



**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:**

**INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES**

**TEMA: IMPLEMENTAR UN SISTEMA DE MANEJO LINEAL PARA UNA MESA DE CONTROL NUMÉRICO COMPUTARIZADO.**

**AUTOR: LUIS HERNÁN HERRERA HERRERA**

**TUTOR: ING. Mg. RENÉ ERNESTO CORTIJO LEYVA**

**AÑO: 2018**

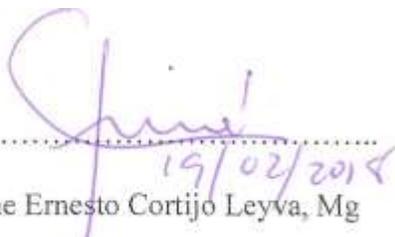
## APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación "IMPLEMENTAR UN SISTEMA DE MANEJO LINEAL PARA UNA MESA DE CONTROL NUMÉRICO COMPUTARIZADO", presentado por la Sr. Luis Hernán Herrera Herrera, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito DM. enero del 2018

TUTOR

  
-----  
19/02/2018  
Ing. Rene Ernesto Cortijo Leyva, Mg

## **AGRADECIMIENTO**

En primera instancia agradezco a mis formadores personas de gran sabiduría quienes se han esforzado por ayudarme a llegar a culminar mi meta planteada.

Tus esfuerzos son impresionantes y tu amor para mi es invaluable, junto con mi padre me has educado, me han proporcionado todo lo que necesitado. Tus consejos y enseñanzas las aplico cada día.

Te doy gracias, madre.

ATT: Luis Hernán Herrera Herrera

## **DEDICATORIA**

Mi tesis se la dedico a mi Dios quien me guio por un buen camino, dándome fuerzas para seguir adelante sin desmayar, enseñándome a encarar los problemas presentados en el camino.

Con todo el amor y el cariño a mis queridos Padres por sus consejos, sacrificios, comprensión, ayuda en los momentos difíciles. Me han dado todo lo que soy como persona principios, perseverancia, valores y carácter para conseguir mis metas planteadas.

ATT: Luis Hernán Herrera Herrera

# INDICE

## PÁGINA

<b>INDICE .....</b>	<b>I</b>
<b>INDICE DE FIGURAS .....</b>	<b>V</b>
<b>INDICE DE TABLAS .....</b>	<b>IX</b>
<b>RESUMEN .....</b>	<b>X</b>
<b>ABSTRACT .....</b>	<b>XI</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO .....</b>	<b>1</b>
<b>PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>2</b>
<b>FORMULACIÓN DEL PROBLEMA.....</b>	<b>2</b>
<b>JUSTIFICACIÓN .....</b>	<b>3</b>
<b>OBJETIVO GENERAL.....</b>	<b>3</b>
<b>OBJETIVO ESPECÍFICO .....</b>	<b>4</b>
<b>CAPÍTULO I.....</b>	<b>5</b>
<b>1. FUNDAMENTACIÓN TEÓRICA .....</b>	<b>5</b>
1.1.Plasma .....	5

1.2.Powermax 30 AIR.....	6
1.3.Dimensiones y peso de la fuente de energía del plasma.....	6
1.4.Dimensiones de la antorcha.....	7
1.5.Especificaciones.....	8
1.6.Sistema mecánico.....	8
1.7.Motor paso a paso.....	9
1.8.Secuencias de los motores paso a paso unipolares.....	11
1.9. Wantai mini stepper 42BYGHW811.....	14
1.10. Cnc shield.....	15
1.11. Driver a4988.....	17
1.12. Arduino uno.....	18
1.13. Módulo bluetooth para arduino hc-06.....	20
1.14. Relé arduino.....	23
1.15. Labview.....	24
<b>CAPITULO II.....</b>	<b>26</b>
<b>2. PROPUESTA.....</b>	<b>26</b>
2.1 análisis del diseño de la máquina de corte cnc propuesta.....	26
2.2. Diseño de la máquina automatizada:.....	27
2.3. Ventajas de la máquina automatizada propuesta.....	28
2.4. Descripción de la operación de corte propuesto.....	29
2.5. Especificación y detalle del software requerido y su programación.....	29
2.6. Configurar grbl.....	30
2.7. Ergonomía aplicada a la máquina automatizada propuesta.....	32
<b>2.8. DISEÑO PARA SEGURIDAD.....</b>	<b>32</b>

<b>CAPITULO III.....</b>	<b>33</b>
<b>3.1. DIAGRAS DE INTERCONEXIÓN DE DISPOSITIVOS. ....</b>	<b>33</b>
3.1.1. Diagrama eléctrico .....	33
3.1.2. Diagrama electrónico.....	34
<b>3.2. DIAGRAMA DEL SHIELD .....</b>	<b>35</b>
3.2.1. PCV Frontal shield.....	36
3.2.2. PCV Trasero shield.....	36
<b>3.3. DESARROLLO.....</b>	<b>37</b>
3.3.1. Diagrama flujo de procesos de diseño de aplicación para alimentación de una fuente ac – dc.....	37
3.3.2. Desarrollo de una aplicación para encender una fuente para una mesa cnc por medio de bluetooth.....	38
3.3.3. Programación de bloques en app inventor 2 .....	38
<b>3.3.4. DIAGRAMA DE FLUJO DE PROCESOS DE LOS BLOQUES DE LABVIEW .....</b>	<b>42</b>
3.3.5. Desarrollo del diagrama de bloques en el programa labview .....	43
3.3.6. Diagrama de flujo demostrando el procedimiento de código de arduino.....	70
3.3.7. Desarrollo codigo de arduino .....	71
<b>3.4. IMPLEMENTACIÓN.....</b>	<b>100</b>
3.4.1. Diagrama de flujo la implementación de un sistema de manejo lineal para una mesa de control numérico computarizado. ....	100
3.4.2. Implementación mecánica.....	101
3.4.3. Implementación del sistema electrónico.....	105
3.4.4. Implenentacion interfaz labviuw.....	106

3.4.5. Implementación interfaz g -code sender.....	106
<b>3.5. PRUEBAS DE FUNCIONAMIENTO .....</b>	<b>107</b>
3.5.1. Sistema mecánico.....	107
3.5.2. Sistema electrónico .....	109
3.5.3. Hardware driver a4988 .....	110
3.5.4. Firmware.....	111
3.5.5. Software.....	111
3.6. Análisis económico .....	113
3.6.1. Costos de materiales y equipos para la mesa de corte con plasma cnc .....	113
3.6.2. Rentabilidad de la mesa cnc. ....	115
<b>CONCLUSIONES.....</b>	<b>116</b>
<b>RECOMENDACIONES .....</b>	<b>118</b>
<b>BIBLIOGRAFÍA.....</b>	<b>119</b>

## INDICE DE FIGURAS

	<b>PÁGINA</b>
<b>Figura 1.</b> Proceso de funcionamiento del plasma. ....	6
<b>Figura 2.</b> Medidas y peso de la fuente del plasma. ....	7
<b>Figura 3.</b> Medidas de la antorcha del plasma. ....	7
<b>Figura 4.</b> Motor paso a Paso. ....	9
<b>Figura 5.</b> Estructura motor paso a paso bipolar. ....	10
<b>Figura 6.</b> Estructura motor paso a paso unipolar. ....	10
<b>Figura 7.</b> Movimiento del rotor indicando la secuencia de paso completo doble. ....	11
<b>Figura 8.</b> Movimiento del rotor indicando la secuencia de paso completo Básica. ....	12
<b>Figura 9.</b> Movimiento del rotor indicando la secuencia de medio paso. ....	13
<b>Figura 10.</b> Motor wantai mini stepper 42BYGHW811 .....	14
<b>Figura 11.</b> Módulo CNC shield para arduino. ....	16
<b>Figura 12.</b> Driver A4988, para controlar motores Paso a paso desde Arduino. ....	17
<b>Figura 13.</b> Sistema en bebido Arduino UNO. ....	19
<b>Figura 14.</b> Módulo bluetooth para arduino HC-06. ....	20
<b>Figura 15.</b> Módulo Relé para Arduino. ....	23
<b>Figura 16.</b> Símbolo del módulo relé. ....	23
<b>Figura 17.</b> Logo programa Labview, National Instruments. ....	25
<b>Figura 18.</b> Corte manual con plasma .....	26
<b>Figura 19.</b> Relación Operario - Máquina .....	27
<b>Figura 20.</b> Ergonomía aplicada a una máquina CNC. ....	32
<b>Figura 21.</b> PCV frontal shield .....	36

<b>Figura 22.</b> PCV trasero shield .....	36
<b>Figura 23.</b> Diagrama de flujo de proceso de aplicación con bluetooth para controlar fuente de poder. ....	37
<b>Figura 24.</b> Activación de Bluetooth.....	38
<b>Figura 25.</b> Selección de Dispositivo .....	39
<b>Figura 26.</b> Botones de encendido y apagado.....	39
<b>Figura 27.</b> Aplicación terminada en la página APP Inventor 2.....	40
<b>Figura 28.</b> Código Arduino para conectar con smartphone por medio de Bluetooth. ....	41
<b>Figura 29.</b> Diagrama de procesos de bloques de LabView.....	42
<b>Figura 30.</b> Indica proceso de lectura del archivo de código G.....	43
<b>Figura 31.</b> Configuración del puerto serial. ....	45
<b>Figura 32.</b> Proceso de acciones. ....	47
<b>Figura 33.</b> Caso por defecto. ....	48
<b>Figura 34.</b> Lectura de comandos. ....	49
<b>Figura 35.</b> Acción encerrar.....	49
<b>Figura 36.</b> Acción limpiar gráfico. ....	50
<b>Figura 37.</b> Movimiento manual de los ejes. ....	51
<b>Figura 38.</b> Acción sin movimiento. ....	52
<b>Figura 39.</b> Selección de la distancia de avance de los ejes. ....	53
<b>Figura 40.</b> Almacenamiento de posición de los ejes. ....	55
<b>Figura 41.</b> Lectura de código G.....	57
<b>Figura 42.</b> Lectura de código sin datos.....	58
<b>Figura 43.</b> Lectura del código G.....	58

<b>Figura 44.</b> Lectura del puerto serial.....	61
<b>Figura 45.</b> No detecta datos.....	61
<b>Figura 46.</b> Lectura de datos.....	61
<b>Figura 47.</b> Verificación de procesos.....	62
<b>Figura 48.</b> Control de número de líneas.....	63
<b>Figura 49.</b> Bloque de lectura y gráfico.....	65
<b>Figura 50.</b> Almacenamiento de coordenadas.....	66
<b>Figura 51.</b> Diagrama de flujo de código de Arduino.....	70
<b>Figura 52.</b> Funciones de Arduino.....	82
<b>Figura 53.</b> Función serial_data.....	83
<b>Figura 54.</b> Función leer_línea.....	85
<b>Figura 55.</b> Función distancia_mm.....	87
<b>Figura 56.</b> Función CalcMaxSpeed.....	88
<b>Figura 57.</b> Función setFeedrate.....	89
<b>Figura 58.</b> Función de posiciones del motor.....	91
<b>Figura 59.</b> Función line.....	92
<b>Figura 60.</b> Función line_safe.....	93
<b>Figura 61.</b> Función atan3.....	95
<b>Figura 62.</b> Función arc.....	96
<b>Figura 63.</b> Implementación de un sistema de manejo lineal para una mesa de control numérico computarizado.....	100
<b>Figura 64.</b> Eje lineal “X”.....	101
<b>Figura 65.</b> Eje lineal “Y”.....	102

<b>Figura 66.</b> Eje lineal “Z”.....	103
<b>Figura 67.</b> Estructura Final.....	104
<b>Figura 68.</b> Cotas Horizontales.....	104
<b>Figura 69.</b> Cotas Frontales. ....	104
<b>Figura 70.</b> Diagrama de flujo de proceso.....	105
<b>Figura 71.</b> Panel de control LabVIEW. ....	106
<b>Figura 72.</b> Panel de control Universal G-Code Sender V1.0.6. ....	107

## INDICE DE TABLAS

	<b>PÁGINA</b>
<b>Tabla 1.</b> Especificaciones de la fuente del plasma. ....	8
<b>Tabla 2.</b> Secuencia de conmutación de bobinas, paso completo doble. ....	11
<b>Tabla 3.</b> Secuencia de conmutación de bobinas, paso completo Básica.....	12
<b>Tabla 4.</b> Secuencia de conmutación de bobinas, medio paso.....	13
<b>Tabla 7.</b> Especificaciones del Driver A4988.....	18
<b>Tabla 8.</b> Especificaciones Arduino UNO.....	19
<b>Tabla 9.</b> Especificaciones del módulo bluetooth para Arduino HC-06. ....	22
<b>Tabla 10.</b> Proceso de corte con plasma.....	29
<b>Tabla 11.</b> Comandos GRBL. ....	31
<b>Tabla 12.</b> Movimientos mecánicos. ....	108
<b>Tabla 13.</b> Control de Movimientos. ....	109
<b>Tabla 14.</b> Calibración driver A4988. ....	110
<b>Tabla 15.</b> Calibración Micro pasos del motor .....	111
<b>Tabla 16.</b> Interconexiones para movimientos.....	112
<b>Tabla 17.</b> Tabla de costos.....	113

## RESUMEN

El presente trabajo describe el diseño, construcción e implementación de una máquina de Control Numérico Computarizado “CNC” que realizara la función de cortar diseños mecánicos con plasma. Este proyecto se ha desarrollado con el objetivo de realizar piezas que son indispensable para crear nuevos proyectos en la industria.

Ya que todas las máquinas CNC tienen el mismo principio en el control de los ejes X, Y, Z; se puede integrar un plasma de corte de metal. En el documento se describe cómo funciona alimentación y control de la máquina por medio de la plataforma de un Sistema Embebido Arduino, montar en la máquina el módulo CNC SHIELD, El cual recibirá las diferentes señales para controlar los Drivers de los motores.

Se ha integrado un módulo Bluetooth para Arduino HC-06, para controlar la alimentación de la máquina CNC por medio de una aplicación implementada en un celular con sistema operativo Android.

Palabras Claves: Shield, CNC, Driver, Sistema Embebido, Plasma, Arduino.

## **ABSTRACT**

The present work describes the design, construction and implementation of a Computerized CNC "CNC" machine that performs the function of cutting mechanical objects with plasma. This project has been developed with the aim of making pieces that are indispensable to create new projects in the industry.

Since all the machines CNC have the same principle in the control of the axes X, Y, and Z; a metal cutting plasma can be integrated. The document describes how power and control of the machine works through the platform of an Arduino Embedded System, mounting on the machine the CNC SHIELD module, which will receive the different signals to control the drivers of the motors.

A Bluetooth module for Arduino HC-06 has been integrated to control the power supply of the CNC machine through an application implemented in a cell phone with Android operating system.

Keywords: Shield, CNC, Driver, Embedded System, Plasma, Arduino.

# INTRODUCCIÓN

## ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO

Las máquinas CNC nacieron por la necesidad de tener que automatizar procesos que son complicados realizarlos manualmente, nace del concepto Control Numérico Computarizado. Jhon Parsons con el apoyo del MIT INSTITUTO TECNOLÓGICO FOR MASSACHUSETTS, desarrollaron las primeras máquinas CNC en tres ejes que funcionaban por medio de lectura de datos con tarjetas perforadas.

En un inicio las máquinas funcionaban con válvulas de vacío, luego fueron remplazadas por transistores que dieron un gran salto en la revolución en la revolución de la tecnología.

Con los microprocesadores se mejoró la velocidad y precisión de las máquinas de control numérico computarizado.

Los sistemas embebidos han logrado controlar muchas máquinas que funcionan en tiempo real, esto permite implementar nuevas funciones al tener un software cada vez más flexible. El hardware de los sistemas embebidos controla dispositivos externos, facilitando la integración de pantallas táctiles, programación en menor tiempo y operación eficiente.

En nuestro país como muchos otros en vías de desarrollo, tenemos una amplia industria en metalmecánica, en las cuales se emplean máquinas convencionales tales como tornos, fresadoras, prensas las cuales carecen de flexibilidad, se requiere de un personal debidamente capacitado que están expuestos a cambios drásticos en la industria; como la implementación de la tecnología de automatización en máquinas con control numérico computarizado.

## **PLANTEAMIENTO DEL PROBLEMA**

La implementación de una máquina CNC es ampliamente utilizada en la industria para la creación y moldeo de piezas complejas de manera rápida y precisa. No obstante, la implementación de este tipo de tecnología en la industria resulta sumamente costosa, el mantenimiento, la capacitación para todas las personas que lo utilice.

Es importante que los individuos que intervienen en la toma de decisiones y acciones que afectan directamente en el proceso productivo, adquieran un conocimiento con respecto a la tecnología que pone a disposición de la Automatización Integrada. Por lo tanto, la implementación de este equipo resulta importante para mejorar la producción de la industria nacional.

## **FORMULACIÓN DEL PROBLEMA**

¿Cómo diseñar e implementar un sistema de corte por plasma en una máquina de control numérico computarizado “CNC” controlada por sistema embebidos para la creación y moldeo de piezas?

## **JUSTIFICACIÓN**

La construcción de esta máquina responde a la necesidad que tienen los talleres mecánicos para realizar sus piezas metálicas en menos tiempo y a mayor escala.

Este tipo de máquinas únicamente se consiguen en el mercado a un precio poco accesible para las empresas de industrias poco desarrolladas, las máquinas más económicas que se consiguen en el mercado ecuatoriano de este tipo son importadas de China y Japón a un precio aproximado de \$20.000 pero, aunque no son tan costosas en un comienzo no poseen suficientes garantías ni soporte para su implementación generado el pago por mantenimiento e importación de repuestos.

Con respecto a los trabajadores que cortan manualmente con el plasma: No les quita o desplaza de su área laboral, sino más bien les permite capacitarse y ser más técnicos al usar esta máquina herramienta, además disminuye el Riesgo laboral al estar lejos de la antorcha del plasma, evitando exposición a la radiación del arco eléctrico, así como también a las quemaduras por alta temperatura del arco y proyección de partículas calientes a alta velocidad (chispas).

## **OBJETIVO GENERAL**

- Implementar un Sistema de Manejo Lineal para una mesa de Control Numérico Computarizado.

## **OBJETIVO ESPECÍFICO**

- Desarrollar un software de comunicación PC-Máquinas para el control numérico computarizado.
- Elaborar un App que permita controlar la alimentación de la máquina por medio de un Smartphone mediante bluetooth.
- Diseñar el programa en el software LabView en lenguaje de bloques en la cual se demostrará los movimientos en los ejes X, Y, Z.
- Implementar una interfaz de comunicación entre Universal G-Code Sender y Arduino por medio de programación para demostrar movimientos en los ejes X, Y, Z.
- Realizar pruebas de operación del control de posición de la herramienta (plasma) en placas de acero de diferente espesor (de 0.6 a 8 mm).
- Fabricar un prototipo mecánico de una máquina automatizada CNC.
- Ajustar la interfaz cambiando parámetros de programación para la operación de la multi-herramienta de la máquina.

# CAPÍTULO I

## 1. FUNDAMENTACIÓN TEÓRICA

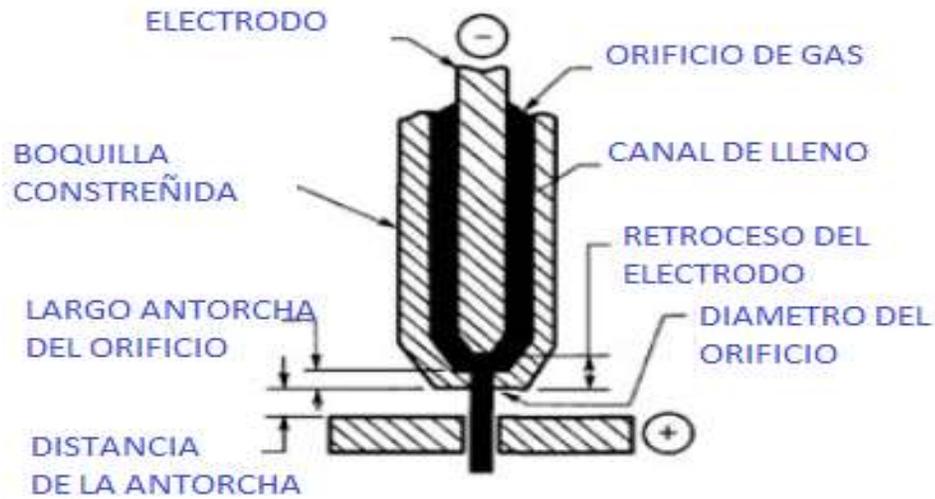
En este capítulo se aportará información acerca de las máquinas de control numérico computarizado, en la cual se analizan los fundamentos para desarrollar los diferentes sistemas mecánicos, eléctricos y de control a utilizar.

### 1.1. PLASMA

En la naturaleza se pueden encontrar tres tipos de materia sólido, líquido, gaseoso y el cuarto estado de la materia que es el plasma. A muy elevadas temperaturas, los electrones tienen suficiente energía como para salir de su órbita del núcleo del átomo, generando iones de carga positiva.

El plasma es el estado en el que se encuentran las estrellas por su elevada temperatura. En la atmósfera terrestre solo podemos conseguir el plasma por medios artificiales.

Este estado de la materia se produce bajo gran presión y temperatura, la cual puede ser aprovechada para generar energía, mediante los reactores de fusión, pero la tecnología necesaria para confinar el plasma mediante campos magnéticos, para alcanzar las temperaturas del estado plasma, no es 100% fiable, está aún en desarrollo.



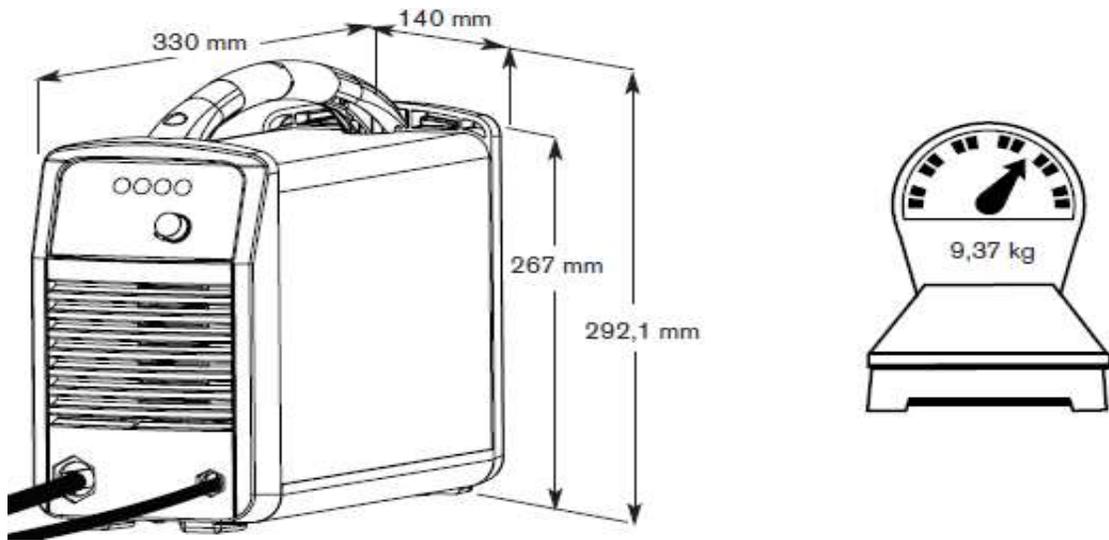
**Figura 1.** Proceso de funcionamiento del plasma. (sapiensman, 2012)

## 1.2. POWERMAX 30 AIR

Este plasma es el óptimo a usarse ya que hace posible el corte uniforme con la optimización del flujo de aire proveniente del compresor interno compacto. Con la prestación arco piloto continuo, corta rápidamente rejilla de metal o metal oxidado, sin tener que disparar de nuevo el gatillo, permitiendo velocidades de corte rápidas y una calidad de corte superior.

## 1.3.DIMENSIONES Y PESO DE LA FUENTE DE ENERGÍA DEL PLASMA

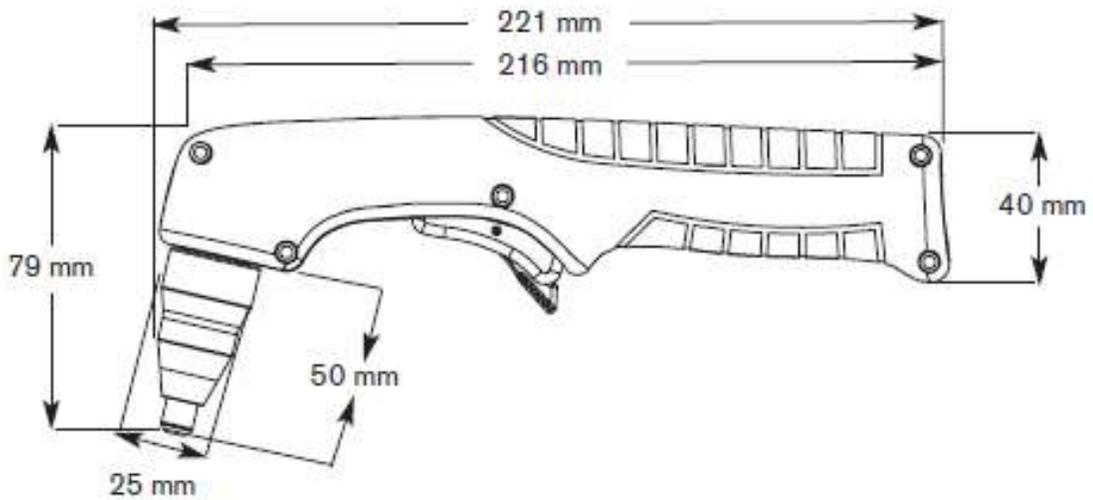
Las dimensiones de esta fuente son óptimas para incorporarle en la parte inferior de la mesa de corte CNC. La antorcha y mangueras no son muy pesadas para acoplarle al puente móvil de la mesa de corte CNC.



**Figura 2.** Medidas y peso de la fuente del plasma. (Hypertherm, 2016)

#### 1.4. DIMENSIONES DE LA ANTORCHA

La antorcha y las mangueras se ajustan en peso y medidas al acople del eje Z que se encuentra en el puente móvil de la mesa de corte CNC.



**Figura 3.** Medidas de la antorcha del plasma. (Hypertherm, 2016)

## 1.5. ESPECIFICACIONES

Nos permite conocer voltajes de entrada y salida de potencia que requiere para pleno funcionamiento, amperaje requerido y la temperatura óptima para el funcionamiento del plasma.

**Tabla 1.**

*Especificaciones de la fuente del plasma. (Hypertherm, 2016)*

<b>Voltajes de Entrada</b>	<b>120V - 240V, 50 / 60 Hz</b>
<b>Corriente de entrada a 2.5kW</b>	120V - 240V, 29 / 15 A
<b>Corriente de salida</b>	15 - 30 A
<b>Voltaje de salida nominal</b>	83 VCD
<b>Ciclo de trabajo a 40 °C</b>	120V - 240V, 20 / 30 %
<b>Voltaje en circuito abierto</b>	256 VCD
<b>Tipo fuente de energía</b>	Inversor – IGBT
<b>Requisito al motor</b>	5,5kW plena capacidad, Salida 30A

## 1.6. SISTEMA MECÁNICO.

En este sistema se dará información acerca de los componentes mecánicos con la que consta la máquina de control numérico computarizado.

## 1.7.MOTOR PASO A PASO

Los motores paso a paso (P-P) pueden verse como motores eléctricos sin sistema de conmutación. Típicamente, todas las bobinas del motor están en el estator y el rotor, un imán permanente o en el caso de motores de reluctancia variables, un bloque de algún material magnéticamente blando. Toda la conmutación debe ser manejada externamente por el controlador del motor, habitualmente los motores, controladores están diseñados para que el motor pueda ser mantenido en una posición o rotar en uno u otro sentido. La mayoría de estos motores pueden ser manejados a frecuencias de audio permitiendo un giro rápido con un controlador apropiado, pueden ser arrancados y parados en posiciones controladas.



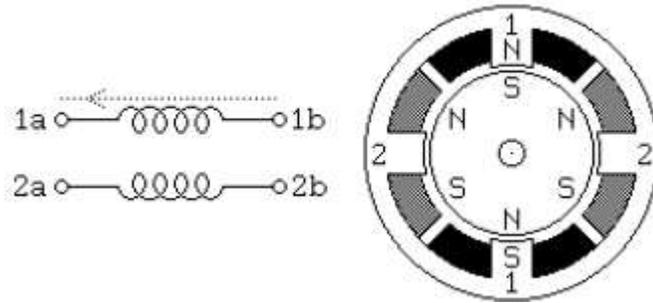
**Figura 4.** Motor paso a Paso. (González, 2002)

En los motores paso a paso se encuentran dos tipos:

### **1.7.1. BIPOLAR**

Estos tienen generalmente cuatro cables de salida. Necesitan ciertos trucos para ser controlados, debido a que requieren del cambio de dirección del flujo de corriente a través de las

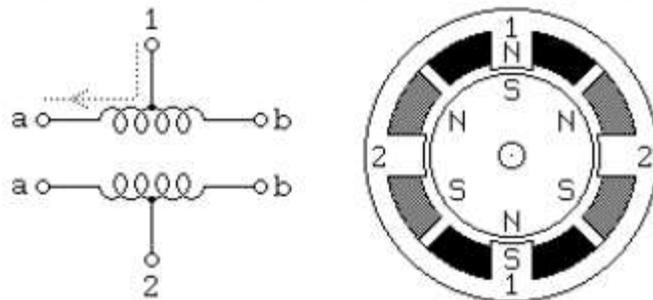
bobinas en la secuencia apropiada para realizar un movimiento. Esto hace que la controladora se vuelva más compleja y costosa. Su uso no es tan común como en el caso de los de tipo unipolar.



**Figura 5.** Estructura motor paso a paso bipolar. (González, 2002)

### 1.7.2. UNIPOLAR

Estos motores suelen tener 8, 6 o 5 cables de salida, dependiendo de su conexionado interno. Este tipo se caracteriza por ser más simple de controlar.



**Figura 6.** Estructura motor paso a paso unipolar. (González, 2002)

## 1.8. SECUENCIAS DE LOS MOTORES PASO A PASO UNIPOLARES

Existen tres tipos de secuencias para los motores paso a paso unipolares que se detallaran a continuación:

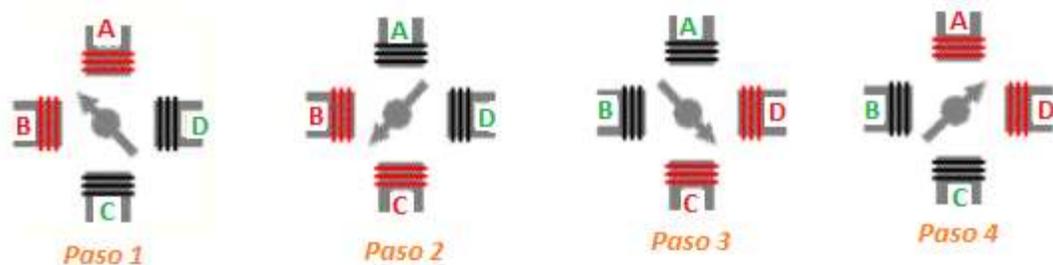
### 1.8.1. SECUENCIA DE PASO COMPLETO DOBLE:

Consiste en activar dos bobinas al mismo tiempo, de forma que el eje del motor se orienta al punto medio de las bobinas activadas. Avanzará siempre a un paso ya que siempre van a estar dos bobinas activas y dos bobinas desactivadas.

**Tabla 2.**

*Secuencia de conmutación de bobinas, paso completo doble.* (González, 2002)

PASO	BOBINA A	BOBINA B	BOBINA C	BOBINA D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1



**Figura 7.** Movimiento del rotor indicando la secuencia de paso completo doble. (González, 2002)

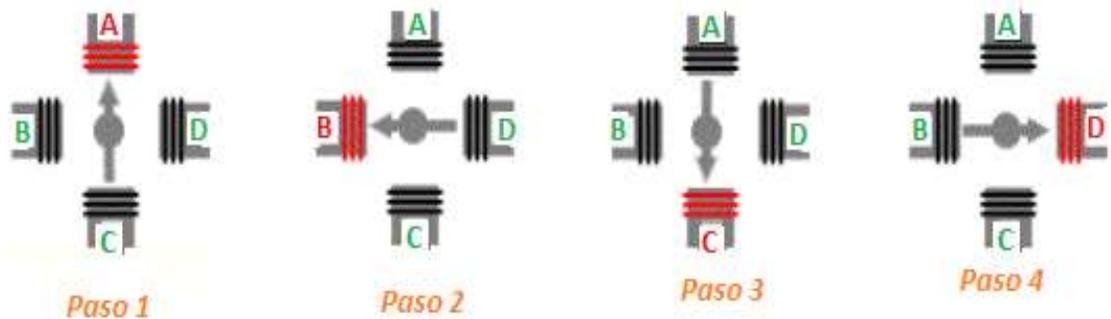
### 1.8.2. SECUENCIA DE PASO COMPLETO BÁSICA

Consiste en activar la bobina independientemente, de forma que el eje del motor de orienta de acuerdo con la bobina que se encuentra activa.

**Tabla 3.**

*Secuencia de conmutación de bobinas, paso completo Básica. (González, 2002)*

PASO	BOBINA A	BOBINA B	BOBINA C	BOBINA D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1



**Figura 8.** Movimiento del rotor indicando la secuencia de paso completo Básica. (González, 2002)

### 1.8.3. SECUENCIA DE MEDIO PASO

Consiste en combinar las secuencias anteriores, se obtiene un paso más corto, se le puede denominar medio paso, es decir que se activa primero una sola bobina, luego se activan dos bobinas para posicionar al eje entre las dos y así sucesivamente.

**Tabla 4.**

*Secuencia de conmutación de bobinas, medio paso. (Gonzáles, 2002)*

PASO	BOBINA 1	BOBINA 2	BOBINA 3	BOBINA 4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1



**Figura 9.** Movimiento del rotor indicando la secuencia de medio paso. (Gonzáles, 2002)

## 1.9. WANTAI MINI STEPPER 42BYGHW811

Este modelo de motor paso a paso es uno de los modelos óptimos para instalar en los equipos CNC, ya que es liviano, tiene poco consumo de potencia y el voltaje es variable de acuerdo con el consumo del driver a usar.



**Figura 10.** Motor wantai mini stepper 42BYGHW811. (pbcllinear, 2015)

### 1.9.1. ESPECIFICACIONES

Este motor paso a paso NEMA 17 es bipolar, tiene un ángulo de paso de  $1.8^\circ$  (200 pasos por vuelta) y cada bobinado es de 1.2 A - 4 V, capaz de cargar con 3.2 kg/cm (44 oz-in).

Es un motor muy robusto ampliamente utilizando en máquinas que usan control numérico computarizado, aunque este es ligeramente más potente del recomendado para garantizar la mejor fiabilidad.

### 1.9.2. CARACTERÍSTICAS

Se detallará el voltaje, torque, potencia y medidas de diferentes partes del motor.

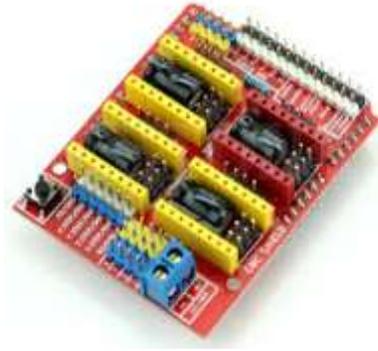
**Tabla 5.**

*Características del motor.* (pbcllinear, 2015)

<b>PESO</b>	350 gramos (13 oz)
<b>DIÁMETRO DEL EJE</b>	5 mm "D"
<b>LONGITUD DEL EJE</b>	25 mm
<b>PASOS POR VUELTA</b>	200 (1, 8°/paso)
<b>CORRIENTE</b>	1.2 Amperios por bobinado
<b>TENSIÓN</b>	4 V
<b>RESISTENCIA</b>	3.3 Ohm por bobina
<b>TORQUE</b>	3.2 kg/cm (44 oz-in)
<b>INDUCTANCIA</b>	2.8 mH por bobina

### 1.10. CNC SHIELD

El módulo CNC shield para arduino permite que sea fácil el desarrollo de proyectos CNC (Control Numérico Computarizado). Utiliza firmware open-source de Arduino para controlar cuatro motores paso a paso por medio de cuatro controladores de motor A4988.



**Figura 11.** Módulo CNC shield para arduino. (Kruger, 2013)

### ***1.10.1 CARACTERÍSTICAS***

Se describen los parámetros de cómo usar el CNC shield, se observa voltajes, corrientes y como se acopla con los drivers para permitir obtener el movimiento de los motores paso a paso.

**Tabla 6.**

*Características módulo CNC Shield para Arduino.* (Kruger, 2013)

<b><i>CARACTERÍSTICAS</i></b>	
<b>1</b>	Compatible con GRBL 0.8c. (Firmware de código abierto que se ejecuta en un Arduino UNO que convierte los comandos de código G en señales paso a paso.
<b>2</b>	Soporte de 4 ejes (X, Y, Z, A-Puede duplicar X, Y, Z o hacer un 4to eje completo con firmware personalizado usando los pines D12 y D13)
<b>3</b>	2 x topes finales para cada eje (6 en total)
<b>4</b>	Habilitación y dirección del cabezal
<b>5</b>	Activar refrigerante
<b>6</b>	Utiliza controladores escalables compatibles con Pololu A4988. (A4988, DRV8825 y otros)

- 
- 8** Los motores paso a paso se pueden conectar con conectores molex de 4 pines.
- 
- 9** Se ejecuta en 12-36V DC. (Por el momento sólo los controladores Pololu DRV8825 pueden manejar hasta 36V por lo que considere el voltaje de funcionamiento al alimentar
- 
- 10** Puentes para configurar el Micro-Stepping para los controladores paso a paso. (Algunos controladores como el DRV8825 pueden hacer hasta 1/32 micro-stepping)
- 

### **1.11. DRIVER A4988**

Los Drivers A4988 simplifican el manejo de motores paso a paso desde un procesador como Arduino. Permiten manejar los altos voltajes e intensidades que requieren estos motores, limitan la corriente que circula por el motor.

Para controlar únicamente se requiere dos salidas digitales, una para indicar el sentido de giro y otra para comunicar que queremos que el motor avance un paso. Además, permiten realizar microstepping, una técnica para conseguir precisiones superiores al paso nominal del motor.



**Figura 12.** Driver A4988, para controlar motores Paso a paso desde Arduino. (Llamas, 2016)

### 1.11.1 ESPECIFICACIONES

Se dará a conocer cómo obtener el pleno funcionamiento del driver, con que voltaje esta alimentado la potencia requerida y formulas.

**Tabla 7.**

*Especificaciones del Driver A4988. (Llamas, 2016)*

<b>Modelo</b>	<b>A4988</b>
<b>Color</b>	Rojo
<b>Intensidad máxima</b>	2 A
<b>Tensión máxima</b>	35 V
<b>Microsteps</b>	16
<b>Rs típico</b>	0.05, 0.1 o 0.2
<b>Fórmulas</b>	$I_{\max} = V_{\text{ref}} / ( 8 * R_s)$
	$V_{\text{ref}} = I_{\max} * 8 * R_s$

### 1.12. ARDUINO UNO

Arduino es una herramienta para hacer que los ordenadores puedan sentir y controlar el mundo físico a través de tu ordenador personal. Es una plataforma de desarrollo de computación física de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear software para la placa.

Puedes usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos.



**Figura 13.** Sistema en bebido Arduino UNO. (Galdós, 2010)

### ***1.12.1. ESPECIFICACIONES***

Se dará a conocer cómo obtener el pleno funcionamiento del sistema embebido Arduino, con que voltaje esta alimentado, la potencia requerida, entradas y salidas tanto análogos como digitales.

**Tabla 8.**

*Especificaciones Arduino UNO.* (Galdós, 2010)

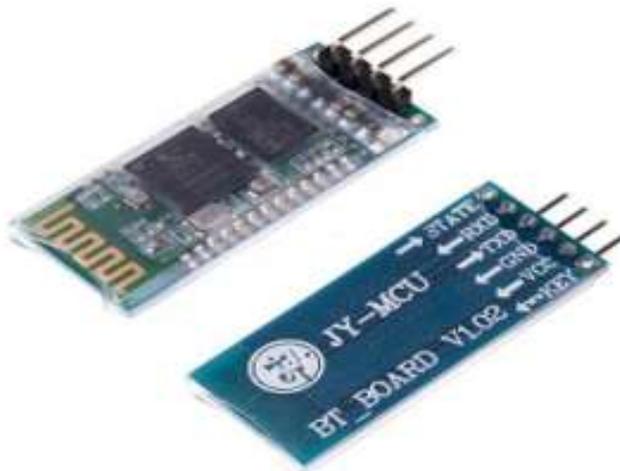
<b>Microcontroller</b>	ATmega328
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limits)</b>	6-20V
<b>Digital I/O Pins</b>	14 (of which 6 provide PWM output)
<b>Analog Input Pins</b>	6
<b>DC Current for I/O Pin</b>	40 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328)
<b>SRAM</b>	2 KB (ATmega328)

<b>EEPROM</b>	1 KB (ATmega328)
<b>Clock Speed</b>	16 MHz

### 1.13. MÓDULO BLUETOOTH PARA ARDUINO HC-06

EL módulo Bluetooth HC-06 viene configurado de fábrica como esclavo y no se lo puede cambiar, pero otras características se pueden configurar usando comandos AT, estas características vienen por defecto con valores predeterminados que se muestran a continuación:

- Nombre por defecto: “linvor” o “HC-06”
- Código de emparejamiento por defecto: 1234
- La velocidad por defecto (baud rate): 9600



**Figura 14.** Módulo bluetooth para arduino HC-06. (Electrónicos Caldas, 2016)

EL módulo HC-06 tiene dos estados los cuales es importante conocer:

### Modo AT (Desconectado)

- Entra a este modo tan pronto alimentas el módulo, cuando no se ha establecido una conexión bluetooth con ningún otro dispositivo
- EL LED del módulo está parpadeando (frecuencia de parpadeo del LED es de 102ms)
- En este modo es cuando se debe enviar los comandos AT en caso se quiera configurar, si se envían otros datos diferentes a los comandos AT el HC-06 los ignorará.

### Modo Conectado

- Entra a este modo cuando se establece una conexión con otro dispositivo bluetooth.
- El LED permanece prendido sin parpadear
- Todos los datos que se ingresen al HC-06 por el Pin RX se transmiten por bluetooth al dispositivo conectado, de modo que los datos recibidos se devuelven por el pin TX. La comunicación es transparente.
- En este modo el HC-06 no puede interpretar los comandos AT.

#### ***1.13.1. ESPECIFICACIONES***

Podemos observar las características para conectar la aplicación a realizarse correctamente con el Bluetooth HC-06.

**Tabla 9.**

*Especificaciones del módulo bluetooth para Arduino HC-06. (Electrónicos Caldas, 2016)*

<b>Chip de radio</b>	CSR BC417143
<b>Frecuencia</b>	2.4 GHz, banda ISM
<b>Modulación</b>	GFSK (Gaussian Frequency Shift Keying)
<b>Antena de PCB</b>	PCB incorporada
<b>Potencia de emisión</b>	$\leq 6$ dBm, Clase 2
<b>Alcance</b>	5m a 10m
<b>Sensibilidad</b>	$\leq -80$ dBm a 0.1% VER
<b>Velocidad</b>	Asincrónica: 2 Mbps (max.)/160 kbps, sincrónica: 1 Mbps/1 Mbps
<b>Seguridad</b>	Autenticación y encriptación (Password por defecto: 1234)
<b>Perfiles</b>	Puerto serial Bluetooth
<b>Módulo</b>	Montado en tarjeta con regulador de voltaje y 4 pines suministrando acceso a VCC, GND, TXD, y RXD
<b>Consumo de corriente</b>	30 mA a 40 mA
<b>Voltaje de operación</b>	3.6 V a 6 V
<b>Dimensiones totales</b>	1.7 cm x 4 cm aprox.
<b>Temperatura de operación</b>	-25 °C a +75 °C

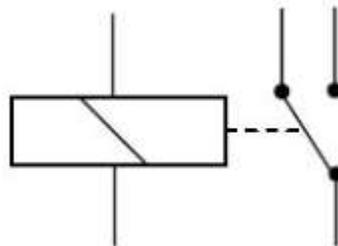
## 1.14. RELÉ ARDUINO

Es un interruptor que podemos activar mediante una señal eléctrica. En su versión más simple es un pequeño electro-imán que cuando lo excitamos mueve la posición de un contacto eléctrico de conectado a desconectado o viceversa.



**Figura 15.** Módulo Relé para Arduino. (Electrónicos Caldas, 2016)

El símbolo del relé muestra la bobina y en este caso, un accionado que conmuta entre dos contactos, pero también existen relés de múltiples contactos. Mediante una señal de control de poca intensidad que excite la bobina podemos conmutar grandes tensiones o intensidades.



**Figura 16.** Símbolo del módulo relé. (Electrónicos Caldas, 2016)

## 1.15. LABVIEW

LabView constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos. Las ventajas que proporciona el empleo de LabView se resumen en las siguientes:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de cuatro a diez veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

Los programas desarrollados mediante LabView se denominan **Instrumentos Virtuales (VIs)**, porque su apariencia, funcionamiento imitan los de un instrumento real. Sin embargo, son análogos a las funciones creadas con los lenguajes de programación convencionales.

Los VIs tienen una parte interactiva con el usuario y otra parte de código fuente, el cual aceptan parámetros procedentes de otros VIs. Todos los VIs tienen un panel frontal para controlar

los movimientos, también un diagrama de bloques donde se realizará todo el programa. Las paletas contienen las opciones que se emplean para crear y modificar los VIs.



**Figura 17.** Logo programa Labview, National Instruments. (esi2 , 2009)

## CAPITULO II

### 2. PROPUESTA

#### 2.1 ANÁLISIS DEL DISEÑO DE LA MÁQUINA DE CORTE CNC PROPUESTA

El diseño de la máquina automatizada propuesta se obtuvo en base a un análisis de estudio de diferentes máquinas de control numérico computarizado las cuales se encuentran en el mercado del sector industrial.

Anteriormente el operario realizaba un trabajo manual por cada corte de pieza en la cual tiene que medir y trazar pieza por pieza para luego ejecutar manualmente el corte con el Plasma.



**Figura 18.** Corte manual con plasma (Codinter, 2018)

La operación de corte con plasma es una parte de las diferentes operaciones para la manufactura industrial, el operario al realizar el trazado y medida de las piezas a cortar, demora a las otras operaciones que prosiguen a la operación de corte.

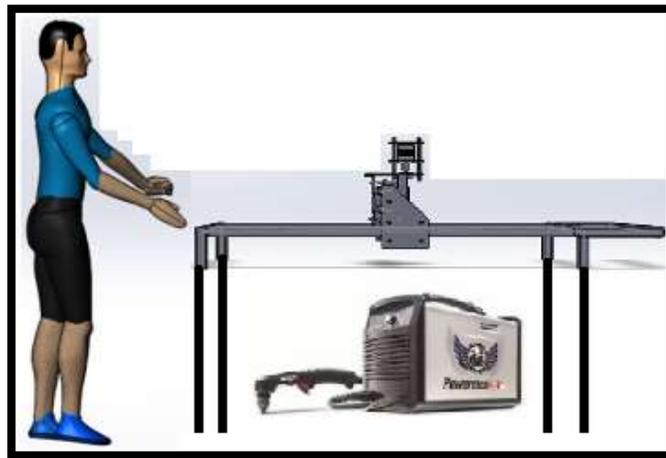
Por lo tanto, ante el problema mencionado anteriormente se diseñó una máquina automatizada que optimice los cortes de planchas metálicas, reduciendo los tiempos.

## 2.2. DISEÑO DE LA MÁQUINA AUTOMATIZADA:

El diseño de la máquina automatizada se basó en el uso de tomas de medidas de diversas máquinas de control numérico computarizado con el objetivo de proteger la integridad física del operario y optimizar la operación de corte de piezas metálicas con plasma.

### 2.2.1. La máquina automatizada tiene las siguientes dimensiones:

- Ancho: 756 mm, 75,6 cm, 0,756 m
- Largo: 1010 mm, 101 cm, 1,01 m
- Altura total: 1.20 m
- Altura de la base a la mesa de trabajo: 1 m



**Figura 19.** Relación Operario - Máquina

Como se puede observar la máquina tiene una disposición horizontal la cual facilita la manipulación de la plancha metálica, por lo cual se pueden colocar fácilmente en la mesa de corte con plasma.

Posee tres desplazamientos en sus 3 ejes X, Y, Z, para que se pueda realizar el corte de la plancha de metal.

### ***2.2.2. La máquina automatizada propuesta consta de las siguientes partes:***

- Perfiles estructurales de aluminio.
- Perfiles cuadrados de acero inoxidable.
- Ejes lineales de desplazamiento.
- Carros de desplazamiento.
- Motor paso a paso.
- Computadora de control de la máquina automatizada.

## **2.3. VENTAJAS DE LA MÁQUINA AUTOMATIZADA PROPUESTA**

### ***2.3.1. Ventajas:***

- La máquina automatizada propuesta, está diseñada para que su traslado sea sencillo y fácil.
- La máquina automatizada propuesta, es totalmente desmontable y de fácil traslado.
- La máquina automatizada propuesta, es de fácil uso en su manipulación.

## 2.4. DESCRIPCIÓN DE LA OPERACIÓN DE CORTE PROPUESTO

En el siguiente cuadro se aprecia las operaciones necesarias para realizar el corte de una plancha metálica, con la máquina automatizada propuesta:

**Tabla 10.**

*Proceso de corte con plasma.*

---

<b>PROCESO DE CORTE DE UNA PIEZA CON UNA MAQUINA CNC DE CORTE CON PLASMA</b>	
<b>1</b>	Posicionar la plancha metálica a la máquina de corte
<b>2</b>	Vectorizar y crear código G de la pieza metálica a realizar
<b>3</b>	Encender Máquina
<b>4</b>	Inicia proceso de corte
<b>5</b>	Parar la máquina
<b>6</b>	Retirar las piezas cortadas
<b>7</b>	Enfriamiento de pieza
<b>8</b>	Fin de proceso

---

## 2.5. ESPECIFICACIÓN Y DETALLE DEL SOFTWARE REQUERIDO Y SU PROGRAMACIÓN

El software que utiliza para poder realizar los cortes de planchas metálicas de forma automatizada es: Universal G-Code Sender, LabView, Arduino.

### ***2.5.1. Lenguaje G-CODE***

La forma más o menos estándar de comunicarse con un robot es usando el lenguaje G Code. Es un sencillo sistema de texto que permite los movimientos de la máquina. Es uno de los formatos más veteranos, ya que su primera versión se remonta a los años 50.

### ***2.5.2. CNC y G-Code***

Las máquinas robotizadas se denominan CNC vienen de “Computador de Control Numérico”. Es la forma común para no decir “máquina robotizada con motores”.

El lenguaje universal de las máquinas CNC se llama G-Code. Es un fichero de texto con todos los movimientos de la máquina.

## **2.6. CONFIGURAR GRBL**

Para comunicarnos con el GRBL, se puede hacer directamente a través de un puerto serial. Pero si preferimos algo más gráfico, se debe usar el programa **Universal-G-Code-Sender**.

Esta es una aplicación en Java que nos permite enviar ficheros G-Code e interactuar con nuestra máquina CNC de forma sencilla.

GRBL interpreta directamente G-CODE, pero para configurar las máquinas CNC, el intérprete incluye una serie de comandos especiales.

### 2.6.1. Comandos GRBL

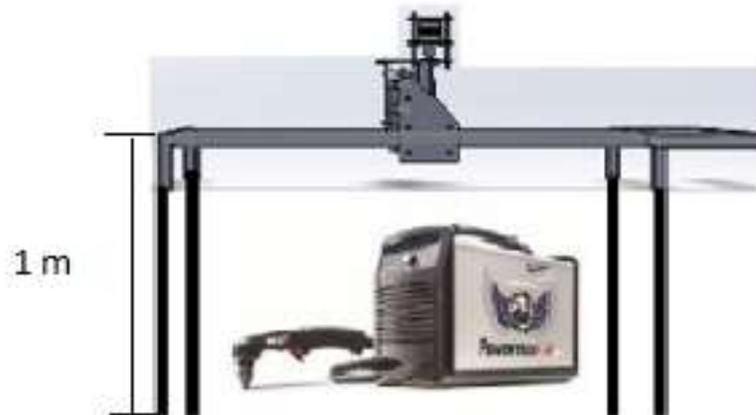
**Tabla 11.**

*Comandos GRBL.*

LISTA DE COMANDOS GRBL	
1	<b>\$\$</b> (mostrar los parámetros de GRBL)
2	<b>\$#</b> (mostrar los parámetros #)
3	<b>\$G</b> (mostrar el estado del parseador de GCODE)
4	<b>\$I</b> (mostrar la información del firmware, build, etc.)
5	<b>\$N</b> (mostrar los bloques de inicio)
6	<b>\$x=value</b> (almacenar un valor en la EEPROM)
7	<b>\$Nx=line</b> (almacenar un bloque de inicio)
8	<b>\$C</b> (comprobar el modo actual de GCODE)
9	<b>\$X</b> (desactivar el bloqueo de alarma)
10	<b>\$H</b> (hacer un ciclo homing)
11	<b>!</b> (pausar el avance)
12	<b>~</b> (reiniciar un ciclo o el continuar después de la pausa)
13	<b>?</b> (mostrar el estado actual, posición, etc.)
14	<b>ctrl-x</b> (reiniciar GRBL)

## 2.7. ERGONOMÍA APLICADA A LA MÁQUINA AUTOMATIZADA PROPUESTA

Los beneficios de la máquina se centran en su diseño ergonómico, el cual proporciona al operario una posición ergonómica frente a la máquina y no necesita de la presencia del operario durante todo el trabajo.



**Figura 20.** Ergonomía aplicada a una máquina CNC.

La máquina tiene una altura de 1 metro de la base a la mesa de trabajo lo que ayuda al operario durante la colocación del material ya que el esfuerzo que realiza no es tan grande ya que el material se encuentra sostenido a una altura aceptable para el cuerpo.

## 2.8. DISEÑO PARA SEGURIDAD

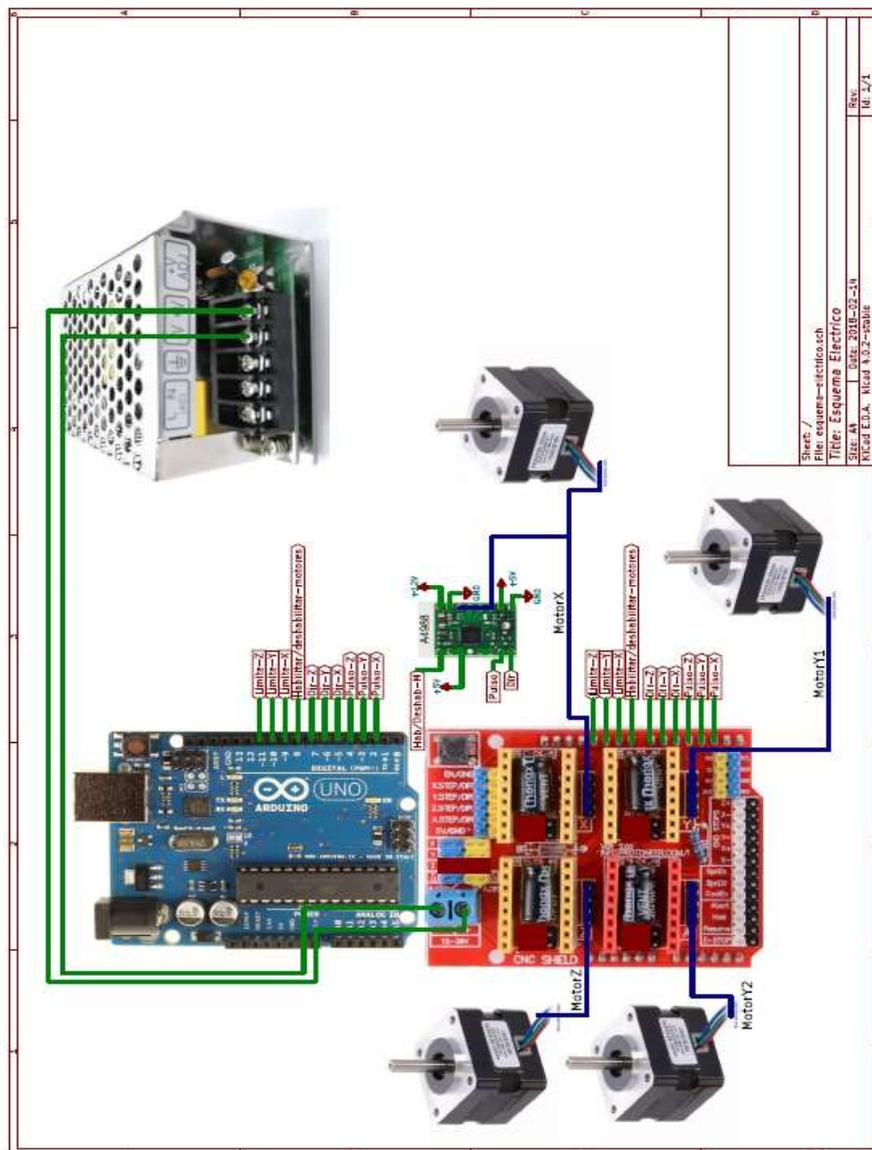
El diseño de la máquina y del sistema de movimiento fue pensado de tal forma que se cumplieran con las normas de seguridad básicas de funcionamiento, el cableado fue dispuesto en canaletas flexibles, las correas, piezas móviles no están al alcance del operario, la rigidez de la máquina y las velocidades de operación mitigan el riesgo de sufrir un accidente.

## CAPITULO III

### 3.1. DIAGRAS DE INTERCONEXIÓN DE DISPOSITIVOS.

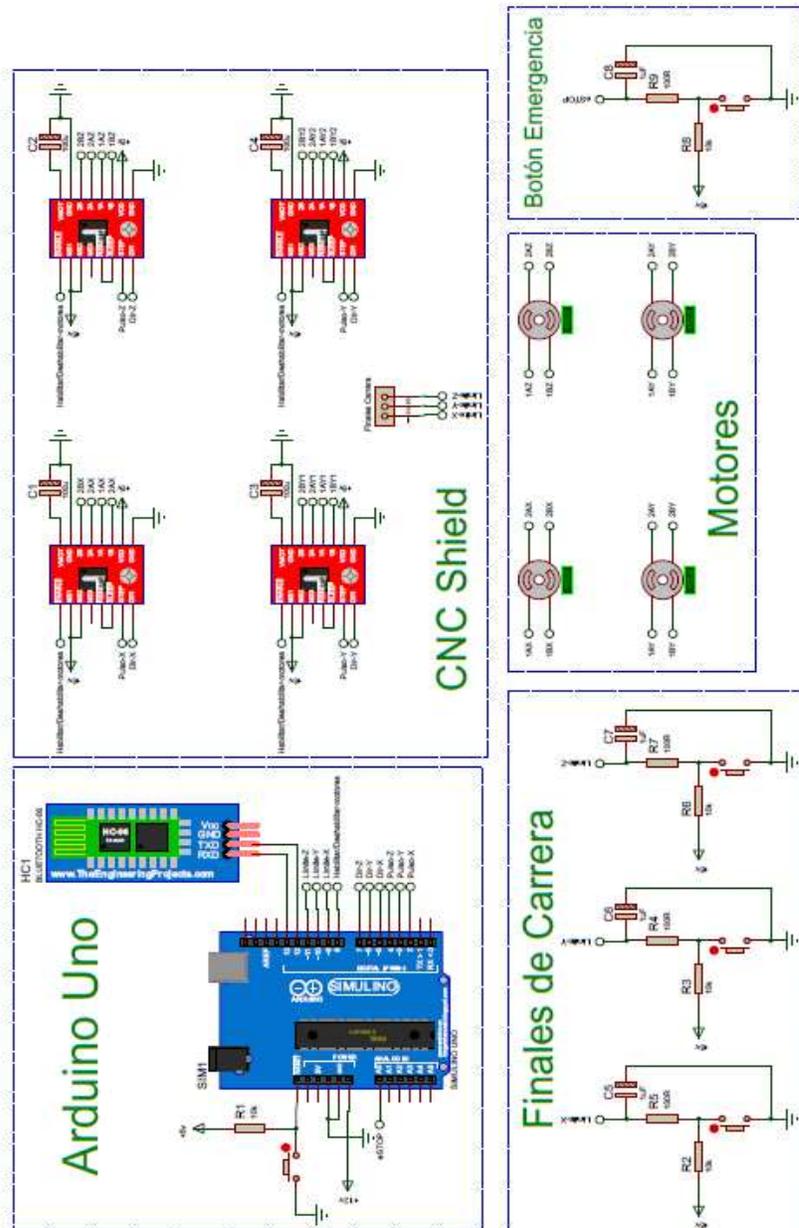
#### 3.1.1. DIAGRAMA ELÉCTRICO

Se puede observar la distribución de la alimentación, donde se puede apreciar los cambios de voltajes necesarios para el funcionamiento de cada uno de los dispositivos implementados.



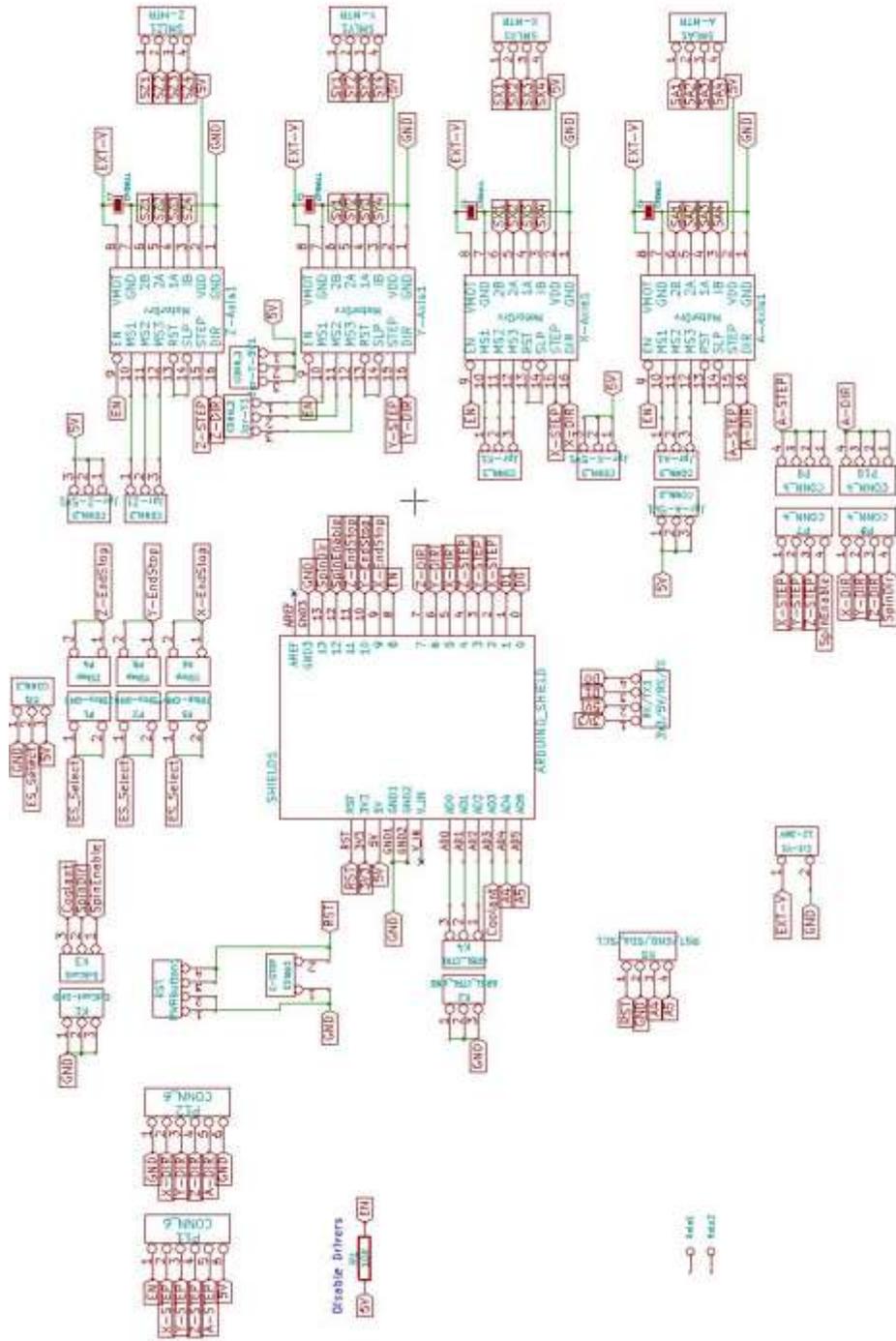
### 3.1.2. DIAGRAMA ELECTRÓNICO

Se observa la implementación de la máquina desde Arduino, CNC shield, drivers, motor paso a paso, para obtener los movimientos de los ejes X, Y, Z.



### 3.2. DIAGRAMA DEL SHIELD

Se observa la distribución desde el CNC shield hasta los drivers para controlar los motores paso a paso.



### 3.2.1. PCV FRONTAL SHIELD

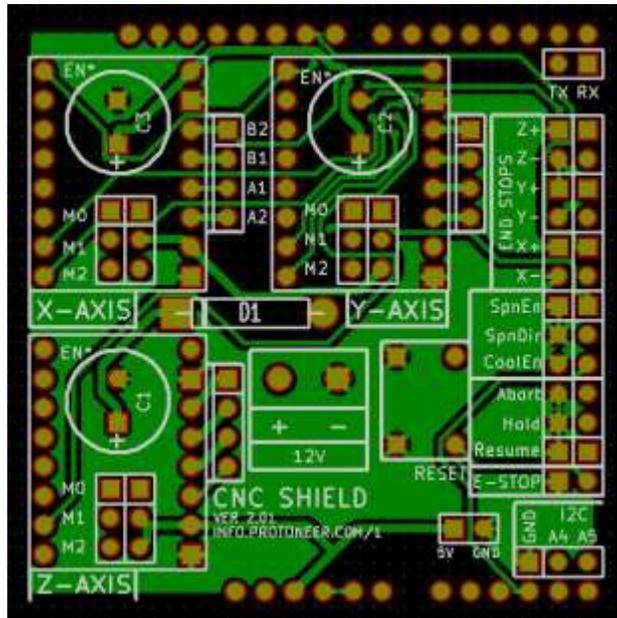


Figura 21. PCV frontal shield

### 3.2.2. PCV TRASERO SHIELD

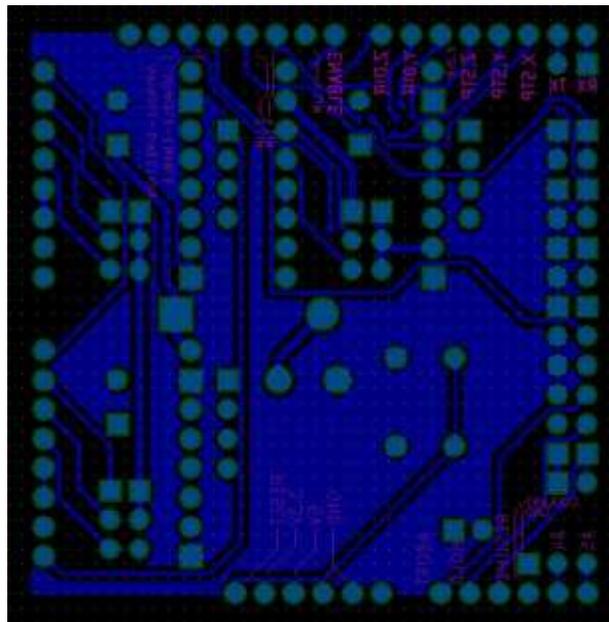
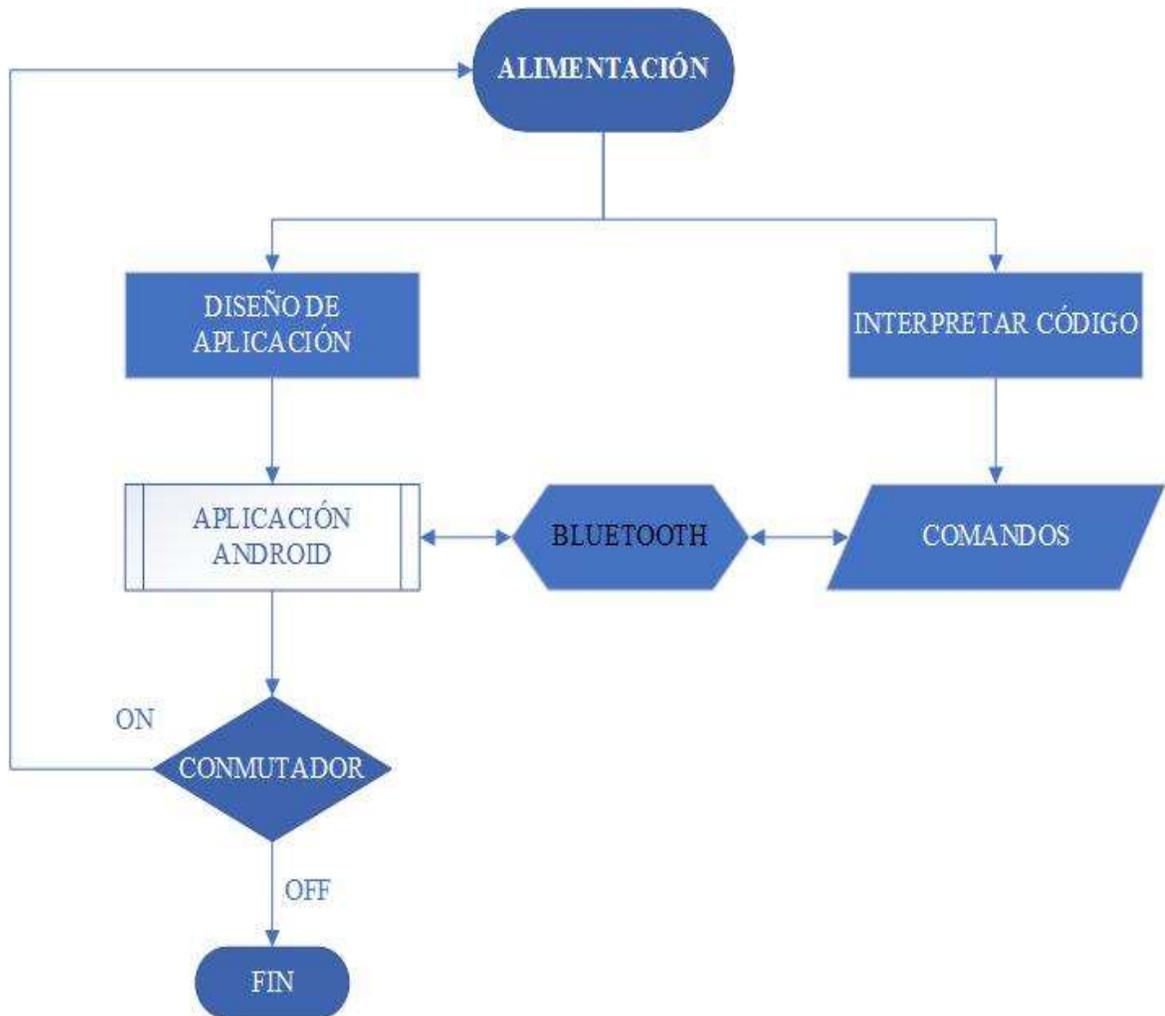


Figura 22. PCV trasero shield

### 3.3. DESARROLLO

#### 3.3.1. DIAGRAMA FLUJO DE PROCESOS DE DISEÑO DE APLICACIÓN PARA ALIMENTACIÓN DE UNA FUENTE AC – DC.



**Figura 23.** Diagrama de flujo de proceso de aplicación con bluetooth para controlar fuente de poder.

### 3.3.2. DESARROLLO DE UNA APLICACIÓN PARA ENCENDER UNA FUENTE PARA UNA MESA CNC POR MEDIO DE BLUETOOTH

La aplicación se realizó en línea por medio de la página APP Inventor 2, que permite la conexión entre un dispositivo Smartphone con sistema operativo Android y el sistema embebido Arduino Uno.

El Arduino Uno se encuentra conectado a un módulo relé de Arduino y un módulo Bluetooth CH-06 los cuales permiten energizar la fuente de poder que alimenta la máquina de corte con plasma CNC.

### 3.3.3. PROGRAMACIÓN DE BLOQUES EN APP INVENTOR 2

#### 3.3.3.1. HABILITAR EL MÓDULO DE BLUETOOTH

Activa el botón del bluetooth “CONECTAR” (ListPicker1), que apenas se abre la aplicación, sin que se presione ningún botón (Before Picking), el dispositivo busca la lista de equipos con bluetooth conectados en el entorno (Cliente Bluetooth 1) y genera una lista de direcciones con los nombres (Addresses And Names), para que luego seleccione con cual dispositivo conectarse.



**Figura 24.** Activación de Bluetooth.

### 3.3.3.2. SELECCIONAR DISPOSITIVO

Al presionar el botón de Bluetooth de la aplicación “CONECTAR” llama al listado generado de clientes en el ítem anterior y permite la selección del dispositivo con el cual conectarse. (list picker selection).



Figura 25. Selección de Dispositivo

### 3.3.3.3. BLOQUES PARA CREAR BOTÓN DE ENCENDER Y APAGAR FUENTE

Por medio de estos bloques de programación se generan los botones que formaran parte de la aplicación de encendido y apagado que se instaló en el smartphone, que se visualiza por medio de dos botones. Que permite seleccionar cual señal mandar, la del botón 1 “ENCENDER FUENTE DE PODER” o botón 2 “APAGAR FUENTE DE PODER”.

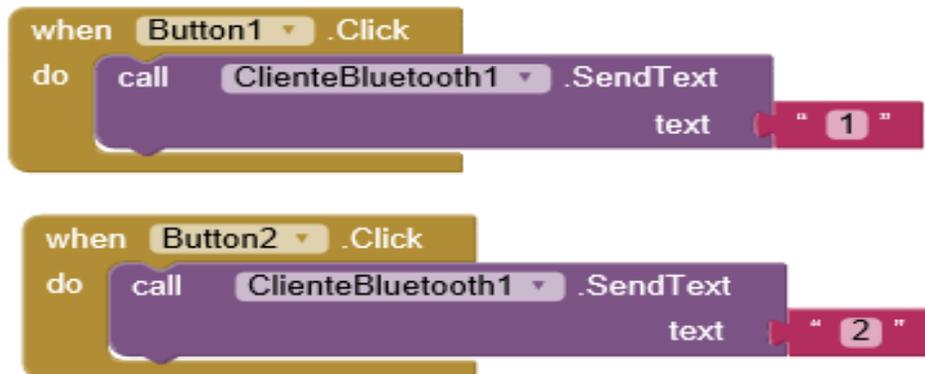


Figura 26. Botones de encendido y apagado

### 3.3.3.4. APLICACIÓN TERMINADA

Una vez desarrollada la aplicación en APP inventor 2 permite descargar la aplicación al smartphone por medio de sistema operativo Android.



**Figura 27.** Aplicación terminada en la página APP Inventor 2

### 3.3.3.4. COMUNICACIÓN BLUETOOTH CON ARDUINO

Por medio de los comandos descritos en la **Figura 26**, obtenemos la comunicación entre el smartphone y el arduino. Con la aplicación instalada en el smartphone se puede activar o desactivar el relé, en cual permite encender o apagar la fuente de poder la cual permite alimentar la máquina Automatizada de corte.

## CNC\_BT

```
1 int led13=13;
2 int estado=0;
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(led13,OUTPUT);
7 }
8
9 void loop(){
10  if(Serial.available()>0){
11     estado = Serial.read();
12  }
13  if (estado =='1'){
14     digitalWrite(led13,HIGH);
15  }
16  if(estado=='2'){
17     digitalWrite(led13,LOW);
18  }
19 }
20 |
```

**Figura 28.** Código Arduino para conectar con smartphone por medio de Bluetooth.

### 3.3.4. DIAGRAMA DE FLUJO DE PROCESOS DE LOS BLOQUES DE LABVIEW

Se observa los pasos realizados para la programación implementada en Labview.

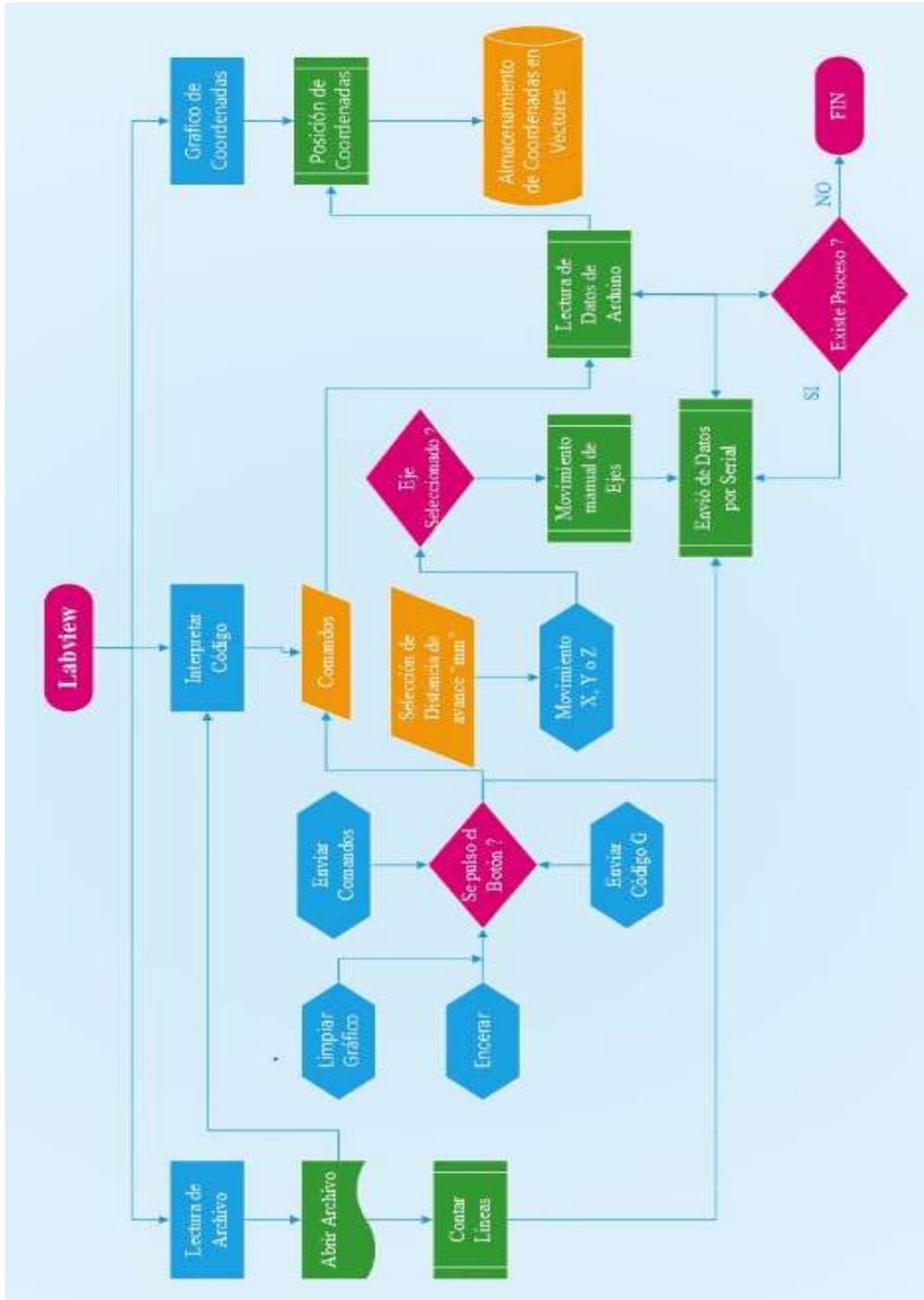
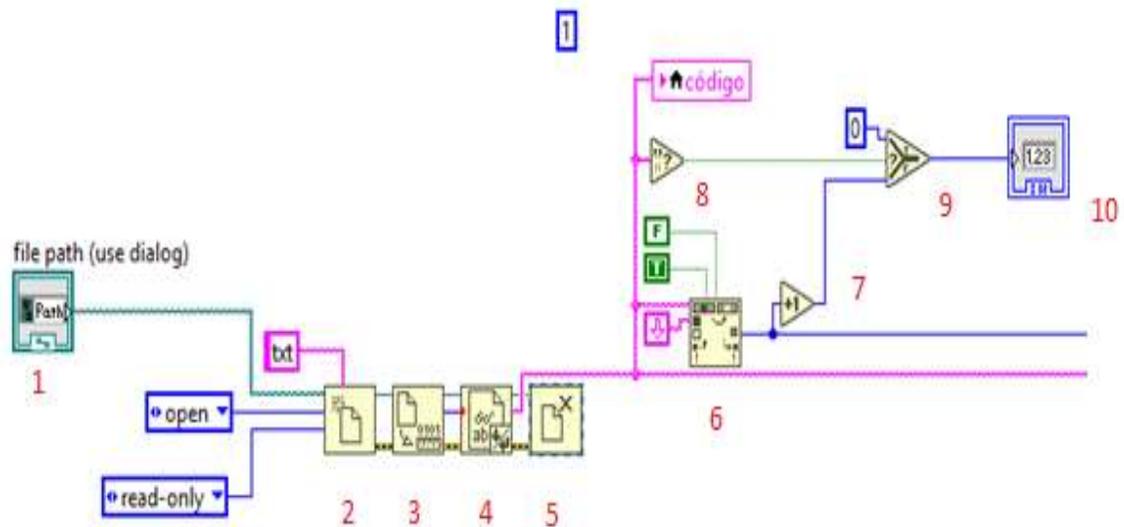


Figura 29. Diagrama de procesos de bloques de LabView.

### 3.3.5. DESARROLLO DEL DIAGRAMA DE BLOQUES EN EL PROGRAMA

#### LABVIEW

##### 3.3.5.1. BLOQUE DE LECTURA DEL ARCHIVO DE CÓDIGO G

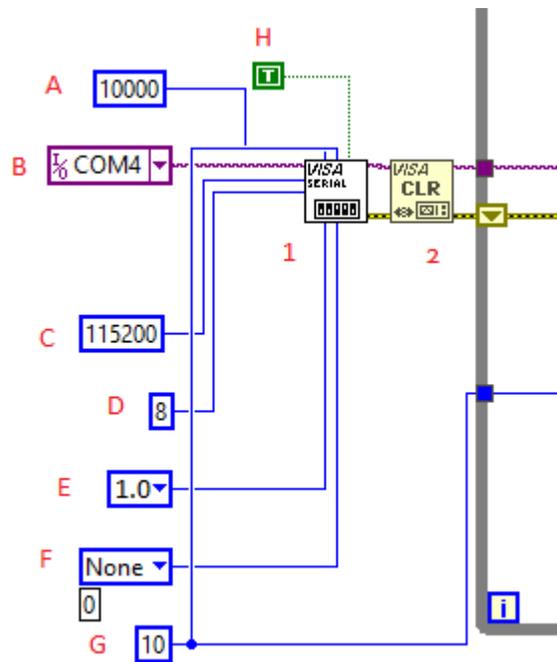


**Figura 30.** Indica proceso de lectura del archivo de código G.

- 1- File Path .- Permite abrir los archivos de un directorio alojado en un sistema operativo.
- 2- Open/Create/Replace File.- Autoriza abrir y leer el archivo que contiene la información del código G.
- 3- Get File Size.- Obtiene el tamaño del archivo e indica el tamaño del archivo en Bites.

- 4- Read From Text File.- Se encarga de leer el número de caracteres o líneas del archivo importado.
- 5- Close File.- Cierra el archivo una vez leído todas las líneas para que no lea en bucle infinito.
- 6- Search and Replace String.- Importa todo el contenido del archivo leído y busca usando en bloque "Line Feed Constant", el número de líneas del archivo, con un sumador que acumula el número de líneas encontradas.
- 7- Increment.- Acumula el número de líneas.
- 8- Empty String / Path.- Indica si existe o no un archivo que no esté vacío para poder visualizar el número de líneas obtenidas.
- 9- Select.- Permite indicar el número de líneas, en caso de que el archivo este vacío asignará el valor de cero; caso contrario asignará el valor de las líneas contadas.
- 10- Indicator.- Visualiza el valor obtenido del número líneas.

### 3.3.5.2. BLOQUE DE CONFIGURACIÓN DE PARÁMETROS DEL PUERTO SERIAL.



**Figura 31.** Configuración del puerto serial.

1. VISA Configure Serial Port .- Configura la comunicación serial a un dispositivo externo.
  - a.- Esta constante permite determinar el tiempo máximo de espera entre la conexión de la PC al dispositivo.
  - b.- Esta constante permite seleccionar el puerto COM encontrado al conectar el dispositivo con la PC.
  - c.- Establece la velocidad de comunicación entre el dispositivo y la PC.

d.- Asigna el tamaño de Bites usados en la comunicación.

e.- Permite asignar la cantidad de bites para detener el proceso de comunicación del último carácter enviado.

f.- Constante por defecto permanece deshabilitada ya que únicamente se utiliza cuando no se puede tener una comunicación constante con el dispositivo.

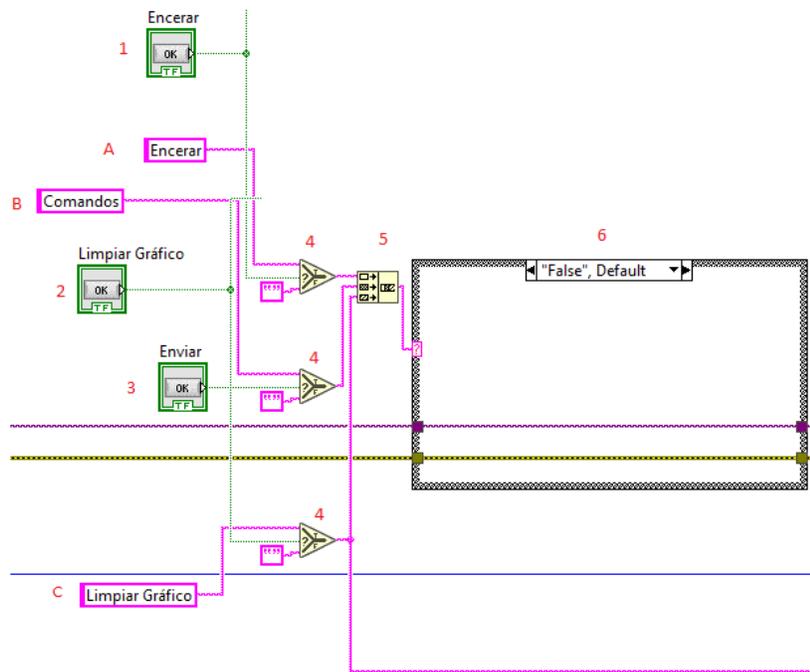
g.- Esta constante permite leer el número de caracteres de un máximo de 255, y detecta el número de caracteres leídos hasta que exista un salto de línea; estableciendo el valor de 10 o 0xA.

h.- La constante al estar activa indica que lea hasta que termine la lectura de caracteres de cada línea.

2. VISA Clear.- Este bloque permite encerrar el puerto COM para iniciar una comunicación serial sin datos.

### 3.3.5.3. BLOQUE DE FUNCIONES DE CONTROL

Indica el proceso de funcionamiento de encerrar, limpiar gráfico y enviar comandos.



**Figura 32.** Proceso de acciones.

1.- El botón de control “Encerrar” (1) permite enviar la cadena de caracteres “Encerrar” (A).

2.- El botón de control “Limpiar Gráfico” (2) permite enviar la cadena de caracteres “Limpiar Gráfico” (C).

3.- El botón de control “Enviar” (3) permite enviar la cadena de caracteres “Comandos” (B).

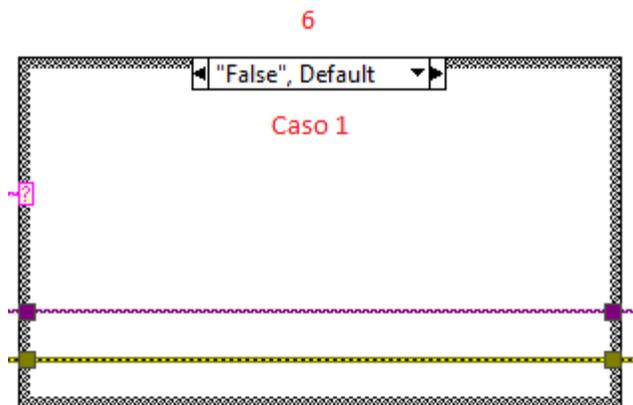
4.- Los tres bloques selectores son encargados de enviar la constante asignada a cada botón, siempre que se pulse el botón; caso contrario se envía una constante vacía para que el puerto serial no lea ningún dato.

5.- Concatenate Strings.- Une la cadena de caracteres de las constantes al pulsar cualquiera de los botones descritos anteriormente.

6.- Case Structure.- Contiene sub-diagramas que se activaran de acuerdo al botón pulsado con la constante determinada. A continuación, se interpretará cada uno de los casos:

Caso 1:

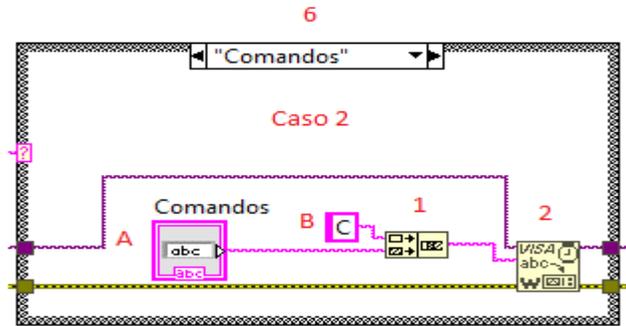
Determina el caso por defecto, donde no existe paso de datos al puerto serial.



**Figura 33.** Caso por defecto.

- Caso 2:

Determina el caso cuando se pulsa el botón Enviar donde existe paso de datos por puerto serial.



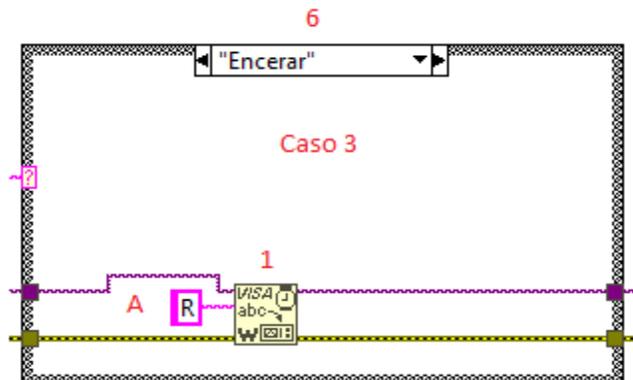
**Figura 34.** Lectura de comandos.

1.- Concatenate Strings.- Envía el dato de la constante “C” (B) o el dato ingresado en la caja de “Comandos” (A), para que Arduino interprete la lectura del código G ingresado.

2.- VISA Write.- Este bloque permite enviar datos usando la comunicación serial.

- Caso 3:

Determina el caso cuando se pulsa el Botón Encerar donde existe paso de datos por el puerto serial.

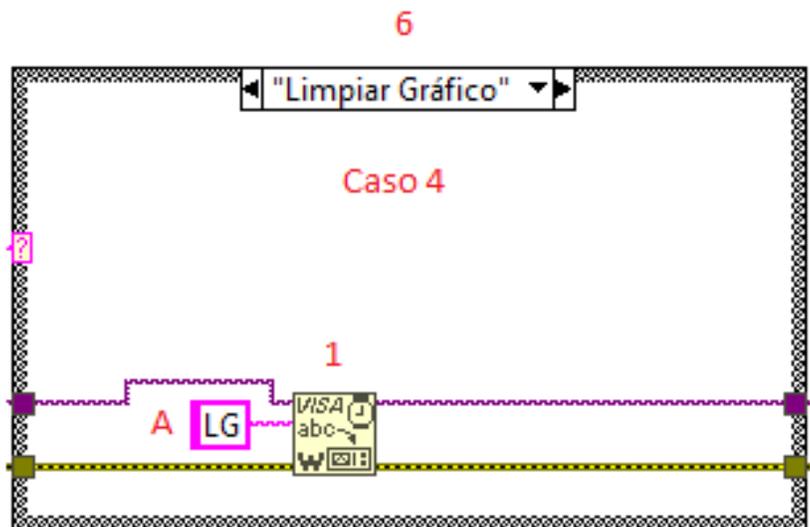


**Figura 35.** Acción encerar.

1.- VISA Write.- Envía datos a través de la comunicación serial, usando la constante “R” (A), para que Arduino interprete la letra R que encerrará los datos.

Caso 4:

Determina el caso cuando se pulsa el Botón Limpiar Gráfico donde existe paso de datos por el puerto serial.

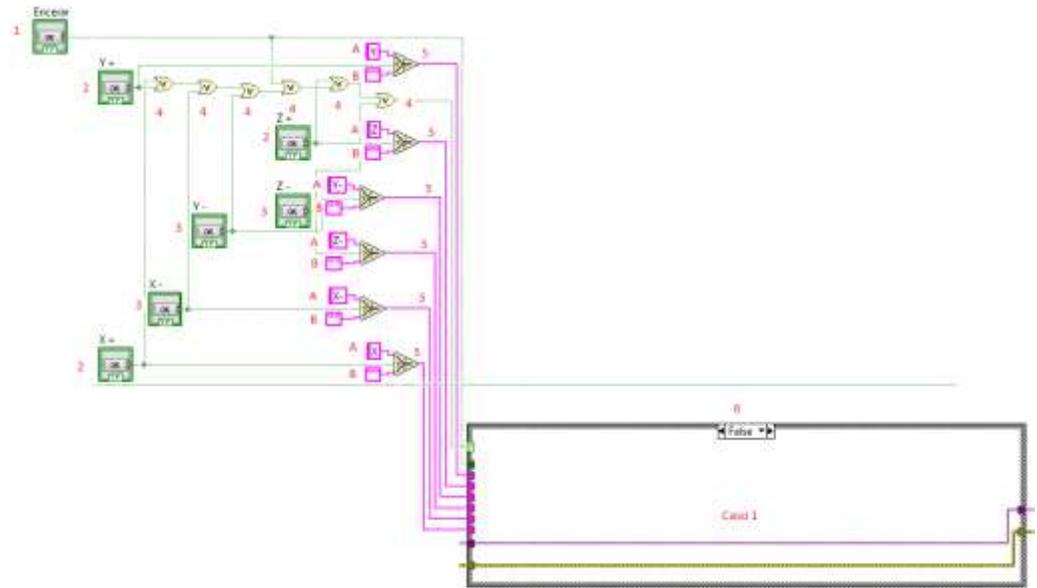


**Figura 36.** Acción limpiar gráfico.

1.- VISA Write.- Permite enviar datos a través de la comunicación serial, usando la constante “LG” (A), para que Arduino interprete las letras LG que encerrara el gráfico.

### 3.3.5.4. BLOQUE DE MOVIMIENTO DE EJES X, Y, Z

Analiza cómo mover los ejes X, Y, Z; por medio de botones que permite el avance del husillo escogiendo a través de una caja de opciones la distancia requerida.



**Figura 37.** Movimiento manual de los ejes.

- 1.- Encerar. - Encera los valores de los ejes X, Y, Z.
- 2.- Bloque X+, Y+, Z+. - Permiten mover el eje del husillo en avance positivo, de acuerdo a la distancia seleccionada en las opciones del bloque RadioButton.
- 3.- Bloque X-, Y-, Z-. - Estos bloques permiten mover el eje del husillo en avance negativo, de acuerdo a la distancia seleccionada en las opciones del bloque RadioButton.

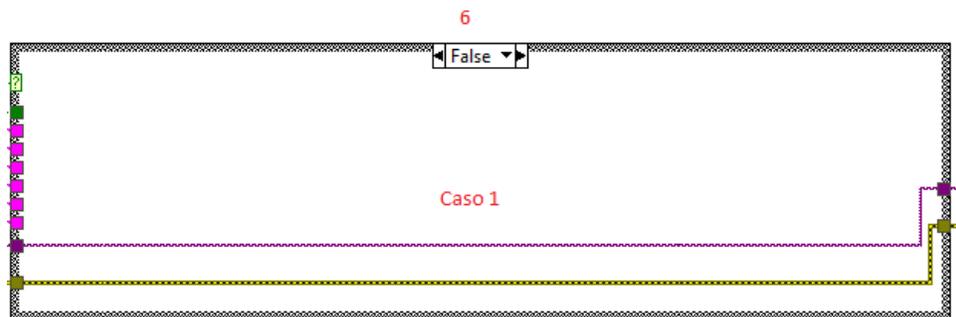
4.- Bloque OR.- Une todas las señales del bloque de botones de los ejes para poder obtener una señal del botón que sea pulsado, la cual permite habilitar “TRUE” del Case Structure.

5.- Bloque Selector.- Selecciona una de las constantes X+, X-, Y+, Y-, Z+, Z-; para indicar al programa con que eje se va trabajar en el avance del husillo.

6.- Case Structure.- En este bloque se tiene dos casos:

Caso 1:

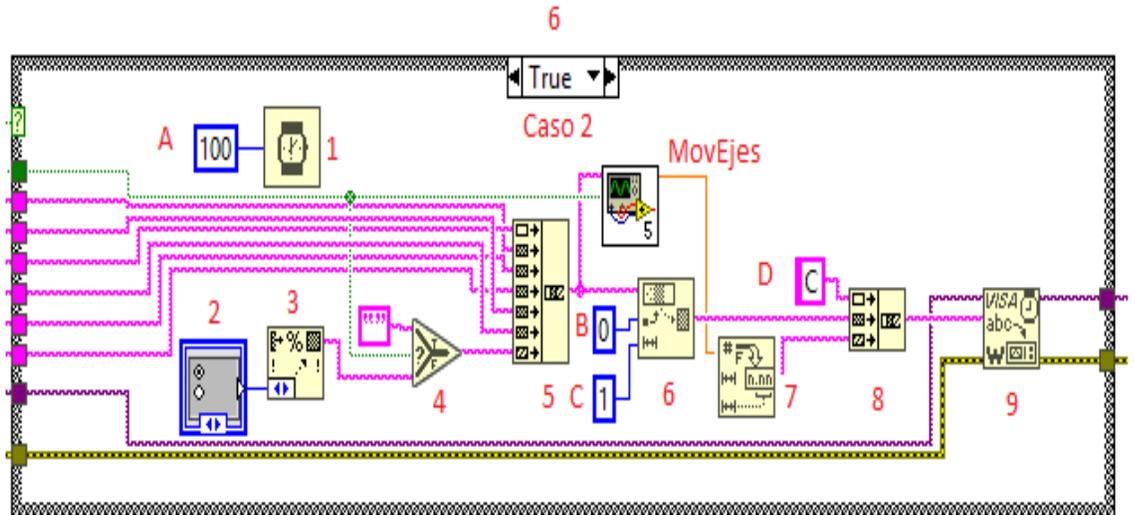
Determina el caso por defecto, donde no existe paso de datos al puerto serial.



**Figura 38.** Acción sin movimiento.

Caso 2:

Detalla el funcionamiento de las señales de cada botón pertenecientes a los ejes X, Y, Z; para interactuar con los datos del avance del husillo que se acumularan dentro de un Sub-Diagrama de bloques “MovEjes”.



**Figura 39.** Selección de la distancia de avance de los ejes.

- 1.- Wait.- Es un tiempo de espera, en este caso de 100 ms (A), que ayuda a detectar el pulso del botón una sola vez en dicho tiempo.
- 2.- RadioButton.- Selecciona una sola opción de distancia de los parámetros establecidos.
- 3.- Format Into String.- Lee los valores de distancia establecidos en cada botón a seleccionar en el bloque de RadioButton.
- 4.- Selector.- Habilita el paso de datos cuando se haya pulsado cualquier botón de los ejes.

5.- Concatenate String.- Une la cadena de caracteres de las constantes de los ejes X+, X-, Y+, Y-, Z+, Z-; para que el programa reconozca la letra del eje y el sentido, ya sea este positivo o negativo.

6.- String Subset.- Extrae la letra de cada eje sin el signo, para que se pueda aumentar la letra escogida con el valor de distancia acumulada en el sub-bloque. La constante "0" (B), indica la posición donde va a comenzar los datos, mientras que la constante "1" (C), indica la longitud de la cadena de caracteres.

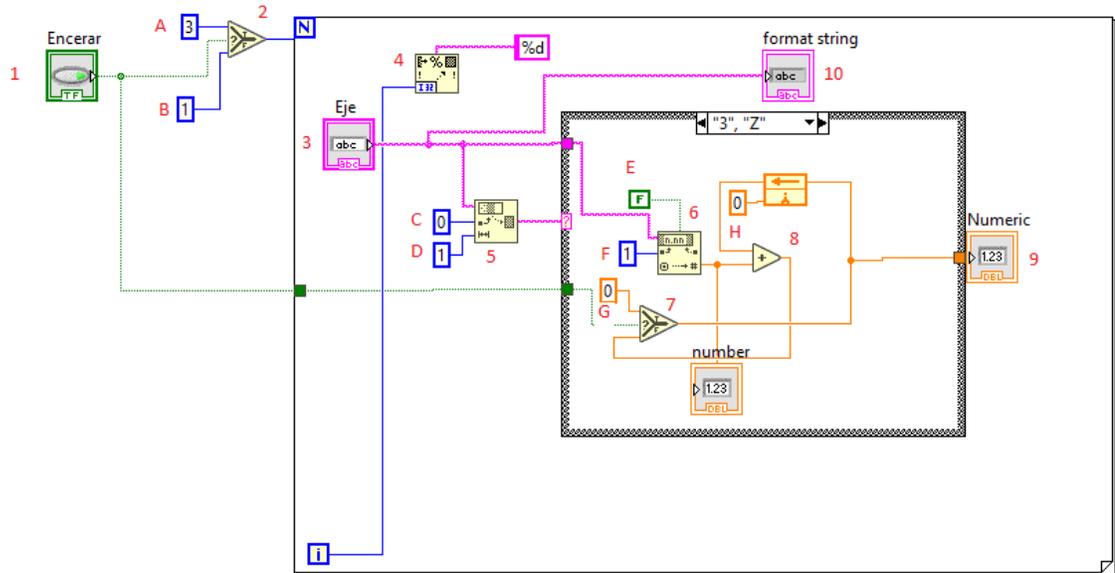
7.- Number To Fractional String.- Transforma el número acumulado en el sub-bloque de cada eje, convirtiendo en una cadena de caracteres con punto flotante que se enviara a Arduino.

8.- Concatenate String.- Envía la constante "C" (D), más la letra de cada eje más el valor de distancia acumulado, para que Arduino interprete que se está enviando los datos de distancia de cada eje.

9.- VISA Write.- Este bloque permite enviar datos a través de comunicación serial, usando la constante "C" (D), para que Arduino interprete las letras de cada husillo con sus respectivos valores almacenados.

### 3.3.5.5. SUB-DIAGRAMAS “MovEjes”

Muestra el Sub-Diagrama que controla los ejes de la máquina, guardando las posiciones de acuerdo con la distancia y eje seleccionado.



**Figura 40.** Almacenamiento de posición de los ejes.

- 1.- Bloque Control Encerar. - Resetea los valores acumulados de cada eje.
- 2.- Boton Select.- Este bloque se usa para activar el paso de la constante “1” (B), la cual permite encerar todos los ejes una sola vez, caso contrario el valor de la constante “3” (A), deja leer los datos en los 3 ejes.
- 3.- Bloque Indicator.- Obtiene los valores leídos de Arduino.

4.- Bloque Format Into String.- Indica el número de veces que se repite el ciclo “For Loop”.

5.- Bloque String Subset.- Extrae el caracter de la letra de cada eje. Al ser una sola letra la longitud máxima de la constante es “1” (D), y comenzará desde una posición “0” (C).

6.- Bloque Fract/Exp String To Number Function.- Este bloque permite extraer el valor de la distancia y eje seleccionado, convirtiendo en cadena de caracteres a número de punto decimal. Para solo extraer el valor del eje se toma la posición “1” (F), excluyendo la letra del eje seleccionado; mientras que la constante “F”(E), indica la lectura del punto decimal.

7.- Bloque Select.- Se usa para encerrar los valores de los ejes. Al pulsar el botón “Encerrar” (I) resetea los valores almacenados en el bloque “Feedback Node” (H).

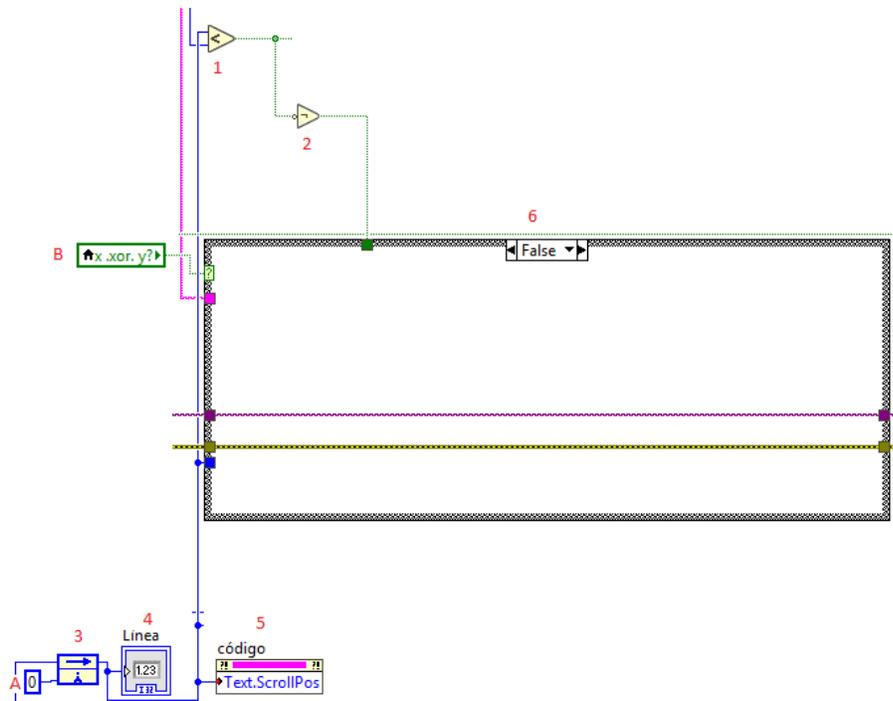
8.- Bloque Add.- Sumar o resta los valores de distancia de los ejes.

9.- Bloque Indicator.- Permite enviar los valores de distancia de los ejes desde el sub-sistema al diagrama de bloques principal.

10.- Bloque Indicator Format String.- Visualiza los valores de distancia respecto a cada eje seleccionado.

### 3.3.5.6. BLOQUE LECTURA DEL CÓDIGO G.

Permite interpretar las líneas de código G, leyendo hasta un máximo de 255 caracteres por línea.



**Figura 41.** Lectura de código G.

1.- Bloque Less.- Determina el número de líneas obtenidas del archivo es menor al número de líneas (4), procesadas por Arduino hasta finalizar el número total de líneas reconocidas del archivo.

2.- Bloque Not.- Permite indicar si el bloque de Less (1) cumple la condición, caso contrario finaliza el proceso de línea de código.

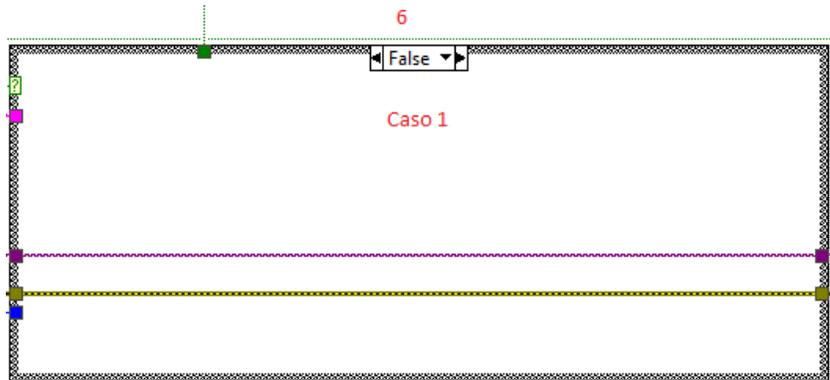
3.- Feedback Node. - Acumula el número de líneas de código G leídas por Arduino.

4.- Bloque Línea.- Vvisualiza el número de líneas de código G leídas.

5.- Bloque Scroll Position.- Desplaza la barra respecto al número de línea leída.

Caso 1:

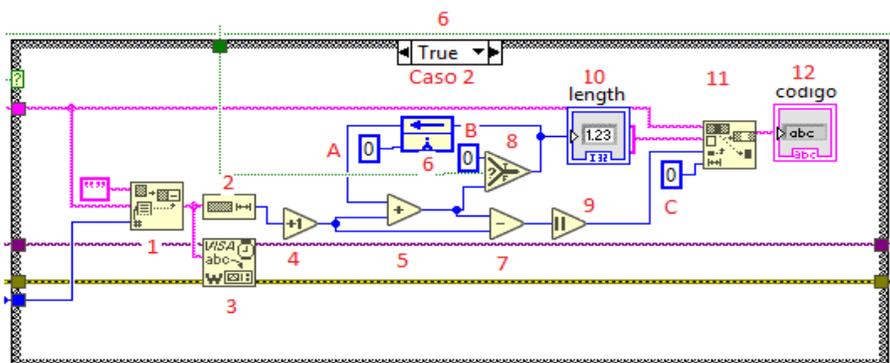
Se determina el caso por defecto, donde no existe paso de datos al puerto serial.



**Figura 42.** Lectura de código sin datos.

Caso 2:

Realiza el proceso de lectura del código G.



**Figura 43.** Lectura del código G.

1.- Pick Line.- Selecciona la línea correspondiente al último valor acumulado del proceso leído por Arduino, siendo esta la encargada de leer todos los caracteres pertenecientes a línea en proceso.

2.- String Length.- Este bloque permite saber el tamaño máximo de la cadena de caracteres en la línea.

3.- VISA Write.- Envía datos a través de la comunicación serial, usando los caracteres obtenidos por la asignación de cada línea del bloque Pick Line, para que Arduino interprete el texto obtenido del código G.

4.- Bloque Increment.- Toma el valor total del tamaño de la cadena de caracteres y suma uno para indicar el salto de línea, que es un caracter extra.

5.- Bloque Add.- Suma el total de la cadena de caracteres existentes en el archivo para poder indicar la última posición del ultimo caracter leído.

6.- Feedback Note.- Indica el ultimo caracter de línea que se acumulado en cada lectura de línea.

7.- Subtract.- Este bloque da la diferencia entre el tamaño total del número de caracteres leídos del archivo con el total del tamaño de la cadena de caracteres por línea, para poder remplazar con un caracter al comienzo de cada línea.

8.- Bloque Select.- Es para encerrar los datos acumulados.

9.- Absolute Value.- Es el resultado del valor obtenido del bloque Subtract en absoluto.

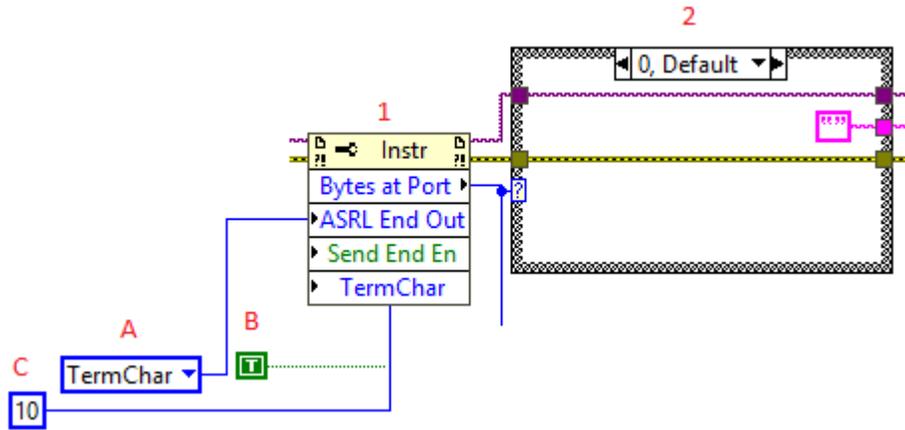
10.- Bloque Indicador “Length”.- Visualiza el tamaño del total de los caracteres leídos de todas las líneas procesadas.

11.- Replace Substring.- Reemplaza el símbolo “>” al inicio de cada línea, para indicar la posición de la línea que se está procesando.

12.- Bloque Indicador “código”.- Permite visualizar el código G importado aumentando el símbolo usado en el bloque Replace Substring.

### ***3.3.5.7. BLOQUE DE LECTURA DE DATOS SERIAL.***

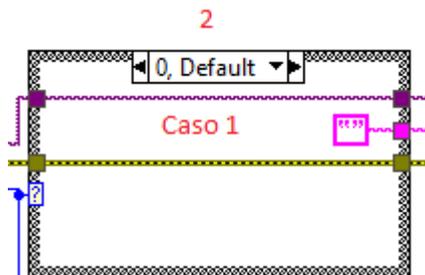
Encarga de recibir la información del Arduino para que LabView, interprete la información.



**Figura 44.** Lectura del puerto serial.

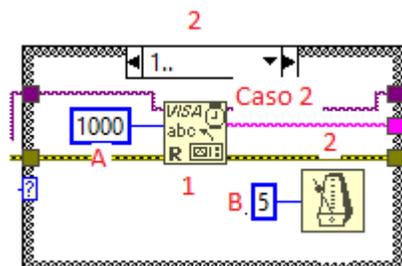
1.- Bytes at Serial Port.- Configura la lectura de datos serial entre Labview y cualquier dispositivo externo. En este caso se ha configurado la lectura de datos por línea “TRUE” (B), detectando hasta el último caracter “TermChar” (A) y el salto de línea “10” (C).

Caso 1: Determina el caso por defecto, donde no existe proceso de datos en puerto serial.



**Figura 45.** No detecta datos

Caso 2:



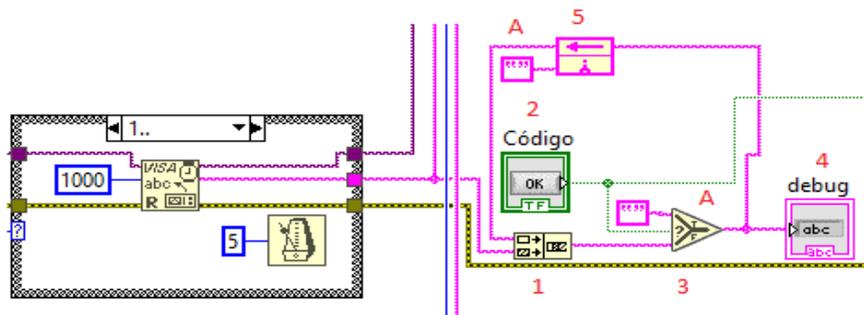
**Figura 46.** Lectura de datos

1.- VISA Read.- Permite la lectura de los datos a través del puerto serial, donde se lee hasta un máximo de “1000” (A) Bytes.

2.- Wait Until Next ms Multiple Function.- Este bloque permite sincronizar el proceso de lectura de datos cada “5” (B) ms.

### 3.3.5.8. BLOQUE DE DEPURACIÓN DE DATOS RECIBIDOS DE ARDUINO.

Visualiza la lectura de todos los datos recibidos, para poder analizar los procesos que se están ejecutando.



**Figura 47.** Verificación de procesos.

1.- Concatenate Strings.- Enlaza los datos leídos por Arduino y con el bloque “Feedback node” (5), ir acumulando la información que será desplegada en el indicador “debug” (4).

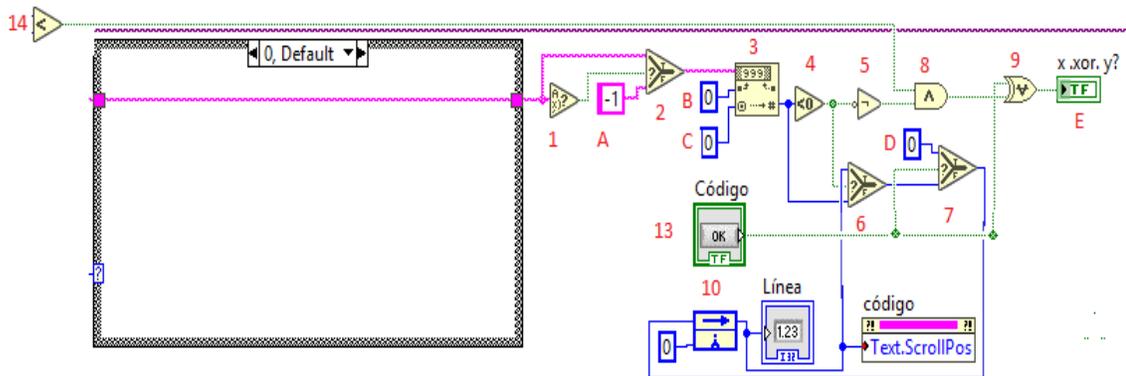
2.- Bloque Control “Código”.- Encera los datos almacenados en el bloque “Concatenate Strings”(1), una vez que se ha pulsado.

3.- Bloque Select.- Este bloque permite pasar los datos al bloque “Feedback node” (5) y al bloque “Concatenate Strings”(1), siempre y cuando no se pulse el .- Bloque Control “Código”(2); caso contrario enviara la constante “Empty String Constant” (A), que borraría datos almacenados.

4.- Bloque indicador “debug”.- Visualiza toda la información leída a través del puerto serial.

### 3.3.5.9. BLOQUE CONTADOR DE LINEAS DE CODIGO G

Permite contar el número de líneas que se va procesando en Arduino, cuando se inicia la lectura del código G.



**Figura 48.** Control de número de líneas.

1.- Bloque Printable.- Permite retornar el valor del TRUE si existe algún caracter, caso contrario retorna FALSE. Con este bloque se puede activar o desactivar el paso de datos al bloque “Select” (2).

2.- Bloque Select.- Este bloque permite el paso de datos cuando existe algún caracter, caso contrario envía el valor de “-1” (A).

3.- Bloque Decimal String to Number.- Convierte la cadena de caracteres leída a través de Arduino o con la constante “-1” (A) en números decimales enteros. La constante “0” (B) y (C), configuran el bloque sin Offset y entero de 32-Bit con signo.

4.- Bloque Less Than 0.- Compara un valor, si este es menor que cero retorna un valor TRUE que activa el bloque “Select” (6).

5.- Bloque Not.- Invierte la señal obtenida por el bloque “Less Than 0” (4), para poder comparar dicha señal con el “Bloque And”(8).

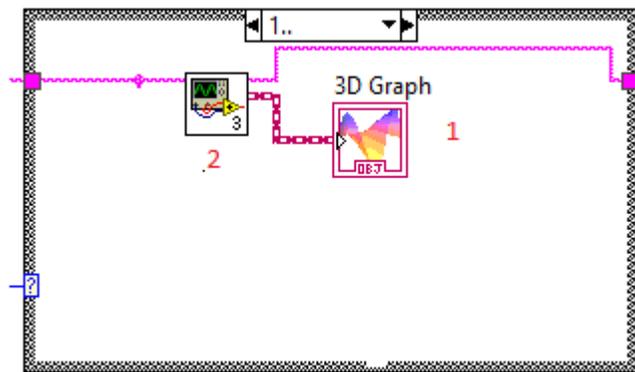
6.- Bloque Select.- Obtiene el valor de número de línea, si el valor es menor de cero retorna un valor de “-1” (A), caso contrario retorna el valor del número de línea procesada en el bloque “Feedback Node” (10).

7.- Bloque Select.- Permite encerrar el contador de líneas cuando se ha pulsado el “Botón control Código”(13).

8.- Bloque And.- Habilita el contador de líneas si el valor total de líneas sumadas es menor “bloque Less” (14), que el valor total de líneas del archivo del código G.

9.- Bloque Exclusive Or.- Este bloque permite habilitar la lectura del código G cuando sea pulsado el “Botón control Código”(13) y cuando el número total de líneas procesadas sea menor “bloque Less” (14) al total de líneas del archivo.

### 3.3.5.10. BLOQUE GRÁFICO EN COORDENADAS X, Y, Z.

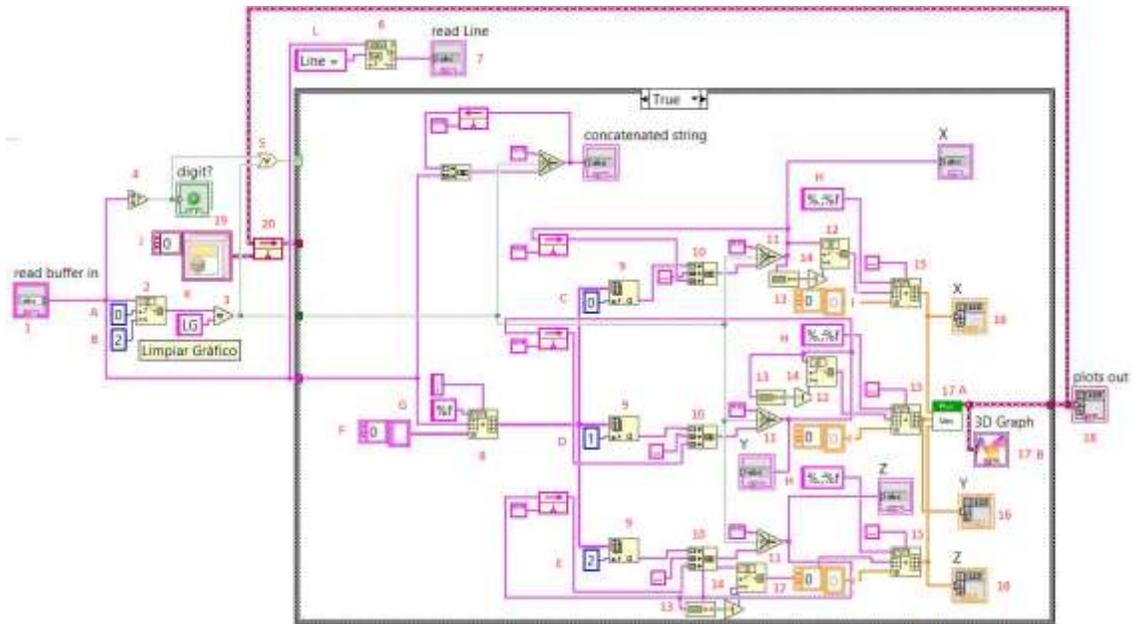


**Figura 49.** Bloque de lectura y gráfico.

1.- Bloque 3D Graph.- Visualiza los datos obtenidos de las coordenadas X, Y, Z; procesadas por Arduino e interpola dichas coordenadas formando el gráfico generado por código G.

#### 2.- SUB-SISTEMA:

Lee e interpola los datos enviados de Arduino a Labview, guardando las coordenadas en un vector que corresponde a cada eje.



**Figura 50.** Almacenamiento de coordenadas.

- 1.- Bloque Indicador Read Buffer in.- Lee los datos procesados por Arduino.
- 2.-Bloque String Subset.-Subtrae caracteres del bloque “Bloque Indicador Read Buffer in” (1), para poder comparar hasta dos caracteres iniciando con la constante “0” (A) hasta un tamaño máximo “2” (B).
- 3.- Bloque Equal.- Compara los dos primeros caracteres extraídos con la constante “LG” (K).
- 4.- Bloque Decimal Digit.- Detecta si la cadena de caracteres es un número.
- 5.- Bloque OR.- Activa el Case Structure al pulsar el botón de limpiar gráfico, para poder encerrar los datos de las coordenadas cuando la constante “LG” (K) es igual a la constante LG

enviada por Arduino. Cuando el “Bloque Decimal Digit” (4), detecta un número activa de igual manera el Case Structure.

6.- Bloque Match Pattern.- Busca la cadena de caracteres de la constante “Line =” (L), y comparar con la cadena de caracteres obtenida de Arduino. En el caso de encontrar dicha constante extraerá los caracteres posteriores, indicando en número de línea que se encuentra el proceso.

7.- Bloque Indicador Read Line.- Este bloque permite indicar el resultado del bloque “Bloque Match Pattern” (6).

8.- Bloque Spreadsheet String To Array Function.- Separa una cadena de caracteres a partir de un caracter en común, por ejemplo “ 10;12;8” , separa por el caracter común “ ; “, obteniendo el vector “10 12 8”.

9.- Bloque Index Array. – Guarda por posiciones el vector obtenido del bloque “Spreadsheet String To Array Function” (8). El valor de la posición 0 corresponde a X, de la posición 1 corresponde a Y, y de la posición 2 corresponde a Z.

10.- Bloque Concatenate Strings.- Este bloque permite guardar las posiciones de las coordenadas, separando cada coordenada con un espacio.

11.-Bloque Select.- Este bloque permite encerrar las posiciones guardadas cuando el bloque “Equal” (3), envía una señal TRUE ya que se accionado el botón “Limpiar Grafico” (LG).

12.- Bloque String Subset.- Permite obtener el valor de la coordenada suprimiendo el espacio asignado por el bloque “Concatenate Strings” (10).

13.- Bloque String Length.- Obtiene el tamaño de caracteres guardadas en el vector, por lo que restando “-1” se obtendrá el tamaño de los caracteres sin el espacio asignado del bloque “Concatenate Strings” (10).

14.- Bloque Decrement.- Resta el valor de uno al bloque “String Length” (13).

15.- Bloque Spreadsheet String To Array Function.- Separa los valores de las coordenadas de cada eje y convertirlas en Array con punto decimal.

16.- Bloque Indicator.- Visualiza el valor de cada vector correspondiente a las coordenadas X, Y, Z.

17A.- Bloque 3D Line Plot Setup VI.- Permite interpolar los valores de las coordenadas X, Y, Z, para enviar dichos datos al bloque “3D Graph” (17B).

17B.- Bloque 3D Graph.- Grafica los datos obtenidos en el bloque “3D Line Plot Setup VI” (17 A).

18.- Bloque indicator.- Este bloque permite enlazar los valores del bloque “3D Graph” (17B), para poder visualizar el bloque “3D Graph” (17B), en el diagrama de bloques principal.

19.- Bloque Surface In.- Crea una clase de objetos que almacena la información de las coordenadas del código G.

20.- Bloque Feedback Node.- Este bloque permite almacenar la información de las coordenadas anteriores con las actuales.

### 3.3.6. DIAGRAMA DE FLUJO DEMOSTRANDO EL PROCEDIMIENTO DE CÓDIGO DE ARDUINO

Se observa los pasos realizados para la programación implementada en Arduino.

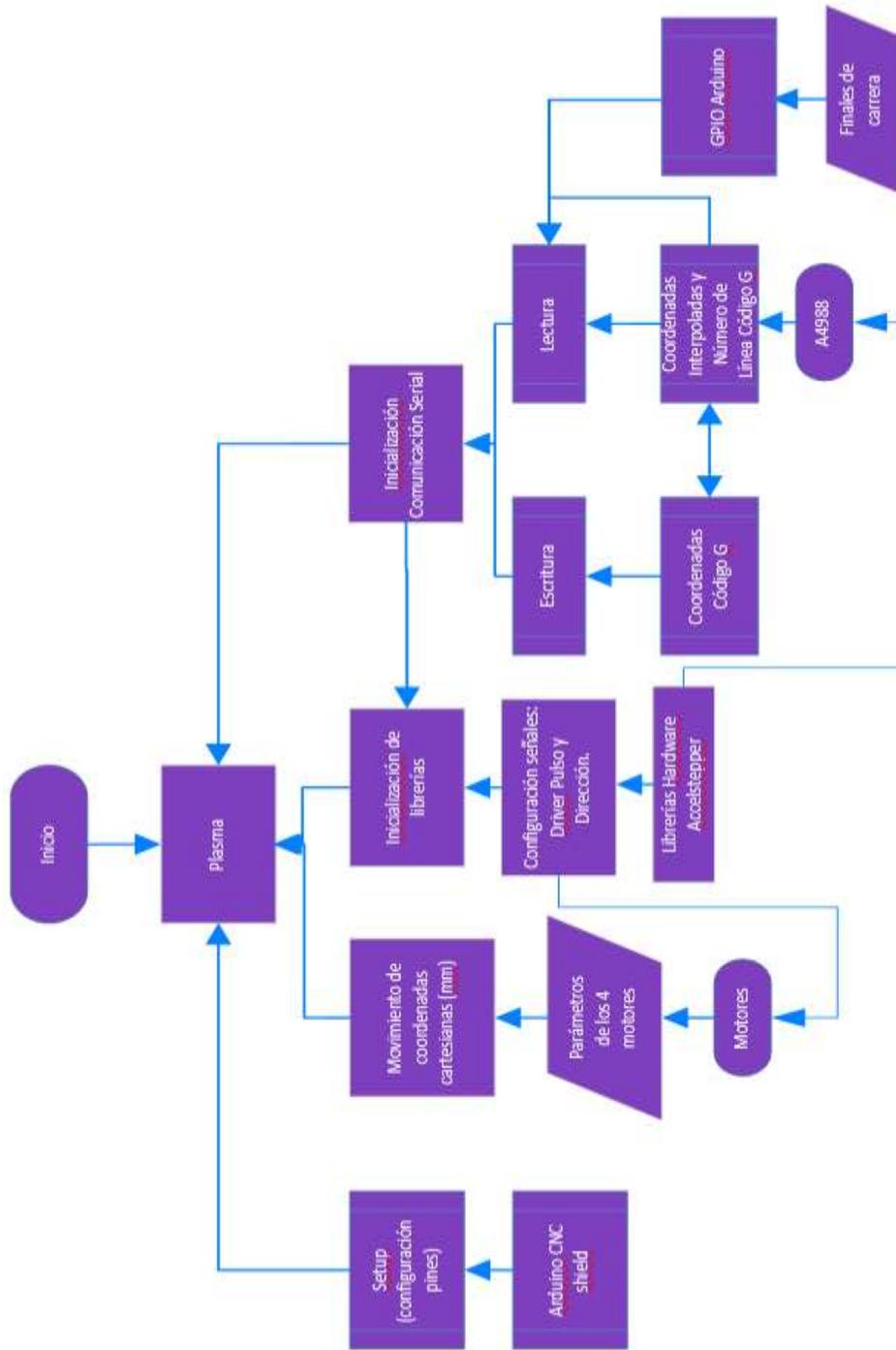


Figura 51. Diagrama de flujo de código de Arduino.

### 3.3.7. DESARROLLO CODIGO DE ARDUINO

En la figura 51 se analiza el proceso de interacción del código que controla la máquina CNC Plasma. Se procede a configurar los pines de la plataforma con los del shield CNC de Arduino, una vez determinados cuales funcionan como entradas y salidas se indica a la librería Accelstepper las señales que funcionaran como pulso y dirección de cada motor.

Para poder interactuar con todos los motores se usa la librería “MultiStepper” que será la que controle el movimiento simultaneo de todos los controladores de los motores interpolando las coordenadas de los ejes X, Y, Z. Esta información de movimiento se obtiene por comunicación serial y está programada para usar con código G para movimiento lineal y curvilíneo, controlando la velocidad de avance según la distancia asignada evitando movimientos abruptos en curvas cerradas por exceso de velocidad. Por lectura serial se controla también el conteo de líneas para determinar cuándo ha terminado de procesar todos los datos de código G.

#### **Importación de librerías.**

```
#include <AccelStepper.h>
#include <MultiStepper.h>
#include <math.h>
#include <stdio.h>
```

Las librerías usadas en la plataforma de Arduino son:

Accelstepper.- Permite interactuar con diferentes configuraciones que controlan motores dc sin escobillas de 3 pines y motores p a p unipolares, bipolares con un driver o control de bobinas

con configuración de 2, 3 o 4 conexiones. La usada en esta aplicación es la de pulso y dirección que pertenece al control de motor a pasos con un driver.

MultiStepper.- Permite integrar el control de hasta 10 motores a pasos para poder moverlos de forma simultánea. Se ha modificado a la librería para poder tener las funciones que está presentes en la librería “AccelStepper”, ya que al utilizar la configuración de “MultiStepper” se inhabilita la mayoría de funciones.

Math.- Esta librería integra todas las funciones matemáticas necesarias para poder realizar los cálculos de los movimiento lineales y curvilíneos utilizados al generar las rutas vectoriales de una imagen interpretadas por código g.

Stdio.- Esta librería es utilizada en la manipulación de caracteres o cadena de caracteres usados para poder determinar las instrucciones de código G con el valor designado en cada coordenada de los ejes X, Y y Z.

### **Definición de pines usados en los motores**

```
//define pines usados en los motores
#define MX_STEP 2
#define MX_DIR 5
#define MY_STEP 3
#define MY_DIR 6
#define MZ_STEP 4
#define MZ_DIR 7
#define M_Enable 8
```

Para configurar los pines con el CNC shield de Arduino se usa los pines 2, 3, 4 para la señal de pulso; los pines 5, 6, 7 para la señal de dirección y el pin 8 para activar todos los drivers de los motores con la señal enable. Para el uso del eje X se utiliza dos motores que se mueven de forma sincronizada con las señales establecidas de los pines 2 y 5.

### Definición de parámetros del motor

```
//define parámetros de motor
#define Pasos_por_rev 200 // Pasos por revolución
#define Min_velocidad 1 // 1 Paso/Segundo
#define Max_velocidad 20000 // 20000 Pasos/Segundo (Máxima frecuencia) 20khz
```

Se define los parámetros que usa el motor a pasos que es de 200 pasos por revolución, la velocidad mínima será de 1 paso por segundo y la máxima de 20000 pasos por segundo que es el límite de frecuencia soportado por Arduino con la librería “AccelStepper”.

```
//define parámetros de transmisión de movimiento
#define Paso_Husillo 1.0 // Valor paso husillo en mm/rev.
#define MicropasosMX 8 // Valor Micropasos en Driver Motor X
#define MicropasosMY 8 // Valor Micropasos en Driver Motor Y
#define MicropasosMZ 8 // Valor Micropasos en Driver Motor Z
#define MM_por_segmento 1.0 // Subdivision en segmentos de 1 mm
```

Los parámetros de transmisión de movimiento son calculados de acuerdo al tipo de mecanismo usado, siendo en este caso husillos con paso de 1mm por revolución. Los valores de los micro-pasos serán de 8 en todos los motores permitiendo un balance de torque y velocidad de

avance; con respecto a la velocidad de avance los motores se limitan a la frecuencia máxima por lo que a más micro-pasos menor velocidad de avance de los motores.

La variable de “MM\_por\_segmento” permite subdividir el trayecto de los ejes X, Y, Z en trayectos de 1mm, ya que si no se subdividiera no se podría interrumpir con otras funciones el movimiento de los motores hasta que lleguen a la distancia dada por los comandos o código G. Por lo tanto, cada 1 mm o menor que 1mm que avanza cada motor de forma independiente o interpolada retorna el valor de la última posición y a la vez permite verificar si se activaron nuevas instrucciones como parada de emergencia o finales de carrera.

### **Definición de pines usados para botón de emergencia y finales de carrera**

```
//define pines usados para los finales de carrera y boton de emergencia
#define estop      10
#define limiteX   11
#define limiteY   12
#define limiteZ   13
```

Se define los pines usados para el uso del botón de emergencia y finales de carrera usados en la máquina. Se ha usado tres entradas para detectar los límites máximos de los finales de carrera en los ejes X, Y, Z, pero se puede usar compuertas OR que permitan detectar también los límites mínimos de la máquina, esto indicaría a los pines de entrada cuando ha tocado el carro el final de carrera del límite máximo o mínimo de cada eje. Para el uso del botón de emergencia se usa un pulsador normalmente cerrado y cuando se pulse se abra dando la instrucción de desactivar los controladores de los motores.

## Asignación de librería para el uso con controladores de motores

```
5 AccelStepper MotorX(AccelStepper::DRIVER, MX_STEP, MX_DIR); //pin step 2 pin dir
6 AccelStepper MotorY(AccelStepper::DRIVER, MY_STEP, MY_DIR); //pin step 3 pin dir
7 AccelStepper MotorZ(AccelStepper::DRIVER, MZ_STEP, MZ_DIR); //pin step 4 pin dir
MultiStepper Motores;
```

Las variables usadas con la librería “AccelStepper” controlan cada motor y de esta forma se indica que se los controla con drivers con los pines asignados al paso y la dirección de cada motor. Se crea una variable llamada “Motores” para poder agregar a la librería “MultiStepper” todos los motores que van a funcionar de forma simultánea interpolando las coordenadas; este proceso se lo realiza en la función “setup()” de Arduino.

## Variables globales

```
String Serial_data; //Datos leídos por el puerto serial.
char data[64]; //Información almacenada de Serial_data con rango máximo de 64
caracteres
char Comandos[64]; //Información almacenada de data (copia de data).
char delimiters[] = "CRGXYZIJKDF("; //Letras a buscar de los datos leídos por el
puerto serial.
char* val; //valores obtenidos de los comandos
float xpasos_por_mm; //variable global usada para calcular la distancia en mm
(escala)
float ypasos_por_mm; //variable global usada para calcular la distancia en mm
(escala)
```

```

float Zpasos_por_mm; //variable global usada para calcular la distancia en mm
(escala)
float max_speed_x; // Maxima velocidad X (mm/s)
float max_speed_y; // Maxima velocidad Y (mm/s)
float max_speed_z; // Maxima velocidad Z (mm/s)
float feed_x; // Feedrate X (paso/s)
float feed_y; // Feedrate Y (paso/s)
float feed_z; // Feedrate Z (paso/s)
float Xposition, Yposition, Zposition, Iposition, Jposition, Kposition,
Vel_avance; //Variables para almacenar posiciones.
int Direccion;
float posX, posY, posz; //Almacena posición
int dibujo; //Verifica si se va dibujar una línea o arco (sentido)
long linea_texto;

```

A continuación, se define el tipo de variables globales usadas las cuales almacenaran los datos procesados en cada función de cálculo, posición y velocidad.

Serial\_data.- Permite almacenar datos tipo String obtenidos de Labview.

data[64].- Permite almacenar cadena de caracteres con una longitud máxima de 64 caracteres leídos a través del puerto serial.

Comandos[64].- Todos los datos almacenados en data[64] son copiados a la variable Comandos[64] para que la información de separación de caracteres realizada por la función “strtok” no modifique los valores originales de data[64].

Delimiters[].- Se encarga de separar la cadena de caracteres con la función “strtok”, si algún carácter coincide. Ejemplo “G0 X10 Y10”, los delimitantes son “G, X, Y” y los separa por “0”, “10”, “10”; estos son almacenados en un puntero llamado “val”.

Val.- Es una variable tipo char puntero que almacena los datos obtenidos de los comandos y código G.

Xpasos\_por\_mm.- Almacena el valor flotante calculado de la escala X.

Ypasos\_por\_mm.- Acumula el valor flotante calculado de la escala Y.

Zpasos\_por\_mm.- Guarda el valor flotante calculado de la escala Z.

max\_speed\_x.- Obtiene los valores calculados de la máxima velocidad con respecto al valor de la escala en X en mm/min.

max\_speed\_y.- Consigue los valores calculados de la máxima velocidad con respecto al valor de la escala en Y en mm/min.

max\_speed\_z.- Alcanza los valores calculados de la máxima velocidad con respecto al valor de la escala en Z en mm/min.

feed\_x.- Adquiere los valores de la velocidad vectorial en X en pasos por segundo que es la frecuencia que determina la máxima velocidad de avance.

feed\_y.- Obtiene los valores de la velocidad vectorial en Y en pasos por segundo que es la frecuencia que determina la máxima velocidad de avance.

feed\_z.- Consigue los valores de la velocidad vectorial en Z en pasos por segundo que es la frecuencia que determina la máxima velocidad de avance.

Xposition.- Almacena los valores obtenidos de la posición en X por comandos o código G.

Yposition.- Acumula los valores obtenidos de la posición en Y por comandos o código G.

Zposition.- Guarda los valores obtenidos de la posición en Z por comandos o código G.

Iposition.- Almacena los valores obtenidos de la posición relativa en I, centro de arco en X por comandos o código G.

Jposition.- Recibe los valores obtenidos de la posición relativa en J, centro de arco en Y por comandos o código G.

Kposition.- Almacena los valores obtenidos de la posición relativa en K, centro de arco en Z por comandos o código G.

Vel\_avance.- Obtiene la velocidad de avance en mm/min ingresada por comandos o código

G.

Direccion.- Recibe los valores de -1 y 1 según el sentido horario o anti-horario que requiere el arco.

pos\_x.- Almacena la última posición calculada en X.

pos\_y.- Acumula la última posición calculada en Y.

pos\_z.- Guarda la última posición calculada en Z.

línea\_texto.- Almacena el número de líneas procesados del código G.

### **Configuración setup()**

```
void setup() {  
  Serial.begin(115200);  
  pinMode(M_Enable, OUTPUT);  
  digitalWrite(M_Enable, 0);  
  pinMode(MX_STEP, OUTPUT);  
  pinMode(MX_DIR, OUTPUT);  
  pinMode(MY_STEP, OUTPUT);  
  pinMode(MY_DIR, OUTPUT);  
  pinMode(MZ_STEP, OUTPUT);  
  pinMode(MZ_DIR, OUTPUT);  
  pinMode(estop, INPUT);  
  pinMode(limiteX, INPUT);  
  pinMode(limiteY, INPUT);  
}
```

```

pinMode(limitez,INPUT);
vel_avance = 500;
distancia_mm_degree();
calcMaxSpeed();
MotorX.setMinPulsewidth( 1 );
MotorY.setMinPulsewidth( 1 );
MotorZ.setMinPulsewidth( 1 );
MotorX.setMaxSpeed( 5000 );
MotorY.setMaxSpeed( 5000 );
MotorZ.setMaxSpeed( 5000 );
//MotorX.setPinsInverted(1,0,0);
MotorY.setPinsInverted(1,0,0);
Motores.addStepper( MotorX );
Motores.addStepper( MotorY );
Motores.addStepper( MotorZ );
Serial.println( "Listo");
}

```

La función setup () permite configurar todas las funciones de hardware que se van usar; como la lectura y escritura serial a 115200 baudios, pines de entrada y salida para los pulsadores y motores, parámetros de inicialización de las variables y por último añadir las variables de los motores X, Y, Z de la librería “AccelStepper” a la “MultiStepper”.

Al terminar de configurar todo lo que se encuentra en la función de setup(), se envía un mensaje de “Listo” por comunicación serial, el cual indica que se puede comenzar a operar la máquina desde Labview.

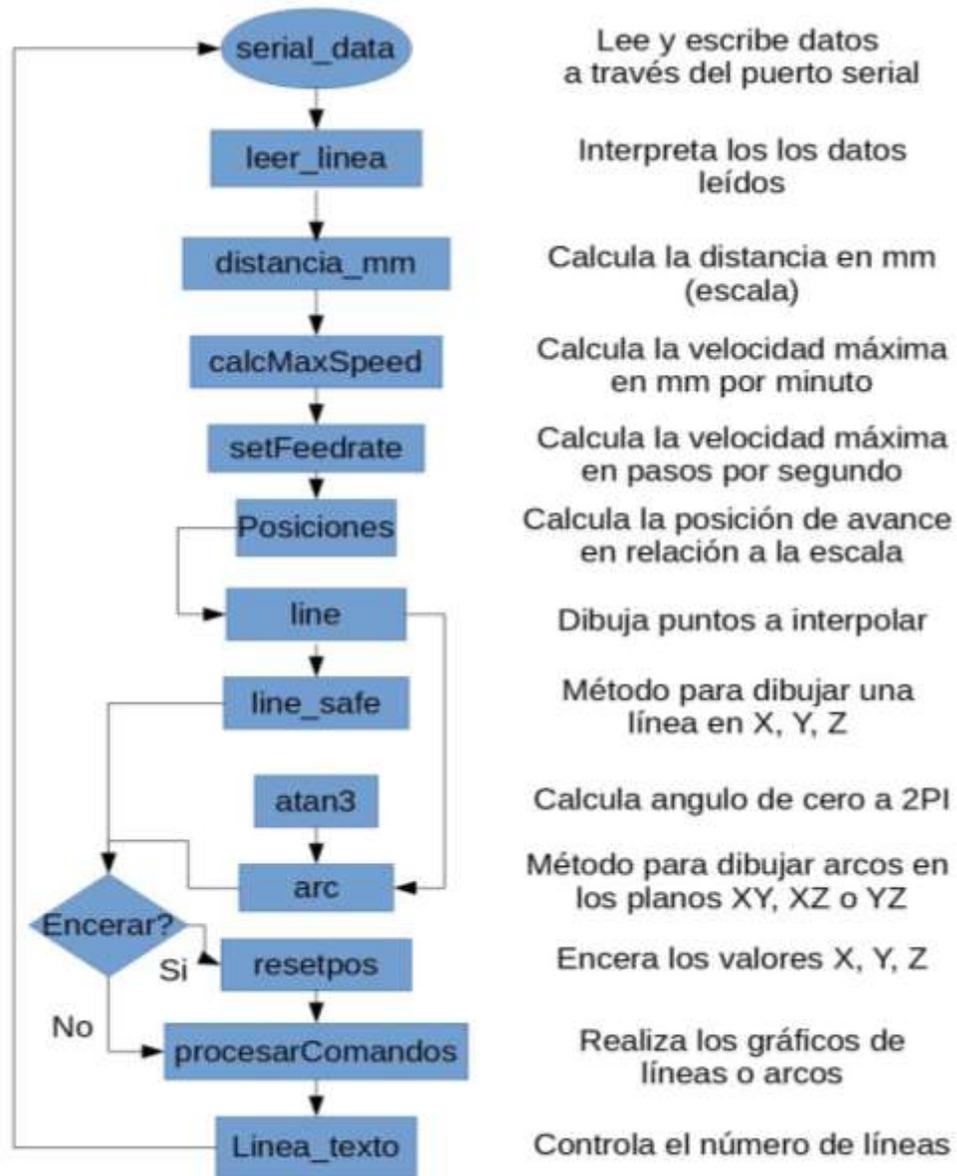
## Lectura continua del código de Arduino en función loop()

```
void loop() {  
  if (Serial.available() > 0) {  
    Serial_data = Serial.readStringUntil('\n');  
    leer_linea(Serial_data);  
    Serial.println(Serial_data);  
    Serial.print("Line = ");  
    Serial.println(linea_texto);  
    linea_texto++;  
    //pause(estop);  
  }  
}
```

En el ciclo loop se verifica si existen datos enviados a través de la comunicación serial; si hay se lee la cadena de caracteres de comandos, código G o botones de la interfaz gráfica de Labview y se procede a verificar los datos. Si el dato es obtenido por comando o botones retorna un número de línea de -1 ya que no se está procesando el archivo de código G, con esto se indica a Labview que no ejecute el archivo de texto importado; caso contrario si se activa el botón de enviar código G comienza la suma de líneas y comunicación entre la línea obtenida del texto de Labview y la acumulada en Arduino.

### Funciones de Arduino

Se indica el proceso de las funciones que Arduino realiza para dar las instrucciones de movimiento a la CNC Plasma.



**Figura 52.** Funciones de Arduino

### Procesos de las funciones

Se detalla el proceso de cada función que se encarga de calcular y obtener los valores para que el controlador de cada motor realice los movimientos correspondientes.

## Función serial\_data

Se explica el proceso de lectura de datos a través de la comunicación serial.

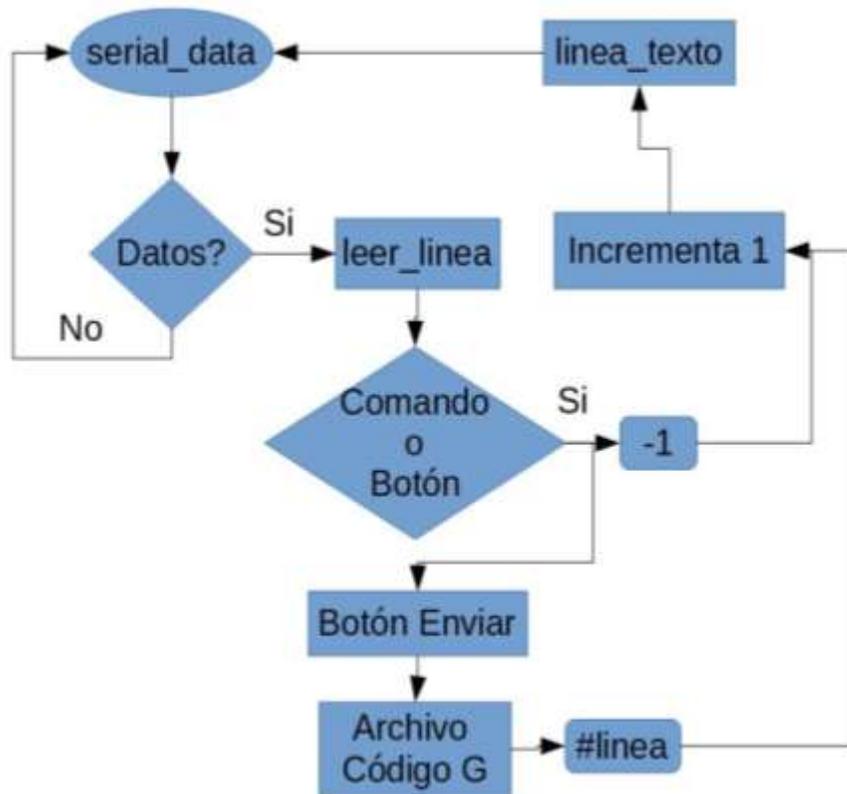


Figura 53. Función serial\_data

En la comunicación serial general se indica al programa de LabVIEW cuando procesa acciones que no sean la lectura del archivo de código G que reciba el valor de -1, así al incrementar dicho valor con 1 esta será igual a cero por lo que no se procesará el código G hasta que el valor sea mayor a cero. El valor de línea texto será mayor a cero cuando se pulse el botón “Enviar” que se encarga de procesar todo el archivo de texto importado en LabView.

## **Función leer\_línea**

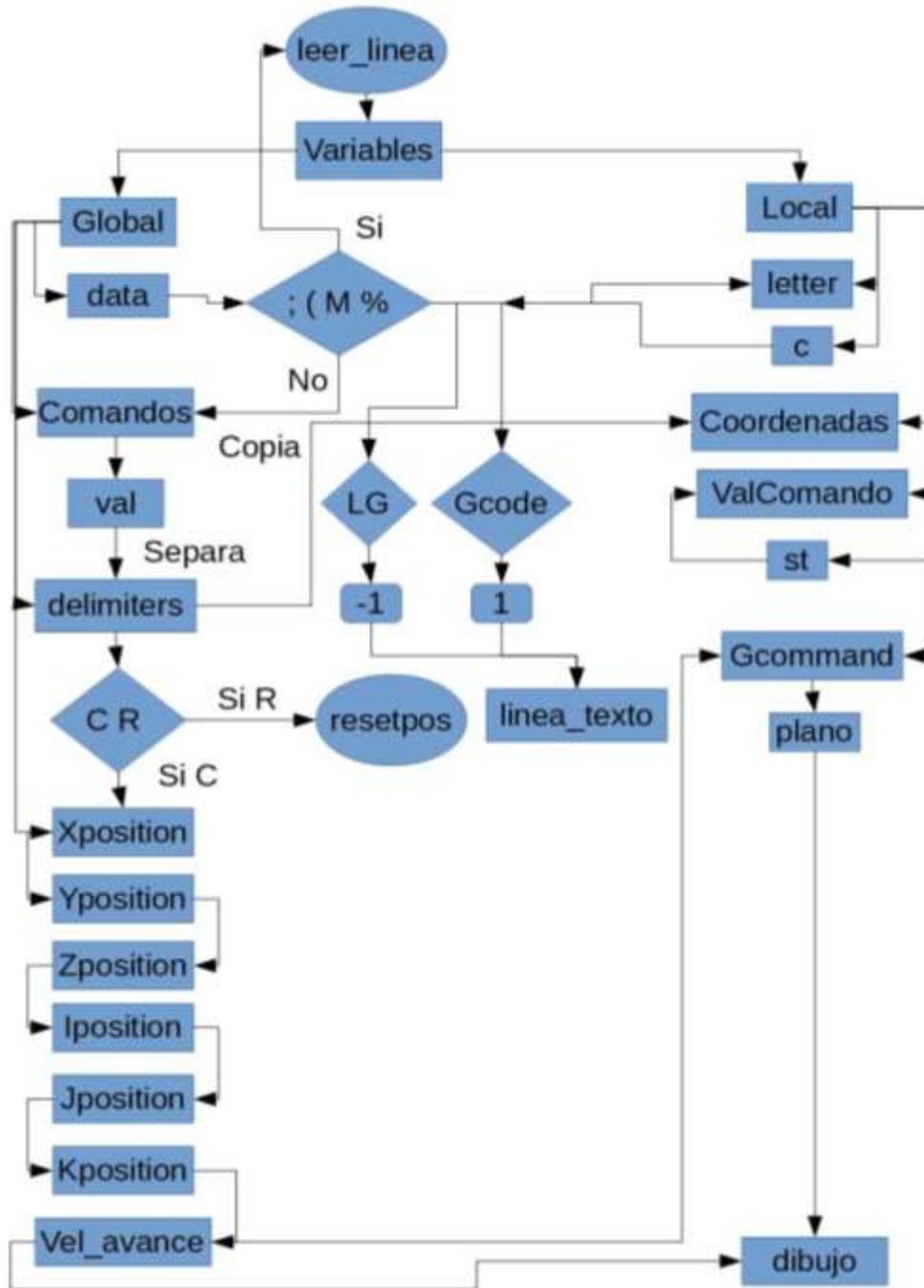
La variable global “data” es la encargada de leer un máximo de 64 caracteres de cada línea de código G. Las primeras posiciones de los caracteres suelen tener “; (M %” que son usados en el código para comentar y en el caso de “M” no se usa código auxiliar, por lo que al encontrar estos caracteres en alguna línea del texto los ignora y no procesa la lectura de código G.

En la variable global “Comandos” se almacena toda la cadena de caracteres leída de cada línea de la variable “data”; es necesario copiar ya que al separar los caracteres se cambia los datos almacenados de la variable “data”, y esta es necesaria para poder buscar luego los caracteres originales de cada línea.

La variable global “val” almacena los valores obtenidos al separar cada letra del valor correspondiente; para esto se usa la función “strtok” con los delimitantes que descomponen el código G. Existe dos tipos de variables que se usan con caracteres establecidos como “C” que indica cuando se ingresa los datos por comando y “R” que indica cuando se pulsa el botón de encerrar todas las variables de posición, que se encuentra en Labview.

Las dos variables de “LG” que sirve para limpiar la pantalla de labview al pulsar el botón de limpiar y la variable “Gcode” que inicia el proceso de lectura del archivo de código G cuando se pulsa el botón de “Ok”. Los valores de -1 se envían con cualquier botón que no sea el de iniciar la lectura de código G y el valor de 1 se usa para cuando se lee el archivo de texto de código g.

Se analiza las variables locales y globales que son necesarias para poder interpretar las instrucciones de código G. La función “leer\_línea” lee la cadena de caracteres del código G y las descompone para obtener los valores de G0, G1, G2, G3, X, Y, Z, I, J y Velocidad de avance.



**Figura 54.** Función leer\_línea

Las variables globales “Xposition, Yposition, Zposition, Iposition, Jposition, Kposition y Vel\_avance”, almacenan los valores del código G leído por el texto o por comandos.

La variable global de “línea\_texto” irá acumulando los valores de cada línea leída siempre y cuando el valor de línea sea diferente de cero. Por lo que si se aplica la fórmula de incremento en 1, cuando el valor es de -1 da resultado cero y cuando el valor es 1 va incrementando cada línea procesada en 1 hasta finalizar la lectura de todos los datos de texto.

Las variables locales son utilizadas únicamente en cada función por lo que la mayoría son variables auxiliares para procesar los valores temporalmente.

La variable local “letter” es usada para almacena en vector la cadena de caracteres y la variable local “c” es la auxiliar para incrementar en uno cada posición donde se almacena los caracteres.

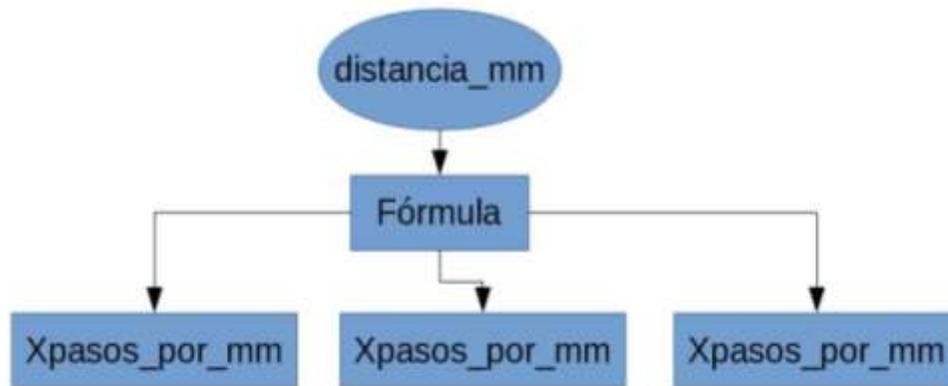
La variable local “Coordenadas” almacena los valores de cada instrucción del código G y con la variable auxiliar “ValComando” se incrementa en uno la posición donde se almacena los valores; lo mismo ocurre con la variable auxiliar “st” que incrementa en uno cuando encuentra una letra correspondiente al código G y la almacena en las variables globales Xposition, Yposition, Zposition, Iposition, Jposition, Kposition y Vel\_avance”.

La variable local “plano” solo es utilizada cuando encuentra instrucciones de código G para arcos, como I, J, K. El cálculo del arco solo es usado para graficar en dos planos por lo que el tipo de caracteres aceptados son para los planos “IJ, IK y JK”; en el caso de existir caracteres con las letras “IJK”, se ignora dicha instrucción ya que para ese proceso se debe utilizar la fórmula de cálculo de hélices, la cual no está implementada.

En la variable global “dibujo” se almacena el valor para graficar una línea, arco en el plano IJ, arco en el plano IK o arco en el plano JK.

## Función distancia\_mm

Se determina la escala que convierte los pasos por vuelta del motor a micro-pasos por mm.



**Figura 55.** Función distancia\_mm

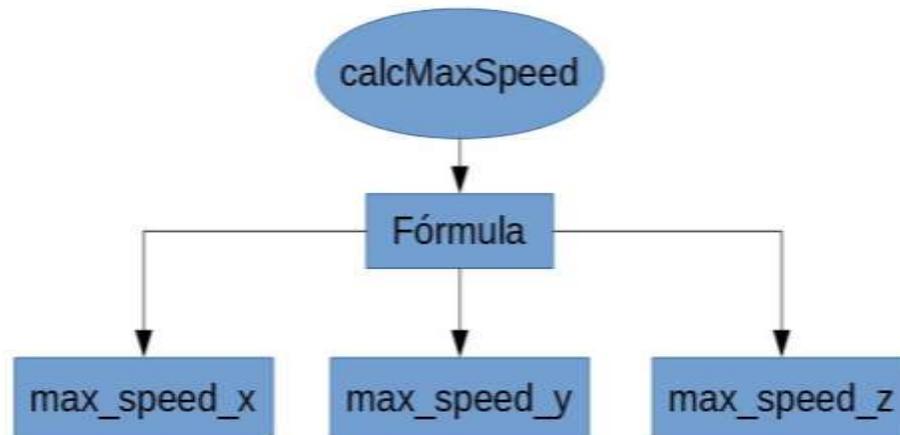
$$dist = \frac{\frac{pasos}{rev} * \frac{upasos}{pasos}}{\frac{mm}{rev}} = \frac{upasos}{mm}$$

### Ecuación número 1

Para calcular la distancia se usa la ecuación 1, donde el motor tiene 200 pasos/rev, el controlador se configura con 8 micro-pasos y el husillo tiene un paso de 1mm por revolución. Al usar el mismo husillo, motores y tipo de configuración del controlador del motor, la escala es la misma en los tres casos.

## Función CalcMaxSpeed

Determina el cálculo de la máxima velocidad en mm por minuto.



**Figura 56.** Función CalcMaxSpeed

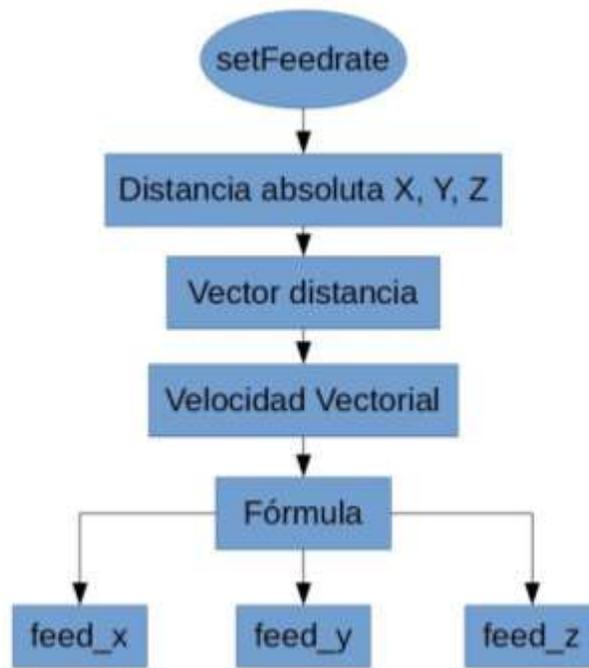
$$\text{max\_speed} = \frac{\text{Max\_velocidad} * 60.0}{\text{distancia\_mm}}$$

### **Ecuación número 2**

La ecuación 2 calcula la máxima velocidad de cada motor. Para obtener el valor de la velocidad máxima se necesita el valor máximo de la frecuencia soportado por Arduino con la librería “AccelStepper”, la cual es de 20000 pasos por segundo (20Khz), esta se multiplica por 60 para convertir los segundos a minutos y se divide para el valor de distancia\_mm de cada motor obtenida en la ecuación 1.

## Función setFeedrate

La figura 57 indica el proceso de cálculo de la velocidad de avance, convirtiendo los mm por minuto a pasos por segundo.



**Figura 57.** Función setFeedrate

$$\text{distancia absoluta} = \text{última posición} - \text{posición asignada}$$

### Ecuación número 3

La ecuación 3 calcula la diferencia entre la última posición y la posición a llegar, obteniendo el valor absoluto de la distancia en los ejes de X, Y, Z.

$$\text{vector dist} = \sqrt{\text{distancia absoluta}X^2 + \text{distancia absoluta}Y^2 + \text{distancia absoluta}Z^2}$$

### Ecuación número 4

La ecuación 4 calcula el vector de la distancia entre los ejes X, Y, Z.

$$Vel\ vectorial = \frac{Vel\_avance * distancia\ absoluta}{vector\ dist}$$

#### **Ecuación número 5**

La ecuación 5 calcula la velocidad vectorial en mm por minuto en los ejes X, Y, Z. Una vez obtenida se compara con la máxima velocidad calculada en la ecuación x calcMaxSpeed para restringir que la velocidad de avance obtenida del código G no sea mayor a la máxima velocidad permitida.

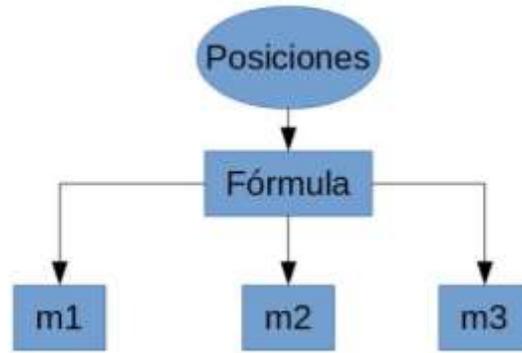
$$feed = \frac{Vel\ vectorial * dist}{60}$$

#### **Ecuación número 6**

La ecuación 6 calcula la velocidad de avance final dependiendo si la distancia a recorrer es corta o larga, varía la velocidad para determinar que el movimiento de los motores no sea abrupto. Esta velocidad de avance se convierte de mm por minuto a pasos por segundo para trabajar con las mismas unidades de velocidad de la librería “AccelStepper”, usando el valor de la ecuación 4.

### **Función Posiciones**

La figura 58 indica cómo se controla el valor de la distancia de avance de los motores de mm a pulsos que serán enviados a cada controlador para que muevan los motores.



**Figura 58.** Función de posiciones del motor.

$$motor\# = posición\ asignada(mm) * distancia\_mm(\frac{upasos}{mm})$$

**Ecuación número 7**

La ecuación 7 calcula la cantidad de pulsos en entero necesaria para mover cada motor al valor de distancia requerido.

**Función line**

La figura 59 indica el proceso de uso de la librería MultiStepper de Arduino para mover los motores a la posición asignada por el operador o código G, configura la velocidad de avance adecuada según la distancia, lee los límites de la máquina por medio de los finales de carrera y el botón de emergencia.

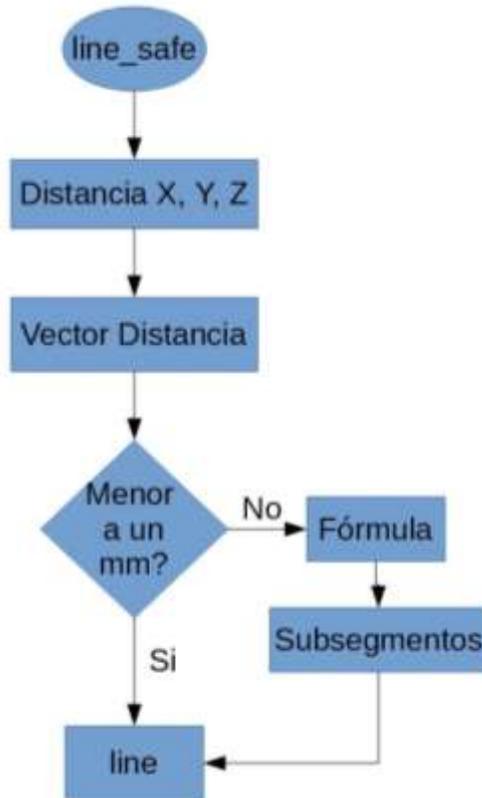


**Figura 59.** Función line.

La función line es utilizada para realizar cada movimiento de los motores cuando se va a graficar una línea o curva.

### **Función line\_safe**

La figura 60 calcula el vector distancia para que la librería “MultiStepper” interpole las coordenadas de los ejes X, Y, Z. Una vez obtenido el vector distancia se verifica que este no sea mayor a un mm para poder verificar los finales de carrera y botón de emergencia de la función “line” cada mm que avanza y así poder detener los motores en el caso de que alguno de estos sea pulsado.



**Figura 60.** Función line\_safe.

$$distancia = posición\ asignada - última\ posición$$

**Ecuación número 8**

La ecuación 8 calcula la diferencia entre la posición a llegar y la última posición, obteniendo el valor de la distancia en los ejes de X, Y, Z.

Una vez obtenido los valores de distancia de los ejes X, Y, Z; se calcula el vector de la distancia con la ecuación 4. Si el vector es menor de un mm se grafica la línea caso contrario la línea se divide en sub-segmentos del vector distancia aplicando la ecuación 10.

$$\text{Segmento} = \frac{\text{vector distancia}}{\text{MM\_por\_segmento}}$$

**Ecuación número 9**

$$\text{Subsegmentos} = \sum_{i=1}^{\text{Segmento}} \left( \left( \text{distancia} * \frac{i}{\text{Segmento}} \right) + \text{última posición} \right)$$

**Ecuación número 10**

$$\text{line} = \langle \text{SubsegmentoX} | \text{SubsegmentoY} | \text{SubsegmentoZ} \rangle$$

**Ecuación número 11**

La ecuación 11 calcula la línea que va dibujar en los puntos X, Y, Z, se aplica la misma fórmula con los valores correspondientes de posición de cada eje.

### **Función atan3**

La figura 61 determina el valor del ángulo usando la función matemática atan2 que calcula el arco tangente desde  $-\pi$  hasta  $\pi$ . Al obtener valores negativos de los ángulos en radianes se aplica la ecuación x para poder obtener los ángulos en radianes desde 0 hasta  $2\pi$ .



**Figura 61.** Función atan3.

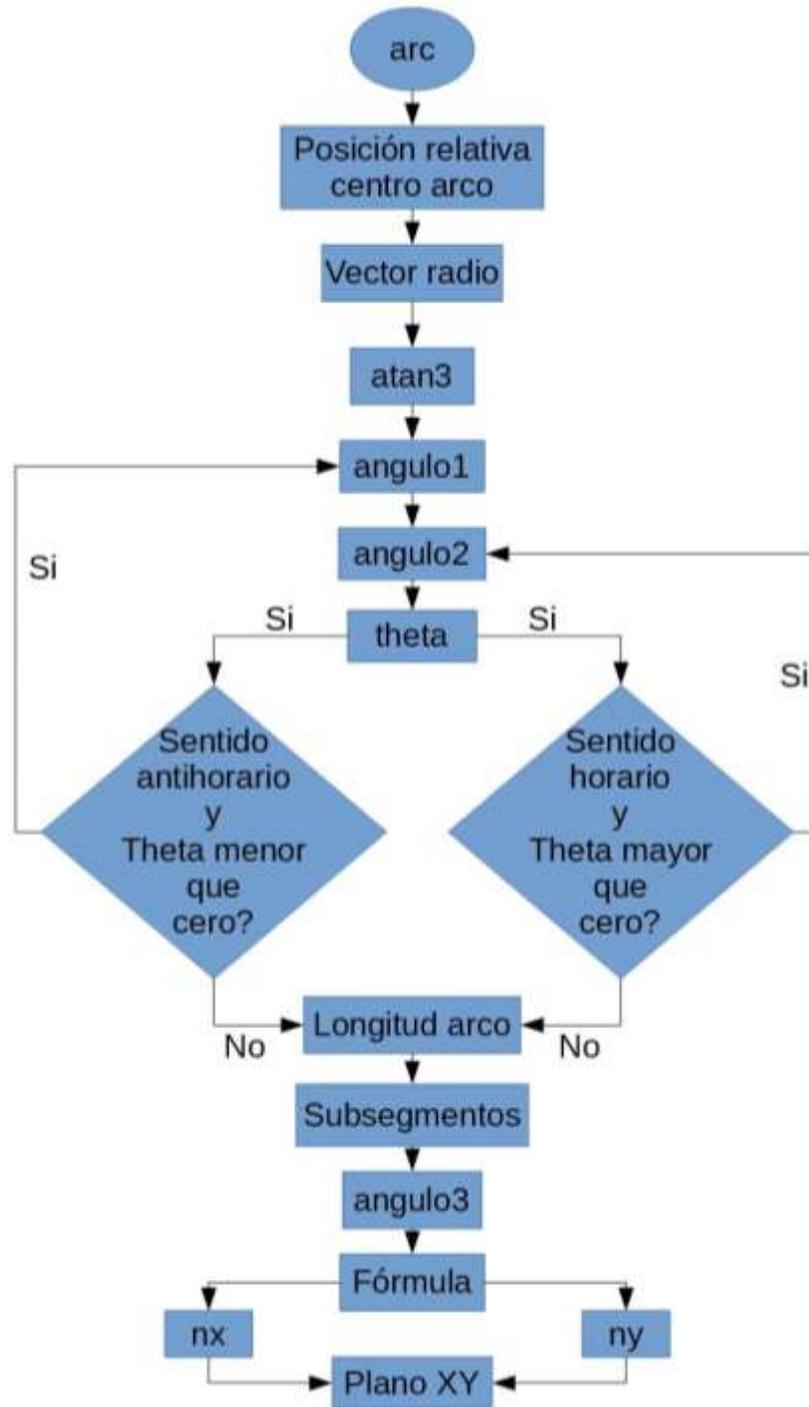
$$\acute{a}ngulo = 2\pi + \acute{a}ngulo$$

**Ecuación número 12**

La ecuación 12 recalcula el ángulo en radianes desde 0 hasta  $2\pi$ , siempre y cuando el valor del ángulo sea menor a cero.

**Función arc**

La figura 62 calcula las líneas que van a formar parte del arco y el sentido que tomará respecto a si el código G es G2 o G3. En el caso de ser G2 el arco se formará en sentido horario caso contrario si es G3 el arco se realizará en sentido anti-horario.



**Figura 62.** Función arc.

*centro arco = última posición + punto centro*

**Ecuación número 13**

La ecuación 13 calcula la posición del centro del arco tomando en cuenta la posición anterior y la nueva posición del centro en los ejes X, Y. Por ejemplo, si la última posición es X=0 y Y=0, y se quiere realizar un arco tomando como centro X=10 y Y=0; la posición del centro del arco será X=10 y Y=0.

$$vector\ radio = \sqrt{(última\ posX - centro\ arcoX)^2 + (última\ posY - centro\ arcoY)^2}$$

#### **Ecuación número 14**

La ecuación 14 determina el radio respecto al centro en X, Y. Para obtener el radio se determina la hipotenusa, ese va ser el valor del radio a lo largo del arco.

$$angulo1 = atan3 * ((última\ posY - centro\ arcoY), (última\ posX - centro\ arcoX))$$

#### **Ecuación número 15**

La ecuación 15 indica el valor del primer ángulo con el que va trazar la línea del arco, este ángulo se forma con respecto al punto inicial.

$$angulo2 = atan3 * ((posfinalY - centro\ arcoY), (posfinalX - centro\ arcoX))$$

#### **Ecuación número 16**

La ecuación 16 indica el valor del segundo ángulo con el que va trazar la línea del arco, este ángulo se forma con respecto al punto final.

$$\theta = \text{angulo2} - \text{angulo1}$$

#### **Ecuación número 17**

La ecuación 17 indica el grado de inclinación de la línea que va formar el arco con respecto a los valores de ángulo 1 y 2.

Si el valor del ángulo theta es menor que cero y el sentido del arco es horario, se usa la ecuación 18.

$$\text{angulo2} += 2\pi$$

#### **Ecuación número 18**

La ecuación 18 indica cuando se debe compensar el ángulo 2 para que forme el arco de forma correcta dependiendo de la inclinación de la línea y el sentido horario requerido.

Si el valor del ángulo theta es mayor que cero y el sentido del arco es anti-horario, se usa la ecuación 19.

$$\text{angulo1} += 2\pi$$

#### **Ecuación número 19**

La ecuación 19 indica cuando se debe compensar el ángulo 1 para que forme el arco de forma correcta dependiendo de la inclinación de la línea y el sentido anti-horario requerido.

Al recalcular los valores de los ángulos 1 y 2 se vuelve aplicar la ecuación x para encontrar el nuevo valor de theta.

$$\text{longitud arco} = |\theta| * \text{vector radio}$$

#### **Ecuación número 20**

La ecuación 20 determina el valor de la longitud de arco obteniendo el valor absoluto de theta y multiplicando por el valor del vector radio. Con la longitud de arco se va indicar en cuantos sub-segmentos de línea se tiene que dividir el arco.

$$Segmento = \frac{longitud\ arco}{MM\_por\_segmento}$$

**Ecuación número 21**

$$angulo\ 3 = \left( theta * \sum_{i=1}^{Segmento} \left( \frac{i}{Segmento} \right) \right) + angulo1$$

**Ecuación número 22**

$$Subsegmento\ nx = \sum_{i=1}^{Segmento} (centro\ arcoX + \cos(angulo3) * vector\ radio)$$

**Ecuación número 23**

$$Subsegmento\ ny = \sum_{i=1}^{Segmento} (centro\ arcoY + \sin(angulo3) * vector\ radio)$$

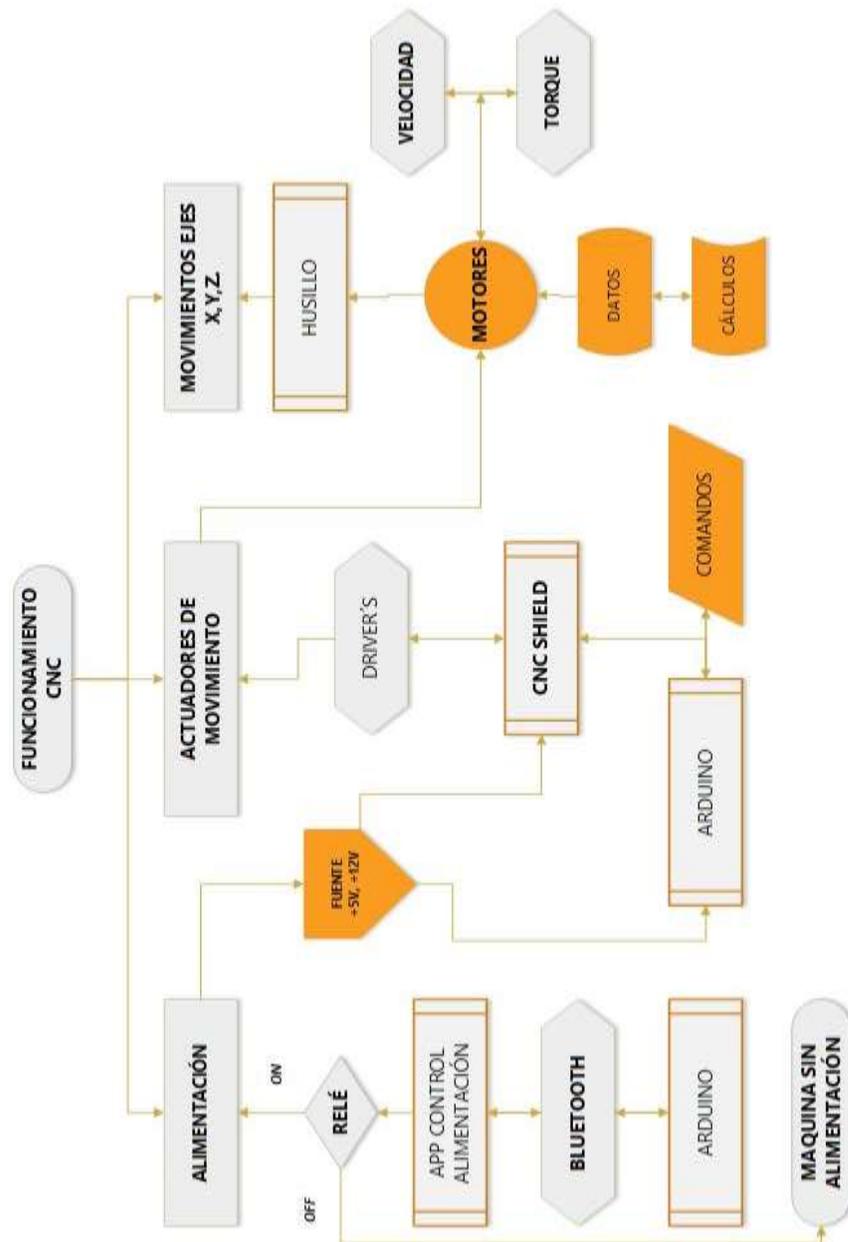
**Ecuación número 24**

$$line = \langle Subsegmento\ nx | Subsegmento\ ny | última\ posZ \rangle$$

**Ecuación número 25**

### 3.4. IMPLEMENTACIÓN

#### 3.4.1. DIAGRAMA DE FLUJO LA IMPLEMENTACIÓN DE UN SISTEMA DE MANEJO LINEAL PARA UNA MESA DE CONTROL NUMÉRICO COMPUTARIZADO.



**Figura 63.** Implementación de un sistema de manejo lineal para una mesa de control numérico computarizado.

### 3.4.2. IMPLEMENTACIÓN MECÁNICA

Describe el diseño de las piezas y estructuras realizadas que conforman el sistema del Router CNC.

El Router fue dividido en tres sistemas fundamentales lineales, sistema del eje lineal “X”, sistema del eje lineal “Y” y sistema del eje lineal “Z”; los cuales permiten un traslado preciso de la herramienta de corte.

#### 3.4.2.1. SISTEMA DEL EJE LINEAL “X”

El movimiento en “X”, requiere de un conjunto de piezas mecánicas que se acoplen para que permitan trasladar el eje “Z”. El mecanismo del eje “X” descansa y se desplaza sobre el husillo del eje “Y”.

El eje “X”, funciona con dos piezas principales: el husillo y la tuerca, el primero está sujeto directamente por medio de un acople flexible al motor paso a paso, el cual elimina cualquier variación del husillo con respecto al motor, la barra contiene una tuerca la cual realiza el desplazamiento por el rodamiento del husillo.



**Figura 64.** Eje lineal “X”.

### **3.4.2.2. SISTEMA DEL EJE LINEAL “Y”**

El movimiento en “Y”, requiere de un conjunto de piezas mecánicas que se acoplen para que proporcionen un desplazamiento a lo largo de los ejes horizontales.

El eje “Y”, funciona con dos piezas principales: dos husillos y dos tuercas, el primero está sujeta directamente por medio de un acople flexible al motor paso a paso, el cual elimina cualquier variación del husillo con respecto al motor, la barra contiene una tuerca la cual realiza el desplazamiento por el rodamiento de los husillos.



**Figura 65.** Eje lineal “Y”.

### **3.4.2.3. SISTEMA DEL EJE LINEAL “Z”**

El movimiento en “Z”, requiere de un conjunto de piezas mecánicas que se acoplen para que proporcionen un desplazamiento vertical y es el responsable de llevar a cabo la profundidad de corte de la pieza, el eje “Z”, se encuentra montado sobre el eje “X”

El eje “Z”, funciona con dos piezas principales: el husillo y la tuerca, el primero está sujeta directamente por medio de un acople flexible al motor paso a paso, que elimina cualquier variación del husillo con respecto al motor, la barra contiene una tuerca la cual realiza el desplazamiento por el rodamiento del husillo.

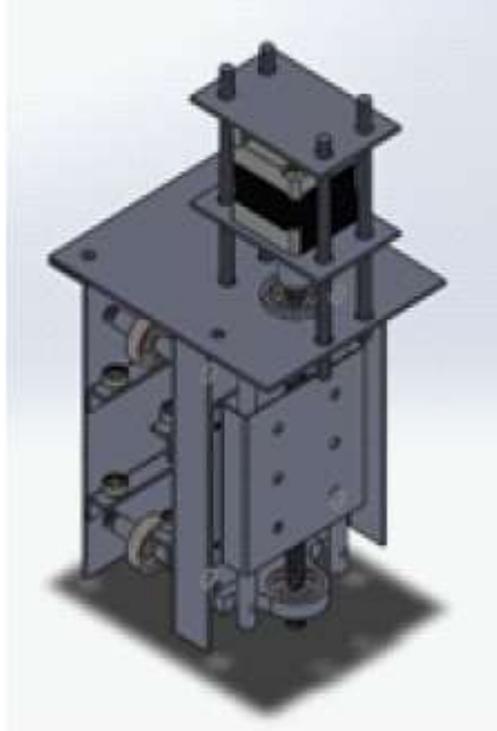
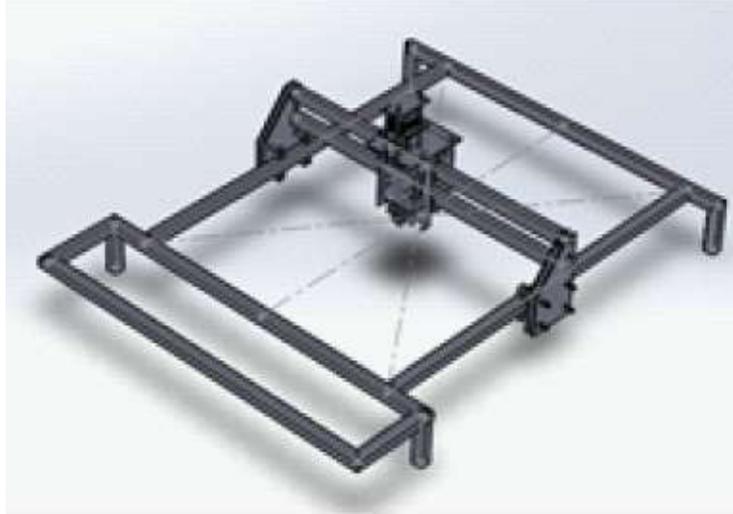


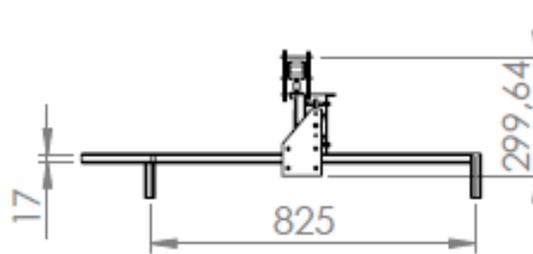
Figura 66. Eje lineal “Z”.

#### ***3.4.2.4. MONTAJE FINAL DE LAS ESTRUCTURAS***

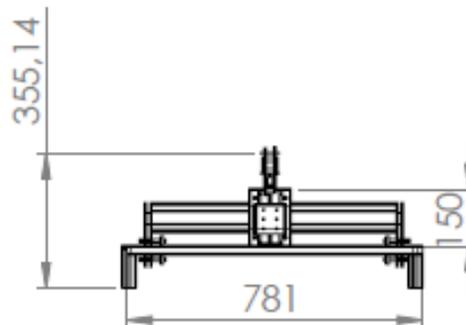
A continuación, se presenta el montaje de la estructura del router con todas sus partes que la componen.



**Figura 67.** Estructura Final.



**Figura 68.** Cotas Horizontales.



**Figura 69.** Cotas Frontales.

### 3.4.3. IMPLEMENTACIÓN DEL SISTEMA ELECTRÓNICO

A continuación, se observará como fue el procedimiento para lograr el funcionamiento del router, de acuerdo con un orden establecido: el Arduino Uno, la CNC shield, el driver que se usara para el control de Motores y finalmente la comunicación.

Una vez cargado el programa en Arduino se interactuó con el CNC shield y los drivers, llegando a tener una comunicación con el programa realizado en el Software Labview. Ya con nuestro Panel de Control en Labview podemos mandar pulsos para que el sistema lineal se mueva en diferentes coordenadas como son X, Y, Z.

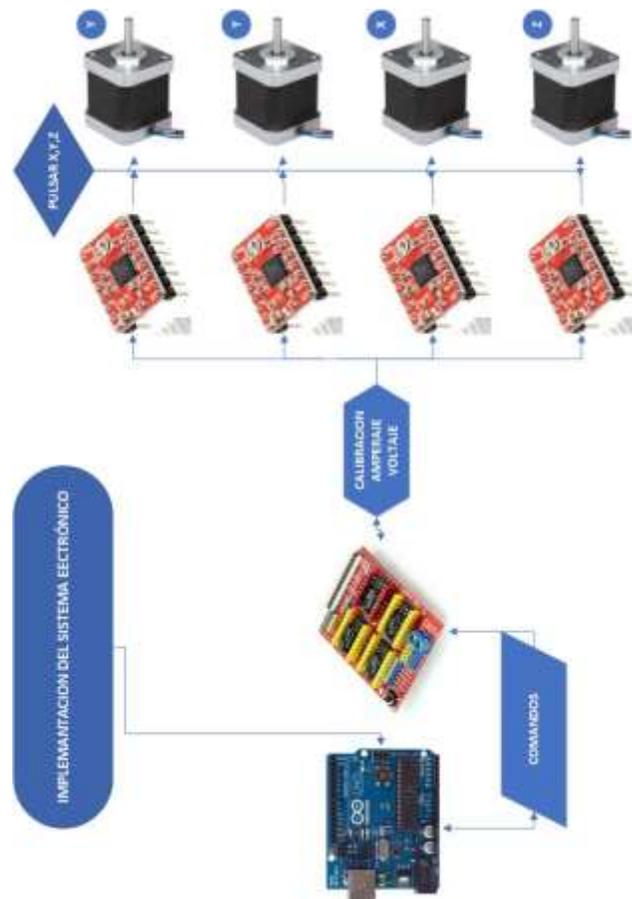
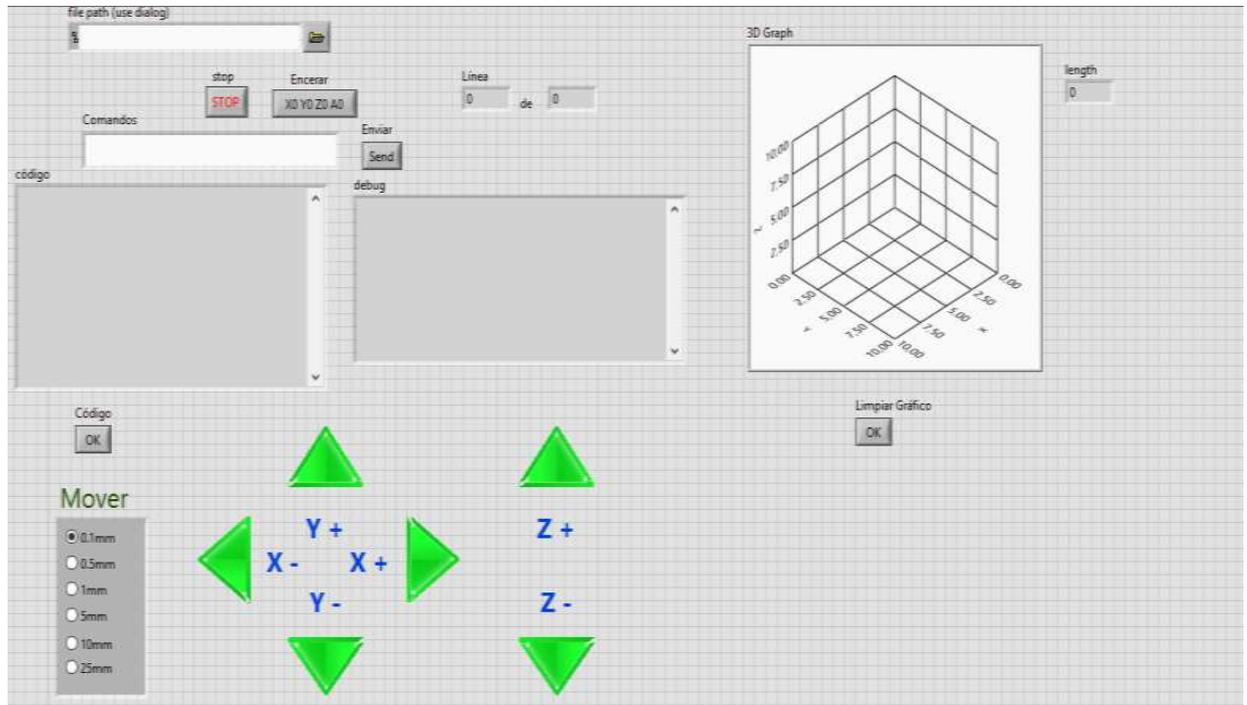


Figura 70. Diagrama de flujo de proceso

### 3.4.4. IMPLENENTACION INTERFAZ LABVIUW

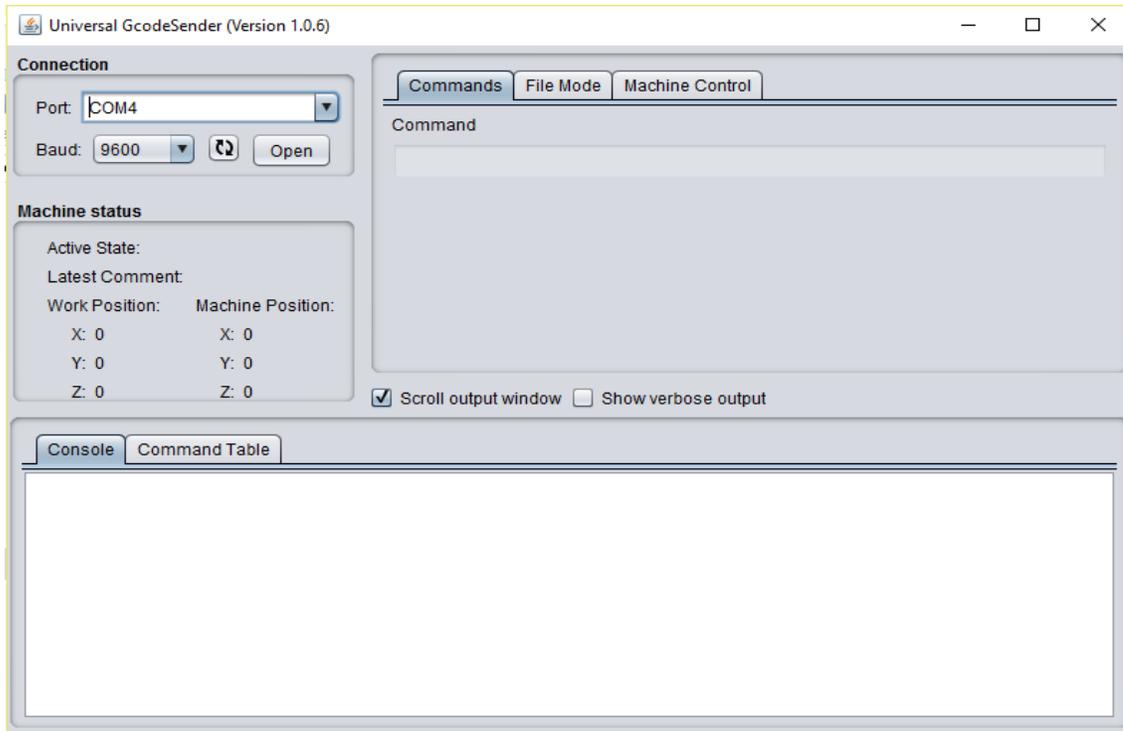
Con los diferentes bloques de programación se desarrolló el panel de control de un equipo de corte CNC. El programa se comunica con el Arduino para obtener los movimientos en X, Y, Z.



**Figura 71.** Panel de control LabVIEW.

### 3.4.5. IMPLEMENTACIÓN INTERFAZ G -CODE SENDER

Se interactuó el programa Universal G-Code Sender con el Arduino para demostrar por este método también el movimiento de los ejes en X, Y, Z, que facilitará los movimientos lineales por medio de una aplicación Android.



**Figura 72.** Panel de control Universal G-Code Sender V1.0.6.

### **3.5. PRUEBAS DE FUNCIONAMIENTO**

#### **3.5.1. SISTEMA MECÁNICO**

Una vez completado el montaje total de la máquina, está lista para trabajar y comprobar sus diferentes sistemas, en el caso del sistema mecánico, es decir el sistema motriz que incluyen motores, ejes lineales horizontales, ejes lineales verticales, ejes deslizantes, rodamientos deslizantes, además la estructura soporte, las primeras pruebas deben realizarse con cuidado.

**Tabla 12.***Movimientos mecánicos.*

<b>ACCIÓN</b>	<b>PARTES</b>	<b>VERIFICACIÓN</b>	<b>OBSERVACIONES</b>
	<b>MECÁNICAS</b>		
	<b>INVOLUCRADAS</b>		
	Estructura	La estructura no presenta ninguna alteración, trabaja en óptimas condiciones.	
	Ejes lineales Y	Deslizamiento a una velocidad constante	
	Eje lineal X	Deslizamiento a una velocidad constante	La estructura es plena para cortar con plasma,
<b>Movimientos</b>	Eje lineal Z	Deslizamiento a una velocidad constante	obteniendo peso idóneo para que los movimientos
<b>Mecánicos</b>	Soportes	No se afecta con las Vibraciones	sean en diferentes sentidos
	Rodamientos	Tiene comportamiento normal	
	Motor Paso a Paso	Permitió los micro-pasos necesarios, obteniendo el torque y velocidad requerido	

### 3.5.2. SISTEMA ELECTRÓNICO

Se controla la maquina CNC con la interconexión del software **Arduino- Universal G-Code Sender** y **Arduino- LabVIEW**, para controlar los motores.

El sistema electrónico se conecta desde el control de la fuente de poder, arduino, CNC shield, drivers y motores, para obtener los movimientos dados desde un panel de control.

**Tabla 13.**

*Control de Movimientos.*

ACCIÓN	PARTES ELECTRÓNICAS INVOLUCRADAS	VERIFICACIÓN	OBSERBACIONES
<b>Control de los Movimientos</b>	Fuente de Poder	Ingresa 110 o 220V, que transforma el voltaje a requerir: +12V, +5V	
	Arduino	El número de entradas y salidas, digitales y analógicas permiten acoplarse con el CNC Shiel	Todos estos elementos acoplados a un panel de control permitieron
	CNC Shield	Permite la comunicación de los drivers con el Arduino	realizar movimientos deseados
	Drivers	Se regula el amperaje manualmente para poder controlar los movimientos de los motores	

Motores	Por la interconexión de los módulos mencionados se logra los movimientos deseados
---------	---

### 3.5.3. HARDWARE DRIVER A4988

Permitió la conexión de los pines estándar para socket SHIEL CNC Arduino Uno.

**Tabla 14.**

*Calibración driver A4988.*

<b>VALOR (Voltios)</b>	<b>RESULTADO OPERACIÓN DEL MOTOR</b>
<b>0,30</b>	Valor fuera de rango
<b>0,45</b>	Zumbido de motor sin moverse
<b>0,60</b>	Zumbido fuerte del motor, solo pulsa
<b>0,75</b>	Baja el zumbido, gira normal, pierde pasos, bajo torque
<b>0,90</b>	Operación normal, no hay pérdida de pasos, bajo torque
<b>0,96</b>	Operación normal, buen torque
<b>1,00</b>	Operación normal buen torque, recalienta driver
<b>1,05</b>	Pierde torque, aumenta recalentamiento
<b>1,30</b>	Valor fuera de rango

### 3.5.4. FIRMWARE

Por medio del programa **Universal G-Code Sender**, se calibraron por seteo los valores de los pasos por milímetro dentro de los rangos.

**Tabla 15.**

*Calibración Micro pasos del motor.*

<b>VALOR (STEP/mm)</b>	<b>RESULTADO OPERACIÓN DEL MOTOR</b>
<b>10</b>	Movimiento continuo, sin pérdida de pasos
<b>30</b>	Movimiento continuo, sin pérdida de pasos
<b>50</b>	Movimiento continuo, sin pérdida de pasos
<b>142</b>	Movimiento continuo, sin pérdida de pasos
<b>200</b>	Movimiento continuo, sin pérdida de pasos
<b>400</b>	Movimiento continuo, sin pérdida de pasos
<b>800</b>	Pérdida ocasional de pasos
<b>1500</b>	Pérdida de pasos secuencial
<b>10000</b>	Vibración y recalentamiento del motor

### 3.5.5. SOFTWARE

Con estas combinaciones se puede realizar movimientos en X, Y, Z.

**Tabla 16.***Interconexiones para movimientos.*

<b>ACCIÓN</b>	<b>PARTES</b>	<b>VERIFICACIÓN</b>	<b>OBSERBACIONES</b>
	<b>PROGRAMABLES</b>		
	<b>INVOLUCRADAS</b>		
		Se conectó por medio de	
	ARDUINO - APP	Bluetooth al arduino con el	
	INVENTOR 2	Smartphone para alimentar	
		la maquina CNC.	
		Se conectó el arduino con	
		el software y el panel de	
		control predeterminado	
	ARDUINO - UNIVERSAL	V1.0.6 por medio de un	Se realizó
	G-CODE SENDER	puerto de comunicación	movimientos por
<b>PANEL DE</b>		COM, para observar los	medio de
<b>CONTROL</b>		movimientos realizados en	programación en
		X,Y,Z.	X,Y,Z, el cual por
		Se conectó el arduino con	medio de una interfaz
		LabView y el panel de	puede interpretar
		control creado en	código G para realizar
		programación de bloques,	piezas en proceso.
	ARDUINO - LABVIEW	por medio de un puerto de	
		comunicación COM, para	
		observar los movimientos	
		realizados en X,Y,Z.	

### 3.6. ANÁLISIS ECONÓMICO

#### 3.6.1. COSTOS DE MATERIALES Y EQUIPOS PARA LA MESA DE CORTE CON PLASMA CNC

Con el fin de realizar un análisis económico comparativo entre la manufacturación de la mesa y la importación de esta, se registraron cada uno de los costos de la elaboración del proyecto, esto es en base a los materiales, accesorios y equipos seleccionados, además se analizó costos de mano de obra y ensamblaje.

**Tabla 17.**

*Tabla de costos.*

<b>DESCRIPCIÓN</b>	<b>CANTIDAD</b>	<b>Costo Unitario</b>	<b>Costo Total</b>
<b>Arduino Uno</b>	2	\$10,71	\$21,42
<b>Kit Cables 65 Jumper</b>	1	\$1,34	\$1,34
<b>Pernos 8x40 acero</b>	20	\$0,16	\$3,2
<b>Tuercas 8M acero</b>	40	\$0,1	\$4
<b>Pernos 12x40</b>	18	\$0,45	\$8,1
<b>Tuercas 12mm acero</b>	18	\$0,15	\$2,7
<b>Varillas roscadas 1/4</b>	5	\$1,1	\$5,5
<b>Varillas roscadas 5/16</b>	3	\$1,45	\$4,35
<b>Macho Hembra 40</b>	1	\$1,79	\$1,79
<b>Juego de cables largos</b>	1	\$2,23	\$2,23

<b>HC-06 Bluetooth</b>	1	\$8,04	\$8,04
<b>CNC Shield V3</b>	2	\$8,93	\$17,86
<b>Driver paso a paso A4988</b>	8	\$3,57	\$28,56
<b>Nema 17 Motor Pasos</b>	4	\$14,25	\$57
<b>Arduino Uno Case</b>	1	\$14,29	\$14,29
<b>CTO.ROD.PLANA 1/8</b>	2	\$4,46	\$8,92
<b>CTO.ROD.PLANA 5/32</b>	1	\$0,3	\$0,3
<b>CTO.ROD.PLANA 3/16</b>	1	\$0,6	\$0,6
<b>CTO.ROD.PLANA GALV 5/16</b>	1	\$2,5	\$2,5
<b>CTO.ROD.PRES. GALV 5/16</b>	1	\$1,5	\$1,5
<b>CTO.ROD.PRES.3/16 NEGRA</b>	1	\$0,6	\$0,6
<b>ROD, PLANA HIERRO 5/16</b>	1	\$2	\$2
<b>TUBE CONTACT S.C.R O,35 X 1,125</b>	4	\$0,82	\$3,28
<b>NOZZLE SLIP TYPE, 625</b>	1	\$22,85	\$22,85
<b>DEFLECTOR</b>	1	\$14,94	\$14,94
<b>RADNOR 16 OUNCE JAR NOZZLE</b>	1	\$5,73	\$5,73
<b>GEL</b>			
<b>PMX 30 CSA INTERNATIONAL HAND</b>	1	\$1889,35	\$1889,35
<b>SYSTEM</b>			
		<b>Sub-total</b>	\$2132,95
		<b>IVA %</b>	\$255,95
		<b>TOTAL</b>	\$2.388,90

### **3.6.2. RENTABILIDAD DE LA MESA CNC.**

La mesa de control numérico computarizado de corte con plasma, en el mercado se consigue en precios sumamente elevados debido a las salvaguardias, salida de divisas, e impuestos de importación.

Los precios de las máquinas CNC en el extranjero se encuentran en precios a partir de \$ 1.000 las más básicas y hasta alrededor de \$ 60.000 las más sofisticadas, sin incluir precios de importación a Ecuador, ni garantía, mucho menos mantenimiento.

Cuando se puso en práctica este proyecto se pensó en precios de competitividad con empresas importadoras de máquinas CNC, facilitando precios al sector industrial mecánico, para que en el Ecuador todos los talleres mecánicos se permitan tener y manipular maquinas automatizadas facilitando los tiempos de trabajo y tener más desarrollo industrial.

## CONCLUSIONES

Después del análisis y discusión de los resultados obtenidos se presentará las conclusiones del presente estudio, permitiendo visualizar el correcto funcionamiento de una máquina automatizada para corte de piezas con plasma. A continuación, se detallarán las conclusiones en base a los objetivos específicos planteados.

Con respecto al primer objetivo específico, se logró comunicar la PC con la máquina por medio de la programación realizada en Arduino obteniendo movimiento de los carros mecánicos que se desplazan en sus respectivos ejes.

En relación con el segundo objetivo específico, la aplicación realizada en APP Inventor 2 se conectó correctamente por medio de Bluetooth con el Arduino, permitiendo cerrar y abrir la fase controlada por el relé que permite alimentar y apagar la fuente de poder de la mesa de corte de control numérico computarizado de corte con plasma.

De acuerdo con el tercer objetivo específico, al interconectarse el Arduino con el programa realizado en LabView, se obtuvo un panel de control, el cual se comunicará por un puerto de comunicación con el programa de Arduino, permitiendo observar el desplazamiento de los ejes lineales en X, Y, Z.

De la misma manera en el cuarto objetivo específico, al Interconectarse el Arduino con el Universal G-Code Sender, se logró verificar el torque, velocidad de los motores paso a paso ya

conectados con el husillo, obteniendo el desplazamiento uniforme de acuerdo con el programa implementado de control.

Con el Universal G-Code Sender V1.0.6., se logró interpretar el código G, con ello la máquina logra dibujar las piezas vectorizadas con el programa Inkscape.

Ahora bien, con respecto al quinto objetivo específico, la máquina de corte CNC, está diseñada como una multi-herramienta que puede ser usada para corte con plasma, impresión de circuitos electrónicos en baquelita y taladro.

Finalmente, en relación con el último objetivo específico, se obtuvo un diseño mecánico con buenos resultados ya que los desplazamientos son nítidos, con un factor de vibración bien reducido, con un peso manipulable, logrando una ergonomía Hombre-Máquina.

Logrando construir una máquina de corte con control numérico computarizado a un precio sumamente económico al alcance de las pequeñas y medianas industrias enfocadas en la mecánica y electrónica.

## **RECOMENDACIONES**

- Capacitar al personal que va a estar a cargo de la manipulación de la máquina automatizada.
- Durante el proceso de corte de la pieza se debe verificar que el personal se rija a los estándares de seguridad usando vestimenta industrial protección necesaria para la manipulación de la máquina.
- Verificar conexiones eléctricas ya que estamos manipulando voltajes alternos y continuos, de modo que si existen variaciones de voltajes podrían quemar los sistemas como fuente, sistemas embebidos, motores.
- Realizar mantenimiento mecánico y electrónico preventivos cada año.

## BIBLIOGRAFÍA

- Codinter. (01 de Enero de 2018). *Capacitacion del compresor* . Obtenido de Codinter:  
<https://www.codinter.com/en/how-to-use-plasma-cutting-equipment-some-recommendations/>
- Electrónicos Caldas. (05 de Febrero de 2016). *HC-06*. Obtenido de Guangzhou HC Information Technology Co., Ltd. : [http://www.electronicoscaldas.com/datasheet/HC-06\\_Wavesen.pdf](http://www.electronicoscaldas.com/datasheet/HC-06_Wavesen.pdf)
- esi2 . (15 de Febrero de 2009). *Tutorial Labview*. Obtenido de esi2 :  
<http://www.esi2.us.es/~asun/LCPC06/TutorialLabview.pdf>
- FuentePropia. (01 de Agosto de 2017).
- Galdós, B. P. (18 de Abril de 2010). *Comenzando con Arduino*. Obtenido de Edificio Centro Tecnológico Cádiz:  
[http://www.uca.es/recursos/doc/Unidades/Unidad\\_Innovacion/Innovacion\\_Docente/ANE\\_XOS\\_2011\\_2012/22232441\\_310201212102.pdf](http://www.uca.es/recursos/doc/Unidades/Unidad_Innovacion/Innovacion_Docente/ANE_XOS_2011_2012/22232441_310201212102.pdf)
- González, V. (02 de Mayo de 2002). *Motores paso a paso*. Obtenido de Fundamentos de Robótica:  
[http://platea.pntic.mec.es/vgonzale/cyr\\_0204/cyr\\_01/robotica/sistema/motores\\_p-p.htm](http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/motores_p-p.htm)
- Hypertherm. (02 de Agosto de 2016). *Powermax30Air*. Obtenido de Hypertherm:  
[https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwiBsseG9LDZAhUCsFMKHRniBLIQFggvMAE&url=https%3A%2F%2Fwww.hypertherm.com%2FDownload%3FfileId%3DHYP134494%26zip%3DFalse&usg=AOvVaw3EIVXAt4nfaVVz\\_dstys0W](https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwiBsseG9LDZAhUCsFMKHRniBLIQFggvMAE&url=https%3A%2F%2Fwww.hypertherm.com%2FDownload%3FfileId%3DHYP134494%26zip%3DFalse&usg=AOvVaw3EIVXAt4nfaVVz_dstys0W)
- Kruger, B. (29 de Septiembre de 2013). *Protoneer*. Obtenido de Arduino CNC Shield:  
<https://blog.protoneer.co.nz/arduino-cnc-shield-v3-00-assembly-guide/>

Llamas, L. (08 de 23 de 2016). *Motores paso a paso con Arduino y driver A4988 o DRV8825*.

Obtenido de Luisllamas: <https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825/>

pbcllinear. (06 de Noviembre de 2015). *Stepper Motor NEMA 17*. Obtenido de pbcllinear:

[https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0ahUKEwjJvp\\_t-](https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0ahUKEwjJvp_t-7DZAhWR7VMKHTn1AOsQFgg9MAM&url=http%3A%2F%2Fwww.pbcllinear.com%2FDownload%2FDataSheet%2FStepper-Motor-Support-Document.pdf&usg=AOvVaw05BICl8Uk4CRPIFriEhhq0)

[7DZAhWR7VMKHTn1AOsQFgg9MAM&url=http%3A%2F%2Fwww.pbcllinear.com%2FDownload%2FDataSheet%2FStepper-Motor-Support-](https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0ahUKEwjJvp_t-7DZAhWR7VMKHTn1AOsQFgg9MAM&url=http%3A%2F%2Fwww.pbcllinear.com%2FDownload%2FDataSheet%2FStepper-Motor-Support-Document.pdf&usg=AOvVaw05BICl8Uk4CRPIFriEhhq0)

[Document.pdf&usg=AOvVaw05BICl8Uk4CRPIFriEhhq0](https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0ahUKEwjJvp_t-7DZAhWR7VMKHTn1AOsQFgg9MAM&url=http%3A%2F%2Fwww.pbcllinear.com%2FDownload%2FDataSheet%2FStepper-Motor-Support-Document.pdf&usg=AOvVaw05BICl8Uk4CRPIFriEhhq0)

sapiensman. (12 de FEBRERO de 2012). *SOLDADURA Y CORTE CON ARCO DE PLASMA*.

Obtenido de sapiensman:

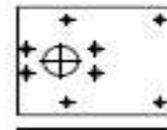
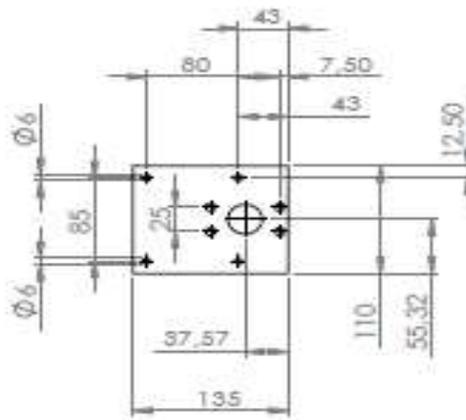
[http://www.sapiensman.com/tecnoficio/soldadura/soldadura\\_de\\_plasma.php](http://www.sapiensman.com/tecnoficio/soldadura/soldadura_de_plasma.php)

# **ANEXOS**

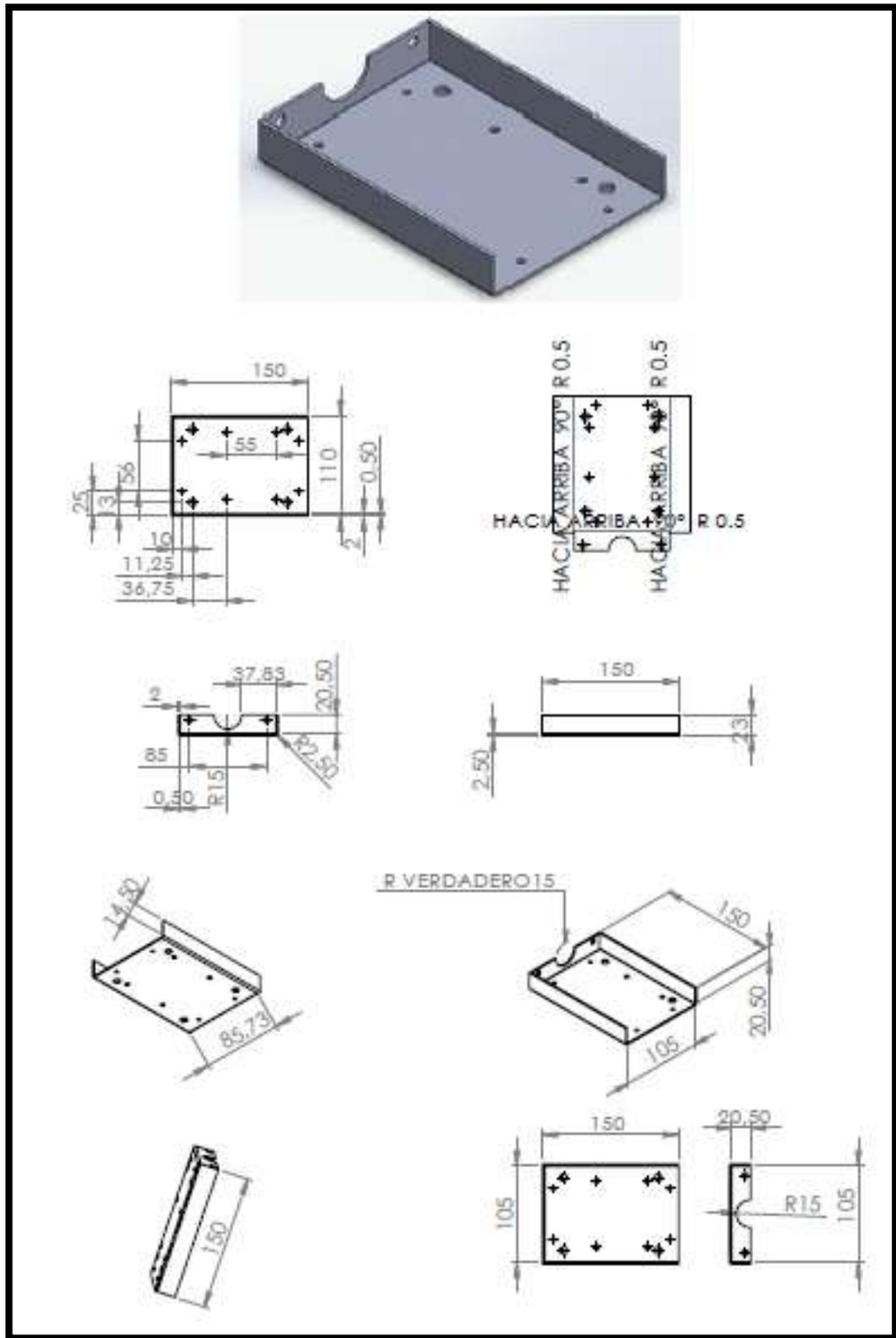
## ANEXO A

### SISTEMA MECÁNICO

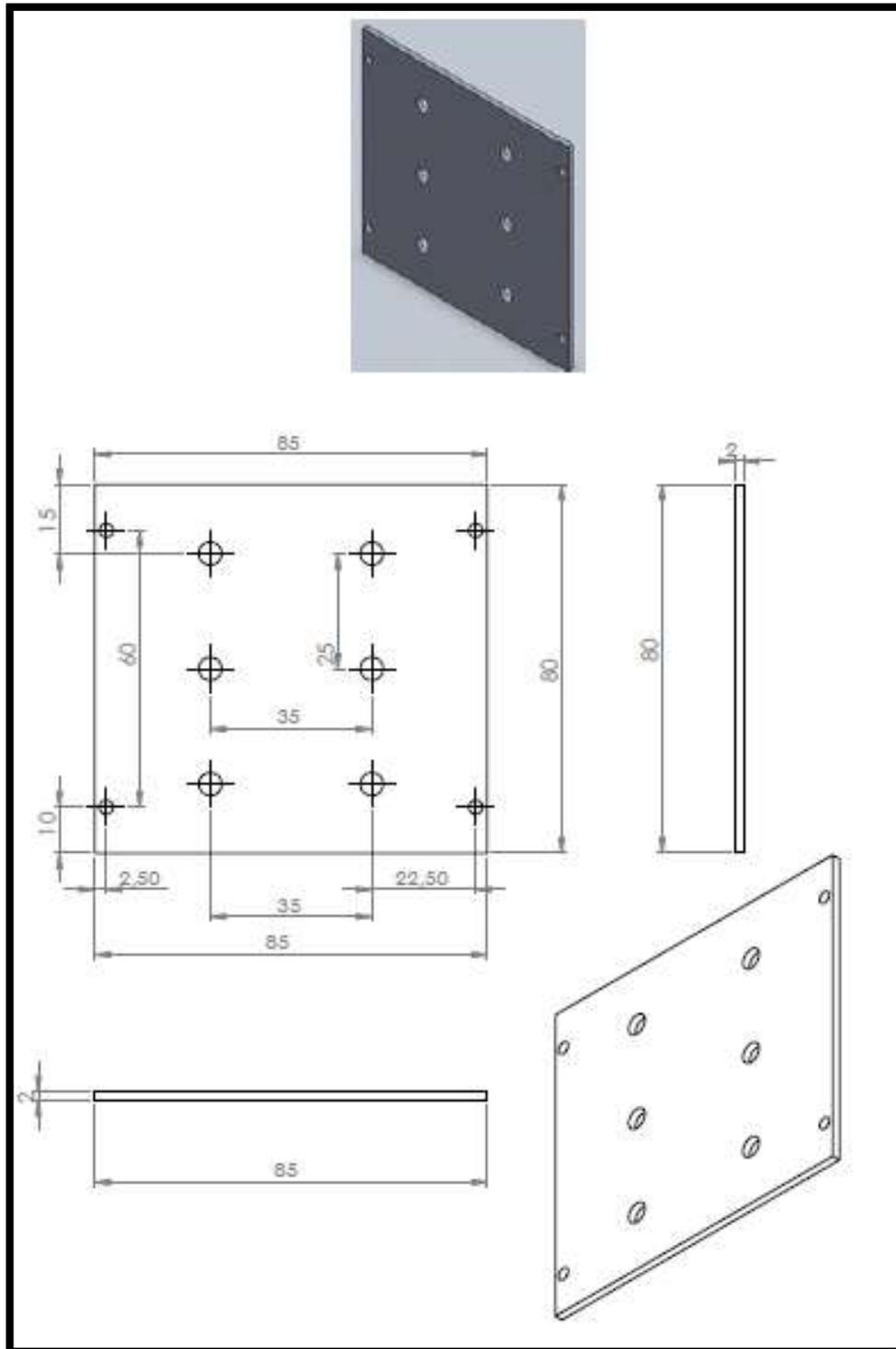
- Base del motor del carro Z.



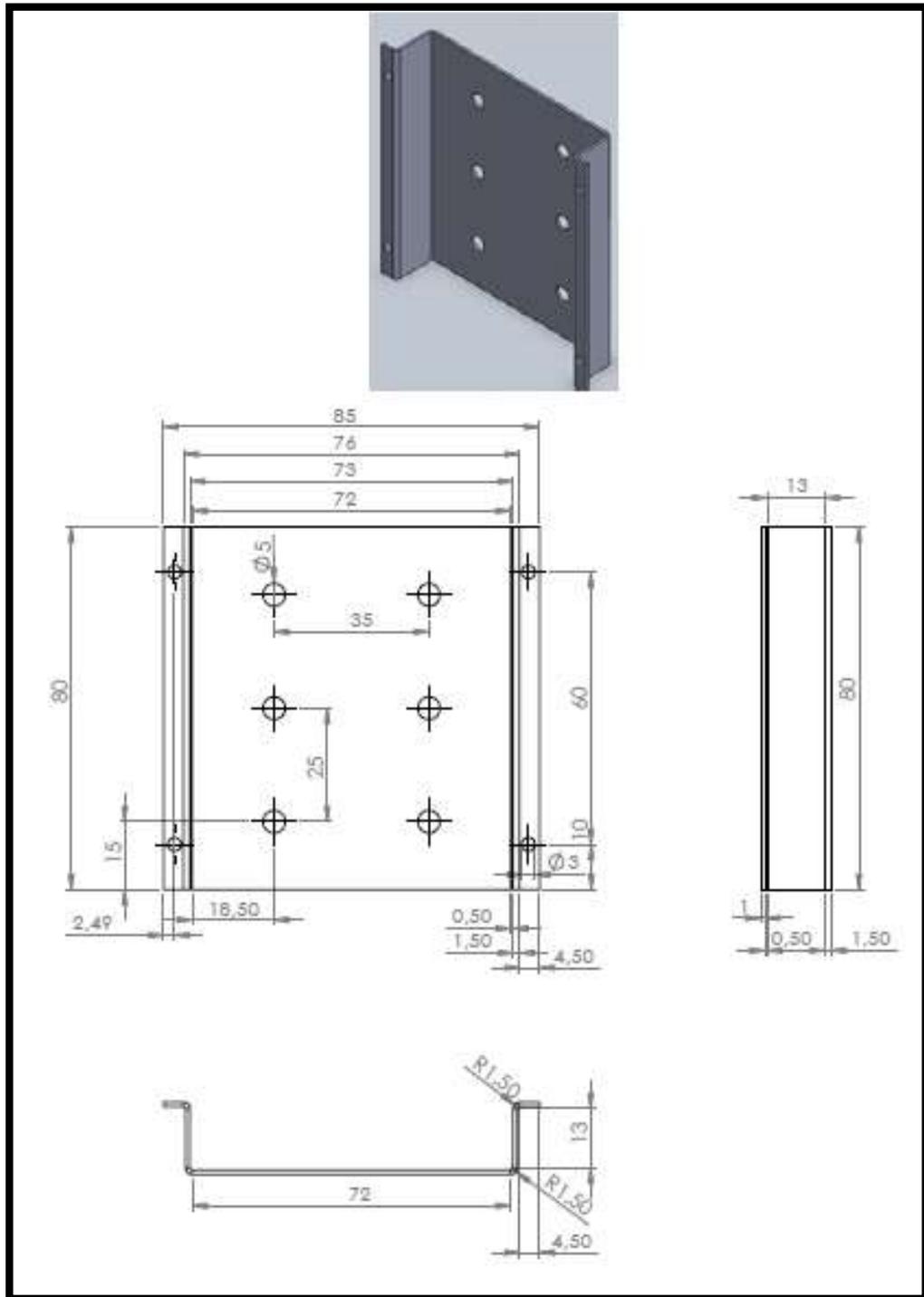
- Chapa delantera A del Carro Z.



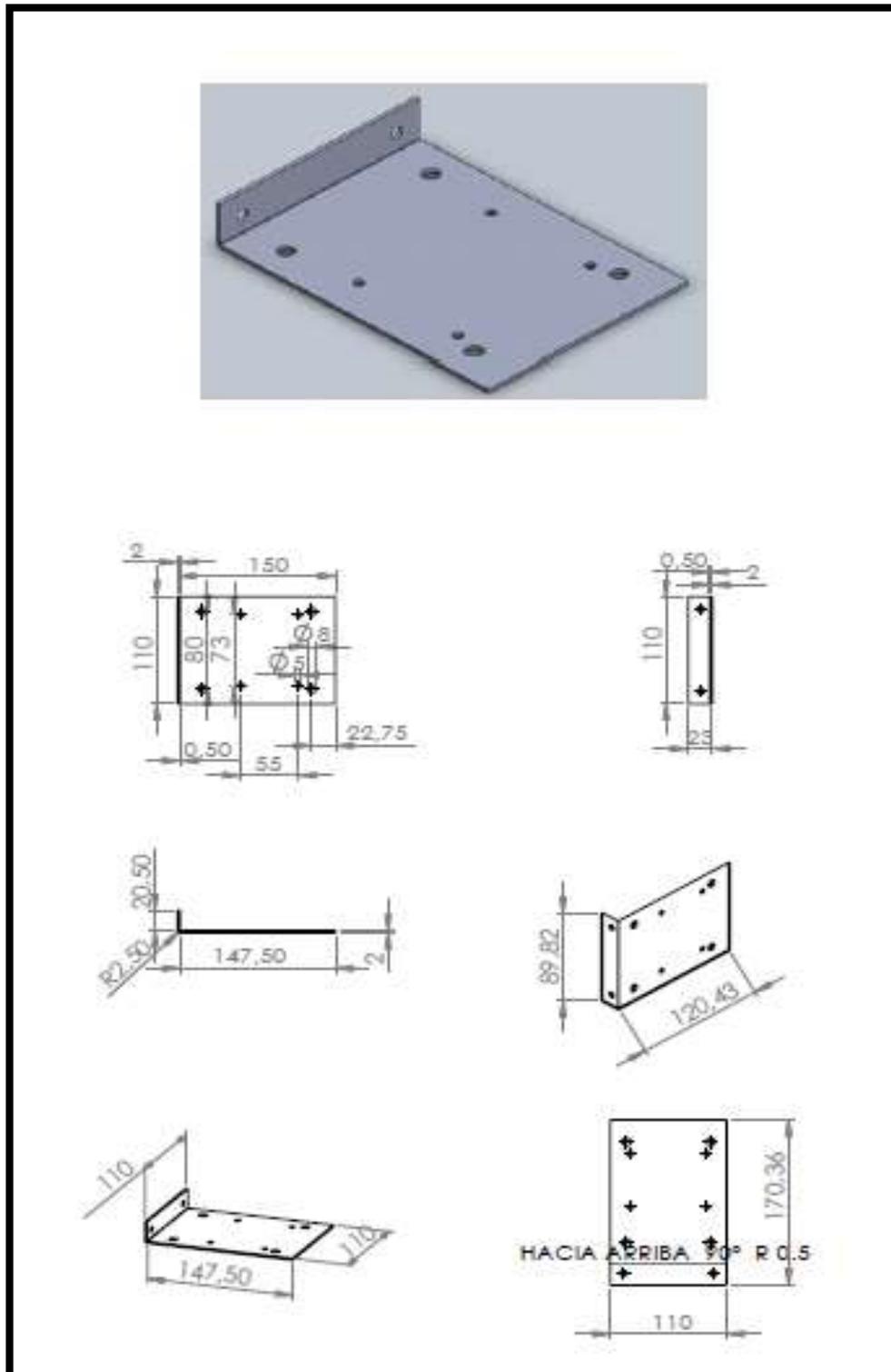
- Chapa delantera B del carro Z



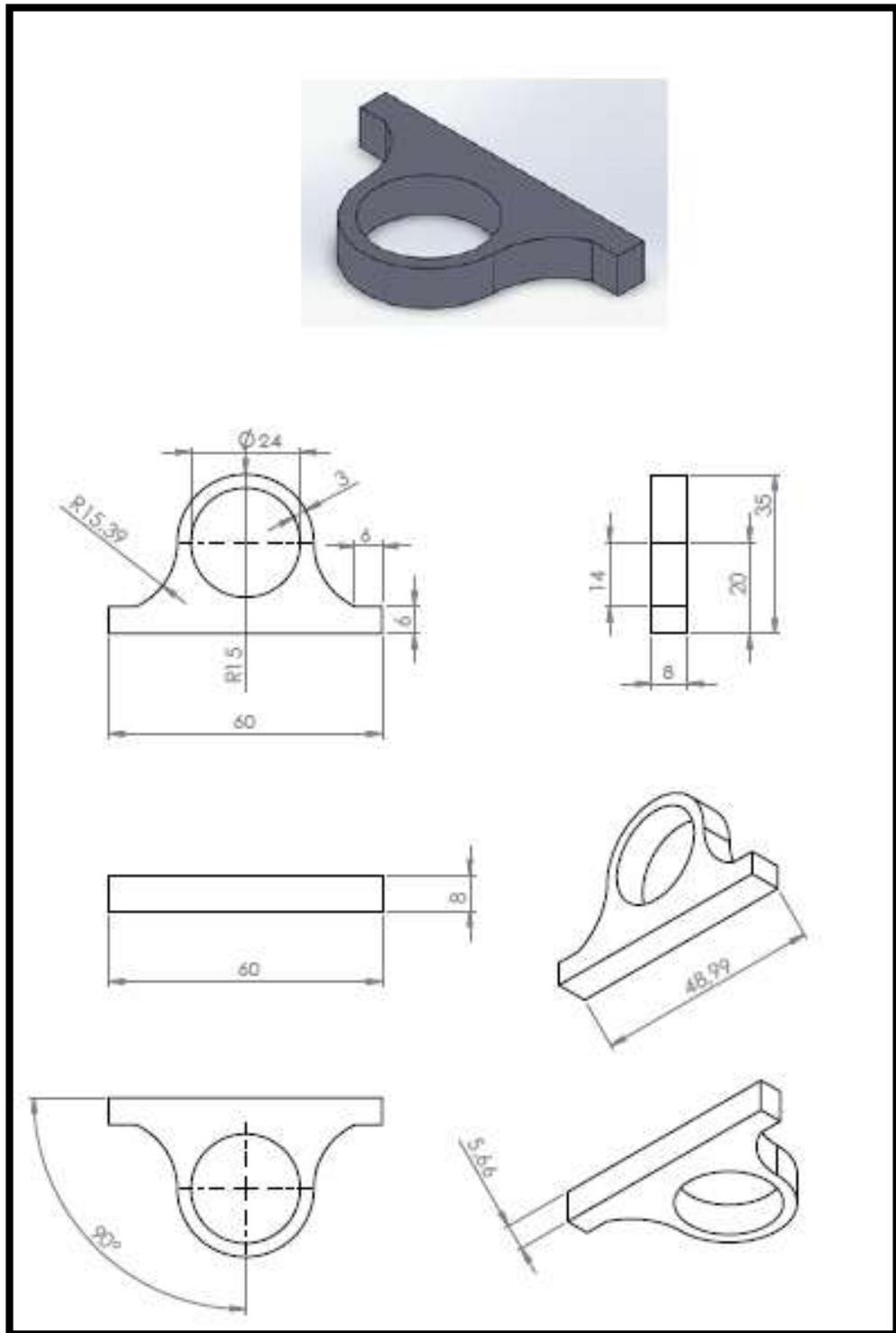
- Chapa delantera C del carro Z



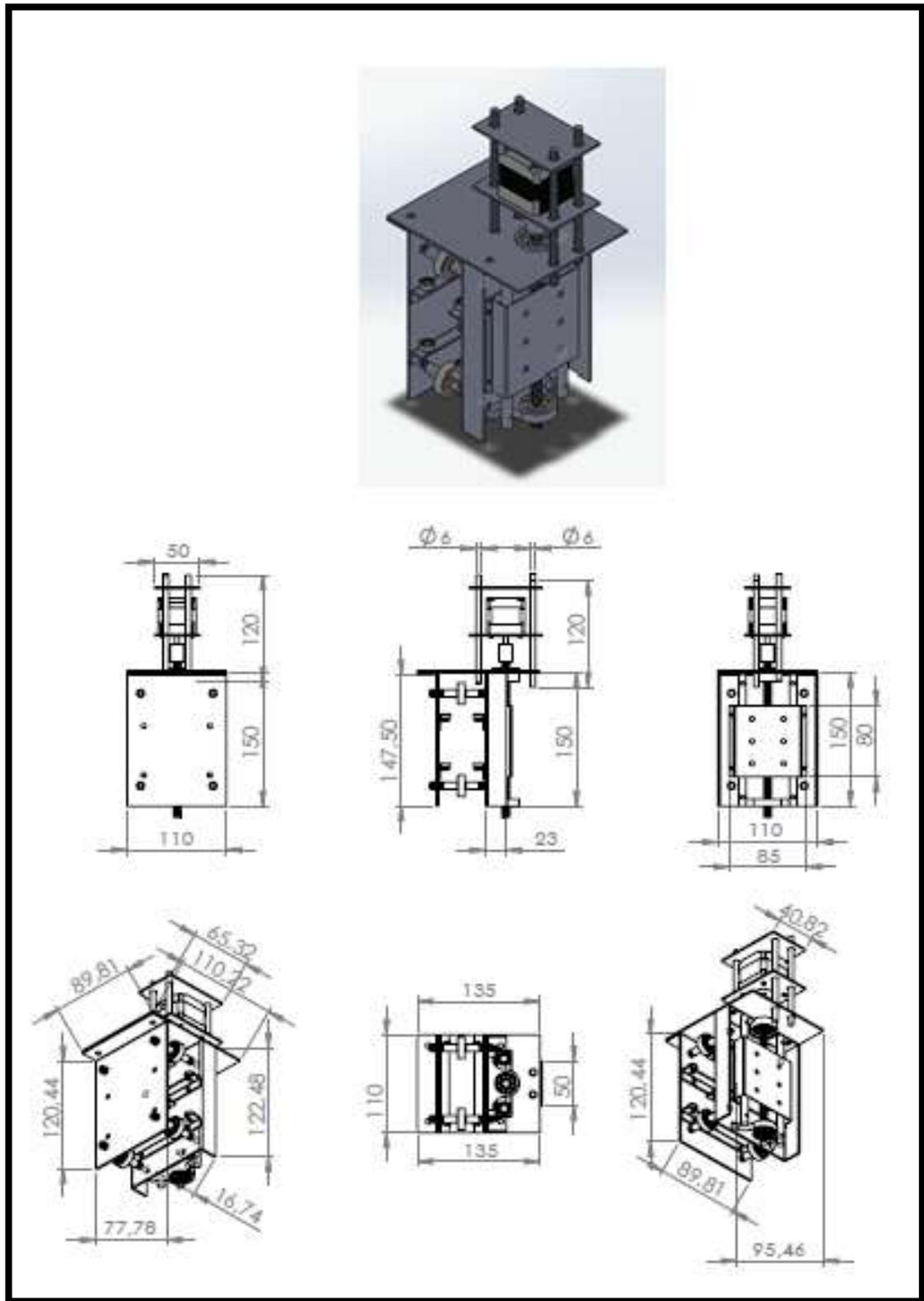
- Chapa trasera del carro Z



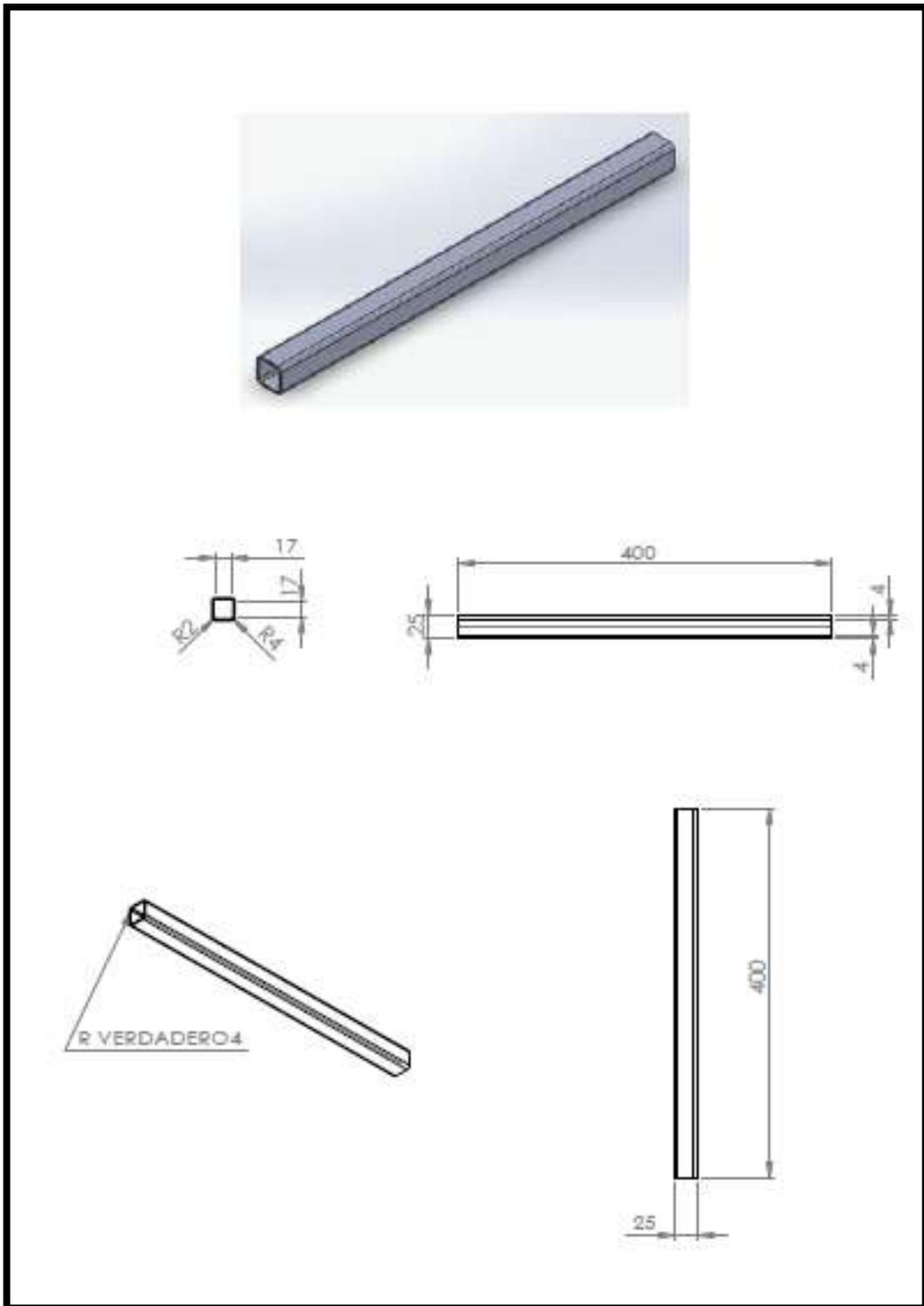
- Chumacera.



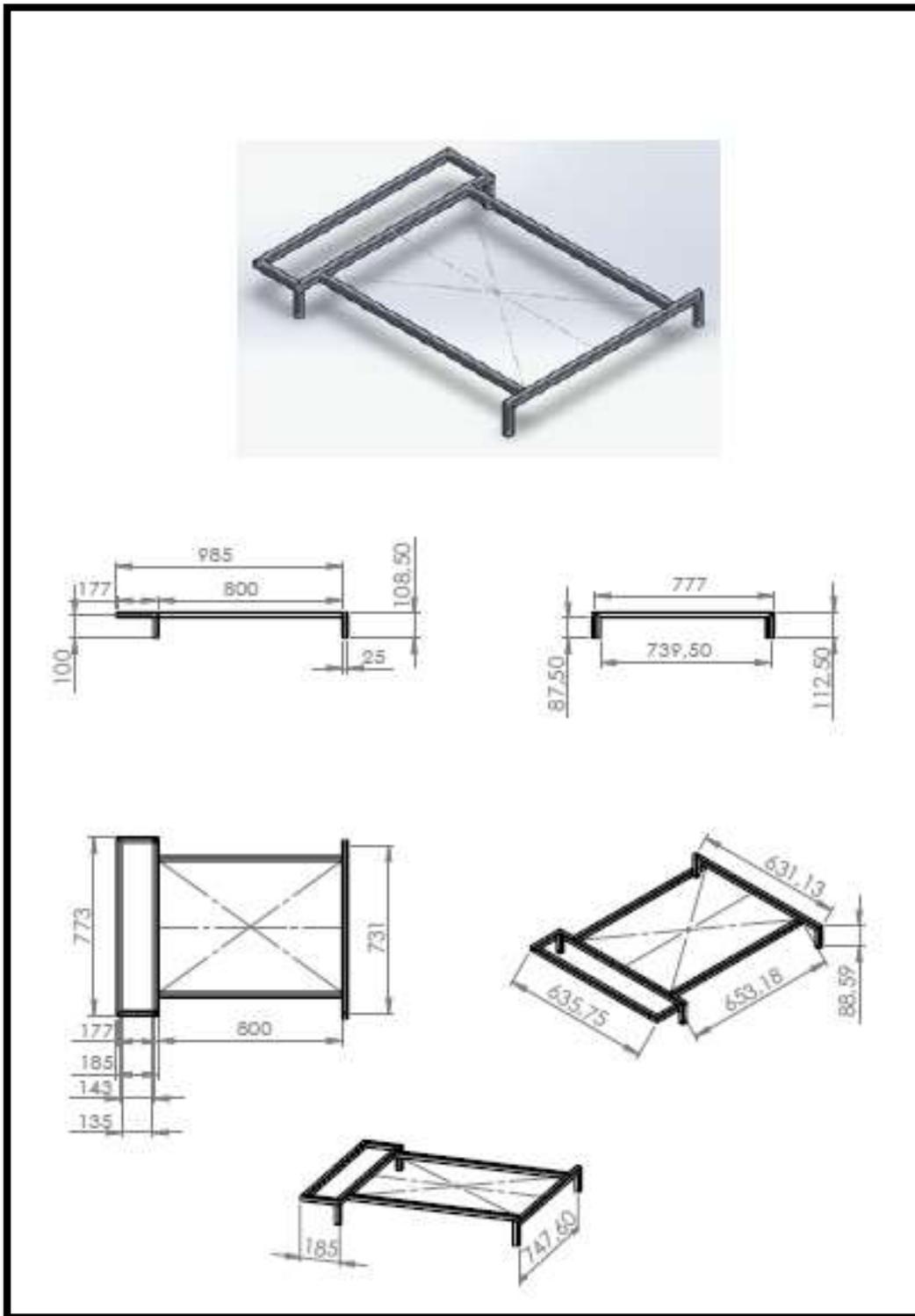
- Eje móvil Z.



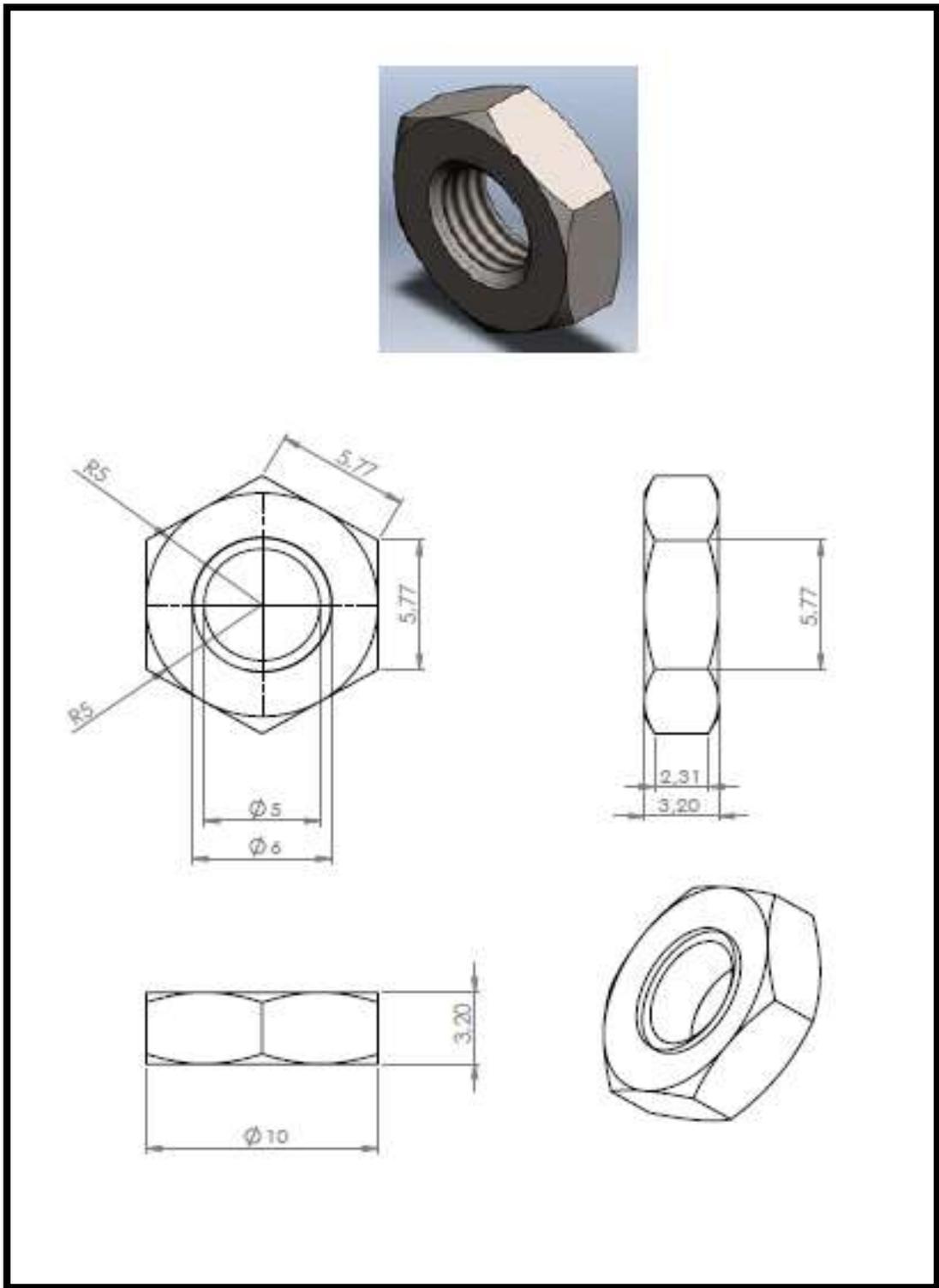
- Tubo cuadrado de Acero inoxidable.



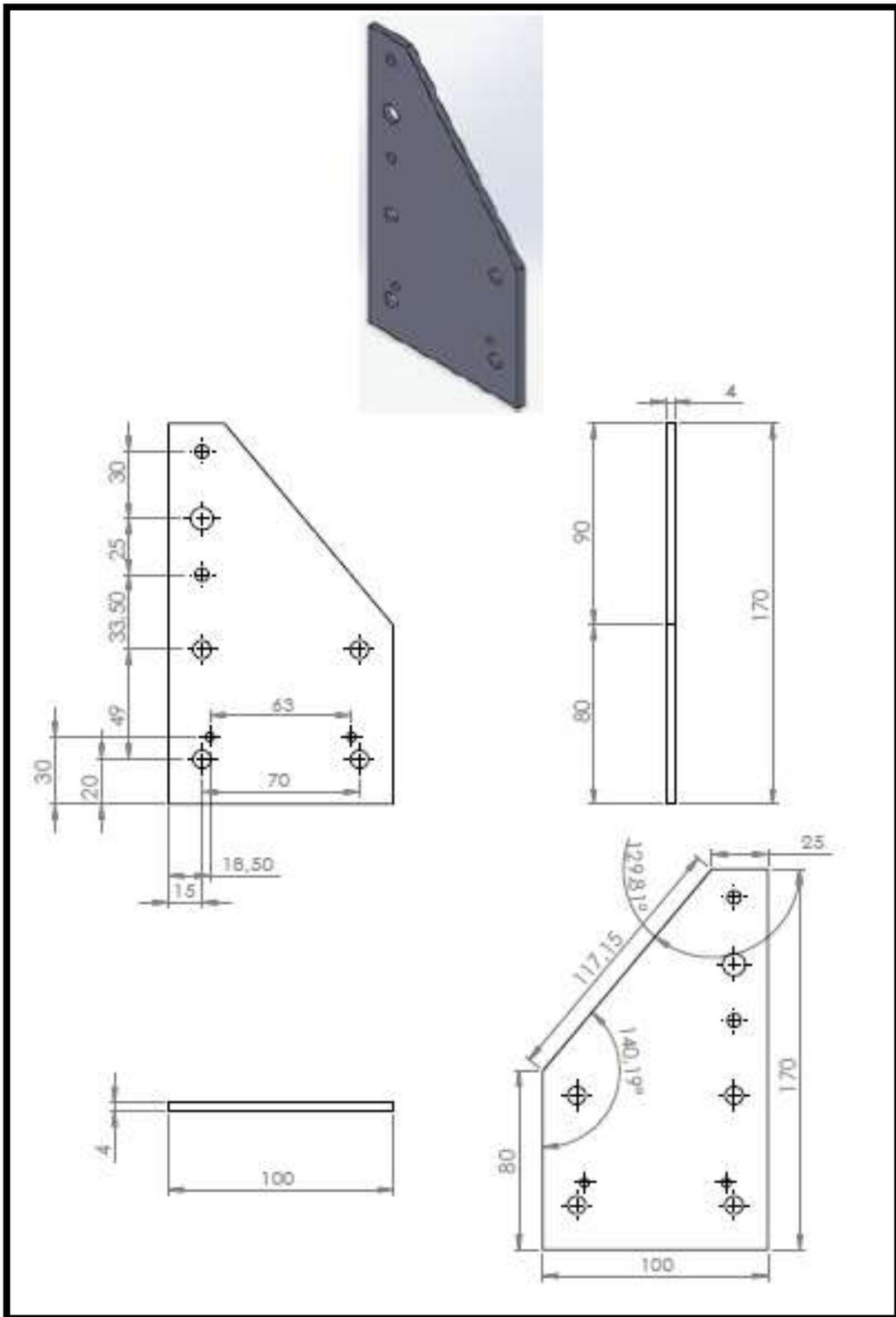
- Estructura de la máquina.



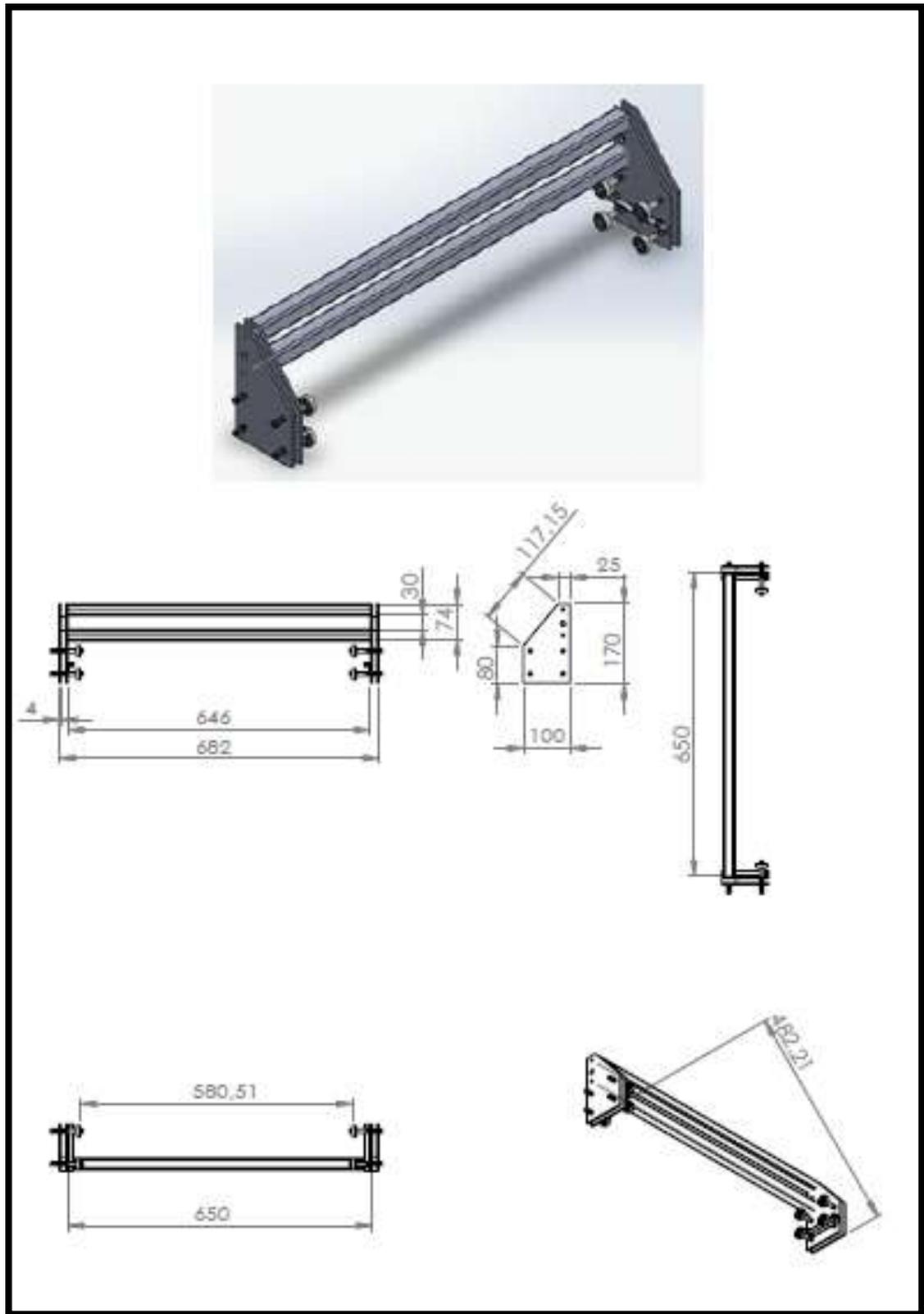
- Tuerca usada en la máquina.



- Soportes laterales del eje X.



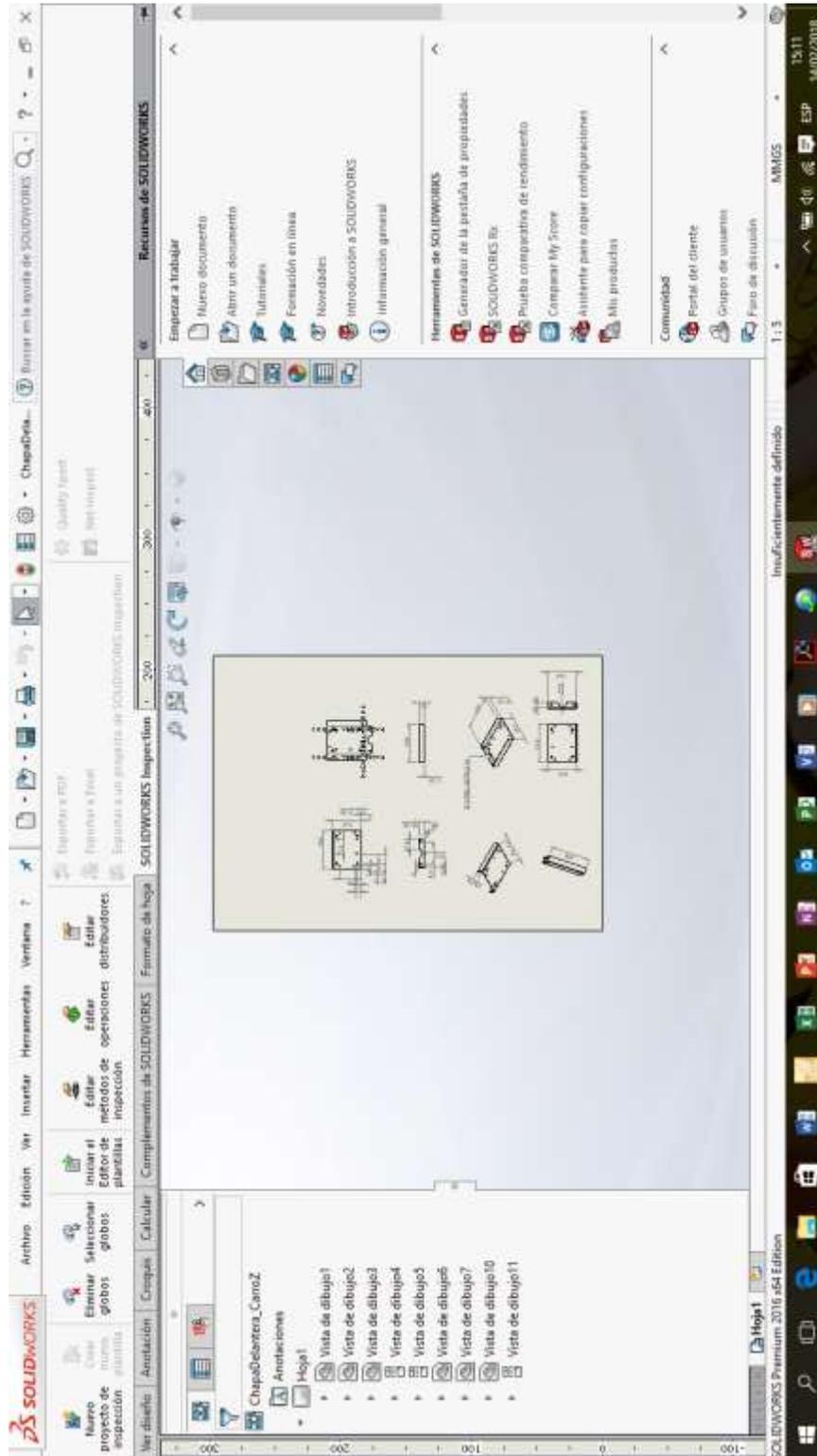
- Ensamblaje del eje móvil X.



- Modelo de la máquina CNC en SolidWorks.

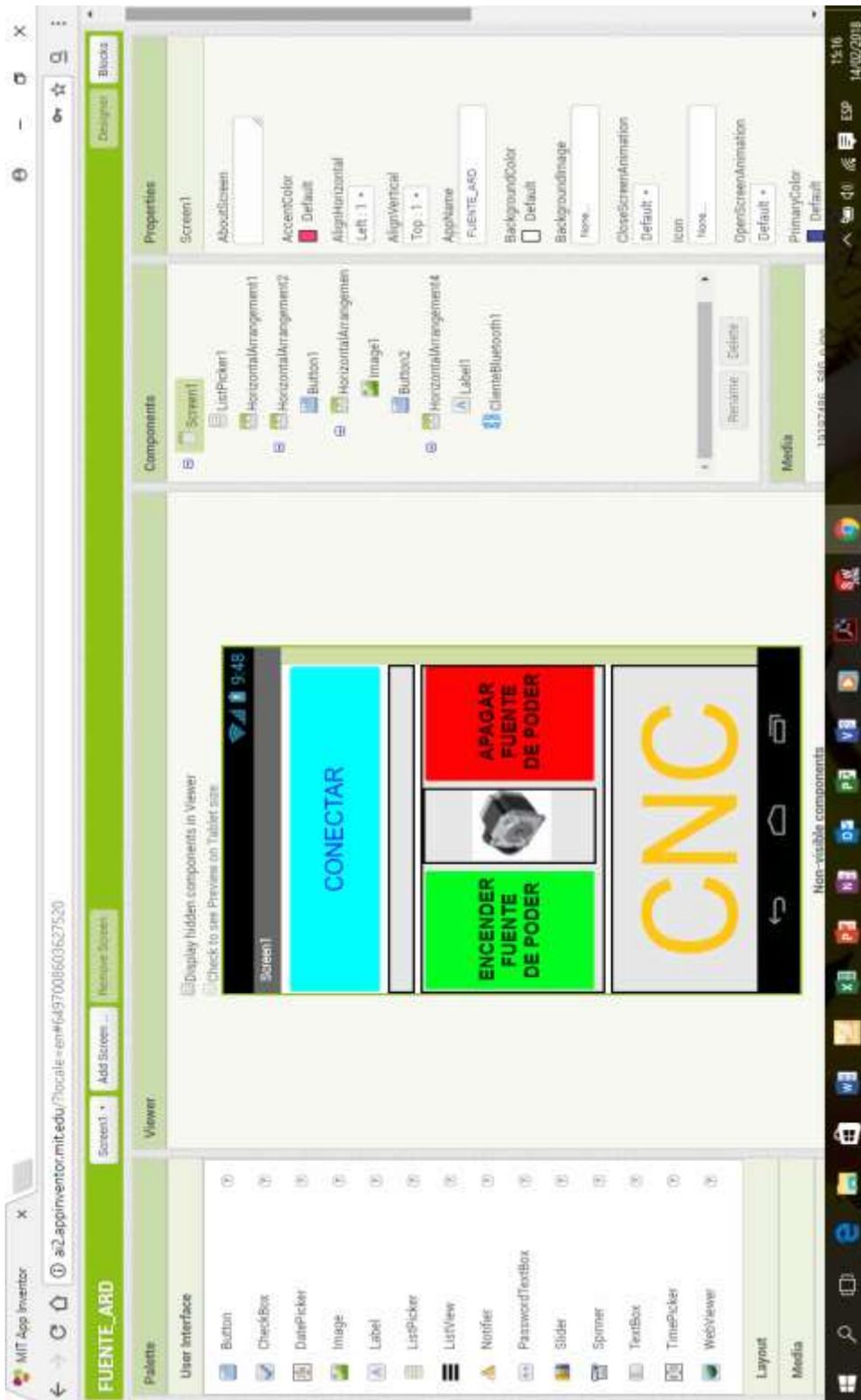


- SolidWorks simulación de piezas mecánicas.



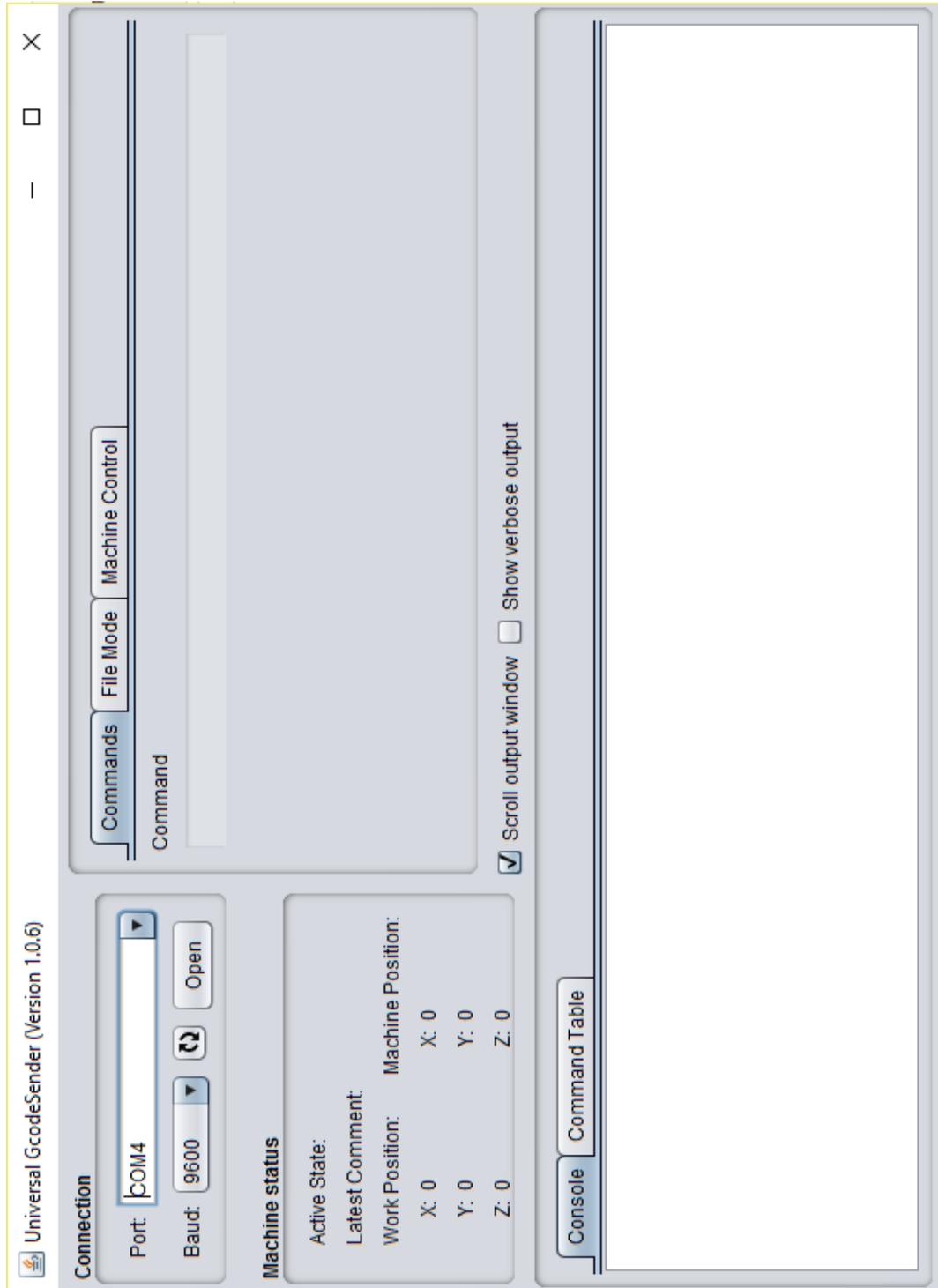
## ANEXO B

### SOFTWARE DE APLICACIÓN DE ALIMENTACIÓN.



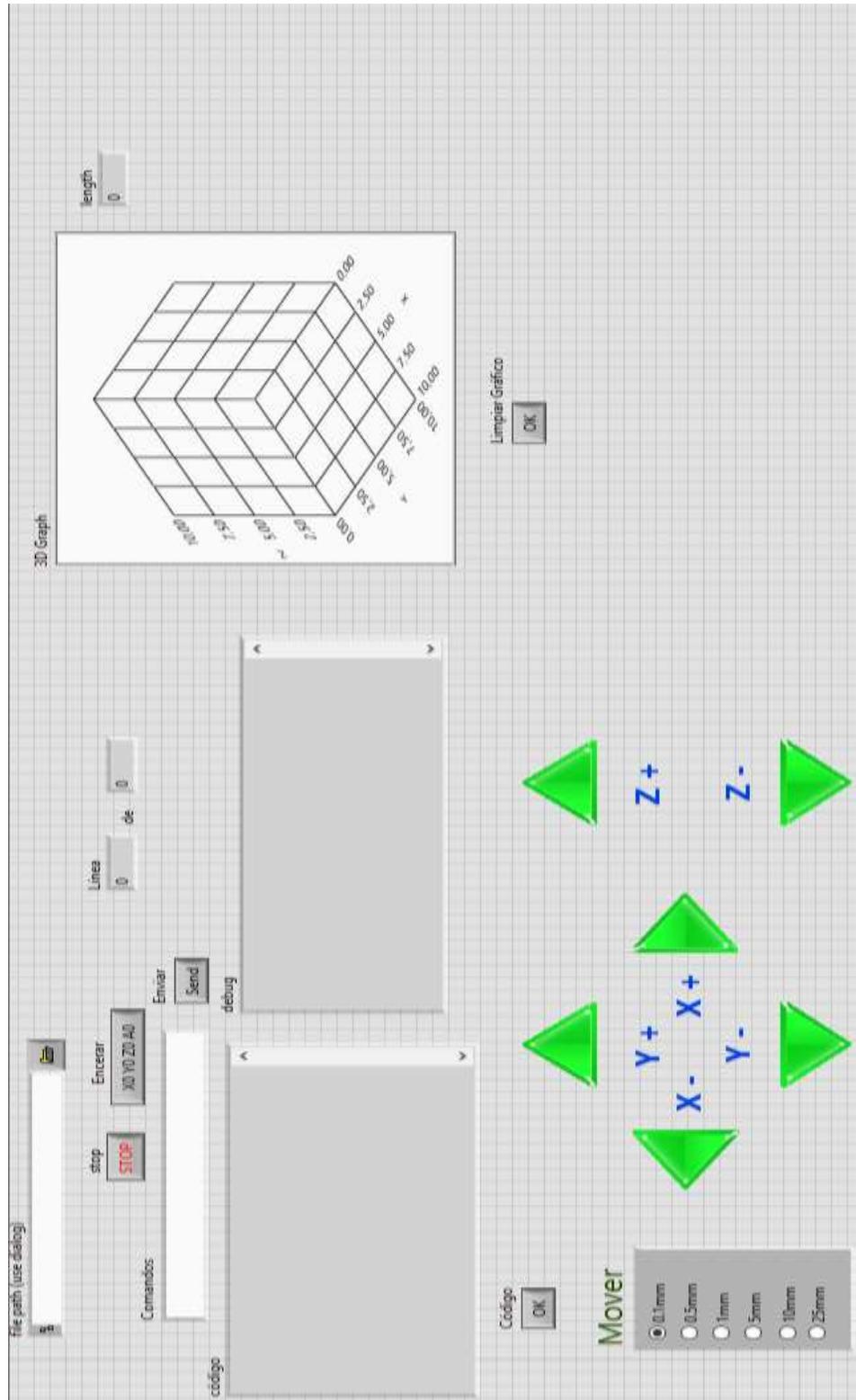
# ANEXO C

## PANEL DE CONTROL UNIVERSAL G-CODE SENDER V1.0.6.



# ANEXO D

## PANEL DE CONTROL LABVIEW.



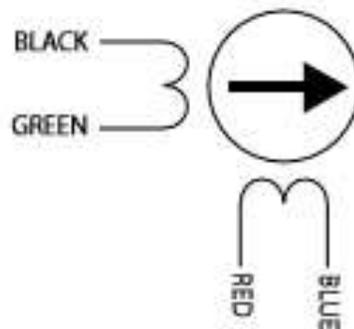
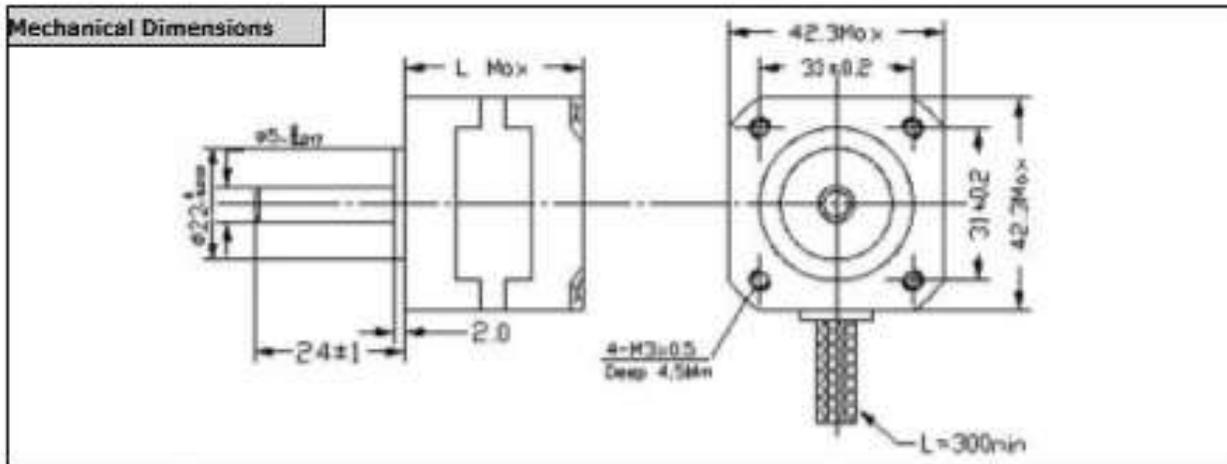
## ANEXO E

### DATA SHEET MOTOR WANTAI MINI STEPPER 42BYGHW811

General Specifications	
Item	Specification
Step Accuracy	5%
Temperature Rise	80°C Max
Ambient Temperature	-20°C~+50°C
Insulation Resistance	100MΩ Min., 500VDC
Dielectric Strength	500 VAC for one minute

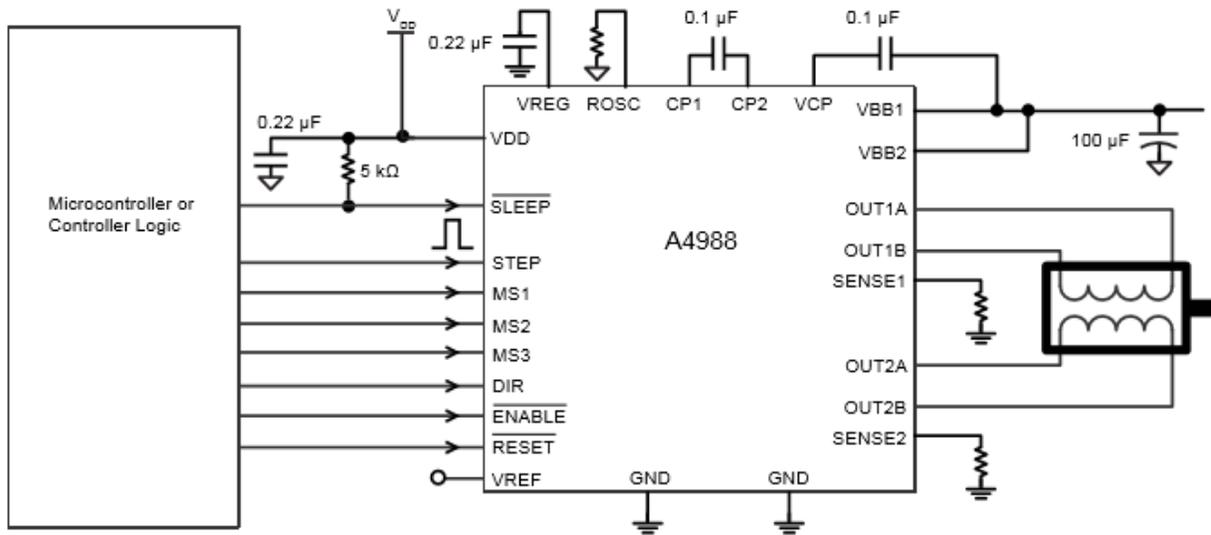


Electrical Specifications											
Model No.	Step Angle	Motor Length	Rated Voltage	Rated Current	Phase Resistance	Phase Inductance	Holding Torque	# of Leads	Rotor Inertia	Detent Torque	Weight
	°	mm	V	A	Ω	mH	g-cm		g-cm <sup>2</sup>	g-cm	kg
42BYGHW811	1.8	48	3.1	2.5	1.25	1.8	4800	4	68	280	0.34



## ANEXO F

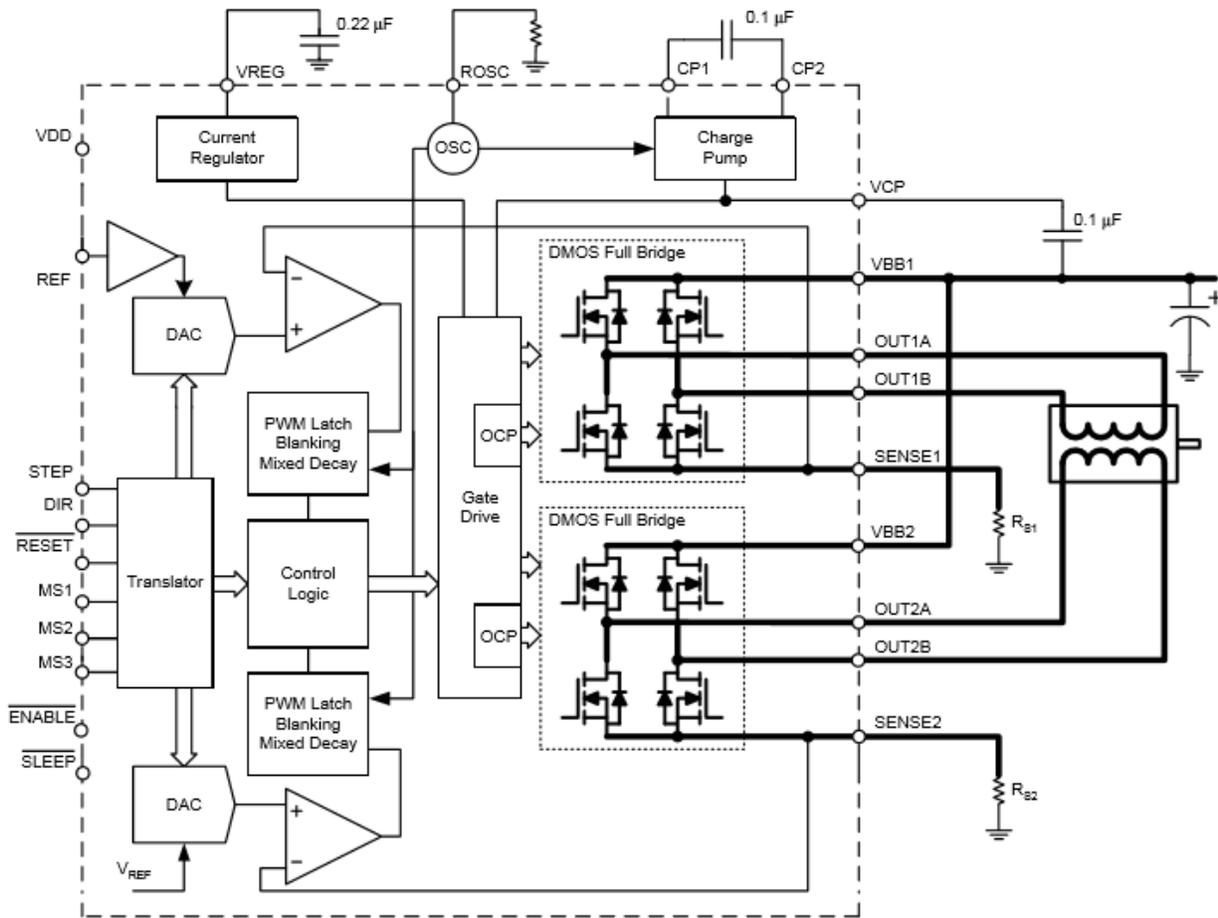
### DATA SHEET DRIVER A4988



### ÍNDICES ABSOLUTOS MÁXIMOS

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	$V_{BB}$		35	V
Output Current	$I_{OUT}$		$\pm 2$	A
Logic Input Voltage	$V_{IN}$		-0.3 to 5.5	V
Logic Supply Voltage	$V_{DD}$		-0.3 to 5.5	V
Motor Outputs Voltage			-2.0 to 37	V
Sense Voltage	$V_{SENSE}$		-0.5 to 0.5	V
Reference Voltage	$V_{REF}$		5.5	V
Operating Ambient Temperature	$T_A$	Range S	-20 to 85	°C
Maximum Junction	$T_J(max)$		150	°C
Storage Temperature	$T_{stg}$		-55 to 150	°C

## DIAGRAMA DE BLOQUE FUNCIONAL



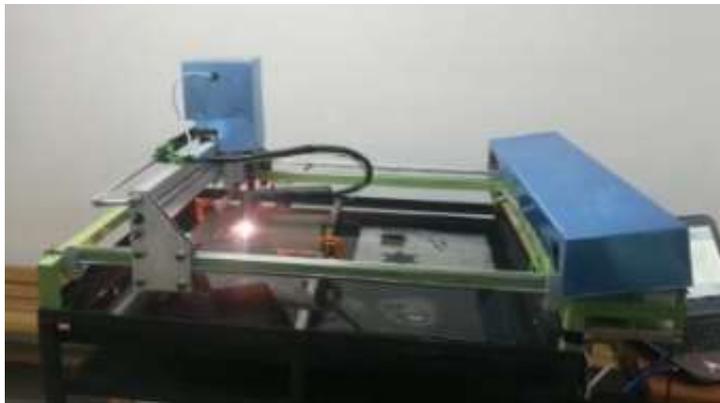
## ANEXO H

- En las figuras se puede observar el montaje del plasma en la mesa de CNC.



- Implementación de la antorcha de corte en la máquina CNC.





## DECLARACIÓN Y AUTORIZACIÓN

Yo, Luis Hernán Herrera Herrera, CI 050298404-0 autor del trabajo de graduación:

**Implementar un sistema de manejo lineal para una mesa de control numérico computarizado, previo a la obtención del título de Ingeniero en Electrónica digital y Telecomunicaciones en la UNIVERSIDAD TECNOLÓGICA ISRAEL.**

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de difundir el respectivo trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de graduación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, marzo de 2018

**Atentamente.**



**Luis Hernán Herrera Herrera.**  
C.I. 050298404-0

id	Modo de entrega	Nombre de tarea	Duración	Comienza	Fin	Av
1	MS	PROYECTO PIC	103 días	mar 21/4/17	jue 31/8/17	
2	MS	Definición del Plan de Proyecto de Tesis	18,5 días	mar 21/4/17	vie 5/5/17	
3	MS	Investigar el tema de proyecto a investigar	6 días	mar 21/4/17	mar 18/4/17	
4	MS	Elaboración del Documento del Proyecto de Tesis	del 2,5 días	mié 19/4/17	vie 21/4/17	3
11	MS	Rectificación y corrección del Plan	4 días	lun 24/4/17	vie 28/4/17	4
12	MS	Entrega de documentos	1 día	vie 28/4/17	lun 1/5/17	11
13	MS	Elaboración del cronograma de tareas	4 días	lun 1/5/17	vie 5/5/17	12
14	MS	FIN	8 horas	vie 5/5/17	vie 5/5/17	13
15	MS	Planificación del Diseño	19,75 días	mar 9/5/17	lun 5/6/17	2
16	MS	Recopilación de información	13,13 días	mar 9/5/17	vie 26/5/17	
17	MS	Investigar y entender lenguaje de programación que usó el Arduino	5 horas	mar 9/5/17	mar 9/5/17	

Tarea	Resumen estado	Tareas estemas
Definición	Tarea manual	Hito oculto
Hito	no duracion	Fecha límite
Resumen	informe de resumen manual	Progreso
Resumen del proyecto	Resumen manual	Progreso manual
Tarea invisible	solo el contenido	
Hito invisible	solo fin	

id	Nombre de tarea	Duración	Comienza	Fin	Av
1	PROYECTO PIC	103 días	mar 21/4/17	jue 31/8/17	
2	Definición del Plan de Proyecto de Tesis	18,5 días	mar 21/4/17	vie 5/5/17	
3	Investigar el tema de proyecto a investigar	6 días	mar 21/4/17	mar 18/4/17	
4	Elaboración del Documento del Proyecto de Tesis	del 2,5 días	mié 19/4/17	vie 21/4/17	3
11	Rectificación y corrección del Plan	4 días	lun 24/4/17	vie 28/4/17	4
12	Entrega de documentos	1 día	vie 28/4/17	lun 1/5/17	11
13	Elaboración del cronograma de tareas	4 días	lun 1/5/17	vie 5/5/17	12
14	FIN	8 horas	vie 5/5/17	vie 5/5/17	13
15	Planificación del Diseño	19,75 días	mar 9/5/17	lun 5/6/17	2
16	Recopilación de información	13,13 días	mar 9/5/17	vie 26/5/17	
17	Investigar y entender lenguaje de programación que usó el Arduino	5 horas	mar 9/5/17	mar 9/5/17	

Id	Modo de tareas	Nombre de tareas	Duración	Comienzo	Fin	PN
18	1	Investigar cómo controlar motores paso a paso con Arduino	8 horas	mié 10/5/17	mié 10/5/17	
19	1	Investigar que Arduino vamos a usar	2 horas	jue 11/5/17	jue 11/5/17	
20	1	Investigar módulos extras a utilizar con el Arduino	3 horas	vie 12/5/17	vie 12/5/17	
21	1	Investigar que es PWM	2 horas	sáb 13/5/17	sáb 13/5/17	
22	1	Investigar que es Código G	3 horas	dom 14/5/17	dom 14/5/17	
23	1	Investigar catálogo de diferentes tipos de plasma	8 horas	lun 22/5/17	lun 22/5/17	
24	1	Investigar especificaciones de diferentes motores paso a paso	3 horas	mar 23/5/17	mar 23/5/17	
25	1	Investigar las tablas de verdad que dan movimiento a los motores paso a paso	1 hora	mié 24/5/17	mié 24/5/17	

Tarea	Resumen rascayo	Tareas conexas
Swirlin	Tarea manual	Moto escorio
Hilo	solo dibujo	Pedra lente
Resumen	Informe de resumen manual	Programa
Resumen del proyecto	Resumen manual	Proyecto manual
Tareas conectadas	solo el conector	
Hilo rascayo	solo fin	

Id	Modo de trabajo	Nombre de tarea	Duración	Comienzo	Fin	Fin	
26	🔍	Investigar tipo de fuentes a utilizar 5V, 12V, 48V, 110V diferentes Amperios FIN	1 hora	vie 26/5/17	vie 26/5/17		
27	🔍	FIN	0 horas				
28	🔍	Elegir materiales a usar	3,63 días	vie 26/5/17	mié 31/5/17		
29	🔍	Arduino 1 y Arduino Mega	2 horas	vie 26/5/17	vie 26/5/17		
30	🔍	Plasma PowerMas30 AIR	4 horas	sáb 27/5/17	sáb 27/5/17		
31	🔍	Motor paso a paso Ventai Mini Stepper	4 horas	dom 28/5/17	dom 28/5/17		
32	🔍	Diferente espesor del Aluminio usado en la Mesa	6 horas	lun 29/5/17	lun 29/5/17		
33	🔍	Diferente espesor en los Angulos que soportaran la Mesa	8 horas	lun 29/5/17	lun 29/5/17		
34	🔍	Selección de sensores	5 horas	mié 31/5/17	mié 31/5/17		
35	🔍	FIN	0 horas				
36	🔍	Pruebas de Funcionamiento	0,75 días	sáb 3/6/17	lun 5/6/17		
<p>Proyecto: Cronograma_Luis_He Fecha: lun 19/2/18</p>							
<p>Tarea</p> <p>Descripción</p> <p>Hito</p> <p>Recursos</p> <p>Estado del proyecto</p> <p>Tarea inactiva</p> <p>Hito inactivo</p>			<p>Recurso inactivo</p> <p>Tarea manual</p> <p>solo datos</p> <p>Informe de recursos manual</p> <p>Recurso manual</p> <p>solo el recurso</p> <p>solo fin</p>			<p>Tarea extendida</p> <p>Hito extendido</p> <p>Tarea libre</p> <p>Progreso</p> <p>Progreso manual</p>	<p>0</p> <p>+</p> <p>+</p> <p>+</p> <p>+</p> <p>+</p>

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Pre
37	0	Verificar el funcionamiento del plasma al 100%	6 horas	sáb 3/5/17	sáb 3/5/17	Estudiante
38	0	Verificar el funcionamiento de los Argónes	6 horas	dom 4/5/17	dom 4/5/17	Estudiante
39	0	Verificar el funcionamiento perfecto del torque y rpm de los motores	6 horas	lun 5/5/17	lun 5/5/17	Estudiante
40	0	FIN	0 horas			
41	0	Diseño del Sistema	38 días	mar 6/5/17	jue 29/5/17 15	Estudiante
42	0	Diseño Mecánico del Prototipo	3 días	mar 6/5/17	jue 8/5/17	Estudiante
43	0	Diseño de la Interfaz de operación de la multibherramienta	2 días	vie 9/5/17	dom 11/5/17	Estudiante
44	0	Diseño control de motores	2 días	lun 12/5/17	mar 13/5/17	Estudiante
45	0	Configuración de motores por eje	3 días	mié 14/5/17	vie 15/5/17	Estudiante
46	0	Configuración del controlador electrónico	2 días	sáb 17/5/17	dom 18/5/17	Estudiante

Tarea	Resumen ejecutivo	Tareas internas
División	Tarea principal	Hdo externo
Nota	solo duración	Fecha límite
Resumen	Informe de resumen manual	Progress
Resumen del proyecto	Resumen manual	Avance manual
Tarea inactiva	modo al completar	
Nota Inactive	modo No	

Id.	Módulo de trabajo	Nombre de tareas	Duración	Comienzo	Fin	Orn
47	✓	Configuración de sensores	3 días	dom 18/6/17	mar 20/6/17	
48	✓	Diseño del control de posición de la herramienta en el eje Z, considerando las señales de sensores y actuadores.	3 días	jue 22/6/17	lun 26/6/17	
49	✓	Diseño interfaz Hombre Máquina	3 días	mar 27/6/17	jue 29/6/17	
50	✓	FIN	0 horas			
51	✓	Implementación del Diseño	22 días	vie 30/6/17	lun 31/7/17	41
52	✓	Ensamblaje del prototipo	2 días	vie 30/6/17	lun 3/7/17	
53	✓	Montaje de los mecanismos	2 días	vie 7/7/17	sáb 8/7/17	
54	✓	Montaje de motor en eje Z	2 días	dom 9/7/17	lun 10/7/17	
55	✓	Montaje y prueba del control del motor	1 día	vie 14/7/17	vie 14/7/17	
56	✓	Instalación de los sensores para el control de posicionamiento de eje Z en el Router	2 días	sáb 15/7/17	lun 17/7/17	

Tarea	Resumen acciones	Tareas asociadas
Diseño	Tarea manual	Hilo externo
Hilo	solo ejecución	Fecha límite
Resumen	Informe de resumen manual	Progreso
Resumen de proyecto	Resumen manual	Progreso manual
Tarea FIN/OT	solo el contenido	
Hilo interno	solo fin	

Id	Modo de tareas	Nombre de tarea	Duración	Comienzo	Fin	Fin
57	☑	Calibración de los parámetros de los sensores para el control de posicionamiento	1 día	lun 17/7/17	lun 17/7/17	17-4-2017 mar. abr. may. jun. jul. ago. sep. oct. nov.
58	☑	Análisis y depuración de fallos	1 día	vie 21/7/17	vie 21/7/17	Estudiante
59	☑	Implementación del la interfaz de operación de la multiherramienta	2 días	sáb 22/7/17	dom 23/7/17	Estudiante
60	☑	Montaje de la(s) herramienta(s) de trabajo en el portaherramientas (eje 2).	2 días	dom 23/7/17	lun 24/7/17	Estudiante
61	☑	Enlace entre la operación de la(s) herramienta(s) de trabajo del Router y la interfaz de operación.	2 días	vie 28/7/17	dom 30/7/17	Estudiante
62	☑	Recopilación de datos de pruebas ejecutadas para levantamientos de desvíos y calibración para puesta en marcha del sistema.	2 días	dom 30/7/17	lun 31/7/17	Estudiante
63	☑	FIN	0 horas			30/6

Tarea	Resumen trabajo	Tareas asociadas
División	Tarea mensual	Hábito semanal
Hito	solo durante	Fecha límite
Recurso	Informe de resumen mensual	Progreso
Recurso del proyecto	Resumen mensual	Progreso semanal
Tarea asociada	solo el contenido	
Hito asociado	solo de	

Id	Modo de trabajo	Nombre de tarea	Duración	Comienzo	Fin	Fin
64	MC	Pruebas de funcionamiento	19 días	vie 4/8/17	mié 30/8/17	51
65	MC	Pruebas mecánicas del eje Z del prototipo	2 días	vie 4/8/17	sáb 5/8/17	
66	MC	Pruebas de los mecanismos	2 días	dom 6/8/17	lun 7/8/17	
67	MC	Pruebas eléctricas y mecánicas del ensamblaje	2 días	vie 11/8/17	sáb 12/8/17	
68	MC	Prueba de operación de los sensores para control de posicionamiento de eje Z en el Router	2 días	dom 13/8/17	lun 14/8/17	
69	MC	Pruebas del control de posición de la herramienta en el eje Z	2 días	vie 18/8/17	lun 21/8/17	
70	MC	Pruebas del enlace entre la operación de la herramienta de trabajo del Router y la interfaz de operación	2 días	vie 25/8/17	sáb 26/8/17	

Tarea	Resumen Inicial	Resumen Intermedio	Resumen Final
Oración	Resumen Inicial	Tarea manual	Tarea intermedia
Inicio	Resumen Inicial	Tarea manual	WTO externo
Resumen	Resumen Inicial	solo duración	Fecha inicio
Resumen del proyecto	Resumen Inicial	Informe de resumen manual	Progreso
Tarea inactiva	Resumen Inicial	Resumen manual	Progreso manual
Finalización	Resumen Inicial	solo el comienzo	C
	Resumen Inicial	solo fin	3

Id	Modo de trabajo	Nombre de tarea	Duración	Comienzo	Fin	Pre																												
71	0	Prueba de operación del control de posición de la herramienta (plasma) en placas de acero de diferente espesor (de 0.6 a 8 mm)	2 días	dom 27/8/17	lun 28/8/17																													
72	0	Prueba de operación del control de posición de la herramienta (dril) en placas de madera y bagueta.	3 días	lun 28/8/17	mié 30/8/17																													
73	0	FIN	0 horas																															
74	0	Elaboración del Documento Técnico	23 días	mar 1/8/17	jue 31/8/17 04																													
75	0	Redactar Fase Inicial	9.5 días	mié 2/8/17	mar 15/8/17																													
76	0	Portada	1 hora	mié 2/8/17	mié 2/8/17																													
77	0	Dedicatorias	1 hora	mié 2/8/17	mié 2/8/17																													
78	0	Agradecimientos	1 hora	jue 3/8/17	jue 3/8/17																													
79	0	Índice	5 horas	mar 8/8/17	mar 8/8/17																													
80	0	Listas de tablas y figuras	6 horas	mié 9/8/17	mié 9/8/17																													
81	0	Resumen	3 horas	jue 10/8/17	jue 10/8/17																													
<p>Proyecto: Cronograma_Luz_He Fecha: lun 19/2/18</p>																																		
<table border="0"> <tr> <td>Tarea</td> <td>Resumen ejecutivo</td> <td>Resumen ejecutivo</td> <td>Tareas externas</td> </tr> <tr> <td>Origen</td> <td>Tarea manual</td> <td>Tarea manual</td> <td>Hijo externo</td> </tr> <tr> <td>Hijo</td> <td>solo duración</td> <td>solo duración</td> <td>Fecha límite</td> </tr> <tr> <td>Resumen</td> <td>informe de resumen manual</td> <td>informe de resumen manual</td> <td>Programa</td> </tr> <tr> <td>Resumen del proyecto</td> <td>Resumen manual</td> <td>Resumen manual</td> <td>Proyecto manual</td> </tr> <tr> <td>Tareas inactive</td> <td>solo el comienzo</td> <td>solo el comienzo</td> <td></td> </tr> <tr> <td>Hijo inactive</td> <td>solo fin</td> <td>solo fin</td> <td></td> </tr> </table>							Tarea	Resumen ejecutivo	Resumen ejecutivo	Tareas externas	Origen	Tarea manual	Tarea manual	Hijo externo	Hijo	solo duración	solo duración	Fecha límite	Resumen	informe de resumen manual	informe de resumen manual	Programa	Resumen del proyecto	Resumen manual	Resumen manual	Proyecto manual	Tareas inactive	solo el comienzo	solo el comienzo		Hijo inactive	solo fin	solo fin	
Tarea	Resumen ejecutivo	Resumen ejecutivo	Tareas externas																															
Origen	Tarea manual	Tarea manual	Hijo externo																															
Hijo	solo duración	solo duración	Fecha límite																															
Resumen	informe de resumen manual	informe de resumen manual	Programa																															
Resumen del proyecto	Resumen manual	Resumen manual	Proyecto manual																															
Tareas inactive	solo el comienzo	solo el comienzo																																
Hijo inactive	solo fin	solo fin																																



Id.	Modo de Zonas	Nombre de tareas	Duración	Comienzo	Fin	Fin
96	M	Redactar Apéndice	22 días	mar 1/8/17	mié 30/8/17	
97	P	Listas de abreviaturas o siglas ordenadas alfabéticamente	3 horas	mar 1/8/17	mar 2/8/17	Estudiante; Tutor
98	P	Datos concretos sobre instrumentos de investigación	6 horas	mar 8/8/17	mar 8/8/17	Estudiante; Tutor
99	P	Dibujos, diagramas, fotos e cualquier otro material similar	7 horas	mar 15/8/17	mar 15/8/17	Estudiante; Tutor
100	P	Gráficas, tablas y cuadros estadísticos	2 días	mar 29/8/17	mié 30/8/17	Estudiante; Tutor
101	P	FIN	0 horas			1/8
102	P					
103	P					

Tarea	Resumen interactivo	Tareas asignadas
Zonas	Tarea manual	Hito asignado
Hito	solo detección	Fecha límite
Asignación	informe de resumen mensual	Programa
Resumen del proyecto	Resumen manual	Programa mensual
Tareas inactivas	solo el colaborador	
Hito inactivo	solo hito	

Proyecto: Cronograma\_Luis\_16  
Fecha: lun 19/2/16

# Plagiarism Checker X Originality Report



Plagiarism Quantity: 9% Duplicate



Date	domingo, Febrero 18, 2018
Words	1299 Plagiarized Words / Total 16028 Words
Sources	More than 485 Sources Identified.
Remarks	Low Plagiarism Detected - Your Document needs Optional Improvement.

*Handwritten signature*  
19/02/2018