



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

AUTORES: NOBOA MONTENEGRO, ANDREA ESTEFANÍA

NOBOA MONTENEGRO, DIANA SOFÍA

**TEMA: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
ELECTRÓNICO CON INTERFACE A PC PARA AUTOMATIZAR UNA
MÁQUINA DE ESCRIBIR BRAILLE”**

DIRECTOR: ING. TIPÁN, EDGAR

CODIRECTOR: ING. IBARRA, ALEXANDER

SANGOLQUÍ, JULIO 2014

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL

CERTIFICADO

Ing. Edgar Tipán
Ing. Alexander Ibarra

CERTIFICAN

Que el trabajo titulado “**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO CON INTERFACE A PC PARA AUTOMATIZAR UNA MÁQUINA DE ESCRIBIR BRAILLE**”, realizado por Andrea Estefanía Noboa Montenegro y Diana Sofia Noboa Montenegro, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas – ESPE.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

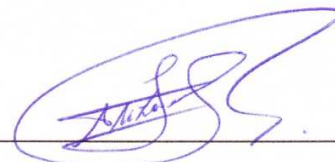
El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a las Srtas. Andrea Estefanía Noboa Montenegro y Diana Sofia Noboa Montenegro que entreguen al Ingeniero Luis Orozco Brito MSc, en su calidad de Director de la Carrera.

Sangolquí, 28 de Julio del 2014



Ing. Edgar Tipán.

DIRECTOR



Ing. Alexander Ibarra MSc.

CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL

DECLARACIÓN DE RESPONSABILIDAD

Nosotras, Andrea Estefanía Noboa Montenegro y
Diana Sofía Noboa Montenegro

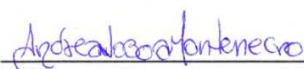
DECLARAMOS QUE:

El proyecto de grado **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO CON INTERFACE A PC PARA AUTOMATIZAR UNA MÁQUINA DE ESCRIBIR BRAILLE”**, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie, de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 28 de Julio del 2014



Andrea Estefanía Noboa Montenegro



Diana Sofía Noboa Montenegro

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

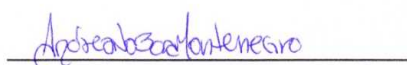
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL

AUTORIZACIÓN

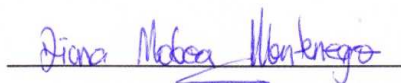
Nosotras, Andrea Estefanía Noboa Montenegro y
Diana Sofía Noboa Montenegro

Autorizamos a la Universidad de las Fuerzas Armadas – ESPE la publicación, en la biblioteca virtual de la Institución del trabajo **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO CON INTERFACE A PC PARA AUTOMATIZAR UNA MÁQUINA DE ESCRIBIR BRAILLE”**, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 28 de Julio del 2014



Andrea Estefanía Noboa Montenegro



Diana Sofía Noboa Montenegro

DEDICATORIA

Quiero dedicar este trabajo a las personas especiales en mi vida, que de una u otra manera, han sabido ayudarme y apoyarme incondicionalmente en todo momento.

A Dios por darme la fortaleza para sobrellevar cada bueno y mal momento.

A mis padres Marcelo y Rosa Soledad por su amor, cariño, comprensión, soporte y apoyo, que me han brindado todo el tiempo.

A mis hermanas Soledad, Marcela y Diana por ser mis cómplices y mejores amigas, por hacer de cada momento algo inolvidable y por el cariño y amor muy fuerte que nos tenemos.

A mi novio Esteban, por su apoyo y ayuda incondicional siempre y en todo momento, a lo largo de todos estos años.

A Roberto por ser un niño muy alegre, feliz y sobretodo especial en mi vida, Dios te bendiga siempre.

Andrea Noboa Montenegro

DEDICATORIA

Dedico mi gran esfuerzo y perseverancia a Dios, no puedo pasar por alto la fortaleza que me brindo cuando sentía que me abatida.

A mis queridos y amados padres, Marcelo por enseñarme día a día que lo que se desea se lo obtiene luchando sin rendirse, que los sueños siempre son grandes pero nunca difíciles de alcanzar, Rosa Soledad por dedicarme su tiempo y paciencia y decirme las palabras más dulces y llenas de amor que me ayudaron a seguir adelante.

A mis hermanas: Soledad y Marcela por darme su apoyo incondicional, por tener siempre esa palabra de aliento que me impulsaba a seguir adelante reconociendo que siempre el esfuerzo tiene su recompensa. Andrea mi hermana, mi amiga, por ser mi apoyo, porque este sueño lo empezamos juntas y juntas lo terminamos.

A mi novio Christian, por ser mi fortaleza y mi motor de seguir luchando por lo que deseo, porque siempre has estado a mi lado empujándome y ayudándome a que las cosas sigan su camino.

Y en especial a todos los niños que tienen un corazón inmenso, que ven con los ojos del alma y conocen en lo más profundo la verdadera esencia de las personas. A Roberto por tener una gran inocencia y por ser ese niño que todos llevamos dentro.

Diana Noboa Montenegro.

AGRADECIMIENTO

Queremos agradecer a Dios por permitirnos cumplir con esta meta tan grande para nosotras, por no rendirnos nunca, darnos la fortaleza y fe para seguir adelante cumpliendo todos nuestros sueños, metas y objetivos.

Agradecemos al Ingeniero Edgar Tipán por todo el tiempo que se ha tomado para ayudarnos, guiarnos, darnos ideas orientarnos y llevar a cabo la finalización de éste proyecto con éxito.

Agradecemos al Ingeniero Alexander Ibarra por todas las recomendaciones, sugerencias, ayuda y apoyo que nos brindó durante la elaboración del proyecto.

Gracias Ingenieros por todas las enseñanzas que hemos recibido de ustedes a lo largo de nuestra formación personal y académica.

Agradecemos de todo corazón la buena voluntad y colaboración incondicional de la Sra. Lilia Tipán, que con toda su paciencia y desinterés nos ayudó en todo momento, sobretodo nos ayudó con la herramienta más importante de éste proyecto: la máquina de escribir Braille Perkins, gracias por su amistad, por todos los ánimos que nos dio, y los buenos deseos que siempre nos brindó.

Agradecemos a nuestros padres por tener mucha paciencia con nosotras y ayudarnos incondicionalmente en todo momento, gracias por todo, sin ustedes no sería igual.

Diana Noboa Montenegro

Andrea Noboa Montenegro

INDICE DE CONTENIDO

CAPÍTULO 1.....	3
GENERALIDADES	3
1.1. Introducción.....	3
1.2. Antecedentes.....	4
1.3. Justificación e importancia	8
1.4. Alcance	8
1.5. Objetivos.....	9
1.5.1. General	9
1.5.2. Específicos.....	9
CAPÍTULO 2.....	10
MARCO TEÓRICO.....	10
2.1 Discapacidades en el Ecuador	10
2.1.1 Situación de las personas no videntes.....	12
2.2 Sistema Braille.....	13
2.2.1 Historia del Braille	15
2.2.2 Sistema Braille.....	17
2.2.3 Introducción al Alfabeto Braille	18
2.2.4 Escritura y Lectura Braille.....	24
2.2.5 Ventajas y desventajas.....	26
2.2.6 Impresión Braille	27
2.3 Recursos de Hardware	31
2.3.1 Motores eléctricos	31
2.3.2 Actuador Lineal	41

2.3.3	Microcontroladores.....	45
2.3.4	Bus Universal en Serie USB.....	48
2.3.5	Fuente de Alimentación.....	51
2.4	Recursos de Software	51
2.4.1	CCS Compiler	51
2.4.2	Java.....	55
2.4.3	NetBeans IDE.....	57
CAPÍTULO 3.....		59
DISEÑO DE HARDWARE		59
3.1	Elección y justificación de los elementos a usar	59
3.1.1	Análisis de la fuerza que necesita el actuador para presionar una tecla de la máquina de escribir Braille	59
3.1.2	Actuadores Eléctrico, Hidráulico y Neumático	62
3.1.3	Especificaciones y características del actuador eléctrico	64
3.1.4	Distribución de los pines utilizados.....	69
3.2	Bus Universal en Serie (Universal Serie Bus – USB)	71
3.2.1	USB del microcontrolador 18f2550	71
3.3	Construcción de la estructura mecánica	74
3.4	Diseño de la placa electrónica	79
3.4.1	Conexión del microcontrolador 18f2550.....	79
3.4.2	Conexiones de los actuadores lineales eléctricos	80
CAPÍTULO 4.....		91
DISEÑO DE SOFTWARE		91
4.1	Desarrollo del código fuente	91
4.1.1	Programación del CCS Compiler para el microcontrolador 18f2550.....	91

4.1.2	Líneas de programación usando código fuente Java en el entorno de NetBeans	93
4.2.	Diseño de la interfaz grafica.....	105
4.2.1.	Barra de menú.....	106
CAPÍTULO 5.....		116
ANALISIS DE RESULTADOS		116
5.1	Parámetros de funcionamiento	116
5.1.1	Instrumentos de medición.....	117
5.2	Pruebas realizadas en el hardware	117
5.2.1	Trabajo del actuador lineal	117
5.2.2	Trabajo en el motor DC	123
5.3	Pruebas realizadas en el software	125
5.3.1	Conexión USB.....	125
5.3.2	Barra de Menú y Botones	126
5.3.3	Impresión.....	128
5.4	Prueba Global del Sistema.....	129
5.5	Análisis de resultados	138
5.6	Manual de operación y mantenimiento	141
5.7	Análisis económico	141
5.7.1	Costos directos.....	141
5.7.2	Costos indirectos	143
5.7.3	Resumen de Costos.....	143
5.8	Costo / beneficio.....	144
CAPÍTULO 6:.....		145
CONCLUSIONES Y RECOMENDACIONES.....		145

INDICE DE FIGURAS

Figura 1 Regleta y Punzón. (BAC).....	6
Figura 2 Máquina de Escribir Braille – Perkins (BAC).....	7
Figura 3 Signo generador del Sistema Braille	17
Figura 4 Distribución y Tamaño del Signo Generador.....	18
Figura 5 Alfabeto Braille.....	19
Figura 6 Vocales con Acento.....	20
Figura 7 Signos Especiales.....	20
Figura 8 Letras del Alfabeto Braille de la Serie 1.....	20
Figura 9 Letras del Alfabeto Braille de la Serie 2.....	21
Figura 10 Letras del Alfabeto Braille de la Serie 3.....	21
Figura 11 Letras del Alfabeto Braille de la Serie 4.....	21
Figura 12 Letras del Alfabeto Braille de la Serie 5.....	21
Figura 13 Vocales con Signo.....	21
Figura 14 Signos de Interés.....	22
Figura 15 Signos de letras mayúsculas y números.....	22
Figura 16 Ejemplos de letras en mayúsculas.....	23
Figura 17 Representación de los Números.....	23
Figura 18 Representación de números enteros de más de dos cifras	23
Figura 19 Representación de números decimales	24
Figura 20 Representación de Signos Matemáticos	24
Figura 21 Lectura unimanual.....	26
Figura 22 Lectura bimanual.....	26
Figura 23 Máquina Braille con sus respectivas partes. (Tiflología)	28

Figura 24 Posición de las teclas según los puntos del signo generador.....	29
Figura 25 Impresora Braille. (BAC).....	31
Figura 26 Clasificación de los Motores DC.....	32
Figura 27 Clasificación de los Motores AC.....	33
Figura 28 Motor DC.....	33
Figura 29 Polos magnéticos que generan el movimiento de un motor	34
Figura 30 Circuito del Puente H para controlar el motor.....	35
Figura 31 Modulación PWM para controlar la velocidad del motor.	36
Figura 32 Estructura interna de un servomotor.....	37
Figura 33 Diferentes anchos de pulso de un PWM.....	38
Figura 34 Motores paso a paso.	39
Figura 35 Secuencia de movimiento de motor paso a paso.	40
Figura 36 Secuencia del motor paso a paso.	40
Figura 37 Actuadores eléctrico, neumático e hidráulico.....	41
Figura 38 Solenoide de núcleo móvil	42
Figura 39 Solenoides de empuje y arrastre.	43
Figura 40 Piñón y cremallera.....	43
Figura 41 Actuador Lineal Eléctrico.....	45
Figura 42 Bus Universal en Serie	49
Figura 43 Plug del USB	50
Figura 44 Distribución de los cables del conector USB	51
Figura 45 Ventana Principal del CCS Compiler IDE	53
Figura 46 Estructura del Programa CCS Compiler.....	54
Figura 47 Interfaz de NetBeans IDE.....	58
Figura 48 Peso de 0.45 [kg], Figura 49 Tecla presionada por 0.45 [kg].....	60

Figura 50 Peso de 0.63 [kg], Figura 51 Tecla presionada por 0.63 [kg].....	60
Figura 52 Peso 0.77 [kg], Figura 53 Tecla presionada por 0.77 [kg].....	60
Figura 54 Peso de 1[kg] , Figura 55 Tecla presionada por 1 [kg].....	61
Figura 56 Actuador lineal de 20 [mm].....	67
Figura 57 Actuador Lineal Eléctrico.....	67
Figura 58 PIC 18f2550.....	69
Figura 59 Distribución de los pines Microcontrolador 18f25550.....	70
Figura 60 Energía entregada por el USB.....	72
Figura 61 Fuente de alimentación 12 [V] dc 15 [A].....	72
Figura 62 Estructura metálica de acero estructural.....	75
Figura 63 Soportes de nylon para teclas.....	75
Figura 64 Soportes de nylon acoplados a los actuadores lineales.....	76
Figura 65 Activación del actuador lineal.....	76
Figura 66 Topes de nylon para sujeción de la máquina.....	77
Figura 67 Motor reductor, poleas dentadas, banda.....	77
Figura 68 Sistema del recorrido del carro implementado en la estructura.....	78
Figura 69 Carcasa de la estructura mecánica.....	78
Figura 70 Posicionamiento de los actuadores y soportes.....	78
Figura 71 Posicionamiento del sistema del recorrido del carro.....	79
Figura 72 Conexión del microcontrolador 18f2550.....	80
Figura 73 Circuito de conexión del actuador.....	84
Figura 74 Funcionamiento del integrado L293D.....	86
Figura 75 Conexión del integrado L293D con el microcontrolador 18f2550 y los actuadores lineales.....	86
Figura 76 Distribuciones de pines del integrado L293D.....	87

Figura 77 Diseño Placa Electrónica en el simulador Proteus	89
Figura 78 Diseño de la placa electrónica en ARES para el circuito impreso.....	90
Figura 79 Diseño de la placa electrónica, circuito impreso	90
Figura 80 Circuito Impreso	90
Figura 81 Diagrama de flujo del programa del microcontrolador	92
Figura 82 Paquetes existentes en el proyecto.....	93
Figura 83 Asignación del Signo Generador al Código Braille	94
Figura 84 Diagrama de flujo de la clase controlador_decodificador.	96
Figura 85 Diagrama de flujo de la clase controlador_principal.....	98
Figura 86 Diagrama de flujo de la clase controlador_principal.....	99
Figura.87 Diagrama de flujo de la clase envio_usb	101
Figura 88 Diagrama de flujo de la clase recibo_usb.....	102
Figura 89 Diagrama de flujo	103
Figura 90 Diagrama de Flujo de la clase Gemes_tesis	104
Figura 91 Interfaz gráfica del proyecto.....	105
Figura 92 Barra de menús de la interfaz gráfica del proyecto	105
Figura 93 Contenido del Menú “Archivo”.....	106
Figura 94 Ventana de Dialogo que permite abrir un documento	108
Figura 95 Archivo abierto.....	108
Figura 96 Ventana emergente que indica un error.....	109
Figura 97 Ventana de dialogo menú Guardar	109
Figura 98 Ventana de dialogo que muestra advertencia	110
Figura 99 Ventana emergente, da opciones a imprimir	111
Figura 100 Interfaz, el documento se está imprimiendo.....	111
Figura 101 Ventaja emergente que indica si se desea cerrar el programa.	112

Figura 102 Menú “Editar”	112
Figura 103 Actuador trabajando con 5 [V]	118
Figura 104 Actuador trabajando a 12 [V]	118
Figura 105 Trabajo del actuador	119
Figura. 106 Actuadores Lineales ubicados uno al lado del otro	120
Figura. 107 Sistema de pivote acoplado a cada tecla.....	120
Figura 108 Letra “q” interpretada por los actuadores lineales	121
Figura 109 Letra “z” interpretada por los actuadores	121
Figura. 110 Ubicación del actuador lineal para el salto de línea	122
Figura. 111 Actuador lineal del salto de línea desactivado.....	122
Figura. 112 Actuador lineal del salto de línea activado	123
Figura 113 Movimiento de la banda	123
Figura 114 Sistema del carro en la posición inicial	124
Figura. 115 Sistema del carro en la posición final	124
Figura. 116 USB Desconectado	125
Figura 117 Cable USB conectado.....	126
Figura 118 Menú Abrir	126
Figura 119 Menú guardar.....	127
Figura 120 Ventana emergente que permite salir del sistema.....	127
Figura 121 Botón que permite borrar el área de texto	127
Figura 122 Proceso de impresión de texto	128
Figura 123 Presentación del software	129
Figura 124 Pantalla principal	130
Figura 125 Ingreso de texto por teclado.....	130
Figura 126 Función Guardar	131

Figura 127 Guardar Archivo	131
Figura 128 Guardado exitoso.....	131
Figura 129 Función Borrar.....	132
Figura 130 Función Abrir	132
Figura 131 Abrir Archivo	133
Figura 132 Archivo Abierto.....	133
Figura 133 Función Imprimir.....	134
Figura 134 Ventana emergente, seleccionar una de las opciones	134
Figura 135 Pantalla bloqueada.....	134
Figura 136 La Impresión acabó	135
Figura 137 Ventana emergente, escoja una de las opciones	135
Figura 138 Mensaje de advertencia, coloque nueva hoja	135
Figura 139 Escoja una opción con respecto al papel	136
Figura 140 Impresión de “Bienvenidos al sistema”	136
Figura 141 Impresión de “Bienvenidos al sistema”	137
Figura 142 Frase impresa: “Bienvendos al sistema”	137
Figura 143 Palabra “Bienvenidos”.....	137
Figura 144 Palabra “Al”.....	137
Figura 145 Palabra “Sistema”	138

INDICE DE CUADROS

Cuadro 1 Características de trabajo y funcionamiento.	47
Cuadro 2 Parámetros de Conexión.....	51
Cuadro 3 Características de archivos generados al compilar el código fuente.....	52
Cuadro 4 Tipos de Datos.....	55
Cuadro 5 Símbolos de programación.....	56
Cuadro 6 Tipo de datos en Java.	57
Cuadro 7 Actuadores Eléctricos, Neumáticos e Hidráulicos	63
Cuadro 8 Ventajas y desventajas de actuadores eléctricos	64
Cuadro 9 Especificaciones técnicas de actuadores lineales.	66
Cuadro 10 Parámetros del microcontrolador 18f2550.....	69
Cuadro 11 Características 18f2550 (Data_Sheet_PIC18F2550, 2009).....	70
Cuadro 12 Características eléctricas de elementos	73
Cuadro 13 Especificaciones técnicas de la fuente de alimentación	74
Cuadro 14 Valores máximos de trabajo del TIP 122	81
Cuadro 15 Características eléctricas del TIP 122	82
Cuadro 16 Valores máximos de entrada para el driver L293D.....	87
Cuadro 17 Función de los terminales del driver L293D.....	88
Cuadro 18 Asignación de Mnemónicos	115
Cuadro 19 Costos de actuadores y fuente	141
Cuadro 20 Costos de la estructura para la Máquina Braille.....	142
Cuadro 21 Costos de placa electrónica y elementos electrónicos.....	142
Cuadro 22 Costos varios	143
Cuadro 23 Resumen de Gastos	143

INDICE DE TABLAS

Tabla 1 Valores de la fuerza en [N]	62
Tabla 2 Corriente medida de los actuadores:	73
Tabla 3 Alfabeto Braille representado en Código Braille	95
Tabla 4 Caracteres Especiales representados en Código Binario	95

RESUMEN

El presente documento detalla el diseño e implementación de un sistema con interface a PC para automatizar una máquina de escribir Braille, en el cual se profundiza la investigación y aprendizaje del Alfabeto Braille pues se debe especificar a cada uno de los códigos Braille dependiendo de si es letra en mayúscula, letra en minúscula, números o caracteres especiales, además se debe tomar en cuenta todas las reglas gramaticales del Braille. Se desarrolla el software bajo la plataforma NetBeans junto con Java, la programación de este software tiene el objetivo principal de convertir el alfabeto latino en alfabeto Braille utilizando las herramientas que este desarrollador de software ofrece al programador y se lo hace operable al usuario mediante el uso de una interfaz gráfica sencilla y funcional para que una persona no vidente pueda operar tranquilamente el sistema con la ayuda de audio y controles mnemónicos utilizando el teclado de la computadora o en su defecto el ratón. El desarrollo del hardware se enfoca en el estudio de los diferentes dispositivos electrónicos necesarios para implementar este proyecto, tales como los actuadores lineales eléctricos que ayudaran a tener un control preciso en cada una de las teclas de la máquina, como también la comunicación USB del sistema que es un elemento fundamental en el proyecto, pues establece la comunicación entre el PC utilizando la librería jPic USB de Java y el microcontrolador 18f2550 que controla los actuadores que automatizarán al sistema.

Palabras clave:

BRAILLE

MÁQUINA DE ESCRIBIR BRAILLE

AUTOMATIZAR

JAVA

18f2550

NO VIDENTE

ABSTRACT

This paper described the design and implementation of a system with a PC interface to automate a Braille typewriter, in which research and learning Braille Alphabet deepens as you must specify each of the Braille codes depending on if uppercase letter, lowercase letter, numbers or special characters, you must also take considered all the grammatical rules of Braille. Software is developed under the NetBeans platform with Java, programming this software has the main aim of making a conversion of the Latin alphabet in Braille alphabet using the tools that the software developer provides to the programmer and the user does this interface operable using a simple and functional graphical interface for a blind person who can operate the system their self with help of audio controls and mnemonics using the computer keyboard or mouse. Hardware development focuses on the study of different electronic devices needed to implement this project, such as electric linear actuators that help to control each of the keys of the machine, as well as the USB communication system which is a key element in the project, it establishes communication between the PC using the USB library jpic Java and microcontroller 18f2550 responsible for controlling the actuators to automate the system.

Keywords:

BRAILLE

BRAILLE TYPEWRITER

AUTOMATE

JAVA

18f2550

NO PSYCHIC

JPicUSB

CAPÍTULO 1

GENERALIDADES

1.1. Introducción

Cuando se habla de automatizar un proceso, en este caso, automatizar una máquina de escribir Braille se está hablando de sustituir el proceso manual por una máquina automatizada que realizará éste trabajo. En este caso, para transcribir un texto al lenguaje Braille, el usuario tendrá que hacerlo a mano, transcribiendo letra por letra en la máquina de escribir Braille, con la automatización de éste proceso, el usuario deberá tener el texto en formato digital y el sistema automatizado empezará a transcribir el texto al lenguaje Braille solamente con la orden de ejecución realizada por el usuario.

Al hablar de automatización, también se debe hablar de ventajas evidentes que esto trae hacia el proceso, pues en primer lugar, se tiene una mejoría en los costos del proceso, al implementar un sistema que trabaje con más eficiencia y eficacia que un trabajo manual, la rapidez del trabajo se duplica, se puede tener un control de calidad al realizarse un trabajo más uniforme y un mejor servicio que se brinda hacia el usuario.

Por otra parte, se debe tener un conocimiento básico sobre el origen de la escritura en Braille. Se conoce que en el año 1819, Charles Barbier propuso la utilización de puntos en relieve para la escritura de textos, desarrollando un sistema que se basaba en la utilización de dos matrices de 6 líneas y 6 columnas respectivamente para representar las letras del alfabeto y ciertos sonidos habituales, logrando incorporar cerca de 4000 signos diferentes, en donde cada caracter podría ser representado máximo por 12 puntos o mínimo 2 puntos en relieve.

A Louis Braille, quien era una persona no vidente desde los 4 años de edad, le pareció muy complejo el sistema de Barbier, por este motivo empezó a mejorarlo y simplificarlo, proponiendo la utilización de una sola matriz de 3 filas y 2 columnas

para representar la unión entre dos caracteres, escribiendo cada carácter de derecha a izquierda con la ayuda de un punzón.

Gracias al adelanto tecnológico, las personas no videntes hoy en día, pueden tener la suerte de tener el código Braille incorporado en diversos elementos y herramientas como las máquinas de escribir, equipos, programas informáticos, libros, impresoras Braille, síntesis de voz, Braille hablado, entre otros, facilitando su desempeño en la sociedad, ya que cuentan con los medios necesarios para alcanzar una autonomía personal,. (Espejo).

1.2. Antecedentes

Anteriormente en el Ecuador, las atenciones hacia las necesidades para personas con diferentes discapacidades fueron muy pocas, básicamente fue un criterio de caridad y beneficencia, pero, años más tarde, estas necesidades fueron haciéndose prioridades para varios organismos tomando en cuenta los campos de la educación, el bienestar social, salud, entre otros, hasta convertirse actualmente en una necesidad principal de ayuda, colaboración, cooperación y sobretodo reinserción hacia la comunidad de todas las personas con diferentes rangos de discapacidades físicas.

En el Ecuador existen aproximadamente 15 millones de habitantes tomando como referencia el dato del Instituto Nacional de Estadística y Censos (INEC) del año 2010 (INEC, 2010), de los cuales, según las estadísticas del Consejo Nacional de Igualdad de Discapacidades (Conadis), existen en el país 42082 personas con discapacidad visual (CONADIS, 2013), puede que no sea una discapacidad al 100%, pero existen diferentes categorías que hacen que sea una discapacidad, haciendo de la inclusión social algo importantísimo para poder acceder a diferentes oportunidades, mejorando la condición de vida y sobretodo del aprendizaje.

La vista, desde el momento del nacimiento, es un canal sensorial social. Según estudios realizados, hasta los doce años la mayoría de las nociones aprendidas se captan a través de la vista, en una proporción del 83%, frente a los estímulos captados por los otros sentidos, que se reparten entre el 17% de los restantes.

Los ojos que comienzan captando tan sólo un juego de luces y sombras, activan zonas del cerebro que emiten respuestas motrices, y esta actividad sensorio-motriz es la clave del desarrollo del niño/a. Lo que con el ojo observa, quiere tocar con la mano y cuando ha tocado aquello, quiere ir más lejos. A la primera etapa de concentración visual sigue otra de atención, y a estas dos una tercera de reconocimiento visual. (Valdez).

Es por esto que nace el Sistema Braille para que las personas no videntes puedan tener una comunicación con el medio que los rodea. Por el año de 1819 el capitán del ejército francés Charles Barbier propuso la idea de utilizar puntos en relieve, los mismos que los llamó “escritura nocturna”, éste sistema fue un sistema de letras con puntos, es decir, un código alfabético basado en grupos de puntos, años más tarde este sistema lo probó Louis Braille reinventándolo utilizando un sistema de 8 puntos, aunque lo simplificó dejándole en el sistema universalmente conocido y adaptado de 6 puntos.

Desde que Louis Braille reinventó el sistema Braille, muchas personas no videntes han contado con una herramienta muy eficaz para poder escribir, leer, componer o dedicarse a ciertas tareas que estaban limitados por falta de conocimientos.

Este sistema es un alfabeto universal, no un idioma, ya que se puede representar letras, signos de puntuación, números, símbolos matemáticos, música, etc. En un sistema de celdas de seis puntos en relieve, los mismos que se encuentran organizados en una matriz de tres filas por dos columnas, que están numeradas de arriba hacia abajo y de izquierda a derecha.

La codificación de los símbolos dependerá de la ausencia o presencia de puntos, es decir la ausencia o presencia de puntos determinará que letra del alfabeto corresponde.

El Sistema Braille se adecua estructural y fisiológicamente a las características del sentido del tacto.

Se adapta perfectamente a las terminaciones nerviosas de la yema de los dedos, y así los signos son transmitidos al cerebro, y la persona no vidente los entiende perfectamente. La lectura en Braille no presenta excesiva dificultad respecto a la lectura en tinta. Los elementos básicos en el proceso de adquisición de la lectura son los mismos para personas no videntes y también para personas videntes, pues se la realiza mediante el tacto letra a letra y no a través del reconocimiento de las palabras completas, como sucede en tinta.

Por ello se trata de una tarea lenta en un principio, que requiere de una gran concentración difícil de alcanzar a edades tempranas.

La escritura del Braille es más rápida que la lectura y suele presentar menos dificultad, además de que se la puede elaborar a mano o a máquina. En la escritura a mano se precisa disponer de una pauta o de una regleta, de un punzón y de un papel.

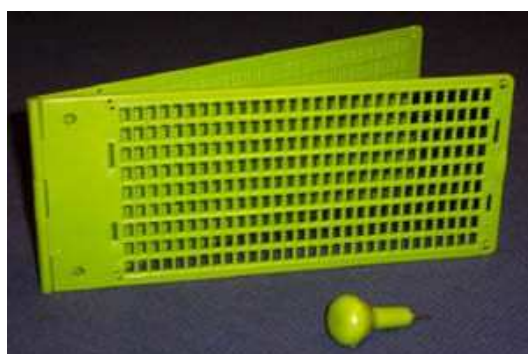


Figura 1 Regleta y Punzón. (BAC)

Pasos que se deben tomar en cuenta para utilizar la regleta y punzón:

- Para que la lectura de lo escrito a mano pueda realizarse normalmente de izquierda a derecha, es necesario empezar a escribir de derecha a izquierda, invirtiendo la numeración de los puntos del cajetín. De esta manera el rehundido que se hace al escribir quedará como un punto en relieve situado en el lugar correcto cuando se le da la vuelta al papel.
- Antes de empezar conviene adquirir precisión mecánicamente en el punteado por lo que se pueden hacer series de puntos.

- Todos los puntos deben tener un relieve idéntico. Para ello hay que adquirir una gran precisión mecánica.

En la escritura a máquina en Braille contiene 6 teclas, una para cada uno de los puntos del signo generador de Braille. También tiene un espaciador, una tecla para el retroceso y otra para el cambio de línea



Figura 2 Máquina de Escribir Braille – Perkins (BAC)

El modelo denominado Perkins - Brailier, fabricado por la Perkins School of the Blinds en Massachusetts, USA, es la máquina empleada comúnmente.

Las teclas se pueden pulsar cada una por separado o bien simultáneamente, permitiendo construir la combinación que constituye un elemento Braille de una sola vez. Cada tecla debe pulsarse con un dedo determinado, de forma que la escritura se realiza con la máxima rapidez y el mínimo esfuerzo, procurando que la disposición de las manos sea lo más cómoda posible. (El Portal de las Personas con Discapacidad)

El avance de la tecnología parece relegar a ciertos sectores de la población, entre los cuales se encuentran las personas con capacidades especiales específicamente las personas no videntes.

Las máquinas de escribir mecánicas en código Braille fueron evolucionando hasta crearse las impresoras Braille que son altamente útiles e invaluable si se considera que gracias a ellas es posible un avance mucho más rápido en el aprendizaje y preservación de ideas que las personas no videntes pueden expresar.

1.3. Justificación e importancia

Sin padecer alguna discapacidad, en algún momento, se puede sufrir algún percance que interfiera en la vida cotidiana surgiendo una discapacidad, una de ellas podría ser una discapacidad visual, y es en ese momento que se le da la importancia que tiene el sistema Braille, la necesidad de investigar y conocer más sobre el Braille, llegando a comprender las diferentes dificultades que deben vivir día a día las personas no videntes en una sociedad en la cual la indiferencia existe y la falta de conocimientos y recursos limitan tantos alcances que pueden llegar a tener las personas con alguna discapacidad, logrando así, una inclusión social justa y total.

El proyecto se justifica por la necesidad de darle una solución al tema planteado, por lo tanto, la universidad a través de sus docentes y estudiantes, por medio de los diferentes conocimientos adquiridos, se debe proporcionar un sistema automatizado de escritura mediante la máquina de escribir Braille, de tal manera que se pueda acceder a un texto en Braille de una forma más rápida.

1.4. Alcance

El proyecto se limita a realizar un sistema automático de escritura mediante el uso de la máquina de escribir Braille, un PC y un sistema de control electrónico que facilite el acceso a textos escritos en Braille para personas no videntes, de esta manera se obtendrán textos en Braille a partir de cualquier texto en formato digital, cumpliendo con las normas específicas de este tipo de escritura.

Se automatizará una máquina de escribir Braille creando un sistema electromecánico implementado por actuadores eléctricos que accionarán cada una de las teclas de la máquina, de esta manera se podrá transferir cualquier texto convencional a escritura Braille, teniendo como fuente base un texto en formato digital, y mediante el diseño de un software especial se convertirá éste texto en Lenguaje Braille, el mismo que se enviará al controlador de la máquina de escribir para su respectiva escritura.

1.5. Objetivos

1.5.1. General

- Diseñar e implementar un sistema electrónico con interface a PC para automatizar una máquina de Escribir Braille

1.5.2. Específicos

- Investigar, conocer e indagar las normas y parámetros del alfabeto, lectura y escritura Braille para tener un conocimiento previo al estudio e implementación del proyecto.
- Identificar los diferentes elementos y componentes que se van a usar para controlar el sistema electrónico de automatización.
- Acoplar mecánicamente el sistema de automatización a la máquina de escribir para controlar su funcionamiento
- Diseñar el software específico para convertir el texto convencional en texto Braille
- Realizar un análisis económico y los costos de diseño e implementación del sistema de automatización.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Discapacidades en el Ecuador

“Las personas videntes suelen tener millares de deseos; el ciego, sólo uno...” .

Louis Braille.

Cuando se piensa en discapacidad hay que tener en cuenta de que no se trata de algún padecimiento o enfermedad, mas bien, se la debe considerar como la restricción o impedimento de la capacidad de un ser humano para realizar una actividad determinada.

Según estudios realizados en el Ecuador, se ha encontrado que existen alrededor de 294.000 personas con discapacidad con una prevalencia del problema de 2,43 por ciento, tomando en cuenta que los discapacitados se ubican especialmente en la provincia costera de Guayas (74.800 casos) y en la andina de Pichincha (45.000 casos) , las más pobladas del país. Su distribución por sexo es de 49,57 por ciento mujeres y 50,43 por ciento hombres, esto según estadísticas del INEC para el año 2010. (Herrera, 2010)

En la actualidad el Ecuador, ha evolucionado la idea de inclusión hacia las personas con discapacidad, gracias a la ayuda e implementación de proyectos por parte de la Vicepresidencia de la República, que ha dirigido su objetivo principal hacia el respeto de los derechos de las personas con discapacidad tal como lo ordena la Constitución de la República, en donde, muchas de las veces éstas personas permanecían en el anonimato excluidos incluso por su propia familia, que se avergonzaba de ellas, y la del Estado ecuatoriano, que no las insertaba en sus políticas sociales.

La Constitución de la República del Ecuador, pone en marcha el cumplimiento de principios, derechos y procedimientos de las normas con el objetivo de incluir a

las personas no videntes dentro de la sociedad con una igualdad de derechos y oportunidades, brindándoles una calidad de vida superior.

Es lamentable observar en el país la falta de cultura para tratar con personas no videntes y la parcial inclusión en la sociedad, pues es muy claro ver que en servicios públicos como transporte, aún no presentan las facilidades necesarias de uso a éstos usuarios, deben andar acompañados de alguien que les ayude a tomar un bus; o al momento de caminar por las calles, hace mucha falta semáforos que tengan alarmas y les ayuden a movilizarse de manera más independiente.

Ahora, viene lo más importante y difícil de lograr: el factor humano, pues en la sociedad aún es muy común discriminar a una persona con discapacidad, sin despertar la más mínima conciencia de solidaridad y ayuda hacia una persona que tiene dificultad. El momento en que cambien los valores humanos y el sentido de apoyo, se llegara a tener una sociedad que colabore con esta inclusión que tantos ecuatorianos aún deseamos.

En Ecuador existe un sinnúmero de instituciones y fundaciones en pro ayuda de las personas no videntes, como es el caso de la Federación Nacional de Ciegos del Ecuador (FENCE), que es una organización autónoma que agrupa a instituciones y organizaciones de y para ciegos, que coordina, asesora, capacita y defiende derechos, impulsa la inserción laboral, la inclusión social para fortalecer a sus filiales y asociados, promoviendo la representatividad del sector. (Discapacidades del Ecuador)

Se propone reorientar el sistema de gestión que viene desarrollando la Federación en beneficio directo de sus filiales; teniendo como referentes la Convención de las Naciones Unidas de los Derechos de las personas con Discapacidad, Constitución Política del país, Ley de Discapacidades, Ley 13-97 de Defensa del Ciego Ecuatoriano, su Estatuto, Reglamentos y más normativas vigentes (FENCE).

Tiene como paradigma el cumplimiento de derechos y la inclusión de las personas con discapacidad visual, privilegia los siguientes principios:

- Promoción de derechos
- Igualdad de oportunidades
- Integración y participación
- El respeto de la dignidad inherente
- La autonomía individual, incluida la libertad de tomar las propias decisiones
- La no discriminación
- La participación e inclusión plena y afectiva en la sociedad;
- El respeto por la diferencia y la aceptación de las personas con discapacidad, como parte de la diversidad y la condición humana;
- La accesibilidad
- La igualdad entre el hombre y la mujer.

Gracias al apoyo de las Instituciones públicas y privadas del país y organismos Internacionales, la Federación cada día se va consolidando con líderes capaces y comprometidos, que buscan el progreso y desarrollo de sus instituciones y asociados.

2.1.1 Situación de las personas no videntes

Uno de los mejores inventos que ha tenido el mundo fue el de Louis Braille, una auténtica revolución cultural para todas las personas no videntes, pues tuvieron una inclusión hacia la alfabetización, alcanzando cada vez más sus sueños de tener acceso a la educación completa, a la cultura, a llegar a conocer al mundo y sus alrededores con sus propios medios: sus dedos.

La interrogante más grande al momento de mencionar a las personas no videntes, es pues sin duda, su situación en la sociedad, en el día a día, al ser un sector minoritario y aún discriminado en el país y la sociedad, no tiene más opción que integrarse a las personas videntes. Por este motivo, las personas videntes tienen la obligación de aprender más de su mundo, conociendo las técnicas y materiales que las personas no videntes emplean para su desenvolvimiento cotidiano, no sólo por curiosidad, sino con el fin de comprenderlos mejor.

Es importante tratar a las personas no videntes con el respeto y la dignidad que merece todo ser humano y brindarles la oportunidad de demostrar todas sus

capacidades y su productividad en la sociedad, pues así tendrán una vida independiente y satisfactoria sintiéndose útiles dentro de la sociedad.

Su situación cada vez es más alentadora y gratificante, pues es más común la creación de escuelas y fundaciones que dan un apoyo tanto a las personas no videntes como a sus padres, beneficiándolos en una educación integral, ofreciendo un continuo proceso de formación y habilitación permitiéndoles alcanzar autosuficiencia e independencia facilitando la inserción en la vida laboral y social.

2.2 Sistema Braille

Desde la antigüedad hubo diversos inventos para facilitar la lectura y escritura a las personas no videntes tales como: letras de madera, en relieve, regletas y punzones, nudos de distinto grosor en una cuerda, etc., pero su utilización fue poco extendida por las deficiencias intrínsecas de estos códigos o los materiales empleados. La ceguera se convertía, por tanto, en un obstáculo para el acceso a la comunicación escrita.

En la segunda mitad del siglo XVIII empieza a cambiar la actitud de la sociedad ante la discapacidad en general y la ceguera en particular. Concretamente, en 1784, en París, Valentin Haüy funda el primer centro educativo para personas no videntes llamado Institution National des Jeunes Aveugles.

Es en este instituto donde se utilizaba para leer el sistema de Haüy que consistía en la grabación en alto relieve de los caracteres que se utilizan en vista. Con este sistema las personas no videntes podían leer, aunque muy despacio, pero no podían escribir.

En 1819, el capitán de artillería Charles Barbier de la Serre (1767-1861), presenta en el centro educativo de Haüy un sistema punti forme (escritura nocturna o sonografía) que inventó para que los soldados se enviaran mensajes en relieve, para poder ser leídos al tacto, en la oscuridad.

Eran signos que representaban sonidos que se leían con las yemas de los dedos y que se podían escribir con un punzón. Pero los signos resultaban demasiado grandes y no representaban la ortografía de las palabras, sino su sonido.

Louis Braille (1809-1852), estudiante ciego del instituto fundado por Haüy analiza el sistema de Barbier y realiza varias modificaciones: reduce el tamaño de los signos e inventa un alfabeto.

En 1827 se publica, finalmente, el código de lectoescritura para personas no videntes: el Sistema Braille, también adapta el sistema a las matemáticas, la música y las ciencias, y desarrolla un sistema de abreviaturas. Inventa también un punzón que permite la escritura. Como curiosidad, es interesante saber que el título de la presentación del Sistema Braille es exactamente: Procedimiento para la escritura de palabras, música y canto llano por medio de puntos para uso de los ciegos y arreglado por ellos. Intentaba, también, facilitar a las personas sin visión la lectura y escritura de partituras musicales (de hecho, Luis Braille era profesor de música).

El Sistema Braille no fue aceptado ni difundido fácilmente ya que suponía un cambio drástico con respecto a la tendencia anterior. Es en el año 1840 cuando se acepta oficialmente.

En 1878, en el Congreso Internacional celebrado en París, se decide promoverlo como método universal al considerarlo el mejor sistema de lectoescritura para personas con ceguera, por su probada utilidad didáctica.

Fernández del Campo (2004) sintetiza de esta forma la importancia del sistema Braille: ¿Fue consciente Louis Braille de la potencia de su sistema, del instrumento que ponía en manos de los ciegos? ... “las posibilidades de su invento superan las necesidades que venía a solucionar y, sin pérdida de coherencia, permite responder a un sinnúmero de retos no vislumbrados en el momento de la creación. Como tantas otras veces a lo largo de la historia, la obra, cual dotada de vida propia, honraba a su creador, rindiendo frutos inesperados.

La difusión del sistema Braille como método universal de comunicación escrita para personas ciegas ha sido un factor decisivo en favor de la integración social y educativa de las personas con discapacidad visual. Hoy en día, el acceso a la información de estas personas es una realidad gracias, al sistema Braille.

El sistema Braille sigue siendo, para las personas con ceguera, el mejor medio de acceso al mensaje escrito, al cálculo, la música, la literatura... En suma, a la cultura. (Educación Inclusiva)

2.2.1 Historia del Braille

Desde la antigüedad hubo diversos inventos para facilitar la lectura y escritura a las personas no videntes tales como: letras de madera, en relieve, regletas y punzones, nudos de distinto grosor en una cuerda, etc., pero su utilización fue poco extendida por las deficiencias intrínsecas de estos códigos o los materiales empleados. La ceguera se convertía, por tanto, en un obstáculo para el acceso a la comunicación escrita.

En la segunda mitad del siglo XVIII empieza a cambiar la actitud de la sociedad ante la discapacidad en general y la ceguera en particular. Concretamente, en 1784, en París, Valentin Haüy funda el primer centro educativo para personas no videntes llamado Institution National des Jeunes Aveugles.

Es en este instituto donde se utilizaba para leer el sistema de Haüy que consistía en la grabación en alto relieve de los caracteres que se utilizan en vista. Con este sistema las personas no videntes podían leer, aunque muy despacio, pero no podían escribir.

En 1819, el capitán de artillería Charles Barbier de la Serre (1767-1861), presenta en el centro educativo de Haüy un sistema punti forme (escritura nocturna o sonografía) que inventó para que los soldados se enviaran mensajes en relieve, para poder ser leídos al tacto, en la oscuridad.

Eran signos que representaban sonidos que se leían con las yemas de los dedos y que se podían escribir con un punzón. Pero los signos resultaban demasiado grandes y no representaban la ortografía de las palabras, sino su sonido.

Louis Braille (1809-1852), estudiante ciego del instituto fundado por Haüy analiza el sistema de Barbier y realiza varias modificaciones: reduce el tamaño de los signos e inventa un alfabeto.

En 1827 se publica, finalmente, el código de lectoescritura para personas no videntes: el Sistema Braille, también adapta el sistema a las matemáticas, la música y las ciencias, y desarrolla un sistema de abreviaturas. Inventa también un punzón que permite la escritura. Como curiosidad, es interesante saber que el título de la presentación del Sistema Braille es exactamente: Procedimiento para la escritura de palabras, música y canto llano por medio de puntos para uso de los ciegos y arreglado por ellos. Intentaba, también, facilitar a las personas sin visión la lectura y escritura de partituras musicales (de hecho, Luis Braille era profesor de música).

El Sistema Braille no fue aceptado ni difundido fácilmente ya que suponía un cambio drástico con respecto a la tendencia anterior. Es en el año 1840 cuando se acepta oficialmente.

En 1878, en el Congreso Internacional celebrado en París, se decide promoverlo como método universal al considerarlo el mejor sistema de lectoescritura para personas con ceguera, por su probada utilidad didáctica.

Fernández del Campo (2004) sintetiza de esta forma la importancia del sistema Braille: ¿Fue consciente Louis Braille de la potencia de su sistema, del instrumento que ponía en manos de los ciegos? ... “las posibilidades de su invento superan las necesidades que venía a solucionar y, sin pérdida de coherencia, permite responder a un sinnúmero de retos no vislumbrados en el momento de la creación.

Como tantas otras veces a lo largo de la historia, la obra, cual dotada de vida propia, honraba a su creador, rindiendo frutos inesperados.

La difusión del sistema Braille como método universal de comunicación escrita para personas ciegas ha sido un factor decisivo en favor de la integración social y educativa de las personas con discapacidad visual. Hoy en día, el acceso a la información de estas personas es una realidad gracias, al sistema Braille.

El sistema Braille sigue siendo, para las personas con ceguera, el mejor medio de acceso al mensaje escrito, al cálculo, la música, la literatura... En suma, a la cultura. (Educación Inclusiva)

2.2.2 Sistema Braille

El Sistema Braille es un código de lectura, el mismo que está diseñado y orientado a personas no videntes para que puedan leer a través del tacto. Fue inventado en el siglo XIX por Louis Braille.

Está basado principalmente en un símbolo formado por 6 puntos: aquellos que estén en relieve representarán una letra o signo de la escritura en caracteres visuales. Se forma en base a una cuadrícula de seis puntos dispuestos de forma vertical, en dos columnas de tres puntos cada una.

Los puntos se los debe enumerar hacia abajo, de esta manera quedan los puntos 1, 2 y 3 de la primera columna, y los puntos 4, 5 y 6 son los puntos de la segunda columna.

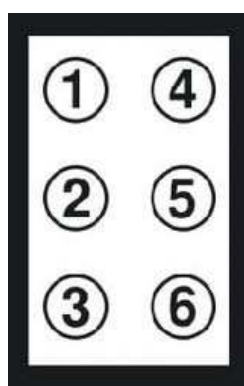


Figura 3 Signo generador del Sistema Braille

El Sistema Braille no es un idioma, es un código, por esta razón se debe tomar en cuenta que las particularidades y la sintaxis siempre serán las mismas que para los caracteres visuales, de la misma manera se considera la distribución y el tamaño en el que deben estar ubicados el Signo Generador (código Braille), ya que tienen una distribución estándar, esto se debe al fruto de la gran experiencia de Louis Braille, ya que llegó a la investigación exhaustiva de que las terminaciones nerviosas de la yema de los dedos están capacitadas para captar este tamaño en particular.

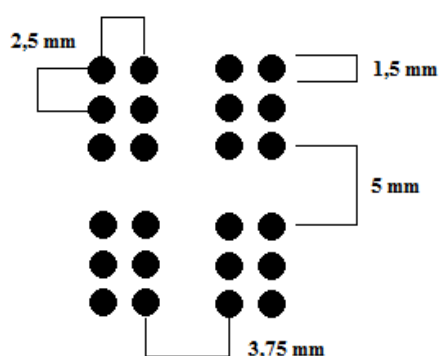


Figura 4 Distribución y Tamaño del Signo Generador.

Las personas no videntes realizan la lectura mediante el tacto con las yemas de los dedos, debe desplazarse de izquierda a derecha, teniendo como resultado un literal de la lengua en la que está leyendo, a diferencia de lo que se leería en la lengua de signos ya que éste posee su propia gramática y estructura, tratándose así de un idioma, como se dijo anteriormente, el Braille es una forma de codificación de una lengua que ya existe.

2.2.3 Introducción al Alfabeto Braille

A raíz de la idea de Louis Braille, muchas personas no videntes tienen a la mano esta herramienta poderosa que les permite la lectura y escritura apoyados en un sistema poderoso, válido y eficaz de comunicación, representando las letras del alfabeto común, los números, signos de puntuación, símbolos matemáticos, música, entre otros.

Como se explico anteriormente, el alfabeto está basado en una celda de seis puntos en relieve ubicados estratégicamente en una matriz de tres filas por dos columnas, enumeradas de arriba hacia abajo y de izquierda a derecha.

Hay que tener en cuenta que la presencia o ausencia de puntos permite la codificación de los símbolos, de esta manera se sabrá que letra es la que se está representando, con los seis puntos se puede obtener 64 combinaciones diferentes.

Las 64 combinaciones son insuficientes para poder cubrir con todo el alfabeto y todos los signos que existen, es por eso que se utilizan signos diferenciadores especiales utilizados como prefijos para representar caracteres como: letra en mayúscula, números, signos de puntuación, símbolos matemáticos, caracteres especiales, etc. Por esta razón se habla de sistema y no de alfabeto, ya que en base a estas combinaciones se han ido desarrollando distintos códigos que sirven para la representación de la música, matemática, ciencias, etc.

- **Alfabeto Braille**

El alfabeto Braille está compuesto y estructurado sobre seis puntos en relieve que permite a las personas no videntes leer mediante el sentido del tacto, representa letras, números, notaciones musicales, etc.

Alfabeto y Números

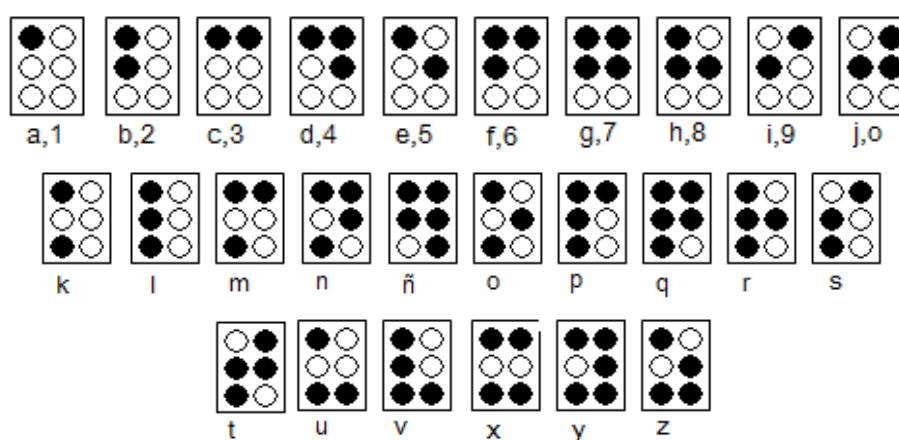


Figura 5 Alfabeto Braille.

Vocales con Acento

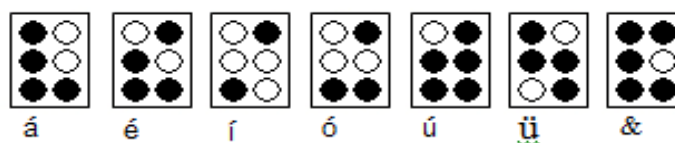


Figura 6 Vocales con Acento.

Signos Especiales

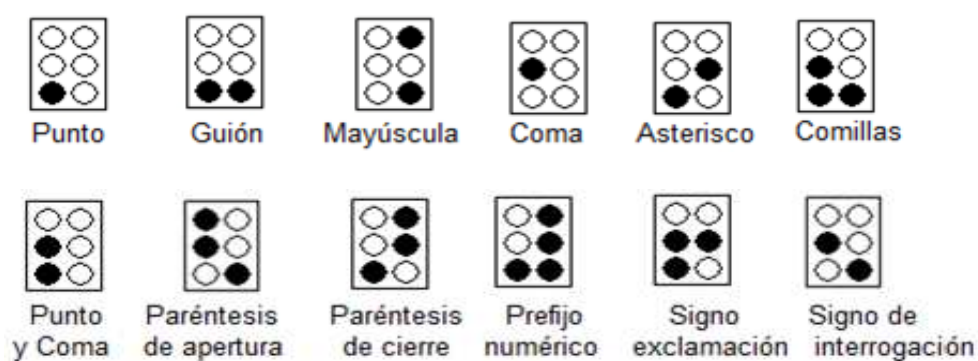


Figura 7 Signos Especiales.

El diseño del código Braille está creado con un diseño de manera lógica, empleando series, que de forma ordenada va generando el código Braille.

Serie 1: se utiliza los cuatro puntos superiores, es decir, los puntos 1, 2, 4, y 5, formándose las diez primeras letras del alfabeto.

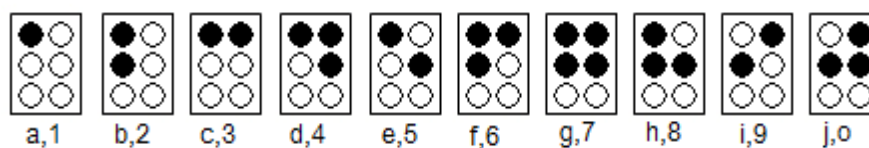


Figura 8 Letras del Alfabeto Braille de la Serie 1.

Serie 2: se añade el punto 3 a los puntos ya formados de la serie 1. Se obtienen las letras desde la K hasta la letra T, la letra Ñ es una excepción.

Luis Braille fue francés. Existe la letra Ñ en el Alfabeto Braille en español.

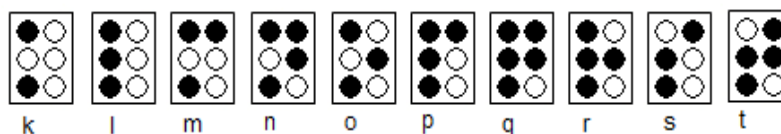


Figura 9 Letras del Alfabeto Braille de la Serie 2.

Serie 3: se forma a partir de los puntos de la Serie 2, añadiéndole el punto 6.

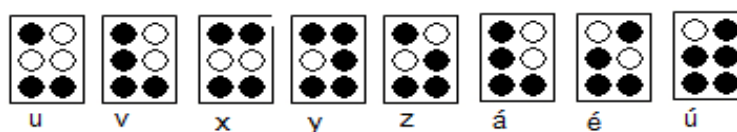


Figura 10 Letras del Alfabeto Braille de la Serie 3.

Serie 4: las letras se forman a partir de la serie 1 y añadiéndole el punto 6. Esta serie se caracteriza por la formación de los signos es francés, para el estudio del caso, se representa solo las letras del alfabeto en español.

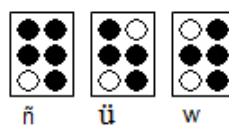


Figura 11 Letras del Alfabeto Braille de la Serie 4.

Serie 5: ésta serie define los puntos de puntuación, se utiliza los puntos de la serie 1 omitiendo los puntos de la mitad superior de la matriz.

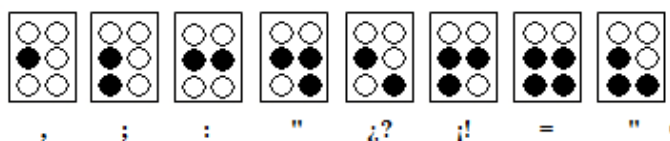


Figura 12 Letras del Alfabeto Braille de la Serie 5

Vocales con tilde:

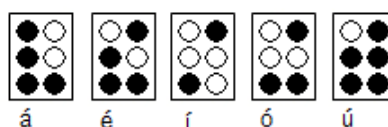


Figura 13 Vocales con Signo.

Otros Signos

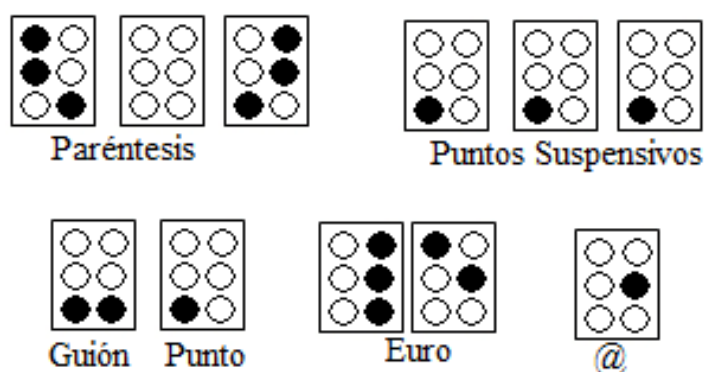


Figura 14 Signos de Interés.

Signos de Letras Mayúsculas y Números

Como ya se explicó anteriormente, con el signo generador se obtienen 64 combinaciones básicas en el Alfabeto Braille, que son insuficientes para representar todo el alfabeto, signos y grafemas necesarios, es por esto que, se debe utilizar signos complementarios usados como prefijos que ayudan a formar las letras mayúsculas, números o si es el caso una nota musical.

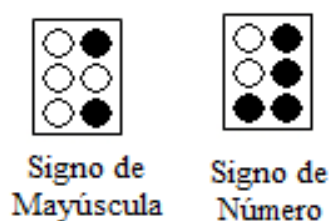
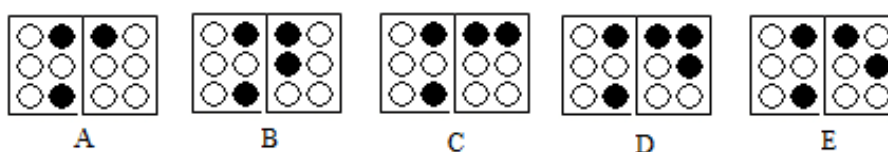


Figura 15 Signos de letras mayúsculas y números.

El Signo de Mayúscula se forma por la combinación de los puntos 4 y 6, se le debe anteponer esta combinación a cualquier letra que se desee esté en mayúscula



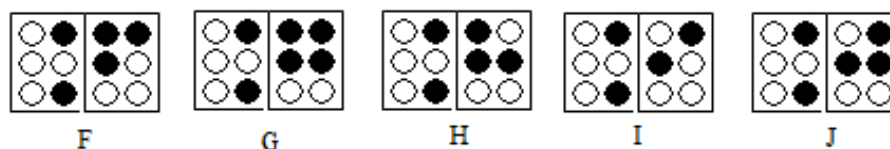


Figura 16 Ejemplos de letras en mayúsculas.

El signo número se forma con la combinación de los puntos 3, 4, 5 y 6. Ésta combinación se le antepone a las letras formadas en la Serie 1, de ésta manera se obtienen los números del 1 al 10.

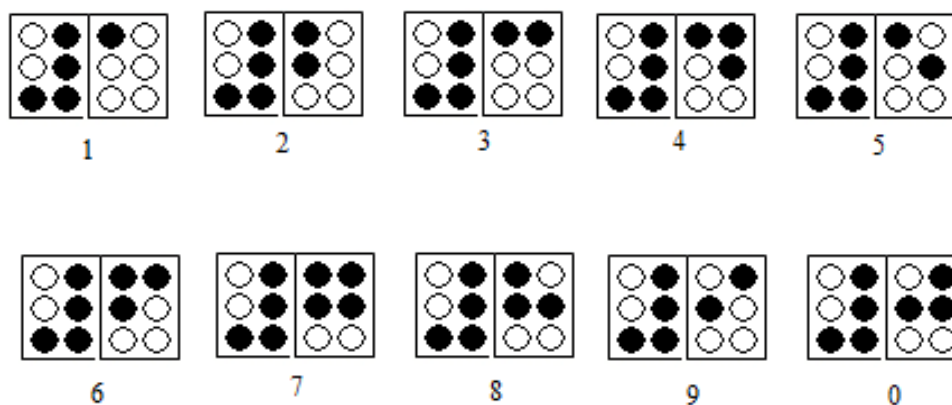


Figura 17 Representación de los Números.

Para representar números enteros de dos o más cifras, se debe anteponer el signo de número a cualquier combinación de número.

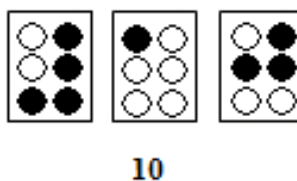


Figura 18 Representación de números enteros de más de dos cifras

Para representar números decimales, se debe colocar el signo de número delante de la primera cifra, separando por la coma, la misma que se representa con el punto 2, si la representación numérica es alta, se debe utilizar el punto 3 para separar las unidades.

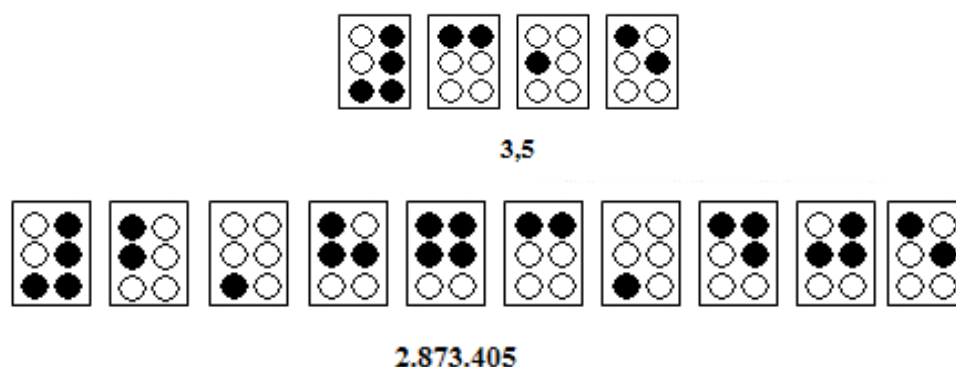


Figura 19 Representación de números decimales

Representación de Signos Matemáticos

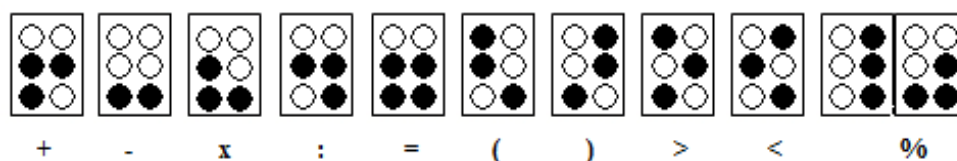


Figura 20 Representación de Signos Matemáticos

2.2.4 Escritura y Lectura Braille

Escritura Braille

Las personas no videntes consideran que la escritura es más rápida que la lectura Braille, debido a que presenta menor dificultad. Existen dos formas de escribir Braille, la una es escribiendo a mano y la otra escribiendo a máquina.

Escritura a mano: en la escritura a mano, se necesita obligatoriamente de una punzón o regleta, un punzón y papel especial (papel grueso). Se debe tener en cuenta ciertas consideraciones para poder escribir como:

- Se escribe de derecha a izquierda, invirtiendo la numeración de los puntos del signo generador, así los puntos marcados al escribir quedaran como puntos en relieve al momento de dar la vuelta al papel y de ésta manera poder leer.
- Se necesita alta precisión mecánica en el punteado y practica.
- Todos los puntos deben tener el mismo relieve.

Escritura a máquina: esta escritura es más fácil, las características básicas de este máquina es tener 6 teclas, una para cada uno de los puntos del signo generador de Braille. También tiene un espaciador, una tecla para el retroceso y otra para el cambio de línea.

La máquina de escribir Braille más común es la Perkins, ésta fue fabricada por la Perkins School of the Blinds en Massachusetts, USA. Las 6 teclas corresponden a cada punto del signo generador, es por esto que se pueden pulsar las teclas cada una por separado o simultáneamente, construyendo la combinación de los códigos Braille, tomando en cuenta que a cada tecla le corresponde un dedo en específico para ser pulsada de esta manera las manos están en una posición cómoda.

- **Lectura Braille**

Bien es cierto que se necesita de un gran aprendizaje y comprensión del sistema, ya que se usa un código diferente al alfabeto en tinta, en comparación con la escritura Braille no presenta mucha dificultad, y se lo realiza en el mismo sentido que se realiza la escritura en tinta, es decir de izquierda a derecha.

En la lectura a tinta se realiza mediante el reconocimiento de las palabras completas, en cambio en la lectura Braille se debe utiliza el tacto de izquierda a derecha, para poder reconocer letra por letra y así ir leyendo el texto en Braille planteado.

A la lectura se le clasifica en dos fases:

- Fase 1:

lectura unimanual: Los dedos índices son usadas como lectores, deben ir juntos, de esta manera inician la lectura en cada línea, cuando llegan al final de la misma se retrocede sobre ella. Cuando se realiza este retroceso y se llega a la mitad, se desciende a la siguiente línea, al termino del retroceso se llega al principio de la línea, empezando la lectura nuevamente.



Figura 21 Lectura unimanual.

- Fase 2:

Lectura bimanual: Esta fase realiza el movimiento de las manos doble debido a que cada mano lee aproximadamente la mitad del renglón. Se empieza leyendo la primera línea con los dedos índices de cada mano pero unidos, cuando se llega a la mitad de la línea se deben separar. De esta manera la mano derecha termina de leer el renglón, mientras tanto la mano izquierda debe ir a la siguiente línea descendiendo y retrocediendo al principio de esta línea y así continuar con la lectura.



Figura 22 Lectura bimanual.

2.2.5 Ventajas y desventajas

El sistema Braille presenta siempre sus preocupaciones para aquellas personas que lo usan y quieren involucrarse en él, pues siempre se siente una necesidad de promocionar su uso en la lectura y escritura Braille, tanto por parte de los usuarios

potenciales como por parte de las instituciones, entidades y empresas con actividad pública y privada, para comunicarse con las personas no videntes o hacer accesibles sus servicios y productos, logrando así tener una buena comunicación entre videntes y no videntes, rompiendo barreras que antes eran difíciles de superar.

Al realizar una comparación constructiva entre la lectura Braille y los sistemas que permiten acceder a libros mediante la voz, el sistema Braille ayuda a la comprensión de la lectura, pues la concentración del que lee es mayor y por lo tanto retiene mas información que cuando escucha, fomentando el aprendizaje.

La universalidad del sistema Braille es muy importante, pues todos los símbolos y letras son las mismas en cualquier parte de mundo, además que su relativa sencillez de uso cuando se domina ayuda al temprano aprendizaje y les brinda a las personas no videntes a llevar una autonomía personal, muy indispensable para desenvolverse en la sociedad con la seguridad plena de que son incluidos en su medio social o económico.

Una dificultad muy grande se encuentra al momento de aprender ciertas signografías especiales para materias específicas como matemáticas, física, química, música, informática e incluso para los diferentes idiomas, pues el código original esta realizado en francés y fue necesario sustituir algunos signos y en otros casos se ha tenido que crear nuevos signos como la letra “ñ” en el idioma español, además de que su complejidad radica en el aprendizaje del código, pues cada uno de estos signos tiene su significado dependiendo del contexto de su uso, por ejemplo, el signo de admiración, representa también, en el contexto de las matemáticas, el signo de sumar.

2.2.6 Impresión Braille

La impresión de texto en formato Braille, desde hace muchos años se la ha realizado mediante las máquinas de escribir Braille, que son netamente mecánicas, con el avance de la tecnología, estas máquinas han ido evolucionando hasta llegar a tener impresoras Braille, que ya son dispositivos electrónicos que permiten imprimir textos e imágenes simples en este formato.

- **Máquina de escribir Braille**

La máquina de escribir Braille es una máquina estándar consta de las siguientes 9 teclas:

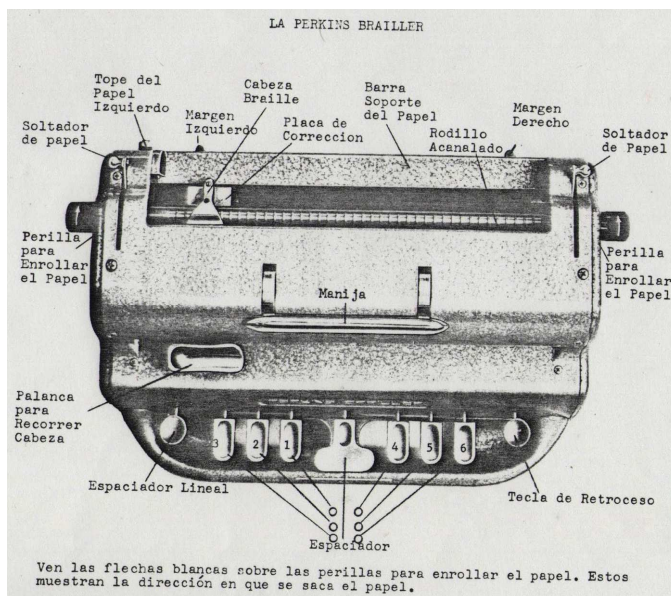


Figura 23 Máquina Braille con sus respectivas partes. (Tiflogía)

- Tecla de espacio.
- Tecla para retroceder un espacio.
- Tecla para cambiar de línea.
- Seis teclas una por cada punto Braille.
- Un timbre que indica cuando se aproxima al final del margen derecho.

Para escribir cualquier letra en el código Braille, se debe presionar las teclas correspondientes a cada combinación de forma simultánea, de esta manera si por ejemplo se desea teclear la letra “g” se debe presionar simultáneamente las teclas 1,2,4,5, así se generará la letra “g”.

Para separar las palabras se debe presionar la tecla de espacio, y si hubo un error, con la tecla de retroceso se puede corregir éste error.

El dominio de la máquina de escribir se logra mediante la práctica, adquiriendo una velocidad aceptable y sobretodo obteniendo el conocimiento perfecto de la máquina, llegando al dominio de cada tecla, tomando en cuenta que:

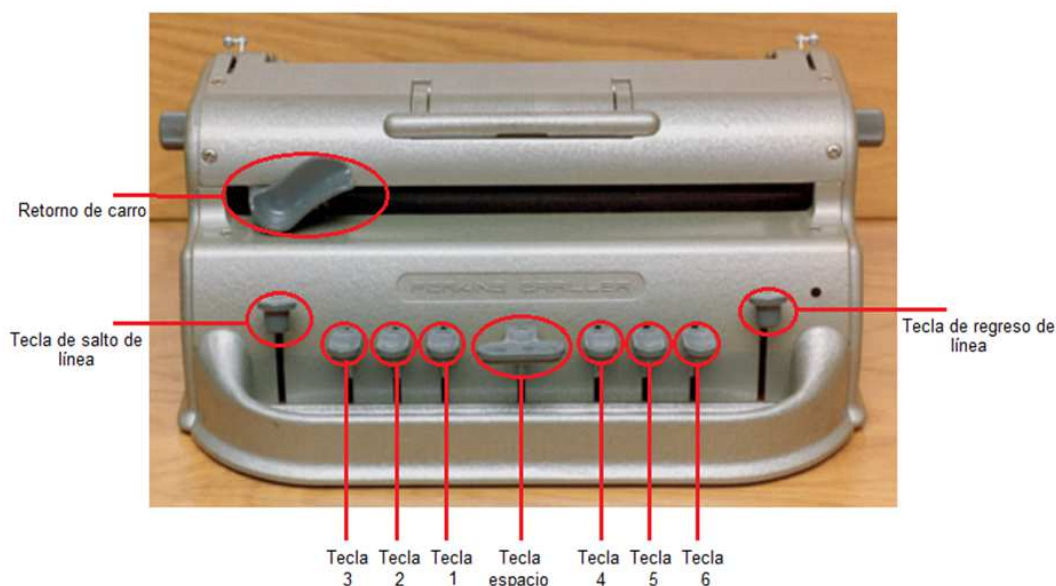


Figura 24 Posición de las teclas según los puntos del signo generador.

- La tecla 1 con el índice izquierdo.
- La tecla 2 con el corazón izquierdo.
- La tecla 3 con el anular izquierdo.
- La tecla 4 con el índice derecho.
- La tecla 5 con el corazón derecho.
- La tecla 6 con el anular derecho.
- La tecla espaciadora con el pulgar.

- **Impresora braille**

Son periféricos que, conectados a un dispositivo informático que les envíe texto, imprimen en código Braille sobre soporte de papel, plástico, u otras superficies. A pesar de que son impresoras preparadas para el sistema Braille, por lo que realizan semi perforaciones con matrices de 6 u 8 puntos y separaciones entre líneas y caracteres, algunos modelos son capaces de producir gráficos en forma de imágenes

en relieve, bien usando esta matriz con sus separaciones, bien logrando realizar líneas continuas de puntos con los que conforma gráficos. (Sistema Braille)

Una impresora Braille al igual que una impresora convencional de tinta, permite imprimir en código Braille cualquier documento editado en el computador, su aspecto parecido a una impresora convencional, así como la funcionalidad de sus partes tales como las bandejas de entrada y salida, lo único que difiere con respecto a las “impresoras normales” es su tamaño, pues las impresoras Braille deben ser más grandes por el tipo y dimensiones de papel que usan.

El mecanismo que realizan estas impresoras difieren entre las marcas y modelos, por ejemplo algunas permiten imprimir o marcar los puntos Braille en ambas caras del papel, y se puede usar un papel continuo o cortado.

Algo muy novedoso al usar una impresora como estas es que se puede realizar dibujos simples utilizando los caracteres del código Braille, de manera que las personas no videntes pueden sentir tocando el contorno del dibujo.

El funcionamiento de una impresora Braille consiste en convertir rápidamente todo texto a código Braille, incluyendo algunas tablas y sencillos gráficos, los mismos que una vez impresos hacen accesible mediante la lectoescritura a la información hacia las personas no videntes

Todas las impresoras Braille poseen un software que permite hacer la conversión o transcripción al código Braille permitiendo la edición de textos en todo formato como PDF, documentos de office, etc. siendo compatible con sistemas operativos como Windows 2000, XP, vista y Windows 7.

Las características del software permiten la transcripción fiel al código Braille de funciones matemáticas, además de que algunas impresoras permiten la interacción del usuario en red ajustando sus necesidades.

La manera a la cual estas impresoras Braille se conectan hacia el dispositivo que contiene el texto que va a ser impreso, es de la misma manera que la hacen las impresoras convencionales, esto depende claro, del modelo de cada impresora pero

usan conexiones tales como: la línea serie y/o el cable paralelo, además pueden usar conexiones USB, bluetooth, cable de red, etc.



Figura 25 Impresora Braille. (BAC)

2.3 Recursos de Hardware

2.3.1 Motores eléctricos

Con los grandes avances de la tecnología y la ciencia, hoy en día el uso de motores eléctricos es muy importante ya que son la base del progreso y el sustento primordial de las máquinas que se usan hoy en día; tanto en la vida diaria como en el trabajo. Por esta razón es necesario conocer su construcción, funcionamiento y las aplicaciones.

El motor eléctrico es una máquina eléctrica capaz de transformar la energía eléctrica (energía que hace funcionar al sistema) en energía mecánica (energía que realiza el trabajo), y su funcionamiento se centra en usar inducción electromagnética que produce la electricidad para producir movimiento, es por esto que los motores de inducción son los más usados, pues aparte de utilizar la energía eléctrica son limpios y de fácil transportación, son simples.

Los motores eléctricos tiene característica generales muy específicas como:

- Rendimiento: se representa por la letra griega η . Es la razón entre la potencia útil que generan y la potencia absorbida.

- Velocidad nominal: se representa por la letra n . Se la conoce también como velocidad de poco giro y es la velocidad angular del eje, es decir, el número de revoluciones por minuto (RPM) a las que gira.
- Potencia: se mide en (CV=736 W) o caballos de vapor. Es el trabajo que el motor realiza en la unidad de tiempo a velocidad de giro específica.
- Par motor: se mide en $\text{kg}\cdot\text{m}$ (kilogramos por metro) o en Nm (newtons-metro). Es el momento de rotación que actúa sobre el eje del motor y determina su giro.
- Estabilidad: es cuando el motor no sufre ningún defecto al momento de girar altas velocidades sin usar demasiada energía eléctrica en cierto intervalo de tiempo.

A continuación se muestra una clasificación de los motores eléctricos tanto AC como DC.



Figura 26 Clasificación de los Motores DC

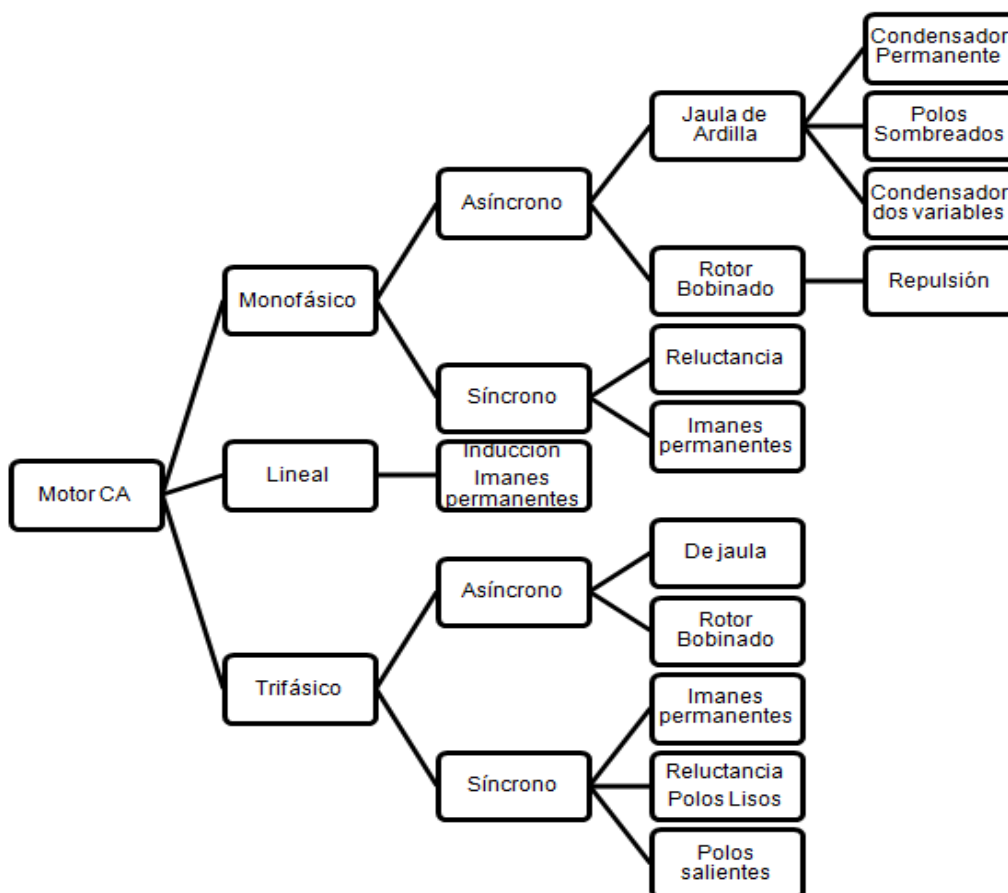


Figura 27 Clasificación de los Motores AC

- **Motor DC**

Los motores de corriente continua (DC) necesitan alimentarse con corriente continua, y por este motivo deben obligadamente contar con un dispositivo capaz de convertir la corriente alterna en corriente continua.



Figura 28 Motor DC

El funcionamiento se caracteriza por una velocidad variable que depende del tipo de control que lo maneje ajustándola según sean las necesidades, por esta misma razón son flexibles y precisos. Sus aplicaciones se restringen debido al costo, por eso se los usa en casos específicos en donde el gasto sea equivalente a la aplicación, instalación y mantenimiento.

Principio de funcionamiento y control

El funcionamiento de los motores se basa en la aplicación de dos principios: el de inducción (Michael Faraday – 1831) que dice que si un conductor se mueve a través de un campo magnético o está situado en las proximidades de otro conductor por el que circula una corriente de intensidad variable, se induce una corriente eléctrica en el primer conductor. Y el segundo principio (André Ampère – 1820), en el que establece: que si una corriente pasa a través de un conductor situado en el interior de un campo magnético, éste ejerce una fuerza mecánica o f.e.m. (fuerza electromotriz), sobre el conductor. (Cuervo)

Es decir, que a través de la fuerza de atracción y repulsión que existen entre polos, en electromagnetismo se conoce la existencia del polo norte y polo sur, estas dos regiones son las que reúnen las líneas de fuerza de un imán. Por esta razón un motor se forma con polos alternados entre el estator y el rotor, tomando en cuenta que los polos magnéticos iguales se repelen, y polos magnéticos diferentes se atraen, con esta ley se produce el movimiento de rotación en un motor eléctrico.

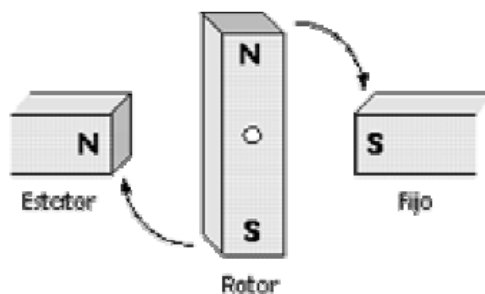


Figura 29 Polos magnéticos que generan el movimiento de un motor

Para el control de un motor se debe tomar en cuenta que es lo que se quiere controlar, ya sea la velocidad o la dirección del giro del motor.

Para controlar el giro o dirección de los motores se puede usar una configuración de transistores o Puente H que es un circuito electrónico que permite que el motor DC gire para ambos sentidos (avance o retroceso).

Su nombre proviene debido a la representación en la grafica del circuito, pues se construye con cuatro interruptores (mecánicos o mediante transistores).

Cuando los interruptores Q2 y Q5 están cerrados y Q3 y Q4 abiertos, se aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores Q2 y Q5 y cerrando Q3 y Q4, el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

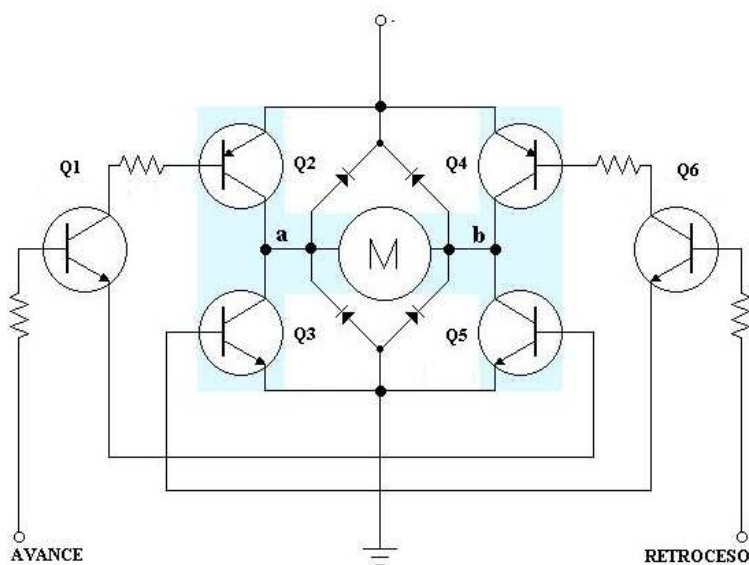


Figura 30 Circuito del Puente H para controlar el motor

Se presenta más dificultad el momento de controlar la velocidad, pues es difícil alimentarlos con voltajes variables, debido a esto se recurre a la electrónica digital, en donde se puede realizar un PWM.

Esta modulación se basa en la aplicación de un tren de pulsos de período fijo e ir variando el ancho del pulso, de esta manera el motor recibe un voltaje efectivo menor lo que hace que el motor gire a menor velocidad.

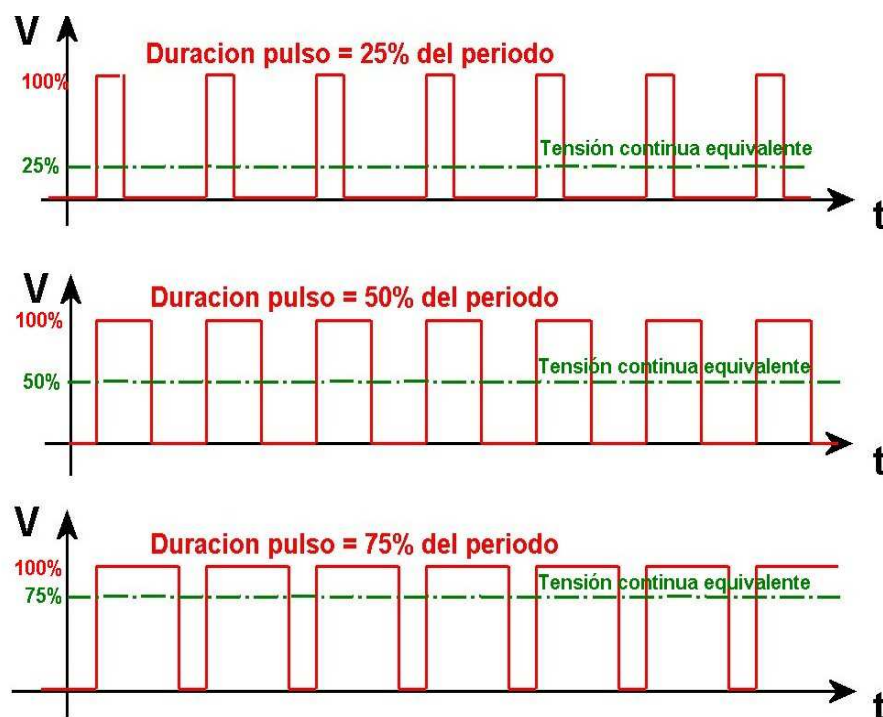


Figura 31 Modulación PWM para controlar la velocidad del motor.

- **Servomotor**

Se caracteriza por tener un eje de rendimiento controlado que puede tener posiciones angulares específicas, según sea la señal codificada que se le envíe o el tipo de control que se le este aplicando, siempre que exista la señal en la entrada el servomotor su posición angular del engranaje permanecerá hasta que ingrese otra señal que cambiara la posición angular de los piñones.

El funcionamiento del servomotor tiene mucha relación con la estructura, pues presenta circuitos de control y un potenciómetro conectado al eje central del servomotor que ayuda al circuito de control calibrar el movimiento angular deseado, de este modo se puede saber si el ángulo es el correcto ya que el motor estará apagado, en cambio si el circuito de control detecta que el ángulo no es el correcto, el motor empezara a girar hasta llegar al ángulo correcto.

Los servomotores se componen por:

- Motor
- Potenciómetro
- Circuito de Control
- Carcasa del Servomotor
- Ranura de salida
- Tren de Engranés



Figura 32 Estructura interna de un servomotor.

- Motor de corriente continua (DC): el motor gira en un sentido a velocidad máxima, y si se invierte la polaridad de alimentación del motor, cambia el sentido de giro del motor.
- Engranajes reductores: se encuentra un tren de engranes que cumple con la función primordial de reducir la alta velocidad de giro del motor principal para así amplificar la capacidad de torque
- Sensor de desplazamiento: se ubica en el eje de salida del servomotor que sirve para conocer la posición angular del motor.
- Circuito de control: es una placa electrónica en la cual se encuentra el control de la posición por retroalimentación, es decir, compara la señal de entrada de referencia o posición deseada con la posición actual medida por el potenciómetro.
- La diferencia que existe entre la posición actual y la posición deseada es amplificada y ésta se utiliza para mover el motor en la dirección necesaria.

Principio de funcionamiento del servomotor

El servomotor tiene tres cables, dos cables de alimentación que son positivo y negativo y un cable de datos que envía señales respectivas al motor para indicar la posición deseada de control mediante señales con Modulación de Ancho de Pulsos o sus siglas en ingles (PWM).

Las señales de control se envían con la ayuda del PWM, éstas señales son pulsos positivos de duración proporcional a la posición deseada y se repiten cada 50 Hz o cada 20 ms, cuando trabajan en un rango de 90° los pulsos PWM pueden estar comprendidos en un rango entre 0.9 a 2.1 ms, aunque también existen servos que trabajan en un rango de 180° y los pulsos PWM van de 0.5 a 2.5 ms.

En la siguiente figura se puede observar los diferentes anchos de pulso que envía el PWM en un periodo de 20 ms tomando en cuenta un pulso mínimo (0°), un pulso en la posición neutral (90°) y un pulso máximo (180°)

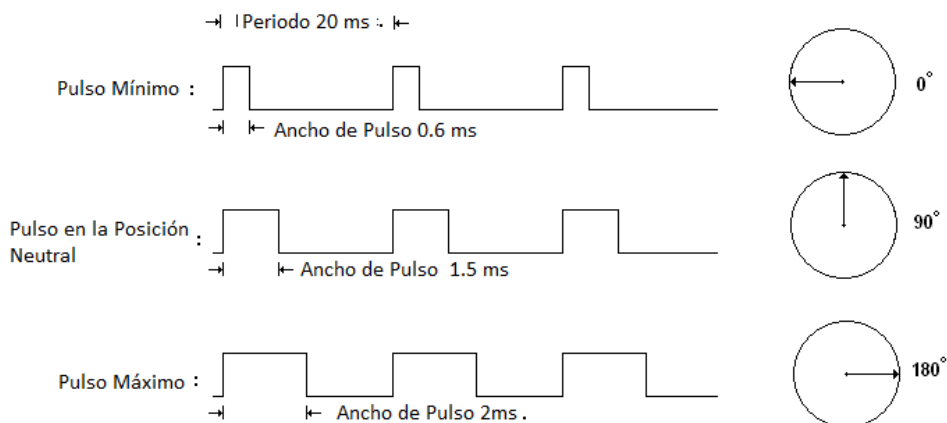


Figura 33 Diferentes anchos de pulso de un PWM.

- **Motor a pasos**

A diferencia de los motores convencionales que giran en forma libre al alimentarle a un cierto valor de voltaje, el motor paso a paso transforma o convierte los impulsos eléctricos en movimientos de giro fijos o desplazamientos angulares discretos y precisos que se incrementan, es decir, se mueve mediante una serie de

pasos o grados por lo que pueden ser controlados entre valores que van desde $1,80^\circ$ hasta 90° .

Por ejemplo, para dar una vuelta entera de 360° con la configuración de 90° se necesitaran 4 pasos, y si se lo hace con la configuración de $1,8^\circ$ será necesario dar 200 pasos. Sus aplicaciones se basan en la precisión y la repetitividad que tienen al momento de posicionar un giro o desplazamiento angular.

Existen tres tipos de motores paso a paso:

- Motor de reluctancia variable: se los usa para aplicaciones en las que se necesitan pasos muy pequeños y se compone de un rotor en forma dentada de hierro dulce o de material magnético y se alinea con los polos bobinados del estator minimizando la reluctancia rotor-estator, es decir, que el espacio entre los polos del estator permanece ocupado por el rotor.
- Motor de magnetización permanente: sus siglas en inglés Permanent Magnet Synchronous Motor (PMSM). Tienen en su estructura un imán permanente de forma cilíndrica conocidos como motores de flujo radial. Presentan un bajo momento de inercia.
- Motor paso a paso híbrido: el funcionamiento de este motor se basa en la unión del funcionamiento de los otros dos tipos de motores paso a paso, el rotor se compone de anillos de material ferro magnético en forma dentada que se encuentran en un imán permanente en forma axial y tiene un número de dientes distinto al del estator.



Figura 34 Motores paso a paso.

Secuencias para el movimiento de un motor paso a paso

Existen dos secuencias básicas para mover un motor paso a paso, estas dependerán del número de bobinas que se activen, según la configuración que se le programa desde el microcontrolador.

La primera se trata de la activación de una sola bobina por separado, con esta secuencia se tiene una fuerza muy baja, pues la única bobina activada es la que sujeta al eje del motor





Paso	A B C D	Motor
1	1 0 0 0	
2	0 1 0 0	
3	0 0 1 0	
4	0 0 0 1	

Figura 35 Secuencia de movimiento de motor paso a paso.

En la segunda secuencia se activaran dos bobinas a la vez, con esto el campo magnético de los motores tiene más potencia y los pasos serán más violentos en comparación con la secuencia anterior.





Paso	A B C D	Motor
1	1 1 0 0	
2	0 1 1 0	
3	0 0 1 1	
4	1 0 0 1	

Figura 36 Secuencia del motor paso a paso.

2.3.2 Actuador Lineal

Los tipos de los actuadores son: electrónicos, hidráulicos, neumáticos y eléctricos, dependen del tipo de energía o manera en la que se activan y de la fuerza que actúa sobre ellos como son la fuerza motriz en eléctricos ya sea un motor eléctrico o una solenoide, la presión hidráulica o la presión neumática y son los más usados en la electrónica, mecatrónica, robótica, etc. tomando siempre en cuenta el uso y la aplicación a la que se le está diseñando.

Si se necesita una aplicación en donde lo mas primordial es la posición, se debe usar actuadores neumáticos, en cambio, si se necesita una potencia alta se debe usar actuadores hidráulicos que cumplan con las especificaciones de diseño, tomando en cuenta que estos sistemas requieren un equipo que suministre de energía ya sea neumática o hidráulica y no olvidarse del continuo mantenimiento que necesitan por esta razón, estos actuadores son más costosos que los actuadores electrónicos.



Figura 37 Actuadores eléctrico, neumático e hidráulico.

- **Solenoide**

Un solenoide no es más que un electroimán formado por un conductor eléctrico en casi todos los casos un alambre de cobre esmaltado que se encuentra enrollado en forma helicoidal, por este alambre se conduce una corriente eléctrica que produce un campo magnético.

Cuando se introduce un núcleo de hierro dulce o acero dentro de este campo magnético, aparecerá sobre él una fuerza que tratará de introducirlo en el solenoide,

cuando coincida la posición de equilibrio se alcanzará cuando coincidan el centro del solenoide y el del embolo.

Esta fuerza será proporcional a la corriente que circula por el arrollamiento, es decir a mayor corriente mayor fuerza, por lo que si ponemos un resorte antagónico que se oponga a esta fuerza obtendremos un sistema de fuerzas que llegará al equilibrio cuando la fuerza ejercida por el resorte sea igual a la ejercida por el campo magnético producido por el solenoide

Si dentro de este campo magnético, se coloca un núcleo móvil de acero, se obtendrá una fuerza, la misma que de alguna manera tratará de introducirlo en el solenoide.

Existirá una posición de equilibrio que será cuando coincida el centro del solenoide con el centro del núcleo móvil. Cuando llega al estado de equilibrio los dos materiales, la fuerza será proporcional a la corriente que circula por el devanado, por lo tanto, a mayor corriente mayor fuerza.

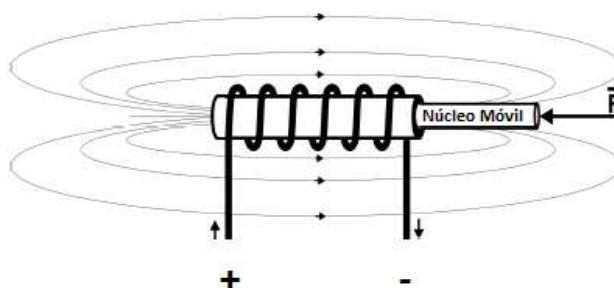


Figura 38 Solenoide de núcleo móvil

Este núcleo móvil puede ser de dos tipos, luego debe determinarse la naturaleza de la aplicación que definirán las características de que difieren por la aplicación que tiene cada uno.

El primer tipo se lo conoce como arrate, es decir el núcleo móvil se retrae y se lo usa para aplicaciones como seguridad de puertas, el segundo tipo es el de empuje en donde el núcleo móvil lo que hace es salir con fuerza hasta cierta distancia y

comúnmente se los usa en el motor de arranque de los carros, entre otras aplicaciones.



Figura 39 Solenoides de empuje y arrastre.

- **Actuador Lineal Eléctrico**

Cuando se habla de un actuador lineal eléctrico se habla de un dispositivo que tiene la capacidad de convertir el movimiento de rotación del motor interno que generalmente es de corriente continua en movimiento lineal, la mayoría de los actuadores lineales usan el mecanismo de piñón y cremallera, que permite realizar este movimiento giratorio en lineal continuo, son precisos ya que el piñón que es la rueda dentada engrana perfectamente en la cremallera como se muestra en la figura , cuando el piñón gira sus dientes encajados empujan a los dientes de la cremallera convirtiendo el movimiento rotatorio en lineal, es decir que un actuador lineal se compone de un motor de corriente continua, un engranaje y un eje roscado, incluyendo una tuerca.

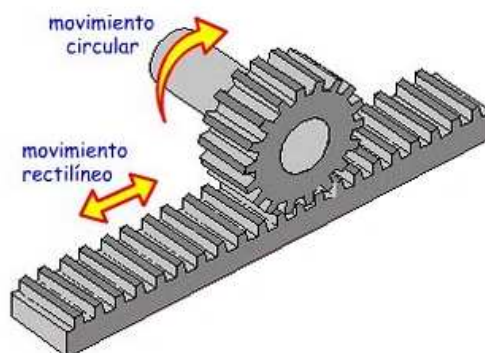


Figura 40 Piñón y cremallera

La ventaja principal de este tipo de actuadores es la facilidad de alimentación siempre y cuando se haga una simple comparación con los actuadores neumáticos o hidráulicos que necesitan una fuente de alimentación más compleja, es de suma importancia recalcar que los cables eléctricos que contienen en su estructura física sirven tanto para transferir los datos o señales de control como también transferir la electricidad que lo alimentara, y como su señal se transmite mediante cables no hay limitaciones en cuanto a la distancia entre el actuador y la fuente de poder, esta característica no se presenta en los sistemas neumáticos o hidráulicos en donde se limita la distancia entre el actuador y la fuente de poder debido a la naturaleza misma de su alimentación.

Aplicaciones de los actuadores lineales

Este actuador lineal eléctrico ofrece una respuesta de funcionamiento rápida, incluso si se trata de levantar pesos, lo hace a una velocidad moderadamente rápida, las aplicaciones para estos actuador es muy extensa pues se los usa para abrir y cerrar puertas eléctricas, se los usa en las camillas eléctricas de hospitales, en utilizan en cubiertas de compartimiento, cama de tracción quirúrgica, cama de masaje, sofá, silla de masaje, equipos eléctricos, o cualquier otro trabajo pesado, para levantar cargas pesadas tales como troncos o en la industria o se los usa para aplicaciones personalizadas que necesitan delicadeza y precisión, con un control más fino de movimiento.

Todas estas aplicaciones radican en la composición física del actuador, pues se forma de un motor eléctrico de alto torque capaz de levantar o empujar un peso alto de masa a una fuerza considerable, el accionamiento es durable garantizando una vida larga de excelente rendimiento y lo más importante, presentan todos sus ejes lubricados optimizando un funcionamiento fiable, la estructura se encuentra totalmente sellada lo que permite una protección máxima de agentes externos que puedan causar daño como el polvo, la humedad, calor, etc.

Asegurando el máximo rendimiento de trabajo sin ocasionar daños a tercero o en el mismo actuador.



Figura 41 Actuador Lineal Eléctrico

2.3.3 Microcontroladores

Es un circuito integrado que tiene la capacidad de ser programado y ejecutar cualquier orden que se haya grabado en su memoria, en su interior tiene las tres unidades importantes de una computadora, es decir, la Unidad Central de Procesamiento o CPU, Memoria y Periféricos de entrada y salida.

Por su reducido tamaño se denomina controlador incrustado, esta característica facilita para que esté incorporado en el dispositivo que va a gobernar, además es un dispositivo que tiene un bajo costo.

Los microcontroladores son dispositivos de baja potencia ya que tienen como corriente máxima de salida 200 [mA] y consumen una potencia de 50 [mW], estos dispositivos se encuentran integrados dentro de otro dispositivo que son capaces de controlar cualquier sentencia que se haya guardado en la memoria ROM, esta programación o líneas de sentencias son orientadas a realizar tareas específicas y son escritas en cualquier lenguaje ensamblador u otro lenguaje válido para microcontroladores, estos programas ensambladores se encargan de codificar las instrucciones que se ejecutan en el microcontrolador.

Éstas características favorecen para utilizarlos como el cerebro de grandes sistemas embebidos, los mismos que controlan todo tipo de sistemas complejos tales como aplicaciones industriales de automatización, robótica, domótica, maquinaria

industrial, equipos médicos, inclusive dispositivos comunes como automóviles, televisores, teléfonos, entre otros.

- **Especificaciones y características técnicas del microcontrolador**

Para cualquiera que sea el proyecto que se va a implementar se debe tomar en cuenta las características generales y técnicas para poder determinar el modelo y familia del microcontrolador que se va a usar, estas son:

El microcontrolador debe tener una capacidad de procesar datos a una velocidad importante, ya que el envío de datos desde el computador por medio de la interfaz grafica va a realizarse constantemente en intervalos de 1 segundo y se desea una respuesta significativa de velocidad rápida en donde los datos recibidos no se pierdan, por lo que es importante encontrar un microcontrolador que maneje datos de 8 bits o 16 bits.

El microcontrolador debe tener un puerto de comunicación USB debido a que es por este medio la comunicación entre la interfaz gráfica y el control de los motores que presionaran las teclas de la máquina de escribir Braille.

El número de puertos de entrada y salida que posea el microcontrolador es muy importante, pues se necesitaran nueve puertos de salida y si es el caso se necesitaran dos puertos de entrada.

La cantidad de información que se va a manejar en este proyecto no es tan extensa, ya que la mayoría de líneas de programación se las realizara en el programa Java con plataforma en NetBeans con el mismo fin de no saturar la memoria y velocidad de respuesta del microcontrolador, por esta razón se piensa que es suficiente tener un microcontrolador que posea una longitud de palabra de no más de 8 bits y en caso extremo se puede usar un microcontrolador que tenga 16 bits.

Como una característica también principal de la búsqueda del microcontrolador se ha considerado el consumo de energía del mismo, pues debe ser un microcontrolador capaz de tener un consumo bajo de energía pero a su vez con una

velocidad de respuesta alta ante alguna interrupción o activación de una señal proveniente de un sensor o pulsador

- **Microcontrolador de la familia 18fxxx**

Después de haber hecho todo el análisis pertinente en todos los microcontroladores que se encuentran en el mercado se ha decidido usar un microcontrolador de la familia PIC18 ya que es el más óptimo y favorece para el desarrollo de este proyecto presentando periféricos USB.

Ésta familia combina el nivel máximo de rendimiento e integración, con la arquitectura de 8 bits, trabaja a 16 millones de instrucciones por segundo (MIPS) de potencia de procesamiento, tiene periféricos avanzados como:

- Control de Área de Red (Controller Area Network – CAN)
- Bus Universal en Serie (Universal Serial Bus – USB)
- Ethernet
- Display de Cristal Líquido (Liquid Crystal Display – LCD)

A continuación se mostraran las características principales de trabajo y funcionamiento de los microcontroladores pertenecientes a esta familia.

Cuadro 1

Características de trabajo y funcionamiento

MICROCONTROLADORES PIC18F2455, PIC18F2550, PIC18F4455, PIC18F4550					
CARACTERISTICAS	18F2455	18F2550	18F4455	18F4550	
Frecuencia de Operación	Hasta 48MHZ	Hasta 48MHZ	Hasta 48MHZ	Hasta 48MHZ	Hasta 48MHZ
Memoria de programa (Byte)	24.576	32.768	24.576	24.576	32.768
Memoria RAM de Datos (bytes)	2.048	2.048	2.048	2.048	2.048
Memoria EEPROM datos	256	256	256	256	256
Interrupciones	18	19	20	20	20
Líneas de E/S	24	24	35	35	35

CONTINUA →

Temporizadores	4	4	4	4
Módulos de Comparación/ Captura/ PWM	2	2	1	1
Módulos de Comparación/ Captura/ PWM mejorado	0	0	1	1
Canales de Comunicación Serie	MSSP, EUSART	MSSP, EUSART	MSSP, EUSART	MSSP, EUSART
Canal USB	1	1	1	1
Puerto Paralelo de Transmisión de Datos (SPP)	0	0	1	1
Cables de Conversión A/D de 10 bits	10 Canales	10 Canales	13 Canales	13 Canales
Comparadores Analógicos	2	2	2	2
Juego de Instrucciones	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)

Fuente: (Data_Sheet_PIC18F2550, 2009)

2.3.4 Bus Universal en Serie USB

El USB fue desarrollado alrededor del año 1990 y es aquí en donde se detalló como deberán ser los conectores, protocolos y cables que serán usados en un dispositivo tipo bus que permitiría la comunicación y conexión entre ordenadores dispositivos o periféricos eléctricos y además tiene la propiedad de suministrar alimentación eléctrica con los dispositivos ya nombrados.

Con el diseño del USB se logró normalizar la conexión entre los periféricos tales como el mouse, teclado, cámara digitales, micrófonos, parlantes, escáneres, impresoras, sistemas de adquisición de datos, tarjetas de red, de televisión, de sonido, etc.

Tal es la importancia y facilidad de este conector que ha sacado del mercado a los puertos serie y paralelo que anteriormente se usaban para conectar varios periféricos, es por esta razón que en los ordenadores y la mayor parte de dispositivos electrónicos los clásicos puertos serial y paralelo han sido reemplazados por USB.



Figura 42 Bus Universal en Serie

- **Clasificación del dispositivo USB**

La clasificación de este dispositivo se basa en la velocidad de transmisión de datos.

- Baja velocidad versión USB 1.0. este tipo de conector puede transferir datos a una velocidad máxima de 188 kbps o 1,5 Mbps, estos generalmente se los encuentra en los periféricos externos como el mouse, teclado, cámaras web, etc., es decir en dispositivos en donde no se necesita una transmisión de datos de velocidad alta.
- Velocidad completa versión USB 1.1. este tipo de conector puede transferir datos a una velocidad máxima de 12 Mbps o 1,5 MBps.
- Alta velocidad versión USB 2.0. este tipo de conector puede transferir datos a una velocidad máxima de 480 Mbps o 60 MBps, esta velocidad es consecuencia de la construcción física del cable pues posee cuatro hilos, dos de ellos se los usa sirven para la alimentación y los otros dos sirven para la transmisión y recepción de datos.
- Súper alta velocidad versión USB 3.0. este tipo de conector puede transferir datos a una velocidad máxima de 4,8 Gbps o 600 MBps, como un nombre mismo lo dice, esta velocidad es diez veces mayor al USB 2.0 y esto se debe a que en su diseño físico se han incluido cinco contactos adicionales, actualmente muchas compañías ya diseñan sus portátiles y placas base incluyendo al USB 3.0.

- **Funcionamiento del Plug USB**

La manera en la que el USB transmite los datos es mediante cuatro hilos, estos hilos están separados por dos pares de cable trenzado que tienen una impedancia de 90Ω .

El primer par trenzado se los asigna a los cables encargados de transmitir y receptor los datos y se los denominan D+ y D- , la transmisión y recepción de los datos se los hace mediante la comunicación half dúplex en la que se transmite o se recepta los datos uno a la vez, nunca se puede transmitir o receptor datos al mismo tiempo.

En cambio, el USB 3.0 que realiza una comunicación full dúplex, es decir que transmite y recepta datos al mismo tiempo, pues al tener cinco hilos utiliza independientemente los cables de par trenzado para cada una de las tareas.

La transmisión de los datos varía según sea la versión del USB que se esté usado, así, para la versión USB 1.0 y USB 1.1 cuando se transmite cero lógico puede ser de 0 a 0.3 [V] y para cuando se transmite un uno lógico puede variar de 2,8 a 3,6 [V]. para el USB 2.0 esta variación esta en ± 400 [mV].

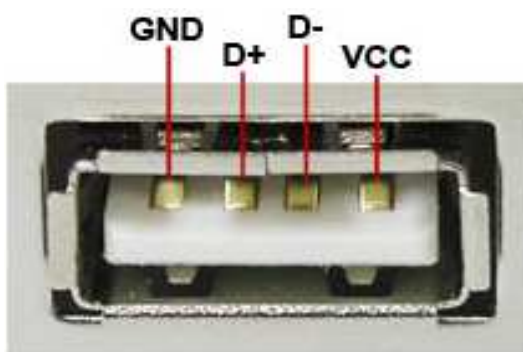


Figura 43 Plug del USB

- **Funcionamiento del cable USB**

El cable USB va a ser el enlace entre la computadora y la tarjeta de control en la aplicación. Se debe tener en cuenta los siguientes parámetros:

Cuadro 2

Parámetros de Conexión

PIN	NOMBRE	COLOR DEL CABLE	DESCRIPCIÓN
1	VCC	Rojo	+ 5 V
2	D-	Blanco	Data -
3	D+	Verde	Data +
4	GND	Negro	Tierra

Con la correcta ubicación de los parámetros tanto en el cable USB como en el PIC 18f2550, la comunicación USB va a ser correcta.

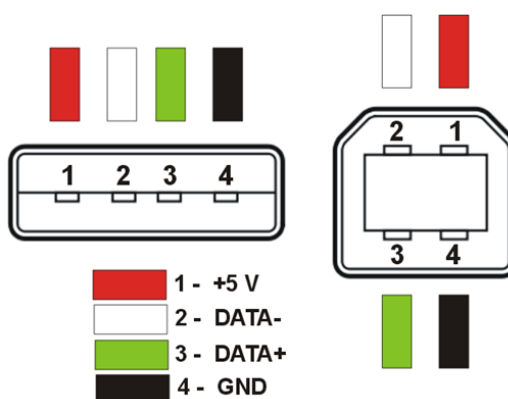


Figura 44 Distribución de los cables del conector USB

2.3.5 Fuente de Alimentación

La fuente de alimentación es aquel dispositivo capaz de convertir la corriente alterna en corriente continua y por ende el voltaje también se convierte, es decir que este dispositivo puede reducir la tensión de entrada de la fuente en AC que tiene un valor de 220 [V] o 115 [V] con la ayuda de un transformador estos valores llegan a ser menores obteniendo por lo general tensiones de 5 a 12 [V] que son los más comunes entre los dispositivos electrónicos que usan un voltaje o tensión continua.

2.4 Recursos de Software

2.4.1 CCS Compiler

El software PIC C Compiler, es un compilador C, conocido también como Compilador PCW de la casa CCS Inc. que incluye los operadores estándares del

lenguaje C y las funciones están almacenadas en bibliotecas específicas para los registros del microcontrolador.

Los desarrolladores que utilizan este software, tienen una herramienta poderosa, ya que el compilador CCS posee más de 307 funciones que están integradas y así simplifican las funciones de hardware del dispositivo como:

- Temporizadores
- Módulos PWM
- Convertidores A/D
- Controladores LCD
- Memoria externa de buses

Una característica principal que tiene el Compiler PCW es que está dentro de un entorno de desarrollo integrado (IDE).

Éste entorno de desarrollo integrado IDE, facilita el progreso de cada fase que se compone un proyecto en general, es decir desde la etapa de edición, programación, compilación y depuración de errores.

Al tener un archivo ya compilado, se genera un archivo fuente .C, este archivo .C se traduce a un lenguaje máquina para que pueda ser leído por el microcontrolador, generándose un nuevo archivo en formato hexadecimal con extensión .hex.

A continuación se explicará cada una de las extensiones generadas al momento de compilar el programa del microcontrolador.

Cuadro 3

Características de archivos generados al compilar el código fuente

ARCHIVO GENERADO	CARACTERÍSTICAS
.c	Archivo de origen, contiene el código fuente.
.h	Archivos estándar o personalizados de cabecera utilizados para definir pines, registros, registros de bits, funciones y directivas del preprocesador.
CONTINUA →	

.tre	Archivo de árbol (tree) muestra el árbol de llamadas, detalla cada función relacionada con la ROM y RAM
.sym	Símbolo de mapa que muestra la ubicación de cada registro y qué variables del programa se almacenan en cada lugar.
.sta	Archivo de estadísticas, muestra la RAM, ROM y PILA en uso, proporcionando información sobre los códigos fuente.
.lst	Archivo de lista, muestra las líneas de código fuente C y el código de ensamblado asociado generado para esa línea.
.hex	El compilador genera archivos .HEX estándar que son compatibles con todos los programadores de Microcontroladores, la salida del compilador puede ser hex 8-bit, hex 16-bit, y los archivos binarios.
.cof	Se trata de un binario que contiene código máquina y la información de depuración.
.esym	Estos archivos se generan para los usuarios de IDE, contiene información de Identificadores y Comentario.
.err	Archivo de error del compilador

El IDE está compuesto por:

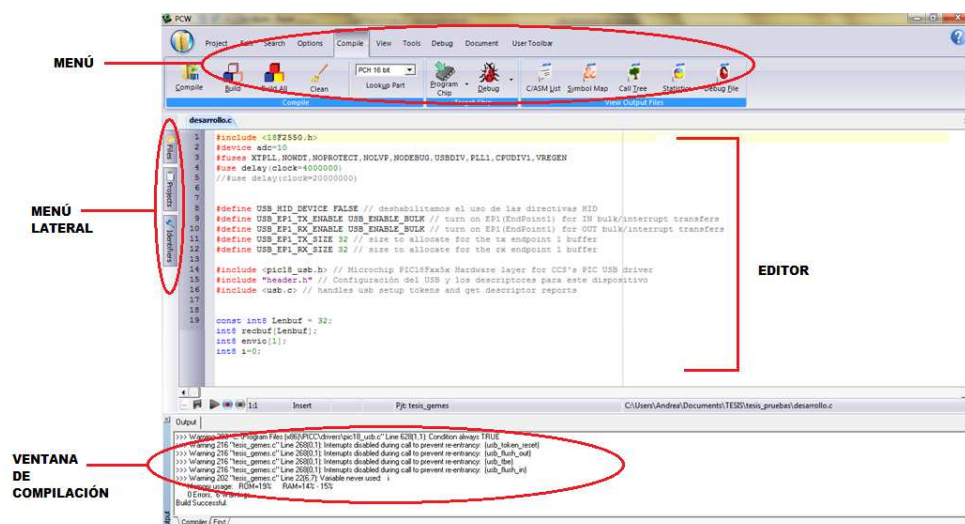


Figura 45 Ventana Principal del CCS Compiler IDE

- Menú: contiene las funciones del IDE, está dividido en secciones, cada una contiene una función específica según sea el caso de necesidad del desarrollador
- Menú Lateral: contiene los archivos del proyecto actual. Muestra dos submenús, uno llamado Proyectos que muestra proyectos recientes y el otro

llamado Identificadores se muestra todas las variables, definiciones, prototipos e identificadores del proyecto actual.

- Ventana de Compilación: se indica la depuración de errores si es que existe e indica el estado actual del código fuente para ser luego grabado en el microcontrolador.
- Editor: éste espacio es el área de trabajo principal del IDE, es aquí donde se deben ingresar y editar los diferentes códigos fuente de programación.

Los programas necesitan una estructura que consiste en cuatro pasos principales:

- Directivas, Funciones, Instrucciones, Comentarios

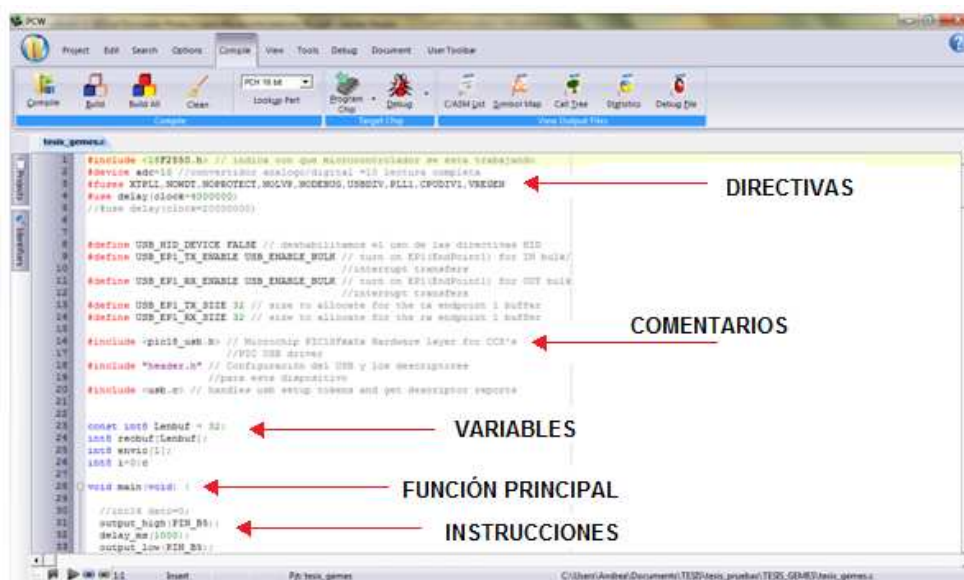


Figura 46 Estructura del Programa CCS Compiler

El encabezado de todo programa en el Compilador CCS se lo llama Directiva, éste controla la conversión del programa a código de máquina por parte del compilador, además se debe declarar los fuses que establecen los distintos bits de configuración del microcontrolador, establecer librerías, entre otros.

Las Variables, se declaran las variables a utilizar con un nombre específico y tamaño de la memoria RAM que se almacenaran en el microcontrolador, determinando el tipo de dato a declarar (entero, flotante, carácter, etc.). En el siguiente cuadro se indican los Tipos de Datos que maneja el Compilador CCS

Cuadro 4

Tipos de Datos

TIPO	TAMAÑO	RANGO	DESCRIPCION
Int 1 - Short	1 bit	0 a 1	Entero de 1 bit
Int 8 - Int 8	8 bit	0 a 255	Entero
Int 16 - Long	16 bit	0 a 65535	Entero de 16 bit
Int 32	32 bit	0 a 4294967295	Entero de 32 bit
Float	32 bit	-1.5×10^{45} a 3.4×10^{38}	Como flotante
Char	8 bit	0 a 255	Carácter
Void	-	-	Sin valor
Signed Int 8	8 bit	-128 a 127	Entero sin signo
Signed Int 16	16 bit	-32768 a 32767	Entero largo con signo
Signed Int 32	32 bit	-2147483648 a 214748647	Entero de 32 bit con signo

La Función Principal, define al programa principal en sí, pues es aquí en donde se escriben las instrucciones para que el microcontrolador las ejecute cuando empieza a trabajar.

Los Comentarios, permiten al programador describir que significan ciertas líneas de programación sin afectar a la ejecución del programa.

2.4.2 Java

El lenguaje de programación llamado Java fue distribuido inicialmente por el año 1995 y fue creado por Sun Microsystems, con el tiempo los derechos fueron adquiridos por Oracle, quien es la compañía responsable de su distribución, las líneas de programación se compilan en bytecode lo que conlleva a que las aplicaciones se puedan ejecutar en cualquier plataforma sin importar la arquitectura que ésta tenga, es decir que es un lenguaje de propósito general, concurrente, orientado a objetos y basado en clases (Java) una de las mayores ventajas de usar este tipo de lenguaje de programación es que puede ser escrito o programado una sola vez y ejecutado en cualquier dispositivo ya sea computador, teléfono celular, consola de juegos, páginas web, aplicaciones de video, etc., es decir que no necesita de una plataforma específica para ser ejecutada y tampoco necesita que se lo recompile cada vez que se lo va a correr o ejecutar, por esto hoy en día este lenguaje es uno de los más usados a nivel mundial,

Java tiene un propósito Write Once, Run Anywhere (WORA), es decir que se escribe o se programa una sola vez y que se puede ejecutar en cualquier parte, siendo un lenguaje de programación independiente del entorno de ejecución.

Un componente de Java es Máquina Virtual de Java (Java Virtual Machine - JVM), permite que cualquier secuencia de programación se ejecute en cualquier plataforma sin importar su origen, esto se debe a la forma en la que se codifican están líneas de programación pues se recopilan en una codificación universal llamada bytecode. El otro componente es el Entorno de Java (Java Development Kit - JDK), provee las herramientas necesarias para el desarrollo de las líneas secuenciales de programación en Java.

A continuación se muestra símbolos de programación en Java, parte esencial de las líneas de código fuente.

Cuadro 5

Símbolos de programación

SEPARADOR	SÍMBOLO	DESCRIPCIÓN
Paréntesis	()	Delimitan listas de parámetros. Modifican la precedencia de una expresión. Delimitan condiciones. Indican el tipo en las coerciones.
Llaves	{ }	Definen bloques de código. Delimitan las listas de valores iniciales de los arrays.
Corchetes	[]	Declaran vectores y permiten acceder a sus elementos.
Punto y coma	;	Terminan instrucciones.
Coma	,	Separan identificadores en declaraciones. Encadenan expresiones
Punto	.	Acceden a los atributos y métodos de una clase Acceden a sub paquete de un paquete.

Se pueden identificar cuatro elementos que se encuentran en un código fuente de Java, estos son:

- Comentario: el compilador de Java lo ignora y son de uso exclusivo del programador ya que describe lo que hace alguna parte específica del código fuente, son delimitados por el signo // o se puede escribir al inicio de la frase /* y al final para cerrar el comentario se coloca */.

- Clase: conjuntos de objetos que pueden compartir características entre sí, una clase está compuesto de atributos, métodos y eventos. Los atributos son los valores que definen y componen a una clase y los métodos son las operaciones que hacen que la clase funcione y además la define, los eventos son la manera en la que todas las clases se relacionan, tanto entre los métodos como en los atributos. A una clase se la define por la palabra propia *class* seguida del nombre colocando al inicio el carácter { y al final el carácter }.
- Método: define a una clase ya que es aquí en donde se programa el código fuente secuencialmente y que finalmente van a ser ejecutadas para operar la aplicación que el programador haya realizado. Los métodos se los reconoce con la palabra *main()* y de igual manera se coloca los caracteres {}, tanto al inicio como al final.
- Sentencia: es la línea que el programador escribe para conformar un programa, estas sentencias se leen y se cumplen secuencialmente, deben terminar con el carácter punto y coma (;), pues este símbolo le indica al compilador que la sentencia a finalizado.

En Java se encuentran los tipos de datos primitivos que son:

Cuadro 6

Tipo de datos en Java.

TIPO	DEFINICIÓN
Boolean	True o false.
Char	Carácter Unicode de 16 bits
Byte	Entero en complemento a dos con signo de 8 bits.
Short	Entero en complemento a dos con signo de 16 bits.
Int	Entero en complemento a dos con signo de 32 bits.
Long	Entero en complemento a dos con signo de 64 bits.
Float	Real en punto flotante según la norma IEEE 754 de 32 bits
Double	Real en punto flotante según la norma IEEE 754 de 64 bits.

2.4.3 NetBeans IDE

Es uno de los tantos entornos que se pueden desarrollar usando el lenguaje Java, y es éste entorno de desarrollo integrado libre lo que le hace tan útil como

herramienta a la programación en Java, es un producto libre de restricciones ya que el código es abierto y distribuido gratuitamente, principalmente por internet.

La comunidad de NetBeans es extensa con programadores de todas partes del mundo y se mantiene en constante crecimiento, es por esta razón que este entorno es de gran utilidad ya que basta con buscar en internet a algún miembro de la comunidad para solicitar ayuda.

El IDE NetBeans fue diseñado para escribir, compilar, depurar o buscar errores y ejecutar programas que hayan sido desarrollados por programadores trabajando bajo módulos que proveen funciones definidas que pueden ser el soporte de las diferentes versiones de Java, permitiendo al programador trabajar de inmediato.

Se puede observar en la figura la Interfaz de NetBeans IDE:

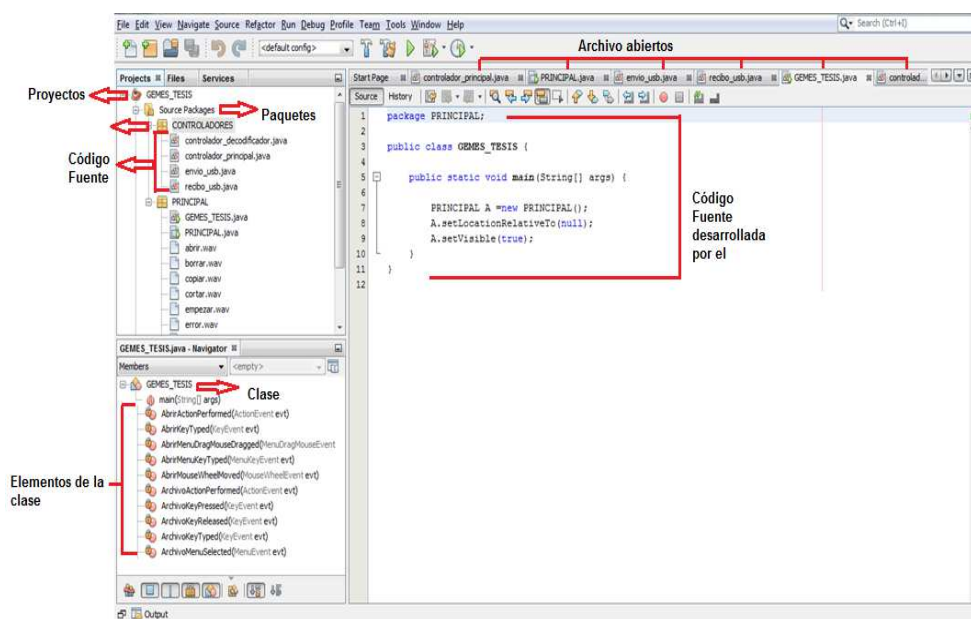


Figura 47 Interfaz de NetBeans IDE.

- **Projects:** contiene los proyectos sobre los que se trabaja o han sido trabajados en el último tiempo.
- **Files:** permite navegar por las carpetas del proyecto.
- **Services:** permite gestionar todas las conexiones que tiene NetBeans hacia los servicios como Bases de datos.

CAPÍTULO 3

DISEÑO DE HARDWARE

3.1 Elección y justificación de los elementos a usar

3.1.1 Análisis de la fuerza que necesita el actuador para presionar una tecla de la máquina de escribir Braille

Antes de utilizar un actuador en especial, sea éste un actuador eléctrico, neumático o hidráulico, primero se debe analizar las características específicas del accionamiento mecánico que tiene la máquina de escribir, como por ejemplo, la longitud máxima que deben ser presionadas las teclas para que marquen el papel. También es importante saber a qué longitud debe ser presionada la tecla del mecanismo de regreso del carro. Y por último saber la presión a ejercer para poder presionar dichas teclas.

Haciendo análisis de cada uno de estos puntos se sabe que:

- Longitud a la cual deben ser presionadas las teclas para poder marcar el papel es de 20 [mm].
- Longitud a la cual debe ser presionada la tecla de salto de línea es de 30 [mm].

La fuerza es el generador idóneo que permite modificar la cantidad de movimiento de los materiales. Trabajando bajo el Sistema Internacional de Unidades, la unidad de medida de la fuerza es el Newton [N].

Sabiendo que m = masa y a = aceleración, por lo tanto se tiene que:

$$F = m * a \quad \rightarrow \quad F [N] = m[kg] * a \left[\frac{m}{s^2} \right]$$

Ahora bien, teniendo la ecuación principal de la fuerza, se debe tener en cuenta cual va a ser el peso necesario para aplicar a dicha fuerza, esto se supo mediante varias pruebas experimentales que consistieron en la utilización de varios pesos que ayuden a presionar las teclas.

Se realizaron varias pruebas con pesos desde 0.45 [kg] – 1 [kg]. Obteniéndose varios resultados.

En las siguientes figuras se muestran las pruebas realizadas con diferentes pesos



Figura 48 Peso de 0.45 [kg]



Figura 49 Tecla presionada por 0.45 [kg]



Figura 50 Peso de 0.63 [kg]



Figura 51 Tecla presionada por 0.63 [kg]



Figura 52 Peso 0.77 [kg]



Figura 53 Tecla presionada por 0.77 [kg]



Figura 54 Peso de 1[kg]



Figura 55 Tecla presionada por 1 [kg]

El peso ideal para presionar la tecla es de 1 [kg], por lo tanto la fuerza necesaria para presionar la tecla es de:

Datos:

$$m = 1 \text{ [kg]}, \quad a = 9.8 \left[\frac{m}{s^2} \right]$$

$$F \text{ [N]} = m[\text{kg}] * a \left[\frac{m}{s^2} \right]$$

$$F \text{ [N]} = 1[\text{kg}] * 9.8 \left[\frac{m}{s^2} \right]$$

$$F = 8.90 \text{ [N]}$$

Al hacer el proceso experimental con los diferentes pesos, la masa que presionó la tecla correctamente fue la de 1 [kg], con lo cual marca en el papel. Aproximando el valor de la fuerza por motivos comerciales, se tiene $F=10$ [N], por lo tanto, éste es el valor principal a considerar para poder elegir un actuador.

En la siguiente tabla se muestran todos los valores calculados de la fuerza en Nétwtones a partir de:

0.45 [kg] a 0.90 [kg].

Tabla 1

Valores de la fuerza en [N]

MASA [KG]	ACELERACIÓN $\left[\frac{m}{s^2}\right]$	FUERZA [N]
0.45	9.8	4.45
0.5	9.8	4.9
0.54	9.8	5.34
0.59	9.8	5.79
0.63	9.8	6.23
0.68	9.8	6.68
0.72	9.8	7.12
0.77	9.8	7.57
0.81	9.8	8.01
0.86	9.8	8.46
0.90	9.8	8.90

3.1.2 Actuadores Eléctrico, Hidráulico y Neumático

Un problema principal para la automatización de la máquina de escribir Braille es el punto de encontrar algún tipo de actuador que permita accionar las teclas de la máquina de una forma precisa, de tal manera que puedan marcar el papel con el punto en relieve, las características físicas también deben ser consideradas debido al espacio físico que tiene la máquina.

- **Cuadro comparativo de actuadores**

En el mercado existen varios actuadores, éstos funcionan de forma eléctrica, hidráulica y neumática. Cada uno de estos actuadores tienen ventajas y desventajas y analizando cada uno de ellos y comparándolos entre sí se puede tomar una decisión respecto al tipo de actuador que se utilizará.

Todos los puntos de vista que se han tomado en cuenta, ayudarán a comparar estos actuadores entre sí, y de ésta manera, tomar una decisión final para poder trabajar con el mejor actuador, bajo las condiciones de trabajo iniciales.

A continuación se explicará éstas ventajas y desventajas mediante un cuadro comparativo.

Cuadro 7

Actuadores Eléctricos, Neumáticos e Hidráulicos

	ELÉCTRICOS	HIDRÁULICOS	NEUMÁTICOS
Fuente de energía	Electricidad	Fluido (aceite mineral) (500 – 100 bar)	Aire a presión (5 – 10 bar)
Opciones	Corriente continua Corriente alterna Motor paso a paso Servomotor Actuadores Lineales	Cilindros Motor de paletas Motor de pistones axiales	Cilindros Motor de paletas Motor de pistón
Tamaño del sistema	Pequeño, considerar cables eléctricos y fuente eléctrica	Grande, considerar mangueras, bomba y caja de aceite	Mediado, considerar las mangueras y el compresor
Costo	Bajo	Alto	Bajo
Potencia	Potencia alta	Alta relación potencia – peso, desarrolla grandes fuerzas	Alta relación potencia – peso, desarrolla grandes fuerzas
Instalación	Rápida instalación (corriente, voltaje, cables)	Requieren instalación especial (bomba, caja de aceite, mangueras)	Requieren instalación especial (compresor, filtros, mangueras)
Velocidad	Velocidad de respuesta rápida	Velocidad de respuesta muy lenta	Velocidad de respuesta rápida
Mantenimiento	Fácil Mantenimiento y acceso a diferentes elementos utilizados en el proceso	Mantenimiento complicado debido al acceso de los diferentes elementos utilizados en el proceso	Mantenimiento complicado debido al acceso de los diferentes elementos utilizados en el proceso
Ruido	Silencioso	Ruidoso	Ruidoso
Robustez	Medianamente robusto	Altamente robusto	Robustos y duraderos
Precisión	Muy preciso y fiables	Preciso	Preciso

Con las diferentes características obtenidas en la tabla comparativa, el Actuador Eléctrico es el más óptimo debido a todas las ventajas que lleva a favor comparándolas con los otros actuadores, tales como el espacio en la instalación es muy pequeña, su costo es accesible, no es ruidoso, tiene la potencia necesaria para trabajar en el sistema de automatización, y sobre todo la fuente de alimentación es pequeña no necesita de un sistema de funcionamiento muy grande.

3.1.3 Especificaciones y características del actuador eléctrico

- Cuadro comparativo de actuadores eléctricos

Los actuadores eléctricos tienen varias opciones, que se derivan según sea la aplicación a la cual van a ser utilizados, de esta manera, se tiene actuadores eléctricos como motores de corriente alterna, motores de corriente continua, motor sin escobillas, motor paso a paso, servomotores, actuadores lineales. A continuación se realizará cuadro comparativo para elegir el actuador eléctrico más adecuado para la aplicación

Cuadro 8

Ventajas y desventajas de actuadores eléctricos

	VENTAJAS	DESVENTAJAS
Motor DC	<p>Amplio rango de variación de velocidad, facilidad de control de posición, torque, velocidad y tamaño</p> <p>Alta eficiencia, silenciosos, limpios y apenas vibran.</p> <p>Rendimiento elevado (típicamente en torno al 80%, aumentando el mismo a medida que se incrementa la potencia de la máquina</p> <p>Se controla mediante PWM</p> <p>Es fácil de manipular, se puede controlar el arranque y el paro</p> <p>Trabajan a con cualquier valor de torque y velocidad. Ofrece gran potencia por tiempos cortos</p>	<p>Tiene problemas de mantenimiento debido al desgaste de algunas de las piezas</p> <p>Baja relación peso / potencia, ya que si se desea mayor potencia, el motor debe ser grande.</p> <p>Incapacidad de funcionar a velocidades bajas, necesita de etapa reductora.</p> <p>El problema del torque y la velocidad se soluciona empleando cajas reductoras, permitiendo la disminución de la velocidad de giro y aumentar el torque del motor.</p> <p>Debido a la gran velocidad de giro, suele ser algo ruidosos</p>
Motor a pasos	<p>Este motor es muy comercial, es fácil adquirirlo.</p> <p>Fácil de trabajar debido a la secuencia binaria de programación, por lo tanto el ángulo de rotación es proporcional a los pulsos de entrada.</p>	<p>Se sobrecalienta de manera continua.</p> <p>Muy difícil de controlar con altas velocidades y altas frecuencias, pierde pasos.</p> <p>Tiene resonancia cuando el control no es el adecuado.</p>

CONTINUA →

Motor a pasos	Exactitud en la posición y repetición de movimientos Tiene una respuesta rápida frente al arranque, parada y reversa. No tiene contacto con escobillas en el motor, lo que le hace muy fiable.	Si se desea trabajar con una potencia elevada, el tamaño del motor aumenta y el costo también. No se puede conseguir motores de alta potencia con facilidad.
Servomotor	Se controla mediante PWM con pulsos entre 0.5[ms] a 2.5[ms] Motor de corriente continua con etapa reductora y sensor de posicionamiento Altos valores de par con baja tensión. Incluyen un sensor de posición Son precisos y sumamente poderosos a pesar de su tamaño. No consume mucha energía	Resolución máxima de 180 grados, limitado por el circuito de control. Irreversible a la etapa reductora o tren de engranajes Necesidad de un circuito alternativo de seguridad para que funcione en caso de fallo. Necesidad de un circuito driver para poder controlarlo con mayor precisión. Si se desea trabajar con un torque elevado el tamaño del motor y el costo aumentan
Solenoides	Trabaja por un campo magnético uniforme e intenso. Alta fuerza y potencia, Fácil mantenimiento y es silencioso Debido a su accionamiento eléctrico, se puede instalar en lugares remotos y ser controlados por interruptores eléctricos. Presenta un núcleo móvil que puede ser de tipo “pull” (arrastre) o de tipo “push” (empuje) que presenta fuerza muy alta en carreras cortas	El costo es alto, Consumo de corriente elevado Al consumir una alta corriente, se eleva la temperatura, lo que provoca pérdida del aislamiento del alambre de cobre causando un corto circuito. Si se necesita un accionamiento con carreras largas, la fuerza disminuye Si se necesita tener una fuerza elevada de accionamiento, se debe tener un devanado grande, esto hace que la solenoide sea pesada
Actuador lineal	Realiza un movimiento lineal simple, seguro y limpio con un control de movimiento preciso y suave. Fácil instalación y necesita de menos espacio. Tiene una larga vida útil con un mantenimiento prácticamente nulo, asegurando el costo total de operación muy bajo en comparación con otros sistemas	En el mercado existen actuadores con carreras limitadas y estándares. Al tener carreras estándares se debe acoplar a las necesidades de la aplicación para que trabaje correctamente. Cuando se tiene un actuador pequeño, los engranes del conjunto de la caja reductora es de plástico, limitando la vida útil.

CONTINUA →

Actuador Lineal	Los sistemas son silenciosos, limpios, no tóxicos y de energía eficiente.	Mientras más fuerza debe emplear más corriente debe consumir
	Trabajan con energía eléctrica, esto permite un movimiento suave sin la necesidad de lubricar alguna parte del sistema.	

Una vez analizados los diferentes actuadores eléctricos, se optó por utilizar los actuadores lineales eléctricos, debido a su gran precisión de trabajo, como son lineales, simulan un dedo presionando la tecla de forma puntual, el tamaño idóneo para implementar sobre la máquina, la fuerza de trabajo es muy bueno, esto permitirá marcar el papel, el bajo costo los hace accesibles al tomar en cuenta que se trabajará con nueve actuadores, la carrera tiene la longitud exacta que necesitan las teclas para ser presionadas y la energía de alimentación es limpia.

En el siguiente cuadro se dará a conocer las especificaciones técnicas de los actuadores lineales que se utilizaran en el sistema de automatización

Cuadro 9

Especificaciones técnicas de actuadores lineales.

Especificaciones técnicas	
Voltaje de funcionamiento	12 [V] DC ± 2 [V]
Máxima corriente	2.3 [A]
Mínimo de corriente	0.15 [A]
Fuerza máxima de carga	4.5 – 5.5 [Kg]
Carrera	20 [mm]
Control de tiempo	0.2 [seg]
Temperatura de trabajo	-30 [°C] - +70 [°C]
Tiempo de vida útil	100000 veces
Peso	0.11 [Kg]
Carrera	20[mm] - 50 [mm]
Motor	Motor de corriente continua

Los actuadores empleados para las 6 teclas correspondientes al signo generador y la tecla del espacio, utilizarán un actuador lineal eléctrico que tiene de carrera 20 [mm], como lo muestra la figura:



Figura 56 Actuador lineal de 20 [mm]

Para la tecla del salto de línea se empleará un actuador lineal eléctrico con una carrera de 30 [mm] a 12 [V].



Figura 57 Actuador Lineal Eléctrico

Microcontrolador 18f2550

El Microcontrolador 18f2550 contiene un puerto USB, trabaja baja velocidad (1.5Mbps, con un reloj de 6Mhz) ó alta velocidad (12 Mbps con un reloj de 48 Mhz.).

Con 48 Mhz (velocidad máxima) el período de ejecución de cada instrucción es de 83.3 nanosegundos, considerando que cada instrucción se ejecuta en 4 ciclos de reloj; este tipo de comunicaciones soporta transferencias de control, interrupción, masivas e isócronas, tiene la ventaja de tener una comunicación a alta velocidad con un dispositivo USB y el microcontrolador. Además tiene un módulo conversor A/D con 10 canales de entrada y una memoria de datos Electrical Erasable Programmable Read Only Memory (EEPROM) de 256 bytes. La operación de borrado y programación es muy sencilla y se puede grabar y borrar tantas veces como se quiera.

Este microcontrolador cuenta con un completo diseño de hardware, pues puede trabajar con un oscilador interno como con uno externo y trabaja con los dos perfectamente, en el encapsulado también se encuentra un circuito multiplicador de frecuencia Phase Locked Loop (PLL), estos dos circuitos son los encargados de controlar la velocidad de operación del microcontrolador, pues el oscilador interno o externo determina la frecuencia y la configuración del PLL la multiplica según los requerimientos de operación.

Se ha escogido el PIC 18f2550, debido a que dispone de un puerto de comunicación USB, la longitud de palabra es de 8 bits (alto rendimiento arquitectónico), la memoria de programación es Flash y tiene una memoria ROM de datos donde se guardan los datos adquiridos a través de actuadores, los mismos que serán enviados al computador dependiendo de la programación requerida del microcontrolador. Tiene un consumo bajo de potencia, la memoria direccionable de programación es de hasta 2 MB, la memoria RAM es de 4 Kb.

El puerto USB de este microcontrolador funciona a 1,5 Mbps (baja velocidad) y máximo a 12 Mbps (velocidad completa), soporta interrupciones masivas e asincrónicas, el módulo conversor A/D tiene 10 canales de entrada y la memoria EEPROM de 256 bytes, con esta característica la memoria es de lectura, programación y borrado, es decir, se puede leer, grabar y borrar el microcontrolador tantas veces sean necesarias.

Básicamente es un microcontrolador CMOS Flash potente con 32 bits con puerto USB V2.0 y lo más importante es que es fácil de programar. En el mercado se encuentra en una cápsula Dual In Line Package (DIP), es decir, es un circuito integrado con encapsulado rectangular, con 28 pines ubicados simétricamente de forma paralela en los lados (14 pines por lado), sus dimensiones son 10.34 x 17.87 x 2.50 mm, lo que le hace interesante es que es un microcontrolador Small Outline Integrated Circuit (SOIC) es un circuito integrado pequeño, fácil de manejar.



Figura 58 PIC 18f2550

Características eléctricas principales del microcontrolador 18f2550.

Cuadro 10

Parámetros del microcontrolador 18f2550

PARAMETROS PIC 18F2550	ESPECIFICACIÓN
Voltaje de Alimentación	5 V
Voltaje en cualquier pin con respecto a VSS (excepto VDD, MCLR y A4)	0.3V a (VDD + 0.3V)
Voltaje en MCLR con respecto a VSS	0 V a 13,25 V
Voltaje en VDD con respecto a VSS	-0.3V a 7.5 V
Corriente máxima de salida	300 mA
Corriente máxima de entrada	250 mA
Corriente máxima de E/S de puertos	25 mA
Corriente máxima de E/S de puertos	200 mA
Potencia de disipación total	1 W
Temperatura de almacenamiento	-65 °C a 150 °C
Temperatura ambiente	-40 °C a 85 °C
Frecuencia de Operación	48 MHz DC
Memoria de Programa (bytes)	32768
Puertos de entrada y salida	Puertos A, B, C, (E)
Timers	4
Módulos de PWM	2

3.1.4 Distribución de los pines utilizados

En la Figura se puede apreciar la distribución de cada uno de los pines para el microcontrolador 18f2550.

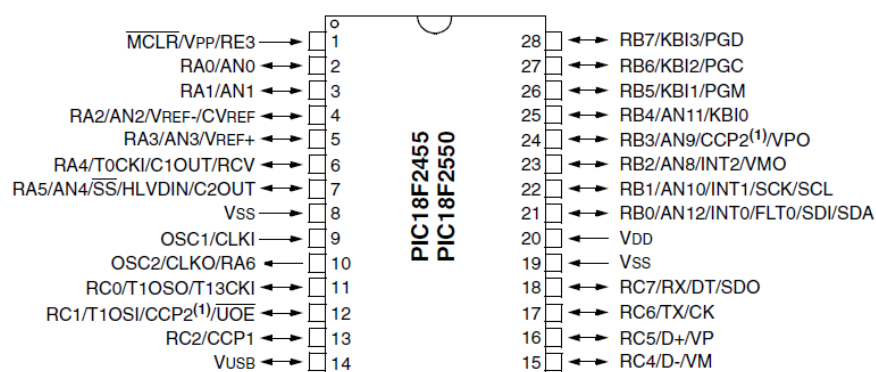


Figura 59 Distribución de los pines Microcontrolador 18f25550

En el siguiente cuadro se indican las características de los pines que se están utilizando en la programación del proyecto.

Cuadro 11

Características 18f2550 (Data_Sheet_PIC18F2550, 2009)

NOMBRE	PIN	E/S	DESCRIPCIÓN	FUNCIÓN
MCLR	1	Entrada	Master Clear	Reset
AN0	2	Entrada	Entrada Analógica	Estado del USB
A1	3	E/S	E/S Digital	Activa el actuador del carro de retorno
A2	4	E/S	E/S Digital	Activa motor del carro de retorno
A3	5	E/S	E/S Digital	Activa motor del carro de retorno
Vss	8		Tierra de referencia para pines lógicos de E/S	
	19			
OSC1	9	Entrada	Entrada del reloj	
OSC2	10	Salida	Salida del reloj	
VUSB	14		Regular de voltaje USB 3.3 V	Capacitor 4,7 Uf
D-	15	E/S	Diferencial USB (E/S)	Datos de recepción y transmisión
D+	16	E/S	Diferencial USB (E/S)	Datos de recepción y transmisión
VDD	20		Fuente positiva para pines lógicos de E/S	
B0	21	E/S	E/S/Digital	Activación tecla 1
B1	22	E/S	E/S/Digital	Activación tecla 2
B2	23	E/S	E/S/Digital	Activación tecla 3
B3	24	E/S	E/S/Digital	Activación tecla 4
B4	25	E/S	E/S/Digital	Activación tecla 5
B5	26	E/S	E/S/Digital	Activación tecla 6
B6	27	E/S	E/S/Digital	Activación tecla espacio
B7	28	E/S	E/S/Digital	Activación tecla salto de línea

3.2 Bus Universal en Serie (Universal Serie Bus – USB)

3.2.1 USB del microcontrolador 18f2550

El microcontrolador 18f2550, soporta comunicaciones con un puerto USB 2.0, la señal de salida se genera con la potencia de salida con un valor de 3.3VDC, este voltaje entrega el Pin 14, en el cual se debe colocar un condensador de 4.7 [uF] de desacoplo a tierra.

Una de las ventajas de usar una conexión USB es que para la alimentación del microcontrolador no es necesario colocar una fuente externa, pues el cable le alimenta directamente conectando a un ordenador con 5 [V].

- **Características del puerto USB del microcontrolador 18f2550**
- USB V2.0
- Velocidad Baja (1,5 Mbs)
- Velocidad Alta (12 Mbs)
- Soporta control de interrupción
- Soporta hasta 32 puntos finales (16 bidireccional)
- RAM de acceso dual 1-Kbyte para USB
- Transmisión de puerto paralelo (SPP) para la salida continua USB
- Transferencias (dispositivos 40/44-pin solamente)

El microcontrolador 18f2550 tiene una interfaz compatible USB con velocidades altas y bajas, lo que le permite tener una comunicación rápida entre cualquier host USB y el microcontrolador 18f2550

El Serial Interface Engine – Motor de Interface Serial (SIE) se conecta directamente al USB, utilizan un transceptor interno, también se utiliza un transceptor externo. Tiene disponible un regulador de voltaje interno de 3,3 [V] para alimentar el transceptor interno cuando se tiene aplicaciones de 5 [V].

Existe una memoria de puerto dual, ya que la memoria que tiene el núcleo del microcontrolador y la SIE es compartida.

Se ha proporcionado un puerto Paralelo Streaming que ayuda cuando existe transferencia de datos muy grandes.

- **Energía entregada por el USB**

Esta energía proviene del USB, de ésta manera, cuando se desea realizar alguna aplicación se puede alimentar el circuito de la aplicación directamente con el Voltaje proporcionado por el USB

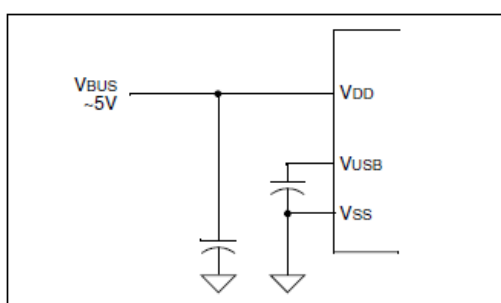


Figura 60 Energía entregada por el USB

Fuente de alimentación

La fuente de alimentación tiene específicamente 12 [V] DC y 20 [A]. Éstas características eléctricas fueron tomadas en cuenta debido a que los actuadores trabajan con 12 [V] DC y una corriente máxima de 2.5 [A].



Figura 61 Fuente de alimentación 12 [V] dc 15 [A]

En el siguiente cuadro se explica las características eléctricas de todos los elementos utilizados en el sistema.

Cuadro 12

Características eléctricas de elementos

VOLTAJE Y CORRIENTE	APLICACIONES
12 [V]	Voltaje que consumen los actuadores
5 [V]	Alimentación del USB Alimentación de microcontrolador
2.5 [A]	Corriente que consumen los actuadores

Para obtener un valor promedio de la corriente consumida por los actuadores se ha medido la corriente de dicho actuador mientras está en funcionamiento en un intervalo de 30 segundos, obteniendo los siguientes resultados:

Tabla 2

Corriente medida de los actuadores:

TIEMPO [S]	CORRIENTE [A]
2	2.40
4	2.55
6	2.49
8	2.39
10	2.54
12	2.21
14	1.61
16	2.58
18	2.02
20	1.80
22	2.64
24	2.09
26	3.42
28	2.20
30	3.40
Promedio	2.42

Obteniendo un promedio de trabajo de la corriente consumida en un intervalo de 30 segundos, se adquirió un valor de 2.42 [A], a éste valor hay que aproximarle un valor de corriente comercial, por lo tanto el valor de la corriente consumida por cada actuador es de 2.5 [A].

En el caso más extremo estarán trabajando al mismo tiempo cinco actuadores que consumirán aproximadamente 12 – 13 [A].

Ees por ésta razón que se ha considerado trabajar con una corriente que brinde 15 [A].

Cuadro 13

Especificaciones técnicas de la fuente de alimentación

FUENTE DE ALIMENTACIÓN	
Corriente de salida	15 [A]
Voltaje de entrada	110 – 120 [V] AC
Frecuencia	50 – 60 [Hz]
Temperatura de trabajo	0 – 40 [°C]
Temperatura de almacenamiento	-20 a 60 [°C]
Protecciones	Sobrecargas, tensión de cortocircuito
Humedad de ambiente	0 - 90 % sin condensación
Material	Caja de metal con base de aluminio
Dimensiones	198 x 109 x 50 [mm]
Peso	808 [gr]

3.3 Construcción de la estructura mecánica

La estructura mecánica se ha construido a partir de un armazón metálico de acero estructural lo suficientemente robusto para soportar el peso de todo el sistema, es decir, soportar el peso de la estructura que tolera los actuadores y el peso de la máquina de escribir.

Se ha considerado también construir un armazón que sea capaz de ajustar a la máquina de escribir sin dañar físicamente ningún punto de la misma, y que al mismo tiempo la sujete para que no tienda a moverse de su posición original cuando esté trabajando.

En la siguiente figura se indica la estructura metálica de acero estructural, se ajusta a las medidas físicas de la máquina de escribir.



Figura 62 Estructura metálica de acero estructural

Debido al espacio físico que tienen las teclas, para que los actuadores lineales puedan calzar y posicionarse de la manera correcta, se debe acoplar un soporte de nylon para que las teclas puedan ser presionadas.

Estos soportes de nylon tienen el tamaño preciso para no chocar entre si y permitan el fácil movimiento de las actuadores y además que cumplan con la función de presionar a cada tecla. A continuación se muestra los soportes de nylon antes de ser montados en el armazón de acero estructural.



Figura 63 Soportes de nylon para teclas

Acoplados al mecanismo los soportes de nylon ayudan a que los actuadores lineales queden perpendicularmente a cada tecla, y el accionamiento lineal sea correcto, en la siguiente figura se muestra el acople de los soportes de nylon junto con los actuadores.



Figura 64 Soportes de nylon acoplados a los actuadores lineales

Al activarse el actuador lineal, presiona el soporte de nylon, y éste a su vez trabaja haciendo pivote sobre la tecla, lo que hace que la tecla marque el papel perfectamente.



Figura 65 Activación del actuador lineal

Para que la máquina de escribir calce en la estructura, se ha colocado topes de nylon en la parte inferior para que la maquina entre y haga contacto con estos soportes, haciendo que la máquina quede muy sujeta y firme, y a la vez no se estropee físicamente. En la siguiente figura se indican los topes de nylon.



Figura 66 Topes de nylon para sujeción de la máquina

La construcción del recorrido del sistema del carro tomó cierto tiempo hasta poder definir los parámetros de funcionalidad con diferentes elementos.

Finalmente se decidió hacerlo mediante el empleo de:

- Motor reductor
- Poleas dentadas
- Banda

De tal manera que al ser activado el motor reductor permita el recorrido del sistema del carro tanto de ida como de regreso.

En la siguiente figura se muestra el sistema ya implementado en la estructura.



Figura 67 Motor reductor, poleas dentadas, banda

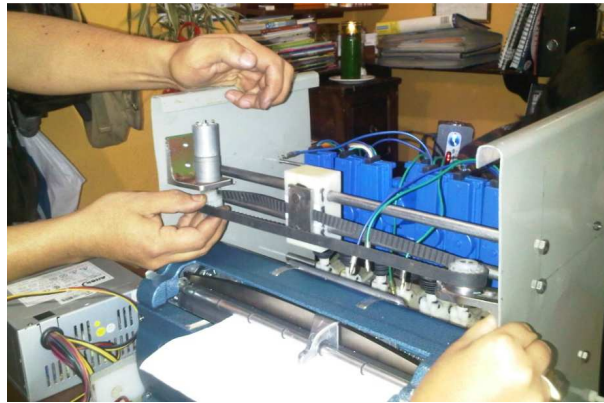


Figura 68 Sistema del recorrido del carro implementado en la estructura

A continuación se mostraran diferentes figuras en las cuales se indicara la estructura finalizada con todos los acoples necesarios y respectivos para el correcto funcionamiento del proyecto.



Figura 69 Carcasa de la estructura mecánica



Figura 70 Posicionamiento de los actuadores y soportes



Figura 71 Posicionamiento del sistema del recorrido del carro

3.4Diseño de la placa electrónica

Para el diseño del circuito, fue necesario tomar en cuenta cuales son los elementos idóneos que sirvan para el buen desempeño del sistema en el proyecto.

Fue necesario dividir en dos partes al circuito eléctrico:

- Circuito digital que será el encargado de tener en su configuración al microcontrolador 18f2550 y todas las conexiones principales para su funcionamiento
- Circuito de potencia en donde se usaran elementos electrónicos que ayuden a la activación de los motores para que presionen las teclas.

3.4.1 Conexión del microcontrolador 18f2550

La conexión de los elementos necesarios para que el microcontrolador 18f2550 funcione correctamente básicamente son: un cristal de 20 MHz, capacitores, y resistencias.

En la figura se indica la conexión del microcontrolador 18f2550 para que trabaje con conexión USB.

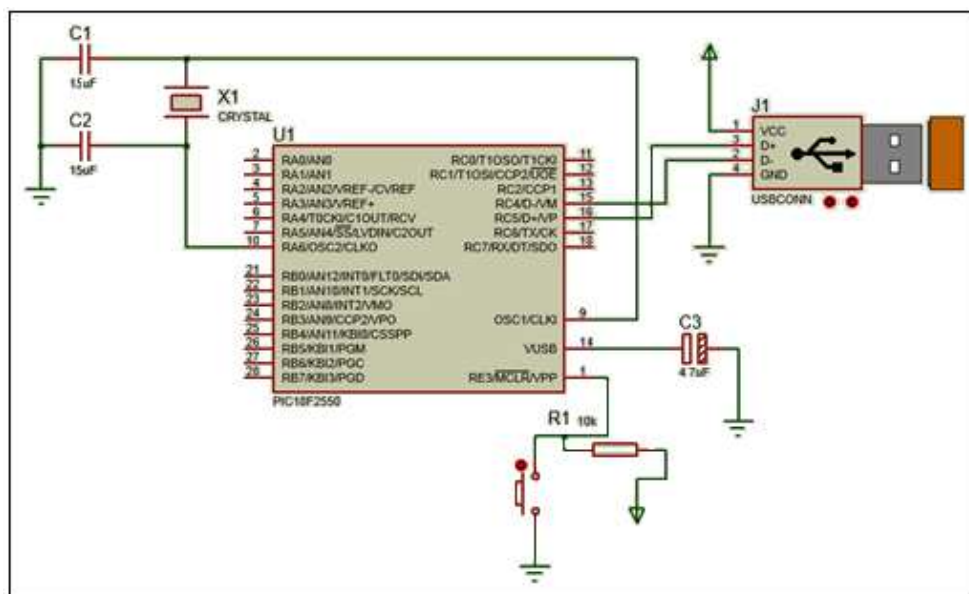


Figura 72 Conexión del microcontrolador 18f2550

Se debe conectar un capacitor de 100nF de desacoplo entre los pines 19 y 20 para que almacene energía y estabilice la tensión de alimentación. Los pines 8 y 19 van a tierra y el pin 20 a Vcc del USB. En el pin 14 se debe conectar un capacitor electrolítico de 0.47 uF a 100 V.

3.4.2 Conexiones de los actuadores lineales eléctricos

- **Actuadores lineales eléctricos que activan las teclas**

Para conectar a los actuadores lineales eléctricos que se escogieron para este proyecto, fue necesario diseñar una etapa de potencia, pues el microcontrolador que es el elemento encargado de enviar las señales respectivas para que cada uno de éstos se activen trabaja a 5 [V] dc y los motores trabajan a 12[V]dc.

Para que los dos circuitos eléctricos trabajen a diferentes valores de voltaje dc independientemente se decidió usar un transistor que sea capaz de trabajar como switch o interruptor de tal manera que el circuito digital trabaje a 5[V] y el circuito de potencia trabaje a 12[V].

Para la elección del transistor es necesario tomar en cuenta los valores máximos con los que trabaja este transistor y que no excedan tanto en el circuito como en el

elemento para que ninguno de los dos circuitos sufra un daño extremo. La siguiente tabla muestra estos valores:

Cuadro 14

Valores máximos de trabajo del TIP 122

SIMBOLO	PARÁMETRO	RANGO	UNIDAD
V_{CBO}	Voltaje Colector-Base		
	TIP120:	60	V
	TIP121:	80	V
	TIP122:	100	V
V_{CEO}	Voltaje Colector-Emisor		
	TIP120:	60	V
	TIP221:	80	V
	TIP122:	100	V
V_{EBO}	Voltaje Emisor-Base	5	V
I_C	Corriente Colector (DC)	5	A
I_{CP}	Corriente Colector (Pulso)	8	A
I_B	Corriente Base (DC)	120	mA
P_C	Disipación Colector ($T_a=25C$)	2	W
	Disipación Colector ($T_c=25C$)	65	W
T_J	Temperatura de Junción	150	C
T_{STG}	Temperatura Almacenamiento	de -65 ~150	C

(Data_Sheet_TIP_122)

Para este caso se tomó en cuenta el voltaje entre el colector emisor V_{ce} y el voltaje colector base V_{cb} que en los dos casos debe ser máximo de 100[V] y en el circuito del proyecto se tendrá un voltaje máximo de 12[V].

por lo que se encuentra dentro del rango de trabajo seguro, también es importante tener en cuenta la corriente del colector I_c que es de máximo 5[A], esta corriente es la de trabajo de cada uno de los actuadores lineales que tienen un valor máximo de 2.5[A] razón por la que también se encuentra dentro del valor de trabajo seguro.

En el siguiente cuadro se mostrará los valores característicos eléctricos del transistor.

Cuadro 15

Características eléctricas del TIP 122

SIMBOLO	PARÁMETRO	TEST	MIN.	MAX.	UNI.
VCEO(sus)	Voltaje Colector				
	Emisor				
	TIP120:	Ic= 100 mA Ib=0	60		V
	TIP121:		80		V
TIP122:		100		V	
ICEO	Voltaje Colector				
	Emisor				
	TIP120:	VCE=30V, IB=0		0,5	mA
	TIP121:	VCE=40V, IB=0		0,5	mA
TIP122:	VCE=50V, IB=0		0,5	mA	
ICBO	Voltaje Colector				
	Emisor				
	TIP120:	VCB=60V, IE=0		0,2	mA
	TIP121:	VCB=80V, IE=0		0,2	mA
TIP122:	VCB=100V, IE=0		0,2	mA	
IEBO	Corriente de corte OFF	VBE=5V, IC=0			
Hfe	Ganancia corriente DC	VCE=3V IC=0,5A	1000		
		VCE=3V IC=3 ^a	1000		
VCE(sat)	Voltaje Colector-Emisor	IC=3A, IB=12Ma		2,0	V
		IC=5A, IB=20mA		4,0	V
VBE(on)	Voltaje Base-Emisor	VCE=3V, IC=0, f=0,1MHz		2,5	V
Cob	Capacitancia Salida	VCB=10V, IE=0, f=0,1MHz		200	pF

(Data_Sheet_TIP_122)

Estas características son necesarias revisarlas para conocer los valores específicos que deben tener tanto la corriente como el voltaje para que el transistor trabaje como un interruptor.

Para cumplir con este propósito se debe conocer que un transistor funciona como un interruptor si en su configuración de conexión es capaz de pasar rápidamente de corte a saturación y viceversa, es decir que cuando se encuentra en corte es un interruptor abierto y en saturación es un interruptor cerrado, por este motivo se debe conocer el valor de voltaje y corriente con que va a trabajar el circuito.

El voltaje V_{cc} es el voltaje nominal del circuito, y la corriente es la corriente de saturación en el colector I_{csat} , a continuación se calculara la el valor mínimo de esta

corriente y en su defecto se calculará el valor mínimo de la resistencia que deberá ir conectada en la base para permitir este valor de corriente.

$$I_{Bsatmin} = \frac{I_{Csat}}{\beta} \qquad R_{bmax} = \frac{V}{I_{Bsatmin}}$$

En donde:

- $I_{Bsatmin}$ es la corriente mínima en la base
- I_{Csat} es la corriente de saturación en el colector
- R_{bmax} es el valor de la resistencia en la base
- V es el voltaje con el que trabaja el circuito
- $I_{Bsatmin}$ es la corriente en la base mínima para lograr tener el disparo

$$I_{Bsatmin} = \frac{I_{Csat}}{\beta}$$

$$I_{Bsatmin} = \frac{2.5[A]}{1000}$$

$$I_{Bsatmin} = 2.5 [mA]$$

$$R_{bmax} = \frac{V}{I_{Bsatmin}}$$

$$R_{bmax} = \frac{12[V]}{0.0025[A]}$$

$$R_{bmax} = 4800 \Omega \approx 4.8 K\Omega$$

Como se ha calculado el valor de la resistencia de base máximo que garantice que el transistor trabaje como interruptor es de $4.8 K\Omega$, hay que tener claro el valor de la resistencia en la base que se va a usar debe ser de por lo menos 4 veces menor que R_{Bmax} , por esta razón se decidió usar una resistencia de $1 K\Omega$, con este valor el transistor trabaja como interruptor sin problema, de esta manera cuando el

microcontrolador 18f2550 envía la señal de control hacia la base, el transistor empieza a trabajar como interruptor lo que activa a los actuadores lineales de manera independiente con el valor de voltaje con el que trabajan.

La siguiente figura se muestra el diseño de la etapa de potencia del circuito, se colocó un diodo como forma de protección tanto del transistor como del microcontrolador que se encuentra conectado a su base, pues se está activando carga inductiva, en este caso los motores de cada uno de los actuadores, y el transistor al pasar rápidamente de corte a saturación y viceversa presenta la conocida patada inductiva, lo que produce que el transistor llegue a quemarse, esta conexión de protección se la conoce como diodo volante, esta configuración consiste en conectar un diodo en paralelo polarizado inversamente con la carga inductiva permitiendo evitar el sobre voltaje que existe cuando el motor arranca, además evitar que la corriente del motor se retroalimente por el transistor quemándolo. También se colocó un capacitor en paralelo con el diodo y el actuador debido a que cuando se energiza una carga inductiva este crea un campo electromagnético y por ende tiene una corriente que puede afectar al circuito digital.

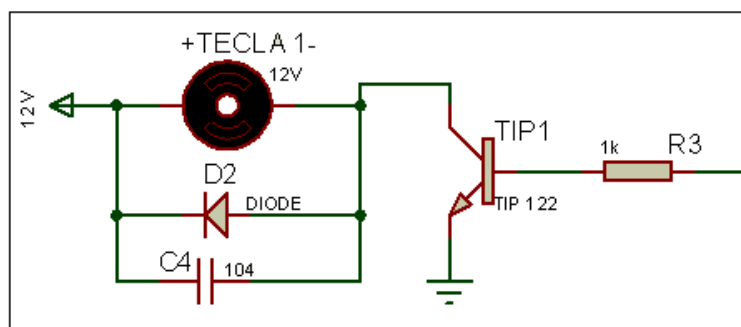


Figura 73 Circuito de conexión del actuador

- **Actuadores lineales eléctricos que activan el salto de línea y el carrito**

La conexión para estos dos actuadores difiere de la conexión de los actuadores que activarán a las teclas, esto se debe a que la activación de las teclas solo necesita un pulso que las presionan y por su estructura mecánica el pistón regresa a su posición normal al desconectar la alimentación, a diferencia de los actuadores

lineales eléctricos que activan al salto de línea y al movimiento del carrito, pues estos dos actuadores necesitan regresar a su posición inicial mediante las líneas programación del microcontrolador 18f2550, es decir, es necesario que estos dos actuadores lineales tengan un movimiento bidireccional y esto se lo hace mediante el uso del integrado L293D que controla motores de corriente continua.

El integrado L293D es un puente H conocido también como medios puentes, es por esto que es muy útil al momento de conducir corrientes bidireccionales de hasta 1 [A] con valores de voltaje desde 4.5[V] hasta 36 [V], en general se puede usar con cualquier carga que requiera corriente y voltaje con altos valores.

Las entradas son de tipo TTL es decir lógica transistor a transistor utilizado para la construcción de circuitos electrónicos digitales, las entradas y salidas del dispositivo trabajan normalmente con 5 [V], comprenden dos estados lógicos definidos por el rango de voltaje: L bajo (0,0[V] hasta 0,8[V]) y H alto (2,4[V] hasta Vcc).

Este integrado puede activar a dos motores DC pues presenta cuatro configuraciones en Puente H completo, lo que permite el cambio bidireccional de cada uno de los motores.

Presenta dos puertos que habilitan a cada motor independientemente, es muy versátil pues no necesita adicionar la conexión de diodos que eviten el regreso de la corriente debido a la conexión de cargas inductivas ya que los lleva incorporados en el integrado.

La siguiente imagen muestra cómo trabaja este integrado, mediante la configuración del Puente H y la manera en la que hace que el motor se trabaje de forma bidireccional según sean las necesidades del programador.

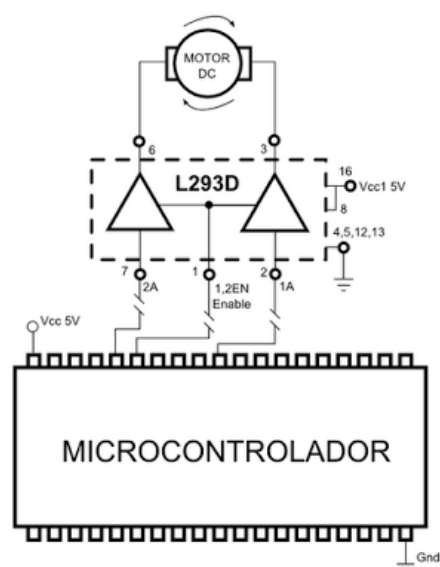


Figura 74 Funcionamiento del integrado L293D

La siguiente imagen muestra la conexión del driver L293D con el microcontrolador 18f2550, se puede observar que los terminales que corresponden a EN1 y EN2 están conectados al terminal A4 del microcontrolador, pues para que el driver active los motores es necesario activar en alto a estos pines, las entradas 2, 7, 10 y 15 corresponden a la señal que envía el microcontrolador para controlar el movimiento del motor, y los demás pines se conectan tal como se muestra en la figura.

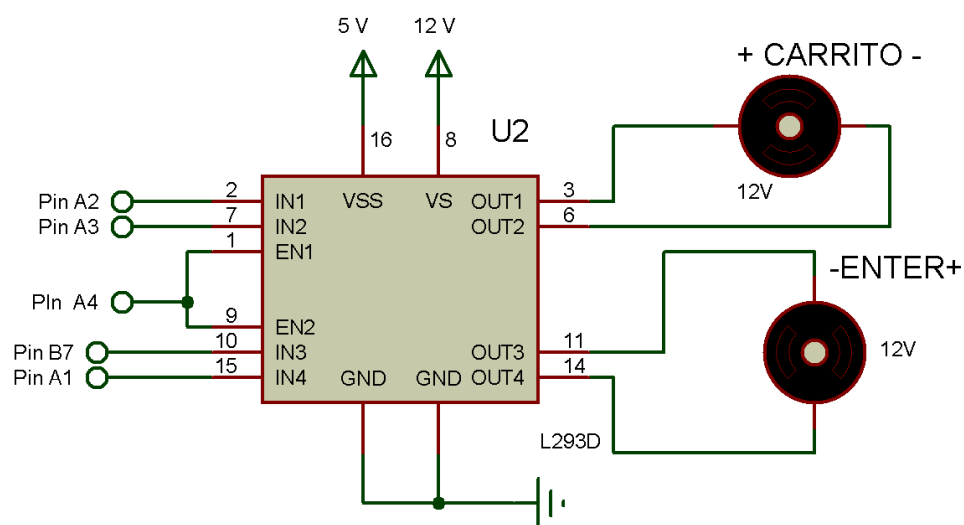


Figura 75 Conexión del integrado L293D con el microcontrolador 18f2550 y los actuadores lineales

El siguiente cuadro indica la funcionalidad de cada uno de los pines del integrado

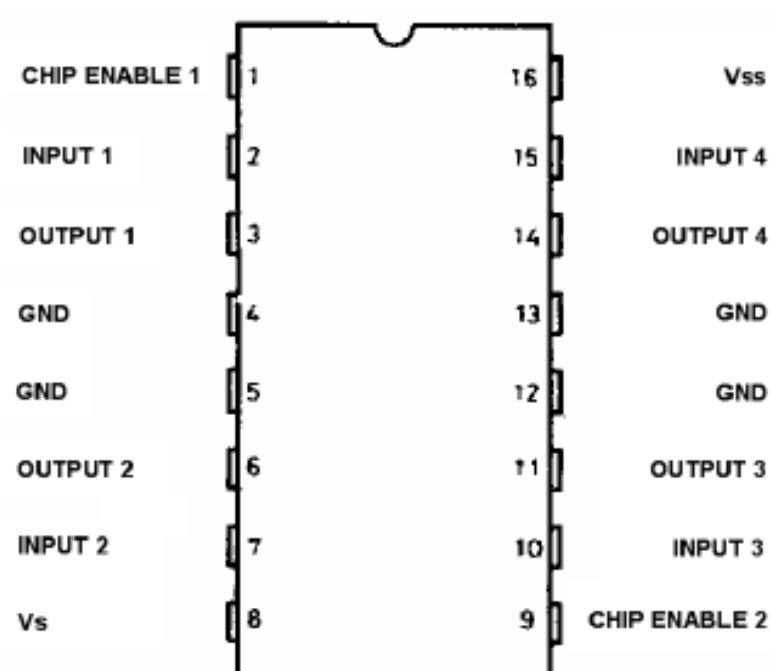


Figura 76 Distribuciones de pines del integrado L293D

Cuadro 16

Valores máximos de entrada para el driver L293D

Símbolo	Significado	Valor máximo
Vs	Fuente de alimentación (motores)	36 [V]
Vss	Fuente de alimentación de la lógica	36 [V]
Vi	Voltaje de entrada	7 [V]
Vinh	Voltaje de habilitación	7 [V]
Iout	Corriente pico de salida	2 [A]
Ptot	Disipación de potencia	5 [W]

Cuadro 17 Función

Función de los terminales del driver L293D

Terminal	Función
1	Habilitación del puente 1
2	Entrada del amplificador 1
3	Salida del amplificador 1
4	GND
5	GND
6	Salida del amplificador 2
7	Entrada del amplificador 2
8	Fuente de alimentación (motores)
9	Habilitación del puente 2
10	Entrada del amplificador 3
11	Salida del amplificador 3
12	GND
13	GND
14	Salida del amplificador 4
15	Entrada del amplificador 4
16	Fuente de alimentación (lógica)

6.3. Placa Electrónica

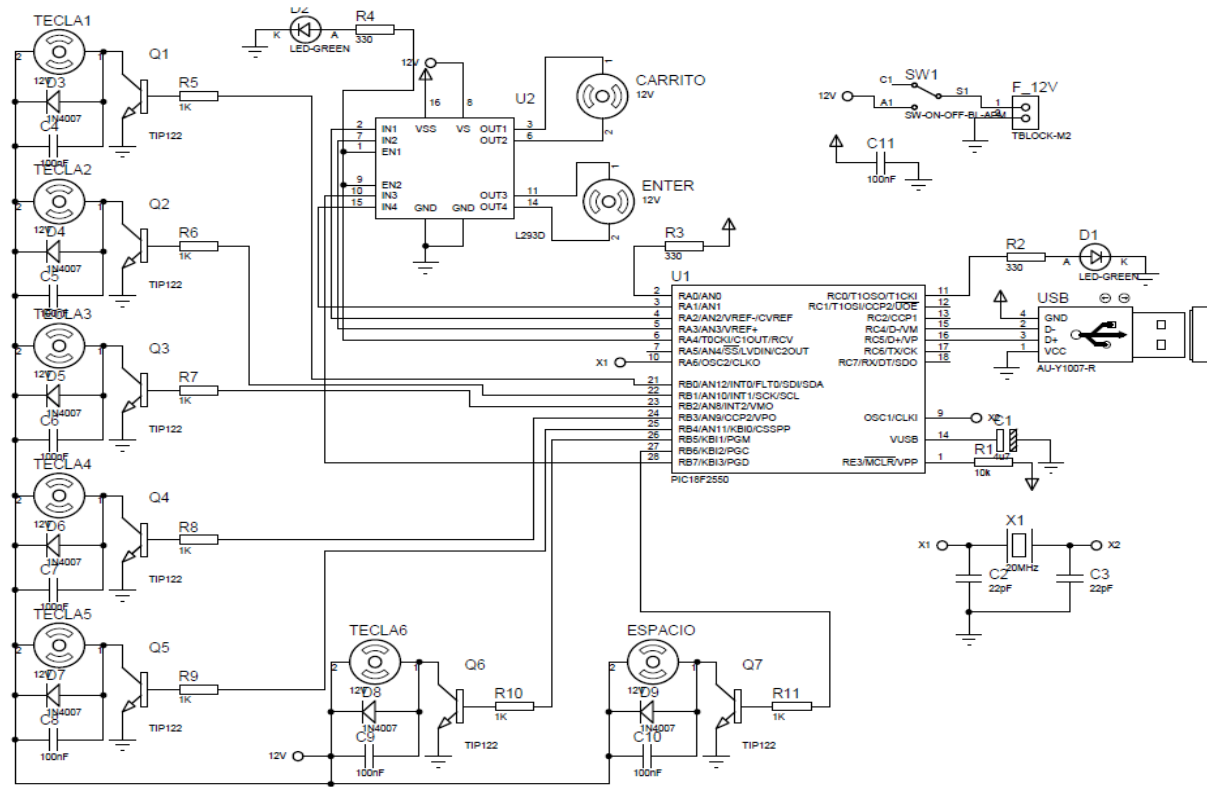


Figura 77 Diseño

Placa Electrónica en el simulador Proteus

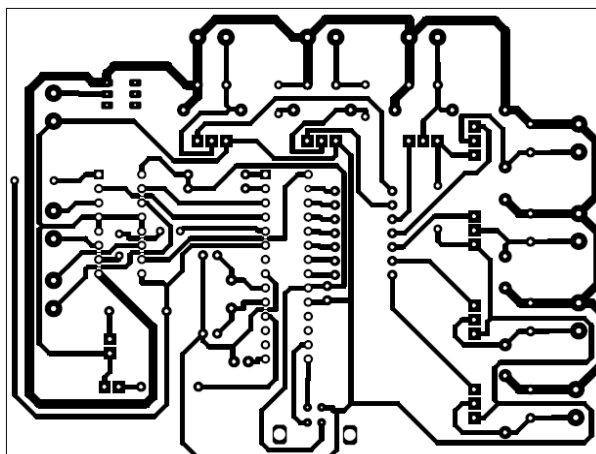


Figura 78 Diseño de la placa electrónica en ARES para el circuito impreso

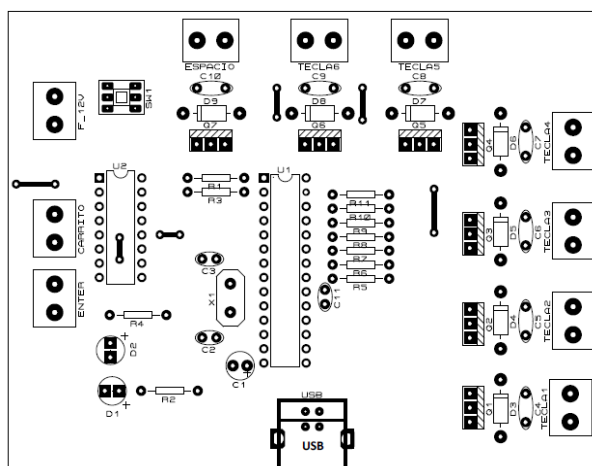


Figura 79 Diseño de la placa electrónica, circuito impreso

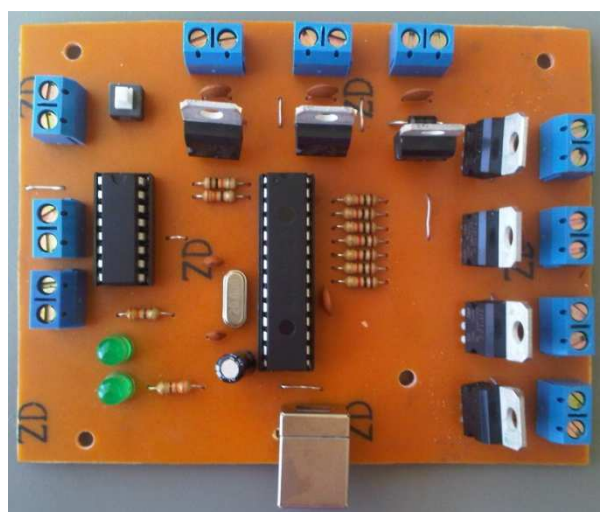


Figura 80 Circuito Impreso

CAPÍTULO 4

DISEÑO DE SOFTWARE

4.1 Desarrollo del código fuente

4.1.1 Programación del CCS Compiler para el microcontrolador 18f2550

- **Encabezado**

En el encabezado deben estar registrados algunos puntos importantes como el tipo de microcontrolador a usar, el convertidor análogo/digital, los fuses que establecen los distintos bits de configuración del microcontrolador, especificar la velocidad del oscilador, activación de transferencia y recepción de datos masivos del USB, también deben estar los drivers para el microcontrolador

- **Declaración de variables**

Para empezar con la declaración de las variables, se debe tomar en cuenta que:

- `const int8 Lenbuf = 32` : es la longitud del buffer de lectura en el puerto USB
- `int8 recbuf(Lenbuf)` : se declara el buffer que recibe el dato para almacenarlo
- `int8 envio[1]` : se declara variables enteras de 8 bits
- `int8 i=0` : se declara la variable i, de 8 bits

- **Void Main**

El código fuente principal, empieza generando una señal de salida en alto para indicar que la conexión del microcontrolador esta correcta, luego se realiza la inicialización de la conexión USB que va a tener con el PIC 18f2550. Una vez generadas las configuraciones iniciales, se debe programar la recepción de cada una de las letras del alfabeto Braille. El microcontrolador recibe un byte de lectura, este dato enviado es el número entero de cada letra, codifica esta señal y envía a la salida

los pulsos necesarios para que se activen los actuadores. Las líneas de programación son repetitivas, espera recibir el número entero enviado desde el interfaz gráfico, luego compara dicho valor con cada uno de los números que tiene almacenados en el buffer de recepción del dato y envía el número binario correspondiente que representa las letras del Alfabeto Braille, y se activan los actuadores.

El microcontrolador envía una señal al interfaz gráfico indicando si tiene conexión física entre el la interfaz gráfica y microcontrolador, para esto el microcontrolador recibe una señal analógica, luego le envía ésta señal a java, y aquí se indica al usuario el estado de la conexión de USB.

El siguiente diagrama muestra la secuencia que se usó para la programación efectuada en el programa CCS Compiler para el microcontrolador 18f2550.

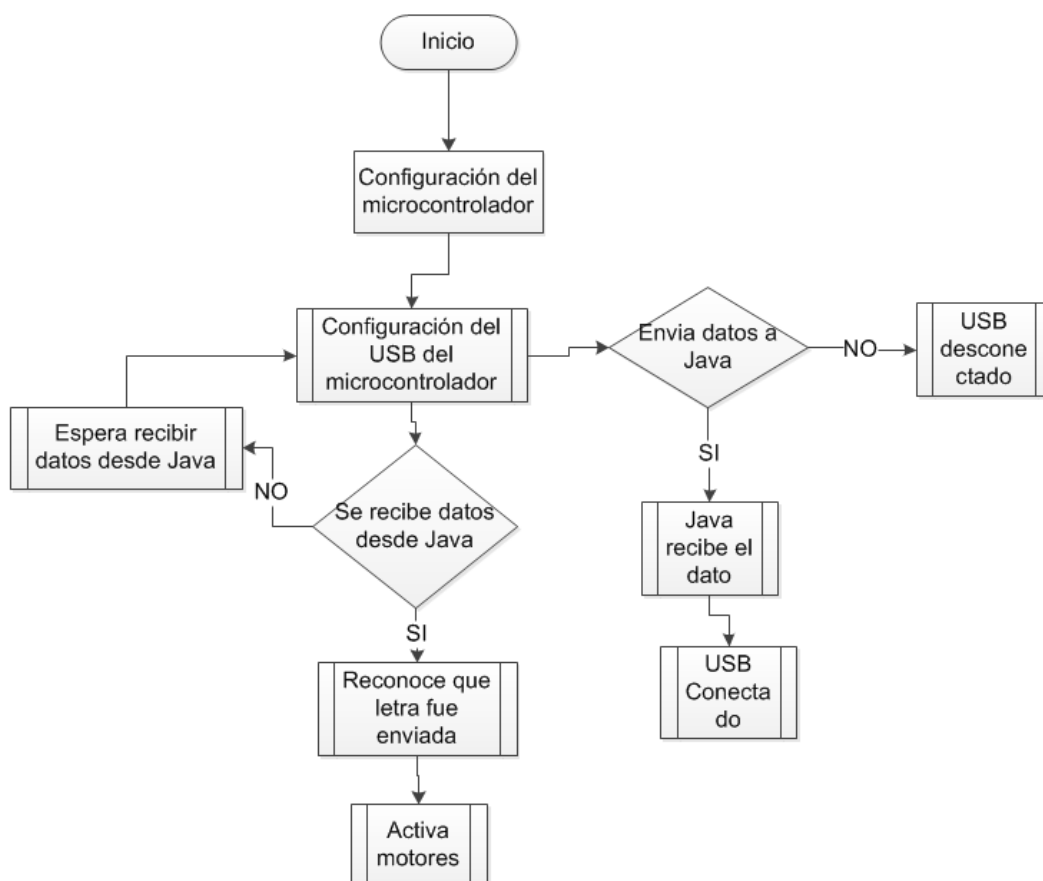


Figura 81 Diagrama de flujo del programa del microcontrolador

4.1.2 Líneas de programación usando código fuente Java en el entorno de NetBeans

Para la facilitar la líneas del código fuente en este programa fue necesario separar dicho código por paquetes llamados “Controladores” y “Principal”, cada uno de éstos contiene sus respectivas clases, cumpliendo cada uno con una función independientemente de las otras pero guardando una relación entre sí que permite integrar todo el proyecto, tal como muestra la siguiente figura.

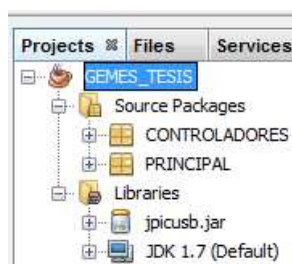


Figura 82 Paquetes existentes en el proyecto.

Controladores: contiene a las clases que cumplen con la función de controlar y realizar la codificación del alfabeto latino básico a alfabeto Braille, a su vez contiene también las clases que realizan la comunicación USB entre el PC y el microcontrolador y viceversa.

Principal: cumple con la integración de todas las clases y métodos para ejecutar al proyecto y de esta manera pueda ser operado.

- **Librería JPicUSB**

Esta librería ayuda a establecer la comunicación entre el computador y el microcontrolador 18f2550 por medio del puerto serial USB, tiene la capacidad de implementar a todas las funciones de la API USB de Microchip, al momento en que se menciona esta clase Java realiza una llamada hacia una librería dinámica que es parte de NetBeans, en este caso hace el llamado de la librería jpicusb.dll que es el que facilita el camino para tener una comunicación óptima entre las dos interfaces mediante el cable USB.

Es necesario agregar esta librería al proyecto, de la misma manera los componentes tanto el `jpibusb.jar` que es la interfaz java con la librería dinámica `jpibusb.dll`, como el `javadoc` que contiene toda la documentación de la clase y se encarga de describir los métodos.

- **Paquete Controladores**

Como ya se menciona anteriormente, es aquí en donde se realiza la conversión del alfabeto latino al alfabeto Braille, además se desarrolla la programación para que exista una comunicación USB utilizando la librería `JPicUSB`, permitiendo tanto el envío como el recibo de información.

Clase Controlador_decodificador

Es esta clase la que cumple con la función específica de convertir el alfabeto latino en el alfabeto Braille en cualquier momento en que el usuario ingrese texto. El método `decodifica(char caracter)` permite obtener el número `ascii` de cada una de las letras ingresadas y a su vez retorna una variable `int` que se obtiene de la codificación propia de las letras en Alfabeto Braille transformadas en código binario tomando en cuenta el Código Generador respectivo de cada letra.

Mediante la siguiente figura se explicara gráficamente la codificación propia de todo el Alfabeto Braille transformado en Código Binario.

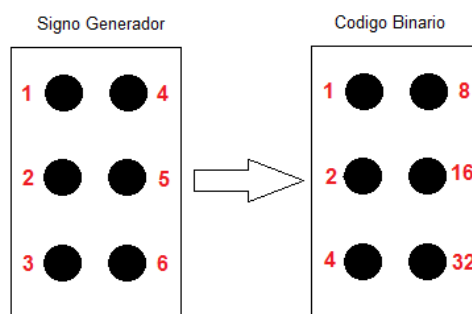


Figura 83 Asignación del Signo Generador al Código Braille

En las tablas está la codificación propia del Alfabeto Braille al Código Binario, como se muestra a continuación:

Tabla 3

Alfabeto Braille representado en Código Braille

LETRA AS Y NÚMEROS	LETRA BRAILLE EN NÚMERO DECIMAL	LETRA BRAILLE EN CÓDIGO BINARIO
a, 1	1	0000001
b, 2	3	0000011
c, 3	9	00001001
d, 4	25	00011001
e, 5	17	00010001
f, 6	11	00001011
g, 7	27	00011011
h, 8	19	00010011
i, 9	10	00001010
j, 0	26	00011010
k	5	00000101
l	7	00000111
m	13	00001101
n	29	00011101
o	21	00010100
p	15	00001111
q	31	00011111
r	23	00010111
s	14	00001110
t	30	00011110
u	37	00100101
v	39	00100111
w	58	00111010
x	45	00101101
y	61	00111101
z	53	00110101

Tabla 4

Caracteres Especiales representados en Código Binario

CARACTERES ESPECIALES	CARACTERES BRAILLE EN NÚMEROS DECIMAL	CARACTERES BRAILLE EN CÓDIGO BINARIO
Espacio	64	01000000
¡	22	00010110
“	38	00100110
&	47	00101111
(35	00100011
)	28	00011100
*	20	00010100
+	22	00010110
,	2	00000010

CONTINUA →

-	36	00100100
.	4	00000100
/	50	00110010
:	18	00010010
;	6	00000110
<	42	00101010
=	54	00110110
>	21	00010101
¿?	34	00100010
@	16	00010000
[55	00110111
]	62	00111110
	56	00111000

Por ejemplo si la letra ingresada es la f, el valor que retorna es el número 11 el mismo que se representa en número binario con 00001011 y es éste número el que representa a los actuadores que se deben activar para que la letra sea representada en código Braille.

A continuación se muestra el diagrama de flujo que se uso en la programación de la clase controlador_decodificador.

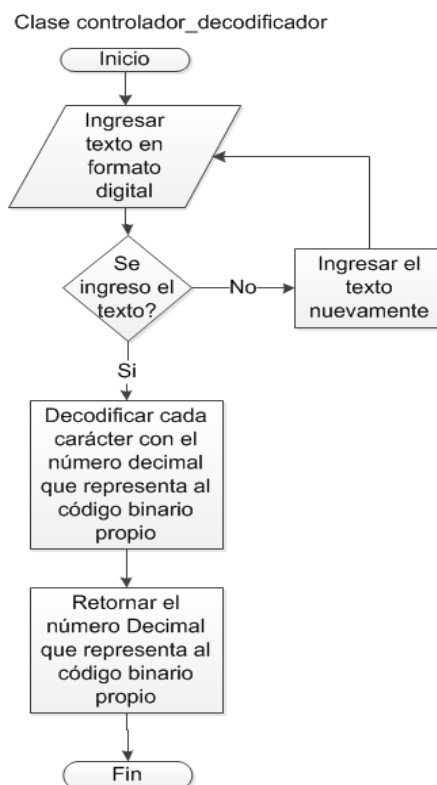


Figura 84 Diagrama de flujo de la clase controlador_decodificador.

Clase Controlador_principal

En esta clase se toma a toda la cadena de caracteres que se desea escribir en alfabeto Braille y la separa en caracteres individuales para reconocerlos uno a uno, si son letras mayúsculas, minúsculas, números y caracteres especiales, una vez que los reconoce los codifica y envía el dato hacia el microcontrolador 18f2550 mediante el cable USB.

Se trabaja con hilos o Thread debido ya que se necesita enviar continuamente la información y datos hacia el microcontrolador, de esta manera mientras el programa se opera, al mismo tiempo envía los datos.

La programación de esta clase ha sido separada en cuatro partes, la primera parte se refiere a las letras en mayúscula, la siguiente es para las letras minúsculas, la tercera es para los números y finalmente se toman en cuenta los caracteres especiales.

La lógica de programación es la misma para cada uno de éstos casos ya que se empieza comparando a cada letra con el correspondiente número ascii, y es aquí en donde las letras minúsculas se encuentran en el rango que van desde el 97 al 122, las letras mayúsculas van desde el 65 al 90, los números están dentro del rango que van desde el 48 al 57 y por último los caracteres especiales que se encuentran entre 32 al 47, lo que se hace es enviar el número ascii de la letra ingresada hacia la clase controlador_decodificador que es la que identifica a cada una de las letras del alfabeto latino y las codifica en una codificación propia del alfabeto Braille en código binario y éste retornara el número entero que representa al número binario.

Cabe recalcar que a las letras mayúsculas y a los números se les debe anteponer un prefijo que representa si la letra es mayúscula o número.

Es en este método también que se integra a la clase envio_usb, la misma que establece la conexión entre el computador y el microcontrolador 18f2550 mediante la sentencia pic_salida(salida) que es el que envía la información adquirida.

Se realiza la contabilización de los caracteres que van a ser impresos en la hoja tamaño A4 ya que se ha limitado a que la hoja tenga 30 caracteres, de la misma manera se contabiliza el número de saltos de línea para que sean 27, cuando la impresión ha llenado en su totalidad a la hoja, el programa tiene una pausa con el tiempo suficiente para cambiarla de forma manual.

A continuación se muestra el diagrama de flujo que se uso en la programación de la clase controlador_principal.

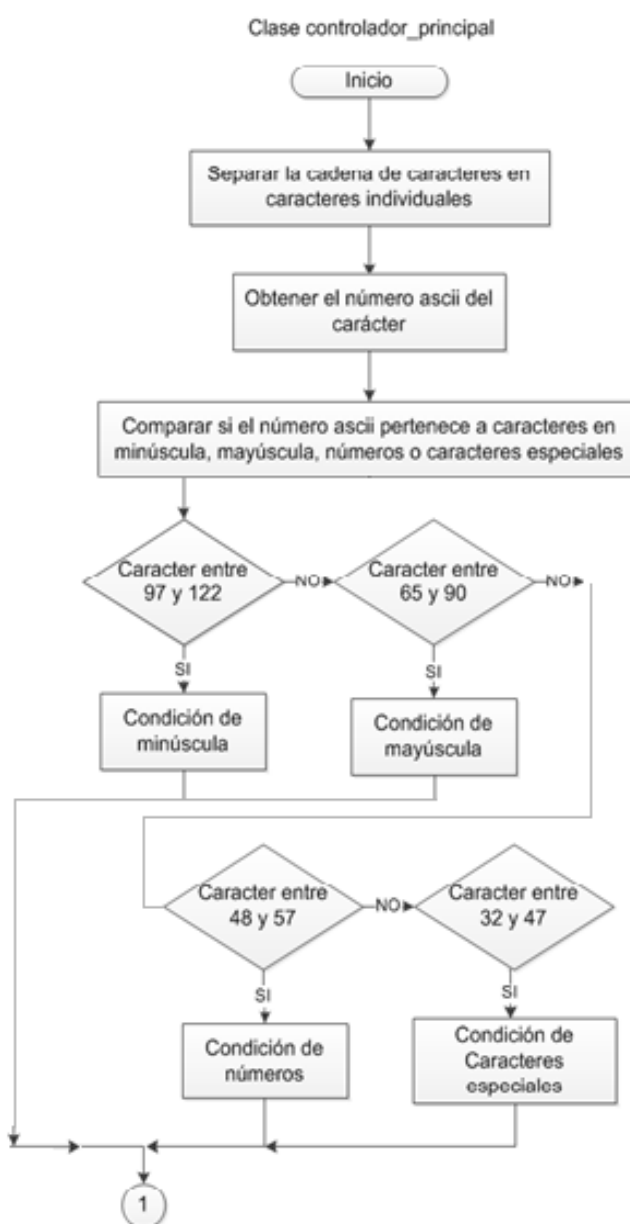


Figura 85 Diagrama de flujo de la clase controlador_principal.

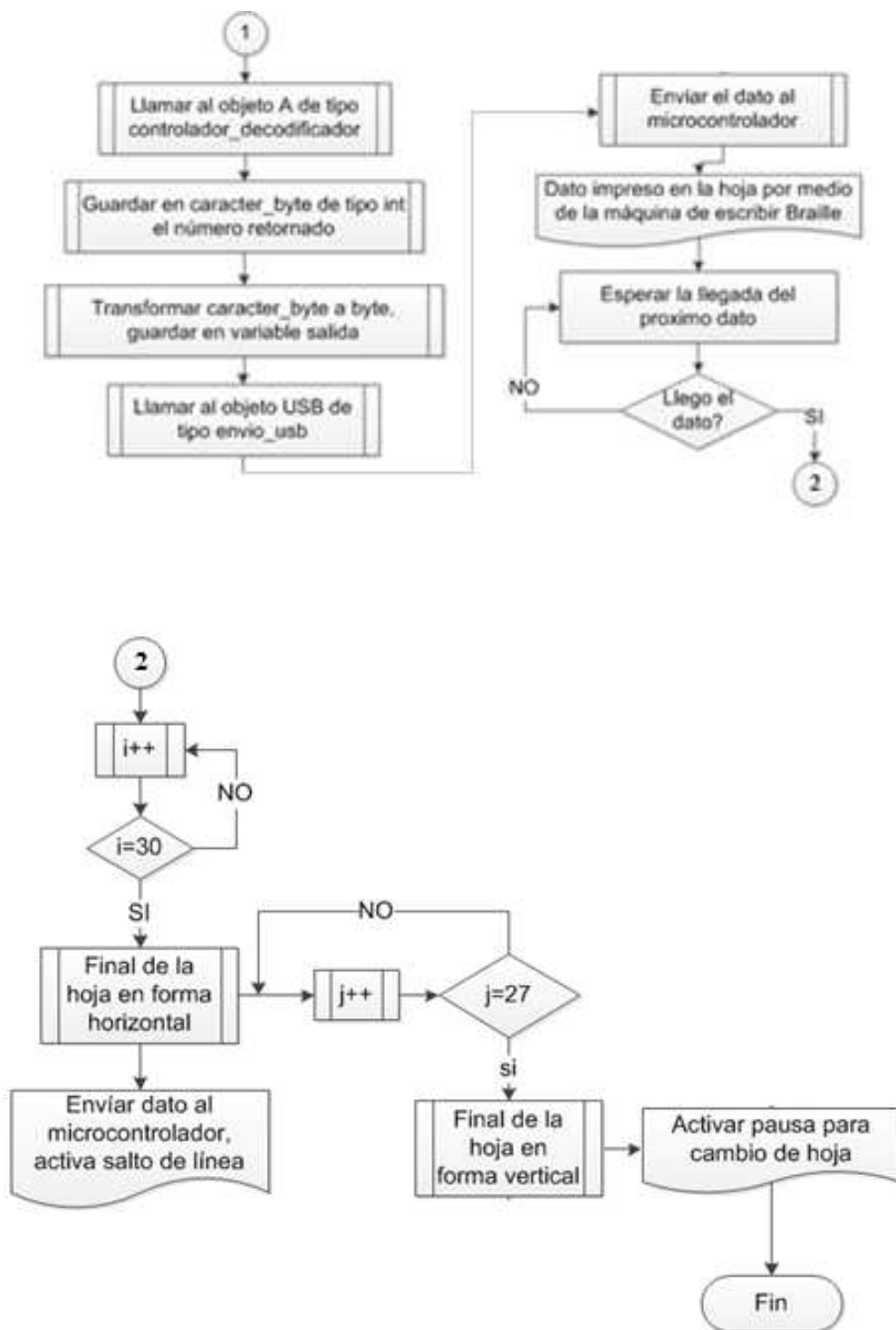


Figura 86 Diagrama de flujo de la clase controlador_principal.

Clase Envio_usb

El objetivo principal de esta clase es realizar la conexión USB entre el computador mediante el entorno de NetBeans y Java con el microcontrolador 18f2550, básicamente en esta clase se abre el canal USB, por así decirlo, que permitirá la comunicación e intercambio de datos.

Se tiene dos métodos llamados “método” y “pic_salida” , con esto se establece la comunicación del USB entre el microcontrolador y la interfaz gráfica.

Todo está programado bajo hilos, de esta manera se envía y se recibe información todo el tiempo, y en el caso de haber algún error en el envío y recepción de datos, éste será informado oportunamente.

Es importante establecer el VID y PID al momento de enviar datos hacia el microcontrolador, ya que con estos parámetros se establece el canal de comunicación al llamar todos los componentes de la librería JPicUsb, se configura la librería utilizando sus propios métodos de inicialización por defecto.

Los datos enviados son de tipo byte, se almacenan en un arreglo para luego ser enviados hacia el microcontrolador.

Se utiliza el método rápido QWrite para poder establecer una conexión al vid&pid por defecto, iface.QWrite(out, 1, 1000), el primer parámetro representa al arreglo en bytes enviados, el segundo parámetro es la longitud del dato en bytes y el tercer parámetro el tiempo con el que se van a enviar los datos en milisegundos.

Una vez que la conexión se ha establecido el método retorna el número de bytes que se han escrito con éxito.

A continuación se muestra el diagrama de flujo que se usó en la programación de la clase envio_usb.

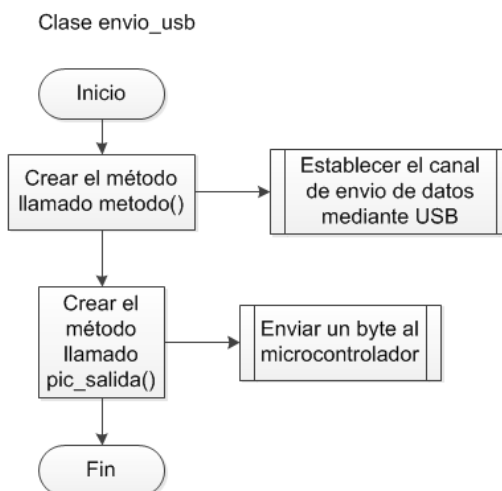


Figura.87 Diagrama de flujo de la clase envio_usb

Clase Recibo_usb

La clase recibo_usb, tiene el objetivo de recibir todos los datos que el microcontrolador envía, de ésta manera puede controlar las diferentes funciones que se realizan en el sistema al recibir cualquier dato desde el microcontrolador.

Básicamente con ésta clase se puede conocer si el cable USB está correctamente conectado o no en el sistema, ya que el cable al momento de ser conectado a la computadora envía al microcontrolador 5 V DC, y el microcontrolador envía esta señal al método recibo_usb, y éste informa el estado físico del cable USB usuario. Esta clase trabaja bajo hilos, por lo tanto enviará y recibirá datos constantemente, si existiese algún error informará al usuario la falla. Se debe llamar al `iface.set_vidpid("vid_04d8&pid_000b")`, para poder establecer el VID y PID por defecto para poder establecer el canal de comunicación cuando se llama a la librería JPicUsb puesto que utilizan sus propios métodos de inicialización (`set_vid&pid`, `set_instance`).

Los datos son recibidos en un arreglo de bytes, y utilizando el método rápido `iface.QRead(2,500)` se recibirá el valor de 2 datos de 500 bytes cada uno. A los datos recibidos en el arreglo de bytes, se deben convertir en una cadena de codificación utf-8 y son almacenados, UTF-8 significa 8 Bit Unicode Transformation Format, que es un formato de codificación de caracteres Unicode.

El estándar Unicode asigna un código a cada caracter, los tamaños de cada dato utilizan 8 bits, y lo más importante es que codifica caracteres del Código ASCII, el microcontrolador envía el número entero 120, este es decodificado como un número del código ASCII. Con este código se determina si el dato fue enviado y cumplir con las instrucciones asignadas. En este caso debe indicar si está conectado el USB o no. A continuación se muestra el diagrama de flujo que se uso en la programación de la clase `recibo_usb`.

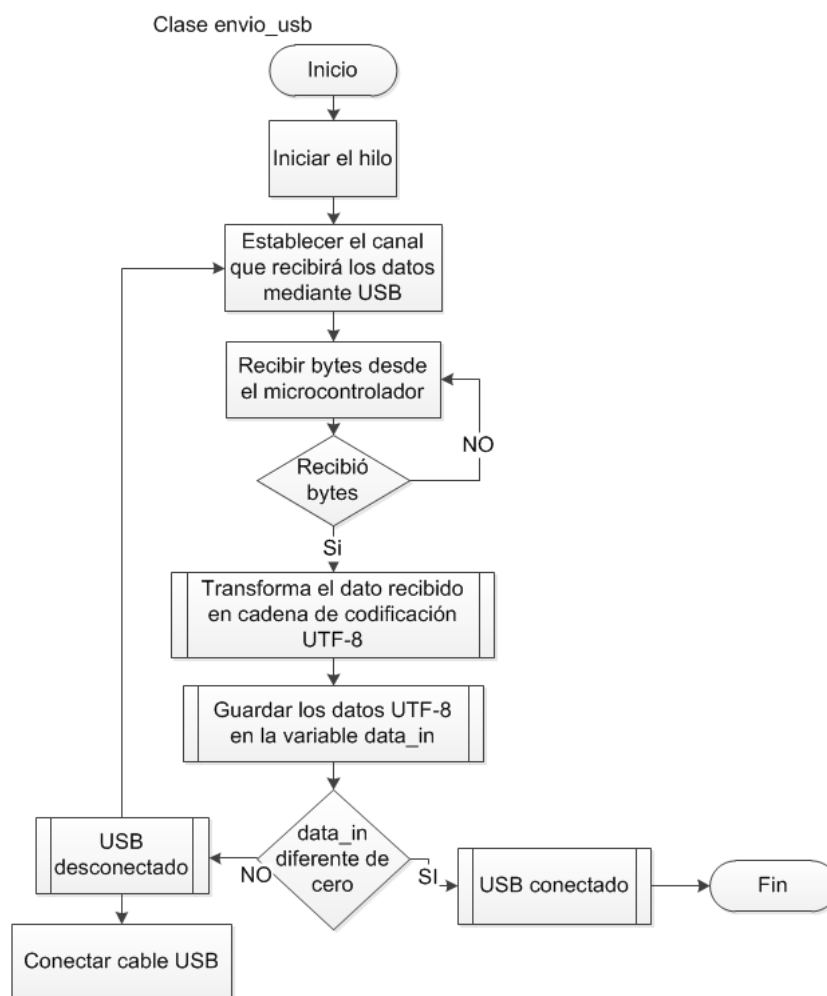


Figura 88 Diagrama de flujo de la clase `recibo_usb`

- **Paquete Principal**

En el paquete principal se han desarrollado dos clases.

La una que se llama `gemes_tesis` y la otra que se llama `principal` y tienen la funcionalidad de integrar a todas las clases en un solo proyecto para que puede ejecutarse y ser operado.

A continuación se muestra el diagrama de flujo que se usó en la programación de la clase `Principal`.

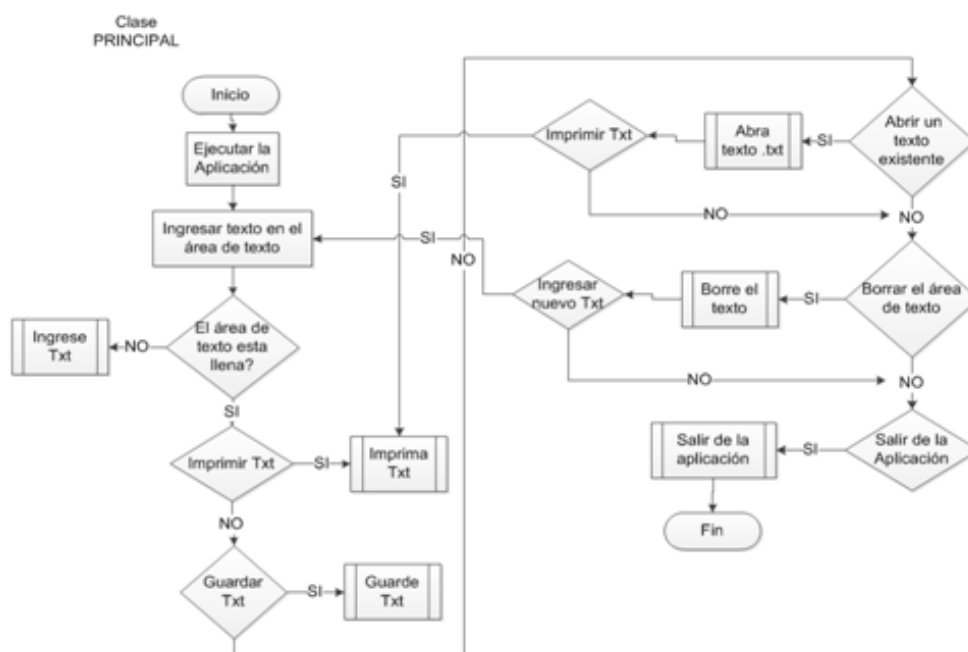


Figura 89 Diagrama de flujo

Clase `Gemes_tesis`

En esta clase se encuentra el método `void main` el mismo que ejecuta al proyecto usando los métodos propios de esta clase, tales como `setLocationRelative(null)` que hace que la ventana principal se ejecute siempre en un mismo lugar en toda la pantalla de la computadora, y también se tiene al método `setVisible(true)` que hace que la ventana principal se muetsre visible cada vez que se la ejecute.

A continuación se muestra el diagrama de flujo que se uso en la programación de la clase Gemes_tesis.

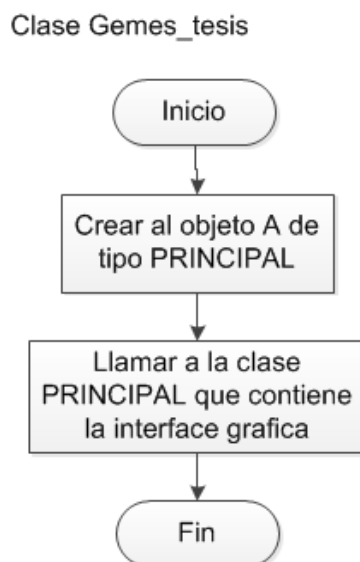


Figura 90 Diagrama de Flujo de la clase Gemes_tesis

Clase Principal

En esta clase se ha desarrollado la parte gráfica del programa, es la más importante ya que es aquí en donde se desarrolla la parte operativa que va a ser manejada por el usuario.

El usuario interactúa con el proyecto para escribir, copiar, abrir, entre otros cualquier documento que se encuentre en un formato digital para convertirlo en alfabeto Braille y de esta manera envíe a imprimir.

Como se indica en la figura, la interfaz gráfica es muy sencilla, para que de ésta manera le sea fácil interpretar al usuario el nuevo entorno gráfico del proyecto.

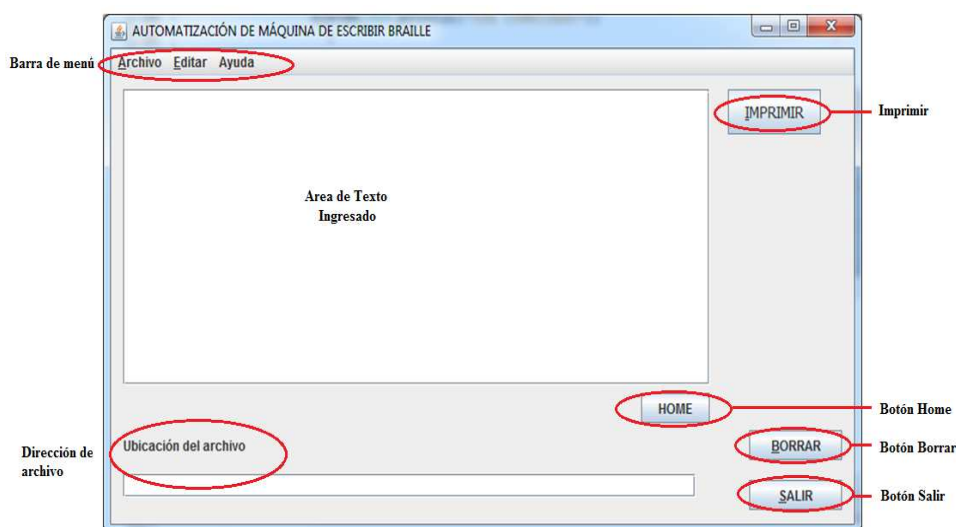


Figura 91 Interfaz gráfica del proyecto

La interfaz se compone de tres botones básicos, cumplen con la función de imprimir el texto en alfabeto Braille, de borrar todo texto que se encuentre en el área de texto y el tercer botón cumple con la función de salir del sistema.

Tiene una barra de menú, cada menú contiene varias opciones que ayudan a que el usuario interactúe de forma fácil dentro del sistema y sea capaz de abrir, guardar, copiar, entre otros, un archivo y a su vez lo envíe a imprimir.



Figura 92 Barra de menús de la interfaz gráfica del proyecto

4.2. Diseño de la interfaz grafica

Una Interfaz Gráfica se la conoce con el nombre de Graphical User Interfaz (GUI) por sus siglas en ingles, donde esta un conjunto de componentes gráficos que interactúan con el usuario y el proceso de aplicación. Éste conjunto de componentes gráficos está compuesto por botones, ventanas emergentes, cuadros de texto, cuadros de dialogo, entre otros.

Hay que tener en cuenta que se debe diseñar una aplicación sencilla que sea fácil de interpretar por el usuario, luego se debe programar las diferentes instrucciones de programación necesarias para ejecutar correctamente la interfaz.

La interfaz gráfica se diseña sobre un frame, este frame será el contenedor principal de toda la GUI donde estarán los componentes gráficos.

Los componentes de una GUI son objetos de las clases que heredan de la clase base, lo mas importantes son JButton, JLabel, JTextField, JCheckBox, JRadioButton, entre otras.

4.2.1. Barra de menú

En toda interfaz grafica es muy importante tener una barra menú que permitirá que el usuario acceda a las opciones que sean de mayor importancia dentro del proyecto, en este caso se ha incorporado tres menús: Archivo, Editar y Ayuda.

- **Archivo**

Este pertenece al primer ítem del menú del proyecto desarrollado, se han considerado las opciones tales como “Abrir”, “Guardar”, “Imprimir”, “Salir”, las mismas que el usuario puede tener acceso con solo hacer un clic sobre cualquiera de estas.

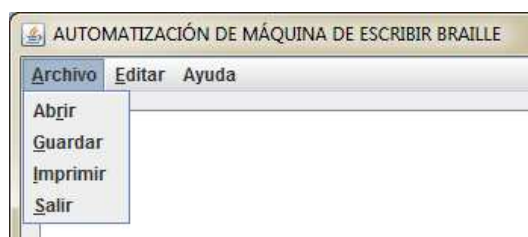


Figura 93 Contenido del Menú “Archivo”

Abrir

Este es un submenú del menú “Archivo”, como el nombre lo indica el objetivo principal es ofrecer la posibilidad de abrir cualquier documento que tenga como

extensión *.txt, la programación de este submenú se ha dividido en dos partes, la primera parte permite abrir una ventana de dialogo en donde es posible buscar el archivo o documento que se desea abrir, y la segunda parte es la que permite abrir un buffer o canal para leer el archivo que se desea abrir.

Para crear la ventana de dialogo y explorar los archivos que se desean abrir se ha usado el método `FileNameExtensionFilter` que permite filtrar a la ventana de dialogo para abrir solo archivos que tengan una extensión *.txt.

También se usa el método `JFileChooser` que permitirá escoger el archivo que se desee abrir dentro de la ventana de dialogo.

Una vez creada la ventana de dialogo, es ésta la que permite explorarla para seleccionar el archivo que se desea abrir es necesario abrir el camino o el buffer que leerá al archivo.

Para esto se usa el método `FileReader` que es el que va a leer al archivo ya que contiene la ruta del archivo seleccionado. El método `BufferedReader` es el que va a escribir o colocar a este archivo en el cuadro de texto.

Todas las funciones que se han usado para crear la ventana de dialogo que permite abrir archivos y documentos con extensión *.txt son parte de la librería `javax.swing`, por esta razón se debe importar la librería: `javax.swing.filechooser.FileNameExtensionFilter`.

La siguiente figura muestra a la ventana de dialogo que mediante una exploración sobre ella permite abrir el archivo deseado, siempre con extensión *.txt.

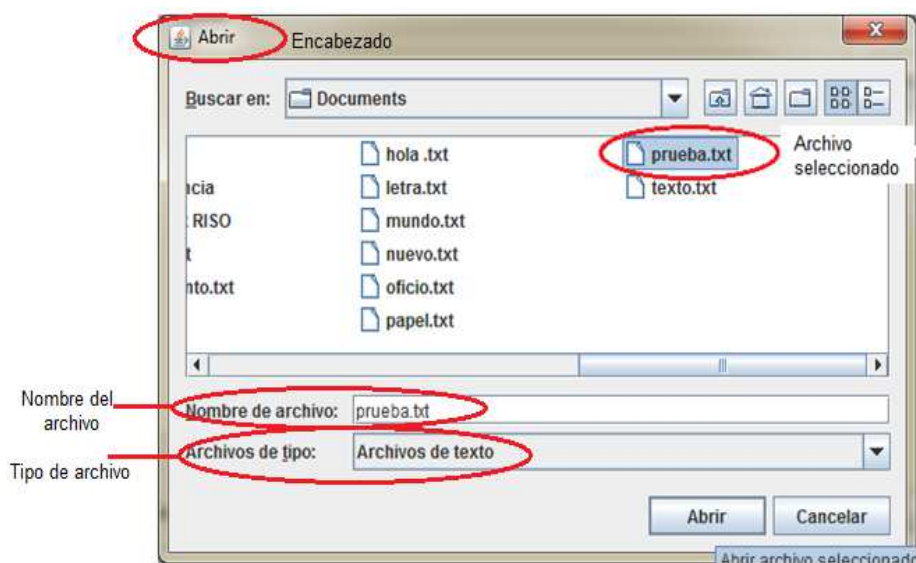


Figura 94 Ventana de Dialogo que permite abrir un documento

La siguiente figura muestra a la interfaz gráfica del proyecto con el area de texto totalmente llena con el contenido del archivo seleccionado previamente y abierto.

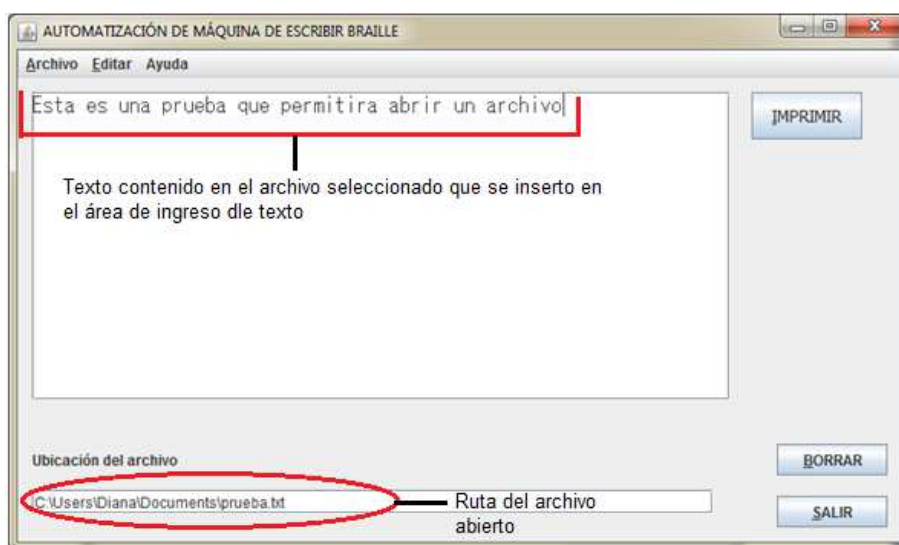


Figura 95 Archivo abierto.

También se a considerado la posibilidad de advertir al usuario sobre posibles errores al momento de abrir algún archivo, es por esto que se han introducido ventanas emergentes que indican dicho error, tal como indica la siguiente figura:



Figura 96 Ventana emergente que indica un error

Guardar

Este menú permite guardar todo texto que haya sido escrito en el área de texto de la interfaz gráfica del proyecto, para esto es necesario abrir una ventana de dialogo que permita explorar las carpetas en donde se desea guardar el archivo de tipo *.txt. Se usa también un filtro para guardar texto con extensión *.txt, siempre y cuando el área de texto no se encuentre vacía.

Se crea una ventana de dialogo para explorar la carpeta en donde se desea guardar el archivo, para esto se usa el método `.getSelectedFile` de la clase `fileChooser` se obtiene la ruta del archivo. Para acceder a toda la información que se encuentra en el área de texto se usa el método `PrintWriter`. Una vez que se ejecute el proyecto, el menú de Guardar se visualiza de la siguiente forma, tal como se muestra en la figura.

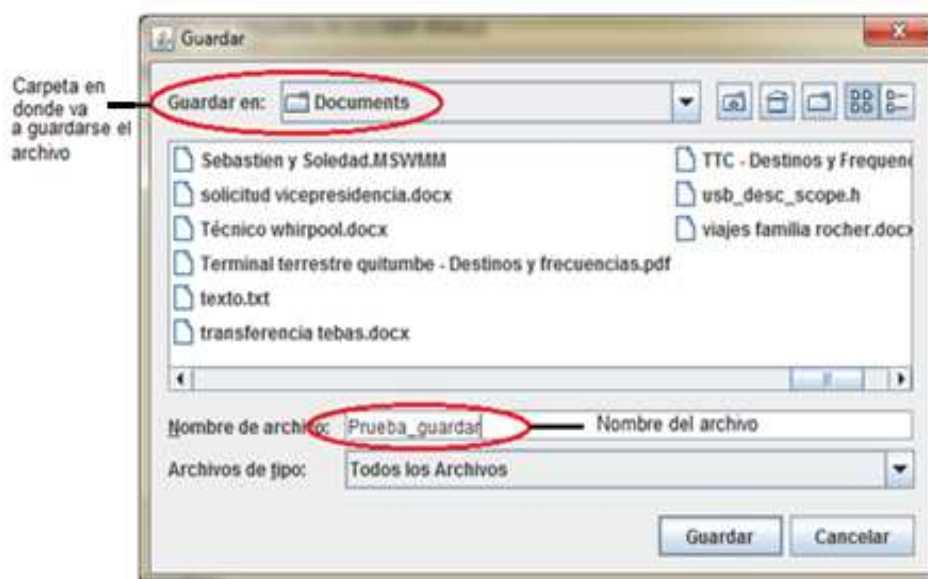


Figura 97 Ventana de dialogo menú Guardar

De igual manera se crearon ventanas emergentes que indicarán si existe algún error al momento de guardar el archivo, de la misma manera indicarán si el archivo se guardo exitosamente, tal como se muestra en la siguiente figura:



Figura 98 Ventana de dialogo que muestra advertencia

Imprimir

Este submenú cumple con la función principal del proyecto: imprimir. Es aquí donde todas las instrucciones de programación se ejecutan, cada uno de los métodos creados anteriormente de forma separada se unifican para ejecutarse todos en conjunto, y de esta manera, cumplir con el objetivo principal de imprimir el texto en el Alfabeto Braille.

Cuando se ejecuta la opción imprimir del menú, aparece una ventana emergente que le pregunta al usuario si desea imprimir, éste mensaje es por seguridad, ya que se puede ejecutar dicha acción por accidente, de esta manera el usuario puede cancelarla si lo desea.

Mientras se está imprimiendo el texto, se debe restringir al área de ingreso de texto, con el fin de evitar que se sobre escriba algún texto sobre él, ocasionando problemas de impresión, perdida de datos, perdida de comunicación y sincronización con el texto actual, esto se puede hacer usando el método disable, mostrando un mensaje de Advertencia!!! el documento se está imprimiendo.

A continuación se indicará gráficamente la ejecución de la acción Imprimir

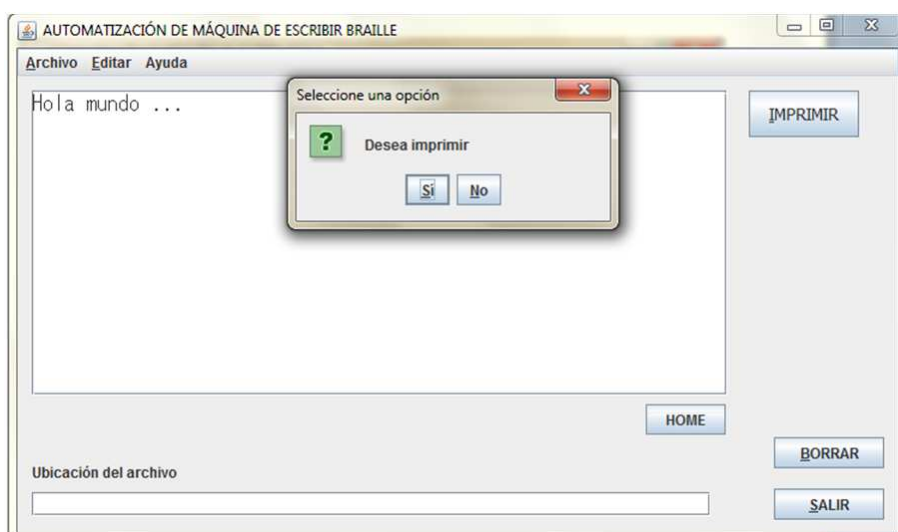


Figura 99 Ventana emergente, da opciones a imprimir

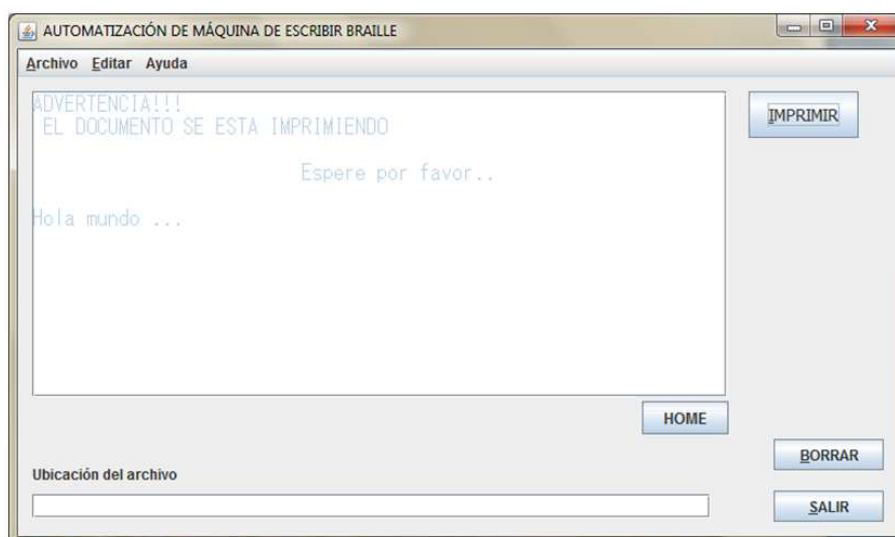


Figura 100 Interfaz, el documento se está imprimiendo

Salir

Permite cerrar el programa al momento en que el usuario quisiera mientras se ejecuta. Se crea una ventana emergente que pregunta al usuario si realmente desea cerrar el programa, si ha presionado la opción no, la ventana emergente se cierra la interfaz grafica se la puede usar normalmente y si se ha presionado la opción si, el proyecto simplemente se cierra con la sentencia `System.exit(0);`.

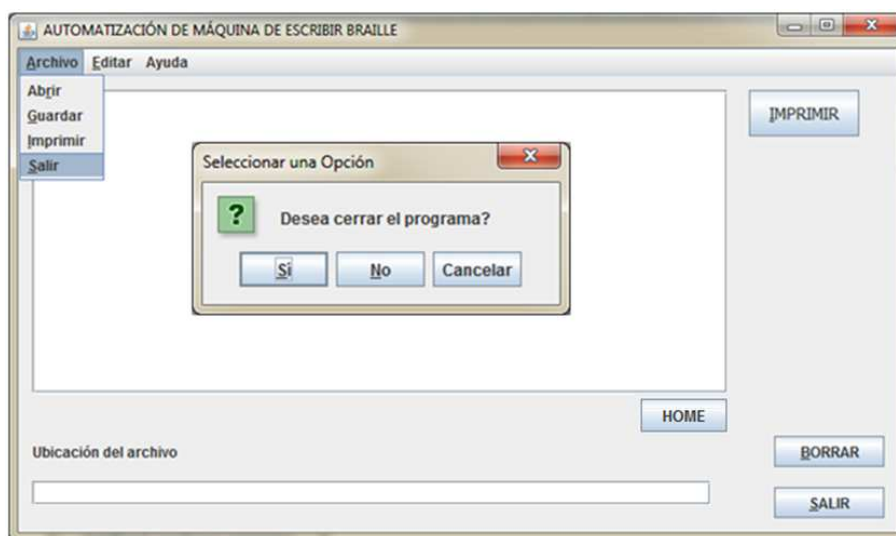


Figura 101 Ventaja emergente que indica si se desea cerrar el programa.

- **Editar**

El segundo menú de la barra se trata del Editar, donde se encuentran las opciones de Copiar, Cortar y Pegar a cualquier parte del texto.

La siguiente figura muestra el menú Editar.

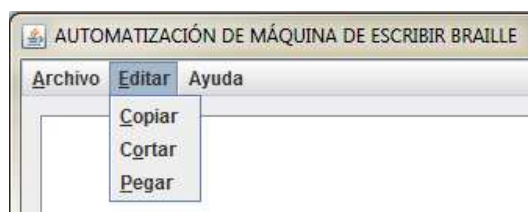


Figura 102 Menú “Editar”

Copiar

Esta opción permite copiar cualquier texto que este dentro del área de texto así como también de alguna fuente externa, todo depende de las necesidades del usuario. Se utilizó el método copy de la clase javax.swing.

Cortar

Esta opción permite cortar el texto que se encuentra en área de texto, de la misma manera que el menú copiar, también se puede cortar el texto de una fuente externa, todo depende de las necesidades del usuario. Se utilizó el método `cut` de la clase `javax.swing`.

Pegar

Esta opción permite pegar el texto ya sea que se haya copiado o cortado del área de texto o alguna fuente externa. Se utilizó el método `paste` de la clase `javax.swing`.

- **Área de ingreso del texto**

El área de ingreso del texto es donde se debe ingresar el texto que se quiere imprimir en Alfabeto Braille. Admite el ingreso de cualquier texto, sea por medio del teclado o algún texto que se encuentre guardado en algún dispositivo siempre y cuando sea un archivo con extensión `.txt`.

Para que el texto ingresado este alineado, se ha utilizado las funciones `setLineWrap` que realiza un salto de línea en cualquier lugar de la palabra y el `setWrapStyleWord` que realiza un salto de línea buscando espacios entre las palabras.

- **Acción de los botones**

Botón imprimir

Este botón permite imprimir el texto ingresado, realiza la misma función que realiza el sub-menú imprimir.

Botón borrar

Este botón borra todo el texto que se encuentra en el área de texto, sea cual sea la necesidad del usuario, al presionar este botón esta acción será realizada. Se utilizó el

método el método null, lo que hace nulo al texto y luego se habilitará el cuadro de texto para poder introducir nuevamente texto.

Botón salir

El botón salir permite abandonar la aplicación, para esto se utiliza la función exit junto con el método System.out(0), y el programa se cerrará.

Antes de cerrarse todo el programa por completo, aparecerá una ventana emergente en la cual le preguntará al usuario si desea salir de programa, de esta manera le da la opción de cancelar el proceso si el caso no es el de cerrar el programa.

Botón home

El botón Home, es un botón diseñado para casos de emergencia como por ejemplo, cuando el sistema pierde energía y los actuadores que trabajan ara el salto de línea y el retorno del carro, quedan activados a la mitad del camino, y necesariamente deben regresar a su posición original para poder empezar a trabajar nuevamente. Puede haber varios puntos de emergencia que hagan que los actuadores no hayan terminado con su función, y cuando cualquier emergencia ocurra, presionando este “botón home” los actuadores irán a sus posiciones originales. Es importante que cada cierto tiempo, a pesar de no haber ocurrido cualquier tipo de emergencia, presione el botón home como medida de seguridad para el sistema.

- **Mnemónicos**

Tomando en cuenta las necesidades de aplicación en el diseño, es conveniente introducir mnemónicos en la interfaz debido a que son utilizados para activar directamente un button, checkbox, radio buttons, entre otros, sin utilizar el mouse, sino, simplemente el teclado dentro de la barra de menú o los diferentes botones que se encuentran en la aplicación.

Para agregar el mnemónico se debe añadir en la ventana de generación del código de programación la función .setMnemonic(“nombre”), y dentro del paréntesis

de debe poner la letra que se desee sea el link usando el teclado mediante la combinación de `alt+''letra''`. Por ejemplo, para el caso `imprimir_boton.setMnemonic('I')`; donde la letra *I* será el identificativo para activar el botón imprimir con el teclado. Por lo general la primera letra del nombre de cada acción, es la letra del mnemónico correspondiente, tal como lo indica el siguiente cuadro.

Cuadro 18

Asignación de Mnemónicos

ACCIÓN DEL COMPONENTE	LETRA DEL MNEMÓNICO
Imprimir_boton	I
Borrar_boton	B
Salir_boton	S
Archivo	A
Abrir	R
Guardar	G
Imprimir	I
Salir	S
Editar	E
Copiar	C
Cortar	O
Pegar	P

- **Sonido en ventanas emergentes**

Otra característica importante que tiene la interfaz gráfica es que cada evento tiene un sonido característico, de ésta forma, el usuario va a reconocer mediante el sentido del oído, las acciones que está realizando. Por ejemplo reconocerá si está imprimiendo, o abriendo un archivo o saliendo de la aplicación.

Para poder introducir el sonido, hay que importar la librería `java.applet.AudioClip`;, reproduce un clip de sonido, éste sonido es un archivo con extensión `.wav`. Dentro de cada una de las acciones que tiene la interfaz, se agrega el sonido específico de cada uno. Las instrucciones de programación son las mismas para todos los casos, lo único que cambia es el archivo `.wav` específico.

CAPÍTULO 5

ANÁLISIS DE RESULTADOS

5.1 Parámetros de funcionamiento

Se ha decidido tomar en cuenta una serie de parámetros que van a permitir realizar las pruebas de funcionamiento, ayudando a determinar un protocolo de pruebas dentro del proyecto con la intención de obtener toda la información necesaria para alcanzar los objetivos planteados.

- Corriente: se medirá la corriente que entrega cada uno de los actuadores, de tal manera se adquirirá una fuente de poder adecuada al consumo en todo el sistema.
- Voltaje: se medirá el voltaje de todos los elementos que forman parte del proyecto, por la misma razón que se medirá a la corriente, pues es necesario conocer el voltaje necesario que se consumirá para proveerlo del voltaje necesario, teniendo un sistema en perfecto funcionamiento.
- Potencia: es muy importante este parámetro en lo que se refiere a la seguridad del proyecto, se debe medir los valores de la potencia que entrega cada uno de los actuadores dentro del sistema en funcionamiento para así tomar las medidas de seguridad necesarias, tanto para proteger a la tarjeta electrónica como al usuario.
- Tiempo: se medirá el tiempo aproximado en el que cada uno de los actuadores presionaran las respectivas teclas de la máquina de Escribir Braille, y además se tomara el tiempo con que se envíen los datos desde la computadora hasta el microcontrolador 18f2550.
- Fuerza: este parámetro es sumamente necesario medir para conocer el valor de la fuerza a la que deben ser presionadas cada una de las teclas para que el papel sea marcado sin que se rompa y los signos generadores sean legibles.

5.1.1 Instrumentos de medición

Es tan trascendental el conocimiento y estudio del uso de los instrumentos de medición en este proyecto ya que permitirá reconocer que tipo de elementos son los necesarios usar para que el funcionamiento del mismo sea correcto.

- **Multímetro:** que pueda medir la corriente y el voltaje que consume cada uno de los actuadores usados en el proyecto. El rango de este instrumento de medición deberá ser capaz de medir 5 [A] y 12 [V].
- **Balanza:** para poder medir el peso necesario para presionar las teclas de la Máquina de escribir Braille.

5.2 Pruebas realizadas en el hardware

5.2.1 Trabajo del actuador lineal

Es importante realizar una prueba que garantice e indique que el actuador lineal que se va a usar posea la suficiente fuerza para marcar el papel sin que llegue a romperlo, además que el tiempo de vida útil del actuador sea amplio para que el sistema sea funcional.

- **Marcar el papel**

Para ésta prueba fue necesario acoplar el actuador a una estructura de madera a una distancia de separación de 2 mm de la tecla con el fin de saber si tenía la suficiente fuerza para marcarlo tomando en cuenta la carrera. Se trabajó con valores comerciales de voltaje (5,10 y 12 [V]).

5 [V]: al trabajar con este valor, el actuador se activó, pero la fuerza no fue la suficiente como para marcar la tecla en el papel.



Figura 103 Actuador trabajando con 5 [V]

12 [V]: Este es el valor de voltaje ideal para que la tecla sea presionada y marque el papel sin que se rompa y que sea legible para la persona no vidente.



Figura 104 Actuador trabajando a 12 [V]

- **Resistencia**

Una vez que se comprobó que el actuador funciona sin problemas a 12 [V], es necesario comprobar la resistencia de trabajo en cuanto al tiempo, se debe tomar en cuenta que el trabajo de los actuadores es por lo menos de 30 veces hasta encontrar una pausa porque cada hoja es de tamaño A4 (21 x 29.7 [cm]), por lo tanto se marcarán 30 caracteres en cada línea, es decir un promedio de 840 caracteres por hoja.

La prueba inicialmente se trabajo con un cristal de 4 MHz, el actuador trabajo aproximadamente 240 veces hasta que llego a quemarse, esto debido a que la

velocidad con el cristal era rápida y la pausa entre activación y desactivación era muy corta.

Al tener este inconveniente, se decidió cambiar a un cristal de 20 MHz con un intervalo de trabajo de activación de 50 segundos y de desactivación de 20 segundos, lo que favoreció al actuador y pudo trabajar sin problema de recalentamiento por aproximadamente 30 minutos seguidos, llegando a marcar 1050 veces el papel.



Figura 105 Trabajo del actuador

- **Implementación de los actuadores dentro de la estructura**

Para la implementación de los actuadores lineales dentro de la estructura metálica, se tomo en cuenta la ubicación de cada una de las teclas de la máquina de escribir Braille, ya que éstas, necesitan ser presionadas de forma lineal.

A cada uno de los actuadores se les acopló un sistema de pivote para que las teclas puedan ser presionadas de tal manera que el mecanismo lineal de los actuadores no se vea interferido por problemas de ubicación física, ya que el ancho de cada actuador sobrepasa el ancho de las teclas, y no pueden ser ubicadas uno al lado del otro sin algún mecanismo extra, ya que no llegan a presionar las teclas directamente.

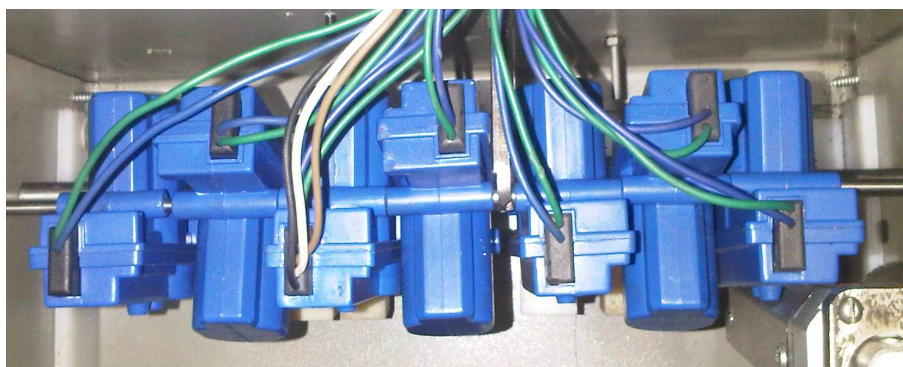


Figura. 106 Actuadores Lineales ubicados uno al lado del otro

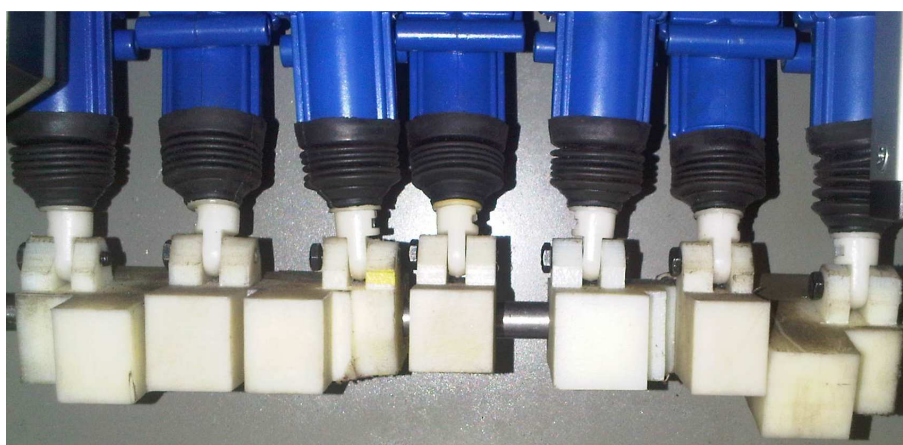


Figura. 107 Sistema de pivote acoplado a cada tecla

Como se puede observar en la figura, cada tecla tiene acoplado un sistema de pivote hecho de nylon, para que al momento en que el actuador se active, éste pivote mediante un movimiento angular, aplique la misma fuerza que está aplicando el actuador sobre cada tecla.

Así se tiene un trabajo muy optimo para presionar las teclas y éstas puedan marcar el papel sin ningún problema.

En las siguientes figuras se indican algunas letras interpretadas por los actuadores, como por ejemplo la letra q y la letra z.



Figura 108 Letra “q” interpretada por los actuadores lineales



Figura 109 Letra “z” interpretada por los actuadores

- **Trabajo del actuador lineal para el salto de línea**

La tecla del salto de línea, como se ha explicado anteriormente, tiene una carrera de 3 [mm], por ésta razón se utilizó otro tipo de actuador, trabaja a 12 [V] DC, el tiempo de trabajo es lenta a comparación con los actuadores empleados para las teclas de las tetras, como va a trabajar mucho menos tiempo que las teclas de las letras y con otro objetivo, éste es muy optimo para efectuar el trabajo de salto de línea. Éste actuador es activado mediante la señal enviada por el microcontrolador, y

activado con 12 [V] de una fuente externa mediante el trabajo de un transistor como switch, como ya se explico en el capítulo 3.

En la siguiente figura se muestra la ubicación del actuador dentro de la estructura metálica, debido a sus características físicas fue fácil ubicarle en un lugar estratégico dentro del la estructura, y así actúa de forma lineal sobre la tecla del salto d línea sin necesidad de acoplar un sistema de pivote o algún sistema extra para su funcionamiento.

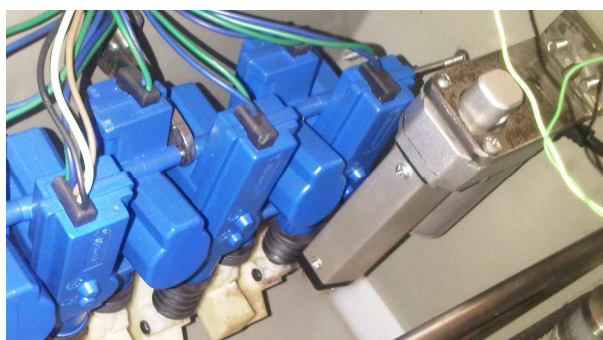


Figura. 110 Ubicación del actuador lineal para el salto de línea

En la siguiente figura se indica el actuador lineal del salto de línea desactivado.



Figura. 111 Actuador lineal del salto de línea desactivado

En la siguiente figura se indica el actuador lineal del salto de línea activado.



Figura. 112 Actuador lineal del salto de línea activado

5.2.2 Trabajo en el motor DC

El motor DC está acoplado en el sistema de movimiento del carro, para la prueba de funcionamiento se envió varias señales de activación desde el microcontrolador hacia el motor acoplado al carro en donde éste se mueve desde la izquierda hacia la derecha y viceversa.



Figura 113 Movimiento de la banda

El movimiento de izquierda a derecha y viceversa de este sistema, es con el objetivo de regresar al carro que contiene la cabeza de los punzones, hacia su posición original, de ésta manera, cada vez que existe un salto de línea, el carro regresa nuevamente hacia la izquierda que es la posición inicial y empieza de nuevo con la escritura en Braille.

En la siguiente figura se muestra este sistema del carrito en la posición inicial

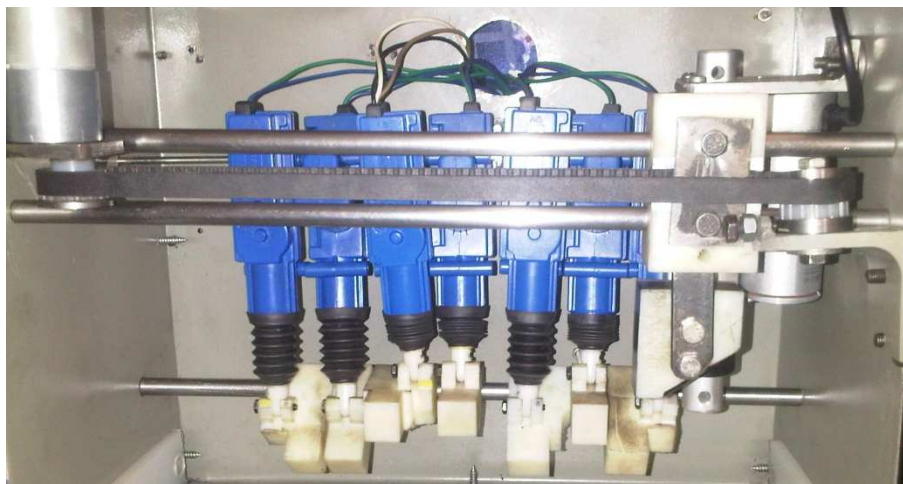


Figura 114 Sistema del carrito en la posición inicial

En la siguiente figura se muestra la ubicación del sistema del carrito en la posición final, de aquí el microcontrolador mandara las señales de control para que el carrito regrese nuevamente a su posición inicial.



Figura. 115 Sistema del carrito en la posición final

El sistema del carrito se compone por dos sistemas, el principal es el sistema del motor y la banda, que permite desplazarse de izquierda a derecha y viceversa, y el segundo es un sistema de acople a la banda en forma diagonal que permite presionar la pieza de retorno del carrito, facilitando el desplazamiento del mismo sin problemas.

5.3 Pruebas realizadas en el software

5.3.1 Conexión USB

La conexión USB es fundamental para el funcionamiento total del sistema, una de las condiciones de la programación fue que si el cable USB no está conectado, el sistema no funciona, mostrándose una ventana emergente que indica que el sistema se cerrará y que se debe conectar el cable USB, por lo tanto el paso principal para el funcionamiento total del sistema es tener el cable USB conectado.

Primera prueba: cable USB desconectado. En consola se muestra el mensaje de que el cable USB esta desconectado e inmediatamente se muestran las dos ventanas emergentes que se indican a continuación

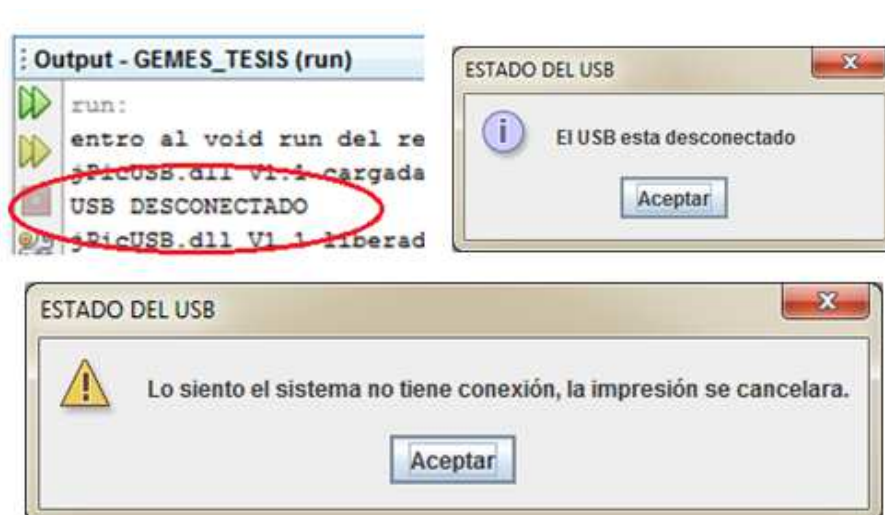


Figura. 116 USB Desconectado

Segunda prueba: cable USB conectado: en consola se muestra el mensaje de que el cable está conectado y el sistema funciona correctamente y se encuentra en espera de que el usuario realice cualquier acción.

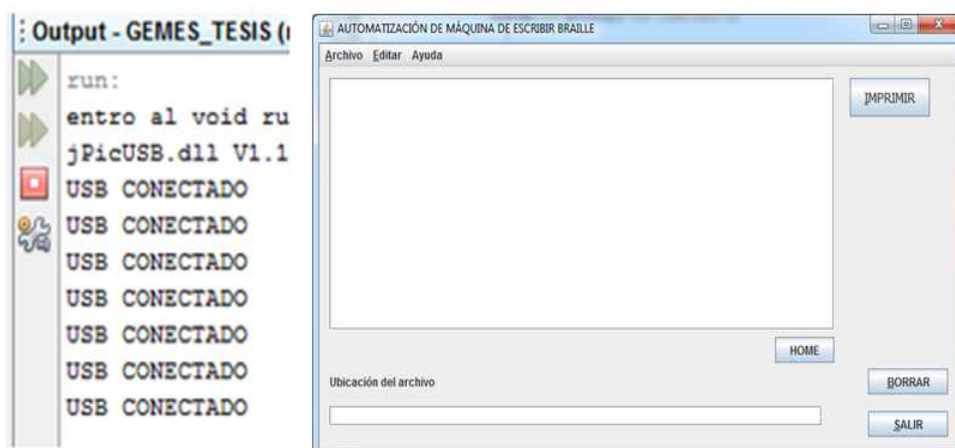


Figura 117 Cable USB conectado

5.3.2 Barra de Menú y Botones

Prueba menú abrir: se despliega una ventana de exploración en donde permite buscar el archivo con extensión *.txt, una vez que se ha seleccionado, se abre y se observara el archivo abierto en el área de texto. En el caso de que el archivo no se abrió se desplegará una ventana emergente que indique error al abrir.

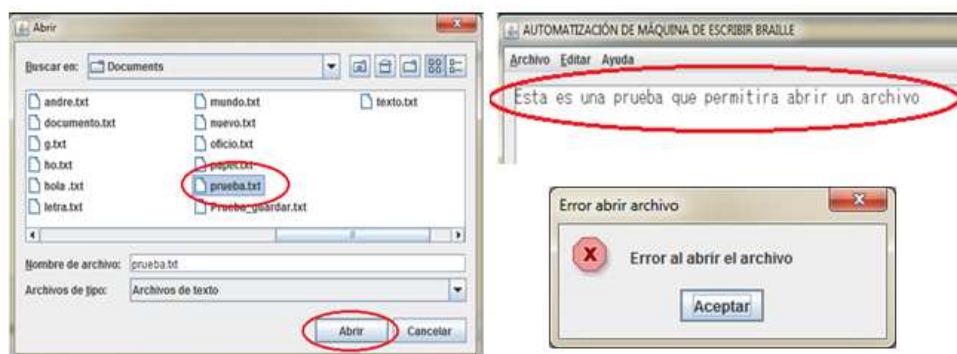


Figura 118 Menú Abrir

Menú Guardar: se despliega una ventana de exploración que permite buscar la ruta en donde se va a guardar el archivo y se escribirá el nombre del mismo, una vez que se ha guardado se desplegará una ventana emergente que indique que la acción fue correcta, la ventana emergente que indicará que existe error será cuando se quiera guardar un archivo vacío que no contenga texto.

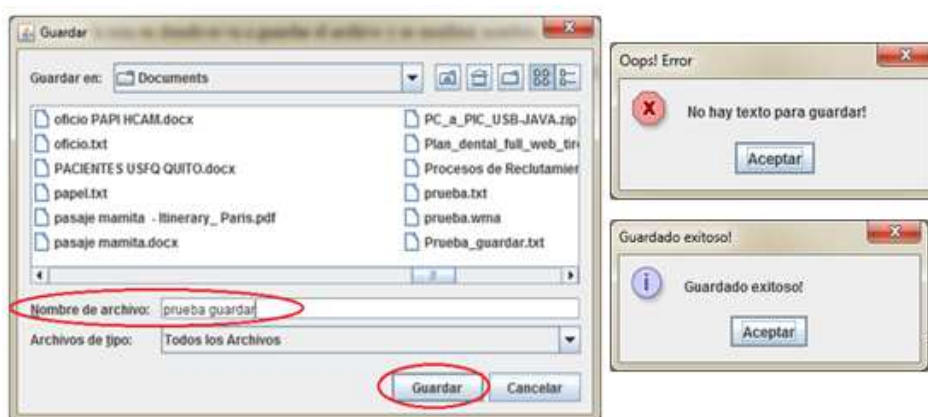


Figura 119 Menú guardar

Botón Salir: se desplegará una ventana emergente que indica si se desea salir del sistema.

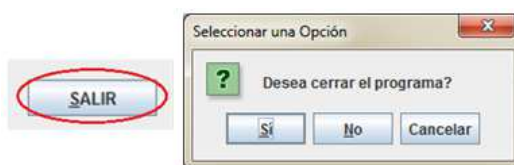


Figura 120 Ventana emergente que permite salir del sistema

Botón borrar: borrará todo el texto que se encuentre en el área de texto.

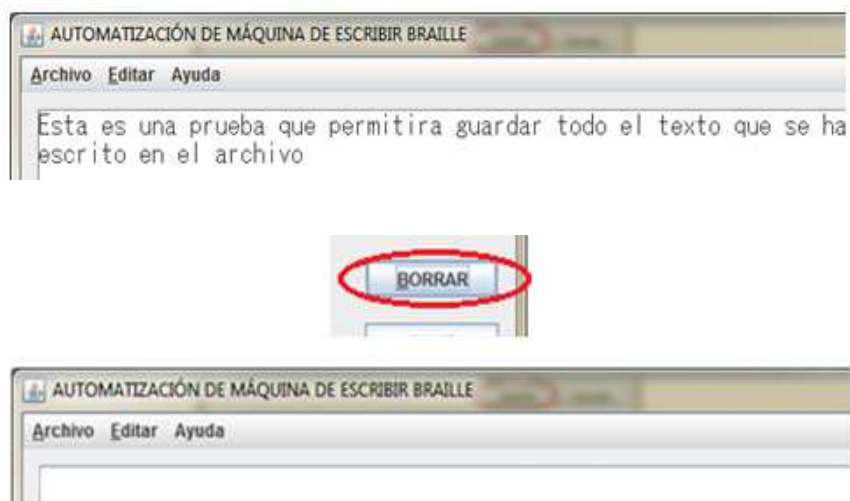


Figura 121 Botón que permite borrar el área de texto

5.3.3 Impresión

Cuando el texto va a ser impreso o tecleado mediante la máquina de escribir Braille, se desplegará una ventana emergente que le pregunta al usuario si desea imprimir para continuar con el proceso, esto se lo hizo como norma de seguridad, una vez que se acepte la impresión se podrá observar en el área de texto una advertencia que indica que el texto está siendo impreso y además ésta área se deshabilita con el objetivo de no permitir escribir texto hasta que la impresión termine, ésta advertencia se indica mediante un mensaje en una ventana emergente, una vez que la impresión haya terminado, se despliega una ventana emergente que pregunta si desea imprimir más texto, con esto el área de texto se borra y queda libre para que el usuario realice la acción que más le convenga.

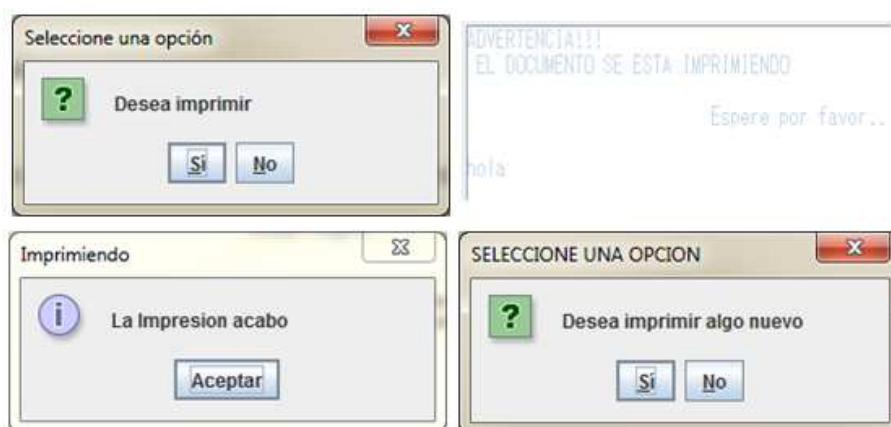


Figura 122 Proceso de impresión de texto

- **Estado del papel**

En la programación se consideró limitar el número de caracteres que se imprimen, tanto de forma horizontal que serán 30 caracteres, como vertical que serán 27 saltos de línea, de esta manera la hoja queda llena en su totalidad con los caracteres impresos.

Cuando llega el sistema a imprimir los 30 caracteres en forma horizontal, el sistema tiene una pausa para que el microcontrolador envíe una señal de control al

actuador lineal encargado de presionar la tecla del salto de línea y adicional envía otra señal de control al motor para que trabaje el sistema de recorrido del carro.

Al finalizar la impresión, el software le indicará que ya termino de imprimir, se debe sacar la hoja manualmente de la máquina.

5.4 Prueba Global del Sistema

Para la prueba global del sistema, se va a realizar una impresión de un texto, el cual será ingresado directamente por teclado.

Se va a probar algunas de las funciones de los menús, y finalmente se mostrara la hoja impresa en Braille el texto inicialmente ingresado.

- Abrir la aplicación: al abrir la aplicación saldrá en la pantalla la carátula de la misma, se debe esperar un momento y se abrirá la aplicación.

Esta caratula le proporcionara una introducción al software.



Figura 123 Presentación del software

- El software está ejecutándose, la pantalla principal es la siguiente

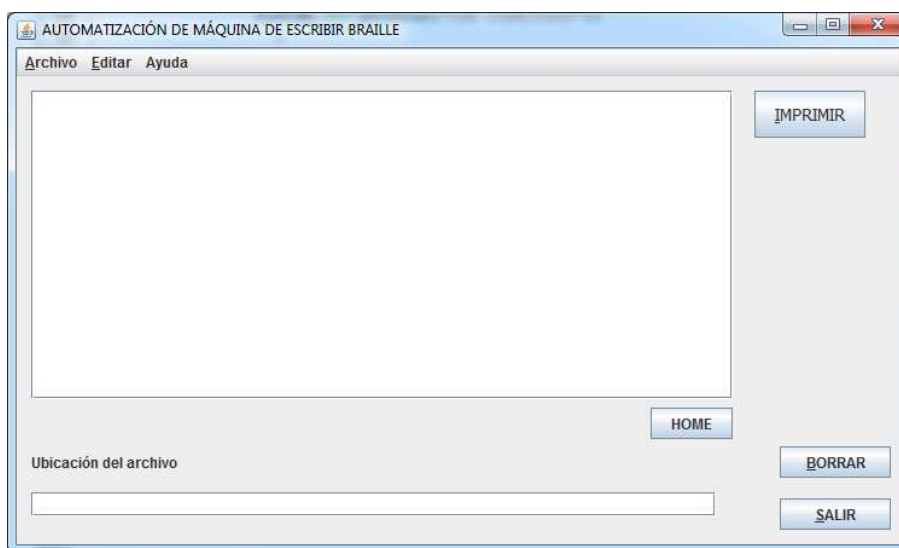


Figura 124 Pantalla principal

- Se introduce el texto mediante el teclado, se va a escribir “Bienvenido al sistema”

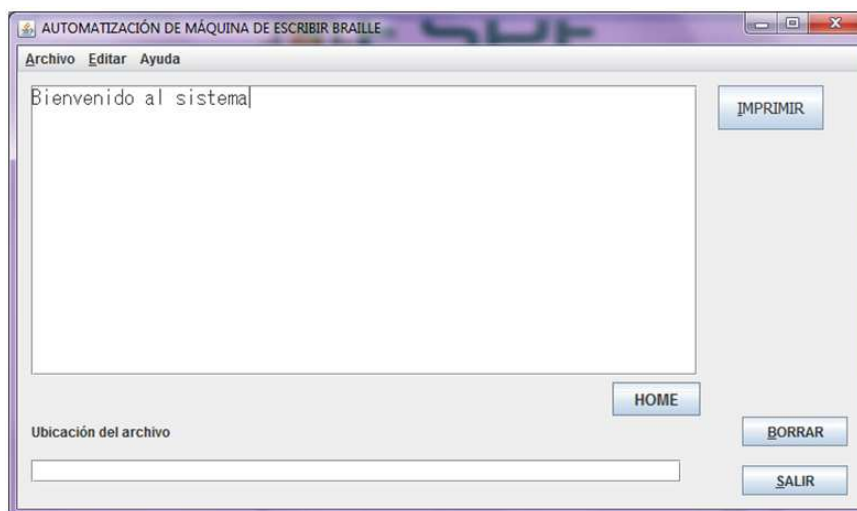


Figura 125 Ingreso de texto por teclado

- Se va a probar algunas funciones de la barra de menú, primero se guardará el archivo, luego se borrará la pantalla y se volverá a abrir el documento guardado

-

- Guardar Archivo

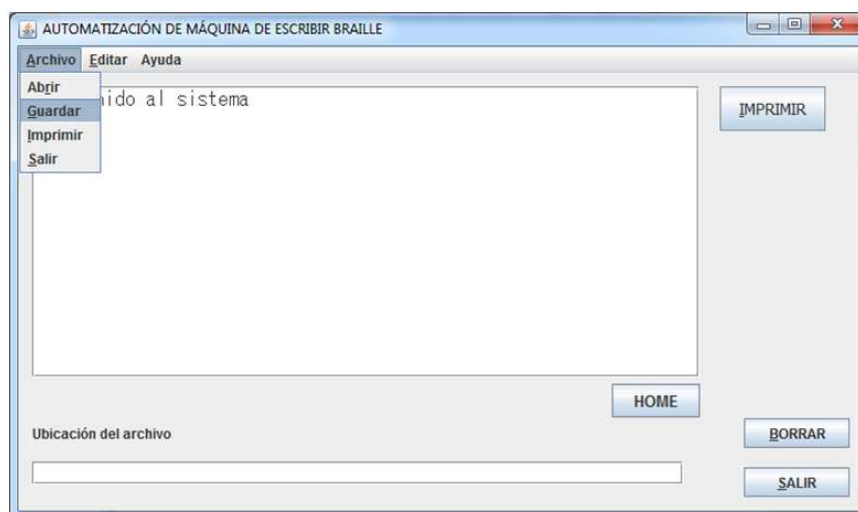


Figura 126 Función Guardar

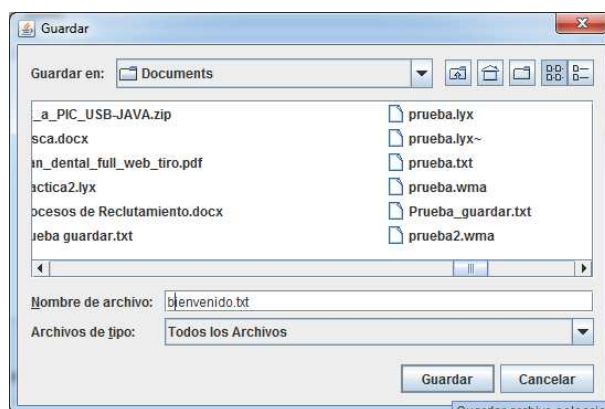


Figura 127 Guardar Archivo

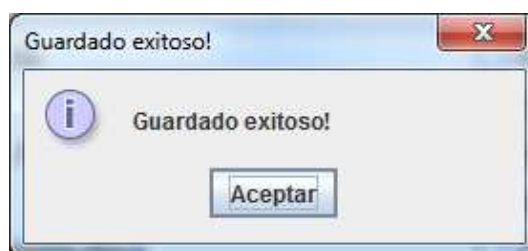


Figura 128 Guardado exitoso

- Borrar Pantalla

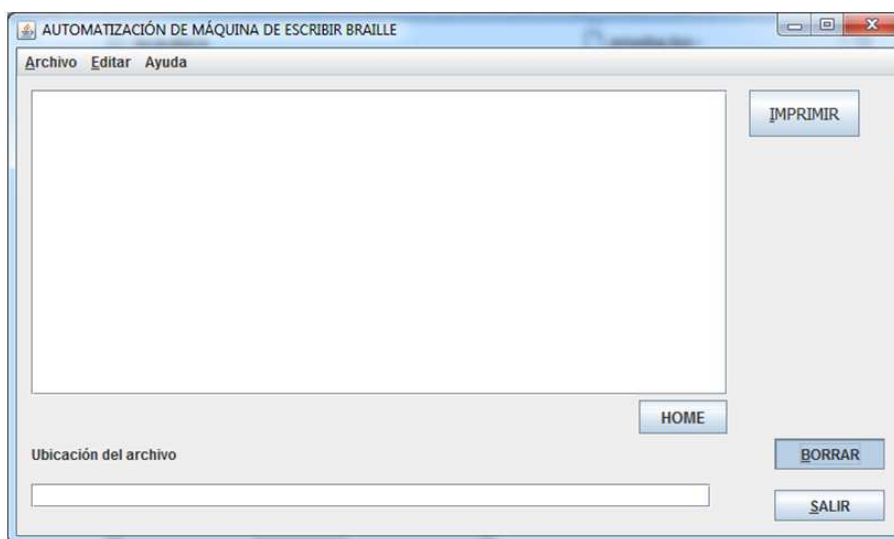


Figura 129 Función Borrar

- Abrir Documento

El documento que se va a abrir es el que inicialmente se escribió mediante el teclado y guardó como documento *.txt

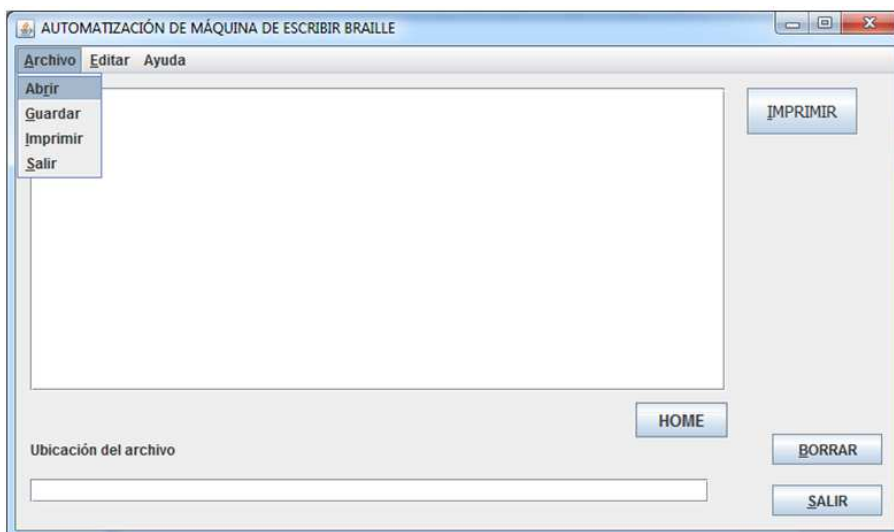


Figura 130 Función Abrir

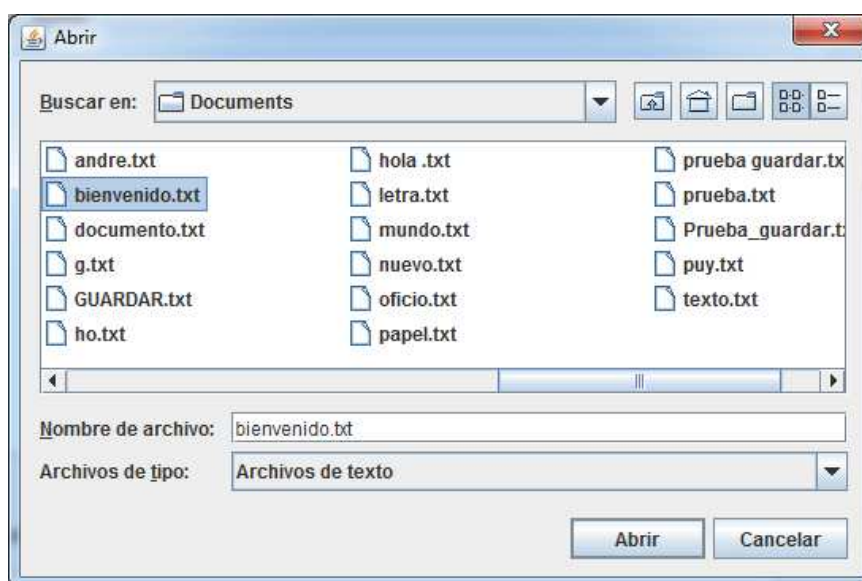


Figura 131 Abrir Archivo

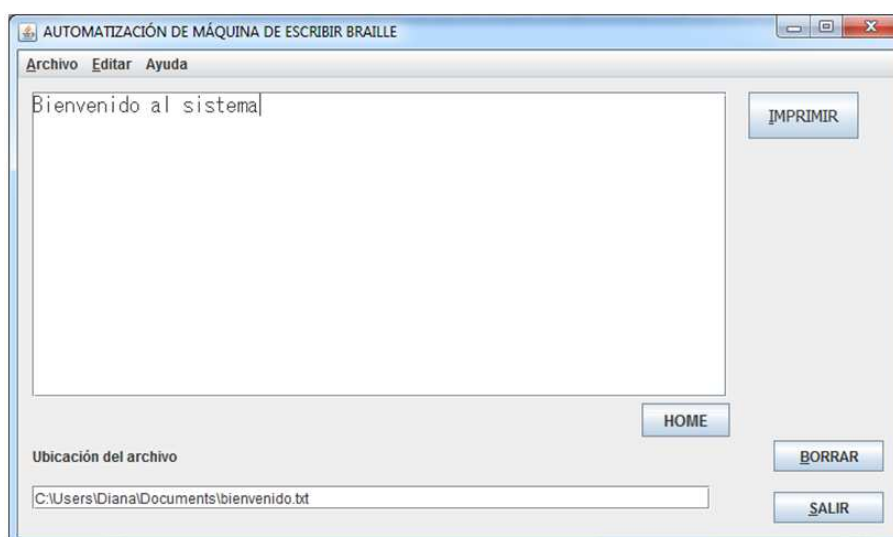


Figura 132 Archivo Abierto

- Ahora que ya se tiene abierto el documento, se procederá a imprimir, se observa en la pantalla principal que el área de texto se ha bloqueado por un momento hasta que termine de imprimirse, esto es con el objetivo de que no se pueda escribir nada mientras se está imprimiendo, evitando cualquier inconveniente de sincronización al momento de enviar los datos al microcontrolador

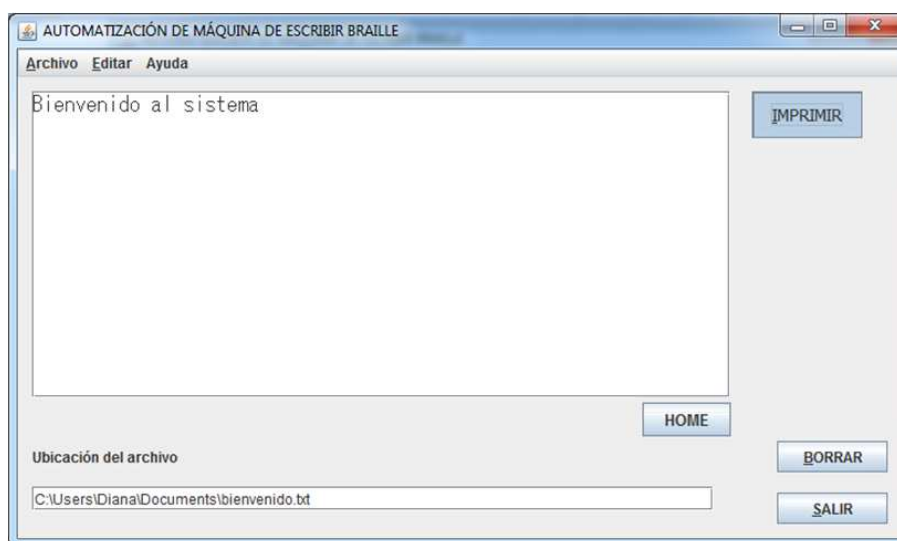


Figura 133 Función Imprimir

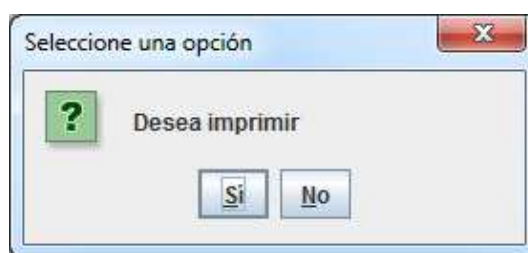


Figura 134 Ventana emergente, seleccionar una de las opciones

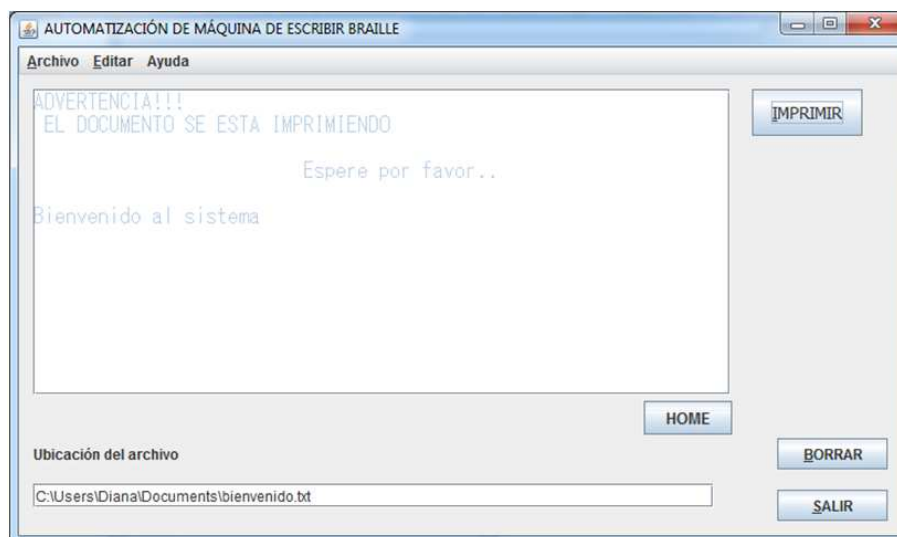


Figura 135 Pantalla bloqueada



Figura 136 La Impresión acabó

- Cuando haya terminado la impresión preguntará si desea imprimir de nuevo o no, si su opción es SI, la ventana de la pantalla principal se habilitara nuevamente, y así podrá volver a ingresar un texto e imprimirlo nuevamente, o realizar cualquier acción que se desee. Si la opción es NO, la ventana se cerrara, por ende el software se cerrará.

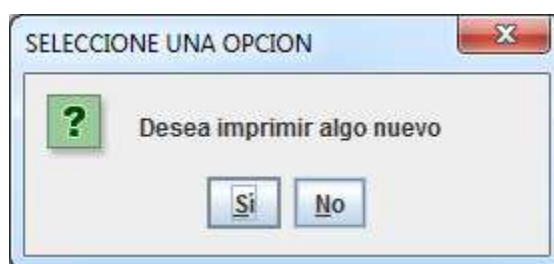


Figura 137 Ventana emergente, escoja una de las opciones

- Cuando la hoja se haya llenado, el software le indicará que coloque nuevo papel, esperará un tiempo prudente hasta que pueda ingresar una nueva hoja de papel. La impresión se pausara por un momento, cuando ya esté listo el papel, presione aceptar y la impresión continuara.

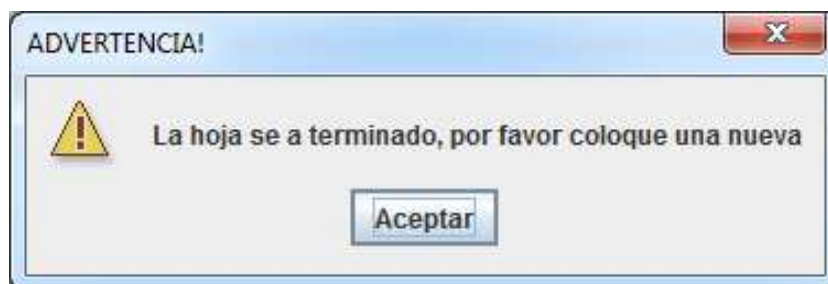


Figura 138 Mensaje de advertencia, coloque nueva hoja

Cuando haya pasado un tiempo prudente, vuelve a preguntar si colocó nueva hoja, si se escoge SI la impresión continuará, si el caso es NO, esperara un tiempo prudente y volverá hacer la misma pregunta, si al tercer intento no se coloca nuevo papel, el software se cerrará automáticamente por seguridad.

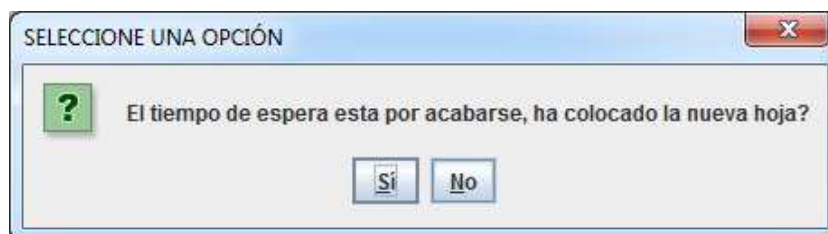


Figura 139 Escoja una opción con respecto al papel

- A continuación se indica la hoja impresa en Braille, se debe comparar con el alfabeto en Braille, para corroborar que la impresión fue correcta
-
- Intento 1

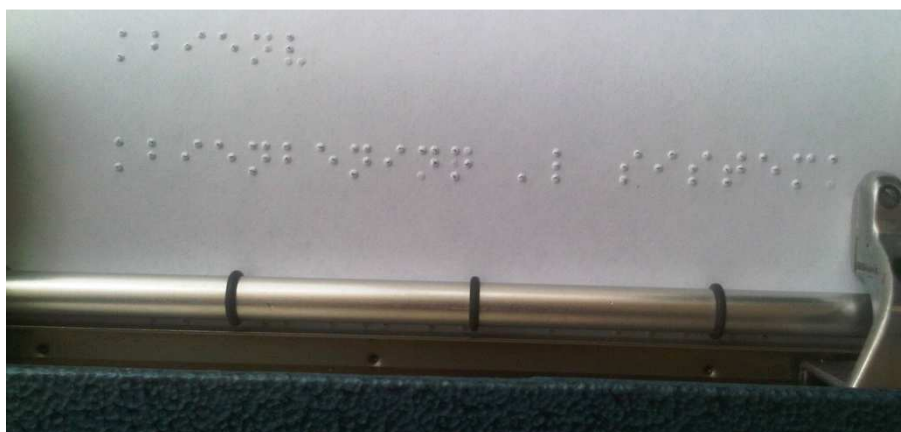


Figura 140 Impresión de “Bienvenidos al sistema”

Como se puede observar en la figura, las letras “D” y “O” de la palabra “Bienvenidos” tiene errores de impresión, el error radica en la programación del microcontrolador, ya que la palabra de control que envía la señal a la salida del mismo estaba mal direccionada, haciendo que se activen otros actuadores, y además la letra “S” de la palabra “Bienvenidos” no se imprimió, también por errores en la palabra de control de la letra “S”.

- Intento 2

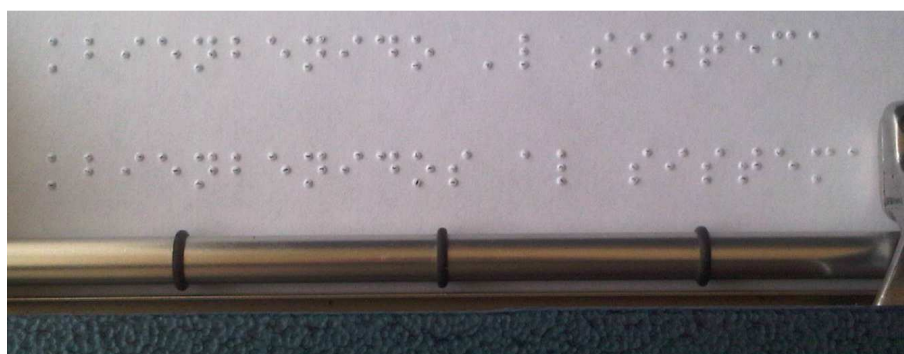


Figura 141 Impresión de “Bienvenidos al sistema”

El error fue corregido, y la frase “Bienvenidos al sistema” está escrito correctamente. Para comprobar esto, se comparará con el alfabeto Braille.

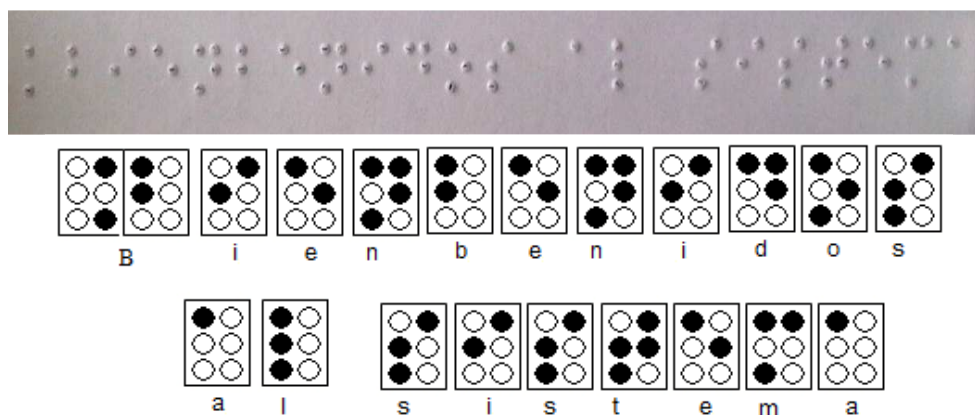


Figura 142 Frase impresa: “Bienvedos al sistema”

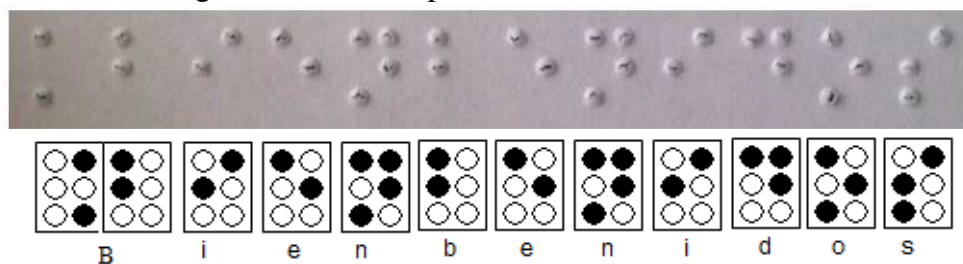


Figura 143 Palabra “Bienvenidos”

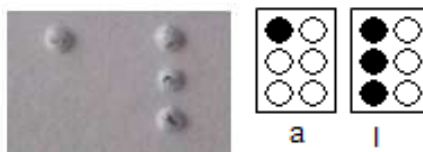


Figura 144 Palabra “Al”

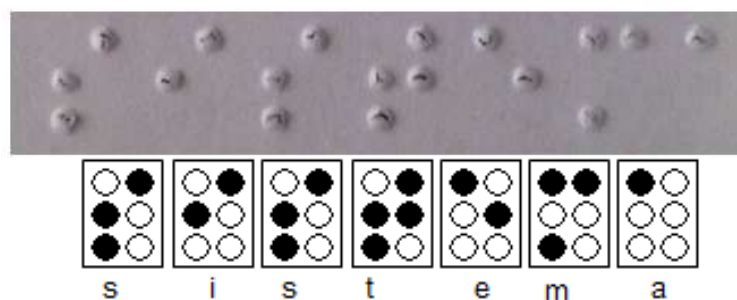


Figura 145 Palabra “Sistema”

5.5 Análisis de resultados

Después de realizar varias pruebas de funcionamiento, se encontraron varios errores, tanto en software como en hardware, los mismos que fueron solucionados a lo largo del proceso de la automatización, con el fin de tener un trabajo óptimo y eficiente, causando el menos riesgo de problemas de ejecución.

Algunos errores de software que inicialmente causaron problemas fue la importación de las librerías JPicUsb de Java, ya que la versión que se instaló fue la JPicUsb 1.0, y la versión actualizada de NetBeans es compatible con la JPicUsb 1.1.1, ésta versión actualizada se la puede descargar de la página web de Java.

Otro error fue la comunicación USB, hay que recordar que la comunicación USB es una comunicación que envía y recibe datos a una alta velocidad, y si existe alguna falla sea ésta de software o hardware, es imposible tener una correcta comunicación, el principal error fue tener una mala configuración en el protoboard donde se realizó las pruebas de funcionamiento, ya que en el pin 14 del microcontrolador 18f2550 es importante tener conectado un capacitor electrolítico de 4,7 uF, el mismo que regula el voltaje y provee de 3.3 V constantes, con este voltaje no existe inestabilidad en el microcontrolador. Otro punto importante es tener conectado un capacitor de 100nF de desacople entre los pines 19 y 20, con el fin de almacenar energía y estabilice el voltaje de alimentación al microcontrolador.

Los datos de envío y recepción de NetBeans deben estar sincronizados con los datos de envío y recepción del microcontrolador, ya que si los datos que se están enviando desde la aplicación de NetBeans no están sincronizados con los datos que

recibe el microcontrolador o viceversa, la comunicación no se efectuara, por esta razón es importante tener una sincronización de datos tanto de transmisión como de recepción.

El driver que provee la casa comercial del microcontrolador para la comunicación USB, debe ser instalado correctamente, ya que si el microcontrolador no detecta el driver respectivo de Microchip, la comunicación no existe. Hay que verificar que ése driver este instalado en la maquina en la cual se realizara la ejecución del software.

Tanto el cable USB como los puertos de la computadora deben estar en perfecto estado, ya que si no lo están existe problemas de conexión y desconexión constantes mientras se está realizando cualquier envío y recepción de datos, ocasionando problemas de ejecución.

En cuanto a la estructura metálica, se la diseño de tal manera que sea fácil de introducir la máquina de escribir Braille, debido a sus características físicas, la máquina queda muy estable, esto evita algún desperfecto de las teclas y de los actuadores.

Después de realizar un exhaustivo análisis y configuración de cada una de las letras del alfabeto Braille, y después de tener varios errores de impresión con letras equivocadas, el software es capaz de imprimir cada una de las letras deseadas, así como también, números y caracteres especiales, dentro de todas las reglas gramaticales y conjugaciones específicas que tiene el alfabeto Braille, la única limitante que tiene el software es la escritura en español, debido a que la programación fue desarrollada en la plataforma Java, y ésta plataforma está en inglés, no reconoce las vocales tildadas y la letra Ñ y ñ, es por esto, que antes de imprimir cualquier texto, hay que verificar que no existan vocales tildadas ni la letra “eñe”, para evitar errores, se las debe reemplazar por la letra “ene” y por vocales sin tilde.

Para poder controlar el actuador del salto de línea, se realizaron varias pruebas, para ver cuál es la carrera adecuada a la que debe desplazarse éste actuador con el fin

de realizar el salto de línea, ya que si es muy corto no realiza un salto de línea adecuado, mientras que si es muy largo, puede causar problemas en la tecla.

El cristal de oscilación adecuado para trabajar en esta aplicación es el de 20MHz, ya que con este reloj la sincronización de datos de transmisión y recepción son constantes sin acelerarlo ni retrasarlo, lo que permite trabajar perfectamente sin perder ningún dato.

Debido a la incompatibilidad que tienen las aplicaciones de Java con el software Jaws, fue necesario agregarle audio a cada una de las ventanas emergentes del software, así como también a las diferentes funciones de la barra menú, con el objetivo de que las personas no videntes puedan saber en qué lugar del software se encuentran y sobretodo conocer cada una de las funciones que se está realizando, los mensajes de advertencia, de confirmación y de información.

Es necesario escuchar los mensajes de audio que el software proporciona, para saber que acción se debe realizar, los mensajes indican también los mnemónicos que puede presionar para ejecutar dicho trabajo.

Tomando en cuenta cada uno de los errores que fueron apareciendo a lo largo del proceso de la automatización, y al solucionarlos a tiempo con el respectivo análisis de su defecto, la automatización que se ejecuta es una automatización muy buena, ya que no existe perdidas de datos, las letras que se imprimen son las correctas, la estructura es adecuada para la máquina, los actuadores trabajan muy bien, y la comunicación es excelente, con lo que se podría decir, que el trabajo de ejecución es bueno.

Hay que recordar que éste diseño es un prototipo, por lo que a medida que trabaje, tal vez irán surgiendo algunos desperfectos en su funcionamiento, es por esto que este proceso de automatización queda abierto para que se pueda ir mejorando su desempeño en diferentes aspectos, pueden ser estos: incluir el lenguaje en español, realizar una estructura metálica más óptima, utilizar otro tipo de actuadores lineales, entre otros y también ampliar nuevas aplicaciones de conversión Braille tales como signos matemáticos, signos musicales, entre otras aplicaciones.

5.6 Manual de operación y mantenimiento

En el manual de operación y mantenimiento se detallan los parámetros con los cuales debe funcionar el sistema, así como también la manera en la que no se debe usar el sistema y las posibles soluciones en el caso de encontrarse con algún error, también se ha considerado el mantenimiento que dará una vida útil aceptable a la máquina y al sistema.

El manual de operación y mantenimiento se muestra en el ANEXO

5.7 Análisis económico

Para la implementación y diseño del proyecto, se tiene un presupuesto base de 1500 dólares americanos, financiados en su totalidad por las autoras del proyecto.

5.7.1 Costos directos

Se ha considerado todos los materiales y elementos necesarios para construir todo el sistema, siendo estos: estructura mecánica, actuadores y motores dc, elementos electrónicos para la placa, fuente de poder.

Cuadro 19

Costos de actuadores y fuente

No.	Descripción	Cantidad	Valor unitario	Valor total
1	Actuador lineal carrera 2[mm], 12[V]	7	6	42
2	Actuador lineal carrera 5[mm], 12[V]	1	95	95
3	Motor reductor 80[rpm], 12[V], 18Kgcm	1	40	40
4	Fuente de poder 12[V], 20[A]	1	35	35
TOTAL				212

Cuadro 20

Costos de la estructura para la Máquina Braille

No.	Descripción	Cantidad	Valor unitario	Valor total
1	Caja de acero estructural	1	250	150
2	Accesorios de nylon para pulsar teclas	1	100	100
3	Sistema de recorrido lineal con rodamientos lineales auto entrantes, poleas dentadas, y banda	1	300	150
4	Mano de obra	1	300	300
5	Gastos Extras	1	50	50
TOTAL				750

Cuadro 21

Costos de placa electrónica y elementos electrónicos

No.	Descripción	Cantidad	Valor unitario	Valor total
1	Ruteado, perforación y soldadura de la placa	1	40	40
2	Resistencia 10k ohm	11	0.02	0.20
3	Capacitor electrolítico 4.7uF/25V	1	0.07	0.07
4	Capacitor 15 pF	2	0.07	0.14
5	Capacitor 100 nF	8	0.07	0.57
6	Zocalo 28 pines	1	0.13	0.13
7	Zocalo 16 pines	1	0.09	0.09
8	Transistor Tip122 NPN	7	0.45	3.12
9	Led 5V	2	0.07	0.14
10	Diodo 1N4007	7	0.07	0.50
11	Bornera 2 pines	10	0.22	2.23
12	Switch 2 posiciones	1	0.27	0.27
13	Conector USB tipo B	1	0.58	0.58
14	Cristal 20MHz	1	0.49	0.49
15	Baquelita	1	10	10
16	Cable USB	1	4	4
17	Bornera de cable	5	0.25	1.25
18	Cable	2 [m]	2.50	5
19	Espagueti Termico	5 [m]	0.40	2
20	Enchufe blindado 3 terminales	1	0.60	0.60
21	Cable Sucre 3x18 AWG	3 [m]	1.25	3.75
			Subtotal:	75.13
			IVA:	9.01
			Total:	84.15

5.7.2 Costos indirectos

Cuadro 22

Costos varios

No.	Descripción	Cantidad	Valor unitario	Valor total
1	Copias bibliográficas	100	0.03	3
2	Impresiones blanco y negro	700	0.05	35
3	Impresiones a color	80	0.30	24
3	Anillados	4	3.50	14
4	Empastado de tomos de tesis			
5	Útiles de oficina			50
6	Movilización			100
TOTAL				226

5.7.3 Resumen de Costos

Cuadro 23

Resumen de Gastos

Descripción	Valor total
Costos directos	1046.15
Costos indirectos	226
Imprevistos	250
TOTAL	1522,14

El análisis económico de este proyecto tiene varios factores que deben ser tomados en cuenta, en primer lugar se analizarán los costos directos, que son los costos que influyen netamente en los gastos de construcción e implementación del sistema. Con estos gastos se analizará la importancia de realizar este proyecto en comparación con algún sistema de automatización o impresora Braille que existe en el mercado actualmente.

Los gastos directos engloban todos los elementos, dispositivos y materiales utilizados, tales como actuadores, fuente eléctrica, placa electrónica, estructura metálica, entre otros elementos que están mencionados en los cuadros. Todos estos elementos utilizados suman una cantidad de 1046.15 dólares americanos, que haciendo referencia con la economía nacional es un valor accesible tomando en cuenta la dificultad y trabajo del sistema, tanto de diseño como de construcción.

Uno de los factores para abaratar costos, fue la importación de algunos elementos como la fuente de poder y el actuador lineal para el salto de línea, éstos se los adquirió mediante una compra por internet, de ésta manera se abarato un poco los gastos y sobretodo fue fácil adquirirlos, ya que lamentablemente aquí en el país no se encontró fácilmente el actuador con las características físicas necesarias y la fuente de poder tenía un precio elevado que sobrepasaba el presupuesto general para el proyecto.

5.8 Costo / beneficio

La necesidad de diseñar e implementar este proyecto radica básicamente en realizar un proyecto que sea de un costo bajo y accesible, ya que, si se compara con el presupuesto que se necesitaría para adquirir una impresora Braille comercial, la diferencia es muy notable, se ha investigando en algunos portales web de fundaciones, instituciones y organizaciones orientadas a la ayuda social de personas no videntes ubicadas en Argentina, México y España, y se conoció que los precios de impresoras Braille oscilan entre 3500 a 7000 dólares americanos, tomando en cuenta que el precio más bajo es para impresoras usadas y con tecnología desactualizada, mientras que el precio más alto ocupan las impresoras nuevas, con tecnología actualizada, con alimentación directa de papel, de tamaño pequeño, entre otras características.

Aquí en el país no es fácil acceder a una máquina de escribir Perkins estándar, y mucho más difícil es acceder a una impresora Braille. Una máquina de escribir Braille se la puede adquirir directamente en el portal de Perkins en Estados Unidos, o por medio de la FENCE, el precio de la maquina oscila entre unos 1000 dólares americanos aquí en el Ecuador, por lo que si ya se tiene una máquina de escribir Braille estándar, es mucho más fácil y económicamente más accesible adquirir el sistema de automatización, ya que, si al valor de la máquina de escribir Braille estándar se le suma el costo de la implementación y construcción del sistema de automatización, sumarían aproximadamente unos 2000 dólares americanos, comparando el valor con el costo más bajo de una impresora Braille comercial, aún así, éste sistema sigue siendo más accesible y económico.

CAPÍTULO 6:

CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones

- Se diseñó e implementó un sistema capaz de automatizar a una máquina de escribir Braille, que está dirigido tanto a personas no videntes como videntes, pues la interfaz contiene funciones básicas y necesarias para imprimir cualquier documento que se encuentre en formato digital hacia Braille, este sistema ayuda a disminuir el tiempo de transcripción de un texto digital en alfabeto latino hacia alfabeto Braille, con la orden de ejecución en el software que lo hará el usuario.
- La interfaz a PC desarrollada en el presente proyecto es una interfaz gráfica que contiene las operaciones básicas para el manejo del sistema, su implementación es sencilla debido a que está dirigido a personas no videntes por esta razón es una interfaz más funcional que llamativa, presenta mnemónicos dentro de la programación para un fácil acceso a las funciones del sistema y contiene audio que le permite al usuario conocer en que lugar de la interfaz se encuentra y que esta realizando el programa en tiempo real.
- El Sistema Braille es un código de lectura diseñado para que las personas no videntes puedan leerlo mediante el tacto ya que son signos en alto relieve basado principalmente en la combinación de 6 puntos que se encuentran dispuestos en una matriz de dos columnas y tres filas.
- El Sistema Braille es un código pues todos sus signos se adaptan al idioma en el que la persona no vidente lo escribe y/o lee y como tal se debe respetar las particularidades y la sintaxis que componen a este código, de la misma manera se considera la distribución y el tamaño en el que deben estar ubicados en el Signo Generador (código Braille) debido a que éste signo es estándar, Louis Braille llegó a determinar que las yemas

de los dedos son capaces de captar este tamaño y distribución en cada signo.

- Debido a que el código Braille se basa en la composición de 6 puntos, se tiene un número limitado de signos que no representan a todas las letras del alfabeto, por esta razón existen signos diferenciadores especiales utilizados como prefijos para representar caracteres como: letras en mayúscula, números, signos de puntuación, símbolos matemáticos, caracteres especiales, signos musicales, entre otros.
- Los actuadores lineales ideales para la automatización del presente proyecto son aquellos actuadores lineales eléctricos que tengan como características técnicas 20[mm] de carrera para las teclas de los puntos y 30[mm] de carrera para la tecla del salto de línea, fuerza máxima de 1[N], pues se comprobó que ésta es la fuerza necesaria para que el papel se marque sin que se rompa y que el signo Braille sea legible.
- El microcontrolador 18f2550 utilizado en el presente proyecto desarrolla una alta y veloz comunicación con el USB, la transmisión y recepción de datos se la realiza byte a byte entre el PC y el microcontrolador, además el driver de este microcontrolador que establece la comunicación, es de fácil acceso y no tiene costo, lo que no implica un gasto extra al momento de adquirirlo.
- Tanto el entorno de programación del microcontrolador (PIC C Compiler) como el de la interfaz grafica (Java – NetBeans) proveen al programador las librerías necesarias que establecen y facilitan el desarrollo de prototipos como el que se implemento debido a la comunicación USB que permite al usuario interactuar con el software y mediante la ejecución de la orden envía a imprimir en Braille lo que necesita automatizando al sistema.

- Se construyó una estructura metálica que acopla los actuadores lineales eléctricos en el lugar exacto para que presionen a cada una de las teclas de la Máquina de Escribir Braille de tal manera que se controlan mediante el microcontrolador y el circuito electrónico diseñado para tal objetivo.
- El lenguaje de desarrollo del código para la conversión del alfabeto latino al alfabeto Braille es una plataforma de distribución gratuita que se encuentra en inglés, por lo que el sistema no reconoce los caracteres especiales de la lengua en español como son las vocales tildadas y la letra ñe.
- Los datos de transmisión y recepción de la aplicación de NetBeans deben estar sincronizados con los datos de transmisión y recepción programados en el microcontrolador 18f2550, ya que si esta sincronización no existe se corre el riesgo de perder datos produciendo que la ejecución del sistema no sea óptima ni exacta.
- El costo de la construcción e implementación del sistema tiene un valor accesible para la población ecuatoriana tomando en cuenta la aplicación a la que se le va adaptar y sobre todo a la dificultad y trabajo del sistema, tanto de diseño como de construcción comparado con el alto precio actual de una impresora Braille comercial, este sistema de automatización es económico.
- El proyecto desarrollado presenta una gran ventaja, pues al ser un programado en Java que es una multiplataforma, el archivo ejecutable del software puede ser instalado en cualquier sistema operativo independientemente de la versión, solo se necesita instalar las librerías y drivers que permiten la comunicación USB entre el computador y el microcontrolador y además se necesita instalar el JRE de java que es el que permite que el ejecutable corra en cualquier plataforma.

7.2 Recomendaciones

- Previo el desarrollo de un proyecto como éste, es importante investigar sobre las posibles soluciones al problema planteado, tales como los elementos electrónicos y eléctricos que se desean usar, pues deben ser elementos de fácil acceso comercial, que no representen un mayor gasto que el presupuestado y que el tiempo que toma en adquirirlos no sea demasiado largo, pues provocará la pérdida de tiempo en la elaboración del proyecto.
- Se recomienda ocupar una fuente de poder que suministre el voltaje y la corriente necesarios para que todos los actuadores lineales eléctricos y el motor que se encuentran implementados en la estructura mecánica funcionen correctamente y cumplan con el objetivo específico designado evitando así cualquier daño en el sistema.
- Es recomendable que antes de realizar el circuito electrónico impreso y transferirlo a la placa electrónica, se realicen pruebas de funcionamiento tanto en el simulador como en un circuito implementado en el protoboard para que justifiquen y comprueben que el circuito diseñado está en óptimas condiciones de uso y sin errores.
- Es de suma importancia usar lenguajes desarrolladores de código que sean de alto nivel ya que éstos cuentan con las librerías y drivers necesarios que permiten establecer una correcta transmisión y recepción de datos entre el PC y el microcontrolador 18f2550 mediante el cable USB con alta confiabilidad en lo que se refiere a la comunicación.
- Al momento de trabajar con el sistema, se recomienda colocar a la estructura metálica sobre una mesa que sea estable, de esta manera tanto la estructura como la máquina de escribir Braille no sufrirán daños al momento en que se encuentra en funcionamiento.

- Como se utilizó un lenguaje de desarrollo del código para la conversión del alfabeto latino al alfabeto Braille en una plataforma en inglés, los caracteres especiales de la lengua en español como son las vocales tildadas y la letra ñe no serán reconocidas, por este motivo se recomienda que el usuario borre y reemplace las vocales tildadas por vocales no tildadas y de la misma manera la letra ñe la reemplace por la letra ene en el área de texto de la interfaz gráfica antes de ejecutar la orden de impresión, pues estos caracteres no se van a imprimir y se presentaran inconvenientes gramaticales al momento de leer el texto en Braille.
- Se recomienda realizar una limpieza de polvo y partículas del medio ambiente de la estructura metálica y de la Máquina de Escribir Braille cada cierto tiempo de uso para que todos los mecanismos funcionen sin problema.
- Cuando ocurra algún tipo de emergencia como por ejemplo, falta de energía mientras está trabajando el sistema, o pérdida de datos porque la computadora se apago, o el cable USB se desconecto en la mitad del proceso, entre otros, y los actuadores del retorno del carro y del salto de línea se quedaron activados en la mitad del camino, es importante presionar el “Botón Home” para que éstos actuadores regresen a sus posiciones originales, y de esta manera el sistema trabaje correctamente y no exista ningún daño físico de la estructura, así como también de los actuadores.
- Al finalizar la impresión, el sistema le indicará que ya acabo de imprimir y le preguntará si desea imprimir un nuevo documento o no, indiferentemente de cuál sea la opción que escoja, deberá sacar la hoja de la máquina de escribir manualmente, e ingresar una nueva hoja para poder continuar con un nuevo proceso de impresión

REFERENCIAS BIBLIOGRAFICAS

- BAC. (n.d.). *Biblioteca Argentina Para Ciegos*. Retrieved from BAC: <http://www.bac.org.ar>
- Cuervo, A. (n.d.). *Motor DC*. Retrieved from www.cienciamia.net/fisica/notasfisica2/3.14%20Motores.doc
- Data_Sheet_PIC18F2550. (2009). *Data_Sheet_PIC18F2550*. Retrieved 2014, from Microchip Technology Inc: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>
- Data_Sheet_TIP_122. (n.d.). *Microchip Technology Inc*. Retrieved from <http://www.adafruit.com/datasheets/TIP120.pdf>
- Discapacidades del Ecuador*. (n.d.). Retrieved from Federaciones nacionales de y para la discapacidad del Ecuador: http://www.discapacidadesecuador.org/portal/index.php?option=com_content&task=blogcategory&id=37&Itemid=109
- Educación Inclusiva*. (n.d.). Retrieved from Personas con Discapacidad Visual: http://www.ite.educacion.es/formacion/materiales/129/cd/pdf/m5_dv.pdf
- FENCE*. (n.d.). Retrieved from Federación Nacional de Ciegos del Ecuador: <http://www.fenceec.org/>
- Herrera, J. (2010, Diciembre 03). La discapacidad en Ecuador. *Últimas Noticias*, p. 08. Retrieved from <http://www.ultimasnoticias.ec/noticias/834-la-discapacidad-en-ecuador.html>
- Java*. (n.d.). Retrieved from <http://www.java.com/es>
- Sistema Braille*. (n.d.). Retrieved from <http://www.webacic.cat/es/casisbra.htm>
- Tiflogía*. (n.d.). Retrieved from Educación y rehabilitación de ciegos y disminuidos visuales: http://tiflogia.blogspot.com/2014_03_01_archive.html

ACTA DE ENTREGA

El proyecto fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Universidad de las Fuerzas Armadas – ESPE, desde:

Sangolquí, 28 de Julio de 2014

ELABORADO POR:

Andrea Estefanía Noboa Montenegro

ANDREA ESTEFANÍA NOBOA MONTENEGRO

1720902905

Diana Sofía Noboa Montenegro

DIANA SOFÍA NOBOA MONTENEGRO

1720902897

AUTORIDAD

Luis Orozco
Ing. Luis Orozco MSc.

