



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

**DISEÑO E IMPLEMENTACIÓN DE UN
DISPOSITIVO DE CONTROL NUMÉRICO
COMPUTARIZADO MULTIFUNCIONAL
ARTESANAL**

Br. José V. Muñoz M.

Mérida, septiembre, 2022



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO DE
CONTROL NUMÉRICO COMPUTARIZADO
MULTIFUNCIONAL ARTESANAL**

Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero
Electricista

Br. José V. Muñoz M.

Tutor: MSc. Luís Ramón Araujo Rangel

Mérida, septiembre, 2022

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO DE CONTROL
NUMÉRICO COMPUTARIZADO MULTIFUNCIONAL ARTESANAL**

Br. José V. Muñoz M.

www.bdigital.ula.ve

Trabajo de Grado, presentado en cumplimiento parcial de los requisitos exigidos para optar al título de Ingeniero Electricista, aprobado en nombre de la Universidad de Los Andes por el siguiente Jurado.

Prof. Francisco Javier Vilorio M.

Prof. Alex Joel Caliz

Prof. Luís Ramón Araujo Rangel

Autor: **José V. Muñoz M. Diseño e implementación de un dispositivo de control numérico computarizado multifuncional artesanal.** Universidad de Los Andes. Tutor: Luis Ramón Araujo Rangel. Septiembre, 2022.

RESUMEN

Este trabajo está enfocado en el diseño e implementación de un sistema de control numérico computarizado (CNC), que proporciona la integración de diferentes componentes y tecnologías de automatización que le permite al usuario el control y monitorización de una máquina CNC artesanal de forma fácil y segura, a través de una interfaz gráfica intuitiva. Dicha aplicación interpreta la información suministrada por el usuario en lenguaje de programación con formato de código G y la transforma en coordenadas vectoriales (X, Y, Z), donde un microcontrolador se encarga de interpretarlas, para así accionar el mecanismo de funcionamiento, donde el movimiento giratorio de los motores de paso es convertido en el movimiento rectilíneo de los ejes de desplazamiento, para que una herramienta se posicione vectorialmente sobre una pieza o material de trabajo y cumpla la misión de taladrar, cortar, roscar, fresar, o dibujar de forma totalmente automática. Además, realizar un estudio técnico-económico de las piezas eléctricas y mecánicas, con el fin de ensamblar un sistema de precisión milimétrica construido con materiales adquiridos en el mercado local, que, a su vez, trabaje bajo una plataforma de software basado en microchip, con costo reducido y comodidad multifuncional.

Descriptores: CNC, máquinas de control numérico, microcontroladores, software, interfaz.

ÍNDICE GENERAL

APROBACIÓN	ii
RESUMEN.....	iii
INTRODUCCIÓN	1-2

Capítulo 1: Planteamiento del problema

1.1. Evaluación del problema	3
1.2. Justificación	3
1.3. Objetivos.....	4
1.3.1. Objetivo General.....	4
1.3.2. Objetivos Específicos.....	5
1.4. Metodología.....	5
1.5. Plan de actividades	5-6
1.6. Cronograma de actividades	6
1.7. Tipo de financiamiento.....	6

Capítulo 2: Marco teórico

2.1. Antecedentes.....	7
2.2. Bases Teóricas	8
2.2.1. Control Numérico Computarizado	8
2.2.2. Código G.....	11
2.2.3. Inkscape	11
2.2.4. Drivers A4988.....	13
2.2.5. PIC18F4550	15
2.2.6. MPLAB X IDE	15
2.2.7. MPLAB XC8	16
2.2.7.1.Creación de un programa en MPLAB X IDE con XC8	17

2.2.7.2. Código fuente	17
2.2.7.3. Palabra de configuración	18
2.2.8. Blender	20
2.2.9. Unity	21
2.2.10. Proteus 8.11	22
2.2.11. MikroProg Suite for Pic	23
2.2.12. Virtual Serial Ports Emulator	24
2.2.13. Hércules Setup	24
2.2.14. Microsoft Visual Studio IDE	25
2.3. Funcionamiento general	26

Capítulo 3: Diseño

3.1. Consideraciones de diseño	28
3.2. Mecanismo y dimensiones	29
3.2.1. Ejes y rodamientos	29
3.2.2. Tornillos de arrastre mecanizado	32
3.2.3. Motores	35
3.3. Estructura	38
3.3.1. Diseño principal	39
3.3.2. La base de la máquina	40
3.3.3. La torre	42
3.3.4. La caja de soporte de la herramienta	43
3.4. Circuito de control	45
3.5. Estudio técnico-económico de los elementos de una CNC	51

Capítulo 4: Software

4.1. Control de movimiento	56
4.2. Bloque de comando	57
4.2.1. Acción	58
4.2.2. Instrucción	58
4.3. Controlador CNC	58

4.3.1. Función Principal	59
4.3.2. Función para procesar la línea	61
4.3.3. Función para dibujar la línea	62
4.3.4. Función para dibujar arco	65
4.3.4.1. Consideración 1	70
4.3.4.2. Consideración 2	71
4.3.5. Función para mover los motores y reposo del motor.....	72
4.4. Conexión electrónica.....	73
4.5. Velocidad de control	74
4.6. Interfaz gráfica.....	75
4.6.1. Paso 1: Ejecutar la aplicación CNC	84
4.6.2. Paso 2: Abrir panel de ejecución	85
4.6.3. Paso 3: Seleccionar el archivo deseado.....	86
4.6.4. Paso 4: Comprobar en la pre-visualización.....	87
4.6.5. Paso 5: Abrir el menú de conexiones.....	88
4.6.6. Paso 6: Volver al panel de Ejecución y seleccionar iniciar.....	89
4.6.7. Paso 7: Apremiar y vigilar la ejecución del sistema	91
4.6.8. Paso 8: Salir del programa.....	92
4.7. Pruebas de precisión.....	93
CONCLUSIONES Y RECOMENDACIONES	
Conclusiones.....	96
Recomendaciones	97
Bibliografías	98

ÍNDICE DE FIGURAS

Figura 2.1: CNC 3018 Pro.....	10
Figura 2.2: Código de trayectorias y sistema de coordenadas desarrollado en Inkscape	11
Figura 2.3: Esquema de conexión del driver A4988	14
Figura 2.4: Diagrama funcional de bloques del driver A4988	14
Figura 2.5: Esquema de conexión del PIC18F4550	15
Figura 2.6: Programa escrito en MPLAB X IDE y compilado en MPLAB XC8 Compiler.....	16
Figura 2.7: Código base de un proyecto.....	17
Figura 2.8: Configuración de los bits para un PIC18.....	19
Figura 2.9: Palabra de configuración de los bits en código	19
Figura 2.10: Estructura del equipo modelado en Blender.....	20
Figura 2.11: Diseño de la interfaz gráfica a través de la plataforma Unity.....	21
Figura 2.12: Esquema electrónico del sistema mediante el programa Proteus	22
Figura 2.13: Mikro Prog Suite for PIC compilando programa al PIC18F4550	23
Figura 2.14: VSPE emparejando puertos seriales para las pruebas de software	24
Figura 2.15: Hércules como terminal virtual para la comunicación serial	25
Figura 2.16: Programa escrito sobre Microsoft Visual Studio IDE.....	26
Figura 2.17: Diagrama de funcionamiento de una CNC.....	27
Figura 3.1: Representación del desplazamiento de los ejes de una CNC	29
Figura 3.2: Ejes ópticos de acero	30
Figura 3.3: Soporte de rieles de aluminio	31
Figura 3.4: Bujes deslizantes.....	31
Figura 3.5: Tipos de tornillos de arrastre en cuanto al número de hélices	32
Figura 3.6: Tornillo de arrastre con sus respectivas tuercas de fricción	33
Figura 3.7: Acoplamiento de Aluminio	34
Figura 3.8: Rodamientos lineales	35
Figura 3.9: Dimensiones de soporte y motor paso a paso NEMA 17 bipolar.....	37

Figura 3.10: Dimensiones del diseño en milímetros modelado en 3D	39
Figura 3.11: Ensamblaje de la base de la máquina modelado en 3D.....	40
Figura 3.12: Cama de la base por donde deslizan los bujes del eje Y	41
Figura 3.13: Cama y base ensambladas modelado en 3D	41
Figura 3.14: Torre del diseño modelado en 3D	42
Figura 3.15: Soporte del eje X ubicado en la caja del eje Z modelada en 3D.....	43
Figura 3.16: Torre ensamblada a la base modelada en 3D.....	43
Figura 3.17: Caja de soporte para la herramienta de la CNC modelado en 3D	44
Figura 3.18: Diseño final de la máquina	44
Figura 3.19: Representación visual de la fuente de poder	45
Figura 3.20: Calibración del voltaje de referencia de los drivers	46
Figura 3.21: Esquema de conexiones y modelo de una Shield CNC	48
Figura 3.22: Placa PCB perforada de dimensiones 7x9 cm	48
Figura 3.23: Base para circuito integrado de 40 pines.....	48
Figura 3.24: Sensores finales de carrera	49
Figura 3.25: Módulo de comunicación serial FT232.....	49
Figura 3.26: Módulo relé.....	50
Figura 3.27: Dremel model 275.....	50
Figura 3.28: Ubicación de Tornillos Mérida	51
Figura 3.29: Ubicación de FERRETODO La Milagrosa	52
Figura 3.30: Ubicación de Ferrelectrico Caracas, Mérida	52
Figura 3.31: Ubicación de Electrónica Nevada, Mérida	53
Figura 4.1: Formato de frases conformadas en un comando.....	57
Figura 4.2: Diagrama de flujo de la función principal.....	59
Figura 4.3: Configuración de datos para la comunicación asíncrona	60
Figura 4.4: Diagrama de flujo de la función para procesar línea	61
Figura 4.5: Diagrama de flujo de la función para dibujar la línea	63
Figura 4.6: Trayectoria de una línea descrita por una CNC	64
Figura 4.7: Diagrama de flujo de la función para dibujar arco	65
Figura 4.8: Representación gráfica del proceso de interpolación circular	66

Figura 4.9: Triángulo formado por la cuerda y el radio para la interpolación circular.....	67
Figura 4.10: Ángulo θ que relaciona el punto inicial con el centro de la circunferencia.....	68
Figura 4.11: Triángulo rectángulo formado por las distancias “i” y “j”	69
Figura 4.12: Diagramas con las consideraciones con respecto al punto inicial y el centro	70
Figura 4.13: Diagramas con las consideraciones de los ángulos reflejo.....	71
Figura 4.14: Diagrama de conexión final del PIC18F4550	73
Figura 4.15: Modo de decaimiento para incrementos de $\frac{1}{4}$ pasos.....	75
Figura 4.16: Logo de Val CNC	76
Figura 4.17: Pantalla de carga de la aplicación	76
Figura 4.18: Menú de animación de la plataforma Unity	77
Figura 4.19: Características de los botones seleccionables.....	81
Figura 4.20: Función para abrir un nuevo menú	81
Figura 4.21: Puertos de comunicación disponibles a través de un botón desplegable	82
Figura 4.22: Funciones relacionadas a la comunicación serial	82
Figura 4.23: Barra de tareas de la interfaz de usuario de ValCNC.....	83
Figura 4.24: Texto descriptivo al presionar el menú información.....	83
Figura 4.25: Abrir o ejecutar la aplicación.....	84
Figura 4.26: Pantallas introductorias de la aplicación	84
Figura 4.27: Ubicación del panel de ejecución en la aplicación	85
Figura 4.28: Ubicación del botón de abrir archivos en el panel de ejecución.....	85
Figura 4.29: Seleccionar el archivo que se desea ejecutar	86
Figura 4.30: Pre-visualización de un archivo de código G	87
Figura 4.31: Ubicación del menú para establecer conexiones	88
Figura 4.32: Menú desplegable de los puertos de comunicación	88
Figura 4.33: Administrador de dispositivos del panel de control.....	89
Figura 4.34: Botón para iniciar la conexión con el puerto de comunicación.....	89
Figura 4.35: Sistema iniciado a nivel de simulación	90
Figura 4.36: Lienzo en blanco e iniciado el sistema.....	90
Figura 4.37: Apreciación de los mecanismos a nivel de simulación y físico.....	91
Figura 4.38: Pausar el mecanismo a nivel de simulación y físico	91

Figura 4.39: Boceto final obtenido	92
Figura 4.40: Ícono para salir de la aplicación.....	92
Figura 4.41: Animación final de la aplicación	82
Figura 4.42: Pruebas de precisión con un vernier digital.....	95

www.bdigital.ula.ve

ÍNDICE DE TABLAS

Tabla 1: Características de los motores de paso NEMA.....	36
Tabla 2: Características eléctricas de los motores de paso NEMA 17HD48002H-22B.....	37
Tabla 3: Presupuesto de piezas mecánicas de la estructura	53
Tabla 4: Presupuesto de piezas electrónicas de la estructura	54
Tabla 5: Presupuesto de CNC prefabricada	54
Tabla 6: Presupuesto seleccionado para este proyecto	55
Tabla 7: Funciones de movimiento del código G.....	58
Tabla 8: Ejemplo ilustrativo de la función dibujar la línea	64
Tabla 9: Organigrama de conexión externa del PIC18F4550	74
Tabla 10: Íconos que definen la barra de tareas de la interfaz de usuario del equipo	78
Tabla 11: Submenú de la opción “Panel de ejecución”	79
Tabla 12: Submenú de la opción “Ajuste”	80
Tabla 13: Submenú de la opción “Palanca de mando”	80
Tabla 14: Cálculo de precisión del eje Z.....	93
Tabla 15: Cálculo de precisión del eje Y	94
Tabla 16: Cálculo de precisión del eje X	94

INTRODUCCIÓN

A lo largo historia los avances tecnológicos han ayudado al ser humano a facilitar las labores del día a día, hasta el punto que no se aplique esfuerzo para realizarlas. En ese mismo orden de ideas, para que una tecnología pueda avanzar, es necesario saber su funcionamiento para apreciar sus limitaciones. Actualmente, los sistemas de manufactura son caracterizados por su complejidad, la participación de diferentes disciplinas de conocimiento y la demanda creciente de nuevas herramientas y técnicas para resolver problemas difíciles [1] .

En cuanto a los procesos de manufactura para la construcción de moldes, las máquinas de control numérico computarizado o CNC, han tenido un gran crecimiento desde los años sesentas y setentas con la invención del microchip y la implementación de este en los procesos de mecanizado. Verdaderamente, las tendencias de la automatización en la industria, exigen el empleo de equipos especiales para controlar y llevar a cabo los procesos de fabricación, donde prevalece que los automatismos sean eficientes y de mayor exactitud [1].

Así pues, este proyecto surge a partir de la problemática actual de generar un sistema versátil para el desarrollo y optimización de diferentes herramientas que requieren precisión, para desempeñar oficios bajo los cuales la mayoría de las empresas se ven afectadas.

Esta investigación está estructurada en 4 capítulos, que se componen de la siguiente manera:

- Capítulo I, se evalúa el problema, así como los objetivos que dirigen el proyecto. Se exponen la justificación y la metodología con un plan de actividades; su respectivo cronograma y se especifica el tipo de financiamiento.
- Capítulo II, se exponen las bases teóricas y los antecedentes que sustentan el funcionamiento general de un sistema CNC.
- Capítulo III, se puntualiza el mecanismo del sistema, como también, las mejores piezas electrónicas y mecánicas para construirla, además, se lleva a cabo un estudio técnico-

económico de cómo y dónde adquirir los elementos de la máquina en el mercado local que validen su construcción.

- Capítulo IV, se explica el funcionamiento del equipo enfocado en el software del mismo. Seguidamente, se detalla el manejo y uso de la interfaz de usuario, su monitorización y control. Este marco también se engloban las pruebas necesarias que corroboran la precisión milimétrica del sistema.

Finalmente, se detallan las conclusiones de la investigación, seguido de recomendaciones y referencias bibliográficas consultadas durante la ejecución del proyecto.

www.bdigital.ula.ve

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

El planteamiento del problema se conforma de los siguientes aspectos: evaluación del problema, justificación, objetivos, metodología, plan de actividades, cronograma de actividades y tipo de financiamiento.

1.1 EVALUACIÓN DEL PROBLEMA

Una máquina CNC es un artefacto electromecánico, cuyo sistema se usa esencialmente con el fin de crear o mejorar herramientas a nivel industrial a lo largo del mundo. En Venezuela actualmente adquirir una máquina de control numérico es una tarea totalmente complicada, ya que son de costo muy elevado y no existen proveedores o marcas reconocidas que suministren el mercado potencial de consumidores. Específicamente en la ciudad de Mérida, sus piezas son escasas de encontrar para su respectivo ensamblaje [2]; esto produce que la industria utilice herramientas inadecuadas para un oficio, limitando así su calidad y producción. Debido a esto, surge la necesidad de diseñar y construir un equipo mecanizado automático multifuncional que brinde facilidad, eficiencia y economía; capaz de facilitar la realización de un trabajo, a través de la mejora o elaboración de cualquier instrumento que sirva como una herramienta apropiada para su oficio.

1.2 JUSTIFICACIÓN

La automatización y control de procesos ofrece diferentes herramientas de desarrollo con el fin de proponer e implementar planes de mejora y dar solución a problemas específicos, tanto a nivel operativo, como a nivel administrativo [3].

La automatización es una evolución de la mecanización en la industria que utiliza dispositivos de alta capacidad de control, para lograr procesos de fabricación o producción eficientes [4]. Los cuales presentan beneficios en la automatización que con frecuencia

aparecen de forma intangible e inesperada, tales como: la mejora de la calidad, el aumento de las ventas, la mejora de las relaciones laborales, y la mejora de la imagen de la empresa [5]. En efecto, la elaboración de este proyecto responde a una necesidad en particular de los mercados altamente competitivos que se ven obligados a automatizar sus procesos de producción, minimizando costos, aumentando la productividad, haciendo más rápidos, precisos y de mejor calidad sus procesos; para ello, hacen uso de diferentes tipos de sistemas mecanizados que rigen el funcionamiento de una maquinaria. Entre ellos, el más destacado es el sistema CNC, ya que posee la flexibilidad de actualización para poder producir nuevos productos sin añadir o quitar nuevas piezas; mientras que, un sistema de producción en cadena debería cambiar absolutamente toda la maquinaria y partes que la conforman para conseguir fabricar un producto diferente [6].

En este orden de ideas, a pesar de sus ventajas, la aplicación de un control numérico actualmente no es una opción para las organizaciones, ya que la fabricación de estos modelos son de alta complejidad, tienen un elevado costo de producción y sumado a esto, la empresa argumenta que las fechas de entrega de los productos que elaboran son cada vez menores y les exigen mayor precisión y un alto control de calidad, lo cual es cada vez resulta más complicado [7].

Por esta razón, se justifica la implementación de una máquina que funcione a través de un sistema numérico computarizado, de bajo costo, que sea multifuncional y que logre solucionar necesidades cotidianas presentes en las empresas de forma totalmente automática, construido de manera artesanal a partir de materiales adquiridos en el mercado local.

1.3 OBJETIVOS

Los objetivos de la investigación se dividen según su escala en dos categorías, objetivos generales y específicos. Los cuales, se enuncian a continuación:

1.3.1 Objetivos generales

Diseñar y construir un dispositivo de control numérico computarizado multifuncional artesanal.

1.3.2 Objetivos específicos

- Investigar varios elementos de validación, con la posibilidad que puedan ser controlados por un microcontrolador, y medir la posibilidad de que puedan ser adquirido en el mercado local. Además de establecer los protocolos y algoritmos de comunicación.
- Construir un prototipo físico que alojará un microcontrolador PIC, y los demás componentes necesarios para la implementación, que permitan comunicarse con los elementos de validación.
- Desarrollar una aplicación que tenga la capacidad de llevar a cabo todas las tareas necesarias para el correcto funcionamiento del dispositivo.

1.4 METODOLOGÍA

La metodología para la realización de este trabajo se empieza con el estudio del circuito a partir de un diagrama de bloques, para así determinar los componentes que necesarios para construir un prototipo físico, desarrollando dicha tarjeta a nivel de software de simulación que verifiquen el funcionamiento de los componentes y la programación del microcontrolador. Seguido de esto, la construcción e implementación de un prototipo físico en donde se pueda tener toda la circuitería necesaria para todos y cada uno de los elementos que conforman el dispositivo. Finalmente unir todos los componentes, y realizar las pruebas necesarias para validar el prototipo.

1.5 PLAN DE ACTIVIDADES

El plan de actividades de la investigación se centra en ocho puntos fundamentales, que se organizan como el procedimiento más óptimo para la realización de los objetivos, los cuales son los siguientes:

- 1) Investigar varios elementos de validación para decidir cuáles son factibles para este trabajo, y adquirir los componentes en el mercado local que hagan falta para su implementación.

- 2) Diseñar un algoritmo básico para el software del microcontrolador.
- 3) Montar una simulación del circuito para empezar a programar el microcontrolador.
- 4) Realizar una propuesta del prototipo de hardware, que incluye todos los componentes, así como el desarrollo del chasis o sistema físico.
- 5) Escribir un Informe de Avance del Trabajo de Grado.
- 6) Construir el prototipo, lo que incluye el diseño del PCB, su construcción y pruebas de hardware.
- 7) Completar el desarrollo del software y las pruebas necesarias que validen su correcto funcionamiento
- 8) Escribir la Monografía del Trabajo Realizado.

1.6 CRONOGRAMA DE ACTIVIDADES

El cronograma de actividades ilustra la lista de todos los elementos terminales expuestos en el plan de actividades del proyecto, estimando el comienzo y final previsto para la investigación. Como se expone seguidamente:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1.5.1	X	X	X	X												
1.5.2		X	X	X	X	X										
1.5.3					X	X	X									
1.5.4					X	X	X	X								
1.5.5							X	X								
1.5.6									X	X	X	X	X	X		
1.5.7										X	X	X	X	X	X	
1.5.8												X	X	X	X	X

Se plantea entonces un cronograma de 16 semanas para cumplir con el plan de actividades a través de la metodología señalada.

1.7 TIPO DE FINANCIAMIENTO

- Autofinanciamiento.

CAPÍTULO II

MARCO TEÓRICO

El marco teórico se constituye del grupo de conceptos y constructos que representan un enfoque determinado del cual se deriva la explicación del problema planteado. Este capítulo engloba las siguientes vertientes: antecedentes, bases teóricas y funcionamiento general de una CNC.

2.1 ANTECEDENTES

García I y otros [8], en su tesis de Grado para obtener el título de Ingeniero Mecatrónico, titulada “Diseño e implementación de un control CNC para crear modelos y esculturas en tercera dimensión a partir de un diseño CAD”. Desarrollada en la Universidad de San Buenaventura, Bogotá, plantean como objetivo principal la instalación de un Control Numérico Computarizado, para automatizar una fresadora manual con el fin de fabricar modelos en tercera dimensión, con lo cual, emplearon una metodología en la que efectuaba un estudio comparativo de los diferentes sistemas actuales distribuidos y centralizados. Posteriormente realizaron un análisis de los componentes necesarios de toda aplicación del hardware y sus costos; así como la estructura física, lógica y el modelo de gestión o arquitectura a aplicar en el sistema de control computarizado. Finalmente, obtuvieron los datos necesarios para la implementación del sistema de control y establecieron las ventajas y desventajas existentes en instalar el sistema CNC en Colombia.

Sánchez [9], en su trabajo de grado para obtener el título de Ingeniero en Electrónica y Telecomunicaciones, titulado “Generador de código CNC para proceso de fresado, empleando visión computarizada” en la Universidad Autónoma del Estado de Hidalgo, en México, persigue implementar una técnica de ingeniería inversa en la industria de la manufactura que permita controlar y visualizar un prototipo a través de la extracción de piezas de otros

sistemas. Gracias a la obtención de diferentes elementos compatibles. Sánchez, consiguió satisfacer las expectativas planteadas, logrando una buena precisión en su diseño.

Jiménez [10], en su trabajo de grado titulado “Diseño, Programación, Simulación y fabricación en CNC con programas de CAD/CAM” para optar por el título de Ingeniero Mecánico Electricista en la Universidad Nacional Autónoma de México, implementó un sistema integrado CAD/CAM que vincula exitosamente con una fresadora FESC viable para la construcción de placas de circuitos impresos. Jiménez logró mejorar las prácticas de laboratorio de la universidad UAM ahorrando un tiempo considerable con la fabricación de distintos moldes con su diseño.

Canelones, [11], en su tesis de grado titulada “Diseño y Construcción de un equipo de control numérico (CNC) para el fresado de circuitos impresos (PCB), utilizando software libre” para la obtención del título de Ingeniero Electricista en la Universidad de los Andes, Venezuela, donde persigue el propósito de diseñar un hardware para el control de una máquina de fresado de forma rápida y eficiente, empleando el uso de un dispositivo Raspberry Pi para almacenar la información suministrada en forma de coordenadas vectoriales, y a partir de estas, tallar su respectivo circuito impreso. Finalmente, determinó que es factible la implementación de diferentes dispositivos que convirtieron a la máquina CNC en un equipo independiente y compacto.

2.2 BASES TEÓRICAS

Se presenta a continuación, una recopilación de consideraciones teóricas en las que se sustenta la investigación.

2.2.1 Control Numérico Computarizado

El control numérico por computadora o CNC es un sistema que permite el control de la posición de un elemento montado en el interior de una máquina-herramienta, mediante un software especialmente diseñado para ello [12].

Su funcionamiento está basado en el posicionamiento sobre los ejes X, Y, Z; gracias a lo cual, a una misma pieza se la puede taladrar, cortar, roscar, fresar, o desbastar en todos sus planos de forma totalmente automática [12].

El control numérico por computadora es un concepto moderno en las industrias de fabricación y producción. Sin embargo, el concepto de CNC (control numérico computarizado) se remonta a la idea básica de NC, o el control numérico que tuvo sus orígenes en la industria de la aviación [12].

Una CNC utiliza una técnica de mecanizado que se encontró en el siglo XVIII. De hecho, la primera ‘máquina de torneado con marco de metal’ fue inventada en 1751, fue la primera de una larga serie de máquinas con el objetivo de crear operaciones mecánicas más precisas que las que son posibles mediante técnicas artesanales. Esto marcó el comienzo de la industrialización [13].

La idea del control numérico comenzó cuando la automatización de máquinas incorporó conceptos específicos de lógica programable. Al principio, las primeras máquinas NC se construyeron en la década de 1940 y aparecieron máquinas un poco más avanzadas, en la década de 1950. Estas máquinas de fabricación se construyeron en base a las herramientas que se modifican con los motores para mover los controles de la máquina, estos controles siguieron puntos específicos que se introdujeron en la máquina en cinta perforada. Estos primeros mecanismos pronto se mejoraron con computadoras, tanto analógicas como digitales. La introducción de la tecnología informática en el concepto de control numérico condujo a lo que ahora se conoce como control numérico por computadora [13].

Fue hasta la Guerra Fría en 1949 que se abordó el desarrollo de la automatización. En ese momento, la compañía Parsons Works fue comisionada por la Marina de los EE. UU, para aumentar la productividad de su línea de producción de palas para helicópteros. John T. Parsons luego motorizó los ejes de las máquinas para hacer estas cuchillas. Trabajando con IBM (Corporación Internacional de Máquinas Comerciales), estudió las posibilidades de controlar estas máquinas por computadora. Este fue el punto de partida del método de mecanizado CNC [13].

En 1952, Richard Kegg, en colaboración con el MIT (Instituto Tecnológico de Massachusetts), crearon la primera fresadora CNC: la *Cincinnati Milacron Hydrotel*. Cinco años más tarde, en 1958, solicitó una patente para un “Aparato controlado por motor para máquina herramienta de posicionamiento”. Este fue el nacimiento comercial de esta tecnología [13].

Antes de los proyectos del MIT, Parsons Corporation en Traverse City, Míchigan. John Parsons, quien fundó la compañía, descubrió cómo calcular las coordenadas de la superficie aerodinámica en un multiplicador IBM 602^a (Calculadora electromecánica). Luego, introduce estos puntos de datos en un perforador de plantillas suizo. Hasta la fecha, esta se consideraba la primera máquina de control numérico real, ya que fabricaba productos (plantillas de palas de helicóptero, en este caso) al alimentar tarjetas perforadas en un sistema, y el sistema luego leía y producía las piezas en base a información pre-programada [13]. En perspectiva, la tecnología de control numérico se trasladó a las décadas de 1960 y 1970, donde presentaban una forma muy familiar a la de una máquina CNC que actual. Luego, la tecnología digital entró en juego y la automatización en los procesos de producción se volvió más eficiente que nunca. De hecho, debido a lo avanzado que son las computadoras hoy en día, es muy común encontrar máquinas CNC en todas las industrias exitosas [13].

En pocas palabras, el control numérico computarizado es el uso de una computadora para controlar y monitorear los movimientos de una herramienta. Entre esas herramientas, tanto estáticas como portátiles, destacan: fresadora, torno, rectificadora, máquina de corte por láser, por chorro de agua o por electroerosión, estampadora, prensa, brazo robotizado, etc. Las máquinas de gran porte cuentan con una computadora dedicada que forma parte del equipo, y la mayoría dispone de un sofisticado sistema de realimentación que monitorea y ajusta constantemente la velocidad y posición de la herramienta de corte. Hoy en día, el controlador CNC trabaja en conjunto con una serie de motores, así como componentes de accionamiento para desplazar los ejes de la máquina de manera controlada y ejecutar los movimientos programados [13]. Estas tienen la siguiente apariencia:



Figura 2.1: CNC 3018 Pro [14]

2.2.2 Código G

En términos generales, G-code o código G es un lenguaje mediante el cual las personas pueden ordenar a máquinas-herramientas controladas por computadora qué hacer y cómo hacerlo. Esos "qué" y "cómo" están definidos mayormente por instrucciones como a dónde moverse, cuán rápido moverse y qué trayectoria seguir; en este caso, las máquinas típicas que son controladas con G-code son sistemas del control numéricos como fresadoras, cortadoras, tornos e impresoras 3D [15].

2.2.3 Inkscape

Inkscape es un editor de gráficos vectoriales libre y de código abierto. Inkscape puede crear y editar diagramas, líneas, gráficos, logotipos, e ilustraciones complejas. Se usa para convertir archivos a formato de código G [16].

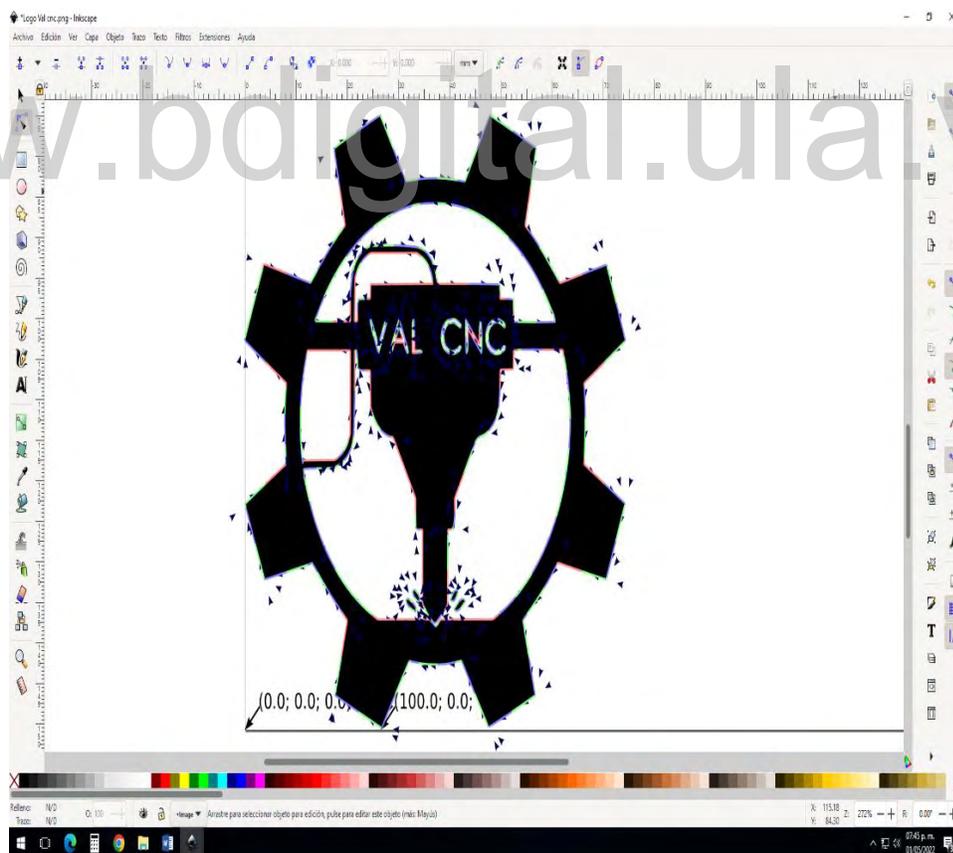


Figura 2.2: Código de trayectorias y sistema de coordenadas del logo ValCNC desarrollado por Inkscape

Para convertir una imagen en código G, es necesario seguir esta serie de pasos, específicamente 17, para que el trayecto que se forme en código sea adecuado para trabajar en el sistema:

1. Abrir la aplicación de Inkscape.
2. Importar el archivo que se quiera vectorizar.
3. Dirigirse a la pestaña de archivos, específicamente a las propiedades del documento.
4. Cambiar el tamaño de la página a uno personalizado adaptado al sistema.
5. Seleccionar la imagen y ajustarla al tamaño del cuadro o lienzo.
6. Con la imagen seleccionada, dirigirse a la pestaña trayecto y luego a mapa de bits.
7. En este menú, alterar la luminosidad y actualizar la imagen, una vez obtenida la opacidad deseada, aplicar.
8. Eliminar la imagen original y reposicionar la nueva imagen en el cuadro o lienzo.
9. Activar el desvío dinámico y seleccionar la nueva imagen.
10. Dirigirse a la pestaña de Extensiones y luego a opción *Gcode tools*, en ella seleccionar Biblioteca de herramientas.
11. Seleccionar la forma de cilindro y aplicar.
12. Con la herramienta de selección, mover el folio y con herramienta de edición de texto, alterar los parámetros (Recomendablemente el diámetro a 2mm).
13. Seleccionar la imagen en desvío dinámico y dirigirse de nuevo a la pestaña de Extensiones, luego a *Gcode tools* y ahora a puntos de orientación.
14. En este menú se altera la penetración a -1.
15. Luego dirigirse nuevamente a Extensiones, *Gcode tools* y ahora a trayecto a *Gcode*.

16. En la pestaña de preferencias, cambiar el nombre y la dirección.

17. Volver a la pestaña de trayecto a *Gcode* y aplicar.

2.2.4 Drivers A4988

El uso de un driver A4988 para el control de un motor paso a paso se encarga de generar todas las señales necesarias para su funcionamiento, además, añade las protecciones necesarias de temperatura y corriente. Otra ventaja que proporciona es que solo se necesita un par de puertos del microcontrolador, dejando más puertos libres para otras funciones [17].

Este producto es una placa de soporte o una placa de arranque para el controlador del A4988 de Allegro con protección contra sobre-corriente [17]. Estas son algunas de las características clave del controlador:

- Interfaz de control de paso y dirección simple.
- 6 resoluciones de pasos diferentes: paso completo, 1/2-paso, 1/4-paso, 1/8-paso, 1/16-paso y 1/32-paso.
- Control de corte inteligente que selecciona automáticamente el modo correcto de caída de corriente (decaimiento rápido o decaimiento lento).
- Desconexión térmica, bloqueo de bajo voltaje y protección contra cortocircuitos y carga en cortocircuito.
- El control de corriente ajustable le permite configurar la salida de corriente máxima con un potenciómetro, que le permiten usar voltajes por encima del voltaje nominal de su motor paso a paso, para lograr tasas de pasos más altas [17].

El controlador requiere un voltaje de suministro lógico (3 – 5,5 V) para ser conectado a través de los pines VDD y GND y un voltaje de suministro del motor (8 – 35 V) para ser conectado a través de VMOT y GND. Puede entregar hasta aprox. 1 A por fase sin disipador de calor o flujo de aire forzado. Compatible con motores paso a paso de 4, 6 y 8 cables que pueden ser accionados por el A4988 si están conectados correctamente [17].

En ese mismo orden de idea, se presenta un esquema de conexión del driver A4988 controlado a través de un microcontrolador lógico.

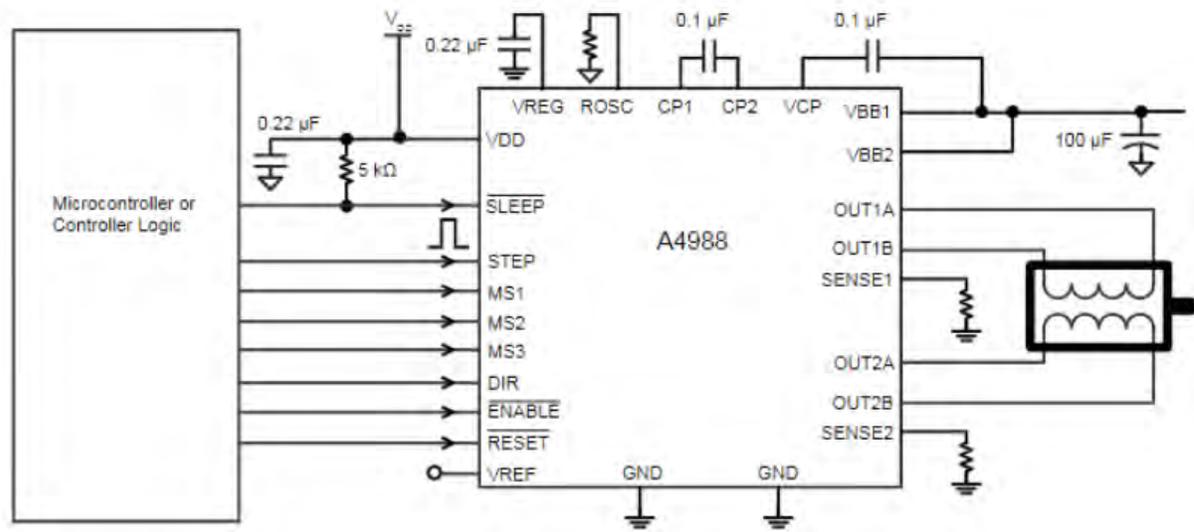


Figura 2.3: Esquema de conexión del driver A4988 [18]

El driver A4988, como se ve en la figura 2.3, necesita de una serie de componentes electrónicos para su funcionamiento, los cuales generalmente se encuentran sobre placas o *shields* que facilitan la instalación y utilización del mismo. Internamente el driver se constituye de las siguientes compuertas lógicas:

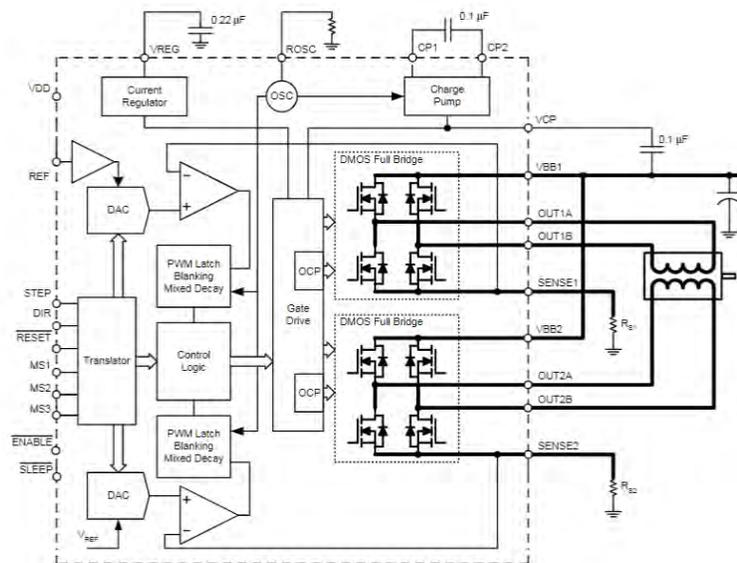


Figura 2.4: Diagrama funcional de bloques del driver A4988 [18]

En la figura resalta la posición de los diodos en forma de puente H para energizar correctamente las bobinas del motor y hacerlo funcionar de forma horaria o anti-horaria.

2.2.5 PIC18F4550

Un PIC es un circuito integrado programable (*Programmable Integrated Circuit*), el cual contiene todos los componentes para poder realizar y controlar una tarea, por lo que se denomina como un microcontrolador [19]. Los PIC son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. [19].

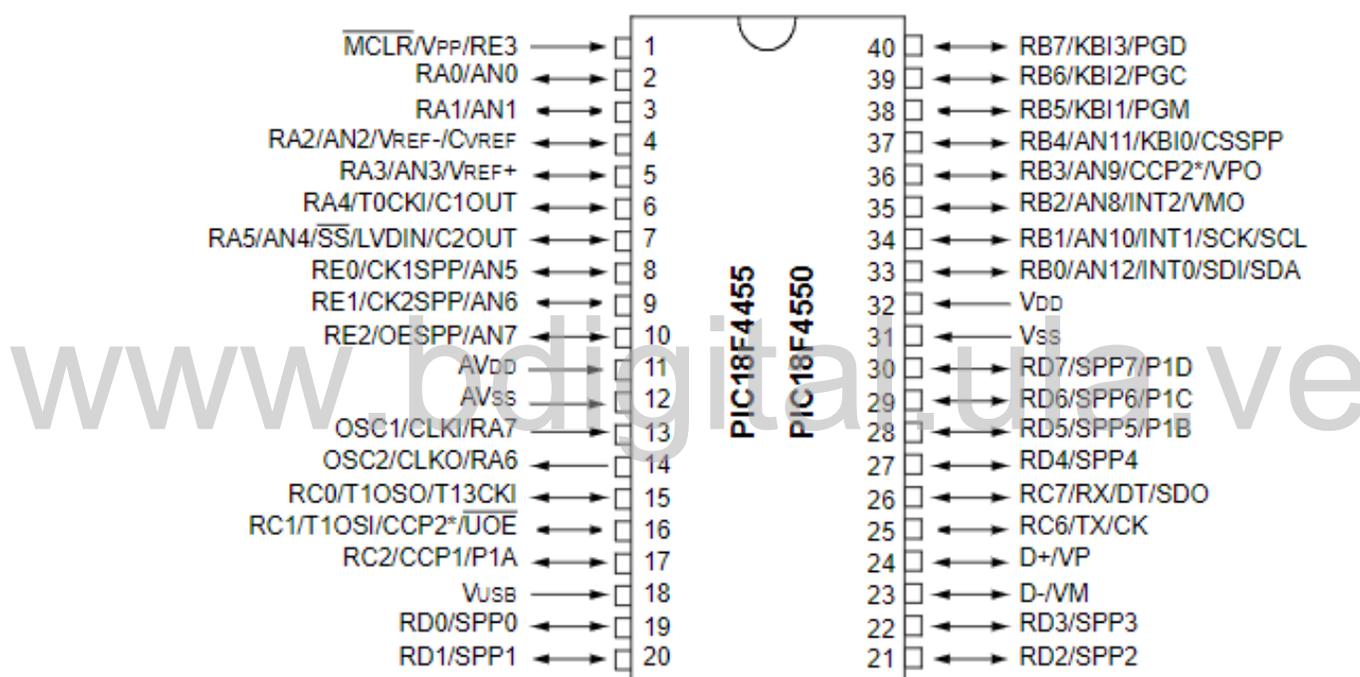


Figura2.5: Esquema de conexión del PIC18F4550 [20]

Al tener 40 pines de entrada/salida y, además, ser un microcontrolador de gama media, el PIC18F4550 es una opción viable para controlar un sistema de control numérico que funcione como prototipo.

2.2.6 MPLAB X IDE

MPLAB es un editor IDE gratuito, destinado a productos de la marca Microchip. Este editor es modular, permite seleccionar los distintos microcontroladores soportados, además de permitir la grabación de estos circuitos integrados directamente al programador. Es un

programa que corre bajo Windows, Mac OS y Linux. Presenta las clásicas barras de programa, de menú, de herramientas de estado, etc. El ambiente MPLAB® posee editor de texto, ensamblador y simulación (no en tiempo real) [21]. Este entorno de desarrollo es apropiado para programar un PIC18F4550 a través de un compilador como el MPLAB XC8.

2.2.7 MPLAB XC8

MPLAB XC8 C es un compilador independiente, optimizado para ISO C90 (popularmente conocido como ANSI C). Es compatible con todos los microcontroladores PIC® de 8 bits: PIC10, PIC12, PIC16 y dispositivos de la serie PIC18F. El compilador está disponible para varios sistemas operativos populares. El compilador está disponible en tres modos de funcionamiento: gratuito, estándar o PRO. El estándar y modos de operación PRO son modos con licencia y requieren un número de serie para habilitarlos. El modo gratuito está disponible para clientes sin licencia. El compilador básico, el funcionamiento, los dispositivos compatibles y la memoria disponible son idénticos en todos los modos. Los modos solo difieren en el nivel de optimización empleado por el compilador [22].

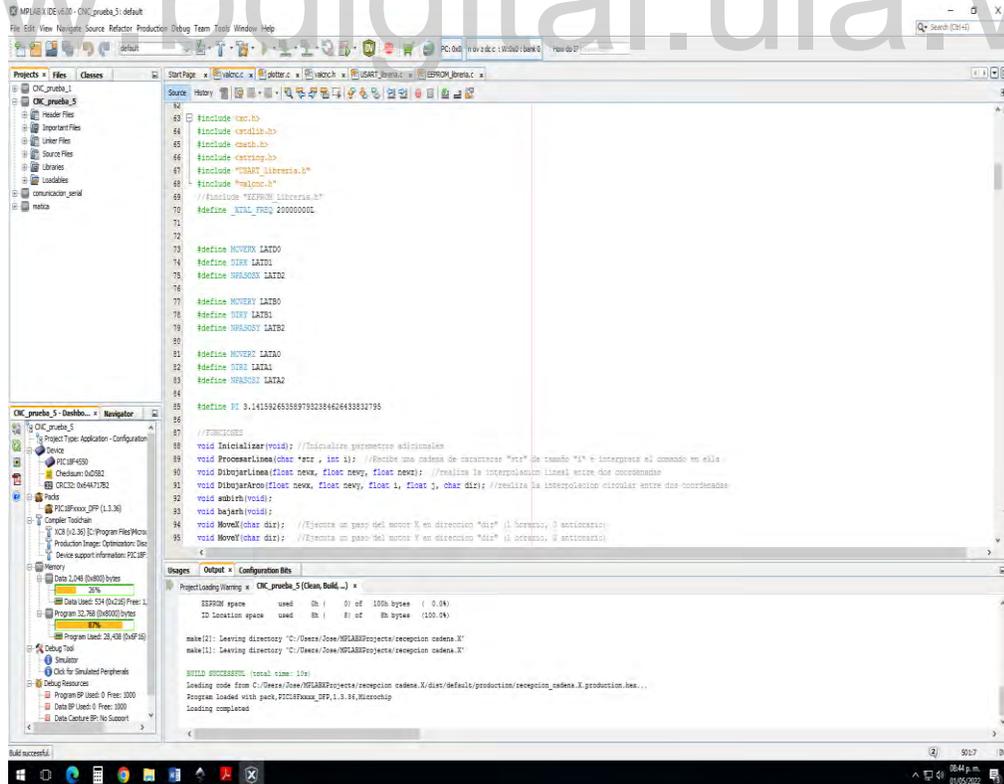


Figura 2.6: Programa escrito sobre MPLAB X IDE y compilado en MPLAB XC8Compiler

En efecto, la figura mostrada muestra el inicio del programa de un sistema CNC, donde se pueden observar la inclusión de macros, librerías y prototipos de las funciones empleadas en el mismo, con una compilación exitosa a través de compilador XC8 en su versión gratuita.

2.2.7.1 CREACIÓN DE UN PROYECTO EN MPLAB X IDE

Esta sección describe cómo crear un proyecto en MPLAB X IDE usando el MPLAB XC8 Compilador C, solo se deben seguir los pasos otorgados por el asistente de creación de nuevo proyecto [23]. El proceso se explica en los siguientes pasos:

1. El paso 1: establecer el tipo de proyecto.
2. El paso 2: seleccionar el dispositivo de destino.
3. El paso 3: seleccionar el encabezado del dispositivo.
4. El paso 4: seleccionar la herramienta para ejecutar el código del proyecto.
5. El paso 5: aplicar algunas selecciones de la herramienta de depuración.
6. El paso 6: seleccionar la herramienta para compilar el código fuente.
7. El paso 7: especificar el nombre y la ruta del proyecto.
8. El paso 8: completar la creación del proyecto.

2.2.7.2 CÓDIGO FUENTE

El código que se presenta a continuación es en realidad un pequeño programa que podría ser la base de todos los proyectos MPLAB XC8. Aunque podría interpretarse como trivial, el código es completamente válido, y compila y ejecuta, según sea necesario [23].

```
#include <xc.h>

int main(void)
{
    return 0;
}
```

Figura2.7: Ejemplo del código base de un proyecto

Este código inicial se puede utilizar para comenzar cada proyecto construido con MPLAB XC8. C. Cada programa C debe tener una, y solo una, función llamada *main()*; sin embargo, el prototipo exacto para esta función puede variar de compilador a compilador. Para todos los compiladores MPLAB XC, puede usar el prototipo que se muestra arriba. Como *main()* devuelve un entero, debe haber una declaración de devolución con un valor de retorno especificado. El valor “0” indica que ese *main()* regresó exitosamente. La inclusión del archivo de encabezado, *<xc.h>*, permite que el código en este archivo fuente tenga acceso al compilador o características específicas del dispositivo [23]. Por consiguiente, es común y necesario incluirlo en prácticamente todos los archivos de origen.

Así pues, el MPLAB X IDE sabe qué compilador ejecutar al construir el código fuente, pero las opciones o configuración, se pueden usar para alterar cómo funciona el compilador. Las opciones predeterminadas son aceptables para la mayoría de los proyectos [23].

2.2.7.3 PALABRA DE CONFIGURACIÓN

Resulta claro que todos los dispositivos Microchip PIC superiores a 8 bits, deben configurarse para garantizar su correcta operación. Algunas configuraciones afectan el funcionamiento fundamental del dispositivo, como aquellos para el reloj de instrucciones. Si esa configuración es incorrecta, es posible que el programa no se ejecute. Las opciones de configuración se especifican mediante bits especiales en el dispositivo.

La cantidad y el tipo de configuración varían de un dispositivo a otro. Es necesario consultar la hoja de datos del microprocesador que se está utilizando para controlar cada configuración. La forma más fácil para configurar un dispositivo es usando la Ventana de Bits de Configuración, una característica de MPLAB X IDE. Los siguientes pasos describen cómo abrir dicha ventana: Seleccione la pestaña Ventana, luego Vistas de memoria PIC, luego Bits de configuración en los menús [23].

El menú que se muestra a continuación en la Figura 2.8, es específico para el PIC18F4550, microcontroladores de otra gama, tienen otras características de configuración en su menú, dependiendo de su modelo.

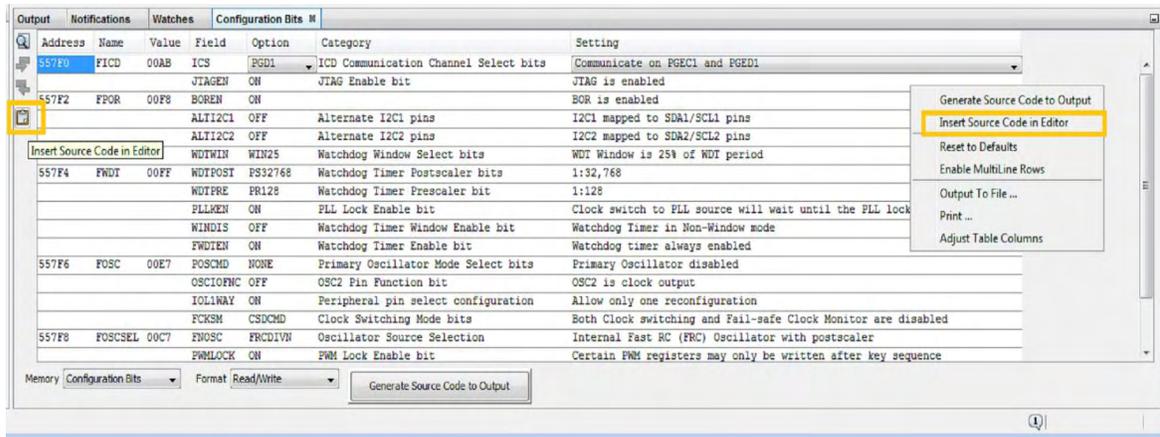


Figura2.8: Configuración de los bits para un PIC18

Las columnas “Nombre” y “Campo” ayudan a encontrar la configuración equivalente en la ficha de datos del dispositivo. La columna “Categoría” describe lo que controla la configuración. La columna “Configuración” muestra el estado actual de esa configuración.

```
#include <xc.h>

// CONFIG1
#pragma config WDTEN = OFF
#pragma config STVREN = ON
#pragma config XINST = OFF
#pragma config CP0 = OFF

// CONFIG2
#pragma config FOSC = HSPLL
#pragma config FCMEN = ON
#pragma config IESO = ON
#pragma config WDTPS = 32768

// CONFIG3
#pragma config EASHFT = ON
#pragma config MODE = XM16
#pragma config BW = 16
#pragma config WAIT = OFF
#pragma config CCP2MX = DEFAULT
#pragma config ECCPMX = DEFAULT
#pragma config PMPMX = DEFAULT
#pragma config MSSPMX = MSK7

int main(void)
{
    return 0;
}
```

Figura2.9: Palabra de configuración de los bits en código

Para la sección del oscilador, se debe coincidir los circuitos del oscilador del hardware. Si esto no es correcto, el reloj del dispositivo puede no funcionar. En el caso de un simulador como una herramienta de depuración, esta configuración puede ser ignorada. Por lo general, las placas de desarrollo utilizan osciladores de cristal de alta velocidad [23].

En cuanto al conjunto de instrucciones extendido. Esta configuración de PIC18 debe estar deshabilitada. El compilador MPLAB XC8 C no soporta este conjunto de instrucciones. Por lo que hay que cambiar la configuración haciendo clic en la línea correspondiente en la columna “Configuración” y eligiendo configuración apropiada en la lista desplegable. Con un clic en el botón “Generar código fuente” para salir. El código generado aparecerá en la ventana de Fuente de Bits de Configuración [23]. Por consiguiente, hay que copiar el código de esta ventana a un archivo de origen. Como este no es un código ejecutable y debe colocarse fuera de las definiciones de funciones.

2.2.8 BLENDER

El programa Blender es un programa informático de multiplataforma, dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales [24].

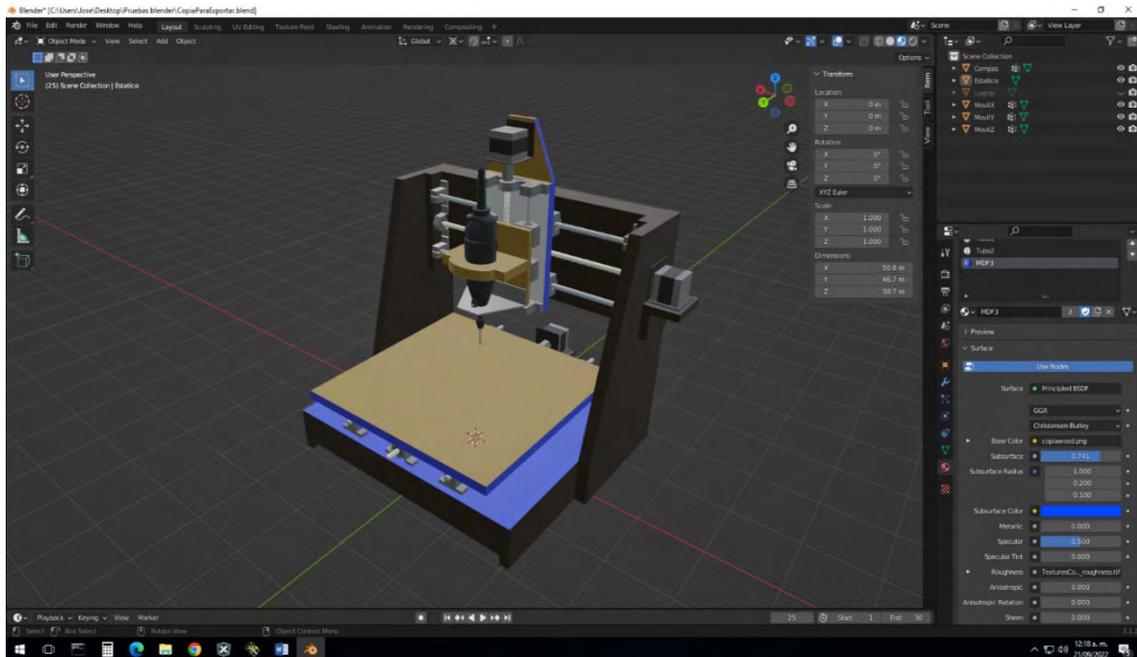


Figura2.10: Estructura del equipo modelado a través de la plataforma Blender

Esta plataforma también es usada como desarrollo de contenido digital utilizando la técnica procesal de nodos, edición de vídeo, escultura (incluye topología dinámica) y pintura digital.

El programa fue inicialmente distribuido de forma gratuita, pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre. Actualmente es compatible con todas las versiones de Windows, macOS, GNU/Linux, Android, Solaris, FreeBSD e IRIX [24].

2.2.9 UNITY

Unity es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, Mac OS, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas. A partir de su versión 5.4.0 ya no permite el desarrollo de contenido para navegador a través de su *plugin web*, en su lugar se utiliza WebGL. Unity tiene dos versiones: Unity Professional (pro) y Unity Personal [25].

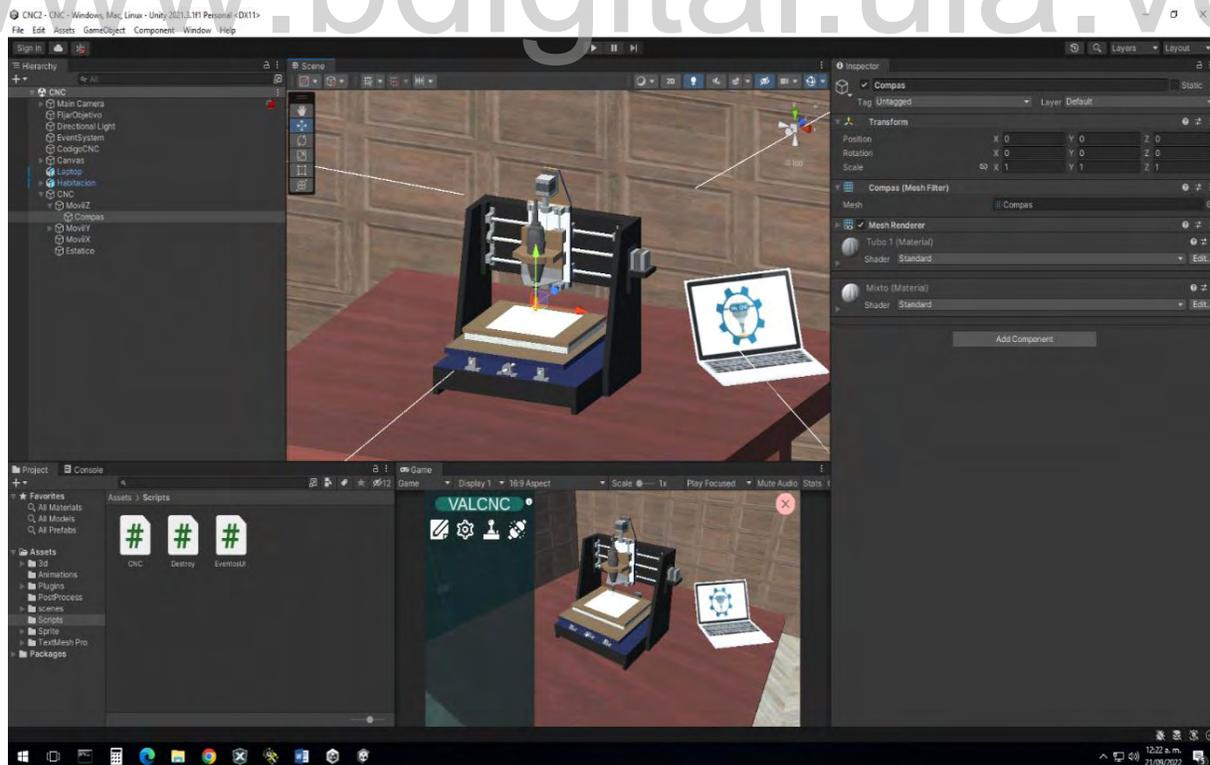


Figura 2.11: Diseño de la interfaz gráfica a través de la plataforma Unity

Unity puede usarse junto con Blender, 3ds Max, Maya, Softimage, Modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks, Allegorithmic Substance. Los cambios realizados a los objetos creados con estos productos se actualizan automáticamente en todas las instancias de ese objeto durante todo el proyecto sin necesidad de volver a importar manualmente [25].

2.2.10 PROTEUS

Proteus es una aplicación para la ejecución de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño del esquema electrónico, programación del software, construcción de la placa de circuito impreso, simulación de todo el conjunto, depuración de errores, documentación y construcción.

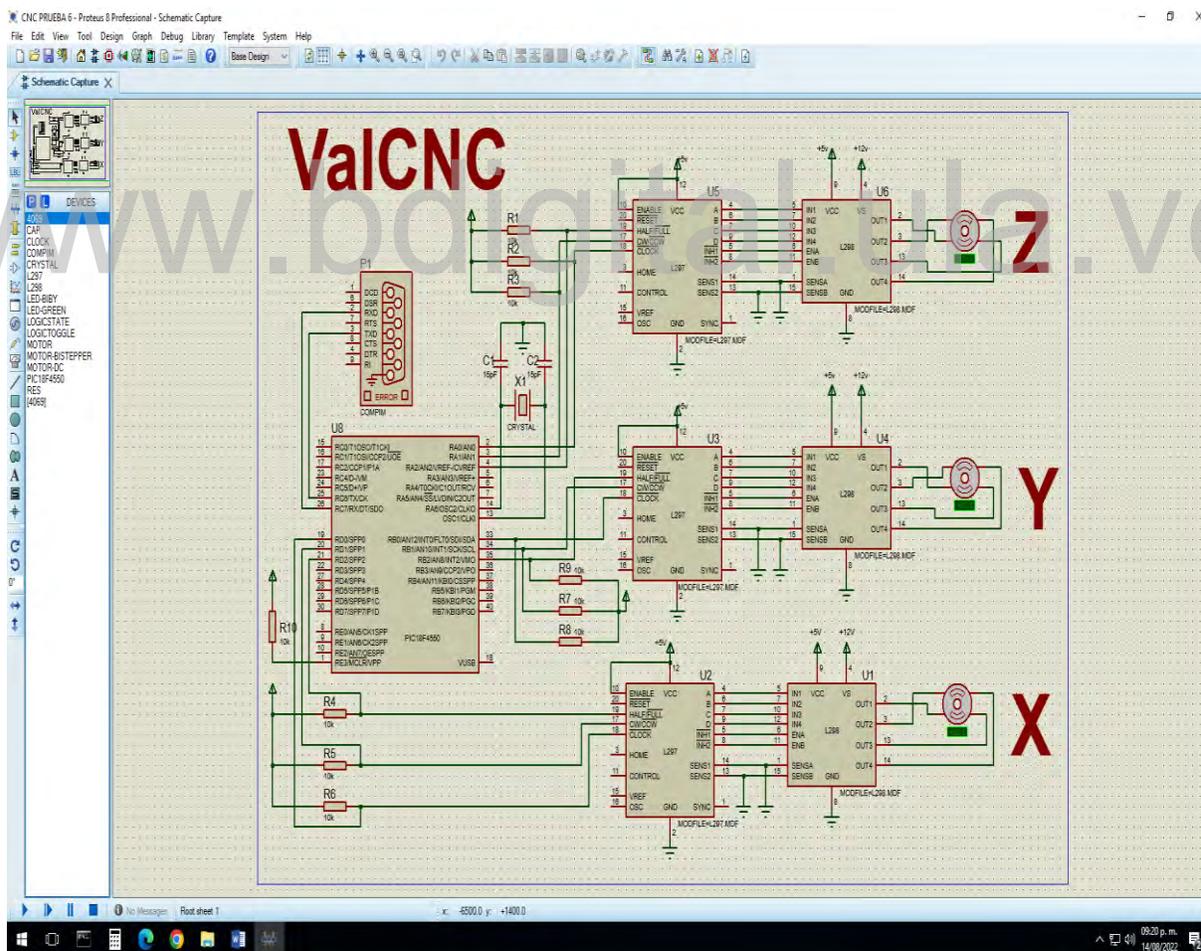


Figura 2.12: Diseño del esquema electrónico del sistema mediante el programa Proteus, versión 8.11

En efecto, Proteus es un sistema completo de diseño electrónico que combina un avanzado programa de captura de esquemas, un sistema de simulación mixto (analógico y digital) basado en Spice, y un programa para disposición de componentes en placas de circuito impreso y auto-ruteado. Se trata de un software comercial fabricado por Labcenter Electronics, caracterizado por su potencia y facilidad de uso [26].

La razón principal que conduce a simular con Proteus es su capacidad para construir circuitos basados en microcontroladores, con toda la electrónica que rodea a estos diseños. Con la capacidad de hacerlo en tiempo real, de forma interactiva, y a partir del fichero ejecutable creado por cualquier ensamblador o compilador.

2.2.11 MikroProg Suite for PIC

El programa mikroProg Suite for PIC™ de Shareware en la categoría de *Miscellaneous* es creado para programar los microcontroladores PIC de Microchip. La interfaz gráfica de este programa es clara y fácil de utilizar en compañía de una placa de desarrollo como EasyPic.



Figura 2.13: MikroProg Suite for PIC compilando el programa de al pic18F4550

La ventana principal de programa incluye las opciones básicas para programar los microcontroladores. Además, hay dos opciones de programación avanzadas que permiten a los usuarios con más experiencia ajustar los bits de configuración. El programa incluye las ventanas que proporcionan informaciones básicas sobre el microcontrolador seleccionado, que permiten monitorizar el microcontrolador [27].

Este software se utiliza para programar con todas las familias de microcontroladores de Microchip, que incluyen: PIC10F, PIC12F, PIC16F, PIC18F, PIC18F-K, PIC18F-J, PIC24F-KA, PIC24F-J, PIC24H-J, dsPIC30F, dsPIC30F- SMPS, dsPIC33F-J y PIC32MX. Además, es compatible la mayoría de los programadores de MikroElektronika para dispositivos Microchip [27].

2.2.12 Virtual Serial Ports Emulator

El Virtual Serial Ports Emulator o VSPE, es una aplicación que permite mediante un software del puerto COM virtual, emular paquetes de puerto COM virtuales con parámetros personalizables, ofreciendo una solución versátil que se puede utilizar en diversos escenarios, donde se necesite probar la simulación de un circuito electrónico [28].

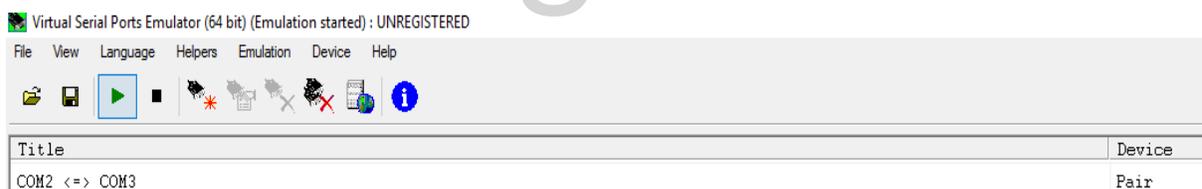


Figura 2.14: VSPE emparejando puertos seriales para las pruebas de software del sistema

En efecto, el VSPE funciona para simular un puerto serial virtual de manera de comunicar una simulación con un terminal virtual, para realizar las respectivas pruebas de funcionamiento que facilita la construcción del diseño del software del microcontrolador.

2.2.13 HÉRCULES SETUP

El Hércules SETUP se utiliza como un útil terminal de puerto serie (terminal RS-485 o RS-232), terminal UDP/IP y terminal TCP/IP *Client Server*. Actualmente su versión 3.2.8 es *freeware* e incluye muchas funciones en una sola utilidad. A través de dispositivos originales

(convertidor serie/Ethernet, RS-232/búfer Ethernet o controlador de E/S) se puede utilizar para la configuración UDP (Protocolo de Datagrama de Usuarios) [29].

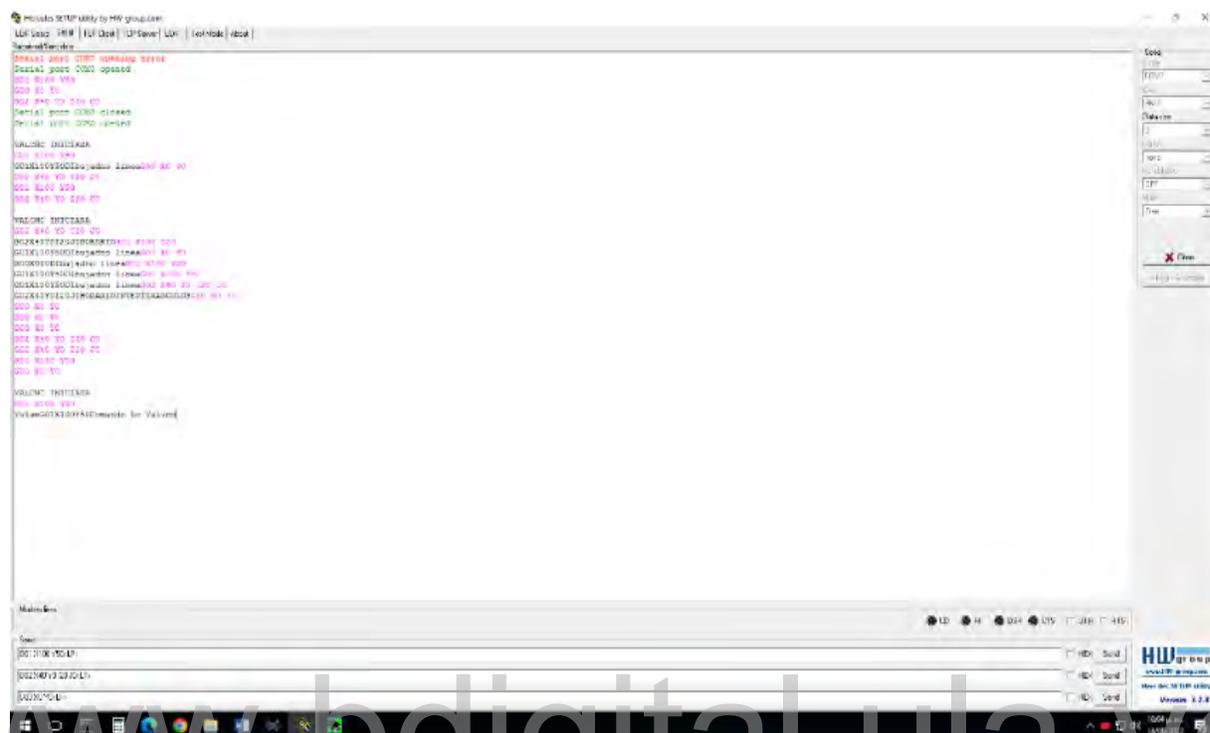


Figura 2.15: Hércules usado como terminador virtual para comunicar el computador con el PIC a través de un puerto serie

2.2.14 Microsoft Visual Studio IDE

Un entorno de desarrollo integrado (IDE) es un programa con numerosas características que respalda muchos aspectos del desarrollo de software. El IDE de Visual Studio es un panel de inicio creativo que se puede usar para editar, depurar y compilar código y, después, publicar una aplicación. Aparte del editor y el depurador estándar que proporcionan la mayoría de IDE, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas más características para facilitar el proceso de desarrollo de software [30].

Visual Studio ofrece herramientas como diseñadores, editores, depuradores, los cuales permiten programar en diferentes lenguajes de programación como C++, Python, HTML5, JavaScript, C#, VisualBasic y F#. También permite desarrollar utilizando los *framework* de

desarrollo web del tipo ASP.NET y Node.js; también tiene cualidades como desarrollar aplicaciones para las plataformas móviles como Windows Phone, IOS y Android [30].

```

53 void Update()
54 {
55     if (isAction)
56     {
57         MovILX.transform.position = Vector3.MoveTowards(MovILX.transform.position, finalX, speed * Time.deltaTime/velocidad);
58         MovILY.transform.position = Vector3.MoveTowards(MovILY.transform.position, finalY, speed * Time.deltaTime/velocidad);
59         finalZ = new Vector3(finalX, finalY, finalZ);
60         MovILZ.transform.position = Vector3.MoveTowards(MovILZ.transform.position, finalZ, speed * Time.deltaTime/velocidad);
61     }
62     if (Vector3.Distance(MovILX.transform.position, finalX) <= hueco
63         && Vector3.Distance(MovILY.transform.position, finalY) <= hueco
64         && Vector3.Distance(MovILZ.transform.position, finalZ) <= hueco)
65     {
66         isAction = false;
67     }
68 }
69
70 if (Input.GetButton("Click0") && libertaCamara) //Presionar el Click derecho del mouse
71 {
72     Cursor.SetCursor(cursor[1], Vector2.zero, CursorMode.Auto); //Cambiar el icono del cursor
73     camara.m_Axis.m_InputAxisName = "Mouse X"; //Activar el movimiento horizontal
74     camara.m_VAxis.m_InputAxisName = "Mouse Y"; //Activar el movimiento vertical
75 }
76 else
77 {
78     camara.m_Axis.m_InputAxisName = "na"; //Desactivar el movimiento horizontal
79     camara.m_VAxis.m_InputAxisName = "na"; //Desactivar el movimiento vertical
80 }
81
82 if ((camara.m_Orbits[0].m_Height + Input.GetAxis("Mouse ScrollWheel") * velocidadZoom) && libertaCamara)
83 //Cerrar la rueda del mouse para acercarse o retroceder al zoom, rotar, bajar o subir el zoom y girar la rueda
84 {
85     if ((camara.m_Orbits[0].m_Height + Input.GetAxis("Mouse ScrollWheel") * velocidadZoom < zer) && libertaCamara)
86     {
87         camara.m_Orbits[0].m_Height += Input.GetAxis("Mouse ScrollWheel") * velocidadZoom;
88         camara.m_Orbits[0].m_Radius = (camara.m_Orbits[0].m_Height + Input.GetAxis("Mouse ScrollWheel") * velocidadZoom) / 15;
89         camara.m_Orbits[1].m_Radius += Input.GetAxis("Mouse ScrollWheel") * velocidadZoom;
90         camara.m_Orbits[2].m_Radius = (camara.m_Orbits[0].m_Height + Input.GetAxis("Mouse ScrollWheel") * velocidadZoom) / 15;
91         camara.m_Orbits[2].m_Height -= Input.GetAxis("Mouse ScrollWheel") * velocidadZoom;
92     }
93 }

```

Figura 2.16: Programa escrito sobre Microsoft Visual Studio IDE

2.3 FUNCIONAMIENTO GENERAL DE UNA CNC

La mayoría de las máquinas que usan un sistema de control numérico computarizado, son operadas mediante comandos programados en un medio de almacenamiento, en comparación con el mando manual mediante volantes o palancas. Se realiza mediante un sistema de coordenadas que especificarán el movimiento de la herramienta de corte.

Así pues, para un programa informático ejecutado por una computadora, controla los movimientos de la herramienta de trabajo con relación a los ejes de coordenadas de la máquina. Por tanto, para el mecanizado de una pieza operada cualquier usuario en general, se sigue el siguiente algoritmo explicado a través de un diagrama:

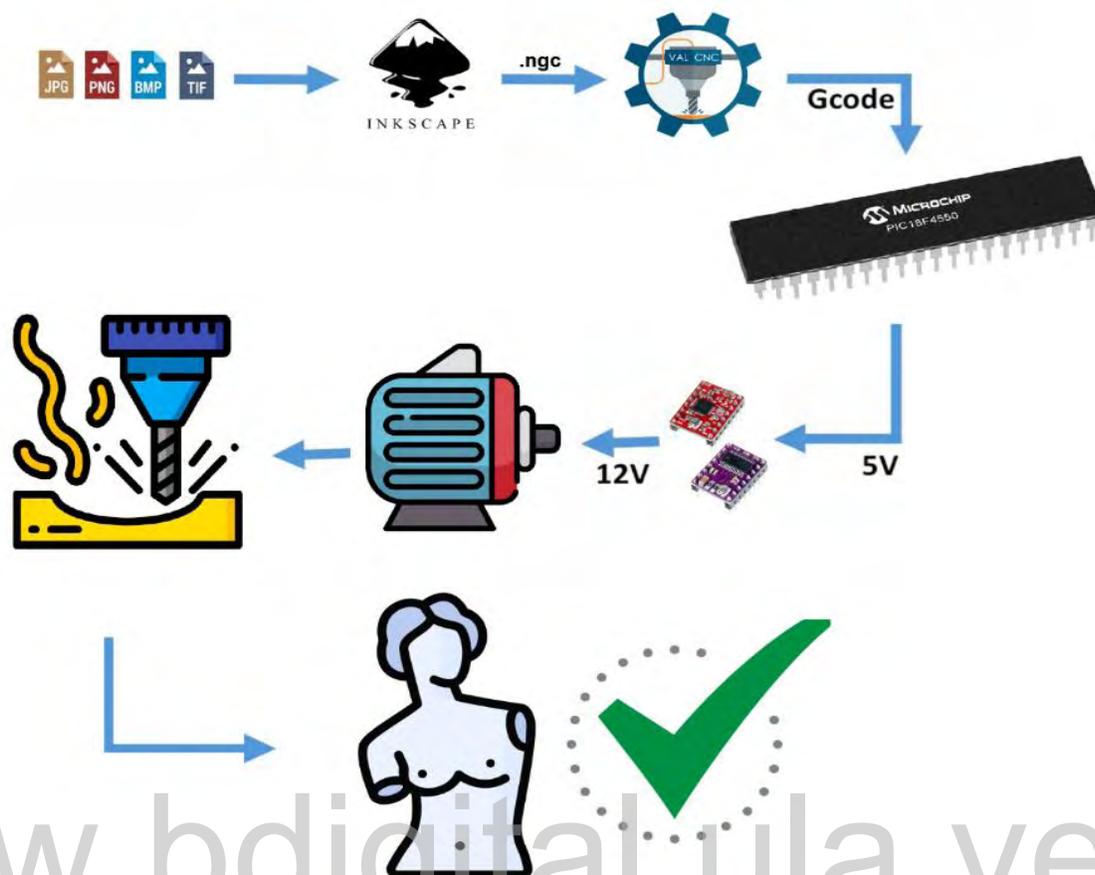


Figura 2.17: Diagrama de funcionamiento general de una CNC

Para que una CNC pueda tallar en papel, madera, metal o plástico; solo se debe tomar cualquier archivo preferiblemente en formato JPG, mediante el uso de un dispositivo digital con sistema operativo Windows o Android, el cual va a trabajar sobre el programa Inkscape, que se encarga de la vectorización de la imagen a un lenguaje de código G en una extensión en formato NGC, y esta se va a comunicar con el microcontrolador a través de un programa. Dicho programa opera y monitorea el movimiento de la herramienta, a la vez que, el microcontrolador funciona como intermediario para guardar las instrucciones en el sistema, para luego interpretarlas y enviarla en forma de pulsos eléctricos a los drivers respectivos con el fin de añadir las protecciones necesarias de temperatura y producir correctamente los niveles de corriente y voltaje, para poder controlar los motores paso a paso, que son el músculo de la CNC y controlan el deslizamiento del eje X, Z, Y. La operación de estos motores permite mover una herramienta de forma milimétrica para lograr tallar la imagen del archivo vectorizada anteriormente.

CAPÍTULO III

DISEÑO

En este capítulo se explica detalladamente cada una de las etapas de construcción del equipo, planteando las consideraciones del diseño, el mecanismo, dimensiones y estructura de la máquina.

3.1 CONSIDERACIONES DE DISEÑO

Para la construcción de una máquina de control numérico existen una serie de requisitos y consideraciones, que se deben tomar en cuenta para que el diseño de la misma sea completamente óptimo y funcional. Estas consideraciones se enuncian a continuación:

- **Rígida (firmeza).**

Para que el sistema CNC funcione correctamente, la estructura debe ser lo suficientemente estable, ya que existe vibración en el mecanismo y puede generar errores. Por lo que es absolutamente necesario que el centro de masa de la máquina se encuentre ubicado en la base o mesa de trabajo donde posa el lienzo.

- **Compacta**

El sistema debe tener las dimensiones y peso adecuado para que tenga la posibilidad de transportarse manualmente y que todas sus piezas estén ensambladas correctamente en un mismo dispositivo, con la posibilidad de que sus piezas sean removibles. Por consiguiente, la máquina debe tener un equilibrio armónico entre su tamaño y el espacio útil de trabajo que proporcione la misma.

- **Fácil mantenimiento**

El equipo debe tener la posibilidad de ser reajustado tanto para el área de hardware como para el software, de forma accesible, fácil y sencilla. Y de esta manera, aumentar el periodo de vida útil de la misma.

- **Lo más económica posible**

Las piezas a elegir para la CNC se seleccionarán en función de los costos y buscando que el sistema sea lo más preciso posible. Con la particularidad de que todas sean obtenidas en el mercado local.

3.2 MECANISMO Y DIMENSIONES

Todos los mecanismos y dispositivos que controlan los diferentes movimientos y acciones que realizan las máquinas CNC para llevar a cabo distintas instrucciones, son una combinación en conjunto de una serie de motores, así como componentes de accionamiento para desplazar los ejes de la máquina de manera controlada y ejecutar los movimientos programados [31]. Por lo tanto, la CNC va a constar de distintos elementos indispensables para su mecanismo, como lo son:

3.2.1 Ejes y rodamientos

Un eje es un elemento constructivo destinado a guiar el movimiento de rotación de una pieza o de un conjunto de piezas, como un rodamiento o engranaje [32]. En los sistemas de control numérico, se aplica el concepto de “eje”, a las direcciones de los diferentes desplazamientos de las partes móviles de la máquina. Evidentemente, para poder programar el movimiento de dichos elementos, se les asigna una letra. Este diseño de letras está normalizado; no puede ser cambiada en ningún caso. Los signos "+" y "-", no tienen un sentido matemático sino de dirección. En efecto, estas máquinas disponen de un mínimo de tres ejes X, Y, Z. Dos de ellos X e Y se asocian al movimiento en el plano horizontal (longitudinal y transversal) de la mesa de trabajo, mientras que el tercero es el desplazamiento vertical del cabezal de la máquina [33].

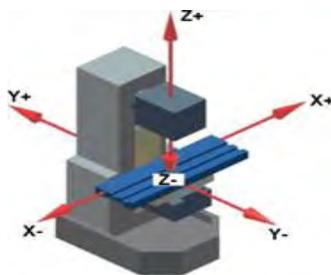


Figura3.1: Representación del desplazamiento de los ejes de una CNC [33]

Los recorridos de la herramienta se originan por la acción única o combinada de los desplazamientos de cada uno de sus ejes. Todas las máquinas de CNC tienen dos o más grados de movimiento sobre los ejes, y a su vez, cada eje o grado de movimiento pueden ser lineal o rotativo. Este concepto está ligado a la complejidad de la máquina, esto es, entre más ejes tiene una máquina más compleja es o tiene mayor capacidad de maquinar piezas complejas [33].

Un eje óptico lineal es un dispositivo de rotación automática es la pieza ideal y más usada para instrumentos de grabación automática, computadora, impresoras de precisión, tanque de gas especial, máquinas de madera automáticas y otras máquinas industriales; y debido a su demanda, puede ser encontrado fácilmente en el mercado local [33]. En cuanto a su dureza, son realmente consistentes. Por lo que pueden extender la vida de un instrumento ordinario.

Los ejes de una máquina CNC deben ser lineales y cilíndricos, de acero sólido, no hueco, que permitan un deslizamiento suave de un buje. Con medidas de aproximadamente 8x200mm para el eje Z y 8x400mm para los ejes X e Y. A su vez, los ejes deben estar sostenidos por soportes de eje de riel de aluminio para cada extremo, que eviten que los rieles se tambaleen [31].



Figura3.2: Ejes ópticos de acero

Los soportes de rieles de aluminio que sostendrán a los ejes ópticos, deben tener un orificio por donde pueda el eje de acero en cada extremo. Por lo tanto, unas dimensiones de 42x32x11mm son ideales para el diseño, como se muestra a continuación:



Figura3.3: Soporte de rieles de aluminio

Los bujes deslizantes son el elemento de una máquina donde se apoya y gira un eje. Puede ser una simple pieza que sujeta un cilindro de metal o un conjunto muy elaborado de componentes que forman un punto de unión, que van a moverse a través de los ejes, los cuales básicamente cumplen una función de anti-fricción de los elementos metálicos en las máquinas, que, a su vez, evitan o minimizar las vibraciones, el desgaste, o los ruidos en el equipo. En efecto, estos bujes deben ser atravesados por los rieles, por lo que unos bujes deslizantes de características de 8x34mm, son ideales para el mecanismo a construir [33].



Figura3.4: Bujes deslizantes

3.2.2 Tornillos de arrastre mecanizado

El tornillo de arrastre o husillo de fricción es un elemento que está formado por un tornillo y una tuerca de forma que el movimiento giratorio del husillo se convierte en el movimiento rectilíneo de la tuerca, por lo que se puede aprovechar el movimiento de la lineal de la utilizando rodamientos en sus extremos que faciliten el movimiento rotatorio del eje roscado. Por otro lado, el husillo más usado en la maquinaria industrial y de precisión, es el husillo de bolas. Aquí no hay fricción entre la tuerca y el tornillo, sino rodadura de bolas de acero, de forma que el tornillo está en contacto con las bolas y las bolas con la tuerca, obteniéndose un movimiento suave y preciso, y con buena rigidez [34].

Evidentemente es importante conocer el tipo de rosca, en función de la forma del perfil del filete, del número de filetes que tenga, del paso de la propia rosca, y del sentido de giro de avance del tornillo [34].

Para la construcción de un sistema CNC, es importante el enfoque en el número de vueltas que permita el tornillo, en relación al número de hélices del mismo. En el mercado local se encuentran con cierta facilidad 2 tipos de tornillos de arrastre, las de rosca sencilla o las de doble rosca. La diferencia entre ambas salta a la vista, pues la tuerca de doble hélice es más rápida, mientras que la de una hélice es más precisa.

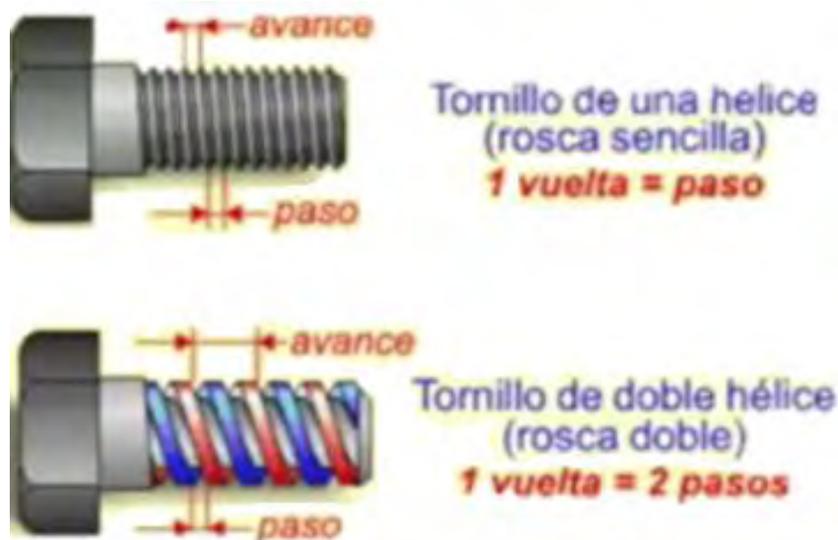


Figura 3.5: Tipos de tornillos de arrastre en cuanto al número de hélices [34]

En este mismo orden de ideas, el husillo sin fricción y el tornillo de una hélice, ofrecen una excelente relación entre calidad y precio, por lo tanto, tornillos roscados con la misma dimensión que los ejes ópticos son opción a elegir para este diseño.



Figura 3.6: Diseño de los tornillos de arrastre con sus respectivas tuercas de fricción

Ahora bien, para hacer girar los tornillos de arrastre, es fundamental acoplar el eje del motor al eje del tornillo, que por lo general no son del mismo tamaño. Por lo tanto, se necesita una pieza que logre unir mecánicamente ambos elementos. Evidentemente, un acoplamiento mecánico funciona para conectar secciones de diferentes ejes, con dirección paralela, con el fin lograr un alineamiento entre ambos [17].

Por consiguiente, un acople flexible, que permita fijar y al mismo tiempo, unir ejes con dimensiones distintas, que de un extremo tenga un orificio de 8mm para empalmar el tornillo de arrastre, y del otro extremo, un orificio que encaje con el eje del motor. Una pieza con estas características es el elemento conveniente para este caso.

En general, en las ferreterías se consiguen con cierta facilidad distintas variedades de acoples flexibles de aluminio sólido que permita el ajuste de los rieles, que a su vez cumpla con las dimensiones adecuadas para el sistema CNC, como el que se muestra a continuación en la figura:



Figura 3.7: Acoplamiento de Aluminio

Una vez acoplados el tornillo de arrastre con el motor, es necesario agregar un elemento rotatorio que reduzca la fricción de las piezas del sistema. Los rodamientos presentan la ventaja de ser una buena solución para garantizar un funcionamiento suave y eficiente en máquinas con movimiento rotativo, ya que cumplen con factores como costo, tamaño, peso, capacidad de carga, durabilidad y precisión [35].

Los rodamientos de bolas están compuestos por esferas rodantes, separadas por dos "pistas" o anillos de rodamiento, que sirven para reducir la superficie de contacto y la fricción en los planos móviles. La rotación de las bolas reduce el coeficiente de fricción cuando las superficies planas rozan entre sí. Como la superficie de contacto entre las bolas y las pistas de rodadura es relativamente pequeña, los rodamientos de bolas tienen, en vista de su tamaño, una menor capacidad de carga que otros rodamientos con elementos rodantes [35]. En este sentido, se pretende el uso de rodamientos del tipo lineal para ambos extremos del tornillo con dimensiones de 55x13x30mm., que para este caso son mejores del tipo lineal que los radiales, ya que permiten un desplazamiento lineal de los ejes, pues estos funcionan como guías lineales, así pues, permiten el movimiento en una dirección a lo largo de un eje lineal.



Figura3.8: Rodamientos lineales

3.2.3 Motores

Para controlar de forma muy precisa el giro de un eje para hacer movimientos de precisión y así controlar el movimiento de una máquina CNC es necesario utilizar un motor paso a paso [17].

El motor paso a paso es un motor de corriente continua sin escobillas en el que la rotación se divide en un cierto número de pasos resultantes de la estructura del motor. Este motor paso a paso convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de avanzar una serie de grados (paso) dependiendo de sus entradas de control [36].

Actualmente, los motores y generadores paso a paso se rigen por los estándares principales para la producción o NEMA, por sus siglas *National Electric Manufactures Assciation*, Son los Estándares principales en los Estados Unidos y la Norma Nema MG1 dicta los estándares para la producción de motores y generadores a nivel continental, Por tanto, existen varios tipos de motores NEMA que se diferencian por el tamaño del cuadro, el diámetro, el rango de par típico y el rango de velocidad típico.

Tabla 1: Características de los motores de paso NEMA

Tamaño del cuadro	Diámetro (mm)	Rango de par típico (Nm)	Rango de velocidad típico (RPM)
NEMA 8	20	0,01 - 0,04	0-1000
NEMA 11	28	0,06 - 0,12	0-1000
NEMA 14	35	0,05 - 0,5	0-1000
NEMA 16	39	0,1 - 0,25	0-1000
NEMA 17	41	0,2 - 1	0-1800
NEMA 23	57	0,5 - 3	0-1800
NEMA 24	60	1,2 - 4,6	0-1800
NEMA 34	86	3 - 12	0-1800
NEMA 42	102	12 - 20	0-1800

Para este caso, el músculo perfecto para mover el sistema, son motores paso a paso nema 17, ya que son los más comerciales y además, son capaces de mover cargas superiores a los 3,2kg/cm [36].

Por otra parte, cumplen con las siguientes características:

- Corriente continua sin escobillas
- Está equipado con 1,8 de ángulo de paso (200pasos/revolución)
- Dimensiones: 40x40x40 mm con una punta de 22mm.
- Corriente de cada fase es 1,7 A y resistencia de fase de 1,5Ω, y la celebración momento es 40Ncm.
- Resistencia de sensado 10Ω
- Nema 17 bipolar 4hilos: amarillo a +, azul A-, verde B +, rojo B- [36].

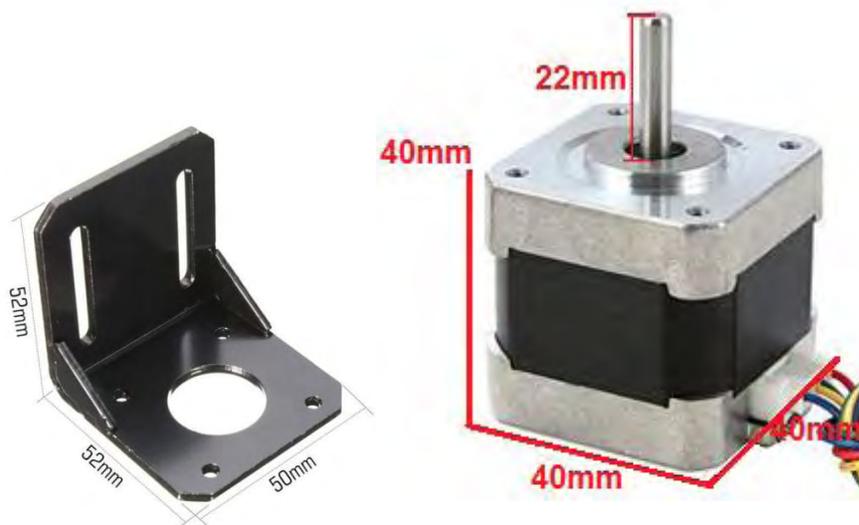


Figura 3.9: Dimensiones de soporte y motor paso a paso NEMA 17 bipolar.

Además, los motores de paso nema 17, del modelo 17HD48002H-22B cuentan con características eléctricas y mecánicas, tales como: ratio de voltaje, corriente máxima de fase, resistencia de sensado, inductancia de fase, peso y longitud, las cuales se expresan en la siguiente tabla:

Tabla 2: Características eléctricas de los motores de paso NEMA modelo 17HD48002H-22B [36]

Modelo	Ratio de voltaje	Corriente máxima de fase	Resistencia de sensado	Inductancia de fase	Peso	Longitud
17HD48002H-22B	2,8 V	1,68 A	10 Ω	3,2 mH	0,35kg	>48 mm

En resumen, para completar la parte mecánica y eléctrica de una máquina CNC, se deben adquirir los siguientes elementos:

- 2 piezas de ejes ópticos de acero de 8x200mm para el eje Z.
- 4 piezas de ejes ópticos de acero de 8x400mm para los ejes X e Y.
- 1 pieza de varilla de tornillo de arrastre de acero inoxidable de 200x8mm con tuerca para el eje Z.

- 2 piezas de varilla de tornillo de arrastre de acero inoxidable de 400x8mm con tuerca para los ejes X e Y.
- 6 piezas x 55x13x30mm cojinete o rodamiento de bloque de aleación de Zinc, 2 para cada husillo.
- 12 piezas x 42x32x11mm soporte de eje de riel de aluminio, 2 para cada riel.
- 12 piezas x buje deslizante de 8x34,5mm, 2 para cada riel.
- 3 piezas x 5x8mm D19L25 Acoplamiento de eje Flexible, 1 para cada motor.
- 3 Motores paso a paso Nema modelo 17HD48002H-22B.
- 3 Soportes de Acero para Motor Nema 17.
- 3 Sensores finales de carrera o *Limit Switch End Stop*, 1 para cada eje.
- Dremel 3000 Model 275.

Además de estos elementos, las piezas mecánicas necesitan de una estructura funcional donde cada una de ellas pueda realizar su función específica adecuadamente.

3.3 ESTRUCTURA

La estructura de una máquina es la parte sobre la que se apoyan todos los componentes que la conforman. Funciona como la columna vertebral del sistema y se construye de manera que todas sus piezas funcionen apropiadamente. En general, se fabrican a partir de piezas o elementos de hierro, aluminio, zinc o plástico [17]. Para cumplir con las consideraciones de diseño, se elige la construcción de la edificación a partir de piezas en madera MDF, puesto que este material garantiza rigidez por estar fabricado a partir de fibras de madera, y a su vez, se consiguen con cierta facilidad ya que son reciclados de otros trabajos. Lo que asegura economía en la estructura. Por lo tanto, se define a continuación, la arquitectura de la máquina mediante un diseño modelado y renderizado de forma tridimensional, bajo la plataforma de Blender. Para posteriormente usar los diseños en una interfaz gráfica que simule el movimiento de la misma.

3.3.1 Diseño principal

Para realizar una estructura funcional es necesario conocer las dimensiones de cada una de las piezas que conforma el equipo, las cuales deben corresponder en magnitud con los elementos reales, para así, garantizar el desenvolvimiento adecuado del sistema, por lo tanto, la presentación de las piezas mecánicas y estructurales que conforma el diseño la máquina, es la siguiente:

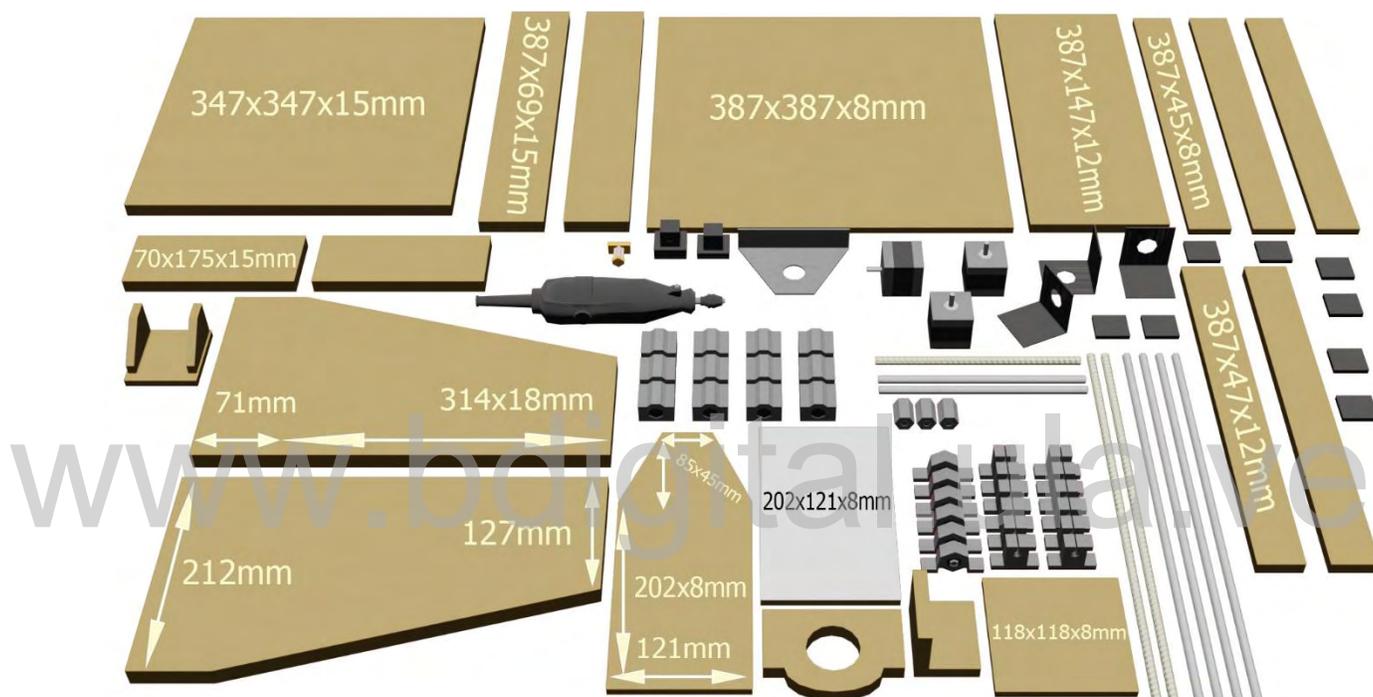


Figura 3.10: Dimensiones del diseño en milímetros modelado en 3D

Para lograr el ensamblaje óptimo de las piezas, es necesario crear un diseño que garantice que el conjunto de elementos se encargue del movimiento excelso y funcional de cada uno de los ejes coordenados (X, Y, Z), acoplados a su respectivo motor.

En este mismo orden de ideas; se pretende un diseño de 3 estructuras: una torre que soporte el eje X, correspondiente al movimiento lineal de izquierda a derecha; la base o bastidor que funcione tanto como cimiento para el eje Y, como para la estructura principal, correspondiente al movimiento de adentro hacia afuera; finalmente, la caja de soporte donde va ubicado el eje Z, correspondiente al movimiento lineal de arriba hacia abajo. Las cuales van

a estar unidas a partir de tornillos m3x25mm y m5x35mm, para que aseguren estabilidad mientras los motores produzcan movimiento.

3.3.2 La base de la máquina

Lo primero a considerar cuando se diseña un sistema de control numérico, es la base o bastidor, que es el elemento estructural principal que mantendrá unido el conjunto de la máquina, además, permite determinar la ubicación de los motores de desplazamiento y de los tornillos de avance junto con el resto de los elementos, además de ser factor que garantiza que el centro de masa del equipo repose sobre la mesa de trabajo [17].

La base está determinada en parte por el tamaño de los ejes y los tornillos de arrastre. Por ello, la alternativa de madera MDF de 12mm de grosor, es lo suficientemente sólida y firme para actuar como bastidor y soportar trabajo de los motores, y su vez, permite un diseño robusto, que no es susceptible a que la estructura se doble por su propio peso o un peso externo, y no es tan sensible a vibraciones [17].

La base, la cual pretende estar en contacto directo con el piso o cualquier superficie sólida, como una mesa; está formada por 6 tablas. 3 de ellas con medidas 387x45x8mm, unidas en paralelo con 2 tablas de dimensiones 387x65x15mm, conformando así, la forma de un bastidor o cuadro doble. Sobre las 3 tablas va a reposar la base de la estructura, esto con el fin de asegurar una máquina poco susceptible a vibraciones, de medidas 387x387x8mm. A modo de orientación, se muestra un modelado del diseño expuesto para la base:

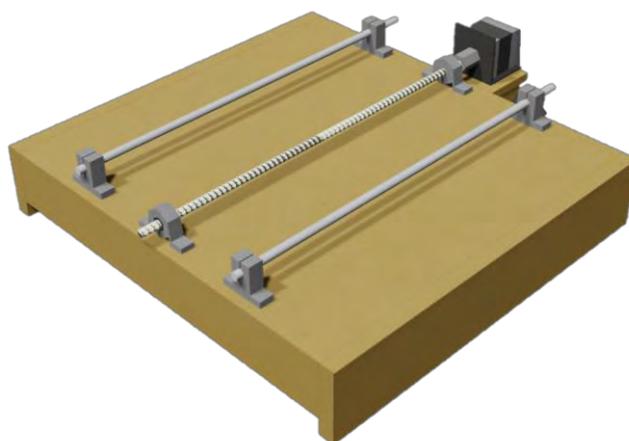


Figura 3.11 Ensamblaje de la base de la máquina modelado en 3D

Sobre la base, van a reposar el juego de ejes y tornillos que dan movimiento al eje de coordenadas Y. Con el tornillo de arrastre ubicado en el centro de la base, separado a 100 mm de cada uno de los rieles que acompañan al mecanismo del eje. Acoplado a su respectivo motor, que reposa sobre su propia cama de madera, con un acoplamiento de eje flexible, como se aprecia en la figura. Sobre el mecanismo, por medio de los bujes deslizantes y la tuerca del husillo va a reposar la mesa de trabajo o cama de diseño. La cama es cuadrada y tiene 347mm por lado, como se muestra a continuación en la figura tridimensional:

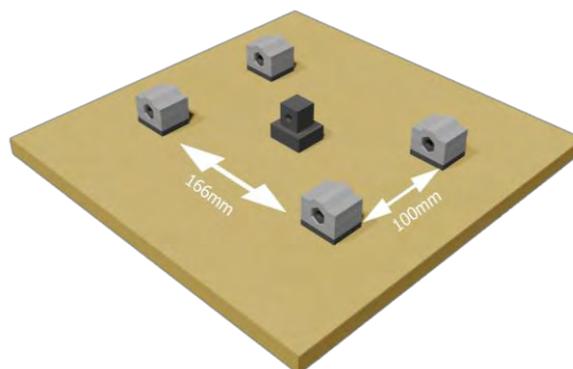


Figura 3.12: Cama de la base por donde van a deslizar los bujes del eje Y modelado en 3D

Evidentemente, la conjunción de la base con la mesa de trabajo, forman la primera parte del diseño, es decir, el ensamblaje de las figuras 3.11 y 3.12, resulta finalmente en la siguiente:

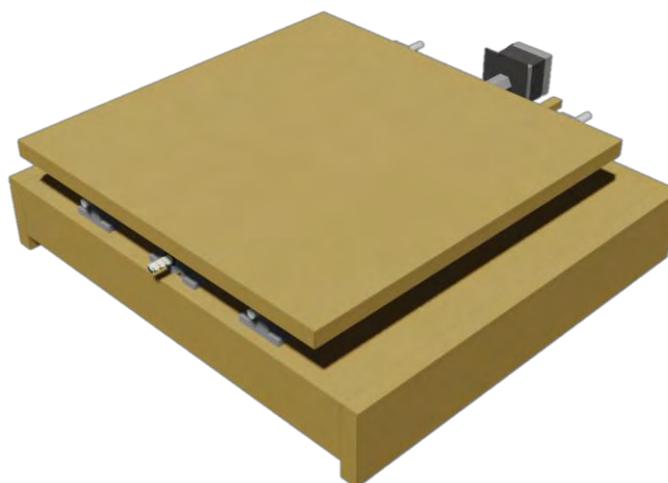


Figura 3.13: Cama y base ensambladas modelado en 3D

3.3.3 La torre

A la base, se les agregan a los costados, soportes en forma de torre que son los encargados de sostener el mecanismo para el eje X. Siguiendo el mismo orden de ideas, estos soportes son 2 tablas de medidas 387x212x18mm., con una perforación de diámetro de 20mm para incrustar el motor con su respectivo mecanismo. Para no realizar más perforaciones a las torres y debilitar la estructura, se agregan 2 tablas de dimensiones 175x70x15mm, con perforaciones necesarias para impedir el movimiento innecesario de su mecanismo. Además, se le agrega a la torre, una base adicional, donde puedan reposar el sistema eléctrico, que incluye fuente de alimentación y placa electrónica. Como se muestra a continuación:



Figura 3.14: Torre del diseño modelado en 3D

Cabe destacar que las tablas utilizadas para la edificación de torre del diseño, necesitan ser las de mayor grosor, para que brinden un mayor soporte a la estructura, ya que esta va a estar adosada al bastidor. Por otra parte, a través de los rieles del eje X, de forma análoga a lo sucedido en el mecanismo del bastidor, van a pasar los bujes deslizantes que comunican con la caja del eje Z, como se muestra en la siguiente figura:

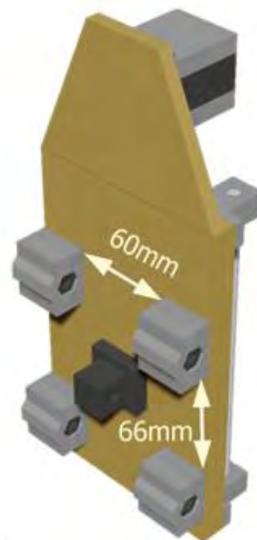


Figura 3.15: Soporte del eje X ubicado en la caja del eje Z modelada en 3D

A continuación, se muestra un ensamblaje de las estructuras ya edificadas, que son las de la base y la torre, a partir de una vista posterior, para apreciar el espacio que puede ocupar la fuente de alimentación y la placa del circuito:

www.bdigital.ula.ve

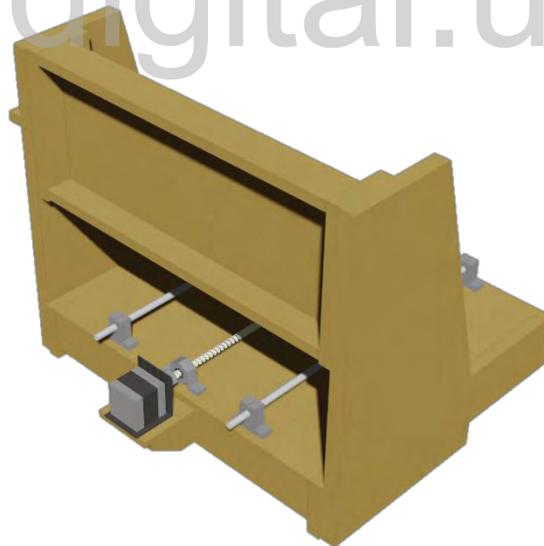


Figura 3.16: Torre ensamblada a la base modelada en 3D

3.3.4 Caja de soporte de la herramienta

Para culminar la construcción de la estructura principal, solo hace falta ensamblar la caja que soporta el eje Z, y esta consta de una base en forma de hexágono irregular capaz de

soportar el motor y el mecanismo del eje Z, unida a través de los bujes deslizantes al mecanismo que soporta la torre. La base principal de este mecanismo, es la que va a soportar la herramienta de trabajo, y tiene dimensiones de 120x120x8mm. Como se muestra a continuación:

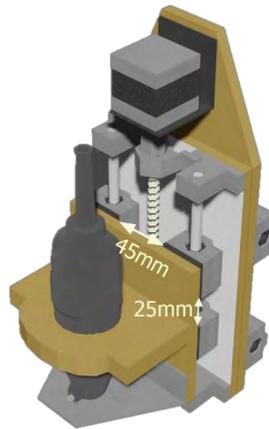


Figura 3.17: Caja de soporte para la herramienta de la CNC modelado en 3D

En definitiva, una vez ensambladas la torre, la base y la caja de soporte, se obtiene el diseño principal de la máquina. Los motores de cada eje van a ir conectados a un circuito de control, que se encarga de la alimentación y del correcto funcionamiento de los componentes electrónicos.



Figura 3.18: Diseño final de la máquina

3.4 CIRCUITO DE CONTROL

El circuito se alimenta con 2 distintos niveles de tensión, 12V para la habilitación de los motores nema 17 a través de los drivers A4988; 5V para la alimentación del microcontrolador PIC18F4550. Estos serán suministrados a partir de una fuente de poder modelo ATX202-3545, con capacidad de 600W /20A.



Figura 3.19: Representación visual de la fuente de poder que proporciona los niveles de tensión [37]

Para el control de los motores de paso, se utilizan los drivers A4988 el cual, genera todas las protecciones y señales necesarias para su funcionamiento óptimo. Para evitar sobrecalentamiento, se le agregan disipadores de calor. Los cuales deben ser calibrados a un nivel de voltaje de referencia (V_{ref}), que indique la corriente máxima soportada por los motores de paso para evitar poner en riesgo los mismos.

La corriente a través de los devanados del motor está controlada por una regulación de corriente PWM de frecuencia fija. Cuando se habilita un puente H, la corriente aumenta a través del devanado a una velocidad que depende del voltaje de corriente continua e inductancia del devanado. Una vez que la corriente alcanza el umbral de corte actual, el puente desactiva el actual hasta el comienzo del siguiente ciclo PWM. En los motores paso a paso, la regulación de corriente se utiliza para variar la corriente en los dos devanados de forma semi-sinusoidal. para proporcionar un movimiento suave. La corriente de corte PWM se establece mediante un comparador, que compara el voltaje a través de una resistencia de detección de

corriente, multiplicado por un factor de 5, con un voltaje de referencia con un 10% de protección [18].

Por lo tanto, la ecuación que describe lo expresado anteriormente, se traduce de la siguiente manera:

$$I_{MÁX} = \frac{V_{REF}}{5 R_{SENSE}} - 10\%$$

Como la resistencia de sensado tiene un valor de 10Ω y de la tabla 2 de las características eléctricas de los motores nema modelo 17HD48002H-22B, se conoce que la corriente máxima de fase que soportan estos motores es de 1,68 A. Sin lugar a dudas, el V_{ref} , queda expresado de esta forma:

$$V_{REF} = \frac{1,68 A \times 5}{10 \Omega} - 10\%$$

$$V_{REF} = 0,84 V - 0,084 V$$

$$V_{REF} = 0,756 V$$

En efecto, la calibración de los drivers para el control de los pasos debe ser a un voltaje de referencia de valor 0,75 V.

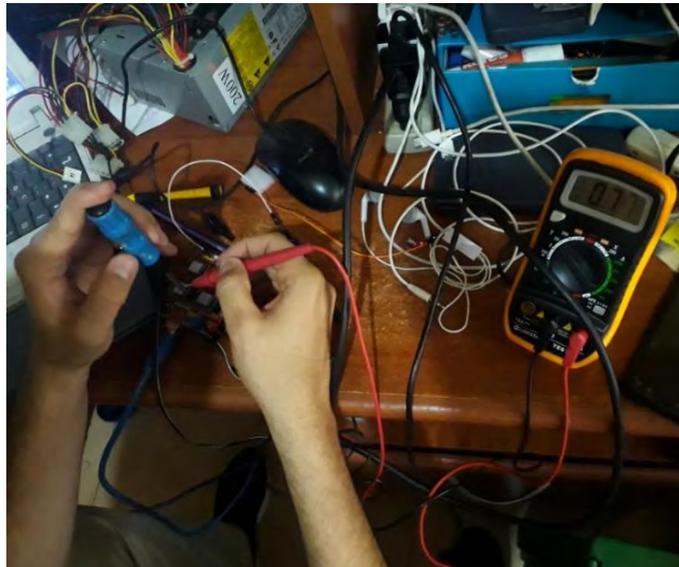


Figura 3.20: Calibración del voltaje de referencia de los drivers

Para la frecuencia del oscilador se usa el oscilador interno, oscilando a 8Mz, lo que permite que el microcontrolador trabaje sobre ese rango, el cual es suficientemente rápido si se necesita para transmisión y recepción de datos de una cadena de comandos.

En el caso de las salidas del microcontrolador, las cuales operan a corrientes en el orden de los 25mA, se conectan a una resistencia de protección y luego a su respectiva alimentación, sin interrumpir la corriente necesaria de entrada de los drivers.

Por otra parte, el circuito de control va a reposar sobre sobre una placa Shield especializada para el diseño de CNC. Compatible con los drivers para motores paso a paso Pololu A4988 (Allegro) o los DRV8825 (Texas Inst.).

El CNC Shield V3 es un módulo de expansión que se usa principalmente en máquinas de grabado e impresoras 3D que te permite empezar a construir una maquina CNC, esto en conjunto de la tarjeta de desarrollo para trabajar con Arduino [38], pero se va a realizar una adaptación de los puertos para que trabaje bajo una plataforma de software basado en microchip, específicamente en el PIC18F4550 como microcontrolador del sistema.

Por otro lado, El Shield CNC, permite manejar los 3 motores de pasos de forma independiente (X, Y, Z) y 1 motor adicional (A) que cual permite el control de un extrusor o de un *spindle*, para ser usados como herramienta por el sistema, por medio de 2 puertos IO.

Especificación y características del Shield CNC 3.0:

- Alimentación: 12-36V DC. (Dependiendo de los controladores utilizados)
- Soporte para 4 ejes (X, Y, Z, A)
- Soporta hasta 6 finales de carrera 2 por cada eje X, Y y Z
- *Jumpers* para control de *micro-stepping*
- Los motores pueden ser conectados con bornes tipo Molex de 4 pines
- Dimensiones: 69 x 53x 19 mm
- Comunicación serial de transmisión directa con la placa.

- Comunicación serial de recepción directa con la placa.
- Salidas: Dirección o *Spindleenable* y refrigerante o *Coolantenable* [39].

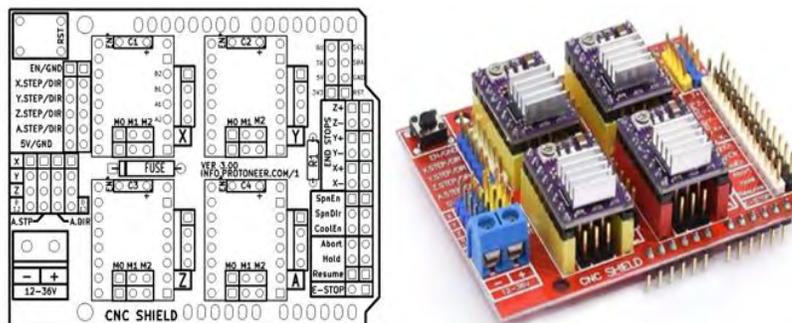


Figura 3.21: Esquema de conexiones y modelo de una Shield CNC [38]

La placa PCB donde se alojará la Shield CNC, es una placa de medidas 70x90mm, la cual cumple con las dimensiones necesarias para la distribución óptima de los componentes electrónicos usados.

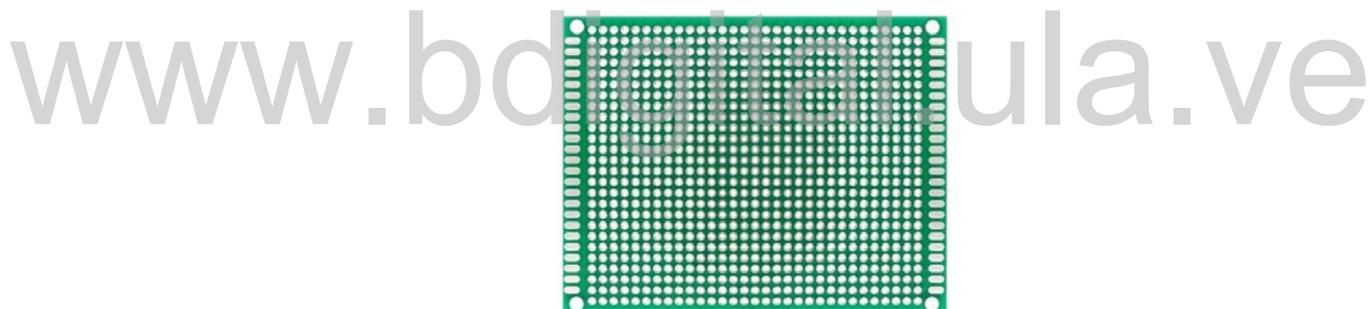


Figura 3.22: Placa PCB perforada de dimensiones 7x9 cm [40]

Resulta claro que, para alojar correctamente en la placa PCB el PIC18F4550, es necesario una base de 40 pines que le proporcione un correcto funcionamiento de todas sus entradas y salidas análogas/digitales.

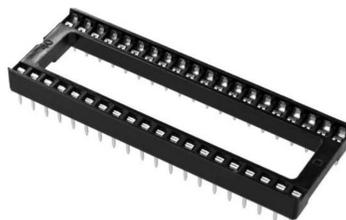


Figura 3.23: Base para circuito integrado de 40 pines

En perspectiva, la base permite que el circuito integrado programable funcione adecuadamente, además de ofrecer comodidad en su manejo.

Para la protección mecánica del equipo, se le agregan interruptores de posición o sensores finales de carrera, que detectan la ubicación de un elemento móvil mediante accionamiento mecánico, es decir, una vez la palanca presione el botón, se envía un pulso eléctrico que provoca que se detengan los motores paso a paso. Estos sensores son usados comúnmente para precisar la posición inicial de la máquina, o punto de partida, en coordenadas (0,0,0).



Figura 3.24: Sensores finales de carrera

Para la comunicación serial, se usa el interfaz serial USB-UART (FT232). El cual, se alimenta directamente del puerto USB, y su pin VCCIO puede ser puesto a 3,3V por medio del Jumper. Utiliza un puerto Mini USB (Mini-B) para conectarse al ordenador, para esto puede usar el Cable Mini USB. La tarjeta posee dos regletas de 9 pines con espacios de 0,1" (2,54mm) entre pines. El terminal de salida de 3,3V puede proveer hasta 50mA para lógica externa y sus dimensiones son de 18x35mm.

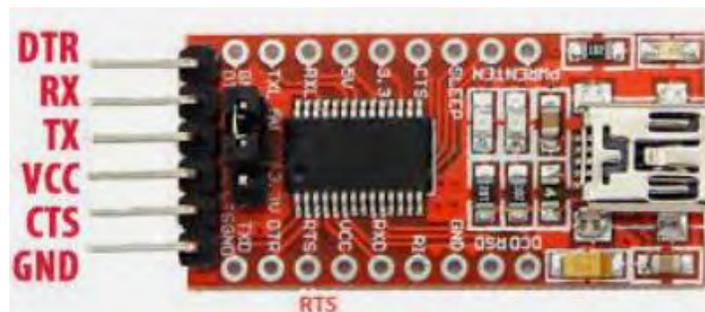


Figura 3.25: Módulo de comunicación serial FT232

Para el acoplamiento de la herramienta en el circuito, se hace uso de un relé. El relé es un dispositivo electromagnético. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

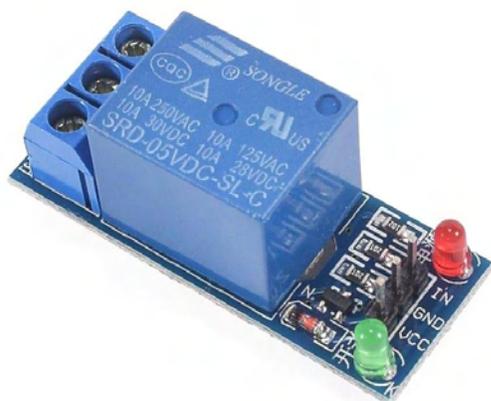


Figura 3.26: Módulo relé

En este sentido, un Dremel es la herramienta más adecuada para ser incluida en el diseño, ya que ofrece una gran variedad de funciones, en las que resaltan tareas como el bricolaje, reparaciones, soldar, lijar, cortar, pulir, perforar, pegar, grabado y más trabajos a una velocidad óptima.



Figura 3.27: Dremel model 275

3.5 ESTUDIO TÉCNICO-ECONÓMICO DE LOS ELEMENTOS DE UNA CNC

Se presenta un estudio comparativo, para definir la viabilidad de adquisición de los elementos de un control numérico en el mercado local, tomando en cuenta que la misma logre los distintos procesos como taladrar, cortar, roscar, fresar, o desbastar en todos sus planos de forma totalmente automática a una misma pieza. Para desarrollar una propuesta viable económicamente, se realiza una investigación en mercado local con el fin de poder obtener los costos de cada componente que conforman al sistema CNC, asimismo, se investiga el costo de adquisición de una máquina ya construida, para así, poder realizar la comparación entre ambos presupuestos y determinar su viabilidad. En las siguientes tablas comparativas, divididas en piezas mecánicas y piezas eléctricas, respectivamente; se construye una tabla de presupuesto para una CNC prefabricada y luego se procede a comparar con los precios de una máquina ya construida, y así confirmar la factibilidad de la propuesta.

En la ciudad de Mérida, Venezuela, existen distintas localidades que cuentan con mercancía apta para la construcción de la máquina. Ahora bien, en cuanto a las piezas mecánicas, los sitios donde se pueden adquirir estos elementos, son los siguientes:

Opción 1: **Tornillos Mérida**. Ubicados en la Avenida 16 de septiembre, Mérida 5101, Mérida. Cuentan con el kit de piezas completo para construir el mecanismo.



Figura 3.28: Ubicación de Tornillos Mérida

Opción 2: **FERRETODO La Milagrosa**. Avenida los próceres al lado de Mr. Freno, Troncal7, Mérida 5101, Mérida. Este local tiene una sección en el segundo piso, donde se trabaja con sistemas de control numérico y cuentan con tornos y elementos electromecánicos ideales para un sistema de control numérico.

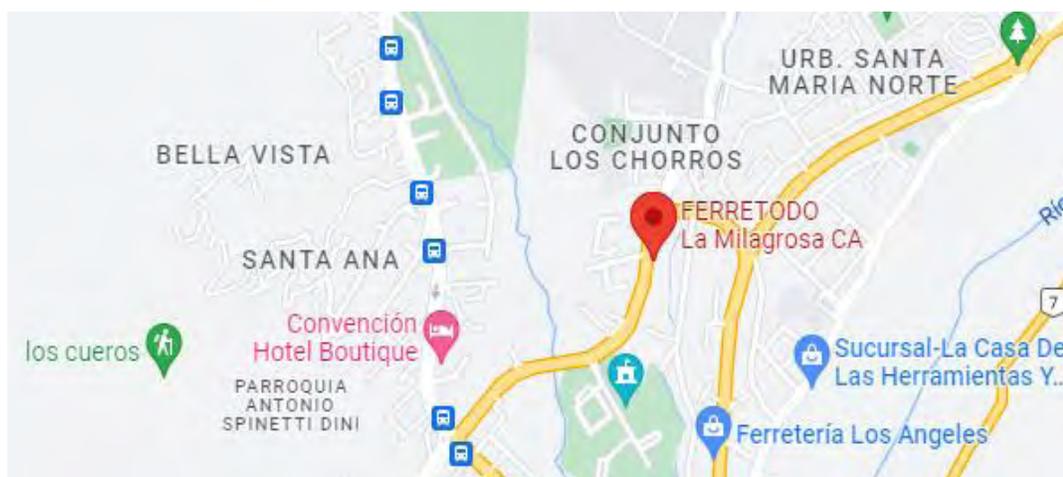


Figura 3.29: Ubicación de FERRETODO La Milagrosa

Opción 3: **Ferrelectrico CARACAS Mérida C.A.** Ubicados en la Calle principal, media cuadra arriba de la entrada a las Delicias. Sector Santa Juana, Local 2-72. HRGJ+FM, Mérida 5101. Es una ferretería con una excelente atención donde se pueden adquirir piezas mecánicas a bajo costo.

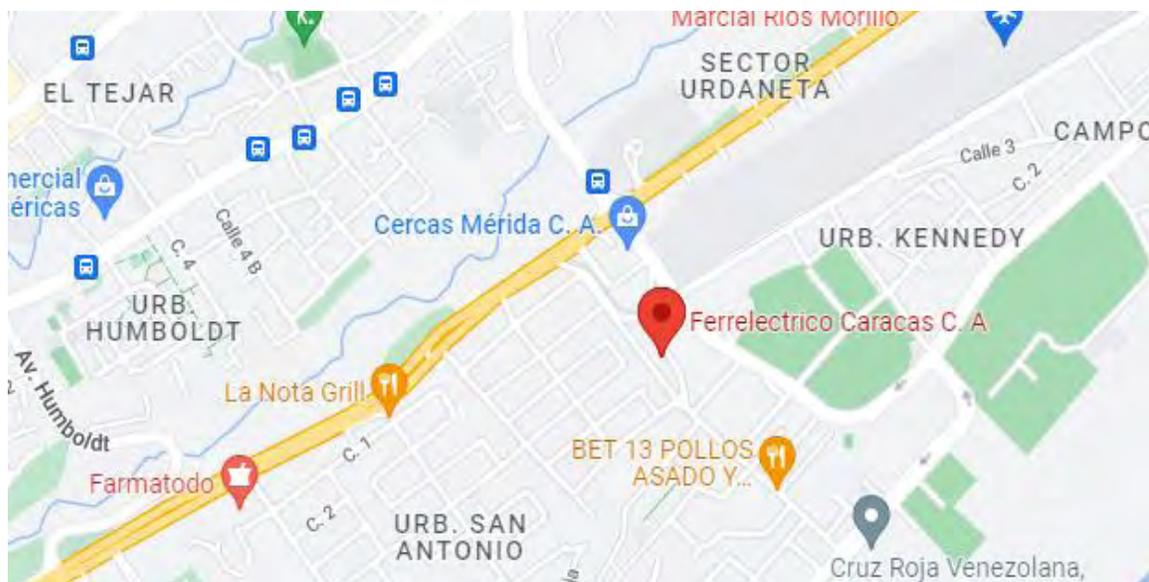


Figura 3.30: Ubicación de Ferrelectrico Caracas, Mérida

Una vez conocido los puntos donde se pueden adquirir las piezas mecánicas, tanto en forma individual como en formato de kit para CNC, se procede a mostrar un cuadro comparativo de piezas.

Tabla 3: Presupuesto de piezas mecánicas de la estructura

	Opción 1	Opción 2	Opción 3	Unidades
Varillas y ejes óptico de acero (kit de 15 piezas)	\$28,39	\$(19,94-45,1)	\$39,99	3
Cojinete de bloque de aleación de Zinc	\$8,99	\$(2,65\$-8,99)	\$14,99	2
Soporte de eje de riel de aluminio de 42x32x11mm	\$8,3	\$15,99	\$17,50	4
Varilla de tornillo de acero inoxidable de 200x8mm con tuerca	\$11,99	\$14,04	\$34.40	1
Buje deslizante 8x34,5mm	\$12,49	\$13,59	\$11,99	4
Acoplamiento de eje Flexible de 5x8mm	\$4,74\$	\$4,50	\$5,00	2

En cuanto a las piezas electrónicas del sistema. Todas fueron adquiridas en la opción 4, que es: **NEVADA ELECTRONICS, C.A.** Ubicados 2,2 Km de la facultad de medicina, específicamente en el centro, en la Avenida. 6, entre calles 23 y 24, Mérida. Es el sitio más surtido en cuanto a electrónica en la ciudad de Mérida.

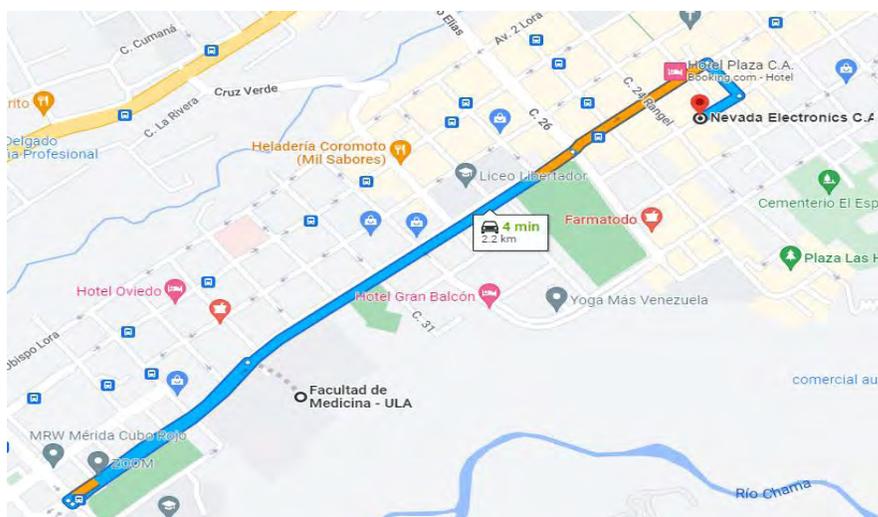


Figura 3.31: Ubicación de Electrónica Nevada, Mérida

El resto de piezas electrónicas fueron donadas o adquiridas del taller, como se muestran en el siguiente cuadro:

Tabla 4: Presupuesto de piezas electrónicas del sistema

	Opción 4	Donación	Taller de electrónica ULA	Unidades
Microcontrolador PIC18F4550	\$13,17	-		1
Módulo controlador A4988	\$5,29	-	-	3
DREMEL Modelo 275	\$25,53	-	-	1
Motores Paso a paso NEMA 17	\$24,99	-	-	1
Módulo FT. 232	\$5,00		-	1
Sensor de final de carrera	\$0,86	-	-	3
CNC SHIELD	\$4,99	-	-	1
Placa PCB 70x90mm	\$1,99	-	-	1
Módulo relé 125VAC-10A	\$1,99	-	-	1

Como se pretende mostrar si la construcción artesanal de una máquina CNC es factible, se procede a comparar valores del precio actual de máquinas CNC prefabricadas en rangos, que se muestran a continuación en la tabla 5. Ampliando la cobertura de búsqueda a no solo el mercado local, sino el global de todas las máquinas de control numérico que se encuentran en los comercios electrónicos más populares, para así, elaborar eficientemente un promedio de costo que se pueda comparar con el precio de elaboración del equipo, con las piezas elegidas para el sistema y así demostrar la factibilidad de su fabricación.

Tabla 5: Presupuesto de CNC prefabricada

	AMAZON	ALIEXPRESS	Mercado local	Unidades
CNC	(600,02-1676,09)\$	(383,52 – 453,36)\$	(500 – 1,200)\$	1

En efecto, existen diferentes costos en la construcción de un control numérico, pero para matizar, se seleccionaron a partir de su funcionalidad, es decir, los presupuesto mostrados en la tabla 5, son de sistemas de control de fresadoras, plotter, cortadoras de uso doméstico. Ya que industrialmente, la fabricación de estas máquinas es más costoso. Siguiendo el mismo orden de ideas, ahora se procede a comparar el costo comercial de la máquina, con el costo de elaborarla artesanalmente con la utilización de materiales reciclables como pueden ser maderas del tipo MDF para la construcción de su estructura.

Tabla 6: Presupuesto seleccionado para este proyecto

COMPONENTES	COSTO COMERCIAL	COSTO DE ELABORARLA
AMAZON	\$600,02	
ALIEXPRESS	\$383,52	
MERCADO LOCAL	\$500	\$(28,39 x3) + \$(5,29 x3) + \$25,53 + \$24,99 + \$(0,86 x3)
PROMEDIO	\$494,51	+ \$4,99 + \$1,99 + \$1,99 + \$19,99
SUBTOTAL	\$494,51	
ENVÍO	10%	
TOTAL	\$543,96	\$183,10

Luego de una revisión exhaustiva a través de varios sitios web, y de observar las distintas opciones de materiales y costos de los mismos para la construcción de una CNC, en comparación a su adquisición directamente en línea o en el mercado local, así como también, tomando en cuenta el recargo adicional del envío (un aproximado de 10% por envío), este estudio demuestra lo factible que es ensamblar la CNC, en cuanto al punto de vista económico, ya que este mismo representa solo el 33,66% de la inversión de comprar una máquina prefabricada, a pesar de agregar \$19,99 de construcción adicional como lo son arandelas, tuercas y tonillos. Para precisar esta afirmación, se realiza el siguiente cálculo, donde 'X', representa una proporción escalar del costo de elaboración de la máquina, con respecto a costo comercial de una misma prefabricada. 'Y' representa el porcentaje de ahorro con la construcción de la máquina de forma artesanal.

$$\left(\begin{array}{l} 543,96 \rightarrow 100\% \\ 183,10 \rightarrow X \end{array} \right)$$

$$X = \frac{183,10 \times 100}{543,96} = 33,66\%$$

$$Y = 100 - X = 100 - 33,66 = 66,34\%$$

CAPÍTULO IV

SOFTWARE

A continuación, se describe la programación del microcontrolador y la interfaz gráfica del sistema, con la finalidad de dar una idea clara del funcionamiento del sistema.

Básicamente, el controlador de las máquinas CNC recibe instrucciones del usuario y mediante un software, convierte esas instrucciones en señales eléctricas destinadas a activar los motores que, a su vez, pondrán en marcha el sistema de accionamiento [31].

4.1 CONTROL DE MOVIMIENTO

Todas las máquinas CNC comparten una característica en común: tienen dos o más direcciones programables de movimiento llamadas ejes. Un eje de movimiento puede ser lineal (en línea recta) o rotatorio (en una trayectoria circular). Una de las primeras especificaciones que implica la complejidad de una máquina CNC es la cantidad de ejes que tiene. En términos generales, a mayor cantidad de ejes, mayor complejidad [31].

Los ejes de una máquina CNC son un requisito para generar los movimientos necesarios para el proceso de fabricación. Para este caso, los ejes ubican la herramienta sobre el orificio a mecanizar y realizan la operación de posicionar la herramienta. Los ejes se denominan con letras. Los nombres más comunes de los ejes lineales son 'X', 'Y' y 'Z' [31].

El control de movimiento puede realizarse mediante dos sistemas, que pueden funcionar individualmente o combinados entre sí:

1. **Valores absolutos**, donde las coordenadas del punto de destino son referidas al punto de origen de coordenadas. Se usan las variables 'X' e 'Y' (medida del diámetro final) y 'Z' (medida en dirección paralela al eje de giro del husillo) [31].
2. **Valores incrementales**, donde las coordenadas del punto de destino son referidas al punto actual. Se usan las variables I y J (distancia radial) y W (medida en dirección paralela al eje de

giro del husillo) [31]. En efecto, para este caso, se usan valores absolutos para el control de movimiento de los ejes 'X, Y, Z'. Y valores incrementales para las distancias radiales I y J.

4.2 BLOQUE DE COMANDO

Para controlar el movimiento de los motores paso a paso del sistema, es necesario que recibir datos geométricos en forma de programa, para que el sistema de encargo de interpretarlos posteriormente. Para el formato de dicha cadena, la ISO (Organización Internacional de Normalización) estandarizó un lenguaje de programación para maquinaria CNC en su norma 6983 [31]. Este lenguaje se compone de un bloque de palabras con el siguiente orden:

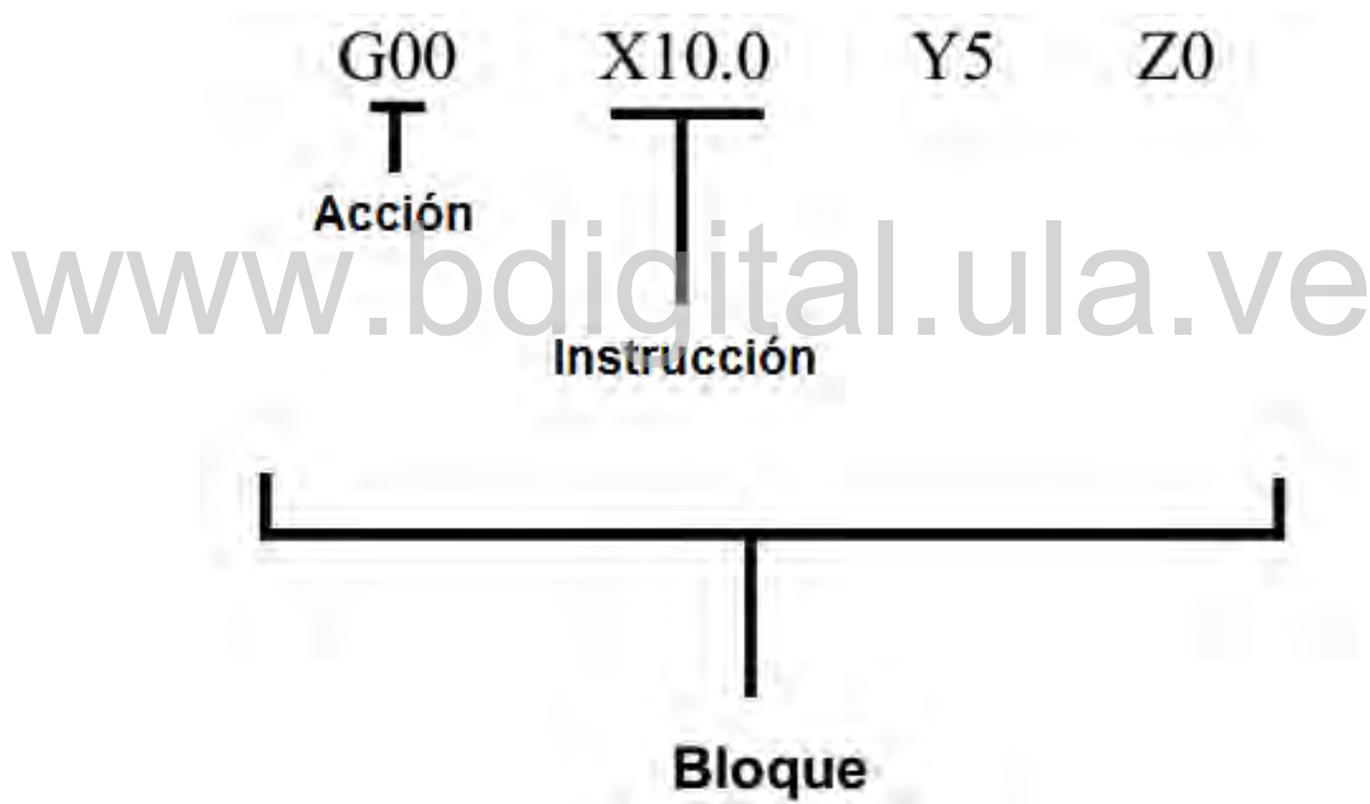


Figura 4.1: Formato de frases conformadas en un comando

Dicho bloque de comando es el lenguaje en código G que es traducido por el software del sistema de la máquina para su respectivo accionamiento, y se descompone en acción e instrucción, los cuales cumplen con distintas funciones, que son enunciadas de la siguiente manera:

4.2.1 Acción

La acción describe las funciones de movimiento de la máquina mientras que el código M describe las funciones misceláneas que se requieren para el mecanizado de la pieza, pero que no corresponden a los movimientos de la máquina (por ejemplo, arranque y detención del husillo, cambio de herramienta, refrigerante, detención del programa) [31].

Tabla 7: Funciones de movimiento del código G

CÓDIGOS G
G00: Posicionamiento rápido (sin tallado)
G01: Interpolación Lineal (Tallando)
G02: Interpolación circular (horaria)
G03: Interpolación circular (anti-horaria)

En efecto, los comandos mostrados en la tabla 7, representan las acciones principales y necesarias para el movimiento de la máquina.

4.2.2 Instrucción

Junto con la palabra de acción, cada código contiene instrucciones de dirección de las variables, identificadas con otras letras y definidas de la siguiente manera: X, Y y Z para el movimiento de los ejes; I y J para la localización del centro de un arco. Junto a la variable, una serie de números que, para el caso de X, Y, Z corresponden a las coordenadas absolutas del punto destino con respecto al origen y para el caso de I y J, indican la distancia que hay entre el punto inicial y el centro de la circunferencia que describe el arco, descrito en coordenadas incrementales [31].

4.3 CONTROLADOR CNC

Este es el componente clave que interpreta el programa y acciona la serie de comandos en orden secuencial. A medida que lee el programa, el controlador activa las funciones apropiadas de la máquina e impulsa el movimiento de los ejes para su correspondiente tallado. Por lo que es el elemento más importante de la máquina [31].

Un controlador PIC18F4550 es el encargado de interpretar el bloque de comando en su respectivo orden y esto lo logra a través de una serie de funciones, que son explicadas a través de una serie de diagramas de flujo y una breve descripción del funcionamiento de cada una de ellas, como se expresa a continuación:

4.3.1 Función Principal

Es la función que se encarga de recibir una serie de caracteres, utilizando el protocolo de comunicación UART (transmisor-receptor asíncrono universal), y al ser una comunicación asíncrona, el transmisor y el receptor no comparten una señal de reloj común. Dichos caracteres pasan por un proceso de validación, para así, convertirlos en una cadena. Este proceso se realiza de la siguiente manera:

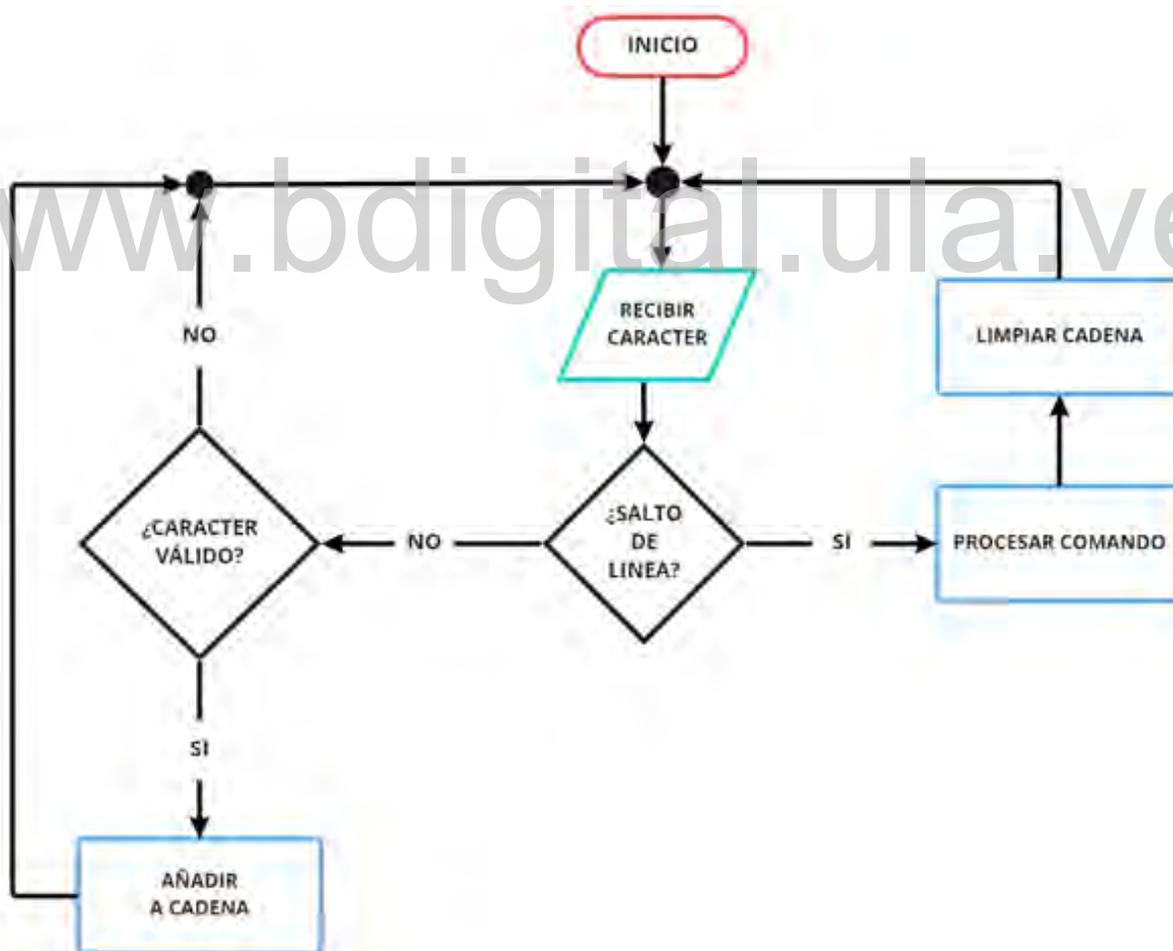


Figura4.2: Diagrama de flujo de la función principal

Al trabajar con una comunicación asíncrona, el registro SPBRG, tiene la función de indicarle al hardware la velocidad de transmisión de los datos. La misma debe tener una relación con el bit BRGH, que se encuentra dentro del registro TXSTA. Sucede pues, que el bit BRGH es el encargado de asignar el tipo de comunicación asíncrona en alta o baja velocidad. Para este caso BRGH está configurado para baja velocidad, por lo que el rango de valores para la transmisión debe ser menor o igual a 19200 baudios. En efecto, 9600 es un valor que permite una transmisión de datos correcta para recibir un bloque menor de 128 caracteres.

Cabe considerar que, para acomodar el dato enviado, este pasa por un proceso de validación, donde elimina los espacios del bloque y lo restringe para recibir solo los caracteres correctos. A su vez, estos datos se van almacenando en una cadena, para luego ser procesados y convertidos en valores de movimiento, y así, realizar la instrucción del bloque de comando en la función de procesar la línea.

```

1  #include <xc.h>
2  #include "USART_libreria.h"
3
4  void USART_Init(long BAUD) {
5      //Configuración de los pines
6      TRISBbits.RC6 = 0;      //RC6 = Tx -> Salida
7      TRISBbits.RC7 = 1;      //RC7 = Rx -> Entrad
8
9      //Baudios
10     SPBRG = (unsigned char) (((20000000L/BAUD)/64)-1);
11
12     //Configuración
13     TXSTAbits.BRGH = 0;      //Low Speed
14     TXSTAbits.SYNC = 0;      //Asincrono
15     RCSTAbits.SPEN = 1;      //Habilitar Tx y Rx
16
17     //Transmisión
18     TXSTAbits.TX9 = 0;      //8 bits sin pariedad
19     TXSTAbits.TXEN = 1;      //Activamos transmisión
20
21     //Recepción
22     RCSTAbits.RC9 = 0;      //8 bits
23     RCSTAbits.CREN = 1;      //Activamos recepción
24 }
25

```

Figura4.3: Configuración de datos para la comunicación asíncrona

La figura mostrada, evidencia la configuración de los bits para el registro TXSTA encargado de la transmisión de datos.

4.3.2 Función para procesar la línea

Esta función se encarga de entender el bloque, y decidir la acción correspondiente al comando dependiendo del caso, para ello, sigue la siguiente secuencia de flujo:

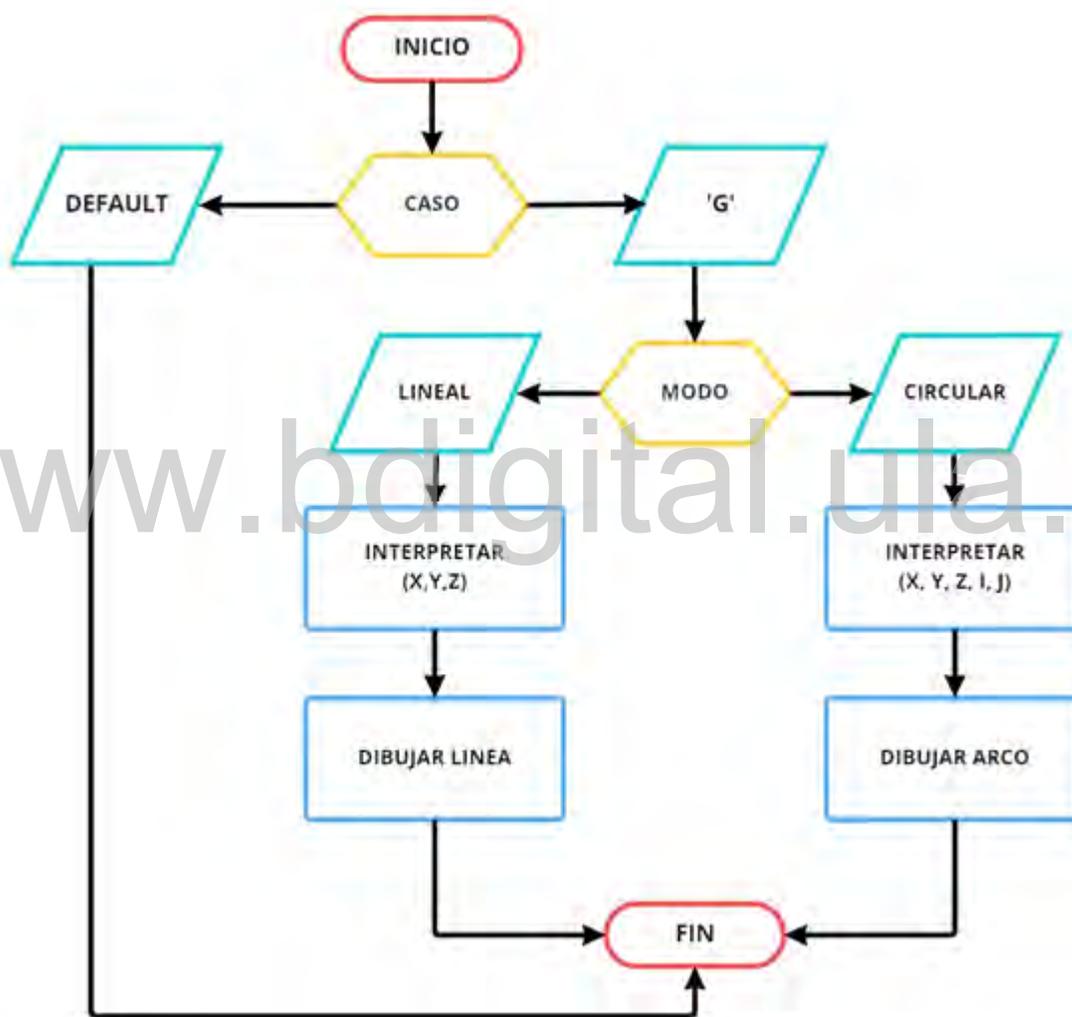


Figura4.4: Diagrama de flujo de la función para procesar línea

Donde 'G' corresponde a las acciones de código G señaladas en la tabla 7 y (X, Y, Z, I y J) constituyen a los valores de movimiento de la instrucción. En otras palabras, una vez recibida la cadena de caracteres, se delimita la acción del comando para su respectiva función

de movimiento, y se escanea la posición de las instrucciones a través de la sentencia *strchr*, función incluida en la librería *<string.h>*, la cual precisa la posición de un dato en la cadena de caracteres.

Una vez fijadas las posiciones de las sentencias, el dato indicador de la coordenada, se convierte en un valor flotante a través de la función *atof*, que es una palabra reservada del lenguaje C, que transforma los datos en una variable de tipo decimal o punto flotante, valor que será escalado y convertido en una cantidad de pasos que pueda ser entendido por los motores, para que se ejecute el correcto movimiento de los mismos. De este proceso se encarga la función para dibujar la línea.

4.3.3 Función para dibujar la línea

Para dibujar la línea, el programa procesa las coordenadas vectoriales (X, Y, Z), las opera con la coordenada anterior, con el fin de hallar su módulo y dirección para realizar el proceso de interpolación lineal entre dos puntos, para posteriormente accionar los motores en el orden adecuado.

A nivel de sistemas de control numérico, el concepto de interpolación lineal hace referencia a las trayectorias lineales que son calculadas mediante puntos iniciales y finales, para posteriormente crear una trayectoria de fresado en el plano o en el espacio de trabajo. En este mismo orden de ideas, la interpolación entre los ejes de una CNC es muy necesario, especialmente en los ejes 'X' e 'Y'. Por ello, se asocian sincrónicamente a través de una cuenta, la cual controla a su vez la velocidad de los motores por separado.

El siguiente diagrama indica la trayectoria de flujo de la función para dibujar la línea, Donde 'K' representa el contador del paso actual del módulo de mayor magnitud entre 'X' y 'Y'. En cuanto a CUENTA, se interpreta como la sumatoria del módulo de menor magnitud en cada iteración, cuando este valor supera o iguala al módulo mayor, se considera que debe dar un paso en el eje de menor magnitud, es decir; la variable CUENTA, corresponde a la pendiente en relación a los módulos mayor y menor respectivamente.

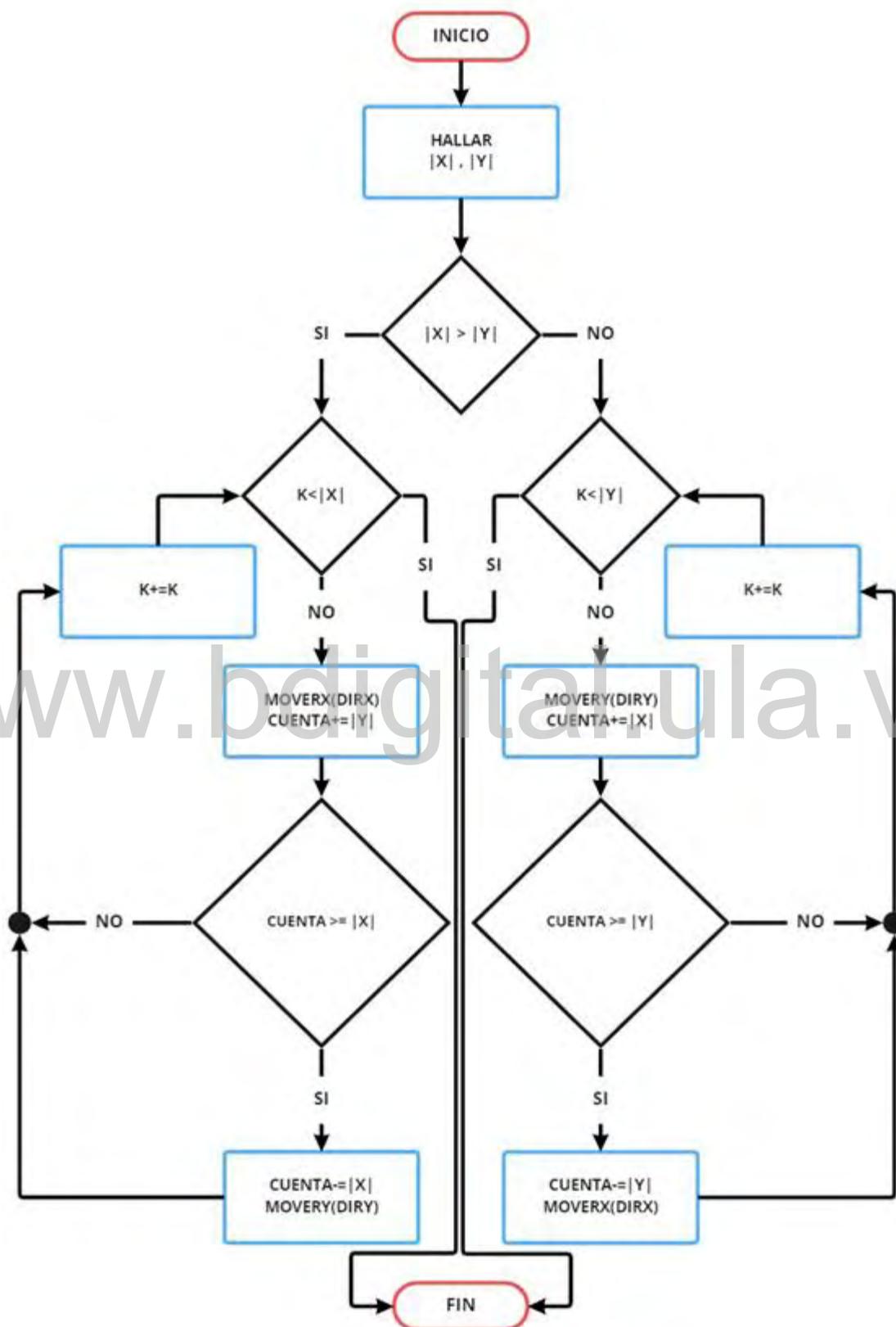


Figura4.5: Diagrama de flujo de la función para dibujar la línea

A modo de orientación, se modela el flujo de la función. Por ejemplo, si se requiere recorrer 7 pasos en dirección horizontal y 3 pasos en dirección vertical, la función dibujar la línea realiza la siguiente tabla:

Tabla 8: Ejemplo ilustrativo de la función dibujar la línea

Iteración	Recorrido X	Recorrido Y	Cuenta
1 ^{era}	1	0	3
2 ^{da}	2	0	6
3 ^{ra}	3	1	2
4 ^{ta}	4	1	5
5 ^{ta}	5	2	1
6 ^{ta}	6	2	4
7 ^{ma}	7	3	0

Por lo tanto, se demuestra en la siguiente gráfica, el comportamiento de la línea que se traza en el ejemplo mostrado:

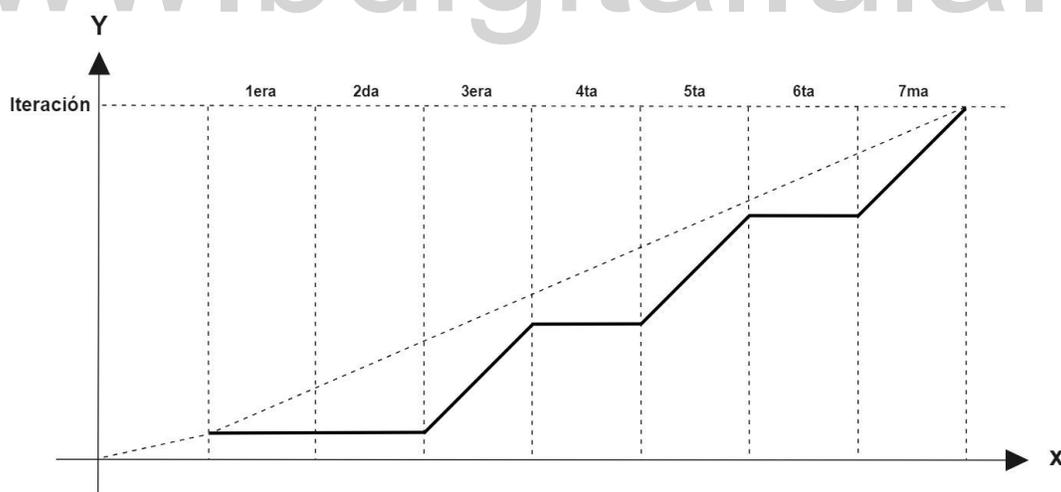


Figura4.6: Trayectoria de una línea descrita por una CNC

Evidentemente, la línea mostrada en el diagrama, representa como la función dibujar línea describe la trayectoria que traza una CNC, donde se puede notar que no describe un tramo recto. Sin embargo, luego de escalar las coordenadas de trayectoria, provoca que el curso que sigue la línea sea percibido como una línea recta.

4.3.4 Función para dibujar arco

Para el caso de la función dibujar arco. Se realiza a través de esta secuencia:

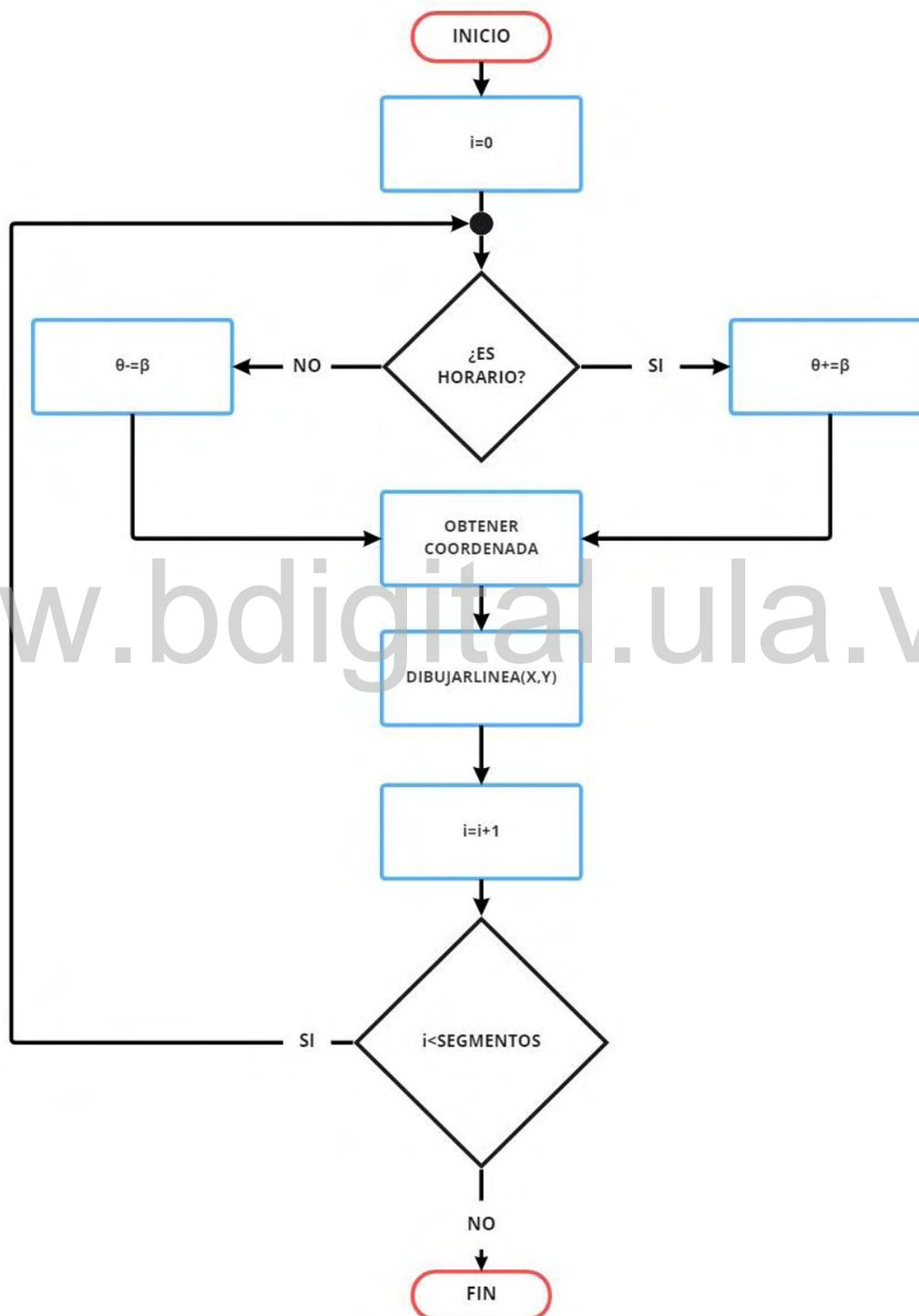


Figura4.7: Diagrama de flujo de la función para dibujar arco

El diagrama de flujo de la función para dibujar arco se interpreta en esta secuencia: Se determina la dirección en la cual se va a recorrer la circunferencia, para operar las coordenadas y obtener una serie de segmentos que corresponden al arco a dibujar, estos a su vez, se van a unir a través de la función para dibujar línea, a este método se le conoce con el nombre de proceso de interpolación circular entre dos puntos, o sea que funciona de la forma que se muestra a continuación:

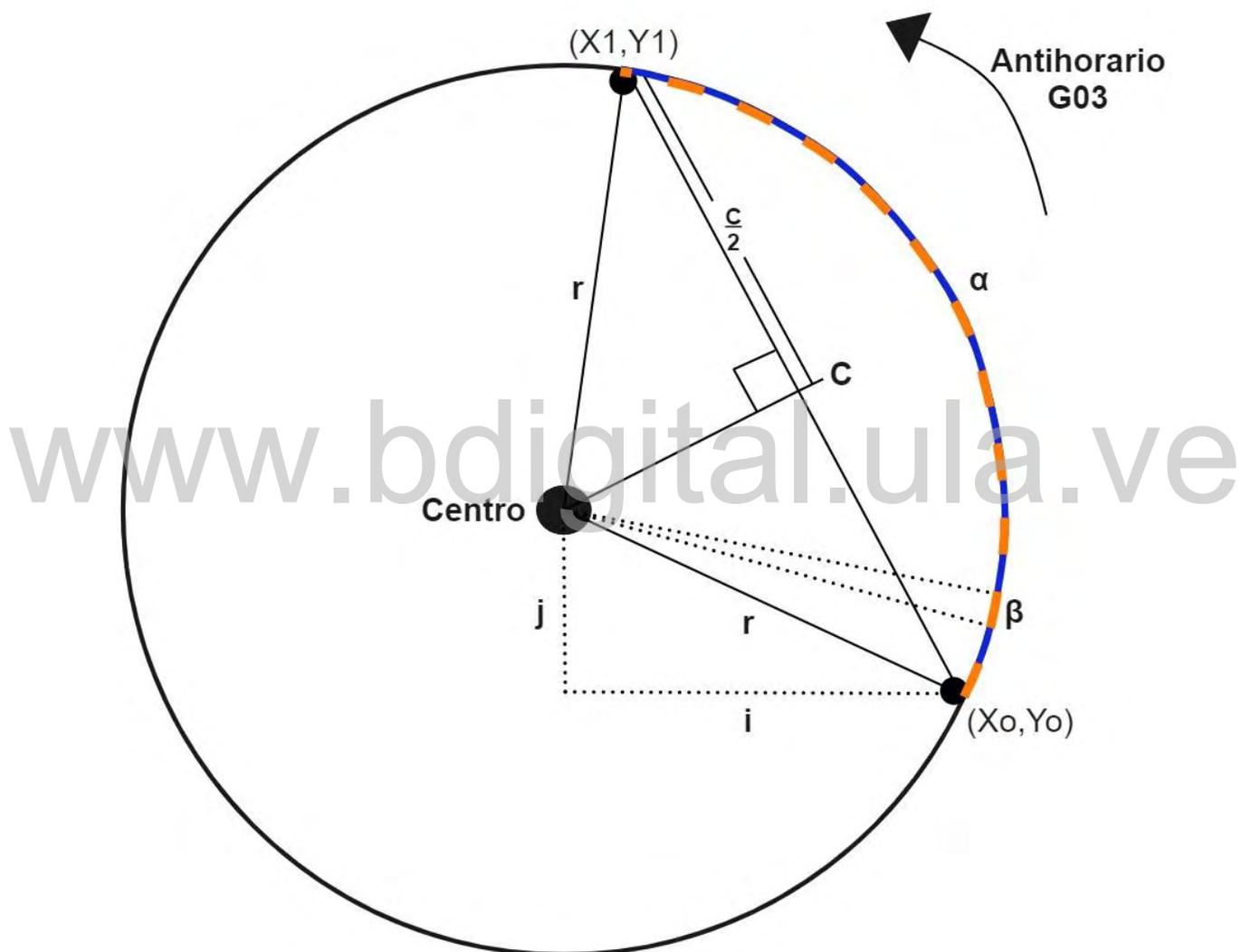


Figura4.8: Representación gráfica del proceso de interpolación circular

La interpolación circular entre un punto inicial (X_0, Y_0) y un punto final (X_1, Y_1) se plantea como una relación geométrica entre dos puntos que están sobre una circunferencia que describe la trayectoria de un arco. En general, se forma un triángulo isósceles (un triángulo

que tiene, al menos, dos lados de igual longitud), con el radio (r) como definido como la longitud de los catetos en conjunción con el centro de la circunferencia y la cuerda (C) como la hipotenusa. De este modo, es posible formar dos triángulos rectángulos semejantes, como se muestra en la figura 4.8. A fin de, establecer una fórmula que permita relacionar el ángulo α con los valores conocidos. En efecto, Las razones trigonométricas de un ángulo α son las razones obtenidas entre los lados de un triángulo rectángulo, es decir; el seno de un ángulo α se define como la razón entre el cateto opuesto y la hipotenusa. Lo que se expresa de este modo:

$$\text{Sen}(\alpha) = \left(\frac{\text{Cateto Opuesto}}{\text{Hipotenusa}} \right)$$

En resumidas cuentas, es posible calcular el ángulo α despejando y sustituyendo en la ecuación por los valores correspondientes. A modo de orientación, resulta claro que el análisis de expresa de esta forma:

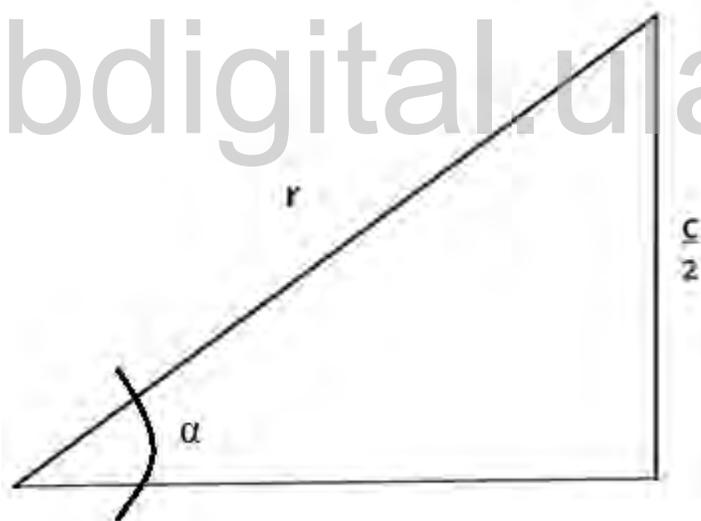


Figura4.9: Triángulo rectángulo formado por la cuerda y el radio para el proceso de interpolación circular

La figura mostrada, evidencia de manera sencilla como queda expresado el ángulo α matemáticamente:

$$\alpha = \text{Arcsen} \left(\frac{C}{2R} \right)$$

Sin embargo, dicho ángulo obtenido, representa solo el ángulo que forma el triángulo rectángulo. Resulta claro que el verdadero ángulo α es el que forman ambos triángulos rectángulos, por lo que, la fórmula final para hallar el valor de α , debe escalarse. Dicho de otro modo, la fórmula se expresa a continuación:

$$\alpha = 2 \times \text{Arcsen}\left(\frac{C}{2R}\right)$$

El ángulo α representa un trayecto circular descrito en radianes y los valores de cuerda y radio son siempre positivos, lo que hace que su dominio se vea restringido al siguiente rango: $[0, \pi]$. Por consiguiente, hay que tomar en cuenta la posición del punto inicial en radianes considerando la distancia del mismo con respecto al centro de la circunferencia (i, j). Esto con el fin de que α sea capaz de reproducir ángulos que sobrepasen los 180° . Evidentemente, los usos de la función atan2 en el lenguaje C facilitan esta labor ya que esta sentencia resuelve gran parte de la problemática ya expresada, y lo hace de esta manera:

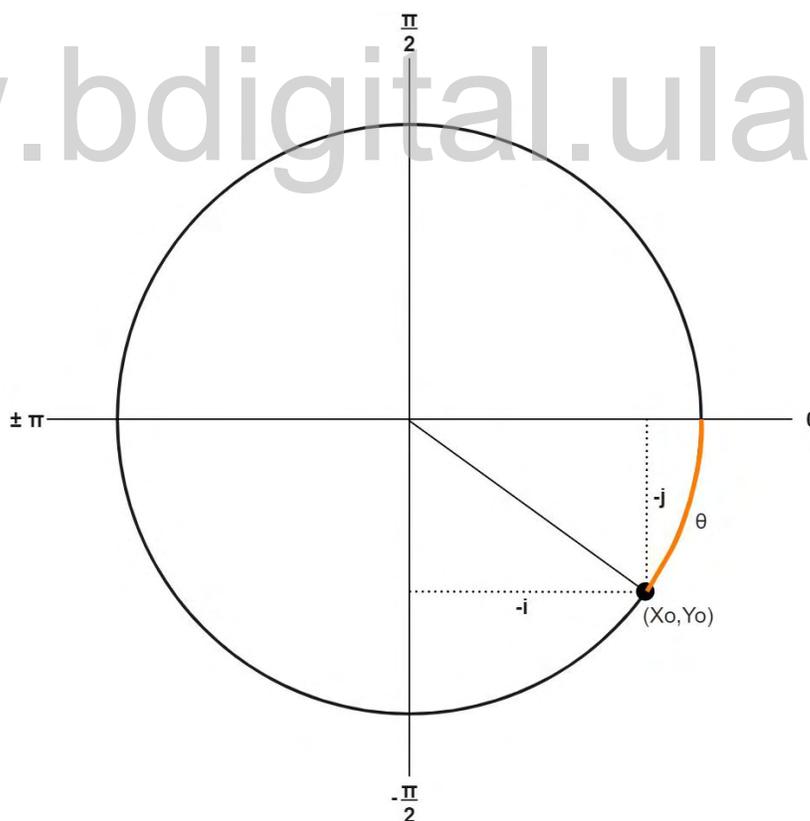


Figura4.10: Ángulo θ que relaciona las distancias del punto inicial con el centro de la circunferencia

Dicho de otro modo, el ángulo θ , también denotado como “angulosiguiente”, describe el ángulo formado en relación a las distancias del punto inicial con el centro de la circunferencia. En este sentido, la tangente de θ relaciona las razones obtenidas entre los catetos, lo que se expresa de esta manera:

$$\text{Tan}(\theta) = \left(\frac{\text{Cateto Opuesto}}{\text{Cateto Adyacente}} \right)$$

En este mismo orden de ideal, es posible calcular el ángulo θ despejando y sustituyendo en la ecuación por los valores correspondientes. A modo de orientación, resulta claro que el análisis se exprese de esta forma:

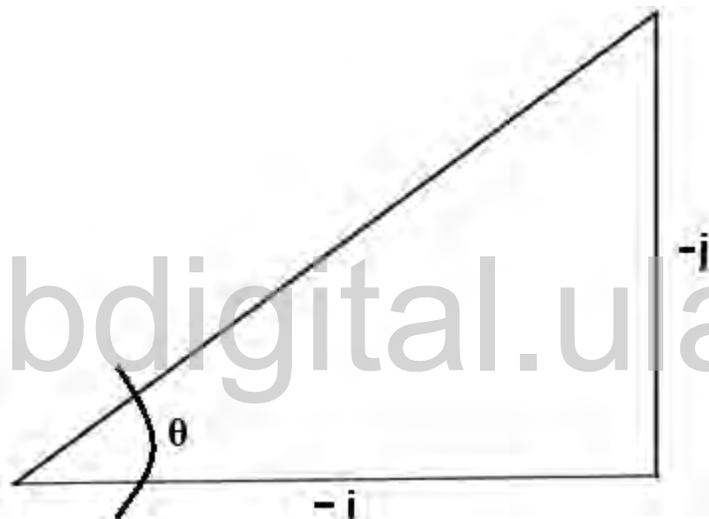


Figura4.11: Triángulo rectángulo formado por las distancias “i” y “j” correspondientes a las longitudes del punto inicial al centro de la circunferencia que describe el arco

La figura anterior evidencia de manera sencilla y eficaz como queda expresado el ángulo θ matemáticamente:

$$\theta = \text{Arctan} \left(\frac{-j}{-i} \right)$$

En definitiva, una vez conocido el valor de θ , salta a relucir que su dominio oscila en el rango de: $(-\pi, \pi]$, lo que se manifiesta como una solución para denotar las consideraciones en los que α no se encuentra definido.

4.3.4.1 Consideración 1: Donde el punto inicial se encuentra alineado verticalmente con el centro de la circunferencia, es decir; si $i=0$. Por consiguiente, se presentan 4 casos, donde el valor de la distancia “j” es mayor a 0, y “j” es menor a 0; Tanto para el arco horario, como el antihorario (G02 y G03 respectivamente).

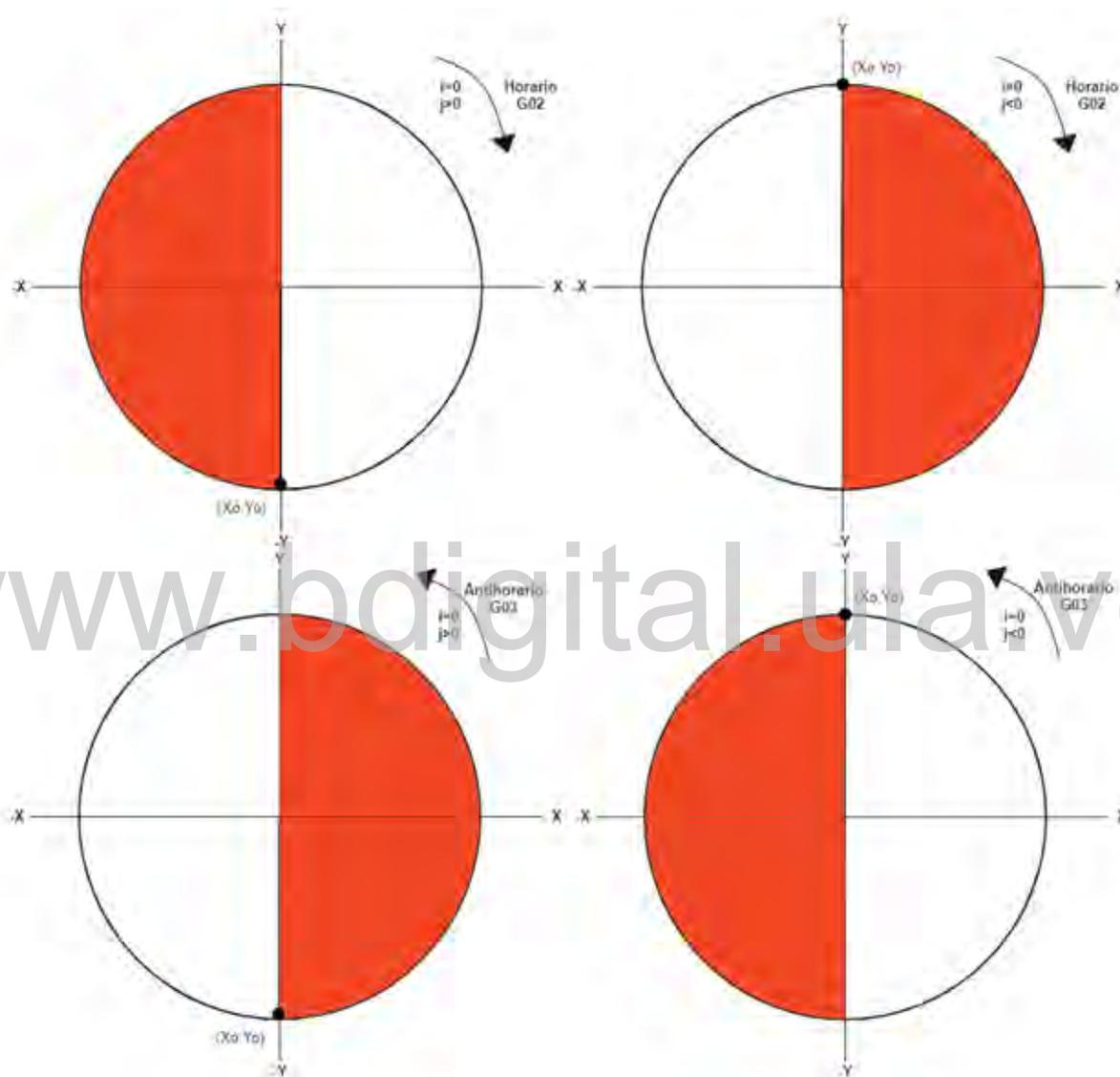


Figura4.12: Diagramas con las consideraciones del punto inicial alineado verticalmente con el centro de la circunferencia

El diagrama mostrado, ilustra como los ángulos agudos (es el ángulo formado por dos semirrectas con amplitud en el rango de: $[0, \frac{\pi}{2}]$, es decir, mayor de 0° y menor de 90°), rectos

(es el ángulo de amplitud igual a $\frac{\pi}{2}$, equivalente a 90°), y obtusos (es el ángulo formado por dos semirrectas cuya amplitud se encuentra el rango de: $[\frac{\pi}{2}, \pi]$, es decir, mayor de 90° y menor de 180°), que se forman entre el punto inicial y el punto final, no suponen una problemática, ya que estos valores se encuentran dentro del dominio de α .

4.3.4.2 Consideración 2: Donde presenta la solución a los ángulos reflejo (es uno que mide más de 180° pero menos de 360°).

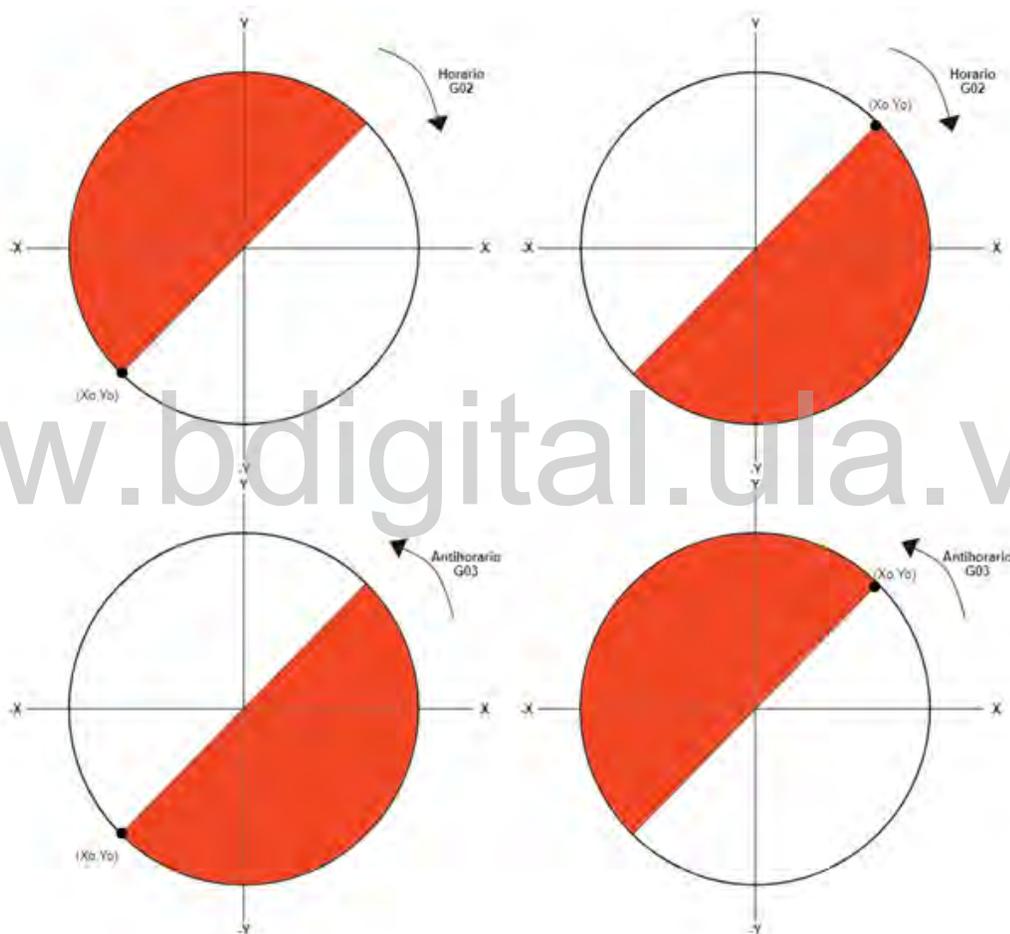


Figura 4.13: Diagramas con las consideraciones de los ángulos reflejo

En los diagramas de la figura 4.12 y 4.13, se encuentra expresado la trayectoria que logra describir el ángulo α debido a su dominio. Es necesario un ajuste en la fórmula en los casos contrarios. Visto de esta forma, la expresión matemática se define de la siguiente manera:

$$\alpha = 2\pi - \left(2 \times \text{Arcsen} \left(\frac{C}{2R} \right) \right)$$

Esta operación invierte el ángulo α de manera que consigue su ángulo conjugado, que representa los valores donde α no se encuentra definido.

Una vez definido α en todo el dominio de una circunferencia, es necesario segmentar su valor relacionándolo con la longitud de su arco. Así pues, se divide el ángulo α en pequeños tramos denotados como β . Esto con el fin de trazar la superficie del arco a través de la función dibujar línea. En efecto, β se le suma o resta al ángulo θ dependiendo de si la acción del bloque es en sentido horario o antihorario y se realiza una estructura de repetición hasta que el valor de θ sea igual a la cantidad de segmentos en que β compone a α .

4.3.5 Función para mover los motores y reposo del motor

Las funciones anteriores, llaman adecuadamente a la función para hacer girar los motores. Al recibir un pulso eléctrico como señal se activa el campo magnético giratorio del rotor alrededor del estator. Cuando esta corriente fluye a través de las bobinas del estator, se crea un campo magnético a su alrededor. Los flujos magnéticos particulares tienen una dirección e intensidad que dependen de la intensidad y la dirección de la corriente que fluye a través de una bobina determinada. En definitiva, la señal recibida pone en funcionamiento el giro de los motores debido al campo magnético que se crea en el rotor del mismo, tanto para 'X', 'Y' o 'Z'. La dirección va a estar determinada a partir de la diferencia entre la posición actual y final en coordenadas vectoriales, si este valor es positivo, el motor va a girar en el sentido positivo del eje de coordenadas y viceversa.

Una vez los motores hayan recibido el respectivo pulso, se activa la función de reposo motor, la cual es un retardo que les da el tiempo suficiente para producir su rotación, que para los motores de 1800 RPM y 200 pasos por vuelta es de 166,667 μ S. Una vez transcurrido ese retardo, la función reposo motor posiciona el controlador de pasos del motor en una señal de bajo para recibir el siguiente pulso. Dichas funciones se cumplen la cantidad de veces que represente la Acción en el bloque de comandos multiplicado por un escalar llamado pasos por milímetro, el cual es un escalar que tiene el valor de 200 unidades, lo que indica que cada

Cabe destacar que para simular los drivers A4988, se usan los componentes L297 y L298 en serie para su respectivo modelado, ya que ese componente no se encuentra en la lista de sensores estándar del desarrollador de diseños electrónicos Proteus versión 8.11.

Tabla 9: Organigrama de conexión externa del PIC18F4550

PIC18F4550	FUNCIÓN	CONEXIÓN	DESCRIPCIÓN
RA0	Controlador	<i>Step</i> A4988-Z	Controla el número de pasos del motor del eje Z
RA1	Controlador	<i>Dir</i> A4988-Z	Controla el sentido de giro del motor del eje Z
RA2	Modificador	<i>Half</i> A4988-Z	Modifica el grado de giro del motor del eje Z
RA6	Oscilador 1	CRYSTAL	Gestiona la frecuencia de oscilación del PIC
RA7	Oscilador 2	CRYSTAL	Gestiona la frecuencia de oscilación del PIC
RB0	Controlador	<i>Step</i> A4988-Y	Controla el número de pasos del motor del eje Y
RB1	Controlador	<i>Dir</i> A4988-Z	Controla el sentido de giro del motor del eje Y
RB2	Modificador	<i>Half</i> A4988-Z	Modifica el grado de giro del motor del eje Y
RC6	Transmisor	Ordenador	Transmite los datos de forma asíncrona
RC7	Receptor	Ordenador	Transmite los datos de forma asíncrona
RD0	Controlador	<i>Step</i> A4988-X	Controla el número de pasos del motor del eje X
RD1	Controlador	<i>Dir</i> A4988-Z	Controla el sentido de giro del motor del eje X
RD2	Modificador	<i>Half</i> A4988-Z	Modifica el grado de giro del motor del eje X
RE3	Alimentación	VCC	(2.5 – 5.5)V

4.5 VELOCIDAD DE CONTROL

Los motores de paso nema 17 cuentan con un ángulo de giro de 1,8° (200 pasos / 1 vuelta), además; cuentan con 6 resoluciones de pasos diferentes: paso completo, 1/2-paso, 1/4-paso, 1/8-paso, 1/16-paso y 1/32-paso.

Para este caso, se trabaja a pasos completos para no forzar a que los motores trabajen por debajo del 50% de su capacidad nominal, ya que, si los motores se mantienen trabajando al 70% de su capacidad, preserva su periodo de vida útil [18].

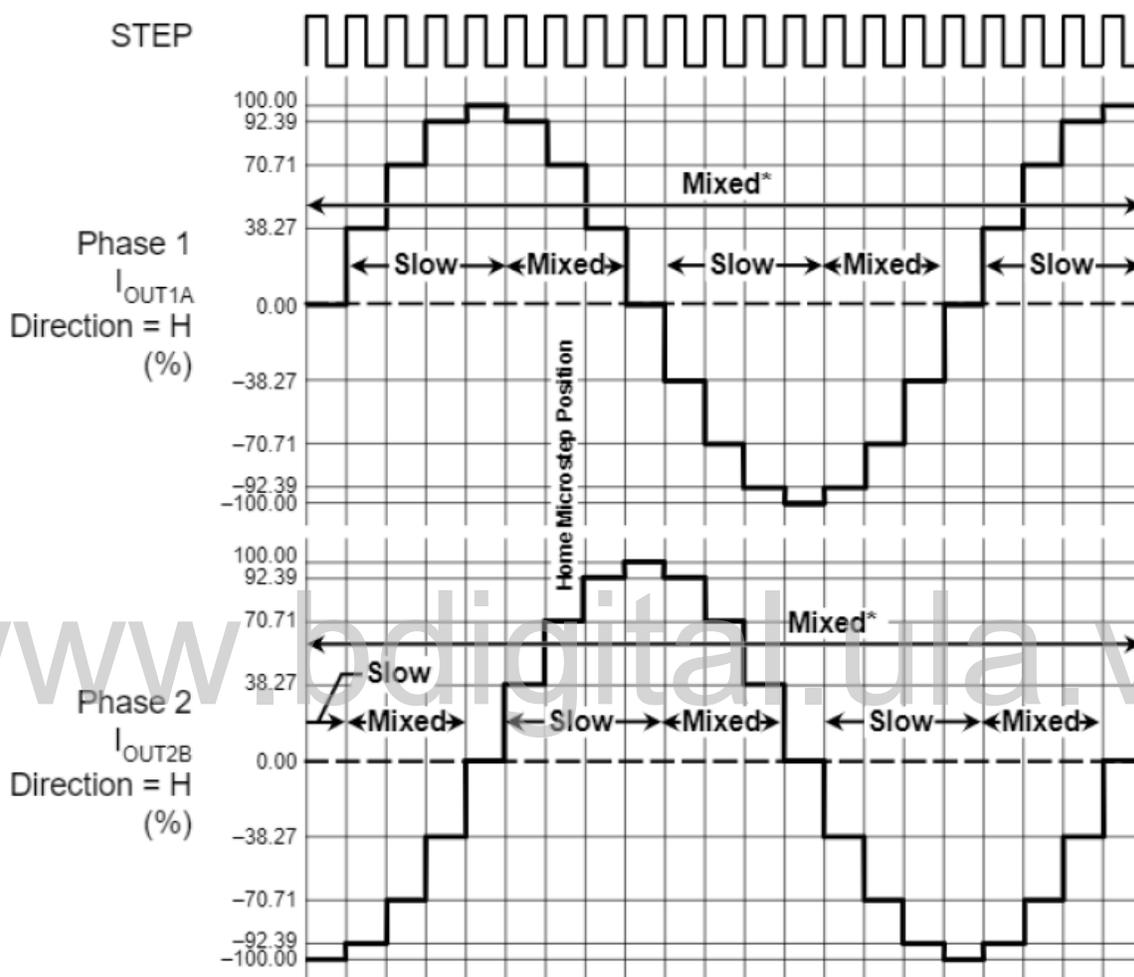


Figura 4.15: Modo de decaimiento para incrementos de $\frac{1}{4}$ pasos [18]

4.6 INTERFAZ GRÁFICA

La interfaz gráfica de usuario, es un programa informático que actúa de medio de interacción con el usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Desarrollada en Unity, consiste principalmente en proporcionar un entorno visual sencillo para permitir la interacción con el sistema y la simulación como acompañamiento visual. Siguiendo el mismo orden de ideas, la interfaz se compone de los siguientes elementos:

Ícono del programa: Simboliza el archivo que se ejecuta y viene representado por el siguiente logo:



Figura 4.16: Logo de Val CNC

En efecto, el logo mostrado en la figura cumple con la función de representar con una imagen su contenido.

Pantalla de carga: Una vez ejecutada la aplicación, se muestra primeramente una marca de agua de la plataforma de desarrollo Unity. Seguido de este, se muestra un plano secuencia del logo de ValCNC haciendo un giro de 360° sobre sí mismo. Como se muestra a continuación:

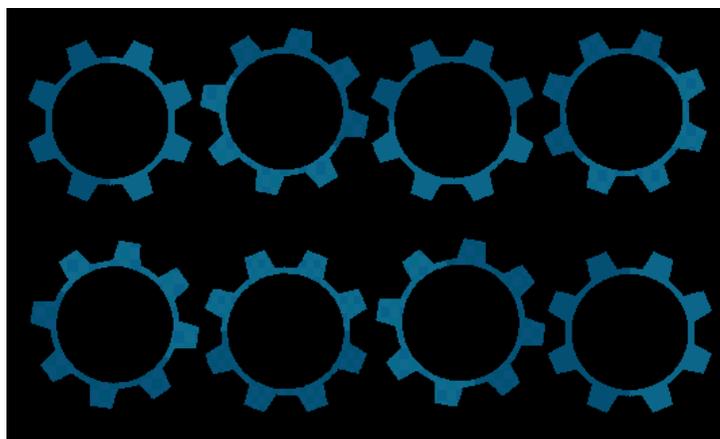


Figura 4.17: Pantalla de carga de la aplicación

Una vez transcurrido el plano secuencia mostrado en la figura anterior, aparece la animación del logo de Val CNC con la broca metálica de corte girando. Para así dar paso a la pantalla principal de la interfaz donde se encuentra la barra de tareas.

Esto se logra mediante el menú de animación de Unity, que permite establecer fotogramas clave (*keyframe*) de los atributos que componen la imagen.

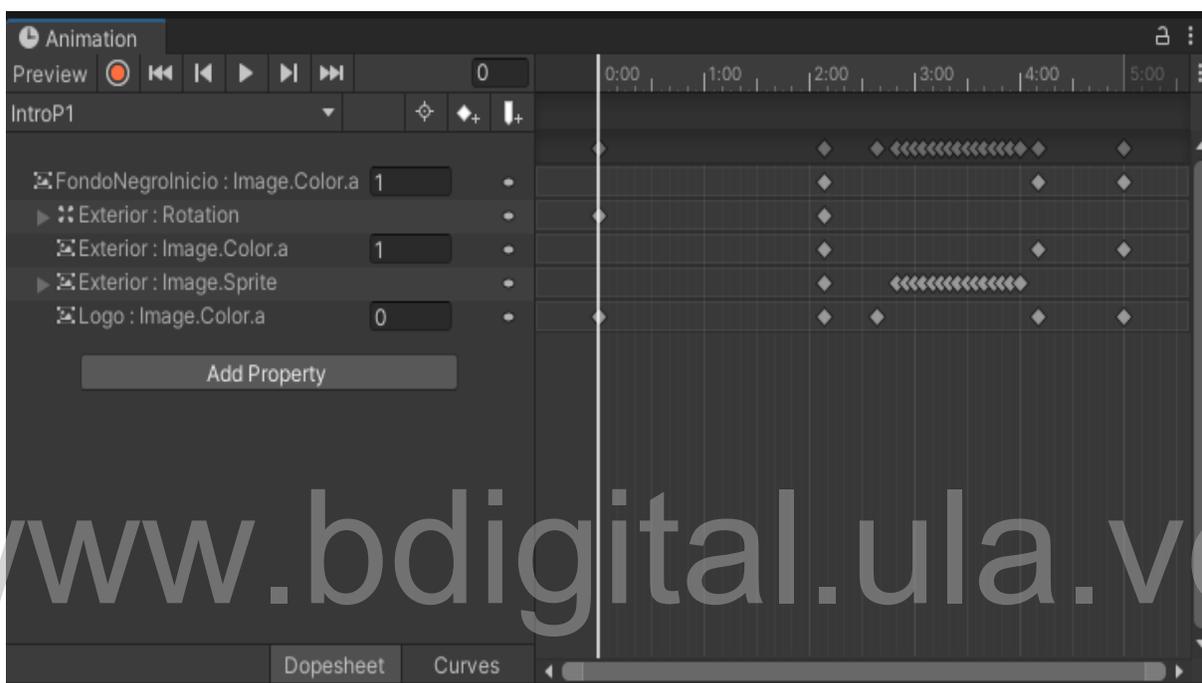


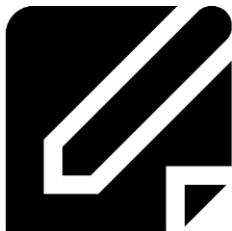
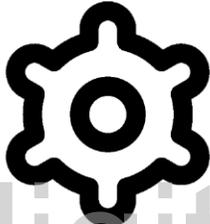
Figura 4.18: Menú de animación de la plataforma Unity

En este mismo orden de ideas, en la figura 18, se muestra la línea de tiempo de la animación, la cual altera atributos de varias imágenes para lograr el efecto visual en el transcurso de 5 segundos, secuenciados de la siguiente manera:

1. Primero se altera la rotación de la imagen correspondiente a la tuerca del logotipo de la aplicación, haciéndola girar durante 2 segundos.
2. Luego, hace aparecer el resto del logo, alterando la transparencia (*Color.a*), seguido de esto, se alterna la imagen (*sprite*) de la punta del taladro por una ligeramente más abajo, que produzca un efecto visual de movimiento de la broca, en el transcurso de 2 segundos.
3. En el próximo segundo, todas las imágenes se transparentan para dar visión a la pantalla principal.

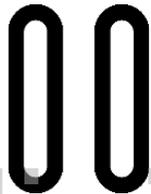
Barra de tareas: las opciones de interfaz de la barra de tareas principal se representan gráficamente de la siguiente tabla:

Tabla 10: Íconos que definen la barra de tareas de la interfaz de usuario del equipo

Barra de tareas principal de ValCNC		
Nombre	Ícono	Descripción
Panel de ejecución		Permite controlar la reproducción de los bloques de comando a través de las opciones de “Reanudar” y “Pausar”.
Ajustes		Configura y selecciona las opciones del funcionamiento del sistema CNC (Escala y velocidad).
Palanca de mando o control manual		Permite controlar manualmente la posición del sistema y reubicar la herramienta.
Establecer conexión		Designa un puerto de datos e inicia la comunicación informática.

En este sentido, al activar cada una de las funciones principales, se despliega un menú correspondiente a las tareas designadas. En efecto, dichas opciones son una extensión de la función principal, divididas así en rutinas, como se muestra a continuación:

Tabla 11: Submenú de la opción “Panel de ejecución”

Opción del submenú	Ícono	Descripción
Limpiar lienzo		Limpiar las texturas que se encuentren en lienzo, dejándolo completamente en blanco
Iniciar o reanudar		Inicia o reanuda el funcionamiento del sistema a nivel virtual y físico (si está habilitada la conexión)
Pausar		Detiene el funcionamiento del sistema a nivel virtual y físico (si está habilitada la conexión)
Detener		Concluye la ejecución de los bloques de comando
Abrir explorador		Abre el explorador de archivos para buscar en los ficheros los documentos en formato de código G.

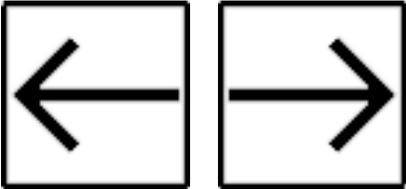
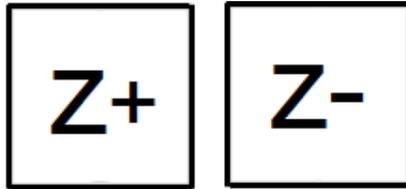
Evidentemente, al accionar la opción de ajustes, se extiende un menú de forma análoga a la opción anterior, pero a diferencia del panel de ejecución, este no cuenta con botones en forma de íconos, sino que cuenta con cuadros de entrada de datos, que ofrecen características que ajustan lo que el usuario visualiza en pantalla, esto con el fin de embellecer la simulación. Este sub-menú se compone de 3 opciones, que se describen en la siguiente tabla:

Tabla 12: Submenú de la opción “Ajuste”

Nombre del submenú	Descripción
Escala	Es una unidad virtual que simula la graduación de la longitud de la línea proporcionalmente
Velocidad	Es una unidad virtual que simula la velocidad de movimiento de la herramienta
Grosor	Es una unidad virtual que simula el espesor de la línea sobre el lienzo

La opción de “palanca de mando” también incluye diferentes alternativas de movimiento para la máquina, tanto a nivel físico como virtual, las cuales se enuncian a continuación en la siguiente tabla:

Tabla 13: Submenú de la opción “Palanca de mando”

Opción del submenú	Ícono	Descripción
“Adelante” y “Atrás” EJE Y		Desplaza el eje Y en dirección positiva y negativa a nivel físico y virtual
“Izquierda” y “Derecha” EJE X		Desplaza el eje X en dirección positiva y negativa a nivel físico y virtual
“Arriba” y “Abajo” EJE Z		Desplaza el eje Z en dirección positiva y negativa a nivel físico y virtual

La plataforma Unity permite crear botones seleccionables a través de un clic. Estos cuentan con 2 características: imagen y botón. La imagen corresponde a la textura que muestra

la interfaz, además de ser el tamaño del área interactiva con el usuario. En cuanto a la característica botón, designa la función a ejecutarse al momento de ser activada.



Figura 4.19: Características de los botones seleccionables

En efecto, la función de detección *On Click ()* para este caso, llama a la opción de interfaz *AbrirNuevoMenu*, lo que invoca a la siguiente rutina:

```
public void AbrirNuevoMenu(GameObject menuNuevo)
{
    //Desactiva el ultimo menu, y activa el nuevo "menuNuevo", y lo hace el menu actual
    if (menuActual != null)
        menuActual.SetActive(false);
    menuActual = menuNuevo;
    menuActual.SetActive(true);
}
```

Figura 4.20: Función para abrir un nuevo menú

La función `AbrirNuevoMenu` necesita los parámetros correspondientes al menú asignado para el botón y su tarea consiste en la desactivación del menú actual abierto (en caso de existir uno), la reasignación del menú nuevo como el actual y su posterior activación.

En el caso del menú “Establecer conexión”, posee un botón desplegable para seleccionar el puerto de comunicación serial. Además, dispone de un botón para iniciar o establecer la respectiva comunicación con el sistema.

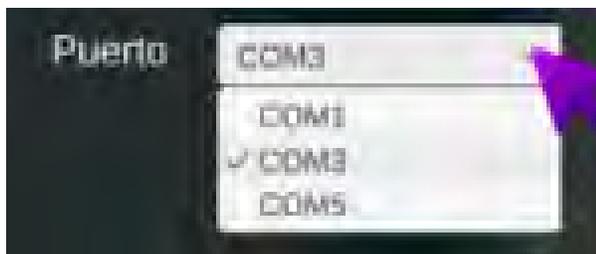


Figura 4.21: Puertos de comunicación disponible a través de un botón desplegable

En la figura anterior, se muestra un menú desplegable que posee la información de todos los puertos disponibles de conexión. Estos puertos se actualizan cada vez que se activa el menú de conexión.

```

1 referencia
public void HacerConexion(string puerto)
{
    myPort = new SerialPort(puerto, 9600);
    myPort.Parity = Parity.None;
    myPort.StopBits = StopBits.One;
    myPort.DataBits = 8;
    myPort.Handshake = Handshake.None;
    myPort.ReadTimeout = 10;
    if (!myPort.IsOpen)
    {
        myPort.Open();
        EnviarSerial("C" + '\n');
    }
}

3 referencias
public void EnviarSerial(string dato)
{
    if (myPort != null)
    {
        if (myPort.IsOpen)
        {
            myPort.Write(dato);
        }
    }
}

```

Figura 4.22: Funciones relacionadas a la comunicación serial

En el caso de la comunicación serial, es necesario crear un puerto específico para hacer la comunicación, y ese puerto tiene que tener características válidas para que la comunicación se efectúe de manera correcta y en coordinación apropiada, ejemplo de esto es que deben tener la misma tasa de envío (baudios). Siempre y cuando el puerto se haya conectado de forma exitosa, se podrá transmitir y recibir datos.

Desde una visión más general, la barra de tareas principal de la interfaz de usuario en el contexto del proceso de interacción persona-computadora, posibilita a través del lenguaje visual, una interacción amigable con el sistema, agrupado entonces con el siguiente diseño:



Figura 4.21: Barra de tareas de la interfaz de usuario de ValCNC

A modo de orientación, se agrega un botón adicional de información, el cual despliega un texto descriptivo de cada uno de los botones principales, para que el usuario tenga facilidad y comodidad en el manejo de la interfaz. Como se muestra a continuación:

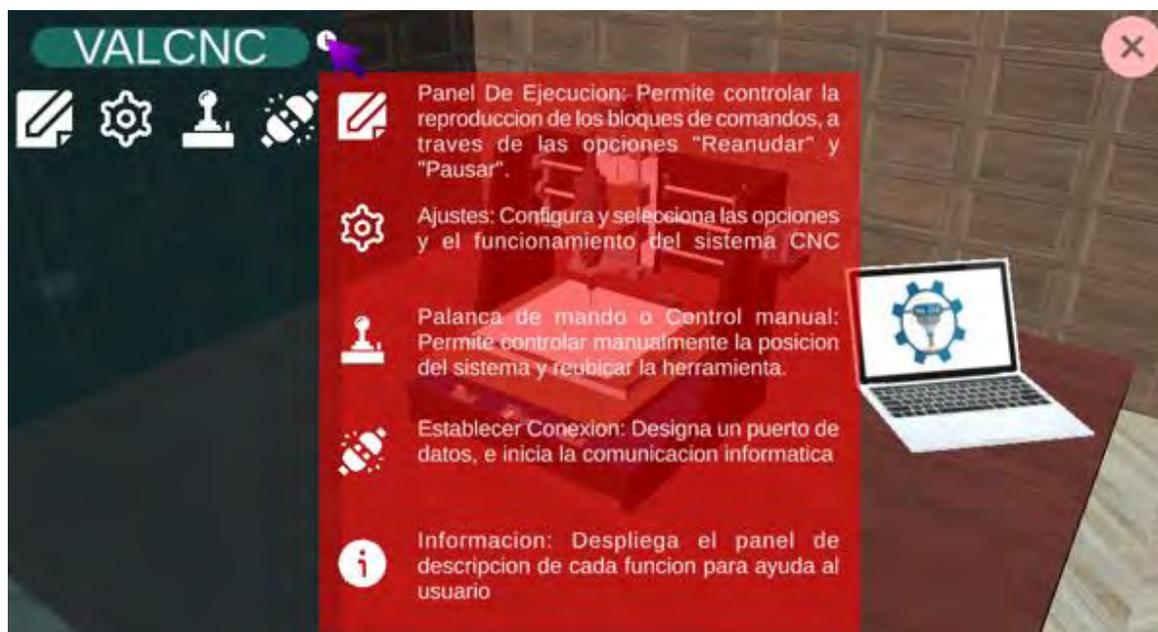


Figura 4.22: Texto descriptivo al presionar el menú información

Resulta claro, que la interfaz se rigie por una serie de pasos para su correcto funcionamiento. El cual procede de esta manera:

4.6.1. Paso 1: Ejecutar la aplicación CNC.

Para ejecutar la aplicación, simplemente se selecciona el archivo y con el segundo clic del mouse se despliega una lista de comandos, en ella se elige la opción de Abrir. Como se muestra en la figura:

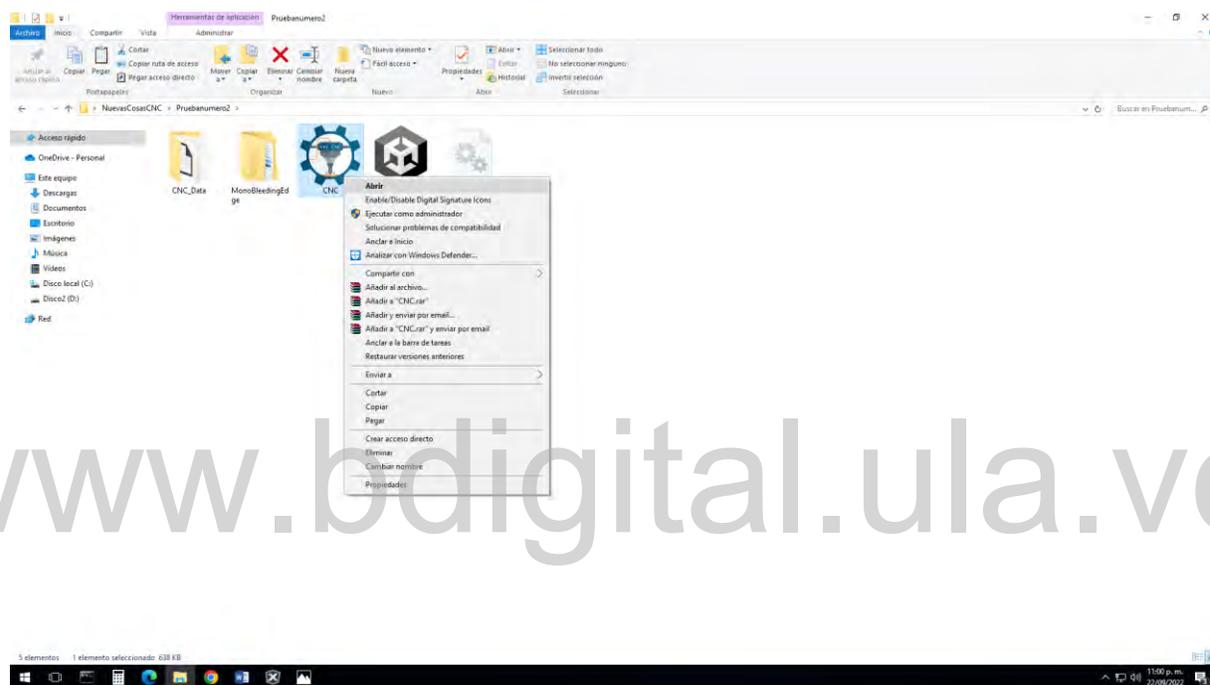


Figura 4.20: Abrir o ejecutar la aplicación

Una vez ejecutada la aplicación, se muestra la animación que se enunció anteriormente, la cual es la siguiente:

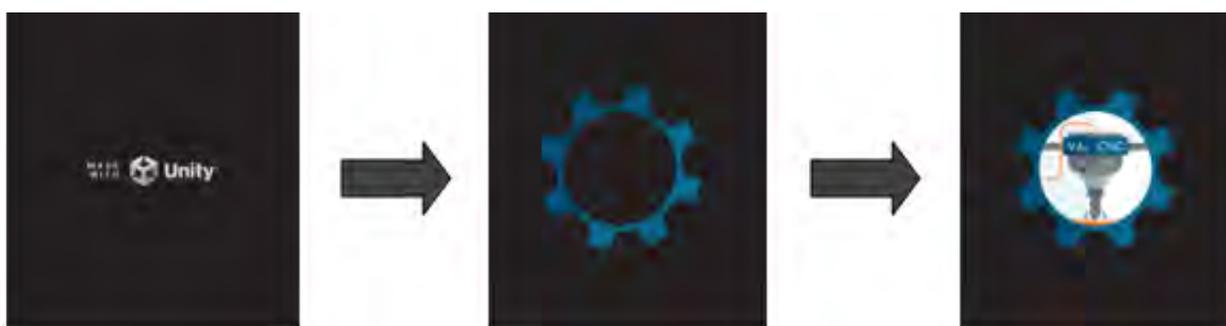


Figura 4.21: Pantallas introductorias de la aplicación

4.6.2. Paso 2: Abrir panel de ejecución

El panel de ejecución es el botón que se encuentra de primero de izquierda a derecha en la barra de tareas principal, que contiene la opción de abrir explorador en su despliegue. Como se muestra a continuación:

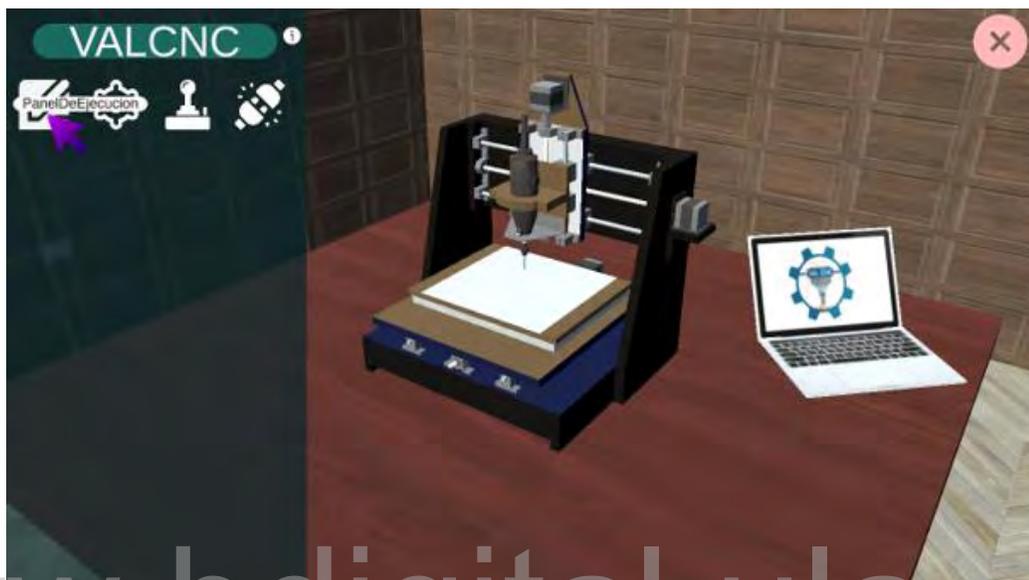


Figura 4.22: Ubicación del panel de ejecución en la aplicación

Una vez accedido al despliegue de su menú, haciendo clic en el ícono con forma de carpeta, se puede acceder al explorador de archivos.



Figura 4.23: Ubicación del botón de abrir archivos en panel de ejecución en la aplicación

4.6.3. Paso 3: Seleccionar el archivo deseado

Luego de hacer clic sobre el ícono de la carpeta, se abre el explorador de archivos y carpetas, donde se selecciona el archivo que se desea tallar.

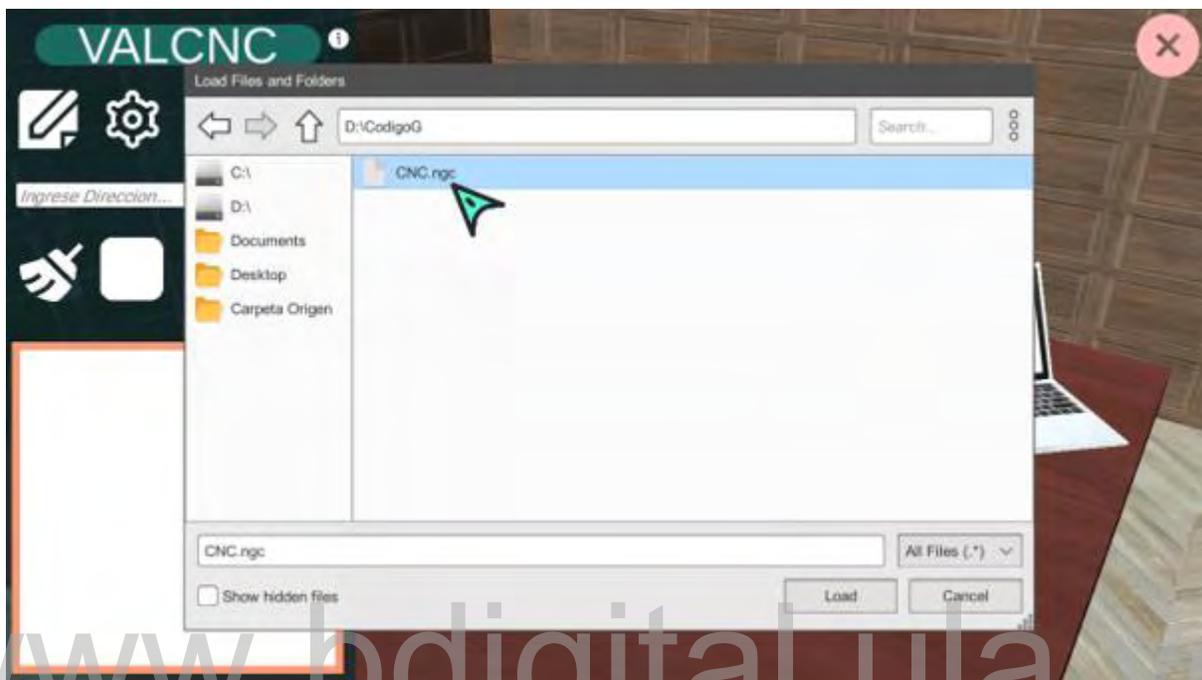


Figura 4.24: Seleccionar el archivo que se desea ejecutar

Luego de seleccionar el archivo, se da clic en “Cargar” o “Load” lo que permite la importación del archivo a la aplicación. Esto se logra mediante el uso del “**Runtime File Browser**”. El cual es un *Assets* o complemento con capacidades funcionales y operativas de acceso libre en la plataforma de Unity store [25]. Este complemento ayuda a mostrar diálogos de guardar/cargar archivos durante la aplicación con un navegador de archivos basado en uGUI. Por lo que aporta las siguientes características:

- Se comporta de manera similar al selector de archivos de Windows.
- Posibilidad de buscar por nombre o filtrar por tipo.
- Enlaces rápidos.
- Interfaz de usuario sencilla.
- Arrastrable y redimensionable.

- Posibilidad de elegir carpetas en lugar de archivos.
- Se puede cambiar la piel fácilmente.
- Admite permisos de tiempo de ejecución en Android M+.

Estas características convierten el explorador de archivos en una herramienta intuitiva y fácil de manejar para el usuario.

4.6.4. Paso 4: Comprobar en la pre-visualización que el archivo se exportó correctamente

Una vez seleccionado el archivo, se muestra la conversión de las coordenadas del código de G en el resultado del trazado deseado. Por ejemplo, luego de vectorizar el logo de la aplicación a través del Inkscape y cargarlo en la aplicación, se puede observar lo siguiente:

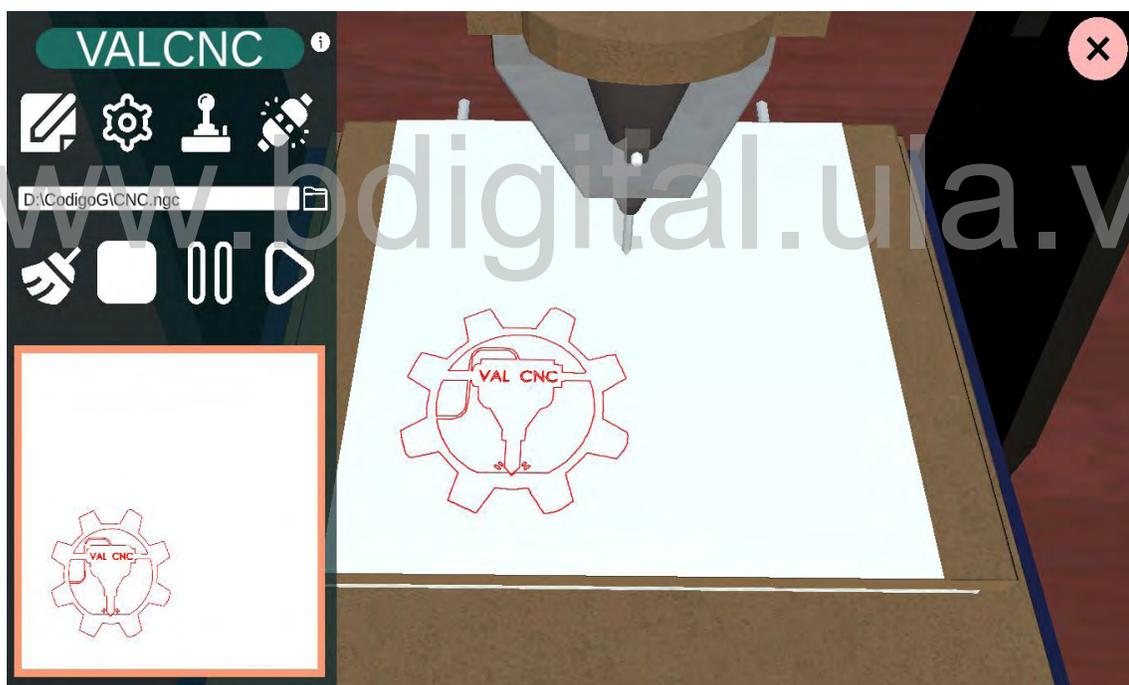


Figura 4.25: Pre-visualización de un archivo de código G

En efecto, la visualización previa del trayecto, ayuda al usuario a verificar si el resultado que se desea es el adecuado. De no ser el esperado, se puede utilizar la herramienta de limpiar, que se encuentra sobre este mismo menú. Es recomendable chequear el archivo convertido en código G a través del Inkscape para evitar el trazo de una imagen no deseada. De ser correcto, se procede a comunicarse con el sistema.

4.6.5. Paso 5: Abrir el menú de conexiones

El menú de conexiones es el botón que se encuentra de primero de derecha a izquierda en la barra de tareas principal. Como se muestra a continuación:

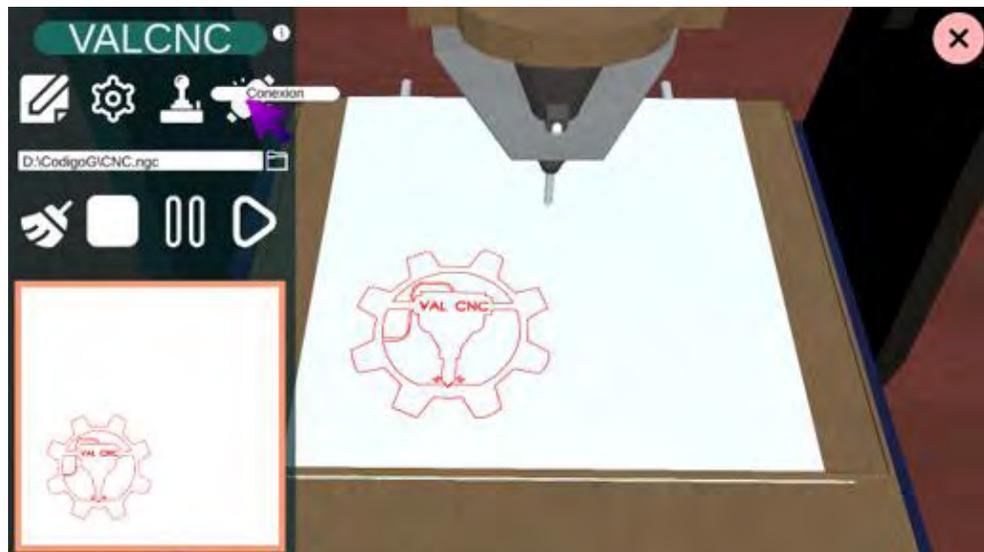


Figura 4.26: Ubicación del menú para establecer conexiones

Una vez accedido al menú de establecer conexiones, se procede a dirigirse al botón desplegable de puertos y seleccionar el puerto de comunicación donde se encuentra conectado el sistema.

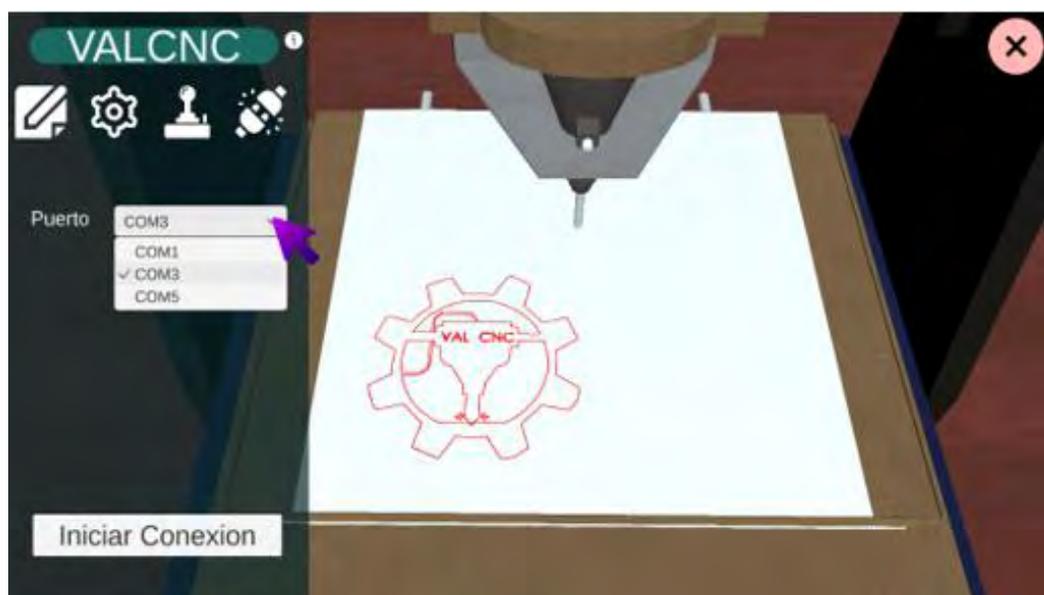


Figura 4.27: Menú desplegable de los puertos de comunicación

Para conocer la ubicación del puerto donde se encuentra conectada la máquina. Hay que dirigirse al panel de control, en la sección de administrador de dispositivos, específicamente en la pestaña de Puertos (COM y LTP).

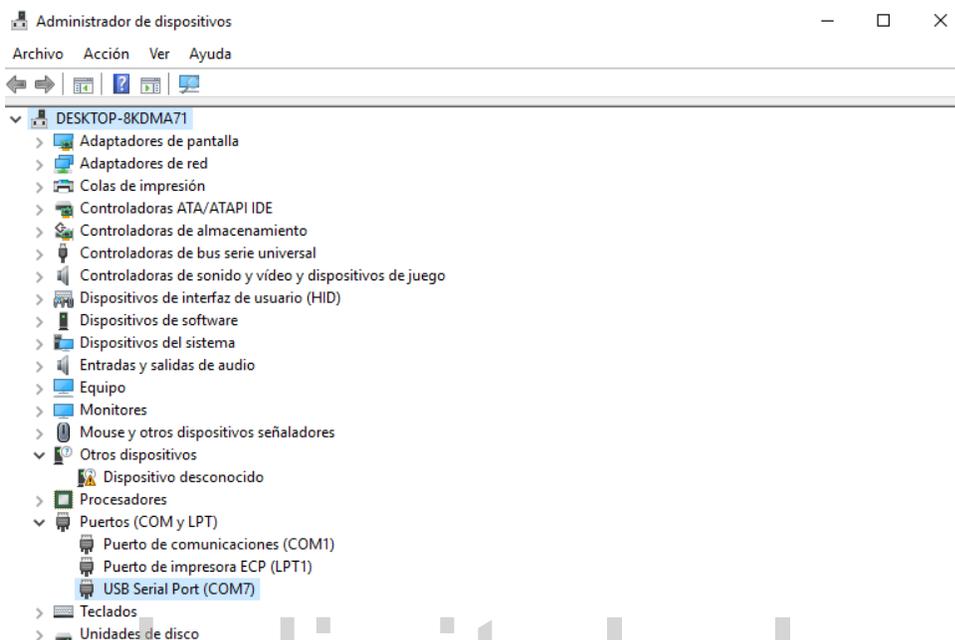


Figura 4.28: Administrador de dispositivos del panel de control, donde se encuentra ubicado el puerto COM donde está conectado el sistema

Evidentemente, es necesario tener conectado el puerto de comunicación para poder seleccionarlo en el menú desplegable de la aplicación. Una vez seleccionado el puerto COM correcto, se debe iniciar la conexión a través del botón de “Iniciar Conexión”.

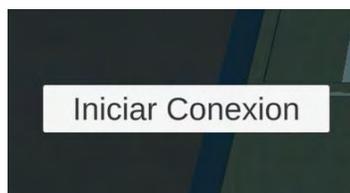


Figura 4.29: Botón para iniciar la conexión con el puerto de comunicación

4.6.6. Paso 6: Volver al panel de Ejecución y seleccionar la opción de iniciar

Una vez establecida la comunicación con el sistema. Se procede a volver al botón del panel de ejecución, haciendo clic sobre él y desplegando su sub-menú. Este cuenta con la

opción de “Iniciar” simbolizada por un triángulo que mira en dirección hacia la derecha, que luego presionarla, da comienzo a la ejecución el mecanismo.

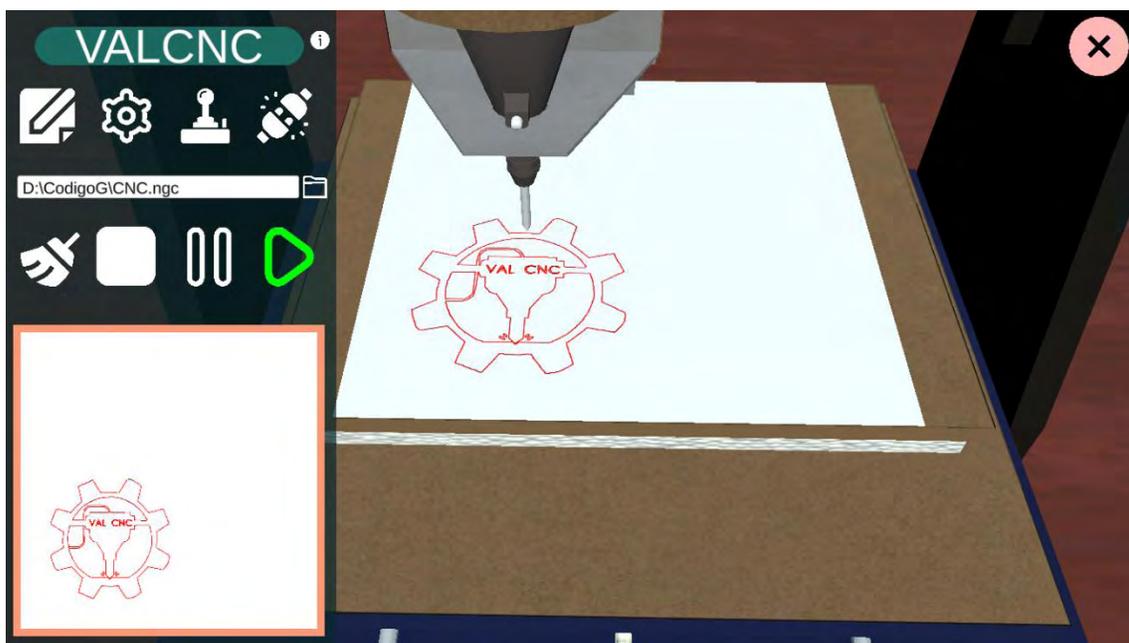


Figura 4.30: Sistema iniciado a nivel de simulación

Una vez iniciado el mecanismo de funcionamiento, es recomendable borrar o limpiar el lienzo mediante la opción de limpiar, para que la simulación del trazo no se solape con la previsualización.

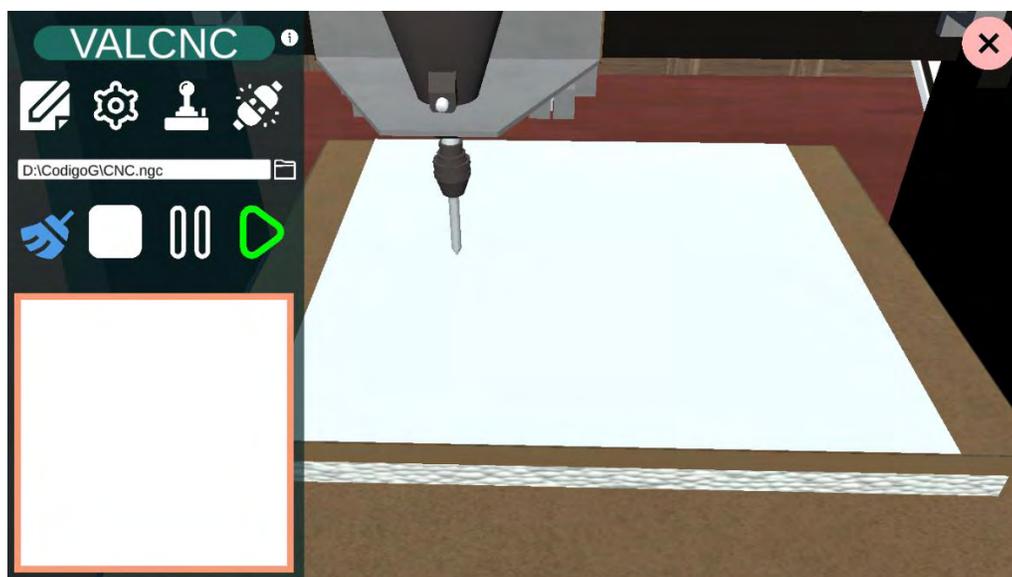


Figura 4.31: Lienzo en blanco e iniciado el sistema

4.6.7. Paso 7: Apreciar y vigilar la ejecución del sistema

Además de apreciar el funcionamiento tanto a nivel de simulación como a nivel físico, es necesario vigilar el mecanismo de agentes externos que pudieran entorpecer su operación.

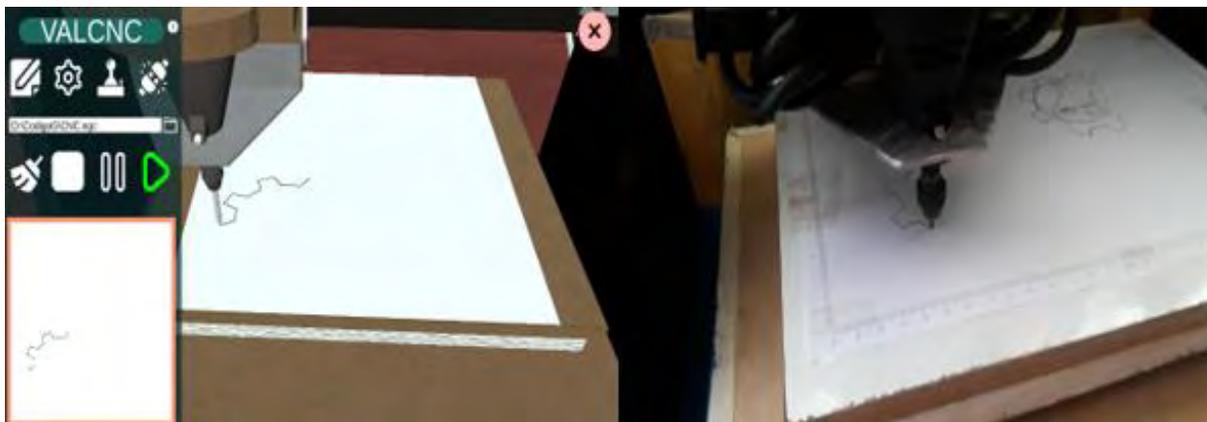


Figura 4.32: Apreciación de los mecanismos a nivel de simulación y físico

De ocurrir algún problema externo, es posible pausar el mecanismo para que se puedan corregir. Esto se hace a través de la opción de “Pausar”, la cual detiene momentáneamente la ejecución de los bloques de comando luego de finalizar el comando que esté trazando en el instante.

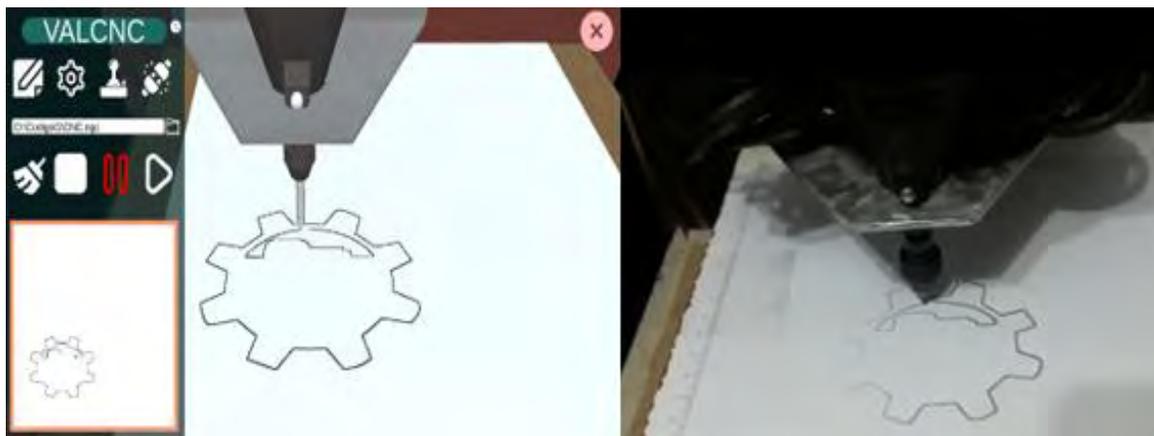


Figura 4.33: Pausar el mecanismo a nivel de simulación y físico

Si se desea continuar con la ejecución de los bloques de comandos, se puede dar clic en la opción de “Reanudar” y el programa inicia nuevamente el mecanismo, teniendo como punto

de partida el comando siguiente al que finalizó la máquina en el momento de ser pausada. Una vez terminado todos los comandos, el resultado es el siguiente:

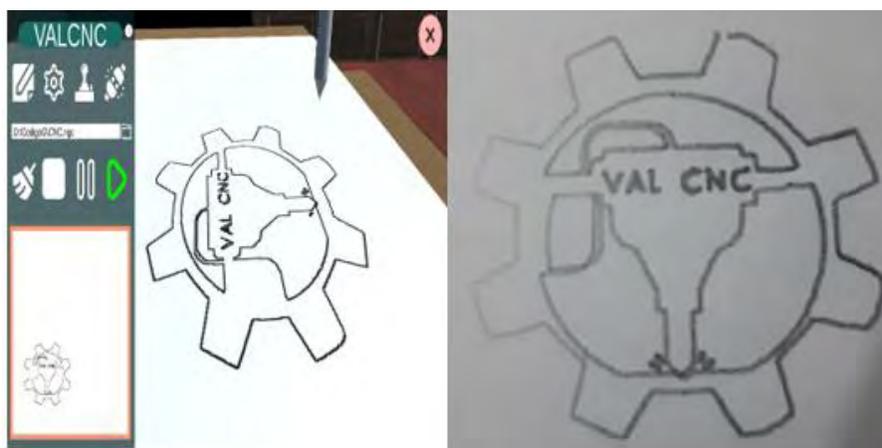


Figura 4.34: Boceto final obtenido

4.6.8. Paso 8: Salir del programa

Para finalizar el uso de la aplicación, se puede dar clic en la opción de Cerrar, la cual se encuentra ubicada en la parte superior derecha. Tiene la forma de un círculo con una “X” en su interior.



Figura 4.35: Ícono para salir de la aplicación.

Una vez presionado la opción de “Salir”, se despliega una animación, mostrando el siguiente mensaje:



Figura 4.36: Animación final de la aplicación

4.7 PRUEBAS DE PRECISIÓN

La característica más importante de una máquina CNC es su precisión [17], pero en cualquier actividad científica o técnica es inevitable la existencia de errores que se encuentran presentes, ya sea consecuencia del mecanismo o del software que lo controla, Por lo tanto, es necesario conocer los fenómenos que producen dichos errores, para así, cuantificarlos para corroborar la fiabilidad del equipo. En cuanto a la presión del mismo, es conveniente conocer el valor real que producen los motores sobre el tornillo de arrastre. En teoría, como se trabaja con los motores NEMA 17 de 200 pulsos por vuelta (siempre y cuando estén configurados para pasos completos), y tornillos de arrastre de 8mm por vuelta; es decir, con cada vuelta que realice un motor, la máquina se debería trasladar linealmente 8mm. Por lo tanto, es imprescindible demostrar si se cumple físicamente este fenómeno. En efecto, el uso de un instrumento de medición altamente sensible como lo es un vernier digital, el cual cuenta con una sensibilidad de 0,01mm. Dicho valor es eficaz para calcular el error en una medición directa en centésimas de milímetro.

Para las pruebas de precisión se realizan 10 tramos de 8mm y se cuantifica el error promedio. Este proceso se realiza para cada motor, y los resultados se muestran a continuación:

Tabla 14: Cálculo de precisión del eje Z

Tramo(+)	1	2	3	4	5	6	7	8	9	10
Medición(mm)	8,05	16,09	24,21	32,20	40,29	48,44	56,52	64,57	72,61	80,60
Valor Real(mm)	8	16	24	32	40	48	56	64	72	80
Error(mm)	0,05	0,04	0,12	0,01	0,09	0,15	0,08	0,05	0,04	0,01

Con los datos de la tabla 14, es posible calcular un error absoluto a través del promedio de los errores, por lo tanto:

$$E_{absolutoz} = \left(\frac{0,05 + 0,04 + 0,12 + 0,01 + 0,09 + 0,15 + 0,08 + 0,05 + 0,04 + 0,01}{10} \right)$$

$$E_{absolutoz} = 0,064mm$$

Este valor de error absoluto se traduce como la precisión que garantiza la máquina en el eje Z, es decir, la máquina de asegurar una precisión de $(8,00 \pm 0,06)$ mm por cada vuelta del motor en este eje. En cuanto al eje Y, el resultado es el siguiente:

Tabla 15: Cálculo de precisión del eje Y

Tramo(+)	1	2	3	4	5	6	7	8	9	10
Medición(mm)	8,09	16,19	24,23	32,28	40,36	48,48	56,60	64,71	72,75	80,86
Valor Real(mm)	8	16	24	32	40	48	56	64	72	80
Error(mm)	0,09	0,10	0,04	0,05	0,08	0,12	0,12	0,11	0,14	0,10

Con el uso de los datos mostrados en la tabla 15, es posible calcular un error absoluto a través del promedio de los errores, por lo tanto:

$$E_{absolutoy} = \left(\frac{0,09 + 0,10 + 0,04 + 0,05 + 0,08 + 0,12 + 0,12 + 0,11 + 0,14 + 0,10}{10} \right)$$

$$E_{absolutoy} = 0,086mm$$

El error absoluto del eje Y se interpreta como la precisión que garantiza la máquina sobre el eje Y. Por lo tanto, la máquina de asegurar una precisión de $(8,00 \pm 0,09)$ mm por cada vuelta sobre el tornillo de arrastre del eje Y. Para el eje X, se expresan los cálculos a continuación:

Tabla 16: Cálculo de precisión del eje X

Tramo(+)	1	2	3	4	5	6	7	8	9	10
Medición(mm)	8,08	16,19	24,23	32,29	40,35	48,41	56,53	64,61	72,70	80,76
Valor Real(mm)	8	16	24	32	40	48	56	64	72	80
Error(mm)	0,08	0,11	0,04	0,06	0,06	0,06	0,12	0,08	0,09	0,06

Con los valores expresados en la tabla 16, es posible calcular un error absoluto a través del promedio de los errores, por lo tanto:

$$E_{absolutox} = \left(\frac{0,08 + 0,11 + 0,04 + 0,06 + 0,06 + 0,06 + 0,12 + 0,08 + 0,09 + 0,06}{10} \right)$$

$$E_{absolutox} = 0,076mm$$

Al igual que en el eje Y, el error absoluto del eje X asegura una precisión de $(8,00 \pm 0,08)$ mm por cada vuelta del tornillo sobre el eje X.



Figura 4.37: Pruebas de precisión con un vernier digital

CONCLUSIONES

Los resultados obtenidos en este trabajo, dan por cumplidos los objetivos propuestos para el diseño y construcción de un equipo de control numérico computarizado fabricado de forma artesanal con capacidad multifuncional.

Se usaron piezas adquiridas en el mercado local de la ciudad y con todos sus periféricos ensamblados en una armazón de diseño propio.

El software utilizado se desarrolló específicamente para este diseño y cumple con un mecanismo de precisión milimétrica menor o igual a 0,09mm. Con una interfaz gráfica capaz de llevar a cabo todas las tareas necesarias para el correcto funcionamiento del sistema de forma fácil y sencilla, a través de una comunicación serial con un ordenador.

Por consiguiente, este prototipo se presenta como una solución multifuncional, por ejemplo: para la construcción de circuitos PCB de una forma más ecológica y ambientalmente sana, para cortes y perforaciones en papel, madera o plástico e incluso como impresora 3D.

Los costos económicos resultaron más bajos y viables, con un ahorro del 66,34% en comparación con la adquisición de una máquina CNC adquirida en el mercado nacional e internacional.

Finalmente, el contenido abarcado en este trabajo, generó un prototipo como punto de partida para la construcción de otras máquinas, que funcionen mediante un control numérico de coordenadas, como lo son los drones, tornos, brazos robóticos, líneas de ensamblaje, entre otros; ya que el principio de funcionamiento es el mismo.

RECOMENDACIONES

A pesar que la CNC construida concretó las pruebas con errores imperceptibles, con un diseño atractivo, y una estructura con un software funcional, existen algunas recomendaciones que lograrían mejorar el rendimiento en todas sus etapas. Estas recomendaciones se enuncian a continuación:

A nivel de software, se recomienda utilizar un microcontrolador de mayor capacidad, que permita almacenar todos los comandos de acciones del lenguaje de programación para maquinaria CNC de la norma 6983 descrita por la Organización Internacional de Normalización.

A nivel mecánico, es necesario el uso de tonillos de arrastre de doble hélice para todos los ejes del mecanismo; además, se recomienda el uso de tuercas de fricción anti-*backlash*, que disminuyen el error de precisión.

En cuanto a tamaño, es posible escalar las dimensiones de la estructura, para así, tener un mayor espacio de trabajo útil.

En diseño, se recomienda readaptar dualidad en su arquitectura para que sea más cómodo, y se pueda cambiar una herramienta por otra, además, se recomienda acondicionar el equipo para que funcione como impresora 3D. También se sugiere añadir sensores de finales de carrera en ambos extremos de los ejes para protección del equipo y adicionar disipadores de ruido para todo el sistema.

A nivel artístico, se recomienda darles mejor acabado a las superficies del armazón con masillas y pintura, así como también, agregar materiales como acrílicos y calcomanías entre otros, para ocultar la parte mecánica de la estructura, y que contribuyan a embellecimiento visual del mecanismo y de toda la CNC.

REFERENCIAS BIBLIOGRÁFICAS

R. RAJAGOPALAN, ««Applications of Neural Network in Manufacturing»,» de
 [1] *Proceedings of the 29th Annual Hawaii International Conference on System Sciences*,
 Hawaii, 1996.

M. TOLOSA, «Ocompra,» 5 Octubre 2017. [En línea]. Available:
 [2] <https://www.ocompra.com/venezuela/item/maquina-de-corte-laser-714211099/>. [Último
 acceso: 19 Febrero 2022].

E. GARCÍA, *Automatización de procesos industriales*, México: UNIVERSITAT,
 [3] 1999.

J. M. LESHERAS, *Tecnología mecánica y metrotecnica*, España:
 [4] DONOSTIARRA, 2000.

M. GOOVER, *Automation, Productions Systems and computer-integrated*
 [5] *manufacturing*, Reino Unido: PEARSON, 2013.

D. ZAMORA, *Sistemas de producción y operaciones financieras*, España:
 [6] EDICIONES UNIVERSIDAD DE VALLADOLID, 2017.

J. ROLDÁN, *Planificación y control de la producción*, Ecuador: IDIUC, 2001.
 [7]

I. GARCÍA, J. LAGOS, L. URREGO y P. DELGADO, *Diseño e implemantación*
 [8] *de un control CNC para crear modelos y esculturas en tercera dimensión a partir de un*
diseño CAD, Bogotá: Universidad de San Buenaventura, 2009.

G. SÁNCHEZ, *Generador de código CNC para el proceso de fresado, empleando*
 [9] *visión computarizada*, México: Universidad Autónoma del estado de Hidalgo, 2014.

A. JIMÉNEZ, Diseño, Programación, Simulación y fabricación en CNC con
[10] programas de CAD/CAM, México: Universidad Nacional Autónoma de México, 2018.

J. CANELONES, Diseño y construcción de un equipo de control numérico
[11] (CNC) para el fresado de circuitos impresos (PCB), utilizando software libre, Venezuela:
Universidad de Los Andes, 2014.

«Mundo Compresor,» 30 Marzo 2020. [En línea]. Available:
[12] <https://www.mundocompresor.com/diccionario-tecnico/cnc>. [Último acceso: 2022
Febrero 20].

«Forestal Maderero,» 30 Agosto 2019. [En línea]. Available:
[13] <https://www.forestalmaderero.com/articulos/item/la-historia-del-cnc-control-numericopor-computadora.html>. [Último acceso: 20 Febrero 2022].

ZHJBD, «Amazon,» 2022. [En línea]. Available: <https://www.amazon.com/-/es/3018-PRO-Control-pl%C3%A1stico-acr%C3%ADlico-11-811x7-087x1-772/dp/B09NKLPHPH/>. [Último acceso: 01 Abril 2022].

«GnipseI,» [En línea]. Available: <https://gnipseI.com/files/mill-g-code/>. [Último
[15] acceso: 2022 Febrero 21].

Fundación Wikimedia, Inc., «Wikipedia,» 23 Febrero 2022. [En línea]. Available:
[16] <https://es.wikipedia.org/wiki/Inkscape>. [Último acceso: 2022 Marzo 01].

V. GARCÍA, «Diario Electrónico hoy,» Marzo 05 2020. [En línea]. Available:
[17] <https://www.diarioelectronicohoy.com/blog/>. [Último acceso: 2022 Marzo 01].

Allegro, «Alldatasheet,» 2018. [En línea]. Available:
[18] <https://www.alldatasheet.com/view.jsp?Searchword=A4988&sField=4>. [Último acceso:
05 Febrero 2022].

Fundación Wikimedia, Inc., «Wikipedia,» 26 Febrero 2022. [En línea]. Available:
[19] <https://es.wikipedia.org/wiki/Blender>. [Último acceso: 02 Marzo 2022].

Microchip, «Alldatasheet,» 2018. [En línea]. Available:
 [20] <https://www.alldatasheet.com/view.jsp?Searchword=PIC18F4550&sField=4>. [Último acceso: 05 Febrero 2022].

Microchip , «Microchip Developer,» 2021. [En línea]. Available:
 [21] <https://microchipdeveloper.com/mplabx:start>. [Último acceso: Marzo 01 2022].

Microchip, «Microchip Developer,» 2022. [En línea]. Available:
 [22] <https://www.microchip.com/en-us/education/developer-help/learn-tools-software/mcu-mpu/compiler>. [Último acceso: 2022 Marzo 01].

J. MARTINEZ, «Joelbuenrostro.blog,» 15 Marzo 2018. [En línea]. Available:
 [23] <http://joelbuenrostroblog.blogspot.com/2018/03/guia-de-introduccion-de-xc8.html>.
 [Último acceso: 2022 Marzo 02].

Wiki, «Wiki.Blender,» 15 Febrero 2022. [En línea]. Available:
 [24] <https://wiki.blender.org/wiki/>. [Último acceso: 02 Marzo 2022].

Unity, «Unityes,» 2022. [En línea]. Available: <https://unity.com/es>. [Último
 [25] acceso: 02 Marzo 2022].

«Labcenter,» Labcenter Electronics, 30 Marzo 2022. [En línea]. Available:
 [26] <https://www.labcenter.com/whyvsm/>. [Último acceso: 12 Abril 2022].

«Mikroe,» MikroElektronika d.o.o., Mayo 2022. [En línea]. Available:
 [27] <https://www.mikroe.com/mikroprog-pic-dspic-pic32>. [Último acceso: 12 Mayo 2022].

«Eterlogic,» www.eterlogic.com, 2022. [En línea]. Available:
 [28] <http://www.eterlogic.com/Products.VSPE.html>. [Último acceso: 12 Mayo 2022].

«Soft32,» ITNT Soft32, Febrero 2022. [En línea]. Available: [https://hercules-
 \[29\] setup.soft32.com/](https://hercules-setup.soft32.com/). [Último acceso: 12 Mayo 2022].

«Microsoft.com,» Microsoft, Mayo 2022. [En línea]. Available:
 [30] <https://visualstudio.microsoft.com/es/>. [Último acceso: 12 Mayo 2022].

U. VITIELLO, «De máquinas y herramientas,» 28 Diciembre 2015. [En línea].
 [31] Available: <https://www.demaquinasyherramientas.com/mecanizado/introduccion-a-la-tecnologia-cnc>. [Último acceso: 20 Marzo 2022].

«ES-academic,» Marzo 2010. [En línea]. Available: <https://es-academic.com/dic.nsf/eswiki/405989>. [Último acceso: 11 Mayo 2022].

V. ZARABOZO, «Formacionmecanizado,» 2022. [En línea]. Available:
 [33] <https://formacionmecanizado.com/fresadora/>. [Último acceso: 30 Marzo 2022].

«Ncservice,» 2022. [En línea]. Available: <https://www.ncservice.com/es/sabes-que-es-un-husillo-y-cual-es-el-mas-empleado-en-las-fresadoras-cnc>. [Último acceso: 26 Mayo 2022].

C. VALERO, «GGbearings,» 2022. [En línea]. Available:
 [35] <https://www.ggbearings.com/es/por-que-elegir-ggb/preguntas-frecuentes/cojinetes/>.
 [Último acceso: 3 Mayo 2022].

«Staticboard,» 19 Junio 2020. [En línea]. Available:
 [36] <https://www.staticboards.com/showpost.php?postID=21>. [Último acceso: 2022 Marzo 02].

ASTECC, «AMAZON,» 20 Febrero 2013. [En línea]. Available:
 [37] <https://www.amazon.com/-/es/As-Tec-SUPPLY-200W-ATX202-3545-REV-00/dp/B00B8706DU>. [Último acceso: 12 Abril 2022].

ARDUSYSTEM, «Naylampmechatronic,» 30 Junio 2017. [En línea]. Available:
 [38] <https://naylampmechatronics.com/ardusystem-shields/68-shield-cnc-v3-grbl.html>.
 [Último acceso: 2 Febrero 2022].

A. RUÍZ, «Uelectronics,» 2022. [En línea]. Available:
 [39] <https://uelectronics.com/producto/cnc-shield-v3-para-arduino-uno/>. [Último acceso: 30 Marzo 2022].

M. MORENO, «Mercado Libre,» 20 Enero 2022. [En línea]. Available:

[40] https://articulo.mercadolibre.com.ve/MLV-575990825-placas-de-pcb-perforada-7x9cm-_JM#position=2&search_layout=stack&type=item&tracking_id=d995760c-ea9b-430d-9495-4540fdb1969d. [Último acceso: 18 Abril 2022].

YOFULY, «Amazon,» [En línea]. Available: https://www.amazon.com/-/es/Yofuly-Router-Control-m%C3%A1quina-grabado/dp/B08FHZQCCP/ref=sr_1_9?keywords=cnc%2B3018%2Bpro&pd_rd_r=05ba3ace-27cc-4b11-a2b2-181bfb571712&pd_rd_w=aZTfB&pd_rd_wg=i7R2A&pf_rd_p=4fa0e97a-13a4-491b-a127-133a554b4da3&pf_rd_r=RHGDZG. [Último acceso: 2021 Febrero 21].

Fundación Wikimedia, Inc., «Wikipedia,» Febrero 10 2022. [En línea]. Available: [42] https://es.wikipedia.org/wiki/Microcontrolador_PIC. [Último acceso: 2022 Marzo 01].

I. L. DE LA CRUZ DIMAS, «Sites.google.com,» Google, 17 Abril 2013. [En [43] línea]. Available: <https://sites.google.com/site/mecatroneanos/classroom-news/asdasdfasdas>. [Último acceso: 13 Mayo 2022].

www.bdigital.ula.ve