



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRABAJO DE TITULACION

Propuesta tecnológica, previo a la obtención de Título de:

**INGENIERO EN ELECTRÓNICA Y
TELECOMUNICACIONES**

“Implementación de un sistema electromecánico braille, para ayuda
en la enseñanza y aprendizaje de personas no videntes”

AUTOR

Franklin Antonio López Silva

PROFESOR TUTOR

Ing. Sendey Agustín Vera González Mgtr.

La Libertad – Ecuador

2024

AGRADECIMIENTO

Agradezco a Dios por permitirme cumplir esta meta, darme la luz y la sabiduría para afrontar todas las dificultades que se pudieron presentar a lo largo de todos estos años, los cuales me han fortalecido para seguir adelante con mis propósitos.

A mis padres, por el apoyo incondicional y el sacrificio que han hecho, les agradezco por la confianza puesta en mí, su enseñanza me ha servido para ser una persona de bien y aportar a la sociedad mediante mis acciones.

A mi esposa y mi hija que han estado siempre pendientes de mí, han sido un pilar fundamental para seguir adelante, sus palabras me han dado la motivación para continuar con mis estudios y no decaer frente a los obstáculos.

A mi tutor, por haberme ayudado en el proceso de este proyecto y lograr finalizar con éxito gracias a su guía y recomendación.

A todos mis familiares más cercanos por formar parte de mi vida, les agradezco porque cada uno ellos me han dado siempre un mensaje de motivación y han sido también participes en mi proceso académico.

Franklin Antonio López Silva

DEDICATORIA

Este trabajo va dedicado a mis padres, Wilson López y Mirian Silva, que me han ayudado durante mi etapa estudiantil y me han brindado la confianza para seguir adelante, gracias a ellos he podido cumplir la meta de ser un profesional en esta carrera. A mi esposa Helena que me ha apoyado durante mi etapa final de estudio y gracias a ella he logrado finalizar con éxito este proyecto. También dedico este trabajo a mi hija Samara que es mi motor para avanzar cada día, sus palabras de confianza me alientan y me motivan para lograr el mejor porvenir para mi familia.

Franklin Antonio López Silva

APROBACION DEL TUTOR

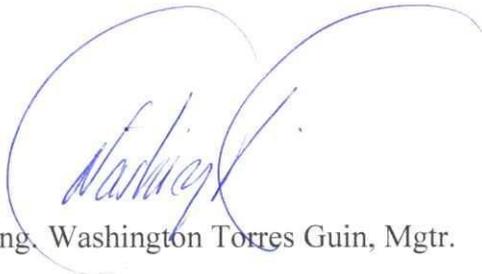
En mi calidad de tutor del trabajo de titulación denominado “**Implementación de un sistema electromecánico braille, para ayuda en la enseñanza y aprendizaje de personas no videntes**”, elaborado por el estudiante **López Silva Franklin Antonio**, de la carrera de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.

La Libertad, 4 de enero del 2024



Ing. Sendey Vera González Mgt

TRIBUNAL DE GRADO



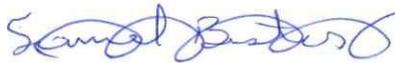
Ing. Washington Torres Guin, Mgtr.

Decano de la Facultad



Ing. José Sánchez Aquino, Mgtr.

Director de la Carrera



Ing. Samuel Bustos Gaibor, Mgtr.

Docente de Área



Ing. Sendey Vera González, Mgtr.

Docente Tutor



Ab. María Rivera González, Mgtr.

Secretaria General

UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES

“Implementación de un sistema electromecánico braille, para ayuda en la enseñanza y aprendizaje de personas no videntes”

AUTOR: Franklin Antonio López Silva

PROFESOR TUTOR: Ing. Sendey Agustín Vera González Mgtr.

RESUMEN:

En este trabajo se presenta la construcción de una impresora braille, utilizando tecnología de código abierto para personas con discapacidad visual. Las impresoras braille están disponibles en el mercado sin embargo el costo para acceder a ellas es elevado, por ende, este proyecto se presenta como alternativa para tener acceso a este tipo de tecnología y que sirva para ayudar a las personas no videntes. Se partió de un sistema de máquina de lenguaje de control numérico CNC al cual se envían Códigos G mediante un programa de código abierto Visual Studio, el cual recrea una interfaz sencilla y practica para ingresar texto y transformarlo en comandos de código G. Se utiliza una placa Arduino UNO, con un microcontrolador ATmega328P al cual se instala un firmware de código abierto grbl que sirve para traducir los comandos del código G en instrucciones para controlar el movimiento de dos motores de paso a paso, uno encargado de mover la hoja y el segundo para el desplazamiento del carrete en donde se instala una bobina electromagnética (solenoides) que es la responsable de realizar los puntos de relieve en código braille. Además, se utilizó una placa de control CNC Shield, en la cual es posible instalar dos drivers A4988 para el control de los motores paso a paso y el actuador solenoide. Como resultado se obtuvo una impresora braille, que cumple con el fin propuesto de generar textos con simbología braille para ayudar a las personas no videntes.

PALABRAS CLAVE: Impresora Braille, Discapacidad visual, firmware grbl, Arduino, Máquina CNC, Visual Studio.

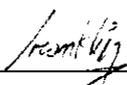
ABSTRACT

This work presents the construction of a braille printer, using open source technology for people with visual disabilities. Braille printers are available on the market, however the cost to access them is high, therefore, this project is presented as an alternative to have access to this type of technology and to help blind people. It started from a CNC numerical control language machine system to which G Codes are sent through an open source program Visual Studio, which recreates a simple and practical interface to enter text and transform it into G code commands. A Arduino UNO board, with an ATmega328P microcontroller to which an open source grbl firmware is installed that serves to translate the G code commands into instructions to control the movement of two stepper motors, one in charge of moving the blade and the second for the movement of the reel where an electromagnetic coil (solenoid) is installed, which is responsible for making the relief points in braille code. In addition, a CNC Shield control board was used, on which it is possible to install two A4988 drivers to control the stepper motors and the solenoid actuator. As a result, a braille printer was obtained, which meets the proposed purpose of generating texts with braille symbols to help blind people.

KEYWORDS: Braille Printer, Visual disability, Grbl firmware, Arduino, CNC machine, Visual Studio.

DECLARACION

El contenido del presente trabajo de graduación es de mi responsabilidad; el Patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.



Franklin Antonio López Silva

TABLA DE CONTENIDOS

AGRADECIMIENTO.....	I
DEDICATORIA	II
APROBACION DEL TUTOR.....	III
TRIBUNAL DE GRADO	IV
RESUMEN	V
ABSTRACT.....	VI
DECLARACION	VII
TABLA DE CONTENIDOS.....	VIII
INDICE DE FIGURAS.....	XI
INDICE DE TABLAS	XV
LISTA DE ANEXOS	XVI
INTRODUCCIÓN	1
CAPITULO I	2
FUNDAMENTACIÓN.....	2
1.1 ANTECEDENTES.....	2
1.2 DESCRIPCIÓN DEL PROYECTO.....	3
1.3 OBJETIVOS DEL PROYECTO.....	3
1.3.1 Objetivo General.....	3
1.3.2 Objetivos específicos	4
1.4 RESULTADOS ESPERADOS.....	4
1.5 JUSTIFICACIÓN.....	5
1.6 METODOLOGIA.....	7
CAPITULO II	8
PROPUESTA TECNOLÓGICA.....	8
2.1 MARCO CONTEXTUAL.....	8
2.2 MARCO CONCEPTUAL.....	9
2.2.1 Sistema Braille	9

2.2.2 Medios de impresión de códigos braille	14
2.2.3 Impresoras Braille	17
2.2.4 Control Numérico por Computadora (CNC)	20
2.2.5 Programación CNC	20
2.2.6 Lenguaje de programación CNC	21
2.2.7 Software de programación CNC	23
2.2.8 Controlador CNC	23
2.2.9 Arduino	24
2.2.10 Protocolos de comunicación Arduino	25
2.2.11 Actuadores eléctricos	27
2.2.12 Sistemas electromecánicos	27
2.2.13 Controlador de motor paso a paso	30
2.2.14 Solenoides	30
2.2.15 Diseño 3D	32
2.2.16 Impresión 3D	32
2.2.17 Impresora Ultimaker s5	33
2.2.18 Software para comunicación	33
2.2.19 GitHub	34
2.2.20 Interfaz de comunicación	34
2.2.21 Puerto de comunicación serie	34
2.3 MARCO TEÓRICO	35
2.4 COMPONENTES DE LA PROPUESTA	36
2.4.1 Componentes físicos	36
2.4.2 Componentes lógicos	47
2.5 DISEÑO DE LA PROPUESTA	52
2.5.1 Diagrama de conexiones de la impresora braille	52
2.5.2 Diseño Físico de la impresora braille	53
2.5.3 Diseño electrónico de la impresora braille	58
2.5.4 Diseño y desarrollo del software de control y comunicación	63

2.6 COSTOS	75
2.7 PRUEBAS Y RESULTADOS	77
2.7.1 Pruebas y resultados del análisis del código Braille	77
2.7.2 Pruebas y resultados de la revisión de dispositivos electrónicos existentes	81
2.7.3 Prueba y resultados de la identificación de herramientas de Hardware	83
2.7.4 Pruebas y resultados del análisis de Software para conversión de Texto a Braille	86
2.7.5 Pruebas y resultados del desarrollo del Sistema Electromecánico	90
2.7.6 Pruebas y resultados de funcionamiento.....	94
2.7.7 Pruebas y resultados de evaluación por personas no videntes	104
CONCLUSIONES	107
RECOMENDACIONES	109
BIBLIOGRAFÍA	110
ANEXOS	113

INDICE DE FIGURAS

Figura 1 Signo generador braille con los puntos numerados. Elaborado por el autor	9
Figura 2 Signo generador braille para lectura y escritura. Elaborado por el autor	10
Figura 3 Alfabeto Braille. Elaborado por el autor.....	10
Figura 4 Ejemplo para letra B. Elaborado por el autor	11
Figura 5 Números Braille. Elaborado por el autor.....	11
Figura 6 Vocales tildadas. Elaborado por el autor	12
Figura 7 Signos de puntuación. Elaborado por el autor	12
Figura 8 Dimensión braille estándar, hundido 0,5 mm.....	13
Figura 9 Regleta Braille 4 líneas por 28 celdas y punzón	14
Figura 10 Ejemplo escribir letra A. Elaborado por el autor	15
Figura 11 Escritura manual braille realizada con regleta y punzón	15
Figura 12 Maquina Perkins	16
Figura 13 Partes de Maquina Perkins	16
Figura 14 Escritura braille realizada con Maquina Perkins	17
Figura 15 ViewPlus Tiger Embraille.....	18
Figura 16 Index Basic D-V5	19
Figura 17 Index Everest D-V5	19
Figura 18 Maquina CNC para taladro	20
Figura 19 Tarjeta desarrollo ARDUINO UNO con ATmega328P	25
Figura 20 Protocolo de comunicación I2C	26
Figura 21 Interfaz maestro-esclavo SPI.....	26
Figura 22 Motor eléctrico DC	28
Figura 23 Motor paso a paso.....	28
Figura 24 Partes de un motor paso a paso	29
Figura 25 Conexión Unipolar y conexión bipolar de un motor paso a paso	29

Figura 26 Representación de campos magnéticos y corriente en solenoide.....	31
Figura 27 Solenoide lineal con muelle	31
Figura 28 Solenoide giratorio.....	32
Figura 29 Impresora Ultimaker S5	33
Figura 30 Motor paso a paso KS42SH40-1204A	37
Figura 31 Solenoide 24 V.....	38
Figura 32 CNC Shield V3	39
Figura 33 Driver A4988	40
Figura 34 Arduino UNO ATmega328P.....	42
Figura 35 Fuente de poder conmutada 24 VDC 5A.....	43
Figura 36 Tablero MDF 3mm	44
Figura 37 Ejes cromados DIAMETRO 6mm LARGO 400mm.....	45
Figura 38 Polea GT2 DIAMETRO 5 mm	45
Figura 39 Acople Flexible 6mm a 5mm.....	46
Figura 40 Rodamiento LM06UU DIAMETRO 6mm.....	46
Figura 41 Correa dentada GT2 ANCHO 6mm.....	47
Figura 42 Software IDE Arduino	48
Figura 43 Firmware grbl v 0.9.....	48
Figura 44 Software Universal Gcode Sender versión 2.1.2	49
Figura 45 Software Altium Designer.....	50
Figura 46 Software SolidWorks	50
Figura 47 Software Ultimaker Cura	51
Figura 48 Braille Gcode Generator	51
Figura 49 Esquema de conexiones. Elaborado por el autor	53
Figura 50 Diseño 3D impresora braille SolidWorks. Elaborado por el autor	54
Figura 51 Implementación impresora braille. Elaborado por el autor.....	54

Figura 52 Soporte lateral izquierdo y soporte lateral derecho. Elaborado por el autor	55
Figura 53 Estructura para solenoide. Elaborado por el autor	56
Figura 54 Diseño de base realizado con corte laser en MDF. Elaborado por el autor.....	56
Figura 55 Diseño de base en MDF. Elaborado por el autor	57
Figura 56 Instalación de motor sobre soporte izquierdo. Elaborado por el autor.....	57
Figura 57 Instalación de motor sobre soporte derecho. Elaborado por el autor	58
Figura 58 Circuito de potencia para el solenoide. Elaborado por el autor	59
Figura 59 Diseño esquemático del circuito de control. Elaborado por el autor.....	62
Figura 60 Circuito de control. Elaborado por el autor	62
Figura 61 Proceso de control y comunicación. Elaborado por el autor.....	63
Figura 62 Ventana programa. Elaborado por el autor.....	64
Figura 63 Ventana de archivo ZIP. Elaborado por el autor.....	64
Figura 64 Ventana Ejemplos. Elaborado por el autor	65
Figura 65 Librería compilada en la placa Arduino UNO. Elaborado por el autor.....	66
Figura 66 Interfaz gráfica UGS. Elaborado por el autor.....	66
Figura 67 Tabla de parámetros para calibración UGS. Elaborado por el autor.....	68
Figura 68 Punto Zero inicio de trayectoria en UGS. Elaborado por el autor	69
Figura 69 Interfaz Braille Gcode Generator (vista original). Elaborado por el autor.....	70
Figura 70 Diagrama de flujo (código general Braille Gcode Generator). Elaborado por el autor	71
Figura 71 Variables iniciales. Código C# Visual Studio	72
Figura 72 Parámetros iniciales Gcode. Código C# Visual Studio	72
Figura 73 Tipo colección Dictionary (Diccionario). Código C# Visual Studio	73
Figura 74 Actualización xOffset y conteo descendente -1. Código C# Visual Studio	74
Figura 75 Limites de fila. Código C# Visual Studio.....	74
Figura 76 Regleta Braille 29 x 4. Elaborado por el autor	78

Figura 77 Matriz braille posiciones del signo generador Elaborado por el autor.....	78
Figura 78 Errores ortográficos y gramaticales. Elaborado por el autor.....	79
Figura 79 Impresora de tinta en desuso. Elaborado por el autor.....	84
Figura 80 Maquina CNC para taladro	84
Figura 81 Generación de movimiento de ejes X, Y. Elaborado por el autor.....	89
Figura 82 Impresión de letra C con comandos G. Elaborado por el autor	89
Figura 83 Prueba de desplazamientos ejes X, Y. Elaborado por el autor.....	91
Figura 84 Prueba del punto de relieve. Elaborado por el autor.....	92
Figura 85 Comandos G(arriba), puntos en relieve (abajo). Elaborado por el autor.....	93
Figura 86 Puntos de relieve con máximas repeticiones. Elaborado por el autor	93
Figura 87 Puntos de relieve final. Elaborado por el autor.....	94
Figura 88 Alfabeto de minúsculas a imprimir. Elaborado por el autor	94
Figura 89 Alfabeto de minúsculas en código braille. Elaborado por el autor	95
Figura 90 Alfabeto de mayúsculas a imprimir. Elaborado por el autor	95
Figura 91 Alfabeto de mayúsculas en código braille. Elaborado por el autor.....	95
Figura 92 Números enviar a imprimir. Elaborado por el autor.....	96
Figura 93 Números en código braille. Elaborado por el autor	96
Figura 94 Vocales tildadas enviadas a imprimir. Elaborado por el autor.....	97
Figura 95 Vocales tildadas en código braille. Elaborado por el autor.....	97
Figura 96 Signos de puntuación y caracteres especiales. Elaborado por el autor.....	98
Figura 97 Signos de puntuación y caracteres especiales en código braille	98
Figura 98 Lectura evaluación de los caracteres braille Elaborado por el autor.....	104
Figura 99 Envío de texto a imprimir. Elaborado por el autor	104
Figura 100 Construcción de estructura de impresora braille.....	114
Figura 101 Ensamble final de componentes.....	114
Figura 102 Pruebas de funcionamiento	115

Figura 103 Pruebas de validación con persona no vidente	116
---	-----

INDICE DE TABLAS

Tabla 1 Lista de códigos G y M.....	22
Tabla 2 Especificaciones técnicas motor paso a paso.....	37
Tabla 3 Especificaciones técnicas Solenoide	38
Tabla 4 CNC Shield V3.	40
Tabla 5 Especificaciones técnicas driver A4988.	41
Tabla 6 Especificaciones técnicas TIP110	42
Tabla 7 Especificaciones técnicas Arduino UNO	43
Tabla 8 Especificaciones técnicas fuente alimentación 24VDC 5A	44
Tabla 9 Dimensiones impresora braille. Elaborado por el autor.....	55
Tabla 10 Relación de pines de Arduino UNO con placa CNC Shield. Elaborado por el autor ...	61
Tabla 11 Costos de materiales de estructura. Elaborado por el autor	75
Tabla 12 Costos de mano de obra. Elaborado por el autor	76
Tabla 13 Costos de materiales electrónicos. Elaborado por el autor	76
Tabla 14 Costo final. Elaborado por el autor	77
Tabla 15 Correspondencias Letras y números en braille. Elaborado por el autor.....	80
Tabla 16 Correspondencias signos de puntuación en braille. Elaborado por el autor.....	81
Tabla 17 Correspondencias signos de puntuación en braille. Elaborado por el autor.....	87
Tabla 18 Correspondencias signos de puntuación en braille. Elaborado por el autor.....	88
Tabla 19 Comandos G para letra C. Elaborado por el autor	89
Tabla 20 Pasos y grados obtenidos para distancias de la matriz braille.....	91
Tabla 21 Medidas braille recomendadas. Elaborado por el autor.....	99
Tabla 22 Errores absolutos y relativos primera distancia. Elaborado por el autor	99
Tabla 23 Errores absolutos y relativos segunda distancia. Elaborado por el autor	100

Tabla 24 Errores absolutos y relativos tercera distancia. Elaborado por el autor.....	100
Tabla 25 Errores absolutos y relativos cuarta distancia. Elaborado por el autor.....	101
Tabla 26 Errores absolutos y relativos quinta distancia. Elaborado por el autor	101
Tabla 27 Errores absolutos y relativos altura de relieve. Elaborado por el autor.....	102
Tabla 28 Medidas braille recomendadas. Elaborado por el autor.....	102
Tabla 29 Tiempo de impresión. Elaborado por el autor	103
Tabla 30 Nivel de satisfacción impresión de alfabeto mayúsculas.....	105
Tabla 31 Nivel de satisfacción impresión de alfabeto minúsculas	105
Tabla 32 Nivel de satisfacción impresión de vocales tildadas.....	106
Tabla 33 Nivel de satisfacción impresión de números	106
Tabla 34 Nivel de satisfacción impresión de signos de puntuación.....	106

LISTA DE ANEXOS

Anexo 1 Construcción y pruebas de impresora braille.....	114
Anexo 2 Planos de construcción.....	117
Anexo 3 Códigos Funcionamiento Braille Gcode Generator Visual Studio	121

INTRODUCCIÓN

El proyecto se basa en la implementación de un sistema electromecánico braille, para ayudar en la enseñanza y aprendizaje de personas no videntes, herramienta que será de gran utilidad y necesaria para ayudar a fortalecer la educación de las personas con discapacidad visual, en la actualidad las personas no videntes se alfabetizan por medio del sistema braille, los métodos de enseñanza de este sistema de lectoescritura se fundamentan en prácticas manuales poco convencional, considerando el avance tecnológico del siglo actual.

Según las investigaciones realizadas para este proyecto se identifica que la velocidad de escritura es lenta, esto retrasa el aprendizaje y la capacidad de mejorar las condiciones de las personas no videntes, la escritura con el sistema especial se hace símbolo a símbolo, por lo tanto, no existe otro método que sea más fácil y seguro. La investigación resalta que existe dificultad en el proceso de escritura, a consecuencia del sistema tradicional manual.

Como resultado de este trabajo, se obtuvo una impresora braille que genere escritura sobre papel en forma de relieve con punzadas claramente identificadas con su respectivo alfabeto. Aprovechando el avance tecnológico actual.

La investigación está enfocada a mejorar y fortalecer el acceso a la educación de las personas con discapacidad visual, la propuesta del diseño del sistema electromecánico será una alternativa práctica y de fácil acceso para mejorar tales circunstancias de discapacidad. Mejorar las condiciones de educación siempre será una prioridad aún más considerando este sector prioritario de la sociedad ecuatoriana.

CAPITULO I

FUNDAMENTACIÓN

1.1 ANTECEDENTES

En Ecuador, de acuerdo con el *Consejo Nacional para la Igualdad de Discapacidades* (CONADIS), se registra aproximadamente 455.289 personas con alguna discapacidad. Las personas con discapacidad visual representan el 11,47% de la población ecuatoriana, se considera un porcentaje considerable en relación a problemas, en respuesta a una necesidad prioritaria [1]. El primer sistema Braille se implementó en el año 1825 por Louis Braille de origen francés, quien sufrió un accidente que lo dejó ciego a la edad de 3 años y le permitiría perfeccionar la técnica de lectura y escritura desarrollado por Charles Barbier muchos años antes, que consistía en un sistema de relieve utilizando puntos y guiones. Braille trabajó en la reducción de ese sistema con el uso de una cuadrícula de seis puntos con dos columnas y tres filas que se leen de izquierda a derecha el mismo que se utiliza hasta el día de hoy [2].

En la actualidad las Universidades promueven alternativas de investigación en sus estudiantes con el fin de desarrollar nuevas herramientas que aporten al sector con discapacidad visual, donde la Universidad Politécnica Salesiana crearon un teclado mediante el cual se puede leer en braille el alfabeto a niños invidentes y permite ingresar palabras al ordenador al pulsar los botones. Su funcionalidad está controlada por un microcontrolador PIC 16F871, mediante la programación ejecutada en el mismo verifica el botón pulsado y a través de comunicación serial transmite el dato respectivo a un chip de voz el cual reproduce el audio que pertenece al botón pulsado [3].

Es necesario considerar que todos los proyectos direccionados a mejorar las condiciones de discapacidad visual, son alternativas que se proponen para aportar y fortalecer el sistema de aprendizaje de las personas no videntes del Ecuador y el mundo.

1.2 DESCRIPCIÓN DEL PROYECTO

El diseño de la impresora braille es de software Open Source y fundamenta su aplicación similar a una máquina CNC y permite crear un sistema electromecánico con el uso de una bobina electromagnética o solenoide para realizar puntos de relieve sobre papel de esa manera obtener una capacidad de impresión considerable y aceptable a los rangos dispuesto por la codificación braille.

Mediante esta implementación se pretende incorporar una herramienta que tiene como misión imprimir textos con simbología braille en textura de alto relieve para la lectoescritura que utilizan las personas con discapacidad visual y que sirve de ayuda al aprendizaje de código braille.

La operación de la impresora se ejecuta mediante un código en Arduino que se encarga de recibir los caracteres del computador mediante el programa Visual Studio, que envía la información de texto. El control de los motores utiliza una sección de drivers para ubicar espacialmente el cabezal en posición exacta, y el cabezal consiste de un solenoide para marcar los puntos de relieve que caracterizan la simbología braille sobre una hoja de cartulina Bristol A4, creando un documento físico con escritura braille para uso de lectura de las personas con discapacidad visual.

1.3 OBJETIVOS DEL PROYECTO

1.3.1 Objetivo General

Implementar una impresora braille mediante un sistema electromecánico de tecnología Open Source para transcribir textos con la finalidad de ayudar en la enseñanza y aprendizaje a las personas con discapacidad visual.

1.3.2 Objetivos específicos

- Estudiar el código braille simbología y conversión de cada letra o número y las estructuras ortográficas y gramaticales.
- Realizar un estudio de los dispositivos electrónicos que se utilizan para la educación de personas no videntes.
- Determinar las herramientas hardware y software necesarias para el desarrollo del sistema electromecánico que construya la impresora braille.
- Construir el sistema electromecánico de la impresora braille para impresión de una hoja con símbolos braille como material de ayuda para la enseñanza y aprendizaje de personas no videntes.
- Evaluar el funcionamiento de la impresora braille para validar el sistema con pruebas a personas no videntes.

1.4 RESULTADOS ESPERADOS

Con este proyecto se espera obtener los siguientes resultados:

- Obtener con claridad el conocimiento relacionado al sistema alfabético y números que se utilizan en el código braille y relacionarse con este tipo de lenguaje para la transcripción de textos.
- Adquirir información acerca de los diferentes dispositivos electrónicos que se utilizan para la educación no vidente, evaluar las ventajas y desventajas de cada uno, además de realizar el análisis de costos en el mercado.

- Obtener las herramientas y materiales para la construcción de la impresora, adicionalmente se estudia los programas a utilizar para desarrollar el control electrónico.
- Implementar el prototipo y desarrollar el software que controle el sistema de motores e interprete los datos obtenidos del tablero braille y transcriba sobre las hojas de papel, asociar el ordenador con la impresora para enviar textos a la impresora.
- Validar el funcionamiento correcto de la impresora braille y que se apruebe la debida corrección con la participación a las personas con discapacidad visual.

1.5 JUSTIFICACIÓN

La implementación de una impresora braille busca mejorar los métodos de enseñanza tradicional e incentivar la lectoescritura braille, de esta manera profundizar el estudio del sistema braille y así también reemplazar el punzón de papel y la regleta clásica por un sistema más sofisticado que cumpla con las características para satisfacer las necesidades actuales.

El presente proyecto es de gran importancia pues ayudará a personas no videntes a tener acceso a una impresora de bajo costo y poder en ella imprimir textos en braille, facilitando el trabajo que conlleva elaborar este tipo de escritos, logrando así mejorar los procesos de enseñanza y aprendizaje.

Es posible debido a que se cuenta con el equipo tecnológico y recurso monetario que el autor del proyecto necesita para realizarlo, en cuanto al aspecto tecnológico se eligió la placa controladora CNC Shield V3 y Arduino UNO, la primera extiende las características de Arduino para proporcionar las funciones de control de una maquina CNC se conectan entre sí montando el Shield sobre el Arduino. Se utiliza un cable USB como puerto serial conectado al Arduino para instalar el código GRBL en el microcontrolador ATmega 328P para interpretar el código G que luego envíe las señales de control a los drivers. La ventaja de estos componentes es su flexibilidad para organizar

y controlar sistemas electromecánicos desde el ordenador y facilitar la interacción entre usuario-maquina. El software y código se pueden descargar de manera gratuita y son de código abierto disponibles para maquinas CNC.

Dos motores paso a paso ayudan a generar los movimientos precisos para facilitar el manejo de la hoja sobre la que se genera la escritura braille. Un primer motor paso a paso es el responsable del movimiento del carrito de impresión y un segundo motor paso a paso controla el rodillo para el avance de la hoja. Dos controladores o drivers A4988 conectados al CNC Shield V3 son los que regulan el movimiento de los motores.

El elemento percutor es un componente muy importante en este proyecto, se usa un solenoide (bobina de alambre de cobre enrollada en un cilindro), muy conocido por su capacidad para crear un campo magnético en su interior cuando se le aplica corriente. El tipo de solenoide que se adapta al sistema es un solenoide lineal con carcasa metálica rectangular, el cual tiene un agujero en medio donde se ubica un émbolo, el mismo que al energizar se empuja hacia su interior, un resorte genera el retorno del émbolo a su posición inicial. Este tipo de solenoides son los menos complejos para trabajar y menos costosos.

A través de la herramienta Visual Studio, se mostrará una interfaz de usuario para las opciones de configuración y de ingreso de texto para imprimir, esta herramienta permite tener la accesibilidad entre el usuario y la impresora.

Posteriormente, la presente investigación dará apertura a futuros trabajos de investigación relacionados con la automatización de sistemas tradicionales en el campo educativo de las personas con discapacidad visual en la provincia y el país.

Finalmente, es necesario mencionar que el proyecto tendrá un impacto social muy importante en la provincia, ya que facilitará el acceso a la lectoescritura braille, así también, se convertirá en un componente necesario para los docentes que trabajan con personas invidentes. De esta forma contribuye en el desarrollo académico e inclusión de personas con discapacidad visual.

1.6 METODOLOGIA

El presente trabajo aborda los siguientes tipos de investigaciones:

Investigación explicativa

La metodología que se va a implementar para la impresora braille se utilizaran materiales electrónicos con sistemas de control digital, desarrollo de software Open Source. Esta investigación aportará a mejorar propuestas futuras en temas tecnológicos relacionados al aporte investigativo.

Investigación de campo

Durante el proceso de la metodología de campo se analizaron diversos prototipos de impresoras braille que a la vez cada una de ellas tiene métodos poco convencionales que en la actualidad se siguen utilizando y que de alguna manera son de utilidad para las personas con discapacidad visual. Una vez experimentado el manejo de los elementos para escritura braille pude palpar que se necesita trabajar en un sistema más técnico.

Investigación diagnostica

Durante el proceso de la impresora braille se analizó el mecanismo más factible y funcional para desarrollar el proceso, determinar el funcionamiento óptimo y de calidad para las funciones que se plantean elaborar.

CAPÍTULO II

PROPUESTA TECNOLÓGICA

2.1 MARCO CONTEXTUAL

La discapacidad visual se define como una condición que afecta la percepción de las imágenes de forma parcial o total. El sentido de la vista nos permite identificar los objetos a distancia que ya se conocen o que se presentan por primera vez [4].

Las estadísticas de Censo de la población en Ecuador, determinó que el 13,2% son personas algún tipo de discapacidad y representan un total de 1'600.000 personas de las cuales 363.000 personas sufren deficiencias visuales. El método de enseñanza a estas personas se basa en una combinación de estímulos auditivos y táctiles. Los estudiantes no videntes utilizan el sistema de lectoescritura braille desde 1825. En Ecuador este sistema se ha utilizado alrededor de 50 años, pero no se lo aplica en niveles básicos, se utiliza en mayor medida la palabra y las actividades participativas para transmitir el conocimiento que estimulan el tacto y el sistema auditivo de los niños con discapacidad visual [5].

Considerando la actualidad y los avances tecnológicos es imprescindible el código braille como método de enseñanza, sin embargo, hay que considerar que la alfabetización implica mucho más que el aprendizaje del código. Por lo tanto, será esencial desarrollar un sistema tecnológico actualizado y con mayor capacidad de procesamiento de información para personas con discapacidad visual.

2.2 MARCO CONCEPTUAL

Es fundamental detallar los conceptos previos que comprenden el desarrollo del proyecto se mencionan a continuación.

2.2.1 Sistema Braille

El código braille es un sistema de lectura y escritura usado por las personas no videntes o con enfermedades visuales, es una representación de símbolos alfabéticos y numéricos se basa en una matriz de seis puntos táctiles, se dispone de dos columnas con tres filas. Con este método se puede representar letras, números, signos de puntuación, símbolos matemáticos e incluso música.

La creación de los caracteres se compone de un matriz de dos columnas y tres filas que forman el signo generador, se puede combinar estos seis puntos para obtener los distintos caracteres, se obtienen 64 combinaciones. Para identificar cada punto de la matriz que forman el signo generador están numerados según su posición.



Figura 1 Signo generador braille con los puntos numerados. Elaborado por el autor

El método de lectura en el sistema braille se realiza de izquierda a derecha mientras que la escritura se inicia de derecha a izquierda de manera que se invierte la posición de las columnas, tal como se muestra en la figura 2.

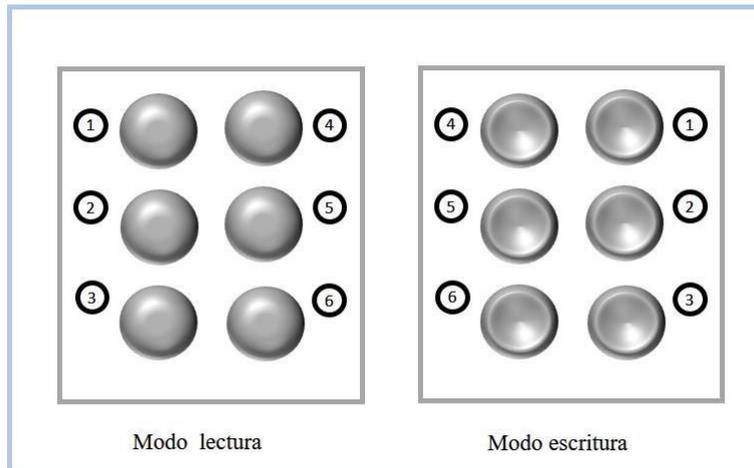


Figura 2 Signo generador braille para lectura y escritura. Elaborado por el autor

Alfabeto en Braille

Mediante la matriz braille se puede representar los diferentes caracteres del alfabeto como se puede ver en la figura 3, los puntos en negro son los puntos del signo generador que están en relieve.

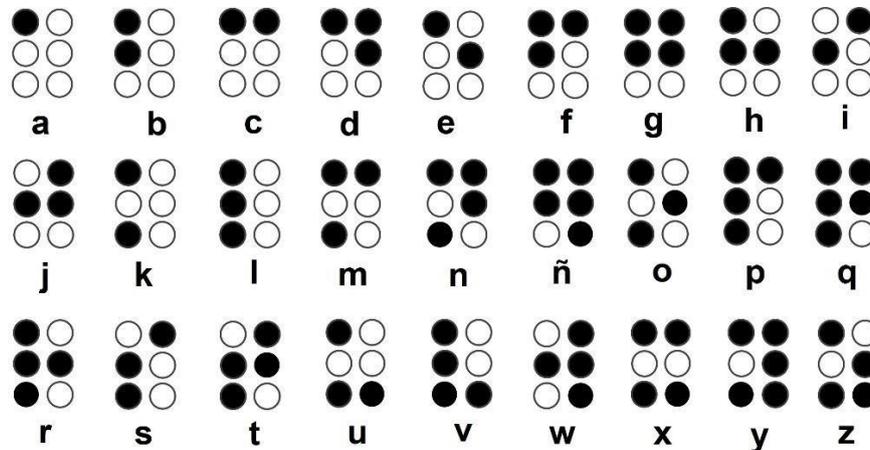


Figura 3 Alfabeto Braille. Elaborado por el autor

Las letras en mayúsculas en braille adquieren una característica diferenciadora, la cual se basa en dos matrices, la primera es el indicador de mayúscula y la segunda la letra [6]. Este diferenciador le da un carácter muy importante para no cometer un error ortográfico, por ejemplo, en caso de escribir al inicio de una oración o párrafo, los nombres, apellidos, después de un punto, después de puntos suspensivos o en caso de Siglas de instituciones o organizaciones como UNESCO o ONU.

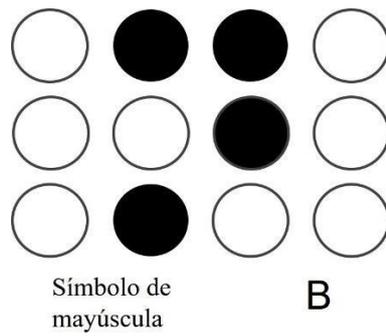


Figura 4 Ejemplo para letra B. Elaborado por el autor

Números en Braille

Los números se identifican por dos matrices, la primera es el símbolo de número y la segunda matriz representa cada número similar al patrón usado en las letras como se muestra en la figura 5. De esta manera se les añade una característica particular a los números en código braille.

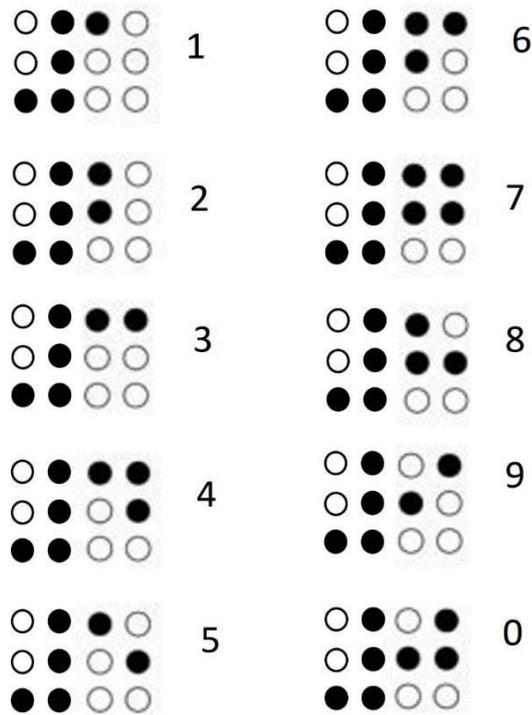


Figura 5 Números Braille. Elaborado por el autor

Vocales tildadas

En el caso de las vocales tildadas se representa mediante una combinación específica cada vocal en el sistema braille. Como en lenguaje es importante definir las tildes ya que son importantes para las reglas ortográficas definen el acento en las palabras e influyen en el sentido de las mismas.

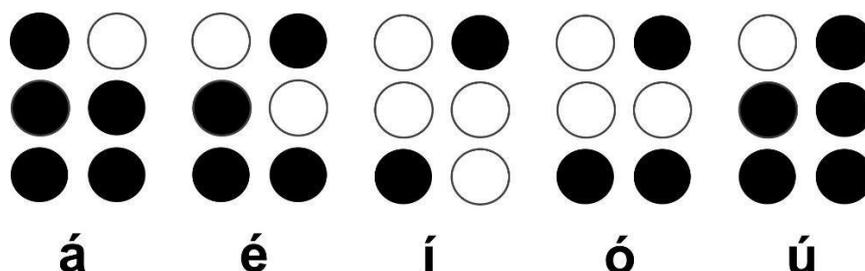


Figura 6 Vocales tildadas. Elaborado por el autor

Signos de puntuación

Las combinaciones que producen los signos de puntuación están conformadas de la siguiente manera.

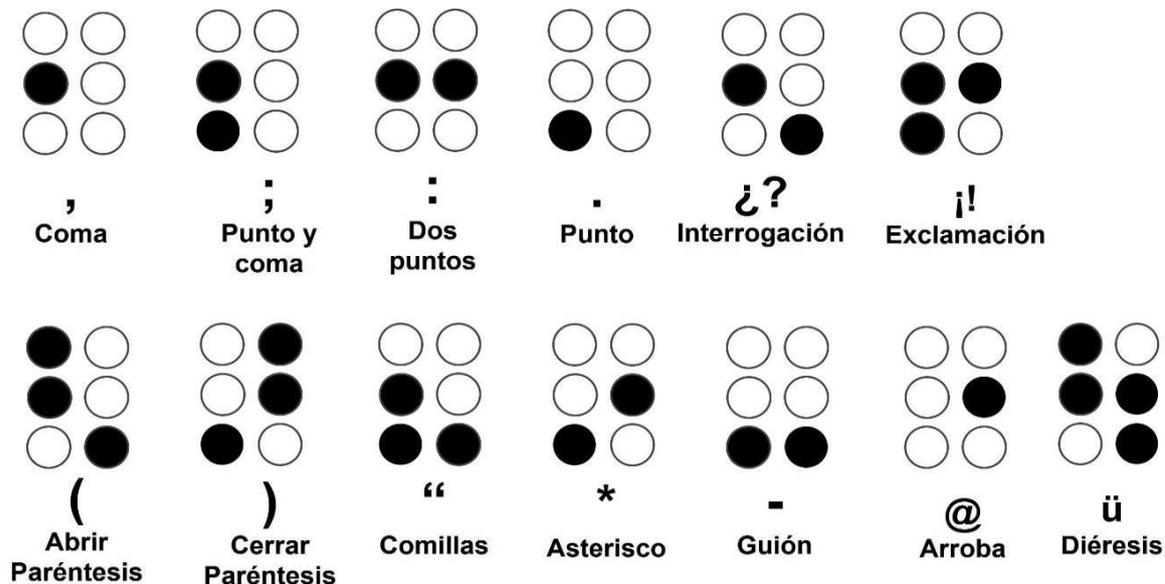


Figura 7 Signos de puntuación. Elaborado por el autor

Dimensiones braille

Una celda braille cumple con ciertos parámetros dimensionales y estas sirven para tener las medidas correctas para lectura de los caracteres. Estos parámetros permiten la lectura táctil de los caracteres braille.

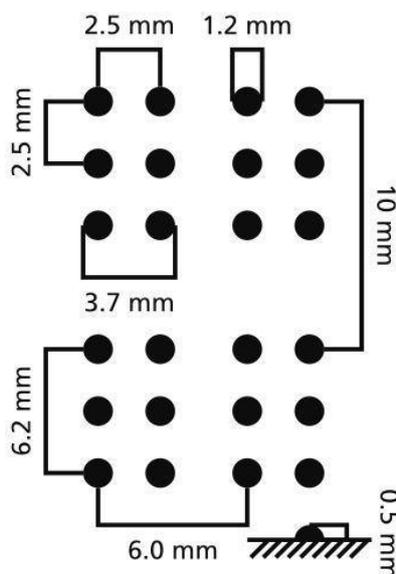


Figura 8 Dimensión braille estándar, hundido 0,5 mm

Fuente: <https://i0.wp.com/blog.cemebe>.

2.5 mm: Distancia horizontal entre puntos de la misma celda.

2.5 mm: Distancia vertical entre puntos de la misma celda.

6.0 mm: Distancia entre celdas (primera columna celda 1 y primera columna celda 2)

6.2 mm: Distancia entre el primer punto y el tercer punto de la celda

10.0 mm: Distancia vertical entre celdas (primer punto celda 1 fila 1 y el primer punto celda 2 fila 2).

0.5 mm: Altura de relieve de puntos

Reglas Ortográficas y Gramaticales

El sistema braille se rige bajo las reglas ortográficas y gramáticas según el idioma bajo el que se esté comunicando, en este caso para la lengua en español que establece la Real

Academia Española (RAE), en sistema braille no está exento de los errores ortográficos y gramaticales típicos como el uso de mayúsculas y minúsculas o las tildes, en gramática es necesario seguir las reglas principales como correcto tiempo verbal, el uso de los signos de puntuación y la estructura de la oración.

2.2.2 Medios de impresión de códigos braille

Método manual

Para escribir con el sistema braille se lo hace en forma manual mediante una regleta especializada y un punzón. La regleta está dividida por celdas con los puntos del signo generador tal como se muestra en la figura 9.

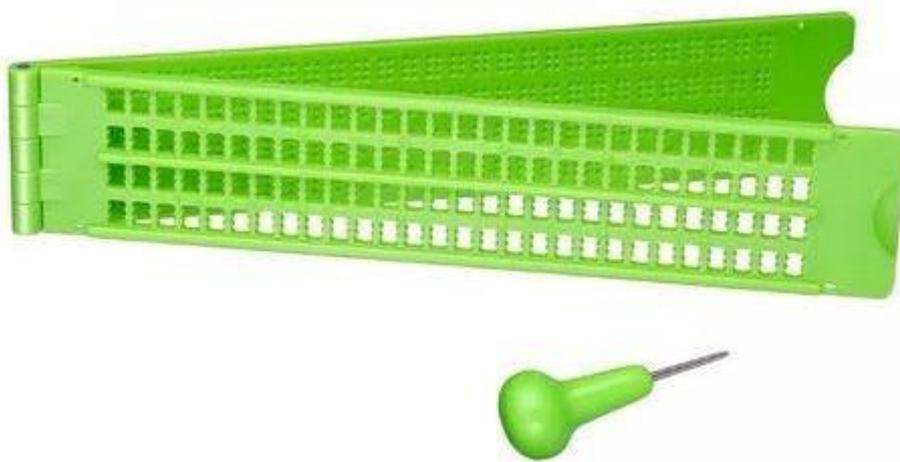


Figura 9 Regleta Braille 4 líneas por 28 celdas y punzón

Fuente: <https://www.tiflostore.cl/>

Este método es impartido como base común de aprendizaje en etapas iniciales y permite punzar sobre una hoja de papel para crear escritos con codificación braille. Para escribir los caracteres se inicia por punzar sobre la regleta los signos de manera invertida, es decir, se toma el orden de los puntos del signo generador invertidos y se punza de derecha a izquierda.

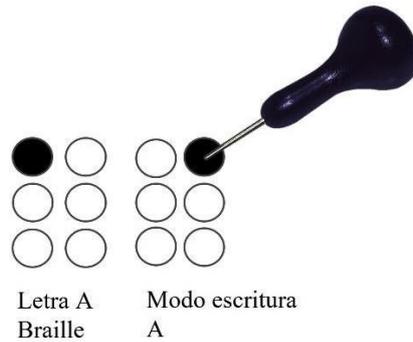


Figura 10 Ejemplo escribir letra A. Elaborado por el autor

Este tipo de regletas tienen diferentes tamaños y colores, las medidas comunes son las siguientes; regleta de 4 líneas por 28 celdas, regleta de 6 líneas por 19 celdas y la regleta de 27 líneas por 30 celdas, son elaboradas de material plástico resistente. El punzón está construido en plástico o madera con una punta de acero redondeado para no romper la hoja. La técnica empleada para escribir a mano debe ser realizada correctamente para obtener una precisa escritura de los caracteres, de modo que el método toma mucho tiempo realizar.

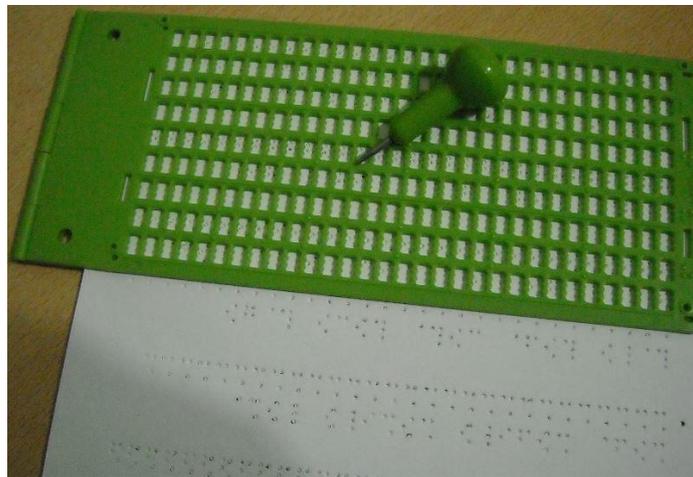


Figura 11 Escritura manual braille realizada con regleta y punzón

Fuente: <https://prevenirlaceguera.blogspot.com>

Máquina Perkins

La máquina Perkins es un dispositivo mecánico mediante el cual es posible escribir braille (ver figura 12), la escritura se realiza tal y como se lee el signo generador. Se puede

imprimir 25 líneas con 42 caracteres, su estructura exterior es de aluminio que asegura su durabilidad.



Figura 12 Maquina Perkins

Fuente: <https://www.sta-mexico.com>

Una maquina Perkins está construida en una estructura metálica con un sistema mecánico que consta de seis teclas conectadas a seis punzones que siguen el patrón del signo generador, una tecla espaciadora, una tecla de cambio de línea y una tecla de retroceso.

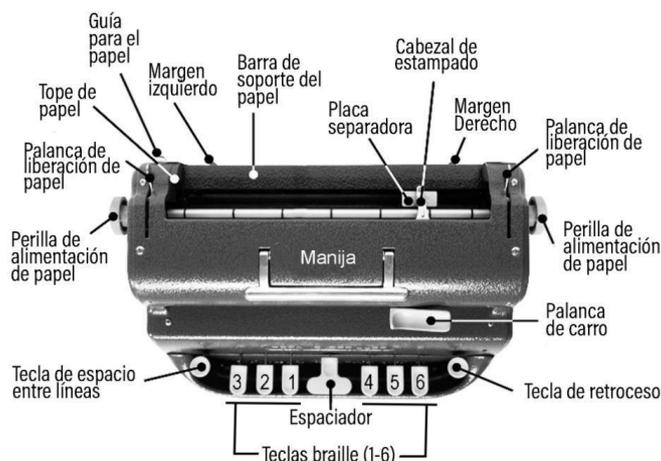


Figura 13 Partes de Maquina Perkins

Fuente: <https://opencluster-tech.cl>

Para escribir se deben presionar las teclas que componen el carácter que se desea expresar es muy importante para la persona invidente ubicar los dedos en el lugar adecuado y tener noción clara de la posición de cada punto del signo generador. Esta máquina permite una escritura más rápida a diferencia de la regleta y punzón, así también tiene una mejor calidad en el relieve de los caracteres [7].

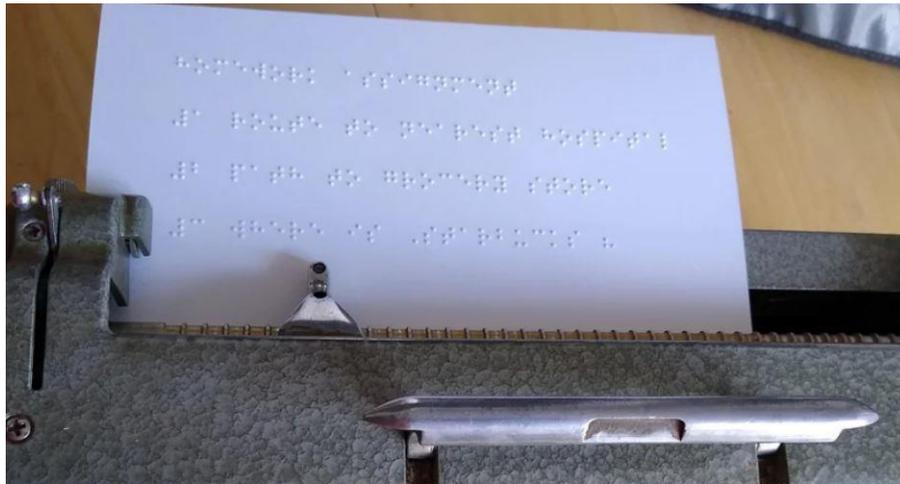


Figura 14 Escritura braille realizada con Maquina Perkins

Fuente: <https://www.reddit.com>

2.2.3 Impresoras Braille

Las impresoras braille son dispositivos electrónicos que han tecnificado el método de escritura, lo que se logra con una regleta y punzón o una maquina Perkins se puede reemplazar mediante el uso de un sistema autónomo, controlado por software.

Una impresora braille se conecta a un dispositivo electrónico el cual envían los datos a imprimir, el sistema de la impresora es capaz de traducir las palabras al braille, luego los puntos se crean mediante una serie de martillos o alfileres sin filo accionados para grabar los puntos braille en una hoja de papel, se utiliza un papel más grueso que el que se usa para impresión en tinta [8].

Existen impresoras braille de diferentes precios, los modelos más caros tienen mayores opciones y mejor precisión en el punzado incluso pueden imprimir documentos a doble cara, algunas son capaces de producir imágenes táctiles. La velocidad de impresión es medida en caracteres por segundo (CPS) y también incide en el elevado costo.

ViewPlus Tiger Embraille

ViewPlus Tiger es una impresora de escritorio ligera y compacta, ideal para uso doméstico o de oficina. No es la más rápida, imprime caracteres braille a una velocidad de 25 CPS, sin embargo, al ser una impresora de tamaño compacto es muy útil para uso personal. Posee la capacidad de cambiar entre diferentes mecanismos de alimentación de papel. Su país de fabricación es Reino Unido con un costo de 2.112,00 euros.

Impresión [9]

- Modo de impresión: a una cara.
- Altura de hundido: ajustable en tres alturas

Papel [9]

- Ancho de papel: 3 a 8,5 pulgadas (77 mm a 216 mm)
- Papel: alimentador continuo de hojas sueltas

Conexión / compatibilidad [9]

- Conexión: USB (1.1, 2.0)
- Compatibilidad: Windows7, 8/8.1, 10



Figura 15 ViewPlus Tiger Embraille

Fuente: <https://shop.rnib.org.uk>

Index Basic D-V5

Basic D-V5 es una impresora compacta y potente. Es un dispositivo muy vendido e incluye las últimas actualizaciones como conexión Wi-Fi, memoria USB y vinculación con dispositivos móviles para imprimir. Es capaz de imprimir hasta 100 caracteres por

segundo, utiliza papel plegado en acordeón para imprimir paginas a doble cara. Su costo es de 3646,12 dólares estadounidenses en el país de origen [10].



Figura 16 Index Basic D-V5

Fuente: <https://www.sightandsound.co.uk>

Index Everest D-V5

Es una impresora muy potente capaz de grabar en braille a altas velocidades 140 caracteres por segundo, puede imprimir 400 páginas A4 por hora. Es posible conectar a Wi-Fi e imprimir desde dispositivos móviles o memorias USB. Su precio es de 4395,00 dólares estadounidenses, al importar el producto su valor aumentaría por los impuestos de importación establecidos en el Ecuador [11].



Figura 17 Index Everest D-V5

Fuente: <https://www.indexbraille.com>

2.2.4 Control Numérico por Computadora (CNC)

El Control Numérico por Computadora es un sistema que consiste en el control preciso de las herramientas en los ejes X, Y, Z. Utiliza un software informático preprogramado escrito en un lenguaje llamado código G el cual es un lenguaje de programación numérico de puntos coordinados, mediante ellos se puede establecer parámetros para operar las herramientas en direcciones con respecto a los ejes, se puede conseguir un rápido movimiento, así también rotación en sentido horario y antihorario. Se pueden automatizar herramientas como fresadoras, amoladoras, tornos, taladros, chorros de agua y láseres [12].

El CNC permite que las máquinas produzcan artículos más rápido y con mejor acabado superficial sin la necesidad de mecanizado manual y mínima participación humana. Las maquinas CNC son rentables y reducen costos de fabricación, con menor consumo energético y mayor eficiencia de producción, escalabilidad y menos desperdicio de material [12].

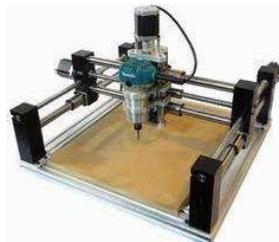


Figura 18 Maquina CNC para taladro

Fuente: <https://stores.2023saleonline.ru>

2.2.5 Programación CNC

La programación CNC se realiza por medio de instrucciones o códigos para controlar máquinas de control numérico por computadora (CNC). Al escribir los comandos se utiliza el lenguaje de programación código G, que especifica las acciones que debe realizar la maquina CNC. Los comandos incluyen información sobre movimientos de herramientas, velocidades de avance, velocidades de husillo, cambios de herramienta

entre otros parámetros que sirven para llevar a cabo una operación de mecanizado específico [13].

La programación de máquinas CNC se puede hacer de dos tipos diferentes:

- Programación manual: Un operador realiza de manera manual, en contacto directo con la máquina de control numérico. El programa realizado se ejecuta siguiendo las instrucciones especificadas por el operador y analizadas por la máquina.
- Programación asistida por la computadora: Este tipo de programación hace uso de softwares de diseño asistido por computadora CAD, en donde primero se diseña la pieza que va a ser mecanizada, entre los más usados están: Autocad, SolidWorks, ANSYS. Luego el diseño se convierte en un lenguaje entendible por la máquina en códigos G.

2.2.6 Lenguaje de programación CNC

El lenguaje de programación CNC se identifica con un lenguaje conocido como G y M. Es un lenguaje de programación vectorial que describe parámetros de maquinado como avance de herramienta y velocidad de husillo. Se definen G y M como Instrucciones Generales (G) e Instrucciones Misceláneas (M). Aunque se nombre al sistema códigos G, se usan otras letras y números que se designan para diversas funciones dentro de un programa CNC. Las más máquinas CNC tienen tres operaciones básicas que el sistema adapta para crear las piezas deseadas, tales como:

- Muévete en línea recta rápidamente
- Muévete en línea recta a una velocidad de avance específica
- Moverse en círculo a una velocidad de avance específica

Código G y M

El código G (también RS27D) es el lenguaje de programación CNC más usado. Están dispuesto en un formato alfanumérico y controla las acciones geométricas. Mediante el código G se puede producir los movimientos de las maquinas CNC e indica donde empezar, cómo moverse y cuando detenerse al fabricar una pieza.

Por otro lado, el código M, es un conjunto de comandos auxiliares que permite el control de todas las operaciones no geométricas de la máquina. Los operadores utilizan estos códigos para controlar acciones como detener programas, inundar la maquina con refrigerante y apagarla después de la reducción de la temperatura [14].

En la tabla 1, se describen algunos de los códigos más usados en máquinas CNC.

Código G	Función
G00	Posicionamiento rápido en un plano XY específico
G01	Movimiento de avance lineal
G02	Movimiento interpolación circular (en el sentido horario)
G03	Movimiento interpolación circular (en el sentido antihorario)
G04	Pausa, va acompañada de la letra P para pausar por un tiempo determinado
G10	Establece punto de referencia offset
G17	Seleccionar plano X, Y
G18	Seleccionar plano Z, X
G19	Seleccionar plano Z, Y
G21	Definir unidades métricas
G29	Mover a una ubicación a través del punto de referencia
G44	Compensación de la longitud de la herramienta
Código M	Función
M00	Parada del programa
M01	Parada del programa: opcional
M02	Fin del programa
M03	Arranque del husillo sentido horario
M04	Arranque del husillo sentido antihorario
M05	Parada del husillo
M06	Cambio de herramienta

Tabla 1 Lista de códigos G y M
Fuente: <https://gcodetutor.com>

2.2.7 Software de programación CNC

Las maquinas CNC funcionan con un software de programación el cual es responsable de crear las instrucciones y los códigos que la maquina CNC necesita para realizar una tarea. La mayoría de los softwares CNC se ejecutan sin intervención humana, todo el proceso está mecanizado automáticamente, muy preciso y eficiente [15].

Existe una variedad de softwares CNC que permiten escoger el que mejores prestaciones ofrezca en cuanto se debe considerar la compatibilidad con el sistema operativo, la variedad de funciones, opciones de personalización, facilidad de uso y gratuito o de pago.

Entre los softwares CNC más usados se encuentran LinuxCNC, Caballete, GRBL, PlanetCNC, Universal Gcode Sender (UGS), los cuales están disponibles de manera gratuita.

2.2.8 Controlador CNC

Contiene el cerebro de un sistema CNC, permite el vínculo entre un sistema informático y los componentes mecánicos de la máquina CNC. Su principal función es recibir las señales condicionadas de un computador e interpretar esas señales en movimiento mecánico a través de la salida del motor.

La palabra “controlador” se podría referir a un solo dispositivo, pero generalmente se refiere al sistema de control completo de la máquina. Este sistema puede incluir circuitos de protección, controladores de motor paso a paso o servomotor, fuente de alimentación, finales de carrera, controles de potencia y otros periféricos [16].

Los controladores actuales requieren que los operadores creen un programa que el controlador debe seguir, para ello los operadores reciben ayuda de softwares como paquetes de diseño asistido por computadora (CAD) y software de fabricación asistida por computadora (CAM) junto con el software del controlador para crear el código numérico necesario, como el código G. [16]

Tres son los componentes importantes del controlador CNC, la unidad de fuente de alimentación, el sistema de protección de circuitos y los controladores del motor

- Unidad de fuente de alimentación: La máquina CNC requiere una línea de comunicación de bajo voltaje, a través de la cual el computador envía las ordenas para que la máquina realice las acciones deseadas y una fuente de energía que proporciona energía para mover, cortar y otras operaciones similares.
- Sistema de protección de circuitos: Contiene una placa de ruptura para aislar las señales del computador, así poder distribuir las señales a los controladores deseados.
- Controladores de motor: Reciben la señal de comunicación y luego coordinan los pulsos de corriente y voltaje deseados para generar el movimiento en los motores de accionamiento.

2.2.9 Arduino

Arduino es una placa de desarrollo con código abierto (OPEN SOURCE), se utiliza con software y hardware libre y se utiliza para desarrollar proyectos de automatización. Es una placa mediante la cual se puede controlar sensores y actuadores físicos. Los programas desarrollados con Arduino son ejecutados mediante un ordenador, se alimentan a través de la conexión USB o con fuentes de alimentación externas para funciones autónomas. La placa Arduino tiene pines de entrada mediante el cual puede recibir datos externos de sensores y enviar información en pines de salida para actuadores.

Arduino es una placa basada en un microcontrolador Atmel de la familia AVR ATmega328 P, en el cual se escribe con el lenguaje de programación en entorno Arduino IDE, este lenguaje de medio nivel, trata con objetos básicos como caracteres, bits, direcciones de memoria y números, entre otros [17].

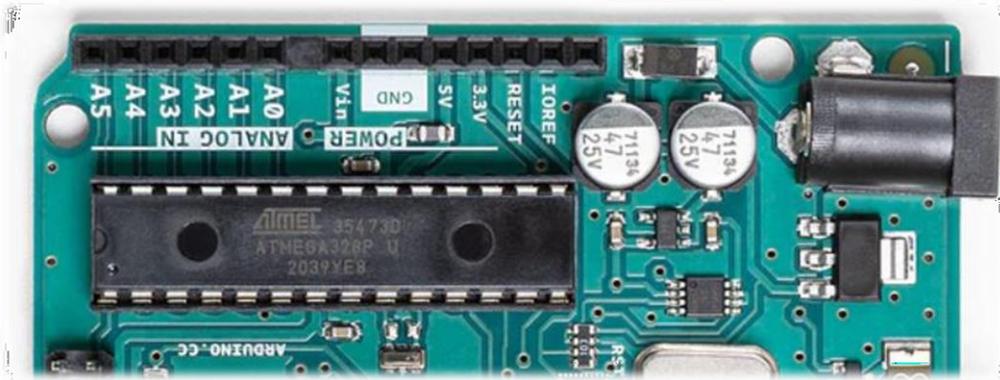


Figura 19 Tarjeta desarrollo ARDUINO UNO con ATmega328P

Fuente: <https://store-usa.arduino.cc>

2.2.10 Protocolos de comunicación Arduino

Existen diferentes tipos de protocolos de comunicación tales como I2C y SPI.

Protocolo de comunicación I2C

Es un protocolo de comunicación serial, que se realiza mediante dos cables en un modo de comunicación síncrona con una arquitectura **maestro- esclavo** (*master-slave*).

Este protocolo funciona con dos cables. El primer cable, SCL o Serial Clock Line (*línea de reloj*), se encarga de gestionar la señal de reloj. Se administra por el maestro. EL segundo cable, SDA o Serial Data Line (*línea de datos en serie*), se encarga de gestionar la señal de datos, todos los participantes envían datos por medio del mismo [18].

A continuación, se puede ver la arquitectura del protocolo I2C para conectar varios dispositivos.

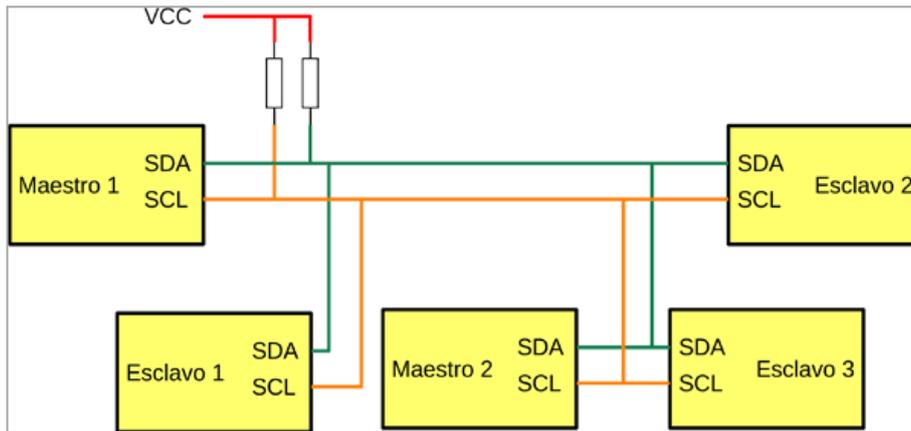


Figura 20 Protocolo de comunicación I2C

Fuente: <https://programarfacil.com>

Protocolo de comunicación SPI

Este protocolo tiene una arquitectura de tipo **maestro- esclavo** y la comunicación de datos entre maestros y esclavo se efectúan en dos líneas independientes, una es para el maestro a los esclavos y en la otra de los esclavos al maestro. Por tanto, la comunicación es **Full Duplex**, por ende, el maestro puede enviar y recibir datos al mismo tiempo. El dispositivo maestro proporciona una señal de reloj, misma que mantiene a todos los dispositivos sincronizados. Este protocolo funciona por medio de 4 líneas principales: CS, MOSI, MISO y CLK. Este último es la señal de reloj que administra los datos, CS es la salida del maestro y permite seleccionar el dispositivo esclavo al que enviará los datos, MOSI es la salida del maestro hacia el esclavo y MISO es la entrada del maestro desde el esclavo para recibir los datos [19].

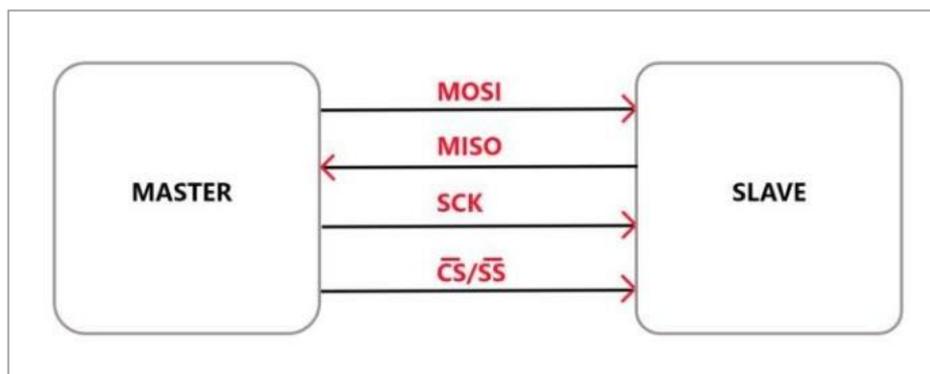


Figura 21 Interfaz maestro-esclavo SPI

Fuente: <https://www.makerguides.com>

2.2.11 Actuadores eléctricos

Son dispositivos formados por un motor eléctrico y un reductor que permite el accionamiento de cualquier dispositivo para realizar determinada acción o movimiento. La estructura de los actuadores eléctricos es simple a diferencia de los neumáticos o hidráulicos, los actuadores eléctricos funcionan por medio de una fuente eléctrica.

Los actuadores eléctricos están involucrados en un sin número de procesos industriales algunas de sus aplicaciones en sistemas de visión, movimiento de brazos articulados, manipulación automática de objetos, elevadores verticales y desplazamiento de cargas.

2.2.12 Sistemas electromecánicos

Se definen como sistemas híbridos de variables mecánicas y eléctricas. Las aplicaciones de sistemas electromecánicos se utilizan en tecnologías de control de movimiento sofisticado, dispositivos portátiles electrónicos/fotográficos, vehículos y subsistemas vehiculares, entre otros [20].

Los motores eléctricos y motores paso a paso demuestran el funcionamiento de los dispositivos electromecánicos.

Motores eléctricos

Son dispositivos que convierten la energía eléctrica en energía mecánica mediante el uso de un sistema de engranajes y campos magnéticos. Su fuente de alimentación se puede dar por medio de una variedad de fuentes de corriente directa (DC) y corriente alterna (AC), incluidas baterías, rectificadores, redes eléctricas, inversores y generadores eléctricos.



Figura 22 Motor eléctrico DC

Fuente: <http://www.tw-motor.net>

Motores paso a paso

Los motores paso a paso convierten la electricidad en rotación. El funcionamiento de un motor de pasos es mediante pulsos eléctricos provenientes de un controlador permitiendo el desplazamiento mecánico, cada pulso de electricidad hace girar el motor un paso. El movimiento discreto del motor paso a paso lo hace ideal en un sistema de control digital y su aplicación permite tener desplazamientos precisos [21].

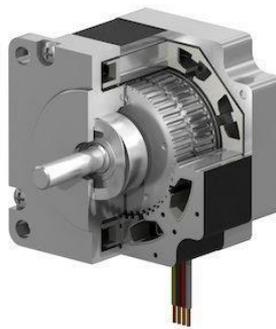


Figura 23 Motor paso a paso

Fuente: <https://www.motioncontroltips.com>

El motor paso a paso está compuesto por rotor y estator. Las bobinas se encuentran montadas en lugares específicos alrededor del rotor y este a su vez se encuentra montado sobre el eje. Tanto el rotor como el eje giran juntos, dos mitades de imán constituyen el rotor. El estator es una parte estacionaria que permanece inmóvil y está formada por un conjunto de bobinas de alambre. El controlador del motor se encarga de regular el

momento en que la corriente eléctrica fluye a través de las bobinas lo que crea un campo electromagnético que produce la rotación. Los motores paso a paso tienen múltiples bobinas que se organizan en grupos llamados “fases”. Al energizar cada fase en secuencia, el motor gira un paso [22].

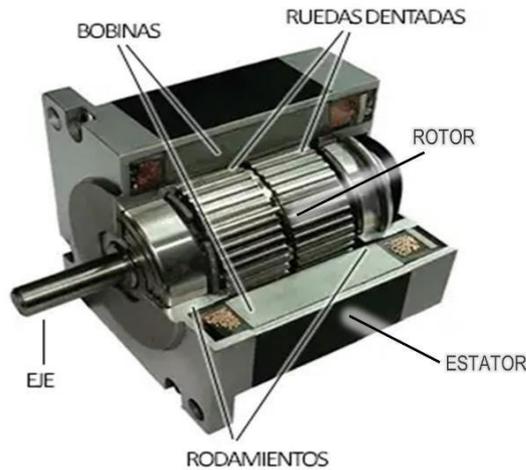


Figura 24 Partes de un motor paso a paso

Fuente: <https://www.luisllamas.es>

Los motores paso a paso se dividen en unipolares y bipolares. El motor unipolar trabaja con una polaridad de corriente y el motor bipolar trabaja con dos polaridades, lo que significa que la dirección del flujo de corriente en la bobina es variable [22]. Los motores bipolares tienen cuatro conductores mientras que los unipolares cinco o seis. Los motores unipolares pueden ser más sencillos de manejar, por otro lado, los motores bipolares requieren de circuitos de control y complejos para su operación.

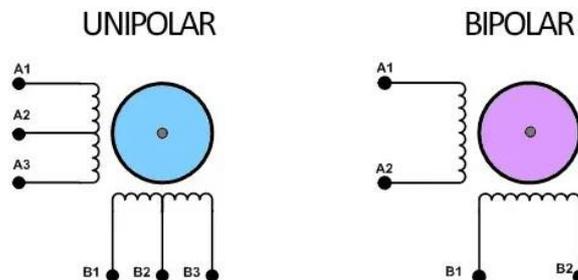


Figura 25 Conexión Unipolar y conexión bipolar de un motor paso a paso

Fuente: <https://www.luisllamas.es>

2.2.13 Controlador de motor paso a paso

Un controlador o driver es un dispositivo electrónico que se utiliza para accionar el motor, ya que por sí solo no hace nada. Estos dispositivos permiten controlar los voltajes e intensidades que se están suministrando al motor. Usar un controlador permite el manejo de la potencia que usan los motores de paso a paso, ya que los mismos no pueden ser controlados únicamente por el microcontrolador de una placa Arduino [23].

Dependiendo del tipo de motor que se vaya a usar, existen drivers para motor unipolar y drivers para motor bipolar. Su diferencia radica en la complejidad de cada uno de ellos, los drivers para motor unipolar son más sencillos y fáciles de controlar ya que la corriente que circula por las bobinas del motor siempre va en una misma dirección. Los controladores de motor bipolar son complejos y el mismo debe encargarse de activar la corriente en una u otra dirección, permitiendo girar el motor en ambos sentidos.

Utilizar drivers para los motores forman parte de la correcta conexión electrónica que debe realizarse para el uso de los motores paso a paso, sirven para separar la parte de control con la de potencia del motor paso a paso. Y evitar el daño del microcontrolador o la placa de desarrollo (Arduino).

2.2.14 Solenoides

Un solenoide es un objeto cilíndrico que genera un campo magnético cuando una corriente fluye a través de su cable, también se puede utilizar un inductor, en lugar de un electroimán, para imposibilitar cambios en la corriente eléctrica.

Un solenoide consiste en una bobina de alambre en forma de sacacorchos envuelta alrededor de un pistón que generalmente está hecho de hierro. Los solenoides usan imanes, los cuales pueden encenderse o apagarse usando una computadora o quitando la corriente. Esto hace que particularmente se usen en switches o válvulas, se puede encontrar en llaves de carro, cerradura de puertas y una variedad de sistemas industriales automáticos [24].

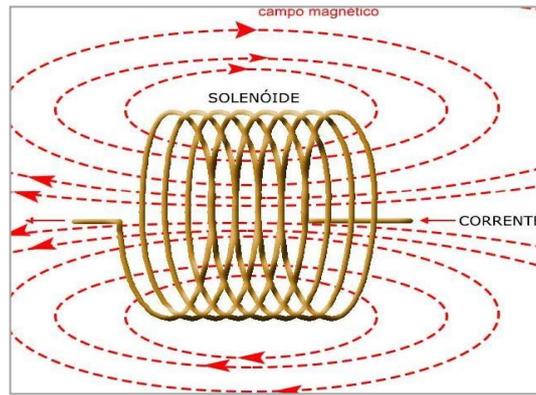


Figura 26 Representación de campos magnéticos y corriente en solenoide

Fuente: <http://www.incb.com.mx>

Solenoides lineales

Son dispositivos electromagnéticos que proporcionan un movimiento lineal a través de un embolo de hierro como núcleo, en aplicación se pueden comparar con el principio de los relés electromecánicos y se manejan mediante transistores o Mosfets. El movimiento lineal se incluye en aplicaciones de jalar y empujar. La mayoría tiene un resorte para devolver el émbolo a su posición inicial. Pueden ser unidireccionales o bidireccionales [25].



Figura 27 Solenoide lineal con muelle

Fuente: <https://www.amazon.com>

Los solenoides lineales se pueden utilizar en varias aplicaciones como control de válvulas de irrigación, transmisiones hidráulicas, cerraduras de puertas válvulas de control hidráulico, entre otras [26].

Solenoides giratorios

Este tipo de solenoide es un dispositivo electromecánico que se utiliza para movimiento rotatorio, pueden girar en el sentido horario y antihorario. El núcleo está montado en un disco plano. Tiene gran utilidad cuando se necesita un movimiento angular muy pequeño que no se podría realizar con un motor paso a paso [26].



Figura 28 Solenoide giratorio

Fuente: <https://industrysurfer.com>

En aplicación estos dispositivos se pueden utilizar en láseres, equipos médicos, clasificadores y máquinas expendedoras [26].

2.2.15 Diseño 3D

Un diseño 3D permite proyectar un objeto en tres dimensiones, mediante el uso de técnicas de diseño asistido por computadora (Computer Aided Design, CAD). Un diseñador 3D trabaja para transformar un concepto unidimensional en representaciones tridimensionales. Se requiere un profundo conocimiento del software de diseño 3D y herramientas CAD para crear modelos y representaciones 3D. Algunos de los softwares más usados son SOLIDWORKS, FreeCAD, Sketchup entre otros.

2.2.16 Impresión 3D

Es un proceso de creación de objetos tridimensionales, se realiza mediante un proceso de adición, que se logra al colocar capas sucesivas de material que salen de una pequeña

parte llamada *extrusor*, el cual es el encargado de expulsar el material caliente en forma de una fina tira plástica para crear las piezas 3D. Una ventaja de la impresión 3D es la facilidad de producir formas complejas utilizando menos material que los métodos de fabricación tradicional. Los plásticos y las aleaciones de metal son de los materiales más usados en impresoras 3D, pero también actualmente existen impresoras 3D capaces de usar hormigón hasta tejido vivo [27].

2.2.17 Impresora Ultimaker s5

Es una impresora 3D de escritorio de fácil uso, muy útil para imprimir piezas precisas de calidad industrial. Con su gran volumen de impresión (330x240x300mm), permite obtener piezas de gran tamaño. Cuenta con una interfaz fácil de usar para los usuarios donde pueden configurar y controlar la impresión, el tipo de material usado va desde el PLA y ABS hasta materiales más avanzados como el nylon o el policarbonato [28].



Figura 29 Impresora Ultimaker S5

Fuente: <https://tienda.sicnova3d.com>

2.2.18 Software para comunicación

Se define el software de comunicación como el programa que sirve para comunicar al ordenador con la impresora, mediante el ingreso de los textos por teclado que son convertidos en coordenadas en código G, que luego son enviadas a la placa de control para que muevan los motores y actuador que plasme los caracteres braille. Para cumplir

este proceso se utiliza software de código abierto que permita enviar los textos a imprimir de manera correcta.

2.2.19 GitHub

Es un portal en la nube en donde se almacena códigos fuente de muchos desarrolladores en el mundo, es una herramienta muy útil para cargar y descargar programas o aplicaciones, los cuales se pueden ir desarrollando en colaboración con proyectos compartidos con otros programadores. En esta plataforma se pueden hallar muchos códigos realizados por otros desarrolladores que los ofrecen de manera libre para descarga, lo que produce un gran intercambio de conocimiento y trabajos investigativos.

Está basado en un sistema de control de versiones (VCS) que ayuda a rastrear modificaciones hechas en el código fuente durante el desarrollo de un proyecto, esto sirve para comparar y analizar los cambios, así también revertir si algo está mal [29].

2.2.20 Interfaz de comunicación

Se puede definir como la comunicación que se establece entre el ordenador con los dispositivos periféricos, se unen a través de cables enlazados a sus respectivos puertos de conexión o también mediante conexiones inalámbricas y permiten el intercambio de información.

2.2.21 Puerto de comunicación serie

Un puerto serie o puerto en serie es una interfaz de comunicaciones de datos digitales, es utilizado por computadoras y periféricos. A través de ellos se transmite la información bit a bit, enviando un solo bit a la vez en contraste con el puerto en paralelo que puede transmitir varios bits en forma simultánea. Trabajan bajo el estándar RS-232 [30].

Existen tres tipos de comunicación en serie:

- Simplex: El envío de los datos es unidireccional, se transmite desde el emisor hacia un receptor en una sola dirección.
- Dúplex: Se utilizan dos cables para enviar y recibir los datos, el emisor y el receptor pueden transmitir simultáneamente.
- Semi- Dúplex: Similar al dúplex, mientras uno transmite el otro únicamente recibe la señal, cuando termina puede transmitir datos de respuesta.

Para usar una interfaz de comunicación existe el chip UART o USART (Transmisor y Receptor Asíncrono Universal). Permiten convertir las señales y tensiones de la CPU al estándar de comunicación RS-232.

La comunicación de Arduino UNO es en serie, y tiene el soporte integrado para UART. El Arduino puede transmitir y recibir datos a la PC a través del cable USB [31].

2.3 MARCO TEÓRICO

La educación y el aprendizaje tienen una relación muy estrecha con los sentidos ya que mediante los mismos se puede explorar y aprender de todo lo que nos rodea. La vista es el sentido que más destaca, el filósofo Aristóteles planteó esta jerarquía en la que la vista es el sentido más importante, seguido del oído, el olfato, el tacto y el gusto [32].

Con el advenimiento de la tecnología, el sistema braille se ha modernizado, proporcionando herramientas que optimicen el proceso educativo. Las personas con discapacidad visual cuentan en la actualidad con máquinas electromecánicas conocidas como impresoras braille, estas automatizan el proceso de escritura, pero los costos actuales son elevados, lo que propone un reto a los investigadores para crear impresoras braille al alcance de las personas no videntes.

La propuesta realizada en la Escuela Superior Politécnica de Chimborazo con el título “Diseño y construcción de una impresora braille de bajo costo, utilizando plataforma hardware de código abierto” menciona que el proyecto busca elaborar una impresora braille de bajo costo que ayude a personas de pocos recursos económicos que están en el proceso de aprendizaje mediante el uso de componentes electrónicos reciclados de impresoras dañadas o sin usar para implementar un sistema capaz de imprimir textos en código braille [33].

En Lima año 2015, en la UNIVERSIDAD DE SAN MARTÍN DE PORRES-PERU, los estudiantes Cleidy Geraldinne Ato y Jesús Manuel Chinchá realizaron su trabajo de graduación titulado “Diseño de un sistema automático de impresión de caracteres de código braille basado en comparación y utilizando un actuador electromagnético” en este trabajo se desarrolló una impresora de caracteres braille, adaptada sobre una impresora de inyección de tinta, este sistema permite traducir e imprimir textos en caracteres braille. Para la impresión utiliza un electroimán que ejerce el punzado para plasmar los relieves de código braille sobre una hoja de papel A4 de 120gr [34].

Miguel Ángel Maldonado en el año 2015, estudiante de la UNIVERSIDAD DEL AZUAY-ECUADOR, realizó su trabajo de graduación titulado “Diseño programación e implementación de una impresora braille” en el siguiente trabajo se construyó un sistema de impresión braille utilizando la plataforma de programación Arduino y una interface de usuario en LabVIEW para manipular el sistema desde un computador. Esta impresora tiene motores paso a paso que son los responsables de desplazar un solenoide y el arrastre de hojas para la impresión de textos en relieve con código braille [35].

El desarrollo tecnológico actual permite que está presente propuesta de implementación de un sistema electromecánico pueda imprimir textos braille que sirvan para la enseñanza y aprendizaje de personas no videntes.

2.4 COMPONENTES DE LA PROPUESTA

2.4.1 Componentes físicos

Para la implementación de la propuesta se detallan los componentes físicos utilizados con sus respectivas características y uso.

2.4.1.1 Motor paso a paso

Para desplazar tanto el carrete como el rodillo para el movimiento de la hoja se utilizan dos motores paso a paso, se eligió el modelo KS42SH40-1204A, comúnmente conocido en la gama de los motores paso a paso NEMA 17, trabaja a corriente de 1,2 A, se los

encuentra en impresoras 3D, maquinas CNC y aplicaciones de control de precisión. Sus características se muestran en la tabla a continuación.



Figura 30 Motor paso a paso KS42SH40-1204A

Fuente: <https://protosupplies.com>

MOTOR PASO A PASO	
Modelo	KS42STH40-1204 ^a
Compatibilidad estándar	NEMA-17
Numero de fase	2(bipolar)
Alambres de plomo	4 (se utilizan 4 pines del conector JST de 6 pines)
Clasificaciones operativas	
Voltaje estándar	2,0 VDC
Corriente por fase	1.2 A
Resistencia de fase	1,7 ohmios
Torque de retención	0,4 Nm (mínimo)
Angulo de paso	1,8 grados
Conteo de pasos (rotación 360 grados)	200
Dimensiones	
Caja ancho x alto x largo	42 x 42 x 40 mm (1,65 x 1,65 x 1,57 pulgadas)
Diámetro del eje	5 mm (0,2 pulgadas)
Longitud del eje	22 mm (0,87 pulgadas)
Peso	290

Tabla 2 Especificaciones técnicas motor paso a paso.

Fuente: <https://gcodetutor.com>

2.4.1.2 Solenoide 24 VDC

Para realizar los puntos braille se utiliza un actuador electromecánico, llamado solenoide. Las especificaciones técnicas del solenoide se pueden ver en la tabla 3.

SOLENOIDE	
Voltaje	24 VDC
Perfil	Cuadrado
Tipo	Empujar
Corriente	3 A
Fuerza	3.5 N
Dimensiones	40 x 24 x 24 mm (L*W*H)
Peso	145 g

Tabla 3 Especificaciones técnicas Solenoide.

Fuente: <https://www.transmotec.com>

El solenoide genera un movimiento lineal, está formado por una bobina con un orificio interior en donde se ubica un vástago. Al energizar la bobina, se empuja el vástago hacia abajo para realizar la función de punzado. En la parte inferior del vástago se instala una punta de acero con el diámetro de dimensión braille (0.5 mm) y en la parte superior del vástago se ubica un resorte que permite el retorno a su posición inicial.



Figura 31 Solenoide 24 V

Fuente: <https://www.transmotec.com>

2.4.1.3 CNC Shield V3

Es una placa de control compatible con Arduino UNO y MEGA, permite construir una maquina CNC, está diseñada para conectar cuatro drivers A4988 o DRV8825, y tener el control de tres motores paso a paso en los ejes (X, Y, Z) y un motor paso a paso adicional que trabaja duplicado a uno de los anteriores [36].

Muy útil y compatible con GRBL, el cual es un firmware OPEN SOURCE para Arduino UNO que convierte códigos G en comandos para motores paso a paso se pueden construir proyectos de robótica incluye enrutadores CNC, contadoras Láser, Brazo robótico con una gran capacidad para controlar con precisión los grados de rotación.

Para energizar el Shield se puede utilizar fuentes de alimentación entre 12 – 36 VDC. La fuente de alimentación debe ser de aproximadamente de 2 A por cada motor, por esa razón si se utiliza cuatro motores se necesita una fuente de 8 A. Es recomendado el uso de una fuente de alimentación de 12VDC/ 8.5 A o 24 VDC/5 A [36].

Esta placa es la encargada del control de los motores de paso en los ejes X, Y, el actuador solenoide en el eje Z. Permite distribuir la alimentación y la traducción de las instrucciones enviadas desde la interfaz del ordenador y que convierte las coordenadas en los caracteres braille sobre el papel.

Se especifican las características técnicas en la tabla a continuación.

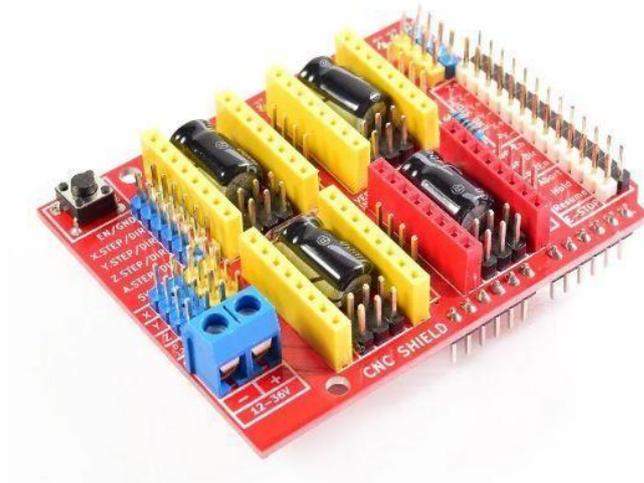


Figura 32 CNC Shield V3

Fuente: <https://altronics.cl/shield-cnc-v3-arduino>

CNC Shield V3	
Voltaje de entrada	12 – 36 VDC
Compatible ARDUINO	UNO
Compatible Firmware	GRBL
Conexiones	2 finales de carrera por cada eje
Numero zócalos ejes	X, Y, Z A
Drivers compatibles	A4988 o DRV8825
Tipo Bornes	Molex de 4 pines
Dimensiones	6.9 cm x 5.3 cm x 1.9 cm
Peso	31.6 g
Fusible	En placa

Tabla 4 CNC Shield V3.

Fuente: <https://altronics.cl/shield-cnc-v3-arduino>

2.4.1.4 Controlador o driver A4988

Para controlar el motor paso a paso, es necesario usar un driver compatible con las características del motor paso a paso. Se optó por el driver A4988, el cual es ideal para el manejo de motor paso a paso bipolar de hasta 2 A. El driver A4988 se conecta al motor paso a paso que mueve el carrete, su conexión es sencilla, mediante los pulsos digitales enviados por el microcontrolador Arduino se puede controlar el giro y los pasos del motor. Cuenta con un potenciómetro en el módulo que permite controlar la corriente que se enviará al motor paso a paso. En la tabla 5. Se especifican las características técnicas del driver A4988.

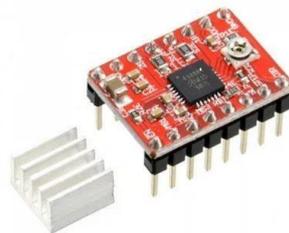


Figura 33 Driver A4988

Fuente: <https://electronicathido.com>

DRIVER A4988	
Voltaje de alimentación mínimo	8 VDC
Voltaje de alimentación máxima	35 VDC
Corriente por bobina máxima	2 A
Capacidad	Micro-stepping
Pines de comunicación con microcontrolador	STEP y DIRECTION
Resoluciones	<ul style="list-style-type: none"> ○ Paso completo ○ 1/2 de paso ○ 1/4 de paso ○ 1/8 de paso ○ 1/16 de paso
Numero de pines	16 pines

Tabla 5 Especificaciones técnicas driver A4988.

Fuente: <https://electronicathido.com>

2.4.1.5 Control Solenoide

Para activar o desactivar el solenoide se construye un circuito de potencia para separar de la señal al Arduino. Es necesario que el circuito trabaje con 24 VDC como se especifica en las especificaciones técnicas del solenoide. Se eligió un transistor TIP 110 el cual permite la interconexión con Arduino. Son transistores NPN de base epitaxial de silicio en configuración monolítica Darlington. El transistor funciona como interruptor y para encender el solenoide se maneja la base del transistor usando un pin digital de Arduino. En la tabla 6 se encuentra las características del transistor TIP110.

TRANSISTOR TIP110	
Fabricante	SAVANTIC
Serie	TIP 110/111/112
Voltaje Colector – emisor (V_{CE})	60 VDC
Voltaje Colector – base (V_{CB})	60 VDC
Disipación total del dispositivo	50 W

Temperatura de operación (Tj)	150 °C
-------------------------------	--------

Tabla 6 Especificaciones técnicas TIP110

Fuente: <https://html.alldatasheet.es/>

Como complemento para el TIP110, se selecciona un diodo 1N4002 con una corriente de 1A que sirve para prevenir la alta tensión generada cuando se detiene el solenoide, ya que esto puede dañar el Transistor o incluso el Arduino. La bobina del solenoide al activarse tiene las propiedades similares a las de un motor, por lo que al detenerse el campo magnético cae y genera una tensión muy alta a través del solenoide en la dirección inversa, por ende, el diodo ayuda como vía rápida para la descarga de toda la energía almacenada en el solenoide.

2.4.1.6 Microcontrolador ATmega328P (Arduino UNO)

Arduino UNO es una placa compacta y fácil de manejar, tiene una plataforma de software libre que se puede descargar desde su página web, el lenguaje de programación está basado en lenguaje C++. Para encender el Arduino UNO se conecta a la computadora con un cable USB, también se lo puede conectar a una fuente de alimentación o una batería. Mediante el cable USB se puede conectar la placa Arduino UNO. En la tabla 10 se describen las características técnicas.

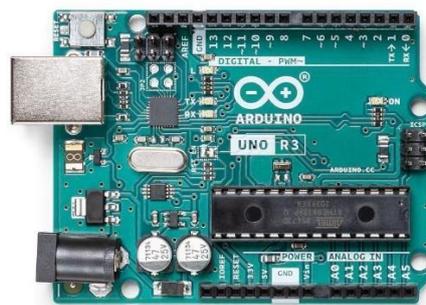


Figura 34 Arduino UNO ATmega328P

Fuente: <https://store-usa.arduino.cc>

ARDUINO UNO	
Fabricante	Arduino
Microcontrolador	ATmega328P

Tensión de funcionamiento	5 VDC
Voltaje de entrada (recomendado)	7 – 12 VDC
Voltaje de entrada (límite)	6 – 20 VDC
Pines de E/S digitales	14 (6 pines PWM)
Pines de E/S digitales PWM	6
Pines de entrada analógica	6
Corriente Continua por pin de E/S	20 mA
Corriente Continua para pin de 3,3 V	50 Ma
Memoria flash	32 KB (ATmega328P)
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
Longitud	68,6 mm
Ancho	54,4 mm
Peso	25

Tabla 7 Especificaciones técnicas Arduino UNO.

Fuente: <https://store-usa.arduino.cc>

2.4.1.7 Fuente de Alimentación

Para la alimentación del sistema se utiliza una fuente externa de 24 VDC/ 5 A. Este voltaje alimentará el solenoide. Así también, distribuye el voltaje a la placa controladora que provee de alimentación a los motores paso a paso. Las características de la fuente externa se observan en la tabla 8.



Figura 35 Fuente de poder conmutada 24 VDC 5A

Fuente: <https://www.velasco.com.ec>

FUENTE DE PODER CONMUTADA 24 VDC 5 A	
Fabricante	EVL
Modelo	TPS-2405
Voltaje de entrada	110 – 264 VAC (50/60 Hz)
Voltaje de salida	24 VDC
Corriente de salida max.	5 A
Potencia max.	120 W
Dimensiones	120 x 98 x 38 mm (L*W*H)
Peso	0.6 Kg.

Tabla 8 Especificaciones técnicas fuente alimentación 24VDC 5A.

Fuente: <https://www.velasco.com.ec>

2.4.1.8 Componentes físicos de la estructura

Para la construcción de la impresora braille fue importante seleccionar los materiales que comprenden la estructura física.

Base MDF

Para la construcción de la base se usa tablero MDF, el cual es un material hecho a partir de fibras de madera y resinas sintéticas comprimidas, su composición permite que pueda ser tallada o fresada. Se utiliza en la fabricación de muebles, modelos arquitectónicos, construcción de prototipos, artículos decorativos o juguetes. Al tener una alta calidad de superficie se puede cortar las dimensiones correctas mediante tecnología láser.



Figura 36 Tablero MDF 3mm

Fuente: <https://www.komexpma.com>

Ejes acerados cromados

Son barras de acero de gran resistencia a la corrosión y desgaste, reducen la fricción ya que tienen una superficie con buena retención de aceite, sirven como ejes para instalar el carrete y para hacer girar el rodillo de ingreso de hoja. El eje utilizado posee un diámetro de 6mm por 400 mm de largo.



Figura 37 Ejes cromados DIAMETRO 6mm LARGO 400mm

Fuente: <https://roboticsec.com>

Polea GT2

Es un elemento de aleación de aluminio usado en máquinas CNC, diseñadas para generar un movimiento lineal, tienen dientes redondeados que encajan con los dientes de una correa de manera precisa. Se utilizan dos medidas de polea de 20 dientes, para eje de 5mm y eje de 6.5 mm, adicional el diámetro del piñón es de 12 mm. Están diseñadas para trabajar junto a correas GT2 de 6 mm de ancho.



Figura 38 Polea GT2 DIAMETRO 5 mm

Fuente: <https://roboticsec.com>

Acoplamiento Flexible

Un acoplamiento es un elemento de aluminio maquinado, con ranurado helicoidal con ajustes de tornillos hexagonales, son utilizados para transmitir potencia desde un motor a un eje. Son muy utilizados en máquinas CNC o impresoras 3D, existen acoples del tipo rígidos y flexibles, los acoplamientos flexibles permiten solucionar los problemas de alineación cuando se une un motor a un eje. El acoplamiento usado es un Acople Flexible 6.35 mm a 5 mm.



Figura 39 Acople Flexible 6mm a 5mm

Fuente: <https://roboticsec.com>

Rodamiento lineal

Es un elemento mecánico como guía lineal que dispone de rodillos en su interior de tipo cilíndrico, lo que produce movimientos circulares. Este tipo de rodamiento reduce la fricción entre los ejes y las piezas que se conectan, permiten tener desplazamientos sobre un eje lineal en línea recta muy precisos, suaves y estables, fabricados en acero, también llamados casquillos lineales. El modelo utilizado es el LM06UU con diámetro 6mm y sirven para encajar perfectamente con los ejes de acero de 6mm.



Figura 40 Rodamiento LM06UU DIAMETRO 6mm

Fuente: <https://roboticsec.com>

Correa dentada GT2

Una correa dentada es un elemento necesario en los desplazamientos lineales generados por un motor, el material del cual están fabricados es neopreno y fibras de vidrio en el núcleo que mantienen su resistencia. A diferencia de las correas lisas una correa dentada tiene dientes que encajan sobre los dientes de la polea, esto produce un movimiento lineal preciso. La correa dentada GT2 es usada en aplicaciones de impresoras 3D y sistemas CNC. Es utilizada para mover el carrete y ubicar el solenoide para punzar sobre el papel.



Figura 41 Correa dentada GT2 ANCHO 6mm

Fuente: <https://roboticsec.com>

2.4.2 Componentes lógicos

A continuación, se da a conocer los softwares empleados en el desarrollo del proyecto, mediante estos se puede diseñar la estructura, programar el microcontrolador, configurar las instrucciones para transformar los códigos G en movimiento y realizar el diseño de las conexiones previo a la implementación.

2.4.2.1 IDE Arduino

El software IDE Arduino es un software de código abierto que se utiliza para programar las placas Arduino. Contiene un editor de texto para escribir código que soporta lenguaje de programación (C / C++), tiene un área de mensajes para mostrar errores y comentarios al guardar, dispone de un monitor serial donde se inicia el intercambio de datos que se presentan en una interfaz. Los programas escritos usando IDE Arduino son llamados

Sketches, son guardados con la extensión de archivo. ino. Es un software compatible con diferentes sistemas operativos como Windows, Linux, Mac OS.



Figura 42 Software IDE Arduino

2.4.2.2 Firmware grbl

Es el programa para el control de sistemas CNC, tiene las instrucciones necesarias para transformar el archivo GCode en movimiento de los motores. Es un programa compatible con placas Arduino, se carga el firmware en Arduino Uno para interpretar las instrucciones enviadas por el ordenador y luego traducir en ordenes de movimiento a los drivers que se conectan en el módulo CNC Shield. Es un programa libre de código abierto y de fácil descarga, la versión usada es grbl v 0.9, descargada del portal GitHub.



Figura 43 Firmware grbl v 0.9

Fuente: <https://github.com/grbl/grbl>

2.4.2.3 Universal Gcode Sender (UGS)

Es un software de aplicación de códigos G gratuito que consta de todas las funciones que se utilizan para controlar sistemas CNC. En esta plataforma se puede enviar comandos G a impresoras 3D y maquinas CNC, esta herramienta ayuda a mantener una interacción tiempo real entre la computadora y la máquina, lo que permite una mayor supervisión del control de la CNC, los motores y herramientas por ende sirve como campo de pruebas muy útil, preciso y eficiente. En este proyecto se usa la versión 2.1.2, descargada de manera gratuita desde su portal web www.universalgcodesender.com.

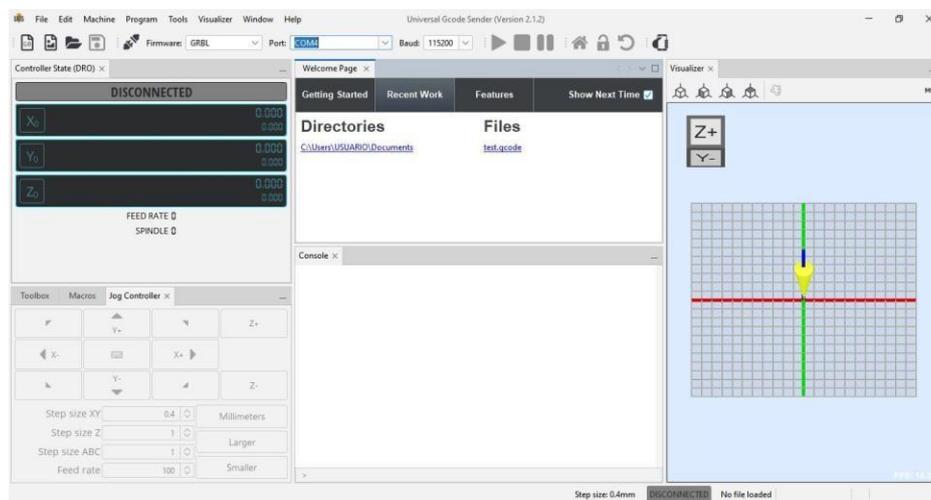


Figura 44 Software Universal Gcode Sender versión 2.1.2

2.4.2.4 Altium Designer

Es un software para diseño electrónico y PCB (Printed Circuit Board) para placas de circuito impreso, incluye herramientas para todas las tareas de diseño de circuitos: desde la captura de diseño esquemático, simulación de circuitos, análisis de integridad de señales, diseño PCB y el desarrollo de sistemas integrados basados en FPGA. A través de este programa se realizó los diagramas de conexiones de los diferentes componentes electrónicos.



Figura 45 Software Altium Designer

2.4.2.5 SolidWorks

Es un software de diseño asistido por computadora (Computer Aided Design, CAD) que permite realizar y modelar piezas en 3D. Ofrece funciones que le permiten ver diferentes ángulos, renderizar el diseño y ejecutar simulaciones. Este software tiene tres tipos de archivos: partes, ensamblajes y dibujos, que se crean interactuando con los comandos de su interfaz de usuario. Mediante este software se realizó el diseño general de la impresora braille, así también las diferentes piezas con sus dimensiones específicas.

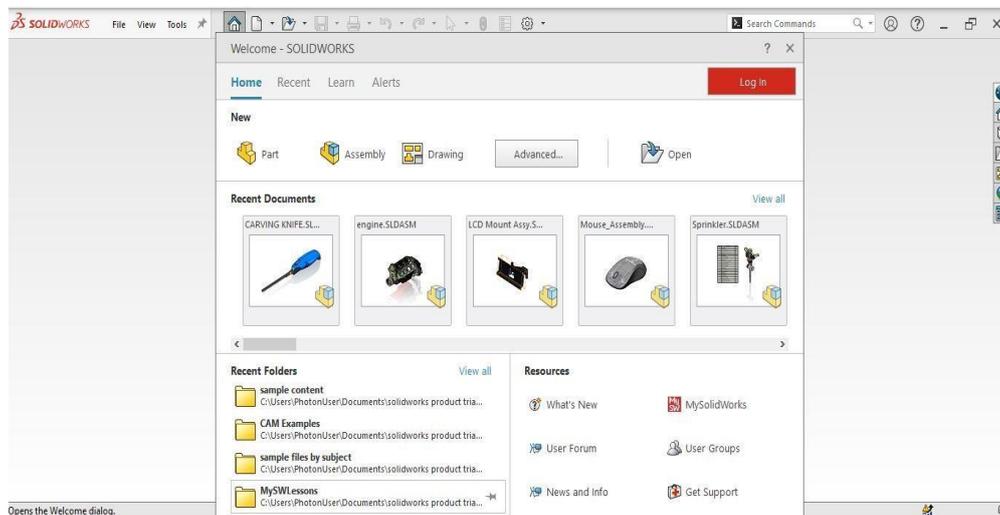


Figura 46 Software SolidWorks

2.4.2.6 Ultimaker Cura

Es el software más usado en impresión 3D, con el podrás transformar cualquier diseño 3D realizado en el ordenador en CAD en un archivo que se puede imprimir en una impresora 3D. Es un software de fácil uso y compatible con diferentes impresoras como

Creality, Ultimaker, Anet, entre otras. Los formatos de archivo admitidos son STL, OBJ, X3D Y 3MF.

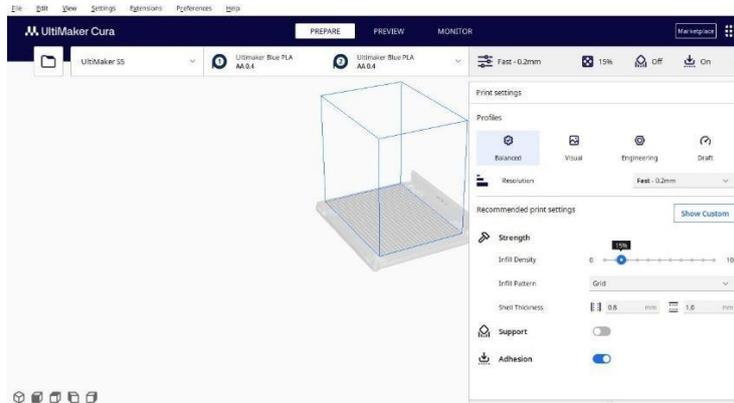


Figura 47 Software Ultimaker Cura

2.4.2.7 Braille Gcode Generator

Este programa sirve para convertir el texto ingresado a lenguaje CNC en códigos G, lo que le permite al usuario interactuar de manera sencilla enviando letras, números y caracteres para imprimir en sistema braille. Es de código abierto basado en lenguaje C# realizado en un entorno de desarrollo Visual Studio. Así también, el código facilita los cambios necesarios para el funcionamiento preciso de la impresora Braille. Este programa es el software de comunicación entre el usuario y la impresora que permite ingresar texto de manera sencilla en la interfaz y que produzca la impresión de los caracteres braille.

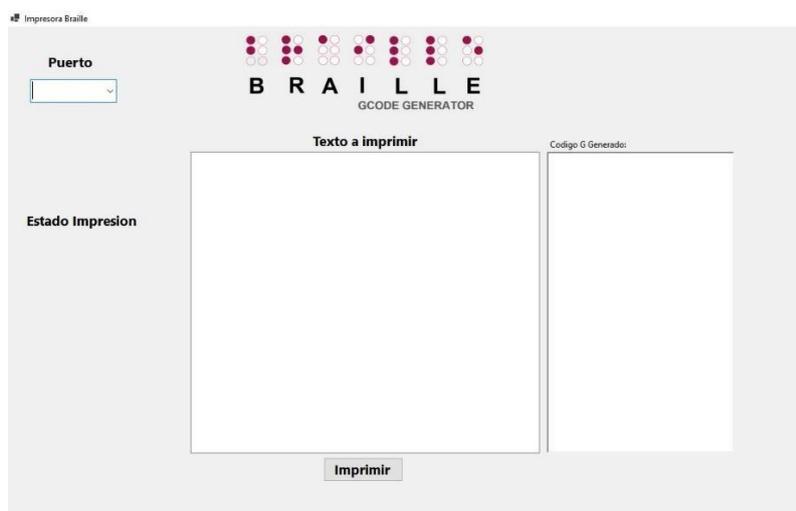


Figura 48 Braille Gcode Generator

2.5 DISEÑO DE LA PROPUESTA

Luego de la selección de los componentes físicos y lógicos que permiten cumplir con el desarrollo del proyecto se establece las conexiones entre los diferentes dispositivos.

2.5.1 Diagrama de conexiones de la impresora braille

En el diagrama de la figura 49, se visualiza las conexiones físicas de la impresora braille, el cual consta de la placa de Arduino UNO, una placa de control CNC Shield, los drivers A4988, los motores de paso a paso, solenoide y una fuente de alimentación de 24 VDC.

El Arduino UNO por medio del firmware grbl se encarga de interpretar los comandos en código G para convertirlos en el control de los motores y la activación del solenoide. La conexión de la placa al ordenador es a través del puerto USB, el cual alimenta la placa con 5V y también permite tener la conexión serial para la transferencia de los datos desde la interfaz de Visual Studio.

La fuente externa de 24 VDC permite distribuir la alimentación para los motores paso a paso de 12 VDC y el solenoide que trabaja a 24 VDC, la placa CNC Shield dispone de los pines para adaptar los drivers y conectar los motores, así también un puerto para conectar el pin digital que envía la señal 5 V o 0 V, para activar y desactivar el solenoide respectivamente.

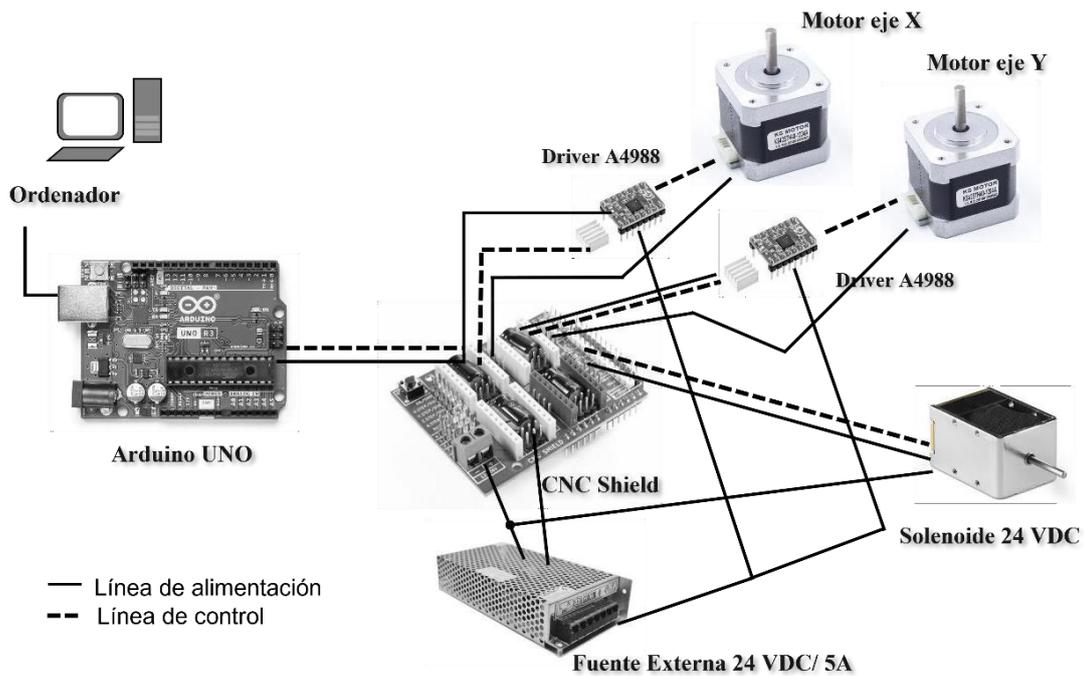


Figura 49 Esquema de conexiones. Elaborado por el autor

2.5.2 Diseño Físico de la impresora braille

Para el diseño 3D de la impresora braille se lo realizó en el software SolidWorks, donde se diseñó las partes que componen la estructura, entre ellos están: los soportes para instalar los motores y los ejes transversales, el carrito donde se encaja el actuador solenoide y la base de la impresora. Las dimensiones totales se especifican en la tabla 9, para realizar la base se eligió material del tipo MDF, por su facilidad para moldear y realizar los cortes a medida, los soportes se obtuvieron a través de impresión 3D, creados en la Ultimaker S5 hechos de material PLA. Se instalan dos ejes paralelos entre los soportes los cuales permiten el desplazamiento del carrito donde se sitúa el solenoide y otro eje que sirve para la rotación del rodillo al ingreso de papel (ver figura 50).

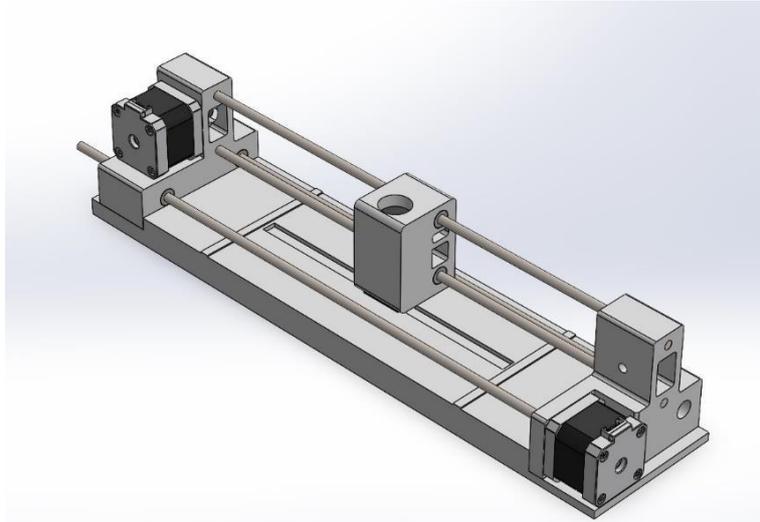


Figura 50 Diseño 3D impresora braille SolidWorks. Elaborado por el autor

En la figura 51 se observa el diseño final de la impresora con todos los materiales y elementos que hacen posible el movimiento de los motores y el accionamiento del solenoide para la impresión de caracteres braille.

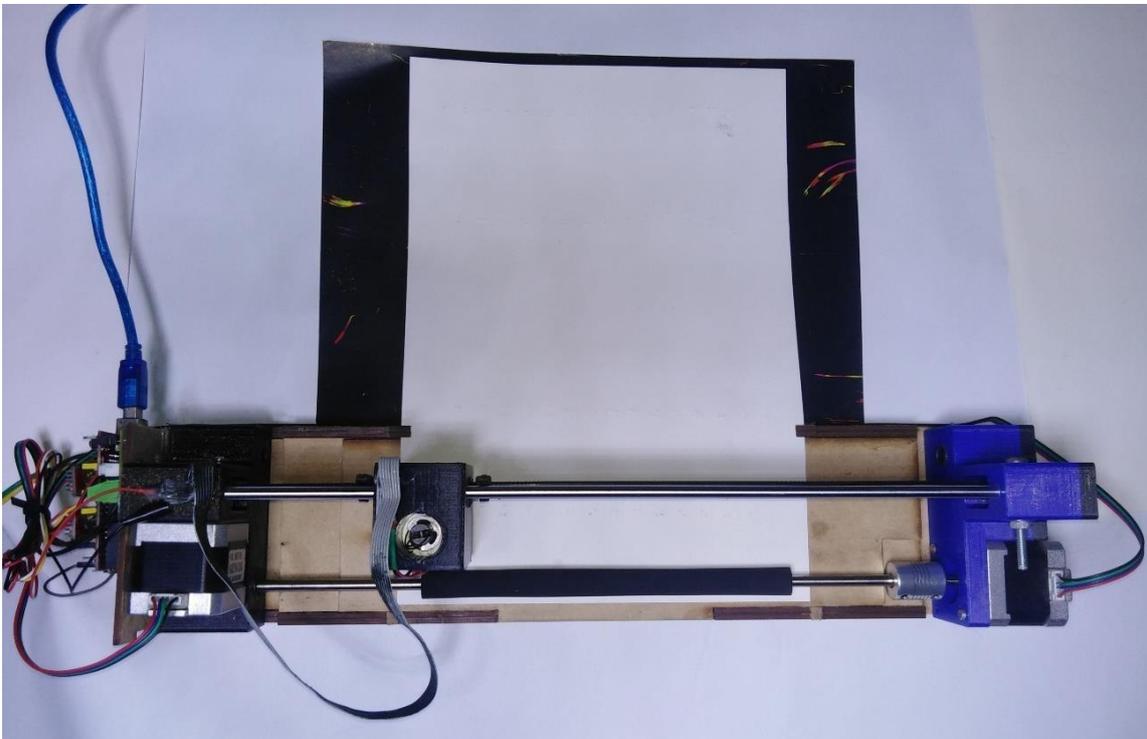


Figura 51 Implementación impresora braille. Elaborado por el autor

Dimensiones Impresora Braille	
Largo total (L)	41 cm
Ancho total (W)	10 cm
Alto total (H)	13 cm

Tabla 9 Dimensiones impresora braille. Elaborado por el autor

Entre los diseños 3D que se realizaron están el soporte lateral derecho y el soporte lateral izquierdo, cada soporte tiene agujeros precisos donde encajan los rodamientos lineales sobre los que se instalan los ejes para el carrete, así también los agujeros donde se instalan las poleas de la correa dentada y los huecos para ajustar los motores con tornillos. Para que los soportes estén sujetos a la base se ponen tornillos en la parte inferior entre la base y los soportes de manera que se forme una estructura rígida y estable para el movimiento tanto del rodillo como el carrete. (ver figura 52).

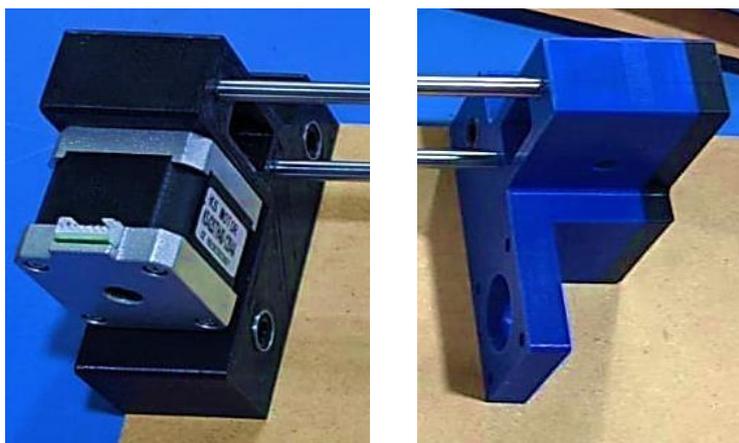


Figura 52 Soporte lateral izquierdo y soporte lateral derecho. Elaborado por el autor

Para el solenoide se diseñó una pieza en impresión 3D en donde encaje perfectamente y que a su vez se pueda desplazar de manera paralela a los ejes, la pieza tiene agujeros donde se instalan los rodamientos lineales y por donde atraviesan los ejes, además tiene una ranura por donde se ajusta la correa dentada (ver figura 53).

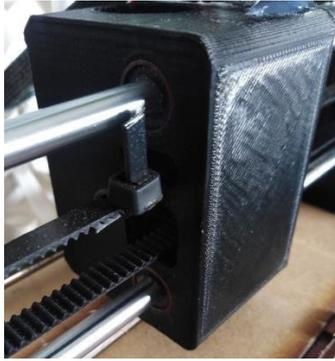


Figura 53 Estructura para solenoide. Elaborado por el autor

La base de la estructura está diseñada en 3D en SolidWorks y hecha de material MDF, tiene una longitud de 43 cm, ancho de 10 cm y una altura 2 cm, los cortes fueron realizados mediante tecnología de corte láser para tener las dimensiones precisas obtenidas del diseño 3D, esta base mantiene la estructura y permite el desplazamiento del papel. La base también tiene dos límites laterales que sirven para guiar el papel estrictamente las medidas del ancho de papel es de 21 cm. Así también la base tiene un devanado triangular en donde se ubica una placa de madera de balsa de 26 cm de longitud por 3 mm alto y 1.2 cm de ancho que sirve para recibir las punzadas accionadas por el solenoide permitiendo que se plasmen los puntos sobre el papel de manera legible (ver figura 54).



Figura 54 Diseño de base realizado con corte laser en MDF. Elaborado por el autor

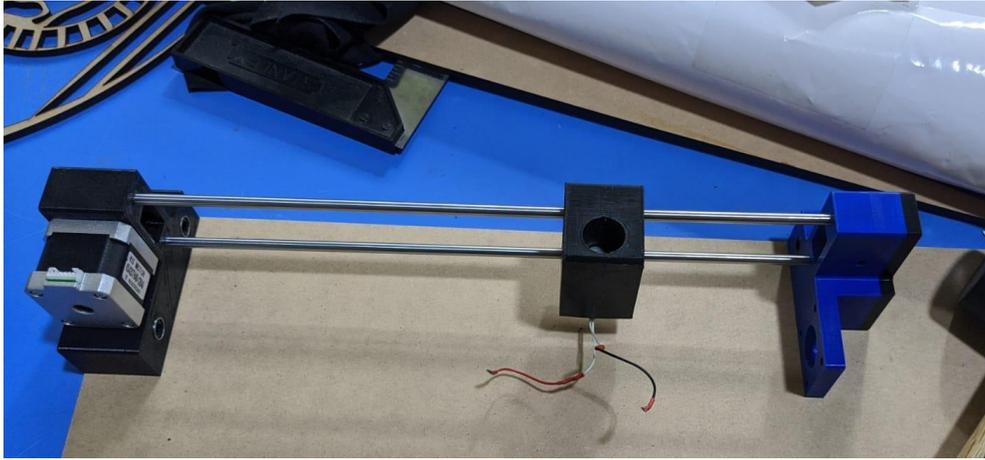


Figura 55 Diseño de base en MDF. Elaborado por el autor

Los motores de paso a paso se ubican en los bordes laterales de la base:

El primer motor se instala sobre el soporte lateral derecho y se une al eje del rodillo con un acoplamiento flexible, este motor es el responsable de mover la hoja a través de la rotación de los rodillos incorporados sobre el eje, el acoplamiento evita los problemas de alineación entre el eje del motor y el eje del rodillo, lo que ayuda a tener un desplazamiento constante y estable de la hoja sobre la base, tal como se ve en la figura 56.

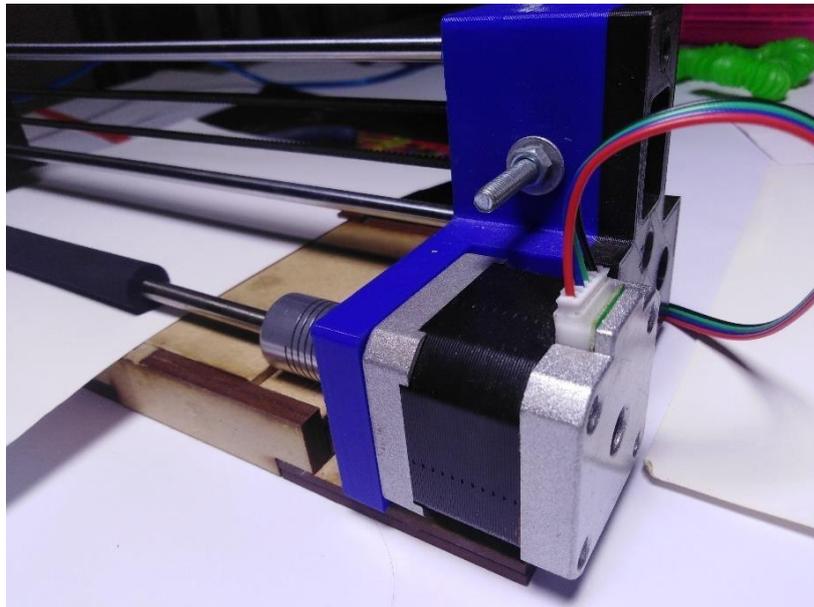


Figura 56 Instalación de motor sobre soporte izquierdo. Elaborado por el autor

El segundo motor paso a paso se ubica sobre el soporte lateral derecho, al motor se acopla una polea GT2 y sirve para montar una correa dentada GT2, que es la responsable de

mover el carrete sobre los ejes que los sostiene. El movimiento del motor produce el desplazamiento preciso del carrete, la correa está montada con un ajuste correcto entre los dos extremos (ver figura 57).

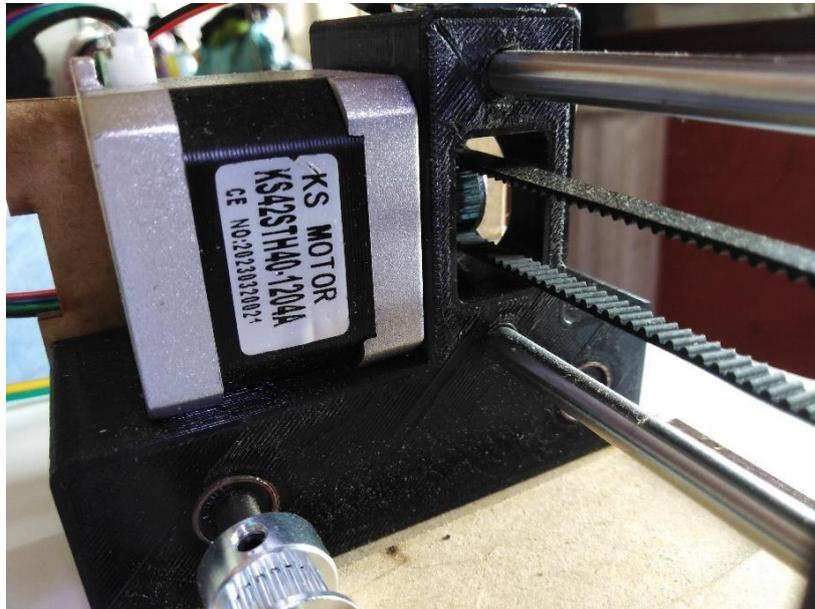


Figura 57 Instalación de motor sobre soporte derecho. Elaborado por el autor

2.5.3 Diseño electrónico de la impresora braille

En esta fase se muestran los diferentes circuitos electrónicos que interconectarán el Módulo Arduino Uno y la placa Shield CNC para el control de los motores de paso y el sistema de accionamiento del solenoide. Además, se empleó Altium Designer para diseñar los esquemas electrónicos, ya que es una herramienta muy útil para el diseño de cualquier circuito electrónico con una gran variedad de elementos electrónicos disponibles en su biblioteca.

2.5.3.1 Diseño de circuito para el solenoide

El solenoide cumple la función principal de realizar los puntos en relieve. Al ser un actuador electromecánico se debe controlar como elemento de potencia al encender y apagar la bobina del solenoide. De acuerdo a la tabla de especificaciones, el solenoide se alimenta a un voltaje de 24 VDC. Trabaja en modo ON/OFF al conectar y desconectar la

fuente. Para controlar el accionamiento del solenoide con una señal digital del Arduino UNO, es necesario implementar una etapa de potencia, mediante un circuito electrónico que ayude activar el solenoide que genere la fuerza lineal al momento en que se envía el voltaje alto o HIGH desde el microcontrolador.

El Transistor TIP 110 se usa como interruptor, maneja un corriente de 2 A y tiene un switcheo bastante rápido. Adicionalmente se añade un diodo en paralelo al solenoide que soporta una corriente de 1A y su función es proteger al transistor de la alta tensión generada cuando se apaga la bobina del solenoide. Así también se necesita una resistencia que proteja a la placa Arduino, limitando la corriente a 20 mA se puede utilizar la ecuación (1) para hallar el valor de la resistencia. Así se obtiene separar la etapa de potencia con el microcontrolador Arduino (ver figura 58).

$$R = \frac{V}{I} \quad (1)$$

$$R = \frac{5V}{20 \text{ mA}} = 250 \Omega$$

Se utiliza la resistencia de 220 Ω disponible en el mercado.

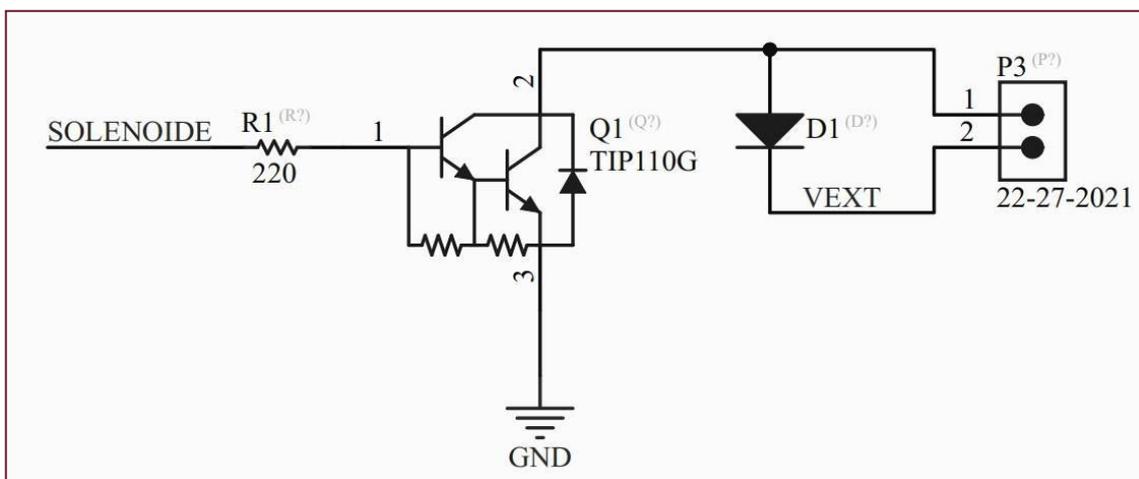


Figura 58 Circuito de potencia para el solenoide. Elaborado por el autor

2.5.3.2 Diseño circuito de control

Una vez seleccionados todos los elementos que conforman la impresora braille, se procedió a instalar el circuito de control con el circuito Shield seleccionado. Se requiere principalmente la placa Arduino UNO, la cual se conecta al CNC Shield junto con dos drivers A4988 para cada motor de pasos, además al CNC Shield se conectará el actuador solenoide.

Pin Arduino UNO	Nombre Pin Arduino	Elemento	Pin elemento
RESET	RESET	-	-
3V3	Salida de 3 VDC	CNC Shield	PIN 3V3
5 V	Salida de 5 VDC	CNC Shield Driver A4988 EJE X Driver A4988 EJE Y	PIN 5V VDD VDD
GND	GND	CNC Shield Driver A4988 EJE X Driver A4988 EJE Y	PIN GND GND GND
GND	GND	CNC Shield	PIN GND
VIN	Voltaje de entrada	CNC Shield	-
A0	Pin Analógico 0	CNC Shield	PIN Abort
A1	Pin Analógico 1	CNC Shield	PIN Hold
A2	Pin Analógico 2	CNC Shield	PIN Resume
A3	Pin Analógico 3	CNC Shield	PIN CoolEN
A4	Pin Analógico 4	CNC Shield	-
A5	Pin Analógico 5	CNC Shield	-
D0 (RX)	Pin Digital 0 (RX)	CNC Shield	PIN RX
D1 (TX)	Pin Digital 0 (TX)	CNC Shield	PIN TX
D2 (INT 0)	Pin Digital 2 (INT 0)	CNC Shield Driver A4988 EJE X	PIN X STEP
D3 (PWM, INT 1)	Pin Digital 3 (PWM, INT 0)	CNC Shield Driver A4988 EJE Y	PIN Y STEP
D4	Pin Digital 4	CNC Shield	PIN Z
D5 (PWM)	Pin Digital 5 (PWM)	CNC Shield Driver A4988 EJE X	PIN X. STEP/DIR DIR

D6 (PWM)	Pin Digital 6 (PWM)	CNC Shield Driver A4988 EJE Y	PIN Y. STEP/DIR DIR
D7	Pin Digital 7	CNC Shield	PIN Z. STEP/DIR
D8	Pin Digital 8	CNC Shield Driver A4988 EJE X Driver A4988 EJE Y	PIN EN/GND ENABLE ENABLE
D9 (PWM)	Pin Digital 9 (PWM)	CNC Shield	PIN X+
D10 (PWM)	Pin Digital 10 (PWM)	CNC Shield	PIN Y+
D11 (PWM)	Pin Digital 11 (PWM)	CNC Shield TIP 110	PIN Z+ PIN 1
D12	Pin Digital 12	CNC Shield	PIN SpnEn
D13 (SCK)	Pin Digital 13 (SCK)	CNC Shield	PIN SpnDir
GND	Pin GND	CNC Shield	PIN GND
AREF	Pin AREF	CNC Shield	-

Tabla 10 Relación de pines de Arduino UNO con placa CNC Shield. Elaborado por el autor

En la tabla 10, se puede visualizar los pines de Arduino UNO usados para conectar los componentes.

En la figura 59 se muestra el diseño esquemático del circuito diseñado en el software Altium Designer. Incluye el Arduino UNO, los controladores para cada motor de pasos, así también, el circuito de control del solenoide, conformado por el transistor y el diodo para el accionamiento del mismo

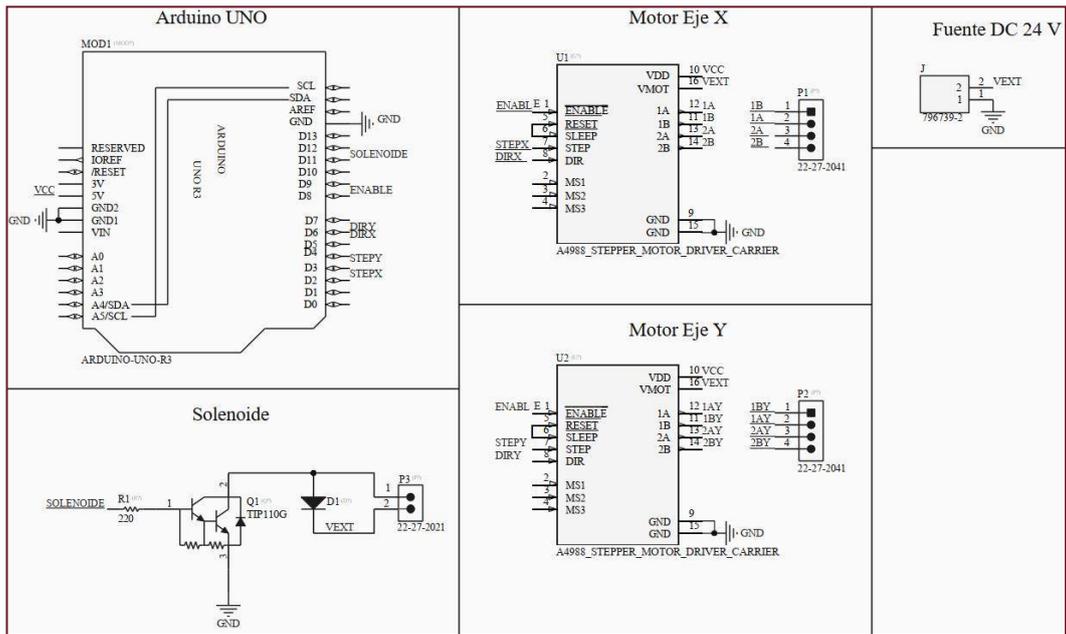


Figura 59 Diseño esquemático del circuito de control. Elaborado por el autor

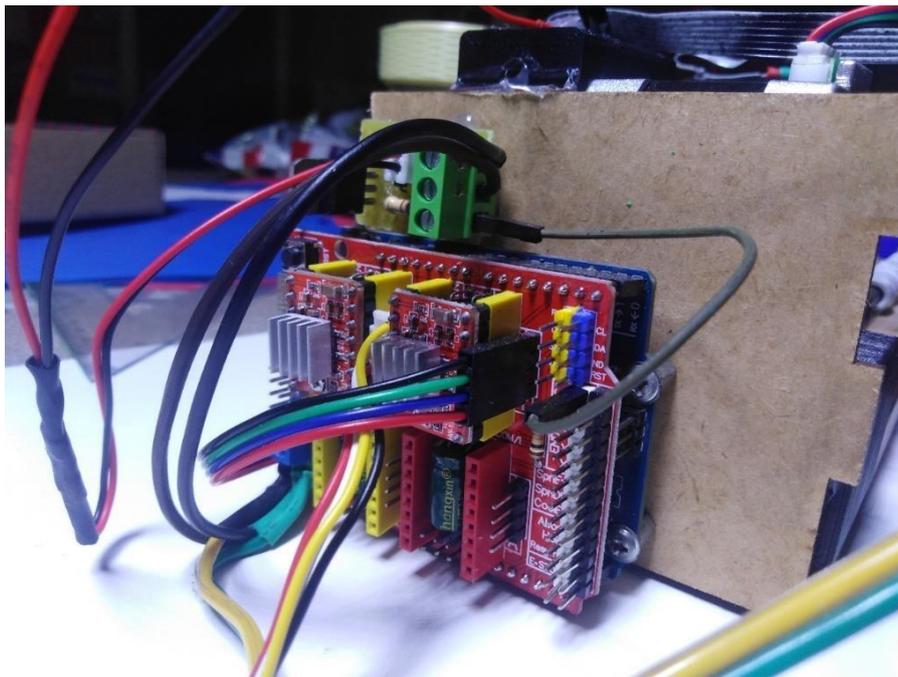


Figura 60 Circuito de control. Elaborado por el autor

2.5.4 Diseño y desarrollo del software de control y comunicación

Una vez estructurado el hardware se procedió a diseñar el funcionamiento que tendría el software que permite la interpretación de los caracteres braille convertidos en códigos G para accionar el actuador y el movimiento de los motores. La impresora braille será capaz de imprimir textos braille de manera automática al ingresar los textos por teclado en la interfaz Braille Gcode Generator.

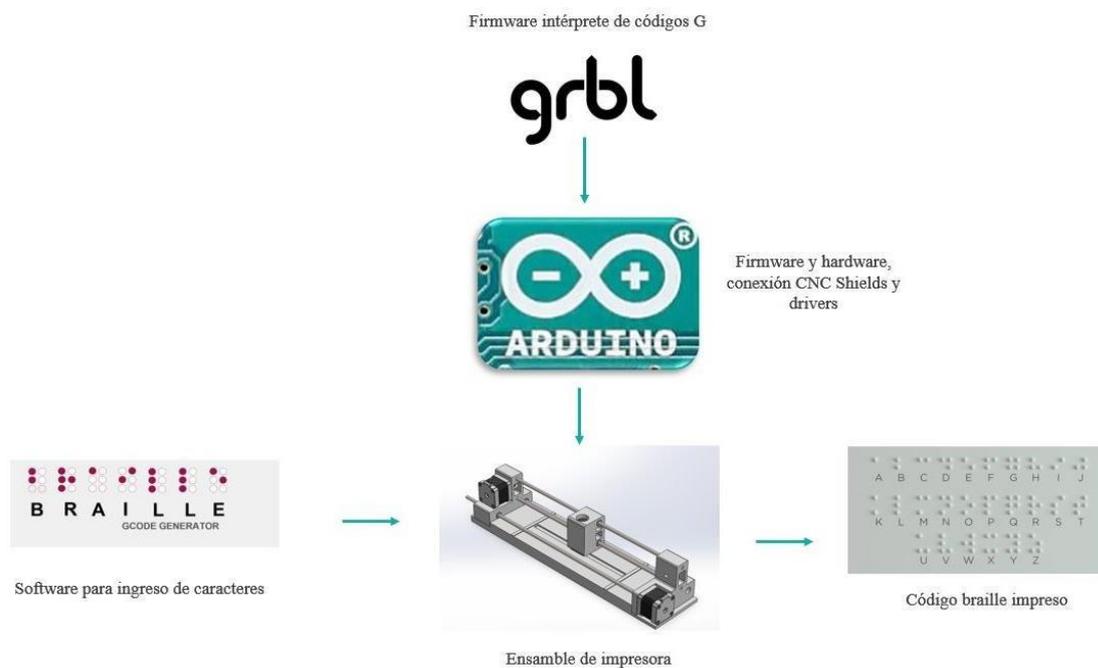


Figura 61 Proceso de control y comunicación. Elaborado por el autor

Para el funcionamiento de la impresora se usó la librería grbl v.0.9, es de código abierto y descarga libre desde GitHub, sirve para el control del movimiento de los motores de paso, compatibles con Arduino UNO, esta librería acepta códigos G que son interpretados dentro de la librería para que el módulo Arduino reciba las instrucciones y realice los movimientos correctos y precisos.

2.5.4.1 Programación Arduino

Para la programación de Arduino se realizó la compilación de la librería grbl usando el entorno de desarrollo integrado IDE de Arduino. Se añade la librería a la biblioteca para usarla dentro de la plataforma el comando para habilitarla es `#include grbl.h`, esta librería comprende el firmware para el control de máquinas CNC, tiene la mayoría de funciones para configurar el control y el desplazamiento de la máquina al transformar el archivo GCode en movimiento de los motores y accionamiento de los actuadores.

Es necesario subir la librería desde IDE Arduino *Programa* → *Incluir Librería* → *Añadir biblioteca .ZIP...*

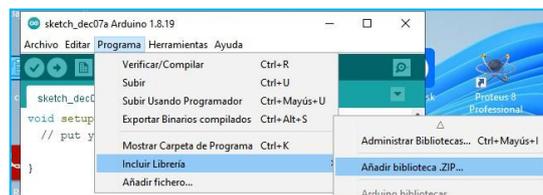


Figura 62 Ventana programa. Elaborado por el autor

Escoger el archivo ZIP desde el directorio que contiene el archivo `grbl.h`

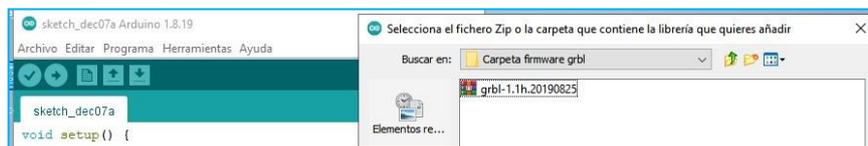


Figura 63 Ventana de archivo ZIP. Elaborado por el autor

Una vez cargada la librería a la biblioteca, se procede a crear un archivo ejemplo que contiene todas las funciones disponibles de firmware grbl v.1.1

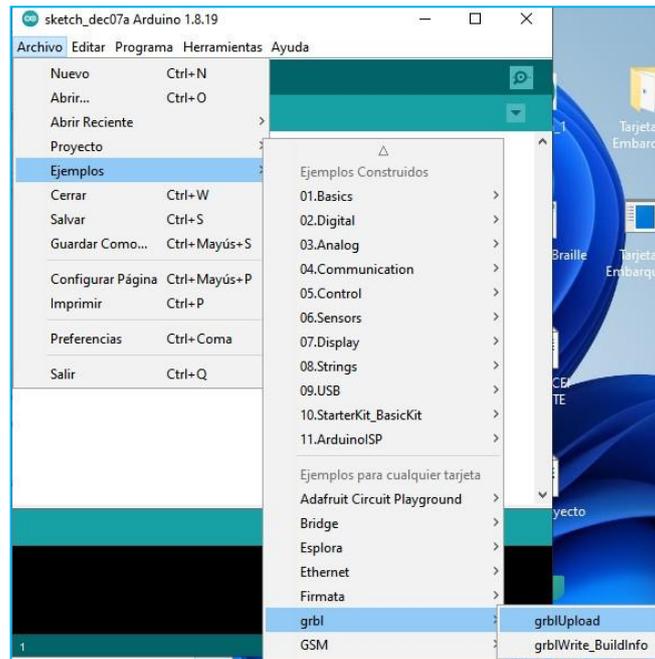


Figura 64 Ventana Ejemplos. Elaborado por el autor

Finalmente, se procede a compilar la librería en el IDE Arduino, se conecta la placa Arduino mediante el cable USB, es importante desconectar la placa CNC Shield mientras se sube el firmware grbl para no quemar la placa de control.

Una vez cargada la librería a la placa se procede hacer una configuración de parámetros de calibración, se puede intervenir de manera manual con comandos específicos para grbl en la ventana de *monitor serie* con un método de prueba y error en donde se realiza el test de calibración de los motores paso a paso. Así también, se puede realizar pruebas de forma más rápida con un software de programación CNC. Para tal fin se utiliza el Universal Gcode Sender, el cual facilita la configuración de los parámetros CNC.

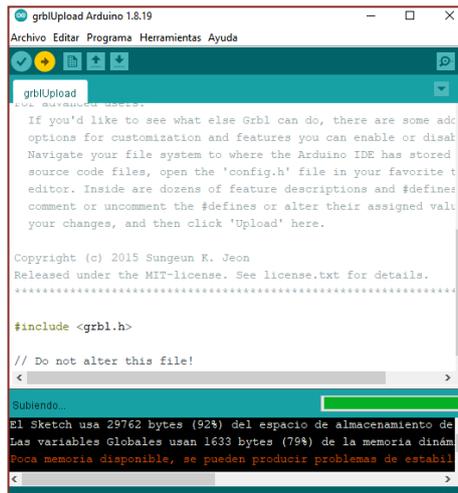


Figura 65 Librería compilada en la placa Arduino UNO. Elaborado por el autor

2.5.4.2 Calibración de parámetros iniciales CNC a través de Universal Gcode Sender

UGS es un software que se utiliza para enviar los comandos desde el ordenador a la maquina CNC, es compatible con la versión 0.9 del firmware grbl y se puede configurar la velocidad y resolución de pasos de los motores.

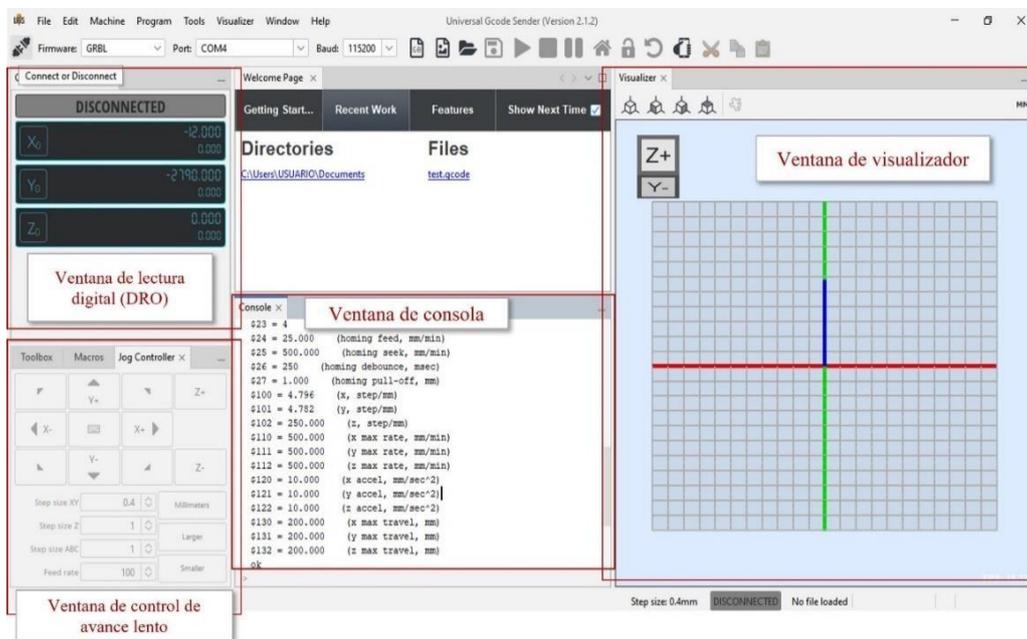


Figura 66 Interfaz gráfica UGS. Elaborado por el autor

En la figura 66 se observa la interfaz gráfica del programa, en esta se encuentra la ventana de consola, en donde se puede hacer la calibración de los movimientos de los ejes, adicional se usa la ventana *Jog Controller* para realizar las pruebas y comprobar si tuvo éxito la calibración. Las dimensiones braille están en milímetros por ende es necesario hacer un ajuste preciso para el movimiento de los motores en unidades milimétricas.

La ventana *Jog Controller* dispone de botones para mover en los ejes X, Y, Z, en la opción *Step size XY* se ingresa 1 y se toma como unidad de medida milímetros, la opción *Feed rate* es la velocidad a la cual se mueven y de momento se mantiene sin cambios. Posteriormente se pone en marcha el sistema en tiempo real y se obtiene los resultados que se muestran en la *Ventana de lectura digital*.

En la ventana de consola se muestran los valores de firmware grbl y en ella se realiza la configuración de los parámetros de calibración. Se ingresa los comandos \$\$ para desplegar el listado de parámetros configurables de la tabla (ver figura 67).

```

>>> $$
$0 = 10      (step pulse, usec)
$1 = 25      (step idle delay, msec)
$2 = 0       (step port invert mask:00000000)
$3 = 0       (dir port invert mask:00000000)
$4 = 0       (step enable invert, bool)
$5 = 0       (limit pins invert, bool)
$6 = 0       (probe pin invert, bool)
$10 = 3      (status report mask:00000011)
$11 = 0.010  (junction deviation, mm)
$12 = 0.002  (arc tolerance, mm)
$13 = 0      (report inches, bool)
$20 = 0      (soft limits, bool)
$21 = 0      (hard limits, bool)
$22 = 1      (homing cycle, bool)
$23 = 4      (homing dir invert mask:00000100)
$24 = 25.000 (homing feed, mm/min)
$25 = 500.000 (homing seek, mm/min)
$26 = 250    (homing debounce, msec)
$27 = 1.000  (homing pull-off, mm)
$100 = 4.796 (x, step/mm)
$101 = 4.782 (y, step/mm)
$102 = 250.000 (z, step/mm)
$110 = 500.000 (x max rate, mm/min)
$111 = 500.000 (y max rate, mm/min)
$112 = 500.000 (z max rate, mm/min)
$120 = 10.000 (x accel, mm/sec^2)
$121 = 10.000 (y accel, mm/sec^2)
$122 = 10.000 (z accel, mm/sec^2)
$130 = 200.000 (x max travel, mm)
$131 = 200.000 (y max travel, mm)
$132 = 200.000 (z max travel, mm)
ok

```

Figura 67 Tabla de parámetros para calibración UGS. Elaborado por el autor

Para variar los parámetros, se ingresa el dato a configurar, por ejemplo $\$102=200.000$ (z , $step/mm$) el cual ya viene configurado de fábrica y se coloca en la ventana consola el dato $\$102=250.000$ y se presiona tecla *enter*, luego para verificar el cambio se vuelve a escribir $\$$ e inmediatamente se despliega la lista de parámetros con los cambios realizados.

La impresora braille dispone de dos desplazamientos en el eje X y el eje Y, el movimiento del carrete y el rodillo respectivamente, el eje Z únicamente tiene la activación de la herramienta (solenoides), por ende, es necesario hacer la calibración de los pasos por mm de los motores en X, Y.

En los parámetros $\$100$ (x , $step/mm$), $\$101$ (y , $step/mm$), $\$102$ (x , $step/mm$) se calibra el número de pasos de los ejes X, Y, Z que sirven para mover un milímetro. Los parámetros adicionales no sufren ningún cambio, permanecen con la configuración de fábrica.

Para configurar se usa la herramienta *Jog Controller*, el cual permite hacer una serie de pruebas mientras se mueven los botones, el resultado obtenido es el siguiente:

- $\$100 = 4.796$ (x , $step/mm$)
- $\$101 = 4.782$ (y , $step/mm$)
- $\$102 = 250.000$ (z , $step/mm$)

Estos valores se instalan en el firmware grbl y está listo para poner en marcha. Por último, se debe configurar los valores *Offset*, o los valores *Zero* para iniciar la trayectoria (ver figura 68).

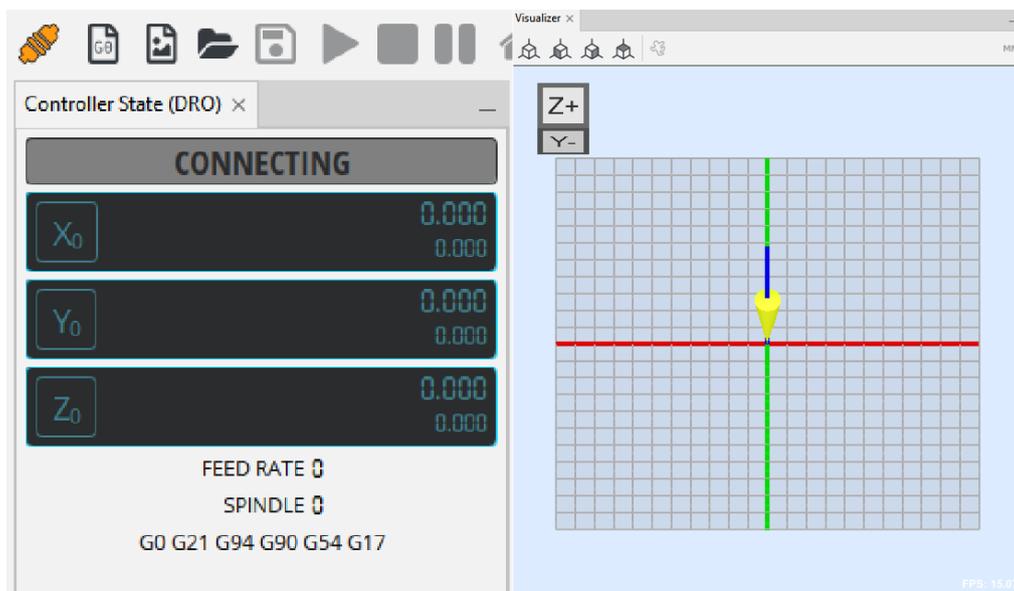


Figura 68 Punto Zero inicio de trayectoria en UGS. Elaborado por el autor

Para establecer estos puntos es necesario hacer una revisión visual sobre la impresora braille y definir el punto en donde va a iniciar a punzar el solenoide para generar los caracteres braille.

2.5.4.3 Comunicación entre el Computador y la impresora

Para lograr la comunicación entre el computador y la impresora, que permita generar los caracteres braille a partir del ingreso de texto por el teclado, se emplea el programa Braille Gcode Generator (ver figura 69), el cual está diseñado en lenguaje C# con Visual Studio. La lógica usada en el código permite convertir el alfabeto, números y signos especiales en coordenadas en códigos G para cada uno de los puntos que componen la matriz braille.

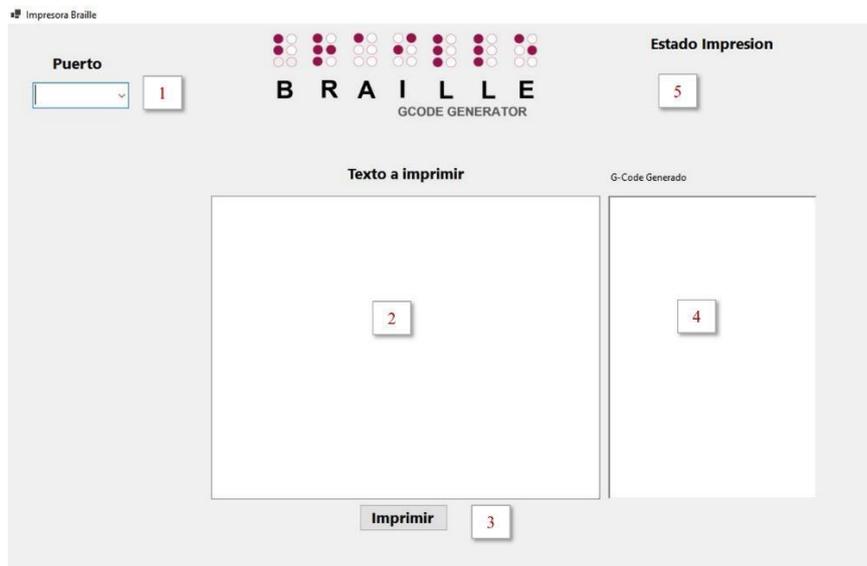


Figura 69 Interfaz Braille Gcode Generator (vista original). Elaborado por el autor

- 1.- Seleccionar el puerto de comunicación con la placa Arduino (Puerto COM).
- 2.- Espacio para ingreso de texto, números, signos especiales.
- 3.- Botón para enviar a imprimir los caracteres braille.
- 4.- Ventana de códigos G generados a partir del texto ingresado
- 5.- Estado de impresión

Estas son las partes que componen la interfaz, su aplicación es muy útil para tener una buena interacción entre el usuario con la impresora. En la figura 70, se presenta el diagrama de flujo que corresponde al algoritmo general en él se detalla la estructura y funcionamiento del software.

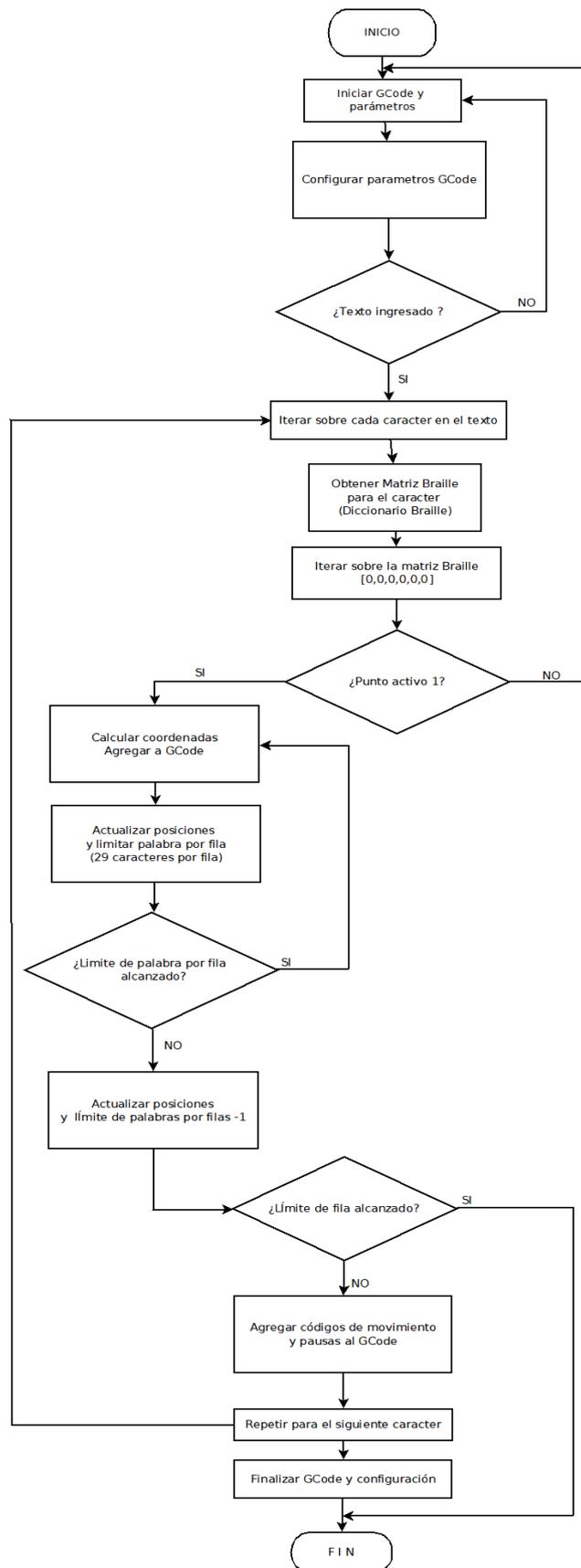


Figura 70 Diagrama de flujo (código general Braille Gcode Generator). Elaborado por el autor

Al iniciar, el programa se comienza con la inicialización de una lista GCode para almacenar las instrucciones de códigos G, se definen variables iniciales se expresan en la figura 71. Se consideran las dimensiones de la matriz braille.

```
List<string> gcode = new();  
  
double xOffset = 0.0;  
double yOffset = 10; //cambi  
double xIncrementWord = 6.0;  
double yIncrementRow = 5; //  
int limitWordRow = 29;  
double increment = 2.5;
```

Figura 71 Variables iniciales. Código C# Visual Studio

Se define tipo variable *double* e *int*, la primera para números decimales de tamaño de 64 bits y la segunda para número entero tamaño de 32 bits.

xOffset=0; *Punto de partida de carrete eje X*

yOffset= 10; *Punto inicial Y, la hoja es arrastrada por el rodillo 10 mm*

xIncrementalWord= 6; *Separación entre celdas de 6 mm*

yIncrementalRow= 5; *Separación entre filas de 5 mm*

limitWordRow= 29; *Límite de caracteres, 29 por fila.*

increment=2.5; *Distancia entre cada punto de la celda braille 2.5 mm*

Luego se configura los parámetros GCode, la configuración inicial consta de los códigos G expresados en la tabla 1.

```
// parametros iniciales de GCODE  
gcode.Add("G21;");  
gcode.Add("G90;");  
gcode.Add("G17;");  
gcode.Add("G94;");  
gcode.Add("G0 X0 Y0;");  
gcode.Add("G1 X0 Y0 F1000;");
```

Figura 72 Parámetros iniciales Gcode. Código C# Visual Studio

Al ingresar el texto, se itera sobre cada carácter del texto usando un bucle *foreach*, es cual es un tipo de bucle especial de C# utilizado para recorrer todos los elementos del texto. Para luego verificar el arreglo de la representación Braille de cada carácter utilizando una estructura de datos de tipo colección llamado Dictionary, en donde se realiza la correspondencia de los caracteres con la matriz braille.

```
BrailleChars = new Dictionary<char, int[]>
{
    {'a', new int[] {1, 0, 0, 0, 0, 0}},
    {'b', new int[] {1, 0, 1, 0, 0, 0}},
    {'c', new int[] {1, 1, 0, 0, 0, 0}},
    {'d', new int[] {1, 1, 0, 1, 0, 0}},
    {'e', new int[] {1, 0, 0, 1, 0, 0}},
    {'f', new int[] {1, 1, 1, 0, 0, 0}},
    {'g', new int[] {1, 1, 1, 1, 0, 0}},
}
```

Figura 73 Tipo colección Dictionary (Diccionario). Código C# Visual Studio

Si existe la representación del carácter se itera sobre la matriz Braille del carácter. Luego se calculan las coordenadas finales en función de los elementos de la matriz iguales a 1 para dar como resultado las coordenadas G que se almacenan en la lista Gcode.

Cada letra tiene una representación matricial almacenada en el Dictionary, la letra se recorre con *foreach*, luego se vincula la letra con la matriz braille correspondiente, extrae la matriz y recorre con un ciclo *for* entre los índices de la matriz, [6]= [0, 1, 2, 3, 4, 5], para evaluar los índices se utiliza una sentencia *Switch*, evalúa para cada *caso* (0), *caso* (1), *caso* (2), *caso* (3), *caso* (4), *caso* (5). Si el índice es igual a 1, asignar el valor para *finalX* / *finalY* obtenido desde las instrucciones para cada caso. Finalmente se añade los valores *finalX* / *finalY* a la lista Gcode mediante las instrucciones a continuación.

```
gcode.Add($"G1 X{finalX: F2} Y{finalY:F2};");
gcode.Add("M3 S1000;");
gcode.Add("G4 P0.2;");
gcode.Add("M3 S0;");
```

Estos son los códigos G para generar el carácter braille, la posición de ejes G1 X/Y, activación de solenoide M3 S1000, pausa G4 P0.2 y apagado de solenoide M3 S0.

Le sigue el proceso de actualizar las posiciones después de procesar cada uno de los caracteres, así también se limita la cantidad de palabras por fila que es de 29 caracteres.

Al no alcanzar el límite se procede a actualizar las posiciones y reducir -1 al valor de límite de palabras por fila.

```
xOffset += xIncrementWord;  
LimitWordRow--;
```

Figura 74 Actualización xOffset y conteo descendente -1. Código C# Visual Studio

Cuando el conteo límite de palabras por filas == 0, se procede a actualizar la posición en X/Y. La variable yOffset se almacena el valor de la variable constante yIncrementRow= 5.00 + el valor inicial que se asignó de 10.00. El nuevo valor para xOffset= 0.00; para regresar a su estado de partida y se reinicia el conteo en 29 caracteres por fila (ver figura 75).

Mientras el número de caracteres sea igual o menor a 29 el proceso concluye enviando la orden para que el motor saque la hoja con el texto braille impreso. Cuando el número es mayor a 29 caracteres, primero se realiza los primeros 29 que corresponderían a una primera fila, los caracteres restantes se realizan en las siguientes filas según el tamaño del texto. Para realizar la impresión de los caracteres restantes los motores se mueven coordinados en simultaneo hasta ubicar el carrito en posición de partida y el motor eje Y arrastra la hoja hasta posicionarla en 5 mm de separación entre la primera fila y la segunda fila.

```
if (LimitWordRow == 0)  
{  
    yOffset += yIncrementRow;  
    xOffset = 0.00;  
    LimitWordRow = 29;  
}
```

Figura 75 Límites de fila. Código C# Visual Studio

EL código concluye con la asignación de los códigos G, para arrastrar la hoja hasta terminar saliendo del rodillo. Se envía la distancia total del desplazamiento Y del rodillo se asigna a un valor de 310 mm el cual es el valor del cálculo de la dimensión de la hoja y a X =0.00 para mover el carrito hasta punto de partida o reposo. Luego se escribe posición de offsets en la coordenada 0 expresada mediante el código L20 Y0 y finalmente se reinicia la posición de X/Y en cero y se envían a la lista Gcode.

```
gcode.Add("G1 X0.00 Y310.00;");
```

```
gcode.Add("G10 P0 L20 Y0;");
```

```
gcode.Add("G0 X0 Y0;");
```

```
return string.Join("\n", gcode);
```

Esta serie de instrucciones en el código del programa se guardan en la lista Gcode que es enviada al Arduino por el puerto COM, la velocidad de transferencia de los datos es de 115200 baudios utilizado en la comunicación de las maquinas CNC. Al recibir los códigos G, el firmware grbl instalado en el Arduino interpreta las instrucciones y las traduce en los movimientos de los motores y el accionamiento del solenoide para obtener los textos braille.

2.6 COSTOS

Costos de materiales de estructura

N°	DESCRIPCION	CANTIDAD	VALOR
1	Base MDF	1	\$ 10.00
2	Impresiones 3D	3	\$ 31.36
3	Ejes de acero diámetro 6mm/largo 400 mm	3	\$ 13.39
4	Polea GT2 20 dientes diámetro 6.35 mm	2	\$ 5.36
5	Acople Flexible 6.35 mm a 5 mm	1	\$ 3.57
6	Correa dentada GT2 ancho 6 mm	1 mt	\$ 2.68
7	Rodamiento lineal LM6UU diámetro 6 mm	5	\$ 12.05
8	Tornillos sujetadores	6	\$ 1, 25
9	Tornillo de estrella 2 pulg. diámetro 6 mm	1	\$ 0,25
10	Pegamento y cinta	1	\$ 2,00
TOTAL			\$ 81.91

Tabla 11 Costos de materiales de estructura. Elaborado por el autor

Costo de mano de obra

DESCRIPCION	HORAS	VALOR POR HORAS
Trabajo de ingeniería y armado	100	\$3
TOTAL		\$ 300

Tabla 12 Costos de mano de obra. Elaborado por el autor

Costos de materiales electrónicos

N°	DESCRIPCION	CANTIDAD	VALOR
1	Arduino UNO Atmega 328P + cable USB	1	\$20.00
2	Motores NEMA 17 42STH40-1204A + cable	2	\$ 28.12
3	CNC Shield V3	1	\$ 5.50
4	Driver A4988	2	\$ 5.00
5	Solenoides 24 VDC	1	\$ 14.00
6	Fuente Externa 24 VDC 5 A	1	\$ 24.00
7	Transistor TIP110	1	\$ 1.50
8	Resistencias	1	\$ 0.05
9	Diodo IN4002	1	\$ 0.15
10	Bornera	1	\$ 0.40
11	Molex	1	\$ 0.40
12	Cables para conexiones	2 mt	\$ 4.00
13	Cable para alimentación 110 V	1 mt	\$ 2.00
TOTAL			\$ 105.12

Tabla 13 Costos de materiales electrónicos. Elaborado por el autor

Costo final

DESCRIPCION	VALOR
Costos de materiales de estructura	\$ 81.91
Costo de mano de obra	\$ 300
Costos de materiales electrónicos	\$ 105.12
TOTAL	\$ 487.03

Tabla 14 Costo final. Elaborado por el autor

2.7 PRUEBAS Y RESULTADOS

2.7.1 Pruebas y resultados del análisis del código Braille

Durante el proceso de estudiar el código braille se realizó un análisis a detalle para comprender la disposición gráfica y táctil de las letras y números. En específico el análisis se centró en identificar los caracteres equivalentes en braille.

Así también, estudiar las estructuras ortográficas y gramaticales que rigen el código braille, se consultó las fuentes bibliográficas con la normativa y guías especializadas.

Simbología del código braille

Las pruebas realizadas para llevar a cabo la investigación de la simbología asociada a cada letra del alfabeto y los números del código braille incluyeron revisión bibliográfica de fuentes especializadas en donde se definen la representación visual y táctil de cada carácter braille. Así también, se realizó un examen directo de la regleta braille 29 caracteres por 4 líneas, que sirvió para identificar y comprender la disposición de los puntos del signo generador braille. Así también se identificó la cartulina Bristol A4 perfecta para obtener la legibilidad de los caracteres. Este examen se logró mediante la colaboración la profesora Jackeline Quimí, especializada en educación a niños con discapacidad visual de la Fundación Melvin Jones de la Libertad, Santa Elena.



Figura 76 Regleta Braille 29 x 4. Elaborado por el autor

Conversión de letras y Números

Se examinaron las especificaciones para la conversión para cada letra y número en código braille, se realiza el estudio de los puntos que conforman la matriz braille, de ese modo identificar la posiciones que tiene el signo generador del 1 al 6. Mediante el uso de la regleta braille se realizó las pruebas para obtener los caracteres braille, es necesario reconocer las posiciones para poder punzar los puntos correctos, ya que al usar la regleta se toman los puntos invertidos, los números del 1 al 3 van en la fila de la derecha y los puntos 4, 5 y 6 en el lado izquierdo, de ese modo se obtiene el relieve de los códigos braille para poder hacer la lectura.

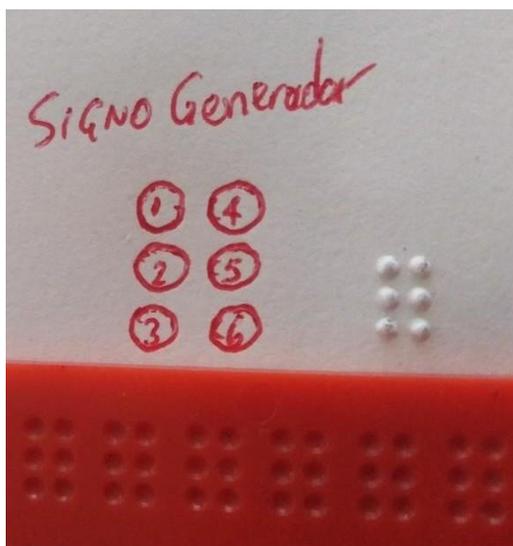


Figura 77 Matriz braille posiciones del signo generador Elaborado por el autor

Investigación de las Estructuras Ortográficas y Gramaticales

Las pruebas incluyen una exploración al Documento técnico de la Comisión Braille Española, en el mismo se muestran los signos braille específicos para el uso de mayúsculas, minúsculas, signos de puntuación y vocales tildadas, estas corresponderían a las estructuras ortográficas y gramaticales que se deben respetar en el código braille. Del mismo modo se hace estudio de textos braille realizados por estudiantes no videntes de la Fundación Melvin Jones en los que se pueden visualizar los errores comunes de la escritura braille.

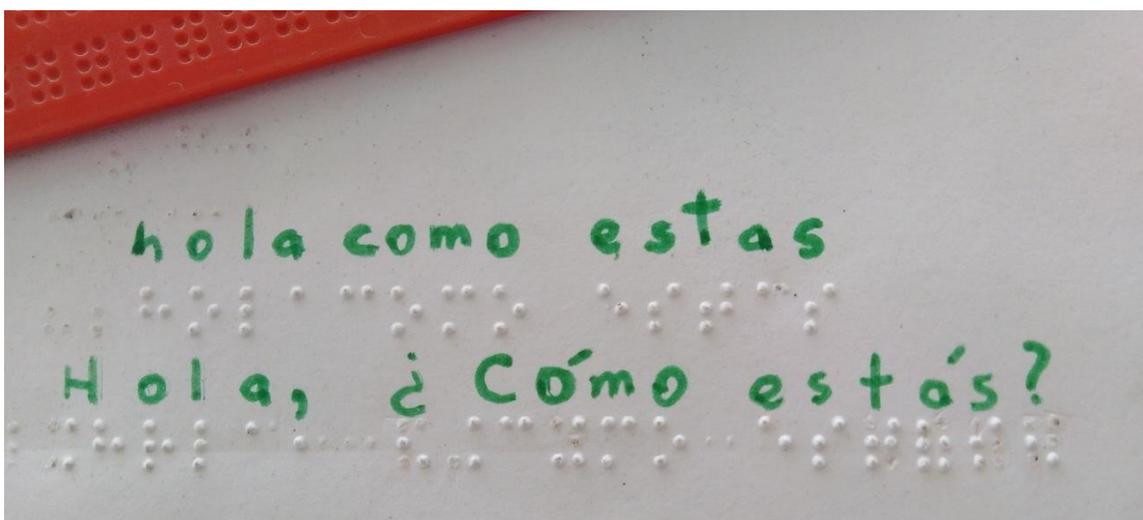


Figura 78 Errores ortográficos y gramaticales. Elaborado por el autor

La figura 78, se analizó la primera oración la cual contiene varios errores ortográficos, reconocidos al hacer la correspondencia de los signos braille en letras y signos. La segunda oración cumple la estructura gramatical sin faltas ortográficas.

Como resultado de las pruebas, se logró una identificación muy precisa de los caracteres braille correspondientes a las letras, números y los signos de puntuación. Esto proporciona una base sólida para el sistema de conversión en la impresora braille (ver tablas 15 y 16) utilizadas para la implementación posterior.

Letras Minúsculas/ Mayúsculas		Conversión braille		Letras Minúsculas/ Mayúsculas		Conversión Braille		Números	Conversión braille
a	A			Ñ	Ñ			1	
b	B			O	O			2	
c	C			P	P			3	
d	D			Q	Q			4	
e	E			R	R			5	
f	F			S	S			6	
g	G			T	T			7	
h	H			U	U			8	
i	I			V	V			9	
j	J			W	W			0	
k	K			X	X				
l	L			Y	Y				
m	M			Z	Z				
n	N								

Tabla 15 Correspondencias Letras y números en braille. Elaborado por el autor

Signos de puntuación	Conversión braille	Signos de puntuación	Conversión Braille	Signos de puntuación	Conversión Braille
Coma	⠠⠠ ⠠⠠ ⠠⠠	Abrir paréntesis	⠠⠠ ⠠⠠ ⠠⠠	Dieresis	⠠⠠ ⠠⠠ ⠠⠠
Punto	⠠⠠ ⠠⠠ ⠠⠠	Cerrar paréntesis	⠠⠠ ⠠⠠ ⠠⠠		
Punto y coma	⠠⠠ ⠠⠠ ⠠⠠	Comillas	⠠⠠ ⠠⠠ ⠠⠠		
Dos puntos	⠠⠠ ⠠⠠ ⠠⠠	Asterisco	⠠⠠ ⠠⠠ ⠠⠠		
Interrogación	⠠⠠ ⠠⠠ ⠠⠠	Guion	⠠⠠ ⠠⠠ ⠠⠠		
Exclamación	⠠⠠ ⠠⠠ ⠠⠠	Arroba	⠠⠠ ⠠⠠ ⠠⠠		

Tabla 16 Correspondencias signos de puntuación en braille. Elaborado por el autor

2.7.2 Pruebas y resultados de la revisión de dispositivos electrónicos existentes

Se realizó una investigación de los dispositivos electrónicos diseñados para la educación de personas con discapacidad visual. Con el fin de explorar los diversos equipos tecnológicos existentes y comprender acerca de la contribución de estos al desarrollo educativo.

La revisión de los dispositivos electrónicos incluye en preferencia impresoras de códigos braille, las cuales sirven para obtener textos para personas no videntes y utilizarlos en los procesos educativos.

Los resultados de esta revisión se muestran en un cuadro comparativo entre tres de los dispositivos electrónicos utilizados para personas no videntes basados en impresión de códigos braille.

Cuadro comparativo ventajas y desventajas de impresoras Braille

	View Tiger Embraille	Index Basic D-V5	Index Everest D-V5
			
Ventajas	<ul style="list-style-type: none"> -Ligera y compacta - Ajuste de hundido de puntos -Compatible Windows 1, 8/8.1, 10 pro -Conexión USB 	<ul style="list-style-type: none"> -Impresora compacta - Muy vendido - Conexión wifi - Impresión doble cara -Velocidad de impresión 100 caracteres por segundo 	<ul style="list-style-type: none"> -Impresora muy potente -Alta velocidad de impresión 140 caracteres por segundo -Conexión wifi -Programa de instalación desde www.indexbraille.com -Driver compatible Windows 7, 8, 10. Y Mac OS X 10.8 - Asistente IP fija mediante cable de red
Desventajas	<ul style="list-style-type: none"> -Impresión a una cara -Costo 2.112,00 euros (\$ 2330.15) -Sin acceso inalámbrico Wifi 	<ul style="list-style-type: none"> - Tipo de papel plegado acordeón -Costo 3.646,12 dólares estadounidenses 	<ul style="list-style-type: none"> -Tamaño para uso empresarial -Costo 4.395,00 dólares estadounidenses

Estos dispositivos electrónicos se encuentran disponibles en el mercado y ayudan a obtener textos braille, sin embargo, el alcance a ellas es una desventaja enorme para todo tipo de personas incluso para las instituciones que se dedican a educación de personas no videntes, en la provincia de Santa Elena no se dispone de ninguna de ellas, esto aumenta la brecha de inclusión educativa de las personas no videntes.

La implementación de este proyecto de impresora braille representa un 79.10 % de reducción en el costo tomando como referencia la impresora View Tiger Embraille siendo esta última la más económica de entre las impresoras braille comerciales.

2.7.3 Prueba y resultados de la identificación de herramientas de Hardware

Durante la fase inicial del proyecto, se realizó pruebas exhaustivas para identificar opciones de hardware disponibles para construir la impresora braille. El propósito fue seleccionar las mejores alternativas para cumplir con los requisitos del proyecto. Se identificó dos opciones de hardware.

Impresora de tinta en desuso

Una impresora de tinta tiene un sistema mecánico de motores para mover la hoja de papel y el movimiento de un carrete para inyectar la tinta. En la primera etapa de la investigación se obtuvo una impresora de tinta continua en desuso con el objetivo de reutilizar el sistema del movimiento de los motores, quitar los cartuchos para usar el carrete e incorporar un conjunto de sensores. El modelo que se usaría para las pruebas es el Canon PIXMA iP1900.



Figura 79 Impresora de tinta en desuso. Elaborado por el autor

Máquina CNC

Una segunda opción de hardware es una máquina CNC la cual tiene un hardware compacto y estable, basado en ejes y rieles con motores paso a paso, que se encuentran en impresoras 3D o máquinas de corte laser. Es tecnología con bastante auge actual por lo tanto existe muchas piezas y componentes que se venden por separado para construcción de las máquinas.

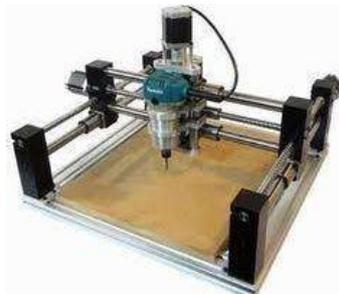
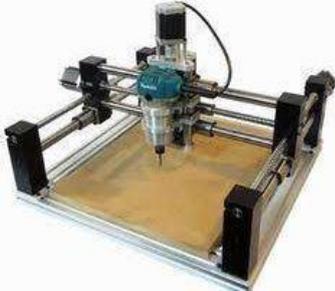


Figura 80 Máquina CNC para taladro

Fuente: <https://stores.2023saleonline.ru>

Los resultados obtenidos de estas pruebas sirven para seleccionar el hardware para implementación del proyecto, a continuación, se presenta un cuadro comparativo entre estas dos opciones de hardware.

Cuadro comparativo para seleccionar el hardware

	Impresora de tinta en desuso	Maquina CNC
		
Calidad de componentes	<p>-Tienen componentes que se pueden reemplazar, son de calidad estandarizada para equipos de impresión de tinta continua con marcas como EPSON, CANON, HP.</p> <p>- Respecto a la estructura tienen desgaste por ser un equipo en desuso.</p>	<p>-Son de gran calidad, desarrollado en materiales metálicos, estructuras y soportes en impresión 3D de material PLA.</p>
Precisión de componentes	<p>-Al ser una impresora en desuso los componentes están ya desgastados, pierden precisión.</p>	<p>-Tienen muy buena precisión son componentes estandarizados para construcción de máquinas CNC e impresoras 3D.</p>
Precio	<p>-Al ser una impresora en desuso su valor no es representativo, es fácil encontrarlas como desechos.</p>	<p>El precio varía en relación al tipo de aplicación: Maquina simple \$50 a \$200. Maquina mas compleja superior a los \$200.</p>
Disponibles	<p>-Según el tipo de versión o la continuidad del modelo de impresora muchos componentes no se encuentran en el mercado a la venta en el país.</p>	<p>-Los componentes se encuentran a la venta en gran cantidad de empresas dedicadas a impresoras 3D y componentes electrónicos.</p>

Basado en estos puntos el hardware seleccionado es el de maquina CNC, su construcción es de fácil aplicación, además incorpora un conjunto de materiales que se pueden conseguir en el mercado. Al ser tecnología de precisión basado en comandos G existen muchos sistemas y programas que se adaptan perfectamente y son compatibles con cualquier sistema operativo. Así también los componentes de control cumplen con el objetivo de desplazamiento de motores para la impresión braille, que en particular deben ofrecer una alta precisión para el punzado.

Con la adquisición de estos materiales de hardware se puede reducir el costo que representa una impresora comercial en un 91.9 %, por lo tanto, esta opción cumple con el objetivo planteado para determinar el hardware para la construcción de la impresora braille.

2.7.4 Pruebas y resultados del análisis de Software para conversión de Texto a Braille

En simultaneo a la identificación del hardware se llevó a cabo un análisis a detalle de los programas y código diseñados para convertir el texto en braille. Es un proceso muy importante ya que es necesario tener precisión en la conversión de los caracteres braille.

El código debe funcionar de manera que cada letra, signo de puntuación o número, tenga una representación compuesta de una matriz de 6 elementos, 2 filas por 3 columnas y dependiendo del carácter establecer los puntos a marcar.

Una vez que se escogió el hardware de maquina CNC se utilizó un software que utilice un generador de Códigos G, para ello se implementa el software Braille Gcode Generator el cual es un software basado en código abierto realizado en C# para transformar letras, números y signos en códigos G.

Como primer paso es configurar los parámetros G, en el Universal Gcode Sender, con el fin de encontrar una correspondencia exacta entre el desplazamiento del motor y la distancia recorrida, la unidad de medida son los milímetros para obtener precisión en las dimensiones braille.

Para calcular el porcentaje de eficiencia de la calibración de pasos por 1 mm, se divide el número de aciertos para la cantidad de muestras.

Para el eje X, se configuró el número de pasos para 1 mm.

N.º Prueba	Calibración de pasos	Medida tomada	N.º. Aciertos
1	250.000	52.13 mm	NO
2	10.000	2.09 mm	NO
3	5.000	1.04 mm	SI
4	4.900	1.02 mm	SI
5	4.800	1.00083 mm	SI
6	4.799	1.00062 mm	SI
7	4.798	1.00042 mm	SI
8	4.797	1.00021 mm	SI
9	4.796	1.00 mm	SI
10	4.795	0.999 mm	NO

Tabla 17 Correspondencias signos de puntuación en braille. Elaborado por el autor

$$\%Eficiencia = \frac{\text{cantidad de aciertos}}{\text{cantidad de muestras}} * 100\% \quad (2)$$

$$\%Eficiencia = \frac{7}{10} * 100\%$$

$$\%Eficiencia = 70 \%$$

Para el eje Y, se configuró el número de pasos para 1 mm.

N.º Prueba	Calibración de pasos	Medida tomada	N.º. Aciertos
1	250.000	52.28 mm	NO
2	5.000	1.046 mm	NO
3	4.900	1.0247 mm	SI
4	4.800	1.0038 mm	SI

5	4.799	1.0036 mm	SI
6	4.790	1.0018 mm	SI
7	4.788	1.0013 mm	SI
8	4.784	1.00042 mm	SI
9	4.782	1.00 mm	SI
10	4.780	0.999 mm	NO

Tabla 18 Correspondencias signos de puntuación en braille. Elaborado por el autor

$$\%Eficiencia = \frac{\text{cantidad de aciertos}}{\text{cantidad de muestras}} * 100\%$$

$$\%Eficiencia = \frac{7}{10} * 100\%$$

$$\%Eficiencia = 70 \%$$

Los resultados obtenidos de la calibración son del 70% para cada motor, cabe mencionar que el valor de 250.000 viene de fabrica con el firmware grbl. A partir de ese valor se debe ir calibrando el valor que mejor se acerca a la medida de 1mm.

La siguiente prueba es obtener las coordenadas G del programa Braille Gcode Generator para transformar el texto en caracteres braille con las correctas medidas braille para el carácter convertido. Se realizaron las pruebas de la conversión mediante los comandos G arrojados por el software y puestos en marcha en Universal Gcode Sender.

Para la letra C, se obtiene los siguientes comandos.

G21	Entrada de datos en milímetros
G90	Coordenadas absolutas intersecciones de los ejes X, Y
G17	Selección de plano de trabajo X, Y
G94	Velocidad de corte expresada en avance por minutos
G0 X0 Y0	Mover a una posición marcha rápida
G1 X0 Y0 F1000	Posición de partida 0, 0 velocidad de partida
G1 X0,00 Y10,00	Posición inicial de X= 0 mm y Y= 10,00 mm
M3 S1000	Activación de solenoide
M3 S0	apagado de solenoide
G1 X2,50 Y10,00	Desplazamiento en X = 2,5 mm y Y= 10,00 mm

M3 S1000	Activación de solenoide
P0.2	Pausa de 0.2 segundos
M3 S0	Apagado de solenoide
G1 X0.00 Y310.00	Regreso posición de partida de X= 0 mm y Y= mover hasta 310 mm salida de papel impreso
G0 X0 Y0	Posición de partida X=0 mm y Y=0 mm

Tabla 19 Comandos G para letra C. Elaborado por el autor

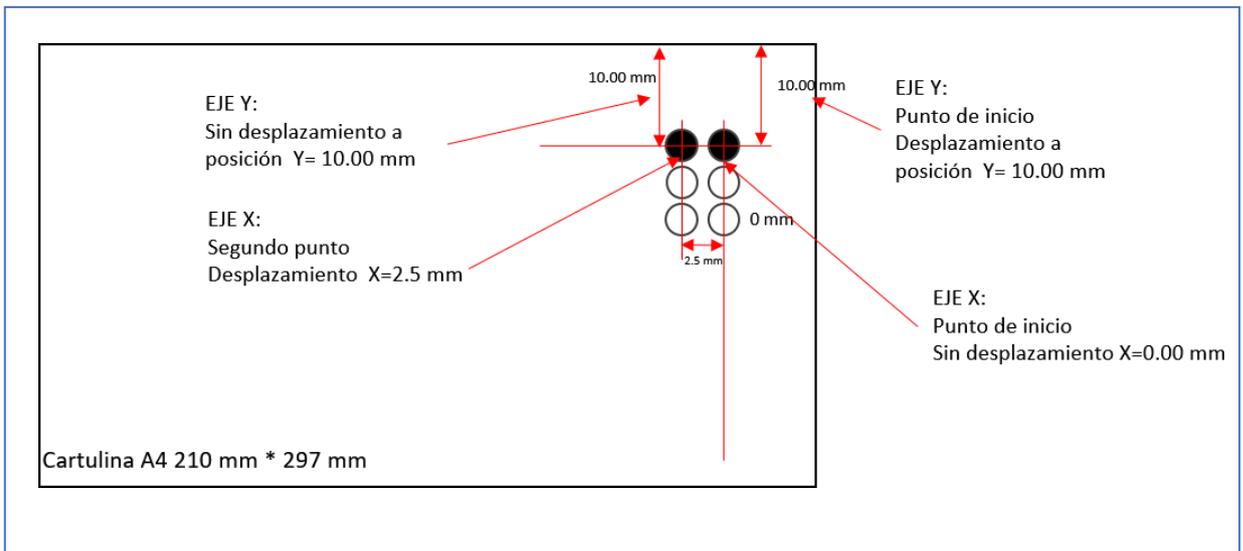


Figura 81 Generación de movimiento de ejes X, Y. Elaborado por el autor

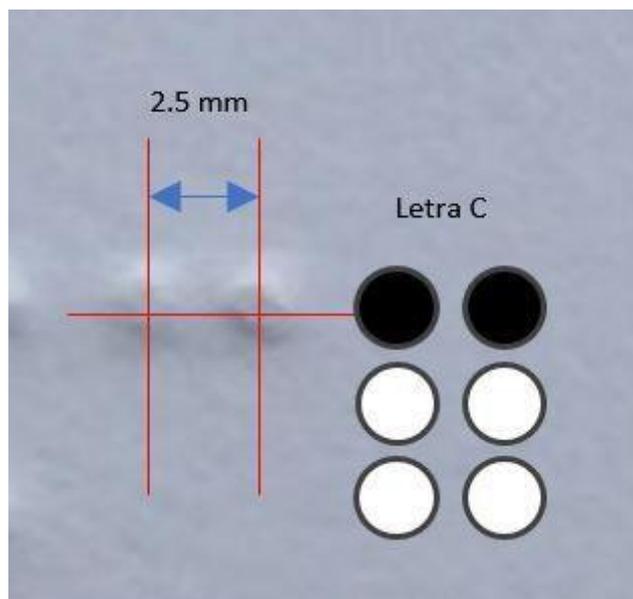


Figura 82 Impresión de letra C con comandos G. Elaborado por el autor

Los comandos G, recibidos por Braille Gcode Generator producen los desplazamientos de los motores paso a paso y la activación del solenoide. Este software es eficiente y cumple con los requisitos para la conversión precisa de los caracteres braille.

2.7.5 Pruebas y resultados del desarrollo del Sistema Electromecánico

Esta fase comprendió la construcción de la impresora braille siguiendo cada una de las especificaciones detalladas anteriormente, se debe en este punto realizar una integración de las herramientas de hardware y el software para armar cuidadosamente los componentes electromecánicos que garanticen el funcionamiento preciso y eficiente.

Prueba de motores paso a paso

Se necesitaron dos motores de paso a paso, los cuales cumplen la función de realizar los desplazamientos del carrete y de la hoja de manera coordinada y precisa. Para generar la matriz braille es necesario utilizar las medidas, las cuales se definieron al inicio del capítulo II, figura 8.

Para calcular la distancia que recorre el motor al dar un paso se utiliza la ecuación 3.

$$D = \frac{2 * \pi * r * n^{\circ}}{360^{\circ}} \quad (3)$$

Se define:

D: Distancia recorrida por el motor en un paso

π : Constante pi

r: Radio del piñón del motor

n° : Grados por paso

El desplazamiento del carrete es a través del motor en el eje X, tiene un ángulo de 1.8° grados por paso y el diámetro del piñón es de 12 mm (polea GT2). Aplicando la ecuación 2, la distancia que se desplazará el carrete es de 0.188 mm por paso.

El movimiento del rodillo para arrastrar la hoja se realiza por el motor en el eje Y, este motor según sus especificaciones técnicas realiza una vuelta de 360° con 200 pasos, la distancia que arrastra el papel se la toma de manera experimental es de 52.6 mm al dar una vuelta, el rodillo por lo tanto desplaza la hoja 0.263 mm. El número de pasos que se debe tener para las distancias para la matriz braille se detallan en la tabla 20.

Motor	Distancia	Pasos	Grados	Distancia Obtenida (experimental)
Eje X	1.0 mm	5 pasos	9°	1.025 mm
	2.5 mm	13 pasos	23.4°	2.523 mm
	6.00 mm	32 pasos	57.6°	6.004 mm
Eje Y	1.00 mm	4 pasos	7.2°	1.030 mm
	2.5 mm	10 pasos	18°	2.528 mm
	10.0 mm	38 pasos	68.4°	10.09 mm

Tabla 20 Pasos y grados obtenidos para distancias de la matriz braille.

Elaborado por el autor

La primera prueba para comprobar el desplazamiento de los motores en los ejes X, Y. Se utilizó una lámina de papel carbón que se ubica sobre la cartulina para marcar los puntos. El resultado se puede observar en la figura 83.

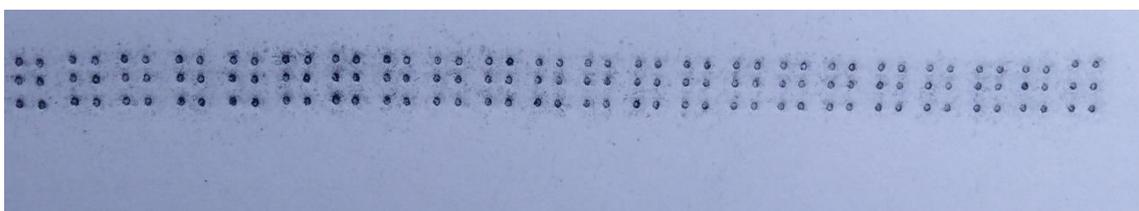


Figura 83 Prueba de desplazamientos ejes X, Y. Elaborado por el autor

Las punzadas realizadas por el solenoide se pueden visualizar claramente sobre la hoja, las medidas son uniformes, el desplazamiento de los motores está coordinados y se tiene un desplazamiento con gran porcentaje de precisión. Del mismo modo se hace la prueba para formar los puntos en relieve, el resultado se puede observar en la figura 84.



Figura 84 Prueba del punto de relieve. Elaborado por el autor

Placa controladora CNC Shield

Al realizar las pruebas con esta placa el funcionamiento de los drivers A4988 y los motores paso a paso se realizó de manera óptima. La fuente externa de 24 VDC alimenta la placa CNC que a su vez distribuye el voltaje para los drivers y los motores. Antes de montar los drivers es necesario calibrarlos, este proceso se realiza con el fin de reducir la corriente nominal que se suministra a los motores de paso. El driver posee un reóstato similar a un tornillo que se debe hacer girar para conocer ese valor referencial se debe hallar el valor de la tensión de referencia mediante la ecuación 4.

$$V_{ref} = I_{max} * 8 * R_S \quad (4)$$

Corriente por fase motor pasos:

$I_{max} = 1.2 \text{ A}$

Resistencia de montaje superficial A4988:

$R_S = 0.1 \Omega$

El valor obtenido mediante la ecuación es de 0.96 VDC, se debe girar el tornillo hasta encontrar ese valor. Para hacer la calibración se conecta el driver por los pines VDD y GND a los pines 5V y GND de una placa Arduino, luego se mide con un multímetro por

los terminales ubicado en el GND del driver y el terminal positivo del multímetro al tornillo del driver, de esta forma se debe calibra el driver, el proceso se repite en los dos controladores, una vez calibrados ya se pueden montar sobre la CNC.

Solenoides

Se utiliza el solenoide para formar los puntos de relieve braille. Este trabaja a 24 VDC con una fuerza de 3.5 N tal como se puede ver en la tabla 3. Para accionar el solenoide se envía en la programación los comandos G al Arduino para encender y apagar el solenoide, sin embargo, no fueron suficientes para formar los relieves tal como se puede ver en la figura 85. Para ello se realizó ajustes para aumentar las veces de encendido y apagado hasta encontrar la condición que permita formar mejor el relieve de los puntos.

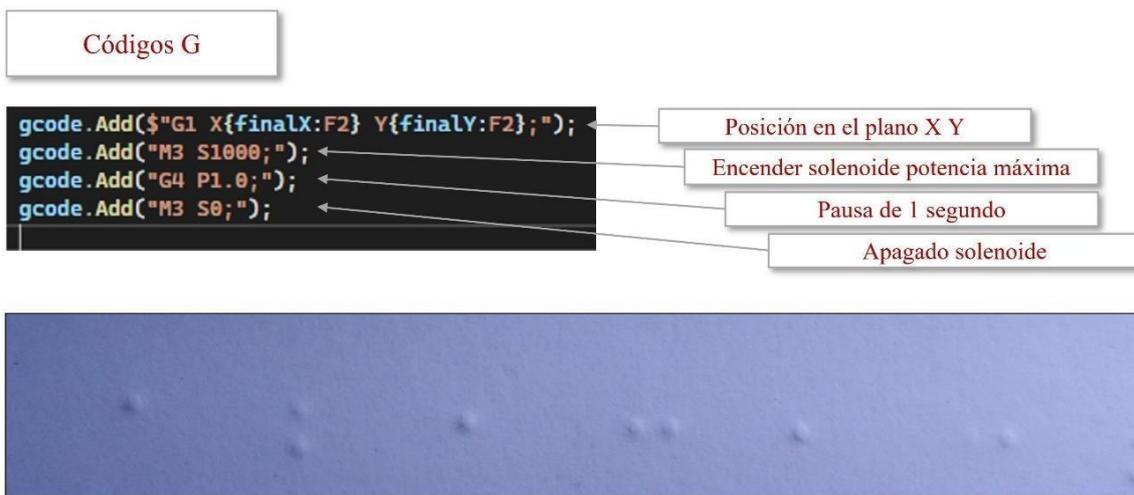


Figura 85 Comandos G(arriba), puntos en relieve (abajo). Elaborado por el autor

Si se añade un número muy alto de repeticiones para encendido y apagado, el solenoide rompe la hoja tal como se ve en la figura 86.



Figura 86 Puntos de relieve con máximas repeticiones. Elaborado por el autor

Realizando los ajustes a los códigos G en la parte de la programación se determinó que a través de cuatro repeticiones de encendido y apagado se obtiene los puntos braille con mejor relieve tal como se visualiza en la figura 87.

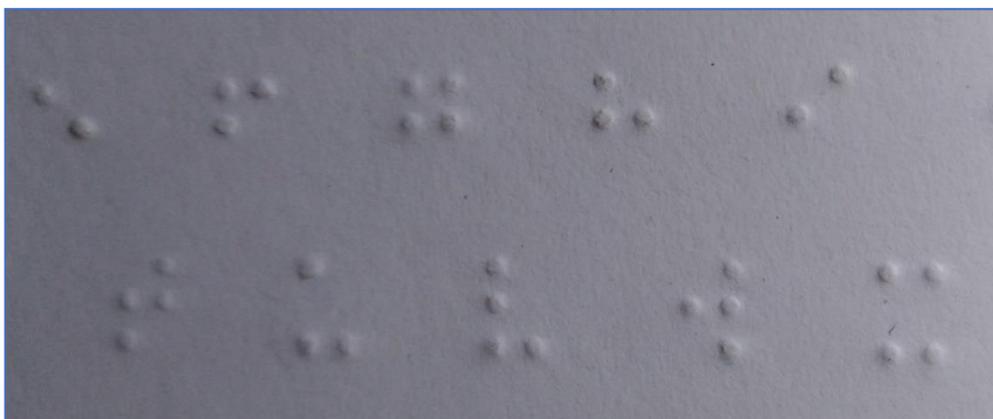


Figura 87 Puntos de relieve final. Elaborado por el autor

2.7.6 Pruebas y resultados de funcionamiento

Impresión del alfabeto

La primera prueba de impresión es el alfabeto tanto en minúsculas como mayúsculas, cada carácter se toma su referencia del signo generador de la matriz braille, el alfabeto se escribe en la interfaz de Braille Gcode Generator para enviar a imprimir (ver figura 88).

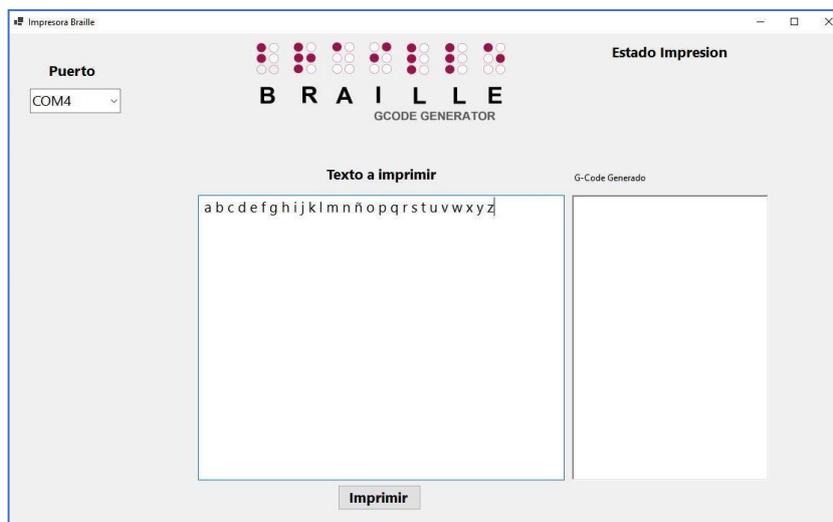


Figura 88 Alfabeto de minúsculas a imprimir. Elaborado por el autor

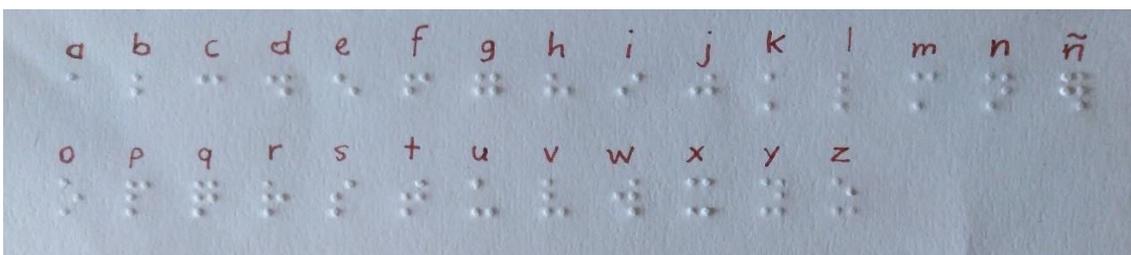


Figura 89 Alfabeto de minúsculas en código braille. Elaborado por el autor

Del mismo modo se escribe el alfabeto a imprimir en mayúsculas, el resultado obtenido se puede visualizar en la figura 91.

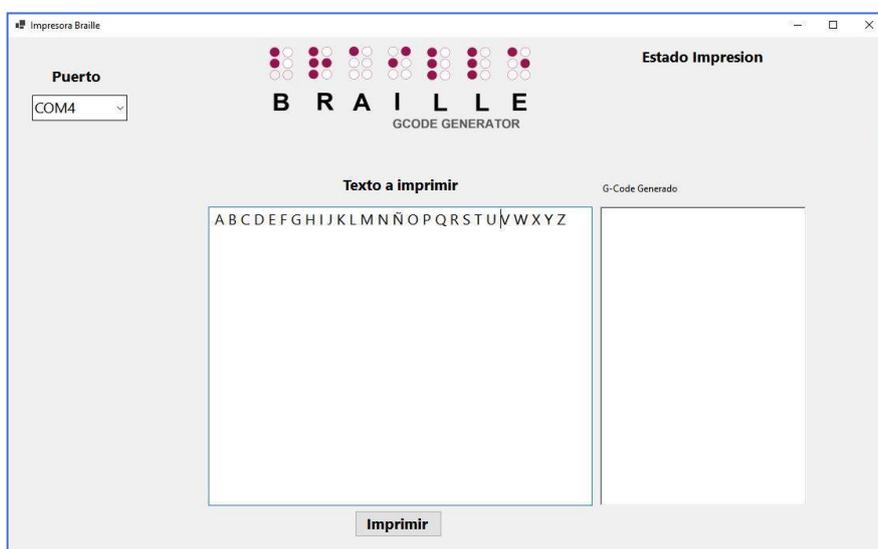


Figura 90 Alfabeto de mayúsculas a imprimir. Elaborado por el autor

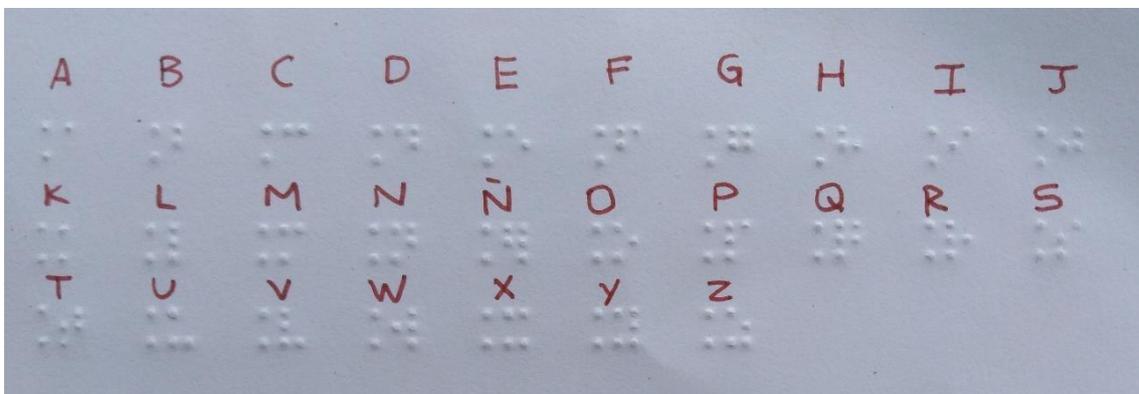


Figura 91 Alfabeto de mayúsculas en código braille. Elaborado por el autor

Impresión de números

Los números en Braille se componen de dos matrices, una con el identificador de número y seguido del número correspondiente se puede visualizar el resultado en la figura 93.

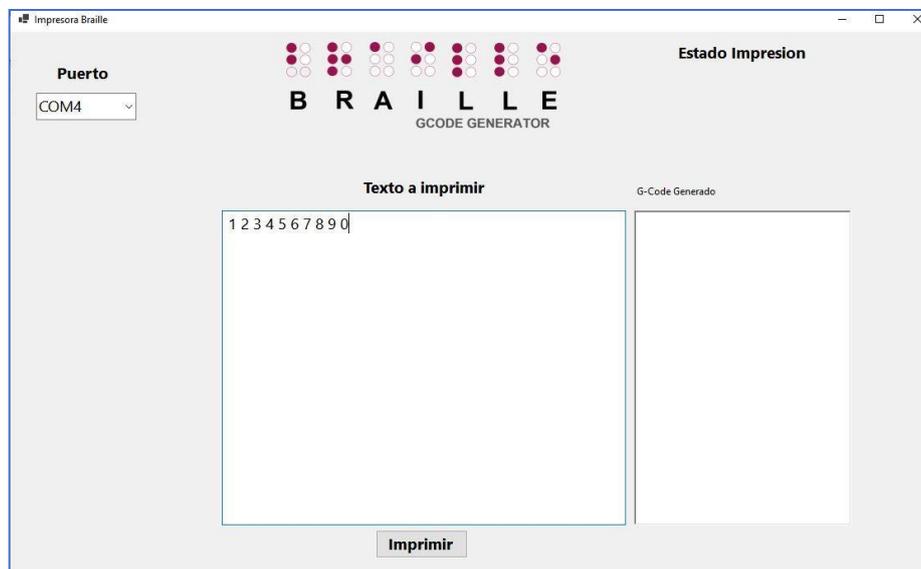


Figura 92 Números enviar a imprimir. Elaborado por el autor

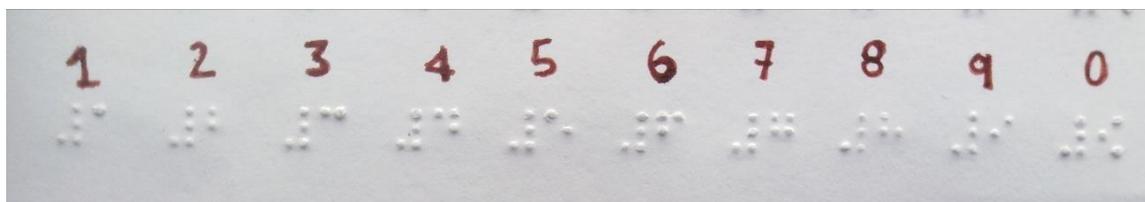


Figura 93 Números en código braille. Elaborado por el autor

Vocales tildadas

Las vocales tildadas tienen su propia simbología braille, se puede ver el resultado en la figura 95.

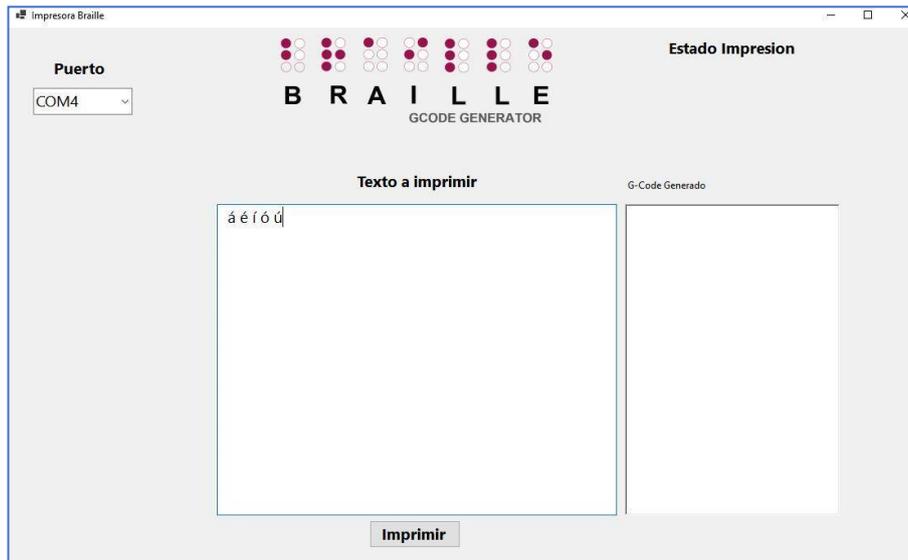


Figura 94 Vocales tildadas enviadas a imprimir. Elaborado por el autor



Figura 95 Vocales tildadas en código braille. Elaborado por el autor

Signos de puntuación y caracteres especiales

El resultado de la impresión se puede visualizar en la figura 97.

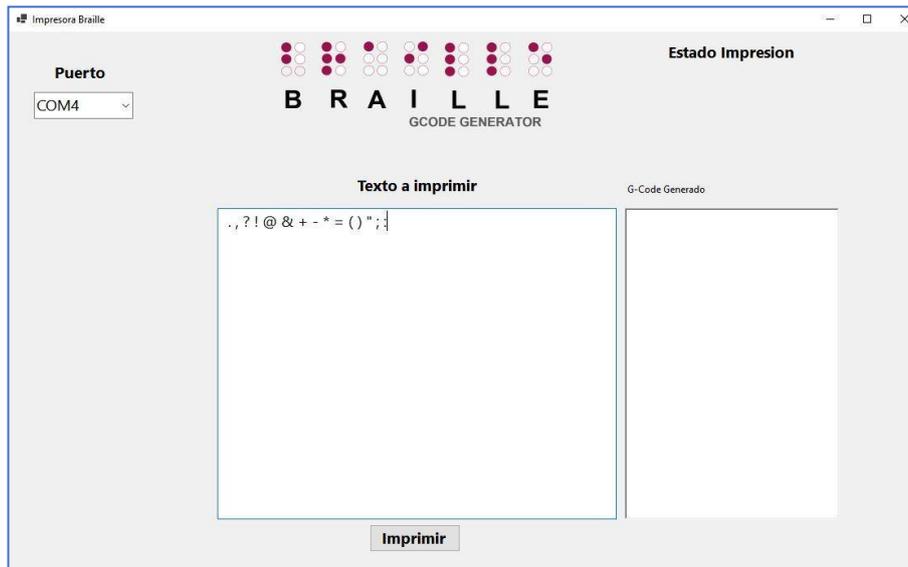


Figura 96 Envío de signos de puntuación y caracteres especiales. Elaborado por el autor

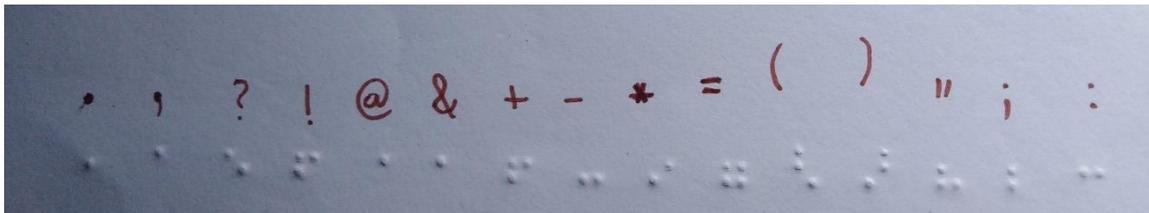


Figura 97 Signos de puntuación y caracteres especiales en código braille.

Elaborado por el autor

Precisión de medidas braille

Para la comprobación de las medidas braille es necesario comparar los valores obtenidos de las pruebas de impresión con el valor real recomendado en la figura 8.

Las medidas consideradas se muestran en la tabla a continuación.

Distancias	Medidas
Distancia horizontal entre puntos de la misma celda	2.5 mm
Distancia vertical entre puntos de la misma celda	2.5 mm
Distancia entre celdas (primera columna celda 1 y primera columna celda 2)	6.0 mm
Distancia entre el primer punto y el tercer punto de la celda	6.2 mm

Distancia vertical entre celdas (primer punto celda 1 fila 1 y el primer punto celda 2 fila 2).	10.0 mm
Altura de relieve de puntos	0.5 mm

Tabla 21 Medidas braille recomendadas. Elaborado por el autor

Con la tabla de medidas braille recomendados se realiza una comparación con cálculos de los errores absolutos y relativos. En la tabla 22 se muestran los resultados obtenidos para la Distancia horizontal entre puntos de la misma celda.

N° de Muestras	Parámetro recomendado	Distancia medida	Error Absoluto	Error Relativo
1	2.5 mm	2.3 mm	0.2	8%
2	2.5 mm	2.4 mm	0.1	4%
3	2.5 mm	2.5 mm	0	0%
4	2.5 mm	2.5 mm	0	0%
5	2.5 mm	2.6 mm	0.1	4%
6	2.5 mm	2.4 mm	0.1	4%
7	2.5 mm	2.5 mm	0	0%
8	2.5 mm	2.5 mm	0	0%
9	2.5 mm	2.4 mm	0.1	4%
10	2.5 mm	2.5 mm	0	0%

Tabla 22 Errores absolutos y relativos primera distancia. Elaborado por el autor

La Distancia vertical entre puntos de la misma celda, la medida recomendada es de 2.5 mm, las medidas tomadas de las pruebas de impresión se muestran en la tabla 23.

N° de Muestras	Parámetro recomendado	Distancia medida	Error Absoluto	Error Relativo
1	2.5 mm	2.5 mm	0	0%
2	2.5 mm	2.4 mm	0.1	4%
3	2.5 mm	2.5 mm	0	0%

4	2.5 mm	2.5 mm	0	0%
5	2.5 mm	2.4 mm	0.1	4%
6	2.5 mm	2.5 mm	0	0%
7	2.5 mm	2.5 mm	0	0%
8	2.5 mm	2.5 mm	0	0%
9	2.5 mm	2.5 mm	0.1	4%
10	2.5 mm	2.4mm	0.1	4%

Tabla 23 Errores absolutos y relativos segunda distancia. Elaborado por el autor

La Distancia medida entre celdas, desde el punto en la primera columna de la celda 1 y el punto en la primera columna de la celda 2, en la tabla 24 se muestran los errores absolutos y relativos.

N° de Muestras	Parámetro recomendado	Distancia medida	Error Absoluto	Error Relativo
1	6 mm	6.2 mm	0.2	3.3%
2	6 mm	6.3 mm	0.3	5%
3	6 mm	6.1 mm	0.1	1.7 %
4	6mm	6.1 mm	0.1	1.7%
5	6 mm	6 mm	0	0%
6	6 mm	6.1 mm	0.1	1.7%
7	6 mm	6.1 mm	0.1	1.7%
8	6mm	6 mm	0	0%
9	6 mm	6.2 mm	0.2	3.3%
10	6 mm	6.3 mm	0.3	5%

Tabla 24 Errores absolutos y relativos tercera distancia. Elaborado por el autor

La distancia de puntos en una misma celda entre el primer punto y el tercero, se calculan los errores absolutos y relativos de las pruebas de impresión (ver tabla 25).

N° de Muestras	Parámetro recomendado	Distancia medida	Error Absoluto	Error Relativo
-----------------------	------------------------------	-------------------------	-----------------------	-----------------------

1	6.2 mm	6.3 mm	0.1	1.6 %
2	6.2 mm	6.4 mm	0.2	3%
3	6.2 mm	6.2 mm	0	0 %
4	6.2 mm	6.5 mm	0.3	4.8%
5	6.2 mm	6.2 mm	0	0%
6	6.2 mm	6.2 mm	0	0%
7	6.2 mm	6.3 mm	0.1	1.6%
8	6.2 mm	6.2 mm	0	0%
9	6.2 mm	6.3 mm	0.1	1.6%
10	6.2 mm	6.4 mm	0.2	3%

Tabla 25 Errores absolutos y relativos cuarta distancia. Elaborado por el autor

Los errores absolutos y relativos de la distancia vertical entre celdas (primer punto celda 1 fila 1 y el primer punto celda 2 fila 2), el parámetro recomendado es de 10 mm.

N° de Muestras	Parámetro recomendado	Distancia medida	Error Absoluto	Error Relativo
1	10 mm	10.1 mm	0.1	1 %
2	10 mm	10.1 mm	0.1	1%
3	10 mm	10.2 mm	0.2	2 %
4	10 mm	10 mm	0	0%
5	10 mm	10 mm	0	0%
6	10 mm	10.2 mm	0.2	2%
7	10 mm	10.1 mm	0.1	1%
8	10 mm	10.1 mm	0.1	1%
9	10 mm	10 mm	0	0%
10	10 mm	10 mm	0	0%

Tabla 26 Errores absolutos y relativos quinta distancia. Elaborado por el autor

La medida de altura de relieve de puntos es un valor muy importante para el código braille ya que se basa en los relieves sobre el papel, el parámetro recomendado es de 0.5 mm.

N° de Muestras	Parámetro recomendado	Distancia medida	Error Absoluto	Error Relativo
1	0.5 mm	0.5 mm	0	0%
2	0.5 mm	0.5 mm	0	0%
3	0.5 mm	0.5 mm	0	0%
4	0.5 mm	0.5mm	0	0%
5	0.5 mm	0.5mm	0	0%
6	0.5 mm	0.5 mm	0	0%
7	0.5 mm	0.5 mm	0	0%
8	0.5 mm	0.5 mm	0	0%
9	0.5 mm	0.5 mm	0	0%
10	0.5 mm	0.5 mm	0	0%

Tabla 27 Errores absolutos y relativos altura de relieve. Elaborado por el autor

Finalmente, con los cálculos de los errores absolutos y relativos de cada una de las distancias obtenidas se realiza un análisis de los valores con mayor error. En la tabla 28 se muestran los valores más altos obtenidos.

Distancias	Error Absoluto	Error Relativo
Distancia horizontal entre puntos de la misma celda	0.2 mm	8%
Distancia vertical entre puntos de la misma celda	0.1 mm	4%
Distancia entre celdas (primera columna celda 1 y primera columna celda 2)	0.3 mm	5%
Distancia entre el primer punto y el tercer punto de la celda	0.3 mm	4.8%
Distancia vertical entre celdas (primer punto celda 1 fila 1 y el primer punto celda 2 fila 2).	0.2 mm	2%
Altura de relieve de puntos	0 mm	0%

Tabla 28 Medidas braille recomendadas. Elaborado por el autor

Mediante la tabla 28, se observa los valores máximos de errores absolutos y errores relativos. El máximo error absoluto no supera los 0.3 mm y el valor del error relativo más alto es de 8% que se obtuvo de la medida de la distancia horizontal entre los puntos de la misma celda lo que se deduce que las medidas de impresión tienen un 0.3 mm de incertidumbre en todas las medidas y pueden estar alejadas de las medidas originales hasta un 8%, que se encuentran dentro de los parámetros permitidos y demuestra la eficacia en la impresión con las medidas braille establecidas.

Tiempo de impresión por hoja

Para el tiempo de impresión por hoja es necesario utilizar una hoja de cartulina de las dimensiones 210 mm por 297 mm, el número de caracteres es de 29 por fila, lo que da un total de 696 caracteres incluidos los espacios en blanco que corresponde a una matriz de cero puntos. Para hallar un valor de tiempo máximo se envió a imprimir un símbolo que represente una matriz completa, marcando todos los puntos del signo generador.

Impresión	Tiempo (h:m:s)
Carácter	0:00:36
Fila de 29 caracteres	0:04:40
Impresión por página 24 filas	1: 10:58

Tabla 29 Tiempo de impresión. Elaborado por el autor

El cálculo máximo del tiempo por hoja es de una hora con diez minutos y cincuenta y ocho segundos, el tiempo es bastante alto por que se tiene un solo punzón (actuador) que ejecuta los puntos sobre la cartulina, en comparación con las impresoras del mercado el tiempo de impresión de una hoja completa es lento, ya que las impresoras pueden imprimir inclusive a 140 caracteres por segundo.

2.7.7 Pruebas y resultados de evaluación por personas no videntes

Las pruebas realizadas incluyen la evaluación de la legibilidad de los caracteres impresos al tacto, así también opiniones sobre la utilidad del material impreso.



Figura 98 Lectura evaluación de los caracteres braille Elaborado por el autor

Para realizar las pruebas se solicitó la participación del señor Enrique Conforme Suarez, una persona no vidente con conocimientos en código braille. Se envió a imprimir los caracteres en braille y una vez finalizados se entregó al no vidente para su lectura.

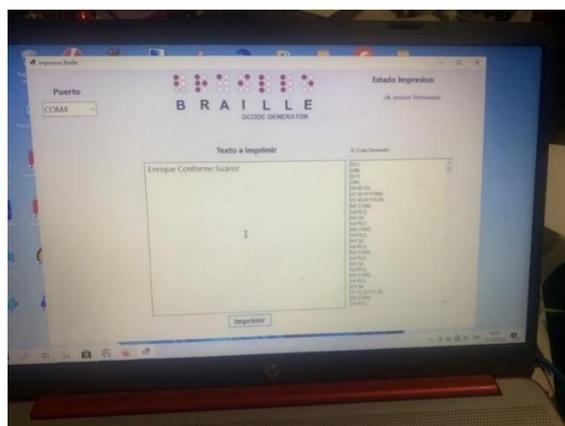


Figura 99 Envío de texto a imprimir. Elaborado por el autor

La profesora Jackeline Quimí, también formó parte del proceso de evaluación de los caracteres braille, el no vidente identificó carácter a carácter el texto enviado por la impresora braille. Se realizaron pruebas de impresión para evaluar el nivel de satisfacción.

Impresión de mayúsculas: Se realizó un total de diez pruebas con los caracteres braille del alfabeto en mayúsculas en la tabla se obtiene el nivel de satisfacción.

Nivel	Número de pruebas	Porcentaje
Muy satisfecho	9	90%
Moderadamente satisfecho	1	10%
Poco Satisfecho	0	0%
No satisfecho	0	0%
	Total	100%

Tabla 30 Nivel de satisfacción impresión de alfabeto mayúsculas

Impresión de minúsculas: Se realizó un total de diez pruebas con los caracteres braille del alfabeto en minúsculas en la tabla se obtiene el nivel de satisfacción.

Nivel	Número de pruebas	Porcentaje
Muy satisfecho	9	90%
Moderadamente satisfecho	1	10%
Poco Satisfecho	0	0%
No satisfecho	0	0%
	Total	100%

Tabla 31 Nivel de satisfacción impresión de alfabeto minúsculas

Impresión de vocales tildadas: Se realizó un total de diez pruebas con los caracteres braille de las vocales tildadas en la tabla se obtiene el nivel de satisfacción.

Nivel	Numero de pruebas	Porcentaje
Muy satisfecho	9	90%
Moderadamente satisfecho	1	10%
Poco Satisfecho	0	0%
No satisfecho	0	0%
	Total	100%

Tabla 32 Nivel de satisfacción impresión de vocales tildadas

Impresión de números: Se realizó un total de diez pruebas con los caracteres braille de los números en la tabla se obtiene el nivel de satisfacción.

Nivel	Numero de pruebas	Porcentaje
Muy satisfecho	9	90%
Moderadamente satisfecho	1	10%
Poco Satisfecho	0	0%
No satisfecho	0	0%
	Total	100%

Tabla 33 Nivel de satisfacción impresión de números

Impresión de signos de puntuación: Se realizó un total de diez pruebas de signos de puntuación en la tabla se obtiene el nivel de satisfacción.

Nivel	Numero de pruebas	Porcentaje
Muy satisfecho	9	90%
Moderadamente satisfecho	1	10%
Poco Satisfecho	0	0%
No satisfecho	0	0%
	Total	100%

Tabla 34 Nivel de satisfacción impresión de signos de puntuación

Los resultados obtenidos de estas pruebas fueron una buena legibilidad de los códigos sobre la cartulina A4 que se puede interpretar fácilmente por la persona no vidente, la conversión del texto es exacto y preciso, el software cumple el fin de convertir el texto en códigos y el sistema electromecánico funciona correctamente. El nivel obtenido es “muy satisfecho” con un 90%.

CONCLUSIONES

El análisis de los códigos braille sirvieron para identificar y documentar las correspondencias precisas de las letras, números y signos de puntuación en el código braille, las pruebas realizadas aportan para entender la base del sistema y garantizar en gran medida la representación correcta. La comprensión de reglas ortográficas y gramaticales también fueron de gran aporte para obtener la coherencia y precisión en la impresión de los códigos braille.

Los dispositivos electrónicos existentes ofrecen un gran aporte para la impresión de textos braille para los procesos educativos de personas no videntes, sin embargo, el costo representa una limitación enorme para tener el acceso a estos dispositivos, el equipo más económico representa un costo de 2330.15 dólares americanos, un valor que no es costeable para cualquier persona no vidente. El costo de la impresora braille implementada mediante este proyecto es de 487.03 dólares americanos, que representa una reducción muy significativa del 79.1% del costo de una impresora en el mercado.

Para el diseño del hardware se utiliza la tecnología de máquina CNC pues permitió tener mayor precisión milimétrica que se necesita para obtener los desplazamientos del carrete y del rodillo. Así como también sus componentes CNC son de fácil acceso ya que se consiguen a precios accesibles en tiendas electrónicas el costo de los materiales CNC para la impresora braille es de 81.91 dólares americanos. Del mismo modo el software utilizado para la construcción se basa en comandos G, que coordina de manera exacta la conversión de letras, los números y signos de puntuación. Este software es fundamental para vincular al no vidente con la máquina mediante su interfaz Braille Gcode Generator.

Mediante los motores paso a paso, los drivers y el solenoide se construyó el sistema electromecánico, que permite desplazar un rodillo en el eje Y para el movimiento de la hoja y el carrete en el eje X para el actuador solenoide, los drivers A4988 controlan los motores y la placa CNC Shield proporciona el control de los comandos del firmware grbl para comandos G. La construcción de este sistema permite generar la impresión de una hoja con los símbolos braille que sirve como material para ayuda a las personas no videntes.

La impresora braille fue probada para la impresión de letras, números y signos de puntuación, obteniendo la eficacia de la impresión con un margen de error en las medidas

de ± 0.3 mm y hasta un 8% de imprecisión en las dimensiones recomendadas para la impresión de los caracteres braille. Así también, se pudo identificar el tiempo de impresión de una hoja completa, el cual es muy alto a comparación de las impresoras braille comerciales.

Finalmente, la impresora fue evaluada por la maestra Jackeline Quimí especializada en sistema braille de la Fundación Melvin Jones de la Libertad, así también, el señor Enrique Conforme Suarez, persona no vidente que revisaron los textos impresos en código braille y se evaluó el nivel de satisfacción mediante las pruebas de impresión y se obtiene un 90% “muy satisfecho”. Así también mostraron su satisfacción por la implementación de este tipo de proyecto como herramienta de ayuda a los procesos de aprendizaje y enseñanza de las personas no videntes.

RECOMENDACIONES

- Verificar exhaustivamente los códigos braille mediante documentación acertada y comprobada, para no causar confusión en la conversión de los caracteres braille.
- Utilizar los equipos de hardware que mejor precisión y disponibilidad ofrezcan para el proceso de construcción, ya que de esta manera se disminuye tiempo de adaptar los componentes electromecánicos.
- Se recomienda el uso del software adaptado a lenguaje de programación con códigos G, ya que facilita la conversión de texto para enviar mediante ordenes de coordenadas para desplazamiento de motores y actuadores.
- Realizar un mantenimiento preventivo cada seis meses, realizando una limpieza de los diferentes sistemas de la impresora braille, es primordial mantener engrasado los ejes y los rodamientos para asegurar un óptimo funcionamiento del sistema electromecánico.
- Se recomienda para trabajos futuros, crear una interfaz que permita el ingreso de archivos .doc o del tipo .pdf. para mejorar la accesibilidad de textos para enviar a imprimir.
- Para disminuir el tiempo de impresión se recomendaría implementar un sistema de seis solenoides para punzar sobre el papel, esto permitiría crear la matriz braille con mayor rapidez, logrando textos braille en menor tiempo.

BIBLIOGRAFÍA

- [1] ElTelegrafo.com, «5000 alumnos con discapacidad visual estan en planteles regulares,» 12 febrero 2019. [En línea]. Available: <https://www.eltelegrafo.com.ec/noticias/sociedad/6/alumnos-discapacidad-visual-planteles-regulares>. [Último acceso: 10 mayo 2023].
- [2] J. M. Sadurní, «INVENTOS UNIVERSALES,» 04 enero 2022. [En línea]. Available: https://historia.nationalgeographic.com.es/a/louis-braille-inventor-lectura-tactil_17545. [Último acceso: 10 mayo 2023].
- [3] P. E. Sanchez Jaime, «Repositorio Institucional de la Universidad Politécnica Salesiana,» junio 2011. [En línea]. Available: <https://dspace.ups.edu.ec/handle/123456789/1422>. [Último acceso: 10 mayo 2023].
- [4] Consejo Nacional de Fomento Educativo, «Discapacidad visual,» 2010. [En línea]. Available: <https://www.gob.mx/cms/uploads/attachment/file/106810/discapacidad-visual.pdf>. [Último acceso: 07 mayo 2023].
- [5] A. Barragan, «Repositorio.puce.edu,» 16 diciembre 2013. [En línea]. Available: <http://repositorio.puce.edu.ec/bitstream/handle/22000/5629/T-PUCE-5785%281%29.pdf?sequence=1&isAllowed=y>. [Último acceso: 07 mayo 2023].
- [6] GRUPO SOCIAL ONCE, «Signografía básica de las lenguas cooficiales españolas,» 3 junio 2023. [En línea]. Available: <https://www.once.es/servicios-sociales/braille/comision-braille-espanola/documentos-tecnicos/documentos-tecnicos-relacionados-con-el-braille/documentos/b2-signografia-basica-lenguas-cooficiales>. [Último acceso: 18 diciembre 2023].
- [7] D. Ogalde, «Conoce mas sobre: Maquina de escribir Perkins Brailier,» 6 febrero 2023. [En línea]. Available: <https://opencluster-tech.cl/2023/02/06/conoce-mas-sobre-maquina-de-escribir-perkins-brailier/>. [Último acceso: 2 noviembre 2023].
- [8] Guide Dogs, «Braille Printer,» [En línea]. Available: <https://www.guidedogs.org.uk/getting-support/information-and-advice/how-can-technology-help-me/braille-printer/>. [Último acceso: 4 noviembre 2023].
- [9] RNIB, «ViewPlus Tiger Embraile embosser,» [En línea]. Available: <https://shop.rnib.org.uk/viewplus-tiger-embraille-embosser-2908>. [Último acceso: 4 noviembre 2023].
- [10] Sight and sound, «Impresora en relieve Index Basic-DV5,» [En línea]. Available: <https://www.sightandsound.co.uk/product/index-basic-d-embosser/>. [Último acceso: 4 noviembre 2023].
- [11] INDEX BRAILLE, «Everest-D V5,» [En línea]. Available: <https://www.indexbraille.com/en-us/braille-embossers/everest-d-v5?c=EUR>. [Último acceso: 4 noviembre 2023].

- [12] K. Yasar, «computer numerical control (CNC),» marzo 2023. [En línea]. Available: <https://www.techtarget.com/searcherp/definition/computer-numerical-control-CNC>. [Último acceso: 30 noviembre 2023].
- [13] AUTODESK, «CNC programming software witch G-code,» [En línea]. Available: <https://www.autodesk.com/solutions/cnc-programming#:~:text=CNC%20programming%20refers%20to%20the,cutting%2C%20milling%2C%20and%20drilling..> [Último acceso: 30 noviembre 2023].
- [14] RAPIDDIRECT, «G and M Codes:,» 21 diciembre 2023. [En línea]. Available: <https://www.rapiddirect.com/blog/g-and-m-codes/>. [Último acceso: 10 diciembre 2023].
- [15] J. mentira, «Los 6 mejores software de programación CNC,» [En línea]. Available: <https://www.runsom.com/es/technology/top-6-cnc-software-for-different-skill-levels/>. [Último acceso: 5 diciembre 2023].
- [16] CNC ROUTER SOURCE, «El controlador CNC,» [En línea]. Available: <http://www.cncroutersource.com/cnc-controller.html>. [Último acceso: 30 noviembre 2023].
- [17] BeJob, «QUE ES LA PROGRAMACION CON ARDUINO Y PARA QUE SIRVE,» 14 febrero 2017. [En línea]. Available: <https://www.bejob.com/que-es-la-programacion-con-arduino-y-para-que-sirve/#:~:text=Lenguaje%20de%20la%20programaci%C3%B3n%20con%20Arduino%3A%20C%2B%2B&text=La%20plataforma%20Arduino%20se%20programa,es%20similar%20a%20C%2B%2B..> [Último acceso: 07 mayo 2023].
- [18] Ediciones-eni.com, «Ediciones-eni,» s/f. [En línea]. Available: <https://www.ediciones-eni.com/open/mediabook.aspx?idR=26f910ec5d01ed09654484ca03c4f456>. [Último acceso: 10 mayo 2023].
- [19] Makerguides, «Comunicación SPI maestro-esclavo y ejemplo de lectura SPI de Arduino,» s/f. [En línea]. Available: <https://www.makerguides.com/es/master-slave-spi-communication-arduino/>. [Último acceso: 10 mayo 2023].
- [20] System Controls, «SISTEMAS ELECTROMECHANICOS,» s/f. [En línea]. Available: <http://system-controls.com/electro-systems/electro-mechanical-systems/>. [Último acceso: 15 mayo 2023].
- [21] Orientalmotor, «Basics of Stepper Motors,» 2022. [En línea]. Available: <https://www.orientalmotor.com/stepper-motors/technology/stepper-motor-basics.html>. [Último acceso: 20 octubre 2023].
- [22] TME Electronic Components, «MOTOR PASO A PASO,» 08 septiembre 2020. [En línea]. Available: <https://www.tme.eu/es/news/library-articles/page/41861/Motor-paso-a-paso-tipos-y-ejemplos-del-uso-de-motores-paso-a-paso/>. [Último acceso: 23 octubre 2023].
- [23] Isaac, «DRV8825: el drvier para motores paso a paso,» 4 agosto 2021. [En línea]. Available: <https://www.hwlibre.com/drv8825/>. [Último acceso: 24 octubre 2023].

- [24] HARTECH, «Dispositivos electromecánicos: descripción general,» s/f. [En línea]. Available: <https://www.har-tech.com/en/electromechanical-devices/>. [Último acceso: 15 mayo 2023].
- [25] G. V. DAMASO, J. P. G. RAMIRÉZ, O. A. ORBE, D. M. MORENO, M. I. H. HERNANDEZ, A. G. ROMERO y J. C. M. LEVARO, «CONTROLES ELECTRICOS,» [En línea]. Available: <https://saint-hyoga.wixsite.com/controlselectricos/en/blanco-c15o5>. [Último acceso: 24 octubre 2023].
- [26] Industrysurfer, «¿Que son los solenoides y los campos magnéticos de solenoides?,» [En línea]. Available: <https://industrysurfer.com/blog-industrial/ingenieria/ingenieria-electrica-ingenieria/hogar/que-son-los-solenoides-y-los-campos-magneticos-de-solenoides/>. [Último acceso: 24 octubre 2023].
- [27] AUTODESK, «IMPRESION 3D,» [En línea]. Available: <https://latinoamerica.autodesk.com/solutions/3d-printing>. [Último acceso: 23 noviembre 2023].
- [28] SICNOVA, «Impresora 3DUltimaker S5,» [En línea]. Available: <https://tienda.sicnova3d.com/impresoras3d-ultimaker/9170-impresora-3d-ultimaker-s5>. [Último acceso: 24 noviembre 2023].
- [29] J. A. Saavedra, «Qué es Github y para que sirve,» 4 junio 2023. [En línea]. Available: <https://ebac.mx/blog/que-es-github>. [Último acceso: 5 diciembre 2023].
- [30] J. A. Castillo, «Puerto serie - Qué es, para qué sirve y tipos,» 7 marzo 2020. [En línea]. Available: <https://www.profesionalreview.com/2020/03/07/puerto-serie-que-es-para-que-sirve-y-tipos/>. [Último acceso: 30 noviembre 2023].
- [31] BINARY UPDATES, «Comunicación serie en Arduino Uno,» [En línea]. Available: <https://binaryupdates.com/serial-communication-in-arduino-uno/>. [Último acceso: 30 noviembre 2023].
- [32] A. C. Espejo Cobo, «El lenguaje de los sentidos,» 20 enero 2022. [En línea]. Available: <https://www.archiletras.com/actualidad/el-lenguaje-de-los-sentidos/#:~:text=A%20trav%20C3%A9s%20de%20la%20vista,distinguir%20y%20percibir%20los%20olores..> [Último acceso: 25 octubre 2023].
- [33] C. P. ESCOBAR VINUEZA, «ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO,» febrero 2017. [En línea]. Available: <http://dspace.esPOCH.edu.ec/handle/123456789/6521>. [Último acceso: 25 octubre 2023].
- [34] C. G. A. BRITO y J. M. C. LLECLLISH, «RepositorioAcademicoUSMP,» 2015. [En línea]. Available: <https://hdl.handle.net/20.500.12727/2006>. [Último acceso: 25 octubre 2023].
- [35] M. A. M. Gutiérrez, «Universidad del Azuay,» 2015. [En línea]. Available: <http://dspace.uazuay.edu.ec/handle/datos/4857>. [Último acceso: 25 octubre 2023].
- [36] altronics, «Shield CNC V3 para arduino,» [En línea]. Available: <https://altronics.cl/shield-cnc-v3-arduino>. [Último acceso: 29 noviembre 2023].

ANEXOS

Anexo 1 Construcción y pruebas de impresora braille



Figura 100 Construcción de estructura de impresora braille

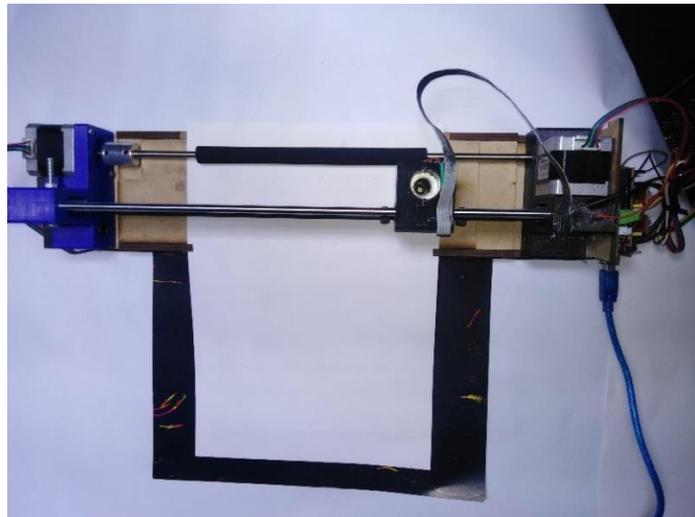
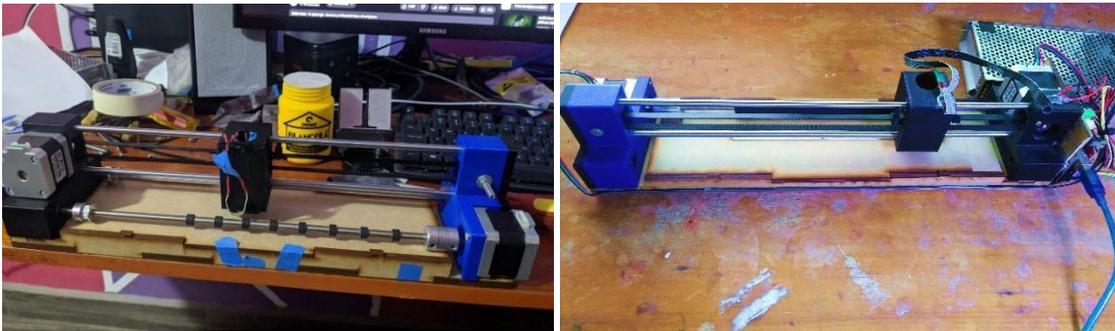


Figura 101 Ensamble final de componentes



Figura 102 Pruebas de funcionamiento

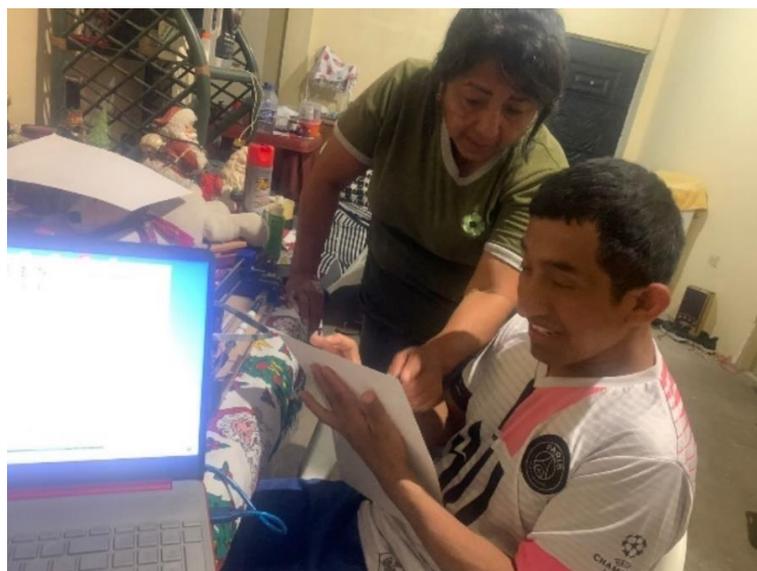
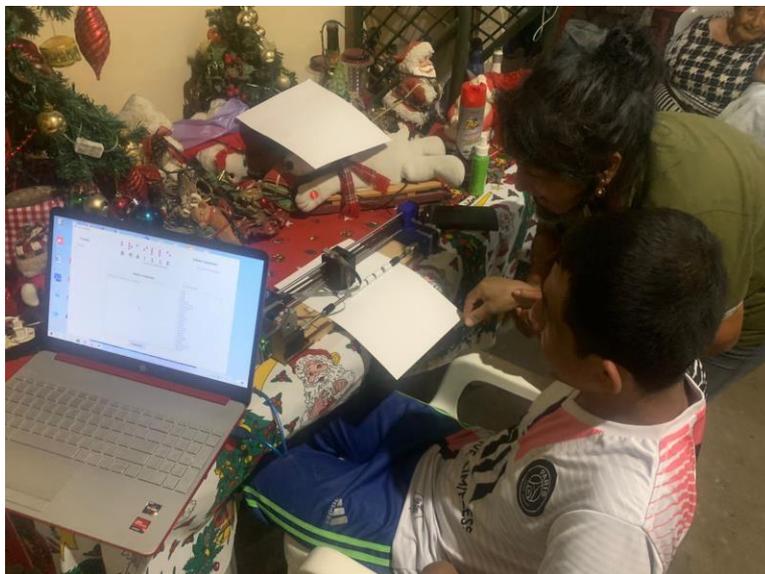
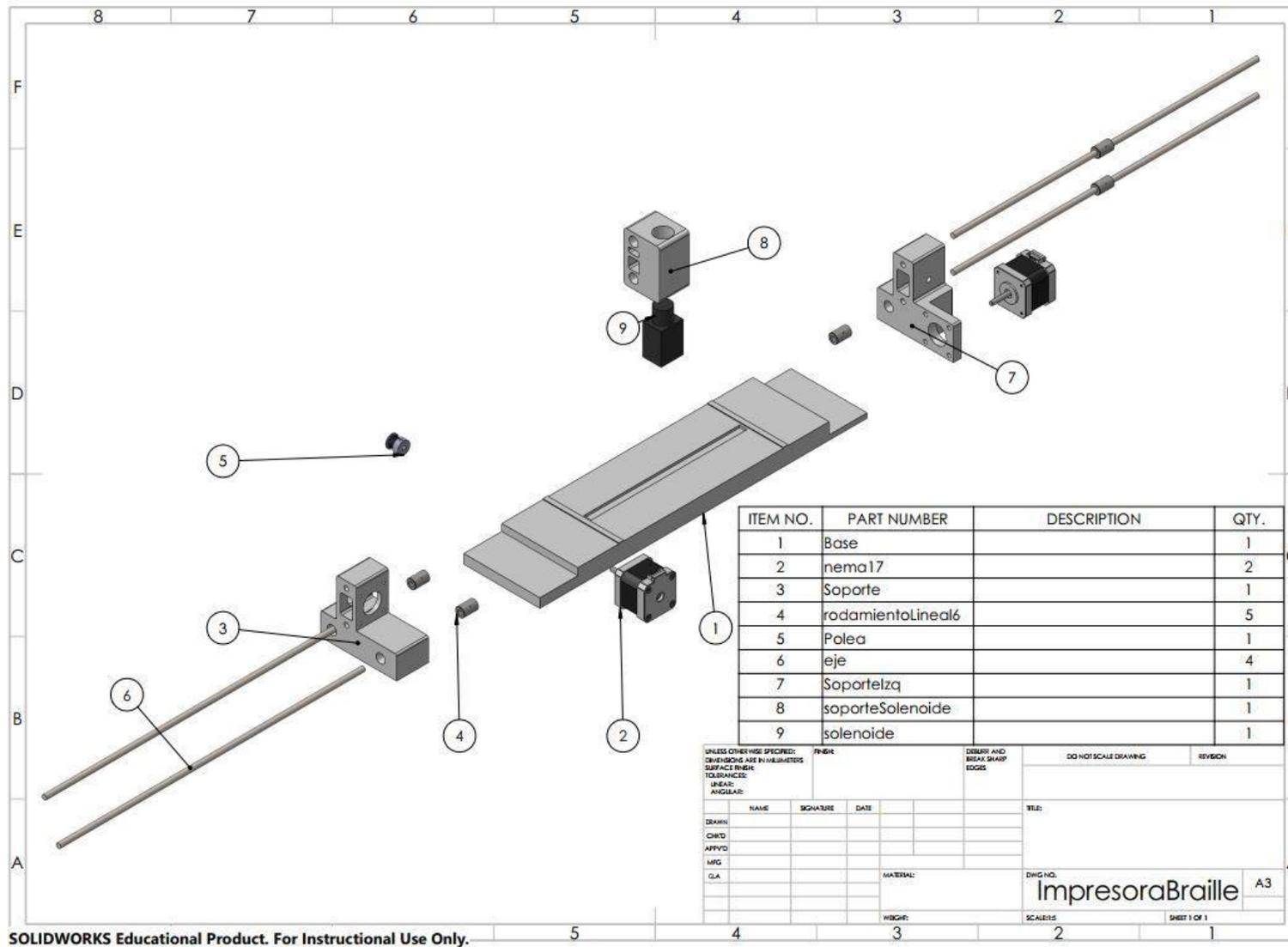
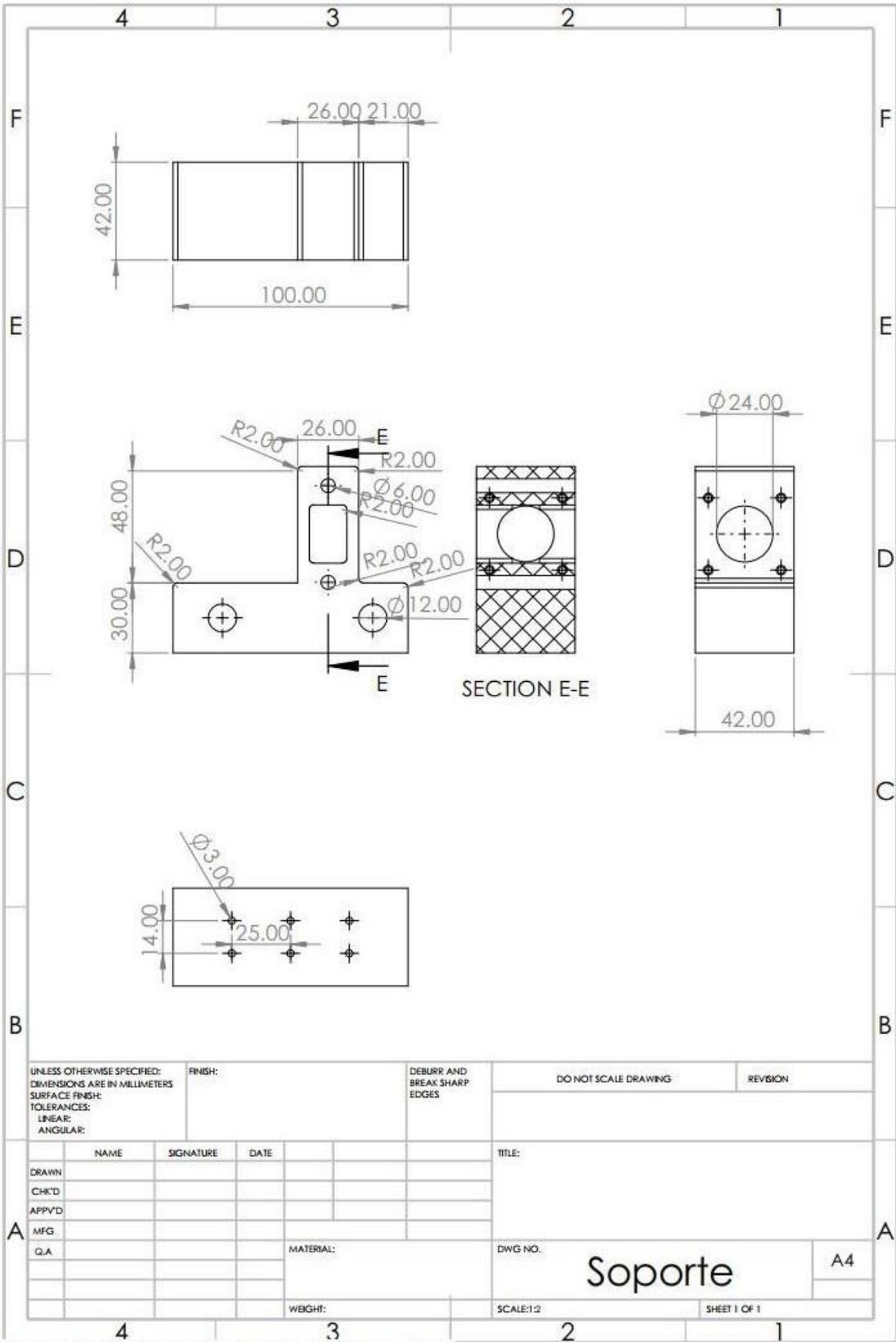


Figura 103 Pruebas de validación con persona no vidente

Anexo 2 Planos de construcción





UNLESS OTHERWISE SPECIFIED:
 DIMENSIONS ARE IN MILLIMETERS
 SURFACE FINISH:
 TOLERANCES:
 LINEAR:
 ANGULAR:

FINISH:

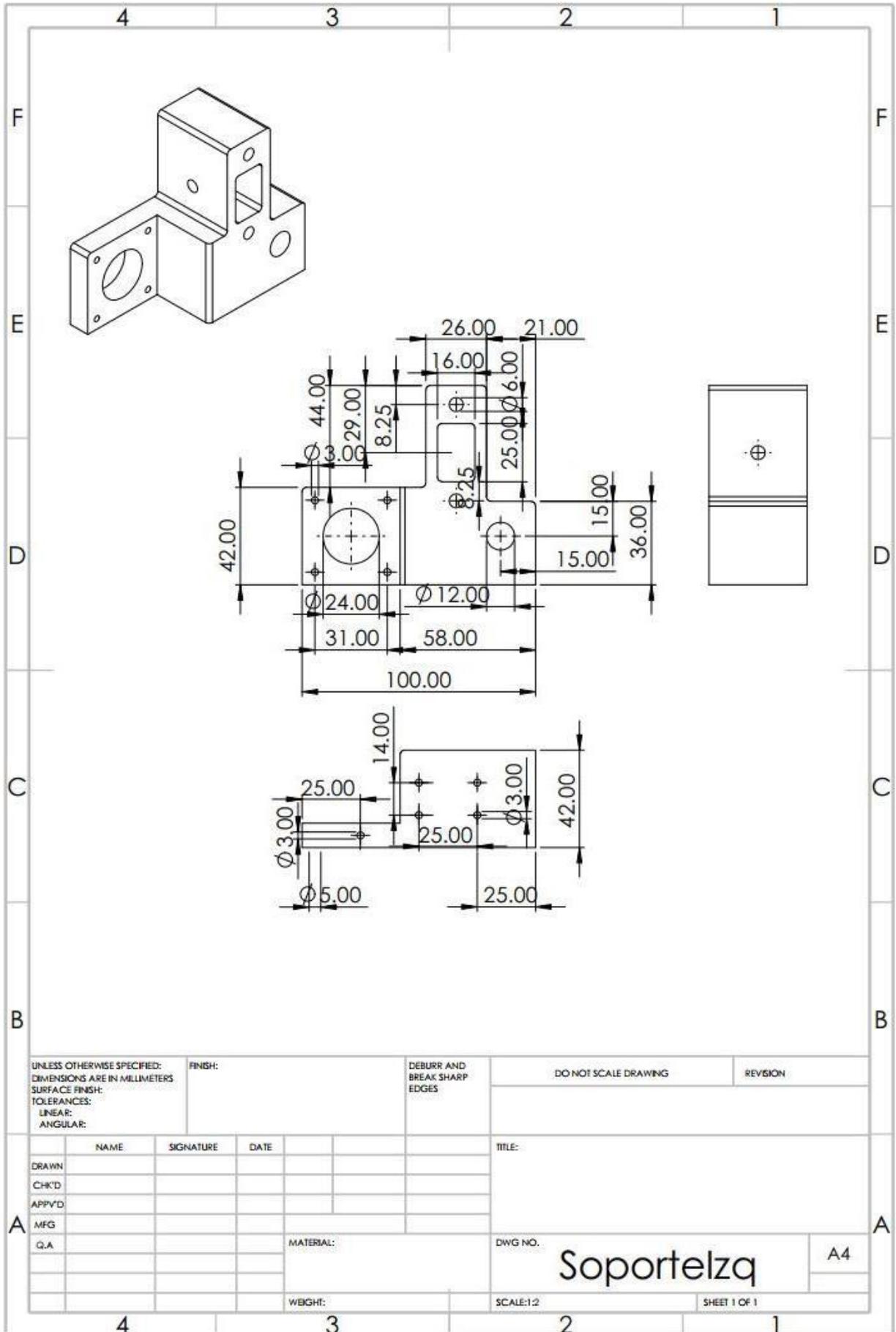
DEBURR AND
 BREAK SHARP
 EDGES

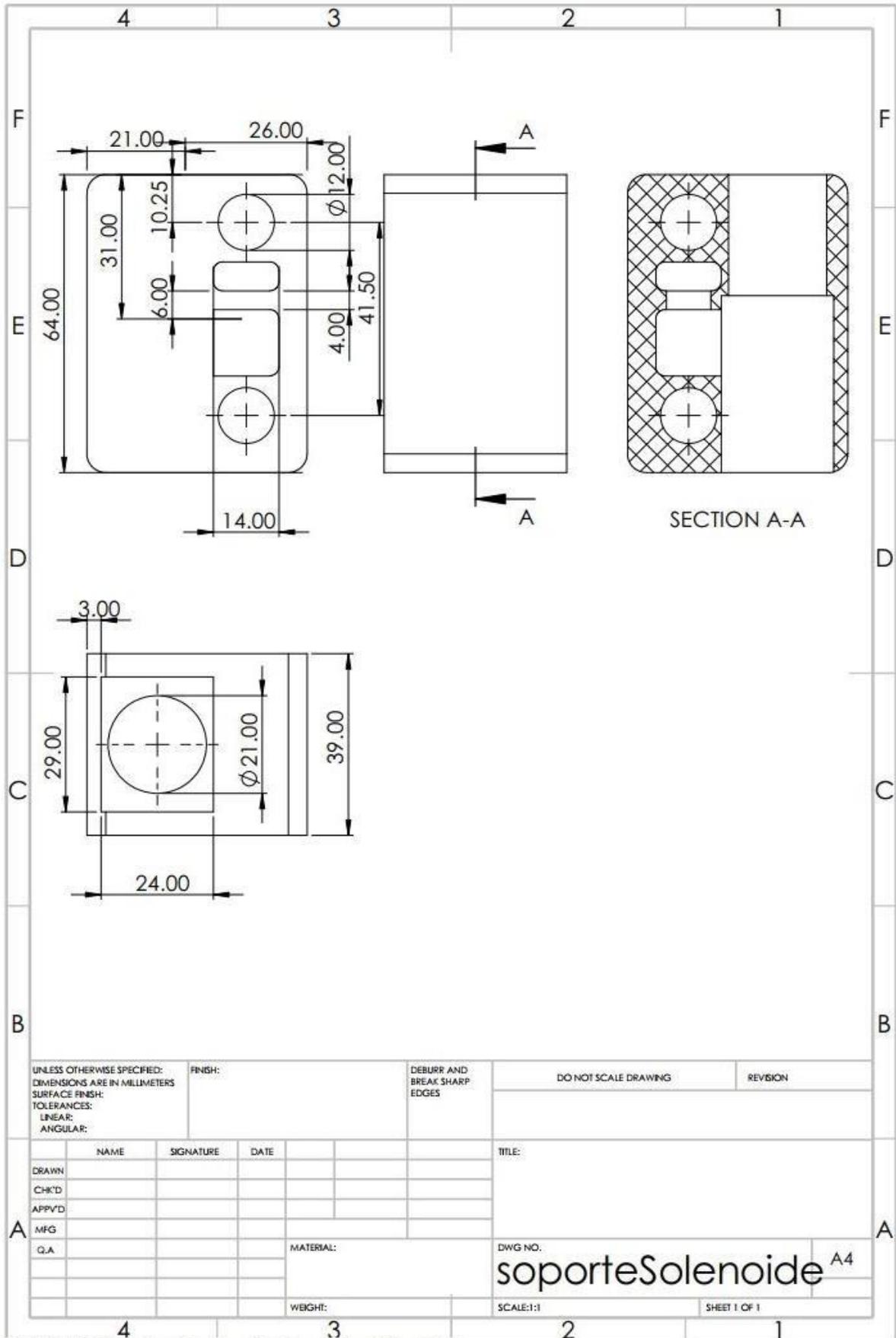
DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APP'VD			
MFG			
Q.A			

TITLE:	<h1>Soporte</h1>
DWG NO.	
SCALE: 1:2	SHEET 1 OF 1





SOLIDWORKS Educational Product. For Instructional Use Only.

Anexo 3 Códigos Funcionamiento Braille Gcode Generator Visual Studio

GcodeGenerator.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ImpresoraBraille;

public static class GCodeGenerator
{
    public static string GeneratorGCode(string text)
    {
        List<string> gcode = new();

        double xOffset = 0.0;
        double yOffset = 10.0;
        double xIncrementWord = 6.0;
        double yIncrementRow = 5.0;
        int limitWordRow = 29;
        double increment = 2.5;

        gcode.Add("G21;");
        gcode.Add("G90;");
        gcode.Add("G17;");
        gcode.Add("G94;");
        gcode.Add("G0 X0 Y0;");
        gcode.Add("G1 X0 Y0 F1000;");

        foreach (char character in text)
        {
            if (BrailleDictionary.BrailleChars.TryGetValue(character, value: out var brailleMatrix))
            {
                for (int index = 0; index < brailleMatrix.Length; index++)
                {
                    if (brailleMatrix[index] == 1)
                    {
                        double finalX, finalY;

                        switch (index)
                        {
                            case 0:
                                finalX = xOffset;
                                finalY = yOffset;
                                break;

                            case 1:
                                finalX = xOffset + increment;
                                finalY = yOffset;
                                break;

                            case 2:
                                finalX = xOffset;
                                finalY = yOffset + increment;
                                break;

                            case 3:
                                finalX = xOffset + increment;
                                finalY = yOffset + increment;
                                break;
                        }
                    }
                }
            }
        }
    }
}
```

```

        break;

    case 4:
        finalX = xOffset;
        finalY = yOffset + increment * 2;
        break;

    case 5:
        finalX = xOffset + increment;
        finalY = yOffset + increment * 2;
        break;

    default:
        throw new InvalidOperationException("Indice no valido en la matriz Braille");
    }

    gcode.Add($"G1 X{finalX:F2} Y{finalY:F2};");
    gcode.Add("M3 S1000;");
    //gcode.Add("G4 P0.4;");
    gcode.Add("M3 S0;");
    }
}

xOffset += xIncrementWord;
limitWordRow--;

if (limitWordRow == 0)
{
    yOffset += yIncrementRow;
    xOffset = 10.00;
    limitWordRow = 29;
}
}
}

gcode.Add("G1 X0.00 Y310.00;");
gcode.Add("G10 P0 L20 Y0;");
gcode.Add("G0 X0 Y0;");
return string.Join("\n", gcode);
}
}

```

Form1.cs

```
using System.IO.Ports;
using System.Reflection.Metadata;
using System.Text;

namespace ImpresoraBraille
{
    public partial class Form1 : Form
    {
        private SerialPort serialPort;
        public Form1()
        {
            InitializeComponent();

            string[] ports = SerialPort.GetPortNames();
            comPortComboBox.Items.AddRange(ports);

            serialPort = new SerialPort();
        }

        private async void sendButton_Click(object sender, EventArgs e)
        {
            if (comPortComboBox.SelectedItem != null)
            {
                string selectedPort = comPortComboBox.SelectedItem.ToString();
                string textToSend = inputTextBox.Text;
                string gcodeToSend = GCodeGenerator.GeneratorGCode(textToSend.ToLower());
                richTextBox1.Text = gcodeToSend;

                try
                {
                    serialPort.PortName = selectedPort;
                    serialPort.BaudRate = 115200;
                    serialPort.Open();

                    // Limpiar cualquier residuo en el buffer
                    serialPort.DiscardInBuffer();

                    await Task.Delay(2000);

                    serialPort.WriteLine("$X;");

                    await Task.Delay(500);

                    string response = serialPort.ReadExisting();

                    // Mostrar la respuesta en el TextBox
                    labelResponse.Text = response + Environment.NewLine;

                    string[] lines = gcodeToSend.Split('\n');

                    foreach (string line in lines)
                    {
                        serialPort.WriteLine(line);

                        // Agrega un tiempo de espera antes de intentar leer la respuesta
                        await Task.Delay(200);
                    }
                }
            }
        }
    }
}
```

```

        // Leer la respuesta del puerto serial
        response = serialPort.ReadExisting();

        // Mostrar la respuesta en el TextBox
        labelResponse.Text = response + Environment.NewLine;
    }

    statusLabel.Text = "Impresion Terminada.";
}
catch (Exception ex)
{
    statusLabel.Text = "Error al enviar el código G: " + ex.Message;
}
finally
{
    if (serialPort.IsOpen)
    {
        serialPort.Close();
    }
}
}
else
{
    statusLabel.Text = "Selecciona un puerto COM.";
}
}
}

private void pictureBox1_Click(object sender, EventArgs e)
{
}

private void pictureBox2_Click(object sender, EventArgs e)
{
}
}
}
}

```

BrailleDictionary.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ImpresoraBraille;

public class BrailleDictionary
{
    public static Dictionary<char, int[]> BrailleChars { get; }

    static BrailleDictionary()
    {
        BrailleChars = new Dictionary<char, int[]>
        {
            {'a', new int[] {1, 0, 0, 0, 0, 0}},
            {'b', new int[] {1, 0, 1, 0, 0, 0}},
            {'c', new int[] {1, 1, 0, 0, 0, 0}},
            {'d', new int[] {1, 1, 0, 1, 0, 0}},
            {'e', new int[] {1, 0, 0, 1, 0, 0}},
        }
    }
}

```

```

    {'f', new int[] {1, 1, 1, 0, 0, 0}},
    {'g', new int[] {1, 1, 1, 1, 0, 0}},
    {'h', new int[] {1, 0, 1, 1, 0, 0}},
    {'i', new int[] {0, 1, 1, 0, 0, 0}},
    {'j', new int[] {0, 1, 1, 1, 0, 0}},
    {'k', new int[] {1, 0, 0, 0, 1, 0}},
    {'l', new int[] {1, 0, 1, 0, 1, 0}},
    {'m', new int[] {1, 1, 0, 0, 1, 0}},
    {'n', new int[] {1, 1, 0, 1, 1, 0}},
    {'o', new int[] {1, 0, 0, 1, 1, 0}},
    {'p', new int[] {1, 1, 1, 0, 1, 0}},
    {'q', new int[] {1, 1, 1, 1, 1, 0}},
    {'r', new int[] {1, 0, 1, 1, 1, 0}},
    {'s', new int[] {0, 1, 1, 0, 1, 0}},
    {'t', new int[] {0, 1, 1, 1, 1, 0}},
    {'u', new int[] {1, 0, 0, 0, 1, 1}},
    {'v', new int[] {1, 0, 1, 0, 1, 1}},
    {'w', new int[] {0, 1, 1, 1, 0, 1}},
    {'x', new int[] {1, 1, 0, 0, 1, 1}},
    {'y', new int[] {1, 1, 0, 1, 1, 1}},
    {'z', new int[] {1, 0, 0, 1, 1, 1}},
    {'0', new int[] {0, 1, 1, 0, 0, 1}},
    {'1', new int[] {0, 1, 1, 1, 0, 1}},
    {'2', new int[] {1, 0, 0, 0, 1, 1}},
    {'3', new int[] {1, 0, 1, 0, 1, 1}},
    {'4', new int[] {1, 1, 0, 0, 1, 1}},
    {'5', new int[] {1, 1, 0, 1, 1, 1}},
    {'6', new int[] {1, 0, 0, 1, 1, 1}},
    {'7', new int[] {1, 1, 1, 0, 1, 1}},
    {'8', new int[] {1, 1, 1, 1, 1, 1}},
    {'9', new int[] {1, 0, 1, 1, 1, 1}},
    {'.', new int[] {0, 0, 1, 1, 0, 1}}, // Punto decimal
    {',', new int[] {0, 0, 1, 0, 0, 1}}, // Coma
    {'?', new int[] {0, 1, 0, 1, 1, 0}}, // Signo de interrogación
    {'!', new int[] {0, 1, 1, 0, 1, 1}}, // Signo de exclamación
    {'@', new int[] {0, 1, 1, 1, 0, 1}}, // Símbolo de arroba
    {'&', new int[] {0, 1, 0, 0, 1, 0}}, // Ampersand
    {'+', new int[] {0, 1, 0, 1, 0, 1}}, // Signo de suma
    {'-', new int[] {0, 1, 0, 0, 0, 1}}, // Signo de resta
    {'*', new int[] {0, 1, 0, 1, 1, 1}}, // Asterisco
    {'/', new int[] {0, 1, 0, 1, 0, 0}}, // Barra inclinada
    {'=', new int[] {0, 1, 0, 0, 1, 1}}, // Signo de igual
    {' ', new int[] {0, 0, 0, 0, 0, 0}},
};
}
}

```



FranklinAntonioLopezSilva_PropuestaTecnologica

3%
Textos
sospechosos



3% Similitudes
< 1% similitudes
entre comillas
< 1% Idioma no
reconocido

Nombre del documento: FranklinAntonioLopezSilva_PropuestaTecnologica.docx
ID del documento: b5a0d619a1188e692d670746309e805e0eaf63c6
Tamaño del documento original: 20 MB

Depositante: SENDEY AGUSTIN VERA GONZALEZ
Fecha de depósito: 3/1/2024
Tipo de carga: interface
fecha de fin de análisis: 3/1/2024

Número de palabras: 24.131
Número de caracteres: 157.081

Ubicación de las similitudes en el documento:



Fuentes principales detectadas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	repositorio.puce.edu.ec Diseño de material educativo para niños y niñas de 5 a ... http://repositorio.puce.edu.ec/80/vmlu/bitstream/22000/5629/7/T-PUCE-5785(1).pdf.txt	< 1%		Palabras idénticas: < 1% (126 palabras)
2	repositorio.upse.edu.ec https://repositorio.upse.edu.ec/bitstream/46000/8877/1/UPSE-TET-2022-0004.pdf 3 Fuentes similares	< 1%		Palabras idénticas: < 1% (100 palabras)
3	repositorio.upse.edu.ec Implementación de un prototipo de robot móvil con visi... http://repositorio.upse.edu.ec/8080/jspui/bitstream/46000/1598/3/IMPLEMENTACIÓN_DE_UN_PROTO... 3 Fuentes similares	< 1%		Palabras idénticas: < 1% (70 palabras)
4	repositorio.upse.edu.ec https://repositorio.upse.edu.ec/bitstream/46000/8260/1/UPSE-TET-2022-0001.pdf 3 Fuentes similares	< 1%		Palabras idénticas: < 1% (64 palabras)
5	dspace.espoch.edu.ec Diseño y construcción de una impresora braille de bajo co... http://dspace.espoch.edu.ec/bitstream/123456789/6521/3/1/08T0190.pdf.txt	< 1%		Palabras idénticas: < 1% (67 palabras)

Fuentes con similitudes fortuitas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Documento de otro usuario #d1e264 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (40 palabras)
2	www.instrumentaciondigital.es Actuadores eléctricos y sus funciones principale... https://www.instrumentaciondigital.es/actuadores-electricos-y-sus-funciones/	< 1%		Palabras idénticas: < 1% (38 palabras)
3	Documento de otro usuario #41c554 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (22 palabras)
4	repositorio.upse.edu.ec Estudio de factibilidad para virtualización en la nube co... http://repositorio.upse.edu.ec/8080/jspui/bitstream/46000/3681/6/UPSE-TIV-2016-0040.pdf.txt	< 1%		Palabras idénticas: < 1% (31 palabras)
5	repositorio.upse.edu.ec https://repositorio.upse.edu.ec/bitstream/46000/9262/1/UPSE-TI1-2023-0022.pdf	< 1%		Palabras idénticas: < 1% (35 palabras)

Fuentes mencionadas (sin similitudes detectadas) Estas fuentes han sido citadas en el documento sin encontrar similitudes.

- <https://0.wp.com/blog.cemebe>
- <https://www.tiflastore.cl/>
- <https://prevenirlaceguera.blogspot.com>
- <https://www.sta-mexico.com>
- <https://opencluster-tech.cl>