Universidad Rey Juan Carlos

Escuela Superior de Ciencias Experimentales y Tecnología

GRADO EN INGENIERÍA
DE TECNOLOGÍAS INDUSTRIALES

Trabajo Fin de Grado

SISTEMA PORTÁTIL DE RECONOCIMIENTO DE TEXTO PARA LA ASISTENCIA DE PERSONAS CIEGAS O CON DIFICULTAD EN LA VISIÓN

Víctor Manuel Navarro Pérez

Directores: María Cristina Rodríguez Sánchez y Rubén Nieto Capuchino

Curso Académico 2020/2021

Grado en Ingeniería de Tecnologías Industriales

Trabajo Fin de Grado

El presente trabajo, titulado **SISTEMA PORTÁTIL DE RECONOCIMIENTO DE TEXTO PARA LA ASISTENCIA DE PERSONAS CIEGAS O CON DIFICULTAD EN LA VISIÓN**, constituye la memoria correspondiente a la asignatura de Trabajo Fin de Grado que presenta D. **Víctor Manuel Navarro Pérez** en Ingeniería de Tecnologías Industriales. Este trabajo ha sido realizado en la **Universidad Rey Juan Carlos** en el **Departamento de Matemática Aplicada, Ciencia e Ingeniería de los Materiales y Tecnología Electrónica**, bajo la dirección de **María Cristina Rodríguez Sánchez** y **Rubén Nieto Capuchino**.

Móstoles, 14 de Julio de 2021

Víctor Manuel Navarro Pérez

Grado en Ingeniería de Tecnologías Industriales

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE FÓRMULAS

1. RESUMEN

Debido a la dificultad que tienen las personas con problemas en la visión, ciegas o analfabetas a la hora de tratar con un texto físico y la desventaja, tanto en el mundo laboral como en el personal, que sufren al no ser capaces de leerlo, sumado al alto precio que tienen los dispositivos lectores capaces de compensar esta deficiencia, el alumno Víctor Manuel Navarro Pérez comenzará el desarrollo de dispositivo lector mucho más económico, ya que costaría aproximadamente 60 €, y que será capaz de transcribir y reproducir un texto físico mediante una voz sintética.

Este proyecto ofrecerá una alternativa viable y económica con respecto a los dispositivos actuales, ya que estaría desarrollado por completo a través de elementos de software de código abierto y hardware libre, lo que abarataría por mucho su coste y supondría una mayor facilidad para la adquisición de los componentes y la replicación de las funcionalidades para un usuario interesado. Además, será un dispositivo pequeño lo suficientemente portátil para caber en un bolsillo, lo cual presentaría una gran ventaja con respecto a la mayoría de las opciones que se encuentran ahora en el mercado, y estará desarrollado teniendo a las personas ciegas en mente, diseñado con una carcasa que se adaptaría a sus necesidades.

El diseño y la implementación del dispositivo se han ideado para ser evaluado en diferentes formatos, con distintas fuentes y en diversas condiciones de luminosidad. El dispositivo hace uso de una cámara para capturar una imagen del exterior, un software de variación de imagen para hacer que la imagen presente unas características óptimas, un software OCR (*Optical Character Recognition*) para transcribir el texto, un software TTS (*text to speech*) para transformar el texto a voz y un reproductor de sonido que reproduce el archivo de voz a través de unos altavoces o un dispositivo alternativo conectado, como unos auriculares.

Una vez desarrollado el dispositivo, se ha comprobado su practicidad con una persona con problemas en la visión y se han anotado posibles mejoras realizables, presentando un gran futuro potencial. Además, se ha tramitado una solicitud para que su diseño se recoja dentro de un marco de “modelo de utilidad” para defender la novedad de la estructura y se ha liberado el método de software diseñado en internet por si alguna persona quiere replicar el dispositivo para uso personal o mejorarlo.

2. INTRODUCCIÓN

La lectura es un medio de comunicación de suma importancia en el mundo actual para las personas. A través de ella se comunica todo tipo de información, desde la más irrelevante, como la marca de un frigorífico, hasta la más vital, como una señal de peligro por toxicidad, pasando por saberes universales que pueden guiarnos a entender mundo que nos rodea, a hacer cosas que no habíamos hecho nunca, a meternos en la mente de otra persona e, incluso, a vivir.

Desde la aparición de la escritura cuneiforme en la Sumeria del año 3.500 A.C., la lectura cambió por completo el proceso de desarrollo del ser humano. Acababa de surgir un medio que permitía la transmisión de conocimientos entre personas que podrían encontrarse a una diferencia de kilómetros o incluso siglos entre ellas, dejando atrás la posible inexactitud y limitación que conllevaba la transmisión exclusivamente oral. No solo otorgaba la capacidad de difundirse a un número mucho mayor de personas, sino que daba al conocimiento la posibilidad de hacerse virtualmente inmortal [1].

La escritura ha sido la raíz de muchos otros avances que se consideran claves para el desarrollo de la humanidad. De ella derivó la aritmética, la cual permitió usar y desarrollar las matemáticas, así como la física o la contabilidad. Por otro lado, se pudo desarrollar más el modelo de estado, al tener la posibilidad de plasmar las leyes, solidificar tradiciones, entre otros [1]. Es una herramienta exclusiva de la humanidad que ha permitido, a ésta, colocarse a un nivel de complejidad superior al de cualquier otro animal de la Tierra y define, en gran parte, lo que es el ser humano.

Sin embargo, esta poderosa herramienta tiene una condición, se trata de una herramienta visual. Esto implica que, mientras que a una persona instruida en la lectura y con buena visión puede resultarle trivial leer un texto impreso, las personas analfabetas, con problemas en la visión o ciegas se enfrentan a muchas más dificultades cuando se trata de la lectura de textos.

Actualmente, aproximadamente 750 millones de personas en el mundo no saben leer ni escribir [2] y, solo en España, según un estudio del 2016, el número de personas analfabetas funcionales de más de 16 años de edad asciende hasta las 700.000 [3]. Además, se estima que, a nivel mundial, aproximadamente 2.200 millones de personas tienen algún tipo de deficiencia visual, siendo la mayoría de ellas mayores de 50 años.

La mayoría de las deficiencias visuales son causadas por errores oculares de refracción y cataratas y, de esos 2.200 millones de personas mencionadas anteriormente, aproximadamente 1.000 millones presentan un deterioro moderado o grave en la visión [4]. Hasta hace poco, estas

personas se encontraban con serias dificultades a la hora de equipararse con otra que no tuviese deficiencias en la visión si su deficiencia no podía compensarse con gafas u operación ocular.

Para aquellas personas con deficiencia severa en la vista, la única forma que tenían para poder leer algo era recurrir a textos escritos en Braille, que es un sistema táctil ideado a mediados del siglo XIX basado en 6 puntos sobresalientes y agrupados en distintas distribuciones que permiten obtener hasta 64 combinaciones diferentes [5]. Sin embargo, este método exige que el texto esté adaptado para usar esta modalidad de lectura y, además, exige el conocimiento o aprendizaje del método que, si bien una persona ciega de nacimiento ha podido aprender como método sustitutivo de la lectura convencional, para una persona que inicialmente no lo era y ha adquirido un deterioro de la visión con el paso de los años representaría también una barrera extra que tendría que sobrepasar.

Afortunadamente, hoy en día, con el avance tecnológico e internet, las personas que tengan algún tipo de deficiencia o dificultad en la visión cuentan con bastantes más ayudas que antaño. Entre ellas, podemos encontrar programas que pueden leer un texto digital usando un software tipo *text to speech* (de texto a discurso), que adapta el texto escrito a una voz sintética que se reproduce. Los ejemplos más conocidos de este tipo de ayudas son "JAWS" (*Job Access With Speech*) [6] que lee el contenido de una pantalla electrónica, y "DAISY" [7] que además permite adaptar los libros digitales (o escaneados) y convertirlos en *multimedia books*, una especie de libros que se leen automáticamente sin interacción externa [8]. Además de éstas, existen otras ampliamente usadas como las opciones "lee en voz alta" del "Adobe Reader" [9] la funcionalidad con la misma función de "Microsoft Word" [10] la de Google (GTTS) [11] entre otras [12].

Por otro lado, existen múltiples audiolibros que se pueden encontrar en internet y también existen múltiples esfuerzos por convertir el entorno general de internet en un lugar mucho más accesible para personas con discapacidades, destacando la World Wide Web Consortium [13] que establece ciertas recomendaciones para hacer los contenidos de las páginas web más accesibles para todo el mundo.

También están apareciendo dispositivos electrónicos, impulsados por la mejora en la tecnología como puede ser el "Nokia Braille Reader" [14] una *APP* diseñada para poder transcribir los mensajes SMS recibidos en un *smartphone* a Braille, haciendo uso de la retroalimentación táctil del móvil, para facilitar la lectura de ellos. Un punto negativo puede ser que sólo funciona en determinados dispositivos Nokia, haciendo que no sea accesible para personas que tengan un dispositivo distinto.

Todo lo marcado arriba son avances importantes. Sin embargo, todos estos programas, aplicaciones y ayudas necesitan un texto que esté en formato digital para poder leerlo. Esto puede ser un punto bastante limitante para cierto grupo de la población, como el caso de una persona anciana que no está habituada al uso de la tecnología.

Si hablamos de la lectura de un texto no digital, las personas con dificultad o discapacidad visual están más limitadas, aun contando con la posibilidad de escoger un texto adaptado, como libro con la letra suficientemente grande para que el lector la pueda ver o los libros y textos especiales escritos en braille. Para los textos que no están adaptados, existe la posibilidad de escanear el texto para usar uno de los programas descritos anteriormente o de usar la cámara del smartphone con alguna aplicación de traducción como "Google Translate" [15] para habilitar captura y lectura de texto. Sin embargo, el uso de estas aplicaciones es bastante complicado para una persona ciega.

Existen dispositivos, como "Scanmarker Air" [16] con forma de bolígrafo, capaces de reconocer texto y reproducirlo al pasarlo por las líneas de un texto, cubriendo así las necesidades de una persona con dificultad en la visión. Sin embargo, las personas ciegas tendrían dificultades para usarlo debido a su funcionamiento, ya que se requiere saber dónde están las líneas por las que hay que pasar el dispositivo.

En los últimos años, ha habido cierto interés para sobrepasar esta limitación. En este sentido, en 2015, se empezó a comercializar "OrCam" [17], un dispositivo que se acopla a una patilla de las gafas y que permite reconocer y leer automáticamente textos mediante inteligencia artificial. Ha continuado mejorando sus prestaciones desde entonces y ha ido incorporando funcionalidades como el reconocimiento facial. Sin embargo, su mayor inconveniente es su alto precio, que puede alcanzar y superar los 3.500 €, siendo un importante costo que muchas familias no pueden permitirse [18].

En 2017 apareció "Seeing AI" [19], una aplicación creada por Microsoft que usa la cámara de un iPhone [20] para realizar una lectura instantánea del texto. No obstante, solo está disponible en iOS [21] y requiere de cierta adaptación para usarla, además de tener conocimientos de uso del Smartphone.

En la tabla 1, se muestra un resumen de las diferentes tecnologías y dispositivos comentados anteriormente.

Tabla 1: Comparación entre las distintas ayudas que existen.

	JAWS o DAISY	**Adobe Reader, Microsoft Word, GTTS**	**Nokia Braille**	**Google Translate**	**Scanmarker Air**	**OrCam**	**Audiolibros**	**Seeing AI**
Portátil	Parcial	Parcial	Sí	Sí	Sí	Sí	Parcial	Sí
Necesitad de digital	Sí	Sí	Sí	No	No	No	Sí	No
Conocimiento de Braille	No	No	Sí	No	No	No	No	No
Apto para ciegos	Sí	Sí	Sí	No	No	Sí	Sí	Sí
Facilidad de uso	Media	Fácil	Fácil	Difícil	Fácil	Fácil	Media	Media
Fácilmente obtenible	Parcial	Sí	Parcial	Sí	Parcial	No	Sí	Parcial
Precio	Caro	Gratuito	Caro	Gratuito	Caro	Muy caro	Medio	Gratuito

Todas y cada una de estas opciones presenta distintas ventajas e inconvenientes. Sin embargo, no parece haber ningún dispositivo que sea capaz de aunar todas las ventajas en él, haciendo que una persona ciega deba hacer uso de una combinación de estas funcionalidades dependiendo de la situación y texto que quiera leer. Con el objetivo de facilitar lo máximo posible la lectura a cualquier tipo de personas con dificultades, en este Trabajo Fin de Grado se presentará un dispositivo sencillo que permita leer cualquier tipo de texto y con un precio asequible.

Este documento está organizado de la siguiente manera; en el apartado 3 se muestran los objetivos; en el apartado 4 se realiza la planificación del proyecto; en el apartado 5 se realiza el planteamiento teórico; en el apartado 6 se detalla el montaje y el código realizado. Los resultados obtenidos se exponen en el apartado 7 y en el apartado 8 se recoge el presupuesto. Por último, las conclusiones y líneas futuras se exponen en el apartado 9.

3. OBJETIVOS

Debido a la carencia de un dispositivo en el mercado actual que aúne todos los beneficios marcados en la tabla 1, el principal objetivo y motivación que se tiene en este Trabajo Fin de Grado es tratar de desarrollar un dispositivo asequible económicamente para la mayor parte de la población mientras que éste mantiene una funcionalidad correcta. Para cumplir este objetivo de la mejor manera se deberán cumplir, en la medida de lo posible, la mayoría de los puntos mostrados en la tabla 1.

En vista de esto, los objetivos que se plantearán para este Trabajo Fin de Grado serán los siguientes:

- Diseño y creación de un dispositivo capaz de leer un texto en formato físico a partir de materiales de bajo coste, con el fin de que sea asequible. Además, deberá ser portátil y fácil de usar.
- Diseño de una cubierta o sistema que facilite su uso, poniendo especial atención en características que puedan ayudar a una persona ciega a utilizarlo.
- Implementación de un programa que sea capaz de ejecutar todo el proceso de lectura y reproducción del texto usando el dispositivo diseñado.
- Analizar la precisión de reconocimiento y aplicar mejoras para aumentar la capacidad de lectura de varios tipos de texto con diversas características, como distintos tipos de fuentes, tamaños de letra, iluminación, colores o distribución.

De forma adicional se pretenderán alcanzar los siguientes objetivos opcionales:

- Analizar y optimizar el dispositivo para personas con problemas en la visión, ciegas o analfabetas, realizando pruebas de su utilidad con alguien que lo requiera.
- Definir un modelo de utilidad para el dispositivo por su practicidad e intentar que el dispositivo sea fácilmente replicable por personas sin demasiados conocimientos técnicos que quieran hacer un uso personal de él.

4. PLANIFICACIÓN DEL PROYECTO

Para cumplir los objetivos marcados y alcanzar el desarrollo del Trabajo Fin de Grado, se organizarán las tareas en un orden lógico específico.

1. En primer lugar, se deberá investigar y definir los componentes existentes en el mercado actual y la combinación de ellos que puede dar como resultado la funcionalidad propuesta. Para ello, se deberán seleccionar los componentes con la mejor opción calidad-precio, así como un tamaño reducido. Una vez que se seleccionen los componentes, se realizará la compra.
2. Cuando se hayan obtenido los componentes, se procederá al diseño de su distribución en el espacio y ensamblaje, con el objetivo de que ocupen el menor espacio posible y alcanzar el objetivo de realizar un dispositivo portable.
3. Con los componentes ensamblados, se realizará la programación de cada funcionalidad, así como el diseño de una carcasa exterior para integrar todos los componentes. De esta forma será posible empezar a tener resultados de software y, al mismo tiempo, proporcionar cierto aspecto final al dispositivo.
4. Una vez determinado el diseño de la carcasa, se iniciarán los trámites para realizar el modelo de utilidad, presentándolo como proyecto en desarrollo. Esta tarea es un proceso lento que se prolongará incluso hasta después del final del Trabajo Fin de Grado. Sin embargo, es una labor administrativa que no debería requerir especial dedicación.
5. El proceso de impresión 3D de la carcasa se realizará cuando se revise el diseño, sin embargo, debido a posibles cambios que pudiesen surgir durante el desarrollo, se llevará a cabo en la última fase del Trabajo Fin de Grado.
6. Por último, una vez se haya montado el dispositivo en el interior de la carcasa, generando así el primer prototipo, sería conveniente probarlo con personas ciegas para determinar su funcionalidad. Con esta prueba se tratarán de corregir las diferentes dificultades que se puedan encontrar, así como implementar y mejorar las diferentes sugerencias en las siguientes versiones del dispositivo.

Además de lo expuesto anteriormente, se deberá tener en cuenta el tiempo para la realización de la documentación del Trabajo Fin de Grado, la cual comienza cuando se tienen los diseños de la carcasa terminados. Se estimará que para la realización de la memoria se invertirán 110 horas, aproximadamente.

Todo lo descrito anteriormente se va a distribuir en el tiempo, adecuándose al tiempo de dedicación del Trabajo Fin de Grado, determinando, en este caso, un periodo de cinco meses para su la realización. En la tabla 2 se muestra dicha distribución.

Tabla 2: Diagrama de Gantt del proyecto.

Actividades	Tiempo de realización en semanas																					
	Febrero				Marzo				Abril				Mayo				Junio				Julio	
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Investigación y selección de componentes																						
Compra de los componentes																						
Diseño del dispositivo																						
Ensamblaje del dispositivo																						
Programación y primera optimización																						
Diseño de la carcasa																						
Proceso para el modelo de utilidad																						
Impresión 3D de la carcasa																						
Evaluación y optimización																						
Escritura del TFG																						

5. PLANTEAMIENTO TEÓRICO DEL PROYECTO

En este apartado se describe y explica el proceso y los criterios de selección, tanto de los componentes de hardware como de los elementos de software, añadiendo, al final, un planteamiento teórico del funcionamiento del dispositivo.

5.1 SELECCIÓN DE COMPONENTES DE HARDWARE

Para la realización de este dispositivo se va a tratar de seleccionar componentes con las siguientes características:

- Tamaño reducido, para que sea lo más portable posible.
- Precio reducido con el fin de que el dispositivo sea económico.
- Alta funcionalidad en su campo, para que sea lo más eficiente posible

A la hora de seleccionar los componentes hay que adecuarse lo más posible a estas tres características, encontrando un balance entre ellas, ya que las tres son indirectamente proporcionales entre sí.

En la figura 1 se muestra un diagrama de bloques de los diferentes módulos del dispositivo. A continuación, se describen cada uno de los cinco módulos según su funcionalidad.

Módulo de procesamiento: Este módulo es la unidad principal de procesamiento y sirve de base de conexión para todos los demás módulos, dirigiendo las tareas e instrucciones que deben realizar, además de procesar otras instrucciones en el mismo para alcanzar la funcionalidad deseada del dispositivo.

Módulo de sonido: Este módulo se encarga de decodificar una señal de audio recibida del módulo de procesamiento con el fin de reproducirlo a través de unos altavoces, auriculares, etc. Sería conveniente que los componentes seleccionados para suplir la función de este módulo incluyesen distintas vías de reproducción para una mayor comodidad y accesibilidad.

Módulo de energía: Este módulo proporciona energía a todos los componentes del sistema. Es preferible que el componente que desempeñe esta función sea pequeño y con gran capacidad para una mayor portabilidad.

Módulo de captura: Este módulo se encarga de recopilar información visual del exterior y transmitirla, como información de entrada, al módulo de procesamiento para su interpretación.

Módulo de interacción: Este módulo permite al usuario manipular e interactuar con el dispositivo mediante el envío de comandos, controlados por el usuario a través de acciones, al módulo de procesamiento.

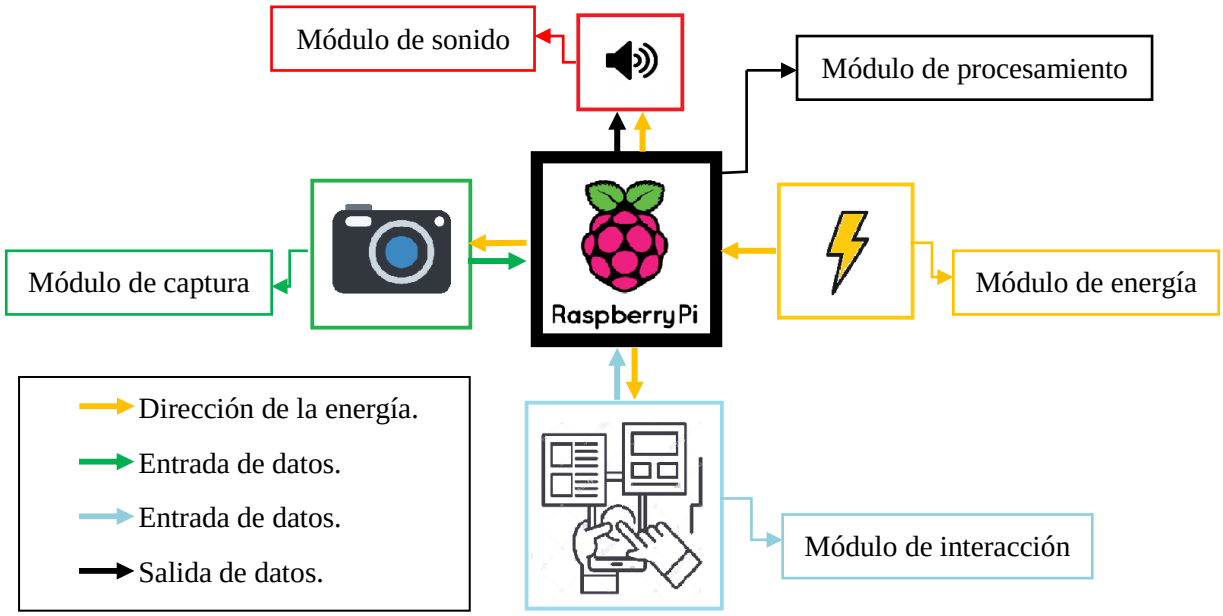


Figura 1: Diagrama de bloques de los módulos del dispositivo.

5.1.1 MÓDULO DE PROCESAMIENTO

Para realizar este proyecto de un modo económico se ha optado por usar una "Raspberry Pi Zero" [22] como unidad base de procesamiento, eligiendo los demás componentes en función de esta decisión.

El hecho de usar una "Raspberry" consiste principalmente porque su modelo "Zero" tiene unas dimensiones reducidas (6,5 x 3 x 0,5 cm), haciendo que el dispositivo final sea portátil. Además, teniendo en cuenta que el proyecto usa técnicas de visión artificial, la Raspberry presenta también una mayor potencia frente a otras alternativas como puede ser el "Arduino Nano" [23] en estos ámbitos. En la figura 2, se muestra una clasificación de dispositivos utilizados para visión artificial. Como se puede apreciar, otros dispositivos como las FPGA o la "Nvidia Jetson" [24] presentan una potencia mayor que la Raspberry Pi Zero, sin embargo, son mucho más complejas de utilizar, su precio es mayor y sus dimensiones son, generalmente, superiores, por lo tanto, no se adecuan a las características principales que debe tener el dispositivo.

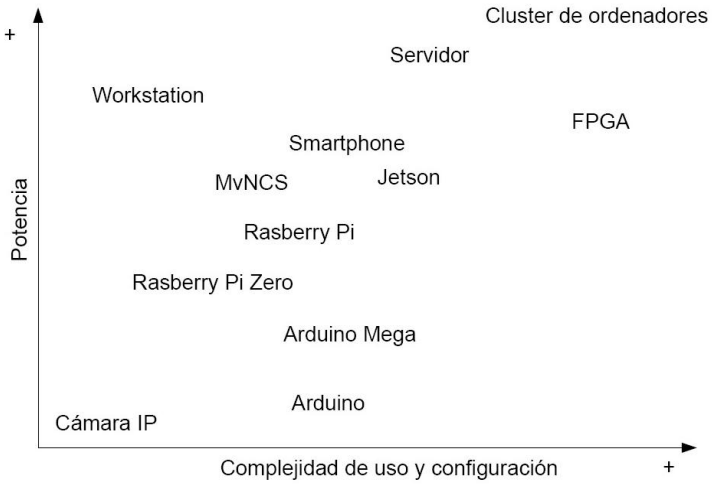


Figura 2: Clasificación de dispositivos para visión artificial [25].

Por otro lado, la "Raspberry Pi" permite usar "Python" [26] un lenguaje de programación sencillo de utilizar y con una amplia cantidad de librerías y una gran comunidad en la red, pudiendo facilitar mucho el desarrollo del dispositivo.

5.1.2 MÓDULO DE SONIDO

La Raspberry Pi Zero no cuenta con un módulo con conexiones de salida de sonido en la tarjeta de desarrollo y no cuenta, tampoco, con el controlador que lo pueda decodificar. Por lo tanto, requiere crear o adquirir un dispositivo que pueda decodificar un audio y transmitirlo a un dispositivo de reproducción.

Una opción podría haber sido diseñar un circuito electrónico que hiciese esta función, sin embargo, y pese a que quizá hubiese sido la opción más barata, la complicación del proyecto crecería innecesariamente y no marcaría una diferencia tan importante en cuanto a precio de otras opciones ya creadas. Además, es una opción que resultaría menos accesible para personas sin conocimientos de electrónica o sin los componentes para realizar y montar un circuito impreso.

La alternativa que se ha considerado es utilizar una tarjeta tipo HAT (*Hardware Attached on Top*), que se conecta en la parte superior de la Raspberry, a través de los pines que tiene para este propósito, actuando como tarjeta de sonido.

Después de valorar diferentes modelos de tarjetas HAT, se ha optado por la tarjeta "ReSpeaker 2-Mics Pi HAT" [27] ya que es una de las opciones más económicas y se acopla a la perfección a la Raspberry sin aumentar demasiado el tamaño total. Además, cuenta con características adicionales como:

- Un botón programable, que se usa como método de activación del proceso de fotografía del dispositivo.
- Una conexión tipo *mini-jack*, que permite la conexión de unos auriculares.
- Dos micrófonos, que permite realizar implementaciones de control por voz.
- Comunicación I2C y GPIO para la conexión de otros dispositivos como una pantalla, sensores o altavoces.

A continuación, se debe elegir el dispositivo de reproducción predeterminado. Entre las salidas de sonido admitidas en el HAT seleccionado anteriormente, la opción que más integra el dispositivo consiste en unos altavoces conectados al dispositivo. Después de evaluar varios modelos, se ha seleccionado un altavoz de forma oval de 1 W y 8 Ohmios. Cuenta con una conexión tipo *Molex Picoblade* que tiene utilidad para una conexión con el HAT, sin embargo, como se expondrá más adelante, no se ha utilizado este conector por incompatibilidades.

5.1.3 MÓDULO DE ENERGÍA

Para alimentar el dispositivo existen múltiples opciones, entre ellas la conexión directa a través de un regulador de 5 V a una toma de corriente. Sin embargo, puesto que uno de los objetivos era hacer el dispositivo lo más portable posible, esta opción queda descartada.

Teniendo en cuenta esto, las otras opciones que se pueden implementar son:

- Alimentación mediante pila alcalina.
- Alimentación mediante pila de botón.
- Alimentación mediante batería.

A continuación, se detallan las características de cada una de ellas.

- **Alimentación mediante pila alcalina.**

  Tras realizar una búsqueda de diferentes portapilas que permitan alimentar el dispositivo, se han encontrado múltiples soportes para pilas AA con un reducido coste. Sin embargo, presenta unos inconvenientes, que son su tamaño elevado y la tensión que proporcionan.

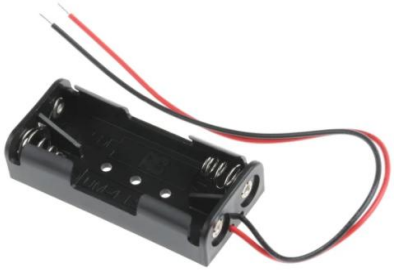


Figura 3: Adaptador para dos pilas AA [28].

  La Raspberry necesita una tensión de unos 5 V para funcionar [29]. Una pila alcalina, por el contrario, proporciona 1,5 V. Para obtener una tensión de 5 V se requieren 3 pilas AA o AAA, haciendo que las dimensiones del dispositivo se incrementen de forma notable. Se puede optar por utilizar un portapilas de 2 pilas AA o AAA, como el que se muestra en la figura 3, y utilizar un convertidor DC-DC *Step up* para proporcionar los 5 V. Sin embargo, el tamaño del conjunto se vería incrementado.

- **Alimentación mediante pila de botón.**

  Una pila de botón tiene un tamaño más reducido que una pila alcalina. En el mercado existen soportes para pilas a un precio reducido que pueden albergar entre una y dos pilas de botón. Puesto que una pila de botón proporciona una tensión de 3V, con dos de ellas se obtiene un voltaje superior a los 5V que requiere la Raspberry. Sin embargo, como se puede ver en la figura 4, los portapilas de dos pilas de botón tienen una forma alargada, lo cual incrementa el tamaño y, además, su precio es diez veces superior al de los adaptadores de una sola pila, figura 5.

  Por otro lado, la autonomía que pueden las pilas de botón es reducida, ya que la capacidad no supera los 200 mAh. Por tanto, puede que no sea suficiente para alimentar este dispositivo durante un tiempo considerable.

Figura 4: Adaptador para dos pilas de botón [30].

Figura 5: Adaptador para una pila de botón [31].

- **<u>Alimentación mediante batería</u>.**

  La alimentación mediante batería requiere dos cosas; la batería y el controlador de la batería. Hay una gran variedad de tamaños y capacidades, sin embargo, la opción más apropiada es una batería preparada para la Raspberry Pi Zero. Estas integran el controlador de la batería y conservan la forma de la Raspberry, haciendo que sea más fácil de acoplar. Sin embargo, la gran mayoría de este tipo de baterías están diseñadas en forma HAT y, puesto que ya se está usando para la tarjeta de sonido una placa tipo HAT, no es posible acoplarlas a la Raspberry salvo que se modificaran los pines que incorpora el HAT de la tarjeta de sonido.

  Por otro lado, hay una batería que no se conecta mediante HAT, sino a los contactos que tiene la Raspberry, siendo ésta la "PiSugar Portable" [32] que se muestra en la figura 6. Esta batería "PiSugar" cuenta con dos modelos, uno de 900 mAh (PowerPack M) y otro de 1200 mAh (PowerPack L), cuyas características se resumen en la Tabla 3.

Tabla 3: Diferencias entre los modelos de batería PiSugar [33].

PiSugar

	PiSugar PowerPack M	PiSugar PowerPack L
Output Current	1.2A	2.4A
Battery Capacity	900mAh	1200mAh
Battery Life	3 ~ 4 Hour	5 ~ 6 Hour
USB Charging Port	mircoUSB	mircoUSB
Size (board)	65x30x1.5 mm	65x30x1.5 mm
Size (battery)	65x30x4 mm	52x27x10 mm
Weight	39.5g	44.5g

Debido a las dimensiones del modelo de 900 mAh, éste es adecuado para el propósito del proyecto. Sin embargo, no está disponible, y sólo se encuentra el modelo de 1200 mAh, que tiene un tamaño superior, haciendo el dispositivo menos manejable.

Figura 6: Baterías PiSugar [33].

Por otro lado, estas baterías cuentan con la desventaja de que tienen un precio muy superior a las otras dos opciones presentadas anteriormente, siendo el precio del modelo de 1200 mAh de la PiSugar de unos 40 € [33]. Esto implica que el precio final del dispositivo se incremente de forma notable, sin embargo, se trata de una buena opción a considerar.

Con todas estas consideraciones anteriores, se ha decidido utilizar un soporte para una pila de botón, junto con un convertidor DC-DC de 3 V a 5 V. Esta opción es la más económica y se adecua con el reducido tamaño del dispositivo, sin embargo, se puede cambiar a alimentación por batería si la opción de la pila de botón no resulta suficiente.

5.1.4 MÓDULO DE CAPTURA

Como la cámara debe de ir conectada directamente a la Raspberry Pi Zero, había que seleccionar una cámara que fuese compatible con la misma. Además, la Raspberry Pi Zero cuenta con un puerto de conexión de cámara distinto al de una Raspberry Pi normal [34].

Para elegir una cámara compatible se puede optar, bien por una cámara cualquiera compatible con cualquier Raspberry, pero con un adaptador *flex* para Raspberry Pi Zero; o bien por una cámara especializada en Raspberry Pi Zero con el conector flex preparado.

La variedad de marcas, modelos y tipos de cámaras para Raspberry, en general, es muy diversa, como cámaras con ojo de pez de distintos grados, cámaras sin filtro de infrarrojos, que son utilizadas para aplicaciones de visión nocturna, cámaras con enfoque variable, etc. En definitiva, todos ellos tienen en común que suelen estar acopladas a una placa controladora con unas dimensiones que, para este proyecto, quizá si puedan considerarse algo elevadas, teniendo en cuenta que la cámara se va a colocar en el frontal del dispositivo final.

Las cámaras preparadas para la Raspberry Pi Zero, por el contrario, suelen tener un tamaño más reducido, ocupando la placa controladora lo mismo que el tamaño de la cámara. Además, su precio no es mucho más elevado que las mencionadas anteriormente. No obstante, como punto negativo, su variedad es mucho más limitada, estando solo disponibles un modelo de cada tipo.

Finalmente se ha seleccionado una cámara específica para Raspberry Pi Zero sin filtro infrarrojo, concretamente el modelo "ZeroCam NOIR without infrared filter" [35]. Este modelo presenta la desventaja de que no está preparada para ajustar el enfoque, que se puede adaptar, pero para eso se debe manipular la cámara y ésta se puede romper durante el proceso. Por tanto, para este proyecto se utiliza la cámara sin modificar.

5.1.5 MÓDULO DE INTERACCIÓN

Puesto que la tarjeta de sonido cuenta con un botón programable, se tiene el módulo de interacción cubierto, ya que sirve de activación para el proceso programado. Además, el soporte para pila de botón cuenta con un interruptor que será usado para el encendido y apagado del dispositivo.

No se requiere añadir más interacción con el usuario, sin embargo, como se ha mencionado anteriormente, existe la posibilidad de utilizar el conector I2C para incluir alguna pantalla o usar el GPIO para incluir alguna funcionalidad adicional relacionada con la interacción con el usuario como posible mejora en un futuro diseño.

5.2 SELECCIÓN DE ELEMENTOS DE SOFTWARE

A continuación, se va a describir brevemente los distintos programas usados para realizar cada una de las funciones del software del dispositivo, justificando la elección de cada uno de ellos. Además, se mencionan otros programas que se han utilizado en el desarrollo del producto con el fin de dejar clara la función de todos. En concreto se analizarán cinco apartados: sistema de reconocimiento de textos, sistema de alteración de imagen, sistema de transformación de texto a voz, sistema de reproducción y otros programas utilizados.

5.2.1 SISTEMA DE RECONOCIMIENTO DE TEXTOS

Encontrar o desarrollar un software preciso que permita el reconocimiento de textos a partir de una imagen es uno de los puntos clave del proyecto, ya que, alrededor de este concepto, se desarrolla todo el producto. La capacidad antes descrita es conocida como OCR (*Optical Character Recognition*) o reconocimiento óptico de caracteres.

Desde su primera aparición en la Alemania de 1929 con una invención de Gustav Tauschek, los sistemas OCR han sufrido ciertos cambios. En sus orígenes consistían en una máquina mecánica dotada de un fotodetector y un engranaje que contenía un patrón de letras y giraba cuando se producía alguna coincidencia entre el fotodetector y el patrón, haciendo girar un tambor de impresión y plasmando el resultado en un papel [36].

Más adelante, con la llegada de los computadores, los OCR evolucionaron para que, en vez de imprimir el resultado en un papel, lo transformase en datos para ser leídos por una computadora. Sin embargo, seguían siendo sistemas mecánicos y no tuvieron mucho éxito en el mercado [37].

No fue hasta 1974 que se creó el primer OCR digital por Raymond Kurzweil, capaz de reconocer texto a partir de un escáner [38] y, a partir del sistema de funcionamiento de éste, se desarrollaron los OCR más avanzados que se encuentran en la actualidad.

El funcionamiento general de un OCR digital es común entre todos los que hay es:

1. Se produce una caracterización de la imagen que puede ser mediante una binarización (poner la imagen en blanco y negro usando un umbral) o una segmentación en escala de grises.
2. Se segmenta la imagen a partir de los contornos o regiones de ésta, para aislar las zonas que contengan texto. A continuación, se descompone el texto en distintas entidades lógicas, tomando como guía los espacios entre palabras, que a su vez se descompondrán en unidades lógicas aún más pequeñas, correspondiendo éstas a cada letra de la palabra.
3. Se reducen los componentes de cada bloque lógico mediante la eliminación sucesiva del contorno, logrando así obtener unas formas estilizadas con la proporción original.
4. Por último, se compara cada bloque con unos patrones y se elige el resultado más aproximado usando un porcentaje de probabilidad [39].

Sin embargo, los OCR tienen una desventaja, necesitan unas condiciones muy determinadas para funcionar de forma óptima. Entre ellas, se debe tener una imagen a analizar con una buena calidad y nitidez, sin gradientes de luz o variación cromática importante y con una fuente y tamaño de texto constante.

Actualmente, existen varios programas OCR digitales en el mercado, varios de ellos gratuitos. Sin embargo, al ser una tecnología todavía en desarrollo, muchos de ellos son imprecisos si no se cumplen unas condiciones específicas mencionadas anteriormente. Uno de los programas gratuitos más flexibles y precisos que existen se llama “Tesseract OCR”, que usa una red neuronal para la comparación de patrones [40] y es el que se ha seleccionado para este proyecto, en concreto, en la versión de Python, que es la librería “PyTesseract”.

Tesseract OCR es un programa desarrollado por HP que pasó a ser de código abierto a partir de 2005 y que ha estado financiado por “Google” desde el 2006 [41]. Es considerado el sistema OCR de código abierto más preciso que existe a hoy en día [42].

5.2.2 SISTEMA DE ALTERACIÓN DE IMAGEN

Para intentar compensar los posibles defectos que se puedan encontrar en la imagen y adecuar la imagen a las características con las que el OCR funciona de forma óptima, es necesario un programa de transformación de imagen.

Existen distintos programas que pueden cubrir esta necesidad como son "Microsoft Computer Vision API" [43] "Amazon Rekognition" [44] y "Google Cloud Vision API" [45] entre otros [46]. Sin embargo, el más utilizado para el reconocimiento y transformación de imágenes es "OpenCV", que tiene una adaptación a Python y su uso es sencillo, por lo que es el más idóneo para el desarrollo de este proyecto. En concreto, se utiliza la versión `opencv-contrib-python`, la cual añade funcionalidades extra que resultan necesarias para el funcionamiento.

Figura 7: Logo de "OpenCV" [47].

OpenCV (*Open Computer Vision*), cuyo logo se muestra en la figura 7, es una librería muy potente de código abierto desarrollada originalmente por Intel en 1999 que, hoy, sigue siendo la librería más popular de visión artificial. Generalmente se usa para diversas aplicaciones, como la detección de movimiento, el reconocimiento de objetos o la reconstrucción 3D a partir de imágenes, entre otros [47].

En cuanto a la modificación de imágenes, las opciones que ofrece son muy amplias como son:

- Alteración de colores. Usado para detección de objetos o para variar el histograma.
- Segmentación por umbral de la imagen. Usado para definir contornos.
- Transformación geométrica de la imagen. Usado para recortar, escalar, trasladar o cambiar la perspectiva de la imagen.
- Suavizado de imagen. Usado para reducir el ruido, presentado en forma de gránulos.
- Transformaciones morfológicas. Usado para estilizar o ensanchar contornos.
- Definir gradientes. Usado para detectar gradientes de color en la imagen.
- Definir bordes. Usado para detectar los posibles bordes del interior de una imagen.
- Detección de patrones. Usado para la detección de formas.
- Mezclar imágenes mediante el método piramidal. Usado para hacer transiciones.
- Generar histogramas. Usado para obtener información de una imagen [48], [49].

Entre las opciones listadas, en este proyecto sólo se van a realizar transformación de la imagen para adaptarla a las condiciones óptimas del OCR. Las técnicas usadas de OpenCV son las relacionadas con la alteración de colores, segmentación por umbral de la imagen, suavizado de la imagen y transformaciones morfológicas.

El realizar la transformación previamente podría conllevar mejoras en el reconocimiento, aunque algunas de ellas se realizan dentro del programa del OCR.

Cabe destacar que, como se puede apreciar, por todas las funcionalidades listadas que puede desempeñar el OpenCV, se podría crear un sistema OCR directamente usando este programa, sin embargo, a la hora de usar los patrones de reconocimiento sobre los que se comparará la imagen, los resultados que se obtengan serán peores que si se usa un sistema de OCR con unos patrones ya entrenados por una red neuronal, como es el caso de Tesseract OCR.

5.2.3 SISTEMA DE TRANSFORMACIÓN DE TEXTO A VOZ

Una vez obtenido el resultado proporcionado por el programa de OCR, se debe transformar a un archivo de audio, para poder reproducirlo, posteriormente, a través de un dispositivo de sonido. Para ello, se requiere un tipo de programa denominado “Text to speech”, abreviado como TTS, que se podría traducir de forma literal como “de texto a discurso”.

El funcionamiento de un sistema TTS es:

1. Se convierte a texto las posibles cifras, abreviaturas, siglas, etc.
2. Se sustituye cada palabra por su transcripción fonética.
3. Se segmenta el texto en unidades prosódicas (frases, acentos, entonaciones, etc.).
4. Por último, se sintetiza el texto transformado usando una voz sintética [50].

A la hora de determinar un sistema TTS adecuado para el proyecto, es necesario evaluar ciertas características:

- Tener una dicción correcta en español.
- Reproducir el sonido con una voz natural, evitando la voz robótica.
- Poder funcionar sin conexión a internet.
- Ser de libre uso.
- Realizar la transformación del texto a voz de forma rápida.

Actualmente, los programas TTS de libre uso más populares son “gTTS” [51], “Pico TTS” [52], “Festival TTS” [53], “eSpeak” [54] y “PyTTSx3” [55]. En la Tabla 4 se recoge una comparación de los programas TTS mencionados, tras haber realizado pruebas con cada uno de ellos para evaluar el más adecuado para el proyecto.

Tabla 4: Comparación de resultados entre los distintos TTS.

	gTTS	Pico TTS	Festival TTS	eSpeak	PyTTSx3
Dicción	Correcta	Correcta	Sin datos propios	Correcta	Incorrecta
Voz	Natural	Natural	Sin datos propios	Robótica	Robótica
Offline	No	Sí	Sí	Sí	Sí
Velocidad	Estándar	Estándar	Sin datos propios	Rápido	Rápido

Cabe destacar que al programa PyTTSx3 se le ha asignado dicción incorrecta debido a que en ocasiones no reproducía todo el texto a convertir, interrumpiendo la reproducción a mitad de la frase. En cuanto al programa Festival TTS, varias características se marcan como "Sin datos propios" debido a que el programa no contiene el idioma español, y es necesario instalarlo aparte. Sin embargo, los paquetes de voces en español son difíciles de encontrar a través del proveedor oficial, ya que la web de la fuente original no funciona correctamente.

Como se puede apreciar en la Tabla 4, el programa Pico TTS es el único programa que cumple todas las características que se buscan para el proyecto, por lo que se ha escogido Pico TTS frente a los demás.

Figura 8: Logo de la compañía SVOX [56].

Pico TTS es un programa de código abierto de síntesis de discurso creado por la compañía SVOX, cuyo logo se muestra en la figura 8. Desde 2009 ha sido implementado de forma integral en el sistema operativo "Android" [57] de Google [56].

5.2.4 SISTEMA DE REPRODUCCIÓN

Para reproducir el archivo de voz generado con el programa TTS seleccionado, es necesario un reproductor de audio. La tarjeta de sonido seleccionada permite distintos métodos para reproducirlo, en este caso, se han probado diferentes métodos.

El primer método consiste en reproducir el fichero de audio generado a través del terminal de comandos de la Raspberry. Para ello se utiliza el siguiente comando [58]

```
aplay -f cd -Dhw:1 output.wav
```

Donde "output.wav" es el archivo de voz que se quiere reproducir. Sin embargo, este método reproduce el archivo con acoplamiento y con muy mala calidad, haciendo que sea, en cierto modo, molesto cuando se reproduce.

Puesto que el método anterior no cumple con los requisitos que se esperan, se procede a probar con un reproductor que se pueda integrar mediante código de Python. Con este requisito, es posible realizar la reproducción de sonido con Python mediante diversos programas [59]de los cuales se experimenta en este proyecto con:

- Playsound [60].
- Simpleaudio [61].
- Winsound [62].

Se realizaron pruebas con todos ellos, sin embargo, Playsound no reproduce el audio de forma adecuada y Simpleaudio produce problemas con el formato del fichero de audio. Por último, Winsound reproduce el audio, pero se sigue produciendo problemas de acoplamiento. Dado que los acoplamientos se producen usando dos métodos distintos, se procede a comprobar que el audio a reproducir no es el causante de este error. Para ello, se ha hecho uso del reproductor externo "VLC media player" [63] el cual se encuentra instalado de forma predeterminada en el sistema operativo "Linux" [64] "Raspbian" [65] de la Raspberry.

Sin embargo, haciendo uso del reproductor VLC media player, la calidad del audio se ve mejorada y se deja de detectar el acoplamiento, reproduciendo el audio de forma clara. Por esta razón y con el fin de integrar todo en un único formato de programación, se ha tratado de encontrar e instalar versión para Python del reproductor VLC media player. No obstante, se vuelve a producir el acoplamiento al reproducir el audio mediante este método. Por tanto, se ha decidido usar el reproductor VLC media player de forma externa, mediante la programación de su ejecución, mediante el terminal de Raspbian.

5.2.5 OTROS PROGRAMAS UTILIZADOS

Además de los programas utilizados, mencionados anteriormente, se ha hecho uso de otros programas, como "Thonny Python IDE" [66] y "AlsaMixer" [67].

Thonny Python IDE es un editor de código de Python que se utiliza para programar y depurar el código realizado durante el proyecto. Se ha seleccionado este programa, frente a otros similares, porque cuenta con un interfaz simple e intuitiva, con todas las herramientas básicas y capaz de realizar depuración, que ayuda a descubrir los posibles fallos que pueda tener un código. Además, se encuentra instalado de forma predeterminada en el sistema operativo Raspbian (ver Anexo III) [66].

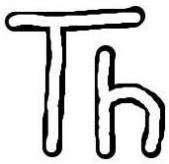


Figura 9: Logo del Thonny Python IDE [68].

Thonny Python IDE, cuyo logo se muestra en la figura 9, usa la versión 3.7 de Python, por lo que todo el código y las instalaciones que se hagan tienen que ser compatibles con dicha versión. Esto es importante tenerlo en cuenta a la hora de instalar los programas necesarios para el proyecto.

Otro de los programas que se ha utilizado es ALSAMixer. Se trata de un mezclador de audio creado para la arquitectura ALSA (*Advanced Linux Sound Architecture*) de Linux y fue usado en este proyecto para configurar todas las salidas de audio del dispositivo final, tanto los auriculares, como los altavoces, nivelando el volumen de cada una de ellas. Esta configuración es importante para el correcto funcionamiento del dispositivo y no es posible realizarla de forma automática, requiriendo seguir unos pasos específicos que serán descritos en la parte de implementación (Anexo III.II.VII).

5.2.6 LISTA DE LOS PROGRAMAS SELECCIONADOS

A modo de resumen, y teniendo en cuenta lo expuesto anteriormente, a continuación, se enumeran los programas seleccionados que se utilizan en el ámbito de este proyecto:

- Tesseract OCR.
- OpenCV.
- Pico TTS.
- VLC Media Player.
- Thonny Python IDE.
- ALSA Mixer.

5.3 TEORÍA DE FUNCIONAMIENTO

En este apartado se describe brevemente el proceso de funcionamiento teórico del dispositivo con el fin de esquematizar el proyecto, así como entender la funcionalidad de cada parte de este trabajo. Esto se realiza mediante un diagrama de flujo y una explicación más detallada del mismo.

5.3.1 DIAGRAMA DE FLUJO

El diagrama de flujo mostrado en la figura 10 ilustra el funcionamiento y uso del dispositivo. En la sección 5.3.2 se describe cada una de las partes que lo compone. Los diagramas de flujo de menor tamaño que aparecen recuadrados con una línea discontinua muestran una expansión con mayor detalle del bloque que se señala.

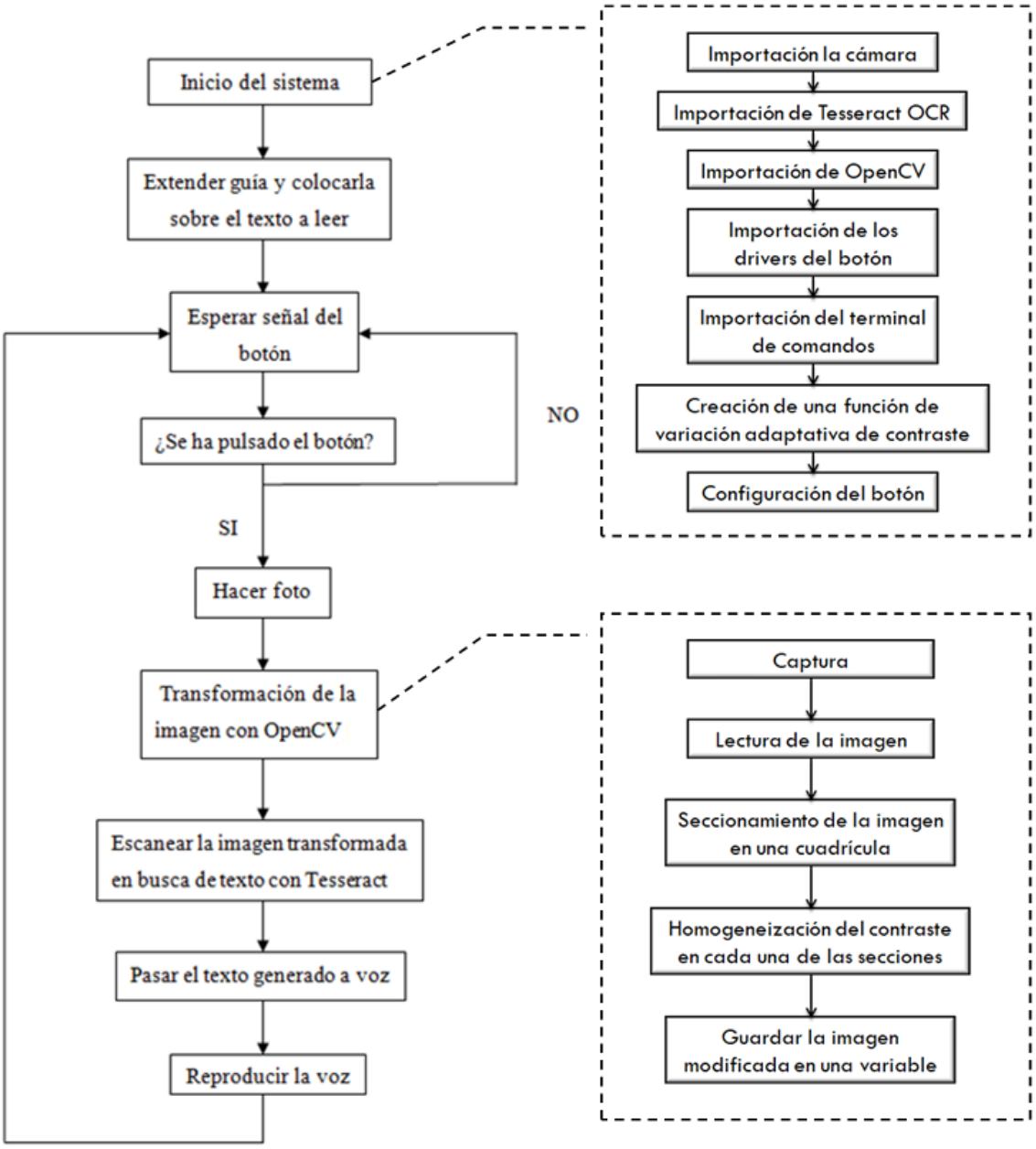


Figura 10: Diagrama de flujo del funcionamiento.

5.3.2 DESCRIPCIÓN TEÓRICA DEL FUNCIONAMIENTO

A partir del diagrama de flujo mostrado, se procede a explicar de forma detallada cada uno de los bloques. Siguiendo la secuencia del diagrama de flujo, el procedimiento de funcionamiento del dispositivo es el siguiente:

1. *Inicio del sistema*. El dispositivo se enciende mediante el interruptor situado en el conector de la pila.
2. *Extender guía y colocarla*. Se despliega la barra extensible, que incorpora el dispositivo, hasta su máxima longitud de forma manual. Se debe apoyar su extremo sobre el texto y se extienden sus dos solapas laterales, con el fin de aplanar el documento a leer.
3. *Esperar señal del botón* y *hacer foto*. Mediante el botón situado en la placa de sonido, se acciona la cámara, que realiza una foto de la página.
4. *Transformación de la imagen*. La foto se procesa a través del software de código libre OpenCV, alterando la imagen de la forma que sea óptima para el OCR. Las principales transformaciones que se realizan son variación de contraste, variación de DPI (*dots per inch*), reducción de ruido o binarización entre otras, así como la combinación de cada una de ellas.
5. *Buscar texto en la imagen*. Analiza el texto con el programa Tesseract OCR, mediante su configuración óptima, creando una variable que contiene el texto reconocido.
6. *Pasar el texto generado a voz*. El texto almacenado en la variable se convierte en un fichero de audio usando el programa Pico TTS.
7. *Reproducir la voz*. El archivo de audio generado se reproduce a través del altavoz, de forma predeterminada, o a través de la conexión de auriculares si se encuentran conectados, usando el reproductor VLC media player.
8. *Fin de reproducción y vuelta*. Una vez que concluye la reproducción del audio, el dispositivo se queda a la espera de una nueva pulsación del botón, que inicia de nuevo el proceso desde el punto 3 (*Esperar señal del botón)*.
9. *Apagado del sistema*. En el caso de querer apagar el dispositivo, simplemente se debe cambiar la posición del interruptor de encendido.

6. IMPLEMENTACIÓN

En este apartado se describe el proceso de implementación de todas las funcionalidades, seguido de la descripción del montaje del dispositivo, junto con el diseño e instalación.

6.1 FUNCIONALIDADES MEDIANTE CÓDIGO

Todas las funcionalidades del dispositivo están implementadas dentro de un único código de Python, que se ejecuta al encender el dispositivo. A continuación, se describen las partes más importantes del código, sin embargo, el código completo se encuentra en el Anexo IV.

En primer lugar, dentro del marco de los pasos 1 "*Inicio del sistema*" y 2 "*Extender guía y colocarla"* del apartado 5.3.2, una vez que se arranca el sistema operativo, se deben importar todas las librerías que se van a utilizar. A continuación, se realiza la asignación del botón y la carga de la configuración del ALSA Mixer definida previamente, la cual es importante realizar debido a que el ALSA Mixer vuelve a su configuración por defecto cada vez que se apaga el dispositivo [69].

Las librerías se importan al código mediante uno de los dos tipos de comandos descritos a continuación; el primero importa la librería completa, mientras que el segundo importa únicamente una función determinada de la librería.

```
import NombreDePrograma
from NombreDePrograma import NombreDeFuncionalidad
```

La carga de la configuración del ALSA Mixer se realiza a través del siguiente comando.

```
os.system('sudo alsactl restore')
```

En segundo lugar, tras cargar las librerías y la configuración inicial, se debe ejecutar una sucesión de comandos de forma indefinida, hasta que se apague el producto, haciendo uso de un bucle "while" infinito. Dentro de los comandos que se deben ejecutar dentro del bucle están los relacionados con el funcionamiento de interés del dispositivo. Esto se hace de esta manera para asegurarnos de ejecutar todos los comandos más significativos, a la vez de evitar que el dispositivo vuelva a importar todas las librerías de nuevo al terminar el código, requiriendo más tiempo de forma innecesaria.

En el bucle "while" general descrito y correspondiendo con el paso 3 *"Esperar señal del botón* y *hacer foto"*, primero se notifica con un pitido, se inicializa la cámara y se espera a la señal del botón. El programa no avanza hasta que el botón sea pulsado, lo cual marca el inicio del proceso de captura del texto. Esto se hace mediante el siguiente código.

```
i=0 # Aplica la condición del while para que sólo salga cuando haya pulsación de botón.
while (i == 0):
    boton = GPIO.input(17) # Lee el botón.
    if boton: # Si no se produce pulsación del botón.
        sleep(0.001) # Reposa durante un milisegundo para evitar abrumar el procesador.
        continue # Continúa el bucle "while".
    else: # Si se produce pulsación del botón.
        sleep(2) # Espera dos segundos para que la cámara pueda ajustarse.
        camera.capture('/home/pi/Desktop/TFG/ImagenesProyecto/image1.jpg') # Hace una captura.
        i=1 # Sale del bucle "while".
```

Posteriormente, una vez se ha tomado la captura, se emite un sonido, la cámara se apaga y se inicia el proceso de modificación de la imagen, correspondiente al punto 4 "*Transformación de la imagen*". En concreto, dicha modificación consiste en la variación del contraste de forma adaptativa (CLAHE) que se realiza a través de los siguientes comandos [70].

```
img = cv2.imread(file_path, 0) # Carga la imagen en gris.
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8)) # Crea el contraste adaptativo.
cl1 = clahe.apply(img) # Aplica el contraste.
```

La variación del contraste, en concreto, divide la imagen en secciones y varía el contraste de cada una de ellas, lo cual lo hace mucho más efectivo que una variación de contraste fijo a toda la imagen, ya que este último usa todo el histograma de la imagen como punto de referencia y la equilibra en consecuencia, mientras que el contraste adaptativo usa el histograma discretizado de cada sección, variando cada una de forma independiente. Con este paso se consiguen nivelar los posibles gradientes de luz que pueda tener la captura tomada [70].

Continuando con el paso 5 "*Buscar texto en la imagen"*, se ejecuta el programa OCR sobre la imagen modificada, creando una variable que contiene el texto reconocido, y se utiliza mediante el siguiente comando.

```
original=pytesseract.image_to_string(img, config='-l spa --oem 3 --psm 1')
```

Como se puede observar, dentro del paréntesis del comando descrito, aparecen varias configuraciones. Estas configuraciones se explican con más detalle en el apartado 6.7 "Configuración de Tesseract".

A continuación, siguiendo con el paso 6 "*Pasar el texto generado a voz"*, se debe aplicar el TTS sobre la variable que contiene el resultado del reconocimiento realizado por el OCR. Esto se hace mediante el siguiente comando [71].

```
os.system('pico2wave -l es-ES -w grabacion.wav "%s"' %original)
```

El programa TTS usado, PICO TTS, se usa mediante el terminal a través del comando de Python "os.system". En esta función se indica el nombre del programa junto con una serie de parámetros, siendo la variable de entrada "original", que debe tener un "%" delante. Además, el comando se configura para que sintetice el texto usando la dicción en español ("es-ES" en el código) y crea un archivo de sonido ".wav".

Posteriormente, el archivo ".wav" generado se reproduce a través del VLC media player, con un comando de terminal, de la misma forma que hemos hecho antes, cumpliendo así el paso 7 "*Reproducir la voz"*.

```
os.system('cvlc --play-and-exit /home/pi/grabacion.wav') # Reproduce el audio.
```

La configuración de "CVLC" permite abrir el programa sin mostrar las opciones gráficas y la opción "play and exit" realiza la reproducción y cierra el programa una vez haya finalizado su reproducción.

Por último, según el paso 8 "*Fin de reproducción y vuelta"*, al no haber más instrucciones, el código vuelve al inicio del bucle infinito, ejecutando de nuevo las instrucciones descritas hasta que se apague el dispositivo.

6.2 EQUIPAMIENTO ADICIONAL

Para ser capaces de ensamblar todos los componentes, interactuar con la Raspberry Pi Zero, desarrollar el código e imprimir las piezas de la cubierta exterior, ha sido necesario disponer de cierto equipamiento adicional. En este apartado se muestran todo el equipamiento adicional necesario, junto con una breve descripción de su uso en el proyecto.

El equipamiento adicional necesario se enumera a continuación.

- Pantalla con conexión HDMI: Usada para la visualización del sistema operativo y poder programar directamente en él.
- Cable de HDMI a mini-HDMI: Usado para conectar la Raspberry a la pantalla.
- Adaptador de USB a micro-USB: Usado para conectar periféricos a la Raspberry Pi Zero, que sólo tiene conexiones tipo micro-USB.
- Teclado y ratón con conexión USB: Usados para poder interactuar con la Raspberry.
- Cable y cargador de 1 A Y 5 V con conexión micro-USB: Usados para hacer conexión entre el puerto de alimentación de la Raspberry y la corriente.
- Soldador: Usado para calentar el estaño y poder realizar las soldaduras necesarias.
- Estaño para soldar: Usado como material de adicción a la soldadura.
- Pasta para soldar: Usada para reducir el punto de fusión del estaño y para ayudar a su agarre en el punto determinado que se quiera soldar.
- Cable: Usado para poder conectar los distintos componentes entre sí.
- Impresora 3D: Usada para imprimir todas las piezas de la cubierta exterior.

A la hora de realizar el proyecto, ha sido necesario soldar ciertos componentes, por lo que también es recomendable tener experiencia en este ámbito. Además, es recomendable disponer del siguiente equipamiento complementario para el proceso de soldadura.

- Extractor de humos: Para extraer los gases tóxicos liberados durante la soldadura.
- Agarres tipo cocodrilo: Para mantener sujetos los objetos que se van a soldar entre sí.

6.3 ENSAMBLAJE DE LOS COMPONENTES

Con el fin de ilustrar un correcto ensamblaje del dispositivo, en este apartado se describe, de forma detallada, el proceso de montaje de cada componente apoyado por ilustraciones e incluyendo, al final del apartado, un esquema general de cómo debería quedar si se han seguido los pasos correctamente. Además, en el Anexo I se detallan todos los componentes que se deben ensamblar.

En primer lugar, se debe soldar el conector macho de 40 pines a los pines de la Raspberry. Hay que tener en cuenta, que se debe soldar por el extremo corto de los pines y dirigiendo los pines largos hacia arriba siendo esto especialmente importante, ya que es necesario conectar una tarjeta HAT a ellos y, por tanto, se debe garantizar que el orden de pines a los que se conecte sea el apropiado. El resultado debería quedar como la figura 11.

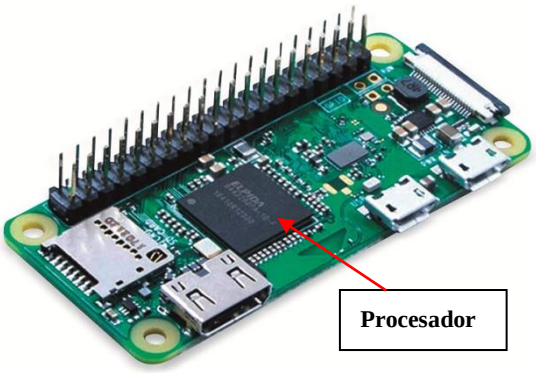


Figura 11: Raspberry con los pines soldados.

También es recomendable incorporar un disipador de calor al procesador, para ello se debe despegar el adhesivo y se colocar el disipador sobre el procesador. En la figura 11 se muestra la localización del procesador sin disipador.

A continuación, se debe conectar la cámara. Para ello, hay que retraer ligeramente la pestaña negra del conector e insertar el extremo del flex de la cámara con la parte dorada hacia abajo. Una vez hecho esto, se debe volver a poner la pestaña negra en su sitio, fijando y asegurando la cámara al conector para evitar que ésta se desconecte. Se muestra el proceso en la figura 12.

Figura 12: Proceso de conexión de la cámara a la Raspberry.

A continuación, se conecta la tarjeta de sonido HAT mediante su conexión de pines hembra, señalados en la figura 13, a los pines que se han soldado antes en la Raspberry. Debe quedar con un aspecto como el que se muestra en la figura 14.

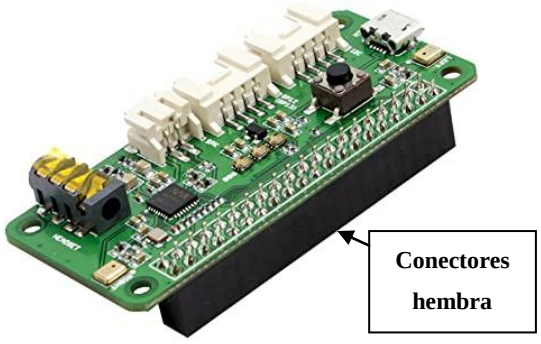


Figura 13: Imagen de la tarjeta de sonido mostrando la localización de los pines hembra.

Figura 14: Tarjeta de sonido conectada a la Raspberry [72].

Como el altavoz usado, mostrado en la figura 15, tiene un puerto de conexión más pequeño que el que está instalado en la tarjeta de sonido, se ha decidido cortar el conector y soldarlo a los pines de la tarjeta de sonido. Para ello, se han pelado los cables y se han soldado los extremos a los pines correspondientes al conector de altavoces. Cabe destacar que se debe soldar el cable rojo al pin de la izquierda (A) y el negro al de la derecha (B), como se muestra en la figura 16, con el fin de mantener la polaridad correcta.

Figura 15: Altavoz escogido [73].

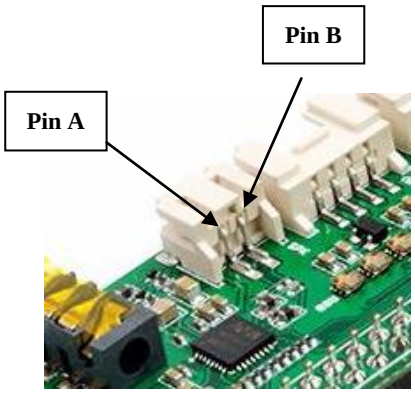


Figura 16: Tarjeta de sonido conectada a la Raspberry.

Para realizar la alimentación de la Raspberry, se van a soldar dos cables entre el portapilas y el regulador DC-DC. Para ello, se deben cortar dos cables, uno de 5 cm y otro 7 cm de longitud, y pelar sus extremos. El cable de 5cm conecta entre el contacto negativo del soporte de la pila de botón y el contacto de toma de tierra (GND) del regulador a 5V. Con el cable de 7cm se suelda a la toma positiva del soporte de la pila de botón y el otro extremo a la toma de voltaje de entrada del regulador a 5V (VIN). En la figura 17 se muestra un diagrama de la conexión que se ha descrito.

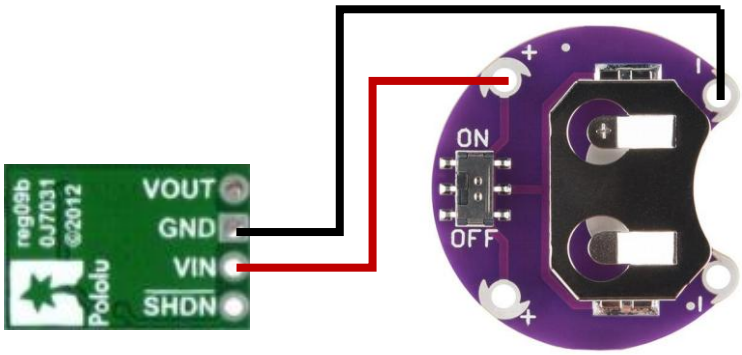


Figura 17: Conexión entre el soporte de pila y el regulador a 5V.

A continuación, se debe conectar el regulador DC-DC con la Raspberry. Para ello, se necesitan dos cables de 7 cm cada uno. El extremo de un cable va soldado a la conexión de tierra (GND) y el extremo de otro cable a la conexión de voltaje de salida (VOUT) del regulador a 5V.

Por último, se debe soldar el extremo del cable conectado a tierra (GND) del regulador a la conexión a tierra de la Raspberry, indicada como "D" en la figura 18; y el extremo del cable conectado a la conexión de voltaje de salida (VOUT) a la conexión de corriente de la Raspberry, señalada como "C" en la figura 18.

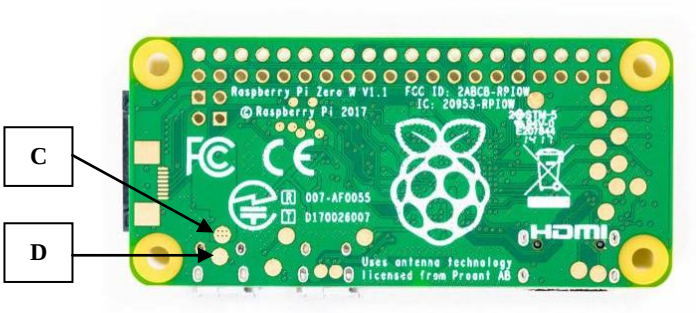


Figura 18: Parte trasera de una Raspberry Pi Zero con sus conexiones de corriente señaladas.

Por último, en la figura 19 se muestra el diagrama del montaje final del dispositivo.

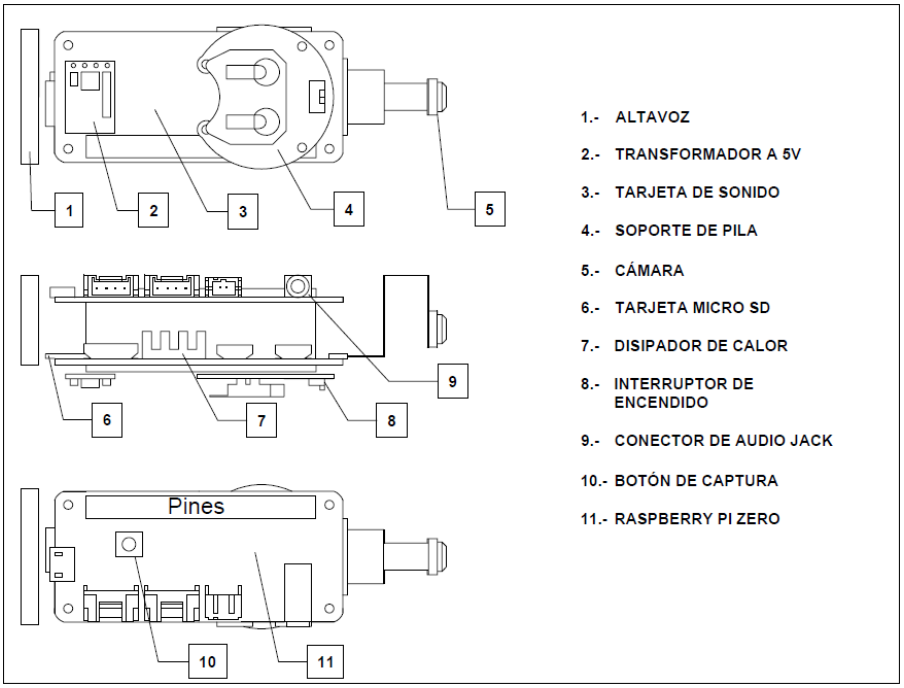


Figura 19: Esquema completo de los componentes ensamblados.

6.4 DESARROLLO DE LA CUBIERTA EXTERIOR

Para hacer el dispositivo fácil y seguro de manejar, es necesario fabricar una carcasa que pueda mantener todos los componentes juntos, fijos y protegidos. A la hora de diseñar dicha carcasa, se ha usado el programa "Autocad" [74] en su funcionalidad de 3D. En este apartado se describen cada una de las siete partes diseñadas con sus características particulares de diseño.

6.4.1 PLANO 1. PARTE SUPERIOR DE LA CARCASA

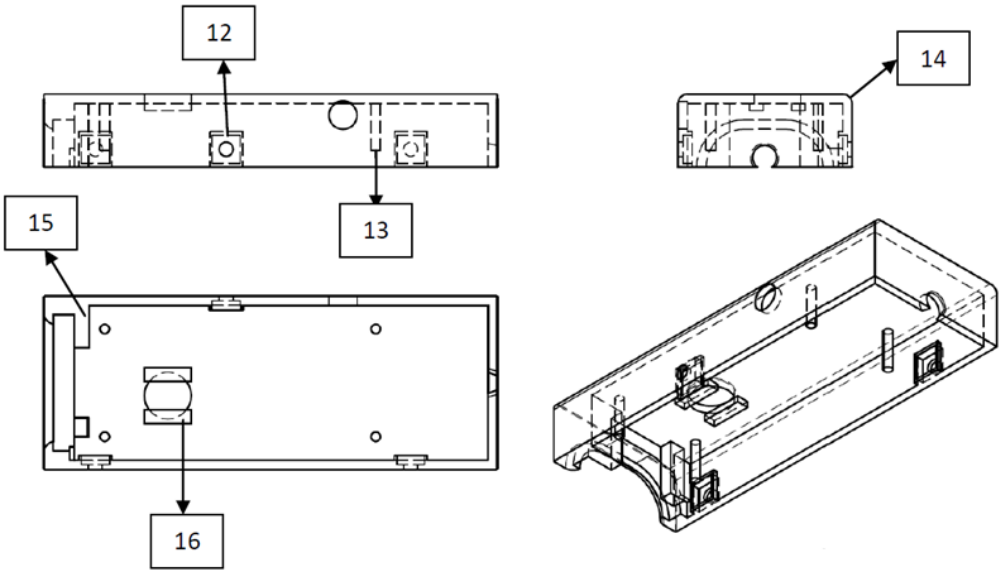


Figura 20: Plano de la carcasa superior.

Este diseño es la parte superior de la carcasa. Cuenta con orificios para todos los dispositivos de entrada/salida necesarios encontrados en la mitad superior del dispositivo (cámara, altavoz y entrada de auriculares). Además, cuenta con un hueco adicional en la parte superior que está diseñado para colocar el botón.

Como se puede observar en el punto 12 de la figura 20, hay tres huecos en forma de cuadrado en el interior de la carcasa. Estos forman parte del método de unión entre la carcasa superior e inferior, que se realiza mediante tres uniones roscadas formadas entre un tornillo corto y una rosca cuadrada, que se acopla a dichos espacios. El tornillo es pasante hasta pasar los orificios de la carcasa inferior, punto 17 de la figura 21, con el objetivo de asegurar la pieza.

El diseño cuenta con unos salientes cilíndricos, con el punto 13, diseñados para introducirlos a través de los orificios de fijación de la Raspberry y de la tarjeta de sonido HAT con el objetivo de fijar estos componentes a la carcasa y evitar que se desplacen con el uso. También tiene los bordes redondeados, indicados con el punto 14, para mejorar la comodidad en el agarre.

Por otro lado, se han incorporado unos salientes, marcados con el punto 15, colocados en lo que sería la zona trasera del altavoz con el fin de evitar que éste se introduzca hacia el interior del dispositivo una vez que se haya instalado. Estos salientes sólo se encuentran en la carcasa superior, y no en la inferior, para que el altavoz se pueda introducir desde abajo durante el montaje, ya que si estuviesen en ambas carcasas el montaje sería más complicado.

Por último, tiene unos pequeños salientes en la zona del botón, señalado con el punto 16, cuya función es ejercer de topes laterales para el dispositivo de fijación del botón, como se verá más adelante en la figura 23, y evitar que el botón rote.

6.4.2 PLANO 2. PARTE INFERIOR DE LA CARCASA

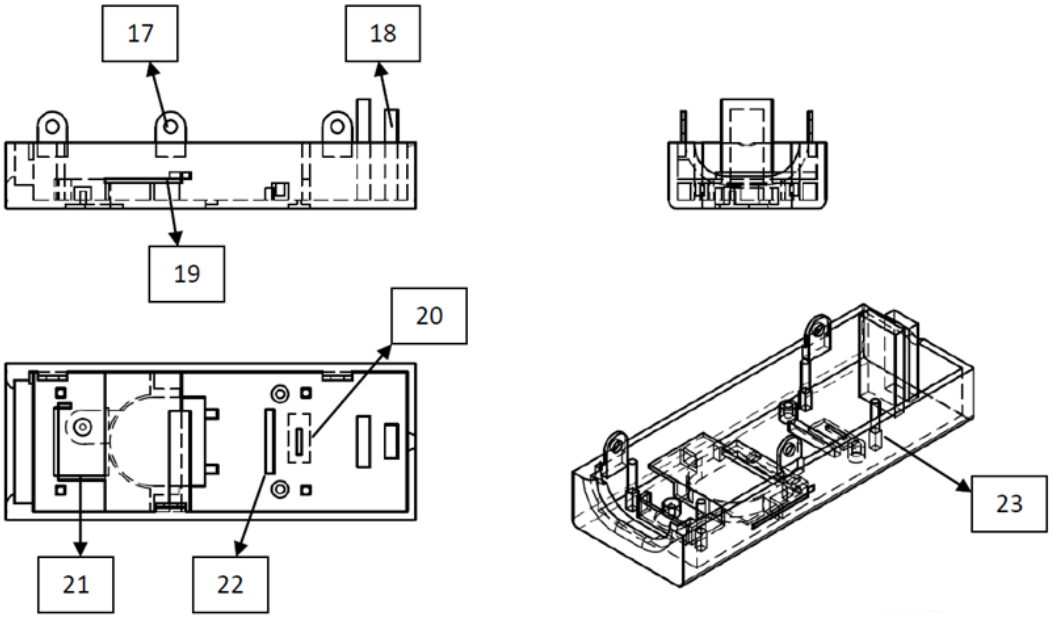


Figura 21: Plano de la carcasa inferior.

El diseño de la figura 21 representa la parte inferior de la carcasa que, como se puede observar, es bastante más compleja que la parte superior, ya que se encarga de fijar y distribuir todos los componentes que se sitúan en la parte inferior de la Raspberry.

Los salientes marcados con 17, que sobresalen desde los laterales de la carcasa y que tienen un agujero pasante, están diseñados para coincidir con los orificios, marcados con 12, de la carcasa superior y tienen la función, de recibir el tornillo pasante y realizar el cierre del dispositivo.

Las prominencias marcadas con 18 alojan el cable flex de la cámara y tienen la función de ejercer de superficie donde se colocarán las partes adhesivas que contiene el flex y ayudar a dirigirlo durante el montaje, al mismo tiempo que se mantiene fijo y ocupa el menor espacio posible. Además, ejerce cierta presión sobre la cámara con el fin de mantenerla posicionada correctamente en el dispositivo final.

Por otro lado, tiene un pequeño embellecedor en la zona del compartimento de la pila, marcado con 19, que ayuda a que no se vea el interior electrónico. Si se retira la tapa del compartimento, para sustituir la pila, este compartimento ayuda a dirigir la pila hacia la ranura preparada para la ella. Además, ayuda a mantener el soporte de la pila en su lugar ejerciendo de tope en su parte superior.

La rendija marcada con 20 está diseñada para introducir la prominencia de la deslizadera que engancha con el interruptor del soporte de la pila. Cabe destacar que debe tener el tamaño justo para tener en cuenta el movimiento de la deslizadera sin pasarse y reducir sus grados de libertad al mínimo. En esta misma zona hay un pequeño hueco no pasante de forma rectangular más grande que la rendija que está diseñado para introducir el cuerpo de la deslizadera en él, evitando así que ésta sobresalga demasiado y reduciendo la posibilidad de que la prominencia de la deslizadera se pueda partir por algún incidente.

El punto marcado con 21 está diseñado como compartimento para el regulador de voltaje y evitar así que se mueva de forma lateral. Para evitar que oscile verticalmente es necesario poner cinta aislante en la parte superior después de haber acoplado el regulador durante el montaje. Se ha evitado incluir otro saliente que restrinja el movimiento vertical para evitar que el montaje se vuelva más complejo. Cabe destacar que, al ser un diseño inicial, es aconsejable tener algo de libertad en el posicionamiento de los componentes.

La prominencia marcada con 22 sirve como tope para evitar que la pila se introduzca más de la cuenta, evitando que sea difícil de extraer la pila *a posteriori*.

Como se puede observar en la figura 21, las prominencias marcadas con 23 de la carcasa inferior tienen una diferencia con respeto a la carcasa superior, marcadas con 13, que consiste en que, para la carcasa inferior, la base de las prominencias es cuadrada con el objetivo de ejercer como tope para los orificios de las tarjetas, de la Raspberry y de sonido, y evitar que las éstas se introduzcan más de la cuenta en el dispositivo, permitiendo mantener el suficiente espacio para alojar el resto de los componentes inferiores.

6.4.3 PLANO 3. PARTE SUPERIOR DEL BOTÓN

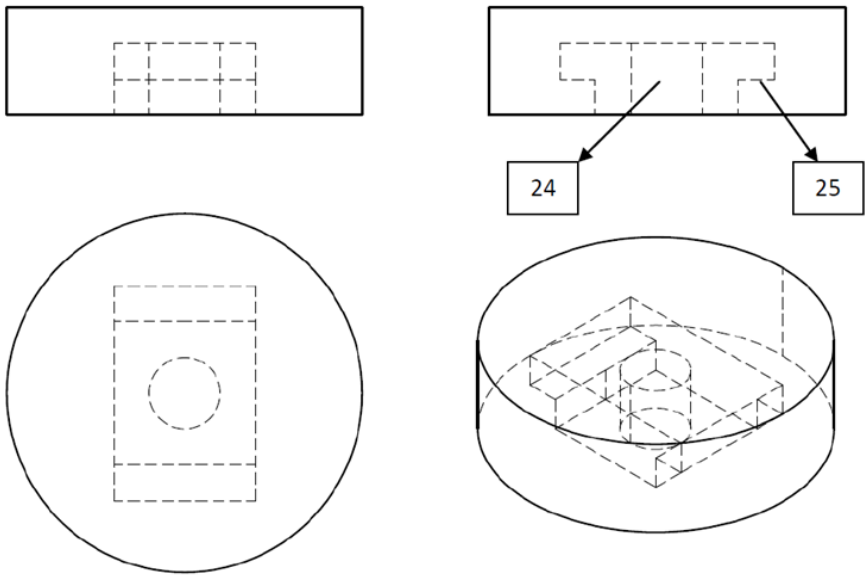


Figura 22: Plano de la parte superior del botón.

La figura 22 representa la parte superior de las dos partes que componen el botón. Esta parte va a estar en contacto con el exterior, sobresaliendo ligeramente del nivel de la carcasa para ayudar a una persona ciega a localizarlo. Tiene ciertas particularidades, como el hueco que presenta la pieza, diseñado para poder introducir por él las pestañas de la parte inferior, marcadas con 26 en la figura 23. En el hueco, tiene una prominencia interna, marcada con 24, extendiéndose desde la parte superior interna del botón, cuyo objetivo es presionar el botón electrónico, localizado en la tarjeta de sonido, cuando la pieza del botón sea pulsada, ya que es el encargado de iniciar todo el proceso de reproducción del texto.

También cuenta con unos espacios internos en los laterales marcados con 25 diseñados para alojar las pestañas, marcadas con 26 en la figura 23, correspondientes a la pieza de fijación del botón.

6.4.4 PLANO 4. PARTE INFERIOR DEL BOTÓN

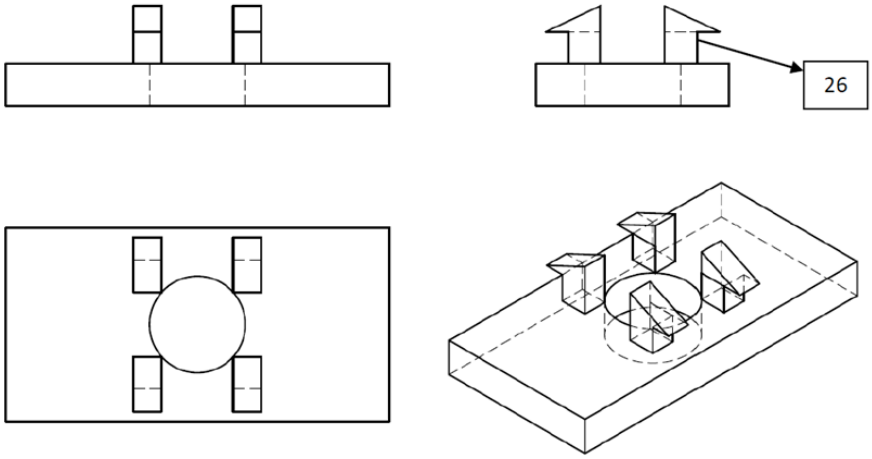


Figura 23: Plano de la parte inferior del botón.

La figura 23 representa la parte inferior del botón, que realiza la fijación y evita que el botón se desprenda cuando el dispositivo se coloque boca abajo. Cuenta con un cuerpo de forma rectangular que se encaja entre los salientes de la carcasa superior, marcados con 16, para evitar que el botón rote.

Como se puede observar, presenta unos salientes o pestañas que sobresalen hacia arriba y tienen un triángulo al final. El objetivo de estos triángulos es hacer que la pieza pueda deslizarse hacia el interior de la pieza superior del botón, pero, una vez introducida por completo, evitar que la pieza pueda salirse. Estos salientes deben tener menor altura que el espacio que se ha proporcionado, en el punto marcado con 25 en la parte superior, con el fin de permitir que exista el juego suficiente para el presionado del botón.

Por último, cuenta con un hueco pasante en el centro de la pieza diseñado para permitir el paso de la prominencia interna, marcada con 24, de la parte superior del botón.

6.4.5 PLANO 5. DESLIZADERA

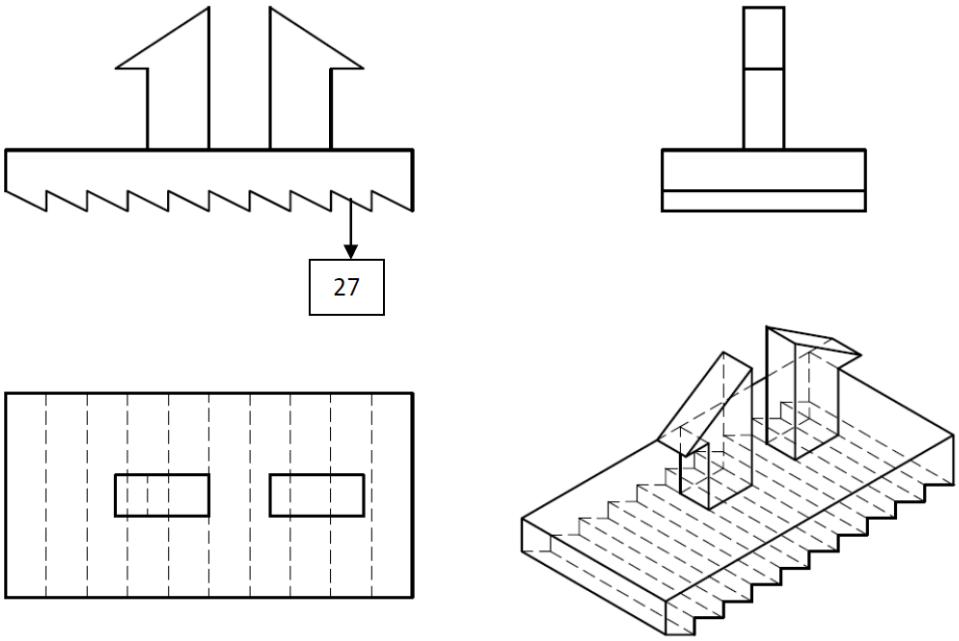


Figura 24: Plano de la deslizadera.

La figura 24 representa la deslizadera, que se trata de una parte externa y tiene como objetivo realizar el accionado de encendido y apagado del dispositivo. Sobresale ligeramente del nivel de la carcasa, como la parte superior del botón, para ayudar a su localización para una persona ciega.

El cuerpo tiene una forma rectangular, coincidiendo con el espacio marcado con 20 dejado en la carcasa inferior, para reducir los grados de libertad al mínimo y hacer que sólo pueda deslizarse en una dirección. Además, se ha diseñado con rugosidad, en este caso estriada, como se muestra en el punto 27, ya que ayuda a su localización y reconocimiento para una persona ciega.

Esta pieza también tiene unos salientes o pestañas con un triángulo al final. El objetivo de estos triángulos, de nuevo, es hacer que la pieza pueda deslizarse evitando que se pueda salir una vez introducido. La pieza se introduce a través del hueco presente en el centro de la pieza de fijación de la deslizadera, descrita más adelante en la figura 25. El espacio que hay entre los dos salientes está diseñado para enganchar el interruptor del soporte de la pila de botón.

6.4.6 PLANO 6. PIEZA DE FIJACIÓN DE LA DESLIZADERA

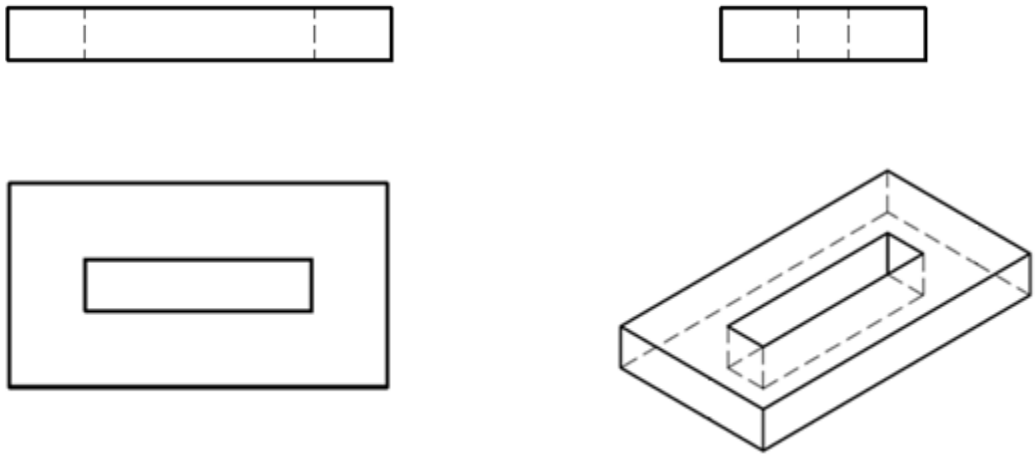


Figura 25: Plano de la pieza de fijación de la deslizadera.

La figura 25 representa la pieza de fijación de la deslizadera que sirve para fijar la deslizadera y evitar que pueda desprenderse del dispositivo. La forma de esta pieza no es demasiado importante, en este caso el cuerpo se ha realizado de forma rectangular para que ocupe el menor espacio posible mientras hace su función, pero podría tener otra forma si fuera necesario.

Sin embargo, el espacio rectangular del centro de la figura requiere que sus dimensiones se adecuen para que los salientes de la deslizadera se puedan introducir mediante apriete, pero que no se salgan a posteriori.

Esta pieza se mueve junto con la deslizadera como su fuesen una pieza sola, la única diferencia es que la deslizadera lo hace por la superficie exterior de la carcasa inferior y esta pieza lo hace sobre la superficie interior.

6.4.7 PLANO 7. COBERTURA DEL COMPARTIMENTO DE LA PILA

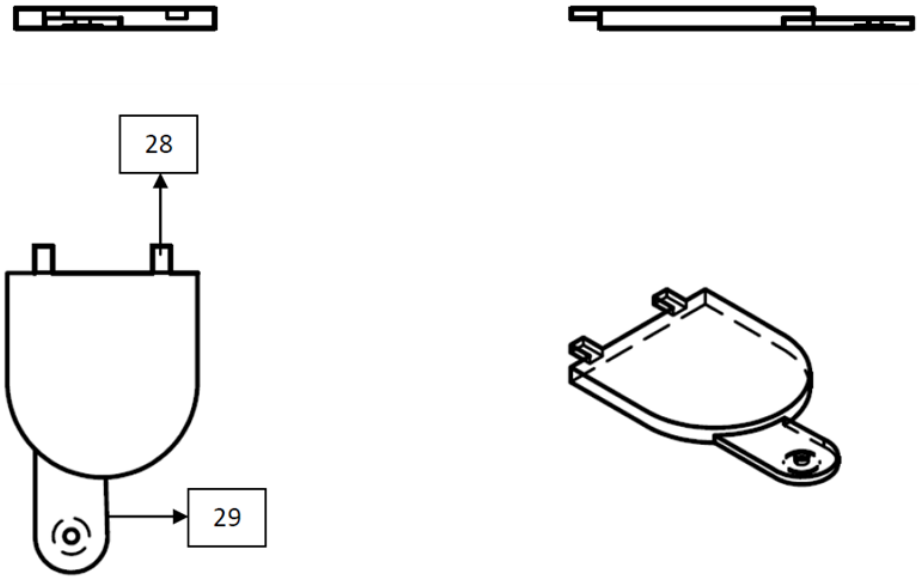


Figura 26: Plano de la tapa de cobertura del compartimento de la pila de botón.

Por otro lado, la figura 26 representa la tapa para el compartimento de la pila de botón. Tiene únicamente un fin estético, aunque podría ayudar a evitar la entrada de polvo al dispositivo.

Cuenta con un par de lengüetas, indicadas con 28, que sirven para guiar la tapa y fijarla cuando está colocada, haciendo algo de presión sobre la carcasa inferior. También tiene una prominencia en su longitud con un orificio al final, indicada con 29, que sirve para colocar un tornillo y fijarla a la carcasa inferior. Además, esta prominencia está desviada de forma intencionada del centro para evitar que el tornillo entre en conflicto con el regulador de voltaje, que se encuentra localizado inmediatamente debajo.

El orificio de sujeción, indicado en 29, tiene a su alrededor una disminución de espesor con un perímetro circular diseñada para albergar la cabeza del tornillo una vez colocado y evitar que sobresalga. Esto hace que el dispositivo sea más agradable al tacto, ya que la cabeza del tornillo no se notaría apenas al pasar la mano sobre él, mientras sigue siendo lo suficientemente reconocible para una persona ciega.

6.5 DESARROLLO DE LA VARILLA EXTENSIBLE

Con el objetivo de hacer que el manejo del dispositivo sea fácil para una persona ciega, es conveniente diseñar una varilla extensible que pueda posarse sobre la página del texto a leer, colocando el dispositivo a una distancia óptima de captura y condicionando la página para que presente las menores irregularidades posibles. En este apartado se describe cada parte de su diseño.

6.5.1 PLANO 8. CABEZAL DE LA VARILLA EXTENSIBLE

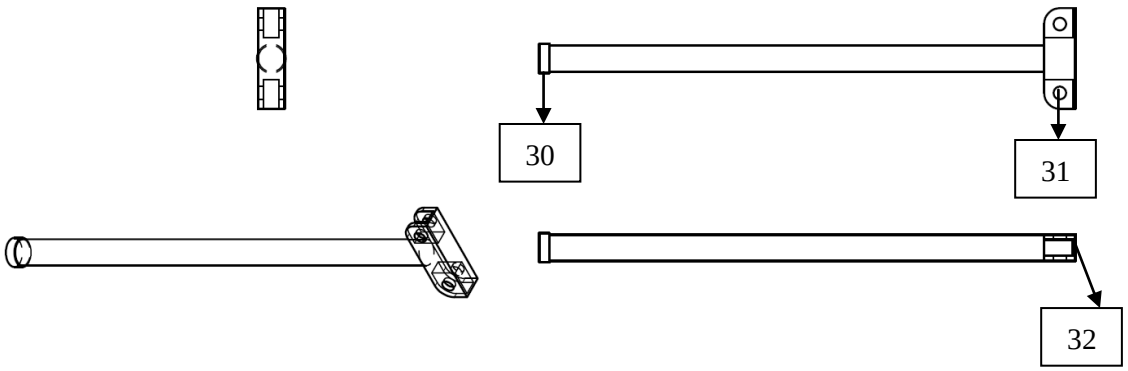


Figura 27: Plano del cabezal de la varilla extensible.

La figura 27 representa el cabezal, o parte límite, de lo que es la varilla extensible. Su función consiste en sostener los dos brazos laterales de la varilla, que están atornillados al cuerpo principal de la varilla. Su extremo liso se posaría sobre el papel que se quiere leer.

Posee un saliente cilíndrico en uno de sus extremos marcado con 30 que sirve de tope para la pieza anterior, que se coloca dentro de ésta en el diseño final. Además, cuenta con unos espacios en los lados del cabezal diseñados para albergar en ellos los extremos de los brazos que se atornillan a él mediante los orificios que se pueden observar en el dibujo en la marca 31, junto con un límite plano perpendicular a la varilla, señalado con 32, cuya función consiste en ejercer de tope para el despliegue de los brazos, ayudando, así a una persona ciega a desplegarlos correctamente.

6.5.2 PLANO 9. EXTENSIÓN DE LA VARILLA EXTENSIBLE

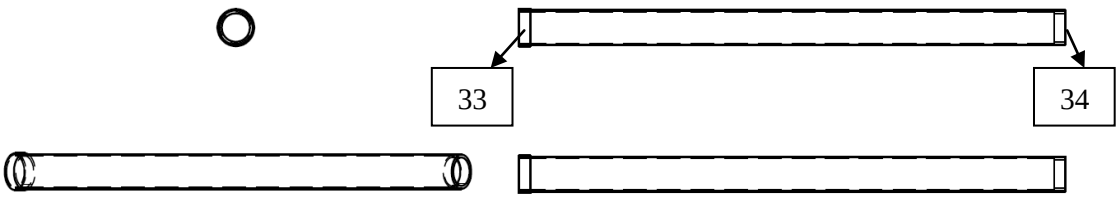


Figura 28: Plano una de las extensiones de la varilla extensible.

La figura 28 muestra una de las tres extensiones que se usan en la varilla extensible. Estas se introducen unas en las otras, teniendo la última el cabezal en su interior, y cuya función es extender la varilla a medida que se vayan desplegando.

Para conseguir el desplegado, el estrechamiento, marcado con 34, del extremo derecho de la extensión ejerce de tope para el saliente cilíndrico, marcado con 33, de la extensión anterior o con el saliente, marcado con 30 en la figura 27, del cabezal, en el caso en el que sea la última extensión.

6.5.3 PLANO 10. BRAZO DE LA VARILLA EXTENSIBLE

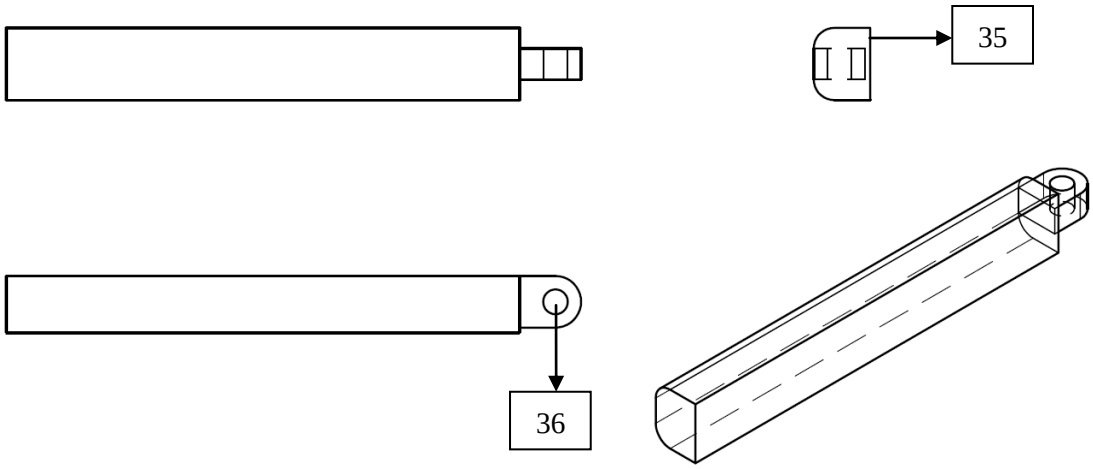


Figura 29: Plano una de las extensiones de la varilla extensible.

La figura 29 representa uno de los dos brazos que se atornillarían al cabezal de la varilla. Su función consiste en posarse sobre la página a leer por su parte lisa marcado con 35 y hacer cierta presión para alisarla lo máximo posible, haciendo que la lectura del dispositivo se produzca en las mejores condiciones.

Están diseñados para que puedan plegarse hacia el interior del dispositivo cuando no se usan, esto se consigue mediante una articulación que permite la rotación, que consiste en un tornillo pasante que atraviesa uno de los orificios de la marca 31 del cabezal en la figura 27 y el orificio del brazo, marcado con 36 en la figura 29. Se hace lo mismo para el otro brazo, pero teniendo en cuenta que la cara lisa debe quedar en la parte externa.

6.5.4 PLANO 11 Y 12. VARILLA EXTENSIBLE ENSAMBLADA

En la figura 30 se muestra la disposición del conjunto de la varilla extensible.

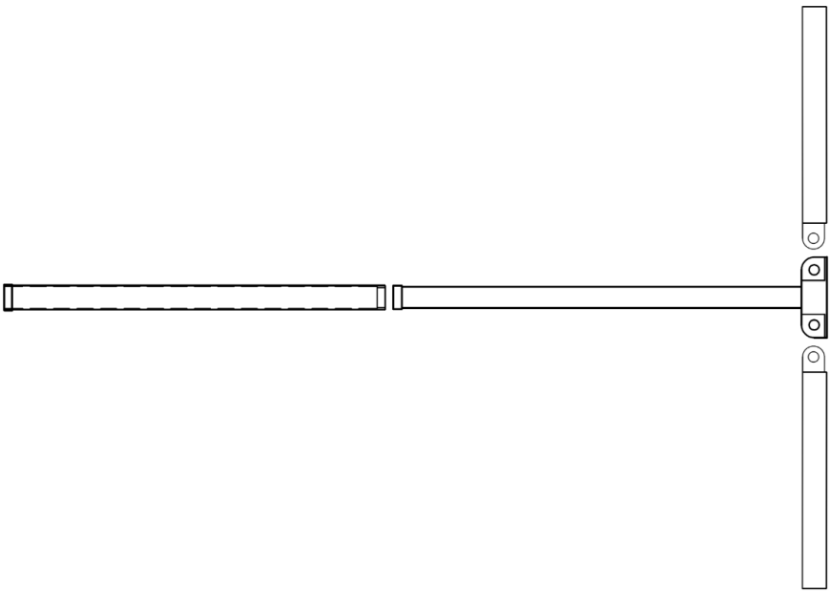


Figura 30: Plano del conjunto de la varilla extensible en su posición desplegada.

Si el dispositivo no se encuentra en uso, la varilla no tiene que estar en su posición extendida, siendo su posición contraída la mostrada en la figura 31.

Figura 31: Plano del conjunto de la varilla extensible en su posición contraída.

El conjunto de la barra extensible se ha diseñado para que no mida más de 9,5 cm de longitud para evitar superar el tamaño de la carcasa.

6.6 ENSAMBLAJE DEL DISPOSITIVO

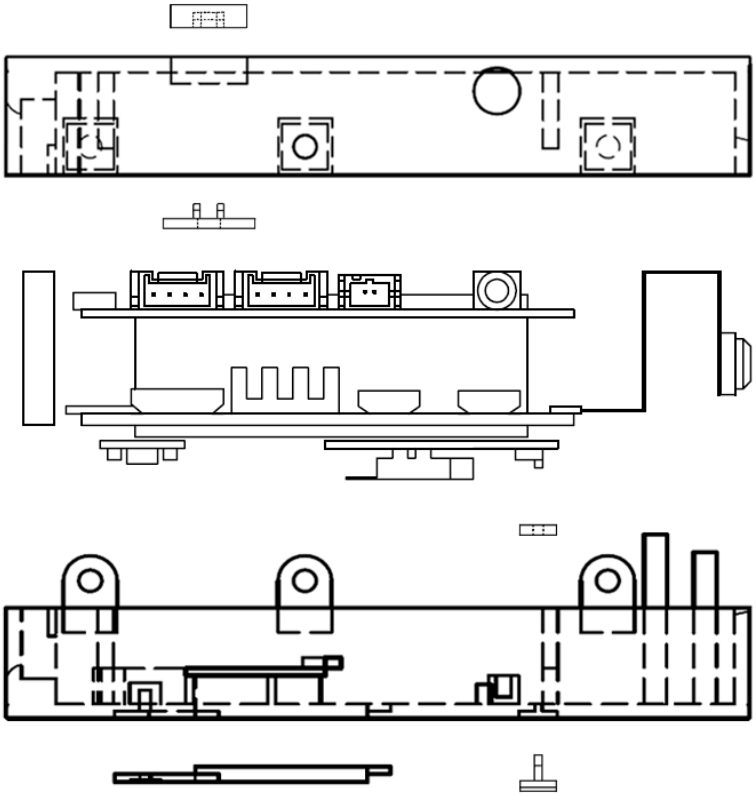


Figura 32: Plano de ensamblado.

En la figura 32 se muestra un esquema general de la distribución de todas las partes del dispositivo descritas anteriormente. Sin embargo, para facilitar el montaje, en este apartado se describen los pasos para ensamblar correctamente el dispositivo. Se deben tenerse en cuenta a la hora de montarlo, ya que, si no es así, puede conllevar que ciertas partes del diseño no funcionen correctamente o reciban daños. La varilla extensible se coloca en el lateral izquierdo del dispositivo una vez que se haya completado el resto de la instalación.

Para ensamblar el dispositivo se necesitan:

- 3 tuercas cuadradas tipo DIN 562 M3,5 [75].
- 3 tornillos M3 x 6mm Wafer Head (“tornillos tipo 1” en este trabajo) [76], [77].
- 3 tornillos M2 x 4mm Wafer Head (“tornillos tipo 2” en este trabajo) [76], [77].
- 1 destornillador de punta Phillips #00 (PH00).
- Cinta adhesiva aislante.

A continuación, se describen los pasos para realizar el ensamblaje del dispositivo.

1. Se introduce la deslizadera en su hueco reservado debajo de la carcasa inferior.
2. Mientras se realiza algo de presión para evitar que la deslizadera se mueva de sitio, se bloquea introduciendo la pieza de fijación de la deslizadera desde la parte superior de la carcasa inferior, mediante la inserción de los salientes de la deslizadera entre el hueco de la pieza de fijación y haciendo presión hasta que los triángulos de la deslizadera sobresalgan por completo.
3. Tomando la carcasa superior, se debe introducir la parte inferior del botón entre las dos prominencias localizadas debajo del orificio dejado para el botón (indicadas como 16 en la figura 20) con las pestañas por el interior de éste.
4. Mientras se mantiene la parte inferior fija, se debe colocar la parte superior del botón, con el hueco que tiene hacia el interior a través del orificio superior de la carcasa superior. Se debe presionar hasta que se sienta que las dos piezas han encajado.
5. A continuación, se debe introducir el soporte para la pila de botón, con el interruptor mirando hacia la parte de debajo, desde la parte de arriba de la carcasa inferior hacia el hueco destinado para ella, donde se introduce el lado opuesto al interruptor a través de la rendija dejada para el compartimento y se presiona ligeramente el soporte hasta que encaja con la superficie inferior.
6. Se debe asegurar el soporte de la pila a la carcasa inferior con “tornillos tipo 2” introduciéndolos a través de los orificios visibles de la parte superior del soporte.
7. Ahora se debe instalar el regulador de voltaje a 5V. Para ello, se introduce en el hueco cuadrado que se localiza en la parte interior de la carcasa inferior, indicado como 21 en la figura 21, y se pone cinta aislante por encima para asegurarlo.
8. A continuación, se introduce la Raspberry en la carcasa inferior con la tarjeta de sonido HAT estando en la parte superior, con cuidado de meter cada uno de los salientes cilíndricos a través de los agujeros de ella, y se presiona hasta que se encuentra el tope.
9. Posteriormente, se retiran las protecciones de los adhesivos del flex de la cámara y se pega el más cercano a la Raspberry sobre un lateral más interior del saliente, marcado con 18 en la figura 21, y el más lejano sobre el lateral opuesto más exterior. Se debe hacer que la cámara se pose sobre el hueco de la carcasa inferior dedicado para ella.
10. A continuación, se despega la protección de adhesivo del altavoz y se posiciona en su compartimento, próximo al punto 15 de la figura 20. La parte que contiene el adhesivo debe mirar hacia el exterior y los cables dirigirse hacia el lado derecho.

11. Dentro de cada uno de los compartimentos cuadrados, indicados como 12 en la figura 20, se debe introducir una tuerca cuadrada y atornillar un "tornillo tipo 1" desde el exterior hasta que éste esté en el límite de la tuerca, sin sobresalir.
12. Tomando la carcasa superior, se coloca por encima de la carcasa inferior, se presiona hasta que se unan por completo y se atornillan los tornillos laterales.
13. Por último, se coloca una pila de botón en su compartimento, que se sebe deslizar hacia la ranura del soporte, y se coloca la tapa del compartimento, donde se sebe introducir primero la parte que contiene las lengüetas y luego hacer una ligera presión sobre la otra parte hasta que queda paralela a la carcasa inferior. Se termina atornillándola con un "tornillo tipo 2".

6.7 CONFIGURACIÓN DE TESSERACT

Tesseract OCR cuenta con diversas opciones que pueden mejorar o empeorar el resultado del reconocimiento de texto. Cada una de éstas representa un beneficio para cierto tipo de texto. El objetivo de este apartado es encontrar la mejor configuración para la mayor cantidad de textos posibles teniendo en cuenta que el dispositivo está destinado principalmente para reconocer textos en papel.

Como se ha indicado en el apartado 6.1 "Funcionalidades mediante código", donde el OCR ya se encuentra programado con varias opciones en su línea de código, en este apartado se explica cada una de estas configuraciones, siendo en concreto la línea de código expuesta en el apartado 6.1 la mostrada a continuación.

```
original=pytesseract.image_to_string(img, config='-l spa --oem 3 --psm 1')
```

Mientras que "pytesseract.image_to_string" es simplemente una sentencia para llamar a la función que empieza a reconocer el texto, todo lo que está dentro de los paréntesis, salvo "img", que es el nombre del archivo que se quiere analizar, son las configuraciones utilizadas. Como se puede ver, están dentro de la variable "config". Estas configuraciones son el diccionario, el motor y la distribución, respectivamente.

6.7.1 DICCIONARIO

La primera configuración, "-l spa", marca el diccionario que se debe usar para el reconocimiento de texto. En este caso es "spa" debido a que son las letras asignadas por Tesseract OCR para referirse al diccionario en español, que es el que se va a usar para este proyecto. Sin embargo, se pueden seleccionar hasta otros 65 idiomas si se descarga el

diccionario de cada uno de ellos. Por defecto, se incluye el diccionario inglés, que se usa de forma predeterminada si la configuración "-l" no se completa con otro diccionario [78].

6.7.2 MOTOR DEL OCR

```
OCR Engine modes:
  0    Legacy engine only.
  1    Neural nets LSTM engine only.
  2    Legacy + LSTM engines.
  3    Default, based on what is available.
```

Figura 33: Tipos de motor de "Tesseract" [79].

La segunda configuración se refiere al modo del motor del OCR (*OCR Engine Modes*), hay cuatro opciones; solo motor antiguo, solo motor de redes LSTM neuronales, antiguo motor y redes neuronales y lo que haya disponible.

El menú de selección original de estas opciones se muestra en la figura 33. Cada uno de estos motores representa distintos grados de velocidad y precisión.

En general, usar únicamente el motor antiguo no es recomendable, ya que se comenten bastantes errores en la detección de caracteres. Usar sólo las redes neuronales presenta resultados mejores que la opción mencionada anteriormente y no penaliza la velocidad de detección, mientras que usar la combinación de las dos mejoras aún más los resultados, pero penaliza la velocidad de detección. Por último, la opción de usar lo que haya disponible es similar a la opción combinada, pero accede a las bases por si hubiese alguna opción extra que se pueda utilizar y la aplica [80].

En este proyecto se ha determinado por usar la opción 3, combinación de los dos, ya que es la que mejor relación entre precisión y velocidad presenta para los tipos de texto más comunes que se quieren analizar.

6.7.3 DISTRIBUCIÓN DE LA IMAGEN

La configuración de la distribución de imagen del Tesseract OCR es, con diferencia, la opción más importante que se puede configurar, ya que esta opción puede llevar de obtener un resultado más preciso al no tener resultados directamente. Las distintas opciones que se pueden seleccionar se pueden ver en la figura 34.

```
Page segmentation modes:
  0    Orientation and script detection (OSD) only.
  1    Automatic page segmentation with OSD.
  2    Automatic page segmentation, but no OSD, or OCR.
  3    Fully automatic page segmentation, but no OSD. (Default)
  4    Assume a single column of text of variable sizes.
  5    Assume a single uniform block of vertically aligned text.
  6    Assume a single uniform block of text.
  7    Treat the image as a single text line.
  8    Treat the image as a single word.
  9    Treat the image as a single word in a circle.
 10    Treat the image as a single character.
 11    Sparse text. Find as much text as possible in no particular order.
 12    Sparse text with OSD.
 13    Raw line. Treat the image as a single text line,
       bypassing hacks that are Tesseract-specific.
```

Figura 34: Distribuciones de imagen en "Tesseract" [79].

Esta configuración debe tener en cuenta la posición en la que se encuentra el texto en la imagen que se quiera analizar, por ejemplo, si se tiene un cartel con el texto más separado sería conveniente usar una distribución "11" o "12", diseñadas para este tipo de texto, ya que una opción "7" no obtendría buenos resultados. Si, por el contrario, se tiene un texto continuo en un párrafo, las opciones "4", "5" o "6" proporcionan mejores resultados que el resto de las opciones.

Como el dispositivo está diseñado para poder actuar con textos muy distintos de forma automática, la opción más conveniente, y la que se va a utilizar, es la "1" que se corresponde a una segmentación automática de la página usando OSD (*Orientation and script detection*). El OSD permite detectar el texto en el caso en el que la orientación de la imagen no sea la ideal, que puede ser importante si el usuario es ciego o haya texto en distintas orientaciones. Esta opción, sin embargo, tiene el inconveniente de que presenta un mayor coste computacional y, por lo tanto, se ve penalizada la velocidad de obtención de resultados.

7. RESULTADOS

En este apartado se exponen los resultados obtenidos en las pruebas realizadas para determinar las mejores configuraciones posibles del OpenCV y Tesseract OCR. Además, se expone el dispositivo prototipo con sus principales características.

7.1 RESULTADOS DE LAS PRUEBAS CON OPENCV

Este apartado contiene los resultados de las múltiples pruebas realizadas, mediante OpenCV, con ocho fotografías para ver la variación en la precisión del OCR con cada alteración de la imagen. Además, se incluye una breve explicación de los métodos que se usan para hacer la comparación. En concreto, a cada imagen se la sometió a las mismas variaciones:

- Una lectura de la imagen, pero en gama de grises. Esto equivale a una lectura básica con el OCR, por lo que se usa este resultado como base para la comparación.
- Una variación de contraste adaptativo.
- Un cambio en los DPI (*dots per inch*) de la imagen, que varía la resolución de la imagen a 300x300, ya que supuestamente funciona mejor con el OCR.
- Una binarización Gaussiana, que ayuda a establecer contornos en función del color.
- Una eliminación de ruido y una binarización adaptativa tipo “Otsu” (Segmentación por umbralización).
- Una combinación entre cambio de DPI, eliminación de ruido y binarización tipo Otsu [48],[49].

Por último, se comparan los resultados obtenidos con el original usando un método cuantitativo que calcula los parámetros WER (*word error rate*), MER (*match error rate*) y WIL (*Word information lost*) y un método más subjetivo usando la funcionalidad Difflib incluida en Python.

7.1.1 WER, MER Y WIL

Con el fin de comparar la efectividad de cada método de alteración frente al reconocimiento, se han calculado una serie de parámetros de forma cuantitativa. En concreto, mediante la librería Jiwer [81], se ha calculado el parámetro WER, MER y WIL, que señalan, de formas distintas, la precisión que tiene un reconocimiento de texto con respecto a su similitud al texto original. Cada método se explica brevemente a continuación [82].

- WER (*Word error rate*): Es un parámetro que mide la precisión de un reconocimiento mediante la comparación de palabras entre el texto reconocido y el original. En concreto, en este proyecto se usa su variante diseñada para usarse en oraciones, conocida como “WER in CSR”, donde CSR viene de “*connected speech recognition*”.

Funciona mediante el emparejamiento de palabras similares entre sí de forma no simétrica y siguiendo su orden de aparición. Esto puede causar que el valor resultante pueda ser superior a "1" si el reconocimiento del texto otorga un resultado desordenado con respecto al texto original. Su fórmula se muestra en la ecuación (1), donde "S" es el número de sustituciones, "D" el número de eliminaciones, "I" el número de inserciones y "H" el número de palabras correctas.

$$WER(CSR) = \frac{S+D+I}{H+S+D} \qquad (1)$$

- MER (*match error rate*): Es una alternativa al WER que otorga la probabilidad de que una pareja de palabras no sea correcta. Aunque es similar a WER, se diferencia principalmente en que es una función de comparación simétrica y es usada generalmente cuando la probabilidad de error es muy alta, evitando que el valor exceda de "1" [83], como sucedería usando WER. Su fórmula se muestra en la ecuación (2).

$$MER = 1 - \frac{H}{H+S+D+I} \qquad (2)$$

- WIL *(Word information lost)*: Es un método que resulta de una aproximación a un método mucho más complejo, que no se cubre en este informe, llamado RIL (*Relative information lost*). Otorga una aproximación de la proporción de información perdida debido a la presencia de errores. Su fórmula se muestra en la ecuación (3).

$$WIL = 1 - \frac{H^2}{(H+S+D)(H+S+I)} \qquad (3)$$

7.1.2 DIFFLIB Y LEYENDA PARA LEER SUS RESULTADOS

Difflib es una funcionalidad de Python que contiene clases y funciones destinadas a la comparación de secuencias, ya sean éstas archivos, directorios, textos, entre otros.

En este proyecto se va a usar exclusivamente para la comparación entre el texto generado por el OCR, cuando se ha pasado por una modificación concreta de la imagen, y el texto original. Esto tiene el objetivo de ver las alteraciones de imagen producen textos más cercanos al texto real.

Cuando termina su proceso, Difflib devuelve una variable de salida en forma de bloque de texto que almacena las líneas de texto que contienen diferencias superpuestas a la línea original correspondiente, señalando, además, sus diferencias.

Esta variable de salida indicada usa una leyenda para marcar las variaciones entre textos que es la siguiente:

- El carácter "**-**" al comienzo de una línea indica que dicha línea aparece en el texto leído por el OCR, pero no en el original.
- El carácter "**+**" al comienzo de una línea indica que dicha línea aparece en el texto original, pero no en el leído por el OCR.
- El carácter de espacio en blanco al comienzo de una línea indica que dicha línea es coincidente en los dos textos.
  El carácter "?" al comienzo de una línea muestra con cierto detalle las diferencias que hay en las líneas no coincidentes.

En las líneas con "**?**" se utilizan los caracteres "+", "-" y "^" para marcar los elementos que son distintos:

- El carácter "-" es para señalar aquellos caracteres que aparecen en el texto leído por el OCR, pero no en el original.
- El carácter "+" para señalar aquellos que aparecen en el texto original y no en el leído por el OCR.
- El carácter "^" para destacar los caracteres diferentes tanto en el texto leído por el OCR como en el original [84].

Un ejemplo de resultado del Difflib es el mostrado a continuación.

Ejemplo: (Sacado de "Texto 1" con variación de contraste adaptativo aplicada)

```
-
- señalados para la práctica de las actuaciones. Las actuaciones de comprobación hada se
?                                                                                  ^
+ señalados para la práctica de las actuaciones. Las actuaciones de comprobación limitada se
?                                                                          +       ^^^^
-
- realizarán en las oficinas de la Administración tributaria, salvo las que procedan según la normas
?       ^
+ realizarán en las oficinas de la Administración tributaria, salvo las que procedan según la normativa
?       ^^^^
```

7.1.3 RESULTADOS DE LAS PRUEBAS

En este apartado se muestran tanto los datos cuantitativos como los subjetivos, junto con un análisis de cada uno de ellos. En primer lugar, en la tabla 5 se muestran los datos cuantitativos correspondientes a los valores de WER, MER y WIL de cada variación.

Tabla 5: Comparación cuantitativa de resultados entre las distintas pruebas con OpenCV.

		Original	Contraste adaptativo	Variación de DPI	Binarización Gaussiana	B. Otsu + Elim. de ruido	DPI + Otsu + Elim. de ruido
Texto 1	WER	0,116	**0,108**	0,163	0,989	0,244	0,247
	MER	0,114	**0,105**	0,163	0,989	0,237	0,242
	WIL	0,190	**0,175**	0,273	0,995	0,402	0,414
Texto 1 Cerca	WER	0,187	**0,162**	0,239	1	0,271	0,458
	MER	0,183	**0,157**	0,230	1	0,261	0,411
	WIL	0,300	**0,254**	0,346	1	0,423	0,591
Texto 1 Lejos	WER	0,22	**0,161**	0,659	0,626	0,422	0,539
	MER	0,217	**0,157**	0,652	0,563	0,407	0,527
	WIL	0,352	**0,265**	0,777	0,764	0,633	0,765
Texto 1 Luz	WER	0,374	**0,114**	0,241	0,898	0,241	0,308
	MER	0,373	**0,109**	0,233	0,898	0,224	0,28
	WIL	0,458	**0,163**	0,338	0,957	0,347	0,401
Libro borroso	WER	0,441	0,596	**0,206**	1	1	0,802
	MER	0,351	0,471	**0,2**	1	1	0,626
	WIL	0,464	0,617	**0,332**	1	1	0,82
Teatro	WER	0,962	**0,769**	0,846	0,971	1	1,24
	MER	0,962	**0,714**	0,846	0,971	1	0,806
	WIL	0,981	**0,907**	0,893	0,993	1	0,942
Texto con ilustración	WER	0,286	0,439	1	1	**0,429**	1,429
	MER	0,286	**0,375**	1	1	0,429	0,714
	WIL	0,49	**0,554**	1	1	0,674	0,837
Texto con ilustración 2	WER	0,971	0,928	**0,523**	1	0,855	1
	MER	0.944	0,889	**0,5**	1	0,843	0,852
	WIL	0,997	0,987	**0,676**	1	0,969	0,973

Al mostrar los valores relativos a los errores encontrados, éstos deben ser interpretados con una consideración de mejoría descendente, siendo un valor próximo a 0 correspondiente a un texto muy similar al original, y un valor próximo a 1 un reconocimiento completamente diferente al original.

Cabe destacar que en los valores de WER de la variación de "DPI+Otsu+Elim. de ruido" correspondientes a los textos de "Teatro", "Texto con ilustración" obtienen valores superiores a 1, y "Texto con ilustración 2", se obtiene un valor de 1 cuando, en realidad, si se está obteniendo un resultado del reconocimiento, evidenciado por los valores de MER y WIL, que no son 1. Esto es debido a lo expuesto anteriormente en el apartado 7.1.1, donde se explica que el parámetro WER puede otorgar resultados superiores a 1 en casos en los que la probabilidad de error sea alta o haya mucho desorden en el texto reconocido. En estos casos, se debe usar preferentemente el resultado de MER o de WIL.

Debido a que los resultados cuantitativos obtenidos no tienen en cuenta situaciones en las que un reconocimiento distinto al original puede, de forma efectiva, tener una dicción muy similar a éste y, por lo tanto, inapreciable, como en el caso de la sustitución de una "e" por una "a" o la alteración leve de una palabra, se ha decidido hacer una valoración mediante la comparación subjetiva de textos usando la librería Difflib. En concreto, para establecer un criterio de comparación, se toma el resultado de la variación en gris como texto original y se compara con el del resto de variaciones. En la tabla 6 se recogen los resultados de esta comparación subjetiva.

Según la comparación hecha con los resultados de la imagen correspondiente al "texto 1", de un reconocimiento original en gris con un reconocimiento original bastante bueno se obtiene que:

- Se aprecia una ligera mejoría cuando se aplica la variación de contraste.
- El cambio de DPI mejora algunos reconocimientos de ciertas partes del texto, pero empeora bastante otros en gran medida.
- La binarización gaussiana no funciona, hace que el OCR no devuelva ningún resultado.
- Aplicando la combinación entre el eliminador de ruido y la binarización tipo Otsu, los reconocimientos empeoran notablemente.
- Por último, con un proceso combinado de cambio de DPI, el eliminador de ruido y la binarización tipo Otsu actuando a la vez, el resultado es aún peor.

Para la imagen llamada "texto 1 cerca", sin embargo, se obtiene un reconocimiento original deficiente. En este caso, la comparación resulta en:

- Una clara la mejoría al aplicar la variación de contraste adaptativo, mejorando el resultado del reconocimiento de forma sobresaliente.
- El cambio de DPI no presenta mejoría en el resultado, pero tampoco el empeoramiento claro que se puede intuir en los datos cuantitativos de la tabla 5.
- La binarización gaussiana no funciona.

- El eliminador de ruido junto con la binarización tipo Otsu generan un reconocimiento levemente más entendible, pese a los datos cuantitativos.
- Con el cambio de DPI, el eliminador de ruido y la binarización tipo Otsu actuando a la vez no se aprecia mejoría o empeoramiento notable.

En la llamada “texto 1 lejos”, se obtiene un reconocimiento inicial aceptable. Aplicando cada variación y comparando cada resultado con el reconocimiento original se obtiene:

- Una notable mejoría aplicando la variación de contraste adaptativo.
- Un claro empeoramiento en todas las demás variaciones, que daban unos resultados caóticos y de muy baja calidad.

La imagen del “texto 1 Luz” otorga un reconocimiento inicial difícil de entender, posiblemente debido al gradiente de luz de la fotografía, siendo un resultado peor al del texto 1 (normal) aun siendo la posición del texto y la distribución muy similar. Con la comparación se observan los siguientes resultados:

- La variación de contraste, de nuevo, mejoraba notablemente el reconocimiento.
- El cambio de DPI también ejerce cierta mejora, pero, a su vez, causa que algunas frases se corten a la mitad, lo cual tampoco es una consecuencia deseable.
- La binarización gaussiana funciona en este caso. Sin embargo, presenta un reconocimiento caótico e ininteligible.
- Aplicando el eliminador de ruido y la binarización tipo Otsu, los reconocimientos mejoran visiblemente, aunque de forma algo leve en comparación a la variación de contraste.
- Con el cambio de DPI, el eliminador de ruido y la binarización tipo Otsu actuando a la vez, si bien se aprecia cierta mejoría con respecto al reconocimiento inicial, empeora el reconocimiento frente a la misma opción sin aplicar el cambio de DPI.

Para la imagen relativa al “libro borroso”, el reconocimiento inicial es bueno, aunque se observan ciertas irregularidades, que seguramente se deben a que la imagen contiene algo de texto localizado en el borde izquierdo, correspondiente a la página anterior, que está siendo reconocido también. Esto se podría solucionar con un recortador de bordes, presentándose como una posible mejora futura. Al comparar:

- La variación del contraste presenta un ligero empeoramiento en el reconocimiento, sobre todo en las primeras líneas del texto.
- El cambio de DPI presenta, por el contrario, una mejoría muy notable, aunque ciertas palabras reconocidas sufren de errores que no aparecen en el reconocimiento inicial.

- Tanto la binarización gaussiana como la tipo Otsu devuelven resultados nulos.
- La combinación del cambio de DPI, el eliminador de ruido y la binarización tipo Otsu resultan en un severo empeoramiento con respecto al reconocimiento original.

El texto de "Teatro" presenta un reconocimiento inicial malo, donde muchas líneas no son reconocidas y se producen saltos constantes. Comparando, se observa que:

- La variación de contraste ayuda a que parte de esas líneas se vuelvan reconocibles, pese a que el rendimiento general sigue siendo pobre, que se traduce en los altos valores cuantitativos de la tabla 5.
- Aplicando el cambio de DPI, la diferencia no se vuelve notable.
- La binarización gaussiana no funciona más que en una línea, siendo, además, errónea.
- La binarización tipo Otsu no funciona.
- La combinación del cambio de DPI, el eliminador de ruido y la binarización tipo Otsu ayuda levemente a reconocer una mayor cantidad de líneas. En este caso, para la valoración cuantitativa de la tabla 5, es aconsejable tomar el valor de MER, que mejora con respecto al original, ya que el valor de WER presenta un resultado mayor que "1".

La imagen denominada "texto con ilustración" presenta un reconocimiento inicial perfecto. Al tener un resultado inicial perfecto, lo máximo a lo que se aspira es a obtener lo mismo incluso después de aplicar las variaciones. Sin embargo, la comparación resulta en que:

- Tanto la variación de contraste como la binarización tipo Otsu con eliminación de ruido empeoran el resultado levemente.
- El cambio de DPI elimina una línea por completo de las dos que contiene la imagen.
- La binarización gaussiana no funciona.
- La combinación del cambio de DPI, el eliminador de ruido y la binarización tipo Otsu resultan en un empeoramiento severo del reconocimiento.

Por último, del "texto con ilustración 2" se obtiene un reconocimiento inicial pasable. El resultado obtenido de la comparación muestra:

- Una clara mejoría al aplicar la variación de contraste.
- Un claro empeoramiento al aplicar el resto de los cambios. Esto entra en conflicto con los datos cuantitativos mostrados en la tabla 5, donde se produce una clara mejoría al aplicar el cambio de DPI, no al aplicar la variación de contraste. Según la valoración subjetiva, sin embargo, no es así.

7.1.4 CONCLUSIONES DE LAS PRUEBAS

A partir de los resultados obtenidos, tanto de forma cuantitativa como subjetiva, se deduce que la variación adaptativa de contraste permite una mejora en la mayoría de los casos para el reconocimiento de texto mediante OCR. Sin embargo, en los libros encuadernados con gran variación de luz o inclinación, no se obtiene una mejora, por lo tanto, no es aconsejable su uso.

Por otro lado, mediante la binarización gaussiana no se obtuvieron los resultados que se esperaban, ya que no generó resultados correctos en la mayoría de los casos. En cuanto a la binarización tipo Otsu, aunque genera resultados en la mayoría de los casos, los resultados muestran que degrada el reconocimiento. Por último, el cambio de DPI presenta unos resultados inconsistentes, ya que en algunos casos mejora el reconocimiento y en otros lo degrada. Esto hace que su uso sea desaconsejado.

En la Tabla 6 se resumen los resultados expuestos anteriormente, donde se ha comparado las transformaciones por el numero de veces en las que una variación es efectiva.

Tabla 6: Comparación subjetiva de resultados entre las distintas pruebas con OpenCV.

	Original	Contraste adaptativo	Variación de DPI	Binarización Gaussiana	B. Otsu + Elim. de ruido	DPI + Otsu + Elim. de ruido
Texto 1	Bueno	⊖	⊖	✗	✗	✗
Texto 1 Cerca	Malo	✓	✗	✗	⊖	✗
Texto 1 Lejos	Medio	✓	✗	✗	✗	✗
Texto 1 Luz	Medio	✓	⊖	✗	⊖	⊖
Libro borroso	Bueno	✗	✓	✗	✗	✗
Teatro	Malo	✓	✗	✗	✗	⊖
Texto con ilustración	Perfecto	✗	✗	✗	✗	✗
Texto con ilustración 2	Medio	✓	✗	✗	✗	✗
Total		5,5	2	0	1	1

El valor numérico se ha calculado de la siguiente manera:

 = +1 Será usado cuando el cambio suponga una mejora clara en el reconocimiento.

 = +0,5 Será usado cuando el cambio suponga ligera mejora en el reconocimiento.

 = +0 Será usado cuando el cambio no suponga una mejora o empeore el resultado.

Según el valor numérico visto en la tabla 6, la única variación que presenta una mejora clara en una gran diversidad de tipos de texto es la variación de contraste adaptativo (CLAHE) y es, por lo tanto, la escogida para implementar en el código general.

7.2 PROTOTIPO

En este apartado se muestran imágenes ilustrativas del dispositivo ya ensamblado, donde se marca cada una de las características más relevantes que lo componen y un párrafo que describe algunas de las dificultades encontradas. Además, se incluyen imágenes de algunas de las pruebas realizadas para evaluarlo y un vídeo que muestra una de estas pruebas. En concreto, la figura 35 muestra la parte delantera del dispositivo, la figura 36 la parte trasera y las figuras 37 y 38 muestran imágenes de algunas de las pruebas realizadas con voluntarios para evaluar las capacidades del dispositivo. El siguiente enlace muestra un vídeo donde se realiza una prueba con el dispositivo. Enlace al vídeo: https://youtu.be/6XTw_95ZLsA

Durante el montaje se nota que la impresión 3D reprodujo las piezas de la carcasa con unas proporciones algo distintas a las diseñadas debido a la expansión del material de impresión, lo que causa que varias piezas no encajen en sus sitios correspondientes. Para realizar el montaje, se hace uso de una lima y de un *cutter* con el fin de adaptar las piezas. En concreto, al ser necesario conectar la Raspberry a la corriente como alternativa temporal al sistema de energía fallido, se habilita un orificio en el lado derecho del dispositivo, liberando el conector de corriente al exterior. Además, se pega el botón a su soporte, ya que no encajan, y se liman las superficies correspondientes a la cámara y el altavoz. En el futuro, puesto que se debe rediseñar la carcasa inferior para acomodar la batería, es preciso tener en cuenta la expansión del material, de aproximadamente entre 0,5 mm y 1 mm, para que este problema no suceda de nuevo.

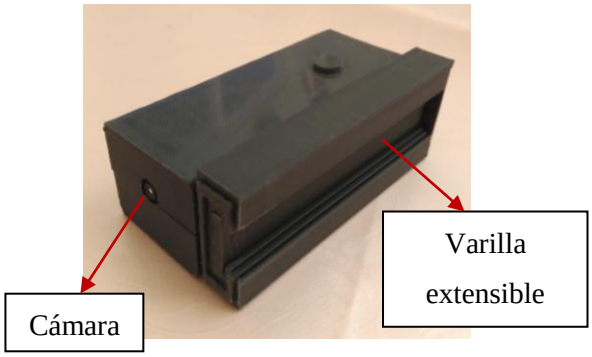


Figura 35: Parte delantera del dispositivo.

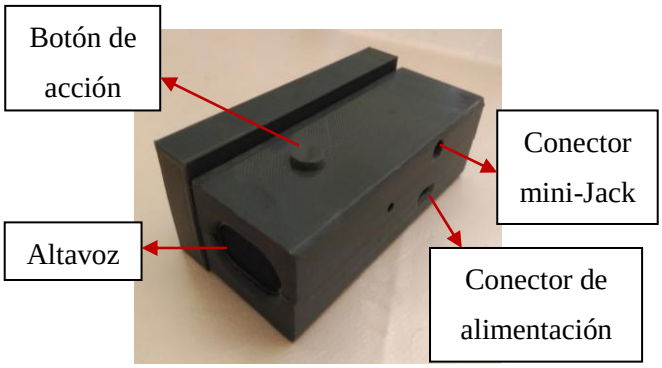


Figura 36: Parte trasera del dispositivo.

Figura 37: Imagen de una prueba con una persona anciana con cataratas.

Figura 38: Imagen de una prueba con una persona ciega.

7.3 RESULTADOS GENERALES DEL PROTOTIPO

En este apartado se describen los resultados obtenidos a partir del prototipo ensamblado. En concreto, se usa una estructura similar a la usada en el apartado 5.1 “selección de componentes de hardware”, describiendo el desempeño de cada módulo con el objetivo de comparar la teoría a la realidad. Además, se incluyen testimonios e impresiones de una persona anciana, una ciega y otras personas de prueba, y la decisión de crear un modelo de utilidad.

7.3.1 MÓDULO DE PROCESAMIENTO

A continuación, se describe el desempeño de la Raspberry Pi Zero en la realización de la función exigida. Se debe tener en cuenta que los datos mostrados han sido tomados usando el entorno de escritorio de Raspbian, que en el dispositivo final no se usa, por lo que los datos correspondientes pueden ser peores a los que el dispositivo final otorga.

En la tabla 7 se muestra el tiempo de duración de cada proceso para cada tipo de imagen. Los resultados indican que, a mayor cantidad de palabras tenga el texto, mayor es el tiempo necesario para realizar los procesos destinados al análisis por OCR, la conversión a voz mediante el programa TTS y la reproducción. Sin embargo, los tiempos necesarios para realizar la captura de la imagen y la variación de contraste se mantienen constantes independientemente del tamaño del texto.

Tabla 7: Duración de los procesos dependiendo de la imagen.

En segundos	Importación	Captura	V. Contraste	OCR	TTS	Reproducción	Total
Libro Borroso	37,06	3,65	0,49	102,43	8,7	72,42	224,75
Teatro	27,4	3,54	0,46	64,19	7,54	67,13	**170,26**
Texto1	27,43	3,55	0,47	**257,67**	26,16	202,56	517,84
Texto1 Cerca	26,25	3,54	0,48	182,52	19,13	148,84	380,76
Texto1 Lejos	26	3,54	0,48	208,65	29,62	244,35	512,64
Texto1 Luz	25,72	3,56	0,5	221,78	23,31	191,59	466,46
Texto Ilust.	24,89	3,55	0,49	**15,88**	0,81	7,54	**53,16**
Texto Ilust. 2	24,84	3,57	0,51	44,98	4,25	35,43	113,58

El tiempo de importación se mantiene constante exceptuando el valor correspondiente a “LibroBorroso”, que es un 27% superior. Esto es debido a que, al ser el primer experimento, la caché se encuentra limpia, sin embargo, los próximos experimentos cuentan ya con parte de la información guardada en la caché, por lo que el proceso se agiliza.

Además, se observa que el proceso es especialmente ineficiente para los textos que contienen ilustraciones que ocupan la mayor parte de la imagen, ya que en “TextoConIlustración” requiere casi un minuto para terminar el proceso teniendo el texto dos frases, mientras que “Teatro” requiere el triple de tiempo para completar el proceso, pero superando este texto al primero por diez veces en el número de líneas.

Se debe añadir que el dispositivo tarda 2 minutos y 23 segundos en iniciarse, que se deben sumar al tiempo de importación. Sin embargo, una vez se realiza la primera foto, ninguno de estos dos procesos volverá a repetirse hasta el reinicio del dispositivo, por lo que sus tiempos no deben tenerse en cuenta más que una vez para un uso continuado.

Además de la duración del proceso, debido al tamaño del dispositivo y al entorno cerrado en el que se encuentra, se miden las temperaturas con el fin de evitar sobrecalentamiento, y con ello una pérdida de rendimiento.

Los resultados referentes a las temperaturas se muestran en la tabla 8. Se puede ver que éstas se mantienen muy estables independientemente del texto a leer y del proceso en el que se encuentre el dispositivo. La media tiene un valor de 48,731 ºC, siendo la máxima temperatura alcanzada los 51,382 ºC en la CPU y 51,4 ºC en la GPU, y la mínima 44,926 ºC en la CPU y 45,5 ºC en la GPU. Puesto que el procesador de la Raspberry está diseñado para soportar temperaturas en el rango de -40 °C a 85 °C [85], no existe riesgo de sobrecalentamiento.

Tabla 8: Temperaturas de los procesos dependiendo de la imagen.

En ºC	Módulo	Importación	Captura	Var. Contraste	OCR	TTS	Reproducción	Media
Libro Borroso	CPU	47,616	47,616	47,078	49,768	50,306	46,54	48,15
	GPU	47,6	47,6	48,2	49,8	50,8	46,5	48,42
Teatro	CPU	48,692	48,154	48,692	49,768	50,306	46,54	48,69
	GPU	48,7	48,7	49,2	49,8	50,3	46,5	48,87
Texto1	CPU	48,692	48,154	49,23	50,844	50,844	44,926	48,78
	GPU	48,7	48,2	48,7	50,8	50,8	45,5	48,78
Texto1 Cerca	CPU	48,692	48,154	48,692	50,844	50,844	45,464	48,78
	GPU	47,6	48,7	48,7	50,8	50,8	45,5	48,68
Texto1 Lejos	CPU	48,692	48,154	48,692	50,844	51,382	45,464	48,87
	GPU	48,7	48,7	48,7	50,8	51,4	45,5	48,97
Texto1 Luz	CPU	48,692	48,154	48,692	50,844	50,844	45,464	48,78
	GPU	48,7	48,7	48,7	50,8	50,8	45,5	48,87
Texto Ilust.	CPU	48,692	47,616	48,154	48,692	49,23	48,154	48,42
	GPU	48,7	48,2	48,7	48,7	49,2	48,2	48,62
Texto Ilust. 2	CPU	48,692	48,692	48,692	50,306	49,768	47,078	48,87
	GPU	48,7	48,7	49,2	50,3	50,3	47,6	49,13

7.3.2 MÓDULO DE SONIDO

El altavoz de 1 W instalado ofrece un sonido suficiente para el entendimiento, sin embargo, la calidad es mejorable. Además, durante la prueba del sonido, aparece un fallo no previsto, el cual consiste en que el sonido no deja de reproducirse a través de los altavoces cuando se conecta un dispositivo de reproducción alternativo a través del puerto mini-Jack del dispositivo. Esto es problemático en determinados casos en los que el usuario quiere disfrutar del dispositivo sin molestar a la gente que tiene alrededor mediante el uso de unos auriculares.

El dispositivo reproduce el sonido en un rango de 69 dB a 81 dB. Este nivel de sonido no se puede alterar de forma manual por el momento, ya que no se ha contemplado un regulador para variarlo en la versión del prototipo actual.

7.3.3 MÓDULO DE ENERGÍA

El sistema de energía planteado inicialmente otorga un resultado nulo, el dispositivo no se enciende. Para averiguar la causa se hace uso de un multímetro y se toman voltajes en DC de las tomas del regulador, de la pila y del portapilas. Se obtiene como resultado que la diferencia de potencial entre los bornes de la pila es de 2,67 V, entre los bornes del portapilas es de 2,13 V y entre las tomas del regulador DC-DC es de 2,29 V.

Estos resultados indican que, aunque teóricamente es posible, en la realidad en ningún momento se alcanzan los 5 V requeridos por la Raspberry para funcionar. Se observa que se produce una caída importante de potencial entre la pila y el portapilas, que hace que el potencial caiga a menos de 2,7 V, que es el mínimo voltaje que el regulador requiere para funcionar con eficiencia, como se muestra en la figura 43 del Anexo I. Esto causa que el regulador DC-DC aumente levemente el potencial, pero a niveles insuficientes.

En definitiva, pese a que se suponía de antemano que el sistema de energía propuesto podía dar problemas en cuanto al tiempo de autonomía del dispositivo, no se podía anticipar que fuese totalmente inviable debido a la baja eficiencia del portapilas, ya que esta característica no aparece señalada en sus especificaciones técnicas. En un futuro se puede solucionar este problema sustituyendo el sistema de energía propuesto por una batería PiSugar, como se expone en el apartado 5.1.3 “Módulo de energía”.

7.3.4 MÓDULO DE CAPTURA

Se realizan pruebas a la captura imágenes probando el dispositivo en distintas condiciones de luminosidad, medida previamente en lux, a una distancia constante de 30 cm y observando que el reconocimiento no varía significativamente. Los resultados obtenidos muestran que el módulo de captura toma imágenes apropiadas para el OCR cuando funciona en un rango de entre 90 lux y 5286 lux aproximadamente, lo que indica que, salvo en condiciones de penumbra u obscuridad, el dispositivo debe ser capaz de funcionar correctamente, aunque se observa que, si la inclinación de la imagen capturada es muy elevada, la precisión cae drásticamente. En un modelo futuro, se debe valorar la posibilidad de incluir una luz LED para estabilizar lo máximo posible las condiciones lumínicas del entorno.

7.3.5 MÓDULO DE INTERACCIÓN

Para valorar la practicidad del dispositivo como dispositivo lector, se prueba en una persona mayor con cataratas. Al principio, la persona mayor muestra desconcierto porque no conoce el dispositivo y lo considera un trozo de metal. Al explicarle el funcionamiento, la persona mayor considera el dispositivo fácil de usar y, tras unos momentos, usa el dispositivo con normalidad. Sin embargo, debido a la baja calidad del altavoz, en ciertas ocasiones le cuesta entender lo que el dispositivo reproduce.

También se valora la practicidad para una persona con ceguera a través de dos voluntarios. Éstos tienen problemas para realizar el reconocimiento debido a que la varilla extensible no es lo suficientemente rígida, lo que implicaba dificultades para colocar el dispositivo totalmente perpendicular al texto. Además, uno de ellos echa en falta un detector de márgenes y un modo para aumentar la velocidad de la voz que se reproduce. También se recopilaron sugerencias para posibles funcionalidades futuras que podrían ser útiles, como la funcionalidad bluetooth para la conexión del dispositivo con una tabla electrónica de Braille, un botón específico para repetir la lectura.

En cuanto a la facilidad de uso para una persona ciega, además de con los voluntarios ya mencionados, se prueba en diversas personas con los ojos vendados, ninguna de ellas con conocimiento previo del diseño del dispositivo. Todas consiguen orientarlo correctamente en posición vertical, aunque una de ellas lo orienta de forma invertida. Sin embargo, una vez se les enseña la posición correcta del dispositivo, este problema se soluciona y son capaces de usarlo sin problema.

Por último, en cuanto a la ergonomía, se pregunta a todas las personas de prueba si les resulta cómodo el uso del dispositivo y se reciben respuestas afirmativas de todas ellas, aunque una persona comenta que le resultaría más cómodo si el botón se localizase en la parte inferior del dispositivo en vez de en la superior.

7.3.6 MODELO DE UTILIDAD

Pese a las mejoras que se pueden aplicar al prototipo, el dispositivo sigue siendo un producto práctico que supone una alternativa económica a productos similares para las personas con problemas de la visión, ciegas o analfabetas, por lo que se procede al planteamiento y presentación de un modelo de utilidad, el cual se encuentra actualmente en trámites por el equipo de gestión de patentes de la Universidad Rey Juan Carlos.

8. PRESUPUESTO

En este apartado se describe el presupuesto estimado para reproducir el dispositivo actual. Además, se incluye un presupuesto para la siguiente versión del dispositivo, que altera la elección de ciertos componentes con el objetivo de solucionar ciertas carencias que el prototipo actual tiene. Para la reproducción del dispositivo actual, el presupuesto se expone en la tabla 9, donde no se incluye el coste total de la impresión en 3D.

Tabla 9: Presupuesto para la recreación del prototipo actual.

Componente	Precio (€)/Ud.	Unidades	Precio total (€)
Raspberry Pi Zero W	10,50	1	10,50
Tarjeta de memoria micro-SD 16GB	4,00	1	4,00
Mini-disipador de calor 6mm de altura	1,30	1	1,30
Tarjeta de sonido ReSpeaker 2-Mics Pi HAT	10,00	1	10,00
Minialtavoz de 8 Ω y 1 W	3,10	1	3,10
Soporte para una pila de botón Lilypad 20 mm	2,41	1	2,41
Regulador DC-DC S7V8F5 Pololu 5 V	6,20	1	6,20
Cámara ZeroCam NOIR sin filtro de infrarrojos	16,18	1	16,18
Conector macho de 40 pines COM1101	1,15	1	1,15
Tuerca DIN 562 M3,5	0,50	3	1,50
Tornillo M3 x 6 mm Wafer Head	0,05	3	0,15
Tornillo M2 x 4 mm Wafer Head	0,05	3	0,15
Gramos de material ABS para impresión 3D	0,015	45	0,68
Coste total			**57,32**

Por contraposición, en la tabla 10 se expone el presupuesto generado para crear la siguiente versión del dispositivo, donde se incluyen unos altavoces mejores, una batería PiSugar, un interruptor para realizar el encendido y un potenciómetro de rueda para darle al usuario la capacidad de alterar el volumen del sonido.

Tabla 10: Presupuesto para la siguiente versión del dispositivo.

Componente	Precio (€)/Ud.	Unidades	Precio total (€)
Raspberry Pi Zero W	10,50	1	10,50
Tarjeta de memoria micro-SD 16 GB	4,00	1	4,00
Mini-disipador de calor 6 mm de altura	1,30	1	1,30
Tarjeta de sonido ReSpeaker 2-Mics Pi HAT	10,00	1	10,00
Altavoz Innovateking-EU de 8 Ω y 3 W	5,50	1	5,50
Batería PiSugar Portable de 1200 mAh	37,99	1	37,99
Potenciómetro de rueda de 16 mm de diam.	0,04	1	0,04
Cámara ZeroCam NOIR sin filtro de infrarrojos	16,18	1	16,18
Interruptor deslizante EG1201A	0,55	1	0,55
Diodo LED SparkFun COM-11450	0,77	2	1,54
Conector macho de 40 pines COM1101	1,15	1	1,15
Tuerca DIN 562 M3,5	0,50	3	1,50
Tornillo M3 x 6 mm Wafer Head	0,05	3	0,15
Tornillo M2 x 4 mm Wafer Head	0,05	3	0,15
Gramos de material ABS para impresión 3D	0,015	45	0,68
Coste total			**91,23**

Además de lo expuesto anteriormente, se deben tener en cuenta las horas empleadas para la realización del dispositivo. El total de horas empleadas asciende a las 378 horas aproximadamente que, aplicando el salario medio por hora de un Ingeniero Industrial en España (13,38 €/hora) [86], equivalen a un coste adicional de 5.057,64 € que se corresponden al coste de desarrollo.

9. CONCLUSIONES

En este apartado se describen las conclusiones a las que se han llegado al realizar este Trabajo Fin de Grado. Se analiza el grado de éxito al alcanzar los objetivos propuestos y, posteriormente, se explica un posible futuro para el dispositivo creado.

En cuanto a los objetivos propuestos, se han abordado y evaluado todos ellos y se han cumplido del siguiente modo:

- Se ha logrado crear un dispositivo capaz de leer un texto en formato físico a partir de materiales de bajo coste. Siendo, además, portátil y fácil de usar.
- Se ha diseñado una carcasa sencilla que, aunque requiere mejora, es apta para el uso de persona ciega y, por lo tanto, también es apta para el uso de gente con dificultades en la visión o analfabeta.
- Se ha desarrollado un programa capaz de ejecutar todo el proceso de lectura y reproducción del texto.
- Se ha analizado la precisión del reconocimiento y se han aplicado algunas mejoras que la incrementan para la mayoría de los tipos de texto. Aunque todavía requiere una mayor mejora.
- Se ha probado el dispositivo con varios voluntarios, entre ellos una persona anciana con problemas en la visión y dos ciegos, y se han tomado notas para una posible mejora futura.
- Se han iniciado los trámites para crear un modelo de utilidad del dispositivo.

El cumplimiento de estos objetivos demuestra que la creación de un dispositivo de lectura de texto físico que cumple todos los requisitos de la tabla 1 y mientras es accesible económicamente es posible, aunque requiere más desarrollo. Se ha generado un repositorio github con el código diseñado por si alguna persona quiere usarlo o mejorarlo. El repositorio del proyecto se encuentra en github donde está disponible el código en el siguiente enlace: https://github.com/tortugal897/BoligrafoLector/tree/main

Este dispositivo puede presentarse a una empresa especializada en el diseño de productos para ciegos con el fin de conseguir financiación para continuar con su desarrollo, avalando el dispositivo mediante su posible modelo de utilidad, donde se aplicarían las mejorar descritas en el apartado 8. “Presupuesto”, además de optimizar el código y diseñar una carcasa aún más accesible.

10. LÍNEAS FUTURAS

En este apartado se describen las líneas futuras de trabajo que se le pueden aplicar al dispositivo actual con el objetivo de hacerlo más cómodo, accesible y funcional para el público objetivo. En concreto, se exponen las mejoras en orden de implementación preferente.

1. Implementación de una batería PiSugar, que servirá de sustitución al sistema de alimentación propuesto inicialmente. Al ser la portabilidad uno de los puntos clave del dispositivo, se considera una labor prioritaria funcionar de forma autónoma, sin depender de la cercanía a una fuente de corriente ni de un cable.
2. Mejorar la calidad de sonido. Con el fin de mejorar la calidad de reproducción, es necesario sustituir el altavoz actual por uno de mayor calidad.
3. Implementación de un regulador de volumen, que permitirá al usuario ajustar el nivel de sonido según sus necesidades. Esto se lograría añadiendo un potenciómetro.
4. Incorporación de diodos LED, que permitirán al dispositivo trabajar siempre en unas condiciones lumínicas más constantes.
5. Incluir un botón para aumentar la velocidad de lectura, que es especialmente útil para las personas con ceguera, que están más acostumbradas a escuchar audios a altas velocidades.
6. Mejorar la carcasa y la varilla extensible. Todas las mejoras listadas anteriormente son mejoras de hardware que tienen que ser contempladas por la carcasa, además de mejorar la distribución de ésta para hacerla más ergonómica. La varilla extensible debe hacerse con un diseño más fijo, que facilite el posicionamiento del dispositivo sobre el texto y con otro material que permita reducir su volumen mientras se mantiene su resistencia, como el aluminio o el acero.
7. Mejorar la velocidad de procesamiento mediante la optimización del código y el uso del chip gráfico de la Raspberry Pi Zero como unidad de procesamiento paralela. Cuanto más se acerque el dispositivo a la inmediatez, más competitivo será.
8. Incorporación de un método de reconocimiento de márgenes que avise al usuario, mediante sonido, cuando el texto completo se encuentre dentro del rango de la cámara. Esto es especialmente útil para las personas ciegas.
9. Implementación de una funcionalidad Bluetooth que permita al dispositivo conectarse a unos altavoces externos, unos auriculares o una tabla de Braille.
10. Añadir una función de almacenamiento de lecturas en la memoria por si se desea leer un texto escaneado en otro momento o repetir la lectura. Requeriría conectar el dispositivo a un ordenador o la adición de un botón cuya única función sea repetir la última lectura.

11. BIBLIOGRAFÍA

[1] PÉREZ, J. A. *La escritura: el mejor invento de la humanidad*. 23 julio 2020. Actualizado a 16 de abr. de 2021. Disponible en: https://www.uvrcorrectoresdetextos.com/post/la-escritura-el-mejor-invento-de-la-humanidad

[2] INSTITUTO DE ESTADÍSTICA DE LA UNESCO. El mapa de la alfabetización en el mundo. En: *El orden mundial*. 28 noviembre 2019. Disponible en: https://elordenmundial.com/mapas/mapa-alfabetizacion-en-el-mundo/

[3] AGENCIA EFE. En España hay casi 700.000 personas analfabetas. En: *elDiario.es*. 08 septiembre 2016, 17:44h. Disponible en: https://www.eldiario.es/sociedad/espana-personas-analfabetas_1_3841406.html

[4] GBD 2019 BLINDNESS AND VISION IMPAIRMENT COLLABORATORS. Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: the Right to Sight: an analysis for the Global Burden of Disease Study. En: *THE LANCET Global Health*. 01 febrero 2021. Volume 9, Issue 2. E144-E160. Disponible en: https://www.thelancet.com/journals/langlo/article/PIIS2214-109X(20)30489-7/fulltext

[5] ONCE. *El Braille: lectura, aprendizaje, alfabeto y signos*. Disponible en: https://www.once.es/servicios-sociales/braille

[6] FREEDOM SCIENTIFIC. *JAWS®*. Disponible en: https://www.freedomscientific.com/products/software/jaws/

[7] THE DAISY CONSORTIUM. *DAISY Format*. Disponible en: https://daisy.org/activities/standards/daisy/

[8] LEAS, D., PERSOON, E., SOIFFER, N. y ZACHERLE, M. Daisy 3: a standard for accessible multimedia books. En: *IEEE Computer Society*. IEEE multiMedia 15 (2008) H. 4 S. 28-37. ISSN 1070-986X. Disponible en: https://core.ac.uk/download/pdf/197558712.pdf

[9] ADOBE. *Adobe acrobat reader. El mejor visor PDF gratuito*. Disponible en: https://acrobat.adobe.com/es/es/acrobat/pdf-reader.html

[10] MICROSOFT. *Microsoft Word*. Disponible en: https://www.microsoft.com/es-es/microsoft-365/word

[11] GOOGLE CLOUD. *Text-To-Speech*. Disponible en: https://cloud.google.com/text-to-speech/?hl=es

[12] OLLÉ, C. Herramientas para leer sin ver. En: *ComeIn: Revista de los Estudios de Ciencias de la Información y de la Comunicación*. Marzo 2012, núm. 9. ISSN 1696-3296. Disponible en: https://comein.uoc.edu/divulgacio/comein/es/numero09/articles/Article-Candela-Olle.html

[13] W3C WEB ACCESSIBILITY INITIATIVE. *W3C Accessibility Standards Overview*. 30 abril 2021. Disponible en: https://www.w3.org/WAI/standards-guidelines/

[14] PUERTO, K. Nokia Braille Reader en vídeo. En: *Xataka móvil*. 17 septiembre 2009. Disponible en: https://www.xatakamovil.com/aplicaciones/nokia-braille-reader-en-video

[15] GOOGLE PLAY. *Traductor de Google*. 1 junio 2021. Disponible en: https://play.google.com/store/apps/details?id=com.google.android.apps.translate&hl=es&gl=US

[16] SCANMARKER. *Scanmarker Air*. Disponible en: https://scanmarker.com/es/product/scanmarker-air-negro/

[17] ORCAM. *Orcam: Esto es magia*. Disponible en: https://www.orcam.com/es/

[18] BRACERO, F. OrCam My Eye 2, un ojo inteligente para abrir el mundo a los invidentes. En: *La Vanguardia*. 05 octubre 2019, 00:05. Actualizado a 15 octubre 2019, 12:51. Disponible en: https://www.lavanguardia.com/tecnologia/20191007/47800399773/orcam-my-eye-2-ojos-inteligentes-para-invidentes.html

[19] MICROSOFT PRENSA. *Microsoft presenta Seeing AI en español, la app gratuita que facilita el reconocimiento y descripción del entorno a personas ciegas*. 3 diciembre 2019. Disponible en: https://news.microsoft.com/es-es/2019/12/03/microsoft-presenta-seeing-ai-en-espanol-la-app-gratuita-que-facilita-el-reconocimiento-y-descripcion-del-entorno-a-personas-ciegas/

[20] APPLE. *iPhone 12 y iPhone 12 mini*. Disponible en: https://www.apple.com/es/iphone/

[21] APPLE. *iOS 14*. Disponible en: https://www.apple.com/es/ios/ios-14/

[22] RASPBERRY PI. *Raspberry Pi Zero*. Disponible en: https://www.raspberrypi.org/products/raspberry-pi-zero/

[23] ARDUINO.CC. *Arduino Nano*. Disponible en: https://store.arduino.cc/arduino-nano

[24] NVIDIA. *Tienda Jetson*. Disponible en: https://www.nvidia.com/es-es/autonomous-machines/jetson-store/

[25] VÉLEZ, J. F. *Tema 5: Implantación y despliegue de sistemas de visión artificial*. Departamento de Informática y Estadística de la URJC. Consulta: 12 junio 2021. Formato en PDF, 3,3MB. No publicado.

[26] PYTHON. *Python™*. Disponible en: https://www.python.org/

[27] SEEED, THE IOT HARDWARE ENABLER. *ReSpeaker 2-Mics Pi HAT*. Disponible en: https://www.seeedstudio.com/ReSpeaker-2-Mics-Pi-HAT.html

[28] RS PRO. *Portapilas para 2 pilas AAA, montaje Chasis*. Disponible en: https://es.rs-online.com/web/p/portapilas/5123552/?cm_mmc=ES-PLA-DS3A-_-google-_-CSS_ES_ES_Pilas_%26_Bater%C3%ADas_y_Cargadores_Whoop-_-(ES:Whoop!)+Portapilas-_-5123552&matchtype=&pla-339536766682&gclid=CjwKCAjwtpGGBhBJEiwAyRZX2nBz2-5PEMV91b0B82J8fxslfWl-h4z8JboRFsRlkWM5tStj1NiYlhoC9NQQAvD_BwE&gclsrc=aw.ds

[29] DOUTEL, F. Cómo alimentar tu Raspberry Pi Zero con dos pilas AA. En: *Xataka smart home*. 21 Diciembre 2015. Actualizado a 21 diciembre 2015, 23:44. Disponible en: https://www.xatakahome.com/trucos-y-bricolaje-smart/como-alimentar-tu-raspberry-pi-zero-con-dos-pilas-aa

[30] GTIWUNG. *GTIWUNG 20Pcs CR2032 Soporte de Batería, 2 x 3V CR2032 Soporte de Batería Pilas de Botón con Interruptor de Cable, Portapilas para Pilas Botón*. Disponible en: https://www.amazon.es/GTIWUNG-Soporte-Bateria-Interruptor-Portapilas/dp/B0823W1MX7

[31] PIMORONI. *LilyPad Coin Cell Battery Holder - Switched - 20mm*. Disponible en: https://shop.pimoroni.com/products/lilypad-coin-cell-battery-holder-switched-20mm

[32] PISUGAR. *PiSugar Portable*. Disponible en: https://www.pisugar.com/

[33] PISUGAR. *Pisugar Portable 1200 mAh batería de Litio módulo de Potencia para Raspberry PI-Zero, PI-Zero W/WH Accesorios de Modelo (no Incluye Raspberry PI) (1200ma)*. Disponible en: https://www.amazon.es/Pisugar-port%C3%A1til-Compatible-Raspberry-Accesorios/dp/B07RDNT8CY

[34] UPTON, E. Zero grows a camera connector. En: *Raspberry Pi Blog*. 16 mayo 2016. Disponible en: https://www.raspberrypi.org/blog/zero-grows-camera-connector/

[35] THEPIHUT. *ZeroCam NOIR - Camera for Raspberry Pi Zero*. Disponible en: https://thepihut.com/products/zerocam-noir-camera-for-raspberry-pi-zero

[36] *Optical Character Recognition – History of Optical Character Recognition (OCR)*. HISTORY COMPUTER. The History of Computing. Disponible en: https://history-computer.com/optical-character-recognition-history-of-optical-character-recognition-ocr/

[37] G., J. Digitalización OCR. En: *Novadoc 102 innovación. Blog Toda la actualidad informática para instituciones documentales*. 14 diciembre 2018. Disponible en: http://blog.102novadoc.es/digitalizacion-ocr

[38] EVANS, H. Ray Kurzweil. En: *Who made America?*. 30 junio 2004. Disponible en: https://www.pbs.org/wgbh/theymadeamerica/whomade/kurzweil_hi.html

[39] *Reconocimiento óptico de caracteres*. Wikipedia: la enciclopedia libre. 27 mayo 2021, 09:44. Disponible en: https://es.wikipedia.org/wiki/Reconocimiento_%C3%B3ptico_de_caracteres

[40] ONIEVA, D. Pasa a Word el texto de cualquier foto o PDF con estos programas OCR. En: *SZ Softzone*. 08 junio 2021, 15:56. Disponible en: https://www.softzone.es/programas/utilidades/mejores-aplicaciones-ocr-escritura-texto/

[41] *Tesseract OCR*. Wikipedia: la enciclopedia libre. 23 diciembre 2019, 21:04. Disponible en: https://es.wikipedia.org/wiki/Tesseract_OCR

[42] *Reconocimiento de imagen y texto (OCR) para el procesamiento de imágenes Python*. Programador clic. 2020. Disponible en: https://programmerclick.com/article/2602385282/

[43] MICROSOFT AZURE. *Computer Vision. Servicio de inteligencia artificial que analiza el contenido de imágenes y vídeos*. Disponible en: https://azure.microsoft.com/es-es/services/cognitive-services/computer-vision/#pricing

[44] AWS. Amazon Rekognition. Automatice su análisis de imagen y video con aprendizaje automático. Disponible en: https://aws.amazon.com/es/rekognition/?blog-cards.sort-by=item.additionalFields.createdDate&blog-cards.sort-order=desc

[45] GOOGLE CLOUD. *Vision AI*. Disponible en: https://cloud.google.com/vision

[46] G2. *OpenCV Alternatives & Competitors*. Disponible en: https://www.g2.com/products/opencv/competitors/alternatives

[47] *OpenCV*. Wikipedia: la enciclopedia libre. 04 junio 2021, 23:53. Disponible en: https://es.wikipedia.org/wiki/OpenCV

[48] Xiong, N. *Introduction to Image Processing in Python*. Consulta: 12 junio 2021. Formato en Cuaderno de Google Drive, 5,79MB. Disponible en: https://colab.research.google.com/github/xn2333/OpenCV/blob/master/Image_Processing_in_Python_Final.ipynb

[49] MANK, A. Image Processing in OpenCV. En: *OpenCV-Python Tutorials*. 10 agosto 2014. Actualizado a 08 julio 2016. Disponible en: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html

[50] *Conversor texto-voz*. Wikipedia: la enciclopedia libre. 22 mayo 2021, 21:13. Disponible en: https://es.wikipedia.org/wiki/Conversor_texto-voz

[51] GOOGLE. gTTS 2.2.2. En: *PyPI*. 4 febrero 2021. Disponible en: https://pypi.org/project/gTTS/

[52] SVOXMOBILEVOICES. *Introducing SVOX Mobile Voices*. 6 octubre 2010. Disponible en: https://svoxmobilevoices.wordpress.com/page/2/

[53] THE CENTER FOR SPEECH TECHNOLOGY RESEARCH. THE UNIVERSITY OF EDINBURGH. *The Festival Speech Synthesis System*. Diciembre 2014. Disponible en: https://www.cstr.ed.ac.uk/projects/festival/

[54] JONSD, MAVISON, RHDUNN, VALDISVI. eSpeak text to speech. En: *SourceForge*. 06 diciembre 2017. Disponible en: http://espeak.sourceforge.net/

[55] BHAT, N. M. pyttsx3 2.90. En: *PyPI*. 6 julio 2020. Disponible en: https://pypi.org/project/pyttsx3/

[56] *SVOX*. Wikipedia: la enciclopedia libre. 11 enero 2021, 05:01. Disponible en: https://en.wikipedia.org/wiki/SVOX

[57] ANDROID. *Presentamos Android 11*. Disponible en: https://www.android.com/intl/es_es/

[58] SEEED, THE IOT HARDWARE ENABLER. *Getting Started with Raspberry Pi*. Disponible en: https://wiki.seeedstudio.com/ReSpeaker_2_Mics_Pi_HAT_Raspberry/

[59] DE LANGEN, J. Playing and Recording Sound in Python. En: *Real Python*. 2019. Disponible en: https://realpython.com/playing-and-recording-sound-python/

[60] MARKS, T. S. playsound 1.2.2. En: *PyPI*. 29 junio 2017. Disponible en: https://pypi.org/project/playsound/

[61] HAMILTON, J. simpleaudio 1.0.4. En: *PyPI*. 29 noviembre 2019. Disponible en: https://pypi.org/project/simpleaudio/

[62] PYTHON. *Winsound - Sound-playing interface for Windows*. 14 junio 2021. Disponible en: https://docs.python.org/3/library/winsound.html

[63] VIDEOLAN. *VLC media player*. Disponible en: https://www.videolan.org/vlc/index.es.html

[64] ADEVA, R. Todo sobre Linux, el sistema operativo de código abierto. En: *AZ adsl zone*. 04 junio 2021, 18:47. Disponible en: https://www.adslzone.net/reportajes/software/que-es-linux/

[65] RASPBERRY PI FOUNDATION, THE DEBIAN PROJECT y otros. *Welcome to Raspbian*. Disponible en: https://www.raspbian.org/

[66] UNIVERSITY OF TARTU, CIBERNETICA y RASPBERRY PI FOUNDATION. *Thonny. Python IDE for beginners*. 18 mayo 2021. Disponible en: https://thonny.org/

[67] *alsamixer*. Wikipedia: la enciclopedia libre. 20 septiembre 2019, 07:20. Disponible en: https://es.wikipedia.org/wiki/Alsamixer

[68] I.E.S. LOPEZ DE ARENAS. *Programando con Python. Thonny*. 15 abril 2018. Disponible en: http://lopezdearenas.org/python/2018/04/15/thonny/

[69] MEDINA, E. Cómo forzar el guardado permanente de los cambios en alsamixer. En: *MuyLinux*. 10 octubre 2019. Disponible en: https://www.muylinux.com/2019/10/10/alsamixer-forzar-guardado-permanente/

[70] OPENCV. OPEN SOURCE COMPUTER VISION. *Histograms - 2: Histogram Equalization*. Disponible en: https://docs.opencv.org/master/d5/daf/tutorial_py_histogram_equalization.html

[71] SVOX AG. *SVOX Pico. Speech Output Engine SDK. 1.0.0*. 20 abril 2009. Formato en PDF, 0,4MB. Disponible en: http://dafpolo.free.fr/telecharger/svoxpico/SVOX_Pico_Manual.pdf

[72] SEEED, THE IOT HARDWARE ENABLER. ReSpeaker 2-Mics Raspberry Pi HAT. En: *Digi-Key Electronics*. Disponible en: https://www.digikey.es/es/product-highlight/s/seeed/respeaker-2-mics-raspberry-pi-hat

[73] PIMORONI. *Mini Oval Speaker - 8 Ohm 1 Watt*. Disponible en: https://shop.pimoroni.com/products/mini-oval-speaker-8-ohm-1-watt

[74] AUTODESK. AutoCAD. Disponible en: https://www.autodesk.es/products/autocad/overview?panel=buy&mktvar002=4417848|SEM|8054028189823645911 99kwd-297275808151&plc=ACDIST&term=1-YEAR&support=ADVANCED&quantity=1&ef_id=CjwKCAjwtpGGBhBJEiwAyRZX2mCmj-bY6qb9OxubP9aNS8O15avbVSHzuk3o5nlqlNcE_m1d-Xo7uBoCjbwQAvD_BwE:G:s&s_kwcid=AL!11172!3!429967133905!e!!g!!autocad!805402818 9!82364591199&mkwid=sJovdvX0U

[75] INDUSTRIAS UGATU S. L. Ficha técnica. Tuerca cuadrada, forma baja DIN 562. Formato en PDF, 0,4MB. Disponible en: https://www.ugatu.com/media/uploads/pdf/DIN%20562.pdf

[76] *List of DIN standards*. Wikipedia: la enciclopedia libre. 22 mayo 2021, 07:25 (UTC). Disponible en: https://en.wikipedia.org/wiki/List_of_DIN_standards

[77] LAPTOPSCREWS. *More than you ever wanted to know about measuring screws. How To Measure A Screw*. Disponible en: https://www.laptopscrews.com/Sizing.htm

[78] TESSERACT-OCR. Data Files for Version 4.00. En: *tessdoc. Tesseract documentation*. 29 noviembre 2016. Disponible en: https://tesseract-ocr.github.io/tessdoc/Data-Files

[79] RAJ, A. Optical Character Recognition (OCR) using Tesseract on Raspberry Pi. En: *Circuit Digest*. 21 agosto 2019. Disponible en: https://circuitdigest.com/microcontroller-projects/optical-character-recognition-ocr-using-tesseract-on-raspberry-pi

[80] WEIL, S., MAYO, C., WAJER, M. y otros. TESSERACT (1) Manual Page. En: *GitHub*. 11 abril 2021, 10:43 CEST. Disponible en: https://github.com/tesseract-ocr/tesseract/blob/master/doc/tesseract.1.asc

[81] VAESSEN, N. jiwer 2.2.0. En: *PyPI*. 18 noviembre 2020. Disponible en: https://pypi.org/project/jiwer/

[82] MORRIS, A. C., MAIER, V. Y GREEN, P. *From WER and RIL to MER and WIL: improved evaluation measures for connected speech* recognition. Formato en PDF, 0,11MB. 04-08 octubre 2004. Disponible en: https://www.isca-speech.org/archive/archive_papers/interspeech_2004/i04_2765.pdf

[83] KAFLE, S. Y HUENERFAUTH, M. *Evaluating the Usability of Automatically Generated Captions for People who are Deaf or Hard of Hearing*. Formato en PDF, 1,92MB. 2017. Disponible en: https://arxiv.org/ftp/arxiv/papers/1712/1712.02033.pdf

[84] PHERKAD. Difflib: encontrando las diferencias. En: *Python 3 para impacientes*. 20 mayo 2016, 17:21. Disponible en: https://python-para-impacientes.blogspot.com/2016/05/difflib-encontrando-las-diferencias.html

[85] RASPBERRY PI FOUNDATION. *Documentation, FAQs, Table of contents, What is its operating temperature? Does it need a heatsink?*. Disponible en: https://www.raspberrypi.org/documentation/faqs/#pi-performance-temps

[86] *Salario medio para Ingeniero Industrial en España 2021*. Talent.com. 2021. Disponible en: https://es.talent.com/salary?job=ingeniero+industrial#:~:text=El%20salario%20ingeniero%20industrial%20promedio,13%2C38%20%E2%82%AC%20por%20hora.

[87] HATTERSLEY, L. Raspberry Pi 4, 3A+, Zero W - specs, benchmarks & thermal tests. En: *The Magi Magazine. The official Raspberry Pi magazine*. 2019. Disponible en: https://magpi.raspberrypi.org/articles/raspberry-pi-specs-benchmarks

[88] KINGSTON. *Kingston SDCS/16GB Tarjeta de Memoria Sd 1, 16 gb, Negro*. Disponible en: https://www.amazon.es/Kingston-SDCS-16GB-velocidades-Adaptador/dp/B079H6PDCK

[89] PIMORONI. *Raspberry Pi 3 Heatsink – 6mm (works with HATs)*. Disponible en: https://shop.pimoroni.com/products/heatsink?variant=17021246791

[90] SEEED, THE IOT HARDWARE ENABLER. *ReSpeaker 2-Mics pHAT*. Disponible en: https://shop.pimoroni.com/products/respeaker-2-mics-phat

[91] SEEED, THE IOT HARDWARE ENABLER. *Overview*. Disponible en: https://wiki.seeedstudio.com/ReSpeaker_2_Mics_Pi_HAT/

[92] POLOLU. Pololu 5V Step-Up/Step-Down Voltage Regulator S7V8F5. Disponible en: https://shop.pimoroni.com/products/pololu-5v-step-up-step-down-voltage-regulator-s7v8f5

[93] PIMORONI. *Camera Module for Raspberry Pi Zero – Without infra-red filter*. Disponible en: https://shop.pimoroni.com/products/raspberry-pi-zero-camera-module?variant=40236918090

[94] PIMORONI. *Male 40-pin 2x20 HAT Header – Male 40-pin Header*. Disponible en: https://shop.pimoroni.com/products/male-40-pin-2x20-hat-header?variant=10476117383

[95] RASPBERRY PI FOUNDATION. *Raspberry Pi Zero W*. Formato en PDF, 0,18MB. 2015. Disponible en: https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/rpi_SCH_ZeroW_1p1_reduced.pdf

[96] CANONICAL. *Ubuntu downloads*. 2021. Disponible en: https://ubuntu.com/download

[97] RASPBERRY PI FOUNDATION. *Raspberry Pi OS*. Disponible en: https://www.raspberrypi.org/software/

[98] NUTTALL, B. Y JONES, D. New opencv builds. En: *Piwheels*. 27 septiembre 2018. Disponible en: https://blog.piwheels.org/new-opencv-builds/

[99] NUTTALL, B. Y JONES, D. opencv-contrib-python. En: *Piwheels*. 08 junio 2021. Disponible en: https://www.piwheels.org/project/opencv-contrib-python/

[100] JERRM. Comentario en: *libttspico0 & libttspico-utils missing in Buster*. En: *Raspbian Bugs*. 24 agosto 2019. Disponible en: https://bugs.launchpad.net/raspbian/+bug/1835974

[101] PYTHON. *Difflib - Helpers for computing deltas*. 14 junio 2021. Disponible en: https://docs.python.org/3/library/difflib.html

[102] HELLMANN D. difflib – Compare sequences. En: *PyMOTW*. 11 julio 2020. Disponible en: https://pymotw.com/2/about.html

[103] DUGGAN, M. cdifflib 1.2.5. En: *PyPI*. 08 mayo 2019. Disponible en: https://pypi.org/project/cdifflib/

[104] KILI, A. How to Install and Run VLC Media Player as Root in Linux. En: *TecMint. The ideal linux blog for sysadmins and geeks*. 28 abril 2017. Disponible en: https://www.tecmint.com/run-vlc-media-player-as-root-in-linux/

[105] DEXTER INDUSTRIES. *Five Ways to Run a Program on Your Raspberry Pi at Startup*. 2021. Disponible en: https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/

[106] RASPBERRY PI FOUNDATION. *Getting started with the Camera Module*. Disponible en: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/7

[107] HUGHES, D. *API Reference. Picamera 0.3 documentation*. 2013. Disponible en: https://picamera.readthedocs.io/en/release-0.3/api.html

ANEXO I. ESPECIFICACIONES TÉCNICAS Y COSTE

En este apartado se exponen las especificaciones técnicas de los componentes elegidos, junto con su precio de compra y una ilustración de cada uno.

ANEXO I.I RASPBERRY PI ZERO

Modelo: Raspberry Pi Zero W [87]. Visible en la figura 39.

Dimensiones: 65 x 30 x 5 mm

Microprocesador: Broadcom BCM2835 SoC

CPU: ARM11 de 1 GHz de velocidad

RAM: 512 MB

Tecnología inalámbrica: 2.4 GHz 802.11n wireless LAN

Tecnología bluetooth: Bluetooth Classic 4.1 y LE

Alimentación: 5 V, administrados por un conector micro-USB tipo B

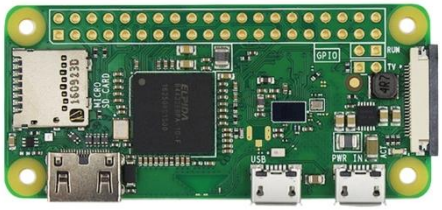


Figura 39: Imagen de una Raspberry Pi Zero W [87].

Video y Audio: Vídeo a 1080P HD y audio estéreo a través del conector mini-HDMI

Pines: 40 pines de propósito general

Conexiones adicionales: Puerto para conexión de cámara

Precio: 10,50 €

ANEXO I.II TARJETA MICROSD

Para funcionar, la Raspberry necesita una tarjeta microSD que contenga el sistema operativo. La microSD que se usa para este proyecto es la mostrada a continuación, visible en la figura 40, pero cualquier otra tarjeta microSD con una capacidad de, al menos, 8 GB es también apropiada.

Modelo: Kingston 16GB microSDHC [88]

Capacidad: 16 GB

Velocidad de lectura: Hasta 100 MB/s

Velocidad de escritura: 10 MB/s

Precio: 4 €

Figura 40: Imagen de la microSD [88].

ANEXO I.III DISIPADOR DE CALOR

Puesto que el procesador de la Raspberry es sometido a una alta exigencia, se instala un disipador de calor encima de él. En concreto, para este proyecto, se usa uno lo suficientemente pequeño para que no entre en conflicto con la tarjeta de sonido HAT instalada por encima. El disipador se muestra en la figura 41 junto con sus características.

Modelo: Raspberry Pi 3 Heatsink – 6mm [89]

Dimensiones: 14,5 x 14,5 x 6 mm

Precio: 1,30 €

Figura 41: Imagen del disipador de calor [89].

ANEXO I.IV TARJETA DE SONIDO

Modelo: ReSpeaker 2-Mics Pi HAT [90]

Dimensiones: 65 x 30 x 7 mm

Códec: WM8960 con "driver" de altavoz de clase D

Salidas de audio: Mini-Jack de 3.5 mm (con salida de 40 mW) y conector de altavoz tipo JST2.0 (proveyendo 1 W para cargas de 8 Ω)

Alcance: Hasta 3 metros

Conectores extra: Soporte para GPIO e I2C

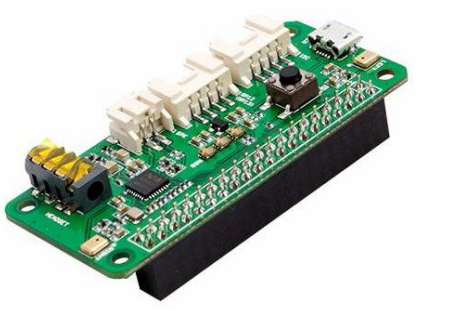


Figura 42: Imagen del ReSpeaker 2-Mics Pi HAT [90].

Funcionalidades extra: Un botón programable, tres LEDs APA102 RGB y dos micrófonos.

Precio: 10 €

En la figura 42 se muestra una imagen de la tarjeta HAT y en la figura 43 su esquema.

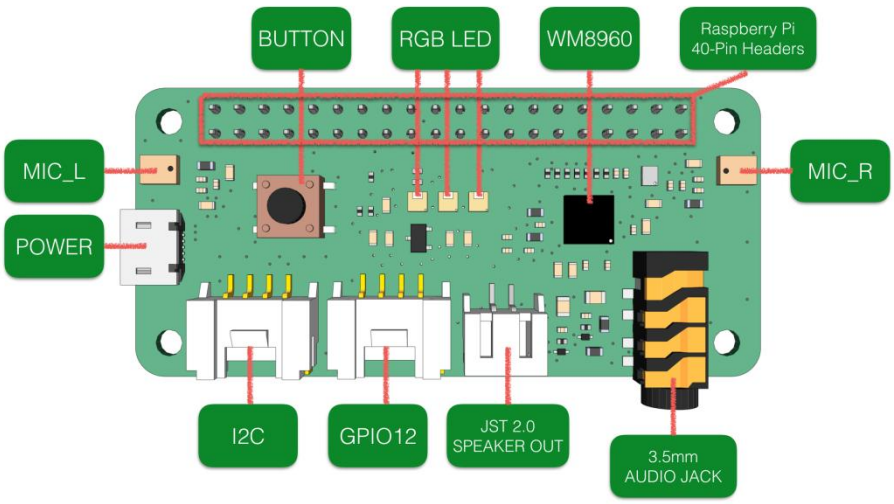


Figura 43: Esquema del ReSpeaker 2-Mics Pi HAT [91].

ANEXO I.V ALTAVOZ

Modelo: Mini oval speaker -8 Ohm 1 Watt [73]. Figura 44.

Dimensiones: 30 x 20 x 5 mm

Potencia: 1 W

Resistencia: 8 Ω

Peso: 1.4 g

Longitud del cable: ~110 mm

Conector: Molex #0532610271 "PicoBlade" de 1.25 mm de paso.

Precio: 3,10 €

Figura 44: Imagen del mini altavoz oval [73].

ANEXO I.VI SOPORTE PARA PILA

Modelo: LilyPad Coin Cell Battery Holder - Switched - 20mm [31]. Figura 45.

Dimensiones: 32 mm de diámetro, 6 mm de altura

Capacidad: Pila de botón de hasta 20 mm de diámetro

Corriente de salida: 3 V

Funcionalidades extra: Interruptor de encendido

Precio: 2,41 €

Figura 45: Imagen del soporte de pila "LiliPad" [31].

ANEXO I.VII REGULADOR DE 3V A 5V

Modelo: Pololu 5V Step-Up/Step-Down Voltage Regulator S7V8F5 [92]. Figura 46.

Código de desarrollador PCB: reg09b

Otras marcas PCB: 0J7031

Dimensiones: 11 x 17 x 3 mm

Voltaje de entrada: de 2.7 V a 11.8 V

Voltaje de salida: 5 V fijos con +5/-3% de tolerancia

Corriente típica de salida: de 500 mA a 1 A

Protección integrada: Para sobrecalentamiento y cortocircuito

Figura 46: Imagen del regulador S7V8F5 [92].

Extra: Funcionalidad de ahorro de energía para corrientes bajas (menores a 0.2 mA) para aumentar eficiencia. En las figuras 47 y 48 se muestra su eficiencia y corriente según el voltaje.

Precio: 6,20 €

Gráficas de eficiencias y corrientes según voltaje:

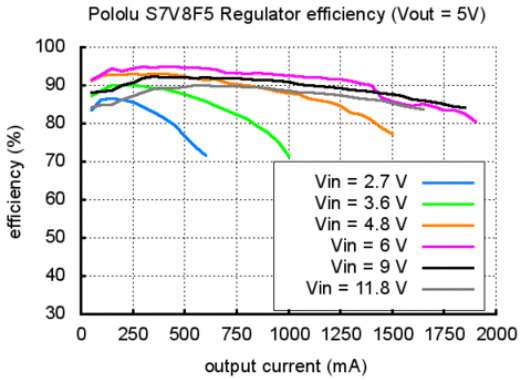


Figura 47: Eficiencia según voltaje y corriente [92].

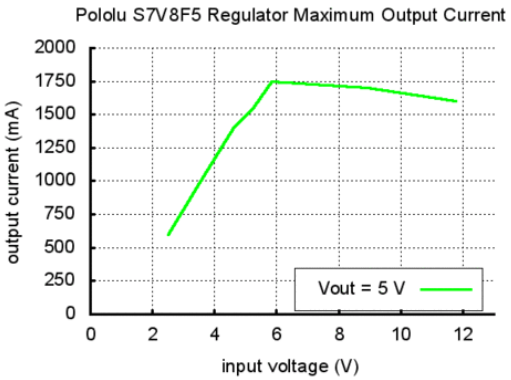


Figura 48: Corriente según voltaje [92].

Existen otros reguladores válidos para este proyecto, pero se selecciona éste entre todos debido a que es el que presenta mayores eficiencias de conversión para distintos voltajes y su mínimo admisible es de 2,7 V, suficiente para los 3 V teóricos que se obtienen de la pila. Algunos reguladores no llegan a esta cifra. Además, la diferencia de precio es mínima.

ANEXO I.VIII CÁMARA

Modelo: ZeroCam NOIR without infrared filter [93]. Figura 49.

Dimensiones: 60 x 11,4 x 5,1 mm

Resolución: Sensor de 5 MP, 2592x1944 pixeles

Lente: f2.9 con 3.60 mm de distancia focal

Campo de visión: 53,50 º en horizontal y 41,41 º en vertical

Vídeo: 1080p a 30 FPS (o 60 FPS a 720p; 90 FPS a 480p)

Funcionalidades extra: Conexión especializada para Raspberry Pi Zero. Sin sensor de infrarrojos.

Precio: 16,18 €

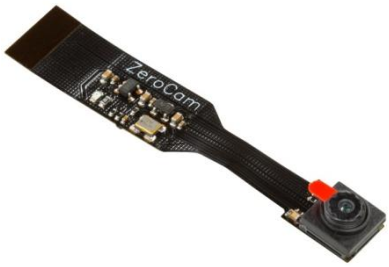


Figura 49: Imagen de la cámara ZeroCam NOIR [93].

ANEXO I.IX CONECTOR MACHO DE 40 PINES:

Debido a que para este proyecto se usa la Raspberry Pi Zero W y no el modelo “Zero WH”, que tiene los pines ya soldados, se requiere soldar un conector de 40 pines a la placa con el fin de conectar la tarjeta de sonido HAT. El usado se muestra en la figura 50.

Modelo: Male 40-pin Header COM1101 [94]

Dimensiones: 51 x 5 x 12 mm

Número de pines: 2 filas de 20 pines cada una

Precio: 1,15 €

Figura 50: Imagen del conector de 40 pines [94].

ANEXO I.X CARACTERÍSTICAS TOTALES:

Dimensiones del disp. interior: 9,45 x 3,2 x 2,7 cm

Dimensiones totales: 9,7 x 4,7 x 3,1 cm

Peso aproximado: 32 gramos

Salida de audio: Altavoces de 1 W (predet.), conexión mini-Jack de auriculares

Alimentación: Pila de botón de 20 mm de diám.

Posibles extras: Funcionalidad Bluetooth 4.1/ LE, control mediante voz.

Precio: 57,94 €

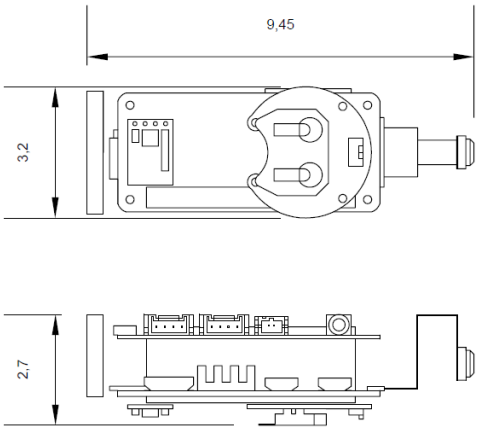


Figura 51: Dimensiones aproximadas del dispositivo interior.

En la figura 51 se muestra el interior del dispositivo y en las figuras 52 y 53 se muestra el dispositivo completo, con la carcasa ya incorporada, aunque sin la varilla extensible.

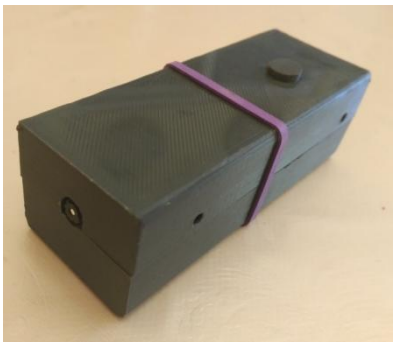


Figura 52: Ilustración real de la cara frontal del dispositivo completo.

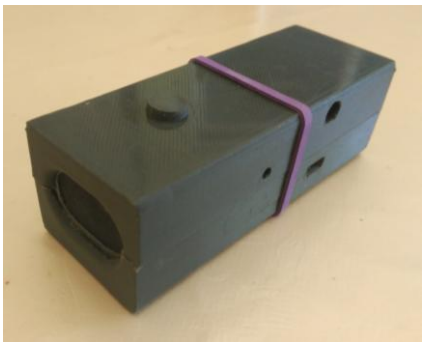


Figura 53: Ilustración real de la parte trasera del dispositivo completo.

ANEXO II. ARQUITECTURA DE LA RASPBERRY ZERO

En este apartado se muestran ilustraciones de varios planos técnicos de la Raspberry Pi Zero. En concreto, la figura 54 muestra la arquitectura del microprocesador tipo SoC y la figura 55 muestra la numeración GPIO de los pines de la Raspberry.

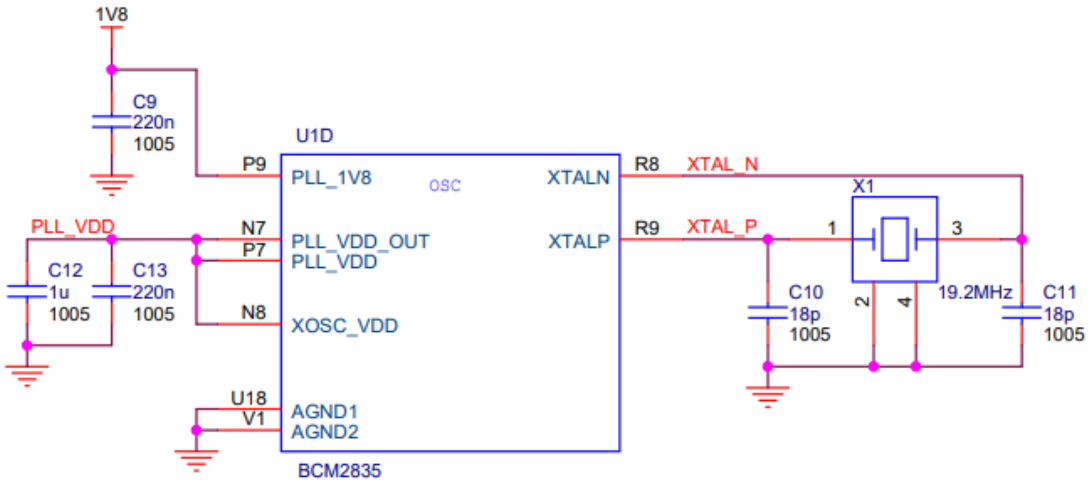


Figura 54: Arquitectura del microprocesador de la Raspberry [95].

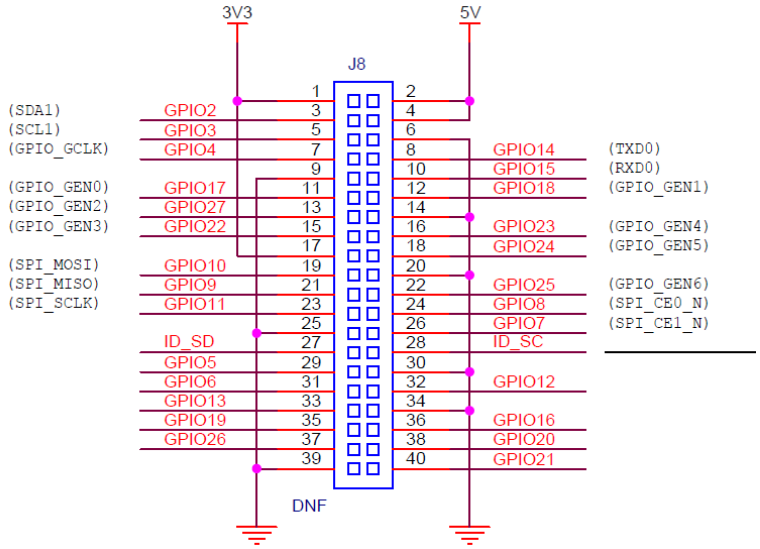


Figura 55: Numeración GPIO de los pines [95].

ANEXO III. INSTALACIÓN DEL SISTEMA OPERATIVO Y ELEMENTOS DE SOFTWARE

En este apartado se muestra todo el proceso de instalación, tanto del sistema operativo, como de cada elemento de software que usa, tanto en la ejecución de las funcionalidades del dispositivo, como para el desarrollo de éste.

ANEXO III.I INSTALACIÓN DEL SISTEMA OPERATIVO

Antes de realizar el montaje del dispositivo, la Raspberry se debe programar para ejecutar los comandos del lector y, para ello, se necesita instalar un sistema operativo en una tarjeta micro-SD e introducirla en la ranura preparada para este propósito en la Raspberry. En concreto, para este proyecto se utiliza el sistema operativo Raspbian, anteriormente conocido como "Raspberry Pi OS", debido a que es un sistema operativo creado exclusivamente para este tipo de dispositivos y resulta más eficiente y sencillo de usar que otros, pero se puede instalar otro sistema operativo como "Ubuntu" [96] si se prefiere.

El sistema operativo Raspbian cuenta con una versión de escritorio gráfico y una versión con únicamente el terminal de comandos. Para el modelo definitivo de este dispositivo se termina usando la versión que usa sólo la consola, ya que el dispositivo no incluye una pantalla y cargar el escritorio gráfico supone el empleo innecesario de recursos. Sin embargo, durante el desarrollo se usa la versión con escritorio para facilitar el proyecto, por lo que ésta es la versión de Raspbian descrita en este Trabajo Fin de Grado.

A continuación, se describe el proceso para la instalación de dicha versión del S.O.

1. Para instalar el sistema operativo en la tarjeta SD, se utiliza el software "Raspberry Pi Imager" [97], que se debe descargar desde la página web de Raspberry Pi OS, cuya dirección es: https://www.raspberrypi.org/software/.

2. Una vez dentro, se debe pinchar en el recuadro “Download for Windows”, marcado como “Paso A” en la figura 56.

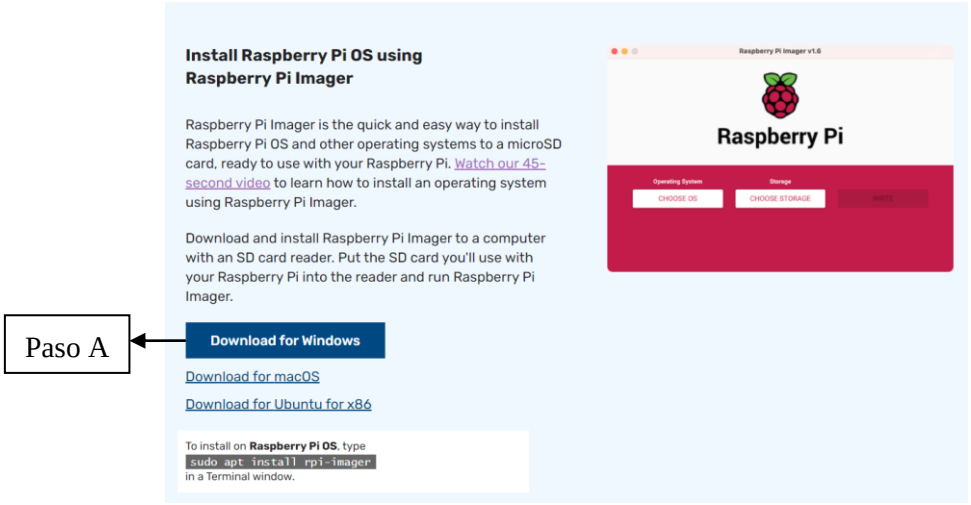


Figura 56: Página de descarga del instalador del sistema operativo.

3. En la carpeta de “Descargas” de tu equipo, se debe hacer doble clic sobre el archivo que se acaba de descargar y pulsar “Sí” cuando se demanden permisos de administrador. Posteriormente, se debe hacer clic sobre “install” y luego sobre “finish”.
4. A continuación, se debe iniciar el programa Raspberry Pi Imager instalado e introducir una tarjeta micro-SD en su equipo. El proceso que se usa en los siguientes pasos formatea por completo la tarjeta micro-SD introducida, por lo que es conveniente retirar previamente cualquier información importante que se desee conservar de su interior.
5. En el programa, se debe seleccionar la opción “Choose OS” que se encuentra señalada como “Paso B” en la figura 57.

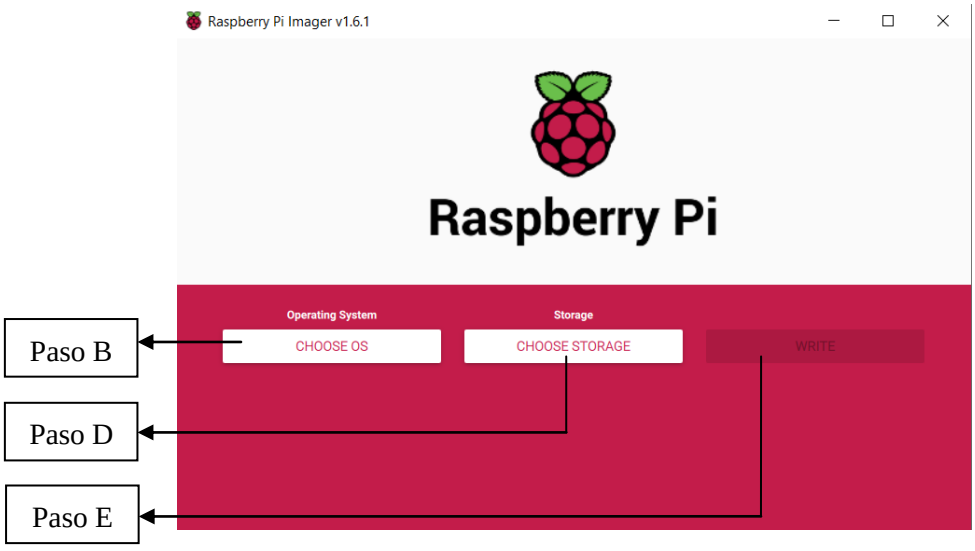


Figura 57: Raspberry Pi Imager.

6. En la ventana que se abre, se debe seleccionar la opción “Raspberry Pi OS (32-bit)”, señalada como “Paso C” en la figura 58. Si se quisiese seleccionar la versión sin entorno de escritorio, se debería seleccionar la opción “Raspberry Pi OS (other)” y posteriormente cliquear en “Raspberry Pi OS Lite (32-bit)”.

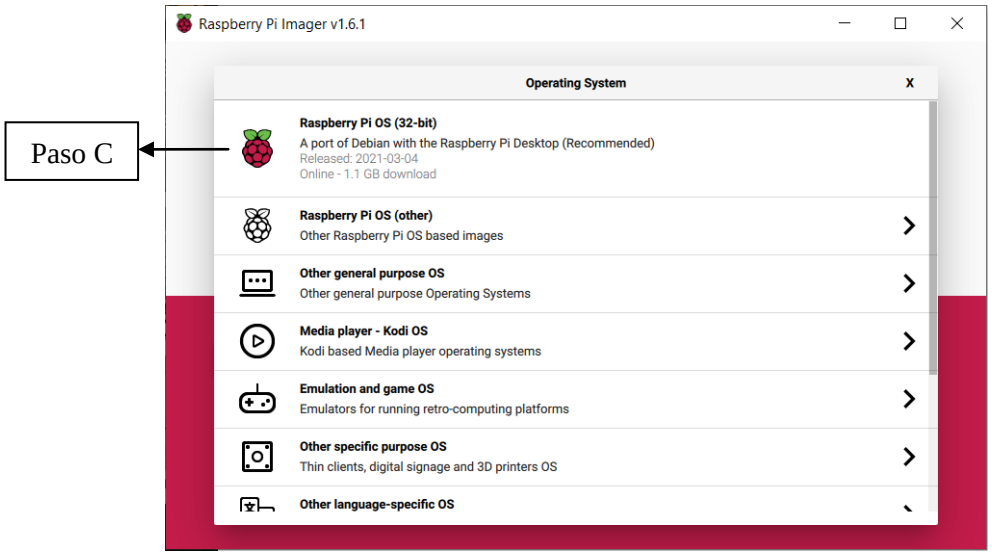


Figura 58: Opciones de sistema operativo del “Raspberry Pi Imager”.

7. A continuación, se ha de seleccionar la opción “Choose storage”, visible como “Paso D” en la figura 57, y seleccionar la dirección donde se encuentra la tarjeta micro-SD.
8. Posteriormente, se debe hacer clic en “Write”, que se puede en la figura 57 como “paso E”, y esperar hasta que el proceso complete.
9. Por último, se debe retirar la micro-SD de forma segura e introducirla en la Raspberry.

Una vez se conecta el dispositivo a la corriente o se enciende el interruptor de encendido, la Raspberry inicia y muestra las opciones de usuario, las cuales no se cubren en este informe.

ANEXO III.II COMANDOS DE INSTALACIÓN

Aunque algunos programas descritos en la sección 5.2.6 “Lista de los programas seleccionados” ya están instalados por defecto en el sistema operativo Raspbian, como ALSA Mixer o Thonny Python, otros necesitan una instalación. En Raspbian las instalaciones se hacen a través del terminal. En este apartado se describen los comandos necesarios para la instalación y configuración básica de los programas necesarios.

Cabe destacar que existen dos tipos de instalaciones distintas. Una se realizada a través de un comando tipo "`apt-get`", que instala el programa en el sistema operativo para darle un uso general dentro de la Raspberry, y la otra se realiza a través de comandos "`pip`", que instala el programa en la carpeta específica de Python para su uso exclusivo mediante dicho lenguaje.

Como en este proyecto usa un *script* programado para ejecutarse en Python, se intenta usar las instalaciones tipo `pip` siempre que se puede para facilitar el trabajo y reducir posibles errores. Es importante recordar que Thonny Python usa la versión 3.7 de Python, lo que significa que los comandos pip se deben instalar en la carpeta de esta versión. Para realizar esto, se necesita usar "`pip3`" al declarar el comando, en lugar de `pip`.

Otra cosa que es recurrente en los comandos que se muestran y, por lo tanto, es importante destacar, es el comando "`sudo`", que asigna permiso de administrador al comando al ejecutarse, permitiendo que el programa se instale con los permisos de administrador. Los pasos para la instalación de todos los programas usados se muestran en los siguientes apartados.

ANEXO III.II.I INSTALACIÓN DE TESSERACT

Los siguientes comandos instalan el Tesseract OCR de forma general y luego el módulo para uso específico de "Python3". La última línea instala el diccionario en español, ya que va a ser el idioma en el que van a estar los textos que se van a dictar en este proyecto. Se podría instalar otro si se desea [79].

```
$ sudo dpkg - -configure –a
$ sudo apt-get install tesseract-ocr
$ sudo pip3 install pytesseract
$ sudo apt-get install tesseract-ocr-spa
```

ANEXO III.II.II INSTALACIÓN DE OPENCV

Para realizar la instalación de OpenCV, se muestran a continuación las líneas de instalación.

```
$ sudo apt install libatlas3-base libsz2 libharfbuzz0b libtiff5 libjasper1 libopenexr22 libilmbase23 libgstreamer1.0-0 libavcodec58 libavformat58 libavutil56 libswscale4 libqtgui4 libqt4-test libqtcore4
$ sudo pip3 install opencv-contrib-python==4.4.0.46 libwebp6
$ sudo apt-get install libopencv-dev python3-opencv
```

Las primeras tres líneas están destinadas a la instalación de las librerías necesarias para la instalación y uso del `OpenCV-contrib-python` [98].

El segundo comando instala la versión 4.4.0.46 de `OpenCV-contrib-python`. El motivo de usar esta versión es que la última versión no funciona en la arquitectura "buster" de Linux [99], que es la que tiene la Raspberry Pi Zero que vamos a utilizar, como se puede ver en la figura 59.

Por último, la última línea instala OpenCV. Esto es necesario para el correcto funcionamiento de la versión usada en Python, ya que usa archivos y recursos de la versión general.

Version	Released	Stretch	Buster
4.5.2.52	2021-05-07	⊖	⊖
4.5.1.48	2021-01-02	✗	✗
4.4.0.46	2020-11-02	✗	✓
4.4.0.44	2020-09-22	⊖	⊖
4.4.0.42	2020-08-16	⊖	⊖

Figura 59: Versiones de OpenCV-contrib-python y las arquitecturas que lo soportan [99].

ANEXO III.II.III INSTALACIÓN DEL CONTROLADOR DE LA TARJETA DE SONIDO

Para la instalación del controlador de la tarjeta de sonido es necesario cambiar el *kernel* del sistema operativo, por lo que, si se siguen los siguientes comandos, éste se desactualiza a la versión 1.20200819-1, ya que el controlador de la placa de sonido HAT que se va a utilizar no es compatible con el kernel que viene por defecto, sin embargo, puede que en siguientes versiones de kernel no sea necesario realizar este cambio. Si se planea utilizar la Raspberry para otro fin a parte del de este proyecto que requiera el uso del kernel más reciente, se debe valorar si se quiere sustituir esta tarjeta de sonido HAT por otra más reciente o, como alternativa, usar una tarjeta micro-SD distinta para el proyecto que lo requiera, ya que el kernel forma parte del sistema operativo y no es aconsejable cambiarlo de forma intermitente dentro de la misma tarjeta.

A continuación, se muestran los comandos para la instalación del driver del controlador de la tarjeta en el kernel de Linux.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ git clone https://github.com/respeaker/seeed-voicecard.git
$ cd seeed-voicecard
$ sudo ./install.sh –compat-kernel
$ reboot
```

Las primeras dos líneas buscan actualizaciones para el sistema operativo y las instalan si existen, la tercera descarga el paquete de instalación de la dirección web especificada, la cuarta lleva el directorio de la consola a la dirección específica marcada para realizar la instalación del driver con el comando encontrado en la línea siguiente (que es el comando que también desactualizará el "kernel") y la última inicia el proceso de reinicio de la "Raspberry", lo que hace que el driver comience a funcionar cuando se arranque el kernel del sistema operativo [91].

ANEXO III.II.IV INSTALACIÓN DE PICO TTS

A continuación, se muestran los comandos para la instalación de Pico TTS, donde los primeros dos comandos descargan los dos archivos de instalación, de dos direcciones web distintas, que posteriormente se instalan con el último [100]. Se deben escribir los comandos tal y como se representan, incluyendo los guiones y los espacios.

```
$ wget http://archive.raspberrypi.org/debian/pool/main/s/svox/libttspico-utils_1.0+git20130326-3+rpi1_armhf.deb
$ wget http://archive.raspberrypi.org/debian/pool/main/s/svox/libttspico0_1.0+git20130326-3+rpi1_armhf.deb
$ apt-get install -f ./libttspico0_1.0+git20130326-3+rpi1_armhf.deb ./libttspico-utils_1.0+git20130326-3+rpi1_armhf.deb
```

ANEXO III.II.V INSTALACIÓN DEL BOTÓN DE LA TARJETA DE SONIDO

El siguiente comando instala la funcionalidad de botón de la tarjeta de sonido. En concreto, el botón usa el puerto GPIO17 de la Raspberry Pi Zero para funcionar [91]. Parte de la arquitectura de la Raspberry Pi Zero se puede encontrar en el Anexo II.

```
$ sudo pip3 install rpi.gpio
```

ANEXO III.II.VI INSTALACIÓN DE MEJORA DE DIFFLIB

La librería "Difflib" [101] es una herramienta incluida en Python que permite la comparación de textos. Se usa para la comparación de resultados generados con distintas configuraciones de OpenCV para ver si se produce, o no, alguna mejora en el reconocimiento del texto. "SequenceMatcher" es el proceso principal que usa esta librería [102].

```
$ pip3 install cdifflib
```

Este comando instala una utilidad que sustituye el proceso SequenceMatcher de la librería original de Difflib por un equivalente escrito en C++, siendo cuatro veces más rápida [103].

ANEXO III.II.VII INSTALACIÓN DE JIWER

La librería Jiwer es usada para la comparación del texto reconocido con respecto al original mediante el cálculo del WER, MER y WIL. Es una librería que se usa exclusivamente durante el desarrollo y no resulta necesaria si solo se quiere replicar el dispositivo. El comando de instalación para esta librería se muestra a continuación [81].

```
$ pip3 install jiwer
```

ANEXO III.II.VIII CONFIGURACIÓN DE ALSA MIXER

Para que se pueda reproducir el sonido correctamente por todos los puertos de salida que contiene la tarjeta de sonido HAT, es necesario configurar el nivel de decibelios de cada una, ya que, por defecto, algunas salidas marcan decibelios nulos.

```
$ alsamixer
$ sudo alsactl store
```

Para conseguir esto se inicia ALSA Mixer escribiendo la primera línea y se selecciona la tarjeta de sonido "seeed-2mic-voicecard" con la tecla "F6". A continuación, una vez se ve por pantalla una imagen similar a la figura 60, se configuran los niveles usando las flechas de dirección laterales para desplazarse entre los dispositivos y las de arriba y abajo para subir o bajar el volumen.

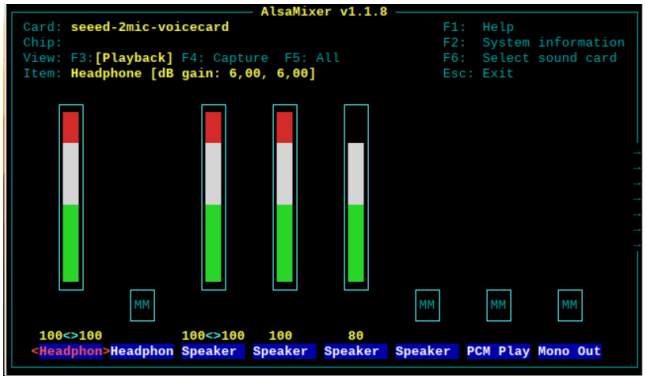


Figura 60: Configuración del "ALSA Mixer".

Una vez completada la configuración, se debe cerrar el programa y escribir la segunda línea en la barra de comandos, que guarda esta configuración para que se pueda utilizar en el futuro [69].

ANEXO III.II.IX CONFIGURACIÓN DE VLC MEDIA PLAYER

VLC media player, de manera predeterminada, no puede usarse desde el modo *root* y, puesto que el dispositivo final ejecuta el código de forma automática al inicio usando el modo root, es necesario introducir el comando a continuación para que VLC se pueda usar [104].

```
$ sudo sed -i 's/geteuid/getppid/' /usr/bin/vlc
```

Además, puesto que se usa un pitido en el código para notificar cuándo el dispositivo puede comenzar el proceso de captura, es necesario descargar el archivo de sonido de dicho pitido, posicionarlo dentro del directorio principal de la Raspberry, el cual es `/home/pi`, y llamar al archivo "beep-02.wav", si no se llama ya así. El archivo usado para este proyecto se puede conseguir a través del siguiente enlace: https://www.soundjay.com/buttons/beep-02.wav

También se usa un sonido para indicar que se ha realizado la captura, el cual se debe llamar "camera-shutter-click-03.wav". Puede conseguirse a través del siguiente enlace: https://www.soundjay.com/mechanical/camera-shutter-click-03.wav

ANEXO III.II.X CONFIGURACIÓN PARA EL ARRANQUE DEL CÓDIGO AL INICIO

Por último, con el fin de ejecutar el código de forma automática al encender el dispositivo, es necesario seguir el procedimiento descrito en este apartado. Es aconsejable ejecutar los pasos a continuación una vez se haya comprobado que el dispositivo funciona correctamente usando el entorno gráfico y un compilador de código como Thonny Python para evitar posibles errores.

Primero, se abre el terminal y se abre el archivo ".bashrc" de la Raspberry que, si se tiene habilitada la opción de iniciar el dispositivo con la consola en vez de con el entorno gráfico, se ejecuta al inicio. Esto se hace introduciendo el código mostrado a continuación.

```
$ sudo nano /home/pi/.bashrc
```

Una vez abierto el archivo, se deben escribir, en la última línea de éste, un par de comandos, donde el primero habilita que el segundo comando se ejecute al inicio, y el segundo comando ejecuta el código general del dispositivo. Una vez escritos, se debe pulsar "Ctrl+X", luego "S" y por último "Intro". La figura 61 muestra el archivo con los comandos, que se muestran a continuación, aplicados [105].

```
$ echo Running at boot
$ sudo python /home/pi/CodigoGeneral.py
```

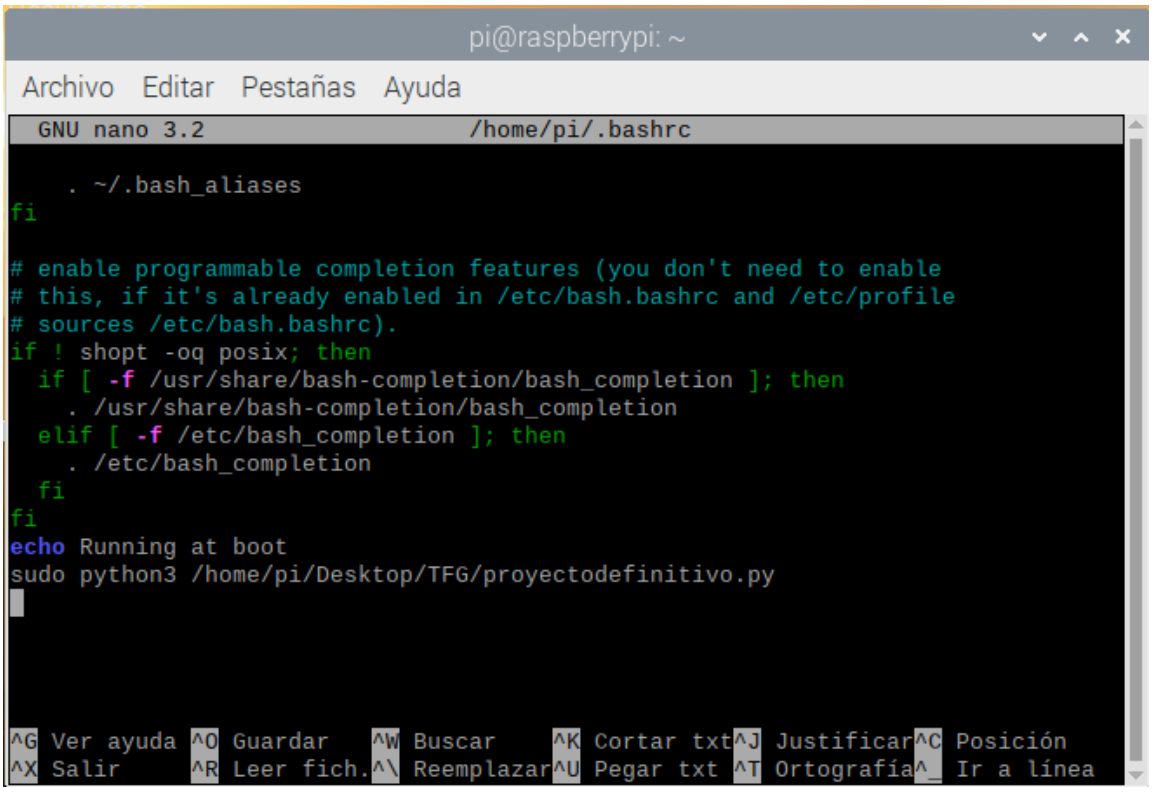


Figura 61: Archivo .bashrc con la modificación aplicada.

Como último paso, se configura la Raspberry para iniciarse a través del terminal, en vez de a través del entorno gráfico, iniciando el programa “configuración de Raspberry Pi” y, en la opción de “inicio” de la pestaña de “sistema”, seleccionando “a consola”.

ANEXO IV. CÓDIGOS EMPLEADOS

En este apartado se expone tanto el código que se emplea en el desarrollo del dispositivo, como el código realizado durante el desarrollo para averiguar las alteraciones realizables a la imagen capturada que dan los mejores resultados de reconocimiento del OCR.

ANEXO IV.I CÓDIGO GENERAL

A continuación, se expone el código general de Python que se emplea en el dispositivo, junto con anotaciones que definen el propósito de cada comando. El código se muestra a continuación.

```
from picamera import PiCamera # Importa la librería que controla la cámara.
from time import sleep # Importa la librería que permite programar con los tiempos con facilidad.
import pytesseract # Importa la librería que controla el "Tesseract OCR".
import cv2 # Importa la librería de "OpenCV", que permite la alteración de las imágenes.
from PIL import Image # Importa la funcionalidad que nos permite leer imágenes de forma fácil.
import RPi.GPIO as GPIO # Importa la librería que sirve para controlar el botón.
import os # Importa la consola de la "Raspberry" para poder usarla desde el propio código.

def variación_contraste_adaptativo(file_path): # Función: Modifica la imagen variando el contraste.
    img = cv2.imread(file_path, 0) # Carga la imagen en gris.
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8)) # Crea el contraste adaptativo.
   # Este contraste, en concreto, divide la imagen en secciones y varía el contraste de cada una.
   # Es más fiable que variar el contraste de forma general.
    cl1 = clahe.apply(img) # Aplica el contraste.
    return cl1 # Devuelve el resultado de la función.

os.system('sudo alsactl restore') # Restaura la configuración de "ALSA Mixer".

GPIO.setmode(GPIO.BCM) # Configura "GPIO" en la numeración "BCM".
GPIO.setup(17, GPIO.IN) # Inicializa el botón (es 17 porque usa el puerto "GPIO17").

while (1): # Inicia un bucle infinito que mantenga al dispositivo ejecutando continuamente
    # el programa y evitar, así, que se tengan que importar las librerías de nuevo.

    # Inicia el programa.
    os.system('cvlc --play-and-exit /home/pi/beep-02.wav') # Reproduce un pitido.
    camera = PiCamera() # Inicia el proceso de la cámara de la "Raspberry".
    camera.rotation = 90 # Modifica la posición de la imagen 90º a la izquierda [106].
    i=0 # Aplica la condición del while para que sólo salga cuando haya pulsación de botón.
```

```
while (i == 0):
    boton = GPIO.input(17) # Lee el botón.
    if boton: # Si no se produce pulsación del botón.
        sleep(0.001) # Reposa durante un milisegundo para evitar abrumar el procesador.
        continue # Continúa el bucle “while”.
    else: # Si se produce pulsación del botón.
        sleep(2) # Espera dos segundos para que la cámara pueda ajustarse.
        camera.capture('/home/pi/image1.jpg') # Hace una captura.
        i=1 # Sale del bucle “while”.
        os.system('cvlc --play-and-exit /home/pi/camera-shutter-click-03.wav') # Reproduce
        # un sonido que indica que se ha realizado la fotografía.

camera.close() # Apago la cámara para ahorrar batería [107].

Direccion_archivo = '/home/pi/image1.jpg' # Variable.
# Esta variable contendrá la dirección en la que está guardada la captura antes tomada.

img = variación_contraste_adaptativo(Direccion_archivo) # Llama a la función de contraste.

original=pytesseract.image_to_string(img, config='-l spa --oem 3 --psm 1') # Inicia el OCR.

os.system('pico2wave -l es-ES -w grabacion.wav "%s"' %original) # Inicia el programa TTS.
# En concreto, pasa el texto a un archivo de sonido “.wav”.
# “PICO TTS” no tiene una funcionalidad en “python” sencilla, por lo que se usará a través de
# la consola

os.system('cvlc --play-and-exit /home/pi/grabacion.wav') # Reproduce el audio.
# “CVLC” abre el programa “VLC” sin opciones gráficas.
# La opción “play and exit” cierra automáticamente el reproductor una vez ha terminado
# su función, permitiéndole continuar al código, que volverá a los inicios del bucle “while”.
# La razón por la que se usa el comando mediante consola en vez de una librería de “Python”
# se describirá a continuación.
```

Existe una librería para Python del VLC media player que se puede usar. Se debe instalar a través de la consola introduciendo el comando mostrado a continuación.

```
pip3 install python-vlc
```

Donde el código para usarla en el proyecto es el mostrado a continuación.

```
Import vlc # Importa la librería que controla el "VLC Media Player".
audio = vlc.MediaPlayer('/home/pi/grabacion.wav') # Introduce el archivo
# a reproducir en el "VLC".
audio.audio_set_volume(80) #Reduce teóricamente el volumen, pero no funciona.
audio.play() # Reproduce el archivo de audio.
sleep(1.5) # Espera un segundo y medio para que inicie la grabación sin problemas.
duration = audio.get_length() / 1000 # Establece una duración del audio a reproducir
sleep(duration) # Hace que el código se detenga hasta que la reproducción se complete
```

Sin embargo, el uso de esta librería produce acoplamientos en el audio. Después de numerosos intentos para hacer una reproducción correcta a través de la reducción de volumen y procesos similares, al final se opta por usar la versión base ejecutada a través del terminal, que reproduce el archivo de forma nítida y sin problemas.

ANEXO IV.II CÓDIGO EMPLEADO PARA LAS PRUEBAS CON OPENCV

En este apartado, se expone el código que se usa para implementar todas las pruebas incluidas en el apartado 7.1 "Resultados de las pruebas con OpenCV". El código se muestra a continuación.

```
import tempfile # Importa una librería que permite crear archivos temporales.
import pytesseract # Importa el "OCR".
import cv2 # Importa "OpenCV".
import numpy as np # Importa una librería que permite usar matrices.
from PIL import Image # Importa la funcionalidad que nos permite leer imágenes de forma fácil.
from cdifflib import CSequenceMatcher # Importa la mejora del "difflib" realizada en "C++".
import difflib # Importa el comparador de textos.
```

```
import jiwer # Importa el comparador de textos que otorga WER, MER y WIL.
import sys # Importa el sistema, que permite usar comandos del intérprete desde este código.
from textosorig import texto_original # Importa la variable de salida de una función definida
# en otro código descrito más abajo.

IMAGE_SIZE = 1800 # Declara el tamaño objetivo.
BINARY_THREHOLD = 180 # Declara el nivel de binarización.

# Función: Combina la reducción de ruido y la binarización tipo “Otsu” con el cambio de “DPI”.
def process_image_for_ocr(file_path):
    temp_filename = set_image_dpi(file_path) #Llama a la función para cambiar “DPI”.
    im_new = remove_noise_and_smooth(temp_filename) #Llama a la función de quitar ruido
# y binarización tipo “Otsu”.
    return im_new # Devuelve el resultado.

# Función: Cambio de “DPI”.
def set_image_dpi(file_path):
    im = Image.open(file_path) #Carga la imagen.
    length_x, width_y = im.size #Analiza los tamaños por defecto de la imagen cargada y los guarda.
    factor = max(1, int(IMAGE_SIZE / length_x)) #Crea un factor de proporción de la imagen.
    size = factor * length_x, factor * width_y # Usa el factor para aumentar el tamaño.
    im_resized = im.resize(size, Image.ANTIALIAS) # Cambia el tamaño de la imagen
# al tamaño aumentado y aplica un suavizado de bordes.
    temp_file = tempfile.NamedTemporaryFile(delete=False, suffix='.jpg') # Crea un archivo temp.
    temp_filename = temp_file.name # Asigna un nombre al archivo temporal.
    im_resized.save(temp_filename, dpi=(300, 300)) # Guarda en el archivo temporal la imagen
# con los “DPI” objetivo asignados.
    return temp_filename # Devuelve el resultado.

# Función: Binarización tipo “Otsu” para suavizar.
def image_smoothening(img):
# Primero se hace una binarización inicial y luego se realiza la binarización tipo “Otsu”:
    ret1, th1 = cv2.threshold(img, BINARY_THREHOLD, 255, cv2.THRESH_BINARY)
    ret2, th2 = cv2.threshold(th1, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```

```
    blur = cv2.GaussianBlur(th2, (1, 1), 0) # Se hace un suavizado y se repite la binarización.
    ret3, th3 = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    return th3 # Devuelve el resultado.


# Función: Eliminación de ruido y suavizado.
def remove_noise_and_smooth(file_name):
    img = cv2.imread(file_name, 0) # Carga la imagen en gris.
    filtered=cv2.adaptiveThreshold(img.astype(np.uint8),255,
cv2.ADAPTIVE_THRESH_MEAN_C,  cv2.THRESH_BINARY,  41, 3) # Algunas binarizaciones
# básicas, como la binarización de la media de una zona menos la constante “C”.
    kernel = np.ones((1, 1), np.uint8) # Crea una matriz de números 1.
    opening = cv2.morphologyEx(filtered, cv2.MORPH_OPEN, kernel) # Elimina el ruido.
    closing = cv2.morphologyEx(opening, cv2.MORPH_CLOSE, kernel) # Suaviza la imagen.
    img = image_smoothening(img) # Llama a la función de binarizado tipo “Otsu”.
    or_image = cv2.bitwise_or(img, closing) # Guarda la imagen binarizada o la suavizada si la
 # binarización no pudo realizarse.
    return or_image # Devuelve el resultado.


# Función: Pasar la imagen a gama de grises.
def pasar_a_gris(file_path):
    img = cv2.imread(file_path, 0) # Carga la imagen en gris.
    return img # Devuelve el resultado.


# Función: Binarización gaussiana.
def binarización_gaussiana(file_path):
    img = cv2.imread(file_path, 0) # Carga la imagen en gris.
    th3 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 11, 2) # Realiza la binarización gaussiana.
    return th3 # Devuelve el resultado.


# Función: Cambio del contraste de forma adaptativa tipo “CLAHE”.
def variación_contraste_adaptativo(file_path):
    img = cv2.imread(file_path, 0) # Carga la imagen en gris.
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8)) # Crea el contraste adaptativo.
```

```
    cl1 = clahe.apply(img) # Aplica el contraste tipo “CLAHE”.
    return cl1# Devuelve el resultado.


# Elegir la imagen a modificar eliminando el comentario en la línea de la imagen que se desee:
#Libro borroso: Asigna la dirección de la imagen “libroborroso” y el valor 1 a “NumeroImagen”.
# Direccion_archivo = '/home/pi/Desktop/imagenes/libroborroso.jpg'; NumeroImagen = 1
# Libro de teatro: Hace lo mismo, pero con la imagen “teatro” y el valor 2.
# Direccion_archivo = '/home/pi/Desktop/imagenes/teatro.jpg'; NumeroImagen = 2
# Texto largo:
# Direccion_archivo = '/home/pi/Desktop/imagenes/texto1.jpg'; NumeroImagen = 3
# Texto largo de cerca:
# Direccion_archivo = '/home/pi/Desktop/imagenes/texto1cerca.jpg'; NumeroImagen = 4
# Texto largo más lejos:
# Direccion_archivo = '/home/pi/Desktop/imagenes/texto1lejos.jpg'; NumeroImagen = 5
# Texto largo con una fuente de luz aplicada:
# Direccion_archivo = '/home/pi/Desktop/imagenes/texto1luz.jpg'; NumeroImagen = 3
# Texto corto con una gran imagen:
# Direccion_archivo = '/home/pi/Desktop/imagenes/textoconilustracion.jpg'; NumeroImagen = 6
# Texto corto y grande con una gran imagen:
# Direccion_archivo = '/home/pi/Desktop/imagenes/textoconilustracion2.jpg'; NumeroImagen = 7


# Elegir la modificación a aplicar:
# Si solo queremos cambiar los dpi:
# ImagenModif = set_image_dpi(Direccion_archivo)  # Llama a la función correspondiente.
# Si solo queremos que emborrone (quita puntos por quitar ruido):
# ImagenModif = image_smoothening(Direccion_archivo)
# Si solo queremos que suavice y quite ruido:
# ImagenModif = remove_noise_and_smooth(Direccion_archivo)
# Si queremos que haga todo lo anterior:
# ImagenModif = process_image_for_ocr(Direccion_archivo)
# Si queremos que pase solo a gris:
# ImagenModif = pasar_a_gris(Direccion_archivo)
# Si queremos una binarización gaussiana:
# ImagenModif = binarización_gaussiana(Direccion_archivo)
```

```
# Para hacer la variación de contraste:
# ImagenModif = variación_contraste_adaptativo(Direccion_archivo)


# Analiza con el OCR:
TextoReconoc = pytesseract.image_to_string(ImagenModif,config='-l spa --oem 3 --psm 1')


# Llama a la función que contiene el texto original (en otro archivo) usando el valor asignado
# a “NumeroImagen” como entrada:
textoOrig = texto_original (NumeroImagen)
lineas1 = TextoReconoc.splitlines() # Divide cada párrafo en líneas para el análisis del “difflib”.
lineas2 = textoOrig.splitlines() # Hace lo mismo, pero con la variable del texto original.


diferencia = difflib.Differ() # Llama a una función del “difflib” y la renombra como “diferencia”.
generador_diferencia = diferencia.compare(lineas1, lineas2) # Compara el texto del OCR
# con el original.


print ('\n'.join(generador_diferencia)) # Muestra los resultados de Difflib por pantalla.
print ("WER")
print(jiwer.wer(lineas2, lineas1)) # Muestra el valor de WER por pantalla.
print ("MER")
print(jiwer.mer(lineas2, lineas1)) # Muestra el valor de MER por pantalla.
print ("WIL")
print(jiwer.wil(lineas2, lineas1)) # Muestra el valor de WIL por pantalla.
```

El código de arriba llama a una función “texto_original” que no se encuentra en el código. Esta función se encuentra dentro de otro archivo programado en Python, localizado en la misma carpeta, llamado “textosorig.py”, del cual se importa a este código mediante el comando visto al inicio de este código, el cual se expone a continuación.

```
from textosorig import texto_original
```

El archivo que se menciona antes contiene los textos originales que usa Difflib para hacer la comparación. El código que contiene este archivo se puede ver a continuación.

```
# Texto original de la imagen “libroborroso”:
libroborrosoOrig = """LOS CINCO EN LAS ROCAS DEL DIABLO 63
```

```
no tener que ir a buscar agua todos los días. ¡Qué bien lo
pasamos!
-Bueno, pues me parece que esta vez aún te lo pasarás
mejor - dijo Dick -, Porque ahora estás mejor acompañado.
Estoy seguro de que entonces te sentías un poco solo.
-Es verdad. Pero tenía conmigo a Travieso -repuso
Manitas.
Cuando el monito oyó pronunciar su nombre, saltó a los
brazos del niño y se acurrucó en ellos, encantado.
-¿Y cuál es la siguiente habitación de este maravilloso
faro? -preguntó Julián.
-Sólo queda una más, la habitación del fanal -explicó
Manitas-. Os la enseñaré. Antes era la habitación más im-
portante del faro, pero ahora no se usa. Está completamente
olvidada y abandonada. Venid a verla.
¡Qué orgulloso se mostraba Manitas de su faro!"""

# Texto original de la imagen "textoconilustracion":
textoconilustracionOrig = """Capítulo Catorce
La bella historia de Rut"""
# Etc.: El resto de textos se omitirán para ahorrar espacio, ya que no aportan más valor informativo.

def texto_original (NumeroImagen): # Función: Asigna la variable según el texto solicitado.
    if NumeroImagen == 1:
        textoOrig = libroborrosoOrig
    elif NumeroImagen == 2:
        textoOrig = teatroOrig
    elif NumeroImagen == 3:
        textoOrig = texto1Orig
    elif NumeroImagen == 4:
        textoOrig = texto1cercaOrig
    elif NumeroImagen == 5:
        textoOrig = texto1lejosOrig
    elif NumeroImagen == 7:
```

```
    textoOrig = textoconilustracion2Orig
  else:
    textoOrig = textoconilustracionOrig
  return textoOrig # Devuelve el resultado.
```

ANEXO V. IMÁGENES EMPLEADAS PARA ANALIZAR OPENCV

En este apartado se muestran las imágenes que se usan para realizar las pruebas de OpenCV. En concreto, contiene las imágenes correspondientes a "libroborroso", "teatro", "texto1", "texto1cerca", "texto1lejos", "texto1luz", "textoconilustracion" y "textoconilustracion2" representados en las figuras de la 62 a la 69, respectivamente.

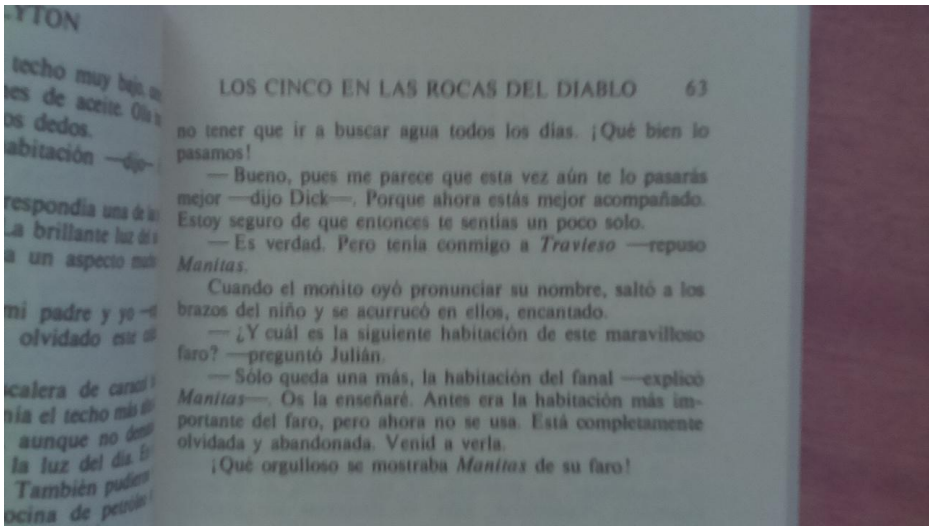


Figura 62: Imagen correspondiente a "libroborroso".

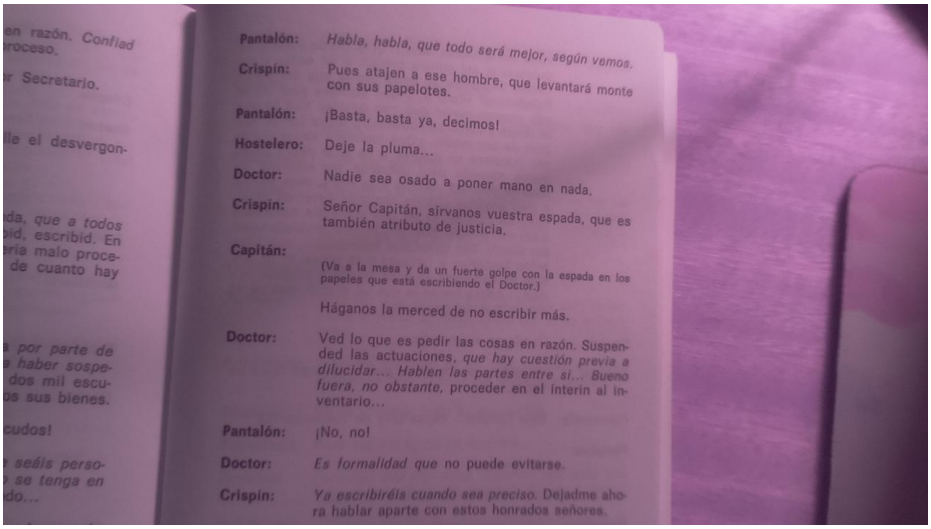
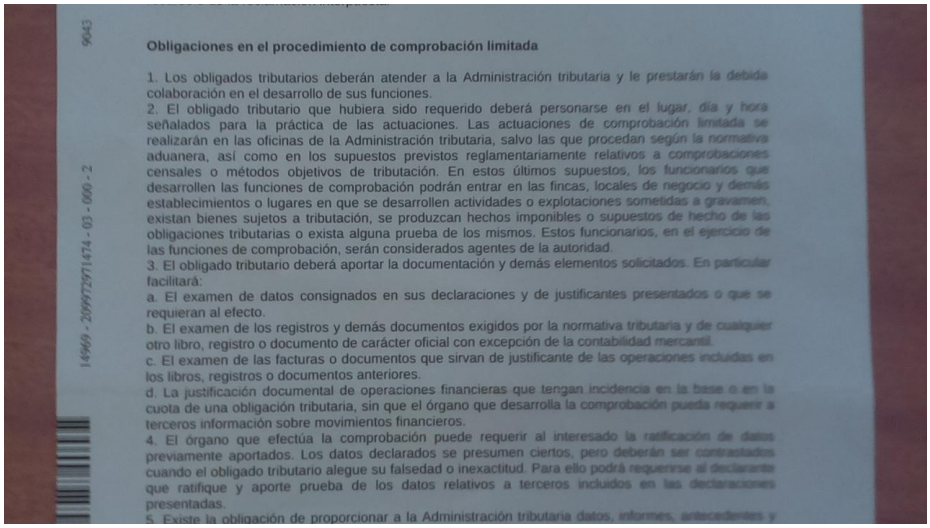


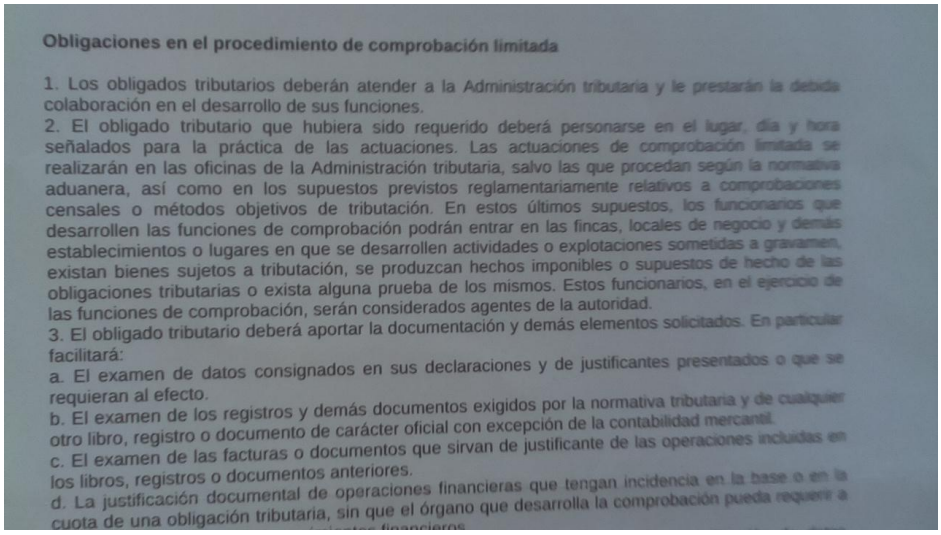
Figura 63: Imagen correspondiente a "teatro".



Obligaciones en el procedimiento de comprobación limitada

1. Los obligados tributarios deberán atender a la Administración tributaria y le prestarán la debida colaboración en el desarrollo de sus funciones.
2. El obligado tributario que hubiera sido requerido deberá personarse en el lugar, día y hora señalados para la práctica de las actuaciones. Las actuaciones de comprobación limitada se realizarán en las oficinas de la Administración tributaria, salvo las que procedan según la normativa aduanera, así como en los supuestos previstos reglamentariamente relativos a comprobaciones censales o métodos objetivos de tributación. En estos últimos supuestos, los funcionarios que desarrollen las funciones de comprobación podrán entrar en las fincas, locales de negocio y demás establecimientos o lugares en que se desarrollen actividades o explotaciones sometidas a gravamen, existan bienes sujetos a tributación, se produzcan hechos imponibles o supuestos de hecho de las obligaciones tributarias o exista alguna prueba de los mismos. Estos funcionarios, en el ejercicio de las funciones de comprobación, serán considerados agentes de la autoridad.
3. El obligado tributario deberá aportar la documentación y demás elementos solicitados. En particular facilitará:
a. El examen de datos consignados en sus declaraciones y de justificantes presentados o que se requieran al efecto.
b. El examen de los registros y demás documentos exigidos por la normativa tributaria y de cualquier otro libro, registro o documento de carácter oficial con excepción de la contabilidad mercantil.
c. El examen de las facturas o documentos que sirvan de justificante de las operaciones incluidas en los libros, registros o documentos anteriores.
d. La justificación documental de operaciones financieras que tengan incidencia en la base o en la cuota de una obligación tributaria, sin que el órgano que desarrolla la comprobación pueda requerir a terceros información sobre movimientos financieros.
4. El órgano que efectúa la comprobación puede requerir al interesado la ratificación de datos previamente aportados. Los datos declarados se presumen ciertos, pero deberán ser contrastados cuando el obligado tributario alegue su falsedad o inexactitud. Para ello podrá requerirse al declarante que ratifique y aporte prueba de los datos relativos a terceros incluidos en las declaraciones presentadas.
5. Existe la obligación de proporcionar a la Administración tributaria datos, informes, antecedentes y

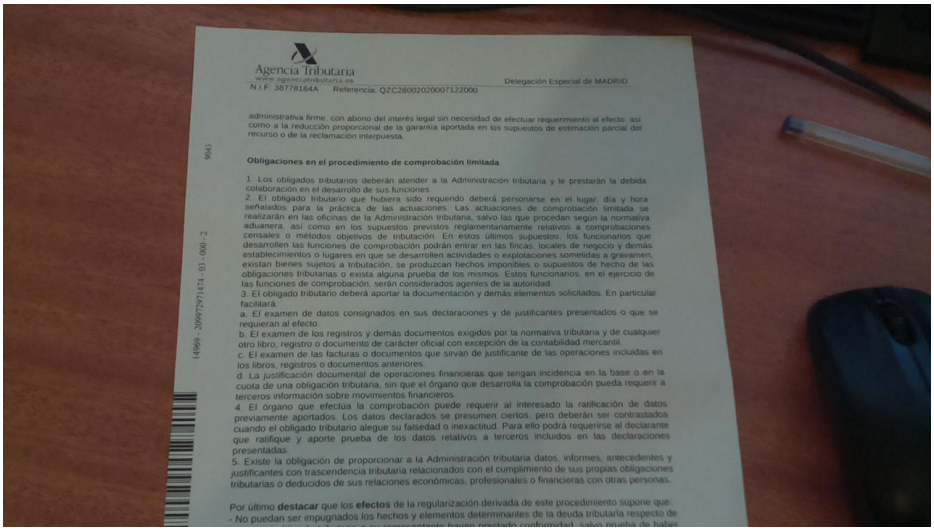
Figura 64: Imagen correspondiente a "texto1".



Obligaciones en el procedimiento de comprobación limitada

1. Los obligados tributarios deberán atender a la Administración tributaria y le prestarán la debida colaboración en el desarrollo de sus funciones.
2. El obligado tributario que hubiera sido requerido deberá personarse en el lugar, día y hora señalados para la práctica de las actuaciones. Las actuaciones de comprobación limitada se realizarán en las oficinas de la Administración tributaria, salvo las que procedan según la normativa aduanera, así como en los supuestos previstos reglamentariamente relativos a comprobaciones censales o métodos objetivos de tributación. En estos últimos supuestos, los funcionarios que desarrollen las funciones de comprobación podrán entrar en las fincas, locales de negocio y demás establecimientos o lugares en que se desarrollen actividades o explotaciones sometidas a gravamen, existan bienes sujetos a tributación, se produzcan hechos imponibles o supuestos de hecho de las obligaciones tributarias o exista alguna prueba de los mismos. Estos funcionarios, en el ejercicio de las funciones de comprobación, serán considerados agentes de la autoridad.
3. El obligado tributario deberá aportar la documentación y demás elementos solicitados. En particular facilitará:
a. El examen de datos consignados en sus declaraciones y de justificantes presentados o que se requieran al efecto.
b. El examen de los registros y demás documentos exigidos por la normativa tributaria y de cualquier otro libro, registro o documento de carácter oficial con excepción de la contabilidad mercantil.
c. El examen de las facturas o documentos que sirvan de justificante de las operaciones incluidas en los libros, registros o documentos anteriores.
d. La justificación documental de operaciones financieras que tengan incidencia en la base o en la cuota de una obligación tributaria, sin que el órgano que desarrolla la comprobación pueda requerir a

Figura 65: Imagen correspondiente a "texto1cerca".



Agencia Tributaria
www.agenciatributaria.es
N.I.F. 38778164A Referencia: QZC280020200007122000
Delegación Especial de MADRID

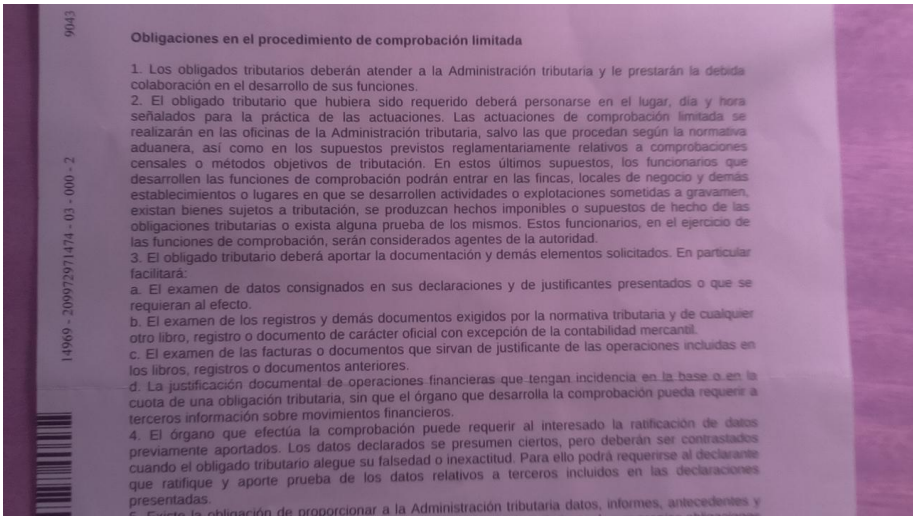
administrativa firme, con abono del interés legal sin necesidad de efectuar requerimiento al efecto, así como a la reducción proporcional de la garantía aportada en los supuestos de estimación parcial del recurso o de la reclamación interpuesta.

Obligaciones en el procedimiento de comprobación limitada

1. Los obligados tributarios deberán atender a la Administración tributaria y le prestarán la debida colaboración en el desarrollo de sus funciones.
2. El obligado tributario que hubiera sido requerido deberá personarse en el lugar, día y hora señalados para la práctica de las actuaciones. Las actuaciones de comprobación limitada se realizarán en las oficinas de la Administración tributaria, salvo las que procedan según la normativa aduanera, así como en los supuestos previstos reglamentariamente relativos a comprobaciones censales o métodos objetivos de tributación. En estos últimos supuestos, los funcionarios que desarrollen las funciones de comprobación podrán entrar en las fincas, locales de negocio y demás establecimientos o lugares en que se desarrollen actividades o explotaciones sometidas a gravamen, existan bienes sujetos a tributación, se produzcan hechos imponibles o supuestos de hecho de las obligaciones tributarias o exista alguna prueba de los mismos. Estos funcionarios, en el ejercicio de las funciones de comprobación, serán considerados agentes de la autoridad.
3. El obligado tributario deberá aportar la documentación y demás elementos solicitados. En particular facilitará:
a. El examen de datos consignados en sus declaraciones y de justificantes presentados o que se requieran al efecto.
b. El examen de los registros y demás documentos exigidos por la normativa tributaria y de cualquier otro libro, registro o documento de carácter oficial con excepción de la contabilidad mercantil.
c. El examen de las facturas o documentos que sirvan de justificante de las operaciones incluidas en los libros, registros o documentos anteriores.
d. La justificación documental de operaciones financieras que tengan incidencia en la base o en la cuota de una obligación tributaria, sin que el órgano que desarrolla la comprobación pueda requerir a terceros información sobre movimientos financieros.
4. El órgano que efectúa la comprobación puede requerir al interesado la ratificación de datos previamente aportados. Los datos declarados se presumen ciertos, pero deberán ser contrastados cuando el obligado tributario alegue su falsedad o inexactitud. Para ello podrá requerirse al declarante que ratifique y aporte prueba de los datos relativos a terceros incluidos en las declaraciones presentadas.
5. Existe la obligación de proporcionar a la Administración tributaria datos, informes, antecedentes y justificantes con trascendencia tributaria relacionados con el cumplimiento de sus propias obligaciones tributarias o deducidos de sus relaciones económicas, profesionales o financieras con otras personas.

Por último **destacar** que los **efectos** de la regularización derivada de este procedimiento supone que:
- No puedan ser impugnados los hechos y elementos determinantes de la deuda tributaria respecto de

Figura 66: Imagen correspondiente a "texto1lejos".



Obligaciones en el procedimiento de comprobación limitada

1. Los obligados tributarios deberán atender a la Administración tributaria y le prestarán la debida colaboración en el desarrollo de sus funciones.
2. El obligado tributario que hubiera sido requerido deberá personarse en el lugar, día y hora señalados para la práctica de las actuaciones. Las actuaciones de comprobación limitada se realizarán en las oficinas de la Administración tributaria, salvo las que procedan según la normativa aduanera, así como en los supuestos previstos reglamentariamente relativos a comprobaciones censales o métodos objetivos de tributación. En estos últimos supuestos, los funcionarios que desarrollen las funciones de comprobación podrán entrar en las fincas, locales de negocio y demás establecimientos o lugares en que se desarrollen actividades o explotaciones sometidas a gravamen, existan bienes sujetos a tributación, se produzcan hechos imponibles o supuestos de hecho de las obligaciones tributarias o exista alguna prueba de los mismos. Estos funcionarios, en el ejercicio de las funciones de comprobación, serán considerados agentes de la autoridad.
3. El obligado tributario deberá aportar la documentación y demás elementos solicitados. En particular facilitará:
a. El examen de datos consignados en sus declaraciones y de justificantes presentados o que se requieran al efecto.
b. El examen de los registros y demás documentos exigidos por la normativa tributaria y de cualquier otro libro, registro o documento de carácter oficial con excepción de la contabilidad mercantil.
c. El examen de las facturas o documentos que sirvan de justificante de las operaciones incluidas en los libros, registros o documentos anteriores.
d. La justificación documental de operaciones financieras que tengan incidencia en la base o en la cuota de una obligación tributaria, sin que el órgano que desarrolla la comprobación pueda requerir a terceros información sobre movimientos financieros.
4. El órgano que efectúa la comprobación puede requerir al interesado la ratificación de datos previamente aportados. Los datos declarados se presumen ciertos, pero deberán ser contrastados cuando el obligado tributario alegue su falsedad o inexactitud. Para ello podrá requerirse al declarante que ratifique y aporte prueba de los datos relativos a terceros incluidos en las declaraciones presentadas.

14969 - 20997297147A - 03 - 000 - 2

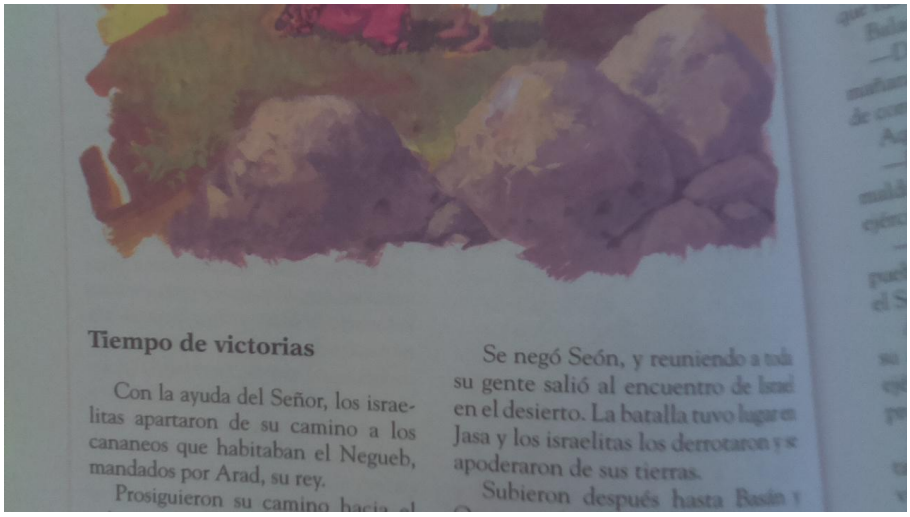
9043

Figura 67: Imagen correspondiente a "texto1luz".

CAPÍTULO CATORCE

La bella historia de Rut

Figura 68: Imagen correspondiente a "textoconilustracion".



Tiempo de victorias

Con la ayuda del Señor, los israelitas apartaron de su camino a los cananeos que habitaban el Negueb, mandados por Arad, su rey.

Prosiguieron su camino hacia el

Se negó Seón, y reuniendo a toda su gente salió al encuentro de Israel en el desierto. La batalla tuvo lugar en Jasa y los israelitas los derrotaron y se apoderaron de sus tierras.

Subieron después hasta Basán y

Figura 69: Imagen correspondiente a "textoconilustracion2".