

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniero de Sistemas**

**TEMA:
DESARROLLO DE UN SISTEMA PARA RECONOCIMIENTO DE TEXTO Y
CONVERSIÓN A AUDIO, UTILIZANDO RASPBERRY PI PARA PERSONAS NO
VIDENTES.**

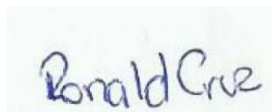
**AUTOR:
RONALD LUCIANO CRUZ GARCÉS**

**TUTOR:
MANUEL RAFAEL JAYA DUCHE**

Quito, marzo del 2021

CESIÓN DE DERECHOS DE AUTOR

Yo, Ronald Luciano Cruz Garcés, con documento de identificación N° 1600534554, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy el autor del trabajo de titulación con el tema: **DESARROLLO DE UN SISTEMA PARA RECONOCIMIENTO DE TEXTO Y CONVERSIÓN A AUDIO, UTILIZANDO RASPBERRY PI PARA PERSONAS NO VIDENTES.**, mismo que ha sido desarrollado para optar por el título de: **INGENIERO DE SISTEMAS** en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



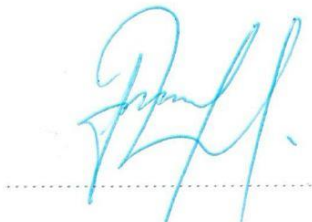
.....
RONALD LUCIANO
CRUZ GARCÉS
C.I: 1600534554

Quito, marzo del 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: DESARROLLO DE UN SISTEMA PARA RECONOCIMIENTO DE TEXTO Y CONVERSIÓN A AUDIO, UTILIZANDO RASPBERRY PI PARA PERSONAS NO VIDENTES., realizado por Ronald Luciano Cruz Garcés, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, marzo del 2021

A handwritten signature in blue ink, appearing to be 'Manuel Rafael Jaya Duché', written over a horizontal dashed line.

.....
MANUEL RAFAEL JAYA DUCHE
CI: 1710631035

DEDICATORIA

Dedico este trabajo en primer lugar a Dios, por darme la fuerza, sabiduría y por haberme guiado por el camino del bien.

A mi familia que siempre han sido un pilar fundamental en toda etapa de mi vida.

A mis padres por toda la dedicación, la comprensión y el apoyo que nunca me ha faltado en cada etapa de mi vida, por el apoyo con los recursos necesarios para poder cumplir mis estudios.

A mis hermanos por acompañarme en los peores momentos, por su compañía consejos que jamás faltaron.

A mi hija por ser mi fuente de inspiración, por darme la fuerza para seguir adelante día a día.

RONALD LUCIANO CRUZ GARCÉS

AGRADECIMIENTO

Agradecer a Dios por haberme dado la fuerza, la sabiduría, salud y protección para poder culminar esta etapa de mi vida.

Quiero agradecer a la Universidad Politécnica Salesiana que ha contribuido en mi formación personal y profesional, a mi tutor de proyecto de titulación el Ing. Rafael Jaya por haberme orientado y haber realizado el trabajo de la mejor manera, a mis padres por toda la paciencia y el esfuerzo que hicieron para poder tener mi formación profesional, a mi mejor amigo por sus consejos, por todo el apoyo que a depositado a lo largo de mi carrera universitaria y profesional.

RONALD LUCIANO CRUZ GARCÉS

INDICE

| | |
|--|----|
| INTRODUCCIÓN | 1 |
| Antecedentes..... | 1 |
| Problema..... | 1 |
| Justificación | 2 |
| Objetivos..... | 3 |
| Objetivo General..... | 3 |
| Objetivos específicos | 3 |
| Metodología..... | 3 |
| CAPITULO 1..... | 7 |
| ESTADO DEL ARTE..... | 7 |
| 1.1. Discapacidad Visual | 7 |
| 1.2. Discapacidad visual en Ecuador..... | 7 |
| 1.3. Sistemas Embebidos | 8 |
| 1.4. Microcomputador | 8 |
| 1.5. Tipos de Microcomputadores..... | 9 |
| 1.5.1. Tarjeta Raspberry Pi..... | 9 |
| 1.5.2. ASUS Tinker Board..... | 9 |
| 1.5.3. UDOO X86..... | 9 |
| 1.5.4. Orange Pi | 9 |
| 1.5.5. Cuadro comparativo entre microcomputadores..... | 10 |
| 1.6. Lenguaje de programación..... | 11 |
| 1.6.1. Lenguaje de programación Python | 11 |
| 1.6.2. Thonny Python..... | 12 |
| 1.6.3. PyCharm..... | 12 |
| 1.6.4. Qt Designer..... | 12 |
| 1.7. Sistema operativo Raspberry Pi OS | 13 |
| 1.8. Librería OpenCV | 13 |
| 1.9. Tesseract | 13 |
| 1.10. Modulo cámara Raspberry Pi IR-CUT | 14 |
| CAPITULO 2..... | 15 |
| DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO | 15 |
| 2.1. Diagrama general de funcionamiento..... | 15 |
| 2.1.1. Diagrama de secuencias | 17 |
| 2.1.2. Diagrama de componentes | 18 |

| | | |
|-------|---|----|
| 2.2. | Raspberry Pi 4 B | 19 |
| 2.3. | Cámara Raspberry Pi..... | 20 |
| 2.4. | Parlantes USB Genius SP-U150X..... | 21 |
| 2.5. | Memoria Micro SD..... | 21 |
| 2.6. | Hardware utilizado para el desarrollo del prototipo | 22 |
| 2.7. | Diagrama de bloques | 23 |
| 2.8. | Instalación del sistema operativo Raspberry Pi OS..... | 23 |
| 2.9. | Configuración SSH en Raspberry Pi OS | 31 |
| 2.10. | Habilitar VNC en Raspberry Pi OS | 35 |
| 2.11. | Habilitar Raspberry Pi Camera..... | 39 |
| 2.12. | Instalación librería OpenCV | 42 |
| 2.13. | Descripción de código..... | 53 |
| 2.14. | Estructura del dispositivo | 56 |
| 2.15. | Algoritmo utilizado para el reconocimiento de texto..... | 57 |
| 2.16. | Proceso realizado por el algoritmo | 58 |
| | CAPITULO 3..... | 60 |
| | PRUEBAS Y RESULTADOS | 60 |
| 3.1. | Pruebas del prototipo..... | 60 |
| 3.2. | Pruebas con poca luz | 67 |
| 3.3. | Pruebas de distancia entre cámara y texto a leer..... | 69 |
| 3.4. | Análisis de resultados | 73 |
| | CONCLUSIONES | 74 |
| | RECOMENDACIONES | 76 |
| | GLOSARIO DE TÉRMINOS | 77 |
| | LISTA DE REFERENCIAS..... | 78 |

INDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Número de personas con discapacidad. | 8 |
| Tabla 2. Cuadro comparativo de microcomputadores. | 10 |
| Tabla 3. Asignación de partes para el prototipo del sistema. | 20 |
| Tabla 4. Características técnicas del módulo de cámara. | 20 |
| Tabla 5. Características técnicas de parlantes USB. | 21 |
| Tabla 6. Palabras reconocidas en porcentaje. | 60 |
| Tabla 7. Pruebas con intensidad de luz | 67 |
| Tabla 8. Pruebas de distancia..... | 73 |

INDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Ciclo para el diseño del prototipo. | 4 |
| Figura 2. Desarrollo del prototipo..... | 5 |
| Figura 3. Funcionamiento general del prototipo..... | 15 |
| Figura 4. Diagrama de secuencias de la interacción de componentes. | 17 |
| Figura 5. Diagrama de componentes..... | 18 |
| Figura 6. Raspberry Pi 4B. | 19 |
| Figura 7. Módulo de cámara IR-CUT. | 20 |
| Figura 8. Parlantes USB Genius SP-U150X. | 21 |
| Figura 9. Memoria Micro SD. | 22 |
| Figura 10. Hardware utilizado para el desarrollo del prototipo..... | 22 |
| Figura 11. Diagrama de bloques, interacción del prototipo. | 23 |
| Figura 12. Opciones de descarga de Raspberry Pi Imager. | 24 |
| Figura 13. Instalación Raspberry Pi Imager. | 25 |
| Figura 14. Instalación de Raspberry Pi OS. | 25 |
| Figura 15. Selección de Raspberry Pi OS. | 26 |
| Figura 16. Selección de la tarjeta Micro SD..... | 27 |
| Figura 17. Instalación de Raspberry Pi OS en la tarjeta Micro SD. | 27 |
| Figura 18. Inicio y configuración del sistema operativo Raspberry Pi OS. | 28 |
| Figura 19. Configuración de país, zona horaria, idioma Raspberry Pi OS. | 29 |
| Figura 20. Cambiando la contraseña en el sistema Raspberry Pi OS. | 29 |
| Figura 21. Asignación de una red Wifi. | 30 |
| Figura 22. Actualización del sistema operativo Raspberry Pi OS. | 31 |
| Figura 23. Comando para iniciar la configuración en Raspberry Pi OS. | 32 |
| Figura 24. Configuración en Raspberry Pi OS. | 32 |
| Figura 25. Habilitar SSH en Raspberry Pi OS..... | 33 |
| Figura 26. Conexión desde Windows a Raspberry Pi OS..... | 34 |
| Figura 27. Alerta de seguridad de Putty. | 34 |
| Figura 28. Conexión exitosa entre Windows a Raspberry Pi OS. | 35 |
| Figura 29. Ingresar a la configuración para habilitar VNC. | 36 |
| Figura 30. Habilitación completada de VNC. | 37 |
| Figura 31. Conexión remota desde Windows a Raspberry Pi OS. | 38 |
| Figura 32. Credenciales para la conexión remota..... | 38 |
| Figura 33. Escritorio Remoto Raspberry Pi OS..... | 39 |
| Figura 34. Ventana para habilitar la cámara en Raspberry Pi OS. | 40 |
| Figura 35. Prueba de funcionamiento de cámara..... | 40 |
| Figura 36. Captura de imagen por la camara IR-CUT..... | 41 |
| Figura 37. Comandos para iniciar la cámara mediante líneas de código en Python..... | 42 |
| Figura 38. Ventana de configuración de herramientas de Raspberry Pi OS. | 42 |
| Figura 39. Ventana para expansión de archivos del sistema operativo..... | 43 |
| Figura 40. Expansión de archivos en la Micro SD correcta. | 44 |
| Figura 41. Instalación de la herramienta CMake. | 45 |
| Figura 42. Instalación de paquetes para lectura de varios formatos de imagen. | 45 |
| Figura 43. Instalación de las dependencias de OpenCV. | 46 |
| Figura 44. Instalación de las dependencias de OpenCV. | 46 |

| | |
|---|----|
| Figura 45. Instalación de las dependencias de la librería OpenCV. | 47 |
| Figura 46. Instalación de herramienta de optimización de OpenCV. | 47 |
| Figura 47. Instalación de Python versión 3. | 48 |
| Figura 48. Proceso de instalación de Imutils para Python. | 49 |
| Figura 49. Instalación del paquete Numpy..... | 49 |
| Figura 50. Instalación de la librería OpenCV..... | 50 |
| Figura 51. Instalación de Pillow en Python..... | 51 |
| Figura 52. Instalación del contenedor Tesseract..... | 51 |
| Figura 53. Instalación de la biblioteca Pyttsx3..... | 52 |
| Figura 54. Instalación de Espeak para la generación de archivos mp3..... | 53 |
| Figura 55. Librerías, objetos usados para el funcionamiento de la cámara..... | 54 |
| Figura 56. Valores por defecto utilizados para la librería OpenCV..... | 54 |
| Figura 57. Carga de la imagen a la librería OpenCV..... | 55 |
| Figura 58. Conversión y reproducción de texto detectado a audio..... | 56 |
| Figura 59. Estructura del prototipo. | 57 |
| Figura 60. Diagrama de los procesos realizados por el algoritmo..... | 58 |
| Figura 61. Primera prueba capturada por el prototipo. | 61 |
| Figura 62. Texto convertido por el prototipo. | 62 |
| Figura 63. Segunda prueba realizada tomada de un artículo de revista. | 64 |
| Figura 64. Texto reconocido por el prototipo..... | 65 |
| Figura 65. Reconocimiento de letras de tamaño 8..... | 66 |
| Figura 66. Tercera prueba con letra tamaño 8..... | 67 |
| Figura 67. Prueba con luz de intensidad baja. | 68 |
| Figura 68. Segunda prueba con luz de intensidad media. | 68 |
| Figura 69. Tercera prueba con luz de intensidad fuerte. | 69 |
| Figura 70. Prueba a 20 cm de distancia..... | 70 |
| Figura 71. Prueba con cámara a 30 cm de distancia. | 71 |
| Figura 72. Prueba a 40 cm de distancia..... | 72 |

Resumen

El presente proyecto tiene como objetivo principal ayudar a las personas con discapacidad visual. Se presenta un prototipo que su principal objetivo será detectar un documento impreso y transformarlo a un formato de audio, el cual reproducirá el texto detectado de la captura previamente realizada. El concepto presenta un solo módulo que permite reproducir un archivo de audio mediante la tecnología de detección de imágenes en tiempo real. El proyecto utiliza la tarjeta Raspberri Pi para la parte de procesamiento y una cámara compatible con la tarjeta antes mencionada para la captura de imágenes. Finalmente, los resultados obtenidos muestran el diseño del prototipo de captura y conversión a audio, se realiza varios experimentos para mostrar la precisión de captura y lectura del texto impreso. Este proyecto tiene la ventaja de utilizar dispositivos portátiles que aparte de este prototipo se puede usar para más tareas. De acuerdo a las pruebas realizadas se puede determinar que la distancia óptima para el buen enfoque es 30 cm desde la cámara hacia el documento ya que a esa distancia enfoca el total de la hoja, la iluminación es una parte importante a la hora de procesar texto impreso ya que con una buena iluminación el prototipo puede reconocer mejor el texto y entregar mejor información a la persona que lo solicita, la tarjeta Raspberry Pi 4 ofrece los recursos necesarios para el correcto funcionamiento del prototipo ya que cuenta con 4 GB en RAM para su procesamiento, adicional es compatible con la cámara IR-CUT.

Abstract

The main objective of this project is to help people with visual disabilities. A prototype is presented whose main objective will be to detect a printed document and transform it into an audio format, which will reproduce the text detected from the previously made capture. The concept features a single module that allows an audio file to be played back using real-time image detection technology. The project uses the Raspberri Pi card for the processing part and a camera compatible with the aforementioned card for image capture. Finally, the results obtained show the design of the prototype of capture and conversion to audio, several experiments are carried out to show the precision of capture and reading of the printed text. This project has the advantage of using portable devices that apart from this prototype can be used for more tasks. According to the tests carried out, it can be determined that the optimal distance for good focus is 30 cm from the camera to the document since at that distance the entire sheet focuses, lighting is an important part when processing text printed since with good lighting the prototype can better recognize the text and deliver better information to the person requesting it, the Raspberry Pi 4 card offers the necessary resources for the correct functioning of the prototype since it has 4 GB of RAM for its Additional processing is supported by IR-CUT camera.

INTRODUCCIÓN

Antecedentes

Conocer los desafíos a los que se enfrentan las personas no videntes o con alguna discapacidad visual, puede ayudar a que las personas videntes comprendan a todo lo que se tiene que enfrentar una persona con discapacidad visual.

Las personas con poca visión suelen tener problemas para moverse sobre todo en entornos poco conocidos. El movimiento físico es uno de sus grandes retos. Por esta razón muchos de ellos siempre van acompañados de algún amigo o familiar. Las personas con discapacidad visual se aprenden todos los detalles, por muy pequeños que sean, de su casa ya que muebles cambiados de sitio o movido pueden suponerles un problema.

Hoy en día la mayoría de las personas que poseen algún tipo de discapacidad se les limita acceder en su totalidad a los servicios que la sociedad ofrece, por otro lado, las instituciones públicas y privadas no poseen al 100% de las condiciones estructurales para facilitar la integración de una persona con discapacidad visual a las mismas, poseen deficiencia al momento de facilitarles capacitaciones.

La tecnología es un recurso muy importante en la vida actual para la información y la comunicación, las personas con discapacidad visual tienen dificultades para la utilización de las mismas, a partir de esto se conocen las insuficiencias de esta parte de la población.

Problema

La discapacidad es considerada un problema en la sociedad, existe una exclusión por parte de las personas tanto en el ámbito social como en el laboral, todo esto abarca la falta de oportunidades en las organizaciones, la discapacidad visual no limita a una persona para

poder conseguir un trabajo, ya que, al no tener el sentido de la vista, dichas personas pueden desarrollar otros sentidos como el tacto y conocen muy bien su espacio de trabajo.

En el Ecuador existe un registro de 485.325 personas registradas con algún tipo de discapacidad de las cuales existen 56.570 personas con discapacidad visual según el Consejo Nacional para la Igualdad de Discapacidades – CONADIS (CONADIS, 2020).

En nuestro país la mayoría de personas con discapacidad visual no tienen un acceso adecuado a trabajo, transporte. Elementos que son esenciales para su óptimo desarrollo en su vida diaria.

El presente proyecto tiene como finalidad ayudar a las personas no videntes en general en su área de trabajo, ya que tienen dificultad para leer un texto impreso. Mediante esta herramienta se pretende ayudar a dichas personas a que puedan saber qué es lo que dice en un texto impreso.

Justificación

El apoyo en la igualdad de la comunicación es muy necesario en la actualidad, ya que en un mundo donde la información llega en segundos de cualquier parte del mundo no se puede pensar que las personas con discapacidad visual no puedan acceder a los medios necesarios para informarse ya sea una noticia, un escrito que reciban.

Tomando en cuenta las necesidades de las personas invidentes, se decidió construir un prototipo que ayude a las personas no videntes aprovechando su fácil construcción y programación, con este prototipo se presenta la oportunidad de desenvolverse en el ambiente laboral ya que no necesitaría de alguien más al momento de leer texto impreso.

Así se puede aprovechar la tecnología que se proporciona hoy en día para lograr futuras mejoras en su calidad de vida y las personas no videntes se encuentren mejor integradas en la sociedad y el ambiente laboral.

Objetivos

Objetivo General

Desarrollar un prototipo de un dispositivo portátil, que permita leer textos y reproduzca las palabras mediante la voz humana sintetizada para personas no videntes en general.

Objetivos específicos

Realizar una investigación acerca de los sistemas embebidos, tipos de microcomputadores: ventajas, desventajas de cada uno, beneficios, costos y proyecciones que pueden brindar para realizar el presente proyecto.

Analizar equipos, dispositivos, herramientas, software que se usarán para la creación de un sistema interactivo que pueda facilitar el diario vivir de las personas que lo usen.

Diseñar un prototipo que permita la lectura de texto impreso y sea orientado a personas con discapacidad visual.

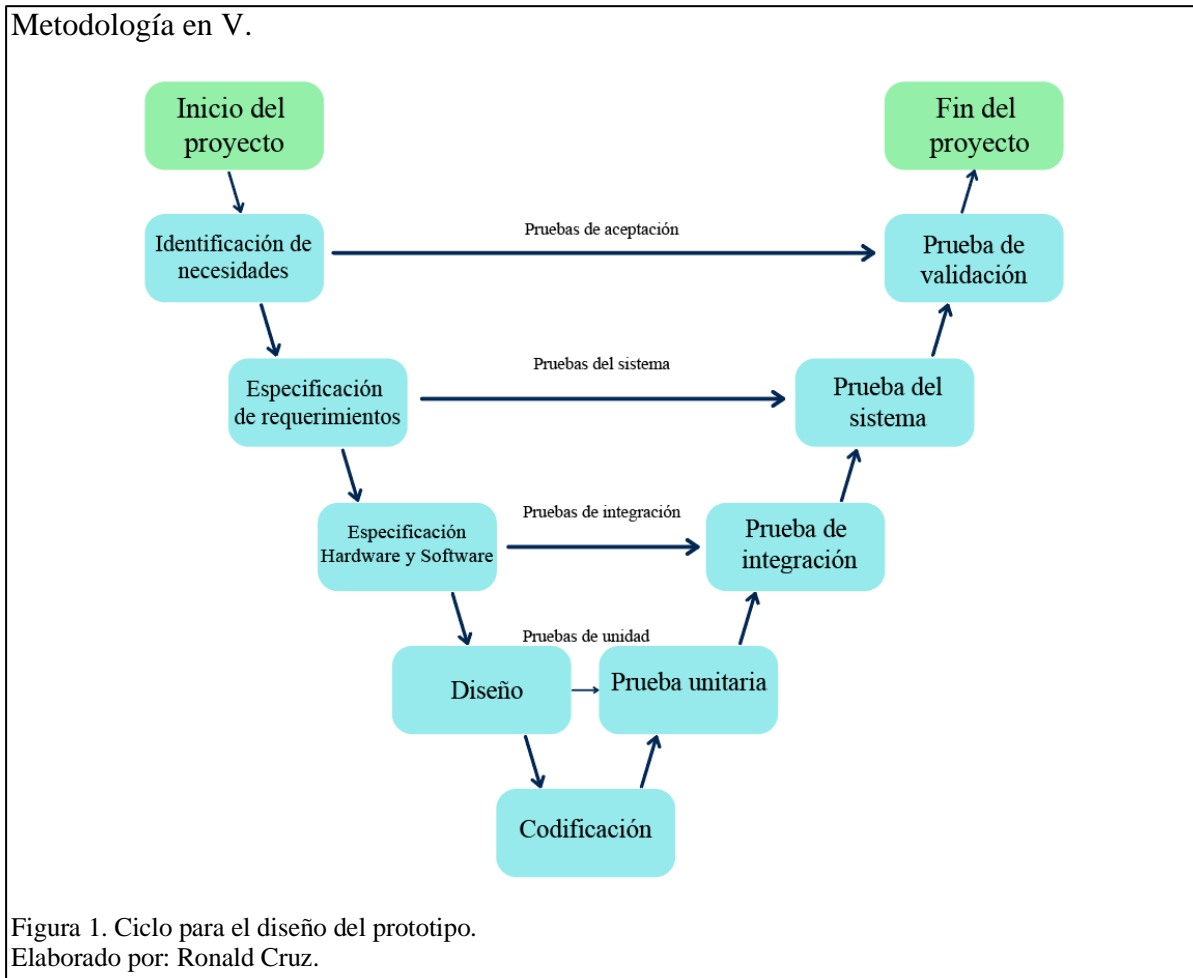
Desarrollar un sistema prototipo que sea de fácil uso utilizando un dispositivo embebido.

Realizar las pruebas necesarias de su buen funcionamiento y fácil uso.

Metodología

El presente proyecto utilizará la metodología modelo en V, ya que primero analiza la problemática, después que se especifica los problemas a detalle, se debe plantear los

requisitos que se usará con respecto al software y hardware, se debe documentar las especificaciones y requerimientos, todo esto para quedar claro de los procesos a realizar en el proyecto, por último, se debe desarrollar el o los módulos que tendrá el prototipo.



El prototipo consta de un módulo que tendrá los botones: capturar, limpiar, un modo automático, por otra parte, tendrá un espacio para una previsualización del objeto capturado, adicional tendrá los botones para poder rotar la imagen, y cambiar de lado ya sea izquierdo o derecho.

El prototipo estará ubicado en una base fija que será adecuada para poner la hoja impresa, la tarjeta Raspberry Pi, parlantes y la cámara, todo esto estará adecuado para un entorno de oficina, esto quiere decir que se requerirá de buena iluminación.

La información que envíe el prototipo podrá ser escuchada por una salida de audio ya que es la mejor opción para una persona no vidente.

Definición de requisitos

Las características que requiere el prototipo es que capture una imagen que contenga texto, lo digitalice y devuelva una nota de audio leyendo las palabras que tenga el documento antes capturado.

Desarrollo del prototipo

Los dos métodos que se ocuparán en encender la cámara y tomar la foto serán enviadas a la librería OpenCV quien se encargará de tratar la imagen y a su vez procesará la foto y la digitalizará, existe otro método en el cual se procesará la imagen, la transformará a un formato de audio y este se reproducirá automáticamente todas las palabras que a reconocido de la imagen capturada.

Procesos descritos en imágenes del prototipo.

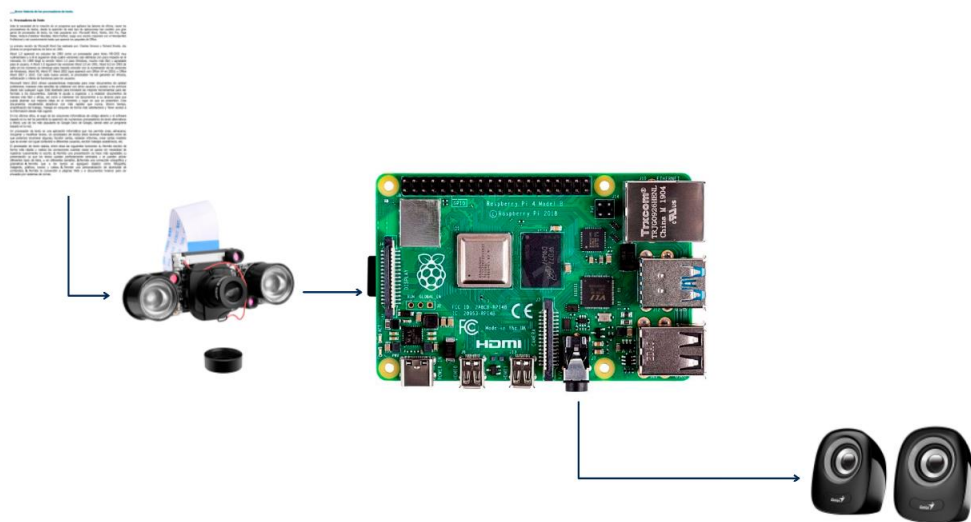


Figura 2. Desarrollo del prototipo.
Elaborado por: Ronald Cruz.

Diseño del prototipo

Para el diseño del prototipo se tendrá en cuenta la parte de captura de fotos se debe contar con una cámara con una resolución aceptable, la base donde va la hoja impresa debe ser adecuada para este proceso, así como la ubicación de la cámara, se debe contar con una buena iluminación para que el software que procese la información pueda leer de la mejor manera el objeto capturado, se debe tomar en cuenta el periférico de salida para que la persona que esté utilizando el prototipo sepa que está funcionando de manera adecuada.

Pruebas

Se realizará pruebas del correcto funcionamiento del software, como también pruebas de integración para poder verificar el correcto funcionamiento colectivo tanto del software como hardware, además se realizará pruebas del sistema para verificar que no exista ningún error en las interfaces, código, tiempo de ejecución,

CAPITULO 1

ESTADO DEL ARTE

1.1. Discapacidad Visual

“La discapacidad visual se define como la dificultad que presentan algunas personas para participar en actividades propias de la vida cotidiana, que surge como consecuencia de la interacción entre una dificultad específica relacionada con una disminución o pérdida de las funciones visuales y las barreras presentes en el contexto en que desenvuelve la persona” (Ministerio de Educación de Chile, 2007).

La OMS en 1992 definió a una persona con Baja Visión, “aquella con una incapacidad en la función visual aun después de tratamiento y/o refracción común, con una agudeza visual en el mejor ojo de 0.3 a percepción de luz o con un campo visual inferior a 10° desde el punto de fijación, pero que se use, es decir, potencialmente capaz de usar la visión para la planificación y ejecución de tareas” (Organización Mundial de la Salud, 1994).

1.2. Discapacidad visual en Ecuador

En Ecuador según el Consejo Nacional para la Igualdad de Discapacidades (CONADIS), existen un total de 485.325 personas con discapacidad registradas en el Registro Nacional de Discapacidades, de las cuales el 11.66% tienen discapacidad visual (CONADIS, 2020).

En la tabla 1, se muestra el número de personas registradas en Ecuador que presentan algún tipo de discapacidad:

Tabla 1. Número de personas con discapacidad.

| Tipo de discapacidad | Personas con discapacidad |
|-----------------------------|----------------------------------|
| Física | 226.347 |
| Intelectual | 108.312 |
| Auditiva | 67.929 |
| Visual | 56.570 |
| Psicológica | 26.167 |

Nota: Personas con discapacidad registradas.

Elaborado por: Ronald Cruz.

1.3. Sistemas Embebidos

“El sistema embebido hace referencia a todo circuito electrónico digital capaz de realizar operaciones de computación, generalmente en tiempo real, que sirven para cumplir una tarea específica de un producto” (Arriarán, 2015).

Los sistemas embebidos no son semejantes a los sistemas de cómputo usados en las laptops o computadoras de escritorio, estos tienen recursos limitados y sus aplicaciones son específicas y esto les hace muy útiles en diferentes campos: automotriz, domótica, reproductores multimedia, equipos médicos, entre otros.

Contiene un microprocesador que es capaz de ejecutar tareas a una velocidad determinada, una característica principal de un sistema embebido es su bajo consumo de energía, ya que al ser compactos están diseñados para realizar tareas específicas sin mayor esfuerzo.

1.4. Microcomputador

“Es una computadora cuya unidad central es un microprocesador, es un sistema completo que lleva, además del microprocesador, una memoria y controladores de entrada y salida, para conectarlo como periféricos exteriores. Es pequeña a comparación de los servidores, estaciones de trabajo” (Norris, 1984).

1.5. Tipos de Microcomputadores

1.5.1. Tarjeta Raspberry Pi

“Raspberry Pi es un ordenador de placa única, desarrollada en Reino Unido por la Fundación Raspberry Pi, para que las personas puedan aprovechar el poder de la informática y las tecnologías digitales para el trabajo, resolver problemas y expresarse creativamente” (Fundación Raspberry Pi, 2020).

1.5.2. ASUS Tinker Board

Tinker board es una minicomputadora de una sola tarjeta de un tamaño muy pequeño que ofrece excelente rendimiento, a su vez aprovecha la compatibilidad con los demás componentes. “La Tinker Board ofrece a los creadores, a los entusiastas de IoT, a los aficionados y a todos aquellos que gusten hacer las cosas por si mismos una plataforma capaz para construir y llevar sus ideas a la realidad” (ASUS, tinker board: tinker your way to the future, 2019).

1.5.3. UDOO X86

“UDOO X86 II puede ejecutar todo el software disponible para el mundo de la PC, desde juegos hasta transmisión de video, desde editores gráficos hasta plataformas de desarrollo profesional, además de todo el software o el mundo Arduino, incluidos todos los bocetos, bibliotecas y Arduino. Está basado en procesadores x86 de nueva generación Quad Core de 64 bits fabricados por Intel. Maravillosos procesadores concentrados en 14 nm, con una cantidad de consumo de energía de 5 o 6 vatios” (SECO SPA, 2019).

1.5.4. Orange Pi

Es un microcomputador que utiliza código abierto diseñado y fabricado por la empresa Shenzhen Xunlong Software CO., Limited. Se pueden ejecutar varios sistemas operativos como: Android, distribuciones de Linux (Ubuntu, Debian), también el sistema

operativo Raspberry Pi OS. Utiliza una memoria RAM según su modelo: desde 256MB hasta 2 GB DDR3 SDRAM, según el modelo puede tener puertos USB, ethernet, WIFI, bluetooth, puerto HDMI, puede soportar la tecnología 2G para poder trabajar con Internet de las Cosas.

“Se puede construir un ordenador, un centro multimedia (Smart TV), un sistema de vigilancia, un servidor de impresoras inalámbrico, un marco de fotos digital, un fotomatón ultra portátil para fiestas y eventos, un servidor de archivos, juegos, instalar Android, instalar Linux. Y mucho más, porque Orange Pi 2 se basa en código abierto” (CO., Shenzhen Xunlong Software, 2020).

1.5.5. Cuadro comparativo entre microcomputadores

Tabla 2. Cuadro comparativo de microcomputadores.

| | ASUS Tinker Board | Raspberry Pi 4 B | UDOO X86 | Orange Pi |
|------------------------|---------------------------|---|--------------------------------------|--------------------------------------|
| Procesador | Rockchip Quad-Core RK3288 | Broadcom BCM2711, @ 1.5GHz | 2.56 GHZ INTEL PENTIUM N3710 | H3 Quad-core Cortex-A7 H.265/HEVC 4K |
| RAM | 2GB Dual Channel DDR3 | 4GB LPDDR2 SDRAM | 8 GB DDR3L DUAL CHANNEL | 1GB DDR3 |
| Video Interface | HD MI | Full-size HDMI | 1 x HDMI 2 x mini DP++ | HDMI |
| Audio | RTL ALC4040 CODEC | 4-pole stereo output and composite video port | Microphone+Headphone combo connector | MIC |
| LAN | RTL GB LAN | Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps) | Gigabit Ethernet LAN interface | 10/100 Ethernet RJ45 |
| Wi-Fi | 802.11 b/g/n | 2.4GHz and 5GHz IEEE 802.11.b/g/n/a | M.2 Key E slot opcional | No |

| | | | | |
|----------------------------|----------------------|--|------------|--------------------------|
| | | c wireless LAN | | |
| Bluetooth | Bluetooth V4.0 + EDR | Bluetooth 4.2, BLE | No | No |
| USB | 4 x USB 2.0 | 4 USB 2.0 | 3x USB 3.0 | 3 USB 2.0, 1 USB 2.0 OTG |
| CSI puerto cámara | 1 puerto CSI | 1puerto CSI para Raspberry Pi camera | No | No |
| DSI puerto pantalla | 1 puerto DSI | 1 puerto DSI para Raspberry Pi touchscreen display | No | No |
| Incluye PoE | No | Si | No | No |
| Precio | \$100 | \$60 | \$90 | \$50 |

Nota: Cuadro comparativo de microcomputadores.
Elaborado por: Ronald Cruz.

Después de la comparación en la Tabla 2, se procedió a la elección del microcomputador Raspberry Pi 4 B, porque cuenta con el puerto CSI exclusivamente para el módulo de la cámara, además cuenta con un procesador Broadcom BCM2711 y una memoria RAM de 4 GB que permite el procesamiento de las imágenes en tiempo real, por otra parte, cuenta con la documentación y ejemplos de procesamiento de imágenes.

1.6. Lenguaje de programación

“Un lenguaje de programación se usa para escribir programas; estos constan de secuencias de instrucciones que se codifican y a su vez, son traducidos para que las computadoras los entiendan” (Díaz, 2019).

1.6.1. Lenguaje de programación Python

Python es un lenguaje multiplataforma, lo que significa que puede usarse en multitud de sistemas distintos. Funciona en ordenadores con sistemas operativos Linux, BSD, Apple, Windows, entre otros, existen otras versiones para otros dispositivos, como terminales telefónicos inteligentes. Python dispone por medio del uso de bibliotecas, de herramientas

para aprovechar las posibilidades concretas que le brinda cada plataforma, pero también es posible escribir programas evitando el uso de esas bibliotecas específicas, de modo que ese programa funcione indistintamente en cualquier ordenador. Es un software libre, se distribuye bajo licencia “Python Software Foundation License”. Significa que se distribuye gratuitamente y no necesita del pago de licencias o royalties para su uso, ya sea privado o comercial (Gutiérrez, 2016).

1.6.2. Thonny Python

Thonny es un IDE de Python, es un proyecto Open Source lo que quiere decir que se lo puede descargar de forma gratuita, ideal para aprender a usar el lenguaje de programación, cuenta con una interfaz amigable e intuitiva con el usuario, disponible en las plataformas como Windows, Linux (Fedora), Mac. Uno de los puntos más destacables de Thonny es que cuenta con un depurador para poder encontrar y leer fácilmente los errores en los programas que se están creando.

1.6.3. PyCharm

PyCharm se considera uno de los entornos de desarrollo más completos para el lenguaje de programación Python, es un IDE multiplataforma ya que se lo puede usar en Windows como en Linux, creado por JetBrains. Las herramientas que ofrece este IDE están adecuadas y mejoradas para el desarrollo de aplicaciones en Python.

1.6.4. Qt Designer

Es una herramienta para poder crear interfaces gráficas para el usuario, es una herramienta de fácil uso, nada más con arrastrar y soltar componentes como: botones, cuadros de texto, checkbox, entre otros, para crear una interfaz gráfica de usuarios de forma rápida, esta herramienta necesita al lenguaje de programación Python versión 3 o superior para poder funcionar correctamente.

1.7. Sistema operativo Raspberry Pi OS

Es un sistema operativo basado en Linux en su distribución Debian, este se distribuye de forma libre bajo licencia GLP, el mismo es creado para ser usado en el hardware de Raspberry Pi, utiliza programas básicos y utilidades, fue creado por desarrolladores fans de la tarjeta Raspberry Pi con el fin que se la pueda utilizar para la educación y proyectos.

Este sistema operativo es de muy fácil instalación, el mismo cabe en una Memoria Micro SD, cuenta con un muy buen soporte por parte de la Fundación Raspberry Pi, como también por parte de la comunidad Debian.

1.8. Librería OpenCV

OpenCV es una librería de código abierto creada por la empresa Intel, es multiplataforma, que se distribuye bajo licencia BSD (Berkeley Software Distribution), esta librería se utiliza para el tratamiento y análisis de imágenes mediante algoritmos que utilizan la inteligencia artificial.

Entre los usos más importantes de la mencionada librería está la detección facial y de objetos, en el ámbito de la seguridad, fotografía, OpenCV cuenta con más de 2500 algoritmos para el análisis y el tratamiento de las imágenes, se la distribuye de forma gratuita y se la puede usar para fines comerciales y educativos.

1.9. Tesseract

Tesseract es una herramienta de código abierto desarrollado inicialmente por HP, ahora es mantenido por Google, es un motor de reconocimiento de imágenes, se lo pueden usar directamente el código los programadores o a su vez se puede instalar como una aplicación, admite una gran variedad de idiomas, entre ellos: inglés, español, hindú.

1.10. Modulo cámara Raspberry Pi IR-CUT

Este módulo de cámara soporta las versiones de la tarjeta Raspberry Pi en sus modelos B/ B+, A+, 3, 2, 1, cuenta con una placa IR LED infrarrojo que soporta la función de visión nocturna.

Trabaja en una interfaz dedicada a través del bus CSI de Raspberry Pi, cuenta con un sensor de cámara web OV5647 con resolución de 5 MP y puede grabar videos a 1080p @ 30 fps.

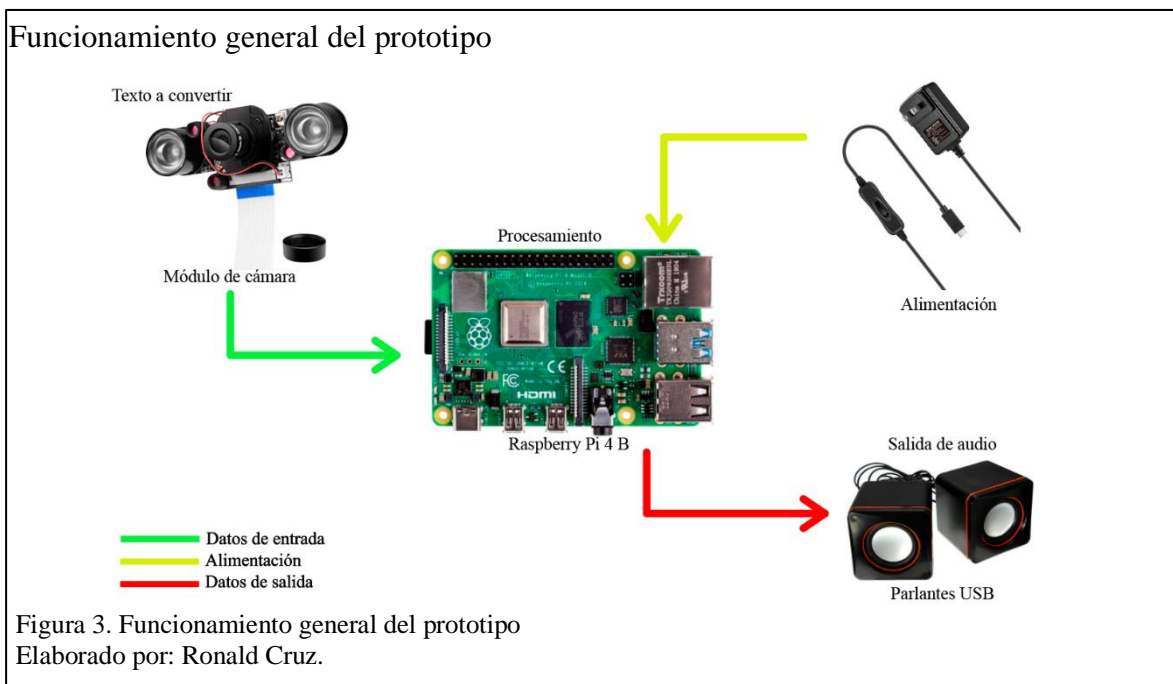
CAPITULO 2

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO

En el presente capítulo, se detallará el diseño y la construcción del sistema embebido con los diferentes componentes electrónicos en el prototipo, así como su funcionamiento y estructura tanto del hardware como del software.

2.1. Diagrama general de funcionamiento

En la figura 3, se muestra el diseño y construcción del prototipo del sistema para reconocimiento de texto y conversión a audio, utilizando Raspberry PI para personas no videntes.

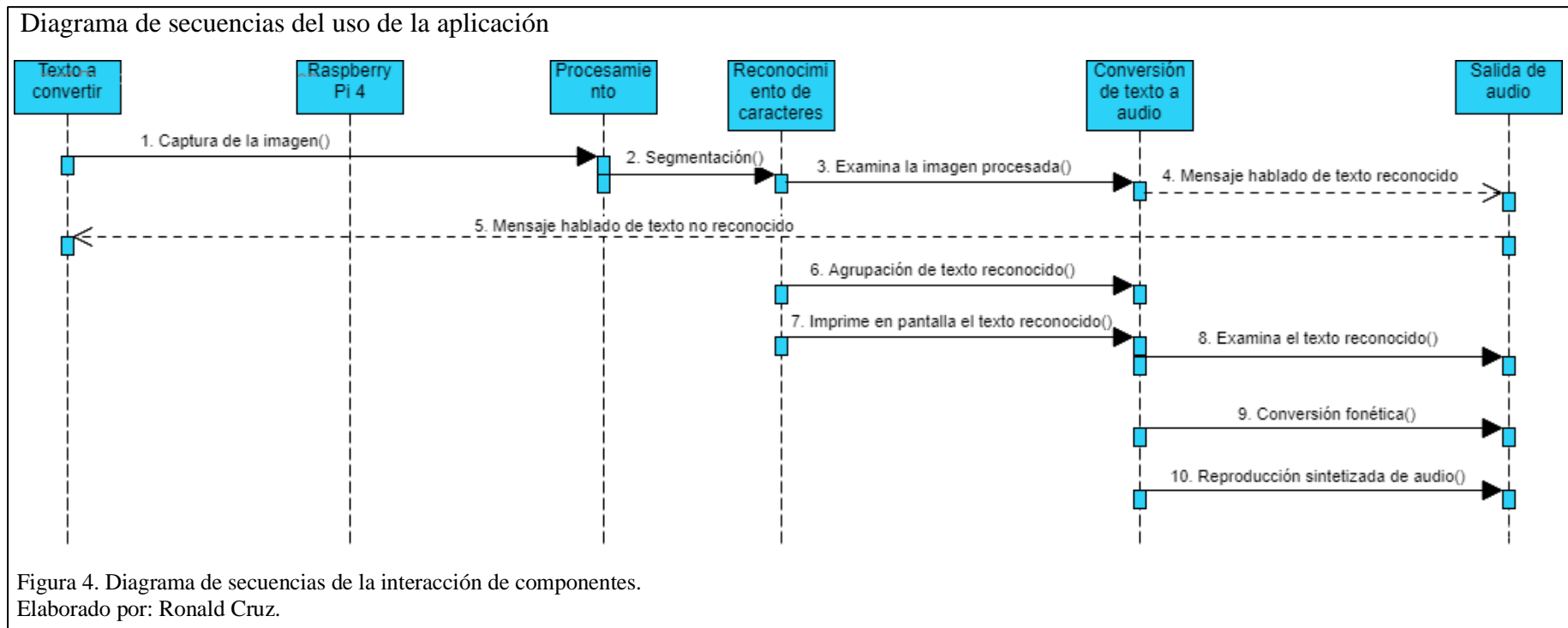


En el diagrama se muestra la tarjeta Raspberry Pi 4 B que es la encargada de controlar todo el sistema, y es alimentada con una fuente de poder de 5V, este sistema está siendo utilizado para a entrada y salida de datos digitales, el bloque que corresponde a la entrada se compone de un módulo de cámara que cuenta con dos sensores LED infrarrojo, esta permite

el soporte de visión nocturna, este módulo se conecta a la tarjeta Raspberry Pi por medio de un bus CSI que envía los datos capturados hacia la misma.

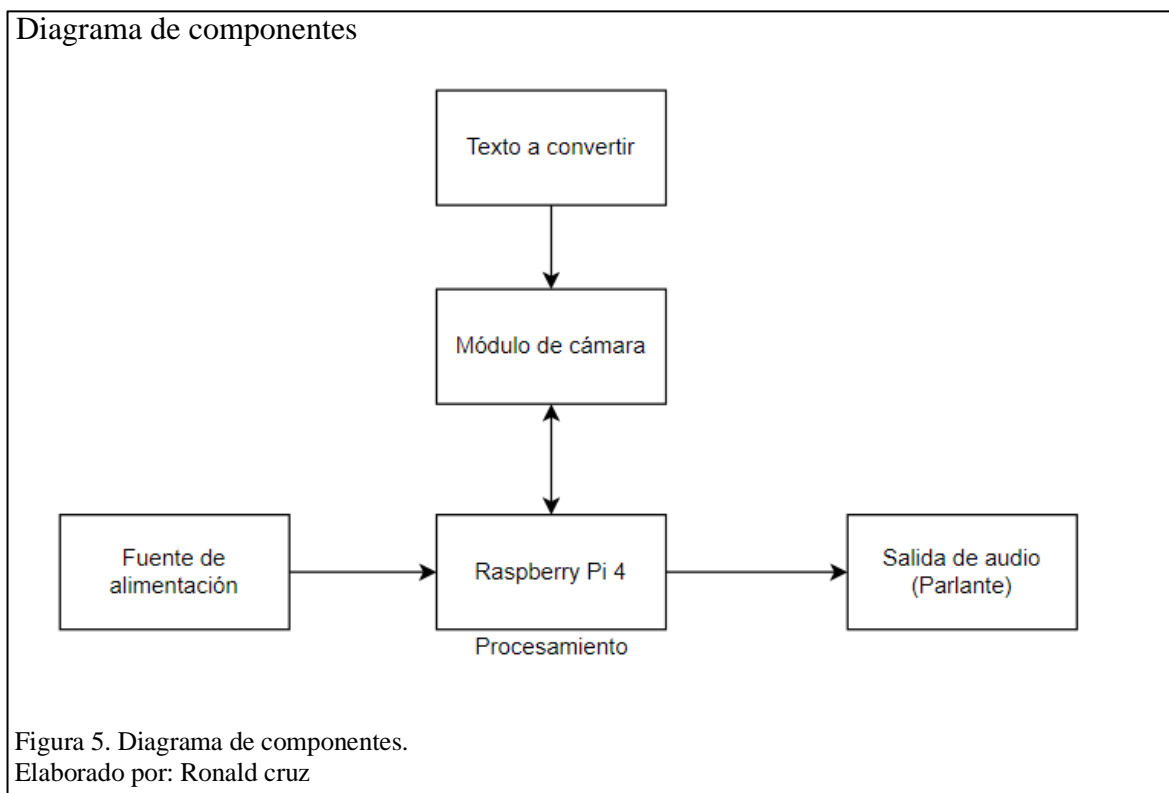
La tarjeta Raspberry Pi es la encargada de procesar los datos ingresados por medio de la cámara, entonces los datos procesados son enviados hacia los parlantes USB que guían a la persona no vidente mediante una voz que indica lo que dice el texto escrito.

2.1.1. Diagrama de secuencias



Como se puede observar en la figura 4. Como primer punto se realiza la captura de la imagen por medio del módulo de cámara que está conectada con la tarjeta Raspberry Pi 4, como siguiente paso se realiza una segmentación de la imagen capturada, esto quiere decir que extrae las regiones de interés de la imagen. Después para el reconocimiento óptico de caracteres (OCR) se examina la imagen procesada y compara los caracteres encontrados con los caracteres existentes en la librería de Python, como paso siguiente la aplicación enviará un mensaje de voz que diga “Texto reconocido”, caso contrario si no reconoce ninguna palabra enviará un mensaje que dice “no existe texto” y se reinicia el programa. Después realiza una agrupación del texto reconocido y lo prepara para ser leído, luego imprime en pantalla el texto que reconoció, en la conversión a audio examina el texto reconocido antes de ser leído, finalmente reproduce con una voz sintetizada el texto reconocido anteriormente.

2.1.2. Diagrama de componentes



El diagrama de componentes como se puede ver en la figura 5. Está constituido por:

Raspberry pi 4: Es el ordenador de tamaño reducido que será encargado de realizar el procesamiento y almacenar la información, dicha información se la obtendrá del módulo de cámara.

Módulo de cámara Raspberry Pi: encargada de capturar las imágenes una vez que la aplicación funcione.

Fuente de alimentación: La tarjeta Raspberry Pi 4 según su ficha técnica, utiliza una alimentación de 5.1 V con una corriente de 3.0 A.

Salida de audio: Reproduce el texto convertido en voz sintetizada.

2.2. Raspberry Pi 4 B

Es un microcomputador que está diseñado para crear pequeños proyectos y prototipos, estimula la investigación y la imaginación, además, su diseño y tamaño permite acoplarse al prototipo del sistema para ayudar a las personas no videntes.

Tarjeta Raspberry Pi 4B



Figura 6. Raspberry Pi 4B.
Elaborado por: (Raspberry Pi Foundation, 2018)

Tabla 3. Asignación de partes para el prototipo del sistema.

| Distribución de partes de la Raspberry Pi 4 B | Descripción |
|---|-----------------------------------|
| Micro USB Power | Alimentación |
| SD Card Slot | Sistema Operativo RASPBerry PI OS |
| CSI Camera Connector | Cámara Raspberry PI IR-CUT |
| Audio OUT | Parlantes |
| Pines GPIO | Ventilación |

Nota: Utilización de partes de la Raspberry Pi para el prototipo.

Elaborado por: Ronald Cruz.

2.3. Cámara Raspberry Pi

El módulo cuenta con un sensor OV5647 con 5 MP, que permite grabar videos a 1080p y fotografías HD, permite trabajar con todos los modelos Raspberry Pi gracias al conector CSI, además cuenta con un sensor infrarrojo que permite capturar y procesar fotos nocturnas. En la tabla 3.2 se detallan las características técnicas del módulo de cámara.

Cámara IR-CUT

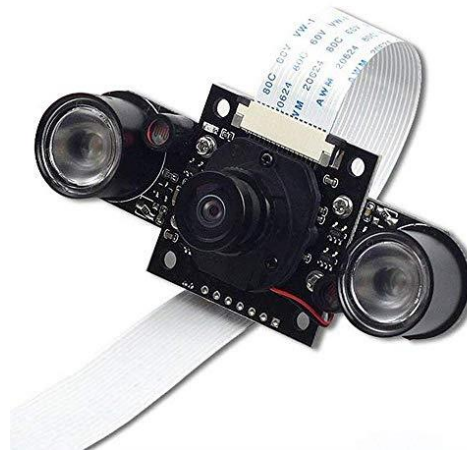


Figura 7. Módulo de cámara IR-CUT.

Elaborado por: (Arducam, 2020).

Tabla 4. Características técnicas del módulo de cámara.

| | |
|--------------------------------|--|
| Resolución | 5MP-videos y fotos HD |
| Alimentación | 3.3 V |
| Tamaño óptico del lente | ¼ pulgadas |
| Imagen | Alta velocidad, elimina la distorsión de la luz del día. |
| Sensores | LED Infrarrojo, soporta la visión nocturna. |

Nota: Características técnicas del módulo de cámara IR - CUT.

Elaborado por: Ronald Cruz.

2.4. Parlantes USB Genius SP-U150X

En la figura 8, se detallan las características técnicas de los parlantes USB que se usarán en el prototipo.

Parlantes USB Genius SP-U150X.



Figura 8. Parlantes USB Genius SP-U150X.
Elaborado por: Ronald Cruz.

Tabla 5. Características técnicas de parlantes USB.

| | |
|---------------------|---------------------------------|
| Potencia | Dos altavoces de 2 Watts |
| Conectividad | USB |
| Frecuencia | 100 Hz a 22.000 Hz |
| Conector | Jack de 3.5 |

Nota: Características técnicas de los parlantes USB Genius SP-U150X.
Elaborado por: Ronald Cruz.

2.5. Memoria Micro SD

Micro Secure Digital, es un chip de almacenamiento usado en dispositivos portátiles, en el caso del proyecto que se está desarrollando se lo usará en la tarjeta Raspberry Pi 4 B para la instalación del sistema operativo Raspbian, el lenguaje de programación Python y el procesamiento de imágenes.

Memoria Micro SD.



Figura 9. Memoria Micro SD.
Elaborado por: Ronald Cruz.

2.6. Hardware utilizado para el desarrollo del prototipo

En la figura 10, se muestra el hardware que se utilizará para el desarrollo del prototipo, consta de la tarjeta Raspberry Pi, el módulo de cámara y una carcasa de protección para la tarjeta antes mencionada.

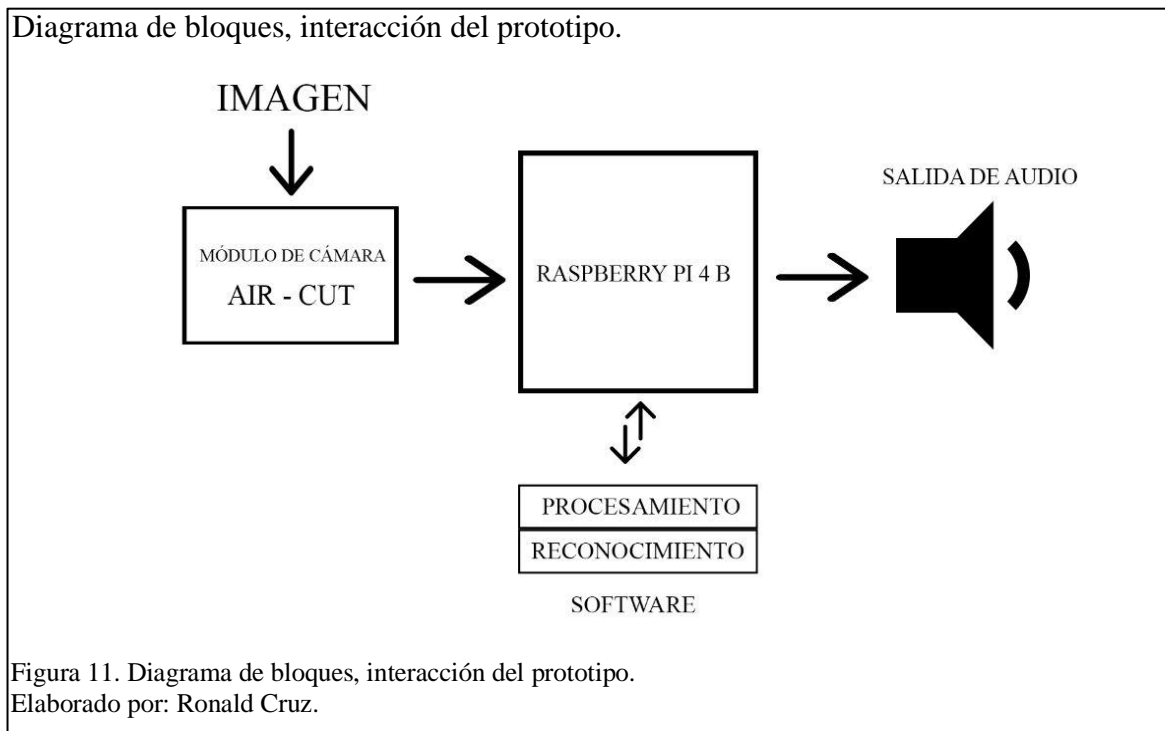
Hardware armado que se usará en el desarrollo del prototipo.



Figura 10. Hardware utilizado para el desarrollo del prototipo.
Elaborado por: Ronald Cruz.

2.7. Diagrama de bloques

A continuación, en la figura 11, se muestra el diagrama de bloques sobre el funcionamiento y la interacción del prototipo a desarrollar.



2.8. Instalación del sistema operativo Raspberry Pi OS

Primero se debe descargar la aplicación de escritorio Raspberry Pi Imager, disponible para las plataformas Microsoft Windows, Linux en su distribución Ubuntu, MacOS de la página: (Raspberry Pi Foundation, 2020).

Opciones de descarga de Raspberry Pi Imager para las plataformas: Windows, Ubuntu y MacOS

Instale el sistema operativo Raspberry Pi usando Raspberry Pi Imager

Raspberry Pi Imager es la forma rápida y fácil de instalar Raspberry Pi OS y otros sistemas operativos en una tarjeta microSD, lista para usar con su Raspberry Pi. [Mire nuestro video de 40 segundos](#) para aprender cómo instalar un sistema operativo usando Raspberry Pi Imager.

Descargue e instale Raspberry Pi Imager en una computadora con un lector de tarjetas SD. Coloque la tarjeta SD que usará con su Raspberry Pi en el lector y ejecute Raspberry Pi Imager.

[Descarga para Windows](#)

[Descarga para macOS](#)

[Descarga para Ubuntu para x86](#)

Para instalar en **Raspberry Pi OS**, escriba

```
sudo apt install rpi-imager
```

en una ventana de Terminal.

Figura 12. Opciones de descarga de Raspberry Pi Imager.
Elaborado por Ronald Cruz.

Una vez descargada la aplicación de escritorio, se procede a la instalación de la misma, en este caso se usará para la plataforma Microsoft Windows.

Instalación de la aplicación de escritorio Raspberry Pi Imager.

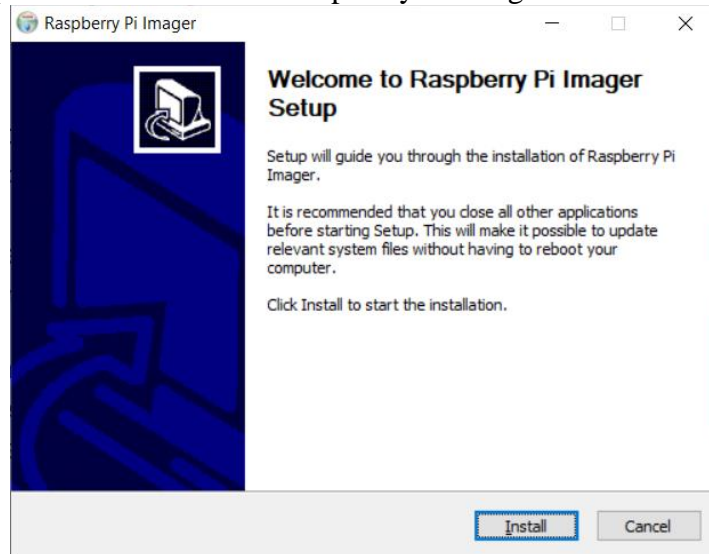


Figura 13. Instalación Raspberry Pi Imager.
Elaborado por: Ronald Cruz.

Luego de haber instalado Raspberry Pi Imager, se procede con la instalación del Sistema Operativo de Raspberry, para ello se debe tener a la mano una tarjeta Micro SD, para la elaboración del proyecto se usará una tarjeta micro SD de 8 GB, una ranura para la tarjeta mencionada.

Selección de Raspberry Pi OS y la tarjeta Micro SD para su instalación.

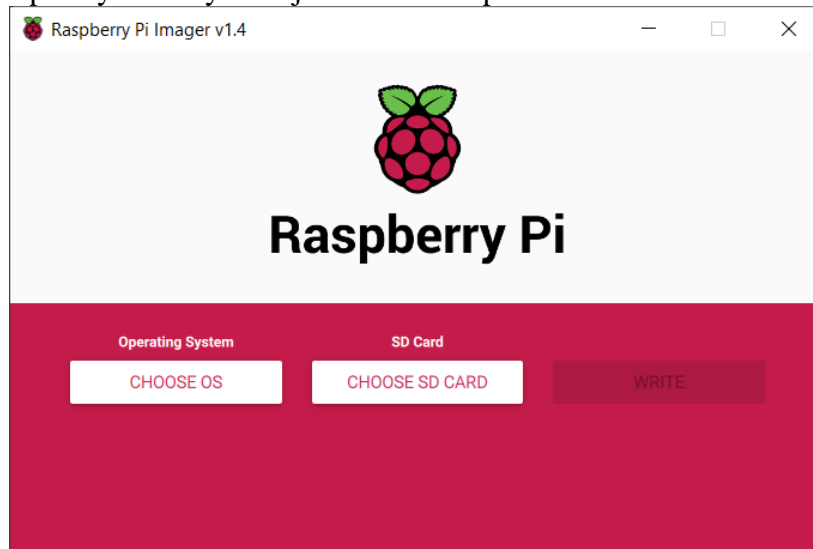
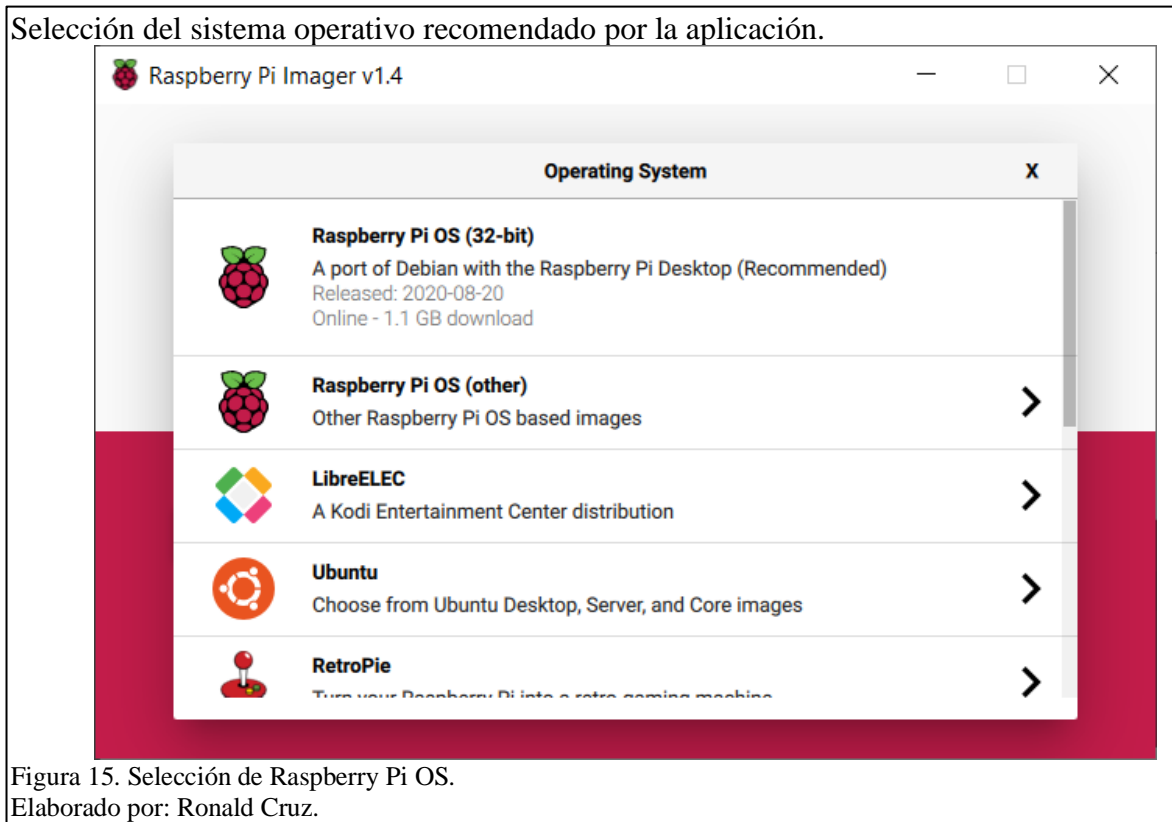


Figura 14. Instalación de Raspberry Pi OS.
Elaborado por: Ronald Cruz.

A continuación, se selecciona el sistema operativo que se va a utilizar, en este caso se usará el Raspberry Pi OS que recomienda la aplicación, después se selecciona la memoria

Micro SD que se muestra, luego de seleccionar el sistema operativo y la tarjeta de memoria se procede a escribir el sistema operativo en la misma en la opción Write, mostrará un mensaje preguntando que se va a borrar todo lo que se encuentre guardado y si está seguro para continuar.



Selección de la tarjeta Micro SD de 8 GB

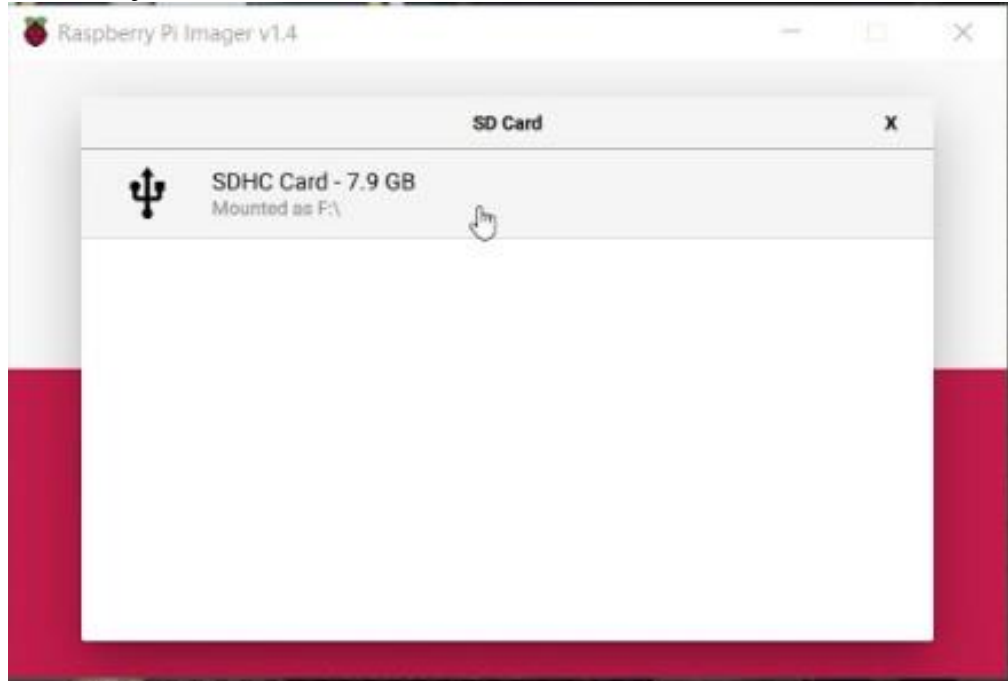


Figura 16. Selección de la tarjeta Micro SD.
Elaborado por: Ronald Cruz.

Instalación de Raspberry Pi OS en la tarjeta Micro SD.



Figura 17. Instalación de Raspberry Pi OS en la tarjeta Micro SD.
Elaborado por Ronald Cruz.

Luego de haber instalado y verificado la instalación del sistema operativo, mostrará un mensaje indicando que ha sido instalado con éxito, a continuación, se inserta la tarjeta Micro SD en la Raspberry Pi y se procede a encenderla, mostrará la primera pantalla donde

pedirá que se realice la configuración básica: idioma, usuario, contraseña, conectarse a una red y actualización de software.



Dentro de la configuración del sistema operativo Raspberry Pi OS se incluye cambiar la localización, idioma, las versiones actuales de este sistema operativo han simplificado mucho la configuración del mismo ya que cuenta con una ventana gráfica de la configuración lo cual lo hace más amigable a la vista del usuario.

Configuración de país, zona horaria, idioma.



Figura 19. Configuración de país, zona horaria, idioma Raspberry Pi OS.
Elaborado por: Ronald Cruz.

El usuario y contraseña permite al usuario protegerse de amenazas que se presentan en el día a día, por eso es muy importante un cambio de contraseña que sea segura y robusta para evitar contratiempos y ataques provenientes de internet.

Cambio de contraseña en Raspberry Pi OS.

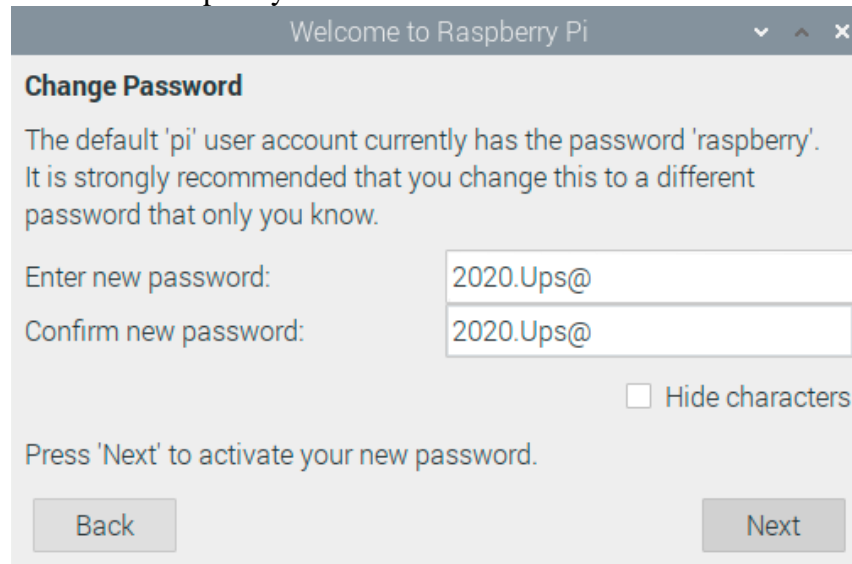
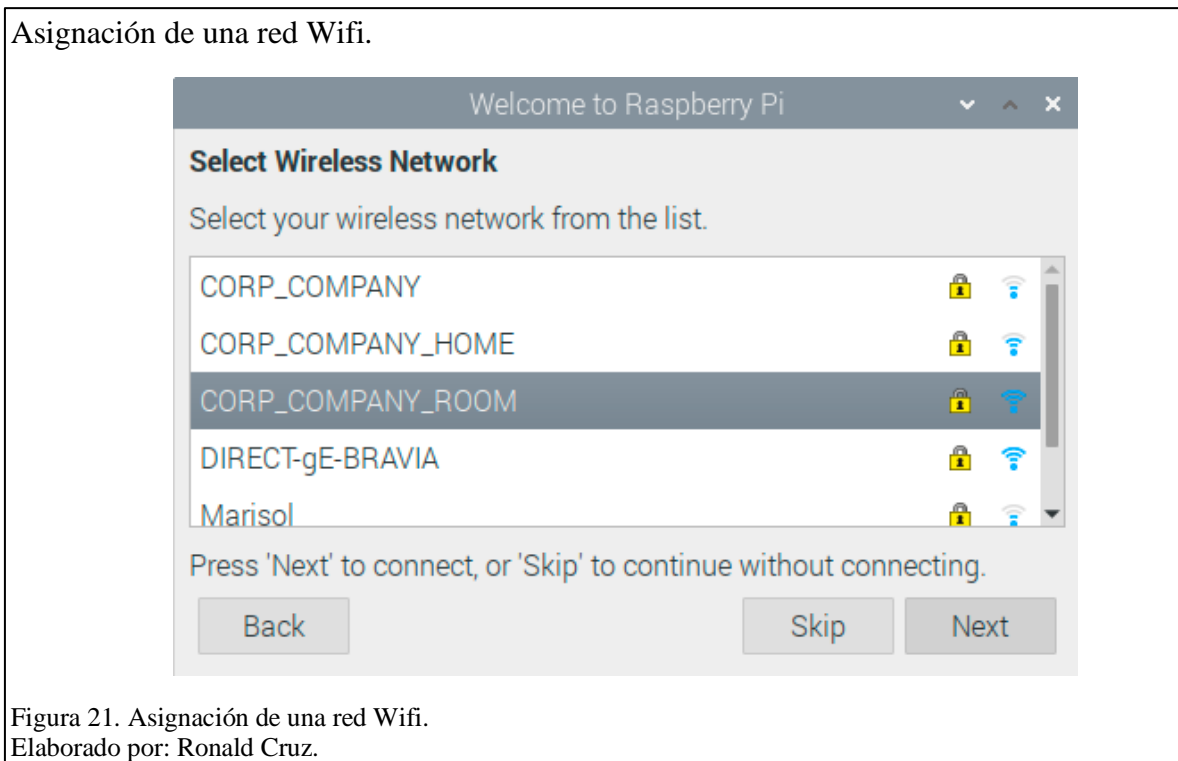


Figura 20. Cambiando la contraseña en el sistema Raspberry Pi OS.
Elaborado por: Ronald Cruz.

Se debe conectar la tarjeta Raspberry Pi a internet para poder actualizar su sistema operativo e instalar aplicaciones y componentes que son requeridos para el proyecto.

Se puede conectar a una red mediante un cable o a través de una conexión Wifi, en este caso se conectará a la red por medio de Wifi.



Es importante actualizar el sistema operativo para que la tarjeta Raspberry Pi funcione correctamente, por otra parte, poder obtener nuevos complementos o funciones que agregue la nueva versión, además para evitar las posibles fallas que se presenten y se vea comprometida la información que se encuentre guardada.

Actualización del sistema operativo.



Figura 22. Actualización del sistema operativo Raspberry Pi OS.
Elaborado por: Ronald Cruz.

2.9. Configuración SSH en Raspberry Pi OS

El servicio SSH se encuentra deshabilitado por defecto en la tarjeta Raspberry Pi, se debe habilitar después del primer arranque del sistema operativo Raspberry Pi OS.

Para habilitar el servicio SSH desde la terminal se debe escribir el comando: `sudo raspi-config`.

Comando para iniciar la configuración en Raspberry Pi OS.

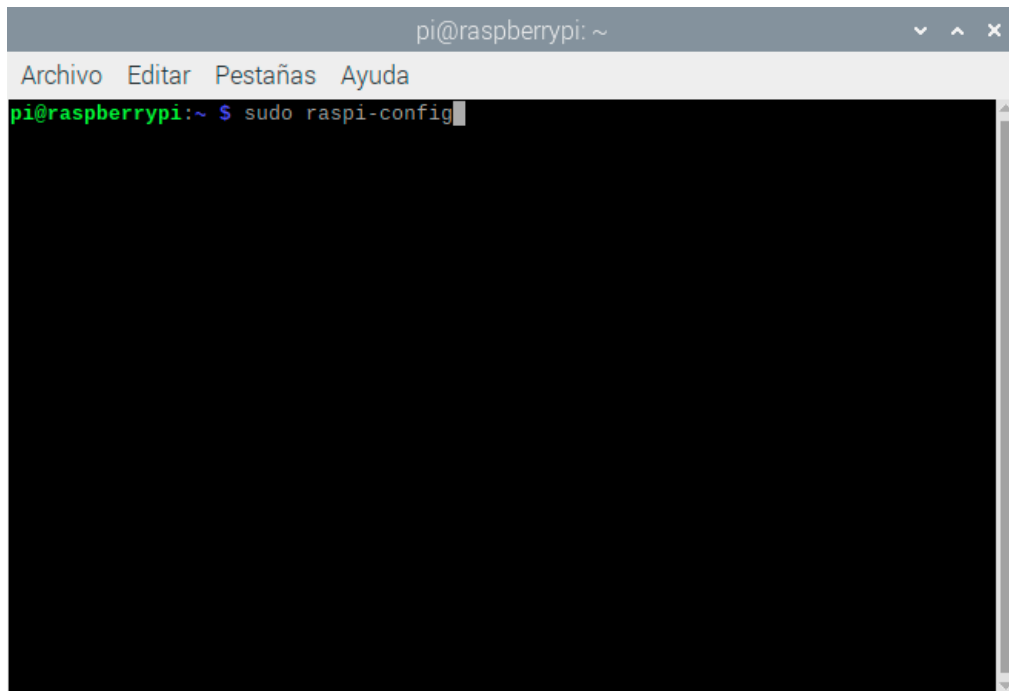


Figura 23. Comando para iniciar la configuración en Raspberry Pi OS.
Elaborado por: Ronald Cruz.

Se abrirá la ventana Configuration Tools del sistema operativo Raspberry Pi OS, a continuación, seleccionar la opción Interface Options.

Ventana de configuración en Raspberry Pi OS.

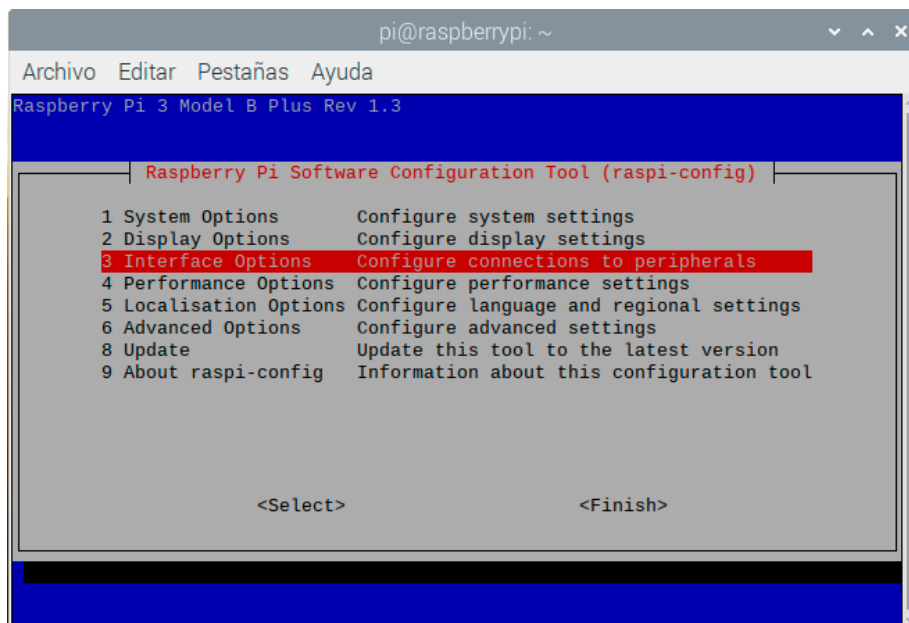
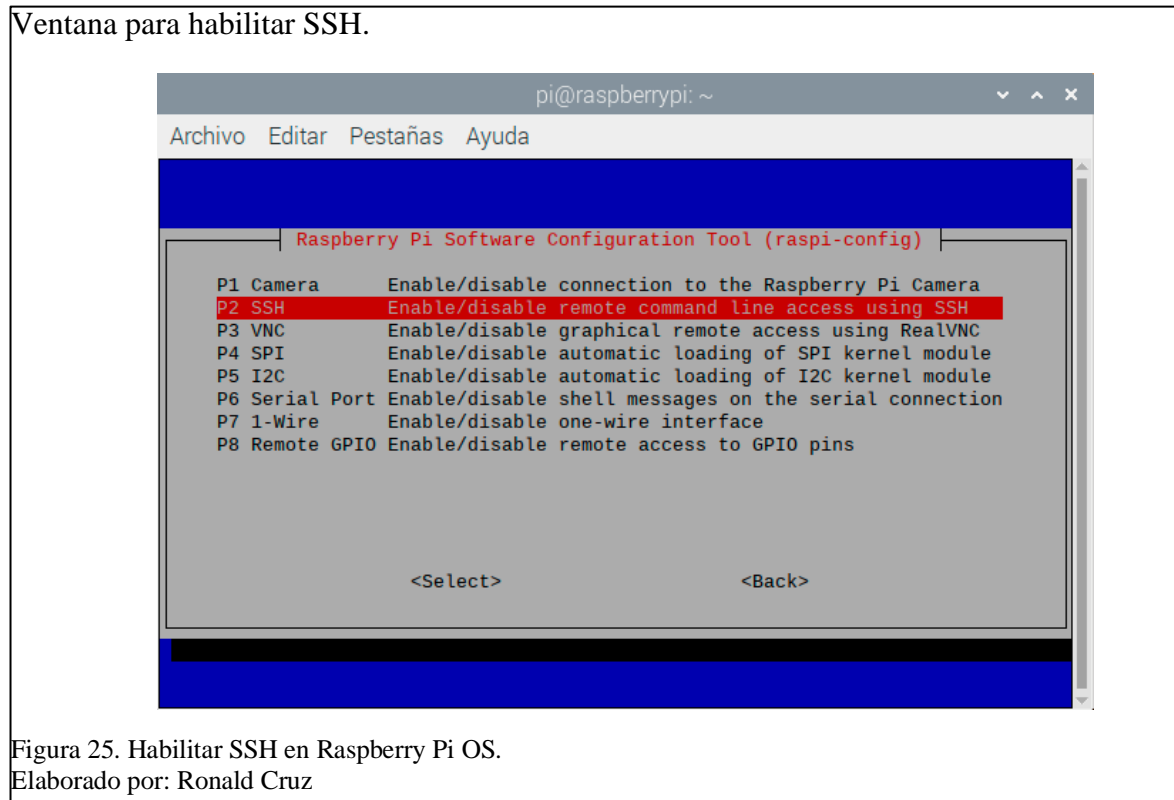


Figura 24. Configuración en Raspberry Pi OS.
Elaborado por: Ronald Cruz.

Como siguiente paso navegar y seleccionar la opción SSH, preguntará si está seguro habilitar el servicio, presionar si, el servicio quedará habilitado.



Para iniciar el acceso remoto desde Microsoft Windows 10 se usará la aplicación Putty, para ello se descargará de la siguiente página: (Chiark Organization, 2020), una vez descargado se procede a abrir el mismo y se ingresa la configuración de conexión: dirección IP, puerto de conexión, tipo de conexión, sesión guardada.

Probando conexión con Putty a Raspberry Pi OS.

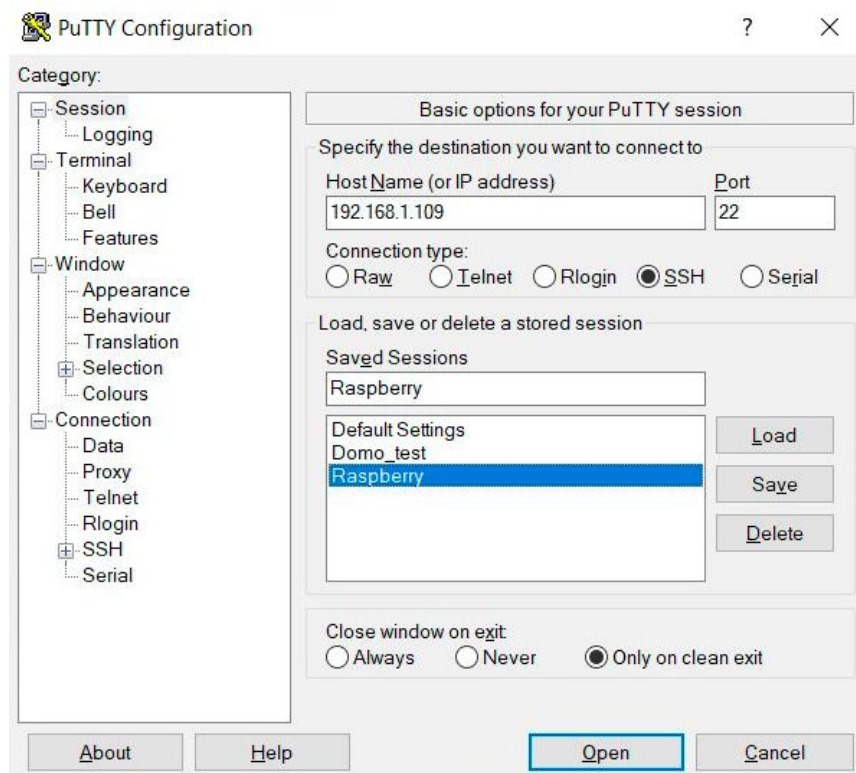


Figura 26. Conexión desde Windows a Raspberry Pi OS.

Elaborado por: Ronald Cruz.

Si por primera vez que se conecta al servidor desde su computador, mostrará el siguiente resultado. Aceptar la conexión haciendo clic en Sí.

Alerta de seguridad de conexión por primera vez.



Figura 27. Alerta de seguridad de Putty.

Elaborado por: Ronald Cruz.

Una vez que la sesión SSH se encuentre abierta, mostrará la ventana desde la terminal pidiendo usuario y contraseña, una vez ingresadas las credenciales mostrará un mensaje indicando que ya se encuentra conectado.



Figura 28. Conexión exitosa entre Windows a Raspberry Pi OS.
Elaborado por: Ronald Cruz.

2.10. Habilitar VNC en Raspberry Pi OS

Para habilitar VNC en la terminal se debe digitar el comando: `sudo raspi-config`.

Configuración para habilitar VNC.

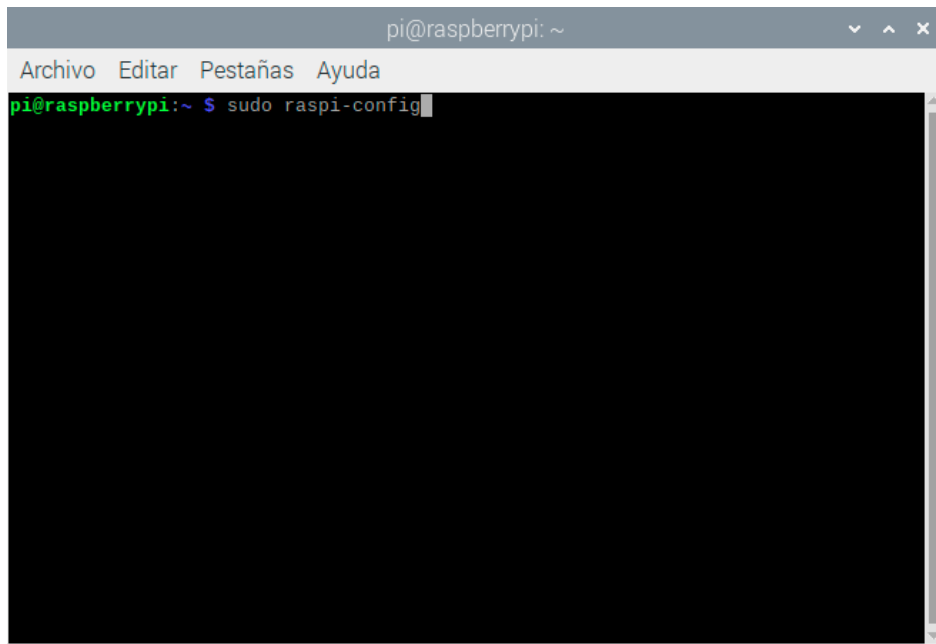


Figura 29. Ingresar a la configuración para habilitar VNC.
Elaborado por: Ronald Cruz.

Mostrará la pantalla Configuration Tools de Raspberry Pi OS, donde se debe seleccionar la opción Interface Options, a continuación, seleccionar la opción VNC, preguntará si desea habilitar esta opción, presionar Yes, quedará habilitado VNC.

Habilitación exitosa de VNC.

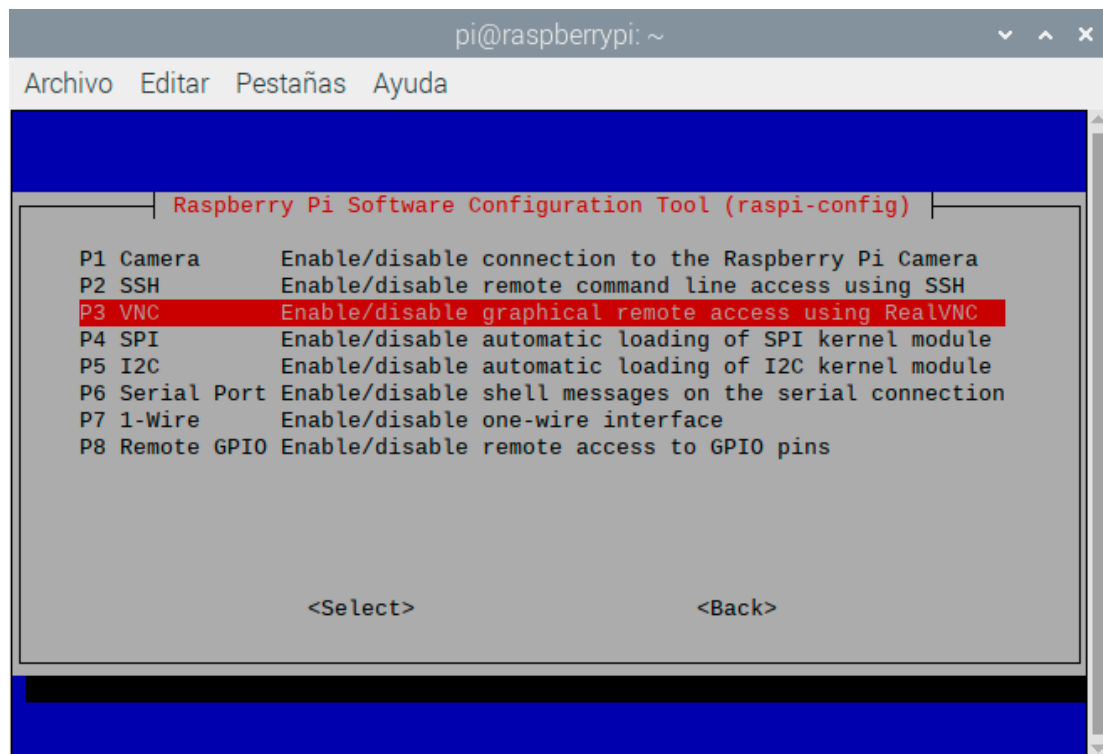


Figura 30. Habilitación completada de VNC.
Elaborado por: Ronald Cruz.

Por otra parte, en Windows se procede a descargar e instalar la aplicación VNC Viewer de la siguiente dirección: (RealVNC Limited, 2020), una vez instalado abrirlo y se ingresa la dirección IP de la tarjeta Raspberry Pi.

Conexión remota vía VNC Viewer.

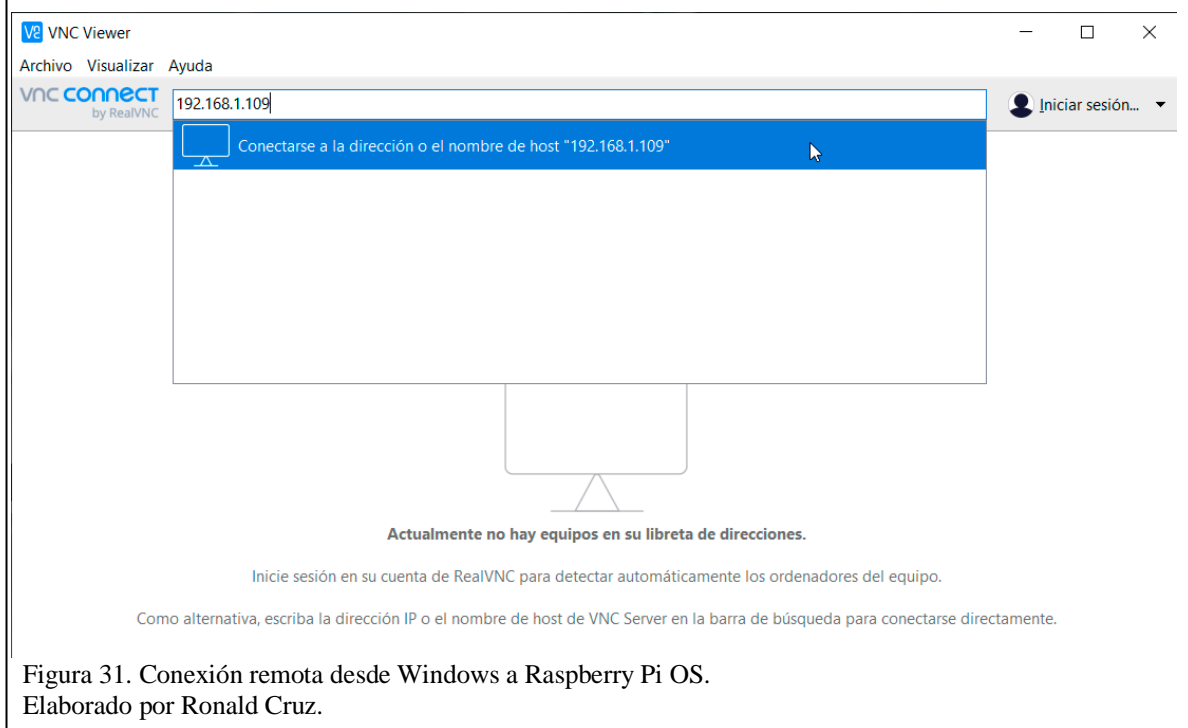


Figura 31. Conexión remota desde Windows a Raspberry Pi OS.
Elaborado por Ronald Cruz.

Pedirá que se ingrese sus credenciales, seguidamente mostrará un mensaje indicando que no existió una conexión anterior y si desea realmente conectarse, presionar en el botón Continuar.

Ingreso de credenciales para la conexión remota.

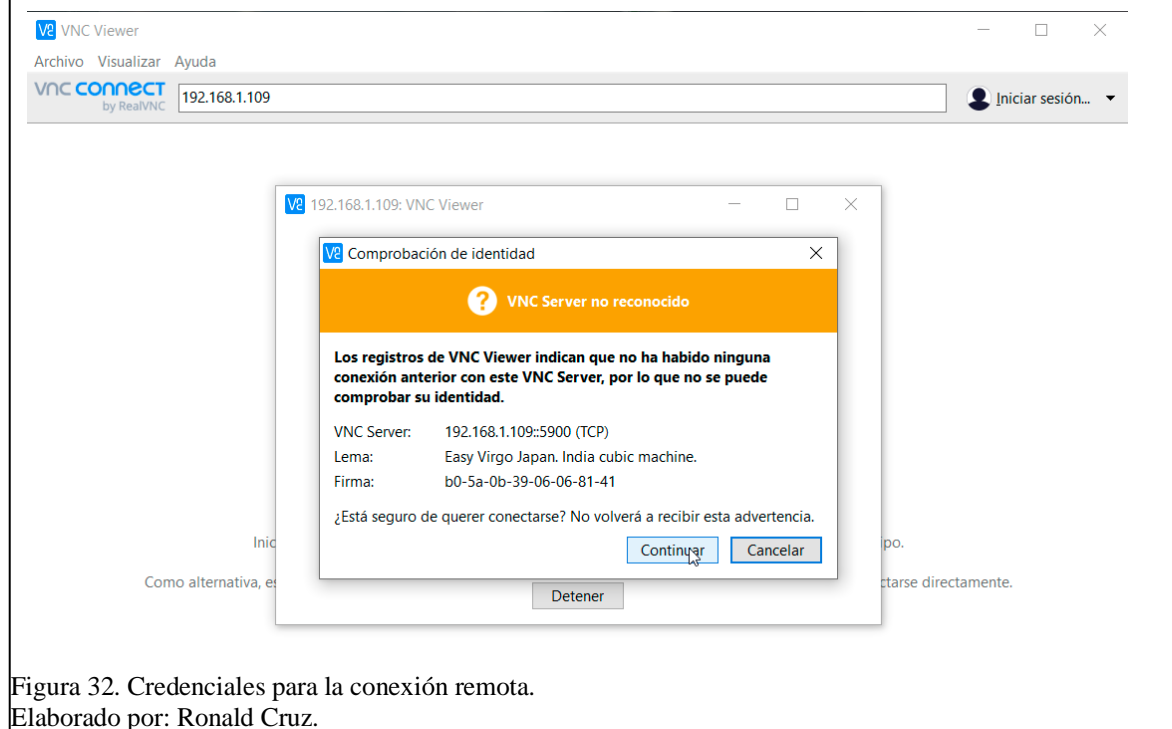


Figura 32. Credenciales para la conexión remota.
Elaborado por: Ronald Cruz.

Al finalizar, se podrá visualizar el escritorio del sistema operativo Raspberry Pi OS.



2.11. Habilitar Raspberry Pi Camera

Existen dos maneras para habilitar la cámara instalada en la tarjeta Raspberry Pi: la primera es desplegando la herramienta de configuración, en la pestaña de interfaces se muestra la opción para activar y desactivar la cámara:

Ventana configuración Raspberry Pi OS.

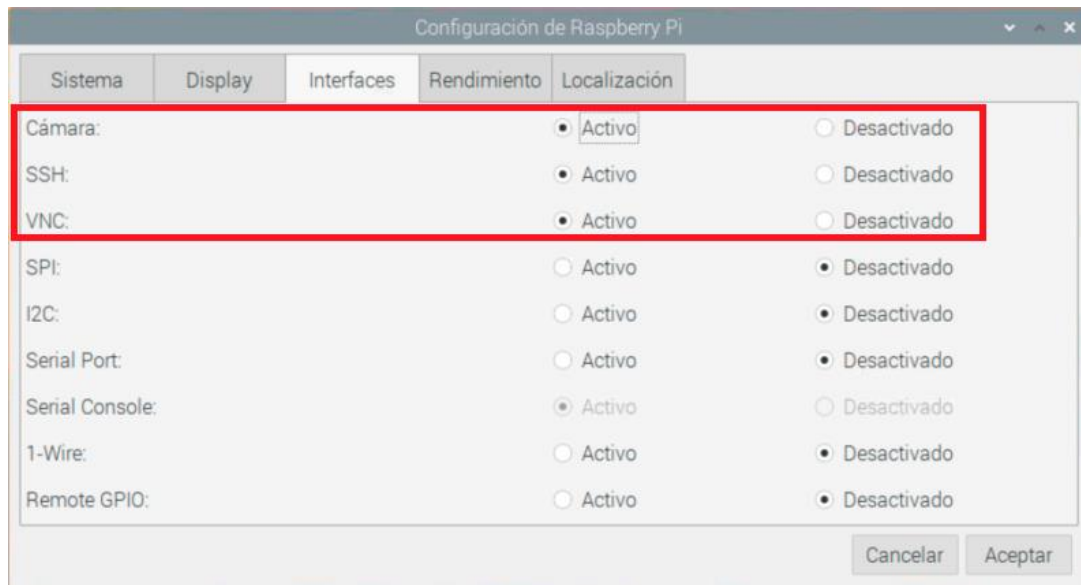


Figura 34. Ventana para habilitar la cámara en Raspberry Pi OS.
Elaborado por: Ronald Cruz.

El comando que se utiliza en la terminal para tomar fotos es `raspistill` y puede ser ejecutado desde la línea de comandos o creando un script, para probar que la cámara esta correctamente instalada se escribe dicho comando en la terminal del sistema, la imagen se guardará como `ImagenPruebaCameraPI.png`:

Prueba de funcionamiento de cámara.

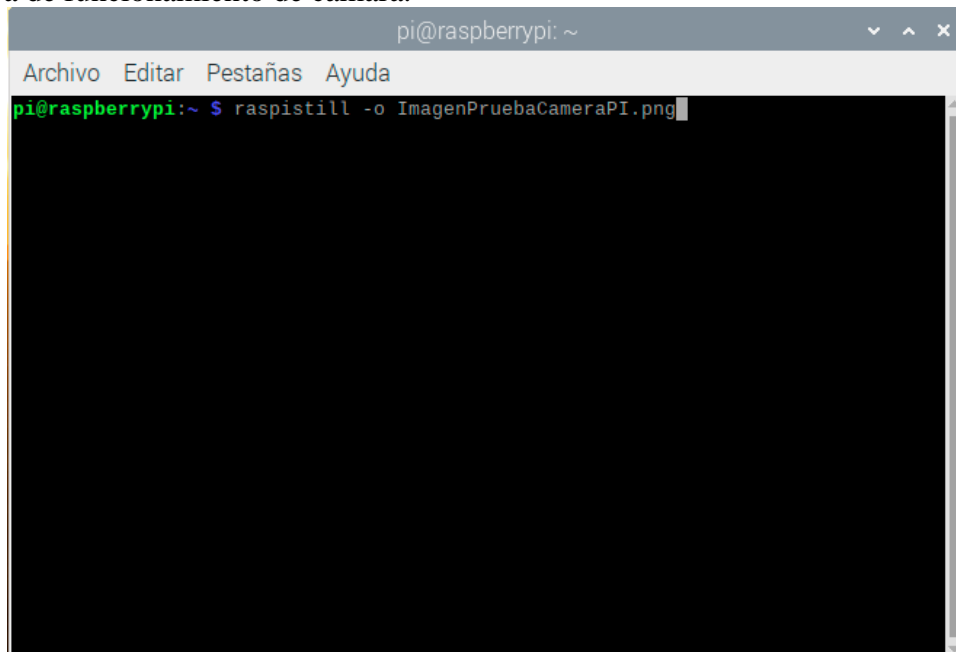


Figura 35. Prueba de funcionamiento de cámara.
Elaborado por: Ronald Cruz.

Una vez guardada la imagen se puede observar que ya captura imágenes, lo que queda es configurar y calibrar para que pueda capturar imágenes en buena resolución.

Imagen capturada por la cámara IR-CUT.



Figura 36. Captura de imagen por la cámara IR-CUT.
Elaborado por: Ronald Cruz.

La siguiente forma es mediante el lenguaje de programación Python con el IDE Thonny Python con los siguientes comandos, donde los comandos: `camera.start_preview()` realiza una previsualización de la captura que va a realizar, `sleep` sirve para que la cámara tenga el tiempo necesario para enfocar el objetivo a capturar, `camera.capture` es la ruta donde se va a guardar el objetivo capturado, `camera.stop_preview()` detiene la previsualización de la cámara.

Iniciar cámara mediante líneas de código en Python.

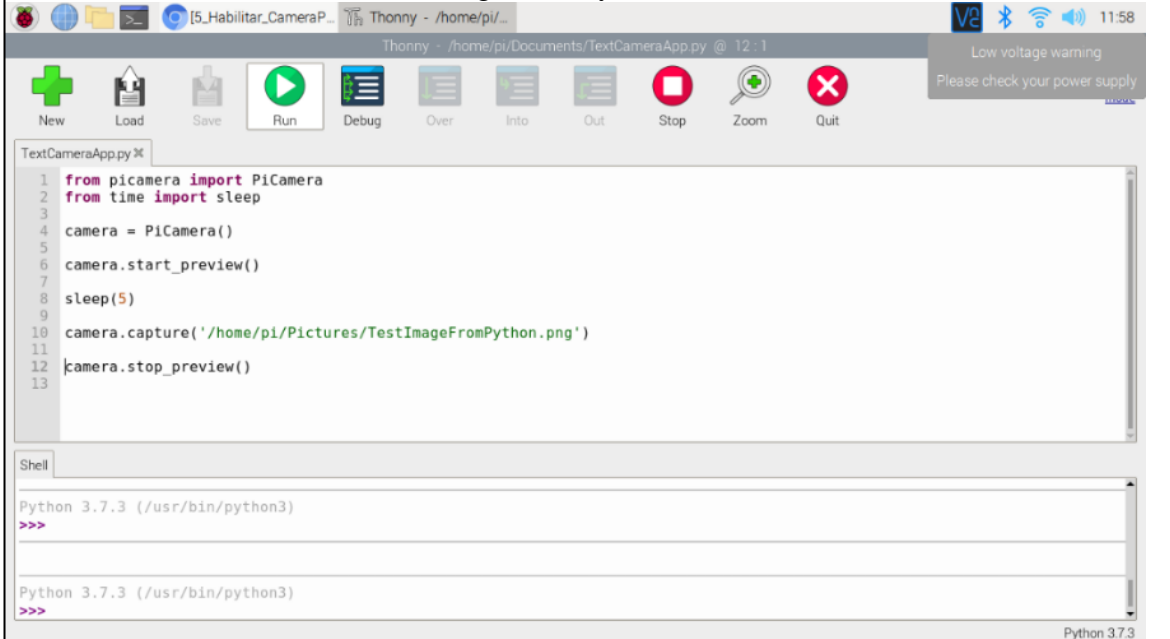


Figura 37. Comandos para iniciar la cámara mediante líneas de código en Python.
Elaborado por: Ronald Cruz.

2.12. Instalación librería OpenCV

Como primer paso se debe correr en el a consola de comandos: `sudo raspi-config`, aparecerá la pantalla de configuración de herramientas, seleccionar la opción número 6 y presionar enter.

Configuración de herramientas en Raspberry Pi OS.

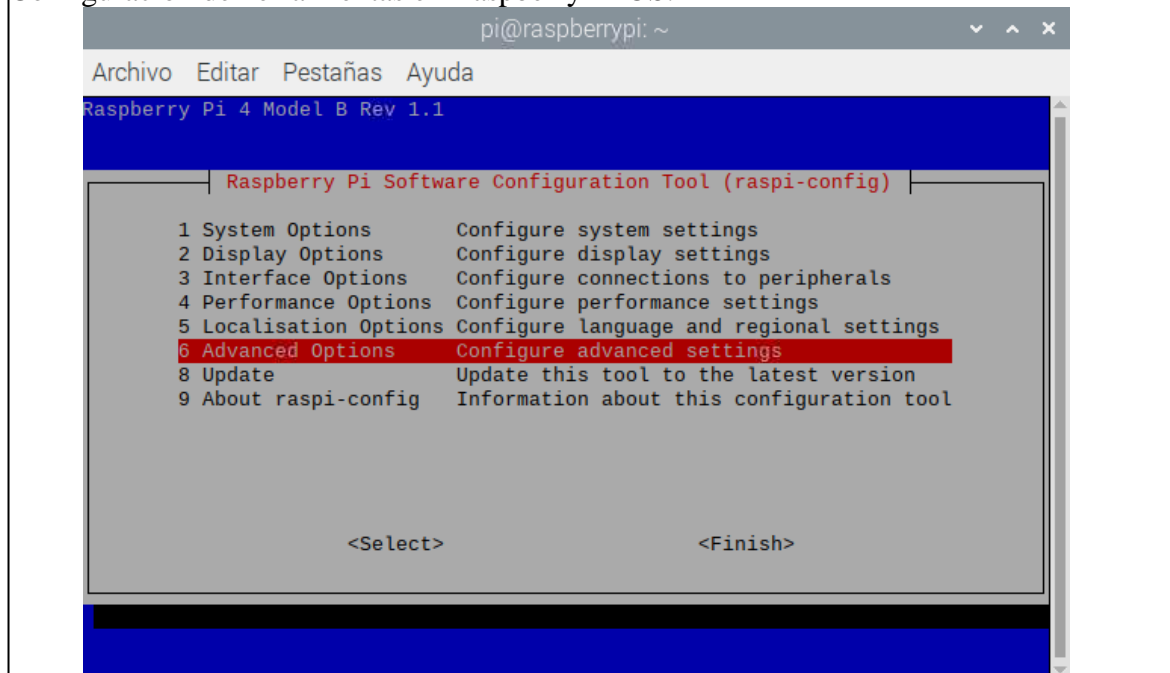


Figura 38. Ventana de configuración de herramientas de Raspberry Pi OS.
Elaborado por: Ronald Cruz.

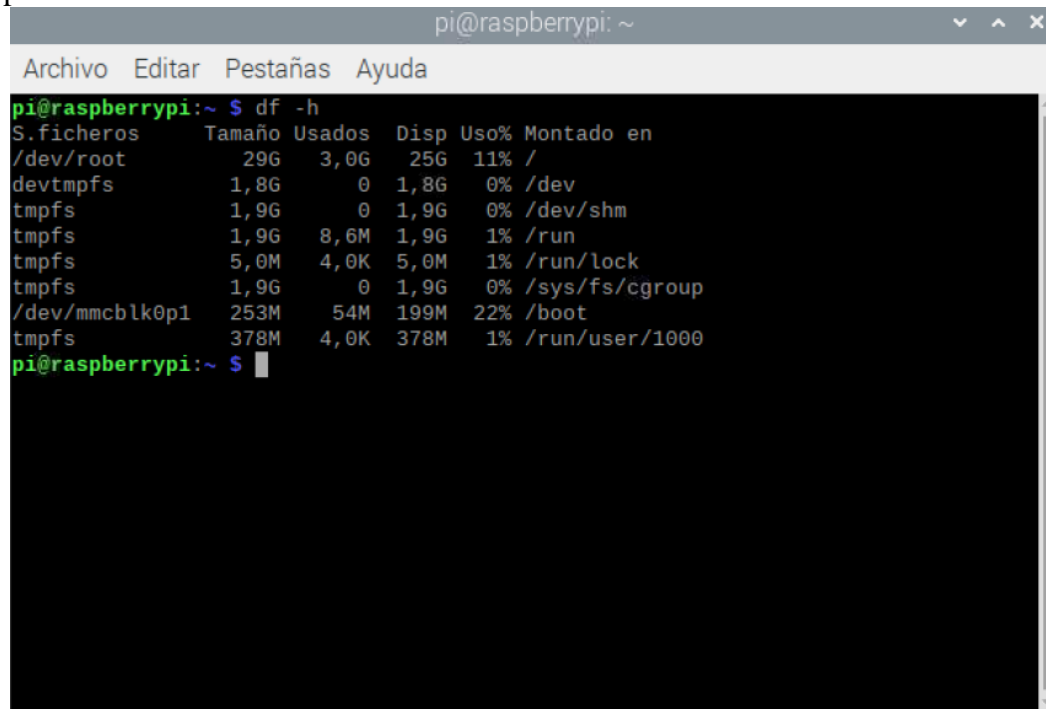
Seleccionar la primera opción: expandir sistema de archivos, permite expandir el sistema de archivos en la tarjeta Micro SD, para poder tener más espacio de almacenamiento e instalar la librería OpenCV. Una vez seleccionado y expandido los archivos, reiniciar el sistema.



Figura 39. Ventana para expansión de archivos del sistema operativo.
Elaborado por: Ronald Cruz.

Luego de haber reiniciado el sistema, por medio de la terminal escribir el siguiente comando: `df -h`, que sirve para verificar si la expansión de archivos se realizó correctamente.

Expansión correcta de archivos.

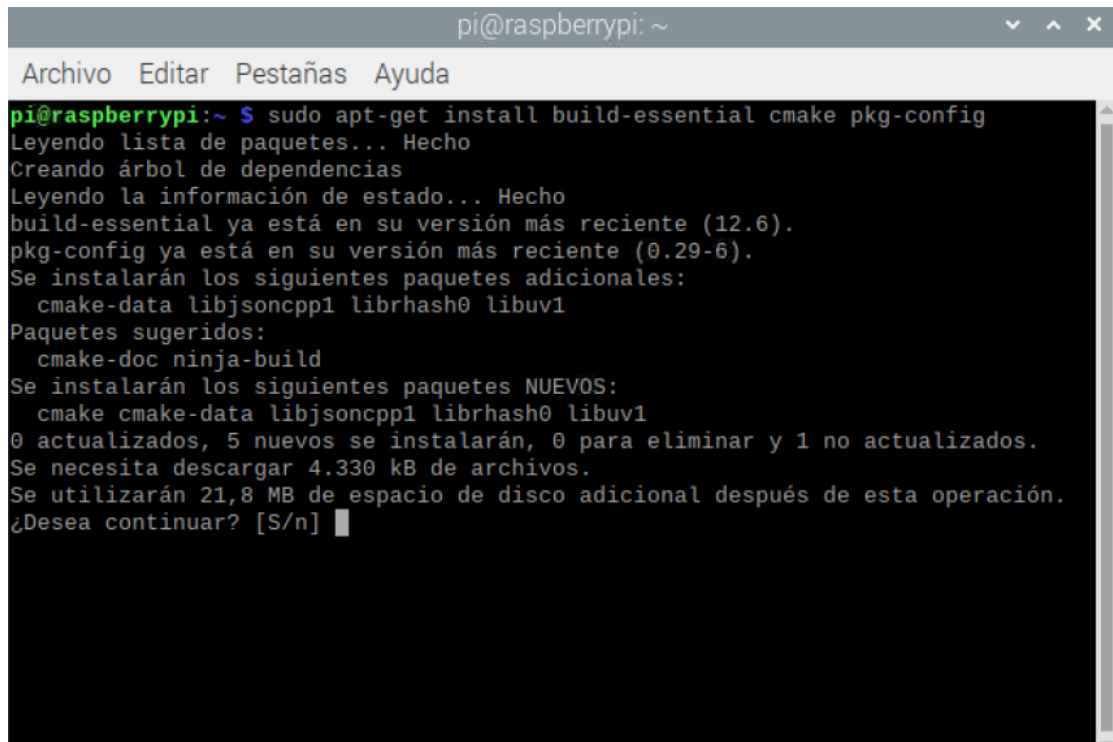


```
pi@raspberrypi:~ S df -h
S.ficheros      Tamaño Usados  Disp  Uso% Montado en
/dev/root        29G   3,0G   25G   11% /
devtmpfs         1,8G     0   1,8G    0% /dev
tmpfs            1,9G     0   1,9G    0% /dev/shm
tmpfs            1,9G   8,6M   1,9G    1% /run
tmpfs            5,0M   4,0K   5,0M    1% /run/lock
tmpfs            1,9G     0   1,9G    0% /sys/fs/cgroup
/dev/mmcblk0p1  253M    54M  199M   22% /boot
tmpfs            378M   4,0K   378M    1% /run/user/1000
pi@raspberrypi:~ S
```

Figura 40. Expansión de archivos en la Micro SD correcta.
Elaborado por: Ronald Cruz.

A continuación, se debe instalar algunas dependencias que necesita la librería para funcionar en la figura 41, se puede observar que se instalará la herramienta CMake que permite configurar el proceso de compilación de la librería OpenCV.

Instalación de la herramienta CMake.



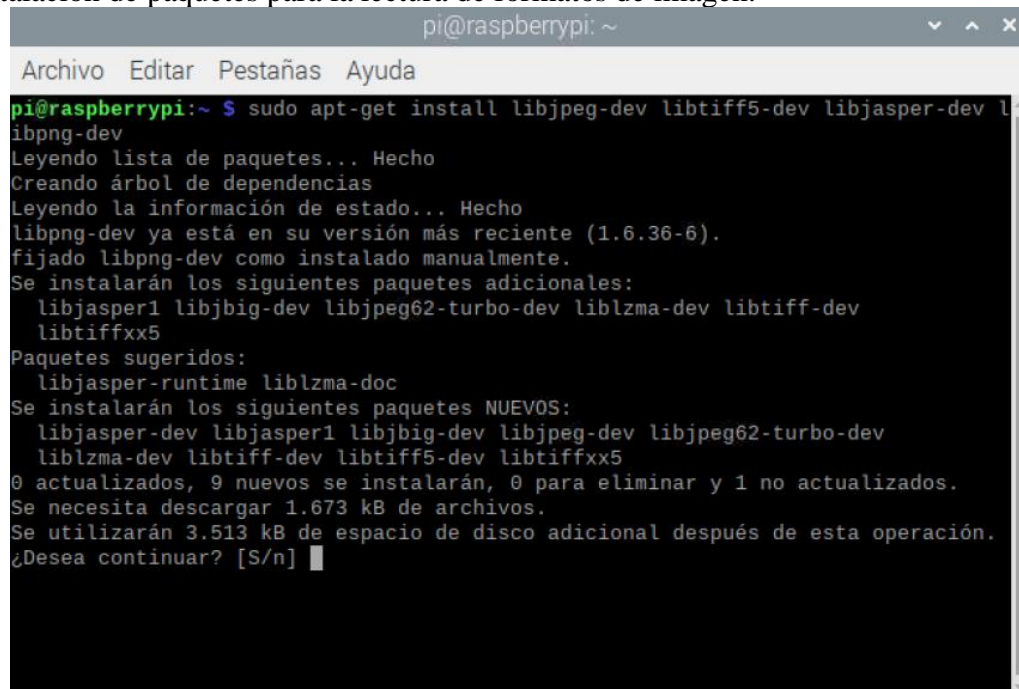
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ sudo apt-get install build-essential cmake pkg-config  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
build-essential ya está en su versión más reciente (12.6).  
pkg-config ya está en su versión más reciente (0.29-6).  
Se instalarán los siguientes paquetes adicionales:  
  cmake-data libjsoncpp1 librhash0 libuv1  
Paquetes sugeridos:  
  cmake-doc ninja-build  
Se instalarán los siguientes paquetes NUEVOS:  
  cmake cmake-data libjsoncpp1 librhash0 libuv1  
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 1 no actualizados.  
Se necesita descargar 4.330 kB de archivos.  
Se utilizarán 21,8 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Figura 41. Instalación de la herramienta CMake.

Elaborado por: Ronald Cruz.

Seguidamente se van a instalar paquetes que permitirán cargar varios formatos de imagen como, por ejemplo: PNG, JPG, JPEG, TIFF.

Instalación de paquetes para la lectura de formatos de imagen.



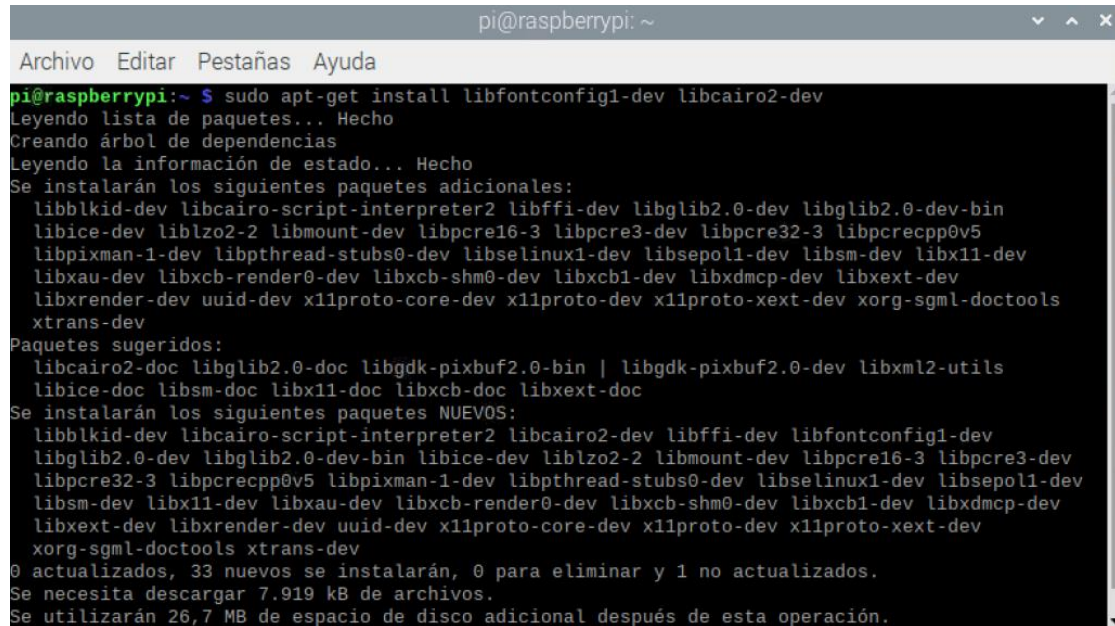
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
libpng-dev ya está en su versión más reciente (1.6.36-6).  
fijado libpng-dev como instalado manualmente.  
Se instalarán los siguientes paquetes adicionales:  
  libjasper1 libjbig-dev libjpeg62-turbo-dev liblzma-dev libtiff-dev  
  libtiffxx5  
Paquetes sugeridos:  
  libjasper-runtime liblzma-doc  
Se instalarán los siguientes paquetes NUEVOS:  
  libjasper-dev libjasper1 libjbig-dev libjpeg-dev libjpeg62-turbo-dev  
  liblzma-dev libtiff-dev libtiff5-dev libtiffxx5  
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 1 no actualizados.  
Se necesita descargar 1.673 kB de archivos.  
Se utilizarán 3.513 kB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Figura 42. Instalación de paquetes para lectura de varios formatos de imagen.

Elaborado por: Ronald Cruz.

La librería Open CV, cuenta con un módulo llamado Highgui, su funcionamiento es que muestra una pantalla básica con la imagen que la aplicación va a leer. Para ello, se necesita instalar tres dependencias:

Instalación del módulo Highgui.

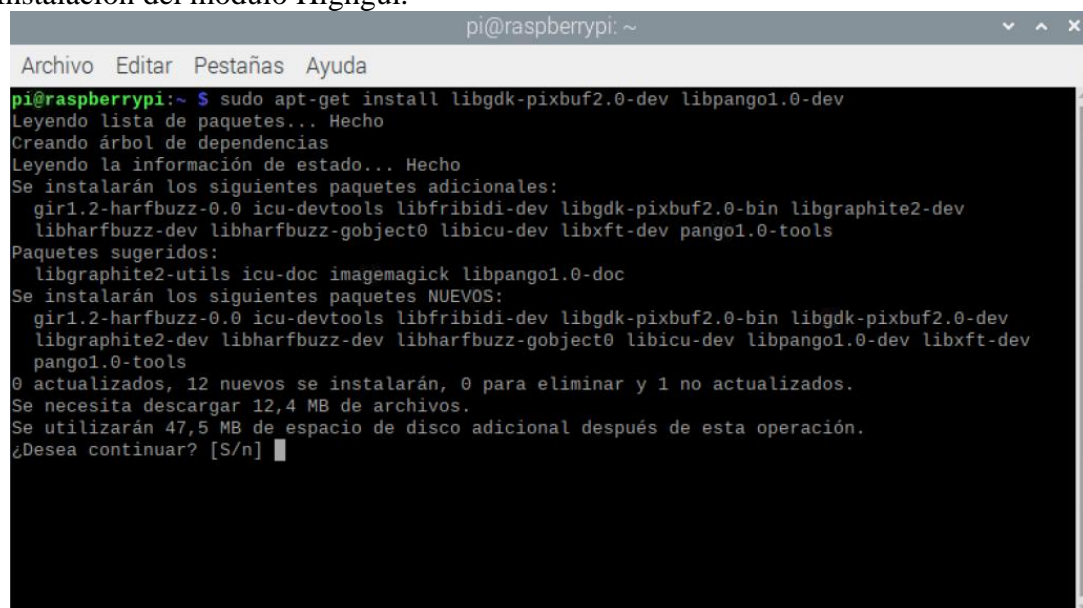


```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ sudo apt-get install libfontconfig1-dev libcairo2-dev  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  libblkid-dev libcairo-script-interpreter2 libffi-dev libglib2.0-dev libglib2.0-dev-bin  
  libice-dev liblzo2-2 libmount-dev libpcre16-3 libpcre3-dev libpcre32-3 libpcrecpp0v5  
  libpixman-1-dev libpthread-stubs0-dev libselinux1-dev libsepol1-dev libsm-dev libx11-dev  
  libxau-dev libxcb-render0-dev libxcb-shm0-dev libxcb1-dev libxdmcp-dev libxext-dev  
  libxrender-dev uuid-dev x11proto-core-dev x11proto-dev x11proto-xext-dev xorg-sgml-doctools  
  xtrans-dev  
Paquetes sugeridos:  
  libcairo2-doc libglib2.0-doc libgdk-pixbuf2.0-bin | libgdk-pixbuf2.0-dev libxml2-utils  
  libice-doc libsm-doc libx11-doc libxcb-doc libxext-doc  
Se instalarán los siguientes paquetes NUEVOS:  
  libblkid-dev libcairo-script-interpreter2 libcairo2-dev libffi-dev libfontconfig1-dev  
  libglib2.0-dev libglib2.0-dev-bin libice-dev liblzo2-2 libmount-dev libpcre16-3 libpcre3-dev  
  libpcre32-3 libpcrecpp0v5 libpixman-1-dev libpthread-stubs0-dev libselinux1-dev libsepol1-dev  
  libsm-dev libx11-dev libxau-dev libxcb-render0-dev libxcb-shm0-dev libxcb1-dev libxdmcp-dev  
  libxext-dev libxrender-dev uuid-dev x11proto-core-dev x11proto-dev x11proto-xext-dev  
  xorg-sgml-doctools xtrans-dev  
0 actualizados, 33 nuevos se instalarán, 0 para eliminar y 1 no actualizados.  
Se necesita descargar 7.919 kB de archivos.  
Se utilizarán 26,7 MB de espacio de disco adicional después de esta operación.
```

Figura 43. Instalación de las dependencias de OpenCV.

Elaborado por: Ronald Cruz.

Instalación del módulo Highgui.

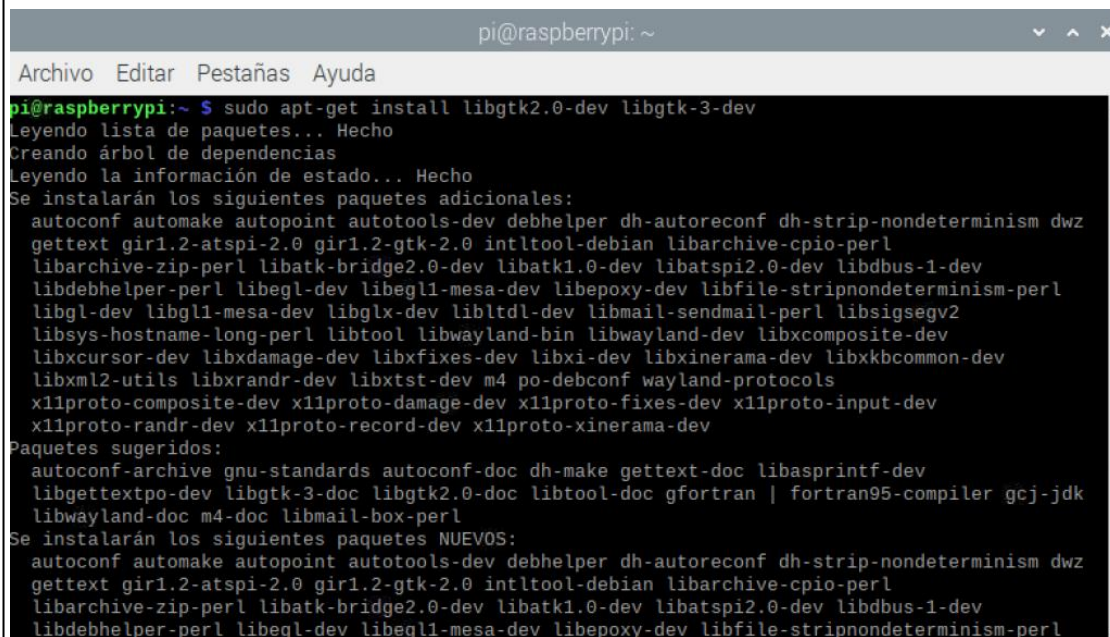


```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ sudo apt-get install libgdk-pixbuf2.0-dev libpango1.0-dev  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  gir1.2-harfbuzz-0.0 icu-devtools libfribidi-dev libgdk-pixbuf2.0-bin libgraphite2-dev  
  libharfbuzz-dev libharfbuzz-gobject0 libicu-dev libxft-dev pango1.0-tools  
Paquetes sugeridos:  
  libgraphite2-utils icu-doc imagemagick libpango1.0-doc  
Se instalarán los siguientes paquetes NUEVOS:  
  gir1.2-harfbuzz-0.0 icu-devtools libfribidi-dev libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-dev  
  libgraphite2-dev libharfbuzz-dev libharfbuzz-gobject0 libicu-dev libpango1.0-dev libxft-dev  
  pango1.0-tools  
0 actualizados, 12 nuevos se instalarán, 0 para eliminar y 1 no actualizados.  
Se necesita descargar 12,4 MB de archivos.  
Se utilizarán 47,5 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Figura 44. Instalación de las dependencias de OpenCV.

Elaborado por: Ronald Cruz.

Instalación del módulo Highgui.



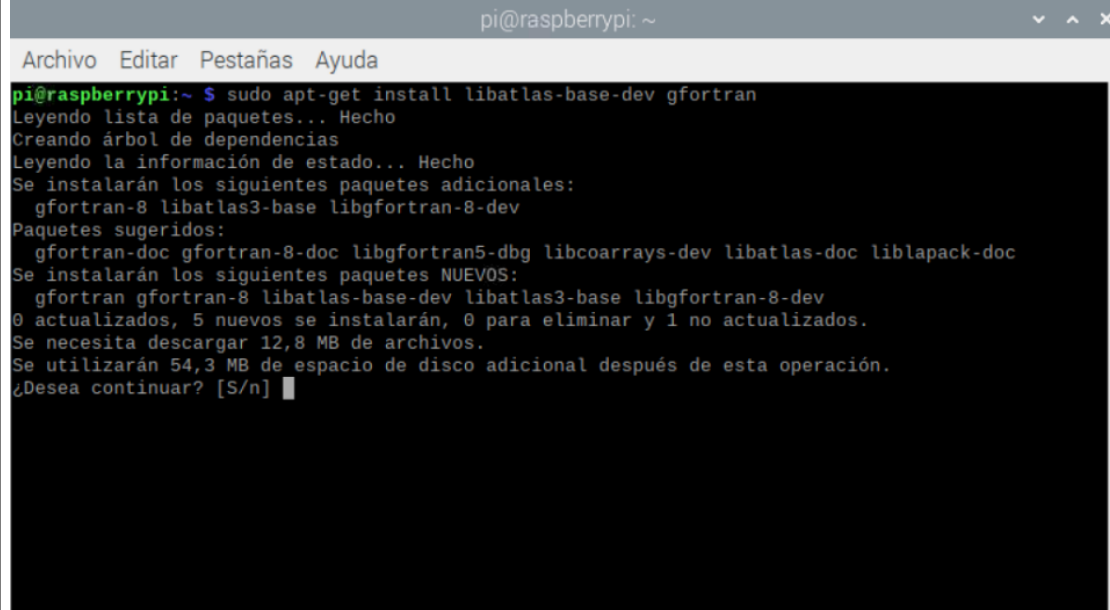
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get install libgtk2.0-dev libgtk-3-dev  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-strip-nondeterminism dwz  
gettext gir1.2-atspi-2.0 gir1.2-gtk-2.0 intltool-debian libarchive-cpio-perl  
libarchive-zip-perl libatk-bridge2.0-dev libatk1.0-dev libatspi2.0-dev libdbus-1-dev  
libdebhelper-perl libegl-dev libegl1-mesa-dev libepoxy-dev libfile-stripnondeterminism-perl  
libgl-dev libgl1-mesa-dev libglx-dev libltdl-dev libmail-sendmail-perl libsigsegv2  
libsys-hostname-long-perl libtool libwayland-bin libwayland-dev libxcomposite-dev  
libxcursor-dev libxdamage-dev libxfixes-dev libxi-dev libxinerama-dev libxkbcommon-dev  
libxml2-utils libxrandr-dev libxtst-dev m4 po-debconf wayland-protocols  
x11proto-composite-dev x11proto-damage-dev x11proto-fixes-dev x11proto-input-dev  
x11proto-randr-dev x11proto-record-dev x11proto-xinerama-dev  
Paquetes sugeridos:  
autoconf-archive gnu-standards autoconf-doc dh-make gettext-doc libasprintf-dev  
libgettextpo-dev libgtk-3-doc libgtk2.0-doc libtool-doc gfortran | fortran95-compiler gcj-jdk  
libwayland-doc m4-doc libmail-box-perl  
Se instalarán los siguientes paquetes NUEVOS:  
autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-strip-nondeterminism dwz  
gettext gir1.2-atspi-2.0 gir1.2-gtk-2.0 intltool-debian libarchive-cpio-perl  
libarchive-zip-perl libatk-bridge2.0-dev libatk1.0-dev libatspi2.0-dev libdbus-1-dev  
libdebhelper-perl libegl-dev libegl1-mesa-dev libepoxy-dev libfile-stripnondeterminism-perl
```

Figura 45. Instalación de las dependencias de la librería OpenCV.

Elaborado por: Ronald Cruz.

La librería OpenCV puede generar varias operaciones matriciales, las mismas que se las puede optimizar instalando la siguiente dependencia, es importante colocar esta herramienta de optimización ya que la tarjeta Raspberry Pi tiene sus recursos limitados.

Instalar herramientas de optimización.



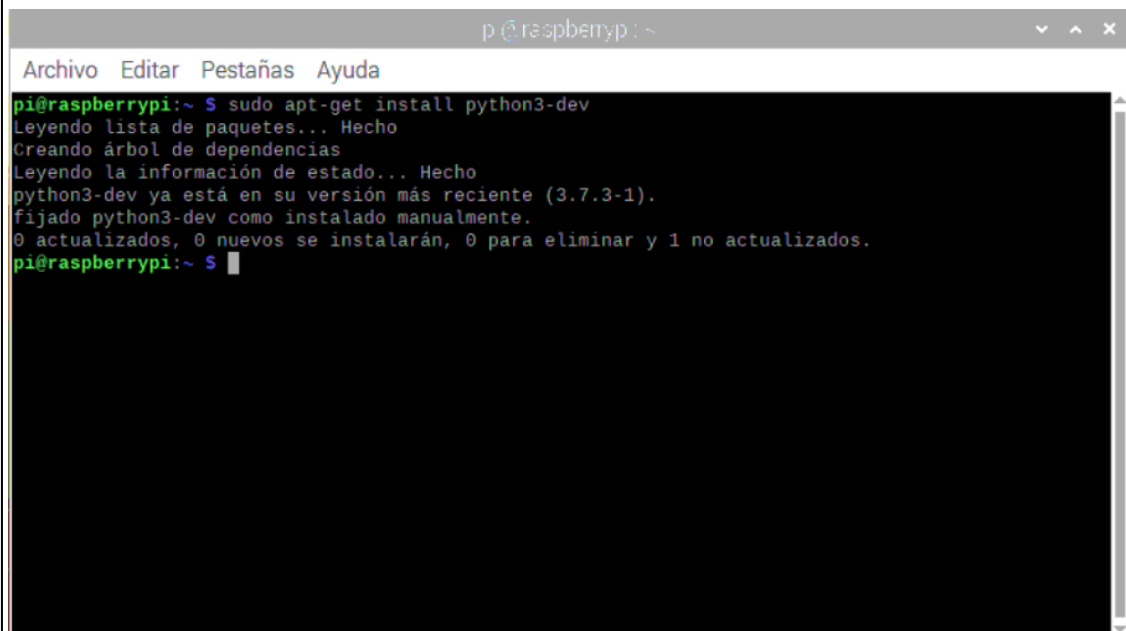
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ sudo apt-get install libatlas-base-dev gfortran  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
gfortran-8 libatlas3-base libgfortran-8-dev  
Paquetes sugeridos:  
gfortran-doc gfortran-8-doc libgfortran5-dbg libcoarrays-dev libatlas-doc liblapack-doc  
Se instalarán los siguientes paquetes NUEVOS:  
gfortran gfortran-8 libatlas-base-dev libatlas3-base libgfortran-8-dev  
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 1 no actualizados.  
Se necesita descargar 12,8 MB de archivos.  
Se utilizarán 54,3 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Figura 46. Instalación de herramienta de optimización de OpenCV.

Elaborado por: Ronald Cruz.

Por otro lado, se procede a la instalación de Python en su versión 3, es muy importante instalar las últimas versiones ya que las librerías trabajan de esta manera.

Instalación de Python versión 3.

A terminal window titled 'p @ raspberryp: ~' with a menu bar containing 'Archivo', 'Editar', 'Pestañas', and 'Ayuda'. The terminal output shows the command 'sudo apt-get install python3-dev' being executed. The output text is: 'Leyendo lista de paquetes... Hecho', 'Creando árbol de dependencias', 'Leyendo la información de estado... Hecho', 'python3-dev ya está en su versión más reciente (3.7.3-1).', 'fijado python3-dev como instalado manualmente.', '0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.', and finally 'pi@raspberrypi:~ \$' with a cursor.

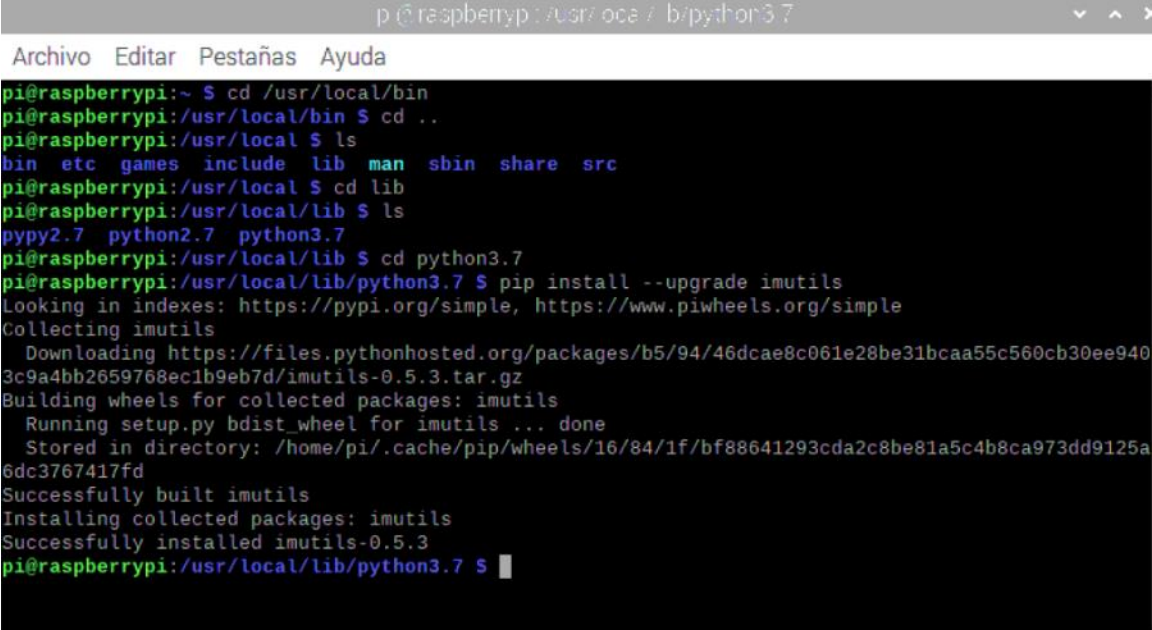
```
pi@raspberrypi:~ $ sudo apt-get install python3-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
python3-dev ya está en su versión más reciente (3.7.3-1).
fijado python3-dev como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
pi@raspberrypi:~ $
```

Figura 47. Instalación de Python versión 3.

Elaborado por: Ronald Cruz.

Luego de haber instalado Python versión 3, dirigirse a la carpeta `/usr/local/lib/python3.7.5` y proceder a la instalación de `imutils` que permite el procesamiento de imágenes básicas como son: rotación, cambio de tamaño, detección de bordes, etc.

Instalación de Imutils.

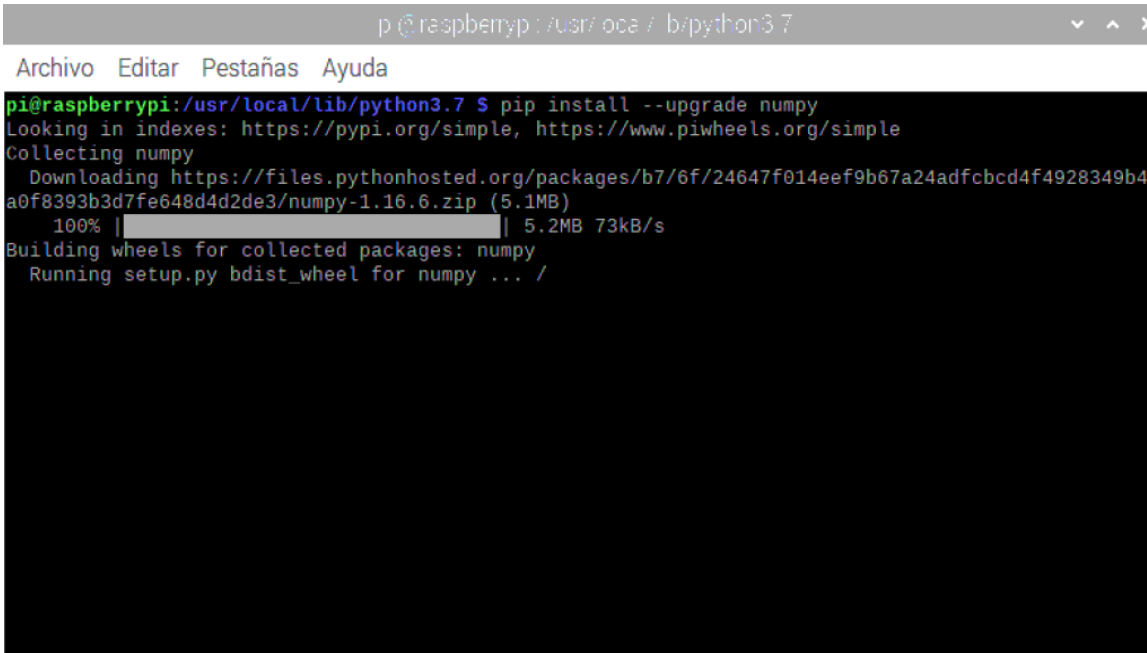


```
p @ raspberrypi: /usr/local/bin/python3.7
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ cd /usr/local/bin
pi@raspberrypi:/usr/local/bin $ cd ..
pi@raspberrypi:/usr/local $ ls
bin  etc  games  include  lib  man  sbin  share  src
pi@raspberrypi:/usr/local $ cd lib
pi@raspberrypi:/usr/local/lib $ ls
pypy2.7  python2.7  python3.7
pi@raspberrypi:/usr/local/lib $ cd python3.7
pi@raspberrypi:/usr/local/lib/python3.7 $ pip install --upgrade imutils
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting imutils
  Downloading https://files.pythonhosted.org/packages/b5/94/46dcae8c061e28be31bcaa55c560cb30ee9403c9a4bb2659768ec1b9eb7d/imutils-0.5.3.tar.gz
Building wheels for collected packages: imutils
  Running setup.py bdist_wheel for imutils ... done
  Stored in directory: /home/pi/.cache/pip/wheels/16/84/1f/bf88641293cda2c8be81a5c4b8ca973dd9125a6dc3767417fd
Successfully built imutils
Installing collected packages: imutils
Successfully installed imutils-0.5.3
pi@raspberrypi:/usr/local/lib/python3.7 $
```

Figura 48. Proceso de instalación de Imutils para Python.
Elaborado por: Ronald Cruz.

Numpy es un paquete de instalación fundamental para la computación con matrices en Python, proporciona un objeto de matriz N-Dimensional además que se puede usar como un contenedor multidimensional.

Instalación de Numpy.

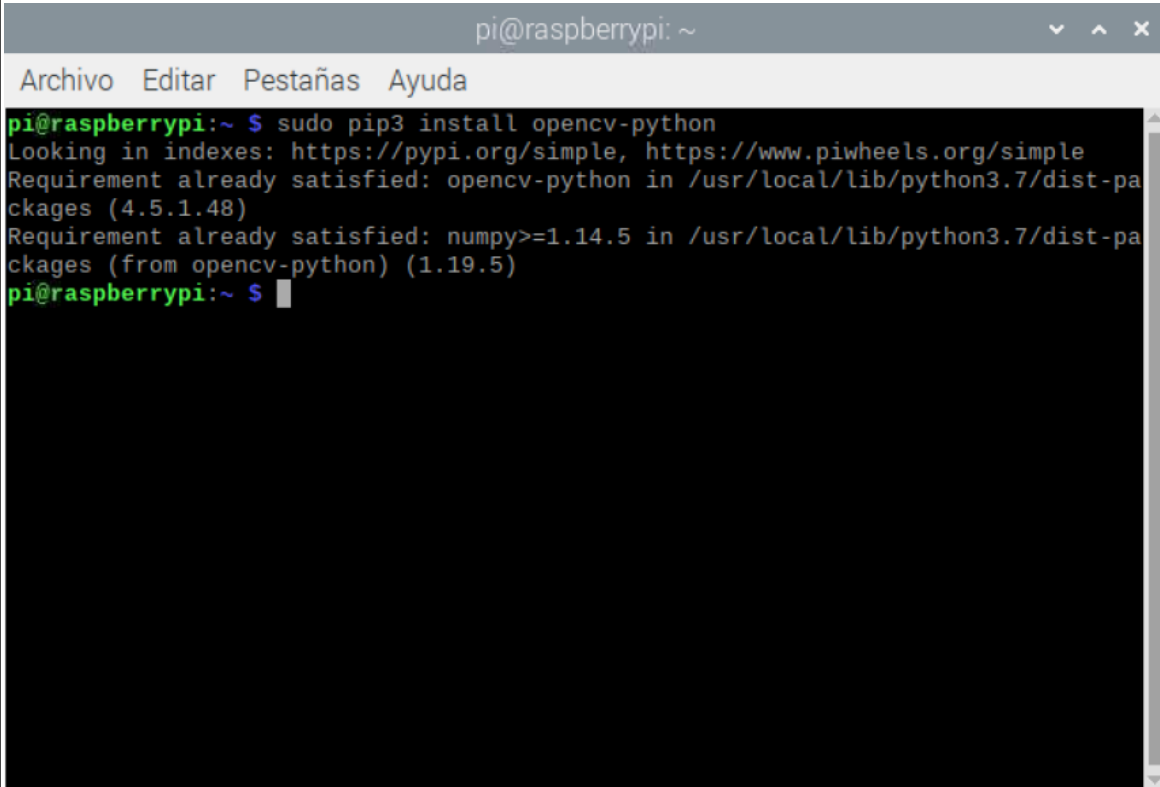


```
p @ raspberrypi: /usr/local/bin/python3.7
Archivo Editar Pestañas Ayuda
pi@raspberrypi:/usr/local/lib/python3.7 $ pip install --upgrade numpy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/b7/6f/24647f014eef9b67a24adfcabcd4f4928349b4a0f8393b3d7fe648d4d2de3/numpy-1.16.6.zip (5.1MB)
    100% |████████████████████████████████████████| 5.2MB 73kB/s
Building wheels for collected packages: numpy
  Running setup.py bdist_wheel for numpy ... /
```

Figura 49. Instalación del paquete Numpy.
Elaborado por: Ronald Cruz.

Como siguiente paso es la instalación de OpenCV.

Instalación de OpenCV.



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ sudo pip3 install opencv-python  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-pa  
ckages (4.5.1.48)  
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-pa  
ckages (from opencv-python) (1.19.5)  
pi@raspberrypi:~ $
```

Figura 50. Instalación de la librería OpenCV.
Elaborado por: Ronald Cruz.

Pillow es una biblioteca de imágenes de Python, su principal función es acceder de forma rápida a los datos que se encuentran almacenados y poder procesar las imágenes de manera más rápida.

Instalación de Pillow.

```
p@raspberrypi:/usr/local/lib/python3.7
Archivo Editar Pestañas Ayuda
pi@raspberrypi:/usr/local/lib/python3.7 $ pip install Pillow
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: Pillow in /usr/lib/python2.7/dist-packages (5.4.1)
pi@raspberrypi:/usr/local/lib/python3.7 $ pip install gTTS
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting gTTS
  Downloading https://files.pythonhosted.org/packages/31/27/6a8726c3c103e6d23fb4d7c1f1e85448a06d5ff85da0bd5a9f34e81ab9/gTTS-2.2.1.tar.gz
  Installing build dependencies ... done
Requirement already satisfied: six in /usr/lib/python2.7/dist-packages (from gTTS) (1.12.0)
Requirement already satisfied: click in /usr/lib/python2.7/dist-packages (from gTTS) (7.0)
Requirement already satisfied: requests in /usr/lib/python2.7/dist-packages (from gTTS) (2.21.0)
Building wheels for collected packages: gTTS
  Running setup.py bdist_wheel for gTTS ... done
  Stored in directory: /home/pi/.cache/pip/wheels/3d/a6/b0/36b5514bc0b42a22f8e87e92c2a53d1b9322cf7e94e01401ef
Successfully built gTTS
Installing collected packages: gTTS
  The script gtts-cli is installed in '/home/pi/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed gTTS-2.2.1
pi@raspberrypi:/usr/local/lib/python3.7 $
```

Figura 51. Instalación de Pillow en Python.

Elaborado por: Ronald Cruz.

Instalación de Tesseract, este contenedor es importante para el reconocimiento y lectura de imágenes.

Instalación de Tesseract.

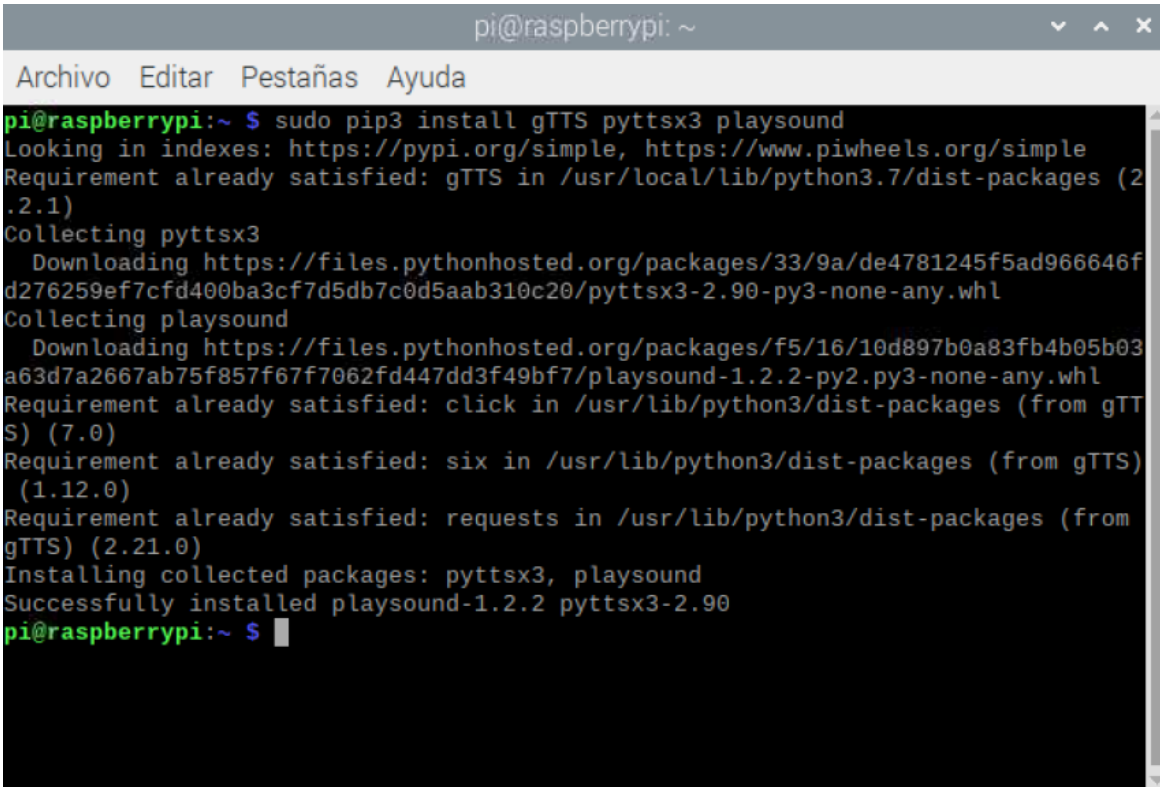
```
p@raspberrypi:/usr/local/lib/python3.7
Archivo Editar Pestañas Ayuda
pi@raspberrypi:/usr/local/lib/python3.7 $ pip install pytesseract
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pytesseract
  Downloading https://www.piwheels.org/simple/pytesseract/pytesseract-0.3.7-py2.py3-none-any.whl
Requirement already satisfied: Pillow in /usr/lib/python2.7/dist-packages (from pytesseract) (5.4.1)
Installing collected packages: pytesseract
  The script pytesseract is installed in '/home/pi/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pytesseract-0.3.7
pi@raspberrypi:/usr/local/lib/python3.7 $
```

Figura 52. Instalación del contenedor Tesseract.

Elaborado por: Ronald Cruz.

Como siguiente paso se debe instalar pytttsx3, es la biblioteca de conversión de texto a audio, es compatible con Python 2 y 3.

Instalación de Pytttsx3.



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ sudo pip3 install gTTS pytttsx3 playsound  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Requirement already satisfied: gTTS in /usr/local/lib/python3.7/dist-packages (2.2.1)  
Collecting pytttsx3  
  Downloading https://files.pythonhosted.org/packages/33/9a/de4781245f5ad966646fd276259ef7cfd400ba3cf7d5db7c0d5aab310c20/pytttsx3-2.90-py3-none-any.whl  
Collecting playsound  
  Downloading https://files.pythonhosted.org/packages/f5/16/10d897b0a83fb4b05b03a63d7a2667ab75f857f67f7062fd447dd3f49bf7/playsound-1.2.2-py2.py3-none-any.whl  
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from gTTS) (7.0)  
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from gTTS) (1.12.0)  
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from gTTS) (2.21.0)  
Installing collected packages: pytttsx3, playsound  
Successfully installed playsound-1.2.2 pytttsx3-2.90  
pi@raspberrypi:~ $
```

Figura 53. Instalación de la biblioteca Pytttsx3.
Elaborado por: Ronald Cruz.

Como siguiente paso se debe instalar Espeak para poder generar un archivo mp3 del texto convertido.

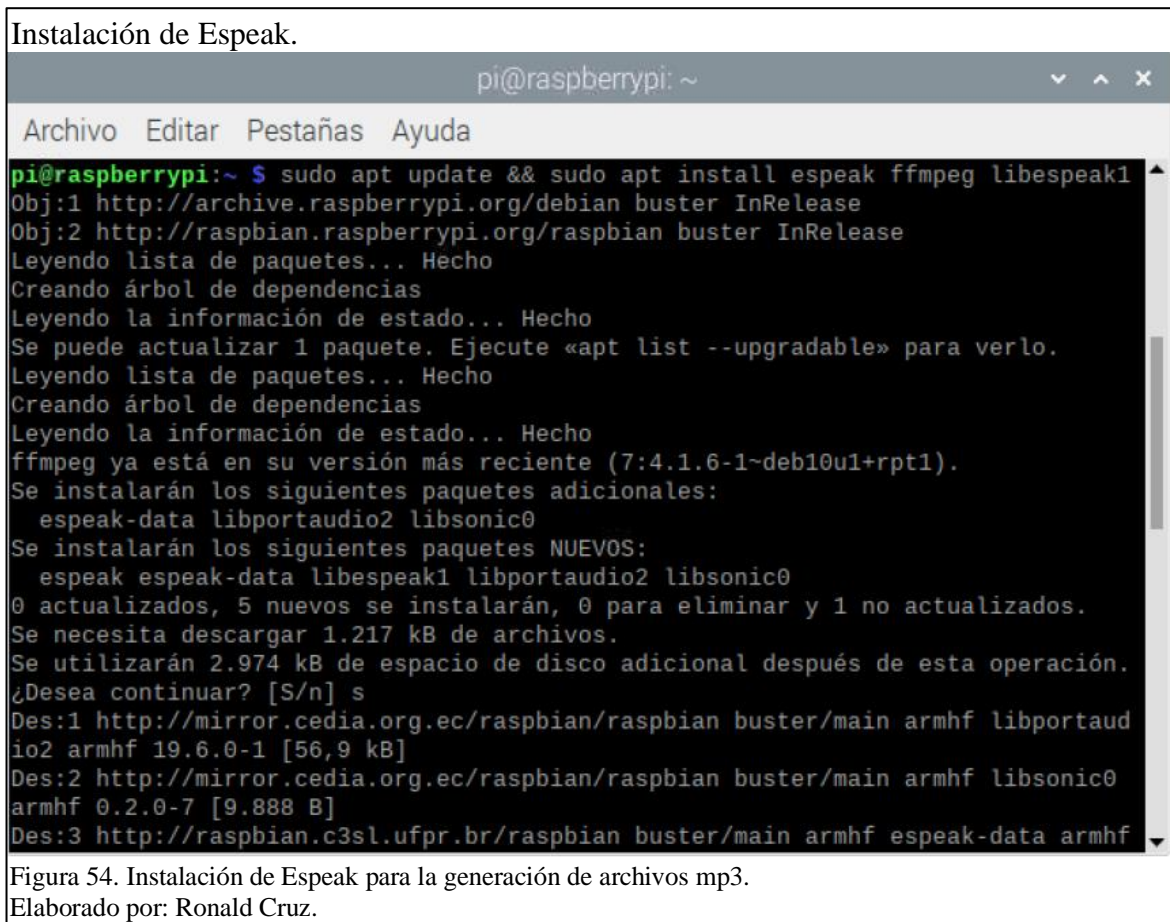


Figura 54. Instalación de Espeak para la generación de archivos mp3.
Elaborado por: Ronald Cruz.

2.13. Descripción de código

En el siguiente apartado, se describe el código utilizado para el desarrollo del proyecto.

El la figura 55, se muestra la parte de importación de las librerías para poder inicializar el módulo de cámara, a su vez se puede visualizar los objetos que permiten arrancar y generar una previsualización del objetivo a capturar. Finalmente se encuentra el objeto que permite capturar y guardar la imagen, en un directorio previamente creado.

Librerías, objetos usados para el funcionamiento de la cámara.

```
● ● ●  
  
#Importación de librería picamera.  
try:  
    import picamera  
    from picamera import *  
    import picamera.array  
except ImportError:  
    print("No tienes configurado PI CAMERA")  
  
#Inicializar objeto y arrancar previsualización de cámara.  
  
self.camera = PiCamera()  
self.camera.start_preview(fullscreen=False,window=(100,100,400,600))  
  
#Captura y guardado en un directorio de la imagen.  
  
self.camera.capture(self.rutaImagen)  
pixmap = QPixmap(self.rutaImagen)  
self.speakerApp.imgView.setScaledContents(1)  
self.speakerApp.imgView.setPixmap(pixmap.scaled(512, 384))
```

Figura 55. Librerías, objetos usados para el funcionamiento de la cámara.
Elaborado por: Ronald Cruz.

Valores que se debe generar para el funcionamiento de la librería OpenCV y tenga una labor correcta en el prototipo creado.

Valores por defecto usados por la librería OpenCV.

```
● ● ●  
  
#Valores por defecto que se necesita para la libreria OpenCV  
  
ap = argparse.ArgumentParser()  
ap.add_argument("-east", "--east", type=str, default="frozen_east_text_detection.pb",  
help="Ruta del archivo EAST para detectar")  
ap.add_argument("-c", "--min-confidence", type=float, default=0.5,  
help="Probabilidad mínima requerida para inspeccionar una región")  
ap.add_argument("-w", "--width", type=int, default=640,  
help="Redimensión del ancho de imagen (Múltiplos de 32)")  
ap.add_argument("-e", "--height", type=int, default=640,  
help="Redimensión del alto de imagen (Múltiplos de 32)")  
args = vars(ap.parse_args())
```

Figura 56. Valores por defecto utilizados para la librería OpenCV.
Elaborado por: Ronald Cruz.

Carga de imagen a la librería OpenCV al proyecto, en este apartado se realiza la carga de la imagen, que consiste en leer y cargar la ruta de la imagen, redimensiona la imagen con

variables previamente calculadas, además toma el tiempo de inicio y fin y realiza una impresión del tiempo calculado entre los tiempos tomados.

```
Carga de imagen a OpenCV.

#Carga de imagen en la librería OpenCV.

image = cv2.imread(self.rutaImagen)
orig = image.copy()
(H, W) = image.shape[:2]
(newW, newH) = (args["width"], args["height"])
rW = W / float(newW)
rH = H / float(newH)
image = cv2.resize(image, (newW, newH))
(H, W) = image.shape[:2]
layerNames= [
    "feature_fusion/Conv_7/Sigmoid",
    "feature_fusion/concat_3"
]
net = cv2.dnn.readNet("frozen_east_text_detection.pb")
blob = cv2.dnn.blobFromImage(image, 1.0, (W, H), (123.68, 116.78, 103.94), swapRB=True,
crop=False)
start = time.time()
net.setInput(blob)
(scores, geometry) = net.forward(layerNames)
end = time.time()
print("[INFO] Texto detectado en {:.6f} segundos".format(end - start))
```

Figura 57. Carga de la imagen a la librería OpenCV.
Elaborado por: Ronald Cruz.

Convertir y reproducir el texto detectado a audio, primero se debe generar un archivo de audio a partir del texto en español, después se inicializa la librería del reproductor, toma el archivo de audio que se encuentra guardado en el directorio antes asignado, reproduce el archivo que se encuentra en formato mp3 en idioma español, finalmente espera a que termine la reproducción del mismo y finaliza el proceso, adicional se muestra una sentencia para que cuando no encuentre texto o exista una hoja en blanco, pueda decir la frase “No existe texto”.

Conversión y reproducción de texto a audio.

```
# Convertir y reproducir el texto en audio.

text = pytesseract.image_to_string(orig)
try:
    gTTS(text, lang='es').save(self.rutaAudio)
    pygame.init()
    pygame.mixer.init()
    pygame.mixer.music.load(self.rutaAudio)
    pygame.mixer.music.play()
    pygame.event.wait()
except:
    self.engine.say("No existe texto")
    self.engine.runAndWait()
```

Figura 58. Conversión y reproducción de texto detectado a audio.
Elaborado por: Ronald Cruz.

2.14. Estructura del dispositivo

El dispositivo fue montado sobre una tabla de madera de 40x40 cm, donde reposará la hoja con el texto a capturar, adicional la cámara cuenta con un soporte de madera de 36 cm de alto y 20 cm hacia el centro de la tabla, contará con dos parlantes donde se escuchará la salida de audio, una lámpara adicional para situaciones con poca luz.

Estructura del prototipo.

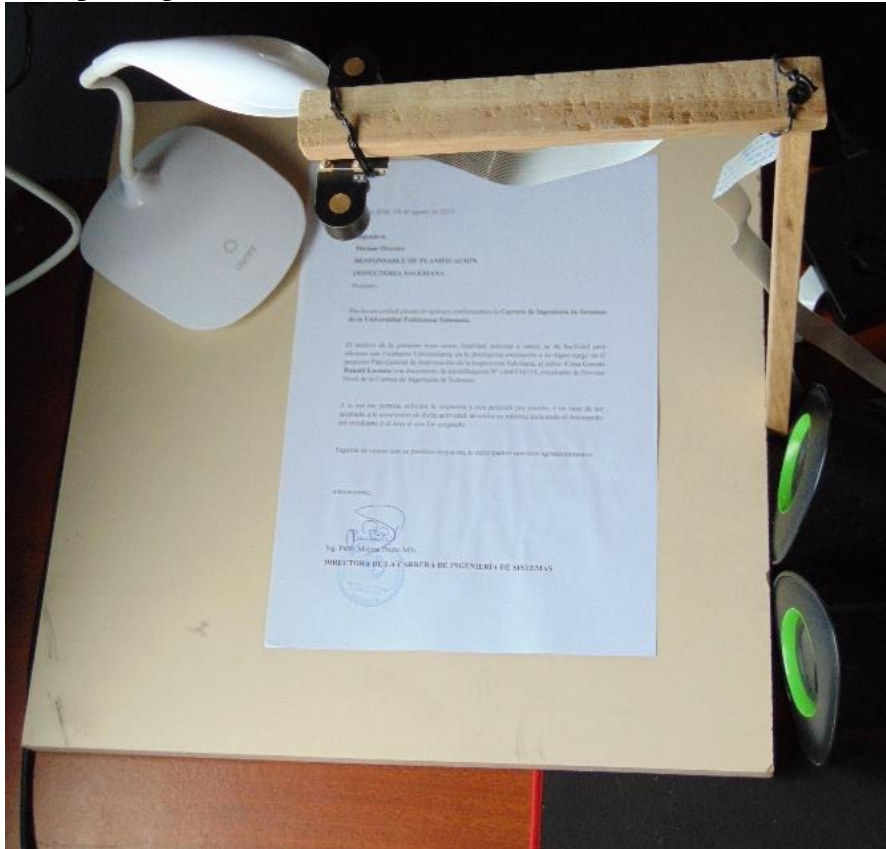


Figura 59. Estructura del prototipo.
Elaborado por: Ronald Cruz.

2.15. Algoritmo utilizado para el reconocimiento de texto

El algoritmo de reconocimiento de texto consiste en tratar la imagen capturada, se la segmenta y transforma a caracteres entendibles por cualquier computador, para este proyecto se obtuvo la ayuda de una librería de libre uso llamada PyTesseract y el uso de librerías de OpenCV, todas estas son compatibles con el lenguaje de programación Python, además tienen las características de ser de libre uso.

2.16. Proceso realizado por el algoritmo

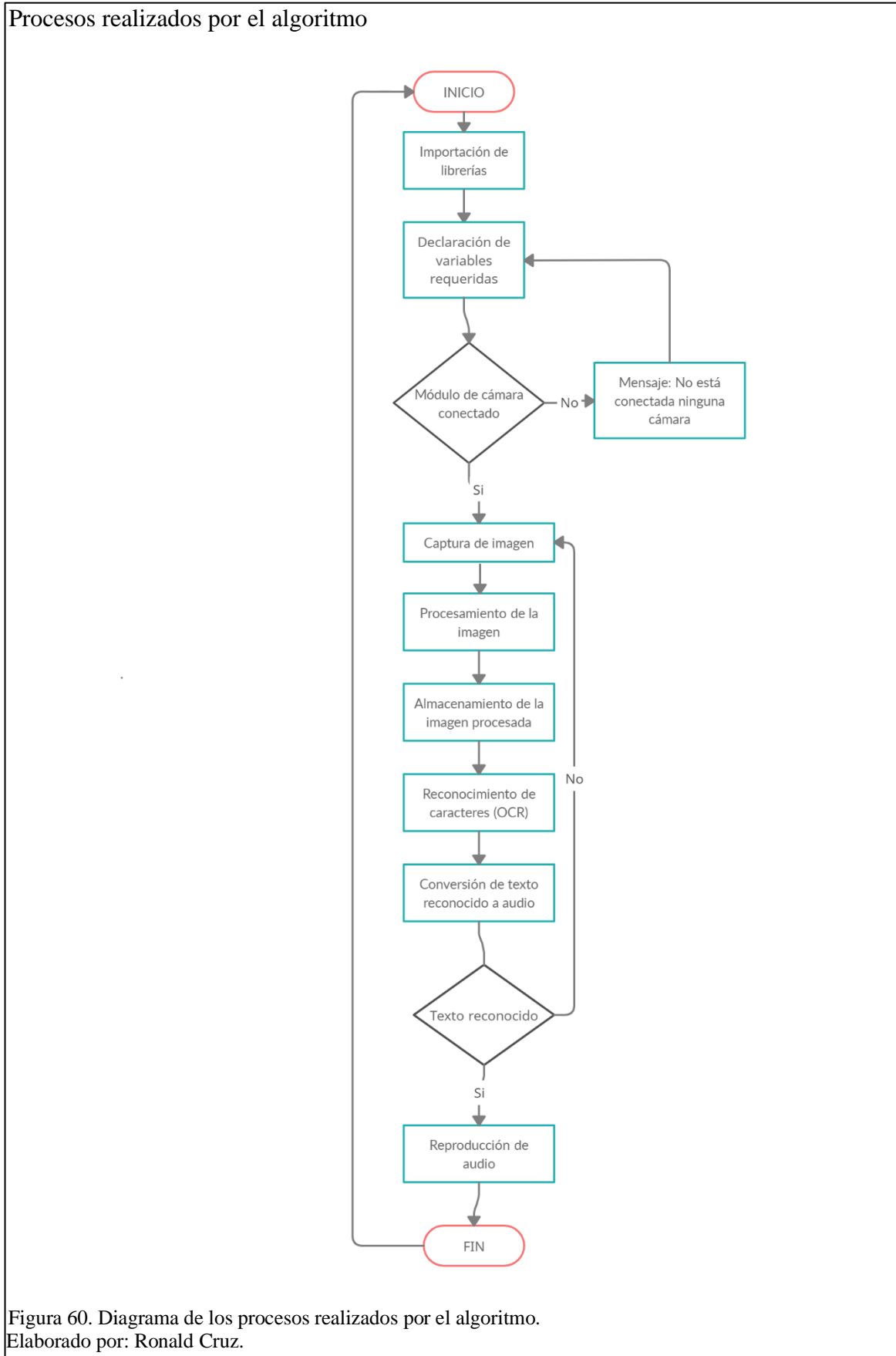


Figura 60. Diagrama de los procesos realizados por el algoritmo.
Elaborado por: Ronald Cruz.

La figura 60, hace referencia a los métodos que realiza el algoritmo, que son los siguientes procesos:

- El algoritmo inicia con la importación de las librerías necesarias para su funcionamiento y declara las variables necesarias para poder tratar la imagen a procesar.
- El algoritmo realiza la detección de la cámara, si existe conectada una cámara continua con el proceso, caso contrario regresa al paso anterior, como siguiente paso el algoritmo realiza la captura de la imagen y la guarda para procesar la misma que es la conversión a escala de grises para poder limpiar y reconocer el texto capturado.
- Luego realiza el reconocimiento óptico de caracteres (OCR), como siguiente paso es realizar la conversión del texto reconocido a un formato de audio por medio de la librería espeak.
- El siguiente paso del algoritmo es reproducir en forma de voz los caracteres reconocidos por el mismo, esta voz se reproducirá mediante un parlante.
- Finalmente, el algoritmo regresa al paso inicial hasta esperar un nuevo reconocimiento de texto y conversión a audio.

CAPITULO 3

PRUEBAS Y RESULTADOS

En este capítulo, se presenta las pruebas y resultados que verifican el cumplimiento de cada una de las funciones descritas. Para ellos se debe tomar en cuenta los siguientes aspectos:

- El documento que se va a procesar debe contener letras, números o símbolos impresos.
- Las palabras que contiene el papel debe ser texto en español.
- El prototipo debe estar en una adecuada iluminación para que exista una mejor lectura del documento.
- El prototipo debe contar con un administrador para poder iniciar el modo automático y la persona con discapacidad visual lo pueda usar.

3.1. Pruebas del prototipo

Luego de haber creado el prototipo, se procedió a realizar las pruebas respectivas que aseguren el funcionamiento del mismo y determine los resultados.

Tabla 6. Palabras reconocidas en porcentaje.

| Porcentaje de palabras reconocidas | | |
|---|-------------------|---------------------------------------|
| Pruebas | Porcentaje | Número de palabras reconocidas |
| Prueba 1 | 82,50% | 166 |
| Prueba 2 | 81,95% | 277 |
| Prueba 3 | 93,27% | 348 |

Nota: Porcentaje de reconocimiento de palabras.
Elaborado por: Ronald Cruz.

Como primera muestra del funcionamiento se procedió a leer el documento que se muestra en la figura 61:

Primera prueba generada por el prototipo.

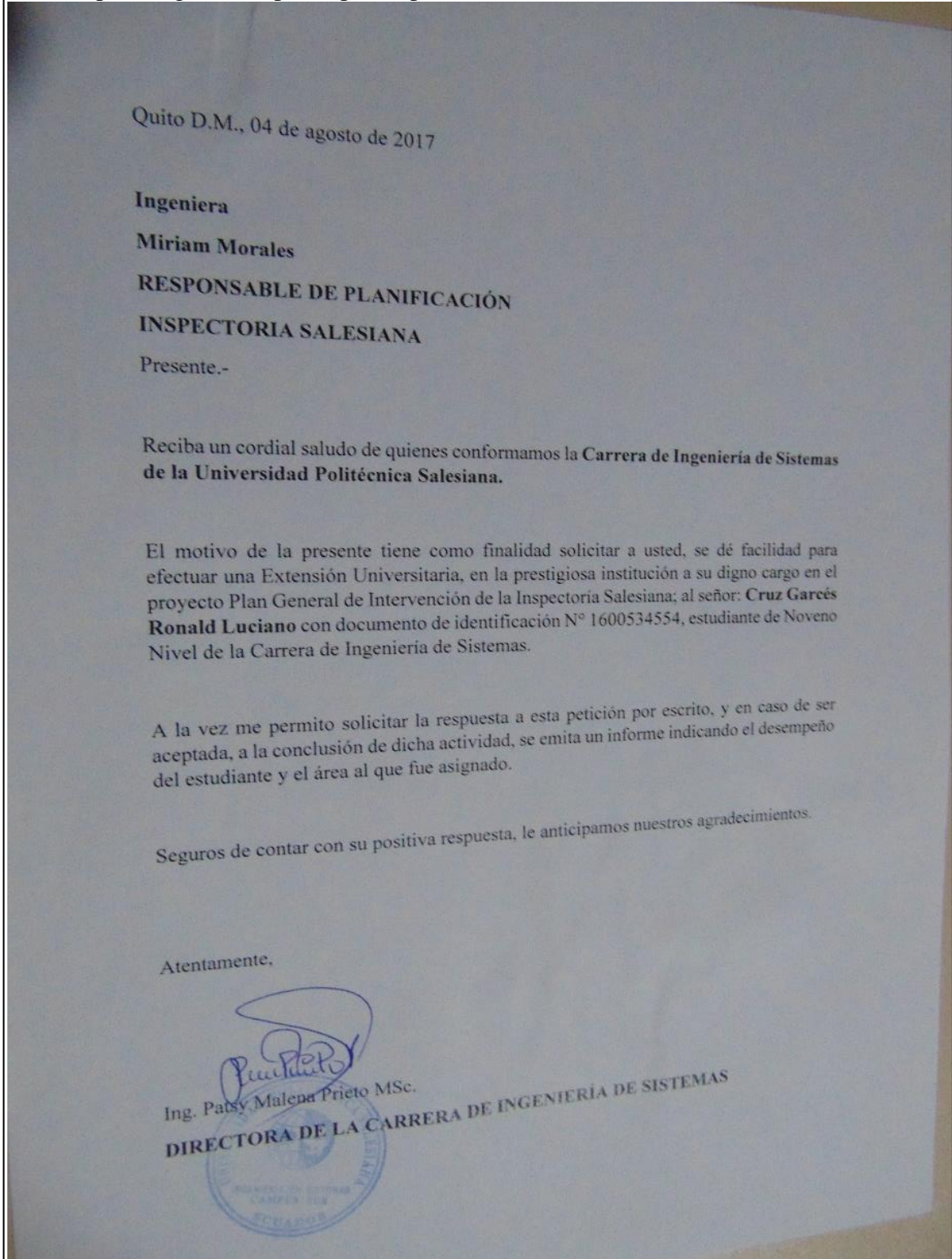


Figura 61. Primera prueba capturada por el prototipo.
Elaborado por: Ronald Cruz.

Texto convertido por el prototipo, con un tiempo transcurrido de 10.22 segundos, el tiempo es calculado en la aplicación, en el cual se toma el tiempo inicial desde la captura de

imagen y el tiempo final que es hasta donde termina el proceso de captura y conversión a texto. Entonces se toma estos dos valores y se resta, es decir TIEMPO INICIAL – TIEMPO FINAL, dando como resultado el TIEMPO TRANSCURRIDO.

```
Texto escaneado y convertido por el prototipo.
Shell
Quito D.M., 04 de agosto de 2017

Ingeniera

Miriam Morales

RESPONSABLE DE PLANIFICACION
INSPECTORIA SALESIANA

Presente.-

Reciba un cordial saludo de quienes conformamos la Carrera de Ingenieria de Sistemas
de la Universidad Politécnica Salesiana.

El motivo de la presente tiene como finalidad solicitar a usted, se dé facilidad para
efectuar una Extension Universitaria, en la prestigiosa institucion a su digno cargo en el
proyecto Plan General de Intervención de la Inspectoria Salesiana; al sefior: Cruz Garcés
Ronald Luciano con documento de identificacion N° 1600534554, estudiante de Noveno
Nivel de la Carrera de Ingenieria de Sistemas.

A la vez me permito solicitar la respuesta a esta peticién por escrito, y en caso de ser
aceptada, a la conclusién de dicha actividad, se emita un informe indicando el desempeio
del estudiante y el area al que fue asignado.
```

Figura 62. Texto convertido por el prototipo.
Elaborado por: Ronald Cruz.

Como final del proceso se escuchará la voz por medio de los parlantes conectados a la tarjeta Raspberry Pi, el texto que la aplicación reconoció, con un porcentaje de 82.5% de lectura del documento, este porcentaje es posible calcularlo ya que en el programa se implementó un contador de palabras y gracias a esto se pudo realizar una regla de 3 que permite calcular el porcentaje real de palabras reconocidas. El documento impreso tiene un total de 160 palabras, la regla de 3 se puede ver en la siguiente fórmula:

$$x = \frac{C \cdot B}{P}$$

Donde:

C es el número de palabras calculadas por el contador del programa, B es el 100%, P es el total de palabras que tiene el documento impreso, la fórmula quedaría reemplazada de la siguiente manera:

$$x = \frac{132.100}{160} = 82.5\%$$

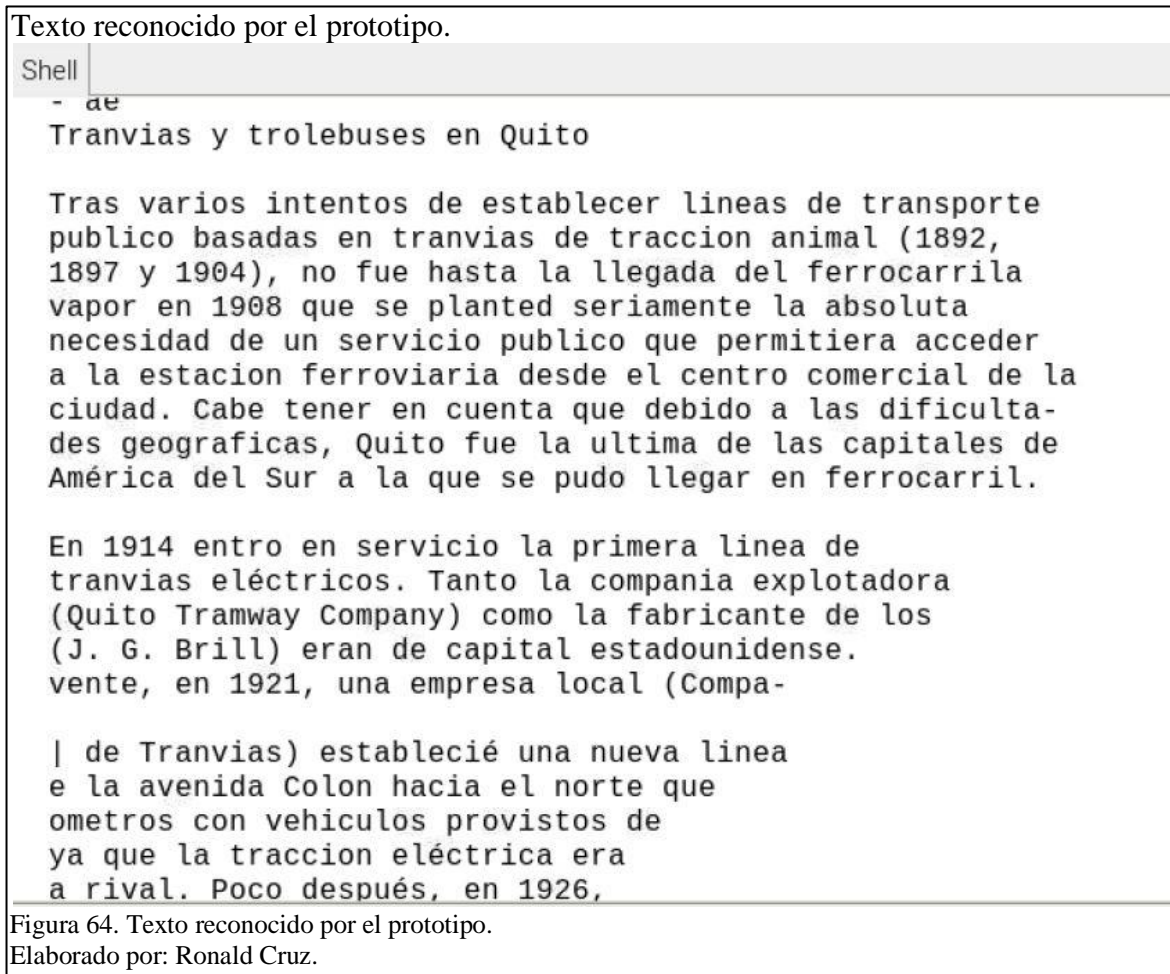
Segunda prueba realizada, se tomó un artículo de revista en el cual se pudo constatar que tuvo la capacidad de leer el 81.95% del texto impreso de dicho artículo, el porcentaje se realizó mediante la fórmula que quedaría de la siguiente manera:

$$x = \frac{227.100}{277} = 81.95\%$$



Figura 63. Segunda prueba realizada tomada de un artículo de revista.
Elaborado por: Ronald Cruz.

Como resultado se tiene el texto convertido por el prototipo con un tiempo transcurrido de 11.26 segundos:



la siguiente prueba, se tomó un manual de usuario en el cual se puede observar que contiene texto de tamaño pequeño de un rango de 0.6 a 0.8 cm, en la prueba realizada se puede verificar que reconoce dicho texto. Como se muestra en la figura 62.

Reconocimiento de letras tamaño 8.

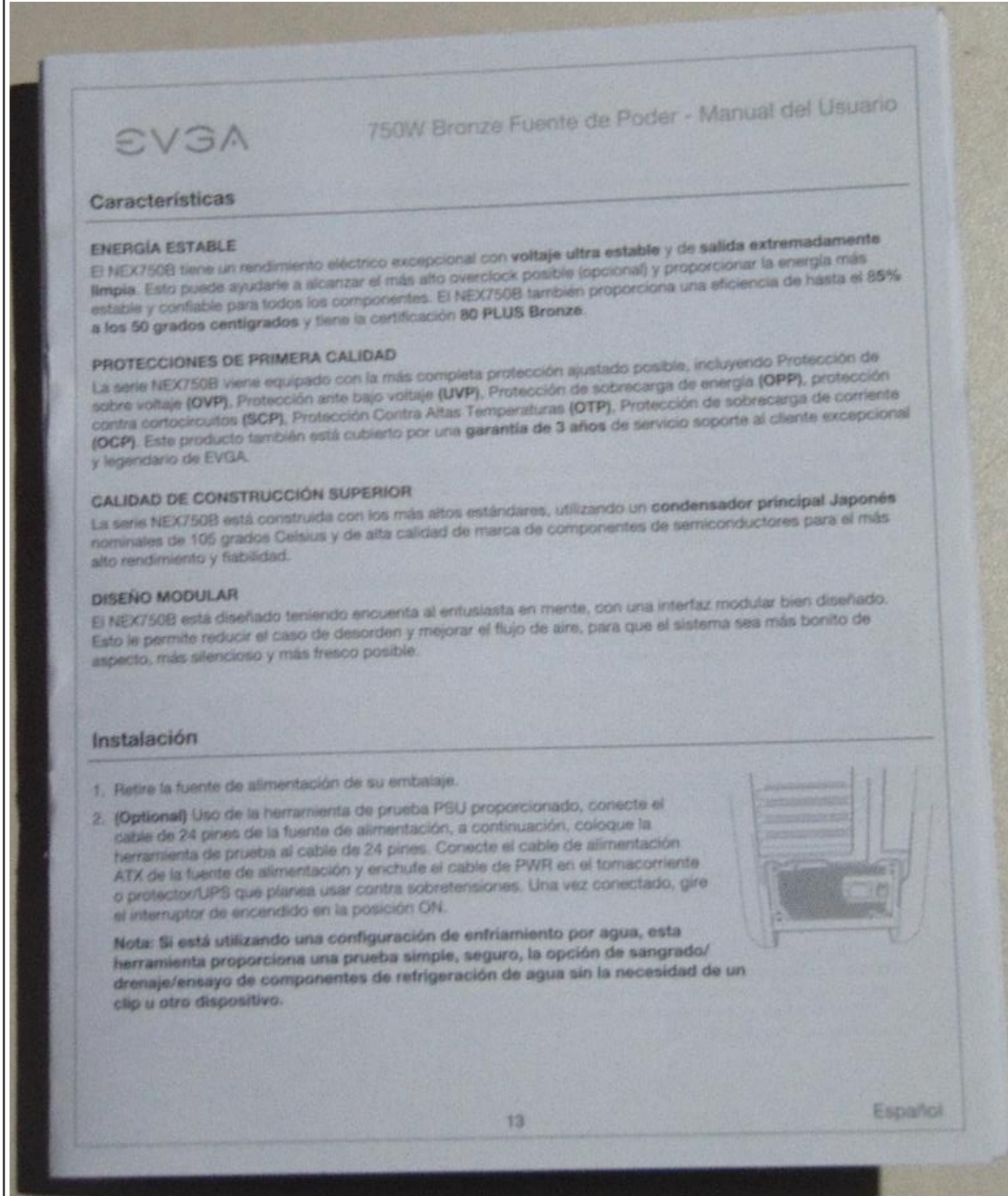


Figura 65. Reconocimiento de letras de tamaño 8.
Elaborado por: Ronald Cruz.

La página de dicho manual contiene 348 palabras de las cuales fueron reconocidas 327, aplicando la fórmula de la regla de 3 da como resultado 93.27%.

$$x = \frac{327.100}{348} = 93.27\%$$

Tercera prueba con letra tamaño de 0.6-0.8 cm.

```
Shell
IA mm4>mr m q q
ENERG to eléctrico excepcional con voltaje ultra estable y de salida extremadamente
508 tiene UN rendimien! . v .
mzvau alcanzar el más lo overclock posible (opcional) y proporcionar la energía más
o ucmm w<cn_°_m a , ]
ra todos los componentes. El NEX7508 también proporciona una eficiencia de hasta el 85%
grados Y tiene la certificación 80 PLUS Bronze.
2>0._.moo_ozmw DE PRIMERA CALIDAD
La serie NEX750B viene equipado con la más completa protección ajustado posible, incluyendo Fr
ot
...oa$<0=w...w ...o<34 protección ante bajo
voltaje (UVP), Protección de sobrecarga de aneraia (OPP), prott cción
contra cortocircuitos (SCP), Protección Contra Altas Temperaturas (OTP), Protección de sobreca
rga de comiente
(OCP). Este producto también está cublerto por una garantía de 3 años de servicio soporte al c
liente excepcional
```

Figura 66. Tercera prueba con letra tamaño 8.
Elaborado por: Ronald Cruz.

3.2. Pruebas con poca luz

En este apartado se realizará pruebas de reconocimiento de texto con tres intensidades de luz para verificar la eficacia del prototipo, la lampara utilizada tiene un voltaje de 5v a 2 w de potencia, contiene 20 leds.

Tabla 7. Pruebas con intensidad de luz

| Pruebas con luz artificial | | |
|----------------------------|----------------------|---------------------------------|
| Intensidad | Palabras reconocidas | Tiempo transcurrido en segundos |
| Luz baja | 270 | 11,79 |
| Luz media | 300 | 11,23 |
| Luz alta | 327 | 11,31 |

Nota: Pruebas con luz artificial.
Elaborado por: Ronald Cruz.

Se realiza la prueba con la intensidad de luz baja para poder medir la efectividad de la cámara que fue utilizada para la creación del prototipo, dando como resultado el reconocimiento de 270 palabras según el contador implementado en el prototipo en un tiempo de 11,79 segundos.

Prueba con de intensidad baja.



Figura 67. Prueba con luz de intensidad baja.
Elaborado por: Ronald Cruz.

Como se muestra la figura 68, la segunda prueba que se realizó con una luz de intensidad media en el cual tuvo una respuesta de 300 palabras reconocidas en un tiempo de 11,23 segundos.

Luz de intensidad media.

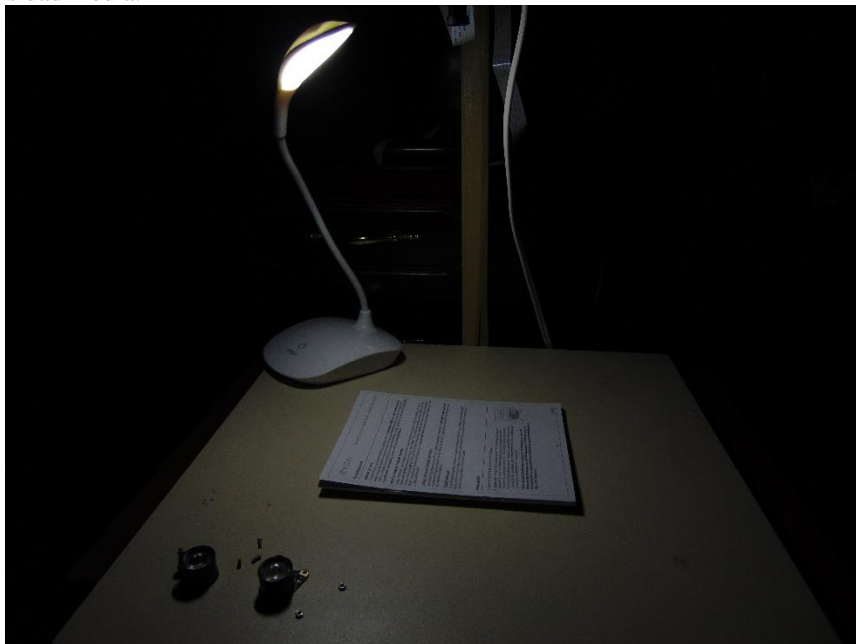


Figura 68. Segunda prueba con luz de intensidad media.
Elaborado por: Ronald Cruz.

Tercera prueba, se realizó con luz de intensidad alta, en el cual tuvo una respuesta de 327 palabras reconocidas en un tiempo de 11,31 segundos, se puede observar que mientras exista mejor iluminación el prototipo tendrá mejores resultados para reconocer el documento impreso.

Luz de intensidad fuerte.



Figura 69. Tercera prueba con luz de intensidad fuerte.
Elaborado por: Ronald Cruz.

3.3. Pruebas de distancia entre cámara y texto a leer

La primera prueba como se puede observar en la figura 70. Se realizó a 20 cm de distancia entre la cámara y el documento impreso, dando como resultado no poder enfocar todo el documento y no poder leer todo el texto que aparece en el papel.

Cámara a 20 cm de distancia.

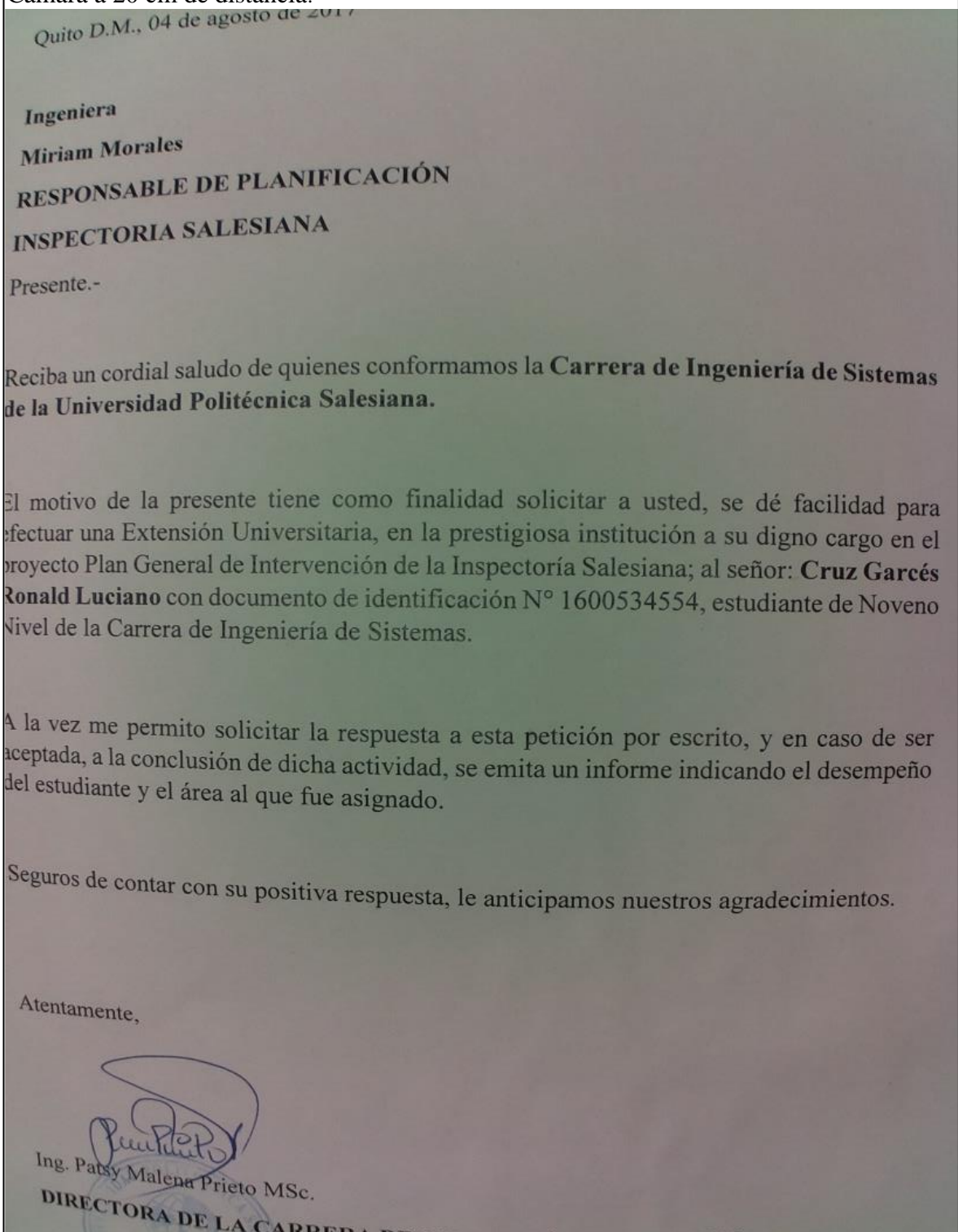


Figura 70. Prueba a 20 cm de distancia.

Elaborado por: Ronald Cruz.

Como segunda prueba de distancia, se la realizó a 30 cm dando como resultado un enfoque total hacia el documento impreso, como se puede observar en la figura 71.

Cámara a 30 cm de distancia.

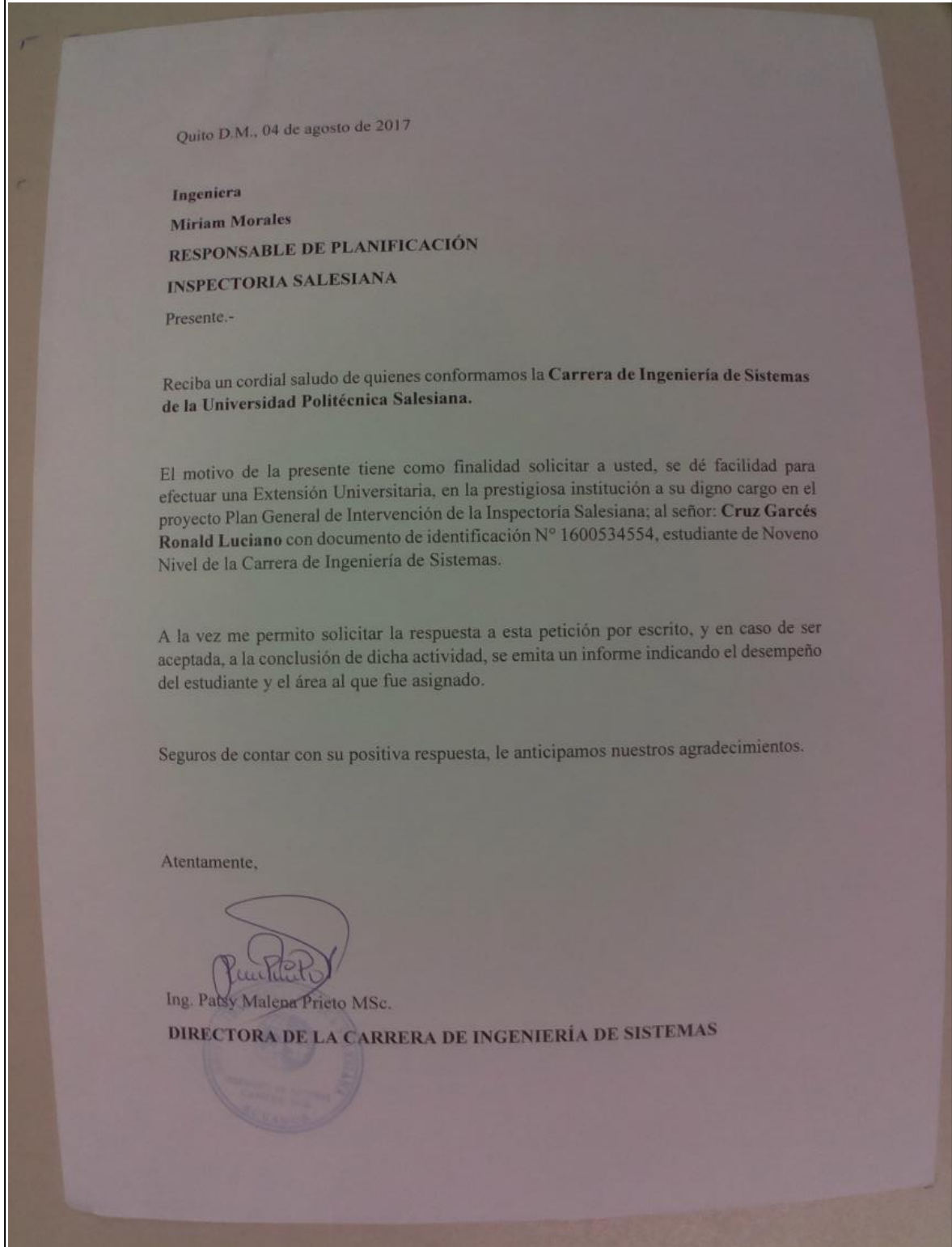


Figura 71. Prueba con cámara a 30 cm de distancia.
Elaborado por: Ronald Cruz.

Tercera prueba de distancia se la realizó a 40 cm, dando como resultado una captura de imagen distante y difícil reconocimiento de texto para el prototipo, como se puede observar en la figura 72.

Cámara a 40 cm de distancia.

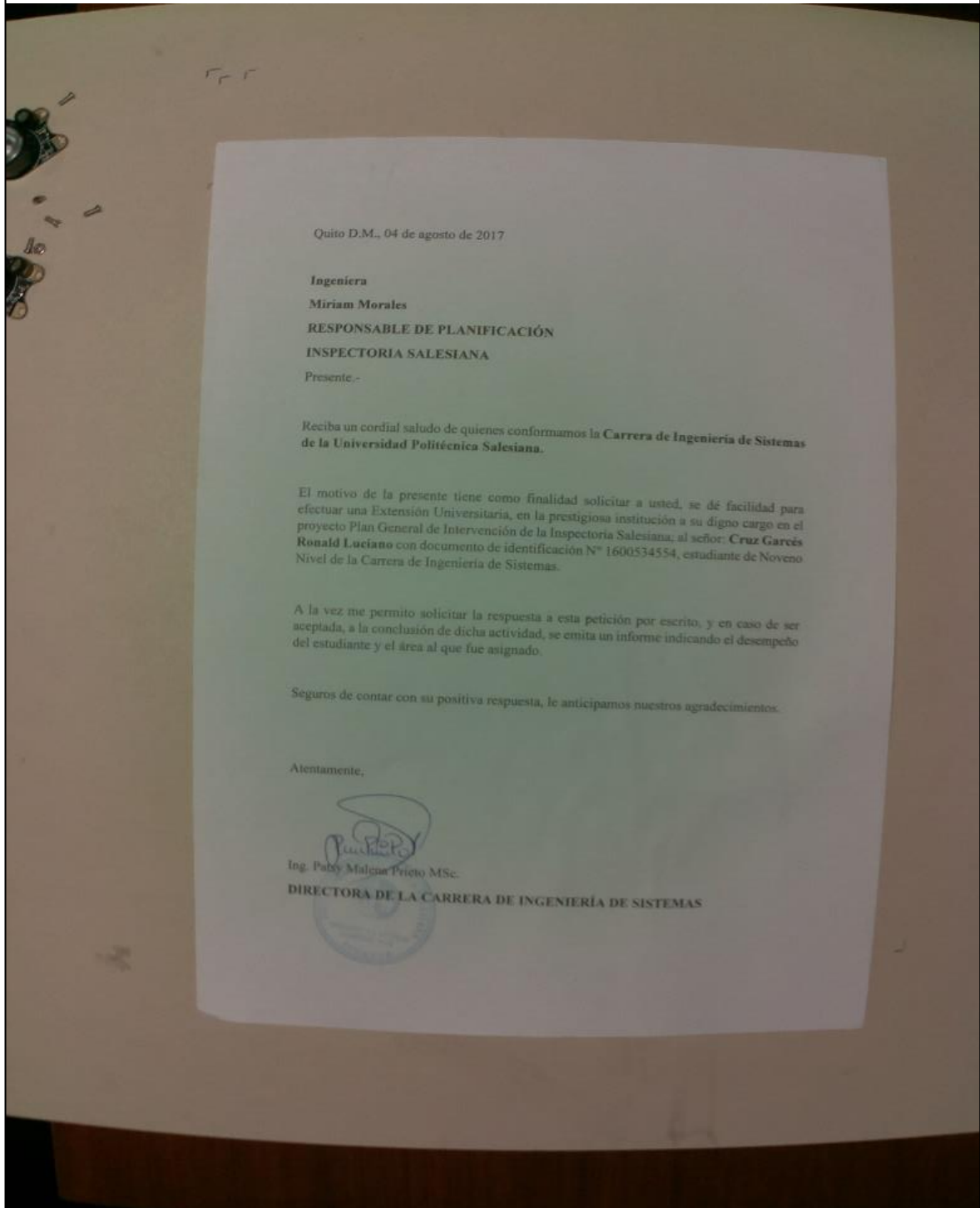


Figura 72. Prueba a 40 cm de distancia.
Elaborado por: Ronald Cruz.

Tabla 8. Pruebas de distancia.

| Pruebas de distancia | | |
|-----------------------------|-------------------------|---------------------------|
| Distancia | Lectura correcta | Lectura incorrecta |
| 20 cm | | x |
| 30 cm | x | |
| 40 cm | | x |

Nota: Distancia entre el lente de la cámara y la hoja impresa.

Elaborado por: Ronald Cruz.

3.4. Análisis de resultados

Luego de haber realizado las pruebas con impresiones diferentes, el prototipo tiene la posibilidad de leer texto de tamaño 6 en adelante, además se realizó pruebas de distancia entre la cámara y el documento desde 20 cm, 30 cm, 40 cm, dando como resultado la mejor distancia es 30 cm ya que permite al prototipo capturar una imagen óptima y reconocer la mayor cantidad de palabras, la siguiente prueba realizada fue con tres tipos de intensidad de luz dando como resultado de la carencia adecuada de luz puede llegar afectar al correcto funcionamiento del prototipo, ya que al capturar con poca luz el motor de detección presenta letras diferentes o simplemente envía la frase “Texto no detectado”, además se debe mencionar que el prototipo funciona de mejor manera con luz artificial ya que se realizó pruebas de luz de día y el enfoque de la cámara no es la óptima.

CONCLUSIONES

- El prototipo puede ayudar a las personas no videntes, por la facilidad de uso ya que cuenta con un modo automático y brindar una lectura comprensible a las personas que lo usen y se los podrá incluir en un ambiente laboral porque no tendrán el obstáculo al no poder leer un documento impreso.

- Se realizó una investigación acerca de los componentes usados para la creación de este prototipo, dando como resultado la mejor opción la tarjeta Raspberry Pi, ya que cuenta con los recursos necesarios para procesar este tipo de proyecto en el menor tiempo posible.

- Se pudo comprobar que el algoritmo utilizado para el reconocimiento de imágenes y la conversión a audio cuenta con la precisión y dar una lectura adecuada hacia la persona que lo está escuchando.

- Se observó que el prototipo es capaz de reconocer letras de tamaño de 6 en adelante, el tamaño en centímetros es de 0,6 cm en adelante, entregando un buen entendimiento a la hora de la reproducción del audio generado.

- Se realizó pruebas con diferente intensidad de luz y se pudo comprobar que mientras exista mejor iluminaria el prototipo entregará mejores resultados a la hora de procesar el documento a ser capturado, cabe recalcar que las pruebas fueron realizadas con una lampara de voltaje 5V a 2w de potencia, el prototipo funciona correctamente con 450 lumens que sería el equivalente a una bombilla LED utilizadas en el hogar.

- El prototipo desarrollado no fue probado con una persona con discapacidad visual, es por eso que en el diseño del prototipo se agregó unos bordes de referencia para que la persona que lo use pueda agregar correctamente la hoja que desea escuchar.

- El porcentaje de precisión del prototipo luego de haber realizado las pruebas respectivas es del 85.91% de efectividad.

- De acuerdo a las pruebas realizadas se puede determinar que la distancia óptima para el buen enfoque es 30 cm desde la cámara hacia el documento ya que a esa distancia enfoca el total de la hoja, la iluminación es una parte importante a la hora de procesar texto impreso ya que con una buena iluminación el prototipo puede reconocer mejor el texto y entregar mejor información a la persona que lo solicita, la tarjeta Raspberry Pi 4 ofrece los recursos necesarios para el correcto funcionamiento del prototipo ya que cuenta con 4 GB en RAM para su procesamiento, adicional es compatible con la cámara IR-CUT.

RECOMENDACIONES

- Se recomienda no exceder los 30 cm de distancia para la adecuada captura de la imagen a ser procesada, este valor asegura la toma completa de la hoja tamaño A4 e inferiores y podrá ser reconocido y leído por el prototipo.
- Además, se recomienda utilizar una cámara con una mejor resolución, por otra parte, se debe contar con una buena iluminación para que el prototipo pueda realizar de la mejor manera su trabajo.
- Se recomienda utilizar la tarjeta Raspberry Pi 4 B porque cuenta con los recursos necesarios para que el prototipo funcione muy bien sin tanta demora.

GLOSARIO DE TÉRMINOS

- **Discapacidad Visual:** Disminución de la capacidad visual de algunas personas.
- **Tarjeta SD:** Tarjeta de memoria para dispositivos digitales. SD (Secure Digital).
- **OCR:** Reconocimiento óptico de caracteres.
- **OpenCV:** Librería para el tratamiento de imágenes.
- **USB:** Universal Serial Bus.
- **SSH:** Secure Shell.
- **VNC:** Virtual Network Computing.

LISTA DE REFERENCIAS

- Arducam. (2020). *Cameras for RaspberryPi*. Obtenido de Cameras for RaspberryPi: www.arducam.com
- Arriarán, S. S. (2015). *Todo sobre Sistemas Embebidos*. Lima - Perú: Universidad Peruana de Ciencias Aplicadas.
- ASUS. (2019). *tinker board: tinker your way to the future*. Obtenido de tinker board: tinker your way to the future: <https://www.asus.com>
- ASUS. (2019). *tinker board: tinker your way to the future*. Obtenido de tinker board: tinker your way to the future: <https://www.asus.com>
- Chiark Organization. (22 de Noviembre de 2020). *Download PuTTY: latest release*. Obtenido de Download PuTTY: latest release: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- CO., Shenzhen Xunlong Software. (2020). *Orange Pi en español*. Obtenido de Orange Pi en español: <https://orangepiweb.es>
- CONADIS. (Marzo de 2020). *Consejo de Discapacidades*. Obtenido de <https://www.consejodiscapacidades.gob.ec>
- Díaz, C. A. (2019). *Programacion en JAVA I: El entorno de programación*. Buenos Aires: Six Ediciones.
- Fundación Raspberry Pi. (2020). *Acerca de: Raspberry Pi*. Obtenido de Raspberry Pi: <https://www.raspberrypi.org>
- Gutiérrez, A. P. (2016). *Python Paso a Paso*. Madrid: Ra-Ma SA.
- Ministerio de Educación de Chile. (2007). *Guía de apoyo técnico pedagógico: Discapacidad Visual*. Santiago de Chile: KDiseño.
- Norris, B. (1984). *Microprocesadores y microcomputadores y sistemas de derivacion para fuentes de potencia*. México: Compañía Editorial Continental.
- Organización Mundial de la Salud. (1994). *Manejo de la baja visión de los niños*. Córdoba(Argentina).
- Raspberry Pi Foundation. (2018). *Raspberry Pi 4*. Obtenido de Raspberry Pi 4: www.raspberrypi.org
- Raspberry Pi Foundation. (2020). *Raspberry Pi OS*. Obtenido de Raspberry Pi OS: <https://www.raspberrypi.org/software/>
- RealVNC Limited. (2020). *Descargar VNC*. Obtenido de Descargar VNC: <https://www.realvnc.com/es/connect/download/viewer/>
- SECO SPA. (2019). *UDOO X86 II*. Obtenido de UDOO X86 II: <https://www.udoo.org>