



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



Tecnológico Nacional de México

Instituto Tecnológico de Celaya

Departamento de Ingeniería Mecánica

Tesis de Maestría

**“GENERACIÓN Y ANÁLISIS DE MAQUINADOS CNC MEDIANTE UN
ROBOT SERIAL”**

Presenta:

JOSE DANIELMATA MEDINA

Asesor interno:

DR. HORACIO OROZCO MENDOZA

Septiembre 2024

CELAYA, GUANAJUATO

DEDICATORIA

Dedico este proyecto a mi familia quienes siempre impulsaron mis metas. A mis hermanos, que siempre están conmigo en los momentos difíciles y en mis logros. A mis profesores y asesores, cuya orientación ha encaminado mi camino académico y profesional.

AGRADECIMIENTOS

Quiero expresar mi profundo agradecimiento a todas las personas que contribuyeron de manera significativa a la realización de este proyecto.

En primer lugar, agradezco a mi director de tesis por su orientación a lo largo de este proceso. Sus conocimientos y consejos fueron fundamentales para dar forma al trabajo de investigación.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías por brindarme el apoyo durante toda mi estancia de posgrado a la par del Instituto tecnológico de Celaya que me dieron la oportunidad de seguir formándome como estudiante y lo más importante en mi crecimiento como persona.

RESUMEN.

Este proyecto se centra en el desarrollo de una estación de maquinado CNC que combina la versatilidad de un robot industrial con una herramienta de corte en el efector final para producir maquinados tipo CNC. En un contexto de creciente demanda de eficiencia y flexibilidad en la industria manufacturera, la estación de maquinado robótico busca ofrecer una solución integral para optimizar los procesos de fabricación flexible.

La justificación de esta investigación radica en la necesidad de superar los desafíos actuales relacionados con la adaptabilidad y precisión en entornos de maquinado CNC. La elección estratégica de integrar un robot industrial acoplado a un husillo apunta a maximizar la flexibilidad del sistema y mejorar la eficiencia del proceso de maquinado.

Los objetivos del proyecto incluyen el desarrollo de un sistema de control autónomo para el husillo, mediante un variador de frecuencia, la implementación de programación para el robot industrial y la realización de pruebas para evaluar la precisión y eficiencia del sistema.

Se espera que este proyecto contribuya significativamente al avance de la automatización industrial, proporcionando una solución innovadora y adaptable a las demandas cambiantes de la fabricación moderna.

ABSTRACT.

This project focuses on the development of a CNC machining station that combines the versatility of an industrial robot with a cutting tool on the end effector to produce CNC-type machining. In the context of increasing demand for efficiency and flexibility in the manufacturing industry, the robotic machining station seeks to offer a comprehensive solution to optimize flexible manufacturing processes.

The justification for this research lies in the need to overcome current challenges related to adaptability and precision in CNC machining environments. The strategic choice to integrate an industrial robot by coupling a spindle aims to maximize the system's flexibility and improve the efficiency of the machining process.

The project's objectives include the development of an autonomous control system for the spindle, using a frequency inverter, the implementation of programming for the industrial robot, and conducting tests to evaluate the system's precision and efficiency.

This project is expected to significantly contribute to the advancement of industrial automation, providing an innovative and adaptable solution to the changing demands of modern manufacturing.

INTRODUCCIÓN

El contenido del proyecto está dirigido a la rama de la manufactura específicamente a las áreas de control numérico y robótica. El Departamento de Ingeniería Mecánica cuenta con diversos equipos de maquinado convencionales como lo son tornos paralelos, taladros de banco, fresadoras y rectificadoras. Además, tiene disponibles tres centros CNC (Control Numérico Computarizo), dos fresadoras y un torno. Cabe mencionar que el control numérico se aplica también en otro tipo de máquinas herramientas como mandrinadoras, punzadoras, dobladoras, plegadoras, prensas, máquinas de electroerosión, etc.

La importancia del control numérico en procesos de maquinado radica en su exactitud y precisión, reduciendo los tiempos ciclo e incrementando notoriamente la producción, aunque es cierto que existen algunos inconvenientes. Las fresadoras CNC son las más apropiadas para la fabricación de herramientales y piezas de maquinaria. El principal inconveniente de utilizar fresadoras CNC es la zona de trabajo delimitada por el tamaño de la estación, la cual predispone a mecanizar una pieza en múltiples operaciones, o en su defecto dividir la pieza en secciones y volver a ensamblarla después de cada operación de maquinado.

La mayoría de las fresadoras CNC son capaces de controlar únicamente desplazamientos lineales coordinados de 3 ejes. Para fabricar formas complejas es necesario contar con 5 ejes continuos, sin embargo, los centros de maquinado de 5 ejes siguen siendo poco accesibles y de un costo elevado. Este proyecto estudia la solución obtenida al conformar una estación de maquinado utilizando un robot industrial ABB, que cuenta con 6 ejes lo que permitirá tener un espacio de trabajo mayor y la facilidad de maquinar piezas que requieran 6 grados de libertad.

En el capítulo uno se describe el marco de referencia el cual se conforma por los antecedentes, planteamiento del problema, objetivos, justificación, hipótesis, alcances y limitaciones. En el capítulo dos, el marco teórico, se recopila información relevante acerca del control numérico, robots ABB y el variador de frecuencia. La puesta en marcha del robot, la integración de la estación y el control de velocidad del husillo se describe en el capítulo tres y capítulo cuatro. En el capítulo cinco se muestran los resultados obtenidos del proyecto, así como las soluciones a los problemas que surgieron durante el proyecto. En el capítulo seis se discuten las conclusiones del proyecto.

ÍNDICE

DEDICATORIA	I
AGRADECIMIENTOS	II
RESUMEN.....	III
ABSTRACT.....	IV
INTRODUCCIÓN	V
1. CAPÍTULO I MARCO DE REFERENCIA	1
1.1 ANTECEDENTES Y ESTADO DEL ARTE	1
1.2 PLANTEAMIENTO DEL PROBLEMA	5
1.3 HIPÓTESIS	6
1.4 OBJETIVO GENERAL	6
1.5 OBJETIVOS ESPECÍFICOS	6
1.6 ALCANCES	7
1.7 LIMITACIONES	7
2 CAPÍTULO II MARCO TEÓRICO.....	8
2.1 SISTEMAS DE CONTROL NUMÉRICO	8
2.2 ROBÓTICA INDUSTRIAL.....	12
2.3 RECURSOS PARA EL DESARROLLO DEL PROYECTO	14
2.4 CABEZAL DE FRESADO.....	21
2.5 INGENIERÍA INVERSA.....	24
3 CAPÍTULO III METODOLOGÍA Y DESARROLLO DEL PROYECTO	27
3.1 REHABILITACIÓN DEL MANIPULADOR	27
3.2 INTEGRACIÓN DE LA ESTACIÓN	33
4. CAPÍTULO IV GENERACIÓN DE LOS PROGRAMAS DE MAQUINADO	41
4.1 PROGRAMACIÓN EN <i>FUSION 360</i>	41
4.2 PROGRAMACIÓN EN <i>ROBOTDK</i>	46
4.3 CONFIGURACIÓN DEL VARIADOR DE FRECUENCIA	48
4.4 INGENIERÍA INVERSA.....	53
5. CAPÍTULO V ANÁLISIS DE RESULTADOS	57

5.1 PRUEBAS DE MAQUINADO EN MATERIAL DELRIN Y ALUMINIO	57
5.2 MAQUINADO DE SUPERFICIES PLANAS	58
5.3 MAQUINADO DE TRAYECTORIAS CURVAS CON ESTRATEGIAS MULTI-EJES.....	59
5.4 MAQUINADO DE MODELO GENERADO MEDIANTE INGENIERÍA INVERSA	62
5.5 ÁREAS DE OPORTUNIDAD.....	64
CONCLUSIONES	66
REFERENCIAS.....	69
ANEXOS.....	72
ANEXO 1 PARÁMETROS DE CORTE EN OPERACIÓN DE FRESADO	72
ANEXO 2 ARRANQUE EN FRIO DEL BRAZO ROBÓTICO.	73
ANEXO 3 CALIBRACIÓN DE LOS EJES	78
ANEXO 4 PROGRAMA DE POSICIONAMIENTO DE CALIBRACIÓN.....	80
ANEXO 5 DEFINICIÓN DE PLANOS DE TRABAJO.	81
ANEXO 6 DEFINICIÓN DEL <i>TCP</i>	83
ANEXO 7 HABILITACIÓN DE LOS MÓDULOS I/O.....	85
ANEXO 8 ENVÍO DE DATOS POR PUERTO SERIAL.	88
ANEXO 9 PLANO DE FABRICACIÓN DE LA PLACA DE SUJECCIÓN DEL HUSILLO	91
ANEXO 10 VERIFICACIÓN DE COORDENADAS DE PLANOS Y OBJETOS.	92
ANEXO 11 FUNCIONES <i>MODBUS</i>	95
ANEXO 12 PROGRAMA DE COMUNICACIÓN PARA <i>ARDUINO MEGA</i>	96
ANEXO 13 PROGRAMA DE MAQUINADO	101

ÍNDICE DE FIGURAS

Figura 1.1	Centro de maquinado de husillo horizontal. [1]-----	1
Figura 1.2	Primera fresadora CNC de Richard Kegg. -----	2
Figura 1.3	Máquina herramienta hexápoda y vista detallada de la herramienta de corte. [4] -----	3
Figura 1.4	Operación de soldadura mediante <i>robot ABB</i> . [6] -----	4
Figura 1.5	Operación de fresado mediante robot <i>KUKA</i> . [7]-----	4
Figura 2.1	Máquina multi eje CNC. [11] -----	8
Figura 2.2	Trayectoria lineal G01 [11]. -----	10
Figura 2.3	Trayectoria circular G02 [11]. -----	10
Figura 2.4	Operaciones de fresado a) Perfil, b) Caja, c) Contorno. [1] -----	11
Figura 2.5	Modelos de robot ABB. -----	15
Figura 2.6	Componentes, a) Manipulador, b) Controlador <i>IRB 6400</i> . -----	15
Figura 2.7	Dimensiones del brazo robótico IRB 6400. -----	16
Figura 2.8	Panel de control. -----	17
Figura 2.9	<i>Teach Pendant</i> . -----	17
Figura 2.10	Programación <i>Online/Automatic</i> . [16] -----	18
Figura 2.11	Programación <i>Offline/ Manual</i> . [16] -----	18
Figura 2.12	Página web de <i>RobotDK</i> . -----	20
Figura 2.13	Página web del programa <i>Fusion 360</i> de <i>Autodesk</i> . -----	21
Figura 2.14	Fresadora de husillo vertical. [4] -----	21
Figura 2.15	Motor <i>Hua jiang</i> y variador de frecuencia. -----	22
Figura 2.16	<i>VDF H100 series</i> . [17] -----	22
Figura 2.17	Módulos de control de velocidad, a) Manual, b) <i>RS-485</i> , c) Entradas digitales. -----	23
Figura 2.18	Circuito del variador H100. [17]-----	23
Figura 2.19	Proceso de ingeniería inversa. [21]-----	25
Figura 2.20	Escáner <i>EiScan -Pro+</i> . -----	26
Figura 3.1	Metodología del proyecto. -----	27
Figura 3.2	Metodología para puesta en marcha. [22] -----	28
Figura 3.3	Manuales IRB 6400, a) Producto, b) Usuario, c) Rapid. -----	28
Figura 3.4	Unidad de sustitución USB. -----	29
Figura 3.5	Software emulador <i>Floppy Manager</i> . -----	29
Figura 3.6	Sistemas de coordenadas del manipulador. -----	30
Figura 3.7	Método de definición del <i>TCP</i> , a) 4 puntos, b) 6 puntos. -----	31
Figura 3.8	Definición de <i>TCP</i> . -----	31

Figura 3.9	Módulos de comunicación en el controlador.-----	32
Figura 3.10	Parámetros del puerto serial. -----	32
Figura 3.11	Componentes de mecanizado robótico.-----	33
Figura 3.12	Diagrama para la integración de la estación. -----	34
Figura 3.13	Módulo de mecanizado en: a) <i>RobotDk</i> , b) <i>Fusion 360</i> .-----	35
Figura 3.14	Montaje de estructura física. -----	35
Figura 3.15	Dimensiones de la brida de montaje del robot en mm. [15]-----	36
Figura 3.16	Placa de sujeción del husillo.-----	36
Figura 3.17	Ubicación de los sistemas de coordenadas. -----	37
Figura 3.18	Detalles de coordenadas.-----	37
Figura 3.19	Modelo <i>IRB 6400</i> en <i>RobotDK</i> .-----	38
Figura 3.20	Modelos de la estación. -----	38
Figura 3.21	Posicionamiento de objetos en <i>RobotDK</i> . -----	39
Figura 3.22	Variables del sistema de trabajo <i>TCP</i> y <i>Wobj</i> .-----	39
Figura 4.1	Diagrama de operaciones en la programación. -----	41
Figura 4.2	Secuencia de programación en <i>Fusion 360</i> . -----	42
Figura 4.3	Apartado en <i>Fusion360</i> para enviar el modelo 3D a <i>RobotDk</i> . -----	42
Figura 4.4	Configuración del módulo de fabricación. -----	43
Figura 4.5	Estrategias de mecanizado. -----	44
Figura 4.6	Parámetros de corte en <i>Fusion 360</i> . -----	45
Figura 4.7	Simulación en <i>Fusion 360</i> . -----	45
Figura 4.8	Envío de código NC. -----	46
Figura 4.9	Secuencia de programación en <i>RobotDk</i> . -----	46
Figura 4.10	Módulo mecanizado en <i>RobotDk</i> . -----	47
Figura 4.11	Árbol de operaciones <i>RobotDk</i> . -----	48
Figura 4.12	Cableado del husillo. -----	48
Figura 4.13	Cableado de botonera industrial. -----	49
Figura 4.14	Software <i>Commix</i> . -----	50
Figura 4.15	Proceso de envío de datos <i>USB-RS485</i> . -----	51
Figura 4.16	Trama de datos <i>Modbus</i> . -----	51
Figura 4.17	Proceso de envío de datos por medio de <i>Arduino Mega</i> . -----	52
Figura 4.18	Metodología para ingeniería inversa. -----	54
Figura 4.19	Generación de nube de puntos. -----	54
Figura 4.20	Alineación de la nube de puntos. -----	55

Figura 4.21	Generación de malla en archivo <i>stl</i> . -----	55
Figura 4.22	Visualización de mallado del modelo en Fusion 360. -----	56
Figura 4.23	Reducción del número de superficies de mallado. -----	56
Figura 4.24	Transformación de malla a sólido. -----	56
Figura 5.1	Mecanizado de pieza usando material Delrin. -----	57
Figura 5.2	Mecanizado de pieza usando material de aluminio. -----	57
Figura 5.3	Herramienta de corte de diámetro 30 mm. -----	58
Figura 5.4	Maquinado en superficie plana. -----	59
Figura 5.5	Ángulo de suavizado en maquinado de trayectoria curvas. -----	60
Figura 5.6	Posición de la prensa, A) Superior izquierda, B) Inferior izquierda. -----	60
Figura 5.7	Maquinado de trayectoria curva y dirección de medición de rugosidad. -----	61
Figura 5.8	Modelo 3D de automóvil generado por escáner. -----	62
Figura 5.9	Inclinación de herramienta en estrategias Multi-ejes. -----	63
Figura 5.10	Maquinado de automóvil a escala. -----	63
Figura 5.11	Amortiguamiento de vibraciones en función de la cantidad de componentes. -----	64
Figura 5.12	Errores de singularidad en trayectorias circulares. -----	65
Figura 5.13	Problemática en mecanizado con multi-ejes. -----	65

ÍNDICE DE TABLAS

Tabla 2.1	Direcciones principales del NC. -----	9
Tabla 2.2	Clasificación de brazos robóticos conforme a su generación. [12]-----	13
Tabla 2.3	Clasificación general de los robots. [14] -----	13
Tabla 2.4	Sectores de aplicación de las estaciones de mecanizado robótico. [9]-----	14
Tabla 2.5	Rango angular de movimientos del <i>IRB 6400</i> . -----	16
Tabla 2.6	Softwares comercial para simulación de celdas robóticas. [9]-----	19
Tabla 2.7	Requisitos de equipo de cómputo del escáner <i>EiScan -Pro+</i> . -----	26
Tabla 4.1	Ajuste de parámetros, para el control por botonera. -----	49
Tabla 4.2	Trama de datos para realizar operaciones en <i>VDF</i> . -----	52
Tabla 4.3	Ajuste de parámetros del <i>VDF</i> , para el control por módulo <i>RS485</i> . -----	53
Tabla 5.1	Comparativa dimensional del mecanizado en material delrin y aluminio. -----	58
Tabla 5.2	Mediciones de rugosidad (μm) en maquinado plano. -----	59
Tabla 5.3	Parámetros de corte para trayectoria curva de radio 25 mm.-----	60
Tabla 5.4	Mediciones de rugosidad en (μm) de la trayectoria curva. -----	61
Tabla 5.5	Parámetros de corte para maquinado de figura de automóvil. -----	62

CAPÍTULO I

MARCO DE REFERENCIA

1.1 ANTECEDENTES Y ESTADO DEL ARTE

Los robots *ABB IRB 6400* fueron adquiridos por el Departamento de Ingeniería Mecánica en el año 2013 como parte de una donación para uso estudiantil por parte de la empresa *Hirotec S.A.C.V*; desde su habilitación en la cual participaron estudiantes de la carrera de Ingeniería Mecatrónica del Tecnológico de Celaya se han desarrollado proyectos de licenciatura, así como de maestría. Algunos de estos proyectos son: implementación de corte por plasma, sincronización de dos brazos robóticos, la simulación de una estación virtual de videojuegos. Cabe mencionar que el Laboratorio de Manufactura cuenta con dos centros de maquinado de tres ejes y un centro de torneado, con los cuales se realizan proyectos de mecanizado.

Un centro de maquinado, como el que se ilustra en la Figura 1.1, es una máquina herramienta altamente automatizada capaz de realizar múltiples operaciones de maquinado en una instalación bajo control numérico computarizado, con mínima intervención humana. Las operaciones típicas realizadas en un centro de maquinado son el fresado y el taladrado, que utilizan herramientas de corte giratoria [1].

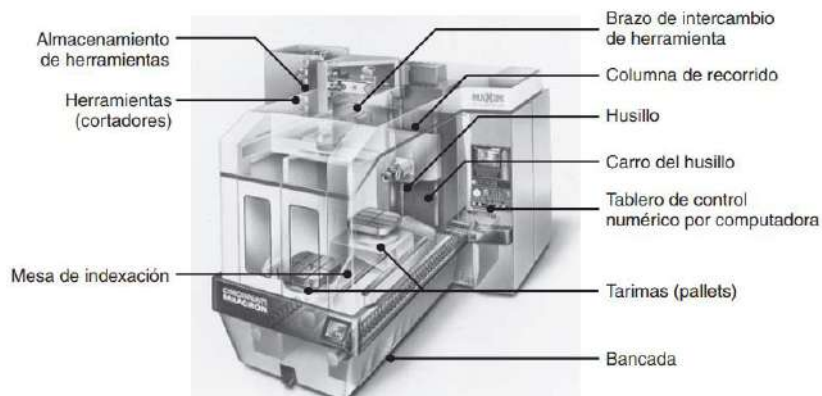


Figura 1.1 Centro de maquinado de husillo horizontal. [1]

Las máquinas herramienta de control numérico fueron desarrolladas para cumplir con los requisitos de mecanizado en contornos de piezas complejas para el sector de aviación. La compañía *Parsons* y el Instituto de Tecnología de Massachusetts (MIT) colaboraron para desarrollar la primera máquina herramienta de control numérico en 1952 [2].

En el sistema, conocido como “*THE MIT*”, se basaron en herramientas existentes que fueron modificadas con motores para mover la herramienta siguiendo puntos especificados en una cinta perforada, ver Figura 1.2. Con el tiempo, los mecanismos iniciales se mejoraron con la incorporación de computadoras analógicas, dando origen a los centros de maquinado que conocemos hoy en día.

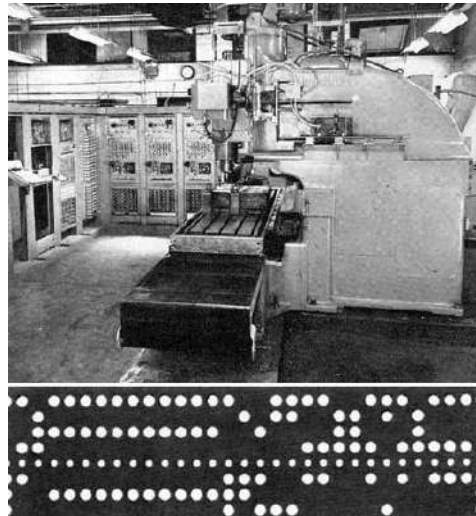


Figura 1.2 Primera fresadora CNC de Richard Kegg.

Las técnicas convencionales de eliminación de material, como el fresado CNC, han demostrado ser capaces de enfrentar casi cualquier desafío de mecanizado. Por otro lado, la principal desventaja de utilizar máquinas CNC convencionales es el área de trabajo restringida y las limitaciones en la forma de las piezas producidas. Desde un punto de vista conceptual, la tecnología de robots industriales podría proporcionar una excelente base para el mecanizado, ya que es flexible y rentable en términos de costos [3].

Frecuentemente se busca mejorar el diseño de los centros de mecanizado. Algunos aspectos importantes en la innovación de diseño de máquinas de control numérico son: impartir mayor flexibilidad de maquinado a las máquinas herramienta; aumentar su envolvente de maquinado (es decir, el espacio dentro del cual se puede realizar el maquinado); hacer una máquina más ligera [4].

El NIST (*National Institute of Standards and Technology*) logró crear una estructura de máquina herramienta verdaderamente innovadora la cual es un marco de diseño independiente octaédrico, conocida como máquina hexápoda ver Figura 1.3. Esta máquina se basa en un mecanismo conocido como plataforma *Stewart* (llamado así en honor de D. Stewart), invento desarrollado en 1966 para posicionar simuladores de cabinas de aeronaves [4].

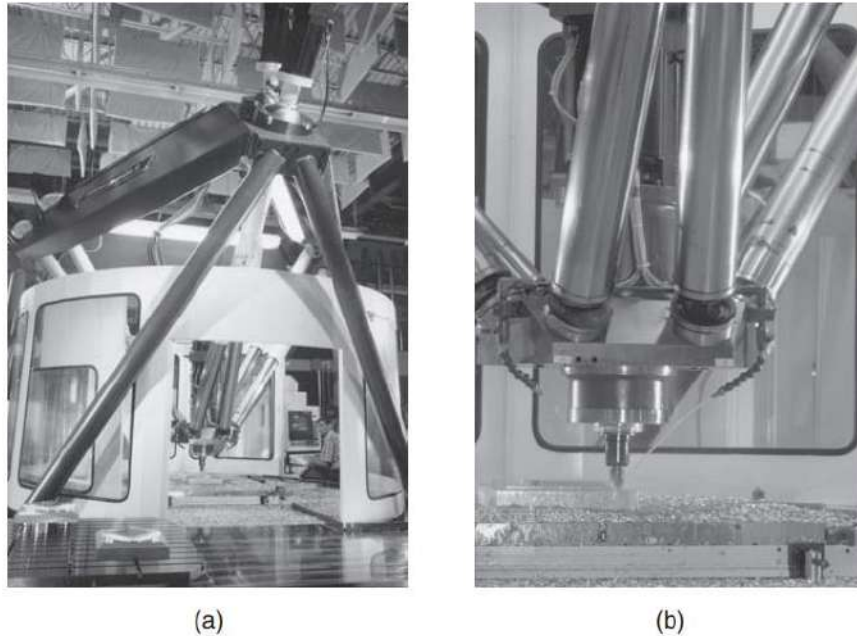


Figura 1.3 Máquina herramienta hexápoda y vista detallada de la herramienta de corte. [4]

En la maquina hexápoda la pieza de trabajo se monta sobre una mesa estacionaria; se utilizan tres pares de brazos telescópicos (puntales o piernas). Durante el maquinado el controlador acorta en forma automática algunos brazos y extiende otros, de manera que el cortador pueda seguir una trayectoria especificada.

Las máquinas hexápodas tienen alta rigidez; no son tan masivas como los centros de maquinado; cuentan con una envolvente de maquinado extensa (con mayor acceso a la zona de trabajo); tienen la capacidad de mantener la herramienta de corte perpendicular a la superficie y su flexibilidad es alta (con seis grados de libertad) en la producción de partes con diversas geometrías y tamaños sin necesidad de volver a fijar el trabajo en proceso [4]. El costo aproximado de la máquina en el año 2008 fue de \$500,000 dólares.

Por lo general, los robots industriales se utilizan en tareas de manipulación, pintura, soldadura y montaje ver Figura 1.4, ya que estos procesos requieren únicamente control de movimiento punto a punto (*PTP*) donde se definen puntos específicos sin priorizar la trayectoria de entre dichos puntos. Las máquinas herramienta CNC son los equipos de fabricación más recomendados para el maquinado de partes mecánicas, aunque tienen algunas desventajas para fabricar formas complejas, donde se requiere el control continuo de 5 ejes. Los centros de maquinado de 5 ejes son muy costosos y no tan ampliamente disponibles [5].



Figura 1.4 Operación de soldadura mediante robot *ABB*. [6]

Empresas internacionales como *KUKA Roboter GmbH* dedicada a la fabricación de robots industriales y soluciones de automatización cuentan con celdas de maquinado CNC. El robot *KUKA 120 R2700 extra-HA* de la serie *KUKA KR QUANTEC* (Figura 1.5), diseñado para requisitos de alta precisión, se usa en aplicaciones de fresado y láser. En *Asset International*, en el sur de Gales, se maquinan los grandes componentes de tuberías de plástico con un husillo fresador de alta velocidad especialmente diseñado acoplado al brazo robótico, la solución fue desarrollada por el integrador *Eugen Riexinger GmbH & Co. KG en Bad Liebenzell*



Figura 1.5 Operación de fresado mediante robot *KUKA*. [7]

La creación de una estación de maquinado CNC mediante un robot industrial presenta un amplio campo de discusión y desarrollo, obteniendo ventajas en algunos aspectos por encima de un centro de maquinado; la comparativa de discusión general se centra con relación a: el precio de la estación, la precisión, el espacio de trabajo y la calidad de maquinados.

Las máquinas herramienta CNC de varios ejes que se utilizan en el proceso de maquinado, tienen inconvenientes, primero, las máquinas herramienta de precisión son muy costosas; segundo, el modo de procesamiento es fijo, por lo que generalmente no tienen suficiente flexibilidad y capacidad de

maquinado; tercero, la configuración es compleja, por lo que es difícil formar un sistema integrado de "maquinado-medida" [8].

Por otra parte, el mecanizado con robots posee algunas ventajas únicas sobre los procesos de maquinado convencionales, aunque presenta algunos problemas a la hora de realizar operaciones. El problema más común es el mecanizado de características huecas o cavidades profundas, donde pueden producirse colisiones entre el portaherramientas y la superficie de la pieza. Una forma de superar este problema es mediante el mecanizado por capas, lo que convierte a los robots industriales en excelentes candidatos para las operaciones de mecanizado de Herramientas Rápidas (RT, por sus siglas en inglés) [3].

Otra problemática es la dificultad de controlar la trayectoria, lo cual puede llevar frecuentemente a que la estructura robótica caiga en puntos de singularidad. La segunda razón es la baja rigidez de la estructura robótica, la cual no es tan eficiente para procesos de corte de alta precisión [5].

El mecanizado robótico no reemplazará al mecanizado CNC en aplicaciones de 3-4 ejes, pero es apropiado para procesos donde el centro de mecanizado es considerado sobrecalificado en (tolerancias mayores a 0.0001 mm), mientras que un robot *KUKA KR-270* con una repetibilidad de 0.06 mm puede ser calibrado para reducir el error de pose del robot hasta una precisión de posicionamiento de 0.170 mm [9]. Actualmente, se considera que el mecanizado robótico es una alternativa viable inmediata para materiales no metálicos y para algunos metales dependiendo del grado de dureza, la eficiencia dependerá del acabado superficial requerido y complejidad de la pieza final [10].

En los últimos años, se han logrado avances significativos en la investigación de fresado robótico. Sin embargo, aún existen desafíos que requieren una atención especial. Algunas áreas clave para investigaciones futuras en el desarrollo de estaciones de maquinado son: combinar los sistemas de visión y detección de fuerza para permitir el ajuste adaptativo de la postura del robot, otro aspecto importante es abordar la generación de vibraciones en el fresado robótico para mejorar la estabilidad del sistema, detección de características tempranas y condiciones de evolución en la generación de vibración, lo que permitiría evitar vibraciones destructivas y garantizar una operación más estable y precisa [11].

1.2 PLANTEAMIENTO DEL PROBLEMA

Frecuentemente se requieren alternativas para crear piezas complejas que no son posibles de maquinar en una máquina CNC convencional de 3 o 4 ejes sin tener que desmontar la pieza; debido a que los grados de libertad de un robot son mayores así como su alcance, es posible encontrar ventaja por sobre las fresadoras CNC convencionales, además de incrementar exponencialmente el tamaño de las piezas

a maquinar con un robot, concentrándose en los sectores automotriz, aeroespacial, naval y el oficio de carpintería.

El costo de las maquinas CNC de cinco ejes es sumamente elevado y poco accesible, de esta manera al contar con robots articulados es factible generar las mismas piezas adicionando solamente una herramienta apropiada en un husillo acoplado a la brida del robot. De esta manera se implementa un sistema con seis grados de libertad y un mayor campo de trabajo, comparado con el de una máquina CNC convencional.

Este proyecto está enfocado en proporcionar la implementación de robots industriales en procesos de mecanizado, aprovechando que los brazos robóticos típicamente cuentan con 5 o 6 ejes controlados. Los robots ABB disponibles el Laboratorio de Manufactura Avanzada, son idóneos para las aplicaciones industriales debido a su precisión y versatilidad, permitiendo alcanzar el punto final de la trayectoria de forma consistente a lo largo de numerosos ciclos de movimiento. Además, las coordenadas del punto pueden ser ajustadas con gran facilidad, lo que incrementa su adaptabilidad a diferentes tareas de fabricación.

1.3 HIPÓTESIS

El uso de un robot industrial serial acondicionado permitirá el maquinado de piezas cuyas especificaciones tales como dimensiones, tolerancias, acabado superficial, tiempo ciclo logren competir contra el mecanizado por máquinas CNC.

Consecuentemente se espera configurar el robot para el mecanizado de piezas de dimensiones considerables adaptándose a diferentes requerimientos dentro de su espacio de trabajo.

1.4 OBJETIVO GENERAL

Integrar un sistema basado en un robot serial ABB IRB 6400-2.8, para el maquinado de piezas programadas en lenguaje estándar de control numérico computarizado (CNC).

1.5 OBJETIVOS ESPECÍFICOS

- Adaptar un husillo como efector final e implementar su control de velocidad.
- Establecer una metodología para la generación de programas de maquinado de piezas, a partir de sus modelos CAD o de ingeniería inversa.
- Diseñar una metodología para el proceso de maquinado mediante el sistema integrado.

1.6 ALCANCES

Generar programas de maquinado en lenguaje de programación CNC adaptados a los movimientos programados en el robot, lo que implicará hacer uso de programas de aplicación especial capaces de asociar ambos lenguajes de programación.

Implementar un protocolo de comunicación serial entre el robot y la computadora para el intercambio de datos y el control de la velocidad del husillo.

1.7 LIMITACIONES

Por la dureza de los materiales y el tamaño del husillo es necesario trabajar con materiales de baja dureza y buena maquinabilidad como son: aluminio, cobre, latón, polímeros y madera.

El acabado superficial está afectado por:

- Estado físico del robot.
- Rigidez de la mesa de trabajo.

CAPÍTULO II

MARCO TEÓRICO

2.1 SISTEMAS DE CONTROL NUMÉRICO

En la década de 1970, se desarrollaron máquinas herramienta controladas numéricamente por computadora (NC) utilizando minicomputadoras como unidades de control. Una máquina herramienta CNC típica consta de tres unidades fundamentales: la unidad mecánica de la máquina herramienta, las unidades de potencia (motores y amplificadores de potencia) y la unidad CNC [12].

El principio operativo del CN es controlar el movimiento del cabezal de trabajo en relación con la pieza de trabajo y la secuencia en la cual se realizan los movimientos [1]. Donde cada dato del programa representa una posición relativa entre el cabezal y la pieza de trabajo.

En un sistema CNC basado en multiprocesadores, se pueden agregar módulos de CPU adicionales al CNC para ampliar su inteligencia y funciones. Por ejemplo, en un centro de mecanizado multi-ejes, como el que se muestra en la Figura 2.1, [12].

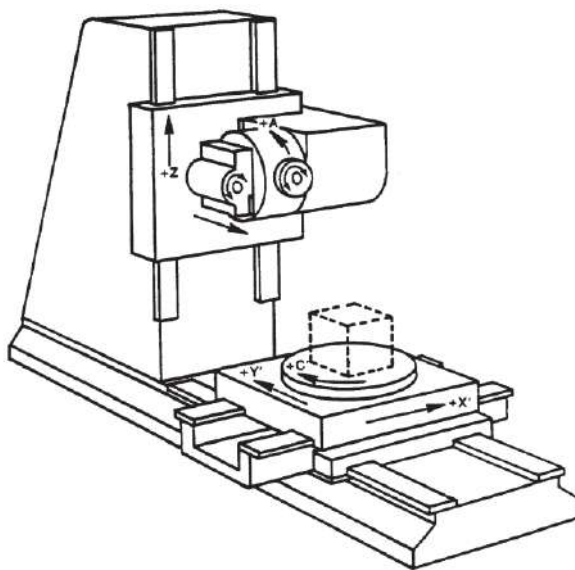


Figura 2.1 Máquina multi eje CNC. [12]

Los programas de control numérico están escritos en un lenguaje estándar internacional (código G y M) que puede ser interpretado por cualquier centro de maquinado, el código de control numérico se transfiere a la unidad en el formato *ASCII*.

2.1.1 Estructura del programa:

Para especificar la posición, se programa un sistema de coordenadas cartesianas, el cual consiste en un sistema de tres ejes (X, Y, Z) donde se pueden agregar tres ejes rotatorios (A, B, C) respectivamente. El eje Z siempre está alineado con la dirección del husillo. El movimiento primario en el eje X normalmente es paralelo a la dimensión más larga de la mesa principal de la máquina herramienta. El eje Y, normalmente es paralelo a la dimensión más corta de la mesa principal de la máquina herramienta [12].

El programa NC se compone de bloques, y cada bloque se compone de varias palabras. Una palabra comienza con un carácter seguido de números que representan un comando específico para la máquina herramienta. La palabra que comienza con G representa las funciones preparatorias, y las palabras M son las funciones diversas en el programa (NC). T representa el número de herramienta. Las letras X, Y, Z seguidas de los escalares representan las longitudes de movimiento en los ejes designados. Cada bloque comienza con la letra N seguida del número de secuencia del bloque [12]. Un bloque de programa (NC) típico como se muestra en la ecuación 2.1. La Tabla 2.1 muestra las direcciones principales del control numérico.

$$N0040 G91 X25.00 Y10.00 Z - 12.55 F150 S1100 T06 M03 M07 \dots (2.1)$$

Tabla 2.1 Direcciones principales del NC.

DIRECCIÓN	DESCRIPCIÓN
N	Número de bloque
G	Funciones preparatorias
M	Funciones Diversas
T	Número de herramienta
X, Y, Z	Coordenadas
F	Velocidad de avance
S	Velocidad del husillo

Los dos comandos fundamentales de interpolación, que se encuentran en cada unidad CNC comercial, se explican a continuación. Interpolación lineal: el código designado es G01, se utiliza en trayectorias rectas indicando las coordenadas iniciales y finales de la trayectoria. Ver Figura 2.2 por Altintas, (2001).

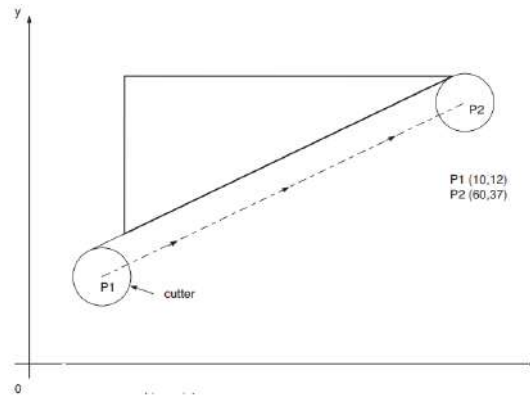


Figura 2.2 Trayectoria lineal G01 [12].

Interpolación circular: los códigos designados son G02 para giro a derecha, G03 para giro a izquierda, se utilizan en trayectorias circulares indicando la coordenada inicial y final, así como el centro o el radio del círculo. Ver Figura 2.3 por Altintas, (2001).

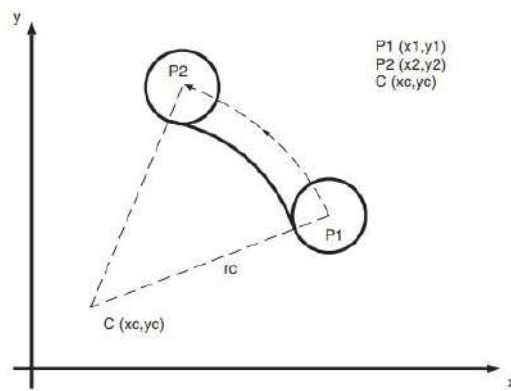


Figura 2.3 Trayectoria circular G02 [12].

Para la generación de los códigos de control numérico se programan las trayectorias y los distintos parámetros de corte. Existen diferentes técnicas de programación:

- Programación manual: la cual se usa en trabajos sencillos en operaciones de contorno o de taladrado donde el programador puede configurar la trayectoria de la herramienta.
- Programación de piezas asistida por computadora: el primer lenguaje fue “*Automatically programmed tooling*” (APT) el cual consiste en definir la geometría de la pieza y especificar la trayectoria (puntos, líneas, círculos) consecutivamente generar la secuencia de operaciones a realizar, al finalizar el programador introduce el programa a la computadora.
- Programación CAD/CAM: es la técnica más actualizada que consiste en utilizar programas especializados donde el programador visualiza la pieza en un sistema 3D, generando la secuencia

de operaciones y visualizando inmediatamente la trayectoria, así como el maquinado en una simulación. Para finalizar, la computadora realiza el postproceso de la información, y genera el código de control numérico correspondiente a una máquina en particular.

2.1.2 Selección de centro de maquinado

La selección del tipo de centro de maquinado depende de diversos factores: el tamaño de las piezas a mecanizar, el tipo de operaciones de maquinado, la precisión dimensional que se requiere y el presupuesto económico. Cabe mencionar que los centros de maquinado son capaces de manejar una amplia variedad de tamaños y formas de partes de manera eficiente, económica y repetitiva; tienen alta precisión dimensional y tolerancias del orden de 0.0025 mm (0.0001 pulgada), [4]. Dentro de las características esenciales con las que debe contar una estación o un centro de maquinado se encuentran:

- Rigidez: es un factor que se define en función de los materiales de los componentes, de la configuración y diseño de los dispositivos de la estación o centro de maquinado; la rigidez afecta directamente a la precisión dimensional del mecanizado de las piezas.
- Amortiguamiento: es un factor importante para reducir las vibraciones durante el proceso, ya que la vibración afecta directamente al acabado superficial del maquinado de las piezas.
- Distorsión térmica: la distorsión térmica de los componentes de la estación afecta directamente a la precisión dimensional de las piezas, se busca que los materiales con que se construye la estación tengan un bajo coeficiente de dilatación térmica.

2.1.3 Operación de fresado

Es una operación donde se utiliza una herramienta de corte rotativa llamada fresa la cual cuenta con varios filos que remueven material dando la forma deseada. Mediante las operaciones de fresado se pueden realizar perfiles, contornos, careados, ranuras y cajas Figura 2.4.

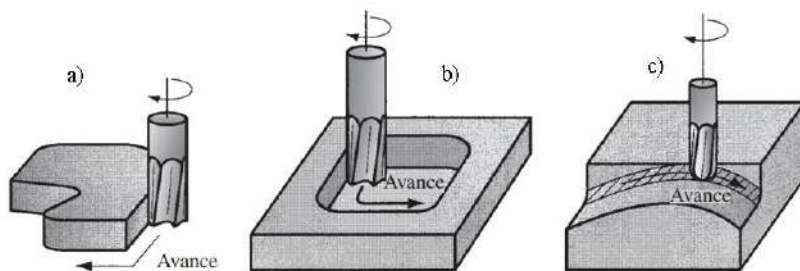


Figura 2.4 Operaciones de fresado a) Perfil, b) Caja, c) Contorno. [1]

Parámetros de corte del fresado

La velocidad de corte (V_c) se calcula en función del diámetro de la fresa y de su velocidad de giro mediante la siguiente formula:

$$V_c = \frac{N * \pi * D}{1000} \quad \dots (2.2)$$

Donde:

N -especifica las revoluciones por minuto.

V_c -velocidad de corte (mts/min).

D - diámetro de la herramienta (mm).

La velocidad de avance (F), se encuentra en función de la velocidad del husillo, del avance por diente y del número de dientes mediante la siguiente formula:

$$F = N * f * Z \quad \dots (2.3)$$

Donde: N - revoluciones por min.

f - avance por diente (mm/rev).

Z - número de filos de la herramienta.

(En el anexo 1 se incluyen tablas de parámetros de corte recomendados).

2.2 ROBÓTICA INDUSTRIAL

Según la *RIA (Robot Industry Association)*, la cual es una organización que agrupa a empresas, instituciones y organismos relacionados con la fabricación, distribución, aplicación e investigación de la robótica, un robot es *“Una máquina programable, de uso práctico, que se puede programar para llevar a cabo una serie de tareas automáticamente, con o sin supervisión humana”*. Es decir, un robot es una herramienta tecnológica que permite automatizar procesos y tareas, reducir errores y aumentar la eficiencia.

Para el mercado asiático, específicamente Japón, un robot industrial es *“cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación”* [13].

Otra definición interesante según J. Pérez Turiel, *“Es una máquina de manipulación automática reprogramable y multifuncional con tres o más ejes que puedan posicionar y orientar materiales, piezas,*

herramientas o dispositivos espaciales para la ejecución de diversas tareas en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento" [14].

Existen varias maneras de clasificar a los robots, conforme a su generación cronológica, de acuerdo con su estructura física, conforme a sus aplicaciones, etc. La clasificación dependerá específicamente del autor, en la Tabla 2.2 se muestra la clasificación brazos robóticos conforme a su generación.

Tabla 2.2 Clasificación de brazos robóticos conforme a su generación. [13]

No.	Clasificación	Definición
1	Primera generación	Repite la tarea programada sin tomar en cuenta las alteraciones en su entorno.
2	Segunda generación	Puede localizar, clasificar y detectar alteraciones en su entorno y adaptar sus movimientos en consecuencia.
3	Tercera generación	Pose la capacidad de planificación automática de tareas mediante microprocesadores avanzados.

De acuerdo con su estructura robótica se clasifican como robots industriales (seriales) los que presentan una configuración de eslabones conectados mediante articulaciones para servir a un propósito como soldar, pintar, ensamblar, mecanizar. Los robots no industriales se emplean para tareas especiales. La Tabla 2.3 proporciona la clasificación general conforme a la estructura robótica.

Tabla 2.3 Clasificación general de los robots. [15]

No.	Clasificación	Definición
1	Robot industrial (serie):	Eslabones conectados programados para realizar tareas. Se pueden subdividir de acuerdo con: • Aplicación. • Sistema de coordenadas. • Método de control y programación.
2	Robot no industrial	• Vehículos guiados automáticamente (AVG). • Robots caminantes (antropomórficos). • Robots paralelos.

Las máquinas-herramienta de control numérico computarizado (CNC) se distinguen de los robots porque éstos tienen capacidades de reprogramación y multifuncionalidad [15]. La siguiente Tabla 2.4 muestra la clasificación de acuerdo con las áreas donde se utilizan las estaciones de mecanizado robótico junto con su operación principal.

Tabla 2.4 Sectores de aplicación de las estaciones de mecanizado robótico. [10]

Segmento del Usuario	Proceso	Producto
Aeroespacial	Rectificado, pulido, taladrado, corte.	Alabe de turbina, estructuras segmentos de ala.
Automotriz	Desbarbado, rectificado, taladrado, fresado, corte.	Motores, paneles de carrocería, puertas, parachoques, matrices de estampado, núcleos de arena.
Moda	Fresado, lijado.	Moldes de maniquíes.
Fundición	Desbarbado, acabado, fresado	Moldes.
Marino	Fresado	Barcos, alabes de turbina.
Medico	Rectificado, pulido.	Prótesis.
Entretenimiento	Fresado.	Mover accesorios.
Plásticos	Fresado.	Moldes, cascos.
Madera	Fresado	Moldes, muebles, molduras.

2.3 RECURSOS PARA EL DESARROLLO DEL PROYECTO

La marca de robots *ABB* cuenta con una amplia gama de modelos, Figura 2.5. El brazo robótico *IRB 6400 ABB* es de propósito industrial, posee seis grados de libertad y se programa utilizando el lenguaje *Rapid*, comúnmente se utiliza en aplicaciones de manipulación de objetos, soldadura, pintura, ensamblaje, etc. Al ser un robot industrial cuenta con diversas estructuras de programación, tiene una precisión de trayectoria lineal de 2.1-2.5 mm, además de protocolos de comunicación industrial que son importantes para controlar otros dispositivos de manera serial o mediante salidas digitales del controlador.

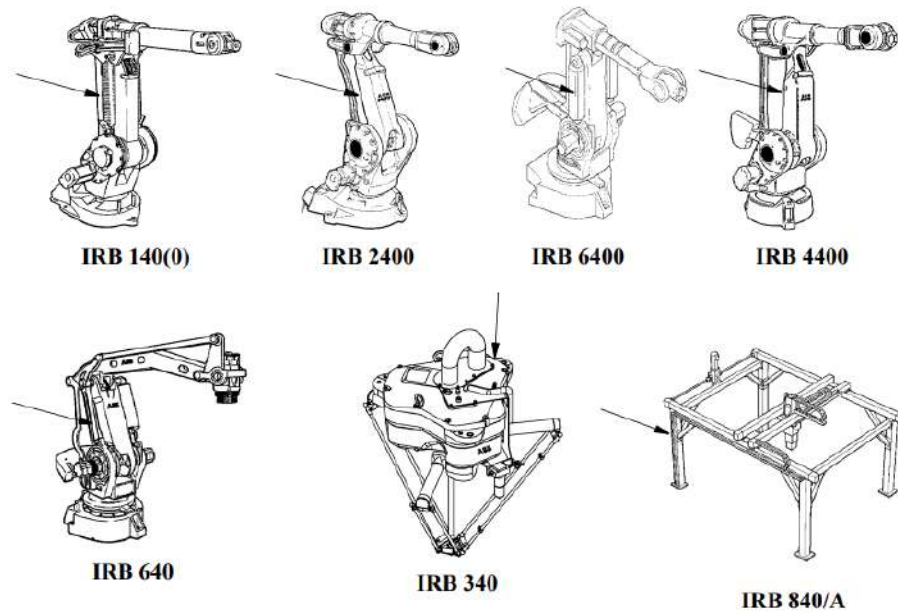


Figura 2.5 Modelos de robot ABB.

2.3.1 Generalidades del robot ABB IRB 6400

La estructura robótica consta de dos partes: el manipulador y el controlador (Figura 2.6). El modelo del robot utilizado en el proyecto es el *ABB-IRB-6400-2.8-120*, el Post-Procesador del controlador es *S4C* con sistema *Baseware 3.0*.

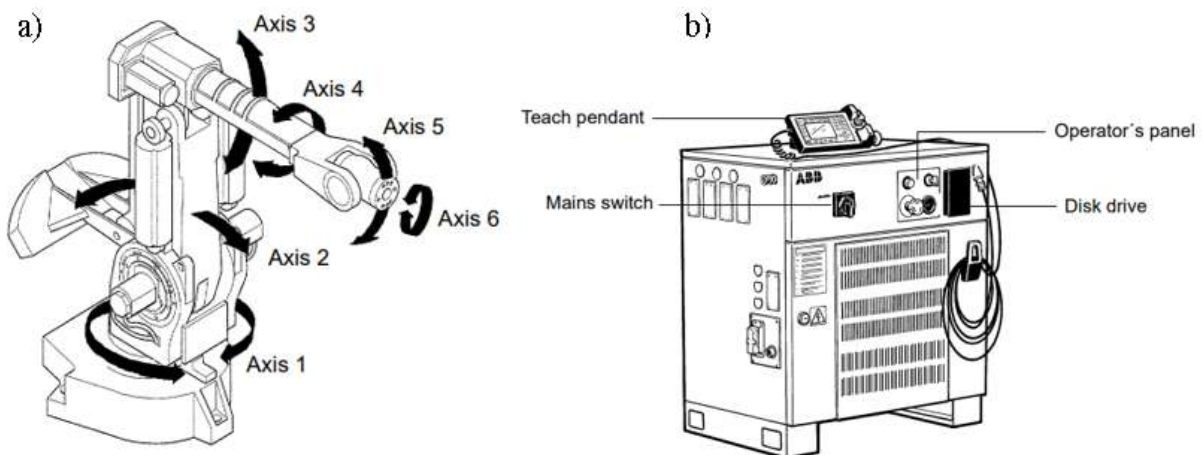


Figura 2.6 Componentes, a) Manipulador, b) Controlador *IRB 6400*.

- El Manipulador o brazo robótico: tiene un peso de 2200kg, y una capacidad de carga de 120kg con una altura hasta de 2943 mm, ver Figura 2.7.

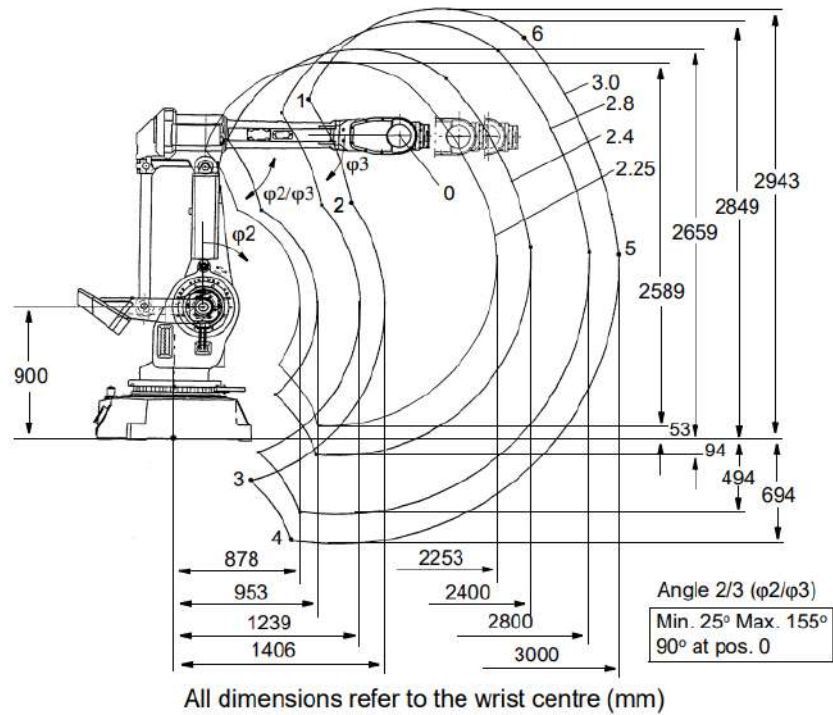


Figura 2.7 Dimensiones del brazo robótico IRB 6400.

El rango de movimientos del brazo robótico se describe en la Tabla 2.5.

Tabla 2.5 Rango angular de movimientos del *IRB 6400*.

Tipo de movimiento	Rango de movimiento
Eje 1- Rotación	+180° a -180°
Eje 2- Traslación	+70° a -70°
Eje 3- Traslación	+105° a -28°
Eje 4- Giro	+300° a -300°
Eje 5- Rotación	+120° a -120°
Eje 6- Giro	+300° a -300°

- Controlador: es el dispositivo electrónico encargado de los movimientos del brazo robótico, aquí se tiene el Panel del operador con los modos de funcionamiento, ver Figura 2.8, además del *Teach Pendant*, el cual es un dispositivo para controlar al brazo de forma remota, ver Figura 2.9.

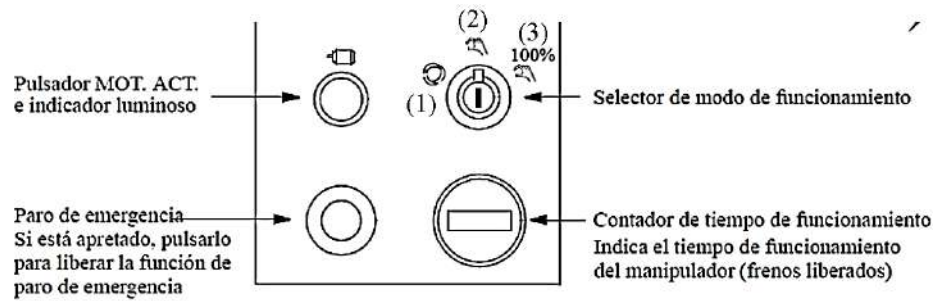


Figura 2.8 Panel de control.

El Panel de control cuenta con un selector de tres posiciones de funcionamiento:

- Modo automático (1), se utiliza para la producción en ciclo.
- Modo manual (2), sirve para la manipulación a velocidad de 250 mm/s.
- Modo manual 100% (3), permite la manipulación a la velocidad máxima programada, solo es recomendable para operadores experimentados.

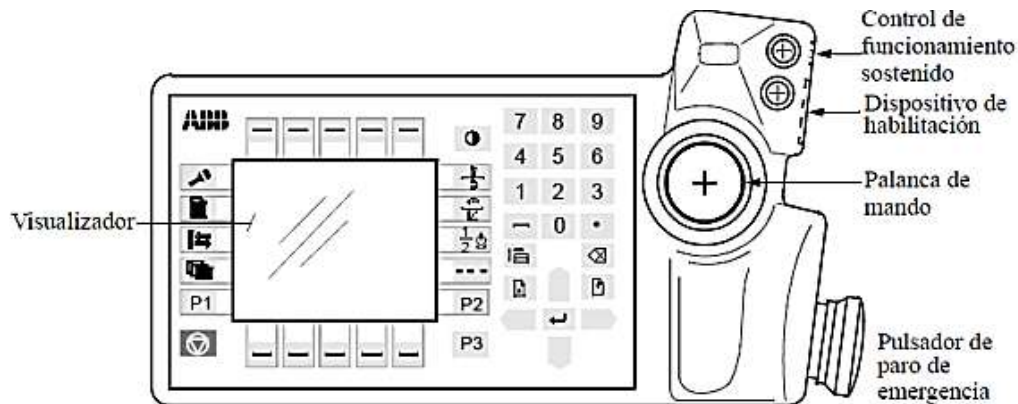


Figura 2.9 Teach Pendant.

Mediante el Teach Pendant se puede mover el brazo robótico con la palanca de mando de manera manual además de configurar programas, entradas y salidas digitales.

El modelo *IRB 6400-2.8* utilizado en el proyecto cuenta con un rendimiento según la ISO 9283, con carga nominal y velocidad de 1 m/s en el plano obteniendo los siguientes valores característicos [16]:

- Repetibilidad de pose unidireccional: $PR = 0.1 \text{ mm}$
- Precisión de trayectoria lineal: $AT = 2.1 - 2.5 \text{ mm}$
- Repetibilidad de trayectoria lineal: $TR = 0.5 - 0.8 \text{ mm}$
- Tiempo mínimo de posicionamiento, hasta 0.4 mm de la posición: $0.2 - 0.3 \text{ segundos}$.

2.3.2 Programación

La programación del robot implica seleccionar las instrucciones y algoritmos necesarios para que el brazo robótico realice los movimientos programados; existen dos maneras programar el modelo *IRB 6400*:

- Programación *Online/Automatic*: La programación *online* implica interactuar directamente con el robot mientras se encuentra en funcionamiento. Se puede efectuar a través de una interfaz de usuario especializada, como el *Teach Pendant* o una interfaz de software, Figura 2.10.

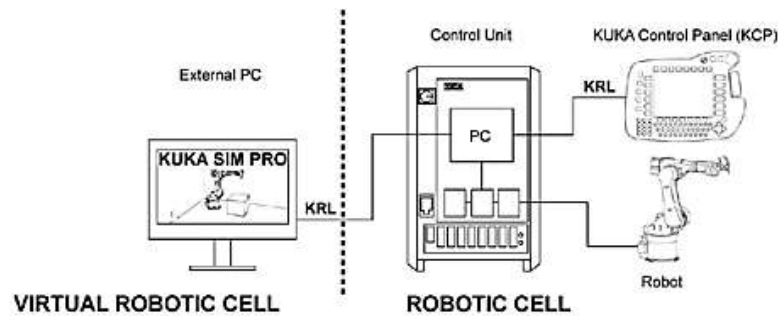


Figura 2.10 Programación *Online/Automatic*. [17]

- Programación *Offline/Manual*: La programación *Offline* se realiza sin la necesidad de que el robot se encuentre en funcionamiento. Generalmente se realiza utilizando un software especializado en una computadora independiente del controlador, donde se puede crear, editar y simular programas de robot en lenguaje *Rapid*, Figura 2.11.

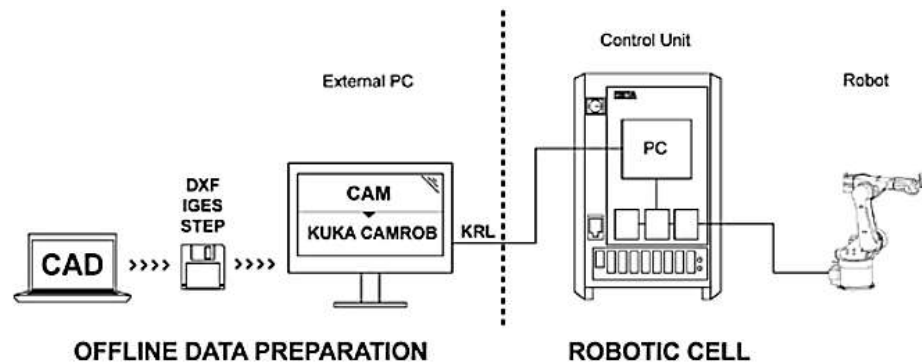


Figura 2.11 Programación *Offline/ Manual*. [17]

La programación *Offline/Manual* es similar a la forma tradicional de trabajar con máquinas CNC (Control Numérico por Computadora), es decir, transfiriendo datos desde el CAD hacia el software CAM y luego como código NC (Control Numérico), también conocido como código G, hacia una máquina

fresadora CNC. Además, el fresado con robot requiere el uso de un Post-Procesador cinemático/simulador [17].

En la Tabla 2.6 se enumeran varios ejemplos de software de simulación cinemática relacionados en el mecanizado con robots disponibles actualmente en el mercado:

Tabla 2.6 Softwares comercial para simulación de celdas robóticas. [10]

Compañía	Software
<i>ABB</i>	<i>RobotWare Machining Force Control (FC)</i> - Control de Fuerza (FC) - Precisión y Cambio de Velocidad (FC SpeedChange).
<i>FANUC</i>	<i>ROBOGUIDE, AccuDeflash.</i>
<i>KUKA</i>	<i>KUKA.CAMRob</i> : Ofrece la conversión de datos CAM en archivos binarios y programas KRL. Transfiere los datos convertidos desde una PC externa al controlador del robot.
<i>MOTOMAN</i>	El software <i>MotoSim EG de Motoman</i> simula la trayectoria del robot generada a partir del código G en el mundo virtual en una PC, lo que permite modelar programas sin conexión y luego descargarlos al controlador del robot.
<i>JABEZ T</i>	<i>Robotmaster</i> : Convierte el código CNC de Mastercam basado en los datos CAD de cada pieza en comandos para el robot.
<i>KMT</i>	<i>CamPro</i> : Software de programación fuera de línea basado en PC que permite a los fabricantes convertir de manera fluida los trayectos de recorte y fresado CAM en programas completos para robots de 6 ejes.

2.3.3 Software para simulación y programación cinemática

RobotDK es un software de programación *Offline/Manual* que genera con facilidad programas robóticos de diversos proyectos de fabricación, como fresado, soldadura, manipulación de objetos, sincronización de robots, configuración de cintas transportadoras, embalaje y etiquetado, paletización, pintura, calibración entre algunos más; el software cuenta con Post- Procesadores que soportan los controladores comerciales: *ABB RAPID (MOD/PRG)*, *FANUC LS (LS/TP)*, *KUKA KRC/IIWA (SRC/java)*, *MOTOMAN Inform (JBI)*, *Universal Robots (URP/script)*, Figura 2.12.



Figura 2.12 Página web de *RobotDK*.

2.3.4 Software para control numérico

Los softwares CAD-CAM son herramientas utilizadas en el diseño y fabricación de productos, especialmente en el ámbito de la ingeniería y la fabricación asistida por computadora (CAM). A continuación, se describen sus componentes principales:

- CAD (Diseño Asistido por Computadora): El software CAD se utiliza para crear modelos digitales tridimensionales (3D) o bidimensionales (2D) de productos, componentes o estructuras. Estos modelos pueden ser muy detallados y precisos, permitiendo a los diseñadores e ingenieros visualizar y desarrollar sus ideas antes de llevarlas a la producción real.
- CAM (Fabricación Asistida por Computadora): Por otro lado, el software CAM se utiliza para convertir los modelos digitales creados en CAD en instrucciones de fabricación que pueden ser interpretadas por máquinas herramienta controladas por computadora, como tornos CNC, fresadoras o máquinas de corte por láser. El software CAM permite generar rutas de herramientas, trayectorias de movimiento y secuencias de operaciones de mecanizado o fabricación basadas en el diseño CAD. Esto incluye estrategias de mecanizado, optimización de trayectorias, simulación de procesos y generación de códigos de control numérico (código G) para las máquinas herramienta.

Autodesk Fusion es una plataforma de software CAD, CAM, CAE y de circuitos impresos de modelado 3D basada en la nube, para el diseño y la manufactura de productos. Es un software versátil didáctico y eficaz que permite realizar proyectos con licencias estudiantiles, su apartado de manufactura cuenta con múltiples comandos que permiten la programación de trayectorias complejas específicas para operaciones que solo se realizan en máquinas CNC de 5-6 ejes, ver la Figura 2.13.



Figura 2.13 Página web del programa *Fusion 360* de Autodesk.

2.4 CABEZAL DE FRESADO

El cabezal es un dispositivo que contiene al husillo y sujetador del cortador. En máquinas verticales, la cabeza puede fijarse o ajustarse verticalmente y girarse en un eje vertical sobre la columna para maquinarse diferentes superficies, Figura 2.14, [4].

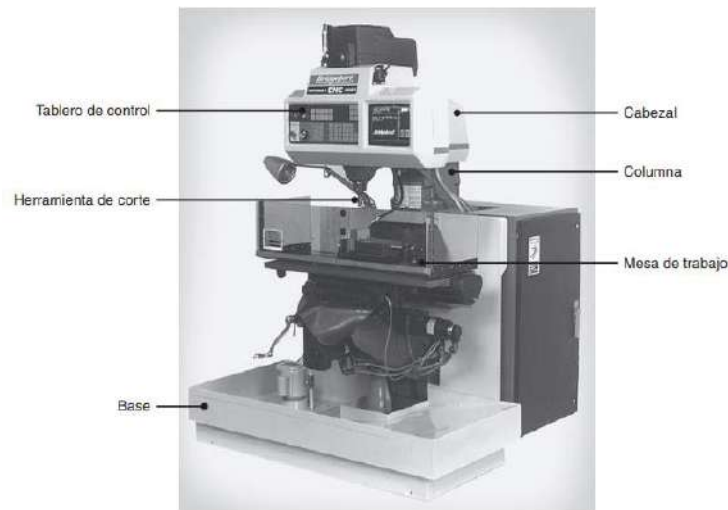


Figura 2.14 Fresadora de husillo vertical. [4]

Un husillo es un componente esencial en una máquina herramienta de corte, en un torno o una fresadora, se encarga de girar la herramienta de corte o la pieza de trabajo. Es un eje central alrededor del cual se monta y gira la herramienta o la pieza para realizar operaciones de mecanizado.

El husillo puede ser accionado por diferentes tipos de motores, como motores eléctricos, motores hidráulicos o motores neumáticos, dependiendo de los requisitos de la aplicación y el tipo de máquina herramienta.

Para el proyecto se utilizó un motor eléctrico trifásico de la marca *Zhong Hua Jiang* con husillo de 800W ver Figura 2.15, el cual permite el control continuo de velocidad hasta 24,000 rpm. Opera con un voltaje

de trabajo 220V y tiene 1.5KW de potencia, es adecuado para operaciones de maquinado, además el sistema permite el uso de refrigerantes.

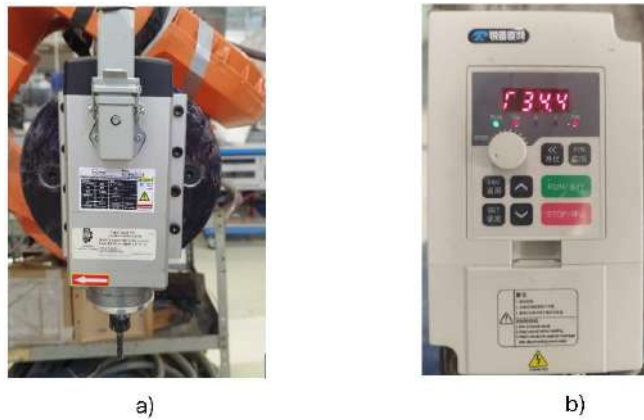


Figura 2.15 Motor *Hua jiang* y variador de frecuencia.

El variador de frecuencia, *H100 series High Performance* (Variable Frequency Drive, VFD por sus siglas en inglés), es un dispositivo electrónico utilizado para controlar la velocidad de motores eléctricos; al ajustar la frecuencia y la tensión suministrada al motor, permite controlar la velocidad de forma precisa y eficiente, lo que resulta en un consumo de energía más bajo y una mayor flexibilidad en la operación de la maquinaria industrial, ver Figura 2.16.



Figura 2.16 VDF *H100 series*. [18]

2.3.6 Funcionamiento básico del VDF

La configuración del variador de frecuencia se realiza conforme al manual de operación *H100 Series High Performance Vector Control* [18].

El variador de frecuencia toma la alimentación de corriente alterna (CA) de la red eléctrica y la convierte en corriente continua (CC) mediante un rectificador. Luego, la corriente continua se convierte nuevamente en corriente alterna a la frecuencia y voltaje deseado mediante un inversor de frecuencia.

2.3.7 Control de velocidad

Al reducir la frecuencia suministrada al motor, se reduce su velocidad, y viceversa. Esto permite ajustar la velocidad del motor según las necesidades de la aplicación sin cambiar físicamente la configuración mecánica del sistema. El variador cuenta con tres modos de control ver Figura 2.17:

- Habilitando el modo manual: Se ajusta la perilla y botones de arranque de velocidad.
- Habilitando la entrada RS-485: Por medio del protocolo de comunicación *Modbus*.
- Habilitando las entradas digitales: Se ajustan los parámetros para recibir señales externas de actuadores eléctricos, permite escoger la lógica de control (NPN o PNP).

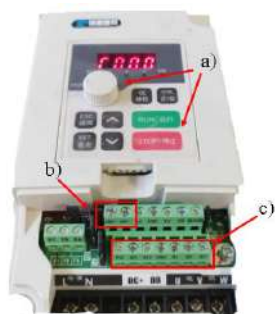


Figura 2.17 Módulos de control de velocidad, a) Manual, b) RS-485, c) Entradas digitales.

Para cada modo de control, es necesario configurar los parámetros de función de acuerdo con el tipo de comunicación. En la Figura 2.18 se muestra el diagrama del circuito principal del variador.

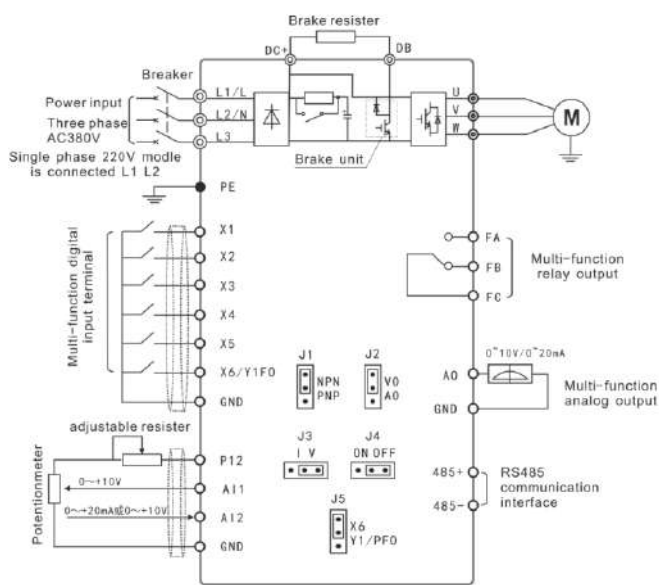


Figura 2.18 Circuito del variador H100. [18]

2.3.8 Protocolo de comunicación *Modbus*

Es un estándar de comunicación serie, se utiliza ampliamente en la industria para la comunicación entre dispositivos electrónicos, como *PLC*'s, sensores y actuadores. Es un protocolo maestro/esclavo, lo que significa que un dispositivo maestro (como un *PLC* o un sistema de control) puede solicitar y recibir datos de uno o varios dispositivos esclavos (sensores, actuadores, etc.), *Modbus* solo permite un maestro y hasta 247 esclavos [19].

El protocolo *Modbus* tiene diferentes variantes, incluyendo *Modbus RTU (Remote Terminal Unit)* que utiliza una comunicación serie de baja velocidad, *Modbus ASCII (American Standard Code for Information Interchange)* que codifica los datos en caracteres *ASCII* legibles [20].

Modbus RTU RS485

Modbus puede implementarse tanto a través del estándar *RS-485* o *RS-232*, dependiendo de las necesidades específicas de la aplicación. *Modbus RTU* utiliza el estándar *RS-485* para la transmisión diferencial de datos entre dispositivos. Los dispositivos se comunican en un esquema de red multipunto o en su defecto diferencial, donde uno o varios dispositivos pueden compartir el mismo bus *RS-485*. El protocolo *Modbus RTU* define cómo se estructuran los mensajes de datos para la comunicación entre dispositivos [20].

Modbus RTU RS-232

Utiliza el estándar *RS-232* para la transmisión serie. A diferencia de *Modbus RTU*, *Modbus ASCII* codifica los datos en caracteres *ASCII* legibles por humanos como cadenas de caracteres, lo que lo hace útil para la depuración y la comunicación con dispositivos serie mediante herramientas de terminal. Sin embargo, debido a las limitaciones de distancia y la naturaleza unidireccional de *RS-232*, *Modbus ASCII* sobre *RS-232* se utiliza generalmente en aplicaciones de corto alcance y punto a punto [20].

2.5 INGENIERÍA INVERSA

La creación de componentes mecánicos dentro de la ingeniería involucra diversos procesos como diseñar, modelar, ensamblar, simular, fabricar y optimizar entre otros, se pueden agrupar dichos procesos y clasificar en dos metodologías:

- Ingeniería directa es el proceso tradicional de diseño que comienza con la conceptualización de modelos con base en la experiencia y conocimiento del ingeniero, se desarrollan especificaciones de diseño como dibujos, planos y ensambles.

- Ingeniería inversa es un proceso de medición, análisis y pruebas para reconstruir la imagen especular de un objeto o recuperar un evento pasado, es un mapa que guía hacia la reconstrucción y reproducción [21]. Respecto a la creación de modelos 3D se define como: el proceso de obtener un modelo CAD geométrico a partir de puntos 3D adquiridos mediante escaneo/digitalización de partes/productos existentes sin especificaciones del producto.

Las razones por las que se decide utilizar ingeniería inversa, comúnmente es por la dificultad de medir superficies curvas, un ejemplo es la creación de prótesis o modelado de partes antropomórficas, otro caso habitual es cuando no se cuenta con la documentación del producto y es necesario replicarlo.

El proceso de ingeniería inversa consiste en tres fases como se muestra en la Figura 2.19. Las tres fases principales son escaneo, procesamiento de puntos y desarrollo de modelos geométricos específicos de la aplicación.

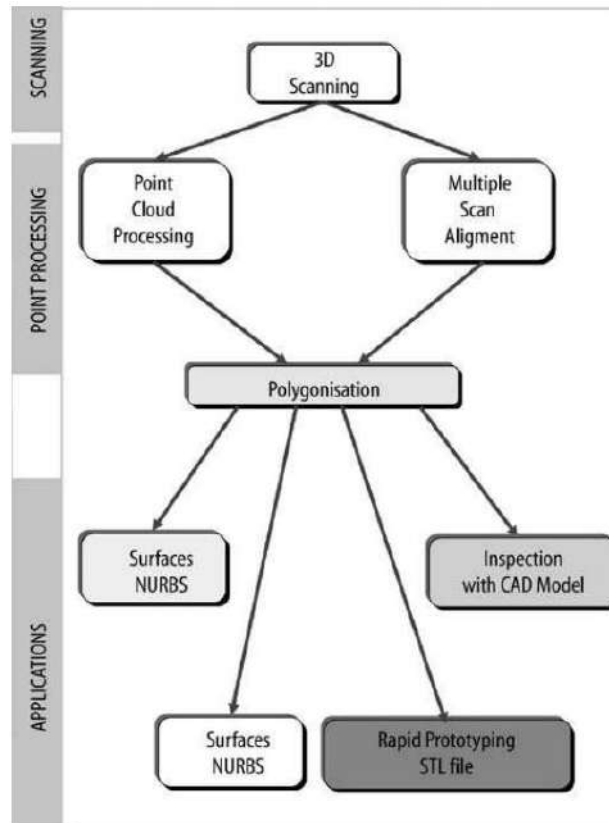


Figura 2.19 Proceso de ingeniería inversa. [22]

La selección de los dispositivos de escaneo dependerá directamente de la precisión requerida, tamaño del objeto, material y el tipo de superficie. Existen dos maneras de realizar el proceso de ingeniería inversa:

- Escaneo por contacto: se emplean sondas directamente sobre la superficie donde se genera cada punto de la pieza automáticamente, es necesario controlar la presión de contacto entre la sonda y la pieza.
- Escaneo sin contacto: permiten capturar los datos de la superficie sin necesidad de contacto físico, los dos tipos más comunes son láser y óptico los cuales permiten capturar una nube de puntos y transformarla en una malla de polígonos triangulares.

El escáner *EiScan -Pro+* de la empresa *Shining 3D* (Figura 2.20) se utilizó en el proyecto para la obtención de modelos tridimensionales de piezas de maquinado. Utiliza tecnología de luz estructurada, además de proporcionar el modo de escaneo manual rápido ideal para superficies extensas.



Figura 2.20 Escáner *EiScan -Pro+*.

La Tabla 2.7 muestra los requisitos del equipo de cómputo para poder utilizar el escáner.

Tabla 2.7 Requisitos de equipo de cómputo del escáner *EiScan -Pro+*.

Modelo	Requerimientos
Sistema Operativo	Win 10 de 64 bits
Procesador	<i>I5 3er d gen / I7-8700</i> o superior.
Tarjeta Grafica	<i>QUADRO P100 / NVIDIA GTX1080</i> o superior.
Puerto USB	Puerto USB 2.0 o 3.0
RAM	8 Gb o superior.

CAPÍTULO III

METODOLOGÍA Y DESARROLLO DEL PROYECTO

En el presente capítulo se describe la metodología utilizada para abordar las actividades programadas, una vez que se han identificado y valorado los dispositivos y componentes del proyecto. En la Figura 3.1 se ilustra a manera de diagrama la metodología de desarrollo del proyecto, realizando previamente la recopilación de información sobre estaciones de mecanizado robótico.

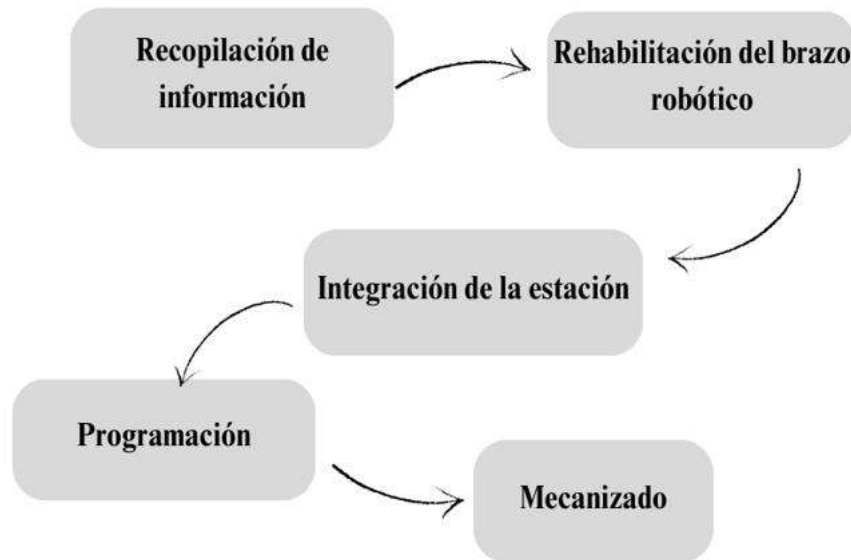


Figura 3.1 Metodología del proyecto.

3.1 REHABILITACIÓN DEL MANIPULADOR

Para la puesta en marcha del robot asignado, modelo *IRB 6400* con el número de serie 64-14775, se verificaron sus componentes de cableado, el estado físico del *Teach Pendant* y las baterías de respaldo del robot al energizarlo.

En este proceso se consideró el trabajo previo de licenciatura de Carreón [23], sobre la rehabilitación de los brazos robóticos que se encuentran en el taller de manufactura, siguiendo su metodología de inicio de operaciones, ver Figura 3.2 [23].

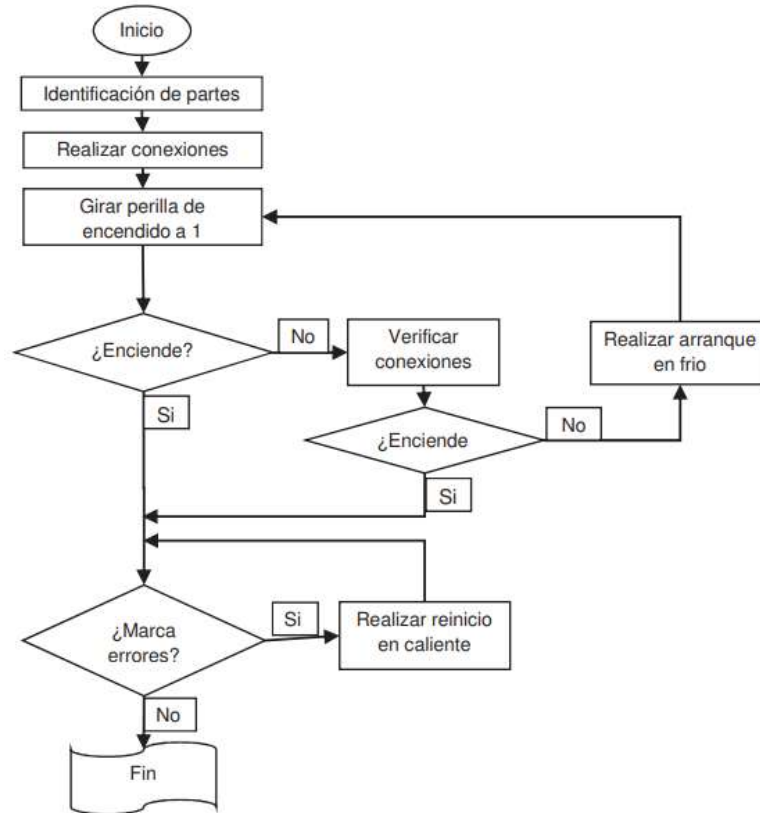


Figura 3.2 Metodología para puesta en marcha. [23]

3.1.1 Puesta en marcha

Los manuales son el material de apoyo para la instalación, configuración y solución de posibles fallas, ver Figura 3.3.

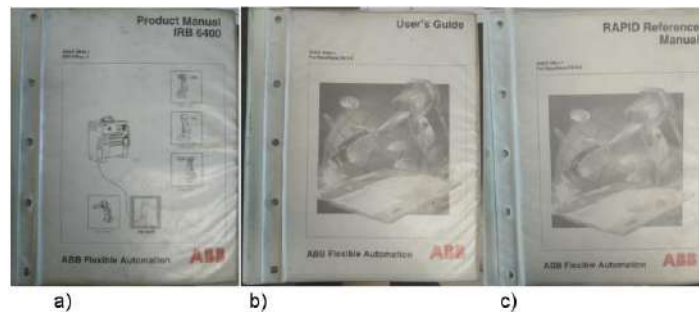


Figura 3.3 Manuales IRB 6400, a) Producto, b) Usuario, c) Rapid.

Debido a que el brazo robótico se mantuvo fuera de operaciones durante un periodo de tiempo prolongado, fue necesario realizar un arranque en frío (ver anexo 2). Para realizar el arranque es necesario contar con los archivos de configuración del controlador, correspondientes al número de serie 64-14775

y al sistema *baseware 3.1*. Anteriormente el brazo robótico funcionaba con unidades de almacenamiento tipo *disquete 3.5*, actualmente el controlador cuenta con una unidad de sustitución (Figura 3.4) que permite identificar las particiones dentro de una memoria *USB* en varias unidades de almacenamiento tipo *disquete*.



Figura 3.4 Unidad de sustitución USB.

Floppy Manager (Figura 3.5), es un software que permite reemplazar, leer y escribir archivos en memoria *USB* particionada en cien unidades de *disquetes* con capacidad de 1.39MB, es importante ejecutar el programa como administrador durante su uso.

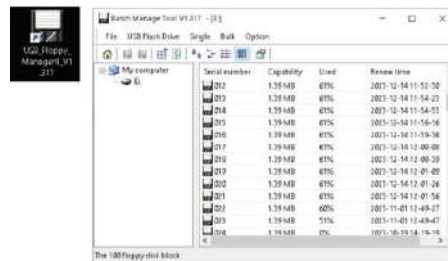


Figura 3.5 Software emulador *Floppy Manager*.

3.1.2 Calibración

Las posiciones del brazo robótico se almacenan en coordenadas rectangulares (valores *x*, *y*, *z*), relacionadas directamente al sistema coordinado de la base del brazo robótico ver Figura 3.6. La calibración implica establecer la posición de referencia de cada uno de los seis ejes. Si el robot o los ejes externos no están calibrados correctamente, resultará en un posicionamiento incorrecto y tendrá un efecto negativo en la agilidad del robot. La calibración se realiza en dos etapas:

- Calibración de ejes: los ejes se colocan en sus posiciones de calibración específicas y se almacenan los valores actuales.
- Actualización de los contadores de revoluciones: se encarga de contar el número de revoluciones del motor de manera correcta para la calibración de los movimientos.

Ambos procedimientos se realizan con el *Teach Pendant* cada que se energiza el controlador, siguiendo la siguiente ruta: *Menú de otras ventanas > Service > View > Calibration > Fine Calibrate o Rev.counter Update*. En el anexo 3 se explica detalladamente el procedimiento de calibración.

Nota: Es conveniente regresar el robot a su posición de calibración antes de apagarlo mediante un programa apropiado (ver anexo 4), ya que al efectuar la calibración manual el sistema de coordenadas es susceptible a cambiar en cada encendido del robot.

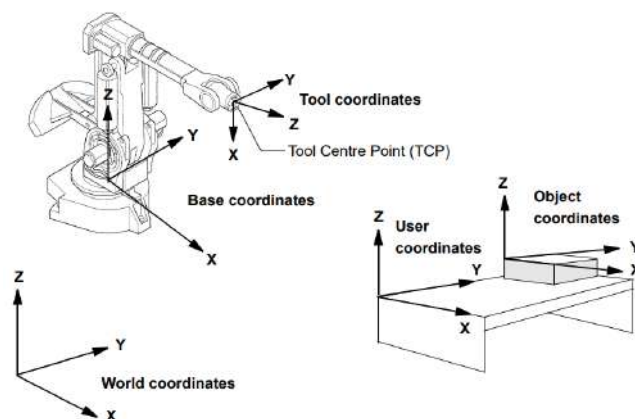


Figura 3.6 Sistemas de coordenadas del manipulador.

Existen cinco tipos de sistemas de coordenadas:

- El sistema de coordenadas “*World*” define una referencia en el piso, se convierte en el punto de partida para los otros sistemas de coordenadas, es muy útil cuando dos robots trabajan juntos.
- El sistema de coordenadas “*Base*” está adjunto a la superficie de montaje base del robot.
- El sistema de coordenadas “*User y Object*” especifican la posición de un accesorio (mesa), o la posición de una pieza de trabajo.
- El sistema de coordenadas “*tool coordinates*” define las referencias de las herramientas acopladas a la brida del robot.

Para definir los sistemas de coordenadas de trabajo es necesario definir tres puntos, uno como origen y dos en la orientación de los ejes (X, Y) en la siguiente ruta: *Menú Programs > File > Open > (Seleccionar el programa) > OK > View > Data > Types Wobjdata > “enter”* se define el tipo de sistema coordenado, en el anexo 5 se explica de manera detallada el proceso para definir los planos de trabajo.

El sistema de coordenadas de la herramienta (*TCP*), especifica el punto central y la orientación de la herramienta respecto a la brida de montaje del brazo robótico. Existen 4 métodos para definir el *TCP* de la herramienta de acuerdo con el manual [16]:

- Método 4 puntos: Se graban 4 diferentes posiciones de la punta de la herramienta en un mismo punto.
- Método 4 puntos sin cambiar orientación: Igual que el anterior, sólo que el eje Z de la herramienta mantiene la dirección de la última articulación.
- Método 5 puntos: Permite definir Z positivo.
- Método 6 puntos: Define Z positivo, también X positivo de la herramienta Figura 3.7.

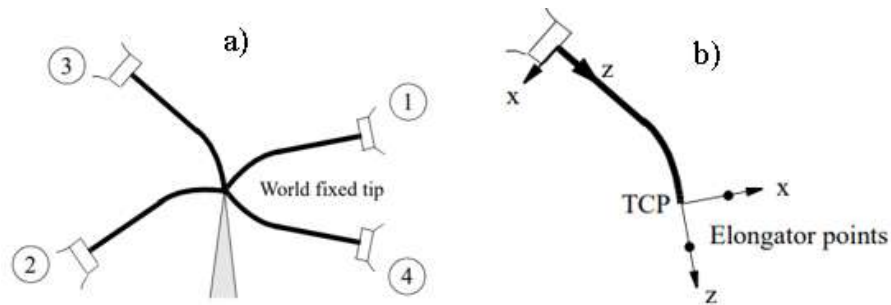


Figura 3.7 Método de definición del TCP, a) 4 puntos, b) 6 puntos.

El TCP se definió mediante el método de los seis puntos debido a que la punta de la herramienta no se encuentra alineada en la dirección de la brida del robot ver Figura 3.8, el proceso de calibración se configura en la siguiente ruta: *Menú Programas > File > Open > (Seleccionar el programa) > OK > View > Data types > tooldata > New > edit points > elegir método de seis puntos* en el anexo 6 se explica el proceso de definición del TCP de la herramienta.



Figura 3.8 Definición de TCP.

3.1.3 Habilitación del módulo de entradas y salidas digitales

El controlador cuenta con cuatro módulos de entradas y salidas digitales *I/O*, son módulos 328 con alimentación de DC 24V. Cada módulo tiene una dirección específica en el *Teach Pendant*, el inferior derecho tiene la dirección número 10, el inferior izquierdo la dirección número 11, el superior derecho 12, superior izquierdo 13, además cuenta con dos puertos de comunicación serial *X1 (SIO1)* para RS232 con adaptador *DB9* macho, y *X2 (SIO 2)* para RS422 con adaptador *DB9* hembra ver Figura 3.9.

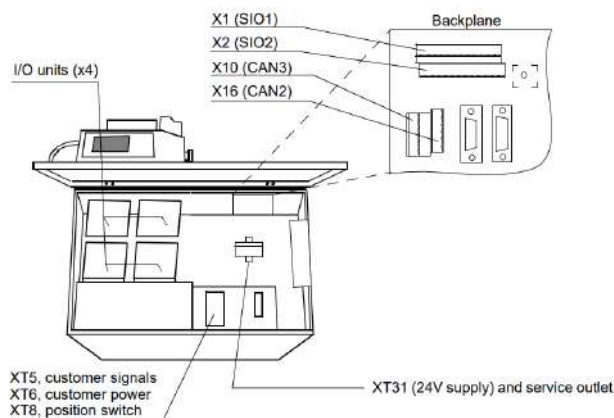


Figura 3.9 Módulos de comunicación en el controlador.

El proceso para habilitar los módulos de las entradas y salidas digitales se describe en el anexo 7, para utilizar la comunicación serial es necesario configurar el puerto serial en la siguiente ruta: *Menú otras ventanas > System Parameters > Comunicación > (seleccionar el canal SIO1-SIO2)* Figura 3.10, donde se configuran los parámetros de envío o recepción de datos como *el baud rate y la paridad*. Al habilitar la comunicación serial y módulos digitales se puede establecer comunicación con dispositivos externos como actuadores y computadoras. En el anexo 8 se describe el proceso para el envío de datos del controlador que posteriormente son recibidos en una *PC* donde son interpretados. El programa utilizado para la recepción es *Hyperterminal*.

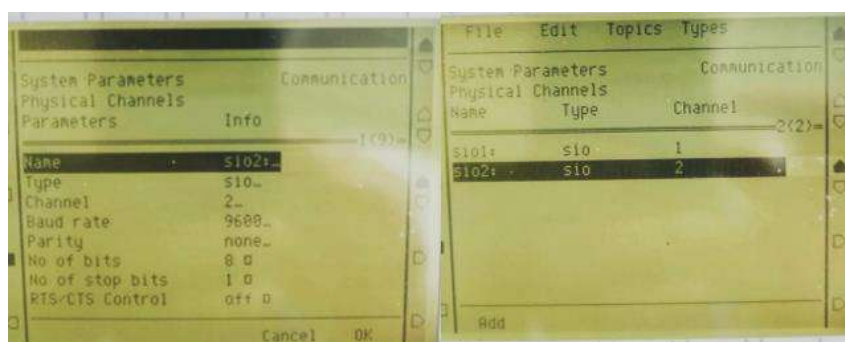


Figura 3.10 Parámetros del puerto serial.

3.2 INTEGRACIÓN DE LA ESTACIÓN

Una célula de mecanizado robótico típica consta de los siguientes componentes integrados en un sistema completo [10]:

- Robot (enlace serial o en paralelo) y controlador.
- Software CAD/CAM compatible con el robot.
- Husillo y herramienta.
- Paquete de cables de utilidad.
- Equipos auxiliares específicos de la aplicación (séptimo eje, mesa de indexación, sensor de fuerza y sistema de visión).

El Laboratorio de Manufactura Avanzada cuenta con los componentes necesarios para la construcción de una estación de mecanizado robótico (Figura 3.11): un brazo robótico *IRB 6400-2.8*, un husillo de la marca *Zhong Hua Jian*, la estructura de sujeción y además de licencias estudiantiles de programas de simulación cinemática y software CAD/CAM.

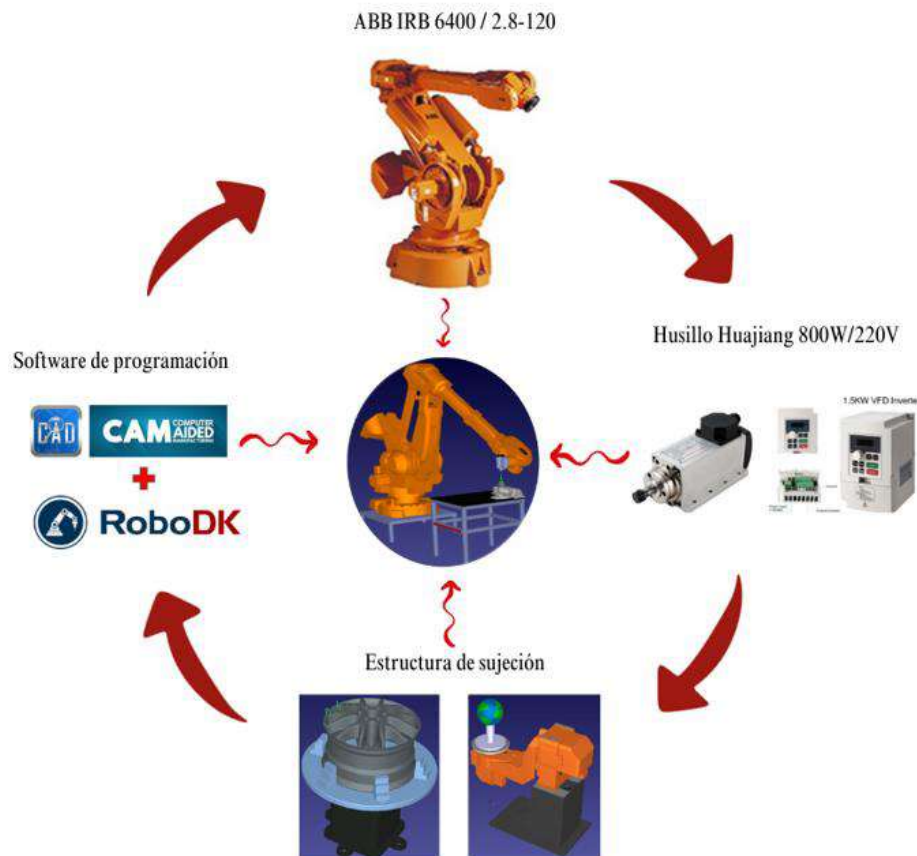


Figura 3.11 Componentes de mecanizado robótico.

El siguiente diagrama describe el proceso para la integración de la estación de mecanizado (Figura 3.12), la integración se realizó simultáneamente al conformar el entorno físico y adaptar el entorno virtual con base a las posiciones de los componentes usando la interfaz del *Teach Pendant*.

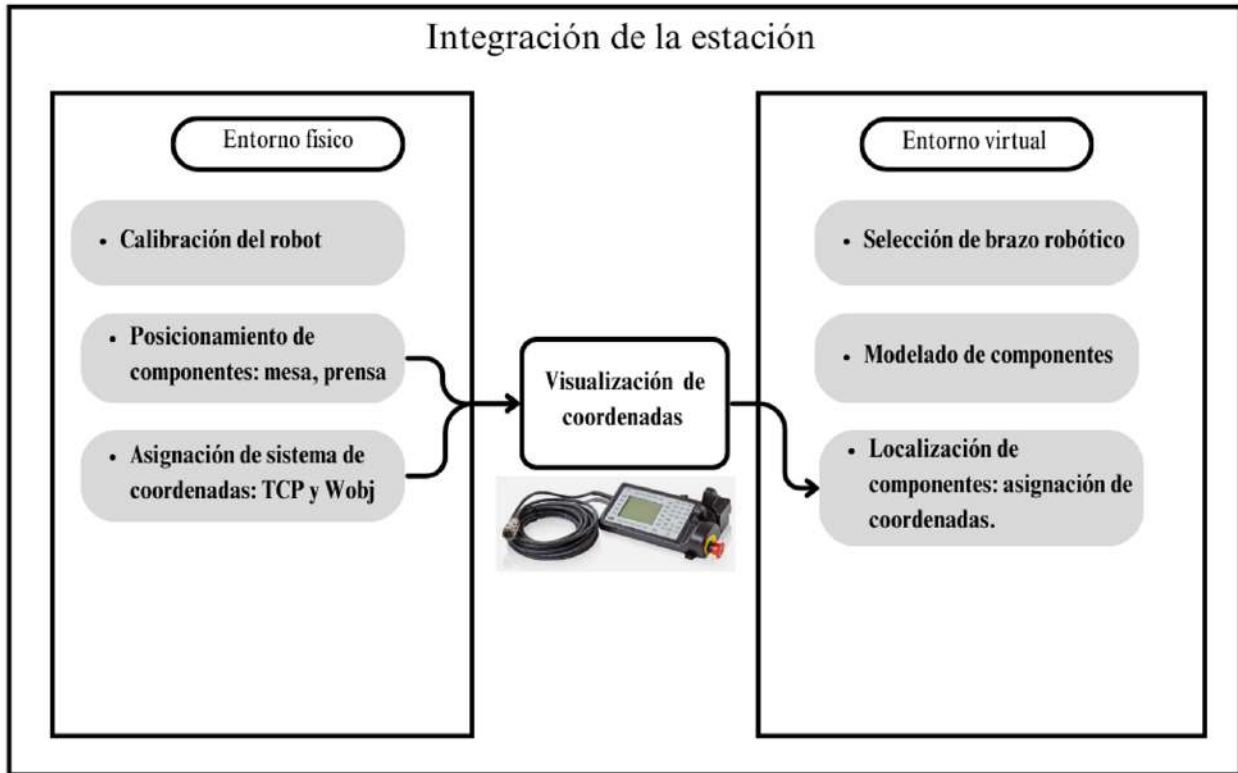
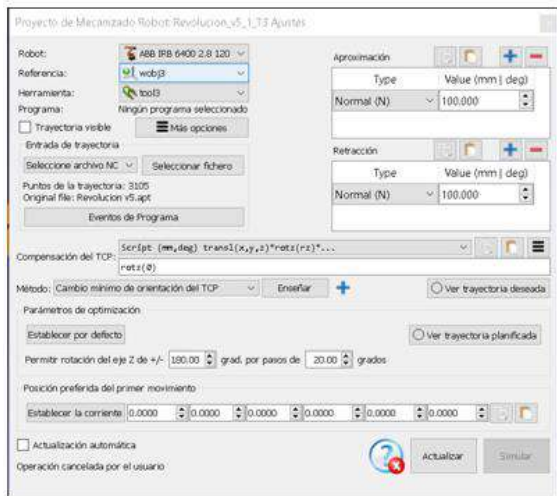


Figura 3.12 Diagrama para la integración de la estación.

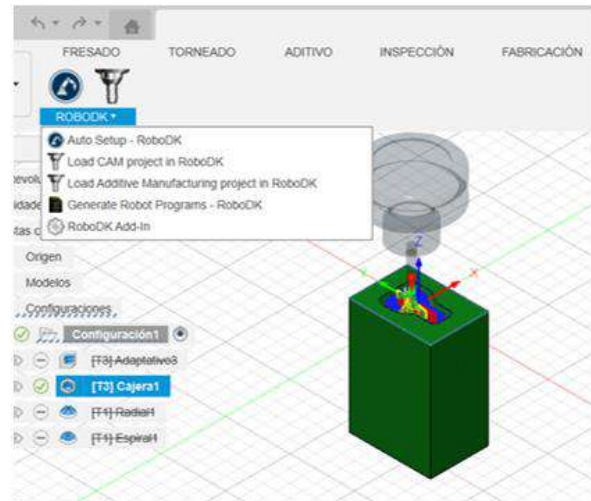
3.2.1 Selección de software

El Departamento de Ingeniería Mecánica cuenta con el apoyo de la compañía *RobotDk* quienes brindan licencias con fines educativos. *RobotDk* es el software predilecto para la simulación de estaciones de robótica industrial además de contar con un módulo específico para la programación de mecanizados.

El software *Fusion 360* de la plataforma estudiantil de *Autodesk* se utilizó para: modelar los componentes de la estación, modelar las piezas de maquinado, realizar la regeneración de modelos mediante ingeniería inversa, además de generar las estrategias de maquinado y crear las trayectorias de las herramientas. *Fusion 360* en versión estudiantil cuenta con la disponibilidad de programación en 4to y 5to eje además de tener un módulo específico para enviar el código de maquinado directamente a *Robotdk* y así posteriormente generar los movimientos del robot, ver Figura 3.13.



a)



b)

Figura 3.13 Módulo de mecanizado en: a) *RobotDk*, b) *Fusion 360*.

3.2.2 Configuración física

La configuración de la estación de mecanizado se realizó con elementos existentes dentro del Laboratorio de Manufactura, la mesa de trabajo tiene dimensiones 80x75x80 cm, fue necesario anclar la mesa al piso mediante taquetes expansivos de 3/8 de pulgada y posteriormente nivelar la estructura. Se adaptó una prensa para la sujeción de piezas, Figura 3.14.



Figura 3.14 Montaje de estructura física.

El brazo robótico cuenta con una brida de montaje en el eje número seis, con un diámetro de 125 mm y seis barrenos roscados M10, Figura 3.15, cuenta con un barreno de localización específico de diámetro 10 mm utilizado como referencia para posicionar herramientas.

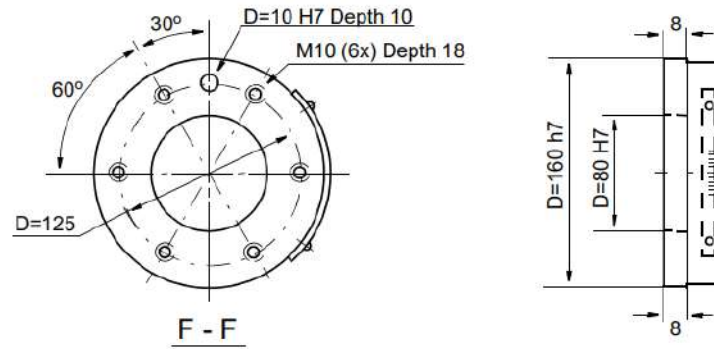


Figura 3.15 Dimensiones de la brida de montaje del robot en mm. [16]

Para acoplar el husillo fresador *Hua jiang* a la brida de montaje se diseñó una placa circular (ver anexo 9) basándose en diseños de proyectos anteriores, con un espesor de 12.5 mm, con diez barrenos avellanados M10 y ocho barrenos roscados M7, para realizar el ensamblaje del husillo fresador, Figura 3.16.



Figura 3.16 Placa de sujeción del husillo.

Se definió la posición de los componentes en el entorno físico, posicionando la mesa y prensa de sujeción bajo los siguientes criterios:

- Área de trabajo del brazo robótico: la configuración de la estructura permite mecanizar cualquier pieza dentro del espacio de la mesa.
- Posición del brazo robótico: se ajustó la prensa en la esquina superior izquierda con el fin de evitar colisiones con la parte central de la mesa a la hora de efectuar el recorrido de trayectorias.
- Rigidez: se buscó proporcionar la mayor rigidez posible a la estructura.

Una vez definido el entorno físico de la estación se realizó el proceso de asignación de sistema de coordenadas el cual consiste en:

1. Definir el *TCP* de la herramienta mediante el método de seis puntos (ver anexo 6).

2. Crear los sistemas de coordenadas “User y Object” (ver anexo 5), ubicados estratégicamente en la esquina inferior izquierda de la mesa y en el centro de la pieza a mecanizar respectivamente
Figura 3.17.

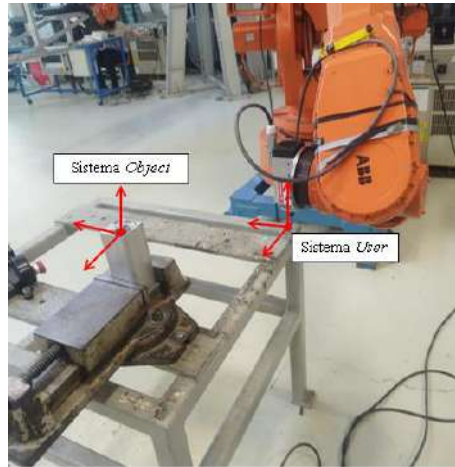


Figura 3.17 Ubicación de los sistemas de coordenadas.

3. Visualización de las coordenadas del TCP, y los planos de trabajo en el *Teach Pendant*: una vez definidos es necesario dirigirse a la siguiente ruta para obtener las coordenadas y así poder ingresar los datos en la simulación virtual.

Menú Programas > File > Open > (Seleccionar el programa) > OK > View > Data types > seleccionar “Tooldata o Wobjdata” > seleccionar el “TCP o Wobj” asignado > ok

Se desplegará una ventana en el *Teach Pendant* con los datos de la herramienta y del plano asignado
Figura 3.18.

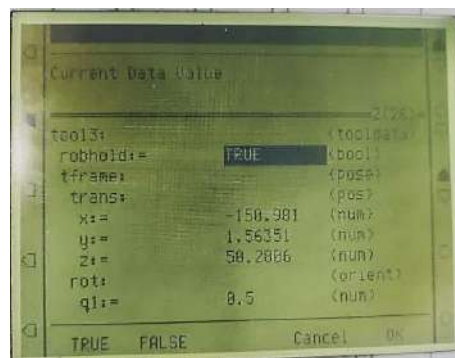


Figura 3.18 Detalles de coordenadas.

Es importante corroborar las posiciones una vez creada la estación virtual, al final se verifica si los planos en la programación son correctos creando programas de aproximación al origen del sistema coordinado.

3.2.3 Configuración virtual

La configuración de la estación virtual se realizó posteriormente a la definición de las posiciones de los componentes en la estación física. *RobotDk* cuenta con su propia librería online que permite descargar brazos robóticos, mesas, sujetadores y estaciones completas, además acepta una gran variedad de formatos de modelos 3D (por ejemplo. *igs/.stl/.step/.obj/.sld etc.*). El modelo del brazo robótico se descargó directamente de la página de *RobotDk* Figura 3.19, un aspecto importante es seleccionar el post-procesador correcto de acuerdo con el modelo de robot.

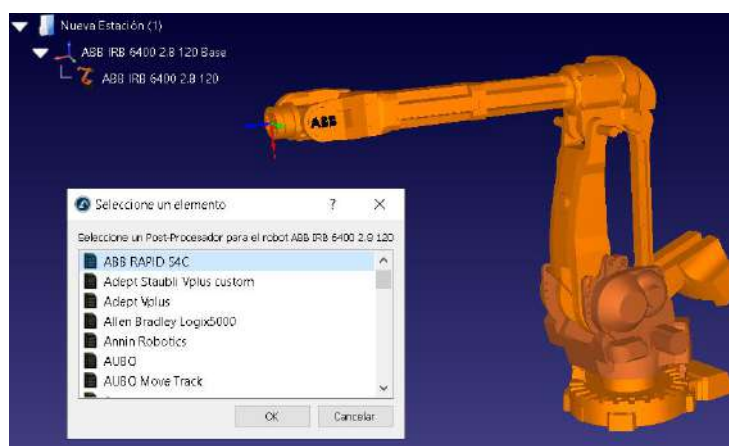


Figura 3.19 Modelo IRB 6400 en RobotDK.

Se generaron los modelos 3d de los dispositivos que conforman la estación ver Figura 3.20. El robot se encuentra sobre un soporte de acero a 31cm del piso, la mesa principal es de acero con una altura de 80cm, la prensa de sujeción fue desacoplada para poder generar el modelo y ensamble de sus dispositivos, el husillo y placa de soporte al igual fue ensamblado.

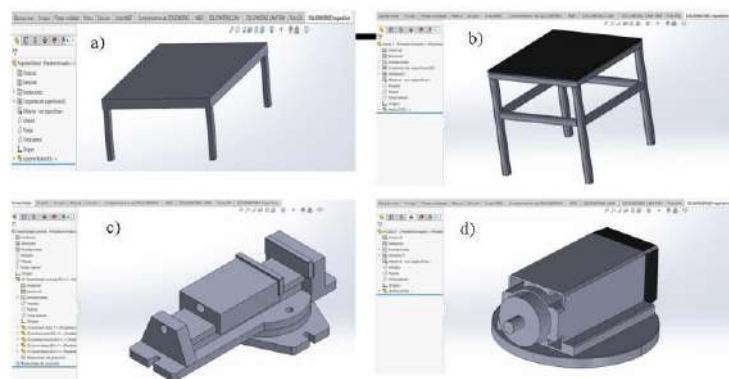


Figura 3.20 Modelos de la estación.

Un aspecto importante para considerar en el modelado de los componentes de la estación es el origen con que se diseñan los modelos en el software CAD, el origen debe ser considerado un *datum* estratégico con el propósito de facilitar su posicionamiento al momento de ser cargado en la simulación virtual.

Existe un apartado específico para modificar la posición de los componentes dentro de la simulación. Presionando F3 sobre el objeto seleccionado se desplegará la ventana de detalles, Figura 3.21, en este apartado se colocarán las coordenadas previamente registradas (Figura 3.18), al realizar el proceso se asegura que la estación virtual sea lo más aproximado a la estación física.

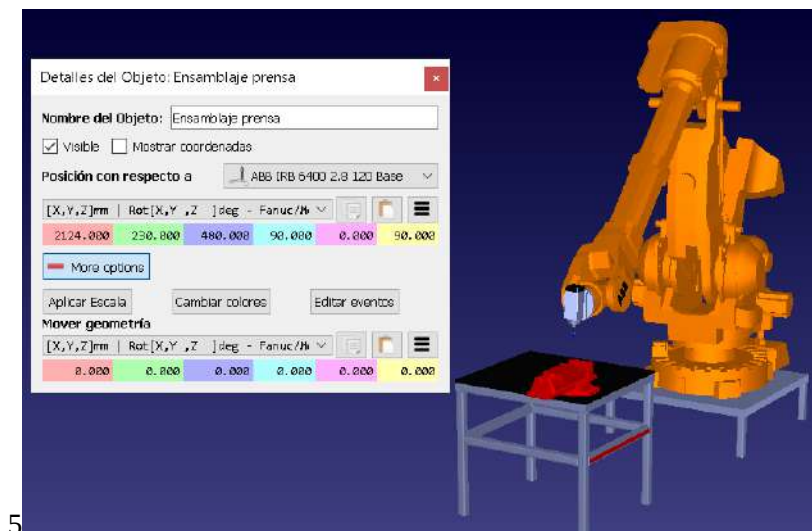


Figura 3.21 Posicionamiento de objetos en *RobotDK*.

Es importante entender que las coordenadas que se obtienen del *Teach Pendant* son respecto a la base del robot que físicamente se encuentra a 31cm del piso, y se deben referenciar de igual forma en el software *RobotDk* de no ser así existirá un desfase en los desplazamientos sobre el eje Z. La estructura del TCP y Wobj (sistemas de trabajo) queda programado como se muestra en la Figura 3.22.

```
! Tool variables: (TCP)
PERS tooldata tool3 :=
[TRUE, [[-152.008, 2.792, 48.474], [0.50880434, -0.50880434, -0.4910378
2, 0.49103782]], [1, [0, 0, 20], [1, 0, 0, 0], 2, 2, 2]];

! Reference variables: (Wobj)
PERS wobjdata Wobj3 := [FALSE, TRUE, "",
[[2183.050, 302.915, 700.700], [0.01252249, -0.00012280, -0.00064736, -
0.99992137]], [[0, 0, 0], [1, 0, 0, 0]]];
```

Figura 3.22 Variables del sistema de trabajo TCP y Wobj.

El último paso es verificar que los planos de trabajo y las posiciones de los componentes coincidan con los programados, la secuencia del proceso de verificación es:

1. Crear un programa de aproximación mediante *RobotDk* que contenga los datos reales del *TCP* y los planos de trabajo.
2. Cargar el programa en el controlador, aparecerán las coordenadas de la herramienta y los planos creados en la ventana de programas Figura 3.18 .
3. Posicionar manualmente el robot sobre el origen de los componentes, sobre el plano de trabajo y el plano del objeto.
4. Verificar visualmente en la ventana movimientos del *Teach Pendant* que el origen del plano programado corresponde al origen físico.

El proceso de verificación de coordenadas se describe en el anexo 10.

CAPÍTULO IV

GENERACIÓN DE LOS PROGRAMAS DE MAQUINADO

La programación del sistema se divide en dos etapas. En la primera se utiliza como plataforma CAD/CAM, el software *Fusion 360*, el cual permite crear los modelos 3D y generar las trayectorias de mecanizado; en la segunda etapa se utiliza el software *Robotdk* para transformar las trayectorias definidas en *código G* a trayectorias en lenguaje *Rapid*. En la Figura 4.1 se muestra un esquema que ilustra las tareas de programación dentro de cada entorno virtual.

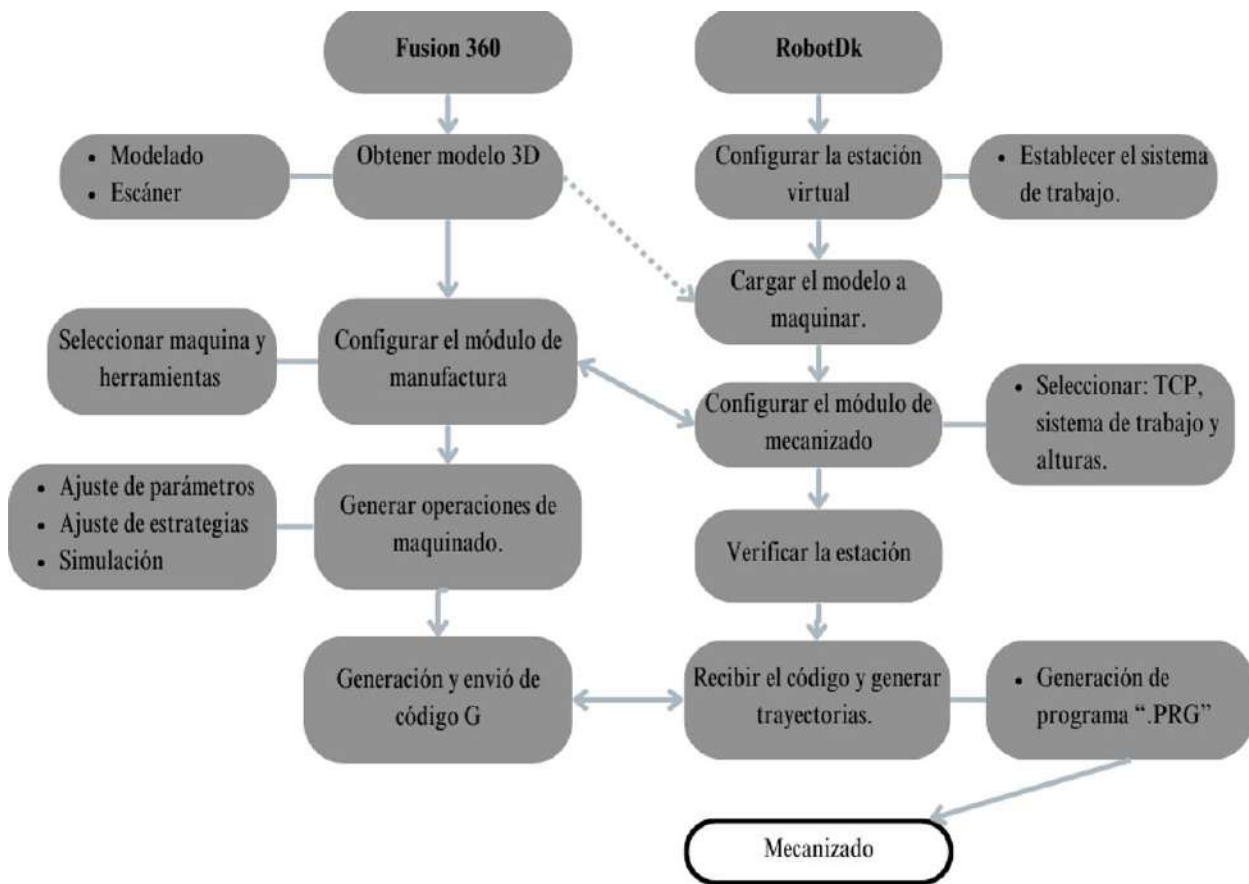


Figura 4.1 Diagrama de operaciones en la programación.

4.1 PROGRAMACIÓN EN *FUSION 360*

La configuración del módulo de fabricación se realizó con la secuencia mostrada en la Figura 4.2, donde a partir de las dimensiones de la pieza final se selecciona un bloque de material y se programan las

estrategias de maquinado adecuadas de acuerdo con las herramientas, el material y la complejidad de las trayectorias de mecanizado.

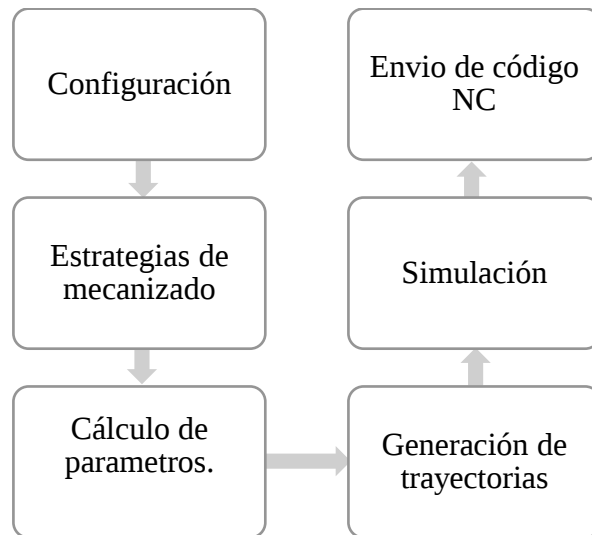


Figura 4.2 Secuencia de programación en *Fusion 360*.

En el entorno de *Fusion 360* existen dos formas de obtener el modelo 3D:

- Modelado: Implica dimensionar y crear el sólido con las herramientas del CAD.
- Escaneo: Implica escanear la pieza mediante una nube de puntos para generar una malla y crear un sólido. El proceso de obtención de modelos por medio de escáner es también llamado ingeniería inversa y se explicará a detalle en el apartado 4.4.

Fusion 360 permite cargar directamente el modelo 3D a la estación virtual en *RobotDk*, así obtener una simulación acorde con la estación física Figura 4.3.

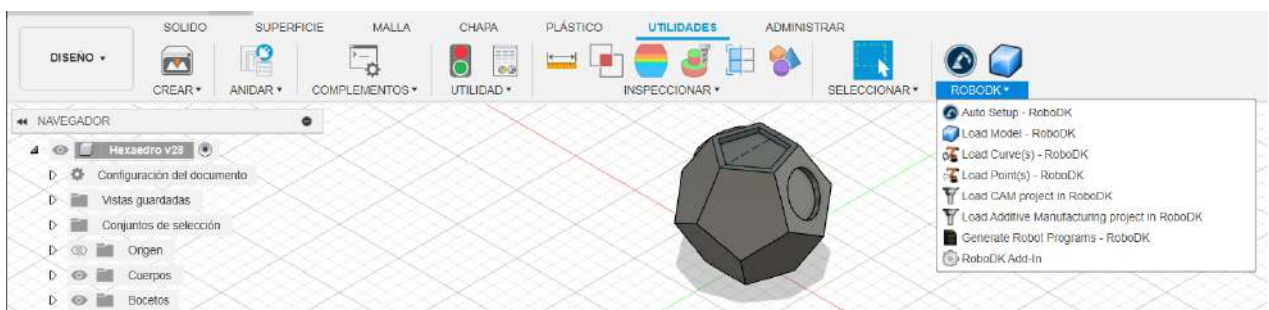


Figura 4.3 Apartado en *Fusion360* para enviar el modelo 3D a *RobotDk*.

Después de contar con el modelo 3D es necesario configurar el módulo de fabricación de *Fusion 360* Figura 4.4, el cual permite configurar la dirección del sistema de coordenadas, el tipo de máquina, las herramientas y las dimensiones del material.

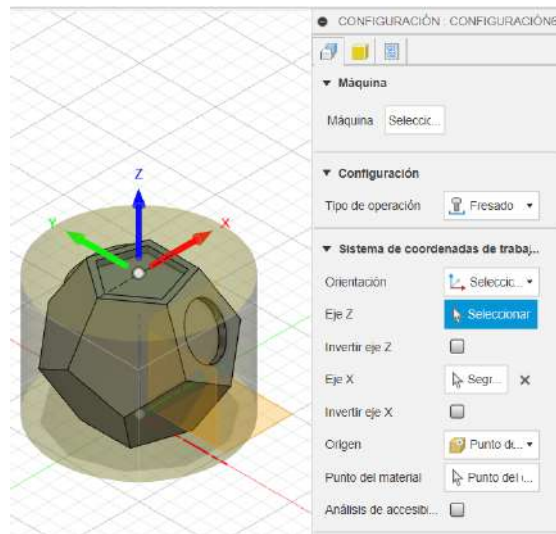


Figura 4.4 Configuración del módulo de fabricación.

Las estrategias de mecanizado se seleccionan en función de una combinación de características, por ejemplo:

- Material: según el tipo de material requiere enfoques de mecanizado específicos debido a sus propiedades físicas (dureza, maquinabilidad).
- Geometría de la pieza: piezas con formas complicadas pueden requerir estrategias de mecanizado en 5 ejes para acceder a todas las áreas de la pieza.
- Tolerancias y acabado requeridos: algunas estrategias pueden ser más adecuadas para lograr ciertas tolerancias o acabados superficiales.
- Herramientas disponibles: las herramientas de corte disponibles, como fresas, brocas, insertos de torneado, etc., influirán en las estrategias de mecanizado que se pueden utilizar. La selección de herramientas adecuadas es fundamental para optimizar el proceso de mecanizado.
- Máquina herramienta y capacidades de sujeción: las capacidades de la máquina herramienta, incluida la velocidad de husillo, la potencia de corte, los ejes disponibles (3, 4 o 5 ejes), y los sistemas de sujeción.
- Costo y tiempo: factores económicos como el costo de las herramientas, el tiempo de mecanizado y la eficiencia del proceso también influirán en la elección de las estrategias.

Fusion 360 en su versión estudiantil permite acceder a toda su carpeta de estrategias de mecanizado en 2D y 3D, además de programar las herramientas multi-ejes y ejes indexados en operaciones de desbaste y acabado Figura 4.5.

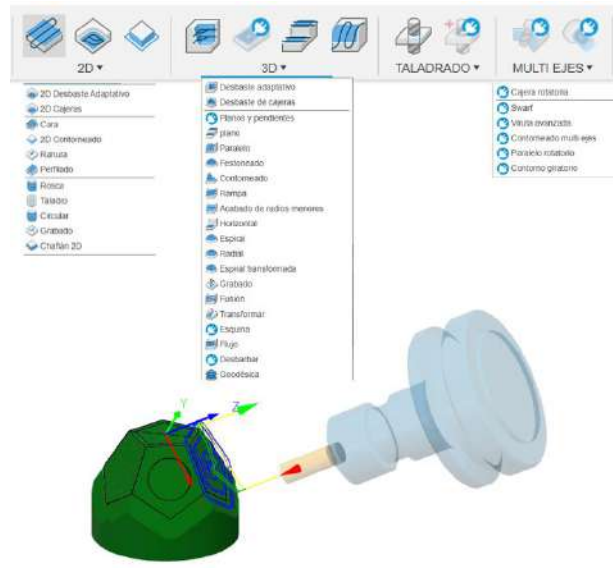


Figura 4.5 Estrategias de mecanizado.

Después de seleccionar la estrategia de mecanizado apropiada es necesario ajustar los parámetros de corte en función de las siguientes características:

- Material de la pieza de trabajo: los plásticos y metales blandos como el aluminio pueden tolerar velocidades de corte más altas que los metales más duros como el acero.
- Tipo de herramienta: herramientas como fresas de alta velocidad de acero (HSS), fresas de carburo de tungsteno tienen diferentes capacidades y limitaciones.
- Rigidez de la máquina y sujeción de la pieza de trabajo: una sujeción deficiente o una máquina con baja rigidez pueden limitar los parámetros de corte que se pueden utilizar sin comprometer la calidad del mecanizado.
- Tolerancias y acabado superficial requeridos: tolerancias más estrictas pueden requerir velocidades de corte más bajas y etapas más finas para garantizar la precisión, un acabado superficial de calidad requiere velocidades de corte y avance específicas.
- Potencia del husillo y velocidad máxima de la máquina: la velocidad del husillo se ve limitada a 24,000 rpm.

Los procesos de mecanizado se efectuaron en material *delrin*, el otro material utilizado fue aluminio, material seleccionado por su baja dureza y buena maquinabilidad.

Las velocidades de corte y avance se vieron limitadas por el tamaño de la herramienta, ya que el modelo del husillo está restringido a utilizar cortadores que se acoplen al portaherramientas con un *colet ER11* permitiendo sujetar máximo un diámetro de siete milímetros de zanco.

La velocidad de corte y avance se calcularon conforme a los parámetros del anexo 1, el parámetro de avance por carga o avance por penetración se mantuvo a 1/3 de la velocidad de avance, ver Figura 4.6.

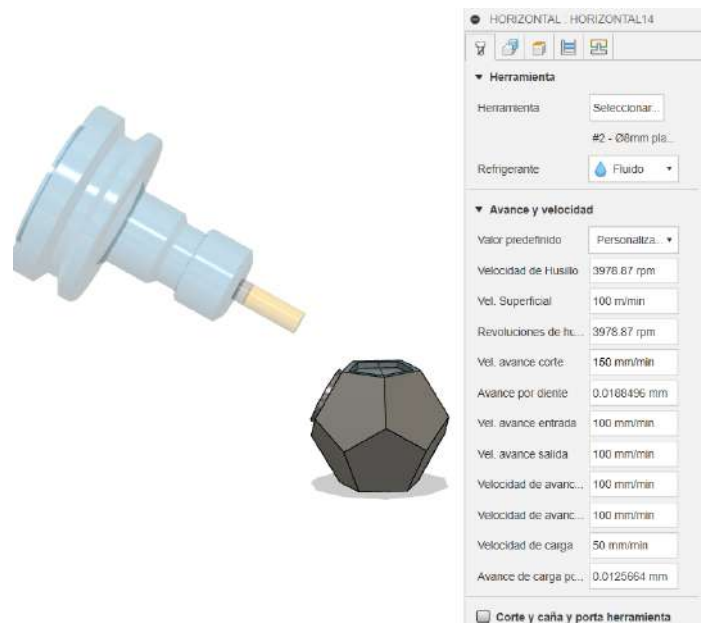


Figura 4.6 Parámetros de corte en *Fusion 360*.

Fusion 360 permite simular las trayectorias generadas y hacer una verificación de colisiones, revisando al mismo tiempo la remoción de material Figura 4.7.

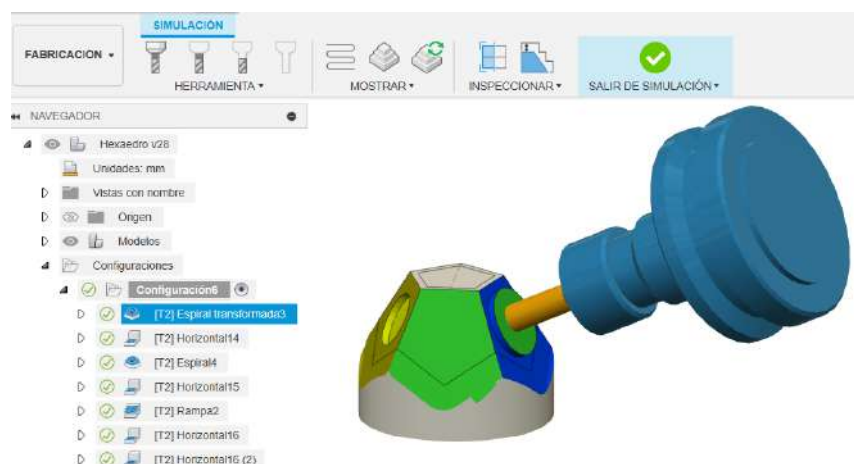


Figura 4.7 Simulación en *Fusion 360*.

Al concluir la programación de estrategias y la simulación de trayectorias, se envía el código NC (Figura 4.8) al entorno virtual de *RobotDk* donde previamente se genera una estación con las coordenadas del sistema “*Object*” donde se posicionará la pieza de material a mecanizar.

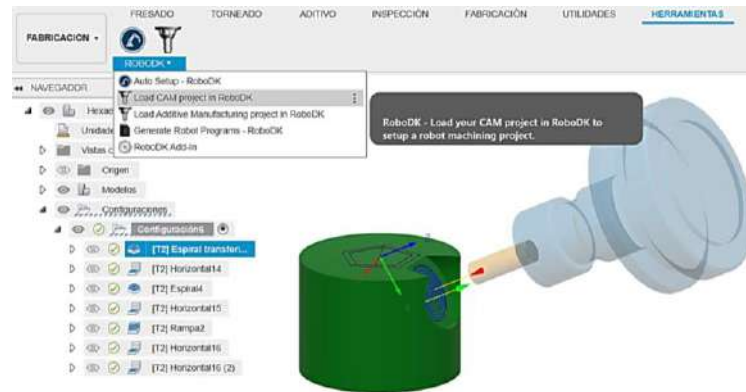


Figura 4.8 Envío de código NC.

4.2 PROGRAMACIÓN EN *ROBOTDK*

En el siguiente apartado se describe la configuración de la estación virtual donde se definieron las posiciones: de los componentes, el *TCP* de la herramienta, los planos de trabajo y el modelo de pieza a mecanizar en *RobotDk*. La secuencia de programación en el software *RobotDk* se muestra en la Figura 4.9.

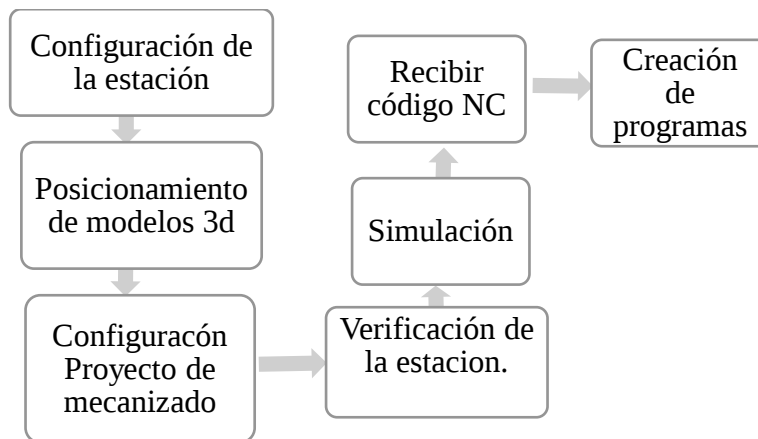


Figura 4.9 Secuencia de programación en *RobotDk*.

El *Teach Pendant* es el enlace para conformar las posiciones físicas y obtener los datos de las coordenadas para configurar la estación virtual, un factor importante es la habilidad para manipular el robot ya que la precisión dimensional se verá comprometida si la estación virtual difiere de la estación física.

En el apartado “módulo de mecanizado” Figura 4.10, se definen los parámetros de maquinado:

- Sistema “*Object*”: aquí se configuran las coordenadas de la pieza reales mostradas en el *Teach Pendant*, el punto de origen y orientación de los ejes coordenados se programa acorde a la posición del sistema seleccionado en *Fusion 360*.
- Alturas y velocidad de aproximación: se definen la altura y velocidad de seguridad de aproximación programada entre 20-30 mm/s, para movimientos que involucren cambio de orientación de la herramienta precisos se programan a 5 mm/s.
- Ajuste de posiciones del robot: permite cambiar las posiciones de los ejes al realizar el programa de mecanizado y buscar la mejor orientación acorde a la estación física.
- Compensación del *TCP*: aquí podemos variar el ángulo de ataque de las herramientas y ajustar el ángulo de corte para que la herramienta se mantenga normal a la superficie.
- Simulación: permite simular las trayectorias programadas y verificar visualmente colisiones, el tiempo ciclo entre la simulación en *Fusion 360* y *RobotDk* varía en un rango entre 10-40 seg.

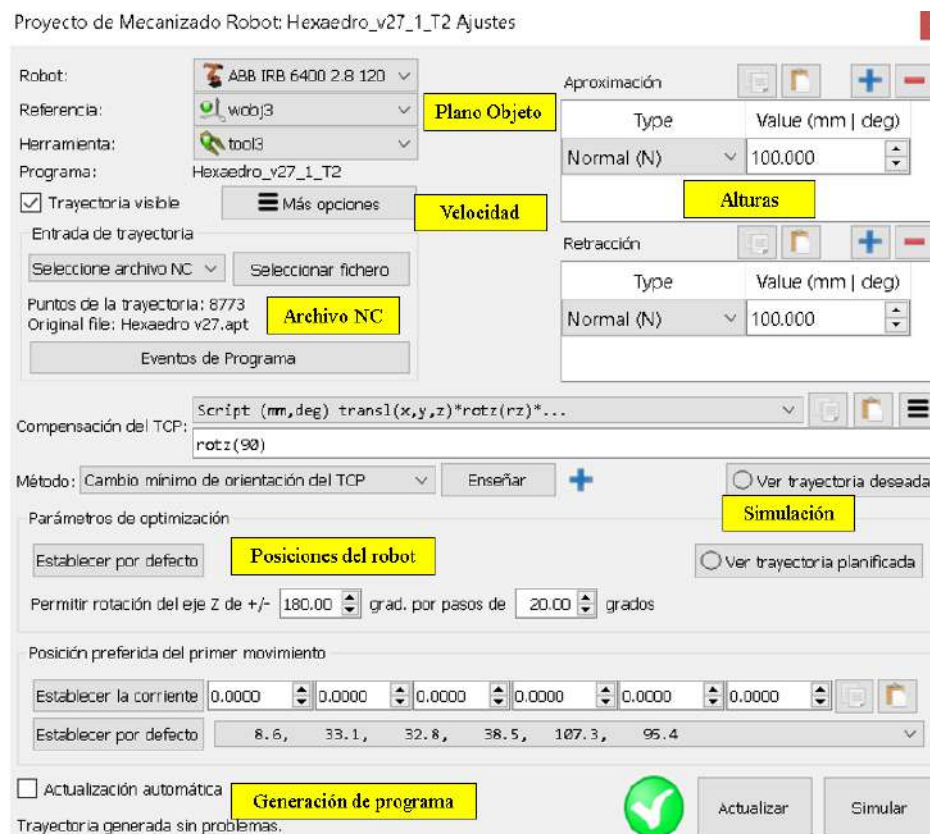


Figura 4.10 Módulo mecanizado en *RobotDk*.

En el árbol de operaciones de *RobotDk* se genera el programa de maquinado Figura 4.11, es importante realizar el proceso de verificación del sistema “*Object*” (ver anexo 10) antes de maquinar, para verificar que coincida el sistema virtual con el sistema real. La capacidad máxima de líneas de código de programación que soporta el controlador es de 7000 por lo que es necesario generar un programa por operación. La estructura de código en lenguaje *Rapid* de la estación automatizada se muestra en el anexo 13.

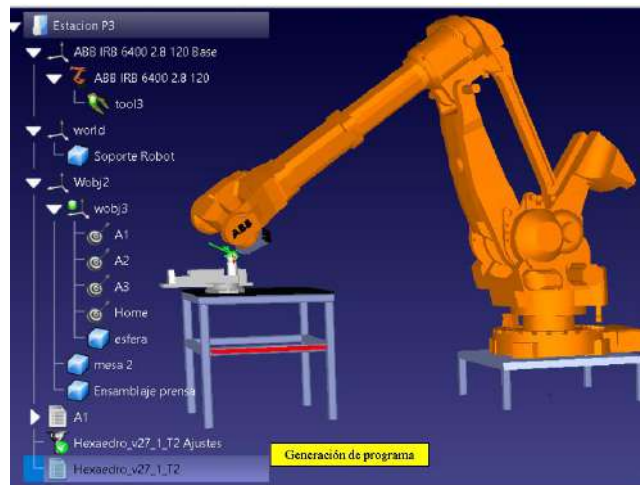


Figura 4.11 Árbol de operaciones RobotDk.

4.3 CONFIGURACIÓN DEL VARIADOR DE FRECUENCIA

El arranque y cableado del husillo se realizó conforme al manual del VDF [18], suministrando 220V de corriente monofásica, la conexión del husillo se realizó como se muestra en la Figura 4.12, de esta manera el variador entrega una corriente trifásica.

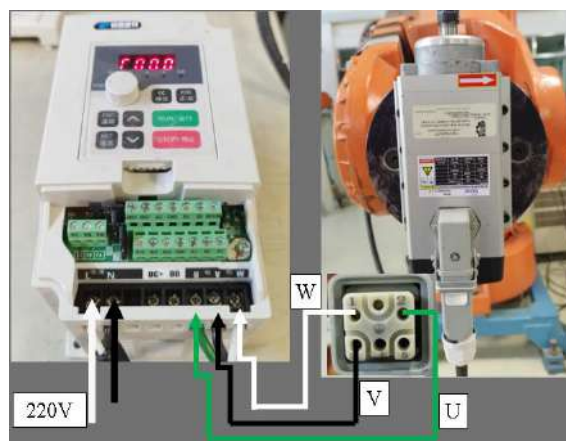


Figura 4.12 Cableado del husillo.

La primera solución para realizar el control del variador de frecuencia fue mediante una botonera industrial que cuenta con un potenciómetro y un par de botones (*B1* y *B2*), con dos posiciones para el encendido y cambio de giro, ambos botones son normalmente abiertos (*NO*). La configuración dentro del variador de frecuencia se ajustó según la lógica *NPN* para las entradas digitales y el potenciómetro, el diagrama de cableado se muestra en la Figura 4.13 .

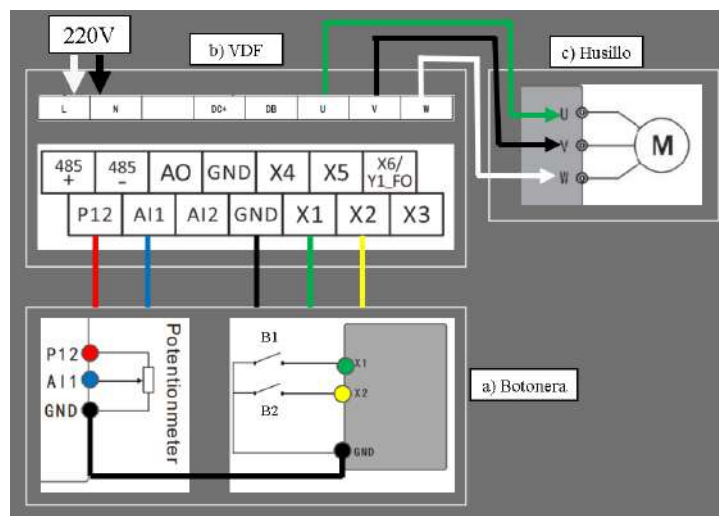


Figura 4.13 Cableado de botonera industrial.

La configuración de los parámetros en el *VDF*, para el control mediante potenciómetro y botonera se describe en la Tabla 4.1.

Tabla 4.1 Ajuste de parámetros, para el control por botonera.

Parámetro	Ajuste	Descripción
F000	0	Permite el cambio de parámetros; se regresa a #1 al finalizar.
F001	1	Tipo de control, habilita las terminales digitales X1, X2.
F002	1	Control de frecuencia, habilita el potenciómetro externo (AI1).
F003, F004	400	Es el valor de la frecuencia principal.
F005	400	Es el valor máximo de frecuencia depende de la velocidad máxima del husillo (24,000 rpm).
F044	1	Habilita la entrada X1 como función de encendido.
F045	5	Habilita la entrada X2 como función de sentido de giro.
F067	0	Establece la lógica interna de los transistores (0-NPN, 1-PNP)

4.3.1 Control del VDF por medio de protocolo *Modbus RTU*

Se desarrollaron distintas soluciones para el funcionamiento de la estación de manera automática, similar a una estación CNC donde se accionan los motores directamente en el código del programa principal de la máquina.

El protocolo *Modbus RTU* por medio del estándar *RS-485* permite establecer comunicación entre un dispositivo externo (maestro) y el variador de frecuencia (esclavo), para controlar remotamente la velocidad. Las dos alternativas consideradas para el control de velocidad por *RS-485* fueron las siguientes:

1. Control por medio del software *Commix*.
2. Control por medio de una tarjeta *Arduino Mega*.

El Software *Commix* envía la trama de datos *Modbus RTU* desde una *PC* hacia el variador, por medio del puerto *COM* de la *PC*. Es necesario establecer los parámetros de envío de datos: *baudrate*, *paridad* y *data bits* en ambos dispositivos (maestro-esclavo) como se muestra en la Figura 4.14.

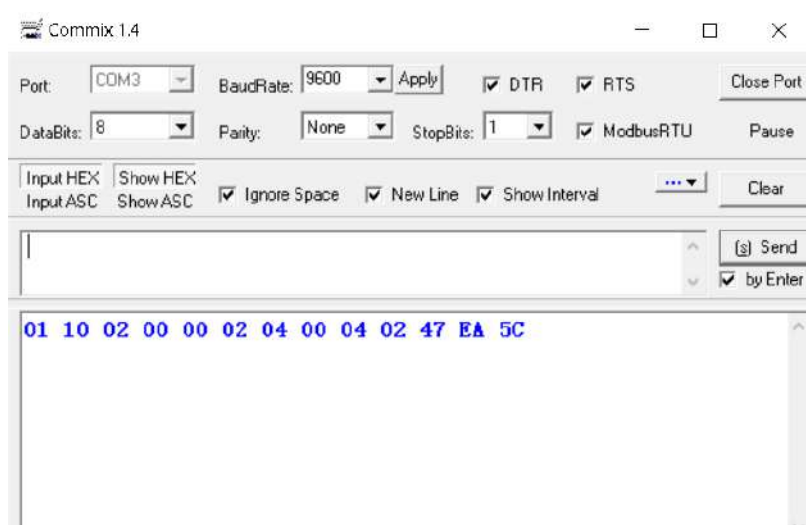


Figura 4.14 Software *Commix*.

Para la transferencia de datos hacia el *VDF* por medio del software *Commix* es necesario contar con una unidad *USB a RS-485*, además descargar el controlador *CH340* en la *PC* para identificar el puerto *COM*, ver Figura 4.15.



Figura 4.15 Proceso de envío de datos *USB-RS485*.

Es importante comprender el arreglo de la trama de datos en el protocolo *Modbus*, la trama de datos se divide en tres secciones: campo de dirección, código de función, datos y el CRC (chequeo de redundancia cíclica), Figura 4.16.

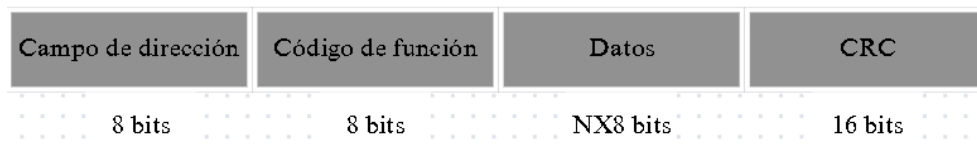


Figura 4.16 Trama de datos *Modbus*.

- Campo de dirección: se establece la dirección del esclavo, en caso de tener varios dispositivos asociados.
- Código de función: se establecen las direcciones de las funciones o registros, a escribir o leer.
- Datos: se envía o recibe el valor de interés, en este caso la frecuencia de giro de manera Hexadecimal.
- *CRC Modbus 16*: se establece a través de un generador online, metiendo la trama de datos de cada instrucción.

El *VDF* cuenta con sus propias funciones del protocolo *Modbus*, dentro del manual existe un apartado específico donde se describe la trama de datos, las funciones se muestran en el anexo 11.

Para establecer la frecuencia es necesario enviar el valor en lenguaje hexadecimal escalado por diez para que el variador pueda interpretarlo. La siguiente tabla muestra la trama de datos para diversas funciones realizadas por el *VDF* mediante el software *Commix*.

Tabla 4.2 Trama de datos para realizar operaciones en *VDF*.

Función	Dirección	Registro	Datos	CRC
Establece 300 HZ	01	06 02 01	0B B8	DEF0
Establece 150 HZ	01	06 02 01	05DC	DB7B
Arranque	01	05 00 48	FF00	-
Paro	01	05 00 4B	FF00	-

La segunda alternativa utilizada tiene como propósito controlar automáticamente el *VDF* por medio de una tarjeta *Arduino Mega* recibiendo el valor programado de la velocidad directamente del controlador del robot, el proceso de envío de datos se muestra la Figura 4.17.

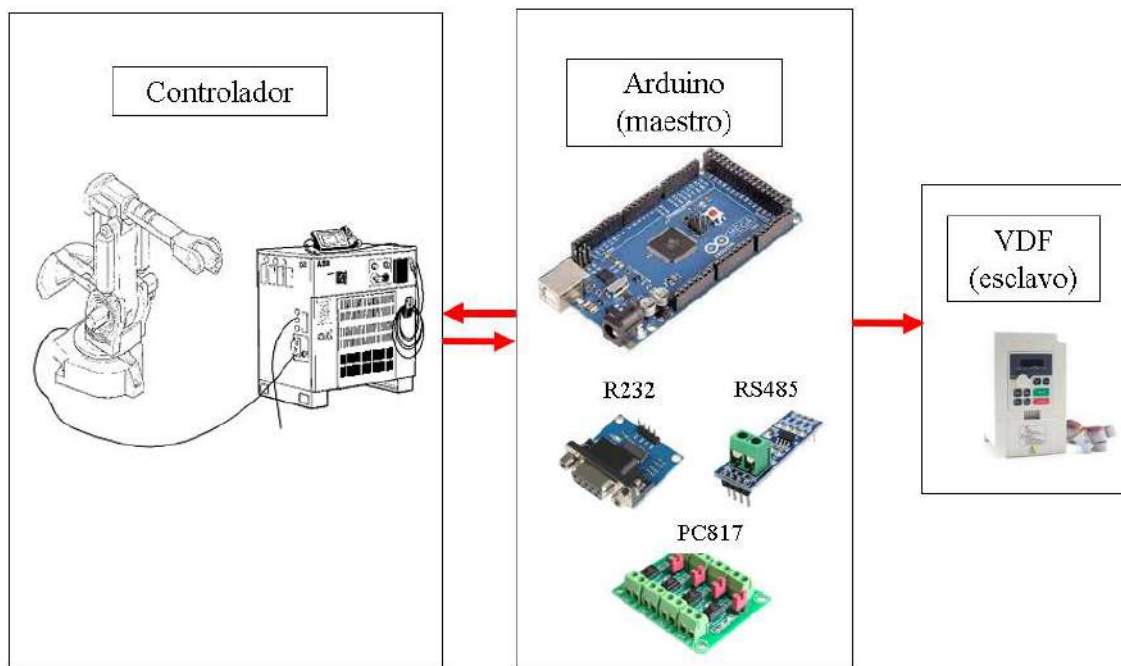


Figura 4.17 Proceso de envío de datos por medio de *Arduino Mega*.

El proceso consiste en recibir una cadena de caracteres indicando el valor de las revoluciones por minuto, simultáneamente transformar y enviar la trama de datos al *VDF*. Es necesario utilizar la librería *Modbus RTU* y una placa de *Arduino Mega* para usar dos puertos seriales simultáneamente, uno para la comunicación con el controlador y otro para la comunicación con el *VDF*. La secuencia del programa dentro del apartado “*void loop*” es la siguiente:

1. Habilitar la comunicación serial *RS232* en la placa *Arduino*.
2. Recibir el valor de la velocidad por serial 1 (*RS232*) a manera de “*string*” (cadena de caracteres).

3. Procesar la velocidad y guardarla en un registro (*Modbus*).
4. Procesar la señal de los pines digitales configurados para el arranque o paro, al igual se guarda en un registro.
5. Enviar la trama de datos usando *RS485* y la librería *Modbus* de *Arduino* hacia el *VDF*.

El programa en *Arduino* y las conexiones entre los módulos se describen en el anexo 12. La Tabla 4.3 muestra los parámetros del *VDF* que se modificaron en el variador de frecuencia para habilitar el módulo *RS485* para controlar la velocidad, el arranque y el paro del husillo.

Tabla 4.3 Ajuste de parámetros del *VDF*, para el control por módulo *RS485*.

Parámetro	Ajuste	Descripción
F000	0	Permite el cambio de parámetros; se regresa a #1 al finalizar.
F001	2	Habilita el módulo <i>RS485</i> .
F002	2	Permite el control de frecuencia por el módulo <i>RS485</i> .
F003, F004	400	Es la frecuencia principal.
F005	400	El valor máximo de frecuencia depende de la velocidad máxima del husillo (24,000 rpm).
F163	1	Define la dirección del <i>VDF</i> como esclavo 1.
F164	1	Velocidad de transmisión de 9600 bits/s
F165	3	Define el modo de comunicación 8N1 para <i>RTU</i> .
F169	1	Define el protocolo de comunicación <i>Modbus</i> .

4.4 INGENIERÍA INVERSA

El proceso requiere usar el software *EinScan Pro de Shinning 3D* para realizar la interfaz con el escáner y efectuar la generación de nube de puntos y alineación, por otro lado, *Fusion 360* permite realizar la transformación y regeneración de los archivos en modelos sólidos. La precisión de la ingeniería inversa radica en la calidad de malla y la resolución que genera el escáner para la creación de la nube de puntos.

La metodología para realizar ingeniería inversa de los modelos maquinados se describe mediante el diagrama de la Figura 4.18.

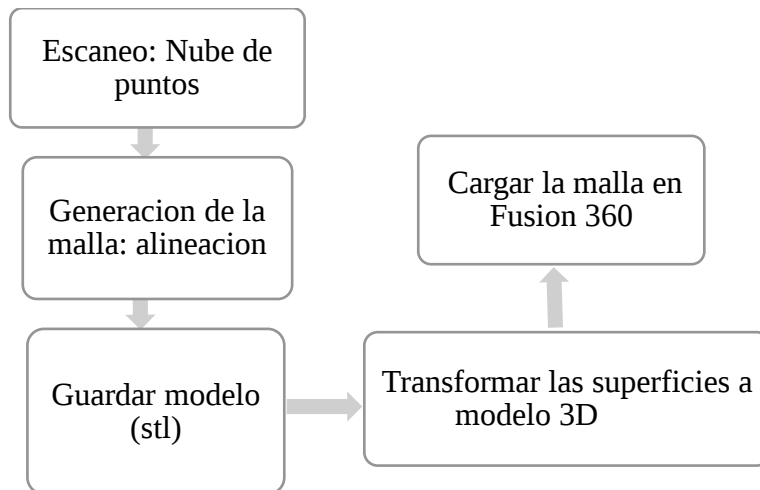


Figura 4.18 Metodología para ingeniería inversa.

El software *EinScan Pro*, guía al usuario durante el proceso que se realiza de manera secuencial partiendo desde la calibración previa del equipo de acuerdo con el manual de uso [24].

1. El primer paso es generar la nube de puntos Figura 4.19, enfocando la luz estructurada alrededor de toda la pieza para detectar superficies y capturar los puntos que correspondan al modelo.

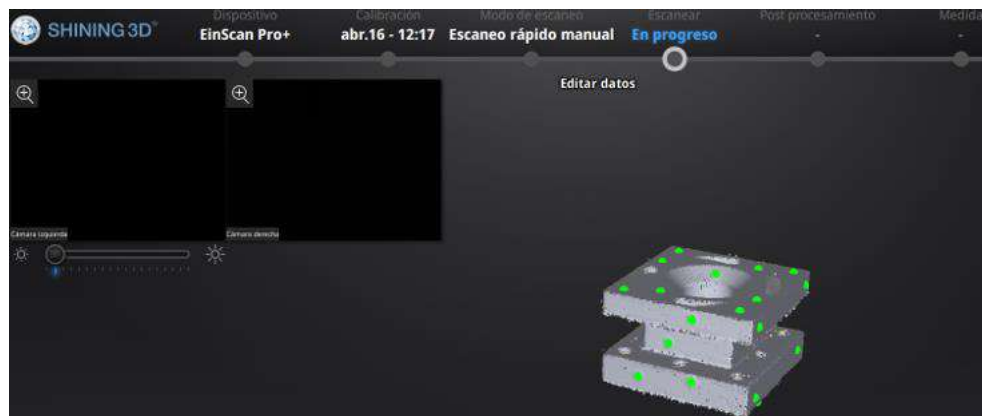


Figura 4.19 Generación de nube de puntos.

2. *EinScan Pro* permite crear varios proyectos en simultaneo y generar diferentes perspectivas del mismo modelo con la finalidad de hacer una construcción precisa y evitar huecos en la malla de puntos Figura 4.20. Es necesario alinear los proyectos mediante la selección de 3 puntos en común para que el software alinee ambas nubes de puntos y las unifique en un solo grupo.

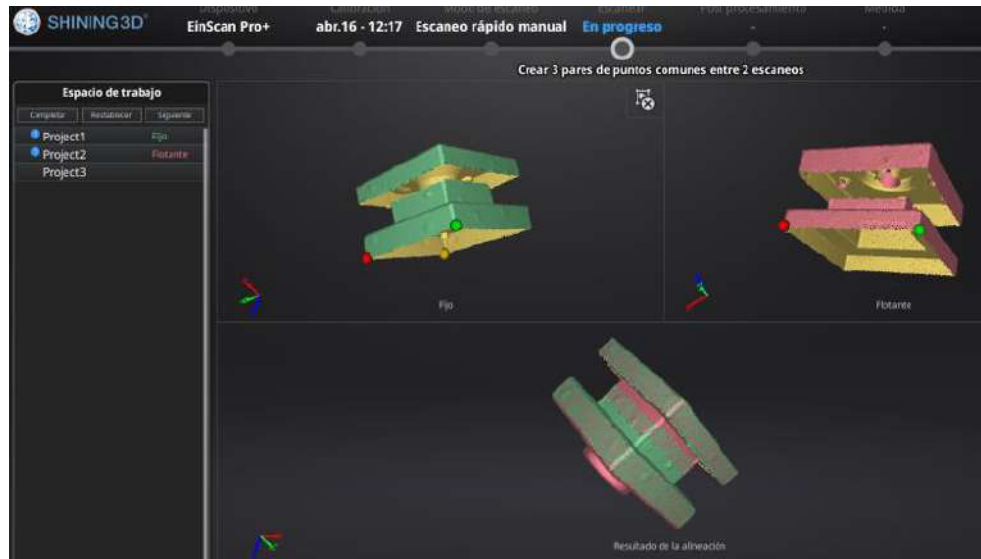


Figura 4.20 Alineación de la nube de puntos.

3. El último paso es generar la malla del modelo y guardarla en un archivo compatible con *Fusion 360*, la extensión de los archivos es “. *stl*” Figura 4.21.

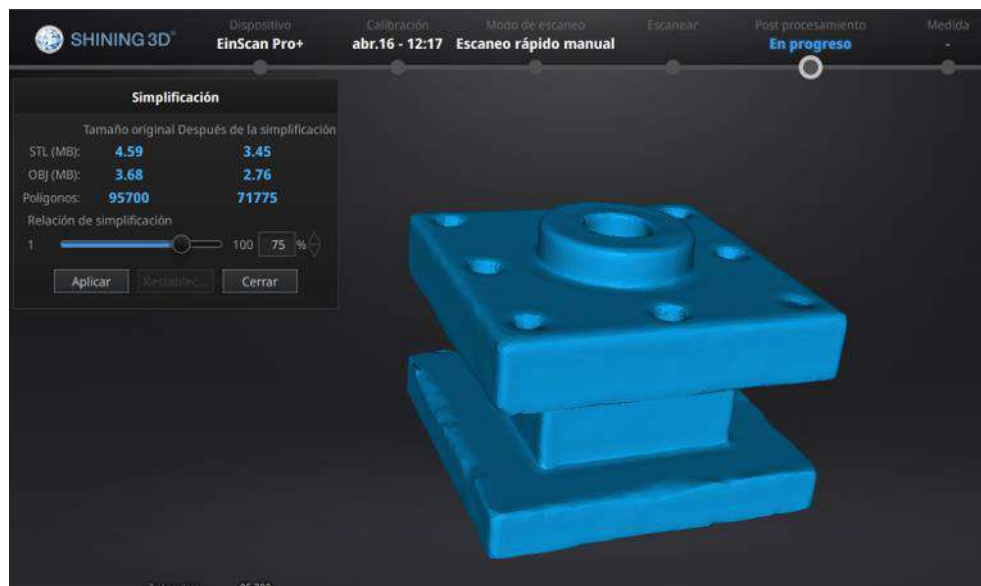


Figura 4.21 Generación de malla en archivo *stl*.

La transformación de la malla de puntos a modelo 3D se genera en *Fusion 360* de manera automática mediante los siguientes pasos:

1. Inicialmente abrir el archivo “. *stl*” en el módulo de malla, Figura 4.22, además de preparar la malla para operaciones posteriores con la herramienta “*reparar*” la cual cierra los espacios que presente la nube de puntos.

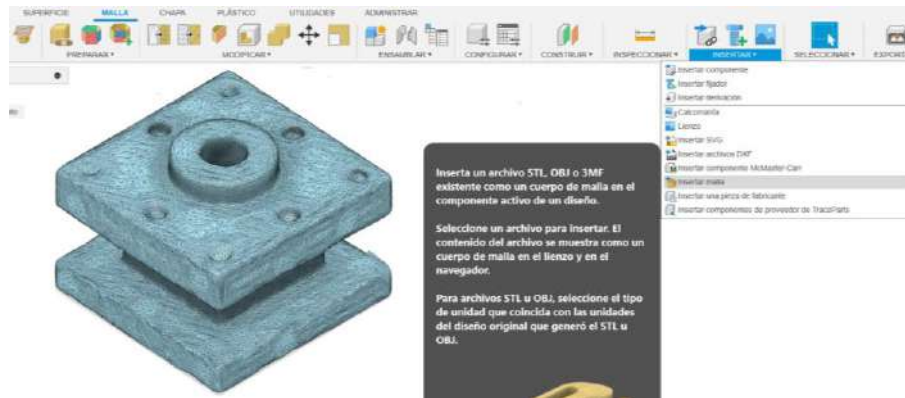


Figura 4.22 Visualización de mallado del modelo en Fusion 360.

- Es necesario reducir el número de elementos entre 10,000-20,000 Figura 4.23 para que el software pueda realizar el procesamiento, dirigiéndose a la sección “modificar malla” con la herramienta de “reducir”.



Figura 4.23 Reducción del número de superficies de mallado.

- Automáticamente *Fusion 360* convierte las superficies de mallado a superficies del modelo sólido con la herramienta de “convertir malla” (*mesh to be reep en inglés*), Figura 4.24. Ahora el modelo se visualiza como solido en el árbol de operaciones y se pueden generar las operaciones de maquinado en el módulo de fabricación.

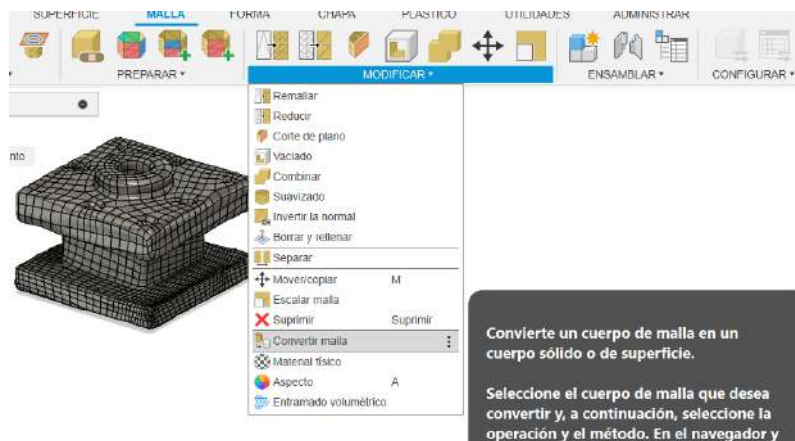


Figura 4.24 Transformación de malla a sólido.

CAPÍTULO V

ANÁLISIS DE RESULTADOS

5.1 PRUEBAS DE MAQUINADO EN MATERIAL DELRIN Y ALUMINIO

Los primeros maquinados se realizaron en material *delrin* Figura 5.1, acetal antiestático usado en la fabricación de componentes electrónicos con alta maquinabilidad. Los parámetros seleccionados que se emplearon con un cortador plano de 6 mm fueron: velocidad del husillo de 3500 rpm, la velocidad de avance a 100 mm/min y velocidad de carga a 50 mm/min. El desbaste se realizó con un cortador plano de Ø6 mm, el acabado con un cortador de bola de Ø6 mm, la desviación dimensional máxima que se presento fue de 0.2 mm.

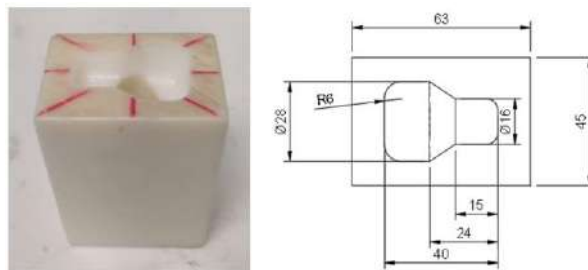


Figura 5.1 Mecnizado de pieza usando material Delrin.

Por otra parte, se realizaron maquinados en aluminio donde la eficiencia del maquinado se incrementó a velocidades de avance bajas, Figura 5.2. Los parámetros de corte seleccionados para un cortador con diámetro de 6 mm fueron: velocidad del husillo de 4500 rpm, velocidad de avance de 60 mm/min, velocidad de carga de 40 mm/min, profundidades de corte en el intervalo de 0.4-0.6 mm en el desbaste y 0.1-0.3 mm en acabado. De igual manera el desbaste se realizó con un cortador plano de Ø6 mm, el acabado con un cortador de bola de Ø6 mm, la desviación dimensional máxima que se presento fue de 0.13 mm.

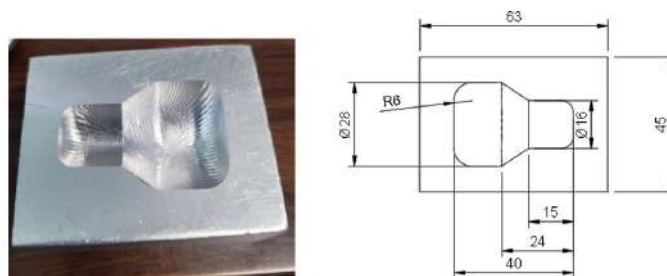


Figura 5.2 Mecnizado de pieza usando material de aluminio.

La Tabla 5.1 muestra la comparación dimensional entre el modelo y las dimensiones reales al efectuar el mecanizado robótico, el resultado de cada dimensión es el valor promedio de un total de ocho mediciones.

Tabla 5.1 Comparativa dimensional del mecanizado en material delrin y aluminio.

Delrin		Aluminio	
Modelo	Real	Modelo	Real
Ø 28	Ø 27.90	Ø 28	Ø 27.87
Ø 16	Ø 15.95	Ø 16	Ø 15.89
R6	R5.93	R6	R5.95
40	40.2	40	40.10
24	24.07	24	23.95
15	15.08	15	15.10

5.2 MAQUINADO DE SUPERFICIES PLANAS

Se diseñó y fabricó un porta-herramienta que permite acoplar un buril de acero de alta velocidad, con medida nominal de un cuarto de pulgada, ver Figura 5.3. El porta herramienta permite variar el diámetro de corte y ajustarse al collet del husillo ER11 manteniendo un diámetro de zanco de 7 mm. Los parámetros óptimos programados para la herramienta con diámetro de corte de 30 mm son:

- Desbaste: Profundidad de corte de 0.5 mm, velocidad del husillo 4500 rpm, velocidad de avance 60 mm/min, velocidad de carga de 40 mm/min.
- Acabado: Profundidad de corte de 0.1-0.3 mm, velocidad del husillo de 4500 rpm, velocidad de avance de 50 mm/min, velocidad de carga de 40 mm/min.

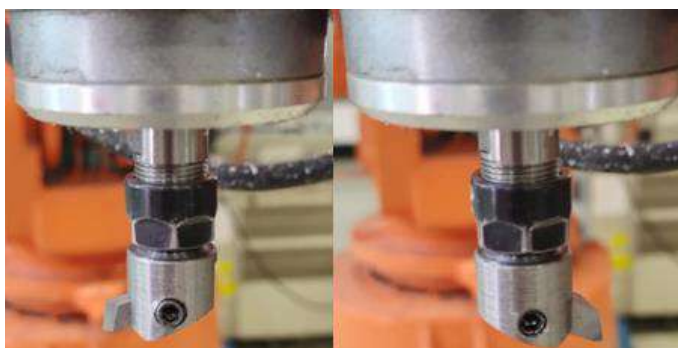


Figura 5.3 Herramienta de corte de diámetro 30 mm.

Se realizó el maquinado de una superficie plana en ambas direcciones programando los parámetros de corte óptimos para el acabado usando la herramienta con diámetro 30 mm. Para evaluar el acabado superficial se utilizó un rugosímetro *Mitutoyo sj-210*, haciendo tres mediciones en la superficie (izquierda, centro, derecha) en dirección de la ruta de maquinado Figura 5.4.

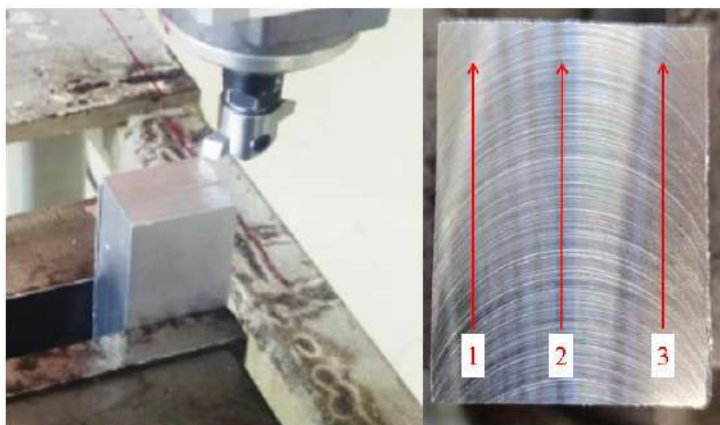


Figura 5.4 Maquinado en superficie plana.

La Tabla 5.2 muestra los valores de las mediciones de rugosidad en micrómetros (μm) de acuerdo con los parámetros: longitud de corte (λ_c) de .8 mm, velocidad de medición de .5 mm/s, longitudes de muestreo (N) de 5 bajo la normativa ISO1997.

Tabla 5.2 Mediciones de rugosidad (μm) en maquinado plano.

	Rugosidad Media Aritmética (RA)	Rugosidad media cuadrática RQ	Amplitud media (RZ)
1	1.704	2.051	8.083
2	1.593	1.933	7.431
3	1.632	1.970	8.116

5.3 MAQUINADO DE TRAYECTORIAS CURVAS CON ESTRATEGIAS MULTI-EJES

Fusion 360 permite controlar los ángulos de ataque de la herramienta en las estrategias multi-ejes, y así mejorar el acabado superficial. Es necesario utilizar cortadores de bola para las trayectorias de acabado curvas. Se realizó el maquinado de una pieza con un radio de 25 mm con ángulo de suavizado de 5° Figura 5.5,

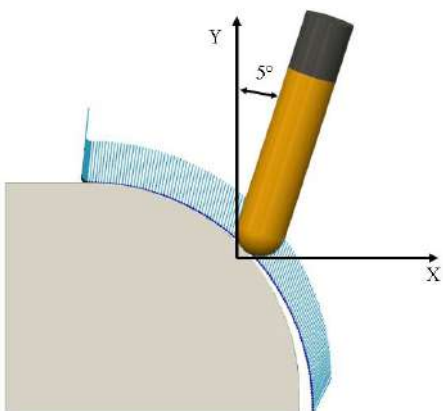


Figura 5.5 Ángulo de suavizado en maquinado de trayectoria curvas.

El maquinado se realizó en dos posiciones de la mesa de trabajo bajo los mismos parámetros para evaluar y comparar la rugosidad de la misma trayectoria respecto a dos configuraciones diferentes del brazo robótico, la primera posición se realizó en la parte superior izquierda y la segunda posición en la parte inferior izquierda Figura 5.6.

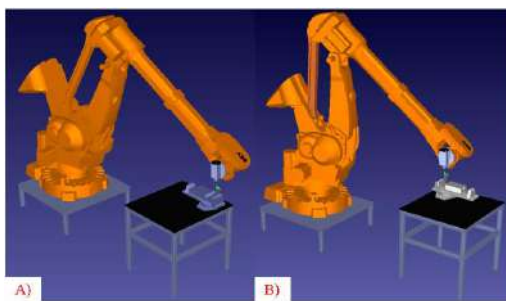


Figura 5.6 Posición de la prensa, A) Superior izquierda, B) Inferior izquierda.

La Tabla 5.3 describe los parámetros de corte programados en la trayectoria curva realizada en aluminio con un cortador de bola de diámetro 6 mm.

Tabla 5.3 Parámetros de corte para trayectoria curva de radio 25 mm.

Parámetro	Valor
Diámetro cortador	6 mm
Angulo de suavizado	5°
Velocidad del husillo	4500 rpm
Velocidad de Avance	60 mm/min
Velocidad de Carga	40 mm/min
Ancho de pasada	0.2 mm

Posteriormente para evaluar la rugosidad superficial con el rugosímetro *Mitutoyo sj-210* se realizaron 4 mediciones sobre la superficie Figura 5.7.

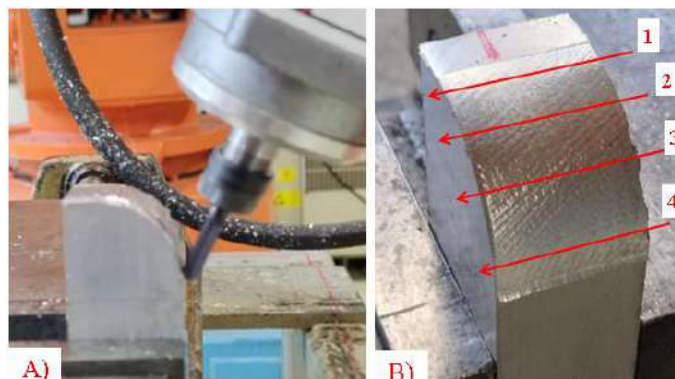


Figura 5.7 Maquinado de trayectoria curva y dirección de medición de rugosidad.

La Tabla 5.4 muestra los valores de las mediciones de rugosidad en micrómetros (μm) para la posición superior izquierda (1) y la posición inferior izquierda (2) de acuerdo con los parámetros del rugosímetro: longitud de corte (λ_c) de 0.8 mm, velocidad de medición de 0.5 mm/s, longitudes de muestreo (N) de 5 bajo la normativa ISO1997.

Tabla 5.4 Mediciones de rugosidad en (μm) de la trayectoria curva.

		Rugosidad Media Aritmética (RA)	Rugosidad media cuadrática RQ	Amplitud media (RZ)
1	1	2.355	2.949	12.116
	2	2.28	2.847	11.529
	3	2.282	2.758	10.605
	4	2.215	2.677	9.487
2	1	2.257	2.655	10.055
	2	2.008	2.308	7.716
	3	1.916	2.254	8.316
	4	2.062	2.436	8.815

La comparativa de los valores de rugosidad en la Tabla 5.4 respecto a las dos posiciones en las que se realizó el maquinado arroja una desviación máxima entre los valores de la tercera medición de rugosidad media aritmética de $0.366 \mu\text{m}$, concluyendo que: el robot es capaz de maquinar con el mismo acabado independientemente de la configuración de sus ejes.

5.4 MAQUINADO DE MODELO GENERADO MEDIANTE INGENIERÍA INVERSA

La implementación de ingeniería inversa para obtención de modelos 3D permite crear superficies a partir de escanear y digitalizar piezas a escala Figura 5.8 para posteriormente generar las trayectorias de maquinado multi-ejes.



Figura 5.8 Modelo 3D de automóvil generado por escáner.

Se realizó el maquinado del modelo real de una figura clásica de automóvil “Volkswagen tipo 1” en aluminio utilizando la herramienta de corte de diámetro 30 mm para el desbaste Figura 5.3, y un cortador de bola de diámetro de 6 mm. La Tabla 5.5 muestra los parámetros de corte programados para cada operación.

Tabla 5.5 Parámetros de corte para maquinado de figura de automóvil.

Parámetro	Desbaste diámetro 30 mm	Acabado diámetro 6 mm
Velocidad husillo	4500 rpm	4500 rpm
Velocidad de avance	60 mm/min	50 mm/min
Velocidad de carga	40 mm/min	40 mm/min
Profundidad máxima	0.5 mm	-
Carga óptima	24 mm	-
Ángulo de herramienta	90°	15°-30°
Ancho de pasada	-	0.3 mm

La programación de estrategias con multi-ejes permite inclinar el ángulo de la herramienta y maquinar superficies que no pueden ser maquinadas en una fresadora CNC de tres ejes Figura 5.9 . El rango de

ángulos se modifica respecto a los requerimientos del maquinado, el rango utilizado en el proyecto se encuentra entre 5°-35°, por otro lado, mediante el brazo robótico es posible mantener la punta del cortador normal a la superficie. Es necesario considerar los límites de cada eje y las colisiones que puedan existir.

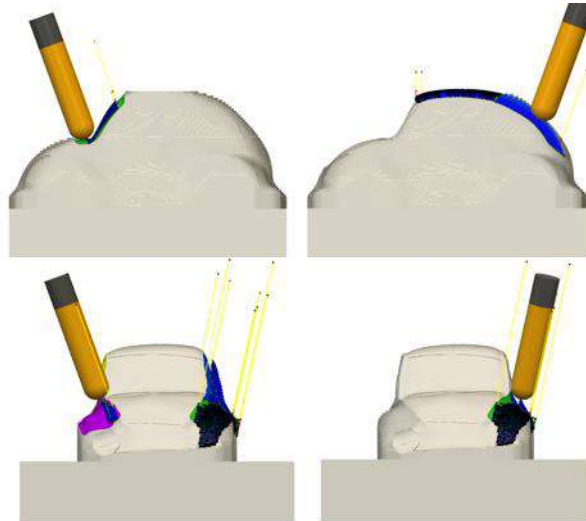


Figura 5.9 Inclinación de herramienta en estrategias Multi-ejes.

Al realizar el maquinado con un cortador de diámetro de 6 mm y con ancho de pasada de 0.3 mm en las operaciones de acabado, se pone en evidencia los cambios entre superficies y hendiduras Figura 5.10. La solución primaria consiste reducir el diámetro del cortador y el ancho de pasada, así como la velocidad de cambio en aproximación programada en el módulo de mecanizado de *RobotDK* inferior a 5 mm/s.

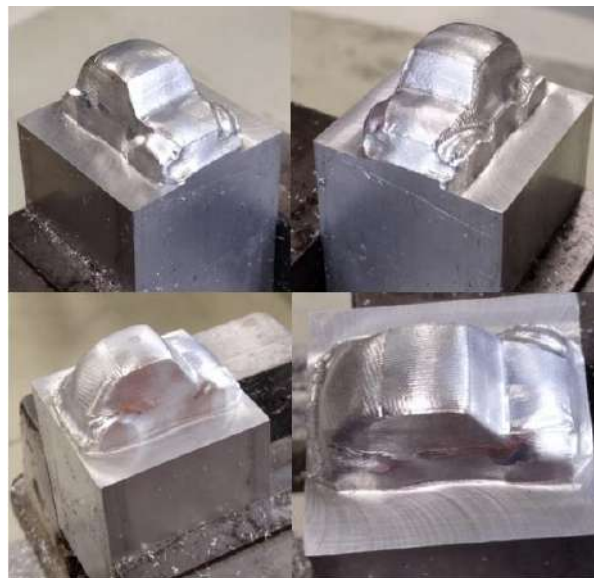


Figura 5.10 Maquinado de automóvil a escala.

5.5 ÁREAS DE OPORTUNIDAD

Uno de los problemas que surgieron durante el desarrollo del proyecto es relacionado con las vibraciones no deseadas del sistema. En diversos artículos se discute el efecto de las vibraciones en la calidad de la superficie producida. Debido a la baja frecuencia natural del cuerpo articulado del robot, puede producirse resonancia debido a las vibraciones del proceso de mecanizado [3].

Por otro lado, las máquinas CNC usualmente son muy robustas, proporcionando mayor rigidez en comparación con la estación del proyecto, en donde se tiene una mesa de acero con una altura de 80cm lo cual incrementa el par torsional debido a las fuerzas en el maquinado aumentando notoriamente la vibración la cual afecta directamente al acabado superficiales.

Es posible reducir la amplitud de las vibraciones aumentando la rigidez o amortiguando el sistema, los cambios en los parámetros de corte no influye significativamente en la magnitud de las vibraciones forzadas. Se observa que el amortiguamiento en general aumenta con la cantidad de componentes y áreas de contacto como se ilustra en Figura 5.11. Sin embargo, al aumentar el número de uniones, la rigidez general del sistema de la máquina suele disminuir [4].

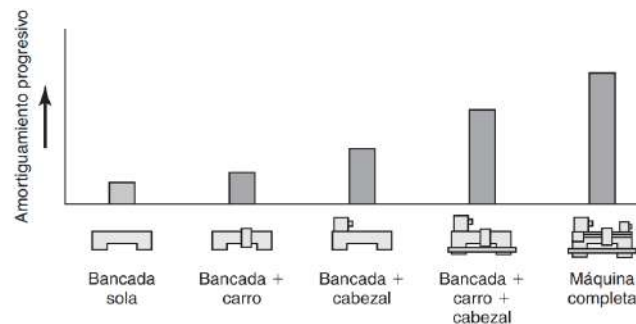


Figura 5.11 Amortiguamiento de vibraciones en función de la cantidad de componentes.

Otro de los problemas que surgieron en la programación es la aparición de puntos de singularidad que pueden afectar al movimiento del robot, a pesar de que dentro de la simulación virtual no se reporten errores de singularidad. Usualmente las singularidades aparecen en recorridos o trayectorias circulares forzando al brazo robótico a que alinee dos o más ejes, el otro caso se presenta cuando los ejes del robot se encuentran cerca de su límite de alcance ver Figura 5.12.

La solución implica dividir los maquinados de las superficies curvas en secciones pequeñas de esta manera el robot evita cambios bruscos en sus ejes para alcanzar la trayectoria, otra solución es colocar la pieza de trabajo en una posición distinta y compensar la rotación de la herramienta en el módulo de mecanizado de *RobotDk*.

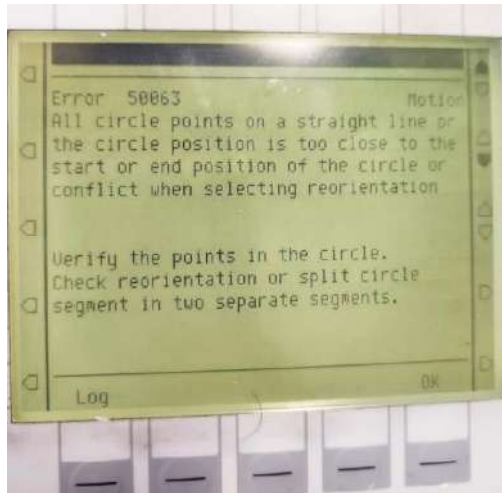


Figura 5.12 Errores de singularidad en trayectorias circulares.

En el proceso de mecanizado de piezas curvas usando las estrategias de multi-ejes, la problemática que se presenta son irregularidades debido a la oscilación que se presenta cuando el brazo robótico cambia de configuración y rota sus ejes para poder mantener el ángulo de ataque normal a la superficie de la herramienta, al realizar la acción se presenta inestabilidad debido a la rigidez de las articulaciones. En comparación con la simulación donde no se presenta la problemática en una pieza de revolución con un radio de 10 mm Figura 5.13.

La solución consiste en mantener constante el ángulo de la herramienta o en su defecto seccionar el maquinado para evitar los cambios repentinos de los ejes además de reducir la velocidad de aproximación en el módulo de mecanizado de RobotDk por debajo de 5 mm/s en los puntos donde el brazo robótico cambia la configuración de sus ejes.

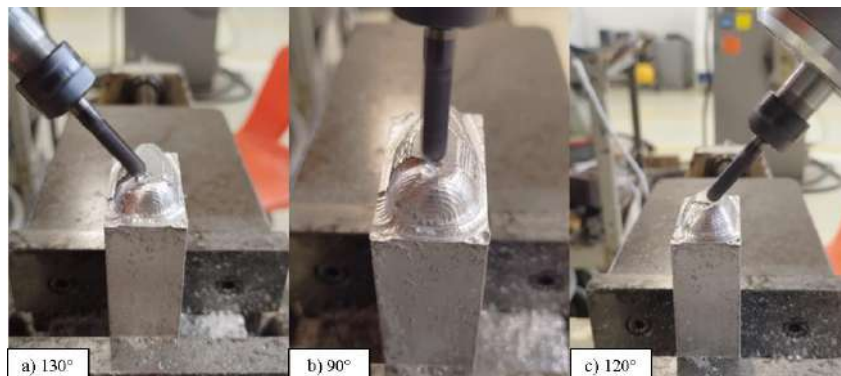


Figura 5.13 Problemática en mecanizado con multi-ejes.

CONCLUSIONES

Las principales ventajas de la estación de maquinado robótico radica en tres áreas importantes: cuenta con una área de trabajo superior al de una fresadora CNC, lo que permite programar trayectorias extensas; la segunda característica importante es la programación de seis ejes continuos ayuda a mecanizar áreas que no serían logradas en una máquina de tres ejes sin realizar cambios de orientación de las piezas además de permitir variar los ángulos de inclinación del eje de la herramienta para obtener acabados uniformes; la tercer ventaja es la flexibilidad de programación de las trayectorias que ofrece adaptar la estación para maquinar diferentes tipos de geometrías en el mismo espacio de trabajo.

Por otra parte, en el desarrollo del proyecto se presentaron diversas problemáticas como la vibración que se genera al maquinar, la tendencia para solucionar el problema es aumentar la rigidez de los componentes de la estación además la programación permite variar los parámetros para corregir singularidades y encontrar la relación entre la velocidad de avance y ancho de pasada de corte, adecuado a cada material y contribuir a la disminución de la vibración.

La precisión de la estación radica dos aspectos importantes: primero en la capacidad para definir el espacio de trabajo, es importante definir con exactitud el centro de la herramienta y los planos de trabajo ya que al ser procesos que involucran la habilidad del operador y la perspectiva, de manera involuntaria se generarán errores en los valores que afectarán directamente al control dimensional, segundo el estado físico y las características específicas del robot como precisión, repetibilidad, exactitud del control de trayectoria, rigidez, estabilidad estructural y un entorno controlado afectaran directamente al control dimensional.

Al efectuar una conformación apropiada entre la definición del espacio, la programación y la aplicación de estrategias de mecanizado convenientes para el material seleccionado, el uso de un robot *ABB* en una celda de trabajo resultará en una reducción de costos de equipos significativa en comparación con un centro de maquinado.

Los objetivos del proyecto fueron cubiertos en su totalidad para finalizar con la automatización de la estación de mecanizado robótico agregando únicamente la tarjeta de *Arduino Mega* para el control de la velocidad del husillo asemejándose a una máquina CNC moderna. Además, se establecieron métodos para la generación de modelos CAD mediante ingeniería inversa, la generación de programas de control numérico y el desarrollo de maquinados mediante el sistema integrado.

Personalmente estoy satisfecho con la realización del proyecto, las tareas que se efectuaron de principio a fin ayudaron a reafirmar mis conocimientos sobre la programación de control numérico. Es un plus

tener personal dentro del Laboratorio de Manufactura capacitado para apoyar a los estudiantes, además de la participación de mis asesores que me orientaron en el área que me causaba incertidumbre que es la robótica ya que es la primera vez que realice un trabajo acerca de este tema. El proyecto me ha enseñado la importancia de la planificación y la atención al detalle en la fase de diseño e implementación, me ha proporcionado una valiosa experiencia práctica en la resolución de problemas que surgen al trabajar con equipos robóticos. Este trabajo no solo me ha permitido aplicar conocimientos teóricos en un entorno real, sino que también me ha dado una visión clara de las posibilidades y limitaciones actuales del maquinado robótico.

NOMBRE DEL TRABAJO

Tesis_JoseDanielMataV1.pdf

AUTOR

José Daniel Mata Medina

RECuento DE PALABRAS

21146 Words

RECuento DE CARACTERES

119811 Characters

RECuento DE PÁGINAS

114 Pages

TAMAÑO DEL ARCHIVO

6.2MB

FECHA DE ENTREGA

Sep 2, 2024 10:31 AM CST

FECHA DEL INFORME

Sep 2, 2024 10:36 AM CST**● 15% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 11% Base de datos de Internet
- Base de datos de Crossref
- 10% Base de datos de trabajos entregados
- 3% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

REFERENCIAS

- [1] M. P. A. J. E. A. L. U. F. & P. F. J. Groover, Introducción a los procesos de manufactura, McGraw-Hill Interamericana, 2014.
- [2] Koren, Computer Control of Manufacturing Systems, McGraw Hill, 1983.
- [3] J. D. C. S. P. & C. G. Pandremenos, « Machining with robots: a critical review.,» *Proceedings of DET2011*, pp. 1-9, 2011.
- [4] S. & S. S. R. Kalpakjian, Manufactura, ingeniería y tecnología, Pearson educación, 2002.
- [5] R. B. O. C. T. A. V. I. A. N. C. A. L. & R. S. G. Breaz, «Using Serial Industrial Robots in CNC Milling Procesess. In Proc. IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics.,» *In Proc. IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics.*, 2015.
- [6] ABB, «new.abb.com,» 2024. [En línea]. Available: <https://new.abb.com/news/detail/112647/cstmr-swedish-metal-fabricator-beats-capacity-squeeze-using-abb-cobots>.
- [7] KUKA Robots, «KUKA,» 2016. [En línea]. Available: <https://www.kuka.com/es-mx/empresa/prensa/news/2016/09/kuka-robot-mills-oversized-plastic-pipes>.
- [8] Q. G. Z. Y. Y. J. J. X. Wenbo Wang, «Robotics and Computer-Integrated Manufacturing,» *Wenbo Wang, Qiang Guo, Zhibo Yang, Yan Jiang, Jinting Xu, A state-of-the-art review on robotic milling of complex parts with high efficiency and precision, Robotics and Computer-Integrated Manufacturing, Volume 79, 2023, 102436, ISSN 0736-5845.*, vol. volume 79, n° 102436, 2023.
- [9] A. K. S. C. B. F. A. P. Yier Wu, «Geometric calibration of industrial robots using,» *Robotics and Computer-Integrated Manufacturing*, 2015.
- [10] J. & G. C. DePree, «Robotic machining white paper project,» *Halcyon Development-Robotic Industries Association (RIA).*, 2008.

- [11] X. T. C. Z. F. P. R. Y. L. Z. Z. L. J. W. Zerun ZHU, «High precision, and efficiency robotic milling of complex parts: Challenges, approaches and trends,» *Chinese Journal of Aeronautics* , vol. 35, pp. 22-46, 2022.
- [12] Y. & B. A. A. Altintas, Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design, vol. V, 2001.
- [13] A. B. B. C. P. L. F. & A. R. Cruz, Fundamentos de robótica., 2007.
- [14] J. F. M. J. R. P. J. Pérez Turiel, «APLICACIONES DE LA ROBÓTICA: ULTIMAS TENDENCIAS Y NUEVAS PERSPECTIVAS,» *DYNA*, p. 61, 2002.
- [15] S. K. Saha, Introducción a la robótica, Mexico: McGRAW-HILL/INTERAMERICANA EDITORES, 2010.
- [16] P. A. ABB Robotics, Product Manual IRB 6400, Suecia, 1993.
- [17] J. & B.-C. S. Braumann, «Parametric robot control. Integrated CAD/CAM for Architectural Design,» *ACADIA*, pp. 242-251, 2011.
- [18] J. D. E. L. CO, Operation Manual H100 Series High Performance Vector Control, 2021.
- [19] G. Thomas, «Introduction to the Modbus Protocol,» *The Extension*, pp. 1-4, 2008.
- [20] R. Zurawski, Industrial Communication Technology Handbook, vol. X, San Francisco, California: CRC press, 2014.
- [21] W. Wang, Reverse engineering: Technology of reinvention, Crc Press, 2010.
- [22] V. Raja y K. J. Fernandes, Reverse engineering: an industrial perspective, Springer Science & Business Media., 2007.
- [23] J. R. Carreón Villal, Artist, *Diseño y Operacion de una estación Robótica tipo industrial*. [Art]. Instituto Tecnológico de Celaya, 2014.
- [24] Shinning 3D, USER MANUAL EinScan Pro, 2021.
- [25] P. A. ABB Robotics, User's Guide, Suecia, 1993.

[26] P. A. ABB Robotics, RAPID Reference Manual, Suecia, 1993.

ANEXOS

ANEXO 1 PARÁMETROS DE CORTE EN OPERACIÓN DE FRESADO

Tabla A. 1 Parámetros de corte.

Material	Velocidad de corte (V_c) m/min	Avance por filo (f) mm/filo
Aluminio	70-100	0.05-0.1
Latón	50-70	0.05-0.1
Hierro fundido	25-40	0.025-0.05
Acero bajo al carbono	30-40	0.025-0.05
Acrílico	20-30	0.035-0.075
Plástico (Delrin)	25-40	0.04-0.95

Tabla A. 2 Parámetros de corte en centros de fresado numérico comerciales.

Material	Velocidad corte (V_c)m/min		Avance (F) mm/min		
	Desbaste	Acabado	Acabado (mm) Profundidad	Desbaste (mm) Profundidad	
			0-1mm	1-3mm	3-6mm
Aluminio	90-120	150-200	100-120	120-150	80-100
Latón	80-100	100-120	100-120	120-150	80-100
Hierro fundido	60-70	90-110	55-65	65-75	45-55
Acero bajo carbono	100-200	120-160	45-55	55-65	35-45
Acero aleado	80-100	110-140	35-45	45-55	25-35
Acrílico	50-70	90-110	100-120	120-150	80-100
Plástico (Delrin)	50-70	100-150	150-200	120-150	50-70

Tabla A. 3 Parámetros de corte óptimos del proyecto.

Material	Velocidad del Husillo (rpm)	Avance F (mm/min)	Velocidad de carga (mm/min)	Profundidad de corte (mm)	
				Desbaste	Acabado
Delrin	2000-3500	80-110	50-70	0.5-0.8	0.1-0.3
Aluminio	3500-4500	50-70	30-40	0.4-0.6	0.1-0.3

ANEXO 2 ARRANQUE EN FRIO DEL BRAZO ROBÓTICO.

El arranque en frío consiste en cargar los archivos de configuración al controlador del robot, es necesario seguir los siguientes puntos:

1. Verificar las conexiones eléctricas del robot y controlador.
2. Contar con los archivos de instalación en una memoria *USB*, para guardar archivos en memoria *USB* es necesario realizar la siguiente secuencia:
 - Abrir *USB Floppy Manager* como administrador
 - Insertar y formatear la memoria en formato 1.44M en cien particiones.

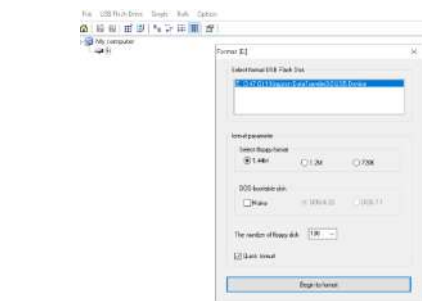


Figura A. 1 Formateo de memoria en *floppy manager*.

- Ubicar las carpetas de archivos *Key* y *baseware 3.1* (proporcionados por el asesor).

El archivo *key* depende del número de serie del controlador, el número 64-14775 corresponde al modelo del proyecto.

Nombre	Fecha de modificación	Tipo	Tamaño
BaseWare	20/02/2024 11:02 a.m.	Carpeta de archivos	
Key Disk	20/02/2024 11:02 a.m.	Carpeta de archivos	

Nombre	Fecha de modificación	Tipo	Tamaño
64-06869	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06870	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06871	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06872	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06873	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06874	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06875	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06876	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06877	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06878	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-06885	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-14314	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-14775	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-14776	20/02/2024 11:02 a.m.	Carpeta de archivos	
64-14829	20/02/2024 11:02 a.m.	Carpeta de archivos	

Figura A. 2 Archivos *key* según número del robot.

El archivo *baseware* depende del controlador

Nombre	Fecha de modificación	Tipo	Tamaño
BaseWare 3.0 3HAB2390-2 Ver. 49	20/02/2024 11:02 a. m.	Carpeta de archivos	
BaseWare 3.1 3HAB2396-2 Ver. 23	20/02/2024 11:02 a. m.	Carpeta de archivos	
BaseWare 3.2 3HAC2700-2 Ver. 06	20/02/2024 11:03 a. m.	Carpeta de archivos	
Folder Settings	20/02/2024 11:03 a. m.	Carpeta de archivos	

BaseWare > BaseWare 3.1 3HAB2396-2 Ver. 23			
Nombre	Fecha de modificación	Tipo	Tamaño
BootDisk 1(8)	20/02/2024 11:02 a. m.	Carpeta de archivos	
BootDisk 2(8)	20/02/2024 11:02 a. m.	Carpeta de archivos	
BootDisk 3(8)	20/02/2024 11:02 a. m.	Carpeta de archivos	
BootDisk 4(8)	20/02/2024 11:02 a. m.	Carpeta de archivos	
BootDisk 5(8)	20/02/2024 11:02 a. m.	Carpeta de archivos	
BootDisk 6(8)	20/02/2024 11:02 a. m.	Carpeta de archivos	
BootDisk 7(8)	20/02/2024 11:02 a. m.	Carpeta de archivos	
BootDisk 8(8)	20/02/2024 11:02 a. m.	Carpeta de archivos	

Figura A. 3 Archivos baseware 3.1.

- Guardar en *Floppy Manager* los archivos “Key” y “Baseware 3.1”

Es necesario copiar los archivos en una dirección *Floppy de la USB*, una carpeta para el archivo *Key 64-14775* además de guardar independientemente cada “boot disk” del *baseware 3.1*, de esta forma se generarán un total de nueve *Floppys* en la memoria *USB*.

The figure consists of three screenshots of the 'Batch Manage Tool V1.31T' software interface. The top screenshot shows the 'Upload floppy disk block' window with a table of disk blocks (000-005) and their status. The middle screenshot shows a file explorer view of the 'FLOPPY000' directory containing files like 'AccessoRaid', 'key.d', and 'key'. The bottom screenshot shows the 'Download floppy disk block' window with a table of disk blocks (000-007) and their status.

Block number	Capacity	Used	Renew time
000	1.39 MB	2%	2024-02-07 15:43:25
001	1.39 MB	61%	2024-02-22 09:54:47
002	1.39 MB	61%	2023-12-11 10:52:15
003	1.39 MB	61%	2023-12-11 10:53:06
004	1.39 MB	61%	2023-12-11 10:53:35
005	1.39 MB	61%	2023-12-11 10:54:00

Block number	Capacity	Used	Renew time
000	1.39 MB	2%	2024-02-07 15:43:25
001	1.39 MB	61%	2024-02-22 09:54:47
002	1.39 MB	61%	2023-12-11 10:52:15
003	1.39 MB	61%	2023-12-11 10:53:06
004	1.39 MB	61%	2023-12-11 10:53:35
005	1.39 MB	61%	2023-12-11 10:54:00
006	1.39 MB	61%	2023-12-11 10:54:36
007	1.39 MB	61%	2023-12-11 10:54:06

Figura A. 4 Cargar archivos en unidad *floppy*.

3. Desconectar las baterías previamente por cinco minutos. Posteriormente reconectarlas.
4. Encender el controlador.
5. En el *Teach Pendant* aparecerá el mensaje para insertar el *Key disk*, el cual debe ser el correspondiente al número de serie del manipulador (64-14775).

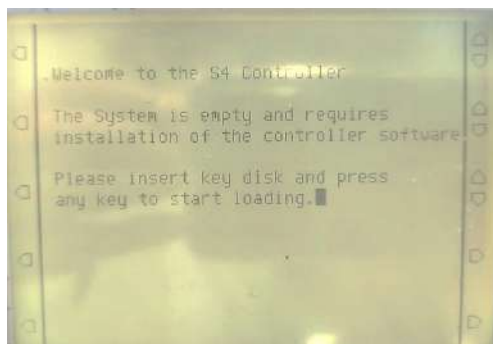


Figura A. 5 Insertar el disco key correspondiente.

6. Posteriormente en el *Teach Pendant* aparecerá el mensaje para insertar cada “*boot disk*” en este caso correspondientes al *Baseware 3.1*.

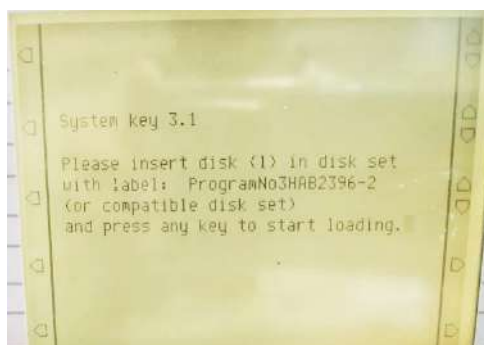


Figura A. 6 Insertar los discos de instalación.

7. Después de insertar los “*boot disk*” aparecerán tres opciones para el modo de instalación:

- *Silent* = Es una instalación rápida, no pide información extra.
- *AddOpt* = Permite elegir idioma, configurar hora y fecha.

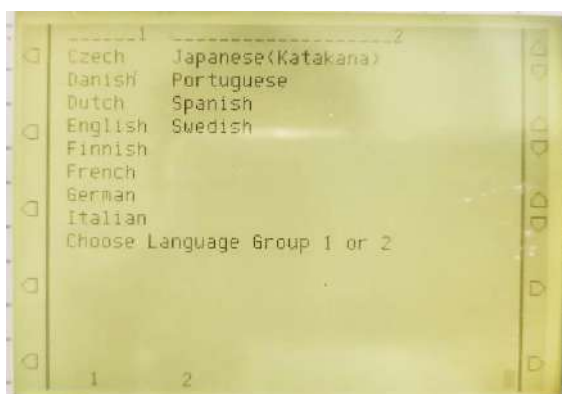


Figura A. 7 Seleccionar el lenguaje.

- *Query* = Pide archivos de respaldo de parámetros del manipulador. En caso de seleccionar *query* aparecerán más pasos.

Seleccionar *Silent* en caso de no tener respaldo de los parámetros del manipulador.

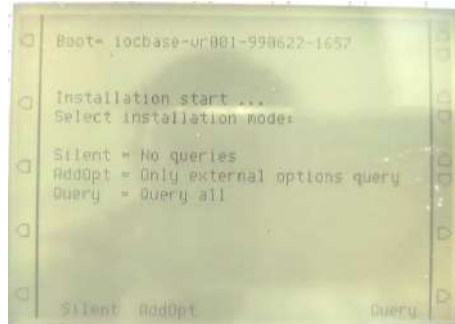


Figura A. 8 Seleccionar el tipo de instalación.

8. Al finalizar la instalación se debe seleccionar *DC2* en función del modelo.

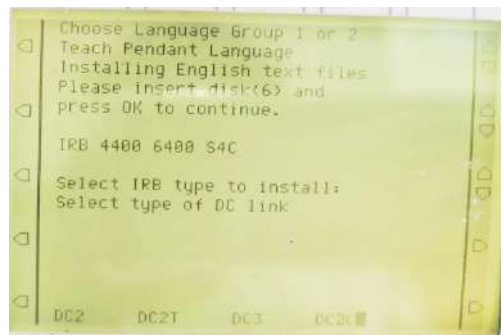


Figura A. 9 Seleccionar el tipo de DC link.

El manipulador se reiniciará automáticamente y encenderá normalmente.

9. Al finalizar el arranque en frío, es posible que se muestren mensajes de error en el *Teach Pendant*:
 - Baterías descargadas: es conveniente dejar el controlador en uso por 36 horas continuas.
 - Ejes no encontrados: es necesario calibrar el manipulador.
 - Contador de revoluciones desactualizado: es necesario calibrar los motores de los ejes.



Figura A. 10 Errores comunes en el arranque.

Finalmente se podrá manipular el brazo robótico mientras no aparezcan mensajes que señalen un error fatal.

ANEXO 3 CALIBRACIÓN DE LOS EJES

Cuando al encender de nuevo el controlador se presenta el error de “ejes no encontrados” debido a que el robot pierde referencia, es necesario llevar a cabo el proceso de calibración y actualización de los contadores de rotación de los motores.

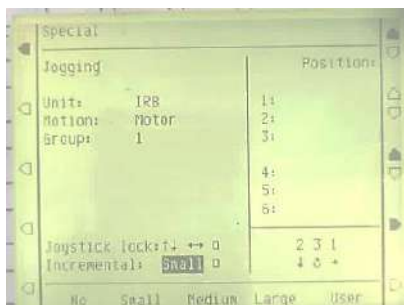


Figura A. 11 Ejes no calibrados.

El proceso de calibración consiste en la siguiente secuencia:

1. Posicionar los seis ejes del robot en sus marcas de calibración de manera manual en la ventana *joging* del *Teach Pendant*.

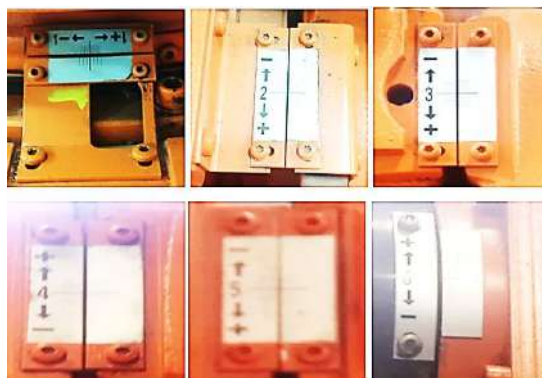


Figura A. 12 Posicionamiento de ejes.

2. Seguir la siguiente ruta en el *Teach Pendant*: *Menú de otras ventanas > Service > View > Calibration > Fine Calibrate o Rev.counter Update*.

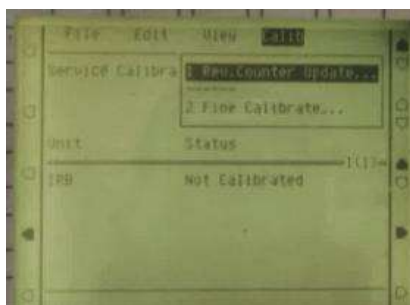


Figura A. 13 Ventana de calibración.

3. Es necesario realizar ambos procesos de calibración en “*fine calibrate* y *rev.counter update*”. Una vez posicionados los ejes, el controlador permite seleccionar la opción de calibrar todos los ejes simultáneamente seleccionando los seis ejes y presionando *OK*.

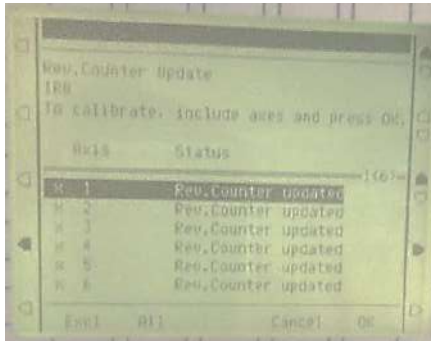


Figura A. 14 Calibración de los contadores de revolución.

4. Al concluir el proceso aparecerá un mensaje que indica que el robot fue sincronizado (*IRB synchronized*). En la ventana de jogging aparecerá la referencia de cada eje posicionado en cero.

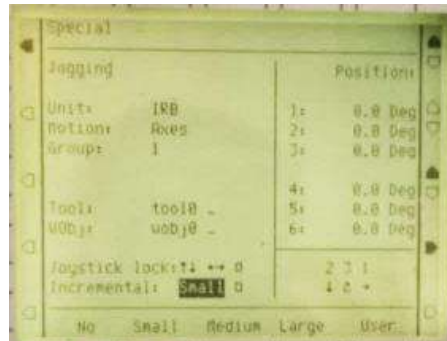


Figura A. 15 Calibración de los ejes.

ANEXO 4 PROGRAMA DE POSICIONAMIENTO DE CALIBRACIÓN.

El programa de posicionamiento en la posición de origen se utiliza una vez realizada la calibración, es recomendable regresar el robot a su posición antes de apagarlo mediante un programa como el que se presenta a continuación, ya que al realizar el proceso de calibración de manera manual existirán variaciones en los planos de referencia.

%%%

VERSION:1

LANGUAGE:ENGLISH

%%%

MODULE Home

! Tool variables:

PERS tooldata tool3 := [TRUE,[-152.503,2.000,49.500],[0.50000000,-0.50000000,-0.50000000,0.50000000]],1,[0,0,20],[1,0,0,0],0,0,0.005]; ! opcional, ya que no siempre es la misma herramienta.

! Reference variables:

PERS wobjdata wobj3 := [FALSE, TRUE, "", [[2079.700,232.400,723.400],[1.00000000,0.00000000,0.00000000,0.00000000]],[[0,0,0],[1,0,0,0]]];

PROC Main()

ConfJ \On;

ConfL \Off;

! Program generated by RoboDK v5.5.4 for ABB IRB 6400 2.8 120 on 19/10/2023 14:53:11

! Using nominal kinematics.

MoveAbsJ

[[0.000000,0.000000,0.000000,0.000000,0.000000,0.000000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z1,tool3 \WObj:=wobj3;

ENDPROC

ENDMODULE

ANEXO 5 DEFINICIÓN DE PLANOS DE TRABAJO.

La definición de los planos de trabajo es sumamente importante para crear trayectorias de maquinado debido a que el controlador requiere posicionar las coordenadas de la herramienta respecto de un sistema coordenado tridimensional que a su vez será posicionado respecto al sistema coordenado de referencia de la base del robot “*Wobj0*”. La definición de los planos de trabajo se realiza mediante la siguiente secuencia:

1. Crear un programa nuevo o abrir uno existente donde previamente se definieron las coordenadas del TCP de la herramienta.

Dirigirse a la ruta: *Programas >File > Open > (Seleccionar el programa) > OK*

En caso de crear un nuevo programa, seguir la siguiente ruta: *Programas >File > New > (Nombrar programa) > OK*

2. Se definirá el nombre del plano, la herramienta con que se definirán los puntos del plano, el tipo de plano “*User u Objetc*”. El plano se definirá por medio de tres puntos, dos en X y uno en Y el primer punto será el origen del plano.

Dirigirse a la ruta de Data types: *View> Data Types > wobjdata> New*

Seleccionar el nuevo *wobj*, presionar el menú *Special* y elegir la opción *define Coord*. A continuación, seleccionar el método de los tres puntos para el tipo de plano y seleccionar la herramienta con la cual se configurarán lo puntos.

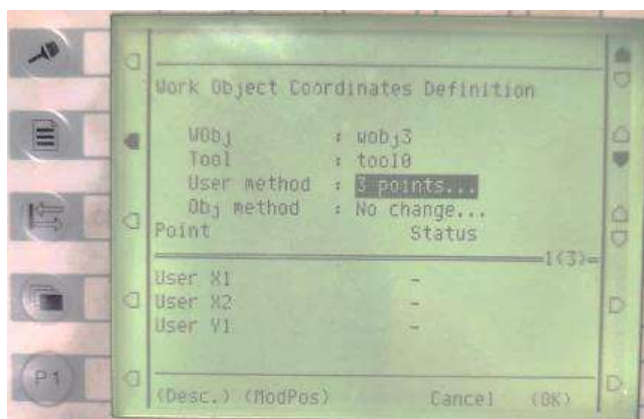


Figura A. 16 Definición de plano de trabajo.

3. Posicionar el TCP de la herramienta un punto para el origen del plano, un punto para el eje X positivo y otro para el eje Y positivo, Z se configura automáticamente de acuerdo con la regla de la mano derecha.

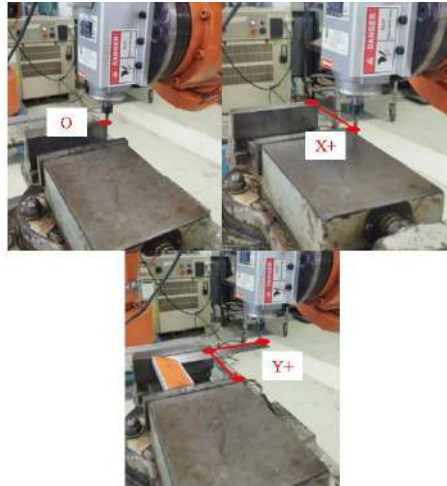


Figura A. 17 Localización de puntos del plano.

4. Guardar las coordenadas de cada punto presionando “modpos” en la ventana de programas.

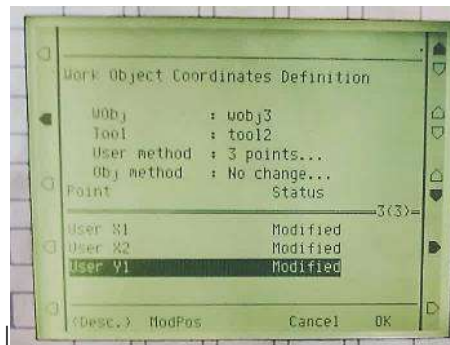


Figura A. 18 Guardar las coordenadas de los puntos.

5. Visualizar las coordenadas del plano para posteriormente pasar los datos a *RobotDk*



Figura A. 19 Visualización de los datos del plano.

En la ruta *Programas > View > Data types > wobjdata > “plano creado” > Ok* se encuentran los planos de referencia existentes.

ANEXO 6 DEFINICIÓN DEL TCP

TCP (Tool Center Point) es un concepto fundamental en la robótica, especialmente cuando se trabaja con robots para aplicaciones de maquinado CNC. Definir correctamente el *TCP* es crucial por varias razones: El *TCP* permite que el robot conozca la posición exacta de la herramienta que está utilizando, esto es esencial para realizar maquinados precisos, donde cualquier desviación puede resultar en errores de fabricación. Además, ayuda a calcular correctamente las trayectorias de movimiento del robot, asegurando que la herramienta siga el camino deseado sin desviaciones.

La definición del *TCP* se realiza siguiendo la siguiente secuencia:

1. Abrir o crear el programa donde se definirá la herramienta.

Botón Programa >File > Open > (Seleccionar el programa) > OK

Para crear un nuevo programa: *Botón Programa > File > New > (Nombrar el programa) > OK*

2. En la ventana de programas dirigirse a la ruta de Data types: *View> Data Types > tooldata> New*

Se definirá el método de definición dando clic en el botón *Special> Define. Coord. >ok*, existen 4 métodos:

- 4 puntos: se alcanzan 4 diferentes orientaciones de la punta de la herramienta en un mismo punto de manera manual mediante el Teach Pendant.
- 4 puntos sin cambiar orientación: el eje Z de la herramienta mantiene la dirección del efector seis.
- 5 puntos: Permite definir Z positivo con la orientación del efector final.
- 6 puntos: se guardan cuatro posiciones además definir Z y X positivos con la orientación del efector final.



Figura A. 20 Calibración del TCP.

3. Guardar los puntos de calibración del *TCP*, el método de los 6 puntos ayuda a configurar la orientación del eje de la herramienta y así poder ajustarla de tal manera que se encuentre normal

a la superficie, se graban los seis puntos posicionando cada uno y dando clic en la opción *Modpos*. Las cuatro primeras posiciones son arbitrarias tratando de que se encuentren una sobre cada cuadrante del punto, para definir la orientación de Z debe procurarse que el eje del cortador se encuentre normal a la superficie.

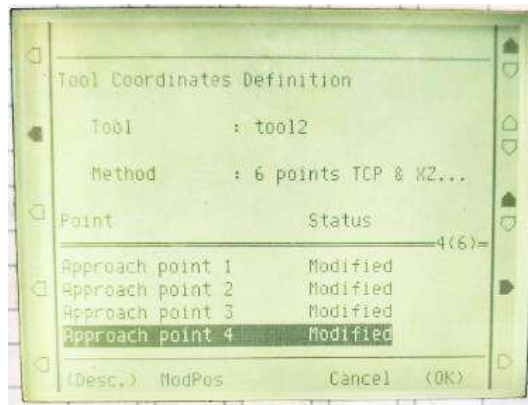


Figura A. 21 Definición de los puntos del TCP.

4. Visualizar las coordenadas de la herramienta para posteriormente pasar los datos a *RobotDk*

En la ruta *Programas > View > Data types > tooldata > "tool creada" > Ok* se encuentran los planos de referencia existentes.

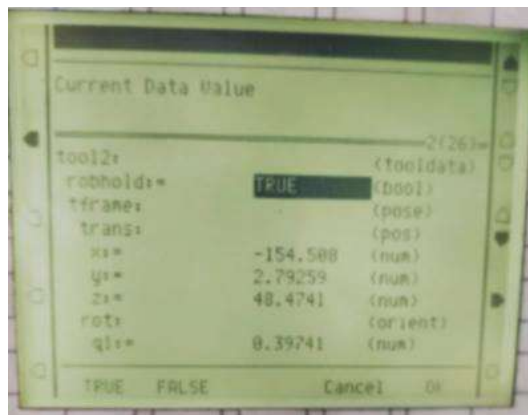


Figura A. 22 Visualización de los datos del TCP.

ANEXO 7 HABILITACIÓN DE LOS MÓDULOS I/O

El controlador IRB 6400 cuenta con cuatro módulos de entradas y salidas digitales, son los módulos 328 a 24v DC. El diagrama de conexiones se muestra a continuación

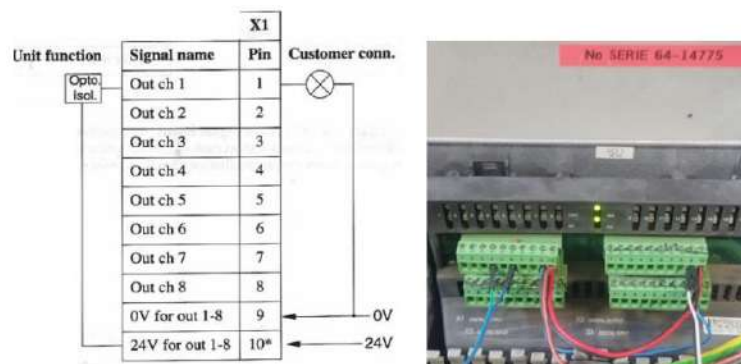


Figura A. 23 Módulo 10 de señales digitales.

El módulo inferior derecho tiene la dirección 10, el inferior izquierdo la 11, el superior izquierdo la 12 y el superior derecho 13 cada módulo tiene que habilitarse en el controlador individualmente mediante los siguientes pasos:

1. Dirigirse a la siguiente ruta: *Menú otras ventanas > System Parameters > IO signals > Types > Unit*
2. En la ventana de “Unit” dar clic en “Add” para agregar y configurar el módulo 328. Se desplegará una ventana:

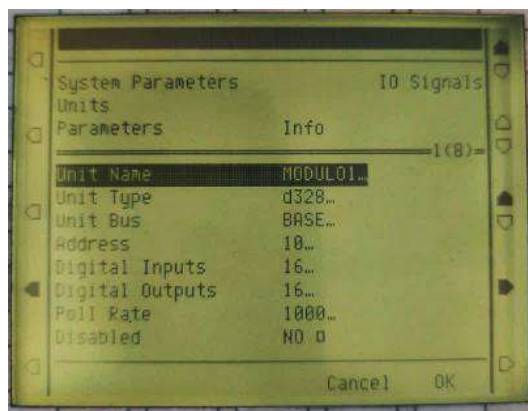


Figura A. 24 Parámetros del módulo de señales digitales.

Se configuran los campos de la siguiente manera: nombrar el módulo (módulo1), tipo de módulo (328), dirección (10 corresponde inferior derecha), número de entradas y salidas digitales (16), poll rate (1000).

3. Posteriormente reiniciar el controlador *Menú otras ventanas > Service > Restart*



Figura A. 25 Reiniciar el controlador.

4. En la ventana de “*User Signals*” seleccionar “*Add*” para agregar y configurar las entradas o salidas en el módulo para poder identificarlas en la programación. Dirigirse a la ruta:

Menú otras ventanas > System Parameters > IO signals > Types > User Signals

Se despegará una ventana para declarar cada señal digital:

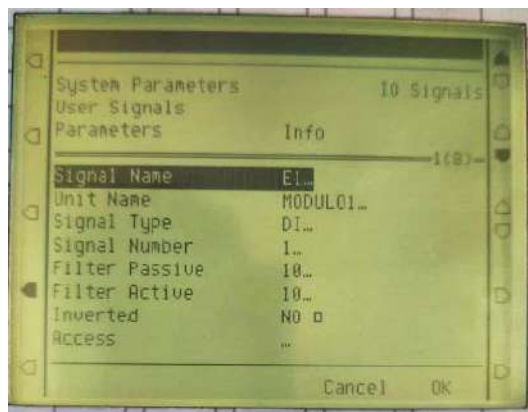


Figura A. 26 Parámetro para configurar señal digital.

Se configuran los campos de acuerdo con la siguiente manera: nombre de la señal (E para entrada y S para salida), módulo al cual pertenecen (1-4), tipo de señal (DI-entrada y DO-salida), número de señal (1-16), *Poll Rate* (100), *Filter Pasive* (10) y *Filter Active* (10), *inverted* (no).

La ventana de “*user signals*” aparecerá como se muestra en la Figura A. 27.

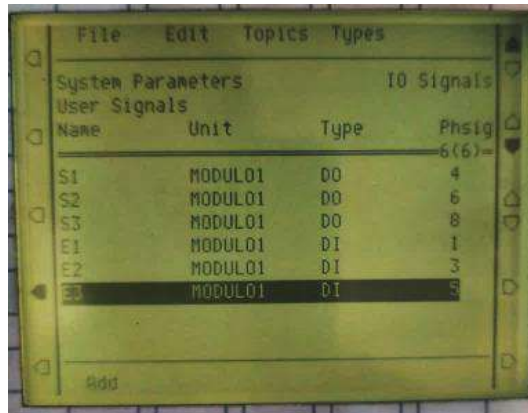


Figura A. 27 Señales digitales del módulo.

5. El controlador solicitará reiniciar para poder guardar las modificaciones.

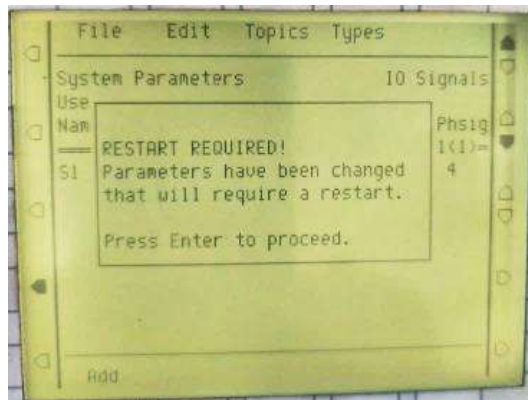


Figura A. 28 Mensaje de reinicio de sistema.

6. Corroborar que las señales aparezcan declaradas en la ventana de señales digitales "I/O".

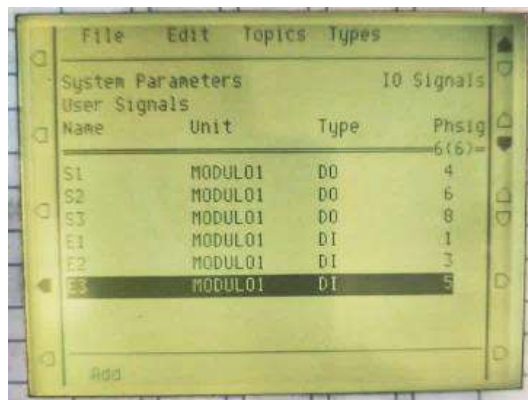


Figura A. 29 Señales digitales declaradas.

ANEXO 8 ENVIÓ DE DATOS POR PUERTO SERIAL.

El controlador del proyecto IRB 6400 cuenta con dos módulos para comunicación serial por medio de los puertos RS232 y RS485. El puerto designado “sio1” físicamente es un conector DB9 macho y el puerto “sio2” es un conector DB9 hembra.

Para utilizar la comunicación serial es necesario configurar el puerto serial en la siguiente ruta:

Menú otras ventanas > System Parameters > Comunicación > (seleccionar el canal SIO1-SIO2)

Se configuran los parámetros de envío o recepción de datos como *el baud rate y la paridad, número de bits de parada etc.* Al habilitar la comunicación serial se puede establecer comunicación con dispositivos externos, tal es el caso de la recepción de datos en una PC provenientes del controlador donde son interpretados por un programa como *Hyperterminal*.

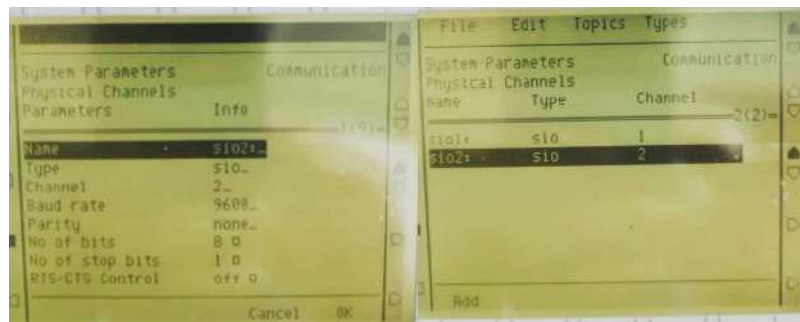


Figura A. 30 Parámetros del puerto serial.

Una vez configurado el módulo serial es necesario crear un programa para habilitar y enviar los datos del controlador que se clasifican de la siguiente manera:

- *bool* datos lógicos (verdadero o falso).
- *num* valores numéricos (decimal o entero).
- *string* cadenas de caracteres.
- *byte* valores de 0 hasta 255.

La secuencia en el programa para enviar datos tipo “string” es la siguiente:

```
%%%
```

```
VERSION:1
```

```
LANGUAGE:ENGLISH
```

```
%%%
```

```
MODULE COMUNICACION
```

```
VAR iodev frase; ! Asigna la variable
```

```
PROC main()
```

```
Open "sio2: ",frase\Write; ! Habilita el puerto serial "sio2"
```

```
Write frase,"MAESTRIA2024"; ! Envía la cadena de caracteres escrito entre las ""
```

```
ENDPROC
```

```
ENDMODULE
```

Los parámetros del programa para la recepción de datos en la PC (*Hyperterminal*) se configuran de la siguiente manera:

1. Elegir el nombre de la conexión y el tipo.



Figura A. 31 Nueva conexión de puerto.

2. Seleccionar el puerto serial donde físicamente se encuentra la conexión.



Figura A. 32 Selección del puerto serial.

3. Configurar los parámetros de la comunicación de la misma forma que los del puerto “sio2” del controlador.

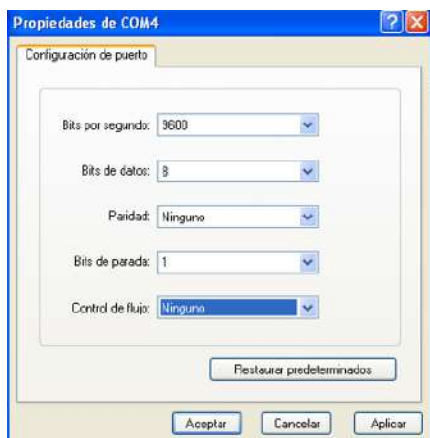


Figura A. 33 Configuración del puerto serial.

4. Recibir los datos.

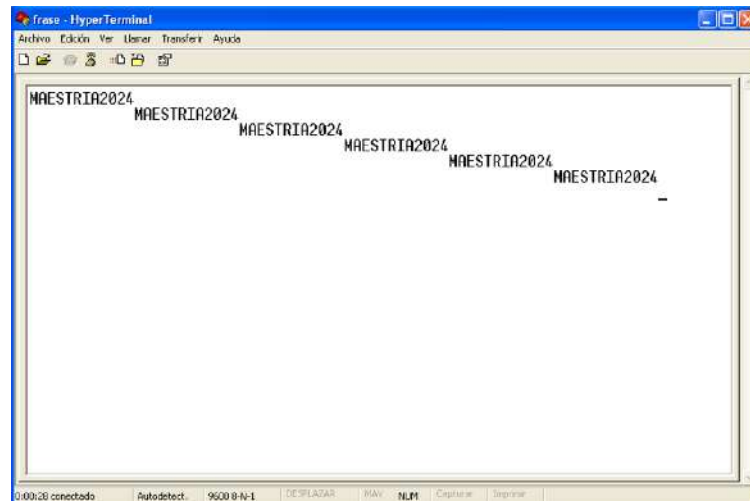


Figura A. 34 Recepción y visualización de datos.

Los datos pueden ser recibidos por diversos medios de comunicación por ejemplo un controlador Arduino, una computadora PC con Python, donde se interpretarán.

ANEXO 9 PLANO DE FABRICACIÓN DE LA PLACA DE SUJECCIÓN DEL HUSILLO

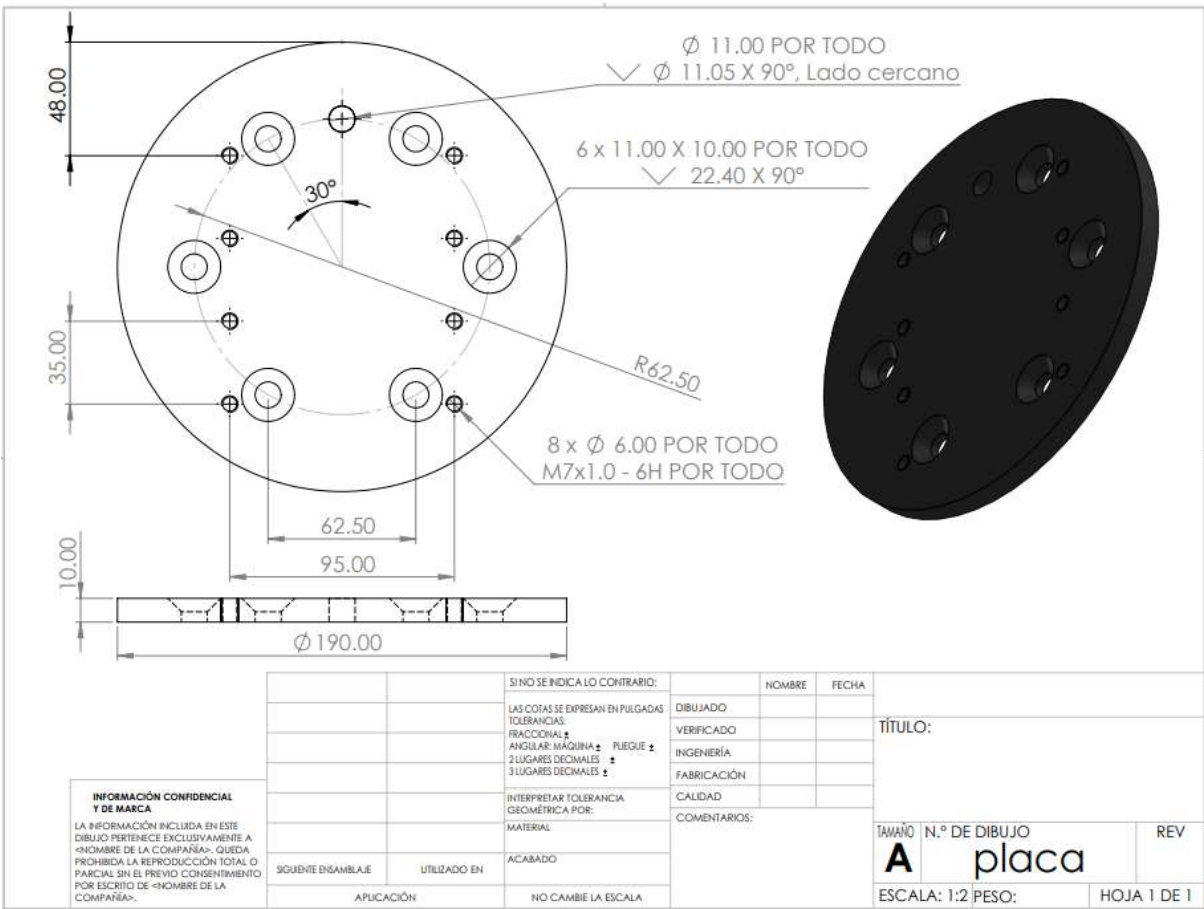


Figura A. 35 Plano de fabricación del herramental.

ANEXO 10 VERIFICACIÓN DE COORDENADAS DE PLANOS Y OBJETOS.

El proceso consiste en verificar que los sistemas de coordenadas sean acordes a los programados conforme a las posiciones físicas visualizadas en el *Teach Pendant*. Una vez definido el TCP y los planos de trabajo en el *Teach Pendant* se guardan las coordenadas y se transfieren al software *RobotDk* para crear las trayectorias, el proceso de verificación se realiza mediante la siguiente secuencia:

1. Crear una estación cargando los datos del *TCP* y las coordenadas de los planos de trabajo.



Figura A. 36 Configuración de los datos “Tool y Wobj”.

2. Crear un programa de trayectoria de aproximación al plano de trabajo (*wobj3*), con punto de aproximación.

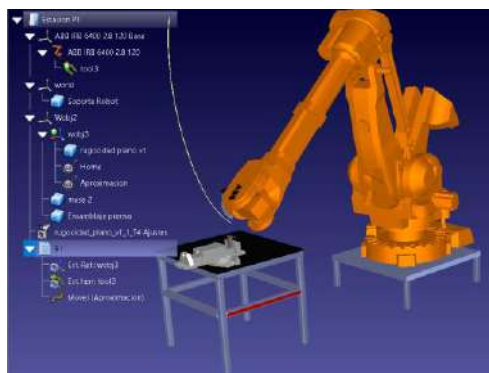


Figura A. 37 Trayectoria de aproximación.

3. Generar el programa y guardarlo en una memoria *USB*.

```

000
VERSIONS
LANGUAGE: ENGLISH
000
MODULE: BL

{ *****
{ Define customized variables here
{ ...
{ La siguiente seccion lee datos del TCP y UDP directo de los comandos generados en el Tshark device
{ Tool variables:
PDBS Tooldata tool1 to [0000,[-32,81,8,2,26,82,00],[0,50000000,0,50000000,0,50000000,0,50000000],[1],[0,0,0,0],[1,0,0,0],[2,2,2]]

{ Reference variables:
PDBS subdata sub1 to [FA00, TRUE, "",[2002,800,200,500,00,00],[0,000750,0,0000000,0,0000000,0,0211],[[0,0,0],[1,0,0,0]]]

MOD Main()
GetP3 Off;
GetP4 Off;
{ Source generated by WinBox v3.3.6 for AMD DRG 600 2.0 100 to 14/0/2250 151512
{ Using internal libraries.
WinBox[["15,110660,35,97168,38,32020,0,-38060,181,80000,0,572400],[38,00,0f,00,3f,00,3f,00,3f,00],[000,0,tool1,\\WinBox\\winBox]]
*****
}
ENDPROC

```

Figura A. 38 Ejemplo de generación de programa.

4. Cargar el programa al controlador y ejecutar el programa en la ruta *Test>Start*.

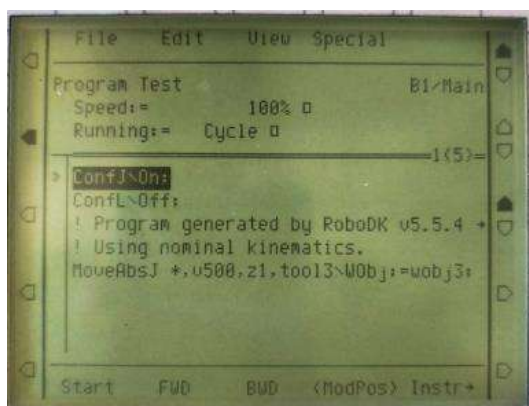


Figura A. 39 Ventana de programas para ejecución de secuencia.

5. Posicionar el *TCP* de la herramienta en el origen del plano de trabajo ("*User y Object*").



Figura A. 40 Posicionamiento del TCP en el origen programado.

En el *Teach Pendant* en la ventana del *jogging* con tipo de movimiento *Linear* se visualizan las coordenadas del TCP respecto a la base del robot (*Wobj0*) o plano de trabajo (*Wobj3*), de esta manera se pueden reajustar los sistemas coordenados.

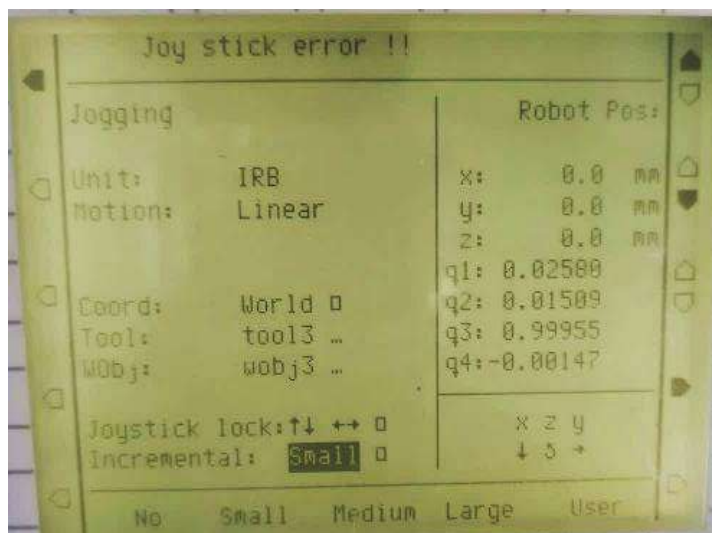


Figura A. 41 Verificación de posicionamiento en el origen.

En caso de que las coordenadas no coincidan se puede cambiar el sistema de referencia al plano de origen de la base del robot ahí se muestran las coordenadas físicas. Posteriormente se repete el proceso de verificación.

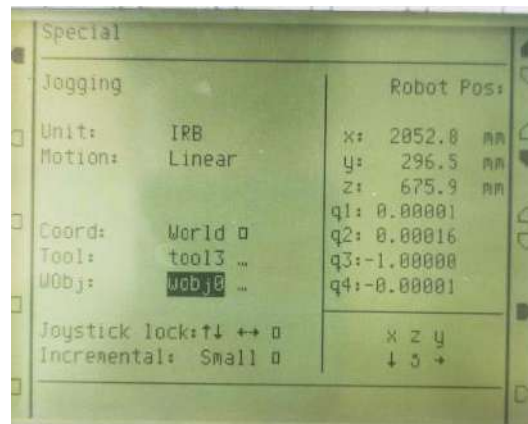


Figura A. 42 Visualización de la posición actual del *TCP* respecto a la base.

ANEXO 11 FUNCIONES MODBUS

Funciones de registro *Modbus*: se especifican las funciones del protocolo *Modbus*.

Parameter	Function	Specification
01H	Read parameter address status	Read from 1 to 32
03H	Read holding register	Read from 1~20. Read all parameters, communication-specific variables
04H	Read input register	Read from 1~20.
05H	Write single parameter address	Write data FF00H, parameter address ON; write 0000H, parameter address OFF
06H	Write single holding register	All parameters and communication-specific variables can be rewritten
0FH	Write multiple parameter address	Write from 1~32, parameter address start from 0048H
10H	Write multiple holding registers	Rewritten from 1~20. All parameters and communication-specific variables can be rewritten

Figura A. 43 Funciones del protocolo Modbus.

Registros específicos de comunicación.

Definition	Address	Specification
Internal parameters	0000H~00FFH	Inverter parameter address, F000~F255
Communication	0200H	Main control bit, BIT0-BIT7 mapping parameter address 0048H~004FH, BIT8 virtual input terminal enable
Specific Instruction	0201H	Given frequency, given frequency address when F002=2
Variable	0202H	Virtual input terminal1, BIT0-BIT15 corresponding digital input terminal function 01-16. Digital input terminal function 1-8 already exists in the main control bit, the modification here is invalid.
	0203H	Virtual input terminal 2, BIT0-BIT15 corresponding digital input terminal function 17-32
	0204H	EDO, Communication control digital output terminal, BIT3 corresponding (FA, FB, FC) relay terminal
	0205H	EAO, Communication control analog output terminal AO
	0206H~020FH	Reverse

Figura A. 44 Registros específicos del modelo del variador de frecuencia.

ANEXO 12 PROGRAMA DE COMUNICACIÓN PARA ARDUINO MEGA

Para realizar la comunicación serial “Controlador-Arduino-Variador” se necesita contar el módulo RS232- DB9 hembra para establecer comunicación serial con el controlador del robot, al igual el módulo TTL-RS485 para enviar los datos al variador desde la placa Arduino, el módulo optoacoplador PC 817 con dos canales para reducir el voltaje de las señales digitales, además de utilizar la librería “SimpleModbusMaster” para realizar la comunicación ModbusRTU. El accionamiento del variador se llevó a cabo de dos maneras:

- Señales digitales: El controlador envía una señal digital de 24V para el accionamiento, en el programa las variables fueron declaradas en el pin siete y ocho; es necesario reducir el voltaje de la señal mediante un módulo optoacoplador PC 817 con dos canales, el cual reduce el voltaje a 5V.
- Accionamiento manual: Se realiza mediante dos pulsadores que envían la señal previamente declarada en el programa en los pines cuatro y cinco.

El programa en la placa Arduino se describe a continuación:

```
#include <SimpleModbusMaster.h>
```

```
/* La librería SimpleModbusMaster permite la comunicación Modbus/RTU vía RS232 o RS485. Para comunicarse con un esclavo, lo primero que hay que hacer es crear un paquete o bloque de datos. * Es decir, una estructura de datos que contenga información relevante asociada a ese paquete de datos y que será escaneada o analizada de modo secuencial para la correcta ejecución de esa comunicación. Cada paquete contendrá la siguiente información: el identificador del paquete, la dirección del esclavo con el que comunicarse, la función MODBUS a realizar (leer o escribir bits o coils, leer o escribir registros, etc.), la dirección del registro a partir del cual se van a escribir o leer los datos, el número de datos a escribir o leer, y finalmente la estructura de datos enteros o tipo int a escribir o leer. Podría ser un array de enteros como en este caso... */
```

```
#define baud 9600
```

```
#define timeout 1000
```

```
#define polling 200
```

```
#define retry_count 10
```

```
#define TxEnablePin 11
```

```
#define RS485_TX 1 // Pin para TX del módulo RS485.
```

```
#define RS485_RX 0 // Pin para RX del módulo RS485.
```

```
String frase = ""; // se define la variable frase para recibir el dato de velocidad tipo string.
```

```
int rpm = 0; // rpm será el valor numérico de la "frase".
```

```
bool marcha, paro, d1, d2; // variables de arranque y paro del variador.
```

```
enum // se generan los paquetes necesarios utilizando la instrucción enum.
```

```
{
```

```
    PACKET1, // En estos paquetes se insertarán no sólo el dato a mandar sino otros necesarios para la comunicación.
```

```
    PACKET2, // En este paquete se insertarán los datos para la escritura, por ejemplo.
```

```
    TOTAL_NO_OF_PACKETS // Con esta instrucción se actualiza sólo el número de paquetes a utilizar.
```

```

};

Packet packets[TOTAL_NO_OF_PACKETS]; // Se crea un array de paquetes de datos para la comunicación.

packetPointer packet1 = &packets[PACKET1]; // Crea un puntero a PACKET1 donde almacenar los registros relativos a la lectura de datos.

packetPointer packet2 = &packets[PACKET2]; // Crea un puntero a PACKET1 donde almacenar los registros relativos a la escritura de datos.

unsigned int writeRegs[2]; // Se crea un array de 2 enteros donde escribir los datos a mandar. Un entero para la CONTROL WORD y otro para la velocidad.

unsigned int readRegs[2]; // Se crea un array de 2 enteros donde se podrían guardar los datos leídos.

void setup()
{
    Serial.begin(9600);

    Serial1.begin(9600); // se habilita el serial1 Pines 18 19 rs232

    pinMode(RS485_TX, OUTPUT); // se definen los pines 0 e 1 como salida rs485

    pinMode(RS485_RX, INPUT);

    pinMode(TxEnablePin, OUTPUT); // define el pin 11 para habilitar en envío y recepción del módulo rs485 (DE_RE.

    pinMode(4, INPUT); // se definen los pines como entradas

    pinMode(5, INPUT);

    pinMode(7, INPUT_PULLUP);

    pinMode(8, INPUT_PULLUP);

    modbus_construct(packet2, 1, PRESET_MULTIPLE_REGISTERS, 512, 2, writeRegs); // Con esta instrucción se configura o elabora el paquete de datos a enviar.

    // En este caso, se van a escribir 2 registros empezando en la dirección 512 sería la dirección 0200 hexadecimal que pertenece a los parámetros de registro del VDF. // Son 2 bytes/reg. Recordar que las variables unsigned int ocupan 2 bytes.

    modbus_configure(&Serial, baud, SERIAL_8N1, timeout, polling, retry_count, RS485_TX, packets, TOTAL_NO_OF_PACKETS); /*Se configura la comunicación. * Sus parámetros indican: El puerto serie sobre el que vamos a trabajar. La velocidad de transmisión. El formato de los bytes a transferir (SERIAL_8N1: 1 start bit, 8 data bits, 1 stop bits). El tiempo permitido para que el esclavo responda (entre 1000 y 5000mS). El tiempo que se da. Al esclavo tras finalizar la comunicación para que entre en su estado de reposo (entre 100 y 200mS). Número de intentos o solicitudes de comunicación que realiza el maestro cuando falla esta, antes de anular esa comunicación o dar error. Pin empleado para conmutar la recepción/envío de paquetes. El pin 11-rs485_tx activa o desactiva el envío o recepción*/
}

void loop()
{
    marcha = digitalRead(4); // Marcha accionado por botón.

    paro = digitalRead(5); // Paro accionado por botón.

    d1 = digitalRead(7); // Marcha automática por señal digital del robot.

    d2 = digitalRead(8); // Paro automático por señal digital del robot.

    // Se maneja la trama Modbus RS-232

    if (Serial1.available())
    {
        String frase = Serial1.readStringUntil("\n"); // se lee el valor de la velocidad en forma de string.

        rpm = frase.toInt(); // se transforma el string a valor numérico.

        float velocidad = rpm / 60.0 * 10.0; // se escala el valor del rpm a HZ por diez para poder leerlo.
    }
}

```

```

delay(50);

/* aquí podemos revisar si estamos recibiendo datos vía R232 al imprimirlos en el monitor serial.
Serial.print("Velocidad en HZ: ");
Serial.println(velocidad);
Es opcional para verificar errores.
*/

if (velocidad > 0 && velocidad < 4000) // se decide si la velocidad está dentro del rango.
{
    digitalWrite(TxEnablePin, HIGH); // Se habilita la transmisión RS-485,(DE-RE) del RS485.
    digitalWrite(RS485_TX, HIGH); // Habilitar la transmisión RS-485 del pin Tx del Arduino
    modbus_update(); // Actualiza la comunicación Modbus cíclicamente.
    writeRegs[1] = velocidad; // escribe en el registro el valor de velocidad .
    writeRegs[0] = 4; // escribe el número 4 para el giro sentido horario, 3 si es en sentido antihorario.
    delay(250);
    digitalWrite(TxEnablePin, LOW);
    digitalWrite(RS485_TX, LOW); // Deshabilitar la transmisión RS-485.
}
}

// Manejar la trama Modbus RS-485.
if (marcha == HIGH || d1 == LOW) // este apartado se creó para enviar la trama de datos de manera directa.
{
    digitalWrite(TxEnablePin, HIGH);
    digitalWrite(RS485_TX, HIGH);
    byte frame[] = {
        0x01, // Dirección del esclavo (VDF).
        0x05, // Código de función (0x05 para escribir un solo registro).
        0x00, 0x48, // high y low bit parámetros de registro.
        0xFF, 0x00, // high y low bit datos de escritura.
        0X0C,
        0X2C,
    };
    Serial.write(frame, sizeof(frame));
    delay(250);
    digitalWrite(TxEnablePin, LOW);
    digitalWrite(RS485_TX, LOW); // Deshabilitar la transmisión RS-485
}

if (paro == HIGH || d2 == LOW)
{

```

```

digitalWrite(TxEnablePin, HIGH);
digitalWrite(RS485_TX, HIGH);

byte frame[] = {
    0x01, // Dirección del esclavo (VDF)
    0x05, // Código de función (0x05 para escribir un solo registro).
    0x00, 0x4B, // high y low bit parámetros de registro.
    0xFF, 0x00, // high y low bit datos de escritura.
    0XFC,
    0X2C,
};

Serial.write(frame, sizeof(frame));
delay(250);
digitalWrite(TxEnablePin, LOW);
digitalWrite(RS485_TX, LOW); // Deshabilitar la transmisión RS-485.
}

frase = ""; // limpia el string recibido para iniciar de nuevo.
Serial1.flush();
}

```

En la Figura A. 45 se muestra conceptualmente el arreglo para la comunicación entre la placa Arduino con los módulos RS232, RS485, y el optoacoplador PC 817.

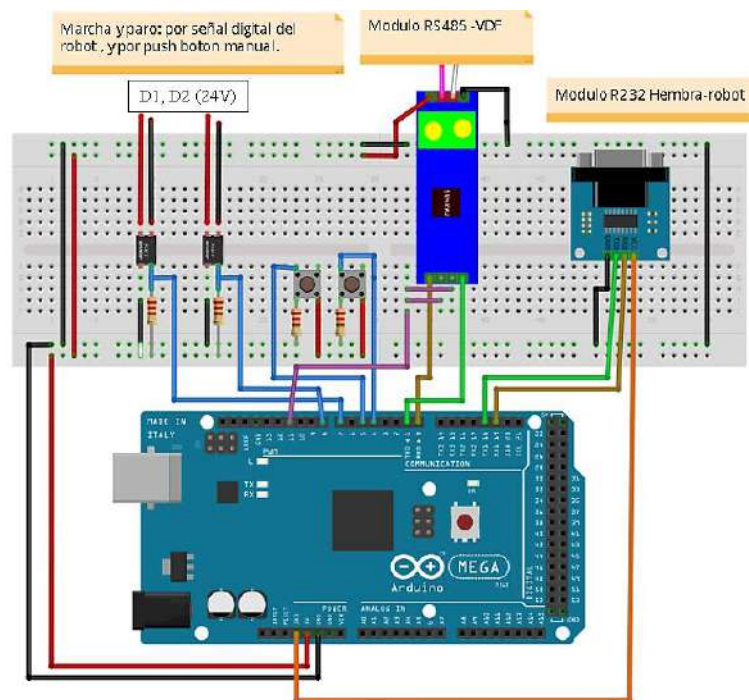


Figura A. 45 Componentes del arreglo para la comunicación serial.

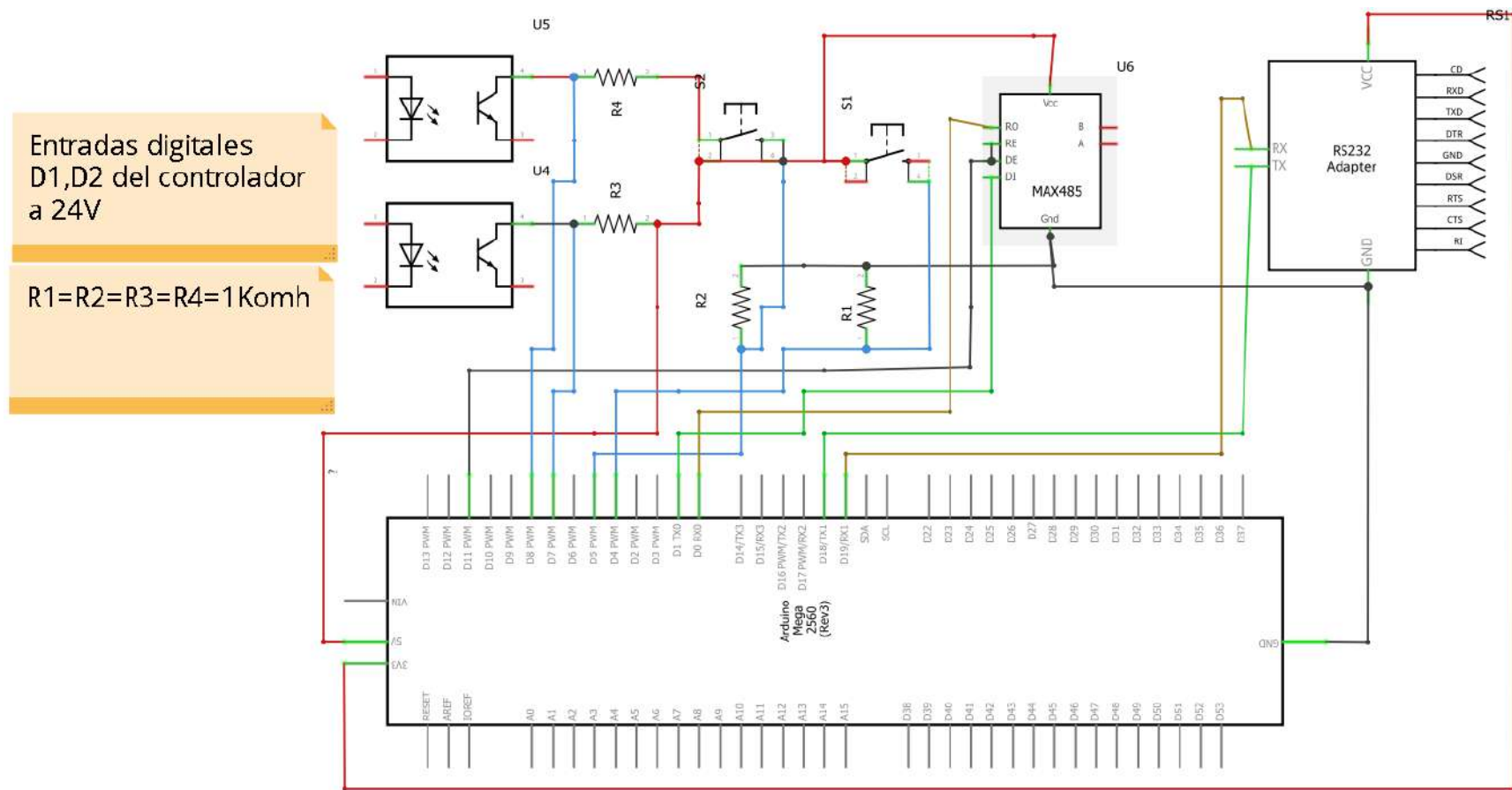


Figura A. 46 Diagrama del arreglo para la comunicación serial.

ANEXO 13 PROGRAMA DE MAQUINADO

Generación del programa de maquinado

- 1. Si la cantidad de líneas de programación excede la cantidad de 7000 líneas es necesario subdividir el programa en subprogramas, debido a que la comunicación del robot se realiza por medio de la memoria *USB a floppy*. En caso contrario, se dejan los recorridos en un programa principal.



Figura A. 47 Generación de programas.

- 2. Generar los programas y ajustarlos manteniendo el *"TCP"* y el *"Wobj"*. RobotDk genera varios programas con capacidad máxima de 1.44MB seleccionando una carpeta en el escritorio.

robot dk > Bochito > Desbaste			
Nombre	Fecha de modific...	Tipo	Tamaño
Bochito_v3_1_T4	18/06/2024 01:30 ...	Archivo PRG	2 KB
Bochito_v3_1_T4_0	18/06/2024 01:30 ...	Archivo PRG	1,207 KB
Bochito_v3_1_T4_1	18/06/2024 01:30 ...	Archivo PRG	1,218 KB
Bochito_v3_1_T4_2	18/06/2024 01:30 ...	Archivo PRG	536 KB

Figura A. 48 Programas generados por RobotDk.

- 3. Editar cada programa para guardarlo individualmente, la estructura general de un programa queda de la siguiente manera:

```

1  %%%
2  VERSION:1
3  LANGUAGE:ENGLISH
4  %%%
5  MODULE Programa1
6  ! Define customized variables here
7  VAR iodev rpm;
8  ! Tool variables:
9  PERS tooldata tool3 := [1000,[-152.008,2.792,48.474],[0.50880434,-0.50880434,-0.49103782,0.49103782]],[1,[0,0,20],[1,0,0,0],2,2,2]];
10 ! Reference variables:
11 PERS wobjdata Wobj3 := [FALSE, TRUE, "", [[2183.050,302.915,700.700],[0.01252249,-0.00012280,-0.00064736,-0.99992137]],[[0,0,0],[1,0,0,0]]];
12 PROC Prg_Tmp1()
13   ConfJ \On;
14   ConfL \Off;
15   ! inicio de trayectoria
16   .....
17   .....
18   ! final de trayectoria
19 ENDPROC
20 PROC Main()
21   ConfJ \On;
22   ConfL \Off;
23   Prg_Tmp1; ! llama al subprograma
24   EXIT;
25 ENDPROC
26 ENDMODULE

```

Figura A. 49 Visualización de código del programa.

4. Guardar cada subprograma en formato ABB (extensión. PRG) en un “floppey”.

Serial numb...	Capability	Used	Renew time
000	1.39 MB	2%	2024-05-14 14-21-...
001	1.39 MB	85%	2024-06-28 14-09-...
002	1.39 MB	84%	2024-06-28 14-09-...
003	1.39 MB	12%	2024-06-28 14-10-...
004	1.39 MB	78%	2024-06-06 11-59-...

Figura A. 50 Guardar en unidad floppy.

A continuación, se muestra un ejemplo del programa de maquinado de manera automática enviando el valor de la velocidad de husillo programada hacia el Arduino:

```
%%%
```

```
VERSION:1
```

```
LANGUAGE:ENGLISH
```

```
%%%
```

```
MODULE Programa1
```

```
! Define customized variables here
```

```
VAR iodev rpm;
```

```
! Tool variables:
```

```
PERS tooldata tool3 := [TRUE,[-152.008,2.792,48.474],[0.50880434,-0.50880434,-0.49103782,0.49103782]],[1,[0,0,20],[1,0,0,0],2,2,2]];

```

```
! Reference variables:
```

```
PERS wobjdata Wobj3 := [FALSE, TRUE, "", [[2183.050,302.915,700.700],[0.01252249,-0.00012280,-0.00064736,-0.99992137]],[[0,0,0],[1,0,0,0]]];

```

```
PROC Prg_Tmp1()
```

```
ConfJ \On;
```

```
ConfL \Off;
```

```
!SetRPM(4500.0);
```

```
Open "sio1:","rpm\Write; ! Habilita la comunicación serial
```

```
Write rpm,"4500";
```

```

Close rpm;

Open "sio1:","rpm\Write;

Write rpm,"4500";

Close rpm; ! El valor se envia dos veces para asegurarse

SetDO S1,1; ! La salida S1 activa el husillo

WaitTime 2;

SetDO S1,0;

! Inicio de trayectoria

! La velocidad de posicionamiento inicial se programa entre 20-30mm/s

! Las velocidades de posicionamiento rápido dentro de la trayectoria se programan a 5mm/s

MoveAbsJ [[15.026400,29.542100,36.646500,-80.957300,102.051000,50.383900],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v20,z1,tool3
\WObj:=Wobj3;

!! Los movimientos de corte respetan la velocidad programada en Fusion 360, entre 40-60mm/min

MoveL[[3.089,34.236,2.196],[0.00000000,0.99992159,0.01252258,0.00000000],[0,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[1.00
,500,5000,1000],z1,tool3 \WObj:=Wobj3;

MoveL[[7.801,0.856,10.738],[0.00000000,0.99992159,0.01252258,0.00000000],[0,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[1.00
,500,5000,100],z1,tool3 \WObj:=Wobj3;

MoveL[[7.878,0.116,10.738],[0.00000000,0.99992159,0.01252258,0.00000000],[0,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[1.00
,500,5000,100],z1,tool3 \WObj:=Wobj3;

MoveL[[7.891,0.628,10.738],[0.00000000,0.99992159,0.01252258,0.00000000],[0,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[1.00
,500,5000,1000],z1,tool3 \WObj:=Wobj3;

! final de trayectoria

ENDPROC

PROC Main()

ConfJ \On;

ConfL \Off;

Prg_Tmp1; ! LLama al subprograma

SetDO S2,1; ! Al finalizar apaga al husillo

WaitTime 2;

SetDO S2,0;

EXIT; ! Sale del ciclo y espera cargar un nuevo programa

ENDPROC

ENDMODULE

```