



**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Ciudad



**INSTITUTO TECNOLÓGICO DE CD. GUZMÁN**

**TESIS**

TEMA:

**PLOTTER XYZ PARA IMPRESIÓN DE CARACTERES EN PIEZAS  
METÁLICAS PARA LA EMPRESA IP INDUSTRIAL SA DE CV**

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA ELECTRÓNICA**

PRESENTA:

**ING. CARLOS ALBERTO RUIZ LÓPEZ**

DIRECTOR DE TESIS:

**DR. SERGIO SANDOVAL PÉREZ**

CODIRECCIÓN DE TESIS:

**DR. RUBÉN RUELAS LEPE**

CD. GUZMÁN JALISCO, MÉXICO, AGOSTO DE 2019

## Dedicatoria:

A mis padres, el Sr Pablo Ruiz y a la Sra. Teresa López Mejía por su apoyo incondicional en todas las etapas de mi vida y el respaldo que me han brindado a lo largo de mis estudios.

A mi familia, amigos y compañeros que me apoyado y motivado para seguir superándome día con día.

A mis profesores que han compartido sus conocimientos conmigo, sin su apoyo y guía no sería la persona que hoy soy.

## Agradecimientos:

Gracias a Dios por haberme dado la capacidad de terminar este proyecto satisfactoriamente a lado de las personas que supieron ayudarme a lograr mis objetivos.

Al Instituto Tecnológico de Ciudad Guzmán por brindarme la confianza y aceptarme en la maestría de Ingeniería en Electrónica.

Al Consejo Nacional de Ciencia y Tecnología por brindarme el apoyo económico con el cual pude solventar los gastos escolares.

A mi director de Tesis Dr. Sergio Sandoval Pérez por su apoyo y guía durante la realización de este trabajo.

A mi codirector de tesis Dr. Rubén Ruelas Lepe por su apoyo para que la vinculación de este proyecto fuera posible.

A cada uno de los profesores de posgrado por su dedicación, aportaciones y conocimientos compartidos para que este proyecto terminara satisfactoriamente.

## **Resumen**

En este trabajo se presenta el desarrollo e implementación de un sistema de control con interfaz de usuario (GUI por su nombre en inglés, Graphical User Interface) para una máquina de micropercusión, con la que se planea imprimir caracteres en piezas metálicas y plásticos.

La redacción realizada por micropercusión se define por una sucesión de puntos. La técnica de marcado por micropercusión es una solución económica y sin consumibles para un marcado permanente. Los sistemas de micropercusión son particularmente adecuados para el marcado permanente de los metales y piezas de plástico, independientemente del entorno de trabajo o de propiedades de la superficie.

La GUI se desarrolla en el IDE Visual Studio con el lenguaje de programación C#. En ella se configuran las propiedades y características de la etiqueta a marcar, posteriormente la GUI codifica la marca a un lenguaje que el sistema de control puede interpretar, así mismo controla el inicio de marcado, pausado y detención.

Para el sistema de control se implementa un microcontrolador Arduino® UNO con la librería Open Source GRBL. El actuador es un punzón neumático con una punta de tungsteno.

Palabras clave: Máquina de micropercusión, Interfaz de usuario, Marca, Sistema de control, Visual Studio, C#, Arduino, GRBL, Punzón neumático.

# Abstract

This project presents the development and implementation of a control system with graphical user interface (GUI) for a micropercussion machine with which it is planned to print characters on metal and plastic parts.

The wording made by micropercussion is defined by a succession of points. The micropercussion marking technique is an economical and without consumables solution for permanent marking. Micropercussion systems are particularly suitable for permanent marking of metals and plastic parts, regardless of work environment or surface properties.

The GUI is developed in the Visual Studio IDE with the programming language C#. It sets the properties and characteristics of the label, then the GUI encodes the mark to a language that the control system can interpret, also the GUI controls the start, pause and detention of marking.

An Arduino® UNO microcontroller with the Open Source GRBL library is implemented for the control system. The actuator is a pneumatic punch with a tungsten peen.

Keywords: Micropercussion machine, User interface, Mark, Control system, Visual Studio, C#, Arduino, GRBL, Pneumatic punch.

## Índice general

<b>1. Introducción</b>	<b>2</b>
1.1 Definición del problema	2
1.2 Exposición del problema	2
1.3 Justificación del problema	2
1.4 Objetivos	3
1.4.1 Objetivo general	3
1.4.2 Objetivos específicos	3
1.5 Hipótesis	3
<b>2. Modelo físico del plotter XYZ para impresión de caracteres</b>	<b>6</b>
2.1 Sistema de movimiento en el eje "X"	6
2.2 Sistema de movimiento en eje "Y"	7
2.3 Sistema de movimiento en dirección "Z"	8
2.4 Montaje final de las estructuras	10
2.5 Vistas del ensamblaje final	11
<b>3 Fundamentos teóricos</b>	<b>19</b>
3.1 Introducción a las máquinas de grabado con la técnica de micropercusión	19
3.2 Control numérico computarizado.	20
3.2.1 Principio de funcionamiento	20
3.2.2 Programación en el control numérico	21
3.2.2.1 Programación manual	21
3.2.2.2 Programación automática	22
3.2.3 Caracteres más usados.	22
3.2.4 Plano cartesiano	26
3.3 Electrónica y control	27
3.3.1 Microcontrolador	27
3.3.2 Arduino UNO	28
3.3.3 Firmware	29
3.3.4 Controlador digital para motores a pasos DM422C (Fully digital stepping drive)	30

3.3.5 Motores a pasos _____	30
3.3.6 Sensor final de carrero (Limit Switch) _____	31
3.4 Interfaz gráfica de usuario _____	32
3.4.1 Visual Studio IDE _____	33
3.4.2 C# _____	34
3.4.3 C# y los objetos _____	34
3.5 Elementos mecánicos _____	35
3.5.1 Estructura _____	35
3.5.2 Sistema de desplazamiento _____	36
3.5.3 Transmisión _____	36
<b>4 Técnicas de programación _____</b>	<b>39</b>
4.1 Firmware del proyecto _____	39
4.1.1 Instalación de GRBL versión 1.1 _____	40
4.2 Interfaz de usuario (GUI) _____	47
4.2.1 Pantalla principal _____	47
4.2.2 Módulo marca _____	47
4.2.3 Módulo control manual _____	49
4.2.4 Módulo imprimir _____	49
4.2.5 Modulo conexión _____	50
<b>5 Interfaces hardware _____</b>	<b>52</b>
5.1 PCB, placa controladora _____	52
<b>6 Resultados _____</b>	<b>58</b>
<b>7 Conclusiones _____</b>	<b>65</b>
<b>Referencias bibliográficas _____</b>	<b>67</b>

## Índice de figuras

<b>Figura 2.1:</b> Sistema de movimiento en eje X. ....	7
<b>Figura 2.2:</b> Sistema de movimiento en eje Y. ....	8
<b>Figura 2.3:</b> Sistema de movimiento en eje Z. ....	9
<b>Figura 2.4:</b> Ensamblaje final de la máquina. ....	10
<b>Figura 2.5:</b> Vista frontal. ....	11
<b>Figura 2.6:</b> Vista posterior. ....	12
<b>Figura 2.7:</b> Vista lateral derecha. ....	13
<b>Figura 2.8:</b> Vista lateral izquierda. ....	14
<b>Figura 2.9:</b> Vista superior. ....	15
<b>Figura 2.10:</b> Vista inferior. ....	15
<b>Figura 2.11:</b> Vista en perspectiva. ....	16
<b>Figura 3.1:</b> Ejemplo de microcontroladores. ....	28
<b>Figura 3.2:</b> Fotografía de un Arduino UNO. ....	29
<b>Figura 3.3:</b> Driver DM422C para motores a paso. ....	30
<b>Figura 3.4:</b> Motor a pasos. ....	31
<b>Figura 3.5:</b> Limit switch. ....	32
<b>Figura 3.6:</b> Logotipo Visual Studio. ....	33
<b>Figura 3.7:</b> Sistema de guía Hiwin. ....	36
<b>Figura 3.8:</b> Sistema de transmisión por correa dentada. ....	37
<b>Figura 4.1:</b> Logotipo del firmware GRBL. ....	39
<b>Figura 4.2:</b> Instalación de la librería en el IDE de Arduino. ....	40
<b>Figura 4.3:</b> Apertura del sketch para la instalación de GRBL en Arduino UNO. ....	41
<b>Figura 4.4:</b> Configuración de puerto y tarjeta Arduino. ....	42
<b>Figura 4.5:</b> Carga del firmware GRBL en el Arduino UNO. ....	43
<b>Figura 4.6:</b> Vista software Universal Gcode Sender. ....	44
<b>Figura 4.7:</b> Vista principal de la interfaz de usuario. ....	47
<b>Figura 4.8:</b> Módulos de la aplicación. ....	48
<b>Figura 4.9:</b> Módulo marca de la interfaz. ....	48
<b>Figura 4.10:</b> Módulo control manual. ....	49
<b>Figura 4.11:</b> Módulo imprimir. ....	49
<b>Figura 4.12:</b> Módulo conexión. ....	50
<b>Figura 5.1:</b> Diseño del PCB en Altium. ....	52
<b>Figura 5.2:</b> Top layer del PCB. ....	53
<b>Figura 5.3:</b> Imagen preliminar del ruteado del PCB. ....	54
<b>Figura 5.4:</b> Ruteado del PCB con ayuda de un CNC. ....	54
<b>Figura 5.5:</b> PCB ruteado. ....	55
<b>Figura 5.6:</b> Montaje de los componentes en el PCB. ....	55
<b>Figura 5.7:</b> PCB montado y conectado. ....	56



<b>Figura 6.1:</b> Ensamble final de la máquina.....	58
<b>Figura 6.2:</b> Pantalla principal de la interfaz. ....	59
<b>Figura 6.3:</b> Conexión a la máquina. ....	60
<b>Figura 6.4:</b> Inicializado de la máquina. ....	60
<b>Figura 6.5:</b> Agregado de una etiqueta y su configuración. ....	61
<b>Figura 6.6:</b> Perilla de ajuste de altura o eje Z.....	62
<b>Figura 6.7:</b> Máquina con pieza a marcar. ....	62
<b>Figura 6.8:</b> Manejo de impresión desde la interfaz. ....	63
<b>Figura 6.9:</b> Resultado de la impresión.....	63

## Índice de tablas

<b>Tabla I:</b> Descripción de las funciones de códigos G.....	23
<b>Tabla II:</b> Descripción de las funciones de códigos M. ....	25
<b>Tabla III:</b> Valores de las configuraciones del Software GRBL. ....	44

**CAPÍTULO I**  
**INTRODUCCIÓN**

## **1. Introducción**

Actualmente es de suma importancia la identificación y trazabilidad de los productos a lo largo de todo su proceso de fabricación y cadena de suministros, por ello es que se implementan métodos de marcaje directo de piezas. Resulta perfecto a la hora de encontrar piezas para su mantenimiento o retirada, y puede ser de ayuda en la resolución de la responsabilidad y la garantía. Este proceso se emplea frecuentemente en los sectores aeroespacial y de la automoción para marcas códigos DataMatrix en 2D y alfanuméricos en montajes y piezas individuales.

El principio del marcaje por micropercusión se basa en una deformación en hueco de la materia mediante una herramienta vibrante. Éste crea una sucesión de puntos (impactos) que forman un marcaje directo en la pieza, permanente e infalsificable.

### 1.1 Definición del problema

La empresa IP Industrial, una empresa mexicana dedicada al desarrollo de soluciones integrales e innovadoras para la industria, utilizando las más modernas herramientas de diseño, manufactura y automatización. Necesita el desarrollo de una máquina de micropercusión de bajo costo para satisfacer las necesidades de sus clientes.

## **1.2 Exposición del problema**

Se necesita el desarrollo de un control para una máquina de micropercusión junto con una interfaz de usuario en computadora.

## **1.3 Justificación del problema**

La Empresa IP Industrial necesita una máquina de marcado de micropercusión para satisfacer las necesidades de sus clientes, la solución debe ser de bajo costo, ya que las implementadas actualmente tienen costos muy elevados y es tecnología extranjera, con el desarrollo de la máquina será más fácil aplicarla a los proyectos a medida.

## **1.4 Objetivos**

En esta sección se describen el objetivo general y los objetivos específicos que marcan el rumbo de esta investigación.

### **1.4.1 Objetivo general**

Generar un control e interfaz para máquina de grabado por micropercusión.

### **1.4.2 Objetivos específicos**

- Generar controlador para los motores a pasos de la máquina.
- Desarrollar algoritmos de grabado.
- Desarrollar interfaz de usuario para la máquina.
- Seleccionar elementos de bajo costo y alta precisión.
- Seleccionar el elemento de marcado neumático.

## **1.5 Hipótesis**

El método de grabado por puntos (micropercusión) es el más adecuado para el desarrollo del proyecto, ya que la simplicidad del actuador ahorra tiempo y dinero en el desarrollo de la máquina, se utiliza un microcontrolador Arduino para el control de los motores, se tiene un control numérico para el posicionamiento de los motores y con una interfaz de usuario a través de una computadora se carga los caracteres y números que la máquina graba en las piezas.

## **1.6 Limitaciones**

Las limitaciones pueden irse generando conforme se realiza el proyecto, algunas de ellas la selección de materiales puede no ser lo más barato para mantener una calidad, otra de las limitaciones puede ser que el software GRBL no permita ajustarse al sistema.

## **1.7 Contribución del trabajo a la solución del problema**

La máquina, interfaz de usuario y control desarrollado en este trabajo permitirá tener una máquina de grabado por micropercusión de bajo costo y alta precisión. Así mismo es posible adaptarla para proyectos a medida.

**CAPÍTULO II**  
**MODELO FÍSICO**

## **2. Modelo físico del plotter XYZ para impresión de caracteres**

A continuación, se describe el diseño de los mecanismos y estructuras que conforman los sistemas del Plotter.

Un mecanismo es un dispositivo que transforma el movimiento en un patrón deseable, y por lo general desarrolla fuerzas muy bajas y transmite poca potencia. Una máquina, en general, contiene mecanismos que están diseñados para producir y transmitir fuerzas significativas [1]. El plotter fue dividido en dos sistemas, los cuales desarrollan las funciones fundamentales.

En primer lugar, se tiene un sistema de movimiento de direcciones, este sistema proporciona el movimiento en los ejes coordenados X, Y. El segundo sistema es el que permite posicionar la máquina a la altura adecuada para el objeto a grabar.

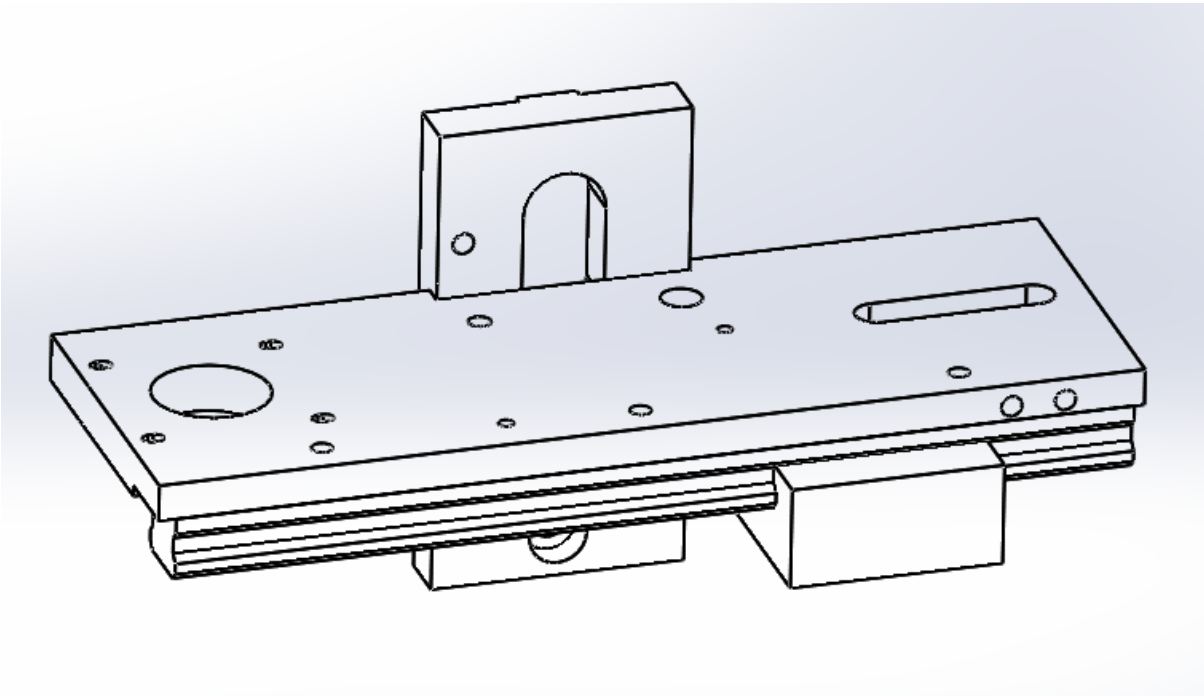
El primer sistema que permite el movimiento del punzón, está constituido por dos guías lineales de precisión tipo Hiwin, para el eje X de 180 mm y para el eje Y de 108 mm los cuales son movidos por un sistema de poleas, se implementa una banda de paso XML de 1200 mm y para el eje Y una banda paso XML de 80mm. Este tipo de sistema proporciona alta precisión y un avance rápido, ya que está especialmente diseñada para desplazar objetos de una forma sencilla además posibilita que el movimiento lineal pueda ser transmitido por motor rotativo.

### **2.1 Sistema de movimiento en el eje “X”**

El movimiento en dirección X, implica el uso de un sistema de desplazamiento que permita la tarea de trasladar el sistema del eje Y. Este sistema funciona a través de un sistema de guía lineal de precisión que mueve un bloque o carrito lineal MGN12h, sobre el cual se monta el sistema del eje Y, El carrito está anclado a una polea, la cual es movida por un motor a pasos y una transmisión de polea.



Este sistema está diseñado sobre una placa de aluminio que a su vez va montada sobre el sistema del eje Z.

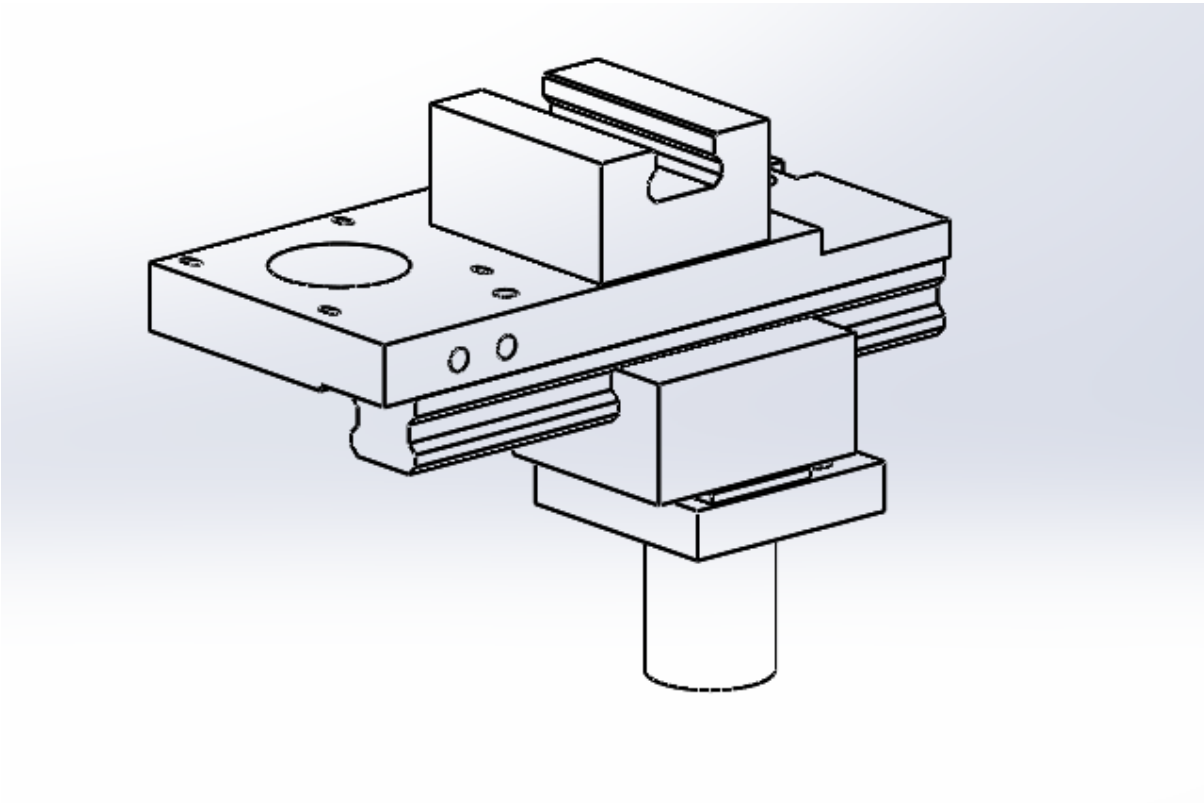


*Figura 2.1: Sistema de movimiento en eje X.*

## **2.2 Sistema de movimiento en eje “Y”**

El movimiento del plotter en dirección Y proporciona un desplazamiento a lo largo de eje horizontal. Para poder realizar la tarea de deslizamiento, los sistemas que se seleccionaron fueron los mismos que para el eje X

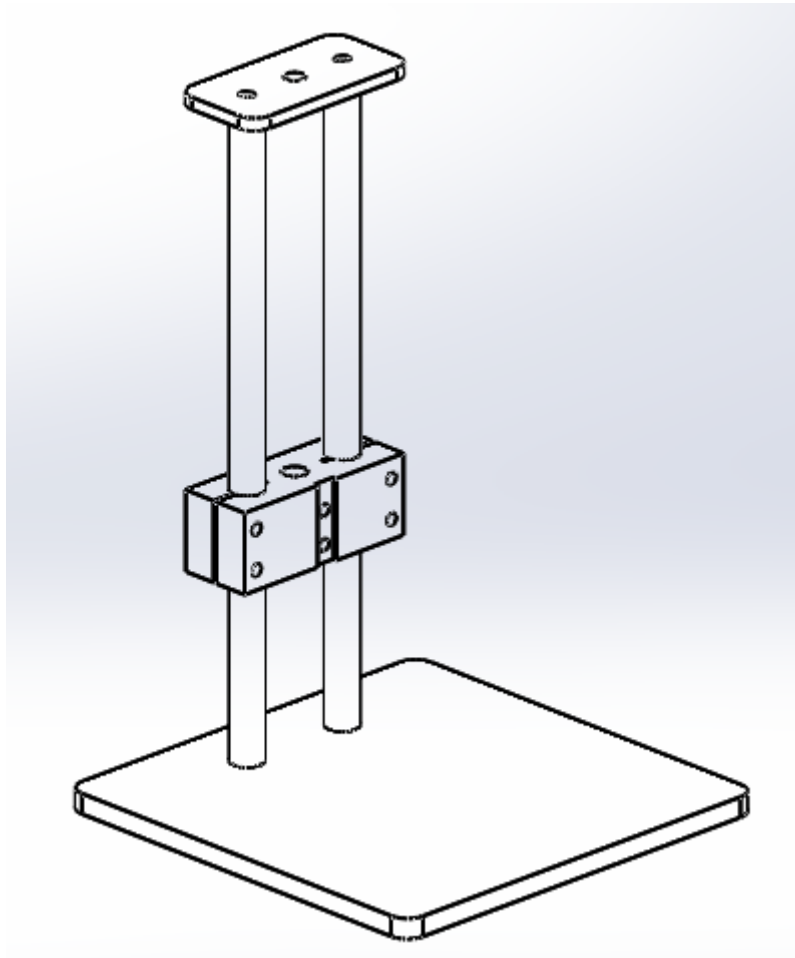
Este sistema de deslizamiento está fijado al carrito que mueve el motor del eje X, el sistema de deslizamiento del eje Y mueve el elemento de marcaje (puzón neumático), que está sujeto al bloque o carrito.



*Figura 2.2: Sistema de movimiento en eje Y.*

## **2.3 Sistema de movimiento en dirección “Z”**

El sistema de movimiento en dirección “Z” está constituido por un sistema de huesillo, este soporta el resto de la máquina y proporciona un desplazamiento vertical y es el responsable de llevar a cabo la profundidad de impacto de la aguja del punzón, este sistema se diseña para ser activado mediante una perilla (manual) ya que una vez fijada la altura de marcaje, se fija la base de la máquina para evitar que las vibraciones de impacto desajusten el resto de la máquina.

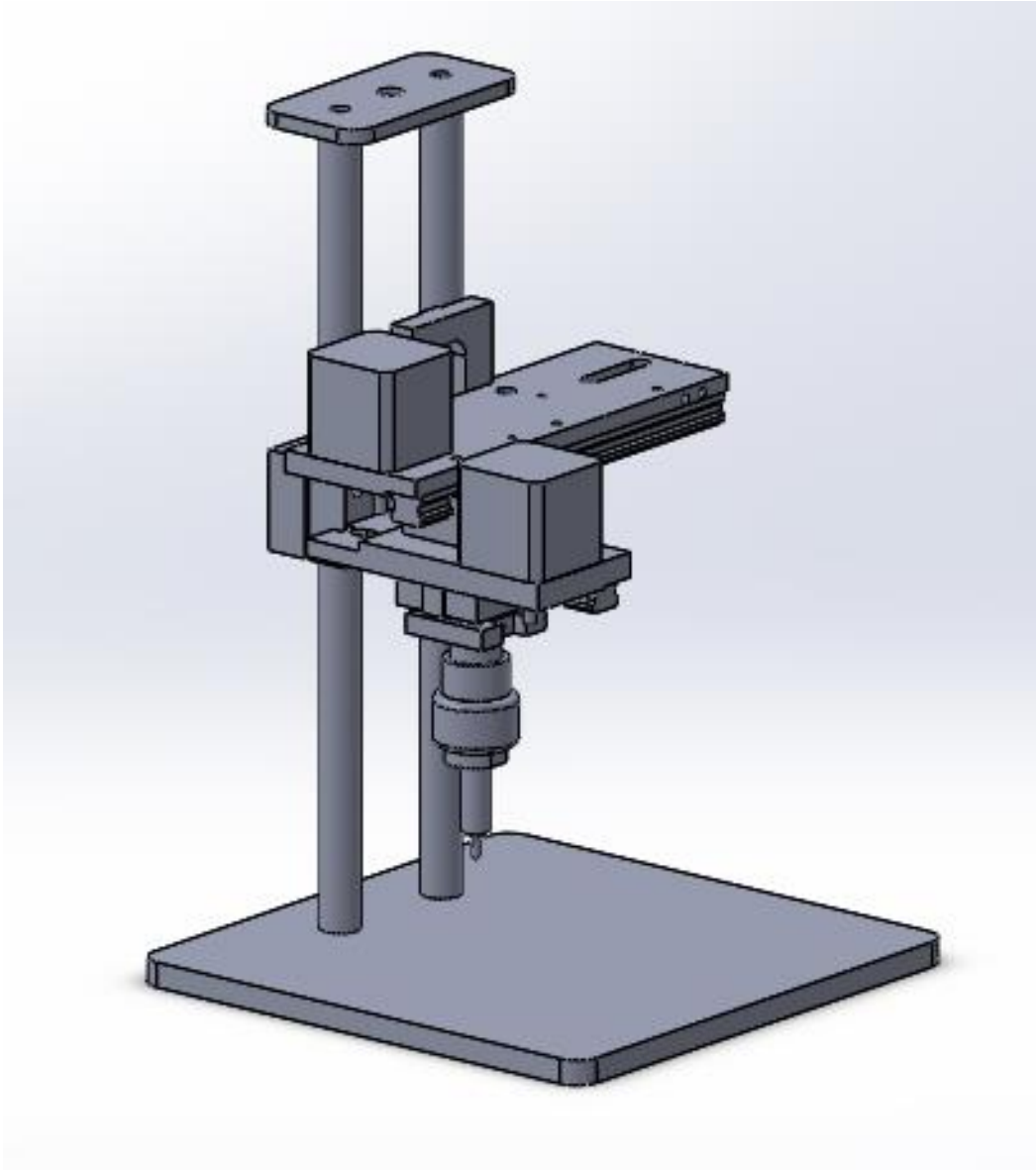


*Figura 2.3: Sistema de movimiento en eje Z.*

Hasta aquí se tienen los sistemas de movimiento de la máquina para cada eje, lo que sigue es la unión de ellos, para obtener un mecanismo de movimiento en los ejes XYZ.

## 2.4 Montaje final de las estructuras

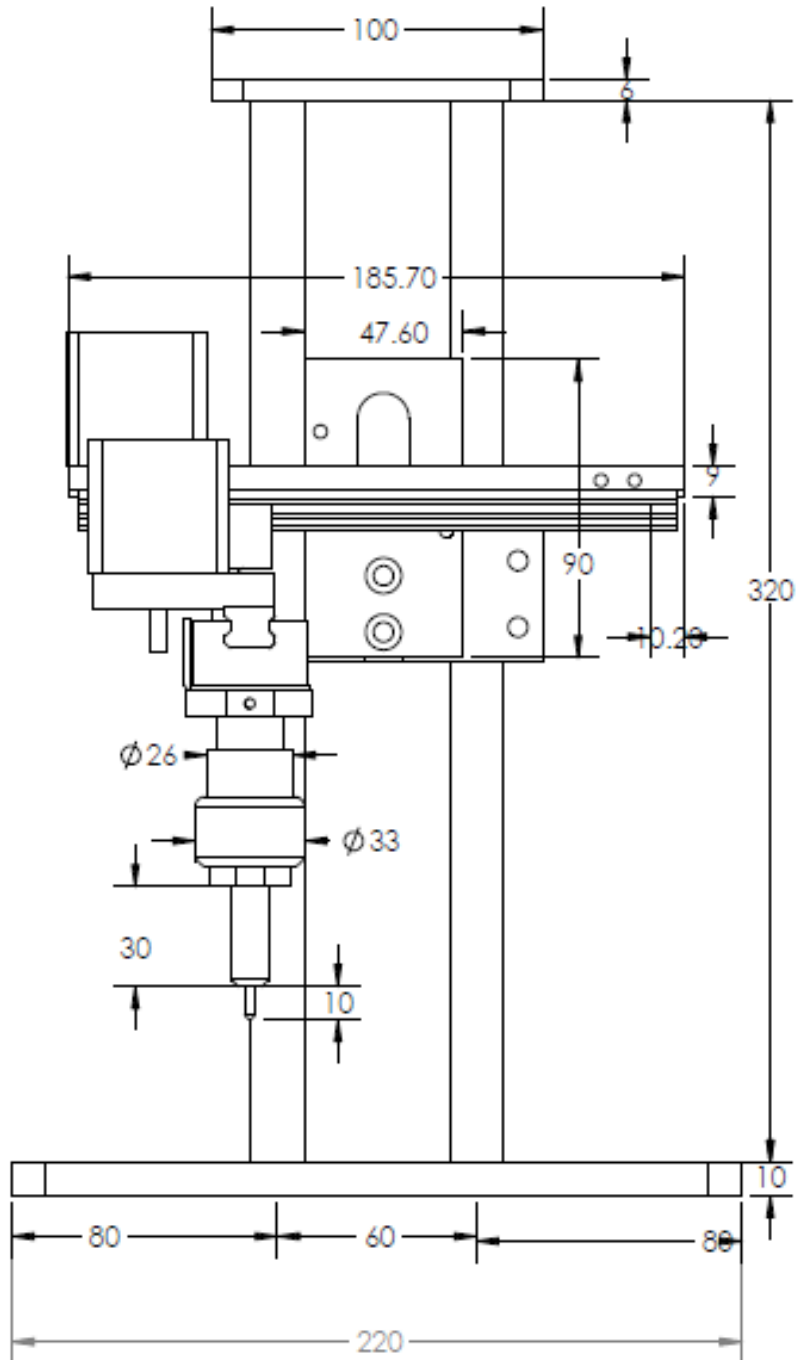
A continuación, se presenta una imagen del Plotter con todas las partes que lo componen.



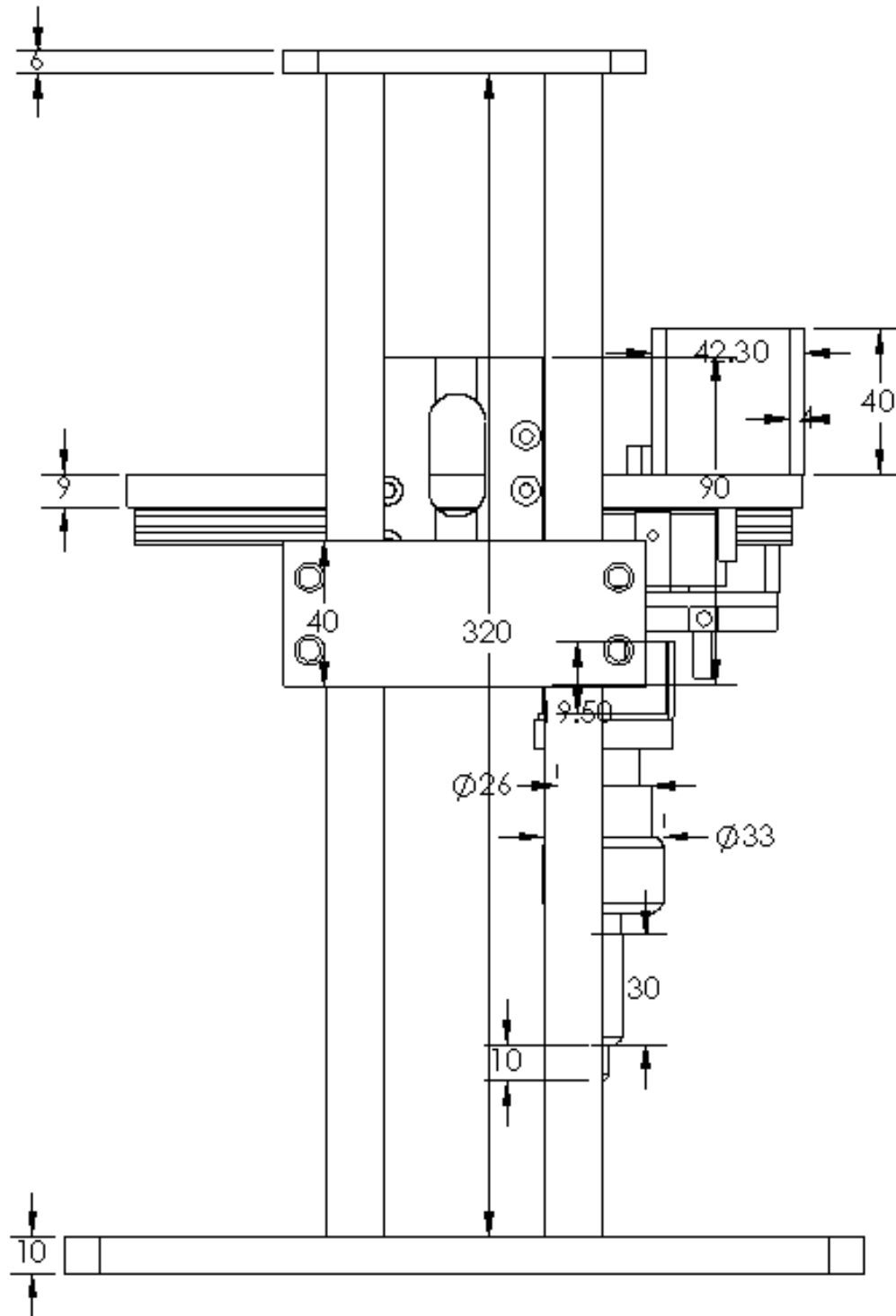
*Figura 2.4: Ensamblaje final de la máquina.*

## 2.5 Vistas del ensamblaje final

A continuación se presentan las figuras del sistema mecánico del equipo.



*Figura 2.5: Vista frontal.*



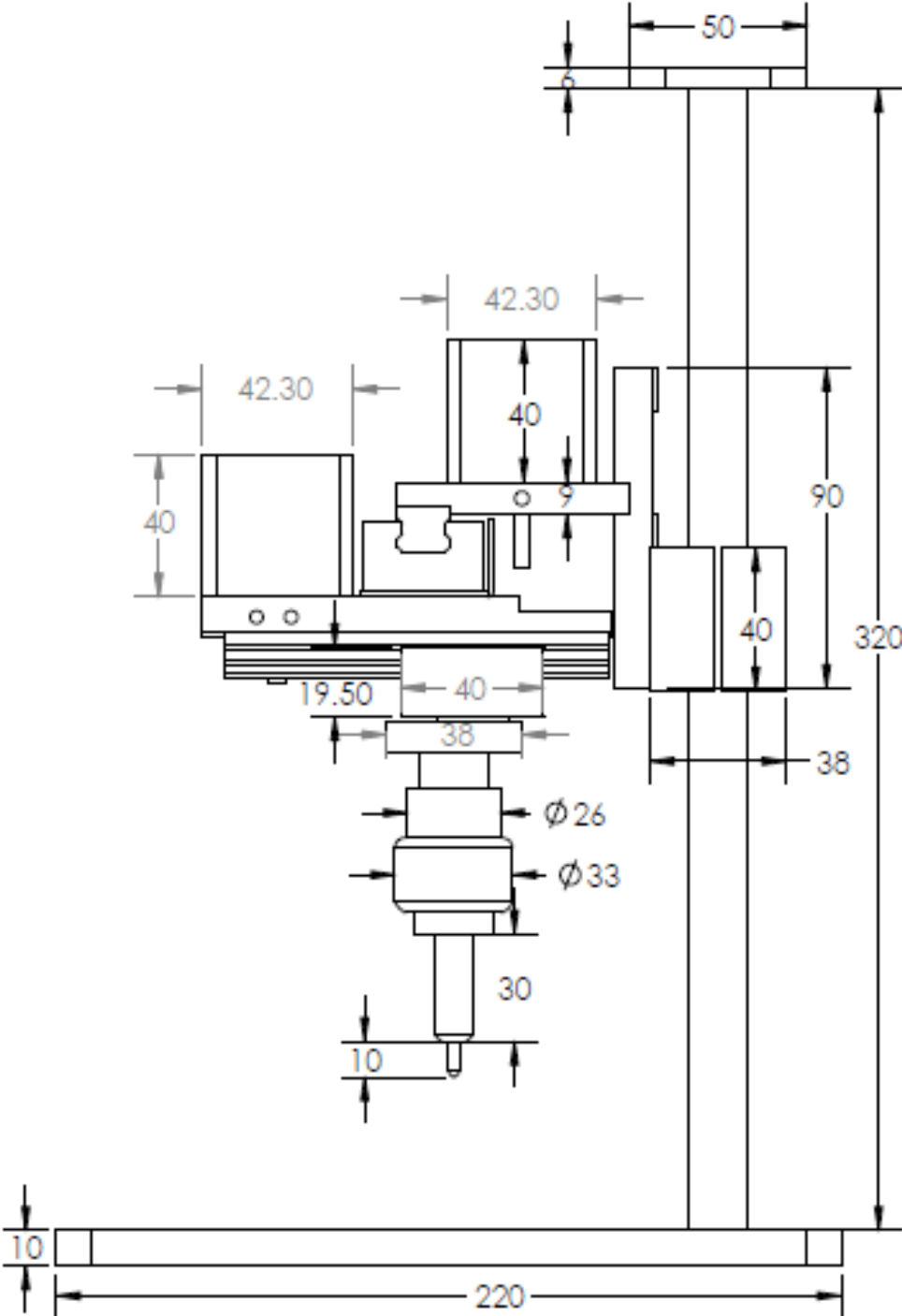


Figura 2.7: Vista lateral derecha.

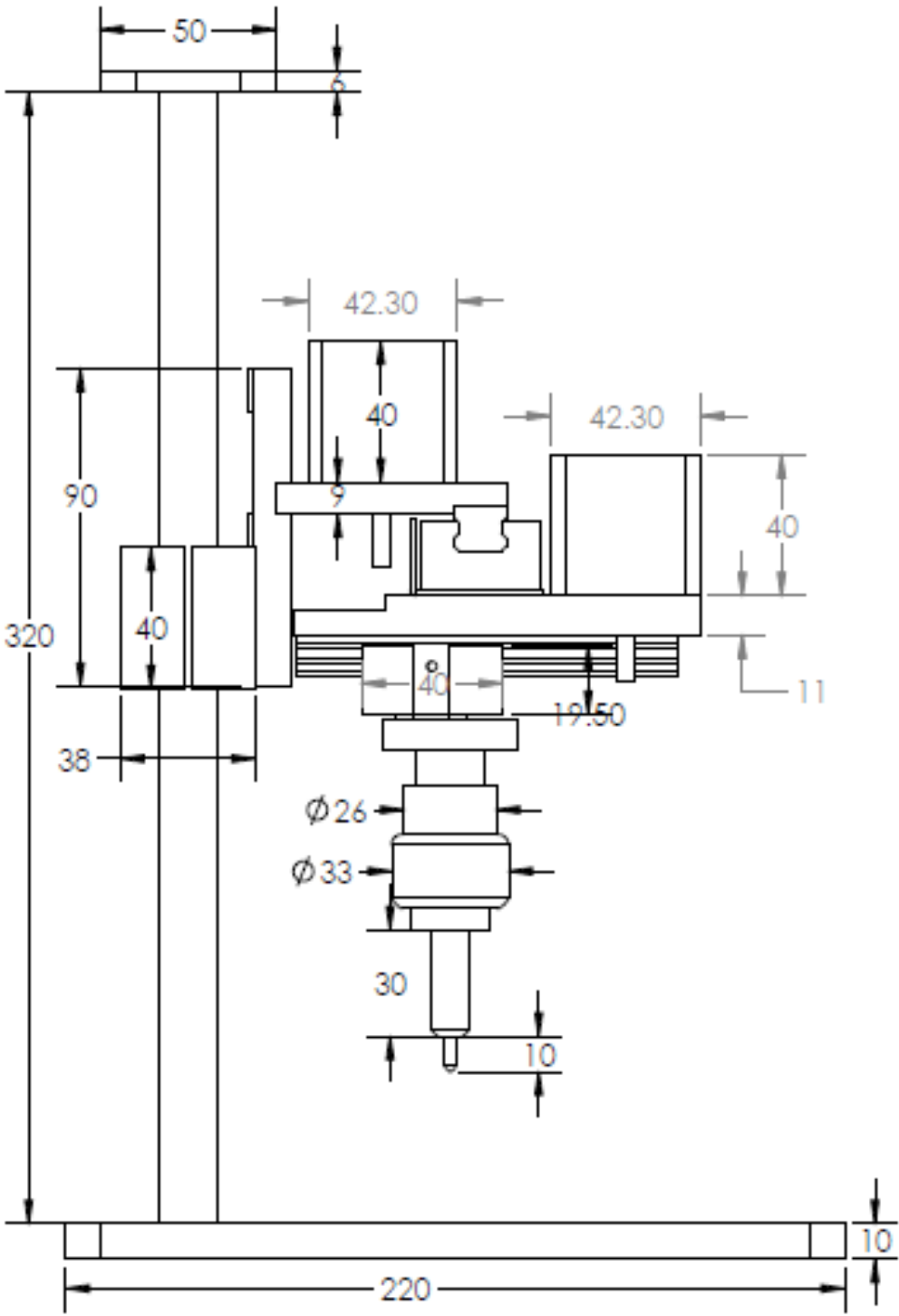


Figura 2.8: Vista lateral izquierda.



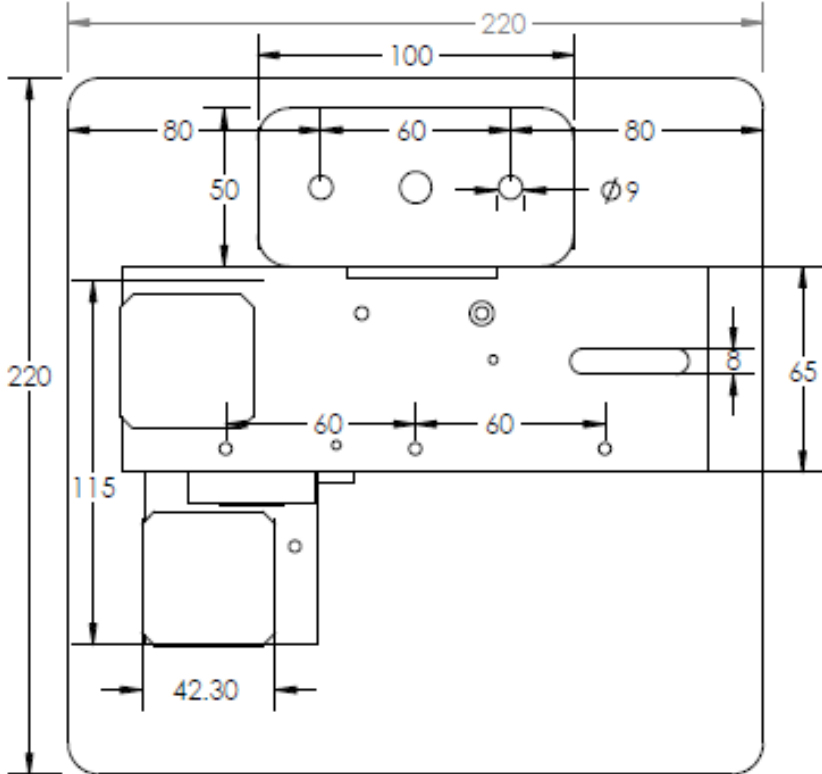


Figura 2.9: Vista superior.

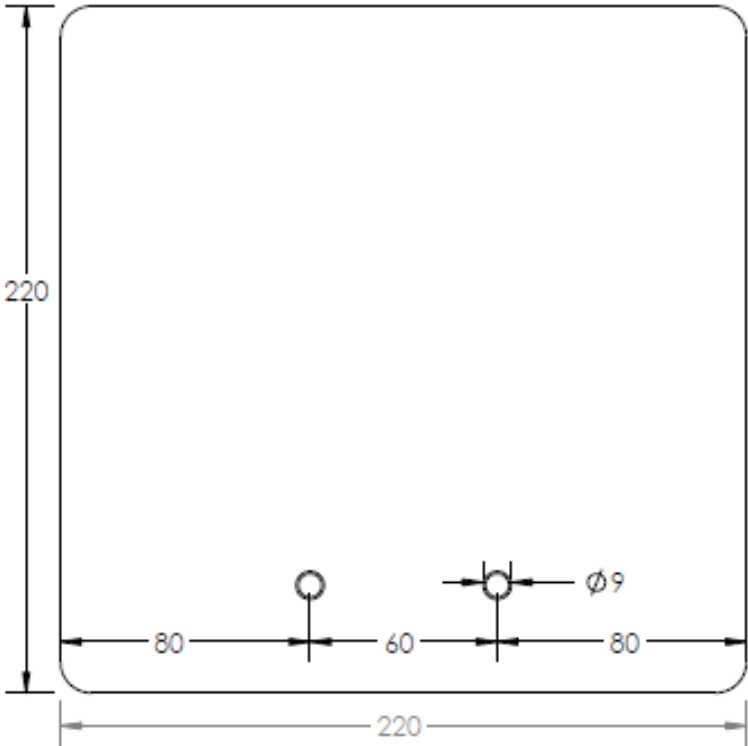
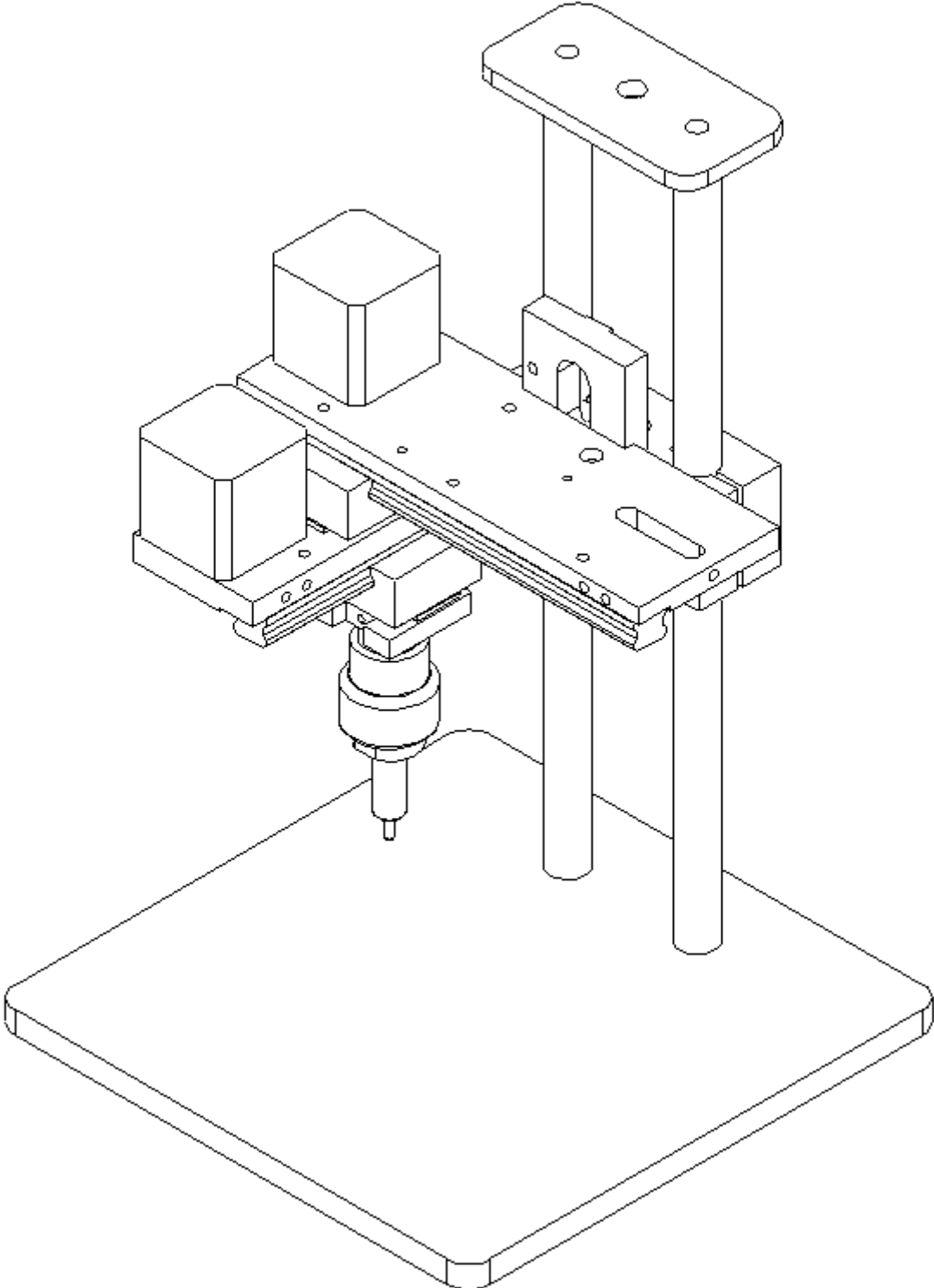


Figura 2.10: Vista inferior.



*Figura 2.11: Vista en perspectiva.*

## CAPÍTULO II. MODELO FÍSICO

Todo este equipo de la máquina de micropercusión se realizará un registro de modelo industrial ante el IMPI.

**CAPÍTULO III**  
**FUNDAMENTOS TEÓRICOS**

### **3. Fundamentos teóricos**

En el presente capítulo se presentan los fundamentos teóricos que ayudan a conocer bajo qué conceptos se desarrolló el trabajo de investigación. Además se integran definiciones que dan soporte al conocimiento en esta investigación.

#### **3.1 Introducción a las máquinas de grabado con la técnica de micropercusión**

Entendemos por grabado el resultado del trabajo realizado sobre una superficie de madera o metal, llamada plancha, por medio de instrumentos cortantes, punzantes o de ácidos que atacan la superficie metálica.

Existen diferentes técnicas de marcaje de piezas en las cuales destacan el grabado laser, la micropercusión y el maquinado, siendo estas técnicas de las más precisas y de las que cuentan con mayor participación en el mercado.

Estos métodos se pueden usar en una amplia variedad de materiales pero se debe tener en cuenta que cada sustrato presenta aspectos únicos, como la aspereza, la capacidad de soportar la tensión térmica y la fragilidad del material que se está marcando.

Teniendo como referencia estos métodos se opta por desarrollar la máquina con la tecnología de micropercusión, debido a las ventajas que presenta, principalmente en el bajo costo, la eficacia en el marcado de piezas metálicas permanentemente, también es la opción más viable para los requerimientos de los clientes de la empresa IP Industrial.

Durante un marcaje por micropercusión, la materia es repujada y no quemada, fresada o transformada como en las otras tecnologías de marcaje. No se crea ninguna viruta, por lo tanto, la pieza tiene exactamente el mismo peso antes como después del marcaje.

Esto significa también que la micropercusión no crea ningún inicio de rotura en la materia, cualquiera que sea el grosor de la pieza marcada.

## CAPÍTULO III. FUNDAMENTOS TEÓRICOS

Existen dos tecnologías para el marcado por micropercusión, neumática y electromagnética, de este modo puede responder lo mejor posible a todas las aplicaciones: marcajes profundos y rápidos sobre material en bruto con códigos DataMatrix muy precisos en pequeñas superficies mecanizadas.

El marcaje por puntos suele requerir una inversión inicial reducida y proporciona marcas permanentes. Dado que se trata simplemente de generar muescas en la superficie, no se tiene constancia de que esta tecnología dañe ni ponga en peligro la integridad del producto que se está marcando. Sin embargo, presenta gastos continuos de mantenimiento y sustitución de las agujas que generan las muescas debido al desgaste al que se someten durante el proceso de marcaje. Además, es posible que algunos productos más finos no resulten adecuados para el marcaje por puntos, ya que no existe suficiente sustancia para generar la muesca sin perforar el material.

### **3.2 Control numérico computarizado.**

CNC Significa "Control Numérico Computarizado". En una máquina CNC, a diferencia de una máquina convencional o manual, una computadora controla la posición y velocidad de los motores que accionan los ejes de la máquina. Gracias a esto puede hacer movimientos que no se pueden lograr manualmente como círculos, líneas diagonales y figuras complejas tridimensionales [2].

Las máquinas CNC son capaces de mover la herramienta al mismo tiempo en los 3 ejes para ejecutar trayectorias tridimensionales como las que se requieren para el maquinado de moldes y troqueles.

#### **3.2.1 Principio de funcionamiento**

Para mecanizar una pieza se usa un sistema de coordenadas que especificarán el movimiento de la herramienta de corte. El sistema se basa en el control de los movimientos de la herramienta de trabajo con relación a los ejes de coordenadas de la máquina, usando un

programa informático ejecutado por una computadora. En el caso de un torno, hace falta controlar los movimientos de la herramienta en dos ejes de coordenadas: el eje de las X para los desplazamientos longitudinales del carro y el eje de las Z para los desplazamientos transversales de la torre. En el caso de las fresadoras se controlan también los desplazamientos verticales, que corresponden al eje Y. Para ello se incorporan servomotores en los mecanismos de desplazamiento del carro y la torreta, en el caso de los tornos y en la mesa en el caso de la fresadora; dependiendo de la capacidad de la máquina, esto puede no ser limitado únicamente a tres ejes.

### **3.2.2 Programación en el control numérico**

La programación es la base del control numérico, conocer dicha programación es absolutamente imprescindible para cualquier operario que intervenga en el proceso constructivo, desde la oficina técnica, hasta la mecanización última. [3]

#### **3.2.2.1 Programación manual**

En este caso, el programa pieza se escribe únicamente por medio de razonamientos y cálculos que realiza un operario. El programa de mecanizado comprende todo el conjunto de datos que la máquina necesita para la mecanización de la pieza.

A la información en conjunto que corresponde a una misma fase del mecanizado se le denomina bloque o secuencia y se numera para facilitar su búsqueda. Este conjunto de información es interpretado por el intérprete de órdenes. Una secuencia o bloque de programa debe contener todas las funciones geométricas, funciones máquina y funciones tecnológicas del mecanizado. De tal modo, un bloque de programa consta de varias instrucciones. El comienzo del control numérico ha estado caracterizado por un desarrollo caótico de los códigos de programación pues cada constructor utilizaba el suyo particular. Posteriormente, se vio la necesidad de normalizar los códigos de programación como condición indispensable para que un mismo programa pudiera servir para diversas máquinas con tal de que fuesen del mismo tipo.

### **3.2.2.2 Programación automática**

En este caso, los cálculos los realiza un computador, a partir de datos suministrados por el programador dando como resultado el programa de la pieza en un lenguaje de intercambio llamado APT, que posteriormente será traducido mediante un post-procesador al lenguaje máquina adecuado para cada control por computadora.

### **3.2.3 Caracteres más usados.**

Los caracteres más usados comúnmente, regidos bajo la norma DIN 66024 y 66025 son, entre otros, los siguientes:

- N: es la dirección correspondiente al número de bloque o secuencia. Esta dirección va seguida normalmente de un número de tres o cuatro cifras. En el caso del formato N03, el número máximo de bloques que pueden programarse es 1000 (N000 hasta N999).
- X, Y, Z: son las direcciones correspondientes a las cotas según los ejes X, Y, Z de la máquina herramienta (Y planos cartesianos). Dichas cotas se pueden programar en forma absoluta o relativa, es decir, con respecto al cero pieza o con respecto a la última cota respectivamente.
- G: es la dirección correspondiente a las funciones preparatorias. Se utilizan para informar al control de las características de las funciones de mecanizado, como por ejemplo, forma de la trayectoria, tipo de corrección de herramienta, parada temporizada, ciclos automáticos, programación absoluta y relativa, etc. La función G va seguida de un número de dos cifras que permite programar hasta 100 funciones preparatorias diferentes.



**Ejemplos:**

*Tabla I: Descripción de las funciones de códigos G.*

Código	Funcionamiento
G00	El trayecto programado se realiza a la máxima velocidad posible, es decir, a la velocidad de desplazamiento en rápido.
G01	Los ejes se gobiernan de tal forma que la herramienta se mueve a lo largo de una línea recta.
G02	Interpolación circular en sentido horario.
G03	Interpolación circular en sentido anti horario.
G05	Trabajo en arista matada.
G07	Trabajo en arista viva.
G09	Trayectoria circular definida por 3 puntos.
G10	Anulación de la imagen espejo.
G11	Imagen espejo en X.
G12	Imagen espejo en Y.
G17	Selección del plano XY.
G18	Selección del plano XZ.
G19	Selección del plano YZ.
G25	Salto incondicional.
G33	Indica ciclo automático de roscado.

CAPÍTULO III. FUNDAMENTOS TEÓRICOS

G37	Entrada tangencial.
G38	Salida tangencial.
G40	Cancela compensación.
G41	Compensación de corte hacia la izquierda.
G42	Compensación de corte a la derecha.
G43	Compensación del largo de las herramientas.
G44	Anulación de la compensación del largo de las herramientas.
G53/G59	Traslados de origen.
G70	Programación en pulgadas.
G71	Programación en milímetros.
G72	Factor de escala.
G73	Giro del sistema de coordenadas
G77	Es un ciclo automático que permite programar con un único bloque el torneado de un cilindro, etc.
G87	Cajera rectangular
G88	Cajera circular.
G90	Programación de cotas absolutas.
G91	Programación de cotas incrementales.
G94	Velocidad de avance F en mm/min.

### CAPÍTULO III. FUNDAMENTOS TEÓRICOS

- M: es la dirección correspondiente a las funciones auxiliares o complementarias. Se usan para indicar a la máquina herramienta que se deben realizar operaciones tales como parada programada, rotación del husillo a derechas o a izquierdas, cambio de útil, etc. La dirección m va seguida de un número de dos cifras que permite programar hasta 100 funciones auxiliares diferentes.

#### **Ejemplos:**

*Tabla II: Descripción de las funciones de códigos M.*

M00	Provoca una parada incondicional del programa, detiene el husillo y la refrigeración.
M01	Alto opcional.
M02	Indica el fin del programa. Se debe escribir en el último bloque del programa y posibilita la parada del control una vez ejecutadas el resto de las operaciones contenidas en el mismo bloque.
M03	Activa la rotación del husillo en sentido horario.
M04	Activa la rotación del husillo en sentido anti horario, etc.
M08	Lubricación ON.
M09	Lubricación OFF.
M10	Encendido del accesorio de iluminación.
M30	Final del programa con vuelta al inicio.
M05	Parada del cabezal.
M06	Cambio de herramienta (con parada del programa o sin) en las máquinas de cambio automático no conlleva la parada del programa.

## CAPÍTULO III. FUNDAMENTOS TEÓRICOS

- F: es la dirección correspondiente a la velocidad de avance. Va seguida de un número de cuatro cifras que indica la velocidad de avance en mm/min.
- S: es la dirección correspondiente a la velocidad de rotación del husillo principal. Se programa directamente en revoluciones por minuto, usando cuatro dígitos.
- I, J, K: son direcciones utilizadas para programar arcos de circunferencia. Cuando la interpolación se realiza en el plano X-Y, se utilizan las direcciones I y J. Análogamente, en el plano X-Z, se utilizan las direcciones I y K, y en el plano Y-Z, las direcciones J y K.
- T: es la dirección correspondiente al número de herramienta. Va seguido de un número de cuatro cifras en el cual los dos primeros indican el número de herramienta y los dos últimos el número de corrección de las mismas.
- A: ángulo
- G: función preparatoria
- M: función auxiliar
- N: numeración del bloque
- P: programa
- R: radio

### 3.2.4 Plano cartesiano

Las **coordenadas cartesianas** o **coordenadas rectangulares** (sistema cartesiano) son un tipo de coordenadas ortogonales usadas en espacios euclídeos, para la representación gráfica de una relación matemática (funciones matemáticas y ecuaciones de geometría

analítica), o del movimiento o posición en física, caracterizadas por tener como referencia ejes ortogonales entre sí que concurren en el punto origen. En las coordenadas cartesianas se determinan las coordenadas al origen como la longitud de cada una de las proyecciones ortogonales de un punto dado sobre cada uno de los ejes. La denominación de 'cartesiano' se introdujo en honor de René Descartes, quien las utilizó por primera vez de manera formal [4].

El sistema en sí es un sistema bidimensional, que se denomina plano cartesiano. El punto de intersección de las rectas, por definición, considera como el punto cero de las rectas y se conoce como origen de coordenadas. Al eje horizontal o de las abscisas se le asigna los números reales de las equis ("x"); y al eje vertical o de las ordenadas se le asignan los números reales de las yes ("y"). Al cortarse las dos rectas, dividen al plano en cuatro regiones o zonas, que se conocen con el nombre de cuadrantes:

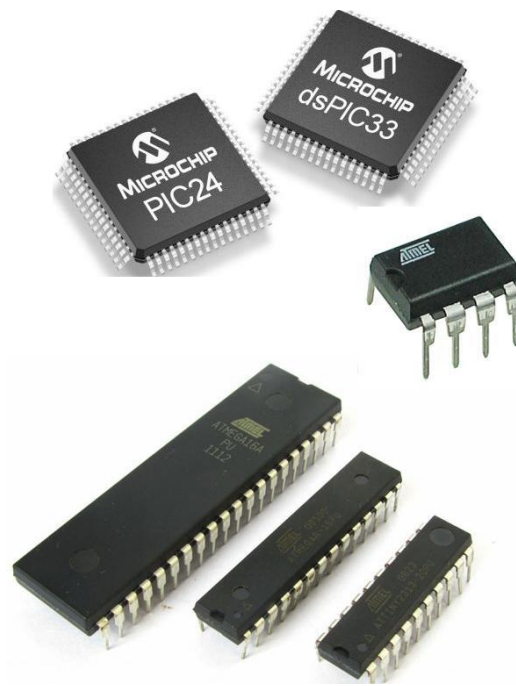
- Primer cuadrante "I": Región superior derecha
- Segundo cuadrante "II": Región superior izquierda
- Tercer cuadrante "III": Región inferior izquierda
- Cuarto cuadrante "IV": Región inferior derecha

### **3.3 Electrónica y control**

En esta sección se describen los elementos seleccionados para el desarrollo electrónico con el que se planea controlar la máquina.

#### **3.3.1 Microcontrolador**

Un microcontrolador es un ordenador en un único chip [figura de abajo]. La palabra micro indica que el dispositivo es pequeño, y controlador indica que el dispositivo se puede usar en aplicaciones de control. Otro término usado para los microcontroladores es el de controlador embebido, puesto que la mayoría de los microcontroladores se integran (o se embeben) junto con dispositivos que se encargan de controlar [5].



*Figura 3.1: Ejemplo de microcontroladores.*

### 3.3.2 Arduino UNO

Se elige la tarjeta Arduino UNO por contar con las características suficientes para la elaboración del proyecto, también porque el firmware GRBL está desarrollado especialmente para esta tarjeta.

Arduino UNO es una tarjeta basada en el microcontrolador ATmega328P, cuenta con 14 entradas/salidas digitales (de las cuales 6 pueden ser usadas como PWM), cuenta con 6 entradas analógicas, un cristal de cuarzo de 16MHz, un puerto USB, una toma de corriente, comunicación I2C. Lo que abre un gran panorama y la hace ideal para la lectura de sensores analógicos y digitales. El microcontrolador atmega328 que se encuentra en la placa Arduino UNO se programa mediante el lenguaje de programación Arduino que está basado en el

lenguaje wiring y el entorno está inspirado en processing, lo que hace muy intuitivo el uso del software para el desarrollo de aplicaciones.[6]



**Figura 3.2:** Fotografía de un Arduino UNO.

### 3.3.3 Firmware

Cuando hablamos de firmware nos referimos al conjunto de instrucciones de un programa informático que se encuentra registrado en una memoria ROM, flash o similar. Estas instrucciones fijan una lógica primaria que ejerce el control de los circuitos de alguna clase de artefacto.

El concepto de firmware fue utilizado por primera vez en el año 1967, en un artículo que Rudy Meléndez publicó en la revista *Datamation*.

El firmware forma parte del hardware, ya que se encuentra integrado a la electrónica, pero también es considerado parte del software, al estar desarrollado bajo un lenguaje de programación. Se podría decir que funciona como el nexo entre las instrucciones que llegan al dispositivo desde el exterior y sus diversas partes electrónicas.

Cualquier firmware debe cumplir tres funciones claras. En primer lugar, otorgar al sistema las rutinas fundamentales de funcionamiento y respuesta con respecto a las peticiones usuales que recibe y debe satisfacer al usuario.

### 3.3.4 Controlador digital para motores a pasos DM422C (Fully digital stepping drive)

Se elige el controlador DM422C para los motores paso a paso, es un versátil controlador digital basado en DSP con un algoritmo avanzado de control. Brinda un único nivel de suavidad al sistema, proveyendo un torque óptimo y se anula la inestabilidad de rango medio. La tecnología de auto-prueba y auto-configuración ofrece una respuesta óptima con diferentes motores y es fácil de usar. Los controladores pueden operar con mucho menos ruido, menos calentamiento y movimientos más suaves que la mayoría de las unidades en el mercado.



*Figura 3.3: Driver DM422C para motores a paso.*

### 3.3.5 Motores a pasos

Los motores paso a paso son dispositivos electromagnéticos, rotativos, incrementales que convierten pulsos digitales en rotación mecánica.

La cantidad de rotación es directamente proporcional al número de pulsos y la velocidad de rotación es relativa a la frecuencia de los pulsos. Estos motores son simples de



## CAPÍTULO III. FUNDAMENTOS TEÓRICOS

operar en una configuración de lazo cerrado y debido a su tamaño proporcionan un excelente torque a baja velocidad. [7]

La conmutación se debe manejar de manera externa con un controlador electrónico y, típicamente, los motores y sus controladores se diseñan de manera que el motor se pueda mantener en una posición fija y también para que se lo pueda hacer girar en un sentido y en el otro.[8]

Entre los beneficios de estos motores se incluyen:

- Un diseño efectivo y bajo costo
- alta confiabilidad
- libres de mantenimiento (no disponen de escobillas)
- Lazo abierto (no requieren dispositivos de realimentación)
- Límite conocido al “error de posición dinámica”



*Figura 3.4: Motor a pasos.*

### 3.3.6 Sensor final de carrero (Limit Switch)

Dentro de los componentes electrónicos, se encuentra el final de carrera o sensor de contacto (también conocido como "interruptor de límite"), son dispositivos electrónicos, neumáticos o mecánicos situados al final del recorrido o de un elemento móvil, como por

ejemplo una cinta transportadora, con el objetivo de enviar señales que puedan modificar el estado de un circuito.

Internamente pueden contener interruptores normalmente abiertos (NA), cerrados (NC) o conmutadores dependiendo de la operación que cumplan al ser accionados, de ahí la gran variedad de finales de carrera que existen en mercado.

Los finales de carrera están fabricados en diferentes materiales tales como metal, plástico o fibra de vidrio.



*Figura 3.5: Limit switch.*

### **3.4 Interfaz gráfica de usuario**

La Interfaz Gráfica de Usuario (GUI por su nombre en inglés, Graphical User Interface) es parte fundamental de cualquier aplicación; al comenzar a trabajar con una computadora el usuario comienza a interactuar con la Interfaz, ya sea la del sistema operativo, la de un software en particular o la de cualquier sitio web. Es donde comienza la interacción hombre-computadora. El diseño de la GUI no se lo debe considerar como una tarea secundaria y sin importancia; por el contrario el equipo de desarrollo debe contar con integrantes especializados en el tema.

## CAPÍTULO III. FUNDAMENTOS TEÓRICOS

Requerimientos particulares:

La interfaz de usuario debe cumplir con los requisitos para que la máquina funcione correctamente, a continuación se presentan las funcionalidades:

- Configuración de la comunicación.
- Estado de la máquina
- Control manual
- Marcas
- Impresión

### 3.4.1 Visual Studio IDE



*Figura 3.6: Logotipo Visual Studio.*

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows, Linux y macOS.

Visual Studio es un conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento, permitiendo

## CAPÍTULO III. FUNDAMENTOS TEÓRICOS

a los desarrolladores crear sitios y aplicaciones web, así como otros servicios web en cualquier entorno que soporte la plataforma.

En palabras más específicas, Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML [9].

### **3.4.2 C#**

C # es un lenguaje de programación muy flexible y potente con una historia interesante. Fue desarrollado por Microsoft Corporation por una variedad de razones, algunas técnicas, algunas políticas y otras de marketing.

C # se parece mucho a los lenguajes de programación C ++ y Java, ya que ha prestado (o mejorado) las características proporcionadas por estos lenguajes. Los orígenes de Java y C ++ se remontan a un lenguaje llamado C, que es un lenguaje altamente eficiente y entretenido que se inventó a principios de los años setenta. C es famoso ya que el lenguaje del sistema operativo UNIX fue escrito y fue especialmente diseñado para esto [10].

### **3.4.3 C# y los objetos**

El lenguaje C # está orientado a objetos. Los objetos son un mecanismo organizativo que le permite dividir su programa en partes sensibles, cada una de las cuales está a cargo de parte del sistema general. El diseño orientado a objetos hace que los proyectos grandes sean mucho más fáciles de diseñar, probar y extender. También le permite crear programas que pueden tener un alto grado de confiabilidad y estabilidad.

La programación orientada a objetos es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial [11].

### **3.5 Elementos mecánicos**

En esta sección se describen los principales componentes que integran la máquina.

#### **3.5.1 Estructura**

En ingeniería, una estructura es el conjunto de elementos resistentes, convenientemente unidos entre sí, que reaccionan ante el efecto de cargas aplicadas. Su finalidad es resistir las acciones internas y transmitir las cargas hacia los apoyos, manteniendo la forma general de un sólido [12].

Sobre esta van montadas las guías y las transmisiones. La estructura puede ser de acrílico, madera, de aluminio o de acero.

La estructura debe ser lo suficientemente resistente para no torcerse y contrarrestar los esfuerzos generados por el movimiento de los ejes. En una impresora 3d, cortador láser, o un plotter no se ejercen grandes esfuerzos en la máquina por lo que solo debe preocuparse de que la máquina sea hecha con un material lo suficientemente rígido para no vibrar. En el caso de que la máquina sea de corte por desprendimiento de viruta la estructura debe ser de un material igual o más rígido que el que queremos cortar.

La estructura también contiene elementos de fijación como tornillos, angulares y otros accesorios que la mantienen unida.

### 3.5.2 Sistema de desplazamiento

Para que una máquina CNC puede mover sus diferentes ejes, es necesario contar con un sistema de desplazamiento. El sistema de desplazamiento es el encargado de soportar las altas cargas y al mismo tiempo permitir el movimiento de los diferentes ejes en forma lineal, este sistema cuenta con dos elementos: Las guías lineales y los rodamientos, las guías lineales son los elementos que se colocan en toda la trayectoria por donde se permitirá al eje de la máquina desplazarse, y los rodamientos son los elementos que se desplazan sobre las guías lineales.

Unos de los sistemas que por su excelencia son usados por las marcas de máquinas CNC más reconocidas de la industria son las guías lineales de precisión, conocida por muchos erróneamente como guías Hiwin.



*Figura 3.7: Sistema de guía Hiwin.*

### 3.5.3 Transmisión

La transmisión es la parte de la máquina encargada de convertir el movimiento rotativo de los motores en movimientos lineales.

Existen varios tipos de transmisiones, para este trabajo se opta por utilizar un sistema de correa dentada.

### CAPÍTULO III. FUNDAMENTOS TEÓRICOS

Este sistema es uno de los más usados en impresoras 3d y máquinas hobbistas, su costo es bajo, no requiere mantenimiento y es silencioso. Su principal problema es que no es conveniente usarlo en máquinas grandes y de gran torque ya que tiende a estirarse al ser sometido a cargas medias perdiendo precisión.



*Figura 3.8: Sistema de transmisión por correa dentada*

Los elementos descritos en este capítulo son los seleccionados para el funcionamiento de la máquina, los principales criterios para su selección son el bajo costo sin sacrificar la calidad y precisión.

**CAPÍTULO IV**  
**TÉCNICAS DE PROGRAMACIÓN**



## 4. Técnicas de programación

### 4.1 Firmware del proyecto

En la primera para de este capítulo se explica el firmware utilizado en el proyecto junto con su configuración.



*Figura 4.1: Logotipo del firmware GRBL.*

Se opta por utilizar el firmware de GRBL para este proyecto, por ser un programa desarrollo especialmente para la tarjeta Arduino UNO y cuenta con la característica Open Source, y tiene años desarrollo, siendo un programa muy sólido y confiable, GRBL implemente el control de los motores interpretando el código “G”.

A continuación se da una breve descripción del software.

GRBL es un software gratuito, de código abierto y de alto rendimiento para el controlar el movimiento de las máquinas que se mueven, que hacen cosas o que hacen que las cosas se muevan, y se ejecutarán directamente en un Arduino. Si el movimiento de los fabricantes fuera industrial, GRBL sería el estándar industrial. [13]

## 4.1.1 Instalación de GRBL versión 1.1

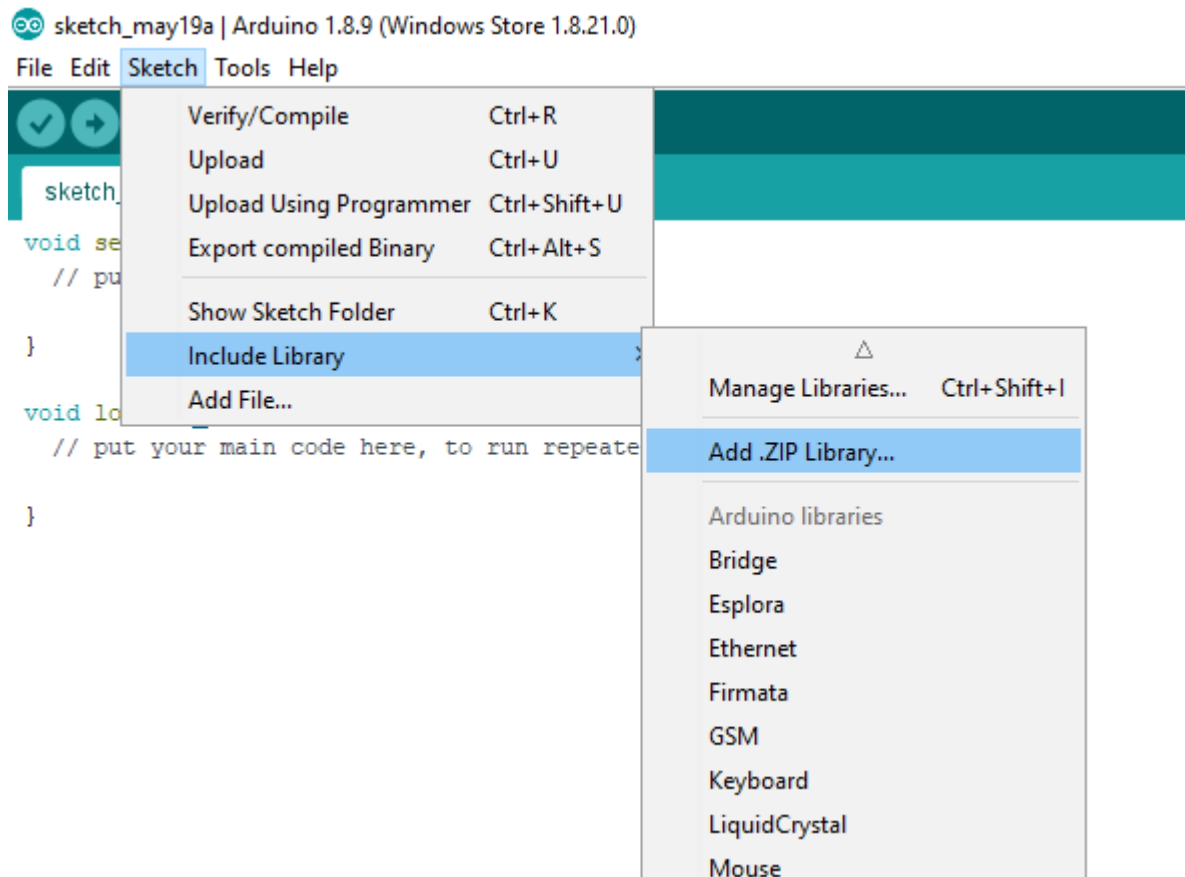
Hay dos formas de instalar el Firmware de GRBL en Arduino, en este trabajo se elige la instalación a través del IDE de Arduino.

Primero debemos instalar el IDE de Arduino en nuestra computadora, podemos descargar el instalador para Windows directamente desde la página de Arduino:

<https://www.arduino.cc/en/Main/Software>

Una vez instalado el IDE, debemos descargar el Firmware GRBL en la versión 1.1, lo podemos hacer desde GitHub (<https://github.com/gne> (Grbl, 2017)[a/grbl](https://github.com/gne)).

Una vez descargados los archivos necesarios abrimos el IDE de Arduino y nos vamos a la pestaña Sketch, nos vamos a Include Library, luego se da clic sobre Add .ZIP Library.

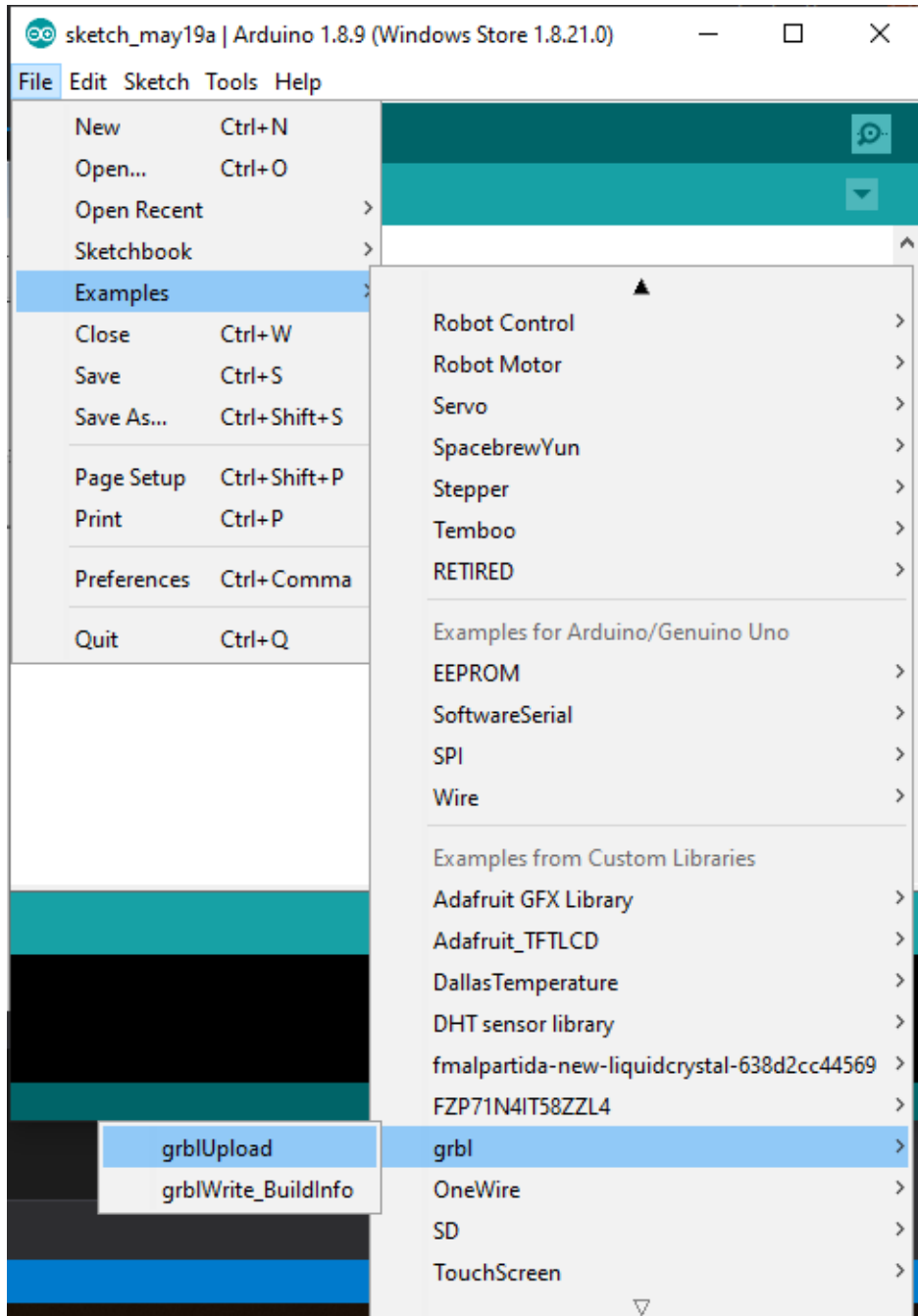


**Figura 4.2:** Instalación de la librería en el IDE de Arduino.

## CAPÍTULO IV. TÉCNICAS DE PROGRAMACIÓN

Se abre una nueva ventana de explorador de archivos y se busca el archivo grbl-master.zip, se da clic en Open, se espera a que se instale la librería.

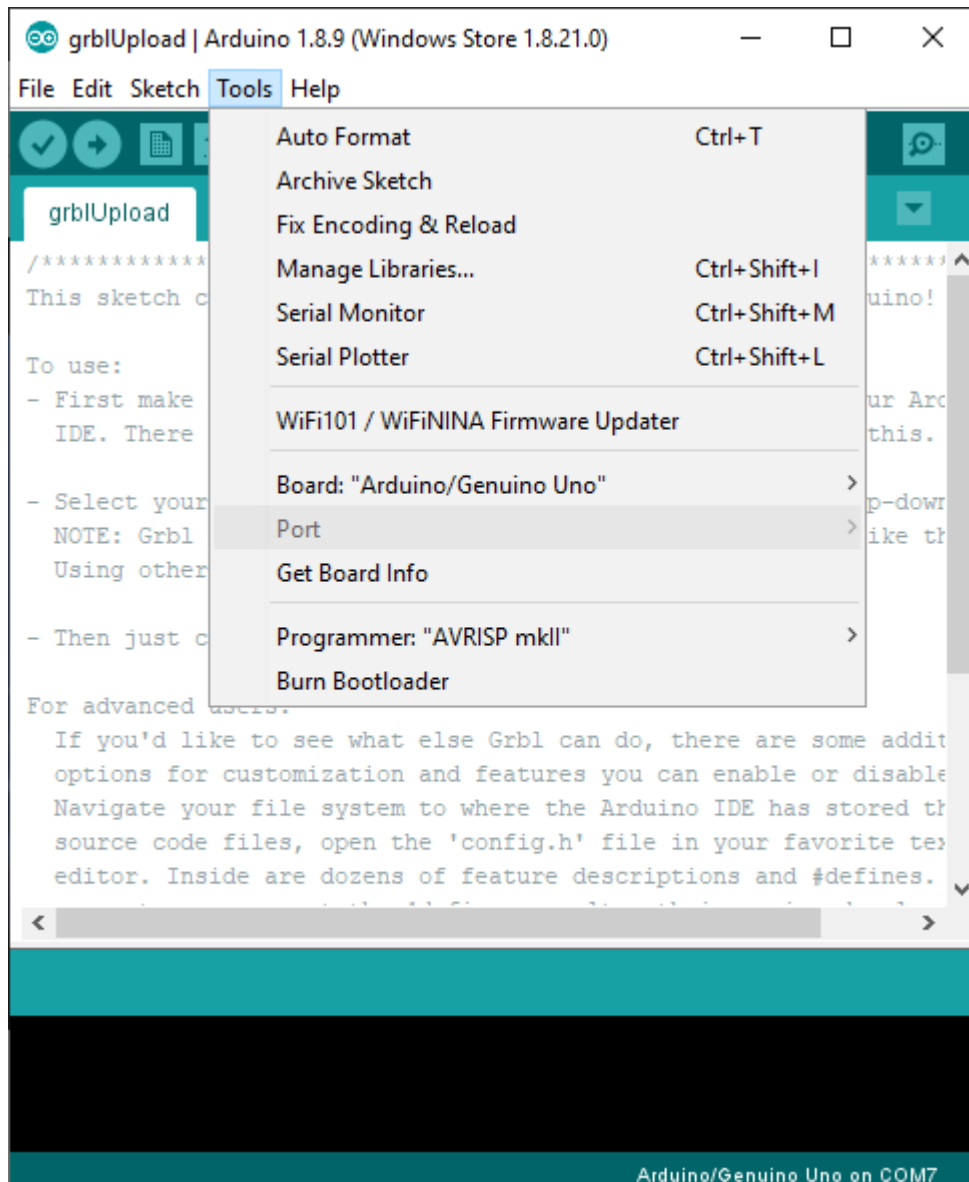
Después en el IDE de Arduino se dirige a la pestaña File, se da clic en Examples, se busca grbl y se elige grblUpload.



**Figura 4.3:** Apertura del sketch para la instalación de GRBL en Arduino UNO.

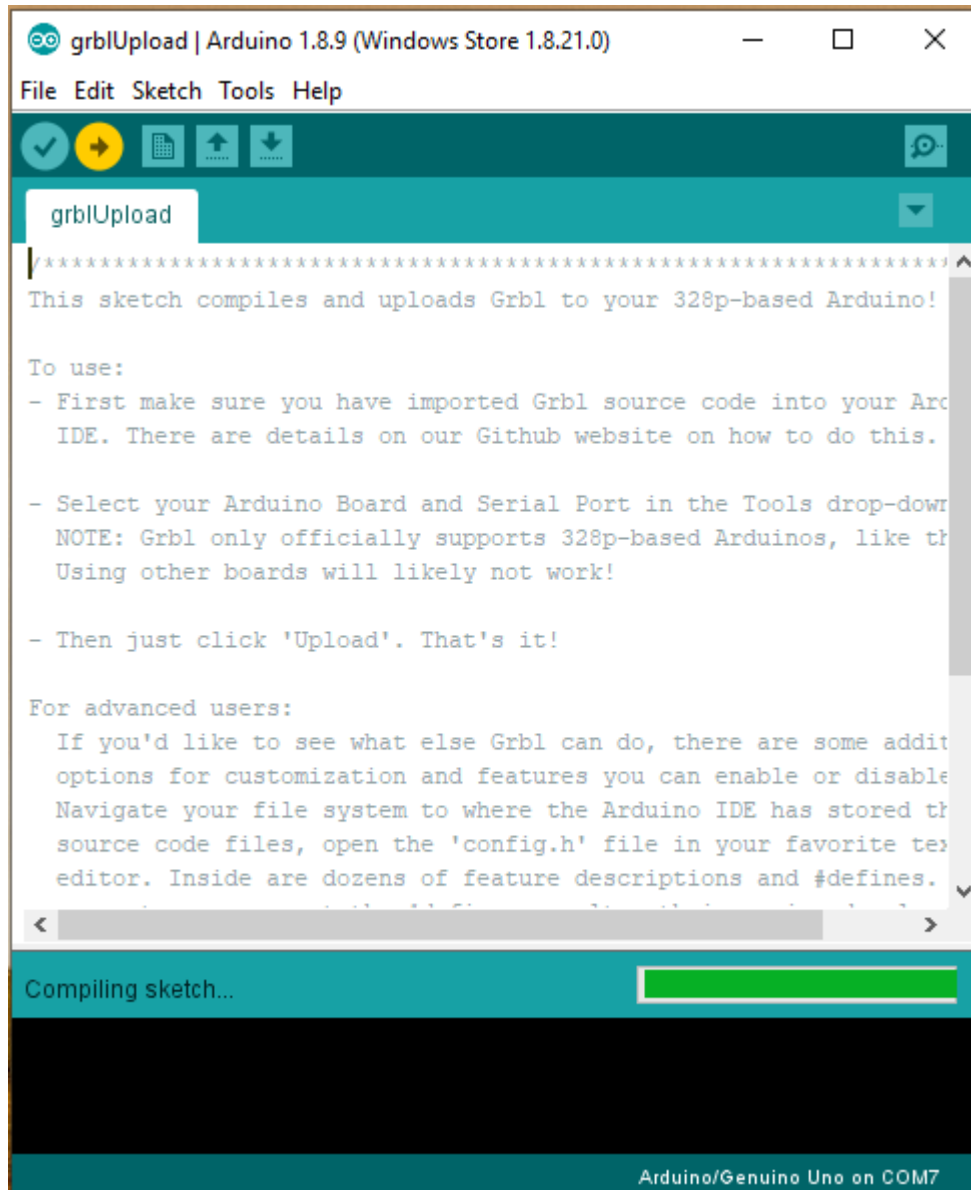
Lo siguiente es elegir la tarjeta “Arduino/Genuino UNO” en la pestaña Tools-Board.

Se conecta el Arduino Uno a la computadora y se elige el puerto que se le asigna.



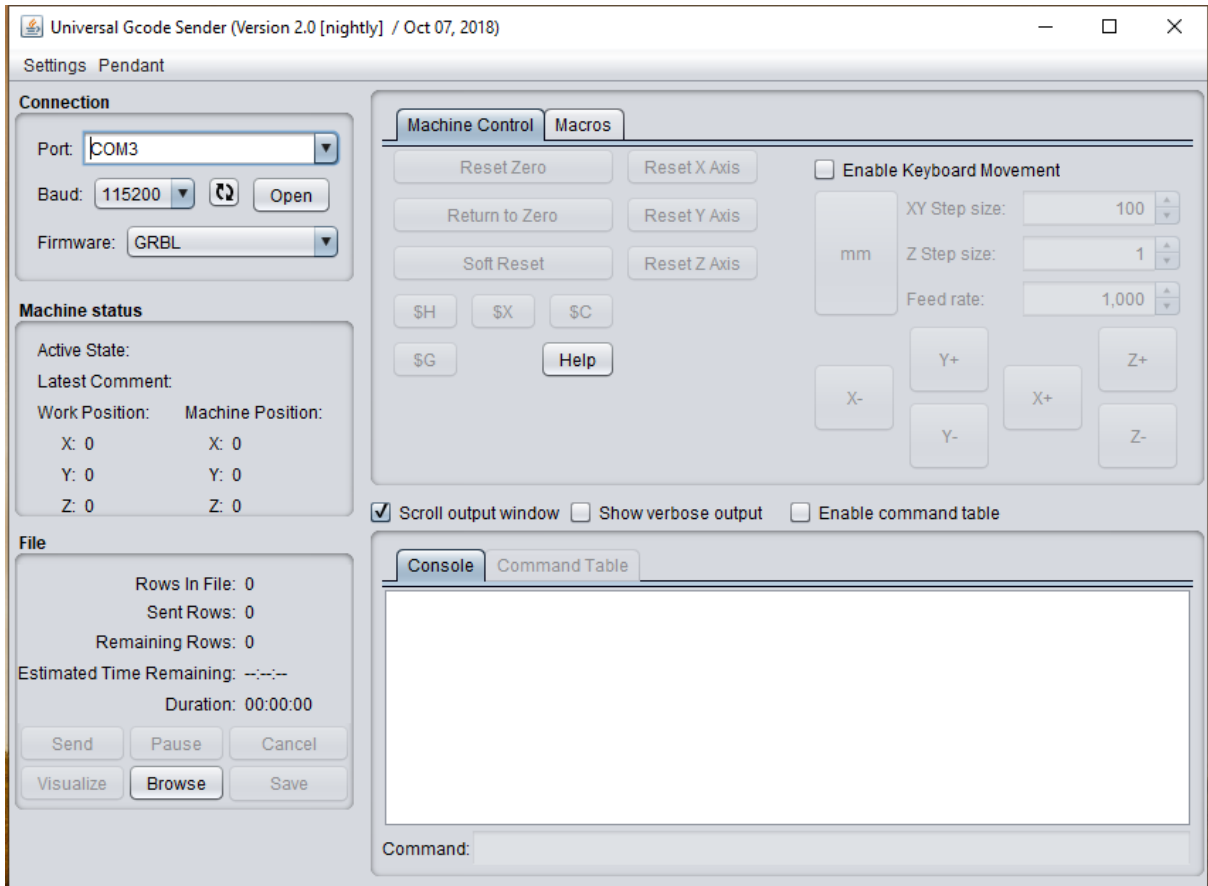
*Figura 4.4: Configuración de puerto y tarjeta Arduino.*

Por último se da clic en el botón upload y se espera a que termine el proceso.



*Figura 4.5: Carga del firmware GRBL en el Arduino UNO.*

Después de la instalación del firmware en el Arduino se necesita configurarlo con las especificaciones para la máquina para ello se utiliza un interfaz libre llamada UniversalGCodeSender que está desarrollada en Java [14].



**Figura 4.6:** Vista software Universal Gcode Sender.

Se conecta el Arduino al ordenador, se cargan las siguientes configuraciones

**Tabla III.** Valores de las configuraciones del software GRBL.

Configuración	Valor	Descripción
\$0 =	10	(Step pulse time, microseconds)
\$1 =	25	(Step idle delay, milliseconds)
\$2 =	0	(Step pulse invert, mask)
\$3 =	3	(Step direction invert, mask)
\$4 =	1	(Invert step enable pin, boolean)

CAPÍTULO IV. TÉCNICAS DE PROGRAMACIÓN

\$5 =	1	(Invert limit pins, boolean)
\$6 =	0	(Invert probe pin, boolean)
\$10 =	0	(Status report options, mask)
\$11 =	0.010	(Junction deviation, millimeters)
\$12 =	0.002	(Arc tolerance, millimeters)
\$13 =	0	(Report in inches, boolean)
\$20 =	0	(Soft limits enable, boolean)
\$21 =	1	(Hard limits enable, boolean)
\$22 =	1	(Homing cycle enable, boolean)
\$23 =	2	(Homing direction invert, mask)
\$24 =	25.000	(Homing locate feed rate, mm/min)
\$25 =	1500.000	(Homing search seek rate, mm/min)
\$26 =	250	(Homing switch debounce delay, milliseconds)
\$27 =	1.000	(Homing switch pull-off distance, millimeters)
\$30 =	255	(Maximum spindle speed, RPM)
\$31 =	0	(Minimum spindle speed, RPM)
\$32 =	1	(Laser-mode enable, boolean)
\$100 =	72.730	(X-axis travel resolution, step/mm)

CAPÍTULO IV. TÉCNICAS DE PROGRAMACIÓN

\$101 =	72.730	(Y-axis travel resolution, step/mm)
\$102 =	250.000	(Z-axis travel resolution, step/mm)
\$110 =	1200.000	(X-axis maximum rate, mm/min)
\$111 =	1200.000	(Y-axis maximum rate, mm/min)
\$112 =	500.000	(Z-axis maximum rate, mm/min)
\$120 =	10.000	(X-axis acceleration, mm/sec <sup>2</sup> )
\$121 =	10.000	(Y-axis acceleration, mm/sec <sup>2</sup> )
\$122 =	10.000	(Z-axis acceleration, mm/sec <sup>2</sup> )
\$130 =	120.000	(X-axis maximum travel, millimeters)
\$131 =	61.000	(Y-axis maximum travel, millimeters)
\$132 =	10.000	(Z-axis maximum travel, millimeters)



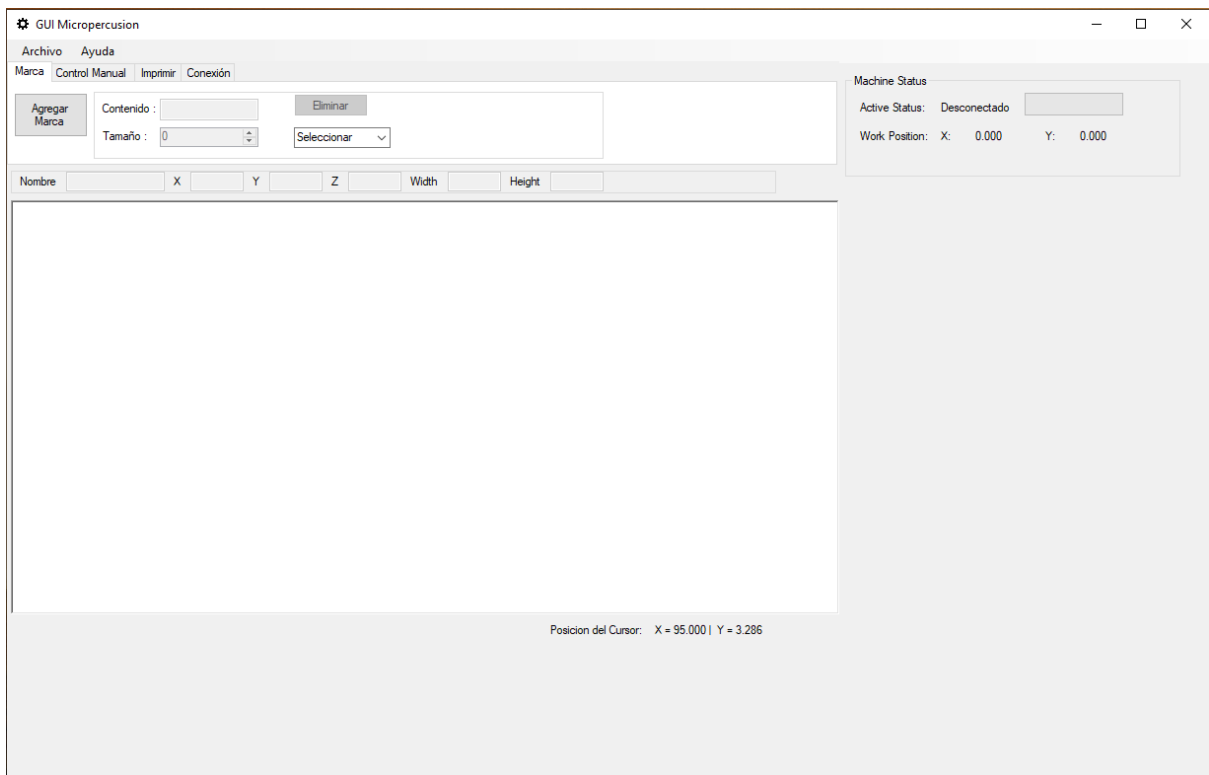
## 4.2 Interfaz de usuario (GUI)

La Interfaz es fundamental en toda aplicación y/o sitio web. Es la responsable de transmitir o hacerle saber al usuario lo que es capaz de hacer el producto.

Para este proyecto se necesita desarrollar una interfaz de usuario especializada.

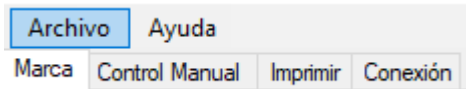
### 4.2.1 Pantalla principal

La siguiente figura muestra la pantalla principal de la interfaz.



**Figura 4.7:** Vista principal de la interfaz de usuario.

En ella se muestra el área de trabajo de la máquina junto con un cuadro donde se presenta el estado en tiempo real de la máquina, también se puede observar la barra de menú principal con los distintos módulos de la aplicación.

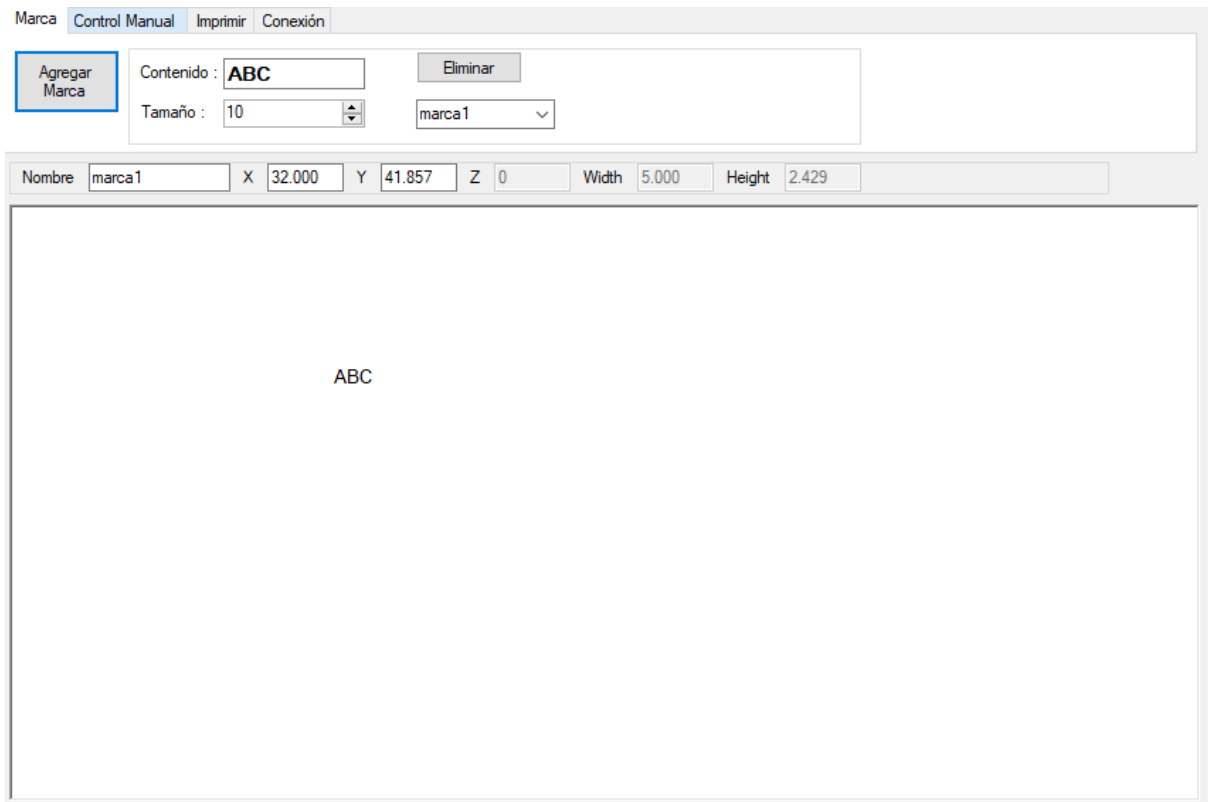


*Figura 4.8: Módulos de la aplicación.*

La aplicación cuenta con cuatro módulos los cuales son descritos a continuación.

## 4.2.2 Módulo marca

En módulo de Marca se puede acceder a la principal función del sistema, donde se agregan, configuran y eliminan las marcas que la máquina hace, al agregar una marca esta aparece en el área de trabajo de la máquina, donde se ajusta la posición deseada, al ser seleccionada una marca automáticamente aparecen sus propiedades en los respectivos cuadros.



*Figura 4.9: Módulo marca de la interfaz.*

### 4.2.3 Módulo control manual

En este módulo se muestran las funciones de movimiento de la máquina de forma manual, esto sirve para que el usuario pueda mover la máquina a una posición deseada o comprobar el área de trabajo. Las funciones de movimiento manual se activan una vez que se establece la comunicación con la máquina y se ha inicializado la misma.

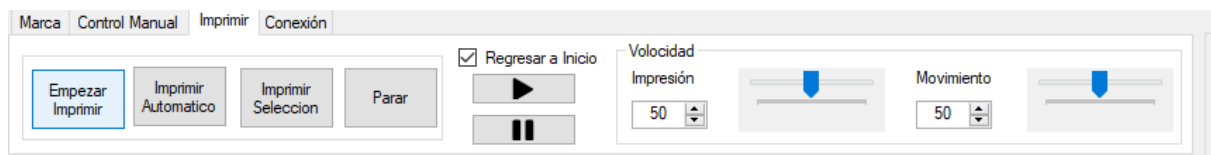


*Figura 4.10: Módulo control manual.*

### 4.2.4 Módulo imprimir

En este módulo se dan las opciones de impresión de la marca, donde se puede mandar a imprimir las marcas que se encuentran en el área de trabajo, un botón para mandar a imprimir automáticamente, este imprime de forma repetitiva lo que está en el área de trabajo hasta que desactive, también se cuenta con un botón para imprimir solo la marca que se tenga seleccionada y un botón para mandar desactivar la impresión automática.

También se muestran las configuraciones de velocidad de impresión y velocidad de movimiento de la máquina cuando no está imprimiendo.



*Figura 4.11: Módulo imprimir.*

## 4.2.5 Módulo conexión

En este módulo se configura la conexión con la máquina, en él se elige el puerto de comunicación y la velocidad de la comunicación.



*Figura 4.12: Módulo conexión.*

**CAPÍTULO V**  
**INTERFACES HARDWARE**

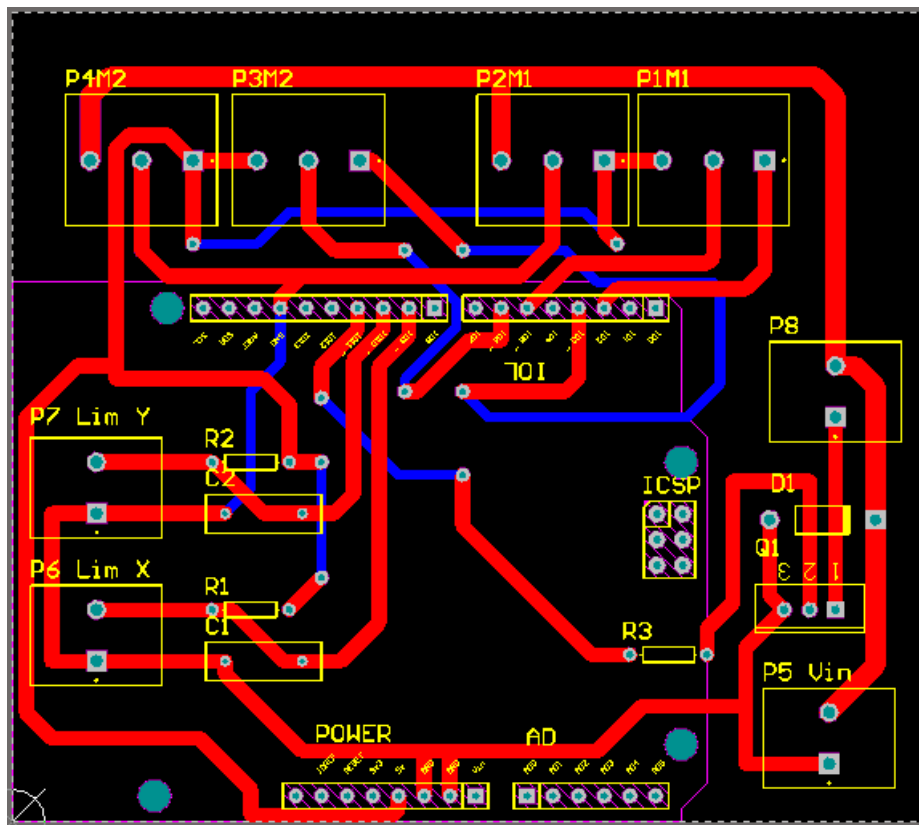
## 5. Interfaces hardware

El termino hardware se refiere a cualquier parte de la computadora que se puede tocar. El hardware consiste en dispositivos electrónicos interconectados que podemos usar para controlar la operación, así como la entrada y salida de la computadora. [15]

### 5.1 Placa controladora (PCB)

Se comienza por el diseño del PCB (printed circuit board) en el software Altium, en el cual se contemplan los requerimientos necesarios para el correcto funcionamiento del plotter, dejando espacio suficiente para futuras actualizaciones o mejoras.

Primero se contempla el área necesaria para incorporar los elementos, se toma en cuenta que la corriente que pasa por las pistas es hasta 1.5A, se elige un grosor de 1.5mm, se elige una baquelita de 90 x 80 mm. El diseño queda de la siguiente manera:

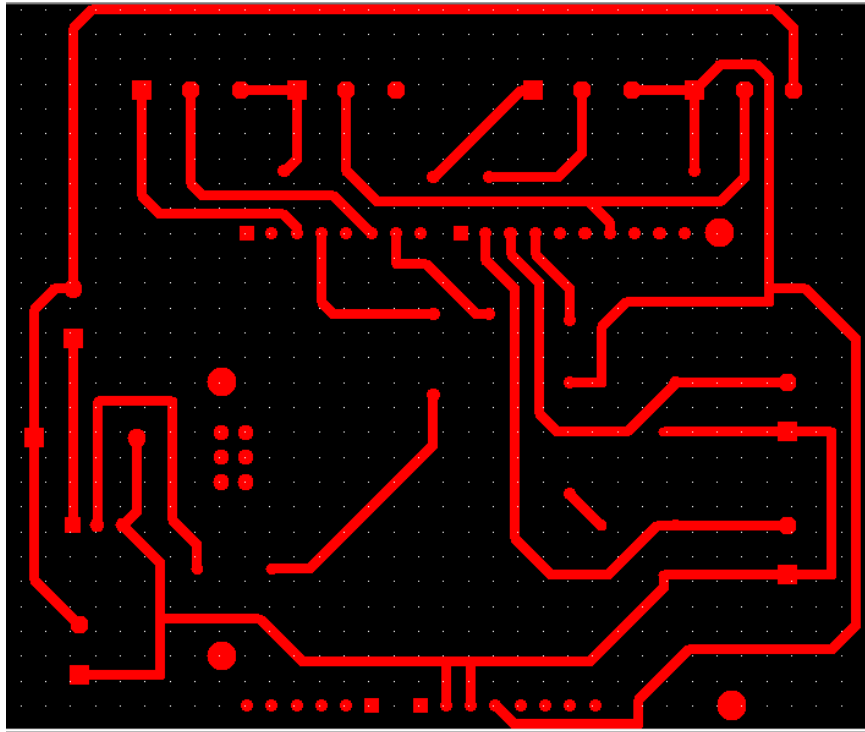


*Figura 5.1: Diseño del PCB en Altium.*

Después se asegura que los componentes queden con una separación óptima para su buen funcionamiento e identificación.

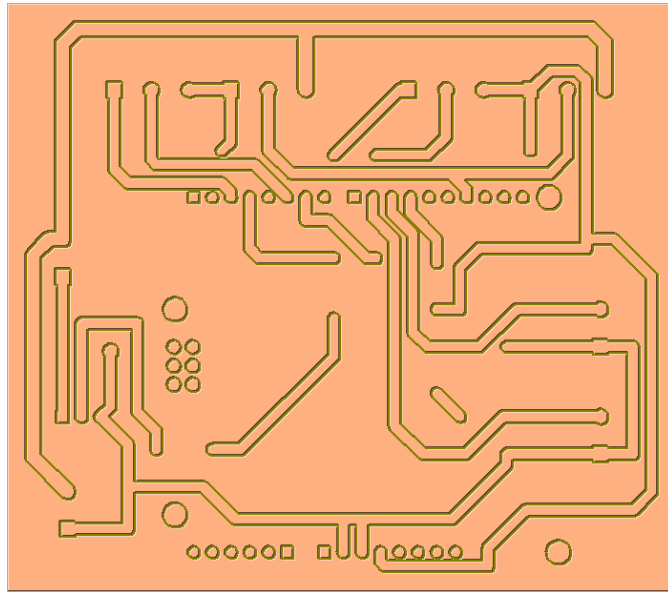
Para la elaboración del PCB se utiliza una baquelita comercial de 10cm x 10cm. Con un CNC se hace el ruteado de las pistas que se describe a continuación.

Primero se exporta el diseño de las pistas en Altium en un archivo Gerber.



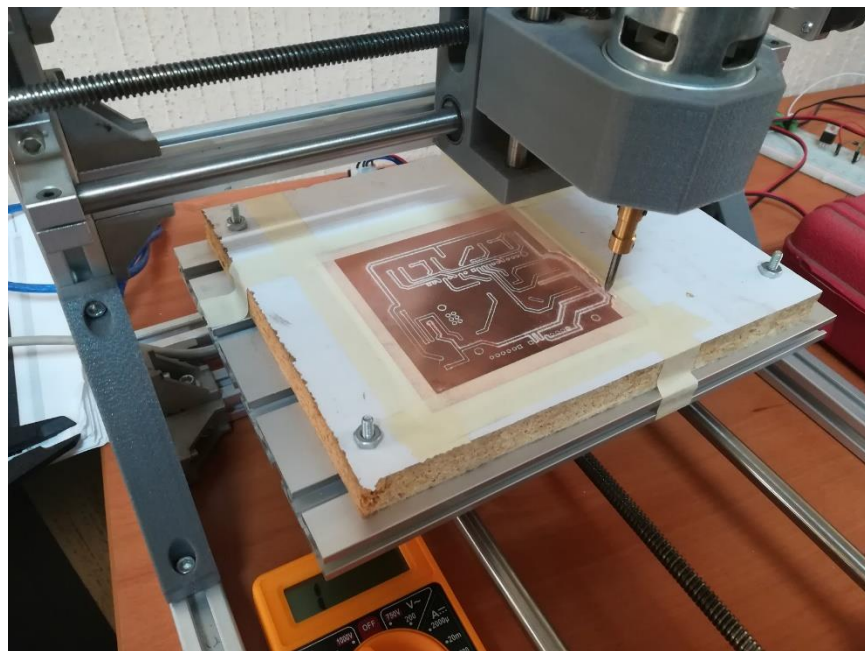
*Figura 5.2: Top layer del PCB.*

Con ayuda del programa CopperCam, se importa el archivo Gerber para generar el código G que interpreta el CNC y poder hacer el ruteado de las pistas.



*Figura 5.3: Imagen preliminar del ruteado del PCB*

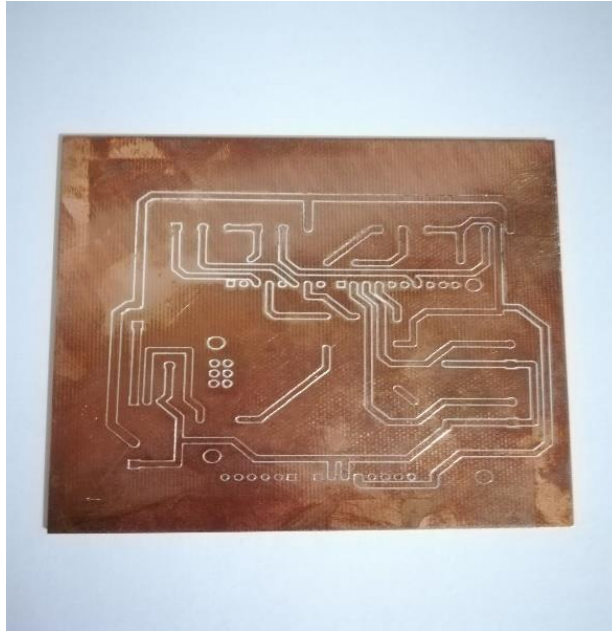
A continuación se procede con el ruteado de la baquelita.



*Figura 5.4: Rateado del PCB con ayuda de un CNC.*

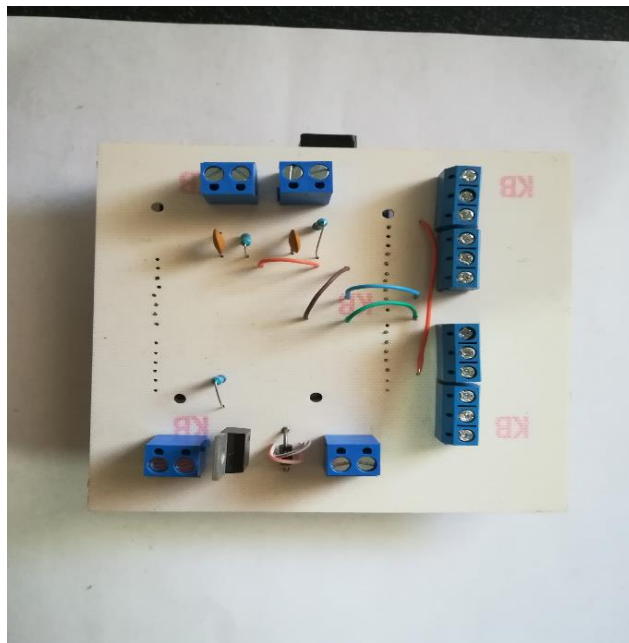
Una vez terminado el proceso de ruteado tenemos el siguiente resultado.





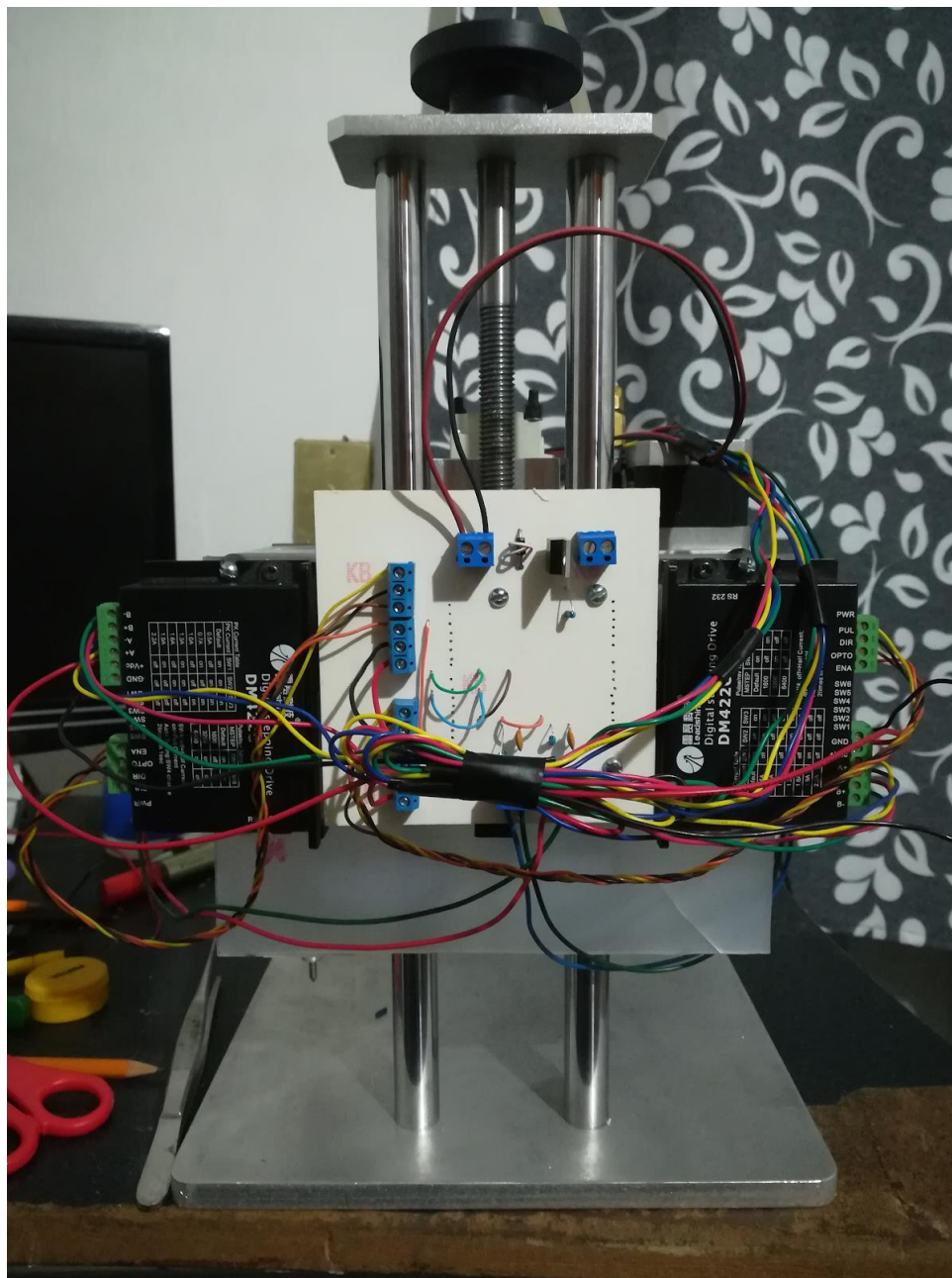
**Figura 5.5:** PCB ruteado.

Se comprueba el aislado de las pistas y se procede con la perforación y montaje de componentes.



**Figura 5.6:** Montaje de los componentes en el PCB.

Se sueldan los componentes y se comprueba su funcionamiento, para ello se monta en la máquina y se conecta el cableado.



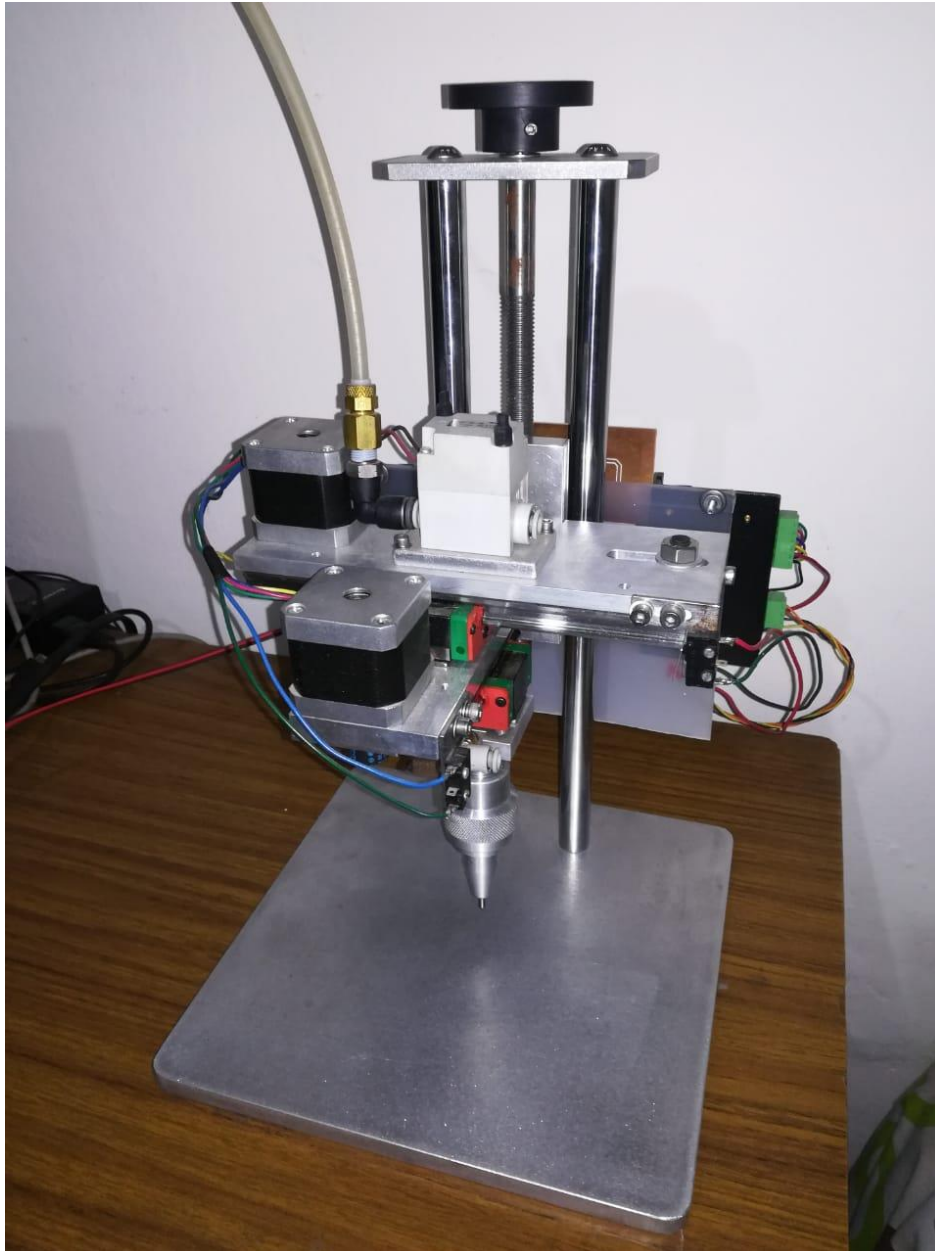
*Figura 5.7: PCB montado y conectado.*

En la Figura 5.7 podemos apreciar la placa desarrollada montada sobre la máquina, se conecta el cableado y se prueba el funcionamiento, dando un resultado positivo.

# **CAPÍTULO VI**

# **RESULTADOS**

## 6. Resultados



*Figura 6.1: Ensamble final de la máquina.*

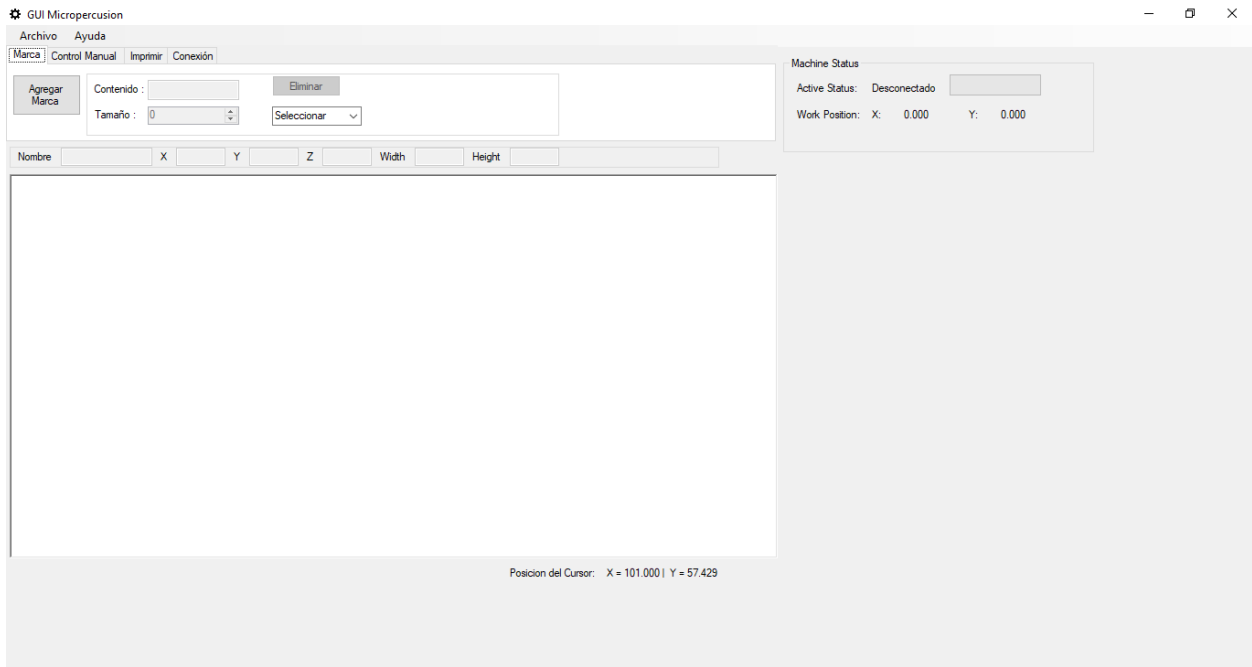
En la figura 6.1 se aprecia el ensamble final de la máquina.

## CAPÍTULO VI. RESULTADOS

El sistema queda funcionando de la siguiente manera, se tiene una interfaz gráfica en una computadora y por medio de un cable USB se conecta a la máquina.

La máquina necesita una fuente de voltaje a 24V.

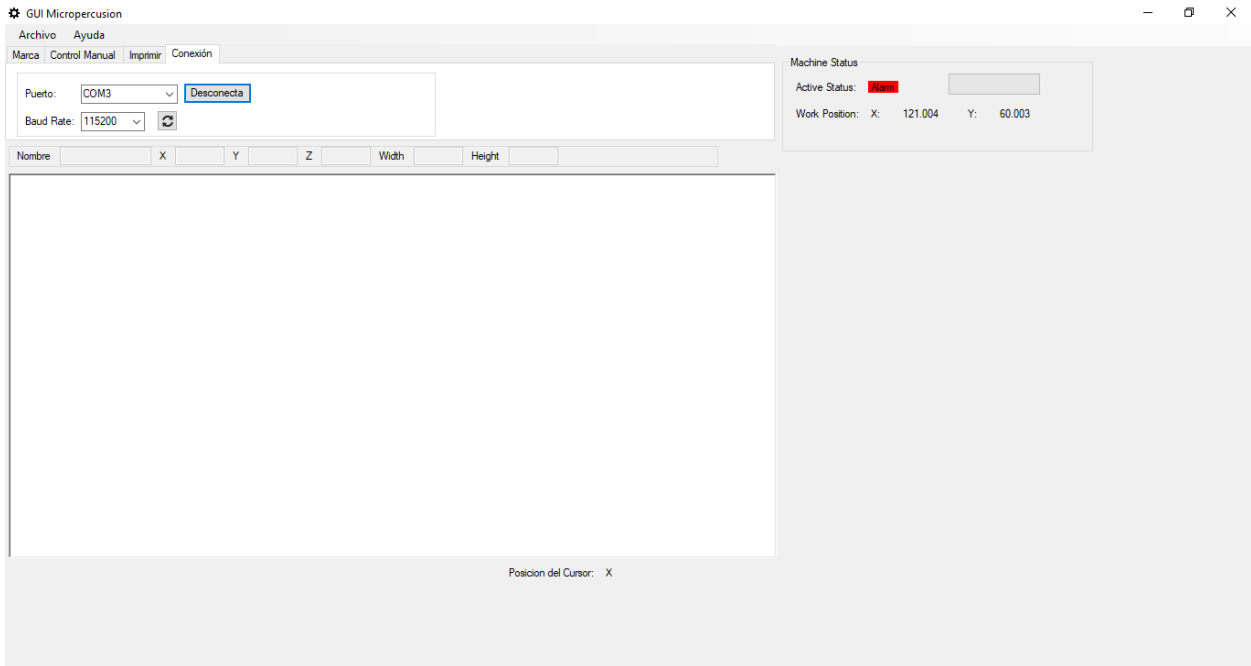
Para realizar una marca lo primero que debe hacer es abrir la aplicación.



**Figura 6.2:** Pantalla principal de la interfaz.

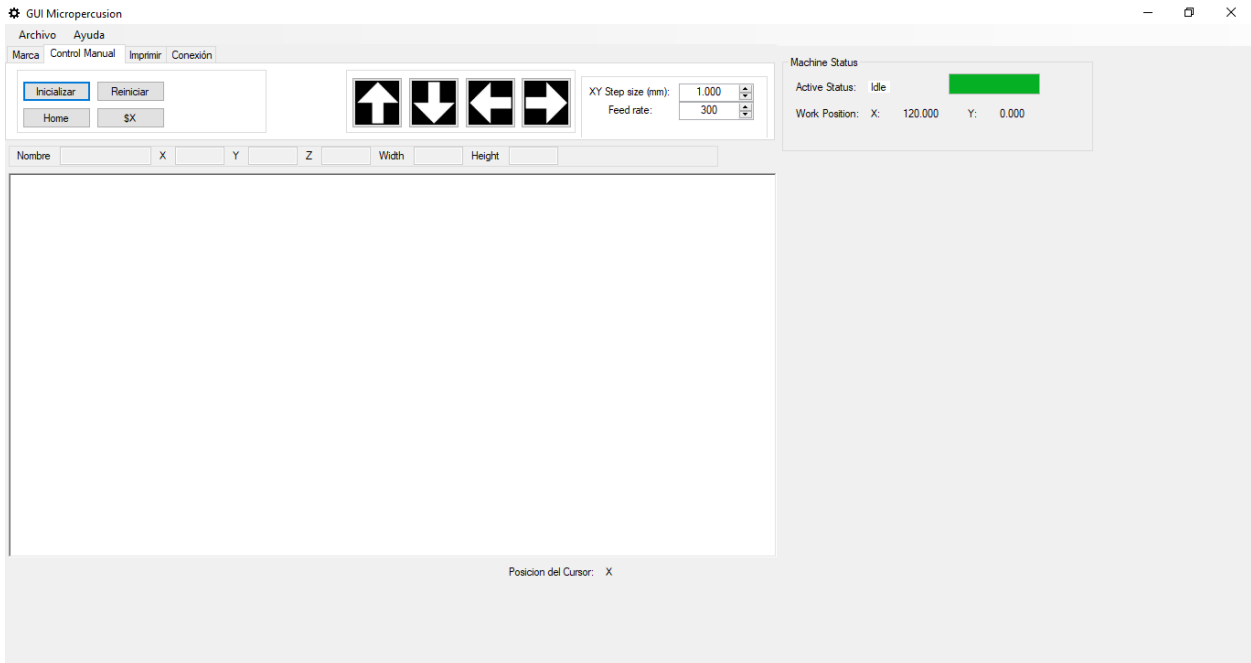
Una vez en la interfaz se dirige a la pestaña de conexión, se configura el puerto de la máquina y se presiona “conectar”, al elegir correctamente el puerto el status de la máquina será Alarm.

## CAPÍTULO VI. RESULTADOS



**Figura 6.3:** Conexión a la máquina.

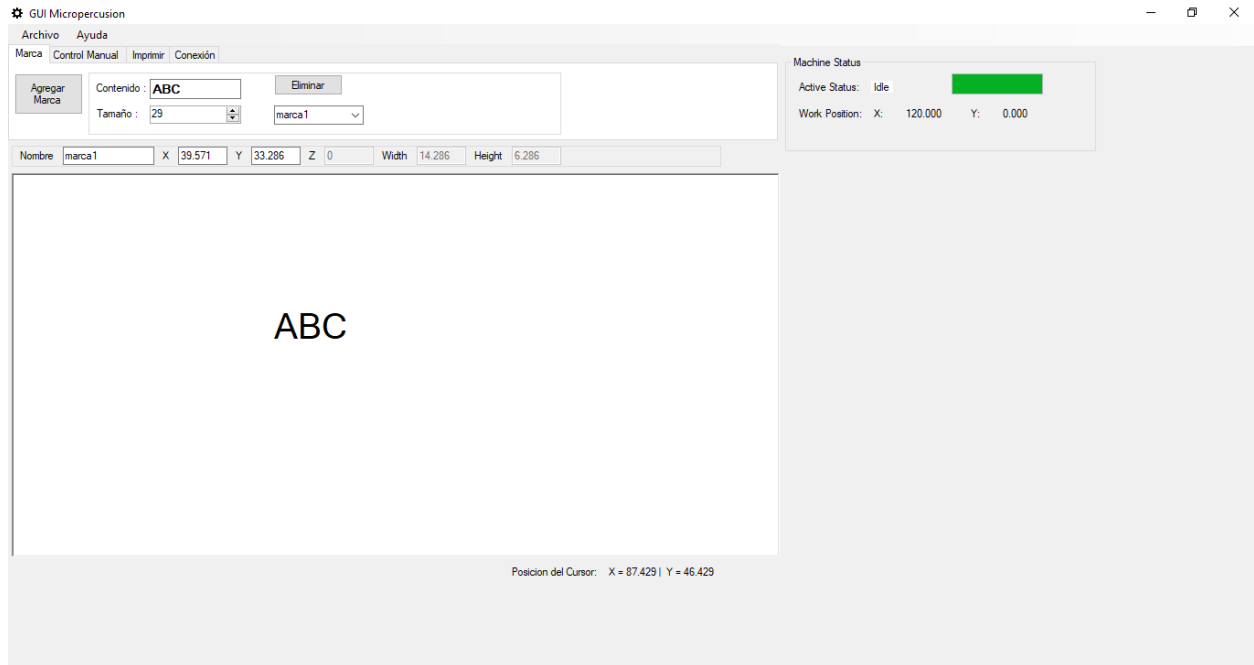
Posteriormente nos vamos a control manual y se presiona “Inicializar”, es necesario esperar a que la máquina termine el proceso.



**Figura 6.4:** Inicializado de la máquina.

## CAPÍTULO VI. RESULTADOS

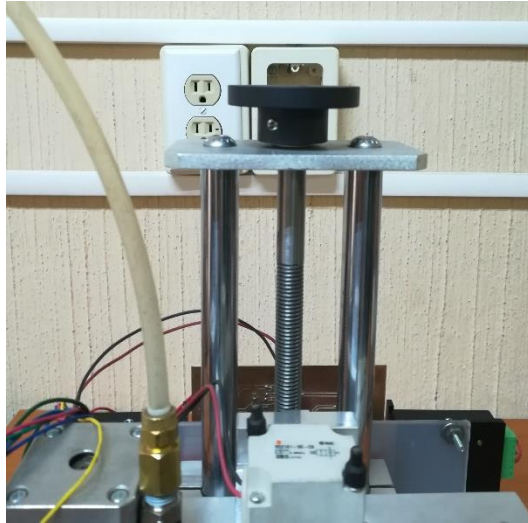
Ahora la máquina está preparada. Lo siguiente es agregar una etiqueta para marcar, se dirige a la pestaña de “Marca” y se presiona en “Agregar marca”, el cuadro blanco representa el área de trabajo de la máquina aparece una nueva etiqueta, se puede mover al dar clic sobre ella y mover el mouse, también se puede configurar el tamaño de la marca.



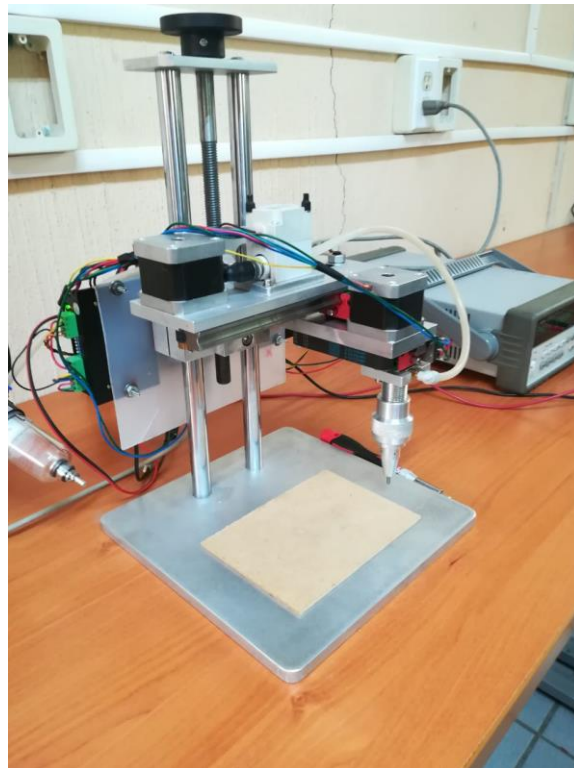
**Figura 6.5:** Agregado de una etiqueta y su configuración.

Una vez configurada la etiqueta es acta para imprimirla, para ello es necesario dirigirse a la pestaña “Imprimir”.

Antes de mandar a imprimir ajustamos la altura para la pieza que se marca con ayuda de la perilla manual de la máquina.



*Figura 6.6: Perilla de ajuste de altura o eje Z.*

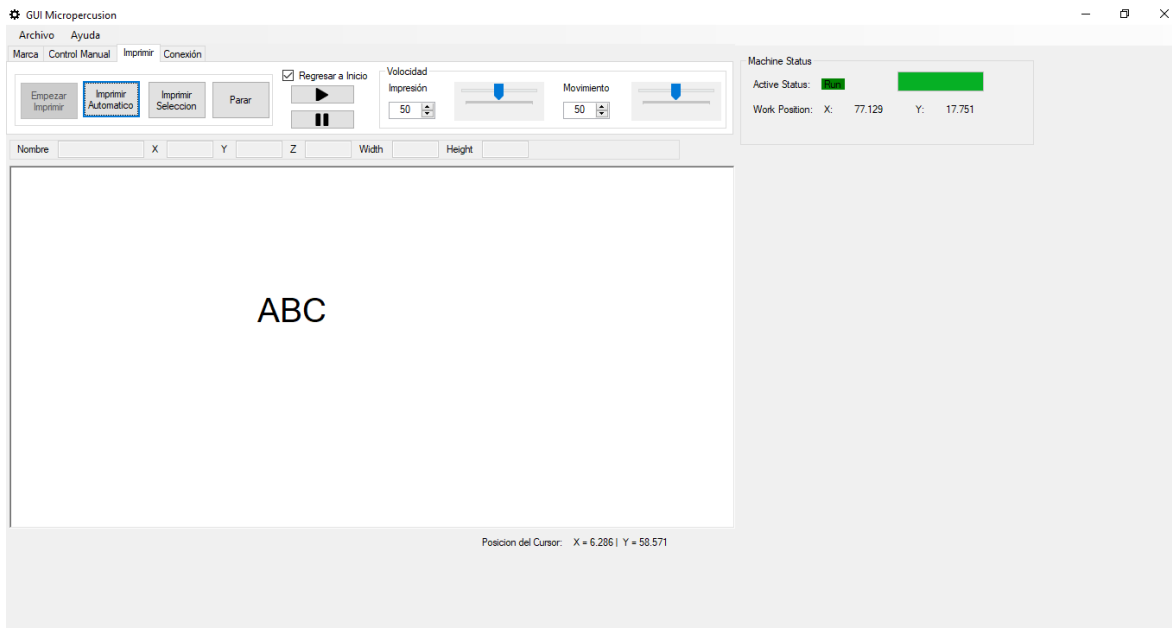


*Figura 6.7: Máquina con pieza a marcar.*

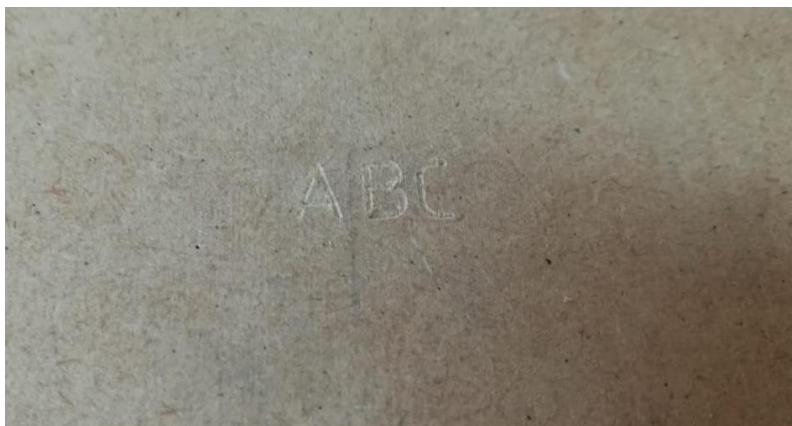
Una vez ajustada la altura para la pieza que se desea marcar se presiona el botón “Imprimir” y se observa el resultado.



## CAPÍTULO VI. RESULTADOS



*Figura 6.8: Manejo de impresión desde la interfaz.*



*Figura 6.9: Resultado de la impresión.*

En la figura 6.8 se observa el resultado de la marca, se utiliza un trozo de madera para esta prueba. Pero la maquina esta diseñada para diferentes tipos de materiales.

**CAPÍTULO VII**  
**CONCLUSIONES**

## 7. Conclusiones

En este trabajo se analizó, diseñó, desarrollo e implemento un sistema de control con interfaz de usuario para una máquina de micropercusión con la que se graban caracteres alfanuméricos.

La combinación del software GRBL y los drives digitales para los motores a paso DM422C, resultaron en una muy buena combinación con la que se obtienen un funcionamiento muy aceptable.

Se desarrolló una interfaz de usuario que permite el acceso para el control de la máquina, en la interfaz se agregan las marcas que se desean, se ajustan de acuerdo a los requerimientos, una vez hecho esto se manda a imprimir, la interfaz codifica estas marcas en un lenguaje de instrucciones que el Arduino con el software GRBL interpreta y ejecuta, GRBL implementa una serie de controles con los cuales se obtiene un resultado bastante aceptable.

La máquina es capaz de generar marcas de caracteres alfanuméricos en un área de trabajo de 60 mm x 115 mm. Tiene un ajuste manual sobre el eje Z con un recorrido de 150 mm. Estas especificaciones sirvieron para marcar una variedad de objetos y tipos de materiales.

Como recomendación se debe tener cuidado en el ajuste del eje Z, es de manera manual por lo cual una mala calibración resulta una marca deficiente o muy profunda. Se debe tener cuidado al manejar la máquina porque no cuenta con cubiertas de protección y trabajando puede ser peligrosa.

Para trabajos futuros se planea implementar un control de ajuste automático sobre el eje Z con el cual desde la interfaz se ajusta la profundidad de la marca; Se puede implementar un control sobre la frecuencia de marcado esto ayuda a tener marcas más definidas u optimizar el tiempo de marcado. Por último se puede mejorar la interfaz de usuario mejorando su aspecto, integrando el marcaje y decodificado de códigos en 2D e impresión de logotipos.

## CAPÍTULO VII. CONCLUSIONES

De manera general se desarrolló una interfaz de usuario muy intuitiva en C# la cual se enlaza con el Arduino, esta combinación y teniendo en cuenta que se está utilizando software libre (Open Source) para su desarrollo da como resultado una máquina de bajo costo y cumple con las necesidades que la empresa IP Industrial impuso.

## Referencias bibliográficas

- [1] Norton, R. L. (2013). *Diseño de maquinaria: Síntesis y análisis de máquinas y mecanismos* (5 ed.). México D. F.: McGraw-Hill.
- [2] A. Albert. Understanding CNC Routers, FPIInovations.
- [3] Teruel, F. C. (2011). *Control numérico y programación*. Barcelona, España: Alfaomega.
- [4] Lehmann, C. H. (1989). *Analytic geometry. D.F. México: LIMUSA*.
- [5] Ibrahim, D. (2007). *Programación de microcontroladores PIC*. Barcelona España: Marcombo.
- [6] *Arduino Uno Rev 3* . (2019). Obtenido de Store.arduino.cc:  
<https://store.arduino.cc/usa/arduino-uno-rev3>
- [7] Jennings, S. (2002). *Motores Por Paso*. Obtenido de  
[http://sapiens.itgo.com/motores\\_por\\_pasos](http://sapiens.itgo.com/motores_por_pasos)
- [8] Miguel, P. A. (2003). *Electrotecnia* (4 ed.). Madrid, España: Thomson.
- [9] Bucarely, C. M. (2005). Visual Basic 6.0. Grupo Experto Bucarely.
- [10] Miles, R. (2009). *C# Development*. Hull, Uk: Department of Computer Science, The University of Hull.
- [11] Moreno, F. (2000). *Introducción a la OOP*. Madrid, España: Grupo EIDOS.
- [12] Mejía, M. E. (2016). *Análisis estructural básico*. Bogotá, Colombia: Digiprint editores.
- [13] *Grbl. (1 de Mayo de 2017)*. Obtenido de GitHub: <https://github.com/grbl/grbl>
- [14] *Universal Gcode Sender*. (s.f.). Obtenido de ugs website:  
[https://winder.github.io/ugs\\_website/](https://winder.github.io/ugs_website/)
- [15] Díaz, C. y. (2003). Apuntes para la asignatura Informática I. D.F.: Fondo Editorial FCA.

# Anexos

## Anexo A

**gob.mx**

Instituto Nacional del Derecho de Autor  
 Dirección del Registro Público del Derecho de Autor

RPDA-01

Solicitud de Registro de Obra

Homoclave del formato RPDA-01	No. de trámite
Fecha de publicación del formato en el DOF 16      02      2016	Fecha de solicitud del trámite DD      MM      AAAA

Instrucciones: deberá llenar a máquina o con letra de molde legible, sin tachaduras o enmendaduras.

<input checked="" type="checkbox"/> Datos del autor	<input type="checkbox"/> Coautor	<input type="checkbox"/> Seudónimo
---	----------------------------------	------------------------------------

Datos generales	Domicilio particular
CURP (obligatorio): <b>RULC921201HCEZPRO8</b>	Código postal: <b>28975</b>
RFC: <b>RULC921201309</b>	Calle: <b>Coquimatlan</b>
Nombre(s): <b>Carlos Alberto</b>	Número exterior: <b>19</b> Número interior:
Primer apellido: <b>RUIZ</b>	Colonia: <b>López Mateos</b>
Segundo apellido: <b>López</b>	Municipio o Delegación: <b>Villa de Alvarez</b>
Sexo: <b>Masculino</b>	Estado: <b>Colima</b>
Fecha de nacimiento: <b>01      12      1992</b> <small>DD      MM      AAAA</small>	País: <b>Mexico</b>
Lugar de nacimiento: <b>Colima</b>	País: <b>Mexico</b>
Nacionalidad (opcional): <b>Mexicana</b>	País: <b>Mexico</b>
Porcentaje (%) y tipo de participación: <b>90 %</b>	País: <b>Mexico</b>
Teléfono (opcional): <b>312 3146518</b>	País: <b>Mexico</b>
Extensión (opcional):	País: <b>Mexico</b>
Teléfono móvil (opcional): <b>312 102 6485</b>	País: <b>Mexico</b>
Correo electrónico (opcional): <b>carl_557@hotmail.com</b>	País: <b>Mexico</b>

S.E.P.  
 DELEGACIÓN FEDERAL DE LA SECRETARÍA  
 DE FOMENTO RURAL Y PESQUERÍA A.I.S.C.O.  
  
**Ancl**

En cumplimiento de lo establecido en el artículo 17 de la Ley Federal de Procedimiento Administrativo, los formatos para solicitar trámites y servicios deberán publicarse en el Diario Oficial de la Federación (DOF).

MEXICO

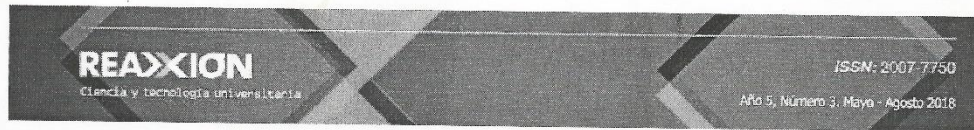
S.E.P.

Contacto:  
 Puebla #143, Col. Roma Norte, Deleg.  
 Cuauhtémoc  
 Distrito Federal, C.P. 06700 Tel:  
 55 5326 1820 y 55 5110 1183 400

Página 1 de 4

## Anexo B

Artículo: Modelado y simulación del automóvil en Matlab-Simulink-Carsim aplicado a la ... Page 1 of 9



[Inicio \(Index.html\)](#)

[Contenido \(NumeroenCurso.html\)](#)

[Autores \(Autores.html\)](#)

[Quiénes somos \(misvisvalo.html\)](#)

[Cartelera \(carteleraexterna.html\)](#)

[Contacto \(registro.html\)](#)

**Modelado y simulación del automóvil en Matlab-Simulink-Carsim aplicado a la estabilidad del vehículo.** [Ver pantalla completa / Imprimir \(Art Impr Modelado y simulacion del automovil en Matlab-Simulink-Carsim aplicado a la estabilidad del vehiculo.html\)](#)  
**Tecnológico Nacional de México-Instituto Tecnológico de Cd. Guzmán, México.**

Por: [Sergio Sandoval Pérez \(Cur Sergio Sandoval Perez.html\)](#), [Jorge Alberto Cárdenas Magaña \(Cur Jorge Alberto Cardenas Magana.html\)](#), [Emmanuel Vega Negrete \(Cur Emmanuel Vega Negrete.html\)](#), [Carlos Ruiz López \(Cur Carlos Ruiz Lopez.html\)](#), [Lucía Trinidad Pérez Cervera \(Cur Lucia Trinidad Perez Cervera.html\)](#), [Salvador González Palomares \(Cur Salvador Gonzalez Palomares.html\)](#)

### Resumen

Este artículo presenta el modelado matemático no lineal de un automóvil utilitario, tomando en cuenta la dinámica de la velocidad longitudinal, velocidad lateral y velocidad angular de viraje o yaw. Estas dinámicas se simularán en simulink con las normas ISO 3888-1 e ISO 7401 que servirán como plataforma para el uso del CarSim, el cual es un software que simula el comportamiento dinámico de cualquier tipo de automóvil y las simulaciones arrojadas son vistas en 3-D. Además de establecer estas simulaciones como herramientas de diseño para la industria automotriz.

**Palabras Clave:** Automóvil, Simulación, CarSim.

### Abstract

This article presents the nonlinear mathematical modeling of a utility vehicle, taking into account the dynamics of longitudinal velocity, lateral velocity and angular velocity of yaw. These dynamics will be simulated on Simulink with the norms ISO 3888-1 and ISO 7401 that will serve as a platform for the use of CarSim, which is a software that simulates the dynamic behavior of any type of automobile and the simulations thrown are 3-D views. In addition to establishing these simulations as design tools for the automotive industry.

**Keywords:** Automobile, Simulation, CarSim.

### Introducción

[http://reaxion.utleon.edu.mx/Art\\_Modelado\\_y\\_simulacion\\_del\\_automovil\\_en\\_Matlab-Si...](http://reaxion.utleon.edu.mx/Art_Modelado_y_simulacion_del_automovil_en_Matlab-Si...) 09/10/2018



En los últimos años, el desarrollo de una filosofía integral ha cambiado la forma de investigación en el área de diseño de nuevos dispositivos para el control del automóvil. Actualmente se cuenta con gran potencia computacional que permite trabajar con unidades de control en los vehículos, debido al mejoramiento de dispositivos electrónicos y al avance en la tecnología que permiten desarrollar nuevos sistemas de control que evitan cualquier restricción desde el punto de vista tecnológico. Considerando algunas variables físicas del automóvil, el diseño de un sistema de control para la velocidad lateral es uno de los principales tópicos de investigación en el área de control de automóviles.

Estos dispositivos modifican las dinámicas de los vehículos imponiendo fuerzas o momentos en el cuerpo del automóvil de diferentes maneras<sup>1,2,3,5,9,10,11</sup>, y también pueden utilizar "sensores inteligentes" que permiten mediciones precisas de las variables que se desean medir con alta eficiencia en la acción que el controlador esté realizando, lo que contribuye a la estabilidad del vehículo, la seguridad y confort para la persona que conduce<sup>7,13</sup>.

El movimiento de un vehículo puede ser categorizado en términos de su funcionalidad, así como de su maniobrabilidad y estabilidad. El funcionamiento está enfocado sobre el peso y los movimientos longitudinales del chasis; la estabilidad y las características de manejo generalmente se refieren a la respuesta lateral/maniobrabilidad de las acciones de conducción del automóvil que se requieran<sup>4,14</sup>. Una gran variedad de modelos de bajo, medio y alto orden están disponibles para describir el movimiento traslacional y rotacional del automóvil. Un modelo del vehículo de bajo orden de un grado de libertad (DOF por sus siglas en inglés) donde el automóvil considera una masa puntual y con ello, es suficiente para generar una plataforma de velocidad. Un modelo de dos grados de libertad es considerado de medio orden por su análisis<sup>12</sup>, teniendo en cuenta que las llantas frontales y traseras del vehículo son tomadas como una sola llanta delantera y una trasera (modelo de la bicicleta). En este modelo, en la llanta delantera se puede introducir la dinámica del conductor y del volante; además, los efectos del ángulo roll y de transferencia de masa son despreciados lo que permite estudiar la respuesta lateral/dirección del vehículo para pequeños ángulos de maniobrabilidad en velocidades longitudinales. Este artículo presenta el modelo matemático de alto orden y la transferencia de masa del mismo sistema; todos estos parámetros deben ser medidos para la completa realización a futuro de algunos diseños<sup>8</sup>. Esta medición de parámetros físicos se pueden realizar mediante la utilización de un software como el CarSim, el cual ofrece una amplia gama de soluciones tanto para el tratamiento de datos como para la simulación en una manera visual muy agradable; además de obtener gráficas de cada una de las variables físicas a medir, también considera las características del automóvil como la altura del mismo con respecto a un centro de gravedad, las longitudes frontal y trasera con respecto a las llantas y al centro de masa. Otra característica importante de este software llamado CarSim es que puede relacionarse con el programa Matlab-simulink, conocido por diseñadoras/es, investigadoras/es y estudiantes de ingeniería, obteniendo grandes aplicaciones a niveles de investigación tanto en educación como en la industria.

Una de las aplicaciones es la implementación de algoritmos de control para el automóvil y enviarlos al CarSim para simularlos y corroborar si son correctos, ahorrando tiempo y dinero, por lo que esta herramienta, a nivel mundial, es importante en el diseño automotriz. Este artículo de investigación comienza por la sección dos, donde se muestra el modelo matemático del vehículo a utilizar con sus dinámicas revisadas; en la parte tres, se establecen las simulaciones del comportamiento del modelo matemático del automóvil y se observa la interconexión del Matlab-Simulink con el CarSim y por último, se presentan las conclusiones y las referencias.

---

### Modelo Matemático del Vehículo

---



El modelo dinámico (Figura 1) presentado en esta sección es usado en las simulaciones numéricas para estimar valores del automóvil. En ecuación (1) se muestra el modelo matemático del vehículo utilizado para la simulación que considera tres grados de libertad; la velocidad longitudinal, la velocidad lateral y la velocidad angular yaw, pueden expresarse usando el llamado modelo de la bicicleta<sup>1</sup>;

(1)

$$\begin{aligned} m(\dot{v}_x - \omega_z v_y) &= a_x \\ m(\dot{v}_y + \omega_z v_x) &= a_y \\ j_z \dot{\omega}_z &= \mu(F_{f,y} l_f - F_{r,y} l_r) \end{aligned} \tag{1}$$

Donde  $m, j_z$  son la masa y la inercia del vehículo con respecto al eje perpendicular;  $x, y, l_f, l_r$  son las distancias desde el centro de gravedad del vehículo (CG) a la llanta frontal y trasera;  $a_x, a_y$  son la aceleración longitudinal y lateral;  $v_x, v_y$  son las velocidades longitudinal y lateral del CG;  $\omega_z$  es la velocidad angular yaw;  $\mu$  es el coeficiente de fricción. Además  $F_{f,y}, F_{r,y}$  son las fuerzas laterales frontales y traseras de las llantas respectivamente y se supone que se tiene un coeficiente de fricción alto entre las llantas y el suelo.

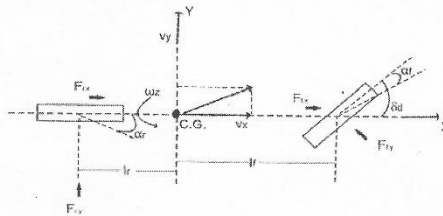


Fig. 1. Modelo no lineal de la bicicleta.

De (1) se obtiene el sistema dinámico;

(2)

$$\begin{aligned} \dot{v}_x &= \omega_z v_y + \frac{1}{m} a_x \\ \dot{v}_y &= -\omega_z v_x + \frac{1}{m} a_y \\ \dot{\omega}_z &= \frac{\mu}{j_z} (F_{f,y} l_f - F_{r,y} l_r) \end{aligned}$$

Este modelo describe las dinámicas del vehículo bajo las siguientes consideraciones:

- Dinámica pitch no considerada.
- El sistema es rígido.
- La suspensión es pasiva.

Las aceleraciones longitudinal  $a_x$  y lateral  $a_y$  en (2) pueden ser expresadas en términos de las fuerzas del neumático longitudinal y lateral;

(3)

$$a_x = \mu(F_{f,x} + F_{r,x})/m$$

(4)

$$a_y = \mu(F_{f,y} + F_{r,y})/m$$

Y pueden ser medidas por acelerómetros usualmente a bordo de automóviles modernos. En donde  $F_{r,x}$ ,  $F_{r,y}$  son las fuerzas longitudinales del neumático frontal y trasera. Las fuerzas longitudinal y lateral pueden ser definidas mediante la siguiente fórmula mágica de Pacejka;

(5)

$$F_{j,i} = D_{j,i} \sin(C_{j,i} \arctan(B_{j,i} \alpha_j))$$

Con  $i=x, y, j=f, r$ . Las constantes  $B_{j,i}$ ,  $C_{j,i}$ ,  $D_{j,i}$  en (5) son experimentalmente determinadas (ver Tabla 1). El ángulo de deslizamiento de los neumáticos  $\alpha_f, \alpha_r$  definidos como Di Cairano<sup>6</sup>;

(6)

$$\alpha_f = \delta_d + \delta_c - \frac{v_y + l_f \omega_z}{v_x}$$

(7)

$$\alpha_r = -\frac{v_y - l_r \omega_z}{v_x}$$

Donde  $\delta_d$  el ángulo de la rueda,  $\delta_c$  componente del ángulo de control impuesto a la llanta delantera.

**Tabla 1.** Parámetros reales usados en la simulación del vehículo.

$m = 1550 \text{ kg}$	$D_{r,y} = 7834 \text{ N}$
$j_z = 3552 \text{ kg m}^2$	$B_{f,y} = 6.9$
$l_f = 1.53 \text{ m}$	$C_{f,y} = 1.78$
$l_r = 1.38 \text{ m}$	$D_{f,y} = 7240 \text{ N}$
$B_{r,y} = 10$	$\mu = 0.9$
$C_{r,y} = 1.32$	

## Simulaciones

El comportamiento del modelo matemático del automóvil se simula (Simulink- Matlab-CarSim), bajo una maniobra ISO 3888-1, también conocida como "double change" con un ángulo  $\delta_d$ , que se muestra en la Figura 2.

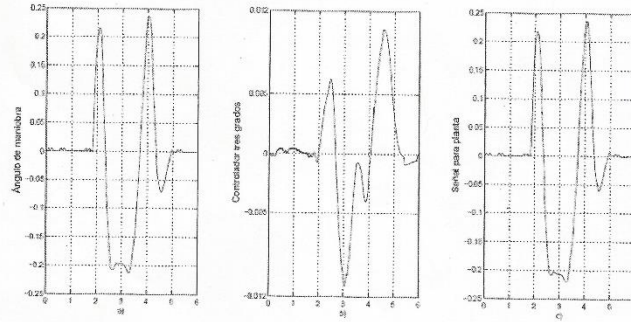


Fig. 2. a)  $\delta_d$  (rad vs s); b)  $\delta_c$  (rad vs s); c)  $\delta = \delta_d + \delta_c$  (rad vs s).

En este modelo matemático en ecuación (2), la simulación de la velocidad angular yaw controlada con respecto a un sistema de referencia se puede ver en la Figura 3.

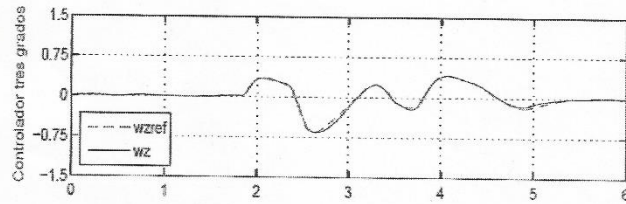


Fig. 3. Velocidad angular de viraje  $\omega_z$  (rad vs s).

La simulación de la velocidad lateral controlada con respecto a un sistema de referencia se muestra en la Figura 4.

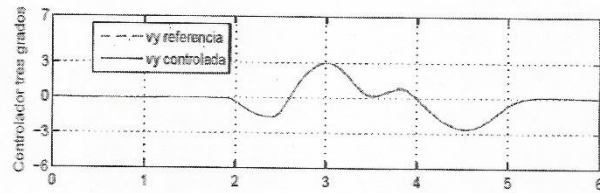


Fig. 4. Velocidad lateral  $v_y$  (m/s vs s).

**Maniobra ISO 7401**, también conocida como "Escalón" con un ángulo  $\delta_{d1}$  que se muestra en la Figura 5, la simulación de la velocidad angular en la Figura 6 y la simulación de la velocidad lateral en la Figura 7.

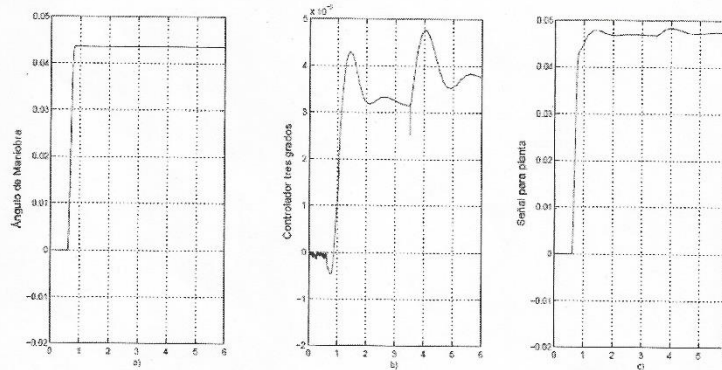


Fig. 5. a)  $\delta_d$  (rad vs s); b)  $\delta_c$  (rad vs s); c)  $\delta = \delta_d + \delta_c$  (rad vs s).

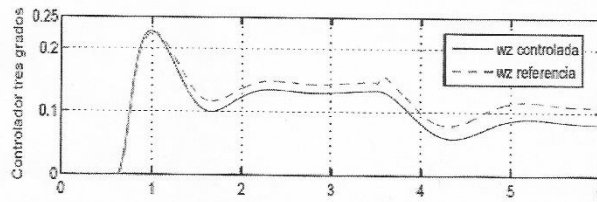


Fig. 6. Velocidad angular de viraje  $\omega_z$  (rad vs s).

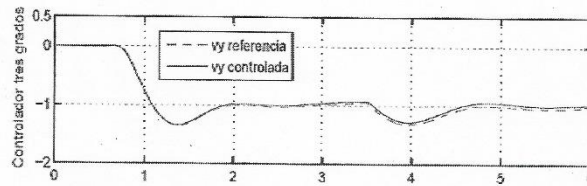


Fig. 7. Velocidad lateral  $v_y$  (m/s vs s).

La forma de interconectar el Matlab-Simulink con el CarSim, se observa en la Figura 8. El inicio del software CarSim ofrece la siguiente pantalla:



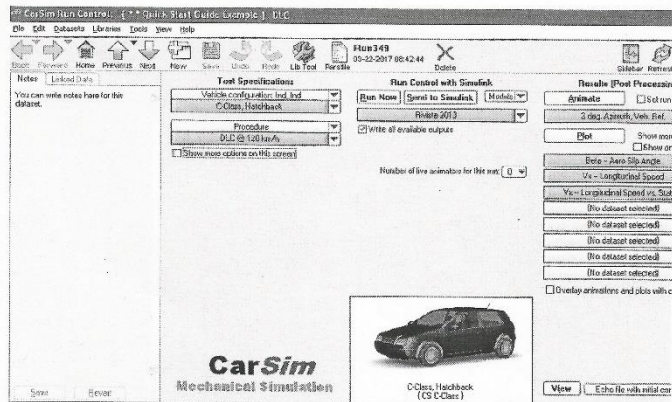


Fig. 8. Panel frontal del software CarSim-Simulink.

A continuación, se trabaja con la programación de los algoritmos del automóvil (análisis matemático), se empiezan a programar los blocks en Simulink, estos valores son los que se utilizan para el modelo matemático del vehículo (ver Figura 9), teniendo en cuenta que se envían al software CarSim para su evaluación como se muestra en Figura 10.

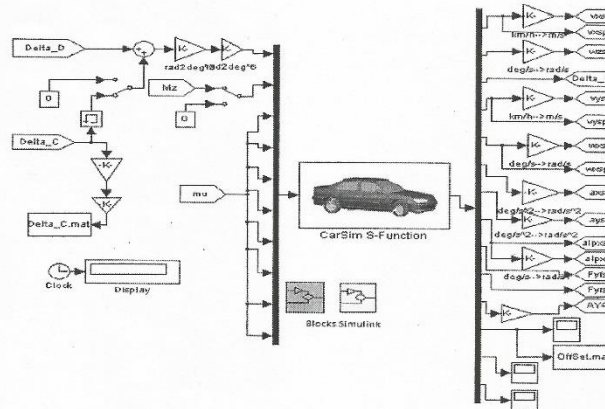
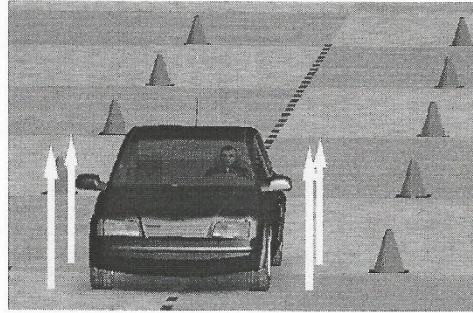


Fig. 9. Programación Simulink-CarSim.



**Fig. 10.** Simulación en software CarSim.

### Conclusiones

En este trabajo de investigación se propuso un modelo no lineal de tres grados de libertad para el automóvil aplicando un control activo en Matlab-simulink para obtener las dinámicas del vehículo controladas, comparando con un modelo de referencia que proporciona las variables del automóvil en condiciones ideales (llantas nuevas, suspensión nueva, etc.). Todos los algoritmos matemáticos tanto para el controlador como para el sistema de referencia no se analizaron porque se solo presentó el funcionamiento del software CarSim, dado que tiene una aplicación para simulación de los modelos matemáticos de controladores que se diseñan; se pudo observar que funcionan correctamente en la parte de aplicación física a futuro en el vehículo. También, por diferentes teorías de control se pueden proponer observaciones para las dinámicas del automóvil y aplicarlas al software CarSim y ver su funcionalidad. Se obtuvo como conclusión que a partir de un modelo matemático sencillo de control se pueden realizar aplicaciones al simulador y definir diseños para aplicaciones como pueden ser a direcciones asistidas para la estabilidad y maniobrabilidad en el automóvil. Por último, cabe mencionar que este tipo de investigaciones favorecen que el estudiantado mexicano se inicie en el campo del diseño automotriz para lograr los centros de investigación necesarios en la actualidad.

### Símbolos

$m$  Masa del vehículo ( $kg$ )

$j_z$  Momento de inercia del Vehículo ( $kg\ m^2$ )

$l_f, l_r$  Longitud desde el centro del vehículo hacia la llanta frontal y trasera ( $m$ )

$v_x$  Velocidad longitudinal ( $m/s$ )

$v_y$  Velocidad lateral ( $m/s$ )

$\omega_z$  Velocidad angular yaw del vehículo ( $rad/s$ )

$\alpha_f, \alpha_r$  Ángulo de deslizamiento lateral de las llantas frontal y trasero ( $rad/s$ )

$\delta_0$  Componente del ángulo de la llanta impuesto por el conductor (*rad*)

$\omega_z$  Velocidad angular yaw del vehículo (*rad/s*)

$F_{r,y}$ ,  $F_{r,y}$  Fuerzas frontal y trasera lateral de las llantas (*N*)

$F_{r,x}$ ,  $F_{r,x}$  Fuerzas frontal y trasera longitudinal de las llantas (*N*)

$a_x$ ,  $a_y$  Aceleración longitudinal y lateral (*m/s<sup>2</sup>*)

## Referencias

1. ACOSTA, C, DI GENNARO, S, SÁNCHEZ Maria. *An Adaptive controller applied to an anti-lock braking system laboratory*. Revista DYNA de la Universidad Nacional de Colombia, Vol. 83, 2016, p. 69-77. ISSN # 0012-7353.
2. ACKERMANN, J, GULDNER, J, STEINHAUSNER, R, UTKIN, V. *Linear and nonlinear design for robust automatic steering*. IEEE Transactions on Control System Technology, Vol. 3, 1995, No. 1, p. 132-143.
3. BASLAMISLI, S. C, POLAT, I, Kose, I. E. *Gain Scheduled Active Steering Control Based on a Parametric Bicycle Model*. IEEE Intelligent Vehicles Symposium, 2007, p.1168-1173.
4. BIANCHI, D, BORRI, A, DI BENEDETTO, M. D, DI GENNARO, S, BURGIO, G. *Adaptive Integrated Vehicle Control Using Active Front Steering and Rear Torque Vectoring*. International Journal of Vehicle Autonomous Systems, Special Issue on: "Autonomous and Semi-Autonomous Control for Safe Driving of Ground Vehicles, Vol. 8, 2010, No. 2/3/4, p. 85-105.
5. BURGIO, G, ZEGELAAR, P. *Integrated Vehicle Control Using Steering and Brakes*. International Journal of Control, Vol. 79, 2006, No. 2, 162-169.
6. DI CAIRANO, S, TSENG, H. E, BERNARDINI, D, BEMPORAD, A. *Vehicle Yaw Stability Control by Coordinated Active Front Steering and Differential Braking in the Tire Sideslip Angles Domain*. IEEE Transactions on Control Systems Technology, Vol. 21, No. 4, 2013, p. 1236-1248.
7. EARL Beal, C, CHRISTIAN GERDES, J. *Model Predictive Control for Vehicle Stabilization at the Limits of Handling*. IEEE Transactions on Control Systems Technology, Vol. 21, 2013, No. 4, p. 1258-1269.
8. HEYDINGER, G. J, GARROTT, W. R, CHRSTOS, J. P, GUENTHER, D. A. *Methodology for validating vehicle dynamics simulations*. SAE, Paper 900128, 1990.
9. KARBALAEI, R, KAZEMI, G. R, TABATABAEI, S. H. *Design of an Integrated AFS/DYC based on fuzzy logic control*. IEEE International Conference on Vehicle Electronics and Safety, 2007, p. 1-6.
10. MALAN, S, BORODANI, P, GORTAN, L. *Robust performance design for a car steering device*. Proceedings of the 33rd IEEE Conference on Decision and Control, 1994, p. 474-479.
11. REN, H, CHEN, S, LIU, G, ZHENG, K. *Vehicle State Information Estimation with the Unscented Kalman Filter*. Advances in Mechanical Engineering, Vol. 2014, 2014, p. 11-20.
12. YACINE, Z, ICHALAL, D, AIT-OUFROUKH, N, MAMMAR, S, DJENNOUNE, S. *Takagi-Sugeno Observers: Experimental Application for Vehicle Lateral Dynamics Estimation*. IEEE Transactions on Control Systems Technology, Vol. 23, 2015, No. 2, p. 754-760.
13. ZAKARIA, M, DWIJOTOMO, A, AZMAN, M, TAMALDIN, N. *Development Motor Control Unit for Electronic Steering System Test Rig*. 2014 IEEE 10th International Colloquium on Signal Processing and its Applications (CSPA), 2014, p. 42-47.
14. ZHAO, L, LIU, Z. *Vehicle State Estimation with Friction Adaptation for Four-Wheel Independent Drive Electric Vehicle*. Proceeding of the 11th World Congress on intelligent Control and Automation, 2014, p. 4527-4531.

Fecha de recepción

20/02/2018

Fecha de aceptación

10/04/2018

Fecha de publicación

31/05/2018

Año 5, Número 3. Mayo - Agosto 2018