



**UNIVERSIDAD SURCOLOMBIANA
GESTIÓN SERVICIOS BIBLIOTECARIOS**



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 2

Neiva, 21 de enero de 2022

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Angelo Sebastian Alejandro Mosquera Pelaez, con C.C. No. 1075306346,

Andrés Camilo Castañeda Palencia, con C.C. No. 1075302444,

_____, con C.C. No. _____,

_____, con C.C. No. _____,

Autor(es) de la tesis y/o trabajo de grado o _____

titulado DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO PORTABLE DE LECTURA BRAILLE MEDIANTE RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR)

presentado y aprobado en el año 2021 como requisito para optar al título de

Ingeniero Electrónico;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.
- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



**UNIVERSIDAD SURCOLOMBIANA
GESTIÓN SERVICIOS BIBLIOTECARIOS**



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

2 de 2

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores", los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

Angelo Sebastian Alejandro Mosquera Pelaez

Firma:

EL AUTOR/ESTUDIANTE:

Andrés Camilo Castañeda Palencia

Firma:

EL AUTOR/ESTUDIANTE:

Firma: _____

EL AUTOR/ESTUDIANTE:

Firma: _____

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



TÍTULO COMPLETO DEL TRABAJO: DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO PORTABLE DE LECTURA BRAILLE MEDIANTE RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR)

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Mosquera Pelaez	Angelo Sebastian Alejandro
Castañeda Palencia	Andrés Camilo

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Salgado Patrón	José de Jesús

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre

PARA OPTAR AL TÍTULO DE: Ingeniero Electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Ingeniería Electrónica






CIUDAD: Neiva

AÑO DE PRESENTACIÓN: 2021

NÚMERO DE PÁGINAS: 115

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas x Fotografías x Grabaciones en discos Ilustraciones en general x Grabados
Láminas Litografías Mapas Música impresa Planos x Retratos Sin ilustraciones
Tablas o Cuadros x

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN SERVICIOS BIBLIOTECARIOS					   	
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 3

SOFTWARE requerido y/o especializado para la lectura del documento:

MATERIAL ANEXO:

PREMIO O DISTINCIÓN *(En caso de ser LAUREADAS o Meritoria):*






PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>	<u>Español</u>	<u>Inglés</u>
1. Braille	Braille	6 Portable	Portable
2. Inclusión social	Social inclusion	7. Celda dinámica	Dynamic cell
3. Discapacidad visual	Visual Disability	8.	
4. Prototipo	Prototype	9.	
5. OCR	OCR	10.	

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

El objetivo de este proyecto fue diseñar e implementar un prototipo que permita traducir texto impreso a texto braille haciendo uso de una cámara y una celda dinámica braille. Para esto se aplicaron técnicas de reconocimiento óptico de caracteres a partir del análisis matricial de las imágenes obtenidas y por medio de diferentes etapas con las que se realiza el procesamiento y adecuación de los datos, así como su respectiva conversión a caracteres ASCII, para finalmente ser representados en una celda dinámica braille, basada en motores vibratorios ERM tipo disco, cuyo bajo consumo

energético y tamaño permiten un diseño compacto. El dispositivo también recibe comandos desde una aplicación web, lo que permite que sea usado como una herramienta pedagógica para enseñar braille. Además de lo anterior, se agregaron elementos como botones y elementos lumínicos de señalización que permiten una mejor experiencia con el fin de optimizar la experiencia de usuario y el uso del equipo en el ámbito pedagógico. Esta herramienta le permitirá a pedagogos y terapeutas explorar nuevas formas de enseñar braille.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN SERVICIOS BIBLIOTECARIOS					   	
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	3 de 3

ABSTRACT: (Máximo 250 palabras)

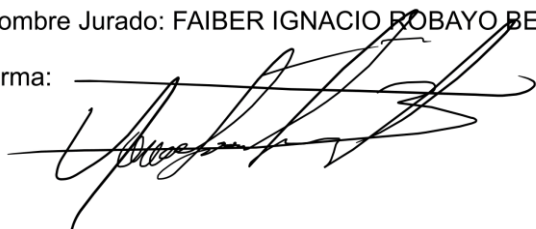
The aim of this project is to design and develop a prototype that can translate printed text into braille using a camera and a dynamic braille cell. In order to do it, we implement optical character recognition techniques using matrix analysis of obtained pictures through several stages in which the data processing is performed, as well as its corresponding translation to ASCII characters, to finally represent them with a dynamic braille cell based on ERM vibrating coin-type motors, allowing a small size and low energy consumption model.

We can also represent commands sent through a web app which allows the device to be used as a didactic braille teaching tool. In addition, the device has some key elements like buttons and LED indicators to improve the user experience and the general use of the device in an academic environment. Each thing mentioned above, makes the device a remarkable tool for pedagogues and therapists to explore new ways to teach braille.

APROBACIÓN DE LA TESIS

Nombre Jurado: FAIBER IGNACIO ROBAYO BETANCOURT

Firma:



Nombre Jurado: FERNAND DIAZ FRANCO

Firma:



Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO PORTABLE DE
LECTURA BRAILLE MEDIANTE RECONOCIMIENTO ÓPTICO DE
CARACTERES (OCR)**

**ANGELO SEBASTIAN ALEJANDRO MOSQUERA PELAEZ
ANDRES CAMILO CASTAÑEDA PALENCIA**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA - HUILA
2021**

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO PORTABLE DE
LECTURA BRAILLE MEDIANTE RECONOCIMIENTO ÓPTICO DE
CARACTERES (OCR)**

**ANGELO SEBASTIAN ALEJANDRO MOSQUERA PELAEZ
ANDRES CAMILO CASTAÑEDA PALENCIA**

**Tesis de grado presentada para aplicar
al título de Ingeniero Electrónico**

**Director:
JOSÉ DE JESÚS SALGADO PATRÓN
Máster en Ingeniería Electrónica y de Computadores**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA - HUILA
2021**

Nota de aceptación

Firma del director de Tesis

Firma del Jurado

Firma del Jurado

Neiva, 04 de mayo de 2021

Mis más profundos agradecimientos a mi padre Jairo Mosquera Moreno y a mi tía María Nury Mosquera Moreno, quienes guiaron mi camino y me dieron una valiosa educación, fundando en mí valores que mantendré por el resto de la vida. Me gustaría agradecer también a mi tía Alby Mosquera Moreno, a mi tío Fernando Vesga Gonzales y a mi prima Luisa Fernanda Vesga Mosquera, cuyo apoyo y consejos me ayudaron a seguir adelante en los momentos más difíciles, quiero agradecer a toda mi familia por confiar en mí durante todo este tiempo, los llevo en lo más profundo de mi corazón, a mis amigos, la familia que escogí durante este trayecto, quienes me ayudaron a crecer con sus puntos de vista y formas de actuar y finalmente a todo el personal de la universidad, a los profesores por su paciencia y extraordinaria capacidad para transmitir conocimientos, a los monitores, a las personas encargadas de los laboratorios, a las personas encargadas de los servicios generales y a los administrativos, su gran labor ayuda a formar a jóvenes como yo para aportar a la región y considero que no siempre es valorada como se debería.

Angelo Sebastián Alejandro Mosquera Peláez

A mi familia, por su apoyo incondicional durante cada objetivo que me he planteado, a mis padres por los sacrificios que hicieron posible mi escolaridad, por brindarme las oportunidades y hacerme una persona capaz de lidiar con los obstáculos. A mi hermano, porque aún sin entender a plenitud los objetivos a los que dedico mis esfuerzos, siempre ha estado para mí y se ha convertido en un pilar fundamental de mi motivación diaria. A mis más cercanos amigos, en especial a mi compañero de proyecto, *Angelo Mosquera*, cuya amistad y comprensión ha impulsado mi actuar desde hace varios años. A mis maestros y demás profesionales que la vida universitaria me permitió conocer, modelos a seguir desde entonces y a todas las personas cuyas enseñanzas contribuyeron de una u otra manera a mi desarrollo académico y personal.

Andrés Camilo Castañeda Palencia

AGRADECIMIENTOS

Agradecemos a nuestro director de tesis José de Jesús Salgado, por guiarnos durante todo el proceso, al diseñador y gestor del tecnoparque Jorge Pérez, cuyo acompañamiento y experiencia nos permitió consolidar el diseño del equipo orientándolo al usuario final, así como a los demás profesionales del nodo Neiva de la misma institución, por sus críticas constructivas y la disposición brindada. A don Luis Eduardo Aroca Sánchez, fundador y director de la IPS Asolivihuilá, por abrirnos las puertas de su institución y aportar con su valioso conocimiento a este proyecto. Al pedagogo Erinson Arciniegas Hernández, cuyo profundo conocimiento del sistema braille nos permitió definir el desarrollo del dispositivo. Finalmente agradecemos a todas las personas que participaron en las diferentes pruebas y entrevistas que se hicieron durante el desarrollo de este prototipo, así como a las personas que compartieron sus opiniones y consejos, aunque no participaron de forma directa en el proyecto.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	Pág. 15
1 OBJETIVO.....	Pág. 16
1.1 OBJETIVO GENERAL.....	Pág. 16
1.2 OBJETIVOS ESPECÍFICOS	Pág. 16
2 SISTEMA BRAILE	Pág. 17
3 PRINCIPIO DE FUNCIONAMIENTO DE LOS ACTUADORES DE LA CELDA BRAILLE DINÁMICA	Pág. 18
4 RECONOCIMIENTO ÓPTICO DE CARACTERES	Pág. 19
5 DISPOSITIVO.....	Pág. 20
6 SOFTWARE	Pág. 28
6.1 SOFTWARE CENTRAL DEL DISPOSITIVO.....	Pág. 29
6.1.1 Saludo	Pág. 29
6.1.2 Modo analógico	Pág. 30
6.1.3 Modo digital	Pág. 30
6.1.4 Modo sleep	Pág. 30
6.1.5 Librería brailfun.....	Pág. 30
6.2 APLICACIÓN WEB	Pág. 32
7 SISTEMA DE RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR). Pág. 34	
7.1 DIAGRAMA DE BLOQUES DEL SISTEMA DE RECONOCIMIENTO ÓPTICO DE CARACTERES.....	Pág. 34
7.2 ADQUISICIÓN/CAPTURA DE LA IMAGEN	Pág. 35
7.3 PREPROCESAMIENTO.....	Pág. 35
7.4 DETECCIÓN DE BORDES Y CONTORNOS.....	Pág. 35
7.5 CARACTERIZACIÓN DEL TEXTO	Pág. 35
7.6 ENTRENAMIENTO Y EXTRACCIÓN DEL MODELO	Pág. 36
7.7 RECONOCIMIENTO DEL TEXTO	Pág. 36
7.8 EXTRACCIÓN DE CARACTERES ASCII	Pág. 36
8 RESULTADOS	Pág. 37
9 ANÁLISIS DE RESULTADOS	Pág. 48
9.1 ANÁLISIS DEL DISPOSITIVO.....	Pág. 48
9.2 ANÁLISIS DE LA PRUEBA DE LA CELDA BRAILLE DINÁMICA.....	Pág. 48
9.3 ANÁLISIS DE LA PRUEBA REALIZADA CON LA CÁMARA.....	Pág. 49
9.4 ANÁLISIS DE LA PRUEBA DE RENDIMIENTO DE LOS ALGORITMOS OCR	Pág. 49

9.5	ANÁLISIS DE LA PRUEBA INTEGRAL DEL DISPOSITIVO	Pág. 50
10	CONCLUSIONES	Pág. 51
11	RECOMENDACIONES.....	Pág. 52
	BIBLIOGRAFÍA.....	Pág. 53
	ANEXOS	Pág. 56

LISTA DE TABLAS

Cuadro 1. Cantidad promedio de intentos por estímulo.....	Pág. 39
Cuadro 2. Tiempo de lectura promedio en segundos por estímulo.....	Pág. 40
Cuadro 3. Porcentaje de acierto promedio por estímulo	Pág. 40
Cuadro 4. Calidad de la imagen según la distancia entre el lente y el papel	Pág. 41
Cuadro 5. Tamaño de letra necesario para capturar totalmente un máximo de 3 letras por imagen	Pág. 42
Cuadro 6. Comparativa del tiempo de ejecución de los algoritmos de reconocimiento de caracteres.....	Pág. 42
Cuadro 7. Porcentaje de acierto del algoritmo de reconocimiento óptico	Pág. 43
Cuadro 8. Tiempo de respuesta en segundos en el modo digital	Pág. 45
Cuadro 9. Tiempo de respuesta en segundos en el modo analógico	Pág. 46
Cuadro 10. Tiempo de encendido	Pág. 46
Cuadro 11. Tiempo de apagado.....	Pág. 47

LISTA DE FIGURAS

Figura 1. Representación del sistema braille convencional según el estándar del Reino Unido	Pág. 17
Figura 2. Vista de despiece de un motor vibratorio ERM tipo disco	Pág. 18
Figura 3. Dirección de vibración de los motores ERM	Pág. 18
Figura 4. Fotografía del dispositivo	Pág. 20
Figura 5. Diagrama de bloques del prototipo.	Pág. 20
Figura 6. Imagen de la Raspberry Pi Zero W	Pág. 21
Figura 7. Imagen de un vibrador ERM tipo disco	Pág. 22
Figura 8. Diagrama esquemático del circuito de potencia la celda braille dinámica	Pág. 22
Figura 9. Vista isométrica de la celda braille dinámica.....	Pág. 23
Figura 10. Vistas acotadas de la celda braille dinámica.....	Pág. 23
Figura 11. Fotografía de la celda braille dinámica.....	Pág. 23
Figura 12. Vista de despiece de la celda braille dinámica.....	Pág. 23
Figura 13. Vista lateral acotada de un punto de la celda braille dinámica	Pág. 23
Figura 14. Raspberry Pi Camera Board v2. Vista frontal y posterior.....	Pág. 24
Figura 15. Circuito esquemático de los LED de iluminación de la cámara	Pág. 25
Figura 16. Circuito esquemático del LED de indicación	Pág. 25
Figura 17. Vista isométrica del módulo Powerboost 1000C.....	Pág. 26
Figura 18. Circuito esquemático del conversor análogo-digital MCP3008	Pág. 27
Figura 19. Imagen de la batería de polímero de litio	Pág. 27
Figura 20. Diagrama del software del dispositivo.....	Pág. 28
Figura 21. Diagrama de flujo del software central del dispositivo	Pág. 29
Figura 22. Representación gráfica de la letra 'r' en braille	Pág. 29
Figura 23. Menú principal de la aplicación web.....	Pág. 32
Figura 24. Menú de configuración de perfil de la aplicación web.....	Pág. 32
Figura 25. Captura del conjunto de datos llamado "funciones" con sus respectivos datos.....	Pág. 33
Figura 26. Captura del conjunto de datos llamado "perfiles" con sus respectivos datos	Pág. 33
Figura 27. Diagrama de bloques del algoritmo de entrenamiento del sistema de reconocimiento óptico de caracteres.....	Pág. 34
Figura 28. Diagrama de bloques del algoritmo de reconocimiento del sistema OCR	Pág. 34
Figura 29. Fotografía isométrica del dispositivo	Pág. 37
Figura 30. Fotografía isométrica del dispositivo destapado	Pág. 38
Figura 31. Fotografía lateral del dispositivo destapado.....	Pág. 38
Figura 32. Fotografía de la PCB del dispositivo	Pág. 38
Figura 33. Diagrama de barras de la cantidad promedio de intentos por estímulo	Pág. 39
Figura 34. Diagrama de barras del tiempo de lectura promedio por estímulo	Pág. 40
Figura 35. Diagrama de barras del porcentaje de acierto promedio por estímulo.....	Pág. 41
Figura 36. Diagrama de bloques de las etapas del modo digital.....	Pág. 44
Figura 37. Diagrama de líneas del tiempo promedio de las etapas del modo digital	Pág. 45
Figura 38. Diagrama de bloques de las etapas del modo analógico.....	Pág. 45
Figura 39. Diagrama de líneas del tiempo promedio de las etapas del modo analógico	Pág. 46
Figura 40. Cantidad de caracteres representados en una carga completa de la batería	Pág. 47

Figura 41. Tiempo de descarga con un uso intensivo del dispositivo	Pág. 47
Figura 42. Circuito esquemático del dispositivo	Pág. 56
Figura 43. Visualización 3D de la PCB del dispositivo	Pág. 57
Figura 44. Vista inferior del diseño de la PCB del dispositivo	Pág. 57
Figura 45. Vista superior del diseño de la PCB del dispositivo	Pág. 57
Figura 46. Vista inferior del diseño de la PCB de la celda braille dinámica	Pág. 57

LISTA DE ANEXOS

Anexo A. Circuito esquemático y diseño de la PCB del dispositivo.	Pág. 56
Anexo B. Código del software del dispositivo.....	Pág. 58
Anexo C. Código de la librería Brailfun.	Pág. 66
Anexo D. Código de la aplicación web.	Pág. 72
Anexo E. Código de entrenamiento del algoritmo de OCR	Pág. 80
Anexo F. Código de ejecución del algoritmo de OCR	Pág. 82
Anexo G. Tablas de medición de la autonomía del dispositivo.	Pág. 84

GLOSARIO

Actuador: Dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o “actuar” otro dispositivo mecánico.

Algoritmo: Concatenación de pasos lógicos que permiten llegar a la solución de un determinado problema.

Braille: Es un sistema de lectura y escritura táctil usado por personas con limitaciones visuales.

GPIO: (Entrada/Salida de Propósito General) es un pin genérico en un chip, cuyo comportamiento se puede programar por el usuario en tiempo de ejecución.

USB: (Bus Universal en Serie) es un bus de comunicaciones que sigue un estándar que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.

PCB: (Printed Circuit Board) es un soporte físico en donde se instalan componentes electrónicos y eléctricos y se interconectan entre ellos.¹

OCR: El reconocimiento óptico de caracteres (OCR por sus siglas en inglés) es el proceso electrónico o mecánico de conversión de imágenes de texto impreso o escrito a texto digital.

Tesseract: Es un motor de reconocimiento óptico de caracteres de código libre que reconoce más de 100 idiomas.

Wrapper: En ciencias de la computación, un wrapper es cualquier entidad que encapsula (envuelve) otro ítem. Son usados principalmente con 2 propósitos: Convertir datos a un formato compatible o disminuir la complejidad de la entidad subyacente usando abstracción.

Umbralización: En procesamiento digital de imágenes, la umbralización (o thresholding²) es una técnica de segmentación con la que se crea una imagen binaria a partir de la imagen inicial. Con esta técnica, cada píxel de la imagen original es reemplazado por un píxel de color negro o blanco de acuerdo con el resultado de la comparación entre la intensidad del píxel en cuestión y una constante T.

¹ “▷ Qué es una PCB o Placa de Circuito Impreso. Uso, cómo se fabrica”, *Profesional Review* (blog), el 11 de febrero de 2019, <https://www.profesionalreview.com/2019/02/11/pcb-que-es/>.

² “Thresholding”, en *Wikipedia*, consultado el 3 de abril de 2021, [https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))

RESUMEN

El objetivo de este proyecto fue diseñar e implementar un prototipo que permita traducir texto impreso a texto braille haciendo uso de una cámara y una celda dinámica braille. Para esto se aplicaron técnicas de reconocimiento óptico de caracteres a partir del análisis matricial de las imágenes obtenidas y por medio de diferentes etapas con las que se realiza el procesamiento y adecuación de los datos, así como su respectiva conversión a caracteres ASCII, para finalmente ser representados en una celda dinámica braille, basada en motores vibratorios ERM tipo disco, cuyo bajo consumo energético y tamaño permiten un diseño compacto. El dispositivo también recibe comandos desde una aplicación web, lo que permite que sea usado como una herramienta pedagógica para enseñar braille.

Además de lo anterior, se agregaron elementos como botones y elementos lumínicos de señalización que permiten una mejor experiencia con el fin de optimizar la experiencia de usuario y el uso del equipo en el ámbito pedagógico. Esta herramienta le permitirá a pedagogos y terapeutas explorar nuevas formas de enseñar braille.

ABSTRACT

The aim of this project is to design and develop a prototype that can translate printed text into braille using a camera and a dynamic braille cell. In order to do it, we implement optical character recognition techniques using matrix analysis of obtained pictures through several stages in which the data processing is performed, as well as its corresponding translation to ASCII characters, to finally represent them with a dynamic braille cell based on ERM vibrating coin-type motors, allowing a small size and low energy consumption model.

We can also represent commands sent through a web app which allows the device to be used as a didactic braille teaching tool. In addition, the device has some key elements like buttons and LED indicators to improve the user experience and the general use of the device in an academic environment. Each thing mentioned above, makes the device a remarkable tool for pedagogues and therapists to explore new ways to teach braille.

INTRODUCCIÓN

Desde sus inicios el lenguaje escrito ha sido una pieza clave para intercambiar información. La facilidad a la hora de reproducir y asimilar textos ha logrado que la escritura se imponga como un medio crucial para la transmisión de conocimiento, lo que añade un gran sesgo en la sociedad, si analizamos la situación de la comunidad con impedimentos visuales, que contrario al resto de las personas, lidian con numerosas dificultades para acceder a la producción literaria ³. Esta problemática está aún más acentuada en el material impreso cuya lectura resulta casi imposible a menos que se tenga un acompañante que pueda transmitir oralmente el contenido.

Este proyecto sienta las bases de un dispositivo que permitirá a las personas invidentes acceder a miles de documentos como ensayos, revistas, periódicos, material científico y novelas que actualmente son de difícil acceso sin el acompañamiento de terceros. Es por esto por lo que se propone diseñar un dispositivo portable y de fácil uso que capture el texto mediante una cámara, lo convierta en caracteres ASCII mediante OCR (reconocimiento óptico de caracteres) y posteriormente envíe la información a un actuador braille, ayudando a cerrar la brecha académica, laboral y cultural que existe entre las personas con impedimentos visuales y aquellas que no cuentan con esta problemática.

³ World Health Organization, International Agency for the Prevention of Blindness, *VISION 2020 THE RIGHT TO SIGHT - Global Initiative for the Elimination of Avoidable Blindness: : Action Plan 2006-2011*. (France: World Health Organization, 2007), https://www.who.int/blindness/Vision2020_report.pdf.

1 OBJETIVO

1.1 OBJETIVO GENERAL

Diseñar e implementar un dispositivo portable de lectura braille mediante reconocimiento óptico de caracteres (OCR).

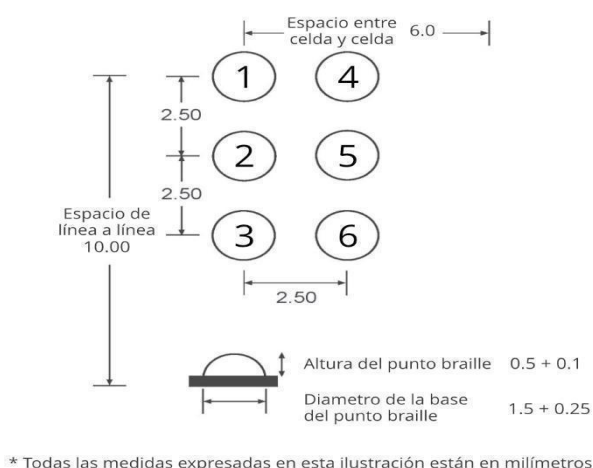
1.2 OBJETIVOS ESPECÍFICOS

- Adquirir y procesar imágenes mediante una técnica de OCR.
- Evaluar la capacidad de las cámaras portables a la hora de aplicar técnicas OCR en una computadora Raspberry Pi Zero.
- Convertir caracteres ASCII en señales adecuadas para un actuador braille.
- Validar el funcionamiento del dispositivo mediante pruebas con usuarios invidentes.

2 SISTEMA BRAILE

El sistema braille consiste esencialmente en la impresión manual, mecánica o informatizada de combinaciones de puntos en relieve sobre una sucesión de celdas braille, cada una con seis puntos dispuestos en dos columnas de tres puntos cada una. Con las 63 combinaciones resultantes se pueden representar las distintas letras, números, signos de puntuación y cualquier sistema de signos específicos (matemáticas, ajedrez, etc.) que se deseen. Para poder identificar mejor los puntos que constituyen un carácter se crea la enumeración mostrada en la Figura 1: ⁴

Figura 1. Representación del sistema braille convencional según el estándar del Reino Unido



Fuente: UK Association for Accessible Formats (UKAAF), "Standard Dimensions for the UK Braille Cell" (UK Association for Accessible Formats (UKAAF), abril de 2017), <https://www.ukaaf.org/wp-content/uploads/BrailleStandardDimensionsFinal.pdf>.

En una celda dinámica los puntos en relieve pueden ser representados a través de diferentes actuadores como motores lineales, servomotores⁵ o impulsos eléctricos⁶. En este caso se optó por usar motores vibratorios ERM tipo disco,⁷ debido a que su naturaleza de funcionamiento, en comparación con las otras opciones electromecánicas, permite un diseño sencillo que se ajusta a los objetivos del proyecto, gracias a que son de fácil acceso en el mercado, pequeños y de bajo costo. La representación a través de pulsos eléctricos fue descartada debido a que su funcionamiento demanda componentes de mayor tamaño que afectan directamente la portabilidad del equipo.

⁴ Comisión Braille Española. Organización Nacional de Ciegos Españoles (ONCE), *GUÍAS DE LA COMISIÓN BRAILLE ESPAÑOLA, Signografía básica*, Primera Edición (Madrid: Artes Gráficas Palermo, S.L., 2005), <https://sid.usal.es/idsocs/F8/FDO12069/signografiabasica.pdf>.

⁵ "Tactile (Device)", en *Wikipedia*, el 12 de diciembre de 2019, [https://en.wikipedia.org/w/index.php?title=Tactile_\(device\)&oldid=930397893](https://en.wikipedia.org/w/index.php?title=Tactile_(device)&oldid=930397893).

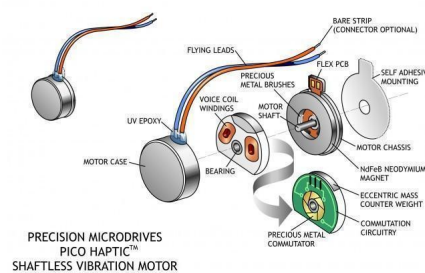
⁶ A Puertas et al., "Braille Line Using Electrical Stimulation", *Journal of Physics: Conference Series* 90 (el 1 de noviembre de 2007): 012091, <https://doi.org/10.1088/1742-6596/90/1/012091>.

⁷ Tatoko, "Tatoko 20PCS 10mmx3mm Mini Vibration Motors DC 3V Flat Coin Button-Type", Amazon, el 24 de marzo de 2019, <https://www.amazon.com/tatoko-Vibration-Button-Type-Vibrating-Appliances/dp/B07Q1ZV4MJ>.

3 PRINCIPIO DE FUNCIONAMIENTO DE LOS ACTUADORES DE LA CELDA BRAILLE DINÁMICA

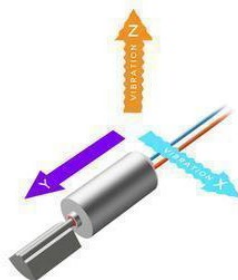
Los motores de masa rotativa excéntrica (ERM por sus siglas en inglés), cuentan con 3 polos de conmutación, su principio de funcionamiento consiste en un imán que está adherido al chasis del motor y un rotor compuesto por una masa descentrada con dos bobinas y dos escobillas que están en contacto con un conmutador, estas bobinas crean un campo magnético inducido por el conmutador que, al interactuar con el imán, producen el movimiento del rotor. Como consecuencia de este movimiento y debido a que el rotor cuenta con una masa excéntrica, se genera una vibración ⁸ en dos ejes, el eje X y el eje Z ⁹. En la Figura 2 podemos ver cada uno de sus componentes internos y en la Figura 3, la dirección de vibración para este tipo de motores.

Figura 2. Vista de despiece de un motor vibratorio ERM tipo disco



Fuente: “Coin Vibration Motors - Precision Microdrives”, consultado el 30 de mayo de 2020, <https://www.precisionmicrodrives.com/vibration-motors/coin-vibration-motors/>

Figura 3. Dirección de vibración de los motores ERM



Fuente: “AB-028 : Vibration Motor Comparison Guide - Precision Microdrives”, consultado el 30 de mayo de 2020, <https://www.precisionmicrodrives.com/content/ab-028-vibration-motor-comparison-guide/>

Para usar este tipo de motores como actuadores de una celda braille dinámica se limitó el movimiento en el eje X de forma tal que solo exista movimiento en el eje Z, permitiendo al punto braille subir y bajar en repetidas ocasiones cuando es activado y que permanezca abajo cuando no es activado.

⁸ “Coin Vibration Motors - Precision Microdrives”, consultado el 30 de mayo de 2020, <https://www.precisionmicrodrives.com/vibration-motors/coin-vibration-motors/>.

⁹ “AB-028 : Vibration Motor Comparison Guide - Precision Microdrives”, consultado el 30 de mayo de 2020. <https://www.precisionmicrodrives.com/content/ab-028-vibration-motor-comparison-guide/>.

4 RECONOCIMIENTO ÓPTICO DE CARACTERES

El reconocimiento óptico de caracteres (OCR por sus siglas en inglés), es una de las tareas fundamentales de la visión por computador que comprende una serie de técnicas ampliamente desarrolladas a lo largo de los años y cuya versatilidad ha permitido implementaciones que van desde soluciones con bajo costo computacional, usando el análisis matricial de las imágenes, hasta otras de mayor robustez, creadas mediante redes neuronales¹⁰. Esta evolución comprende además la aparición de múltiples niveles de abstracción a través de diferentes herramientas, permitiendo que importantes aplicaciones sean llevadas a cabo sin que el usuario tome partida directamente de todos los procesos subyacentes, como lo sería *Tesseract*, tecnología diseñada originalmente como un programa ejecutable desde la consola a través de líneas de comando y cuyo uso básico solo implica que el usuario especifique la imagen en la que se buscará el texto.

Para abordar la selección y uso de un determinado enfoque del OCR, es necesario evaluar diferentes aspectos fundamentales del escenario en el cual la aplicación se utilizará. Estos son¹¹:

- **Densidad del texto:** Si el texto en la imagen se encuentra focalizado o disperso
- **Estructura del texto:** Si el texto está presente bajo una estructura definida como, por ejemplo, a través de filas/líneas con un espaciamiento determinado entre ellas.
- **Fuente:** Si el texto es impreso o si es manuscrito
- **Tipo de carácter:** Uso de caracteres especiales, números, diferentes lenguajes, etc.
- **Artefactos:** Cantidad de ruido presente. Si la captura de las imágenes se realiza en entornos abiertos, bajo diferentes iluminaciones o bajo condiciones controladas, a través del uso de escáneres y/o en entorno cerrados.
- **Ubicación:** Si el texto en la imagen se encuentra en zonas totalmente definidas o si su ubicación tiene un comportamiento aleatorio.

Considerando cada uno de los factores antes mencionados se tendrá una noción íntegra de las necesidades y complejidad de la solución a implementar, de manera tal que pueda trazarse un plan de acción que incluya cada una de las etapas necesarias, evitando uso desmesurado de recursos, haciendo énfasis en la optimización del costo computacional que conlleva.

Durante el desarrollo de este dispositivo se procuró el uso de condiciones controladas, la aplicación pensada para usarse con texto impreso color negro, ubicación constante dentro del área de captura, fuente Arial 18, iluminación unilateral con LED, distancia focal estática y enfoque de acuerdo con las capacidades de la cámara (Raspberry Pi Camera Board v2). Teniendo en cuenta este escenario controlado, fue posible añadir flexibilidad a la estructuración y selección de las técnicas de reconocimiento, priorizando en la optimización del costo computacional debido al rendimiento limitado de la Raspberry Pi Zero W y evaluando incluso herramientas intuitivas tipo ready-to-use, como *Tesseract*.

¹⁰ "Optical Character Recognition", en *Wikipedia*, el 7 de abril de 2021, https://en.wikipedia.org/w/index.php?title=Optical_character_recognition&oldid=1016528333.

¹¹ Gidi Shperber, "A Gentle Introduction to OCR", Medium, el 15 de febrero de 2021, <https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa>.

5 DISPOSITIVO

Este prototipo fue diseñado para traducir texto impreso a braille mediante una cámara y una celda braille dinámica, además puede representar caracteres enviados de forma digital mediante la aplicación web, donde también se puede ajustar la duración, potencia y tipo de vibración de la celda. En la Figura 4 se puede ver una fotografía del dispositivo.

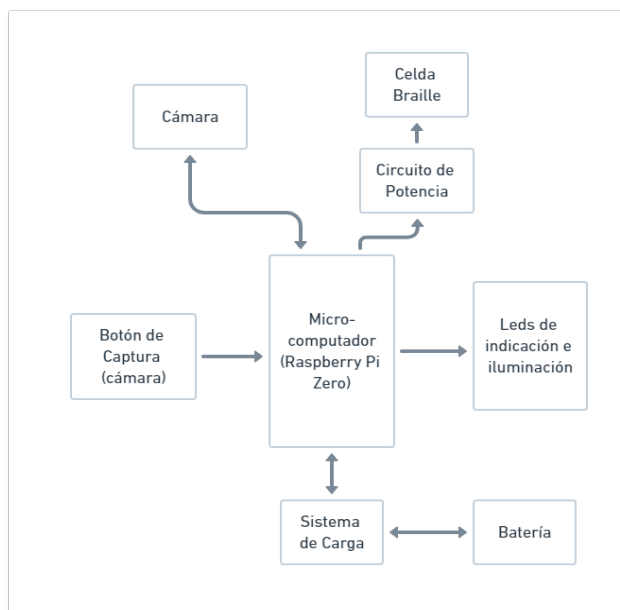
Figura 4. Fotografía del dispositivo



DIAGRAMA DE BLOQUES DEL PROTOTIPO

En la Figura 5 se muestran todos los componentes que comprende el equipo. En las subsecciones posteriores se encuentra una descripción detallada de cada uno de estos elementos.

Figura 5. Diagrama de bloques del prototipo.

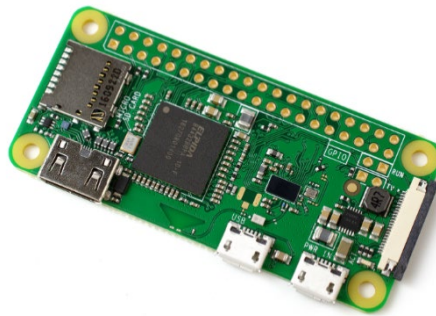


MICROCOMPUTADOR (RASPBERRY PI ZERO)

Esta computadora es el eje central del dispositivo, allí se procesan las imágenes que vienen de la cámara, se traducen los caracteres ASCII a braille, se envían los impulsos necesarios para activar la celda braille, se leen los comandos que vienen de la aplicación web, se envían las señales de indicación para los LEDs y se procesan las señales de los botones.

La Raspberry Pi Zero W (Figura 6) es una computadora programable de una sola placa de tamaño, precio y consumo energético reducido. Cuenta con entradas y salidas de propósito general (GPIO), 512 MB de RAM, un procesador de un solo núcleo de 1 GHz, conexión bluetooth y wifi, un puerto mini HDMI y dos puertos micro USB, uno de alimentación y otro para conectar dispositivos externos (teclado, ratón, USB hub, etc.)¹².

Figura 6. Imagen de la Raspberry Pi Zero W



Fuente: Gareth Halfacree from Bradford UK, The wireless variant of the Raspberry Pi Zero microcomputer, known as the Raspberry Pi Zero W., el 2 de marzo de 2017, el 2 de marzo de 2017, Raspberry Pi Zero W, [https://commons.wikimedia.org/wiki/File:Raspberry_Pi_Zero_W_\(33209067455\).png](https://commons.wikimedia.org/wiki/File:Raspberry_Pi_Zero_W_(33209067455).png)

CELDA BRAILLE DINÁMICA

La celda braille dinámica es el componente del dispositivo encargado de representar los caracteres en braille. Cuenta con 6 puntos organizados de la misma forma que el braille convencional, los puntos tienen 2.5 mm de diámetro y funcionan mediante un vibrador ERM tipo disco (Figura 7), como se describió en la sección 2.2 de este documento. Los vibradores tienen un voltaje nominal de 3V, funcionan con una corriente de 90mA y producen una frecuencia de vibración de 9000 RPM ¹³.

¹² The Raspberry Pi Foundation, "Buy a Raspberry Pi Zero W – Raspberry Pi", consultado el 31 de mayo de 2020, <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>.

¹³ Jinlong Machinery & Electronics, "Coin Vibration Motor Part No. C1026B002F Datasheet", el 11 de septiembre de 2009, <https://www.parallax.com/sites/default/files/downloads/28821-Flat-Coin-Vibration-Motor-Documentation.pdf>.

Figura 7. Imagen de un vibrador ERM tipo disco

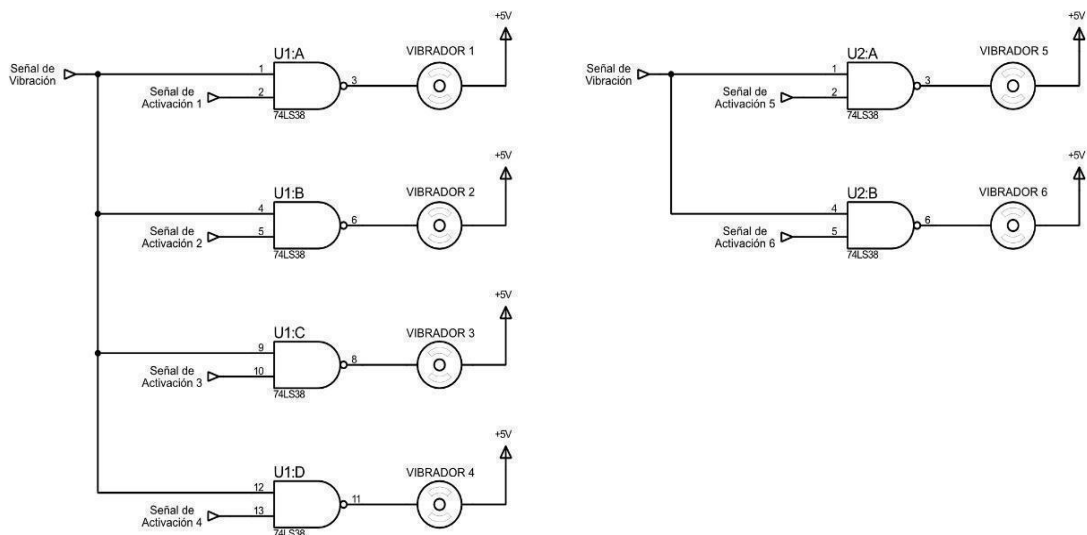


Fuente: Tatoko, "Tatoko 20PCS 10mmx3mm Mini Vibration Motors DC 3V Flat Coin Button-Type"

CIRCUITO DE POTENCIA

El mecanismo de activación de cada uno de los puntos es igual al de una compuerta AND, donde por un lado tenemos la señal de vibración (PWM), que es común para los 6 puntos y sirve para regular la potencia y crear diferentes tipos de vibración, y por otro lado la señal de activación (señal digital, 0V o +5V), que es individual y sirve para determinar cuáles puntos se activan. Para su implementación se usaron dos unidades del circuito integrado **74LS38** que cuenta con 4 compuertas NAND independientes de colector abierto ¹⁴. El diagrama del circuito de potencia se muestra en la siguiente figura (Figura 8):

Figura 8. Diagrama esquemático del circuito de potencia la celda braille dinámica



¹⁴ Fairchild Semiconductor International, Inc., "DM74LS38 Quad 2-Input NAND Buffer with Open-Collector Outputs Datasheet", mayo de 2000, <https://www.futurlec.com/Datasheet/74Is/74LS38.pdf>.

A continuación, en las Figuras 9 y 12 se ve una representación isométrica de la celda braille, en las Figuras 10 y 13 se ven los planos acotados de la celda y en la Figura 11 se ve una fotografía de la celda braille implementada.

Figura 9. Vista isométrica de la celda braille dinámica

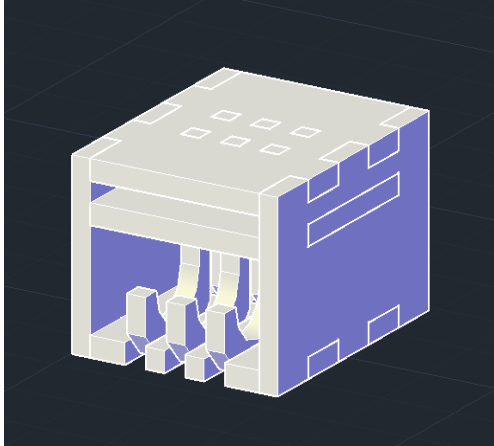


Figura 12. Vista de despiece de la celda braille dinámica

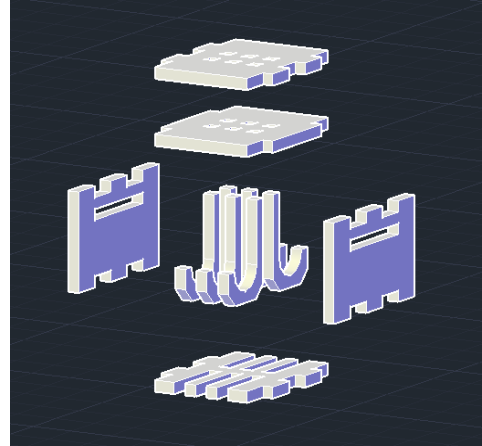
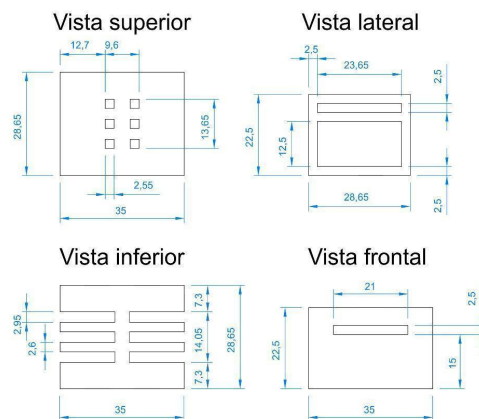
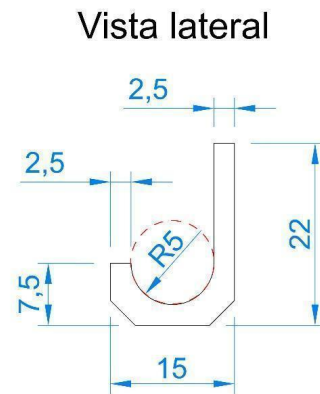


Figura 10. Vistas acotadas de la celda braille dinámica



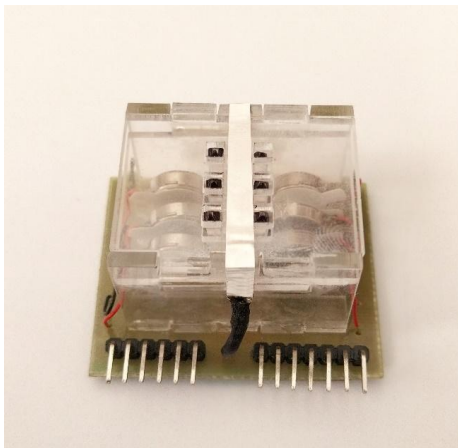
* Todas las medidas expresadas en esta ilustración están dadas en milímetros.

Figura 13. Vista lateral acotada de un punto de la celda braille dinámica



* Todas las medidas expresadas en esta ilustración están dadas en milímetros.

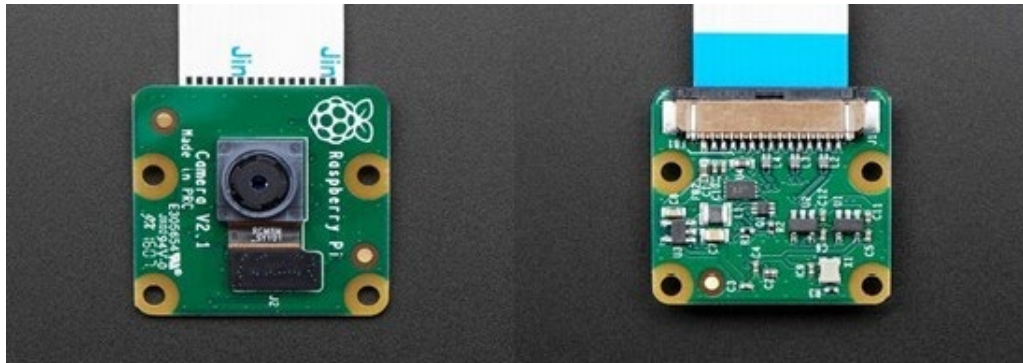
Figura 11. Fotografía de la celda braille dinámica



CÁMARA

La cámara seleccionada fue la *Raspberry Pi Camera Board v2* (Figura 14).

Figura 14. Raspberry Pi Camera Board v2. Vista frontal y posterior



Fuente: Adafruit Industries, “Raspberry Pi Camera Board v2 - 8 Megapixels”, consultado el 3 de mayo de 2021, <https://www.adafruit.com/product/3099>

Entre sus principales características encontramos las siguientes:

- 8 megapíxeles de resolución nativa
- Sensor de imagen Sony IMX219
- Tamaño del sensor óptico: 1/4"
- Píxel de 1.12 μm x 1.12 μm con tecnología OmniBSI de alto rendimiento (alta sensibilidad, diafonía baja, bajo ruido)
- Enfoque ajustable de manera manual.

BOTONES DE CAPTURA Y ENCENDIDO

El botón de captura se encarga de activar mediante un pin de interrupción la función de la cámara requerida para tomar y procesar una foto.

El botón de encendido se encarga de conectar el pin P6-01 (RUN) de la Raspberry Pi Zero a tierra, haciendo que se reinicie el procesador y con esto, que se encienda la placa.¹⁵

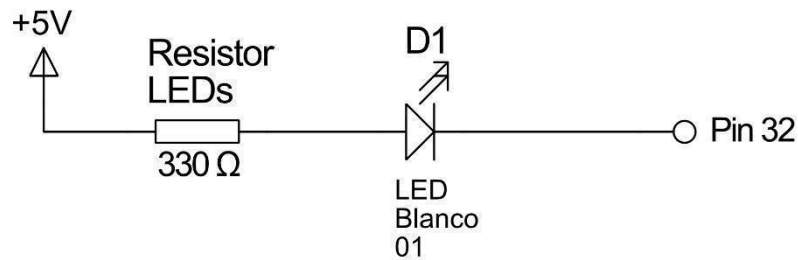
ILUMINACIÓN

El dispositivo cuenta con un LED blanco para iluminar el texto capturado por la cámara, Este LED tiene un ángulo de iluminación de 25 grados, 5 mm de diámetro, un consumo aproximado de 3.5 V a 25 mA, una intensidad lumínica de entre 1500 y 3000 mcd¹⁶ y está conectado de forma tal que se active en bajo, es decir, cuando el pin de control esté en 0V. En la Figura 15 se muestra el diagrama de conexión para esta parte de la circuitería.

¹⁵ “RPi Low-Level Peripherals”, en *Ellinux.Org*, el 28 de junio de 2019, https://elinux.org/RPi_Low-level_peripherals#P6_header.

¹⁶ “White 5mm Water Clear LED - 25deg - 2250mCd”, Kitronik Ltd, consultado el 3 de junio de 2020, <https://kitronik.co.uk/products/35109-white-5mm-water-clear-led-25deg-2250mcd>.

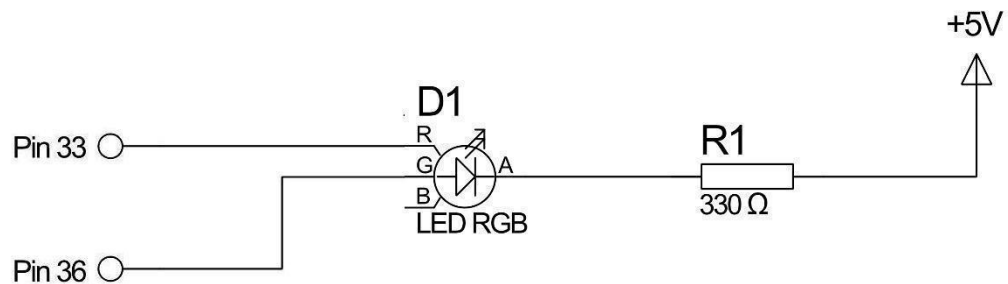
Figura 15. Circuito esquemático de los LED de iluminación de la cámara



Además de lo anterior, el dispositivo tiene un LED RGB de indicación que se ilumina en rojo cuando está en “modo analógico”, es decir, cuando recibe y procesa datos de la cámara; se ilumina en verde cuando está en “modo digital”, es decir, cuando recibe y procesa datos de la aplicación web y cuando está en el modo hibernación se apaga.

Se usó un LED RGB de ánodo común, 5 mm de diámetro, con un consumo aproximado de 3.5V a 30 mA por cada canal de color, un ángulo de iluminación de 25 grados aproximadamente y una intensidad lumínica de 800 mcd para el canal rojo y 4000 mcd para el canal verde ¹⁷. En la Figura 16 se muestra el diagrama esquemático para la parte de la circuitería correspondiente al LED de indicación.

Figura 16. Circuito esquemático del LED de indicación

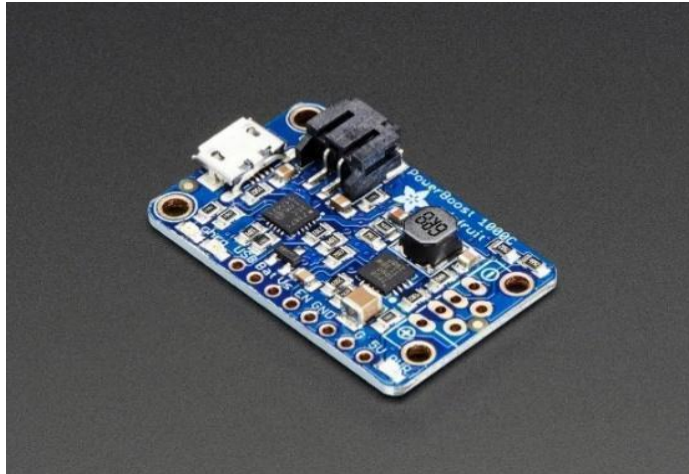


¹⁷ CHINA YOUNG SUN LED TECHNOLOGY CO., LTD., “RED/GREEN/BLUE Triple Color LED Model No.: YSL-R596CR3G4B5C-C10 Datasheet” (Sparkfun), consultado el 31 de mayo de 2020, <https://www.sparkfun.com/datasheets/Components/YSL-R596CR3G4B5C-C10.pdf>.

SISTEMA DE CARGA

El sistema de carga consta de dos bloques. Por un lado, está el módulo **Powerboost 1000C** de Adafruit (Figura 17), un módulo de carga para baterías de ion de litio y baterías de polímero de litio de 3.7V. Tiene una entrada micro-USB, soporta hasta 1A de carga, su eficiencia es del 90% y cuenta además con un pin de Enable para habilitar la salida y un pin Bat para poder medir el voltaje de la batería ¹⁸.

Figura 17. Vista isométrica del módulo Powerboost 1000C



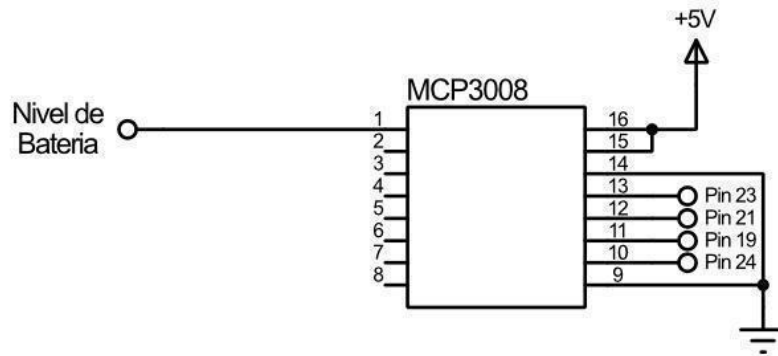
Fuente: Adafruit Industries, "PowerBoost 1000 Charger - Rechargeable 5V Lipo USB Boost @ 1A", consultado el 4 de junio de 2020, <https://www.adafruit.com/product/2465>

Por otro lado, está el circuito integrado **MCP3008**, un conversor análogo-digital (ADC) de 8 canales con una resolución de 10-bits y una comunicación serial mediante el protocolo SPI ¹⁹. Se usó este conversor debido a que la Raspberry no tiene ningún módulo de conversión análogo-digital integrado y porque su puerto de comunicación SPI permite usar los pines de la Raspberry dedicados a ese protocolo, en lugar de los pines de propósito general (GPIO). En la Figura 18 se encuentra el diagrama esquemático usado para la conexión del integrado.

¹⁸ Adafruit Industries, "PowerBoost 1000 Charger - Rechargeable 5V Lipo USB Boost @ 1A", consultado el 4 de junio de 2020, <https://www.adafruit.com/product/2465>.

¹⁹ Microchip Technology Inc., "MCP3004/3008 2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface Datasheet" (Microchip Technology Inc., 2008), <http://ww1.microchip.com/downloads/en/DeviceDoc/21295d.pdf>.

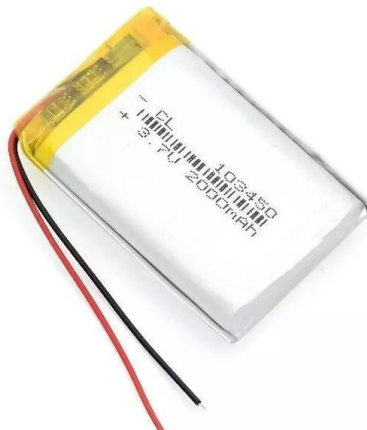
Figura 18. Circuito esquemático del conversor análogo-digital MCP3008



BATERÍA

Finalmente tenemos la batería, se usó una batería de polímero de litio (Figura 19) de 2000 mAh, con unas dimensiones de 10 mm x 34 mm x 50 mm y un peso aproximado de 37 gramos. Tiene un voltaje nominal de 3.7V, un voltaje máximo de 4.2V y cuenta con un circuito de protección para prevenir daño por sobre carga o descarga ²⁰. Elegimos esta batería por su reducido tamaño y considerable capacidad de carga.

Figura 19. Imagen de la batería de polímero de litio



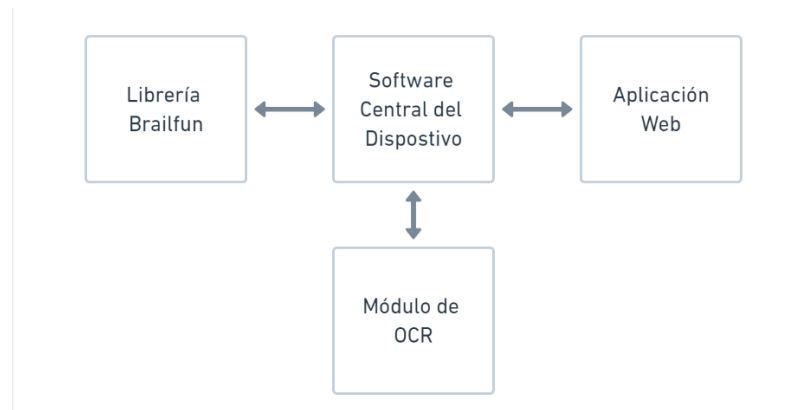
Fuente: Control Dinamico SAS, "Bateria Polimero De Litio 3.7v, 2000mah 103450 Lipo - \$ 27.800", MercadoLibre, consultado el 4 de junio de 2020, https://articulo.mercadolibre.com.co/MCO-485643308-bateria-polimero-de-litio-37v-2000mah-103450-lipo-_JM

²⁰ Control Dinamico SAS, "Bateria Polimero De Litio 3.7v, 2000mah 103450 Lipo - \$ 27.800", MercadoLibre, consultado el 4 de junio de 2020, https://articulo.mercadolibre.com.co/MCO-485643308-bateria-polimero-de-litio-37v-2000mah-103450-lipo-_JM.

6 SOFTWARE

Este dispositivo a nivel de software cuenta con cuatro componentes que se relacionan entre sí como se muestra en la Figura 20.

Figura 20. Diagrama del software del dispositivo

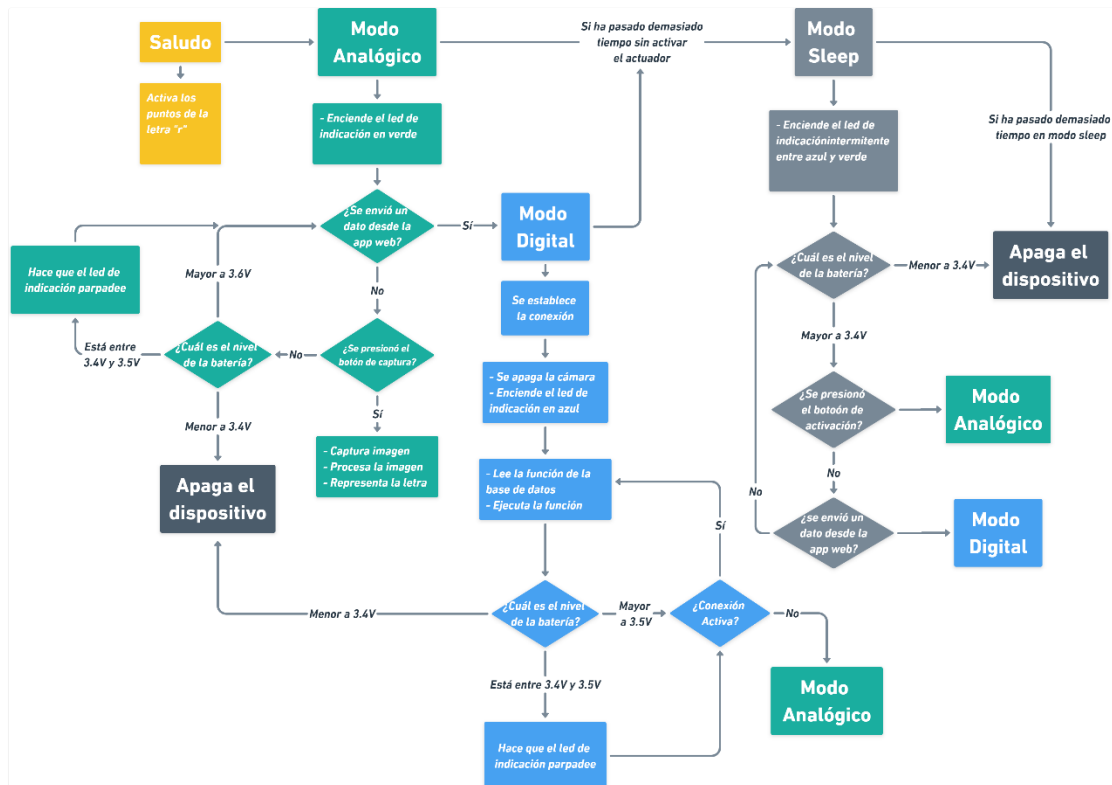


1. El software central del dispositivo, desarrollado en Python, configurado para iniciar durante el encendido de la Raspberry y el cual se ejecuta de forma cíclica pasando por múltiples etapas durante la interacción con el usuario.
2. La librería *Brailfun*, desarrollada para este proyecto y cuya función principal es generar las señales que controlan la celda braille dinámica y transformar texto ASCII a braille.
3. La aplicación web, que sirve para enviar palabras o conjuntos de puntos al dispositivo y también para configurar características de la vibración tales como potencia, tiempo de duración y tipo de señal de vibración.
4. El módulo de OCR, diseñado para activarse con señal externa a través del botón correspondiente y cuyo propósito es capturar una imagen, ejecutar cada una de las etapas del procesamiento, realizar el reconocimiento del texto presente en el área objetivo y brindar como salida los caracteres ASCII reconocidos.

6.1 SOFTWARE CENTRAL DEL DISPOSITIVO

En la Figura 21 se presenta el diagrama de flujo del software del dispositivo y posteriormente se encuentran descritas cada una de las etapas que lo conforman.

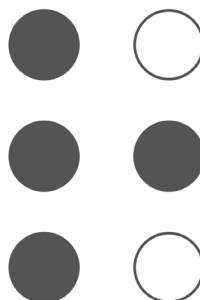
Figura 21. Diagrama de flujo del software central del dispositivo



6.1.1 Saludo

Este módulo se ejecuta después de cargar todas las librerías y las configuraciones del dispositivo. Lo que hace es representar la letra “r”, tal como muestra la Figura 22, lo que indica al usuario que el dispositivo está encendido y se inició de forma correcta.

Figura 22. Representación gráfica de la letra 'r' en braille



6.1.2 Modo analógico

En este modo se capturan y procesan datos de la cámara. Es el primer modo en el que entra el dispositivo después del saludo inicial (encendido/puesta en marcha). Inicialmente se enciende la cámara y el LED de iluminación. El LED de indicación se torna rojo, posteriormente se inicia un ciclo en el cual comprueba si se enviaron datos desde la aplicación web, en cuyo caso pasa al modo digital, de lo contrario se captura una foto con la cámara, se procesa y se representa la letra resultante en la celda braille.

6.1.3 Modo digital

En este modo se capturan y procesan datos de la aplicación web. Para entrar en este modo se tiene que presionar el botón “establecer conexión” en la interfaz web. Cuando se establece la conexión, la cámara y el LED de iluminación se apagan y el LED de indicación se torna verde, posteriormente se inicia un ciclo en el cual se lee y ejecuta la función enviada por el usuario.

6.1.4 Modo sleep

Este es el modo hibernación o de ahorro de batería. En este modo se apaga la cámara y los LEDs del dispositivo. Para entrar en este modo el dispositivo debe estar inactivo por un tiempo establecido, por defecto son 15 minutos. Si el dispositivo se mantiene en el modo sleep por más de 15 minutos, se apagará. Para salir del modo sleep hay dos opciones: Se puede presionar el botón de activación en la celda braille dinámica, lo que hará que entre en el modo analógico, o se puede presionar el botón “establecer conexión” en el aplicativo web para entrar al modo digital.

6.1.5 Librería brailfun

Esta librería de Python se creó con el propósito de agrupar de forma organizada todos los atributos y funciones que se usan para controlar la celda dinámica braille. Contiene la clase *new_cell*, con la cual se pueden crear objetos llamados celdas, con sus propios atributos y funciones. Los atributos asociados al objeto son los siguientes:

vibration_pins: Esta variable contiene un vector de 7 valores, donde el primer valor es el pin donde va conectada la señal de vibración PWM y los otros seis son los pines de activación de cada uno de los vibradores. Por defecto son los siguientes pines BCM [18, 4, 17, 27, 22, 23, 24]

power: Esta variable contiene un valor flotante entre 0.0 y 5.0, con el cual se regula la potencia de vibración. Por defecto es 4.

time_on: Esta variable contiene un valor flotante entre 0.0 e infinito que determina el tiempo en segundos que deben permanecer activos los vibradores cuando se envía una señal. Por defecto es 3.

time_off: Esta variable contiene un valor flotante entre 0.0 e infinito que determina el tiempo en segundos que deben permanecer inactivos los vibradores entre una señal y la siguiente. Por defecto es 1.

signal_type: Esta variable contiene un valor entero entre 1 y 8 que determina qué tipo de señal de vibración PWM se envía a los vibradores cuando se activan. La señal puede ser cuadrada, triangular, click (logarítmica + exponencial), rampa, exponencial, sinusoidal, reverse click (exponencial + logarítmica), logarítmica. Por defecto es 1. A continuación, se muestran las funciones que tiene el objeto:

pinout: Esta función recibe como entrada 7 variables correspondientes a los pines de vibración y activación de la celda braille y sirve para cambiar los pines que vienen por defecto para controlar la celda braille.

randletter: Esta función no recibe ninguna entrada y devuelve una letra al azar en ASCII.

translator: Esta función recibe una letra ASCII y devuelve su equivalente braille en un vector de 6 valores binarios.

trigger: Esta función recibe un vector de seis valores binarios y activa la celda dinámica braille de acuerdo con el vector de entrada, tomando en cuenta los atributos del objeto.

generator: Esta función no recibe ninguna entrada y lo que hace es activar individualmente cada uno de los puntos de la celda braille en orden ascendente y de forma consecutiva.

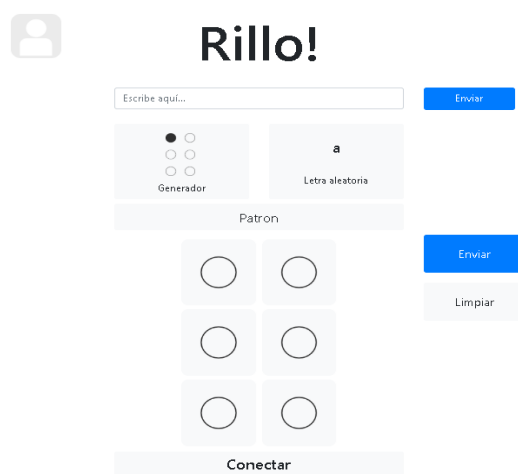
writer: Esta función recibe un conjunto de letras ASCII, las traduce y representa en la celda braille dinámica.

randomvib: Esta función activa un conjunto de puntos al azar en la celda braille dinámica.

6.2 APLICACIÓN WEB

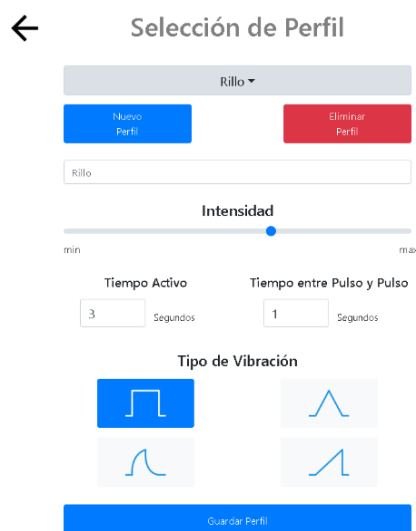
El aplicativo web fue creado para brindar a los usuarios del dispositivo de enseñanza braille, un entorno diferente al entorno controlado planteado inicialmente para el uso de la herramienta. En esta aplicación se puede enviar una señal *generator*, la cual, como se describió en la sección anterior para la función correspondiente, hará que se activen individualmente cada uno de los puntos de la celda braille por orden ascendente y de forma consecutiva, además de permitir la representación de un conjunto de letras, una letra al azar o un patrón determinado, según lo que se ingrese en la interfaz mostrada en la Figura 23.

Figura 23. Menú principal de la aplicación web



Adicionalmente, es posible cambiar la configuración de la celda a través de la creación de diferentes perfiles, de manera tal que cada usuario del dispositivo personalice parámetros como potencia de vibración, tiempo de activación, tiempo de espera entre señales y tipo de señal de acuerdo con sus preferencias y necesidades. La Figura 24 muestra la interfaz para la personalización de los perfiles de usuario.

Figura 24. Menú de configuración de perfil de la aplicación web



La comunicación entre el dispositivo y la aplicación web se hace mediante internet usando una base de datos en tiempo real de la plataforma Firebase.

Firebase es una plataforma de desarrollo para aplicaciones web y móvil ²¹ que cuenta con una base de datos multiplataforma NoSQL en la nube, la cual sincroniza los datos (almacenados en formato JSON²²) para todos los usuarios de la aplicación en tiempo real. En la base de datos se guardan dos conjuntos de datos que son leídos y modificados tanto por la aplicación como por el dispositivo.

Por un lado, está el conjunto de datos llamado “funciones”, en el cual se guardan los datos correspondientes a los comandos enviados por la aplicación, donde tendremos variables como *conectado*, que es un booleano que indica si hay una conexión establecida entre el dispositivo y la aplicación web; también está la variable *función*, que indica qué comando se está enviando desde la aplicación web al dispositivo; la variable *dato*, que es complementaria a la variable *función* y que contiene los datos que se van a ejecutar con la función, es decir, si la función es “escribir”, entonces la variable *dato* tendría las letras que se representarán en la celda Braille. Por último, está la variable *recibido*, que indica si la función fue recibida y ejecutada por el dispositivo. El otro conjunto de datos se llama “perfiles”, y en él se guarda una variable por cada perfil creado. Esta variable es un vector que contiene las características de vibración de ese perfil en concreto y una variable booleana que indica cuál es el perfil activo.

Figura 25. Captura del conjunto de datos llamado “funciones” con sus respectivos datos

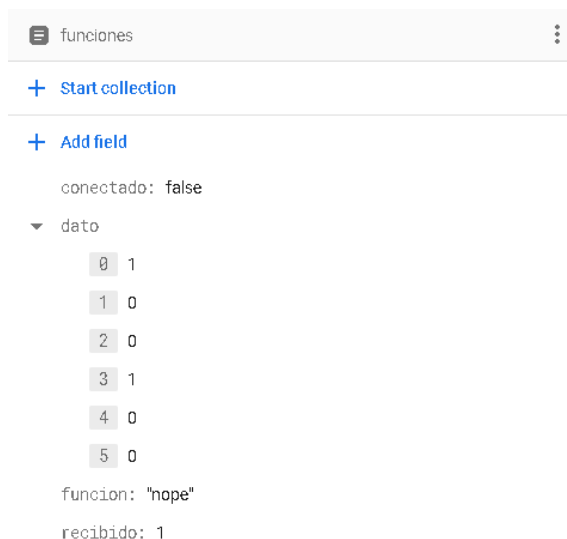
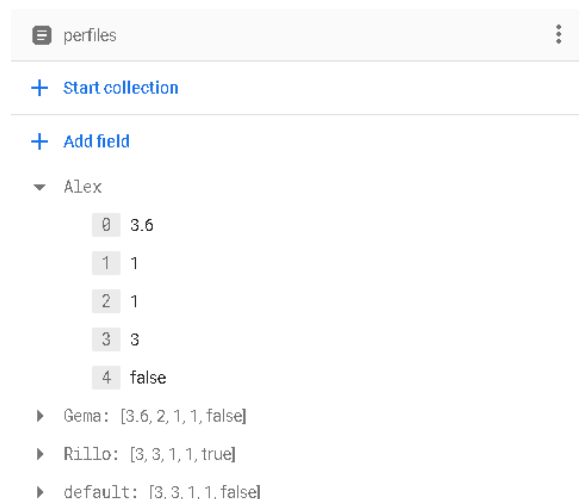


Figura 26. Captura del conjunto de datos llamado “perfiles” con sus respectivos datos



²¹ “Firebase”, en *Wikipedia*, el 13 de mayo de 2020, <https://en.wikipedia.org/w/index.php?title=Firebase&oldid=956484625>.

²² “Firebase Realtime Database”, Firebase Documentation, consultado el 6 de junio de 2020, <https://firebase.google.com/docs/database>.

7 SISTEMA DE RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR)

Para el sistema de reconocimiento óptico de caracteres y considerando lo explicado en la sección 3, fue seleccionado el análisis/procesamiento matricial de las imágenes en conjunto con un algoritmo KNN (K-Nearest-Neighbor)²³ básico de Machine Learning de tipo supervisado. El algoritmo está diseñado para trabajar como una función que se activa al presionar el botón de captura en el equipo y cuyo propósito general es digitalizar la imagen, procesarla, realizar el reconocimiento de caracteres y dar como salida una lista con los caracteres en formato ASCII identificados.

Para el algoritmo se usaron las dos siguientes librerías:

- Opencv²⁴: Es una librería de visión por computador y Machine Learning de software libre. Fue creada con el fin para establecer una estructura común para las aplicaciones de visión por computador y promover el uso de este tipo de tecnologías en los productos de la industria.
- Numpy²⁵: Es una librería de software libre cuyo propósito principal es hacer posible la computación numérica con *Python*.

7.1 DIAGRAMA DE BLOQUES DEL SISTEMA DE RECONOCIMIENTO ÓPTICO DE CARACTERES.

Con el ánimo de reducir al máximo la carga computacional del algoritmo y considerando que el modelo KNN, al ser una técnica de Machine Learning de tipo supervisado, implica un entrenamiento previo, se subdividió el sistema en dos bloques principales: Entrenamiento y Reconocimiento; los cuales difieren entre sí en algunas de las etapas del procesamiento y en la salida esperada. El algoritmo presente en el prototipo final es el algoritmo de reconocimiento. La etapa de entrenamiento fue realizada empleando como insumo diferentes imágenes tomadas con la cámara bajo las mismas condiciones de uso del dispositivo final, las cuales fueron posteriormente clasificadas en dos grupos: imágenes de entrenamiento e imágenes de prueba; con estas últimas se realizó la validación del algoritmo. En las Figuras 27 y 28 se pueden ver los diagramas de bloques del algoritmo.

Figura 27. Diagrama de bloques del algoritmo de entrenamiento del sistema de reconocimiento óptico de caracteres

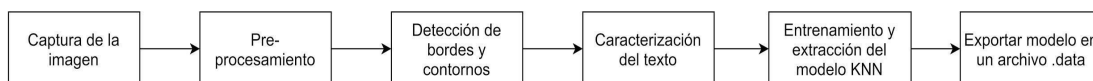
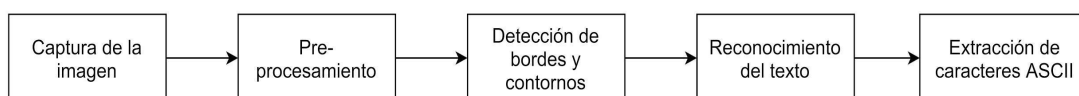


Figura 28. Diagrama de bloques del algoritmo de reconocimiento del sistema OCR



²³ “k-nearest neighbors algorithm - Wikipedia”, consultado el 4 de mayo de 2021, https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

²⁴ “About”, OpenCV documentation, consultado el 2 de abril de 2021, <https://opencv.org/about/>

²⁵ NumPy documentation, consultado el 2 de abril de 2021, <https://numpy.org/doc/stable/>

7.2 ADQUISICIÓN/CAPTURA DE LA IMAGEN

En esta etapa se realiza la digitalización²⁶ de la imagen. La captura se realiza al presionar el botón de captura cuando el dispositivo se encuentra en la posición deseada. Al detectar esta señal, la Raspberry ejecuta el algoritmo mediante el cual se captura la imagen.

7.3 PREPROCESAMIENTO

En esta etapa se realiza un acondicionamiento previo de la captura/imagen. El objetivo principal de esta etapa es adecuar la imagen con la que se trabajará, eliminando la información innecesaria y de esta manera, permitiendo al algoritmo extraer las características de interés de forma eficiente y efectiva²⁷. Comprende los siguientes procesos:

- Redimensionamiento, para estandarizar el tamaño de las imágenes y disminuir la cantidad de píxeles/información con la que tendrá que lidiar el modelo.
- Selección del área objetivo a partir del área de cobertura total disponible, para descartar las secciones que no contienen información de interés.
- Conversión de la imagen capturada a escala de grises, proceso después del cual la imagen pasa de contener información de 3 canales (uno por cada color básico), a un solo canal, correspondiente a una imagen que considera sólo variaciones en la escala de grises.
- Suavizado de bordes, para eliminar el ruido y conservar los patrones de mayor relevancia en la imagen.

7.4 DETECCIÓN DE BORDES Y CONTORNOS

En esta etapa se realiza la segmentación de la imagen. Para ello se emplea un método de umbralización adaptativo y posteriormente se realiza la identificación de contornos²⁸. Finalmente se realiza un filtrado de los contornos no deseados estableciendo límites para los parámetros asociados a cada uno de ellos, tales como, altura, área, ancho, etc. Estos límites se establecen conociendo los valores característicos y esperados para estos parámetros bajo las condiciones definidas para el uso del dispositivo, tales como distancia focal de la cámara, tamaño de la fuente, etc.

7.5 CARACTERIZACIÓN DEL TEXTO

En esta etapa se llevó a cabo el etiquetado de cada uno de los contornos identificados, esto es, cada una de las áreas/porciones de la imagen, resultantes del procesamiento anterior, fueron asignadas de forma manual a la letra del abecedario que representan. De esta manera se obtienen los datos de entrenamiento y validación para el modelo.

²⁶ "Digitization", en *Wikipedia*, el 30 de abril de 2021, <https://en.wikipedia.org/w/index.php?title=Digitization&oldid=1020616029>.

²⁷ BARATH KUMAR, "Image Preprocessing - Why Is It Necessary?", Medium, el 10 de febrero de 2021, <https://medium.com/spidernitt/image-preprocessing-why-is-it-necessary-8895b8b08c1d>.

²⁸ "Edge Detection", en *Wikipedia*, el 30 de abril de 2021, https://en.wikipedia.org/w/index.php?title=Edge_detection&oldid=1020616177.

7.6 ENTRENAMIENTO Y EXTRACCIÓN DEL MODELO

Usando los datos de entrenamiento y validación de la etapa anterior, se entrena el modelo KNN y posteriormente es extraído con extensión .data para ser usado como insumo en el algoritmo de reconocimiento, presente en el dispositivo final.

7.7 RECONOCIMIENTO DEL TEXTO

En esta etapa se realiza la comparación directa de la imagen capturada y el set de datos entrenado con el modelo en el algoritmo de entrenamiento. El método empleado y de acuerdo con el funcionamiento general del algoritmo KNN (K-Nearest-Neighbors), predice el resultado a partir de la comparación entre la matriz de datos de la imagen actual y el conjunto de matrices de datos de las imágenes utilizadas con el entrenamiento del modelo, evaluando una cantidad de muestras $k=1$ (para nuestro caso) cercanas a la muestra de la comparación.

7.8 EXTRACCIÓN DE CARACTERES ASCII

Una vez se ha clasificado la matriz de datos de la imagen actual, se envía la letra codificada en formato ASCII a la salida del algoritmo de reconocimiento para ser utilizado en etapas posteriores del algoritmo principal en la representación de su equivalente en Braille a través de la celda dinámica.

8 RESULTADOS

A lo largo del proyecto se hicieron múltiples pruebas con diferentes enfoques que buscaban evaluar de forma, tanto independiente como conjunta, los diferentes componentes del dispositivo.

Por un lado, se hizo una prueba a la celda braille dinámica, en la cual se verificó su capacidad para representar caracteres del sistema braille de forma legible, luego se evaluó la cámara con el fin de establecer los parámetros básicos de operación del dispositivo, posteriormente se midió el rendimiento del algoritmo de OCR, comparando los resultados del análisis matricial y los obtenidos con *Tesseract*, y Finalmente se llevaron a cabo pruebas end-to-end al dispositivo, en las cuales se midieron los tiempos de respuesta, la autonomía y los tiempos de encendido y apagado del dispositivo. Además de lo anterior y de forma paralela a las pruebas antes mencionadas, se realizaron múltiples pruebas y ajustes al diseño de la carcasa del dispositivos, evaluando factores como ergonomía, practicidad, portabilidad y facilidad para la integración.

DISPOSITIVO

El dispositivo cuenta con dos botones, uno de captura cuya función es activar la función de la cámara para tomar una foto, además de esto tiene un LED blanco para iluminar el texto capturado por la cámara y un led RGB de indicación en el cual se puede ver el modo de operación actual (apagado, encendido, en hibernación, leyendo datos de la cámara o de la aplicación web). En la Figura 29 se muestra el modelo final usado para el dispositivo.

Figura 29. Fotografía isométrica del dispositivo



Figura 30. Fotografía isométrica del dispositivo destapado

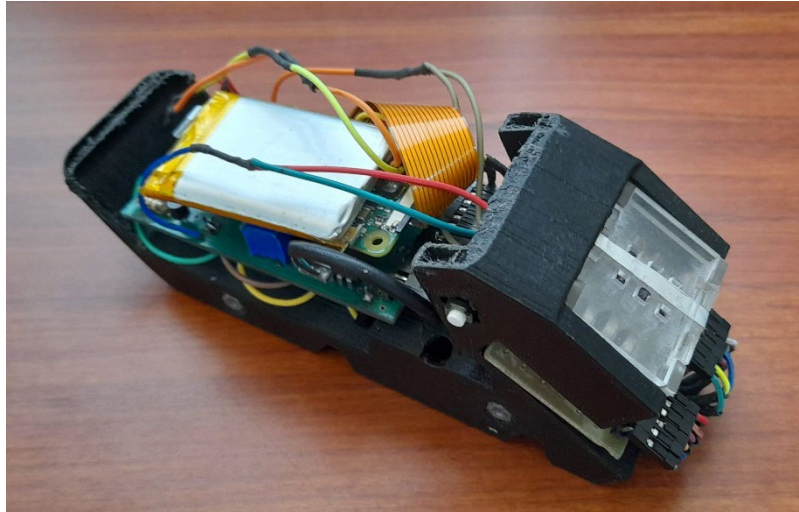


Figura 31. Fotografía lateral del dispositivo destapado

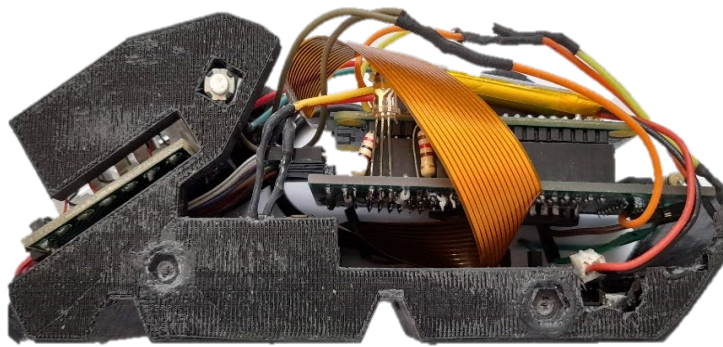
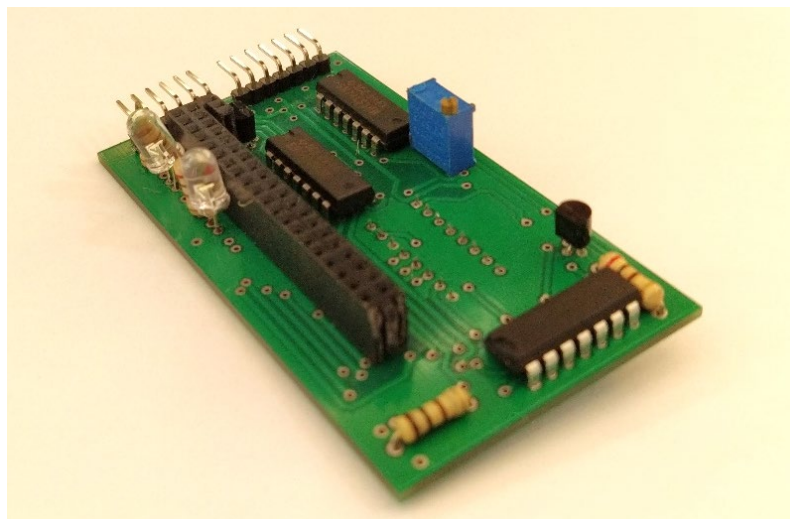


Figura 32. Fotografía de la PCB del dispositivo



PRUEBAS DE LA CELDA BRAILLE DINÁMICA

Con esta prueba se midió la capacidad del dispositivo para representar combinaciones de puntos de forma legible tomando en cuenta tres variables: la cantidad de intentos por estímulo, el tiempo de lectura y el porcentaje de acierto. En esta participaron 6 personas videntes que no tenían un conocimiento previo del sistema braille.

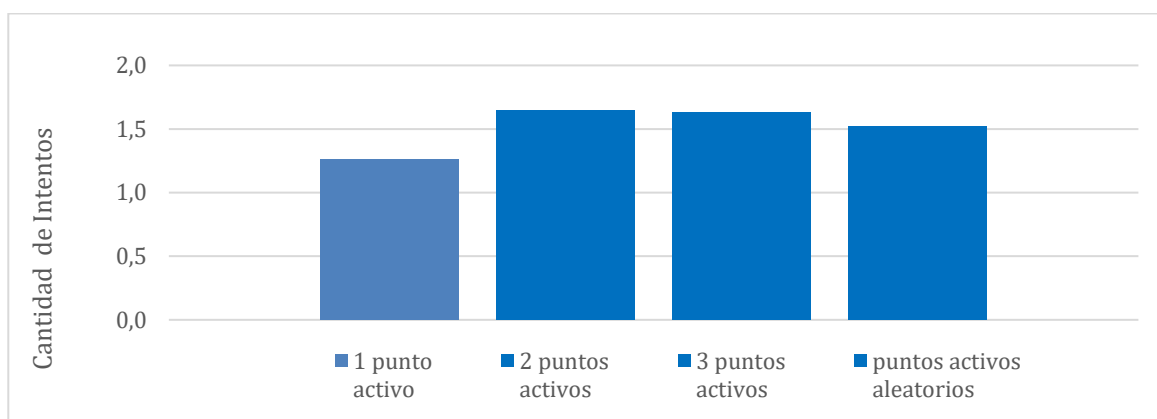
Inicialmente se les explicó en qué consistía la prueba, su objetivo y se les pidió su autorización para guardar y analizar los datos adquiridos durante el desarrollo de la prueba, posteriormente se hizo un entrenamiento en el cual se les explicó qué era una celda braille, donde estaban ubicados los puntos y se enviaron varios estímulos de manera tal que pudieran familiarizarse con la sensación.

La prueba estuvo constituida de cuatro etapas, en cada una se enviaron 9 estímulos y se les pidió que dijeran en voz alta qué puntos habían sido activados. El participante se podía tomar todo el tiempo que quisiera antes de dar su respuesta y podía pedir que se repitiera el estímulo la cantidad de veces que considerara necesaria. En la primera etapa se enviaron estímulos con 1 punto activo, en la segunda estímulos con 2 puntos activos, en la tercera estímulos con 3 puntos activos y en la cuarta se enviaron estímulos con 1, 2 o 3 puntos activos de forma aleatoria.

Cuadro 1. Cantidad promedio de intentos por estímulo

	Participante						
	1	2	3	4	5	6	Promedio
1 punto activo	1.2	1.6	1.1	1.3	1.3	1.0	1.3
2 puntos activos	1.6	1.6	2.3	2.2	1.2	1.0	1.6
3 puntos activos	2.7	1.6	1.6	2.0	1.0	1.0	1.6
Puntos activos aleatorios	1.8	2.1	1.8	1.3	1.1	1.0	1.5

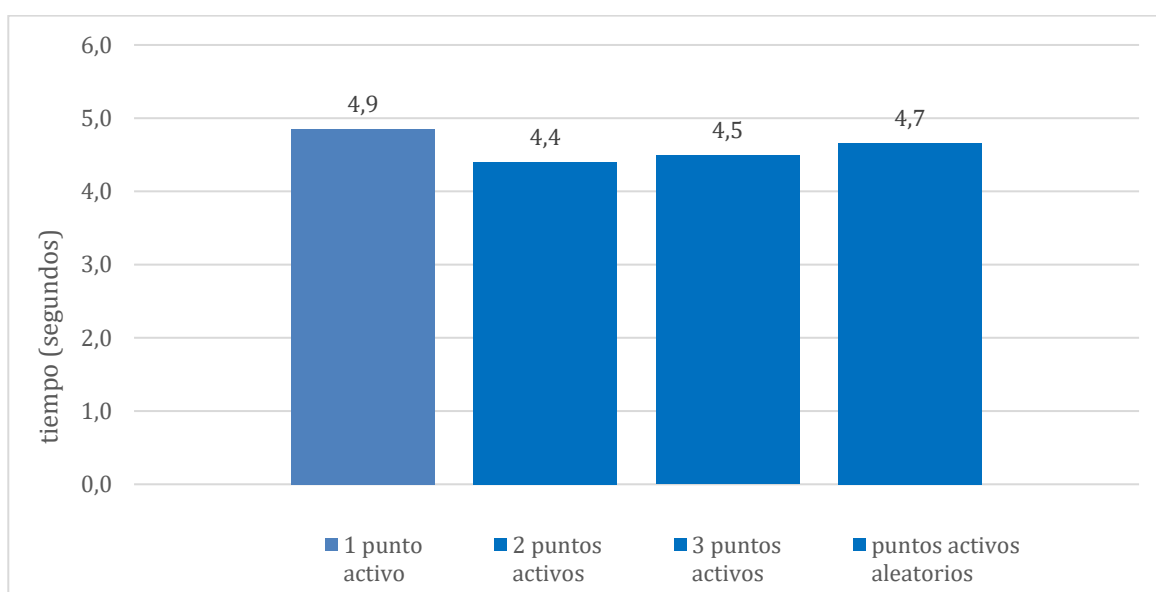
Figura 33. Diagrama de barras de la cantidad promedio de intentos por estímulo



Cuadro 2. Tiempo de lectura promedio en segundos por estímulo

	Participante						
Participante	1	2	3	4	5	6	Promedio
1 punto activo	4.0	3.8	3.0	12.8	2.6	2.9	4.9
2 puntos activos	3.9	3.3	5.0	7.9	3.2	3.1	4.4
3 puntos activos	4.5	3.5	3.1	8.9	3.2	3.7	4.5
Puntos activos aleatorios	6.1	3.5	2.7	10.2	2.8	2.6	4.7

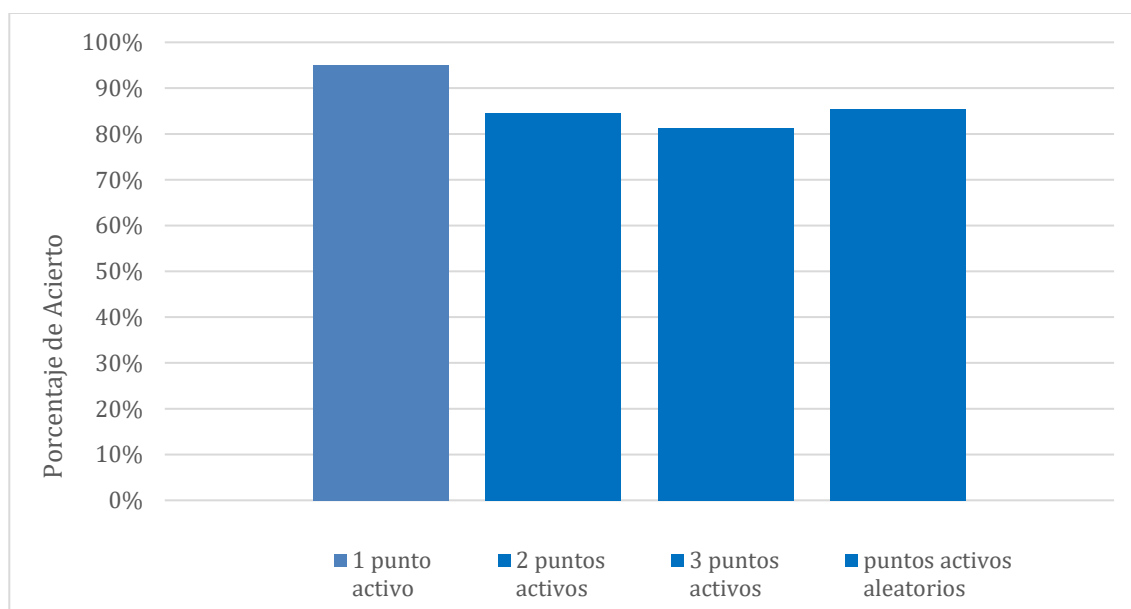
Figura 34. Diagrama de barras del tiempo de lectura promedio por estímulo



Cuadro 3. Porcentaje de acierto promedio por estímulo

	Participante						
	1	2	3	4	5	6	Promedio
1 punto activo	93%	100%	100%	85%	100%	93%	95%
2 puntos activos	81%	93%	85%	78%	96%	74%	85%
3 puntos activos	78%	96%	96%	61%	78%	78%	81%
Puntos activos aleatorios	89%	89%	100%	67%	76%	93%	86%

Figura 35. Diagrama de barras del porcentaje de acierto promedio por estímulo



PRUEBAS DE LA CÁMARA

Con esta prueba se buscó definir la distancia mínima permitida entre el papel y el lente de la cámara de manera tal que el enfoque pudiera ajustarse para que la calidad y definición de imagen ofrecida por la cámara no se viera afectada. Los resultados muestran los siguiente:

Cuadro 4. Calidad de la imagen según la distancia entre el lente y el papel

Distancia entre el papel y el lente (cm)	Calidad de imagen (enfoque)		
	Deficiente	Regular	Óptima
1	x		
1.2		x	
1.4		x	
1.6			x
1.8	x		

Para determinar el tamaño de la letra usando la fuente Arial se realizaron capturas manteniendo una distancia focal constante de 1.6 cm y se evaluó si el área contenía 3 letras consecutivas.

Cuadro 5. Tamaño de letra necesario para capturar totalmente un máximo de 3 letras por imagen

Tamaño de fuente (Arial)	Cobertura total de máximo 3 letras	
	Sí	No
12		x
14		x
16	x	
18	x	

RENDIMIENTO DE LOS ALGORITMOS OCR

Seguidamente se evaluó el rendimiento de cada una de las secciones que componen los algoritmos de reconocimiento de caracteres. El objetivo fue seleccionar el algoritmo que tuviera una menor carga computacional para la Raspberry durante su ejecución. Los algoritmos considerados fueron el de análisis matricial y el que hacía uso de la herramienta Tesseract.

Cuadro 6. Comparativa del tiempo de ejecución de los algoritmos de reconocimiento de caracteres.

ALGORITMO		TIEMPO PROMEDIO DE EJECUCIÓN (s)		
CÓDIGO	ÍTEM ANALIZADO	LED	LED + FLASH	FLASH
Tesseract	Importación de librerías	4.59417	3.9717	3.88951
	Creación de la instancia para la captura de video	0.00754	0.00786	0.00766
	Captura y procesamiento	6.736624	6.958498	8.473154
Procesamiento Matricial	Importación de librerías	3.68718	3.68923	3.73602
	Entrenamiento	0.18314	0.17715	0.22781
	Captura y procesamiento	0.8764602	0.811006	0.825852

Por último, se evaluó la eficacia con la que el algoritmo de OCR realiza el reconocimiento de los caracteres presentes en la imagen, para ello se tomaron múltiples imágenes de diferentes letras y se analizó el porcentaje de acierto con el que estas fueron reconocidas.

Además de lo anterior y pese a que el escenario de uso planteado para el dispositivo no los contempla, se consideraron casos en los que el área de captura de la cámara contenía de forma parcial o total más de un carácter, con el fin de evaluar los límites del algoritmo y su rendimiento en condiciones ajenas a su uso esperado.

Cuadro 7. Porcentaje de acierto del algoritmo de reconocimiento óptico

Imagen	letras en la imagen	¿Se reconoce texto?	Letras identificadas	Acierto (%)
0	Q	Sí	Q	100
1	W	Sí	W	100
2	E	Sí	E	100
3	R	Sí	R	100
4	T	Sí	T	100
5	Y	Sí	Y	100
6	U I	Sí	U I	100
7	I	Sí	I	100
8	O	Sí	O	100
9	P	Sí	P	100
10	A	Sí	A	100
11	S	Sí	S	100
12	D	Sí	D	100
13	F G/2	No	E G	50
14	G	Sí	G	100
15	<u>H</u>	No	Ñ	0
16	J	Sí	J	100
17	K	Sí	K	100
18	L	No	E	0
19	Ñ	Sí	Ñ	100
20	Z	Sí	Z	100
21	X	Sí	X	100

22	C	Sí	G	0
23	V	Sí	V	100
24	B	Sí	B	100
25	N	Sí	Ñ	0
26	M	Sí	M	100
27	E R	No	O R	50
28	R T	Sí	R T	100
29	T Y	Sí	T Y	100
30	Y U	Sí	Y U	100
31	U I	Sí	U I	100
32	S D	Sí	S D	100
PROMEDIO DE ACIERTO (%)				84.85

PRUEBA INTEGRAL DEL DISPOSITIVO

Se hicieron pruebas para medir los tiempos de respuesta del dispositivo en segundos, tanto en el modo analógico (usando la cámara), como en el modo digital (usando la aplicación). Para hacer estas pruebas se enviaron de forma individual los caracteres “g”, “u”, “r”, “e”, “n” y se midió el tiempo en las diferentes etapas que comprende su representación, desde el envío de la señal, hasta la vibración final en la celda braille dinámica.

A continuación, se presenta el diagrama de las etapas del modo digital acompañado de los resultados de la prueba

Figura 36. Diagrama de bloques de las etapas del modo digital

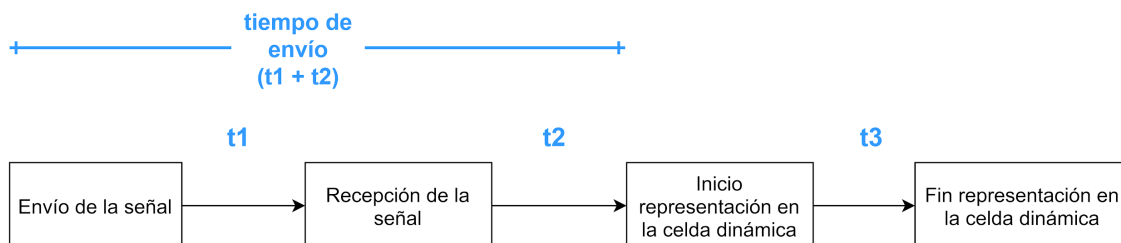
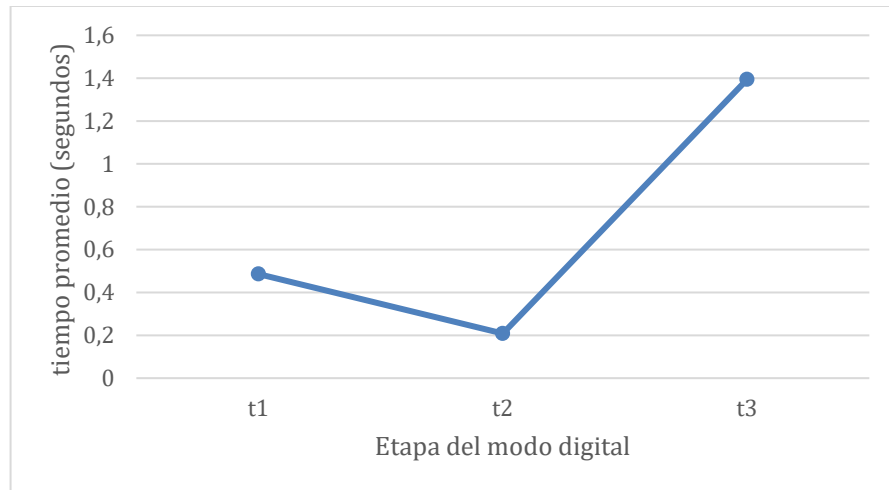


Figura 37. Diagrama de líneas del tiempo promedio de las etapas del modo digital



Cuadro 8. Tiempo de respuesta en segundos en el modo digital

Carácter	t1	t2	t3	Tiempo de envío	Tiempo total
g	0.541966	0.199032	1.41128	0.740998	2.152278
u	0.457655	0.19792	1.396575	0.655575	2.05215
r	0.496777	0.256494	1.384347	0.753271	2.137618
e	0.418771	0.201851	1.395005	0.620622	2.015627
n	0.514559	0.186338	1.384849	0.700897	2.085746
Promedio	0.4859456	0.208327	1.3944112	0.6942726	2.0886838

A continuación, se presenta en la Figura 38 el diagrama de las etapas del modo digital acompañado de los resultados de la prueba

Figura 38. Diagrama de bloques de las etapas del modo analógico

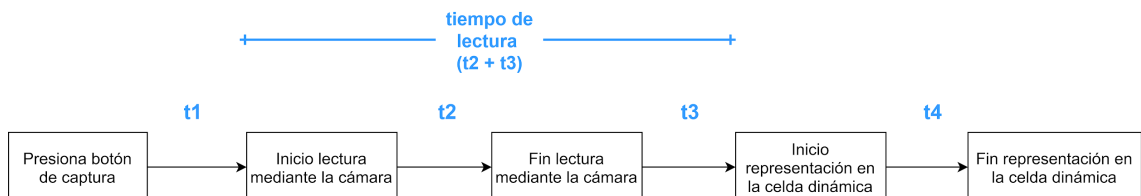
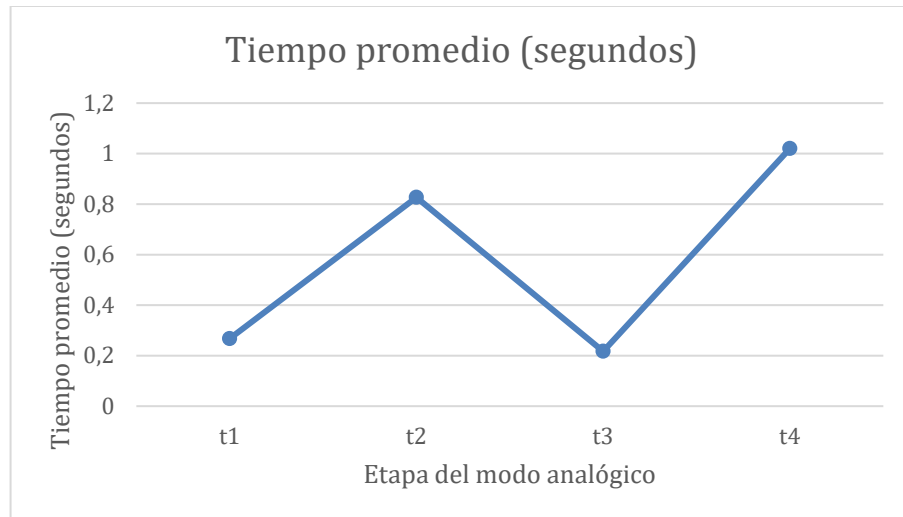


Figura 39. Diagrama de líneas del tiempo promedio de las etapas del modo analógico



Cuadro 9. Tiempo de respuesta en segundos en el modo analógico

Carácter	t1	t2	t3	t4	Tiempo de lectura	tiempo total
g	0.252599	0.825258	0.196915	1.017222	1.022173	2.291994
u	0.243439	0.827103	0.269874	1.016649	1.096977	2.357065
r	0.195844	0.783877	0.178133	1.034226	0.96201	2.19208
e	0.409255	0.815391	0.259383	1.016659	1.074774	2.500688
n	0.236506	0.88151	0.181	1.016569	1.06251	2.315585
Promedio	0.267528	0.826627	0.217061	1.020265	1.043688	2.331482

Los tiempos de encendido y apagado se midieron encendiendo y apagando el dispositivo 5 veces mientras se usaba un cronómetro. El tiempo de encendido se midió desde el momento en que fue presionado el botón de encendido, hasta el momento en que se activa la función saludo(), lo que garantiza medir no solo el tiempo que tarda el sistema operativo de la Raspberry (Raspbian) en iniciar, sino también cuánto tarda en cargar el firmware de Rillo.

El tiempo de apagado se midió desde el momento en que se envía el comando de apagado mediante la terminal de comandos (*sudo shutdown -h now*), hasta el momento en que el LED de indicación de la Raspberry se apaga.

Cuadro 10. Tiempo de encendido

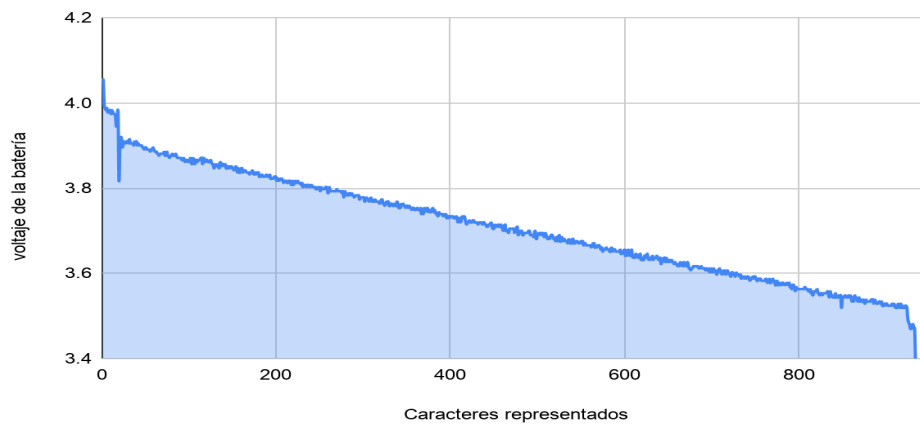
Numero de intento	tiempo de encendido (s)
1	74.29
2	74.83
3	77.3
4	74.76
5	74.14

Cuadro 11. Tiempo de apagado

Numero de intento	tiempo de apagado (s)
1	12.53
2	12.21
3	12.24
4	12.23
5	12.01

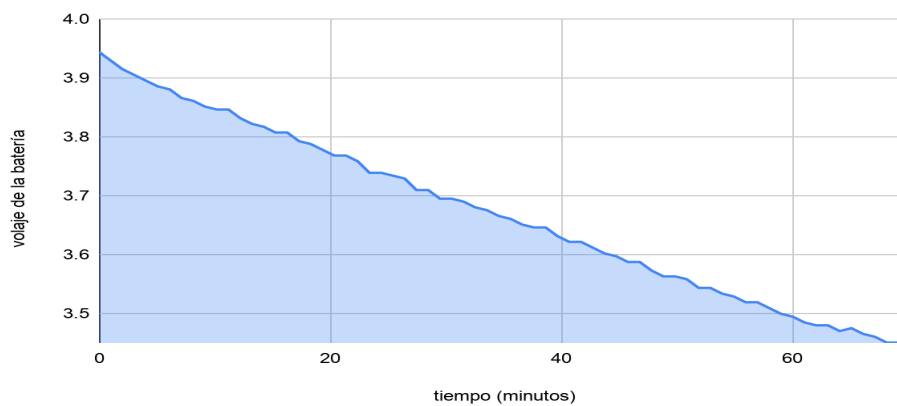
Para esta prueba se representaron letras extraídas de un texto generado aleatoriamente. Estas letras se activaron durante 3 segundos, con 1 segundo de espera entre cada una de ellas, y se configuró el equipo para tomar una foto con cada de ejecución, esto con el fin de simular de la forma más fehaciente posible el uso esperado del dispositivo.

Figura 40. Cantidad de caracteres representados en una carga completa de la batería



En el segundo escenario se simuló un uso intensivo, en el que todos los puntos de la celda se activaron al mismo tiempo durante 1 minuto, después de lo cual se toma una foto y se repite el ciclo de forma indefinida. El propósito principal de la prueba fue evaluar de la forma más exigente posible la capacidad de la batería utilizada.

Figura 41. Tiempo de descarga con un uso intensivo del dispositivo



9 ANÁLISIS DE RESULTADOS

9.1 ANÁLISIS DEL DISPOSITIVO

La PCB diseñada tiene un alto nivel de replicabilidad debido a su baja cantidad de componentes, además la disposición de la regleta tipo hembra permite una conexión sencilla tipo *shield* con la Raspberry Pi Zero, este diseño puede ser mejorado usando componentes de superficie para disminuir su tamaño, precio y portabilidad.

El modelo final del dispositivo es ambidiestro, permite un agarre ergonómico y su diseño posee una estructura modular que facilita la integración de los componentes internos. Los botones de interacción cuentan con puntos en relieve para su identificación mediante el tacto y cuenta con indicadores lumínicos que facilitan el acompañamiento por parte de tutores. La batería y el módulo de carga dotan de portabilidad al dispositivo y permiten que este pueda ser cargado usando un conector micro-usb.

9.2 ANÁLISIS DE LA PRUEBA DE LA CELDA BRAILLE DINÁMICA

El objetivo de esta prueba fue verificar la capacidad del dispositivo para representar de forma legible combinaciones de puntos del sistema braille. Es importante resaltar que esta prueba fue aplicada a personas videntes, sin conocimiento previo del sistema braille.

Los resultados obtenidos para el porcentaje de acierto son destacables, teniendo en promedio un 81% de acierto en la etapa 3, considerada la etapa más difícil, y llegando a un 95% de acierto en la etapa 1, la más sencilla, estos resultados avalan la hipótesis de que el dispositivo es capaz de representar de forma legible los puntos de la celda braille, incluso considerando el escaso entrenamiento que tenían los participantes y su nulo conocimiento del sistema braille.

El siguiente factor por resaltar es el bajo número de intentos que tomaron los participantes para dar una respuesta, dado que todos en promedio requirieron la ejecución del estímulo solo entre 1 y 2 veces, a pesar de que podían solicitar que se activara de nuevo las veces que quisieran.

Finalmente tenemos el tiempo de lectura, es decir, el tiempo que tardaron los participantes entre la activación del estímulo y la respuesta final. Este tiempo es bastante alto, llegando casi a los 5 segundos de lectura por estímulo en la primera etapa, suponemos que esto se debe a la falta de experiencia de los participantes con el dispositivo y que es un factor que puede mejorarse a medida que la persona se familiarice más con este tipo de estímulos, no obstante, esta es una hipótesis que deberá validarse en un estudio posterior.

9.3 ANÁLISIS DE LA PRUEBA REALIZADA CON LA CÁMARA

Los resultados obtenidos muestran que, al ajustar manualmente el lente a su mínima capacidad de enfoque con la herramienta correspondiente, la distancia focal tiene una relación directamente proporcional con la calidad de la imagen entre 1.2 cm y 1.6 cm, siendo este último valor el punto máximo a partir del cual la calidad de la imagen entra en detrimento y la relación entre los dos parámetros se hace inversamente proporcional.

Para la selección de la fuente se analizaron los resultados obtenidos de la captura de múltiples imágenes del texto impreso bajo una distancia focal constante de 1.6 cm, de acuerdo con los valores mínimos obtenidos en la prueba anterior. Con esta prueba se buscaba identificar el tamaño de fuente con la cual el área de captura de la imagen abarcara como máximo 3 letras totalmente visibles. Para la prueba se utilizó la fuente Arial, debido a que es considerada una de las fuentes de mayor uso. Los resultados muestran que tanto el tamaño de fuente 16 como 18 cumplen con las expectativas, por lo que cualquiera de los dos sería una selección pertinente, no obstante, el uso del equipo está pensado para tamaño de fuente 18.

9.4 ANÁLISIS DE LA PRUEBA DE RENDIMIENTO DE LOS ALGORITMOS OCR

El objetivo de la prueba inicial fue seleccionar el algoritmo a usar con el dispositivo final, debido a que durante el desarrollo del estado del arte se optó por considerar también como opción *Tesseract*, herramienta ampliamente aceptada a nivel mundial y que se ajustaba al escenario inicialmente planteado para el dispositivo, gracias a los numerosos *wrappers* que se encuentran disponibles para python, el lenguaje escogido para el algoritmo del proyecto. La carga computacional que implica Tesseract en comparación con el procesamiento a través de análisis matricial era una preocupación. *Tesseract* usa para el reconocimiento una red neuronal tipo LSTM (Long Short Term Memory), por lo que se esperaba una mayor carga computacional, lo cual pudo comprobarse con los resultados obtenidos y mostrados en la [sección 7.3](#), donde los tiempos de la captura y procesamiento de cada imagen con Tesseract fueron 7 veces superiores a los tiempos de procesamiento a través de análisis matricial para la mayor parte de las imágenes analizadas, por lo que su uso para este dispositivo fue descartado.

Posteriormente y con el fin de evaluar el rendimiento y/o confiabilidad del algoritmo diseñado, se realizaron diferentes pruebas bajo escenarios incluso más complejos de lo planteado para el uso del dispositivo final. Se realizaron capturas a texto donde se incluían tanto una única letra en toda el área de captura, como múltiples letras, así como una combinación entre letras total y parcialmente visibles. Los resultados muestran un porcentaje de acierto promedio del 84.85 %. Los valores mínimos de acierto obtenidos corresponden en su mayoría a lo inicialmente esperado en la evaluación de las letras "N", "C" y "L", donde existen similitudes claras en algunas secciones de los contornos identificados en las letras como "Ñ", "G" y "E" respectivamente.

Para la identificación de más de una letra se obtuvieron resultados óptimos, mostrando que en el 91.66% de los casos en los que se capturaron 2 letras consecutivas, pudo realizarse el correcto reconocimiento de cada letra, caso contrario a lo visto para las imágenes que incluían letras total y parcialmente definidas, donde los resultados muestran que la identificación de las letras parcialmente definidas se realiza de forma errónea.

9.5 ANÁLISIS DE LA PRUEBA INTEGRAL DEL DISPOSITIVO

Los tiempos de respuesta, tanto en el modo analógico como en el digital son bastante bajos, lo cual brinda una experiencia óptima para el usuario. Por otro lado, pese a que los tiempos de encendido son altos y los de apagado son bajos, esto no obstruye la experiencia de uso, dado que el dispositivo no requiere ser reiniciado durante su uso habitual y además porque cuenta con la función saludo(), diseñada como indicador para que el usuario reconozca fácilmente cuando el dispositivo esté listo para ser usado.

La diferencia en el tiempo de descarga entre el uso esperado y el uso intensivo no es tan amplia como se esperaba, nuestras observaciones apuntan a que se requiere una mayor cantidad de energía para activar cada punto que la requerida para mantenerlo activo, por ende, aunque se activan menos puntos en el uso esperado, la activación y desactivación constante demanda mayor energía.

10 CONCLUSIONES

- El dispositivo es capaz de representar caracteres braille de forma confiable y legible.
- El dispositivo junto a la aplicación web, pueden servir como material didáctico para enseñar el sistema Braille.
- El dispositivo es portable y puede ser usado por personas diestras o zurdas.
- Una celda braille con puntos de un diámetro superior a 8 mm es ilegible.
- Es necesario realizar un entrenamiento previo al uso del dispositivo, para garantizar que la persona se familiarice con el dispositivo y con los estímulos que este provee.
- Tener un correcto posicionamiento del dedo en la celda braille mejora considerablemente la experiencia de uso del dispositivo y la legibilidad de los caracteres en él representados.
- Hacer un diseño de PCB tipo shield facilita considerablemente la conexión con la Raspberry y la replicabilidad del dispositivo.
- Hacer uso de plataformas como Firebase reducen los tiempos de implementación y conexión entre una página web y una placa de desarrollo como la Raspberry.
- Ajustar de forma manual el enfoque de la cámara dotó al proyecto de flexibilidad al permitirnos variaciones de la distancia entre el lente y la hoja con el texto impreso, lo que fue de gran utilidad durante la consolidación del modelo y la disposición de cada componente en el diseño final.
- Durante las pruebas realizadas con la cámara fue posible evidenciar que, si bien la cámara ofrece grandes prestaciones, la calidad de la imagen es afectada enormemente cuando se realizan capturas en movimiento, por lo que fue necesario definir criterios de uso que garantizaran que al momento de hacer una captura, la cámara estuviera lo más estática posible.
- La selección de la fuente es un punto crucial durante el uso del algoritmo, debido a que el rendimiento de este, considerando el enfoque y tipo de análisis realizado, puede verse afectado con fuentes que poseen grandes similitudes entre letras, como consecuencia de la robustez reducida que el diseño ofrece.
- La iluminación de la imagen es un parámetro importante durante el procesamiento de cada captura, en especial para la umbralización, donde se contrastan directamente las porciones oscuras de las iluminadas con el fin de facilitar la identificación de contornos.
- Los tiempos de respuesta del dispositivo son bajos, esto facilita la interacción dinámica con el dispositivo, haciéndolo propicio para el uso en ambientes de aprendizaje de braille.
- El dispositivo tiene una autonomía de aproximadamente una hora de uso continuo y se descarga de forma lineal, lo que permite su uso de forma confiable en sesiones de lectura.

11 RECOMENDACIONES

- **SISTEMA DE AUTENTICACIÓN:** Desarrollar un sistema de inicio de sesión que permita a los usuarios conectarse específicamente a su dispositivo con su respectivo usuario y contraseña y teniendo la posibilidad de vincular dispositivos.
- **MATERIAL DIDACTICO:** Diseñar material didáctico que pueda ser usado con y sin la cámara del dispositivo para enseñar o practicar el sistema Braille.
- **SISTEMA DE SEÑALIZACIÓN:** Desarrollar un sistema de señalización braille que le permita a las personas invidentes ubicarse y moverse dentro de un párrafo sin necesidad de elementos externos o ayuda de terceros.
- **OPTIMIZACIÓN DE MODELO 3D:** Para desarrollos futuros se recomienda considerar dentro del modelo 3D a usar, puntos de anclaje y permitan asegurar cada una de las piezas del equipo al modelo, de manera tal que no existan riesgos de daño por uso normal como consecuencia del movimiento interno de los componentes. Además de esto, la estructura modular del modelo puede ser mejorada para facilitar la integración final de los componentes y con esto, facilitar la replicabilidad del diseño, lo que a su vez incrementará el alcance/impacto del dispositivo gracias a que podrá estar al alcance de una mayor cantidad de personas invidentes.

BIBLIOGRAFÍA

“AB-028 : Vibration Motor Comparison Guide - Precision Microdrives”. Consultado el 30 de mayo de 2020. <https://www.precisionmicrodrives.com/content/ab-028-vibration-motor-comparison-guide/>.

“About”, OpenCV documentation, consultado el 2 de abril de 2021, <https://opencv.org/about/>

CHINA YOUNG SUN LED TECHNOLOGY CO., LTD. “RED/GREEN/BLUE Triple Color LED Model No.: YSL-R596CR3G4B5C-C10 Datasheet”. Sparkfun. Consultado el 31 de mayo de 2020. <https://www.sparkfun.com/datasheets/Components/YSL-R596CR3G4B5C-C10.pdf>.

“Coin Vibration Motors - Precision Microdrives”. Consultado el 30 de mayo de 2020. <https://www.precisionmicrodrives.com/vibration-motors/coin-vibration-motors/>.

Comisión Braille Española. Organización Nacional de Ciegos Españoles (ONCE). *GUÍAS DE LA COMISIÓN BRAILLE ESPAÑOLA, Signografía básica*. Primera Edición. Madrid: Artes Gráficas Palermo, S.L., 2005. <https://sid.usal.es/idos/F8/FDO12069/signografiabasica.pdf>.

Control Dinamico SAS. “Bateria Polimero De Litio 3.7v, 2000mah 103450 Lipo - \$ 27.800”. MercadoLibre. Consultado el 4 de junio de 2020. https://articulo.mercadolibre.com.co/MCO-485643308-bateria-polimero-de-litio-37v-2000mah-103450-lipo-_JM.

Fairchild Semiconductor International, Inc. “DM74LS38 Quad 2-Input NAND Buffer with Open-Collector Outputs Datasheet”, mayo de 2000. <https://www.futurlec.com/Datasheet/74ls/74LS38.pdf>.

“Firebase”. En *Wikipedia*, el 13 de mayo de 2020. <https://en.wikipedia.org/w/index.php?title=Firebase&oldid=956484625>.

Firestore Documentation. “Firestore Realtime Database”. Consultado el 6 de junio de 2020. <https://firebase.google.com/docs/database>.

Industries, Adafruit. “PowerBoost 1000 Charger - Rechargeable 5V Lipo USB Boost @ 1A”. Consultado el 4 de junio de 2020. <https://www.adafruit.com/product/2465>.

Jinlong Machinery & Electronics. “Coin Vibration Motor Part No. C1026B002F Datasheet”, el 11 de septiembre de 2009. <https://www.parallax.com/sites/default/files/downloads/28821-Flat-Coin-Vibration-Motor-Documentation.pdf>.

Microchip Technology Inc. “MCP3004/3008 2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface Datasheet”. Microchip Technology Inc., 2008. <http://ww1.microchip.com/downloads/en/DeviceDoc/21295d.pdf>.

NumPy documentation, consultado el 2 de abril de 2021, <https://numpy.org/doc/stable/>

Profesional Review. “▷ Qué es una PCB o Placa de Circuito Impreso. Uso, cómo se fabrica”, el 11 de febrero de 2019. <https://www.profesionalreview.com/2019/02/11/pcb-que-es/>.

“Digitization”. En *Wikipedia*, el 30 de abril de 2021.
<https://en.wikipedia.org/w/index.php?title=Digitization&oldid=1020616029>.

“Edge Detection”. En *Wikipedia*, el 30 de abril de 2021.
https://en.wikipedia.org/w/index.php?title=Edge_detection&oldid=1020616177.

“k-nearest neighbors algorithm - Wikipedia”. Consultado el 4 de mayo de 2021.
https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

KUMAR, BARATH. “Image Preprocessing - Why Is It Necessary?” Medium, el 10 de febrero de 2021. <https://medium.com/spidernitt/image-preprocessing-why-is-it-necessary-8895b8b08c1d>.

“Optical Character Recognition”. En *Wikipedia*, el 7 de abril de 2021.
https://en.wikipedia.org/w/index.php?title=Optical_character_recognition&oldid=1016528333.

Puertas, A, P Purés, A M Echenique, y J P Graffigna y G Ensinck. “Braille Line Using Electrical Stimulation”. *Journal of Physics: Conference Series* 90 (el 1 de noviembre de 2007): 012091. <https://doi.org/10.1088/1742-6596/90/1/012091>.

“RPi Low-Level Peripherals”. En *ElLinux.Org*, el 28 de junio de 2019.
https://elinux.org/RPi_Low-level_peripherals#P6_header.

Shperber, Gidi. “A Gentle Introduction to OCR”. Medium, el 15 de febrero de 2021.
<https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa>.

“Tactile (Device)”. En *Wikipedia*, el 12 de diciembre de 2019.
[https://en.wikipedia.org/w/index.php?title=Tactile_\(device\)&oldid=930397893](https://en.wikipedia.org/w/index.php?title=Tactile_(device)&oldid=930397893).

Tatoko. “Tatoko 20PCS 10mmx3mm Mini Vibration Motors DC 3V Flat Coin Button-Type”. Amazon, el 24 de marzo de 2019. <https://www.amazon.com/tatoko-Vibration-Button-Type-Vibrating-Appliances/dp/B07Q1ZV4MJ>.

The Raspberry Pi Foundation. “Buy a Raspberry Pi Zero W – Raspberry Pi”. Consultado el 31 de mayo de 2020. <https://www.raspberrypi.org/products/raspberrypi-zero-w/>.

“Thresholding”, en *Wikipedia*, consultado el 3 de abril de 2021,
[https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))

UK Association for Accessible Formats (UKAAF). “Standard Dimensions for the UK Braille Cell”. UK Association for Accessible Formats (UKAAF), abril de 2017.
<https://www.ukaaf.org/wp-content/uploads/BrailleStandardDimensionsFinal.pdf>.

UK, Gareth Halfacree from Bradford. *The wireless variant of the Raspberry Pi Zero microcomputer, known as the Raspberry Pi Zero W*. el 2 de marzo de 2017. Raspberry

Pi Zero W.

[https://commons.wikimedia.org/wiki/File:Raspberry_Pi_Zero_W_\(33209067455\).png](https://commons.wikimedia.org/wiki/File:Raspberry_Pi_Zero_W_(33209067455).png).

Kitronik Ltd. "White 5mm Water Clear LED - 25deg - 2250mCd". Consultado el 3 de junio de 2020. <https://kitronik.co.uk/products/35109-white-5mm-water-clear-led-25deg-2250mcd>.

World Health Organization, International Agency for the Prevention of Blindness.

VISION 2020 THE RIGHT TO SIGHT - Global Initiative for the Elimination of Avoidable Blindness: : Action Plan 2006-2011. France: World Health Organization, 2007.

https://www.who.int/blindness/Vision2020_report.pdf.

ANEXOS

Anexo A. Circuito esquemático y diseño de la PCB del dispositivo.

Figura 42. Circuito esquemático del dispositivo

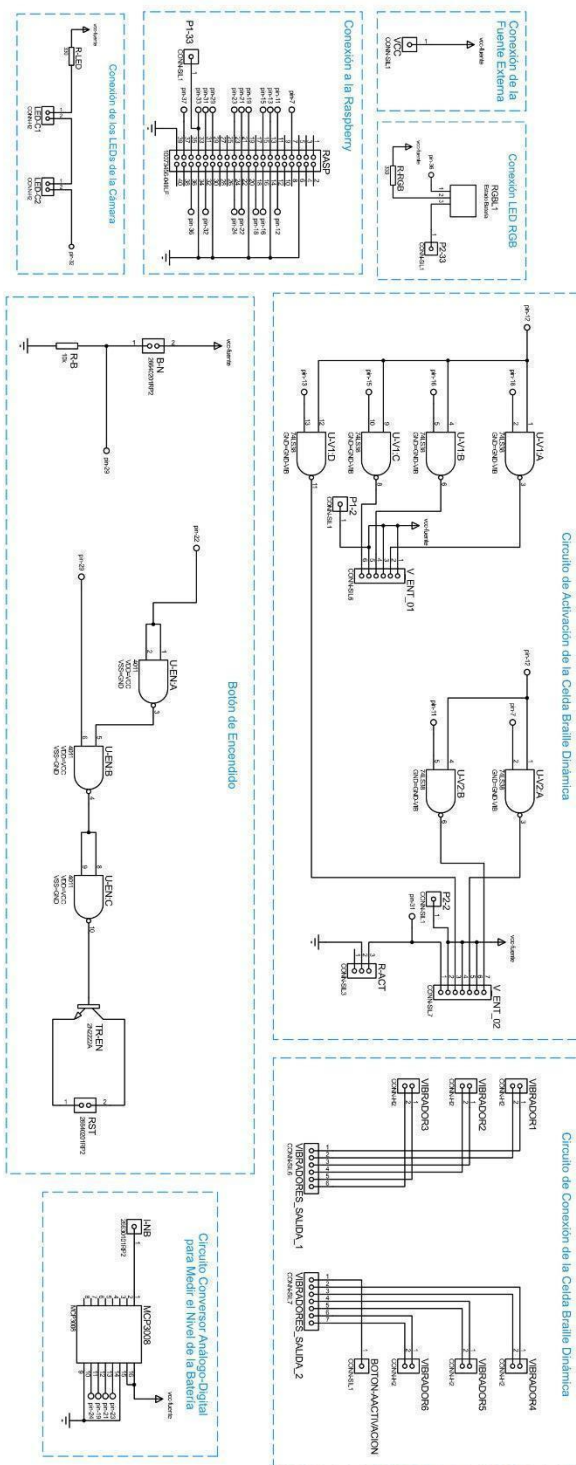


Figura 43. Visualización 3D de la PCB del dispositivo



Figura 45. Vista superior del diseño de la PCB del dispositivo

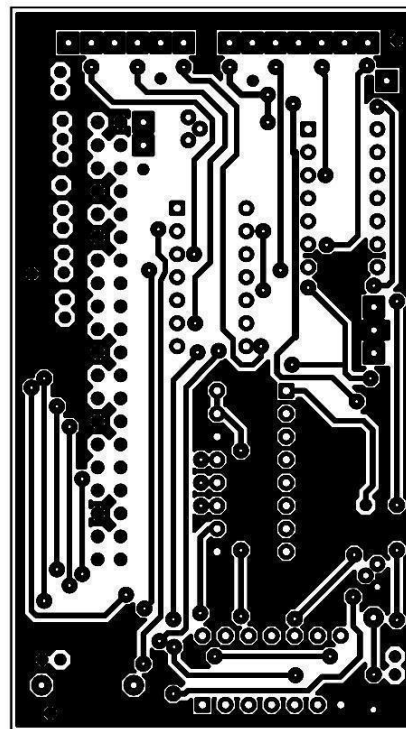


Figura 44. Vista inferior del diseño de la PCB del dispositivo

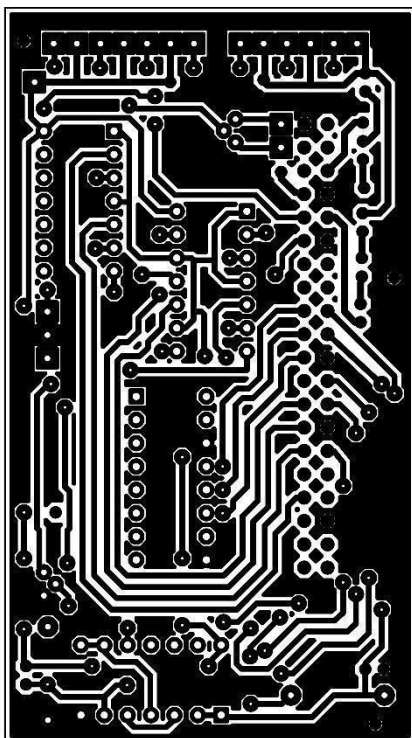
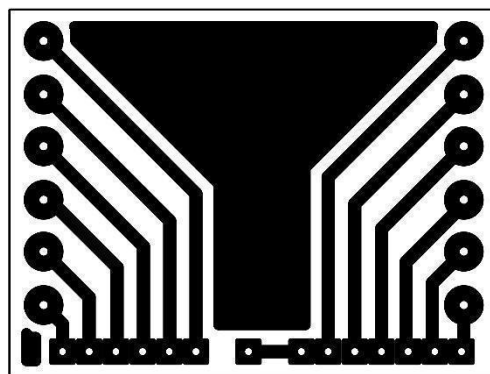


Figura 46. Vista inferior del diseño de la PCB de la celda braille dinámica



Anexo B. Código del software del dispositivo.

Software del dispositivo.py

```
import os
os.system("sudo pigpiod")

import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore
import pigpio as GPIO
import spidev
import csv
import time
import threading
import brailfun

# Use a service account
cred = credentials.Certificate(r'/home/pi/rillo-web-firebase-adminsdk-n5611-e48238a56c.json')
firebase_admin.initialize_app(cred)

db = firestore.client()
brf = brailfun.new_cell(5, 3, 1, 1)
pi = brf.pi
spi = spidev.SpiDev()
spi.open(0,0)

#Software Rillo v1.1 (08/03/2020)
#Testeado en la version 3.7.3

print("inicializando")

# Variables

nivel_bateria = 0
t_sleep = 0.5*60          # 5 minutos de inactividad para entrar al modo sleep
t_shutdown = 1*60         # 30 minutos en modo sleep para apagarse
t_activated = time.time()
t_lectura_bateria = 60    # Se mide el nivel de batería cada 60 segundos
canal_input_bateria = 0   # canal del MCP3008 en el cual ingresa la señal del nivel de batería
, 0 por defecto
modo_actual = "analogico" # Modo en el que se encuentra rillo (puede ser digital, analogico o
sleep)
stop_parpadeo_led = False # Esta variable nos indica si los leds deben dejar de parpadear (sol
o parpadean cuando rillo tiene la batería baja)

#Leer variables de vibración del csv
with open('variables_rillo.csv') as csvDataFile:
    csvReader = csv.reader(csvDataFile)
    for row in csvReader:

        if row[0] == 'intensidad':
            brf.power = float(row[1])
        elif row[0] == 'tiempo_on':
            brf.time_on = float(row[1])
        elif row[0] == 'tiempo_off':
            brf.time_off = float(row[1])
        elif row[0] == 'signal':
            brf.signal_type = int(row[1])

#Pines

pin_indicador_encendido = 25      #pin 22
pin_boton_activacion = 6          #pin 31
pin_boton_bateria = 5             #pin 29
```

```

pin_led_rgb_g = 16      #pin 36
pin_led_rgb_b = 13      #pin 33
pin_leds_camara = 26     #pin 37

pi.set_mode(pin_indicador_encendido, GPIO.OUTPUT)
pi.set_mode(pin_boton_activacion, GPIO.INPUT)
pi.set_mode(pin_boton_bateria, GPIO.INPUT)
pi.set_mode(pin_led_rgb_g, GPIO.OUTPUT)
pi.set_mode(pin_led_rgb_b, GPIO.OUTPUT)
pi.set_mode(pin_leds_camara, GPIO.OUTPUT)

pi.set_glitch_filter(pin_boton_activacion, 500)
pi.set_glitch_filter(pin_boton_bateria, 500)

pi.write(pin_indicador_encendido, 1)
pi.write(pin_led_rgb_g, 1)
pi.write(pin_led_rgb_b, 1)
pi.write(pin_leds_camara, 1)

#Excepcion

class apagadox(Exception): pass

# funciones

def saludo():
    print("saludo")

    r_braille = [1, 2, 3, 5]

    for punto in r_braille:
        celda_braille = [0, 0, 0, 0, 0, 0]
        celda_braille[punto] = 1
        brf.trigger(celda_braille)

def apagado_automatiko():
    print("rillo out")
    pi.stop()
    spi.close()
    os.system('sudo killall pigpiod')
    os.system('sudo shutdown -h now')

def potencia_leds_camara(potencia):
    """ Esta función sirve para cambiar la potencia de la salida pwm de los leds de la camara
    potencia_pwn, es un entero que recibe un valor entre 0 y 255 """

    global pin_leds_camara

    pi.write(pin_leds_camara, potencia)

def parpadeo_leds():
    """Esta función está hecha para ser llamada por un thread de forma tal que el led de activac
    ion parpadee cuando la batería tenga un nivel de carga bajo"""

    global modo_actual, stop_parpadeo_led

    cambio_luz = True

    while True:
        if modo_actual == "analogico":
            led_activacion("verde")

        elif modo_actual == "digital":
            led_activacion("azul")

        elif modo_actual == "sleep":
            cambio_luz = not(cambio_luz)

        if cambio_luz:

```

```

        led_activacion("verde")
    else:
        led_activacion("azul")

    else:
        print("error modo_actual esta mal definido en la funcion parpadeo_leds, se esperan 1
os valores analogico o digital")

    if modo_actual == "sleep":
        time.sleep(1)
        led_activacion("off")
        time.sleep(2)
    else:
        time.sleep(0.5)
        led_activacion("off")
        time.sleep(0.5)

    if stop_parpadeo_led:
        if modo_actual == "analogico":
            led_activacion("verde")

        elif modo_actual == "digital":
            led_activacion("azul")

        elif modo_actual == "sleep":
            led_activacion("off")

    else:
        print("error modo_actual esta mal definido en la funcion parpadeo_leds, se esperan
los valores analogico o digital")
        break

def medir_bateria():
    """ Esta función mide el nivel de voltaje de la bateria, si este esta por debajo del nivel i
ndicado hace parpadear el led de indicacion"""
    print("funcion medir bateria")

global nivel_bateria, canal_input_bateria, stop_parpadeo_led

spi.max_speed_hz = 1350000
adc = spi.xfer2([1, (8+canal_input_bateria)<<4,0])
data = ((adc[1]&3) << 8) + adc[2]

nivel_bateria = data*5.0/1023

try:
    doc_ref = db.collection(u'rillo-main').document(u'nivel-bateria')
    doc_ref.update({
        u'recibido': bateria
    })
except:
    print("error, no puede acceder a la base de datos al enviar el nivel de la bateria")

if nivel_bateria >= 3.4:

    stop_parpadeo_led = True

    try:
        thread_parpadeo.join()
    except:
        print("thread_parpadeo aun no ha sido creado")

elif nivel_bateria < 3.4:

    stop_parpadeo_led = False
    thread_parpadeo = threading.Thread(target=parpadeo_leds)

    try:

```

```

        thread_parpadeo.start()
    except:
        print("Error, tratando de iniciar thread_parpadeo de nuevo, este error no crea conflictos")

    elif nivel_bateria <= 3.2:

        raise apagadox

    else:
        print("error, led bateria, nivel medido no identificado")

    return nivel_bateria

def modo_activo(gpio, level, tick):
    print("modo activacion")

    global t_activated, modo_actual, interrupcion_activo, stop_parpadeo_led

    stop_parpadeo_led = True

    try:
        thread_parpadeo.join()
    except:
        print("thread_parpadeo aun no ha sido creado")

    t_activated = time.time()
    modo_actual = "analogico"
    interrupcion_activo.cancel()
    del(interrupcion_activo)

def modo_analogico():
    #En este modo Rillo recibe y procesa datos desde la camara
    print("modo analogico")

    #encender camara %?
    print("camara encendida")

    try:
        doc_ref = db.collection(u'rillo-main').document(u'funciones')
        doc_ref.update({
            u'conectado': False
        })
    except:
        print("error, no puede acceder a la base de datos al tratar de escribir el nivel de bateria estando en el modo digital")

    global t_sleep, t_lectura_bateria, modo_actual

    modo_actual = "analogico"

    potencia_leds_camara(0)
    brf.trigger([1,0,0,0,0,0])
    led_activacion('verde')

    t_ini_ejecucion = time.time()
    t_ini_bateria = t_ini_ejecucion
    t_bateria = 0
    t_ejecucion = 0

    while t_ejecucion <= t_sleep:

        try:
            doc_ref = db.collection(u'rillo-main').document(u'funciones').get()
            funciones = doc_ref.to_dict()

            if funciones['recibido']:
                pass
            else:
                modo_digital()

```

```

except:
    print("error, no se puede acceder a la base de datos en el modo analogico")

#Lectura camara %?
#print("leyendo camara, return 2 variables, [movimiento (boolean), letra (String)")
#lectura_camara = detectar_letra() %?
lectura_camara = [1, "a"]

try:
    pass
    brf.writer(lectura_camara[1])
    print("verde!")

except:
    pass

if lectura_camara[0]:
    t_ini_ejecucion = time.time()

#Revisar si en la base de datos hubo un cambio, si lo hubo entrar al modo digital

if t_bateria >= t_lectura_bateria:

    t_ini_bateria = time.time()
    medir_bateria()

    t_ejecucion = time.time() - t_ini_ejecucion
    t_bateria = time.time() - t_ini_bateria

modo_sleep()

def modo_digital():
    print("modo digital")
    #En este modo Rillo recibe y envia datos a un servidor de firebase que se comunica con la ap
    p

    #apagar camara %?
    print("camara apagada")

    try:
        doc_ref = db.collection(u'rillo-main').document(u'funciones')
        doc_ref.update({
            u'conectado': True
        })
    except:
        print("error, no puede acceder a la base de datos al tratar de escribir el nivel de bate
ria estando en el modo digital")

    global t_sleep, modo_actual

    modo_actual = "digital"

    potencia_leds_camara(1)
    brf.trigger([0,0,0,0,0,1])
    led_activacion('azul')

    t_ini_ejecucion = time.time()
    t_ini_bateria = t_ini_ejecucion
    t_bateria = 0
    t_ejecucion = 0

    while t_ejecucion <= t_sleep:

        try:
            doc_ref = db.collection(u'rillo-main').document(u'funciones').get()

```

```

        funciones = doc_ref.to_dict()
    except:
        print("error, no se puede acceder a la base de datos en el modo digital")

    if funciones['recibido']:
        pass
    else:
        representar_datos(funciones['funcion'], funciones['dato'])

    if funciones['conectado']:
        pass
    else:
        modo_analogico()

    if t_bateria >= t_lectura_bateria:

        try:
            doc_ref = db.collection(u'rillo-main').document(u'nivel-bateria')
            doc_ref.update({
                u'bateria': nivel_bateria
            })
        except:
            print("error, no puede acceder a la base de datos al tratar de escribir el nivel
de bateria estando en el modo digital")

        t_ini_bateria = time.time()

        t_ejecucion = time.time() - t_ini_ejecucion
        t_bateria = time.time() - t_ini_bateria

    modo_sleep()

def modo_sleep():
    """ En este modo se apaga la camara, los leds de la camara y """
    print("modo sleep")
    # apagar_camara() %?
    print("camara apagada")

global t_lectura_bateria, t_shutdown, modo_actual, interrupcion_activo, stop_parpadeo_led

modo_actual = "sleep"
potencia_leds_camara(1)

stop_parpadeo_led = False
thread_parpadeo = threading.Thread(target=parpadeo_leds)

try:
    thread_parpadeo.start()
except:
    print("Error, tratando de iniciar thread_parpadeo de nuevo, este error no crea conflicto
s")

t_ejecucion = 0
t_ini = time.time()
t_ini_bateria = t_ini
t_bateria = 0

interrupcion_activo = pi.callback(pin_boton_activacion, 1, modo_activo)

while (t_ejecucion <= t_shutdown) and modo_actual == "sleep":

    try:
        doc_ref = db.collection(u'rillo-main').document(u'funciones').get()
        funciones = doc_ref.to_dict()

        if funciones['recibido']:
            pass

```

```

        else:

            stop_parpadeo_led = True

            try:
                thread_parpadeo.join()
            except:
                print("thread_parpadeo aun no ha sido creado")

            modo_digital()

    except:
        print("error, no se puede acceder a la base de datos en el modo sleep")

    if t_bateria >= t_lectura_bateria:

        try:
            doc_ref = db.collection(u'rillo-main').document(u'nivel-bateria')
            doc_ref.update({
                u'bateria': nivel_bateria
            })
        except:
            print("error, no puede acceder a la base de datos al tratar de escribir el nivel
de bateria estando en el modo digital")

            t_ini_bateria = time.time()

            t_ejecucion = time.time() - t_ini
            t_bateria = time.time() - t_ini_bateria

    if modo_actual == "sleep":
        raise apagadox

    elif modo_actual == "analogico":

        try:
            modo_analogico()

        except apagadox:
            apagado_automatico()

    else:
        print("Error, el modo actual es distinto de los valores digital, analogico o sleep en la
funcion modo_sleep()")

def led_activacion(color_led):
    # print("Led activacion")
    # Hay 4 colores verde, azul, cian y off

    if color_led == 'verde':
        pi.write(pin_led_rgb_g, 0)
        pi.write(pin_led_rgb_b, 1)

    elif color_led == 'azul':
        pi.write(pin_led_rgb_g, 1)
        pi.write(pin_led_rgb_b, 0)

    elif color_led == 'off':
        pi.write(pin_led_rgb_g, 1)
        pi.write(pin_led_rgb_b, 1)

    else:
        print("error: led_activacion, color led no identificado")

def boton_nivel_bateria(gpio, level, tick):

    """ Esta funcion es activada por una interrupcion dada por el boton del dispositivo, lo que
hace es representar el nivel de la batería
    en 3 estados, alto, medio o bajo, haciendo vibrar la celda braille"""

```

```

nivel_bateria = medir_bateria()

print(f"\nnivel de bateria: {nivel_bateria}\n")

niveles_bateria_braille = [[1, 0, 0, 1, 0, 0], [0, 1, 0, 0, 1, 0], [0, 0, 1, 0, 0, 1]]

if nivel_bateria >= 3.7:
    for n_bateria in niveles_bateria_braille:
        brf.trigger(n_bateria)

elif nivel_bateria >= 3.5:
    for n_bateria in range(0,2):
        brf.trigger(niveles_bateria_braille[n_bateria])

elif nivel_bateria >= 3.3:
    brf.trigger(niveles_bateria_braille[0])

else:
    print("error, boton nivel bateria, nivel medido no identificado")

def datos_recibidos():
    try:
        doc_ref = db.collection(u'rillo-main').document(u'funciones')
        doc_ref.update({
            u'recibido': 1
        })
    except:
        print("error, no puede acceder a la base de datos al confirmar que se recibió un dato")

def representar_datos(funcion, datos):
    print("representar datos")

    if funcion == 'escribir':
        brf.writer(datos)
        datos_recibidos()

    elif funcion == 'perfil_vibracion':
        brf.power = datos[0]
        brf.time_on = datos[1]
        brf.time_off = datos[2]
        brf.signal_type = datos[3]

        datos_csv = [['intensidad', brf.power], ['tiempo_on', brf.time_on], ['tiempo_off', brf.time_off], ['signal', brf.signal_type]]

        with open('variables_rillo.csv', 'w', newline='') as myFile:
            writer = csv.writer(myFile)
            writer.writerow(datos_csv)

        brf.trigger([0,0,0,0,0,1])

        datos_recibidos()

    elif funcion == 'generador':
        brf.generator()
        datos_recibidos()

    elif funcion == 'celda':
        brf.trigger(datos)
        datos_recibidos()

    elif funcion == "nope":
        datos_recibidos()

    else:
        print("error, funcion no identificada en la lectura de datos digitales")

```

```

#interrupciones

interrupcion_bateria = pi.callback(pin_boton_bateria, GPIO.RISING_EDGE, boton_nivel_bateria)

saludo()

try:
    modo_analogico()
except apagadox:
    apagado_automatico()

```

Anexo C. Código de la librería Brailfun.

Código librería brailfun.py

```

import pigpio as GPIO
import math
import time
import random
#import numpy as np
#from numpy import interp    # To scale values

# This code uses BCM pins
# Tested on python 3.5.3

class new_cell:

    # The default physical pins are 12, 7, 11, 13, 15, 16 and 18
    # The first value is the signal BCM pin (pwm) then the next values are the vibrator BCM pins
    # from 1 to 6

    vibration_pins = [18, 4, 17, 27, 22, 23, 24]

    #GPIO setup
    pi = GPIO.pi()

    for pin in vibration_pins:
        pi.set_mode(pin, GPIO.OUTPUT)
        pi.write(pin, 0)

    def __init__(self, power = 5, time_on = 3, time_off = 1, signal_type = 1):
        self.power = power
        self.time_on = time_on
        self.time_off = time_off
        self.signal_type = signal_type

    @classmethod
    def pinout(cls, signal_pin = vibration_pins[0], v1 = vibration_pins[1], v2 = vibration_pins[
2], v3 = vibration_pins[3], v4 = vibration_pins[4], v5 = vibration_pins[5], v6 = vibration_pins[
6]):
        """ This function prints the pinout and helps the user change the vibration pinout, the
first parameter is the signal pin (pwm) and the v1 to v6 parameters are the i/o pins for each vi
brator"""
        cls.vibration_pins = [signal_pin, v1, v2, v3, v4, v5, v6]

        for pin in cls.vibration_pins:
            new_cell.pi.set_mode(pin, GPIO.OUTPUT)
            new_cell.pi.write(pin, 0)

        print("\nPinout\nSignal pin: {}\nVibrator pins: {}".format(cls.vibration_pins[0], cls.vi
bration_pins[1:]))

    @staticmethod

```

```

def clamp(value):
    """ this function limitates the value between 0 and 255"""

    if(value>255):
        return 255
    elif(value<0):
        return 0
    else:
        return value

@classmethod
def close(cls):
    cls.pi.stop()

def s_square(self):
    #square signal

    output=255*(self.power/5.0)
    new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0], output)

    time.sleep(self.time_on)

    output=0
    new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0], output)

    time.sleep(self.time_off)

#señal triangular
def s_triad(self):

    t_ini = time.time()

    # In this cycle the vibration signal is calculated and activated through a pwm signal

    while time.time() <= (t_ini + self.time_on):

        if time.time() < (t_ini + self.time_on/2.0):
            # First half of the cycle, t_current is calculated from 0 to time_on/2

            output = (255*(self.power/5.0))/(self.time_on/2.0)*(time.time()-
t_ini) # Here we calculate the output based on the line equation
            t_ini1 = time.time() # Aux variable to know the exact moment when the first cycle ends

        else:

            # Second half of the cycle, t_current is calculated from time_on/2 to time_on
            output = -(255*(self.power/5.0))/(self.time_on/2.0)*(time.time()-
t_ini1)+(255*(self.power/5.0))

            output = new_cell.clamp(output)
            new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)

            output=0
            new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)
            time.sleep(self.time_off)

#señal seno
def s_sine(self):

    t_ini = time.time()

    # In this cycle the vibration signal is calculated and activated through a pwm signal

    while time.time() <= (t_ini + self.time_on):

```

```

        output = (255*(self.power/5.0))*math.sin((math.pi/self.time_on)*(time.time()-
t_ini)) # In this line we calculate the half cycle of the sin function going from 0 to time_on
and having a 255 in amplitude
        new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)

        output=0
        new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0], output)
        time.sleep(self.time_off)

#señal logaritmica
def s_log(self):
    t_ini = time.time()

    # In this cycle the vibration signal is calculated and activated through a pwm signal

    while time.time() <= (t_ini + self.time_on):

        if time.time() < (t_ini + self.time_on/2.0):
            # First half of the cycle, t_current is calculated from 0 to time_on/2

            t_current = (2*(10-
1)/(self.time_on))*(time.time() - t_ini) + 1 # In this line we remap the time so that is goes
s from 1 to 10
            t_ini1 = time.time()

        else:
            # Second half of the cycle, t_current is calculated from time_on/2 to time_on

            t_current = (2*(1-
10)/(self.time_on))*(time.time() - t_ini1) + 10 # In this line we remap the time so that is go
es from 10 to 1

            output = (255.0*math.log10(t_current))*(self.power/5.0)
            output = new_cell.clamp(output)

            new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)

            output=0
            new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)
            time.sleep(self.time_off)

#señal exponencial
def s_exp(self):
    t_ini = time.time()

    # In this cycle the vibration signal is calculated and activated through a pwm signal

    while time.time() <= (t_ini + self.time_on):

        if time.time() < (t_ini + self.time_on/2.0):
            # First half of the cycle, t_current is calculated from 0 to time_on/2

            t_current = (2*(math.log(10)-(-
math.log(10)))/(self.time_on))*(time.time() - t_ini) + (-
math.log(10)) # In this line we remap the time so that is goes from -ln(10) to ln(10)
            t_ini1 = time.time()

        else:
            # Second half of the cycle, t_current is calculated from time_on/2 to time_on

            t_current = (2*((-math.log(10))-
math.log(10))/(self.time_on))*(time.time() - t_ini1) + math.log(10) # In this line we remap the
time so that is goes from ln(10) to -ln(10)

```

```

        output = (25.5*math.exp(t_current))*(self.power/5.0)
        output = new_cell.clamp(output)

        new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0], output)

    output=0
    new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)
    time.sleep(self.time_off)

#señal click (logratimica + exponencial)
def s_click(self):
    t_ini = time.time()

    # In this cycle the vibration signal is calculated and activated through a pwm signal
    while time.time() <= (t_ini + self.time_on):
        if time.time() < (t_ini + self.time_on/2.0):
            # First half of the cycle, t_current is calculated from 0 to time_on/2

            t_current = (2*(10-
1)/(self.time_on))*(time.time() - t_ini) + 1 # In this line we remap the time so that is goes fr
om 10 to 1
            output = (255.0*math.log10(t_current))*(self.power/5.0)
            t_ini1 = time.time()

        else:
            # Second half of the cycle, t_current is calculated from time_on/2 to time_on

            t_current = (2*((-math.log(10))-
math.log(10))/(self.time_on))*(time.time() - t_ini1) + math.log(10) # In this line we remap the
time so that is goes from -ln(10) to ln(10)
            output = (25.5*math.exp(t_current))*(self.power/5.0)

        output = new_cell.clamp(output)

        new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)

    output=0
    new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)
    time.sleep(self.time_off)

#señal rev-click (exponencial + logartimica)
def s_revclick(self):
    t_ini = time.time()

    # In this cycle the vibration signal is calculated and activated through a pwm signal
    while time.time() <= (t_ini + self.time_on):
        if time.time() < (t_ini + self.time_on/2.0):
            # First half of the cycle, t_current is calculated from 0 to time_on/2

            t_current = (2*(math.log(10)-(-
math.log(10)))/(self.time_on))*(time.time() - t_ini) + (-
math.log(10)) # In this line we remap the time so that is goes from -ln(10) to ln(10)
            t_ini1 = time.time()
            output = (25.5*math.exp(t_current))*(self.power/5.0)

        else:
            # Second half of the cycle, t_current is calculated from time_on/2 to time_on

```

```

        t_current = (2*(1-
10)/(self.time_on))*(time.time() - t_ini1) + 10 # In this line we remap the time so that is goes
from 10 to 1
        output = (255.0*math.log10(t_current))*(self.power/5.0)

        output = new_cell.clamp(output)

        new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)

        output=0
        new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)
        time.sleep(self.time_off)

#señal rampa
def s_ramp(self):

    t_ini = time.time()

    # In this cycle the vibration signal is calculated and activated through a pwm signal
    while time.time() <= (t_ini + self.time_on):

        output = (255*(self.power/5.0))/(self.time_on)*(time.time()-t_ini)
        new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)

        output=0
        new_cell.pi.set_PWM_dutycycle(new_cell.vibration_pins[0],output)
        time.sleep(self.time_off)

    @staticmethod
    def randletter():

        # This function generates a random letter

        alfabeto_regular = "abcdefghijklmnñopqrstuvwxyz"
        return alfabeto_regular[random.randint(0,26)]

    @staticmethod
    def translator(letter):

        # This function translates a letter to a braille dot pattern

        letter.lower()

        regular_alphabet = "abcdefghijklmnñopqrstuvwxyz"
        braille_alphabet = [[0,0,0,0,0,0,0],[1,0,0,0,0,0,0],[1,1,0,0,0,0,0],[1,0,0,1,0,0,0],
,0],[1,0,0,1,1,0,0,0],[1,0,0,0,1,0,0,0],[1,1,0,1,0,0,0,0],[1,1,0,1,1,0,0,0],[1,1,0,0,1,0,0,0],[0
,1,0,1,0,0,0,0],[0,1,0,1,1,0,0,0],[1,0,1,0,0,0,0,0],[1,1,1,0,0,0,0,0],[1,0,1,1,0,0,0,0],[1,0,1,1
,1,0,0,0,0],[1,1,0,1,1,1,0,0],[1,0,1,0,1,0,0,0],[1,1,1,1,0,0,0,0],[1,1,1,1,1,0,0,0],[1,1,1,0,1,0,0
,0],[0,1,1,1,0,0,0,0],[0,1,1,1,1,0,0,0],[1,0,1,0,0,1,0,0],[1,1,1,0,0,1,0,0],[0,1,0,1,1,1,0,0],[1
,0,1,1,0,1,0,0],[1,0,1,1,1,1,0,0],[1,0,1,0,1,1,0,0]]

        braille_dictionary = dict(zip(regular_alphabet, braille_alphabet))

        return braille_dictionary[letter]

    def trigger(self, dot_pattern):

        '''This function triggers the actuator

        V_braille ==> braille pattern, it's a boolean array with 6 values, ex {1,0,0,1,1,0
}
        signal ==> type of vibratino signal, it's a number between 1 and 8, 1 - Squa
re signal
                                                    2 - Tria
ngle signal

```

<pre> k signal (logarithmic + exponential) signal nential signal signal rse click signal (exponential + logarithmic) rithmic signal ... for iteration_num in range(6): new_cell.pi.write(new_cell.vibration_pins[iteration_num+1], dot_pattern[iteration_num]) if self.signal_type == 1: self.s_square() elif self.signal_type == 2: self.s_triad() elif self.signal_type == 6: self.s_sine() elif self.signal_type == 8: self.s_log() elif self.signal_type == 5: self.s_exp() elif self.signal_type == 3: self.s_click() elif self.signal_type == 7: self.s_revclick() elif self.signal_type == 4: self.s_ramp() else: print("Error: wrong signal selector") for iteration_num in range(6): new_cell.pi.write(new_cell.vibration_pins[iteration_num+1], 0) def generator(self): # This function activates all the dots in the cell going one by one for active_dot in range(0,6): vector_generator = [0,0,0,0,0,0] vector_generator[active_dot] = 1 self.trigger(vector_generator) def writer(self, sentence): # This function writes the sentence in the braille cell, the sentence can be a letter, a word or a paragraph. </pre>	<pre> 3 - Click 4 - Ramp 5 - Expo 6 - Sin 7 - Reverse 8 - logarithmic </pre>
---	--

```

    for letter in sentence:
        caracter_braille = self.translator(letter)
        self.trigger(caracter_braille)

def randomvib(self):

    # This function actives dots randomly in the cell

    vector_random = [0,0,0,0,0,0]

    for dot in range(0,6):
        vector_random[dot] = random.randint(0,1)

    self.trigger(vector_random)

```

Anexo D. Código de la aplicación web.

Aplicación web

HTML

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Rillo!</title>

<!-- Firebase -->
<script src="https://www.gstatic.com/firebasejs/6.2.4/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/6.2.4/firebase-firestore.js"></script>
<script>
// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyDDs9uzyIGzS97U9EjP9bzAolzCQKHShHM",
  authDomain: "rillo-web.firebaseio.com",
  databaseURL: "https://rillo-web.firebaseio.com",
  projectId: "rillo-web",
  storageBucket: "rillo-web.appspot.com",
  messagingSenderId: "78389839019",
  appId: "1:78389839019:web:4c049ee8d70d42a3"
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
</script>

<!-- Bootstrap -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPm49" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></script>

<link href="css/style.css" rel="stylesheet"/>
</head>

<body>

```

```

<div class="container">

  <div class="row">

    <a href="profiles.html" class="col-3 mt-4" >
      
    </a>
    <h1 class="col-5 text-center align-middle" id="titulo-rillo">Rillo!</h1>
    <div class="col-4"></div>

  </div>

  <div class="row" style="margin-top: 2%;">

    <div class="col-3"></div>
    <div class="col-5">
      <input type="text" class="form-control" id="barra-
escribir" placeholder="Escribe aquí...">
    </div>

    <div class="col-4">
      <button type="button" class="btn btn-primary" id="boton-
enviar" onclick="enviador();">Enviar</button>
    </div>

  </div>

  <div class="row mt-4">

    <div class="col-3"></div>
    <div class="col-5">
      <div class="row">
        <div class="col-6">
          <button type="button" class="btn btn-light" id="boton-
generador" onclick="send_generador();"> <br> Generador</button>
        </div>

        <div class="col-6">
          <button type="button" class="btn btn-light" id="boton-l-
aleatoria" onclick="send_letra_aleatoria();">
            <h1 id="letra-aleatoria">a</h1>
            <p id="texto-l-aleatoria">Letra aleatoria</p>
          </button>
        </div>
      </div>
    </div>

    <div class="col-4"></div>

  </div>

  <div class="row mt-2">

    <div class="col-3"></div>
    <div class="col-5">
      <div class="row">
        <div class="col">
          <button type="button" class="btn btn-light align-middle" id="boton-
patron" onclick="toggle_patron();"><p>Patron</p></button>
        </div>
      </div>
    </div>

  </div>

```

```

        <div class="col-4"></div>

    </div>

    <div id = "celda"></div>

    <div class="row">
        <div class="col-3"></div>
        <div class="col-5">
            <div class="row">
                <div class="col">
                    <button type="button" class="btn btn-light" id="boton-
conectar" onclick="conectado_boton();" > <h4 id="texto-conectar">Conectar</h4></button>
                </div>
            </div>
        </div>

        <div class="col-4"></div>

    </div>

</div>

<script>

var db = firebase.firestore();
var celda_patron_on = false;
var array_patron = [0, 0, 0, 0, 0, 0];

function enviador(){
    var lectura_barra = document.getElementById('barra-escribir').value;

    if(lectura_barra!="")
    {
        db.collection("rillo-main").doc("funciones").update({
            dato: lectura_barra,
            funcion: "escribir",
            recibido: 0
        })
    }

    document.getElementById("boton-enviar").classList.remove("btn-primary");
    document.getElementById("boton-enviar").classList.add("btn-warning");
    setTimeout(function(){
        document.getElementById("boton-enviar").classList.remove("btn-warning");
        document.getElementById("boton-enviar").classList.add("btn-primary");
        document.getElementById('barra-escribir').value = '';}, 500);

}

function send_generador(){
    if(lectura_barra!="")
    {
        db.collection("rillo-main").doc("funciones").update({
            dato: lectura_barra,
            funcion: "escribir",
            recibido: 0
        })
    }

    document.getElementById("boton-enviar").classList.remove("btn-primary");
    document.getElementById("boton-enviar").classList.add("btn-warning");
    setTimeout(function(){

```

```

        document.getElementById("boton-enviar").classList.remove("btn-warning");
        document.getElementById("boton-enviar").classList.add("btn-primary");}, 500);
    }

    function send_generador(){

        db.collection("rillo-main").doc("funciones").update({
            funcion: "generador",
            recibido: 0
        })

        document.getElementById("boton-generador").classList.remove("btn-light");
        document.getElementById("boton-generador").classList.add("btn-primary");
        document.getElementById("generador-icon").src = "imagenes/gen-iconw.png"
        setTimeout(function(){
            document.getElementById("boton-generador").classList.remove("btn-primary");
            document.getElementById("boton-generador").classList.add("btn-light")
            document.getElementById("generador-icon").src = "imagenes/gen-icon.png";}, 500);
    }

    function send_letra_aleatoria(){

        var abecedario = "abcdefghijklmnñopqrstuvwxyz";
        var num_random = Math.floor((Math.random() * 26) + 1);

        var letra = abecedario[num_random]

        db.collection("rillo-main").doc("funciones").update({
            dato: letra,
            funcion: "escribir",
            recibido: 0
        })

        document.getElementById('letra-aleatoria').innerHTML = letra;
        document.getElementById("boton-l-aleatoria").classList.remove("btn-light");
        document.getElementById("boton-l-aleatoria").classList.add("btn-primary");
        setTimeout(function(){
            document.getElementById("boton-l-aleatoria").classList.remove("btn-primary");
            document.getElementById("boton-l-aleatoria").classList.add("btn-light");}, 500);
    }

    db.collection("rillo-main").doc("funciones")
        .onSnapshot(function(doc) {

            conectado_listener(doc.data());
        });

    function conectado_listener(data){
        if(data["conectado"]){
            document.getElementById("boton-conectar").classList.remove("btn-light");
            document.getElementById("boton-conectar").classList.add("btn-primary");
            document.getElementById('texto-conectar').innerHTML = "Desconectar";
        }
        else{
            document.getElementById("boton-conectar").classList.remove("btn-primary");
            document.getElementById("boton-conectar").classList.add("btn-light")
            document.getElementById('texto-conectar').innerHTML = "Conectar";
        }
    }

    function conectado_boton(){

        var ver_conexión = new Promise((resolve, reject) => {
            db.collection("rillo-main").doc("funciones").get().then(function(doc) {
                if (doc.exists) {

                    datos_conexion = doc.data();
                    resolve(doc.data());
                }
            })
        })
    }

```

```

        } else {
            // doc.data() will be undefined in this case
            reject = [];
            console.log("No such document!");
        }
    }).catch(function(error) {
        console.log("Error getting document:", error);
    });
});

ver_conexión.then((datos_conexion) => {
    if(datos_conexion["conectado"]){
        db.collection("rillo-main").doc("funciones").update({
            conectado: false
        })
    }
    else{
        db.collection("rillo-main").doc("funciones").update({
            funcion: "nope",
            recibido: 0
        })
    }
}).catch((error) => console.log("error"))

}

function toggle_patron(){

    celda_patron_on = !celda_patron_on;

    if(celda_patron_on){
        array_patron = [0, 0, 0, 0, 0, 0];
        document.getElementById('celda').innerHTML = '<div class="row mt-2 mb-4"><div class="col-3"></div><div class="col-5"><div class="row"><div class="col"><div class="row"><div class="col text-center col-celda"><button type="button" class="btn btn-light boton_celda_iz" id="btn_celda_1" onclick="celda_press(1)"></button><button type="button" class="btn btn-light boton_celda_iz" id="btn_celda_2" onclick="celda_press(2)"></button><button type="button" class="btn btn-light boton_celda_iz" id="btn_celda_3" onclick="celda_press(3)"></button></div><div class="col text-center col-celda"><button type="button" class="btn btn-light boton_celda_der" id="btn_celda_4" onclick="celda_press(4)"></button><button type="button" class="btn btn-light boton_celda_der" id="btn_celda_5" onclick="celda_press(5)"></button><button type="button" class="btn btn-light boton_celda_der" id="btn_celda_6" onclick="celda_press(6)"></button></div></div></div><div class="col-4"><button type="button" class="btn btn-primary btn_celda" id="btn_enviar_celda" onclick="enviar_celda()"><p>Enviar</p></button><button type="button" class="btn btn-light btn_celda" onclick="limpiar_celda()"><p>Limpiar</p></button></div></div>';
    }
    else{
        document.getElementById('celda').innerHTML = '';
    }
}

function celda_press(numero_celda){

    switch(numero_celda) {
        case 1:
    
```

```

        if(array_patron[0]){
            array_patron[0] = 0;
            document.getElementById("btn_celda_1").classList.remove("btn-primary");
            document.getElementById("btn_celda_1").classList.add("btn-light");
            document.getElementById("circulo_celda_1").src = "imagenes/circulo_celda.png"
        }
        else{
            array_patron[0] = 1;
            document.getElementById("btn_celda_1").classList.remove("btn-light");
            document.getElementById("btn_celda_1").classList.add("btn-primary");
            document.getElementById("circulo_celda_1").src = "imagenes/circulo_celdaw.png"
        }

        break;
    case 2:
        if(array_patron[1]){
            array_patron[1] = 0;
            document.getElementById("btn_celda_2").classList.remove("btn-primary");
            document.getElementById("btn_celda_2").classList.add("btn-light");
            document.getElementById("circulo_celda_2").src = "imagenes/circulo_celda.png"
        }
        else{
            array_patron[1] = 1;
            document.getElementById("btn_celda_2").classList.remove("btn-light");
            document.getElementById("btn_celda_2").classList.add("btn-primary");
            document.getElementById("circulo_celda_2").src = "imagenes/circulo_celdaw.png"
        }

        break;
    case 3:
        if(array_patron[2]){
            array_patron[2] = 0;
            document.getElementById("btn_celda_3").classList.remove("btn-primary");
            document.getElementById("btn_celda_3").classList.add("btn-light");
            document.getElementById("circulo_celda_3").src = "imagenes/circulo_celda.png"
        }
        else{
            array_patron[2] = 1;
            document.getElementById("btn_celda_3").classList.remove("btn-light");
            document.getElementById("btn_celda_3").classList.add("btn-primary");
            document.getElementById("circulo_celda_3").src = "imagenes/circulo_celdaw.png"
        }

        break;
    case 4:
        if(array_patron[3]){
            array_patron[3] = 0;
            document.getElementById("btn_celda_4").classList.remove("btn-primary");
            document.getElementById("btn_celda_4").classList.add("btn-light");
            document.getElementById("circulo_celda_4").src = "imagenes/circulo_celda.png"
        }
        else{
            array_patron[3] = 1;
            document.getElementById("btn_celda_4").classList.remove("btn-light");
            document.getElementById("btn_celda_4").classList.add("btn-primary");
            document.getElementById("circulo_celda_4").src = "imagenes/circulo_celdaw.png"
        }

        break;
    case 5:
        if(array_patron[4]){
            array_patron[4] = 0;
            document.getElementById("btn_celda_5").classList.remove("btn-primary");
            document.getElementById("btn_celda_5").classList.add("btn-light");
            document.getElementById("circulo_celda_5").src = "imagenes/circulo_celda.png"
        }
        else{
            array_patron[4] = 1;
            document.getElementById("btn_celda_5").classList.remove("btn-light");
            document.getElementById("btn_celda_5").classList.add("btn-primary");
            document.getElementById("circulo_celda_5").src = "imagenes/circulo_celdaw.png"
        }

        break;
    case 6:

```

```

        if(array_patron[5]){
            array_patron[5] = 0;
            document.getElementById("btn_celda_6").classList.remove("btn-primary");
            document.getElementById("btn_celda_6").classList.add("btn-light");
            document.getElementById("circulo_celda_6").src = "imagenes/circulo_celda.png"
        }
        else{
            array_patron[5] = 1;
            document.getElementById("btn_celda_6").classList.remove("btn-light");
            document.getElementById("btn_celda_6").classList.add("btn-primary");
            document.getElementById("circulo_celda_6").src = "imagenes/circulo_celdaw.png"
        }
        break;
    default:
        // code block
    }
}

function limpiar_celda(){
    array_patron = [0, 0, 0, 0, 0, 0];

    document.getElementById("btn_celda_1").classList.remove("btn-primary");
    document.getElementById("btn_celda_1").classList.add("btn-light");
    document.getElementById("circulo_celda_1").src = "imagenes/circulo_celda.png"

    document.getElementById("btn_celda_2").classList.remove("btn-primary");
    document.getElementById("btn_celda_2").classList.add("btn-light");
    document.getElementById("circulo_celda_2").src = "imagenes/circulo_celda.png"

    document.getElementById("btn_celda_3").classList.remove("btn-primary");
    document.getElementById("btn_celda_3").classList.add("btn-light");
    document.getElementById("circulo_celda_3").src = "imagenes/circulo_celda.png"

    document.getElementById("btn_celda_4").classList.remove("btn-primary");
    document.getElementById("btn_celda_4").classList.add("btn-light");
    document.getElementById("circulo_celda_4").src = "imagenes/circulo_celda.png"

    document.getElementById("btn_celda_5").classList.remove("btn-primary");
    document.getElementById("btn_celda_5").classList.add("btn-light");
    document.getElementById("circulo_celda_5").src = "imagenes/circulo_celda.png"

    document.getElementById("btn_celda_6").classList.remove("btn-primary");
    document.getElementById("btn_celda_6").classList.add("btn-light");
    document.getElementById("circulo_celda_6").src = "imagenes/circulo_celda.png"
}

function enviar_celda(){
    db.collection("rillo-main").doc("funciones").update({
        dato: array_patron,
        funcion: "celda",
        recibido: 0
    })

    document.getElementById("btn_enviar_celda").classList.remove("btn-primary");
    document.getElementById("btn_enviar_celda").classList.add("btn-warning");
    setTimeout(function(){
        document.getElementById("btn_enviar_celda").classList.remove("btn-warning");
        document.getElementById("btn_enviar_celda").classList.add("btn-primary");
    }, 500);
}

</script>

</body>
</html>

```

CSS

```
#profile-icon{
  margin-left: 50%;
  width: 30%;
  border-radius: 20%;
}

#titulo-rillo{
  margin-top: 2%;
  font-size: 4.5vw;
}

#barra-escribir{
  width: 100%;
  margin-left: auto;
}

#boton-enviar{
  width: 40%;
  margin-right: auto;
}

#boton-generador{
  width: 100%;
}

#generador-icon{
  margin-top: 5%;
  margin-bottom: 5%;
  width: 25%;
}

#boton-l-aleatoria{
  height: 100%;
  width: 100%;
}

#letra-aleatoria{
  font-size: 150%;
  margin: 0;
}

#texto-l-aleatoria{
  display: block;
  text-align: center;
  margin-top: 20%;
  margin-bottom: 0%;
  line-height: 100%;
}

#boton-conectar{
  margin-top: 5%;
  width: 100%;
}

.col-celda{
  padding: 0% 1% 0% 1%;
}

#boton-patron{
  width: 100%;
}

#boton-patron p{
  font-size: 1.3em;
  margin: 0%;
}

.boton_celda_iz{
```

```

        float: right;
        margin: 3% 0% 0% 50%;
        height: 35%;
        border-radius: 10%;
    }

    .boton_celda_der{
        float: left;
        margin: 3% 50% 0% 0%;
        height: 35%;
        border-radius: 10%;
    }

    .circulo_celda{
        width: 100%;
    }

    .btn_celda{
        float:left;
        height: 20%;
        width: 45%;
        margin-right: 50%;
        margin-bottom: 5%;
    }

    .btn_celda p{
        font-size: 1.2em;
        margin: 0%;
    }

```

Anexo E. Código de entrenamiento del algoritmo de OCR

Training code for OCR (Manually)

import sys

import numpy as np

import cv2

#-----PRE-PROCESSING-----

letras = [] ##Lista con las letras identificadas de manera manual

responses = [] #Lista con las letras identificadas de manera manual (Valor decimal equivalente)

b = 1 #Variable de control para la selección de imágenes de entrenamiento

samples = np.empty((0,100))

keys = [i for i in range(97,123)]

keys.append(241) #Codigo para la letra 'ñ'

while b<28:

#-----

#ADQUISICIÓN

#Realizamos el request para obtener la imagen de la cámara en cada ciclo

image_n = f"Training/Image ({b}).jpg"

img = cv2.imread(image_n, cv2.IMREAD_COLOR)

cv2.imshow('Input Image', img)

#HACEMOS UN RESIZE PARA DEFINIR EL TAMAÑO ADECUADO DE LA IMAGEN A MOSTRARSE

```

img = cv2.resize(img,(600,400))
width = len(img[0])*(2/3)
img = img[:,0:int(width)]
cv2.imshow('Resized Image', img)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

#Image Smoothing
blur = cv2.GaussianBlur(gray,(5,5),0)#suaviza los bordes
cv2.imshow("Smoothed", blur)

#-----SEGMENTATION-----
-----
thresh = cv2.adaptiveThreshold(blur,255,1,1,31,2)
cv2.imshow('Preprocessed Image', thresh)

#-----EDGES DETECTION-----
-----
contours,hierarchy =
cv2.findContours(thresh,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)

for cnt in contours:
    if (cv2.contourArea(cnt)>1000 and cv2.contourArea(cnt)<5600) or
(cv2.contourArea(cnt)>5800 and cv2.contourArea(cnt)<18000):
        [x,y,w,h] = cv2.boundingRect(cnt)
        if h>100:
            #Se grega try para que el código se ejecute cuando la letra queda cerca al
límite superior de la imagen
            print("Altura",h)
            try:
                print(cv2.contourArea(cnt))
                cv2.rectangle(img,(x,y-50),(x+w,y+h),(0,0,255),2)
                roi = thresh[y-50:y+h,x:x+w]
                cv2.imshow('letra',roi)
            except cv2.error:
                print("Contour area", cv2.contourArea(cnt))
                cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
                roi = thresh[y:y+h,x:x+w]
                cv2.imshow('letra',roi)
            roismall = cv2.resize(roi,(10,10))
            cv2.imshow('norm',img)
            key = cv2.waitKey(0)

#-----CHARACTERISTICS EXTRACTION-----
-----
if key == 27: # (escape to quit)
    sys.exit()
elif key == 8: # backspace(del) para saltar la letra/contorno reconocido
    continue
elif key in keys:
    print("Se ingresó la letra: {}".format(chr(key)))
    letras.append(chr(key))
    responses.append(int(key))
    sample = roismall.reshape((1,100))

```

```

        cv2.imshow("almacenada", sample)
        samples = np.append(samples, sample, 0)
    b += 1
#-----
# Para verificar si todas las letras del alfabeto fueron identificadas:
alfabeto = ['q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p', 'a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'ñ', 'z', 'x', 'c', 'v',
'b', 'n', 'm']
letras_no_ide = 0
for letra in alfabeto:
    if letra in letras:
        continue
    else:
        letras_no_ide += 1
        print("La letra {} no fue identificada.".format(letra))
print("Letras identificadas: {}. \nLetras no identificadas: {}".format(len(letras),
letras_no_ide))
print("Letras identificadas:\n", letras)
#-----
#Para almacenar las letras identificadas
responses = np.array(responses, np.float32)
responses = responses.reshape((responses.size, 1))
print("Tamaño muestras: ", samples.shape)
print("Tamaño de respuestas: ", responses.shape)
print("training complete")
np.savetxt('generalsamples.data', samples)
np.savetxt('generalresponses.data', responses)

```

Anexo F. Código de ejecución del algoritmo de OCR

```

import cv2
import numpy as np

def camera_reader(model, image_size: tuple=(600, 400), image_crop: float=2/3,
blur_amount: int=5):
    """ Read the camera input and processes it to get a string using ocr.

    Parameters
    -----
    model : [type]
        Trained ocr model
    image_size : tuple, optional
        Dimensions of the image resizing, by default (600, 400)
    image_crop : float, optional
        Zone of the image that will be processed (from 0 to 1.0), by default 2/3
    blur_amount : int, optional
        Amount of blur in the processed image, by default 5

    Returns
    """

```

```

-----
read_character: str
    Recognized character
"""

capture = cv2.VideoCapture(0)
_, img = capture.read()
# cv2.imwrite("SavedImage.jpg", img)
img = cv2.resize(img, image_size)
width = len(img[0])*image_crop
img = img[:,0:int(width)]
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (blur_amount,)*2, 0)
thresh = cv2.adaptiveThreshold(blur, 255, 1, 1, 31, 2)
contours, hierarchy =
cv2.findContours(thresh, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)

read_character = " "
for cnt in contours:
    if cv2.contourArea(cnt)>2000:
        [x,y,w,h] = cv2.boundingRect(cnt)
        if h>100:
            try:
                roi = thresh[y-50:y+h,x:x+w]
            except cv2.error:
                roi = thresh[y:y+h,x:x+w]

            try:
                roismall = cv2.resize(roi, (10,10))
                roismall = roismall.reshape((1,100))
                roismall = np.float32(roismall)
                retval, results, neigh_resp, dists = model.findNearest(roismall, k = 1)
                read_character = str(chr((results[0][0])))
            except cv2.error:
                print("Error durante la lectura")

capture.release()
cv2.destroyAllWindows()
return read_character

```

Anexo G. Tablas de medición de la autonomía del dispositivo.

USO INTENSIVO

Tabla 1. Medición de la autonomía del dispositivo con un uso intensivo

voltaje bateria	Autonomia (minutos)
3.944281525	0
3.929618768	1.016666667
3.914956012	2.033333333
3.905180841	3.05
3.89540567	4.066666667
3.885630499	5.1
3.880742913	6.116666667
3.866080156	7.133333333
3.861192571	8.15
3.8514174	9.166666667
3.846529814	10.18333333
3.846529814	11.2
3.831867058	12.21666667
3.822091887	13.23333333
3.817204301	14.25
3.80742913	15.26666667
3.80742913	16.28333333
3.792766373	17.3
3.787878788	18.31666667
3.778103617	19.33333333
3.768328446	20.35
3.768328446	21.36666667
3.758553275	22.38333333
3.739002933	23.4
3.739002933	24.41666667
3.734115347	25.43333333
3.729227761	26.45
3.709677419	27.46666667

3.709677419	28.48333333
3.695014663	29.5
3.695014663	30.51666667
3.690127077	31.53333333
3.680351906	32.55
3.675464321	33.56666667
3.66568915	34.58333333
3.660801564	35.6
3.651026393	36.61666667
3.646138807	37.63333333
3.646138807	38.65
3.631476051	39.66666667
3.62170088	40.68333333
3.62170088	41.7
3.611925709	42.71666667
3.602150538	43.73333333
3.597262952	44.75
3.587487781	45.76666667
3.587487781	46.78333333
3.572825024	47.81666667
3.563049853	48.83333333
3.563049853	49.85
3.558162268	50.86666667
3.543499511	51.88333333
3.543499511	52.9
3.53372434	53.91666667
3.528836755	54.93333333
3.519061584	55.95
3.519061584	56.96666667
3.509286413	57.98333333
3.499511241	59
3.494623656	60.01666667

3.484848485	61.03333333
3.479960899	62.05
3.479960899	63.06666667
3.470185728	64.08333333
3.475073314	65.1
3.465298143	66.11666667
3.460410557	67.13333333
3.450635386	68.15
3.450635386	69.16666667
3.43597263	70.18333333

USO ESPERADO

Tabla 2. Medición de la autonomía del dispositivo con un uso esperado

Voltaje de la batería	Cantidad de letras	Autonomia (minutos)
4.051808407	1	0.07
4.051808407	2	0.15
3.998044966	3	0.22
3.988269795	4	0.3
3.983382209	5	0.37
3.988269795	6	0.45
3.978494624	7	0.52
3.978494624	8	0.6
3.983382209	9	0.67
3.983382209	10	0.75
3.973607038	11	0.82
3.983382209	12	0.9
3.978494624	13	0.98
3.973607038	14	1.05
3.973607038	15	1.13
3.968719453	16	1.2
3.944281525	17	1.28
3.968719453	18	1.35

3.973607038	19	1.43
3.817204301	20	1.5
3.900293255	21	1.58
3.919843597	22	1.65
3.914956012	23	1.73
3.89540567	24	1.8
3.910068426	25	1.88
3.905180841	26	1.95
3.905180841	27	1.97
3.905180841	28	2.03
3.910068426	29	2.12
3.910068426	30	2.18
3.905180841	31	2.27
3.914956012	32	2.33
3.905180841	33	2.42
3.905180841	34	2.48
3.905180841	35	2.57
3.905180841	36	2.63
3.900293255	37	2.72
3.910068426	38	2.78
3.905180841	39	2.87
3.910068426	40	2.93
3.900293255	41	3.02
3.905180841	42	3.08
3.900293255	43	3.17
3.900293255	44	3.23
3.900293255	45	3.32
3.900293255	46	3.38
3.89540567	47	3.47
3.89540567	48	3.53
3.890518084	49	3.62
3.89540567	50	3.68

3.89540567	51	3.77
3.890518084	52	3.83
3.890518084	53	3.92
3.890518084	54	3.98
3.890518084	55	4.07
3.885630499	56	4.07
3.890518084	57	4.15
3.890518084	58	4.22
3.89540567	59	4.3
3.890518084	60	4.37
3.885630499	61	4.45
3.885630499	62	4.52
3.885630499	63	4.6
3.880742913	64	4.67
3.880742913	65	4.75
3.875855327	66	4.82
3.880742913	67	4.9
3.880742913	68	4.97
3.880742913	69	5.05
3.880742913	70	5.13
3.880742913	71	5.2
3.885630499	72	5.28
3.875855327	73	5.35
3.880742913	74	5.43
3.885630499	75	5.5
3.875855327	76	5.58
3.875855327	77	5.65
3.870967742	78	5.73
3.875855327	79	5.8
3.875855327	80	5.88
3.880742913	81	5.95
3.875855327	82	6.03

3.875855327	83	6.1
3.875855327	84	6.18
3.880742913	85	6.25
3.875855327	86	6.33
3.870967742	87	6.4
3.870967742	88	6.48
3.870967742	89	6.55
3.870967742	90	6.63
3.870967742	91	6.7
3.866080156	92	6.78
3.866080156	93	6.85
3.870967742	94	6.93
3.866080156	95	7
3.861192571	96	7.08
3.866080156	97	7.15
3.866080156	98	7.23
3.861192571	99	7.3
3.866080156	100	7.32
3.861192571	101	7.38
3.870967742	102	7.47
3.861192571	103	7.53
3.856304985	104	7.62
3.870967742	105	7.68
3.861192571	106	7.77
3.856304985	107	7.83
3.866080156	108	7.92
3.866080156	109	7.98
3.861192571	110	8.07
3.866080156	111	8.13
3.866080156	112	8.22
3.866080156	113	8.28
3.870967742	114	8.37

3.870967742	115	8.43
3.856304985	116	8.52
3.870967742	117	8.58
3.866080156	118	8.67
3.861192571	119	8.73
3.866080156	120	8.82
3.866080156	121	8.9
3.861192571	122	8.97
3.861192571	123	9.05
3.861192571	124	9.12
3.866080156	125	9.2
3.861192571	126	9.27
3.856304985	127	9.35
3.856304985	128	9.42
3.846529814	129	9.43
3.8514174	130	9.5
3.856304985	131	9.58
3.8514174	132	9.65
3.8514174	133	9.73
3.856304985	134	9.8
3.846529814	135	9.88
3.861192571	136	9.95
3.856304985	137	10.03
3.8514174	138	10.1
3.856304985	139	10.18
3.856304985	140	10.25
3.8514174	141	10.33
3.856304985	142	10.4
3.846529814	143	10.48
3.846529814	144	10.55
3.8514174	145	10.63
3.846529814	146	10.7

3.8514174	147	10.78
3.846529814	148	10.85
3.8514174	149	10.93
3.846529814	150	11
3.846529814	151	11.08
3.841642229	152	11.15
3.846529814	153	11.23
3.8514174	154	11.3
3.846529814	155	11.38
3.836754643	156	11.38
3.846529814	157	11.47
3.836754643	158	11.53
3.836754643	159	11.62
3.846529814	160	11.68
3.846529814	161	11.77
3.836754643	162	11.83
3.841642229	163	11.92
3.841642229	164	11.98
3.841642229	165	12.07
3.836754643	166	12.13
3.836754643	167	12.22
3.836754643	168	12.28
3.836754643	169	12.37
3.831867058	170	12.43
3.836754643	171	12.52
3.836754643	172	12.58
3.841642229	173	12.67
3.836754643	174	12.75
3.831867058	175	12.75
3.836754643	176	12.83
3.831867058	177	12.9
3.836754643	178	12.98

3.836754643	179	13.05
3.836754643	180	13.13
3.826979472	181	13.2
3.826979472	182	13.28
3.826979472	183	13.35
3.831867058	184	13.43
3.826979472	185	13.5
3.826979472	186	13.58
3.826979472	187	13.65
3.826979472	188	13.73
3.831867058	189	13.8
3.826979472	190	13.88
3.826979472	191	13.95
3.826979472	192	14.03
3.831867058	193	14.1
3.822091887	194	14.18
3.822091887	195	14.25
3.822091887	196	14.27
3.822091887	197	14.33
3.826979472	198	14.42
3.826979472	199	14.48
3.826979472	200	14.57
3.817204301	201	14.63
3.822091887	202	14.72
3.822091887	203	14.78
3.817204301	204	14.87
3.822091887	205	14.93
3.822091887	206	15.02
3.822091887	207	15.08
3.817204301	208	15.17
3.812316716	209	15.23
3.817204301	210	15.32

3.817204301	211	15.38
3.817204301	212	15.47
3.812316716	213	15.53
3.812316716	214	15.62
3.817204301	215	15.68
3.817204301	216	15.77
3.812316716	217	15.83
3.80742913	218	15.92
3.817204301	219	15.98
3.80742913	220	16.07
3.812316716	221	16.07
3.817204301	222	16.15
3.812316716	223	16.22
3.812316716	224	16.3
3.817204301	225	16.37
3.812316716	226	16.45
3.812316716	227	16.52
3.812316716	228	16.6
3.80742913	229	16.68
3.812316716	230	16.75
3.802541544	231	16.83
3.80742913	232	16.9
3.80742913	233	16.98
3.80742913	234	17.05
3.802541544	235	17.13
3.80742913	236	17.2
3.80742913	237	17.28
3.802541544	238	17.35
3.80742913	239	17.43
3.802541544	240	17.5
3.80742913	241	17.58
3.802541544	242	17.65

3.802541544	243	17.73
3.802541544	244	17.8
3.802541544	245	17.88
3.797653959	246	17.95
3.797653959	247	18.03
3.802541544	248	18.1
3.797653959	249	18.18
3.802541544	250	18.25
3.792766373	251	18.33
3.802541544	252	18.4
3.797653959	253	18.48
3.797653959	254	18.55
3.797653959	255	18.63
3.797653959	256	18.63
3.802541544	257	18.72
3.802541544	258	18.78
3.797653959	259	18.87
3.787878788	260	18.93
3.802541544	261	19.02
3.797653959	262	19.08
3.792766373	263	19.17
3.792766373	264	19.23
3.792766373	265	19.32
3.792766373	266	19.38
3.792766373	267	19.47
3.792766373	268	19.53
3.792766373	269	19.62
3.797653959	270	19.68
3.792766373	271	19.77
3.792766373	272	19.83
3.792766373	273	19.92
3.792766373	274	19.98

3.792766373	275	20.07
3.787878788	276	20.13
3.792766373	277	20.22
3.778103617	278	20.22
3.792766373	279	20.3
3.787878788	280	20.37
3.787878788	281	20.45
3.782991202	282	20.52
3.782991202	283	20.6
3.787878788	284	20.67
3.787878788	285	20.75
3.782991202	286	20.83
3.787878788	287	20.9
3.782991202	288	20.98
3.782991202	289	21.05
3.778103617	290	21.13
3.782991202	291	21.2
3.782991202	292	21.28
3.782991202	293	21.35
3.782991202	294	21.43
3.782991202	295	21.5
3.773216031	296	21.58
3.778103617	297	21.65
3.778103617	298	21.73
3.778103617	299	21.8
3.778103617	300	21.88
3.778103617	301	21.95
3.768328446	302	22.03
3.773216031	303	22.03
3.778103617	304	22.12
3.773216031	305	22.18
3.773216031	306	22.27

3.768328446	307	22.33
3.773216031	308	22.42
3.778103617	309	22.48
3.768328446	310	22.57
3.773216031	311	22.63
3.768328446	312	22.72
3.768328446	313	22.78
3.773216031	314	22.87
3.768328446	315	22.93
3.768328446	316	23.02
3.768328446	317	23.08
3.768328446	318	23.17
3.76344086	319	23.23
3.768328446	320	23.32
3.773216031	321	23.38
3.773216031	322	23.47
3.76344086	323	23.53
3.76344086	324	23.62
3.758553275	325	23.68
3.76344086	326	23.7
3.768328446	327	23.77
3.768328446	328	23.85
3.768328446	329	23.92
3.758553275	330	24
3.758553275	331	24.07
3.76344086	332	24.15
3.76344086	333	24.22
3.758553275	334	24.3
3.758553275	335	24.37
3.76344086	336	24.45
3.758553275	337	24.45
3.758553275	338	24.53

3.768328446	339	24.62
3.76344086	340	24.68
3.753665689	341	24.77
3.753665689	342	24.83
3.753665689	343	24.92
3.76344086	344	24.98
3.758553275	345	25.07
3.76344086	346	25.13
3.758553275	347	25.22
3.758553275	348	25.28
3.753665689	349	25.37
3.758553275	350	25.43
3.758553275	351	25.52
3.753665689	352	25.58
3.758553275	353	25.67
3.753665689	354	25.73
3.753665689	355	25.82
3.748778104	356	25.88
3.753665689	357	25.97
3.753665689	358	26.03
3.748778104	359	26.12
3.748778104	360	26.18
3.748778104	361	26.27
3.753665689	362	26.27
3.748778104	363	26.35
3.748778104	364	26.42
3.748778104	365	26.5
3.753665689	366	26.57
3.739002933	367	26.65
3.748778104	368	26.72
3.739002933	369	26.8
3.753665689	370	26.87

3.743890518	371	26.95
3.753665689	372	27.02
3.748778104	373	27.1
3.753665689	374	27.17
3.743890518	375	27.25
3.748778104	376	27.32
3.743890518	377	27.4
3.743890518	378	27.4
3.748778104	379	27.48
3.753665689	380	27.55
3.743890518	381	27.63
3.748778104	382	27.7
3.743890518	383	27.78
3.739002933	384	27.85
3.739002933	385	27.93
3.734115347	386	28
3.743890518	387	28.08
3.739002933	388	28.15
3.739002933	389	28.23
3.739002933	390	28.3
3.739002933	391	28.38
3.734115347	392	28.4
3.739002933	393	28.47
3.734115347	394	28.55
3.739002933	395	28.62
3.729227761	396	28.7
3.734115347	397	28.77
3.734115347	398	28.85
3.729227761	399	28.92
3.734115347	400	29
3.734115347	401	29.07
3.734115347	402	29.15

3.734115347	403	29.22
3.729227761	404	29.3
3.729227761	405	29.37
3.729227761	406	29.45
3.734115347	407	29.52
3.729227761	408	29.6
3.71945259	409	29.67
3.729227761	410	29.75
3.729227761	411	29.82
3.71945259	412	29.9
3.729227761	413	29.97
3.734115347	414	30.05
3.729227761	415	30.12
3.734115347	416	30.2
3.729227761	417	30.27
3.71945259	418	30.35
3.714565005	419	30.42
3.71945259	420	30.5
3.71945259	421	30.57
3.724340176	422	30.58
3.71945259	423	30.65
3.71945259	424	30.73
3.71945259	425	30.8
3.724340176	426	30.88
3.724340176	427	30.95
3.71945259	428	31.03
3.71945259	429	31.1
3.71945259	430	31.18
3.71945259	431	31.25
3.714565005	432	31.33
3.71945259	433	31.4
3.71945259	434	31.48

3.71945259	435	31.55
3.71945259	436	31.63
3.714565005	437	31.7
3.71945259	438	31.78
3.71945259	439	31.85
3.714565005	440	31.93
3.714565005	441	32
3.709677419	442	32.08
3.709677419	443	32.17
3.714565005	444	32.23
3.714565005	445	32.32
3.714565005	446	32.38
3.71945259	447	32.47
3.709677419	448	32.53
3.704789834	449	32.62
3.704789834	450	32.68
3.714565005	451	32.77
3.709677419	452	32.83
3.709677419	453	32.92
3.709677419	454	32.98
3.714565005	455	33.07
3.709677419	456	33.13
3.709677419	457	33.22
3.714565005	458	33.28
3.709677419	459	33.37
3.704789834	460	33.43
3.699902248	461	33.52
3.709677419	462	33.58
3.714565005	463	33.67
3.699902248	464	33.73
3.699902248	465	33.82
3.699902248	466	33.88

3.695014663	467	33.97
3.704789834	468	34.03
3.704789834	469	34.12
3.704789834	470	34.18
3.704789834	471	34.27
3.704789834	472	34.27
3.704789834	473	34.35
3.695014663	474	34.42
3.704789834	475	34.5
3.704789834	476	34.57
3.709677419	477	34.65
3.690127077	478	34.72
3.695014663	479	34.8
3.695014663	480	34.87
3.699902248	481	34.95
3.699902248	482	35.02
3.690127077	483	35.1
3.690127077	484	35.17
3.695014663	485	35.25
3.695014663	486	35.32
3.695014663	487	35.4
3.695014663	488	35.47
3.695014663	489	35.55
3.699902248	490	35.62
3.699902248	491	35.7
3.690127077	492	35.77
3.690127077	493	35.85
3.690127077	494	35.93
3.690127077	495	36
3.685239492	496	36.08
3.699902248	497	36.15
3.695014663	498	36.23

3.680351906	499	36.3
3.690127077	500	36.38
3.690127077	501	36.45
3.695014663	502	36.53
3.690127077	503	36.6
3.695014663	504	36.68
3.690127077	505	36.75
3.695014663	506	36.83
3.690127077	507	36.9
3.685239492	508	36.98
3.695014663	509	37.05
3.690127077	510	37.13
3.685239492	511	37.2
3.680351906	512	37.22
3.685239492	513	37.28
3.680351906	514	37.37
3.685239492	515	37.43
3.690127077	516	37.52
3.690127077	517	37.58
3.685239492	518	37.67
3.680351906	519	37.73
3.680351906	520	37.82
3.675464321	521	37.88
3.685239492	522	37.97
3.675464321	523	38.03
3.680351906	524	38.12
3.690127077	525	38.18
3.675464321	526	38.27
3.675464321	527	38.33
3.685239492	528	38.42
3.675464321	529	38.48
3.680351906	530	38.57

3.680351906	531	38.63
3.675464321	532	38.72
3.675464321	533	38.78
3.675464321	534	38.87
3.670576735	535	38.93
3.680351906	536	39.02
3.675464321	537	39.08
3.670576735	538	39.17
3.670576735	539	39.23
3.680351906	540	39.32
3.680351906	541	39.38
3.670576735	542	39.4
3.670576735	543	39.47
3.675464321	544	39.55
3.675464321	545	39.62
3.675464321	546	39.7
3.670576735	547	39.78
3.670576735	548	39.85
3.670576735	549	39.93
3.675464321	550	40
3.675464321	551	40.08
3.675464321	552	40.15
3.66568915	553	40.23
3.670576735	554	40.3
3.670576735	555	40.38
3.66568915	556	40.45
3.66568915	557	40.53
3.66568915	558	40.6
3.66568915	559	40.68
3.66568915	560	40.75
3.66568915	561	40.83
3.660801564	562	40.9

3.66568915	563	40.98
3.670576735	564	41.05
3.670576735	565	41.13
3.660801564	566	41.2
3.660801564	567	41.28
3.660801564	568	41.35
3.66568915	569	41.43
3.66568915	570	41.5
3.66568915	571	41.58
3.660801564	572	41.65
3.655913978	573	41.73
3.660801564	574	41.8
3.651026393	575	41.88
3.655913978	576	41.95
3.655913978	577	42.03
3.655913978	578	42.1
3.660801564	579	42.18
3.655913978	580	42.25
3.655913978	581	42.33
3.655913978	582	42.33
3.651026393	583	42.42
3.651026393	584	42.48
3.660801564	585	42.57
3.655913978	586	42.63
3.651026393	587	42.72
3.655913978	588	42.78
3.655913978	589	42.87
3.651026393	590	42.93
3.651026393	591	43.02
3.651026393	592	43.08
3.651026393	593	43.17
3.651026393	594	43.23

3.655913978	595	43.32
3.646138807	596	43.38
3.646138807	597	43.47
3.651026393	598	43.53
3.651026393	599	43.62
3.655913978	600	43.7
3.641251222	601	43.77
3.651026393	602	43.78
3.641251222	603	43.85
3.655913978	604	43.93
3.646138807	605	44
3.646138807	606	44.08
3.641251222	607	44.15
3.655913978	608	44.23
3.641251222	609	44.3
3.636363636	610	44.38
3.641251222	611	44.45
3.636363636	612	44.53
3.646138807	613	44.6
3.641251222	614	44.68
3.641251222	615	44.75
3.636363636	616	44.83
3.641251222	617	44.9
3.646138807	618	44.98
3.631476051	619	45.05
3.636363636	620	45.13
3.631476051	621	45.2
3.631476051	622	45.28
3.641251222	623	45.35
3.641251222	624	45.37
3.636363636	625	45.43
3.646138807	626	45.52

3.636363636	627	45.58
3.631476051	628	45.67
3.636363636	629	45.73
3.631476051	630	45.82
3.636363636	631	45.88
3.641251222	632	45.97
3.631476051	633	46.03
3.636363636	634	46.12
3.636363636	635	46.18
3.636363636	636	46.27
3.641251222	637	46.33
3.636363636	638	46.42
3.631476051	639	46.48
3.636363636	640	46.57
3.636363636	641	46.63
3.62170088	642	46.72
3.631476051	643	46.78
3.626588465	644	46.87
3.636363636	645	46.93
3.636363636	646	47.02
3.626588465	647	47.08
3.636363636	648	47.17
3.626588465	649	47.17
3.631476051	650	47.25
3.626588465	651	47.32
3.631476051	652	47.4
3.626588465	653	47.47
3.631476051	654	47.55
3.626588465	655	47.62
3.626588465	656	47.7
3.62170088	657	47.77
3.626588465	658	47.85

3.62170088	659	47.93
3.62170088	660	48
3.616813294	661	48.08
3.62170088	662	48.15
3.616813294	663	48.23
3.626588465	664	48.3
3.62170088	665	48.38
3.62170088	666	48.45
3.616813294	667	48.53
3.626588465	668	48.6
3.616813294	669	48.68
3.62170088	670	48.75
3.616813294	671	48.83
3.626588465	672	48.9
3.62170088	673	48.98
3.611925709	674	49.05
3.611925709	675	49.13
3.607038123	676	49.13
3.611925709	677	49.22
3.611925709	678	49.28
3.616813294	679	49.37
3.616813294	680	49.43
3.616813294	681	49.52
3.616813294	682	49.58
3.616813294	683	49.67
3.616813294	684	49.73
3.611925709	685	49.82
3.611925709	686	49.88
3.611925709	687	49.97
3.616813294	688	50.03
3.611925709	689	50.12
3.611925709	690	50.18

3.611925709	691	50.27
3.611925709	692	50.33
3.607038123	693	50.42
3.611925709	694	50.48
3.607038123	695	50.57
3.607038123	696	50.57
3.611925709	697	50.65
3.602150538	698	50.72
3.611925709	699	50.8
3.607038123	700	50.87
3.602150538	701	50.95
3.611925709	702	51.02
3.607038123	703	51.1
3.607038123	704	51.17
3.602150538	705	51.25
3.597262952	706	51.32
3.607038123	707	51.4
3.607038123	708	51.47
3.611925709	709	51.55
3.602150538	710	51.63
3.602150538	711	51.7
3.597262952	712	51.78
3.607038123	713	51.85
3.602150538	714	51.93
3.607038123	715	52
3.597262952	716	52.08
3.602150538	717	52.15
3.607038123	718	52.23
3.602150538	719	52.3
3.597262952	720	52.38
3.597262952	721	52.45
3.592375367	722	52.47

3.602150538	723	52.53
3.602150538	724	52.62
3.597262952	725	52.68
3.597262952	726	52.77
3.592375367	727	52.83
3.602150538	728	52.92
3.597262952	729	52.98
3.597262952	730	53.07
3.597262952	731	53.13
3.592375367	732	53.22
3.597262952	733	53.28
3.587487781	734	53.37
3.592375367	735	53.43
3.592375367	736	53.52
3.592375367	737	53.58
3.587487781	738	53.67
3.592375367	739	53.73
3.587487781	740	53.82
3.592375367	741	53.88
3.592375367	742	53.97
3.587487781	743	54.03
3.582600196	744	54.05
3.587487781	745	54.12
3.587487781	746	54.2
3.592375367	747	54.27
3.587487781	748	54.35
3.592375367	749	54.42
3.592375367	750	54.5
3.587487781	751	54.57
3.582600196	752	54.65
3.582600196	753	54.72
3.587487781	754	54.8

3.587487781	755	54.87
3.587487781	756	54.95
3.582600196	757	55.02
3.582600196	758	55.1
3.582600196	759	55.17
3.582600196	760	55.25
3.582600196	761	55.32
3.587487781	762	55.4
3.582600196	763	55.48
3.57771261	764	55.55
3.582600196	765	55.63
3.587487781	766	55.7
3.57771261	767	55.78
3.57771261	768	55.85
3.582600196	769	55.93
3.587487781	770	56
3.582600196	771	56.08
3.582600196	772	56.15
3.572825024	773	56.23
3.572825024	774	56.3
3.582600196	775	56.38
3.572825024	776	56.45
3.572825024	777	56.53
3.57771261	778	56.6
3.572825024	779	56.68
3.572825024	780	56.68
3.57771261	781	56.77
3.57771261	782	56.83
3.572825024	783	56.92
3.572825024	784	56.98
3.572825024	785	57.07
3.57771261	786	57.13

3.572825024	787	57.22
3.57771261	788	57.28
3.57771261	789	57.37
3.567937439	790	57.43
3.567937439	791	57.52
3.563049853	792	57.58
3.572825024	793	57.67
3.572825024	794	57.73
3.567937439	795	57.82
3.558162268	796	57.88
3.567937439	797	57.97
3.563049853	798	58.03
3.563049853	799	58.12
3.563049853	800	58.18
3.563049853	801	58.27
3.563049853	802	58.33
3.563049853	803	58.42
3.563049853	804	58.48
3.563049853	805	58.57
3.563049853	806	58.63
3.567937439	807	58.72
3.563049853	808	58.78
3.563049853	809	58.87
3.563049853	810	58.93
3.558162268	811	59.02
3.558162268	812	59.08
3.563049853	813	59.17
3.553274682	814	59.25
3.558162268	815	59.32
3.548387097	816	59.4
3.553274682	817	59.4
3.558162268	818	59.48

3.558162268	819	59.55
3.558162268	820	59.63
3.563049853	821	59.7
3.553274682	822	59.78
3.553274682	823	59.85
3.548387097	824	59.93
3.553274682	825	60
3.553274682	826	60.08
3.548387097	827	60.15
3.553274682	828	60.23
3.553274682	829	60.3
3.553274682	830	60.38
3.553274682	831	60.45
3.553274682	832	60.53
3.558162268	833	60.6
3.553274682	834	60.68
3.543499511	835	60.75
3.558162268	836	60.83
3.548387097	837	60.9
3.548387097	838	60.98
3.543499511	839	61.05
3.543499511	840	61.13
3.543499511	841	61.2
3.553274682	842	61.28
3.543499511	843	61.35
3.543499511	844	61.43
3.543499511	845	61.5
3.548387097	846	61.58
3.543499511	847	61.65
3.548387097	848	61.73
3.519061584	849	61.8
3.543499511	850	61.88

3.543499511	851	61.95
3.543499511	852	62.03
3.548387097	853	62.1
3.543499511	854	62.18
3.543499511	855	62.25
3.543499511	856	62.27
3.548387097	857	62.33
3.543499511	858	62.42
3.548387097	859	62.48
3.543499511	860	62.57
3.53372434	861	62.63
3.543499511	862	62.72
3.53372434	863	62.78
3.548387097	864	62.87
3.543499511	865	62.93
3.543499511	866	63.02
3.538611926	867	63.1
3.53372434	868	63.17
3.543499511	869	63.25
3.53372434	870	63.32
3.538611926	871	63.4
3.53372434	872	63.47
3.53372434	873	63.55
3.538611926	874	63.62
3.53372434	875	63.7
3.528836755	876	63.77
3.538611926	877	63.85
3.53372434	878	63.92
3.53372434	879	64
3.53372434	880	64.07
3.53372434	881	64.15
3.53372434	882	64.22

3.538611926	883	64.3
3.538611926	884	64.37
3.538611926	885	64.38
3.528836755	886	64.45
3.538611926	887	64.53
3.53372434	888	64.6
3.528836755	889	64.68
3.528836755	890	64.75
3.528836755	891	64.83
3.53372434	892	64.9
3.53372434	893	64.98
3.528836755	894	65.05
3.53372434	895	65.13
3.523949169	896	65.2
3.523949169	897	65.28
3.528836755	898	65.35
3.523949169	899	65.43
3.528836755	900	65.43
3.528836755	901	65.52
3.523949169	902	65.58
3.523949169	903	65.67
3.523949169	904	65.73
3.523949169	905	65.82
3.523949169	906	65.88
3.528836755	907	65.97
3.523949169	908	66.03
3.528836755	909	66.12
3.523949169	910	66.18
3.519061584	911	66.27
3.519061584	912	66.33
3.519061584	913	66.42
3.528836755	914	66.48

3.523949169	915	66.57
3.519061584	916	66.63
3.528836755	917	66.72
3.519061584	918	66.8
3.523949169	919	66.87
3.519061584	920	66.95
3.519061584	921	67.02
3.523949169	922	67.1
3.523949169	923	67.17
3.519061584	924	67.25
3.494623656	925	67.32
3.484848485	926	67.4
3.479960899	927	67.47
3.470185728	928	67.55
3.470185728	929	67.55
3.479960899	930	67.63
3.479960899	931	67.7
3.470185728	932	67.78
3.465298143	933	67.85
3.377321603	934	67.93