

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISPOSITIVO DE ENTRADA PARA PERSONAS CON DISCAPACIDAD VISUAL

DESARROLLO DE UNA PROPUESTA DE UN PROTOTIPO DE DISPOSITIVO DE ENTRADA (TECLAS) PARA UN COMPUTADOR PERSONAL ORIENTADO A PERSONAS CON DISCAPACIDAD VISUAL

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN
TECNOLOGÍAS DE LA INFORMACIÓN**

CONSUELO ALEJANDRA ESCOBAR JURADO

consuelo.escobar@epn.edu.ec

DIRECTOR: Ing. XAVIER ALEXANDER CALDERÓN HINOJOSA M.Sc.

xavier.calderon@epn.edu.ec

DMQ, septiembre 2024

CERTIFICACIONES

Yo, Consuelo Alejandra Escobar Jurado declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

CONSUELO ALEJANDRA ESCOBAR JURADO

Certifico que el presente trabajo de integración curricular fue desarrollado por Consuelo Alejandra Escobar Jurado, bajo mi supervisión.

M.Sc. XAVIER ALEXANDER CALDERÓN HINOJOSA
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CONSUELO ALEJANDRA ESCOBAR JURADO

M.Sc. XAVIER ALEXANDER CALDERÓN HINOJOSA

DEDICATORIA

Para mi familia, quienes me han brindado un gran apoyo durante mi carrera universitaria y consejos en los buenos y malos momentos. A mi madre Consuelo, que me ha enseñado la importancia de hacer lo que te apasiona con amor, de su fuerza y constancia día a día. A mi padre Nelson por mostrarme sus enseñanzas como la comprensión y a siempre brindar una mano amiga hacia los demás. A mis hermanos Isabel, Bethzabé y Nelson, por la unión y su gran afecto, a quienes considero mis preciadas luces en mi vida.

A mis amigos Alexander, Carlos, Carol, Sandy, Ana, Natasha, Adrián, Edwin y Steven por acompañarme en estos semestres y compartir valiosos momentos tanto en lo académico como en lo personal que me han dejado un gran aprendizaje.

Consuelo Alejandra Escobar Jurado

AGRADECIMIENTO

A mi tutor de Trabajo de Integración Curricular, Ing. Xavier Alexander Calderón Hinojosa M.Sc , por su guía y enseñanza para la elaboración de este trabajo.

Al Ing. Danilo Pilacúan por su apoyo con respecto al desarrollo de las etiquetas del teclado.

A Diana Cerón, Diana Torres e Isaac Torres por su apoyo como voluntarios en este proyecto.

A mis profesores por todo el conocimiento impartido en clases, por inspirarme a buscar más allá de lo aprendido y aplicar estos conocimientos en el diario vivir.

Al Ing. Pablo Hidalgo M.Sc, por haberme apoyado desde el primer día que fue asignado como mi tutor y durante toda mi carrera universitaria. Por sus enseñanzas como profesor y por su calidad de ser humano que se ha vuelto un gran ejemplo para mí.

A mis padres y hermanos por su comprensión, amor y paciencia durante todo este trayecto.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	VI
ÍNDICE DE CÓDIGOS	VII
ÍNDICE DE TABLAS	VIII
RESUMEN.....	IX
ABSTRACT.....	X
1. INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO	3
1.4.1 Alfabeto Braille	3
1.4.2 Dispositivos de entrada para personas con discapacidad visual.....	4
1.4.2.1 Definición	4
1.4.2.2 Teclados de gran tamaño.....	5
1.4.2.3 Teclados con braille	5
1.4.3 Programas CAD	7
1.4.3.1 Definición	7
1.4.3.2 AutoCAD	8
1.4.3.3 Blender.....	9
1.4.3.4 SolidWorks	10
1.4.3.5 FreeCAD	11
1.4.3.6 OpenSCAD	11
2. METODOLOGÍA.....	13
2.1 REQUERIMIENTOS	13
2.2 DISEÑO	13
2.3 IMPLEMENTACIÓN.....	34

2.4	PRUEBAS.....	36
3.	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	39
3.1	RESULTADOS.....	39
3.1.1	VALIDACIÓN DE REQUERIMIENTOS.....	39
3.2	CONCLUSIONES	41
3.3	RECOMENDACIONES	42
4.	REFERENCIAS BIBLIOGRÁFICAS.....	43
5.	ANEXOS	45

ÍNDICE DE FIGURAS

Figura 1.1	Signo generador de Braille [3].	3
Figura 1.2	Alfabeto Braille [3].	3
Figura 1.3	Teclado de gran tamaño Nuklz N [8].	5
Figura 1.4	Teclado de celdas 1x4 [9].	5
Figura 1.5	Teclado Braille to text [10].	6
Figura 1.6	Módulo Braille numérico [11].	6
Figura 1.7	Logickeyboard [12].	7
Figura 1.8	Clasificación de programas CAD [15].	8
Figura 1.9	AutoCAD [18].	9
Figura 1.10	Modelado 3D AutoCAD [16].	9
Figura 1.11	Blender [20].	10
Figura 1.12	SolidWorks [22].	10
Figura 1.13	FreeCAD [24].	11
Figura 1.14	Área de trabajo OpenSCAD [26].	12
Figura 1.15	Modelado 3D con OpenSCAD [26].	12
Figura 2.1	Teclado multimedia Meetion K600M [27].	13
Figura 2.2	Teclas de alfabeto diseñadas en Braille [28].	14
Figura 2.3	Teclas de números diseñadas en Braille [28].	14
Figura 2.4	Teclas de símbolos diseñadas en Braille [28].	14
Figura 2.5	Teclas de F1 a F10 diseñadas en Braille [28].	15
Figura 2.6	Teclas de comandos diseñadas en Braille.	15
Figura 2.7	Teclas diseñadas en Braille [28].	15

Figura 2.8 Teclas multimedia de teclado Meetion diseñadas con OpenSCAD.	22
Figura 2.9 Etiquetas multimedia diseñadas con OpenSCAD.....	24
Figura 2.10 Etiquetas adicionales para teclado Meetion diseñadas con OpenSCAD.....	27
Figura 2.11 Etiqueta letra d con puntos guía en Braille.	30
Figura 2.12 Etiqueta letra d sin puntos guía en Braille.	31
Figura 2.13 Etiquetas de alfabeto sin puntos guía en Braille.	31
Figura 2.14 Etiquetas de números sin puntos guía en Braille.	31
Figura 2.15 Etiquetas de símbolos sin puntos guía en Braille.	31
Figura 2.16 Etiquetas de funciones sin puntos guía en Braille.	32
Figura 2.17 Etiquetas de comandos sin puntos guía en Braille.	32
Figura 2.18 Etiquetas de símbolos adicionales sin puntos guía en Braille.....	32
Figura 2.19 Etiquetas multimedia sin puntos guía en Braille.	32
Figura 2.20 Etiquetas letras F y J sin puntos guía en Braille.	33
Figura 2.21 Implementación de etiquetas Braille diseñadas.....	35
Figura 2.22 Etiquetas teclado multimedia con símbolo.....	35
Figura 2.23 Etiquetas teclado multimedia en Braille.	35
Figura 2.24 Etiquetas de teclado en Braille sin puntos guía.	36
Figura 2.25 Etiquetas de teclado multimedia en Braille sin puntos de apoyo.	36
Figura 2.26 Etiquetas de F y J subrayadas.	36
Figura 2.26 Pruebas con voluntarios.....	37
Figura 2.27 Pruebas con voluntarios.....	37
Figura 2.28 Pruebas con voluntarios.....	38

ÍNDICE DE CÓDIGOS

Código 2.1 Dimensiones de etiquetas [28].	16
Código 2.2 Números en Braille [28].....	16
Código 2.3 Dibujo de números en 3D [28].....	17
Código 2.4 Módulo drawTile [28].....	17
Código 2.5 Módulo backplate [28].....	17

Código 2.6 Arreglo de etiquetas de comandos.....	18
Código 2.7 Módulo drawDoubleTile [28].....	18
Código 2.8 Dibujo de comandos en 3D [28].	19
Código 2.9 Etiquetas con símbolo de teclas multimedia.....	22
Código 2.10 Arreglo etiquetas en Braille de teclas multimedia.	24
Código 2.11 Dibujo de etiquetas en Braille de teclas multimedia.	24
Código 2.12 Dibujo de etiquetas adicionales para el teclado Meetion.	30
Código 2.13 Dibujo de línea bajo letra en Braille.	34

ÍNDICE DE TABLAS

Tabla 2.1 Teclas multimedia del teclado Meetion.	19
Tabla 2.2 Etiquetas en Braille adicionales para teclado Meetion.	23
Tabla 2.3 Etiquetas en Braille adicionales para teclado Meetion.	25
Tabla 5.1 Alfabeto en Braille.	45
Tabla 5.2 Números en Braille.	47
Tabla 5.3 Signos en Braille.	48
Tabla 5.4 Funciones en Braille	49
Tabla 5.5 Comandos en Braille.	50

RESUMEN

En la actualidad, el uso de dispositivos de entrada como teclado para un computador es de uso cotidiano. En este Trabajo de Integración Curricular se indica el diseño de teclas en braille mediante el programa CAD OpenSCAD de un teclado estándar para personas con discapacidad visual. El presente documento consta de 3 capítulos. Para el primer capítulo se aborda el estudio de dispositivos de entrada existentes para personas con discapacidad visual, alfabeto Braille y programas CAD existentes. En el segundo capítulo se muestran los requerimientos, diseño de teclas en 3D mediante el programa CAD OpenSCAD, implementación y pruebas de funcionalidad. En el tercer capítulo se indican los resultados obtenidos, el análisis de las encuestas realizadas a las personas con discapacidad visual, conclusiones y recomendaciones.

PALABRAS CLAVE: Discapacidad visual, Teclado, Programa CAD, Braille, OpenSCAD.

ABSTRACT

Actually, the use of input devices as a keyboard for a computer is considered in daily life. In this Curricular Integration Work shows the design of keycaps in Braille for a standard keyboard with OpenSCAD, a CAD software, oriented to blind people. This document has 3 chapters. The first chapter describes the study of input devices for blind people, Braille alphabet and CAD software. The second chapter explains requirements, design of 3D keycaps with CAD software OpenSCAD, implementation and functionality tests. In the third chapter, obtained results, analysis of surveys for blind people, conclusions and recommendations.

KEYWORDS: Blind people, Keyboard, CAD software, Braille, OpenSCAD.

1. INTRODUCCIÓN

Los dispositivos de entrada para personas con discapacidad visual que existen en la actualidad como teclados permiten el uso del computador con mayor facilidad. En el Ecuador existen 55478 personas con discapacidad visual, de las cuales 3813 personas estudian y 10981 trabajan, en el año 2023 [1]. El equipamiento tecnológico en hogares ecuatorianos en el mismo año es del 33.1% [2]. Esto indica que el equipamiento en hogares continúa incrementando y se ha vuelto indispensable el uso de un computador en el hogar debido a las funciones que permite realizar como acceder a archivos, crear y modificar documentos, acceder a Internet, utilizar programas, etc. Sin embargo, la cantidad de teclados orientados a personas con discapacidad visual es baja con respecto a teclados estándar que se encuentran en el mercado.

Los programas CAD para modelado en 3D permiten desarrollar prototipos en todo ámbito ya sea para ingeniería, arquitectura, etc. Con esto, se puede mejorar el diseño antes de ser implementado físicamente y posteriormente realizar pruebas de diseño. Como se puede observar es una gran ventaja ya que permite ahorrar recursos. Existen programas CAD de licencia gratuita o de pago, de los cuales se muestran características para determinar el programa más adecuado para realizar el diseño en 3D.

El alfabeto Braille es conocido por personas con discapacidad visual para la lectura y escritura, está conformado por seis puntos, se puede representar el alfabeto español y caracteres especiales que pueden ser utilizados para representar cada una de las teclas en el teclado a diseñar.

Como alternativa respecto a adquirir un teclado orientado a personas con discapacidad visual, el prototipo de etiquetas de teclas a desarrollar en el presente Trabajo de Integración Curricular muestra el diseño de teclas en Braille para un teclado con diseño 3D mediante el programa CAD OpenSCAD. Para la implementación, se realizará la impresión de cada una de las teclas diseñadas, las cuales podrán ser añadidas sobre cada tecla de un teclado estándar, para esto se tendrá en cuenta las dimensiones de las teclas como ancho, largo y espacio entre teclas. Se realizarán pruebas de funcionalidad por parte de voluntarios con discapacidad visual. Con esto, se mejorará la adaptación de un teclado estándar para ser utilizado por personas con discapacidad visual y se incrementa el uso de teclados existentes en el hogar.

1.1 OBJETIVO GENERAL

Desarrollar un prototipo de teclas de un teclado para personas con discapacidad visual.

1.2 OBJETIVOS ESPECÍFICOS

1. Analizar dispositivos de entrada como teclados para personas con discapacidad visual.
2. Diseñar etiquetas de teclas para el teclado mediante un programa CAD.
3. Evaluar el funcionamiento del prototipo de teclas diseñado.

1.3 ALCANCE

El propósito del proyecto de Trabajo de Integración Curricular es desarrollar un prototipo de etiquetas de teclas para un teclado dirigido a personas con discapacidad visual.

Fase de planteamiento:

En esta fase se realizará el estudio de dispositivos de entrada existentes para personas con discapacidad visual. Estudio del alfabeto braille, historia y representación de letras, números y signos de puntuación. Finalmente, el estudio de programas CAD de objetos 3D para realizar el diseño de las etiquetas de teclas para teclado.

Fase de implementación:

En esta fase se diseñará cada una de las etiquetas de teclas para personas con discapacidad visual mediante el uso del software de diseño de objetos 3D seleccionado. Una vez obtenidas las etiquetas se realizará el armado que constituye en agregar la etiqueta de la tecla modelada a la tecla según el teclado seleccionado.

Fase de evaluación y análisis de resultados:

Se realizarán pruebas del funcionamiento de las teclas en el teclado a un grupo de personas con discapacidad visual. Para el análisis de los resultados obtenidos se utilizarán encuestas (al menos 3) dirigidas a las personas con discapacidad visual que participarán en el proyecto.

1.4 MARCO TEÓRICO

1.4.1 Alfabeto Braille

Fue desarrollado por Louis Braille con la finalidad de que las personas con discapacidad visual tengan acceso a la información. Está conformado por una casilla de 6 puntos, 2 columnas 3 filas, como en Figura 1.1. Cada combinación representa letras, signos y números. El alfabeto es universal, esto significa que se utiliza en todo el mundo para diferentes aplicaciones: lecturas en libros, señalización en espacios públicos como paradas, estaciones y espacios interiores de estas construcciones para facilitar el acceso [3].

Para las medidas estándar de una celda del alfabeto Braille se debe tener en cuenta [3]:

- Ancho: 3.6 y 4.65 mm.
- Alto: 6 y 6.9 mm.
- Relieve: 0.2 y 0.5 mm.

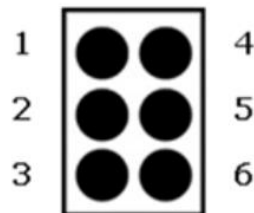


Figura 1.1 Signo generador de Braille [3].

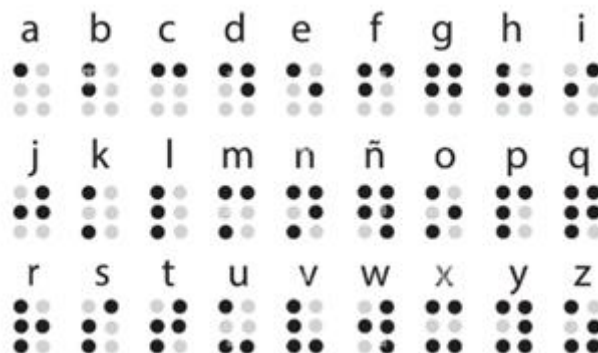


Figura 1.2 Alfabeto Braille [3].

El alfabeto Braille como se puede observar en Figura 1.2 está formado por series [4]:

- Primera serie: Con los puntos superiores 1,2,4 y 5 se forman las letras del alfabeto de a – j.
- Segunda serie: Los puntos a utilizar son del literal anterior con adición el punto 3, las letras a representar son de k – t.
- Tercera serie: Está formada por los puntos de la segunda serie con adición el punto 6. Las letras a representar son: u, v, x, y, z, ç, signo generador, á, é, ú.
- Cuarta serie: Formada con los puntos de la primera serie con adición el punto 6. Las letras a representar son ñ, ü, w.
- Quinta serie: La serie está formada por los puntos 2, 3, 5, 6 con la misma distribución que en la primera serie.

1.4.2 Dispositivos de entrada para personas con discapacidad visual

1.4.2.1 Definición

Los dispositivos conectados a un computador son conocidos como periféricos, permiten de acuerdo a su clasificación realizar funcionalidades en el computador. Los dispositivos de entrada efectúan el ingreso de datos, letras o caracteres al computador, para esto, se utiliza un teclado, pantallas táctiles, mouse, etc. Los dispositivos de salida muestran información ingresada en el computador al usuario, por ejemplo: la impresora o monitor de pantalla. Existen dispositivos que realizan ambas funcionalidades de entrada y salida como una pantalla táctil [5].

Se han realizado adaptaciones de estos dispositivos para personas con diferentes tipos de discapacidad: física, intelectual, auditiva, visual, psicosocial y lenguaje [6].

Con la finalidad de disminuir la brecha de acceso a la tecnología. En este caso, se realizará un enfoque en los dispositivos para personas con discapacidad visual. Existen variedades de estos de estos teclados con teclas grandes para personas con baja visión o teclados con braille [7].

1.4.2.2 Teclados de gran tamaño

Estos teclados poseen un gran tamaño respecto a las teclas que conforman el teclado. Está dirigido para personas que son parcialmente ciegas.



Figura 1.3 Teclado de gran tamaño Nukiz N [8].

1.4.2.3 Teclados con braille

- **Teclado de celdas 1x4:** Este teclado está conformado por 3 celdas de teclas 1x4, la celda horizontal representa los números del 1 al 4 donde el valor de 1 muestra la tecla enter, 2 la tecla de espaciado, 3 representa menos y 4 la coma. Para las celdas verticales cuya numeración es opuesta, las dos primeras teclas representan espacio y punto, las 6 teclas resultantes de ambas celdas representan a una célula braille. El propósito es mejorar la escritura de mensajes ya sea de correo electrónico como aplicaciones de Word [9].

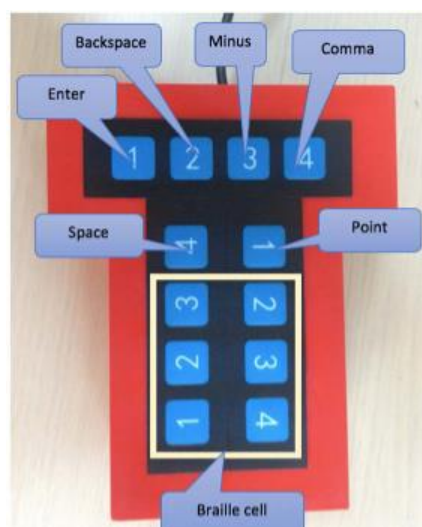


Figura 1.4 Teclado de celdas 1x4 [9].

- **Braille to text:** El teclado está construido por 8 teclas donde 6 corresponden a cada uno de los puntos en lenguaje Braille, 1 tecla para espacio y 1 tecla para limpieza. Las seis teclas se encuentran distribuidas según el orden de los caracteres Braille. Además, posee un pad adicional para lectura de texto a Braille [10].

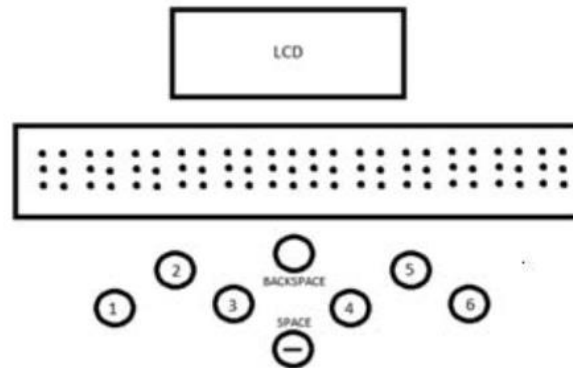


Figura 1.5 Teclado Braille to text [10].

- **Módulo Braille numérico para el aprendizaje de operaciones matemáticas simples:** Este módulo está formado por un teclado 4x4 con teclas en Braille que representan a los números del 0 al 9 con los operadores de suma, resta y multiplicación. El teclado funciona como dispositivo de entrada para para un módulo conformado por 12 solenoides y controlado por medio de Arduino Uno [11].

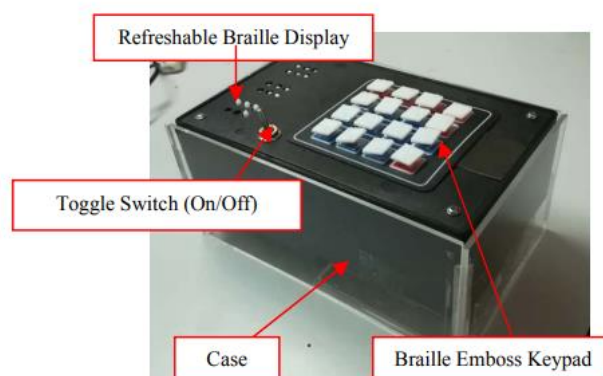


Figura 1.6 Módulo Braille numérico [11].

- **Teclado Braille Logickeyboard:** El teclado está conformado por teclas que han sido diseñadas con alfabeto braille de seis puntos, posee 108 teclas que representan a cada una de las teclas y también posee teclas para control de volumen [12].



Figura 1.7 Logickeyboard [12].

1.4.3 Programas CAD

1.4.3.1 Definición

Las siglas CAD provienen de Computer Aided Design o Diseño Asistido por Ordenador, estos programas son conocidos en varias ramas de ingeniería para implementar el diseño de representaciones en 2D o 3D de forma computacional [13].

Los primeros sistemas CAD fueron introducidos en los años 1960. Los computadores podían realizar más cálculos de ingeniería y apoyo en resolución de problemas. Sketchpad es un programa desarrollado en aquella época por el MIT que muestra una interacción a tiempo real de dibujo en esta máquina debido a que se podía dibujar en la pantalla, otras funcionalidades como definir sombra, límites y relaciones entre objetos han sido considerados como modelado con parámetros en los programas CAD [14].

Durante los años 1960 a 1970, se desarrollaron los primeros programas CAD comerciales como PRONTO y ADAM. Durante los años 1980 a 1990 se añadieron las funcionalidades de modelado 3D para la representación de diseños, uno de los programas CAD desarrollados fue SolidWorks en 1995. Para los años 2000 a 2010, las funcionalidades agregadas son la simulación del comportamiento, lo cual disminuye los prototipos en físico. Para los años 2010 y actualidad, se añaden funcionalidades como realidad aumentada e impresión 3D, se desarrollan programas CAD basados en la nube lo cual incrementa la colaboración en el diseño. En el futuro se busca añadir funcionalidades de IA (Inteligencia Artificial) y ML (Machine Learning) para automatizar tareas de diseño y optimización [14].

Los principales proveedores de este tipo de software son: Autodesk, Dassault Systems y Siemens en el año 2013. Programas con mayores funcionalidades implican que se requiere de un entrenamiento comprensivo y años de experiencia para ser utilizados y aprovechados de la mejor manera. También, existen programas con menores o limitadas funcionalidades orientados a pequeñas y medianas empresas que requieren cursos introductorios para ser utilizados [15].

Para la clasificación de los programas CAD se puede realizar de acuerdo a la aplicación y al tipo de licenciamiento.

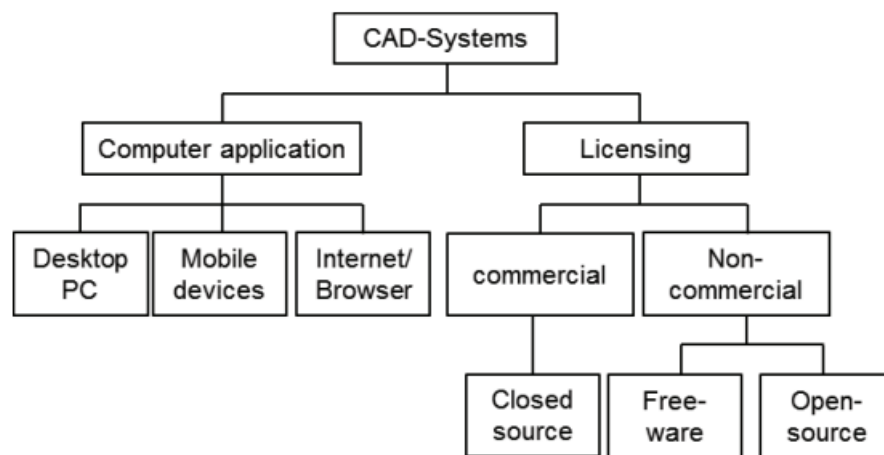


Figura 1.8 Clasificación de programas CAD [15].

Con el uso de estos programas se pueden observar ventajas como: disminución en el precio para productos, mejoras en la producción, documentación acerca del programa para realizar el diseño, etc [13].

1.4.3.2 AutoCAD

Fue desarrollado en 1982 por AutoDesk, una empresa multinacional. Es compatible para Windows y iOS [16].

AutoCAD tiene características como: realizar diseños en 2D y 3D, automatizar tareas y personalizar el entorno de trabajo para integrar APIs, es una herramienta con licencia de pago, permite la colaboración entre más personas con el uso de diferentes dispositivos ya sean de escritorio, web o móviles. La versión más actual es AutoCAD 2024. Posee

herramientas como: Architecture, Mechanical, Electrical, Map 3D, MEP, Plant 3D, Raster Design, etc. [17]



Figura 1.9 AutoCAD [18].

Utiliza cuatro tipos de modelado 3D: estructura alámbrica, sólido, superficie y malla.

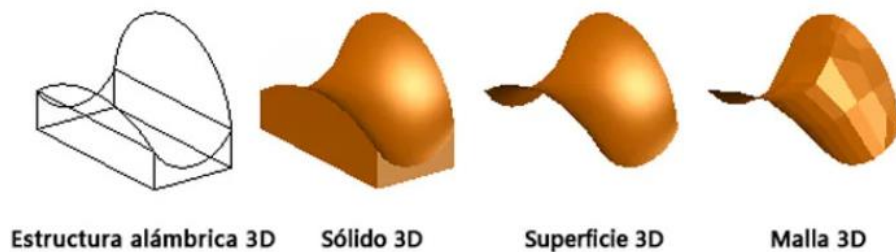


Figura 1.10 Modelado 3D AutoCAD [16].

1.4.3.3 Blender

Fue desarrollado en 1995 por NeoGeo, sin embargo, fue presentado al público en 2004. Es una herramienta gratuita, open-source. Permite realizar modelado en 3D y animaciones [19]. Está disponible en plataformas como Linux, iOS, Windows, Android. Utiliza un modelo de malla avanzado y esculpido [20].

Las técnicas de modelado 3D que utiliza son [19]:

- Extrusión y biselado: generar varias formas por medio de la selección de una cara borde en este programa y el biselado le brinda mayor profundidad.
- Ciclo de cortes y ciclo de bordes: permiten brindar mayor detalle a los modelos. En este caso, ciclo de cortes brinda mayor forma y estructura. Ciclo de bordes son bordes conectados que pueden integrar al modelo.

- Edición proporcional: permite ajustar el diseño sin modificar las proporciones del objeto, vértices, bordes o formas.
- Esculpir: modifica la forma del modelo con técnicas similares a las tradicionales con las herramientas de este programa.

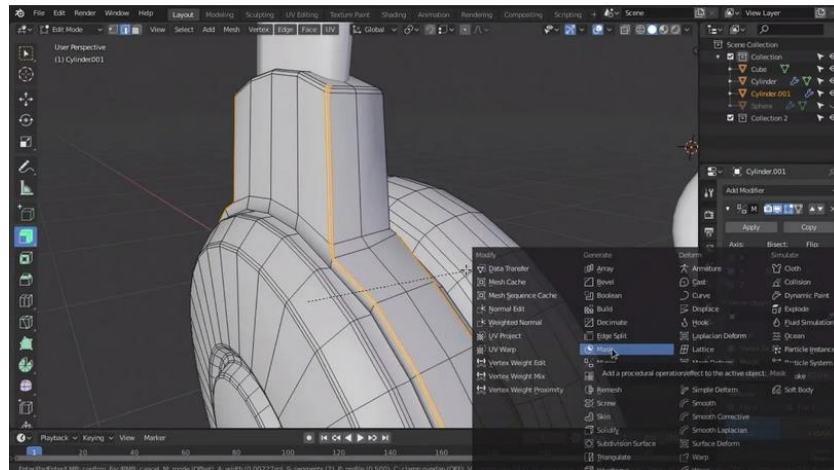


Figura 1.11 Blender [20].

1.4.3.4 SolidWorks

Fue desarrollado en 1995 por Jon Jirschtick pertenece a la marca Dassault Systèmes que se enfoca en soluciones 3D. Los dibujos son creados por modelos o vistas en el archivo de documento. Muestra la asociación entre ensamblajes y dibujos en el proceso de diseño. Es una herramienta con licencia de pago [21].

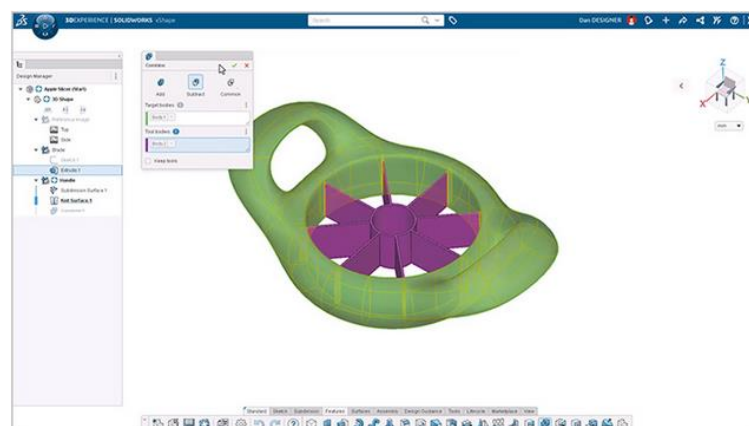


Figura 1.12 SolidWorks [22].

Está disponible para Windows, utiliza modelado paramétrico, esto significa que los valores de relaciones y cotas se almacenan en el modelo. Para el proceso de diseño se realiza un

croquis 2D con líneas, círculos y triángulos. Para el modelado 3D se utiliza operaciones de extrusión, revolución, redondeo, etc. En el ensamblaje se puede agregar más piezas para realizar análisis de movimiento. Para visualizar el ensamblado se realiza un cambio al módulo de dibujo que permite identificar diferentes vistas del diseño. Con esta herramienta se disminuyen los costos de fabricación de un producto en específico. Posee tres versiones: Standard, Profesional y Premium [23].

1.4.3.5 FreeCAD

Es una herramienta gratuita de código abierto (open source). Es multiplataforma: Windows, iOS y Linux. Permite utilizar varios formatos abiertos como: STEP, IGES, STL, SVG, DXF, OBJ, IFC, DAE, etc. Utiliza un motor de geometría basado en la tecnología Open Cascade. Los objetos en FreeCAD son paramétricos, esto indica que cada objeto posee propiedades numéricas, de texto y otros objetos. Su última versión estable es la 0.21 [24].

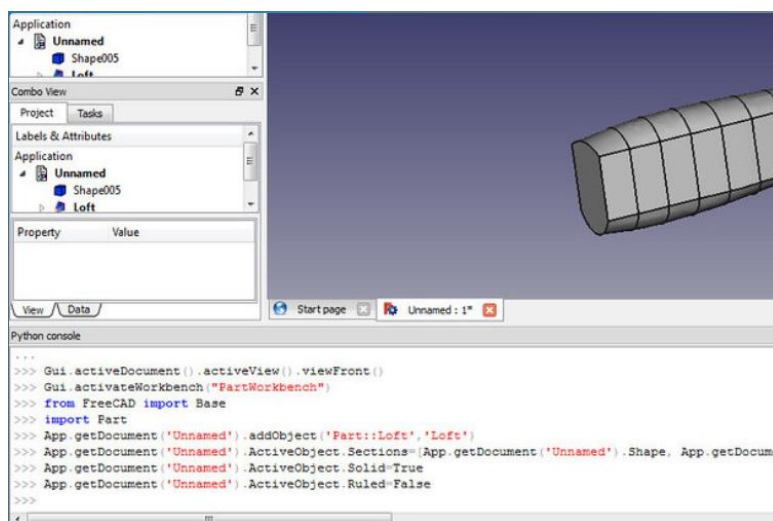


Figura 1.13 FreeCAD [24].

1.4.3.6 OpenSCAD

Es una herramienta de software gratuito para modelado en 3D. Permite realizar el diseño de objetos mediante un script que contiene los parámetros, funciones y módulos respectivos para posteriormente indicar el diseño en 3D al realizar el renderizado. Utiliza dos técnicas para el modelado: CSG conocida como Geometría Sólida Constructiva y extrusión de delineado 2D. Se encuentra disponible para Unix (Linux), Windows y MAC OS [25].

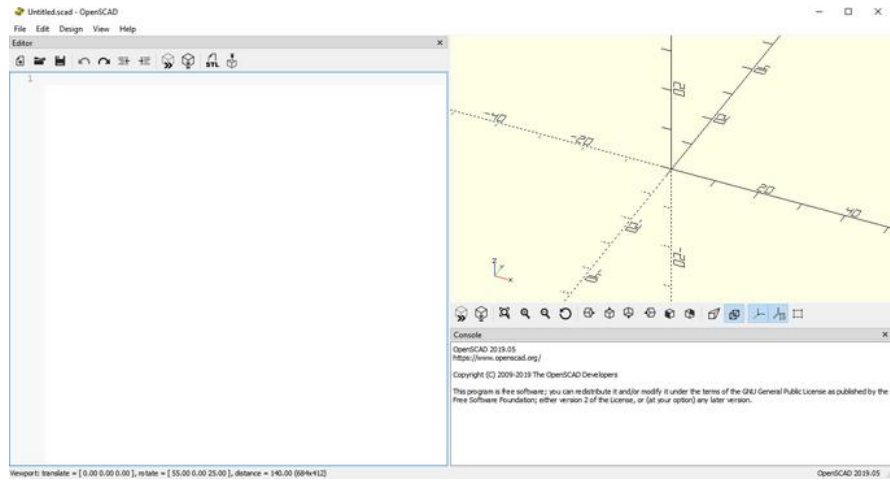


Figura 1.14 Área de trabajo OpenSCAD [26].

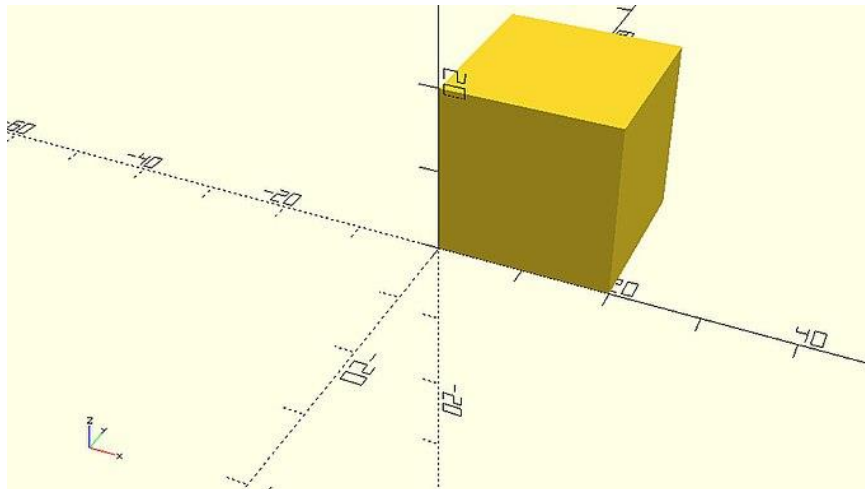


Figura 1.15 Modelado 3D con OpenSCAD [26].

2. METODOLOGÍA

En este capítulo se muestran los requerimientos, diseño, implementación y pruebas. En la sección 2.1 se indica los requerimientos para realizar el diseño de las etiquetas de teclado, en la sección 2.2 se muestra el diseño mediante el software de modelado 3D OpenSCAD y selección del teclado. En la sección 2.3 se describe la implementación del diseño de etiquetas con impresión 3D y su colocación en el teclado. En la sección 2.4 se muestran las pruebas realizadas a los voluntarios.

2.1 REQUERIMIENTOS

Los requerimientos para las etiquetas de teclado son:

- Las etiquetas deben incluir lenguaje Braille para que se puedan identificar las teclas.
- Las etiquetas deben incorporar diseño de teclas multimedia para control de sonido como volumen, pausa, acceso a aplicaciones, etc. según el diseño del teclado.
- Las etiquetas deben brindar facilidad de uso en el teclado.
- Las etiquetas deben permitir a los usuarios que se adapten rápidamente a su uso.

2.2 DISEÑO

Para la selección del teclado, se debe tener en cuenta las siguientes características: tipo de teclado membrana, distribución de teclas QWERTY, cable de conexión USB y teclas multimedia que muestran mayor facilidad en accesibilidad a funciones adicionales como subir volumen, bajar volumen, pausar, reproducción, etc. El teclado seleccionado que cumple con las características anteriormente mencionadas es Meetion K600M como se puede observar en Figura 2.1.



Figura 2.1 Teclado multimedia Meetion K600M [27].

Las dimensiones de las bases de todas las etiquetas son de 11 mm de largo y ancho, la altura es de 0.25 mm, estas mediciones son correspondientes al tamaño de las teclas de teclado Meetion seleccionado.

El programa seleccionado para el diseño de teclas es OpenSCAD debido a la ventaja principal que presenta como es el modelado 3D por medio del uso de un script. Para el diseño de teclas en 3D se utilizó como base el diseño de etiquetas en OpenSCAD de davidakerr con licencia CC by SA que permite compartir y adaptar el diseño compartido por el autor. Contiene 26 letras como se puede observar en Figura 2.2.

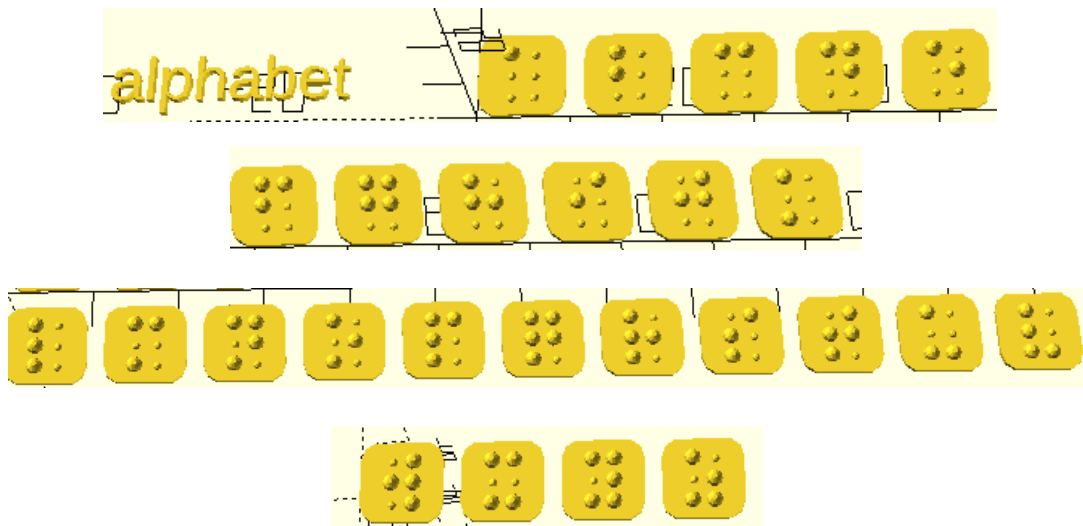


Figura 2.2 Teclas de alfabeto diseñadas en Braille [28].

Contiene 10 números del 0 al 9 como en Figura 2.3.

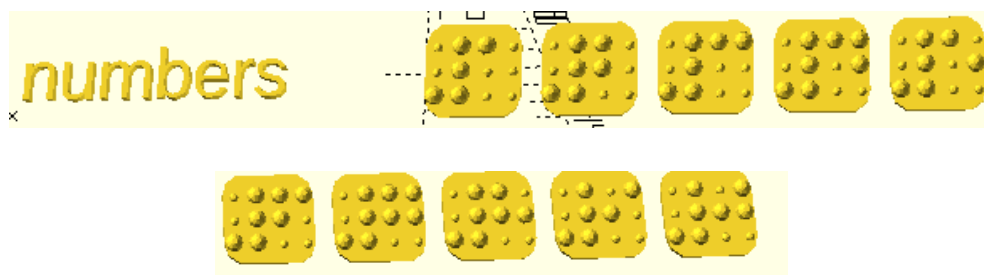


Figura 2.3 Teclas de números diseñadas en Braille [28].

Contiene 8 símbolos como “;”, “(”, “)” como en Figura 2.4.



Figura 2.4 Teclas de símbolos diseñadas en Braille [28].

Además 10 funciones para F1 a F9 como en Figura 2.5.

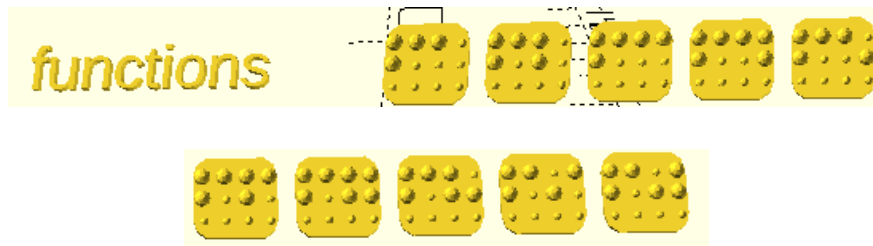


Figura 2.5 Teclas de F1 a F10 diseñadas en Braille [28].

Contiene 16 de comandos como ESC, Barra espaciadora como en Figura 2.6.

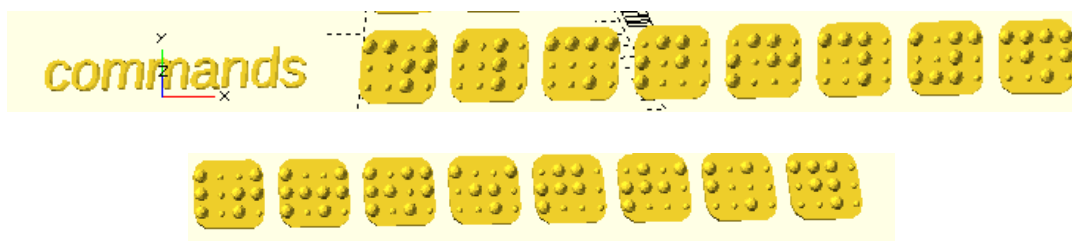


Figura 2.6 Teclas de comandos diseñadas en Braille [28].

Posee 3 etiquetas de notación adicionales como “°” en Figura 2.7. Las etiquetas con sus respectivas equivalencias en Braille se encuentran en Anexo I.



Figura 2.7 Teclas diseñadas en Braille [28].

Como modificaciones al diseño base, se indica el cambio del tamaño de etiqueta a 11 mm y la altura de 0.25 mm para adaptar a las dimensiones del teclado Meetion como se muestra en el Código 2.1.

```
// Thickness of the bottom base in mm
baseHeight_text_box = 0.25;

// Key Size in mm
keySize_text_box = 11;

// Key corner rounding radius in mm (min 2.2)
cornerRadius_text_box = 2.2;
```

```
// Dot Size Multiplier
dotRadius = 0.7;
// How much to raise the braille mm
dotDepth = 1;
```

Código 2.1 Dimensiones de etiquetas [28].

Para la sección de números, cada número fue almacenado en un arreglo de seis posiciones donde la primera posición inicia por la primera fila primera columna, seguida de la segunda fila primera columna, tercera fila primera columna, primera fila segunda columna, segunda fila segunda columna y tercera fila segunda columna. En la definición del tamaño de los puntos son arreglos en los ejes (x, y, z), existen 3: el tamaño de $_0 = [0,0,0]$, $_1 = [\text{dotDepth} / 2, \text{dotRadius} / 0.5, \text{dotRadius} * 0.3]$ y $_2 = [\text{dotDepth}, \text{dotRadius}, 0]$. Los puntos guía son $_1$ y los puntos en Braille son $_2$ [28] como se muestra en Código 2.2.

```
//Sección números
// 1 2 3 4 5 6
numbersign = [_1, _1, _2, _2, _2, _2];
n1 = [_2, _1, _1, _1, _1, _1];
n2 = [_2, _2, _1, _1, _1, _1];
n3 = [_2, _1, _1, _2, _1, _1];
n4 = [_2, _1, _1, _2, _2, _1];
n5 = [_2, _1, _1, _1, _2, _1];
n6 = [_2, _2, _1, _2, _1, _1];
n7 = [_2, _2, _1, _2, _2, _1];
n8 = [_2, _2, _1, _1, _2, _1];
n9 = [_1, _2, _1, _2, _1, _1];
n0 = [_1, _2, _1, _2, _2, _1];
numbers = [n1, n2, n3, n4, n5, n6, n7, n8, n9, n0];
```

Código 2.2 Números en Braille [28].

En la parte de dibujar la tecla, se coloca el arreglo símbolo de número (numbersign) con los números almacenados en los arreglos numbers, lo cual genera una etiqueta con dos letras formada por tres filas y cuatro columnas como se observa en Código 2.3.

```
// DRAW NUMBERS
if(setSelection == "numeric" || setSelection == "all"){
    renderLabel(yPositionModifier(3), "numbers");
    for(i = [0:len(numbers) - 1]){
```

```

drawDoubleTile(numbersign, numbers[i], i, yPositionModifier(3));
}
}

```

Código 2.3 Dibujo de números en 3D [28].

Para las etiquetas que poseen 2 filas 3 columnas, esto indica que es una letra, el módulo para el dibujo de la base con puntos en Braille es drawTile() en Código 2.4. Las etiquetas que utilizan este componente son: alfabeto y símbolos.

```

module drawTile(symbol, xOffset, yOffset){
  translate([(xOffset * (keySize + tileSpacing)), -yOffset * (keySize + tileSpacing), 0]){
    backplate();
    intersection(){
      Char_display_single(symbol);
      translate([0, 0, -baseHeight / 100])scale([1, 1, 10 * scaleBraille])backplate();
    }
  }
}

```

Código 2.4 Módulo drawTile [28].

El módulo drawTile() utiliza la función backplate() para realizar el dibujo de la base formada por un cubo y un cilindro, para agregar estas figuras y se genere la forma de tecla, se utiliza la función minkowski() en Código 2.5.

```

module backplate(){
  translate([cornerRadius, cornerRadius, 0]){
    minkowski(){
      wtf = keySize - cornerRadius * 2;
      cube(size = [wtf, wtf, 0.1]);
      cylinder(h = baseHeight, r = cornerRadius);
    }
  }
}

```

Código 2.5 Módulo backplate [28].

Para el caso de comandos que poseen etiquetas formadas por dos letras, por ejemplo: shift = [s, h], se toman las primeras letras de la palabra que indica la flecha, la tecla se agrupa en el arreglo de comandos formado por las etiquetas de comandos como se indica en Código 2.6.

```

controlKey = [c, t];
altKey = [a, l];
cmdKey = [c, m];
spacebar = [s, b];
shiftKey = [s, h];
capslock = [c, l];
up = [u, p];
down = [d, n];
left = [l, t];
right = [r, t];
returnKey = [r, n];
esc = [e, s];
tab = [t, b];
tilde = [t, i];
backspace = [b, k];
del = [d, l];

commands = [controlKey, altKey, cmdKey, spacebar, shiftKey, capslock, up, down, left,
right, returnKey, esc, tab, tilde, backspace, del];

```

Código 2.6 Arreglo de etiquetas de comandos.

Con `drawDoubleTile()` se genera la base de cada etiqueta del teclado con su respectivo número en Braille formado por el símbolo para número, arreglo de números y posición en el eje “y” de todas las teclas de esta sección, fila 3 [28]. Para este caso la etiqueta está formada por 4 columnas y 3 filas como se indica en código 2.7. Las etiquetas de funciones y comandos también utilizan este módulo.

```

module drawDoubleTile(symbol1, symbol2, xOffset = 0, yOffset = 0){
  translate([(xOffset * (keySize + tileSpacing)), -yOffset * (keySize + tileSpacing), 0]){
    backplate();
    intersection(){
      Char_display_standard2(symbol1, symbol2);
      translate([0, 0, -baseHeight / 100])scale([1, 1, 10 * scaleBraille])backplate();
    }
  }
}

```

Código 2.7 Módulo `drawDoubleTile` [28].

Para dibujar las teclas formadas por dos palabras, en este caso comandos, se llama al módulo drawDoubleTile() donde se pasa los parámetros símbolo 1 y símbolo 2 de cada etiqueta formada por dos letras, como se puede observar en Código 2.8.


```
// DRAW COMMAND KEYS
if(setSelection == "commands" || setSelection == "all"){
  renderLabel(yPositionModifier(6), "commands");
  for(i = [0:len(commands) - 1]){
    drawDoubleTile(commands[i][0], commands[i][1], i, yPositionModifier(6));
  }
}
```

Código 2.8 Dibujo de comandos en 3D [28].

Como alternativa al diseño de etiquetas en Braille, se realiza el diseño de etiquetas con el símbolo de las teclas multimedia del teclado Meetion según la funcionalidad que realizan como se encuentra en Tabla 2.1. La fuente Segoe UI Symbol contiene todos los símbolos con su equivalente Unicode hexadecimal para representar cada símbolo en la etiqueta con OpenSCAD.

Tabla 2.1 Teclas multimedia del teclado Meetion.

Función	Símbolo	Unicode hexadecimal
Acceso a Inicio	⏴	2302
Acceso a correo	✉	2709
Refrescar	↺ ↻	21b7, 21b7
Buscar	🔍	e2fb
Acceso a música	🎵	266b
Reproducir/Pausar	▶	23ef
Subir volumen	🔊	e247
Bajar volumen	🔇	e2c1
Mute	🔇×	e246

Acceso a marcadores en navegador		e1c1
-------------------------------------	---	------

Por ejemplo para la etiqueta de subir volumen, se definió la posición de la tecla mediante un arreglo en [x,y,z] llamado `posiciones_teclas`, luego se crea otra variable de tipo string para asignar el símbolo llamado `simbolos_teclas` que contiene el valor en Unicode “e246”, a continuación se utiliza la función `translate()` de OpenSCAD para el arreglo de posiciones con el valor `posiciones_teclas` para que la etiqueta se traslade a la posición indicada. Posteriormente, se utiliza el módulo `backplate()` reutilizado del código de davidakerr para generar la forma de la tecla, que consiste en unir dos figuras un rectángulo y un cilindro para que la etiqueta obtenga la forma rectangular con bordes redondeados. Para que el símbolo se ubique en el centro de la tecla se utiliza nuevamente `translate()` con la modificación que se añade el tamaño del símbolo al arreglo `posicion_tecla` asignado anteriormente. Luego se utiliza `rotate()` para cambiar la dirección de símbolo 90° en el eje Z [0,0,90]. Para darle forma en 3D al símbolo, se utilizó la función `linear_extrude()` que permite a partir del símbolo generado en 2D agregar altura para que se genere la forma en 3D. Finalmente, con la función `text()` se genera el símbolo con el tamaño asignado y la fuente, en este caso Segoe UI Symbol. Esto se puede observar en Código 2.9.

```
// Número de fragmentos
$fn = 25;
// Arreglo de posiciones de teclas
posiciones_teclas = [[1, 0, 1], [1, 12, 1], [1, 24, 1], [1, 36, 1], [1, 48, 1], [1, 60, 1], [1, 72,
1], [1, 84, 1], [1, 96, 1]];

// Arreglo de símbolos de teclas
simbolos_teclas = ["\u2302", "\u2709", "\ue2fb", "\ue1c1", "\ue247", "\ue246",
"\u23ef", "\u266b", "\ue2c1"];

// Tamaño de teclas en x,y,z
tamano = [11, 11, 0.25];

// Valor para redondear las teclas
radio = 2.2;
```

```

// Tamaño del símbolo
tamaño_simbolo = 4;
tamaño_texto = 2;

//Módulo para generar la base de las teclas, tomado del código de David Kerr
module backplate(){
  translate([radio
, radio
, 0]){
    minkowski(){
      wtf = tamaño[0] - radio
      * 2;
      cube(size = [wtf, wtf, 0.1]);
      cylinder(h = tamaño[2], r = radio
      );
    }
  }
}

// Bucle para generar las teclas con sus respectivos símbolos.
for (i = [0 : len(posiciones_teclas
) - 1]) {
  pos = posiciones_teclas
[i];
  symbol = simbolos_teclas[i];
  translate(pos) {
    backplate();

    // Generación de símbolos
    translate([tamaño[0]/2, tamaño[1]/2, tamaño[2]])
    rotate([0, 0, 90])
    linear_extrude(height = tamaño_texto)
    text(symbol, size = tamaño_simbolo, halign = "center", valign = "center", font =
"Segoe UI Symbol");
  }
}

```



```

}

//Tecla con flechas circulares izquierda derecha
posicion_tecla1 = [1, 108, 1]; //Tecla izquierda
simbolo_derecha1 = "\u21b7";
translate(posicion_tecla1) //Posición en x,y,z
backplate(); //Componente generador de la base de la etiqueta
translate([posicion_tecla1[0] + tamaño[0]/2, posicion_tecla1[1] + tamaño[1]/2,
posicion_tecla1[2] + tamaño[2]]) //Posición del símbolo
rotate([0, 0, 90]) //Rotación 90°
linear_extrude(height = tamaño_texto) //Extrusión lineal para modelado en 3D
text(simbolo_derecha1, size = tamaño_simbolo, halign = "center", valign = "center",
font = "Segoe UI Symbol"); // Generación de símbolo

simbolo_izquierda1 = "\u21b7"; // Tecla derecha
translate(posicion_tecla1); //Posición en x,y,z
translate([posicion_tecla1[0] + tamaño[0]/2 + 2.5, posicion_tecla1[1] + tamaño[1]/2,
posicion_tecla1[2] + tamaño[2] ]) //Posición símbolo
rotate([0, 0, 270]) //Rotación 90°
linear_extrude(height = tamaño_texto) //Extrusión lineal para modelado en 3D
text(simbolo_izquierda1, size = tamaño_simbolo, halign = "center", valign = "center",
font = "Segoe UI Symbol"); // Generación de símbolo

```

Código 2.9 Etiquetas con símbolo de teclas multimedia.

Al guardar el script se visualiza el diseño de las etiquetas en Código 2.9 como se muestra en la Figura 2.8.



Figura 2.8 Teclas multimedia de teclado Meotion diseñadas con OpenSCAD.

Para el diseño de las teclas multimedia en alfabeto Braille, se realizan modificaciones al diseño base de davidakerr para añadir estas etiquetas. Las etiquetas en Braille adicionales se encuentran en Tabla 2.2.

Tabla 2.2 Etiquetas en Braille adicionales para teclado Meetion.

Etiqueta	Braille
Inicio (H)	
Correo (C)	
Búsqueda (B)	
Carpeta de marcadores (F)	
Subir volumen (S)	
Disminuir volumen (D)	
Mute (N)	
Refrescar (R)	
Reproductor de música (M)	
Reproducir y pausar (P)	

Las etiquetas en Braille para las teclas multimedia generadas con OpenSCAD se encuentran en Figura 2.9.

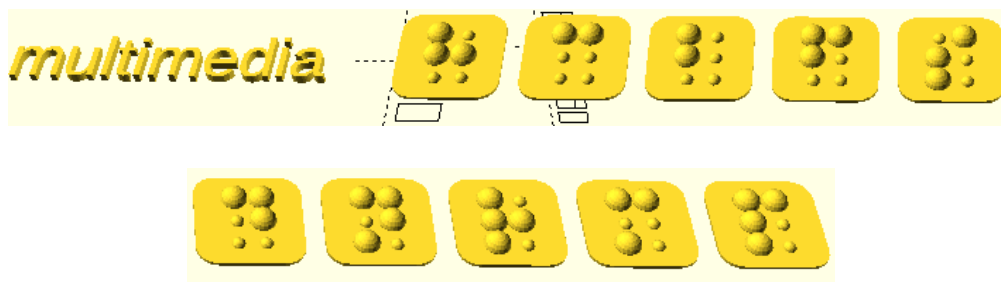


Figura 2.9 Etiquetas multimedia diseñadas con OpenSCAD.

Para generar las etiquetas, se agregaron más líneas de código al diseño base de davidakerr como el arreglo para teclas multimedia que contiene las letras en Braille como se encuentra en Código 2.10 y la función de dibujo de las teclas que genera la base rectangular con bordes redondeados, puntos en Braille y puntos guía en Código 2.11.

```
multimedia = [h,c,b,f,s,d,n,r,m,p];
```















Código 2.10 Arreglo etiquetas en Braille de teclas multimedia.










```
//Modificación multimedia
if(setSelection == "multimedia" || setSelection == "all"){
  renderLabel(yPositionModifier(12), "multimedia");
  for(i = [0:len(multimedia) - 1]){
    drawTile(multimedia[i], i, yPositionModifier(12));
  }
}
```

Código 2.11 Dibujo de etiquetas en Braille de teclas multimedia.

Adicionalmente, se agregaron etiquetas que no se encontraban en el diseño de davidakerr y que se encuentran en el teclado Meetion K600M como: símbolos para realizar operaciones de la parte numérica del teclado como “+”, “-”, “*” y “/”, F10, F11 y F12, las etiquetas para PrintScreen, etc. que fueron diseñadas como modificación al código de davidakerr como se muestra en la Tabla 2.3.

Tabla 2.3 Etiquetas en Braille adicionales para teclado Meetion.

Etiqueta	Símbolo en Braille
F10	
F11	
F12	
#	
\$	
%	
*	
}	
>	
i	
“	
?	
&	
-	

.	
Bloque numérico (BN)	
Imprimir pantalla(PS)	
Bloque desplazamiento(MD)	
Pause Int(PI)	
Insert (IN)	
Fin (FN)	
Inicio (HM)	
Ñ	

La modificación del código de davidakerr incluye teclas duplicadas como Shift, Alt, Ctrl. etc. Con esto se completaron todas las etiquetas de las teclas del teclado como se puede observar en Figura 2.10.

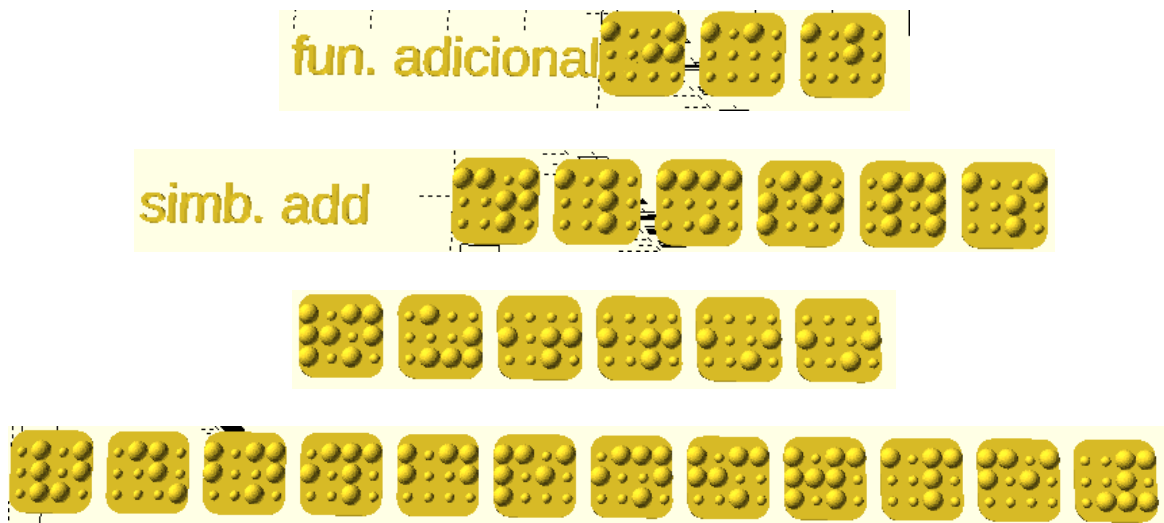




Figura 2.10 Etiquetas adicionales para teclado Meetion diseñadas con OpenSCAD.

Para las etiquetas adicionales, cada símbolo se representa en un arreglo de 6 posiciones como se ha explicado anteriormente. Existen símbolos que se pueden representar con una letra como: “!”, “””, “?””, etc. estos símbolos se agruparon en un arreglo llamado símbolos3 ya que al momento de dibujar la tecla se utiliza el módulo drawTile(). Para símbolos como “#”, “\$”, “+”, bloque numérico, impresión de pantalla, etc. están formados por dos letras, esto indica que utilizan el módulo drawDoubleTile() por lo que son agrupados en el arreglo símbolos2. Las etiquetas de F10, F11 y F12 se agrupan en otro arreglo llamado functions2 que utilizan también el módulo drawDoubleTile como se indica en Código 2.12. El código completo se encuentra en Anexo II.

```
//Modificación !, ", ñ, ?, ´,/, -
exclamation = [_1,_2,_2,_1,_2,_1];
doublecomilla = [_1,_2,_2,_1,_1,_2];
enie = [_2,_2,_1,_2,_2,_2];
question =[_1,_2,_1,_1,_1,_2];
inclinada = [_1,_1,_2,_1,_2,_1];
guion = [_1,_1,_2,_1,_1,_2];
punto = [_1,_2,_1,_1,_2,_2];
// #,$,%,+(2),*(2), }, <>, & Modificación
numeral_i = [_1,_1,_1,_2,_2,_2];
numeral_d = [_2,_1,_1,_2,_2,_2];
numeral = [numeral_i,numeral_d];
dollar = [a, s];
percentage_i = [_1,_1,_1,_2,_1,_2];
```

```

percentage_d = [_1,_1,_2,_1,_2,_2];
percentage = [percentage_i,percentage_d];
plus_i = [_1,_2,_1,_1,_1,_1];
plus_d = [_1,_2,_2,_1,_2,_1];
plus = [plus_i,plus_d];
multiplication_d = [_1,_1,_2,_1,_2,_1];
multiplication = [plus_i, multiplication_d];
llave = [numeral_d,twodota];
greater_i =[_1,_1,_1,_2,_1,_1];
greater_d =[_2,_2,_1,_1,_1,_2];
greater = [greater_i,greater_d];
punto2= [_1,_1,_1,_1,_1,_2];
amp = [_2,_2,_2,_2,_1,_2];
ampersand = [punto2, amp];
// Modificación teclas Bloque num
bloquenum = [b,n];
imprpant = [i,p];
bloquedesp = [b,d];
pauseint = [p,i];
insert = [i,n];
inicio = [h,m];
repag = [r,p];
avpag = [a,p];
fin = [f,i];
//Modificación f10,f11,f12

```

```

functions2 = [n10,n11,n12];

//Otros símbolos

simbolos2 = [controlKey, altKey, cmdKey,shiftKey,numeral,
dollar,returnKey,percentage,plus, plus, multiplication, multiplication, llave, greater,
bloquenum, imprpant, bloquedesp,pauseint, insert, inicio,repag,avpag,fin, ampersand];

simbolos3 = [exclamation, doublecomilla,enie, question, doublecomilla,inclinada, guion
,punto];

//Dibujo de etiquetas

//Modificación functions 2

if(setSelection == "functions2" || setSelection == "all"){

    renderLabel(yPositionModifier(1), "fun. adicionales");

    for(i = [0:len(functions2) - 1]){

        drawDoubleTile(a, functions2[i], i, yPositionModifier(1));

    }

}

//Modificación símbolos

if(setSelection == "simbolos2" || setSelection == "all"){

    renderLabel(yPositionModifier(2), "simb. add");

    for(i = [0:11]){

        drawDoubleTile(simbolos2[i][0],simbolos2[i][1], i, 2);

    }

    for(i = [12:23]){

        drawDoubleTile(simbolos2[i][0],simbolos2[i][1], i - 12, 3);

    }

}

//Modificación símbolos

```



```

if(setSelection == "simbolos3" || setSelection == "all"){
    for(i = [0:len(simbolos3) - 1]){
        drawTile(simbolos3[i], i, yPositionModifier(4));
    }
}
//Modificación multimedia
if(setSelection == "multimedia" || setSelection == "all"){
    renderLabel(yPositionModifier(5), "multimedia");
    for(i = [0:len(multimedia) - 1]){
        drawTile(multimedia[i], i, yPositionModifier(5));
    }
}
}

```

Código 2.12 Dibujo de etiquetas adicionales para el teclado Meetion.

Como se puede observar en todas las etiquetas Braille anteriores, están conformadas por los puntos grandes que representan el alfabeto Braille y los puntos pequeños que se pueden utilizar como puntos guía cuando se tienen representaciones de una letra, dos columnas y tres filas para diferenciar de las formadas por dos letras, esto indica cuatro columnas y tres filas.

Sin embargo, en la sección de pruebas con la primera prueba realizada por los voluntarios se generó confusión respecto al reconocimiento de las etiquetas con los puntos pequeños guía. Por lo que se decidió modificar nuevamente el diseño para eliminar los puntos guía. Por ejemplo: para la letra d los puntos guía representados por `_1` y puntos Braille representados por `_2` son: `d = [_2, _1, _1, _2, _2, _1]` [28] como se muestra en Figura 2.11, con la modificación sin los puntos guía `d = [_2, _0, _0, _2, _2, _0]` como se muestra en Figura 2.12.



Figura 2.11 Etiqueta letra d con puntos guía en Braille.



Figura 2.12 Etiqueta letra d sin puntos guía en Braille.

Con esta aclaración, todas las etiquetas del teclado fueron modificadas sin los puntos guías en su diseño, como se puede observar en Figura 2.13, Figura 2.14, Figura 2.15, Figura 2.16, Figura 2.17, Figura 2.18 y Figura 2.19. El código con las etiquetas modificadas se encuentra en el Anexo II.

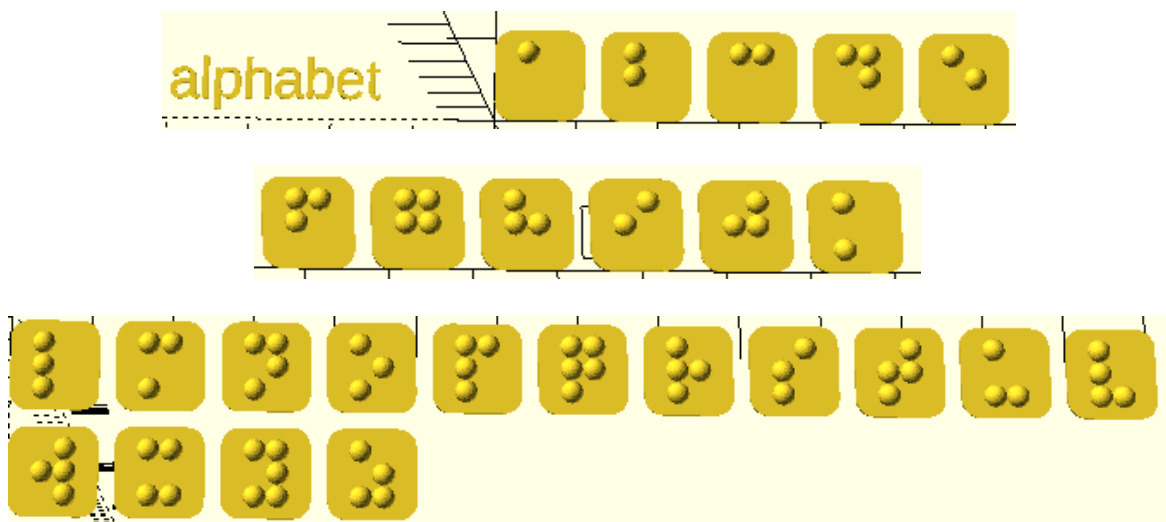


Figura 2.13 Etiquetas de alfabeto sin puntos guía en Braille.

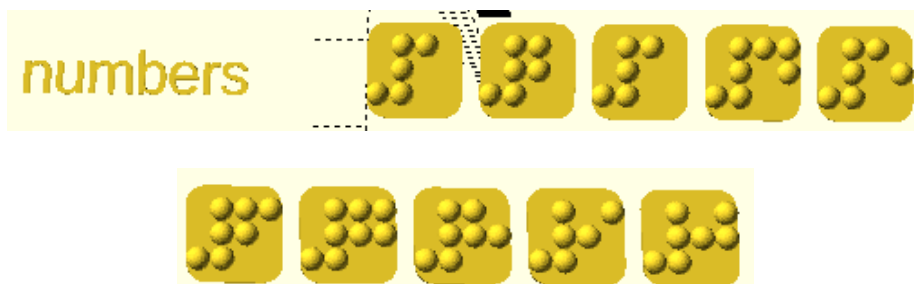


Figura 2.14 Etiquetas de números sin puntos guía en Braille.



Figura 2.15 Etiquetas de símbolos sin puntos guía en Braille.





Figura 2.16 Etiquetas de funciones sin puntos guía en Braille.

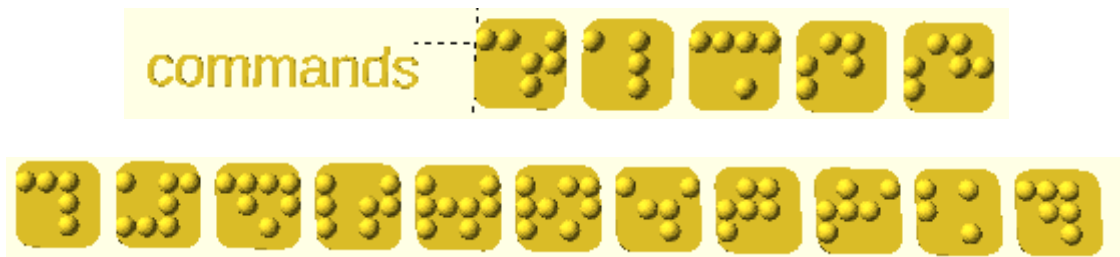


Figura 2.17 Etiquetas de comandos sin puntos guía en Braille.

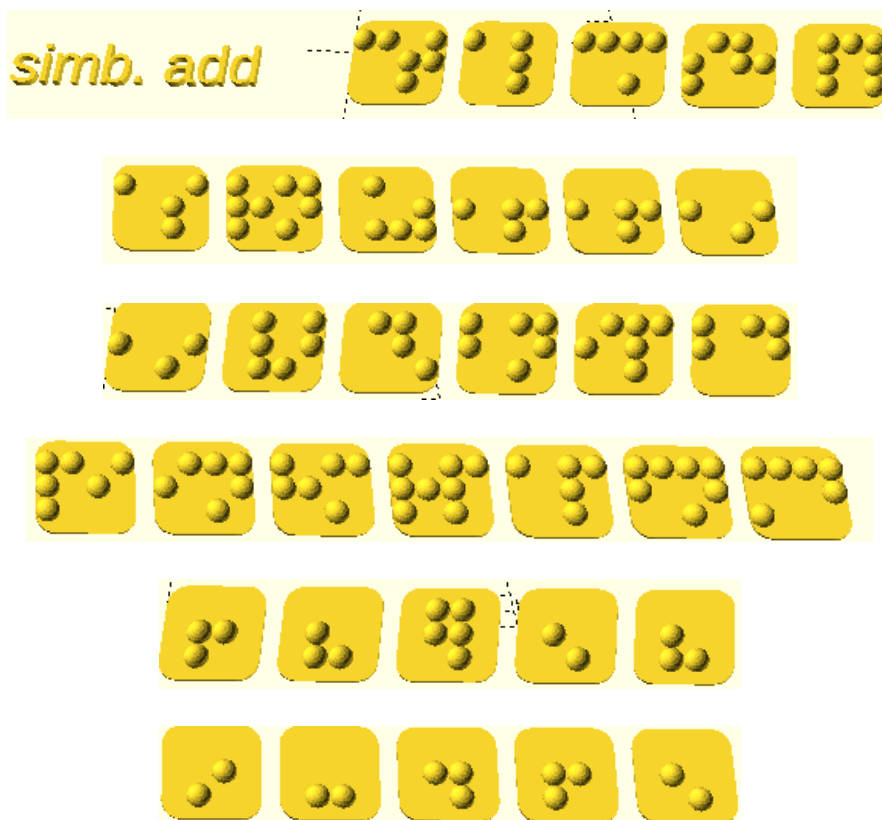


Figura 2.18 Etiquetas de símbolos adicionales sin puntos guía en Braille.

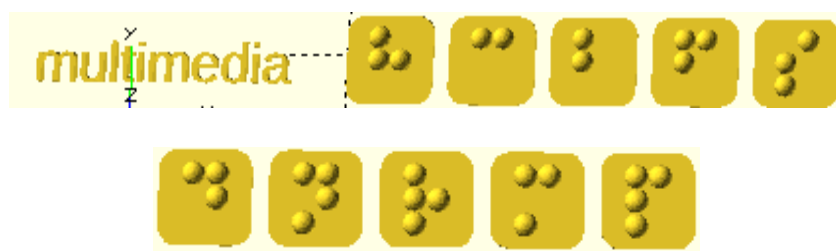


Figura 2.19 Etiquetas multimedia sin puntos guía en Braille.

En las observaciones respecto a la primera prueba, se adiciona las teclas F y J sin puntos de guía y con una línea bajo estas teclas con la finalidad de que a partir de estas etiquetas el usuario coloque correctamente sus dedos en el teclado, el diseño se indica en Figura 2.20.

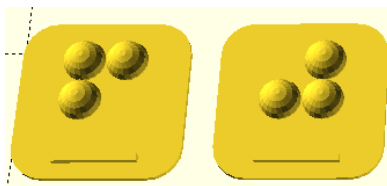


Figura 2.20 Etiquetas letras F y J sin puntos guía en Braille.

Para agregar una línea adicional bajo las líneas F y J, se agregó un módulo llamado *drawLine()* que dibuja un cubo y lo traslada según la ubicación de la etiqueta, también se creó el módulo *drawTileLine()* en base al código utilizado como base, donde se dibuja el símbolo en Braille de una letra, en este caso F y J, luego se utiliza la función *backplate()* para generar la base de la etiqueta y luego se llama a *drawLine()* mencionado anteriormente para dibujar toda la tecla. Finalmente, el arreglo formado por las letras F y J se pasa la función *drawTileLine()* para que se muestren las teclas como se indica en Código 2.13.

```
// Dibujo de cubo
module drawLine() {
  translate([keySize / 4, keySize * 0.1, baseHeight]) {
    cube([keySize*0.5, baseHeight*2, baseHeight*2]);
  }
}

// Dibujo de tecla con línea debajo
module drawTileLine(symbol, xOffset, yOffset) {
  translate([(xOffset * (keySize + tileSpacing)), -yOffset * (keySize + tileSpacing), 0]) {
    backplate();
    intersection() {
      Char_display_single(symbol);
      translate([0, 0, -baseHeight / 100]) scale([1, 1, 10 * scaleBraille]) backplate();
    }
  }
  drawLine();
}
```

```

}
//Arreglo con símbolos F y J
linea = [f, j];

//Modificación línea
if(setSelection == "línea" || setSelection == "all"){
    renderLabel(yPositionModifier(13), "línea");
    for(i = [0:len(linea) - 1]){
        drawTileLine(linea[i], i, yPositionModifier(13));
    }
}
}

```

Código 2.13 Dibujo de línea bajo letra en Braille.

2.3 IMPLEMENTACIÓN

Al momento de terminar el diseño de las etiquetas con OpenSCAD, se procede a guardar y renderizar lo que permite al modelo darle mayor precisión que la visualización en el entorno de trabajo al guardar el modelo y finalmente exportar el archivo como STL para realizar la impresión en 3D. El material a utilizar es PLA, que es bastante conocido y común para impresión en 3D debido a que permite lijar, pintar y realizar post procesamiento de forma más sencilla. PLA (Ácido Poliláctico) proviene de fuentes como maicena o caña de azúcar [29].

Se realizó la impresión de todas las etiquetas con la consideración de los 2 casos para realizar pruebas como se muestra a continuación. Para el primer caso las etiquetas diseñadas en Braille como se puede observar en la Figura 2.21 a excepción de las etiquetas multimedia que indican el símbolo como se indica en Figura 2.22.



Figura 2.21 Implementación de etiquetas Braille diseñadas.



Figura 2.22 Etiquetas teclado multimedia con símbolo.

Para el segundo caso las teclas multimedia están representadas por la letra en Braille que representa su funcionalidad como se indica en Figura 2.23. De acuerdo a esto se realizarán las pruebas correspondientes.



Figura 2.23 Etiquetas teclado multimedia en Braille.

De las modificaciones de diseño respecto a la eliminación de puntos guía, también se realizó la impresión y colocación las etiquetas de la misma forma que las etiquetas con puntos guía donde una vez que se guarda el archivo con formato *.scad*, se realiza la renderización del modelo y luego conversión a *.stl* para la impresión en 3D. El teclado sin puntos de apoyo se muestra en la Figura 2.24.



Figura 2.24 Etiquetas de teclado en Braille sin puntos guía.

De la misma forma, las etiquetas en Braille sin puntos guía para teclas multimedia se agregaron al teclado como se puede observar en Figura 2.25.



Figura 2.25 Etiquetas de teclado multimedia en Braille sin puntos de apoyo.

También se puede observar el detalle de las teclas con línea abajo como es el caso de las líneas F y J para brindar al usuario una mejor ubicación respecto a las teclas del teclado. Como se puede observar en Figura 2.26.

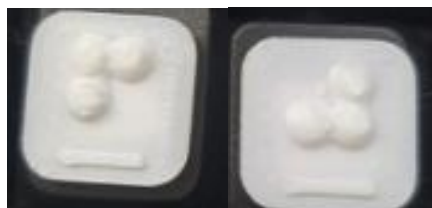


Figura 2.26 Etiquetas de F y J con línea abajo del símbolo.

2.4 PRUEBAS

Esta sección se realizó con el apoyo de tres voluntarios. Antes de las pruebas de validación, se realizó el reconocimiento de cada una de las etiquetas colocadas en el teclado con los voluntarios, esto permitió que se puedan familiarizar con cada una de las teclas y cómo se encuentran colocadas en el teclado Meefion. Adicionalmente, se utilizó el Narrador de

Windows cuya función principal es dictar mediante voz las ventanas que han sido abiertas y las teclas presionadas.

Las pruebas para validación de requerimientos se llevaron a cabo mediante la ejecución de dos tareas: la primera sobre el acceso y uso de un programa del equipo, en este caso Word y la segunda sobre uso y acceso del reproductor de música.

Para el acceso y uso de Word, cada voluntario con el uso del teclado, realiza una búsqueda al programa Word y posteriormente a ejecutar el programa, escribe una frase corta. Esta prueba tiene la finalidad de comprobar el uso de las etiquetas en Braille.

Para uso y acceso del reproductor de música, cada voluntario desde el Escritorio de Windows, utiliza las etiquetas para teclas multimedia para acceder al reproductor de música como es el símbolo de música y presiona otras etiquetas como como subir volumen, bajar volumen, mute, play y pause, los símbolos correspondientes se muestran en Tabla 2.1. De la misma forma con las etiquetas en Braille de las etiquetas multimedia como se muestra en Tabla 2.2.

Como se puede observar en Figura 2.26, Figura 2.27 y Figura 2.28, las pruebas mencionadas anteriormente realizadas por los voluntarios.



Figura 2.27 Pruebas con voluntarios.



Figura 2.28 Pruebas con voluntarios.



Figura 2.29 Pruebas con voluntarios.

Al terminar las tareas, los voluntarios respondieron a una encuesta sobre la experiencia con el teclado. Se realizó una reunión para reconocer el teclado e indicar observaciones acerca del mismo.

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En esta sección se muestra los Resultados, Conclusiones y Recomendaciones. En la sección 3.1 se indica sobre la validación de los requerimientos de acuerdo a las pruebas realizadas en el anterior capítulo. En la sección 3.2 se trata sobre las conclusiones respecto al prototipo de etiquetas para teclado implementado y finalmente, en la sección 3.3 se indica sobre las recomendaciones del prototipo y programa de modelado 3D.

3.1 RESULTADOS

3.1.1 VALIDACIÓN DE REQUERIMIENTOS

Para la validación de los requerimientos de las etiquetas del teclado, se basó en las preguntas de las encuestas respondidas por los voluntarios como se puede observar en Tabla 3.1. Las encuestas realizadas se encuentran en Anexo III.

Tabla 3.1 Requerimientos y observaciones.

Requerimiento	Observaciones
Las etiquetas deben incluir alfabeto Braille para que se puedan identificar las teclas.	Cumplido. Según la pregunta 12 de la encuesta las etiquetas se consideraron entre el rango de Buenas a muy Buenas.
Las etiquetas deben incorporar diseño de teclas multimedia para control de sonido como volumen, pausa, acceso a aplicaciones, etc. según el diseño del teclado.	Cumplido. Según la pregunta 13 las etiquetas multimedia se consideraron en el rango de Buenas a muy Buenas tanto en símbolo como en Braille.
Las etiquetas deben brindar facilidad de uso en el teclado.	Cumplido. La mayoría de etiquetas fueron reconocidas por los voluntarios, pero no en su totalidad como se

	esperaba, por lo que inicialmente se consideró como parcialmente cumplido. Con la retroalimentación realizada por los mismos, se modificó el diseño con las etiquetas sin puntos guías. Con esto, el requerimiento se cumple.
Las etiquetas deben permitir a los usuarios que se adapten rápidamente a su uso.	Cumplido. Según la pregunta 4 la dificultad para aprender a usar el teclado fue Baja.

De la información obtenida en las encuestas realizadas se tiene que según la pregunta 14, que trata sobre una comparación entre las etiquetas multimedia tanto en símbolo como en las etiquetas en Braille, existe preferencia del 66.6% de las teclas multimedia con alfabeto Braille debido a la facilidad de aprendizaje respecto a la funcionalidad con la letra a comparación de los símbolos que requieren explicación adicional para ser utilizadas. Para la consideración de las teclas multimedia con símbolo se recomienda la existencia de un estándar para los símbolos.

Respecto al material utilizado para las teclas según la pregunta 11 de la encuesta se encuentra entre el rango de Neutro a Muy Bueno, esto indica que PLA es un material adecuado para la impresión de las etiquetas en Braille. La textura del material permitió el reconocimiento de los puntos Braille en las etiquetas diseñadas.

Respecto a la comodidad con el uso prolongado de las etiquetas en el teclado, pregunta 5, se encuentra entre Neutro y Muy cómodo, esto muestra que, durante el reconocimiento de las etiquetas y pruebas correspondientes, no hubo molestia al presionar las etiquetas en el teclado ya que la finalidad de estas teclas está orientado al uso diario.

Respecto a las preguntas 7,8,9 que tratan sobre el uso de aplicaciones como el Narrador de Windows, Navegador y Word, no hubo problemas, esto indica que las etiquetas cumplen con la funcionalidad de accesibilidad por medio del uso de las etiquetas en Braille en el computador.

En la sección de sugerencias se obtuvo que de preferencia eliminar los puntos guía del diseño para reconocer todas las etiquetas en el teclado, utilizar líneas bajo las letras F y J para mejorar la ubicación de manos con respecto al teclado. Todas estas modificaciones tanto de los puntos guías y como las líneas bajo el símbolo.

3.2 CONCLUSIONES

Para el modelado en 3D existen varias herramientas a utilizar según el tipo de diseño a realizar. En este caso, el diseño de etiquetas para el teclado, la herramienta OpenSCAD indicó varias ventajas como: generar las etiquetas en 3D mediante un script con los parámetros y métodos requeridos, realizar modificaciones respecto al tamaño de teclas y visualizar el diseño.

Respecto al diseño de etiquetas, se realizaron dos diseños. El primero: etiquetas en Braille con los puntos guía y el segundo: etiquetas en Braille sin los puntos guía. En la sección de pruebas se pudo observar que, entre los dos diseños, el diseño con los puntos guía si permitió la identificación de las etiquetas en Braille, pero no de forma completa. Por lo que se muestra el diseño con modificación de las etiquetas sin puntos guía para mejorar el prototipo en este aspecto.

En el diseño con teclas multimedia también se realizaron dos diseños: el primero con alfabeto Braille donde una letra representa la funcionalidad de la tecla y el segundo donde se tomó el ícono que representa la funcionalidad. En las pruebas y encuestas realizadas se obtuvo que uno de los tres voluntarios mostró mayor preferencia respecto a las teclas con el ícono y dos mostraron mayor preferencia con respecto al alfabeto Braille.

Con las etiquetas impresas en 3D se logró adaptar un teclado común, conocido como el prototipo implementado. Esto indica que se pudo brindar accesibilidad a personas con discapacidad visual para usar este periférico, en cuanto a costo fue económico porque el material de las etiquetas es PLA. Lo cual indica que es una buena opción ya que no es necesario comprar un teclado sino más bien utilizar un teclado ya existente en el hogar.

Se pudo realizar las validaciones de acuerdo a los requerimientos especificados para el prototipo de teclas, donde el requerimiento respecto a la facilidad de uso fue parcialmente cumplido inicialmente. Por lo que, se realizaron modificaciones al diseño con la eliminación de los puntos guía de acuerdo a la retroalimentación de los voluntarios guía para cumplir completamente con la validación.

Respecto a las encuestas realizadas por los voluntarios, el tamaño de las etiquetas debe ser similar al tamaño de la tecla para facilitar el uso de todas las etiquetas del teclado, la comodidad del material se consideró como buena, el tamaño de los puntos en Braille también fue el indicado para reconocer las letras.

3.3 RECOMENDACIONES

Es importante verificar si los diseños tienen licencia para su respectiva modificación, con esto se puede adaptar el diseño según los requerimientos de los usuarios, en este caso los voluntarios.

Al utilizar OpenSCAD se debe tener en cuenta la documentación existente para realizar el diseño 3D como funciones que permiten trasladar el diseño, realizar extrusión lineal, generar figuras como cubos, cilindros, generar texto, unir figuras, etc.

Para la realización de pruebas con programas de lectura de pantalla además del Narrador de Windows se pueden utilizar otras herramientas como JAWS o NVDA que son programas similares.

Se recomienda el programa OpenSCAD para diseñar múltiples veces un mismo modelo debido a la programación mediante scripts que permite modificar de forma más sencilla parámetros.

Para adaptar las teclas multimedia en un teclado se debe tener en cuenta el teclado a utilizar para incrementar la personalización del teclado.

4. REFERENCIAS BIBLIOGRÁFICAS











- [1] “Estadísticas de Discapacidad – Consejo Nacional para la Igualdad de Discapacidades”. Consultado: el 13 de febrero de 2024. [En línea]. Disponible en: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>
- [2] “202307_Tecnologia_de_la_Informacion_y_Co-unicacion-TICs.pdf”. Consultado: el 13 de febrero de 2024. [En línea]. Disponible en: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2023/202307_Tecnologia_de_la_Informacion_y_Co-unicacion-TICs.pdf
- [3] educ ar devteam, “Sistema braille”. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.educ.ar/recursos/158983/sistema-braille>
- [4] “Educación Inclusiva”. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: http://www.riate.org/version/v1/materiales_en_prueba/e_inclusiva_discapacidad/unidad_5/m5_estructura_sistema.htm
- [5] “Periférico: Qué es, tipos y ejemplos”, Ceupe. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.ceupe.com/blog/periferico.html>
- [6] “Estadísticas de Discapacidad – Consejo Nacional para la Igualdad de Discapacidades”. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>
- [7] “Productos de Apoyo Déficit Visual. Unidad de Accesibilidad Digital”. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://web.ua.es/es/accesibilidad/educacion-inclusiva/productos-de-apoyo-deficit-visual.html#>
- [8] “Nuklz: Nuklz N - Teclado de impresión grande”, Amazon.com. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.amazon.com/stores/Nuklz/NuklzN-Tecladodeimpresi%C3%B3ngrande/page/DD0D72C9-7D16-447C-9AD9-C87ADC8BAF4A>
- [9] V. H. Sîrbu, I. Şerban, y I. C. Roşca, “Braille Keyboard for People with Low Vision”, en *2019 E-Health and Bioengineering Conference (EHB)*, nov. 2019, pp. 1–4. doi: 10.1109/EHB47216.2019.8970042.
- [10] S. Sultana, A. Rahman, F. H. Chowdhury, y H. U. Zaman, “A novel Braille pad with dual text-to-Braille and Braille-to-text capabilities with an integrated LCD display”, en *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, jul. 2017, pp. 195–200. doi: 10.1109/ICICT1.2017.8342559.
- [11] M. S. M. Tahir, N. H. H. M. Hanif, y H. Md. Yusuf, “Numerical Braille Module for Learning Simple Mathematical Operations”, en *2019 7th International Conference on Mechatronics Engineering (ICOM)*, oct. 2019, pp. 1–5. doi: 10.1109/ICOM47790.2019.8952054.
- [12] “Amazon.com: Logickeyboard con alfabeto Braille de 6 puntos | Incluye lámpara de teclado LogicLight™ | USB con cable | Hub USB integrado | Compatible con PC con Windows : Salud y Hogar”. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.amazon.com/-/es/Logickeyboard-alfabeto-LogicLightTM-integrado-Compatible/dp/B09MDJ98MX?th=1>
- [13] “CAD / Diseño asistido por ordenador”, Siemens Digital Industries Software. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.plm.automation.siemens.com/global/es/our-story/glossary/computer-aided-design-cad/12507>
- [14] “The Future of CAD Technology: Innovations and Implications | Neural Concept”. Consultado: el 22 de enero de 2024. [En línea]. Disponible en:










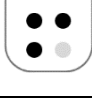
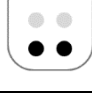

- <https://www.neuralconcept.com/post/the-future-of-cad-technology-innovations-and-implications>
- [15] S. Junk y C. Kuen, "Review of Open Source and Freeware CAD Systems for Use with 3D-Printing", *Procedia CIRP*, vol. 50, pp. 430–435, ene. 2016, doi: 10.1016/j.procir.2016.04.174.
- [16] "AutoCAD: ¿cuáles son las características del software? - 3Dnatives". Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.3dnatives.com/es/autocad-cuales-caracteristicas-del-software-020420202/#!>
- [17] "Autodesk AutoCAD 2024 | Software AutoCAD 3D y 2D". Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.autodesk.mx/products/autocad/overview?term=1-YEAR&tab=subscription&plc=ACDIST>
- [18] T. Cieres Vallori, *Manual esencial de AutoCAD: nivel inicial*. Editorial Nobuko, 2021. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://bvirtual.epn.edu.ec:2129/es/ereader/epn/217359/>
- [19] M. S. Hosen, S. Ahmmed, y S. Dekkati, "Mastering 3D Modeling in Blender: From Novice to Pro", *ABC Res. Alert*, vol. 7, núm. 3, pp. 169–180, dic. 2019, doi: 10.18034/ra.v7i3.654.
- [20] "Cómo elegir el mejor software de CAD 3D: Una guía exhaustiva", Formlabs. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://formlabs.com/latam/blog/software-cad/>
- [21] "Creating a New SOLIDWORKS Document - 2021 - SOLIDWORKS Help". Consultado: el 22 de enero de 2024. [En línea]. Disponible en: https://help.solidworks.com/2021/english/SolidWorks/sldworks/HelpViewerDS.aspx?version=2021&prod=SolidWorks&lang=english&path=sldworks%2fHIDD_NEW_DOCUMENT.htm&id=1aa4a1fe643d4bc49e8242515c3596c9
- [22] "Novedades de SOLIDWORKS 2024", SOLIDWORKS. Consultado: el 22 de enero de 2024. [En línea]. Disponible en: <https://www.solidworks.com/es/product/whats-new>
- [23] "¿Qué es SolidWorks? - ADR Formación". Consultado: el 22 de enero de 2024. [En línea]. Disponible en: https://www.adrformacion.com/knowledge/ingenieria-y-proyectos/_que_es_solidworks_.html
- [24] "FreeCAD: Tu propio modelador paramétrico 3D". Consultado: el 22 de enero de 2024. [En línea]. Disponible en: https://www.freecad.org/features.php?lang=es_ES
- [25] "OpenSCAD - About". Consultado: el 23 de julio de 2024. [En línea]. Disponible en: <https://openscad.org/about.html>
- [26] "OpenSCAD Tutorial/Chapter 1 - Wikibooks, open books for an open world". Consultado: el 23 de julio de 2024. [En línea]. Disponible en: https://en.wikibooks.org/wiki/OpenSCAD_Tutorial/Chapter_1
- [27] "El mejor teclado multimedia USB al por mayor | Reunión". Consultado: el 23 de julio de 2024. [En línea]. Disponible en: <https://www.meetion.es/mt-k600m.html>
- [28] "Release Version 3.0.0 · davidakerr/braille_keyboard_covers", GitHub. Consultado: el 19 de junio de 2024. [En línea]. Disponible en: https://github.com/davidakerr/braille_keyboard_covers/releases/tag/v3.0.0
- [29] "What is PLA? (Everything You Need To Know)". Consultado: el 23 de julio de 2024. [En línea]. Disponible en: <https://www.twi-global.com/technical-knowledge/faqs/what-is-pla.aspx>

5. ANEXOS

ANEXO I

Tabla 5.1 Alfabeto en Braille.

Etiqueta	Significado
	A
	B
	C
	D
	E
	F
	G
	H
	I
	J

	K
	L
	M
	N
	O
	P
	Q
	R
	S
	T
	U
	V





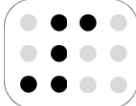
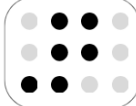
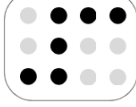

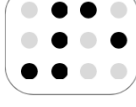

	W
	X
	Y
	Z

Tabla 5.2 Números en Braille.

Etiqueta	Significado
	1
	2
	3
	4
	5
	6

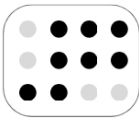
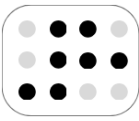
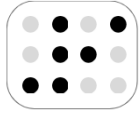







	7
	8
	9
	0

Tabla 5.3 Signos en Braille.

Etiqueta	Significado
	.
	,
	;
	:
	(
)







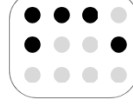

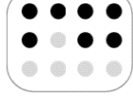

	-
	/

Tabla 5.4 Funciones en Braille

Etiqueta	Significado
	F1
	F2
	F3
	F4
	F5
	F6
	F7
	F8

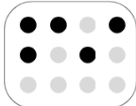








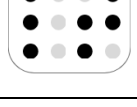

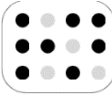
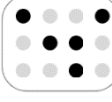



	F9
---	----

Tabla 5.5 Comandos en Braille.

Etiqueta	Significado
	Control
	Alt
	Cmd
	Barra espaciadora
	Shift
	CapsLock
	Up (Arriba)
	Down (Abajo)
	Left (Izquierda)

	Right (Derecha)
	Return (Enter)
	ESC
	TAB
	Backspace
	Delete

ANEXO II

Códigos.

Link: [Códigos etiquetas Braille](#)

ANEXO III.

Encuestas.

Link: [Encuestas](#)