que los movimientos de una persona y autonomía suficiente, como la detección de piezas, para seguir por sí solo el hilo del programa mientras el operario se encarga de otras funciones.

En vista general de la aplicación, sería acertado decir que se han superado todos los niveles de colaboración con el robot:

1. Colaboración secuencial: En el primer momento el robot coge dos piezas con los brazos mientras que el operario ensambla por su cuenta el eje con un lateral.
2. Cooperación: Mientras el robot sujeta la rueda y un lateral en una posición específica, el operario une el otro lateral a través del eje y la tuerca alada.
3. Colaboración receptiva: Se produce en 3 casos puntuales durante el programa en la lectura de los sensores de las pinzas y el módulo de aire, lo que provoca una respuesta del robot cuando el operario altera los parámetros.

7. Conclusiones

Del estudio, discusión y análisis de los datos experimentales se desprende un funcionamiento correcto del sistema. Puede concluirse que se han alcanzado de manera satisfactoria los objetivos iniciales. El sistema desarrollado ha permitido el ensamblado de una rueda apta para el funcionamiento de un carro de compra de tal manera que se ha podido estudiar el funcionamiento de un robot colaborativo de dos brazos.

- En primer lugar, al tratarse de un robot de siete articulaciones la cinemática del mismo es más compleja. Esto deriva en que gracias a esta articulación extra el robot es capaz de alcanzar más puntos que un robot convencional o un brazo humano, lo que aporta flexibilidad al sistema. A pesar de ello, se deben tener en cuenta los tipos de movimiento, *MoveL* o *MoveJ*, ya que durante esta aplicación se han asignado los puntos mediante el guiado del robot, teniendo que modificarlos ligeramente por problemas de cinemática por ser, en su mayoría, movimientos lineales.
- Otra decisión importante durante el desarrollo del proyecto fue el ajuste del entorno para la obtención de imágenes de calidad mediante el sistema de visión. Al principio, fue necesario ajustar el enfoque de la cámara manualmente para obtener una imagen nítida a la altura estimada para que el robot, posteriormente, pudiese tomar la pieza. Seguidamente, se escogió un patrón de calibración de 5 mm en lugar del patrón de 12 mm, proporcionado por *RobotStudio*, para ajustar mejor la resolución de la imagen al pequeño tamaño de las piezas. Finalmente, se optimizó el tiempo de ejecución del código en la tarea T_ROB_L activando la cámara en el momento de espera del operario y no durante el propio uso del módulo de visión, con lo que se concluye que es clave el acondicionamiento del entorno de trabajo.
- Como última conclusión se puede decir que el estudio y realización de la aplicación con los módulos de visión y aire del robot presentan la ventaja de ser volátiles dentro del entorno industrial. Se debe a que permiten desarrollar aplicaciones mucho más diversas y adaptables en cuanto al reconocimiento y tomado de piezas, superando adversidades, como formas complejas, delicadeza de los materiales, ajustando la fuerza de agarre de la pinza del robot u otros factores que puedan afectar al operario, como la temperatura o la suciedad.

Finalmente, es necesario comentar las posibles mejoras a realizar en tiempos futuros.

- Por una parte, queda disponible seguir mejorando el sistema de visión introduciendo distintos patrones para el reconocimiento de piezas o seguir acondicionando el entorno de trabajo, sobre todo respecto a la iluminación del mismo, evitando flashes y sombras.
- Por otra parte, como se ha comentado anteriormente, sería posible ajustar más los tiempos del robot reduciendo las esperas y aumentando la velocidad de ciertos movimientos, respetando la seguridad del operario.
- Por último, se podría aumentar la interacción entre el robot y la persona obteniendo una comunicación mayor mediante sensores, ya sean a través de lecturas de variables como la corriente o el par ejercido, o mediante otros sensores disponibles del robot o sensores externos que se le incorporen. De esta forma, se puede completar el quinto nivel de interacción referente a la colaboración receptiva.

8. Referencias

- ABB. (30 de 10 de 2018). *Manual del producto. Pinza del IRB 14000*. Obtenido de <https://abb.sluzba.cz/Pages/Public/IRC5UserDocumentationRW6/es/3HAC054949%20PM%20IRB%2014000%20Gripper-es.pdf>
- ABB AB, Robotics. (2015). *Manual del Producto IRB 14000*. Obtenido de <https://library.e.abb.com/public/c9f66b2b80bc42bba49c20970bed1b0d/3HAC052982%20PS%20IRB%2014000-en.pdf?x-sign=09i961Kw9WYypn8AUDHOiC8hUT/kTjLE+VudbTeATcD5UcH0biKarrKLuRwEL+rKw>
- ABB. (s.f.). *YuMi-Creando un futuro automatizado juntos*. Obtenido de <https://new.abb.com/products/robotics/es/robots-industriales/yumi>
- Beneito, J. A. (2018). *TFG- Control de fuerza con el brazo robot colaborativo UR3*. Valencia: UPV.
- COGITI. (2018). *COGITI Valencia*. Obtenido de <https://www.copitival.es/cogitiv/esp/inicio>
- Escalette, G. (s.f.). *Universal Robots UR5*. Obtenido de <https://www.universal-robots.com/es/productos/robot-ur5/>
- FANUC. (s.f.). *Robot colaborativo CR-35iA*. Obtenido de <https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/%D1%80obots-colaborativos/robot-colaborativo-cr35ia>
- FANUC. (s.f.). *Robot colaborativo CR-4iA*. Obtenido de <https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/%D1%80obots-colaborativos/collaborative-cr4ia>
- FANUC. (s.f.). *Robot colaborativo CR-7iA y CR-7iA/L*. Obtenido de <https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/%D1%80obots-colaborativos/collaborative-cr7ial>
- Gömemann, O. (20 de Agosto de 2017). *SICK. Sensor Intelligence*. Obtenido de SEGURIDAD DE LOS ROBOTS COLABORATIVOS MITOS – REALIDADES – FUTURO:

http://www.osalan.euskadi.eus/contenidos/informacion/ponencias_jt170720__cursoverano/es_def/adjuntos/ponencia_jt170720_otto_goernemann.pdf

ISO/TC 299 Robotics. (Noviembre de 2000). *International Organization for Standardization*. Obtenido de ISO 14539:2000:
<https://www.iso.org/standard/24062.html>

ISO/TC 299 Robotics. (Julio de 2011). *International Organization for Standardization*. Obtenido de ISO 10218-1:2011:
<https://www.iso.org/standard/51330.html>

ISO/TC 299 Robotics. (Julio de 2011). *International Organization for Standardization*. Obtenido de ISO 10218-2:2011:
<https://www.iso.org/standard/41571.html>

ISO/TC 299 Robotics. (Marzo de 2012). *International Organization for Standardization*. Obtenido de ISO 8373:2012 :
<https://www.iso.org/standard/55890.html>

ISO/TC 299 Robotics. (Febrero de 2016). *International Organization for Standardization*. Obtenido de ISO/TS 15066:2016:
<https://www.iso.org/standard/62996.html>

Jiva, E. I. (2019). *TFM- Desarrollo de la teleoperación de robots industriales y colaborativos mediante técnicas avanzadas de visión artificial*. Valencia: UPV.

KUKA. (s.f.). *LBR iiwa*. Obtenido de <https://www.kuka.com/es-es/productos-servicios/sistemas-de-robot/robot-industrial/lbr-iiwa>

Moreno, C. A. (2018). *TFM- Desarrollo de aplicaciones industriales con robots colaborativos utilizando el middleware de control de robots Robot Operating System*. Valencia: UPV.

Pelegrí, J. (9 de Mayo de 2018). *La industria 4.0: Camino hacia la automatización y la vanguardia*. Obtenido de Universal Robots: <https://blog.universal-robots.com/es/industria-40>

Pelegrí, J. (10 de Enero de 2019). *La cuarta revolución industrial: cobots y automatización*. Obtenido de Universal Robots: <https://blog.universal-robots.com/es/revolucion-industrial-cobots>

- Rebollar, F. S. (2019). *TFM- Implementación de una solución de robótica colaborativa para fabricación de interiores en el sector del automóvil. Desarrollo estratégico de su estándar global*. Valencia: UPV.
- Tsuyoshi SAITO1, T. H. (2015). Global harmonization of safety regulations for the use of industrial robots-robots-permission of collaborative operation and a related study by JNIOOSH. *J-STAGE*.
- Universal Robots. (17 de Mayo de 2016). *History of the cobots*. . Obtenido de How it all began.:
<https://www.skyfish.com/sh/9894fa74f7822448ae2282631eb5ebd06c49d947/39c50527/609203/viewer/19052438>
- Universal Robots. (2018). *Beyond the Cobot Buzz*. Obtenido de <https://info.universal-robots.com/cobots-vs-traditional-industrial-robots>
- Universal Robots. (s.f.). *Universal Robot UR3*. Obtenido de <https://www.universal-robots.com/es/productos/robot-ur3/>
- Universal Robots. (s.f.). *Universal Robots UR10*. Obtenido de <https://www.universal-robots.com/es/productos/robot-ur10/>
- Viladomat, J. (25 de Junio de 2019). *PAL Robotics*. Obtenido de TIAGo++, the robot you need for bi-manual tasks: <http://blog.pal-robotics.com/tiago-bi-manual-robot-research/>
- Wikipedia. (9 de Julio de 2019). *Cobot*. Obtenido de <https://es.wikipedia.org/wiki/Cobot>

Trabajo Final de Máster

ANEXOS

Sara Rabadán Mayordomo

1. PRESUPUESTO

En este capítulo se va a exponer el presupuesto realizado durante este proyecto, incluyendo maquinaria, materiales necesarios y personal requerido, ya que existe una necesidad de conocer el coste económico del proyecto en el caso de querer ponerlo en marcha en el entorno industrial en un futuro.

Primeramente, se expone el coste de la maquinaria utilizada en el cual solo se ha tenido en cuenta el coste del robot adquirido para el desarrollo de la aplicación.

Tabla 15: Presupuesto de la maquinaria

MAQUINARIA			
Concepto	Precio (€/u)	Cantidad (u)	Precio total (€)
Robot IRB 14000	40000	1	40.000,00 €
TOTAL			40.000,00 €

Se prosigue con el cálculo del coste de los materiales utilizados durante la aplicación de ensamblado teniendo en cuenta la cantidad utilizada de cada uno de ellos. Cabe destacar que el material de impresión utilizado para imprimir las piezas en 3D se ha calculado a través del peso de las piezas, en gramos.

Tabla 16: Presupuesto del material

MATERIAL			
Concepto	Precio (€/u)	Cantidad (u)	Precio total (€)
Material impresión PLA	14,8	0,3	4,44 €
Cable Ethernet	4,7	1	4,70 €
Recipiente plástico	0,83	2	1,66 €
Cinta antideslizante	2,9	0,2	0,58 €
Tornillo 5vmm	0,029	4	0,12 €
Tornillo 7vmm	0,0293	1	0,03 €
Tuerca hexagonal	0,2	4	0,80 €
Tuerca mariposa	0,5	1	0,50 €
TOTAL			12,83 €

Respecto a la mano de obra, se ha tenido en cuenta el precio por proyecto de un ingeniero obtenido a través de la información del colegio de ingenieros industriales de Valencia (COGITI, 2018).

Tabla 17: Presupuesto de mano de obra

MANO DE OBRA			
Concepto	Precio (€/h)	Cantidad (h)	Precio total (€)
Puesta en marcha	35	10	350,00 €
Diseño CAD	35	40	1.400,00 €
Programación RAPID	35	120	4.200,00 €
Desarrollo ensamblado	35	130	4.550,00 €
TOTAL			10.500,00 €

El presupuesto total se calcula a través de la suma de los módulos anteriores, teniendo en cuenta un concepto de gastos generales del 7% y aplicándole, posteriormente, los impuestos de IVA correspondientes del 21%.

Tabla 18: Presupuesto total

PRESUPUESTO TOTAL	
Maquinaria	40.000,00 €
Material	12,83 €
Mano de obra	10.500,00 €
Conceptos generales	3.535,90 €
IVA	11.350,23 €
PRECIO TOTAL	65.398,95 €

Tras lo descrito anteriormente, el coste total de la realización del proyecto asciende a la cantidad de 65.398,95€.

2. PLANOS

I. RUEDA

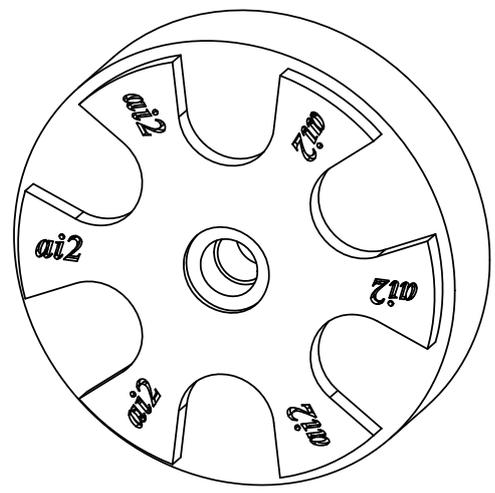
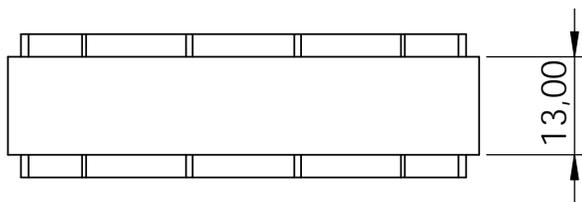
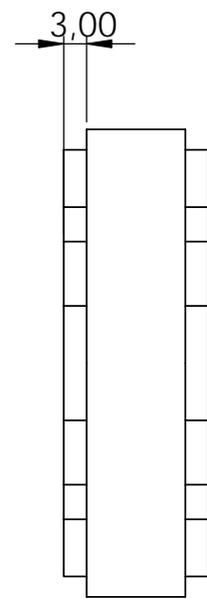
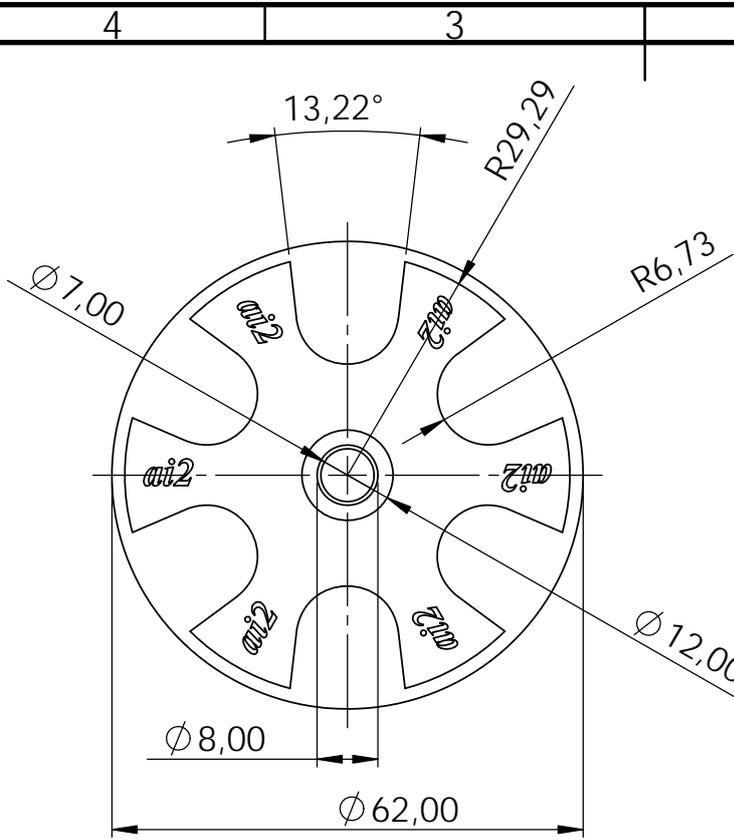
II. SOPORTE RUEDA

III. LATERAL DERECHO

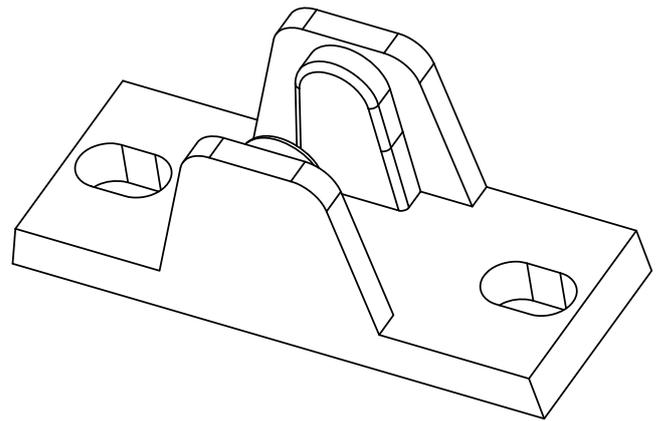
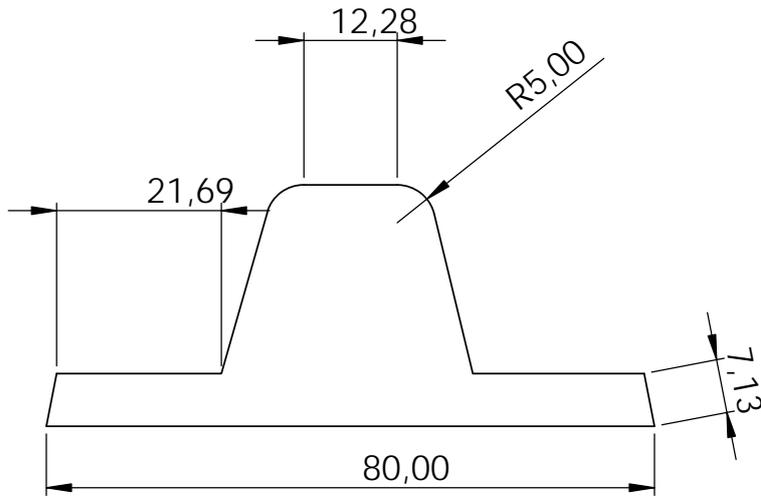
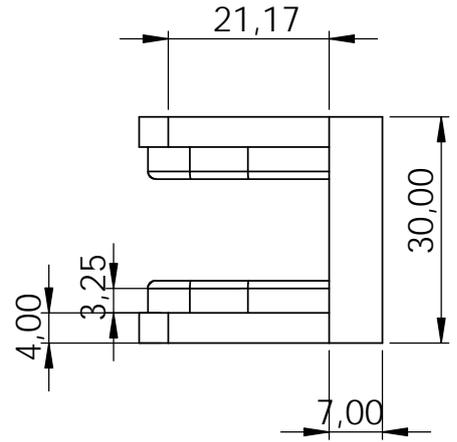
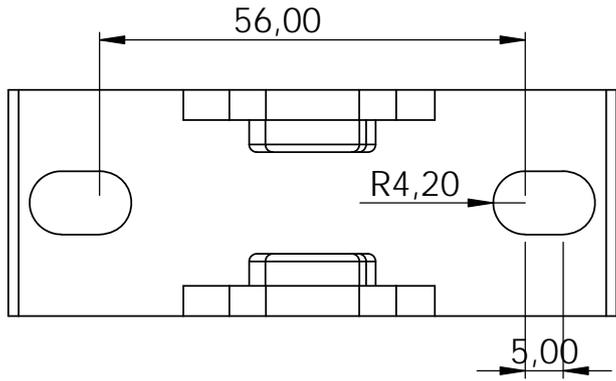
IV. LATERAL IZQUIERDO

V. SOPORTE LATERALES

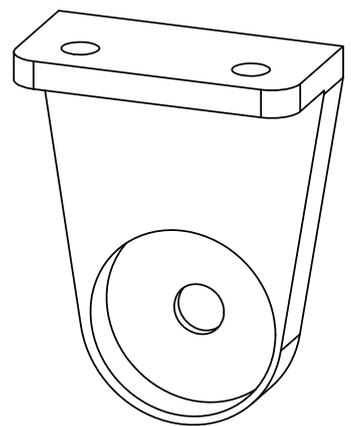
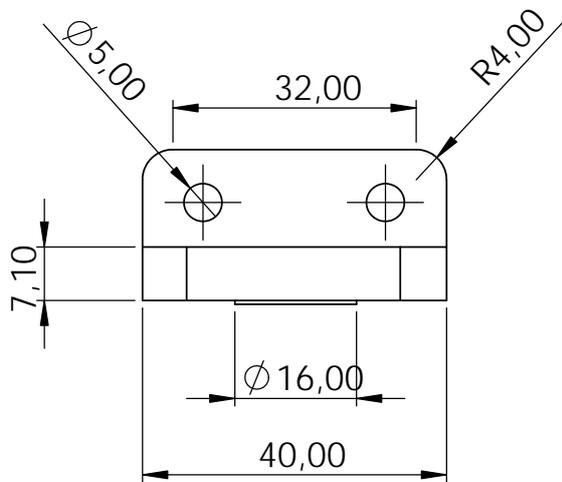
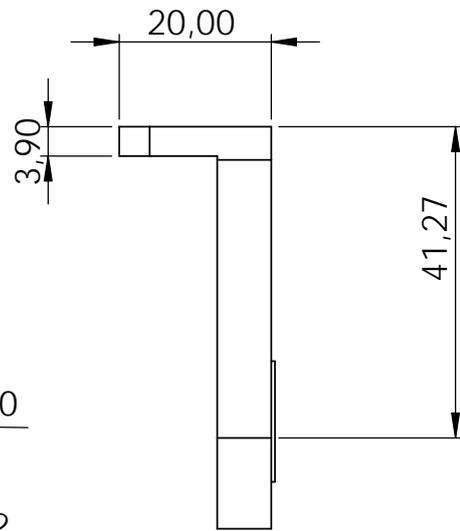
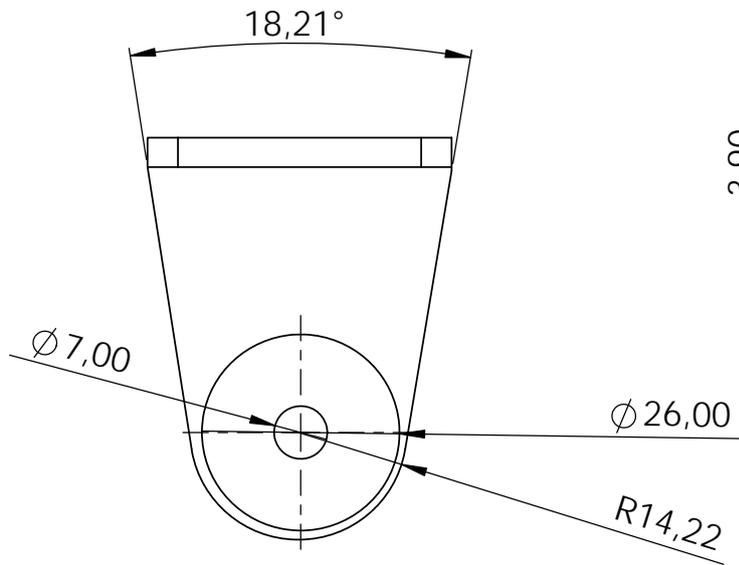
VI. SOPORTE SUPERIOR



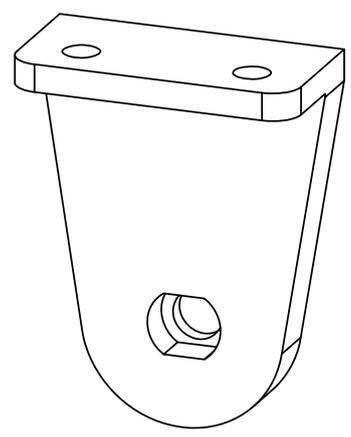
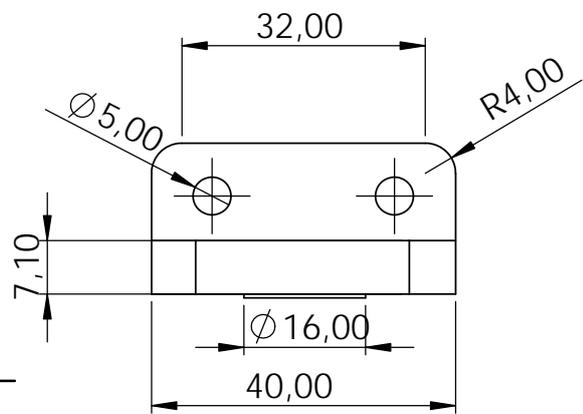
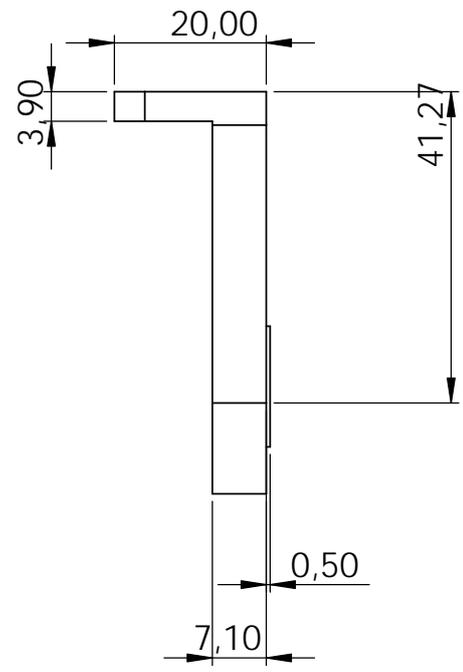
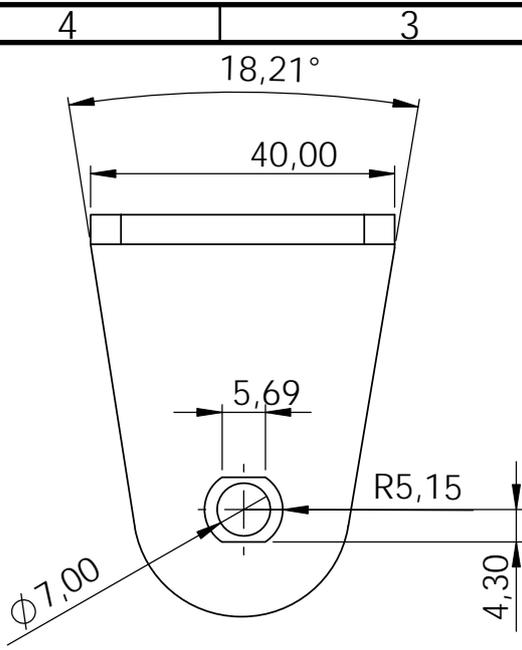
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO: RUEDA			
DIBUJ.						N.º DE DIBUJO Nº 1			
VERIF.									
APROB.						ESCALA:1:1			
FABR.									
CALID.				MATERIAL:		HOJA 1 DE 1			
				PESO:					



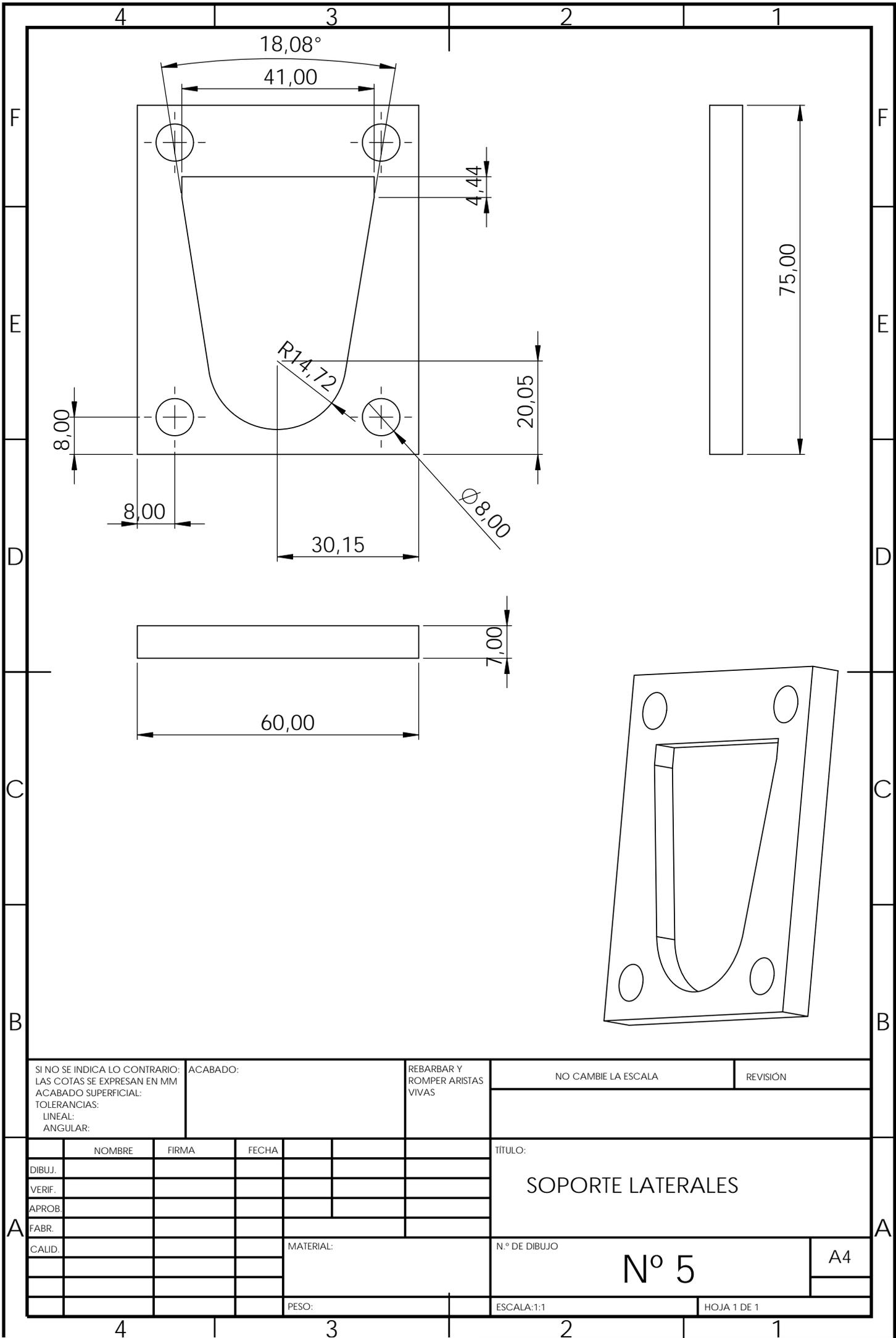
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
DIBUJ.		FIRMA		FECHA		TÍTULO: SOPORTE RUEDA			
VERIF.						N.º DE DIBUJO Nº 2			
APROB.						ESCALA:1:1			
FABR.						HOJA 1 DE 1			
CALID.				MATERIAL:		A4			
				PESO:					



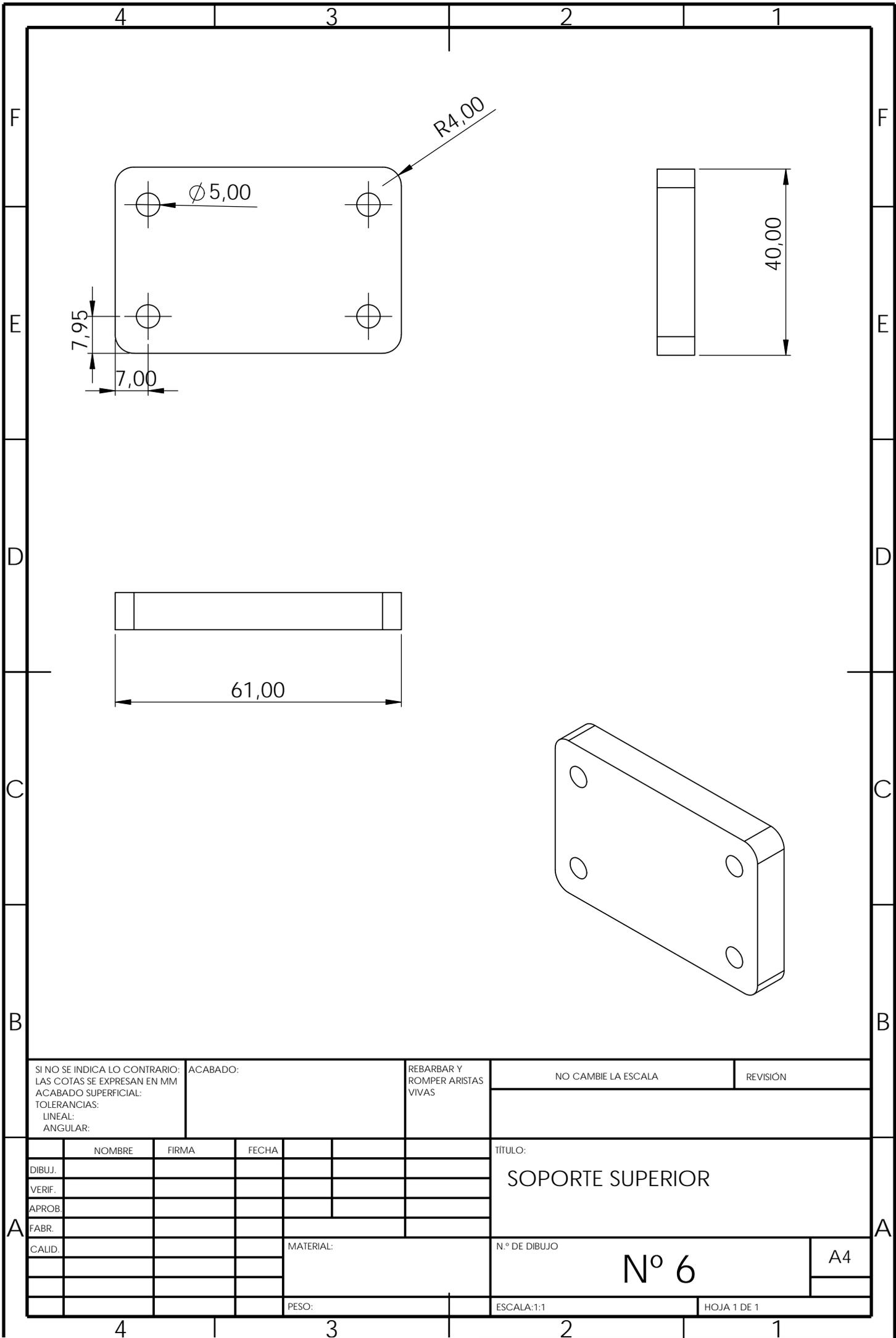
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO: LATERAL DERECHO			
DIBUJ.		VERIF.		APROB.		MATERIAL:			
FABR.		CALID.		PESO:		N.º DE DIBUJO Nº 3		A4	
ESCALA: 1:1		HOJA 1 DE 1							



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO: LATERAL IZQUIERDO			
DIBUJ.		VERIF.		APROB.		FABR.		CALID.	
MATERIAL:		N.º DE DIBUJO		N.º 4		A4			
PESO:		ESCALA: 1:1		HOJA 1 DE 1					



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO: SOPORTE LATERALES			
DIBUJ.		VERIF.		APROB.		N.º DE DIBUJO Nº 5			
FABR.		CALID.		MATERIAL:		ESCALA:1:1			
PESO:		ESCALA:1:1		HOJA 1 DE 1					



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO: SOPORTE SUPERIOR			
DIBUJ.						N.º DE DIBUJO Nº 6			
VERIF.									
APROB.						A4			
FABR.									
CALID.				MATERIAL:		ESCALA:1:1			
				PESO:		HOJA 1 DE 1			

3. CÓDIGO

I. T_ROB_L

II. T_ROB_R

```
!***** II - T_ROB_L *****!
```

```
MODULE MainModule
```

```
    PERS loaddata load1:=[0.11,[6.8,-0.6,-27.4],[1,0,0,0],0,0,0.01];
    PERS tooldata GripperL:=[TRUE,[[0,0,136],[1,0,0,0]],[0.23,[8.2,11.7,52],
[1,0,0,0],0.00021,0.00024,0.00009]];
    PERS tooldata YuMi_App_Default_Tool_L := [TRUE,[[0,0,84],[1,0,0,0]],[0.25,[7,0,40],
[1,0,0,0],0,0,0]];
    PERS string YuMi_App_Default_ToolName_L := "GripperL";

    PERS tasks YuMi_App_task_list{2} := [{"T_ROB_L"}, {"T_ROB_R"}];
    VAR syncident YuMi_App_sync1;
    VAR syncident YuMi_App_sync2;
    VAR syncident YuMi_App_sync3;
    VAR syncident YuMi_App_sync4;
    VAR syncident YuMi_App_sync5;
    VAR syncident YuMi_App_sync6;
    VAR syncident YuMi_App_sync7;
    VAR syncident YuMi_App_sync8;
    VAR syncident YuMi_App_sync9;
    VAR syncident YuMi_App_sync10;
    VAR syncident YuMi_App_sync11;

    CONST robtarjet p_LIni:=[495.60,21.70,370.94],[0.0297595,0.0516422,-0.701001,0.710665],
[-1,-2,2,5],[167.681,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarjet p_LAgarre1:=[454.44,337.84,65.31],[6.35775E-05,-0.7237,0.690115,-1.13387E-05],
[-1,-2,1,5],[-122.335,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarjet p_LEncaje1:=[553.18,15.06,318.01],[0.723659,-0.0566702,0.0384603,-0.686751],
[-1,0,1,4],[114.591,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarjet p_LEncaje2:=[592.08,-22.76,394.56],[0.499362,-0.514868,0.485119,-0.500208],
[-2,0,1,5],[92.3451,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarjet pCoord1_L10:=[540.74,42.17,320.92],[0.42637,0.507476,0.491592,-0.564813],
[-1,0,1,4],[162.818,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarjet p_LFinal:=[273.59,75.73,32.25],[1.29194E-05,-0.0249127,-0.99969,1.14023E-05],
[-1,0,1,5],[-163.042,9E+09,9E+09,9E+09,9E+09,9E+09]];

    !Variables de comunicacion con el robot
    VAR num posOpen;

    !Declaracion de variables para la vision
    CONST string myjob := "calibracion5mm.job";
    VAR cameratarjet mycameratarjet;

    CONST robtarjet p_LVision:=[606.83,176.98,355.25],[0.519616,-0.471904,0.515616,-0.491371],
[-1,0,1,5],[94.1762,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarjet p_LAgarre2:=[34.03,-17.55,20.75],[0.041352,0.0252738,-0.998726,0.014075],
[-1,-1,2,5],[-142.469,9E+09,9E+09,9E+09,9E+09,9E+09]];
    TASK PERS wobjdata mywobj:=[FALSE,TRUE,"",[479.846,197.671,0.842837],
[1,5.88045E-05,-2.2628E-05,7.44836E-05]],[[-33.5589,4.3408,0],[0.99465,0,0,0.103301]]];

PROC MoveToDetectedObject()

    !Posicion de captura de imagen
    MoveJ p_LVision, v1000, fine, GripperL\WObj:=wobj0;
    CamReqImage camara1;
    CamGetResult camara1, mycameratarjet;
```

```
!Cambio de coordenadas de la camara al objeto
mywobj.oframe := mycameratarget.cframe;

!Ir a posicion de agarre calculada
MoveL p_LAgarre2, v500, fine, GripperL \Wobj:=mywobj;
g_GripIn;
```

ENDPROC

PROC main()

```
!Posicion inicial
g_GripOut;
MoveJ p_LIni, v1000, fine, GripperL\Wobj:=wobj0;
WaitSyncTask YuMi_App_sync1, YuMi_App_task_list;

!Posicion de seguridad y agarre de la rueda
MoveJ Offs(p_LAgarre1,0,0,180), v1000, fine, GripperL\Wobj:=wobj0;
MoveL p_LAgarre1, v500, fine, GripperL\Wobj:=wobj0;
g_GripIn;
GripLoad load1;
MoveL Offs(p_LAgarre1,0,0,180), v500, fine, GripperL\Wobj:=wobj0;

!Posicion de encaje de la rueda y los triangulos
MoveJ p_LEncaje1, v1000, fine, GripperL\Wobj:=wobj0;
WaitSyncTask YuMi_App_sync2, YuMi_App_task_list;

!Configuracion de la camara
CamSetProgramMode camara1;
CamLoadJob camara1, myjob;
CamSetRunMode camara1;

!Movimiento sincronizado para posicion de seguridad de agarre
SyncMoveOn YuMi_App_sync3, YuMi_App_task_list;
MoveJ pCoord1_L10\ID:=10, v1000, fine, GripperL\Wobj:=wobj0;
SyncMoveOff YuMi_App_sync4;

!Cambio de pinza de agarre
WaitSyncTask YuMi_App_sync5, YuMi_App_task_list;
g_GripOut;
MoveL Offs(pCoord1_L10,0,100,0), v500, fine, GripperL\Wobj:=wobj0;
WaitSyncTask YuMi_App_sync6, YuMi_App_task_list;

!Detección de la pieza superior por vision
MoveToDetectedObject;

!Posición de encaje final
MoveJ Offs(p_LEncaje2,0,220,0), v1000, fine, GripperL\Wobj:=wobj0;
WaitSyncTask YuMi_App_sync7, YuMi_App_task_list;
MoveL p_LEncaje2, v100, fine, GripperL\Wobj:=wobj0;

!Retirada
WaitSyncTask YuMi_App_sync8, YuMi_App_task_list;
g_JogIn;
WaitTime 1;
MoveJ Offs(p_LEncaje2,-50,0,0), v50, fine, GripperL\Wobj:=wobj0;
```

```
WaitSyncTask YuMi_App_sync9, YuMi_App_task_list;

!Segunda espera del operador mediante las pinzas
posOpen:= g_GetPos();

WHILE posOpen < 18 DO
    posOpen:= g_GetPos();
ENDWHILE

WaitTime 1;
WaitSyncTask YuMi_App_sync10, YuMi_App_task_list;

!Fin de programa
MoveJ Offs(p_LIni,100,100,0), v1000, fine, GripperL\WObj:=wobj0;
WaitSyncTask YuMi_App_sync11, YuMi_App_task_list;
```

```
ENDPROC
```

```
ENDMODULE
```

```
!***** II - T_ROB_R *****!
```

```
MODULE MainModule
```

```
    PERS tooldata GripperR:=[TRUE,[[0,0,136],[1,0,0,0]],[0.23,[8.2,11.7,52],
[1,0,0,0],0.00021,0.00024,0.00009]];
    PERS tooldata YuMi_App_Default_Tool_R := [TRUE,[[0,0,84],[1,0,0,0]],[0.215,[8.7,12.3,49.2],
[1,0,0,0],0.00021,0.00024,0.00009]];
    PERS string YuMi_App_Default_ToolName_R := "GripperR";
    TASK PERS wobjdata wobjCoord:=[FALSE,FALSE,"ROB_L",[[0,0,0],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];

    PERS tasks YuMi_App_task_list{2} := [{"T_ROB_L"}, {"T_ROB_R"}];
    VAR syncident YuMi_App_sync1;
    VAR syncident YuMi_App_sync2;
    VAR syncident YuMi_App_sync3;
    VAR syncident YuMi_App_sync4;
    VAR syncident YuMi_App_sync5;
    VAR syncident YuMi_App_sync6;
    VAR syncident YuMi_App_sync7;
    VAR syncident YuMi_App_sync8;
    VAR syncident YuMi_App_sync9;
    VAR syncident YuMi_App_sync10;
    VAR syncident YuMi_App_sync11;

    !Declaración de posiciones a alcanzar
    CONST robtarget p_RIni:=[[492.96,5.34,369.18],[0.721449,-0.688998,0.0293367,0.0627081],
[1,0,-2,4],[-173.256,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget p_RAgarre1:=[[371.94,-94.85,77.13],[0.514796,-0.514798,0.484753,0.48475],
[1,0,-2,4],[160.847,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget p_REncaje1:=[[529.04,-74.19,239.14],[0.055187,0.675738,0.734932,0.0143701],
[1,1,0,5],[158.84,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget p_RAgarre2:=[[537.76,2.20,322.10],[0.526641,-0.412841,0.582614,0.461272],
[1,2,1,5],[-168.416,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget pCoord1_R10:=[[1.92,6.11,196.48],[0.0416059,-0.99911,-0.00653611,-0.00227873],
[1,2,1,5],[-168.414,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget p_REncaje2:=[[586.73,-83.84,388.69],[0.510992,-0.511291,-0.488649,-0.488559],
[1,2,-1,5],[-149.624,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget p_REncaje3:=[[591.82,-83.06,329.92],[0.471639,-0.471638,0.526831,0.526842],
[1,2,1,5],[-158.279,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget p_RFInal:=[[301.90,-402.96,98.25],[0.0206709,0.668197,0.743637,0.00950073],
[0,1,-1,5],[-172.321,9E+09,9E+09,9E+09,9E+09,9E+09]];

    !Variables de comunicacion con el robot
    VAR num posClose;
    VAR num presion1;

    PROC main()

    !Estado inicial
    g_JogIn;
    g_VacuumOff1;
    g_BlowOff1;
    g_BlowOff2;

    !Posicion inicial
```

```

MoveJ p_RIni,v1000,fine,GripperR;
WaitSyncTask YuMi_App_sync1, YuMi_App_task_list;

!Posicion de seguridad y succion del triangulo
MoveJ Offs(p_RAgarre1, 0,0,35),v1000,fine,GripperR;
MoveL p_RAgarre1,v500,fine,GripperR;
g_VacuumOn1;
WaitTime 1;
MoveL Offs(p_RAgarre1, 0,0,35), v500, fine, GripperR\WObj:=wobj0;

!Posicion de seguridad y encaje de la rueda y los triangulos
MoveJ Offs(p_REncaje1, 0,-60,0), v1000, fine, GripperR\WObj:=wobj0;
WaitSyncTask YuMi_App_sync2, YuMi_App_task_list;
MoveL p_REncaje1, v500, fine, GripperR\WObj:=wobj0;

!Espera del operador mediante las pinzas
g_JogOut;
posClose:= g_GetPos();

WHILE posClose > 24 DO
    posClose:= g_GetPos();
ENDWHILE

!Retirada
WaitTime 1;
g_VacuumOff1;
MoveL Offs(p_REncaje1, 0,-100,0), v500, fine, GripperR\WObj:=wobj0;

!Movimiento sincronizado para posicion de seguridad de agarre
SyncMoveOn YuMi_App_sync3, YuMi_App_task_list;
MoveJ pCoord1_R10\ID:=10, v1000, fine, GripperR\WObj:=wobjCoord;
SyncMoveOff YuMi_App_sync4;

!Cambio de pinza de agarre
MoveL p_RAgarre2, v500, fine, GripperR\WObj:=wobj0;
g_GripIn;
WaitSyncTask YuMi_App_sync5, YuMi_App_task_list;
WaitSyncTask YuMi_App_sync6, YuMi_App_task_list;

!Posicion de seguridad y encaje final
MoveJ Offs(p_REncaje2, 0,-60,100), v1000, fine, GripperR\WObj:=wobj0;
WaitSyncTask YuMi_App_sync7, YuMi_App_task_list;
MoveJ p_REncaje2, v500, fine, GripperR\WObj:=wobj0;

!Primera espera del operador mediante soplado
g_BlowOn1;
presion1 := g_GetPressure1();

WHILE presion1 < 108 DO
    presion1:= g_GetPressure1();
ENDWHILE

WaitTime 1;
g_BlowOff1;
WaitSyncTask YuMi_App_sync8, YuMi_App_task_list;

!Giro de la posicion de encaje
WaitSyncTask YuMi_App_sync9, YuMi_App_task_list;

```

```
MoveL p_REcaje3, v1000, fine, GripperR\WObj:=wobj0;  
WaitSyncTask YuMi_App_sync10, YuMi_App_task_list;
```

```
!Deposicion de la pieza ensamblada  
MoveJ p_RFinal, v1000, fine, GripperR\WObj:=wobj0;  
WaitSyncTask YuMi_App_sync11, YuMi_App_task_list;
```

```
ENDPROC
```

```
ENDMODULE
```