ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Módulo traductor de imágenes a voz para enseñanza de personas no videntes.

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telemática

Presentado por:

Carlos Javier Apolo Peralta

Marcelo Eduardo Romero Segura

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

Carlos Apolo

El presente proyecto se lo dedico a mis padres, pilares fundamentales en mi vida y a lo largo de toda mi carrera. A mis hermanas y hermanos quienes desde la distancia me dedican su tiempo y sus oraciones. A Yuly, mi compañera de vida, quien me acompaña y me cuida. A Dios, que me escucha siempre en mi soledad.

Marcelo Romero

Dedico este proyecto a mis padres, hermanos y tías. Su apoyo y amor inquebrantables han sido mi fuerza motriz. Sus sacrificios, orientación y aliento han hecho posible este viaje. Este proyecto es un símbolo de nuestro viaje compartido y de mi gratitud por todo lo que han hecho.

AGRADECIMIENTOS

Carlos Apolo

”¡gracias sean dadas a Dios, que nos da la victoria por nuestro Señor Jesucristo!”1Cor.15:57

Mi más sincero agradecimiento a: Mis padres, especialmente a Lucía, mi madre. Por su paciencia, esfuerzo y apoyo siempre incondicional. A Yuly, mi compañera de vida, por su motivación y apoyo en el desarrollo de este trabajo. A mis amigos y compañeros, por su apoyo y empeño. A mis profesores, que se han esforzado en compartir sus bastos conocimientos no solo en materia académica, sino también en experiencias de vida que aportaron a mi formación humana.

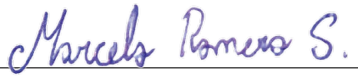
Marcelo Romero

Gracias de todo corazón a mis padres por su amor incondicional, su apoyo y sus ánimos a lo largo de mi trayectoria académica. Su fe inquebrantable en mí ha sido una fuente de fuerza e inspiración. También quiero dar las gracias a mis hermanos por su apoyo constante y por estar siempre a mi lado. Han sido mis mayores animadores y siempre me han empujado a luchar por la excelencia. Además, me gustaría extender mi más sincero agradecimiento a mis tías por haberme proporcionado un hogar durante mis primeros años de universidad. Su generosidad y hospitalidad han sido inestimables, y siempre les estaré agradecido.

DECLARACIÓN EXPRESA

”Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Carlos Javier Apolo Peralta y Marcelo Eduardo Romero Segura damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Carlos Apolo



Marcelo Romero

EVALUADORES

Firmado electrónicamente por:
IGNACIO MARIN GARCIA

Ignacio Marín

PROFESOR DE LA MATERIA

Firmado electrónicamente por:
RAYNER STALYN DURANGO ESPINOZA

Rayner Durango

PROFESOR TUTOR

RESUMEN

La normativa vigente en Ecuador indica que debe garantizarse educación de calidad para todos, incluyendo a las personas con discapacidad, adecuando los espacios educativos a sus necesidades [1]. En nuestro país, existe una carencia de herramientas tecnológicas que permitan dar apoyo en la educación de personas con capacidad visual disminuida. La solución que se propone consiste en un módulo de lectura para ayudar a los estudiantes con discapacidad visual en su educación. Para esto, como objetivo se plantea implementar un dispositivo prototipo de bajo costo empleando la tarjeta de desarrollo Raspberry Pi y tecnología de código abierto para Reconocimiento Óptico de Caracteres (OCR en inglés) y conversión Texto a Voz (TTS en inglés). El alto costo de las herramientas alternativas y la poca presencia en el mercado local presenta un reto para las instituciones educativas con presupuesto limitado, lo que expone la necesidad de desarrollar una solución más asequible. El prototipo fue construido usando una tarjeta de desarrollo Raspberry Pi y un módulo de cámara dispuesto sobre una plataforma para capturar imágenes del material didáctico. La lectura se llevó a cabo a través de un proceso de reconocimiento de caracteres y la posterior traducción del texto digitalizado a voz. El dispositivo además posee botones señalizados en Braille para controlar la narración del texto en audio. Las lecturas de las pruebas demuestran que es posible desarrollar dispositivos de bajo costo que realicen conversión de imágenes a voz con tecnología Open Source y obtener resultados con una precisión de al menos un 85% y un audio con voz emulada fácil de escuchar y comprender.

Palabras Clave: educación, no vidente, Raspberry, TTS, OCR, OpenSource.

ABSTRACT

Current regulations in Ecuador indicate that quality education must be guaranteed for all, including people with disabilities, adapting educational spaces to their needs. In our country, there is a lack of technological tools to support the education of people with visual impairment. The proposed solution consists of a reading module to help visually impaired students in their education. For this, the objective is to implement a low-cost prototype device using the Raspberry Pi development board and open source OCR (acronym for Optical Character Recognition) and TTS (acronym for Text-To-Speech) technology. The high cost of alternative tools and the limited presence in the local market presents a challenge for educational institutions with limited budgets, which exposes the need to develop a more affordable solution. The prototype was built using a Raspberry Pi development board and a camera module arranged on a platform to capture images of the teaching material. The reading was carried out through a process of character recognition and subsequent translation of the digitized text to speech. The device also has Braille buttons to control the audio narration of the text. The test readings demonstrate that it is possible to develop low-cost devices that perform image-to-speech conversion with Open Source technology and obtain results with an accuracy of at least 85% and a voice-emulated audio that is easy to hear and understand.

Keywords: education, visual-impaired, Raspberry, TTS, OCR, OpenSource.

ÍNDICE GENERAL

Glosario

Frame Cuadro, traducido del inglés. Unidad mínima posible de registrar para conformación de un vídeo.

Machine Learning El aprendizaje automático (Machine Learning, ML) es un subcampo de la inteligencia artificial (IA) que implica el desarrollo de algoritmos y modelos que permiten a los ordenadores aprender de los datos sin ser programados explícitamente..

OpenSource Software cuyo código fuente original está disponible gratuitamente y puede redistribuirse y modificarse.

Pin Puerto o terminal pequeño de conexión en un dispositivo electrónico. Plural: Pines.

Thresholding Umbralización, traducido del inglés. Proceso que consiste en segmentar imágenes con el fin de diferenciar un objeto de interés del fondo de una imagen.

Wi-Fi Tecnología de redes que permite conectar dispositivos de forma inalámbrica.

Acrónimos

API Interfaz de Programación de Aplicaciones (del inglés Application Programming Interfaces).

ESPOL Escuela Superior Politécnica del Litoral.

FIEC Facultad de Ingeniería en Electricidad y Computación.

GPIO Puerto de Entrada y Salida para Propósito General (del inglés General Purpose Input/Output).

GPS Sistema de Posicionamiento Global (del inglés Global Positioning System).

gTTS Texto A Voz de Google(del inglés Google Text-to-Speech).

HTTP Protocolo de Transferencia de Hipertexto (del inglés Hypertext Transfer Protocol).

OCR Reconocimiento Oṕtico de Carateres (del inglés Optical Character Recognition).

OS Sistema Operativo (del inglés Operating System).

QR código de Respuesta Rápida (del inglés Quick Response code).

TTS Texto A Voz (del inglés Text-to-Speech).

USD Dólar Estadounidense (del inglés United States Dollar).

SIMBOLOGÍA

mm	Milímetro
cm	Centímetro
v	Voltios
b	bytes
s	segundos
GB	Gigabyte
p	Píxeles
MP	Megapíxeles
ppp	Píxeles por pulgada

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN

En Ecuador existen alrededor de 54.328 personas con algún grado de discapacidad visual y de ellos al menos un 10% están en edad escolar de primaria o secundaria, según cifras oficiales a enero de 2022 [4]. Para respetar el derecho a acceder a educación de calidad amparado por la legislación vigente en el país, se debería garantizar que al menos en las unidades educativas fiscales, existan las condiciones necesarias para una educación que incluya a personas con algún tipo de discapacidad, entre ellas la discapacidad visual. Actualmente se tienen centros educativos especializados para educación a personas no videntes, sin embargo, en la mayor parte de casos no se cuenta con herramientas suficientes o adecuadas para personas no videntes, por lo que se hace necesario que más centros educativos no especializados puedan equiparse con tecnología para brindar apoyo a estas personas.

La oferta de tecnologías de apoyo a la educación inclusiva llega a ser escasa, debido a los altos costos que podrían representar, por ello, existe la necesidad de aportar a dicho mercado herramientas o dispositivos que no sólo se ajusten mejor a la capacidad adquisitiva de las instituciones educativas, sino que, además, se puedan fácilmente adaptar a cualquier ambiente educativo: escuelas, colegios o incluso en el hogar.

Por esta razón, se plantea la investigación de tecnologías que permitan la creación de un módulo de lectura que facilite la integración de personas no videntes en las instituciones regulares del país. Este dispositivo debe tener un bajo costo de adquisición y, poder incorporarse al sistema de enseñanza actual con pocos requisitos de implementación. El módulo de lectura debe ser capaz de obtener el texto del material educativo convencional tales como páginas de libros, cuadernos, carteles y luego reproducirlo en formato de audio.

1.1 Descripción del problema

La educación inclusiva implica el rediseño no solo de las metodologías de enseñanza, sino que, además, el uso de material pedagógico adecuado para cada tipo de necesidad del estudiante. El material dispuesto por el ministerio de educación son guías para el desarrollo de implementos, material didáctico y acceso a preparación para mejorar su labor pedagógica, sin embargo, estos no abarcan con atención a la formación de personas con necesidades educativas especiales, como es el caso de las personas con discapacidad visual.

Una de las herramientas principales, y en la mayoría de casos la única, es la máquina de escribir ”*Braille Perkins*”, que está diseñada para ser un sistema de aprendizaje de lectura y escritura utilizando teclas con el sistema de puntos del lenguaje Braille. A pesar de que se constituye como una herramienta básica muy importante, no se cuenta con material didáctico adicional que permita un desarrollo en otros conocimientos necesarios.

1.2 Justificación

A pesar de que en el mercado existen varias propuestas ofertadas como herramientas para brindar apoyo a personas no videntes, el costo de las mismas incurre en que sean más difíciles de adquirir sobre todo para instituciones educativas que no cuenten con el presupuesto suficiente como para cubrir los gastos de adquisición del al menos uno de estos dispositivos. Esto hace que el desarrollo de herramientas tecnológicas más asequibles para adecuar ambientes educativos sea urgente, en tanto que la necesidad de brindar una educación digna y de calidad a personas con necesidades educativas especiales sea respondida de mejor manera.

1.3 Objetivos

Dada a conocer la problemática respecto a la falta de herramientas tecnológicas para enseñanza a personas no videntes, el objetivo general que se plantea para este proyecto es implementar un dispositivo prototipo de bajo costo empleando la tarjeta de desarrollo Raspberry Pi y tecnología de código abierto OCR (Reconocimiento Óptico de Caracteres, en español) y TTS (Texto a Voz, en español) con el fin de traducir imágenes con texto en audio para educación de personas no videntes.

Se dispone de tres objetivos específicos que garantizan la integración de tecnología tanto hardware como software para cumplir con este propósito planteado.

- Integrar un módulo de cámara en una tarjeta de desarrollo Raspberry Pi para capturar imágenes que contengan texto utilizando técnicas de procesamiento de imágenes .
- Implementar tecnología OCR utilizando librerías OpenSource en Python, basadas en Machine Learning y redes neuronales para obtener el texto contenido en las imágenes capturadas por el dispositivo.
- Emplear tecnología TTS para reproducir en altavoz el texto obtenido de las imágenes capturadas por el dispositivo.

1.4 Estado del arte

Hoy en día existen varias alternativas comerciales que se ofertan como soluciones para escolaridad de personas con capacidad visual disminuida. Lo más común es encontrarse con la máquina de escribir *Braille Perkins*, diseñada para aprendizaje de escritura y lectura en lenguaje Braille.[5]

En Ecuador, se han presentado varios proyectos que buscan también aportar con herramientas para la escolaridad de personas no videntes, que tienen un enfoque distinto a la lectura o escritura. Por ejemplo, el trabajo de Chica-Moreta [6] es un proyecto diseñado para niños que emplea estímulos sensoriales para la identificación de partes del cuerpo humano. Por otra parte, se presentan propuestas en las cuales se estudia el diseño y desarrollo de materiales pedagógicos complementarios llamados Mood-Boards para la enseñanza a personas no videntes en materia de biología. [7]

Las tecnologías actuales para la escolaridad de personas no videntes, están enfocadas mayormente a la enseñanza de lectura y escritura del sistema Braille o la lectura en altavoz de textos digitales. A continuación, se detallan los más importantes.

- **Pantalla Braille**

  Las pantallas braille son dispositivos que convierten texto digital al sistema de lectura braille a través de Pines organizados en matrices en una pantalla que se actualiza constantemente conforme se navega por el contenido, también cuentan con modo escritura para tomar notas y guardarlas en la memoria del dispositivo. Actualmente las pantallas braille pueden conectarse a otros dispositivos como PCs, tablets o smartphones y permiten realizar funciones de navegación entre aplicaciones o en internet. Estos dispositivos presentan limitaciones en cuanto a capacidad de escritura y lectura puesto que solo se muestra una línea de texto, además, son altamente costosos con precios que exceden los 2000 USD dependiendo del número de celdas que poseen y de las funciones adicionales. Las dos pantallas Braille más baratas en el mercado son el Braille Me y el Orbit Reader 20, ambas con un precio aproximado de 500 USD. [8]

- **Audiolibros para enseñanza**

  Son libros producidos en formato de audio que funcionan como material educacional para personas con discapacidad visual. Estos libros se encuentran disponibles en formato mp3, CDs y en plataformas web de pago o gratuitas. Al estar en formato de audio facilita la toma de apuntes puesto que se puede detener, avanzar, retroceder o repetir un fragmento del libro. Las plataformas de audiolibros disponen de una gran cantidad de libros y cursos, los cuales se pueden acceder ya sea mediante compra directa, realizando una suscripción de pago o gratuita en la página o simplemente descargándolo sin costo alguno dependiendo del artículo y el sitio. Entre los sitios web de audiolibros más populares se encuentran Audible, Scribd, Google AudioBooks, Librivox y Audiobooks.com. El precio de la suscripción a una plataforma de audiolibro suele variar de entre 6 a 15 USD. [9]

- **Aplicaciones móviles con OCR**

  Son aplicaciones que utilizan la cámara del dispositivo móvil para tomar fotos y que mediante OCR(acrónimo de Optical Character Recognition), que es una tecnología que permite a las computadoras leer y reconocer texto impreso en una imagen o un documento escaneado, reconoce y digitaliza los caracteres en la imagen para luego reproducirlos en el altavoz. Cuentan con un asistente de voz que se encarga de guiar el proceso de captura de la foto asegurándose de que la imagen tenga un correcto ángulo y que se encuentre dentro del marco requerido. Algunas de estas aplicaciones poseen funcionalidades adicionales de reconocimiento de objetos y de lugares e incluso asistencia guiada usando GPS. Ejemplos de aplicaciones actuales son Lookout by Google, Voice OCR, Natural Readers y Voice Dream Scanner, cada una ofreciendo distintas características como reconocimiento de texto en tiempo real, sin conexión a internet o reconocimiento de texto manuscrito.[10]

- **Escáner OCR**

  Dispositivo físico diseñado para realizar el reconocimiento óptico de caracteres (OCR) en texto impreso o manuscrito. Suele constar de varios componentes, como una fuente de luz, una lente óptica, un sensor de imagen y una unidad de procesamiento. La fuente de luz ilumina el documento y la lente óptica enfoca la luz sobre el sensor de imagen, que captura una representación electrónica del documento. A continuación, la unidad de procesamiento procesa los datos de la

imagen para mejorar su calidad, aplicar algoritmos de OCR para reconocer los caracteres y convertirlos en texto legible por máquina. A continuación, la salida suele procesarse de nuevo para corregir cualquier error o imprecisión que pueda haberse introducido durante el proceso de OCR. El hardware de un escáner de OCR puede variar desde dispositivos autónomos hasta sistemas integrados que se incorporan a sistemas de procesamiento de documentos más grandes. El diseño del hardware de un escáner de OCR debe tener en cuenta factores como la resolución de la imagen, la calidad de la imagen, el tamaño del documento y la velocidad de procesamiento para garantizar unos resultados de OCR precisos y eficaces.[11]

1.5 Escenarios de la propuesta

El presente proyecto está enfocado a una fácil implementación, es decir, que no requiera mayor configuración o requisitos externos para su despliegue y uso. Así, se logrará la integración del dispositivo a cualquier ambiente escolar como parte de las herramientas tecnológicas que brinden apoyo en la educación de personas no videntes.

El escenario principal, serán las aulas o laboratorios de centros educativos, especialmente aquellos que cumplan con alguna o varias de las características que se mencionan a continuación:

- **Ubicación en zona rural**. - El acceso a estos centros educativos especializados puede representar un problema de movilidad para las personas con discapacidad visual que no viven cerca de los mismos.
- **Escasos recursos**. - El centro educativo no cuenta con los recursos suficientes para adquisición de servicios o equipos tecnológicos que permitan la integración de personas con discapacidad visual.
- **Disponibilidad de soluciones alternativas**. - Muchas de las herramientas tecnológicas mencionadas en la Sección 1.4 no están disponibles en el mercado nacional, lo que implica que deben ser adquiridos en el extranjero y asumir los costos de importación.

Finalmente, otro escenario posible es el hogar de la persona con discapacidad visual, en el cual se dará un uso personal del dispositivo como herramienta de lectura.

CAPÍTULO 2

2. METODOLOGÍA

En este capítulo se describirán los componentes, su utilidad y cómo se integran para cumplir con el objetivo planteado en este trabajo. Los componentes comprenden tanto el hardware como el software a emplear. El funcionamiento del prototipo precisa de la integración de una base o plataforma de dimensiones específicas con un módulo de cámara conectado a una tarjeta de desarrollo, en el caso de este proyecto se utilizó una Raspberry Pi. Sobre la base se deben colocar hojas o materiales que contengan texto, la cámara está dispuesta sobre un soporte a una altura adecuada para la correcta captura de imágenes. Una vez obtenida la imagen se realiza la digitalización del texto detectado.

2.1 Métodos y técnicas de la investigación

Para la búsqueda de información técnica, trabajos similares e información útil respecto a la implementación de las tecnologías y componentes de hardware a utilizar, se ha decidido emplear el método científico.

Para la recopilación de información relevante que permitió establecer parámetros de uso, patrones de diseño u otras consideraciones que deben tomarse en cuenta para el desarrollo de tecnologías para personas con discapacidad visual [12], se tiene la técnica experimental. De esta manera, se realizaron pruebas y simulaciones de funcionamiento del dispositivo prototipo y las diferentes mecánicas asociadas a éste.

Para evaluar el correcto del funcionamiento del dispositivo respecto a sus componentes eléctricos y electrónicos, se emplea la técnica de medición, a través de herramientas como voltímetros y amperímetros.

2.2 Diseño de la solución

Con base en la investigación realizada en el Capítulo 1 sobre las herramientas para OCR o *Reconocimiento Óptico de Caracteres* (del inglés Optical Character Recognition) y asistentes TTS existentes en el mercado, se ha establecido que la forma más eficaz para facilitar el uso del dispositivo es emplear una estructura base para la colocación de las páginas que contengan texto en un tamaño máximo de hoja correspondiente al formato A4, es decir 21,0cm por 29,7cm. Esto permite que las imágenes sean capturadas de manera adecuada respecto a la distancia y posicionamiento de lo que se requiera leer. De la misma manera, con el fin de responder a las necesidades de una persona no vidente, se dispone de botones señalizados en alto relieve con lenguaje Braille para un control apropiado de las funciones del dispositivo.

El funcionamiento básico del dispositivo prototipo comprende tres (3) etapas principales. Cada una está diseñada para aportar de manera secuencial al funcionamiento de todo el dispositivo.

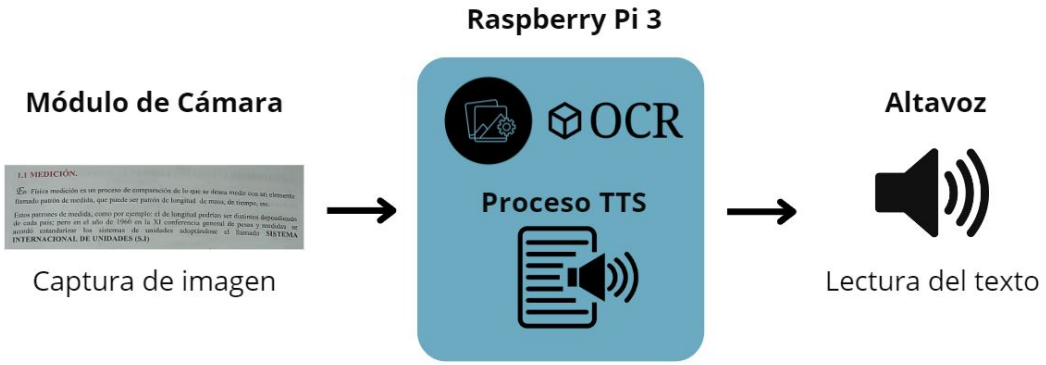


Figura 2.1: **Arquitectura propuesta** *Se muestran los procesos principales y dispositivos encargados de llevarlos a cabo.*

Como se muestra en la Figura 2.1, el módulo de cámara se encarga de fotografiar el material de lectura, luego, empleando como unidad de cómputo la tarjeta de desarrollo Raspberry Pi, la imagen será procesada por el software desarrollado en Python para reconocer y digitalizar los caracteres y finalmente, generar un archivo de audio con el texto narrado para ser escuchado en el periférico de audio conectado por el usuario.

2.3 Selección de hardware

El criterio principal de selección de los componentes de hardware, es que en la implementación del prototipo se minimice el costo de sus elementos. Por ello, se optó por el uso de una tarjeta de desarrollo que cumpla con este requisito y a su vez posea una capacidad de procesamiento adecuada para el desarrollo de las tareas necesarias. Para almacenar el sistema operativo, el software y los datos generados, se añade una unidad de almacenamiento externa. Además, se seleccionaron módulos que no supongan problemas de compatibilidad en la integración con la tarjeta de desarrollo y, que cuenten con la documentación pertinente a su funcionamiento.

- **Raspberry Pi.** La tarjeta de desarrollo Raspberry Pi es un ordenador con placa simple, de bajo costo y con dimensiones pequeñas. Utiliza un sistema operativo basado en Linux llamado Raspberry Pi OS. El modelo 3B, es corresponde a la tercera generación de estas placas.[13] Gracias a sus pines GPIO o pines de entrada/salida para propósito general, permite la conexión de los diferentes elementos que se consideraron para este proyecto.

  En cuanto a los requerimientos para este trabajo, un ordenador de escritorio o portátil, tiene capacidades mucho mayores de procesamiento y almacenamiento. Sin embargo, representaría una inversión importante en adquisición de hardware y software, y no sería un uso eficiente de las capacidades de estos dispositivos, siendo que la tarea que deben cumplir no representa una carga operativa que justifique el uso de alguna de estas opciones. Por otra parte, es poco práctico ya que los ordenadores no están especialmente diseñados para personas con discapacidad visual y habría que implementar mecanismos adicionales para lograr esta adaptación. También, se necesitaría más espacio de manera que se pueda colocar tanto el ordenador como la base para receptar las hojas o implementos que serán fotografiados.

  Por los motivos expuestos anteriormente, se seleccionó la tarjeta de desarrollo Raspberry Pi modelo 3B, dado que cumple con los requerimientos de costo, capacidad de operación suficiente e integración con otras tecnologías. Además, al ser un componente de dimensiones pequeñas, lo hace idóneo para su adaptación

con la base o carcasa.

- **Cámara Pi** Es el módulo de cámara oficial de Raspberry Pi el cual permite tomar fotos y grabar videos en alta definición. Existen tres versiones diferenciadas en aspectos de hardware tales como el tamaño, resolución, modos de video, tamaño de píxel y sensor. La cámara cuenta con un cable flex que se conecta directamente en un puerto llamado CAMERA en la Raspberry Pi. El módulo a utilizar para el prototipo es la Pi Camera v1.3, la cual tiene un sensor de 5MP y enfoque fijo, capaz de capturar imágenes estáticas con una resolución de 2592 x 1994 píxeles. [14]

  Un dispositivo alternativo al módulo Pi Camera sería una cámara web, sin embargo, aquellas con precio similar son de peor calidad en cuanto a la resolución de las imágenes estáticas. Además, las imágenes capturadas presentan el efecto ojo de pez, el cual hace que la imagen presenten curvatura en los bordes, acentuándose más ante planos cercanos al foco, esto se debe a que este tipo de cámaras están diseñados para capturar escenarios amplios para videoconferencia.

- **Memoria microSD.**

  La memoria es necesaria para ejecutar el sistema operativo en la Raspberry Pi. Todas las unidades Raspberry Pi traen de fábrica una ranura para tarjetas de memoria SD o microSD en caso del modelo elegido. La capacidad de memoria requerida varía desde 8GB hasta 32GB. [15]Para este prototipo se utiliza una tarjeta microSD de 16GB, la cual servirá también para alojar los datos de las imágenes capturadas y el texto extraído.

2.4 Selección de software

El uso del software necesario no debe agregar costos monetarios al proyecto. Por lo tanto, como se plantea en el objetivo del mismo, se utilizan software y librerías OpenSource. A la vez, estos deben operar en un procesador con arquitectura de 32-bits dentro del sistema operativo Raspberry Pi.

- **Raspberry OS.**

  Es un sistema operativo de código abierto basado en Linux-Debian y diseñado específicamente para funcionar con el hardware Raspberry Pi. Actualmente se pueden encontrar varias versiones del mismo con diferencia en su arquitectura, 32 o 64 bits dependiendo del hardware donde requiera ser instalado. [16] Es importante mencionar esto, ya que algunos paquetes necesarios para la instalación de librerías, utilidades y herramientas necesarias como OpenCV, sólo son compatibles con un sistema operativo de 32 bits. Por ello se seleccionó para minimizar cualquier incidente o retraso en el desarrollo del proyecto aun pudiendo existir soluciones más eficientes

- **Python.**

  Es un lenguaje de programación interpretado que permite implementar paradigmas de programación orientada a objetos, programación estructurada y programación funcional [17]. En este proyecto el paradigma es el de programación estructurada, dado que solo se recurre a estructuras de control de secuencia, selección e iteración y se codificaron funciones que son invocadas de forma secuencial. Este lenguaje permite el uso de las librerías necesarias tanto para el reconocimiento de caracteres, así como para la conversión de este texto en un formato audible a través del altavoz del dispositivo.

  El uso de este lenguaje de programación es crucial para el presente proyecto, ya que la mayor parte del código necesario está descrito en el mismo y es posible importarlo minimizando riesgos y tiempos de desarrollo.

- **Tesseract.**

  Tesseract es un motor de reconocimiento óptico de caracteres desarrollado bajo licencia por la marca Hewlett-Packard en 1980 actualmente apoyado por Google.[18] Desde su versión 4 integra redes neuronales con algoritmos pre-entrenados de DeepLearning. Este software contiene librerías con las instrucciones necesarias para efectuar el reconocimiento de caracteres en imágenes que contienen texto y cuenta con extensa documentación, ejemplos y aplicaciones similares que agilizan su implementación.

  Los productos similares ofertados por los principales proveedores de servicios en la nube como Google con AutoVision ML y AWS con Textract, poseen un costo mensual por uso lo que incrementa los costos de desarrollo y producción. Por lo tanto, se escogió este software en el desarrollo del presente proyecto dada su facilidad de implementación e integración con otras herramientas de procesamiento de imágenes como OpenCV, y, permite eliminar costos externos ya que no requiere pago por uso.

  La Figura 2.2 presenta las etapas en el proceso de reconocimiento de caracteres que ejecuta Tesseract. En el paso de binarización la imagen digital se convierte en una imagen en escala de grises sin afectar las propiedades de la imagen, este resultado se procesa por software para analizar su estructura: fondo, texto, gráficos o separación de columnas de texto. Una vez que estos componentes se identifican, Tesseract procede a delimitarlos por contorno, ya sea palabras, líneas o párrafos y asignarles un orden para luego iterar sobre estos segmentos delimitados y efectuar el reconocimiento de texto. El resultado es un texto digitalizado que se puede editar y guardar en un archivo de texto simple.

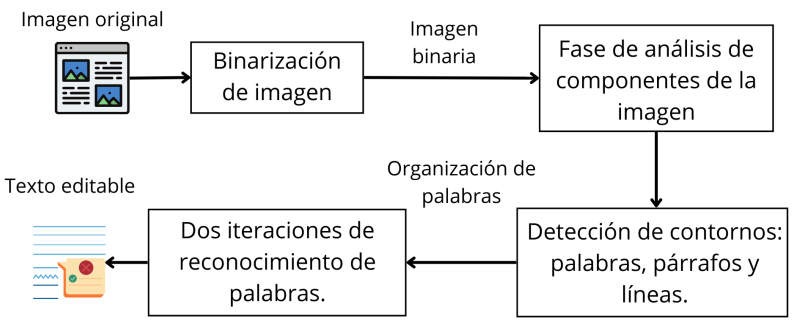


Figura 2.2: **Funcionamiento de Tesseract para OCR**. [2]

- **OpenCV.**

  Es una librería de código abierto que dispone de algoritmos de visión por computador que sirven como marco de trabajo para el desarrollo de aplicaciones enfocadas al procesamiento y análisis de imágenes.[19] Entre las funcionalidades que ofrece se tiene la segmentación y reconocimiento de objetos, reconocimiento de gestos, seguimiento de movimiento en tiempo real, calibración de cámara, entre otras. En este proyecto hace uso de las funciones de enmarcado de objetos o elementos para generar contornos y bloques que permitan la diferenciación de caracteres en las imágenes capturadas.[20] OpenCV cuenta con funciones para el preprocesamiento de imágenes tales como escalado de grises, umbralización, reducción de ruido, entre otras, a su vez permite la integración con otras librerías OCR, como TesseractOCR. Por estas razones se utiliza en este trabajo como herramienta para preprocesar las imágenes capturadas por el dispositivo. De esta manera, le aporta a Tesseract imágenes con mejor calidad permitiendo mejorar la precisión del proceso OCR.

- **Librería gTTS.**

  gTTS es una librería de código abierto para Python, cuya función es operar con el API Google Translate y generar archivos de audio en formato mp3 a partir de una cadena de texto. En el caso de este proyecto, la cadena de caracteres es extraída de las imágenes tomadas por el dispositivo. Se puede utilizar con Python mediante varias funciones predefinidas que no requieren mayor intervención, además, puede ser usada ejecutando comandos por consola. El uso del API Google Translate no tiene ningún costo y puede emular la voz en varios idiomas con sus respectivas variantes.[21]

- **API de Cloud Vision**

  Cloud Vision es un conjunto de herramientas de software relacionadas con temas de visión artificial, desarrolladas por Google Cloud. Entre algunas de las funciones más destacadas se encuentra la detección de objetos, detección de rostros, contenido explícito, etc. La función que ha sido seleccionada para este proyecto es la de reconocimiento de texto. Para este fin, Cloud Vision dispone de un API que es compatible con Python y posee funciones prácticas para ejecutar OCR. Entre las ventajas se pueden mencionar que omite resultados con errores, caracteres

o palabras que carezcan de cualquier sentido, evitando así que estos incidan negativamente en la composición del texto y posteriormente en la generación del audio. También, tiene la capacidad de corregir segmentos de la imagen que presenten curvaturas logrando obtener con mucha precisión el texto de estas partes. [22] El costo de este servicio por detección de etiqueta, texto o texto en documentos tiene un valor de $1,50 cada bloque de 1000 unidades, siendo las primeras 1000 detecciones gratuitas. [23]

La tabla 2.1, detalla la versión de software que se empleó en este proyecto tanto del sistema operativo para la Raspberry, el lenguaje Python para la implementación del código, y las librerías para procesamiento de las imágenes, digitalización del texto y lectura con voz emulada.

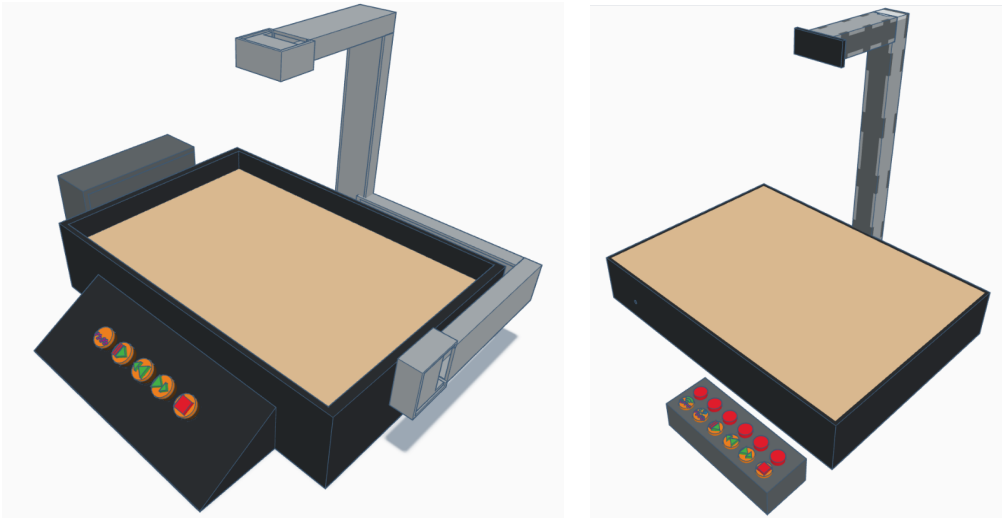
Tabla 2.1: **Especificación de la versión de los componentes de software**

Software	Raspberry OS	Python	Tesseract	Tesseract OCR	OpenCV	Google TTS	Cloud Vision API
Versión	5.15, 32-bit	3.9	4.1.1	0.0.1	4.5.5	2.12.3	v1p4 beta1

2.5 Diseño de dispositivo

De acuerdo con lo establecido al inicio de esta sección, se planteó un modelo tridimensional del dispositivo. En la Figura 2.3 se observan los modelos tridimensionales propuestos para el prototipo. Para ambas propuestas se ha considerado que se ajuste a las medidas de una lámina tamaño A4, esto es 21,0cm de ancho y 30,0cm de largo para la superficie donde será colocada la hoja que contenga el texto que se requiera leer. En el caso del modelo A se dispone la cámara Pi al interior de una carcasa a una altura de 30,0cm sobre el centro de la superficie para las hojas. La carcasa está adherida a un brazo movible, que puede girar hacia un costado, lo que permitirá que el dispositivo pueda ser almacenado con mayor facilidad. La tarjeta de desarrollo se encuentra al interior de una carcasa al costado izquierdo, dispuesta de manera que sus puertos de red, carga eléctrica y salida de audio de 3,5mm sean fáciles de encontrar y usar. Se incluyen también

botones para iniciar el proceso de captura de la imagen y para el control de la reproducción del audio una vez que la lectura se inicie.



(a) Modelo A (b) Modelo B

Figura 2.3: **Modelos tridimensionales propuestos para prototipo**

En el caso del modelo B, se dispuso la tarjeta de desarrollo Raspberry al interior de la base, colocada de tal forma que se pueda acceder a los puertos necesarios. A diferencia del modelo A, los botones de control están dispuestos en un módulo separado de la base principal y las marcas de indicación se colocan bajo los botones. Además, para este caso se consideró el diseño del brazo para una cámara Pi como dispositivo de captura de imágenes. Las demás características como medidas y función del brazo son iguales a la del modelo A.

En este proyecto se ha escogido el modelo B para el desarrollo del prototipo, ya que la superficie lisa facilita la colocación de libros y evita que las pastas se levanten de modo que el texto quede paralelo a la ubicación de la cámara. Además, es ocupa menos superficie al contener la Raspberry Pi colocada bajo la carcasa.

Botones de control

Los botones se utilizan para indicar que se utilice conexión a Internet por Wi-Fi, iniciar el proceso de captura de imagen y posteriormente controlar la reproducción del audio. Esto permitirá al usuario iniciar, pausar, avanzar, retroceder y detener la reproducción del audio según convenga. Estos botones están dispuestos en un circuito colocado al interior de la carcasa del control y está conectado a la tarjeta de desarrollo mediante cables.

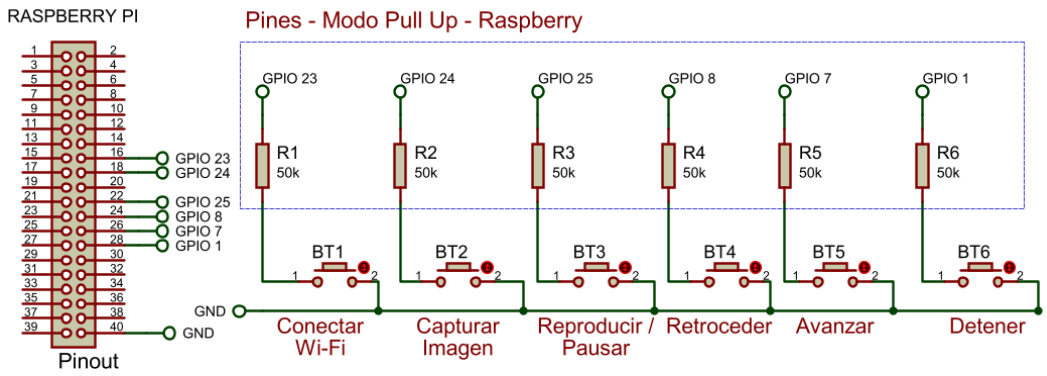


Figura 2.4: **Diagrama de circuito para botones de control**

En la Figura 2.4, se muestra el diseño esquemático del circuito y cómo está conectado a la tarjeta según la disposición de pines GPIO.

Operación

El dispositivo, requiere que el usuario sea asistido en dos pasos previos para poder empezar a leer. Estos pasos son el encendido y la conexión a Internet, que es un requisito imprescindible. Para encender el equipo se necesita conectarlo a una toma de corriente normal de 110v y presionar el interruptor colocado en el cable. Una vez encendido, el equipo indicará que está preparado para conectarse a Internet, para lo cual se le debe conectar un cable de red en el puerto correspondiente, o en su defecto, utilizar una red Wi-Fi. Para la conexión inalámbrica se necesita un código QR (acrónimo de Quick Response), que es un tipo de código de barras bidimensional que se puede escanear con una cámara o un lector de códigos QR para acceder a información como las credenciales de acceso de una red en este caso; este código deberá generarse con la ayuda de una persona que asista al usuario. Una vez que se obtenga este código se lo debe colocar en la superficie de la carcasa bajo la cámara, luego el usuario debe oprimir el botón Wi-Fi

en el control para que la cámara se encienda, capture el código y se conecte a Internet, lo cual se indicará con un aviso audible.

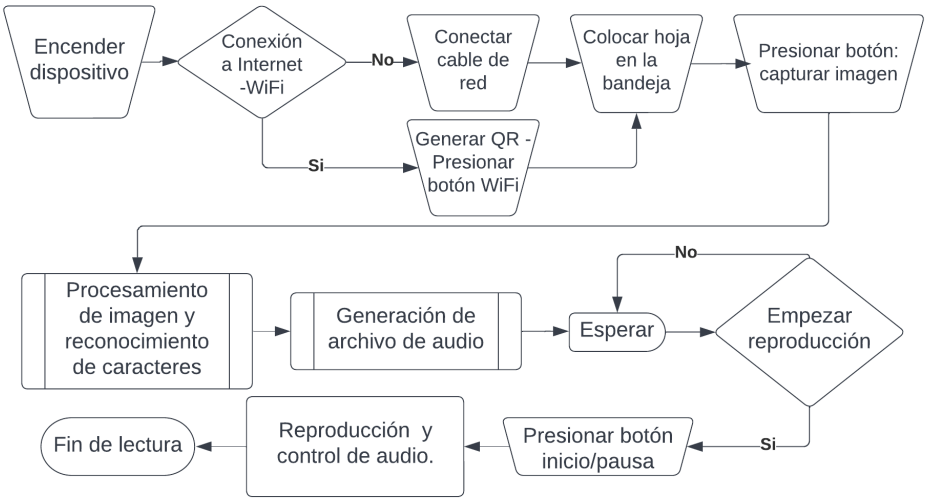


Figura 2.5: **Diagrama de funcionamiento** *Lógica de funcionamiento del dispositivo.*

En la Figura 2.5 se presenta el diagrama de operación del dispositivo, que muestra la secuencia de funcionamiento desde el encendido y las interacciones del usuario hasta la reproducción del audio. Una vez que el equipo esté encendido y conectado a Internet, estará listo para realizar los procesos de captura y lectura del texto, lo que se resume a tres (3) pasos en forma general. Primero, el usuario coloca la hoja, libro o cuaderno sobre la bandeja y presiona el botón de *captura de imagen*, que activará la cámara para que tome la fotografía. Esta imagen se preprocesa para dar paso al reconocimiento de caracteres, procesos en los cuales el usuario no interviene. Segundo, una vez que el texto esté listo y convertido en voz, el dispositivo emitirá una alerta audible indicando al usuario que puede iniciar la reproducción del audio oprimiendo el botón *Play*. Finalmente, el usuario podrá controlar la reproducción de la lectura en altavoz a través de los botones correspondientes dispuestos en el mando del dispositivo.

Lógica de programación

La forma en la que se lleva a cabo el proceso de capturar la imagen, extraer el texto y convertirlo en audio, sucede de manera secuencial, es decir, que cada proceso ocurre uno tras otro [24]. Para este caso, cada salida de un proceso es la entrada del siguiente.

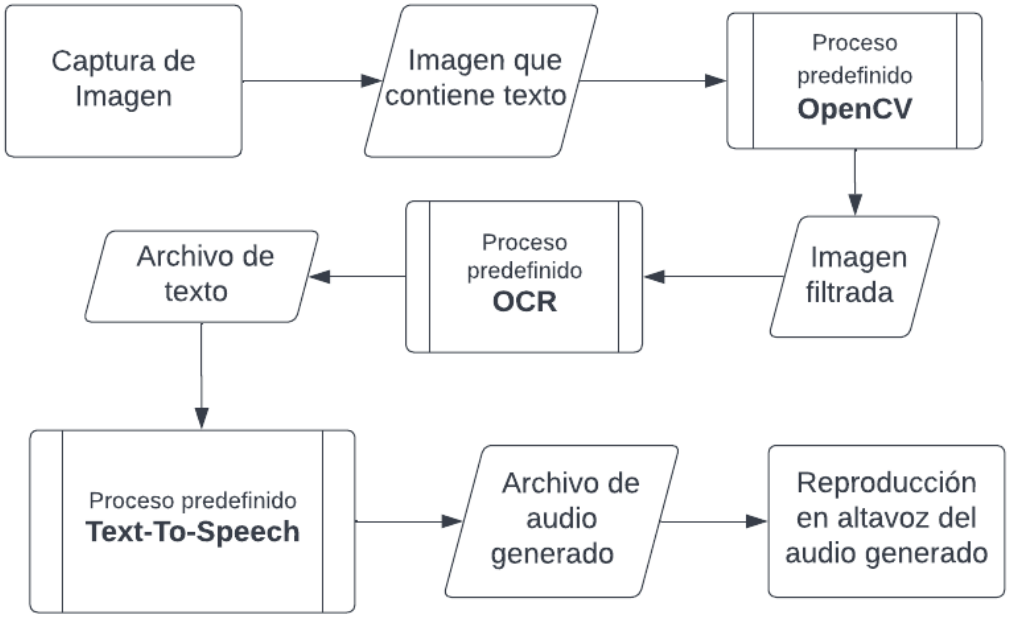


Figura 2.6: **Lógica de programación** *Lógica de la programación implementada en el dispositivo.*

En la Figura 2.6 se presenta el diagrama de flujo para el funcionamiento del software. Una vez la imagen es capturada, esta pasa a ser preprocesada por el módulo siguiente, donde se le aplican los filtros correspondientes. La imagen tratada se envía al proceso OCR efectuado por Tesseract y éste devuelve los caracteres, o líneas de texto encontradas en la imagen. El texto es guardado en un archivo de formato simple, dado que el entregado originalmente por el proceso OCR no es legible por la siguiente función en la secuencia, que es la conversión en audio. Finalmente, el proceso TTS se efectúa enviando línea por línea el texto del archivo guardado en el paso anterior, el API devuelve un flujo de bytes que van siendo almacenados en un mismo archivo de audio consolidado, que es el que será reproducido y escuchado en altavoz.

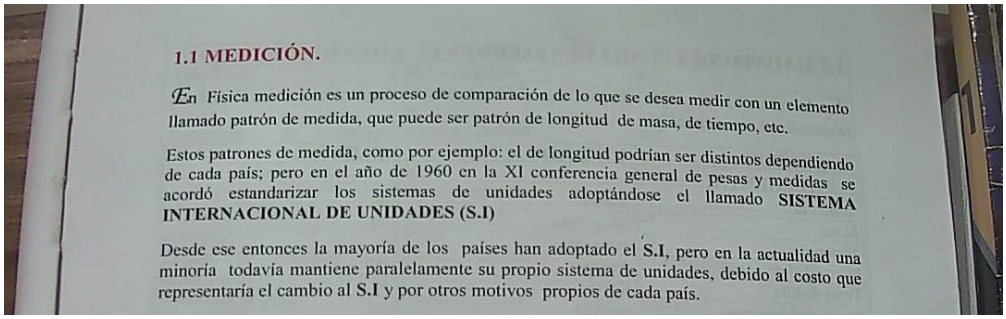
El preprocesamiento de las imágenes consiste en modificar sus propiedades, ya sea en su color, en sus píxeles, en su tamaño u orientación, entre otros . En la Tabla 2.2 se describe de manera general los filtros disponibles en OpenCV que facilitan el reconocimiento de caracteres en fotografías.

Tabla 2.2: **Filtros aplicados en el pre procesamiento de imágenes [3]**

Filtro	**Descripción**
Grayscale	Convierte la imagen a escala de grises.
Simple thresholding	Aplica un umbral basado en valores fijos.
Otsu's automatic thresholding	Aplica un umbral basado en valores determinados por un proceso de separación del primer plano y el fondo de a imagen.
Adaptive thresholding	Examina píxeles en sectores circundantes y establece un umbral adaptativo para cada sector.
Adaptive thresholding & Gaussian weighting	Aplica un umbral basado en valores computados por la comparación de pesos Gaussianos con una matriz de medias.

Thresholding, hace referencia a un proceso de umbralización, que consiste en separar regiones de la imagen en función de la intensidad de sus píxeles [3]. Los filtros de umbralización de la biblioteca OpenCV (a menudo denominada ”CV2” por el nombre del módulo que contiene las funciones de visión por ordenador) permiten establecer un valor umbral y, a continuación, separar los píxeles de una imagen en dos grupos: los que tienen intensidades superiores al valor umbral y los que tienen intensidades inferiores al valor umbral.

A continuación, en la Figura 2.7 se presenta como ejemplo una imagen antes (a) y después (b) de ser preprocesada.



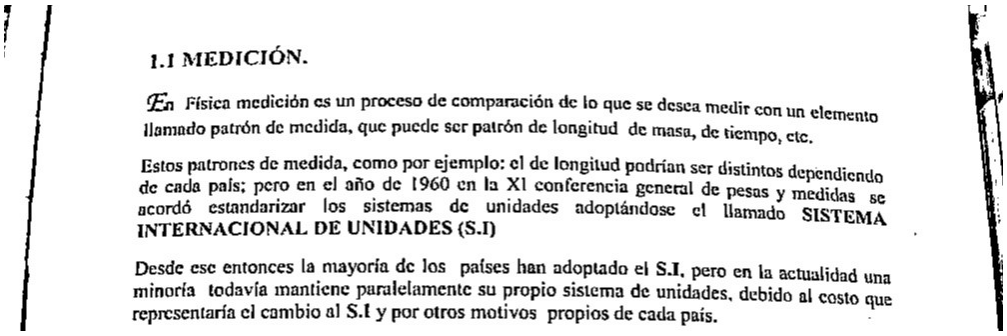
1.1 MEDICIÓN.

En Física medición es un proceso de comparación de lo que se desea medir con un elemento llamado patrón de medida, que puede ser patrón de longitud de masa, de tiempo, etc.

Estos patrones de medida, como por ejemplo: el de longitud podrían ser distintos dependiendo de cada país; pero en el año de 1960 en la XI conferencia general de pesas y medidas se acordó estandarizar los sistemas de unidades adoptándose el llamado **SISTEMA INTERNACIONAL DE UNIDADES (S.I)**

Desde ese entonces la mayoría de los países han adoptado el **S.I**, pero en la actualidad una minoría todavía mantiene paralelamente su propio sistema de unidades, debido al costo que representaría el cambio al **S.I** y por otros motivos propios de cada país.

(a) Recorte de imagen sin filtrar



1.1 MEDICIÓN.

En Física medición es un proceso de comparación de lo que se desea medir con un elemento llamado patrón de medida, que puede ser patrón de longitud de masa, de tiempo, etc.

Estos patrones de medida, como por ejemplo: el de longitud podrían ser distintos dependiendo de cada país; pero en el año de 1960 en la XI conferencia general de pesas y medidas se acordó estandarizar los sistemas de unidades adoptándose el llamado **SISTEMA INTERNACIONAL DE UNIDADES (S.I)**

Desde ese entonces la mayoría de los países han adoptado el **S.I**, pero en la actualidad una minoría todavía mantiene paralelamente su propio sistema de unidades, debido al costo que representaría el cambio al **S.I** y por otros motivos propios de cada país.

(b) Filtro aplicado: Adaptative Thresholding

Figura 2.7: **Recorte de imagen antes (a) y después (b) de preprocesamiento**

A esta imagen se le aplicó un filtro de Thresholding adaptativo, que, diferencia del Thresholding simple, que aplica un único valor de umbral a toda la imagen, el modo adaptativo calcula un umbral local para cada píxel en función de las intensidades de los píxeles circundantes. Este método es útil para imágenes con iluminación desigual o contraste variable. Esto facilita una clara diferenciación entre el fondo y las letras del texto, como se observa.

CAPÍTULO 3

3. RESULTADOS

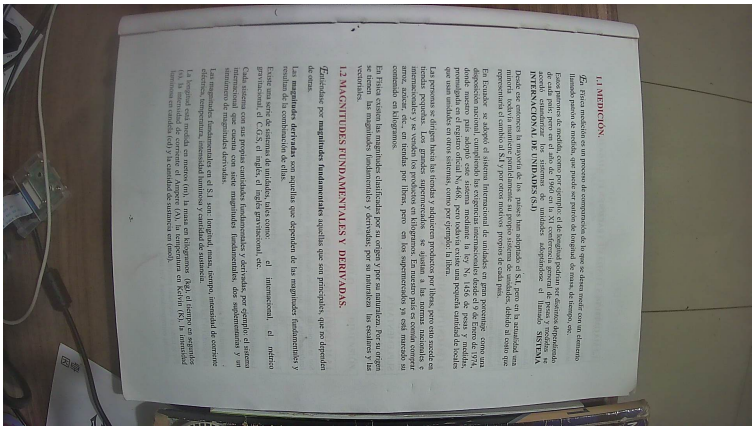
En este capítulo se muestran las pruebas de funcionamiento del dispositivo utilizando dos motores OCR y se detallan los parámetros para evaluar la precisión y tiempo de respuesta. Además, se presenta una comparación y análisis de los resultados obtenidos en las pruebas. Finalmente se describen los hallazgos generales respecto a los tres puntos importantes del proyecto: precisión en reconocimiento de caracteres, calidad de las imágenes y calidad del audio resultante.

3.1 Pruebas de funcionamiento

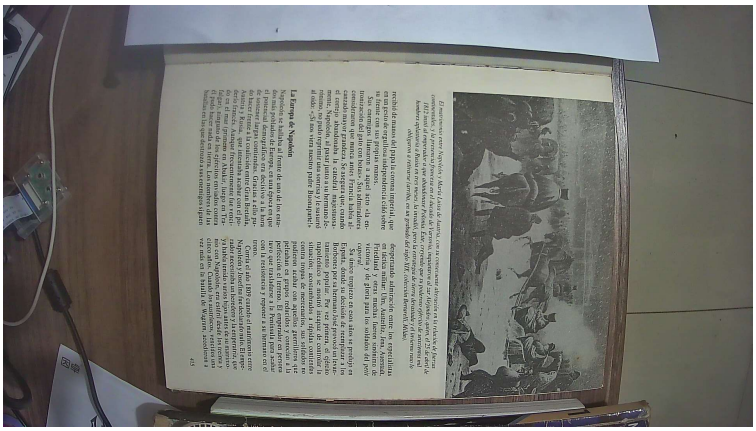
Las pruebas de funcionamiento del prototipo permitieron medir el tiempo de respuesta de todo el proceso y la precisión en la digitalización del texto. Para esto se efectuó la captura de varias imágenes bajo un ambiente controlado en un laboratorio de la FIEC en la ESPOL y se llevó a cabo la traducción de texto a voz usando las dos opciones de software presentadas en este trabajo para efectuar el reconocimiento de caracteres en imágenes. Se debe entender por ambiente controlado que tanto las condiciones de iluminación y la colocación del dispositivo son los adecuados para que las fotografías sean claras y no presenten sombras. Además, se procuró que las hojas o material a fotografiar presenten poca o nula curvatura de manera que esto no incida en mayor medida en los resultados esperados.

Este procedimiento consistió en la captura de cuatro imágenes distintas, las cuales presentan características específicas como combinación de texto e imágenes, variedad en cantidad de texto y segmentación del texto, esto último, para observar el comportamiento del proceso OCR al enfrentarse a capturas con una sola columna, dos

o una combinación de ambas. Todas las imágenes recibieron previamente el mismo tratamiento con filtros para mejorar su calidad y luego efectuar el reconocimiento de caracteres y la conversión en audio. En la Figura 3.1 se puede observar dos de las capturas sin preprocesar efectuadas con el dispositivo.



(a) Página de un texto de física



(b) Página de un libro de historia

Figura 3.1: **Capturas de prueba**

En las muestras presentadas en la Figura 3.1(a) se tiene una página compuesta por una única columna de texto sin imágenes, a diferencia de la Figura 3.1(b) donde se tiene una imagen y texto distribuido en varias columnas o segmentos. Estas mediciones se presentarán en las subsecciones posteriores.

Como parámetros de interés, para cada captura se ha considerado la cantidad de columnas que presenta el texto, la cantidad de párrafos y la cantidad de palabras presentes en el texto original. Cabe destacar que dentro del conteo de palabras también se consideran números y siglas.

En la tabla 3.1 se describe de manera general la composición de las imágenes de prueba tomadas por el dispositivo.

Tabla 3.1: **Composición de imágenes para prueba de funcionamiento**

Parámetro	**Imagen 1**	**Imagen 2**	**Imagen 3**	**Imagen 4**
Columnas de texto	1	Múltiple	1	Múltiple
Párrafos	14	8	12	16
Palabras en texto original	432	460	459	519

El valor *Múltiple* hace referencia a una captura que presenta una segmentación variada, es decir, contiene partes con una columna y otras con dos columnas de texto separadas. Es necesario aclararlo, debido a que esto incide directamente en la calidad de los resultados. Ahora, el parámetro párrafos hace referencia a la agrupación de una o varias líneas de texto, así, los títulos también se consideran como párrafos. La cantidad de palabras en cada captura se obtuvo mediante conteo manual.

Para evaluar la precisión, se compararon los resultados obtenidos utilizando tanto Tesseract como Cloud Vision, y, combinando el uso de dos filtros: Escala de grises y Umbralización. De los resultados se toma la cantidad de símbolos generados, la cantidad de palabras detectadas y cuántas de ellas presentan errores.

Resultados con Tesseract

En la tabla 3.2 se muestran los tiempos de respuesta para cada proceso involucrado en la traducción de una imagen a voz empleando la librería Tesseract para el reconocimiento de caracteres.

Tabla 3.2: **Tiempos de respuesta medidos con Tesseract**

Proceso	**Tiempo promedio [s]**
Captura de imagen	4.7
preprocesamiento de imagen	5.3
OCR - generación de archivo de texto	20.0
TTS	2.0

El tiempo que toma la captura de la imagen es indicado por el propio software que lleva a cabo el proceso. Además, para poder tener una captura de imagen adecuada se debió saltar o esperar 100 Frames, puesto que si se toma el primero, la cámara no toma una captura adecuada dado que está en su proceso de inicio. Finalmente, teniendo en cuenta que archivo de texto se genera a la vez que el proceso OCR se lleva a cabo, el tiempo que corresponde a estas funciones se toma en conjunto.

Tabla 3.3: **Precisión en OCR con Tesseract**

Palabras en texto original	432	460	459	519
Filtro Aplicado:	**Escala de grises**			
Parámetro	**Imagen 1**	**Imagen 2**	**Imagen 3**	**Imagen 4**
Símbolos detectados	450	580	521	376
Palabras detectadas	428	444	495	375
Palabras con error	15	42	44	18
Filtro Aplicado:	**Umbralización adaptativa**			
Parámetro	**Imagen 1**	**Imagen 2**	**Imagen 3**	**Imagen 4**
Símbolos detectados	451	598	498	472
Palabras detectadas	439	483	478	469
Palabras con error	24	57	76	49

En la tabla 3.3 se muestran los valores referentes a precisión obtenida empleando la librería Tesseract para el reconocimiento de caracteres. El término *Símbolos* hace referencia al conjunto de palabras y caracteres sean erróneos o no, que se han detectado en el proceso. Esto se toma en cuenta dado que, si se envían al paso siguiente para ser convertidos en audio, introducen errores que entorpecen la comprensión del oyente respecto a lo que se está narrando.

En algunos casos la cantidad de palabras reconocidas es considerablemente menor a la cantidad de palabras originales. Esto, se debe a errores ocasionados por ruido en la imagen provocando que Tesseract no interprete como palabra ese segmento o en su defecto interprete caracteres erróneos. Por lo tanto, estos fallos se descartaron del conteo al carecer de sentido. Finalmente, se tiene el conteo de palabras que han podido ser reconocidas, pero contienen algún error ortográfico o están mal estructuradas, por ejemplo, ”Napoleón” interpretado por Tesseract como ”Napolcón”, tomando la letra ”e” como una ”c”. Con la Figura 3.2 que muestra una gráfica con una comparación de los resultados utilizando el filtro en escala de grises.

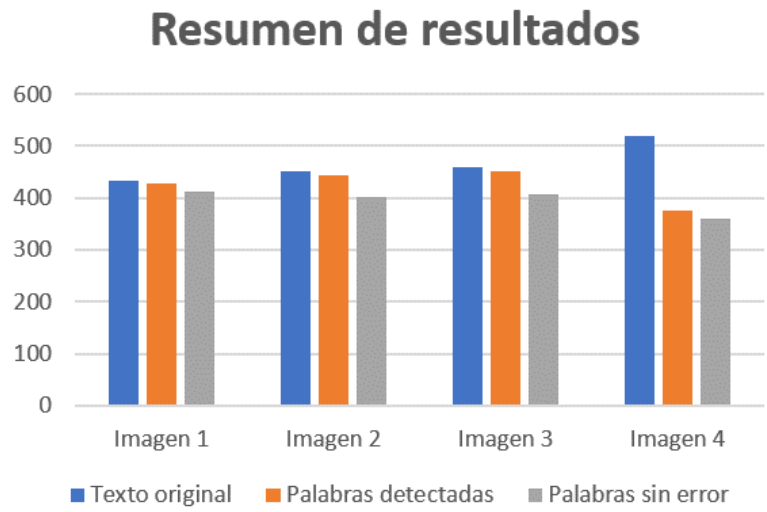


Figura 3.2: **Comparación de resultados con filtro Grayscale**

El porcentaje de error con el filtro escala de grises considerando las palabras acertadas que no tienen fallas, solo es aceptable en el caso de las muestras 1 y 3 con un 4.4% y 1% respectivamente, contrario a las muestras 2 y 4 donde el error es 10.67% y 30.83%. El grado de error en las muestras 2 y 4 no es aceptable, sin embargo, a pesar de que las otras muestras tengan un bajo porcentaje de error las palabras faltantes afectan

la estructura gramatical del texto y aquellas que tengan fallas afectarán negativamente a la comprensión del texto dado que no son removidas del mismo.
Ocurre de manera similar para las pruebas utilizando el filtro Thresholding donde se tienen errores de 3.9%, 5.3%, 12.4% y 18.7% para la imagen 1, 2, 3, y 4 respectivamente. En este caso el error se reduce el error en las muestras 2 y 4, pero aumenta en la muestra 3. El gráfico comparativo se muestra a continuación en la Figura 3.3.

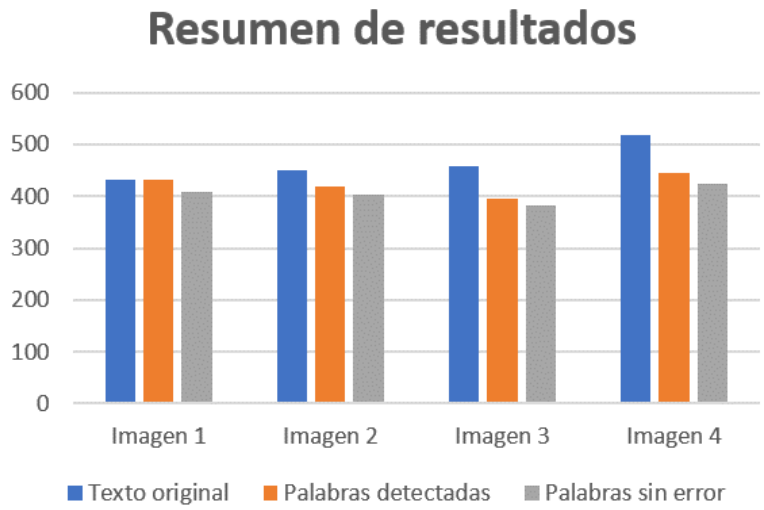


Figura 3.3: **Comparación de resultados con filtro Thresholding**

Resultados con Cloud Vision

En la tabla 3.4 se muestran los tiempos de respuesta para cada proceso involucrado en la traducción de una imagen a voz empleando el API Cloud Vision para el reconocimiento de caracteres.

Tabla 3.4: **Tiempos de respuesta medidos con Cloud Vision**

Proceso	Tiempo promedio [s]
Captura de imagen	4.7
preprocesamiento de imagen	5.3
OCR - generación de archivo de texto	6.0
TTS	2.0

Para este caso el tiempo que le toma a Cloud Vision ejecutar el proceso OCR es mucho menor al tiempo logrado por Tesseract, en consecuencia, el audio resultante está listo en menos tiempo.

En la tabla 3.5 se muestran los valores referentes a precisión obtenida empleando el API Cloud Vision para el reconocimiento de caracteres.

Tabla 3.5: **Precisión en OCR con Cloud Vision**

Palabras en texto original	432	460	459	519
Filtro Aplicado:	**Escala de grises**			
Parámetro	**Imagen 1**	**Imagen 2**	**Imagen 3**	**Imagen 4**
Símbolos detectados	432	450	459	519
Palabras detectadas	432	450	459	519
Palabras con error	0	0	0	0
Filtro Aplicado:	**Umbralización adaptativa**			
Parámetro	**Imagen 1**	**Imagen 2**	**Imagen 3**	**Imagen 4**
Símbolos detectados	431	449	461	518
Palabras detectadas	430	449	459	518
Palabras con error	5	2	17	3

La precisión medida con Cloud Vision es absoluta cuando se emplea el filtro a escala de grises, ya que detecta cada una de las palabras sin ningún tipo de error. En el caso del filtro Thresholding, si bien permite distinguir mejor el texto, en ciertos casos puede distorsionar algún caracter debido a que le agrega o quita uno o varios pixeles demás, esto provoca que una letra cambie su forma, como por ejemplo una letra ”T” por una ”F”. Sin embargo, la precisión sigue siendo aceptable ya que el error no supera el 5% con ninguna de las muestras.

El la Figura 3.4 se muestra la comparativa de resultados para las cuatro muestras tomadas, aplicando el filtro Grayscale.

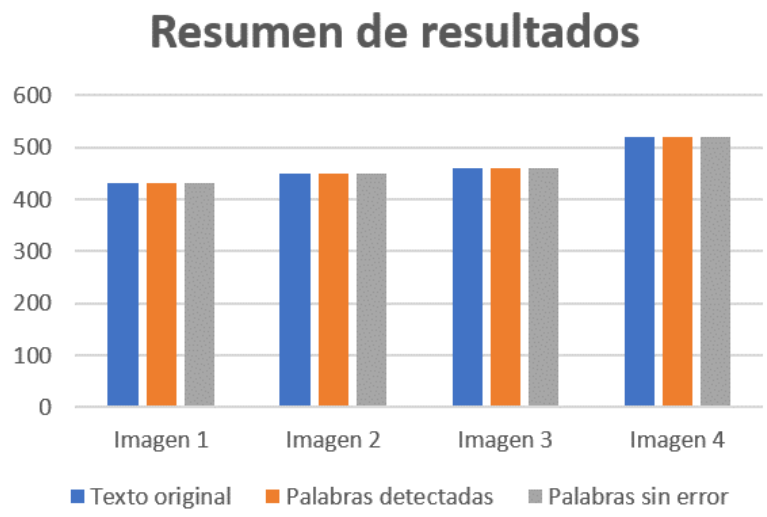


Figura 3.4: **Comparación de resultados con filtro Grayscale**

Se observa que la cantidad de caracteres reconocidos es absoluta, es decir, logró reconocer todos los caracteres sin error alguno. Por el contrario, el filtro Thresholding provoca que se introduzcan algunos errores en los caracteres. Sin embargo, la precisión obtenida sigue siendo mejor que la obtenida por Tesseract. En la Figura 3.5 se muestra el resumen de los resultados para el caso de prueba con el filtro Thresholding.

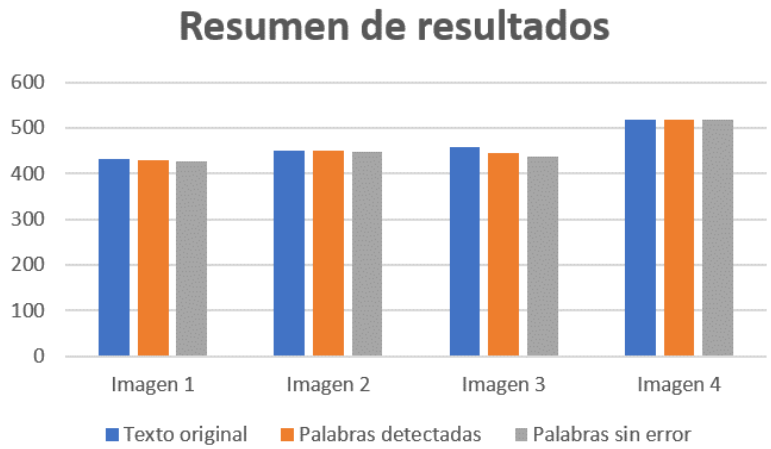


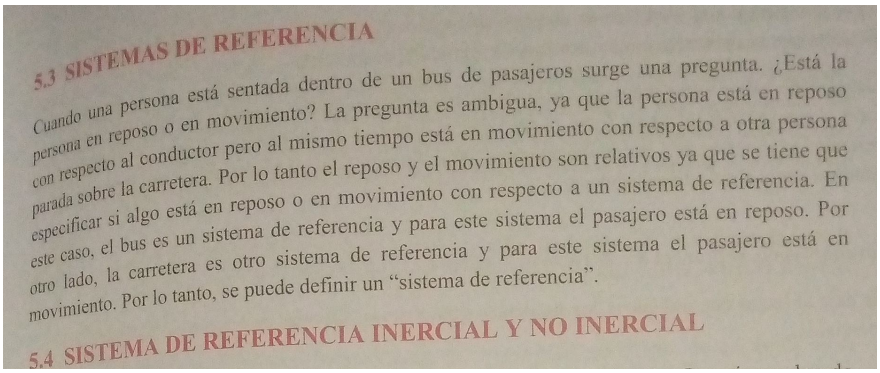
Figura 3.5: **Comparación de resultados con filtro Thresholding**

Cabe destacar, que en ambas pruebas por bajo que sea el error no significa que el texto será poco afectado, dado que como se ha mencionado previamente, los errores ortográficos o de ausencia de palabras afectan a la coherencia de texto, en consecuencia al ser escuchado no será debidamente comprendido .

3.2 Calidad de las imágenes

La calidad de las imágenes capturadas está sujeta a varios parámetros, tanto del ambiente como de la disposición de la hoja, y, entre los principales factores se tienen la iluminación y la presencia de curvaturas en la superficie. La variación de alguno de estos aspectos representará severos cambios en los resultados del texto extraído haciendo que se necesiten distintos filtros al momento de preprocesar cada imagen. Para el tratamiento de las fotografías se hacen uso de filtros de la librería OpenCV, cuyo objetivo es producir una imagen con caracteres mejor definidos en forma y color para así facilitar a Tesseract la extracción del texto, cabe destacar que la resolución de las imágenes captadas por la cámara Pi utilizada en este proyecto es de 2592x1944 píxeles.

La curvatura presente en hojas de libros o cuadernos se puede disminuir levantando la siguiente página hacia los 90 grados con respecto a la página que está sobre la bandeja del dispositivo, además, la superficie sobre la que se colocan las hojas u otro material, será lisa y sin bordes para evitar curvaturas por levantamiento de las hojas o la pasta de un libro. En la Figura 3.6 se presenta un recorte de la parte superior de la imagen utilizada para llevar a cabo las pruebas con la simulación para comparar la precisión de Tesseract frente a Cloud Vision respecto a cómo afecta al resultado la curvatura de las imágenes.


5.3 SISTEMAS DE REFERENCIA

Cuando una persona está sentada dentro de un bus de pasajeros surge una pregunta. ¿Está la persona en reposo o en movimiento? La pregunta es ambigua, ya que la persona está en reposo con respecto al conductor pero al mismo tiempo está en movimiento con respecto a otra persona parada sobre la carretera. Por lo tanto el reposo y el movimiento son relativos ya que se tiene que especificar si algo está en reposo o en movimiento con respecto a un sistema de referencia. En este caso, el bus es un sistema de referencia y para este sistema el pasajero está en reposo. Por otro lado, la carretera es otro sistema de referencia y para este sistema el pasajero está en movimiento. Por lo tanto, se puede definir un "sistema de referencia".

5.4 SISTEMA DE REFERENCIA INERCIAL Y NO INERCIAL

Figura 3.6: **Imagen capturada** *Imagen utilizada para probar la precisión del proceso OCR.*

Los resultados logrados con Cloud Vision se muestran en la Figura 3.7 y se puede observar la afectación ocasionada por la curvatura de la imagen en las líneas iniciales del texto, sin embargo, Cloud Vision evita procesar alguna de ellas, lo que resulta en un texto con pérdida, pero con menos imprecisiones en sus líneas.

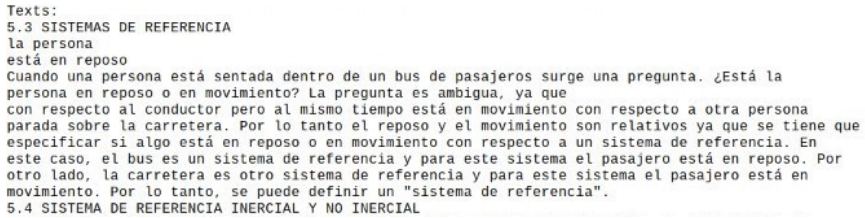


Figura 3.7: **Resultado OCR de Cloud Vision** *Texto resultante del proceso OCR con la herramienta Cloud Vision.*

La Figura 3.8 muestra parte del resultado entregado por Tesseract. La curvatura de la imagen afecta mucho más el resultado final, que además de confundir las líneas, entrega lo que logra reconocer en estas y produce un resultado bastante impreciso y desordenado. Otra observación es que inserta muchos más saltos de línea en el texto. Por lo tanto, este resultado es prácticamente incomprensible una vez generado el archivo de audio.

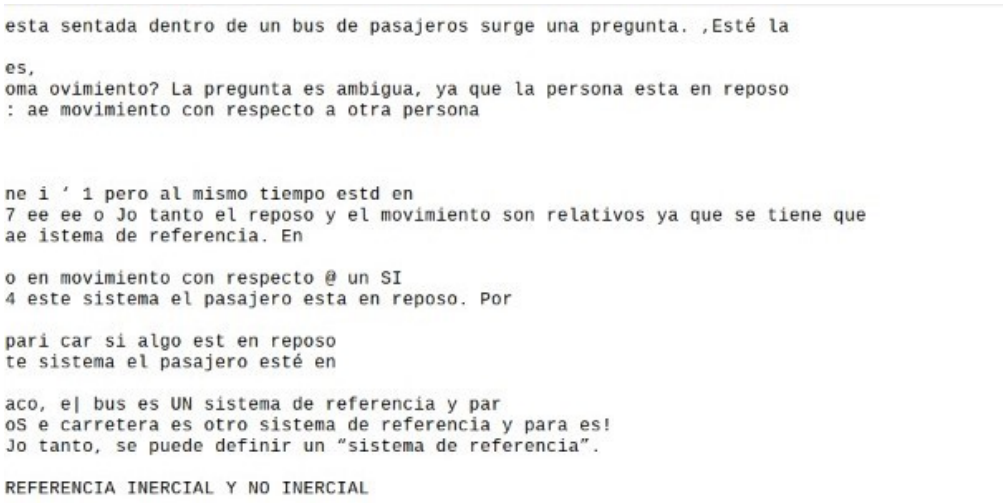


Figura 3.8: **Resultado OCR de Tesseract** *Texto resultante del proceso OCR con la herramienta Tesseract.*

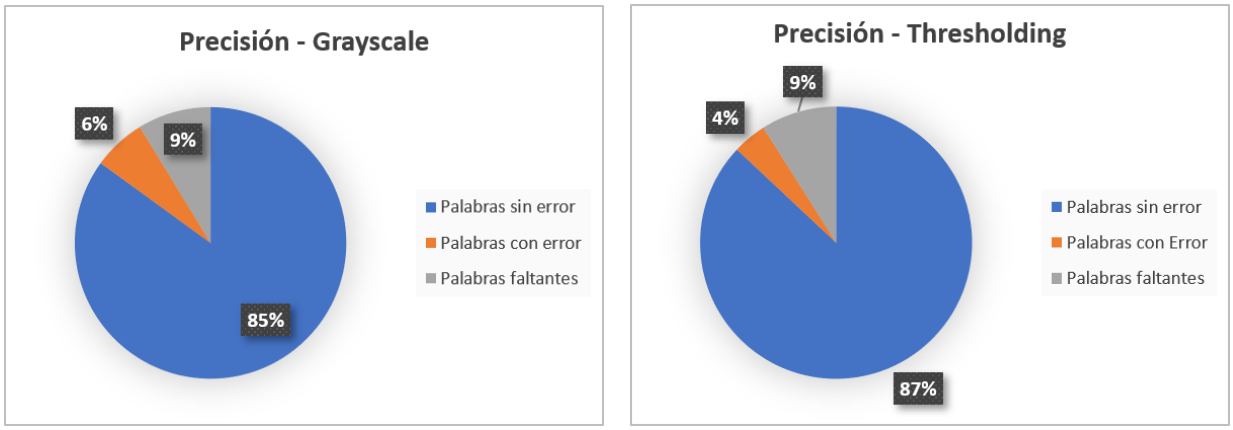
3.3 Precisión en reconocimiento de caracteres

Siguiendo el flujo de las operaciones del dispositivo se llega al reconocimiento de caracteres, esta etapa es alimentada por la imagen preprocesada. En consecuencia, la precisión del texto resultante depende en gran medida de la calidad de la imagen recibida.

Simulando el proceso OCR con la herramienta Tesseract se tuvo una precisión aproximada del 60% en relación al texto en la imagen, esto considerando que en

esta prueba se ingresó imágenes con algún grado de curvatura, lo cual no puede ser corregido adecuadamente en la etapa de preprocesamiento. Otro inconveniente es la falta de precisión para reconocer ciertos caracteres especiales como barras, signos de puntuación, además de errores al representar espacios y saltos de línea. Incluso, cuando la imagen presenta curvas, Tesseract puede llegar a duplicar algunas frases o palabras, interpretando el desnivel de una línea como si se tratase de dos.

Para minimizar los inconvenientes en el reconocimiento y mejorar la precisión del texto resultante, se optó por probar Cloud Vision como alternativa para el proceso OCR. Los resultados obtenidos en la simulación y prueba de funcionamiento con esta API fueron precisos, logrando no solo mejorar la fidelidad del texto, sino que, además, minimiza el inconveniente de la curvatura en algunas imágenes. Sin embargo, el uso de esta API requiere conexión a internet y una cuenta en Google Cloud. A continuación, se la Figura 3.9, se muestra la precisión lograda por Tesseract.

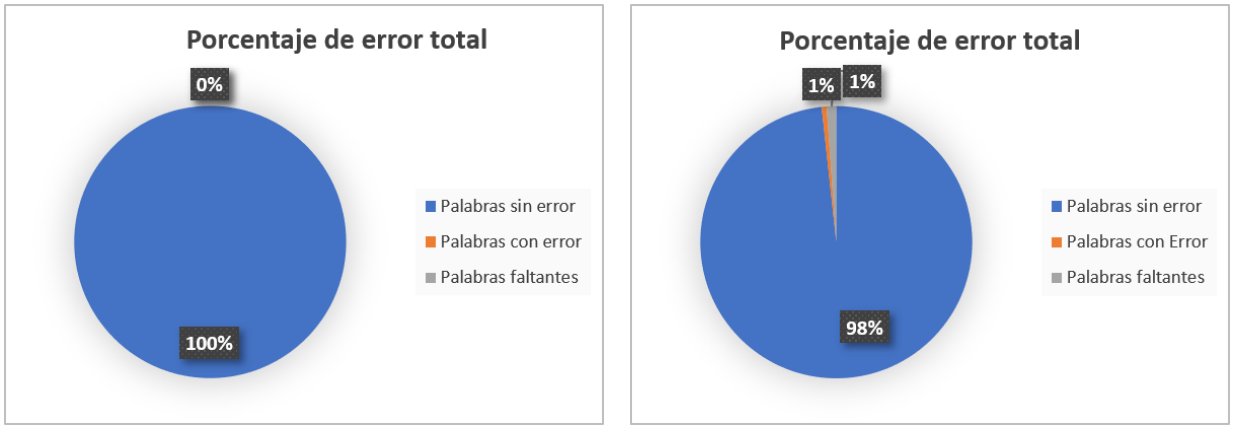


(a) Porcentaje de error total con filtro Grayscale (b) Porcentaje de error total con filtro Thresholding

Figura 3.9: **Porcentaje de error total:** Comparando el filtro Grayscale y Thresholding usando Tesseract

Como se observa, la precisión mejora apenas un 2% con el filtro de umbralización aplicado. sin embargo, debe tenerse en cuenta que a pesar de que el porcentaje de error sea relativamente bajo, las palabras faltantes afectan la estructura gramatical del texto y aquellas que tengan fallas afectarán negativamente a la comprensión del texto dado que no son removidas del mismo.

La precisión conseguida utilizando el API de Cloud Vision, es del 100% preprocesando la imagen únicamente con el filtro Grayscale. De manera similar que en el caso de Tesseract, el filtro Thresholding incide de forma negativa en la precisión del resultado. En la Figura 3.10 se muestran los resultados obtenidos con la prueba de ambos filtros.



(a) Porcentaje de error total con filtro Grayscale (b) Porcentaje de error total con filtro Thresholding

Figura 3.10: **Porcentaje de error total:** Comparando el filtro Grayscale y Thresholding usando Cloud Vision

Con el uso de Cloud Vision para el proceso de reconocimiento de caracteres se obtiene un resultado completamente adecuado, ya que acierta en signos de puntuación, acentos, saltos de línea y espaciado lo que se traduce en un texto coherente y permite que el audio de la lectura sea comprensible en su totalidad.

3.4 Calidad de la voz emulada

Utilizando la herramienta pyTTS fue posible generar los archivos de audio correspondientes al texto digitalizado de las imágenes de prueba. Sin embargo, la calidad de esta voz es muy baja debido a que es una herramienta que no posee un sintetizador de voz adecuado. Los resultados de pyTTS pueden escucharse siguiendo el enlace dispuesto en el apéndice A. La calidad producida por esta librería no puede ser mejorada de ninguna forma, ya que no solo carece de configuraciones adicionales para el idioma español, sino que, además, no es posible mejorar la calidad de emulación de a voz utilizando otro motor de emulación.

gTTS produce un resultado audible y comprensible, que responde a la necesidad del proyecto, dado que la claridad de la voz es un aspecto imprescindible dada la naturaleza de su propósito. Por lo tanto, se optó por utilizar esta librería y se aplicaron los cambios necesarios como la conexión a Internet del dispositivo vía inalámbrica o cableada dado que gTTS depende de ello para hacer las solicitudes HTTP al API de Google Translate. Entre las restricciones aplicables al desarrollo del ”dispositivo” se encuentran la longitud en caracteres de las líneas a procesar, siendo un máximo de 100 caracteres por solicitud.[25] Los resultados de gTTS pueden escucharse siguiendo el enlace dispuesto en el apéndice A.

CAPÍTULO 4

4. CONCLUSIONES Y LINEAS FUTURAS

Es posible diseñar dispositivos de bajo costo que logren un resultado aceptable tanto en el reconocimiento de caracteres como en la lectura en altavoz del texto. Gracias al desarrollo de librerías de código abierto y APIs que facilitan el acceso a herramientas como DeepLearning. Se logró responder a los objetivos planteados con éxito, contrastando entre las distintas herramientas de software disponibles y eligiendo aquellas que generen un resultado adecuado para el fin propuesto, que es brindar un asistente de lectura para personas no videntes.

4.1 Conclusiones

1. Dado que el funcionamiento del dispositivo ocurre de manera secuencial esto es captura de imagen, preprocesamiento de imagen, reconocimiento de caracteres y proceso texto a voz, el resultado final está sujeto a calidad de lo que entrega cada subproceso. Por lo tanto, si la fotografía capturada no posee una resolución del al menos 1920x1080p o presenta curvaturas el texto no se reconocerá con la precisión requerida, y, en consecuencia el audio no se generará o en su defecto no el contenido no será comprensible.

2. La librería Tesseract que es la escogida en este proyecto por ser OpenSource, depende en gran medida de la calidad de la imagen que se le entrega, puesto que en fotografías con una resolución menor a 1920x1080p la precisión se reduce de un 87% a un 60%. Por otra parte, Cloud Vision provee una extracción de texto que respeta la estructura presente en la imagen sin requerir de procesamientos complejos en la fotografía obteniendo una precisión del 100% en las pruebas de

funcionamiento realizadas. Por lo tanto, la precisión puede ser mejorada para su producción utilizando Cloud Vision, o en su defecto optar por el uso de un módulo de cámara especializado que permita la captura de imágenes a una resolución de imagen mayor a 5MP.

3. La traducción de texto a voz con la herramienta gTTS, hace uso del API de Google Translate, por lo que emula con alta fidelidad el habla humana a diferencia de pyTTS de Python, que entrega un resultado no adecuado para la correcta comprensión auditiva del usuario ya que es una voz que tiene un sonido muy robotizado y no puede mejorarse puesto que no posee alguna configuración que lo permita.

4. El preprocesamiento de las imágenes aporta una mejora en el reconocimiento de caracteres en la mayoría de los casos. Sin embargo, con base en las pruebas realizadas el filtro que presenta menor cantidad de errores por muestra es el filtro Grayscale obteniendo una precisión promedio del 92% considerando las muestras 1, 2 y 3 de las pruebas. Ocurre igual con Cloud Vision, con el cual se obtuvo una precisión absoluta con el filtro mencionado. Por lo tanto, el filtro ayuda más en el reconocimiento de caracteres es Grayscale.

5. El tiempo de respuesta del dispositivo es de 32 segundos desde que el dispositivo captura la imagen hasta el momento en que el audio está listo para ser reproducido. Este podría reducirse si se mejora el tiempo de respuesta del proceso OCR y la generación del archivo de texto que a Tesseract le toma 20 segundos. Esto se puede lograr utilizado Cloud Vision, ya que entrega su resultado un 70% más rápido de acuerdo con los resultados de las pruebas de funcionamiento.

4.2 Recomendaciones

1. Se recomienda establecer una lista de filtros aplicables para preprocesar imágenes que estén orientados a mejorar la apreciación de texto ya que no todos los filtros existentes aportan una mejoría. Además, se debe considerar que en la calidad de las imágenes que influyen factores como la cantidad de luz en el ambiente, sombras o imperfecciones en el objeto a fotografiar como arrugas o dobleces y presencia de curvatura en la superficie de la hoja. Por ejemplo, si se aplica un filtro como Thresholding, que acentúa el color negro en los píxeles de las letras en una imagen a escala de grises, una arruga muy marcada o dobles puede llegar a marcarse como una línea o trazo en el texto, lo que imposibilitará el reconocimiento de caracteres.

2. Se recomienda considerar el tipo de lomo y tamaño del libro u otro material didáctico al que se desee enfocar el diseño de la bandeja. Dado que, si se coloca un libro con lomo pegado o cocido sobre el dispositivo existe mayor probabilidad se produzcan curvaturas en las páginas al abrirlo y esto afecta al reconocimiento de caracteres, lo que haría necesario efectuar un proceso conocido como *Page-Flattening* para reducirla digitalmente y es tecnología que aún se sigue desarrollando. Una alternativa sería limitar el uso a libros cuyo lomo sea anillado, así las páginas no tendrían ninguna curvatura al abrirlo.

.

4.3 Líneas Futuras

1. Dada la dependencia de la calidad de los resultados de una imagen de alta calidad, se esperaría la implementación de un módulo de cámara especializado, que sea adaptable a tarjetas de desarrollo o tarjetas para producción Raspberry o dispositivos similares y permita capturar fotografías en alta definición.

2. Se podrían diseñar mecanismos de hardware que permitan efectuar ”page-flattening” que es más factible que una solución por software, al ser tecnología aún en desarrollo. Estos mecanismos de hardware pueden basarse en principios de reflexión para utilizar láminas de vidrio que reflejen la página del libro de manera perpendicular a la cámara y evitar que se observen curvaturas.

3. Se esperaría el desarrollo de preprocesadores ”inteligentes” de imágenes para OCR, empleando modelos de Machine Learning para que se apliquen filtros a las fotografías del texto dependiendo de las condiciones que presente.

BIBLIOGRAFÍA

[1] L. O. de Discapacidades, "Ley orgánica de discapacidades," *Quito, Pichincha, Ecuador*, 2012.

[2] C.-A. Boiangiu, R. Ioanitescu, and R.-C. Dragomir, "Voting-based ocr system," *Journal of Information Systems Operations Management*, vol. 10, pp. 470–486, 12 2016.

[3] O. ORG, "Image thresholding." https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html/, 2022.

[4] CONADIS, "Estadísticas de discapacidad." https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/, 2022. Extraído de la web oficial de la CONADIS con datos del Ministerio de Salud Pública del Ecuador.

[5] P. School, "Perkins braille writer." https://www.perkins.org/perkins-brailler/, 2022. Extraído de la web oficial Perkins School for the Blind-Solutions.

[6] C. A. Chica García and K. J. Moreta Garcés, "Implementación de un prototipo didáctico para niños no videntes de 4 a 6 años de edad mediante el uso de sensores, para el reconocimiento de las partes del cuerpo humano en el centro de educación especial julius doethner en la ciudad de ambato.," 2020.

[7] K. L. Meléndez Gavilanes, "Material didáctico para personas invidentes de 20 a 30 años de la unidad educativa" centebad" de latacunga.," 2019.

[8] D. Braune, "Braille me and orbit reader: Braille display comparison." hhttps://www.perkins.org/resource/braille-me-and-orbit-reader-braille-display-comparison/, 2020.

[9] J. Nyström Müller and C. Engström, "Consumer adoption of audiobook streaming services," 2019.

[10] L. E. O. Gil, E. V. Zayas, F. V. Guzmán, I. G. Ortega, F. R. Ramírez, S. A. V. Gamboa, and J. R. Reyes, “Aplicación móvil que traduce ecuaciones matemáticas a voz mediante ocr (mobile app that translates math equations by voice using ocr),” *Pistas Educativas*, vol. 42, no. 136, 2020.

[11] P. K. Charles, V. Harish, M. Swathi, and C. Deepthi, “A review on the various techniques used for optical character recognition,” *International Journal of Engineering Research and Applications*, vol. 2, no. 1, pp. 659–662, 2012.

[12] A. Lundgard, C. Lee, and A. Satyanarayan, “Sociotechnical considerations for accessible visualization design,” in *2019 IEEE Visualization Conference (VIS)*, pp. 16–20, 2019.

[13] R. Pi, “Raspberry pi 3b+ product brief.” `https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf`, 2022.

[14] piCamera, “Pi camera v1.3 documentation.” `https://picamera.readthedocs.io/en/release-1.13/fov.html#`, 2016.

[15] R. Pi, “Raspberry pi documentation - sd cards.” `https://www.raspberrypi.com/documentation/computers/getting-started.html#sd-cards-for-raspberry-pi`, 2022.

[16] R. Pi, “Raspberry pi documentation - os.” `https://www.raspberrypi.com/documentation/computers/os.html`.

[17] J. S. Walker, *Python: La Guía Definitiva para Principiantes para Dominar Python*. Babelcube Inc., 2018.

[18] R. Smith, “Tesseract ocr.” `https://github.com/tesseract-ocr/tesseract`, 12 2021.

[19] OpenCV, “Opencv api reference documentation.” `https://docs.opencv.org/2.4.13.7/modules/core/doc/intro.html`, 2014.

[20] G. A. V. M. Arévalo, J. González, “La librería de visión artificial opencv: Aplicación a la docencia e investigación,” 2003.

[21] P. N. Durette, “gtts (google text-to-speech).” https://gtts.readthedocs.io/en/latest/.

[22] G. Cloud, “Google cloud vision guide.” https://cloud.google.com/vision/docs/features-list?hl=es-419, 2022.

[23] G. Cloud, “Google cloud vision prices.” https://cloud.google.com/vision/pricing?hl=es-419, 2022.

[24] M. Vital-Carrillo, “Estructuras de control para la programación,” *Vida Científica Boletín Científico de la Escuela Preparatoria No. 4*, vol. 7, ene. 2019.

[25] “Conceptos básicos de cloud text-to-speech.” https://cloud.google.com/text-to-speech/docs/basics.

APÉNDICES

Apéndice A: Audios resultantes

En esta sección se presentan enlaces hacia la plataforma YouTube donde se han subido los archivos de audio resultantes utilizando las dos librerías propuestas.

Resultado de **PyTTS:** `https://youtu.be/YFnkkNqiLuw`

Resultado de **gTTS:** `https://youtu.be/YFnkkNqiLuw`

Apéndice B: Modelos tridimensionales propuestos

En esta sección se presentan las vitas frontales de los modelos tridimensionales propuestos, elaborados en la plataforma Tinkercad.

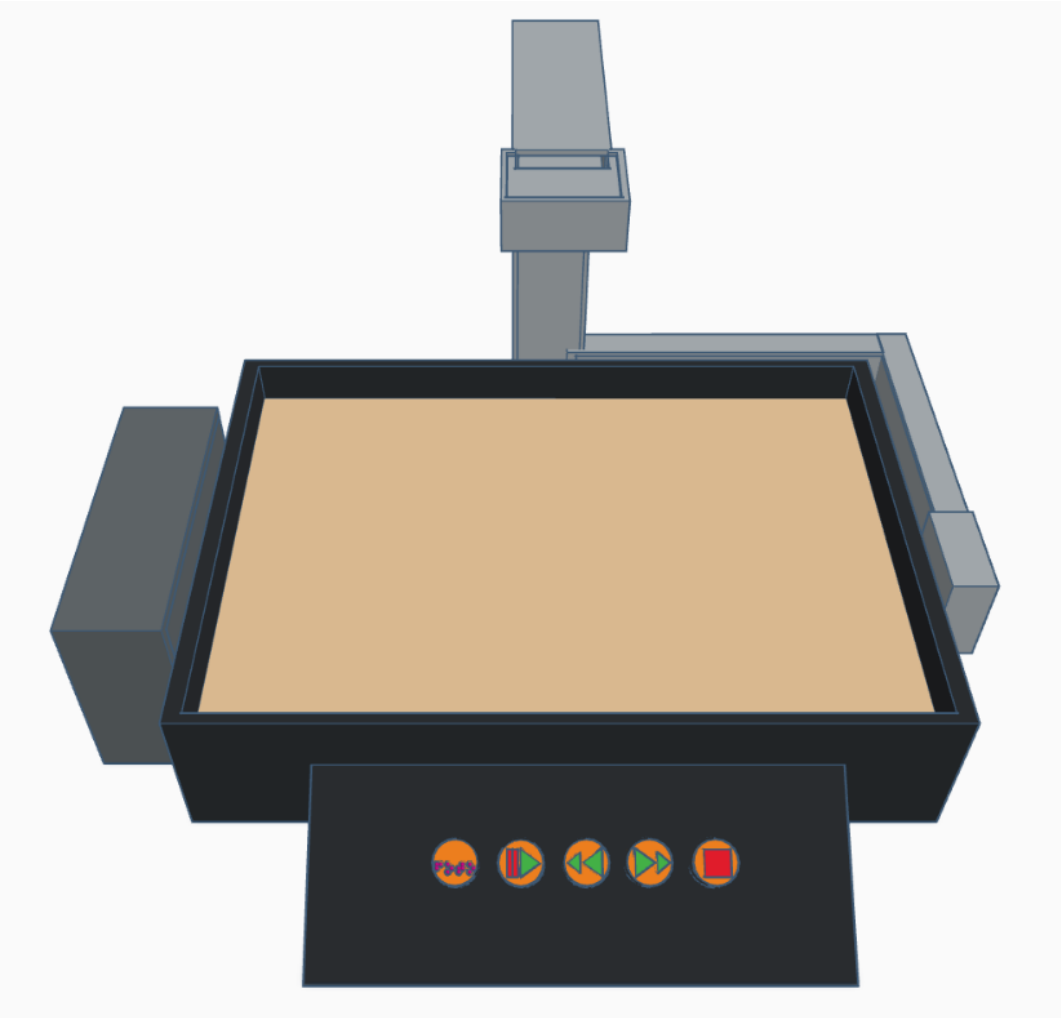


Figura 1: **Vista frontal de Modelo A**

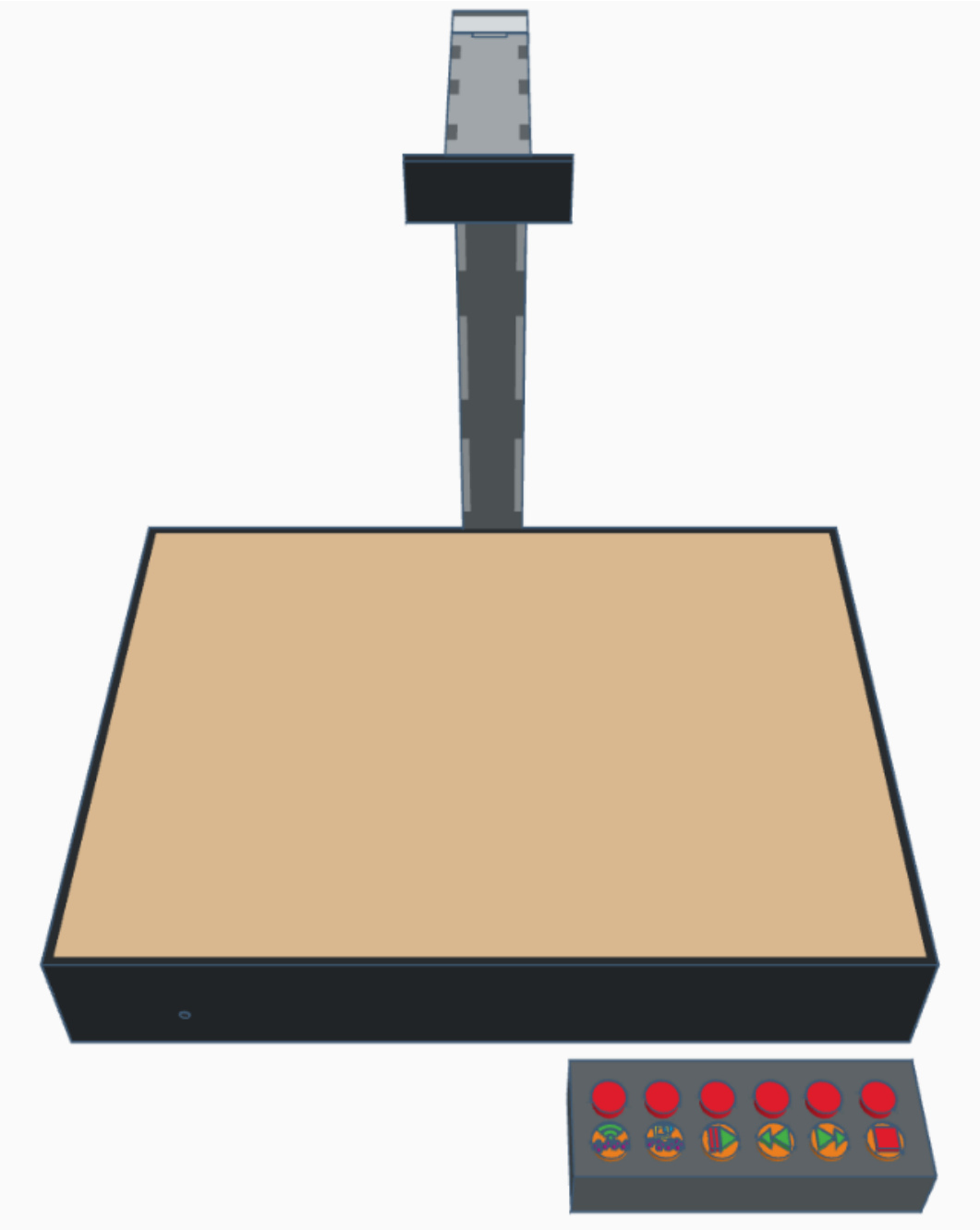


Figura 2: **Vista frontal de Modelo B**